

COBOL2002 言語 拡張仕様編

文法書

4010-1J-805-10

COBOL2002

前書き

■ 対象製品

- P-2636-2354 COBOL2002 Net Developer 05-00 (適用 OS : Windows 10(x64), Windows 11, Windows Server 2019, Windows Server 2022)
- P-2436-6354 COBOL2002 Net Server Suite 05-00 (適用 OS : Windows Server 2019, Windows Server 2022)
- P-2636-4354 COBOL2002 Net Client Suite 05-00 (適用 OS : Windows 10(x64), Windows 11)
- P-2936-2354 COBOL2002 Net Developer(64) 05-00 (適用 OS : Windows 10(x64), Windows 11, Windows Server 2019, Windows Server 2022)
- P-2936-6354 COBOL2002 Net Server Suite(64) 05-00 (適用 OS : Windows Server 2019, Windows Server 2022)
- P-9W36-1251 COBOL2002 Net Server Suite(64) 05-00 (適用 OS : Linux Server 9 (64-bit x86_64))
- P-2636-7354 COBOL2002 Developer Professional 05-00 (適用 OS : Windows 10(x64), Windows 11, Windows Server 2019, Windows Server 2022)
- P-2936-7354 COBOL2002 Developer Professional(64) 05-00 (適用 OS : Windows 10(x64), Windows 11, Windows Server 2019, Windows Server 2022)

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

COBOL2002 は、COBOL2002 規格 (ISO/IEC 1989:2002) の主な機能に対応しています。COBOL85 互換機能 (コンパイラオプション) 指定時、COBOL2002 は、COBOL85 規格 (ISO85, ANSI85, JIS88, JIS92) の上位水準に準拠します。

■ 輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

本書には、X/Open の許諾に基づき X/Open CAE Specification COBOL Language (XO/CAE/91/200 ISBN 1-872630-09-X) Copyright (C) December 1991, X/Open Company Limited の内容が含まれています。

事前に著作権所有者の許諾を得ずに、本書の該当部分を複製、複写および転記することは禁じられています。

■ 発行

2025年4月 4010-1J-805-10

■ 著作権

All Rights Reserved. Copyright (C) 2025, Hitachi, Ltd.

変更内容

変更内容 (4010-1J-805-10) COBOL2002 Net Server Suite(64) 05-00

追加・変更内容	変更箇所
<p>次に示す適用 OS をサポート対象外とした。</p> <ul style="list-style-type: none">• AIX V7.1• AIX V7.2• AIX 7.3• Red Hat Enterprise Linux Server 7 (64-bit x86_64)• Red Hat Enterprise Linux Server 8 (64-bit x86_64)	—

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、次に示すプログラムプロダクトの機能および COBOL 言語の文法について説明したものです。

- P-2636-2354 COBOL2002 Net Developer
- P-2436-6354 COBOL2002 Net Server Suite
- P-2636-4354 COBOL2002 Net Client Suite
- P-2936-2354 COBOL2002 Net Developer(64)
- P-2936-6354 COBOL2002 Net Server Suite(64)
- P-9W36-1251 COBOL2002 Net Server Suite(64)
- P-2636-7354 COBOL2002 Developer Professional
- P-2936-7354 COBOL2002 Developer Professional(64)

■ 対象読者

このマニュアルは、COBOL2002 の機能を知りたい方、または COBOL2002 の日立拡張仕様の文法規則を知りたい方を対象としています。また、COBOL の基本的な言語仕様と、UNIX または Windows の操作方法について理解していることを前提としています。

■ X/Open 仕様情報の記述箇所

次の部分で X/Open の仕様情報を記述しています。

- 「12. 画面節 (SCREEN SECTION) による画面機能」

■ サポート機能一覧

規格

機能	製品種別		
	AIX COBOL 2002	Linux COBOL 2002	Windows COBOL 2002
基本機能	○	○	○
順ファイル	○	○	○
相対ファイル	○	○	○

機能	製品種別		
	AIX COBOL 2002	Linux COBOL 2002	Windows COBOL 2002
索引ファイル	○	○	○
整列併合	○	○	○
プログラム間連絡	○	○	○
組み込み関数	○	○	○
オブジェクト指向	○	○	○
共通例外処理	○	○	○
再帰呼び出し	○	○	○
利用者定義関数	○	○	○
局所場所節 (LOCAL-STORAGE SECTION)	○	○	○
原始文操作	○	○	○
自由形式正書法	○	○	○
TYPDEF 句と SAME AS 句	○	○	○
翻訳指令	○	○	○
区分化*	○	○	○

(凡例)

○：サポートしている

注※

覚え書きとしてサポートしている。

X/Open

機能		製品種別			
		AIX COBOL2 002	Linux COBOL2 002	Windows(x86) COBOL2 002	Windows(x64) COBOL2 002
テキストファイル		○	○	○	○
ファイル共用 (ファイルシェア)	順ファイル	○	○	○	○
	相対ファイル	○	○	○	○

機能	製品種別			
	AIX COBOL2 002	Linux COBOL2 002	Window s(x86) COBOL2 002	Window s(x64) COBOL2 002
索引 (ISAM) ファイル	○	○	○	○
	○	○	×	×
	—	—	○	×
コマンド行および環境変数へのアクセス	○	○	○	○
画面節 (SCREEN SECTION) による画面操作	○	×	○	○
C 言語インタフェース	○	○	○	○
インターナショナルリゼーション※	○	○	—	—

(凡例)

- ：サポートしている
- ×：サポートしていない
- ：該当しない

注※

詳細については、マニュアル「COBOL2002 使用の手引 手引編」の-International オプションの説明を参照のこと。

拡張機能

機能	製品種別					
	AIX(32) COB OL2 002	AIX(64) COB OL2 002	Linu x(x8 6) COB OL2 002	Linu x(x6 4) COB OL2 002	Win dow s(x8 6) COB OL2 002	Win dow s(x6 4) COB OL2 002
日本語	○	○	○	○	○	○
ブール (ビット処理)	○	○	○	○	○	○
アドレス操作	○	○	○	○	○	○
1 バイト 2 進および COMP-X 項目	○	○	○	○	○	○
浮動小数点データ	○	○	○	○	○	○
索引 (ISAM) ファイル機能の拡張 (合成キー, 逆順読み)	○	○	○	○	○	○
CSV ファイル	○	○	○	○	○	○

機能		製品種別					
		AIX(32) COBOL2002	AIX(64) COBOL2002	Linux(x86) COBOL2002	Linux(x64) COBOL2002	Windows(x86) COBOL2002	Windows(x64) COBOL2002
HiRDBによる索引ファイル		○	○	×	×	○	○
Btrieveによる索引ファイル		—	—	—	—	○	×
リモートファイルアクセス		×	×	×	×	○	×
ラージファイル入出力	順ファイル	○	○	○	○	○	○
	相対ファイル	×	×	×	×	×	×
	索引 (ISAM) ファイル	×	×	×	×	○	×
	テキストファイル	○	○	○	○	○	○
	CSV ファイル	○	○	○	○	○	○
	HiRDBによる索引ファイル	×	×	×	×	×	×
	Btrieveによる索引ファイル	—	—	—	—	×	×
COBOL 入出力サービスルーチン		○	○	○	○	○	○
バイトストリーム入出力サービスルーチン		○	○	○	○	○	○
ファイル入出力拡張機能	ファイルサイズがレコード長の整数倍でない固定長形式の順ファイル入出力	○	○	○	○	×	×
	ファイルバッファサイズ指定機能	○	○	○	○	×	×
画面節 (WINDOW SECTION) による画面操作	画面節 (WINDOW SECTION)	○	○	×	×	○	○
	JCPOPUP サービスルーチン	○	○	×	×	○	○
通信節による画面操作 (XMAP3)		○	×	×	×	○	○
COPY 文の接頭辞/接尾辞		○	○	○	○	○	○
プリンタへのアクセス	XMAP3による印刷	○	×	×	×	○	×
	GDI モード印刷	—	—	—	—	○	○
	ESC/P モード印刷	—	—	—	—	○	○
ファイルのディスク書き込み保証 (書き込み時)	順ファイル	○	○	○	○	○	○
	相対ファイル	○	○	○	○	○	○

機能		製品種別					
		AIX(32) COBOL2002	AIX(64) COBOL2002	Linux(x86) COBOL2002	Linux(x64) COBOL2002	Windows(x86) COBOL2002	Windows(x64) COBOL2002
	索引 (ISAM) ファイル	○	○	○	○	○	○
	テキストファイル	×	×	×	×	×	×
	CSV ファイル	×	×	×	×	×	×
	HiRDB による索引ファイル	×	×	×	×	×	×
	Btrieve による索引ファイル	—	—	—	—	×	×
ファイルのディスク書き込み保証 (クローズ時)	順ファイル	○	○	○	○	○	○
	相対ファイル	○	○	○	○	○	○
	索引 (ISAM) ファイル	○	○	○	○	×	×
	テキストファイル	○	○	○	○	×	×
	CSV ファイル	○	○	○	○	○	○
	HiRDB による索引ファイル	×	×	×	×	×	×
	Btrieve による索引ファイル	—	—	—	—	×	×
報告書作成機能		○	○	○	○	○	○
MIOS7 COBOL85 との互換機能		×	×	×	×	○	△
イベントログファイル/syslog ファイル出力機能		×	×	×	×	○	○
データコミュニケーション機能		○	○	○	○	○	○
データベース操作機能 (ODBC インタフェース)		—	—	○	○	○	○
XDM によるデータベースシミュレーション機能	構造型データベース (XDM/SD)	○	○	○	○	○	○
	リレーショナルデータベース (XDM/RD)	○	○	○	○	○	○
OLE2 オートメーション機能	クライアント機能	—	—	—	—	○	○
CGI プログラム作成支援機能		○	×	×	×	×	×
マルチスレッド環境	作成と実行	○	○	○	○	○	○
	デバッグ	△	△	△	△	○	○
MSMQ アクセス機能		—	—	—	—	○	○

機能	製品種別					
	AIX(32) COBOL2002	AIX(64) COBOL2002	Linux(x86) COBOL2002	Linux(x64) COBOL2002	Windows(x86) COBOL2002	Windows(x64) COBOL2002
エンディアン切り替え	×	×	○	○	○	○
Unicode 機能	○	○	○	○	○	○
数字項目のけた拡張機能	×	○	×	○	×	○
動的長基本項目機能	○	○	○	○	○	○
定数長拡張機能	英数字定数長の拡張		○	○	○	○
Java プログラム呼び出し機能	×	×	×	×	○	○

(凡例)

- ：サポートしている
- ×
- △：サポートしている機能であるが、使える機能に一部制限がある
- ：該当しない

デバッグ機能

機能		製品種別		
		AIX COBOL2002	Linux COBOL2002	Windows COBOL2002
実行時デバッグ機能		○	○	○
テストデバッグ機能	GUI モード	×	×	○
	バッチモード	○	○	○
	ラインモード	○	○	×
カバレッジ機能	GUI モード	×	×	○
	バッチモード	○	○	○

(凡例)

- ：サポートしている
- ×

連携機能

機能	製品種別				
	AIX(32) COBOL 2002	AIX(64) COBOL 2002	Linux(x8 6) COBOL 2002	Linux(x6 4) COBOL 2002	Window s COBOL 2002
XML 連携機能	○	○	○	○	○
Cosminexus 連携機能	×	○	×	○	○

(凡例)

- ：サポートしている
- ×：サポートしていない

使用を推奨しない機能について

次の機能は、旧製品との互換を維持するために提供しています。前提となるソフトウェアは販売を終了しています。COBOL2002 の新しい機能への移行を推奨します。

- Btrieve (Pervasive.SQL) による索引編成ファイル
- MIOS7 COBOL85 との互換機能

■ 文法の記号

このマニュアルの構文の説明で使用する記号については、マニュアル「COBOL2002 言語 標準仕様編」を参照してください。

■ プログラム例について

このマニュアルのプログラム例は、断り書きがない場合は Windows(x86) COBOL2002 用です。プログラム例を Windows(x86) COBOL2002 以外で使用するには、プログラムの記述に変更が必要な場合がありますのでご注意ください。

■ COBOL 規格仕様について

COBOL は、事務処理用に最も使用されているプログラミング言語です。当初はビジネス向けプログラミング言語として開発されましたが、第 1 次規格 (ANSI68, ISO72, JIS72)、第 2 次規格 (ANSI74, ISO78, JIS80)、第 3 次規格 (ISO85, ANSI85, JIS88, JIS92)、第 4 次規格 (ISO/IEC 1989:2002, JIS X3002:2011) と規格の改訂を重ね、現在ではビジネス向けだけでなく、汎用プログラミングにも広がっています。

なお、COBOL に関する公式の規格は、ISO (国際規格)、ANSI (アメリカ規格)、および JIS (日本工業規格) で決められています。

目次

前書き	2
変更内容	4
はじめに	5

1	日本語機能	28
1.1	日本語機能とは	29
1.2	全般規則 (日本語機能)	30
1.2.1	日本語文字集合 (日本語機能)	30
1.2.2	文字列 (日本語機能)	30
1.2.3	分離符 (日本語機能)	34
1.2.4	データ項目の字類と項類 (日本語機能)	34
1.2.5	標準けた寄せ規則 (日本語機能)	34
1.3	コーディング規則 (日本語機能)	36
1.3.1	行と文字位置 (日本語機能)	36
1.3.2	行の継続 (日本語機能)	36
1.3.3	注記行 (日本語機能)	36
1.4	見出し部 (日本語機能)	37
1.5	環境部 (日本語機能)	38
1.5.1	特殊名段落 (日本語機能)	38
1.5.2	入出力管理段落の APPLY FORMS-OVERLAY 句 (日本語機能)	38
1.6	データ部 (日本語機能)	39
1.6.1	CHARACTER TYPE 句 (日本語機能)	39
1.6.2	GROUP-USAGE 句 (日本語機能)	39
1.6.3	JUSTIFIED 句 (日本語機能)	41
1.6.4	PICTURE 句 (日本語機能)	42
1.6.5	USAGE 句 (日本語機能)	43
1.6.6	VALUE 句 (日本語機能)	44
1.7	手続き部 (日本語機能)	46
1.7.1	単純条件 (日本語機能)	46
1.7.2	INITIALIZE 文 (日本語機能)	47
1.7.3	INSPECT 文 (日本語機能)	47
1.7.4	MOVE 文 (日本語機能)	48
1.7.5	SEARCH 文 (日本語機能)	49
1.7.6	STRING 文 (日本語機能)	49
1.7.7	UNSTRING 文 (日本語機能)	50

2	テキストファイル機能 51
2.1	テキスト編成 52
2.2	環境部 (テキストファイル機能) 53
2.2.1	ファイル管理段落 (テキストファイル機能) 53
2.3	データ部 (テキストファイル機能) 55
2.3.1	ファイル記述項 (テキストファイル機能) 55
2.4	手続き部 (テキストファイル機能) 56
2.4.1	CLOSE 文 (テキストファイル機能) 56
2.4.2	OPEN 文 (テキストファイル機能) 56
2.4.3	READ 文 (テキストファイル機能) 56
2.4.4	REWRITE 文 (テキストファイル機能) 56
2.4.5	WRITE 文 (テキストファイル機能) 57
2.5	入出力状態の値 (テキストファイル機能) 58
3	ファイル共用機能 59
3.1	環境部 (ファイル共用機能) 60
3.1.1	ファイル記述項 (ファイル共用機能) 60
3.2	手続き部 (ファイル共用機能) 63
3.2.1	CLOSE 文 (ファイル共用機能) 63
3.2.2	DELETE 文 (ファイル共用機能) 63
3.2.3	OPEN 文 (ファイル共用機能) 63
3.2.4	READ 文 (ファイル共用機能) 64
3.2.5	REWRITE 文 (ファイル共用機能) 66
3.2.6	START 文 (ファイル共用機能) 66
3.2.7	UNLOCK 文 (ファイル共用機能) 67
3.2.8	WRITE 文 (ファイル共用機能) 67
4	索引ファイル機能の拡張 68
4.1	機能概要 (索引ファイル機能の拡張) 69
4.1.1	合成キー 69
4.1.2	レコード下降順アクセス機能 69
4.2	環境部 (索引ファイル機能の拡張) 70
4.2.1	ファイル管理記述項 (FILE-CONTROL) (索引ファイル機能の拡張) 70
4.3	データ部 (索引ファイル機能の拡張) 74
4.3.1	DATA FORMAT 句 (索引ファイル機能の拡張) 74
4.4	手続き部 (索引ファイル機能の拡張) 75
4.4.1	READ 文 (索引ファイル機能の拡張) 75
4.4.2	START 文 (索引ファイル機能の拡張) 78

5	ブール形式データを扱う機能 82	
5.1	ブール定数 (ブール形式データを扱う機能)	83
5.2	表意定数 (ブール形式データを扱う機能)	84
5.3	データの表現形式 (ブール形式データを扱う機能)	85
5.3.1	ブール項目 (ブール形式データを扱う機能)	85
5.3.2	標準けた寄せ規則 (ブール形式データを扱う機能)	85
5.3.3	データ配置の方法 (ブール形式データを扱う機能)	86
5.4	データ部 (ブール形式データを扱う機能)	87
5.4.1	JUSTIFIED 句 (ブール形式データを扱う機能)	87
5.4.2	OCCURS 句 (ブール形式データを扱う機能)	87
5.4.3	PICTURE 句 (ブール形式データを扱う機能)	87
5.4.4	REDEFINES 句 (ブール形式データを扱う機能)	88
5.4.5	RENAMES 句 (ブール形式データを扱う機能)	89
5.4.6	SYNCHRONIZED 句 (ブール形式データを扱う機能)	89
5.4.7	USAGE 句 (ブール形式データを扱う機能)	91
5.4.8	VALUE 句 (ブール形式データを扱う機能)	91
5.5	手続き部 (ブール形式データを扱う機能)	93
5.5.1	文に共通な規則 (ブール形式データを扱う機能)	93
5.5.2	ブール形式データが書ける文 (ブール形式データを扱う機能)	95
6	浮動小数点形式データを扱う機能 100	
6.1	浮動小数点数字定数	101
6.2	表意定数 (浮動小数点形式データを扱う機能)	102
6.3	データの表現形式 (浮動小数点形式データを扱う機能)	103
6.3.1	外部浮動小数点形式	103
6.3.2	内部浮動小数点形式	103
6.4	算術符号 (浮動小数点形式データを扱う機能)	105
6.5	標準けた寄せ規則 (浮動小数点形式データを扱う機能)	106
6.6	データの配置 (浮動小数点形式データを扱う機能)	107
6.6.1	データ配置の方法	107
6.6.2	語長の境界と効率	107
6.7	データ部 (浮動小数点形式データを扱う機能)	108
6.7.1	PICTURE 句 (浮動小数点形式データを扱う機能)	108
6.7.2	SYNCHRONIZED 句 (浮動小数点形式データを扱う機能)	109
6.7.3	USAGE 句 (浮動小数点形式データを扱う機能)	109
6.7.4	VALUE 句 (浮動小数点形式データを扱う機能)	110
6.8	手続き部 (浮動小数点形式データを扱う機能)	112
6.8.1	文に共通な指定と規則 (浮動小数点形式データを扱う機能)	112
6.8.2	浮動小数点形式データが書ける文	115

7	報告書作成機能の拡張 119	
7.1	報告書作成機能の VOS3 COBOL85 からの移行用オプション (-CompatIV3)	120
7.1.1	特殊レジスタ (報告書作成機能の拡張)	120
7.1.2	環境部 (報告書作成機能の拡張)	121
7.1.3	データ部 (報告書作成機能の拡張)	121
7.1.4	手続き部 (報告書作成機能の拡張)	133
7.2	報告書作成機能の HITAC8000 シリーズからの移行用オプション (-Cblctr)	135
7.2.1	CBL-CTR 特殊レジスタ (報告書作成機能の拡張)	135
8	データコミュニケーション機能 137	
8.1	データコミュニケーション機能とは	138
8.2	データ部 (データコミュニケーション機能)	139
8.2.1	通信記述項 (CD) (データコミュニケーション機能)	139
8.3	手続き部 (データコミュニケーション機能)	142
8.3.1	COMMIT 文 (データコミュニケーション機能)	142
8.3.2	DISABLE 文 (データコミュニケーション機能)	142
8.3.3	ENABLE 文 (データコミュニケーション機能)	143
8.3.4	RECEIVE 文 (データコミュニケーション機能)	143
8.3.5	ROLLBACK 文 (データコミュニケーション機能)	144
8.3.6	SEND 文 (データコミュニケーション機能)	144
9	データベースアクセス機能 145	
9.1	概説 (データベースアクセス機能)	146
9.2	埋め込み SQL 文 (データベースアクセス機能)	147
9.3	SQL の概要 (データベースアクセス機能)	149
9.3.1	概念 (データベースアクセス機能)	149
9.3.2	全般規則 (データベースアクセス機能)	150
9.4	データ部 (埋め込み SQL 宣言節) (データベースアクセス機能)	154
9.4.1	SQLCODE 変数 (データベースアクセス機能)	155
9.4.2	埋め込み変数 (データベースアクセス機能)	156
9.4.3	標識変数 (データベースアクセス機能)	158
9.5	手続き部 (共通要素) (データベースアクセス機能)	160
9.5.1	値指定と相手指定 (データベースアクセス機能)	160
9.5.2	列指定 (データベースアクセス機能)	160
9.5.3	集合関数指定 (データベースアクセス機能)	161
9.5.4	値式 (データベースアクセス機能)	162
9.5.5	述語 (データベースアクセス機能)	162
9.5.6	比較述語 (データベースアクセス機能)	163
9.5.7	BETWEEN 述語 (データベースアクセス機能)	163

9.5.8	IN 述語 (データベースアクセス機能)	164
9.5.9	LIKE 述語 (データベースアクセス機能)	164
9.5.10	NULL 述語 (データベースアクセス機能)	165
9.5.11	限定述語 (データベースアクセス機能)	166
9.5.12	EXISTS 述語 (データベースアクセス機能)	166
9.5.13	探索条件 (データベースアクセス機能)	166
9.5.14	表式 (データベースアクセス機能)	167
9.5.15	副問い合わせ (データベースアクセス機能)	168
9.5.16	問い合わせ指定 (データベースアクセス機能)	169
9.5.17	問い合わせ式 (データベースアクセス機能)	169
9.6	手続き部 (宣言系) (データベースアクセス機能)	170
9.6.1	WHENEVER (埋め込み例外宣言) 文 (データベースアクセス機能)	170
9.6.2	DECLARE CURSOR (カーソル宣言) 文 (データベースアクセス機能)	172
9.7	手続き部 (制御系) (データベースアクセス機能)	174
9.7.1	コネクション (データベースアクセス機能)	174
9.7.2	トランザクション (データベースアクセス機能)	174
9.7.3	CONNECT 文 (データベースアクセス機能)	175
9.7.4	DISCONNECT 文 (データベースアクセス機能)	176
9.7.5	SET CONNECTION 文 (データベースアクセス機能)	177
9.7.6	COMMIT 文 (データベースアクセス機能)	177
9.7.7	ROLLBACK 文 (データベースアクセス機能)	178
9.8	手続き部 (操作系) (データベースアクセス機能)	179
9.8.1	手続き部 (操作系) の概要 (データベースアクセス機能)	179
9.8.2	SELECT 文 (単一行) (データベースアクセス機能)	180
9.8.3	INSERT 文 (データベースアクセス機能)	181
9.8.4	DELETE 文 (データベースアクセス機能)	181
9.8.5	UPDATE 文 (データベースアクセス機能)	182
9.8.6	OPEN 文 (データベースアクセス機能)	183
9.8.7	FETCH 文 (データベースアクセス機能)	183
9.8.8	CLOSE 文 (データベースアクセス機能)	184
9.8.9	CALL 文 (データベースアクセス機能)	184
9.8.10	EXECUTE IMMEDIATE 文 (データベースアクセス機能)	185
9.8.11	PREPARE 文 (データベースアクセス機能)	185
9.8.12	DEALLOCATE PREPARE 文 (データベースアクセス機能)	186
9.8.13	EXECUTE 文 (データベースアクセス機能)	186
10	コマンド行のアクセス	188
10.1	コマンド行, 環境変数へのアクセス	189
10.1.1	環境部 (SPECIAL-NAMES) (コマンド行, 環境変数へのアクセス)	189

10.1.2	手続き部 (コマンド行, 環境変数へのアクセス)	189
10.2	コマンド行へのアクセス (一括取得)	192
10.2.1	環境部 (SPECIAL-NAMES) (コマンド行へのアクセス (一括取得))	192
10.2.2	手続き部 (コマンド行へのアクセス (一括取得))	192

11 通信節による画面機能 194

11.1	データ部 (通信節による画面機能)	195
11.1.1	通信記述項 (CD) (通信節による画面機能)	195
11.2	手続き部 (通信節による画面機能)	198
11.2.1	SEND 文 (通信節による画面機能)	198
11.2.2	RECEIVE 文 (通信節による画面機能)	199
11.2.3	TRANSCIVE 文 (通信節による画面機能)	199
11.2.4	DISABLE 文 (通信節による画面機能)	200

12 画面節 (SCREEN SECTION) による画面機能 201

12.1	環境部 (SCREEN SECTION)	202
12.1.1	特殊名段落 (SPECIAL-NAMES) (SCREEN SECTION)	202
12.1.2	CONSOLE IS CRT 句 (SCREEN SECTION)	202
12.1.3	CURSOR 句 (SCREEN SECTION)	202
12.1.4	CRT STATUS 句 (SCREEN SECTION)	203
12.2	データ部 (SCREEN SECTION)	205
12.2.1	画面節 (SCREEN SECTION)	205
12.2.2	画面記述項 (SCREEN SECTION)	205
12.2.3	AUTO 句 (SCREEN SECTION)	210
12.2.4	BACKGROUND-COLOR 句 (SCREEN SECTION)	211
12.2.5	BELL 句 (SCREEN SECTION)	212
12.2.6	BLANK 句 (SCREEN SECTION)	212
12.2.7	BLANK WHEN ZERO 句 (SCREEN SECTION)	213
12.2.8	BLINK 句 (SCREEN SECTION)	213
12.2.9	COLUMN 句 (SCREEN SECTION)	214
12.2.10	ERASE 句 (SCREEN SECTION)	215
12.2.11	FOREGROUND-COLOR 句 (SCREEN SECTION)	215
12.2.12	FULL 句 (SCREEN SECTION)	216
12.2.13	GRID 句 (SCREEN SECTION)	217
12.2.14	HIGHLIGHT 句 (SCREEN SECTION)	217
12.2.15	JUSTIFIED 句 (SCREEN SECTION)	218
12.2.16	LEFTLINE 句 (SCREEN SECTION)	218
12.2.17	LINE 句 (SCREEN SECTION)	218
12.2.18	LOWLIGHT 句 (SCREEN SECTION)	219

12.2.19	OVERLINE 句 (SCREEN SECTION)	219
12.2.20	PICTURE 句 (SCREEN SECTION)	220
12.2.21	PROMPT 句 (SCREEN SECTION)	221
12.2.22	REQUIRED 句 (SCREEN SECTION)	222
12.2.23	REVERSE-VIDEO 句 (SCREEN SECTION)	223
12.2.24	SECURE 句 (SCREEN SECTION)	223
12.2.25	SIGN 句 (SCREEN SECTION)	224
12.2.26	SIZE 句 (SCREEN SECTION)	224
12.2.27	UNDERLINE 句 (SCREEN SECTION)	225
12.2.28	USAGE 句 (SCREEN SECTION)	225
12.2.29	VALUE 句 (SCREEN SECTION)	225
12.2.30	ZERO-FILL 句 (SCREEN SECTION)	226
12.3	手続き部 (SCREEN SECTION)	227
12.3.1	ACCEPT 文 (SCREEN SECTION)	227
12.3.2	DISPLAY 文 (SCREEN SECTION)	230
13	画面節 (WINDOW SECTION) による画面機能	233
13.1	概説 (WINDOW SECTION)	235
13.2	言語の基本概念 (WINDOW SECTION)	237
13.2.1	画面入出力機能特殊レジスタ	237
13.2.2	データ項目のサイズ (WINDOW SECTION)	239
13.2.3	画面フィールドの種類とそのサイズ (WINDOW SECTION)	239
13.3	環境部 (WINDOW SECTION)	243
13.3.1	特殊名段落 (SPECIAL-NAMES) (WINDOW SECTION)	243
13.3.2	入出力管理段落 (WINDOW SECTION)	243
13.4	データ部 (WINDOW SECTION)	245
13.4.1	データ記述項 (WINDOW SECTION)	245
13.4.2	BLANK WHEN ZERO 句 (WINDOW SECTION)	249
13.4.3	BLINK 句 (WINDOW SECTION)	249
13.4.4	COLOR 句 (WINDOW SECTION)	250
13.4.5	COLUMN NUMBER 句 (WINDOW SECTION)	251
13.4.6	記述項名句 (WINDOW SECTION)	255
13.4.7	ENTER-CHECK 句 (WINDOW SECTION)	256
13.4.8	FIELD TYPE 句 (WINDOW SECTION)	256
13.4.9	GLOBAL 句 (WINDOW SECTION)	257
13.4.10	HELP WINDOW 句 (WINDOW SECTION)	258
13.4.11	HIGH-INTENSITY 句 (WINDOW SECTION)	260
13.4.12	JUSTIFIED 句 (WINDOW SECTION)	260
13.4.13	LETTER 句 (WINDOW SECTION)	261

13.4.14	LINE NUMBER 句 (WINDOW SECTION)	263
13.4.15	PICTURE 句 (WINDOW SECTION)	267
13.4.16	REPEAT COLUMN 句 (WINDOW SECTION)	269
13.4.17	REPEAT LINE 句 (WINDOW SECTION)	272
13.4.18	RESET 句 (WINDOW SECTION)	275
13.4.19	REVERSE 句 (WINDOW SECTION)	276
13.4.20	RULE LINE 句 (WINDOW SECTION)	276
13.4.21	SECRET 句 (WINDOW SECTION)	279
13.4.22	SIGN 句 (WINDOW SECTION)	279
13.4.23	SOURCE 句 (WINDOW SECTION)	280
13.4.24	USAGE 句 (WINDOW SECTION)	280
13.4.25	USING 句 (WINDOW SECTION)	281
13.4.26	VALUE 句 (WINDOW SECTION)	281
13.5	手続き部 (WINDOW SECTION)	283
13.5.1	フィールド状態条件 (WINDOW SECTION)	283
13.5.2	ACCEPT 文 (WINDOW SECTION)	283
13.5.3	CANCEL 文 (WINDOW SECTION)	285
13.5.4	DISPLAY 文 (WINDOW SECTION)	286
13.5.5	ERASE 文 (WINDOW SECTION)	288
13.5.6	EXIT USE 文 (WINDOW SECTION)	289
13.5.7	REPLY 文 (WINDOW SECTION)	290
13.5.8	SET 文 (WINDOW SECTION)	291
13.5.9	USE 文 (WINDOW SECTION)	294
13.5.10	WAIT 文 (WINDOW SECTION)	297

14 書式印刷機能 298

14.1	環境部 (書式印刷機能)	299
14.1.1	入出力管理記述項 (書式印刷機能)	299
14.2	データ部 (書式印刷機能)	300
14.2.1	データ記述項 (書式印刷機能)	300

15 アドレス操作機能 305

15.1	アドレス操作機能とは	306
15.2	データの表現形式 (アドレス操作機能)	307
15.2.1	アドレス名	307
15.2.2	アドレスデータ項目	307
15.3	データ部 (アドレス操作機能)	308
15.3.1	アドレス名によって参照されるデータ項目の定義	308
15.3.2	アドレスデータ項目の定義	309

- 15.4 手続き部 (アドレス操作機能) 312
- 15.4.1 アドレス名, アドレスデータ項目および ADDR 関数の参照 (アドレス操作機能) 312
- 15.4.2 COMPUTE 文 (アドレス操作機能) 313
- 15.4.3 アドレス型関数 (アドレス操作機能) 316

16 C 言語インタフェース機能 318

- 16.1 全般規則 (C 言語インタフェース機能) 319
- 16.2 データ部 (C 言語インタフェース機能) 320
- 16.2.1 EXTERNAL 句 (C 言語インタフェース機能) 320
- 16.2.2 COMPUTATIONAL-5 (C 言語インタフェース機能) 320
- 16.3 手続き部 (C 言語インタフェース機能) 322
- 16.3.1 手続き部見出し (C 言語インタフェース機能) 322
- 16.3.2 CALL 文 (BY VALUE 指定) (C 言語インタフェース機能) 323

17 CSV ファイル入出力機能 325

- 17.1 概説 (CSV ファイル入出力機能) 326
- 17.1.1 言語の概念 (CSV ファイル入出力機能) 326
- 17.2 READ 文・WRITE 文による入出力 (CSV ファイル入出力機能) 327
- 17.2.1 環境部 (CSV ファイル入出力機能) 327
- 17.2.2 データ部 (CSV ファイル入出力機能) 327
- 17.2.3 手続き部 (CSV ファイル入出力機能) 328
- 17.2.4 入出力状態の値 (CSV ファイル入出力機能) 332
- 17.3 ACCEPT 文・DISPLAY 文による入出力 (CSV ファイル入出力機能) 333
- 17.3.1 環境部 (SPECIAL-NAMES) (CSV ファイル入出力機能) 333
- 17.3.2 手続き部 (CSV ファイル入出力機能) 333

18 OLE2 オートメーションインタフェース機能 336

- 18.1 概説 (OLE2 オートメーションインタフェース機能) 337
- 18.1.1 言語の概念 (OLE2 オートメーションインタフェース機能) 337
- 18.1.2 VARIANT 値と COBOL データの変換規則 (OLE2 オートメーションインタフェース機能) 338
- 18.2 データ部 (OLE2 オートメーションインタフェース機能) 341
- 18.2.1 OLE オブジェクト参照データ項目 (OLE2 オートメーションインタフェース機能) 341
- 18.2.2 バリエーションデータ項目 (OLE2 オートメーションインタフェース機能) 342
- 18.3 手続き部 (OLE2 オートメーションインタフェース機能) 344
- 18.3.1 INVOKE 文 (OLE2 オートメーションインタフェース機能) 344
- 18.3.2 SET 文 (OLE2 オートメーションインタフェース機能) 348
- 18.3.3 OLE オブジェクト参照データ比較の条件式 (OLE2 オートメーションインタフェース機能) 351
- 18.3.4 バリエーションデータ比較の条件式 (OLE2 オートメーションインタフェース機能) 351
- 18.4 プログラム間連絡機能 (OLE2 オートメーションインタフェース機能) 352
- 18.4.1 手続き部見出し 352

18.4.2	CALL 文	352
18.5	組み込み関数機能 (OLE2 オートメーションインタフェース機能)	353
18.5.1	CONVERT-TO-VARIANT 関数 (OLE2 オートメーションインタフェース機能)	353
18.5.2	TYPE-OF-PROPERTY 関数 (OLE2 オートメーションインタフェース機能)	354
18.5.3	TYPE-OF-VARIANT 関数 (OLE2 オートメーションインタフェース機能)	355
18.5.4	VARIANT-TO-INTEGGER 関数 (OLE2 オートメーションインタフェース機能)	356
18.5.5	VARIANT-TO-NUMERIC 関数 (OLE2 オートメーションインタフェース機能)	356
18.5.6	VARIANT-TO-ALPHANUMERIC 関数 (OLE2 オートメーションインタフェース機能)	357
18.5.7	VARIANT-TO-OBJECT 関数 (OLE2 オートメーションインタフェース機能)	357
18.5.8	LENGTH-OF-VARIANT 関数 (OLE2 オートメーションインタフェース機能)	358
18.5.9	組み込み関数の表現形式 (OLE2 オートメーションインタフェース機能)	359
19	データベース操作シミュレーション機能	360
19.1	構造型データベース (XDM/SD) 操作シミュレーション機能	361
19.1.1	データ部 (構造型データベース (XDM/SD) 操作シミュレーション機能)	361
19.1.2	手続き部 (構造型データベース (XDM/SD) 操作シミュレーション機能)	363
19.2	リレーショナルデータベース (XDM/RD) 操作シミュレーション機能	373
19.2.1	全般規則 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	373
19.2.2	見出し部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	374
19.2.3	環境部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	374
19.2.4	データ部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	374
19.2.5	原始文操作機能 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	375
19.2.6	手続き部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	376
19.2.7	特殊レジスタ (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	376
19.2.8	注意事項 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	377
20	HiRDB による索引ファイル入出力機能	378
20.1	概説 (HiRDB による索引ファイル入出力機能)	379
20.2	環境部 (HiRDB による索引ファイル入出力機能)	380
20.2.1	ファイル管理段落 (HiRDB による索引ファイル入出力機能)	380
20.3	データ部 (HiRDB による索引ファイル入出力機能)	381
20.3.1	ファイル記述項 (FD) (HiRDB による索引ファイル入出力機能)	381
20.4	手続き部 (HiRDB による索引ファイル入出力機能)	383
20.4.1	CANCEL 文 (HiRDB による索引ファイル入出力機能)	383
20.4.2	CLOSE 文 (HiRDB による索引ファイル入出力機能)	383
20.4.3	COMMIT 文 (HiRDB による索引ファイル入出力機能)	383
20.4.4	DELETE 文 (HiRDB による索引ファイル入出力機能)	384
20.4.5	EXIT PROGRAM 文, GOBACK 文 (HiRDB による索引ファイル入出力機能)	384
20.4.6	OPEN 文 (HiRDB による索引ファイル入出力機能)	384

20.4.7	READ 文 (HiRDB による索引ファイル入出力機能)	385
20.4.8	REWRITE 文 (HiRDB による索引ファイル入出力機能)	385
20.4.9	ROLLBACK 文 (HiRDB による索引ファイル入出力機能)	385
20.4.10	START 文 (HiRDB による索引ファイル入出力機能)	386
20.4.11	STOP RUN 文 (HiRDB による索引ファイル入出力機能)	386
20.4.12	UNLOCK 文 (HiRDB による索引ファイル入出力機能)	386
20.4.13	WRITE 文 (HiRDB による索引ファイル入出力機能)	386

21 数字項目のけた拡張機能 387

21.1	ソース単位	388
21.2	TURN 指令 (数字項目のけた拡張機能)	389
21.3	定数 (数字項目のけた拡張機能)	390
21.3.1	数字定数	390
21.4	データ項目 (数字項目のけた拡張機能)	391
21.4.1	数字項類 (数字項目のけた拡張機能)	391
21.4.2	数字編集項類 (数字項目のけた拡張機能)	392
21.5	一意参照 (数字項目のけた拡張機能)	393
21.5.1	添字	393
21.6	一意名のいろいろ (数字項目のけた拡張機能)	394
21.6.1	関数一意名	394
21.6.2	部分参照子	394
21.7	環境部 (数字項目のけた拡張機能)	395
21.8	データ部 (数字項目のけた拡張機能)	396
21.8.1	ファイル節 (FILE SECTION) (数字項目のけた拡張機能)	396
21.8.2	作業場所節 (WORKING-STORAGE SECTION) (数字項目のけた拡張機能)	396
21.8.3	局所場所節 (LOCAL-STORAGE SECTION) (数字項目のけた拡張機能)	397
21.8.4	連絡節 (LINKAGE SECTION) (数字項目のけた拡張機能)	397
21.8.5	報告書節 (REPORT SECTION) (数字項目のけた拡張機能)	397
21.8.6	サブスキーマ節 (SUBSCHEMA SECTION) (数字項目のけた拡張機能)	398
21.8.7	画面節 (SCREEN SECTION) (数字項目のけた拡張機能)	398
21.8.8	画面節 (WINDOW SECTION) (数字項目のけた拡張機能)	398
21.8.9	BLANK WHEN ZERO 句 (数字項目のけた拡張機能)	399
21.8.10	BLOCK CONTAINS 句 (数字項目のけた拡張機能)	399
21.8.11	DATA RECORDS 句 (数字項目のけた拡張機能)	399
21.8.12	EXTERNAL 句 (数字項目のけた拡張機能)	399
21.8.13	LINAGE 句 (数字項目のけた拡張機能)	400
21.8.14	OCCURS 句 (数字項目のけた拡張機能)	400
21.8.15	PICTURE 句 (数字項目のけた拡張機能)	400
21.8.16	USAGE 句 (数字項目のけた拡張機能)	401

21.8.17	VALUE 句 (数字項目のけた拡張機能)	401
21.8.18	埋め込み SQL 宣言節 (データベースアクセス機能) (数字項目のけた拡張機能)	402
21.9	手続き部 (数字項目のけた拡張機能)	403
21.9.1	文に共通な指定と機能	403
21.9.2	手続き部見出し	404
21.9.3	ACCEPT 文 (数字項目のけた拡張機能)	404
21.9.4	ADD 文 (数字項目のけた拡張機能)	404
21.9.5	CALL 文 (数字項目のけた拡張機能)	405
21.9.6	COMPUTE 文 (数字項目のけた拡張機能)	405
21.9.7	DISPLAY 文 (数字項目のけた拡張機能)	406
21.9.8	DIVIDE 文 (数字項目のけた拡張機能)	406
21.9.9	EVALUATE 文 (数字項目のけた拡張機能)	406
21.9.10	GO TO 文 (数字項目のけた拡張機能)	407
21.9.11	IF 文 (数字項目のけた拡張機能)	407
21.9.12	INITIALIZE 文 (数字項目のけた拡張機能)	407
21.9.13	INSPECT 文 (数字項目のけた拡張機能)	408
21.9.14	INVOKE 文 (オブジェクト指向) (数字項目のけた拡張機能)	408
21.9.15	MOVE 文 (数字項目のけた拡張機能)	408
21.9.16	MULTIPLY 文 (数字項目のけた拡張機能)	409
21.9.17	PERFORM 文 (数字項目のけた拡張機能)	409
21.9.18	READ 文 (数字項目のけた拡張機能)	410
21.9.19	RELEASE 文 (数字項目のけた拡張機能)	410
21.9.20	RETURN 文 (数字項目のけた拡張機能)	410
21.9.21	REWRITE 文 (数字項目のけた拡張機能)	411
21.9.22	SEARCH 文 (数字項目のけた拡張機能)	411
21.9.23	SET 文 (数字項目のけた拡張機能)	411
21.9.24	START 文 (数字項目のけた拡張機能)	412
21.9.25	STOP 文 (数字項目のけた拡張機能)	412
21.9.26	STRING 文 (数字項目のけた拡張機能)	412
21.9.27	SUBTRACT 文 (数字項目のけた拡張機能)	412
21.9.28	UNSTRING 文 (数字項目のけた拡張機能)	413
21.9.29	WRITE 文 (数字項目のけた拡張機能)	413
21.10	組み込み関数 (数字項目のけた拡張機能)	414
21.10.1	引数 (数字項目のけた拡張機能)	414
21.10.2	関数値 (数字項目のけた拡張機能)	414
21.11	拡張仕様 (数字項目のけた拡張機能)	417
21.11.1	ACCEPT 文	417
21.11.2	CALL 文 (CALL 文の引数指定の拡張)	418
21.11.3	COMPUTE 文 (アドレス操作機能)	418

21.11.4	DISABLE 文 (データコミュニケーション機能)	418
21.11.5	DISPLAY 文	419
21.11.6	ENABLE 文 (データコミュニケーション機能)	420
21.11.7	ENTRY 文	420
21.11.8	ERASE 文 (画面節(WINDOW SECTION))	421
21.11.9	EXAMINE 文	421
21.11.10	FETCH 文 (構造型データベース (XDM/SD) 操作シミュレーション機能)	421
21.11.11	FIND 文 (構造型データベース (XDM/SD) 操作シミュレーション機能)	421
21.11.12	INVOKE 文 (OLE2 オートメーションインタフェース機能)	422
21.11.13	MOVE 文 (日付と時刻用)	422
21.11.14	RECEIVE 文 (データコミュニケーション機能)	422
21.11.15	REPLY 文 (画面機能(WINDOW SECTION))	423
21.11.16	SEND 文 (データコミュニケーション機能)	423
21.11.17	SET 文	423
21.11.18	TRANSFORM 文	424
21.11.19	探索条件文 (構造型データベース (XDM/SD) 操作シミュレーション機能)	424
21.11.20	WRITE 文 (印刷制御文字の直接指定, 区域外書き出し)	425
21.11.21	埋め込み SQL 文 (データベースアクセス機能)	425
21.11.22	条件式 (ポインタ項目の参照)	425

22 動的長基本項目機能 426

22.1	動的長基本項目とは	427
22.2	動的長基本項目が使用できる個所	428
22.2.1	定義できるソース単位	428
22.2.2	定義できる節	428
22.2.3	対象となる文	429
22.2.4	対象となる組み込み関数	429
22.3	全般規則	431
22.3.1	表意定数	431
22.4	環境部 (動的長基本項目機能)	432
22.4.1	特殊名段落 (動的長基本項目機能)	432
22.4.2	DYNAMIC LENGTH STRUCTURE 句	432
22.5	データ部 (動的長基本項目機能)	434
22.5.1	DYNAMIC LENGTH 句	434
22.6	手続き部 (動的長基本項目機能)	437
22.6.1	引数の適合	437
22.6.2	CALL 文 (動的長基本項目機能)	437
22.6.3	DISPLAY 文 (動的長基本項目機能)	438
22.6.4	EVALUATE 文 (動的長基本項目機能)	438

- 22.6.5 IF 文 (動的長基本項目機能) 439
- 22.6.6 INITIALIZE 文 (動的長基本項目機能) 439
- 22.6.7 MOVE 文 (動的長基本項目機能) 439
- 22.7 組み込み関数 (動的長基本項目機能) 441
- 22.7.1 LENGTH 関数 (動的長基本項目機能) 441
- 22.8 コンパイラ環境変数またはコンパイラオプションとの関連 442
- 22.8.1 コンパイラ環境変数 CBLVALUE 442
- 22.8.2 -JPN オプション 442
- 22.8.3 -TruncCheck オプション 442
- 22.9 使用例 443
- 22.9.1 COBOL と C 言語の連携 443
- 22.9.2 COBOL プログラム間の連携 444
- 22.9.3 COBOL のプログラム内で使用する例 444
- 22.10 動的長基本項目を使用したときの注意事項 446

23 Unicode 機能 447

- 23.1 用語の定義 448
- 23.2 組み込み関数 450
- 23.2.1 COUNT-CHAR 関数 450
- 23.2.2 DISPLAY-OF 関数 450
- 23.2.3 LENGTH-OF-SUBSTRING 関数 450
- 23.2.4 NATIONAL-OF 関数 450
- 23.2.5 SUBSTRING 関数 450

24 コンパイラオプションによる仕様の相違 451

- 24.1 言語仕様が異なるコンパイラオプション 452
- 24.2 旧規格仕様を使う場合 454
- 24.2.1 第 1 次規格以前の仕様 (-StdVersion,1) 454
- 24.2.2 第 2 次規格以前の仕様 (-StdVersion,1, -StdVersion,2) 458
- 24.3 VOS3 COBOL85 からの移行用オプション 462
- 24.3.1 2 進項目のけた落とし (-DigitsTrunc) 462
- 24.3.2 分離符の選択 (-DoubleQuote) 462
- 24.3.3 日本語文字の扱い (-V3Rec,Fixed, -V3Rec,Variable) 463
- 24.3.4 行送り制御文字領域削除 (-IgnoreLCC) 466
- 24.3.5 日本語項目に対して埋められる空白の扱い (-JPN,V3JPNSpace) 467
- 24.3.6 日本語項目の扱い (-JPN,Alnum) 468
- 24.3.7 その他の VOS3 COBOL85 互換 (-CompatV3) 471
- 24.4 HITAC8000 シリーズからの移行用オプション (-H8Switch) 473
- 24.4.1 数字項目の転記/比較 (-H8Switch) 473

- 24.5 デバッグ行の有効, 無効 (-DebugLine) 474
- 24.6 主プログラムの指定コマンドインタフェース (-Main,System) 475
- 24.7 転記の長さのチェック (-TruncCheck) 476
- 24.7.1 チェック対象となる項目 476
- 24.7.2 チェック対象とならない項目 476
- 24.7.3 チェック対象となる転記の個所 477
- 24.7.4 受け取り側作用対象が英数字の場合 478
- 24.7.5 受け取り側作用対象が数字の場合 478
- 24.7.6 送り出し側作用対象が 2 進項目で, 受け取り側作用対象が外部 10 進項目 / 内部 10 進項目の場合の規則 (-TruncCheck,Binary オプション指定時のチェック項目) 478
- 24.7.7 -TruncCheck と-TruncCheck,Binary オプションの相違点 479

25 その他の拡張仕様 480

- 25.1 全般規則 481
- 25.1.1 かたかな 481
- 25.1.2 16 進英数字定数 482
- 25.1.3 16 進数字定数 482
- 25.1.4 表意定数 (POINTER 項目) 483
- 25.1.5 特殊レジスタ 483
- 25.1.6 プログラム名, クラス名, インタフェース名, メソッド名および利用者定義関数名の日本語指定 484
- 25.2 環境部 486
- 25.2.1 プログラム間連絡機能の環境部 486
- 25.2.2 ファイル管理記述項 487
- 25.2.3 入出力管理記述項 490
- 25.3 データ部 493
- 25.3.1 ファイル記述項 493
- 25.3.2 データ記述項 494
- 25.4 手続き部 498
- 25.4.1 算術文の作用対象のけた数 498
- 25.4.2 字類条件の内部 10 進項目指定 498
- 25.4.3 ポインタ項目の参照 498
- 25.4.4 ACCEPT 文の拡張 500
- 25.4.5 CALL 文の引数指定の拡張 501
- 25.4.6 CALL 文 (下線付き定数の呼び出し) 502
- 25.4.7 DISPLAY 文 (機能名を直接指定) 502
- 25.4.8 ENTRY 文 503
- 25.4.9 EXAMINE 文と TALLY 特殊レジスタ 505
- 25.4.10 GO TO MORE-LABELS 文 507
- 25.4.11 MOVE 文 (日付と時刻用) 508
- 25.4.12 CORRESPONDING 指定のある MOVE 文 (受け取り側作用対象を複数指定) 509

25.4.13	RELEASE 文 (定数 1 指定)	509
25.4.14	REWRITE 文 (定数 1 指定)	510
25.4.15	PERFORM 文の実行範囲の出口の共有	510
25.4.16	TRANSFORM 文	511
25.4.17	USE 文 (GIVING 指定)	512
25.4.18	USE LABEL 文	512
25.4.19	WRITE 文 (印刷制御文字の直接指定, 区域外書き出し)	513
25.4.20	WRITE 文 (定数 1 指定)	515
25.5	組み込み関数	516
25.5.1	用語の定義 (日本語 EUC)	516
25.5.2	CONVERT-CODE 関数	517
25.5.3	COUNT-CHAR 関数	518
25.5.4	DISPLAY-OF 関数	522
25.5.5	LENGTH-OF-SUBSTRING 関数	523
25.5.6	NATIONAL-OF 関数	527
25.5.7	SUBSTRING 関数	529
25.6	2 進項目の拡張	534
25.6.1	1 バイト 2 進機能	534
25.6.2	COMP-X 項目	535

付録 541

付録 A	Windows COBOL2002 と UNIX COBOL2002 との言語仕様の相違	542
付録 B	LANGOPT=(D)オプションと-JPN,V3JPNSpace オプションを指定した場合の仕様の相違	544
付録 B.1	VOS3 COBOL85 と COBOL2002 での仕様の相違	544
付録 B.2	比較条件の組み合わせ	557
付録 B.3	字類条件の組み合わせ	558
付録 B.4	転記の組み合わせ	560
付録 C	STANDARD オプションと-StdVersion オプションを指定した場合の仕様の相違	563
付録 D	このマニュアルの参考情報	567
付録 D.1	関連マニュアル	567
付録 D.2	このマニュアルでの表記	567

索引 571

1

日本語機能

この章では、日本語機能について説明する。

1.1 日本語機能とは

日本語機能は、利用者が日本語を用いてプログラミングしたり、日本語データを処理したりするためにサポートした機能である。すなわち、この機能を用いると、データ名やファイル名、手続き名などを日本語で記述し、日本語データを定義、操作できる。

日本語で記述したプログラム例を次に示す。

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SAMPLE00.  
ENVIRONMENT DIVISION.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.  
SELECT 得意先マスタ ASSIGN TO SYS010  
        ORGANIZATION IS SEQUENTIAL.  
DATA DIVISION.  
FILE SECTION.  
FD 得意先マスタ RECORDING MODE IS F.  
  01 得意先レコード.  
    02 得意先コード PIC X(5).  
    02 得意先名      PIC N(20).  
    :  
WORKING-STORAGE SECTION.  
77 得意先数  PIC 9(4) COMP VALUE 0.  
  01 得意先一覧.  
    02 得意先名 PIC N(20) OCCURS 100.  
    :  
PROCEDURE DIVISION.  
開始処理.  
  OPEN INPUT 得意先マスタ.  
入力処理.  
  READ 得意先マスタ  
    AT END GO TO 終了処理.  
  PERFORM 得意先レコード処理.  
  GO TO 入力処理.  
得意先レコード処理.  
  ADD 1 TO 得意先数.  
  MOVE 得意先名 OF 得意先レコード  
    TO 得意先名 OF 得意先一覧(得意先数).  
  :  
終了処理.  
  CLOSE 得意先マスタ.  
  STOP RUN.
```

この章では、日本語を取り扱う上で追加される言語仕様を述べる。

1.2 全般規則（日本語機能）

日本語文字は、シフト JIS 標準漢字コードの 2 バイトコードで表される。日本語文字 1 文字は、原始文上で、または記憶領域上で、標準データ形式の 2 文字分の位置を占める。

-UniObjGen オプションを指定した場合、日本語文字は、UTF-16 の 2 バイトコードで表される。日本語文字 1 文字は、記憶領域上で、標準データ形式の 2 文字分の位置を占める。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

このシステムでは、日本語文字を用いて比較する場合、比較は日本語文字集合のコード値の照合順序に従う。

1.2.1 日本語文字集合（日本語機能）

標準仕様編の文字集合で記述した文字に加え、日本語文字および日本語空白を用いることができる。

このシステムでは、シフト JIS 標準漢字コードで決められた漢字を日本語文字として扱う。

-UniObjGen オプションを指定した場合、UCS-4 の範囲（UCS-2 の範囲を含む）を日本語文字として扱う。日本語空白は、IDEOGRAPHIC SPACE (X'3000') とする。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

注意

バイトオーダーがリトルエンディアンの場合、日本語空白は X'0030'を表す。Unicode 機能のバイトオーダーの指定方法については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.1 文字集合」

1.2.2 文字列（日本語機能）

文字列に日本語文字を使用できる。

(1) 利用者定義語

日本語文字を用いて、利用者が定義できる語には、次のものがある。

- アドレス名 (address-name)
- 符号系名 (alphabet-name)
- 通信記述名 (cd-name)

- クラス名 (class-name)
- 字類名 (class-name)
- 翻訳変数名 (compilation-variable-name)
- 構成要素ビュー名 (component-view-name)
- 合成キー名 (composite-key-name)
- 条件名 (condition-name)
- 定数名 (constant-name)
- データ名 (data-name)
- ファイル名 (file-name)
- 指標名 (index-name)
- インデクス名 (index-name)
- インタフェース名 (interface-name)
- メソッド名 (method-name)
- 呼び名 (mnemonic-name)
- OLE メソッド (OLE method) ※
- OLE オブジェクト (OLE object) ※
- OLE プロパティ (OLE property) ※
- 段落名 (paragraph-name)
- パス名 (path-name)
- プログラム名 (program-name)
- プロパティ名 (property-name)
- レコード名 (record-name)
- レコードビュー名 (record-view-name)
- 報告書名 (report-name)
- 手順名 (routine-name)
- 画面名 (screen-name)
- 節名 (section-name)
- 親子集合ビュー名 (set-view-name)
- 記号文字 (symbolic-character)
- 型名 (type-name)
- 利用者定義関数名 (user-function-name)

注※

UNIX COBOL2002 では、使用できない。

日本語文字集合の英文字、数字、特殊文字は英数字文字集合の対応する英文字、数字、特殊文字と等価とみなされる。

-UniObjGen オプションを指定した場合、プログラム名およびメソッド名に UTF-8 で多バイトとなる文字が含まれてはならない。また、-UniObjGen オプションの指定が前提である Linux COBOL2002 では、プログラム名、クラス名、インタフェース名、利用者定義関数名、メソッド名、およびプロパティ名に UTF-8 で多バイトとなる文字が含まれてはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

(2) 定数

(a) 日本語文字定数

日本語文字定数は次に示す形式で表された 1 文字以上 160 文字以下の文字列である。

日本語文字定数の長さは、実行用プログラム中のその表現に適用される。

形式

```
N' {文字1}...'  
NC' {文字1}...'  
ND' {文字1}...'
```

構文規則

1. 文字 1 は日本語文字集合の任意の文字とする。
2. 文字 1 が引用符を表すならば、引用符の 1 文字を表すのに連続する 2 字の日本語文字集合の引用符を用いなければならない。

一般規則

1. 実行用プログラム中の日本語文字定数の値は、文字 1 によって表される値とする。
2. 日本語文字定数の項類は、日本語となる。
3. 日本語文字定数を囲む分離符の引用符は、日本語文字定数の一部ではない。

(b) 16 進日本語文字定数

16 進日本語文字定数は、次に示す形式で表された長さが 1 文字以上 160 文字以下で 4 の倍数の文字列である。

形式

```
NX' {文字1}... '
```

構文規則

1. 文字 1 は 16 進数字の '0' ~ '9', 'A' ~ 'F' の 16 種類である。
2. 長さは 160 文字までで、4 の倍数でなければならない。
3. NX を VALUE 句に指定する場合、文字 1 は PICTURE 句で指定したサイズを超えてはならない。

一般規則

1. 実行用プログラム中の 16 進日本語文字定数の値は、文字 1 によって表される値とする。
2. すべての 16 進日本語文字定数の項類は、日本語となる。
3. 16 進日本語文字定数を囲む分離符の引用符は、16 進日本語文字定数の一部ではない。
4. -UniObjGen オプションを指定した場合、UTF-16 のコード値はバイトオーダがリトルエンディアンであっても UTF-16BE で指定する。なお、UTF-16BE で指定した 16 進日本語文字定数のコード値は、-UniEndian オプションの指定に従い、UTF-16LE、または UTF-16BE に変換される。-UniObjGen オプション、-UniEndian オプション、および Unicode 機能のバイトオーダの指定方法については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

(c) 表意定数

次に示す表意定数を用いることができる。

表 1-1 日本語機能の表意定数

予約語	値
[ALL] ZERO [ALL] ZEROS [ALL] ZEROES	数値ゼロ (0)、または日本語文字集合の何文字かの文字ゼロ (0) を表す。 -UniObjGen オプションを指定した場合、UTF-16 の FULLWIDTH DIGIT ZERO (X'FF10') を表す。
[ALL] SPACE [ALL] SPACES	日本語文字集合の何文字かの空白を表す。 -UniObjGen オプションを指定した場合、UTF-16 の IDEOGRAPHIC SPACE (X'3000') を表す。
[ALL] HIGH-VALUE [ALL] HIGH-VALUES	特殊名段落の中で使用される場合を除いて、その計算機の文字の照合順序で最高の位置を占める日本語文字集合の文字の何文字かを表す。
[ALL] LOW-VALUE [ALL] LOW-VALUES	特殊名段落の中で使用される場合を除いて、その計算機の文字の照合順序で最低の位置を占める日本語文字集合の文字の何文字かを表す。

注意

- バイトオーダがリトルエンディアンの場合、ZERO(ZEROS, ZEROES)は X'10FF'を、SPACE (SPACES)は IDEOGRAPHIC SPACE (X'0030') を表す。Unicode 機能のバイトオーダの指定方法については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。
- 組み込み関数の引数として指定された表意定数の扱いについては、マニュアル「COBOL2002 言語 標準仕様編」[11.2 引数]を参照のこと。

1.2.3 分離符（日本語機能）

標準仕様編の分離符で記述した文字に加え、直後に句読文字の引用符が続く文字「N」、「NC」、「ND」および「NX」も分離符とする。これらの分離符の直前は、空白、または左括弧でなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.2.2 分離符」

1.2.4 データ項目の字類と項類（日本語機能）

日本語文字データの字類は日本語と定義する。また、それを扱う日本語集団項目の項類は日本語、基本項目の項類は日本語、および日本語編集と定義する。

項類が日本語、または日本語編集である基本項目を含む集団項目で日本語集団項目以外の字類は英数字とする。

日本語を含めた字類と項類との関係を次に示す。

表 1-2 データ項目の項類と字類

項目のレベル			字類	項類
基本項目			英字	英字
			数字	数字
			英数字	数字編集 英数字編集 英数字
			日本語	日本語 日本語編集
集団項目	固定長集団項目	日本語集団項目	日本語	日本語
		英数字集団項目		
	可変長集団項目	英数字集団項目		

1.2.5 標準けた寄せ規則（日本語機能）

日本語文字は左端を基準にして、1文字単位にけた寄せをする。そして必要に応じて右端に空白を補ったり右端を切り捨てたりする。

集団項目のけた寄せは、先頭項目のけた寄せと必ず一致する。

例

```

01 A-RECORD.
   05 A-NDATA1      PIC N(2).
   05 FILLER        PIC X(1).
   05 A-GROUP GROUP-USAGE NATIONAL.
      10 A-NDATA2   PIC N(1).
      10 FILLER     PIC N(1).
    
```

図 1-1 記憶域内の配置

X	X+1	X+2	X+3	X+4	X+5	X+6	X+7	X+8	番地
A-RECORD									
A-NDATA1				FILLER	A-GROUP				
					A-NDATA2		FILLER		

A-RECORDの割り付け番地は8の倍数になります。この番地をXとします。
 このとき、先頭項目A-NDATA1の番地もXなので、英数字集団項目A-RECORDと一致します。
 また、日本語集団項目A-GROUPの番地はX+5です。
 先頭項目A-NDATA2の番地もX+5なので、日本語集団項目A-GROUPと一致します。

1.3 コーディング規則（日本語機能）

1.3.1 行と文字位置（日本語機能）

固定形式正書法の場合，原始プログラムは，英数字 1 文字を 1 個分の文字位置，日本語文字を 2 個分の文字位置として 80 個の文字位置で構成される行中に記述される。

原始プログラムの 1 行は次のように表現される。

図 1-2 COBOL 原始プログラムの領域

一連番号領域	標識領域	プログラムテキスト領域	見出し領域
1	6	7 8	72 73 80

上図に示す 8 文字目から 72 文字目までが，原始プログラムを記述する領域である。

1.3.2 行の継続（日本語機能）

日本語文字定数は，複数行にわたって定義できる。

1.3.3 注記行（日本語機能）

注記行には，日本語文字を用いて注釈を記述できる。

1.4 見出し部（日本語機能）

見出し部の注記項に日本語文字を使用してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「7. 見出し部（IDENTIFICATION DIVISION）」

1.5 環境部（日本語機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部（ENVIRONMENT DIVISION）」

1.5.1 特殊名段落（日本語機能）

日本語機能内での特殊名段落の記述についての一般規則を次に示す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落（SPECIAL-NAMES）」

(1) 符号系名句

NATIVE（計算機固有文字集合）を書くと、符号系名は JIS8 単位コード符号系を表す。このシステムに固有な日本語文字集合のシフト JIS 標準漢字コード符号系を表すものではない。

STANDARD-1（英数字文字集合）を書くと、符号系名は JIS X 0201-1976 符号系を表す。標準日本語文字集合である JIS X 0208-1983 符号系を表すものではない。

(2) CURRENCY SIGN 句

この句で指定する通貨編集用文字には、「N」は使用できない。

1.5.2 入出力管理段落の APPLY FORMS-OVERLAY 句（日本語機能）

形式

```
APPLY FORMS-OVERLAY  
TO データ名 ON {ファイル名1}...
```

機能

詳細は「14.1.1 入出力管理記述項（書式印刷機能）」を参照のこと。

1.6 データ部 (日本語機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

1.6.1 CHARACTER TYPE 句 (日本語機能)

形式

$$[\text{CHARACTER TYPE IS}] \left\{ \begin{array}{l} \text{KEIS} \\ \text{POINT-}i \\ \text{GAP-}m \\ \text{FORMAT-}n \\ \text{INTERVAL-}j \\ \text{WIDE} \end{array} \right\}$$

機能

GDI モード印刷機能, または XMAP3 を使用したプリンタ出力機能では, 印刷制御付きの行データを出力できる。CHARACTER TYPE 句は ESC/P モード印刷には有効とならない。このため, レコード中に機能コードを設定して印刷制御をする必要がある。

XMAP3 を使用したプリンタ出力機能を使用する場合, -XMAP,LinePrint オプションを指定すること。-XMAP,LinePrint オプションについては, マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。

構文規則

1. CHARACTER TYPE 句の指定がある項目を含むレコード記述項中に, GROUP-USAGE 句を指定してはならない。

1.6.2 GROUP-USAGE 句 (日本語機能)

形式

`GROUP-USAGE IS NATIONAL`

機能

GROUP-USAGE 句に NATIONAL を指定したものは, 記述項の左辺で定義される集団項目の扱いが特に指定のない限り, 用途が NATIONAL で項類が日本語の基本項目になることを指定する。

構文規則

1. GROUP-USAGE 句は, 記述項の左辺が集団項目でなければならない。
2. NATIONAL 指定がある場合, 記述項の左辺には USAGE NATIONAL が仮定される*。ただし, 記述項の左辺に USAGE 句を指定してはならない。記述項の左辺に従属している基本項目はすべて

日本語項目または日本語編集項目のどちらかでなくてはならない。従属している集団項目はすべて明示的にまたは暗黙的に GROUP-USAGE NATIONAL と記述されている必要がある。

注※

-JPN,Alnum オプションまたは-JPN,V3JPN オプションが有効な場合、記述項の左辺に USAGE NATIONAL は仮定しない。

- GROUP-USAGE 句が指定された集団項目およびその従属項目に、TYPE 句および SAME AS 句を指定してはならない。

表 1-3 GROUP-USAGE 句と TYPE 句または SAME AS 句との組み合わせ

TARGET の条件	TARGET に TYPE 句を指定	TARGET に TYPEDEF 句を指定	TARGET に SAME AS 句を指定	SAME AS で参照するデータ名 1 として TARGET を指定
GROUP-USAGE 句の指定がある集団項目 (例) 01 TARGET ~ GROUP-USAGE NATIONAL. 03 DATA1 ~.	組み合わせて使用できない。			
GROUP-USAGE 句の指定がある集団項目の従属項目 (例) 01 GRP-N GROUP-USAGE NATIONAL. 03 TARGET ~.	組み合わせて使用できない。	01 レベル以外に TYPEDEF 句を指定できない。	組み合わせて使用できない。	TARGET が集団項目の場合、組み合わせて使用できない。 TARGET が基本項目の場合、組み合わせて使用できる。
GROUP-USAGE 句の指定がある集団項目を従属項目に持つ集団項目 (例) 01 TARGET. 03 GROUP-USAGE NATIONAL. 05 DATA1 ~.	TYPE 句を指定した項目の下位にデータ項目は定義できない。	組み合わせて使用できない。	SAME AS 句を指定した項目の下位にデータ項目は定義できない。	組み合わせて使用できない。

- 可変長集団項目またはその従属項目に GROUP-USAGE 句を指定してはならない。
- CHARACTER TYPE 句の指定がある項目を含むレコード記述項中に、GROUP-USAGE 句を指定してはならない。
- 画面節 (SCREEN SECTION および WINDOW SECTION)、報告書節、およびサブスキーマ節の記述項に、GROUP-USAGE 句を指定してはならない。
- 画面節 (SCREEN SECTION) による画面機能の ACCEPT 文または DISPLAY 文の一意名に、GROUP-USAGE 句を含む集団項目を指定してはならない。
- CSV ファイルのレコード記述項に GROUP-USAGE 句を含んでいてはならない。

9. CSV ファイル入出力機能の ACCEPT 文または DISPLAY 文の一意名に、GROUP-USAGE 句を含む集団項目を指定してはならない。
10. CSV ファイル入出力機能の WRITE 文の FROM に指定した一意名に、GROUP-USAGE 句を含む集団項目を指定してはならない。
11. CSV ファイル入出力機能の READ 文の INTO に指定した一意名に、GROUP-USAGE 句を含む集団項目を指定してはならない。

一般規則

1. NATIONAL 指定を書いた場合

- (a) 記述項の左辺は日本語集団となり、その字類と項類は日本語となる。
- (b) 特に指定がない限り日本語集団は、一個の基本データ項目であるかのように扱われる。その用途は NATIONAL で字類と項類は日本語であり、記述は PICTURE N(m)となる。ここで、mはその集団の長さである。
- (c) -JPN,Alnum オプションが有効な場合、記述項の左辺は英数字集団となる。-JPN,Alnum オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。
- (d) -JPN,V3JPN オプションが有効な場合、記述項の左辺は英数字集団となる。ただし、STRING 文、UNSTRING 文、INSPECT 文、LENGTH 関数で参照するときは、日本語項目であるかのように扱われる。また、部分参照の字類と項類は英数字とするが、最左端の文字位置と長さは日本語文字数を表す。-JPN,V3JPN オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

注

GROUP-USAGE NATIONAL の指定が必要とされる理由は、日本語だけを含む集団に対して、切り捨てや日本語の空白詰めが適正に行えるようにする、および INSPECT 等の操作が間違いなく行えるようにするためである。GROUP-USAGE NATIONAL の指定がないとすると、そのような集団項目の内容は、項類が英数字の扱いが半角文字列となってしまうので、データの碎裂や不正操作を招くおそれが高い。

2. 強く型付けされていない集団項目に対して、GROUP-USAGE 句の指定も仮定もなければ、その集団項目は英数字集団項目となる。

1.6.3 JUSTIFIED 句 (日本語機能)

形式

$$\left. \begin{array}{l} \underline{\text{JUSTIFIED}} \\ \underline{\text{JUST}} \end{array} \right\} \text{RIGHT}$$

機能

けた寄せ基準位置を右端に変更することを指定する。

構文規則

1. JUSTIFIED 句は基本項目レベルにだけ指定してもよい。
2. JUST は JUSTIFIED の略記である。
3. JUSTIFIED 句は日本語項目および日本語編集項目に指定してもよい。

一般規則

1. 受け取り側作用対象の記述に JUSTIFIED 句がある場合、日本語文字データの転記は右端から 1 文字単位に行われる。必要に応じて左端に日本語空白を補ったり、左端を切り捨てたりする。
2. JUSTIFIED 句の指定がない場合は、標準けた寄せ規則が適用される。詳細は、「1.2.5 標準けた寄せ規則（日本語機能）」を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.36 JUSTIFIED 句」

1.6.4 PICTURE 句（日本語機能）

形式

$\left. \begin{array}{l} \text{PICTURE} \\ \text{PIC} \end{array} \right\} \text{ IS 文字列}$

機能

日本語項目および日本語編集項目を定義する。

構文規則

1. 日本語文字データを扱う基本項目に対しての PICTURE 文字列は、「N」および「B」の組み合わせで構成する。
2. 日本語文字データを扱う基本項目に対して次の句を書くことはできない。
 - (a) BLANK WHEN ZERO 句
 - (b) SIGN 句
 - (c) USAGE 句（DISPLAY 指定および NATIONAL 指定を除く）

一般規則

1. 日本語項目

- (a) PICTURE 文字列は、「N」だけから成る。
- (b) 文字「N」は、日本語文字 1 文字（2 バイト）を示す。項目の内容は、標準データ形式で表したとき、一つ以上の日本語文字でなければならない。
- (c) 項目のサイズは、16,383 文字以下でなければならない。
- (d) 項目の内容は標準データ形式で表したとき、一つ以上の日本語文字でなければならない。

2. 日本語編集項目

(a) PICTURE 文字列は、「N」と「B」から成り、それぞれ少なくとも一つずつ含まなければならない。

(b) 文字「N」は、日本語文字 1 文字 (2 バイト) を示し、「B」は、日本語空白 1 文字 (2 バイト) を示す。

それぞれ項目の内容は、標準データ形式で表したとき、一つ以上の日本語文字でなければならない。

(c) 項目のサイズは、16,383 文字以下でなければならない。

(d) 編集は、MOVE 文でこの項目へ転記するときに行われる。文字「B」の部分には日本語文字集合の空白が挿入される。

使用例

```
03 A1 PIC N(4).
03 B1 PIC NBNBNB.
03 B2 PIC BNNNNB.
03 B3 PIC BNBNNBNB.
```

A1の内容を

株	式	会	社
---	---	---	---

 とする。

次のMOVE文を実行すると、B1、B2およびB3の内容はそれぞれ右側に示すようになる。

MOVE A1 TO B1	B1の内容	株	式	会	社
MOVE A1 TO B2	B2の内容	株	式	会	社
MOVE A1 TO B3	B3の内容	株	式	会	社

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

1.6.5 USAGE 句 (日本語機能)

形式

[USAGE IS] NATIONAL

機能

計算機の記憶領域内でのデータ項目の表現形式を指定する。

構文規則

1. NATIONAL を指定する USAGE 句を持つ基本項目、またはそのような USAGE 句を持つ集団項目に従属する基本項目は、日本語項目、または日本語編集項目を記述する PICTURE 句の文字列によって宣言されなければならない。すなわち、PICTURE 句の文字は「N」および「B」だけから成っていないなければならない。

一般規則

1. 項類が日本語または日本語編集である基本項目、またはそれを含む集団項目に USAGE 句が指定されなければ、用途は暗黙的に NATIONAL となる。

2. 明示的、または暗黙的に指定された USAGE IS NATIONAL 句は、計算機の記憶領域内で日本語項目を表すために日本語文字集合が用いられること、およびデータ項目が英数字文字集合のバイト境界にけた寄せされることを指定する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

1.6.6 VALUE 句 (日本語機能)

形式

書き方 1 データ項目の初期値を指定する場合

VALUE IS 定数

書き方 2 条件名で表す値を指定する場合

$$\left\{ \begin{array}{l} \text{VALUE IS} \\ \text{VALUES ARE} \end{array} \right\} \{ \text{定数1} \} \dots$$

機能

データ項目の初期値、または条件名で表す値を指定する。

構文規則

1. VALUE 句を用いて、日本語集団項目、日本語項目、または日本語編集項目に、初期値および条件名で表す値を指定できる。この場合、定数は日本語文字定数または表意定数の SPACE (S) でなければならない。
2. 日本語項目または日本語編集項目の VALUE 句に指定された日本語文字定数は、明示的な PICTURE 句で示される大きさを超えてはならない。日本語集団項目の VALUE 句に指定された日本語文字定数は、その集団項目の大きさを超えてはならない。
3. 集団レベルに VALUE 句を指定する場合、その集団内の従属項目の記述に JUSTIFIED 句や SYNCHRONIZED 句があってはならない。

一般規則

1. 項目の項類が日本語、または日本語編集の場合、VALUE 句の定数はすべて日本語文字定数でなければならない。
2. VALUE 句に指定された定数は、「1.2.5 標準けた寄せ規則 (日本語機能)」に従って関連データ項目に収納される。ただし、初期化は JUSTIFIED 句の影響を受けないし、編集も行われない。PICTURE 文字列の「B」に対応する文字位置を日本語文字集合の空白にする場合は、利用者の責任でそのような定義を書かなければならない

使用例

PICTURE 句の指定	VALUE の指定	設定する初期値
N(4)	N'報告書'	報告書△

PICTURE 句の指定	VALUE の指定	設定する初期値
NNBNN	N'株式会社'	株式会社△ (株式△会社とはならない)

(凡例)

△：日本語文字集合の空白

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

1.7 手続き部（日本語機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

1.7.1 単純条件（日本語機能）

比較条件で日本語集団項目、日本語項目、日本語編集項目、および日本語文字定数を作用対象として指定できる。

(1) 両方の作用対象が日本語の場合

- 比較は日本語文字集合の照合順序に従って行う。
- 作用対象の長さとは、その作用対象が含む標準データ形式の日本語の文字位置の個数で表される。両方の作用対象の長さが等しい場合と等しくない場合について、比較の方法を次に示す。
 - 両方の作用対象の長さが等しい場合
両方の作用対象の対応する日本語文字を最左端から最右端に向って一組ずつ順に比較する。それぞれの対応する日本語文字がすべて等しいとき、これらの二つの作用対象は等しいものとする。
等しくない日本語文字が初めて現れたとき、照合順序でより高い位置の日本語文字集合の文字を含む作用対象をより大きいものとする。
 - 作用対象の長さが等しくない場合
短い方の作用対象の右側に、長い方の作用対象の長さに等しくなるまで日本語文字集合の空白があるものとみなして前項と同じように比較をする。

(2) 作用対象の一方だけが日本語の場合

- 作用対象の一方は英数字集団項目でなければならない。
- 比較は日本語作用対象を英数字文字位置に換算し、この文字位置の個数のサイズを英数字作用対象として英数字比較する。

1.7.2 INITIALIZE 文 (日本語機能)

形式

```
INITIALIZE {一意名1} ...  
{REPLACING {  
  ALPHABETIC  
  ALPHANUMERIC  
  ALPHANUMERIC-EDITED  
  BOOLEAN  
  NATIONAL  
  NATIONAL-EDITED  
  NUMERIC  
  NUMERIC-EDITED  
  OBJECT-REFERENCE  
}} DATA BY {一意名2  
             定数1 } ... ]
```

機能

日本語項目または日本語編集項目に初期値を設定する。

一般規則

- 一意名 1 が日本語集団項目を指し示している場合、一意名 1 は集団項目として処理される。一意名 2 が日本語集団項目を指し示している場合、一意名 2 は基本項目として処理される。
- 初期設定は次のようにする。
 - REPLACING 指定がない場合、日本語文字集合の空白を設定する。
 - REPLACING 指定がある場合、一意名 2、または定数の内容を設定する。

使用例

REPLACING 指定がある場合の、初期設定をするコーディングを次に示す。

```
INITIALIZE REPLACING  
NATIONAL DATA BY ALL N' '.
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.24 INITIALIZE 文」

1.7.3 INSPECT 文 (日本語機能)

形式

形式は標準仕様と同じ。

機能

日本語項目中の文字や文字列の出現回数を数えたり、それらをほかの日本語文字列で置き換えたりする。

構文規則

- 一意名 1 は日本語集団項目、または用途が明示的または暗黙的に NATIONAL と記述された基本項目でなければならない。
- 一意名 1 の用途が明示的または暗黙的に NATIONAL のとき、一意名 3 から一意名 7 は用途が明示的または暗黙的に NATIONAL と記述された基本項目でなければならない。

3. 定数はすべて日本語文字定数でなければならない。日本語文字定数は一意名 1 の用途が NATIONAL のときだけ指定できる。

一般規則

1. 一意名 1, 一意名 3 から一意名 7 の用途が NATIONAL のデータ項目のとき, これら一意名のデータ項目の内容は日本語文字集合の文字列とみなされる。
2. 一意名 2 が数え上げるのは日本語文字の個数であり, バイト数ではない。
3. この文に関するその他の規則は, 標準仕様に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.25 INSPECT 文」

1.7.4 MOVE 文 (日本語機能)

形式

形式は標準仕様と同じ。

機能

日本語集団項目, 日本語項目, 日本語編集項目, または日本語文字定数を, 日本語集団項目, 日本語項目, または日本語編集項目へ転記する。

一般規則

1. 送り出し側作用対象が日本語集団項目, 日本語項目, 日本語編集項目, または日本語文字定数で, 受け取り側作用対象が, 日本語集団項目, 日本語項目, または日本語編集項目であるとき「[1.2.5 標準けた寄せ規則 \(日本語機能\)](#)」に従って転記される。
2. 次の表意定数を日本語集団項目, 日本語項目, または日本語編集項目に転記するときは, その項類は日本語とする。
 - ・SPACE (SPACES)
 - ・ZERO (ZEROS, ZEROES)
 - ・HIGH-VALUE (HIGH-VALUES)
 - ・LOW-VALUE (LOW-VALUES)
3. 日本語項目および日本語編集項目以外の基本データ項目を, 日本語集団項目, 日本語項目, または日本語編集項目に転記してはならない。
4. 日本語集団項目, 日本語項目, または日本語編集項目を, 日本語項目および日本語編集項目以外の基本項目に転記してはならない。
5. 書き方 1 の MOVE 文中では, 日本語集団項目は基本項目として扱われる。日本語項目, 日本語編集項目, 日本語文字定数から日本語集団項目への転記, および日本語集団項目から日本語項目, 日本語編集項目への転記は基本項目転記となる。

6. 書き方 2 の MOVE 文中では、日本語集団項目は集団項目として処理される。日本語集団項目に属する項目が選択されて転記される処理は、「COBOL2002 言語 標準仕様編」[10.6.5 CORRESPONDING 指定]に規定された規則に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」[10.8.28 MOVE 文]

1.7.5 SEARCH 文 (日本語機能)

形式

形式は標準仕様と同じ。

機能

指定した条件を満足する日本語集団項目、日本語項目、または日本語編集項目の表要素を探し、対応する指標の値がその表要素を指すようにする。

一般規則

1. 条件は日本語集団項目、日本語項目、日本語編集項目および日本語文字定数の比較条件に従って検査される。
2. この文に関するその他の規則は、標準仕様に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」[10.8.41 SEARCH 文]

1.7.6 STRING 文 (日本語機能)

形式

形式は標準仕様と同じ。

機能

幾つかの日本語項目の内容の一部または全部をつなぎ合わせて、一つの日本語項目に移す。

構文規則

1. 一意名 3 の用途が NATIONAL のとき、定数は日本語文字定数でなければならない。
2. 一意名 3 の用途が NATIONAL のとき、一意名 4 以外の一意名の用途は、明示的または暗黙的に NATIONAL と記述されていなければならない。

一般規則

1. 一意名 3 の用途が NATIONAL のとき、定数 1 または定数 2 に表意定数を使うと、それは用途が NATIONAL である暗黙的な長さ 1 のデータ項目とみなされる。
2. 一意名 3 の用途が NATIONAL のとき、定数 1 または一意名 1 のデータ項目の内容が、日本語から日本語への転記の規則に従って、一意名 3 のデータ項目に転記される。

- 一意名 4 が示すものは受け取り側作用対象中の相対文字位置であって、相対バイト位置ではない。
- この文に関するその他の規則は、標準仕様に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.47 STRING 文」

1.7.7 UNSTRING 文 (日本語機能)

形式

形式は標準仕様と同じ。

機能

送り出し側作用対象の連続した日本語データを分解し、幾つかの受け取り側作用対象に入れる。

構文規則

- 一意名 1 の用途が NATIONAL のとき、定数 1 および定数 2 は、日本語文字定数または表意定数 SPACE でなければならない。
- 一意名 1 の用途が NATIONAL のとき、一意名 2 から一意名 5 は、明示的または暗黙的に項類が日本語のデータ項目として記述されていないなければならない。

一般規則

- 区切り文字に表意定数を用いると、長さ 1 の日本語文字定数とみなされる。
- 一意名 1 の用途が NATIONAL のとき、一意名 1 のデータ項目の内容が、MOVE 文の転記の規則に従って、一意名 4 のデータ項目に転記される。
- 一意名 7 が示すものは受け取り側作用対象中の相対文字位置であって、相対バイト位置ではない。
- この文に関するその他の規則は、標準仕様に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.52 UNSTRING 文」

2

テキストファイル機能

この章では、テキストファイル機能の形式について説明する。

2.1 テキスト編成

テキストファイルとは、テキストエディタによって作成されたファイルである。テキストファイルの各行（テキスト行）は、改行文字（Windows COBOL2002 では X'0D0A', UNIX COBOL2002 では X'0A'）で区切られた文字の並びであるが、COBOL で扱うレコード長は、ファイル記述項（FD）のレコード定義およびテキスト行の長さによって決まる。

このシステムでは、テキストファイルを順ファイルの拡張仕様としてサポートする。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「5.1.7(1) 順編成」

2.2 環境部 (テキストファイル機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

2.2.1 ファイル管理段落 (テキストファイル機能)

形式

```
SELECT [[NOT] OPTIONAL] ファイル名
  ASSIGN TO { 外部装置名1
             データ名
             定数1 } ...
  [ORGANIZATION IS LINE SEQUENTIAL]
  [ACCESS MODE IS SEQUENTIAL]
  [ LOCK MODE IS { AUTOMATIC [WITH LOCK ON RECORD]
                  EXCLUSIVE } ] ]
  [FILE STATUS IS データ名1 [データ名2]]
  [PROCESSING MODE IS SEQUENTIAL]
```

Windows COBOL2002 では、テキストファイルでのファイル共用に関する指定は覚え書きとする。UNIX COBOL2002 では、ファイル全体の施錠は有効であるが、レコード施錠に関する指定は覚え書きとする。

形式で示された指定については、次を除いて順ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.3 ファイル管理段落 (FILE-CONTROL)」

(1) ORGANIZATION IS LINE SEQUENTIAL 句

機能

ORGANIZATION IS LINE SEQUENTIAL 句は、ファイルの論理的構成がテキストファイルであることを指定する。

一般規則

1. ORGANIZATION IS LINE SEQUENTIAL 句は、ファイルの論理的構成がテキストファイルであることを指定する。ファイル編成は、ファイルを生成するときに定まり、後で変更できない。
2. テキストファイルは、順編成と同じでレコードがファイルに書き出されるときに定まる前後の関係によって一意となる永続的な論理ファイル構造である。
3. ORGANIZATION IS LINE SEQUENTIAL 句が指定されたとき、ファイルはおのこのデータを一行ずつ含む可変長レコードの集まりとして扱われる。一行とは文字の並びであり、その終わりはレコード終端コードである。終端コードは、レコードの長さに含まれない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(9) ORGANIZATION 句」

2.3 データ部 (テキストファイル機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

2.3.1 ファイル記述項 (テキストファイル機能)

形式

```
FD ファイル名1
  [IS EXTERNAL]
  [IS GLOBAL]
  [RECORD { CONTAINS 整数3 CHARACTERS
            IS VARYING IN SIZE
            [[FROM 整数4] [TO 整数5] CHARACTERS]
            [DEPENDING ON データ名1]
            CONTAINS 整数6 TO 整数7 CHARACTERS } ]
  [DATA { RECORD IS
          RECORDS ARE } {データ名3} ...]
  [LINAGE IS { データ名4
              整数8 } LINES
    [WITH FOOTING AT { データ名5
                      整数9 } ]
    [LINES AT TOP { データ名6
                   整数10 } ]
    [LINES AT BOTTOM { データ名7
                      整数11 } ]].
```

形式で示された指定については、順ファイルと異なる規則だけを記述する。

構文規則

1. -UniObjGen オプションを指定した場合、ファイル記述項に従属するレコード記述は、用途が DISPLAY の項目に統一しなければならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.3.4 ファイル記述項 (file description entry)」

2.4 手続き部 (テキストファイル機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

2.4.1 CLOSE 文 (テキストファイル機能)

ここでは、順ファイルと異なる規則だけを記述する。

構文規則

1. REEL/UNIT 指定は覚え書きとみなされる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.6 CLOSE 文」

2.4.2 OPEN 文 (テキストファイル機能)

ここでは、順ファイルと異なる規則だけを記述する。

構文規則

1. REVERSED 句, NO REWIND 句は指定できない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.30 OPEN 文」

2.4.3 READ 文 (テキストファイル機能)

READ 文については、順ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」

2.4.4 REWRITE 文 (テキストファイル機能)

REWRITE 文については、順ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」

2.4.5 WRITE 文 (テキストファイル機能)

WRITE 文については、順ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

2.5 入出力状態の値 (テキストファイル機能)

順ファイルと異なる値だけ記述する。

1. 入出力状態 = 04

入力したレコードの長さが、そのファイル固有属性に従っていない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「5.1.12 入出力状態」

3

ファイル共有機能

この章では、ユーザ間のファイルの共有について説明する。

3.1 環境部（ファイル共用機能）

CSV ファイル，および HiRDB による索引ファイルでは，ファイル共用に関する指定は覚え書きとする。

テキストファイルのファイル共用に関する指定は，Windows COBOL2002 では覚え書きとする。UNIX COBOL2002 では，ファイル全体の施錠は有効であるが，レコード施錠に関する指定は覚え書きとする。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

3.1.1 ファイル記述項（ファイル共用機能）

形式

書き方 1 順ファイル

```
SELECT [OPTIONAL] ファイル名1  
ASSIGN TO { 外部装置名1 } ...  
          { 定数1 }  
[RESERVE 整数1 [ AREA ]  
                [ AREAS ] ]  
[[ORGANIZATION IS SEQUENTIAL]  
[ACCESS MODE IS SEQUENTIAL]  
[PADDING CHARACTER IS { データ名1 }  
                       { 定数2 } ] ]  
[RECORD DELIMITER IS { STANDARD-1 }  
                     { NATIVE } ] ]  
[ LOCK MODE IS { AUTOMATIC  
                [ WITH LOCK ON RECORD ]  
                EXCLUSIVE } ] ]  
[FILE STATUS IS データ名4].
```

書き方 2 索引ファイル

SELECT [OPTIONAL] ファイル名1
 ASSIGN TO { 外部装置名1 } ...
 { 定数1 }
 [RESERVE 整数1 [AREA]
 [AREAS]]
 [ORGANIZATION IS] INDEXED
 [ACCESS MODE IS { SEQUENTIAL }
 { RANDOM }
 { DYNAMIC }]
 RECORD KEY IS データ名1
 [ALTERNATE RECORD KEY IS
 データ名2 [WITH DUPLICATES]] ...

[LOCK MODE IS { MANUAL WITH LOCK ON
 MULTIPLE RECORDS
 AUTOMATIC
 [WITH LOCK ON RECORD]
 EXCLUSIVE }]

[FILE STATUS IS データ名3].

書き方 3 相対ファイル

SELECT [OPTIONAL] ファイル名1
 ASSIGN TO { 外部装置名1 } ...
 { 定数1 }
 [RESERVE 整数1 [AREA]
 [AREAS]]
 [ORGANIZATION IS] RELATIVE

[ACCESS MODE IS
 { SEQUENTIAL }
 [RELATIVE KEY IS データ名1]
 { RANDOM }
 { DYNAMIC }
 RELATIVE KEY IS データ名1]

[LOCK MODE IS { MANUAL WITH LOCK ON
 MULTIPLE RECORDS
 AUTOMATIC
 [WITH LOCK ON RECORD]
 EXCLUSIVE }]

[FILE STATUS IS データ名2].

(備考)

このシステムでは、RELATIVE KEY 句を ACCESS MODE 句から独立した一つの句として記述してもよい。

書き方 4 テキストファイル

```

SELECT [OPTIONAL] ファイル名1
  ASSIGN TO { 外部装置名1 } ...
            { 定数1 }
[ORGANIZATION IS] LINE SEQUENTIAL
[ACCESS MODE IS SEQUENTIAL]

[ LOCK MODE IS { AUTOMATIC
                 [WITH LOCK ON RECORD]
                 EXCLUSIVE } ]

[FILE STATUS IS データ名1].

```

機能

LOCK MODE 指定は、ファイルに対して使われる施錠方法を指示する。

Windows COBOL2002 では、テキストファイルでのファイル共用に関する指定は覚え書きとする。UNIX COBOL2002 では、ファイル全体の施錠は有効であるが、レコード施錠に関する指定は覚え書きとする。

なお、詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のファイル共用の説明を参照のこと。

一般規則

1. LOCK MODE 指定を省略すると、ファイルが INPUT モードで開かれていないかぎり排他となる。INPUT モードでは、OPEN 文が成功すると、ファイルは共用となる。
2. LOCK MODE 指定が EXCLUSIVE のとき、OPEN 文が成功するとファイルは排他となる。
3. LOCK MODE 指定が AUTOMATIC または MANUAL のとき、OPEN 文が成功するとファイルは共用となる。
4. WITH LOCK ON RECORD は覚え書きである。
5. LOCK MODE 指定が AUTOMATIC のとき、READ 文の実行が成功するとレコード施錠が要求され、その直後の入出力文が成功するとレコード施錠を解放する。ただし、START 文についてはレコード施錠を解放しない。
6. LOCK MODE 指定が MANUAL のとき、WITH LOCK 指定のある READ 文によってレコード施錠される。
7. LOCK MODE 指定が EXCLUSIVE で開かれたファイルは、ほかの実行単位から開くことはできない。LOCK MODE 指定が EXCLUSIVE の INPUT モードで開かれたファイルに対し、ほかの実行単位が INPUT モードであっても開くことはできない。
8. 共用モードで開かれたファイルは、排他モードを要求しない複数の実行単位から共用できる。
9. 整列併合用ファイルに対して LOCK MODE 指定はできない。

3.2 手続き部（ファイル共用機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

3.2.1 CLOSE 文（ファイル共用機能）

一般規則

1. CLOSE 文の実行が成功すると、閉じられたファイルのレコード施錠、ファイル施錠はすべて解放される。

（備考）

CLOSE 文の WITH LOCK 指定はファイル共用の機能ではない。つまり、ファイルに対してレコード施錠やファイル施錠を保持しているのではない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.6 CLOSE 文」

3.2.2 DELETE 文（ファイル共用機能）

一般規則

1. DELETE 文は、削除するレコードがほかの実行単位に施錠されているとき成功しない。
2. DELETE 文の実行が成功すると、削除したレコードのレコード施錠は解放される。

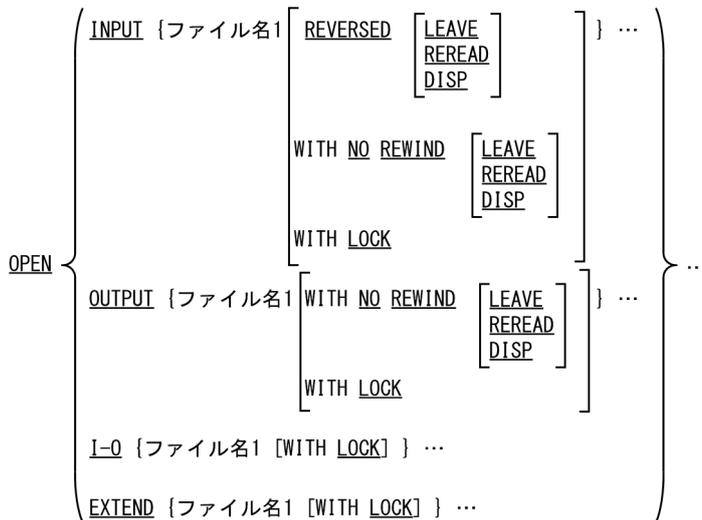
【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.10 DELETE 文」

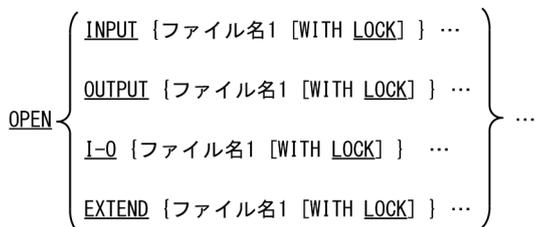
3.2.3 OPEN 文（ファイル共用機能）

形式

書き方 1（順ファイル）



書き方 2 (相対ファイル, 索引ファイル)



一般規則

1. WITH LOCK 指定があるとき、OPEN 文でファイル全体の施錠が要求される（ファイル管理記述項で LOCK MODE IS EXCLUSIVE を指定したときと同様）。
2. ファイル管理記述項で LOCK MODE 指定が EXCLUSIVE または OPEN 文で WITH LOCK 指定があるとき、OPEN 文の実行が成功するとファイルはその実行単位によって排他される。OPEN 文が施錠されているために失敗したとき、入出力状態には'93'が設定される。
3. LOCK MODE 指定が AUTOMATIC または MANUAL で、OPEN 文に WITH LOCK 指定がないとき、ファイルは共用される。共用されたファイルは複数の実行単位で開ける。OPEN 文が施錠されているために失敗したとき、入出力状態には'93'が設定される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.30 OPEN 文」

3.2.4 READ 文 (ファイル共用機能)

形式

書き方 1

READ ファイル名1 [NEXT] RECORD
[INTO 一意名1]
[WITH [NO] LOCK]
[AT END 無条件文1]
[NOT AT END 無条件文2]
[END-READ]

書き方 2

READ ファイル名1 RECORD
[INTO 一意名1]
[WITH [NO] LOCK]
[KEY IS データ名1]
[INVALID KEY 無条件文3]
[NOT INVALID KEY 無条件文4]
[END-READ]

構文規則

1. READ 文の LOCK 指定は、LOCK MODE 指定が EXCLUSIVE のとき指定できない。

一般規則

1. ファイルが INPUT モードで開かれているとき、READ または READ WITH LOCK 文でレコードを施錠することはできない。
2. 暗黙的または明示的に施錠を要求する READ 文は、呼び出そうとするレコードがほかの実行単位によって施錠されていないときだけ成功する。レコードがほかの実行単位によって施錠されているとき、READ 文は不成功となり、入出力状態には'99'が設定され、ファイル位置指示子の変更はしない。
3. READ 文を実行するときに次のレコードが存在しないとき、ファイルの共用にかかわらず、AT END 状態を返す。この状態は、例えば、ほかの実行単位で EXTEND モードで開かれているファイルに対して発生する。LOCK MODE 指定が AUTOMATIC または MANUAL のとき、ほかの実行単位で EXTEND モードで追加されるレコードを読むことができる。LOCK MODE の指定がないときは保証しない。
4. LOCK MODE 指定が AUTOMATIC のとき、それぞれのレコードは読み込みによって施錠され、START 文の規則の記述を除き、その実行単位の入出力文の呼び出しで解放される。WITH NO LOCK 指定のある READ 文はレコード施錠を要求しない。
5. LOCK MODE 指定が MANUAL のとき、その実行単位の READ 文に WITH LOCK 指定があるときだけレコード施錠を要求する。指定がなければ、レコード施錠は要求しない。

(備考)

LOCK MODE 指定が AUTOMATIC または MANUAL で共用によって開かれているファイルについては、オープンモードに関係なく READ 文でレコード施錠の検知をする。LOCK MODE の指定がなく INPUT モードで共用によって開かれているファイルについては、READ 文でレコード施錠の検知は行わない。

施錠されているレコードを再読したとき、次の結果になる可能性がある。

- ・レコードがまだ施錠されている。または再び施錠された。
- ・レコードが削除された場合、KEY 指定のある READ 文は無効キー状態を返す。

- ・レコードが削除された場合、READ NEXT 文はファイルの終わり状態を返すか、または先の READ 文と異なるレコードを読み込もうとする。
 - ・ファイル位置指示子を選択した個所に、新しいレコードが挿入されたとき、そのレコードが施錠されていなければ、READ NEXT 文で新しいレコードが読み込まれる。
- ただし、このシステムでは、このような位置に新しくレコードが挿入されてもファイル位置指示子が示すレコードは、施錠されたレコードのまま、新しく挿入されたレコードをポイントし直すということはない。そのため、この場合の READ NEXT 文では、常に同じレコードを読み込もうとする。すなわち、新しく挿入されたレコードが読み込まれることはない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」

3.2.5 REWRITE 文 (ファイル共用機能)

一般規則

1. 書き換えられるレコードが、ほかの実行単位によって施錠されているときは REWRITE 文は成功しない。
2. LOCK MODE 指定が AUTOMATIC のとき、REWRITE 文が成功すると、存在するレコード施錠は解放される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」

3.2.6 START 文 (ファイル共用機能)

一般規則

1. START 文は、レコード施錠を要求することも、検知することもしない。LOCK MODE 指定が AUTOMATIC のとき、同じ実行単位で取得されたレコード施錠を解放しない。

(備考)

START 文によってレコードに位置づけられた直後の READ NEXT 文で、「ファイルの終わり」または「レコード施錠」状態が返される可能性がある。それは START 文と READ NEXT 文の間に、ほかの実行単位によってレコードが削除または施錠された場合である。READ 文によってキーを取得し、その直後に取得したキーを使って KEY EQUAL または NOT LESS 指定をした START 文も同様に、「無効キー」状態が返される可能性がある。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.45 START 文」

3.2.7 UNLOCK 文 (ファイル共用機能)

形式

`UNLOCK` ファイル名1 $\left[\begin{array}{l} \text{RECORD} \\ \text{RECORDS} \end{array} \right]$

機能

UNLOCK 文は、指定されたファイルに対して実行単位によって取得したすべてのレコード施錠を解放する。

一般規則

1. ファイル名 1 で参照されるファイルは、OPEN 文ですでに開かれていなければならない。
2. UNLOCK 文は、施錠されていなくても成功する。

3.2.8 WRITE 文 (ファイル共用機能)

一般規則

1. 二つまたはそれ以上の実行単位で EXTEND モードで開かれ、共用している相対ファイルに対しレコードを追加するとき、追加するレコードの相対キーの値は必ずしも連続していないが、昇順になっている。
2. 二つまたはそれ以上の実行単位で EXTEND モードで開かれ、共用している索引ファイルに対しレコードを追加するとき、ALTERNATE RECORD KEY 句に DUPLICATES 指定があれば追加するレコードの副レコードキーの重複を許す。DUPLICATES 指定がなければ許さない。
3. LOCK MODE 指定が AUTOMATIC のとき、WRITE 文が成功すると存在しているレコード施錠は解放される。

(備考)

主レコードキーが一意で昇順のときでも、二つまたはそれ以上の実行単位で EXTEND モードで開かれ共用されている索引ファイルに対し、WRITE 文を実行すると、「不当キー、順序エラー」を返し、失敗となる可能性がある。したがって、利用者は、索引ファイルを EXTEND モードで開くとき、排他を指定するべきである。

二つまたはそれ以上の実行単位で追加するレコードの相対キーは、必ずしも追加された位置とは同一でない可能性がある。したがって、利用者は相対ファイルを EXTEND モードで開く場合、排他を指定するべきである。

順ファイルを EXTEND モードで開き、二つまたはそれ以上の実行単位でレコードを追加するとき、実際の書き出し位置はファイルシステムに依存している。

したがって、利用者は順ファイルを EXTEND モードで開く場合、排他を指定するべきである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

4

索引ファイル機能の拡張

この章では、索引ファイル機能の拡張の概要とそれを実現するための規則について説明する。

4.1 機能概要 (索引ファイル機能の拡張)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「5.1.7(3) 索引編成」

4.1.1 合成キー

レコードのフィールドに対し、複数のインデクスを定義すると、複数のレコードキーを利用して目的に応じたレコードを検索できる。また、レコードの複数のフィールドを組み合わせてインデクスを定義すれば、それらのフィールドを一つのレコードキーとして扱える。これを合成キーという。

4.1.2 レコード下降順アクセス機能

レコードのフィールドに対してインデクスを定義して、位置づけをしたレコードからレコードキーの下降順（大きい値から小さい値）に呼び出せる。これをレコード下降順アクセス機能という。

4.2 環境部 (索引ファイル機能の拡張)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

4.2.1 ファイル管理記述項 (FILE-CONTROL) (索引ファイル機能の拡張)

ここでは、ALTERNATE RECORD KEY 句と、RECORD KEY 句について説明する。

合成キーを使用する場合の索引ファイルのファイル管理記述項の書き方を次に示す。

形式

```
FILE-CONTROL.  
SELECT句  
ASSIGN句  
[RESERVE句]  
[ORGANIZATION句]  
[ACCESS MODE句]  
RECORD KEY句  
[ALTERNATE RECORD KEY句]  
[FILE STATUS句]
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4 ファイル管理記述項」

(1) ALTERNATE RECORD KEY 句 (索引ファイル機能の拡張)

形式

```
ALTERNATE RECORD KEY IS  
{ データ名1  
  { 合成キー名1 { SOURCE IS } { データ名2 } ... } [WITH DUPLICATES]  
  {  = } }
```

機能

索引ファイルのレコードに副レコードキーによるアクセス経路を指定する。このとき、合成キーを副レコードキーとして定義できる。

構文規則

1. データ名 1 を指定した場合、合成キーを使用できない。通常の副レコードキーとして処理される。合成キーを使用する場合は、合成キー名 1 を指定する。
2. 合成キー名 1 は、先頭が英字の 31 文字以内の英数字でなければならない (利用者定義語の規則に従う)。
3. データ名 2 は、基本項目でなければならない。集団項目であってはならない。
4. データ名 2 は、添字付けが必要な基本項目であってはならない。

5. データ名 2 は、修飾してもよいが、部分参照してはならない。
6. データ名 2 は、1~8 個まで指定できる。
7. データ名 2 の項目の長さの合計は、255 バイト以下でなければならない。
8. 索引ファイルが可変長項目レコードを含むとき、データ名 2 のサイズは、そのファイルの最小レコードの範囲内に含まれていなければならない。
9. データ名 2 は、再定義項目またはその従属項目であってもよい。ただし、領域が重なってはならない。
10. データ名 2 は、ALTERNATE RECORD KEY 句が書かれたファイル名に関連するレコード記述項中で定義されていなければならない。
11. データ名 2 は、可変長項目の後にある項目であってはならない。
12. データ名 2 は、英数字編集項目、日本語編集項目、ブール項目、アドレスデータ項目、指標データ項目、可変長項目、オブジェクト参照データ項目、外部 10 進項目、内部 10 進項目、外部浮動項目、数字編集項目、英字項目、日本語項目であってはならない。
13. データ名 2 は、8 バイトの 2 進項目であってはならない。
14. 合成キーを構成するフィールドの順は、データ名 2 で指定された順に従う。
15. データ名 2 の指定は、レコード記述項中の並びに関係しない。順不同である。
16. データ名 2 が COMP-X 項目のときの規則については、「25.6.2 COMP-X 項目」の「(6) COMP-X 項目が 2 進項目として扱われる個所」と「25.6.2 COMP-X 項目」の「(9) 索引ファイルと COMP-X 項目」を参照のこと。

一般規則

1. 合成キー名 1 は、START 文、READ 文の KEY 句だけで参照できる。
2. データ名 2 は、レコード中での相対位置、長さおよびデータの属性をファイル生成のときに定義したものと同じにしなければならない。
3. データ名 2 を記述する個数は、ファイル生成のときに定義したものと同じにしなければならない。
4. ファイルに二つ以上のレコード記述項があるとき、データ名 2 はこれらのレコード記述項の一つだけに記述する必要がある。データ名 2 が二つのレコード記述項にわたっている場合の動作は保証しない。
5. ファイルに二つ以上のレコード記述項があるとき、データ名 2 が記述されているレコード記述項以外のレコード記述項でもデータ名 2 と同じ文字位置が、暗黙的にキーとして参照される。
6. 関連するファイル結合子が外部ファイル結合子のとき、その実行単位でそのファイル結合子に関連するすべてのファイル記述項では、データ名 2 に対して関連するレコード中の同じ相対位置に同じデータ記述項を指定しなければならない。
7. レコードキーの一致、不一致の判定は、マニュアル「COBOL2002 言語 標準仕様編」の ALTERNATE RECORD KEY 句を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(5) ALTERNATE RECORD KEY 句」

(2) RECORD KEY 句 (索引ファイル機能の拡張)

形式

RECORD KEY IS
{ データ名1
合成キー名1 { SOURCE IS } { データ名2 } ... }

機能

索引ファイルのレコードに主レコードキーによるアクセス経路を指定する。このとき、合成キーを主レコードキーとして指定する。

構文規則

1. データ名 1 を指定した場合は、合成キーを使用できない。通常のレコードキーとして処理する。合成キーを使用する場合は、合成キー名 1 を指定する。
2. 合成キー名 1 は、先頭が英字の 31 文字以内の英数字でなければならない (利用者定義語の規則に従う)。
3. データ名 2 は、基本項目でなければならない。
4. データ名 2 は、添字付けが必要な基本データ項目であってはならない。
5. データ名 2 は、修飾してもよいが、部分参照してはならない。
6. データ名 2 は、1~8 個まで指定できる。
7. データ名 2 の項目の長さの合計は、255 バイト以下でなければならない。
8. 索引ファイルが可変長項目レコードを含むとき、データ名 2 のサイズは、そのファイルの最小レコードの範囲内に含まれていなければならない。
9. データ名 2 は、再定義項目またはその従属項目であってもよい。ただし、領域が重なってはならない。
10. データ名 2 は、RECORD KEY 句で書かれたファイル名に関連するレコード記述項中で定義されていないなければならない。
11. データ名 2 は、可変長項目の後にある項目であってはならない。
12. データ名 2 は、英数字編集項目、日本語編集項目、ブール項目、アドレスデータ項目、指標データ項目、可変長項目、オブジェクト参照データ項目、外部 10 進項目、内部 10 進項目、外部浮動小数点数字項目、数字編集項目、英字項目、日本語項目、日本語集団項目であってはならない。
13. データ名 2 は、8 バイトの 2 進項目であってはならない。
14. 合成キーを構成するフィールドの順は、データ名 2 で指定された順に従う。
15. データ名 2 の指定は、レコード記述項中の並びに関係しない。順不同である。
16. データ名 2 が COMP-X 項目のときの規則については、「[25.6.2 COMP-X 項目](#)」の「(6) COMP-X 項目が 2 進項目として扱われる個所」と「[25.6.2 COMP-X 項目](#)」の「(9) 索引ファイルと COMP-X 項目」を参照のこと。

一般規則

1. 合成キー名 1 は、START 文、READ 文の KEY 句だけで参照できる。
2. データ名 2 は、レコード中での相対位置、長さおよびデータの属性をファイル生成のときに定義したものと同じにしなければならない。
3. データ名 2 を記述する個数は、ファイル生成のときに定義したものと同じにしなければならない。
4. ファイルに二つ以上のレコード記述項があるとき、データ名 2 はこれらのレコード記述項の一つだけに記述する必要がある。データ名 2 が二つのレコード記述項にわたっている場合の動作は保証しない。
5. ファイルに二つ以上のレコード記述項があるとき、データ名 2 が記述されているレコード記述項以外のレコード記述項でもデータ名 2 と同じ文字位置が、暗黙的にキーとして参照される。
6. 関連するファイル結合子が外部ファイル結合子のとき、その実行単位でそのファイル結合子に関連するすべてのファイル記述項では、データ名 2 に対して関連するレコード中の同じ相対位置に同じデータ記述項を指定しなければならない。
7. レコードキーの一致、不一致の判定は、マニュアル「COBOL2002 言語 標準仕様編」の RECORD KEY 句に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(14) RECORD KEY 句」

4.3 データ部 (索引ファイル機能の拡張)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

4.3.1 DATA FORMAT 句 (索引ファイル機能の拡張)

形式

$$\text{DATA FORMAT IS } \left. \begin{array}{l} \text{STANDARD} \\ \text{REVERSE} \\ \text{NATIVE} \end{array} \right\}$$

機能

DATA FORMAT 句は、外部記憶装置上のデータ表現に使うデータ形式を指定する。

UNIX COBOL2002 では、DATA FORMAT 句は使用できない。

構文規則

1. DATA FORMAT 句は、索引ファイルにだけ指定できる。
2. DATA FORMAT 句は、固定長レコードのファイルにだけ指定できる。
3. NATIVE 指定以外の DATA FORMAT 句を指定したファイルに從属するレコード記述は一つでなければならない。
4. DATA FORMAT 句は、レコード記述中に REDEFINES 句がある場合、被再定義項目と再定義項目の両方に 2 進項目があってはならない。

一般規則

1. STANDARD は、2 進データがビッグエンディアンであることを示す。
2. REVERSE は、2 進データがリトルエンディアンであることを示す。
3. NATIVE は、2 進データがシステムの固有形式であることを示す。
4. ファイルに DATA FORMAT 句を書かないと、NATIVE を指定したものとみなす。

4.4 手続き部 (索引ファイル機能の拡張)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

4.4.1 READ 文 (索引ファイル機能の拡張)

形式

書き方 1 順アクセスの READ 文

```
READ ファイル名1 

|          |
|----------|
| NEXT     |
| PREVIOUS |

 RECORD
```

[INTO 一意名1]
[AT END 無条件文1]
[NOT AT END 無条件文2]
[END-READ]

書き方 2 乱アクセスの READ 文

