

COBOL

プログラミングの手引

はしがき

本書は、オペレーティングシステム Windows および Linux のもとで使用される COBOL のプログラミング技法について、説明したものです。

本書は、COBOL の利用者が、原始プログラムを翻訳しリンクを行い、実行可能プログラムを実行するまでの操作方法について説明したものです。また、『COBOL 言語説明書』の仕様に基づいたプログラミングについて補足説明し、効率のよい COBOL プログラムの組み方について説明しています。COBOL の文法については、『COBOL 言語説明書』を参照してください。

本書の構成は次のとおりです。

章	タイトル	内 容
1	原始プログラムの翻訳, リンク	COBOL の操作法に関する説明
2	外部リポジトリ機能	外部リポジトリ機能の利用方法に関する説明
3	ファイル入出力機能	ファイル入出力機能の使用法に関する説明
4	拡張ファイル入出力機能	拡張ファイル入出力機能の使用法に関する説明
5	中核に関する補足説明	中核機能の使用法に関する補足説明
6	表操作機能	表操作機能の使用法に関する説明
7	プログラム間連絡機能	プログラム間連絡機能の使用法に関する説明
8	利用者定義関数	利用者定義関数機能の使用法に関する説明
9	オブジェクト指向機能	オブジェクト指向機能の使用法に関する説明
10	整列併合機能	整列併合機能の使用法に関する説明
11	翻訳指示機能	翻訳指示機能の使用法に関する説明
12	例外処理機能	例外処理機能に関する説明
13	システムサブルーチン	システムサブルーチンに関する説明
14	プログラミング技法	プログラミング技法に関する説明
15	翻訳リスト	翻訳リストに関する説明
16	データ例外処理機能	データ例外処理機能に関する説明
17	利用者制御変数処理機能	利用者制御変数処理機能に関する説明
18	日本語情報処理機能	日本語情報処理機能に関する説明
19	64 ビットアプリケーションの生成方法	64 ビットアプリケーションの生成方法の説明
20	マルチスレッドプログラミング	マルチスレッドプログラムの作成方法の説明
21	関連製品連携	COBOL 関連製品との連携に関する説明
22	画面制御機能	画面制御機能に関する説明
23	データベース機能	データベース機能に関する説明
24	実行トレース機能	実行トレース機能に関する説明
25	用語機能	用語機能に関する説明
26	ファイル入出力レコード情報生成機能	ファイル入出力レコード情報生成機能に関する説明
27	データレコード情報出力機能	データレコード情報出力機能に関する説明

COBOL の説明書として本書以外に次のものがあります。

- 言語説明書 :『COBOL 言語説明書』
- ユーザーズガイド :『COBOL ユーザーズガイド』
- COBOL 開発環境 :『COBOL 開発環境 利用の手引』

オプション製品および関連製品の説明書として次のものがあります。

- 索引ファイルの利用方法 :『ISAM 利用の手引』
- COBOL SQL アクセスの利用方法 :『COBOL SQL アクセス 言語説明書』
『COBOL SQL アクセス プログラミングの手引』
『COBOL SQL アクセス ユーザーズガイド』

次は, Standard Edition 版および Enterprise Edition for Windows 版にて利用できます。

- 整列併合機能の利用方法 :『SORTKIT/Enterprise 説明書』

次は, Standard Edition 版および Enterprise Edition for Windows 版の SE 動作モードにて利用できます。

- COBOL DB アクセスの利用方法 :『COBOL DB アクセス for Oracle 説明書』
『COBOL DB アクセス for SQL Server 説明書』
- COBOL アプリケーション実行環境の利用方法 :
『COBOL アプリケーション実行環境 ジョブ管理機能 説明書』
『COBOL アプリケーション実行環境 プリンタ機能 説明書』
『COBOL WS エミュレータ 説明書』
- COBOL 拡張システムサブルーチンの利用方法 :
『COBOL 拡張システムサブルーチン 説明書』

次は, Standard Edition 版にて利用できます。

- IFAS FILE :『IFAS FILE, IFASPRO RDB 利用の手引』
- IFASPRO RDB の利用方法 :『IFAS FILE, IFASPRO RDB 利用の手引』
- COBOL GUI の利用方法 :『COBOL GUI ユーザーズガイド』
『COBOL GUI 画面構築ガイド』

次は, Enterprise Edition 版にて利用できます。

- RAF ファイルの利用方法 :『refam 利用の手引』

次は, Enterprise Edition for Linux 版にて利用できます。

- 整列併合機能の利用方法
- 用語機能の利用方法

:『SORTKIT 利用の手引』

:『COBOL 用語エディタ リファレンスマニュアル』

2012 年 5 月 初 版
2012 年 10 月 第 2 版
2013 年 5 月 第 3 版
2013 年 11 月 第 4 版
2014 年 10 月 第 5 版
2015 年 4 月 第 6 版
2015 年 10 月 第 7 版
2016 年 4 月 第 8 版
2017 年 4 月 第 9 版
2018 年 10 月 第 10 版
2018 年 12 月 第 11 版
2019 年 4 月 第 12 版
2020 年 4 月 第 13 版
2021 年 4 月 第 14 版
2022 年 4 月 第 15 版
2023 年 4 月 第 16 版
2024 年 4 月 第 17 版
2025 年 6 月 第 18 版
2026 年 6 月 第 19 版

ご注意

- (1) 本書の内容の一部または全部を無断転載することは禁止されています。
- (2) 本書の内容に関しては将来予告なしに変更することがあります。
- (3) 本書は内容について万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなどお気づきのことがありましたら、ご連絡ください。
- (4) 運用した結果の影響については、(3)項にかかわらず責任を負いかねますのでご了承ください。

Microsoft, Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle, Pro*COBOL は米国 Oracle Corporation の登録商標です。

Linux は Linus Torvalds の米国およびその他の国における商標または登録商標です。

Red Hat, Red Hat Enterprise Linux は米国 Red Hat, Inc. の米国およびその他の国における商標または登録商標です。

Apache, Xerces-C++は、The Apache Software Foundation の商標または登録商標です。

その他、本書に記載の製品名は、各社の商標または登録商標です。

輸出する際の注意事項

本製品（ソフトウェア）は日本国内仕様であり、外国の規格等には準拠しておりません。
本製品は日本国外で使用された場合、当社は一切責任を負いかねます。また、当社は本製品に関して海外での保守サービスおよび技術サポート等は行っておりません。

備考 (1) 本書は次のオペレーティングシステムに対応しています。

【Enterprise for Windows】Windows Server 2022,

Windows Server 2025, Windows 10, Windows 11

【Enterprise for Linux】Red Hat Enterprise Linux 7,

Red Hat Enterprise Linux 8

【Standard】Windows Server 2022,

Windows Server 2025, Windows 10, Windows 11

(2) 本書に使用しているすべての機能は、プログラムプロダクトであり、次のプロダクト名に対応しています。

COBOL

(3) 本書では、COBOL および関連製品の名称を次のように表記します。

製品名	マニュアル表記
SORTKIT/Enterprise	SORTKIT
SORTKIT/Enterprise Client	SORTKIT
COBOL ソート・マージ Client	SORTKIT
COBOL DB アクセス for Oracle	COBOL DB アクセス
COBOL DB アクセス for SQL Server	COBOL DB アクセス

本書の記述について

本書では、次の記号を用いて、Enterprise Edition for Windows 版、Enterprise Edition for Linux 版、および Standard Edition 版の機能差異を説明します。

また、Enterprise Edition for Windows 版の標準動作モードおよび SE 動作モードの機能差異を説明します。

標準動作モードは、ACOS-4 COBOL85 や COBOL85 for IPF からの移行に適したモードであり、Enterprise Edition for Windows 版の動作既定値です。

SE 動作モードは、A-VX COBOL85 や COBOL85 Pro からの移行に適したモードであり、Standard Edition 版と動作互換です。

【Enterprise】記号は、次の機能を表します。

Enterprise Edition for Windows 版の標準動作モードと Enterprise Edition for Linux 版の共通機能

【Enterprise for Windows】記号は、次の機能を表します。

Enterprise Edition for Windows 版の標準動作モードの機能

【Enterprise for Linux】記号は、次の機能を表します。

Enterprise Edition for Linux 版の機能

【Standard】記号は、次の機能を表します。

Standard Edition 版および Enterprise Edition for Windows 版の SE 動作モードの機能
本書で「シフト JIS」と記述している文字コードは「cp932」です。

目 次

第 1 章 原始プログラムの翻訳, リンク	1-1
1.1 原始プログラムの作成から実行の概要.....	1-1
1.2 原始プログラムの作成および修正.....	1-2
1.3 登録集原文の作成および修正.....	1-5
1.4 原始プログラムの翻訳(コンパイル)およびプログラムのリンク.....	1-5
1.4.1 翻訳用コマンドプロンプト.....	1-5
1.4.2 翻訳用コマンドとその使用法.....	1-6
1.4.3 同一翻訳群の原始プログラムの翻訳時の注意点.....	1-36
1.4.4 予約語切り替え機能.....	1-37
1.5 プログラムのリンク.....	1-37
1.5.1 実行可能プログラムの生成.....	1-37
1.5.2 リンク用コマンドの指定方法.....	1-40
1.5.3 マニフェストの埋め込み.....	1-44
第 2 章 外部リポジトリ機能	2-1
2.1 リポジトリ情報	2-1
2.1.1 外部リポジトリの利用方法.....	2-3
2.2 外部リポジトリ操作.....	2-6
2.2.1 オプション動作と既定動作.....	2-6
2.2.2 利用上の注意.....	2-8
2.3 外部リポジトリのファイル名生成規則.....	2-16
2.4 適合検査について.....	2-22
2.5 既存オプションの影響.....	2-23
第 3 章 ファイル入出力機能	3-1
3.1 ファイルの装置およびファイル指定.....	3-1
3.2 レコード形式	3-8
3.2.1 固定長レコード形式.....	3-9
3.2.2 可変長レコード形式.....	3-9
3.3 プリンタファイル.....	3-10
3.3.1 事前準備.....	3-10
3.3.1.1 PRNAMR とプリントマネージャの機能差異	3-10
3.3.1.2 PRNAMR による印字方法	3-11

3.3.1.3	プリントマネージャによる印字方法	3-11
3.3.1.4	PRNAMR とプリントマネージャを併用する印字方法	3-12
3.3.1.5	プリンタの動作モード指定方法	3-12
3.3.2	レジストリ(プリンタ関連)	3-13
3.3.3	プリンタファイルの装置およびファイル指定	3-14
3.3.4	ファイル形式	3-15
3.3.5	論理ページと改行、改ページの動作	3-17
3.3.5.1	改ページと論理ページとの関係	3-17
3.3.5.2	脚書き領域を使用したページ制御	3-19
3.3.6	入出力文	3-21
3.3.7	プリンタ共用機能	3-22
3.3.7.1	プリンタ共用機能の条件	3-23
3.3.7.2	プリンタ共用機能の動作	3-24
3.3.8	スプールファイルの出力機能	3-25
3.3.9	フォームオーバーレイ機能	3-28
3.3.9.1	PRNAMR 経由の印字におけるフォームの使用	3-28
3.3.9.2	フォームオーバーレイ機能使用例	3-30
3.3.10	PASSTHROUGH モード機能	3-36
3.3.10.1	使用方法	3-36
3.3.10.2	使用例	3-37
3.3.10.3	制限事項	3-38
3.4	大容量記憶ファイルの排他／共用	3-39
3.4.1	LOCK MODE 句によるファイルの共用／排他	3-39
3.4.2	ファイルの排他利用	3-39
3.4.3	レコード排他制御	3-40
3.5	大容量記憶ファイル	3-41
3.5.1	レジストリ(ファイルマップ機能)	3-41
3.6	大容量記憶ファイル(順ファイル)	3-43
3.6.1	入出力文	3-43
3.6.2	順ファイルの入出力文の動作	3-45
3.7	大容量記憶ファイル(行順ファイル)	3-47
3.7.1	入出力文	3-47
3.7.2	行順ファイルの入出力文の動作	3-51
3.7.3	行順ファイルのファイル構成および使用上の留意点	3-53
3.8	大容量記憶ファイル(行順ファイル(CSV 形式))	3-54
3.8.1	入出力文	3-54

3.8.2	行順ファイル(GSV 形式)における入出力文の動作	3-56
3.8.3	行順ファイル(GSV 形式)の動作例	3-57
3.8.4	行順ファイル(GSV 形式)の使用上の留意点	3-61
3.9	大容量記憶ファイル(相対ファイル)	3-63
3.9.1	入出力文	3-63
3.9.2	相対ファイルの入出力文の動作	3-68
3.9.3	相対ファイルのファイル構成	3-71
3.10	大容量記憶ファイル(索引ファイル)	3-71
3.10.1	入出力文	3-71
3.10.2	索引ファイルの入出力文の動作	3-78
3.10.3	二重キー	3-83
3.10.4	索引ファイルのファイル構成および使用上の留意点	3-85
3.11	APPLY 句によるファイルの共用／排他(互換機能)	3-86
3.12	入出力状態	3-105
3.12.1	状態キー1	3-105
3.12.2	状態キー2	3-106
3.12.3	状態キー1 と状態キー2 の組み合わせ	3-106
3.12.4	入出力状態の検査	3-110
3.13	コード変換機能	3-112
3.13.1	CODE-SET 句	3-113
3.13.2	READ 文	3-115
3.13.3	REWRITE 文	3-125
3.13.4	WRITE 文	3-125
3.14	GLOBAL 句付きのファイル	3-126
3.15	内部ファイル ID 置換機能	3-127
3.15.1	使用方法	3-128
3.15.2	内部ファイル ID 置換表	3-131
3.15.3	使用例	3-133
3.15.4	注意制限事項	3-134
第4章 拡張ファイル入出力機能		4-1
4.1	概説	4-1
4.2	ファイルの種類	4-1
4.3	ファイルの定義	4-1
4.3.1	ファイル識別リテラル	4-1
4.3.2	ファイルの装置およびファイル指定	4-2

4.4	ファイルの割り当て.....	4-3
4.5	エラー処理	4-8
4.5.1	FILE STATUS 句	4-8
4.5.2	入出力状態値と意味.....	4-9
4.5.3	入出力動作状態の調べ方.....	4-14
4.5.4	AT END 句	4-17
4.5.5	入出力誤り手続き.....	4-19
4.5.6	入出力誤り状態.....	4-21
4.6	レコード形式	4-21
4.6.1	固定長レコード.....	4-22
4.6.2	可変長レコード.....	4-26
4.7	レコード長／ブロック長.....	4-30
4.8	RAF ファイル.....	4-31
4.8.1	削除レコード.....	4-31
4.8.2	順ファイル.....	4-32
4.9	SYSIN/SYSOUT ファイル	4-36
4.9.1	SYSIN ファイル	4-37
4.9.1.1	レコード形式.....	4-37
4.9.1.2	SYSIN ファイルの定義	4-37
4.9.1.3	LABEL RECORD 句.....	4-38
4.9.1.4	SYSIN ファイルの入出力文	4-38
4.9.1.5	SYSIN ファイル使用上の注意.....	4-38
4.9.2	SYSOUT ファイル	4-39
4.9.2.1	レコード形式.....	4-39
4.9.2.2	制御データ	4-40
4.9.2.3	SSF ヘッダ.....	4-40
4.9.2.4	SYSOUT ファイルの定義.....	4-41
4.9.2.5	LABEL RECORD 句.....	4-41
4.9.2.6	SYSOUT ファイルの入出力文	4-42
4.9.2.7	SYSOUT ファイル使用上の注意.....	4-43
4.9.3	LINAGE 句付きのファイル	4-43
4.10	小入出力技法	4-46
4.10.1	ACCEPT 文	4-46
4.10.2	DISPLAY 文	4-50
4.11	その他の補足事項.....	4-53
4.11.1	EXTERNAL 句付きのファイル	4-53

4. 11. 2	BASED 句付きのファイル	4-53
4. 11. 3	READ INTO 文	4-56
第 5 章 中核に関する補足説明		5-1
5. 1	言語の概念	5-1
5. 1. 1	英大文字と英小文字の使い方	5-1
5. 1. 2	句読文字と分離符の使い方	5-3
5. 1. 3	英数字定数と 16 進英数字定数	5-5
5. 1. 4	部分参照	5-6
5. 2	データ部	5-8
5. 2. 1	PICTURE 句	5-8
5. 2. 2	REDEFINES 句	5-10
5. 2. 3	RENAMES 句	5-12
5. 2. 4	VALUE 句での注意事項	5-14
5. 2. 5	数字と符号の重ね合わせに関する注意事項	5-14
5. 3	手続き部	5-15
5. 3. 1	算術文における中間結果	5-15
5. 3. 2	CORRESPONDING 指定	5-23
5. 3. 3	COMBINE 文	5-27
5. 3. 4	IF 文と条件式	5-27
5. 3. 5	INSPECT 文	5-31
5. 3. 6	MOVE 文	5-34
5. 3. 7	STRING 文	5-39
5. 3. 8	UNSTRING 文	5-41
5. 4	小入出力機能	5-47
5. 4. 1	ACCEPT 文	5-47
5. 4. 2	DISPLAY 文	5-51
5. 4. 3	対話型アプリケーションまたは COBOL WS エミュレータ配下での小入出力機能の 注意事項	5-52
5. 5	特殊レジスタ EXCEPTION-CODE	5-52
5. 6	特殊レジスタ EXCEPTION-LINE	5-52
5. 7	特殊レジスタ COMPLETION-CODE	5-53
5. 8	特殊レジスタ FUNCTION-KEY	5-56
5. 9	特殊レジスタ RECORD-LENGTH	5-60
5. 10	特殊レジスタ RDB-STATUS	5-60
5. 11	特殊レジスタ RETURN-CODE	5-61

5.12 外部 SWITCH 機能.....	5-61
5.12.1 コマンドラインツール.....	5-61
5.12.1.1 cobinitcv	5-61
5.12.1.2 cobsetcv	5-62
5.12.1.3 cobreadcv.....	5-63
5.12.1.4 cobtermcv	5-63
5.12.2 使用例.....	5-64
5.12.3 COBOL アプリケーション実行環境下の使用	5-65
5.12.3.1 外部スイッチの設定(SETSSW).....	5-66
5.12.3.2 外部スイッチの設定(CHKSSW).....	5-66
第6章 表操作機能.....	6-1
6.1 概 説	6-1
6.2 表の定義	6-1
6.3 表の初期値	6-4
6.4 表の参照	6-5
6.4.1 指標名を使わない添字付け.....	6-5
6.4.2 指標名を使った添字付け.....	6-6
6.4.3 相対添字付け.....	6-7
6.4.4 添字付けにおけるアドレス計算.....	6-7
6.4.5 指標名を使わない添字付けと指標を使った添字付けの効率.....	6-9
6.5 指標名と指標データ項目.....	6-11
6.5.1 指標名と指標データ項目の内部表現形式.....	6-11
6.6 SEARCH 文.....	6-12
6.6.1 SEARCH(逐次表引き)文.....	6-12
6.6.2 SEARCH ALL(2分表引き)文	6-19
6.7 表の参照時の注意事項.....	6-24
第7章 プログラム間連絡機能.....	7-1
7.1 概 説	7-1
7.2 プログラムの分割.....	7-1
7.3 プログラムの入れ子.....	7-3
7.3.1 局所名と全域名.....	7-4
7.3.2 内部項目と外部項目.....	7-5
7.3.3 COBOL の利用者語の有効範囲	7-6
7.4 サブプログラムの呼び出し.....	7-6

7.4.1	含まれるプログラムの呼び出し	7-7
7.4.2	共通プログラムの呼び出し	7-8
7.4.3	外部プログラムの呼び出し	7-9
7.4.4	プログラム原型を利用した呼び出し	7-10
7.4.5	プログラムの再帰呼び出し	7-12
7.5	プログラム間のデータの受渡し	7-13
7.5.1	引数によるデータの受渡し	7-13
7.5.2	外部項目による受け渡し	7-15
7.5.3	全域名による受け渡し	7-25
7.6	サブプログラムからの復帰	7-26
7.7	宣言手続きの起動	7-27
7.8	プログラムの初期化	7-29
7.8.1	初期状態プログラム	7-30
7.9	CANCEL 文	7-30
7.10	静的リンクと動的リンク	7-31
7.11	動的リンク機能	7-32
7.11.1	Windows 版の動的リンク機能	7-32
7.11.1.1	COBOL プログラムの DLL 化	7-32
7.11.1.2	DLL を使用する実行ファイル生成例	7-32
7.11.1.3	DLL を使用する実行ファイルの実行方法	7-34
7.11.1.4	DLL の検索	7-35
7.11.1.5	注意事項	7-36
7.11.2	Linux 版の動的リンク機能	7-36
7.11.2.1	COBOL プログラムの共有ライブラリ化	7-36
7.11.2.2	共有ライブラリを使用する実行ファイル生成例	7-36
7.11.2.3	共有ライブラリを使用する実行ファイルの実行方法	7-38
7.11.2.4	共有ライブラリの検索	7-39
7.11.2.5	注意事項	7-40
7.12	C/C++プログラムとの連絡	7-41
7.12.1	制御の移行方法	7-42
7.12.2	引数の引き渡し方法	7-43
7.12.3	引数の受け渡し詳細	7-47
7.12.3.1	COBOL から C/C++ を呼び出す場合	7-47
7.12.3.2	C/C++ から COBOL を呼び出す場合	7-50
7.12.4	戻り値の受け取り方法	7-55
7.12.5	外部データによるデータ授受	7-57

7.12.6 その他.....	7-57
7.13 C/C++プログラムとのリンク時の注意事項.....	7-57
7.13.1 全般的な注意.....	7-58
7.13.2 COBOL がメインプログラムの場合	7-59
7.13.3 COBOL がサブプログラムの場合	7-61
7.14 CALL 一意名呼び出し時の詳細エラー	7-68
第8章 利用者定義関数.....	8-1
8.1 概説	8-1
8.2 利用者定義関数の定義.....	8-1
8.3 利用者定義関数の呼び出し.....	8-2
8.4 データの受け渡し.....	8-3
8.5 再帰呼び出し	8-11
8.6 利用者定義関数の利用方法.....	8-12
第9章 オブジェクト指向機能.....	9-1
9.1 オブジェクト指向の概念.....	9-1
9.1.1 クラスとオブジェクト.....	9-1
9.1.2 カプセル化.....	9-2
9.1.3 継承.....	9-3
9.1.4 ポリモルフィズム.....	9-3
9.2 COBOL におけるオブジェクト指向	9-5
9.2.1 クラス定義.....	9-5
9.2.2 ファクトリ定義.....	9-7
9.2.3 インスタンス定義.....	9-7
9.2.4 メソッド定義.....	9-8
9.2.5 オブジェクト参照.....	9-12
9.2.6 メソッド呼び出し.....	9-14
9.2.7 オブジェクトプロパティ.....	9-15
9.2.8 標準クラス.....	9-21
9.2.9 オブジェクト指向での適合	9-26
9.2.10 オブジェクト指向でのファイル.....	9-29
9.3 プログラム例	9-30
9.4 C++からの利用方法.....	9-35
9.5 オブジェクト指向機能の利用方法.....	9-37
9.6 オブジェクト指向機能利用時のメモリ容量.....	9-40

9.7	注意事項	9-42

第 10 章 整列併合機能 10-1

10.1	概 説	10-1
10.2	整列併合用ファイルの設定	10-2
10.3	整列併合機能	10-4
10.3.1	入力ファイルと出力ファイル	10-5
10.3.2	レコードの引き渡しとレコードの受け取り	10-6
10.3.3	キー項目の宣言と評価	10-8
10.4	整列併合処理	10-9
10.4.1	SORT 文	10-11
10.4.2	RELEASE 文	10-13
10.4.3	RETURN 文	10-14
10.4.4	MERGE 文	10-14
10.4.5	入力手続き	10-16
10.4.6	出力手続き	10-16
10.5	可変長レコードの整列併合	10-23
10.6	整列併合プログラムの実行	10-26
10.6.1	整列併合用ファイル	10-26
10.6.2	注意事項	10-26

第 11 章 翻訳指示機能 11-1

11.1	概説	11-1
11.2	COPY 文	11-1
11.2.1	登録集原文名の指定	11-2
11.2.2	REPLACING 句	11-4
11.3	登録集原文	11-6
11.4	デバッグ行と COPY 文	11-6
11.5	選択行と COPY 文	11-6

第 12 章 例外処理機能 12-1

12.1	概説	12-1
12.2	例外手続き	12-2
12.3	例外の説明	12-4
12.4	例外発生時の処理の流れ	12-7
12.5	例外メッセージ制御	12-8

第 13 章 システムサブルーチン	13-1
13.1 概説	13-1
13.2 システムサブルーチンの詳細	13-1
13.2.1 コマンド行の引数の取り込み	13-1
13.2.2 環境変数の取り込み	13-2
13.2.3 環境変数への設定	13-4
13.2.4 エラーコード取得	13-5
13.2.5 シェルコマンドの実行	13-9
13.2.6 4 桁での西暦取得	13-10
13.2.7 コード変換	13-11
13.2.8 基本多言語面以外の文字検査	13-21
13.2.9 プリンタファイル	13-22
13.2.9.1 プリンタファイルのシンボリック制御	13-22
13.2.9.2 プリンタファイルの装置アドレス変更	13-23
13.2.9.3 プリンタファイルの強制クローズ	13-24
13.2.9.4 SPOOL 付加情報の設定	13-25
13.2.9.5 プリンタ印字モードの指定	13-27
13.2.10 別プログラムの起動	13-28
13.2.10.1 別プログラムの起動 (CBLRUN)	13-28
13.2.10.2 別プログラムの起動 (CBLSPAWN)	13-29
13.2.10.3 別プログラムの起動 (CBLCHAINR)	13-30
13.2.10.4 別プログラムの起動 (CBLCHAINC)	13-31
13.2.11 ファイル入出力用システムサブルーチン	13-33
13.2.11.1 A_CBL_START	13-33
13.2.11.2 A_CBL_NOTE	13-34
13.2.11.3 A_CBL_CLEAN	13-35
13.2.11.4 A_CBL_STARTB	13-37
13.2.12 内部ファイル ID 置換表の解放	13-39
13.2.13 ファイル入出力状態の永続誤り解除	13-41
 第 14 章 プログラミング技法	 14-1
14.1 概説	14-1
14.2 実行用プログラムの効率化	14-1
14.2.1 ファイルの入出力処理を早くするための技法	14-2
14.2.2 プログラムが占有する主記憶を少なくするための技法	14-2
14.2.3 データの処理を早くするための技法	14-3

第 15 章 翻訳リスト	15-1
15.1 概説	15-1
15.2 パラメータリスト	15-3
15.3 原始プログラムリスト	15-4
15.4 クロスリファレンスリスト	15-11
15.5 登録集参照リスト	15-14
15.6 診断メッセージリスト	15-15
15.7 データマップリスト	15-16
15.8 用語の展開リスト	15-19
15.9 用語名リスト	15-21
15.10 用語値名参照リスト	15-23
 第 16 章 データ例外処理機能	 16-1
16.1 概説	16-1
16.2 データ例外手続き	16-1
16.3 不正 10 進データ例外と COBOL 文	16-2
16.4 メッセージの出力形式	16-4
16.5 COBOL データ例外処理	16-5
16.6 データ例外発生原因の調べ方	16-5
 第 17 章 利用者制御変数処理機能	 17-1
17.1 概説	17-1
17.2 環境変数	17-1
17.3 コマンドラインツール	17-1
17.3.1 cobinituc	17-1
17.3.2 cobsetuc	17-3
17.3.3 cobreaduc	17-4
17.3.4 cobtermuc	17-5
17.4 使用例	17-6
 第 18 章 日本語情報処理機能	 18-1
18.1 概説	18-1
18.2 ファイル入出力機能	18-1
18.2.1 APPLY SHIFT-CODE 句	18-1
18.2.1.1 改行制御	18-2
18.2.1.2 レコード形式	18-2

18.2.1.3 レコード出力例	18-3
第 19 章 64 ビットアプリケーションの生成方法.....	19-1
19.1 Windows 版の場合	19-1
19.1.1 アプリケーションの生成と実行の環境	19-1
19.1.2 翻訳環境の設定	19-1
19.1.3 コマンドの指定方法	19-3
19.1.4 64 ビットでの言語仕様	19-3
19.2 Linux 版の場合	19-6
19.2.1 アプリケーションの生成と実行の環境	19-6
19.2.2 翻訳環境の設定	19-6
19.2.3 コマンドの指定方法	19-7
19.2.4 64 ビットでの言語仕様	19-7
19.2.5 注意事項	19-10
第 20 章 マルチスレッドプログラミング.....	20-1
20.1 COBOL の提供するマルチスレッド機能	20-1
20.2 アプリケーションの生成と実行の環境	20-2
20.3 用語の説明	20-3
20.4 マルチスレッドプログラミングの留意事項	20-3
20.4.1 COBOL データの有効範囲と生存期間	20-3
20.4.2 マルチスレッドオプション	20-6
20.4.3 実行時エラー発生時の動作	20-7
20.4.4 環境変数	20-7
20.4.5 システムサブルーチン	20-7
20.4.6 ファイル	20-8
20.4.6.1 外部ファイルの操作	20-8
20.4.6.2 内部ファイルの操作	20-8
20.4.7 BASED 指定ファイルの使用	20-9
20.4.8 ポインタ	20-9
20.5 マルチスレッド対応時の他言語連携の留意事項	20-10
20.5.1 プログラム構成	20-10
20.5.2 COBOL プログラムからのスレッド作成	20-11
20.5.3 サブスレッドから COBOL プログラムを実行する場合	20-12
第 21 章 関連製品連携	21-1

21.1	概説	21-1
21.2	アプリケーション生成時の注意	21-3
21.3	動作環境(実行環境)	21-6
21.4	アプリケーション実行時の注意事項	21-12
21.4.1	プログラム名段落の INITIAL 句, COMMON 句	21-12
21.4.2	データ部の諸元	21-12
21.4.3	手続き部の諸元	21-12
第 22 章 画面制御機能		22-1
22.1	概説	22-1
22.1.1	画面の構成	22-1
22.1.2	カスタマイズ機能	22-3
22.1.3	副画面機能	22-3
22.1.3.1	親画面と子画面	22-4
22.1.3.2	主画面	22-4
22.1.3.3	副画面	22-4
22.1.3.4	親画面と子画面の関係	22-5
22.2	画面記述項	22-6
22.3	画面データ記述項	22-11
22.3.1	一般形式	22-11
22.3.2	書き方 1, 書き方 2 と書き方 3, 4 または 5 の組み合わせ	22-15
22.3.3	画面データ記述項の各句の分類	22-15
22.4	行指定	22-17
22.5	桁指定	22-18
22.6	クリア指定	22-19
22.7	入力チェック指定	22-21
22.8	動作指定	22-26
22.9	けい線指定	22-27
22.9.1	指定された項目に対するけい線指定	22-27
22.9.2	指定された項目とは独立したけい線指定	22-27
22.10	色指定	22-30
22.10.1	COLOR 句	22-30
22.10.2	その他の色指定	22-30
22.11	PICTURE 指定	22-31
22.11.1	数字項目に対する入力／出力時の画面上の桁数	22-32
22.12	データ変換指定	22-34

22.13 キー入力時の動作.....	22-37
22.14 データの修正方法.....	22-38
22.15 入力終了条件／状態.....	22-41
22.16 画面制御機能のプログラム例.....	22-42
 第 23 章 データベース機能.....	23-1
23.1 概説.....	23-1
23.1.1 COBOL で使用するまでの手順.....	23-2
23.2 RDB ファイルの装置指定.....	23-2
23.3 DB アクセス文.....	23-4
23.3.1 入出力文と拡張入出力文.....	23-4
23.4 RDB とファイル編成.....	23-7
23.5 入出力文.....	23-7
23.5.1 READ 文.....	23-8
23.5.2 WRITE 文.....	23-10
23.5.3 REWRITE 文.....	23-11
23.5.4 START 文.....	23-12
23.5.5 DELETE 文.....	23-13
23.5.6 SELECT 文と SCRATCH 文.....	23-14
23.5.7 COMMIT 文と ROLLBACK 文.....	23-16
23.6 RDB-STATUS.....	23-16
23.7 入出力状態.....	23-17
23.8 入出力状態の検査.....	23-25
 第 24 章 実行トレース機能.....	24-1
24.1 概説.....	24-1
24.2 トレース情報ファイルの出力方法.....	24-1
24.3 トレース情報ファイルの出力内容.....	24-4
24.4 出力範囲指定の方法.....	24-10
24.5 データ値出力の方法.....	24-13
24.5.1 データ値出力の概要.....	24-13
24.5.2 データ値出力の方法.....	24-15
24.5.3 データ値出力の内容.....	24-21
24.5.4 データ値出力のエラーメッセージ.....	24-24
24.6 注意事項.....	24-25