```
READ ファイル名1 RECORD  
[INTO 一意名1]  
[KEY IS { 合成キー名1  
          データ名1 } ]  
[INVALID KEY 無条件文3]  
[NOT INVALID KEY 無条件文4]  
[END-READ]
```

機能

順アクセスの READ 文は、START 文で決定したアクセス順序に従い、ファイルの次の、または前のレコードを使用できるようにする。

乱アクセスの READ 文は、大記憶ファイルの指定されたレコードを使用できるようにする。

構文規則

- 一意名 1 の記憶領域とファイル名 1 のレコード領域とは、同じであってはならない。
- データ名 1 は、ファイル名 1 に関連するレコードキーとして指定されたデータ項目でなければならない。
合成キー名 1 は、SOURCE 指定、または=指定のある RECORD KEY 句、SOURCE 指定、FOR 指定、または=指定のある ALTERNATE RECORD KEY 句に書かれた合成キー名を指定する。
- データ名 1 は、修飾してもよい。合成キー名 1 は、ファイル名で修飾できる。
- 順アクセス法のファイルにアクセスするときには、書き方 1 を用いる。
- 動的アクセス法のファイルのレコードを順アクセスするときには、NEXT 指定または PREVIOUS 指定を書かなければならない。
- 動的アクセス法のファイルのレコードを乱アクセスするときには、書き方 2 を用いる。

7. 動的アクセス法のファイルを乱アクセスした後、NEXT 指定によって順アクセスする場合、これ以降レコードのアクセスはキーの上昇順に行われる。
8. ファイル名 1 に USE 手続きがないときは、INVALID KEY 指定または AT END 指定を書かなければならない。

一般規則

1. ファイル名 1 のファイルは、READ 文を実行する前に入力モードまたは入出力両用モードで開いておかなければならない。
2. 順アクセス法の索引ファイルに対して、NEXT を指定した場合、または NEXT 指定および PREVIOUS 指定のどちらも省略した場合は、START 文による指定に従って、そのファイルから次の、または前のレコードにアクセスする。PREVIOUS を指定した場合は、START 文による指定に関係なく、そのファイルから前のレコードにアクセスする。
3. READ 文の実行によって、ファイル名 1 に関連する入出力状態の値が更新される。
4. 無条件文 2 または無条件文 4 を指定すると、レコードはその実行前に使用できるようになる。無条件文 2 または無条件文 4 を指定しないと、レコードは READ 文に続く文の実行前に使用できるようになる。
5. このファイルにレコード記述項が二つ以上あるときには、それらのレコードは再定義の規則に従って記憶領域中の同じレコード領域を共用する。READ 文の実行が完了したときの、現在レコードの範囲を超えた位置にあるデータ項目の内容は保証しない。
6. INTO 指定のある READ 文の実行結果は、次に示す規則の順に適応したものと等しい。
 - (a) INTO 指定のない READ 文の実行。
 - (b) 現在レコードが CORRESPONDING 指定のない MOVE 文の規則に従って、レコード領域から一意名 1 の領域へ転記される。現在レコードのサイズは、RECORD 句の規則で定める。暗黙の転記は集団項目転記となる。

このレコードは、入力レコード領域と一意名 1 のデータ項目の両方で使用できるようになる。READ 文の実行が不成功になったときには、この転記はされない。一意名 1 に付けた添字は、レコードを読んだ後で転記する直前に評価される。
7. ファイル位置指示子が、次のレコードがないことを示しているとき、次の順序で動作する。
 - (a) ファイル位置指示子の設定に従って、ファイル終了条件を示す値が、ファイル名 1 に関連する入出力状態に設定される。
 - (b) READ 文に AT END 指定を書くと、AT END 指定の無条件文 1 が実行される。ファイル名 1 に関連する USE 手続きは実行されない。
 - (c) READ 文に AT END 指定を書かないと、ファイル名 1 に関連する USE 手続きを書かなければならず、USE 手続きが実行される。その手続きからは、READ 文に続く次の実行文に復帰する。ファイル終了条件になると、その条件を起こした READ 文の実行ができなくなる。
8. READ 文の実行でファイル終了条件または無効キー条件のどれも起こらなかったとき、AT END 指定または INVALID KEY 指定があっても無視され、次のように動作する。
 - (a) ファイル位置指示子が設定され、そのファイル名 1 に関連する入出力状態が更新される。

(b) ファイル終了条件または無効キー条件のどれでもない例外条件になったとき、ファイル名 1 に関連する USE 手続きが実行された後、USE 文の規則に従って制御が移される。

(c) 例外条件にならなかったとき、レコードは、レコード領域で使用できるようになり、INTO 指定があれば暗黙に転記される。制御は、READ 文の終わり、または無条件文 2 が指定されていれば、そこに移る。後者の場合、無条件文 2 で指定されている各文の規則に従って実行が続く。明示的な分岐または条件文が実行されたとき、制御はその文の規則に従って移され、その他では、無条件文 2 の実行の完了によって、制御が READ 文の終わりに移る。

9. READ 文の実行ができなくなった後の、関連するレコード領域の内容および索引ファイルの参照キーは規定しない。ファイル位置指示子には、有効な次のレコードが確立しなかったことが設定される。

10. 書き方 1 の READ 文の実行が開始されるときファイル位置指示子の設定値は、次の規則に従って、使用できるレコードを定めるために使用される。

索引ファイルでのレコードの比較は、現在の参照キーの値による。

(a) ファイル位置指示子が、有効な次のレコードがないことを示しているとき、READ 文の実行は不成功となる。

(b) 先に実行された OPEN 文または START 文によってファイル位置指示子が設定されているとき、そのファイル位置指示子より大きいかまたは等しい（レコード下降順アクセスのときは小さいかまたは等しい）キー値を持つレコードが選択される。

(c) 先に実行された READ 文によってファイル位置指示子が設定されているときで、かつ現在の参照キーが重複を許さない索引ファイルに対してのとき、そのファイル位置指示子より大きい（レコード下降順アクセスのときは小さい）キー値を持つレコードが選択される。

(d) 先に実行された READ 文によってファイル位置指示子が設定されているときで、かつ現在の参照キーが重複を許す索引ファイルに対してのとき、選択されるレコードは次のどれかである。

- ・ キー値がファイル位置指示子に等しく、かつ、その重複したレコードキー値を持つレコードの組の中で、先に実行された READ 文で使用できるようになったレコードの論理的に直後（レコード下降順アクセスのときは直前）にあるレコード

- ・ キー値がファイル位置指示子より大きい（レコード下降順アクセスのときは小さい）最初のレコード

上記の規則を満たすレコードが見つかり、そのレコードがファイル名 1 に関連するレコード領域で使用できるようになる。

上記の規則を満たすレコードが見つからないとき、ファイル位置指示子は次のレコードが存在しないことを示すように設定され、一般規則 7 に従って実行が続けられる。

レコードが使用できるようになると、ファイル位置指示子には使用できるようになったレコードのレコード番号またはキー値が設定される。

11. 読み込んだレコードのサイズがファイル名 1 のレコード記述項で指定された最小のサイズより小さいとき、読み込まれた最後の有効な文字の右にあるレコード領域の部分の内容は不定である。読み込んだレコードのサイズがファイル名 1 のレコード記述項で指定された最大のサイズより大きいとき、レコードは右側が切り捨てられて最大のサイズとなる。どの場合も、入出力状態はレコード長の不整合を示すように設定される。

12. END-READ 指定は、READ 文の範囲を区切る。

13. 動的アクセス法の索引ファイルに対して、NEXT 指定のある書き方 1 の READ 文を書くと、START 文による指定に従って、そのファイルから次の、または前のレコードにアクセスする。PREVIOUS を指定した場合は、START 文による指定に関係なく、そのファイルから前のレコードにアクセスする。
14. 副レコードキーを参照キーとして順アクセスするとき、その値で指定されるレコードは重複して存在することがある。その場合の順アクセスの READ 文は、WRITE 文または REWRITE 文で重複した値を持つレコードを書き出した順に従って、(レコード下降順アクセスのときは書き出した順とは逆の順に) レコードにアクセスする。
15. 書き方 1 の READ 文の参照キーは、関連するファイルに対して直前に実行された OPEN、READ または START 文によって定まる。
16. 書き方 2 の READ 文で KEY 指定を書くと、そのデータ名 1、または合成キー名 1 がこの READ 文の参照キーとなる。動的アクセス法の場合、そのファイルに対するそれ以降の書き方 1 の READ 文は、別の参照キーが指定されるまで、この参照キーを使用する。
17. 書き方 2 の READ 文で KEY 指定を書かないと、主レコードキーがこの文の参照キーとなる。動的アクセス法の場合、そのファイルに対するそれ以降の書き方 1 の READ 文は、別の参照キーが指定されるまで、この参照キーを使用する。
18. 書き方 2 の READ 文を実行すると、ファイル位置指示子に参照キーの値が設定される。この値は、ファイルのレコード中の対応するデータ項目の値と一致する最初のレコードが見つかるまで比較される。副レコードキーが重複した値を持つとき、最初に見つかるレコードは、大記憶管理システムに渡された重複した副レコードキー値を持つ一連のレコードの最初 (レコード下降順アクセスのときは最後) のものである。一致するレコードがあると、ファイル名 1 に関連するレコード領域で使用できるようになる。一致するレコードがないとき無効キー条件になり、READ 文の実行ができなくなる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」

4.4.2 START 文 (索引ファイル機能の拡張)

形式

書き方 1

START ファイル名1

KEY IS	}	EQUAL TO	}	データ名1
		=		
		GREATER THAN		
		>		
		NOT LESS THAN		
		NOT <		
		LESS THAN		
		<		
		GREATER THAN		
		OR EQUAL TO		
		>=		
		LESS THAN		
		OR EQUAL TO		
		<=		
NOT GREATER THAN				
NOT >				

[INVALID KEY 無条件文1]
 [NOT INVALID KEY 無条件文2]
 [END-START]

書き方 2

START ファイル名1 { FIRST } RECORD

{ LAST }

KEY IS { データ名1 }

{ 合成キー名1 }

[INVALID KEY 無条件文1]
 [NOT INVALID KEY 無条件文2]
 [END-START]

機能

レコードを順アクセスするために、ファイルを論理的に位置づけ、アクセス順序を決定する。

構文規則

1. ファイル名 1 は、順アクセス法または動的アクセス法の索引ファイルの名前でなければならない。
2. データ名 1 は、修飾してもよい。
合成キー名 1 は、ファイル名で修飾できる。
3. ファイル名 1 に関連する USE 手続きがないときは、INVALID KEY 指定を書かなければならない。
4. KEY 指定を書くとき、データ名 1 は次のどれかでなければならない。
 - (a) ファイル名 1 に関連する主レコードキーまたは副レコードキーとして指定されたデータ項目。
 - (b) ファイルのレコード中の項類が英数字または数字のデータ項目で、ファイル名 1 に関連する主レコードキーまたは副レコードキーと左端の文字位置を共有し、そのレコードキーの長さより大きくないもの。ただし、この数字項目のデータ属性およびレコード中での相対位置はファイルを生成したときのものと同じでなければならない。
5. 合成キー名 1 は、SOURCE 指定、または=指定のある RECORD KEY 句、SOURCE 指定、FOR 指定、または=指定のある ALTERNATE RECORD KEY 句に書かれた合成キー名を指定する。
6. データ名 1 は、部分参照してはならない。

一般規則

1. ファイル名 1 のファイルは、START 文を実行する前に入力モードまたは入出力両用モードで開いておかなければならない。
2. KEY 指定を書かないと、比較演算子「IS EQUAL TO」が指定されたものとみなす。
3. START 文の実行によって、レコード領域の内容は変わらないし、ファイル名 1 に関連する RECORD 句の DEPENDING ON 指定で書かれたデータ名のデータ項目の内容も変わらない。
4. KEY 指定の比較演算子で指定された比較は、ファイル名 1 のファイルのレコードに関連するキーと、一般規則 5 で指定されるデータ項目との間で比較される。つまり、そのファイルの文字の照合順序に従って、昇順に並んでいる参照キーについて比較される。作用対象の長さが異なる場合、短い方と同じ長さになるように長い方の右側を切り捨てて文字比較をする。
 - (a) ファイル位置指示子は、キーの値が比較条件を満たす最初のレコードの位置に設定される。
 - (b) ファイル中のどのレコードに対しても比較条件が満たされない場合、無効キー条件になり、START 文の実行はできない。
5. 一般規則 4 による比較では、次のデータ項目が使用される。
 - (a) KEY 指定を書くと、データ名 1 または合成キー名 1 で参照されるデータ項目。
 - (b) KEY 指定を書かないと、RECORD KEY 句に書かれたデータ項目。
6. START 文の実行によって、ファイル名 1 に関連する入出力状態の値が更新される。
7. START 文の実行の成功または不成功による制御の移行は、START 文の INVALID KEY および NOT INVALID KEY 指定の有無による。
8. START 文を実行できなくなると、ファイル位置指示子は有効な次のレコードがないことを示すように設定される。参照キーは規定しない。
9. END-START は、START 文の範囲を区切る。
10. 参照キーは、次のとおりに定める。
 - (a) KEY 指定を書かないと、ファイル名 1 に対して指定した主レコードキーが参照キーになる。
 - (b) KEY 指定を書き、データ名 1 または合成キー名 1 がファイル名 1 のレコードキーである場合、そのレコードキーが参照キーになる。
 - (c) KEY 指定を書き、データ名 1 がファイル名 1 のレコードキーでない場合、データ名 1 のデータ項目と左端の文字位置を共有するレコードキーが参照キーになる。この参照キーが、START 文の位置決めで使用される（一般規則 4 参照）。START 文を実行すると、この参照キーは後続の順アクセスの READ 文にも使用される。
11. 重複した副レコードキー値を持つレコードがあるときは、その中で論理的に先頭（レコード下降順アクセスのときは最後）にあるレコードに位置づける。
12. 比較演算子「EQUAL TO」または「=」を指定した場合、ファイル位置指示子はデータ名 1 の値と等しいキーを持つレコードキーに位置づける。これ以後のレコードは、キーの昇順にアクセスされる。

13. 比較演算子「GREATER THAN」または「>」を指定した場合、ファイル位置指示子はデータ名 1 の値より次に大きいキーを持つレコードに位置づけられる。これ以後のレコードは、キーの上昇順にアクセスされる。
14. 比較演算子「NOT LESS THAN」「NOT <」を指定した場合、ファイル位置指示子はデータ名 1 の値と等しいキーを持つレコードに位置づける。等しいキーがなければ、次に大きいキーを持つレコードに位置づける。
これ以後のレコードは、キーの上昇順にアクセスされる。
15. 比較演算子「LESS THAN」または「<」を指定した場合、ファイル位置指示子はデータ名 1 の値より次に小さいキーを持つレコードに位置づける。
これ以後のレコードは、キーの下降順にアクセスされる。
16. 比較演算子「LESS THAN OR EQUAL TO」または「<=」を指定した場合、ファイル位置指示子はデータ名 1 の値と等しいキーを持つレコードに位置づける。等しいキーがなければ、次に小さいキーを持つレコードに位置づける。
これ以後のレコードは、キーの下降順にアクセスされる。
17. 比較演算子「GREATER THAN OR EQUAL TO」または「>=」を指定した場合、ファイル位置指示子はデータ名 1 の値と等しいキーを持つレコードに位置づける。等しいキーがなければ、次に大きいキーを持つレコードに位置づける。
これ以後のレコードは、キーの上昇順にアクセスされる。
18. 書き方 2 の START 文で LAST RECORD を指定した場合、ファイル位置指示子は最大のキーを持つレコードに位置づける。
これ以後のレコードは、キーの下降順にアクセスされる。
このとき、データ名 1 の内容は比較の対象にならない。
19. 書き方 2 の START 文で FIRST RECORD を指定した場合、ファイル位置指示子は最小のキーを持つレコードに位置づける。これ以後のレコードは、キーの上昇順にアクセスされる。
このとき、データ名 1 の内容は比較の対象にならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.45 START 文」

5

ブール形式データを扱う機能

各けたの値が 0 と 1 だけから成るデータをブールデータという。このシステムでは、ブール項目とブール定数を扱うことができる。

このシステムのブール項目の仕様は、1978 年版の CODASYL COBOL に準拠している。

この章では、ブール定数とブール項目の規則について説明する。

5.1 ブール定数（ブール形式データを扱う機能）

ブール定数を表す規則は、次のとおりである。

- 1文字から160文字までのブール文字（0または1）を引用符で囲み、その左に文字Bを付けて構成する。
- ブール定数の値は、文字Bと引用符を外したブール文字列そのものである。各ブール文字の1けたのサイズはデータ記述項のUSAGE句の指定によって、1ビットまたは1バイトとなる。

ブール定数の例を次に示す。

(例)

ブール定数	値
B'1111'	長さが4でブール文字1だけから成るブール定数を表す。
B'0000'	長さが4でブール文字0だけから成るブール定数を表す。
B'0101'	ブール文字0と1を含むブール定数を表す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.2.1(2)(c) ブール定数」

5.2 表意定数（ブール形式データを扱う機能）

表意定数の ZERO, ZEROS, ZEROES は、文脈によってブール文字 0 を表す。

ALL ブール定数と書くと、このブール定数のブール文字列を何回か反復したものを表す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.7.2 ブール式 (Boolean expressions)」

5.3 データの表現形式（ブール形式データを扱う機能）

ブール項目に適用される項類および字類は、標準仕様編で述べた分類に当てはまらない、まったく独立したものである。ブール項目の項類はブール、字類もブールと定める。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.4.2 データ項目や定数の字類と項類」

5.3.1 ブール項目（ブール形式データを扱う機能）

ブール項目は、PICTURE 句の文字列が文字'1'だけから成るもので、ブール文字の列を表す。ブール文字は、USAGE（用途）が BIT（ビット）か DISPLAY（表示用）かによって、おのおの 1 ビットまたは標準データ形式の 1 文字を表示する。

ビットから成るブール項目を内部ブール項目、文字から成るブール項目を外部ブール項目という。

外部ブール項目を部分参照する場合、そのブール項目と同じ長さの英数字項目に再定義されたものとみなす。

内部ブール項目は部分参照してはならない。

内部ブール項目の指定方法および形式を次に示す。

表 5-1 ブール項目の指定方法および表現形式

PICTURE 句の指定	USAGE 句の指定	表現形式
ブール（文字'1'だけの文字列）	DISPLAY	1 バイトで 1 ブール文字（0 か 1）を表現する。
	BIT	1 ビットで 1 ブール文字（0 か 1）を表現する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

5.3.2 標準けた寄せ規則（ブール形式データを扱う機能）

ブール項目は、最左端のブール位置をそろえて収め、必要なら右側にブール文字 0 を補ったり、右側を切り捨てたりする。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.5.7 データ項目内でのデータのけた寄せ」

5.3.3 データ配置の方法（ブール形式データを扱う機能）

USAGE 句で BIT を指定すると、PICTURE 句で指定したビット数を占める。USAGE 句で DISPLAY を指定すると、英数字項目と同じように配置され、PICTURE 句で指定したバイト数を占める。

5.4 データ部（ブール形式データを扱う機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

5.4.1 JUSTIFIED 句（ブール形式データを扱う機能）

JUSTIFIED 句をブール項目に対して指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.36 JUSTIFIED 句」

5.4.2 OCCURS 句（ブール形式データを扱う機能）

形式

```
OCCURS [整数1 TO] 整数2 TIMES  
[DEPENDING ON データ名1]  
[ { ASCENDING }  
  { DESCENDING } ] KEY IS {データ名2} ……  
[INDEXED BY {指標名1} ……]
```

一般規則

1. KEY IS 指定に書くデータ名 2 は、ブール項目を指すものであってはならない。
2. 用途が BIT のブール項目に対して INDEXED BY 指定を書くときは、そのブール項目に対して SYNCHRONIZED 句を指定しなければならない。

（備考）

INDEXED BY 指定があるブール項目に SYNCHRONIZED 句を指定しないと、このシステムでは SYNCHRONIZED 句を仮定する。

3. DEPENDING ON 指定

USAGE 句で BIT を指定したブール項目に、SYNCHRONIZED 句を指定していないときは、そのブール項目には、DEPENDING ON 指定のある OCCURS 句を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.51 OCCURS 句」

5.4.3 PICTURE 句（ブール形式データを扱う機能）

PICTURE 文字列では、標準仕様に加えてブール項目も指定できる。

(1) ブール項目の指定方法

- PICTURE 文字列は、文字「1」およびその反復回数を表す数字を括弧で囲んだものだけから成る。
- 項目の内容は、ブール文字 0 と 1 の組み合わせでなければならない。
- PICTURE 文字列中の「1」は、1 個のブール文字を含む 1 個のブール位置を示し、項目のサイズに数える。
- PICTURE 文字「1」の前に書けるものは、「1」だけである。
- ブール項目では、挿入編集もゼロ抑制編集もできない。
- 項目のサイズは、2,034 以下でなければならない。PICTURE 文字列の指定と項目の表す値の例を次に示す。

PICTURE文字列	項目の内容	実際の値					
1 (5)	<table border="1"><tr><td>31</td><td>30</td><td>31</td><td>30</td><td>31</td></tr></table> ← 5バイト →	31	30	31	30	31	B' 10101'
31	30	31	30	31			
1 (5) USAGE BIT	<table border="1"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table> ← 5ビット →	1	0	1	0	1	B' 10101'
1	0	1	0	1			

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

5.4.4 REDEFINES 句（ブール形式データを扱う機能）

形式

レベル番号

データ名1
FILLER

REDEFINES データ名2

一般規則

1. データ名 2 が、用途が BIT で SYNCHRONIZED 句指定のないブール項目の場合、データ名 1 は同じ属性でなければならない。すなわち、用途が BIT のブール項目で、かつ SYNCHRONIZED 句指定のない項目でなければならない。
2. 再定義項目、または被再定義項目が内部ブール項目のとき、両方の項目のサイズを同じにするか、または再定義項目を小さくしなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.60 REDEFINES 句」

5.4.5 RENAME 句（ブール形式データを扱う機能）

形式

66 データ名1 RENAME
データ名2 [THROUGH } データ名3
 [THRU

一般規則

1. RENAME 句で THRU を用いた場合、THRU の前後のデータ名 2 とデータ名 3 はどちらもその用途が BIT であってはならない。すなわち、内部ブール項目であってはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.61 RENAME 句」

5.4.6 SYNCHRONIZED 句（ブール形式データを扱う機能）

一般規則

1. SYNCHRONIZED 句を指定した内部ブール基本項目は、バイト境界にけた詰めされたビット位置を占める。
2. SYNCHRONIZED 句を集団項目に指定すると、その集団に含まれる基本項目で、用途が BIT である項目すべてに SYNCHRONIZED の指定が適用される。
3. 遊びビット
用途が BIT の項目とそれ以外のデータ項目が交互に並ぶと、バイト境界に合わせて項目を配置する。このとき、前者（用途が BIT）の項目との間にすき間ができることがある。これを遊びビットという。

(a) 遊びビットが生じる場合

- ・レコードの中で、用途が BIT で SYNCHRONIZED 句指定のないブール項目の後ろにそれ以外の項目が続くとき、この二つの項目の間に生じる。
- ・集団項目の最後の項目が、用途 BIT で SYNCHRONIZED 句指定のないブール項目のとき、その項目の後ろに生じる。

(b) 遊びビットの挿入と仮定

- ・出力ファイルまたは作業場所節中のレコード記述項では、用途が BIT で SYNCHRONIZED 句指定のないブール項目を除いて、その他の項目がバイト境界に置かれるようにレコード内遊びビットを挿入する。
- ・入力ファイルまたは連絡節中のレコード記述項では、用途が BIT で SYNCHRONIZED 句指定のないブール項目を除いて、その他の項目の境界を正しくするために、遊びビットが渡されるデータの中にあるものとみなす。

(c) 遊びビットの数と位置の決定

遊びビットが必要かどうかを決定し、必要な場合には加えるべき遊びビットの数と位置を決めるためにこのシステムは次のようにする。

・バイト境界に配置しなければならない項目、すなわち用途が BIT で SYNCHRONIZED 句指定のないブール項目以外の項目の直前、または集団項目の最後の二つの場所のどちらかにある一連の用途 BIT で SYNCHRONIZED 句指定なしのブール項目が占めるブール位置の合計を求め、この値を 8 で割る。この除算の余り (r) が 0 のときは遊びビットは不要で、余りが 0 でないときは加えられるべき遊びビットは 8-r である。これらの遊びビットは、上記の一連の用途 BIT で SYNCHRONIZED 句指定なしのブール項目の直後にあるものとして各レコードに加えられる。レコード内遊びビットは、その直前のブール項目のレベル番号と等しいレベル番号を持つ項目であるかのように定義され、そのブール項目が属する集団のサイズに含まれる。レコード中に、ブール項目でない項目 SYNCHRONIZED 句が指定された項目があるときには、遊びビットのほかに遊びバイトも追加される。

遊びビットの挿入例を次に示す。

(例)

```
01 A.  
05 B.  
10 C    PIC 1(5) BIT.  
10 FILLER PIC 1(3) BIT.  ... (1)  
10 D    PIC 9(3).  
10 E    PIC 1(7) BIT.  
10 FILLER PIC 1    BIT.  ... (2)  
10 F    PIC 1    DISPLAY  
10 G    PIC 1(6) BIT.  
10 FILLER PIC 1(2) BIT.  ... (3)  
05 H.  
10 I    PIC 1(6) BIT.  
10 J    PIC 1(6) BIT.  
10 FILLER PIC 1(4) BIT.  ... (4)
```

(1)~(4)の FILLER 項目は、次の規則に従って挿入される遊びビットである。

(1)数字項目の前

(2)用途が DISPLAY のブール項目の前

(3)集団項目の前

(4)レコードの最後

・表要素内の遊びビット

OCCURS 句が指定されている集団項目に用途が BIT のブール項目が属している場合も遊びビットが加えられる。すなわち、表要素の各繰り返しを集団項目と見て前項の遊びビットに関する規則を適用する。

表要素内の遊びビットの挿入例を次に示す。

(例)

```
01 A.  
05 B.    OCCURS 2 TIMES  
10 C    PIC 1(7) BIT.
```

```
10 FILLER PIC 1 BIT. ... (1)
10 D PIC 9(5).
10 E PIC 1 BIT.
10 FILLER PIC 1(7) BIT. ... (2)
```

(1)と(2)の FILLER 項目が、挿入される遊びビットである。

(1)数字項目の前

(2)集団項目の最後

・用途 BIT の表要素間の遊びビット

OCCURS 句が、SYNCHRONIZED 句指定なしの用途 BIT のブール項目に対して指定されている場合は、表要素の各繰り返しの間には遊びビットは挿入されない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.80 SYNCHRONIZED 句」

5.4.7 USAGE 句（ブール形式データを扱う機能）

ブール項目に書ける USAGE 句は、DISPLAY または BIT だけである。

- BIT 指定は、ブール項目を表現するのにビットを使うことを指定する。ブール項目を定める PICTURE 句を持つ項目に BIT と指定すると、内部ブール項目となる。
- DISPLAY 指定は、ブール項目を表現するのに標準データ形式の文字を使うことを指定する。ブール項目を定める PICTURE 句を持つ項目に DISPLAY と指定すると、外部ブール項目となる。

(例 1)

内部ブール項目
原始文 : PIC 1(5) USAGE BIT VALUE B'10101'
内部表現:

1	0	1	0	1
---	---	---	---	---

← 5ビット →

(例 2)

外部ブール項目
原始文 : PIC 1(5) USAGE DISPLAY VALUE B'10101'
内部表現:

31	30	31	30	31
----	----	----	----	----

← 5バイト →

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

5.4.8 VALUE 句（ブール形式データを扱う機能）

形式

書き方 1

書き方 2

$$\left. \begin{array}{l} \text{VALUE IS} \\ \text{VALUES ARE} \end{array} \right\} \{\text{定数1}\} \dots$$

機能

1. 初期値を設定する（書き方 1）。
2. 条件名に対応する値を指定する（書き方 2）。

一般規則

書き方 1 に関する規則

1. 定数 1 はブール定数または表意定数 ZERO, ZEROS, ZEROES でなければならない。
2. 定数 1 がブール定数の場合、定数 1 の値は、標準けた寄せ規則に従って収められる。

(例)

05 A PIC 1 (5) VALUE B'111'.

項目には、次のような初期値が収められる。

DATA : A

1	1	1	0	0
---	---	---	---	---

なお、上例では用途の指定がないので、用途は DISPLAY とみなされ 5 バイトとなる。もし、用途を BIT と指定すると 5 ビットとなるが、その場合には、上例のブール定数は 3 ビットの「1」を意味することになる。標準けた寄せ規則は用途 BIT の場合も DISPLAY の場合と同様に適用される。

書き方 2 に関する規則

1. ブール項目を条件変数とする条件名では、定数 1 はブール定数または表意定数 ZERO, ZEROS, ZEROES でなければならない。
2. 定数 1 THRU 定数 2 という書き方はできない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

5.5 手続き部（ブール形式データを扱う機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

5.5.1 文に共通な規則（ブール形式データを扱う機能）

(1) ブール式

(a) ブール式の定義

ブール式とは、ブール値を表現するものであり、次のどれかである。

1. ブール項目を参照する一意名、ブール定数、ALL ブール定数または表意定数 ZERO（ブール要素と呼ぶ）。
2. ブール要素の前に単項ブール演算子 NOT を付けたもの（符号付きブール要素と呼ぶ）。
3. ブール要素や符号付きブール要素を二項ブール演算子でつないだもの。
4. ブール式の前に単項ブール演算子 NOT を付けたもの。
5. 複数のブール式を二項ブール演算子でつないだもの。
6. ブール式を括弧で囲んだもの。

上記 1～6 のブール式は次の規則に従う。

- ブール式中の左括弧と右括弧と 1 対 1 に対応し左括弧に対応する右括弧は右側になければならない。
- ブール式は、左括弧、ブール要素またはブール演算子 NOT で始め、右括弧はブール要素で終わらなければならない。
- ブール式中に、ブール項目やブール定数が合わせて 2 個以上含まれるときには、それらはすべて同じ長さでなければならない。ただし、ALL ブール定数は除く。

(b) ブール演算子

ブール演算子は、単項ブール演算子と二項ブール演算子に分けられる。

- 単項ブール演算子
単項ブール演算子には、ブール否定を表す NOT がある。
- 二項ブール演算子
二項ブール演算子の種類と意味は次のとおりである。

二項ブール演算子	意味
AND	ブール積

二項ブール演算子	意味
OR	ブール和
EXOR	排他ブール和

ブール演算子は予約語であり、語の前後に空白を空けなければならない。

(c) ブール式の計算順序

ブール式の各要素の計算順序は次の規則に従う。

- 2個のブール式が AND, OR または EXOR だけでつながれているとき、または 1 個のブール式の直前に NOT があるときは常に AND, OR, EXOR または NOT はブール演算子である。
- 括弧を使ってブール式中の要素の評価順序を指定できる。括弧がある場合には、括弧内のブール式が最初に計算される。括弧が入れ子になっている場合には、最も内側の括弧から外側の括弧へと計算される。
- 括弧がない場合、または括弧で囲まれたブール式が包含関係で同じ水準にある場合、その計算は次の順位に従う。
 - 1 番目：否定 (NOT)
 - 2 番目：積 (AND)
 - 3 番目：和 (OR と EXOR)

(2) 条件式

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.7.4 条件式 (Conditional expressions)」

(a) 比較条件

比較条件では、ブール項目やブール定数を作用対象として比較をすることができる。ブール作用対象の比較は次の規則に従う。

一般規則

1. 比較条件の右辺および左辺にはブール要素だけを書くことができる。
比較条件中にブール演算子が現れてはならない。
(備考)
比較条件中の AND, OR および NOT は論理演算子とみなされる。
2. ブール項目の用途は何でもよい。つまり、外部ブール項目、内部ブール項目のどちらであってもよい。
3. 比較条件の左辺がブール作用対象なら、右辺もブール作用対象でなければならない。
4. 比較演算子は次のどちらかでなければならない。
IS [NOT] EQUAL TO
IS [NOT] =

5. 両辺の対応するブール文字位置のブール文字が、最左端から最右端に至るまですべて等しいとき、両辺は等しいものとみなされる。
6. 両辺の長さは等しくなければならない。

(b) 条件名条件

条件名条件には、ブール項目に関連づけられた条件名を指定できる。

(c) 字類条件

作用対象の字類が、ブールであるかを調べることができる。

形式

$$\text{一意名1 IS [NOT] } \left\{ \begin{array}{l} \text{NUMERIC} \\ \text{ALPHABETIC} \\ \text{BOOLEAN} \end{array} \right\}$$

一般規則

1. BOOLEAN 指定

一意名 1 のデータ項目のブール検査をする。

- ・作用対象である一意名 1 のデータ項目の用途は、表示用 (DISPLAY) のブール項目、英数字項目または集団項目でなければならない。
- ・検査の対象となる項目が、0 または 1 のブール文字だけから成るときに限ってブール検査の結果は真となる。

5.5.2 ブール形式データが書ける文 (ブール形式データを扱う機能)

(1) ACCEPT 文 (ブール形式データを扱う機能)

一般規則

1. 書き方 1 の ACCEPT 文 (ハードウェア装置からデータを読み込んでデータ項目に入れる文) に書く一意名 1 のデータ項目がブールならば、その用途は表示用 (DISPLAY) でなければならない。
2. 書き方 2 の ACCEPT 文 (仮想データ項目 DATE, DAY, TIME または DAY-OF-WEEK の値をデータ項目に入れる文) に書く一意名のデータ項目は、ブール項目であってはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

(2) COMPUTE 文 (ブール形式データを扱う機能)

形式

COMPUTE {一意名1}… = ブール式

機能

ブール式の値をデータ項目に収める。

構文規則

1. 等号「=」の前後には、一つ以上の空白を空ける。
2. 一意名 1 は、ブール項目でなければならない。

一般規則

1. ブール式の値は、MOVE 文の規則に従って、一意名 1 のデータ項目に転記される。一意名 1 のデータ項目を複数指定しているときは、それぞれの一意名 1 のデータ項目にブール式の値が転記される。ブール式の書き方や計算順序については、「5.5.1 文に共通な規則（ブール形式データを扱う機能）」を参照のこと。ブール式の値をデータ項目に転記する例を次に示す。

(例)

```
01 A PIC 1(3) VALUE B'111'.  
01 B PIC 1(3) VALUE B'101'.  
01 C PIC 1(3).  
:  
COMPUTE C = A AND B.
```

上の COMPUTE 文を実行すると、
C に `101` が転記される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.8 COMPUTE 文」

(3) DISPLAY 文（ブール形式データを扱う機能）

一般規則

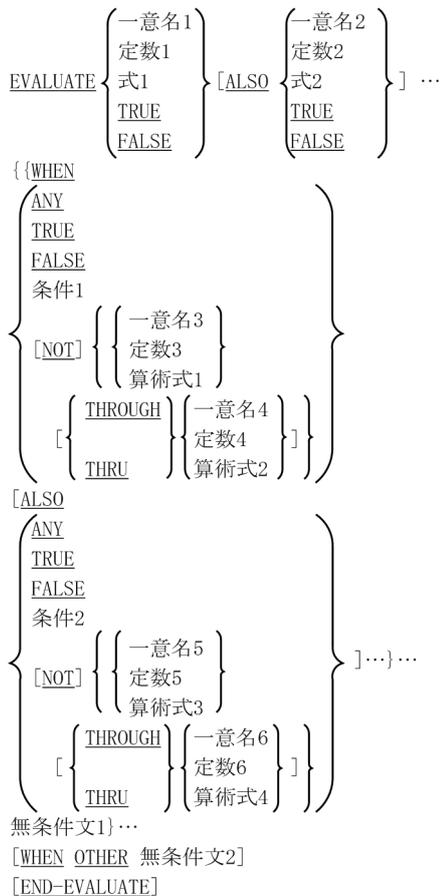
1. 一意名 1 として外部ブール項目および内部ブール項目を指定できる。
外部ブール項目は、表示するときに変換しない。内部ブール項目は、表示するとき外部ブール形式に変換する。転送する項目のサイズは、どちらも PICTURE 句で指定したサイズである。
2. ハードウェア装置へ転送するデータとしてブール定数を指定できる。定数指定のブール定数は、表示するときに変換しない。転送する項目のサイズは、定数の表すサイズである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

(4) EVALUATE 文 (ブール形式データを扱う機能)

形式



構文規則

- 一意名 1~2 はブール項目の名前であってもよい。定数 1~2 はブール定数であってもよい。このとき、範囲式は指定できない。
- 条件 1 または条件 2 は、ブール項目やブール定数を用いた比較条件、ブール項目に関連づけられた条件名条件、ブール検査の字類条件およびそれらを用いた複合条件であってもよい。
- 選択主体や選択対象の中に、ブール演算子が現れてはならない。

一般規則

- 一意名、または定数は、比較条件の規則に従って比較される。
- 条件式は、その評価された真理値が TRUE, FALSE, または対応する条件式の真理値と比較される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.18 EVALUATE 文」

(5) INITIALIZE 文 (ブール形式データを扱う機能)

形式

INITIALIZE [一意名1] ...
[REPLACING {
ALPHABETIC
ALPHANUMERIC
ALPHANUMERIC-EDITED
BOOLEAN
NATIONAL
NATIONAL-EDITED
NUMERIC
NUMERIC-EDITED
OBJECT-REFERENCE
}] DATA BY {一意名2
定数1} ...]

一般規則

1. REPLACING に BOOLEAN を指定した場合、定数 1 はブール定数またはブール値を表す表意定数でなければならない。
2. 一意名 1 にブール項目を指定できる。
REPLACING を指定しない場合、一意名 1 のブール項目にはゼロが初期設定される。
REPLACING を指定した場合、一意名 1 のブール項目は、一意名 2 または定数 1 の値で初期設定される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.24 INITIALIZE 文」

(6) INSPECT 文 (ブール形式データを扱う機能)

TALLYING 指定の一意名 2 (計算用の整数項目) を除くすべての一意名に、用途が DISPLAY であるブール項目 (ブール定数を除く) を指定できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.25 INSPECT 文」

(7) MOVE 文 (ブール形式データを扱う機能)

(a) 項目間の転記の規則

- 受け取り側作用対象が英数字項目のときは、送り出し側作用対象はブール項目またはブール定数であってもよい。その場合、受け取り側作用対象の英数字項目は外部ブール項目とみなされる。
受け取り側作用対象が英数字編集項目のときは、送り出し側作用対象はブール項目であってはならない。
- 受け取り側作用対象がブール項目のときは、送り出し側作用対象はブール項目、ブール定数、0 か 1 から成る英数字項目または表意定数の ZERO でなければならない。送り出し側作用対象が英数字項目の場合、その英数字項目を外部ブール項目とみなして転記する。
このとき項目の転記は、標準的なけた寄せの規則に従って転記される。すなわち、最左端を合わせて収めた上で、送り出し側作用対象の項目が短いときは受け取り側作用対象の右端にブール文字 0 が補われる。送り出し側作用対象の項目が長いときは、右側の余った文字は切り捨てられる。

(例)

```
05 A PIC 1(5).  
05 B PIC X(3).  
05 C PIC X(3).  
:  
MOVE B TO A. ... (1)  
MOVE C TO A. ... (2)
```

B の内容 : 111

C の内容 : 000

のとき、(1)と(2)の MOVE 文を実行すると、ブール項目 A の内容は次のとおりになる。

(1)の場合 11100

(2)の場合 00000

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

(8) プログラム間連絡，利用者定義関数，メソッドの USING 指定と返却項目

呼び出し先プログラム，利用者定義関数，メソッドの手続き部見出し，ENTRY 文，CALL 文，INVOKE 文の USING 指定の作用対象と RETURNING 指定の返却項目にブール項目を書ける。ただし，引数が参照渡しまたは内容渡しの場合，呼び出し元と呼び出し先の双方でバイト境界に調整しておかなければならない。また，INVOKE 文の RETURNING に指定した返却項目もバイト境界に調整しておかなければならない。

6

浮動小数点形式データを扱う機能

この章では、浮動小数点形式データを扱う機能について説明する。

6.1 浮動小数点数字定数

COBOL の標準仕様では、数字定数には固定小数点数字定数しかない。このシステムでは、固定小数点数字定数に加えて、浮動小数点数字定数を扱うことができる。

(備考)

1984 年 CODASYL COBOL には、外部浮動小数点数字項目と浮動小数点数字定数の仕様が入った。

浮動小数点数字定数を表す規則は次のとおりである。

- 仮数部 E 指数部の形式で表す。仮数部、指数部ともに、0~9 の数字、正号 (+)、負号 (-) の文字が使用できる。小数点 (.) は仮数部だけに使用できる。
- 仮数部は 1 けた以上 16 けたまでの数字を含む。指数部は 1 けた以上 2 けたまでの数字を含む。
- 仮数部、指数部ともに、符号文字を複数個書けない。符号を書くときは、仮数部の符号は仮数部の左端に、指数部の符号は指数部の左端におく。符号がない場合は正とみなされる。
- 仮数部には小数点が必ず一つ含まれていなければならない。小数点は、符号文字の左側以外であれば仮数部の任意の文字位置に書いてもよい。
- 浮動小数点数字定数の値は、指数部を 10 の肩にあげたものを仮数部に掛けることによって得られる。値の範囲は絶対値で、 10^{-99} ~ 10^{99} 、または 0 でなければならない。

浮動小数点数字定数の例を次に示す。

(例)

浮動小数点数字定数	値
1.2E03 .12E4	それぞれ 1.2×10^3 と 0.12×10^4 を表す。表現は異なるが値は同じである。
-13.E-2	-13.0×10^{-2} を表す。
3.14E0	3.14×10^0 を表す。
1.2△E△3	誤り。間に空白があるので、1.2 と 3 は固定小数点数字定数、E は語とみなされる。
12E03	誤り。仮数部に小数点がないので、語の 12E03 とみなされる。

6.2 表意定数 (浮動小数点形式データを扱う機能)

表意定数の ZERO が浮動小数点形式の数字項目に転記, または比較されるとき, ZERO は浮動小数点形式データとしての ZERO を意味する。

6.3 データの表現形式（浮動小数点形式データを扱う機能）

数字項目に格納される数値の表現方法の一つに浮動小数点形式がある。

6.3.1 外部浮動小数点形式

PICTURE 句で外部浮動小数点の数字と指定された項目は、外部浮動小数点形式で表現される。その項目の内容は、次の規則に従う。

- 外部浮動小数点形式の数字項目の USAGE は、DISPLAY である。
- 数字を表す文字位置には 0～9 の数字が現れなければならない。
- 符号を表す文字位置には「+」、「-」、または空白が現れなければならない。
- 小数点を表す文字位置には「.」が現れなければならない。
- 仮想部と指数部を区切る文字位置には「E」が現れなければならない。

(注意)

外部浮動小数点形式で表現された数字項目は、そのままでは計算機の機械語命令で演算の対象とすることはできない。このシステムは、外部浮動小数点形式をいったん内部浮動小数点形式などに変換する機械語命令を作り出す。このため実行効率が幾分悪くなるので、計算によく使われる項目は、内部浮動小数点形式のほうがよい。

6.3.2 内部浮動小数点形式

USAGE 句で COMPUTATIONAL-1、または COMPUTATIONAL-2 と指定された項目は、内部浮動小数点形式で表現される。このとき、PICTURE 句を指定してはならない。内部浮動小数点数字項目に割り当てられる記憶領域のサイズは、USAGE 句の指定によって異なる。この関係を次に示す。

表 6-1 USAGE 句の指定とバイト数

USAGE 句の指定	項目が占めるバイト数
COMPUTATIONAL-1	4 バイト
COMPUTATIONAL-2	8 バイト

4 バイトの内部浮動小数点数字項目を単精度内部浮動小数点数字項目、8 バイトの内部浮動小数点数字項目を倍精度内部浮動小数点数字項目と呼ぶ。

浮動小数点数が表現できる値の範囲は、COMPUTATIONAL-1 のときは、絶対値で約 $1.2 \times 10^{-38} \sim 3.4 \times 10^{+38}$ および 0 で、COMPUTATIONAL-2 のときは、絶対値で約 $2.2 \times 10^{-308} \sim 1.8 \times 10^{+308}$ および 0 である。また、有効けた数は、単精度と倍精度では異なる。この関係を次に示す。

表 6-2 内部浮動小数点数の有効けた数

USAGE 句の指定	有効けた数
COMPUTATIONAL-1	約 7 けた
COMPUTATIONAL-2	約 16 けた

(注意 1)

内部浮動小数点形式は、数値を仮数部と指数部に分けて表現したものである。符号や小数点位置などの表現方法は、10 進数値の場合とは異なる。ただし、利用者は、その内部表現形式を考慮する必要はなく、標準データ形式での浮動小数点数 ($+10.5 \times 10^6$ など) を使用する場合と同様に考えればよい。

(注意 2)

一般に浮動小数点数は、数値を正確に表現したものではなく、概略の値を表現するもので誤差を含むことが多い。このため、正確な値が要求される事務計算には、浮動小数点数を使用しないほうがよい。浮動小数点数は、小数点位置を固定小数点では明示できない数や、FORTRAN プログラムに受け渡す指数部を持った数 (FORTRAN での実数) を取り扱う場合に限って使用するとよい。

(注意 3)

マシンや OS の種別によって、浮動小数点の演算結果が異なる場合がある。これは、システムの内部浮動小数点演算の精度、および丸め制御の相違によるものである。丸め制御については、「6.8.1 文に共通な指定と規則 (浮動小数点形式データを扱う機能)」の「(7) 浮動小数点の丸め制御」を参照。

6.4 算術符号（浮動小数点形式データを扱う機能）

数字項目が主記憶装置上で表現されるときには、算術符号（operational sign）を表すためのビット位置を持つ。

- 内部浮動小数点形式では、最左端の 1 ビットが算術符号を表現するのに用いられる。
- 外部浮動小数点形式では、仮数部の左側の 1 バイトに符号を表す文字の「+」か「-」かで算術符号が表現される。

6.5 標準けた寄せ規則（浮動小数点形式データを扱う機能）

項目の転記は標準けた寄せ規則に従う。

浮動小数点形式データには、数字項目の標準けた寄せ規則が適用される。

内部浮動小数点項目から固定小数点形式の数字項目（外部 10 進項目、内部 10 進項目、または 2 進項目）への転記で、固定小数点形式の数字項目のけた数が内部浮動小数点数の有効けた数より大きい場合、数字項目の標準けた寄せ規則に従い、小数点位置を合わせる。この小数点位置を合わせることによって、内部浮動小数点数が格納先のデータ型で表現できない場合は、格納先のデータ型で表現可能な数値へ丸められる。このため、固定小数点形式の数字項目には内部浮動小数点数の有効けたを超える下位のけたに誤差を含んだ値が転記され、演算や比較で使用すると期待する結果にならないことがある。この場合、受け取り側作用対象の固定小数点形式の数字項目に内部浮動小数点項目の有効けた数未満のけた数を指定すること。

小数点位置合わせによって、誤差を含んだ値が転記されるプログラム例を次に示す。このプログラムでは、18 けたの固定小数点形式の数字項目に対し、内部浮動小数点数として表現できない値 998765432100000000 を表現可能な 998765432100000128 に丸めたあと、転記する。

(例)

```
01 A  USAGE COMP-2   VALUE 9.987654321E+0.
01 B  PIC 9(1)V9(9).
01 C  PIC 9(1)V9(17).
      :
COMPUTE B = A.
COMPUTE C = A.
      :
```

(出力結果)

```
B = +9987654321
C = +998765432100000128
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.5.7 データ項目内でのデータのけた寄せ」

6.6 データの配置 (浮動小数点形式データを扱う機能)

6.6.1 データ配置の方法

内部浮動小数点形式の数字項目が、記憶域に占めるバイト数は、USAGE 句の指定で、COMPUTATIONAL-1 のときは 4 バイト、COMPUTATIONAL-2 のときは 8 バイトである。

外部浮動小数点形式の数字項目は、PICTURE 句で指定したサイズを占める。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

6.6.2 語長の境界と効率

内部浮動小数点形式の数字項目は、それぞれ項目の語長の境界に位置づけられていると効率良く操作される。データ項目を語長の境界に位置づけるには二つの方法がある。

- SYNCHRONIZED 句を使う。
- 必要なデータ項目が語長境界に位置づけられるよう、あらかじめデータの構造を決めておく。

SYNCHRONIZED 句を指定したときの項目の配置方法については標準仕様を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.80 SYNCHRONIZED 句」

6.7 データ部（浮動小数点形式データを扱う機能）

ここではおのおのの句で、浮動小数点形式の数字項目に当てはまる規則について記述する。

6.7.1 PICTURE 句（浮動小数点形式データを扱う機能）

内部浮動小数点形式の数字項目には PICTURE 句を指定してはならない。

外部浮動小数点形式の数字項目を PICTURE 句で次のように指定する。

1. PICTURE 文字列は、文字「+」、「-」、「9」、「.」、「V」および「E」の組み合わせから成り、次の形式でなければならない。

$$\left\{ \begin{array}{l} + \\ - \end{array} \right\} \text{仮数部} E \left\{ \begin{array}{l} + \\ - \end{array} \right\} \text{指数部}$$

- 仮数部および指数部の前に文字「+」または「-」がなければならない。
 - 仮数部の後に文字「E」がなければならない。
 - 仮数部は、文字「9」および「.」または「V」の組み合わせから成る。少なくとも1個以上、16個以下の「9」を含まなければならない。仮数部の先頭、間、または終わりに、文字「.」または「V」のどちらか一方を書かなければならない。
 - 指数部は、文字「9」を二つ書いたものから成る。
2. 項目の値は、 $\{\pm\}$ 仮数部 $\times 10^{\{\pm\}}$ 指数部である。値の範囲は、絶対値で単精度の場合 $1.2 \times 10^{-38} \sim 3.4 \times 10^{+38}$ 、または0でなければならない。

PICTURE 文字列の指定と項目の表す例を次に示す。

(例)

PICTURE 文字列	項目の内容	実際の値
+.9999E+99	+.1234E+06	+0.1234 $\times 10^6$
	-.1234E-02	-0.1234 $\times 10^{-2}$
+V9999E+99	+1234E+03	+0.1234 $\times 10^3$
	-1234E+00	-0.1234 $\times 10^0$
-999.99E-99	△123.45E△02	+123.45 $\times 10^2$
	-123.45E-04	-123.45 $\times 10^{-4}$
-999V99E-99	△12345E△08	+123.45 $\times 10^8$
	-12345E△03	-123.45 $\times 10^3$

(備考)

一般に PICTURE 文字列に「+」, 「-」, または「.」を指定すると数字編集項目になるが、外部浮動小数点項目の場合、項類は数字であり計算に使用できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

6.7.2 SYNCHRONIZED 句 (浮動小数点形式データを扱う機能)

SYNCHRONIZED 句を指定したとき、計算機の記憶領域でのデータ項目の表現形式 (USAGE 句) によって項目の配置方法は異なる。内部浮動小数点形式の数字項目を配置するときの規則を次に示す。

USAGE 句	配置される語長の境界	項目が占めるバイト数
COMPUTATIONAL-1	4 バイト境界	4 バイト
COMPUTATIONAL-2	8 バイト境界	8 バイト

(注意)

SYNCHRONIZED 句を省略するときは、バイト境界になるように割り付けられる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.80 SYNCHRONIZED 句」

6.7.3 USAGE 句 (浮動小数点形式データを扱う機能)

形式

$$[\text{USAGE IS}] \left\{ \begin{array}{l} \text{DISPLAY} \\ \text{COMPUTATIONAL-1} \\ \text{COMP-1} \\ \text{COMPUTATIONAL-2} \\ \text{COMP-2} \end{array} \right\}$$

機能

計算機の記憶領域内でのデータ項目の表現形式が浮動小数点形式であることを指定する。

構文規則

1. COMP-1, COMP-2 はそれぞれ、COMPUTATIONAL-1, COMPUTATIONAL-2 の同義語である。

一般規則

1. DISPLAY 指定

数字項目を外部浮動小数点形式で表現するときには、USAGE 句を省略するか、または USAGE IS DISPLAY を指定する。

2. COMPUTATIONAL-1/COMPUTATIONAL-2 指定

内部浮動小数点形式の数字項目であることを指定する。

(a) COMPUTATIONAL-1 のときは4バイトに、COMPUTATIONAL-2 のときは8バイトにそれぞれ収められる。

(b) 内部浮動小数点形式のときは、PICTURE 句を指定してはならない。

(例)

03 A USAGE COMP-1.

03 B USAGE COMP-2.

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

6.7.4 VALUE 句 (浮動小数点形式データを扱う機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

(1) 内部浮動小数点形式の初期値

VALUE 句で内部浮動小数点形式の数字項目に初期値を設定する場合、次の規則に従う。

- 定数は、浮動小数点数字定数、または表意定数の ZERO でなければならない。
- 定数の値は、項目が取り得る値の範囲内になければならず、ゼロでない数字の切り捨てを引き起こしてはならない。

(例)

03 A USAGE COMP-1 VALUE 1.2E03.

03 B USAGE COMP-2 VALUE ZERO.

(2) 外部浮動小数点形式の初期値

外部浮動小数点形式の数字項目には、初期値設定用の VALUE 句を指定できない。

(3) 条件名

形式

$$\text{条件名} \left\{ \begin{array}{l} \text{VALUE IS} \\ \text{VALUES ARE} \end{array} \right\} \left\{ \text{定数1} \left[\left\{ \begin{array}{l} \text{THRU} \\ \text{THROUGH} \end{array} \right\} \text{定数2} \right] \dots \right.$$

一般規則

1. 条件名を定義する VALUE 句は、内部および外部浮動小数点形式のどちらにも指定できる。
2. 浮動小数点項目を条件変数とする条件名では、定数は浮動小数点定数か表意定数の ZERO でなければならない。

6.8 手続き部（浮動小数点形式データを扱う機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

6.8.1 文に共通な指定と規則（浮動小数点形式データを扱う機能）

(1) 算術式

算術式で、数字項目が書けるところならば、どこにでも浮動小数点項目や浮動小数点数字定数を指定できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.7.1 算術式（Arithmetic expressions）」

(2) 条件式

(a) 比較条件

比較条件に浮動小数点形式データを指定できる。その比較方法は、一般の数字項目の場合に準ずる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.7.4 条件式（Conditional expressions）」

作用対象の組み合わせの可否および比較方法を「表 6-3 作用対象の組み合わせによる比較方法」に示す。

(b) 条件名条件

条件名条件に内部、または外部浮動小数点項目に関連づけられた条件名を指定できる。

(c) 字類条件

字類条件には、浮動小数点項目を指定できない。

表 6-3 作用対象の組み合わせによる比較方法

左辺（右辺）	右辺（左辺）	
	外部浮動 小数点項目	内部浮動 小数点項目
集団項目	A	A
英字項目	△	×
英数字項目	△	×
英数字編集項目	△	×

左辺 (右辺)	右辺 (左辺)	
	外部浮動 小数点項目	内部浮動 小数点項目
数字編集項目	△	×
表意定数 (ZERO 除く)	△	×
表意定数 (ZERO)	N	N
英数字定数	△	×
数字定数	N	N
外部 10 進項目	N	N
内部 10 進項目	N	N
2 進項目	N	N
外部浮動小数点項目	N	N
内部浮動小数点項目	N	N

(凡例)

A：文字作用対象として比較

N：数字作用対象として比較

△：書いてもよいが一般に互換性がないので使用しない方がよい。

比較する場合は、文字作用対象として比較する。

×：組み合わせが不可能

注

指標データ項目、ブール項目、オブジェクト参照データ項目、強く型付けされた集団項目との比較はできない。

(3) ROUNDED 指定

算術文で結果を受け入れる項目が、内部、または外部浮動小数点数字項目のときには、その項目に対して ROUNDED 指定を書いてはならない。

(4) ON SIZE ERROR/NOT ON SIZE ERROR 指定

除算およびべき乗を含まない算術文で結果を受け入れる項目が、内部、または外部浮動小数点数字項目のときは、ON SIZE ERROR 指定や NOT ON SIZE ERROR 指定を書いてはならない。

(5) 演算の中間結果についての注意事項

浮動小数点項目や浮動小数点数字定数が算術演算の対象に指定されている場合は、浮動小数点演算が行われる。この場合には中間結果のデータ項目は適用されない。ただし、浮動小数点項目を作用対象に持つ算

術演算で浮動小数点項目が受け取りデータ項目としてだけ指定された場合、算術演算は浮動小数点演算とはならない。

演算の中間結果についてはマニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

(6) けたあふれ条件についての注意事項

(a) 算術文の結果の絶対値が無限大 (Infinity) の場合

関連する結果のデータ項目が浮動小数点項目以外るとき、けたあふれ条件が発生する。

(b) 算術文の結果の絶対値が非数 (Not-a-Number) の場合

けたあふれ条件は発生しない。

(7) 浮動小数点の丸め制御

浮動小数点数が固定小数点項目で表現できない場合、固定小数点項目で表現可能な数値へ丸められる。丸めをする際の方法 (丸めモード) を次の表に示す。例えば、固定小数点項目で表現できない元の浮動小数点数を b とし、固定小数点項目で表現可能な数のうち b の上下で最も近い数を a と c (a, b, c の値は $a < b < c$ の関係にある) とすると、各丸めモードで丸めた結果は次の表のようになる。

表 6-4 丸めモード

丸めモード	丸め処理をした結果
最近値	a と c のうち b に近い方。中間なら偶数の方 (結果の最下位が必ず偶数になる)。
切り捨て (0 方向)	a と c の絶対値が小さい方。

注記: $a < b < c$, a と c は連続した表現可能な数, b は表現できない数

COBOL ではアーキテクチャで最適な演算命令を使用していて、この命令の丸めモードがシステムによって異なる。このため、システムによって切り捨てまたは最近値の丸めモードに従った動作となり、結果として丸められた値が異なる場合がある。浮動小数点数 4.5E0 および 5.5E0 を小数部がない浮動小数点形式以外の数字項目に代入した結果を次の表に示す。

表 6-5 丸めモードにより結果が異なる例

浮動小数点数	丸められた結果	
	最近値	切り捨て (0 方向)
4.5E0	4	4
5.5E0	6	5

なお、システムによって、内部浮動小数点形式データ自体の精度、および丸め制御は異なる（COBOLの仕様ではなく、ハードウェアの浮動小数点形式の演算方法や表現形式の違いによるもの）。このため、丸めモードを合わせたとしても、浮動小数点形式を使用した演算、および転記結果は完全に一致するものではない。

(8) 演算結果や転記結果についての注意事項

無限大 (+INF, -INF), 非数 (NaN), および受け取り側作用対象が格納できる範囲を超える浮動小数点数の値については、演算および転記結果は保証しない。

6.8.2 浮動小数点形式データが書ける文

(1) ACCEPT 文 (浮動小数点形式データを扱う機能)

特定の装置からデータを受け取る場合および日付、または時間をシステムから受け取る場合、ACCEPT 文の作用対象に浮動小数点形式の数字項目が書ける。

ただし、特定の装置からデータを受け取る場合の作用対象は内部浮動小数点形式の数字項目であってはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

(2) ADD 文 (浮動小数点形式データを扱う機能)

ADD 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.2 ADD 文」

(3) CALL 文 (浮動小数点形式データを扱う機能)

CALL 文で指定する USING の作用対象に浮動小数点形式の数字項目が書ける。ただし、内部浮動小数点項目の場合は、呼び出し元プログラムと呼び出し先プログラムの双方でそれぞれの項目を語長の境界に調整しておかなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

(4) COMPUTE 文 (浮動小数点形式データを扱う機能)

算術演算の COMPUTE 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.8 COMPUTE 文」

(5) DISPLAY 文 (浮動小数点形式データを扱う機能)

DISPLAY 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

内部浮動小数点形式の数字項目は、表示するときに、外部浮動小数点形式の数字項目に変換される。

表示するデータの項目のサイズは次のとおりである。

- COMP-1 のとき
PICTURE 句で-.9(8)E-99 と指定した場合と同じサイズ。
すなわち、仮数部が 8 けた、指数部が 2 けたである。
- COMP-2 のとき
PICTURE 句で-.9(17)E-99 と指定した場合と同じサイズ。
すなわち、仮数部が 17 けた、指数部が 2 けたである。
- 指数部が 2 けたをオーバしたときは、指数部を 3 けた表示する。
- 絶対値が無限大 (Infinity) の場合は次のように表示する。

$$\left\{ \begin{array}{l} + \\ - \end{array} \right\} \text{INF}$$

- 値が非数 (Not-a-Number) の場合は次のように表示する。

[-]NANOX 16進数字8けた

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

(6) DIVIDE 文 (浮動小数点形式データを扱う機能)

DIVIDE 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

ただし、REMAINDER 指定がある文の作用対象に浮動小数点形式の数字項目は書けない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.13 DIVIDE 文」

(7) EVALUATE 文 (浮動小数点形式データを扱う機能)

EVALUATE 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.18 EVALUATE 文」

(8) INITIALIZE 文 (浮動小数点形式データを扱う機能)

INITIALIZE 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.24 INITIALIZE 文」

(9) INVOKE 文 (浮動小数点形式データを扱う機能)

INVOKE 文で指定する USING の作用対象と RETURNING の返却項目に浮動小数点形式の数字項目が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.26 INVOKE 文」

(10) MOVE 文 (浮動小数点形式データを扱う機能)

MOVE 文で、浮動小数点項目、または浮動小数点数字定数を用いた場合の可能な組み合わせを「表 6-6 浮動小数点項目、または浮動小数点数字定数を用いた場合の可能な組み合わせ」に示す。なお、項目間の転記の規則は、標準仕様編の MOVE 文に記述してある数字項目の規則に従う。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

「COBOL2002 言語 標準仕様編」 「表 10-9 MOVE 文の種類 of 妥当性」

表 6-6 浮動小数点項目、または浮動小数点数字定数を用いた場合の可能な組み合わせ

送り出し側作用対象	受け取り側作用対象
浮動小数点項目 または浮動小数点数字定数	集団項目 数字編集項目※2 数字項目 (固定小数点形式の数字項目) ※2 数字項目 (浮動小数点項目)
集団項目	数字項目 (固定小数点形式の数字項目) 数字項目 (浮動小数点項目)
英数字項目※1 数字編集項目 表意定数 ZERO 数字項目 (固定小数点) 数字項目 (浮動小数点) 数字定数 (固定小数点) 数字定数 (浮動小数点)	浮動小数点項目

注※1

送り出し側作用対象を符号なし外部 10 進項目とみなして転記する。この組み合わせは一般に互換性がないので使用しない方がよい。

注※2

システムによる丸め制御の相違によって、浮動小数点の転記の結果が異なる場合がある。丸め制御については、「6.8.1 文に共通な指定と規則（浮動小数点形式データを扱う機能）」の「(7) 浮動小数点の丸め制御」を参照のこと。

注

受け取り側作用対象が浮動小数点項目のとき、送り出し側作用対象に、0~9 の数字だけから成る英数字定数や、ALL 定数は指定できない。

(11) MULTIPLY 文（浮動小数点形式データを扱う機能）

MULTIPLY 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.29 MULTIPLY 文」

(12) PERFORM 文（浮動小数点形式データを扱う機能）

VARYING 指定の PERFORM 文に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.31 PERFORM 文」

(13) SUBTRACT 文（浮動小数点形式データを扱う機能）

SUBTRACT 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.48 SUBTRACT 文」

(14) SEARCH 文（浮動小数点形式データを扱う機能）

SEARCH 文の作用対象に、浮動小数点形式の数字項目、または浮動小数点数字定数が書ける。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.41 SEARCH 文」

(15) 手続き部見出し

呼び出し先プログラムや利用者定義関数やメソッドの手続き部見出しで指定する USING の作用対象と RETURNING の返却項目に浮動小数点形式の数字項目が書ける。

7

報告書作成機能の拡張

この章では、報告書作成機能の拡張機能について説明する。

7.1 報告書作成機能の VOS3 COBOL85 からの移行用オプション (-CompatiV3)

ここでは、VOS3 COBOL85 からの移行用オプション (-CompatiV3) で、報告書作成機能を使用することによる相違点だけを記述する。-CompatiV3 オプションの有無に関係なく、共通な一般形式については「COBOL2002 言語 標準仕様編」 「13.1 報告書作成機能とは」を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「13.1 報告書作成機能とは」

7.1.1 特殊レジスタ (報告書作成機能の拡張)

報告書作成機能によって生成される特殊レジスタはプログラム、利用者定義関数、ファクトリ定義、インスタンス定義ごとに報告書記述項に対して生成される。また、報告書作成機能によって生成される特殊レジスタは大域属性を持つ静的データである。

(1) PAGE-COUNTER 特殊レジスタ

PAGE-COUNTER は、報告書記述項に PAGE LIMIT 句の指定がある報告書ごとに作られる特殊レジスタで、報告書作成機能が報告書のページを数えるのに用いる。

- PAGE-COUNTER は、3 バイトの符号なし内部 10 進形式の数字項目で、1～9999 の範囲を表せる。そのけたの数は、PAGE-COUNTER を SOURCE 句の源データ項目に持つ基本項目の PICTURE 句によって決まる。一つの報告書に PAGE-COUNTER を源データ項目に持つ基本項目が複数個あるときには、それらの PICTURE 句が表す数字のけた数は同じでなければならない。
- PAGE-COUNTER は、報告書節および手続き部で参照してもよい。

(2) LINE-COUNTER 特殊レジスタ

LINE-COUNTER は、報告書記述項に PAGE LIMIT 句の指定がある報告書ごとに作られる特殊レジスタで、報告書作成機能が印刷行の表示位置や行送りの数を決めたりページ中の行番号を保持したりするときに用いる。

- LINE-COUNTER は、3 バイトの符号なし内部 10 進形式の数字項目で、1～9999 の範囲を表せる。そのけた数は、LINE-COUNTER を SOURCE 句の源データ項目に持つ基本項目の PICTURE 句によって決まる。
- LINE-COUNTER は、報告書節および手続き部で参照してもよい。手続き部の文で LINE-COUNTER の内部を変更した場合、報告書作成機能が管理しているページの体裁がどうなるかは保証しない。

7.1.2 環境部（報告書作成機能の拡張）

(1) 特殊名段落

形式

SPECIAL-NAMES.
[定数10 IS 呼び名8]…

機能

一つのファイルに複数個の報告書を取めるときに、各報告書の行を識別するための定数を指定する。

一般規則

1. 定数は、一つのファイルに複数個の報告書を取めるときに、各報告書の行を識別するためのものであり、識別名を呼び名で指定する。
 - ・定数は、長さ1文字の英数字定数を書く。ただし、-UniObjGen オプション指定時は、UTF-8で多バイトとなる文字を使用できない。
 - ・定数で英字「E」を指定したときには、PICTURE 句で「E」、すなわち浮動小数点数を指定してはならない。
 - ・値の等しい定数を2回以上指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

7.1.3 データ部（報告書作成機能の拡張）

形式

DATA DIVISION.
FILE SECTION.
{ファイル記述項 {データ記述項} …}…
[WORKING-STORAGE SECTION.
.
.
.]
[LOCAL-STORAGE SECTION.
.
.
.]
[LINKAGE SECTION.
.
.
.]
REPORT SECTION.
{報告書記述項 {報告集団記述項} …}…

構文規則

ファイル節の構成は、一般のファイルと同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

(1) ファイル記述項

形式

FD ファイル名
[BLOCK CONTAINS句]
[RECORD句]
[CODE-SET句]
[LABEL RECORD句]
[VALUE OF句]
[DATA RECORD句]
REPORT句
[RECORDING MODE句]
[GLOBAL句]
[LINAGE句].

構文規則

- REPORT 句を指定したとき、LINAGE 句は指定できない。
- REPORT 句を指定したとき、データ記述項を指定してもよい。

一般規則

1. DATA RECORD 句

REPORT 句を指定したファイル記述項に DATA RECORD 句を指定すると、すなわち、このファイル記述項の後にレコード記述を続けて書くと、一つのファイルに報告書作成機能が作成するレコードと利用者が書き出すレコードの両方を含めることができる。このとき、利用者自身がレコードを書き出すときには、AFTER ADVANCING を指定した WRITE 文を使用しなければならない。

2. RECORD 句（「固定長」形式および「固定長あるいは可変長」形式）

印刷行の行送りを制御するための制御用文字の 1 バイトと、CODE 句を指定したときには印刷行を識別するための文字の 1 バイトを、データレコードのサイズに含める。RECORD 句を省略すると、コンパイラはレコードのサイズとして 133 バイトを仮定する。

可変長レコードでは、各印刷行のサイズは RECORD 句で指定したレコード長の最大値であり、空白行のサイズは 10 バイトである。固定長レコードでは、各印刷行および空白行のサイズが RECORD 句で指定したレコード長の最大値となる。

(2) REPORT 句（報告書作成機能の拡張）

一般規則

1. 報告書名

同じ報告書名を最大二つの REPORT 句で指定できる。すなわち、一つの報告書を最大二つのファイルに収められる。

(3) 報告書記述項

形式

RD 報告書名
[CODE句]
[CONTROL句]
[PAGE LIMIT句]

構文規則

報告書記述項には GLOBAL 句は指定できない。

(4) CODE 句（報告書作成機能の拡張）

形式

WITH CODE 呼び名

一般規則

1. 呼び名

一つのファイルに複数個の報告書を収めるときに、各報告書の行を識別するための文字と関連づける。

(a) 呼び名は、環境部の特殊名段落で長さ 1 文字の英数字定数に関係づけた名前でないといけない。

(5) CONTROL 句（報告書作成機能の拡張）

形式

$$\left\{ \begin{array}{l} \text{CONTROL IS} \\ \text{CONTROLS ARE} \\ \left\{ \begin{array}{l} \{\text{データ名1}\} \dots \\ \text{FINAL}[\text{データ名1}] \dots \end{array} \right\} \end{array} \right\}$$

一般規則

1. データ名は、添字付けできる。

2. このシステムでは、CONTROL HEADING FINAL や CONTROL FOOTING FINAL の報告集団を指定するときでも CONTROL 句で FINAL を書く必要はない。

(6) PAGE LIMIT 句 (報告書作成機能の拡張)

形式

PAGE $\left[\begin{array}{l} \text{LIMIT IS} \\ \text{LIMITS ARE} \end{array} \right]$ 整数1 $\left\{ \begin{array}{l} \text{LINE} \\ \text{LINES} \end{array} \right\}$
[HEADING 整数2]
[FIRST DETAIL 整数3]
[LAST DETAIL 整数4]
[FOOTING 整数5]

一般規則

1. 報告書作成と PAGE LIMIT 句

・PAGE LIMIT 句を指定しないとき

- ・報告書作成機能は、この報告書に対して PAGE-COUNTER と LINE-COUNTER の特殊レジスタを作成しない。
- ・すべての報告集団に相対行番号を指定しなければならない。
- ・この報告書にページ頭書き報告集団が定義されると、最初の GENERATE 文によって表示される報告書頭書き報告集団の後に 1 回だけ表示される。また、ページ脚書き報告集団が定義されると、TERMINATE 文によって表示される報告書脚書き報告集団の前に 1 回だけ表示される。

2. 各整数は符号のない 1 以上の整数で 4 けた以下でなければならない。

(7) 報告集団記述項

形式

書き方 1

01 [データ名]
[LINE NUMBER句]
[NEXT GROUP句]
TYPE句
[USAGE句].

書き方 2

レベル番号 [データ名]
[LINE NUMBER句]
[USAGE句].

書き方 3

レベル番号 [データ名]
 [BLANK WHEN ZERO句]
 [COLUMN NUMBER句]
 [GROUP INDICATE句]
 [JUSTIFIED句]
 [LINE NUMBER句]
 PICTURE句
 [USAGE句]
 {SOURCE句
 SUM句[RESET句]}
 VALUE句

書き方 4

01 [データ名]
 [BLANK WHEN ZERO句]
 [COLUMN NUMBER句]
 [GROUP INDICATE句]
 [JUSTIFIED句]
 [LINE NUMBER句]
 PICTURE句
 [TYPE句]
 [USAGE句]
 [NEXT GROUP句]
 {SOURCE句
 SUM句[RESET句]}
 VALUE句

報告集団の完全な記述は、次のとおりである。

{ 書き方1[[書き方2]{書き方3}...]... }
 { 書き方4 }

構文規則

RESET 句を指定するときには、SUM 句を指定しなければならない。

一般規則

1. データ名

書き方 1 のデータ名には、報告集団名を指定する。その他の書き方のデータ名には、報告書の項目名を指定する。データ名は次の場合以外は、指定する必要はない。指定があってもこのデータ名を環境部、データ部および手続き部で参照できない。

- ・ GENERATE 文で参照する明細報告集団の名前。
- ・ SUM 句の UPON で参照する明細報告集団の名前。
- ・ USE BEFORE REPORTING 文で参照する明細以外の報告集団の名前。
- ・ 報告書節の SOURCE 句、SUM 句または手続きの文で参照する合計カウンタの名前。
- ・ 制御脚書き報告集団中の合計カウンタの修飾語として参照する報告集団の名前。

2. 報告集団記述項の書き方と各句の関連は次のとおりである。

- ・ 書き方 1 の場合
- ・ 報告集団の型を TYPE 句で指定する。
- ・ 書き方 2 の場合

・報告集団が複数個の行から成り、各行が複数個の項目を含むとき、この集団項目記述項の LINE NUMBER 句で従属する基本項目の行番号を指定する。

・この集団項目記述項で LINE NUMBER 句を指定しないで従属する記述項に合計カウンタがないときは、覚え書きとみなす。

・書き方 3 の場合

・印刷行中の印刷表示しなければならない項目を指定するのに用いる。この場合には、COLUMN NUMBER 句と PICTURE 句を指定しなければならない。さらに、SOURCE 句、VALUE 句のうちどれか一つを指定しなければならない。

・明細報告集団中の基本項目で、その報告集団を印刷表示するときに合計値を求める必要がある基本項目を指定するのに用いる。この場合 COLUMN NUMBER 句と PICTURE 句と SOURCE 句を指定する。合計項目は印刷表示しない項目であってもよい。

印刷表示しない項目のときは、COLUMN NUMBER 句、PICTURE 句とを指定しない。

・制御脚書き報告集団中の基本項目で合計カウンタであることを指定するのに用いる。この場合、SUM 句と PICTURE 句を指定しなければならない。

・書き方 4 の場合

報告集団が一つの基本項目から成る「書き方 3 の場合」のどれかの項目であることを指定する。

(8) LINE NUMBER 句 (報告書作成機能の拡張)

形式

```
LINE NUMBER IS  
{ 整数1  
  PLUS 整数2  
  NEXT PAGE }
```

構文規則

絶対行番号および相対行番号指定は、すべての報告集団の最初の LINE NUMBER 句で指定できる。

NEXT PAGE 指定は、報告書の本文と報告書脚書き報告集団の最初の LINE NUMBER 句で指定できる。

一般規則

1. 報告集団との関連

各型の報告集団に対して LINE NUMBER 句の機能と規則を次の表に示す。

2. 整数 1, 整数 2

符号のない 1 以上の整数でなければならない。また、4 けた以下でなければならない。

表 7-1 報告集団の型と LINE NUMBER 句の関連 (-CompatiV3)

報告集団	絶対行番号 (整数 1) 指定	相対行番号 (整数 2) 指定	NEXT PAGE 指定
報告書頭書き報告集団	報告集団の最初の行に指定できる。	<ul style="list-style-type: none">報告集団の最初の行に指定できる。相対行番号を指定すると、PAGE 句の HEADING 整数 2 を基準に数えた相対行番号に最初の行を印刷する。	指定できない。

報告集団	絶対行番号 (整数 1) 指定	相対行番号 (整数 2) 指定	NEXT PAGE 指定
ページ頭書き報告集団	報告集団の最初の行に指定できる。	<ul style="list-style-type: none"> 報告集団の最初の行に指定できる。 相対行番号を指定すると、表示しようとするページに、すでに報告書頭書き報告集団が表示されていれば、LINE-COUNTER の値と相対行番号との和で示される行番号に最初の行を印刷する。その他の場合には、PAGE 句の HEADING 整数 2 を基準に数えた相対行番号に最初の行を印刷する。 	指定できない。
報告書の本文 (制御頭書き, 明細, 制御脚書き報告集団)	<ul style="list-style-type: none"> 報告書最初の行に指定できる。 LINE-COUNTER の値が LINE 句の整数 1 より小さいと、報告集団行の最初の行を現在のページの整数 1 の行に印刷表示する。LINE-COUNTER の値が、LINE 句の整数 1 の値に等しいか大きいと、ページを送って報告集団最初の行を新しいページの整数 1 の行に印刷表示する。 	<ul style="list-style-type: none"> 報告集団の最初の行に指定できる。 現在のページにすでに報告書の本文を印刷表示してあるときには、LINE-COUNTER の値に LINE 句の整数 2 の値を加えた行に報告集団の最初の行を印刷する。ただし、このページに報告集団全体を印刷表示できないときはページを送って報告書の最初の行を新しいページの PAGE 句の FIRST DETAIL 整数 3 の行に印刷表示する。 現在のページにまだ報告書の本文を印刷表示していないときには、報告集団の最初の行を現在のページの PAGE 句の FIRST DETAIL 整数 3 の行に印刷する。ただし、LINE-COUNTER の値が PAGE 句の FIRST DETAIL 整数 3 の値に等しいか大きいと、報告集団の最初の行を現在のページの LINE-COUNTER の値に 1 を加えた行に印刷表示する。 このページに報告集団全体を印刷できないときは、ページを送って報告集団の最初の行を新しいページの PAGE 句の FIRST DETAIL 整数 3 の行に印刷表示する。LINE-COUNTER の値が PAGE 句の FIRST DETAIL 整数 3 の値に等しいか大きくなるのは、直前に印刷した報告書の本文の NEXT GROUP 句で絶対行番号を指定した場合、または PAGE 句の FIRST DETAIL で 1 を指定した場合である。 	<ul style="list-style-type: none"> 指定できる。 ページを送って報告集団の最初の行を PAGE 句の FIRST DETAIL 整数 3 の行に印刷表示する。ただし、LINE-COUNTER の値が FIRST DETAIL 整数 3 の値に等しいか大きいと、報告集団の最初の行を LINE-COUNTER の値に 1 を加えた行に印刷表示する。このページに報告集団を印刷できないときは、ページを送って報告集団の最初の行を新しいページの PAGE 句の FIRST DETAIL 整数 3 の行に印刷表示する。LINE-COUNTER の値が PAGE 句の FIRST DETAIL 整数 3 の値に等しいか大きくなるのは、直前に印刷した報告書の本文の NEXT GROUP 句で絶対行番号を指定した場合である。
ページ脚書き報告集団	報告書の最初の行に指定できる。	<ul style="list-style-type: none"> 報告集団の最初の行に指定できる。 相対番号を指定すると、PAGE 句の FOOTING 整数 5 に 1 を加えた相対行番号に最初の行を印刷表示する。 	指定できない。
報告書脚書き報告集団	報告書の最初の行に指定できる。	<ul style="list-style-type: none"> 報告集団の最初の行に指定できる。 相対番号を指定すると、LINE-COUNTER の値かまたは PAGE 句の FOOTING 整数 5 の値のどちらか大きい方を基準として数えた相対行番号に報告集団の最初の行を印刷表示する。LINE-COUNTER の値が PAGE 句の FOOTING 整数 5 の値よりも大きくなるのは、ページ脚書き報告集団をこのページに印刷した場合である。 	<ul style="list-style-type: none"> NEXT PAGE 指定を書くと、報告書脚書き報告集団だけで単独のページを構成する。 新しいページの PAGE 句の HEADING 整数 2 の行に印刷表示する。

(9) NEXT GROUP 句 (報告書作成機能の拡張)

構文規則

LINE NUMBER 句の指定がなくても、NEXT GROUP 句の指定はできる。

一般規則

1. 報告集団との関連

ページ頭書き報告集団、ページ脚書き報告集団または報告書脚書き報告集団には NEXT GROUP 句を指定してはならない。

2. 整数は 4 けた以下でなければならない。

(10) RESET 句 (報告書作成機能の拡張)

ここでは、RESET 句と SUM 句は独立した句として扱う。

形式

$$\text{RESET ON } \left\{ \begin{array}{l} \text{データ名1} \\ \text{FINAL} \end{array} \right\}$$

構文規則

RESET 句は制御脚書き報告集団中でだけ指定でき、SUM 句と対にして用いなければならない。

一般規則

1. データ名 1

データ名 1 で指定した制御用データ項目の制御の切れ目で合計カウンタをゼロに戻す。

・データ名 1 は、この報告書の CONTROL 句で指定したデータ名 (制御用データ項目の名前) でなければならない。

・データ名 1 は、この RESET 句を指定した制御脚書き報告集団に関する制御用データ項目よりも、より上位のレベルの制御用データ項目の名前でなければならない。

(11) SOURCE 句 (報告書作成機能の拡張)

一般規則

1. 一意名

一意名の代わりに特殊レジスタの TALLY, PAGE-COUNTER, LINE-COUNTER を書いてもよい。

(12) SUM 句 (報告書作成機能の拡張)

形式

$$\text{SUM } \{ \text{一意名1} \} \dots \\ \text{[UPON データ名1]}$$

構文規則

1. SUM 句は、制御脚書き報告集団中の基本項目にだけ指定できる。
2. SUM 句を書いた記述項には、PICTURE 句を指定しなければならない。PICTURE 句で編集用の文字を指定してもよいし、BLANK WHEN ZERO 句などの編集用の句を指定してもよい。これらの編集は合計カウンタを印刷表示するときだけにする。これ以外の場合は、合計カウンタはいつも数字項目を表す。
合計カウンタのサイズは、合計値を求める項目（一意名）の整数部分のけた落ちが生じないよう十分大きくしなければならない。また、関連するデータ項目のけた数+2 とする（英数字項目、または英数字編集項目のときは、2 となる。数字項目のときは、けた数+2 または 18 けたのどちらか小さい方となる）。
3. SUM 句を印刷項目の記述項に書くと、その合計カウンタが源データ項目となり、報告書作成機能は、この合計カウンタの値を表示するために、MOVE 文の規則に従って、印刷項目に転記する。

一般規則

1. 一意名 1

合計値を求めるデータ項目の名前を指定する。

- ・一意名は、ファイル節、作業場所節、局所場所節、連絡節の数字項目の名前で、かつ源データ項目か、または同じ報告書節の合計カウンタの名前でなければならない。
- ・UPON 指定を省略し、かつ SUM 句中の一意名が合計カウンタである場合には、その一意名はこの SUM 句を含んでいる同じ制御脚書き報告集団か、同じ報告書中のこれより低いレベルの制御脚書き報告集団の、どれかで定義されていなければならない。
- ・UPON 指定したときには、一意名は合計カウンタの名前であってはならない。
- ・一意名は部分参照できない。

2. UPON 指定

UPON 指定を書くと、明細集計を選択的にするときに指定する。一つの合計カウンタに対して合計値を求める項目（SOURCE 句で源データ項目指定）を複数の明細報告集団中に指定したとき、特定の明細報告集団の処理中にだけ合計値を求める場合には、UPON のデータ名 1 で、その特定の明細報告集団の名前を指定する。

データ名 1 は次の規則に従う。

- ・データ名 1 は、明細報告集団の名前でなければならない。
- ・データ名 1 は、報告書名で修飾してもよい。

3. 報告書作成機能と SUM 句

・SUM 句の指定があると報告書作成機能は仮想データ項目である合計カウンタを作成する。この合計カウンタは数字項目であり、実行時にこの合計カウンタに一意名 1 の値を加算する。この加算は ADD 文の規則による。

・GENERATE 文または TERMINATE 文の実行中に報告書作成機能は、一意名 1 の値を合計カウンタに加算する。合計カウンタへの加算の時点は一意名 1 の特性で異なり、明細集計 (subtotaling)、横の集計 (crossfooting) および縦の集計 (rolling counters forward) の三つの場合がある。

(i) 一意名 1 がより低いレベルの制御脚書き報告集団の合計カウンタであるとき、合計カウンタとその一意名 1 との加算を縦の集計と呼ぶ。これは、制御の切れ目によって、低いレベルの制御脚書き報告集団が処理されるときにされる。

(ii) 一意名 1 が合計カウンタの名前でこの SUM 句を書いた制御脚書き報告集団中で定義してあるときは横を集計する。

横の集計は、制御の切れ目が起きたとき、この制御脚書き報告集団の処理中にする。横の集計は、合計カウンタの値を制御脚書き報告集団中で記述した順序（上から下へ、左から右への順序）に従って加算する。すなわち、制御脚書き報告集団の最初の合計カウンタに対して、すべての加算がされ、次に制御脚書き報告集団の 2 番目の合計カウンタに対して、すべての加算がされる。この手続きは、すべての横の集計が終わるまで繰り返される。

(iii) 一意名 1 が合計カウンタでないとき、合計カウンタとその一意名 1 との加算を、明細集計と呼ぶ。

SUM 句中に UPON 指定を書くと、その一意名 1 は、指定された明細報告集団に対する GENERATE 文が実行されるたびに加算される。ただし、一意名 1 がこの明細報告集団中の SOURCE 句の源データ項目でないときには、合計カウンタへの加算はされない。

SUM 句中に UPON 指定を書かなければその一意名 1 は、SUM 句が書かれた報告書に対するデータ名を指定したすべての GENERATE 文が実行されるたびに加算される。

(13) TYPE 句（報告書作成機能の拡張）

一般規則

1. PAGE HEADING 指定

この報告集団がページ頭書き報告集団であることを指定する。

- ・一つの報告書には、ページ頭書き報告集団は一つしか指定できない。
- ・報告書作成機能は、ページ頭書き報告集団を各ページの先頭に 1 回ずつ印刷表示する。ただし、次の場合は例外である。

(例外 1)

報告書頭書き報告集団または報告書脚書き報告集団がそれだけで 1 ページを構成する場合には、そのページにページ頭書き報告集団を印刷表示しない。

(例外 2)

報告書頭書き報告集団がそれだけで 1 ページを構成しない場合には、ページ頭書き報告集団を報告書頭書き報告集団に続く 2 番目の報告集団として印刷表示する。

(例外 3)

PAGE LIMIT 句を指定しないとき、最初の GENERATE 文によって印刷表示される報告書頭書き報告集団の後に 1 回だけ印刷表示される。

- ・ページ頭書き報告集団の SOURCE 句は、源データ項目の現在値を用いる。

2. DETAIL 指定

この報告集団が明細報告集団であることを指定する。

- ・一つの報告書には、複数個の明細報告集団を指定してもよい。

- ・明細報告集団の印刷表示は、利用者が手続き部の GENERATE 文で指定する。
- ・明細報告集団中の SOURCE 句は、源データ項目の現在値を用いる。

3. CONTROL FOOTING 指定

この報告集団が制御脚書き報告集団であることを指定する。

- ・一つの報告書には、制御脚書き報告集団は、各制御用データ項目および FINAL に対してそれぞれ一つずつしか指定できない。
- ・報告書作成機能は、制御脚書き報告集団を制御用データ項目に対応する制御集団の最後に 1 回ずつ印刷表示する。FINAL を指定すると TERMINATE 文の実行時に総制御脚書き報告集団を総制御集団の最後に 1 回だけ印刷表示する。
- ・GENERATE 文の実行中に制御の切れ目が起こると 1 番低いレベルの制御脚書き報告集団から始めて、制御の切れ目を起こしたレベルの制御脚書き報告集団までをすべてこの順に印刷表示する。TERMINATE 文を実行すると、すべての制御脚書き報告集団を次に示す順序で印刷表示する。
 - ・小制御脚書き
 - ・中制御脚書き
 - ：
 - ・総制御脚書き（報告書中で 1 回だけ）
- ・制御脚書き報告集団中の SOURCE 句は、源データ項目の現在値を用いる。ただし、源データ項目が制御用データ項目の場合は変化する前の値（その制御集団の最後の明細報告集団に属する値）を用いる。総制御脚書き報告集団の場合は、TERMINATE 文実行時に源データ項目の値を用いる。

4. PAGE FOOTING 指定

この報告集団がページ脚書き報告集団であることを指定する。

- ・一つの報告書には、ページ脚書き報告集団は一つしか指定できない。
- ・報告書作成機能は、ページ脚書き報告集団を各ページの最後に 1 回ずつ印刷表示する。ただし、次の場合は例外である。

(例外 1)

報告書頭書き報告集団または報告書脚書き報告集団がそれだけで 1 ページを構成する場合には、そのページにページ脚書き報告集団を印刷表示しない。

(例外 2)

報告書脚書き報告集団がそれだけで 1 ページを構成しない場合には、そのページでは、ページ脚書き報告集団を報告書脚書き報告集団に先行する報告集団として印刷する。

(例外 3)

PAGE LIMIT 句を指定しないとき、TERMINATE 文によって印刷表示される報告書脚書き報告集団の前に 1 回だけ印刷表示される。

- ・ページ脚書き報告集団の SOURCE 句は、源データ項目の現在値を用いる。

5. REPORT FOOTING 指定

この報告集団が報告書脚書き報告集団であることを指定する。

- ・一つの報告書には、報告書脚書き報告集団は一つしか指定できない。

- ・報告書に対して GENERATE 文が少なくとも 1 回実行されていれば、報告書作成機能は、TERMINATE 文の実行中に、報告書脚書き報告集団を報告書の最後に 1 回だけ印刷表示する。
- ・報告書脚書き報告集団中の SOURCE 句は、TERMINATE 文実行時の源データ項目の値を用いる。

6. 報告集団の制御用データ項目の参照値

制御用データ項目を参照したときの値は、次による。

なお、報告書作成機能が制御の切れ目の検出に使った制御用データ項目の保存された値を、古い値という。

- ・制御脚書き報告集団の制御の切れ目の処理中に、SOURCE 句で制御用データ項目を参照すると、古い値が使われる。

- ・TERMINATE 文が実行されたとき、報告書作成機能は、最も高いレベルの制御の切れ目が検出されたとみなす。そして、制御脚書き報告集団中の SOURCE 句で制御用データ項目を参照すると、古い値が使われる。ただし、総制御脚書き報告集団の場合は、TERMINATE 文実行時の源データ項目の値を用いる。

- ・制御脚書き報告集団の USE 手続きで制御用データ項目を参照すると、古い値が使われる。

- ・報告集団でデータ項目を参照すると上記以外は、すべてその報告集団が処理されるときにそのデータ項目に含まれている現在の値が使われる。

7. 制御脚書き報告集団の処理手順

(処理手順 1)

合計カウンタの横を集計する。すなわち、この報告集団中で定義された合計カウンタであって、同じ報告集団中の SUM 句の作用対象でもある合計カウンタをすべて、SUM 句の主体である合計カウンタに加える。

(処理手順 2)

合計カウンタの縦を集計する。すなわち、この報告集団中で定義された合計カウンタであって、これよりもレベルの高い制御脚書き報告集団中の SUM 句の作用対象でもあるものをすべて、そのレベルの高い制御脚書き報告集団中の合計カウンタに加える。

(処理手順 3)

報告集団のデータ名を参照している USE BEFORE REPORTING 手続きがあれば、USE 手続きを実行する。

(処理手順 4)

SUPPRESS 文が実行されたか、または報告集団が印刷用でなければ、次に処理手順 6 を処理する。

(処理手順 5)

SUPPRESS 文が実行されないで、報告集団が印刷用であれば、印刷行を作成し、報告集団の表示規則に従って、報告集団を表示する。

(処理手順 6)

制御階層中のこのレベルを処理するときに、必要によって合計カウンタにゼロを設定する。

8. 明細報告集団の処理手順

明細報告集団は、利用者が GENERATE 文でその処理時点を指定する。明細以外の報告集団は、報告書作成機能が自動的に処理する。

報告書名を指定した GENERATE 文に対応して行う明細に関する処理は、処理手順 1 による。

(処理手順 1)

合計カウンタの明細集計をする。

(処理手順 2)

SUPPRESS 文が実行されたか、または報告集団が印刷用でなければ、報告集団に対する処理を終了する。

(処理手順 3)

明細報告集団の処理が報告書名を指定した GENERATE 文による場合、報告集団に対する処理を終了する。

処理手順 2 でも処理手順 3 でもなければ、印刷行を作成し、報告集団の表示規則に従って、報告集団を表示する。

(14) VALUE 句 (報告書作成機能の拡張)

構文規則

記号文字を指定してはならない。

7.1.4 手続き部 (報告書作成機能の拡張)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

(1) GENERATE 文 (報告書作成機能の拡張)

一般規則

報告書名を指定すると、報告書作成機能は必要なすべての操作を自動的にするが、明細報告集団の指定数に関係なく明細報告集団の書き出しだけはしない。

(2) SUPPRESS 文 (報告書作成機能の拡張)

一般規則

1. 特定の報告集団の USE 手続き中で SUPPRESS 文を実行すると、その報告集団の印刷表示をしない。つまり、MOVE 1 TO PRINT-SWITCH 文を実行したときと同じである。USE 手続き以外の手続き中で SUPPRESS 文を実行した場合、印刷表示の抑止は保証しない。
2. 自分自身、または自分の上位プログラムに報告書節がないときは、SUPPRESS 文は使用できない。

(3) USE 文（報告書作成機能の拡張）

報告書作成機能に関する USE 文には、報告書作成手続きを指定する USE BEFORE REPORTING 文がある。

(a) USE BEFORE REPORTING 文

形式

USE BEFORE REPORTING 一意名

一般規則

1. 一意名

一意名は、報告書節の明細報告集団以外の報告集団の名前でなければならない。

2. USE 手続きでの報告書作成機能

制御脚書き報告集団の USE 手続きで制御用データ項目を参照すると、変化する前の値を用いる。制御用データ項目以外のデータ項目を参照すると、項目の現在値を用いる。

7.2 報告書作成機能の HITAC8000 シリーズからの移行用オプション (-Cblctr)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「13.1 報告書作成機能とは」

7.2.1 CBL-CTR 特殊レジスタ (報告書作成機能の拡張)

CBL-CTR 特殊レジスタは、手続き部と COBOL の報告書作成機能との間の連絡に使用するために、COBOL が内部的に作成するデータ名であり、次の二つの機能がある。

(1) 報告書作成機能から利用者への報告書の状態の通知

この機能は宣言節の USE BEFORE REPORTING 文 (以下 USE 文と略記する) を指定した節中で指定しなければならない。報告書作成機能は制御の切れ目 (control break) を起こしたレベルを表す調整を CBL-CTR に設定する。利用者は、USE 手続きの中で CBL-CTR の内容を参照でき、その内容によって適当な処置を取ることができる。例えば、USE 文の制御頭書き処理で CBL-CTR の内容が 0 ならば、最初の GENERATE 文が実行され、最初の制御の切れ目 (FINAL) の処理がされている。また、USE 文の制御脚書き処理で CBL-CTR の内容が 0 でないならば、その内容は、報告書作成中に制御の切れ目を起こした制御のレベルを示す整数である。すなわち、その整数が n ならば、CONTROL 句に書いた FINAL を除く、第 n 番目の制御用データ項目に対応する制御の切れ目が起きていることを示している。

USE 文のページ頭書きまたはページ脚書き処理で CBL-CTR の値を参照すると、制御の切れ目が生じた場合の改ページであれば、そのレベルの値 (CBL-CTR = 1~254 の整数) が入っていて、制御の切れ目なしに改ページが起きた場合は CBL-CTR = 255 が入っている。この機能を用いると、利用者はページ頭書きまたはページ脚書きの処理中に CBL-CTR を参照し、制御の切れ目が生じた場合、その制御情報を次のページまで続行するかどうかの処理を自由に選択できる。また、制御の切れ目が生じている間はページ頭書きまたはページ脚書きの出力を抑止できる。

(2) 利用者から報告書作成機能への報告書形式の指定

この機能は、原始プログラムの手続き部の中で報告書の INITIATE 文実行後から最初の GENERATE 文を実行する前までに、MOVE 文で整数 1, 2 または 3 のどれかを CBL-CTR に設定すると使用できる。

(a) MOVE 1 TO CBL-CTR

CBL-CTR に 1 を設定しない場合、制御脚書きで SOURCE 句によって制御用データ項目を参照すると、制御の切れ目処理では制御の切れ目によって変更された新しい値が入っている。すなわち、制御用データ項目の値が変わることによって制御の切れ目が生じ、その時点では制御用データ項目の内容は元の値ではなく新しい値が入っている。

ところが、CBL-CTR に 1 を設定すると、制御の切れ目が生じても、制御脚書きの制御用データ項目には元の値が入る。このため USE BEFORE REPORTING 節中、または SOURCE 句で制御用データ項目を参照すると、制御用データ項目には制御の切れ目を起こした新しい値ではなく、元の値が入っている。

(b) MOVE 2 TO CBL-CTR

CBL-CTR に 2 を設定しない場合、あるレベル制御の切れ目が起きると、制御の切れ目が起きたレベルの制御脚書きを印刷する前に、それより低いレベルの制御用データ項目を持った制御脚書きを印刷する際に、指定されている NEXT GROUP 句をすべて実行する。

ところが、CBL-CTR に 2 を設定すると、印刷する制御脚書きのうち最もレベルの高い制御脚書きの NEXT GROUP 句だけを有効とし、レベルの低いものに関しては制御脚書きの印刷だけをして、NEXT GROUP 句を無視する。

(c) MOVE 3 TO CBL-CTR

CBL-CTR に 1 と 2 を設定した場合の両方の処理をすることを指定する。

CBL-CTR 特殊レジスタの機能一覧を次に示す。

表 7-2 CBL-CTR 特殊レジスタの機能一覧

使用	値	NEXT GROUP 句	SOURCE 句
設定	CBL-CTR に値を設定しない	すべて有効	CF：新しい値 CH：新しい値 USE 手続き：新しい値
	CBL-CTR=1	すべて有効	CF：元の値 CH：新しい値 USE 手続き：制御脚書きのときだけ元の値
	CBL-CTR=2	制御の切れ目を生じた CF に対する NEXT GROUP 句だけ有効	CF：新しい値 CH：新しい値 USE 手続き：新しい値
	CBL-CTR=3	制御の切れ目を生じた CF に対する NEXT GROUP 句だけ有効	CF：元の値 CH：新しい値 USE 手続き：制御脚書きのときだけ元の値
参照	CBL-CTR=0~n	制御の切れ目のレベルを示す。	—

(凡例)

—：該当しない

8

データコミュニケーション機能

データコミュニケーション機能は、種々の端末またはコンピュータシステムとメッセージの受け渡しをする機能である。

この章では、データコミュニケーション機能の形式と機能概略について説明する。

この機能に関する-OpenTP1 オプションは、Windows COBOL2002 で使用できる。

8.1 データコミュニケーション機能とは

データコミュニケーション機能は、オンラインコントロールプログラムを経由して、端末、ファイル、またはほかのプログラムとメッセージの受け渡しをする機能である。ただし、このシステムでは、OpenTP1を対象とする。また、各構文の厳密な意味についてはオンラインコントロールプログラム側で規定しており、この章では、このシステムで行う構文解析についてだけを記述する。

各構文の厳密な意味については、マニュアル「OpenTP1 プログラム作成リファレンス COBOL 言語編」を参照のこと。

8.2 データ部 (データコミュニケーション機能)

形式

DATA DIVISION.
[SUBSCHEMA SECTION. [サブスキーマ記述項]…]
[FILE SECTION. [ファイル記述項]…]
[WORKING-STORAGE SECTION.
[データ記述項]…]
[LOCAL-STORAGE SECTION.
[データ記述項]…]
[LINKAGE SECTION. [データ記述項]…]
[COMMUNICATION SECTION.
[通信記述項]…]

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.1.1 一般形式」

8.2.1 通信記述項 (CD) (データコミュニケーション機能)

COBOL プログラムでは、通信記述項が、通信節の最高のレベルを表す。通信節の見出しの次にレベル指示語の CD、通信記述項および一連の独立した句から成る通信記述項を書く。

形式

CD 通信記述名1

FOR	{	INPUT	}	FILE
		OUTPUT		STORAGE
				JOURNAL
		I-O		PROGRAM
				COMMAND
				NATIVE
				HOST
				REPORT

[MAP NAME IS データ名1]
[MAPPING MODE IS データ名2]
[STATUS KEY IS データ名4]
[SYMBOLIC TERMINAL IS データ名5]
[MESSAGE DATE IS データ名8]
[MESSAGE TIME IS データ名9]
[SYNCHRONOUS MODE IS { SYNC
ASYNC }
データ名10]
[SWITCHING MODE IS { NORMAL
PRIOR }
データ名11]
[NEXT TRANSACTION IS データ名12]
{ ACTIVE { TIME
INTERVAL } IS データ名13
WAITING TIME }
[DETAIL MODE IS {データ名14}…]
[RECEIVING AREA LENGTH IS データ名15]
[NATIVE AREA IS データ名16]
[DESTINATION TYPE IS { PR
CRT }
データ名17]
[TRANSFER MODE IS データ名18]
[PAGE NAME IS データ名6].

機能

通信記述項は、オンラインコントロールプログラムと COBOL プログラムの間の論理的な連絡領域を定める。

構文規則

1. 通信記述項の最初にレベル指示語 CD を書き、その後に通信記述名 1 を書かなければならない。
2. 通信記述名 1 の後に書く句の順序は任意とする。
3. 通信記述中の FOR 句で、次の組み合わせを行ってはならない。
 - ・ INPUT と COMMAND
 - ・ INPUT と PROGRAM
 - ・ INPUT と JOURNAL
 - ・ I-O と JOURNAL
4. DETAIL MODE 句のデータ名は、8 個を超えてはならない。
5. 通信節は、最外側プログラムまたは利用者定義関数だけに指定できる。
6. ACTIVE 句および WAITING TIME 句は、二つ以上同時に指定してはならない。

一般規則

1. 通信記述項は、通信記述名 1 をオンラインコントロールプログラムとの連絡領域に関連づける。
2. 通信記述名 1 の後に書く各句の意味は、オンラインコントロールプログラム側で定める。
3. FOR 句の INPUT, OUTPUT, または I-O に続く指定を省略した場合、メッセージの受信またはメッセージの送信を行う指定となる。

8.3 手続き部 (データコミュニケーション機能)

データコミュニケーション機能の手続き部を書く各文の意味は、オンラインコントロールプログラムが定めるため、このマニュアルでは、各文の形式と構文規則についてだけ記述する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

8.3.1 COMMIT 文 (データコミュニケーション機能)

形式

COMMIT

構文規則

1. COMMIT 文は、通信記述項のないプログラムに書いてもよい。また、特定の通信記述名には関連しない。

8.3.2 DISABLE 文 (データコミュニケーション機能)

形式

DISABLE 通信記述名1 [WITH 一意名1]

構文規則

1. 通信記述名 1 には、DISABLE 文に対応する通信記述項の通信記述名を指定する。
2. 通信記述名 1 で参照する通信記述項は、OUTPUT, INPUT FILE, OUTPUT FILE, I-O FILE, I-O STORAGE, または I-O の指定がなければならない。
3. 一意名 1 には、次のデータ名は指定できない。
 - ・指標データ項目
 - ・アドレスデータ項目
 - ・ブール項目
 - ・オブジェクト参照データ項目
 - ・オブジェクトプロパティ
 - ・関数一意名
4. 一意名 1 は部分参照してはならない。

8.3.3 ENABLE 文 (データコミュニケーション機能)

形式

ENABLE 通信記述名1 [WITH 一意名1]

構文規則

1. 通信記述名 1 には、ENABLE 文に対応する通信記述項の通信記述名を指定する。
2. 通信記述名 1 で参照する通信記述項は、OUTPUT, I-O STORAGE, INPUT FILE, I-O FILE, OUTPUT FILE, または I-O の指定がなければならない。
3. 一意名 1 には、次のデータ名は指定できない。
 - ・指標データ項目
 - ・アドレスデータ項目
 - ・ブール項目
 - ・オブジェクト参照データ項目
 - ・オブジェクトプロパティ
 - ・関数一意名
4. 一意名 1 は部分参照してはならない。

8.3.4 RECEIVE 文 (データコミュニケーション機能)

形式

RECEIVE 通信記述名1 { [FIRST] SEGMENT }
MESSAGE }

[INTO 一意名1]

[BEFORE ERASING] [WAITING]

構文規則

1. 通信記述名 1 には、RECEIVE 文に対応する通信記述項の通信記述名を指定する。
2. 通信記述名 1 で参照する通信記述項は、I-O PROGRAM, INPUT, I-O, INPUT FILE, I-O FILE, INPUT STORAGE または I-O STORAGE の指定がなければならない。
3. 一意名 1 には、次のデータ名は指定できない。
 - ・指標データ項目
 - ・アドレスデータ項目
 - ・ブール項目
 - ・オブジェクト参照データ項目
 - ・オブジェクトプロパティ
 - ・関数一意名

4. 一意名 1 は部分参照してはならない。

8.3.5 ROLLBACK 文 (データコミュニケーション機能)

形式

`ROLLBACK [WITH STOPPING]`

構文規則

1. ROLLBACK 文は、通信記述項のないプログラムに書いてもよい。また、特定の通信記述名には関連しない。

8.3.6 SEND 文 (データコミュニケーション機能)

形式

`SEND 通信記述名1 [FROM 一意名1
[WITH { ESI
EMI }]
[BEFORE RECEIVING MESSAGE INTO 一意名4
PAGE CONTROL IS 一意名3]`

構文規則

1. 通信記述名 1 には、SEND 文に対応する通信記述項の通信記述名を指定する。
2. 通信記述名 1 で参照する通信記述項は、I-O PROGRAM, OUTPUT, I-O, OUTPUT FILE, I-O FILE, OUTPUT COMMAND, I-O COMMAND, OUTPUT STORAGE, I-O STORAGE, OUTPUT PROGRAM または OUTPUT JOURNAL の指定がなければならない。
3. 一意名 1, 一意名 2, 一意名 3 および一意名 4 には、次のデータ名は指定できない。
 - ・ 指標データ項目
 - ・ アドレスデータ項目
 - ・ ブール項目
 - ・ オブジェクト参照データ項目
 - ・ オブジェクトプロパティ
 - ・ 関数一意名
4. 一意名 1, 一意名 2 および一意名 3 は部分参照してはならない。

9

データベースアクセス機能

この章では、データベースアクセス機能について説明する。

この機能は、Windows COBOL2002, Linux COBOL2002 で使用できる。

9.1 概説 (データベースアクセス機能)

データベースアクセス機能は、リレーショナルデータベース管理システム (RDBMS) やそれ以外のデータベース管理システム (DBMS) 上のデータにデータアクセス用の標準言語である構造化照会言語 (SQL) を使ってアクセスする機能である。

SQL を COBOL プログラムに記述したプログラムを SQL 埋め込み COBOL プログラムといい、プログラム中に SQL の文を埋め込むために埋め込み SQL 文を用いる。

ただし、接続するデータベースによって、使用できる機能に制限を持つことがある。

アクセスするデータベースが COBOL 言語用プリコンパイラを用意している場合は、ODBC インタフェース機能と比べ、使用できる構文が多いため、プリコンパイラの使用を推奨する。

AIX COBOL2002 でデータベースにアクセスする場合は、アクセスするデータベースが用意している COBOL 言語用プリコンパイラを使用のこと。

9.2 埋め込み SQL 文 (データベースアクセス機能)

形式

```
EXEC SQL
SQL文
END-EXEC
```

機能

COBOL プログラム中に SQL 文を記述する。

構文規則

1. COPY 文によって展開する本文中、COPY 文中および REPLACE 文中に EXEC, SQL および END-EXEC を指定してはならない。
2. 埋め込み SQL 文は、REPLACE 文の対象とはならない。
3. 埋め込み SQL 文で COBOL プログラムに埋め込むことができる SQL の文を次に示す。

表 9-1 埋め込み SQL 文で COBOL プログラムに埋め込むことができる SQL の文

種類	分類	SQL 文	概要
宣言系	埋め込み SQL 宣言節	BEGIN DECLARE SECTION	埋め込み SQL 開始宣言。
		END DECLARE SECTION	埋め込み SQL 終了宣言。
	埋め込み例外宣言	WHENEVER	埋め込み例外宣言。
	カーソル	DECLARE CURSOR	カーソル宣言。
制御系	コネクション	CONNECT	コネクションを確立する。
		DISCONNECT	コネクションを解除する。
		SET CONNECTION	現行コネクションを変更する。
	トランザクション	COMMIT	コミットでトランザクションを終了させる。
		ROLLBACK	ロールバックでトランザクションを終了させる。
操作系	静的	SELECT	表の指定された行から値を取り出す。
		INSERT	表中に新しい行を作成する。
		DELETE	表から行を削除する。
		UPDATE	表の行を更新する。
	カーソル	OPEN	カーソルを開く。
		FETCH	カーソルを表の次の行に位置づけ、その行を取り出す。
		CLOSE	カーソルを閉じる。

種類	分類	SQL文	概要
	ストアドプロシージャ	CALL	ストアドプロシージャを呼び出す。
	動的	EXECUTE IMMEDIATE	動的 SQL を準備し、実行する。
		PREPARE	動的 SQL を準備する。
		DEALLOCATE PREPARE	準備した動的 SQL を解放する。
		EXECUTE	動的 SQL を実行する。

4. 埋め込み SQL 開始宣言は、埋め込み SQL 宣言節の始まりを表し、埋め込み SQL 終了宣言は、埋め込み SQL 宣言節の終わりを表す。埋め込み SQL 宣言節については「[9.4 データ部 \(埋め込み SQL 宣言節\) \(データベースアクセス機能\)](#)」を参照のこと。
5. 埋め込み SQL 開始宣言および埋め込み SQL 終了宣言以外の埋め込み SQL 文は、手続き部の文が書けるところにならばどこにでも書ける。手続き部に書ける SQL 文については、「[9.6 手続き部 \(宣言系\) \(データベースアクセス機能\)](#)」, 「[9.7 手続き部 \(制御系\) \(データベースアクセス機能\)](#)」および「[9.8 手続き部 \(操作系\) \(データベースアクセス機能\)](#)」を参照のこと。
6. 語 EXEC, SQL および END-EXEC の前後は空白で区切られていなければならない。ただし、END-EXEC とそれに接続する分離符の終止符との間には空白はなくてもよい。

9.3 SQL の概要 (データベースアクセス機能)

埋め込み SQL 文で COBOL プログラムに埋め込むことができる SQL について説明する。

SQL は、COBOL の言語仕様とは異なる 独自の言語仕様を持っている。ただし、動的に準備し、実行する SQL 文については接続するデータベースの規定に従う。

埋め込み SQL 文によるデータベースアクセスの方法は、SQL を COBOL プログラム中に埋め込み実行する方法 (静的な方法) と、実行時に埋め込み SQL の埋め込み変数に設定して実行する方法 (動的な方法) がある。

また、埋め込み SQL 文の SELECT 文は、表から 1 行分の値を取り出す文 (SELECT 文: 単一行) であり、複数行の値を取り出すにはカーソルを使用する。カーソルには静的な方法と動的な方法による書き方がある。

その他にストアードプロシージャを呼び出し、一連の手続きを行う方法もある。

9.3.1 概念 (データベースアクセス機能)

(1) データベース

データベース (database) は、DBMS にあるすべてのデータの集まりとする。また、DBMS によっては、ストアードプロシージャと呼ぶ一連の手続きを DBMS 上に存在させることができる。

(2) 表

表 (table) は、行 (row) と列 (column) で表されるデータの集まりとする。

また、表には次のような形態がある。

表 9-2 表の形態

形態	説明
実表 (base table)	表定義によって定義される名前付きの表
導出表 (derived table)	問い合わせ指定の評価によって、一つ以上の表から直接または間接に導出される表
ビュー表 (viewed table)	ビュー定義によって定義される名前付きの導出表
グループ表 (grouped table)	GROUP BY 句の評価中に導出されるグループの集合
グループビュー (grouped view)	グループ表から導出されるビュー表

(3) 結果

結果 (result) は、問い合わせ式、問い合わせ指定および SELECT 文によって導出される表のデータとする。

導出される表のデータは、空であってもよい。

(4) ナル値

ナル値 (null value) は、データが存在しない未定の値のこととする。

9.3.2 全般規則 (データベースアクセス機能)

SQL の規則と COBOL 言語の規則の異なる部分について説明する。

1. 文字集合で次に示す特殊文字は使用できない。

引用符 ("), 通貨記号 (¥), セミコロン (;), アンパサンド (&)

2. 識別子

識別子は、利用者定義語の規則に従い定義する。また、DBMS によって規定される場合もある。

識別子には、次に示す種類がある。

識別子	内容
認可識別子	表名およびプロシージャ名を修飾する DBMS で規定された名前とする。認可識別子が暗であっても陽であっても、二つの表名が等しいということは、それらの表名と認可識別子は同じであることとする。
表名	表名とは、表に付けられた名前とする。
列名	列名とは、表の列に付けられた名前とする。
相関名	相関名とは、SELECT 文、副問い合わせ指定、または問い合わせ指定で使用され、その指定された FROM 句の有効範囲で表名の別名として付けられた名前とする。
カーソル名	カーソル名とは、カーソルの名前とする。
プロシージャ名	プロシージャ名とは、ストアードプロシージャに付けられた名前とする。
SQL 文名	動的に準備する SQL 文に付けられた名前とする。
コネクション名	確立されたコネクションに付けられた名前とする。

表名およびプロシージャ名は、次の形式で修飾できる。

$$[\text{認可識別子} .] \left\{ \begin{array}{l} \text{表名} \\ \text{プロシージャ名} \end{array} \right\}$$

識別子を構成する文字は、31 字を超えてはならない。

3. 予約語は、COBOL の予約語に加え次に示す語が、SQL では予約語とする。

斜体の語は、COBOL の予約語である。

頭文字	予約語
A	<i>ALL, AND, ANY, AS, ASC, AUTHORIZATION, AVG, ABSOLUTE, ADA, ADD, ALLOCATE, ALTER, ARE, ASSERTION, AT</i>
B	BEGIN, BETWEEN, <i>BY</i> , BIGINT, <i>BINARY, BIT</i> , BIT_LENGTH
C	CHAR, <i>CHARACTER</i> , CHECK, <i>CLOSE, COBOL, COMMIT, CONTINUE, COUNT</i> , CREATE, <i>CURRENT, CURSOR, CALL</i> , CASCADE, CASCADED, CASE, CAST, CATALOG, CHAR_LENGTH, CHARACTER_LENGTH, COALESCE, COLLATE, COLLATION, <i>COLUMN, CONNECT</i> , CONNECTION, CONSTRAINT, CONSTRAINTS, CONVERT, <i>CORRESPONDING</i> , CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP
D	DEC, DECIMAL, DECLARE, <i>DEFAULT, DELETE</i> , DESC, DISTINCT, <i>DOUBLE, DATA, DATA_TYPE, DATE, DAY</i> , DEALLOCATE, DEFERRABLE, DEFERRED, DESCRIBE, DESCRIPTOR, DIAGNOSTICS, DICTIONARY, <i>DISCONNECT</i> , DISPLACEMENT, DOMAIN, DROP
E	<i>END</i> , ESCAPE, <i>EXEC</i> , EXISTS, <i>ELSE, END-EXEC</i> , EXCEPT, <i>EXCEPTION</i> , EXECUTE, <i>EXTERNAL</i> , EXTRACT
F	<i>FETCH</i> , FLOAT, <i>FOR</i> , FOREIGN, FORTRAN, FOUND, <i>FROM, FALSE, FIRST, FULL</i>
G	<i>GO</i> , GOTO, GRANT, <i>GROUP, GET, GLOBAL</i>
H	HAVING, HOLDLOCK, HOUR
I	<i>IN</i> , INDICATOR, <i>INSERT</i> , INT, INTEGER, <i>INTO, IS</i> , IDENTIFIED, IDENTITY, IGNORE, IMMEDIATE, <i>INCLUDE, INDEX</i> , INITIALLY, INNER, <i>INPUT</i> , INSENSITIVE, INTERSECT, <i>INTERVAL</i> , ISOLATION
J	JOIN
K	<i>KEY</i>
L	LANGUAGE, LIKE, <i>LAST, LEFT, LENGTH</i> , LEVEL, LOCAL, LONG, LOWER
M	MAX, MIN, MODULE, MATCH, MINUTE, MONTH, MUMPS
N	<i>NATIONAL</i> , NCHAR, <i>NOT, NULL, NUMERIC, NAME</i> , NAMES, <i>NEXT</i> , NONE, NULLABLE, NULLIF
O	<i>OF, ON, OPEN</i> , OPTION, <i>OR, ORDER</i> , OCTET_LENGTH, <i>OFF, ONLY</i> , OUTER, <i>OUTPUT, OVERLAPS</i>
P	PASCAL, PLI, PRECISION, PRIMARY, PRIVILEGES, <i>PROCEDURE</i> , PUBLIC, PAPAMTER_TYPE, PARTIAL, <i>PERFORM, POSITION</i> , PREPARE, PRESERVE, PRIOR
R	REAL, <i>REFERENCES, ROLLBACK</i> , RESTRICT, REVOKE, <i>RIGHT</i> , ROWS
S	SCHEMA, <i>SECTION, SELECT, SET</i> , SMALLINT, SOME, <i>SQL, SQLCODE, SQLERROR, SUM, SCALE, SCROLL, SECOND, SEQUENCE, SIZE, SQLCA, SQLSTATE, SQLWARNING, STOP, SUBSTRING, SYSTEM</i>
T	<i>TABLE, TO, TEMPORARY, THEN, TIME</i> , TIMESTAMP, TIMEZONE_HOUR, TIMEZONE_MINUTE, TINYINT, <i>TRANSACTION</i> , TRANSLATE, TRANSLATION, <i>TRUE, TYPE</i>

頭文字	予約語
U	UNION, UNIQUE, <i>UPDATE</i> , USER, UNKNOWN, UNSIGNED, UPPER, <i>USAGE</i> , <i>USING</i>
V	<i>VALUES</i> , VIEW, <i>VALUE</i> , VARBINARY, VARCHAR, <i>VARYING</i>
W	WHENEVER, <i>WHERE</i> , <i>WITH</i> , WORK, <i>WHEN</i>
Y	YEAR

4. 定数は、英数字定数、数字定数および浮動小数点数字定数とする。

- (a) 英数字定数の長さは、1～160 字とする。
- (b) 連結式を書くことはできない。
- (c) 数字定数のけた数は、1～18 けたとする。
- (d) 仮数のけた数は、1～16 けた、指数のけた数は、1～2 けたとする。

注意事項

SQL の英数字定数はアポストロフィ (') でしか囲むことができないため、アポストロフィ (') を英数字定数に含むときはアポストロフィ (') を 2 回続けなければならない。

(例)

A'B'C という 5 文字の英数字定数は次のように指定する。
'A"B"C'

5. 行内注記

行内注記は、注釈導入子で始まり、行の最後の文字で終わる。注釈導入子とは、分離符の空白に続く二つ以上の連続するハイフン (-) の並びとする。

この行内注記は、SQL 文中の分離符の空白が許される個所であればどこに現われてもよい。

6. 分離符の句読文字のコンマ (,) は、構文で規定された個所以外に用いることはできない。

(a) 非区切りトークンの後ろにハイフン (-) および注釈導入子を続けてはならない。また、ハイフンの後ろに非区切りトークンを続けてはならない。

7. 算術演算子では、二項演算子のべき乗 (**) を使用できない。

8. 比較演算子を次に示す。

比較演算子	意味
=	等しい
<>	等しくない
<	より小さい
>	より大きい
<=	より小さいか、または等しい
>=	より大きいか、または等しい

9. 浮動指示子

COBOL の場合と同様に、埋め込み SQL 文中に次に示す浮動指示子を使用できる。ただし、- SQL,ODBC,NoCont オプションを指定した場合には、定数継続指示子は無効にし、定数の右閉じアポストロフィとハイフンとして処理する。

指示子	意味
*>	注記指示子
>>	翻訳指令指示子
'-	定数継続指示子（"- は指示子にならない）
>>D	デバッグ指示子

10. 修飾

修飾語と被修飾語をピリオド (.) で連結して修飾する場合、被修飾語が次の形式のときは、直前のピリオド (.) との間に空白を置かなければならない。

[数字] E [-] 数字 [- [利用者定義語構成文字]] または 数字 - 利用者定義語構成文字

なお、修飾の形式については、表名およびプロシージャ名は「2. 識別子」、埋め込み変数は「9.4.2 埋め込み変数 (データベースアクセス機能)」、列名は「9.5.2 列指定 (データベースアクセス機能)」をそれぞれ参照のこと。

9.4 データ部 (埋め込み SQL 宣言節) (データベースアクセス機能)

データ部では、SQLCODE 変数、埋め込み変数および標識変数を定義するデータ記述項を埋め込み SQL 宣言節に記述する。

形式

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.  
[ SQLCODE変数定義 ]  
[ 埋め込み変数定義 ] …  
[ 標識変数定義 ] …  
EXEC SQL END DECLARE SECTION END-EXEC.
```

機能

手続き部で使用する SQLCODE 変数、埋め込み変数および標識変数を定義する。

構文規則

- 埋め込み SQL 宣言節は、埋め込み SQL 開始宣言で始まり、埋め込み SQL 終了宣言で終わる。
- 埋め込み SQL 宣言節は、ファイル節、作業場所節、局所場所節および連絡節に指定できる。
- 埋め込み SQL 宣言節は、データ部の異なる節にわたってはならない。すなわち、一つの節中で完結していなければならない。
- 埋め込み SQL 開始宣言は、01 または 77 レベルの直前でなければならない。
- 埋め込み SQL 終了宣言は、77 レベルの直後またはレコード記述の最後の基本項目の直後でなければならない。
- SQLCODE 変数定義、埋め込み変数定義および標識変数定義は、指定したファイル節、作業場所節、局所場所節、または連絡節で有効なデータ記述項でなければならない。

一般規則

埋め込み SQL 宣言節に記述するデータ項目には、次に示す項目を記述してもよいが、特に指定がないかぎり手続き部の埋め込み SQL 文では使用できない。

- REDEFINES 句を指定した再定義項目
- OCCURS 句を指定した表要素
- 集団項目
- 外部浮動小数点数字項目
- 数字編集項目
- 英数字編集項目
- 日本語編集項目
- 外部ブール項目
- 内部ブール項目
- 英字項目
- 指標データ項目

- 指標名
- アドレスデータ項目
- アドレス名
- 66 レベルの再命名定義
- 78 レベルの定数名
- 88 レベルの条件名
- FILLER 項目
- VARIANT データ項目
- OLE オブジェクト参照データ項目
- オブジェクト参照データ項目
- [SIGN IS] LEADING SEPARATE CHARACTER の指定がない外部 10 進形式の数字項目
- 日本語項目

9.4.1 SQLCODE 変数 (データベースアクセス機能)

機能

埋め込み SQL 文の実行が正常に終了したかどうかを示す状態コードが設定される。

構文規則

SQLCODE 変数は、レベル番号が 01 か、または 77 でデータ名が SQLCODE の 9 けたの符号付き 2 進形式でなければならない。

(例)

```
77 SQLCODE PIC S9(9) USAGE BINARY.
```

一般規則

SQLCODE 変数には、宣言系でない埋め込み SQL 文の実行結果によって次の値が設定される。

SQLCODE 変数に設定される値	意味
0	実行は成功した。
100	該当行が存在しない。
負数 (<0)	実行は失敗し、エラーメッセージを出力した。
上記以外	実行は成功したが警告メッセージを出力した。

9.4.2 埋め込み変数（データベースアクセス機能）

機能

手続き部の埋め込み SQL 文中に指定する変数によってデータベースに値を渡す、またはデータベースから値を取得する。

構文規則

埋め込み変数定義に指定できるデータ記述項は次のどれかでなければならない。ただし、集団項目は下位項目のデータ構造が次のどれかでなければならない。

データ項目	条件	
英数字項目	なし	
数字項目	外部 10 進形式	[SIGN IS] LEADING SEPARATE CHARACTERの指定があること
	2 進形式	項目が占めるバイト数が 1 バイト, 2 バイトおよび 4 バイトであり, 小数点指定がないこと
	内部浮動小数点形式 (4 バイト)	なし
	内部浮動小数点形式 (8 バイト)	なし
集団項目	可変長型形式	L2 xxxx PIC S9(9) USAGE COMP. L2 xxxx PIC X(n).
	ビット型形式	L2 xxxx PIC 1(7) USAGE BIT. FILLER PIC 1(1) USAGE BIT.
	日付型形式	L2 xxxx PIC S9(4) USAGE COMP. L2 xxxx PIC 9(4) USAGE COMP. L2 xxxx PIC 9(4) USAGE COMP.
	時刻型形式	L2 xxxx PIC 9(4) USAGE COMP. L2 xxxx PIC 9(4) USAGE COMP. L2 xxxx PIC 9(4) USAGE COMP.
	日付/時刻型形式	L2 xxxx PIC S9(4) USAGE COMP. L2 xxxx PIC 9(4) USAGE COMP. L2 xxxx PIC 9(9) USAGE COMP.

(凡例)

L2：下位項目のレベル番号

xxxx：任意のデータ名

n：データ長

このシステムでは、COBOL の規則に従っていれば、ほかの句を指定してもよい。

注 1 JUSTIFIED 句の機能は SQL 文では無効となる。

注 2 BLANK WHEN ZERO 句を指定した項目は、SQL 文に指定できない。

注3 OCCURS 句を指定してもよいが、表要素は SQL 文に指定できない。

一般規則

1. 埋め込み変数は、指定する埋め込み SQL 文で対応する列、式、またはパラメタのデータ型に対し、有効な埋め込み変数の型でなければならない。
2. 埋め込み変数の型とデータ項目の対応を次に示す。

埋め込み変数の型		データ項目	-SQL,ODBC オプション指定
文字列型	固定長形式	英数字項目	○
	可変長形式	集団項目 可変長型形式	○
数値型		数字項目 外部 10 進形式	○
		集団項目 数値型形式	×
整数型	符号付き 1 バイト 2 進形式	1 バイト英数字項目	○
		数字項目 2 進形式 (1 バイト, 符号付き)	○
	符号なし 1 バイト 2 進形式	数字項目 2 進形式 (1 バイト, 符号なし)	○
	符号付き 2 バイト 2 進形式	数字項目 2 進形式 (2 バイト, 符号付き)	○
	符号なし 2 バイト 2 進形式	数字項目 2 進形式 (2 バイト, 符号なし)	○
	符号付き 4 バイト 2 進形式	数字項目 2 進形式 (4 バイト, 符号付き)	○
	符号なし 4 バイト 2 進形式	数字項目 2 進形式 (4 バイト, 符号なし)	○
	符号付き 8 バイト 2 進形式	20 バイトの英数字項目	○
	符号なし 8 バイト 2 進形式	20 バイトの英数字項目	○
浮動 小数点型	単精度形式	数字項目 内部浮動小数点形式 (4 バイト)	○
	倍精度形式	数字項目 内部浮動小数点形式 (8 バイト)	○
バイナリ データ型	固定長形式	英数字項目	○
	可変長形式	集団項目 可変長型形式	○
	可変長形式	集団項目 バイナリデータ型形式	×

埋め込み変数の型		データ項目	-SQL,ODBC オプション指定
ビット列型		集団項目 ビット型形式	○
日時型	日付形式	集団項目 日付型形式	○
	時刻形式	集団項目 時刻型形式	○
	日付/時刻形式	集団項目 日付/時刻型形式	○
		集団項目 Sign メンバ付日付時間型形式	×

(凡例)

○：使用できる

×：使用できない

3. 埋め込み変数を手続き部の埋め込み SQL 文に指定する場合、次の形式で指定する。

:[データ名 1 .] … データ名 2

このデータ名を添字付けおよび部分参照してはならない。ただし、修飾はピリオド (.) でデータ名 1 を連結することでデータ名 2 を修飾できる。例えば、:AAA.BBB と記述した場合、COBOL の言語規則の BBB OF AAA と表されたものとみなされる。

9.4.3 標識変数 (データベースアクセス機能)

機能

対応する埋め込み変数の値がナル値かどうかを示す。

構文規則

標識変数定義は、4 バイトの符号付き 2 進形式でなければならない。

(例) 次のように定義する。

```
77 INDVAR001 PIC S9(9) USAGE BINARY.
```

同じ名前の埋め込み変数、標識変数を複数回定義してもよい。

プログラムが入れ子構造になっている場合、埋め込み SQL 文中の埋め込み変数、標識変数の参照は、COBOL として有効な参照でなければならない。

一般規則

1. INSERT 文または UPDATE 文の実行によって、挿入値または更新値にナル値を指定したい場合は、標識変数に-1 を設定する。これによって、指定された値はナル値とみなされる。
2. SELECT または FETCH 文の実行によって取り出した結果の値がナル値の場合、標識変数には-1 が設定される。
3. 標識変数を手続き部の埋め込み SQL 文に指定する場合、次の形式で指定する。

:[データ名 1 .] … データ名 2

このデータ名を添字付けおよび部分参照してはならない。ただし、修飾はピリオド (.) でデータ名 1 を連結することでデータ名 2 を修飾できる。例えば、:AAA.BBB と記述した場合、COBOL の言語規則の BBB OF AAA と表されたとみなされる。

9.5 手続き部 (共通要素) (データベースアクセス機能)

9.5.1 値指定と相手指定 (データベースアクセス機能)

形式

値指定

$$\left\{ \begin{array}{l} \text{埋め込み変数 [INDICATOR 標識変数]} \\ \text{定数} \\ \text{USER} \end{array} \right\}$$

相手指定

埋め込み変数 [INDICATOR 標識変数]

機能

値指定 (value specification) と相手指定 (target specification) は、一つ以上の値または変数を指定する。

構文規則

埋め込み変数および標識変数は埋め込み SQL 宣言節で定義していなければならない。

一般規則

1. 値指定は、表から選択されない値を指定する。
2. 相手指定は、値が代入される埋め込み変数を指定する。または、埋め込み変数の値を渡したい場合に指定する。
3. USER は、接続データベースの規定による値を持つ文字列とする。
4. 埋め込み変数、ならびに標識変数の定義および値については「9.4 データ部 (埋め込み SQL 宣言節) (データベースアクセス機能)」を参照。

9.5.2 列指定 (データベースアクセス機能)

形式

$$\left[\left\{ \begin{array}{l} \text{表名} \\ \text{相関名} \end{array} \right\} \right] \text{列名}$$

機能

列指定 (column specification) は、修飾できる列名を指定する。

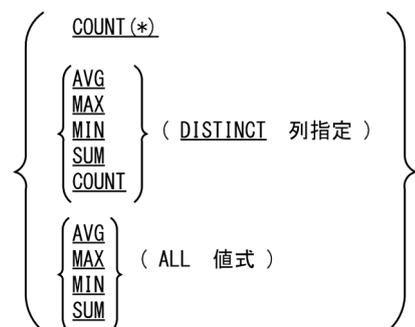
一般規則

1. 列指定のデータは、列名で識別される表の列のデータとする。

2. 列名を表名または相関名で修飾する場合、修飾する表名および相関名は、その文中の有効範囲内であればならない。

9.5.3 集合関数指定 (データベースアクセス機能)

形式



機能

集合関数指定 (set function specification) は、関数の適用によって導出される値を指定する。

一般規則

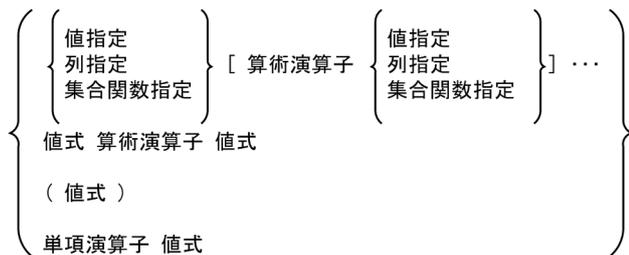
1. DISTINCT 指定の場合の引数の値は、列指定のすべての値からナル値および重複する値を取り除いた値の集合とする。
2. ALL 指定の場合、引数で指定された値式のナル値を含まない値の集合とする。
3. 各関数の機能を次に示す。

関数	機能
COUNT(*)	該当するすべての行の数を返す。
AVG	DISTINCT, ALL 指定の引数で指定された列指定, 値式の平均値を返す。
MAX	DISTINCT, ALL 指定の引数で指定された列指定, 値式の最大値を返す。
MIN	DISTINCT, ALL 指定の引数で指定された列指定, 値式の最小値を返す。
SUM	DISTINCT, ALL 指定の引数で指定された列指定, 値式の合計値を返す。
COUNT	DISTINCT 指定の引数で指定された列指定の値の数を返す。

4. 各関数の引数および結果のデータ型は、接続データベースによって規定される。

9.5.4 値式 (データベースアクセス機能)

形式



機能

値式 (value expression) は、値を指定する。

一般規則

1. DISTINCT 指定の集合関数を含む値式は、どの二項演算子も含んではならない。
2. 値式中の値指定、列指定および集合関数指定のデータ型が文字列の場合、値式に算術演算子を含めてはならない。
3. 値式にナル値が含まれる場合、その値式の結果はナル値とする。
4. 除数は、0 であってはならない。
5. 括弧を使って、値式中の評価順序を指定できる。括弧の中の値式が先に評価される。括弧がない場合、単項演算子、乗除算、加減算の順に左から右へ評価される。
6. 値式の結果のデータ型は、接続データベースによって規定される。

9.5.5 述語 (データベースアクセス機能)

機能

述語 (predicate) は、真、偽、または不定の真偽値を与える評価ができる条件を指定する。

構文規則

述語は、次のどれかとする。

- 比較述語
- BETWEEN 述語
- IN 述語
- LIKE 述語
- NULL 述語
- 限定述語
- EXISTS 述語

一般規則

述語の結果は、それを表の与えられた行に適用することによって導出される。

9.5.6 比較述語 (データベースアクセス機能)

形式

$$\text{値式 比較演算子 } \left\{ \begin{array}{l} \text{値式} \\ \text{副問い合わせ} \end{array} \right\}$$

機能

比較述語 (comparison predicate) は、二つの値を比較する。

一般規則

1. 比較演算子の左辺の値式のデータ型と、右辺の値式または副問い合わせのデータ型は、互いに比較できるデータ型でなければならない。
2. 副問い合わせの結果は、一つの値でなければならない。
3. 比較演算子の左辺の値式または右辺の値式がナル値の場合、もしくは右辺の副問い合わせの結果が空の場合、比較述語の結果は不定となる。値式とナル値を比較するときは、NULL 述語を使用すること。

9.5.7 BETWEEN 述語 (データベースアクセス機能)

形式

値式 [NOT] BETWEEN 値式 AND 値式

機能

BETWEEN 述語 (between predicate) は、範囲比較をする。

一般規則

1. BETWEEN 左辺の値式のデータ型と、右辺の値式および AND 値式のデータ型は、互いに比較できるデータ型でなければならない。
2. 次に示す例は、左右どちらも同じ結果となる。

x BETWEEN y AND z	x >= y AND x <= z
x NOT BETWEEN y AND z	NOT (x BETWEEN y AND z)

9.5.8 IN 述語 (データベースアクセス機能)

形式

$$\text{値式 [NOI] IN } \left\{ \begin{array}{l} \text{副問い合わせ} \\ \text{値指定 { , 値指定 } \dots} \end{array} \right\}$$

機能

IN 述語 (in predicate) は、値の集合にその値が含まれるかを比較する。

一般規則

1. IN 左辺の値式のデータ型と、右辺のすべての値指定または副問い合わせのデータ型は、互いに比較できるデータ型でなければならない。
2. 次に示す例は、左右どちらも同じ結果となる。

x IN (SELECT a FROM A)	x = ANY (SELECT a FROM A)
x NOT IN (SELECT a FROM A)	NOT (x IN (SELECT a FROM A))

9.5.9 LIKE 述語 (データベースアクセス機能)

形式

列指定 [NOI] LIKE 値指定 [ESCAPE 値指定]

機能

LIKE 述語 (like predicate) は、パターン照合比較する。

一般規則

1. 列指定のデータ型、LIKE 右辺の値指定および ESCAPE 指定の値指定のデータ型は、文字列でなければならない。
2. LIKE 左辺の列指定、または右辺の値指定がナル値の場合、結果は不定となる。
3. LIKE 右辺の値指定には、パターン照合比較するための文字列パターンを指定する。文字列パターンでは、下線 (_) は、任意文字指示子を表し、パーセント (%) は、任意文字列指示子を表す。下線 (_) でも、パーセント (%) でもない文字は、それ自身を表す。次にその例を示す。

列指定のデータ	
'A'	…1
'ABC'	…2
'ABCD'	…3
'BCD'	…4

文字列パターン	真となるデータ
'ABC'	2
'%A%'	1, 2, 3
'_BC%'	2, 3
'_C_'	4

4. ESCAPE 指定の値指定は、文字列パターンでの下線 (_), パーセント (%) をそれ自身を表す文字と認識するための 1 文字のエスケープ文字を指定する。例えば、'%A%' という文字列で始まるデータを持つ列 x をパターン照合比較する場合は、次のように指定すれば真となる。

```
x LIKE '¥%A%' ESCAPE '¥'
```

注意事項

文字列パターンを文字列型固定長形式の埋め込み変数で指定する場合、変数の長さより短い文字列パターンを設定すると、右側の空白を含めて文字列パターンとなるため注意が必要である。

例えば、'ABCD' というデータを持つ列 x をパターン照合比較した場合、次のように文字列パターン 'AB%' では結果が真とならない。

```
01 STR001 PIC X(4).
:
MOVE 'AB%' TO STR001.
EXEC SQL
  SELECT .... x LIKE :STR001 .... 列xのデータ'ABCD'との
                                     比較結果は偽

:
MOVE 'AB%' TO STR001.
EXEC SQL
  SELECT .... x LIKE :STR001 .... 列xのデータ'ABCD'との
                                     比較結果は真
```

9.5.10 NULL 述語 (データベースアクセス機能)

形式

```
列指定 IS [ NOT ] NULL
```

機能

NULL 述語 (null predicate) は、ナル値比較する。

一般規則

列指定が、ナル値の列 x の場合、x IS NULL の結果は真となる。

9.5.11 限定述語（データベースアクセス機能）

形式

値式 比較演算子 $\left\{ \begin{array}{l} \text{ALL} \\ \text{SOME} \\ \text{ANY} \end{array} \right\}$ 副問い合わせ

機能

限定述語（quantified predicate）は、限定された値の集合を比較する。

一般規則

1. 比較演算子左辺の値式のデータ型と、右辺の副問い合わせのデータ型は、互いに比較できるデータ型でなければならない。
2. ALL を指定した場合、副問い合わせの結果が空か、または副問い合わせの結果のすべての値と値式が、比較演算子で真となるならば限定述語は真となり、一つでも比較演算子で偽となるならば限定述語は偽となる。
3. SOME と ANY は同義語とし、副問い合わせの結果の値と値式が、一つでも比較演算子で真となるならば限定述語は真となり、副問い合わせの結果が空か、または副問い合わせの結果のすべての値と値式が比較演算子で偽となるならば限定述語は偽となる。
4. 限定述語が真でも偽でもない場合は、不定となる。

9.5.12 EXISTS 述語（データベースアクセス機能）

形式

EXISTS 副問い合わせ

機能

EXISTS 述語（exists predicate）は、空集合比較する。

一般規則

副問い合わせの結果が空でない場合、EXISTS 述語は真となる。結果が空の場合は偽となる。

9.5.13 探索条件（データベースアクセス機能）

形式

[NOI] $\left\{ \begin{array}{l} \text{述語} \\ \text{(探索条件)} \end{array} \right\}$ [$\left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\}$ 探索条件]

機能

探索条件 (search condition) は、指定される条件に論理演算子を適用した結果によって真、偽、または不定の真偽値を与える。

一般規則

1. NOT を指定した述語または (探索条件) の結果による、探索条件の結果を次に示す。

述語または (探索条件) の結果	NOT 指定の探索条件の結果
真	偽
偽	真
不定	不定

2. AND, または OR を指定した場合の右辺と左辺の結果による、探索条件の結果を次に示す。

右辺	左辺	AND 指定の探索条件の結果	OR 指定の探索条件の結果
真	真	真	真
真	偽	偽	真
真	不定	不定	真
偽	真	偽	真
偽	偽	偽	偽
偽	不定	偽	不定
不定	真	不定	真
不定	偽	偽	不定
不定	不定	不定	不定

3. 括弧の中が最初に評価され、NOT, AND, OR の順に左から右に評価される。

9.5.14 表式 (データベースアクセス機能)

形式

```
FROM 表名 [ AS 相関名 ] [ 接続データベース固有指示子 ]  
[ , 表名 [ AS 相関名 ] [ 接続データベース固有指示子 ] ] ...  
[ WHERE 探索条件 ]  
[ GROUP BY 列指定 [ , 列指定 ] ... ]  
[ HAVING 探索条件 ]
```

機能

表式 (table expression) は、表またはグループ表を指定する。

構文規則

表式での接続データベース固有指示子は、HOLDLOCK とする。

一般規則

1. FROM 句で識別される表がグループビューならば、表式に WHERE 句、GROUP BY 句、または HAVING 句のどれも指定してはならない。
2. FROM 句は、一つ以上の表から導出される表とする。
 - (a) 表名、相関名の有効範囲は、FROM 句が含まれる表式を含む最も内側の副問い合わせ、問い合わせ指定、または SELECT 文とする。
 - (b) 接続データベース固有指示子とは、表名の後に指定するオプションのことである。データベースによっては表名の後に、ある特定のオプションを指定できる。例えば、Microsoft の SQL Server では、ロック手法などのオプションを指定できる。
 - (c) 接続データベース固有指示子の構文、機能およびその動作については、現行コネクションで接続しているデータベースに従う。
3. WHERE 句は、先行する FROM 句の結果に探索条件を適用することによって導出される表を規定する。
4. GROUP BY 句は、先行する FROM 句、または WHERE 句で導出される表の行を指定した列が同じ値を持つグループに分割したグループ表とする。
 - (a) 各列指定は、先行する FROM 句および WHERE 句で導出されるの表中の列を一意に参照しなければならない。
5. HAVING 句は、先行する FROM 句、WHERE 句、または GROUP BY 句の結果である複数のグループ表に対し、探索条件の結果が真となるようなグループのグループ表とする。

9.5.15 副問い合わせ (データベースアクセス機能)

形式

(SELECT { ALL } { 値式 }
 { DISTINCT } { * } 表式)

機能

副問い合わせ (subquery) は、表式の結果から導出される値のマルチ集合とする。

一般規則

1. 副問い合わせのデータ型は、値式のデータ型とする。星印 (*) を指定した場合、表式の結果から導出される表の列のデータ型とする。
2. DISTINCT を指定しなかった場合、表式から導出される表の値式の値のマルチ集合とする。
3. DISTINCT を指定した場合、表式から導出される表の値式の値から重複値を取り除いた値のマルチ集合とする。
4. 星印 (*) を指定した場合、表式の結果から導出される表の列は、単独の列を参照しなければならない。ただし、EXISTS 述語に指定される副問い合わせならば、表式の結果から導出される表の列は、単独でない列を参照していてもよい。

5. 値式中に指定する列指定は、表式の結果から導出される表の列を一意に参照する列名でなければならない。

9.5.16 問い合わせ指定 (データベースアクセス機能)

形式

$$\text{SELECT } \left\{ \begin{array}{l} \text{ALL} \\ \text{DISTINCT} \end{array} \right\} \left\{ \begin{array}{l} \text{値式 [, 値式] } \dots \\ * \end{array} \right\} \text{表式}$$

機能

問い合わせ指定 (query specification) は、表式の結果から導出される表とする。

一般規則

1. DISTINCT を指定しなかった場合、表式の結果から導出される表とする。
2. DISTINCT を指定した場合、表式の結果から導出される表の重複行を取り除いた行の表とする。
3. 星印 (*) を指定した場合、表式の結果から導出される表の列がすべて指定されたものとする。
4. 値式中に指定する列指定は、表式の結果から導出される表の列を一意に参照する列名でなければならない。

9.5.17 問い合わせ式 (データベースアクセス機能)

形式

$$\left\{ \begin{array}{l} \text{問い合わせ指定} \\ \text{(問い合わせ式)} \\ \text{問い合わせ式 UNION [ALL] } \left\{ \begin{array}{l} \text{問い合わせ指定} \\ \text{(問い合わせ式)} \end{array} \right\} \end{array} \right\}$$

機能

問い合わせ式 (query expression) は、複数の問い合わせ指定から導出される表の行のマルチ集合の表とする。

一般規則

1. UNION を指定しなかった場合、問い合わせ指定の結果から導出される表とする。
2. UNION を指定した場合、左辺の問い合わせ式と右辺の問い合わせ指定または問い合わせ式から導出される表の行のマルチ集合の表とする。左辺と右辺で導出される表の列は、左辺と右辺、同じのデータ型の並びでなければならない。またその表の列の数は異なっていてはならない。
3. ALL を指定しなかった場合、重複するすべての行を取り除いた行の表とする。

9.6 手続き部 (宣言系) (データベースアクセス機能)

手続き部 (宣言系) では、手続き部に宣言系とする埋め込み SQL 文で記述できる SQL 文について記述する。

宣言系とする SQL 文には、埋め込み例外を宣言する WHENEVER 文とカーソルを宣言する DECLARE CURSOR 文がある。

9.6.1 WHENEVER (埋め込み例外宣言) 文 (データベースアクセス機能)

形式

$$\text{WHENEVER} \left\{ \begin{array}{l} \text{SQLERROR} \\ \text{SQLWARNING} \\ \text{NOT FOUND} \end{array} \right\} \left\{ \begin{array}{l} \text{CONTINUE} \\ \text{STOP} \\ \left\{ \begin{array}{l} \text{GOTO} \\ \text{GO TO} \\ \text{PERFORM} \end{array} \right\} : \text{手続き名} \end{array} \right\}$$

機能

埋め込み例外宣言 (embedded exception declaration) は、SQL 文に例外が生じるときに取られる動作を規定する。

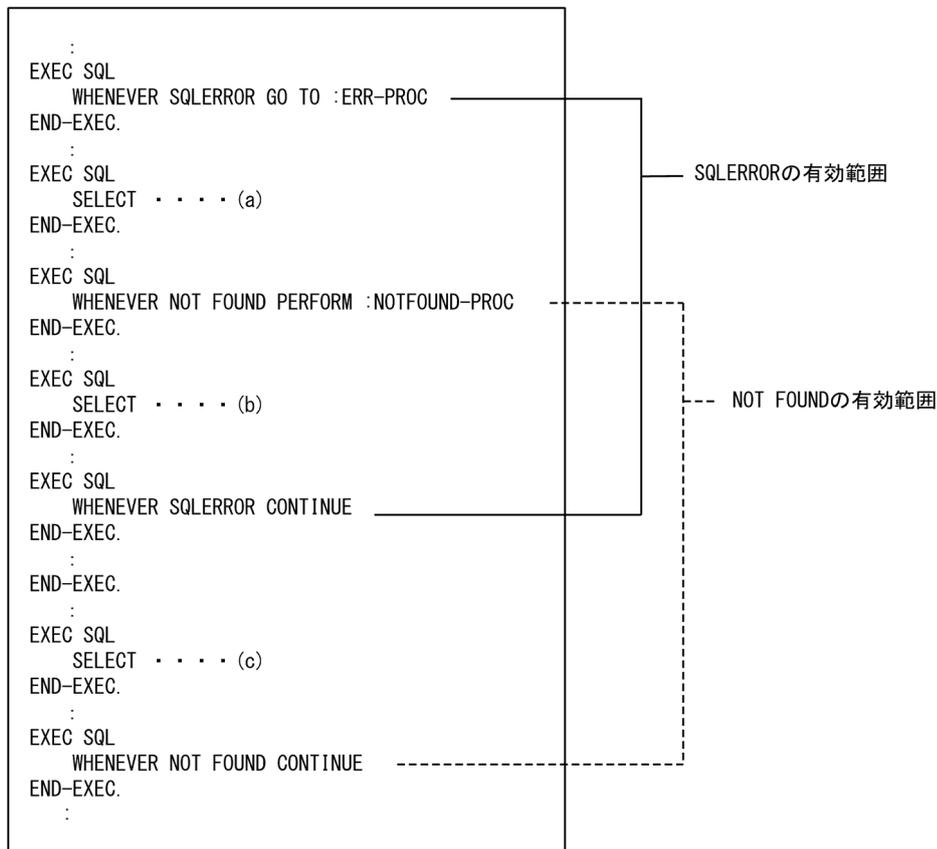
構文規則

- NOT FOUND 指定をした場合、STOP を指定してはならない。
- 手続き名に段落名を指定した場合、節名では修飾できない。
- 埋め込み例外宣言の有効範囲は、プログラム単位、関数単位およびメソッド単位とする。
- GOTO と GO TO は、同義語であり、どちらを書いてもよい。

一般規則

1. 埋め込み例外宣言は、プログラムの実行の流れには関係なく、原始プログラム上に記述された位置によって決まる。すなわち、ある一つの埋め込み例外宣言で指定された処理は原始プログラム上の出現順で次に現れた同じ指定の埋め込み例外宣言までにある、宣言系とする埋め込み SQL 文を除くすべての埋め込み SQL 文に対して有効となる。

次にその有効範囲の例を示す。



(a)の SELECT 文の場合

SQLERROR 条件のときだけ GO TO 文が実行される。

(b)の SELECT 文の場合

SQLERROR 条件のときは GO TO 文が実行され、NOT FOUND 条件のときは PERFORM 文が実行される。

(c)の SELECT 文の場合

NOT FOUND 条件のときだけ PERFORM 文が実行される。

2. 各指定によって、有効範囲の SQL 文実行で各動作が有効となる契機を次に示す。

指定	有効となる契機
SQLERROR	SQLCODE 変数に負の値が設定される契機
SQLWARNING	SQLCODE 変数に 100 でない正の値が設定される契機
NOT FOUND	SQLCODE 変数に 100 が設定される契機

SQLCODE 変数の詳細については「[9.4.1 SQLCODE 変数 \(データベースアクセス機能\)](#)」を参照。

3. 各指定が、有効となる条件を満たした場合の SQL 文実行後の動作を次に示す。

動作	意味
CONTINUE	各指定の有効範囲の終わりを指示し、以前の同じ指定の動作を以降の SQL 文の実行に適用しない。
STOP	COBOL の STOP RUN 文を実行することを指示する。

動作	意味
GOTO または GO TO	COBOL の GO TO 文を実行することを指示する。
PERFORM	COBOL の PERFORM 文を実行することを指示する。

9.6.2 DECLARE CURSOR (カーソル宣言) 文 (データベースアクセス機能)

形式

書き方 1 静的

```

DECLARE   カーソル名  CURSOR FOR

    問い合わせ式 [ ORDER BY   { 符号なし整数 } { ASC }
                    { 列指定   } { DESC }

                    [ , { 符号なし整数 } { ASC }
                      { 列指定   } { DESC } ] ... ]

```

書き方 2 動的

```

DECLARE   カーソル名  CURSOR FOR SQL文名

```

機能

カーソル宣言 (declare cursor) は、カーソルを定義する。

構文規則

- カーソル名は、プログラム単位、利用者定義関数単位およびメソッド単位で一意でなければならない。
- 同一のプログラム、利用者定義関数およびメソッド中に同じカーソル名を指定した OPEN 文がなければならない。
- 書き方 2 は、同一プログラム、利用者定義関数およびメソッド中に同じ SQL 文名を指定した PREPARE 文がなければならない。

注

カーソル名は、同一 SQL 埋め込みホストプログラム中のどのカーソル宣言で指定されるカーソル名とも同一であってはならない。プログラムが入れ子構造になっている場合、カーソル名は、プログラム単位で一意でなければならない。

一般規則

1. 書き方 1 は、問い合わせ式によって導出される表とカーソルを関連づける。
2. ORDER BY 句は、行の並べ替えを指示する。カーソルは、問い合わせ式によって導出される表の行を ORDER BY 句の列を基に並べ替えた表を関連づける。
3. ORDER BY 句の符号なし整数は、問い合わせ式によって導出される表の列の位置とする。
4. ORDER BY 句に符号なし整数、または列指定に DESC を指定すると並べ替えは降順となる。DESC を指定しない場合は、昇順となる。

5. 書き方 2 は、同じ SQL 文名を指定した PREPARE 文の動的 SQL とカーソルを関連づける。

9.7 手続き部 (制御系) (データベースアクセス機能)

手続き部 (制御系) では、制御系とする埋め込み SQL 文で記述できる SQL 文について記述する。

制御系とする SQL 文には、コネクション管理の CONNECT, DISCONNECT, SET CONNECTION 文とトランザクション管理の COMMIT, ROLLBACK 文がある。

9.7.1 コネクション (データベースアクセス機能)

SQL 文を使用して DBMS 上のデータへアクセスできる状態にすることをコネクションを確立するといひ、アクセスできる状態をアクセスできない状態にすることをコネクションを解除するという。

一つまたは複数のコネクションを確立でき、コネクションごとにトランザクションは管理される。

コネクションを確立する方法として、接続先データベース名を使用する方法と接続文字列を使用する方法がある。接続先データベース名、接続文字列の詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の ODBC インタフェース機能の説明を参照のこと。

9.7.2 トランザクション (データベースアクセス機能)

トランザクション (transaction) は、データベース操作を含む操作の並びとし、回復 (recovery) および並行性 (concurrency) の基本単位となる。

トランザクションは、SQL 文が実行されたとき、活性状態にあるトランザクションがなければ初期化され、COMMIT 文または ROLLBACK 文によって終了する。

このシステムでは、表に対する排他制御の定義は、DBMS の定義による。

(1) コミット

トランザクションをコミットすると、そのトランザクションで行われたデータベースに対するすべての変更は、すべての並行トランザクションからアクセスできるようになる。

(2) ロールバック

トランザクションをロールバックすると、そのトランザクションで行われたデータベースに対するすべての変更が取り消される。ただし、コミットされた変更を取り消すことはできない。

9.7.3 CONNECT 文 (データベースアクセス機能)

形式

書き方 1

```
CONNECT [ 埋め込み変数 1 IDENTIFIED BY 埋め込み変数 2 ] [ USING 埋め込み変数 4 ]  
        [ AS コネクション名 ]
```

書き方 2

```
CONNECT TO 埋め込み変数 5 [ AS コネクション名 ]
```

機能

CONNECT 文は、コネクションを確立する。

注

コネクション名は、同一プログラム中でだけ有効である。コネクション名をほかのプログラムにわたって使用することはできない。

構文規則

すべての埋め込み変数は、文字列型でなければならない。

一般規則

1. 埋め込み変数 1 には、接続するユーザ ID を設定しておかなければならない。
2. 埋め込み変数 2 には、接続するユーザ ID のパスワードを設定しておかなければならない。
3. 埋め込み変数 3 には、次の形式でユーザ ID とパスワードを設定しておかなければならない。
"ユーザ ID/パスワード"
4. 埋め込み変数 4 には、接続先データベース名を設定しておかなければならない。
5. 埋め込み変数 5 には、接続文字列を設定しておかなければならない。
6. 埋め込み変数 1~4 が次の場合、省略したものとする。埋め込み変数 5 は次の値であってはならない。
 - (a) 文字列型可変長形式で長さが 0 以下
 - (b) 空白文字列
7. 各埋め込み変数を省略した場合の仮定値を次に示す。

埋め込み変数	仮定値
埋め込み変数 1	ログインユーザ
埋め込み変数 2	パスワードなし
埋め込み変数 3	<ul style="list-style-type: none">• ユーザ ID を省略した場合、ログインユーザ名• パスワードを省略した場合、パスワードなし
埋め込み変数 4	"default"

8. すでに同じコネクション名を使用してコネクションを確立している場合、再度コネクションを確立することはできない。また、すでにコネクション名を指定しないでコネクションを確立している場合、再度コネクション名を指定しないでコネクションを確立することはできない。

9. CONNECT 文を実行しないで次に示す操作系の SQL 文を実行した場合、暗黙に埋め込み変数を省略したコネクション名を指定しない CONNECT 文を実行する。

小分類	SQL 文
静的	DELETE (探索), UPDATE (探索), INSERT, SELECT
カーソル	OPEN (静的)
動的	EXECUTE IMMEDIATE, PREPARE
ストアドプロシージャ	CALL

10. 確立したコネクションに対し、コネクション名で名前が付けられる。

11. コネクションが確立された場合、確立されたコネクションが現行コネクションとなり、以降の SQL 文は現行コネクションに対し実行される。

12. 埋め込み変数 1~5 に設定された値の後ろに半角空白がある場合、半角空白は削除される。

9.7.4 DISCONNECT 文 (データベースアクセス機能)

形式

$$\text{DISCONNECT } \left\{ \begin{array}{l} \text{コネクション名} \\ \text{DEFAULT} \\ \text{ALL} \\ \text{[CURRENT]} \end{array} \right\}$$

機能

DISCONNECT 文は、コネクションを解除する。

構文規則

1. 同一プログラム、利用者定義関数、メソッド中に同じコネクション名を使用した CONNECT 文がなければならない。
2. DISCONNECT 文に、コネクション名、DEFAULT、ALL、CURRENT のどの指定もない場合、CURRENT 指定が仮定される。

一般規則

1. コネクション名指定は、同じコネクション名で確立されたコネクションを解除する。
2. DEFAULT 指定は、コネクション名を指定しないで確立されたコネクションを解除する。
3. ALL 指定は、確立しているすべてのコネクションを解除する。
4. CURRENT 指定は、現行コネクションを解除する。
5. 現行コネクションを解除した後の接続先は、不定となる。

6. トランザクションは、おのこのコネクションに対して次のように終了する。
- (a) トランザクション中に SQLCODE 変数に負の値が設定される契機がなかった場合、コミットして終了する。
 - (b) トランザクション中に SQLCODE 変数に負の値が設定される契機があった場合、ロールバックして終了する。
7. DISCONNECT 文を実行しないでプログラム、利用者定義関数、メソッドの実行が終了した場合、実行の終了処理で暗黙に DISCONNECT ALL を実行する。

9.7.5 SET CONNECTION 文 (データベースアクセス機能)

形式

```
SET CONNECTION { コネクション名 }  
                { DEFAULT }
```

機能

SET CONNECTION 文は、現行コネクションを変更する。

構文規則

同一プログラム、利用者定義関数、メソッド中に同じコネクション名を使用した CONNECT 文がなければならぬ。

一般規則

1. コネクション名指定は、コネクション名で確立されたコネクションに現行コネクションを変更する。
2. DEFAULT 指定は、コネクション名を指定しないで確立されたコネクションに現行コネクションを変更する。
3. 変更するコネクションは、確立されていなければならない。

9.7.6 COMMIT 文 (データベースアクセス機能)

形式

```
COMMIT WORK
```

機能

COMMIT 文は、コミットでトランザクションを終了させる。

一般規則

1. トランザクション中に開かれたすべてのカーソルを閉じる。
2. トランザクション中に行われたデータベースへのすべての変更をコミットしてトランザクションを終了する。

9.7.7 ROLLBACK 文 (データベースアクセス機能)

形式

```
ROLLBACK WORK
```

機能

ROLLBACK 文は、ロールバックでトランザクションを終了させる。

一般規則

1. トランザクション中に開かれたすべてのカーソルを閉じる。
2. トランザクション中になされたデータベースへのすべての変更をロールバックしてトランザクションを終了する。

9.8 手続き部（操作系）（データベースアクセス機能）

9.8.1 手続き部（操作系）の概要（データベースアクセス機能）

手続き部（操作系）では、操作系とする埋め込み SQL 文で記述できる SQL 文について記述する。操作系とする SQL 文は、次に示す四つに分類できる。

分類	SQL 文	概要
静的	SELECT	表の指定された行から値を取り出す
	INSERT	表に新しい行を挿入する
	DELETE	表から行を削除する
	UPDATE	表の行を更新する
カーソル	OPEN	カーソルを開く
	FETCH	カーソルを表の次の行に位置づけ、その行を取り出す
	CLOSE	カーソルを閉じる
ストアードプロシージャ	CALL	ストアードプロシージャを呼び出す
動的	EXECUTE IMMEDIATE	動的に SQL 文を準備し、実行する
	PREPARE	動的に SQL 文を準備する
	DEALLOCATE PREPARE	動的に準備した SQL 文を解放する
	EXECUTE	動的に準備した SQL 文を実行する

(1) 静的に行う方法

静的に行う方法とは、埋め込み SQL 文で SQL 文を指定する方法である。このため実行時は、変数の値以外、その SQL 文自体の構文を変更できない。

埋め込み SQL 文中に埋め込み変数を指定することで変数を使用できる。

(2) カーソル

カーソル (cursor) は、DECLARE CURSOR (カーソル宣言) 文で宣言し、OPEN 文で開かれ、FETCH 文を複数回実行することで表から複数行の値を取り出すことができ、CLOSE 文で閉じられる。

また、カーソルを FETCH 文によって行に位置づけた場合、UPDATE 文、DELETE 文の位置づけの構文を使用することで行を更新、削除できる。

- このシステムでは、カーソル宣言が、カーソル宣言のカーソル名を参照する SQL 文よりも、原文中で後方であってもよい。プログラムが入れ子構造になっている場合、カーソル宣言は、カーソル宣言のカーソル名を参照する SQL 文と同一のプログラム中になければならない。

(3) ストアドプロシージャ

ストアドプロシージャ (stored procedure) は、データベースに格納された一連の手続きとする。ストアドプロシージャを呼び出すことによってその一連の手続きを実行する。ストアドプロシージャを使用すると、ユーザ宣言の変数、条件付き実行などのプログラミング機能をデータベース内で使用できる。

(4) 動的に行う方法

動的に行う方法とは、実行時に動的に SQL 文を構成する方法である。この実行時に構成する SQL 文を、以降では動的 SQL とする。動的 SQL は、接続データベースによって構文、機能およびその動作が規定される。

動的 SQL 中にパラメタマーカと呼ぶ疑問符 (?) を指定し、これに USING 句で埋め込み変数を対応づけることで変数を使用できる。

9.8.2 SELECT 文 (単一行) (データベースアクセス機能)

形式

```

SELECT  { ALL          } { 値式 [ , 値式 ] … }
        { DISTINCT   } { *                }
        INTO 相手指定 [ , 相手指定 ] …
        表式

```

機能

SELECT 文は、表の指定された行から値を取り出す。

一般規則

1. DISTINCT を指定しなかった場合、表式の結果から導出される表とする。
2. DISTINCT を指定した場合、表式の結果から導出される表の重複行を取り除いた行の表とする。
3. 星印 (*) を指定した場合、表式の結果から導出される表の列がすべて指定されたものとする。
4. 値式中に指定する列指定は、表式の結果から導出される表の列を一意に参照する列名でなければならない。
5. 相手指定の数と順序は、値式、または星印 (*) で参照される列の並びに対応する。
6. 相手指定のデータは、各列のデータ型で有効なデータ型でなければならない。
7. 表式によって導出される行が空の場合、SQLCODE 変数に 100 を設定する。

8. 表式によって導出される行が空でない場合、値式、または星印 (*) で参照される列のデータを相手指定に設定する。

9.8.3 INSERT 文 (データベースアクセス機能)

形式

```
INSERT INTO 表名 [ ( 列名 [ , 列名 ] … ) ]  
  
VALUES ( { 値指定 } [ , { 値指定 } ] … )  
      { 問い合わせ指定 }
```

機能

INSERT 文は、表に行を挿入する。

一般規則

1. 列名を省略した場合、表の列がすべて指定されたものとする。
2. 値指定の数と順序は、列名の並びに対応する。
3. 値指定のデータは、対応する各列のデータ型に有効な値のデータでなければならない。
4. VALUES 指定に、各値指定の値、または NULL が指定された場合はナル値で表に新しい行を挿入する。
5. 問い合わせ指定によって導出される表の列の数と順序は、列名の並びに対応する。
6. 問い合わせ指定によって導出される表の列のデータは、対応する各列のデータ型で有効な値のデータでなければならない。
7. 問い合わせ指定を指定した場合、問い合わせ指定によって導出される表のすべての行を表に挿入する。
8. 問い合わせ指定によって導出される表が空の場合、SQLCODE 変数に 100 を設定する。

9.8.4 DELETE 文 (データベースアクセス機能)

形式

書き方 1 探索

```
DELETE FROM 表名 [ WHERE 探索条件 ]
```

書き方 2 位置づけ

```
DELETE FROM 表名 WHERE CURRENT OF カースル名
```

機能

DELETE 文は、表から行を削除する。

構文規則

カーソル名は、同一プログラム、利用者定義関数、メソッド中に DECLARE CURSOR 文でカーソルが宣言されていなければならない。

一般規則

- 書き方 1 は、表から探索条件が真となるすべての行を削除する。
また、探索条件を指定しない場合は表のすべての行を削除する。
- 書き方 2 は、FETCH 文で位置づけられた行を削除する。
- 書き方 2 の表名は、カーソル宣言の FROM 句で指定した表でなければならない。
- 書き方 2 のカーソルは、行に位置づけられていなければならない。

9.8.5 UPDATE 文 (データベースアクセス機能)

形式

書き方 1 探索

UPDATE 表名 SET 列名 = $\left\{ \begin{array}{l} \text{値式} \\ \text{NULL} \end{array} \right\}$ [, 列名 = $\left\{ \begin{array}{l} \text{値式} \\ \text{NULL} \end{array} \right\}$] ...
[WHERE 探索条件]

書き方 2 位置づけ

UPDATE 表名 SET 列名 = $\left\{ \begin{array}{l} \text{値式} \\ \text{NULL} \end{array} \right\}$ [, 列名 = $\left\{ \begin{array}{l} \text{値式} \\ \text{NULL} \end{array} \right\}$] ...
WHERE CURRENT OF カーソル名

機能

UPDATE 文は、表の行を更新する。

構文規則

- カーソル名は、同一プログラム、利用者定義関数、メソッド中の DECLARE CURSOR 文で宣言されなければならない。
- 値式に集合関数指定を含んではならない。

一般規則

- 書き方 1 は、表から探索条件が真となるすべての行を更新する。また、探索条件を指定しない場合は表のすべての行を更新する。
- 書き方 2 は、FETCH 文で位置づけられた行を更新する。
- 書き方 2 の表名は、カーソル宣言の FROM 句で指定した表でなければならない。
- 書き方 2 のカーソルは、行に位置づけられていなければならない。
- 値式のデータは、対応する列のデータ型に有効な値のデータでなければならない。

6. 各値式のデータで行の各列のデータが更新される。ただし、NULL を指定した場合は、ナル値で更新される。

9.8.6 OPEN 文 (データベースアクセス機能)

形式

書き方 1 静的

```
OPEN カーソル名
```

書き方 2 動的

```
OPEN カーソル名 [ USING 相手指定 [ , 相手指定 ] … ]
```

機能

OPEN 文は、カーソルを開く。

構文規則

- カーソル名は、同一プログラム、利用者定義関数、メソッド中の DECLARE CURSOR 文で宣言されていなければならない。
- 書き方 2 のカーソル名は、同一プログラム、利用者定義関数、メソッド中に DECLARE CURSOR 文の書き方 2 でカーソルが宣言されていなければならない。

一般規則

- カーソルは、閉じられていなければならない。
- 書き方 2 は、動的 SQL をカーソルで実行する場合に使用する。
 - 動的 SQL は、準備しておかなければならない。
 - 動的 SQL は、DELETE、INSERT および UPDATE 文であってはならない。
 - USING の相手指定の数および順序は、パラメタマーカの並びに対応し、各相手指定のデータは、各パラメタマーカに対応する列、または式のデータ型に有効な値のデータでなければならない。なお、パラメタマーカに対応する列、または式のデータ型が固定長形式の型の場合、データの長さを合わせる必要がある。
- カーソルが開かれた場合、カーソルは、先頭行の前に位置づけられる。

9.8.7 FETCH 文 (データベースアクセス機能)

形式

静的／動的

```
FETCH カーソル名 INTO 相手指定 [ , 相手指定 ] …
```

機能

FETCH 文は、カーソルを表の次の行に位置づけ、その行の値を取り出す。

構文規則

カーソル名は、同一プログラム、利用者定義関数、メソッド中の DECLARE CURSOR 文で宣言されていなければならない。

一般規則

1. カーソルは、開かれていなければならない。
2. INTO の相手指定の数および順序は、取り出す行の列の並びに対応し、各相手指定の変数の型は、返される結果の値を格納できる変数の型でなければならない。
3. 取り出す行が存在する場合、行の各列の値が、対応する相手指定に設定される。
4. カーソル宣言によって導出される表が空の場合、またはカーソルが最終行もしくは最終行の次に位置づけられている場合は、カーソルを最終行の次に位置づけ、SQLCODE 変数に 100 が設定される。

9.8.8 CLOSE 文 (データベースアクセス機能)

形式

静的／動的

```
CLOSE カーソル名
```

機能

CLOSE 文は、カーソルを閉じる。

構文規則

カーソル名は、同一プログラム、利用者定義関数、メソッド中に DECLARE CURSOR 文でカーソルが宣言されていなければならない。

一般規則

カーソルは、開かれていなければならない。

9.8.9 CALL 文 (データベースアクセス機能)

形式

```
CALL プロシージャ名 [ ( 相手指定 [ , 相手指定 ] … ) ]
```

機能

CALL 文は、ストアードプロシージャを呼び出す。

一般規則

1. 指定した相手指定の数および順序は、ストアードプロシージャのパラメタの並びに対応する。
2. 入力、入出力パラメタと定義されたパラメタに対応する相手指定のデータは、対応するパラメタのデータ型に有効な値のデータでなければならない。
3. 出力、入出力パラメタと定義されたパラメタに対応する相手指定の埋め込み変数の型は、プロシージャで設定される値を格納できる埋め込み変数の型でなければならない。
4. ストアドプロシージャの戻り値は、受け取れない。
5. ストアドプロシージャ内の SELECT 文の結果は、受け取れない。

9.8.10 EXECUTE IMMEDIATE 文 (データベースアクセス機能)

形式

`EXECUTE IMMEDIATE` { 埋め込み変数 }
{ 英数字定数 }

機能

EXECUTE IMMEDIATE 文は、SQL 文を動的に準備し、実行する。

構文規則

埋め込み変数の型は、文字列型でなければならない。

一般規則

1. 埋め込み変数、英数字定数には、動的 SQL を指定しておかなければならない。
2. 動的 SQL には、パラメタマーカを含めてはならない。
3. 動的 SQL に SELECT 文を指定してはならない。
4. 動的 SQL が DELETE 文、INSERT 文、または UPDATE 文で該当の行が存在しない場合、SQLCODE 変数に 100 が設定される。

9.8.11 PREPARE 文 (データベースアクセス機能)

形式

`PREPARE` SQL文名 `FROM` { 埋め込み変数 }
{ 英数字定数 }

機能

PREPARE 文は、実行のために SQL 文を準備する。

構文規則

埋め込み変数の型は、文字列型でなければならない。

注

SQL 文名は、同一プログラム中でだけ有効である。SQL 文名をほかのプログラムにわたって使用することはできない。

一般規則

1. 埋め込み変数、英数字定数には、動的 SQL を指定しておかなければならない。
2. SQL 文名にカーソルを関連づけている場合、カーソルは閉じられていなければならない。
3. 同じ SQL 文名を使用してすでに PREPARE 文を実行している場合、暗黙に DEALLOCATE PREPARE 文が実行される。

9.8.12 DEALLOCATE PREPARE 文 (データベースアクセス機能)

形式

```
DEALLOCATE PREPARE SQL文名
```

機能

DEALLOCATE PREPARE 文は、PREPARE 文で準備した SQL 文を解放する。

構文規則

同一プログラム、利用者定義関数およびメソッド中に同じ SQL 文名を使用した PREPARE 文がなければならない。

一般規則

1. 同じ SQL 文名を使用して PREPARE 文を実行しておかなければならない。
2. SQL 文名にカーソルを関連づけている場合、カーソルは閉じられていなければならない。

9.8.13 EXECUTE 文 (データベースアクセス機能)

形式

```
EXECUTE SQL文名  
  [ USING 相手指定 [ , 相手指定 ] … ]  
  [ INTO 相手指定 [ , 相手指定 ] … ]
```

機能

EXECUTE 文は、準備された SQL 文を実行する。

構文規則

同一プログラム、利用者定義関数、メソッド中に同じ SQL 文名を使用した PREPARE 文がなければならない。

一般規則

1. 動的 SQL は、準備しておかなければならない。
2. USING の相手指定の数および順序は、パラメタマーカの並びに対応し、各相手指定のデータは、各パラメタマーカに対応する列、または式のデータ型に有効な値のデータでなければならない。なお、パラメタマーカに対応する列、または式のデータ型が固定長形式の型の場合、データの長さを合わせる必要がある。
3. 動的 SQL が SELECT 文の場合、結果を受け取るために INTO 指定を記述しなければならない。
4. INTO の相手指定の数および順序は、取り出す行の列の並びに対応し、各相手指定の埋め込み変数の型は、返される結果の値を格納できる埋め込み変数の型でなければならない。
5. 動的 SQL が SELECT 文で、導出される表が空の場合、または、動的 SQL が DELETE 文、INSERT 文、または UPDATE 文で該当の行が存在しない場合、SQLCODE 変数に 100 が設定される。

10

コマンド行のアクセス

コマンド行のアクセス機能は、コマンド行の引数へのアクセスおよび環境変数へのアクセスができる。

この章では、コマンド行のアクセス機能について説明する。

10.1 コマンド行、環境変数へのアクセス

コマンド行のアクセス機能は、コマンド行の引数へのアクセスおよび環境変数へのアクセスができる。

コマンド行の引数へのアクセスは、実行可能ファイルが起動されたコマンド行の引数や、その個数を求めることができる。

環境変数へのアクセスは、環境変数の値を設定することや、値を求めることができる。

10.1.1 環境部 (SPECIAL-NAMES) (コマンド行、環境変数へのアクセス)

形式

```
SPECIAL-NAMES.  
  [ ARGUMENT-NUMBER IS 呼び名4 ]  
  [ ARGUMENT-VALUE IS 呼び名5 ]  
  [ ENVIRONMENT-NAME IS 呼び名6 ]  
  [ ENVIRONMENT-VALUE IS 呼び名7 ]
```

構文規則

呼び名の規則に関しては、マニュアル「COBOL2002 言語 標準仕様編」[8.2.6 特殊名段落 (SPECIAL-NAMES)]を参照のこと。

10.1.2 手続き部 (コマンド行、環境変数へのアクセス)

形式

書き方 1 コマンド行に現れた引数の個数を読み込む

```
ACCEPT 一意名9 FROM 呼び名4 [END-ACCEPT]
```

書き方 2 コマンド行から引数を次々に読み込む

```
ACCEPT 一意名8 FROM 呼び名5  
  [ON EXCEPTION 無条件文3]  
  [NOT ON EXCEPTION 無条件文4]  
  [END-ACCEPT]
```

書き方 3 コマンド行から引数を位置で読み込む

```
DISPLAY { 一意名7 } UPON 呼び名4 [END-DISPLAY]  
         { 整数4 }  
ACCEPT 一意名8 FROM 呼び名5  
  [ON EXCEPTION 無条件文3]  
  [NOT ON EXCEPTION 無条件文4]  
  [END-ACCEPT]
```

書き方 4 環境変数から値を読み込む

```

DISPLAY { 一意名8 }
        { 定数2 } } UPON 呼び名6 [END-DISPLAY]
ACCEPT 一意名10 FROM 呼び名7
        [ON EXCEPTION 無条件文5]
        [NOT ON EXCEPTION 無条件文6]
[END-ACCEPT]

```

書き方 5 環境変数へ値を書き出す

```

DISPLAY { 一意名8 }
        { 定数2 } } UPON 呼び名6 [END-DISPLAY]
DISPLAY { 一意名9 }
        { 定数3 } } UPON 呼び名7
        [ON EXCEPTION 無条件文3]
        [NOT ON EXCEPTION 無条件文4]
[END-DISPLAY]

```

構文規則

- 書き方 1 の一意名 9 および書き方 3 の一意名 7 は、符号なし整数項目として記述されたデータ項目を参照するものでなければならない。
- 次に示すものは英数字データ項目を参照するものでなければならない。
 - 書き方 2 の一意名 8
 - 書き方 3 の一意名 8
 - 書き方 4 の一意名 8 および一意名 10
 - 書き方 5 の一意名 8 および一意名 9
- 書き方 3 の整数 4 は符号なし整数でなければならない。
- 書き方 4 の定数 2 および書き方 5 の定数 2, 3 は英数字定数でなければならない。
 - UniObjGen オプションを指定した場合、書き方 4 の定数 2 および書き方 5 の定数 2, 定数 3 に UTF-8 で多バイトとなる文字が含まれてはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

一般規則

- ARGUMENT-NUMBER (引数の番号) は、書き方 1 の ACCEPT で使う場合、コマンド行に含まれる引数の個数を一意名 9 に求める。また、書き方 3 の DISPLAY で使う場合、一意名 7 の内容または整数 4 をコマンド行上の引数の位置に設定する。
- 書き方 3 の DISPLAY を使わないと、コマンド行の初期位置は先頭の引数に設定される (書き方 2 を参照)。
- 書き方 2 または書き方 3 の ACCEPT 文の実行後に位置は 1 個進められる。
- ARGUMENT-VALUE (引数の値) は、書き方 2 または書き方 3 の ACCEPT 文で使用する場合、コマンド行上の引数の値を一意名 8 に求める。

5. ENVIRONMENT-NAME（環境変数の名称）は、書き方 4 または書き方 5 の DISPLAY で使用する場合、一意名 8 の内容または定数 2 を環境変数名に設定する。
6. ENVIRONMENT-VALUE（環境変数の値）は、書き方 4 の ACCEPT で使用する場合、環境変数の値を一意名 10 に求める。また、書き方 5 の DISPLAY で使用する場合、環境変数の内容に一意名 9 の内容または定数 3 を設定する。
7. 一意名 7 の内容または整数 4 は、0 から 99 の範囲であること。また、COBOL プログラムのコマンド行に現れる引数やスイッチおよびフラグを参照してもよい。
8. 無条件文 3 が実行されるのは、書き方 2 でコマンド行の最後の引数を超えて読み込みをしようとしたとき、または書き方 3 で引数が存在しないときである。
9. 無条件文 5 が実行されるのは、書き方 4 の DISPLAY 文による環境変数名の設定が未実行のとき、または指定した環境変数が存在しないときである。
10. 書き方 5 で無条件文 3 が実行されるのは、書き方 5 の DISPLAY 文による環境変数名の設定が未実行のとき、または指定した環境変数を格納するための十分な空き領域の割り当てができないときである。
11. 無条件文 4、無条件文 6 が実行されるのは、例外が発生しないときである。
12. 書き方 2、書き方 3 および書き方 4 の ACCEPT 文または書き方 5 の DISPLAY 文に、ON EXCEPTION 指定を書かないで例外が発生したときは、実行単位が異常終了する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

10.2 コマンド行へのアクセス（一括取得）

10.2.1 環境部（SPECIAL-NAMES）（コマンド行へのアクセス（一括取得））

形式

SPECIAL-NAMES.
COMMAND-LINE IS 呼び名

構文規則

呼び名の規則に関しては、標準仕様編の特殊名段落（SPECIAL-NAMES）を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落（SPECIAL-NAMES）」

10.2.2 手続き部（コマンド行へのアクセス（一括取得））

(1) ACCEPT COMMAND-LINE

形式

ACCEPT 一意名 FROM { COMMAND-LINE
[END-ACCEPT] COMMAND-LINEに対する呼び名 }

機能

ACCEPT 文の実行時に、コマンド行の引数の値を取得する。ARGUMENT-VALUE と異なり、空白で区切られた複数の引数も 1 回の ACCEPT 文で一括して取得できる。

一般規則

1. 一意名に書ける項目は、機能名 CONSOLE と同等である。
2. コマンド行バッファ内の連続する空白文字は、空白文字一つにまとめる。ただし、DISPLAY COMMAND-LINE を実行した後では、連続する空白文字を一つにまとめない。
3. COMMAND-LINE と ARGUMENT-VALUE は互いに影響しない。ACCEPT ARGUMENT-VALUE で引数の一部を読み込んだ後に、ACCEPT COMMAND-LINE で引数を読み込むと、引数の全体が読み込まれる。
4. ACCEPT COMMAND-LINE を連続して指定した場合、最初の 1 回だけが有効となり、2 回目以降では引数は読み込まれない。

(備考)

一意名が数字項目、数字編集項目、外部浮動項目または内部浮動項目の場合、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の NumAccept オプションに関する記述も参照のこと。

(2) DISPLAY COMMAND-LINE

形式

$$\text{DISPLAY} \left\{ \begin{array}{l} \text{一意名} \\ \text{定数} \end{array} \right\} \text{UPON} \left\{ \begin{array}{l} \text{COMMAND-LINE} \\ \text{COMMAND-LINEに対する呼び名} \end{array} \right\} \\ \text{[END-DISPLAY]}$$

機能

DISPLAY 文の実行時に、一意名または定数で指定された内容を、コマンド行の引数の値を格納しているコマンド行バッファに上書きする。

DISPLAY COMMAND-LINE の実行後に ACCEPT COMMAND-LINE を実行すると、DISPLAY COMMAND-LINE で設定した値が取り出される。

一般規則

1. 一意名に内部ブール項目、可変長集団項目、指標データ項目、オブジェクト参照項目、およびアドレスデータ項目は指定できない。
2. DISPLAY 一意名 UPON COMMAND-LINE を実行すると、一意名の内容でコマンド行バッファの内容を書き換える。ただし、COMMAND-LINE のコマンド行バッファと ARGUMENT-VALUE で使用するバッファは別のバッファとする。すなわち、DISPLAY COMMAND-LINE で書き込んだデータは、ACCEPT COMMAND-LINE でだけ取り出すことができる。
3. DISPLAY 一意名 UPON COMMAND-LINE の作用対象は一つだけしか許されない。一意名の中に空白で区切られたデータが格納されていた場合は、空白を含んだ全体を一つの作用対象とみなす。

11

通信節による画面機能

この章では、通信節を用いて画面データをディスプレイに表示したり、利用者が入力したデータをディスプレイから読み込んだり、帳票データをプリンタに出力したりする機能について説明する。

この機能は AIX(64) COBOL2002 および Linux COBOL2002 では使用できない。

AIX(64) COBOL2002 および Linux COBOL2002 でこの機能を使用した場合、XMAP3 のライブラリが存在しないため、リンク時にエラーとなる。

11.1 データ部（通信節による画面機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

11.1.1 通信記述項 (CD) (通信節による画面機能)

形式

CD 通信記述名1 FOR $\left\{ \begin{array}{l} \text{OUTPUT} \\ \text{I-O} \end{array} \right\} \text{WS}$

MAP NAME IS データ名1
[MAPPING MODE IS データ名2]
[DATA ABSENCE CODE IS データ名3]
[STATUS KEY IS データ名4]
[SYMBOLIC TERMINAL IS データ名5].

機能

通信種別などを指定する。

構文規則

1. 帳票出力用の通信記述項と画面入出力用の通信記述項はおのこのプログラムに 1 回だけ定義できる。
2. 通信記述項の各データ名に対して、データ項目が次の形式で暗黙に定義される。また、これらの領域はこのシステムによって自動的に割り当てられる。

データ名	形式
データ名1	X (8)
データ名2	X (1)
データ名3	X (1)
データ名4	X (5)
データ名5	X (8)

3. 通信記述名 1 の構成規則は、利用者定義語の構成規則による。
4. 通信節は最外側プログラム、利用者定義関数、ファクトリ定義およびインスタンス定義に指定できる。
5. 画面入出力用の通信記述項には、MAP NAME 句を書かなければならない。

一般規則

1. FOR 句は、通信種別を指定する。通信種別と意味を次に示す。
OUTPUT WS：送信先が、プリンタである。
I-O WS：送受信先が、ディスプレイである。
2. MAP NAME 句のデータ名 1 は、出力用物理マップ名を設定する項目である。
3. MAPPING MODE 句のデータ名 2 は、画面表示の形態を指定する。データ名 2 に指定する値と画面表示の形態を次に示す。

'△', '0'

論理項目または制御項目に指定したデータを用いて画面表示をする。ただし、表示属性の変更指示を行わない項目属性については、画面定義時に指定した表示属性となる。

'2'

論理項目または制御項目に指定したデータを用いて画面表示をする。ただし、表示属性の変更指示を行わない項目属性については、前画面の表示属性となる。

'3'

論理項目または制御項目に指定したデータを無視して画面定義時の情報だけで画面表示をする。

データ名 2 に'2'を設定しているとき、表示動作が ERASE モードとなる通信文を発行してはならない。この通信文を発行した場合、動作は保証しない。

MAPPING MODE 句を省略した場合、次を仮定する。

- ・初めての送受信要求または直前の指定と異なる物理マップ名の送受信要求のとき、データ名 2 に'△', '0'が指定されたものとして画面表示をする。

- ・同じ物理マップ名を使用した送受信要求のとき、データ名 2 に'2'が指定されたものとして画面表示をする。ただし、同じ物理マップ名を使用した送受信要求でも、表示動作が ERASE モードの場合は、データ名 2 に、'△', '0'を仮定し、画面表示をする。

MAPPING MODE 句は、FOR 句で I-O WS を指定しているときだけ有効である。FOR 句で OUTPUT WS を指定しているとき、常にデータ名 2 に'△', '0'が指定されたものとして画面表示をする。

(注意)

論理項目とは、プリンタまたはディスプレイ間で、可変なデータを送信または送受信するためのデータ項目である。

制御項目とは、COBOL プログラムとディスプレイまたはプリンタを制御するためのデータ項目である。

4. DATA ABSENCE CODE 句のデータ名 3 は、データ有無コードを指定する。データ有無コードとは、論理項目に値を設定しているかどうかを区別するためのコードである。データ名 3 で指定したコードを論理項目の先頭 1 バイトに設定することで、その論理項目へ値を設定しないことを指示する。
5. 通信文の実行で異常が発生した場合、STATUS KEY 句の指定がないと、次のどちらかの動作になる。
 - (a) 通信文で実行継続できない異常状態が発生すると、エラーメッセージを出力して終了する。
 - (b) 通信文で実行継続できる異常状態が発生すると、エラーメッセージを出力して続行する。
6. 通信文の実行で異常が発生しても処理を続行させたい場合、STATUS KEY 句を指定する。データ名 4 には、通信文の実行状態を表す長さ 5 文字のコードが設定される。

5 文字のコードと意味を次に示す。

コード	意味
00000	正常終了
00008	論理エラー
10004	マッピングエラー
10008	マッピングエラー
20008	入出力エラー

STATUS KEY 句の指定は、異常状態の発生時、ファイルのクローズなどの後処理をしてからプログラムを終了させたいときに有効である。ただし、STATUS KEY 句を指定していても、データ名 4 の値がすべて空白のとき、STATUS KEY 句の指定は無視される。

7. INITIAL 句指定のプログラムで EXIT PROGRAM 文または GOBACK 文を実行した場合、この機能で開かれたファイルは閉じられる。

8. SYMBOLIC TERMINAL 句のデータ名 5

(a) FOR 句で OUTPUT を指定したときは、メッセージの送信先の仮想端末名称を指定する。仮想端末とは、COBOL プログラムで扱う論理的な端末である。

(b) FOR 句で I-O を指定したときは、画面データの送受信先を指定する。画面データの送受信先はディスプレイである。

詳細は、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の通信節による画面入出力機能の説明と XMAP3 のマニュアルを参照のこと。

11.2 手続き部（通信節による画面機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

11.2.1 SEND 文（通信節による画面機能）

形式

```
SEND 通信記述名1 FROM 一意名1  
WITH EMI [NO REPLY]
```

機能

1. ディスプレイに対してメッセージを送信する。このときメッセージを受信しない。
2. ディスプレイに対してメッセージを送受信する。このとき、送信されたメッセージに対する受信の完了を待たない。
3. プリンタへメッセージを送信する。

構文規則

1. 通信記述名 1 には、SEND 文に対応する通信記述項の通信記述名を指定する。
2. 一意名 1 には、送信メッセージ格納領域（出力論理マップ）を指定する。
3. プリンタに対する SEND 文では、NO REPLY 指定をしてはならない。
4. 一意名 1 には、次のデータ名は指定できない。
 - ・ 指標データ項目
 - ・ アドレスデータ項目
 - ・ ブール項目
 - ・ オブジェクト参照データ項目
 - ・ オブジェクトプロパティ
 - ・ 関数一意名
5. 一意名 1 は、部分参照してはならない。

一般規則

1. SEND 文を実行する前に、通信記述項の MAP NAME 句で指定したデータ名 1 のデータ項目に出力物理マップ名を設定しなければならない。
2. ディスプレイに対する NO REPLY 指定のある SEND 文を実行すると、メッセージが送信される。このとき、RECEIVE 文によってメッセージを受信することはできない。
3. ディスプレイに対する NO REPLY 指定のない SEND 文を実行すると、メッセージが送受信される。このとき、RECEIVE 文を実行して、メッセージを受け取らなければならない。NO REPLY 指定のない SEND 文から RECEIVE 文の間にディスプレイに対する送受信以外の文を実行できる。

4. プリンタに対する SEND 文を実行すると、メッセージが送信される。
5. 送信メッセージ格納領域には、SEND 文の実行前に論理データや制御データを設定しておかなければならない。

11.2.2 RECEIVE 文 (通信節による画面機能)

形式

RECEIVE 通信記述名1
FIRST SEGMENT INTO 一意名1

機能

SEND 文によるメッセージ送受信要求の完了を待ち、受信したメッセージを受け取る。

構文規則

1. 通信記述名 1 には、RECEIVE 文に対応する通信記述項の通信記述名を指定する。
2. 一意名 1 には、受信メッセージ格納領域 (入力論理マップ) を指定する。
3. 一意名 1 には、次のデータ名は指定できない。
 - ・指標データ項目
 - ・アドレスデータ項目
 - ・ブール項目
 - ・オブジェクト参照データ項目
 - ・オブジェクトプロパティ
 - ・関数一意名
4. 一意名 1 は、部分参照してはならない。
5. 一意名 1 には強く型付けされた集団項目を指定できない。

一般規則

1. RECEIVE 文は、NO REPLY 指定のない SEND 文に続いて実行しなければならない。NO REPLY 指定のない SEND 文から RECEIVE 文の間に、ディスプレイに対する送受信以外の文を実行できる。
2. 受信メッセージ格納領域は、受信の成功した RECEIVE 文の実行後に参照できる。

11.2.3 TRANSCEIVE 文 (通信節による画面機能)

形式

TRANSCEIVE 通信記述名1
FROM 一意名1 INTO 一意名2

機能

ディスプレイに対してメッセージの送信および受信をする。メッセージ受信の完了を待つ。また、送信だけをすることはできない。

構文規則

1. 通信記述名 1 には、TRANSCEIVE 文に対応する通信記述項の通信記述名を指定する。
2. 一意名 1 には、送信メッセージ格納領域（出力論理マップ）を指定する。
3. 一意名 2 には、受信メッセージ格納領域（入力論理マップ）を指定する。
4. 一意名 1 および一意名 2 には、次のデータ名は指定できない。
 - ・指標データ項目
 - ・アドレスデータ項目
 - ・ブール項目
 - ・オブジェクト参照データ項目
 - ・オブジェクトプロパティ
 - ・関数一意名
5. 一意名 1 および一意名 2 は、部分参照してはならない。
6. 一意名 2 には強く型付けされた集団項目を指定できない。

一般規則

1. TRANSCEIVE 文実行前に、通信記述項の MAP NAME 句で指定したデータ名 1 のデータ項目に出力物理マップ名を設定しなければならない。
2. 送信メッセージ格納領域は、TRANSCEIVE 文の実行前に論理データや制御データを設定しておかなければならない。
3. 受信メッセージ格納領域は、受信の成功した TRANSCEIVE 文の実行後に参照できる。

11.2.4 DISABLE 文（通信節による画面機能）

形式

`DISABLE` 通信記述名 1

機能

1. ディスプレイに出力された画面を消去する。
2. プリンタを閉じる。

構文規則

1. 通信記述名 1 には、DISABLE 文に対応する通信記述項の通信記述名を指定する。

一般規則

1. DISABLE 文は、SEND 文または TRANSCEIVE 文に続いて実行しなければならない。

12

画面節 (SCREEN SECTION) による画面機能

この章では、画面節 (SCREEN SECTION) を用いて画面データをディスプレイに表示したり、ユーザーが入力したデータをディスプレイから読み込んだりする機能について説明する。

この機能は、Linux COBOL2002 では使用できない。Linux COBOL2002 でこの機能に関する次に示す構文を記述した場合、重大エラーを出力する。

- SCREEN SECTION
- ACCEPT 文 (書き方 4)
- DISPLAY 文 (書き方 3)

この章で記載している終了キーとは、ACCEPT 文の実行において、データの入力を終了するときに打鍵するキーであり、[Enter]キー、[実行]キー、[PF]キーが使用できる。

12.1 環境部 (SCREEN SECTION)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

12.1.1 特殊名段落 (SPECIAL-NAMES) (SCREEN SECTION)

形式

```
SPECIAL-NAMES.  
  [CONSOLE IS CRT]  
  [CURSOR IS データ名1]  
  [CRT STATUS IS データ名2]
```

機能

各句の規則を 12.1.2 以降で述べる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落 (SPECIAL-NAMES)」

12.1.2 CONSOLE IS CRT 句 (SCREEN SECTION)

形式

```
CONSOLE IS CRT
```

機能

「12.3.1 ACCEPT 文 (SCREEN SECTION)」および「12.3.2 DISPLAY 文 (SCREEN SECTION)」の書き方 2 で示される形式を許す。

一般規則

作用対象が画面項目でない ACCEPT 文または DISPLAY 文を書き方 2 で扱うようにする。この句がなければ標準仕様の ACCEPT 文または DISPLAY 文として扱われる。

12.1.3 CURSOR 句 (SCREEN SECTION)

形式

```
CURSOR IS データ名1
```

機能

ACCEPT 文が用いるカーソルアドレスを格納するデータ名を指定する。これによって、ACCEPT 文実行時のカーソルの初期位置を指定できる。また、プログラム中で ACCEPT 文終了時のカーソル位置を保持できる。

構文規則

1. データ名 1 は、そのプログラム、利用者定義関数、ファクトリ定義およびインスタンス定義の作業場所節または局所場所節で宣言されていなければならない。また、データ名 1 は、4 文字長か 6 文字長でなければならない。
2. データ名 1 は、暗黙的にまたは明示的に USAGE IS DISPLAY として記述された符号なし整数基本項目を記述するものか、またはそのような項類の基本データ項目から成る集団項目を記述するものでなければならない。

一般規則

1. CURSOR 句が指定されていない場合は、ACCEPT 文実行時のデフォルトのカーソル位置は、画面上の最初の入力フィールドまたは更新フィールドの先頭となる。
2. データ名 1 が 4 文字長の場合、先頭の 2 文字は行番号、残る 2 文字はカラム番号と解釈される。データ名 1 が 6 文字長の場合、先頭の 3 文字は行番号、残る 3 文字はカラム番号と解釈される。
3. カーソルアドレスが入力フィールドまたは更新フィールド中の位置を示していない場合、カーソルの初期位置は未定義として扱われ、最初の入力フィールドまたは更新フィールドの先頭に位置づけられる。
このシステムでは、カーソルの初期位置が未定義のときに、ACCEPT 文終了時のカーソル位置をデータ名 1 に格納する。また、データ名 1 に不当な値が入っていた場合、カーソルの初期位置は未定義として扱われる。
4. CURSOR 句は複数定義してはならない。

12.1.4 CRT STATUS 句 (SCREEN SECTION)

形式

`CRT STATUS IS データ名2`

機能

ACCEPT 文の動作の終了の原因を表す値を格納するデータ名を指定する。

構文規則

1. データ名 2 は、そのプログラム、利用者定義関数、ファクトリ定義およびインスタンス定義の作業場所節または局所場所節で宣言されていなければならない。また、データ名 2 は、3 文字長の集団項目でなければならない。

一般規則

1. CRT STATUS 句が指定されている場合、データ名 2 は各 ACCEPT 文終了後、そのつど更新される。
2. データ名 2 の先頭の文字は CRT 状態キー 1 であり、PICTURE 9 USAGE DISPLAY と記述されたものでなければならない。
3. CRT 状態キー 1 は、ACCEPT 文の動作の終了の原因を表す値であり、「表 12-1 CRT 状態キー 1, CRT 状態キー 2 の意味」に示す値を取り得る。
4. データ名 2 の 2 番目の文字は CRT 状態キー 2 であり、PICTURE X と記述されたものでなければならない。
5. CRT 状態キー 2 は、ACCEPT 文の動作の終了の原因について、より詳細な情報を与える値であり、CRT 状態キー 1 との組み合わせで「表 12-1 CRT 状態キー 1, CRT 状態キー 2 の意味」に示す値を取り得る。
6. データ名 2 の 3 番目の文字は、CRT 状態キー 3 であり、PICTURE X と記述されたものでなければならない。
7. Windows COBOL2002 では、CBLADTRM サービスルーチンによって終了キーを登録した場合、CRT 状態キー 2, 3 に終了キーコードが設定される。CBLADTRM サービスルーチンの詳細については、マニュアル「COBOL2002 ユーザーズガイド」の CBLADTRM サービスルーチンの説明を参照のこと。
8. CRT STATUS 句は複数指定してはならない。

表 12-1 CRT 状態キー 1, CRT 状態キー 2 の意味

CRT 状態キー 1	CRT 状態キー 2	CRT 状態キー 3	CRT 状態キー 1 の意味	CRT 状態キー 2 および 3 の意味
0	0	未使用	正常終了した (Enter キーが押されたかあるいは最終フィールド外にオートスキップした)。	Enter キーが押された。
	1	未使用		最終フィールド外にオートスキップした。
1	X'00'~X'FF'	未使用	ファンクションキーが押された。	ファンクションキー番号。
2	X'00'~X'FF'	未使用	実行キーが押されたかあるいは CBLADTRM サービスルーチンで定義した終了キーが押された。	CBLADTRM サービスルーチンで定義した終了キーが押された (1 バイトコード)。
	X'80'	X'00'~X'FF' (X'7E'は除く)		CBLADTRM サービスルーチンで定義した終了キーが押された (2 バイトコード)。
	X'80'	X'7E'		実行キーが押された。
9	X'00'	未使用	エラーが発生した。	エラー発生のため転記する項目がない。

12.2 データ部 (SCREEN SECTION)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

12.2.1 画面節 (SCREEN SECTION)

形式

SCREEN SECTION.
[画面記述項]…

機能

ACCEPT 文と DISPLAY 文で用いる画面操作機能を提供する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.1.1 一般形式」

12.2.2 画面記述項 (SCREEN SECTION)

形式

書き方 1

レベル番号 { 画面名1
FILLER }

[BLANK SCREEN]

[{ FOREGROUND-COLOR } IS 整数4
{ FOREGROUND-COLOUR }]

[{ BACKGROUND-COLOR } IS 整数5
{ BACKGROUND-COLOUR }]

[[USAGE IS] { DISPLAY }
{ NATIONAL }]

[[SIGN IS] { LEADING }
{ TRAILING }]

[SEPARATE CHARACTER]

[AUTO]

[AUTO-SKIP]

[SECURE]

[NO-ECHO]

[REQUIRED]

[EMPTY-CHECK]

[PROMPT CHARACTER IS { 一意名1 }
{ 定数1 }]

[FULL]

[LENGTH-CHECK]

[ZERO-FILL].

書き方 2

レベル番号 { 画面名1
FILLER }

[BLANK { LINE
SCREEN }]

[BELL]

[BEEP]

[BLINK]

[ERASE { EOL
EOS }]

[HIGHLIGHT]

[LOWLIGHT]

[GRID]

[LEFTLINE]

[OVERLINE]

[REVERSE-VIDEO]

[UNDERLINE]

[SIZE IS { 一意名1
整数1 }]

[LINE [NUMBER IS [{ PLUS
+
- }] { 一意名2
整数2 }]]

[{ COLUMN
COL } [NUMBER IS [{ PLUS
+
- }] { 一意名3
整数3 }]]

[{ FOREGROUND-COLOR
FOREGROUND-COLOUR } IS 整数4]

[{ BACKGROUND-COLOR
BACKGROUND-COLOUR } IS 整数5]

VALUE IS 定数1.

書き方 3

レベル番号 { 画面名1 }
 { FILLER }
 [BLANK { LINE }]
 { SCREEN }]
 [BELL]
 [BEEP]
 [BLINK]
 [ERASE { EOL }]
 { EOS }]
 [HIGHLIGHT]
 [LOWLIGHT]
 [GRID]
 [LEFTLINE]
 [OVERLINE]
 [REVERSE-VIDEO]
 [UNDERLINE]
 [SIZE IS { 一意名1 }]
 { 整数1 }]
 [LINE [NUMBER IS [{ PLUS } { 一意名2 }]]]
 { + } { 整数2 }]]
 { - }]]
 [{ COLUMN } [NUMBER IS [{ PLUS } { 一意名3 }]]]]
 { COL } { + } { 整数3 }]]
 { - }]]
 [{ FOREGROUND-COLOR }] IS 整数4]
 { FOREGROUND-COLOUR }]]
 [{ BACKGROUND-COLOR }] IS 整数5]
 { BACKGROUND-COLOUR }]]
 [PICTURE]
 [PIC]]

IS picture文字列1

{ USING 一意名4
[FROM { 一意名5
定数1 }] [TO 一意名6] }

[[USAGE IS] { DISPLAY
NATIONAL }]

[BLANK WHEN ZERO]

[{ JUSTIFIED } RIGHT
JUST]

[[SIGN IS] { LEADING
TRAILING }] [SEPARATE CHARACTER]

[AUTO
AUTO-SKIP]

[SECURE]

[NO-ECHO]

[REQUIRED]

[EMPTY-CHECK]

[PROMPT CHARACTER IS { 一意名7
定数2 }]

[FULL]

[LENGTH-CHECK]

[ZERO-FILL] .

機能

参照される画面項目の属性、動作、サイズ、位置を指定する。

構文規則

1. 画面記述項はそれぞれ、01～49 または 77 のレベル番号で始まらなければならない。
2. 基本画面記述項はそれぞれ、以下の句のうちのどれかを含まなければならない。
PICTURE, VALUE, BELL, BLANK, COLUMN, LINE
3. 一つの画面項目に対して 2 回以上同じ句が指定された場合、階層的に最下位のレベルで指定されたものが有効となる。
4. 画面名 1 または予約語 FILLER のどちらかを指定する場合、各画面記述項のレベル番号の直後に記述しなければならない。
画面名 1 は画面記述項で記述される画面項目に名前を割り当てるものであり、利用者定義語の規則に従っていなければならない。
5. 画面名 1 または予約語 FILLER を省略する場合、記述される画面項目は FILLER が指定されたものとして扱われる。
6. 画面名 1 および FILLER の後に書く各句の順序は任意である。
7. 予約語 FILLER は画面項目名として使われてもよい。ただし、どんな環境下でも FILLER 項目を明示的に参照することはできない。

8. レベル 01 の項目には、それぞれ画面名を指定しなければならない。
9. 画面項目は ACCEPT 文と DISPLAY 文でだけ参照できる。
ACCEPT 文、DISPLAY 文以外に画面項目が指定された場合エラーとする。
10. SCREEN SECTION はプログラム、利用者定義関数、ファクトリ定義およびインスタンス定義だけに指定できる。
11. FROM、TO、USING 指定で用いるデータ項目は、画面項目と関係している。USING 指定は、FROM と TO 指定の組み合わせでそれぞれに同じ一意名を指定することと等価である。

一般規則

1. 書き方 1 は集団を記述するものである。
2. 書き方 2 は定数画面項目に用いるものである。
3. 書き方 3 はデータ項目である画面項目に用いるものである。
 - ・入力フィールドは、記述項に TO 指定を持つ画面項目である
 - ・出力フィールドは、記述項に FROM 指定を持つ画面項目である
 - ・更新フィールドは、記述項に USING 指定を持つ画面項目である
 - ・FROM 指定および TO 指定と、USING 指定は同時に指定してはならない。
4. 更新フィールドの規則は、記述項に FROM と TO 指定の両方を持つ画面項目にも適用される。ここで、FROM と TO 指定は同じ一意名を参照していてもよい。
5. 書き方 1 で SECURE 句が指定されている場合、その指定はその集団項目に従属する入力フィールドそれぞれに対して適用される。
6. 書き方 1 で AUTO、BACKGROUND-COLOR、BACKGROUND-COLOR、FULL、REQUIRED 句が指定されている場合、その指定はその集団項目に従属する入力フィールドと更新フィールドそれぞれに対して適用される。
7. 画面記述項に LINE と COLUMN 句を記述する場合、画面上で重なり合ったり、端末に即して規定されている COLS や LINES の外にはみ出してしまうような指定をしてはいけない。ただしこのシステムでは 65,536 以上をエラーとする。
8. 書き方 2 の VALUE 句と書き方 3 の PICTURE 句を同時に指定してはならない。

12.2.3 AUTO 句 (SCREEN SECTION)

形式

$\left\{ \begin{array}{l} \text{AUTO} \\ \text{AUTO-SKIP} \end{array} \right\}$

機能

ACCEPT 文の実行中に、入力フィールドまたは更新フィールドの最後の文字が入力されたとき、カーソルを自動的に次のフィールドに移動させる。

構文規則

1. AUTO と AUTO-SKIP は同義語である。
2. AUTO 句は定数画面項目の記述項に指定してはいけない。

一般規則

1. AUTO 句が集団のレベルに指定された場合、その指定はその集団に従属する入力フィールドまたは更新フィールドそれぞれに対して適用される。
2. AUTO 句は ACCEPT 文の実行中にだけ有効であり、DISPLAY 文の実行中には無視される。
3. AUTO 句は基本出力フィールドでは無視される。
4. 入力フィールドが唯一の場合または現在のフィールドが集団に従属する最後のフィールドの場合、ACCEPT 文は完了する。

12.2.4 BACKGROUND-COLOR 句 (SCREEN SECTION)

形式

$\left. \begin{array}{l} \text{BACKGROUND-COLOR} \\ \text{BACKGROUND-COLOUR} \end{array} \right\} \text{ IS 整数5}$

機能

画面項目の背景色を指定する。

構文規則

1. BACKGROUND-COLOR と BACKGROUND-COLOUR は同義語である。
2. この句はあらゆる画面項目に指定してよい。
3. この句が集団のレベルに指定された場合、その指定はその集団のすべての基本従属項目に対して適用される。
4. 整数 5 は 0~7 の値でなければならない。

一般規則

1. この句はカラー画面でだけ有効である。
2. 整数 5 は画面項目の背景色を指定する。値と色の関係は次に示すとおり。

値	色
0	黒
1	青
2	緑
3	空色
4	赤

値	色
5	紫色
6	黄
7	白

12.2.5 BELL 句 (SCREEN SECTION)

形式

$$\left\{ \begin{array}{l} \text{BELL} \\ \text{BEEP} \end{array} \right\}$$

機能

画面項目が表示開始される時、警告音を鳴らす。

構文規則

1. BELL と BEEP は同義語である。
2. BELL 句は基本画面記述項にだけ指定できる。

一般規則

1. この句を指定することで、DISPLAY 文の実行中、指定された画面項目の表示開始時に警告音が鳴る。

12.2.6 BLANK 句 (SCREEN SECTION)

形式

$$\text{BLANK} \left\{ \begin{array}{l} \text{LINE} \\ \text{SCREEN} \end{array} \right\}$$

機能

画面項目が表示される時、画面の 1 行または画面全体をクリアする。

構文規則

1. BLANK SCREEN 句は、どんな画面記述項に対しても指定できる。
2. BLANK LINE 句は、基本画面記述項に対してだけ指定できる。

一般規則

1. BLANK SCREEN 句は、同一記述内のほかの句よりも優先して画面項目の DISPLAY 文実行前に実行される。この句が指定された場合、画面はクリアされる。
2. BLANK LINE 句が指定された場合、この句を記述した画面項目が指定された行の 1 カラム目からその行の終わりまでがクリアされる。

3. BLANK 句と ERASE 句のどちらも指定されなかった場合、画面項目が表示されるときには、その画面項目に対応する文字位置だけ書き換えられ、残りの部分の内容はそのまま保存される。
4. BLANK SCREEN 句は、画面を標準値の背景色と前景色に戻す。
5. BLANK 句は ACCEPT 文中では無視される。

12.2.7 BLANK WHEN ZERO 句 (SCREEN SECTION)

形式

BLANK WHEN ZERO

機能

データ項目の値が 0 のとき、空白を表示する。

一般規則

1. BLANK WHEN ZERO 句は、基本入力フィールドについては無視される。
ただし、このシステムでは有効とする。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.7 BLANK WHEN ZERO 句」

12.2.8 BLINK 句 (SCREEN SECTION)

形式

BLINK

このシステムでは BLINK 句を覚え書きとする。

機能

画面項目が画面上に表示される時、その画面項目を明滅させる。

構文規則

1. BLINK 句はどんな画面記述項にも指定できる。

12.2.9 COLUMN 句 (SCREEN SECTION)

形式

$$\left\{ \begin{array}{l} \text{COLUMN} \\ \text{COL} \end{array} \right\} [\text{NUMBER IS} \\ \left[\left\{ \begin{array}{l} \text{PLUS} \\ + \\ - \end{array} \right\} \right] \left\{ \begin{array}{l} \text{一意名3} \\ \text{整数3} \end{array} \right\}]$$

機能

画面項目の画面上の水平方向の座標を指定する。

構文規則

1. COLUMN と COL は同義語である。
2. PLUS と+は同義語である。
3. COLUMN 句は基本画面記述項に対してだけ指定できる。
4. 整数 3 は 0 以外の符号なし整数でなければならない。
5. 一意名 3 は整数基本データ項目として記述されたものでなければならない。
一意名 3 は部分参照されてはならない。
6. 一意名 3 はファイル節，作業場所節，局所場所節または連絡節で定義されなければならない。

一般規則

1. COLUMN 句は，ACCEPT 文または DISPLAY 文の動作時に画面項目が表示される位置のカラム番号を指定する。
2. COLUMN 句に一意名または整数の指定はあるが，PLUS，+，-のどんな指定もない場合，この句は絶対カラム番号を与える。
3. COLUMN 句で PLUS，+，-のどれかの指定がなされた場合，カラム番号は，その文が画面節 (SCREEN SECTION) で定義されている一つ前の画面項目を画面に表示したかどうかにかかわらず，画面節 (SCREEN SECTION) で定義されている一つ前の画面項目が終わる位置からの相対位置となる。これは，PICTURE または VALUE 句で導かれる項目の現在の長さに依存する。
レベル 01 の場合，カラム番号の計算は画面の 1 カラム目から再スタートする。
4. ACCEPT 文または DISPLAY 文で参照される最初の画面項目のカラム番号は，相対カラム番号で指定してはいけない。最初の画面項目が相対カラム番号で指定された場合は画面の 1 カラム目を基準として計算される。
5. LINE 句を指定して COLUMN 句を省略した場合，COLUMN 1 が仮定される。
6. LINE 句，COLUMN 句をともに省略した場合，次の規則が適用される。
 - ・先行する画面項目が定義されていない場合，LINE 1，COLUMN 1 が仮定される。
 - ・先行する画面項目が定義されている場合，先行する画面項目の行および COLUMN PLUS 1 が仮定される。

7. NUMBER 指定を省略すると、PLUS 1 が仮定される。

12.2.10 ERASE 句 (SCREEN SECTION)

形式

$$\text{ERASE } \left\{ \begin{array}{l} \text{EOL} \\ \text{EOS} \end{array} \right\}$$

機能

行の一部またはカーソル位置以降の画面をクリアする。

構文規則

1. ERASE 句は基本画面記述項に対してだけ指定できる。

一般規則

1. ERASE EOL が指定された場合、クリアはこの句が指定された画面項目の行および列の座標から始まり、その行の終わりまで続く。
2. ERASE EOS が指定された場合、クリアはこの句が指定された画面項目の行および列の座標から始まり、その画面の終わりまで続く。
3. BLANK 句も ERASE 句も指定されていない場合、画面項目が表示されるときには、その画面項目に対応する文字位置だけ書き換えられ、残りの部分の内容はそのまま保存される。
4. ERASE 句は ACCEPT 文中では無視される。

12.2.11 FOREGROUND-COLOR 句 (SCREEN SECTION)

形式

$$\left\{ \begin{array}{l} \text{FOREGROUND-COLOR} \\ \text{FOREGROUND-COLOUR} \end{array} \right\} \text{ IS 整数4}$$

機能

画面項目の前景色を指定する。

構文規則

1. FOREGROUND-COLOR と FOREGROUND-COLOUR は同義語である。
2. この句はあらゆる画面項目に指定してよい。
3. この句が集団のレベルに指定された場合、その指定はその集団のすべての基本従属項目に対して適用される。
4. 整数 4 は 0~7 の値でなければならない。

一般規則

1. この句はカラー画面でだけ有効である。
2. 整数 4 は画面項目の前景色を指定する。値と色の関係は次に示すとおり。

値	色
0	黒
1	青
2	緑
3	空色
4	赤
5	紫色
6	黄
7	白

3. この句が指定されていない場合、前景色はシステム依存の色となる。
4. HIGHLIGHT 句が同時に指定された場合、前景色は明度と輝度を増す。例えば、黒は灰色になる。

12.2.12 FULL 句 (SCREEN SECTION)

形式

{
FULL
LENGTH-CHECK
}

機能

フィールドを完全に空白のままにしておくか、またはそのフィールドをデータで満たすかのどちらかでなければならないことを指定する。

構文規則

1. FULL と LENGTH-CHECK は同義語である。
2. 画面記述項に FULL 句が指定された場合、JUSTIFIED 句は指定してはならない。
3. FULL 句は、入力フィールドまたは更新フィールドに対してだけ有効である。
4. この句が集団のレベルに指定された場合、その指定は従属するすべての入力フィールドまたは更新フィールドに適用される。

一般規則

1. FULL 句は、画面項目を入力させるあらゆる ACCEPT 文の実行中に有効である。ただし、入力受け付け中の画面項目に、カーソルが位置づけられていること。
この句が満足されるまで、終了キーのタイプは拒否される。

- 画面項目が英数字の場合、この句を満たすためには、完全に空白であるかまたは最初と最後の文字が共に空白以外の文字であるかのどちらかでなければならない。
- 画面項目が数字または数字編集の場合、この句を満たすためには、値が0であるかまたはゼロ抑制が実施されるけた位置がないかのどちらかでなければならない。
- 更新フィールドに FULL 句が指定された場合、PICTURE 句の FROM または USING 指定で参照される一意名の内容またはオペレータの入力データのどちらかがこの句を満たすまで終了キーのタイプは拒否される。
- 入力待ちの状態を終了させるためにファンクションキーが用いられた場合は、FULL 句は必ずしも有効ではない。
- FULL 句と REQUIRED 句を同時に指定した場合、オペレータに常にフィールドを全部満たすことを要求するものとなる。
- FULL 句は、基本出力フィールドについては無視される。

12.2.13 GRID 句 (SCREEN SECTION)

形式

GRID

機能

画面項目が画面上に表示される時、その画面項目の各文字の左側に縦線を施す。

構文規則

- GRID 句はどんな画面記述項に対しても指定できる。

12.2.14 HIGHLIGHT 句 (SCREEN SECTION)

形式

HIGHLIGHT

機能

画面項目を画面上に最高明度で表示する。

構文規則

- HIGHLIGHT 句はどんな画面記述項に対しても指定できる。ただし、同じ画面項目に対し HIGHLIGHT 句と LOWLIGHT 句を同時に指定してはならない。

12.2.15 JUSTIFIED 句 (SCREEN SECTION)

形式

$$\left. \begin{array}{l} \underline{\text{JUSTIFIED}} \\ \underline{\text{JUST}} \end{array} \right\} \text{RIGHT}$$

機能

画面項目にデータが転記されるとき、標準的でないけた寄せが行われる。

構文規則

1. JUSTIFIED 句は、定数画面項目の記述項に指定してはならない。
2. JUSTIFIED 句は、英数字編集項目、数字項目、数字編集項目および日本語編集項目に指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.36 JUSTIFIED 句」

12.2.16 LEFTLINE 句 (SCREEN SECTION)

形式

LEFTLINE

機能

画面項目が画面上に表示されるとき、その画面項目の最左端の文字の左側に縦線を施す。

構文規則

1. LEFTLINE 句はどんな画面記述項に対しても指定できる。

12.2.17 LINE 句 (SCREEN SECTION)

形式

$$\underline{\text{LINE}}[\text{NUMBER IS} \\ \left[\begin{array}{l} \underline{\text{PLUS}} \\ + \\ - \end{array} \right] \left\{ \begin{array}{l} \text{一意名2} \\ \text{整数2} \end{array} \right\}]$$

機能

画面項目の画面上の垂直方向の座標を指定する。

構文規則

1. PLUS と+は同義語である。
2. LINE 句は基本画面記述項に対してだけ指定できる。

3. 整数 2 は 0 以外の符号なし整数でなければならない。
4. 一意名 2 は整数基本データ項目として記述されたものでなければならない。
一意名 2 には部分参照があってはならない。
5. 一意名 2 はファイル節、作業場所節、局所場所節または連絡節で定義されなければならない。

一般規則

1. LINE 句は、COLUMN 句とともに用いることで、画面項目の開始座標を確定する。LINE 句は、垂直座標を指定する。
2. PLUS, +, - のどれも指定されていない場合、画面項目の画面上の絶対行番号を指定する。
3. PLUS, +, - のどれかが指定された場合、直前に定義された画面項目を含む行に対する相対行番号を指定する。
4. ACCEPT 文または DISPLAY 文で参照される最初の画面項目の行番号は、相対行番号で指定してはいけない。
5. LINE 句を省略した場合、以下の規則が適用される。
 - ・ 先行する画面項目が定義されていない場合、LINE 1 が仮定される。
 - ・ 先行する画面項目が定義されている場合、先行する画面項目の行が仮定される。
6. NUMBER 指定を省略すると PLUS 1 が仮定される。

12.2.18 LOWLIGHT 句 (SCREEN SECTION)

形式

LOWLIGHT

機能

画面項目を画面上に最低明度で表示する。

もし 2 段階の明度しか使用できない場合、通常の明度と LOWLIGHT 句による明度とは同一となる。

構文規則

1. LOWLIGHT 句はどんな画面記述項に対しても指定できる。ただし、同じ画面項目に対し HIGHLIGHT 句と LOWLIGHT 句を同時に指定してはならない。

12.2.19 OVERLINE 句 (SCREEN SECTION)

形式

OVERLINE

機能

画面項目が画面上に表示されるとき、その画面項目の各文字の上側に横線を施す。

構文規則

1. OVERLINE 句はどんな画面記述項に対しても指定できる。

12.2.20 PICTURE 句 (SCREEN SECTION)

形式

$$\left. \begin{array}{l} \text{PICTURE} \\ \text{PIC} \end{array} \right\} \text{ IS picture文字列1}$$
$$\left. \begin{array}{l} \text{USING 一意名4} \\ \left. \begin{array}{l} \text{FROM 一意名5} \\ \text{定数1} \end{array} \right\} \text{] [TO 一意名6} \end{array} \right\}$$

機能

基本画面項目の一般的な性質と編集要求を記述する。また、この基本画面項目が画面に表示されるときに内容が表示されるべきデータ項目、およびこの基本画面項目に入力または更新されたときに内容が格納されるべきデータ項目を指定する。

構文規則

1. 各 PICTURE 句は、FROM、TO または USING 指定を含んでいなければならない。
2. 一意名 4、一意名 5 および一意名 6 は、ファイル節、作業場所節、局所場所節または連絡節で定義されなければならない。一意名 4、一意名 5 および一意名 6 には部分参照があってはならない。
3. picture 文字列にブール形式および外部浮動小数点形式を指定してはならない。
4. -UniObjGen オプションを指定した場合、定数 1 に日本語文字定数および UTF-8 で多バイトとなる文字を含む英数字定数を指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

一般規則

1. データが一つのデータ項目から編集されて画面に転記され、異なるデータ項目に格納されるとき、画面項目の PICTURE 句には FROM と TO 指定が共に記述されなければならない。
2. データが（ファイルのレコードの読み出し、更新、再書き込みのように）編集されて画面に転記され、同じデータ項目に格納されるとき、画面項目の PICTURE 句には USING 指定が記述されなければならない。
3. 一意名 4、一意名 5、一意名 6 および定数 1 は、PICTURE 句が用いる画面項目と同じ長さである必要はない。
データは MOVE 文の規則に従って転記される。MOVE 文の規則に合わない場合はエラーとする。
4. FROM 指定がある場合、次の規則が適用される。

(a) DISPLAY 文実行中、データは、picture 文字列 1 に従って編集された後、一意名 5 または定数 1 から転記され、画面上に表示される。表示は、LINE 句、COLUMN 句または DISPLAY 文の AT 指定で明示的または暗黙的に定義された画面上の位置から始まる。

(b) FROM 指定は、ACCEPT 文の実行中は何の意味も持たない。

5. TO 指定がある場合、次の規則が適用される。

(a) ACCEPT 文の実行が完了することによって、画面項目に対して入力されたデータは、picture 文字列 1 に従って編集された後、一意名 6 に転記される。画面から画面項目に転記される順序は、ACCEPT 文で参照される画面の画面記述項で画面項目が指定された順序である。

(b) TO 指定は、DISPLAY 文の実行中は何の意味も持たない。

6. USING 指定または FROM と TO 指定がある場合、次の規則が適用される。

(a) DISPLAY 文実行中、データは、一意名 4、一意名 5 または定数 1 から一般規則 4 の (a) に記述されたように転記される。

(b) ACCEPT 文の実行が完了することによって、画面項目に対して入力されたデータは、一意名 4 または一意名 6 に、一般規則 5 の (a) に記述されたように転記される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

使用方法

プログラムの記述にあたっては、非整数画面項目については、数字編集 PICTURE 文字列を指定することが推奨されている。つまり、省略された小数点を含む画面項目の表現はシステムに依存するので、記号「V」を使用しないことが望ましい。

12.2.21 PROMPT 句 (SCREEN SECTION)

形式

$$\text{PROMPT CHARACTER IS } \left\{ \begin{array}{l} \text{一意名1} \\ \text{定数1} \end{array} \right\}$$

機能

ACCEPT 文実行中に画面項目の空白の文字位置に印を付ける。

構文規則

1. PROMPT 句は入力フィールドおよび更新フィールドに指定できる。出力フィールドに指定された場合は、この PROMPT 句を無視する。
2. 一意名 1 は 1 文字の英字項目、英数字項目または日本語項目でなければならない。また、一意名 1 には添字付けおよび部分参照があってはならない。
3. 定数 1 は 1 文字の英数字定数、日本語文字定数または表意定数でなければならない。
-UniObjGen オプションを指定した場合、定数 1 に日本語文字定数および UTF-8 で多バイトとなる文字を含む英数字定数を指定してはならない。-UniObjGen オプションについては、マニュアル

「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

4. 一意名 1 はファイル節，作業場所節，局所場所節または連絡節で定義されなければならない。
5. PROMPT 句は，英字項目，英数字項目および日本語項目に指定できる。その他の項目に指定された場合は，この PROMPT 句を無視する。

一般規則

1. PROMPT 句と SECURE 句を同時に指定すると PROMPT 句は無視される。
2. ACCEPT 文が終了すると PROMPT 文字は空白に置き換わる。
3. 日本語フィールドに日本語項目または日本語文字定数以外を指定してはならない。
4. 日本語以外のフィールドに日本語項目または日本語文字定数を指定してはならない。

12.2.22 REQUIRED 句 (SCREEN SECTION)

形式

{ REQUIRED }
{ EMPTY-CHECK }

機能

入力フィールドまたは更新フィールドに，少なくとも一つの文字を入力しなければならないことを指定する。

構文規則

1. REQUIRED と EMPTY-CHECK は同義語である。
2. 集団項目に REQUIRED 句が指定された場合，その指定は集団に従属するすべての入力フィールドまたは更新フィールドに対して適用される。

一般規則

1. REQUIRED 句は，画面項目を入力させるあらゆる ACCEPT 文の実行中に有効である。ただし，入力受け付け中の画面項目に，カーソルが位置づけられていること。
この句が満足されるまで，終了キーのタイプは拒否され，カーソルはその画面項目の先頭に再度位置づけられる。
2. この句を満たすためには，英数字画面項目は少なくとも一つの空白（日本語項目の場合は倍角の空白）以外の文字を含み，数字画面項目は 0 でない値を持たなければならない。
3. 更新フィールドに REQUIRED 句が指定された場合，PICTURE 句の FROM または USING 指定で参照される一意名の内容またはオペレータの入力データのどちらかがこの句を満たすまで終了キーのタイプは拒否される。
4. 入力待ちを終了させるためにファンクションキーが用いられた場合，REQUIRED 句は必ずしも有効ではない。

5. FULL 句と REQUIRED 句を共に指定した場合、フィールドが利用者によって常に全部満たされることを要求するものとなる。
6. REQUIRED 句は、出力フィールドについては無視される。

12.2.23 REVERSE-VIDEO 句 (SCREEN SECTION)

形式

REVERSE-VIDEO

機能

デフォルトの前景色と背景色または指定された前景色と背景色を入れ換えた状態で画面項目を表示する。

構文規則

1. REVERSE-VIDEO 句はどんな画面記述項に対しても指定できる。

12.2.24 SECURE 句 (SCREEN SECTION)

形式

{ SECURE }
{ NO-ECHO }

機能

カーソルがこの句を指定した入力フィールド内にある間、オペレータがキーボードから入力したデータを画面上に表示しない。

構文規則

1. SECURE と NO-ECHO は同義語である。
2. SECURE 句はどんなフィールドに対しても指定できる。ただし、出力フィールドに指定された場合、この句は無視される。

一般規則

1. SECURE 句が集団のレベルに指定された場合、すべての入力フィールドおよび更新フィールドに対して適用される。
2. SECURE 句が指定されたときは、入力フィールドまたは更新フィールドに入力された文字は画面上に表示されない。

12.2.25 SIGN 句 (SCREEN SECTION)

形式

$$\begin{array}{c} \text{[SIGN IS]} \left\{ \begin{array}{l} \text{LEADING} \\ \text{TRAILING} \end{array} \right\} \\ \text{[SEPARATE CHARACTER]} \end{array}$$

機能

演算符号の表示の位置とモードを明示的に記述する必要がある場合、それらを指定する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.73 SIGN 句」

使用方法

プログラムの記述にあたっては、あらゆる符号付き数字画面項目の SIGN 句の記述で SEPARATE CHARACTER 句を指定することが推奨されている。

12.2.26 SIZE 句 (SCREEN SECTION)

形式

$$\text{SIZE IS} \left\{ \begin{array}{l} \text{一意名1} \\ \text{整数1} \end{array} \right\}$$

機能

画面項目のサイズを指定する。

構文規則

1. SIZE 句は数字項目、数字編集項目以外の画面項目に対して指定できる。ただし、集団項目に指定した場合、この句は無視される。
2. 一意名 1 は符号なしの整数項目でなければならない。一意名 1 には部分参照があってはならない。
3. 整数 1 は符号なしでなければならない。
4. 一意名 1 はファイル節、作業場所節、局所場所節または連絡節で定義されていなければならない。

一般規則

1. SIZE 句は画面項目のバイト数を指定する。
2. SIZE 句はゼロを指定すると無効となる。
3. SIZE 句で指定された値が画面項目で指定された PICTURE 句または VALUE 句の値より小さい場合、画面項目の左側の部分だけが画面に表示される。
4. SIZE 句で指定された値が画面項目で指定された PICTURE 句または VALUE 句の値より大きい場合、SIZE 句で指定された値は無効となる。

5. SIZE 句は最後に評価される。

12.2.27 UNDERLINE 句 (SCREEN SECTION)

形式

UNDERLINE

機能

画面項目が画面上に表示される時、その画面項目の各文字に下線を施す。

構文規則

1. UNDERLINE 句はどんな画面記述項に対しても指定できる。

12.2.28 USAGE 句 (SCREEN SECTION)

形式

[USAGE IS] { DISPLAY }
 { NATIONAL }

機能

計算機の記憶装置中のデータ項目の表現形式を指定する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

12.2.29 VALUE 句 (SCREEN SECTION)

形式

書き方 1 (データ項目)

VALUE IS 定数1

機能

定数画面項目の定数値を定義する。

構文規則

1. 定数 1 は英数字定数、日本語文字定数、記号文字または表意定数でなければならない。ただし、ALL 定数が指定されても無視され、定数の繰り返しは行われない。
2. VALUE 句は基本画面記述項にだけ指定できる。

-UniObjGen オプションを指定した場合、定数 1 に日本語文字定数および UTF-8 で多バイトとなる文字を含む英数字定数を指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

12.2.30 ZERO-FILL 句 (SCREEN SECTION)

形式

ZERO-FILL

機能

フィールド中のユーザ入力のなかった文字位置の値が空白の場合、ゼロで置き換える。

構文規則

1. ZERO-FILL 句は英字または英数字の入力フィールドおよび更新フィールドに指定できる。

一般規則

1. ZERO-FILL 句はデータを画面項目から受け取り側作用対象のデータ項目へ転記するときに、値が空白の場合ゼロで置き換える。

12.3 手続き部 (SCREEN SECTION)

画面操作機能に関する手続き部の文は、ACCEPT 文と DISPLAY 文である。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

12.3.1 ACCEPT 文 (SCREEN SECTION)

形式

書き方 4 (画面節 (SCREEN SECTION)) (画面名)

ACCEPT 画面名1

$$\left[\begin{array}{l} \left\{ \begin{array}{l} \text{AT} \left\{ \begin{array}{l} \text{LINE NUMBER} \left\{ \begin{array}{l} \text{一意名1} \\ \text{整数1} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{COLUMN} \\ \text{COL} \end{array} \right\} \text{NUMBER} \left\{ \begin{array}{l} \text{一意名2} \\ \text{整数2} \end{array} \right\} \end{array} \right\} \\ \text{AT} \left\{ \begin{array}{l} \text{一意名3} \\ \text{整数3} \end{array} \right\} \end{array} \right\} \end{array} \right]$$

[ON EXCEPTION 無条件文1]
[NOT ON EXCEPTION 無条件文2]
[END-ACCEPT]

書き方 4 (画面節 (SCREEN SECTION)) (一意名)

ACCEPT 一意名1

$$\left[\begin{array}{l} \left\{ \begin{array}{l} \text{AT} \left\{ \begin{array}{l} \text{LINE NUMBER} \left\{ \begin{array}{l} \text{一意名2} \\ \text{整数1} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{COLUMN} \\ \text{COL} \end{array} \right\} \text{NUMBER} \left\{ \begin{array}{l} \text{一意名3} \\ \text{整数2} \end{array} \right\} \end{array} \right\} \\ \text{AT} \left\{ \begin{array}{l} \text{一意名4} \\ \text{整数3} \end{array} \right\} \end{array} \right\} \end{array} \right]$$

[ON EXCEPTION 無条件文1]
[NOT ON EXCEPTION 無条件文2]
[END-ACCEPT]

機能

データ項目の内容を画面に表示した後、画面からデータを取り込み、指定されたデータ項目に転記する。この ACCEPT 文の書き方は、画面節 (SCREEN SECTION) で定義された画面項目と関係し、画面操作機能を提供する。

構文規則

1. 画面名、一意名に指定できる属性を次に示す。

表 12-2 画面名と一意名に指定できる属性

属性	画面名と一意名 書き方 4 (画面名)	画面名と一意名 書き方 4 (一意名)
可変長集団項目	—	—
固定長集団項目	○	○
外部 10 進項目	○	○
内部 10 進項目	—	△
内部 2 進項目	—	△
外部浮動小数点項目	—	—
内部浮動小数点項目	—	—
数字編集項目	○	○
英字項目	○	○
英数字項目	○	○
英数字編集項目	○	○
外部ブール項目	—	—
内部ブール項目	—	—
指標データ項目	—	—
日本語項目	○	○
日本語編集項目	○	○
アドレスデータ項目	—	—
オブジェクト参照データ項目	—	—

(凡例)

○：指定可

△：集団項目が指定され、その基本従属項目の場合は無視される。

—：指定不可

2. 画面名 1 は、添字、部分参照されてはならない。書き方 4 (一意名) の一意名 1 は、部分参照されてはならない。一意名 1 が集団項目の場合、OCCURS 句を含んではならない。

書き方 4 (一意名) の一意名 1 は、強く型付けされた集団項目であってはならない。

3. 書き方 4 (画面名) の画面名 1 は少なくとも一つの入力フィールドまたは更新フィールドがなければならぬ。

4. 書き方 4 (画面名) の一意名 3 および書き方 4 (一意名) の一意名 4 は PIC 9(4) または PIC 9(6) のデータ項目でなければならない。また、一意名 3、一意名 4 は、部分参照されてはならない。

5. 書き方 4 (画面名) の整数 3 および書き方 4 (一意名) の整数 3 は、4 けたまたは 6 けたでなければならない。

6. 書き方 4 (画面名) の一意名 1 および一意名 2, 書き方 4 (一意名) の一意名 2 および一意名 3 は整数項目でなければならない。
書き方 4 (画面名) の一意名 1, 一意名 2 および一意名 3 は, 部分参照されてはならない。
7. 書き方 4 (画面名), 書き方 4 (一意名) の整数 1 および整数 2 は符号なし整数でなければならない。ただし, 符号がある場合はその値の絶対値を有効とする。

一般規則

1. AT 指定は, ACCEPT 文の動作を開始する位置の画面上での絶対アドレスを与える。
2. ON EXCEPTION 指定がある場合, 無条件文 1 は ACCEPT 文の動作が異常終了したときにだけ実行される。
NOT ON EXCEPTION 指定がある場合, 無条件文 2 は ACCEPT 文の動作が正常終了したときにだけ実行される。
3. AT 指定がない場合, ACCEPT 文の動作は画面の 1 行 1 カラム目から開始される。
4. 書き方 4 (画面名) の一意名 3 および整数 3, 書き方 4 (一意名) の一意名 4 および整数 3 が 4 けたの場合, 最初の 2 けたは行を, 次の 2 けたはカラムを表す。6 けたの場合は最初の 3 けたが行を, 次の 3 けたがカラムを表す。
5. AT 指定の行およびカラムがともにゼロの場合は, 直前に実行された ACCEPT 文または DISPLAY 文の終了した場所の行, 次のカラムから実行される。
6. AT 指定の行がゼロで AT 指定のカラムがゼロでない場合は, 直前に実行された ACCEPT 文または DISPLAY 文が終了した行の指定したカラムから実行される。
7. AT 指定のカラムがゼロで AT 指定の行がゼロでない場合は, 直前に実行された ACCEPT 文または DISPLAY 文の終了した場所の次のカラムの指定した行から実行される。
8. カラム位置が画面の範囲内に入らない場合, そのフィールドは次の行に継続される。ただし, 次の行に継続すると行番号が画面の範囲内に入らなくなる場合は, 継続されない。
9. 行番号が画面の範囲内に入らない場合, 画面の範囲外のフィールドは無視される。
10. 作用対象が一意名で, AT 指定がある ACCEPT 文は, 書き方 4 (一意名) として扱われる。
ただし, AT 指定がなくても, 特殊名段落で CONSOLE IS CRT が指定されている場合は, 書き方 4 (一意名) として扱われる。
11. 書き方 4 (一意名) で, 一意名 1 が集団項目の場合, その基本項目はデータ部の記述に書いた順番に画面上に表示され, ACCEPT 操作が受け付けられる。
ただし, 基本項目の属性が外部 10 進項目, 数字編集項目, 英字項目, 英数字項目, 英数字編集項目, 日本語項目, 日本語編集項目でないか, 無名項目 (FILLER) であるときは, その項目の長さだけ, 空白が表示され, ACCEPT 操作は無視される。
すべての基本項目が ACCEPT 操作を無視される項目であるときに ON EXCEPTION 指定がされていた場合, ON EXCEPTION の後に記述された無条件文 1 が実行される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

12.3.2 DISPLAY 文 (SCREEN SECTION)

形式

書き方 3 (画面節 (SCREEN SECTION)) (画面名)

DISPLAY 画面名1

$$\left[\left\{ \begin{array}{l} \text{AT} \left\{ \begin{array}{l} \text{LINE NUMBER} \left\{ \begin{array}{l} \text{一意名1} \\ \text{整数1} \end{array} \right\} \\ \text{COLUMN} \\ \text{COL} \end{array} \right\} \text{NUMBER} \left\{ \begin{array}{l} \text{一意名2} \\ \text{整数2} \end{array} \right\} \\ \text{AT} \left\{ \begin{array}{l} \text{一意名3} \\ \text{整数3} \end{array} \right\} \end{array} \right\} \right]$$

[END-DISPLAY]

書き方 3 (画面節 (SCREEN SECTION)) (一意名/定数)

DISPLAY {

$$\left[\left\{ \begin{array}{l} \text{AT} \left\{ \begin{array}{l} \text{LINE NUMBER} \left\{ \begin{array}{l} \text{一意名2} \\ \text{整数1} \end{array} \right\} \\ \text{COLUMN} \\ \text{COL} \end{array} \right\} \text{NUMBER} \left\{ \begin{array}{l} \text{一意名3} \\ \text{整数2} \end{array} \right\} \\ \text{AT} \left\{ \begin{array}{l} \text{一意名4} \\ \text{整数3} \end{array} \right\} \end{array} \right\} \right] \dots$$

[END-DISPLAY]

機能

指定されたデータ項目から画面にデータを転記する。この DISPLAY 文の書き方は、画面節 (SCREEN SECTION) で定義された画面項目と関係し、画面操作機能を提供する。

構文規則

1. 画面名、一意名に指定できる属性は、ACCEPT 文と同じである。「表 12-2 画面名と一意名に指定できる属性」を参照のこと。
画面名 1 は、添字、部分参照されてはならない。また、一意名 1 は、部分参照されてはならない。
書き方 3 (一意名/定数) の一意名は、強く型付けされた集団項目であってはならない。
2. 書き方 3 (画面名) の画面名 1 は、少なくとも一つの出力フィールドまたは更新フィールドがなければならない。
3. 書き方 3 (画面名) の一意名 3 および書き方 3 (一意名/定数) の一意名 4 は PIC 9(4) または PIC 9(6) のデータ項目でなければならない。また、一意名 3 および一意名 4 は部分参照されてはならない。
4. 書き方 3 (画面名) の整数 3 および書き方 3 (一意名/定数) の整数 3 は、4 けたまたは 6 けたでなければならない。

5. 書き方 3 (一意名/定数) の一意名 1 に画面名を指定してはならない。
6. 書き方 3 (一意名/定数) の定数 1 にブール定数, 浮動小数点定数, 小数点を含む数字定数を指定してはならない。
7. 書き方 3 (画面名) の一意名 1 および一意名 2, 書き方 3 (一意名/定数) の一意名 2 および一意名 3 は整数項目でなければならない。
書き方 3 (画面名) の一意名 1, 一意名 2 および一意名 3 は部分参照されてはならない。
8. 書き方 3 (画面名), 書き方 3 (一意名/定数) の整数 1 および整数 2 は符号なし整数でなければならない。ただし, 符号がある場合は, その値の絶対値を有効とする。
9. 書き方 3 (一意名/定数) の一意名 1 は, 部分参照してはならない。
10. -UniObjGen オプションを指定した場合, 書き方 3 (一意名/定数) の定数 1 に日本語文字定数および UTF-8 で多バイトとなる文字を含む英数字定数を指定してはならない。-UniObjGen オプションについては, マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

一般規則

1. AT 指定は, DISPLAY 文の動作を開始する位置の画面上での絶対アドレスを与える。
2. AT 指定がない場合, DISPLAY 文の動作は, 画面の 1 行 1 カラム目から開始される。
また, 書き方 3 (一意名/定数) で 2 番目以降に指定した一意名 1 または定数 1 に AT 指定がない場合, おおの, 前のデータ項目の後に残された現カーソル位置から始まる。
3. 書き方 3 (画面名) の一意名 3 および整数 3, 書き方 3 (一意名/定数) の一意名 4 および整数 3 が 4 けたの場合, 最初の 2 けたは行を, 次の 2 けたはカラムを表す。6 けたの場合は最初の 3 けたが行を, 次の 3 けたがカラムを表す。
4. AT 指定の行およびカラムがともにゼロの場合は, 直前に実行された ACCEPT 文または DISPLAY 文が終了した行の次のカラムから実行される。
5. AT 指定の行がゼロで, AT 指定のカラムがゼロでない場合は, 直前に実行された ACCEPT 文または DISPLAY 文が終了した場所の次の行の指定したカラムから実行される。
6. AT 指定のカラムがゼロで, AT 指定の行がゼロでない場合は, 直前に実行された ACCEPT 文または DISPLAY 文が終了した場所の次のカラムの指定した行から実行される。
7. カラム位置が画面の範囲内に入らない場合, そのフィールドは次の行に継続される。ただし, 次の行に継続すると行番号が画面の範囲内に入らなくなる場合は, 継続されない。
8. 行番号が画面の範囲内に入らない場合, 画面の範囲外のフィールドは無視される。
9. 書き方 3 (一意名/定数) の一意名 1 は USAGE DISPLAY でなければならない。
10. 書き方 3 (一意名/定数) の定数 1 が以下の表意定数の場合, 特別な効果を持つ。
SPACE: 指定したカーソル位置から画面の終わりまでを消去する。
LOW-VALUE: カーソルを指定した位置へ移動する。

11. 作用対象が一意名または定数で、AT 指定がある DISPLAY 文は、書き方 3（一意名／定数）として扱われる。ただし、AT 指定がなくても、特殊名段落で CONSOLE IS CRT が指定されている場合は、書き方 3（一意名／定数）として扱われる。
12. 書き方 3（一意名／定数）で、一意名 1 が集団項目の場合、その基本項目はデータ部の記述に書いた順番に画面上に表示される。
ただし、基本項目の属性が外部 10 進項目、数字編集項目、英字項目、英数字項目、英数字編集項目、日本語項目、日本語編集項目でないときは、これを英数字項目として、その項目の長さだけ表示される。
また、基本項目が無名項目（FILLER）であるときは、属性によらず、項目の長さだけ、空白が表示される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

13

画面節 (WINDOW SECTION) による画面機能

この章では、画面節 (WINDOW SECTION) を用いて画面データをディスプレイに表示したり、ユーザーが入力したデータをディスプレイから読み込んだりする機能について説明する。

この機能は、Linux COBOL2002 では使用できない。Linux COBOL2002 でこの機能に関する次に示す構文を記述した場合、重大エラーを出力する。

- WINDOW SECTION
- ACCEPT 文 (書き方 5)
- CANCEL 文 (書き方 2)
- DISPLAY 文 (書き方 4, 書き方 5)
- ERASE 文
- EXIT 文 (書き方 7)
- REPLY 文
- SET 文 (書き方 13~書き方 16)
- USE 文 (書き方 5, 書き方 6)
- WAIT 文

13.1 概説 (WINDOW SECTION)

画面節 (WINDOW SECTION) による画面機能は、データ部の画面節 (WINDOW SECTION) で画面の各フィールドの位置や属性を指定し、ACCEPT、DISPLAY または REPLY 文によって簡単にデータの入出力する。

画面節 (WINDOW SECTION) では画面の各フィールドをレベル構造を用いて指定する。つまり、上位のレベルで画面の行番号を指定し、下位のレベルでその行に含まれるフィールドのコラム番号と属性を指定する。手続き部では REPLY 文などによって、一つまたは幾つかのフィールドを一度に表示したり入力したりする。

画面入出力機能の簡単なプログラム例を次に示す。

[プログラム例]