第 25 章 用語機能	25-1
25.1 概説	25-1
25.2 用語機能	25-1
25.2.1 用語データ項目の定義と参照	25-1
25.2.2 用語値名	25-2
25.2.3 用語機能の全般規定	25-2
25.2.4 データ記述項の骨組み	25-3
25.2.5 USAGE 句	25-5
25.2.6 手続き部	25-7
25.3 用語機能の例	25-8
 第 26 章 ファイル入出力レコード情報生成機能	 26-1
26.1 概説	26-1
26.2 ファイル入出力レコード情報の出力方法	26-1
26.3 ファイル入出力レコード情報の出力内容	26-2
26.4 注意事項	26-7
 第 27 章 データレコード情報出力機能	 27-1
27.1 概説	27-1
27.2 データレコード情報の出力方法	27-1
27.3 データレコード情報の出力内容	27-2
 付録	 A-1
A. 制限事項/注意事項	A-1
A.1 翻訳時の諸元	A-1
A.1.1 各部に共通なもの	A-1
A.1.2 データ部に関するもの	A-1
A.1.3 手続き部に関するもの	A-3
A.1.4 オブジェクト指向機能に関するもの	A-4
A.1.5 リポジトリ機能に関するもの	A-4
A.2 注意事項	A-4
A.2.1 外部公開名に使用できない文字	A-5
A.2.2 外部公開名に使用可能な文字数	A-6
A.2.3 スタックオーバーフロー発生時の対応	A-8
A.2.4 実行ファイルの大きさの制限	A-16
A.2.5 C++コンパイラからのエラー報告への対応	A-18

A.2.6 COBOL が登録するシグナル	A-19
A.2.7 オプションに使用できない文字	A-20
B 診断メッセージ	B-1
B.1 原始プログラムの診断メッセージ	B-1
B.2 翻訳時のメッセージ	B-181
C 実行時メッセージ一覧	C-1
C.1 実行時エラー	C-1
D 環境変数	D-1
E 例外コード一覧	E-1
F 翻訳オプションや環境変数による仕様差異	F-1
F.1 翻訳オプションによる仕様差異	F-1
F.2 環境変数による仕様差異	F-10
G Unicode 対応について	G-1
G.1 翻訳時の Unicode 対応	G-1
G.2 Unicode のコード範囲と本 COBOL でのサポート範囲	G-1
G.3 文字コードに関連した制限	G-3
H JIS2004 対応について	H-1
H.1 注意事項	H-1
H.2 制限事項	H-2

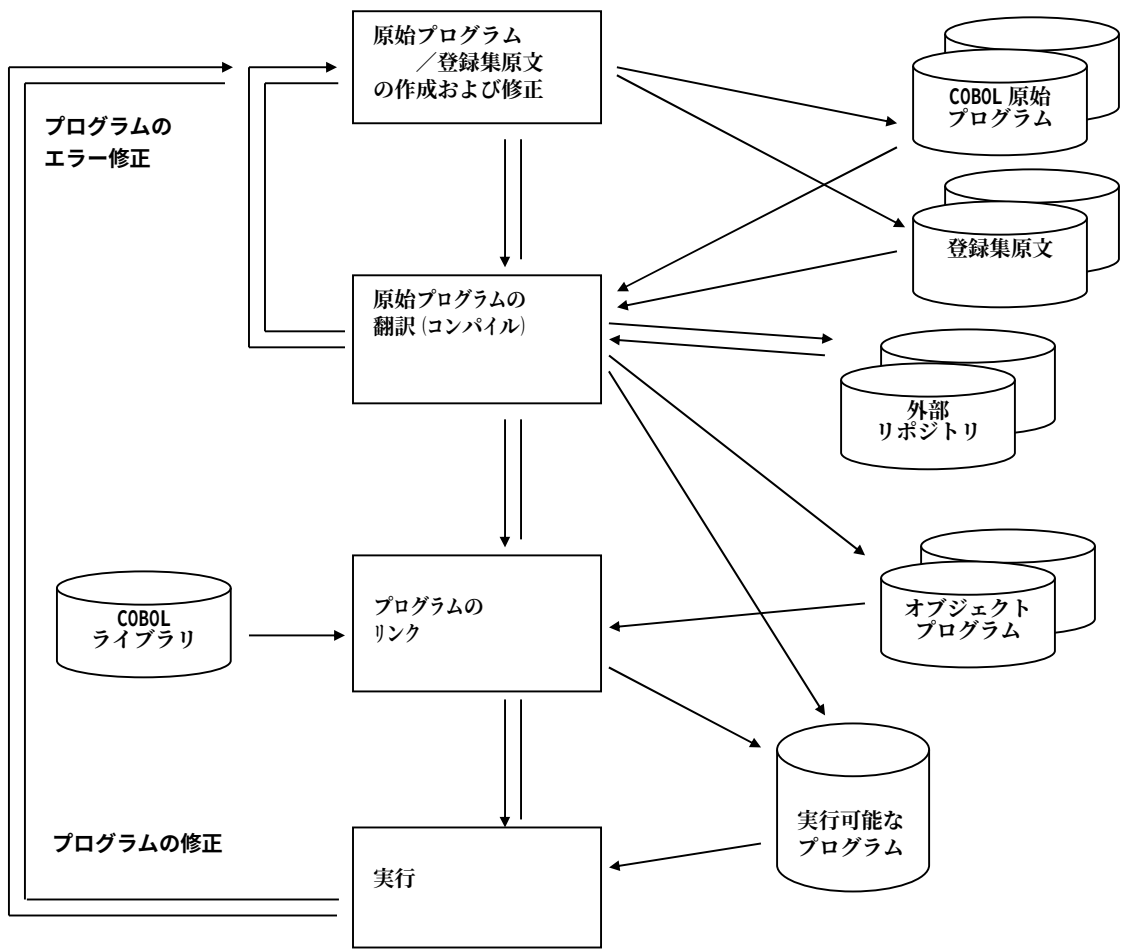
第1章 原始プログラムの翻訳, リンク

本章では、原始プログラムを翻訳するときの原始プログラムの作成、コンパイラの操作法およびリンカの操作法などについて、注意事項を中心に説明する。

1.1 原始プログラムの作成から実行の概要

原始プログラムを作成して実行するまでに必要な処理を、図 1-1 に示す。

図 1-1 原始プログラムの作成から実行までの概要



1.2 原始プログラムの作成および修正

原始プログラムの作成および修正は、エディタ等を使用する。このときに作成するファイル名は、次の規則に従って命名する必要がある。次の規則に従わないファイル名は、COBOL 原始プログラムとみなされない。COBOL 原始プログラムをエディタで作成する場合に、COBOL の正書法に従う必要がある。

また、ファイルの文字コードおよびファイル名はシフト JIS として作成する必要がある。

COBOL 言語には正書法があり、正書法に対応した形式(書式)がある。

本システムでは、形式(書式)に対応したファイルを定義する必要がある。

その際に、形式(書式)は、ファイルの拡張子によって指定する必要がある。

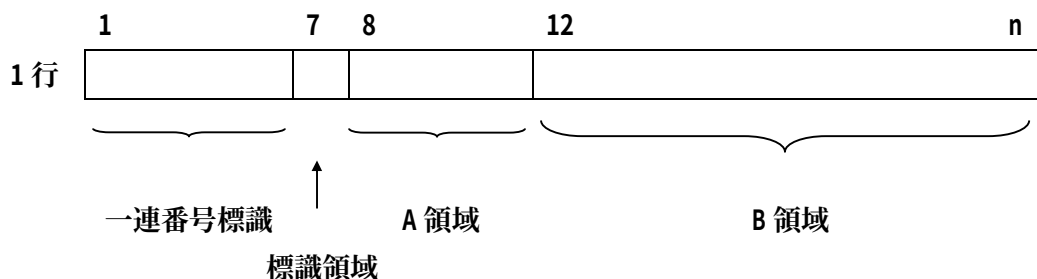
形式	名前	概要	拡張子
形式1	固定形式 (標準:識別領域なし)	「固定形式正書法」に対応した形式である。 一連番号, 標識領域, A 領域, B 領域からなる, <u>最大 255 桁</u> の形式	.cob
形式2	固定形式 (識別領域あり)	「固定形式正書法」に準拠した形式である。 一連番号, 標識領域, A 領域, B 領域, 識別領域からなる, <u>最大 80 桁</u> の形式	.cbl
形式3	自由形式	「自由形式正書法」に対応した形式である。 ソース行のみからなる, <u>最大 255 桁</u> の形式	.cbf
【Standard】 形式4	固定形式 (識別領域あり)	「固定形式正書法」に準拠した形式である。 一連番号, 標識領域, A 領域, B 領域, 識別領域からなる, <u>最大 255 桁</u> の形式	.cbx

固定形式正書法および自由形式正書法についての詳細は、それぞれ、『COBOL 言語説明書』「2.3. 固定形式正書法」、『COBOL 言語説明書』「2.4. 自由形式正書法」を参照。

以降は、各形式についての概要である。

[形式1] 固定形式 (標準: 識別領域なし)

ファイルの拡張子は, ".cob" である。(最大長 $n \leq 255$ 桁(バイト数))

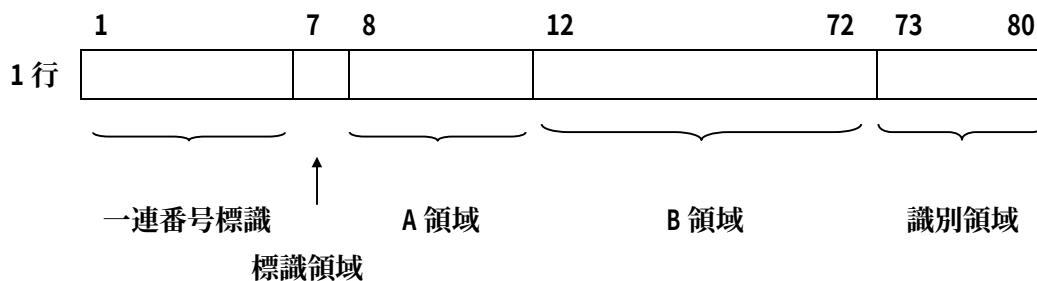


識別領域なしの固定形式である。

エディタで作成時の最右端文字を B 領域の終端とする可変行である。B 領域の終了は最右端有効文字(空白は最終文字とみなさない)までなので, 継続行の指定をするときは, 特に注意が必要である。

[形式2] 固定形式 (識別領域あり)

ファイルの拡張子は, ".cbl" である。(最大長 80 桁(バイト数))

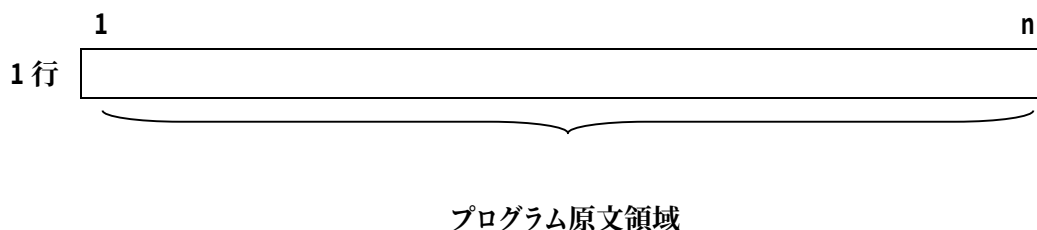


識別領域ありの固定形式である。

72 桁と 73 桁との間を B 領域の終端とする固定行とする。72 桁に満たない場合は, 以降に空白を補い, 73 桁目以降は識別領域とみなす。

[形式3]自由形式

ファイルの拡張子を ".cbf" とする。(最大長 $n \leq 255$ 文字(文字位置の数))

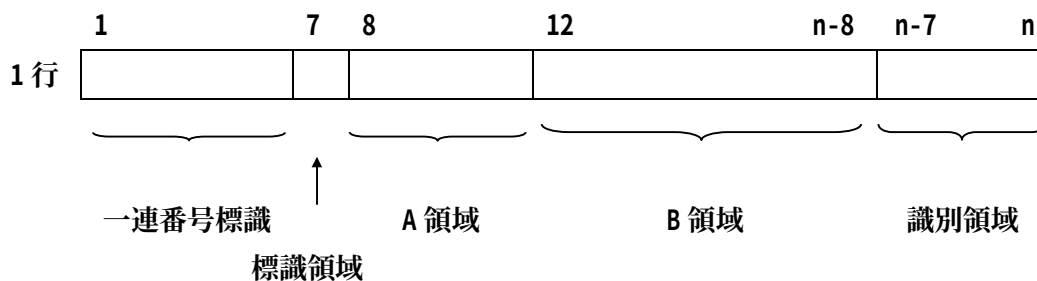


自由形式である。

エディタで作成時の行全体をプログラム原文領域とする可変行である。一連番号領域, 標識領域, A 領域, B 領域, 識別領域がなく, プログラム原文領域のみである。プログラムを行内の任意の位置に記述できる正書法である。浮動標識もプログラム原文領域に記述する。

[Standard] [形式4] 固定形式 (識別領域あり)

ファイルの拡張子は, ".cbx" である。(最大長 $n \leq 255$ 桁(バイト数))



識別領域ありの固定形式である。

エディタで作成時の最右端文字から識別領域(8 文字)分左の文字位置の左側までを B 領域の終端とする可変行である。

1.3 登録集原文の作成および修正

原始プログラムの中の COPY 文によって複写される登録集原文の作成および修正は, 原始プログラムと同様にエディタを使用する。このときに作成するファイルは, 原始プログラムと同じ, 形式 1, 形式 2 または形式 3 で作成する必要がある。

また, ファイルの文字コードおよびファイル名はシフト JIS として作成する必要がある。

- ・原始プログラムの形式に関する規則は, 登録集原文にも適用される。
- ・1つの翻訳単位中に, 形式の異なる登録集原文の混在も可能である。

1.4 原始プログラムの翻訳(コンパイル)およびプログラムのリンク

1.4.1 翻訳用コマンドプロンプト

[Enterprise for Windows] [Standard] 翻訳用コマンドプロンプトは, Enterprise Edition for Windows および Standard Edition で使用できる。

原始プログラムの翻訳は, Visual Studio のコマンドプロンプトを使用する。

Windows Server 2019:

- ・Visual Studio 2019 を使用する場合

[スタート]→[Visual Studio 2019]→[x64 Native Tools Command Prompt for VS 2019]

- ・Visual Studio 2022 を使用する場合

[スタート]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

Windows Server 2022:

- ・Visual Studio 2022 を使用する場合

[スタート]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

Windows 10:

- ・Visual Studio 2015 を使用する場合

[スタート]→[Visual Studio 2015]→[Visual Studio Tools]→[VS2015 x64 Native Tools
コマンド プロンプト]

・Visual Studio 2017 を使用する場合

[スタート]→[Visual Studio 2017]→[VS 2017 用 x64 Native Tools コマンド プロンプト]

・Visual Studio 2019 を使用する場合

[スタート]→[Visual Studio 2019]→[x64 Native Tools Command Prompt for VS 2019]

・Visual Studio 2022 を使用する場合

[スタート]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

Windows 11:

・Visual Studio 2022 を使用する場合

[スタート]→[すべてのアプリ]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

※コマンドプロンプト起動時のカレントディレクトリは, ショートカットプロパティ (上記のスタートメニューから該当項目を右クリックし, 「プロパティ」を選択) の「作業フォルダ」を設定することによって変更できる。

1.4.2 翻訳用コマンドとその使用法

原始プログラムを翻訳(コンパイル)するための, 起動コマンドとその使用法を以降に示す。

名 前	cob:COBOL コンパイラ
形 式	cob [options] files
説 明	

cob は COBOL コンパイラである。

cob は次のようなファイル名の形式に応じて適当なプログラムを起動し, 処理を行う。

COBOL の原始プログラム (注 1) は, COBOL コンパイラにて翻訳する。原始プログラム中にあるハードタブ(タブコードによる水平タブ)の桁間隔は, 8 桁として扱う。C または C++プログラム (注 2) は, COBOL コンパイラから C/C++コンパイラ (注 3) を起動し翻訳する。そして, オブジェクトプログラム (注 4) を現ディレクトリ下に生成する (注 5)。

注意

(注 1) [Enterprise] 拡張子が".cob", ".cbl"または".cbf"のファイル
[Standard] 拡張子が".cob", ".cbl", ".cbf"または".cbx"のファイル

(注 2) C または C++プログラムの拡張子は後述の関連ファイル参照

(注 3) [Enterprise for Windows] [Standard] 使用する c コンパイラは Visual C++ コンパイラである。

[Enterprise for Linux] 使用する c コンパイラは gcc(1) コンパイラである。

(注 4) [Enterprise for Windows] [Standard] COBOL の原始プログラム, C または C++ プログラムの拡張子を ".obj" に変更したファイル

[Enterprise for Linux] COBOL の原始プログラム, C または C++ プログラムの拡張子を ".o" に変更したファイル

(注 5) 1 つの原始プログラムの中に複数のプログラム定義やクラス定義がある場合 (同一翻訳群) のオブジェクトプログラム生成規則については, 「1.4.3 同一翻訳群の原始プログラムの翻訳時の注意点」を参照。

[Enterprise] すべての翻訳が正常に終了した場合, リンク用のコマンド (注 6) を起動する。リンク用のコマンド (注 6) には, コンパイラが生成したオブジェクトプログラムおよび, ".cob", ".cbl" または ".cbf" で終わらないすべてのパス名, リンク用のオプションを渡す。リンカは -o オプションを指定しなければ, 既定の名前 (注 7) の実行可能プログラムファイルまたは共有ライブラリファイルを現ディレクトリ下に生成する。また, 1 つの原始プログラムを翻訳し, リンクした場合は, オブジェクトプログラムを削除する。

[Standard] すべての翻訳が正常に終了した場合, リンク用のコマンド (注 6) を起動する。リンク用のコマンド (注 6) には, コンパイラが生成したオブジェクトプログラムおよび, ".cob", ".cbl", ".cbf" または ".cbx" で終わらないすべてのパス名, リンク用のオプションを渡す。リンカは -o オプションを指定しなければ, 既定の名前 (注 7) の実行可能プログラムファイルまたは共有ライブラリファイルを現ディレクトリ下に生成する。また, 1 つの原始プログラムを翻訳し, リンクした場合は, オブジェクトプログラムを削除する。

注意

(注 6) [Enterprise for Windows] [Standard] link
[Enterprise for Linux] gcc(1)

(注 7) [Enterprise for Windows] [Standard] コマンドラインで指定した最初のファイルのベース名に, 拡張子 .exe を付けた名前
[Enterprise for Linux] a.out

以降に cob のオプションを示す。

-c 翻訳だけを行い, リンクは行わない。生成したオブジェクトファイルを削除しない。
[Standard] -Yi オプションとの同時指定はできない。

-outfile [Enterprise for Windows] [Standard] outfile が円記号"¥"で終了している場合は, outfile をディレクトリ名とみなし, 翻訳用コマンドで指定した最初のファイ

ルのベース名に拡張子 **.exe** を付加した名前です **outfile** ディレクトリに実行可能プログラムファイルを生成する。 **outfile** が円記号 **"¥"** で終了していない場合は、 **outfile** をファイル名とみなし、 **-M** または **-Mw** オプションが指定されたとき、 **outfile** に拡張子があれば、拡張子を **.exe** に置き換えて、実行可能プログラムファイルを生成する。また、 **outfile** に拡張子が無ければ、拡張子 **.exe** を付け加えて、実行可能プログラムファイルを生成する。

-G オプションが指定されたとき、 **outfile** に拡張子があれば、拡張子を **dll/lib/exp** に置き換えて、 **DLL** / インポートライブラリ / エクスポートファイルを生成する。また、 **outfile** に拡張子が無ければ、拡張子 **.dll** を付け加えて、 **outfile.dll/outfile.lib/outfile.exp** という名前で **DLL** / インポートライブラリ / エクスポートファイルを生成する。本オプションで指定するファイル名およびパス名はシフト **JIS** でなければならない。

[Enterprise for Linux] **outfile** という名前で実行可能プログラムファイルを生成する。 **-G** オプションが指定された場合は、 **outfile** という名前で共有ライブラリを生成する。本オプションで指定する **outfile** はファイル名のみ、またはフルパス付のファイル名で、ファイル名およびパス名はシフト **JIS** でなければならない。 **outfile** の前に空白を指定してもよい。

-Cc...

コンパイラの動作を指定するオプションである。

c には **[{a b c d e f g h i j m o q r s t u v w x y z A B D F G H I L N S P U X Y 5 | C}]** のうち 1 つ以上を指定できる。

ただし、 **-CC** と他の **-Cc** オプションは連結して指定できないため、分けて指定しなければならない。

(正しい使用例) **-Cade -CC0**

(誤った使用例) **-CadeC0**

a

すべてのデータ項目を実行効率が高くなるように割り付ける。すなわち、利用者プログラム内のすべてのデータ記述項に **SYNCHRONIZED** 句が指定されたものとみなされる。

本オプションは、プログラムや関数やメソッドの引数 / 返却項目にも影響を与えるため、呼び出す側と呼び出される側の **COBOL** 原始プログラムについて、一方にオプションを指定する場合は、他方にも指定しなければならない。

d

実行時に 10 進データ項目参照時の不正データチェックを行わない。

e

翻訳リストの中に印字される行番号を外部行番号で出力する。

自由形式の場合は, -Ce オプションを無効とし内部行番号を使用する。

- f** **[Standard]** 順編成の大容量記憶ファイルを COBOL 独自形式にする。
オプション指定なしのときは, IFAS FILE 形式である。
SE 動作モードでは, 順編成の大容量記憶ファイルは常に COBOL 独自形式であり, -Cf オプションは使用できない。
- g** **[Standard]** 相対編成の大容量記憶ファイルを COBOL 独自形式にする。
オプション指定なしのときは, IFAS FILE 形式である。
SE 動作モードでは, 相対編成の大容量記憶ファイルは常に COBOL 独自形式であり, -Cg オプションは使用できない。
- h** **[Standard]** 索引編成の大容量記憶ファイルを ISAM 形式にする。
オプション指定なしのときは, IFAS FILE 形式である。
SE 動作モードでは, 索引編成の大容量記憶ファイルは常に ISAM 形式であり, -Ch オプションは使用できない。
- i** INITIAL 句および CANCEL 文は未使用のため, 生成オブジェクトを効率化する。
本オプションを指定する場合は, 実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。
- j** プログラムの固有文字集合(内部コード)を JIPS(J)とする。
対象は日本語項目と日本語編集項目である。
本オプションを指定する場合は, 実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。
[Enterprise] -CU オプションとの同時指定はできない。
[Standard] -Mw オプションとの同時指定はできない。
本オプションはクラス定義と関数定義には使用できない。プログラム定義だけで構成する実行単位にのみ使用可能である。
[Standard] 本オプションは, 画面節を記述した COBOL ソースには指定できない。
- m** 翻訳時に, 実行効率を高めるためにプログラムを書き直すべき記述を指摘するメッセージを出力する。通知の対象, すなわち実行効率を低下させる記述は次のとおりである。

(1) 定義されたデータ項目の割り付け位置が次のいずれかの場合

- (a) 2 バイト 2 進データ項目が 2 バイト境界に割り付けられていない
 - (b) 4 バイト 2 進データ項目が 4 バイト境界に割り付けられていない
 - (c) 8 バイト 2 進データ項目が 8 バイト境界に割り付けられていない
 - (d) 指標データ項目が 4 バイト境界に割り付けられていない
- (2) 表データの定義において, OCCURS 句を含むデータ記述項に従属しているデータ記述項の要素長が 4 の倍数でない場合
- (3) 1つの演算文, または1つの算術中に, 10 進データ項目と 2 進データ項目が混在している場合
- (4) 1つの ADD 文, または1つの SUBTRACT 文, または加減算だけからなる 1 つの算術式中に, 小数部桁数の異なる数字項目を記述している場合
- (5) 次のいずれかにおいて, 10 進データ項目を使用している場合
(実行効率を高めるためには, 2 進データ項目を使用するとよい)
- (a) 添字または部分参照の最左端文字位置および長さ
 - (b) GO TO 文の DEPENDING ON 句の一意名
 - (c) PERFORM 文の TIMES 指定の一意名
 - (d) PERFORM 文の VARYING/AFTER 句の一意名
 - (e) PERFORM 文の FROM 句または BY 句の一意名
 - (f) SEARCH 文の VARYING 句の一意名
 - (g) INSPECT 文の TALLYING 句の一意名
 - (h) STRING 文の POINTER 句の一意名
 - (i) UNSTRING 文の COUNT 句の一意名
 - (j) UNSTRING 文の POINTER 句の一意名
 - (k) UNSTRING 文の TALLYING 句の一意名
 - (l) SET 文の UP/DOWN BY 句の一意名
 - (m) 相対ファイルの RELATIVE KEY 句のデータ名
- o 実行時に, 添字, 可変長集団項目および部分参照の範囲外参照のチェックを行わない。
- q COBOL コンパイラの終了コードを指示する。このオプションの有無により次のように変化する。

(1) -Cq オプションがある場合

- (a) エラーが検出されなかった場合 0
- (b) 警告(WARNING)エラーまたは注意(OBSERVE)が検出された場合 0
- (c) 致命的(FATAL)エラーが検出された場合 2
- (d) その他のエラーの場合 3 以上

(2) -Cq オプションがない場合

- (a) エラーが検出されなかった場合 0
- (b) 警告(WARNING)エラーまたは注意(OBSERVE)が検出された場合 1
- (c) 致命的(FATAL)エラーが検出された場合 2
- (d) その他のエラーの場合 3 以上

備考 コマンドプロンプト画面で `makefile` を使用して, `cob` コマンドを起動する際に使用するオプションである。

r 同桁同符号の外部 10 進数どうしの転記を英数字転記とする。

s 英小文字はその対応する英大文字とみなさず, ソースプログラム中に書かれたとおりオブジェクトファイルに出力する。ただし, コンパイラの内部処理で英小文字はその対応する英大文字とみなして処理を行う。本オプションの処理対象は, 以下である。

- (1) プログラム名段落のプログラム名
- (2) `EXTERNAL` 句指定のあるレベル番号 01 のデータ名
- (3) 内部ファイル識別名のファイル識別リテラル

プログラム名, `EXTERNAL` 句指定のあるレベル番号 01 のデータ名に英小文字を使う場合, システム提供の関数名等と重複しないように注意が必要である。

t `BINARY` 項目へのデータ格納時, データが割り付けられた記憶領域に対応した値まで格納され, `PICTURE` 句の桁数での桁落しは行われない。
指定がない場合は `PICTURE` 句の桁数で 10 進数の桁落しを行う。
ただし, 初期値 `VALUE` 句の設定値については, 10 進数の桁落しを行う。

u プログラム名および外部データ名に含まれる語中のハイフンを下線と等価とみな

す。本オプションの処理対象は、以下である。

- (1) プログラム名段落のプログラム名
- (2) EXTERNAL 句指定のあるレベル番号 01 のデータ名
- (3) CALL 定数指定のプログラム名
- (4) CALL 一意名指定のプログラム名
- (5) CANCEL 定数指定のプログラム名

注意 同一プログラム中にハイフンと下線以外の文字が同じデータ名が複数定義されているプログラムの翻訳が不可となる。

例)

01 A-B PIC X(3) IS EXTERNAL.

01 A_B PIC X(10) IS EXTERNAL.

-Cu オプション未指定時は「A-B」と「A_B」は別々のデータ名であるが、-Cu オプション指定時は同一データ名となり、翻訳不可となる。

v	定数指定の CALL 文において、BY VALUE 指定の引数を BY REFERENCE 指定とみなす。
w	英数字転記の作用対象が重なっている場合の動作を変更する。 本オプション指定ありの場合、転記の送り出し側項目と受け取り側項目の作用対象が重なっている時、先頭から1バイト単位で転記を行う。 本オプション指定なしの場合、処理速度を優先した転記を行う。
y	プリンタファイルの APPLY SHIFT-CODE 句をコメントとする。
z	【Enterprise for Linux】位置独立コード(Position Independent Code)を作成する。 -M との同時指定時、本オプション指定は無効となる (メインプログラムだけでなく、翻訳対象のすべてのコンパイルに対して位置独立コードは生成しない)。
A	PPR-CONTROL-2 指定の SYSOUT ファイルの印刷データに、プリンタ制御コードを埋め込む。
B	外部 10 進数の数字と符号の重ね合わせの文字がブランク文字の場合、0 として扱う。指定がなく、かつ、-Cd オプションの指定がない場合は不正 10 進となる。

本オプションを指定する場合は、実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。

- D** **ACCEPT** 文と **CURRENT-DATE** 関数で取得する日付をシステム日付とする。
本オプションを指定すると環境変数 **COB_JOBDATE** を参照しない。
- F** **[Standard]** **[Enterprise for Windows]** プログラムの固有文字集合(内部コード)を JIS2004 対応の JIPS(J)とする。
対象は日本語項目と日本語編集項目である。
本オプションを指定する場合は、実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。また、以下の関連製品がインストールされている必要がある。
 ・FontAvenue UniAssist コード変換
[Standard] **-Cj**, **-Pg**, **-P1**, **-P2** オプションとの同時指定はできない。
[Enterprise for Windows] **-Cj**, **-CU**, **-MT** オプションとの同時指定はできない。
- G** 外部リポジトリを更新する。
本オプション指定ありの場合、生成先(原始プログラムファイルと同一ディレクトリ)に外部リポジトリが存在しない場合は生成する。生成先に外部リポジトリが存在する場合は更新する。
本オプション指定なしの場合、外部リポジトリの生成および更新は行わない。
-CH オプションとの同時指定はできない。
- H** 外部リポジトリと翻訳対象の要素(プログラム定義、関数定義、クラス定義など)の定義情報との間に矛盾があるかどうかをチェックする。
本オプション指定ありの場合、矛盾があるかを生成のタイミングでチェックし、矛盾があれば、翻訳エラーとする。本オプションの有無に関わらず、外部リポジトリは更新されない。
-CG オプションとの同時指定はできない。
- I** カバレッジ情報と部分情報を出力する。
カバレッジ情報の出力とは、トレース情報に利用者定義の手続き名(節名、段落名)と条件文の分岐情報を出力することを示す。

部分情報の出力とは、ソース単位名、行番号、手続き名で指定した範囲のトレース情報を部分的に出力することを示す。

- L** トレース情報にデータ項目の値を出力する。
 また、データ項目の値を出力するために必要なファイルとして、オブジェクトプログラムを作成する場合はデバッグ情報ファイルを、実行可能プログラムを作成する場合はデバッグ情報ファイル、リンクマップ、データ情報ファイルを出力する。
 本オプションは、オブジェクトプログラムを作成する場合も、実行可能プログラムを作成する場合も、指定する必要がある。
 本オプションを指定して作成していない実行可能プログラムの場合、該当実行可能プログラム内のいずれのデータ項目の値も出力することはできない。
 本オプションを指定して作成していないオブジェクトプログラムが実行可能プログラムに含まれる場合、該当オブジェクトプログラム内のデータ項目の値は出力することはできない。
- N** 偶数桁で PICTURE 文字列に符号 S がない COMP-3 項目を符号領域なし COMP-3 項目として動作させる。
 本オプションを指定する場合は、実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。
- S** 数字と符号の重ね合わせが行われた外部 10 進数の符号桁の値を、Pro*COBOL の表現形式に変更する。オプション指定がなければ、従来どおり本コンパイラ固有の表現形式となる (既定値)。
 本オプションを指定する場合は、実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。
- P** [Enterprise] PPR-CONTROL-3 指定の SYSOUT ファイルの印刷データに、KLP 用のプリンタ制御コードを埋め込む。SSF 形式の SYSOUT ファイルに APPLY SHIFT-CODE 句の WITH PPR-CONTROL-3 指定ありのとき、レコード名または WRITE 文の FROM 一意名に、日本語データ項目があれば漢字文字サイズ指定のプリンタ制御コードを出力し、COLUMN NUMBER 句があれば水平アドレスタブのプリンタ制御コードを出力する。
 プリンタファイルや PRN 形式または TXT 形式の SYSOUT ファイル、APPLY SHIFT-CODE 句の WITH PPR-CONTROL-3 指定なしの SYSOUT ファイルに対して、このオプションは意味を持たない。