```
DATA      DIVISION.
WORKING-STORAGE SECTION.
77 TANKA PIC 9(5).
WINDOW SECTION. ... (1)
01 GAMEN. ... (2)
  02 LINE 5. ... (3)
    03 OUTPUT COLUMN 15          PIC N(3)
      VALUE NC'商品名'. ... (4)
    03 SHOHIN INPUT COLUMN 30    PIC X(10). ... (5)
  02 LINE 8.
    03 OUTPUT COLUMN 15          PIC N(3)
      VALUE NC'数量'.
    03 SURYO INPUT COLUMN 30     PIC 9(5). ... (6)
  02 LINE 11.
    03 OUTPUT COLUMN 15          PIC N(3)
      VALUE NC'金額'.
    03 KINGAKU OUTPUT COLUMN 30  PIC ¥(11)
      VALUE SPACE. ... (7)
PROCEDURE DIVISION.
  ERASE ALL WINDOW. ... (8)
  REPLY GAMEN UPON WINDOW. ... (9)
  COMPUTE KINGAKU = TANKA * SURYO. ... (10)
  DISPLAY KINGAKU UPON WINDOW. ... (11)
```

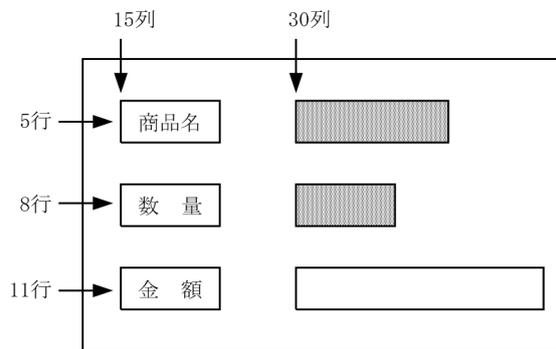
- (1)画面の各フィールドは画面節で指定する。
- (2)画面全体に名前を付ける。
- (3)画面の行番号 (5 行目) を指定する。
- (4)5 行, 15 列の位置に出力フィールドを指定する。このフィールドには「商品名」を表示する。
- (5)5 行, 30 列の位置に入力フィールドを指定する。このフィールドから商品名を入力する。
- (6)8 行, 30 列の位置に入力フィールドを指定する。このフィールドから数量を入力する。
- (7)11 行, 30 列の位置に出力フィールドを指定する。このフィールドには金額を編集して表示する。
- (8)画面全体を消去する。
- (9)初期画面を表示して, 商品名と数量を入力する。つまり, 入力した商品名の単価を, TANKA に求める処理, 出力フィールドをすべて表示した後, 入力フィールドヘデータを入力する。

(10)画面節のデータ名は、一般の文で直接参照できる。

(11)COMPUTE 文で計算した金額を表示する。

(例)

- 次の形式の初期画面を表示する。



(凡例) □ 出力フィールド ■ 入力フィールド

- 商品名と数量のデータを入力する。
- 商品名に対応する単価を求め,
金額 = 単価 × 数量
を計算して画面のフィールドに表示する。

13.2 言語の基本概念 (WINDOW SECTION)

13.2.1 画面入出力機能特殊レジスタ

画面入出力機能特殊レジスタ (WINDOW-STATUS, WINDOW-KEY, WINDOW-CURSOR および WINDOW-KEYCODE) は、画面節 (WINDOW SECTION) があるプログラムごと (プログラム単位, 利用者定義関数単位, ファクトリ単位およびオブジェクト単位) に生成する。

画面入出力機能特殊レジスタは大域属性を持つ静的データである。

(1) WINDOW-STATUS 特殊レジスタ (WINDOW SECTION)

WINDOW-STATUS (終了状態) は画面節に対して作成する特殊レジスタの名前であり、暗に次の記述を持つ。

```
01 WINDOW-STATUS PIC 9(2).
```

WINDOW-STATUS には、ACCEPT 文などの画面入出力文の終了状態が入る。

次に終了状態の値とその意味を示す。

表 13-1 終了状態の値とその意味

値	意味
00	画面入出力文の実行は正常終了した。
01	フィールド状態表示子 (STATUS-n) がオンになる条件が発生した。

(注意)

フィールド状態表示子は、入出力を実行した各フィールドの特別な状態を示すものであり、入力データエラーなどを検出した場合にオンになる。この表示子はフィールド状態条件を指定した IF 文で判定できる。ただし、このシステムでは常に 00 になる。フィールド状態条件の詳細については、「13.5.1 フィールド状態条件 (WINDOW SECTION)」を参照のこと。

(2) WINDOW-KEY 特殊レジスタ (WINDOW SECTION)

WINDOW-KEY (終了キー) は、画面節に対して作成する特殊レジスタの名前であり、暗に次の記述を持つ。

```
01 WINDOW-KEY PIC 9(2).
```

WINDOW-KEY には、ACCEPT または REPLY 文を実行してデータを入力したときに、オペレータが押した終了キーの種類を示す値が入る。

次に終了キーの値とその意味を示す。

表 13-2 終了キーの値とその意味

値	意味
00	[Enter] [実行]
01	[PF1]
:	:
24	[PF24]
25	未使用
:	:
99	未使用

注 1

SET 文でバックカーソルキーに設定したファンクションキー PFn は、終了キーとして使用できない。

注 2

ユーザポップアップ HELP 動作中は、ヘルプキーに設定したファンクションキー PFn は、終了キーとして使用できない。

(例) WINDOW-KEY の参照

```
ACCEPT GAMEN FROM WINDOW.  
EVALUATE WINDOW-KEY  
  WHEN 1 THRU 5 PERFORM SHORI  
  WHEN 6 THRU 9 PERFORM HYOJI  
  WHEN 10      PERFORM TOKUI  
  WHEN OTHER   PERFORM OWARI  
END-EVALUATE.
```

(3) WINDOW-KEYCODE 特殊レジスタ (WINDOW SECTION)

UNIX COBOL2002 では、WINDOW-KEYCODE 特殊レジスタ (WINDOW SECTION) は使用できない。

WINDOW-KEYCODE (終了キーコード) は、画面節に対して作成される特殊レジスタの名前であり、暗に次の記述を持つ。

```
01 WINDOW-KEYCODE PIC X(2).
```

WINDOW-KEYCODE には、WINDOW SECTION の ACCEPT 文および REPLY 文を実行してデータを入力したときにオペレータが押した終了キーのコードが入る。

通常は[Enter]キー、[実行]キーおよび[PF]キーを終了キーとして使用できるが、CBLADTRM サービスルーチンを使用すると任意のキーを終了キーとして追加できる。追加された終了キーを押した場合は、押

した終了キーに対応するコードが設定される。終了キーのコードが1バイトコードの場合は、WINDOW-KEYCODE 特殊レジスタの前1バイトに終了キーのコードが設定され、後ろの1バイトには0 (X'00') が設定される。

CBLADTRM サービスルーチンの詳細については、マニュアル「COBOL2002 ユーザーズガイド」を参照のこと。

(4) WINDOW-CURSOR 特殊レジスタ (WINDOW SECTION)

WINDOW-CURSOR (終了カーソル位置) は画面節に対して作成される特殊レジスタの名前であり、次の記述を持つ。

```
01 WINDOW-CURSOR PIC 9(5).
```

WINDOW-CURSOR には、ACCEPT または REPLY 文を実行したときに、入力操作を終了した時点の画面上カーソル位置 (0~19,199) を設定する。

(注意)

画面上の1行1カラム目のカーソル位置を0とする。

13.2.2 データ項目のサイズ (WINDOW SECTION)

画面節のデータ項目は、PICTURE 句と SIGN 句の一般的な規則によってサイズが決まる。

日本語項目に LETTER IS FREE 句の指定がある場合も、指定がない場合と同様のサイズとなる。

LETTER IS FREE 句は、日本語項目に日本語と英数字のデータが混在することを示す。

(例) 日本語のサイズ

PICTURE 文字列	項目のサイズ (バイト)	
	FREE 句なし	FREE 句あり
N(10)	20	20
N(11)	22	22

13.2.3 画面フィールドの種類とそのサイズ (WINDOW SECTION)

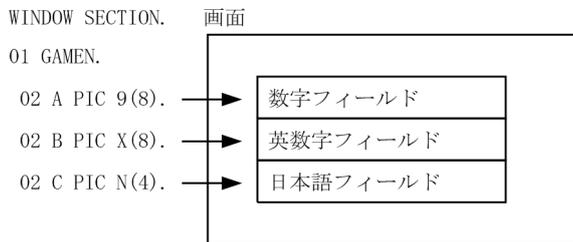
画面フィールドは、画面節で指定したデータ項目の PICTURE 句に従って次の3種類に分類される。

- 数字フィールド：数字 (外部 10 進) および数字編集項目
- 英数字フィールド：英字, 英数字, 英数字編集項目

- 日本語フィールド：日本語、日本語編集項目

これらのフィールドは、各データ項目と1対1に対応する。つまり、画面節でデータ項目を指定すると、それが画面フィールドとなる。また、不当なデータは入力できない。

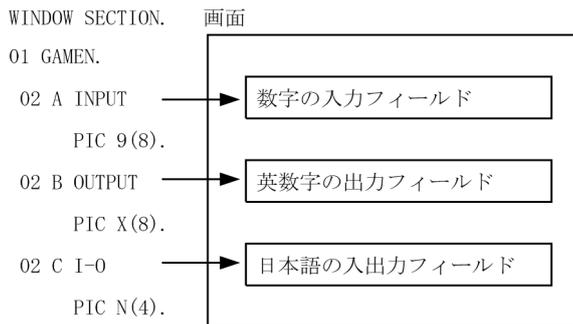
(例) データ項目とフィールドの関係



また、各フィールドはその使い方によって次のように分類できる (FIELD TYPE 句で指定する)。

- 入力フィールド：データを入力する。
- 出力フィールド：データを表示する。
- 入出力フィールド：データを表示した後、入力する。

(例) フィールドの使い方による分類



これらのフィールドのサイズは、データ項目のサイズに等しい場合と異なる場合がある。サイズの決め方を次に示す。ただし、拡大文字の指定がある場合は、「表 13-3 フィールドのサイズの決め方」に示すフィールドのサイズの2倍となる。USING 句と拡大文字の指定が同時にある場合は、USING 句で求まるフィールドのサイズの2倍となる。

(1) 数字フィールドのサイズ

PICTURE 文字列の「S」と「V」を含めてフィールドのサイズに数える。ただし、「V」が文字列の右端にあるときはフィールドのサイズに数えない。

(注意 1)

この規則は符号と小数点をデータとして入力するときに有効である。

(注意 2)

PICTURE 文字列には文字「P」を書いてはならない。

(例) 数字フィールドのサイズ

PICTURE 文字列	SIGN SEPARATE	データ項目のサイズ (バイト)	フィールドのサイズ (バイト)
9(6)	なし	6	6
S9(6)	なし	6	7
S9(6)V	なし	6	7
S9(5)V9	なし	6	8
S9(5)	あり	6	6
S9(5)V9	あり	7	8
¥,¥9.9	—	8	8

(2) 英数字フィールドのサイズ

フィールドのサイズはデータ項目のサイズに等しい。

(例) 英数字フィールドのサイズ

PICTURE 文字列	データ項目のサイズ (バイト)	フィールドのサイズ (バイト)
X(20)	20	20
X(4)BX(5)	10	10

(3) 日本語フィールドのサイズ

フィールドのサイズは、PICTURE 文字列の「N」と「B」はそれぞれ2バイト（日本語1文字）として計算した値に等しい。つまり、LETTER IS FREE 句の有無に関係なくフィールドのサイズは同じである。

(例) 日本語フィールドのサイズ

PICTURE 文字列	LETTER IS FREE	データ項目のサイズ (バイト)	フィールドのサイズ (バイト)
N(10)	なし	20	20
N(11)	なし	22	22
N(3)BN(4)	なし	16	16
N(10)	あり	20	20
N(11)	あり	22	22

LETTER IS FREE 句の有無によってデータ項目のサイズおよびフィールドのサイズは変わらない。

規則 (1) ~ (3) のまとめを次に示す。

表 13-3 フィールドのサイズの決め方

フィールドの種類	フィールドのサイズ	
数字	数字項目	PICTURE 文字列の文字「9」および「S」と「V」の個数。ただし、右側の「V」はフィールドのサイズに数えない。
	数字編集項目	PICTURE 文字列の各文字の個数。つまり、データ項目のサイズに等しい。
英数字	PICTURE 文字列の各文字の個数。つまり、データ項目のサイズに等しい。	
日本語	PICTURE 文字列の文字「N」と「B」の個数×2。つまり、データ項目のサイズに等しい。	

13.3 環境部 (WINDOW SECTION)

13.3.1 特殊名段落 (SPECIAL-NAMES) (WINDOW SECTION)

形式

SPECIAL-NAMES.
[機能名1 IS 呼び名3]...

機能

特殊名段落は、このシステムが定めた機能名と利用者が使う呼び名との関係、通貨編集用に使う文字およびコンマと小数点の機能の交換を指定する。

一般規則

1. 機能名 1 として WINDOW, PF_n (n:1~24) を書ける。
2. WINDOW は画面の名前である。WINDOW に関係づけた呼び名 3 は、ACCEPT 文、DISPLAY 文または REPLY 文で使用できる。
3. PF_n はそれぞれ対応するファンクションキー[F_n]を意味する。PF_n に関連づけた呼び名 3 は USE 文または SET 文で使用できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落 (SPECIAL-NAMES)」

13.3.2 入出力管理段落 (WINDOW SECTION)

(1) APPLY WINDOW-CONTROL-1 句

形式

APPLY WINDOW-CONTROL-1

このシステムでは APPLY WINDOW-CONTROL-1 句を覚え書きとみなす。

機能

SET (FIELD) 文を使用してフィールドの属性を変更するときに、画面内容の変更を SET 文の実行と同時にしないことを指定する。

一般規則

1. APPLY WINDOW-CONTROL-1 句を指定した場合、SET 文はフィールド属性の変更情報を内部的に退避し、次の画面入出力文 (ACCEPT 文、DISPLAY 文、ERASE 文、REPLY 文) で画面に反映させる (画面入出力文は SET 文で指定したフィールドを含まなくてもよい)。

2. この句は、最外側のプログラムで指定されたときだけ有効となり、内側のプログラムで指定されたときは無視される。

13.4 データ部 (WINDOW SECTION)

画面節 (WINDOW SECTION) で画面の様式を定義する。画面に対する入出力をする場合は、必ず画面節を定義しなければならない。

画面節はプログラム定義、利用者定義関数、ファクトリ定義およびインスタンス定義にだけ指定できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

13.4.1 データ記述項 (WINDOW SECTION)

形式

書き方 1

01 データ名
[FIELD TYPE句]
[IS GLOBAL]
[LINE NUMBER句] (書き方3) ※
[USAGE句].

書き方 2

レベル番号 [データ名]
[FIELD TYPE句]
[LINE NUMBER句] (書き方3) ※
[REPEAT LINE句
[RULE LINE IS TABLE]]
[REPEAT COLUMN句]
[USAGE句].

書き方 3

レベル番号 [データ名]
[BLANK WHEN ZERO句]
[BLINK句]
[COLOR句]
[HIGH-INTENSITY句]
[SECRET句]
COLUMN NUMBER句 (書き方3) ※
[ENTER-CHECK句]
[FIELD TYPE句]
[HELP WINDOW句]
[JUSTIFIED句]
[LETTER句]
[LINE NUMBER句] (書き方3) ※
PICTURE句
[RESET句]
[REVERSE句]
[RULE LINE句] (書き方1, 2) ※
[SIGN句]
[SOURCE句]
[VALUE句]
[USAGE句]
[USING句].

書き方 4

77 [データ名]
LINE NUMBER句 (書き方4) ※
COLUMN NUMBER句 (書き方4) ※
[BLINK句]
[COLOR句]
[REVERSE句]
[RULE LINE句] (書き方2) ※
[HIGH-INTENSITY句]
[SECRET句]
[LETTER句]. (書き方2) ※

書き方 5

レベル番号 [データ名]
[LINE NUMBER句] (書き方4) ※
COLUMN NUMBER句 (書き方4) ※
[BLINK句]
[COLOR句]
[REVERSE句]
[RULE LINE句] (書き方2) ※
[HIGH-INTENSITY句]
[SECRET句]
[LETTER句]. (書き方2) ※

注※

各句のそれぞれの書き方を示す。

書き方 6

88 条件名 VALUE句.

機能

データ記述項は、データ項目と画面のフィールドの性質を指定する。

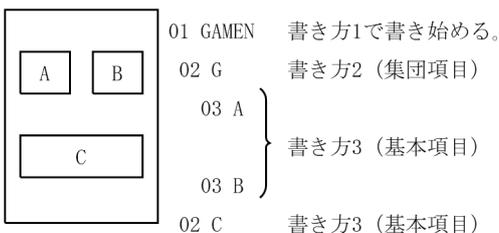
規則

1. データ名の後に書く各句の順序は任意である。
2. 書き方 1 と書き方 2 は集団項目を指定する。書き方 2 のレベル番号は 02~48 でなければならない。
3. 書き方 3 は基本項目を指定する。書き方 3 のレベル番号は 02~49 でなければならない。
4. 書き方 4 は 77 レベル記述項を指定する。
5. 書き方 5 は PICTURE 句のない基本項目を指定する。書き方 5 のレベル番号は 02~49 でなければならない。PICTURE 句のない基本項目および PICTURE 句のない基本項目だけから成る集団項目は、77 レベル記述項と同じ扱いになる。
6. 画面のフィールドは次の規則に従って 2~3 段階の階層で記述する。
 - 1 画面のフィールドの全部または一部を記述する最初の記述項は書き方 1 でなければならない。
 - 書き方 1 の記述項に続けて、書き方 2, 3 の記述項を書ける。
 - 書き方 2 の記述項を書いたときは、それに続けて少なくとも一つの書き方 3 の記述項を書くこと。

(注意)

書き方 2 の記述項の下位項目に別の書き方 2 の記述項を書いてもよい。この場合は 4 段階以上の階層になる。ただし、2~3 段階の階層で記述するのが一般的な使い方である。

(例) 階層構造を用いた画面の定義



フィールドは 3 段階 (A, B) または 2 段階 (C) の階層で記述する。

7. 書き方 2 の規則を次に示す。
 - LINE NUMBER 句と REPEAT LINE 句を同時に指定してはならない。
 - RULE LINE 句を含む記述項には、REPEAT LINE 句も指定すること。
この場合の RULE LINE 句は RULE LINE TABLE 句だけ指定できる。
8. 書き方 3 の各句は次に示す組み合わせで指定すること。なお、表にない句の組み合わせは任意である。

表 13-4 書き方 3 の句の組み合わせ

句の組み合わせ		COLOR	HIGH-INTENSITY	LETTER		RESET	REVERSE	RULE LINE	SECRET
				FREE	FREE 以外				
COLOR		×	×	○	○	○	○	○	×
HIGH-INTENSITY		×	×	○	○	○	○	○	×
LETTER	FREE	○	○	×	×	×	○	○	○

句の組み合わせ		COLOR	HIGH-INTENSIT Y	LETTER		RESET	REVERSE	RULE LINE	SECRET
				FREE	FREE 以外				
	FREE 以外	○	○	×	×	×	○	○	○
RESET		○	○	×	×	×	○	○	○
REVERSE		○	○	○	○	○	×	○	○
RULE LINE		○	○	○	○	○	○	×	○
SECRET		×	×	○	○	○	○	○	×

(凡例)

- ：組み合わせ可能
- ×

9. 書き方 4 および 5 の規則を次に示す。

- 書き方 4 の記述項には、COLUMN NUMBER 句と LINE NUMBER 句およびその他の句を少なくとも一つは書くこと。
- 書き方 4, 5 の各句は「表 13-5 書き方 4 および 5 の句の組み合わせ」に示す組み合わせで指定すること。
- 書き方 4, 5 は、画面に引くけい線やフィールドの属性を指定するものである。このため、データ項目には記憶域を割り当てない。したがって、書き方 4, 5 のデータ名は DISPLAY (WINDOW) 文または ERASE 文だけで参照できる。
- 書き方 5 の記述項の上位項目に REPEAT LINE 句、REPEAT COLUMN 句があってはならない。また、下位項目があってもならない。

表 13-5 書き方 4 および 5 の句の組み合わせ

句の組み合わせ	COLOR	HIGH-INTENSITY	LETTER FREE 以外	REVERSE	RULE LINE	SECRET
COLOR	×	×	○	○	○	×
HIGH-INTENSITY	×	×	○	○	○	×
LETTER FREE 以外	○	○	×	○	○	○
REVERSE	○	○	○	×	○	○
RULE LINE	○	○	○	○	×	○
SECRET	×	×	○	○	○	×

(凡例)

- ：組み合わせ可能
- ×

10. 書き方 6 は条件変数に対応する条件名を指定する。書き方 6 の記述項は、書き方 5 の直後および下位項目が書き方 5 だけの集団項目の直後には指定できない。書き方 6 の規則については、標準仕様編を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

11. 画面節のデータを画面の入出力文 (ACCEPT 文, DISPLAY 文, USE 文, REPLY 文) で部分参照はできない。画面節のデータを画面の入出力文以外の文で部分参照はできる。
12. 画面節のデータ記述項には EXTERNAL 句を書けない。
13. 画面節のデータ記述項に ADDRESSED 句を書くことはできない。また、画面節のデータ記述項にアドレスデータ項目は指定できない。
14. 画面節のデータ記述項に PROPERTY 句, SAME AS 句, TYPE 句, TYPEDEF 句および USAGE OBJECT REFERENCE 句を指定してはならない。
15. 一つのフィールドおよび文字に対し、異なる属性を指定したときの動作は、保証しない。

13.4.2 BLANK WHEN ZERO 句 (WINDOW SECTION)

形式

BLANK WHEN ZERO

機能

BLANK WHEN ZERO (ゼロならば空白) 句は、項目の値がゼロのときにそれを空白にする。

規則

1. BLANK WHEN ZERO 句は、数字または数字編集の基本項目に指定できる。
2. BLANK WHEN ZERO 句を書いた項目に MOVE 文や SOURCE 句でデータを転記するとき、画面からデータ入力したときに項目の値がゼロのとき、この項目の値は空白になる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.7 BLANK WHEN ZERO 句」

13.4.3 BLINK 句 (WINDOW SECTION)

形式

BLINK

このシステムでは BLINK 句を覚え書きとみなす。

機能

BLINK（点滅）句は、フィールドのデータを点滅して表示する。

一般規則

1. BLINK 句は基本項目，77 レベル記述項および PICTURE 句のない基本項目に指定できる。

13.4.4 COLOR 句 (WINDOW SECTION)

形式

COLOR IS {
GREEN
WHITE
RED
BLUE
MAGENTA
SKY-BLUE
YELLOW
BLACK

機能

COLOR（色）句は、フィールドのデータの色を変えて表示する。

規則

1. COLOR 句は基本項目，77 レベル記述項および PICTURE 句のない基本項目に指定できる。
2. COLOR 句を書いた項目を表示したり入力したりすると，そのフィールドのデータは次の色で表示する。

GREEN：緑色

WHITE：白色

RED：赤色

BLUE：青色

MAGENTA：紫色

SKY-BLUE：空色

YELLOW：黄色

BLACK：黒色

なお，白黒のディスプレイに表示した場合はシステムに依存する。

3. COLOR 句の指定がないときは，システムで定義された文字の色となる。また，けい線の色と背景色は常にシステムで定義された色となる。

(例) COLOR 句の指定

```
01 GAMEN.  
02 A PIC N(3) VALUE N'エラー'  
   COLOR RED ~  
   :  
   DISPLAY A UPON WINDOW.
```

DISPLAY 文を実行すると、「エラー」の文字を赤色で表示する。このフィールドの色を再び変更するときは、SET 文を使うとよい。

13.4.5 COLUMN NUMBER 句 (WINDOW SECTION)

形式

書き方 3

COLUMN NUMBER IS
{ 整数4
PLUS 整数5
データ名 }

書き方 4

COLUMN NUMBER IS
整数6 [{ THROUGH } 整数7]
[{ THRU }]
[整数8 [{ THROUGH } 整数9]] ...

機能

COLUMN NUMBER (カラム番号) 句は、画面の行の上でのカラム番号を指定する。

一般規則

- 書き方 3 の COLUMN NUMBER 句は、基本項目、77 レベル記述項および PICTURE 句のない基本項目に指定する。
- 書き方 4 の COLUMN NUMBER 句は、77 レベル記述項および PICTURE 句のない基本項目に指定する。
- 各整数は符号のない 1~3 けたの数字でなければならない。整数は画面の行の上でのカラム番号を指定する。1 行の最初のカラム番号は 1 であり、最後のカラム番号は、Windows COBOL2002 では 160、UNIX COBOL2002 では 80 である。
- 書き方 3 の整数 5 は相対カラム番号を示す。なお、次の規則を適用する。
 - 整数 5 の値は Windows COBOL2002 では 1~159、UNIX COBOL2002 では 1~79 でなければならない。
 - 整数 5 を指定すると、そのカラム番号は直前のカラム番号に整数 5 の値を加えた位置になる。カラム番号は、Windows COBOL2002 では 160 けた、UNIX COBOL2002 では 80 けたを超えてはならない。
- 書き方 3 のデータ名は、カラム番号を可変に定義することを示す。なお、次の規則を適用する。
 - データ名は修飾してもよいが、添字付けしてはならない。

- データ名は整数項目でなければならない。
- データ名の値は Windows COBOL2002 では 1~160, UNIX COBOL2002 では 1~80 でなければならない。それ以外の値のとき、実行時にカラム番号不正で実行時エラーとなる。
- データ名を指定した項目の上位に、REPEAT LINE 句および REPEAT COLUMN 句があってはならない。

書き方 3

1. 整数 4 は、フィールドの左端のカラム番号を示す。

整数 4 の値は、Windows COBOL2002 では 0 または 1~160 であること。0 を指定した場合、そのフィールドは画面入出力文の対象外となる。UNIX COBOL2002 では 0 または 1~80 であること。0 を指定した場合、そのフィールドは画面入出力文の対象外となる。

(例) 0 を指定した場合

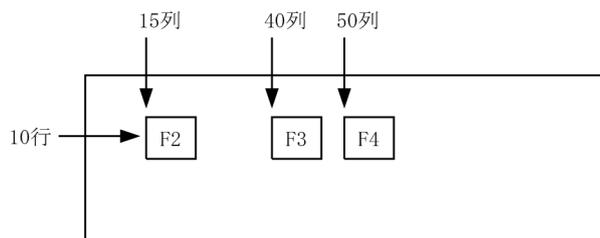
```
01 MAP0.
02 MAP1 LINE 10.
03 F1 COLUMN 10~. ... (1)
03 F2 COLUMN 15~.
02 MAP2 LINE 10.
03 F3 COLUMN 40~.
03 F4 COLUMN 50~.
```

(1) を 03 F1 COLUMN 0 にすると、

```
PROCEDURE DIVISION.
:
MOVE SPACE TO F1.      ... (2)
:
DISPLAY MAP0 UPON WINDOW. ... (3)
:
ACCEPT MAP1 FROM WINDOW. ... (4)
```

(2) F1 に SPACE が設定される。

(3) F1 は対象外のため表示されない。



(4) F1 は対象外のため F2 に対してデータを入力する。

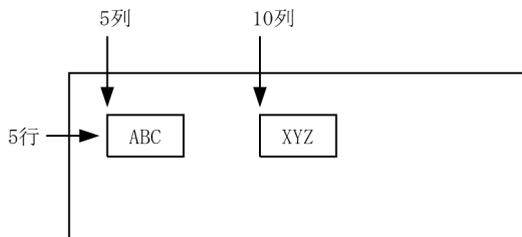
(例) 書き方 3 のデータ名の指定

```
WORKING-STORAGE SECTION.
77 I PIC 9(3).
:
WINDOW SECTION.
01 GAMEN LINE 5.
02 A OUTPUT COLUMN I PIC X(3).
```

```

      :
MOVE 5 TO I.
MOVE 'ABC' TO A.
DISPLAY GAMEN UPON WINDOW.
MOVE 10 TO I.
MOVE 'XYZ' TO A.
DISPLAY GAMEN UPON WINDOW.

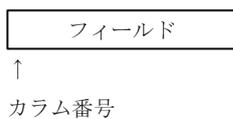
```



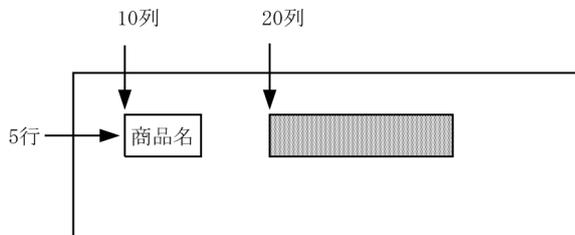
2. 一つのレベルの集団で、最初の COLUMN NUMBER 句には、絶対カラム番号を指定しなければならない。
3. 1 行中のフィールドは、任意の順序で書いてもよいが、その行の左のものから右のものへ順に指定するのが一般的な書き方である。なお、各フィールドは、重ならないように指定すること。
カラム番号の指定方法を次に示す。

図 13-1 書き方 3 の COLUMN 句のカラム番号

〔入力、出力及び入出力フィールド〕



(例) 書き方 3 の COLUMN NUMBER 句の指定



(凡例) 出力フィールド 入力フィールド

```

01 GAMEN LINE 5.
02 A OUTPUT COLUMN 10 PIC N(3) VALUE NC '商品名'.
02 B INPUT COLUMN 20 PIC X(15).
      :
REPLY GAMEN UPON WINDOW.

```

カラム番号は、フィールドの左端の文字位置を指定する。

書き方 4

1. 書き方 4 の COLUMN NUMBER 句は画面に引くけい線のコラム番号を指定する。この句と RULE LINE 句および LINE NUMBER 句を組み合わせると自由なけい線を引ける。さらに、特定のフィールドに関係しないフィールド属性を定義するときにも、その列指定を指定する。

2. 各整数の値は、Windows COBOL2002 では 1~160 でなければならない。ただし、整数 6 は 0 でもよい。UNIX COBOL2002 では 1~80 でなければならない。ただし、整数 6 は 0 でもよい。0 を指定した場合、そのフィールドは画面入出力文の対象外となる。また、次の大小関係を満足しなければならない。

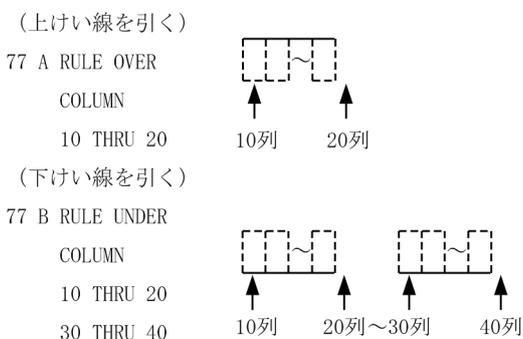
整数 6 < 整数 7 ≤ 整数 8 < 整数 9 . . .

3. けい線の引き方を次に示す。なお、整数 6, 7 に関する規則は、整数 8, 9 およびこれらの繰り返しにも適用する。なお、けい線指定の詳細については、「13.4.20 RULE LINE 句 (WINDOW SECTION)」を参照のこと。

(a) 整数 6 と整数 7 を指定した場合

- 横けい線 (上または下) は、整数 6 のコラム番号の左端から整数 7 のコラム番号の左端まで引く。

(例) 横けい線の指定

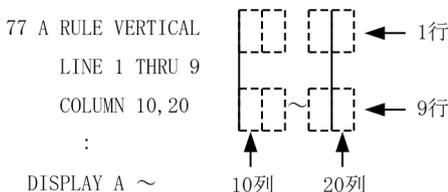


- 縦けい線は、整数 6 のコラム番号と整数 7 のコラム番号に引く。ただし、THROUGH 指定をして縦けい線を引くのは一般的な使い方ではない。

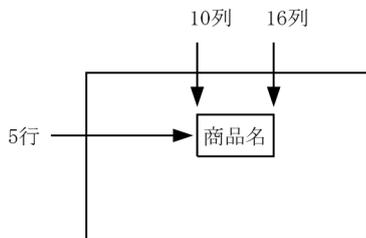
(b) 整数 6 だけを指定した場合

- 横けい線 (上または下) は、整数 6 のコラム番号に引く。ただし、THROUGH 指定を省略して縦けい線を引くのは一般的な使い方ではない。
- 縦けい線は、整数 6 のコラム番号に引く。縦けい線の行番号は LINE NUMBER 句で指定する。

(例) 縦けい線の指定



(例) PICTURE 句のない基本項目による横けい線の指定



```

01 GAMEN LINE 5.
02 A OUTPUT COLUMN 10 PIC N(3) VALUE NC '商品名'.
02 KEISEN-A COLUMN 10 THRU 16 RULE UNDER.
:
DISPLAY GAMEN UPON WINDOW.

```

表示結果

商品名

13.4.6 記述項名句 (WINDOW SECTION)

形式

書き方 1 (データ名)

データ名

機能

記述項名句は、記述するデータの名前を指定する。

一般規則

1. 01 レベルには、データ名を書かなくてはならない。
2. 02～49 レベルと 77 レベルには、データ名を書いてもよいし、省略してもよい。また、FILLER を書いてもよい。
3. 77 レベル記述項および PICTURE 句のない基本項目は、画面入出力文の DISPLAY (WINDOW) 文および ERASE (ATTRIBUTE) 文だけで参照できる。
4. 3.以外のデータ項目は、画面入出力文で参照できる。さらに、MOVE 文や ADD 文などの一般の文でも参照できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.22 記述項名句」

13.4.7 ENTER-CHECK 句 (WINDOW SECTION)

形式

ENTER-CHECK

機能

ENTER-CHECK (入力チェック) 句は、入力または入出力フィールドにオペレータがデータを入力したかどうかを検査する。

一般規則

1. ENTER-CHECK 句は、入力または入出力フィールドに対する基本項目に指定できる。
2. オペレータが、このフィールドに対し、未入力のまま送信動作 ([実行], [PFn]など) を要求した場合は、端末上で未入力エラーとなる。

13.4.8 FIELD TYPE 句 (WINDOW SECTION)

形式

FIELD TYPE IS { INPUT
OUTPUT
I-O }

機能

FIELD TYPE (フィールドの区分) 句は、画面のフィールドの区分 (入力, 出力または入出力) を指定する。

一般形式

1. FIELD TYPE 句は、PICTURE 句のない基本項目でなければ 01~49 のどんなレベルに書いてもよい。集団項目に書くと、その集団に従属する基本項目にその FIELD TYPE 句を適用する。PICTURE 句のない基本項目は、上位レベルの FIELD TYPE 句に関係なく、区分は OUTPUT とみなされる。
2. 上位と下位のレベルに同時に FIELD TYPE 句を書いてもよい。同時に書くと、下位のレベルの句が有効になる。

(例) FIELD TYPE 句の指定

```
01 A OUTPUT.  
02 B ~           :出力フィールド  
02 C INPUT ~    :入力フィールド  
02 D ~           :出力フィールド
```

3. 基本項目およびそれを含む集団項目に FIELD TYPE 句の指定がないと、その基本項目の区分は OUTPUT となる。
4. INPUT は、そのフィールドが入力フィールドであることを示す。

5. OUTPUT は、そのフィールドが出力フィールドであることを示す。

6. I-O は、そのフィールドが入出力フィールドであることを示す。

7. 画面入出力文の各フィールドに対する処理内容を「表 13-6 フィールドに対する処理内容」に示す。

表 13-6 フィールドに対する処理内容

文	区分		
	入力フィールド	出力フィールド	入出力フィールド
ACCEPT	データの入力	終了キーの入力	データの入力
DISPLAY	何もしない	データの表示	データの表示
REPLY	データの入力	データの表示および 終了キーの入力	データの表示および データの入力

13.4.9 GLOBAL 句 (WINDOW SECTION)

形式

IS GLOBAL

機能

データ名が大域名であることを指定する。大域名は、それを記述するプログラムに含まれるすべてのプログラムで使用できる。

一般規則

1. GLOBAL 句は 01 レベルにだけ指定できる。
2. 同じデータ部で同じデータ名が指定された二つのデータ項目に対するデータ記述項は、GLOBAL 句を含んではならない。
3. GLOBAL 句を使用して記述されたデータ名は、大域名とする。大域名に関連するすべての条件名は、大域名とする。
4. 大域名を記述するプログラムの中に直接または間接に含まれるプログラム文は、その名前を再度記述しないで、その名前を参照できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.30 GLOBAL 句」

13.4.10 HELP WINDOW 句 (WINDOW SECTION)

形式

HELP WINDOW IS データ名
HELP KEY IS {呼び名
PF_n* }
HELP LINE NUMBER IS
整数1 [{THROUGH } 整数2]
THRU
HELP COLUMN NUMBER IS
整数3 {THROUGH } 整数4
THRU

注※

PF1~PF24

機能

HELP WINDOW (HELP 画面) 句は、この句を書いたフィールドに対して、ユーザポップアップ HELP 画面名称、HELP キー、HELP 画面の表示位置および表示範囲を指定する。

一般規則

1. HELP WINDOW 句は、入力または入出力フィールドに指定できる。
2. HELP WINDOW 句は、この句を指定したフィールドに対するユーザポップアップ HELP 画面名称、ヘルプキー、HELP 画面の表示位置および表示範囲を指定する。
3. データ名は、画面節で定義された 01 レベルのデータ項目の名前でなければならない。この項目が HELP 画面となる。
4. HELP 画面は出力フィールドだけで構成されなければならない。
5. HELP KEY 句で指定したファンクションキーがヘルプキーとなる。
6. ヘルプキーとして設定したファンクションキーは、ユーザポップアップ HELP 動作中はバックカーソルキーおよび終了キーとして用いることはできない。
7. LINE 句および COLUMN 句は、HELP 画面の表示位置および表示範囲を指定する。各整数は、以下の式を満足しなければならない。

$$1 \leq \text{整数 } 1 < \text{整数 } 2 \leq 24,$$

$$1 \leq \text{整数 } 3 < \text{整数 } 4 \leq 80,$$

(注意)

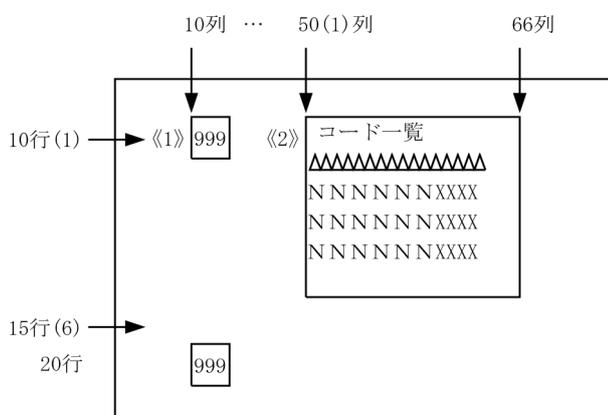
整数 2 を省略した場合は、整数 2=整数 1 となる。

8. ユーザポップアップ HELP は、この句を指定したフィールドに対する ACCEPT 文または REPLY 文発行時に動作する。

HELP WINDOW 句の使用例を次に示す。

(例) データの入力時に、その関連データの一覧を参照するヘルプ画面を表示するプログラム。

- GAMEN から商品コード 《1》 を入力時に、商品コード一覧を表示させる。
- [PF1]を押すとヘルプ画面 POP-MAP を 《2》 に表示する。
- 出力内容は、
 - 1 行, 1 列に見出し語「コード一覧」
 - 2 行, 1 列に空白 (16 文字)
 - 3 行, 1 列に日本語データ (6 文字)
 - 3 行, 13 列に文字データ (4 文字)
 これを 3 回繰り返す。
- 図中の括弧内の行列数は、相対的な位置を示す。
- プログラムは、ヘルプ画面の領域を定義するものと、ヘルプ画面の内容を定義するものとの二つから成る。



(例) HELP 画面の領域の定義

```

01 GAMEN.
02 商品コード LINE 10 COLUMN 10 PIC 9(3) INPUT
   HELP WINDOW IS POP-MAP
   HELP KEY IS PF1
   HELP LINE 10 THRU 15
   HELP COLUMN 50 THRU 66.
02 得意先コード LINE 20
   COLUMN 10 PIC 9(3) INPUT.

```

(例) HELP 画面の内容の定義

```

01 POP-MAP.
02 LINE 1 COLUMN 1 OUTPUT
   COLOR YELLOW REVERSE
   PIC N(5) VALUE NC 'コード一覧'.
02 LINE 2 COLUMN 1 OUTPUT
   COLOR WHITE REVERSE
   PIC X(16) VALUE SPACE.
02 REPEAT 3 TIMES.
03 LINE 3.
04 HD COLUMN 1 OUTPUT
   COLOR WHITE REVERSE
   PIC N(6).

```

04 CO COLUMN 13 OUTPUT
COLOR WHITE REVERSE
PIC X(4).

HELP 画面を定義する場合、次のことに注意する。

- 入力および入出力フィールドを定義してはならない。
- HELP 画面の表示位置は、HELP 句を指定したフィールドと重ならない位置に指定する。
- HELP 画面は、別のフィールドへカーソルが移動するときおよび終了キーが入力されたときに消える。

13.4.11 HIGH-INTENSITY 句 (WINDOW SECTION)

形式

HIGH-INTENSITY

機能

HIGH-INTENSITY (高輝度) 句は、フィールドのデータを高輝度で表示する。

一般規則

1. HIGH-INTENSITY は、基本項目、77 レベル記述項および PICTURE 句のない基本項目に指定できる。
2. HIGH-INTENSITY 句を書いた項目を表示したり入力したりすると、そのフィールドのデータは赤色で表示する。
なお、白黒のディスプレイに表示した場合はシステムに依存する。

13.4.12 JUSTIFIED 句 (WINDOW SECTION)

形式

$\left. \begin{array}{l} \text{JUSTIFIED} \\ \text{JUST} \end{array} \right\} \text{RIGHT}$

機能

JUSTIFIED (けた寄せ) 句は、項目にデータを収めるときに標準的でないけた寄せを指定する。

一般規則

1. JUSTIFIED 句は英字、英数字または日本語の基本項目に指定できる。
2. JUSTIFIED 句を書いた項目に、MOVE 文や SOURCE 句でデータを転記すると、画面からデータを入力したときに右端をそろえてデータを収める。
3. LETTER 句を指定した項目には、JUSTIFIED 句は指定できない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.36 JUSTIFIED 句」

13.4.13 LETTER 句 (WINDOW SECTION)

形式

書き方 1

$$\text{LETTER IS } \left\{ \begin{array}{l} \text{FREE} \\ \text{ALPHANUMERIC} \\ \text{KEIS} \end{array} \right\}$$

書き方 2

$$\text{LETTER IS } \left\{ \begin{array}{l} \text{WIDE} \\ \text{LONG} \\ \text{DOUBLE} \\ \text{EXTEND-1} \\ \text{EXTEND-2} \\ \text{EXTEND-3} \end{array} \right\}$$

機能

LETTER (文字) 句は、フィールドの文字の種類を指定する。

一般規則

書き方 1

1. このシステムでは KEIS の指定がない場合でも、日本語フィールドには全角文字以外の入力できないため、KEIS は覚え書きとする。
2. FREE 指定は、入力、出力または入出力フィールドに対する日本語項目に指定できる。
3. FREE 指定は、日本語フィールドに全角文字と半角文字のデータが混在することを示す。
4. FREE 指定を指定したデータ項目とフィールドのサイズは同じである。「13.2.2 データ項目のサイズ (WINDOW SECTION)」および「13.2.3 画面フィールドの種類とそのサイズ (WINDOW SECTION)」を参照のこと。

(例) データ項目とフィールドのサイズ

```
03 A INPUT PIC N(5)
    LETTER IS FREE ~
```

データ項目とフィールドのサイズは、同じ (10 バイト) である。

5. FREE 句を指定すると、その日本語フィールドには全角文字と半角文字の混在入力および表示ができる。FREE 句を指定しないと、その日本語フィールドに半角文字は入力できない。
6. ALPHANUMERIC 指定は入力または入出力フィールドに対する英数字フィールドを定義する基本項目に指定できる。

7. ALPHANUMERIC 句を指定した場合、そのフィールドには全角入力できない。また、挿入、削除などはすべて半角単位とする。
8. KEIS 指定は入力または入出力フィールドに対する日本語フィールドを定義する基本項目に指定できる。
9. FREE 指定と ALPHANUMERIC 指定または FREE 指定と KEIS 指定は同時に指定できない。

書き方 2

1. このシステムでは、EXTEND-1 (横 2 倍)、EXTEND-2 (縦 2 倍)、EXTEND-3 (縦横 2 倍) を覚え書きとし、指定なしとして処理する。
2. WIDE 指定は 77 レベル記述項、PICTURE 句のない基本項目および入力、出力および入出力フィールドに対する基本項目に指定できる。ただし、入力または入出力フィールドに対する数字フィールドを定義する基本項目には指定できない。
LONG および DOUBLE 指定は、77 レベル記述項、PICTURE 句のない基本項目および出力フィールドに対する基本項目に指定できる。
3. WIDE 指定は、入力、出力および入出力フィールドの文字を拡大して表示する。また、LONG および DOUBLE 指定は、出力フィールドの文字を拡大して表示することを示す。

WIDE : 横 2 倍

LONG : 縦 2 倍

DOUBLE : 横 2 倍 × 縦 2 倍

4. WIDE 指定、DOUBLE 指定を書く場合、次のことに注意すること。

数字項目の場合は、拡大文字の指定がないときのフィールドのサイズの 2 倍のフィールドとなる。その他の場合は、データ項目の 2 倍のサイズのフィールドとなる。

(例) 横 2 倍の拡大文字

売上報告書

03 A PIC N(5) LETTER IS WIDE VALUE NC '売上報告書'.
--

5. LONG 指定、DOUBLE 指定を書く場合、次のことに注意すること。

- 指定したデータ項目が次の行にも確保される。
- RULE LINE 句を同時に指定した場合、RULE LINE 句の指定がその行と次の行の両方にあるものとして処理される。

(例) 縦 2 倍の拡大文字

売上報告書

```
03 A LINE 1 COLUMN 30
    PIC N(5) LETTER LONG
    VALUE NC'売上報告書'.
```

上記の例のとき、LINE 2 にも同じ記述があるものとして処理される。

6. DOUBLE 指定には、4., 5.の両方の規則を適用する。

13.4.14 LINE NUMBER 句 (WINDOW SECTION)

形式

書き方 3

$$\text{LINE NUMBER IS } \left\{ \begin{array}{l} \text{整数4} \\ \text{PLUS 整数5} \\ \text{データ名} \end{array} \right\}$$

書き方 4

$$\begin{array}{l} \text{LINE NUMBER IS} \\ \text{整数6} \left[\begin{array}{l} \left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{整数7} \\ \left[\begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right] \text{整数9} \end{array} \right] \dots \end{array}$$

機能

LINE NUMBER (行番号) 句は、画面上での行番号を指定する。

一般規則

- 書き方 3 の LINE NUMBER 句は、01~49 のどんなレベルに書いてもよい。集団項目に書くと、その集団に従属するすべての基本項目にその LINE NUMBER 句を適用する。
- 書き方 4 の LINE NUMBER 句は、77 レベル記述項および PICTURE 句のない基本項目に指定する。
- 各整数は符号のない 1~3 けたの数字でなければならない。整数は画面の行番号を指定する。画面の最初の行番号は 1 であり、最後の行番号は Windows COBOL2002 では 120、UNIX COBOL2002 では 24 である。

書き方 3

- 整数 4 は絶対行番号を示す。次の規則を適用する。
 - 整数 4 の値は Windows COBOL2002 では 0 または 1~120、UNIX COBOL2002 では 0 または 1~24 であること。0 を指定した場合、そのフィールドは画面入出力命令の対象外となる。

(例) 0 を指定した場合

```
01 MAP0.
02 MAP1 LINE 10~....(1)
```

```

03 F1 COLUMN 10~.
03 F2 COLUMN 15~.
02 MAP2 LINE 10 COLUMN 40~....(5)
02 MAP3 LINE PLUS 5 COLUMN 50~....(6)

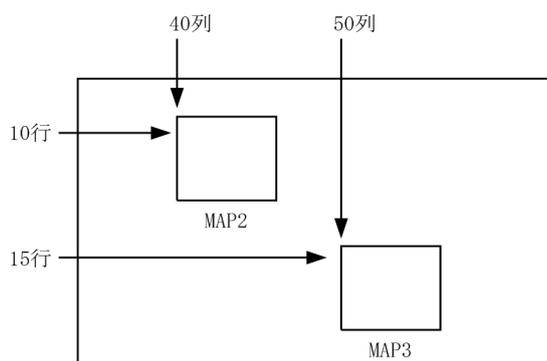
```

(1)を” 02 MAP1 LINE 0” にすると次のようになる。

```

PROCEDURE DIVISION.
  :
  MOVE SPACE TO MAP1. ... (2)
  :
  DISPLAY MAP0 UPON WINDOW. ... (3)
  :
  ACCEPT MAP1 FROM WINDOW. ... (4)
(2) MAP1にSPACEが設定される。
(3) MAP1は対象外のため表示されない。

```



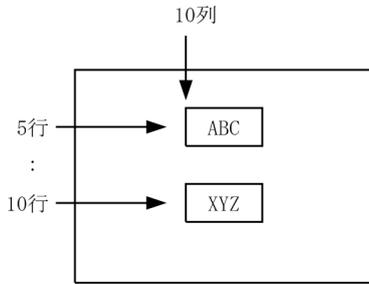
(4)は対象となるフィールドがないため何も実行しない。
ただし、次のようになる。
・ WINDOW-STATUS : 00
・ WINDOW-KEY : 00
・ フィールドステータス:OFF
(6)のようにPLUS指定がある場合、(5)を” 02 MAP2 LINE 0 COLUMN 40.”
とし、MAP2のフィールドを対象外とすると、(6)のMAP3のフィールドも
対象外となる。

(例) 書き方3のデータ名の指定例

```

WORKING-STORAGE SECTION.
77 I PIC 9(3).
  :
WINDOW SECTION.
01 GAMEN LINE I.
  02 A OUTPUT COLUMN 10 PIC X(3).
  :
  MOVE 5 TO I.
  MOVE 'ABC' TO A.
  DISPLAY GAMEN UPON WINDOW.
  MOVE 10 TO I.
  MOVE 'XYZ' TO A.
  DISPLAY GAMEN UPON WINDOW.

```



- 一つの 01 レベルの集団で、絶対行番号の値は任意の順序で書いてもよいし、同じ値のものを書いてもよい。ただし、上の行から下の行へ順に指定するのが一般的な書き方である。
- 一つの 01 レベルの集団で最初の LINE NUMBER 句には、絶対行番号を指定すること。

2. 整数 5 は相対行番号を示す。次の規則を適用する。

- 整数 5 の値は Windows COBOL2002 では 1～119、UNIX COBOL2002 では 1～23 でなければならない。
- 整数 5 を指定したときの行番号は、直前の行番号に整数 5 の値を加えた位置となる。
行番号は Windows COBOL2002 では 120、UNIX COBOL2002 では 24 を超えてはならない。

(例) 絶対行番号と相対行番号の指定

01 A.	(行番号)
02 B LINE 10～.	10行
02 C LINE PLUS 2～.	12行
02 D LINE PLUS 5 ～.	17行
02 E LINE 20～.	20行

行番号は、Windows COBOL2002 では 119 以下、UNIX COBOL 2002 では 23 以下の値で指定する。

3. データ名は行番号を可変に定義することを示す。

次の規則を適用する。

- データ名は修飾してもよいが、添字付けおよび部分参照をしてはならない。
- データ名は整数項目であること。
- データ名の値は Windows COBOL2002 では 1～120、UNIX COBOL2002 では 1～24 でなければならない。それ以外の値のときは、実行時に行カラム番号不正で実行時エラーとなる。
- データ名を指定した項目の上位に、REPEAT LINE 句および REPEAT COLUMN 句があってはならない。
- データ名を指定した項目の下位に、REPEAT COLUMN 句があってはならない。

4. 上位と下位のレベルに同時に LINE NUMBER 句または REPEAT LINE 句を書いてはならない。

5. すべての基本項目は LINE NUMBER 句を含むか、またはこの句（もしくは REPEAT LINE 句）を含む集団項目に従属していなければならない。

書き方 4

1. 書き方 4 の LINE NUMBER 句は、画面に引くけい線の行番号を指定する。この句と RULE LINE 句および COLUMN NUMBER 句を組み合わせると自由なけい線を引ける。さらに、特定のフィールドに関係しないフィールド属性を定義するときも、その行番号を指定する。

2. 各整数の値は Windows COBOL2002 では 1~120 でなければならない。ただし、整数 6 は 0 でもよい。UNIX COBOL2002 では 1~24 でなければならない。ただし、整数 6 は 0 でもよい。0 を指定した場合、そのフィールドは画面入出力文の対象外となる。また、次に示す大小関係を満足しなければならない。

整数 6 < 整数 7 < 整数 8 < 整数 9 . . .

3. けい線の引き方を次に示す。なお、整数 6、整数 7 に関する規則は、整数 8、整数 9 およびこれらの繰り返しにも適用する。

● 整数 6 と整数 7 を指定した場合

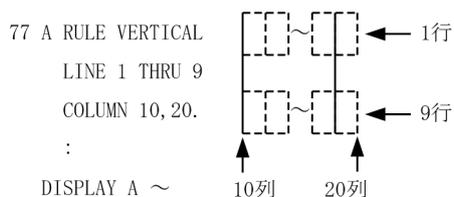
- 横けい線（上または下）は、整数 6 の行番号から整数 7 の行番号までのすべての行に引く。

(注意)

THROUGH 指定をして横けい線を引くのは一般的な使い方ではない。

- 縦けい線は、整数 6 の行番号から整数 7 の行番号まで引く。

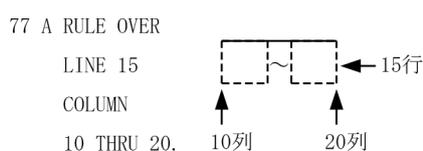
(例) 縦けい線の指定



● 整数 6 だけを指定した場合

- 横けい線（上または下）は、整数 6 の行番号に引く。横けい線のカラム番号は COLUMN NUMBER 句で指定する。

(例) 横けい線の指定例



- 縦けい線は整数 6 の行番号に引く。

(注意)

THROUGH 指定を省略して縦けい線を引くのは一般的な使い方ではない。

4. PICTURE 句のない基本項目は LINE NUMBER 句を含むか、またはこの句を含む集団項目に従属していないなければならない。

13.4.15 PICTURE 句 (WINDOW SECTION)

形式

書き方 1 (基本仕様)

$\left. \begin{array}{l} \text{PICTURE} \\ \text{PIC} \end{array} \right\} \text{IS 文字列}$

機能

PICTURE 句は、基本項目の性質と編集の形式を示す。

一般規則

1. PICTURE 句は基本項目にだけ指定できる。すべての基本項目には、PICTURE 句を指定しなければならない。
2. 数字および数字編集項目を定める PICTURE 文字列には文字「P」を書いてはならない。
3. 画面フィールドは PICTURE 句に従って次の 3 種類に分類する。

- 数字フィールド：数字 (外部 10 進)、数字編集項目
- 英数字フィールド：英字、英数字、英数字編集項目
- 日本語フィールド：日本語、日本語編集項目

これらのフィールドのサイズは、データ項目のサイズに等しい場合とそうでない場合がある。このサイズの決め方については、「[13.2.3 画面フィールドの種類とそのサイズ \(WINDOW SECTION\)](#)」を参照のこと。

4. 数字フィールドにデータを入力する場合と表示する場合の規則を次に示す。

- データを入力する場合

入力データは PICTURE 文字列に従って変換し、データ項目に格納する。

符号付きデータ項目に対し、符号なしでデータ入力をするとき + を仮定する。

データ項目に V 指定がある場合、小数点の位置を合わせて格納する。

入力のけた数がデータ項目のけた数より小さい場合は 0 を付けて格納する。

符号なしのデータ項目にはマイナスの値は入力できない。

(例) 数字フィールドに入力したデータの変換

PICTURE	入力データ	データ項目の値
9(5)	123	00123
9(5)	1234	01234
S9(5)	+1234	01234 ⁺
S9(5)	-12345	12345 ⁻
S999V99	-123.45	12345 ⁻
S999V99	+12.3	01230 ⁺

注
SIGN句があるときはその句に従って符号を設定する。

- データを表示する場合

データ項目の値をフィールドに右詰めで表示する。このとき、必要ならば左側に空白を補って表示する。符号付きの数字項目を表示する場合、先頭または末尾に符号 (+, -) を付けて表示する。仮想小数点 (V) の指定がある場合、その位置にピリオド (.) を表示する。

(例) 数字フィールドの表示

PICTURE	データ項目の値	表示データ
9(5)	12345	12345
S9(5)	01234 ⁻	-01234
S999V99	12345 ⁺	+123.45

5. 数字フィールドへは次のものが入力できる。

数字 (0~9), +, -, ピリオド, コマ (DECIMAL-POINT IS COMMA 句指定時だけ)

特殊名段落の DECIMAL-POINT IS COMMA 句を指定すると、コマと小数点の機能が入れ替わる。

+, -は内部的な符号に変換される。

- 符号ありの数字項目: (3) 16 または (7) 16
- 符号なしの数字項目: (3) 16

6. 英数字フィールドにデータを入力する場合と表示する場合の規則を次に示す。

- データを入力する場合

入力データは左詰めまたは右詰め (RESET 句があるとき) でデータ項目に格納する。ただし、編集文字の位置には入力できない。

(例) 英数字フィールドに入力したデータの格納

PICTURE	RESET	入力データ	データ項目の値
X(5)	なし	ABCDE	ABCDE
XXBXX	なし	ABCD	AB△CD
X(5)	なし	ABC	ABC△△
X(5)	あり	ABC	△△ABC

注
入力データがデータ項目長よりも短い場合には、空白を埋める。

- データを表示する場合
データ項目の値をそのままフィールドに表示する。

7. 日本語フィールドにデータを入力する場合と表示する場合の規則を次に示す。

- データを入力する場合
入力データは左詰めまたは右詰め（RESET 句、JUSTIFIED 句があるとき）でデータ項目に格納する。

(例) 日本語フィールドに入力したデータの格納

PIC	FREE	入力データ	データ項目の値
N (3)	なし	商品名	商品名
N (3)	あり	商品△△	商品
N (3)	あり	商品AB	商品AB
N (3)	あり	AB商品	AB商品

- データを表示する場合
データ項目の値をフィールドに表示する。

(例) 日本語フィールドの表示

PIC	FREE	データ項目の値	表示データ
N (3)	なし	商品名	商品名
N (3)	あり	商品AB	商品AB

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

13.4.16 REPEAT COLUMN 句 (WINDOW SECTION)

形式

```
REPEAT COLUMN IS  
  整数1 TIMES [WITH 整数2]
```

機能

REPEAT COLUMN (繰り返し列) 句は、同じデータ構造を横方向に繰り返す。

一般規則

1. REPEAT COLUMN 句は、01 レベルを除く集団項目に指定できる。
2. REPEAT COLUMN 句を指定した記述項に従属する項目には REPEAT COLUMN 句を指定してはならない。
3. Windows COBOL2002 では整数 1 および整数 2 は符号のない 1～3 けたの数字で、整数 1 は 1～160、整数 2 は 0～158 でなければならない。

UNIX COBOL2002 では整数 1 および整数 2 は符号のない 1~2 けたの数字で、整数 1 は 1~80、整数 2 は 0~78 でなければならない。

4. REPEAT COLUMN 句を書いた記述項のデータ名を参照するときには、添字を付ける。さらに、これに従属するデータ項目の名前を参照するときにも、添字を付ける。ただし、添字の付け方はマニュアル「COBOL2002 言語 標準仕様編」[4.3.1(2) 添字 (Subscript)]を参照のこと。
5. REPEAT COLUMN 句の上位または下位項目に REPEAT LINE 句が指定できる。このとき、表は 2 次元になる。
6. REPEAT COLUMN 句の下位項目には SOURCE 句を書いてはならない。
7. REPEAT COLUMN 句を指定したデータ項目には、LINE NUMBER 句を書いてはならない。
8. REPEAT COLUMN 句を指定したデータ項目の下位には、LINE NUMBER 句および COLUMN NUMBER 句を指定しなければならない。この COLUMN NUMBER 句で指定したカラム番号が、その項目の最初のカラム番号となり、整数 1 の回数だけ繰り返して配置される。
9. 整数 2 は REPEAT COLUMN 句で繰り返すデータ構造の間隔を指定する。整数 1 および整数 2 に従って配置した結果は、画面に収まる範囲内でなければならない。
10. WITH 指定を省略すると、整数 2 は 0 を仮定する。
11. 2 回目以降の繰り返しでのカラム番号は次の手順で決定する。

- REPEAT COLUMN 句の下位構造の中の最小カラム番号を求める。
- REPEAT COLUMN 句の下位構造の中の最大カラム番号を求める。

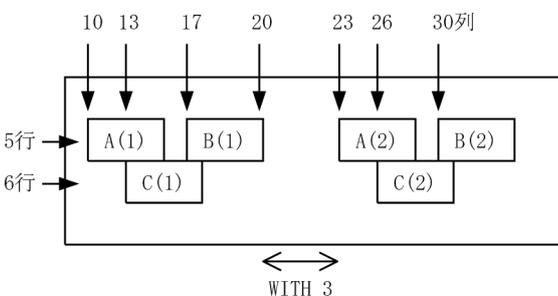
(各 COLUMN 句の値+フィールド長)

- 繰り返し相対値を求める。
(b) - (a) + 整数 2
- 各 COLUMN 句の値に前項を加えた値が、2 回目の繰り返しのカラム番号となる。

(例) フィールドの配列

```

02 REPEAT COLUMN 2 TIMES WITH 3.
03 LINE 5.
04 A PIC X(5) COLUMN 10 ~.
04 B PIC X(3) COLUMN 17 ~.
03 LINE 6.
04 C PIC X(5) COLUMN 13 ~.
    
```



- 構造内での最小のカラム番号:10

- 構造内での最大のカラム番号:20
- 繰り返し相対位置 ((b) - (a) +WITH 値) :13
- 2回目の繰り返しによるカラム番号:
 A (2) =10+13=23
 B (2) =17+13=30
 C (2) =13+13=26

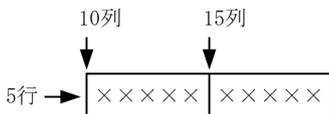
12.REPEAT COLUMN 句の下位項目のフィールド位置は、左から右へ、上から下になるように定義することが望ましい。

13.REPEAT COLUMN 句の上位項目には、RULE TABLE 句を指定してはならない。

14.REPEAT COLUMN 句の下位に指定された最初の LINE NUMBER 句および COLUMN NUMBER 句は絶対指定でなければならない。

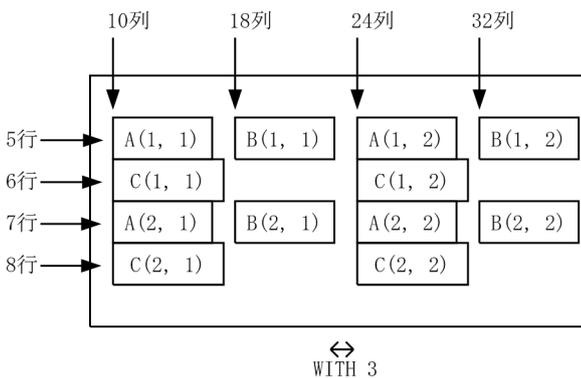
(例) フィールドの配置

```
02 REPEAT COLUMN 2 TIMES.
03 LINE 5.
04 A PIC X(5) COLUMN 10 ~.
```



(例) フィールドの配置 (REPEAT LINE との組み合わせ)

```
02 REPEAT LINE 2 TIMES.
03 REPEAT COLUMN 2 TIMES WITH 3.
04 LINE 5.
05 A PIC X(5) COLUMN 10 ~.
05 B PIC X(3) COLUMN 18 ~.
04 LINE 6.
05 C PIC X(7) COLUMN 10 ~.
```



- A (1, 1) B (1, 1) C (1, 1) の右側に 3 だけ空け、同じものを繰り返す。
- 上記右への繰り返しの後、行間を空けないで次の行に繰り返す。

13.4.17 REPEAT LINE 句 (WINDOW SECTION)

形式

書き方 1

```
REPEAT LINE IS  
  整数1 { THROUGH } 整数2  
        { THRU   }
```

書き方 2

```
REPEAT LINE IS  
  整数3 TIMES [WITH 整数4]
```

機能

REPEAT LINE (繰り返し行) 句は同じ構造を縦方向に繰り返す。

一般規則

1. REPEAT LINE 句は、01 レベルを除く集団項目に指定できる。
2. REPEAT LINE 句を指定した記述項に従属する項目には、REPEAT LINE 句を指定してはならない。
3. Windows COBOL2002 では整数 1、整数 2 および整数 3 は符号のない 1~3 けたの数字で、その値は 1~120 でなければならない。ただし、書き方 2 の整数 4 は、0~118 でなければならない。
UNIX COBOL2002 では整数 1、整数 2 および整数 3 は符号のない 1~2 けたの数字で、その値は 1~24 でなければならない。ただし、書き方 2 の整数 4 は、0~22 でなければならない。
4. REPEAT LINE 句を書いた記述項のデータ名を参照するときには、添字を付けなければならない。さらに、これに従属するデータ項目の名前を参照するときにも添字を付けなければならない。添字の付け方は表操作機能と同じである。

(例) 添字の付け方

```
02 A REPEAT LINE 5 THRU 7.  
03 B PIC ~  
03 C PIC ~  
03 D PIC ~
```

A(1)	B(1)	C(1)	D(1)	← 5行
A(2)	B(2)	C(2)	D(2)	← 6行
A(3)	B(3)	C(3)	D(3)	← 7行

データ名を参照するときには次のように添字を付ける。

```
DISPLAY A(1) UPON WINDOW.
```

```
MOVE NC '商品名' TO C(I).
```

5. REPEAT LINE 句を含む記述項に従属する記述項には、SOURCE 句を書いてはならない。

書き方 1

1. REPEAT LINE 句を書いたデータ項目およびそれに従属する項目には、LINE NUMBER 句を書いてはならない。また、上位レベルに LINE NUMBER 句を書いてはならない。

(例) REPEAT LINE 句の正しい書き方

```
01 A.  
02 B REPEAT LINE ~.    …正しい  
03 C PIC ~  
02 D REPEAT LINE ~.    …正しい  
03 E PIC ~
```

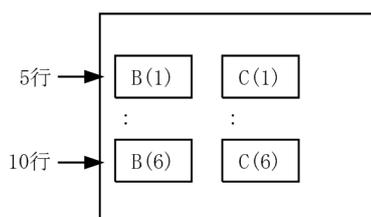
(例) REPEAT LINE 句の誤った書き方

```
01 A.  
02 B REPEAT LINE ~.  
03 C REPEAT LINE ~.    …誤り  
04 D LINE NUMBER ~     …誤り
```

2. 整数 1 および整数 2 は、画面の行番号を指定する。集団項目に REPEAT LINE 句を書くと、この集団に属するすべての基本項目を整数 1 の行番号から整数 2 の行番号まで繰り返して配置する。

(例) REPEAT LINE 句を用いたフィールドの配置

```
02 A REPEAT LINE 5 THRU 10.  
03 B PIC ~  
03 C PIC ~
```



B と C のフィールドを 5 行から 10 行までの各行に繰り返して配置する。

3. 整数 2 は整数 1 より大きい値を指定しなければならない。

書き方 2

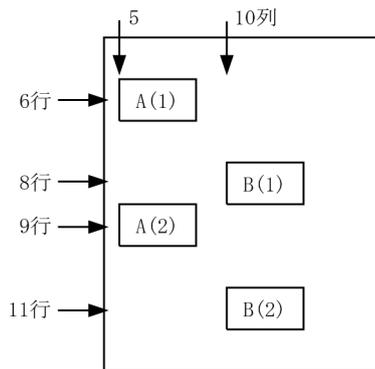
1. REPEAT LINE 句を指定したデータ項目の下位には、LINE NUMBER 句および COLUMN NUMBER 句を指定しなければならない。この LINE NUMBER 句で指定した行番号は、その項目が最初に現れる行番号となり、整数 3 の回数だけ繰り返して配置される。
2. 整数 4 は REPEAT LINE 句で繰り返す項目の間隔を指定する。整数 3 および整数 4 に従って配置した結果は、画面に収まる範囲内でなければならない。
3. WITH 指定を省略すると、整数 4 は 0 とみなす。
4. REPEAT LINE 句の下位に指定された最初の LINE NUMBER 句および COLUMN NUMBER 句は絶対指定でなければならない。

(例 1) フィールドの配置

```

02 REPEAT 2 TIMES.
03 A LINE 6 COLUMN 5 ~.
03 B LINE PLUS 2 COLUMN 10 ~.

```



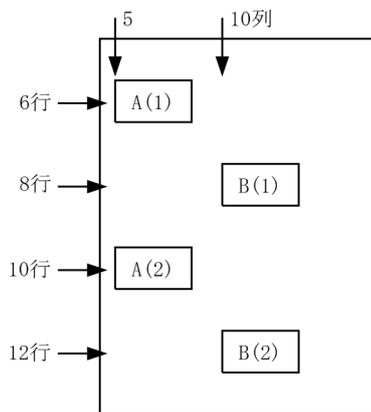
A と B のフィールドを 0 行間隔で 2 回繰り返す。

(例 2) フィールドの配置

```

02 REPEAT 2 TIMES WITH 1.
03 A LINE 6 COLUMN 5 ~.
03 B LINE 8 COLUMN 10 ~.

```



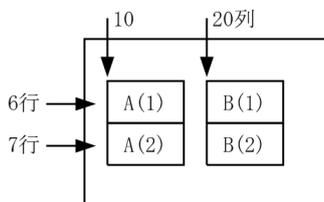
A と B のフィールドを 1 行間隔で 2 回繰り返す。

(例 3) フィールドの配置

```

02 REPEAT LINE 2 COLUMN TIMES.
03 A LINE 6 COLUMN 10 ~.
03 B LINE 6 COLUMN 20 ~.

```



A と B のフィールドを 0 行間隔で 2 回繰り返す。これは、次の REPEAT 句と同じ意味になる。

```

02 REPEAT LINE 6 THRU 7.
03 A COLUMN 10 ~.
03 B COLUMN 20 ~.

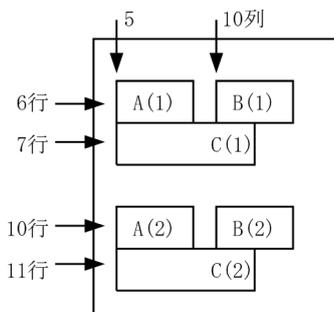
```

(例 4) フィールドの配置

```

02 REPEAT 2 TIMES WITH 2.
03 LINE 6.
04 A COLUMN 5 ~.
04 B COLUMN 10 ~.
03 C LINE 7 COLUMN 5 ~.

```



A と B のフィールドと C のフィールドを 2 行間隔で 2 回繰り返す。

13.4.18 RESET 句 (WINDOW SECTION)

形式

```

{ RESET
  RIGHT [WITH 定数1]
  RESETTED }

```

機能

RESET (右寄せ) 句は画面からの入力データを右に寄せて収めることを指定する。

一般規則

1. RESET と RESETTED は同義語である。
 2. RESET 句は入力または入出力フィールドに対する英字、英数字および日本語の基本項目に指定できる。出力フィールドには指定してはならない。
 3. 定数 1 には次の規則を適用する。
 - 英数字フィールドのとき、定数 1 は 1 文字の英数字定数、16 進英数字定数または表意定数でなければならない。表意定数は ZERO, SPACE, QUOTE, HIGH-VALUE, LOW-VALUE を指定できる。
- ただし、-UniObjGen オプションを指定した場合、定数 1 に UTF-8 で多バイトとなる文字が含まれてはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

- 日本語フィールドのとき、定数 1 は 1 文字（2 バイト）の日本語文字定数でなければならない。ただし、-UniObjGen オプションを指定した場合、定数 1 は指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

4. RESET 句は、ACCEPT 文または REPLY 文を実行して入力したデータを、フィールドとデータ項目の両方に右端をそろえて収める。

5. 定数 1 を書かないと左端には空白を埋める。

6. LETTER 句を指定した項目には、RESET 句は指定できない。

(例) RESET 句の実行例

PICTURE	定数	入力データ	表示データ	データ項目の値
X(4)	なし	AB△△	△△AB	△△AB
X(4)	'*	AB△△	**AB	**AB

(凡例)

△：空白

13.4.19 REVERSE 句 (WINDOW SECTION)

形式

REVERSE

機能

REVERSE (反転) 句はフィールドの背景の色と文字の色を反転して表示する。

一般規則

- REVERSE 句は、基本項目、77 レベル記述項および PICTURE 句のない基本項目に指定できる。
- REVERSE 句を書いた項目を表示したり入力したりすると、そのフィールドは背景の色と文字の色を反転して表示する。

13.4.20 RULE LINE 句 (WINDOW SECTION)

形式

書き方 1

RULE LINE IS SQUARE

書き方 2

`RULE LINE IS { OVER
UNDER
VERTICAL }`

書き方 3

`RULE LINE IS TABLE`

機能

RULE LINE (けい線) 句は、画面に引くけい線を指定する。

一般規則

1. 書き方 1 の RULE LINE 句は基本項目に指定できる。77 レベル記述項および PICTURE 句のない基本項目には指定できない。
2. 書き方 2 の RULE LINE 句は、基本項目、77 レベル記述項および PICTURE 句のない基本項目に指定できる。
3. 書き方 3 の RULE LINE 句は、01 レベルを除く集団項目に指定できる。

(注意)

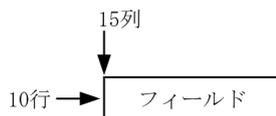
上けい線および下けい線が 2 行以上にわたる場合、最初の 1 行だけにけい線が引かれる。

書き方 1

1. SQUARE (枠けい線) 指定は、画面のフィールドをけい線で囲むことを指定する。

(例) SQUARE 句の指定

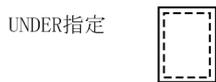
```
03 LINE 10  
COLUMN 15  
RULE SQUARE.
```



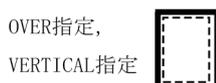
書き方 2

1. OVER (上けい線) 指定、UNDER (下けい線) 指定および VERTICAL (縦けい線) 指定は組み合わせて指定できる。ただし、同じ句を 2 回以上書いてはならない。
2. 各指定は画面に引くけい線の種類を指定する。画面に引くけい線の種類を次に示す。

図 13-2 画面に引くけい線の種類



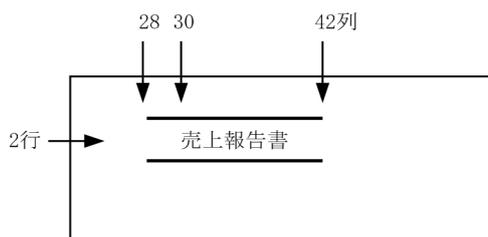
なお、次のように組み合わせてもよい。



- (凡例)  : 画面上の1文字分のフィールドのサイズ
 : 1文字 (1バイト) のサイズ
 : けい線

3. 画面に引くけい線の行番号とカラム番号は、LINE NUMBER 句と COLUMN NUMBER 句で指定する。

(例) 画面の見出しに、上けい線と下けい線を引く

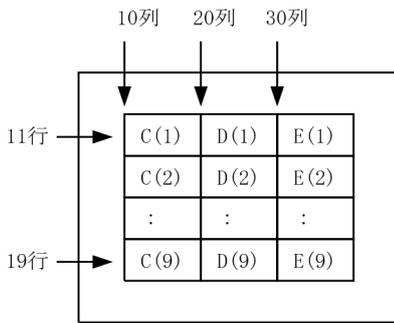


```
77 A LINE 2 COLUMN 28 THRU 42
      RULE LINE OVER, UNDER.
01 B LINE 2.
02 COLUMN 30 PIC N(5)
      VALUE NC'売上報告書'.
      :
      DISPLAY A B UPON WINDOW.
```

書き方 3

- 書き方 3 の RULE LINE 句を含む記述項には、書き方 1 の REPEAT LINE 句を含まなければならない。なお、この記述項に従属する記述項には、PICTURE 句のない基本項目および RULE LINE 句を書いてはならない。
- TABLE (表けい線) 指定は、REPEAT LINE 句で指定した行番号に、各基本項目を表形式のけい線で囲んで表示することを指定する。
- 表形式のけい線は次の規則を満足するときに表示する。
 - DISPLAY 文または REPLY 文の一意名が、この句を書いた項目を含む上位の集団項目の名前である。

(例) TABLE (表けい線) の指定



```
01 A.  
02 B REPEAT LINE 11 THRU 19 RULE LINE TABLE.  
03 C COLUMN 10 PIC ~  
03 D COLUMN 20 PIC ~  
03 E COLUMN 30 PIC ~  
:  
DISPLAY A UPON WINDOW.
```

(DISPLAY 文には TABLE 指定を書いた項目 (B) を含む, 上位の集団項目の名前 (A) を指定する)

13.4.21 SECRET 句 (WINDOW SECTION)

形式

SECRET

機能

SECRET (非表示) 句は画面のフィールドのデータを表示しないことを指定する。

一般規則

1. SECRET 句は, 基本項目, 77 レベル記述項および PICTURE 句のない基本項目に指定できる。
2. SECRET 句を書くと, そのフィールドの出力データまたは入力データは画面に表示されない。

13.4.22 SIGN 句 (WINDOW SECTION)

形式

[SIGN IS] { LEADING }
 { TRAILING }
 [SEPARATE CHARACTER]

機能

SIGN (符号) 句は演算記号の位置を指定する。

一般規則

1. SIGN 句は PICTURE 文字列に文字「S」を含む外部 10 進項目に指定できる。
2. SIGN 句は画面から入力したデータを外部 10 進項目に収めるときにも有効である。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.73 SIGN 句」

13.4.23 SOURCE 句 (WINDOW SECTION)

形式

SOURCE IS 一意名2

機能

SOURCE (源) 句はデータ項目に転記する値の送り出し側作用対象のデータ項目を指定する。

一般規則

1. SOURCE 句はすべての基本項目に指定できる。ただし、REPEAT LINE 句を含む記述項に従属する記述項には、SOURCE 句を書いてはならない。
2. 一意名 2 は報告書節を除くデータ部のどんな節で定義したものであってもよい。
3. 一意名 2 はデータ項目に転記する値の送り出し側作用対象データ項目を指定する。一意名 2 は MOVE 文の規則に従って定義しておかなければならない。
4. SOURCE 句を書くと ACCEPT 文, DISPLAY 文, REPLY 文, ERASE 文または SET 文を実行する直前に、一意名 2 の内容をデータ項目に転記する暗黙的な MOVE 文が実行される。

13.4.24 USAGE 句 (WINDOW SECTION)

形式

[USAGE IS] DISPLAY

機能

USAGE (用途) 句はデータの表現形式を指定する。

一般規則

1. USAGE 句はすべての基本項目に指定できる。
2. DISPLAY (表示用) は、項目の内部表現形式が標準データ形式であることを示す。
3. USAGE 句の指定がないと、その基本項目の USAGE は DISPLAY とみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

13.4.25 USING 句 (WINDOW SECTION)

形式

USING 編集文字列

機能

USING (指定) 句は数字項目の表現形式を指定する。

一般規則

1. USING 句は入出力フィールドまたは出力フィールドとして指定した数字基本項目にだけ指定できる。
2. 編集文字列は、数字編集項目に対する PICTURE 文字列と同じ規則で書かれていなければならない。
3. DISPLAY 文および REPLY 文を用いて画面にデータを出力する場合、編集文字列に従って編集した結果を表示する。編集は画面上でするだけで項目の内容は変化しない。したがって、この項目は演算できる項目 (数字項目) として扱える。
4. 画面上のフィールド長は編集文字列による編集項目のサイズとなる。編集文字列は PICTURE 句で指定した数字項目の長さと同じか、またはそれ以上指定しなければならない。
5. BLANK WHEN ZERO 句を同時に指定できない。

(例) USING フィールドの表示

PICTURE 文字列	USING 文字列	データ項目の サイズ (バイト)	フィールドの サイズ (バイト)	データ項目の値	表示データ
S9(5)	¥9(5)	5	6	12345 ⁻	¥12345
S9(5)	¥9(5)+	5	7	12345 ⁺	¥12345+
S9(4)V9	++,999.9	5	8	12345 ⁻	△△-123.4
S9(4)V9	¥Z,ZZZ.Z	5	8	12345 ⁻	¥△△△△△.1

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.87 USING 句」

13.4.26 VALUE 句 (WINDOW SECTION)

形式

書き方 1 (データ項目)

VALUE IS 定数1

機能

VALUE (値) 句は、データ項目の初期値を定める。

一般規則

1. VALUE 句はすべての基本項目に指定できる。
2. VALUE 句を書くと、プログラムの実行開始時にこの句で定めた値が項目の初期値となる。
3. -UniObjGen オプションを指定した場合、定数 1 に日本語文字定数および UTF-8 で多バイトとなる文字が含まれる英数字定数を指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

13.5 手続き部 (WINDOW SECTION)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

13.5.1 フィールド状態条件 (WINDOW SECTION)

形式

一意名 IS [NOT] STATUS-n

注

STATUS-nはSTATUS-0～STATUS-16

機能

フィールド状態条件は、画面入出力文を実行した後のフィールドの状態を調べる。

一般規則

- 一意名は画面節で指定した基本項目の名前でなければならない。集団項目や 77 レベル記述項および PICTURE 句のない基本項目の名前であってはならない。
- STATUS-n をフィールド状態表示子という。STATUS-n は、画面入出力文を実行し、個々のフィールドの状態をオンまたはオフで示す。
- 条件に NOT を書かないと、フィールド状態表示子がオンのとき検査の結果は真となる。条件に NOT を書くと、この表示子がオフのとき検査の結果は真となる。
- フィールド状態表示子は、該当する画面入出力文の実行によってオンまたはオフに設定され、その文を実行した後に参照できる。
- このシステムでは STATUS-0～STATUS-16 は常にオフ (0) である。

13.5.2 ACCEPT 文 (WINDOW SECTION)

形式

書き方 5 (画面節 (WINDOW SECTION))

ACCEPT {一意名7}...FROM {
WINDOW
呼び名2}

[USING ONLY DATA]

[AUTO-ADVANCE]

[WITH NO WAIT]

[BY END KEY]

[BUZZER]

[IN NUMERIC MODE]

[FIRST FIELD IS 一意名8]

[AUTO-SHIFT 整数3]

機能

ACCEPT (小入力) 文は画面からデータを入力してデータ項目へ格納する。

一般規則

1. 呼び名 2 は環境部の特殊名段落で WINDOW に関係づけた名前ではない。
2. 一意名 7 は、画面節で定義した 01~49 レベルのデータ項目の名前ではない。77 レベル記述項および PICTURE 句のない基本項目の名前ではない。また、PICTURE 句のある基本項目を一つも含まない集団項目ではない。
3. 一意名 7 が集団項目のときは、その集団に属するすべての入力フィールドと入出力フィールドに対してデータを入力する。ただし、その集団に入力フィールドまたは入出力フィールドが一つもない場合には、終了キーの打けんだけを待つ。SET 文で PROTECT (保護) としている場合は、出力フィールドとみなされる。

(例) 集団項目を指定した ACCEPT 文

```
02 A.  
03 B INPUT ~:入力フィールド  
03 C I-0 ~:入出力フィールド  
03 D OUTPUT ~:出力フィールド  
:  
ACCEPT A FROM WINDOW.
```

ACCEPT 文を実行すると、B と C のフィールドに対してデータを入力する。D のフィールドに対しては何もしない。

4. 一意名 7 が基本項目のときは、そのフィールドに対してだけデータを入力する。ただし、そのフィールドが出力フィールドの場合には、終了キーの打けんだけを待つ。
5. USING ONLY DATA 指定は、データを入力するときに、フィールド属性を新たに設定しないことを示す。このフィールドには、ACCEPT 文を実行する前に SET 文または DISPLAY 文を実行してフィールド属性を設定しておく。

(例) USING ONLY DATA 指定の使用

```
03 A INPUT ~  
:  
ACCEPT A FROM WINDOW.  
|  
入力データのエラーを検出  
|  
SET FIELD A TO REVERSE. ... (1)  
ACCEPT A FROM WINDOW USING ONLY DATA. ... (2)
```

(1) エラーを検出したのでフィールドを反転する。

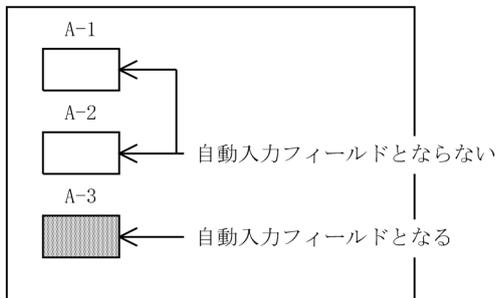
(2) フィールドを反転したままデータを再入力する。

6. AUTO-ADVANCE 指定は項目の長さ分のデータ入力があると、終了キーの打けんを待たないで入力を終了する。ただし、長さ分を入力する前に終了キーを打けんすると、そこで入力は終了する。特にこのフィールドを自動入力フィールドという。

- AUTO-ADVANCE 指定を書いた場合、項目の長さ分のデータを入力して操作を終了したときは、WINDOW-KEY（終了キー）の値として「00」が設定される。
- ACCEPT 文で複数のフィールドを指定した場合、自動入力フィールドとなるのは、画面上で最右下に位置するフィールドである。

(例) AUTO-ADVANCE 指定の画面表示フィールド

• ACCEPT A-1 A-2 A-3 FROM WINDOW AUTO-ADVANCE. は画面上では次に示す入力要求となる。



7. BUZZER 指定は、データ入力の応答待ちになるとき、短発音のブザーを鳴動する。

8. FIRST FIELD 指定は、データの入力待ちになるときに一意名 8 で指定したフィールドにカーソルを位置づける。この指定を書いたとき一意名 8 は一意名 7 に含まれる入力、または入出力フィールドとして定義された基本項目の名前でなければならない。

FIRST FIELD 指定がない場合、画面の左上にある入力フィールドにカーソルを位置づける。入力フィールドがない場合、カーソル位置は変更されないで終了キー入力待ちとなる。

9. BY END KEY 指定, IN NUMERIC MODE 指定, AUTO-SHIFT 指定, WITH NO WAIT 指定は覚え書きとみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

13.5.3 CANCEL 文 (WINDOW SECTION)

形式

書き方 2 (先読みデータのキャンセル)

```
CANCEL MESSAGE { [定数2] } [WITH 一意名3]
                [一意名2]
```

このシステムでは CANCEL 文を覚え書きとみなす。

機能

CANCEL (取り消し) 文は先行入力したデータ (先読みデータ) を取り消す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.5 CANCEL 文」

13.5.4 DISPLAY 文 (WINDOW SECTION)

形式

書き方 4 (画面節 (WINDOW SECTION))

DISPLAY {一意名6}…
UPON { WINDOW }
 { 呼び名2 }
[USING ONLY DATA]
[BUZZER]
[WITH TERMINAL IN ORDER]

書き方 5 (画面節 (WINDOW SECTION))

DISPLAY
{ データ名1 [{ THROUGH } データ名2] }…
 { THRU }
UPON { WINDOW }
 { 呼び名2 }
[USING ONLY DATA]
[BUZZER]
[WITH TERMINAL IN ORDER]

機能

DISPLAY (表示) 文はデータ項目の内容を画面に表示する。

一般規則

1. 呼び名 2 は、環境部の特殊名段落で WINDOW に関係づけた名前でないといけない。
2. BUZZER 指定は、データを表示するとき、短発音のブザーを鳴動する。
3. WITH TERMINAL IN ORDER 指定は覚え書きとみなす。

書き方 4

1. 一意名 6 は画面節で指定したデータ項目の名前でなければならない。このデータ項目は 01~49 または 77 のどんなレベルでもよい。
2. 一意名 6 が集団項目のときは、その集団に属しているすべての基本項目に対して次の規則を適用する。
 - 出力または入出力フィールドにはデータを表示する。
 - 入力フィールドに対しては何もしない。
 - PICTURE 句のない基本項目には、各句の指定に従ってけい線などを表示する。
3. 一意名 6 が基本項目のときは、そのフィールドに対してだけデータを表示する。
4. 一意名 6 が 77 レベル記述項または PICTURE 句のない基本項目のときは、各指定に従ってけい線などを表示する。

5. USING ONLY DATA 指定はデータを出力するときに、フィールド属性を新たに設定しないことを示す。このフィールドには、DISPLAY 文を実行する前に SET 文または DISPLAY 文を実行してフィールド属性を設定しておく。

書き方 5

1. THRU は THROUGH の同義語である。
2. データ名は画面節で指定した 01 レベルまたは 77 レベルのデータ項目の名前でなければならない。
3. THROUGH 指定を書いたとき、データ名 2 はデータ名 1 よりも後に定義されたものでなければならない。
4. 書き方 5 の DISPLAY 文は、範囲を指定してデータ項目の内容を表示する。THROUGH 指定を書くと、データ名 1 からデータ名 2 までのすべてのデータ項目が表示の対象となる。データ名 2 が 01 レベルの集団項目のとき、これに属する最後の基本項目までがその範囲となる。

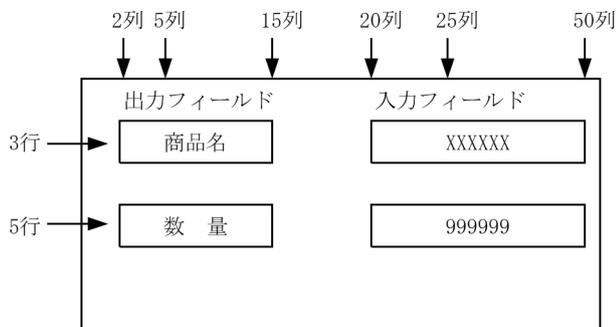
(例 1) 書き方 5 の DISPLAY 文

```
77 A ~
77 B ~
77 C ~
01 D.
  02 D1 ~
  02 D2 ~
```

上記のとき DISPLAY A THRU C ~ は A, B, C を表示する。

DISPLAY A THRU D ~ は A, B, C, D1, D2 を表示する。

(例 2) 実際のコーディング例



```
77 A LINE 3, 5
  COLUMN 2 THRU 15
  20 THRU 50
  RULE OVER, UNDER, VERTICAL.
01 B OUTPUT.
  02 LINE 3 COLUMN 5 PIC N(3) VALUE NC'商品名'.
  02 LINE 5 COLUMN 5 PIC N(3) VALUE NC'数量'.
01 C INPUT.
  02 LINE 3 COLUMN 25 PIC ~
  02 LINE 5 COLUMN 25 PIC ~
```

次の DISPLAY 文を実行すると上の画面が実行される。

DISPLAY A, B, C UPON WINDOW.

または

DISPLAY A THRU C UPON WINDOW.

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

13.5.5 ERASE 文 (WINDOW SECTION)

形式

$$\left. \begin{array}{l} \text{ERASE} \\ \left\{ \begin{array}{l} \text{ALL} \left\{ \begin{array}{l} \text{WINDOW} \\ \text{UNPROTECTED FIELD} \end{array} \right\} \\ \text{UNPROTECTED FIELD } \{\text{一意名1}\} \dots \\ \text{FIELD } \{\text{一意名1}\} \dots \\ \text{ATTRIBUTE } \{\text{一意名1}\} \dots \end{array} \right\} \end{array} \right\}$$

機能

ERASE (消去) 文は画面またはフィールドを消去する。

一般規則

- 一意名 1 は画面節で指定した 01~49 レベルのデータ項目の名前でなければならない。77 レベル記述項および PICTURE 句のない基本項目の名前であってはならない。ただし、ERASE ATTRIBUTE の場合は、77 レベルおよび PICTURE 句のない基本項目を指定してもよい。
- ALL WINDOW 指定は画面全体を消去する。
- ALL UNPROTECTED FIELD 指定は、画面すべての非保護フィールド（入力または入出力フィールド）を空白文字で消去する。
- UNPROTECTED FIELD 指定には次の規則を適用する。
 - 一意名 1 が集団項目のとき、その集団に属するすべての非保護フィールドを空白文字で消去する。
 - 一意名 1 が基本項目のとき、その非保護フィールドを空白文字で消去する。
- FIELD 指定には次の規則を適用する。
 - 一意名 1 が集団項目のとき、その集団に属するすべてのフィールドを空白文字で消去する。
 - 一意名 1 が基本項目のとき、そのフィールドを空白文字で消去する。
- ALL WINDOW 以外の指定は、フィールドのデータだけを消去する。フィールド属性は元の値のままである。
- ATTRIBUTE 指定には次の規則を適用する。
 - 一意名 1 が集団項目のとき、その集団に属するすべてのフィールドのデータとフィールド属性を消去する。
 - 一意名 1 が基本項目のとき、そのフィールドのデータとフィールド属性を消去する。

- 一意名 1 が 77 レベル記述項または PICTURE 句のない基本項目のとき、けい線、色、拡大のフィールド属性を消去する。

(例) ERASE 文の指定