- U** **[Enterprise]** プログラムの固有文字集合 (内部コード) を **Unicode** として扱う。このとき、文字列は UTF-8、日本語文字列は UCS-2 として扱われる。
本オプションを指定する場合は、実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。
-Cj オプションとの同時指定はできない。
本オプションはクラス定義と関数定義には使用できない。プログラム定義だけで構成する実行単位にのみ使用可能である。
- X** **ON SIZE ERROR** 句なしのゼロ除算結果を 0 とする。
- Y** 実行可能プログラムに静的リンクした外部プログラムを一意名呼び出しできるようにする。
本オプションを指定して翻訳したサブプログラムを実行可能プログラムに静的リンクした場合、CALL 定数または CALL 一意名のどちらでも呼び出せるようになる。
[Enterprise for Linux] 本オプションはリンカに作用する。-c オプションを同時に指定している場合、無効となる。
- 5** **COMP** 項目の内部表現形式を **COMP-5** とする。
本オプションは、プログラムや関数やメソッドの引数／返却項目にも影響を与えるため、呼び出す側と呼び出される側の COBOL 原始プログラムについて、一方にオプションを指定する場合は、他方にも指定しなければならない。
[Enterprise] -Fc オプションとの同時指定はできない。
- Cm** **[Enterprise]** -CU オプションが指定されているとき、**CODE-SET** 句のない順／行順／プリンタ／SYSIN／SYSOUT ファイルに **CODE-SET** 句が指定されたものとして扱う。
m には {0 1 2} のうち 1 つを指定できる。
0 **CODE-SET** 句に **ALPHABET** 句で UTF-8 に対応づけられた符号系名が指定されているものとして処理する。
1 **CODE-SET** 句に **ALPHABET** 句で UCS-2 に対応づけられた符号系名が指定されているものとして処理する。
2 **CODE-SET** 句に **ALPHABET** 句で **STANDARD-2** に対応づけられた符号系名が指定されているものとして処理する。
- Hc...** コンパイラの出力リストを指定するオプションである。c には [pscdxwi] のうち 1 つ以上を指定できる (「第 15 章 翻訳リスト」を参照)。

- p** コンパイラ起動時のオプションパラメータの情報を出力する。
- s** 原始プログラムリストを出力する。**c** が指定されない場合, **copy** 展開は行わない。
- c** 登録集原文の展開を行ったソースリストを出力する。**s** と同時に指定しない場合, **s** が指定されたときみなして動作する。
- d** データマップリストを出力する。
- x** クロスリファレンスリストを出力する。
- w** 診断メッセージリストに **W(WARNING)**と **O(OBSERVE)**のエラーレベルのエラーを出力しない。
- i** 原始プログラムリスト出力指定ありのとき, 原始プログラムリストに診断メッセージを埋め込み出力する。
ただし, 原始プログラムリスト出力指定なしのとき, 原始プログラムリストに診断メッセージを埋め込み出力しない。
- f** **[Enterprise for Linux]**用語の展開リストを出力する。ただし, **-T** オプション指定なしのとき, リスト出力しない。
- m** **[Enterprise for Linux]**用語名リストを出力する。ただし, **-T** オプション指定なしのとき, リスト出力しない。
- e** **[Enterprise for Linux]**用語値名参照リストを出力する。ただし, **-T** オプション指定なしのとき, リスト出力しない。
- Dc** デバッグ行に対する処理を指定する。**c** には[yn]のうちの 1 つを指定する。指定がない場合は, 原始プログラム中の **WITH DEBUGGING MODE** 句に従う。
- y** 原始プログラム中の **WITH DEBUGGING MODE** 句の有無にかかわらず, デバッグ標識で示される行を有効にする。
-Dn オプションとの同時指定はできない。

- n** 原始プログラム中の WITH DEBUGGING MODE 句の有無にかかわらず, デバッグ標識で示される行を無効にする。
-Dy オプションとの同時指定はできない。
- ldirectory** 登録集原文を検索するディレクトリを指定する。-l オプションは複数指定でき, 指定された順に原文を検索する。登録集原文が指定されたディレクトリ中に見つからなかった場合は, 原始プログラムファイルのディレクトリを検索する。したがって, -l オプションが指定されなかった場合は, 原始プログラムファイルのディレクトリだけを検索する。また, 登録集原文検索ディレクトリ名はシフト JIS でなければならない。
[Enterprise for Linux] directory の前に空白を指定してもよい。
- M** バッチ型アプリケーションを作成する。翻訳するプログラムがメインプログラムであることを指定する。複数の COBOL 原始プログラムを指定した場合や複数翻訳群の COBOL プログラムを指定した場合は, 最初に出現するプログラム定義をメインプログラムとして翻訳する。COBOL 原始プログラムが単一の場合でも, このオプションは必須である。
-G オプションとの同時指定はできない。
[Standard] -Mw オプションとの同時指定はできない。
- MT** [Enterprise] 複数スレッドで動作可能なスレッドセーフコードを生成する。
本オプションを指定する場合は, 実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。
-Qa, -Qb オプションとの同時指定はできない。
- Mw** [Standard] 対話型アプリケーションを作成する。翻訳するプログラムがメインプログラムであることを指定する。複数の COBOL 原始プログラムを指定した場合や複数翻訳群の COBOL プログラムを指定した場合は, 最初に出現するプログラム定義をメインプログラムとして翻訳する。COBOL 原始プログラムが単一の場合でも, このオプションは必須である。
-Cj, -G, -M オプションとの同時指定はできない。
- Nc...** 原始プログラム中の選択行を翻訳行とするか注記行とするか指定する。
c には [0-9] のうち 1 つ以上または a を指定できる。ただし指定されていない場合は, すべての選択行を注記行とする。
本オプションは, 固定形式の場合のみ有効である。

- [0-9] 指定された数字が標識領域に書かれた選択行を翻訳行として, それ以外の選択行を注記行とする。
- a すべての選択行を翻訳行とする。
- 0c プログラムの最適化のレベルを指定するオプションである。c には[0 1 2]のうち 1 つを指定する。
指定がない場合は ” -01 ” が指定されたものとみなされる。
- 0 最適化を行わない。
-01 オプションまたは-02 オプションとの同時指定はできない。
- 1 局所的最適化(不要文の削除など)を行う。
-00 オプションまたは-02 オプションとの同時指定はできない。
- 2 広域的最適化(中間結果の有効利用など)を行う。
-00 オプションまたは-01 オプションとの同時指定はできない。
- Pc [Standard] COBOL GUIアプリケーションを作成するオプションである。ただし, SE動作モードでは, 使用できない。
cには[g 1 2]のうち1つ以上を指定する。
gと2のみ同時指定できる。
機能の詳細は, COBOL GUIのマニュアルを参照。
- g COBOL GUIアプリケーションを作成する。
-M, -Ca, -Cj, -CS, -CG, -CH, -w0, -w1オプションとの同時指定はできない。
- 1ssifile ssifileという名前でCOBOL GUIの画面節移行機能用に画面節移行ファイルを作成する。
ssifileが円記号"¥"で終了している場合は, ssifileをディレクトリ名とみなし, 翻訳用コマンドで指定した最初の原始プログラムファイルのベース名に拡張子.ssiを付加した名前でssifileディレクトリに画面節移行ファイルを生成する。
ssifileが円記号"¥"で終了していない場合は, ssifileをファイル名とみなし, ssifileに拡張子がなければ, 拡張子.ssiを付加して画面節移行ファイルを生成する。

- ssifile**に拡張子を指定する場合は、拡張子は**.ssi**でなければならない。
-P2, -M, -Ce, -Cj, -CS, -CG, -CH, -w0, -w1オプションとの同時指定はできない。
- 2** COBOL GUIの画面節移行機能を使用して、画面制御機能を実行する。
-P1, -Cj, -CS, -CG, -CH, -w0, -w1オプションとの同時指定はできない。
- Qc** 従来製品互換モードで動作する機能を指定するオプションである。
c には[**abcdeghijlmopq**]のうち 1 つ以上を指定する。
- a** 連絡節に **GLOBAL** 句の記述を可能とする。
 本オプションを指定すると連絡節のデータが静的項目となる。
OCCURS 句の **INDEXED** 指定があれば指標名も静的項目となる。
 再帰呼び出しするプログラムには指定してはいけない。
[Enterprise] -MT オプションとの同時指定はできない。
- b** ベースデータ節に **GLOBAL** 句の記述を可能とする。
 本オプションを指定するとベースデータ節のデータが静的項目となる。
BASED ON 句があればポインタ名も静的項目となる。
OCCURS 句の **INDEXED** 指定があれば指標名も静的項目となる。
 再帰呼び出しするプログラムには指定してはいけない。
[Enterprise] -MT オプションとの同時指定はできない。
- c** 指標名や指標データ項目を構成する出現番号と指標値を **COMP-2** とする。
 本オプションは、プログラムや関数やメソッドの引数／返却項目にも影響を与えるため、呼び出す側と呼び出される側の COBOL 原始プログラムについて、一方にオプションを指定する場合は、他方にも指定しなければならない。
- d** **[Standard]** 可変繰り返し回数が最大として **COMBINE** 文を実行する。
- e** 行順ファイルのレコード右端の空白を削除して書き出す。
 本オプションを指定すると、行順ファイルに対する **WRITE** 文および **REWRITE** 文は、**01** レコードの末尾より連続する英数字の空白を削除し、改行コードを付加して、レコードを書き出す。
 本オプション指定ありの場合、**READ** 文で読み込んだレコードの大きさが **01** レコードの大きさより小さい時、状態キーに"00"を設定する。
-CU オプション指定ありでコード変換してレコード長が変わるような場合、本オプシ

ョンは無効となる。

環境変数 COB_FILETAB 指定ありの場合, 本オプションは無効となる。

- g** [Standard] LINAGE 句の論理ページサイズの評価タイミングを変更する。
- h** [Standard] NUMERIC 検査の結果を変更する。
- i** [Standard] SYSIN ファイルの入力終了条件を指定する。
本オプションを指定すると, ファイルからの入力の場合には, 改行コードを入力したときに入力は終了する。
受け取り側データ項目の桁数が入力データの桁数より大きい場合, 受け取り側データ項目の残りの部分は空白で埋められる。
- j** [Standard] UNSTRING 文の P 付き整数項目の計数結果を変更する。
- l** [Standard] CLOSE 文の WITH LOCK 句をコメントとする。
- m** プログラム名と同じ名前の利用者定義語を定義できる。
定義可能な利用者定義語は, 次のいずれかである。
(1) データ名
(2) 条件名
(3) レコード名
(4) 指標名
(5) 呼び名
(6) 記号文字
(7) ポインタ名
(8) [Standard] 画面記述名
(9) [Enterprise for Windows] [Standard] 定数名
- o** [Standard] OPEN 文の入出力状態および永続誤り発生後の動作を変更する。
[Enterprise] OPEN 文の永続誤り発生後の動作を変更する。
・ [Standard] 相対ファイル(COBOL 独自形式)および索引ファイル(ISAM 形式)の OUTPUT 指定の OPEN 文でファイル名が不正な場合, 入出力状態値 99 にする。オプション指定なしのときは, 入出力状態値 30 になる。
・ 大容量記憶ファイル(COBOL DB アクセスの索引ファイルを除く)に対して, INPUT 指定/I-O 指定/EXTEND 指定の OPEN 文で永続誤り(入出力状態 35)が発生した

後, OUTPUT 指定の OPEN 文を実行可能とする。オプション指定なしのときは, すべてのファイルについて, INPUT 指定/I-O 指定/EXTEND 指定の OPEN 文で永続誤り (入出力状態 35)が発生した後, OUTPUT 指定の OPEN 文は実行できない。入出力状態値 30 になる。

- p** **SHARED-MODE** または **PROTECTED-MODE** 指定の **CSV** 指定なし行順ファイルに対して, ファイル共用機能を使用可能とする。
- q** **[Enterprise for Linux]** **CSV** 指定あり行順ファイルに対して, ファイル共用機能を使用可能とする。
- Rdir** リポジトリ段落で参照する外部リポジトリファイルの検索ディレクトリを指定する。
検索順は次のとおり。
(1) カレントディレクトリ
(2) 本オプションで指定されたディレクトリ (複数指定時は, 指定された順)
外部リポジトリの更新/生成ディレクトリは, 本オプションの指定によらずカレントディレクトリである。
[Enterprise for Linux] **dir** の前に空白を指定してもよい。
- Sc** **EXTERNAL** 句の記述されたデータを実行ファイル間で共有するため, データ定義を指定するオプションである。
c には [**i** **x**] のうち 1 つを指定する。
- i** **EXTERNAL** 句の記述されたデータを参照とする。
-Sx オプションとの同時指定はできない。
- x** **EXTERNAL** 句の記述されたデータを定義とする。
-Si オプションとの同時指定はできない。
- Ustring** 指定された文字列を予約語としないで翻訳を行う。ただし, 文字列は次の条件を満たさなければならない。

(1) 文字は英大文字でなければならない
(2) 文字列の長さは 31 文字以内でなければならない
(3) 次の文字列は指定してはならない

- (a) COPY
- (b) REPLACE
- (c) PREFIX
- (d) IS
- (e) IDENTIFICATION

- h** コンパイルオプションの一覧を表示する。
- G** [Enterprise for Windows] [Standard] DLL を生成する。
 -M オプションとの同時指定はできない。
 [Standard] -Yi, -Mw オプションとの同時指定はできない。
 [Enterprise for Linux] 共有ライブラリを生成する。
 -M オプションとの同時指定はできない。
- Gw** [Enterprise] REST API 対応の COBOL モジュールを生成する。
 -M, -Mw, -Fl, -Zc, -Zx オプションとの同時指定はできない。
 [Enterprise for Windows] -CF, -Zr オプションとの同時指定はできない。
 [Enterprise for Linux] -He, -Hf, -Hm, -T オプションとの同時指定はできない。
- vc** 初期値 VALUE 句が指定されていない基本データ項目に対して初期値を与えることを指定する。c には [i s z] のうち 1 つ, または, 16 進数 [0-9, a-f, A-F] で 2 桁の数字を指定する。指定がない場合は, 初期値を保証しない。
 作業場所節とファイル節に記述された基本データ項目を初期化対象とする。
- i** 初期値のない項目はすべて実行コード系に依存した "0" で表される値を初期値として与える。
- s** データ項目が英字, 英数字, 英数字編集, 数字編集の時には, VALUE IS SPACE が記述されたかのように初期値を与える。他の項目については次のとおりである。
- (1) データ項目の項類が数字のとき
 VALUE IS ZERO が指定されたかのように初期値を与える。
 - (2) データポインタデータ項目のとき
 NULL(空ポインタ)を初期値として与える。
 - (3) データ項目の項類が日本語および日本語編集項目のとき
 実行コード系に依存した日本語文字の空白を初期値として与える。

(4) ブールデータ項目のとき

ブール文字の "0" を初期値として与える。

z	データ項目の項類が英字, 英数字, 英数字編集, 数字編集のときは, VALUE IS ZERO が指定されたかのように初期値を与える。他の項目については -vs 指定のときと同様である。
nn	データ項目の項類が英字, 英数字, 英数字編集, 数字編集のときは, VALUE IS ALL " " nn " " が指定されたかのように初期値を与える。他の項目については -vs 指定の時と同様である (n は 16 進数文字 [0-9, a-f, A-F] の 1 文字を表す)。
-V	コンパイラのバージョンを表示する。
-WC	使用する予約語表を指定するオプションである。 c には [0 1 2] のうち 1 つを指定する。 本オプションの指定がない場合は " -w0 " が指定されたものとみなされる。 -Ustring オプションとの同時指定の場合, 予約語表で予約語とされた文字列であっても -Ustring で指定された文字列は予約語としない。(-Ustring の指定を有効とする。)
0	本コンパイラ既定の予約語表を使用する。 -w1 オプションまたは -w2 オプションとの同時指定はできない。
1	COBOL85 MFA および COBOL85 for IPF の予約語表を使用する。 -w0 オプションまたは -w2 オプションとの同時指定はできない。
2	A-VX COBOL85 および COBOL85 Pro の予約語表を使用する。 -w0 オプションまたは -w1 オプションとの同時指定はできない。
-Fc...	[Enterprise] ACOS-4 互換のためのオプションである。 c には [{abcdejlstvwxyzA r}] のうち 1 つ以上を指定できる。 ただし, Fr と他の ACOS-4 互換オプションは連結して指定できないため, 分けて指定しなければならない。
a	[Enterprise] データ項目の大きさとアラインを ACOS-4 互換とする。 ・1~2 桁の 2 進数を 2 バイトの連続した 16 ビットにする。 ・2 進数に SYNC 指定がある場合, 1-2 桁は 2 バイト境界,

10-18 桁は 4 バイト境界に桁づめする。

本オプションは、プログラムや関数やメソッドの引数／返却項目にも影響を与えるため、呼び出す側と呼び出される側の COBOL 原始プログラムについて、一方にオプションを指定する場合は、他方にも指定しなければならない。

- b** [Enterprise] データポインタデータ項目の割り付け境界を ACOS-4 互換とする。

 - ・SYNC 句の指定ができる。
 - ・SYNC 指定なしの場合、1 バイト境界に割り付ける。
 - ・SYNC 指定ありの場合、4 バイト境界に割り付ける。

本オプションは、プログラムや関数やメソッドの引数／返却項目にも影響を与えるため、呼び出す側と呼び出される側の COBOL 原始プログラムについて、一方にオプションを指定する場合は、他方にも指定しなければならない。

- c** [Enterprise] COMP 項目の内部表現形式を ACOS-4 互換の外部 10 進数とする。

本オプションは、プログラムや関数やメソッドの引数／返却項目にも影響を与えるため、呼び出す側と呼び出される側の COBOL 原始プログラムについて、一方にオプションを指定する場合は、他方にも指定しなければならない。

-C5 オプションとの同時指定はできない。

- d** [Enterprise] 比較規則を ACOS-4 互換とする。

 - ・非整数外部 10 進数と ZERO 以外の表意定数比較を表意定数比較とする。
 - ・算術式と一意名(英数字/英数字編集)の比較を文字比較とする。
 - ・JIPS(J)コード体系に基づく、日本語の大小比較を行う。
 - ・算術式の演算結果の整数部と指標名の出現番号を比較する。

- e** [Enterprise] NUMERIC 検査を ACOS-4 互換とする。

 - ・偶数桁の内部 10 進数の最左端上位 4 ビットが 0x0 以外の場合、偽とする
(-CN オプションが指定されていない時)。不正 10 進エラーにはならない。
 - ・外部 10 進数の独立符号部が空白の場合、真とする。不正 10 進エラーにはならない。

- j** [Enterprise] 小入出力先の既定値を ACOS-4 互換とする。

ACCEPT 文は SYSIN から入力する。DISPLAY 文は SYSOUT へ出力する。

- l** [Enterprise] STOP 文の ABORT 指定を可能とする。

 - ・異常終了時の終了値を環境変数 COB_ABTASK から取得する。

- rmm** **[Enterprise] COMPUTE** 文および条件式の間結果格納領域を指定した小数部桁数に基づく固定値とする。mm には、確保する中間結果格納領域の小数部桁数を指定する。指定値は、1～30 の整数でなければならない。
- ACOS-4 COBOL85 と本 COBOL は中間結果格納領域の生成規則に違いがあるため、ACOS-4 COBOL85 の FIXIR オプションと本オプションは完全互換ではない。各 COBOL の生成規則を理解し注意して使用すること。本 COBOL の中間結果格納領域については「5.3.1 算術文における中間結果」を参照。
- s** **[Enterprise]** 英数字 (英数字項目, 英数字定数, ALL 英数字定数) から外部 10 進数への数字転記において、ゾーン部分のビットの補正を行わない。
- t** **[Enterprise]** 符号なし外部 10 進数の参照時に最右端バイトを数字と符号の重ね合わせの文字として処理する。符号なし、または、正符号付きの値の場合、正の数として扱う。負符号付きの値の場合、負の数として扱う。不正な値の場合、不正 10 進エラーとなる。
- 例)
- 01 A PIC 9.
- 01 B PIC 9.
- A の値が""7B""(+0)の場合,
- DISPLAY A.
- の実行結果は+0 となる。
- A の値が""4A""(-1)の場合,
- ADD 1 TO A.
- の実行結果は 0 となる。
- A の値が""52""(-9)の場合,
- MOVE A TO B.
- の実行結果は 9 となる。送り出し側は本オプションにより-9 として扱うが、受け取り側は本オプションの対象外であるため絶対値 9 となる。
- v** **[Enterprise for Windows]** SYSOUT ファイルに対する WRITE 文の ADVANCING 句に機能名 LN001～LN255 に対応する呼び名を指定することで、SYSOUT ファイルの改行位置の絶対行指定を可能とする。
- 本オプションは、SYSOUT ファイルのレコード形式が SSF の場合のみ有効となる。
- 本オプションを指定する場合は、実行可能プログラム内のすべての COBOL 原始プログラムに指定しなければならない。

-MT オプションとの同時指定はできない。

- w** **[Enterprise for Windows] ROUNDED** 指定のある **COMPUTE** 文において中間結果格納領域の算出方法を **ACOS-4** 互換とする。
COMPUTE 文の受け取り側項目に **ROUNDED** が指定され、かつその受け取り側項目の小数部桁数が小数部桁数の最大値 (**DMAX**) である場合、小数部桁数の最大値 (**DMAX**) に 1 を加える。
 -Fr オプションと同時に指定された場合、中間結果格納領域の小数部桁数は、-Fr オプションで指定された小数部桁数が使用される。
- x** **[Enterprise]** 符号なし外部 10 進数を次のように処理する。**Ft** との同時指定はできない。
 ・送り出し側が符号なし外部 10 進数、受け取り側が英数字項目の場合、符号なし外部 10 進数の最右端バイトを数字と符号の重ね合わせの文字と見なせずに英数字転記する。不正な値の場合、不正 10 進エラーとならない。
 ・送り出し側が符号なし外部 10 進数、受け取り側が日本語項目の場合、符号なし外部 10 進数の最右端バイトを数字と符号の重ね合わせの文字と見なせずに日本語転記する。不正な値の場合、不正 10 進エラーとならない。
 ・その他の場合、符号なし外部 10 進数の最右端バイトを数字と符号の重ね合わせの文字と見なして処理する。不正な値の場合、不正 10 進エラーとなる。**Ft** 指定と同じである。
- y** **[Enterprise for Windows] PPR-CONTROL-1** 指定の **SYSOUT** ファイルの印刷データに、シフトコードおよびプリンタ制御コードを埋め込む。**SSF** 形式の **SYSOUT** ファイルに **APPLY SHIFT-CODE** 句の **WITH PPR-CONTROL-1** 指定ありのとき、レコード名または **WRITE** 文の **FROM** 一意名に、日本語データ項目があればシフトコードを、**CHARACTER TYPE** 句があれば書体指定のプリンタ制御コードを、**COLUMN NUMBER** 句があれば印字位置指定のプリンタ制御コードを出力する。
 プリンタファイルや **PRN** 形式または **TXT** 形式の **SYSOUT** ファイル、**APPLY SHIFT-CODE** 句の **WITH PPR-CONTROL-1** 指定なしの **SYSOUT** ファイルに対して、このオプションは意味を持たない。
- z** **[Enterprise for Windows]** データ例外が発生する手続き文を **ACOS-4** 互換とする。
 次の手続き文について、不正 10 進チェックを行う。
 (1) **DISPLAY** 文の作用対象が数字項目のとき

- (2) COMBINE 文の SHIFTED 句の作用対象が外部 10 進項目のとき
- (3) SORT 文または MERGE 文の KEY 句の作用対象が数字項目のとき

- A** **[Enterprise for Windows]** DIVIDE 文の剰余結果を ACOS-4 互換とする。
剰余は符号付きの補助的な商を用いて計算する。
- T dbname** **[Enterprise for Linux]** 用語機能を使用する。
dbname には使用する用語辞書データベース名を指定する。
-Cj, -CU, -CG, -CH オプションとの同時指定はできない。
- Wcarglist** **[Enterprise for Linux]** gcc(1) コンパイラのオプションに **arglist** を指定する。

COB コマンドの引数に指定した C または C++ プログラムにのみ有効である。
arglist の先頭はコンマ(,)でなければならない。
- Wlarglist** **[Enterprise for Windows]** **[Standard]** リンカのオプションに **arglist** を指定する。
arglist の先頭はコロン(:)でなければならない。
[Enterprise for Linux] リンカのオプションに **arglist** を指定する。
arglist の先頭はコンマ(,)でなければならない。
- Xi** INITIALIZE 文の展開順序を COBOL85 規格仕様とする。
複数の初期化対象を指定順序に従って、単一の INITIALIZE 文で初期化する。
本オプション指定なしの場合、複数の初期化対象を指定順序に従って、別々の INITIALIZE 文で初期化する。
- q** **[Standard]** **[Enterprise for Windows]** コンパイルを行うとき、マニフェストファイルを COBOL アプリケーションファイルに書き込まずに単独のファイルとして生成する。
- p** **[Standard]** **[Enterprise for Windows]** コンパイルを行うとき、マニフェストファイルを COBOL アプリケーションファイルに書き込み、単独のファイルとして生成しない。
- Yflstfile** **[Standard]** **lstfile.list** という名前で翻訳リストを生成する。

lstfile には、フルパス名または相対パス名が指定できる。**lstfile** で拡張子が省略された場合、自動的に **.list** を付ける。

lstfile にドライブ名またはディレクトリ名のみを指定した場合、コマンドラインで指定された原始プログラムファイルに対応する翻訳リスト(ファイルのベース名に **.list** を付けたもの)を、指定されたドライブやディレクトリに生成する。

lstfile にディレクトリ名を指定する場合は、指定した名前の最後に必ず円記号 "¥" を付ける。

本オプションで指定するファイル名およびパス名はシフト JIS でなければならない。

本オプションは、翻訳リストの出力を要求する **-Hc** オプションとともに使用された場合にのみ有効である。

また、本オプションは COBOL ソースの後ろには指定できない。

ドライブ名またはディレクトリ名の指定例を記す。

例) カレントディレクトリ **d:¥test¥**

(1) **cob -Yfsample -Hsc -M prog1.cbl**

原始プログラムファイル **prog1.cbl** がコンパイルされ、カレントディレクトリ (**d:¥test¥**) に **sample.list** という名前で翻訳リストが生成される。

(2) **cob -Yfc:¥list¥ -Hsc -M prog1.cbl prog2.cbl**

c ドライブの **¥list** というディレクトリが存在する場合、原始プログラムファイル **prog1.cbl**, **prog2.cbl** がコンパイルされ、**c** ドライブの **¥list** というディレクトリに、それぞれ **prog1.list**, **prog2.list** という名前で翻訳リストが生成される。

(3) **cob -Yfc:¥list¥ -Hsc -M prog1.cbl**

c ドライブの **¥list** というディレクトリが存在しない場合、エラーを出力しコンパイル処理は停止する。

(4) **cob -Yfc: -Hsc -M prog1.cbl**

原始プログラムファイル **prog1.cbl** がコンパイルされ、**c** ドライブのカレントディレクトリ (**c:¥test¥**) に、**prog1.list** という名前で翻訳リストが生成される。

-Yiiconfile

[Standard] **iconfile** という名前のアイコンを使用する。

iconfile には、フルパス名または相対パス名が指定できる。

本オプションは、**-M**, **-Mw** オプションとともに使用された場合にのみ有効である。

-c, **-G** オプションとの同時指定はできない。

本オプションで指定するファイル名およびパス名はシフト JIS でなければならない。

-Yobjfile [Standard] objfile.obj という名前でオブジェクトファイルを生成する。
objfile には, フルパス名または相対パス名が指定できる。
objfile で拡張子が省略された場合, 自動的に拡張子.obj を付加する。
objfile にドライブ名またはディレクトリ名のみを指定した場合, コマンドラインで指定されたすべての原始プログラムファイルに対応するオブジェクトファイル(各原始プログラムファイルのベース名に.obj を付けたもの)を, 指定されたドライブやディレクトリに生成する。objfile にディレクトリ名を指定する場合は, 指定した名前の最後に必ず円記号"¥"を付ける。
本オプションで指定するファイル名およびパス名はシフト JIS でなければならない。
また, 本オプションは COBOL ソースの後ろには指定できない。

ドライブ名またはディレクトリ名の指定例を記す。

例) カレントディレクトリ d:¥test¥

(1) cob -Yosample -M prog1.cbl

原始プログラムファイル prog1.cbl がコンパイルされ, カレントディレクトリ(d:¥test¥)に sample.obj という名前でオブジェクトファイルが生成される。
prog1.cbl が複数翻訳単位の場合, 1 つ目の定義は sample.obj となり, 2 つ目の定義は sample_2 つ目の定義名.obj という名前でオブジェクトファイルが生成される。

(2) cob -Yoc:¥objdir¥ -M prog1.cbl prog2.cbl

c ドライブの¥objdir というディレクトリが存在する場合, 原始プログラムファイル prog1.cbl, prog2.cbl がコンパイルされ, c ドライブの¥objdir というディレクトリに, それぞれ prog1.obj, prog2.obj という名前でオブジェクトファイルが生成される。

(3) cob -Yoc:¥objdir¥ -M prog1.cbl prog2.cbl

c ドライブの¥objdir というディレクトリが存在しない場合, エラーを出力しコンパイル処理は停止する。

(4) cob -Yoc: -M prog1.cbl

原始プログラムファイル prog1.cbl がコンパイルされ, c ドライブのカレントディレクトリ(c:¥test¥)に, prog1.obj という名前でオブジェクトファイルが生成される。

-Yrregistrykey [Standard] レジストリ機能を利用して, アプリケーションごとまたは複数のアプリケーション共通に, プログラムに対する各種設定を行う。本オプションは, このレジス

トリのキー名を指定できる。

本オプションを指定すると, COBOL コンパイラは **registrykey** でオブジェクト中にレジストリのキー名を埋め込む。

registrykey に日本語は指定できない。

registrykey は **<CompanyName>¥<ProductName>** で指定する必要がある。

本オプションは, -M または -Mw オプションとともに使用された場合にのみ有効である。

例)

(1) **cob -Yrnc¥prog1**

-Zc

ファイル入出力レコード情報ファイルまたはデータレコード情報ファイルを生成する。本オプション指定なしの場合, ファイル入出力レコード情報ファイルまたはデータレコード情報ファイルは生成しない。

c には **[r | c | x | a]** のいずれか 1 つを指定する。

a

PROCEDURE DIVISION USING データ名および **RETURNING** データ名について, オブジェクトファイルと同じフォルダに CSV 形式のデータレコード情報を出力する。区切り文字はコンマである。

生成ファイル名は, COBOL 原始プログラム名+拡張子 **csv** である。

同時指定して有効な既存の翻訳オプションは, -Ca, -Ce, -Cq, -Cu, -Cy, -CN, -C5, -I, -Dy, -Dn, -N[0-9], -Na, -U, -w0, -w1, -w2, -Fa, -Fb, -Fc, -Qc, -Qm である。

その他の既存の翻訳オプションは同時指定しても無効である。

r

[Standard] [Enterprise for Windows]

RFD 形式のファイル入出力レコード情報ファイルを生成する。

区切り文字はセミコロンである。生成するファイルは次のとおり。

- ・大容量記憶ファイルの順編成, CSV 指定なし行順編成, 相対編成, 索引編成
- ・ **[Standard]** RDB ファイルの順編成, 相対編成, 索引編成

生成ファイル名は, ファイル記述項のファイル名+拡張子 **rfd** である。

同時指定して有効な既存の翻訳オプションは, -Ca, -Ce, -Cq, -Cu, -Cy, -C5, -I, -Dy, -Dn, -N[0-9], -Na, -U, -w0, -w1, -w2, -Fa, -Fb, -Fc, -Qc, -Qm である。

その他の既存の翻訳オプションは同時指定しても無効である。

RECORD 句(書き方 1,3,4,5)指定あり, かつ **RECORD** 句の最大値がレコード記述項のレコード定義長より大きい, かつ行順ファイルではない場合, **RECORD** 句(書き方 1,3,4,5)の最大値を最大レコード長とする。

その他の場合, **-Zr/-Zc/-Zx** オプション指定なしのときの最大レコード長とする。

- c** **CSV 形式のファイル入出力レコード情報ファイル**を生成する。
 区切り文字はコンマである。生成するファイルは次のとおり。
- ・大容量記憶ファイルの順編成, **CSV** 指定なし行順編成, 相対編成, 索引編成
 - ・プリンタファイル, **SYSIN** ファイル, **SYSOUT** ファイル
 - ・**[Standard]** RDB ファイルの順編成, 相対編成, 索引編成
- 生成ファイル名は, ファイル記述項のファイル名+拡張子 **csv** である。
 同時指定して有効な既存の翻訳オプションは, **-Ca, -Ce, -Cq, -Cu, -Cy, -CN, -C5, -I, -Dy, -Dn, -N[0-9], -Na, -U, -w0, -w1, -w2, -Fa, -Fb, -Fc, -Qc, -Qm** である。
 その他の既存の翻訳オプションは同時指定しても無効である。
RECORD 句(書き方 1,3,4,5)指定あり, かつ **RECORD** 句の最大値がレコード記述項のレコード定義長より大きい, かつ行順ファイルではない場合, **RECORD** 句(書き方 1,3,4,5)の最大値を最大レコード長とする。
 その他の場合, **-Zr/-Zc/-Zx** オプション指定なしのときの最大レコード長とする。