```

02 A.
03 B INPUT ~
03 C OUTPUT ~
03 D I-0 ~
   :
   ERASE UNPROTECTED A. ... (1)
   ERASE FIELD A.      ... (2)

```

(1)は B と D を消去する。(2)は B, C および D を消去する。

13.5.6 EXIT USE 文 (WINDOW SECTION)

形式

書き方 7 (USE 出口)

EXIT USE

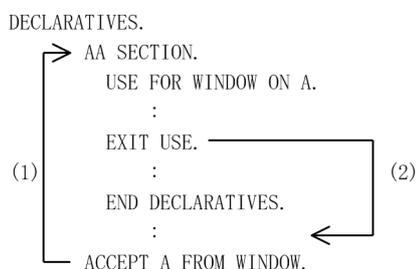
機能

EXIT USE (USE 出口) 文は、本文が書かれている USE 手続きに制御を移した文 (ACCEPT 文, REPLY 文) に制御を移す。

一般規則

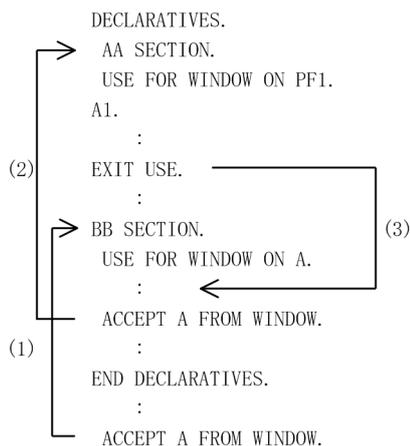
- EXIT USE 文は、USE (FOR WINDOW) 手続き内でだけ指定できる。
- EXIT USE 文に制御が移った場合、次のように実行される。
 - 手続き部分から ACCEPT 文または REPLY 文で制御が移った場合、その文に制御が移る。
 - USE 手続きから ACCEPT 文または REPLY 文で制御が移った場合、その文に制御が移る。
 - ACCEPT 文または REPLY 文以外で制御が移った場合、結果は保証しない。

(例 1) 手続き部分から制御が移った場合の実行順序



- ACCEPT 文によって AA SECTION. に制御が移る。
- EXIT USE 文によって ACCEPT 文を実行する。

(例 2) USE 手続きから制御が移った場合の実行順序



(1) ACCEPT文によってBB SECTIONに制御が移る。

(2) 終了キーとして[PF1]を押したため、このACCEPT文では、AA SECTIONに制御が移る。

(3) EXIT USE文によって制御を移したACCEPT文を実行する。

13.5.7 REPLY 文 (WINDOW SECTION)

形式

書き方 1

```
REPLY {一意名1}...UPON { WINDOW  
                        呼び名1 }  
[USING ONLY DATA]  
[AUTO-ADVANCE]  
[WITH NO WAIT]  
[BY END KEY]  
[BUZZER]  
[IN NUMERIC MODE]  
[FIRST FIELD IS 一意名2]  
[AUTO-SHIFT 整数]
```

書き方 2

```
REPLY データ名1 UPON WINDOW  
HELP LINE NUMBER IS  
{データ名2} [ { THROUGH } {データ名3 } ]  
{整数1} [ { THRU } {整数2} ] ]  
HELP COLUMN NUMBER IS  
{データ名4} { THROUGH } {データ名5 }  
{整数3} { THRU } {整数4} ]
```

機能

REPLY (応答) 文はデータ項目の内容を画面に表示した後、画面からデータを入力する。

一般規則

書き方 1

1. 呼び名 1 は環境部の特殊名段落で WINDOW に関係づけた名前でないといけない。
2. 一意名 1 は画面節で指定した 01~49 レベルのデータ項目の名前でなければならない。77 レベル記述項および PICTURE 句のない基本項目の名前であってはならない。また、PICTURE 句のある基本項目を一つも含まない集団項目であってはならない。
3. REPLY 文は同じ一意名に対して DISPLAY 文と ACCEPT 文をこの順序で書いたのと同じである。
(例 1) 「REPLY A UPON WINDOW.」は次の文を順に書いたのと同じである。

```
DISPLAY A UPON WINDOW.
ACCEPT A FROM WINDOW.
```

(例 2) 「REPLY A UPON WINDOW BUZZER.」は次の文を順に書いたのと同じである。

```
DISPLAY A UPON WINDOW BUZZER.
ACCEPT A FROM WINDOW BUZZER.
```

書き方 2

1. データ名 1 は画面節で定義された 01 レベルのデータ項目の名前でなければならない。この項目がポップアップ画面となる。
2. HELP LINE 指定および HELP COLUMN 指定は、ポップアップ画面の表示位置および表示範囲を指定する。各データ名と整数は、次の式を満足していなければならない。

$$1 \leq \text{データ名 2} < \text{データ名 3} \leq 24$$

$$1 \leq \text{データ名 4} < \text{データ名 5} \leq 80$$

$$1 \leq \text{整数 1} < \text{整数 2} \leq 24$$

$$1 \leq \text{整数 3} < \text{整数 4} \leq 80$$

(注意)

データ名 3 を省略した場合は、データ名 3=データ名 2 となり、整数 2 を省略した場合は整数 2=整数 1 となる。

13.5.8 SET 文 (WINDOW SECTION)

形式

書き方 13 (画面節 (WINDOW SECTION) の属性)

```
SET FIELD{一意名13}...
TO {
  (表示色
  HIGH-INTENSITY
  LOW-INTENSITY
  [NO]BLINK
  [NO]REVERSE
  PROTECT
  SECRET
  ATTRIBUTE
  ) ... }
```

書き方 14 (バックカーソルキー)

SET BACK CURSOR KEY TO 整数3

書き方 15 (ファンクションキー)

SET { 呼び名2 } ... TO { ON }
{ PF_n } { OFF }

注

PF_nはPF1～PF24

書き方 16 (スクロール)

SET FIXED-SCROLL-LINE

TO 整数4 整数5

機能

SET (設定) 文はフィールドの属性変更、バックカーソルキーの種類およびファンクションキー処理手続きの有効、無効を決める。

一般規則

書き方 13

- 一意名 13 は、画面節で指定した集団項目または基本項目の名前でなければならない。77 レベル記述項および PICTURE 句のない基本項目の名前であってはならない。
- 一意名 13 の記述項は、自分自身と上位および下位の記述項に、RULE LINE IS TABLE 句を含んでいてはならない。
- 表示色の予約語とその意味を次に示す。

GREEN：緑色

WHITE：白色

RED：赤色

BLUE：青色

MAGENTA：紫色

SKY-BLUE：空色

YELLOW：黄色

BLACK：黒色

また、その他の予約語の意味を次に示す。

HIGH-INTENSITY：高輝度 (このシステムでは赤色とする)

LOW-INTENSITY：低輝度 (このシステムでは緑色とする)

[NO]BLINK：点滅[点滅しない]

[NO]REVERSE：反転[反転しない]

PROTECT：保護

SECRET：非表示

ATTRIBUTE：一意名自身の属性

4. 表示色, HIGH-INTENSITY, LOW-INTENSITY, SECRET は, どれか一つだけを指定できる。これらのうちの一つと[NO]BLINK, [NO]REVERSE または PROTECT を組み合わせて指定してもよい。ただし, BLINK と SECRET を同時に指定しても, SECRET だけが有効となる。
5. 書き方 13 の SET 文は, 一意名に対するフィールドの属性を各句で指定した属性に変更する。このとき, 変更する属性は画面のフィールドの属性だけであり, 一意名自身の属性は変更しない。
6. このシステムでは BLINK 句を覚え書きとみなす。

(例 1) SET 文の使用

```

03 A INPUT ~
      :
ACCEPT A FROM WINDOW.
      |
      入力データのエラーを検出
      |
SET FIELD A TO RED REVERSE.          ... (1)
ACCEPT A FROM WINDOW USING ONLY DATA. ... (2)

```

- (1) フィールドのデータを赤色で反転表示するように指定する。ただし, A 自身の属性は元のままである。
- (2) フィールドを赤色で反転表示してデータを再入力する。USING ONLY DATA を書かないと A 自身の属性が設定されるため, 期待する結果が得られない。

(例 2) ATTRIBUTE 指定の使用

```

03 A INPUT ~
      :
SET FIELD A TO REVERSE.
      :
SET FIELD A TO ATTRIBUTE. ... (1)

```

- (1) フィールド属性を A 自身の属性に戻す。反転は通常表示に戻る。

書き方 14

1. 整数 3 は符号のない 1~2 けたの数字で, その値は 0~24 でなければならない。
2. 書き方 14 の SET 文はバックカーソルキーを設定または解除する。整数に 1~24 を指定すると, 次のファンクションキーをバックカーソルキーとして設定する。

整数 3 の値	対応するファンクションキー
1	[PF1]キー
2	[PF2]キー
:	:
24	[PF24]キー

整数 3 に 0 を指定すると, バックカーソルキーを解除する (バックカーソルキーとして設定したファンクションキーを通常の終了キーに戻す)。

3. オペレータがバックカーソルキーを打けんしたときの動作を次に示す。
 - カーソルが入力フィールドの先頭にある場合

直前（直前に入力フィールドがないときは最後）の入力フィールドの先頭にカーソルを位置づけて、データの入力を持つ。

- カーソルが入力フィールドの途中にある場合
そのフィールドの先頭にカーソルを位置づけて、データの入力を待つ。

4. バックカーソルキーとして設定したファンクションキーは、終了キーとして用いることはできない。

書き方 15

1. OFF を指定した場合、SET 文で指定したファンクションキーに対応するファンクションキー処理手続きは無効（制御はファンクションキー処理手続きには移らないで、フィールド処理手続きに移る）となる。
2. ON を指定した場合、SET 文で指定したファンクションキーに対応するファンクションキー処理手続きは有効（制御はファンクションキー処理手続きに移る）となる。

書き方 16

1. このシステムは書き方 16 を覚え書きとみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.43 SET 文」

13.5.9 USE 文 (WINDOW SECTION)

形式

書き方 5 (フィールド処理)

USE FOR WINDOW ON {データ名1}...

書き方 6 (ファンクションキー処理)

USE FOR WINDOW ON {呼び名1
PFn }

注

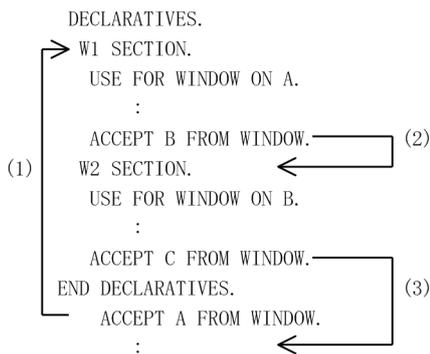
PFnはPF1～PF24

一般規則

1. ファンクションキー処理手続きには、書き方 6 の USE 文を用いる。フィールド処理手続きには、書き方 5 の USE 文を用いる。
2. USE 文は、宣言部分の節の見出しに続けて書き、終止符と空白で止める。この後に手続きを定義する段落を書く。
3. 宣言部分の節からほかの宣言部分および手続き部分の節や段落を参照してはならない。手続き部分からは宣言部分の節や段落を PERFORM 文でだけ参照できる。

4. 一つの基本項目に対する ACCEPT 文または REPLY 文によってデータを入力したとき、USE 手続きを実行する。作用対象が複数個指定されている場合や集団項目が指定されている場合は、USE 手続きは実行しない。
5. USE FOR WINDOW 節内に ACCEPT 文および REPLY 文がある場合、これらの文はその USE 手続きの出口とみなされる。これらの文は、USE FOR WINDOW 節外には記述できない。
6. 一つのプログラム、メソッドおよび利用者定義関数の中で USE FOR WINDOW 節は、そのプログラム、メソッドおよび利用者定義関数の中でだけ参照される。したがって、USE FOR WINDOW 節内の ACCEPT 文または REPLY 文がほかの USE FOR WINDOW 節を起動する場合、USE 手続きの実行が終了すると、最初に USE 手続きを呼び出した文の次の文へ制御が返る。

(例) USE 文の使用例 1



- (1) ACCEPTによって、W1 SECTION (Aに対するフィールド処理手続き) を起動する。
- (2) ACCEPTによって、W2 SECTION (Bに対するフィールド処理手続き) を起動する。
- (3) Cに対するフィールド処理手続きがないため、USE手続きを終了し、最初の ACCEPT文 ((1)のACCEPT文) の次の文を実行する。

7. ACCEPT 文または REPLY 文に対応するフィールド処理手続きが定義されていない場合は、ファンクションキーを押してもファンクションキー処理手続きには制御が移らない。
8. USE FOR WINDOW 節内に RESUME 文を指定してはならない。USE FOR WINDOW 内で RESUME 文を実行したときの動作は保証しない。

書き方 5

1. データ名 1 は、画面節で入力フィールドまたは入出力フィールドとして指定した基本項目の名前でなければならない。
データ名 1 で示すフィールドにデータを入力したとき、そのフィールドに対応するフィールド処理手続きに制御が渡る。
2. 同じデータ名を二つ以上の USE 文に書いてはならない。
3. データ名 1 の REPEAT LINE 句が記述されている項目に従属する項目の名前の場合、繰り返されるすべてのフィールドが USE 手続きへ制御を移す対象となる。このとき、データ名には添字を付けてはならない。

(例) USE 文の使用例 2

```

02 REPEAT 3 TIMES.
03 A I-0 ~.

```

```

02 B REPEAT 5 THRU 10 ~.
03 C INPUT ~.
  :
W1 SECTION.
  USE FOR WINDOW ON A.
  :
W2 SECTION.
  USE FOR WINDOW ON C.
  :
ACCEPT A(1) FROM WINDOW.  ... (1)
REPLY  A(3) UPON WINDOW.  ... (2)
ACCEPT C(5) FROM WINDOW.  ... (3)

```

(1)と(2)は W1 SECTION へ制御が移る。

(3)は W2 SECTION へ制御が移る。

書き方 6

1. 呼び名 1 は環境部の特殊名段落で PF1～PF24 に関係づけた名前でないといけない。
2. PF1～PF24 はそれぞれ対応するファンクションキーを意味する。ACCEPT 文または REPLY 文の終了キーとしてファンクションキー [PFn] を押したとき、キーに対応するファンクションキー処理手続きに制御を移す。ただし、SET BACK CURSOR KEY 文で指定したファンクションキーと同じキーを指定した場合は、終了キーとはならないでバックカーソルキーとして扱う。また、HELP KEY 句で指定したファンクションキーと同一キーを指定した場合は、終了キーとはならずヘルプキーとして扱う。

(例) USE 文の使用例 3

```

DECLARATIVES.
W1 SECTION.
  USE FOR WINDOW ON A.
  :
W2 SECTION.
  USE FOR WINDOW ON PF1.
  :
END DECLARATIVES.
ACCEPT A FROM WINDOW.  ... (1)
ACCEPT B FROM WINDOW.  ... (2)

```

(1)の ACCEPT 文では、終了キーが [PF1] であれば W2 SECTION を実行する。終了キーが [PF1] 以外であれば、W1 SECTION を実行する。

(2)の ACCEPT 文では、対応するフィールド処理手続きが定義されていないため、終了キーが [PF1] であっても W2 SECTION は実行しない。

3. 同じファンクションキーを二つ以上の USE 文に書いてはならない。
4. 一つのプログラム、メソッド、利用者定義関数内の USE 手続きは、そのプログラム、メソッド、利用者定義関数内だけで参照される。二つのプログラム間で USE 手続きの共用はできない。

(例) USE 文の使用例 4

```

IDENTIFICATION DIVISION.
PROGRAM-ID. OYA.
:
DECLARATIVES.
U1 SECTION.
  USE FOR WINDOW ON A, B. } .....(1)
  :
END DECLARATIVES.
:
ACCEPT A FROM WINDOW.      .....(2)
:
IDENTIFICATION DIVISION.
PROGRAM-ID. KO.
:
DECLARATIVES.
U2 SECTION.
  USE FOR WINDOW ON A. } .....(3)
  :
END DECLARATIVES.
:
ACCEPT A FROM WINDOW.      .....(4)
ACCEPT B FROM WINDOW.      .....(5)
:
END PROGRAM KO.
END PROGRAM OYA.

```

(2)のACCEPT文が実行された場合、(1)のU1 SECTIONが実行される。

(4)のACCEPT文が実行された場合、(3)のU2 SECTIONが実行される。

(5)のACCEPT文が実行されても、どのUSE手続きも実行されない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.53 USE 文」

13.5.10 WAIT 文 (WINDOW SECTION)

形式

WAIT

このシステムでは WAIT 文を覚え書きとみなす。

機能

WAIT (終了待ち) 文は、オペレータのデータ入力の終了を待つ。

14

書式印刷機能

この章では、入出力文で書式印刷機能を実行するための言語の形式や規則について説明する。

Linux(x86) COBOL2002 および 64bit 版 COBOL2002 では、XMAP3 を使用したプリンタ出力機能は使用できない。

14.1 環境部（書式印刷機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部（ENVIRONMENT DIVISION）」

14.1.1 入出力管理記述項（書式印刷機能）

(1) APPLY FORMS-OVERLAY 句

形式

```
APPLY FORMS-OVERLAY  
TO データ名 ON {ファイル名1}...
```

機能

この句は、XMAP3 使用時のプリンタに割り当てられたファイルに対して書式を用いた印刷をすることを指定する。詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の書式オーバーレイの説明を参照のこと。

構文規則

1. データ名 1 は長さが 8 バイトである英字項目、英数字項目、または字類が英数字の固定長集団項目でなければならない。
2. ファイル名 1 は、XMAP3 使用時のプリンタに割り当てる順ファイルでなければならない。
3. ファイル名 1 には EXTERNAL 句を指定したファイルは指定できない。
4. ファイル名 1 の ASSIGN 句には、定数またはデータ名は指定できない。

一般規則

1. データ名の領域に、印刷する書式を表す名称を 8 文字以下で、左詰めにして設定する。
2. 印刷する書式を変更する場合は、データ名が表す領域に変更する書式名称を設定して、その内容を書き換えなければならない。
3. 書式印刷を中止する場合は、データ名の領域に空白文字を設定する。空白文字を設定した直後に実行する WRITE 文から書式を印刷しないでレコード内容だけを印刷する。
4. 書式印刷を開始したとき、書式を変更および印刷中止すると、XMAP3 使用時のプリンタは改ページする。
5. この書式制御は、ADVANCING 指定を書いた WRITE 文を使用した場合に利用できる。
6. この句は、LINAGE 句を使用して論理ページを定義したファイルに指定できない。

14.2 データ部（書式印刷機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

14.2.1 データ記述項（書式印刷機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.13 データ記述項 (data description entry)」

(1) CHARACTER TYPE 句

形式

```
[CHARACTER TYPE IS]
{
  KEIS※1
  POINT-l※2
  GAP-m※1※2
  FORMAT-n※2
  INTERVAL-i※2
  WIDE
  FREE※1※3
  SHORT※1※3
  SHORT-1※1※3
}
```

注※1

覚え書き

注※2

l: 0～127

m: 0～511

n: 0～127

i: 0～127

注※3

MIOS7 COBOL85との互換のための日立拡張仕様である

UNIX COBOL2002では、使用できない

機能

レコードを出力する WRITE 文の実行によって、項目の文字サイズ、文字間隔、書体、横倍角開始を表す機能を指定する。この機能は、XMAP3を使用したプリンタ出力機能およびGDIモード印刷機能でも使用できる。CHARACTER TYPE 句はESC/Pモード印刷には有効とならない。このため、レコード中に機能コードを設定して印刷制御をする必要がある。

UNIX COBOL2002では、-XMAP,LinePrint オプション未指定時は、CHARACTER TYPE 句は覚え書きとする。

構文規則

1. XMAP3を使用したプリンタ出力機能の場合、POINT-*l*の*l*の値、FORMAT-*n*の*n*の値およびINTERVAL-*i*の*i*の値については、マニュアル「画面・帳票サポートシステム XMAP3 プログラミン

「グガイド 帳票編」または「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」を参照のこと。

なお、INTERVAL 指定をした場合、GAP 指定はできない。

2. この句を書けるのは、ファイル節、作業場所節、局所場所節または連絡節で定義されるデータ記述項だけである。
3. この句は、RENAMES 句とともに書けない。
4. DEPENDING ON 指定の OCCURS 句が含まれる集団項目およびその下位項目には、CHARACTER TYPE 句を指定できない。
5. WRITE 文で指定したレコード名およびその下位項目、または WRITE 文の FROM 指定の一意名およびその下位項目も、REDEFINES 句と CHARACTER TYPE 句を同時に指定してはならない。

(例)

```
01 OUTREC.  
02 A    PIC X(2).  
02 B.  
03 C    PIC X(4).  
03 D    PIC N(2) CHARACTER TYPE IS POINT-7.  
02 E REDEFINES B.  
03 F    PIC N(2).  
03 G    PIC X(4).  
02 H    PIC S9(5).  
:  
WRITE A-REC FROM OUTREC. (誤) ... (1)  
WRITE A-REC FROM B.      (正) ... (2)
```

(1)の WRITE 文は指定した一意名の下位項目の中に REDEFINES 句と CHARACTER TYPE 句を同時に指定しているので誤りである。

(2)の WRITE 文は正しい書き方である。

6. WRITE 文の FROM 指定の一意名またはその下位項目に CHARACTER TYPE 句が指定されているとき、その一意名は、部分参照されてはならない。
7. WRITE 文の FROM 指定の一意名またはその下位項目に CHARACTER TYPE 句が指定されているとき、その項目を含むレコード中に DISPLAY 以外の USAGE 句を指定してはならない。
8. UNIX COBOL2002 では、WRITE するレコード中に CHARACTER TYPE 句が含まれるファイルの ASSIGN 句には、定数指定またはデータ名指定はできない。
9. WRITE するレコード中に CHARACTER TYPE 句が含まれるファイルには、順ファイル以外は指定できない。
10. CHARACTER TYPE 句を処理するファイルのレコード長は、384 バイト以下でなければならない。
11. CHARACTER TYPE 句を処理するファイルに、FROM 指定なしの WRITE 文と FROM 指定ありの WRITE 文は同時に使用できない。
12. CHARACTER TYPE 句の指定がある項目を含むレコード中に、外部属性を持つ指標名は指定できない。
13. KEIS, POINT-*l*, GAP-*m*, FORMAT-*n*, INTERVAL-*i*, SHORT, SHORT-1 の語は予約語ではない。

これらの語が CHARACTER TYPE 句の一部であるかどうかは文脈で判断される。

これらの語をレベル番号の直後に書いた場合、これらの語がデータ名として扱われる。

(例)

```
03 POINT-7 PIC N(2).
```

と書くと、「FILLER」を書くのを省略した無名項目ではなく、データ名「POINT-7」が指定されたものとみなされる。

一般規則

1. POINT-*l* を書くことによって、文字サイズ（ポイント）を指定できる。*l* の値はポイント数を示す。

上位項目に異なる値の POINT-*l* が指定されていると、そのポイントに戻すための処理が項目の後に実行される。上位項目に POINT-*l* の指定がない場合は、省略値のポイントに戻すための処理が項目の後に実行される。ただし、後に続く項目がなければ、元に戻すための処理はしない。

l の値は、XMAP3 で使用できるポイント数でなければならない（XMAP3 使用時だけ）。

POINT-*l* を書かない場合の省略値のポイントは、GDI モード印刷機能のときは、実行支援の [字サイズ] で指定したポイントとなる。また、XMAP3 を使用するときには、XMAP3 で定義したポイントとなる。

(例)

```
01 A.  
02 A-F PIC X.  
02 A0 CHARACTER TYPE IS POINT-14.  
03 A1 PIC NN VALUE N'下期'.  
03 A2 PIC N(5) POINT-19 VALUE N'实用新案権'.  
03 A3 PIC NN VALUE N'一覧'.
```

とすると、プリンタへの出力は、次のようになる。

下期 实用新案権 一覧

14ポイント開始 19ポイント開始 14ポイント開始

2. FORMAT-*n* を書くことによって、書体を指定できる。

上位項目に異なる値の FORMAT-*n* が指定されていると、その書体に戻すための処理が項目の後に実行される。上位項目に FORMAT-*n* の指定がない場合は、省略値の書体に戻すための処理が項目の後に実行される。ただし、後に続く項目がなければ、元に戻すための処理はしない。

FORMAT-*n* を書かないときの省略値の書体は、実行支援の [書体] の初期状態で指定したフォントとなる。また、XMAP3 を使用するときには、XMAP3 で定義したフォントとなる。

(例)

```
01 A.  
02 A-F PIC X.  
02 A0 CHARACTER TYPE IS FORMAT-2.
```

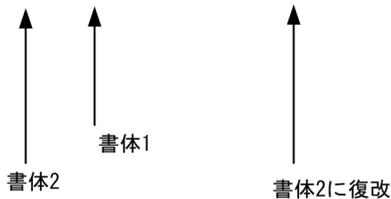
```

03 A1 PIC NN VALUE N'下期'.
03 A2 PIC N(5) FORMAT-1 VALUE N'实用新案権'.
03 A3 PIC NN VALUE N'一覧'.

```

とすると、プリンタへの出力は、次のようになる。

下期**实用新案権**一覧



3. INTERVAL-*i* を書くことによって、各文字間隔を指定できる。

上位項目に異なる値の INTERVAL-*i* が指定されていると、その文字間隔に戻すための処理が項目の後に実行される。上位項目に INTERVAL-*i* の指定がない場合は、省略値の文字間隔に戻すための処理が項目の後に実行される。ただし、後に続く項目がなければ、元に戻すための処理はしない。

INTERVAL-*i* を書かないときの省略値の文字間隔は、0 ポイントである。

実行支援の[全角文字の間隔を半角文字の間隔の2倍で印刷]チェックボックスをチェックしない場合は、全角文字と半角文字の字間隔は $i/2$ ポイントとなる。このチェックボックスをチェックした場合、全角文字の字間隔は i ポイント、半角文字の字間隔は $i/2$ ポイントとなる。

(例)

```

01 A.
02 A-F PIC X.
02 A0 CHARACTER TYPE IS INTERVAL-0.
03 A1 PIC NN VALUE N'下期'.
03 A2 PIC N(5) INTERVAL-8 VALUE N'实用新案権'.
03 A3 PIC NN VALUE N'一覧'.

```

とすると、プリンタへの出力は、次のようになる。

下期実 用 新 案 権 一 覧



4. WIDE を書くことによって、横倍角を指定できる。

上位項目に WIDE の指定がない場合は、省略値の字体に戻すための処理が項目の後に実行される。ただし、後に続く項目がなければ、元に戻すための処理はしない。WIDE を書くことによって、字間隔も2倍になる。

(例)

```

01 A.
02 A-F PIC X.
02 A0.
03 A1 PIC NN VALUE N'下期'.

```


15

アドレス操作機能

この章では、参照したいデータを記憶領域中の番地でポイントするアドレス操作機能について説明する。

15.1 アドレス操作機能とは

アドレス操作機能とは、参照したいデータを記憶領域中の番地でポイントする方法である。ただし、誤った操作をすると、目的以外の領域を破壊してしまうことがある。

アドレス操作機能を用いる場合、アドレス操作を熟知している必要がある。

以降では、COBOL でアドレスの操作をするための言語仕様について記述する。

15.2 データの表現形式 (アドレス操作機能)

15.2.1 アドレス名

アドレス名は、利用者定義語である。利用者定義語の構成規則に従う。

アドレス名は、データ記述項の作業場所節または局所場所節で、ADDRESSED 句を指定することによって定義される。アドレス名には、アドレスが入るための 4 バイトの記憶領域がとられ、用途が BINARY で符号なし 4 バイトの項目と同じに扱う (64bit 版 COBOL2002 では、アドレス名はアドレスが入るための 8 バイトの記憶領域がとられ、用途が BINARY で符号なし 8 バイトの項目と同じに扱う)。

アドレス名には、COBOL が取り扱うデータ項目の項類の概念が適用されない。

15.2.2 アドレスデータ項目

アドレスデータ項目は、USAGE 句で用途はアドレス (address) と指定された項目である。アドレス名の内容を一時的に退避しておき、それを再びアドレス名に戻すためのものである。

アドレスデータ項目には、COBOL が取り扱うデータ項目の項類の概念が適用されない。

15.3 データ部 (アドレス操作機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

15.3.1 アドレス名によって参照されるデータ項目の定義

アドレス名によって参照されるデータ項目は ADDRESSED 句を必須とする。

(1) ADDRESSED 句 (アドレス操作機能)

形式

$$\left. \begin{array}{l} 01 \\ 77 \end{array} \right\} \left[\begin{array}{l} \text{データ名1} \\ \text{FILLER} \end{array} \right] \text{ADDRESSED BY アドレス名1}$$

(注意)

形式中のレベル番号 01 または 77 とデータ名 1 は ADDRESSED 句の一部ではないが、説明の便宜上示してある。

機能

データ項目がアドレスにより参照されることを指定する。

構文規則

1. ADDRESSED 句は作業場所節または局所場所節の 01 または 77 レベルのデータ記述項にだけ指定できる。
2. 01 レベルに GLOBAL 句を指定すると、アドレス名も大域名となる。EXTERNAL 句は、ADDRESSED 句と同時に指定できない。
3. アドレス名 1 は修飾できない。したがって、アドレス名 1 と同じ名前のデータ名があってはならない。
4. ADDRESSED 句と REDEFINES 句を同じデータ記述項に指定してはならない。ただし、ADDRESSED 句を指定したデータ項目を再定義する REDEFINES 句は書いてもよい。
5. ADDRESSED 句を指定したデータレコード記述を再命名する RENAMES 句を書いてよい。ただし、通常の場合と同じく、データレコード記述の最後にまとめて記述しなければならない。
6. ADDRESSED 句を指定したデータレコード記述中のすべてのデータ記述項には、初期値設定用の VALUE 句を書いてはならない。
7. ADDRESSED 句を指定したデータレコード記述中のデータ記述項には、関連する条件名を書いてよい。
8. ADDRESSED 句を指定したデータ記述項に同時に指定できる句は次のとおりである。
 - PICTURE 句

- BLANK WHEN ZERO 句
- GLOBAL 句
- JUSTIFIED 句
- SIGN 句
- SYNCHRONIZED 句
- TYPE 句
- USAGE 句（ただし、オブジェクト参照データ項目には指定できない）

9. データ名 1 は SAME AS 句に指定できない。

10. ADDRESSED 句をファクトリ定義、インスタンス定義中で指定してはならない。

一般規則

1. データ名 1 のデータ項目は、固有の記憶領域を持たない。実行時にアドレス名 1 へアドレスを設定するとそのアドレス名 1 が指すデータ項目が、データ名 1 のデータ項目で再定義されたかのように扱われる。
2. データ名 1 のデータ記述項で定義されるデータ項目の属性と、アドレス名 1 が指すデータ項目の属性との間には、どんな意味でも適用される制限はない。ただし、アドレス名 1 が指すデータ項目の内容が、データ名 1 のデータ項目の定義と矛盾する場合、データ名 1 を参照した結果は保証しない。
3. データ名 1、そのデータレコード記述に従属するデータ記述項のデータ名または条件名などを手続き部で参照する場合は、それ以前にアドレス名 1 に適切なアドレスが設定されていなければならない。そうでない場合、実行結果は保証しない。
4. OCCURS 句の DEPENDING 指定に指定するデータ名は、ADDRESSED 句を指定したデータ名またはそのレコード記述に従属するデータ名を指定してはならない。

15.3.2 アドレスデータ項目の定義

アドレスデータ項目は、アドレス名の内容を一時的に退避したり、プログラム間でアドレス値を受け渡したりする場合に用いる。

(1) PICTURE 句（アドレス操作機能）

アドレスデータ項目のデータ記述項に、PICTURE 句を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

(2) SYNCHRONIZED 句 (アドレス操作機能)

1. アドレスデータ項目のデータ記述項に SYNCHRONIZED 句を指定すると、4 バイトの領域が 4 バイト境界に調整されて割り付けられる (64bit 版 COBOL2002 では、8 バイトの領域が 8 バイト境界に調整されて割り付けられる)。

SYNCHRONIZED 句を指定しないと、バイト境界に割り付けられる。

2. SYNCHRONIZED 句を集団項目に指定したとき、その集団項目に従属するすべてのアドレスデータ項目にこの指定が適用される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.80 SYNCHRONIZED 句」

(3) USAGE 句の ADDRESS 指定

形式

[USAGE IS] ADDRESS

機能

アドレスデータ項目であることを指定する。

構文規則

1. USAGE 句の ADDRESS 指定と同時に PICTURE 句、SAME AS 句および TYPE 句を指定してはならない。
2. アドレスデータ項目は次の場所にかける。
 - 作業場所節
 - 連絡節
 - ファイル節
 - 局所場所節
3. アドレスデータ項目に関連する条件名が記述できる。ただし、条件名の VALUE 句には、表意定数 ZERO と NULL だけが書ける。
4. アドレスデータ項目は、EXTERNAL 句や GLOBAL 句を指定したレコード記述中に書いてもよい。
5. アドレスデータ項目の VALUE 句には、表意定数 NULL だけが書ける。

一般規則

1. USAGE IS ADDRESS 句は、そのデータ項目がアドレスデータ項目であることを指定する。
2. アドレスデータ項目には、4 バイトの記憶領域がとられ、アドレス値 (記憶領域の番地) が 2 進形式で格納される (64bit 版 COBOL2002 では、アドレスデータ項目には 8 バイトの記憶領域がとられ、アドレス値が 2 進形式で格納される)。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

15.4 手続き部（アドレス操作機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

15.4.1 アドレス名，アドレスデータ項目および ADDR 関数の参照（アドレス操作機能）

アドレス名，アドレスデータ項目および ADDR 関数の参照は，次の場合に許される。

(1) 比較条件

アドレス名，アドレスデータ項目または ADDR 関数と比較できるものを次に示す。

- アドレス名
- アドレスデータ項目
- 表意定数 ZERO
- ADDR 関数

(2) プログラム間連絡機能および利用者定義関数の引数と返却項目

次に示す USING の作用対象および利用者定義関数の引数と返却項目にアドレスデータ項目が指定できる。

- 呼び出し先プログラムの手続き部見出しにある USING 指定
- CALL 文の USING 指定（ADDR 関数は BY VALUE 指定の引数にだけ指定できる）
- INVOKE 文（オブジェクト指向機能）の USING 指定（ADDR 関数は BY VALUE 指定の引数にだけ指定できる）
- 利用者定義関数の引数指定と返却項目

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.1 手続き部の構成」

(3) DISPLAY 文

DISPLAY 文でアドレス名，アドレスデータ項目または ADDR 関数の内容を SIGN LEADING SEPARATE 指定の外部 10 進項目に変換後，10 けたの外部 10 進形式（64bit 版 COBOL2002 では 20 けたの外部 10 進形式）で表示する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

(4) COMPUTE 文の作用対象

COMPUTE 文でアドレス名またはアドレスデータ項目にアドレスを設定または更新する。

15.4.2 COMPUTE 文 (アドレス操作機能)

COMPUTE 文でアドレス名またはアドレスデータ項目にアドレスを設定または更新する。

形式

書き方 1

$$\begin{array}{l} \text{COMPUTE } \left\{ \begin{array}{l} \text{アドレス名1} \\ \text{一意名1} \end{array} \right\} = \\ \text{FUNCTION ADDR } \left(\begin{array}{l} \text{一意名2} \\ \text{定数1} \end{array} \right) \\ \text{[END-COMPUTE]} \end{array}$$

書き方 2

$$\begin{array}{l} \text{COMPUTE } \left\{ \begin{array}{l} \text{アドレス名1} \\ \text{一意名1} \end{array} \right\} = \\ \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{アドレス名2} \\ \text{一意名3} \\ \text{ADDR関数} \end{array} \right\} \left[\begin{array}{l} + \\ - \end{array} \right] \left\{ \begin{array}{l} \text{一意名4} \\ \text{定数2} \\ \text{整数型関数} \end{array} \right\} \\ \text{ZERO} \\ \text{ZEROS} \\ \text{ZEROES} \end{array} \right\} \\ \text{[END-COMPUTE]} \end{array}$$

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「11.5 関数の要約」

機能

1. アドレス名またはアドレスデータ項目にアドレスを設定する (書き方 1)。
2. アドレス名またはアドレスデータ項目に外部プログラムのアドレスを設定する (書き方 1)。
3. アドレス名またはアドレスデータ項目の更新をする (書き方 2)。

構文規則

1. 一意名 1 および一意名 3 のデータ項目はアドレスデータ項目でなければならない。
2. 一意名 2 には、アドレスで参照したいデータ項目の名前を指定する。部分参照を指定してはならない。
3. 定数 1 は英数字定数でなければならない。
4. 定数 1 にはプログラム名の規則を適用する。

5. Windows(x86) COBOL2002 の場合、定数 1 に、呼び出し規約が STDCALL の外部プログラム名を指定してはならない。

コンパイラオプションの指定	呼び出し規約	コンパイルの結果
-DynamicLink,Call オプションなし	CDECL	正常終了
	STDCALL	リンクエラー
-DynamicLink,Call オプションあり	CDECL	正常終了
	STDCALL	コンパイルエラー

備考

コンパイルが正常終了する場合は、実行時にプログラムのアドレスが取得できる。

6. 一意名 4 は、整数項目でなければならない。

7. 定数 2 は、正の符号なし整数でなければならない。

8. 一意名 2 には、ファクトリ定義およびインスタンス定義で定義したデータ項目の名前を指定してはならない。

一般規則

- 書き方 1 の一意名 2 を指定した COMPUTE 文を実行すると、一意名 2 のデータ項目の占める記憶領域の先頭番地（アドレス）が、アドレス名 1 または一意名 1 に設定される。
- 書き方 1 のアドレス名 1、一意名 1 に ROUNDED 指定をしてはならない。
- 書き方 1 で定数 1 を指定する場合、アドレス参照したいプログラムの外部プログラム名を指定する。
- 書き方 1 の定数 1 を指定した COMPUTE 文を実行すると、定数 1 で示される外部プログラムの先頭番地がアドレス名 1 または一意名 1 に設定される。

(例)

```
DATA DIVISION.
WORKING-STORAGE SECTION.
    77 KAKUNO    ADDRESS.
PROCEDURE DIVISION.
    COMPUTE KAKUNO = FUNCTION ADDR('SUB_PROGRAM').
    IF KAKUNO = NULL
        THEN DISPLAY 'SUB_PROGRAM NO LOAD'
    END-IF.
```

5. Windows(x86) COBOL2002 では、書き方 1 の定数 1 で示される外部プログラムは、CDECL 属性だけが対象となる。

6. -DynamicLink,Call オプションが指定されている場合、定数 1 で示される外部プログラムが、ロードされた実行可能ファイル、DLL (Windows COBOL2002)、共用ライブラリ (UNIX COBOL2002) のどちらからも見つからなかった場合、アドレスは NULL を返す。

備考

外部プログラムが存在するかどうかは、リンカまたはローダによってチェックされる。外部プログラムが存在しない場合は、外部シンボル未解決エラーとなる。

なお、-DynamicLink,Call オプションの指定がある場合は、リンカやローダでチェックは行われず、実行時にプログラムのアドレスまたは NULL を返す。

定数 1 の指定	プログラムのローディング状態	ADDR 関数で返される値
存在する外部プログラム名	ロードされている	プログラムのアドレス
	ロードされていない	NULL
存在しない外部プログラム名	ロードされていない	NULL

7. 書き方 1 に ON SIZE ERROR, NOT ON SIZE ERROR を指定してはならない。
8. 書き方 2 の COMPUTE 文を実行すると、アドレス名 2 または一意名 3 の内容に、一意名 4 または定数 2 の値を加算または減算し、その結果がアドレス名 1 または一意名 1 に設定される。
9. 書き方 2 のアドレス名 1, 一意名 1 に ROUNDED 指定をしてはならない。
10. 書き方 2 の COMPUTE 文で算術演算式中に乗除算やべき乗を指定してはならない。
11. 書き方 2 に ON SIZE ERROR, NOT ON SIZE ERROR を指定してはならない。
12. 書き方 2 で等号の右辺に括弧を指定してはならない。
13. 書き方 1 で定数 1 を指定する場合、利用者定義関数名、メソッド名およびオブジェクトプロパティ名は指定できない。

(例)

```

DATA          DIVISION.
FILE          SECTION.
FD INFIL     BLOCK  CONTAINS 5 RECORDS.
01 INREC.
  02 HONTAI  PIC X(10).
  02 HUKA    PIC X(20).
FD OUTFIL.
01 OUTREC.
  02 FILLER  PIC X(5).
  02 OUTREC1 PIC 9(6).
  02 FILLER  PIC X(5).
  02 OUTREC2 PIC Z,ZZZ,ZZ9.
WORKING-STORAGE SECTION.
01 HUKAREC   ADDRESSED BY REC-AD1.
  02 HUKAREC1 PIC 9(3).
  02 HUKAREC2 PIC 9(7).
77 KAKUNO   ADDRESS.
PROCEDURE   DIVISION.
  :
  READ INFIL.
  :
  COMPUTE REC-AD1 = FUNCTION ADDR(HUKA). ... (1)
  MOVE HUKAREC2 TO OUTREC2.           ... (2)
  :
  COMPUTE KAKUNO = REC-AD1             ... (3)
  COMPUTE REC-AD1 = REC-AD1 + 10.     ... (4)
  MOVE HUKAREC2 TO OUTREC2.           ... (5)

```

(1)の COMPUTE 文で REC-AD1 にアドレスを設定する（書き方 1）ことによって、(2)の MOVE 文で HUKAREC2 という名前で、HUKA の 4 バイト目から 7 バイトのデータ項目を参照できる。

(4)の COMPUTE 文で REC-AD1 のアドレスを更新する（書き方 2）ことによって、(5)の MOVE 文で HUKAREC2 という名前で、HUKA の 14 バイト目から 7 バイトのデータ項目を参照できる。

(3)の COMPUTE 文では、KAKUNO に REC-AD1 のアドレスを一時退避している。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.8 COMPUTE 文」

15.4.3 アドレス型関数（アドレス操作機能）

アドレス型関数には、字類および項類の概念はない。

この関数値の属性は、アドレスデータ項目とする。

- アドレス型関数は、アドレスデータ項目が記述できるところに書ける。
- アドレス型関数は、アドレスデータ項目の作用対象が許される場所で参照できる。

これらの規則に反した参照をした場合、重大エラーとし、この関数を参照する文を無視する。

(1) ADDR 関数

形式

FUNCTION ADDR (引数1)

機能

ADDR 関数は、引数 1 で指定したデータ項目のアドレスを返す。この関数の型はアドレス型である。

一般規則

1. 引数について

(a) 引数 1 は、指標名、アドレス名を除く一意名でなければならない。ただし、関数一意名は指定できない。

(b) 引数 1 がブール項目のときは、バイト境界に調整しておかなければならない。

(c) -UniObjGen オプションを指定した場合、引数 1 に UTF-8 で多バイトとなる文字を含む英数字定数を指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

2. 関数値について

(a) 引数 1 が一意名のとき、関数値は、指定された引数 1 の占める記憶領域の先頭番地（アドレス）である。

(b) 引数 1 が英数字定数のとき、関数値は、指定された引数 1 の外部プログラムの先頭番地（アドレス）である。なお、Windows(x86) COBOL2002 の場合は、呼び出し規約が CDECL 属性の外部プログラムだけが対象となる。

3. 関数値の表現形式

アドレスデータ項目

4. 関数の定義および関数の表現形式

(a) 関数の定義については、マニュアル「COBOL2002 言語 標準仕様編」[11.5 関数の要約]、関数の表現形式については、マニュアル「COBOL2002 言語 標準仕様編」[11.6 関数の表現形式]を参照のこと。

16

C 言語インタフェース機能

この章では、C 言語とのインタフェースをとるための機能について説明する。

16.1 全般規則 (C 言語インタフェース機能)

C 言語で記述されたプログラムから COBOL プログラムの呼び出し, COBOL プログラムから C 言語で記述されたプログラムの呼び出しを行う場合に使用する。

16.2 データ部 (C 言語インタフェース機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

16.2.1 EXTERNAL 句 (C 言語インタフェース機能)

形式

IS EXTERNAL

機能

EXTERNAL 句で指定されたデータ領域は、C プログラムからの参照、更新が行える。

一般規則

1. 外部変数名称はデータ領域の名称を使用する。ただし、C プログラムと共用するためには、データ領域の名称に 2 バイトコード、ハイフン (-)、下線 (_)、#などの、C 言語の変数で使用できない文字を指定してはならない。
2. C プログラムから参照、更新される EXTERNAL データ項目については INDEXED BY 指定の記述を行ってはならない。
3. 作業場所節のレコード記述項に記述した EXTERNAL 句の左辺として指定されたデータ名は、翻訳単位内で記述された外部プログラム名 (PROGRAM-ID と CALL, CANCEL) と同一であってはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.25 EXTERNAL 句」

16.2.2 COMPUTATIONAL-5 (C 言語インタフェース機能)

形式

[USAGE IS] { COMPUTATIONAL-5 }
 { COMP-5 }

構文規則

1. COMPUTATIONAL-5, COMP-5 を指定された基本データ項目は、数字項目を記述する PICTURE 句の文字列によって宣言されなければならない。すなわち、PICTURE 句の文字列は、「9」、「S」、「V」および「P」だけから成っていなければならない。
2. COMPUTATIONAL-5 と COMP-5 は同義語であり、どちらを書いてもよい。

一般規則

1. COMPUTATIONAL-5, COMP-5 を指定した場合には、コンパイル時に-Comp5 オプションの指定が必要である。
2. COMPUTATIONAL-5, COMP-5 を指定した場合、データ項目の表現形式は COMP を指定した場合と同等とする。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

16.3 手続き部 (C 言語インタフェース機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

16.3.1 手続き部見出し (C 言語インタフェース機能)

形式

```
PROCEDURE DIVISION
[USING
 { [BY REFERENCE] {データ名1} ... } ... ]
 { BY VALUE {データ名2} ... }
[RETURNING データ名3].
```

機能

BY VALUE 指定は、C 言語で記述されたプログラムからパラメタを値渡しで受け取るときに指定する。また、COBOL プログラムからパラメタを値渡しで受け取る时候にも指定する。

構文規則

1. データ名 2, データ名 3 は、添字付け、修飾、部分参照されてはいけない。
2. データ名 2 は、可変長集団項目を指定してはならない。
3. 手続き部見出しの RETURNING 指定は、呼び出し元へ返す値が設定されるデータ項目を指定する。
4. データ名 3 は、連絡節でレベル 01 記述項またはレベル 77 記述項として定義されていなければならない。
5. データ名 3 は、データ名 1 またはデータ名 2 と重複してはならない。

一般規則

1. BY REFERENCE 指定, BY VALUE 指定は、別の BY REFERENCE 指定, BY VALUE 指定が現れるまでそれに続くデータ名に対して有効となる。
2. 最初のデータ名の前にどの指定もない場合、BY REFERENCE 指定が仮定される。
3. BY REFERENCE 指定は、CALL 文の BY REFERENCE 指定, BY CONTENT 指定に対応してなければならない。
4. RETURNING 指定をすると、RETURN-CODE 特殊レジスタを用いて呼び出し元へ復帰情報を返せない。
5. 呼び出し元プログラムと呼び出し先プログラムが、COBOL プログラムの場合、CALL 文に RETURNING 指定がされていないならば、PROCEDURE DIVISION に RETURNING 指定をしてはいけない。CALL 文に RETURNING 指定がされているならば、PROCEDURE DIVISION に RETURNING 指定をしなければならない。

6. PROCEDURE DIVISION の RETURNING 指定のデータ項目と、呼び出し元の CALL 文の RETURNING 指定のデータ項目の長さおよび用途は同じでなければならない。
7. データ名 3 は、メインプログラム (-Main オプションによって main 関数が生成されたプログラム) の場合、8 バイトを超えるデータ項目を指定してはならない。

16.3.2 CALL 文 (BY VALUE 指定) (C 言語インタフェース機能)

形式

書き方 1

```
CALL {一意名1
      定数1
      [BY REFERENCE] {一意名2}...
      USING {BY CONTENT {一意名2}...
            BY VALUE {一意名3
                      定数2}...
            }...
      [RETURNING 一意名4]*
      [ON OVERFLOW 無条件文1]
      [END-CALL].
```

書き方 2

```
CALL {一意名1
      定数1
      [BY REFERENCE] {一意名2}...
      USING {BY CONTENT {一意名2}...
            BY VALUE {一意名3
                      定数2}...
            }...
      [RETURNING 一意名4]*
      [ON EXCEPTION 無条件文1]
      [NOT ON EXCEPTION 無条件文2]
      [END-CALL].
```

注※

CALL 文で定数 1 が指定されたときだけ指定できる。

機能

BY VALUE 指定は、COBOL プログラムから COBOL プログラムまたは C 言語で記述されたプログラムを呼び出す場合に、COBOL プログラムまたは C 言語のプログラムでパラメタを値渡しで受け取る時に指定する。

構文規則

- 一意名 3、一意名 4 は、部分参照されてはいけない。
- 一意名 3 は、可変長集団項目を指定してはならない。
- 定数 2 は、英数字定数、数字定数または表意定数 ZERO でなければならない。
- 定数 2 が数字定数のとき、9 けた以下で小数点の指定があってはならない。

5. CALL 文の RETURNING を指定すると、COBOL プログラムから COBOL プログラムまたは C 言語で記述されたプログラムを呼び出す場合に、復帰情報以外の値を受け取れる。
6. CALL 文の RETURNING 指定には、呼び出し先から返された値を設定するデータ項目を指定する。

一般規則

1. BY REFERENCE 指定, BY VALUE 指定は, 別の BY REFERENCE 指定, BY VALUE 指定が現れるまで, それに続くデータ名に対して有効となる。
2. 最初のデータ名の前に BY VALUE, BY REFERENCE, BY CONTENT のどの指定もない場合, BY REFERENCE 指定が仮定される。
3. PROCEDURE DIVISION の BY REFERENCE 指定は, CALL 文の BY REFERENCE 指定, BY CONTENT 指定に対応していなければならない。
4. CALL BY VALUE の定数と PROCEDURE BY VALUE の受け取り側作用対象との関係および BY VALUE 指定のデータ項目と C 言語の引数指定の関係については, マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。
5. 呼び出し元プログラムと呼び出し先プログラムが, COBOL プログラムの場合, CALL 文で RETURNING 指定をすると, 呼び出し先で復帰情報が設定されていても, 呼び出し元の, RETURN-CODE 特殊レジスタで復帰情報を参照できない。
6. 呼び出し元プログラムと呼び出し先プログラムが COBOL プログラムの場合, 手続き部見出しに RETURNING 指定がされていないならば, CALL 文に RETURNING 指定をしてはいけない。手続き部見出しに RETURNING 指定がされているならば, CALL 文に RETURNING 指定をしなければならない。
CALL 文の RETURNING 指定のデータ項目と, 呼び出し先プログラムの手続き部見出しのデータ項目の長さおよび用途は同じでなければならない。
7. 同一外部プログラムを複数の CALL 文に指定した場合, RETURNING の指定ありと, 指定なしが混在してはならない。また, RETURNING の指定がある場合はそれぞれの RETURNING に指定されたデータ項目の長さおよび用途は同じでなければならない。
8. 外部プログラムを呼び出す場合, RETURNING に指定されたデータ項目の属性で戻り値を参照する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

17

CSV ファイル入出力機能

この章では、CSV ファイル入出力機能の形式について説明する。

17.1 概説 (CSV ファイル入出力機能)

表計算プログラムの CSV (Comma Separated Value) 形式のファイルを COBOL で入出力する機能について説明する。

-UniObjGen オプションを指定した場合の詳細はマニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

17.1.1 言語の概念 (CSV ファイル入出力機能)

CSV ファイルは、表計算プログラムのアクティブシートだけを 1 行ごとにコンマで区切って出力した形式のテキストファイルである。アクティブシートの行の位置はレコードの順番に対応し、列の位置はレコード中にコンマで区切られた位置に対応する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「5.1.7(1) 順編成」

17.2 READ 文・WRITE 文による入出力 (CSV ファイル入出力機能)

17.2.1 環境部 (CSV ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

(1) ファイル管理段落

このシステムでは、CSV ファイルをテキストファイルの拡張機能としてサポートしている。ここでは、テキストファイルと異なる規則だけを記述する。

形式

[ORGANIZATION IS] CSV

機能

ORGANIZATION IS CSV は、ファイルの論理的構成が CSV のファイルであることを指定する。

一般規則

1. ORGANIZATION IS CSV は、ファイルの論理的構成が CSV のファイルであることを指定する。ファイル編成はファイルを生成するときに定まり、後から変更できない。
2. CSV ファイルは、テキストファイルと同様に、レコードがファイルに書き出される時に定まる前後の関係によって一意となる永続的な論理ファイル構造である。
3. ORGANIZATION IS CSV が指定されたとき、ファイルはおおのデータを 1 行ずつ含む可変長レコードの集まりとして扱われる。1 行とは、文字の並びであり、その終わりはレコード終端コードである。終端コードはレコードの長さに含まれない。
4. ファイル共用機能の構文を書いてもよいが、このシステムでは覚え書きとする。

【テキストファイル機能との関連】

「2.2.1 ファイル管理段落 (テキストファイル機能)」

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.3 ファイル管理段落 (FILE-CONTROL)」

17.2.2 データ部 (CSV ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

(1) ファイル記述項 (FD)

ここでは、テキストファイルと異なる規則だけを記述する。

形式

```
FD ファイル名1
  [IS EXTERNAL]
  [IS GLOBAL]
  [RECORD { CONTAINS 整数1 CHARACTERS
            IS VARYING IN SIZE
            [[FROM 整数2][TO 整数3] CHARACTERS]
            CONTAINS 整数4 TO 整数5 CHARACTERS } ]
  [DATA { RECORD IS
          RECORDS ARE } データ名1].
```

構文規則

1. ORGANIZATION IS CSV が指定されたファイルの FD 記述項には、DEPENDENDING ON 指定の RECORD IS VARYING IN SIZE 句を指定できない。
2. LINAGE 句は指定できない。
3. ファイル記述項に従属するレコード記述は一つでなければならない。
4. レコード記述項は DEPENDENDING ON 指定の OCCURS 句を含んでいてはならない。
5. レコード記述に従属する基本項目は、ブール項目以外の用途が表示用の項目でなければならない。
6. レコード記述項に CHARACTER TYPE 句を含んでいてはならない。なお、UNIX COBOL2002 では-XMAP,LinePrint オプション未指定時は、CHARACTER TYPE 句は覚え書きとする。
7. -NumCsv オプション指定時、レコード記述に従属する基本項目が数字編集項目の場合、その項目は数字入出力の対象にならない。英数字項目として処理される。

一般規則

1. レコード記述に REDEFINES 句を含んでいてもよいが、READ 文・WRITE 文では再定義した方は無視され、再定義された方の属性で処理される。

【テキストファイル機能との関連】

「2.3.1 ファイル記述項 (テキストファイル機能)」

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.3.4 ファイル記述項 (file description entry)」

17.2.3 手続き部 (CSV ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

(1) CLOSE 文 (CSV ファイル入出力機能)

CLOSE 文についてはテキストファイルの規則と同じである。

【テキストファイル機能との関連】

[2.4.1 CLOSE 文 (テキストファイル機能)]

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.6 CLOSE 文」

(2) OPEN 文 (CSV ファイル入出力機能)

テキストファイルと異なる規則だけを説明する。

構文規則

1. ORGANIZATION IS CSV が指定されたファイルに対して指定できるモードは入力・出力・拡張だけである。

【テキストファイル機能との関連】

[2.4.2 OPEN 文 (テキストファイル機能)]

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.30 OPEN 文」

(3) READ 文 (CSV ファイル入出力機能)

テキストファイルと異なる規則だけを説明する。

形式

```
READ ファイル名1 [NEXT] RECORD  
[INTO 一意名1]  
[AT END 無条件文1]  
[NOT AT END 無条件文2]  
[END-READ].
```

構文規則

1. 一意名 1 は集団項目またはブール項目以外で用途が表示用の項目でなければならない。一意名が集団項目のとき、従属する基本項目はブール項目以外の、用途が表示用の項目でなければならない。
2. 一意名 1 は部分参照してはならない。
3. 一意名 1 に DEPENDING ON 指定の OCCURS 句を含んではならない。
4. 一意名 1 は再命名項目であってはならない。
5. -NumCsv オプション指定時、一意名 1 の長さはファイル名 1 に従属するレコード長以下でなければならない。
6. -NumCsv オプション指定時、一意名 1 または一意名 1 に従属する基本項目が数字編集項目の場合、その項目は数字入力の対象にならない。英数字項目として処理される。

一般規則

1. 表計算プログラムの CSV ファイルの 1 行は、コンマを区切り文字として、ファイル名 1 または一意名 1 に従属する基本項目の先頭から 1 対 1 に対応して、左詰めで格納される。ただし、-NumCsv オプション指定時はこの規則は適用されない。-NumCsv オプション指定については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の CSV 編成ファイルの説明を参照のこと。
2. INTO 指定がある場合は一意名 1 のレコード構造に従って処理され、INTO 指定がない場合は、ファイル名 1 に従属するレコードの形式に従って処理される。
3. 表計算プログラムの CSV ファイルの 1 行中のセルの個数がファイル名 1 または一意名 1 に従属する基本項目の個数より多いとき、基本項目と対応しないセルは無視される。セルの個数がデータ名 1 に従属する基本項目の個数より少ないとき、セルと対応しない基本項目の値は READ 文の実行前の値から変更されない。
4. READ 文を実行したときに読み込むレコードがない場合、AT END 指定があれば無条件文 1 に制御が移る。その後、各文の実行規則に従って実行が続く。制御の明示移行を起こす手続き分岐文または条件文が実行されると、その文の規則に従って制御が移る。そうでない場合、AT END 指定の無条件文の実行が完了し、その入出力文の終わりに制御が移る。このとき、NOT AT END 指定があっても無視される。
5. READ 文が成功した場合、NOT AT END 指定がなければその入出力文の終わりに制御が移る。NOT AT END 指定があれば、無条件文 2 に制御が移る。その後、各文の規則に従って実行が続く。制御の明示的移行を起こす手続き分岐文または無条件文が実行されると、その文の規則に従って制御が移る。そうでない場合、NOT AT END 指定の無条件文の実行が完了してその入出力文の終わりに制御が移る。
6. 一意名 1 に REDEFINES 句が含まれていてもよいが、再定義した方は無視され、再定義された方の属性で処理される。このとき、再定義した方の項目はブール項目以外で、用途が表示用でなければならない。

【テキストファイル機能との関連】

「2.4.3 READ 文 (テキストファイル機能)」

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」

(4) WRITE 文 (CSV ファイル入出力機能)

テキストファイルと異なる規則だけを記述する。

形式

```
WRITE レコード名1  
[FROM 一意名1]  
[ { BEFORE } ADVANCING { 一意名2 } [ LINE ] ]  
[ { AFTER } { 整数1 } [ LINES ] ]  
[END-WRITE].
```

構文規則

- 1.一意名 1 は集団項目またはブール項目以外で用途が表示用の基本項目でなければならない。一意名 1 が集団項目のとき、従属する基本項目はブール項目以外の用途が表示用の項目でなければならない。
- 2.一意名 1 は部分参照してはならない。
- 3.一意名 1 に DEPENDING 指定の OCCURS 句を含んではならない。
- 4.一意名 1 は再命名項目であってはならない。
- 5.PAGE 指定の ADVANCING 指定を書いてはならない。
- 6.END-OF-PAGE 指定を書いてはならない。
- 7.一意名 1 または一意名 1 に従属する項目に CHARACTER TYPE 句の指定があってはならない。なお、UNIX COBOL2002 では-XMAP,LinePrint オプション未指定時は,CHARACTER TYPE 句は覚え書きとする。
- 8.FROM 句に定数を指定してはならない。
- 9.-NumCsv オプション指定時、一意名 1 の長さはレコード名 1 の長さ以下でなければならない。
- 10.-NumCsv オプション指定時、レコード記述に従属する基本項目が数字編集項目の場合、その項目は数字入出力の対象にならない。英数字項目として処理される。

一般規則

- 1.レコード名 1 または一意名 1 に従属する基本項目の先頭からコンマで区切って出力する。ただし、-NumCsv オプション指定時はこの規則は適用されない。-NumCsv オプション指定については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の入出力手続き文と動作の説明を参照のこと。
- 2.FROM 指定がある場合は一意名 1 のレコード構造に従って処理され、FROM 指定がない場合はレコード名 1 の構造に従って処理される。
- 3.ORGANIZATION IS CSV が指定されたファイルに対して WRITE 文を実行した場合、ADVANCING 指定がないと自動的に一行送りが行われる。ADVANCING 指定がある場合、テキストファイルに対する規則に従って行送りが行われる。
- 4.一意名 1 に REDEFINES 句を含んでもよいが、再定義した方は無視され、再定義された方の属性で処理される。このとき、再定義した方の項目はブール項目以外で用途が表示用でなければならない。

【テキストファイル機能との関連】

[2.4.5 WRITE 文 (テキストファイル機能)]

【標準仕様との関連】

[COBOL2002 言語 標準仕様編] [10.8.55 WRITE 文]

17.2.4 入出力状態の値 (CSV ファイル入出力機能)

テキストファイルと異なる値だけ説明する。

- 入出力状態 = 04

CSV ファイルから入力された一つのセルの文字列長が、入力領域よりも長い。または、数値データのけたあふれが発生した。

【テキストファイル機能との関連】

「2.5 入出力状態の値 (テキストファイル機能)」

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「5.1.12 入出力状態」

17.3 ACCEPT 文・DISPLAY 文による入出力 (CSV ファイル入出力機能)

ACCEPT 文, DISPLAY 文を使用した CSV ファイルの入出力機能は, PC COBOL85 で作成したプログラムを移行するために用意されている古い仕様である。このため, COBOL2002 で新規にプログラムを作成する場合は, READ 文または WRITE 文による CSV ファイル入出力機能を利用することを推奨する。

UNIX COBOL2002 では, ACCEPT 文, DISPLAY 文を使用した CSV ファイルの入出力機能は使用できない。

17.3.1 環境部 (SPECIAL-NAMES) (CSV ファイル入出力機能)

形式

SPECIAL-NAMES.
SYSCSV IS 呼び名2

機能

表計算プログラムの CSV ファイルを示すシステム名 (機能名) である SYSCSV と呼び名 2 を関連づける。

構文規則

標準仕様を参照のこと。

一般規則

標準仕様を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落 (SPECIAL-NAMES)」

17.3.2 手続き部 (CSV ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

(1) ACCEPT 文 (CSV ファイル入出力機能)

形式

ACCEPT 一意名1 FROM 呼び名1.

機能

表計算プログラムの CSV ファイルの 1 行を入力する。

構文規則

1. 一意名 1 は強く型付けされていない集団項目または英数字項目でなければならない。一意名 1 が集団項目のとき、従属する基本項目はすべて英数字項目でなければならない。
2. 一意名 1 は部分参照してはならない。
3. 一意名 1 に DEPENDING 指定の OCCURS 句を含んではならない。
4. 一意名 1 は、再命名項目であってはならない。
5. 呼び名 1 は SYSCSV に関連づけられていなければならない。

一般規則

1. 表計算プログラムの CSV ファイルの 1 行はコンマを区切り文字として一意名 1 または一意名 1 に従属する基本項目の先頭から 1 対 1 に対応して、左詰めで格納される。
2. 表計算プログラムの CSV ファイルの 1 行中のセルの個数が一意名 1 に従属する基本項目個数より多いとき、基本項目と対応しないセルは無視される。セルの個数が一意名 1 に従属する基本項目の個数より少ないとき、セルと対応しない基本項目の値は ACCEPT 文の実行前の値から変更されない。
3. 一意名 1 に REDEFINES 句を含んでいる場合、再定義した方は無視され、再定義された方の属性で処理される。このとき、再定義した方は英数字項目でなければならない。
4. 入力データに引用符 (") が含まれない場合、コンマまたは改行文字 (Windows COBOL2002 では X'0D0A', UNIX COBOL2002 では X'0A') を区切りとしてデータ入力を行う。
5. 入力データに引用符 (") が含まれる場合、次の規則に従ってデータ入力を行う。
 - ・コンマの直後が引用符 (") の場合、次の引用符 (") の直前までをデータとして扱う。
 - ・コンマの直後が引用符 (") でない場合、データ中に含まれる引用符 (") は、文字データとして扱われる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

(2) DISPLAY 文 (CSV ファイル入出力機能)

形式

`DISPLAY` 一意名1 `UPON` 呼び名1.

機能

表計算プログラムの CSV ファイルの 1 行を出力する。

構文規則

1. 一意名 1 は集団項目またはブール項目以外の用途が表示用の基本項目でなければならない。一意名 1 が集団項目のとき、従属する基本項目はブール項目以外で、用途はすべて表示用でなければならない。
2. 一意名 1 は部分参照してはならない。
3. 一意名 1 に DEPENDING 指定の OCCURS 句を含んではならない。

4. 一意名 1 は、再命名項目であってはならない。
5. 呼び名 1 は SYSCSV に関連づけられていなければならない。

一般規則

1. 一意名 1 または一意名 1 に従属する基本項目の先頭からコンマで区切って出力する。
2. 一意名 1 に REDEFINES 句を含んでいる場合、再定義した方は無視され、再定義された方の属性で処理される。このとき、再定義した方の項目はブール項目以外の用途が表示用でなければならない。
3. 出力データを引用符 (") で囲んで出力する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

18

OLE2 オートメーションインタフェース機能

OLE2 オートメーションインタフェース機能は、OLE2 オートメーションに対応しているアプリケーションが公開している OLE オブジェクトを COBOL のプログラムで操作するための機能である。

この章では、OLE2 オートメーションインタフェース機能を使用するための言語の形式や規則、組み込み関数について説明する。なお、OLE2 オートメーションインタフェース機能は、クラス定義、インタフェース定義、利用者定義関数の中では使用できない。

18.1 概説 (OLE2 オートメーションインタフェース機能)

OLE2 オートメーションインタフェース機能は、OLE2 オートメーションサーバアプリケーションが公開している OLE オブジェクトを COBOL のプログラムで操作するための機能である。

例えば、Excel の OLE オブジェクトを操作することで、Excel で作成したシートを COBOL プログラムで印刷できる。

UNIX COBOL2002 では、OLE2 オートメーションインタフェース機能は使用できない。

18.1.1 言語の概念 (OLE2 オートメーションインタフェース機能)

(1) OLE オブジェクト

OLE オブジェクトは、操作する対象であり、OLE プロパティと OLE メソッドを持つ。OLE オブジェクトがどんな OLE プロパティや OLE メソッドを持っているかは、OLE オブジェクトごとに異なる。

(2) OLE プロパティ

OLE プロパティは、OLE オブジェクトの状態または属性であり、値を設定して OLE オブジェクトの状態を変更したり、値を取得して OLE オブジェクトの状態を調べることができる。

(3) OLE メソッド

OLE メソッドは、OLE オブジェクトに対する操作であり、OLE オブジェクトまたは OLE オブジェクト内のデータに対して、指定した操作を実行できる。OLE メソッドによっては、別の OLE オブジェクトを戻り値として返すものもある。

(4) インスタンス

インスタンスは、実行中のアプリケーションの OLE オブジェクトであり、アプリケーションによっては同時に複数のインスタンスを生成することもできる。

(5) コレクション

コレクションは、関連する OLE オブジェクトの集合であり、コレクション自体も一つの OLE オブジェクトである。

コレクション中の一つの OLE オブジェクトを指定するために、コレクションに特定の OLE オブジェクトを表す文字列や、インデクスを指定することで、一つの OLE オブジェクトを参照できる。

(6) コンテナ

コンテナは、ほかの OLE オブジェクトを含んでいる OLE オブジェクトである。OLE オブジェクトへの参照に、そのコンテナへの参照を含む場合は、コンテナと OLE オブジェクトをピリオド (.) で区切らなければならない。

(7) VARIANT 値

VARIANT 値は、OLE2 オートメーション機能（「OLE2 オートメーションインタフェース機能」の手続き文、「サービスルーチンを使った OLE2 オートメーションクライアント機能」）で取り扱うデータであり、さまざまな型のデータを扱うことができる。VARIANT 値には、COBOL が取り扱うデータ項目の項類の概念は適用されない。

(8) OLE オブジェクト参照データ項目

OLE オブジェクト参照データ項目は、USAGE 句で用途が OLE オブジェクト参照 (OBJECT REFERENCE OLE) と指定された項目である。インスタンス化された OLE オブジェクトを保持するために、4 バイト (Windows(x64) COBOL2002 では 8 バイト) の記憶領域が確保される。OLE オブジェクト参照データ項目には、COBOL が取り扱うデータ項目の項類の概念は適用されない。

(9) バリエーションデータ項目

バリエーションデータ項目は、USAGE 句で用途がバリエーション (VARIANT) と指定された項目である。

OLE2 オートメーションインタフェース機能で OLE オブジェクトの OLE プロパティや OLE メソッドを操作する場合には、すべて VARIANT 値で受け渡しをする。INVOKE 文や SET 文で取り扱う VARIANT 値を COBOL のデータ項目に変換せずに、そのまま VARIANT 値として格納するためにバリエーションデータ項目を使用する。バリエーションデータ項目には、VARIANT 値そのものではなく、VARIANT 値を指すポインタが実際には格納される。バリエーションデータ項目には、COBOL が取り扱うデータ項目の項類の概念は適用されない。

18.1.2 VARIANT 値と COBOL データの変換規則 (OLE2 オートメーションインタフェース機能)

(1) COBOL データから VARIANT 値への変換規則

OLE メソッドに渡す引数や OLE プロパティへの設定で指定された COBOL のデータ項目は、次に示す変換規則に従って暗黙的に VARIANT 値に変換される。

表 18-1 COBOL データ項目から VARIANT 値への変換規則

COBOL データ項目	VARIANT 値 ^{※5}
表意定数 (NULL)	VT_NULL (1)
2 進項目 ^{※1} , 外部 10 進項目 ^{※1} , 内部 10 進項目 ^{※1} , 数字定数 ^{※1} , 表意定数 (ZERO)	VT_I2 (2)
2 進項目 ^{※2} , 外部 10 進項目 ^{※2} , 内部 10 進項目 ^{※2} , 数字定数 ^{※2}	VT_I4 (3)
単精度内部浮動小数点項目	VT_R4 (4)
倍精度内部浮動小数点項目, 外部浮動小数点項目, 浮動小数点定数, 2 進項目 ^{※3} , 外部 10 進項目 ^{※3} , 内部 10 進項目 ^{※3} , 数字定数 ^{※3}	VT_R8 (5)
英字項目, 英数字項目, 英数字定数, 固定長集団項目 (日本語集団項目を含む), 可変長集団項目, 日本語項目, 日本語文字定数, 数字編集項目	VT_BSTR (8)
OLE オブジェクト参照データ項目	VT_DISPATCH (9)
内部ブール項目 ^{※4} , 外部ブール項目 ^{※4} , ブール定数 ^{※4}	VT_I2→VT_BOOL (11)

注 1
 組み込み関数の CONVERT-TO-VARIANT 関数での明示的な変換も、いったん上表の変換規則に従った VARIANT 値に変換し、さらに目的の VARIANT 値へと変換する。

注 2
 オブジェクト参照データ項目、アドレスデータ項目および指標データ項目は、バリエーションデータ型には変換できない。

注※1
 4 けたまでの小数けたなし項目。

注※2
 9 けたまでの小数けたなし項目。

注※3
 10 けたを超えるまたは小数けたがある項目。

注※4
 1 けたの項目。

注※5
 括弧内の数字は VARIANT のタイプを示す。

(2) VARIANT 値から COBOL データへの変換規則

OLE メソッドの戻り値や OLE プロパティの取得で受け取った VARIANT 値は、次に示す変換規則に従って暗黙的に COBOL のデータ項目に変換される。

表 18-2 VARIANT 値から COBOL データ項目への変換規則

VARIANT 値※5	COBOL データ項目
VT_I2 (2)	2 進項目※1, 外部 10 進項目※1, 内部 10 進項目※1
VT_I4 (3)	2 進項目※2, 外部 10 進項目※2, 内部 10 進項目※2
VT_R4 (4)	単精度内部浮動小数点項目
VT_R8 (5)	倍精度内部浮動小数点項目, 外部浮動小数点項目, 2 進項目※3, 外部 10 進項目※3, 内部 10 進項目※3
VT_BSTR (8)	英字項目, 英数字項目, 固定長集団項目 (日本語集団項目を含む), 可変長集団項目, 日本語項目
VT_DISPATCH (9)	OLE オブジェクト参照データ項目
VT_BOOL (11) →VT_I2	内部ブール項目※4, 外部ブール項目※4

注 1

組み込み関数の VARIANT 値から COBOL データ項目への明示的な変換も、いったん上表の変換規則に従った VARIANT 値に変換し、さらに目的の COBOL データ項目へと変換する。なお、変換元となる文字の右側に存在する空白は無視される。

注 2

文字列変換 (VT_BSTR 型) の場合、データは常に左端をそろえて収められる。データを設定する領域が小さい場合は右端が切り捨てられる。しかし、大きい場合の空白詰めは行われない。JUSTIFIED 句は適用されない。なお、変換元となる文字の右側に存在する空白は無視される。

注 3

バリエーションデータ型からオブジェクト参照データ項目、アドレスデータ項目および指標データ項目には変換できない。

注※1

4 けたまでの小数けたなし項目。

注※2

9 けたまでの小数けたなし項目。

注※3

10 けたを超えるまたは小数けたがある項目。

注※4

1 けたの項目。

注※5

括弧内の数字は VARIANT のタイプを示す。

18.2 データ部 (OLE2 オートメーションインタフェース機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

18.2.1 OLE オブジェクト参照データ項目 (OLE2 オートメーションインタフェース機能)

OLE オブジェクト参照データ項目は、インスタンス化された OLE オブジェクトのポインタを格納するためのデータ領域である。

形式

レベル番号 データ名 [USAGE IS]
OBJECT REFERENCE OLE.

構文規則

1. レベル番号は、01～49 または 77 でなければならない。
2. OLE オブジェクト参照データ項目は、作業場所節および連絡節に定義しなければならない。
3. OLE オブジェクト参照データ項目は、EXTERNAL 句や GLOBAL 句を指定したレコード記述中に書いてもよい。
4. OLE オブジェクト参照データ項目が独立した基本項目として定義された場合は、EXTERNAL 句や GLOBAL 句を指定できる。ただし、EXTERNAL 句や GLOBAL 句を指定できるのは、オブジェクト参照データ項目が作業場所節で定義されている場合だけである。
5. OLE オブジェクト参照データ項目に VALUE 句を指定してもよい。VALUE 句には、表意定数 ZERO と NULL が指定できる。
6. OLE オブジェクト参照データ項目に関連する条件名が記述できる。ただし、条件名の VALUE 句には表意定数 ZERO と NULL だけが指定できる。
7. USAGE 句の OBJECT 指定と同時に PICTURE 句を指定してはならない。
8. OLE オブジェクト参照データ項目を明示的に参照できる個所を次に示す。
 - ・ OLE2 オートメーションインタフェース機能での INVOKE 文
 - ・ OLE2 オートメーションインタフェース機能での SET 文
 - ・ OLE オブジェクト参照データ比較の条件式
 - ・ CONVERT-TO-VARIANT 関数

一般規則

1. OLE オブジェクト参照データ項目は内部的に 4 バイト (Windows(x64) COBOL2002 では 8 バイト) の領域が確保される。

2. OLE オブジェクト参照データ項目には、生成されたインスタンスの OLE オブジェクトのポインタを格納できる。
3. OLE オブジェクト参照データ項目に SYNCHRONIZED 句の機能が適用された場合は、4 バイト (Windows(x64) COBOL2002 では 8 バイト) 境界に割り付けられる。
SYNCHRONIZED 句の機能が適用されない場合は、バイト境界に割り付けられる。
4. OLE オブジェクト参照データ項目に OLE オブジェクトを格納するとき、すでに別の OLE オブジェクトが格納されている場合は、前に格納された OLE オブジェクトを解放した後に格納される。

18.2.2 バリエーションデータ項目 (OLE2 オートメーションインタフェース機能)

バリエーションデータ項目は、VARIANT 値へのポインタを格納するためのデータ領域である。

形式

レベル番号 データ名 [USAGE IS] VARIANT.

構文規則

1. レベル番号は、01~49 または 77 でなければならない。
2. バリエーションデータ項目は、作業場所節および連絡節に定義しなければならない。
3. バリエーションデータ項目は、EXTERNAL 句や GLOBAL 句を指定したレコード記述中に書いてもよい。
4. バリエーションデータ項目が独立した基本項目として定義された場合は、EXTERNAL 句や GLOBAL 句を指定できる。ただし、EXTERNAL 句や GLOBAL 句を指定できるのは、バリエーションデータ項目が作業場所節で定義されている場合だけである。
5. バリエーションデータ項目に VALUE 句を指定してもよい。VALUE 句には、表意定数 ZERO と NULL が指定できる。VALUE 句を指定していない場合の動作は保証しない。
6. バリエーションデータ項目に関連する条件名が記述できる。ただし、条件名の VALUE 句には表意定数 ZERO と NULL だけが指定できる。
7. USAGE 句の VARIANT 指定と同時に PICTURE 句を指定してはならない。
8. バリエーションデータ項目を明示的に参照できる個所を次に示す。
 - ・ OLE2 オートメーションインタフェース機能での INVOKE 文
 - ・ OLE2 オートメーションインタフェース機能での SET 文
 - ・ バリエーションデータ比較の条件式
 - ・ 利用者定義関数の引数
 - ・ TYPE-OF-VARIANT 関数
 - ・ VARIANT-TO-INTEGGER 関数
 - ・ VARIANT-TO-NUMERIC 関数
 - ・ VARIANT-TO-ALPHANUMERIC 関数
 - ・ VARIANT-TO-OBJECT 関数

一般規則

1. バリエントデータ項目は内部的に 4 バイト (Windows(x64) COBOL2002 では 8 バイト) の領域が確保される。
2. バリエントデータ項目を指定すると、VARIANT 値を COBOL のデータ項目に変換せずに、そのまま VARIANT 値として保持できる。また、バリエントデータ項目には、VARIANT 値そのものではなく、バリエント値を指すポインタが格納される。
3. バリエントデータ項目に SYNCHRONIZED 句の機能が適用された場合は 4 バイト (Windows(x64) COBOL2002 では 8 バイト) 境界に割り付けられる。
SYNCHRONIZED 句の機能が適用されない場合は、バイト境界に割り付けられる。
4. バリエントデータ項目に VARIANT 値へのポインタを格納する場合、すでに別の VARIANT 値へのポインタが設定されているときは、前に格納された VARIANT 値を解放した後に格納する。
5. バリエントデータ項目に EXTERNAL 句を指定した場合の注意事項については、マニュアル「COBOL2002 ユーザーズガイド」の VARIANT 値と COBOL データのやり取りの説明を参照のこと。

18.3 手続き部 (OLE2 オートメーションインタフェース機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

18.3.1 INVOKE 文 (OLE2 オートメーションインタフェース機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.26 INVOKE 文」

(1) 一般的な OLE メソッドの操作

形式

$$\text{INVOKE 一意名7} \left\{ \begin{array}{l} \text{一意名8} \\ \text{定数3} \end{array} \right\}$$
$$\left[\begin{array}{l} \text{USING [BY VALUE} \\ \left\{ \left\{ \begin{array}{l} \text{一意名9} \\ \text{定数4} \end{array} \right\} \left[\begin{array}{l} \text{NAMED} \left\{ \begin{array}{l} \text{一意名10} \\ \text{定数5} \end{array} \right\} \right] \dots \dots \end{array} \right. \end{array} \right]$$
$$\text{[RETURNING 一意名11].}$$

機能

OLE2 オートメーションサーバアプリケーションの一般的な OLE メソッド操作 (CREATEOBJ, GETOBJ 以外) をする。

構文規則

- 一意名 7 は OLE オブジェクト参照データ項目でなければならない。
- 一意名 8 および一意名 10 は、集団項目、英字、英数字、日本語項目でなければならない。
- 定数 3 および定数 5 は、英数字定数または日本語文字定数でなければならない。
- 一意名 9、一意名 11 には、集団項目、数字項目、英字項目、英数字項目、日本語項目、1 けたのブール項目、OLE オブジェクト参照データ項目およびバリエーションデータ項目が指定できる。ただし、一意名 11 は強く型付けされた集団項目を指すものであってはならない。
- 一意名 9 には数字編集項目も指定できる。
- 定数 4 には、数字定数、英数字定数、日本語文字定数、1 けたのブール定数および表意定数 (ZERO, NULL) が指定できる。
- UniObjGen オプションを指定した場合、一意名 8、一意名 9、一意名 10 および一意名 11 には用途が NATIONAL の項目および用途が NATIONAL の項目を従属項目として含む集団項目は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

8. -UniObjGen オプションを指定した場合、定数 3、定数 4 および定数 5 には日本語文字定数および 16 進日本語文字定数は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

一般規則

1. OLE2 オートメーションサーバアプリケーションの OLE メソッド操作では、一意名 7 に、インスタンスの OLE オブジェクトが設定されている OLE オブジェクト参照データ項目を指定しなければならない。一意名 7 に OLE オブジェクトが設定されていない場合は、実行時にエラーとなる。
2. 一意名 7 に設定されている OLE オブジェクトを起点として、一意名 8 または定数 3 で指定された OLE メソッドが操作される。
3. 一意名 8 や定数 3 に指定する OLE メソッドには、起点となる OLE オブジェクトからのコンテナやコレクションを含んで指定しなければならない。
4. 一意名 9、定数 4 には、OLE メソッドへの引数を指定する。引数は指定された順番に OLE メソッドに渡される。引数を省略して指定する場合は、表意定数 NULL を指定する。一意名 9、定数 4 で指定した COBOL のデータ項目は、「表 18-1 COBOL データ項目から VARIANT 値への変換規則」の変換規則に従い VARIANT 値に変換されて OLE メソッドに渡される。一意名 9 でバリエーションデータ項目を指定した場合は、VARIANT 値は変換されずにそのまま OLE メソッドに渡される。
5. 一意名 11 には OLE メソッドからの戻り値が設定される。
 - ・戻り値が OLE オブジェクトを戻す場合は、OLE オブジェクト参照データ項目を指定する。
 - ・戻り値が OLE オブジェクトでない場合は、COBOL のデータ項目を指定するか、またはバリエーションデータ項目を指定する。

COBOL のデータ項目を指定した場合は、「表 18-2 VARIANT 値から COBOL データ項目への変換規則」の変換規則に従い、VARIANT 値から COBOL のデータ項目に変換されて設定される。VARIANT 値から変換できない COBOL のデータ項目が指定されている場合は、実行時にエラーとなる。

また、バリエーションデータ項目が指定された場合は、取得された VARIANT 値へのポインタが設定される。

一意名 11 に 10 けたを超えるまたは小数けたがある数字項目（固定小数点形式の数字項目）を指定した場合は、内部的に確保した倍精度内部浮動小数点項目領域に格納された戻り値が、一意名 11 に設定される。

この結果、システムによる丸め制御の相違によって、浮動小数点の転記の結果が異なる場合がある。丸め制御については、「6.8.1 文に共通な指定と規則（浮動小数点形式データを扱う機能）」の「(7) 浮動小数点の丸め制御」を参照のこと。

6. 一意名 10 および定数 5 には、OLE メソッドの名前付き引数の名称を指定する。
7. 名前付き引数を使うと、あらかじめ指定されている名前を持った引数を指定できるため、構文で決まっている順序とは関係なく、必要な引数だけを任意の順序で指定できる。

(2) CREATEOBJ メソッドの操作

形式

`INVOKE` { 一意名10
 定数6 } 'CREATEOBJ'
`RETURNING` 一意名11.

機能

CREATEOBJ メソッドでは、OLE オブジェクトのインスタンスを生成する。

構文規則

- 一意名 10 は、英字項目、英数字項目または日本語項目でなければならない。
- 定数 6 は、英数字定数または日本語文字定数でなければならない。
- 一意名 11 は、OLE オブジェクト参照データ項目でなければならない。
- UniObjGen オプションを指定した場合、一意名 10 には用途が NATIONAL の項目および用途が NATIONAL の項目を従属項目として含む集団項目は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。
- UniObjGen オプションを指定した場合、定数 6 には日本語文字定数および 16 進日本語文字定数は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

一般規則

- 一意名 10 または定数 6 には、操作の対象となる OLE2 オートメーションサーバアプリケーションのクラス名を設定する。
- 生成されたインスタンスの OLE オブジェクトは、一意名 11 で指定された OLE オブジェクト参照データ項目に格納される。操作の対象となる OLE2 オートメーションサーバアプリケーションのクラス名でない値が設定されている場合は、実行時にエラーとなる。
- OLE オブジェクトのインスタンスがすでに生成されて、OLE オブジェクト参照データ項目に値が格納されていても、再実行された場合は、新たに OLE オブジェクトのインスタンスを生成し、OLE オブジェクト参照データ項目に新しい OLE オブジェクトが格納される。
- 生成した OLE オブジェクトが不要になった場合は、OLE2 オートメーションサーバアプリケーションが提供する終了メソッドで解放しなければならない。

(3) GETOBJ メソッドの操作

形式

```
INVOKE { 一意名12 } 'GETOBJ'  
        { 定数7 }  
[ USING BY VALUE { 一意名13 } ]  
                 { 定数8 }  
RETURNING 一意名14.
```

機能

GETOBJ メソッドでは、実行中の OLE オブジェクトを取得したり、ファイルから OLE オブジェクトのインスタンスを生成する。

構文規則

- 一意名 12 および一意名 13 は、英字項目、英数字項目または日本語項目でなければならない。
- 定数 7 は、英数字定数、日本語文字定数または表意定数 NULL でなければならない。
- 定数 8 は、英数字定数または日本語文字定数でなければならない。
- 一意名 14 は、OLE オブジェクト参照データ項目でなければならない。
- UniObjGen オプションを指定した場合、一意名 12 および一意名 13 には用途が NATIONAL の項目および用途が NATIONAL の項目を従属項目として含む集団項目は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。
- UniObjGen オプションを指定した場合、定数 7 および定数 8 には日本語文字定数および 16 進日本語文字定数は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

一般規則

- 一意名 12 または定数 7 には、操作の対象となる OLE2 オートメーションサーバアプリケーションのクラス名を設定する。
- 一意名 13 または定数 8 には、ディスク上にあるファイル名を絶対パスで指定しなければならない。
- USING 指定の有無によって、生成されているインスタンスの OLE オブジェクトを取得する方法と、新しくインスタンスの OLE オブジェクトを生成する方法がある。
 - ・ USING 指定がない場合は、すでに生成されているインスタンスから OLE オブジェクトを取得する。取得した OLE オブジェクトは、一意名 14 で指定された OLE オブジェクト参照データ項目に格納される。クラス名に対応した OLE オブジェクトのインスタンスが生成されていない場合は、実行時にエラーとなる。また、複数のインスタンスが存在する場合、どんなインスタンスの OLE オブジェクトが取得されるかは保証しない。

・ USING 指定がある場合は、指定されたファイル名からクラス名に対応した OLE オブジェクトのインスタンスを生成する。クラス名の OLE オブジェクトに、対応していないファイル名が指定された場合は、実行時にエラーとなる。

クラス名の指定がない場合（定数 7 に表示定数 NULL が指定された場合）は、ファイル名に基づいて OLE オブジェクトのインスタンスを生成する。またファイル名から OLE オブジェクトのインスタンスが生成された場合に、同じ OLE オブジェクトのインスタンスがすでに生成されて OLE オブジェクト参照データ項目に OLE オブジェクトが格納されていても、新しく生成された OLE オブジェクトが OLE オブジェクト参照データ項目に設定される。

4. 生成した OLE オブジェクトが不要になった場合は、OLE2 オートメーションサーバアプリケーションが提供する終了メソッドで解放しなければならない。

18.3.2 SET 文 (OLE2 オートメーションインタフェース機能)

形式

書き方 17 OLE プロパティからの値の取得

```
SET 一意名14 TO { 一意名15 }  
                { 定数1 }  
WITH 一意名16.
```

書き方 18 OLE プロパティへの値の設定

```
SET { 一意名17 } WITH 一意名18  
   { 定数2 }  
TO { 一意名19 }  
   { 定数3 }.
```

書き方 19 データ項目の初期化

```
SET 一意名20 TO { NULL }  
                { ZERO }.
```

書き方 20 バリエーションデータ項目の設定

```
SET 一意名21 TO 一意名22.
```

機能

書き方 17 の SET 文では、OLE プロパティからの値を取得する。書き方 18 の SET 文では、OLE プロパティへの値を設定する。書き方 19 の SET 文では、データ項目を初期化する。書き方 20 の SET 文では、バリエーションデータ項目の値をほかのバリエーションデータ項目に設定する。

書き方 17 の SET 文で受け取り側作用対象がバリエーションデータ項目の場合、または書き方 20 の SET 文の場合、受け取り側作用対象のバリエーションデータ項目にすでに VARIANT 値が存在するとき、VARIANT 値が設定された領域を解放してバリエーションデータ項目のポインタに NULL を設定後、SET 文の処理をする。

構文規則

1. 一意名 16 および一意名 18 は OLE オブジェクト参照データ項目として定義されていなければならない。
2. 一意名 15 および一意名 17 には、集団項目、英字項目、英数字項目および日本語項目が指定できる。
3. 定数 1 および定数 2 は、英数字定数または日本語文字定数でなければならない。
4. 一意名 14 および一意名 19 に指定できるのは、集団項目、数字項目、英字項目、英数字項目、日本語項目、1 けたのブール項目、バリエントデータ項目または OLE オブジェクト参照データ項目でなければならない。また、一意名 19 は、強く型付けされた集団項目を指すものであってもよい。一意名 21 または一意名 22 はバリエントデータ項目でなければならない。
5. 定数 3 は、数字定数、英数字定数、日本語文字定数、1 けたのブール定数または表意定数 (ZERO, NULL) でなければならない。
6. 一意名 20 は、OLE オブジェクト参照データ項目またはバリエントデータ項目として定義されていなければならない。
7. 一意名 19 には、数字編集項目を指定することもできる。
8. -UniObjGen オプションを指定した場合、一意名 14、一意名 15、一意名 17 および一意名 19 には用途が NATIONAL の項目および用途が NATIONAL の項目を従属項目として含む集団項目は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。
9. -UniObjGen オプションを指定した場合、定数 1、定数 2 および定数 3 には日本語文字定数および 16 進日本語文字定数は指定できない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

一般規則

書き方 17 の規則

1. 一意名 16 に設定されている OLE オブジェクトを起点として、一意名 15 または定数 1 で指定された OLE プロパティの値が取得される。
2. 一意名 16 に OLE オブジェクトが設定されていない場合は、実行時にエラーとなる。
3. 一意名 15 または定数 1 に指定する OLE プロパティは、起点となる OLE オブジェクトからのコンテナやコレクションを含んで指定しなければならない。
4. 一意名 14 には、取得した OLE プロパティ値が設定される。
5. 取得した OLE プロパティ値が OLE オブジェクトの場合は、一意名 14 で OLE オブジェクト参照データ項目を指定しなければならない。
6. OLE プロパティ値が OLE オブジェクトでない場合は、一意名 14 で COBOL のデータ項目またはバリエントデータ項目を指定する。
OLE プロパティ値に COBOL のデータ項目が指定されている場合は、「表 18-2 VARIANT 値から COBOL データ項目への変換規則」の変換規則に従い、VARIANT 値から COBOL のデータ項目に変換されて設定される。VARIANT 値から変換できない COBOL のデータ項目が指定されている場合は、実行時にエラーとなる。

また、一意名 14 でバリエントデータ項目が指定された場合は、変換されずに OLE プロパティ値へのポインタが設定される。

7. 代入元より代入先が大きい場合は、代入先の一意名 14 にオブジェクトプロパティを指定した時と、通常の名を指定した時とで結果が異なる。通常の名を指定した場合は、データ名の長さが、オブジェクトプロパティのとき、余ったデータ領域は空白で埋められる。

書き方 18 の規則

1. 一意名 18 に設定されている OLE オブジェクトを起点として、一意名 17 または定数 2 で指定された OLE プロパティに値が設定される。
2. 一意名 18 に OLE オブジェクトが設定されていない場合は、実行時にエラーとなる。
3. 一意名 17 または定数 2 に指定する OLE プロパティは、起点となる OLE オブジェクトからのコンテナやコレクションを含んで指定しなければならない。
4. 一意名 19 または定数 3 で指定された値が OLE プロパティに設定される。
5. 一意名 19 でバリエントデータ項目が指定された場合は、そのまま OLE プロパティへ値が設定される。
6. 一意名 19 に OLE オブジェクト参照データ項目または COBOL のデータ項目が指定されている場合、もしくは定数 3 が指定された場合は、「表 18-1 COBOL データ項目から VARIANT 値への変換規則」の変換規則に従って VARIANT 値に変換された後に、OLE プロパティに値が設定される。VARIANT 値へ変換できない COBOL のデータ項目が指定されている場合は、実行時にエラーとなる。

書き方 19 の規則

1. 一意名 20 で指定された OLE オブジェクト参照データ項目またはバリエントデータ項目が初期化される。
 - ・OLE オブジェクト参照データ項目が指定された場合は、設定されていたインスタンスオブジェクトを解放し、OLE オブジェクト参照データ項目を初期化する。
 - ・バリエントデータ項目が指定された場合は、ポイントされている VARIANT 値を解放し、バリエントデータ項目を初期化する。

書き方 20 の規則

1. 一意名 22 のバリエントデータ項目がポイントする VARIANT 値が、一意名 21 のバリエントデータ項目がポイントするバリエント領域へ設定される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.43 SET 文」

18.3.3 OLE オブジェクト参照データ比較の条件式 (OLE2 オートメーションインタフェース機能)

形式

$$\left\{ \begin{array}{l} \text{一意名1} \\ \underline{\text{NULL}} \\ \underline{\text{ZERO}} \end{array} \right\} \text{ IS [NOT] } \left\{ \begin{array}{l} \underline{\text{EQUAL TO}} \\ = \end{array} \right\} \left\{ \begin{array}{l} \text{一意名2} \\ \underline{\text{NULL}} \\ \underline{\text{ZERO}} \end{array} \right\}$$

構文規則

- 一意名 1 および一意名 2 は、OLE オブジェクト参照データ項目でなければならない。
- 比較対象が両方とも表意定数 NULL または ZERO であってはならない。

一般規則

- 比較対象の値が同じである場合に、等しい結果になる。
- 一意名 1 と一意名 2 の比較は OLE オブジェクト参照データ項目同士の比較となるため、OLE オブジェクトが等しい場合でも等しい結果とならない。

18.3.4 バリエーションデータ比較の条件式 (OLE2 オートメーションインタフェース機能)

形式

$$\left\{ \begin{array}{l} \text{一意名1} \\ \underline{\text{NULL}} \\ \underline{\text{ZERO}} \end{array} \right\} \text{ IS [NOT] } \left\{ \begin{array}{l} \underline{\text{EQUAL TO}} \\ = \end{array} \right\} \left\{ \begin{array}{l} \text{一意名2} \\ \underline{\text{NULL}} \\ \underline{\text{ZERO}} \end{array} \right\}$$

構文規則

- 一意名 1 および一意名 2 は、バリエーションデータ項目でなければならない。
- 比較対象が両方とも表意定数 NULL または ZERO であってはならない。

一般規則

- 比較対象の値が同じである場合に、等しい結果になる。
- 一意名 1 と一意名 2 の比較はバリエーションデータ項目同士の比較となるため、バリエーション値が等しい場合でも等しい結果とならない。

18.4 プログラム間連絡機能 (OLE2 オートメーションインタフェース機能)

18.4.1 手続き部見出し

手続き部見出しの USING BY REFERENCE の仮引数に OLE オブジェクト参照データ項目やバリエーションデータ項目を指定することで、外部で取得した OLE オブジェクトを使用して、OLE メソッドや OLE プロパティを操作できる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.1.1 一般形式」

18.4.2 CALL 文

CALL 文の USING BY REFERENCE の引数に OLE オブジェクト参照データ項目やバリエーションデータ項目を指定することで、取得した OLE オブジェクトや VARIANT 値を呼び出し先プログラムに渡すことができる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

18.5 組み込み関数機能 (OLE2 オートメーションインタフェース機能)

18.5.1 CONVERT-TO-VARIANT 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
CONVERT-TO-VARIANT (引数1 [引数2])

機能

CONVERT-TO-VARIANT (VARIANT 値変換) 関数は、引数 2 で指定した COBOL のデータ項目と等価な引数 1 で指定した型の VARIANT 値へのポインタを返す。この関数は、OLE2 オートメーションインタフェース機能を使った INVOKE 文での引数や、SET 文での設定値を、特定の型の VARIANT 値として扱う場合に使用する。この関数の型は、バリエーションデータ型である。

一般規則

1. 引数について

- ・引数 1 は、整数項目 (結果が常に整数となる算術式を含む)、整数定数、表意定数 ZERO を指定しなければならない。
- ・引数 2 は、変換する VARIANT 値の型を示す数値を指定しなければならない。指定する数値と VARIANT 値との対応を次に示す。

表 18-3 指定する数値と VARIANT 値との対応

指定する数値	VARIANT 値の型
0	Empty (VT_EMPTY)
1	Null (VT_NULL)
2	整数型 (VT_I2)
3	長整数型 (VT_I4)
4	単精度浮動小数点型 (VT_R4)
5	倍精度浮動小数点型 (VT_R8)
6	通貨型 (VT_CY)
7	日付型 (VT_DATE)
8	文字列型 (VT_BSTR)
9	オブジェクト型 (VT_DISPATCH)
11	ブール型 (VT_BOOL)

決められた数値以外のものが指定されたときの結果は保証しない。

・引数 2 は、集団項目、数字項目（算術式を含む）、数字編集項目、英字項目、英数字項目、日本語項目、1 けたのブール項目、OLE オブジェクト参照データ項目、数字定数、英数字定数、日本語文字定数、1 けたのブール定数または表意定数（ZERO、NULL）でなければならない。

VARIANT 値へ変換できないデータ項目が指定されている場合は、実行時にエラーとなる。

VARIANT 値変換では、「表 18-1 COBOL データ項目から VARIANT 値への変換規則」に示した変換規則の VARIANT 値に合わせて COBOL データ項目を指定すること。変換規則で示されていない通貨型、日付型は英数字項目で指定する。なお、日付型は 03/01/01 09:00 という形式で設定すること。

・引数 1 に 0 または 1 が指定された場合、引数 2 は無視される。2~11 が指定された場合で、引数 2 が指定されていないときはエラーとなる。

・-UniObjGen オプションを指定した場合、引数 2 に用途が NATIONAL の項目、日本語文字定数および 16 進日本語文字定数を指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

2. 関数値について

・関数値は、指定された引数と等価の VARIANT 値へのポインタが返される。

3. 関数の表現形式

・VARIANT 値へのポインタ

・この関数は、(CALL 文、INVOKE 文（オブジェクト指向機能）、利用者定義関数を除く）送り出し側作用対象となるバリエーションデータ項目が指定できるところおよびバリエーションデータ比較の条件式に記述できる。

18.5.2 TYPE-OF-PROPERTY 関数（OLE2 オートメーションインタフェース機能）

形式

FUNCTION
TYPE-OF-PROPERTY (引数1 引数2)

機能

TYPE-OF-PROPERTY（OLE プロパティや OLE メソッドの型取得）関数は、引数 1 で指定された OLE オブジェクトを起点とした、引数 2 で指定された OLE プロパティや OLE メソッドの戻り値である VARIANT 値の型を求める。この関数の型は、整数型である。

一般規則

1. 引数について

・引数 1 は、OLE オブジェクト参照データ項目を指定しなければならない。

・TYPE-OF-PROPERTY 関数では、インスタンス化された OLE オブジェクトの OLE メソッドや、OLE プロパティのタイプを取得する。引数 1 に OLE オブジェクトが設定されていない場合は、実行時にエラーとなる。

- ・引数 2 は、英字、英数字、日本語項目、英数字定数または日本語文字定数でなければならない。
- ・引数 2 には、OLE プロパティや OLE メソッドを、起点となる OLE オブジェクトからのコンテナやコレクションを含んで指定しなければならない。
- ・-UniObjGen オプションを指定した場合、引数 2 に用途が NATIONAL の項目、日本語文字定数および 16 進日本語文字定数を指定してはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。

2. 関数値について

- ・関数値は、OLE プロパティや OLE メソッドの戻り値である、VARIANT 値の型を示す数値である。数値の意味は、「表 18-3 指定する数値と VARIANT 値との対応」を参照のこと。「表 18-3 指定する数値と VARIANT 値との対応」に示される数値以外の値が返される場合は、戻される VARIANT 値とやり取りできる COBOL のデータ型は存在しない。

3. 関数の表現形式

- ・符号付き整数部 5 けたの 2 進項目
- ・この関数は、整数型関数が指定できるところに記述できる。

18.5.3 TYPE-OF-VARIANT 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
TYPE-OF-VARIANT (引数1)

機能

TYPE-OF-VARIANT (VARIANT 値の型取得) 関数は、引数 1 で指定した VARIANT 値の型を求める。この関数の型は整数型である。

一般規則

1. 引数について

- ・引数 1 は、バリエーションデータ項目を指定しなければならない。
- ・引数 1 に指定したバリエーションデータ項目が NULL の場合は、実行時にエラーとなる。

2. 関数値について

- ・関数値は、VARIANT 値の型を示す数値である。数値の意味は「表 18-3 指定する数値と VARIANT 値との対応」を参照のこと。「表 18-3 指定する数値と VARIANT 値との対応」に示される数値以外の値が返される場合は、戻される VARIANT 値とやり取りできる COBOL のデータ型は存在しない。

3. 関数の表現形式

- ・符号付き整数部 5 けたの 2 進項目
- ・この関数は、整数型関数が指定できるところに記述できる。

18.5.4 VARIANT-TO-INTEGGER 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
VARIANT-TO-INTEGGER (引数1)

機能

VARIANT-TO-INTEGGER (整数型数値変換) 関数は、引数 1 で指定した VARIANT 値を COBOL の整数型数値に変換する。この関数の型は整数型である。

一般規則

1. 引数について

- ・引数 1 は、バリエーションデータ項目を指定しなければならない。
- ・引数 1 に指定したバリエーションデータ項目が NULL の場合は、実行時にエラーとなる。
- ・引数 1 に指定したバリエーションデータ項目が整数型数値に変換できない場合は、実行時にエラーとなる。

2. 関数値について

- ・関数値は、VARIANT 値を COBOL の整数型数値に変換した値である。

3. 関数の表現形式

- ・4 バイトの 2 進項目 (PIC S9(9) USAGE COMP)
- ・この関数は、整数型関数が指定できるところに記述できる。

18.5.5 VARIANT-TO-NUMERIC 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
VARIANT-TO-NUMERIC (引数1)

機能

VARIANT-TO-NUMERIC (実数型数値変換) 関数は、引数 1 で指定した VARIANT 値を COBOL の実数型数値に変換する。この関数の型は数字型である。

一般規則

1. 引数について

- ・引数 1 は、バリエーションデータ項目を指定しなければならない。
- ・引数 1 に指定したバリエーションデータ項目が NULL の場合は、実行時にエラーとなる。
- ・引数 1 に指定したバリエーションデータ項目が実数型数値に変換できない場合は、実行時にエラーとなる。

2. 関数値について

- ・関数値は、VARIANT 値を COBOL の実数型数値に変換した値である。

3. 関数の表現形式

- ・倍精度浮動小数点項目
- ・この関数は、数値型関数が指定できるところに記述できる。

18.5.6 VARIANT-TO-ALPHANUMERIC 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
VARIANT-TO-ALPHANUMERIC (引数1)

機能

VARIANT-TO-ALPHANUMERIC (英数字型変換) 関数は、引数 1 で指定した VARIANT 値を COBOL の英数字型に変換する。この関数の型は英数字型である。

一般規則

1. 引数について

- ・引数 1 は、バリエーションデータ項目を指定しなければならない。
- ・引数 1 に指定したバリエーションデータ項目が NULL の場合は、実行時にエラーとなる。
- ・引数 1 に指定したバリエーションデータ項目が文字列に変換できない場合は、実行時にエラーとなる。

2. 関数値について

- ・関数値は、VARIANT 値を COBOL の英数字型に変換した値である。
- ・変換できる最大の文字数は 1024 バイトである。変換の結果この値を超える場合は、切り捨てられた文字列が、関数値として返される。

3. 関数の表現形式

- ・引数に依存した長さの英数字項目
- ・この関数は、英数字型関数が指定できるところに記述できる。

18.5.7 VARIANT-TO-OBJECT 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
VARIANT-TO-OBJECT (引数1)

機能

VARIANT-TO-OBJECT (OLE オブジェクト参照データ型変換) 関数は、引数 1 で指定した VARIANT 値と等価な OLE オブジェクトへのポインタを返す。この関数の型は OLE オブジェクト参照データ型である。

一般規則

1. 引数について

- ・引数 1 は、バリエーションデータ項目を指定しなければならない。
- ・引数 1 に指定したバリエーションデータ項目が NULL の場合は、実行時にエラーとなる。
- ・引数 1 に指定したバリエーションデータ項目が OLE オブジェクト参照データに変換できない場合は、実行時にエラーとなる。

2. 関数値について

- ・関数値は、VARIANT 値と等価な OLE オブジェクトへのポインタが返される。

3. 関数の表現形式

- ・OLE オブジェクトへのポインタ
- ・この関数は、(CALL 文を除く) 送り出し側作用対象の OLE オブジェクト参照データ項目が指定できるところおよび OLE オブジェクト参照データ比較の条件式に記述できる。

18.5.8 LENGTH-OF-VARIANT 関数 (OLE2 オートメーションインタフェース機能)

形式

FUNCTION
LENGTH-OF-VARIANT (引数1)

機能

LENGTH-OF-VARIANT (VARIANT 値の長さ取得) 関数は、引数 1 で指定した VARIANT 値を COBOL の英数字型に変換するときに、必要となる受け取り領域の文字列長を取得する。この関数の型は整数型である。

一般規則

1. 引数について

- ・引数 1 は、バリエーションデータ項目を指定しなければならない。
- ・引数 1 に指定したバリエーションデータ項目が NULL の場合は、実行時にエラーとなる。
- ・引数 1 に指定したバリエーションデータ項目が文字列に変換できない場合は、実行時にエラーとなる。

2. 関数値について

- ・関数値は、VARIANT 値を COBOL の英数字型に変換するときに、COBOL の英数字項目として受け取りに必要な文字列長を取得する。

3. 関数の表現形式

- ・4バイトの2進項目 (PIC S9(9) USAGE COMP) (Windows(x64) COBOL2002 の場合, 8バイトの2進項目 (PIC S9(18) USAGE COMP))
- ・この関数は、整数型関数が指定できるところに記述できる。

18.5.9 組み込み関数の表現形式 (OLE2 オートメーションインタフェース機能)

それぞれの関数について表現形式などの詳細を次に示す。

表 18-4 組み込み関数の表現形式

項番	関数名	表現形式
1	CONVERT-TO-VARIANT	VARIANT 値へのポインタ
2	TYPE-OF-PROPERTY	符号付き整数部 5 けたの 2 進形式 (4 バイト)
3	TYPE-OF-VARIANT	符号付き整数部 5 けたの 2 進形式 (4 バイト)
4	VARIANT-TO-INTEGER	符号付き整数部 9 けたの 2 進形式 (4 バイト)
5	VARIANT-TO-NUMERIC	内部浮動小数点形式 (8 バイト)
6	VARIANT-TO-ALPHANUMERIC	引数に依存した長さの英数字
7	VARIANT-TO-OBJECT	OLE オブジェクトへのポインタ
8	LENGTH-OF-VARIANT	符号付き整数部 9 けたの 2 進形式 (4 バイト) (Windows(x64) COBOL2002 の場合, 符号付き整数部 18 けたの 2 進形式 (8 バイト))

19

データベース操作シミュレーション機能

この章では、構造型データベース (XDM/SD) 操作シミュレーション機能と、VOS3 SQL データベースアクセス機能プログラムをシミュレーションするリレーショナルデータベース (XDM/RD) 操作シミュレーション機能の形式について説明する。

19.1 構造型データベース (XDM/SD) 操作シミュレーション機能

構造型データベース (XDM/SD) 操作シミュレーション機能は、メインフレームで構造型データベース (XDM/SD) を操作する機能をシミュレーションする機能である。

詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/SD)」を参照のこと。

19.1.1 データ部 (構造型データベース (XDM/SD) 操作シミュレーション機能)

形式

DATA DIVISION.
[SUBSCHEMA SECTION.
サブスキーマ記述項]
[FILE SECTION.
[ファイル記述項]…]
[WORKING-STORAGE SECTION.
[データ記述項]…]
[LOCAL-STORAGE SECTION.
[データ記述項]…]
[LINKAGE SECTION.
[データ記述項]…]
[COMMUNICATION SECTION.
[通信記述項]…]
[REPORT SECTION.
[報告書記述項]…]
[SCREEN SECTION.
[画面記述項]…]

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

(1) サブスキーマ記述項 (SUBSCHEMA SECTION)

形式

SUBSCHEMA SECTION.
DB サブスキーマ名 WITHIN スキーマ名 [SUPPRESS]
[INDEX インデクス名1 [インデクス名2]…]
STATUS IS データ名1
[DETAIL CODE IS データ名2]
[RECORD NAME IS データ名3]
[RECORD LENGTH IS データ名4]
[CURSOR NUMBER IS データ名5]
[DYNAMIC KEY IS データ名6].

機能

サブスキーマ記述項は、データベースで操作するデータ構造とその関係を指定する。

構文規則

1. サブスキーマ名およびスキーマ名には、サブスキーマ名句に指定した名称を指定しなければならない。
2. 複数のスキーマと関連するサブスキーマを指定する場合、スキーマ名にはスキーマ名句に指定した代表スキーマ名を指定しなければならない。
3. 指定するサブスキーマに対する XDM 定義ユティリティの整合性チェック機能が実行済みでなければならない。ディクショナリに現、旧二つのサブスキーマを登録してある場合は、どちらかの整合性チェックが済んでいけばよい。ただし、現ディクショナリメンバの整合性チェックが済んでいないときは、旧ディクショナリメンバのサブスキーマが参照される。
また、現ディクショナリメンバから準スキーマを削除した場合は、整合性チェックが済んでいないとみなされる。
4. データ名 5 で指定するカーソルグループ番号は、SUBSCHEMA 句で定義したカーソル画数の範囲でなければならない。ゼロを指定すると、1 を仮定する。
5. 一つの UAP (User Application Program) に対して複数のサブスキーマを定義できる。
6. 翻訳単位の目的プログラム一つに対して、サブスキーマ名は一つである。ただし、入れ子のあるプログラムには記述できない。最外プログラムにだけ指定できる。
7. リンカで結合された UAP 間で同じサブスキーマ名を指定してもよい。
8. データ名 1~6 には、添字付けまたは指標付けを指定できない。
9. データ名 1~6 は、作業場所節、局所場所節または連絡節で定義する。
10. データ名 1 は、長さ 5 文字の固定長集団項目、英数字項目または符号なし外部 10 進項目でなければならない (けた数チェックはしない)。
11. データ名 2 は、長さ 4 文字の固定長集団項目、英数字項目または符号なし外部 10 進項目でなければならない (けた数チェックはしない)。
12. データ名 3 は、長さ 30 文字の固定長集団項目または英数字項目でなければならない (けた数チェックはしない)。
13. データ名 4 は、4 けたの外部 10 進項目、内部 10 進項目または 2 進項目の整数でなければならない (けた数チェックはしない)。
14. データ名 5 は、2 けたの外部 10 進項目、内部 10 進項目または 2 進項目の整数でなければならない (けた数チェックはしない)。
15. データ名 6 は、長さ 20 文字の英数字項目でなければならない。
16. サブスキーマ記述項中に COPY 文を書いてはならない。
17. COPY 文を使ってサブスキーマ節を複写してはならない。
18. レコードビュー名および構成要素ビュー名は、データ操作文以外に COBOL の文で一意名やデータ名として参照できる。
19. 構成要素ビュー名を可変長項目の制御変数として定義できない。

19.1.2 手続き部（構造型データベース（XDM/SD）操作シミュレーション機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

(1) CONNECT 文

形式

CONNECT CURRENT
レコードビュー名 TO {親子集合ビュー名1}….

機能

CONNECT 文は、レコードを一つまたはそれ以上の親子集合の子レコードとして組み入れる。

構文規則

1. 親子集合ビュー名は、二重指定してはならない。
2. 親子集合ビュー名を 2,001 個以上指定してはならない。

(2) DISCONNECT 文

形式

DISCONNECT CURRENT
レコードビュー名
FROM {レコード集合ビュー名1}….

機能

DISCONNECT 文は、一つまたはそれ以上の親子集合の子レコードになっているレコードをその親子集合から切り離す。

構文規則

1. 親子集合ビュー名は、二重指定してはならない。
2. 親子集合ビュー名を 2,001 個以上指定してはならない。

(3) ERASE 文

形式

ERASE [ALL] CURRENT レコードビュー名
[RETAINING RECORD CURRENCY].

機能

ERASE 文は、一つまたはそれ以上のレコードをデータベースから削除する。

(4) FETCH 文

形式

書き方 1

FETCH [FOR UPDATE]
{ FIRST
LAST
NEXT
PRIOR
[RELATIVE] { 整数1 }
DYNAMIC }
レコードビュー名
[INDEXED BY インデクス名1]
[INTO データ名2]
[WHERE 探索条件]
[RETAINING [RECORD]
[親子集合ビュー名2] … CURRENCY]
[USING データ名3].

書き方 2

FETCH [FOR UPDATE]
{ FIRST
LAST
NEXT
PRIOR
[RELATIVE] { 整数1 }
DYNAMIC }
[レコードビュー名]
WITHIN 親子集合ビュー名1
[INTO データ名2]
[WHERE 探索条件]
[RETAINING [RECORD]
[親子集合ビュー名2] … CURRENCY]
[USING データ名3].

書き方 3

FETCH [FOR UPDATE] DATA AREA IS データ名1
{ FIRST
LAST
NEXT
PRIOR }
[レコードビュー名] WITH パス名
[ADVANCING]
[INDEXED BY インデクス名1]
[WHERE 探索条件].

書き方 4

FETCH [FOR UPDATE] DATE AREA IS データ名1

{ FIRST }
{ LAST }

[レコードビュー名]

WITHIN 親子集合ビュー名1

[WHERE 探索条件].

機能

FETCH 文は、データベース中にあるレコードを探索し、そのレコードを利用できるようにする。

構文規則

- 一意名 1 は、4 バイトの 2 進項目でなければならない。
- 整数 1 は、0 を除く -32,767 ~ 32,767 の範囲の値しか指定できない。ただし、書き方 1 は、1 ~ 32,767 の範囲の値または -1 しか指定できない。
- 書き方 2 で、マルチメンバセットで検索条件文を指定するときはレコードビュー名は省略できない。
- 書き方 2 で、レコードビュー名を省略したときは INTO 指定を省略できない。
- データ名 2 とデータ名 3 は、強く型付けされていない固定長集団項目、英字項目または英数字項目でなければならない。
- データ名 2 の長さは、レコードビュー長（マルチメンバセットのときは最大レコードビュー長）以上のサイズでなければならない。
- データ名 1 ~ 3 は、作業場所節、局所場所節または連絡節で定義できる。
- データ名 1 の形式を次に示す。

図 19-1 データ名 1 の形式



- データ名 1 は、強く型付けされていない固定長集団項目または英数字項目でなければならない。
- 書き方 1 で DYNAMIC を指定したとき WHERE 条件を記述しなくてもよい。通常の WHERE 条件を記述すると、そのまま処理される。WHERE 条件を省略すると、WHERE KEY = データ名 (DYNAMIC KEY に指定された) を仮定する。
- 書き方 1、書き方 3 で LAST を指定するときは、INDEXED を指定しなければならない。
- 親子集合ビュー名 2 は、二重に指定してはならない。
- 親子集合ビュー名 2 は、2,001 個以上指定してはならない。
- インデクス名を指定していない場合、条件中に "KEY" を指定してはならない。
- 書き方 3 で NEXT/PRIOR を指定するときは、ADVANCING を指定してはならない。
- データ名 1 は添字付けおよび修飾をしてもよい。

17. データ名 2, 3 は修飾してもよい。

18. 書き方 1 で DYNAMIC を指定する場合は、探索条件には KEY = データ名しか指定できない。

(5) FIND 文

形式

書き方 1