- x** **CSV 形式のファイル入出力レコード情報ファイル**を生成する。
 区切り文字はコンマである。生成するファイルは次のとおり。
- ・大容量記憶ファイルの順編成, **CSV** 指定なし行順編成, 相対編成, 索引編成
 - ・プリンタファイル, **SYSIN** ファイル, **SYSOUT** ファイル
 - ・**[Standard]** RDB ファイルの順編成, 相対編成, 索引編成
- 生成ファイル名は, ファイル記述項のファイル名+拡張子 **csv** である。
 同時指定して有効な既存の翻訳オプションは, **-Ca, -Ce, -Cq, -Cu, -Cy, -CN, -C5, -I, -Dy, -Dn, -N[0-9], -Na, -U, -w0, -w1, -w2, -Fa, -Fb, -Fc, -Qc, -Qm** である。
 その他の既存の翻訳オプションは同時指定しても無効である。
RECORD 句(書き方 1,3,4,5)指定あり, かつ **RECORD** 句の最大値がレコード記述項のレコード定義長より大きい, かつ行順ファイルではない場合, レコード記述項のレコード定義長を最大レコード長とする。
 その他の場合, **-Zr/-Zc/-Zx** オプション指定なしのときの最大レコード長とする。

[Enterprise for Windows] **[Standard]** 他の引数は, リンクエディタの引数, **COBOL** と互換性をもつルーチンのライブラリまたは一般に生成される **COBOL** と互換性をもつオブジェクトプログラムとみなされる。これらのプログラムは, 指定された翻訳結果と合わせて, リンクエディットの **-o** オプションが使用されないかぎり, ベース名に拡張子 **.exe** を付けた名前の実行可能プログラムまたは **-G** オプションが使用されている場合はベース名に拡張子 **.dll** を付けた名前の **DLL** を生成するために(指定された順序で)リンクエディットされる。

[Enterprise for Linux] 他の引数は, gcc(1)コンパイラの引数, COBOL と互換性をもつルーチンのライブラリまたは一般に生成される COBOL と互換性をもつオブジェクトプログラムとみなされる。これらのプログラムは, 指定された翻訳結果と合わせて, -o オプションが使用されないかぎり, a.out という名前の実行可能プログラムまたは共有ライブラリを生成するために(指定された順序で)リンクされる。

関連ファイル

file.cob	原始プログラムまたは登録集原文
file.cbl	
file.cbf	
[Standard] file.cbx	
file.c	C または C++プログラム
file.cpp	
file.cc	[Enterprise for Linux] C または C++プログラム
file.cp	
file.cxx	
file.CPP	
file.c++	
file.C	
file.obj	[Enterprise for Windows] [Standard] オブジェクトプログラム
file.o	[Enterprise for Linux] オブジェクトファイル
file.list	翻訳リスト
file.ico	[Standard] アイコンファイル
file.exe	[Enterprise for Windows] [Standard] 実行可能プログラムファイル
a.out	[Enterprise for Linux] 実行可能プログラムファイルまたは共有ライブラリ

file.exp	[Enterprise for Windows] [Standard] エクスポートファイル
file.lib	[Enterprise for Windows] [Standard] インポートライブラリ
file.dll	[Enterprise for Windows] [Standard] DLL
file.rpg	外部リポジトリファイル
file.rph	外部リポジトリファイル (クラス定義)
cob.exe cobfe.exe cobbe.exe	[Enterprise for Windows] [Standard] COBOL コンパイラ
neccob*.lib	[Enterprise for Windows] [Standard] COBOL ライブラリ
neccob*.dll	[Enterprise for Windows] [Standard] COBOL ランタイム本体
CSGETWINHANDLE.lib	[Standard] COBOL メインウィンドウのウィンドウハンドル取得ライブラリ
CSCOMPEEWV1.lib	[Enterprise for Windows] V1 互換ライブラリ (V1 で生成した COBOL メインプログラムのオブジェクトプログラムを使用する場合, 本ライブラリをリンク時に指定する)
cob cobfe cobbe	[Enterprise for Linux] COBOL コンパイラ
libneccobxml.so	[Enterprise for Linux] XML パーサアクセスライブラリ
libneccob*.a	[Enterprise for Linux] COBOL ライブラリ
libneccob*.so	[Enterprise for Linux] COBOL ランタイム本体

CSFINISH.h 他言語連携用ヘッダファイル

CSHANDLER.h

CSINTERFACE.h

[Standard] CSLNK.h

CSFINISH_MT.H [Enterprise] 他言語連携用ヘッダファイル (マルチスレッド用)

環境変数

[Enterprise for Windows] [Standard] 翻訳, リンク時に PATH, LIB, TMP の環境変数を参照する。環境変数 PATH, LIB については, Visual Studio のバッチを実行することで必要な設定を行う。詳細は「19.1.2 翻訳環境の設定」を参照。

環境変数 TMP は, テンポラリファイル(一時的な作業用のファイル)を作るディレクトリをコンパイラに指示する。テンポラリファイルには, 原始プログラムからオブジェクトプログラムを生成するための中間ファイル, クラス定義翻訳時のクラスヘッダファイル (命名規則は外部リポジトリファイルと同じ), 翻訳処理のためのバイナリファイルがある。

テンポラリファイルはコンパイラが処理を終了するまでに削除する。環境変数 TMP が未設定の場合, 環境変数 TEMP を参照する。環境変数 TMP および TEMP が未設定の場合, テンポラリファイルはカレントディレクトリに生成する。

[Standard] [Enterprise for Windows] -CF オプションを指定すると, 翻訳時に環境変数 FA_JIS2004_CDMAP を参照する。環境変数 FA_JIS2004_CDMAP には「FontAvenue UniAssist コード変換」が提供するユーザ定義マッピングファイルの配置先フォルダ名を絶対パスで指定しておかなければならない。

[Enterprise for Linux] 翻訳, リンク時に TMPDIR, LANG の環境変数を参照する。

環境変数 TMPDIR は, テンポラリファイル(一時的な作業用のファイル)を作るディレクトリをコンパイラに指示する。テンポラリファイルには, 原始プログラムからオブジェクトプログラムを生成するための中間ファイル, クラス定義翻訳時のクラスヘッダファイル (命名規則は外部リポジトリファイルと同じ), 翻訳処理のためのバイナリファイルがある。

テンポラリファイルはコンパイラが処理を終了するまでに削除する。環境変数 TMPDIR が未設定の場合, または, 設定されたディレクトリに生成できなければ, /tmp ディレクトリに生成する。さらに, /tmp ディレクトリに生成できなければ, カレントディレクトリに生成する。

環境変数 LANG には, 翻訳/実行時エラーメッセージの出力コードを指定する。

また, 小入出力文の入出力コードを指定する。

環境変数 LANG に"ja_JP.SJIS"を指定すると, 翻訳／実行時エラーメッセージをシフト JIS コードの日本語で出力する。

環境変数 LANG に"ja_JP.UTF-8"を指定すると, 翻訳／実行時エラーメッセージを ASCII コードで出力する。環境変数 COB_CODECHG と組み合わせることで, 実行時エラーメッセージおよび小入出力文の入出力コードを指定する意味を持つ。

環境変数 LANG が未設定または"ja_JP.SJIS", "ja_JP.UTF-8"以外を指定すると, 翻訳／実行時エラーメッセージを ASCII コードで出力する。

関連項目

gcc(1)

診断

翻訳の結果により, 終了コードは次のように設定される。

(1)-Cq オプションが指定されている場合

エラーが検出されなかった場合 0, 警告(WARNING)エラーまたは注意(OBSERVE)が検出された場合 0, 致命的(FATAL)エラーが検出された場合 2, その他のエラーの場合 3 以上

(2)-Cq オプションが指定されていない場合

エラーが検出されなかった場合 0, 警告(WARNING)エラーまたは注意(OBSERVE)が検出された場合 1, 致命的(FATAL)エラーが検出された場合 2, その他のエラーの場合 3 以上

1.4.3 同一翻訳群の原始プログラムの翻訳時の注意点

本コンパイラでは、ひとつの原始プログラムの中に、複数のプログラム定義、関数定義、クラス定義を記述することができる。このような形式の原始プログラムを同一翻訳群と呼ぶ。

同一翻訳群の原始プログラムを翻訳すると、オブジェクトプログラムが生成されるが、本コンパイラは次のような命名規則でオブジェクトプログラムを生成する。

[Enterprise for Windows] [Standard]

- 1 つ目の定義に対して: ソースファイル名.obj
- 2 つ目の定義に対して: ソースファイル名_2 つ目の定義名.obj
- 3 つ目の定義に対して: ソースファイル名_3 つ目の定義名.obj
- :

例) **sample.cob** というソースファイルの中で、関数 **FUNC1**, 関数 **FUNC2**, 関数 **FUNC3**, クラス **CL1**, クラス **CL2**, プログラム **PROG1** を定義している場合、このプログラムを翻訳すると、

sample.obj	… 関数 FUNC1 に対するオブジェクトプログラム
sample_FUNC2.obj	… 関数 FUNC2 に対するオブジェクトプログラム
sample_FUNC3.obj	… 関数 FUNC3 に対するオブジェクトプログラム
sample_CL1.obj	… クラス CL1 に対するオブジェクトプログラム
sample_CL2.obj	… クラス CL2 に対するオブジェクトプログラム
sample_PROG1.obj	… プログラム PROG1 に対するオブジェクトプログラム

の 6 つのオブジェクトプログラムを生成する。

[Enterprise for Linux]

- 1 つ目の定義に対して: ソースファイル名.o
- 2 つ目の定義に対して: ソースファイル名_2 つ目の定義名.o
- 3 つ目の定義に対して: ソースファイル名_3 つ目の定義名.o
- :

例) **sample.cob** というソースファイルの中で、関数 **FUNC1**, 関数 **FUNC2**, 関数 **FUNC3**, クラス **CL1**, クラス **CL2**, プログラム **PROG1** を定義している場合、このプログラムを翻訳すると、

sample.o	… 関数 FUNC1 に対するオブジェクトプログラム
sample_FUNC2.o	… 関数 FUNC2 に対するオブジェクトプログラム
sample_FUNC3.o	… 関数 FUNC3 に対するオブジェクトプログラム

sample_CL1.o … クラス CL1 に対するオブジェクトプログラム
 sample_CL2.o … クラス CL2 に対するオブジェクトプログラム
 sample_PROG1.o … プログラム PROG1 に対するオブジェクトプログラム

の 6 つのオブジェクトプログラムを生成する。

上記のように、同一翻訳群のプログラムにすると、各定義と、生成されるオブジェクトプログラムとの対応が分かりづらくなる。特に、AS 定数や-Cs オプションを使用して大文字／小文字だけが異なる定義を同一翻訳群に記述するとオブジェクトプログラムを上書きし、正しくコンパイルできない。したがって、原始プログラムを作るときには、1つの原始プログラムに複数の定義を記述するのではなく、ひとつの原始プログラムにひとつの定義を記述するとよい。

1.4.4 予約語切り替え機能

COBOL85 で作成したプログラムで、本コンパイラで追加された予約語を利用者定義語として使用していると、致命的(FATAL)エラーになる場合がある。

ソース互換を確保するには、次のいずれかで対処する。

- (1) 予約語としないオプション(-Ustring)によって、指定した語を予約語としないで翻訳する。
- (2) 予約語表を指定するオプション(-wc)によって、COBOL85 互換の予約語を使用して翻訳する。

予約語表を指定するオプション(-wc)の指定により、予約語表で予約語とされた語を予約語としないで翻訳したい場合は、予約語としないオプション(-Ustring)を同時に指定することで対処する。

オプション指定の書式は、「1.4.2 翻訳用コマンドとその使用法」を参照。

1.5 プログラムのリンク

1.5.1 実行可能プログラムの生成

実行可能プログラムを生成するには、「1.4 原始プログラムの翻訳(コンパイル)およびプログラムの結合」に示したコマンド(cob)を用いる。複数のプログラムをリンクして、実行可能プログラムを

生成する方法を以降に示す。

- (1) [Enterprise for Windows] [Standard] メインプログラムを除いた, サブプログラムの原始プログラムを **cob** で翻訳し, オブジェクトプログラムを生成する。

例 **cob -c sub1.cob sub2.cob**

sub1.cob と **sub2.cob** のサブプログラムのオブジェクトプログラム, **sub1.obj** と **sub2.obj** を生成する。**-c** は翻訳だけを行うように指定するオプションである。

[Enterprise for Linux] メインプログラムを除いた, サブプログラムの原始プログラムを **cob** で翻訳し, オブジェクトプログラムを生成する。

例 **cob -c sub1.cob sub2.cob**

sub1.cob と **sub2.cob** のサブプログラムのオブジェクトプログラム, **sub1.o** と **sub2.o** を生成する。**-c** は翻訳だけを行うように指定するオプションである。

- (2) [Enterprise for Windows] [Standard] 次に, メインプログラムの原始プログラムと, (1)で生成したオブジェクトプログラムを **cob** で翻訳およびリンクし, 実行可能プログラムを生成する。

例 (バッチ型アプリケーション)

cob -M main.cob sub1.obj sub2.obj

[Standard] 例 (対話型アプリケーション)

cob -Mw main.cob sub1.obj sub2.obj

main.cob が翻訳され, そのオブジェクトと **sub1.obj**, **sub2.obj** がリンクされ, **main.exe** という実行可能プログラムを生成する。

[Enterprise for Windows] **-M** は翻訳するプログラムがメインプログラムであることを指定するオプションである。実行可能プログラム名を他の名前にしたいときは, **-o** オプションで他の名前を指定する。

[Standard] **-M** または **-Mw** は翻訳するプログラムがメインプログラムであることを指定するオプションである。実行可能プログラム名を他の名前にしたいときは, **-o** オプションで他の名前を指定する。

[Enterprise for Linux] 次に, メインプログラムの原始プログラムと, (1)で生成したオブジェク

トプログラムを **cob** で翻訳およびリンクし, 実行可能プログラムを生成する。

例 **cob -M main.cob sub1.o sub2.o**

main.cob が翻訳され, そのオブジェクトと **sub1.o**, **sub2.o** がリンクされ, **a.out** という実行可能プログラムを生成する。**-M** は翻訳するプログラムがメインプログラムであることを指定するオプションである。実行可能プログラム名を他の名前にしたいときは, **-o** オプションで他の名前を指定する。

1.5.2 リンク用コマンドの指定方法

cob は、リンク用コマンド (注 1) を呼び出してリンクを行う。その際、**cob** は、リンク用コマンドに対して、**cob** コマンドが指定したライブラリをリンクパラメータに指定する。

次に **cob** のリンクパラメータの指定方法を示す。

注意

(注 1) [Enterprise for Windows] [Standard] link
 [Enterprise for Linux] gcc(1)

(1) メインプログラムが COBOL プログラム、かつ、バッチ型アプリケーションを作成する場合

例 > **cob -M main.cob sub.cob**

このときリンク用コマンドには次のようにオプションが指定される。

[Enterprise for Windows]

```
link /subsystem:console /entry:mainCRTStartup /NODEFAULTLIB /LIBPATH:
"(COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib neccobs.lib
msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"main.exe" "main.obj"
"sub.obj"
```

[Standard]

```
link /subsystem:console /entry:mainCRTStartup /NODEFAULTLIB /LIBPATH:
"(COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib neccobs.lib
msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"main.exe" "main.obj"
"sub.obj"
```

SE 動作モードは、次のとおりである。

```
link /subsystem:console /entry:mainCRTStartup /NODEFAULTLIB /LIBPATH:
"(COBOL インストールディレクトリ)¥lib¥x64¥semode" neccobn.lib neccob.lib
neccobs.lib msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib
/OUT:"main.exe" "main.obj" "sub.obj"
```

[Enterprise for Linux]

```
gcc -oa.out main.o sub.o -L"(COBOL インストールディレクトリ) "/bin/../../lib/  
-lneccobn -lneccob -lneccobs -lpthread -lstdc++ -lm -ldl -lc
```

(2) メインプログラムが他言語(C プログラム)の場合

[Enterprise for Windows] [Standard]

```
例 > cl -c main.c  
      > cob main.obj sub.cob
```

[Enterprise for Linux]

```
例 > gcc -c main.c  
      > cob main.o sub.cob
```

このときリンク用コマンドには次のようにオプションが指定される。

[Enterprise for Windows]

```
link /subsystem:console /entry:mainCRTStartup /NODEFAULTLIB /LIBPATH:"  
(COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib msvcrt.lib  
kernel32.lib vcruntime.lib ucrt.lib /OUT:"main.exe" "main.obj" "sub.obj"
```

[Standard]

```
link /subsystem:console /entry:mainCRTStartup /NODEFAULTLIB /LIBPATH:"  
(COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib msvcrt.lib  
kernel32.lib vcruntime.lib ucrt.lib /OUT:"main.exe" "main.obj" "sub.obj"  
SE 動作モードは, 次のとおりである。
```

```
link /subsystem:console /entry:mainCRTStartup /NODEFAULTLIB /LIBPATH:"  
"(COBOL インストールディレクトリ)¥lib¥x64¥semode" neccobn.lib neccob.lib  
msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"main.exe" "main.obj"  
"sub.obj"
```

[Enterprise for Linux]

```
gcc -oa.out main.o sub.o -L"(COBOL インストールディレクトリ) "/bin/../../lib/  
-lneccobn -lneccob -lneccobs -lpthread -lstdc++ -lm -ldl -lc
```

(3) [Enterprise for Windows] [Standard] DLL 生成の場合

例 > `cob -G sub.cob`

このときリンク用コマンドには次のようにオプションが指定される。

[Enterprise for Windows]

```
link /subsystem:console /NODEFAULTLIB /LIBPATH:" (COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"sub.dll" /IMPLIB:"sub.lib" "/DLL" "sub.obj"
```

[Standard]

```
link /subsystem:console /NODEFAULTLIB /LIBPATH:" (COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"sub.dll" /IMPLIB:"sub.lib" "/DLL" "sub.obj"
```

SE 動作モードは, 次のとおりである。

```
link /subsystem:console /NODEFAULTLIB /LIBPATH: "(COBOL インストールディレクトリ)¥lib¥x64¥semode" neccobn.lib neccob.lib msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"sub.dll" /IMPLIB:"sub.lib" "/DLL" "sub.obj"
```

(4) [Enterprise for Linux] 共有ライブラリ生成の場合

例 > `cob -G sub.cob`

このときリンク用コマンドには次のようにオプションが指定される。

```
gcc -oa.out -shared sub.o -L"(COBOL インストールディレクトリ)"/bin/../../lib/ -lneccobn -lneccob -lpthread -lstdc++ -lm -ldl -lc
```

(5) [Enterprise for Linux] 環境変数 COB_LINK_PARAM による指定

```
例 > export COB_LINK_PARAM="-lneccob -luser1 -luser2 ... -lusern"
> cob -M main.cob sub.cob
```

このときリンク用コマンドには次のようにオプションが指定される。

```
gcc -oa.out main.o sub.o -lneccob -luser1 -luser2 ... -lusern
```

(6) [Enterprise for Linux] cob コマンドおよび環境変数 COB_LINK_PARAM による指定

```
例 > export COB_LINK_PARAM="-lneccob -luser1 -luser2 ... -lusern"  
> cob -M main.cob sub.cob -luser0
```

このときリンク用コマンドには次のようにオプションが指定される。

```
gcc -oa.out main.o sub.o -luser0 -lneccob -luser1 -luser2 ... -lusern
```

(7) [Enterprise] マルチスレッドアプリケーションを生成する場合

```
例 > cob -G -MT sub.cob
```

このときリンク用コマンドには次のようにオプションが指定される。

[Enterprise for Windows]

```
link /subsystem:console /NODEFAULTLIB /LIBPATH: "(COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccobm.lib msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"sub.dll" /IMPLIB:"sub.lib" "/DLL" "sub.obj"
```

[Enterprise for Linux]

```
gcc -oa.out -shared sub.o -L"(COBOL インストールディレクトリ)/bin/../../lib/  
-lneccobn -lneccobm -lpthread -lstdc++ -lm -ldl -lc
```

(8) [Standard] メインプログラムが COBOL プログラム, かつ, 対話型アプリケーションを作成する場合

```
例 > cob -Mw main.cob sub.cob
```

このときリンク用コマンドには次のようにオプションが指定される。

```
link /subsystem:windows /entry:WinMainCRTStartup /NODEFAULTLIB /LIBPATH:  
"(COBOL インストールディレクトリ)¥lib¥x64" neccobn.lib neccob.lib neccobs.lib  
msvcrt.lib kernel32.lib vcruntime.lib ucrt.lib /OUT:"main.exe" "main.obj"  
"sub.obj"
```

リンクパラメータに誤った値を設定した場合には、リンク用コマンド実行時にエラーとなる。

COBOL のライブラリおよび必要なライブラリは、COBOL コンパイラのリリースバージョンによって変更される。

他言語が必要とするライブラリの指定は、他言語の説明書を参照の上、使用すること。

[Enterprise for Linux] gcc を使用してリンクする場合は、ライブラリの指定順序に注意が必要である。スレッドライブラリの指定(-lpthread)は、共用 C ライブラリの指定(-lc)より前でなければならない。

1.5.3 マニフェストの埋め込み

[Enterprise for Windows] [Standard] マニフェストは、Windows 版でのみ使用できる。

[Enterprise for Windows] [Standard] Visual C++ 2015, Visual C++ 2017, Visual C++ 2019 および Visual C++ 2022 では、マニフェスト(.exe.manifest)が生成されない。

本コンパイラは翻訳既定値ではマニフェストファイルを生成しない。

マニフェストファイルを生成する場合は、翻訳オプションを指定する。

-p オプションありのとき、リンカオプションに/MANIFESTを付与してマニフェストファイルを生成後、マニフェストの埋め込み処理を行う。

-q オプションありのとき、リンカオプションに/MANIFEST を付与してマニフェストファイルを生成するが、マニフェストの埋め込み処理は行わない。

cob コマンドは、マニフェストの埋め込みコマンド(mt.exe)を用いて、リンク処理後にマニフェストの埋め込み処理を行う。

(1)実行モジュール生成の場合

例 > cob -M -p main.cob sub.cob

このときリンク処理の後、mt.exe コマンドには次のようにオプションが指定される。

```
mt.exe -manifest main.exe.manifest -outputresource:main.exe;1
```

(2)DLL 生成の場合

例 > **cob -G -p sub.cob**

このときリンク処理の後, **mt.exe** コマンドには次のようにオプションが指定される。

mt.exe -manifest sub.dll.manifest -outputresource:sub.dll;2

マニフェストおよびマニフェスト埋め込みコマンドについては, **Visual C++ 2015**, **Visual C++ 2017**, **Visual C++ 2019** または **Visual C++ 2022** のドキュメントを参照。

第2章 外部リポジトリ機能

本章では、プログラムや関数、メソッドの呼び出し関係やメソッドの継承関係が適合規則に則っているかを検査するために使用する外部リポジトリ機能について説明する。

〔Enterprise for Linux〕本機能を使用する場合、XML パーサ (Apache Xerces-C++ Version 3.1) のインストールが必要である。

2.1 リポジトリ情報

CALL 文の書き方 1 (あふれ指定) や書き方 2 (例外指定) では、サブプログラムを呼び出す場合、呼び出し側と呼び出される側の引数の記述や引数の渡しかたなどが適合規則に則っているかを、コンパイラは検査しない。

このため、呼び出し側と呼び出される側の引数に矛盾があり、実行時に呼び出しに失敗した場合、呼び出しに成功しても、期待どおりの実行結果とならない場合に、障害を特定するのに時間がかかることがある。

これに対して、CALL 文の書き方 3 (プログラム原型) や利用者定義関数、INVOKE 文は、呼び出し側と呼び出される側の引数の記述や引数の渡し方、返却項目の記述などが適合規則に則っているか、また、OVERRIDE 指定のあるメソッドおよびプロパティメソッドの定義の記述において、継承元との間で適合規則に則っているかを、コンパイラが検査する。これを適合検査と呼ぶ。

この適合検査のためには、引数の記述や引数の渡しかたなどの定義情報が必要である。この情報をリポジトリ情報という。

コンパイラは、リポジトリ情報を翻訳中、クラスや利用者定義関数などの定義が出現するたびにメモリ中に一時的に生成し、翻訳が完了すると、物理ファイルに書き出す。

コンパイラがメモリ中に一時的に生成するリポジトリ情報を内部リポジトリ、物理ファイルに書き出したリポジトリ情報を外部リポジトリと呼ぶ。

コンパイラは、内部リポジトリ情報の生成、参照、更新および削除を次のように行う。

・内部リポジトリ情報の生成・参照・更新・削除

実行内容	実行時期
生成	プログラム原型定義、最外側のプログラム定義、関数原型定義、関数定義、クラス定義 (以降、定義側という) のいずれかの翻訳時に、リポジトリ情報をコンパイラ

	のメモリ内に生成する。
参照	リポジトリ段落でプログラム指定子、関数指定子、クラス指定子（以降、参照側という）を記載している場合に、リポジトリ情報を翻訳時に参照する。
更新	翻訳中、定義側を翻訳するたびにリポジトリ情報を追加更新する。
削除	翻訳終了時にメモリ上からリポジトリ情報を削除する。

コンパイラは、外部リポジトリ情報の生成、参照、更新および削除を次のように行う。

・外部リポジトリ情報の生成・参照・更新・削除

実行内容	実行時期
生成/ 更新	プログラム原型定義、最外側のプログラム定義(※)、関数原型定義、関数定義、クラス定義を翻訳した際、-CG オプション(詳細は後述)を指定しており、かつコンパイラが正常終了した場合に、カレントディレクトリに外部リポジトリ情報を生成する。または存在している外部リポジトリ情報を更新する。 (※)-M(メインプログラム)オプションを指定したプログラム定義を除く
参照	リポジトリ段落にプログラム指定子、関数指定子、クラス指定子を記載しており、内部リポジトリ情報上に該当リポジトリ情報がない場合に外部リポジトリ情報を参照する（コンパイラは参照した外部リポジトリ情報を内部リポジトリ情報に追加した後に内部リポジトリ情報を参照する）。
削除	コンパイラは外部リポジトリ情報を削除しない。外部リポジトリ情報が不要な場合には利用者が明示的に削除する必要がある。

コンパイラは内部リポジトリ情報をコンパイラのメモリ上に生成するため、同一翻訳群中で定義側プログラムと参照側プログラムの両方の翻訳を同時に行わず、別翻訳群でそれぞれのプログラムを翻訳する場合には、内部リポジトリ情報を参照することができない。別翻訳群でプログラムを翻訳する場合は、翻訳時に、外部リポジトリ情報を指定する。

利用者が、定義側のプログラム翻訳時オプションで、外部リポジトリ情報を生成／更新を指定し、参照側のプログラム翻訳時に、外部リポジトリ情報を参照する。これによりコンパイラは、外部リポジトリ情報を介して、異なる翻訳群中で定義したプログラム間の適合検査を実施する。

なお、外部リポジトリを生成、参照、更新および削除する機能を外部リポジトリ機能という。

2.1.1 外部リポジトリの利用方法

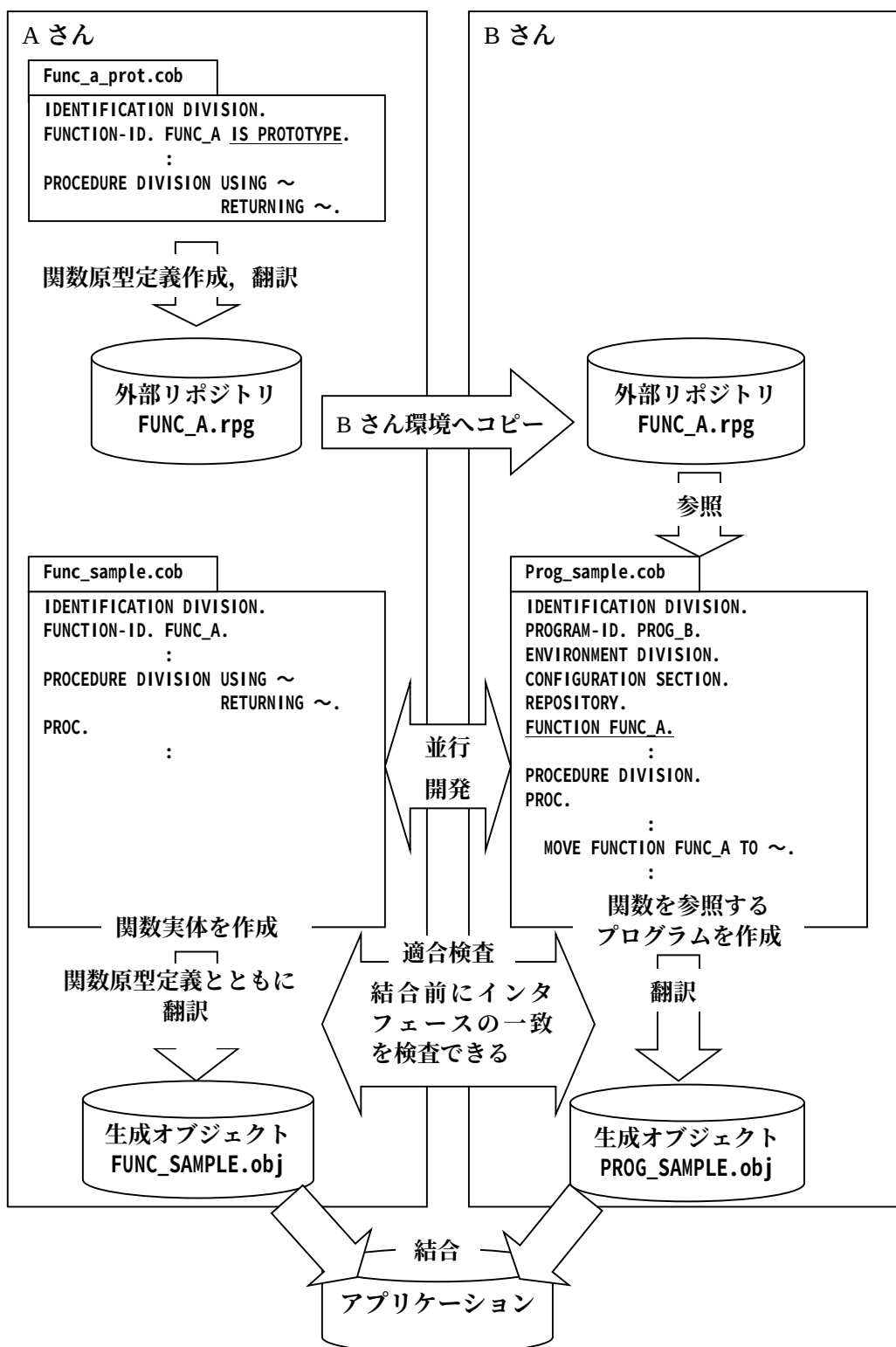
外部リポジトリは、引数の定義側プログラムの翻訳と引数の参照側プログラムの翻訳が異なる翻訳群である場合に使用する。

たとえば、AさんとBさんが共同でプログラム開発を行い、Aさんが関数を作成し、BさんがAさんの作成した関数を参照するプログラムを作成するケースを考える。この場合、関数のインタフェースを取り決めれば、それぞれの開発を進めることができるが、引数や返却項目の数や記述が間違っていないかのインタフェースミスは実際に結合してみるまで分からない。

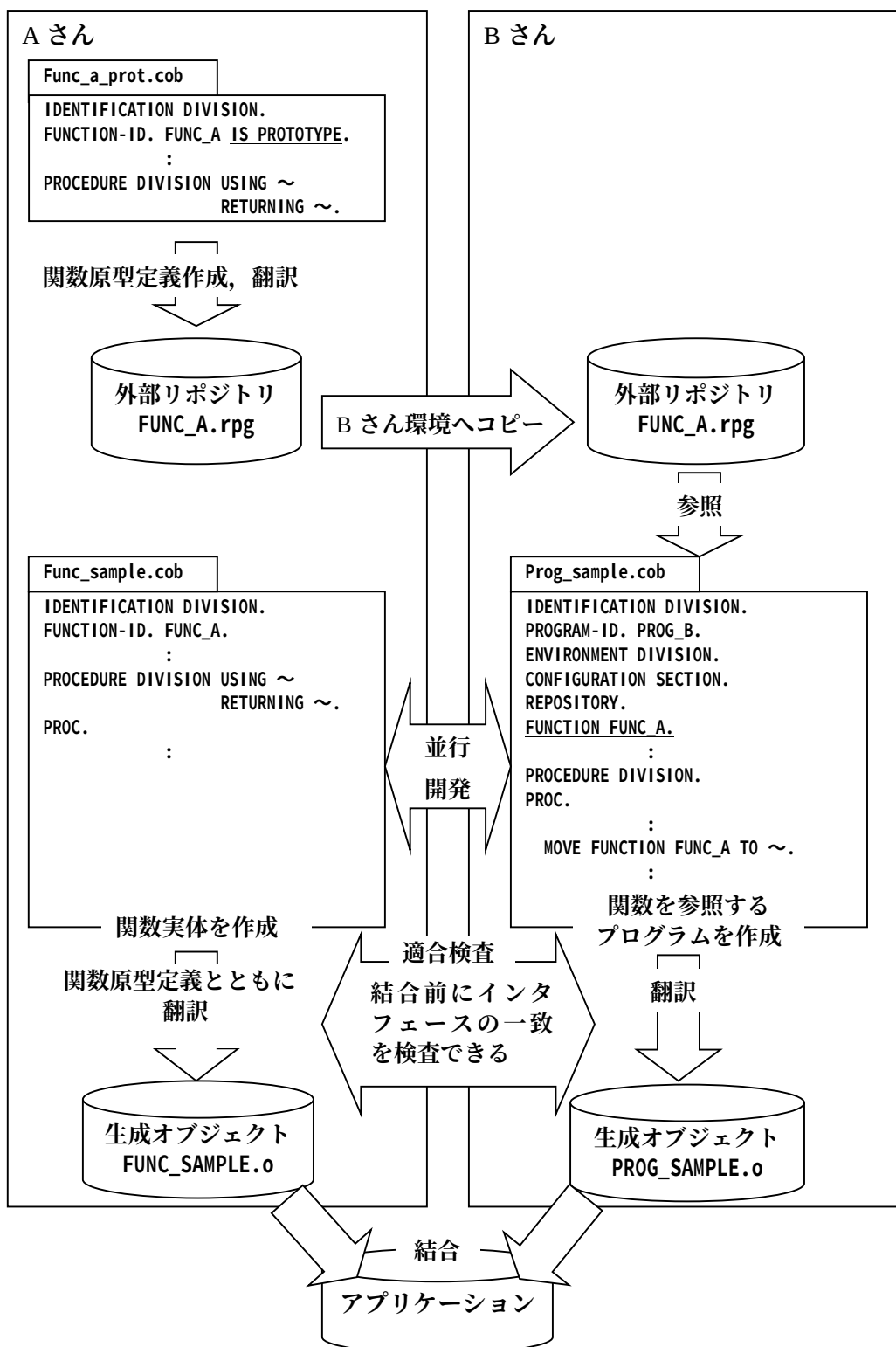
このような場合、Aさんはまず関数原型定義(関数インタフェースのみ)を作成、翻訳し、外部リポジトリを生成する。Aさんはその後関数定義(関数の実体)を作成する。またBさんはAさんが生成した外部リポジトリを参照し、プログラムを作成する。このように、Bさんは、Aさんが関数原型定義作成時に生成した外部リポジトリの情報を使って、翻訳時に適合検査を行うことができるため、プログラムを結合する前にインタフェースミスがないことを確認しながら開発を進めることができる。またAさんは、関数原型定義と関数定義を同一翻訳群で翻訳することによって、関数定義にインタフェースミスがないことを確認しながら開発を進めることができる。

次に利用手順を示す。

[Enterprise for Windows] [Standard]



【Enterprise for Linux】



このように定義側と参照側を別々に翻訳する場合に外部リポジトリは有用である。

Enterprise Edition for Linux から Enterprise Edition for Windows および Standard Edition へ外部リポジトリファイルは移行できない。移行するときは外部リポジトリファイルを生成し直す必要がある。

2.2 外部リポジトリ操作

2.2.1 オプション動作と既定動作

定義側と参照側を同時に(同一翻訳群で)翻訳する場合は、内部リポジトリ情報により、適合検査を行うため、外部リポジトリは必ずしも必要ではない。また、既に外部リポジトリを生成している場合、インタフェースに変更がなければ、定義側を翻訳するたびに外部リポジトリを生成、更新する必要はない。

意図しないインタフェースの変更を伴うソース修正を行い、外部リポジトリを生成、更新すると、そのインタフェースを参照している他のプログラムとの整合がとれなくなり、プログラムの翻訳時に適合検査のエラーが発生する。

このために、外部リポジトリの生成、更新オプション(-CG オプション)を使用する。利用者は、外部リポジトリを生成、更新する必要がある場合のみ-CG オプションを指定する必要がある。

また、外部リポジトリの生成、更新オプションを指定しない場合、利用者がインタフェースの変更を伴うソース修正を行うと、そのインタフェースを参照する他のプログラムとの整合がとれなくなり、リンク時にエラーが発生したり、実行時に動作不正が起こったりする。

このため、外部リポジトリの生成、更新オプションを指定しない場合に、インタフェースの変更があるかどうかを検査するオプション(-CH オプション)を使用する。

-CH オプションを指定すると、翻訳時にカレントディレクトリに外部リポジトリが存在している場合、内部リポジトリに生成したリポジトリ情報と矛盾があるかどうかをコンパイラが検査する。

検査の結果、インタフェースに矛盾が判明した場合、利用者は、インタフェースを修正し、インタフェースの変更を反映するため-CG オプションを指定し、再度翻訳を行う。

複数のメンバで開発をしており、他のメンバの開発した関数やクラスを参照する場合や、別のプロジェクトで既に開発の終わっている関数やクラスを参照するプログラムを作成する場合、参照する外部リポジトリが別々のディレクトリに存在する可能性がある。

このような場合、-R オプションで、参照する外部リポジトリが存在するディレクトリを指定することができる。

複数のメンバで開発をしており、開発中のクラスを他のメンバが参照するような場合、クラスの開発メンバが出力する外部リポジトリを他のメンバが直接参照することは推奨しない。なぜなら、開発

中のソースはいつ、どのような時点でクラスの開発メンバがインタフェースに影響のある修正を入れてしまうか判らないためである。そのような不安定な状態の外部リポジトリを参照すると、クラスの開発メンバがインタフェースを誤っているのか、クラスを参照している他のメンバのプログラムがインタフェースを誤っているのか良く判らない状態に陥ることになる。

このような状態を避けるため、クラスを参照している他のメンバは、クラスの開発メンバが生成する外部リポジトリを直接参照するのではなく、別のディレクトリへコピーした外部リポジトリを参照するように、オプション指定するのがよい。クラスの開発メンバはたとえば、クラスのメソッド定義を作成する度に評価を行い、問題がないことが確認できた外部リポジトリを、他のメンバが参照しているディレクトリへコピーし、通知するようにする。こうすることで、クラスの開発メンバは他のメンバに気兼ねすることなく、クラスの修正を行うことができる。また、クラスを参照する他のメンバもクラスの開発メンバの保証した外部リポジトリを使って開発を進めることができる。

外部リポジトリに関連するオプションの指定有無による動作の違いを次に記載する。

オプション	名前	指定時の動作	指定なし時の動作
-CG	生成／更新指定オプション	カレントディレクトリに外部リポジトリを生成する。既に外部リポジトリが存在する場合には、いったん削除した上で生成する。 なお、翻訳が正常終了した場合にのみ生成し、翻訳エラー等がある場合には生成しない。	外部リポジトリの生成／更新は行なわない。
-CH	矛盾チェックオプション	翻訳時に、一時的にコンパイラメモリ中に生成する内部リポジトリ情報とカレントディレクトリの外部リポジトリ情報に矛盾があれば警告メッセージを出力する。また、内部リポジトリ情報を正しいとみなして翻訳処理を続行する。 矛盾の有無に関わらず、外部リポジトリの生成／更新は行なわない。 なお、本オプションを-CG オプションと同時に指定することはできない。	翻訳時に、一時的にコンパイラメモリ中に生成する内部リポジトリ情報とカレントディレクトリの外部リポジトリ情報に矛盾があるかを検査しない。内部リポジトリ情報を正しいとみなして翻訳処理を続行する。
-Rdir ※1	外部リポジトリ検索ディレクトリ指定オプション	本オプションで指定したディレクトリを、参照する外部リポジトリの検索ディレクトリに加える。※2 リポジトリ情報の検索順は次のようになる。 ①一時的にコンパイラメモリ中に生成する	リポジトリ情報を次の順で検索する。 ①一時的にコンパイラメモリ中に生成する内部リポジトリ情報

		内部リポジトリ情報 ※3 ②カレントディレクトリの外部リポジトリ ③本オプションにて指定したディレクトリの外部リポジトリ(本オプションを複数指定した場合は、指定した順に検索する)	②カレントディレクトリの外部リポジトリ

※1 *dir*は検索するディレクトリ名を表す。

※2 本オプションは、リポジトリ段落でプログラム指定子、関数指定子、クラス指定子を指定した場合に、参照する外部リポジトリの検索ディレクトリを追加するオプションである。プログラム原型定義、最外側のプログラム定義、関数原型定義、関数定義、クラス定義の翻訳時に生成する外部リポジトリの出力ディレクトリはカレントディレクトリであり、このオプションには関係しない。

※3 一時的にコンパイラメモリ中に生成する内部リポジトリ情報は、翻訳中、次の2種類が存在する。

1. 同じ翻訳群中で先に翻訳した翻訳単位(プログラム原型定義、最外側のプログラム定義、関数原型定義、関数定義、クラス定義)から生成したリポジトリ情報。
2. 同じ翻訳群中で先に翻訳した翻訳単位のリポジトリ段落で参照したプログラム指定子、関数指定子、クラス指定子に対するリポジトリ情報。

コンパイラは、後者(2.)のリポジトリ情報が、内部リポジトリ情報中になれば、外部リポジトリから読み込み、内部リポジトリに追加する。

2.2.2 利用上の注意

コンパイラは、翻訳時、リポジトリの参照が必要となった時点で、まず内部リポジトリを参照する。内部リポジトリに情報がなければ、外部リポジトリを参照する。そして、その参照した外部リポジトリ情報を内部リポジトリに追加し再び翻訳を続行する。

このような動作をするため、正しく適合検査をするためにはいくつか注意が必要となる。

(1) 定義と参照の翻訳順序

同じ翻訳群中で、定義と参照の両方がある場合、参照よりも定義を先に翻訳するようになければならない。たとえば、ある翻訳群中でクラス A の定義とクラス A を参照するプログラム B の定義を翻訳する場合、クラス A の定義、プログラム B の定義の順に翻訳するようになければならない。(※ 1)

これは、クラス A の定義を翻訳することによって、クラス A のリポジトリ情報を生成／更新するため、先にプログラム B を翻訳すると、次のような状況が発生するためである。

- ・ クラス A の外部リポジトリが存在している場合

コンパイラは、プログラム B の翻訳時に、外部リポジトリからクラス A のリポジトリ情報を取得し適合検査を行う。そのあと、クラス A の定義の翻訳を行い、クラス A のリポジトリ情報を生成する。生成したクラス A のリポジトリ情報が、プログラム B の翻訳時に参照した外部リポジトリの情報と異なっても、既にプログラム B の翻訳が終了してしまっているため、遡って適合検査を行うことができない。

- ・ クラス A の外部リポジトリが存在しない場合

コンパイラは、プログラム B の翻訳時に、参照すべきクラス A のリポジトリ情報が存在しないため、翻訳エラーを出す。そのあと、クラス A の定義の翻訳を行い、クラス A のリポジトリ情報を生成するが、既にプログラム B の翻訳が終了してしまっているため、遡って適合検査を行うことができない。

このような状況が発生するため、コンパイラは定義と参照の翻訳順序が逆転している場合には、警告メッセージを表示する。

※1 翻訳の順序は、翻訳コマンドに指定した COBOL ソースの指定順になる。また、COBOL ソース中に複数の翻訳単位を記載している場合は、翻訳単位の記載順になる。

次に、定義と参照の翻訳順序と、正しく適合検査が行えるかどうかについて、具体例を示す。

例の中で ○は正しく適合検査が行える ×は正しく適合検査が行えない ことを表す。

例1 別々の COBOL ソースに定義と参照がある場合

Sample_A.cob

```
IDENTIFICATION DIVISION.
CLASS-ID. CLS_A.
:
END CLASS CLS_A.
```

定義

Sample_B.cob

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PROG_B.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS CLS_A.
:
END PROGRAM PROG_B.
```

参照

- cob -M Sample_A.cob Sample_B.cob 定義→参照の順なので正しく適合検査が行える
- × cob -M Sample_B.cob Sample_A.cob 参照→定義の順なので正しく適合検査が行えない

例2 1つの COBOL ソース中に定義と参照がある場合

Sample_X.cob

```
IDENTIFICATION DIVISION.
CLASS-ID. CLS_A.
:
END CLASS CLS_A.

IDENTIFICATION DIVISION.
PROGRAM-ID. PROG_B.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS CLS_A.
:
END PROGRAM PROG_B.
```

定義

参照

Sample_Y.cob

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PROG_B.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS CLS_A.
:
END PROGRAM PROG_B.

IDENTIFICATION DIVISION.
CLASS-ID. CLS_A.
:
END CLASS CLS_A.
```

参照

定義

- cob -M Sample_X.cob 定義→参照の順なので正しく適合検査が行える
- × cob -M Sample_Y.cob 参照→定義の順なので正しく適合検査が行えない

(2) -M オプション(メインプログラム)指定のプログラム定義の翻訳

-M オプションは前述の翻訳の順序に従って、最初に翻訳するプログラム定義をメインプログラムとして扱うというオプションである。

1つの COBOL ソース中に複数のプログラム定義を記載しており、そのプログラム定義間に参照関

係がある場合に「(1) 定義と参照の翻訳順序」に示した問題を避けるためには、定義側のプログラム(サブプログラム)を先に翻訳しなければならない。しかし、-M オプションを指定した場合には最初に翻訳するプログラム定義をメインプログラムとして扱ってしまうので矛盾が生じてしまう。

たとえば、COBOL ソース `Sample_Z.cob` 中にメインプログラム(MAINPROG)とサブプログラム(SUBPROG)の 2 つの翻訳単位を含んでいるとする。

MAINPROG は-M オプションの対象(メインプログラム)とするために、COBOL ソース中で最初の翻訳単位として記載する必要がある。

また、MAINPROG は、SUBPROG を参照するために、リポジトリ段落で SUBPROG を指定する必要がある。

MAINPROG のリポジトリ段落で SUBPROG を参照することを指定すると、MAINPROG の翻訳時に SUBPROG のリポジトリ情報が必要となるので、SUBPROG は、MAINPROG よりも先に翻訳する必要がある。

そうすると COBOL ソース中で、MAINPROG よりも先に SUBPROG を記載する必要があるが出てくるが、MAINPROG より先に記載してしまうと、SUBPROG が-M オプションの対象(メインプログラム)となってしまうため、うまく解決できない。

次に具体例を示す。

例の中で ○は正しく翻訳が行える ×は正しく翻訳が行えない ことを表す。

例 3 1 つの COBOL ソース中にメインプログラムとサブプログラムがある場合

3-1 メインプログラム, サブプログラムの順で記載している場合

Sample_Z.cob

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MAINPROG.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
REPOSITORY.  
PROGRAM SUBPROG.  
:  
END PROGRAM MAINPROG.  
  
IDENTIFICATION DIVISION.  
PROGRAM-ID. SUBPROG.  
:  
END PROGRAM SUBPROG.
```

cob -M Sample_Z.cob

- × メインプログラム(MAINPROG)の翻訳時に, サブプログラムのリポジトリ情報がないため, 翻訳できない

例 3 1 つの COBOL ソース中にメインプログラムとサブプログラムがある場合

3-2 サブプログラム, メインプログラムの順で記載している場合

Sample_Z.cob

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SUBPROG.  
:  
END PROGRAM SUBPROG.
```

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MAINPROG.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
REPOSITORY.  
PROGRAM SUBPROG.  
:  
END PROGRAM MAINPROG.
```

cob -M Sample_Z.cob

× サブプログラム(SUBPROG)をメインプログラムと認識するため(先頭のプログラム定義であるため), 正しく翻訳できない

このような場合は, 次のいずれかの方法で解決できる。

(a) メインプログラムとサブプログラムを別 COBOL ソースに分離する

メインプログラムを Sample_Z1.cob, サブプログラムを Sample_Z2.cob に分離し, 翻訳処理を 2 つに分ける。そして, まず, サブプログラムを翻訳し, サブプログラムの外部リポジトリ情報を生成する。次に, 生成したサブプログラムの外部リポジトリ情報を参照してメインプログラムを翻訳する。こうすることで, 意図したとおりに翻訳できる。

【Enterprise for Windows】【Standard】

例 3 1 つの COBOL ソース中にメインプログラムとサブプログラムがある場合
 3-3 メインプログラムとサブプログラムを別 COBOL ソースに分離する

Sample_Z1.cob

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MAINPROG.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
REPOSITORY.  
PROGRAM SUBPROG.  
:  
END PROGRAM MAINPROG.
```

Sample_Z2.cob

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SUBPROG.  
:  
END PROGRAM SUBPROG.
```

cob -c -CG Sample_Z2.cob

cob -M Sample_Z1.cob Sample_Z2.obj

- メインプログラムの翻訳時、参照するサブプログラムのリ
 ポジトリ情報が生成済みのため、正しく翻訳できる

【Enterprise for Linux】

例 3 1 つの COBOL ソース中にメインプログラムとサブプログラムがある場合
 3-3 メインプログラムとサブプログラムを別 COBOL ソースに分離する

Sample_Z1.cob

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MAINPROG.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
REPOSITORY.  
PROGRAM SUBPROG.  
:  
END PROGRAM MAINPROG.
```

Sample_Z2.cob

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SUBPROG.  
:  
END PROGRAM SUBPROG.
```

cob -c -CG Sample_Z2.cob

cob -M Sample_Z1.cob Sample_Z2.o

- メインプログラムの翻訳時、参照するサブプログラムのリ
 ポジトリ情報が生成済みのため、正しく翻訳できる

(b) サブプログラムの原型定義を追加する

COBOL ソースの先頭にサブプログラムの原型定義を追加し、サブプログラムの原型定義、メインプログラムの定義、サブプログラムの定義の順に記載する。

こうすると、コンパイラは、プログラム原型定義の翻訳によって、サブプログラムのリポジトリ情報を生成した後、メインプログラムの定義の翻訳で、生成したサブプログラムのリポジトリ情報を参照することになるため、意図したとおりに翻訳できる。

例 3 1 つの COBOL ソース中にメインプログラムとサブプログラムがある場合

3-4 サブプログラムの原型定義を追加する場合

Sample_Z.cob
IDENTIFICATION DIVISION. PROGRAM-ID. SUBPROG IS PROTOTYPE. : END PROGRAM SUBPROG.
IDENTIFICATION DIVISION. PROGRAM-ID. MAINPROG. ENVIRONMENT DIVISION. CONFIGURATION SECTION. REPOSITORY. PROGRAM SUBPROG. : END PROGRAM MAINPROG.
IDENTIFICATION DIVISION. PROGRAM-ID. SUBPROG. : END PROGRAM SUBPROG.

cob -M Sample_Z.cob

- プログラム原型定義によって、先にサブプログラムのリポジトリ情報を生成する。プログラム原型定義は-M オプションの対象ではないため、正しく翻訳できる

(c) 適合検査を行わない(リポジトリ段落でサブプログラムの参照をしない)

リポジトリ段落でサブプログラムの参照を指定しない場合 (CALL 文の書き方 1 や書き方 2 を使用する場合)、コンパイラは適合検査を行わない。

ただし、翻訳時に適合検査をしないと、実行時に不整合が発生する可能性があるため、このような方法は推奨しない。

2.3 外部リポジトリのファイル名生成規則

外部リポジトリの実体は1つの物理ファイルではなく、コンパイラは、定義したクラス、関数、プログラムごとに1つずつ独立した物理ファイルとして生成する。

コンパイラが外部リポジトリを生成／参照するときのファイル名は次の規則によって決定する。

“ 種別を表すキーワード ” + “ _ ” + “ 外部公開名 ” + “ .rpg ”

“ 種別を表すキーワード ” は、次の種類がある。

- ・ 定義側：見出し部に記載した翻訳単位の種別

翻訳単位の種別	“ 種別を表すキーワード ”
クラス定義	CLS
関数原型定義／関数定義	FUNC
プログラム原型定義／プログラム定義	PROG

- ・ 参照側：リポジトリ段落に記載した指定子の種別

指定子の種別	“ 種別を表すキーワード ”
クラス指定子	CLS
関数指定子	FUNC
プログラム指定子	PROG

“ 外部公開名 ” は、次の種類がある。

COBOL ソース上の外部公開名の文字種別	“ 外部公開名 ”
英小文字	“ 英小文字 ” + “ ^ ” (例 abc → a^b^c^)
上記以外(*1)	指定したまま

(*1) ただし日本語の外部公開名を除く

COBOL ソース上の外部公開名は、次のように決定する。

- ・ 定義側：見出し部に記載した翻訳単位のソース単位名(AS 指定がある場合は、AS の定数)
次表の下線箇所が外部公開名となる。

翻訳単位の種別	COBOL ソース記述 (IDENTIFICATION DIVISION)

クラス定義	CLASS-ID. <u>クラス名</u> .
	CLASS-ID. クラス名 AS <u>定数</u> .
関数原型定義	FUNCTION-ID. <u>関数原型名</u> IS PROTOTYPE.
	FUNCTION-ID. 関数原型名 AS <u>定数</u> IS PROTOTYPE.
関数定義	FUNCTION-ID. <u>関数名</u> .
	FUNCTION-ID. 関数名 AS <u>定数</u> .
プログラム原型定義	PROGRAM-ID. <u>プログラム原型名</u> IS PROTOTYPE.
	PROGRAM-ID. プログラム原型名 AS <u>定数</u> IS PROTOTYPE.
プログラム定義	PROGRAM-ID. <u>プログラム名</u> .
	PROGRAM-ID. プログラム名 AS <u>定数</u> .

※AS 定数指定時, 定数は大文字／小文字を区別して外部公開名として扱う。-Cs オプションの有無は影響しない。

※AS 定数指定がない時, 外部公開名はすべて大文字となる。ただし, -Cs オプション指定時, プログラム定義の外部公開名の大文字／小文字は区別する。

- ・ 参照側:リポジトリ段落に記載した指定子の名前(AS 指定がある場合は, AS の定数)
次表の下線箇所が外部公開名となる。

指定子の種別	COBOL ソース記述 (ENVIRONMENT DIVISION の REPOSITORY 段落)
クラス指定子	CLASS <u>クラス名</u> .
	CLASS クラス名 AS <u>定数</u> .
関数指定子	FUNCTION <u>関数原型名</u> .
	FUNCTION 関数原型名 AS <u>定数</u> .
プログラム指定子	PROGRAM <u>プログラム原型名</u> .
	PROGRAM プログラム原型名 AS <u>定数</u> .

※AS 定数指定時, 定数は大文字／小文字を区別して外部公開名として扱う。-Cs オプションの有無は影響しない。

※AS 定数指定がない時, 外部公開名はすべて大文字となる。-Cs オプション指定の有無は影響しない。

※クラス定義の場合, 外部リポジトリとして,

“種別を表すキーワード” + “_” + “外部公開名” + “.rpg” に加え,

“種別を表すキーワード” + “_” + “外部公開名” + “.rph” を生成する。

クラス定義の外部リポジトリを参照する場合, これら 2 つのファイルは同じディレクトリに配置しておく必要がある。

次に、外部リポジトリのファイル名の決定規則の例を示す。

例 4 外部リポジトリのファイル名

4-1 翻訳単位から生成する外部リポジトリ

4-1-1 AS 指定なし、-Cs オプションなし

Sample4-1-1.cob
IDENTIFICATION DIVISION. CLASS-ID. <u>ClassA</u> . ① ② : END CLASS ClassA.

cob -c -CG Sample4-1-1.cob

① 翻訳単位の種別：クラス定義

→種別を表すキーワード：CLS

② 翻訳単位のソース単位名：ClassA

→外部公開名：CLASSA

⇒外部リポジトリのファイル名：CLS_CLASSA.rpg

例 4 外部リポジトリのファイル名

4-1 翻訳単位から生成する外部リポジトリ

4-1-2 AS 指定なし、-Cs オプションあり、プログラム定義以外

Sample4-1-2.cob
IDENTIFICATION DIVISION. CLASS-ID. <u>ClassA</u> . ① ② : END CLASS ClassA.

cob -c -CG -Cs Sample4-1-2.cob

① 翻訳単位の種別：クラス定義

→種別を表すキーワード：CLS

② 翻訳単位のソース単位名：ClassA

→外部公開名：CLASSA

⇒外部リポジトリのファイル名：CLS_CLASSA.rpg

※プログラム定義以外は-Cs オプションの有無に拘わらず、外部公開名は大文字

例 4 外部リポジトリのファイル名**4-1 翻訳単位から生成する外部リポジトリ****4-1-3 AS 指定なし, -Cs オプションあり, プログラム定義**

Sample4-1-3.cob
IDENTIFICATION DIVISION. PROGRAM-ID. ProgA. ① ② : END PROGRAM ProgA.

cob -c -CG -Cs Sample4-1-3.cob

①翻訳単位の種別：プログラム定義

→種別を表すキーワード：PROG

②翻訳単位のソース単位名：ProgA

→外部公開名：Pr[^]o[^]g[^]A

⇒外部リポジトリのファイル名：PROG_Pr[^]o[^]g[^]A.rpg

※プログラム定義は-Cs オプション指定時、外部公開名の大文字／小文字を区別

例 4 外部リポジトリのファイル名**4-1 翻訳単位から生成する外部リポジトリ****4-1-4 AS 指定あり**

Sample4-1-4.cob
IDENTIFICATION DIVISION. CLASS-ID. ClassA AS "ClassB". ① ② : END CLASS ClassA.

cob -c -CG Sample4-1-4.cob

①翻訳単位の種別：クラス定義

→種別を表すキーワード：CLS

②翻訳単位のソース単位名：ClassB

→外部公開名：Cl[^]a[^]s[^]s[^]B

⇒外部リポジトリのファイル名：CLS_Cl[^]a[^]s[^]s[^]B.rpg

※AS 定数は-Cs オプションの有無に拘わらず、外部公開名の大文字／小文字を区別

例 4 外部リポジトリのファイル名

4-2 リポジトリ段落で参照する外部リポジトリ

4-2-1. AS 指定なし, -Cs オプションなし

Sample4-2.cob

```

IDENTIFICATION DIVISION.
PROGRAM-ID. ProgA.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS  ClassA.
  ①      ②
      :
END PROGRAM ProgA.

```

cob -M Sample4-2.cob

①指定子の種別：クラス指定子

→種別を表すキーワード：CLS

②指定子の名前：ClassA

→外部公開名：CLASSA

⇒外部リポジトリのファイル名：CLS_CLASSA.rpg

例 4 外部リポジトリのファイル名

4-2 リポジトリ段落で参照する外部リポジトリ

4-2-2. AS 指定なし, -Cs オプションあり

Sample4-2.cob

```

IDENTIFICATION DIVISION.
PROGRAM-ID. ProgA.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS  ClassA.
  ①      ②
      :
END PROGRAM ProgA.

```

cob -M -Cs Sample4-2.cob

①指定子の種別：クラス指定子

→種別を表すキーワード：CLS

②指定子の名前：ClassA

→外部公開名：CLASSA

⇒外部リポジトリのファイル名：CLS_CLASSA.rpg

※-Cs オプションの有無に拘わらず、リポジトリ段落で参照する外部公開名は大文字

例4 外部リポジトリのファイル名

4-2 リポジトリ段落で参照する外部リポジトリ

4-2-3. AS 指定あり、プログラム指定子

Sample4-2-3.cob

```
IDENTIFICATION DIVISION.
PROGRAM-ID. ProgB.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
PROGRAM ① ProgA AS "ProgA" ②.
      :
END PROGRAM ProgB.
```

cob -M Sample4-2-3.cob

①指定子の種別：プログラム指定子

→種別を表すキーワード：PROG

②指定子の名前：ProgA

→外部公開名：Pr[^]o[^]g[^]A

⇒外部リポジトリのファイル名：PROG_Pr[^]o[^]g[^]A.rpg

※AS 定数は-Cs オプションの有無に拘わらず、リポジトリ段落で参照する外部公開名の大文字／小文字を区別

(プログラム定義では、AS 指定をしなくても、-Cs オプションにより外部公開名の大文字／小文字を区別できるが、リポジトリ段落で参照する場合は、外部公開名の大文字／小文字を区別するためには、AS 定数が必要)

例 4 外部リポジトリのファイル名**4-2 リポジトリ段落で参照する外部リポジトリ****4-2-4. AS 指定あり、プログラム指定子以外**

Sample4-2-4.cob