```
FIND [FOR UPDATE]
{
  FIRST
  LAST
  NEXT
  PRIOR
  [RELATIVE] { 整数1
               一意名1 }
  DYNAMIC
}
レコードビュー名
  [INDEXED BY インデクス名1]
  [WHERE 検索条件]
[RETAINING [RECORD]
  [親子集合ビュー名2]…CURRENCY]
[USING データ名1].
```

書き方 2

```
FIND [FOR UPDATE]
{
  FIRST
  LAST
  NEXT
  PRIOR
  [RELATIVE] { 整数1
               一意名1 }
}
[レコードビュー名] WITHIN 親子集合ビュー名1
  [WHERE 検索条件]
[RETAINING [RECORD]
  [親子集合ビュー名2]…CURRENCY]
[USING データ名1].
```

書き方 3

```
FIND [FOR UPDATE]
CURRENT [ [レコードビュー名
           { {OWNER}
             OF 親子集合ビュー名1
           {MEMBER}
           } ] ]
[RETAINING [RECORD]
  [親子集合ビュー名2]… CURRENCY]
[USING データ名1].
```

機能

FIND 文は、データベース中のあるレコードを検索する。

構文規則

1. 書き方 2 で、あるマルチメンバセットで条件を指定するときは、レコードビュー名は省略できない。

2. 書き方 1, 2 は, INTO 指定を除いて FETCH 文 (書き方 1, 2) と同じである。
3. データ名 1 は固定長集団項目, 英字項目または英数字項目のどれかでなければならない。
4. データ名 1 は作業場所節, 局所場所節または連絡節で定義する。
5. 書き方 1 で DYNAMIC を指定したとき WHERE 条件を記述しなくてもよい。通常の WHERE 条件を記述すると, そのまま処理される。WHERE 条件を省略すると, WHERE KEY = データ名 (DYNAMIC KEY に指定された) を仮定する。
6. 書き方 1 で LAST を指定するときは, INDEXED 指定が必要である。
7. 書き方 3 の RETAINING 指定については, 「(4) FETCH 文」を参照のこと。
8. データ名 1 は修飾してもよい。

(6) GET 文

形式

書き方 1

GET レコードビュー名1 [INTO データ名2].

書き方 2

GET DATA AREA IS データ名1
 NEXT [レコードビュー名2]
 [WITHIN 親子集合ビュー名1]
 [INTO データ名2].

機能

GET 文は, FIND 文で位置づけられたレコードを利用できるようにする。

構文規則

1. データ名 2 は, 英字項目, 英数字項目または強く型付けされていない固定長集団項目でなければならない。
2. データ名 2 の長さは, レコードビュー長以上のサイズでなければならない。
3. データ名 2 は, 作業場所節, 局所場所節または連絡節で定義する。
4. DATA AREA を指定した場合, レコードビュー名 2, INTO 指定のどちらかを指定しなければならない。
5. データ名 1 の構文規則については, 「(4) FETCH 文」を参照のこと。
6. データ名 1 は添字付けおよび修飾をしてもよい。
7. データ名 2 は修飾してもよい。

(7) MODIFY 文

形式

MODIFY レコードビュー名 [FROM データ名1].

機能

MODIFY 文は、データベース中のレコードを更新する。

構文規則

1. データ名 1 は、固定長集団項目、英字項目または英数字項目でなければならない。
2. データ名 1 の長さは、レコードビュー長以上のサイズでなければならない。
3. データ名 1 は、作業場所節、局所場所節または連絡節で定義する。
4. データ名 1 は修飾してもよい。

(8) NULLIFY 文

形式

```
NULLIFY CURRENT  
[ { { レコードビュー名  
  { { OWNER }  
  { { MEMBER } } OF 親子集合ビュー名1 } ] .
```

機能

NULLIFY 文は、位置指示文に null (ナル) 値を設定する。

構文規則

1. OWNER OF 親子集合ビュー名 1 は、特異親子集合に対しては指定できない。

(9) RECONNECT 文

形式

```
RECONNECT CURRENT レコードビュー名  
WITHIN {親子集合ビュー名1} … .
```

機能

RECONNECT 文は、一つまたはそれ以上の親子集合の子レコードになっているレコードを、その親子集合内または親子集合の子レコードとして再接続する。

構文規則

1. 親子集合ビュー名は、二重指定してはならない。
2. 親子集合ビュー名を 2,001 個以上指定してはならない。

(10) STORE 文

形式

```
STORE レコードビュー名 [FROM データ名1]  
[RETAINING [RECORD]  
[親子集合ビュー名1] … CURRENCY].
```

機能

STORE 文は、データベースにレコードを格納する。

構文規則

1. データ名 1 は、英字項目、英数字項目または固定長集団項目でなければならない。
2. データ名 1 の長さは、レコードビュー長以上のサイズでなければならない。
3. データ名 1 は、作業場所節、局所場所節または連絡節で定義する。
4. 親子集合ビュー名は、二重指定してはならない。
5. 親子集合ビュー名は、2,001 個以上指定してはならない。
6. データ名 1 は修飾してもよい。

(11) 検索条件文

形式

$$\left\{ \begin{array}{l} \text{[CONTENT OF] 構成要素ビュー名1} \\ \text{一意名1} \\ \text{定数1} \\ \text{KEY} \end{array} \right\} \text{ IS}$$
$$\left\{ \begin{array}{l} \text{GREATER THAN} \\ > \\ \text{LESS THAN} \\ < \\ \text{EQUAL TO} \\ = \\ \text{GREATER THAN OR EQUAL TO} \\ >= \\ \text{LESS THAN OR EQUAL TO} \\ <= \end{array} \right\}$$
$$\left\{ \begin{array}{l} \text{[CONTENT OF] 構成要素ビュー名2} \\ \text{一意名2} \\ \text{定数2} \end{array} \right\}$$

機能

検索条件文は、FIND 文または FETCH 文中の WHERE 指定で使用され、レコードを検索する条件を指定する。

構文規則

1. 一意名は、サブスキーマ節で定義してはならない。
2. 定数に英数字定数（表意定数を含まない）、整数定数、実数定数および日本語文字定数以外は指定できない。
3. 構成要素ビュー名は、基本配列、配列に含まれない固定長集団項目、英数字項目、日本語項目、外部 10 進項目、内部 10 進項目および 2 進項目を指定する（配列に含まれる固定長集団項目はエラーチェックしない）。
4. 複数の条件を指定する場合、論理演算子の AND、OR で検索条件をつなぐ。

5. 検索条件にレコードビュー名は指定できない。

一般規則

1. 検索条件文では、少なくとも比較演算子の左辺または右辺のどちらかに構成要素ビュー名を指定する。
2. 条件の選択条件は 256 個以上指定できない。
3. 条件で一つの選択条件の中の関係条件数は 256 個以上指定できない。
4. KEY を指定した条件には、構成要素ビュー名は指定できない。
5. 条件中に KEY を指定している場合は、複数の関係条件を OR で結合できない。
6. 条件中に KEY を指定した関係条件と一般の関係条件を同時に記述できない。
7. CONTENT が指定された構成要素ビュー名は、構成要素ビュー名としてではなく、COBOL の一意名と同様に扱われる。
8. KEY を指定した関係条件は 3 個以上記述できない。
9. 検索条件文に指定できる項目の組み合わせと形式の組み合わせを「表 19-1 検索条件文に指定できる項目の組み合わせ」と「表 19-2 検索条件文に指定できる形式の組み合わせ」に示す。

表 19-1 検索条件文に指定できる項目の組み合わせ

左辺		右辺														
		構成要素ビュー名						一意名						定数		
		FG	AN	N	ED	ID	BI	FG	AN	N	ED	ID	BI	LA	LN	LJ
構成要素ビュー名	FG	○a	○a	×	×	×	×	○a	○a	×	×	×	×	○a	×	×
	AN	○a	○a	×	×	×	×	○a	○a	×	×	×	×	○a	×	×
	N	×	×	○a	×	×	×	×	×	○a	×	×	×	×	×	○a
	ED	×	×	×	○n	○n	×	×	×	×	○n	×	×	×	○n	×
	ID	×	×	×	○n	○n	×	×	×	×	×	○n	×	×	○n	×
	BI	×	×	×	×	×	×	×	×	×	×	×	○n	×	×	×
一意名	FG	○a	○a	×	×	×	×	×	×	×	×	×	×	×	×	×
	AN	○a	○a	×	×	×	×	×	×	×	×	×	×	×	×	×
	N	×	×	○a	×	×	×	×	×	×	×	×	×	×	×	×
	ED	×	×	×	○n	×	×	×	×	×	×	×	×	×	×	×
	ID	×	×	×	×	○n	×	×	×	×	×	×	×	×	×	×
	BI	×	×	×	×	×	○n	×	×	×	×	×	×	×	×	×
定数	LA	○a	○a	×	×	×	×	×	×	×	×	×	×	×	×	×
	LN	×	×	×	○n	○n	×	×	×	×	×	×	×	×	×	×
	LJ	×	×	○a	×	×	×	×	×	×	×	×	×	×	×	×

(凡例)

○a：文字比較する

○n：数字比較する

×：許されない

AN：英数字

N：日本語

ED：外部 10 進

ID：内部 10 進

BI：2 進

FG：固定長集団

LA：英数字定数

LN：数字定数

LJ：日本語文字定数

表 19-2 検索条件文に指定できる形式の組み合わせ

左辺		右辺			
		定数	構成要素ビュー名		
			基本項目	基本配列の要素	配列に含まれない固定長集団項目
定数		×	○	○	○
構成要素 ビュー名	基本項目	○	○	○	○
	基本配列の要素	○	○	○	○
	配列に含まれない固定長集団項目	○	○	○	○

(凡例)

○：組み合わせ可能

×：組み合わせ不可能

(12) データベース条件文

形式

書き方 1

```

CURRENT
[ ( ( (レコードビュー名1
  { {OWNER} } OF 親子集合ビュー名1
  {MEMBER} ) ) ) ]

```

IS [NOT] ALSO

```

CURRENT
[ ( ( (レコードビュー名2
  { {OWNER} } OF 親子集合ビュー名2
  {MEMBER} ) ) ) ]

```

機能

二つの位置指示子が示すレコードが同一かどうかを判定する。

書き方 2

```
CURRENT  
[[ [レコードビュー名  
  { { OWNER } } OF 親子集合ビュー名1  
  { { MEMBER } } ] ]  
IS [NOT] NULL
```

機能

位置指示子がナル値かどうかを判定する。

書き方 3

```
親子集合ビュー名1 IS [NOT] EMPTY
```

機能

親子集合が空かどうかを判定する。

書き方 4

```
親子集合ビュー名1 CONTAINS
```

```
CURRENT  
[[ [レコードビュー名  
  { { OWNER } } OF 親子集合ビュー名2  
  { { MEMBER } } ] ]
```

機能

レコードが親子集合型の子レコードかどうかを判定する。

19.2 リレーショナルデータベース (XDM/RD) 操作シミュレーション機能

リレーショナルデータベース (XDM/RD) 操作シミュレーション機能は、メインフレームで SQL データベースアクセス機能のリレーショナルデータベース (XDM/RD) を操作する機能をシミュレーションする機能である。

ここでは、リレーショナルデータベース (XDM/RD) を操作する機能の一般規則についてだけ述べる。構文規則、一般規則などの詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系プログラム作成の手引 (XDM/RD E2)」および「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照のこと。

なお、このシステムではリレーショナルデータベースのアクセスはしないで、SQL データ操作言語のコンパイルだけをする。

19.2.1 全般規則 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

(1) 区切り文字

区切り文字を次に示す。

空白	,	.	()	'	"	:
+	*	/	<	>	<=	>=	=
	^	<>					

- 拡張コードは使用できない (このシステムではチェックしない)。
- このシステムでは、「;」「==」「**」も区切り文字として扱う。
- このシステムでは区切り文字をすべてコメントとして扱う。

(2) 予約語

- EXEC, SQL, END-EXEC 以外の SQL の予約語は COBOL の予約語として扱わない。
- EXEC, SQL, END-EXEC は COBOL の予約語としても扱う。
- 拡張コードは使用できない (このシステムではチェックしない)。
- 大文字と小文字は等価とする。
- 大文字と小文字は混在してもよい。

(3) 埋め込み変数

形式

:データ名1 [.データ名2]…

1. データ名 1, 2 は COBOL2002 のデータ名の名称規則に従う。
2. 埋め込み変数中の「:」の後, 「.」の前後に空白は書けない。
3. 一意でないデータ名を明示的に参照するために, 一連の修飾語によって一意にしなければならない。
4. 修飾する必要のない名前を修飾してもよい。一意にする修飾語の組み合わせが幾つもあるとき, どれを用いてもよい。
5. データ名 1 は, レベル指示語に関係づけられる名前で, さらにデータ名 2 が従属する集団項目でなければならない。また, データ名 2 は順に階層系列のより下位のレベルになるように指定する。
6. このシステムでは埋め込み変数のチェックはしない。

19.2.2 見出し部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

見出し部に EXEC SQL~END-EXEC は書けない。ただし, このシステムではコメント扱いとする。

19.2.3 環境部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

環境部に EXEC SQL~END-EXEC は書けない。ただし, このシステムではコメント扱いとする。

19.2.4 データ部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

形式

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC  
データ記述項 …  
EXEC SQL END DECLARE SECTION END-EXEC
```

ただし, このシステムでは EXEC SQL~END-EXEC までをコメント扱いとする。

(1) SQL 連絡領域

形式

COPY SQLCA

または

EXEC SQL INCLUDE SQLCA

END-EXEC

1. SQL 文のリターンコードと関連情報を SQL 連絡領域に設定する。ただし、このシステムでは SQL 連絡領域に何も設定しない。
2. SQL 連絡領域を使用するとき、ほかの場所で SQLCODE を定義してはならない。SQL 連絡領域に SQLCODE が含まれている。
3. SQLCA は、翻訳単位に 1 個だけ指定する。
4. SQLCA のレベル番号は 1 でなければならない。
5. SQLCA は、GLOBAL 句の有無に関係なく、大域属性を持つ。
6. SQLCA は、最外プログラムまたは利用者定義関数で宣言しなければならない。
7. INCLUDE の後は SQLCA でなければならない。
8. このシステムでは SQL 連絡領域のエラーチェックはしない。

(2) SQLCODE 変数

1. SQL 文のリターンコードを設定する。ただし、このシステムでは SQLCODE 変数に何も設定しない。
2. SQLCODE 変数は、埋め込み SQL 宣言節の中で宣言しなければならない。
3. SQLCODE 変数のレベル番号は、1 か 77 でなければならない。
4. SQLCODE 変数の属性は、S9(9) COMP でなければならない。
5. SQLCODE 変数は、翻訳単位に 1 個だけ指定する。
6. SQLCODE 変数は、最外プログラムまたは利用者定義関数で宣言しなければならない。
7. SQLCODE 変数は、GLOBAL 句の有無に関係なく、大域属性を持つ。
8. このシステムでは SQLCODE 変数のエラーチェックはしない。

19.2.5 原始文操作機能 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

- EXEC SQL~END-EXEC 間の COPY 文および INCLUDE 文の展開と原文中の文字列、分離符、正書法のチェックは COBOL の規則に従う。
- 登録集原文中に SQL 文が存在してもよい。

19.2.6 手続き部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

形式

EXEC SQL SQL文 END-EXEC

1. END-EXEC が現れないでプログラムが終了したときは、プログラム終了の直前に END-EXEC を仮定する。
2. END-EXEC が現れないで次の EXEC SQL が現れても、EXEC SQL はコメントとして扱う。

構文規則

1. このシステムでは SQL 文の構文に誤りがあってもエラーにしない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

19.2.7 特殊レジスタ (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

データベースアクセス機能を使用するとき、次の特殊レジスタを生成する。

- RDB-STATUS
- RDB-COUNT
- RDB-RDNODE

1. RDB-STATUS は SQL 文の実行後のステータスコードを X(5)の属性で返す。初期値は 5 バイトの空白である。
2. RDB-COUNT は SQL 文の実行によって操作された行数を S9(9) COMP の属性で返す。初期値はゼロである。
3. RDB-RDNODE は分散アクセス時の RD ノード名称を X(30)の属性で返す。初期値は 30 バイトの空白である。
4. RDB-STATUS, RDB-COUNT, RDB-RDNODE はそれぞれプログラム、利用者定義関数、ファクトリ、インスタンス単位の一つずつ生成され、これらは大域属性を持つ静的データである。
5. このシステムでは値の設定はしない。

19.2.8 注意事項 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)

(1) EXEC SQL~END-EXEC 間での COBOL の語句の扱い

- EXEC SQL~END-EXEC 間に COPY 文および INCLUDE 文があるときは、最初に COPY 文、INCLUDE 文の展開をしてから SQL の処理をする。
- EXEC SQL~END-EXEC 間に COPY 文、INCLUDE 文に関係しない COBOL の語句があるときは、コメントとして扱う。

(2) -SQL,XDM オプションがないときの EXEC SQL~END-EXEC の扱い

-SQL,XDM オプションがないとき、EXEC SQL~END-EXEC は書いてはならない。

20

HiRDB による索引ファイル入出力機能

HiRDB による索引ファイル入出力機能は、COBOL2002 の索引ファイル入出力機能を使って RDB ファイルにアクセスする機能である。

この章では、RDB ファイル入出力をするための言語の形式や規則について説明する。

-UniObjGen オプションの指定が前提である Linux COBOL2002 では、この機能に関する次に示すコンパイラオプションを指定した場合は回復不能エラー、構文を記述した場合は重大エラーを出力する。

- -RDBTran オプション
- ORGANIZATION IS RDB
- COMMIT 文
- ROLLBACK 文

20.1 概説 (HiRDB による索引ファイル入出力機能)

COBOL2002 の索引ファイルの入出力文を使って RDB (Relational Database) にアクセスできる。ただし、この機能は DC 機能 (シミュレーションを含む) と同時に使用できない。

ここでは、マニュアル「COBOL2002 言語 標準仕様編」の索引ファイルと異なる部分についてだけ説明する。

なお、この機能を使うときにファイル共用機能の構文を書いてもよいが、覚え書きとする。

20.2 環境部 (HiRDB による索引ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

20.2.1 ファイル管理段落 (HiRDB による索引ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.3 ファイル管理段落 (FILE-CONTROL)」

(1) ALTERNATE RECORD KEY 句 (HiRDB による索引ファイル入出力機能)

索引ファイルと異なる規則だけを説明する。

- DUPLICATES 指定は覚え書きとする。重複を許すかどうかは、ファイル生成時に指定する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(5) ALTERNATE RECORD KEY 句」

(2) ORGANIZATION 句 (HiRDB による索引ファイル入出力機能)

形式

ORGANIZATION IS RDB

機能

ファイルの論理的構成が RDB 編成であることを指定する。

構文規則

1. データコミュニケーション機能の CD 記述項と同時に使用できない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(9) ORGANIZATION 句」

(3) RECORD KEY 句 (HiRDB による索引ファイル入出力機能)

索引ファイルの規則と異なる規則だけ説明する。

- 主レコードキーの値は、幾つかのレコードで重複してもよい。主レコードキーの重複を許すかどうかはファイル生成時に指定する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(14) RECORD KEY 句」

20.3 データ部 (HiRDB による索引ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

20.3.1 ファイル記述項 (FD) (HiRDB による索引ファイル入出力機能)

形式

```
FD ファイル名1
  [IS EXTERNAL]
  [IS GLOBAL]
  [BLOCK CONTAINS [整数1 TO] 整数2
    { RECORDS
      }
    { CHARACTERS
      } ]
  [RECORD { CONTAINS 整数3 CHARACTERS
    IS VARYING IN SIZE
    [[FROM 整数4][TO 整数5]
    CHARACTERS]
    [DEPENDING ON データ名1]
    CONTAINS 整数6 TO 整数7
    CHARACTERS
  } ]
  [LABEL { RECORD IS } { STANDARD }
    { RECORDS ARE } { OMITTED } ]
  [VALUE OF {処理系作成者語1 IS { データ名2
    }
    定数1
  } ...]
  [DATA { RECORD IS
    }
    { RECORDS ARE } { データ名3} ...]
  [DATA FORMAT IS { STANDARD
    }
    { REVERSE
    }
    { NATIVE
    } ].
```

構文規則

1. RDB ファイルのレコード記述項には、DEPENDING ON 指定の OCCURS 句を指定できない。
2. Windows COBOL2002 では、RDB ファイルのファイル記述項の DATA FORMAT 句は覚え書きとする。UNIX COBOL2002 では、DATA FORMAT 句は使用できない。
3. RDB ファイルのファイル記述項に従属するレコード記述は一つでなければならない。

一般規則

1. RECORD 句を書いた場合、レコード記述から計算した最大の長さの固定長で動作する。
2. データ名 1 の指定に関係なく、レコード記述から計算した最大長で出力動作が行われる。
3. データ名 1 の指定に関係なく、レコード記述から計算した最大長で入力動作が行われる。データ名 1 が指定された場合は、常に最大長が返される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.3.4 ファイル記述項 (file description entry)」

20.4 手続き部 (HiRDB による索引ファイル入出力機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

20.4.1 CANCEL 文 (HiRDB による索引ファイル入出力機能)

HiRDB による索引ファイル入出力機能に対してトランザクション管理機能を使用している場合、次の規則が適用される。

- 取り消されるプログラムに HiRDB による索引ファイル入出力機能が定義されていて、かつそのファイルが開かれているとき、そのファイルは閉じられる。このとき、暗黙的な COMMIT 文が実行される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.5 CANCEL 文」

20.4.2 CLOSE 文 (HiRDB による索引ファイル入出力機能)

HiRDB による索引ファイル入出力機能に対してトランザクション管理機能を使用している場合、次の規則が適用される。これ以外の規則は、索引ファイルの規則と同じである。

- 暗黙的な COMMIT 文が実行される。

トランザクション管理機能については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の HiRDB による索引編成ファイルのトランザクション管理機能についての説明を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.6 CLOSE 文」

20.4.3 COMMIT 文 (HiRDB による索引ファイル入出力機能)

形式

COMMIT

機能

COMMIT 文は、コミットで現行のトランザクションを終了させる。また、ROLLBACK 文の実行やプログラムの異常終了によってファイルなどを復元する場合の復元時点となる。

一般規則

1. HiRDB による索引ファイル入出力機能での対象だけに意味を持つ。
2. 現行のトランザクションを終了させる。
3. 現行のトランザクションによってなされたすべての変更をコミットする。
4. COMMIT 文を実行すると、現行のトランザクションで占有したレコードの排他制御は解除される。
5. COMMIT 文を実行すると、HiRDB による索引ファイル入出力機能では、すべてのファイルのファイル位置指示子は不定となる。なお順アクセスで、継続するレコードを読み込む場合は、COMMIT 文の実行直前のキー値を使って、START 文で位置決めをし直さなければならない。

トランザクション管理機能については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の HiRDB による索引編成ファイルのトランザクション管理機能についての説明を参照のこと。

20.4.4 DELETE 文 (HiRDB による索引ファイル入出力機能)

DELETE 文については、索引ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.10 DELETE 文」

20.4.5 EXIT PROGRAM 文, GOBACK 文 (HiRDB による索引ファイル入出力機能)

HiRDB による索引ファイル入出力機能に対してトランザクション管理機能を使用している場合、次の規則が適用される。

- HiRDB による索引ファイル入出力機能が定義されている初期化属性のプログラムで、かつそのファイルが開かれているとき、そのファイルは閉じられる。このとき、暗黙的な COMMIT 文が実行される。
- COBOL 主プログラムで GOBACK 文を実行すると、STOP RUN 文と同じ動作になる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.20 EXIT 文」

「COBOL2002 言語 標準仕様編」 「10.8.22 GOBACK 文」

20.4.6 OPEN 文 (HiRDB による索引ファイル入出力機能)

索引ファイルの規則と異なる規則だけを説明する。

- OUTPUT モード指定のとき、ファイルの追加書きはしないで、常にファイルを再作成する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.30 OPEN 文」

20.4.7 READ 文 (HiRDB による索引ファイル入出力機能)

索引ファイルの規則と異なる規則だけを説明する。

- PREVIOUS 指定のある READ 文は記述できない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」

20.4.8 REWRITE 文 (HiRDB による索引ファイル入出力機能)

REWRITE 文については、索引ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」

20.4.9 ROLLBACK 文 (HiRDB による索引ファイル入出力機能)

形式

ROLLBACK

機能

ROLLBACK 文は、ロールバックで現行のトランザクションを終了させる。また、直前にコミットした時点の状態まで、ファイルを復元する。

一般規則

1. HiRDB による索引ファイル入出力機能での対象だけに意味を持つ。
2. 現行のトランザクションを終了させる。
3. 現行のトランザクションによってなされたすべての変更を取り消す。
4. ROLLBACK 文を実行すると、現行のトランザクションで占有したレコードの排他制御は解除される。
5. ROLLBACK 文を実行すると、HiRDB による索引ファイル入出力機能でのすべてのファイルのファイル位置指示子は不定となる。なお順アクセスで、継続するレコードを読み込む場合は、ROLLBACK 文の実行直前のキー値を使って、START 文で位置決めをし直さなければならない。

トランザクション管理機能については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のHiRDBによる索引編成ファイルのトランザクション管理機能についての説明を参照のこと。

20.4.10 START 文 (HiRDB による索引ファイル入出力機能)

索引ファイルの規則と異なる規則だけを説明する。

- KEY 指定に記述するデータ名は、主レコードキーまたは副レコードキーと左端の文字位置を共有し、そのレコードキーと同じ長さでなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.45 START 文」

20.4.11 STOP RUN 文 (HiRDB による索引ファイル入出力機能)

HiRDB による索引ファイル入出力機能に対してトランザクション管理機能を使用している場合、次の規則が適用される。

- 開かれている HiRDB による索引ファイル入出力機能が閉じられる。このとき、暗黙的な COMMIT 文が実行される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.46 STOP 文」

20.4.12 UNLOCK 文 (HiRDB による索引ファイル入出力機能)

索引ファイルの規則と異なる規則だけを説明する。

- UNLOCK 文の実行時に、入出力順序の妥当性はチェックするが、実際にレコード排他の解除はしない。レコード排他は RDB システムの定義に従う。

20.4.13 WRITE 文 (HiRDB による索引ファイル入出力機能)

WRITE 文については、索引ファイルの規則と同じである。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

21

数字項目のけた拡張機能

数字項目のけた拡張機能とは、数字項目、数字編集項目および数字定数で扱える数字のけた数の上限を 18 けたから 38 けたに拡張し、算術演算、転記、および比較条件で使用できるようにする機能である。数字項目のけた拡張機能を使用する場合、コンパイル時に `-MaxDigits38` オプションおよび `-IntResult,DecFloat40` オプションを指定する。

この章では、数字項目のけた拡張機能を使用するための言語の形式と規則について説明する。

この機能は、64bit 版 COBOL2002 で使用できる。

21.1 ソース単位

PICTURE 句の数字けた位置の個数に 19~38 を指定した数字項目, 19~38 を指定した数字編集項目および 19~38 けたの数字定数を定義できるソース単位を次に示す。

表 21-1 19~38 けたを指定した数字項目, 数字編集項目および数字定数を定義できるソース単位

ソース単位	数字けた位置の個数が 19~38 けたの数字項目, 数字編集項目および数字定数の定義
最外側のプログラム定義 (その内包するプログラム定義群も含む)	○
内側のプログラム定義 (その内包するプログラム定義群も含む)	○
関数定義	×
クラス定義	×
ファクトリ定義	×
インスタンス定義	×
メソッド定義	×
インタフェース定義	×

(凡例)

○ : 定義できる

× : 定義できない

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「6.3 ソース単位」

21.2 TURN 指令 (数字項目のけた拡張機能)

構文規則

例外名 1 に、EC-SIZE および EC-SIZE で始まる例外名を指定してはならない。なお、例外名 1 に EC-ALL を指定して EC-ALL 例外のチェックを有効にしても、EC-SIZE 例外のチェックは有効とならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「3.3.15 TURN 指令」

21.3 定数（数字項目のけた拡張機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.2.1(2) 定数」

21.3.1 数字定数

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.2.1(2)(b) 数字定数」

(1) 固定小数点数字定数

固定小数点数字定数とは、0 から 9 までの数字、正号、負号、および小数点から成る文字列とする。

このシステムでは、1 けた以上 38 けた以下の長さの固定小数点数字定数が許される。

21.4 データ項目 (数字項目のけた拡張機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.4.2 データ項目や定数の字類と項類」

21.4.1 数字項類 (数字項目のけた拡張機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.4.2(5) 数字項類 (Numeric category)」

(1) 外部 10 進形式

外部 10 進形式に割り当てられる記憶域のサイズの算出方法は、PICTURE 句の数字けた位置の個数が 19~38 の場合でも、けた位置の個数が 1~18 の場合と同様である。1 バイトで数字 1 けたを表現し、SIGN 句に SEPARATE を指定した場合は、演算符号を表す文字が数字列とは分離されて置かれる。外部 10 進形式に割り当てられる記憶域のサイズを次の表に示す。

表 21-2 外部 10 進形式に割り当てられる記憶域のサイズ

演算符号		記憶域のサイズ (単位: バイト数)
符号付き	SIGN 句 SEPARATE 指定あり	PICTURE 句の'9'の個数 + 1
	SIGN 句 SEPARATE 指定なし	PICTURE 句の'9'の個数
符号なし		PICTURE 句の'9'の個数

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.4.2(5)(a) 外部 10 進形式」

(2) 内部 10 進形式

内部 10 進形式に割り当てられる記憶域のサイズの算出方法は、PICTURE 句の数字けた位置の個数が 19~38 の場合でも同様である。内部 10 進形式に割り当てられる記憶域のサイズの算出方法を次に示す。

記憶域のサイズ = PICTURE 句の'9'の個数 / 2 + 1

この算出方法で求める値の単位は、バイト数である。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.4.2(5)(b) 内部 10 進形式」

21.4.2 数字編集項類 (数字項目のけた拡張機能)

数字編集項目の割り当てられる記憶域のサイズの算出方法は、PICTURE 句の数字けた位置の個数が 19～38 の場合でも標準仕様と同じである。算出式を次に示す。

記憶域のサイズ = PICTURE 句の文字位置を表す記号の個数

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.4.2(6) 数字編集項類 (Numeric-edited category)」

21.5 一意参照 (数字項目のけた拡張機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.3.1 一意参照」

21.5.1 添字

構文規則

1. データ名 2 が数字項目のときは、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 整数 1, 整数 2, 整数 3 に 19～38 けたの整数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.3.1(2) 添字 (Subscript)」

21.6 一意名のいろいろ (数字項目のけた拡張機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.3.2 一意名のいろいろ」

21.6.1 関数一意名

構文規則

1. 一部の組み込み関数を除き、引数 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目および数字編集項目を指定してはならない。組み込み関数の引数 1 の詳細は、「21.10.1 引数 (数字項目のけた拡張機能)」を参照のこと。
2. 引数 1 に、けた数が 19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.3.2(2) 関数一意名」

21.6.2 部分参照子

構文規則

1. 最左端位置および長さに指定した算術式に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 最左端位置および長さに指定した算術式に、19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.3.2(3) 部分参照子」

21.7 環境部 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. けた数が 19～38 けたの数字項目を指定してはならない。
2. 数字のけた数が 19～38 けたの数字編集項目を指定してはならない。
3. 19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

「COBOL2002 言語 標準仕様編」 「付録 D.5 環境部 (ENVIRONMENT DIVISION)」

21.8 データ部 (数字項目のけた拡張機能)

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

「COBOL2002 言語 標準仕様編」 「付録 D.6 データ部 (DATA DIVISION)」

21.8.1 ファイル節 (FILE SECTION) (数字項目のけた拡張機能)

構文規則

1. 次のファイルのレコード記述項に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を定義してもよい。

- ・順ファイル
- ・相対ファイル
- ・テキストファイル
- ・HiRDB による索引ファイル※ (Windows(x64) COBOL2002 と AIX(64) COBOL2002 が対象)

注※

ただし、COBOL で使用できる HiRDB の列のデータ型と COBOL のデータ記述の対応については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の HiRDB の定義と COBOL の定義の関連性を参照のこと。

2. 次のファイルのレコード記述項に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を定義してはならない。

- ・索引ファイル
- ・CSV 編成ファイル
- ・整列併合ファイル

3. VALUE 句以外で、けた数が 19～38 けたの数字定数を指定してはならない。VALUE 句の規則は、「21.8.17 VALUE 句 (数字項目のけた拡張機能)」を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.3 ファイル節 (FILE SECTION)」

21.8.2 作業場所節 (WORKING-STORAGE SECTION) (数字項目のけた拡張機能)

構文規則

PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を定義してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.4 作業場所節 (WORKING-STORAGE SECTION)」

21.8.3 局所場所節 (LOCAL-STORAGE SECTION) (数字項目のけた拡張機能)

構文規則

PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を定義してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.5 局所場所節 (LOCAL-STORAGE SECTION)」

21.8.4 連絡節 (LINKAGE SECTION) (数字項目のけた拡張機能)

構文規則

PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を定義してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.6 連絡節 (LINKAGE SECTION)」

21.8.5 報告書節 (REPORT SECTION) (数字項目のけた拡張機能)

構文規則

1. PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を定義してはならない。
2. PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を指定してはならない。
3. けた数が 19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「13.4 データ部 (DATA DIVISION) (報告書作成機能)」

21.8.6 サブスキーマ節 (SUBSCHEMA SECTION) (数字項目のけた拡張機能)

構文規則

1. PICTURE 句の数字けた位置の個数が 19～38 の数字項目，および数字編集項目を定義してはならない。
2. PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
3. PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してはならない。

なお，サブスキーマ節の詳細については，「[19.1.1 データ部 \(構造型データベース \(XDM/SD\) 操作シミュレーション機能\)](#)」を参照のこと。

21.8.7 画面節 (SCREEN SECTION) (数字項目のけた拡張機能)

この節は，Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

構文規則

1. PICTURE 句の数字けた位置の個数が 19～38 の数字項目，および数字編集項目を定義してはならない。
2. PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
3. PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してはならない。
4. けた数が 19～38 けたの数字定数を指定してはならない。

画面節 (SCREEN SECTION) による画面機能の詳細については，「[12. 画面節 \(SCREEN SECTION\) による画面機能](#)」を参照のこと。

21.8.8 画面節 (WINDOW SECTION) (数字項目のけた拡張機能)

この節は，Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

構文規則

1. PICTURE 句の数字けた位置の個数が 19～38 の数字項目，および数字編集項目を定義してはならない。
2. PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
3. PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してはならない。
4. けた数が 19～38 けたの数字定数を指定してはならない。

画面節 (WINDOW SECTION) による画面機能の詳細については、「13. 画面節 (WINDOW SECTION) による画面機能」を参照のこと。

21.8.9 BLANK WHEN ZERO 句 (数字項目のけた拡張機能)

構文規則

1. PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目に BLANK WHEN ZERO 句を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.7 BLANK WHEN ZERO 句」

21.8.10 BLOCK CONTAINS 句 (数字項目のけた拡張機能)

構文規則

整数 1 および整数 2 に、けた数が 19~38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.9 BLOCK CONTAINS 句」

21.8.11 DATA RECORDS 句 (数字項目のけた拡張機能)

構文規則

データ名 1 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目、および数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.18 DATA RECORDS 句」

21.8.12 EXTERNAL 句 (数字項目のけた拡張機能)

構文規則

1. PICTURE 句の数字けた位置の個数が 19~38 の数字項目、またはその数字項目を従属項目として含む集団項目に対して、EXTERNAL 句を指定してはならない。
2. PICTURE 句の数字けた位置の個数が 19~38 の数字編集項目、またはその数字編集項目を従属項目として含む集団項目に対して、EXTERNAL 句を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.25 EXTERNAL 句」

21.8.13 LINAGE 句 (数字項目のけた拡張機能)

構文規則

1. データ名 1, データ名 2, データ名 3, およびデータ名 4 に, PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。
2. 整数 1, 整数 2, 整数 3, および整数 4 に, けた数が 19~38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.41 LINAGE 句」

21.8.14 OCCURS 句 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. データ名 2 に, PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目を指定してはならない。
2. 整数 1, 整数 2 に, 19~38 けたの数字定数を指定してはならない。

書き方 2 の規則

1. データ名 1 に, PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.51 OCCURS 句」

21.8.15 PICTURE 句 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. 文字列 1 の長さは, 最大 60 文字とする。
2. 項類が数字のデータ項目に対しては, 文字列 1 で記述される数字けた位置の個数が 1~38 までの範囲内になくってはならない。ただし, 1~38 の範囲で指定できるのは, 項類が数字のデータ項目で外部 10 進形式, または内部 10 進形式のどちらかだけとする。
3. 項類が数字編集のデータ項目に対しては, 文字列 1 で記述される数字けた位置の個数が 1~38 までの範囲内になくってはならない。

4. 項類が数字編集のデータ項目のサイズは、249 けた以下でなければならない。
5. 項類が数字で外部 10 進形式、または内部 10 進形式のデータ項目に記号「P」を書く場合、数字けた位置の個数は 1～18 でなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

21.8.16 USAGE 句 (数字項目のけた拡張機能)

構文規則

PICTURE 句の数字けた位置の個数が 19～38 の数字項目、またはその数字項目を従属項目として含む集団項目に対して、USAGE BINARY/COMP/COMP-4/COMP-5/COMP-X を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

21.8.17 VALUE 句 (数字項目のけた拡張機能)

構文規則

1. 作業場所節、局所場所節の場合、定数 1、定数 2、および定数 3 に、けた数が 19～38 けたの数字定数を指定してもよい。
2. ファイル節、連絡節の場合、定数 2、定数 3 に、けた数が 19～38 けたの数字定数を指定してもよい。
3. 報告書節の場合、定数 1 にけた数が 19～38 けたの数字定数を指定してはならない。
4. サブスキーマ節の場合、定数 1、定数 2、および定数 3 に、けた数が 19～38 けたの数字定数を指定してはならない。
5. 画面節(WINDOW SECTION)の場合、定数 1 に、けた数が 19～38 けたの数字定数を指定してはならない。*

注※

画面節(WINDOW SECTION)は、Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

画面節 (WINDOW SECTION) による画面機能の詳細については、「[13. 画面節 \(WINDOW SECTION\) による画面機能](#)」を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

21.8.18 埋め込み SQL 宣言節（データベースアクセス機能）（数字項目のけた拡張機能）

この節は、Windows(x64) COBOL2002 および Linux(x64) COBOL2002 上で記述できる。

構文規則

1. 埋め込み SQL 宣言節に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目および数字編集項目を定義してはならない。埋め込み SQL 宣言節の詳細は、「9.4 データ部（埋め込み SQL 宣言節）（データベースアクセス機能）」を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「付録 D.12 データベースアクセス機能の形式」

21.9 手続き部（数字項目のけた拡張機能）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部（PROCEDURE DIVISION）」

「COBOL2002 言語 標準仕様編」 「付録 D.7 手続き部（PROCEDURE DIVISION）」

21.9.1 文に共通な指定と機能

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.6 文に共通な指定と機能」

(1) SIZE ERROR 指定とけたあふれ条件

このシステムでは、中間結果の演算でのけたあふれ条件は次のとおりとなる。

- 中間データ項目が 40 けた 10 進浮動小数点形式のとき、中間結果のオーバフロー／アンダフローが発生した時点でけたあふれとならない。ただし、結果のデータ項目が 10 進／2 進形式の数字項目で、かつ中間結果のオーバフローが発生した場合は、算術演算の結果を結果のデータ項目に格納するときにけたあふれ条件が起こる。
- 中間データ項目が内部浮動小数点形式のとき、中間結果のオーバフロー／アンダフローが発生した場合にけたあふれとなる。

また、SIZE ERROR 指定がない場合にけたあふれ条件が発生すると、次のことが起きる。

次の場合、実行時エラーとなり異常終了する。

- べき乗の評価の規則に違反した場合。
- 除算の演算中や DIVIDE 文中の除数がゼロである場合。

次の場合、実行時エラーとならないが、実行結果はけたあふれが発生している。

- 算術文の結果の絶対値が、関連する結果のデータ項目中に収め得る最大の値を超えた場合。
- 浮動小数点演算で浮動小数点オーバフロー／アンダフローが発生した場合。

演算の中間結果については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の数字項目のけた拡張機能を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.6.4 SIZE ERROR 指定とけたあふれ条件」

(2) 算術文

算術文は、ADD 文、COMPUTE 文、DIVIDE 文、MULTIPLY 文、および SUBTRACT 文である。これらには、幾つかの共通な機能がある。

- 作用対象のデータ記述が同じである必要はない。計算の過程で、すべての必要な変換と小数点の位置合わせが行われる。
- 作用対象の最大けた数は、38 けたとする。
また、作用対象の合成も 38 けた以内でなければならない。ここでの作用対象の合成とは、指定された各作用対象を小数点で位置合わせし、重ね合わせてできる仮想のデータ項目とする。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.6.6 算術文」

21.9.2 手続き部見出し

構文規則

データ名 1 およびデータ名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

「COBOL2002 言語 標準仕様編」 「付録 D.7 手続き部 (PROCEDURE DIVISION)」

21.9.3 ACCEPT 文 (数字項目のけた拡張機能)

構文規則

書き方 1 と書き方 2 の規則

一意名 1 および一意名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

21.9.4 ADD 文 (数字項目のけた拡張機能)

構文規則

書き方 1 と書き方 2 の規則

1. 一意名 1 から一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 定数 1 および定数 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

書き方 2 の規則

一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。

書き方 3 の規則

一意名 4 および一意名 5 の集団項目に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目が含まれていてもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.2 ADD 文」

21.9.5 CALL 文 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. 一意名 2, 一意名 3, および一意名 7 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

21.9.6 COMPUTE 文 (数字項目のけた拡張機能)

構文規則

書き方 1 の規則

1. 一意名 1 および算術式中に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。
3. 算術式中に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.8 COMPUTE 文」

21.9.7 DISPLAY 文 (数字項目のけた拡張機能)

構文規則

書き方 1 の規則

1. 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
2. 定数 1 に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

21.9.8 DIVIDE 文 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. 一意名 1 から一意名 4 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 定数 1 および定数 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

書き方 2 と書き方 3 の規則

1. 一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。

書き方 4 と書き方 5 の規則

1. 一意名 3 および一意名 4 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.13 DIVIDE 文」

21.9.9 EVALUATE 文 (数字項目のけた拡張機能)

構文規則

1. 一意名 1 から一意名 4、算術式 1 から算術式 4、条件 1、および条件 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 一意名 1 から一意名 4、条件 1、および条件 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。
3. 定数 1 から定数 4 に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.18 EVALUATE 文」

21.9.10 GO TO 文 (数字項目のけた拡張機能)

構文規則

書き方 2 の規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.21 GO TO 文」

21.9.11 IF 文 (数字項目のけた拡張機能)

構文規則

1. 条件 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
2. 条件 1 に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.23 IF 文」

21.9.12 INITIALIZE 文 (数字項目のけた拡張機能)

構文規則

1. 一意名 1 および一意名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
2. 一意名 1 の集団項目に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目が含まれていてもよい。
3. 定数 1 に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.24 INITIALIZE 文」

21.9.13 INSPECT 文 (数字項目のけた拡張機能)

構文規則

1. 一意名 1 から一意名 7 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
2. 一意名 1, および一意名 3 から一意名 7 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.25 INSPECT 文」

21.9.14 INVOKE 文 (オブジェクト指向) (数字項目のけた拡張機能)

構文規則

書き方 1 の規則

1. 一意名 3 から一意名 6 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を指定してはならない。
2. 定数 2 に、19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.26 INVOKE 文」

21.9.15 MOVE 文 (数字項目のけた拡張機能)

構文規則

書き方 1 と書き方 2 の規則

1. 一意名 1 および一意名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目を指定してもよい。
2. 一意名 3 および一意名 4 の集団項目に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目, および数字編集項目が含まれていてもよい。
3. 定数 1 に、けた数が 19～38 けたの数字定数を指定してもよい。

一般規則

書き方 1 の規則

送り出し側作用対象が英数字と記述されたとき、送り出し側作用対象は、次の性質を持つ符号なし数字整数であるかのように扱われる。

1. 送り出し側作用対象がデータ項目ならば、そのけた数は送り出し側データ項目の文字位置数とする。

ただし文字位置数が 38 より大きければこの限りではなく、その場合は右端の 38 個の文字位置が使用される。

2. 送り出し側作用対象が表意定数ならば、そのけた数は受け取り側作用対象のけた数と同じであり、その項目の中で表意定数が左から右へ、表意定数の規則に記述されたとおりに反復生成される。受け取り側項目が整数でないとき、けた数には小数点の右側と左側の両方のけた数が含まれる。
3. 送り出し側作用対象が英数字定数ならば、そのけた数はその定数に含まれる文字数と同じである。文字数が 38 を超えるときは送り出し側作用対象の大きさを 38 とし、定数の右端の 38 個の英数字文字だけが使用される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

21.9.16 MULTIPLY 文 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. 一意名 1 から一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 定数 1 および定数 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

書き方 2 の規則

一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.29 MULTIPLY 文」

21.9.17 PERFORM 文 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
2. 一意名 2 から一意名 7、条件 1 および条件 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
3. 整数 1 に、けた数が 19～38 けたの整数を指定してはならない。
4. 条件 1 および条件 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。
5. 定数 1 から定数 4、条件 1 および条件 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.31 PERFORM 文」

21.9.18 READ 文 (数字項目のけた拡張機能)

構文規則

すべての書き方の規則

1. 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
2. データ名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」

21.9.19 RELEASE 文 (数字項目のけた拡張機能)

構文規則

1. 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。
2. 定数 1 に、けた数が 19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.35 RELEASE 文」

21.9.20 RETURN 文 (数字項目のけた拡張機能)

構文規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.38 RETURN 文」

21.9.21 REWRITE 文 (数字項目のけた拡張機能)

構文規則

1. レコード名 1 および一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
2. 定数 1 に、けた数が 19～38 けたの数字定数を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」

21.9.22 SEARCH 文 (数字項目のけた拡張機能)

構文規則

書き方 1 と書き方 2 の規則

1. 一意名 1, 一意名 2, データ名 1, およびデータ名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
2. 一意名 1, データ名 1, およびデータ名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してはならない。
3. 条件 1, 一意名 3, および一意名 4 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
4. 定数 1 および定数 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

書き方 2 の規則

条件名 1, 条件名 2 に関連するデータ名に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.41 SEARCH 文」

21.9.23 SET 文 (数字項目のけた拡張機能)

構文規則

1. 一意名 1, 一意名 2, および一意名 23 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
2. 整数 1, 整数 2 に、19～38 けたの数字定数を指定してはならない。
3. 条件名 1 に関連するデータ名に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.43 SET 文」

21.9.24 START 文 (数字項目のけた拡張機能)

構文規則

1. データ名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.45 START 文」

21.9.25 STOP 文 (数字項目のけた拡張機能)

構文規則

- 定数 2 に、けた数が 19～38 けたの符号なし数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.46 STOP 文」

21.9.26 STRING 文 (数字項目のけた拡張機能)

構文規則

1. 一意名 1 から一意名 4 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
2. 一意名 1 および一意名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.47 STRING 文」

21.9.27 SUBTRACT 文 (数字項目のけた拡張機能)

構文規則

書き方 1 と書き方 2 の規則

1. 一意名 1 から一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。

2. 定数 1 および定数 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

書き方 2 の規則

一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してもよい。

書き方 3 の規則

一意名 4 および一意名 5 の集団項目に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目が含まれていてもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.48 SUBTRACT 文」

21.9.28 UNSTRING 文 (数字項目のけた拡張機能)

構文規則

一意名 4, 一意名 6, 一意名 7, および一意名 8 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.52 UNSTRING 文」

21.9.29 WRITE 文 (数字項目のけた拡張機能)

構文規則

1. レコード名 1 および一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。
2. 定数 1 に、けた数が 19～38 けたの数字定数を指定してもよい。
3. 一意名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
4. 整数 1 に、けた数が 19～38 けたの数字定数を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

21.10 組み込み関数 (数字項目のけた拡張機能)

ここでは、数字項目のけた拡張機能に関連する引数および関数値について説明する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「11. 組み込み関数 (Intrinsic function)」

「COBOL2002 言語 標準仕様編」 「付録 D.8 組み込み関数 (Intrinsic function)」

21.10.1 引数 (数字項目のけた拡張機能)

組み込み関数の引数の規則は、「21.6.1 関数一意名」の規則に従う。

1. MEDIAN 関数の引数に算術式、および組み込み関数は指定できない。
2. 次の組み込み関数の引数に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目および数字編集項目を指定してもよい。
 - ADDR 関数
 - LENGTH 関数

21.10.2 関数値 (数字項目のけた拡張機能)

組み込み関数の関数値の表現形式が、浮動小数点形式以外の数字型 (整数を含む) の場合、-MaxDigits38 オプションの指定に従って最大のけた数を拡張できる。ただし、関数値のけた数が返却値の意味によって固定となる場合は、拡張できない。

次に、組み込み関数の表現形式に対する数字項目のけた拡張機能の適用個所を示す。なお、次の表に記載のない関数については、けた数の拡張が適用されないため、マニュアル「COBOL2002 言語 標準仕様編」 「11. 組み込み関数 (Intrinsic function)」を参照のこと。

関数名	表現形式	備考
ANNUITY(X Y) (均等払い額)	次に示す計算式を生成し既存の処理に従う。 (1) $X=0$ の場合 $\rightarrow 1/Y$ (2) $X>0$ の場合 $\rightarrow X/(1-(1+X)**-Y)$ 関数値は、40 けた 10 進浮動小数点形式とする。	浮動小数点形式を含む場合は、内部浮動小数点形式 (8 バイト)
INTEGER(X) (引数以下の最大の整数)	1. 引数が浮動小数点項目の場合 40 けた 10 進浮動小数点形式。 ただし、有効けた数は 15 けた。 2. 引数が浮動小数点項目以外の場合 符号付きで用途は引数と等しい (数字定数のときは 2 進形式)。 (a) 引数の整数けたが 0 のとき整数部だけ 1 けた。	

関数名	表現形式	備考
	(b)引数の整数けたが 1 以上のとき ・小数部がないときは、引数の整数けた数。 ・小数部があるときは、最大けたを 18 とし引数の整数けた+1。 3. 引数が算術式のとき 40 けた 10 進浮動小数点形式。	
INTEGER-PART(X) (引数の整数部分の整数)	1. 引数が浮動小数点項目の場合 40 けた 10 進浮動小数点形式。 ただし、有効けた数は 15 けた。 2. 引数が浮動小数点項目以外の場合 符号付きで用途は引数と等しい (数字定数のときは 2 進形式)。 (a)引数の整数けたが 0 のとき整数部だけ 1 けた。 (b)引数の整数けたが 1 以上のとき整数部のけた数。 3. 引数が算術式の場合 40 けた 10 進浮動小数点形式。	
MAX(X ₁ ...X _n) (最大値)	引数が数字または整数でかつ浮動小数点形式を含まないとき 1. 引数がある場合 (a)定数のときは 2 進形式 (最大 18 けた)。 (b)定数でないときは引数 1 と同じ形式。 2. 引数が複数の場合は 40 けた 10 進浮動小数点形式。	浮動小数点形式を含む場合は、内部浮動小数点形式 (8 バイト)
MEAN(X ₁ ...X _n) (平均値)	引数が数字または整数でかつ浮動小数点形式を含まないとき 1. 引数がある場合 (a)定数のときは 2 進形式 (最大 18 けた)。 (b)定数でないときは引数 1 と同じ形式。 2. 引数が複数の場合は 40 けた 10 進浮動小数点形式。	浮動小数点形式を含む場合は、内部浮動小数点形式 (8 バイト)
MEDIAN(X ₁ ...X _n) (中央値)	1. 引数がある場合 (a)引数が定数のとき 2 進形式 (最大 18 けた)。 (b)引数が定数でないとき、引数と同じ形式。 2. 引数がすべて整数の場合 40 けた 10 進浮動小数点形式。	左記以外の場合は、内部浮動小数点形式 (8 バイト)
MIDRANGE(X ₁ ... X _n) (最大値と最小値の平均)	浮動小数点形式を含まない場合は 40 けた 10 進浮動小数点形式 1. 引数がある場合 (a)定数のときは 2 進形式 (最大 18 けた)。 (b)定数でないときは引数 1 と同じ形式。 2. 引数が複数の場合は 40 けた 10 進浮動小数点形式。	浮動小数点形式を含む場合は、内部浮動小数点形式 (8 バイト)
MIN(X ₁ ... X _n) (最小値)	引数が数字または整数でかつ浮動小数点形式を含まないとき 1. 引数がある場合 (a)定数のときは 2 進形式 (最大 18 けた)。 (b)定数でないときは引数 1 と同じ形式。 2. 引数が複数の場合は 40 けた 10 進浮動小数点形式。	浮動小数点形式を含む場合は、内部浮動小数点形式 (8 バイト)

関数名	表現形式	備考
MOD(X Y) (法)	次に示す算術式の結果の形式に従う。 <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> $(X - (Y * \text{FUNCTION INTEGER}(X/Y)))$ </div> 関数値は、40 けた 10 進浮動小数点形式。	
NUMVAL(X) (文字列→数字変換)	関数値の表現形式は、外部 10 進形式 18 バイトとなる。 整数けた数：18-Dmax [*] 小数けた数：Dmax [*] Dmax が 19 けたを超えた場合は、Dmax を 18 けたに補正する。	
NUMVAL-C(X) (編集文字列→数字変換)	関数値の表現形式は、外部 10 進形式 18 バイトとなる。 整数けた数：18-Dmax [*] 小数けた数：Dmax [*] Dmax が 19 けたを超えた場合は、Dmax を 18 けたに補正する。	
RANGE(X ₁ ... X _n) (範囲)	<ol style="list-style-type: none"> 1. 引数が 1 個の場合は、2 進形式。(整数部 1 けただけ。値は 0。) 2. 引数が複数で、浮動小数点形式を含まない場合は、40 けた 10 進浮動小数点形式。 3. 引数が複数で、浮動小数点形式を含む場合は、内部浮動小数点形式 8 バイト。 	
REM(X Y) (剰余)	次に示す算術式の結果の形式に従う。 <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> $(X - (Y * \text{FUNCTION INTEGER-PART}(X/Y)))$ </div> 関数値は、40 けた 10 進浮動小数点形式。	
SUM(X ₁ ...X _n) (合計)	次に示す算術式の結果の形式に従う。 <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> $X_1 \dots + X_n$ </div> <ul style="list-style-type: none"> • 引数一つの場合 <ol style="list-style-type: none"> (a) 定数のときは 2 進形式 (最大 18 けた)。 (b) 定数でないときは引数 1 と同じ形式。 • 引数が複数の場合は 40 けた 10 進浮動小数点形式。 	浮動小数点形式を含む場合は、内部浮動小数点形式 (8 バイト)
VARIANCE (X ₁ ...X _n) (分散の近似値)	関数値 <ol style="list-style-type: none"> (1) 引数が 1 個のときは、2 進形式。 整数けたは 1、小数けたは 0。 (2) 引数が複数で、浮動小数点形式を含まないとき 40 けた 10 進浮動小数点形式。 (3) 引数が複数で、浮動小数点形式を含むとき 内部浮動小数点形式 8 バイト。 	

注※

Dmax は、この関数を含む算術式のすべての作用対象と算術文、および転記の受け取り側作用対象のうち最大の小数けた数を示す。

21.11 拡張仕様 (数字項目のけた拡張機能)

ここでは、数字項目のけた拡張機能と組み合わせて使用する拡張仕様の文について説明する。

21.11.1 ACCEPT 文

(1) 画面節による画面機能

この ACCEPT 文 (画面節) は、Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

構文規則

書き方 4 の規則 (画面節(SCREEN SECTION)) (画面名)

- 一意名 1, 一意名 2, および一意名 3 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。
- 整数 1, 整数 2, および整数 3 に、けた数が 19~38 けたの数字定数を指定してはならない。

書き方 4 の規則 (画面節(SCREEN SECTION)) (一意名)

- 一意名 1, 一意名 2, 一意名 3, および一意名 4 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。
- 一意名 1 に、PICTURE 句の数字けた位置の個数が 19~38 の数字編集項目を指定してはならない。
- 整数 1, 整数 2, および整数 3 に、けた数が 19~38 けたの数字定数を指定してはならない。

書き方 5 の規則 (画面節(WINDOW SECTION))

一意名 7, 一意名 8 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目, および数字編集項目を指定してはならない。

なお、画面節 (SCREEN SECTION) による画面機能の ACCEPT 文の詳細については、「[12.3.1 ACCEPT 文 \(SCREEN SECTION\)](#)」を参照のこと。また、画面節 (WINDOW SECTION) による画面機能の ACCEPT 文の詳細については、「[13.5.2 ACCEPT 文 \(WINDOW SECTION\)](#)」を参照のこと。

(2) コマンド行のアクセス機能

構文規則

書き方 1 の規則 (コマンド行の引数や環境変数へのアクセス)

一意名 9 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。

ACCEPT COMMAND-LINE

一意名に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目, および数字編集項目を指定してはならない。

なお、コマンド行のアクセス機能のコマンド行の引数や環境変数へのアクセスの詳細については、「10.1 コマンド行、環境変数へのアクセス」を参照のこと。また、ACCEPT COMMAND-LINEの詳細については、「10.2.2 手続き部（コマンド行へのアクセス（一括取得）」の「(1) ACCEPT COMMAND-LINE」を参照のこと。

21.11.2 CALL 文（CALL 文の引数指定の拡張）

構文規則

一意名 5、一意名 6 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。

なお、CALL 文の引数指定の拡張の詳細については、「25.4.5 CALL 文の引数指定の拡張」を参照のこと。

21.11.3 COMPUTE 文（アドレス操作機能）

構文規則

書き方 1 の規則

- 一意名 2 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目および数字編集項目を指定してもよい。

書き方 2 の規則

- 一意名 4 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してもよい。
- 定数 2 に、けた数が 19～38 けたの数字定数を指定してもよい。

一般規則

書き方 2 の規則

- 一意名 4 に PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定した場合、格納する値は 8 バイトの記憶域に 2 進形式で格納できる値を超えてはならない。
- 定数 2 にけた数が 19～38 けたの数字定数を指定する場合、指定する値は 8 バイトの記憶域に 2 進形式で格納できる値を超えてはならない。

なお、アドレス操作機能の COMPUTE 文の詳細については、「15.4.2 COMPUTE 文（アドレス操作機能）」を参照のこと。

21.11.4 DISABLE 文（データコミュニケーション機能）

この文は、AIX(64) COBOL2002 と Linux(x64) COBOL2002 上で記述できる。

構文規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

なお、データコミュニケーション機能の DISABLE 文の詳細については、「[8.3.2 DISABLE 文 \(データコミュニケーション機能\)](#)」を参照のこと。

21.11.5 DISPLAY 文

(1) 画面節による画面機能

この DISPLAY 文 (画面節) は、Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

構文規則

書き方 3 の規則 (画面節(SCREEN SECTION)) (画面名)

- 一意名 1, 一意名 2, および一意名 3 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
- 整数 1, 整数 2, および整数 3 に、けた数が 19～38 けたの数字定数を指定してはならない。

書き方 3 の規則 (画面節(SCREEN SECTION)) (一意名/定数)

- 一意名 1, 一意名 2, 一意名 3, および一意名 4 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。
- 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字編集項目を指定してはならない。
- 定数 1, 整数 1, 整数 2, および整数 3 に、けた数が 19～38 けたの数字定数を指定してはならない。

書き方 4 の規則 (画面節(WINDOW SECTION))

一意名 6 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

なお、画面節 (SCREEN SECTION) による画面機能の DISPLAY 文の詳細については、「[12.3.2 DISPLAY 文 \(SCREEN SECTION\)](#)」を参照のこと。また、画面節 (WINDOW SECTION) による画面機能の DISPLAY 文の詳細については、「[13.5.4 DISPLAY 文 \(WINDOW SECTION\)](#)」を参照のこと。

(2) コマンド行のアクセス機能

構文規則

書き方 3 の規則 (コマンド行の引数や環境変数へのアクセス)

- 一意名 7 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目を指定してはならない。

2. 整数 4 に、けた数が 19～38 けたの数字定数を指定してはならない。

DISPLAY COMMAND-LINE

1. 一意名に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。
2. 定数に、けた数が 19～38 けたの数字定数を指定してはならない。

なお、コマンド行のアクセス機能のコマンド行の引数や環境変数へのアクセスの詳細については、「10.1 コマンド行、環境変数へのアクセス」を参照のこと。また、DISPLAY COMMAND-LINE の詳細については、「10.2.2 手続き部 (コマンド行へのアクセス (一括取得))」の「(2) DISPLAY COMMAND-LINE」を参照のこと。

(3) CSV ファイル入出力機能

CSV ファイル入出力機能は、Windows(x64) COBOL2002 で使用できる。

構文規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、数字編集項目、またはその数字項目、数字編集項目を従属項目として含む集団項目を指定してはならない。

CSV ファイル入出力機能の詳細については、「17.3.2 手続き部 (CSV ファイル入出力機能)」の「(2) DISPLAY 文 (CSV ファイル入出力機能)」を参照のこと。

21.11.6 ENABLE 文 (データコミュニケーション機能)

この文は、AIX(64) COBOL2002 と Linux(x64) COBOL2002 上で記述できる。

構文規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

なお、データコミュニケーション機能の ENABLE 文の詳細については、「8.3.3 ENABLE 文 (データコミュニケーション機能)」を参照のこと。

21.11.7 ENTRY 文

構文規則

データ名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してもよい。

なお、ENTRY 文の詳細については、「25.4.8 ENTRY 文」を参照のこと。

21.11.8 ERASE 文 (画面節(WINDOW SECTION))

この文は、Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

構文規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。

なお、画面節 (WINDOW SECTION) による画面機能の ERASE 文の詳細については、「13.5.5 ERASE 文 (WINDOW SECTION)」を参照のこと。

21.11.9 EXAMINE 文

構文規則

1. 一意名 1 に、PICTURE 句の数字けた位置の個数が 19～38 の数字項目、および数字編集項目を指定してはならない。
2. 定数 1 から定数 4 に、けた数が 19～38 けたの数字定数を指定してはならない。

なお、EXAMINE 文の詳細については、「25.4.9 EXAMINE 文と TALLY 特殊レジスタ」を参照のこと。

21.11.10 FETCH 文 (構造型データベース (XDM/SD) 操作シミュレーション機能)

構文規則

1. 一意名 1 に、けた数が 19～38 けたの数字項目を指定してはならない。
2. 整数 1 に、けた数が 19～38 けたの数字定数を指定してはならない。

なお、構造型データベース (XDM/SD) 操作シミュレーション機能の FETCH 文の詳細については、「19.1.2 手続き部 (構造型データベース (XDM/SD) 操作シミュレーション機能)」の「(4) FETCH 文」を参照のこと。

21.11.11 FIND 文 (構造型データベース (XDM/SD) 操作シミュレーション機能)

構文規則

1. 一意名 1 に、けた数が 19～38 けたの数字項目を指定してはならない。
2. 整数 1 に、けた数が 19～38 けたの数字定数を指定してはならない。

なお、構造型データベース (XDM/SD) 操作シミュレーション機能の FIND 文の詳細については、「19.1.2 手続き部 (構造型データベース (XDM/SD) 操作シミュレーション機能)」の「(5) FIND 文」を参照のこと。

21.11.12 INVOKE 文 (OLE2 オートメーションインタフェース機能)

この文は、Windows(x64) COBOL2002 上で記述できる。

構文規則

一般的な OLE メソッドの操作

1. 一意名 9 および一意名 11 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。
2. 一意名 9 に PICTURE 句の数字けた位置の個数が 19~38 の数字編集項目を指定してはならない。
3. 定数 4 に 19~38 けたの数字定数を指定してはならない。

なお、OLE2 オートメーションインタフェース機能の詳細については、「18.3.1 INVOKE 文 (OLE2 オートメーションインタフェース機能)」を参照のこと。

21.11.13 MOVE 文 (日付と時刻用)

構文規則

1. 一意名に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目を指定してもよい。

なお、日付と時刻用の MOVE 文の詳細については、「25.4.11 MOVE 文 (日付と時刻用)」を参照のこと。

21.11.14 RECEIVE 文 (データコミュニケーション機能)

この文は、AIX(64) COBOL2002 と Linux(x64) COBOL2002 上で記述できる。

構文規則

- 一意名 1 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目、および数字編集項目を指定してはならない。

なお、データコミュニケーション機能の RECEIVE 文の詳細については、「8.3.4 RECEIVE 文 (データコミュニケーション機能)」を参照のこと。

21.11.15 REPLY 文 (画面機能(WINDOW SECTION))

この文は、Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。

構文規則

1. 一意名 1, 一意名 2 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目, および数字編集項目を指定してはならない。
2. データ名 2 からデータ名 5 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。
3. 整数 1 から整数 4 に、19~38 けたの整数を指定してはならない。

なお、画面節 (WINDOW SECTION) による画面機能の REPLY 文の詳細については、「[13.5.7 REPLY 文 \(WINDOW SECTION\)](#)」を参照のこと。

21.11.16 SEND 文 (データコミュニケーション機能)

この文は、AIX(64) COBOL2002 と Linux(x64) COBOL2002 上で記述できる。

構文規則

書き方 1 の規則

一意名 1, 一意名 2, 一意名 3, および一意名 4 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目, および数字編集項目を指定してはならない。

なお、データコミュニケーション機能の SEND 文の詳細については、「[8.3.6 SEND 文 \(データコミュニケーション機能\)](#)」を参照のこと。

21.11.17 SET 文

この文は、Windows(x64) COBOL2002 および AIX(64) COBOL2002 上で記述できる。ただし、OLE2 オートメーションインタフェース機能は Windows(x64) COBOL2002 だけで記述できる。

(1) 画面機能 (WINDOW SECTION)

構文規則

書き方 13 の規則

一意名 13 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目, および数字編集項目を指定してはならない。

書き方 14 の規則

整数 3 に、19~38 けたの整数を指定してはならない。

なお、画面節 (WINDOW SECTION) による画面機能の SET 文の詳細については、「13.5.8 SET 文 (WINDOW SECTION)」を参照のこと。

(2) ポインタ項目の参照

構文規則

1. ADDRESS OF の一意名に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目を指定してもよい。

ポインタ項目の参照の詳細については、「25.4.3 ポインタ項目の参照」を参照のこと。

(3) OLE2 オートメーションインタフェース機能

構文規則

書き方 1 の構文規則

一意名 14 に PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。

書き方 2 の構文規則

1. 一意名 19 に PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目を指定してはならない。
2. 定数 3 に 19~38 けたの数字定数を指定してはならない。

OLE2 オートメーションインタフェース機能の詳細については、「18.3.2 SET 文 (OLE2 オートメーションインタフェース機能)」を参照のこと。

21.11.18 TRANSFORM 文

構文規則

書き方 1 の規則

一意名 1 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目、および数字編集項目を指定してはならない。

なお、TRANSFORM 文の詳細については、「25.4.16 TRANSFORM 文」を参照のこと。

21.11.19 探索条件文 (構造型データベース (XDM/SD) 操作シミュレーション機能)

構文規則

1. 一意名 1, 一意名 2 に、PICTURE 句の数字けた位置の個数が 19~38 の数字項目を指定してはならない。

2. 定数 1, 定数 2 に, けた数が 19~38 けたの数字定数を指定してはならない。

21.11.20 WRITE 文 (印刷制御文字の直接指定, 区域外書き出し)

構文規則

1. レコード名 1 および一意名 1 に, PICTURE 句の数字けた位置の個数が 19~38 の数字項目, および数字編集項目を指定してもよい。
2. 整数 2 に, けた数が 19~38 けたの数字定数を指定してはならない。

なお, WRITE 文 (印刷制御文字の直接指定, 区域外書き出し) の詳細については, 「[25.4.19 WRITE 文 \(印刷制御文字の直接指定, 区域外書き出し\)](#)」を参照のこと。

21.11.21 埋め込み SQL 文 (データベースアクセス機能)

この文は, Windows(x64) COBOL2002 および Linux(x64) COBOL2002 上で記述できる。

構文規則

1. 埋め込み SQL 文に, PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目を指定してはならない。
2. 埋め込み SQL 文に, けた数が 19~38 けたの数字定数を指定してはならない。

なお, 埋め込み SQL 文 (データベースアクセス機能) の詳細については, 「[9.2 埋め込み SQL 文 \(データベースアクセス機能\)](#)」を参照のこと。

21.11.22 条件式 (ポインタ項目の参照)

構文規則

ADDRESS OF の一意名に, PICTURE 句の数字けた位置の個数が 19~38 の数字項目および数字編集項目を指定してもよい。

なお, 条件式 (ポインタ項目の参照) の詳細については, 「[25.4.3 ポインタ項目の参照](#)」を参照のこと。

22

動的長基本項目機能

この章では、動的長基本項目について説明する。

22.1 動的長基本項目とは

動的長基本項目 (dynamic-length elementary item) は、DYNAMIC LENGTH 句の定義により、英数字項目または日本語項目の長さを実行時に変えることができるデータ項目である。

動的長基本項目を記述したプログラム例を次に示す。

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SAMPLE00.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
SPECIAL-NAMES.  
DYNAMIC LENGTH STRUCTURE C-STRING IS C-STATIC-STRUCTURE.  
    *> 動的長構造名を”C-STRING”として定義する。  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 DYNAMICDATA PIC X DYNAMIC LENGTH C-STRING LIMIT 20.  
    *> 格納できる文字列の最大長を20とした、英数字の動的長基本項目を定義する。  
PROCEDURE DIVISION.  
MOVE 'abc' TO DYNAMICDATA.  
    *> DYNAMICDATAに文字列'abc'が格納され、そのデータ長は3バイトとなる。  
    *> 文字列'abc'の後ろに終端文字(X'00')が設定される。  
MOVE 'abcdef' TO DYNAMICDATA.  
    *> DYNAMICDATAに文字列'abcdef'が格納され、そのデータ長は6バイトとなる。  
    *> 文字列'abcdef'の後ろに終端文字(X'00')が設定される。  
    *> DYNAMICDATAは、C言語など他言語へ渡す文字列として使用できる。
```

動的長基本項目は格納する値によって長さが変わるため、データ項目の長さが非常に長くなるおそれがある。そのため、LIMIT 指定により、動的長基本項目の最大長を定義できる。また、動的長基本項目に新しい値が格納されたとき、項目の長さは自動的に調整される。

なお、動的長基本項目は、動的長基本項目に新しい値が格納された時点での同じ長さを持つ固定長のデータ項目と同様に扱う。

ここでは、動的長基本項目が指定できる構文や規則について説明する。

なお、この章で記載のない構文規則および一般規則については、特に断り書きがない限り、マニュアル「COBOL2002 言語 標準仕様編」を参照のこと。

22.2 動的長基本項目が使用できる個所

22.2.1 定義できるソース単位

動的長基本項目を定義できるソース単位を次に示す。

表 22-1 動的長基本項目を定義できるソース単位

定義名	動的長基本項目への対応
プログラム定義	○
関数定義	×
クラス定義	×
インタフェース定義	×
ファクトリ定義	×
オブジェクト定義	×
メソッド定義	×

(凡例)

○：定義できる

×：定義できない

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「6.3 ソース単位」

22.2.2 定義できる節

動的長基本項目を定義できる節を次に示す。

表 22-2 動的長基本項目を定義できる節

定義名	動的長基本項目への対応
ファイル節 (FILE SECTION)	×
作業場所節 (WORKING-STORAGE SECTION)	○
局所場所節 (LOCAL-STORAGE SECTION)	×
連絡節 (LINKAGE SECTION)	○
報告書節 (REPORT SECTION)	×
画面節 (SCREEN SECTION)	×
画面節 (WINDOW SECTION)	×

定義名	動的長基本項目への対応
通信節 (COMMUNICATION SECTION)	×
サブスキーマ節 (SUBSCHEMA SECTION)	×

(凡例)

○：定義できる

×：定義できない

また、作業場所節や連絡節で定義された動的長基本項目を、「22.2.3 対象となる文」に示す手続き部の許可された個所以外に指定することはできない。環境部やデータ部での参照もできない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

22.2.3 対象となる文

動的長基本項目が指定できる文を次に示す。次に示す個所以外には指定できない。

表 22-3 動的長基本項目が指定できる文

文	指定できる個所
手続き部見出し	データ名 1 (ただし BY REFERENCE だけ)
CALL 文	書き方 1 と書き方 2 の一意名 2 (BY REFERENCE, BY CONTENT)
DISPLAY 文	書き方 1 の一意名 1
EVALUATE 文	<ul style="list-style-type: none"> 一意名 1 と一意名 2 条件 1 と条件 2
IF 文	条件 1
INITIALIZE 文	一意名 1
MOVE 文	書き方 1 の一意名 1, 一意名 2

対象ではない文に指定した場合は、エラーメッセージが出力される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8 文」

22.2.4 対象となる組み込み関数

動的長基本項目が指定できる組み込み関数を次に示す。次に示す個所以外には指定できない。

表 22-4 動的長基本項目を指定できる組み込み関数

組み込み関数	指定できる個所
LENGTH 関数	引数 1

対象ではない組み込み関数に指定した場合は、エラーメッセージが出力される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「11.32 LENGTH 関数」

22.3 全般規則

22.3.1 表意定数

一般規則

1. 次の規則は動的長基本項目には該当しない。

表意定数が VALUE 句に指定されている場合、またはデータ項目や定数や中間結果項目に関連づけられている場合、文字列は 1 文字ずつ右側へと反復されて、でき上がった文字列の長さは、関連するデータ項目か定数の中間結果項目の中の文字位置の個数以上になる。

2. 動的長基本項目に表意定数を関連付ける場合、次の規則が適用される。

- 表意定数が「ALL 定数」以外の場合、文字列の長さは 1 文字とする。
- 表意定数が「ALL 定数」の場合、文字列の長さは定数の長さとする。

(例)

```
01 MY-DYNAMIC PIC X DYNAMIC LENGTH C-STRING
   LIMIT 20.
   ;
MOVE 'abc' TO MY-DYNAMIC.
  *> 動的長基本項目MY-DINAMICに'abc'が転記され、MY-DYNAMICは長さが3となる。

MOVE SPACE TO MY-DYNAMIC.
  *> SPACEは3文字でなく空白1文字が転記され、MY-DYNAMICは長さが1となる。

MOVE ALL 'abc' TO MY-DYNAMIC.
  *> 'abc'が転記され、MY-DYNAMICは長さが3となる。

MOVE ALL SPACE TO MY-DYNAMIC.
  *> SPACEは1文字のため、空白1文字が転記され、MY-DYNAMICは長さが1となる。
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「4.2.1(2)(d) 表意定数の値」

22.4 環境部 (動的長基本項目機能)

22.4.1 特殊名段落 (動的長基本項目機能)

一般形式

```
SPECIAL-NAMES.  
[ DYNAMIC LENGTH STRUCTURE句 ] ...
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落 (SPECIAL-NAMES)」

22.4.2 DYNAMIC LENGTH STRUCTURE 句

一般形式

```
DYNAMIC LENGTH STRUCTURE 動的長構造名1 IS 物理構造名1
```

構文規則

1. 物理構造名 1 は、動的長基本項目に対する物理的な構造を指定する。

このシステムでは物理構造名 1 は C-STATIC-STRUCTURE とする。それ以外の名前は指定できない。C-STATIC-STRUCTURE によって定義される構造は次のとおりである。

- データが終端文字で終わる。内部的には最大長+1 文字の領域を確保する。なお、終端文字は、英数字項目の場合 X'00'、日本語項目の場合 X'0000'となる。
- DYNAMIC LENGTH 句の LIMIT 指定の値、または最初に格納されたデータの長さを上限として動作する。LIMIT 指定がある場合は LIMIT 指定の値が優先される。上限以上の長さのデータを転記しても動的に拡張されず、転記時に切り捨てられる。

(例)

DYNAMIC LENGTH 句の LIMIT 指定が 4 の場合、4+1 文字の領域が確保される。この領域に 3 バイト文字が格納されると残りの領域に終端文字が入る。



最初に格納されたデータには次の二つのケースがある。

- 作業場所節で動的長基本項目が定義され、かつ VALUE 句で初期値を与えた場合は、VALUE 句に指定した定数の長さが上限となる。
- ほかの言語から終端文字(X'00')で終わる文字列が渡された場合は、ほかの言語で渡された領域に格納されているデータの長さを上限とする。

VALUE 句の指定がない作業場所節の動的長基本項目は、最左端から最大長+1 の領域を終端文字で初期化する。

一般規則

1. C-STATIC-STRUCTURE に指定した動的長構造名 1 で定義されるデータの終わりには終端文字が入る。
英数字項目の終端文字は X'00'、日本語項目の終端文字は X'0000'となる。
英数字項目または日本語項目の終端文字は、このシステムが設定する。
なお、C 言語から字類が日本語の動的長基本項目に文字列を渡す場合も、データの終わりが X'0000'でなければならない。また、字類が日本語の動的長基本項目では文字列の先頭から 2 バイトずつ X'0000'を探するため、C 言語から渡すデータに 1 バイト文字が混在してはならない。
2. DYNAMIC LENGTH STRUCTURE 句を指定したプログラムに-H8Switch オプションは指定できない。

22.5 データ部（動的長基本項目機能）

22.5.1 DYNAMIC LENGTH 句

DYNAMIC LENGTH 句は、実行時にそのデータ項目の長さが変化することを定義する。

一般形式

```
DYNAMIC LENGTH 動的長構造名1 [ LIMIT IS 整数1 ]
```

構文規則

1. PICTURE 句はこの項目の字類を示す。PICTURE 句に指定される文字列は、1文字の X または N でなければならない。
ただし、-JPN,V3JPN オプション指定時、PICTURE 文字列は1文字の X でなければならない。
2. 動的長構造名1を指定した場合、特殊名段落の DYNAMIC LENGTH STRUCTURE 句に指定した動的長構造名1と対応する。
3. 作業場所節で定義された DYNAMIC LENGTH 句の LIMIT 指定を省略した場合は、VALUE 句がなくてはならない。
4. 整数1は、英数字項目の場合は、1～16,777,214の値でなければならない。また、日本語項目の場合は、1～16,382の値でなければならない。
5. 01 レベルおよび 77 レベルの項目以外に指定してはならない。
6. DYNAMIC LENGTH 句はレベル番号、データ名、PICTURE 句、USAGE 句、VALUE 句以外と同時に指定できない。
7. 作業場所節で定義した DYNAMIC LENGTH 句で、LIMIT 指定と VALUE 句を同時に指定した場合、VALUE 句に指定した定数の長さは LIMIT 指定の値を超えてはならない。
8. 動的長基本項目を被再定義項目としてはならない。
9. 動的長基本項目を部分参照してはならない。
10. 動的長基本項目に対して、条件名記述項を関連付けてはならない。
11. 集団項目に対して、DYNAMIC LENGTH 句を指定してはならない。

一般規則

1. DYNAMIC LENGTH 句は実行時にその長さが変化するデータ項目を定義する。データ項目の最小の長さは0である。PICTURE 句は字類を決定する。

備考

PIC X または PIC N の指定によって、この動的長基本項目の字類が、英数字か日本語かが決まる。

2. 整数1はそのデータ項目に格納できる文字列の最大長を定義する。LIMIT 指定がない場合の最大長は次のとおりである。

このシステムでは次の場合に LIMIT 指定を省略できる。

- (a) 作業場所節に定義し、VALUE 句が指定されている場合
- (b) 連絡節に定義した場合

LIMIT 指定が省略された場合の、それぞれの最大長は次で定義される。

- (a) VALUE 句で指定した文字列の長さ
 - (b) 呼び出し元プログラムで設定されたデータの長さ（終端文字の前までの長さ）
- ただし(b)の場合、データの長さが次に示す値を超えている場合は、次の値を最大長とする。

動的長基本項目の字類	LIMIT の最大長（文字数）
英数字	16,777,214
日本語	16,382

(例 1)

```
WORKING-STORAGE SECTION.  
  01 DYNAMICDATA PIC X DYNAMIC LENGTH  
      C-STRING VALUE 'ABCDE'. *> LIMIT 5 と同等
```

(例 2)

```
LINKAGE SECTION.  
  01 DYNADATA PIC X DYNAMIC LENGTH C-STRING .  
PROCEDURE DIVISION USING DYNADATA .  
  *> 呼び出し元プログラムで設定されたデータが格納されているDYNADATAの内容が5バイトのユー  
ザデータ+終端文字の場合、LIMIT 5と同等  
  *> 呼び出し元プログラムで設定されたデータが格納されているDYNADATAの内容が16,777,214バ  
イトを超えている場合、LIMIT 16777214 と同等
```

作業場所節で LIMIT 指定と VALUE 句が同時に指定した場合は、LIMIT 指定で最大長を決定する。連絡節で LIMIT 指定を指定した場合は、呼び出し元プログラムで設定されたデータの長さより LIMIT 指定を優先する。

- 3. 動的長基本項目が連絡節に指定され、かつ次のどちらかに該当する場合、その動的長基本項目を受け取り側作用対象に指定してはならない。
 - C 言語で記述されたプログラムから COBOL プログラムが呼ばれる場合
 - 連絡節に指定した動的長基本項目に LIMIT 指定がない場合

備考

このシステムでは、次の個所で動的長基本項目が参照されたとき、（最大長+1）文字目までに終端文字が設定されているかチェックする。終端文字が設定されていない場合は実行時エラーとなる。

- DISPLAY 文の一意名 1
- MOVE 文の送り出し側作用対象
- 比較条件の作用対象
- LENGTH 関数の引数

- 連絡節で定義された仮引数

22.6 手続き部（動的長基本項目機能）

22.6.1 引数の適合

(1) 動的長基本項目

(a) 参照渡しまたは内容渡しの動的長基本項目

1. COBOL 間の呼び出しで、実引数と仮引数のどちらかが動的長基本項目である場合、実引数と仮引数のどちらかが動的長基本項目であるときは、対応する仮引数や実引数も同じ字類の動的長基本項目でなくてはならない。
2. C-STATIC-STRUCTURE に関連付いた動的長基本項目の場合、その LIMIT 指定の値が一致していなければならない。

COBOL 間の呼び出しで、実引数、および仮引数に指定した動的長基本項目の LIMIT 有無と適合規則の関係を次に示す。

表 22-5 LIMIT 指定有無と適合規則

項番	実引数（呼び出し元）	仮引数（呼び出し先）	LIMIT 値の関係	結果
1	LIMIT あり	LIMIT あり	LIMIT の値が一致する	適合する
2	LIMIT あり	LIMIT あり	LIMIT の値が異なる	適合しない
3	LIMIT あり	LIMIT なし	該当なし	適合する
4	LIMIT なし	LIMIT あり	該当なし	適合しない
5	LIMIT なし	LIMIT なし	該当なし	適合する

注 1

VALUE 句の指定があり、LIMIT 指定が省略されている場合、VALUE 句の値のサイズで LIMIT 指定を仮定する。

注 2

この規則は、適合規則およびプログラム間整合性チェック（-DebugCompati オプションまたは-TDInf オプション）に適用する。プログラム間整合性チェックの詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.7.1(2) 基本項目」

22.6.2 CALL 文（動的長基本項目機能）

一般規則

1. COBOL 間の呼び出しで、引数に指定した動的長基本項目は、適合規則およびプログラム間整合性チェック (-DebugCompati オプションまたは-TDInf オプション) の対象とする。適合規則は次のとおりとする。

- 動的長基本項目を呼び出し元プログラムの引数とする場合は、呼び出し先プログラムの引数も同じ字類の動的長基本項目であること。
- C-STATIC-STRUCTURE に関連付いた動的長基本項目の場合、その LIMIT 指定の値が一致していること。

連絡節に記述した動的長基本項目の LIMIT 指定が省略された場合の規則は、「表 22-5 LIMIT 指定有無と適合規則」に従う。

プログラム間整合性チェックの詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

2. BY CONTENT の一意名 2 が連絡節に定義した動的長基本項目の場合、その動的長基本項目には LIMIT 指定がなければならない。

(例)

```
LINKAGE SECTION.  
01 DYNADATA PIC X DYNAMIC LENGTH C-STRING.  
PROCEDURE DIVISION USING DYNADATA.  
CALL 'MAGO' USING BY CONTENT DYNADATA.  
*> この使い方をする場合、DYNADATAにはLIMIT指定が必要。コンパイルエラーとなる。
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

22.6.3 DISPLAY 文 (動的長基本項目機能)

一般規則

動的長基本項目を一意名 1 に指定した場合、その時点のデータを適切なハードウェア装置に転送する。転送するデータに終端文字は含まない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

22.6.4 EVALUATE 文 (動的長基本項目機能)

一般規則

1. 動的長基本項目が指定できる条件式は、比較条件だけである。
2. 動的長基本項目と固定長項目を比較する場合、動的長基本項目はその時点の長さを持つ固定長項目と同等に扱う。終端文字は比較の対象に含まない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.18 EVALUATE 文」

22.6.5 IF 文 (動的長基本項目機能)

一般規則

1. 動的長基本項目が指定できる条件式は、比較条件だけである。
2. 動的長基本項目と固定長項目を比較する場合、動的長基本項目はその時点の長さを持つ固定長項目と同等に扱う。終端文字は比較の対象に含まない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.23 IF 文」

22.6.6 INITIALIZE 文 (動的長基本項目機能)

構文規則

一意名 1 に、連絡節で定義した LIMIT 指定のない動的長基本項目を指定してはならない。

一般規則

REPLACING 指定がない場合は、動的長基本項目が初期化されると長さが 0 になる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.24 INITIALIZE 文」

22.6.7 MOVE 文 (動的長基本項目機能)

構文規則

1. 一意名 2 に、連絡節で定義した LIMIT 指定のない動的長基本項目を指定してはならない。

一般規則

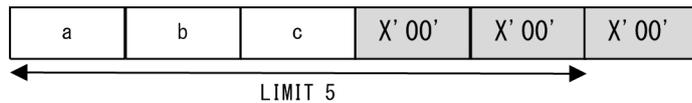
1. 動的長基本項目を送り出し側作用対象に指定した場合、その時点の長さを持つ固定長項目と同等に扱う。転記するデータに終端文字は含まない。
2. 動的長基本項目を受け取り側作用対象に指定した場合、次の動作となる。
 - (a) n 以下の長さの文字列を転記した場合、文字列の転記後、末尾から n+1 まで終端文字を付加する。
 - (b) n を超える長さの文字列を転記した場合、n の長さ分の文字列を転記後、n+1 の位置に終端文字を付加する。ここで n は、受け取り側作用対象に指定した動的長基本項目の最大長とする。

(例)

```
01 DYNADATA PIC X DYNAMIC LENGTH C-STRING LIMIT 5.
```

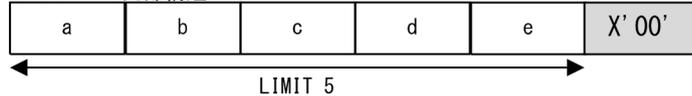
(a) MOVE 'abc' TO DYNADATA.

DYNADATAの内部構造



(b) MOVE 'abcdefgh' TO DYNADATA.

DYNADATAの内部構造



注 網掛け部分は、このシステムが使用する領域である。

注意事項

LOW-VALUE(X'00')が含まれた固定長基本項目または表意定数 LOW-VALUE を動的長基本項目に転記した場合、動的長基本項目の長さは転記された LOW-VALUE の前までとなる。

```
01 DYNADATA PIC X DYNAMIC LENGTH C-STRING LIMIT 5.  
01 FIXDATA PIC X(4) VALUE X'61620064'.
```

```
MOVE FIXDATA TO DYNADATA. *> DYNADATAの長さは2となる  
MOVE LOW-VALUE TO DYNADATA. *> DYNADATAの長さは0となる
```

3. 動的長基本項目を送り出し側作用対象または受け取り側作用対象に指定した場合、受け取り側作用対象に複数の項目を指定してはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

22.7 組み込み関数（動的長基本項目機能）

22.7.1 LENGTH 関数（動的長基本項目機能）

一般規則

動的長基本項目を引数 1 に指定した場合、関数値は、動的長基本項目に現在格納されている値の長さとなる。長さに終端文字は含まない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「11.32 LENGTH 関数」

22.8 コンパイラ環境変数またはコンパイラオプションとの関連

22.8.1 コンパイラ環境変数 CBLVALUE

規則

動的長基本項目はコンパイラ環境変数 CBLVALUE に指定した初期値で初期化されない。

コンパイラ環境変数 CBLVALUE の詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラ環境変数の詳細を参照のこと。

22.8.2 -JPN オプション

規則

-JPN,Alnum オプションを指定した場合、動的長基本項目はすべて字類を英数字として扱う。また、PIC N で指定した動的長基本項目の LIMIT 指定の値は、その 2 倍の値を指定したものと仮定する。

-JPN,Alnum オプションについては、「24.3.6 日本語項目の扱い (-JPN,Alnum)」を参照のこと。

-JPN オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。

22.8.3 -TruncCheck オプション

規則

動的長基本項目は-TruncCheck オプションのチェック対象とならない。

-TruncCheck オプションについては、「24.7 転記の長さのチェック (-TruncCheck)」を参照のこと。形式については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。

22.9 使用例

22.9.1 COBOL と C 言語の連携

(1) C プログラムから COBOL プログラムへ渡す場合

C プログラム側

```
int main(){
    char DATABUF[50];
    strncpy(DATABUF, "abcd", 5);
    SampleTP(DATABUF);
    :
```

COBOL プログラム側

```
IDENTIFICATION DIVISION .
PROGRAM-ID. 'SampleTP'.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
DYNAMIC LENGTH STRUCTURE C-STRING IS C-STATIC-STRUCTURE .
DATA DIVISION.
LINKAGE SECTION.
    01 DYNADATA PIC X DYNAMIC LENGTH C-STRING .
PROCEDURE DIVISION USING DYNADATA .
    DISPLAY DYNADATA. *> 'abcd' が表示される
```

注意事項

- DYNADATA を COBOL プログラム側で更新してはならない。
- DYNADATA の定義に LIMIT 指定を行う場合は、そのサイズは C プログラム側で確保された領域のサイズ-1 とすること（上記の例では、DATABUF のサイズ-1 である 49 とする）。

(2) COBOL プログラムから C プログラムへ渡す場合

COBOL プログラム側

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
DYNAMIC LENGTH STRUCTURE C-STRING IS C-STATIC-STRUCTURE .
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 DYNADATA PIC X DYNAMIC LENGTH C-STRING LIMIT 20 .
PROCEDURE DIVISION .
    MOVE 'abcde' TO DYNADATA. *> DYNADATAには'abcde'+終端文字(X'00')が格納される
    CALL 'csub' USING DYNADATA.
```

C プログラム側

```
int csub(char *data1){
    /* data1はCOBOL側のDYNAMDATAの領域を指す。 */
    /* DYNAMDATA の実領域は21バイトであるが、末尾は終端文字であるため*/
    /* data1に対する操作は20バイト以内で行うこと */
    :
```

22.9.2 COBOL プログラム間の連携

呼び出し元プログラム

```
IDENTIFICATION DIVISION .
PROGRAM-ID. MAINPROG.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
DYNAMIC LENGTH STRUCTURE C-STRING IS C-STATIC-STRUCTURE .
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 DYNADATA PIC X DYNAMIC LENGTH C-STRING LIMIT 30 .
PROCEDURE DIVISION .
    MOVE 'abcde' TO DYNADATA. *> DYNADATAには'abcde'+終端文字(X'00')が格納される
    CALL 'SUBPROG' USING DYNADATA.
```

呼び出し先プログラム

```
IDENTIFICATION DIVISION .
PROGRAM-ID. SUBPROG.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
DYNAMIC LENGTH STRUCTURE C-STRING IS C-STATIC-STRUCTURE .
DATA DIVISION.
LINKAGE SECTION.
    01 DYNADATA2 PIC X DYNAMIC LENGTH C-STRING LIMIT 30 .
PROCEDURE DIVISION USING DYNADATA2 .
    DISPLAY DYNADATA2. *> 'abcde'を表示する
```

注意事項

呼び出し元プログラムと呼び出し先プログラムで、LIMIT の値を一致させること。

22.9.3 COBOL のプログラム内で使用する例

```
IDENTIFICATION DIVISION.
PROGRAM-ID. TESTPROG.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
DYNAMIC LENGTH STRUCTURE C-STRING IS C-STATIC-STRUCTURE.
DATA DIVISION.
```

```

WORKING-STORAGE SECTION.
  01 FIXDATA PIC X(20).
  01 DYNADATA PIC X DYNAMIC LENGTH C-STRING LIMIT 20.
  01 DYNADATA-N PIC N DYNAMIC LENGTH C-STRING LIMIT 5.
PROCEDURE DIVISION .
  MOVE 'error 01' TO DYNADATA. *> 空白詰めは行わない
  DISPLAY DYNADATA. *> 長さ8のメッセージを表示する。

  MOVE 'err 02' TO DYNADATA.
  DISPLAY DYNADATA. *> 長さ6のメッセージを表示する。DYNADATAのサイズが変わる

  INITIALIZE DYNADATA. *> INITIALIZE文を実行すると動的長基本項目の長さは0になる

  MOVE 'error 0003' TO DYNADATA.
  DISPLAY DYNADATA. *> 長さ10のメッセージを表示する

  MOVE DYNADATA TO FIXDATA. *> 動的長基本項目が送り出し側となる例
  *> DYNADATAは長さ10であるため、長さ10の固定項目を転記する場合と同等

  MOVE 'error 0003' TO FIXDATA .
  DISPLAY FIXDATA. *> 長さ10+空白10のメッセージを表示する

  IF DYNADATA = FIXDATA THEN *> DYNADATAは'error 0003'を格納した長さ10の固定項目と
    DISPLAY 'MATCH' *> 同等のため、'MATCH'となる
  END-IF.