```

IDENTIFICATION DIVISION.
PROGRAM-ID. ProgA.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS ClassA AS "ClassB".
①          :          ②
END PROGRAM ProgA.

```

cob -M Sample4-2-4.cob

①指定子の種別：クラス指定子

→種別を表すキーワード：CLS

②指定子の名前：ClassB

→外部公開名：CL^a^s^s^B

⇒外部リポジトリのファイル名：CLS_CL^a^s^s^B.rpg

※AS 定数は-Cs オプションの有無に拘わらず、リポジトリ段落で参照する外部公開名の大文字／小文字を区別

2.4 適合検査について

コンパイラは、CALL 文の書き方 1 または書き方 2 の適合検査を行わない。このため、利用者は、呼び出し側と呼び出される側の引数の記述や引数の渡し方に不整合がないか、注意を払う必要がある。

これらのプログラムの適合検査を行うためには、CALL 文の書き方 3 に書き換える必要があるが、逆に、適合検査を行うと、適合の規則により、エラーを検出するため、そのエラー修正に大幅なプログラム修正を伴う場合がある。この場合、正しく修正することが望ましいが、実績のある既存プログラムであれば、適合検査を行わないで(CALL 文の書き方 1 または書き方 2 のままにして)運用するという選択肢もある。

2.5 既存オプションの影響

既存オプションの中で外部リポジトリに影響するものについて以降に記述する。

オプション	影響	備考
-M	メインプログラムとして指定したプログラム定義に対して -CG(外部リポジトリの生成／更新)や-CH(矛盾チェック)は無効となる	
-Ca	外部リポジトリに出力するデータ項目の大きさに影響する	適合検査時には、データ項目の大きさ、割付位置の違いに注意が必要
-CN		
[Enterprise]		
-Fa		
[Enterprise]		
-Fb		
[Enterprise for Windows]	外部リポジトリに出力するデータ項目の大きさ、内部表現形式に影響する	適合検査時には、データ項目の大きさ、割付位置の違い、内部表現形式の違いに注意が必要
-CM		
-C5		
[Enterprise]		
-Fc		

第3章 ファイル入出力機能

3.1 ファイルの装置およびファイル指定

ファイル管理記述項の **ASSIGN** 句によって、ファイルの装置およびファイルを指定する。形式は、次のとおりである。

<u>FILE-CONTROL.</u>	
<u>SELECT</u> ファイル名 1 <u>ASSIGN</u>	T0 内部ファイル識別名 1 T0 定数 1 <u>USING</u> {データ名 1} ...
<u>FILE SECTION.</u>	
<u>FD</u> ファイル名 1	
<u>LABEL RECORD IS STANDARD</u>	
<u>VALUE OF IDENTIFICATION</u>	データ名 2 定数 2

- 内部ファイル識別名 1 の指定
次の形式で指定する。

指定形式	識別名 /装置名	意味	備考
装置名-データ名 3	DK	大容量記憶ファイル	
	PR	プリンタファイル	
内部ファイル名-装置名	MSD FDU	大容量記憶ファイル	VALUE OF 句必須
	PRN	プリンタファイル	[Enterprise] VALUE OF 句必須
[Standard] 内部ファイル名-装置名[-フォ ーム名]	PRN	プリンタファイル	
[Standard]	PRT	プリンタファイル	

内部ファイル名-装置名[-メディア識別名] [{データ名 5 定数 3}]			

データ名 3 には、実ファイル名を格納するデータ名を指定する。

フォーム名には、フォームオブジェクトファイル名を指定する。

メディア識別名には、プリンタ装置識別名を指定する。

データ名 5 および定数 3 には、フォームオブジェクトファイル名を指定する。

- 定数 1 の指定

英数字定数は次の形式で指定する。

指定形式	識別名 /装置名	意味	備考
“ 装置名-実ファイル名 ”	DK	大容量記憶ファイル	
	PR	プリンタファイル	
“ 実ファイル名-識別名 ”	UFS	大容量記憶ファイル	
“ 内部ファイル名-装置名 ”	MSD FDU	大容量記憶ファイル	VALUE OF 句必須
[Enterprise] “ 実ファイル名-装置名 ”	PRN	プリンタファイル	
[Standard] “ 内部ファイル名-装置名[-フ ォーム名] ”	PRN	プリンタファイル	

実ファイル名には、デバイスタイプ上のファイル名を指定する。

(オペレーティングシステムで許されるパス名を含むファイル名が指定可能である。ただし、パス名やファイル名にハイフンを含む名前は指定できない。)

フォーム名には、フォームオブジェクトファイル名を指定する。

例

[Enterprise for Windows] [Standard]

“ DK-C:¥tmp¥yyy ”

“ C:¥tmp¥yyy-UFS ”

[Enterprise for Linux]

“ DK-/tmp/yyy ”

“ /tmp/yyy-UFS ”

- USING データ名 1 の指定

次の形式で指定する。

指定形式	識別名 /装置名	意味	備考
データ名 6-識別名	UFS	大容量記憶ファイル	
データ名 6-装置名	PRN	プリンタファイル	

データ名 6 は、デバイスタイプ上の実ファイル名を指定する。

[Standard] 上記の指定形式の指定方法および説明を以下に示す。

1. 指定方法

- (a) LABEL RECORD 句が STANDARD 指定のときは、必ず VALUE OF 句によって内部ファイル ID または実ファイル名をデータ名または定数で指定する。ファイルの種類と指定する内部ファイル ID または実ファイル名については、次の表に示す。

ファイルの種類	指定する内部ファイル ID または実ファイル名
大容量記憶ファイル	指定した内容が COBOL ファイルマッパーで関連付けられている場合、内部ファイル ID となり、COBOL ファイルマッパーの指定に従い、実ファイル名やファイル種別等が決定される。 関連付けられていない場合、実ファイル名となり、ドライブおよびディレクトリを含めたファイルの名称 (カレントディレクトリの場合はファイル名のみ) を指定する。
プリンタファイル	ドライブおよびディレクトリを含めたファイルの名称 (プリンタ出力データをファイルに出力する)

[COBOL 独自ファイル]

- 実ファイル名は次のように指定する。
[ドライブ名] [パス名] ファイル名
- 実ファイル名の長さは 255 桁を超えてはならない。

[COBOL 独自ファイル以外]

- COBOL 独自ファイル以外のファイル種別 (IFAS FILE など) の場合、そのファイル種別を使用するファイルエンジン側の説明書を参照。

- (b) 内部ファイル名は、先頭が英字で 8 文字以内の英字、数字または下線で指定する。また、内部ファイル名は、実行可能ファイルの中で一意でなければならない。
- (c) 装置名には、ファイルを割り当てる周辺装置を指定する。周辺装置は次の表に示す 3 桁の

デバイスタイプと、3 桁の数字のデバイスアドレスを指定する。デバイスアドレスは省略可能である。

装 置	デバイスタイプ	デバイスアドレス
大容量記憶ファイル プリンタファイル	MSD/FDU PRN/PRT	000 ～ 009, 990 ～ 998 PRN のみ指定可能(下記参照)

- i. デバイスタイプに MSD または FDU が指定された場合、大容量記憶ファイルが指定されたものとみなされる。
VALUE OF 句で指定された内容が COBOL ファイルマッパーで関係付けられている場合、内部ファイル ID となり、COBOL ファイルマッパーの指定に従い、実ファイル名やファイル種別等が決定される。
VALUE OF 句で指定された内容が COBOL ファイルマッパーで関連付けられていない場合、実ファイル名となり、実ファイル名のドライブ名で指定されたディスクドライブが割り当てられる(ドライブ名が省略された場合はカレントドライブが割り当てられる)。
- ii. デバイスタイプに PRN が指定された場合、プリンタファイルが指定されたものとみなされ、フォーム名を指定できる。ただし、LABEL RECORD 句 が STANDARD 指定されると、実ファイル名のドライブ名で指定されたディスクドライブが割り当てられる。
- iii. デバイスタイプに PRT が指定された場合、プリンタファイルが指定されたものとみなされ、プリンタ装置識別名とフォーム名を指定できる。ただし、LABEL RECORD 句が STANDARD 指定されると、実ファイル名のドライブ名で指定されたディスクドライブが割り当てられる。
- iv. 大容量記憶ファイルのデバイスアドレスは、COBOL ファイルマッパーで関連付けられている場合に有効となる。COBOL ファイルマッパーで関連付けられていない場合はメモ扱いである。

関連付けされている場合、以下の検索順序で関連付けられている情報を検索する。

デバイスアドレス	検索順序
MSDnnn / FDUnnn	(1) MSD/FDUnnn として関連付けされている情報
	(2) 指定なし として関連付けされている情報
MSD999 / FDU999	(1) 指定なし として関連付けされている情報
	(2) MSD/FDU990 ～ 998 として関連付けされている情報を順次
省略	(1) 指定なし として関連付けされている情報
	(2) MSD/FDU000 ～ 009 として関連付けされている情報を順次

(3) MSD/FDU990 ～ 998 として関連付けされている情報を順次

- v. プリンタ(PRN)のデバイスアドレスは、アプリケーション実行環境を併用する場合にのみ有効となる。

指定する値などは、『COBOL アプリケーション実行環境 プリンタ機能 説明書』を参照。

フォーム名は、装置指定としてプリンタを指定した場合に、フォームオーバーレイ機能で使用するフォームを指定する。フォーム名は、英字または数字からなる 8 文字以内で指定する。

メディア識別名は、英字または数字からなる 16 文字以内で、プリンタ装置識別名を指定する。

データ名 5、定数 3 は、123 桁以内の英数字項目または英数字定数で、フォーム名を指定する。

次に、各形式について説明を行う。

① 内部ファイル名-装置名

この形式は、すべてのファイルに対し指定できる形式である。

② 内部ファイル名-装置名 [-フォーム名]

この形式は、デバイスタイプが PRN のプリンタファイルに対して指定できる形式である。

③ 内部ファイル名-装置名 [-メディア識別名] [フォーム名]

この形式は、デバイスタイプが PRT のプリンタファイルに対して指定できる形式である。

動的にフォーム名を変更したい場合、フォーム名にデータ名 5 を指定する。

例(1)

COBOL ファイルマッパーで、内部ファイル ID(CODEDATA)として、C ドライブの¥DATA ディレクトリにある S-codedata という名前の、IFAS 順ファイルをアプリケーションで操作する。

COBOL ファイルマッパーの例題



内部ファイルID	デバイス	ファイル種別	実ファイル名	注釈	変更
CODEDATA		IFAS FILE	C:\¥DATA¥S-codedata	マスタファイル	
MASTFILE		COBOL	C:\¥DATA¥Gyoumu-mast	マスタファイル	

```
FILE-CONTROL.
    SELECT FILE1 ASSIGN TO      INF1-MSD.
:
FD FILE1
    LABEL RECORD IS      STANDARD
    VALUE OF      IDENTIFICATION IS      "CODEDATA".
```

例(2)

Bドライブのルートディレクトリにある **S-codedata** という名前のファイルをアプリケーションで操作する。

```
FILE-CONTROL.
    SELECT FILE1 ASSIGN TO      INF1-MSD.
:
FD FILE1
    LABEL RECORD IS      STANDARD
    VALUE OF      IDENTIFICATION IS      "B:¥S-codedata".
```

例(3)

プリンタファイルを使用する。

```
FILE-CONTROL.
    SELECT FILE1 ASSIGN TO      OUTF-PRN.
```

2. ファイルマップ機能

ファイルマップ機能とは、COBOL ソース上で内部ファイル ID とデバイスとして設定されている名称をキーとして、COBOL ファイルマッパーにより、実ファイル名やファイル種別などを定義する機能を指す。

(a) 内部ファイル ID

ファイルの FD 句に設定する VALUE OF IDENTIFICATION の値である。

(b) 設定

内部ファイル ID と関連付けて設定する値は、以下のとおりである。

ただし、デバイス、ファイル種別以外は COBOL アプリケーション実行時などでの整合性チェックはなし。

項目	意味
デバイス	FILE-CONTROL 段落 SELECT 句のデバイスタイプとデバイスアドレスを指定する。 指定を省略した場合は指定なしが選択される。
ファイル種別	ファイル機能の種類を指定する。 COBOL 独自ファイルまたはその他のファイル種別(COBOL DB アクセス, IFAS FILE, IFASPRO RDB など)を設定する。
実ファイル名	実際作業する際のファイル名を指定する。 この値は省略可能で、省略された場合は内部ファイル ID と同じ値を実ファイル名とみなす。
レコード長	ファイルのレコード長を指定する。 指定を省略することもできる。
ファイル編成	FILE-CONTROL 段落 SELECT 句の ORGANIZATION を指定する。 指定を省略することもできる。
注釈	ファイルに対する説明。 記述の有無に関しては実行に何ら影響しない。

COBOL ファイルマッパーの詳細は、『COBOL ユーザーズガイド』を参照。

(c) ファイル種別とデバイスタイプ

プログラム上のデバイスタイプによって設定できるファイル種別に制限がある。以下の表以外の組合せは、実行時エラーとなるので、注意が必要である。

デバイスタイプ	ファイル種別
MSD/FDU	COBOL 独自ファイル, IFAS FILE, ISAM, COBOL DB アクセス(索引ファイル)
PRN/PRT	指定できない。
RDB	IFASPRO RDB ファイル, COBOL DB アクセス

ファイルマップ機能が使用されていない場合は、動作既定値によりファイル種別が決定する(以下参照)。

ファイル編成	デバイスタイプ	動作既定値
順編成	MSD/FDU	IFAS FILE ^(注1)
行順編成	MSD/FDU	COBOL 独自

相対編成	MSD/FDU	IFAS FILE ^(注2)
索引編成	MSD/FDU	IFAS FILE ^(注3)

注意

(注1) -Cf オプション指定ありのとき、動作既定値を COBOL 独自とする。

(注2) -Cg オプション指定ありのとき、動作既定値を COBOL 独自とする。

(注3) -Ch オプション指定ありのとき、動作既定値を ISAM とする。

SE 動作モードの場合は、ファイル種別は固定であり、変更できない (以下参照)。

ファイル編成	デバイスタイプ	動作既定値
順編成	MSD/FDU	COBOL 独自
行順編成	MSD/FDU	COBOL 独自
相対編成	MSD/FDU	COBOL 独自
索引編成	MSD/FDU	ISAM

(d) 制限事項

ファイルマップ機能を使用するためには、以下のような制限事項がある。

項目	入力可能桁数	入力可能文字
内部ファイル ID	200 桁	”(ダブルコーテーション)=(イコール)以外の記号文字, 英数字, カナ文字, 日本語のみ可能。 これ以外の文字が指定された場合の処理は保証しない。
実ファイル名	259 桁	システムのロングファイル名形式に準じる。 相対ファイルおよび索引ファイルの拡張子を付加するため、 入力可能桁数は4桁短くなる。
注釈	256 桁	特に規定しない。

日本語は 1 文字 2 桁にて計算する。

内部ファイル ID が 200 桁を超える場合は、実ファイル名として認識され、ファイルマップ機能とは認識せず、今までどおり COBOL ソース上の設定のままに動作する。

3.2 レコード形式

3.2.1 固定長レコード形式

次の場合、レコード形式は固定長レコードとなる。

(1) 順・相対・索引ファイルのとき

RECORD 句やレコード記述項の書き方に関係なく、
レコードの入出力は最大レコード長(固定長)とする。

(2) RECORD 句の書き方 1 を記述したとき

例 RECORD CONTAINS 80 CHARACTERS(書き方 1)

...

01 RECORD1 PIC X(20).

01 RECORD2 PIC X(80).

レコードの入出力は最大レコード長(80 バイト)とする。

(3) RECORD 句を省略, または RECORD 句の書き方 3 を記述して,
レコード記述項が単一レコードで, かつ固定長のとき

例 RECORD CONTAINS 10 TO 80 CHARACTERS(書き方 3)

...

01 RECORD1 PIC X(40).

レコードの入出力は単一レコード長(40 バイト)とする。

3.2.2 可変長レコード形式

次の場合、レコード形式は可変長レコードとなる。

(1) RECORD 句の書き方 2 を記述したとき

例 RECORD IS VARYING IN SIZE

FROM 10 TO 80 CHARACTERS

DEPENDENT ON データ名 1(書き方 2)

...

01 RECORD1 PIC X(80).

レコードは RECORD DEPENDING ON デ-タ名 1 の値(可変長)で出力し、
入力したレコードの長さは RECORD DEPENDING ON デ-タ名 1 になる。

- (2) RECORD 句を省略, または RECORD 句の書き方 3 を記述して,
レコード記述項が複数レコード, あるいは可変長集団項目のとき

例 RECORD CONTAINS 40 TO 80 CHARACTERS(書き方 3)

...

01 RECORD1.

02 ITEM1 PIC X(10).

02 ITEM2 PIC X(10) OCCURS 3 TO 7 TIMES DEPENDING ON デ-タ名 2.

レコードは固定部分+OCCURS DEPENDING ON デ-タ名 2 の値(可変長)で出力し、
入力対象のレコード固有の長さで入力する。

3.3 プリンタファイル

3.3.1 事前準備

[Standard] プリンタファイルの事前準備について, 以下に説明する。

プリンタファイル機能を使用するには, アプリケーションを実行するマシンごとに, 以下の準備を行う必要がある。

本 COBOL は, COBOL アプリケーション実行環境(プリンタ機能)(以下, PRNAMR と記述)をサポートしている。なお, 本文中の設定<1>～<4>の詳細は, 「3.3.1.5 プリンタの動作モード指定方法」を参照。

3.3.1.1 PRNAMR とプリントマネージャの機能差異

PRNAMR とプリントマネージャ(以下, PRNMAN と記述)経由では, 以下のような機能差がある。

	PRNMAN	PRNAMR	機能差異項目
LINAGE のプリンタ連動	△	○	PRNMAN では LINAGE のページ行と同じ大きさになるように, プリンタ側に設定する必要がある。
デフォルト改行	不定	1/6	PRNMAN ではプリンタ側に設定されているデフォルトフォントを使用するので, その大きさにより改行幅を決定する。COBOL アプリケーションカスタマイザを使

			用することにより、使用するフォントを変更することが可能。
制御コード	○	○	PRNMAN では PASSTHROUGH モード機能により制御コードの発行が可能となる。
外字	Windows 外字	Express 外字	PRNMAN は Windows 外字を使用可能。 PRNAMR ではサーバ (Express) が存在する場合に使用可能。
PRT	○	○	PRNAMR で PRT 指定をした場合は、PRN のデバイス指定なしとして動作する。
フォーム	無効	○	PRNAMR では VISUALFORMS で作成したフォームを使用する。
メディア識別名	複数プリンタ指定	無効	PRNAMR ではプリントマネージャ経由の印刷ではない場合は無視される。
プリンタ種類	プリントマネージャの設定に依存	制限あり	PRNMAN ではプリントマネージャが対応しているプリンタはすべて対応が可能。 PRNAMR では対応機種に制限がある。

COBOL アプリケーション実行環境(プリンタ機能)の詳細な機能は、『COBOL アプリケーション実行環境 プリンタ機能 説明書』を参照。

PASSTHROUGH モードの詳細な機能は、『3.3.10 PASSTHROUGH モード機能』を参照。

3.3.1.2 PRNAMR による印字方法

PRNAMR を使用して印字を行う場合は、以下のいずれかの方法を選択することにより可能になる。

- ・ PRNAMR をインストールして、環境変数やレジストリは設定しない。
- ・ PRNAMR をインストールして、環境変数に設定<3>を設定する。
- ・ PRNAMR をインストールして、レジストリに設定<1>を設定する。

3.3.1.3 プリントマネージャによる印字方法

プリントマネージャを使用して印字を行う場合は、以下のいずれかの方法を選択することにより可能になる。

- ・ PRNAMR をインストールしない。
- ・ 環境変数に設定<4>を設定する。
- ・ レジストリに設定<2>を設定する。

さらに、以下の順に沿って設定する。

1. プリントマネージャの登録の確認
2. プリントマネージャでのプリンタの指定

3. 各プリンタの設定

2. で指定したプリンタごとに、用紙種類、サイズ、モードなどプリンタの設定を行う。

3.3.1.4 PRNAMR とプリントマネージャを併用する印字方法

PRNAMR を使用する COBOL アプリケーションと、使用しない COBOL アプリケーションを同一マシン内に混在したい場合は、以下の順に沿って設定する。

1. PRNAMR をインストールする。
2. 以下の表のいずれかに従って、各 COBOL アプリケーションの設定を行う。

	レジストリ	環境変数
PRNAMR を使用する (プリントマネージャを使用しない) COBOL アプリケーション用の設定	なし	なし
	設定<1>	なし
	設定<1>	設定<3>
	設定<1>	設定<4>
PRNAMR を使用しない (プリントマネージャを使用する) COBOL アプリケーション用の設定	なし	設定<4>
	設定<2>	なし
	設定<2>	設定<3>
	設定<2>	設定<4>

(例 1)

プリントマネージャを使用する COBOL アプリケーション以外の COBOL アプリケーションが PRNAMR を使用している場合は、次のように設定すれば実現可能になる。

- ・レジストリ 設定<2>
- ・環境変数 なし

(例 2)

PRNAMR を使用する COBOL アプリケーション以外の COBOL アプリケーションがプリントマネージャを使用している場合は、次のように設定すれば実現可能になる。

- ・レジストリ 設定<1>
- ・環境変数 設定<4>

3.3.1.5 プリンタの動作モード指定方法

プリンタの動作モード設定を列記する。

なお、レジストリの設定は COBOL アプリケーションカスタマイザを利用して設定する。

— レジストリによる設定 —

設定<1>:PRNAMR を使用する (プリントマネージャを使用しない)

`CBLAMRMODE=ON`

設定<2>:PRNAMR を使用しない (プリントマネージャを使用する)

`CBLAMRMODE=OFF`

なお、レジストリの設定については、『COBOL ユーザーズガイド』の「2.6.28 COBOL アプリケーション実行環境 (プリンタ機能) の使用を指定する」を参照。

— 環境変数による設定 —

設定<3>:PRNAMR を使用する (プリントマネージャを使用しない)

`COB_AMRMODE=ON`

設定<4>:PRNAMR を使用しない (プリントマネージャを使用する)

`COB_AMRMODE=OFF`

3.3.2 レジストリ(プリンタ関連)

[Standard] プリンタファイルのレジストリについて、以下に説明する。

COBOL アプリケーションカスタマイザを使用して設定を行うことにより、アプリケーション(実行可能ファイル)ごと、または複数のアプリケーション共通に、プログラムに対する各種設定を行える。

レジストリは、既定値またはメインプログラムの翻訳時にコンパイルオプション-Yr で、レジストリのキー名を指定する方法がある。なお、既定値の設定を COBOL アプリケーションカスタマイザで行い、これを使用する場合は、レジストリの指定は必要ない。

レジストリは、プリンタ関連以外もあるが、ここではプリンタ関連の指定についてのみ説明する。

レジストリのすべての指定は、『COBOL ユーザーズガイド』を参照。

プリンタ関連の設定は、ユーザ指定のキー名の Printer サブキーに設定する。

[設定内容]

データの名前	...¥Software¥[既定値／ユーザ指定キー名]¥Printer
値の名前	任意なプリンタ装置識別名(プログラム中で指定したメディア識別名)
データのタイプ	REG_SZ
データ	プリンタ名(プリントマネージャで設定したプリンタ名)

[設定例]

プログラム中のメディア識別名 CHOUHYOU に、SERVER1 に接続されたプリンタ(PRINTER1)を指

定する。

データの名前	HKEY_CURRENT_USER¥Software¥[ユーザ指定キー名]¥Printer
値の名前	CHOUHYOU
データのタイプ	REG_SZ
データ	¥¥SERVER1¥PRINTER1

3.3.3 プリンタファイルの装置およびファイル指定

[Standard] プリンタファイルの装置およびファイル指定について、以下に説明する。

「3.1 ファイルの装置およびファイル指定」で説明したが、ここではプリンタについてのプログラム上の指定について、詳しく説明する。

1. ファイル管理記述項の ASSIGN 句

SELECT ファイル名 1

ASSIGN { T0 { 内部ファイル識別名 2
定数 3 } ... }
T0 { 内部ファイル識別名 4 } ... }

内部ファイル識別名 2、定数 3 または内部ファイル識別名 4 に指定可能な記述は次のとおり。

種類	指定形式	装置タイプ
内部ファイル識別名 2	内部ファイル名-装置名[-フォーム名]	PRN
定数 3	“ 内部ファイル名-装置名[-フォーム名] ”	PRN
内部ファイル識別名 4	内部ファイル名-装置名[-メディア識別名] [{データ名 5 定数 3}]	PRT

内部ファイル識別名 2 または定数 3 の指定は、簡単にプリンタファイルを定義できる。フォームオーバーレイ機能のフォーム名をプリンタファイルに対して指定できるが、アプリケーション実行中に変更することはできない。

内部ファイル識別名 4 の指定は、内部ファイル識別名 2 または定数 3 の指定に比べて以下の指定が可能である。

- ・ プリンタ名をレジストリによるカスタマイズ機能で指定可能
- ・ フォーム名をアプリケーション実行中に指定／変更可能

2. ASSIGN 句とレジストリによるカスタマイズ機能

内部ファイル識別名 4 の ASSIGN 句にメディア識別名を指定する場合、「3.3.2 レジストリ(プリンタ関連)」で説明したようにレジストリに、メディア識別名に対応するプリンタ装置識別名を設定できる。

その際、プログラム中のメディア識別名とレジストリ中のプリンタ装置識別名を同じにする必要がある。

これにより、プリントマネージャを使用する場合は、「通常使うプリンタ」以外のプリンタへの印字が可能になる。

プログラム

メディア識別名にプリンタ装置識別名 CHOUHYOU を指定する。

FILE-CONTROL.

SELECT PRINT-F ASSIGN TO PRINTF-PRT-CHOUHYOU

レジストリ

プリンタ装置識別名 CHOUHYOU を定義する。

データの名前 HKEY_CURRENT_USER¥Software¥[ユーザ指定キー名]¥Printer

値の名前 CHOUHYOU

データのタイプ REG_SZ

データ PRINTER0 など

ASSIGN 句			レジストリ		アプリケーション実行結果
デバイスタイプ		メディア識別名	プリンタ装置識別名	プリンタ名	
PRN	PRT				
指定	—	—	指定は参照されない		既定プリンタ装置
—	指定	—	指定は参照されない		既定プリンタ装置
—	指定	指定	未指定	—	既定プリンタ装置
—	指定	指定	指定	未指定	既定プリンタ装置
—	指定	指定	指定	指定	指定プリンタ装置

3.3.4 ファイル形式

プリンタファイルの形式は、論理ページを設定して行う場合と物理ページに従って行う場合とでプログラムの書き方とプリンタファイルへの出力の動作が異なる。

1. 論理ページ

論理ページとは、プリンタファイルの論理的なページのことをいう。

論理ページの大きさと構造などは、LINAGE 句によって記述する。

論理ページモードとは、出力面(1 論理ページ)の改行／改ページ制御をシステムが自動的に管理するモードである。このモードを使用すれば、利用者はページの形式を決めるだけで、以降の印刷データ出力時の改行／改ページ制御に関しては意識する必要がない。

注意

- 論理ページサイズ(1 ページの行数)は、OPEN 文および WRITE 文の実行時に、LINAGE 句の指定値により決定する。『COBOL 言語説明書』「9.14.24.LINAGE 句」を参照。

[Standard] PRNMAN 使用時で-Qg オプション指定ありのとき、または PRNAMR 使用時は、OPEN 文の実行時にのみ、論理ページサイズを決定する。

2. 物理ページ

物理ページとは、プリンタの装置などによって、プログラムとは関係なく外部からページの大きさを設定したページをいう。物理ページを有効にするためにはプログラムに、LINAGE 句の記述があってはならない(記述があると、論理ページとなる)。

物理ページモードとは、装置に設定されている用紙の大きさや装置の改行／改ページに関するスイッチ類の設定状態を意識し、その環境にあった改行／改ページ制御を利用者自身が行うモードであり、システムによる改行／改ページ制御機能は働かない。装置固有の機能を使用する場合、論理ページモード制御では不十分な場合にこのモードを使用することができる。

注意

- 連続用紙を使用する場合、ページの切れ目で、すき間ができる。給紙方法などによって、その幅は決定される。用紙サイズを「ユーザ定義サイズ」に変更した場合、シートフィーダを選択すれば、その幅は 1 行程度のものになる。
- プリントマネージャの登録サイズの設定により、1 ページの印字行数は変化する。

3.3.5 論理ページと改行，改ページの動作

改ページは論理ページを基に行われる。

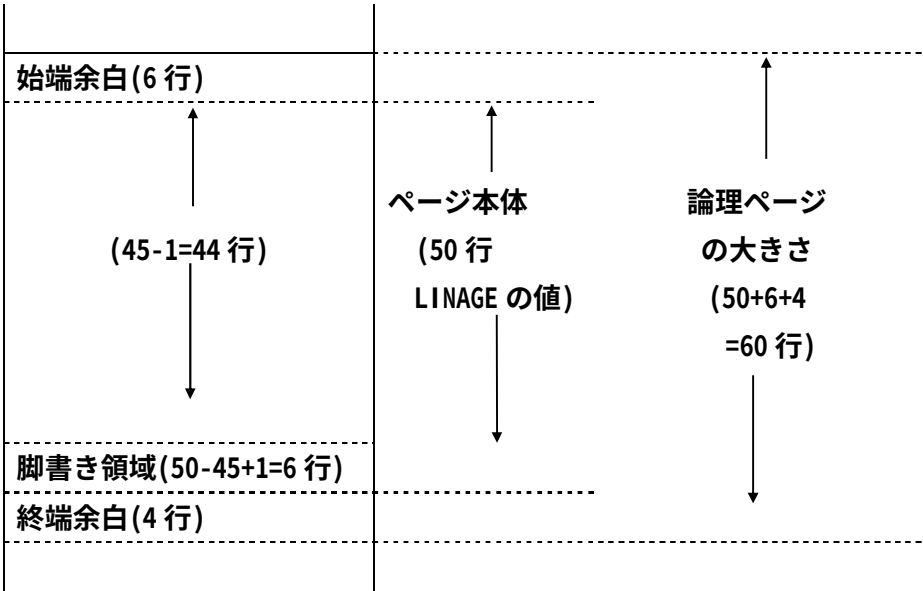
LINAGE 句で TOP 句が指定した場合，TOP 句で指定した余白は OPEN 文の実行時あるいは改ページの際にスキップされる。

3.3.5.1 改ページと論理ページとの関係

次に改ページと論理ページとの関係について，例を用いて説明する。

例 LINAGE 句および WRITE 文を次のように記述した場合

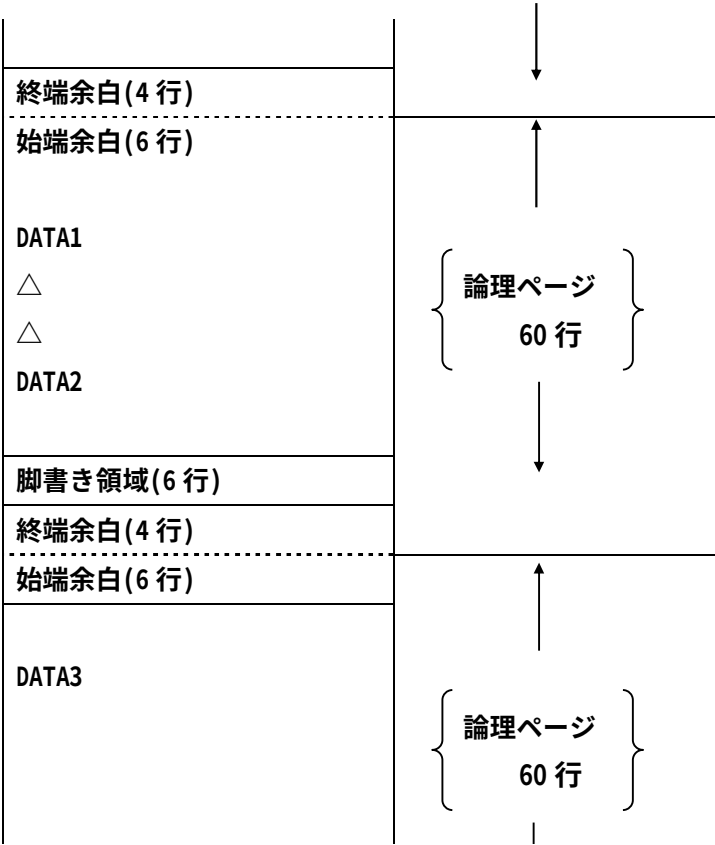
LINAGE IS 50 LINES
WITH FOOTING AT 45
LINES AT TOP 6
LINES AT BOTTOM 4.



FOOTING の 45 という値はページ本体の中で 45 行目以降の行へ印字を行うと，脚書き領域に印字位置が達した(WRITE 文で END-OF-PAGE 条件発生)状態となる行位置である。

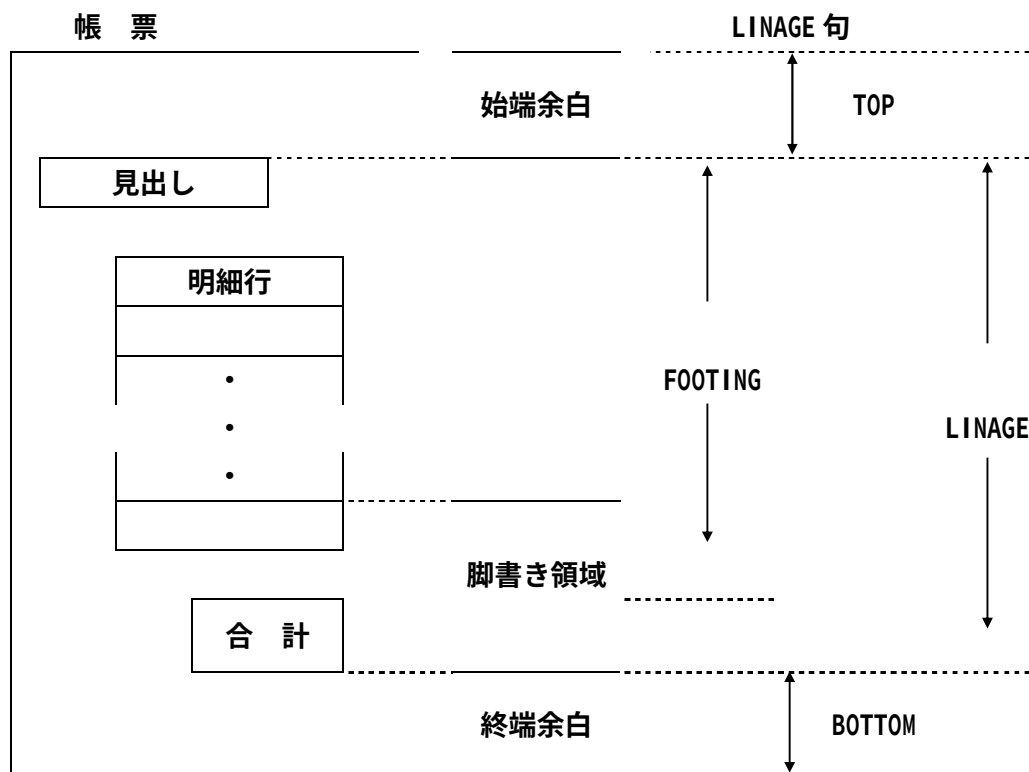
MOVE “ DATA1 ” TO PR-REC.

WRITE PR-REC AFTER PAGE.
MOVE “DATA2” TO PR-REC.
WRITE PR-REC AFTER 3 LINES.
MOVE “DATA3” TO PR-REC.
WRITE PR-REC AFTER PAGE.



3.3.5.2 脚書き領域を使用したページ制御

論理ページを使用するとき脚書き領域を定義し、ページ単位の合計印字あるいは次のページへ移るタイミングを簡単に認識できる。その使用例を次に説明する。



1. 脚書き領域を定義するときの指定方法

- 始端余白は TOP 句で指定する。
- 終端余白は BOTTOM 句で指定する。
- LINAGE 句では、LINAGE と TOP と BOTTOM の合計が帳票の 1 ページ分の行数になるように指定する。
- FOOTING では、明細行の最後の印字を行う WRITE 文が終了したとき印字位置が脚書き領域内になるように行数を指定する。
- 明細行を印字する WRITE 文には、END-OF-PAGE 句を必ず指定する(脚書き領域へ印字位置が達したときこの END-OF-PAGE 句で記述した無条件文が実行される)。
- 通常、脚書き領域内へ合計などを印字する WRITE 文には、END-OF-PAGE 句を指定しない。

2. 使用例

FD ファイル名