  MOVE FIXDATA TO DYNADATA. *> 埋字の空白も転記され、DYNADATAは長さ20となる
  MOVE FUNCTION TRIM(FIXDATA) TO DYNADATA. *> 空白を取る場合はTRIM関数を併用する

  MOVE '12345678901234567890ABC' TO DYNADATA. *> LIMIT 20のため送り出し側の
  *> 先頭20バイトしか格納されない

  MOVE N'あいう' TO DYNADATA-N. *> 空白詰めは行わない
  MOVE N'あいうえおかきく' TO DYNADATA-N. *> N'あいうえお'が格納される
  INITIALIZE DYNADATA-N. *> 初期化され、長さは0になる

```

22.10 動的長基本項目を使用したときの注意事項

動的長基本項目を使用する場合、手続き文での参照時および更新時に終端文字までの長さを求めるため、英数字項目または日本語項目を使用した場合と比べると、プログラム実行時に時間が掛かる。

23

Unicode 機能

Unicode 機能は、シフト JIS で記述された原始プログラムから、Unicode のデータを入出力および操作できる COBOL プログラムを生成する機能である。

この章では、Unicode 機能を使用するための組み込み関数について説明する。

23.1 用語の定義

Unicode 機能で使用する用語について次に示す。

#	用語	説明
1	Unicode	1 バイトでは表現できない文字セットを含むあらゆる文字セットをサポートした文字コードのこと。代表的な符号化文字集合として、UCS-2、UCS-4 がある。代表的なエンコーディングスキーマとして UTF-8、UTF-16 がある。 COBOL2002 では、UCS-4 の範囲 (UCS-2 の範囲を含む) をサポートする。
2	UCS-2	符号化文字集合の一つの形式。1 文字を 2 バイトで表現する。COBOL2002 では、用途が DISPLAY および NATIONAL の項目でサポートする。
3	UCS-4	符号化文字集合の一つの形式。1 文字を 4 バイトで表現する。 COBOL2002 では、用途が DISPLAY および NATIONAL の項目で UCS-4 の範囲をサポートする。
4	UTF-8	エンコーディングスキーマの一つの形式。ASCII 文字を 1 バイト、日本語文字を 3~8 バイト、半角かたかなを 3 バイトで表現する。
5	UTF-16	エンコーディングスキーマの一つの形式。1 つのコード単位を 2 バイトとし、1 文字を 1 コード単位 (2 バイト)、または 2 コード単位 (4 バイト) で表現する。UTF-16 では 2 バイトのコード単位の 1 バイト目を先に書くビッグエンディアン形式 (UTF-16BE) と 1 バイト目を後に書くリトルエンディアン形式 (UTF-16LE) がある。 COBOL2002 では、用途が NATIONAL の項目で UCS-4 の範囲 (UCS-2 の範囲を含む) をサポートする。
6	サロゲート	UTF-16 の拡張で、2 つのコード単位 (4 バイト) で 1 文字を表す機能。この 2 つのコード単位の組み合わせのことをサロゲートペアと呼ぶ。 COBOL2002 では、用途が NATIONAL の項目で使用する。
7	BOM (バイトオーダーマーク)	ファイルの先頭に付加された、Unicode の表現形式を表す情報。COBOL2002 では、テキスト編成ファイルに対してこの情報を付加する。本文中では、Unicode シグニチャと表記する。
8	バイトオーダー	2 バイト以上のデータの記録を行なう順序のこと。例えば、0x1234 のデータを 0x1234 のように最上位のバイトから順番に記録する方式をビッグエンディアン、0x3412 のように最下位のバイトから順番に記録する方式をリトルエンディアンという。2 バイトの UTF-16 は、バイトオーダーを意識する。
9	IVS	漢字を表す Unicode の直後に Variation Selector と呼ばれるコードを付加し、漢字の「異体字」を表現する方法のこと。 COBOL2002 では、用途が DISPLAY および NATIONAL の項目で使用する。
10	文字位置の個数	英数字項目または日本語項目の PICTURE 句に指定した文字位置の個数。 例) <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"><pre>01 DATA1 PIC X(3). *>文字位置の個数は3 01 DATA2 PIC N(5). *>文字位置の個数は5</pre></div>
11	文字数	見た目の文字数。 サロゲートペアや IVS で表現する漢字も文字数は 1 文字とする。

#	用語	説明
12	見た目幅	Unicode 機能の組み込み関数で使用する，文字の見た目の幅。半角文字の幅は 1，全角文字の幅は 2 として扱う。 Unicode 機能の組み込み関数で，文字を見た目幅で数える場合に使用する。

23.2 組み込み関数

COBOL プログラムで Unicode 文字を扱う機能としてサポートする組み込み関数については、「[25.5 組み込み関数](#)」を参照のこと。

23.2.1 COUNT-CHAR 関数

COUNT-CHAR 関数については、「[25.5.3 COUNT-CHAR 関数](#)」を参照のこと。

23.2.2 DISPLAY-OF 関数

DISPLAY-OF 関数については、「[25.5.4 DISPLAY-OF 関数](#)」を参照のこと。

23.2.3 LENGTH-OF-SUBSTRING 関数

LENGTH-OF-SUBSTRING 関数については、「[25.5.5 LENGTH-OF-SUBSTRING 関数](#)」を参照のこと。

23.2.4 NATIONAL-OF 関数

NATIONAL-OF 関数については、「[25.5.6 NATIONAL-OF 関数](#)」を参照のこと。

23.2.5 SUBSTRING 関数

SUBSTRING 関数については、「[25.5.7 SUBSTRING 関数](#)」を参照のこと。

24

コンパイラオプションによる仕様の相違

この章では、コンパイラオプションの指定によって言語仕様が異なるものについて説明する。

24.1 言語仕様が異なるコンパイラオプション

コンパイラオプションの指定によって言語仕様が異なるものを次に示す。

表 24-1 仕様変更コンパイラオプション

オプション	意味
指定なし	第 3, 4 次規格で翻訳する
-StdVersion,1	第 1 次規格で翻訳する
-StdVersion,2	第 2 次規格で翻訳する
-DigitsTrunc	2 進項目のけた落とし
-DoubleQuote	アポストロフィか引用符かの指定
-V3Rec,Fixed	VOS3 の日本語文字の扱いに合わせて翻訳する (固定長レコードの原始プログラム)
-V3Rec,Variable	VOS3 の日本語文字の扱いに合わせて翻訳する (可変長レコードの原始プログラム)
-V3RecFCSpace	空白に対する機能キャラクタの扱い
-V3RecEased,QuoteCheck*	-V3Rec オプションのチェックを緩和する (定数の分離符のチェック)
-V3RecEased,WordCheck*	-V3Rec オプションのチェックを緩和する (語または PICTURE 文字列の制限値のチェック)
-IgnoreLCC*	レコード先頭の印刷制御文字領域
-JPN,Alnum	日本語項目の扱い
-CompatV3	VOS3 COBOL85 互換を指定する
-VOSCBL*	メインフレーム互換機能を有効にする
-H8Switch	数字項目の転記/比較
-Cblctr	CBL-CTR 特殊レジスタ
-DebugLine	デバッグ行を有効にする
-Main,System	コマンドインタフェースを C 言語の main 関数の形式 (argc, argv 形式) に合わせる
-TruncCheck	転記のけたの切り捨てチェック
-JPN,V3JPN*	日本語項目, 日本語編集項目, および日本語文字定数をそれぞれ英数字項目, 英数字編集項目, および英数字定数として扱う
-JPN,V3JPNSpace	VOS3 COBOL85 の LANGOPT=(D)オプション指定時の日本語項目の扱いと同じとする
-LiteralExtend,Alnum*	英数字定数の最大長を拡張する
-IgnoreAPPLY,FILESHARE*	APPLY FILE-SHARE 句を覚え書きとする
-SpaceAsZero*	外部 10 進項目に空白文字があるとき, ゼロとみなして比較, 演算, 転記を実行する

注※

詳細については、マニュアル「COBOL2002 ユーザーズガイド」または「COBOL2002 使用の手引 手引編」を参照のこと。

24.2 旧規格仕様を使う場合

COBOL は、第 1 次規格 (ANSI68, ISO72, JIS72), 第 2 次規格 (ANSI74, ISO78, JIS80), 第 3 次規格 (ISO85, ANSI85, JIS88, JIS92), 第 4 次規格 (ISO/IEC 1989:2002, JIS X3002:2011) と変わってきており、同じ書き方でも異なる言語仕様となっているものがある。どの段階の規格仕様で翻訳するのかは、`-StdVersion,1` または `-StdVersion,2` オプションで指定する。

マニュアル「COBOL2002 言語 標準仕様編」では、第 3 次規格と第 4 次規格の仕様について記述している (`-StdVersion,1`, `-StdVersion,2` オプション指定がない場合)。ここでは、`-StdVersion,1`, `-StdVersion,2` オプションを指定したときの仕様について説明する。

24.2.1 第 1 次規格以前の仕様 (-StdVersion,1)

`-StdVersion,1` オプションは、第 1 次規格以前の仕様で翻訳することを指定する。第 2 次規格以降との間で解釈が異なっている言語仕様について次に示す。

(1) 特殊名段落の CURRENCY SIGN 句

CURRENCY SIGN 句に、`[=]`, `[/]` を指定できる。

(2) SELECT OPTIONAL 句

このシステムでは、OPTIONAL 句は覚え書きとする。

(3) 略記組み合わせ比較条件での NOT の取り扱い

略記組み合わせ比較条件の中の NOT が、比較演算子の一部とも論理演算子ともとれるとき、すなわち、NOT の直後が、次のどれかのときは、この NOT を論理演算子とみなす。

- GREATER
- LESS
- EQUAL
- >
- <
- =
- >=
- <=

(例)

```
A > B AND NOT > C OR D
```

は、次と同じである。

A > B AND NOT A > C OR A > D

-StdVersion,1 オプションが有効でない場合は、次と同じである。

A > B AND A NOT > C OR A NOT > D

(4) MOVE 文の位取り

送り出し側作用対象が位取りした整数項目 (PICTURE 文字列の右端文字が P)、受け取り側作用対象が英数字項目であれば、文字 P にあたるけたを空白とする。また、比較条件の作用対象が位取りした整数項目と英数字項目であれば、整数項目を英数字項目に変換するときに MOVE 文と同様文字 P にあたるけたを空白とみなす。

(例)

```
77 A PIC 999PP VALUE 12300.  
77 B PIC X(5).  
MOVE A TO B.
```

この例で、MOVE 文を実行すると、データ項目 B は、123△△となる (△△は半角空白)。

(注意)

-StdVersion,1 オプションが有効でない場合には、データ項目 B は、12300 となる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

(5) UNSTRING 文の DELIMITED BY ALL 指定

UNSTRING 文に DELIMITED BY ALL 指定を書くと、送り出し側作用対象の文字列中に同じ区切り文字の連続した繰り返しが現れた場合、DELIMITER IN 指定のデータ項目に、区切り文字の繰り返し現れた文すべてが転記される (入りきらなければ右側が無視される)。

(注意)

-StdVersion,1 オプションが有効でない場合には、区切り文字の受け取り側作用対象に区切り文字 1 個分だけが転記される。

(備考)

第 1 次規格には UNSTRING 文はない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.52 UNSTRING 文」

(6) COPY 文 (コンパイラオプションによる仕様の相違)

形式

$$\text{COPY 原文名1} \left[\left\{ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right\} \text{登録集名1} \right]$$
$$\left[\text{REPLACING} \left\{ \text{語1 BY} \left\{ \begin{array}{l} \text{語2} \\ \text{定数1} \\ \text{一意名1} \end{array} \right\} \dots \right\} \right]$$

構文規則

1. REPLACING 指定の中に仮原文 1 や仮原文 2 は書けない。また、予約語 BY の前には複数個の語の組から成る一意名および定数を書けないので、単一の COBOL の語を書かなければならない。
2. COPY 文を書ける個所に制限がある。次の個所にだけ COPY 文を書くことができる。

- 環境部

SOURCE-COMPUTER. COPY 文.

OBJECT-COMPUTER. COPY 文.

SPECIAL-NAMES. COPY 文.

FILE-CONTROL. COPY 文.

SELECT [OPTIONAL]

ファイル名 COPY 文.

SELECT 整列併合用ファイル名 COPY 文.

I-O-CONTROL. COPY 文.

- ファイル節

FD ファイル名 COPY 文.

SD 整列併合用ファイル名 COPY 文.

nn データ名 COPY 文.

(nn は 01 または 02~49 のレベル番号)

- 作業場所節および連絡節

$$\text{nn} \left\{ \begin{array}{l} \text{データ名1} \\ \text{FILLER} \end{array} \right\} [\text{REDEFINES} \text{データ名2}] \text{COPY 文.}$$

(nnは01, 02~49または77のレベル番号)

- 手続き部

節名 SECTION [区分番号] . COPY 文.

段落名. COPY 文.

一般規則

1. データ記述項に COPY 文を書いたとき、複写される登録集原文は、その COPY 文が指定された記述項のレベル番号と同じレベル番号を持つ記述項から始まるのでなければならない。このとき、COPY 文は記述項全体を登録集原文で置き換えるが、語 COPY の前に書いたレベル番号、データ名および REDEFINES 句は置き換わず、原始プログラム中に書いたものがそのまま残る。

(例 1)

01 RECORD-A COPY RECALIB.

原始文 RECALIB が識別する登録集原文は、完全なレコード記述でなければならない。次に正しい記述と誤った記述の例を示す。

(正しい記述例)

```
01 RECORD-B.  
02 RECA1 PIC X(3).  
02 RECA2.  
03 RECA2-1 PIC S9(5).  
03 RECA2-2 PIC S9(9) COMP.
```

(誤った記述例)

```
SYNCHRONIZED.  
02 RECA1 PIC X(3).  
02 RECA2.  
03 RECA2-1 PIC S9(5).  
03 RECA2-2 PIC S9(9) COMP.
```

複写の結果は次のようになる。登録集原文のうち、01 RECORD-B.だけは複写後、原始プログラムに書いた 01 RECORD-A.で置き換えられる。

```
01 RECORD-A.  
02 RECA1 PIC X(3).  
02 RECA2.  
03 RECA2-1 PIC S9(5).  
03 RECA2-2 PIC S9(9) COMP.
```

(例 2)

05 RECORD-C COPY RECBLIB.

原文名 RECBLIB の登録集原文は完全なレコード記述でなければならない。次に例を示す。

```
05 RECORD-D.  
10 RECD1 PIC X(3).
```

複写の結果は次のようになる。

```
05 RECORD-C.  
10 RECD1 PIC X(3).
```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「3.2.2 COPY 文」

(7) JUSTIFIED 句の処理

JUSTIFIED 句の指定がある英字、英数字、英数字編集に VALUE 句で初期値を与えるときに、JUSTIFIED 句の機能を働かせる。

(例)

```
05 DATA-A PIC X(5) JUSTIFIED VALUE 'AB'.
```

DATA-A には、値 AB が右端によせて格納される。左端に空白が埋められる。

(注意)

-StdVersion,1 オプションが有効でない場合には、JUSTIFIED 句の機能は無視される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.36 JUSTIFIED 句」

(8) ADVANCING 指定なしの WRITE 文での仮定

一つのファイルに対する複数個の WRITE 文で、ADVANCING 指定があるものとないものを混用した場合、次のように仮定する。

ADVANCING 指定のある WRITE 文	ADVANCING 指定のない WRITE 文
すべて BEFORE のとき	BEFORE 1 を仮定する
すべて AFTER のとき	AFTER 1 を仮定する
BEFORE と AFTER とがあるとき	BEFORE 1 を仮定する

(注意)

-StdVersion,1 オプションが有効でない場合には、AFTER ADVANCING 1 を仮定する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

24.2.2 第 2 次規格以前の仕様 (-StdVersion,1, -StdVersion,2)

-StdVersion,1 オプションは第 1 次規格以前、-StdVersion,2 オプションは第 2 次規格以前の仕様で翻訳することを指定するが、ここでは第 1 次規格と第 2 次規格の仕様が同じで第 3 次規格と第 4 次規格との間で解釈が異なっている言語仕様について次に示す。

(1) VARYING と AFTER の指定がある PERFORM 文での評価順序

-StdVersion オプションを指定した場合、次の書き方のとき、標準仕様の PERFORM 文の規則と異なる。

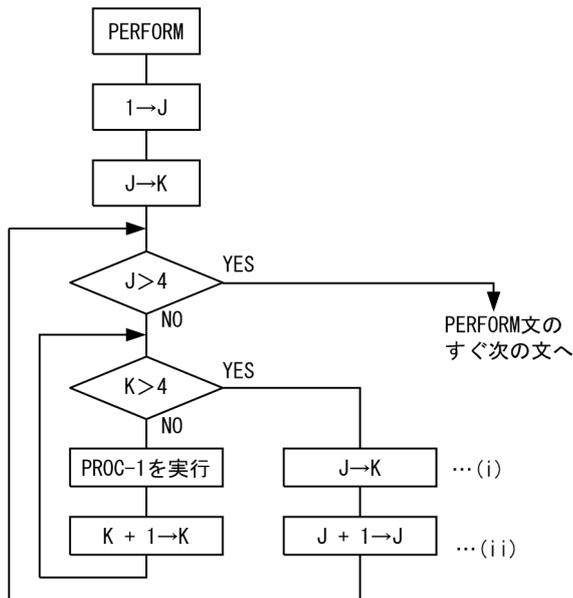

```

01 J      PIC 99 COMP.
01 K      PIC 99 COMP.
01 WK-1.
  02 WK-2      OCCURS 4.
  03 WK-3 PIC X OCCURS 4.
  :
  MOVE SPACE TO WK-1.
  PERFORM PROC-1
    VARYING J FROM 1 BY 1 UNTIL J > 4
    AFTER K FROM J BY 1 UNTIL K > 4,
  :
PROC-1.
  MOVE '*' TO WK-3(J,K)

```

上記の（例 1）の PERFORM 文の流れを次に示す。

図 24-1 PERFORM 文の流れ



-StdVersion オプションを指定して、PERFORM 文を実行すると、「*」の転記は次のとおりに行われる。

WK-2(1) →	*	*	*	*
WK-2(2) →	*	*	*	*
WK-2(3) →		*	*	*
WK-2(4) →			*	*

この例のループ制御変数（J, K）は、次のように変化する。

J	1	1	1	①	2	2	2	②	3	3	③	4	4
K	1	2	3	4	1	2	3	4	2	3	4	3	4

同じ例を、-StdVersion オプションを指定しないで実行すると、「図 24-1 PERFORM 文の流れ」中の (i) と (ii) の処理の順番が入れ替わる。このとき、PERFORM 文の実行の結果、「*」の転記は次のとおりに行われる。

WK-2(1) →	*	*	*	*
WK-2(2) →		*	*	*
WK-2(3) →			*	*
WK-2(4) →				*

ループ制御変数 (J, K) は, 次のように変化する。

J	1	1	1	1	② 2	2	③ 3	④
K	1	2	3	4	2	3	4	4

-StdVersion オプション未指定時の仕様は, 次を示す例 2 のうち PERFORM 文と同じである。

(例 2)

```

PERFORM VARYING J FROM 1 BY 1 UNTIL J > 4
  PERFORM VARYING K FROM J BY 1 UNTIL K > 4
  MOVE '*' TO WK-3(J,K)
END-PERFORM
END-PERFORM.

```

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.31 PERFORM 文」

24.3 VOS3 COBOL85 からの移行用オプション

言語仕様の意味を変更するオプションについて説明する。

なお、報告書作成機能については、「7.1 報告書作成機能の VOS3 COBOL85 からの移行用オプション (-CompatiV3)」を参照のこと。

24.3.1 2進項目のけた落とし (-DigitsTrunc)

-DigitsTrunc を指定すると、COBOL 原始プログラム中の転記文で受け取り側作用対象が 2 進項目の場合、送り出し側作用対象の有効けた (PICTURE 句で指定した 10 進けた数) が受け取り側作用対象のけた数より大きいとき、上位けたの切り捨てを行う。

-DigitsTrunc を指定しないと、COBOL 原始プログラム中の転記文で受け取り側作用対象が 2 進項目の場合、送り出し側作用対象の値の有効けたが受け取り側作用対象の領域 (2, 4 または 8 バイト) に入らないとき、そのバイト数に入りきらない部分を切り捨てる。

(備考)

COBOL は 10 進法を基調にしているので、厳密には -DigitsTrunc が規格の仕様である。

(例)

```
77 A PIC S9(5) COMP VALUE 12345.  
77 B PIC S9(4) COMP.  
:  
MOVE A TO B
```

A の 2 進項目には 2 バイトの領域が割り当てられるので、-32768~+32767 の範囲の数値を格納できる。しかし、-DigitsTrunc 指定時には、けた落としが行われ、転記後は B の項目の値は 2345 となる。なお、例外名 EC-SIZE-TRUNCATION に対する TURN 指令が有効で、送り出し側の有効けた数が受け取り側のけた数より大きいときは、このオプションの指定があっても、上位けたを切り捨てないで、EC-SIZE-TRUNCATION 例外が成立する。

24.3.2 分離符の選択 (-DoubleQuote)

-DoubleQuote を指定すると、表意定数 QUOTE の値や英数字定数の範囲を示す分離符として引用符 (") が用いられる。

-DoubleQuote を指定しないと、表意定数 QUOTE の値や英数字定数の範囲を示す引用符としてアポストロフィ (') が用いられる。

(備考)

規格仕様の分離符は、引用符 (") であるが、日立や他社の多くのコンパイラでは以前からアポストロフィ (') が使用されている。

24.3.3 日本語文字の扱い (-V3Rec,Fixed, -V3Rec,Variable)

(1) 機能

-V3Rec,Fixed, -V3Rec,Variable は、利用者がメインフレーム (VOS3) からファイル転送された、メインフレームでの固定長レコード形式または可変長レコード形式の COBOL 原始プログラムを、メインフレームの日本語の扱いに合わせて翻訳することを可能とするためにサポートした機能である。すなわち、この機能を用いると、日本語混在の COBOL 原始プログラムを、機能キャラクタが付加されたものとみなした、原始プログラムの領域の意識および語の構成規則のエラーチェックを行って翻訳することができる。また、このシステムでの標準コードと拡張コードの等価規則を無効とした形での翻訳を可能とする。

ここでは、日本語混在の COBOL 原始プログラムを、メインフレームの日本語の扱いに合わせて翻訳する上で、標準仕様と異なる言語仕様について述べる。

(2) 文字集合

日本語文字を用いる場合、日本語文字列の直前と直後に、メインフレームでの「KEIS コード開始」と「EBCDIK コード開始」を表す 2 バイトの機能キャラクタ (シフトコード) が、それぞれ付加されたものとみなす。ここでは、以降、これらの機能キャラクタをそれぞれ、[漢]と[E]で表す。

(a) [漢], [E]を仮定する条件

-V3Rec オプションの指定が有効な場合、このシステムで[漢], [E]が仮定される条件を次に示す。

- [漢]が仮定される条件
空白を除く 1 件または連続する拡張コードの直前に[漢]が仮定される。
- [E]が仮定される条件
直前に[漢]を仮定された拡張コードの文字列の後ろで、最初に出現した空白を除く標準コードの直前に[E]が仮定される。

(b) 空白に関する機能キャラクタの扱いによるメインフレームとの差異

このシステムは、空白に関する機能キャラクタの扱いで、メインフレームと以下の差異がある。

- 拡張コードの空白を標準コードの空白 2 個と等価とみなし、日本語文字列の開始文字として扱わない。このため、拡張コードの空白の直前には[漢]が仮定されない。
- 日本語文字列に続く標準コードの空白は日本語文字列として扱う。このため、日本語文字列と改行の間が 0 個以上の半角空白だけのとき、[E]が仮定されない。

空白に関する機能キャラクタの扱いをメインフレームと同等にしたい場合、-V3Rec オプションと合わせて -V3RecFCSpace オプションを指定する。-V3RecFCSpace オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の -V3RecFCSpace オプションを参照のこと。

(例)

日本語文字が含まれた語に対して、機能キャラクタが仮定される例を以下に示す。

-V3Rec オプション指定がない場合

```
01 得意先.  
02 データ PIC N(2) VALUE N'日立'.  
02 空白 PIC N(2) VALUE N'▲▲'.
```

-V3Rec オプション指定だけがある場合

```
01 [漢]得意先[E].  
02 [漢]データ [E]PIC N(2) VALUE N'[漢]日立[E]'.  
02 [漢]空白 [E]PIC N(2) VALUE N'[漢]▲▲[E]'.
```

-V3Rec オプションと合わせて-V3RecFCSpace オプションを指定した場合

```
01 [漢]得意先[E].  
02 [漢]データ [E]PIC N(2) VALUE N'[漢]日立[E]'.  
02 [漢]空白 [E]PIC N(2) VALUE N'[漢]▲▲[E]'.
```

(凡例) ▲：日本語空白

(3) 文字列

利用者定義語、定数および注釈に日本語文字を使用できる。

(a) COBOL の語

COBOL の語とは、31 字以下の文字列であって、利用者定義語、システム名、予約語および関数名のどれかとする。

この機能では、標準コードの文字に換算して 31 字以下の文字列とする。

各英小文字は、その対応する英大文字に等価とみなす。ただし、空白を除く、拡張コードの各文字は、その対応する標準コードの各文字に等価とはみなさない。

拡張コードの空白は、標準コードの 2 字の空白と等価とみなす。

- 利用者定義語

日本語（拡張コード）文字を用いて、利用者が定義できる語には、次のものがある。

- 符号系名 (alphabet-name)
- データ名 (data-name)
- レコード名 (record-name)
- 条件名 (condition-name)
- ファイル名 (file-name)
- 指標名 (index-name)

- 呼び名 (mnemonic-name)
- 記号文字 (symbolic-character)
- 字類名 (class-name)
- 段落名 (paragraph-name)
- 節名 (section-name)

各利用者定義語の語を構成する文字列のサイズは、機能キャラクタを含めて、31 バイト以下でなければならない。したがって、日本語文字だけで語を構成する場合は、拡張コードの文字に換算して、13 字以下でなければならない。

(b) 定数

72 カラム以前に改行文字がある場合、改行文字から 72 カラムに空白を仮定する。

- 英数字定数

英数字定数を構成する文字列のサイズは、機能キャラクタを含めて、160 バイト以下でなければならない。

構文規則

1. 文字列が引用符を表すならば、引用符の 1 字を表すのに、標準コードの連続する 2 字の引用符を用いなければならない。
- 日本語文字定数
日本語文字定数を構成する文字列のサイズは、機能キャラクタを含めて、160 バイト以下でなければならない。したがって、日本語文字だけで語を構成する場合は、78 字以下でなければならない。

形式

```
N' {文字 1} ...'  
NC' {文字 1} ...'  
ND' {文字 1} ...'
```

構文規則

- 文字 1 は、拡張コードの任意の文字および標準コードの空白とする。
- 拡張コードの引用符は、分離符の引用符とはみなさない。したがって、拡張コードの引用符は、文字 1 内の任意の位置に書くことができる。

(4) コーディング規則

(a) 行と文字位置

原始プログラムは英数字文字を 1 文字 (1 バイト) 単位、日本語文字を 2 バイト単位として 80 文字で構成される行中に記述する。

メインフレームの固定長レコード形式によって翻訳する場合、日本語文字定数など前後に付ける機能キャラクタは、プログラム原文領域（8 カラムから 72 カラム）中になければならない。

メインフレームの可変長レコード形式によって翻訳する場合、日本語文字列の前後に付ける機能キャラクタを除くと、原始文記述はプログラム原文領域中に納まっていなければならない。

(b) 日本語文字の継続

日本語文字は、複数行にわたって定義できる。この場合、継続行はプログラム原文領域（8 カラムから 72 カラム）で指定しなければならない。ただし、日本語文字の継続に関しては、機能キャラクタの扱いがメインフレームと異なるため、メインフレームとの完全な互換性はない。

(5) 原始文操作機能の特性

この機能では、原始文操作機能である COPY 文の REPLACING 指定および REPLACE 文によって語の置き換えがあり、置き換えられることによって、本来境界 R までに納まっていた原文語が機能キャラクタを含めた形で論理的に境界 R を超える場合でも、見かけ上の境界 R までに納まる場合は、そのまま置かれる。同様に COPY 文の PREFIXING/SUFFIXING 指定によって語が付加された結果として、機能キャラクタを含めて、論理的に境界 R を超える場合でも、見かけ上の境界 R までに納まる場合は、そのまま置かれる。また、日本語文字の置き換えが発生する COPY 文および REPLACE 文に関しては、機能キャラクタの扱いがメインフレームと異なるため、メインフレームとの完全な互換性はない。

また、COPY 文の PREFIXING/SUFFIXING 指定で指定する語 1 および語 2 に日本語文字を含めてはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「3.2.2 COPY 文」

「COBOL2002 言語 標準仕様編」 「3.2.3 REPLACE 文」

24.3.4 行送り制御文字領域削除 (-IgnoreLCC)

WRITE~ADVANCING 文で書き出すレコードの先頭 1 バイトを出力しないことを指定する。あらかじめ、レコード記述項で行送り制御文字用の 1 バイト分の領域を確保しておいたプログラムをコンパイルするときに指定する。ただし、次のどれかが当てはまる場合、-IgnoreLCC オプションの指定があってもレコードの先頭から出力する。

- 順ファイル以外のファイルの場合
- 順ファイルの WRITE 文に行送りの指定（ADVANCING, POSITIONING）がない場合
- 順ファイルのファイル記述項に LINAGE 指定がある場合

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

24.3.5 日本語項目に対して埋められる空白の扱い (-JPN,V3JPNSpace)

-JPN,V3JPNSpace オプションと-CompatiV3 オプションを同時に指定したとき、日本語集団項目、日本語項目、日本語編集項目、および日本語文字定数をそのままの属性で扱う。これは、メインフレーム (VOS3) COBOL85 で、LANGOPT=(D)オプションと XCOBOL=(-N)オプションを同時に指定したときの日本語項目の扱いに合わせるものである。

なお、-JPN,V3JPNSpace オプションは、VOS3 COBOL85 互換機能である。-JPN,V3JPNSpace オプションについては、マニュアル「COBOL2002 ユーザーズガイド」または「COBOL2002 使用の手引 手引編」を参照のこと。

-JPN,V3JPNSpace オプションを指定してコンパイルすると、日本語集団項目、日本語項目、および日本語編集項目に対して次の扱いとする。

(1) 表意定数 SPACE (SPACES) の扱い

日本語集団項目、日本語項目、および日本語編集項目に対する表意定数 SPACE (SPACES) は、日本語空白 (X'8140') とする。

(例 1)

```
01 A1 PIC N(2) VALUE SPACE.
```

上記の VALUE 句によるデータ名 A1 の内容は、-JPN,V3JPNSpace オプション指定の有無により、次のとおりとなる。

-JPN,V3JPNSpace オプション指定あり	-JPN,V3JPNSpace オプション指定なし
▲▲	△△△△

(凡例)

▲：全角空白 (X'8140')

△：半角空白 (X'20')

(2) 日本語集団項目、日本語項目、および日本語編集項目同士の転記の扱い

日本語集団項目、日本語項目、および日本語編集項目同士の転記で、受け取り側作用対象のけた数が送り出し側作用対象のけた数より長い場合、受け取り側作用対象の右側に日本語空白 (X'8140') を補う。

(例 1)

```
01 A1 PIC N(2) VALUE N'日立'.
01 B1 PIC N(5).

MOVE A1 TO B1.
```

上記の MOVE 文を実行したあとのデータ名 B1 の結果は、-JPN,V3JPNSpace オプション指定の有無により、次のとおりとなる。

-JPN,V3JPNSpace オプション指定あり	-JPN,V3JPNSpace オプション指定なし
日立▲▲▲▲	日立△△△△△△

(凡例)

▲：全角空白 (X'8140')

△：半角空白 (X'20')

(3) 日本語集団項目，日本語項目，および日本語編集項目同士の比較の扱い

日本語集団項目，日本語項目，および日本語編集項目同士の比較で，作用対象のけた数が等しくない場合，短い方の作用対象の右側に，長い方の作用対象のけた数に等しくなるまで日本語空白 (X'8140') があるものとみなして比較する。

(例 1)

```

01 A1  PIC N(2) VALUE N' 株式' .
      :
      :
      IF A1 = N' 株式会社'
      THEN
          DISPLAY N' 株式会社です'
      ELSE
          DISPLAY N' 株式会社ではありません'
      END-IF.

```

上記の IF 文を実行するときデータ名 A1 の内容は，-JPN,V3JPNSpace オプション指定の有無により，次のとおりとなる。

-JPN,V3JPNSpace オプション指定あり	-JPN,V3JPNSpace オプション指定なし
株式▲▲	株式△△△△

(凡例)

▲：全角空白 (X'8140')

△：半角空白 (X'20')

なお，-JPN,V3JPNSpace オプション指定による日本語項目の扱いは，LANGOPT=(D)オプション指定時のすべての仕様と同じにするものではない。VOS3 COBOL85 で LANGOPT=(D)オプションを指定した場合と，COBOL2002 で-JPN,V3JPNSpace オプションを指定した場合の仕様の相違点については，「付録 B LANGOPT=(D)オプションと-JPN,V3JPNSpace オプションを指定した場合の仕様の相違」を参照のこと。

24.3.6 日本語項目の扱い (-JPN,Alnum)

-JPN,Alnum オプションは，日本語項目，日本語編集項目を英数字項目，英数字編集項目として扱う。例えば，PIC N(5)は，PIC X(10)とみなされる。-JPN,Alnum オプションの指定によって日本語の扱いが異なる例を次に示す。

(例 1) 日本語項目と英数字項目間の転記の相違

```
MOVE 英数字項目 TO 日本語項目.
```

- -JPN,Alnum オプション指定時：
英数字項目間の転記の規則に従って転記される。受け取り側作用対象の方が長い場合は、転記が終わった位置より右側の文字位置に半角の空白 (X'20') が埋められる。
- -JPN,Alnum オプション未指定時：
日本語項目同士の転記でないためコンパイルエラーとなる。

(例 2) 空白の転記

```
01 日本語項目 PIC N(5).  
MOVE SPACE TO 日本語項目.
```

- -JPN,Alnum オプション指定時：
半角の空白 (X'20') が転記される。
- -JPN,Alnum オプション未指定時：
全角の空白 (X'8140') が転記される。

(例 3) 日本語項目に対する VALUE 句

```
01 日本語項目 PIC N(2) VALUE '漢字'.  
01 日本語項目 PIC N(2) VALUE ZERO.
```

- -JPN,Alnum オプション指定時：
日本語直接定数、英数字定数、表意定数のどれを指定してもエラーにならない。なお、英数字定数、表意定数 (SPACE(S)以外) はメインフレーム (VOS3) の COBOL では指定できないため、日本語直接定数と表意定数の SPACE(S)だけを使用した方が良い。
- -JPN,Alnum オプション未指定時：
日本語直接定数または表意定数の SPACE(S)でなければならない。したがって、(例 3) はコンパイルエラーとなる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

(例 4) 部分参照での相違

```
01 NIHONGO PIC N(10) VALUE N' 1 2 3 4 5 6 7 8 9 0'.
```

- -JPN,Alnum オプション指定時：
NIHONGO(3:4)は、N' 2 3'となる。
- -JPN,Alnum オプション未指定時：
NIHONGO(3:4)は、N' 3 4 5 6'となる。

(例 5) INSPECT 文での相違

次のコード表現の場合の例を示す。

N'モ'='X'8382'

N'÷'='X'8180'

N'a'='X'8281'

N'垂'='X'889F'

```
      :  
01 NIHONGO PIC N(3) VALUE N'モ÷a'.  
01 I       PIC S9(9) COMP VALUE +0.  
      :  
      INSPECT NIHONGO TALLYING I FOR ALL N'a'  
                    REPLACING ALL N'a' BY N'垂'.  
      :
```

- -JPN,Alnum オプション指定時：
Iは、2となる。
NIHONGOは、X'83889F80889F'となる。
- -JPN,Alnum オプション未指定時：
Iは、1となる。
NIHONGOは、N'モ÷垂'となる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.25 INSPECT 文」

(例 6) STRING 文での相違

```
      :  
01 NIHONG00 PIC N(6) VALUE ALL N'O'.  
01 NIHONG01 PIC N(2) VALUE ALL N'1'.  
01 NIHONG02 PIC N(2) VALUE ALL N'2'.  
01 PTR      PIC S9(9) COMP VALUE +3.  
      :  
      STRING NIHONG01 DELIMITED BY SIZE  
            NIHONG02 DELIMITED BY SIZE  
            INTO NIHONG00 WITH POINTER PTR.  
      :
```

- -JPN,Alnum オプション指定時：
NIHONG00は、N'0 1 1 2 2 0'となる。
- -JPN,Alnum オプション未指定時：
NIHONG00は、N'0 0 1 1 2 2'となる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.47 STRING 文」

(例 7) UNSTRING 文での相違

次のコード表現の場合の例を示す。

N'モ'='X'8382'

N'÷'='X'8180'

N'a'='X'8281'

```
      :  
01  NIHONG01 PIC N(5) VALUE N' モモ ÷ a ÷' .  
01  NIHONG02 PIC N(3) .  
      :  
      UNSTRING NIHONG01 DELIMITED BY ALL N' a '  
      INTO NIHONG02 .  
      :
```

- -JPN,Alnum オプション指定時：
NIHONGO2 は、X'838283202020'となる。
- -JPN,Alnum オプション未指定時：
NIHONGO2 は、N'モモ÷'となる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.52 UNSTRING 文」

24.3.7 その他の VOS3 COBOL85 互換 (-CompatiV3)

-CompatiV3 オプションはメインフレーム (VOS3) との互換用のオプションである。-CompatiV3 オプションは、VOS3 COBOL85 で開発したプログラムを、UNIX または Windows 環境に移行するために使用する。

- ASSIGN 句で指定された外部装置名に関連する環境変数名を変更する。書き方を次に示す。
SYSnnn [-装置クラス] [-装置名] [-編成]
または
[装置クラス-] [装置名-] [編成-] ファイル定義名
 - -CompatiV3 オプション指定時の環境変数名
CBL_SYSnnn または CBL_ファイル定義名
 - -CompatiV3 オプション未指定時の環境変数名
CBL_SYSnnn [_装置クラス] [_装置名] [_編成]
または
CBL_ [装置クラス_] [装置名_] [編成_] ファイル定義名
- 入出力両用ファイルとして開いた順ファイルでは、WRITE 文の効果は REWRITE 文と同じである。WRITE 文を実行する前に、そのファイルに READ 文を実行しておかなければならない。WRITE 文は、直前の READ 文で読み込んだレコードを更新する。
- VOS3 COBOL85 上で使用できる報告書作成機能を記述したプログラムをコンパイルする。
- 画面機能では、複数プログラムで通信文が実行された場合、一つの論理端末を複数のプログラム間で共用し、送受信する。

- DIVIDE 文書き方 4 および書き方 5 で、除算の剰余計算に使う商は、商を計算したときの中間結果と同じけた数、同じ小数点位置、同じ符号の有無を持つ。

(例)

```
DIVIDE A BY B GIVING C REMAINDER D.
```

上記の例の DIVIDE 文は、下記のような連続した操作に変換される（ただし、TMP1、TMP2 および TMP3 は、コンパイラによって作成された中間結果の作業場所を表す）。

```
A / B → TMP1
TMP1 → TMP2
A - (B * TMP2) → TMP3
TMP1 → C
TMP3 → D
```

- -CompatiV3 オプション指定時の TMP2
TMP1 と同じけた数、同じ小数点位置、同じ符号になる。
- -CompatiV3 オプション未指定時の TMP2
商 (C) と同じけた数、同じ小数点位置、同じ符号になる。
- -CompatiV3 オプションと-IntResult,DecFloat40 オプションを同時に指定したとき、算術式と算術文に対する-CompatiV3 オプションの DIVIDE 文書き方 4 および書き方 5 は無効となる。この場合、-IntResult,DecFloat40 オプションの仕様で演算を実行する。詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。
- プログラム中に連結式を書けない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.13 DIVIDE 文」
「COBOL2002 言語 標準仕様編」 「10.8.33 READ 文」
「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」
「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」
「COBOL2002 言語 標準仕様編」 「13.1 報告書作成機能とは」

24.4 HITAC8000 シリーズからの移行用オプション (-H8Switch)

HITAC8000 シリーズ (EDOS/EDOS-MSO) 時代から残っている COBOL プログラムへの移行用として -H8Switch オプションが用意されている。

-H8Switch オプションで変更される言語仕様を記述する。

なお、報告書作成機能については、「7.2 報告書作成機能の HITAC8000 シリーズからの移行用オプション (-Cblctr)」を参照のこと。

24.4.1 数字項目の転記／比較 (-H8Switch)

-H8Switch を指定すると、次の場合の MOVE 文[※]および比較条件の数字項目（または固定小数点数字定数）を英数字項目とみなして転記／比較する。

- MOVE 文で、送り出し側作用対象が数字項目（または固定小数点数字定数）で、受け取り側作用対象が英数字項目（または英数字編集項目）の場合。
- 比較条件で、両辺の組み合わせが数字項目（または固定小数点数字定数）と英数字項目（または英数字編集項目、英字項目、英数字定数）の場合。

-H8Switch オプションを指定しないと、上記の場合の数字項目（または固定小数点数字定数）を英数字項目とみなさずに転記／比較する。

注※

WRITE／REWRITE／RELEASE FROM 文と ACCEPT FROM DAY／DATE／TIME／DAY-OF-WEEK 文にも適用される。

(備考)

-H8Switch 指定なしが規格の仕様である。

【標準仕様との関連】

- 「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」
- 「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」
- 「COBOL2002 言語 標準仕様編」 「10.8.35 RELEASE 文」
- 「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」
- 「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

24.5 デバッグ行の有効, 無効 (-DebugLine)

固定形式正書法の場合, 原始プログラムの行の標識領域 (7 欄) に文字「D」を書くと, その行はデバッグ行となる。このデバッグ行にデバッグのためのデータや手続きを指定しておく。

翻訳時に-DebugLine オプションを指定すると, デバッグ行は翻訳対象となり, このシステムは標識領域が空白の行と同じように翻訳する。-DebugLine オプションを指定しないと, このシステムはデバッグ行を注釈行としてみなして処理する。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「2.4.7 デバッグ行」

24.6 主プログラムの指定コマンドインタフェース (-Main,System)

-Main,System オプションが指定された最外側のプログラムで、PROCEDURE DIVISION USING で引数を指定する場合の規則を以下に示す。

- 第 1 引数は、BY VALUE 指定で整数項目 (PICTURE 句の指定で「P」を含まない) で、かつ 5~9 けたの 2 進項目でなければならない。
- 第 2 引数が BY VALUE 指定の場合は、アドレスデータ項目でなければならない。

24.7 転記の長さのチェック (-TruncCheck)

翻訳時に-TruncCheck オプションを指定すると、転記で送り出し側作用対象の長さが受け取り側作用対象の長さより大きいとき、お知らせメッセージを出す。ただし、送り出し側作用対象および受け取り側作用対象の両方またはどちらかに、チェック対象とならない項目がある場合はチェックしない。

24.7.1 チェック対象となる項目

- 外部 10 進項目
- 内部 10 進項目
- 2 進項目 (1 バイト 2 進項目を含む)
- 数字定数 (16 進数字定数を含む)
- 固定長集団項目
- 英数字項目
- 英数字定数 (16 進英数字定数を含む)
- ALL 英数字定数
- 数字型、整数型、または英数字型の、組み込み関数[※]および利用者定義関数
注[※] 送り出し側が英数字型の組み込み関数で、かつ次に示す条件の場合はチェックしない。
 - 引数が部分参照されていて、長さが可変である
長さが可変になるのは次のどちらかのときとなる。
 - 部分参照の長さがデータ名指定である
 - 部分参照の開始位置がデータ名指定で、長さが省略されている
 - 組み込み関数が MAX 関数または MIN 関数で、データ名の引数が複数指定されている
 - 組み込み関数が TRIM 関数または SUBSTRING 関数である

24.7.2 チェック対象とならない項目

- 可変長集団項目
- 英字項目
- 数字編集項目、英数字編集項目
- 外部浮動小数点項目、内部浮動小数点項目
- 指標データ項目
- 日本語項目、日本語編集項目
- アドレスデータ項目、ポインタ項目

- アドレス名
- ブール項目
- オブジェクト参照データ項目
- バリエーションデータ項目 (UNIX COBOL2002 では使用できない)
- 表意定数 (SPACE, QUOTE, HIGH-VALUE, LOW-VALUE, ZERO, NULL)
- 浮動小数点定数
- ブール定数, ALL ブール定数
- 日本語文字定数, ALL 日本語文字定数
- 部分参照がある一意名
- COMP-X 項目
- 動的長基本項目

24.7.3 チェック対象となる転記の個所

- VALUE 句
- WINDOW SECTION の SOURCE 句
- ACCEPT 文の書き方 2 (日付と時刻を取得する ACCEPT 文)
- INITIALIZE 文の REPLACING 指定
- MOVE 文
- READ 文の INTO 指定
- RELEASE 文の FROM 指定
- RETURN 文の INTO 指定
- SET 文の書き方 4
- WRITE 文の FROM 指定
- COMPUTE 文 (送り出し側作用対象が単一の数字項目や数字定数の場合)
- 上記以外の算術文 (中間結果のけた数^{※1} が受け取り側作用対象のけた数より大きい場合^{※2})

注※1

中間結果のけた数については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

注※2

64bit 版 COBOL2002 の場合、-MaxDigits38 オプションおよび-IntResult,DecFloat40 オプションを指定しているときは、常に中間結果のけた数の方が大きいと判定される。

24.7.4 受け取り側作用対象が英数字の場合

- 受け取り側作用対象が固定長集団項目のときはバイト数で比較する。
- 送り出し側作用対象が数字で、受け取り側作用対象が英数字のときは、送り出し側作用対象はけた数（小数けたを含む）で比較する。
- -H8Switch オプションが指定してあり、送り出し側作用対象が数字で受け取り側作用対象が英数字のときは、数字項目はバイト数で比較する。

24.7.5 受け取り側作用対象が数字の場合

- 整数けた数で比較する。
- 次のすべての条件が重なるときはチェックしない。
 - -DigitsTrunc オプションの指定がある
 - -MaxDigits38 オプションおよび-IntResult,DecFloat40 オプションを指定していない（64bit 版 COBOL2002 の場合）
 - 受け取り側作用対象が 2 進項目
 - VALUE 句の転記でない
- 送り出し側作用対象が 18 けた*を超える英数字項目のときはチェックしない。

注※

64bit 版 COBOL2002 の場合、-MaxDigits38 オプションおよび-IntResult,DecFloat40 オプションを指定しているときは、38 けたになる。

24.7.6 送り出し側作用対象が 2 進項目で、受け取り側作用対象が外部 10 進項目／内部 10 進項目の場合の規則（-TruncCheck,Binary オプション指定時のチェック項目）

-TruncCheck,Binary オプションは、UNIX COBOL2002 で有効なオプションである。

- 整数けた数の比較で、送り出し側作用対象のけた数が大きいときは、お知らせメッセージを出力する（-TruncCheck と同じチェック）。
- 整数けた数の比較で、送り出し側作用対象のけた数が小さくなく、次に示す 2 進項目に格納可能な最大けた数が、受け取り側作用対象の整数けた数より大きいときは、お知らせメッセージを出力する（-TruncCheck,Binary 固有のチェック）。

PICTURE 句のけた数※1	2 進項目に格納できる最大けた数※2
1～2 けた	5 けた（-Bin1Byte オプション指定時は 3 けた）

PICTURE 句のけた数※1	2進項目に格納できる最大けた数※2
3~4 けた	5 けた
5~9 けた	10 けた
10~18 けた	19 けた

注※1

整数部と小数部のけた数の合計値。

注※2

小数部があるときは、小数部のけた数を除いたけた数。

(例)

```
01 A PIC 9(4)V9(2) COMP.
```

上記の場合、整数部と小数部の PICTURE 句のけた数の合計値は 6 けたになる。2 進項目では、PICTURE 句のけた数 6 けたで格納できる最大けた数は 10 けたになるが、これから小数部のけた数 2 を除いた 8 けたを使用する。

24.7.7 -TruncCheck と-TruncCheck,Binary オプションの相違点

-TruncCheck と-TruncCheck,Binary オプションの相違点を次に示す。なお、-TruncCheck,Binary オプションは、UNIX COBOL2002 で有効なオプションである。

	-TruncCheck	-TruncCheck,Binary
MOVE B5 TO E6 ※	ノーエラー	お知らせのメッセージを出力する
MOVE B6 TO E6 ※	ノーエラー	お知らせのメッセージを出力する
MOVE B7 TO E6 ※	お知らせのメッセージを出力する	お知らせのメッセージを出力する

注※

```
01 B5 PIC 9(5) COMP.
01 B6 PIC 9(6) COMP.
01 B7 PIC 9(7) COMP.
01 E6 PIC 9(6).
```

25

その他の拡張仕様

この章では、このシステムで追加した COBOL の句や文について説明する。

25.1 全般規則

25.1.1 かたかな

(1) 文字集合中のかたかな

このシステムは、COBOL 規格の COBOL 文字集合に加えて、次に示す半角かたかなを語に使用できる。

ア, イ, ウ, エ, オ, …, ヤ, ュ, ヨ, …, ワ, シ, ` (濁点), ° (半濁点), ー (長音記号)

長音記号 (ー) は、かたかなで長音を表すときに用いる。

濁点や半濁点はそれぞれ 1 字として文字列のサイズに数える。濁点や半濁点は、かたかなと組み合わせて用いるのが普通であるが、このシステムでは独立の文字として扱う。

かたかなを用いた、正しい語と誤っている語の例を次に示す。

(例 1) 正しい語の例

キイ
ケ` ツホ° ウ
`ー°

(例 2) 誤っている語の例

マスター△ファイル (間に空白があってはならない)

(2) かたかなを使用した利用者定義語

利用者は、次に示す利用者定義語にかたかなを使用できる。

これらの名前には、少なくとも 1 個の英字またはかたかなが含まれていなければならない。ただし、かたかなが含まれていれば英字は含まれていなくてよい。

- 符号系名 (alphabet-name)
- データ名 (data-name)
- レコード名 (record-name)
- 条件名 (condition-name)
- ファイル名 (file-name)
- 指標名 (index-name)
- 記号文字 (symbolic-character)
- 字類名 (class-name)
- 呼び名 (mnemonic-name)

- 段落名 (paragraph-name)
- 節名 (section-name)

次に示す利用者定義語にはかたかなが含まれてはならない。

- プログラム名
- 原文名
- 登録集名

25.1.2 16 進英数字定数

形式

X' [文字] ...'

構文規則

1. 定数として記述できる文字は、'0'~'9', 'A'~'F'の 16 種類である。
2. 長さは 160 文字まで記述できるが、2 の倍数長でなければならない。

一般規則

1. 16 進英数字定数の扱いは、英数字定数と同じとする。したがって、英数字定数が記述できる個所であれば、次の個所を除いて 16 進英数字定数が指定できる。
 - (a) プログラム名
 - (b) 外部装置名
 - (c) 原文名
 - (d) メソッド名

25.1.3 16 進数字定数

形式

H' [文字] ...'

構文規則

1. 定数として記述できる文字は、'0'~'9', 'A'~'F'の 16 種類である。
2. 長さは 8 文字まで記述でき、2 の倍数長である必要はない。

一般規則

1.16 進数字定数の扱いは、数字（整数）定数と同じとする。したがって、数字（整数）定数が記述できる個所であれば、16 進数字定数が指定できる。

25.1.4 表意定数（POINTER 項目）

形式

[ALL] NULL
[ALL] NULLS

機能

一つまたは複数の未設定のポインタ値を表す。NULL の値を持つポインタ変数は、どんなデータ項目も指さない。

一般規則

1. 表意定数 NULL/NULLS が記述できるのは、次の個所である。
 - (a) USAGE POINTER または USAGE ADDRESS が指定された項目に記述された VALUE 句の定数。
 - (b) SCREEN SECTION に記述された VALUE 句の定数。
 - (c) ポインタ比較の条件式中の左辺または右辺。
 - (d) ポインタを設定する SET 文の送り出し側作用対象。
 - (e) CALL 文、INVOKE 文の BY VALUE に指定する定数。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」
「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

25.1.5 特殊レジスタ

(1) プログラム間連絡用特殊レジスタ

(a) RETURN-CODE

システムまたは呼び出し元プログラムへ復帰情報を返したいときには、プログラムの終了の実行に先立って特殊レジスタの RETURN-CODE（復帰コード）に情報を設定しておく。

RETURN-CODE は PICTURE 句で S9(9)と記述されている特殊レジスタの名前で、その用途（USAGE）は 2 進（BINARY）である。

このシステムは、RETURN-CODE に初期値ゼロを設定する。ゼロは、正常終了時の復帰コードを意味する。

(2) 整列併合用特殊レジスタ

(a) SORT-RETURN

SORT-RETURN（整列併合終了情報）は、PICTURE 句で S9(4)と述されている特殊レジスタの名前で、その用途（USAGE）は 2 進（BINARY）である。

SORT-RETURN レジスタは、利用者が強制的に整列併合操作を終了させるときに使用できる。入力手続きまたは出力手続き中でこのレジスタに値 16 を設定すると、次の RELEASE 文または RETURN 文の実行の後、整列併合操作は終了する。

(b) SORT-CORE-SIZE

SORT-CORE-SIZE（整列併合用記憶領域のサイズ）は、PICTURE 句で S9(8)と記述されている特殊レジスタの名前で、その用途（USAGE）は 2 進（BINARY）である。

SORT-CORE-SIZE レジスタは、利用者が作業用メモリのサイズを指定する場合に使用できる。整列処理実行前にこのレジスタに値を設定すると、整列処理開始時にこの値を用いて作業用メモリの確保をする。

(3) LABEL-RETURN 特殊レジスタ

LABEL-RETURN 特殊レジスタは、PICTURE 句で X(1)と記述されている特殊レジスタの名前であり、最外側のプログラムおよび利用者定義関数に対して一つ必ず作成される。

LABEL-RETURN 特殊レジスタは大域属性を持つ静的データである。

LABEL-RETURN 特殊レジスタは本来ユーザラベルを処理するために使用するが、このシステムでは覚え書きとみなす。

25.1.6 プログラム名、クラス名、インタフェース名、メソッド名および利用者定義関数名の日本語指定

プログラム名、クラス名、インタフェース名、メソッド名および利用者定義関数名に日本語が指定できる。

一般規則

1. プログラム名、クラス名、インタフェース名、メソッド名および利用者定義関数名に日本語を指定できる個所を次に示す。

(a) COBOL 原始プログラム内でのプログラム名、クラス名、インタフェース名、メソッド名および利用者定義関数名

ただし、-UniObjGen オプションを指定した場合、プログラム名およびメソッド名に UTF-8 で多バイトとなる文字が含まれてはならない。

また、-UniObjGen オプションの指定が前提である Linux COBOL2002 では、プログラム名、クラス名、インタフェース名、メソッド名、および利用者定義関数名に UTF-8 で多バイトとなる文字が含ま

れてはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

- (b) ccbl2002 コマンドラインでの-SimMain プログラム名/-SimSub プログラム名
- (c) テストデバッガで指定するプログラム名

2. プログラム名に日本語を含んでいる場合、指定できるプログラム名の長さは 31 文字までである。31 文字を超えた場合、31 文字をプログラム名とみなす。

3. 定数以外で指定するプログラム名中の全角文字には等価規則が適用される。

例を次に示す。

(例)

PROGRAM-ID. P プログラム名 A 0 1. は、P プログラム名 A01 とみなされる。

ただし、オプション指定によって、等価規則を適用しないこともできる。詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の-EquivRule オプションを参照のこと。また、定数で指定するプログラム名には、等価規則は適用されない。

4. 全角文字に、プログラム名の変換規則は適用されない。適用されるのは、半角文字だけである。例を次に示す。

(例)

CANCEL ' P 文字#文字¥文字@文字-# ¥@-' は、'P 文字#文字¥文字@文字-012_1' となる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「7.8 プログラム名段落 (PROGRAM-ID)」

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

「COBOL2002 言語 標準仕様編」 「10.8.5 CANCEL 文」

25.2 環境部

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8. 環境部 (ENVIRONMENT DIVISION)」

25.2.1 プログラム間連絡機能の環境部

(1) 特殊名段落 (SPECIAL-NAMES)

形式

SPECIAL-NAMES.
機能名1 IS 呼び名1

機能

機能名を呼び名に関連づける。

構文規則

1. 呼び名1の構文規則については、標準仕様編を参照のこと。

一般規則

1. 機能名1は、CDECL、STDCALLまたはPASCALでなければならない。
2. CDECL指定は、C (cdecl) 呼び出し規約を示す。
3. STDCALL指定は、standard call (stdcall) 呼び出し規約を示す。
4. PASCAL指定は、PASCAL呼び出し規約を示す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.2.6 特殊名段落 (SPECIAL-NAMES)」

(2) 外部プログラム節 (EXTERNAL-PROGRAM SECTION)

形式

EXTERNAL-PROGRAM SECTION.
CALL-CONVENTION.
{ { プログラム名1 } IS 呼び名1 } ...
 { 定数1 }

機能

外部プログラム節は、外部プログラムの属性を指定する。

UNIX COBOL2002 および Windows(x64) COBOL2002 では、外部プログラム節は覚え書きとする。

UNIX COBOL2002 の場合、呼び出し規約は常に各プラットフォームが規定する ABI に従う。
Windows(x64) COBOL2002 の場合、呼び出し規約は常に fastcall となる。呼び出し規約については、マニュアル「COBOL2002 ユーザーズガイド」の呼び出し規約の説明を参照のこと。

構文規則

1. プログラム名 1 は、利用者定義語の規則に従う。
-UniObjGen オプションを指定した場合、プログラム名 1 に UTF-8 で多バイトとなる文字が含まれてはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。
2. 定数 1 は、プログラム名を引用符で囲み、英数字定数として指定する。
-UniObjGen オプションを指定した場合、定数 1 に UTF-8 で多バイトとなる文字が含まれてはならない。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」の Unicode 機能を参照のこと。
3. 呼び名 1 は、特殊名段落で CDECL, STDCALL または PASCAL に関連づけられていなければならない。ただし、PASCAL を指定した場合、STDCALL と同様となる。
4. ほかのプログラムを含むプログラムの外部プログラム節中で明示的または暗黙的に指定した記述項は、含まれるプログラムそれぞれに適用される。
5. 外部プログラム節は、別のプログラムに直接または間接に含まれるプログラムで指定してはならない。

一般規則

1. プログラム名 1 および定数 1 は、呼び名 1 の呼び出し規約を用いている。プログラム名 1 および定数 1 を指定した CALL 文は、呼び名 1 が関連づけられた呼び出し規約に従う。
外部プログラム節に指定されていないプログラムの呼び出し規約は、C の呼び出し規約に従う。
2. プログラム名 1 および定数 1 が COBOL プログラムのときの規則を次に示す。
 - (a) COBOL プログラムが EXE 形式のとき、呼び名 1 は特殊名段落で CDECL に関連づけられていなければならない。
 - (b) COBOL プログラムが DLL 形式のとき、呼び名 1 は特殊名段落で CDECL, STDCALL または PASCAL に関連づけられていなければならない。
3. ほかのプログラムを含むプログラムの外部プログラム節中で明示的または暗黙的に指定した記述項は、含まれるプログラムそれぞれに適用される。

25.2.2 ファイル管理記述項

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4 ファイル管理記述項」

(1) ASSIGN 句のデータ名指定

形式

$$\text{ASSIGN TO } \left\{ \begin{array}{l} \text{外部装置名1} \\ \text{データ名1} \\ \text{定数1} \end{array} \right\}$$

機能

ファイル名 1 のファイルを外部装置名 1，データ名 1 または定数 1 によって参照される記憶媒体に関連づける。

構文規則

1. 定数 1 は英数字定数でなければならない。また、表意定数であってはならない。外部装置名 1 の内容および定数 1 の意味や規則は作成者が定める。
2. データ名 1 は、プログラムの実行中にファイル名を指定するために用いる。
3. データ名 1 はファイル節，作業場所節，局所場所節もしくは連絡節で定義した英数字集団項目，英字項目，または動的長基本項目でない英数字項目でなければならない。また，データ記述項に OCCURS 句があってはならず，データ記述項に OCCURS 句のある項目に従属してはならない。データ名 1 がそれら以外の定義である場合は，外部装置名とみなす。
4. データ名 1 に指定するデータ項目のサイズは，32,767 バイトを超えてはならない。

(2) ASSIGN 句の MULTIPLE REEL/UNIT 指定

形式

$$\text{ASSIGN TO } \left\{ \begin{array}{l} \text{作成者語1} \\ \text{データ名1} \\ \text{定数1} \end{array} \right\} \dots$$
$$[\text{FOR MULTIPLE } \left\{ \begin{array}{l} \text{REEL} \\ \text{UNIT} \end{array} \right\}]$$

このシステムでは ASSIGN 句の FOR MULTIPLE {REEL/UNIT} 指定を覚え書きとみなす。

(3) FILE-LIMIT 句

形式

$$\left\{ \begin{array}{l} \text{FILE-LIMIT IS} \\ \text{FILE-LIMITS ARE} \end{array} \right\} \left\{ \begin{array}{l} \text{データ名1} \\ \text{定数1} \end{array} \right\} \left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \left\{ \begin{array}{l} \text{データ名2} \\ \text{定数2} \end{array} \right\} \dots$$

このシステムでは FILE-LIMIT 句を覚え書きとみなす。

機能

FILE-LIMIT 句は，大記憶媒体上のファイル領域の始めと終わりを指定する。

(4) PASSWORD 句

形式

PASSWORD IS データ名

ただし、索引ファイル、RDB ファイルの場合は以下の形式となる。

RECORD KEY IS データ名1 [PASSWORD IS データ名2]
ALTERNATE RECORD KEY IS データ名1
[PASSWORD IS データ名2]
[WITH DUPLICATES]

(備考)

RECORD KEY 句と ALTERNATE RECORD KEY 句は、書き方を明確にするために示してあるだけで、PASSWORD 句の一部ではない。

このシステムでは PASSWORD 句を覚え書きとみなす。

機能

ファイルの作成、読み込みまたは更新する権利があるかどうかをシステムに問い合わせる。

(5) RESERVE 句の ALTERNATE 指定

形式

$$\underline{\text{RESERVE}} \left\{ \begin{array}{c} \text{NO} \\ \text{整数1} \end{array} \right\} \text{ALTERNATE} \left\{ \begin{array}{c} \text{AREA} \\ \text{AREAS} \end{array} \right\}$$

このシステムでは NO 指定および ALTERNATE 指定を覚え書きとみなす。

機能

割り当てられる入出力領域の個数を指定できる。

一般規則

1. RESERVE 句を書くと、割り当てられる入出力領域の個数は整数 1 の値となる。RESERVE 句を書かない場合、割り当てられる入出力領域の個数は作成者が定める。
2. 整数 1 は、符号のない正の整数で、その値は 1～255 でなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(16) RESERVE 句」

(6) TRACK-AREA 句

形式

$$\underline{\text{TRACK-AREA}} \text{ IS} \left\{ \begin{array}{c} \text{データ名} \\ \text{整数} \end{array} \right\} \text{CHARACTERS}$$

このシステムでは TRACK-AREA 句を覚え書きとみなす。

機能

乱アクセスの索引ファイルにレコードを追加するときの効率を高めたいときに指定する。

(7) SELECT 句の NOT OPTIONAL 指定

形式

```
SELECT [[NOT] OPTIONAL] ファイル名
```

このシステムでは NOT OPTIONAL 指定を覚え書きとみなす。

機能

プログラムで使用するファイルに名前を付ける。

構文規則

1. NOT OPTIONAL (確定) 句は、入出力両方で開くファイルだけに指定できる。

(8) FILE STATUS 句のデータ名 2 指定

形式

```
FILE STATUS IS データ名1 [データ名2]
```

このシステムでは FILE STATUS 句のデータ名 2 指定を覚え書きとみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.4(7) FILE STATUS 句」

25.2.3 入出力管理記述項

(1) APPLY WRITE-ONLY 句

形式

```
APPLY WRITE-ONLY ON  
{ファイル名1}…
```

APPLY WRITE-ONLY 句は、出力ファイルの書き出しの効率を高めるために指定する。ただし、このシステムでは覚え書きとみなす。

構文規則

1. 複数個の APPLY WRITE-ONLY 句を指定してもよい。ただし、同じファイル名が APPLY WRITE-ONLY 句に複数回現れてはならない。
2. ファイル名は整列併合用ファイルの名前であってはならない。

3. ファイル名はブロッキングしてある可変長レコードから成る順ファイルでなければならない。

(2) APPLY FORMS-NAME 句

形式

```
APPLY FORMS-NAME  
TO データ名 1 ON {ファイル名 1}...
```

機能

GDI モードおよび ESC/P モード印刷機能を使用するとき、APPLY FORMS-NAME 句に、スプールに登録する印刷文書名称を設定する。

構文規則

1. データ名 1 は英字、英数字、固定長集団、日本語項目でなければならない。
2. データ名 1 の領域に、スプールに登録する印刷文書名称を設定する。
3. ファイル名 1 は順ファイルまたはテキストファイルでなければならない。
4. ファイル名 1 には EXTERNAL 句を指定したファイルを指定できない。

(3) APPLY FILE-SHARE 句

形式

```
APPLY FILE-SHARE  
ON ファイル名1 [ファイル名2]...
```

機能

二つ以上のプログラムが同じファイルを共用することを指定する。この機能は MIOS7 互換機能である。

構文規則

1. ファイル名 1, ファイル名 2 などは、同じファイル名であってはならない。また、同じファイル名を二つ以上の APPLY 句に書いてはならない。
2. LOCK MODE 句で MANUAL, または EXCLUSIVE を指定したファイルを APPLY FILE-SHARE 句で指定してはならない。
3. 整列用ファイルに APPLY FILE-SHARE 句は指定できない。
4. -IgnoreAPPLY,FILESHARE オプション指定時、覚え書きとみなす。

一般規則

APPLY FILE-SHARE 句を指定すると、ファイル名 1 に対して、LOCK MODE IS AUTOMATIC [WITH LOCK ON RECORD] を指定したものと仮定する。

注意事項

APPLY FILE-SHARE 句は、MIOS7 COBOL85 との互換のための日立拡張仕様である。

UNIX COBOL2002 では、APPLY FILE-SHARE 句は使用できない。ただし、- IgnoreAPPLY,FILESHARE オプション指定時に覚え書きとみなす。

(4) その他の APPLY 句

形式

メインフレームの COBOL で指定可能な APPLY 句を記述できる。形式については、マニュアル「VOS3 COBOL85 言語」を参照のこと。

ただし、このシステムではこれらを覚え書きとみなす。

機能

装置固有の処理を指定する。

(5) RERUN 句

形式

`RERUN ON` 外部装置名

このシステムでは RERUN 句を覚え書きとみなす。

機能

整列併合操作中に、再開のための情報をとることを指定する。

構文規則

1. 外部装置名は標準仕様編の RERUN 句で示した規則に従う。ただし、外部装置名は、一般の RERUN 句に書いた外部装置名と同じものであってもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「8.3.8 RERUN 句」

25.3 データ部

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

25.3.1 ファイル記述項

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.3.4 ファイル記述項 (file description entry)」

(1) LABEL RECORD 句のデータ名指定

形式

$$\text{LABEL} \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \left\{ \begin{array}{l} \text{STANDARD} \\ \text{OMITTED} \\ \text{データ名1 [データ名2] ...} \\ \text{[TOTALING AREA IS} \\ \quad \text{データ名3} \\ \text{TOTALLED AREA IS} \\ \quad \text{データ名4]} \end{array} \right\}$$

このシステムでは LABEL RECORD データ名指定を覚え書きとみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.38 LABEL RECORDS 句」

「COBOL2002 言語 標準仕様編」 「10.8.30 OPEN 文」

(2) RECORDING MODE 句

形式

$$\text{RECORDING MODE IS} \left\{ \begin{array}{l} \text{F} \\ \text{V} \\ \text{U} \\ \text{S} \end{array} \right\}$$

機能

RECORDING MODE 句は、ファイル中のレコードの形式を明示的に指定する。

一般規則

1. F を書くと、固定長レコード形式を指定したことになる。

次の規則が適用される。

- (a) ファイルのレコード記述中に DEPENDING 指定のある OCCURS 句で記述した項目があってはならない。
- (b) このファイル記述項の後に、複数のレコード記述があるときには、各レコード記述から決まるレコードのサイズはすべて同じでなければならない。
- (c) このファイルの RECORD 句に VARYING 指定があってはならない。

2. V を書くと、可変長レコード形式を指定したことになる。

(a) 形式 V では、一つのレコードが完全に一つのブロックに含まれていれば、レコード記述がどのように組み合わせられていてもよい。

3. U を書くと、不定長レコード形式を指定したことになる。このシステムでは、V と同じに扱う。

4. S を書くと、スパンドレコード形式を指定したことになる。このシステムでは、V と同じに扱う。

5. RECORDING MODE 句を省略した場合にレコード形式を決定する規則は、次による。

- ・次のどれかが成り立つならば、形式 V が仮定される。
 - ・ RECORD IS VARYING 句が指定されている。
 - ・ DEPENDING 指定のある OCCURS 句を含むレコード記述がある。
 - ・ サイズの異なるレコードが記述されている。
- ・ 前項の三つの項目に該当しないならば、形式 F が仮定される。

25.3.2 データ記述項

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.13 データ記述項 (data description entry)」

(1) OCCURS 句 (可変長項目の複数回記述)

形式

```
OCCURS [整数1 TO] 整数2 TIMES
  DEPENDING ON データ名1
  [ { ASCENDING }
    { DESCENDING } ] KEY IS {データ名2}... ]...
  [INDEXED BY {指標名1}...]
```

機能

1 レコード中に DEPENDING ON 指定のある OCCURS 句を複数回記述できる。

構文規則

1. 可変長項目、すなわち可変反復データ項目を含む集団項目の記述項に OCCURS 句を指定してもよい。
2. レコード記述中に、DEPENDING ON 指定のある OCCURS 句を書いたデータ項目の後の文字位置を占めるデータ項目を記述してもよい。

3. 可変部分（すなわち、可変反復データ項目以降の文字位置）に定義されたデータ項目を、次の個所に指定してはならない。
- (a) OCCURS 句の KEY IS 指定
 - (b) OCCURS 句の DEPENDING ON 指定
 - (c) RECORD 句の DEPENDING ON 指定
 - (d) RECORD KEY 句および ALTERNATE RECORD KEY 句
 - (e) MERGE 文の KEY 指定
 - (f) FILE STATUS 句
 - (g) RELATIVE KEY 句
- (例)

```
01 TBL.  
02 E OCCURS 10 DESCENDING B INDEXED I.  
03 A PIC X OCCURS 10 TO 100 DEPENDING J  
INDEXED K.  
03 B PIC 9(5).
```

キーが可変部分よりも後にあるので誤りだが、このシステムでは、エラーとしない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.51 OCCURS 句」

(2) PICTURE 記号 ('G')

一般規則

1. このシステムでは、PICTURE 文字列中の 'G' と 'N' は同じとみなす。ただし、CURRENCY SIGN 句の文字列に 'G' を許し、その場合、'G' を通貨記号とみなす。
2. 'G' と 'N' は混在して使用できない。ただし、このシステムではエラーとならない。
 - (a) 同一のソース単位内に PICTURE 記号 'G' の項目と 'N' の項目が混在してもよい。
 - (b) 同一の PICTURE 文字列内に、'G' と 'N' が混在してもよい。ただし CURRENCY SIGN 句に 'G' を指定した場合はこの限りでなく、'G' は通貨編集用記号とみなされる。
3. 日本語項目を定める場合の規則を次に示す。
 - (a) PICTURE 句の文字列は、文字 'N' または 'G' だけから成る。
 - (b) 項目の内容は、標準データ形式で表したとき、一つ以上の拡張コードの文字でなければならない。
4. 日本語編集項目を定める場合の規則を次に示す。
 - (a) PICTURE 句の文字列は、文字 'N'、'G' または 'B' の組み合わせから成り、'N'、'G' または 'B' をそれぞれ少なくとも一つずつ含まなければならない。
 - (b) 項目の内容は、標準データ形式で表したとき、拡張コードの二つ以上の文字でなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

(3) SYNCHRONIZED 句を集団項目に指定

SYNCHRONIZED 句を集団項目に指定してもよい。このときは、その集団に含まれるすべての基本項目に、SYNCHRONIZED 句を指定したのと同じとみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.80 SYNCHRONIZED 句」

(4) 78 定数名の記述

形式

78 定数名 VALUE IS 定数

機能

定数名を指定するのに用いる。

(5) 定数名での VALUE 句の規則

- 定数名記述項には、必ず VALUE 句を書かなければならない。この記述項に書けるのは VALUE 句と定数名だけとする。

定数名は参照されたとき、VALUE 句で指定された定数に置き換える。データ部中で参照するには、定数名は参照する前に定義されていなければならない。

(6) USAGE IS POINTER 句

形式

[USAGE IS] POINTER

一般規則

- POINTER は、ポインタ項目であることを指定する。
このシステムでは、ポインタ項目とアドレスデータ項目は同じとみなす。
ポインタ項目と記載した個所にはアドレスデータ項目が指定できる。
また、アドレスデータ項目と記載した個所にはポインタ項目が指定できる。

(7) USAGE COMP-5 を指定した項目に設定できる初期値 (VALUE 句の値) の拡張

USAGE COMP-5 を指定した項目に設定できる初期値は、USAGE COMP を指定した項目に設定できる初期値より拡張している。

PICTURE 句で指定したけた数と、VALUE 句に指定できる値との対応を次に示す。

PICTURE 句で指定したけた数	VALUE 句に指定できる値
1～4 けた	$-2^{15} \sim 2^{15}-1$
5～9 けた	$-2^{31} \sim 2^{31}-1$
10～18 けた	$-999,999,999,999,999,999 \sim 999,999,999,999,999,999$

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 [9.16.54 PICTURE 句]

「COBOL2002 言語 標準仕様編」 [9.16.88 VALUE 句]

(8) EXTERNAL 句の DYNAMIC 指定

形式

IS EXTERNAL DYNAMIC

機能

外部属性を持つデータ項目の領域を、プログラムの実行時に動的にメモリに割り当てる。

構文規則

- EXTERNAL 句の DYNAMIC 指定は、作業場所節のレコード記述項だけに記述できる。
- EXTERNAL 句の DYNAMIC 指定は、共用ライブラリまたは DLL を作成する場合でも記述できる。
共用ライブラリについては、マニュアル「COBOL2002 使用の手引 手引編」の実行可能ファイルと共用ライブラリの作成を参照のこと。DLL の作成方法については、マニュアル「COBOL2002 ユーザーズガイド」の実行可能ファイルと DLL の作成を参照のこと。

一般規則

- 一つの実行単位の中で、二つ以上のプログラムが同じデータレコードを記述する場合、DYNAMIC 指定のない外部データレコードが混在してはならない。
- DYNAMIC 指定をしたデータ領域は、C プログラムから参照、更新できない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 [9.16.25 EXTERNAL 句]

25.4 手続き部

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

25.4.1 算術文の作用対象のけた数

各作用対象のけた数の合計は、想定した位取りも含めて 30 けた以下であればよい。

25.4.2 字類条件の内部 10 進項目指定

作用対象が数字項目のとき、その用途 (USAGE) は内部 10 進 (PACKED-DECIMAL, COMP-3) であってもよい。

25.4.3 ポインタ項目の参照

ポインタ項目の参照は次の場所に許される。

- ポインタ比較の条件式中の左辺または右辺。
- ポインタを設定する SET 文の送り出し側作用対象または受け取り側作用対象。
- CALL 文 BY VALUE 指定。

(1) ポインタ比較の条件式

形式

$$\left\{ \begin{array}{l} \text{ADDRESS OF 一意名1} \\ \text{一意名2} \\ \text{NULL} \end{array} \right\} \left\{ \begin{array}{l} \text{IS} \\ \text{ARE} \end{array} \right\}$$

[NOT] EQUAL TO

$$\left\{ \begin{array}{l} \text{ADDRESS OF 一意名3} \\ \text{一意名4} \\ \text{NULL} \end{array} \right\}$$

機能

二つの項目の USAGE (用途) が、明示的または暗黙的に POINTER であれば比較できる。
この場合の演算子は、EQUAL と NOT EQUAL だけ使用できる。

構文規則

- 一意名 1 と一意名 3 は、連絡節、局所場所節または作業場所節で定義した 01～49、66 または 77 レベルのデータ項目である。次に示すものは指定できない。
 - 関数一意名とオブジェクトプロパティ
 - ファクトリ定義やインスタンス定義で定義されたデータ項目
 - オブジェクト参照や、強く型付けされた集団項目に従属する基本項目
- 一意名 2 と一意名 4 は、ポインタ項目でなければならない。
- 比較条件対象が両方とも表意定数 NULL の場合、比較はしない。
- 一意名 1 と一意名 3 がブール項目のときは、バイト境界に調整しておかなければならない。

一般規則

二つのアドレスが同じ場合だけ等しい結果となる。それ以外の場合は等しくない。

(2) SET 文

形式

$$\text{SET } \left\{ \begin{array}{l} \text{ADDRESS OF 一意名1} \\ \text{一意名2} \end{array} \right\} \text{ TO } \left\{ \begin{array}{l} \text{ADDRESS OF 一意名3} \\ \text{一意名4} \\ \text{NULL} \end{array} \right\}$$

機能

データ項目のアドレスをポインタ変数に割り当てるのに使用する。

構文規則

- 一意名 1 は、連絡節で定義した 01 レベルまたは 77 レベルの項目である。一意名 1 には、強く型付けされた集団項目を指定できない。
- 一意名 1 は、可変長項目であってはならない。
- 一意名 1 は、部分参照してはならない。
- 一意名 3 は、連絡節、局所場所節または作業場所節で定義した 01～49、66 または 77 レベルの項目である。次に示すものは指定できない。
 - 関数一意名やオブジェクトプロパティ
 - ファクトリ定義やインスタンス定義で定義されたデータ項目
 - オブジェクト参照データ項目や、強く型付けされた集団項目に従属する基本項目
 - 強く型付けされた集団項目
- 一意名 2 と一意名 4 は、ポインタ項目である。
- 一意名 3 がブール項目のときは、バイト境界に調整しておかなければならない。

一般規則

1. TO 指定中のポインタ項目のアドレスが、一意名 2 で表されたポインタ項目へ移される。一意名 1 がある場合、これに従属する任意の参照は TO 指定が識別するアドレスを持つ項目を対象として実行される。TO 指定が識別するアドレスは、一意名 4 に含まれるアドレスまたは一意名 3 のアドレスである。TO 指定に NULL を書くとプログラムのどんなデータも指さないポインタとみなされる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.43 SET 文」

25.4.4 ACCEPT 文の拡張

形式

ACCEPT 一意名
[FROM {
 SYSIN
 SYSIPT
 CONSOLE
 SYSSTD
 呼び名1
}]

構文規則

1. -NumAccept オプションを指定したとき、一意名は用途が表示用でない数字項目でもよい。

一般規則

1. FROM 指定に機能名 (SYSIN, SYSIPT, CONSOLE, SYSSTD) を指定してもよい。それぞれの機能に関連づけた呼び名 1 を指定したのと同じ意味になる。
2. 転送データは受け取り側データ項目の左端から格納される。受け取り側データ項目の文字数が転送データの文字数より大きい場合、残りの部分は空白文字でクリアされる。
3. 受け取り側データ項目が数字または数字編集項目の場合を次に示す。
 - (a) 受け取り側データ項目の文字数または受け取り側データ項目の残り部分の文字数が転送データの文字数より大きいとき、転送データは受け取り側データ項目または受け取り側データ項目の残りの部分に、編集文字に合わせて右詰めされ、SYSSTD の場合だけ続いて残りの部分のデータが要求される。残りの部分を空白文字でクリアする。
 - (b) 受け取り側データ項目の文字数または受け取り側データ項目の残り部分の文字数が転送データの文字数より小さいとき、編集文字に合わせて転送データの左側の部分だけが受け取り側データ項目または受け取り側データ項目の残りの部分に格納される。転送データの右側の残り部分は無視される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.1 ACCEPT 文」

25.4.5 CALL 文の引数指定の拡張

形式（書き方 1，書き方 2 共に共通部分だけ）

$$\begin{array}{l} \text{CALL} \left\{ \begin{array}{l} \text{一意名1} \\ \text{定数1} \end{array} \right\} \\ \text{[USING} \\ \left. \left. \left. \begin{array}{l} \text{[BY REFERENCE]} \\ \left\{ \begin{array}{l} \text{一意名2} \\ \text{ADDRESS OF 一意名5} \end{array} \right\} \dots \\ \text{BY CONTENT} \\ \left\{ \begin{array}{l} \text{定数3} \\ \text{一意名2} \\ \text{LENGTH OF 一意名6} \end{array} \right\} \dots \\ \text{BY VALUE} \\ \left\{ \begin{array}{l} \text{定数4} \\ \text{一意名7} \\ \text{LENGTH OF 一意名6} \end{array} \right\} \dots \end{array} \right\} \dots \right\} \dots \end{array}$$

構文規則

1. 定数 3，定数 4 は，英数字定数，小数点のない 9 けた以内の数字定数，浮動小数点数字定数，および表意定数 ZERO でなければならない。ただし，定数 4 には表意定数 NULL も指定できる。
2. ADDRESS OF に指定する一意名 5 がブール項目のときは，バイト境界に調整しておかなければならない。
3. ADDRESS OF 一意名や LENGTH OF 一意名は，ファイル節，作業場所節，局所場所節，連絡節または画面節（WINDOW SECTION）で定義したデータ項目でなくてはならない。したがって，関数一意名やオブジェクトプロパティであってはならない。また，ADDRESS OF の一意名は，ファクトリ定義やインスタンス定義で定義されたデータ項目であってはならない。
4. ADDRESS OF 一意名は，オブジェクト参照データ項目であってはならない。また，強く型付けされた集団項目に従属する基本項目を指すものであってはならない。
5. ADDRESS OF 一意名は，強く型付けされた集団項目であってはならない。

一般規則

1. ADDRESS OF 一意名 5 を指定している場合，呼び出し先プログラムに渡される値は，一意名 5 のアドレスである。呼び出し先プログラムは，[BY REFERENCE] データ名で引数を受け取る。データ名の属性は，アドレスデータ項目でなければならない。
2. LENGTH OF 一意名 6 を指定している場合，呼び出し先プログラムに渡される値は，一意名 6 の長さである。呼び出し先プログラムは，[BY REFERENCE] /BY VALUE データ名で引数を受け取る。データ名の属性は，4 バイト（64bit 版 COBOL2002 の場合は，8 バイト）の 2 進項目でなければならない。
3. 定数 3 を指定している場合，呼び出し先プログラムに渡される値は，定数値である。呼び出し先プログラムは，[BY REFERENCE] データ名で引数を受け取る。データ名の属性は，指定された定数の種類によって異なる。受け取るデータ名の属性を次に示す。

- (a) 英数字定数：指定した定数と同じ長さの項目
- (b) 数字定数／表意定数 ZERO：4 バイトの 2 進項目
- (c) 浮動小数点定数：USAGE COMP-2 の項目（倍精度内部浮動小数点数字項目）

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

25.4.6 CALL 文（下線付き定数の呼び出し）

この機能は、-UscoreStart オプションが指定されたときに有効になる。

形式（書き方 1，書き方 2 共に共通部分だけ）

CALL 定数 1

構文規則

1. 定数 1 の先頭から '_' または '__'（下線 2 個）を除いた部分をプログラム名とする。
2. 定数 1 は '_' または '__'（下線 2 個）だけから成ってはならない。
3. 上記以外の規則については標準仕様に従う。

一般規則

1. 定数 1 から '_' または '__'（下線 2 個）を除いたプログラム名で呼び出し先プログラムを呼び出す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.4 CALL 文」

25.4.7 DISPLAY 文（機能名を直接指定）

形式

$$\text{DISPLAY } \left\{ \begin{array}{l} \text{一意名} \\ \text{定数} \end{array} \right\} \dots$$

$$\left[\text{UPON } \left\{ \begin{array}{l} \text{CONSOLE} \\ \text{SYSPUNCH} \\ \text{SYSPCH} \\ \text{SYSOUT} \\ \text{SYSLST} \\ \text{呼び名} \end{array} \right\} \right]$$

1. UPON 指定に機能名（CONSOLE, SYSPUNCH, SYSPCH, SYSOUT, SYSLST）を指定してもよい。それぞれの機能名に関連づけた呼び名を指定したのと同じ意味になる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.12 DISPLAY 文」

25.4.8 ENTRY 文

形式

```
ENTRY 定数  
[ USING {データ名1}… ]
```

機能

呼び出し先プログラムの入口点を指定する。

構文規則

1. 原始プログラムが入れ子になっているとき、ENTRY 文を書いてはならない。
2. 定数は、プログラム名の構成規則に従った英数字定数でなければならない。プログラム名の構成規則は、マニュアル「COBOL2002 言語 標準仕様編」 「7.8 プログラム名段落 (PROGRAM-ID)」を参照のこと。
3. 定数は、プログラム名段落で指定したプログラム名と同一であってはならない。
4. 一つの実行単位を構成する幾つかのプログラムのプログラム名および入口名はそれぞれ一意でなければならない。
5. プログラムの手続き部見出しに RETURNING 指定があるプログラムに、ENTRY 文を書いてはならない。ENTRY 文はプログラム定義にだけ指定できる。

一般規則

1. 定数は、呼び出し先プログラムの入口点の名前を指定する。
呼び出すプログラムの CALL 文で渡された制御は、呼び出し先プログラムの入口点が ENTRY 文で指定してあるとき、この ENTRY 文のすぐ後に続く文に移る。
2. USING 指定は、呼び出し先プログラムを参照するデータ項目を指定する。規則は、マニュアル「COBOL2002 言語 標準仕様編」 「10.1.1 一般形式」を参照のこと。

(例)

(呼び出し元プログラム)

```
DATA DIVISION.  
:  
WORKING-STORAGE SECTION.  
01 A PIC X(2).  
01 B PIC 9(3).  
01 C.  
05 D PIC X.  
05 E PIC 9(5).  
01 F.  
05 G OCCURS 10 INDEXED H PIC X(4).
```

PROCEDURE	:	DIVISION.
U. CALL 'SUB1'	:	USING B C. ... (1)
V. CALL 'SUB2'	:	USING A F. ... (2)

(呼び出し先プログラム)

DATA	:	DIVISION.
LINKAGE	:	SECTION.
01 K	:	PIC 9(3).
01 L	:	PIC X(2).
01 M.	:	
05 N OCCURS 10 INDEXED 0	:	PIC X(4).
01 P.	:	
05 Q	:	PIC X.
05 R	:	PIC 9(5).
PROCEDURE	:	DIVISION.
X. ENTRY 'SUB1'	:	USING K P. ... (3)
Y. ENTRY 'SUB2'	:	USING L M. ... (4)

上の例で、(1)の CALL 文を実行すると、制御は呼び出し先プログラムの入口点'SUB1'に移る(3)。呼び出し先プログラムの手続き部で連絡中の K, P, Q, R を参照すると、それは呼び出し元プログラムの B, C, D, E をそれぞれ参照するのと同じ効果を持つ。

(2)の CALL 文を実行すると、制御は呼び出し先プログラムの入口点'SUB2'に移る(4)。このとき、呼び出し先プログラムの手続き部で連絡中の L, M, N を参照すると、それは呼び出し元プログラムの A, F, G をそれぞれ参照するのと同じ効果を持つ。ただし、連絡節中の指標名 O が指す指標は、呼び出すプログラム中の指標名 H が指す指標名と同じではない。

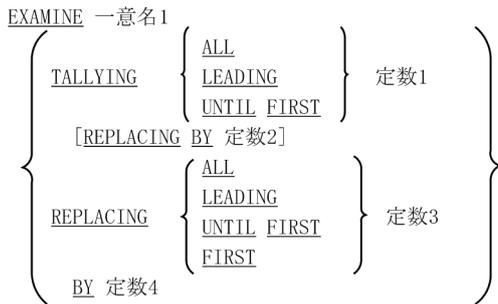
3. プログラム中で、書いたとおり上から下への順序でまたは GO TO 文などで ENTRY 文に制御が移った場合、何も行われずに ENTRY 文の次の文に制御が移る。
4. 呼び出し先プログラムの入口点が手続き部分の先頭のときは、ENTRY 文を指定する必要はない。
5. ENTRY 名称を指定した CANCEL 文を実行しても、何もしない。制御は、明示的な CANCEL 文に続く次の実行文に移る。
6. ENTRY 文の記述があるソースファイルに、-Main オプションを指定しても、ENTRY に制御は渡らない。

【標準仕様との関連】

- 「COBOL2002 言語 標準仕様編」 「7.8 プログラム名段落 (PROGRAM-ID)」
- 「COBOL2002 言語 標準仕様編」 「10.1.1 一般形式」

25.4.9 EXAMINE 文と TALLY 特殊レジスタ

形式



機能

EXAMINE 文は、一つのデータ項目中のある文字の出現回数を調べたり、ある文字をほかの文字で置き換えたりする。

構文規則

- 一意名 1 の用途は、表示用 (DISPLAY) でなければならない。ブール項目は指定できない。
- 定数 (定数 1~定数 4) は、1 文字から成る英数字定数か、符号のない 1 けたの整数かまたは ALL 定数以外の表意定数でなければならない。

一般規則

- けた調べは、一意名 1 のデータ項目の最左端のけたから右の方へ順に行われる。
- TALLYING 指定と TALLY 特殊レジスタ、TALLYING 指定は、ある文字の出現回数を TALLY 特殊レジスタに入れることを指定する。出現回数の指定方法は次のとおりである。

(a) ALL (全部) 指定

すべての定数 1 の出現回数が数えられる。

(例)

A の内容が ABACADAE のとき EXAMINE A TALLYING ALL 'A' を実行すると、TALLYING には 4 が入る。

(b) LEADING (先行) 指定

定数 1 以外の文字が初めて出てくるまで、定数 1 の出現回数が数えられる。もし、データ項目の内容がすべて定数 1 ならば、最右端のけたまで数え上げが行われる。

(例)

B の内容が 00009800 のとき EXAMINE B TALLYING LEADING 0 を実行すると、TALLY には 4 が入る。

(c) UNTIL FIRST (最初まで) 指定

定数 1 が初めて出てくるまで、定数 1 以外の文字出現回数が数えられる。もし、データ項目中に一つも定数 1 がなければ最右端のけたまで数え上げが行われる。

(例)

C の内容が+3456.78 のとき、EXAMINE C TALLYING UNTIL FIRST '.'を実行すると、TALLYには5が入る。

3. REPLACING 指定

ある文字をほかの文字で置き換えることを指定する。置き換えの指定方法は次のとおりである。

(a) ALL (全部) 指定

定数1や定数3は、すべて定数2や定数4で置き換えられる。

(例)

A の内容が△△34567△のとき EXAMINE A REPLACING ALL SPACE BY ZERO を実行すると、A の内容は 00345670 となる。

(b) LEADING (先行) 指定

定数1や定数3以外の文字が出てくるまで、定数1や定数3は、定数2や定数4で置き換えられる。もし、データ項目の内容がすべて定数1や定数3ならば、最右端のけたまで置き換えが行われる。

(例)

B の内容が***ABC**のとき EXAMINE B REPLACING LEADING '*' BY 'Z'を実行すると、B の内容は ZZZABC**となる。

(c) UNTIL FIRST (最初まで) 指定

定数1や定数3が出てくるまで、定数1や定数3以外の文字は、定数2や定数4で置き換えられる。もし、データ項目中に一つも定数1や定数3がなければ、最右端のけたまで置き換えが行われる。

(例)

C の内容がXYZAXYZA のとき EXAMINE C REPLACING UNTIL FIRST 'A' BY 'A'を実行すると、C の内容は AAAAXYZA となる。

(d) FIRST (最初の) 指定

最初に出てきた定数1や定数3は、定数2や定数4で置き換えられる。

(例)

D の内容がABCABCAB のとき EXAMINE D REPLACING FIRST 'C' BY 'X'を実行すると、D の内容は ABXABCAB となる。

4. 一意名が数字項目のときのけた調べおよび置き換えは次の規則に従う。

(a) 最右端 (SIGN LEADING 句の指定があるときは最左端) のけたに符号が重ね合わせられていても、この符号は無視される。

(例)

DATA-1 の内容は次のとおりとする。

0	1	2	3	4	1	2	3	1	2	1
---	---	---	---	---	---	---	---	---	---	---

最右端のけたに重ね合わされている符号は無視され、このけたは1とみなされる。

EXAMINE DATA-1 TALLYING ALL 1 を実行すると、TALLYには、1の個数3が入る。

(b) REPLACING による置き換えを行っても、重ね合わせられている符号は変わらない。

(例)

DATA-1 の内容は次のとおりとする。

1	2	3	1	2	1	0	1
---	---	---	---	---	---	---	---

EXAMINE DATA-1 REPLACING ALL 1 BY 2 を実行すると、DATA-1 の内容は次のようになる。

2	2	3	2	2	2	0	2
---	---	---	---	---	---	---	---

(備考)

EXAMINE 文は、第 1 次規格にあったが、第 2 次規格から削除された。代わりに、より一般的で強力な INSPECT 文が取り入れられた。図 25-1 EXAMINE 文と INSPECT 文の対応に EXAMINE 文と同じデータ操作をする INSPECT 文の書き方を示す。

TALLY 特殊レジスタ

TALLY 特殊レジスタの初期値はゼロである。整数の基本項目が使えるところならばどこでも同じように使用できる。

図 25-1 EXAMINE 文と INSPECT 文の対応

EXAMINE 一意名1 TALLYING { ALL } 定数1 { LEADING } [REPLACING BY 定数2]	MOVE 0 TO TALLY INSPECT 一意名1 TALLYING TALLY FOR { ALL } 定数1 { LEADING } [REPLACING { ALL } 定数1 BY 定数2] { LEADING }
EXAMINE 一意名1 TALLYING UNTIL FIRST 定数1 [REPLACING BY 定数2]	MOVE 0 TO TALLY INSPECT 一意名1 TALLYING TALLY FOR CHARACTERS BEFORE 定数1 [REPLACING CHARACTERS BY 定数2 BEFORE 定数1]
EXAMINE 一意名1 REPLACING { ALL } { FIRST } 定数3 BY 定数4 { LEADING }	INSPECT 一意名1 REPLACING { ALL } { FIRST } 定数3 BY 定数4 { LEADING }
EXAMINE 一意名1 REPLACING UNTIL FIRST 定数3 BY 定数4	INSPECT 一意名1 REPLACING CHARACTERS BY 定数4 BEFORE 定数3

TALLY 特殊レジスタはプログラム中に EXAMINE 文を書いたとき、このシステムがプログラムごとに 1 個だけ生成する特殊レジスタである。

TALLY 特殊レジスタのサイズと用途は、PICTURE 9(5) USAGE COMP である。

25.4.10 GO TO MORE-LABELS 文

形式

GO TO MORE-LABELS

構文規則

1. GO TO MORE-LABELS 文は、プログラムまたは利用者定義関数の USE LABEL 文の手続き中でだけ使用できる。これ以外の個所で使用してはならない。

一般規則

1. GO TO MORE-LABELS 文は、本来複数のユーザラベルを処理するために使用する。
2. GO TO MORE-LABELS 文を実行すると、USE 手続きの最初の文に制御を戻す。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.21 GO TO 文」

25.4.11 MOVE 文 (日付と時刻用)

形式

書き方 1

MOVE WHEN-COMPILED TO 一意名

書き方 2

MOVE { CURRENT-DATE
TIME-OF-DAY } TO 一意名

機能

1. 利用者がプログラムを翻訳した時点の日付と時刻を実行時に得たいときには、WHEN-COMPILED を使う (書き方 1)。
2. MOVE 文の送り出し側作用対象に、特別な予約語 CURRENT-DATE または TIME-OF-DAY と書いて、今日の日付または現在の時刻を得ることができる (書き方 2)。

一般規則

1. WHEN-COMPILED は、暗黙に長さ 20 文字の英数字項目としてこのシステムが確保した記憶場所の名前であり、MOVE 文の送り出し側作用対象としてだけ使うことができる。項類は英数字である。この英数字項目の内容は、YY-MM-DD△△HH:MM:SS△△ (年-月-日△△時:分:秒△△) である。4 けたの西暦を取得する場合は、標準仕様編の WHEN-COMPILED 関数を参照のこと。

(例)

翻訳時の日付と時刻を 2000 年 1 月 1 日 15 時 40 分 32 秒とすると、WHEN-COMPILED には、次のように入っている。

00-01-01 15:40:32

2. CURRENT-DATE は、暗黙に長さ 8 文字の英数字項目としてこのシステムが確保する記憶場所の名前であり、MOVE 文の送り出し側作用対象としてだけ使うことができる。項類は英数字である。プログ

ラム実行中、この MOVE 文を実行した時点の日付が、「MM/DD/YY (月/日/年)」の形式で格納されている。4 けたの西暦を取得する場合は、標準仕様編の CURRENT-DATE 関数を参照のこと。

(例)

今日の日付を 2000 年 1 月 1 日とすると、CURRENT-DATE には、01/01/00 が格納されている。

3. TIME-OF-DAY は、暗黙に符号のない 6 けたの整数の数字項目としてこのシステムが確保する記憶場所であり、MOVE 文の送り出し側作用対象として使うことができる。項類は数字である。プログラム実行中のこの MOVE 文を実行した時点の時刻が「HHMMSS (時, 分, 秒)」の形式で格納されている。

(例)

現在の時刻を 19 時 8 分 12 秒とすると、TIME-OF-DAY には、190812 が格納される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

「COBOL2002 言語 標準仕様編」 「11.18 CURRENT-DATE 関数」

「COBOL2002 言語 標準仕様編」 「11.72 WHEN-COMPILED 関数」

25.4.12 CORRESPONDING 指定のある MOVE 文 (受け取り側作用対象を複数指定)

形式

$$\text{MOVE } \left\{ \begin{array}{l} \text{CORRESPONDING} \\ \text{CORR} \end{array} \right\} \\ \text{一意名1 TO (一意名2) } \dots$$

一般規則

1. CORRESPONDING 指定のある MOVE 文には、受け取り側作用対象を複数個記述してもよい。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.6.5 CORRESPONDING 指定」

「COBOL2002 言語 標準仕様編」 「10.8.28 MOVE 文」

25.4.13 RELEASE 文 (定数 1 指定)

形式

$$\text{RELEASE } \text{レコード名1} \\ \left[\text{FROM } \left\{ \begin{array}{l} \text{一意名1} \\ \text{定数1} \end{array} \right\} \right]$$

一般規則

1. FROM 指定のある RELEASE 文の実行結果は、次に示す文をこの順で実行したものと等しい。

(i) MOVE 文の規則に従った文:

$$\text{MOVE} \left\{ \begin{array}{l} \text{一意名1} \\ \text{定数1} \end{array} \right\} \text{TO} \text{レコード名1}$$

(ii) それに続く FROM 指定のない同じ RELEASE 文

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.35 RELEASE 文」

25.4.14 REWRITE 文 (定数 1 指定)

形式

$$\begin{array}{l} \text{REWRITE} \text{レコード名1} \\ \quad \left[\text{FROM} \left\{ \begin{array}{l} \text{一意名1} \\ \text{定数1} \end{array} \right\} \right] \\ \quad \left[\text{INVALID KEY} \text{無条件文1} \right] \\ \quad \left[\text{NOT INVALID KEY} \text{無条件文2} \right] \\ \quad \left[\text{END-REWRITE} \right] \end{array}$$

一般規則

1. FROM 指定のある REWRITE 文の実行結果は、次に示す文をこの順で実行したものと等しい。

(i) MOVE 文の規則に従った文:

$$\text{MOVE} \left\{ \begin{array}{l} \text{一意名1} \\ \text{定数1} \end{array} \right\} \text{TO} \text{レコード名1}$$

(ii) FROM 指定のない同じ REWRITE 文

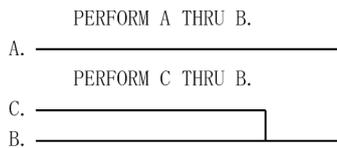
【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.39 REWRITE 文」

25.4.15 PERFORM 文の実行範囲の出口の共有

一つの PERFORM 文の実行範囲の中に、PERFORM 文があり、含まれている PERFORM 文の実行範囲がはじめの PERFORM 文の論理的な実行範囲の内側に完全に含まれている場合、共通の出口を持っているもよい。

(例)



【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.31 PERFORM 文」

25.4.16 TRANSFORM 文

形式

TRANSFORM 一意名1 CHARACTERS
FROM { 一意名2 } TO { 一意名3 }
 { 定数2 } { 定数3 }

機能

TRANSFORM 文は、データ項目中のある文字をほかの文字に変換する。

(備考)

TRANSFORM 文は、COBOL 規格にない。

構文規則

- 一意名 1 は、項目が英字、英数字、数字（外部 10 進だけ）、英数字編集または数字編集の基本項目かまたは強く型付けされていない集団項目を参照するものでなければならない。
- 一意名 2 と一意名 3 は、項類が英字または英数字の基本項目かまたはサイズが可変でない集団項目を参照するものでなければならない。そのサイズは 256 文字以下でなければならない。
- 定数 2 と定数 3 は、英数字定数かまたは ALL 定数以外の表意定数でなければならない。定数 2 や定数 3 に表意定数を指定したときには、そのサイズは 1 文字とみなされる。
- 一意名 3 のデータ項目や定数 3 のサイズは一意名 2 のデータ項目や定数 2 のサイズと同じでなければならない。そうでなければ、定数 3 のサイズは 1 文字でなければならない。
- 一意名 1、一意名 2、一意名 3 は、部分参照してはならない。

一般規則

- 定数 3 のサイズが 1 文字のとき、一意名 1 のデータ項目中の文字で、定数 2 または一意名 2 のデータ項目中に含まれる文字と等しいものは、すべて定数 3 の表す 1 文字に変換される。

(例)

```
TRANSFORM A FROM 'ABCDEFGHI' TO '*'  
TRANSFORM B FROM SPACE TO ZERO.
```

一意名 A 中の文字 A~I のすべてが一つの文字「*」に変換される。また、一意名 B 中の文字「空白」はすべて文字「ゼロ」に変換される。

2. 一般規則 1 以外するとき、一意名 1 のデータ項目中の文字で、定数 2 または一意名 2 のデータ項目中に含まれる文字と等しいものは、定数 3 または一意名 3 のデータ項目中の対応する文字位置にある文字に変換される。

(例)

```
TRANSFORM C FROM '123456789' TO 'ABCDEFGHI'.
```

一意名 C の文字「1」は文字「A」に、文字「2」は文字「B」に、…文字「9」は文字「I」にそれぞれ変換される。

3. 一意名 2 のデータ項目や定数 2 の中に、同じ文字が重複して存在するときの TRANSFORM 文の実行結果は保証しない。

25.4.17 USE 文 (GIVING 指定)

形式

```
USE [GLOBAL] AFTER
STANDARD { EXCEPTION }
          { ERROR }
PROCEDURE ON { {ファイル名1}… }
              { INPUT }
              { OUTPUT }
              { I-O }
              { EXTEND }
[GIVING データ名1 [データ名2]]
```

このシステムでは USE 文の GIVING 指定を覚え書きとみなす。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.53 USE 文」

25.4.18 USE LABEL 文

形式

```
USE { BEFORE } STANDARD
    { AFTER }
[ BEGINNING ] [ REEL ] LABEL PROCEDURE ON { {ファイル名1}… }
[ ENDING ]   [ UNIT ] [ FILE ]           { INPUT }
                                           { OUTPUT }
                                           { I-O }
                                           { EXTEND }
```

構文規則

1. USE 文は、手続き部の宣言部分の節の見出しにすぐ続けて書き、終止符と空白で止める。この後に処理手続きを定義する一つ以上の段落を書く。
2. USE LABEL は、プログラムまたは利用者定義関数の中でだけ使用できる。

一般規則

1. このシステムでは、入出力文から USE LABEL 文に制御が渡ることはない。
USE LABEL 文に制御が渡るのは、PERFORM 文で呼び出した場合だけである。また、GO TO MORE-LABELS 文で制御を戻せる。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.53 USE 文」

25.4.19 WRITE 文（印刷制御文字の直接指定、区域外書き出し）

形式

WRITE 文の拡張仕様は、次の二つの書き方がある。

書き方 1

```
WRITE レコード名1 [FROM 一意名1]
  AFTER POSITIONING
  { 一意名2 } [ LINE ]
  { 整数2 } [ LINES ]
[AT { END-OF-PAGE } 無条件文1]
  { EOP }
[NOT AT { END-OF-PAGE } 無条件文2]
  { EOP }
[END-WRITE]
```

書き方 2

```
WRITE レコード名1 [FROM 一意名1]
  [INVALID KEY 無条件文1]
[END-WRITE]
```

機能

WRITE 文は、出力ファイルや拡張ファイルにレコードを書き出す。

このシステムでは、次の拡張仕様がある。

1. 印刷ページ中の各レコードの行番号を決めるために、印刷制御文字を直接指定できる（書き方 1）。
（備考）

このシステムでは、VOS3 COBOL85 とは異なり、常に AFTER ADVANCING 1 LINE が指定されたものと同じ行送りをする。

2. 記憶媒体上の順ファイルヘレコードを書き出す場合、利用者は、区域外書き出しが発生したとき行う処理を指定できる（書き方 2）。

構文規則

1. 一意名 2 は、長さ 1 文字の英数字基本項目の名前でなければならない。
2. 整数 2 は、符号のない 1 けたの整数で 0, 1, 2, 3 のどれかでなければならない。
3. 書き方 1 の WRITE 文は、順ファイルに対してだけ指定できる。
4. 書き方 2 の WRITE 文を順ファイルに対して指定するとき、NOT INVALID KEY 指定を書いてはならない。
5. LINAGE 句を指定したファイルに、書き方 1 および書き方 2 の WRITE 文を使用してはならない。
6. POSITIONING 指定のある WRITE 文を使用する場合、そのファイルに対するすべての WRITE 文に POSITIONING 指定を書かなければならない。

一般規則

1. POSITIONING 指定のある WRITE 文は、一意名 2 または整数 2 の指定に従って行送りした後、レコードを印刷する。
(備考)
このシステムでは、AFTER ADVANCING 1 LINE が指定されたものと同じ行送りを行う。
2. AT END-OF-PAGE 指定、NOT AT END-OF-PAGE 指定、END-OF-PAGE 指定は、ページの終わりを検出したときの利用者手続きを指定するときに使う。
このファイルは、プリンタファイルでなければならない。
(備考)
このシステムでは、ページの終わりを検出できないので、無条件文 1 が実行されることはない。
3. INVALID KEY 指定
記憶媒体上の順ファイルにファイルの範囲を超えてレコードの書き出しを試みたとき区域外条件が発生し、INVALID KEY 指定があれば、その無条件文に制御が移る。
INVALID KEY 指定がないと、区域外条件が発生した場合、利用者誤り処理手続きの指定があれば、手続きを実行する。利用者誤り処理手続きの指定がなく、FILE STATUS 句の指定がない場合は、異常終了する。
INVALID KEY は大記憶ファイルの WRITE 文に指定しなければならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

25.4.20 WRITE 文 (定数 1 指定)

形式

`WRITE` レコード名1 [`FROM` {一意名1
定数1}]
[`INVALID KEY` 無条件文1]
[`NOT INVALID KEY` 無条件文2]
[`END-WRITE`]

一般規則

1. FROM 指定のある WRITE 文の実行結果は、次に示す文をこの順で実行したものと等しい。

- MOVE 文の規則に従った文:

`MOVE` {一意名1
定数1} `TO` レコード名1

- FROM 指定のない同じ WRITE 文

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.8.55 WRITE 文」

25.5 組み込み関数

ここでは、COBOL プログラムの文字列を文字単位に操作したり、文字エンコーディングを変更したりする組み込み関数について説明する。

なお、ここで説明する組み込み関数には、Unicode 機能、日本語 EUC 環境を前提とするものがある。

Unicode 機能および日本語 EUC 環境については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

#	用途	組み込み関数名	機能概要	参照先
1	文字列の取り出し	SUBSTRING	字類が英字、英数字または日本語のデータ項目から部分文字列を返す。	25.5.7 SUBSTRING 関数
2	文字数の取得	COUNT-CHAR	字類が英字、英数字または日本語のデータ項目の文字数を返す。	25.5.3 COUNT-CHAR 関数
3	文字位置の個数の取得	LENGTH-OF-SUBSTRING	字類が英字、英数字または日本語のデータ項目の部分文字列の文字位置の個数を返す。	25.5.5 LENGTH-OF-SUBSTRING 関数
4	コード変換	DISPLAY-OF ^{*1}	字類が日本語のデータ項目中の UTF-16 の文字列を UTF-8 の文字列に変換して返す。	25.5.4 DISPLAY-OF 関数
5		NATIONAL-OF ^{*1}	字類が英数字のデータ項目中の UTF-8 の文字列を UTF-16 の文字列に変換して返す。	25.5.6 NATIONAL-OF 関数
6		CONVERT-CODE ^{*2}	字類が英字または英数字のデータ項目中の文字列のエンコーディングをキーワードに従って変換した文字列を返す。	25.5.2 CONVERT-CODE 関数

注※1

日本語 EUC 環境では、使用できない。

注※2

Windows COBOL2002 および AIX COBOL2002 では、この関数は使用できない。

25.5.1 用語の定義 (日本語 EUC)

日本語 EUC に対応した組み込み関数を次に示す。

- COUNT-CHAR
- LENGTH-OF-SUBSTRING
- SUBSTRING

これらの日本語 EUC に対応した組み込み関数の説明で使用する用語について次の表に示す。なお、Unicode 機能で使用する用語については、「23.1 用語の定義」を参照のこと。

#	用語	説明
1	日本語 EUC	UNIX 系 OS で標準的に使われる文字コードの 1 つ。ASCII 文字を 1 バイト、半角かなを 2 バイト、日本語文字を 2 バイトや 3 バイトで表現する。
2	文字位置の個数	英数字項目または日本語項目の PICTURE 句に指定した文字位置の個数。 例) <pre>01 DATA1 PIC X(3). *>文字位置の個数は3 01 DATA2 PIC N(5). *>文字位置の個数は5</pre>
3	文字数	見た目の文字数。
4	見た目幅	半角文字を 1、全角文字を 2 とした見た目の幅。

25.5.2 CONVERT-CODE 関数

CONVERT-CODE 関数は、引数に指定された文字列のエンコーディングをキーワードに従って変換した文字列を返す。この関数の型は、英数字である。

Windows COBOL2002 および AIX COBOL2002 では、CONVERT-CODE 関数は使用できない。

(1) 一般形式

FUNCTION CONVERT-CODE (引数1 {
 SJIS-TO-UTF8
 UTF8-TO-SJIS})

(2) 引数

1. 引数 1 は、字類が英字か英数字でなくてはならない。
2. 引数 1 に可変長集団項目を指定してはならない。また、引数 1 を部分参照してはならない。
3. 引数 1 に動的長基本項目を指定してはならない。
4. 引数 1 に続く語が SJIS-TO-UTF8 の場合、引数 1 の文字コードは SJIS でなければならない。また、引数 1 に続く語が UTF8-TO-SJIS の場合、引数 1 の文字コードは UTF-8 でなければならない。
5. 引数 1 に関数一意名およびオブジェクトプロパティは指定できない。
6. 引数 1 が英数字集団項目のとき、その従属項目の字類は英字、または英数字でなければならない。

(3) 関数値

1. 引数 1 に続く語が SJIS-TO-UTF8 の場合、引数 1 のエンコーディングをシフト JIS から UTF-8 に変換した文字列を返す。また、引数 1 に続く語が UTF8-TO-SJIS の場合、引数 1 のエンコーディングを UTF-8 からシフト JIS に変換した文字列を返す。
2. 変換前の文字に対応する変換後の文字を持たない文字が引数 1 中にある場合、文字変換は行わないで、EC-ARGUMENT-IMP 例外条件が成立する。
ただし、-FunctionECSup,CodeConvErr オプションが有効なときは、EC-ARGUMENT-IMP 例外条件は成立しない。-FunctionECSup,CodeConvErr オプションについては、マニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。
3. 関数値の長さは、変換後の文字数を保持するのに必要とする、用途が DISPLAY の文字位置の個数となる。
4. 関数値を転記する受け取り側作用対象に複数の項目を指定してはならない。
5. 関数値の表現形式は、英数字形式となる。

25.5.3 COUNT-CHAR 関数

COUNT-CHAR 関数は、引数 1 に指定された文字列の文字数を返す。この関数の型は、整数とする。

なお、この組み込み関数は、次のどちらかの場合に使用できる。

- -UniObjGen オプションを指定してコンパイル
- 日本語 EUC 環境下でコンパイル

-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

日本語 EUC 環境については、マニュアル「COBOL2002 使用の手引 手引編」の EUC コードを使用した COBOL プログラムを参照のこと。

(1) 一般形式

`FUNCTION COUNT-CHAR (引数1 [WIDTH])`

(2) 引数

1. 引数 1 は、字類が英字か英数字、または日本語でなければならない。
2. 引数 1 に可変長集団項目を指定してはならない。
3. 引数 1 に動的長基本項目を指定してはならない。
4. 引数 1 の文字コードは、次のどちらかでなければならない。

字類	文字コード	
	Unicode 機能	日本語 EUC 環境
英字, 英数字	UTF-8	日本語 EUC
日本語	UTF-16	

5. 引数 1 が英数字集団項目のとき、その従属項目の字類は、英字または英数字でなければならない。

(3) 関数値

1. 関数値は、次のとおりである。

(ア) WIDTH が指定されている場合

関数値は引数 1 の長さを見たい幅で表した整数となる。

(イ) WIDTH が指定されていない場合

関数値は引数 1 の長さを文字数で表した整数となる。

2. 関数値の表現形式は、2 進形式 4 バイト (64bit 版 COBOL2002 では 2 進形式 8 バイト) となる。

(4) 使用例 (Unicode 機能)

(a) シフト JIS での文字数の取得

シフト JIS では、バイト数、文字数、けた数の関係が決まっている。英数字項目は半角 1 文字を 1 バイトで 1 けた、日本語項目は全角 1 文字を 2 バイトで 1 けたである。そのため、文字数を意識したプログラミングは比較的容易である。

例えば、次のように 3 けたの英数字項目に格納されている半角の文字数は 3 文字、3 けたの日本語項目に格納されている文字数は 3 文字である。

```
01 TEST-DATA1 PIC X(10) VALUE 'アイウ'.
*> 3文字
01 TEST-DATAN PIC N(3) VALUE N'あいう'.
*> 3文字
```

次のデータ項目の後ろ空白を除いた文字数を求める文の例を示す。

```
01 TEST-DATA1 PIC X(10) VALUE 'アイウ'.
```

TEST-DATA1の内容

ア	イ	ウ	△	△	△	△	△	△	△
---	---	---	---	---	---	---	---	---	---

(凡例) △ : 半角空白

記述例

```
COMPUTE NUMOFCHAR =  
    FUNCTION LENGTH(FUNCTION TRIM(TEST-DATA1 TRAILING)).  
*> 3が求まる
```

(b) Unicode での文字数の取得

Unicode の場合は、文字数とけた数の関係が定まらないため、けた数から文字数を容易に求めることができない。そこで、引数に指定された Unicode のデータ項目の文字数を返す COUNT-CHAR 関数を使用する。

COUNT-CHAR 関数を使用して、次の UTF-8 のデータ項目の後ろ空白を除いた文字数を求める文の例を示す。

```
01 TEST-DATA1 PIC X(10) VALUE 'アイ'.
```

記述例

```
COMPUTE UNUMOFCHAR =  
    FUNCTION COUNT-CHAR(FUNCTION TRIM(TEST-DATA1 TRAILING)).  
*> 3が求まる
```

(c) Unicode での見た目幅の取得

シフト JIS では、英数字項目に LENGTH 関数を適用すると、けた数が返る。英数字項目では 1 けたは半角 1 文字分となる。また、全角日本語は 2 けたを占めるため、半角 2 文字分となる。

Unicode では、COUNT-CHAR 関数を英数字項目に適用し、引数の WIDTH を指定した場合、半角文字での見た目幅が返却される。COUNT-CHAR 関数を使用することで、英数字項目の場合、シフト JIS と同様に見た目幅で扱うことができるため、既存のシフト JIS を前提としたプログラムからの移行が容易となる。

シフト JIS の場合と Unicode の場合とで、次のデータ項目の見た目幅を取得する文の例を示す。

```
01 TEST-DATA1 PIC X(8) VALUE 'AあC'.  
*> "あ"は、見た目幅2
```

TEST-DATA1の内容

A	あ	C	△	△	△	△
---	---	---	---	---	---	---

(凡例) △ : 半角空白

記述例

- ・シフトJISの場合
COMPUTE NUMOFCHAR =
FUNCTION LENGTH(FUNCTION TRIM(TEST-DATA1 TRAILING)).
*> 4が求まる
- ・Unicodeの場合
COMPUTE UNUMOFCHAR =
FUNCTION COUNT-CHAR(FUNCTION TRIM(TEST-DATA1 TRAILING), WIDTH).
*> 見た目幅として、4が求まる

(5) 使用例 (日本語 EUC 環境)

英数字項目に格納された半角文字の文字数を求める場合、シフト JIS と日本語 EUC では、それぞれ次の例に示すコーディングとなる。

(a) シフト JIS での半角文字の文字数および見た目幅の取得

シフト JIS の場合、半角文字は 1 バイトの固定である。そのため、英数字項目に格納された文字列の文字数と見た目幅は等しくなり、LENGTH 関数で容易に求められる。

(例)

```
01 TEST-DATA1 PIC X(10) VALUE 'アイ'. *> 3文字を格納
:
*> 半角文字の文字数および見た目幅を求める
COMPUTE NUMOFCHAR =
FUNCTION LENGTH(FUNCTION TRIM(TEST-DATA1 TRAILING)).
*> 3が求まる
```

(b) 日本語 EUC での半角文字の文字数および見た目幅の取得

日本語 EUC の場合、半角かたかなが 1 文字 2 バイトとなる。そのため、シフト JIS と同じ方法では見た目幅も文字数も求められない。

日本語 EUC では、LENGTH 関数の代わりに COUNT-CHAR 関数を使用することで文字数を求められる。

(例)

```
01 TEST-DATA1 PIC X(10) VALUE 'アイ'. *> 3文字を格納
:
*> 半角文字の文字数を求める
COMPUTE NUMOFCHAR =
FUNCTION COUNT-CHAR(FUNCTION TRIM(TEST-DATA1 TRAILING)).
*> 3が求まる
```

25.5.4 DISPLAY-OF 関数

DISPLAY-OF 関数は、UTF-16 の文字を UTF-8 の文字に変換した文字列を返す。この関数の型は、英数字である。

なお、この組み込み関数を使用するためには、-UniObjGen オプションの指定が必要である。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

(1) 一般形式

FUNCTION DISPLAY-OF (引数 1)

(2) 引数

1. 引数 1 は、字類が日本語でなくてはならない。
2. 引数 1 を部分参照してはならない。
3. 引数 1 に動的長基本項目を指定してはならない。
4. 引数 1 の文字コードは、UTF-16 でなければならない。
5. 引数 1 に TRIM 関数以外の関数一意名およびオブジェクトプロパティは指定できない。

注意事項

引数 1 に指定する TRIM 関数の引数は、関数一意名であってはならない。

(3) 関数値

1. 引数 1 の UTF-16 の文字を、対応する UTF-8 の文字に変換した文字列を返す。なお、このシステムでは UTF-8 へのコード変換だけとする。
2. 対応する UTF-8 の文字を持たない UTF-16 の文字が引数 1 中にある場合、UTF-8 への文字変換は行わないで、EC-ARGUMENT-IMP 例外条件が成立する。
ただし、-FunctionECSup,CodeConvErr オプションが有効なときは、EC-ARGUMENT-IMP 例外条件は成立しない。-FunctionECSup,CodeConvErr オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。
3. 関数値の長さは、変換後の文字数を保持するのに必要とする、用途が DISPLAY の文字位置の個数となる。
4. 関数値を転記する受け取り側作用対象に複数の項目を指定してはならない。
5. 関数値の表現形式は、英数字形式となる。

(4) 使用例

日本語項目の値を DISPLAY-OF 関数で UTF-8 に文字変換し、英数字項目へ転記する文の例を次に示す。

記述例

```
01 NAT    PIC N(6) VALUE N'あいうえ'.
01 ALNUM  PIC X(100)
          *> 項目長は十分な領域を設定すること。
:
MOVE FUNCTION DISPLAY-OF(NAT) TO ALNUM.          . . . ①
MOVE FUNCTION DISPLAY-OF(FUNCTION TRIM(NAT)) TO ALNUM. . . . ②
```

NATの内容

あ	い	う	え	▲	▲
---	---	---	---	---	---

ALNUMの内容

①

あ	い	う	え	▲	▲	△	△	△	△	△	△	△	△	△	△	△	△	△	△
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

②

あ	い	う	え	△	△	△	△	△	△	△	△	△	△	△	△	△	△	△	△
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(凡例)▲ : 全角空白 △ : 半角空白

右端に全角空白が空白詰めされたデータをDISPLAY-OF関数で英数字項目に変換すると、①のように右端の全角空白もそのままUTF-8に変換される。このような場合、②のようにTRIM関数の引数に指定し、そのTRIM関数をDISPLAY-OF関数に指定すると、全角空白を除いた文字列だけ変換できる。

25.5.5 LENGTH-OF-SUBSTRING 関数

LENGTH-OF-SUBSTRING 関数は、開始位置と長さを指定することによって、データ項目中の文字列から部分文字列の文字位置の個数を返す。開始位置と長さは、見た目幅または文字数で指定できる。この関数の型は、整数とする。

なお、この組み込み関数は、次のどちらかの場合に使用できる。

- -UniObjGen オプションを指定してコンパイル
- 日本語 EUC 環境下でコンパイル

-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

日本語 EUC 環境については、マニュアル「COBOL2002 使用の手引 手引編」の EUC コードを使用した COBOL プログラムを参照のこと。

(1) 一般形式

`FUNCTION LENGTH-OF-SUBSTRING (引数1 引数2 [引数3] [WIDTH])`

(2) 引数

1. 引数 1 は、字類が英字か英数字、または日本語でなければならない。
2. 引数 1 に可変長集団項目を指定してはならない。また、引数 1 を部分参照してはならない。
3. 引数 1 に動的長基本項目を指定してはならない。
4. 引数 1 の文字コードは、次のどちらかでなければならない。

字類	文字コード	
	Unicode 機能	日本語 EUC 環境
英字, 英数字	UTF-8	日本語 EUC
日本語	UTF-16	

5. 引数 2 と引数 3 は、算術式でなければならない。
6. 引数 1, 引数 2 および引数 3 に関数一意名およびオブジェクトプロパティは指定できない。なお、引数 2 および引数 3 については、算術式中も指定できない。
7. 引数 2 および引数 3 は、ゼロでない正の整数でなければならない。
8. 引数 2 および引数 3 に指定する値は次のとおりである。

(ア) 見た目幅で部分文字列の文字位置の個数を返す場合

半角文字 1 文字を 1, 全角文字 1 文字を 2 として数えた値を指定する。このとき、同時に WIDTH も指定すること。

(例)

文字列'a あ c'の見た目幅は 4 となる。この場合、部分文字列'c'の文字位置の個数を返すときは引数 2 に 4, 引数 3 に 1 を指定する。

(イ) 文字数で部分文字列の文字位置の個数を返す場合

半角文字および全角文字の 1 文字を 1 として数えた値を指定する。このとき、WIDTH を指定しないこと。

(例)

文字列'a あ c'の文字数は 3 となる。この場合、部分文字列'c'の文字位置の個数を返すときは引数 2 に 3, 引数 3 に 1 を指定する。

9. 引数 2 で指定する開始位置、または引数 2 で指定する開始位置から引数 3 で指定する長さだけ進めた文字位置は、引数 1 の終端を越えてはならない。
10. 引数 2 で指定する開始位置、または引数 2 で指定する開始位置から引数 3 で指定する長さだけ進めた文字位置は全角文字の途中を指してはならない。
11. 引数 1 が英数字集団項目のとき、その従属項目の字類は英字、または英数字でなければならない。

(3) 関数値

1. 引数 3 が指定されている場合、関数値は、引数 2、引数 3 で指定する開始位置と長さで引数 1 から取り出した部分文字列の、文字位置の個数を返す。このとき、開始位置と長さの単位は次のとおりとする。

(ア) WIDTH が指定されている場合

引数 2、引数 3 の単位は見た目幅として開始位置および長さを評価する。

(イ) WIDTH が指定されていない場合

引数 2、引数 3 の単位は文字数として開始位置および長さを評価する。

2. 引数 3 が指定されていない場合、関数値は、引数 2 で指定する開始位置から引数 1 の終端までの部分文字列の、文字位置の個数を返す。このとき、開始位置の単位は次のとおりとする。

(ア) WIDTH が指定されている場合

引数 2 の単位は見た目幅として開始位置を評価する。

(イ) WIDTH が指定されていない場合

引数 2 の単位は文字数として開始位置を評価する。

3. 関数値の表現形式は、2 進形式 4 バイト (64bit 版 COBOL2002 では 2 進形式 8 バイト) となる。

(4) 使用例 (Unicode 機能)

(a) シフト JIS での部分文字列のけた数の取得

ある文字列の m 文字目から n 文字目を参照したい、または開始位置と長さを指定して文字を書き込みたい場合、シフト JIS では部分参照を使用できる。

次の文字列 NAME-KN の 8 文字目から文字列を転記する文の例を示す。

NAME-KN-RECの内容

LEN	NAME-KN									
7	ヒ	ヲ	チ	△	ヲ	ロ	ウ	△	△	△

(凡例) △ : 半角空白

記述例

```
01 NAME-KN-REC.  
02 NAME-KN PIC X(10) VALUE 'ヒヲチ△ヲロウ△△△'.
```

```
MOVE 'サマ' TO NAME-KN(8:).  
*> 結果 : 'ヒヲチ△ヲサマ'
```

シフト JIS の場合、文字数とけた数が対応しているため、上記のように部分参照のけた数に文字数を指定して処理ができる。

(b) Unicode での部分文字列のけた数の取得

Unicode の場合は、文字数とけた数が対応しないため、シフト JIS と同様のコーディングができない。そこで、文字数をけた数に変換する LENGTH-OF-SUBSTRING 関数を使用する。

LENGTH-OF-SUBSTRING 関数を使用すると、引数に指定した開始位置と長さによって指定した文字列から、該当する文字列が占めるけた数を求めることができるため、その結果によって、部分参照を行うことができる。

英数字項目のときに見た目幅で指定する方法

文字列 NAME-KN の 8 文字目から 2 文字分の文字列を転記する文の例を次に示す。

記述例

```
01 NAME-KN-REC.
02 LEN PIC 9(2) VALUE 7.
02 NAME-KN PIC X(30) VALUE 'ヒヂ ｸﾗｯ'.
COMPUTE NCHARPOS =
    FUNCTION LENGTH-OF-SUBSTRING (NAME-KN, 1, 7, WIDTH).
    *> 19が求まる (半角カタカナ3バイト*6 + 半角空白1バイト)
MOVE 'サ' TO NAME-KN (NCHARPOS + 1 : ).
*> 結果 : 'ヒヂ ｸﾗｯサ'
*> FUNCTION LENGTH-OF-SUBSTRINGの関数値「NCHARPOS」が開始位置に使用できる
```

日本語項目のときに文字数で指定する方法

- シフト JIS の場合

次の文字列 NAME-KN の 6 文字目から文字列を転記する文の例を示す。

NAME-KN-RECの内容

LEN	NAME-KN
5	吉 田 ▲ 太 郎 ▲ ▲ ▲ ▲ ▲

(凡例) ▲ : 全角空白

記述例

```
01 NAME-KN-REC.
02 LEN PIC 9(2) VALUE 5.
02 NAME-KN PIC N(10) VALUE N'吉田 太郎'.