```
LABEL RECORD IS OMITTED  
LINAGE IS 56 LINES  
WITH FOOTING AT 52  
LINES AT TOP 6  
LINES AT BOTTOM 4.
```

01 レコード名.

レコード記述

* 明細行出力の WRITE 文

```
WRITE レコード名 AFTER n LINES  
AT END-OF-PAGE GO TO ~(脚書き処理)
```

* 脚書き領域の WRITE 文

```
WRITE レコード名 AFTER n LINES.
```

:

```
WRITE レコード名 BEFORE n LINES. ←この WRITE 文によって LINAGE の値を  
超える印字位置に改行する。
```

* 見出しの WRITE 文

```
WRITE レコード名 ~
```

3.3.6 入出力文

プリンタファイルに関連する COBOL 文には、次の文がある。

- OPEN 文
- WRITE 文
- CLOSE 文
- USE 文(「3.12 入出力状態」を参照)

表 3-1 プリンタファイルに対する文の動作

文	動 作	状態キー
OPEN 文	・プリンタファイルを使用できるように (OPEN) する	00
	・ [Standard] プリンタファイルを使用できるようにする (プリンタ装置およびポート番号の決定はこの時点で行われる)。	00
	・ [Standard] SG エラー (指定したプリンタ番号が未定義など) (COBOL アプリケーション実行環境 プリンタ機能 使用時)。	30
	・ [Standard] レジストリ設定値不正。	30
	・ [Standard] プリンタビジー (COBOL アプリケーション実行環境 プリンタ機能 使用時)。	91
WRITE 文	・プリンタファイルに対して、次の動作を行う。 1. 改行する前にレコードを印字する。 (BEFORE 句指定)…改行前印字 2. 改行した後にレコードを印字する。 (AFTER 句指定)…改行後印字 3. PAGE 句の指定があると改ページする。このとき BEFORE 句指定があると、改ページする前にレコードを印字し、AFTER 句の指定があると、改ページした後に印字する。 4. LINAGE 句を指定したファイルで END-OF-PAGE 句があると、印字位置が脚書き領域にきたときに WRITE 動作実行後、END-OF-PAGE 句の無条件文を実行する (ページ終了条件)。 印字位置がページ本体を超えたときは、自動的に改ページする。このとき BEFORE 句の指定があれば、改ページ前に印字され、AFTER 句の指定があれば、改ページ後、ページ本体の最初の行に印字する。END-OF-PAGE 句があると、その後無条件文が実行される。(ページあふれ条件の発生)	00

	・コード変換中にコード変換が不成功となった。	95
	・ [Standard] 紙切れ (COBOL アプリケーション実行環境 プリンタ機能 使用時)。	97
CLOSE 文	・プリンタファイルの利用を終了 (CLOSE) する。	00
	・物理エラーが発生。	30
	・ [Standard] 紙切れ (COBOL アプリケーション実行環境 プリンタ機能 使用時)。	97
共通	・メモリ不足エラー。	30
	・スプールファイルアクセス不正(スプール領域不足など)。	30
	・システムがエラーを通知した。	30
	・ [Standard] プリンタが準備されていない (COBOL アプリケーション実行環境 プリンタ機能 使用時)。	30
	・ [Standard] プリンタ装置障害(紙づまり, ネットワーク障害など) (COBOL アプリケーション実行環境 プリンタ機能 使用時)。	30

備考 LINAGE 句を指定したファイルの OPEN 文モードは, 出力モード(OUTPUT)だけを許す。

[Standard] プリンタファイルを使用する場合は, 「3.3.1 事前準備」で説明した準備を行っておく必要がある。

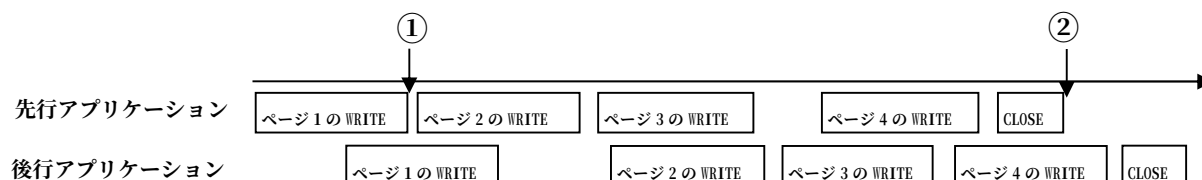
3.3.7 プリンタ共用機能

[Standard] プリンタファイルのプリンタ共用機能について, 以下に説明する。

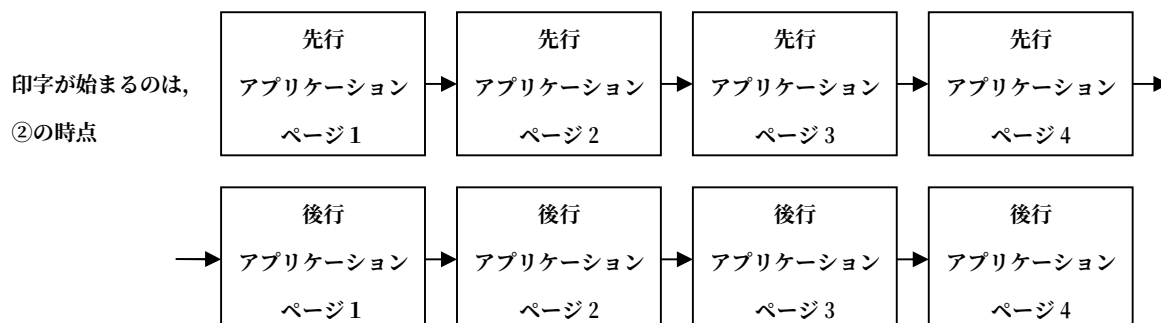
プリンタファイルでは, 複数のアプリケーションから同時に1台のプリンタを共用し, ページ単位に異なるプログラムから出力できるプリンタ共用機能がある。

プリンタ共用機能を使用しないと, Windows のプリントマネージャにより, 複数のアプリケーションから同時に出力(WRITE 文を実行)することは可能だが, 実際にプリンタに印字されるのは, アプリケーションにて, CLOSE 文が正常に実行された順になる。

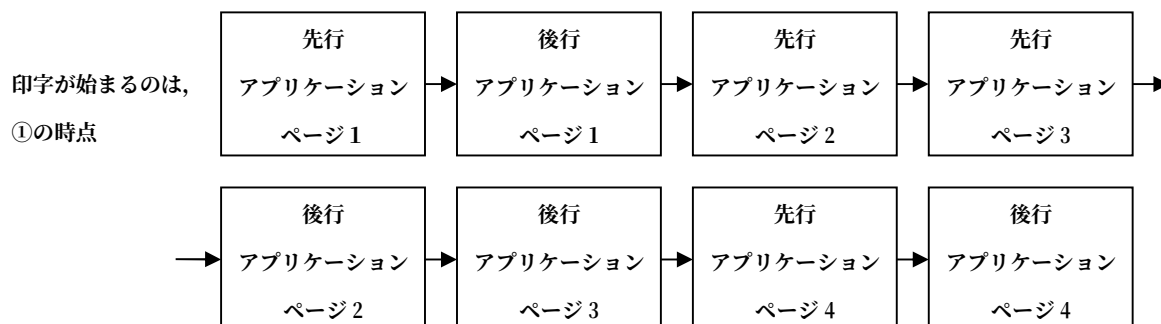
プリンタ共用機能を使用すると, 改ページ動作が行われるごとに用紙が印字される。



共用機能を使用しない場合の印字順



共用機能を使用する場合の印字順



3.3.7.1 プリンタ共用機能の条件

プリンタ共用機能を使用するには、以下の条件が必要となる。なお、これらの条件をすべて満たす必要がある。

1. 論理ページモードである(ファイル記述に LINAGE 句が指定されている)こと。
2. LINAGE 句の各指定(LINAGE, FOOTING, TOP, BOTTOM)が行われているとき、指定はすべて定数であること。
3. ファイルが共用モードであること(入出力管理段落において、APPLY SHARED-MODE が指定されている)。
4. 複数のアプリケーションから共用するときは、各アプリケーションで指定されている論理ページの構成(LINAGE, FOOTING, TOP, BOTTOM 各句の値)が同じであること。
5. 複数のアプリケーションから共用するとき、そのプリンタがすでに OPEN されているならば、すでに OPEN しているアプリケーションからもプリンタ共用機能の指定がされていること。
6. 複数のアプリケーションから共用するとき、そのプリンタがすでに OPEN されているならば、そのプリンタのメディア識別名と同じメディア識別名を指定するか、あるいはメディア識別名を指定せずに OPEN すること。

7. フォームオーバレイ使用時は、すでに OPEN されているプリンタのフォーム名と同じフォーム名を指定すること。
8. LABEL RECORD 句で STANDARD を指定し、直接スプールファイルとして定義していないこと。

3.3.7.2 プリンタ共用機能の動作

プリンタ共用機能は、次のように動作します。

1. 通常のプリンタファイルに対する WRITE 文で、出力データがシステムの確保したバッファに蓄積される。
2. 改ページ動作を伴う WRITE 文、または正常な CLOSE 文の実行で、蓄積された 1 ページ分のデータが実際にプリンタに印字される。

ここで改ページ動作を伴う WRITE 文とは次のことを指す。

- ・ WRITE 文の ADVANCING 句において、AFTER または BEFORE PAGE 句が指定されている。
- ・ WRITE 文に伴う改行によって、ページあふれ条件が発生した。

3.3.8 スプールファイルの出力機能

本 COBOL では、大容量記憶ファイル、標準出力(**stdout**)および標準エラー出力(**stderr**)に出力するプリンタファイルを指定することができる。これをスプールファイル出力機能と呼ぶ。

複数のプログラムから同時に 1 台のプリンタを共用して使用することができないので、同時出力の場合には、スプールファイル出力機能を使用して共用を回避する必要がある。

1. 大容量記憶ファイルへのプリンタファイル出力の記述

```
SELECT ファイル名 ASSIGN TO “PR-ファイル名”
または、
SELECT ファイル名 ASSIGN USING PR-データ名
または、
SELECT ファイル名 ASSIGN TO “ファイル名-PRN”
```

- ファイル名には、大容量記憶ファイル上のファイル名を指定する。
- データ名には、大容量記憶ファイル上のファイル名を格納する。

2. 標準出力(**stdout**)および標準エラー出力(**stderr**)へのプリンタ出力の記述

大容量記憶ファイルへのプリンタファイル出力の記述のファイル名として **stdout** および **stderr** を指定する。

3. 大容量記憶ファイル、標準出力(**stdout**)および標準エラー出力(**stderr**)を指定した場合の手続き部の文

通常のプリンタファイルに対する **OPEN** 文、**WRITE** 文、**CLOSE** 文と同様の記述をする。

なお、標準出力(**stdout**)および標準エラー出力(**stderr**)では通常のプリンタファイルに対する **WRITE** 文では発生しなかった 34(ファイルオーバーフロー)状態キーが発生する。

[Standard] プリンタファイルの直接スプールファイル出力機能について、以下に説明する。

本 COBOL では、直接プリンタ装置に出力するプリンタファイルの他に大容量記憶ファイルに出力するプリンタファイルを指定することができる。このことを直接スプールファイル出力機能と呼ぶ。この機能はフォームオーバレイ機能使用時も利用できる。詳細については「3.3.9 フォームオーバレイ機能」を参照。

1. 大容量記憶ファイルへのプリンタファイル出力の記述

a. SELECT 句

SELECT ファイル名 ASSIGN TO 内部ファイル名—装置指定

b. ファイル記述

直接スプールファイル出力機能を使用するときのファイルは、次のように指定する。

FD ファイル名 1

LABEL RECORD IS STANDARD

VALUE OF { IDENTIFICATION IS { データ名1 } } ...
定数1

- ・ LABEL RECORD 句には、STANDARD を指定する。
- ・ VALUE OF 句には、出力するスプールファイル名を定数、またはデータ名にて指定する。

2. プリントマネージャ経由の印字方法

直接スプールファイル出力機能を使用して出力した大容量記憶ファイルをプリンタに印字する方法は、以下のとおりである。

- ・ ファイルマネージャのコピー機能で「コピー先」に PRN を指定する。
- ・ プロンプト上で次のコマンドを実行する。

> COPY ファイル名 PRN

ただし、直接スプールファイル出力機能を使用した時に指定したプリンタと同等のプリンタドライバを使用するプリンタでなければならない。

注意

直接スプールファイル出力時には、プリントマネージャで一時停止を行ってはならない。一時停止を行うと、再開するまで COBOL アプリケーションの処理も停止する。

3. PRNAMR 経由の印字方法

直接スプールファイル出力機能を使用して出力した大容量記憶ファイルをプリンタに印字する方法は、以下のとおりである。

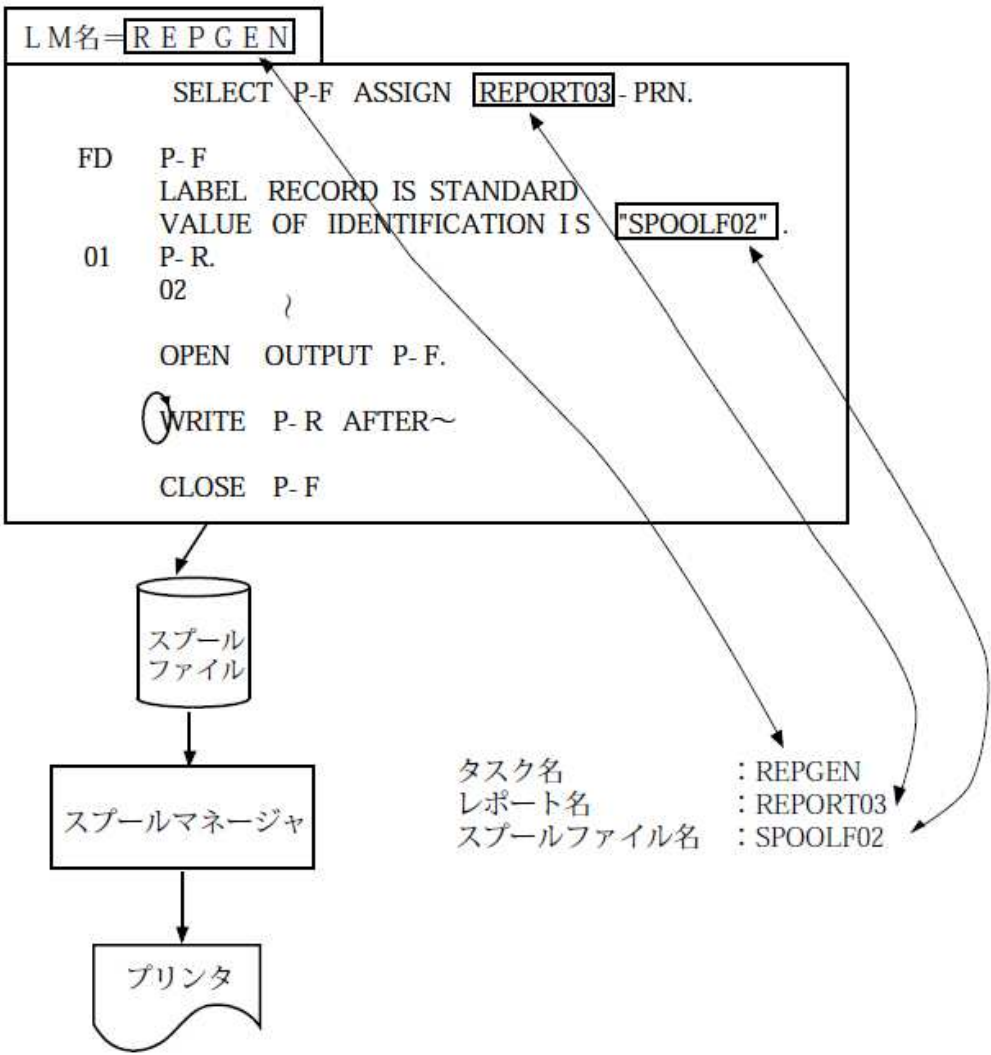
- ・ PRNAMR を使用して直接スプールファイル出力を行った場合は、スプール管理機能により、スプールマネージャで管理されているファイルを出力できる。
スプールマネージャなどの使用方法は、『COBOL アプリケーション実行環境 プリンタ機能説明書』を参照。

通常のプリンタファイルに対する WRITE 文では状態キーとして発生しなかった"34"(ファイルオーバーフロー)等の大容量記憶ファイル(順ファイル)で発生する状態キーが返却される。

スプールマネージャを使用すると次のようになる。スプールマネージャを使用したときの例を、次の図に示す。

- ・ タスク名は、そのプログラム名を使用する。
- ・ レポート名は、SELECT 句の ASSIGN 句の内部ファイル名を使用する。
- ・ スプールファイル名は、VALUE OF 句で指定されたファイル名を使用する。

スプールマネージャ使用時の動作



3.3.9 フォームオーバーレイ機能

[Standard] プリンタファイルのフォームオーバーレイ機能について、以下に説明する。

フォームオーバーレイ機能では、VISUALFORMS で作成したロジカルフォームオーバーレイデータを使用することができる。

VISUALFORMS を使用したフォームオーバーレイ機能では、ロジカルフォームオーバーレイデータを、専用用紙に印字するのと同じ手続きで使用することができる。

ロジカルフォームオーバーレイデータとは、従来レポートや伝票などを印字する場合に罫線や見出し語などがあらかじめ印刷された専用用紙を利用していたが、これを不要とするものである。すなわち、罫線や見出し語の印刷形式をファイル(フォームファイル)に格納しておき、プリンタがレポートや伝票を印字するときにこのファイルを読み取り、同時に印刷するものである。

3.3.9.1 PRNAMR 経由の印字におけるフォームの使用

1. フォームの設計

PRNAMR 経由のフォームオーバーレイ機能では、VISUALFORMS を使用して設計する。

2. COBOL プログラムにおける指定

PRNAMR 経由のフォームオーバーレイ機能を使用する場合、COBOL プログラムにおいては、以下のいずれかを指定します。

- a. ASSIGN 句(デバイスタイプ:PRN)のフォーム名に、フォームファイルを指定する。
- b. ASSIGN 句(デバイスタイプ:PRT)のデータ名または定数に、フォームファイルを指定する。

3. 実行時の設定

PRNAMR 経由の印字となるように設定する。

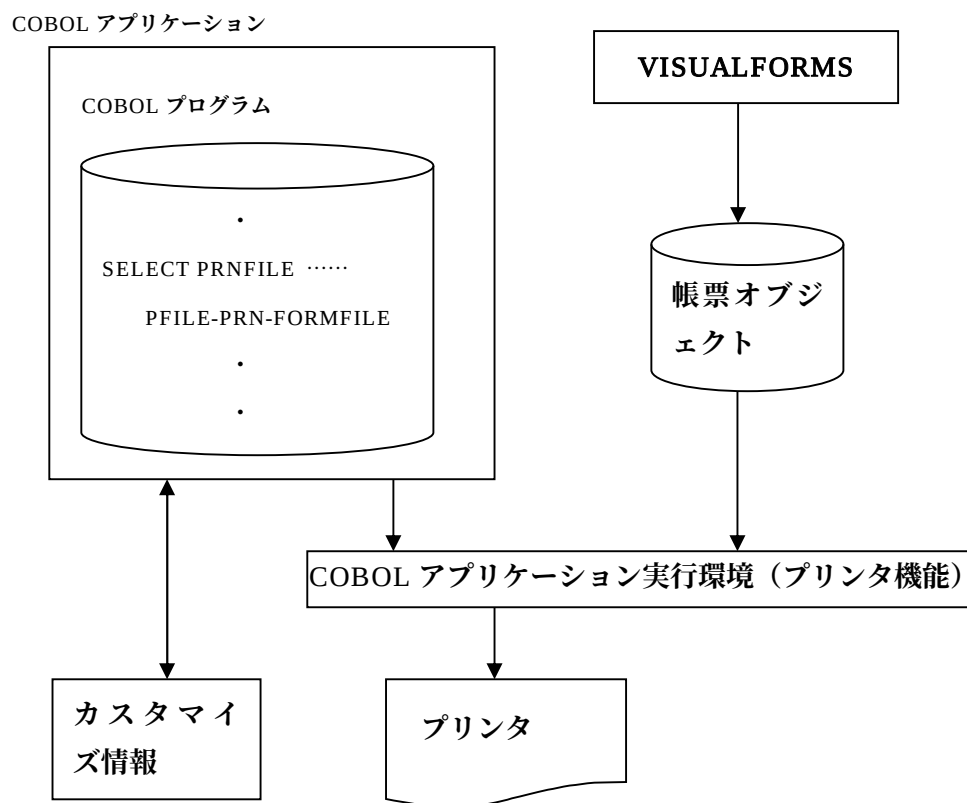
4. 使用方法

通常プリンタファイルに印字するのと同じ手続きで印字を行う。

5. フォームオーバーレイ機能使用時の注意事項

注意事項は『VISUALFORMS 添付のヘルプ』を参照。

VISUALFORMS を使用し COBOL アプリケーションからフォームオーバーレイ印字を行う場合の処理の流れを以下に示す。



3.3.9.2 フォームオーバーレイ機能使用例

以下のようなレポートを作成したい場合

コンピュータ関連消耗品納入書							
平成 7 年 5 月 24 日							
総務部 購買課							
御依頼に対しまして下記納入作業を行いますので、本用紙に捺印の上、返却願います。							
所 属 技術開発 部 P C 課				問題なければ確認印をお願いいたします。		確認印	
氏 名 C O B O L 様							
No.	商 品 名	商品コード	単 価	数 量	金 額	納 入 日	請 求 先 コード
1	3.5' FD	FE35	¥200	40	¥8000	95/ 1/ 1	FD-BUYER
2	5' FD	FE5	¥150	20	¥3000	95/ 1/ 1	FD-BUYER
3	8' FD	FE8	¥150	10	¥1500	95/ 1/ 1	FD-BUYER
4	3.5' FD	FE35	¥200	40	¥8000	95/ 1/ 1	FD-BUYER
購買→要求部門→購買				合 計 金 額 ¥20500			

1. まず以下のようなロジカルフォームオーバーレイをVISUALFORMSを用いて作成し、FORMFILEというフォームファイル名に格納しておく。

コンピュータ関連消耗品納入書							
平成 年 月 日							
総務部 購買課							
御依頼に対しまして下記納入作業を行いますので、本用紙に捺印の上、返却願います。							
所 属		部		課		問題なければ確認印をお願いいたします。	
氏 名				様		確認印	
No.	商 品 名	商品コード	単 価	数 量	金 額	納 入 日	請 求 先 コー ド
1							
2							
3							
4							
購買→要求部門→購買					合計金額		

2. 次にCOBOLで以下のように印字するプログラムを作成し, ASSIGN句のメディア識別名に
FORMFILEを指定する。

7 5 24

技術開発 PC

COBOL

3. 5' FD	PE35	¥200	40	¥8000 95/ 1/ 1	FD-BUYER
5' FD	FE5	¥150	20	¥3000 95/ 1/ 1	FD-BUYER
8' FD	FDE	¥150	10	¥1500 95/ 1/ 1	FD-BUYER
3. 5' FD	FD35	¥200	40	¥8000 95/ 1/ 1	FD-BUYER

¥20500

IDENTIFICATION DIVISION.

PROGRAM-ID. FORM01.

* フォームオーバーレイ機能サンプル *

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. Express5800.

OBJECT-COMPUTER. Express5800.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

 SELECT DATA-FILE ASSIGN TO DATAFILE-MSD.

 SELECT PRN-FILE ASSIGN TO NOUNYUU-PRN-FORMFILE.

DATA DIVISION.

FILE SECTION.

FD DATA-FILE LABEL RECORD IS STANDARD

 VALUE OF IDENTIFICATION "DATA1".

01 DATA-REC.

 02 SHOUHIN-MEI PIC X(16).

 02 SHOUHIN-CODE PIC X(10).

 02 SHOUHIN-TANKA PIC 9(6).

 02 SHOUHIN-SUU PIC 9(4).

 02 KINGAKU PIC 9(10).

 02 NOUNYUU-DAY.

 03 NEN PIC X(2).

 03 TUKI PIC X(2).

 03 HINITI PIC X(2).

 02 SEIKYUU-CODE PIC X(10).

 02 FILLER PIC X(2).

FD PRN-FILE LABEL RECORD IS OMITTED

 LINAGE IS 28 LINES

 WITH FOOTING AT 26

 LINES AT TOP 6

 LINES AT BOTTOM 2.

01 PRT-REC-KYOU.

 02 FILLER PIC X(77).

```

02 TODAY-NEN PIC ZZ.
02 FILLER PIC X(3).
02 TODAY-TUKI PIC ZZ.
02 FILLER PIC X(3).
02 TODAY-HINITI PIC ZZ.
01 PRT-REC-NAMAE.
02 FILLER PIC X(18).
02 SIMEI.
03 SHOZOKU-BU PIC N(7).
03 FILLER PIC X(3).
03 SHOZOKU-KA PIC N(4).
01 PRT-REC-GOUKEI.
02 FILLER PIC X(54).
02 GOUKEI-KINGAKU PIC XXXXXXXXXX9.
01 PRT-REC.
02 FILLER PIC X(14).
02 SHOUHIN-MEI PIC X(16).
02 SHOUHIN-CODE PIC X(10).
02 FILLER PIC X(2).
02 SHOUHIN-TANKA PIC XXXXX9.
02 FILLER PIC X(3).
02 SHOUHIN-SUU PIC ZZZ9.
02 FILLER PIC X(1).
02 KINGAKU PIC XXXXXXXXXX9.
02 FILLER PIC X(1).
02 NOUNYUU-DAY.
03 NEN PIC ZZ.
03 KIGOU1 PIC X(1).
03 TUKI PIC ZZ.
03 KIGOU2 PIC X(1).
03 HINITI PIC ZZ.
02 FILLER PIC X(3).
02 SEIKYUU-CODE PIC X(10).
02 FILLER PIC X(5).
WORKING-STORAGE SECTION.
01 KYOU.

```



```

02 TODAY-NEN PIC 9(2).
02 TODAY-TUKI PIC 9(2).
02 TODAY-HINITI PIC 9(2).
01 GOUKEI PIC 9(10) VALUE ZERO.
01 GYOUSUU PIC 9(3) VALUE ZERO.
01 SHOZOKU.
    02 SHOZOKU-BU1 PIC N(7) VALUE NC" 技術開発 ".
    02 SHOZOKU-KA1 PIC N(4) VALUE NC" P C ".
01 NAMAЕ-SIMEI.
    02 FILLER PIC X(5) VALUE SPACE.
    02 NAMAЕ PIC N(8) VALUE NC"COBOL".
PROCEDURE DIVISION.
HAJIME.
    OPEN OUTPUT PRN-FILE.
    OPEN INPUT DATA-FILE.
    MOVE SPACE TO PRT-REC.
    ACCEPT KYOU FROM DATE.
    ADD 12 TO TODAY-NEN OF KYOU.
LOOP.
    READ DATA-FILE
        AT END
            GO TO OWARI
    END-READ.
    IF LINAGE-COUNTER = 1
        PERFORM MIDASI-KAKIDASI
    END-IF.
    MOVE CORRESPONDING DATA-REC TO PRT-REC.
    ADD KINGAKU IN DATA-REC TO GOUKEI.
    WRITE PRT-REC BEFORE ADVANCING 2 LINES
        AT END-OF-PAGE
        PERFORM GOUKEI-KAKIDASI
        MOVE SPACE TO PRT-REC
        WRITE PRT-REC BEFORE PAGE
    END-WRITE.
    GO TO LOOP.
MIDASI-KAKIDASI.

```