MOVE N'様' TO NAME-KN(6:). *> 結果 : '吉田 太郎様'
```

- Unicode の場合

次の文字列 NAME-KN の 7 文字目から文字列を転記する文の例を示す。

NAME-KN-RECの内容

LEN	NAME-KN
6	吉 ; 田 ▲ 太 郎 ▲ ▲ ▲ ▲

(凡例) ▲ : 全角空白

記述例

```
01 NAME-KN-REC.
02 LEN PIC 9(2) VALUE 6.
02 NAME-KN PIC N(10) .

MOVE NX' D842DFB775303000592A90CE' TO NAME-KN.
*> N'吉田 太郎'を次のMOVE文で転記
*> なお吉は土冠の吉(サロゲートペア文字)

COMPUTE NCHARPOS =
    FUNCTION LENGTH-OF-SUBSTRING (NAME-KN, 1, 5).
    *> 6が求まる
MOVE N'様' TO NAME-KN (NCHARPOS + 1 : ).
*> 結果 : '吉田 太郎様'
*> なお吉は土冠の吉(サロゲートペア文字)
```

(5) 使用例 (日本語 EUC 環境)

半角文字からなる文字列の m 文字目から n 文字目までを参照する、または開始位置と長さを指定して文字を書き込む場合、シフト JIS と日本語 EUC では、それぞれ次の例に示すコーディングとなる。

(a) シフト JIS での部分文字列のけた数の取得

シフト JIS の場合、半角文字が 1 バイト固定であるため、部分参照が使用できる。

(例) 英数字 NAME-KN の 8 文字目から文字列を転記

```
01 NAME-KN PIC X(10) VALUE 'ヒ好 如ウ'.
      :
      MOVE 'サマ' TO NAME-KN(8:).  *> 結果 : 'ヒ好 如ウサマ'
```

(b) 日本語 EUC での部分文字列のけた数の取得

日本語 EUC の場合、文字数とけた数が対応しないため、シフト JIS と同じコーディングができない。そこで、文字数をけた数に変換する LENGTH-OF-SUBSTRING 関数を使用する。

LENGTH-OF-SUBSTRING 関数を使用すると、引数に指定した開始位置と長さによって、指定した文字列から当該文字列が占めるけた数を求めることができる。その結果を使って部分参照できる。

(例) 文字列 NAME-KN の 8 文字目から 2 文字分の文字列を転記

```
01 NAME-KN PIC X(20) VALUE 'ヒ好 如ウ'.
      :
      COMPUTE NCHARPOS =
          FUNCTION LENGTH-OF-SUBSTRING(NAME-KN, 1, 7).
          *> 13が求まる(半角かたかな2バイト*6 + 半角空白1バイト)
      MOVE 'サマ' TO NAME-KN(NCHARPOS + 1 : ).
      *> 結果 : 'ヒ好 如ウサマ'
      *> FUNCTION LENGTH-OF-SUBSTRINGの関数値「NCHARPOS」が
      *> 開始位置に使える
```

25.5.6 NATIONAL-OF 関数

NATIONAL-OF 関数は、UTF-8 の文字を UTF-16 の文字に変換した文字列を返す。この関数の型は、日本語とする。

なお、この組み込み関数を使用するためには、-UniObjGen オプションの指定が必要である。-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

(1) 一般形式

FUNCTION NATIONAL-OF (引数 1)

(2) 引数

1. 引数 1 は、字類が英字か英数字でなくてはならない。
2. 引数 1 に可変長集団項目を指定してはならない。また、引数 1 を部分参照してはならない。
3. 引数 1 に動的長基本項目を指定してはならない。
4. 引数 1 の文字コードは、UTF-8 でなければならない。
5. 引数 1 に TRIM 関数以外の関数一意名およびオブジェクトプロパティは指定できない。

注意事項

引数 1 に指定する TRIM 関数の引数は、関数一意名であってはならない。

6. 引数 1 が英数字集団項目のとき、その従属項目の字類は英字、または英数字でなければならない。

(3) 関数値

1. 引数 1 の UTF-8 の文字データを、対応する UTF-16 の文字に変換した文字列を返す。
なお、このシステムでは UTF-16 へのコード変換だけとする。全角文字変換は行わない。
2. 対応する UTF-16 の文字を持たない英数字文字が引数 1 中にある場合、UTF-16 への文字変換は行わないで、EC-ARGUMENT-IMP 例外条件が成立する。
ただし、-FunctionECSup,CodeConvErr オプションが有効なときは、EC-ARGUMENT-IMP 例外条件は成立しない。-FunctionECSup,CodeConvErr オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。
3. 関数値の長さは、変換後の文字数を保持するのに必要とする、用途が NATIONAL の文字位置の個数となる。
4. 関数値を転記する受け取り側作用対象に複数の項目を指定してはならない。
5. 関数値の表現形式は、日本語形式となる。

(4) 使用例

英数字項目の値を NATIONAL-OF 関数で UTF-16 に文字変換し、日本語項目へ転記する例を次に示す。

記述例

```
01 ALNUM PIC X(6) VALUE 'abcd'.
01 NAT PIC N(50).
   *> 項目長は十分な領域を設定すること。
:
MOVE FUNCTION NATIONAL-OF(ALNUM) TO NAT.           . . . ①
MOVE FUNCTION NATIONAL-OF(FUNCTION TRIM(ALNUM)) TO NAT. . . . ②
```

ALNUMの内容

a	b	c	d	△	△
---	---	---	---	---	---

NATの内容

①	a	b	c	d	△	△	▲	▲	▲	⋯	▲
②	a	b	c	d	▲	▲	▲	▲	▲	⋯	▲

(凡例)▲ : 全角空白 △ : 半角空白

右端に半角空白が空白詰めされたデータをNATIONAL-OF関数で日本語項目に変換すると、①のように右端の半角空白もそのままUTF-16に変換される。このような場合、②のようにTRIM関数の引数に指定し、そのTRIM関数をNATIONAL-OF関数に指定すると、半角空白を除いた文字列だけ変換できる。

25.5.7 SUBSTRING 関数

SUBSTRING 関数は、指定した開始位置と長さによって、指定したデータ項目中の文字列から部分文字列を返す。この関数の型は、引数の型に従って次のとおりとする。

引数の型	関数の型
英字	英数字
英数字	英数字
日本語	日本語

なお、この組み込み関数は、次のどちらかの場合に使用できる。

- -UniObjGen オプションを指定してコンパイル
- 日本語 EUC 環境下でコンパイル

-UniObjGen オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」の Unicode 機能を参照のこと。

日本語 EUC 環境については、マニュアル「COBOL2002 使用の手引 手引編」の EUC コードを使用した COBOL プログラムを参照のこと。

(1) 一般形式

`FUNCTION SUBSTRING (引数1 引数2 [引数3] [WIDTH])`

(2) 引数

1. 引数 1 は、字類が英字か英数字、または日本語でなければならない。

注意事項

-JPN,V3JPN オプション指定時、引数に日本語のデータ項目、日本語の関数一意名、日本語のオブジェクトプロパティおよび日本語文字定数を指定できない。

2. 引数 1 に可変長集団項目を指定してはならない。また、引数 1 を部分参照してはならない。

3. 引数 1 に動的長基本項目を指定してはならない。

4. 引数 1 の文字コードは、次のどちらかでなければならない。

字類	文字コード	
	Unicode 機能	日本語 EUC 環境
英字, 英数字	UTF-8	日本語 EUC
日本語	UTF-16	

5. 引数 2 と引数 3 は、算術式でなければならない。

6. 引数 1, 引数 2, および引数 3 に関数一意名およびオブジェクトプロパティは指定できない。なお、引数 2 および引数 3 については、算術式中も指定できない。

7. 引数 2 および引数 3 は、ゼロでない正の整数でなければならない。

8. 引数 2 および引数 3 に指定する値は次のとおりとする。

(ア) 見た目幅で部分文字列を返す場合

半角文字 1 文字を 1, 全角文字 1 文字を 2 として数えた値を指定する。このとき、同時に WIDTH も指定すること。

(例)

文字列'a あ c'の見た目幅は 4 となる。この場合、部分文字列'c'を取得するときは引数 2 に 4, 引数 3 に 1 を指定する。

(イ) 文字数で部分文字列を返す場合

半角文字および全角文字の 1 文字を 1 として数えた値を指定する。このとき、WIDTH を指定しないこと。

(例)

文字列'a あ c'の文字数は 3 となる。この場合、部分文字列'c'を取得するときは引数 2 に 3, 引数 3 に 1 を指定する。

9. 引数 2 で指定する開始位置、または引数 2 で指定する開始位置から引数 3 で指定する長さだけ進めた文字位置は、引数 1 の終端を越えてはならない。

10. 引数 2 で指定する開始位置、または引数 2 で指定する開始位置から引数 3 で指定する長さだけ進めた文字位置は全角文字の途中を指してはならない。

11. 引数 1 が英数字集団項目のとき、その従属項目の字類は英字、または英数字でなければならない。

(3) 関数値

1. 引数 3 が指定されている場合、関数値は、引数 2、引数 3 で指定する開始位置と長さで引数 1 から取り出した部分文字列を返す。このとき、開始位置と長さの単位は次のとおりとする。

(ア) WIDTH が指定されている場合

引数 2、引数 3 の単位は見た目幅として開始位置および長さを評価する。

(イ) WIDTH が指定されていない場合

引数 2、引数 3 の単位は文字数として開始位置および長さを評価する。

2. 引数 3 が指定されていない場合、関数値は、引数 2 で指定する開始位置から引数 1 の終端までの部分文字列を返す。このとき、開始位置の単位は次のとおりとする。

(ア) WIDTH が指定されている場合

引数 2 の単位は見た目幅として開始位置を評価する。

(イ) WIDTH が指定されていない場合

引数 2 の単位は文字数として開始位置を評価する。

3. 関数値を転記する受け取り側作用対象に複数の項目を指定してはならない。

4. 関数値を受け取る受け取り側のデータ項目が引数 1 のデータ項目と同じである場合、この関数の実行の結果は保証しない。

5. 関数値の表現形式は次のとおり。

(ア) 引数 1 の字類が英字または英数字の場合

部分文字列の長さの英数字形式。

(イ) 引数 1 の字類が日本語の場合

部分文字列の長さの日本語形式。

(4) 使用例 (Unicode 機能)

SUBSTRING 関数を使用して、文字列からある長さの部分文字列を取り出す場合の例について説明する。

英数字項目のときに見た目幅で指定する方法

- シフト JIS の場合

次の文字列 FROMSTR-X の 11 文字目から 10 文字を取り出す文の例を示す。

FROMSTR-Xの内容

カ	ナ	カ	ワ	ケ	ン	△	△	△	神	奈	川	県	△	△
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(凡例) △ : 半角空白

記述例

```
01 FROMSTR-X PIC X(100) VALUE 'かが'ワ'ン△△△神奈川県△△' .
   *>神奈川県は、見た目幅2
01 TOSTR-X PIC X(20) .

MOVE FROMSTR-X(11 : 10) TO TOSTR-X.
   *> 部分参照で、'神奈川県△△'がTOSTR-Xに取得できる
```

- Unicode の場合

見た目幅で指定できる SUBSTRING 関数を使用する。SUBSTRING 関数を使用して、Unicode の文字列 FROMSTR-X の 11 文字目から 10 文字を取り出す文の例を示す。

記述例

```
MOVE FUNCTION SUBSTRING(FROMSTR-X, 11, 10, WIDTH) TO TOSTR-X.
   *> SUBSTRING関数で、'神奈川県△△'がTOSTR-Xに取得できる
```

日本語項目のときに文字数で指定する方法

- シフト JIS の場合

次の文字列 FROMSTR-N の 4 文字目から 3 文字を取り出す文の例を示す。

FROMSTR-Nの内容

□	□	区	吉	田	町	〇	〇	〇	番	地
---	---	---	---	---	---	---	---	---	---	---

記述例

```
01 FROMSTR-N PIC N(50) VALUE '□□区吉田町〇〇〇番地' .
01 TOSTR-N PIC N(10) .
```

```
MOVE FROMSTR-N(4 : 3) TO TOSTR-N.
   *> 部分参照で、'吉田町'がTOSTR-Nに取得できる
```

- Unicode の場合

上記の例が Unicode の場合に、文字数で指定できる SUBSTRING 関数を使用して、次の文字列 FROMSTR-N の 4 文字目から 3 文字を取り出す例を示す。

FROMSTR-Nの内容

□	□	区	吉	田	町	〇	〇	〇	番	地
---	---	---	---	---	---	---	---	---	---	---

記述例

```
01 FROMSTR-N PIC N(50) .
01 TOSTR-N PIC N(10) .
```

```
MOVE NX'25A125A1533AD842DFB77530753AFF10FF10FF10756A5730'
   TO FROMSTR-N.
```

*> N'□□区吉田町〇〇〇番地'をMOVE文で転記

*> なお吉は土冠の吉(サロゲートペア文字)

```
MOVE FUNCTION SUBSTRING(FROMSTR-N, 4, 3) TO TOSTR-N.
```

*> SUBSTRING関数で、'吉田町'がTOSTR-Nに取得できる

*> なお吉は土冠の吉(サロゲートペア文字)

(5) 使用例 (日本語 EUC 環境)

英数字項目から、ある長さの部分文字列を取り出す場合、シフト JIS と日本語 EUC では、それぞれ次の例に示すコーディングとなる。

(a) シフト JIS での部分文字列の取り出し

シフト JIS の場合、m 文字目以降から n 文字を取り出す場合、部分参照が使用できる。

(例) 文字列 FROMSTR-X の 11 文字目から 10 文字取り出し

```
01 FROMSTR-X PIC X(100) VALUE 'かがわん△△△神奈川県△△'.
    *> △：半角空白
    *> 神奈川県は、見在目幅2

01 TOSTR-X PIC X(20).
    :
    MOVE FROMSTR-X(11 : 10) TO TOSTR-X.
    *> 部分参照で、'神奈川県△△'がTOSTR-Xに取得できる
```

(b) 日本語 EUC での部分文字列の取り出し

日本語 EUC の場合、m 文字目以降の開始位置が固定ではないため、シフト JIS と同じコーディングはできない。そこで、文字数または見在目幅の指定で部分文字列を取り出す SUBSTRING 関数を使用する。

(例) 文字列 FROMSTR-X の 11 文字目から 10 文字取り出し

```
01 FROMSTR-X PIC X(100) VALUE 'かがわん△△△神奈川県△△'.
    *> △：半角空白
    *> 神奈川県は、見在目幅2

    :
    MOVE FUNCTION SUBSTRING(FROMSTR-X, 11, 10, WIDTH) TO TOSTR-X.
    *> SUBSTRING関数で、'神奈川県△△'がTOSTR-Xに取得できる
```

(6) 注意事項

次に示す手続き文の一意名および引数に SUBSTRING 関数を指定してはならない。

文/組み込み関数	該当する一意名
DISPLAY 文	[書き方 3] DISPLAY 一意名 5 ~
UNSTRING 文	UNSTRING 一意名 1 ~
TRANSFORM 文	TRANSFORM 一意名 1* FROM 一意名 2 TO 一意名 3
TRIM 関数	FUNCTION TRIM(引数 1)

注※

一意名 1 に関数一意名は指定できない。

25.6 2進項目の拡張

25.6.1 1バイト2進機能

機能

1バイトの2進項目を定義できる。

一般規則

1.1バイト2進機能は、-Bin1Byte オプション指定時だけ有効とする。

2.1バイト2進機能と2進項目のUSAGE句の規則を次に示す。

2進項目のUSAGE句	有効かどうか
BINARY	○
COMPUTATIONAL (COMP)	○
COMPUTATIONAL-4 (COMP-4)	○
COMPUTATIONAL-5 (COMP-5)	○
COMPUTATIONAL-X (COMP-X)	—

(凡例)

○：-Bin1Byte オプションを指定したときは、1バイト2進機能が有効になる。

—：-Bin1Byte オプションの影響を受けない。

3.-Bin1Byte オプションを指定したときの PICTURE 句の'9'の数と割り当てられる記憶域の規則を次に示す。

PICTURE 句の'9'の数	割り当てられる記憶域のバイト数
1～2	1
3～4	2
5～9	4
10～18	8

4.-Bin1Byte オプションを指定したときは、2進項目に対する SYNCHRONIZED 句を無視する。

5.SORT 文と MERGE 文のキーに1バイトの2進項目を指定してはならない。

6.索引ファイルの主レコードキー、副レコードキーおよび合成キーを構成するデータ項目に1バイトの2進項目を指定してはならない。

25.6.2 COMP-X 項目

COMP-X 項目は、符号なし 2 進項目を扱うものであり、2 進項目 (BINARY 項目) が書けるところならば、COMP-X 項目を書ける。ただし、機能によっては COMP-X 項目を 2 進項目として扱う場合がある (「25.6.2 COMP-X 項目」の「(6) COMP-X 項目が 2 進項目として扱われる個所」参照)。

(1) 割り当てられる記憶域

COMP-X 項目に対する PICTURE 文字列中の記号 ('9', 'A'および'X') の個数とその項目に割り当てられる記憶域のサイズの規則に従う。その規則を次に示す。

なお、COMP-X 項目は、-Bin1Byte オプション (1 バイト 2 進機能) 指定の有無の影響を受けない。

PICTURE 記号が'9'の場合		PICTURE 記号が'A'または'X'の場合		
PICTURE 句の'9'の個数	割り当てられる記憶域のバイト数	PICTURE 句の'A'や'X'の個数	割り当てられる記憶域のバイト数	仮定される'9'の個数
1~2	1	1	1	2
3~4	2	2	2	4
5~9	4	4	4	9
10~18	8	8	8	18

(2) 格納できる数値

格納できる数値は、割り当てられた記憶域に収まる 2 進数の最大値が限度となる。

なお、COMP-X 項目は、-DigitsTrunc オプション (2 進項目のけた落とし) の影響を受けない。

(3) 記憶方式

2 進データ項目と同じ記憶方式に従って記憶される。システムのバイナリ形式のデータがリトルエンディアンの場合、-BigEndian, Bin オプションの有無によって記憶方式を切り替える。

-BigEndian, Bin オプションあり：ビッグエンディアン形式で記憶。

-BigEndian, Bin オプションなし：リトルエンディアン形式で記憶。

-BigEndian, Bin オプションについては、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL2002 使用の手引 手引編」のコンパイラオプションを参照のこと。

(4) データ部

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9. データ部 (DATA DIVISION)」

(a) PICTURE 句 (COMP-X 項目)

一般規則

1. PICTURE 文字列を構成する記号は、'9'、'X'、'A'でなければならない。
2. 'X'と'A'は同じ意味として扱う。
3. PICTURE 文字列中に記号の混在を許し、記号が'X'だけから成る文字列と同様に扱う。
4. PICTURE 文字列が記号'9'だけから成る場合、1~18 けたでなければならない。
5. PICTURE 文字列中の記号がすべて'X'、'A'または PICTURE 記号が混在する場合、PICTURE 文字列中の記号の個数は 1, 2, 4, 8 個でなければならない。
6. PICTURE 文字列が記号'X'、記号'A'から成る場合、割り当てられる記憶域と同じバイト数になるように記号'9'が指定されたものとしてけた数を決める。その規則を次に示す。

'A'、'X'の個数	バイト数	仮定される'9'の数
1	1	2
2	2	4
4	4	9
8	8	18

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.54 PICTURE 句」

(b) SYNCHRONIZED 句 (COMP-X 項目)

COMP-X 項目は、SYNCHRONIZED 句の影響を受けないため、直前の項目に続けて配置される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.80 SYNCHRONIZED 句」

(c) USAGE 句 (COMP-X 項目)

形式

$$[\text{USAGE IS}] \left\{ \begin{array}{l} \text{COMPUTATIONAL-X} \\ \text{COMP-X} \end{array} \right\}$$

構文規則

1. COMPUTATIONAL-X と COMP-X は同義語であり、どちらを書いてもよい。
2. 基本項目の PICTURE 句は、それが属する集団項目の USAGE 句と矛盾してはならない。
3. 集団項目の USAGE 句に COMP-X を書いた場合、その集団項目に従属する基本項目または集団項目のデータ記述項中に USAGE 句を書くときは、COMP-X でなければならない。

一般規則

1. 集団項目に COMP-X を書くと、その句は集団に従属する基本項目に適用される。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.86 USAGE 句」

(d) VALUE 句 (COMP-X 項目)

構文規則

1. 定数は符号があってはならない。
2. 定数は小数けたがあってはならない。
3. 定数はデータ項目内に収まるものでなければならず、ゼロ以外の切り捨てを引き起こしてはならない。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「9.16.88 VALUE 句」

(5) 手続き部

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10. 手続き部 (PROCEDURE DIVISION)」

(a) 転記の規則

・ COMP-X 項目から COMP-X 項目へ転記するときの規則

送り出し側作用対象の 2 進値がそのまま受け取り側作用対象に格納される。その場合、送り出し側作用対象の 2 進値が受け取り側作用対象に収まらないときは、受け取り側作用対象の範囲外となる送り出し側作用対象の 2 進上位けたを切り捨てる。

(例 1)

```
01 A PIC 9(9) COMP-X VALUE 65538. ... (1)
01 B PIC 9(4) COMP-X. ... (2)
MOVE A TO B. ... (3)
```

(1)は 4 バイト (00010002)₁₆ である。

(2)は 2 バイトである。

(3)の B には (0002)₁₆ が入る。

・ COMP-X 項目以外の項目から COMP-X 項目へ転記するときの規則

・ 字類が数字となる項目から COMP-X 項目へ転記するときの規則

字類が数字となる項目から COMP-X 項目へ転記するときは、送り出し側作用対象の整数部の値を受け取り側作用対象と同じサイズの符号ありの 2 進値に変換した値を符号ビットも数値の一部として受け取り側作用対象に格納する。

送り出し側作用対象の2進値が受け取り側作用対象と同じサイズの符号ありの2進領域に収まらないときは、2進上位けたを切り捨てた値を受け取り側作用対象に格納する。

(例2)

```
01 A PIC 99V99 VALUE 12.34.  
01 B PIC 9(5) VALUE 65537.  
01 C PIC 9(4) COMP-X.  
    MOVE A TO C.  ... (1)  
    MOVE B TO C.  ... (2)
```

(1)のCには(000C)₁₆が入る。

(2)のCには(0001)₁₆が入る。

・字類が英数字となる項目からCOMP-X項目へ転記するときの規則

1. 集団項目からCOMP-X項目へ転記するときは、送り出し側作用対象の左端バイトが受け取り側作用対象の左端バイトに格納され、送り出し側作用対象の右端バイトまで繰り返される。送り出し側作用対象が受け取り側作用対象より大きいときは、格納できない部分を切り捨てる。また、送り出し側作用対象が受け取り側作用対象より小さいときは、空白が受け取り側作用対象に格納される。
2. 英数字項目からCOMP-X項目へ転記するときは、英数字項目から10進値を作り、その10進値を字類が数字となる項目からCOMP-X項目へ転記するときの規則に従って受け取り側作用対象に格納する。
3. 数字編集項目からCOMP-X項目へ転記するときは、数字編集項目の値は一時的に外部10進項目に転記され、その外部10進項目からCOMP-X項目から字類が数字となる項目からCOMP-X項目へ転記するときの規則に従ってCOMP-X項目に格納される。

・COMP-X項目からCOMP-X項目以外の項目へ転記するときの規則

COMP-X項目からCOMP-X項目以外の項目へ転記するときは、COMP-X項目の値を受け取り側作用対象の属性に合わせて格納する。そのとき、小数点の位置あわせおよび必要なゼロ詰めが標準けた寄せ規則に従って行われる。

(例3)

```
01 A PIC 9(4) COMP-X VALUE 65535.  
01 B PIC 9(4) COMP.  
01 C PIC 9(4).  
    MOVE A TO B.  ... (1)  
    MOVE A TO C.  ... (2)
```

(1)のBには-DigitsTrunc オプション指定がないときは-1が入り、-DigitsTrunc オプション指定があるときは5535が入る。

(2)のCには5535が入る。

【標準仕様との関連】

「COBOL2002 言語 標準仕様編」 「10.5.7 データ項目内でのデータのけた寄せ」

(b) 転記の組み合わせ

COMP-X 項目の転記できる組み合わせは、2 進項目の転記の組み合わせと同じとする。

(c) ON SIZE ERROR 指定 (COMP-X 項目)

該当する項目が算術演算の受け取り側作用対象のときに、その文の中に ON SIZE ERROR または NOT ON SIZE ERROR 指定が含まれるとき、PICTURE 文字列の'9'の数によって、けたあふれ条件が生じたかどうか判定される。けたあふれが発生しないときだけ、演算結果が受け取り側作用対象に収められる。

ON SIZE ERROR または NOT ON SIZE ERROR 指定が含まれない場合、受け取り側作用対象の項目の範囲外になる 2 進上位けたを切り捨てて格納される。

(d) 演算の中間結果

• 演算式中の COMP-X 項目

COMP-X 項目が演算式中にあるときは、2 進項目に拡張して演算する。2 進項目に拡張するときの規則を次に示す。

COMP-X 項目のけた数	2 進項目のけた数
1~ 2	3
3~ 4	5
5~ 9	10
10~18	20

• 中間結果のけた数

演算式中に COMP-X 項目を含むときは、COMP-X 項目を (a) の規則に従って符号付きの 2 進項目に拡張したけた数を使用して中間結果のけた数を求める。

• 中間結果が 30 けたを超えたときの補正值

演算式中に COMP-X 項目を含むときに中間結果が 30 けたを超えたときは、COMP-X 項目を (a) の規則に従って、符号付きの 2 進項目に拡張したけた数を使用して 30 けたを超えたときの補正值を計算する。

(e) 比較条件

• 数字作用対象の比較

COMP-X 項目と数字項目を比較するときは、COMP-X 項目を符号付きの 2 進項目に拡張して数字比較をする。2 進項目に拡張するときの規則については、「(d) 演算の中間結果」を参照のこと。

• 文字作用対象の比較

COMP-X 項目と集団項目を比較するときは、文字比較をする。

(f) 比較の組み合わせ

COMP-X の比較の組み合わせは、2 進項目の比較の組み合わせと同じとする。

(g) DISPLAY 文 (COMP-X 項目)

DISPLAY 文のデータ項目（作用対象）とハードウェア装置の間のデータ変換の規則を次に示す。

作用対象	データ変換	項目のサイズ
COMP-X 項目	LEADING SEPARATE 指定の外部 10 進項目に変換。	割り当てられる記憶域に格納できるけた数を持つ LEADING SEPARATE 指定の外部 10 進項目のサイズ。ただし、8 バイトの COMP-X のときは 18 けたとする。

(6) COMP-X 項目が 2 進項目として扱われる箇所

次に示す個所に COMP-X 項目を指定した場合、2 進項目として扱われる。

- データ部
 - 相対ファイルの相対キー
 - 索引ファイルの主レコードキー、副レコードキー、合成キー中のデータ項目

(7) 整列併合機能と COMP-X 項目

- SORT 文と MERGE 文のキーに COMP-X 項目を書いたときは、符号なしのデータとして大小比較する。
- SORT 文と MERGE 文のキーに COMP-X 項目を書くときは、2, 4, 8 バイトの COMP-X 項目でなければならない。

(8) データベースアクセス機能と COMP-X 項目

埋め込み変数のデータ項目に COMP-X 項目を書いたときは、そのまま符号なしデータとして扱う。

(9) 索引ファイルと COMP-X 項目

索引ファイルの主レコードキーと副レコードキーおよび合成キーを構成するデータ項目に COMP-X 項目を書くときは、2, 4 バイトの COMP-X 項目でなければならない。

付録

付録 A Windows COBOL2002 と UNIX COBOL2002 との言語仕様の相違

Windows COBOL2002 と UNIX COBOL2002 との言語仕様の相違の記述個所を一覧にして次に示す。

表 A-1 Windows COBOL2002 と UNIX COBOL2002 との言語仕様の相違

項目	本文中の記述
OLE メソッド OLE オブジェクト OLE プロパティ	[1.2.2 文字列 (日本語機能)] の [(1) 利用者定義語]
テキスト編成	[2.1 テキスト編成]
テキストファイルでのファイル共用に関する指定	[2.2.1 ファイル管理段落 (テキストファイル機能)] [3.1 環境部 (ファイル共用機能)] [3.1.1 ファイル記述項 (ファイル共用機能)]
レコード施錠に関する指定	[2.2.1 ファイル管理段落 (テキストファイル機能)] [3.1 環境部 (ファイル共用機能)] [3.1.1 ファイル記述項 (ファイル共用機能)]
DATA FORMAT 句	[4.3.1 DATA FORMAT 句 (索引ファイル機能の拡張)]
データベースアクセス機能	[9.1 概説 (データベースアクセス機能)]
CRT STATUS 句	[12.1.4 CRT STATUS 句 (SCREEN SECTION)]
WINDOW-KEYCODE 特殊レジスタ	[13.2.1 画面入出力機能特殊レジスタ] の [(3) WINDOW-KEYCODE 特殊レジスタ (WINDOW SECTION)]
COLUMN NUMBER 句	[13.4.5 COLUMN NUMBER 句 (WINDOW SECTION)]
LINE NUMBER 句	[13.4.14 LINE NUMBER 句 (WINDOW SECTION)]
REPEAT COLUMN 句	[13.4.16 REPEAT COLUMN 句 (WINDOW SECTION)]
REPEAT LINE 句	[13.4.17 REPEAT LINE 句 (WINDOW SECTION)]
CHARACTER TYPE 句	[14.2.1 データ記述項 (書式印刷機能)] の [(1) CHARACTER TYPE 句] [17.2.2 データ部 (CSV ファイル入出力機能)] の [(1) ファイル記述項 (FD)] [17.2.3 手続き部 (CSV ファイル入出力機能)] の [(4) WRITE 文 (CSV ファイル入出力機能)]
ADDR 関数	[15.4.3 アドレス型関数 (アドレス操作機能)] の [(1) ADDR 関数]
OLE2 オートメーションインタフェース機能	[18.1 概説 (OLE2 オートメーションインタフェース機能)]
DATA FORMAT 句	[20.3.1 ファイル記述項 (FD) (HiRDB による索引ファイル入出力機能)]
外部プログラム節	[25.2.1 プログラム間連絡機能の環境部] の [(2) 外部プログラム節 (EXTERNAL-PROGRAM SECTION)]

項目	本文中の記述
APPLY FILE-SHARE 句	[25.2.3 入出力管理記述項] の [(3) APPLY FILE-SHARE 句]
COMP-X 項目	[25.6.2 COMP-X 項目] の [(3) 記憶方式]

付録 B LANGOPT=(D)オプションと-JPN,V3JPNSpace オプションを指定した場合の仕様の相違

ここでは、VOS3 COBOL85 で LANGOPT=(D)オプションを指定した場合と、COBOL2002 で -JPN,V3JPNSpace オプションを指定した場合の仕様の相違点について説明する。

なお、-JPN,V3JPNSpace オプションについては、マニュアル「COBOL2002 ユーザーズガイド」または「COBOL2002 使用の手引 手引編」を参照のこと。

付録 B.1 VOS3 COBOL85 と COBOL2002 での仕様の相違

VOS3 COBOL85 と COBOL2002 での仕様の相違について、次の一覧で示す。

表 B-1 VOS3 COBOL85 と COBOL2002 の仕様の比較

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
表意定数 SPACE(SPACES)	—	日本語項目および日本語編集項目に対する表意定数 SPACE(SPACES)は、KEIS コードの空白(X'A1A1')となる。	日本語項目および日本語編集項目に対する表意定数 SPACE(SPACES)は、日本語空白(X'8140')となる。
日本語定数	カナ漢字定数	漢字文字列に変換できなかった場合、KEIS コードの空白(X'A1A1')が仮定される。	このシステムではカナ漢字定数、KEIS 定数、片仮名定数および、平仮名定数は未サポート。
	KEIS 定数		
	片仮名定数		
	平仮名定数		
分離符	—	分離符として、日本語空白を用いることができる。	VOS3 COBOL85 と同じ。
		日本語文字集合中では、EBCDIK コードの空白(X'40')と日本語空白だけを分離符とする。	日本語文字集合中では、半角空白(X'20')と日本語空白だけを分離符とする。
		日本語空白が分離符とみなされるのは、日本語文字集合中だけである。	VOS3 COBOL85 と同じ。
		日本語空白を表す 16 進定数は X'4040', X'A1A1'のどちらでもよい。	日本語空白を表す 16 進定数は X'2020', X'8140'のどちらでもよい。
条件式 (比較条件)	両方の作用対象が日本語の場合	日本語項目、日本語編集項目および日本語文字定数は、それぞれ英数字項目、英数字編集項目および文字定数と同様の文字作用対象の比較の規則に従って比較される。	VOS3 COBOL85 と同じ。
		作用対象のけた数が等しくない場合、短い方の作用対象の右側に、長い方の作用対象	作用対象のけた数が等しくない場合、短い方の作用対象の右側に、長い方の

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		のけた数に等しくなるまで、KEIS コードの空白(X'A1A1')があるものとみなして、作用対象のけた数が等しいときの規則に従って比較される。	作用対象のけた数に等しくなるまで、日本語空白(X'8140')があるものとみなして、作用対象のけた数が等しいときの規則に従って比較される。
	作用対象の一方の字類が日本語で、他方の字類が日本語でない場合	作用対象の一方は表意定数および集団項目でなければならない。	VOS3 COBOL85 と同じ。
		比較は、日本語作用対象を文字作用対象とみなした標準データ形式の文字位置に換算し、この文字位置の個数の大きさの文字作用対象として行われる。 なお、作用対象のけた数が等しくない場合、短い方の作用対象の右側に、長い方の作用対象のけた数に等しくなるまで EBCDIK コードの空白(X'40')があるものとみなして、作用対象のけた数が等しいときの規則に従って行われる。	比較は、日本語作用対象を文字作用対象とみなした標準データ形式の文字位置に換算し、この文字位置の個数の大きさの文字作用対象として行われる。 なお、作用対象のけた数が等しくない場合、短い方の作用対象の右側に、長い方の作用対象のけた数に等しくなるまで半角空白(X'20')があるものとみなして、作用対象のけた数が等しいときの規則に従って行われる。
		作用対象の一方に表意定数 HIGH-VALUE, LOW-VALUE および ZERO を指定してはならない。	一方に表意定数 HIGH-VALUE, LOW-VALUE および ZERO を指定してもよい。指定した場合の扱いは、次のとおりとなる。※1 <ul style="list-style-type: none"> • 表意定数 HIGH-VALUE : 2 バイトの X'FFFF'として扱う。 • 表意定数 LOW-VALUE : 2 バイトの X'0000'として扱う。 • 表意定数 ZERO : 全角文字のゼロ(X'824F')として扱う。
条件式 (字類条件)	数字 (numeric) 検査	日本語項目および日本語編集項目を使用してはならない。	VOS3 COBOL85 と同じ。
	英字 (alphabetic) 検査	日本語項目および日本語編集項目を使用してはならない。	VOS3 COBOL85 と同じ。
	英小文字 (alphabetic-lower) 検査	日本語項目および日本語編集項目を使用してはならない。	VOS3 COBOL85 と同じ。
	英大文字 (alphabetic-upper) 検査	日本語項目および日本語編集項目を使用してはならない。	VOS3 COBOL85 と同じ。
	ブール (boolean) 検査	日本語項目および日本語編集項目を使用してはならない。	VOS3 COBOL85 と同じ。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
	字類名の検査	日本語項目および日本語編集項目を使用してはならない。	VOS3 COBOL85 と同じ。
	実行用計算機段落の PROGRAM COLLATING SEQUENCE 句	関連する符号系名に STANDARD-1 および STANDARD-2 を指定してはならない。	VOS3 COBOL85 と同じ。
	入出力管理段落の APPLY FORMS-OVERLAY 句	データ名に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	ALTERNATE RECORD KEY 句	データ名 1 が日本語項目の場合、データ名 1 に指定された日本語項目を英数字項目として扱う。	VOS3 COBOL85 と同じ。
	FILE STATUS 句	データ名 1 またはデータ名 2 に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	RECORD KEY 句	データ名 1 が日本語項目の場合、データ名 1 に指定された日本語項目を英数字項目として扱う。	VOS3 COBOL85 と同じ。
	JUSTIFIED 句	JUSTIFIED 句が書かれている日本語項目および日本語編集項目を受け取り側作用対象に指定したとき、送り出し側作用対象の日本語項目および日本語編集項目の大きさが受け取り側作用対象の大きさより大きければ、左端から 1 文字単位に文字を切り捨て、小さければ左端の余りに KEIS コードの空白(X'A1A1')を補って、右端をそろえてデータを収める。	JUSTIFIED 句が書かれている日本語項目および日本語編集項目を受け取り側作用対象に指定したとき、送り出し側作用対象の日本語項目および日本語編集項目の大きさが受け取り側作用対象の大きさより大きければ、左端から 1 文字単位に文字を切り捨て、小さければ左端の余りに日本語空白(X'8140')を補って、右端をそろえてデータを収める。
	PICTURE 句 (日本語編集項目)	文字「N」は、日本語文字 1 文字を示し、「B」は、日本語空白 1 文字を示す。それぞれ標準データ形式で 2 文字分 (2 バイト) の大きさを表す。	VOS3 COBOL85 と同じ。
		編集は、MOVE 文でこの項目転記するときに行われる。文字「B」の部分には、KEIS コードの空白(X'A1A1')が挿入される。	編集は、MOVE 文でこの項目転記するときに行われる。文字「B」の部分には、日本語空白(X'8140')が挿入される。
	VALUE 句	定数 2 に日本語文字定数を指定した場合、THRU は指定できない。	VOS3 COBOL85 と同じ。
		条件変数のデータ項目が日本語項目または日本語編集項目の場合、定数 2 は日本語文字定数または表意定数 SPACE でなければならない。定数が日本語文字定数および	VOS3 COBOL85 と同じ。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		表意定数 SPACE 以外の場合、VALUE 句の指定は無視される。	
		条件変数のデータ項目が英数字項目または英数字編集項目の場合、定数 2 および定数 3 は、文字定数または表意定数でなければならない。定数が文字定数および表意定数以外の場合、VALUE 句の指定は無視される。	VOS3 COBOL85 と同じ。
ACCEPT 文	書き方 1	一意名 1 に日本語項目または日本語編集項目を指定した場合、それぞれ英数字項目または英数字編集項目として扱う。	一意名 1 に日本語項目または日本語編集項目を指定した場合、それぞれ英数字項目または英数字編集項目として扱う。 日本語編集項目が指定された場合は、そのまま扱う。
	書き方 2	一意名 2 に日本語項目および日本語編集項目を指定してはならない。	VOS3 COBOL85 と同じ。
CALL 文	—	一意名 1 に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
		定数 1 に日本語文字定数を指定してはならない。	VOS3 COBOL85 と同じ。
CANCEL 文	—	一意名 1 に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
		定数 1 に日本語文字定数を指定してはならない。指定された場合、CANCEL 文を無視する。	VOS3 COBOL85 と同じ。
EVALUATE 文	—	選択対象の組中の各選択対象は、次の規則に従って、選択主体の組中の同じ順序位置にある選択主体に対応していなければならない。 選択対象中に現れる一意名、定数または算術式は、選択主体の組中の対応する作用対象との比較ができるものでなければならない。なお、一意名および定数に日本語項目、日本語編集項目および日本語文字定数を指定したときの比較は、この表の VOS3 COBOL85 の「条件式 (比較条件)」の規則に従う。	VOS3 COBOL85 と同じ。 なお、一意名および定数に日本語項目、日本語編集項目および日本語文字定数を指定したときの比較は、この表の COBOL2002 の「条件式 (比較条件)」の規則に従う。
INITIALIZE 文	—	REPLACING 指定を書かないと、日本語項目および日本語編集項目には空白 (X'A1A1') が設定される。この場合、初期化される各データ項目を受け取り側作用対	REPLACING 指定を書かないと、日本語項目および日本語編集項目には日本語空白 (X'8140') が設定される。この場合、初期化される各データ項目を

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		象とし、空白を送り出し側作用対象の定数とする基本項目間の MOVE 文であるかのように処理が行われる。 なお、MOVE 文の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。	受け取り側作用対象とし、空白を送り出し側作用対象の定数とする基本項目間の MOVE 文であるかのように処理が行われる。 なお、MOVE 文の規則は、この表の COBOL2002 の「MOVE 文」に従う。
		REPLACING 指定を書いた場合、日本語項目および日本語編集項目は初期化されない。	VOS3 COBOL85 と同じ。
INSPECT 文	-	一意名 1 に日本語項目または日本語編集項目を指定したとき、一意名 3 から一意名 7 までの一意名は、日本語項目または日本語編集項目でなければならない。一意名 3 から一意名 7 までのいずれかの一意名に日本語項目または日本語編集項目以外を指定した場合には、INSPECT 文の指定は無視される。	VOS3 COBOL85 と同じ。
		一意名 1 の用途が明示的または暗黙的な DISPLAY（日本語項目または日本語編集項目を除く）であるならば、定数はすべて文字定数または表意定数でなければならない。	VOS3 COBOL85 と同じ。
		一意名 1 の用途が明示的または暗黙的な DISPLAY（日本語項目または日本語編集項目を除く）であるならば、一意名 2 を除くすべての一意名の用途は、明示的または暗黙的な DISPLAY（日本語項目または日本語編集項目を除く）でなければならない。	VOS3 COBOL85 と同じ。
		一意名 1 に日本語項目または日本語編集項目を指定したとき、定数はすべて日本語文字定数でなければならない。また、定数は ALL 指定のない表意定数 SPACE(SPACES)でなければならない。	VOS3 COBOL85 と同じ。
		定数 3 または一意名 5 のデータ項目の日本語文字数は、定数 1 または一意名 3 のデータ項目の日本語文字数と同じでなければならない。定数 3 に表意定数を書くと、表意定数の日本語文字数は、定数 1 や一意名 3 のデータ項目の日本語文字数に等しいものとみなされる。	VOS3 COBOL85 と同じ。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		一意名 1 に日本語項目または日本語編集項目を指定したとき、定数に表意定数 SPACE(SPACES)を書くと、KEIS コードの空白(X'A1A1')として扱う。	一意名 1 に日本語項目または日本語編集項目を指定したとき、定数に表意定数 SPACE(SPACES)を書くと、日本語空白(X'8140')として扱う。
MERGE 文	-	COLLATING SEQUENCE 句に指定した符号系名が、STANDARD-1 指定または STANDARD-2 指定の場合、MERGE 文を無視する。	COLLATING SEQUENCE 句に指定した符号系名に、STANDARD-1 指定または STANDARD-2 を指定してもよい。
		ファイル名 1 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 2 またはファイル名 3 のファイルのレコードは、ファイル名 1 のファイルに引き渡される時、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が半角空白(X'40')で埋められる。	ファイル名 1 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 2 またはファイル名 3 のファイルのレコードは、ファイル名 1 のファイルに引き渡される時、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が半角空白(X'20')で埋められる。
		ファイル名 4 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 1 のファイルのレコードは、ファイル名 4 のファイルに引き渡される時、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が EBCDIK コードの空白(X'40')で埋められる。	ファイル名 4 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 1 のファイルのレコードは、ファイル名 4 のファイルに引き渡される時、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が半角空白(X'20')で埋められる。
MOVE 文	-	一意名 1 に日本語項目または日本語編集項目を指定したときは、一意名 2 は日本語項目、日本語編集項目または集団項目でなければならない。	VOS3 COBOL85 と同じ。
		一意名 2 に日本語項目または日本語編集項目を指定したときは、一意名 1 は日本語項目、日本語編集項目または集団項目でなければならない。	VOS3 COBOL85 と同じ。
		定数 1 に日本語文字定数を指定したときは、一意名 2 は日本語項目または日本語編集項目でなければならない。	VOS3 COBOL85 と同じ。
		一意名 2 の日本語項目または日本語編集項目で、一意名 1 が集団項目のとき、一意名 2 に対して空白づめが発生した場合、EBCDIK コードの空白(X'4040')が埋められる。	一意名 2 の日本語項目または日本語編集項目で、一意名 1 が集団項目のとき、一意名 2 に対して空白づめが発生した場合、半角空白(X'2020')が埋められる。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		一意名 1 の日本語項目または日本語編集項目で、一意名 2 が集団項目のとき、一意名 2 に対して空白づめが発生した場合、EBCDIK コードの空白(X'4040')が埋められる。	一意名 1 の日本語項目または日本語編集項目で、一意名 2 が集団項目のとき、一意名 2 に対して空白づめが発生した場合、半角空白(X'2020')が埋められる。
		一意名 1 および一意名 2 が日本語項目または日本語編集項目のとき、一意名 2 に対して空白づめが発生した場合、KEIS コード(X'A1A1')が埋められる。	一意名 1 および一意名 2 が日本語項目または日本語編集項目のとき、一意名 2 に対して空白づめが発生した場合、日本語空白(X'8140')が埋められる。
		受け取り側作用対象の字類が日本語の場合、送り出し側作用対象に表意定数 HIGH-VALUE, LOW-VALUE および ZERO を指定してはならない。*2	受け取り側作用対象の字類が日本語の場合、送り出し側作用対象に表意定数 HIGH-VALUE, LOW-VALUE および ZERO を指定してもよい。*1 <ul style="list-style-type: none"> 表意定数 HIGH-VALUE : 2 バイトの X'FFFF'として扱う。 表意定数 LOW-VALUE : 2 バイトの X'0000'として扱う。 表意定数 ZERO : 全角文字のゼロ(X'824F')として扱う。
READ 文	—	INTO 指定のある READ 文の実行結果は、次の規則をこの順に適用したものと等しい。*3 <ul style="list-style-type: none"> INTO 指定のない同じ READ 文の実行 現在レコードが CORRESPONDING 指定のない MOVE 文の規則に従って、レコード領域から一意名 1 の領域へ転記される。現在のレコードの大きさは、RECORD 句の規則で定める。暗黙的の転記はレコード領域を集団項目として扱う転記となる。 <p>なお、MOVE 文の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。</p>	INTO 指定のある READ 文の実行結果は、次の規則をこの順に適用したものと等しい。*4 <ul style="list-style-type: none"> INTO 指定のない同じ READ 文の実行 現在レコードが CORRESPONDING 指定のない MOVE 文の規則に従って、レコード領域から一意名 1 の領域へ転記される。現在のレコードの大きさは、RECORD 句の規則で定める。暗黙的の転記はレコード領域を集団項目として扱う転記となる。 <p>なお、MOVE 文の規則は、この表の COBOL2002 の「MOVE 文」に従う。</p>
RELEASE 文	—	FROM 指定のある RELEASE 文は、次に示す文の実行と等しい。 <ul style="list-style-type: none"> MOVE 文の規則に従った文： MOVE 一意名 1 TO レコード名 1 	VOS3 COBOL85 と同じ。 なお、レコード名 1 または一意名 1 に日本語項目および日本語編集項目を指定したときの MOVE 文の規則は、この表の COBOL2002 の「MOVE 文」に従う。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		<p>なお、レコード名 1 または一意名 1 に日本語項目および日本語編集項目を指定したときの MOVE 文の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。</p> <ul style="list-style-type: none"> • それに続く FROM 指定のない同じ RELEASE 文。 	
RETURN 文	—	<p>INTO 指定のある RETURN 文の実行結果は、次の規則をこの順に適用したものと等しい。*3</p> <ul style="list-style-type: none"> • INTO 指定のない同じ RETURN 文の実行 • 現在レコードが CORRESPONDING 指定のない MOVE 文の規則に従って、レコード領域から一意名 1 の領域へ転記される。現在のレコードの大きさは、RECORD 句の規則で定める。暗黙的の転記はレコード領域を集団項目として扱う転記となる。なお、MOVE 文の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。 	<p>INTO 指定のある RETURN 文の実行結果は、次の規則をこの順に適用したものと等しい。*4</p> <ul style="list-style-type: none"> • INTO 指定のない同じ RETURN 文の実行 • 現在レコードが CORRESPONDING 指定のない MOVE 文の規則に従って、レコード領域から一意名 1 の領域へ転記される。現在のレコードの大きさは、RECORD 句の規則で定める。暗黙的の転記はレコード領域を集団項目として扱う転記となる。なお、MOVE 文の規則は、この表の COBOL2002 の「MOVE 文」に従う。
REWRITE 文	—	<p>FROM 指定のある REWRITE 文は、次に示す文の実行と等しい。</p> <ul style="list-style-type: none"> • MOVE 文の規則に従った文：MOVE 一意名 1 TO レコード名 1 <p>なお、レコード名 1 または一意名 1 に日本語項目および日本語編集項目を指定したときの MOVE 文の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。</p> <ul style="list-style-type: none"> • それに続く FROM 指定のない同じ REWRITE 文。 	<p>VOS3 COBOL85 と同じ。</p> <p>なお、レコード名 1 または一意名 1 に日本語項目および日本語編集項目を指定したときの MOVE 文の規則は、この表の COBOL2002 の「MOVE 文」に従う。</p>
SEARCH 文	—	<p>書き方 2 のデータ名 1, データ名 2, 一意名 3, 一意名 4, 定数 1 および定数 2 に日本語項目, 日本語編集項目および日本語文字定数を指定したときの比較条件の規則は、この表の VOS3 COBOL85 の「条件式 (比較条件)」に従う。</p>	<p>書き方 2 のデータ名 1, データ名 2, 一意名 3, 一意名 4, 定数 1 および定数 2 に日本語項目, 日本語編集項目および日本語文字定数を指定したときの比較条件の規則は、この表の COBOL2002 の「条件式 (比較条件)」に従う。</p>
SORT 文	—	<p>COLLATING SEQUENCE 句に指定した符号系名が、STANDARD-1 指定また</p>	<p>COLLATING SEQUENCE 句に指定した符号系名に、STANDARD-1</p>

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		は STANDARD-2 指定の場合、MERGE 文を無視する。	指定または STANDARD-2 を指定してもよい。
		ファイル名 1 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 2 またはファイル名 3 のファイルのレコードは、ファイル名 1 のファイルに引き渡されるとき、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が EBCDIK コードの空白(X'40')で埋められる。	ファイル名 1 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 2 またはファイル名 3 のファイルのレコードは、ファイル名 1 のファイルに引き渡されるとき、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が半角空白(X'20')で埋められる。
		ファイル名 4 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 1 のファイルのレコードは、ファイル名 4 のファイルに引き渡されるとき、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が EBCDIK コードの空白(X'40')で埋められる。	ファイル名 4 のファイルが固定長レコードだけを含むとき、その固定長よりも少ない文字位置を含むファイル名 1 のファイルのレコードは、ファイル名 4 のファイルに引き渡されるとき、固定長集団項目同士の転記として扱うため、レコードの最後の文字位置の右側が半角空白(X'20')で埋められる。
START 文	—	索引ファイルの場合、KEY 指定を書くと、データ名 1 はファイル名 1 に関連する主レコードキーまたは副レコードキーとして指定されたデータ項目でなければならない。なお、データ名 1 に日本語項目および日本語編集項目を指定したときの規則は、この表の VOS3 COBOL85 の「RECORD KEY 句」および「ALTERNATE RECORD KEY 句」に従う。	VOS3 COBOL85 と同じ。 なお、データ名 1 に日本語項目および日本語編集項目を指定したときの規則は、この表の COBOL2002 の「RECORD KEY 句」および「ALTERNATE RECORD KEY 句」に従う。
STRING 文	—	一意名 3 に日本語項目を指定したとき、定数はすべて日本語文字定数または表意定数 SPACE(SPACES)でなければならない。	VOS3 COBOL85 と同じ。
		一意名 3 の用途が明示的または暗黙的な DISPLAY（日本語項目を除く）であるならば、定数はすべて文字定数または表意定数でなければならない。	VOS3 COBOL85 と同じ。
		一意名 3 に日本語項目を指定した場合、一意名 4 を除くすべての一意名は日本語項目を指定しなければならない。	VOS3 COBOL85 と同じ。
		一意名 3 の用途が明示的または暗黙的な DISPLAY（日本語項目を除く）であるならば、一意名 4 を除くすべての一意名の	VOS3 COBOL85 と同じ。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		用途は、明示的または暗黙的な DISPLAY (日本語項目を除く) でなければならない。	
		一意名 3 に日本語項目を指定した場合、定数に表意定数 SPACE(SPACES)を指定すると、1 文字の KEIS コードの空白 (X'A1A1')として扱う。	一意名 3 に日本語項目を指定した場合、定数に表意定数 SPACE(SPACES)を指定すると、1 文字の日本語空白 (X'8140')として扱う。
		一意名 3 に日本語編集項目を指定してはならない。	VOS3 COBOL85 と同じ。
		STRING 文の実行で、定数 1 の文字列または一意名 1 のデータ項目の内容が、英数字と英数字の間の転記の規則に従って、一意名 3 のデータ項目に転送される。ただし、空白づめは行われない。なお、一意名 4 を除く一意名および定数の字類がすべて日本語のデータ項目の場合の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。	VOS3 COBOL85 と同じ。 なお、一意名 4 を除く一意名および定数の字類がすべて日本語のデータ項目の場合の規則は、この表の COBOL2002 の「MOVE 文」に従う。
UNSTRING 文	-	一意名 1 に日本語項目または日本語編集項目を指定したとき、定数はすべて日本語文字定数または表意定数 SPACE(SPACES)でなければならない。	VOS3 COBOL85 と同じ。
		一意名 1 の用途が明示的または暗黙的な DISPLAY (日本語項目または日本語編集項目を除く) であるならば、定数はすべて文字定数または表意定数でなければならない。	VOS3 COBOL85 と同じ。
		一意名 1 に日本語項目または日本語編集項目を指定した場合、一意名 2、一意名 3 および一意名 5 はすべて日本語項目または日本語編集項目を指定しなければならない。	VOS3 COBOL85 と同じ。
		一意名 1 に日本語項目または日本語編集項目を指定した場合、一意名 4 は日本語項目を指定しなければならない。	VOS3 COBOL85 と同じ。
		一意名 1 の用途が明示的または暗黙的な DISPLAY (日本語項目または日本語編集項目を除く) であるならば、一意名 2、一意名 3、一意名 4 および一意名 5 の用途は、明示的または暗黙的な DISPLAY (日本語項目または日本語編集項目を除く) でなければならない。	VOS3 COBOL85 と同じ。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		一意名 1 に日本語項目または日本語編集項目を指定した場合、定数に表意定数 SPACE(SPACES)を指定すると、1 文字の KEIS コードの空白(X'A1A1')として扱う。	一意名 1 に日本語項目または日本語編集項目を指定した場合、定数に表意定数 SPACE(SPACES)を指定すると、1 文字の日本語空白(X'8140')として扱う。
WRITE 文	—	FROM 指定のある WRITE 文は、次に示す文の実行と等しい。 <ul style="list-style-type: none"> MOVE 文の規則に従った文：MOVE 一意名 1 TO レコード名 1 なお、レコード名 1 または一意名 1 に日本語項目および日本語編集項目を指定したときの MOVE 文の規則は、この表の VOS3 COBOL85 の「MOVE 文」に従う。 それに続く FROM 指定のない同じ WRITE 文。 	VOS3 COBOL85 と同じ。 なお、レコード名 1 または一意名 1 に日本語項目および日本語編集項目を指定したときの MOVE 文の規則は、この表の COBOL2002 の「MOVE 文」に従う。
拡張仕様	16 進文字定数※5	16 進文字定数を指定した VALUE 句を、日本語項目および日本語編集項目に指定してはならない。	VOS3 COBOL85 と同じ。
	PASSWORD 句、 RECORD 句および ALTERNATE RECORD 句	データ名 2 に日本語項目を指定してはならない。	COBOL2002 では PASSWORD 句を覚え書きとみなす。
	APPLY PAGE- DEFINE 句	データ名 1 に日本語項目を指定してはならない。	COBOL2002 では APPLY PAGE- DEFINE 句を覚え書きとみなす。
	EXAMINE 文と TALLY 特殊レジスタ	<ul style="list-style-type: none"> 一意名 1 に日本語項目および日本語編集項目を指定してはならない。 定数 1～定数 4 は、1 文字からなる文字定数か、符号のない 1 けたの整数か、または ALL 定数、記号文字以外の表意定数でなければならない。日本語文字定数は、指定してはならない。 	VOS3 COBOL85 と同じ。
	MOVE 文（日付と時刻用） WHEN-COMPILED CURRENT-DATE TIME-OF-DAY	一意名に日本語項目および日本語編集項目を指定してはならない。	VOS3 COBOL85 と同じ。
	TRANSFORM 文	<ul style="list-style-type: none"> 一意名 1～一意名 3 に日本語項目を指定してはならない。 定数 2 および定数 3 に日本語文字定数を指定してはならない。 	VOS3 COBOL85 と同じ。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
	CURRENT-DATE 関数	一意名 1 に日本語項目および日本語編集項目を指定してはならない。	VOS3 COBOL85 と同じ。
	WHEN-COMPILED 関数	一意名 1 に日本語項目および日本語編集項目を指定してはならない。	VOS3 COBOL85 と同じ。
	ENTRY 文	定数に日本語文字定数を指定してはならない。	VOS3 COBOL85 と同じ。
報告書作成機能	CONTROL 句	データ名 1 に日本語項目および日本語編集項目を指定したときの比較の規則は、この表の VOS3 COBOL85 の「条件式 (比較条件)」に従う。	データ名 1 に日本語項目および日本語編集項目を指定したときの比較の規則は、この表の COBOL2002 の「条件式 (比較条件)」に従う。
特殊入出力装置の取扱い	JIS 標準ファイル (PICTURE 句)	文字列に日本語および日本語編集を指定してはならない。	JIS 標準ファイルは未サポート。
直接, DAM 相対, 索引順ファイルの取扱い	ACTUAL KEY 句	トラック識別子 VOS3 COBOL85 は, ACTUAL KEY のデータ名の先頭 4 バイトを 2 進形式のトラック識別子とみなす。利用者はこれを 5 けた (PIC S9(5)) の 2 進形式の整数項目として定義しておかなければならない。このため, LANGOPT=(D)指定時, トラック識別子として日本語項目および日本語編集項目を指定した場合の実行結果は保証しない。	ACTUAL KEY 句は未サポート。
		レコード識別子 ACTUAL KEY のデータ名のトラック識別子を除いた残りの部分がレコード識別子であり, その大きさは 1~255 バイトでなければならない。レコード識別子は PICTURE (形式) 句や USAGE (用途) 句などでどのように記述されていても, VOS3 COBOL85 は, 割り当てたバイト数をもつ英数字項目とみなして処理する。 キーの一致, 不一致の判定は, 英数字項目同士の比較条件の規則に従って行う。	
	NOMINAL KEY 句	DAM 相対ファイル データ名に日本語項目および日本語編集項目を指定してはならない。	NOMINAL KEY 句は未サポート。

大項目	小項目	VOS3 COBOL85	COBOL2002
		LANGOPT=(D)オプション指定時	-JPN,V3JPNSpace オプション指定時
		索引順ファイル データ名に日本語項目および日本語編集項目を指定してはならない。	
データベース機能	サブスキーマ記述項	データ名1～データ名3およびデータ名6に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	CONNECT 文	データ名1に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	DISCONNECT 文	データ名1に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	ERASE 文	データ名1に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	FETCH 文	データ名1, データ名2およびデータ名3に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	FIND 文	データ名に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	FINISH 文	データ名1およびデータ名2に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	GET 文	データ名1, データ名2およびデータ名3に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	MODIFY 文	データ名1およびデータ名2に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	NULLIFY 文	データ名に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	RECONNECT 文	データ名に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。
	STORE 文	データ名1, データ名2およびデータ名3に日本語項目を指定してはならない。	VOS3 COBOL85 と同じ。

注※1

COBOL2002 では、日本語の表意定数として、次に示すものが対象である。

- ZERO (ZEROS/ZEROES)
- SPACE (SPACES)
- HIGH-VALUE (HIGH-VALUES)
- LOW-VALUE (LOW-VALUES)

注※2

VOS3 COBOL85 では、日本語の表意定数として SPACE(SPACES)だけが対象である。

注※3

INTO に指定されたデータ名が日本語項目の場合でも集団項目の転記となるため、日本語項目の右端に空白を補う場合は、半角空白(X'4040')となる。

注※4

INTO に指定されたデータ名が日本語項目の場合でも集団項目の転記となるため、日本語項目の右端に空白を補う場合は、半角空白(X'2020')となる。

注※5

COBOL2002 では、16 進英数字定数となる。

付録 B.2 比較条件の組み合わせ

比較条件の組み合わせを次の表に示す。

表 B-2 比較条件の組み合わせ

左辺 (右辺)	右辺 (左辺)					
	VOS3 COBOL85 LANGOPT=(D)オプション指定時			COBOL2002 -JPN,V3JPNSpace オプション指定時		
	日本語	日本語編集	日本語定数※1	日本語集団 項目 日本語	日本語編集	日本語定数※1
英数字集団項目	○	○	○	○	○	○
英字	×	×	×	×	×	×
英数字	×	×	×	×	×	×
英数字編集	×	×	×	×	×	×
数字編集	×	×	×	×	×	×
QUOTE	×	×	×	×	×	×
SPACE	○	○	×	○	○	×
HIGH-VALUE	×	×	×	○※2	○※2	×
LOW-VALUE	×	×	×	○※2	○※2	×
ZERO	×	×	×	○※2	○※2	×
文字定数	×	×	×	×	×	×
ALL 文字定数	×	×	×	×	×	×

左辺 (右辺)	右辺 (左辺)					
	VOS3 COBOL85 LANGOPT=(D)オプション指定時			COBOL2002 -JPN,V3JPNSpace オプション指定時		
	日本語	日本語編集	日本語定数※1	日本語集団 項目 日本語	日本語編集	日本語定数※1
数字定数	×	×	×	×	×	×
外部 10 進	×	×	×	×	×	×
内部 10 進	×	×	×	×	×	×
2 進	×	×	×	×	×	×
指標名	×	×	×	×	×	×
指標データ	×	×	×	×	×	×
ブール	×	×	×	×	×	×
外部浮動	×	×	×	×	×	×
内部浮動	×	×	×	×	×	×
アドレス名	×	×	×	×	×	×
アドレスデータ	×	×	×	×	×	×
日本語	○	○	○	○	○	○
日本語編集	○	○	○	○	○	○
日本語定数	○	○	×	○	○	×
ALL 日本語定数	○	○	×	○	○	×

(凡例)

○：組み合わせて使用できる。

×

注※1

日本語定数には、ALL 指定も含む。

注※2

VOS3 COBOL85 ではコンパイルエラーとなるが、COBOL2002 では組み合わせ可能となる。

なお、COBOL2002 での各表意定数の扱いは、次のとおりとなる。

表意定数 HIGH-VALUE：2 バイトの X'FFFF'として扱う。

表意定数 LOW-VALUE：2 バイトの X'0000'として扱う。

表意定数 ZERO：全角文字のゼロ(X'824F')として扱う。

付録 B.3 字類条件の組み合わせ

字類条件の組み合わせを次の表に示す。

表 B-3 字類条件の組み合わせ

左辺 (右辺)	右辺 (左辺)							
	VOS3 COBOL85 LANGOPT=(D)オプション指定時				COBOL2002 -JPN,V3JPNSpace オプション指定時			
	NUMERIC	ALPHABETIC ALPHABETIC- LOWER ALPHABETIC- UPPER	BOOLEAN	字 類 名	NUMERIC	ALPHABETIC ALPHABETIC- LOWER ALPHABETIC- UPPER	BOOLEAN	字 類 名
英数字集団項目	○	○	○	○	○	○	○	○
英字	×	○	×	○	×	○	×	○
英数字	○	○	○	○	○	○	○	○
英数字編集	○	○	×	○	○	○	×	○
数字編集	○	○	×	○	○	○	×	○
外部 10 進	○	×	×	○	○	○	×	○
内部 10 進	○	×	×	×	○	×	×	○
2 進	×	×	×	×	×	×	×	×
指標名	×	×	×	×	×	×	×	×
指標データ	×	×	×	×	×	×	×	×
ブール	×	×	○	×	×	×※1	○※2	×
外部浮動	×	×	×	×	×	×	×	×
内部浮動	×	×	×	×	×	×	×	×
アドレス名	×	×	×	×	×	×	×	×
アドレスデータ	×	×	×	×	×	×	×	×
日本語	×	×	×	×	×	×	×	×
日本語編集	×	×	×	×	×	×	×	×

(凡例)

- ：組み合わせて使用できる。
- ×

注※1

次の項目を指定した場合、出力するメッセージの種類が異なる。

- ・ALPHABETIC 指定の場合、KCCC4451C-S を出力する。
- ・ALPHABETIC-LOWER 指定の場合、KCCC4457C-S を出力する。
- ・ALPHABETIC-UPPER 指定の場合、KCCC4466C-S を出力する。

注※2

内部ブール項目の場合は、エラーメッセージを出力する。

付録 B.4 転記の組み合わせ

(1) 受け取り側作用対象が日本語集団項目、日本語項目または日本語編集項目の場合

受け取り側作用対象が日本語集団項目、日本語項目または日本語編集項目の場合の転記の組み合わせを次の表に示す。

表 B-4 転記の組み合わせ (受け取り側作用対象が日本語集団項目、日本語項目または日本語編集項目の場合)

送り出し側作用対象	受け取り側作用対象			
	VOS3 COBOL85 LANGOPT=(D)指定時		COBOL2002 -JPN,V3JPNSpace オプション指定時	
	日本語	日本語編集	日本語集団項目 日本語	日本語編集
英数字集団項目		○		○
英字		×		×
英数字		×		×
英数字編集		×		×
数字編集		×		×
QUOTE		×		×
SPACE		○		○
HIGH-VALUE		×		○※
LOW-VALUE		×		○※
ALL 定数		×		×
ZERO		×		○※
文字定数		×		×
数字定数		×		×
外部 10 進		×		×
内部 10 進		×		×
2 進		×		×

送り出し側作用対象	受け取り側作用対象			
	VOS3 COBOL85 LANGOPT=(D)指定時		COBOL2002 -JPN,V3JPNSpace オプション指定時	
	日本語	日本語編集	日本語集団項目 日本語	日本語編集
外部ブール		×		×
内部ブール		×		×
ブール定数		×		×
外部浮動		×		×
内部浮動		×		×
日本語		○		○
日本語編集		○		○
日本語文字定数		○		○
ALL 日本語文字定数		○		○

(凡例)

○：組み合わせて使用できる。

×：組み合わせて使用できない。

注※

VOS3 COBOL85 ではコンパイルエラーとなるが、COBOL2002 では組み合わせ使用できる。

なお、COBOL2002 での各表意定数の扱いは、次のとおりとなる。

表意定数 HIGH-VALUE：2 バイトの X'FFFF'

表意定数 LOW-VALUE：2 バイトの X'0000'

表意定数 ZERO：全角文字のゼロ(X'824F')

(2) 送り出し側作用対象が日本語集団項目、日本語項目または日本語編集項目の場合

送り出し側作用対象が日本語集団項目、日本語項目または日本語編集項目の場合の転記の組み合わせを次の表に示す。

表 B-5 転記の組み合わせ(送り出し側作用対象が日本語集団項目, 日本語項目, 日本語編集項目
または日本語定数の場合)

受け取り側作用対象	送り出し側作用対象					
	VOS3 COBOL85 LANGOPT=(D)指定時			COBOL2002 -JPN,V3JPNSpace オプション指定時		
	日本語	日本語編集	日本語定数	日本語集団項目 日本語	日本語編集	日本語定数
英数字集団項目		○			○	
英字		×			×	
英数字		×			×	
英数字編集		×			×	
数字編集		×			×	
外部10進		×			×	
内部10進		×			×	
2進		×			×	
外部ブール		×			×	
内部ブール		×			×	
外部浮動		×			×	
内部浮動		×			×	
日本語		○			○	
日本語編集		○			○	

(凡例)

- ：組み合わせて使用できる。
- ×：組み合わせて使用できない。

付録 C STANDARD オプションと-StdVersion オプションを指定した場合の仕様の相違

ここでは、VOS3 COBOL85 で STANDARD=(1)または STANDARD=(2)オプションを指定した場合と、COBOL2002 で-StdVersion,1 または-StdVersion,2 オプションを指定した場合の仕様の相違点について説明する。

なお、-StdVersion オプションについては、マニュアル「COBOL2002 ユーザーズガイド」または「COBOL2002 使用の手引 手引編」を参照のこと。

表 C-1 VOS3 COBOL85 と COBOL2002 の仕様の比較

項番	差異の生じる言語仕様	VOS3 COBOL85		COBOL2002	
		STANDARD=(1)指定時	STANDARD=(2)指定時	-StdVersion,1 指定時	-StdVersion,2 指定時
1	CURRENCY SIGN 句で指定する定数	L, =, /を指定できる。	L, =, /を指定できない。	=, /が指定できる。	=, /が指定できない。
2	SELECT OPTIONAL 句	覚え書きとする。	覚え書きとしない。	VOS3 COBOL85 と同じ。	
3	RESERVE 句	コンパイラが自動的に割り当ててる一つの出力領域に加えて、RESERVE 句で指定した個数の出力領域を確保する。 なお、RESERVE 句に補助語 ALTERNATE が書かれていると、STANDARD の指定にかかわらず STANDARD=(1)の仕様を使用される。	RESERVE 句で指定した個数の出力領域を確保する。	覚え書きとする。	
4	比較条件の略記法における NOT	略記法の条件中の NOT が、比較演算子一部とも論理演算子とも取れる場合は、論理演算子とみなす。	NOT の直後に GREATER, >, LESS, <, EQUAL, =, >=, <=のどれかが続くとき、NOT は比較条件の一部となる。これ以外の NOT は論理演算子とみなす。	VOS3 COBOL85 と同じ。	
5	MOVE 文の位取り	送り側が位取りした整数項目 (PICTURE 句の右端文字が P のとき) 受け取り側が英数字項	左記の場合、あとに続くゼロを切り捨てない。	VOS3 COBOL85 と同じ。	

項番	差異の生じる言語仕様	VOS3 COBOL85		COBOL2002	
		STANDARD=(1)指定時	STANDARD=(2)指定時	-StdVersion,1 指定時	-StdVersion,2 指定時
		目または英数字編集項目であれば、あとに続くゼロは切り捨て空白とする。			
6	UNSTRING 文の DELIMITED BY ALL 指定	二つ以上の連続する区切り文字が出現した場合、区切り文字の受け取り領域に入るだけ、区切り文字を移す。	二つ以上の連続する区切り文字が出現した場合、区切り文字の受け取り領域には1個(1組)の区切り文字しか移さない。	VOS3 COBOL85 と同じ。	
7	COPY 文の展開	第1次規格の仕様に従った書き方は、旧仕様書として展開する。 第1次規格の仕様に従っていない書き方は、第2次規格の仕様に従った COPY 文とみなし展開する。	第2次規格以降の仕様に従って COPY 文を展開する。	VOS3 COBOL85 と同じ。	
8	独立区分中の PERFORM 文で固定区分中の手続きを呼び出したときの扱い	手続きを出て、呼び出し元に戻ると、独立区分は再度初期設定される。	手続きを出て、呼び出し元に戻ると、独立区分は呼び出し時と同じである。	COBOL2002 では、VOS3 COBOL85 の区分化機能は未サポートである。	
9	JUST 句の処理	JUST 句の指定がある英数字、英字、英数字編集に VALUE 句で初期値を与えるとき、JUST 句の機能を働かせる。	左記の場合、JUST 句の機能を働かせない。	VOS3 COBOL85 と同じ。	
10	WRITE 文の ADVANCING 句の仮定	一つのファイルに対する複数個の WRITE 文で ADVANCING 句があるものとなないものを混用した場合、ADVANCING 句のないものに対して BEFORE ADVANCING 1 か AFTER ADVANCING 1 のどちらかを仮定する。 ADVANCING 句指定のある WRITE 文の指定がすべて BEFORE のとき BEFORE	左記の場合、常に AFTER ADVANCING 1 を仮定する。	VOS3 COBOL85 と同じ。	

項 番	差異の生じる言語 仕様	VOS3 COBOL85		COBOL2002	
		STANDARD=(1)指 定時	STANDARD=(2)指 定時	-StdVersion,1 指 定時	-StdVersion,2 指定時
		ADVANCING 1, す べて AFTER のとき AFTER ADVANCING 1, BEFORE と AFTER 混 在のとき BEFORE ADVANCING 1 を仮 定する。			
11	EVALUATE 文の ALSO の指定	選択主体, または選択対象を複数記述する場合は, それぞれを空白で区切って指定する (VOS3 COBOL85 では, 選択主体, または選択対象を複 数記述する場合, ALSO 指定で区切ってもよい)。		選択主体, および選択対象を複数記述する場 合, ALSO 指定は必須である。	
12	PERFORM 文の VARYING… AFTER 指定 (一つ の PERFORM 文で の 2~3 重ループ) がある場合の反復制 御変数の制御方法	反復制御変数の初期化を行う位置を次のようにす る。 1. TEST AFTER 句指定時 外側のループの反復制御変数を BY 指定の値で 増加させてから, 内側のループの反復制御変数 を FROM 指定の変数で再び初期化する。 2. TEST BEFORE 句指定時 内側のループの反復制御変数を FROM 指定の 変数で初期化してから, 外側のループの反復制 御変数を BY 指定の値で増加させる。		VOS3 COBOL85 と同じ。	
13	END-動詞の記述	四則演算, 入出力などの文が条件文となってい るときだけ END-動詞を記述できる。条件文と文の 対応を次に示す。 ON SIZE ERROR の場合 • ADD • SUBTRACT • MULTIPLY • DIVIDE • COMPUTE AT END/AT EOP/INVALID KEY の場合 • DELETE • READ • REWRITE • WRITE • START ON OVERFLOW/ON EXCEPTION の場合 • CALL • STRING • UNSTRING		左記に示した文が無条件のときでも END-動 詞を記述できる。	

項番	差異の生じる言語仕様	VOS3 COBOL85		COBOL2002	
		STANDARD=(1)指定時	STANDARD=(2)指定時	-StdVersion,1 指定時	-StdVersion,2 指定時
14	SEARCH 文での部分参照の指定	第3次規格でのサポートであり、部分参照は指定できない。		制限のない個所に部分参照を指定できる。	

付録 D このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 D.1 関連マニュアル

このマニュアルは次のマニュアルと関連があります。必要に応じてお読みください。

- COBOL2002 ユーザーズガイド (4010-1J-800)
- COBOL2002 操作ガイド (4010-1J-801)
- COBOL2002 使用の手引 手引編 (4010-1J-802)
- COBOL2002 使用の手引 操作編 (4010-1J-803)
- COBOL2002 言語 標準仕様編 (4010-1J-804)
- COBOL2002 Cosminexus 連携機能ガイド (4010-1J-806)
- COBOL2002 Java プログラム呼び出し機能ガイド (4010-1J-807)
- COBOL2002 メッセージ (4010-1J-809)
- VOS3 COBOL85 言語 (6190-3-723)
- OpenTP1 プログラム作成リファレンス COBOL 言語編 (3000-3-D55)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 概説 (3020-7-511)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド (3020-7-512)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド (3020-7-513)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド (3020-7-514)
- VOS3 データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/RD E2) (6190-6-638)
- VOS3 データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/SD) (6190-6-652)
- VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス (6190-6-656)

付録 D.2 このマニュアルでの表記

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

マニュアルでの表記	製品名
MSMQ	Microsoft Message Queuing Server
SQL Server	Microsoft SQL Server

マニュアルでの表記		製品名
Windows	Windows 10(x64)	Windows 10 Pro 日本語版(64 ビット版)
		Windows 10 Enterprise 日本語版(64 ビット版)
	Windows 11	Windows 11 Pro 日本語版
		Windows 11 Enterprise 日本語版
	Windows Server 2019	Windows Server 2019 Standard 日本語版
		Windows Server 2019 Datacenter 日本語版
	Windows Server 2022	Windows Server 2022 Standard 日本語版
		Windows Server 2022 Datacenter 日本語版

このマニュアル中では、「このシステム」と表現している場合「COBOL2002」を示しています。

また、このマニュアルは、製品種別によって相違点があります。本文中での製品種別ごとの表記を次に示します。

マニュアルでの表記		該当する製品の形名	
Windows COBOL2002 ※1	Windows(x86) COBOL2002	P-2636-2354 P-2436-6354 P-2636-4354 P-2636-7354	
	Windows(x64) COBOL2002※2	P-2936-2354 P-2936-6354 P-2936-7354	
UNIX COBOL2002※1	AIX COBOL2002	AIX(32) COBOL2002	— ※3
		AIX(64) COBOL2002※2	— ※3
	Linux COBOL2002	Linux(x86) COBOL2002	— ※3
		Linux(x64) COBOL2002※2	P-9W36-1251

注※1

Windows COBOL2002 と UNIX COBOL2002 で機能差がない場合、COBOL2002 と表記しています。

注※2

Windows(x64) COBOL2002, AIX(64) COBOL2002, Linux(x64) COBOL2002 で機能差がない場合、64bit 版 COBOL2002 と表記しています。

注※3

該当する製品の形名の詳細は、「リリースノート」でご確認ください。

また、このマニュアルでは各製品を次のように表記しています。

マニュアルでの表記		製品名	
Excel		Microsoft Office Excel	
		Microsoft Excel	
UNIX	AIX	AIX V7.1	
		AIX V7.2	
		AIX 7.3	
	Linux	Linux Server 7 (64-bit x86_64)	Red Hat Enterprise Linux Server 7 (64-bit x86_64)
		Linux Server 8 (64-bit x86_64)	Red Hat Enterprise Linux Server 8 (64-bit x86_64)
		Linux Server 9 (64-bit x86_64)	Red Hat Enterprise Linux Server 9 (64-bit x86_64)
XMAP3		XMAP3 Server	
		XMAP3 Developer Version 5	
		XMAP3 Server Runtime Version 5	
		XMAP3 Client Runtime Version 5	

- 日立 COBOL2002 のことを日立 COBOL2002，または単に COBOL2002 と表記しています。また，プラットフォームを明確にする必要がある場合は，「Windows COBOL2002」「UNIX COBOL2002」のように表記しています。
- 日立 COBOL85 のことを日立 COBOL85，または単に COBOL85 と表記しています。また，プラットフォームを明確にする必要がある場合は，「PC COBOL85」「VOS3 COBOL85」のように表記しています。
- XMAP3 の製品を区別する必要がある場合は，それぞれの製品名称を表記しています。
- コンパイラオプションの説明では，次の表記を使用します。

「XXX オプション」，または単に「XXX」とオプション名が表記されている場合

XXX オプションについて，サブオプションの組み合わせを含む，すべての場合を意味します。

「XXX,YYY オプション」，または単に「XXX,YYY」とサブオプションを含めたオプション名が表記されている場合

XXX,YYY オプションだけの場合を意味します。

「XXX コンパイラオプション」と表記されている場合

リンカオプションなど，ほかのオプションと明確に区別する必要がある場合を意味します。

(例 1)

「-Compile オプション」または「-Compile」と記載している場合，-Compile オプションのサブオプションの組み合わせすべて (-Compile,CheckOnly/-Compile,NoLink) を意味します。

(例 2)

「-Compile,CheckOnly オプション」または「-Compile,CheckOnly」と記載している場合、-Compile,CheckOnly だけを意味します。

索引

記号

- Cblctr 135
- CompatiV3 120, 471
- DebugLine 474
- DigitsTrunc 462
- DoubleQuote 462
- H8Switch 473
- IgnoreLCC 466
- JPN,Alnum 468
- JPN,V3JPNSpace 467
- JPN オプション 442
- StdVersion,1 454, 458
- StdVersion,2 458
- TruncCheck 476
- TruncCheck,Binary 478
- TruncCheck オプション 442
- UniObjGen [CONVERT-TO-VARIANT 関数] 354
- UniObjGen [DISPLAY 文 (SCREEN SECTION)] 231
- UniObjGen [HiRDB による索引ファイル入出力機能] 378
- UniObjGen [INVOKE 文] 344
- UniObjGen [PICTURE 句 (SCREEN SECTION)] 220
- UniObjGen [PROMPT 句 (SCREEN SECTION)] 221
- UniObjGen [RESET 句 (WINDOW SECTION)] 275
- UniObjGen [SET 文] 349
- UniObjGen [TYPE-OF-PROPERTY 関数] 355
- UniObjGen [VALUE 句 (SCREEN SECTION)] 226
- UniObjGen [VALUE 句 (WINDOW SECTION)] 282
- UniObjGen [アドレス型関数] 316
- UniObjGen [全般規則 (日本語機能)] 30
- UniObjGen [手続き部] 190

- UniObjGen [ファイル記述項] 55
- UniObjGen [プログラム間連絡機能の環境部] 487
- UniObjGen [プログラム名, クラス名, インタフェース名, メソッド名および利用者定義関数名の日本語指定] 484
- V3Rec,Fixed 463
- V3Rec,Variable 463

数字

- 16 進英数字定数 482
- 16 進数字定数 482
- 16 進日本語文字定数 32
- 1 バイト 2 進機能 534
- 2 進項目の拡張 534
- 2 進項目のけた落とし 462
- 78 定数名の記述 496

A

- ACCEPT COMMAND-LINE 192
- ACCEPT 文 (CSV ファイル入出力機能) 333
- ACCEPT 文 (SCREEN SECTION) 227
- ACCEPT 文 (WINDOW SECTION) 283
- ACCEPT 文 (数字項目のけた拡張機能) 404
- ACCEPT 文 (ブール形式データを扱う機能) 95
- ACCEPT 文 (浮動小数点形式データを扱う機能) 115
- ACCEPT 文による入出力 333
- ACCEPT 文の拡張 500
- ADDRESSED 句 308
- ADDR 関数 316
- ADDR 関数の参照 312
- ADD 文 115
- ADD 文 (数字項目のけた拡張機能) 404
- ADVANCING 指定なしの WRITE 文での仮定 458
- ALTERNATE RECORD KEY 句 70, 380
- APPLY FILE-SHARE 句 491
- APPLY FORMS-NAME 句 491
- APPLY FORMS-OVERLAY 句 38, 299

APPLY WINDOW-CONTROL-1 句 243
APPLY WRITE-ONLY 句 490
APPLY 句 492
ASSIGN 句の MULTIPLE REEL/UNIT 指定 488
ASSIGN 句のデータ名指定 488
AUTO 句 210

B

BACKGROUND-COLOR 句 211
BELL 句 212
BETWEEN 述語 163
BLANK WHEN ZERO 句 213, 249
BLANK WHEN ZERO 句 (数字項目のけた拡張機能)
399
BLANK 句 212
BLINK 句 213, 249
BLOCK CONTAINS 句 (数字項目のけた拡張機能)
399

C

CALL 文 (下線付き定数の呼び出し) 502
CALL 文 (数字項目のけた拡張機能) 405
CALL 文 (データベースアクセス機能) 184
CALL 文 (動的長基本項目機能) 437
CALL 文 (浮動小数点形式データを扱う機能) 115
CALL 文 [C 言語インタフェース機能] 323
CALL 文 [OLE2 オートメーションインタフェース機能] 352
CALL 文の引数指定の拡張 501
CANCEL 文 285, 383
CBL-CTR 特殊レジスタ 135
CD 139, 195
CDECL 487
CHARACTER TYPE 句 39, 300
CLOSE 文 (CSV ファイル入出力機能) 329
CLOSE 文 (HiRDB による索引ファイル入出力機能)
383
CLOSE 文 (データベースアクセス機能) 184
CLOSE 文 (テキストファイル機能) 56
CLOSE 文 (ファイル共用機能) 63

COBOL データから VARIANT 値への変換規則 338
COBOL と C 言語の連携 (動的長基本項目機能) 443
COBOL の語 464
COBOL のプログラム内で使用する例 (動的長基本項目機能) 444
COBOL プログラム間の連携 (動的長基本項目機能)
444
CODE 句 123
COLOR 句 250
COLUMN NUMBER 句 251
COLUMN 句 214
COMMIT 文 142, 177, 383
COMPUTATIONAL-1 103, 107, 109
COMPUTATIONAL-2 103, 107, 109
COMPUTATIONAL-5 320
COMPUTATIONAL-X 536
COMPUTE 文 95, 115, 313
COMPUTE 文 (数字項目のけた拡張機能) 405
COMPUTE 文の作用対象 313
COMP-1 103, 107, 109
COMP-2 103, 107, 109
COMP-5 320
COMP-X 536
COMP-X 項目 535
COMP-X 項目が 2 進項目として扱われる個所 540
CONNECT 文 175, 363
CONSOLE IS CRT 句 202
CONTROL FOOTING 指定 131
CONTROL 句 123
CONVERT-CODE 関数 517
CONVERT-TO-VARIANT 関数 353
COPY 文 456
CORRESPONDING 指定のある MOVE 文 509
COUNT-CHAR 関数 450, 518
CREATEOBJ メソッドの操作 346
CRT STATUS 句 203
CSV ファイル 326
CSV ファイル入出力機能 325
CSV ファイル入出力機能 [概説] 326

CURRENCY SIGN 句 38
CURSOR 句 202
C 言語インタフェース機能 318

D

DATA FORMAT 句 74
DATA RECORDS 句 (数字項目のけた拡張機能) 399
DATA RECORD 句 122
DEALLOCATE PREPARE 文 186
DELETE 文 63, 181, 384
DETAIL 指定 130
DISABLE 文 142, 200
DISCONNECT 文 176, 363
DISPLAY COMMAND-LINE 193
DISPLAY 文 (COMP-X 項目) 540
DISPLAY 文 (CSV ファイル入出力機能) 334
DISPLAY 文 (SCREEN SECTION) 230
DISPLAY 文 (WINDOW SECTION) 286
DISPLAY 文 (機能名を直接指定) 502
DISPLAY 文 (数字項目のけた拡張機能) 406
DISPLAY 文 (動的長基本項目機能) 438
DISPLAY 文 (ブール形式データを扱う機能) 96
DISPLAY 文 (浮動小数点形式データを扱う機能) 116
DISPLAY 文 [アドレス操作機能] 312
DISPLAY 文による入出力 333
DISPLAY-OF 関数 450, 522
DIVIDE 文 116
DIVIDE 文 (数字項目のけた拡張機能) 406
DYNAMIC LENGTH STRUCTURE 句 432
DYNAMIC LENGTH 句 434
DYNAMIC 指定 497

E

ENABLE 文 143
ENTER-CHECK 句 256
ENTRY 文 503
ENTRY 文 (数字項目のけた拡張機能) 420
ERASE 句 215
ERASE 文 288, 363

EVALUATE 文 97, 116
EVALUATE 文 (数字項目のけた拡張機能) 406
EVALUATE 文 (動的長基本項目機能) 438
EXAMINE 文 (数字項目のけた拡張機能) 421
EXAMINE 文と TALLY 特殊レジスタ 505
EXEC SQL 374
EXECUTE IMMEDIATE 文 185
EXECUTE 文 186
EXISTS 述語 166
EXIT PROGRAM 文 384
EXIT USE 文 289
EXTERNAL 句 320
EXTERNAL 句 (数字項目のけた拡張機能) 399
EXTERNAL-PROGRAM SECTION 486

F

FD 328, 381
FETCH 文 183, 364
FIELD TYPE 句 256
FILE STATUS 句のデータ名 2 指定 490
FILE-CONTROL 70
FILE-LIMIT 句 488
FIND 文 366
FOREGROUND-COLOR 句 215
FROM 句 168
FULL 句 216

G

GENERATE 文 133
GETOBJ メソッドの操作 347
GET 文 367
GLOBAL 句 257
GOBACK 文 384
GO TO MORE-LABELS 文 507
GO TO 文 (数字項目のけた拡張機能) 407
GRID 句 217
GROUP BY 句 168
GROUP-USAGE 句 (日本語機能) 39

H

HAVING 句 168
HELP WINDOW 句 258
HIGHLIGHT 句 217
HIGH-INTENSITY 句 260
HiRDB による索引ファイル入出力機能 378
HiRDB による索引ファイル入出力機能〔概説〕 379
HITAC8000 シリーズからの移行用オプション 135, 473

I

IF 文 (数字項目のけた拡張機能) 407
IF 文 (動的長基本項目機能) 439
INITIALIZE 文 47, 98, 117
INITIALIZE 文 (数字項目のけた拡張機能) 407
INITIALIZE 文 (動的長基本項目機能) 439
INSERT 文 181
INSPECT 文 47, 98
INSPECT 文 (数字項目のけた拡張機能) 408
INVOKE 文 117, 344
INVOKE 文 (オブジェクト指向) (数字項目のけた拡張機能) 408
IN 述語 164

J

JUSTIFIED 句 (SCREEN SECTION) 218
JUSTIFIED 句 (WINDOW SECTION) 260
JUSTIFIED 句 (日本語機能) 41
JUSTIFIED 句 (ブール形式データを扱う機能) 87
JUSTIFIED 句の処理〔第1次規格以前の仕様〕 458

L

LABEL RECORD 句のデータ名指定 493
LABEL-RETURN 特殊レジスタ 484
LEFTLINE 句 218
LENGTH 関数 (動的長基本項目機能) 441
LENGTH-OF-SUBSTRING 関数 450, 523
LENGTH-OF-VARIANT 関数 358
LETTER 句 261

LIKE 述語 164
LINAGE 句 (数字項目のけた拡張機能) 400
LINE NUMBER 句 126, 263
LINE 句 218
LINE-COUNTER 特殊レジスタ 120
LOCK MODE 指定 62
LOWLIGHT 句 219

M

MODIFY 文 367
MOVE 文 48, 98, 117
MOVE 文 (数字項目のけた拡張機能) 408
MOVE 文 (動的長基本項目機能) 439
MOVE 文 (日付と時刻用) 508
MOVE 文の位取り 455
MULTIPLY 文 118
MULTIPLY 文 (数字項目のけた拡張機能) 409

N

NATIONAL-OF 関数 450, 527
NEXT GROUP 句 128
NEXT 指定 75
NULLIFY 文 368
null value 150
NULL 述語 165

O

OCCURS 句 87, 494
OCCURS 句 (数字項目のけた拡張機能) 400
OLE2 オートメーションインタフェース機能 336
OLE2 オートメーションインタフェース機能〔概説〕 337
OLE オブジェクト 337
OLE オブジェクト参照データ項目 338, 341
OLE オブジェクト参照データ比較の条件式 351
OLE プロパティ 337
OLE プロパティからの値の取得 348
OLE プロパティへの値の設定 348
OLE メソッド 337

OLE メソッドの操作 344
ON SIZE ERROR/NOT ON SIZE ERROR 指定 113
ON SIZE ERROR 指定 539
OpenTP1 138
OPEN 文 (CSV ファイル入出力機能) 329
OPEN 文 (HiRDB による索引ファイル入出力機能) 384
OPEN 文 (データベースアクセス機能) 183
OPEN 文 (テキストファイル機能) 56
OPEN 文 (ファイル共用機能) 63
ORGANIZATION IS CSV 327
ORGANIZATION IS LINE SEQUENTIAL 句 53
ORGANIZATION 句 380
OVERLINE 句 219

P

PAGE FOOTING 指定 131
PAGE HEADING 指定 130
PAGE LIMIT 句 124
PAGE-COUNTER 特殊レジスタ 120
PASCAL 487
PASSWORD 句 489
PERFORM 文 118
PERFORM 文 (数字項目のけた拡張機能) 409
PERFORM 文での評価順序 458
PERFORM 文の実行範囲の出口の共有 510
PICTURE 記号 495
PICTURE 句 (COMP-X 項目) 536
PICTURE 句 (SCREEN SECTION) 220
PICTURE 句 (WINDOW SECTION) 267
PICTURE 句 (アドレス操作機能) 309
PICTURE 句 (数字項目のけた拡張機能) 400
PICTURE 句 (日本語機能) 42
PICTURE 句 (ブール形式データを扱う機能) 87
PICTURE 句 (浮動小数点形式データを扱う機能) 108
POINTER 項目 483
PREPARE 文 185
PREVIOUS 指定 75
PROMPT 句 221

R

READ 文 (CSV ファイル入出力機能) 329
READ 文 (HiRDB による索引ファイル入出力機能) 385
READ 文 (索引ファイル機能の拡張) 75
READ 文 (数字項目のけた拡張機能) 410
READ 文 (テキストファイル機能) 56
READ 文 (ファイル共用機能) 64
READ 文による入出力 327
RECEIVE 文 143, 199
RECONNECT 文 368
RECORDING MODE 句 493
RECORD KEY 句 72, 380
RECORD 句 122
REDEFINES 句 88
RELEASE 文 509
RELEASE 文 (数字項目のけた拡張機能) 410
RENAMES 句 89
REPEAT COLUMN 句 269
REPEAT LINE 句 272
REPLACING 指定 456
REPLY 文 290
REPORT FOOTING 指定 131
REPORT 句 122
REQUIRED 句 222
RERUN 句 492
RESERVE 句の ALTERNATE 指定 489
RESET 句 128, 275
RETURNING 322, 324
RETURN 文 (数字項目のけた拡張機能) 410
RETURN-CODE 483
REVERSE 句 276
REVERSE-VIDEO 句 223
REWRITE 文 56, 66, 385, 510
REWRITE 文 (数字項目のけた拡張機能) 411
ROLLBACK 文 144, 178, 385
ROUNDED 指定 113
RULE LINE 句 276

S

SCREEN SECTION (数字項目のけた拡張機能) 398
SEARCH 文 49, 118
SEARCH 文 (数字項目のけた拡張機能) 411
SECRET 句 279
SECURE 句 223
SELECT OPTIONAL 句 454
SELECT 句の NOT OPTIONAL 指定 490
SELECT 文 180
SEND 文 144, 198
SET CONNECTION 文 177
SET 文 291, 348, 499
SET 文 (数字項目のけた拡張機能) 411
SIGN 句 224, 279
SIZE ERROR 指定 403
SIZE 句 224
SORT-CORE-SIZE 484
SORT-RETURN 484
SOURCE 句 128, 280
SPECIAL-NAMES [CSV ファイル入出力機能] 333
SPECIAL-NAMES [WINDOW SECTION] 243
SPECIAL-NAMES [コマンド行, 環境変数へのアクセス] 189
SPECIAL-NAMES [コマンド行へのアクセス (一括取得)] 192
SPECIAL-NAMES [プログラム間連絡機能] 486
SQLCODE 変数 155, 375
SQL 埋め込み COBOL プログラム 146
SQL の概要 149
SQL の規則と COBOL 言語の規則との異なる部分 150
SQL 連絡領域 375
START 文 66, 78, 386
START 文 (数字項目のけた拡張機能) 412
STDCALL 487
STOP RUN 文 386
STOP 文 (数字項目のけた拡張機能) 412
STORE 文 368
STRING 文 49
STRING 文 (数字項目のけた拡張機能) 412

SUBSCHEMA SECTION 361
SUBSTRING 関数 450, 529
SUBTRACT 文 118
SUBTRACT 文 (数字項目のけた拡張機能) 412
SUM 句 128
SUPPRESS 文 133
SYNCHRONIZED 句 89, 109, 310, 536
SYNCHRONIZED 句を集団項目に指定 496

T

TRACK-AREA 句 489
TRANSCEIVE 文 199
TRANSFORM 文 511
TRANSFORM 文 [数字項目のけた拡張機能] 424
TURN 指令 (数字項目のけた拡張機能) 389
TYPE 句 130
TYPE-OF-PROPERTY 関数 354
TYPE-OF-VARIANT 関数 355

U

UNDERLINE 句 225
Unicode 機能 447
UNLOCK 文 67, 386
UNSTRING 文 50
UNSTRING 文 (数字項目のけた拡張機能) 413
UNSTRING 文の DELIMITED BY ALL 指定 455
UPDATE 文 182
USAGE COMP-5 を指定した項目に設定できる初期値 496
USAGE IS POINTER 句 496
USAGE 句 (COMP-X 項目) 536
USAGE 句 (SCREEN SECTION) 225
USAGE 句 (WINDOW SECTION) 280
USAGE 句 (数字項目のけた拡張機能) 401
USAGE 句 (日本語機能) 43
USAGE 句 (ブール形式データを扱う機能) 91
USAGE 句 (浮動小数点形式データを扱う機能) 109
USAGE 句の ADDRESS 指定 310
USE BEFORE REPORTING 文 134

USE LABEL 文 512
USE 文 134, 294, 512
USING 句 281

V

VALUE 句 (COMP-X 項目) 537
VALUE 句 (SCREEN SECTION) 225
VALUE 句 (WINDOW SECTION) 281
VALUE 句 (数字項目のけた拡張機能) 401
VALUE 句 (日本語機能) 44
VALUE 句 (ブール形式データを扱う機能) 91
VALUE 句 (浮動小数点形式データを扱う機能) 110
VALUE 句 (報告書作成機能の拡張) 133
VARIANT 値 338
VARIANT 値から COBOL データへの変換規則 339
VARIANT 値と COBOL データの変換規則 338
VARIANT-TO-ALPHANUMERIC 関数 357
VARIANT-TO-INTEGGER 関数 356
VARIANT-TO-NUMERIC 関数 356
VARIANT-TO-OBJECT 関数 357
VOS3 COBOL85 からの移行用オプション 120, 462

W

WAIT 文 297
WHERE 句 168
WINDOW SECTION (数字項目のけた拡張機能) 398
WINDOW SECTION [概説] 235
WINDOW-CURSOR 特殊レジスタ 239
WINDOW-KEYCODE 特殊レジスタ 238
WINDOW-KEY 特殊レジスタ 237
WINDOW-STATUS 特殊レジスタ 237
WRITE 文 (CSV ファイル入出力機能) 330
WRITE 文 (HiRDB による索引ファイル入出力機能) 386
WRITE 文 (印刷制御文字の直接指定, 区域外書き出し) 513
WRITE 文 (数字項目のけた拡張機能) 413
WRITE 文 (定数 1 指定) 515
WRITE 文 (テキストファイル機能) 57

WRITE 文 (ファイル共用機能) 67
WRITE 文による入出力 327
WRITE 文 [数字項目のけた拡張機能] 425

Z

ZERO-FILL 句 226

あ

相手指定 160
値式 162
値指定 160
アドレス型関数 316
アドレス操作機能 305
アドレス操作機能とは 306
アドレスデータ項目 307
アドレスデータ項目の参照 312
アドレスデータ項目の定義 309
アドレス名 307
アドレス名によって参照されるデータ項目の定義 308
アドレス名の参照 312

い

一意参照 (数字項目のけた拡張機能) 393
一意名のいろいろ (数字項目のけた拡張機能) 394
位置づけ 181, 182
インスタンス 337

う

埋め込み SQL 宣言節 (データベースアクセス機能) (数字項目のけた拡張機能) 402
埋め込み SQL 文 147
埋め込み SQL 文 [数字項目のけた拡張機能] 425
埋め込み変数 156, 374
埋め込み例外宣言 170

え

英数字定数 465
英数字フィールド 241
演算の中間結果 539

か

カーソル 179
カーソル宣言 172
外部ブール項目 85
外部浮動小数点形式 103
外部浮動小数点形式の初期値 110
外部プログラム節 486
拡張仕様 (数字項目のけた拡張機能) 417
格納できる数値 535
下線付き定数の呼び出し 502
かたかな 481
可変長項目の複数回記述 494
画面記述項 205
画面節 (SCREEN SECTION) 205
画面節 (SCREEN SECTION) による画面機能 201
画面節 (WINDOW SECTION) による画面機能 233
画面入出力機能特殊レジスタ 237
画面フィールドの種類とそのサイズ 239
環境部 486
環境部 (CSV ファイル入出力機能) 327
環境部 (HiRDB による索引ファイル入出力機能) 380
環境部 (SCREEN SECTION) 202
環境部 (WINDOW SECTION) 243
環境部 (索引ファイル機能の拡張) 70
環境部 (書式印刷機能) 299
環境部 (数字項目のけた拡張機能) 395
環境部 (テキストファイル機能) 53
環境部 (動的長基本項目機能) 432
環境部 (日本語機能) 38
環境部 (ファイル共用機能) 60
環境部 (報告書作成機能の拡張) 121
環境部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能) 374
環境変数へのアクセス 189
関数値 (数字項目のけた拡張機能) 414

き

記憶方式 535
記述項名句 255

機能概要 69

旧規格仕様を使う場合 454
行送り制御文字領域削除 466
行と文字位置 36, 465
行の継続 36
局所場所節 (数字項目のけた拡張機能) 397

<

区切り文字 373
組み込み関数 450, 516
組み込み関数 (数字項目のけた拡張機能) 414
組み込み関数 (動的長基本項目機能) 441
組み込み関数機能 353
組み込み関数の表現形式 359
グループ表 149

け

けたあふれ条件 403
結果 150
言語の概念 326, 337
言語の基本概念 237
検索条件文 369
原始文操作機能 375
原始文操作機能の特性 466
限定述語 166

こ

合成キー 69
構造型データベース (XDM/SD) 操作シミュレーション機能 360, 361
項目間の転記の規則 98
項類 34
コーディング規則 36, 465
語長の境界と効率 107
コネクション 174
コマンド行へのアクセス 188
コマンド行へのアクセス (一括取得) 192
コミット 174
コレクション 337

コンテナ 338
コンパイラオプション 442, 452
コンパイラオプションによる仕様の相違 451
コンパイラ環境変数 442
コンパイラ環境変数 CBLVALUE 442

さ

作業場所節 (数字項目のけた拡張機能) 396
索引ファイル機能の拡張 68
索引ファイルと COMP-X 項目 540
サブスキーマ記述項 361
サブスキーマ節 (数字項目のけた拡張機能) 398
算術式 112
算術符号 105
算術文 404
算術文の作用対象のけた数 498

し

実表 149
集合関数指定 161
述語 162
主プログラムの指定コマンドインタフェース 475
条件式 94, 112
条件式 [数字項目のけた拡張機能] 425
条件名 110
条件名条件 95, 112
使用例 [動的長基本項目機能] 443
書式印刷機能 298
書体 300
字類 34
字類条件 95, 112

す

数字項目のけた拡張機能 387
数字項目の転記/比較 473
数字項類 (数字項目のけた拡張機能) 391
数字定数 (数字項目のけた拡張機能) 390
数字フィールド 240
ストアドプロシージャ 180

せ

制御脚書き報告集団 132
制御用データ項目 132
静的に行う方法 179
整列併合機能と COMP-X 項目 540
整列併合用特殊レジスタ 484
全般規則 30, 150, 373, 481
全般規則 [動的長基本項目機能] 431

そ

操作系とする SQL 文 179
ソース単位 (数字項目のけた拡張機能) 388

た

第 1 次規格以前の仕様 454
第 2 次規格以前の仕様 458
対象となる組み込み関数 [動的長基本項目機能] 429
対象となる文 [動的長基本項目機能] 429
濁点 481
探索 181, 182
探索条件 166
探索条件文 [数字項目のけた拡張機能] 424
単純条件 46

ち

注意事項 377
注記行 36

つ

通信記述項 139, 195
通信節による画面機能 194

て

定義できる節 [動的長基本項目機能] 428
定義できるソース単位 [動的長基本項目機能] 428
定数 32, 465
定数 (数字項目のけた拡張機能) 390
定数名での VALUE 句の規則 496
データ記述項 245, 494

データ項目 (数字項目のけた拡張機能)	391	データベース条件文	371
データ項目のサイズ	239	データベース操作シミュレーション機能	360
データ項目の初期化	348	テキストファイル機能	51
データ項目の字類と項類	34	テキスト編成	52
データコミュニケーション機能	137	手続き部	498
データコミュニケーション機能とは	138	手続き部 (CSV ファイル入出力機能) [ACCEPT 文・DISPLAY 文による入出力]	333
データの配置	107	手続き部 (CSV ファイル入出力機能) [READ 文・WRITE 文による入出力]	328
データの表現形式	85, 103, 307	手続き部 (C 言語インタフェース機能)	322
データ配置の方法	86, 107	手続き部 (HiRDB による索引ファイル入出力機能)	383
データ部	493	手続き部 (OLE2 オートメーションインタフェース機能)	344
データ部 (CSV ファイル入出力機能)	327	手続き部 (SCREEN SECTION)	227
データ部 (C 言語インタフェース機能)	320	手続き部 (WINDOW SECTION)	283
データ部 (HiRDB による索引ファイル入出力機能)	381	手続き部 (アドレス操作機能)	312
データ部 (OLE2 オートメーションインタフェース機能)	341	手続き部 (構造型データベース (XDM/SD) 操作シミュレーション機能)	363
データ部 (WINDOW SECTION)	245	手続き部 (コマンド行, 環境変数へのアクセス)	189
データ部 (アドレス操作機能)	308	手続き部 (コマンド行へのアクセス (一括取得))	192
データ部 (構造型データベース (XDM/SD) 操作シミュレーション機能)	361	手続き部 (索引ファイル機能の拡張)	75
データ部 (索引ファイル機能の拡張)	74	手続き部 (数字項目のけた拡張機能)	403
データ部 (書式印刷機能)	300	手続き部 (通信節による画面機能)	198
データ部 (数字項目のけた拡張機能)	396	手続き部 (データコミュニケーション機能)	142
データ部 (通信節による画面機能)	195	手続き部 (テキストファイル機能)	56
データ部 (データコミュニケーション機能)	139	手続き部 (動的長基本項目機能)	437
データ部 (テキストファイル機能)	55	手続き部 (日本語機能)	46
データ部 (動的長基本項目機能)	434	手続き部 (ファイル共用機能)	63
データ部 (日本語機能)	39	手続き部 (ブール形式データを扱う機能)	93
データ部 (ブール形式データを扱う機能)	87	手続き部 (浮動小数点形式データを扱う機能)	112
データ部 (浮動小数点形式データを扱う機能)	108	手続き部 (報告書作成機能の拡張)	133
データ部 (報告書作成機能の拡張)	121	手続き部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	376
データ部 (リレーショナルデータベース (XDM/RD) 操作シミュレーション機能)	374	手続き部 [COMP-X 項目]	537
データ部 [COMP-X 項目]	535	手続き部 [データベースアクセス機能]	154
データ部 [データベースアクセス機能]	154	データベース	149
データベース	149	データベースアクセス機能	145, 146
データベースアクセス機能	145, 146	データベースアクセス機能 [概説]	146
データベースアクセス機能 [概説]	146	データベースアクセス機能 [概念]	149
データベースアクセス機能 [概念]	149	データベースアクセス機能と COMP-X 項目	540
データベースアクセス機能と COMP-X 項目	540		

手続き部見出し 118, 322, 352
手続き部見出し〔数字項目のけた拡張機能〕 404
デバッグ行の有効, 無効 474
転記の規則 537
転記の長さのチェック 476

と

問い合わせ式 169
問い合わせ指定 169
導出表 149
動的 SQL 180
動的長基本項目 427
動的長基本項目が使用できる個所 428
動的長基本項目機能 426
動的長基本項目を使用したときの注意事項 446
動的に行う方法 180
特殊名段落 (動的長基本項目機能) 432
特殊名段落 (日本語機能) 38
特殊名段落 [SCREEN SECTION] 202
特殊名段落 [WINDOW SECTION] 243
特殊名段落の CURRENCY SIGN 句 454
特殊名段落 [プログラム間連絡機能] 486
特殊名段落 [報告書作成機能] 121
特殊レジスタ 120, 376, 483
トランザクション 174

な

内部 10 進項目指定 498
内部ブール項目 85
内部浮動小数点形式 103
内部浮動小数点形式の初期値 110
ナル値 150

に

日本語機能 28
日本語機能とは 29
日本語項目に対して埋められる空白の扱い 467
日本語項目の扱い 468
日本語フィールド 241

日本語文字集合 30, 32
日本語文字定数 32, 465
日本語文字の扱い 463
日本語文字の継続 466
入出力管理記述項 299, 490
入出力管理段落 243
入出力状態の値 58, 332

は

バリエーションデータ項目 338, 342
バリエーションデータ項目の設定 348
バリエーションデータ比較の条件式 351
半濁点 481

ひ

比較述語 163
比較条件 94, 112, 312, 539
比較の組み合わせ 539
引数 (数字項目のけた拡張機能) 414
引数の適合 [動的長基本項目機能] 437
ビュー表 149
表 149
表意定数 33, 84, 102, 483
表意定数 [動的長基本項目機能] 431
表式 167
標識変数 158
標準けた寄せ規則 34, 85, 106

ふ

ファイル管理記述項 70, 487
ファイル管理段落 53, 327, 380
ファイル記述項 493
ファイル記述項 (テキストファイル機能) 55
ファイル記述項 (ファイル共用機能) 60
ファイル記述項 [CSV ファイル入出力機能] 328
ファイル記述項 [HiRDB による索引ファイル入出力機能] 381
ファイル記述項 [報告書作成機能の拡張] 122
ファイル共用機能 59

ファイル施錠 63
ファイル節 (数字項目のけた拡張機能) 396
フィールド状態条件 283
ブール演算子 93
ブール形式データが書ける文 95
ブール形式データを扱う機能 82
ブール項目 85
ブール項目の指定方法 88
ブール式 93
ブール式の計算順序 94
ブール式の定義 93
ブール定数 83
副問い合わせ 168
符号系名句 38
浮動小数点形式データが書ける文 115
浮動小数点形式データを扱う機能 100
浮動小数点数字定数 101
プログラム間連絡, 利用者定義関数, メソッドの USING 指定と返却項目 99
プログラム間連絡機能 352
プログラム間連絡機能および利用者定義関数の引数と返却項目 312
プログラム間連絡機能の環境部 486
プログラム間連絡用特殊レジスタ 483
文に共通な規則 93
文に共通な指定と機能 [数字項目のけた拡張機能] 403
分離符 34
分離符の選択 462

ほ

ポインタ項目の参照 498
ポインタ比較の条件式 498
報告集団記述項 124
報告書記述項 123
報告書作成機能の拡張 119
報告書節 (数字項目のけた拡張機能) 397

み

見出し部 37, 374

め

明細報告集団 132

も

文字間隔 300
文字サイズ 300
文字集合 463
文字列 30, 464

よ

横倍角開始 300
予約語 373

り

略記組み合わせ比較条件での NOT の取り扱い 454
利用者定義語 30, 464
リレーショナルデータベース (XDM/RD) 操作シミュレーション機能 360, 373

れ

レコード下降順アクセス機能 69
レコード施錠 62
列指定 160
連絡節 (数字項目のけた拡張機能) 397

ろ

ロールバック 174

わ

割り当てられる記憶域 535