```

MOVE CORRESPONDING KYOU TO PRT-REC-KYOU.
WRITE PRT-REC-KYOU
    BEFORE ADVANCING 8 LINES.
MOVE SHOZOKU-BU1 TO SHOZOKU-BU.
MOVE SHOZOKU-KA1 TO SHOZOKU-KA.
WRITE PRT-REC-NAMAE
    BEFORE ADVANCING 3 LINES.
MOVE NAMAE-SIMEI TO SIMEI.
WRITE PRT-REC-NAMAE
    BEFORE ADVANCING 5 LINES.
MOVE SPACE TO PRT-REC.
MOVE "/" TO KIGOU1 KIGOU2.
GOUKEI-KAKIDASI.
    COMPUTE GYOUSUU = 28 - LINAGE-COUNTER.
    MOVE SPACE TO PRT-REC-GOUKEI.
    MOVE GOUKEI TO GOUKEI-KINGAKU.
    WRITE PRT-REC-GOUKEI
        AFTER GYOUSUU.
    MOVE ZERO TO GOUKEI.
OWARI.
    CLOSE DATA-FILE.
    CLOSE PRN-FILE.
    STOP RUN.

```

3.3.10 PASSTHROUGH モード機能

[Standard] プリンタファイルの PASSTHROUGH モード機能について、以下に説明する。

PASSTHROUGH モード機能を使用することにより、COBOL 独自(プリントマネージャ経由)の場合にプリンタシーケンスの発行が可能となる。

3.3.10.1 使用方法

PASSTHROUGH モード機能は、レジストリまたは環境変数を設定することにより使用可能となるので、使用状況にあわせて使用する。

― レジストリによる設定 ―

レジストリの設定は COBOL アプリケーションカスタマイザを利用して行う。
[プリンタ]タブにて、キー名を設定する。



レジストリの設定値がない場合、環境変数の設定に準じる。ON でも OFF でもない値が設定されている場合、レジストリの設定がないものとして動作する。

― 環境変数による設定 ―

PASSTHROUGH モード機能を使用する場合

COB_PRNTHR=ON

設定値が ON ではない場合、環境変数の指定がないものとして動作する。
レジストリと環境変数を個々にまたは同時に指定された場合、以下の表の組み合わせで動作が決定される。

レジストリ	環境変数	動作
指定なし	ON	PASSTHROUGH モード機能使用
ON	ON	
ON	指定なし	
指定なし	指定なし	PASSTHROUGH モード機能未使用
OFF	指定なし	
OFF	ON	

3.3.10.2 使用例

すべての COBOL アプリケーションを PASSTHROUGH モードで動作させる場合

レジストリ :なし

環境変数 :COB_PRNTHR=ON

ほとんどの COBOL アプリケーションは PASSTHROUGH モードを使用するが、COBOL アプリケーション (α)だけは、PASSTHROUGH モードを使用しない場合

レジストリ (α用):THROUGHMODE:REG_SZ:OFF

環境変数 :COB_PRNTHR=ON

3.3.10.3 制限事項

- ・ Windows 外字は使用できない。
- ・ PASSTHROUGH モードでは、すべてのプリンタの制御を利用者側で行う必要がある。利用形態によっては新たに制御コードの発行が必要な場合もある。また、プリントマネージャ上でのユーザ設定も無視されることになるので注意が必要である。
- ・ PASSTHROUGH モードでは、プリンタに送出する制御コードとして日本語ページプリンタ言語 NPD L(Level 2)を使用している。使用するプリンタが NPD L(Level 2)に対応した機種であることを確認する必要がある。対応していない機種の場合、上記コードを有効とするために 201 エミュレーションモードで使用する。
- ・ プリンタ関連製品を利用する場合は、関連製品側の制御コード対応状況を確認する。
- ・ プリンタの制御は利用者側で行う必要があるが、印刷データについては、利用者側で記述されたシフト JIS から JIPS コードへの変換を自動的に行う。その際には、2 バイト文字コードの前に ESC K(0x1b4b)が付加され、2 バイト文字コードの後に ESC H(0x1b48)が付加される。

3.4 大容量記憶ファイルの排他／共用

【Enterprise for Windows】【Standard】本コンパイラでは、ファイルの排他／共用を行う場合、LOCK MODE 句は使用できない。

【Enterprise for Linux】本コンパイラでは、ファイルの排他／共用を行う場合、LOCK MODE 句は使用できる。ファイルの共用は、索引ファイルのみ使用できる。

【Enterprise for Linux】-MT オプション指定ありの場合、ファイルの排他／共用はサポートしない。

3.4.1 LOCK MODE 句によるファイルの共用／排他

【Enterprise for Linux】

大容量記憶ファイルは、ファイルオープン時にファイルを共用使用するか否かの指定が可能である。ファイルがオープンされた場合には、ファイルは別実行単位のプログラムから同時に参照・更新が可能になる。

大容量記憶ファイルを共用利用する場合には、ファイル管理記述項の LOCK MODE 句で AUTOMATIC または MANUAL を指定し、WITH LOCK 句なしの OPEN 文を実行する。

LOCK MODE 句の一般形式は、次のとおりである。

$$\text{LOCK MODE IS } \left\{ \begin{array}{l} \text{MANUAL [WITH LOCK ON MULTIPLE RECORDS]} \\ \text{AUTOMATIC [WITH LOCK ON RECORD]} \\ \text{EXCLUSIVE} \end{array} \right\}$$

3.4.2 ファイルの排他利用

【Enterprise for Linux】

LOCK MODE 句で EXCLUSIVE が指定された場合、あるいは WITH LOCK 指定の OPEN 文が実行された場合には、そのファイルは排他利用となる。すなわち、そのファイルを開いたプログラム以外の実行単位のプログラムからそのファイルは利用不可となる。

3.4.3 レコード排他制御

[Enterprise for Linux]

大容量記憶ファイル(索引ファイル)が共用利用されている場合、利用者は、レコード単位で排他制御が可能である。

レコード排他制御には、レコード排他制御をすべてシステム任せにする方法(AUTOMATIC 指定)と、利用者がレコード排他制御を行う方法(MANUAL 指定)の2つの方法がある。以下にそれぞれの説明をする。

(1) AUTOMATIC 指定

LOCK MODE 句で AUTOMATIC を指定した場合、レコード排他制御はシステムが自動的に行う。すなわち、システムは READ 文の実行が成功すると、READ の対象となったレコードに対してレコードロックを発行する。レコードロックが発行されたレコードは、他の実行単位のプログラムから更新することはできない。

その自動的に発行されたレコードロックは、他の入出力文が実行された時点で、システムによって自動的に解放される。

システムがかかるレコードロックの数は、1 つのファイルに対して最大 1 つであり、1 つのファイルに対して複数のレコードロックが発生することはない。

(2) MANUAL 指定

LOCK MODE 句で MANUAL を指定した場合、レコード排他制御はすべて利用者の責任で行うことになる。すなわち、READ 文を実行する際、READ したレコードに対してレコードロックを発行するのであれば、WITH LOCK 句を指定する。WITH LOCK 句指定のないかぎり、レコードロックは発生しない。

このレコードロックは、自動的に解放されない。解放するには、UNLOCK 文あるいは CLOSE 文を実行する必要がある。

LOCK MODE IS MANUAL の場合、同時にかけることのできるレコードロックの数に特に制限はない。

3.5 大容量記憶ファイル

大容量記憶ファイルには次種類のファイル編成がある。

- 順ファイル
- 行順ファイル
- 行順ファイル(CSV 形式)
- 相対ファイル
- 索引ファイル

[Enterprise for Windows] [Standard] 順ファイル, 行順ファイル, 行順ファイル(CSV 形式)には共通で, 次の注意事項がある。

1. ^Z に関する注意事項

COBOL で生成されるファイル(順ファイル, 行順ファイル), 行順ファイル(CSV)には, ファイル末尾に^Z(16進 1A)を付加しない。また, ファイルレコード中の^Zは単なるデータとして扱われ, ファイル末尾という意味合いのコードとしては扱わない。たとえば, エディタなどで作成された^Z付きのファイルを行順ファイルとしてアクセスした場合, READ 文では, 最後の^Zも1レコードとして読み込まれ, プログラムの動作に影響を与える可能性がある。エディタなどで作成されたファイルを COBOL のファイルとしてアクセスする場合は, ファイル末尾の^Zを取り除く必要がある。

3.5.1 レジストリ (ファイルマップ機能)

[Standard] 大容量記憶ファイルのレジストリについて, 以下に説明する。

複数のアプリケーションで共通に, 内部ファイル ID に対する各種設定をレジストリに行うことができる。そのキー名に対する設定は COBOL ファイルマッパーを使用し行う必要がある。

なお, アプリケーションに対して特別な設定をせず, 既定値のまま動作させる場合には, レジストリの指定は必要がない。

[設定内容]

次の順でレジストリを検索する。

キーの名前

	HKEY_CURRENT_USER¥Software¥NEC¥CBLAP DEFAULT¥File¥[内部ファイル ID](¥[デバイス])
	HKEY_LOCAL_MACHINE¥Software¥NEC¥CBLAP DEFAULT¥File¥[内部ファイル ID](¥[デバイス])
値の名称	ENGINE
データタイプ	REG_DWORD
データ名	[ファイル種別(数値)] ユーザー指定。
値の名称	FILENAME
データタイプ	REG_SZ
データ名	[実ファイル名] ユーザー指定。

上記やこれ以外の設定内容の詳細は『COBOL ユーザーズガイド』「第 3 章 COBOL ファイルマッパーの使い方」を参照。

3.6 大容量記憶ファイル(順ファイル)

3.6.1 入出力文

順ファイルで使用できる入出力文は、次の文がある。

- OPEN 文
- CLOSE 文
- READ 文
- WRITE 文
- REWRITE 文
- USE 文(「3.12 入出力状態」を参照)

以降に、各文の使用方法および動作を説明する。

(1) OPEN 文(順)

順ファイルの OPEN 文は、ファイルを利用できる状態に (OPEN) する。ただし最初のレコードを読んだり書いたりはしない。

また、OPEN 文のモードによって使用できる入出力文を次の表に示すように定める。

表 3-2 順ファイルの OPEN 文のモードと入出力文の関係

OPEN 文 のモード	文			用途
	READ 文	WRITE 文	REWRITE 文	
入力モード (INPUT)	○	—	—	ファイルの最初からレコードを読み込む。
出力モード (OUTPUT)	—	○	—	ファイルの最初からレコードを書き出す。
入出力両用モード (I-O)	○	—	○	ファイル中のレコードを読んだり、読み込んだレコードを置き換えたりする。置き換えは、READ 文→REWRITE 文の順に行う。
拡張モード (EXTEND)	—	○	—	ファイルの最後にレコードを追加する。

○:使用可能 —:使用不可

備考 -Qo オプション指定ありのとき、大容量記憶ファイル(COBOL DB アクセス形式の索引ファイルを除く)の先行する入出力文で入出力状態 35 の永続誤りが発生した場合であっても、そのファイルに対して OUTPUT 指定の OPEN 文を実行できる。

-Qp オプション指定なしのとき、SHARED-MODE または PROTECTED-MODE 指定の順ファイルに対して、ファイル共用機能は使用できない。

[Enterprise for Linux] 入力モード(INPUT)で開いた場合はファイル共用されるが、レコード排他は働かない。

(2) CLOSE 文(順)

順ファイルの標準ファイル閉じ (CLOSE) を実行する。OPEN したファイルは、必ず CLOSE しなければならない。LOCK 指定があれば、この実行単位ではファイルを再び開くことはできない。

[Standard] -Ql オプション指定ありのとき、LOCK 指定をメモ扱いとする。

(3) READ 文(順)

順ファイルの READ 文は、OPEN 文のモードが入力モード(INPUT)、入出力両用モード(I-O)で OPEN したファイルにだけに使用できる。

READ 文の動作はデータを順に読み込む。すなわち次のようになる。

READ	n 番目のレコードを読む。
↓	
READ	n + 1 番目のレコードを読む。
↓	
READ	n + 2 番目のレコードを読む。
:	

また READ 文では、さらに次の状態が発生する。

- READ 文実行中データの終わりが見つかると AT END 条件になる。

(4) WRITE 文(順)

順ファイルの WRITE 文は、OPEN 文のモードが出力モード(OUTPUT)および拡張モード(EXTEND)で OPEN されたファイルにだけ使用できる。WRITE 文の動作はデータを順に書き込む。すなわち次のようになる。

WRITE	n 番目のレコードを書き出す。
↓	

WRITE n + 1 番目のレコードを書き出す。
↓
WRITE n + 2 番目のレコードを書き出す。
↓
:

ただし拡張モード(EXTEND)で OPEN したファイルの場合は, OPEN 文でファイルの最後に位置づけた後,その後にレコードを追加して書き出す。

また WRITE 文では, さらに次の状態が発生する。

- WRITE 文実行中にファイルの終わりが検出された場合, FILE STATUS 句の指定があると, 状態キー(34)を格納する。ただし 01 レコードの内容は変更しない。

(5) REWRITE 文(順)

順ファイルの REWRITE 文は, 入出力両用モード(I-O)で OPEN したファイルについてだけ使用できる。REWRITE 文は, 直前に READ 文で読み込まれたレコードを新しいレコードで置き換える。すなわち, REWRITE 文を実行する前には, READ 文を実行しなければならない。

READ n 番目のレコードを読む。
↓
READ n + 1 番目のレコードを読む。
↓
REWRITE n + 1 番目のレコードを置き換える。
↓
READ n + 2 番目のレコードを読む。
↓
REWRITE n + 2 番目のレコードを置き換える。
↓
REWRITE 実行時エラー (文実行順序エラー)
↓
:

3.6.2 順ファイルの入出力文の動作

表 3-3 大容量記憶ファイル(順ファイル)に対する文の動作

文	動 作	状態キー
READ 文	・ 物理的な順序で次のレコードを読む	00

	コード変換後のレコードの大きさが、レコード記述の大きさと合わないとき、状態キーに"04"を設定する。レコード記述項で指定した大きさに収まらなかったレコードの内容は切り捨てられる。レコード記述項で指定した大きさに満たなかった場合は、空白を埋める。	04
	- レコードの終わりが検出されると AT END 条件となり、状態キーには"10"を設定する。 - AT END 条件が起こるとレコード領域の内容は保証されない。	10
WRITE 文	レコードを順番に書き出す。	00
	- ファイルの容量を超えて書き出そうとするとファイルオーバーフローとして状態キーに"34"を設定する。 - ただし、FILE STATUS 句の指定のないファイルの場合はオーバーフローのメッセージを表示後、プログラムは異常終了する。	34
REWRITE 文	直前の READ 文で読み込んだレコードを新しいレコードで置換する。	00
	- 直前の文が成功した READ 文でないと、実行順序エラーとして状態キーに"43"を設定する。 - ただし、FILE STATUS 句のないファイルの場合は、実行順序エラーのメッセージを表示後、プログラムは異常終了する。	43
	- REWRITE 文でレコードを書き換えようとしたが、書き換えるレコードが書き換えられるレコードと同じ大きさでないと状態キーに"44"を設定する。 - ただし、FILE STATUS 句の指定のないファイルの場合はファイル入出力エラーのメッセージを表示後、プログラムは異常終了する。	44
共通	- コード変換中にコード変換が不成功となったとき状態キーに"95"を設定する。 - ただし、FILE STATUS 句の指定のないファイルの場合はコード変換エラーのメッセージを表示後、プログラムは異常終了する。 READ 文, WRITE 文, REWRITE 文でコード変換が行われる場合の動作に関しては、「3.13 コード変換機能」を参照。	95

3.7 大容量記憶ファイル(行順ファイル)

行順ファイル(line sequential file)とは、レコードが改行コードで区切られているファイルである。実行動作は順ファイルに準ずる。

[Enterprise for Windows] [Standard] 改行コード:16 進 0D0A

[Enterprise for Linux] 改行コード:16 進 0A

3.7.1 入出力文

行順ファイルで利用できる入出力文は、次の文がある。

- OPEN 文
- CLOSE 文
- READ 文
- WRITE 文
- REWRITE 文
- USE 文(「3.12 入出力状態」を参照)

次に、各文の使用方法および動作を説明する。

(1) OPEN 文(行順)

行順ファイルの OPEN 文は、ファイルを利用できる状態(OPEN)にする。ただし最初のレコードを読んだり書いたりはしない。

また、OPEN 文のモードによって利用できる入出力文を次の表に示すように定める。

表 3-4 行順ファイルの OPEN 文のモードと入出力文の関係

OPEN 文 のモード	文			用途
	READ 文	WRITE 文	REWRITE 文	
入力モード (INPUT)	○	—	—	ファイルの最初からレコードを読み込む。
出力モード (OUTPUT)	—	○	—	ファイルの最初からレコードを書き出す。
入出力両用モード	○	—	○	ファイル中のレコードを読んだり、

(I-O)				読み込んだレコードを置き換えたりする。置き換えは、READ → REWRITE の順で行う。
拡張モード (EXTEND)	—	○	—	ファイルの最後にレコードを追加する。

○:使用可能 —:使用不可

備考 -Qo オプション指定ありのとき、大容量記憶ファイル(COBOL DB アクセス形式の索引ファイルを除く)の先行する入出力文で入出力状態 35 の永続誤りが発生した場合であっても、そのファイルに対して OUTPUT 指定の OPEN 文を実行できる。

 -Qp オプション指定なしのとき、SHARED-MODE または PROTECTED-MODE 指定の CSV 指定なし行順ファイルに対して、ファイル共用機能は使用できない。

 [Enterprise for Linux] 入力モード(INPUT)で開いた場合はファイル共用されるが、レコード排他は働かない。

(2) CLOSE 文(行順)

行順ファイルの標準ファイル閉じ(CLOSE)を実行する。OPEN したファイルは、必ず CLOSE しなければならない。LOCK 指定があれば、この実行単位ではファイルを再び開くことはできない。

[Standard] -Ql オプション指定ありのとき、LOCK 指定をメモ扱いとする。

(3) READ 文(行順)

行順ファイルの READ 文は、OPEN 文のモードが入力モード(INPUT)、入出力両用モード(I-O)で OPEN されたファイルにだけ使用できる。そして、入力ファイルから改行コードで区切られた範囲をレコードとして読み込み、参照可能にする。

改行コードで区切られたレコードが、プログラム上に記述された 01 レコードの大きさより大きい場合、1 回目の READ 文で、FILE STATUS に"04"が戻され、指定したレコードの値に収まらなかったデータは切り捨てる。

また READ 文では、READ 文実行中データの終わりが見つかると、AT END 以下で指定した無条件文が実行される。これを、ファイル終了条件(AT END 条件)が発生したという。

READ 文の動作はデータを順に読み込む。すなわち次のようになる。

```

READ                      n 番目のレコードを読む。
↓
READ                      n + 1 番目のレコードを読む。
↓

```

```

READ          n + 2 番目のレコードを読む。
:

```

(4) WRITE 文(行順)

行順ファイルの WRITE 文は, OPEN 文のモードが出力モード(OUTPUT)および拡張モード(EXTEND)で OPEN されたファイルにだけ使用できる。

01 レコードの末尾より連続する空白を削除しないで, 改行コードを付加して, 出力ファイルに 1 レコードを書き出す。

[Standard] -Qe オプション指定ありのとき, 01 レコードの末尾より連続する空白を削除し, 改行コードを付加して, 出力ファイルに 1 レコードを書き出す。

WRITE 文の動作は, データを順次書き出す。すなわち次のようになる。

```

WRITE          n 番目のレコードを書き出す。
↓
WRITE          n + 1 番目のレコードを書き出す。
↓
WRITE          n + 2 番目のレコードを書き出す。
:

```

ただし, 拡張モード(EXTEND)で OPEN したファイルの場合は, OPEN 文でファイルの最後に位置づけた後, その後にレコードを追加して書き出す。

また WRITE 文の実行中にファイルの終わりが検出された場合, FILE STATUS 句の指定があると, 状態キーに(34)が設定される。ただし, 01 レコードの内容は変更しない。

(5) REWRITE 文(行順)

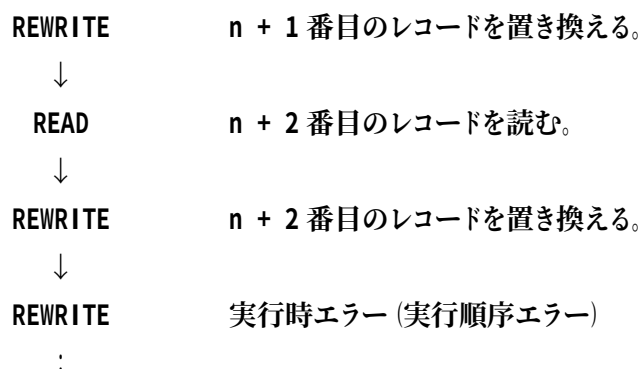
行順ファイルの REWRITE 文は, 入出力両用モード(I-O)モードで OPEN されたファイルについてだけ使用できる。REWRITE 文は, 直前に READ 文で読み込まれたレコードを新しいレコードで置き換える。すなわち, REWRITE 文を実行する前には, READ 文を実行しなければならない。

REWRITE 文の動作は, データを書き換える。すなわち次のようになる。

```

READ          n 番目のレコードを読む。
↓
READ          n + 1 番目のレコードを読む。
↓

```



そして、REWRITE 文では、直前の成功した READ 文で読み込んだ実際のファイルの改行コードで区切られたレコードの大きさと、REWRITE しようとするレコードの大きさが比較される。

比較した結果

- REWRITE しようとするレコードが大きい場合
FILE STATUS 句の指定があると、状態キーに(44)が戻され、REWRITE 文は失敗する。
- 両方の大きさが等しい場合
新しいレコードと置き換える。
- REWRITE しようとするレコードが小さい場合
FILE STATUS 句の指定があると、状態キーに(44)が戻され、REWRITE 文は失敗する。

[Standard] -Qe オプション指定ありのとき、REWRITE しようとするレコードの大きさは、01 レコードの末尾より連続する英数字の空白を削除した大きさである。REWRITE しようとするレコードが小さい場合、足りない所は空白が補われ、置き換える。FILE STATUS 句の指定があると、状態キーに(00)が戻される。

(6) [Standard] 特殊レジスタ RECORD-LENGTH

行順ファイルからの入力において、入力されたレコードの大きさは可変であり、プログラミング上、実際の大きさにより処理を切り分けることが必要となる。そのために、入力されたレコードの大きさが設定されるものとして、特殊レジスタ RECORD-LENGTH がある。

特殊レジスタ RECORD-LENGTH は、COMP-2 項目として、行順ファイルにそれぞれ 1 つずつ生成される。複数の行順ファイルを定義した場合には、ファイル名で修飾して参照する。

なお、REWRITE/WRITE 文の実行時には、この特殊レジスタは使用されない。

3.7.2 行順ファイルの入出力文の動作

表 3-5 大容量記憶ファイル(行順ファイル)に対する文の動作

文	動 作	状態キー
READ 文	物理的な順序で次のレコードを読む	00
	- コード変換後のレコードの大きさが、レコード記述の大きさと合わないとき、状態キーに"04"を設定する。レコード記述項で指定した大きさに収まらなかったレコードの内容は切り捨てられる。レコード記述項で指定した大きさに満たなかった場合は、空白を埋める。 [Standard] -Qe オプション指定ありのとき、コード変換後のレコードの大きさが、レコード記述の大きさより小さいとき、状態キーに"00"を設定する。 - 読み込んだレコードの文字位置の個数がレコード記述項で指定した最大の大きさより大きいとき、または指定された最小の大きさより小さいとき状態キーに"04"を設定する。	04
	- レコードの終わりが検出されると AT END 条件となり、状態キーには"10"を設定する。 - AT END 条件が起こればレコード領域の内容は保証されない。	10
WRITE 文	レコードを順番に書き出す。	00
	- ファイルの容量を超えて書き出そうとするとファイルオーバーフローとして状態キーに"34"を設定する。 - ただし、FILE STATUS 句の指定のないファイルの場合はオーバーフローのメッセージを表示後、プログラムは異常終了する。	34
	- 関連するファイル名の RECORD IS VARYING 句によって許される最大のレコードより大きい、または最小のレコードより小さいレコードを書き出そうとした時、状態キーに"44"を設定する。 - ただし、FILE STATUS 句の指定のないファイルの場合は、ファイル入出力エラーのメッセージを表示後、プログラムは異常終了する。	44
REWRITE 文	直前の READ 文で読み込んだレコードを新しいレコードで置換する。	00
	- 直前の文が成功した READ 文でないと、実行順序エラーとして状態キーに"43"を設定する。 - ただし、FILE STATUS 句のないファイルの場合は、実行順序エラーのメッセージを表示後、プログラムは異常終了する。	43

	• [Enterprise] レコードが書き換えられるレコードと同じ大きさでないとき状態キーに"44"を設定する。 • [Standard] -Qe オプションを指定しないとき、レコードが書き換えられるレコードと同じ大きさでない、状態キーに"44"を設定する。 • [Standard] -Qe オプションを指定したとき、直前に読まれたレコード長より大きいレコードを置換しようとする、状態キーに"44"を設定する。 • ただし、FILE STATUS 句の指定のないファイルの場合はファイル入出力エラーのメッセージを表示後、プログラムは異常終了する。 • 直前の READ で読み込んだレコードの文字位置の個数がレコード記述項で指定された最大の大きさより大きい、または最小の大きさより小さいときに"44"を設定する。	44
	• [Standard] -Qe オプションを指定したとき、直前に読まれたレコード長より小さいレコードを置換しようとする、残りの領域に空白が補われ、置換される。	00
共通	• コード変換中にコード変換が不成功となったとき状態キーに"95"を設定する。 • ただし、FILE STATUS 句の指定のないファイルの場合はコード変換エラーのメッセージを表示後、プログラムは異常終了する。 READ 文, WRITE 文, REWRITE 文でコード変換が行われる場合の動作に関しては、「3.13 コード変換機能」を参照。	95

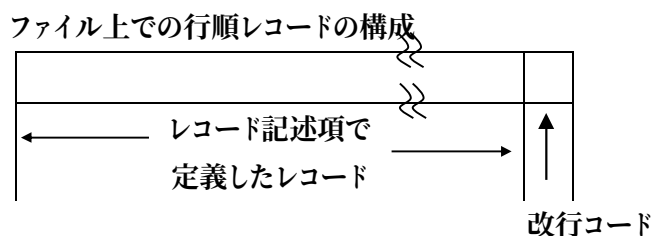
3.7.3 行順ファイルのファイル構成および使用上の留意点

(1) レコードの構成

行順ファイルのレコードは、利用者がレコード記述項で定義したレコードの末尾に改行コードを付与して格納する。

[Enterprise for Windows] [Standard] 改行コード:16 進 0D0A

[Enterprise for Linux] 改行コード:16 進 0A



改行コードで区切られたレコードが、プログラム上に記述された 01 レコードの大きさより大きい場合、または RECORD 句で指定された値の範囲外の場合、READ 文で、FILE STATUS に "04" を設定し、次の処理を行う。

- 実際に読み込んだレコードが大きい場合、指定したレコードの値に収まらなかったデータは切り捨てる。
- 実際に読み込んだレコードが小さい場合、空白(16 進 20)を埋める。

環境変数 COB_FILETAB を設定することにより、READ 文で読み込んだレコード中のタブ文字(16 進 09)が空白に拡張される。拡張されたレコードに対する READ 文の後の REWRITE 文は、ステータス "44" のエラーとなるので注意されたい。

行順ファイルのレコード中に、制御コード(16 進 00～0F)と同等の値が入った場合、改行コードやタブ文字(16 進 09)のように行順ファイルの動作に誤りを発生させる。

3.8 大容量記憶ファイル(行順ファイル(CSV 形式))

行順ファイル(CSV 形式)とは、区切り記号で区切られた行順ファイルである。実行動作は行順ファイルに準ずる。行順ファイル(CSV 形式)のレコードは、用途が表示用のデータ以外は指定できない。

3.8.1 入出力文

行順ファイルの拡張形式として、行順ファイル(CSV 形式)がある。

行順ファイル(CSV 形式)の特徴は、次のとおり。

1. レコードの末尾に改行コードが付与される。
[Enterprise for Windows] [Standard] 改行コード:16 進 0D0A
[Enterprise for Linux] 改行コード:16 進 0A
2. レコード中のデータが区切り記号(コンマ)によって区切られている。
3. レコード中の数字項目以外のデータはダブルクォート(")で囲まれている。
4. データベースとの親和性が高い。

行順ファイル(CSV 形式)で使える入出力文は、次の文がある。

- OPEN 文
- CLOSE 文
- READ 文
- WRITE 文
- USE 文(「3.12 入出力状態」を参照)

次に、各文の使用方法および動作を説明する。

(1) OPEN 文(行順ファイル(CSV 形式))

行順ファイル(CSV 形式)の OPEN 文は、ファイルを利用できる状態(OPEN)にする。ただし最初のレコードを読んだり書いたりしない。

また、OPEN 文のモードによって使える入出力文を次の表に示すように定める。

表 3-6 行順ファイル(CSV 形式)の OPEN 文のモードと入出力文の関係

OPEN 文のモード	文			用途
	READ	WRITE	REWRITE	
入力モード (INPUT)	○	—	—	ファイルの最初からレコードを読み込む。
出力モード (OUTPUT)	—	○	—	ファイルの最初からレコードを書き出す。
拡張モード (EXTEND)	—	○	—	ファイルの最後にレコードを追加する。

○:使用可能 —:使用不可

備考 -Qo オプション指定ありのとき、大容量記憶ファイル(COBOL DB アクセス形式の索引ファイルを除く)の先行する入出力文で入出力状態 35 の永続誤りが発生した場合であっても、そのファイルに対して OUTPUT 指定の OPEN 文を実行できる。

(2) CLOSE 文(行順ファイル(CSV 形式))

行順ファイル(CSV 形式)の CLOSE 文は標準ファイル閉じ(CLOSE)を実行する。OPEN したファイルは、必ず CLOSE しなければならない。LOCK 指定があれば、この実行単位ではファイルを再び開くことはできない。

[Standard] -Ql オプション指定ありのとき、LOCK 指定をメモ扱いとする。

(3) READ 文(行順ファイル(CSV 形式))

行順ファイル(CSV 形式)の READ 文は、入力ファイルから 1 レコード読み込んで、参照可能な状態にする。行順ファイル(CSV 形式)の READ 文は、OPEN 文のモードが入力モード(INPUT)で OPEN されたファイルだけに使用できる。そして、入力ファイルから改行コードで区切られた範囲を区切り記号で区切られたデータ単位(セルと呼ぶ)に分割し、INTO 句がある場合は一意名に従属する基本項目の、INTO 句がない場合はレコードに従属する基本項目の属性にあわせて、セルと基本項目のならびに従い、先頭から 1 対 1 に対応させてレコードとして読み込み、参照可能にする。

次の場合、FILE STATUS 句の指定があると、状態キーに '04' を設定する。

- 1 つのセルの文字列長が対応する基本項目長よりも長い
- 数値データの桁あふれが発生する
- 1 レコード中のセルの個数と対応する基本項目の個数が等しくない

また、READ 文では、READ 文実行中にデータの終わりが見つかると、AT END 句で記述した無条件文を実行する。これをファイル終了条件(AT END 条件)が発生したという。

(4) WRITE 文(行順ファイル(CSV 形式))

行順ファイル(CSV 形式)の WRITE 文は、出力ファイルに 1 レコードを書き出す。行順ファイル(CSV 形式)に対する WRITE 文は、OPEN 文のモードが出力モード(OUTPUT)および拡張モード(EXTEND)で OPEN されたファイルにだけ使用できる。WRITE 文の FROM 句がある場合は、一意名に従属する基本項目の属性にあわせて、基本項目毎に、先頭から区切り記号で区切って出力ファイルに 1 レコードを書き出す。WRITE 文の FROM 句がない場合は、レコードに従属する基本項目の属性にあわせて、基本項目毎に、先頭から区切り記号で区切って出力ファイルに 1 レコードを書き出す。このとき、レコードの終わりに改行コードを付加する。

また行順ファイル(CSV 形式)の WRITE 文では、WRITE 文実行中にファイルの終わりが検出された場合、FILE STATUS 句の指定があると、状態キーに"34"を設定する。ただし、01 レコードの内容は変更しない。FILE STATUS 句の指定がないと、エラーメッセージを表示し、プログラムは異常終了する。

3.8.2 行順ファイル(CSV 形式)における入出力文の動作

表 3-7 大容量記憶ファイル(行順ファイル(CSV 形式))に対する文の動作

文	動 作	状態キー
READ 文	物理的な順序で次のレコードを読む	00
	1 つのセルの文字列長が対応する基本項目の項目長よりも長い場合、数値データの桁あふれが発生する場合、または 1 レコード中のセルの個数と基本項目の個数が等しくない場合、状態キーに"04"を設定する。	04
	- レコードの終わりが検出されると AT END 条件となり、状態キーには"10"を設定する。 - AT END 条件が起こるとレコード領域の内容は保証されない。	10
WRITE 文	レコードを順番に書き出す。	00
	ディスクの空き容量不足が発生した場合、状態キーに"34"を設定する。ただし、FILE STATUS 句の指定のないファイルの場合は、エラーメッセージを表示後、プログラムは異常終了する。	34

3.8.3 行順ファイル(CSV 形式)の動作例

行順ファイル (CSV 形式) の動作の例を次に記述する。

CSV ファイルの 1 レコード内容

ABCDE,1234,XYZ

CSV ファイルのレコードのプログラム定義

01 CSV-REC.

03 CSV-CELL1 PIC X(4).

03 CSV-CELL2 PIC 9(5).

03 CSV-CELL3 PIC X(3).

READ 文の INTO 句/WRITE 文の FROM 句の一意名のプログラム定義

01 DATA1.

03 DATA1-1 PIC X(6).

03 DATA1-2 PIC X(5).

03 DATA1-3 PIC X(3).

01 DATA2.

03 DATA2-1 PIC X(3).

03 DATA2-2 PIC 9(3).

01 DATA3.

03 DATA3-1 PIC X(2).

03 DATA3-2 PIC X(2).

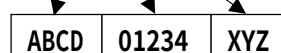
03 DATA3-3 PIC X(2).

03 DATA3-4 PIC X(2).

(1) INTO 句なしの READ 文の例

- READ CSV-FILE

CSV ファイル ABCDE,1234,XYZ



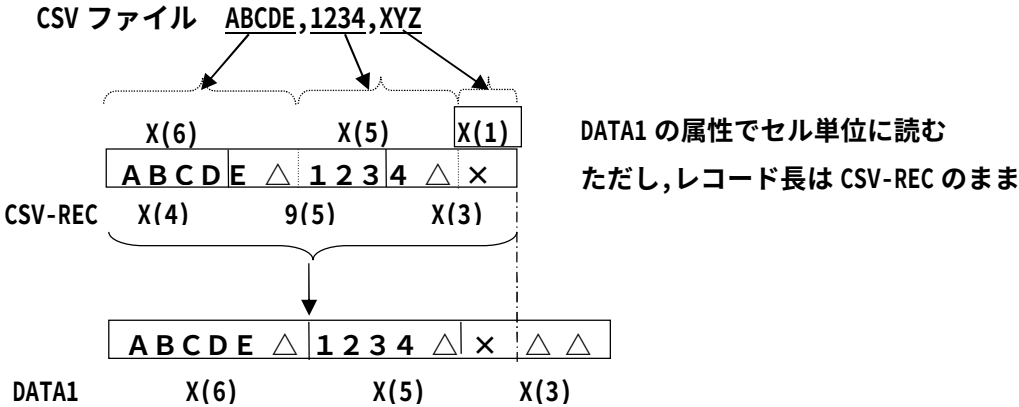
ABCD	01234	XYZ

CSV-REC X(4) 9(5) X(3)

(2) INTO 句ありの READ 文の例

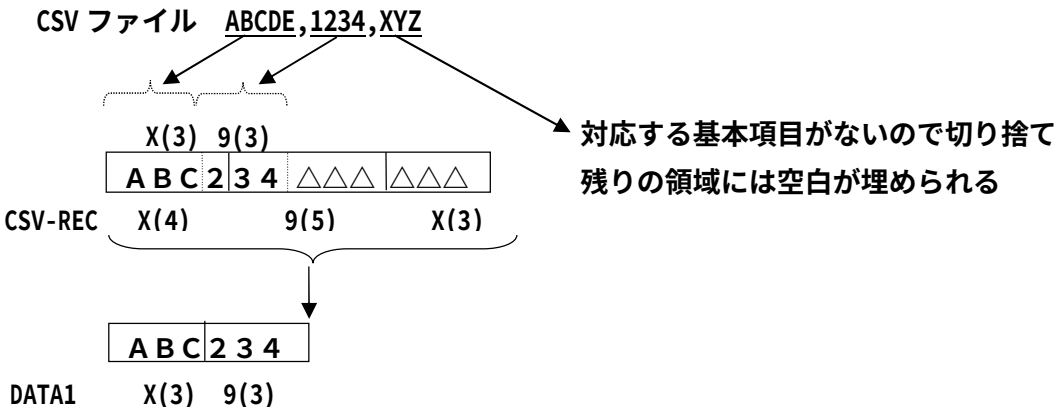
• READ CSV-FILE INTO DATA1

※ “△” は空白を表す



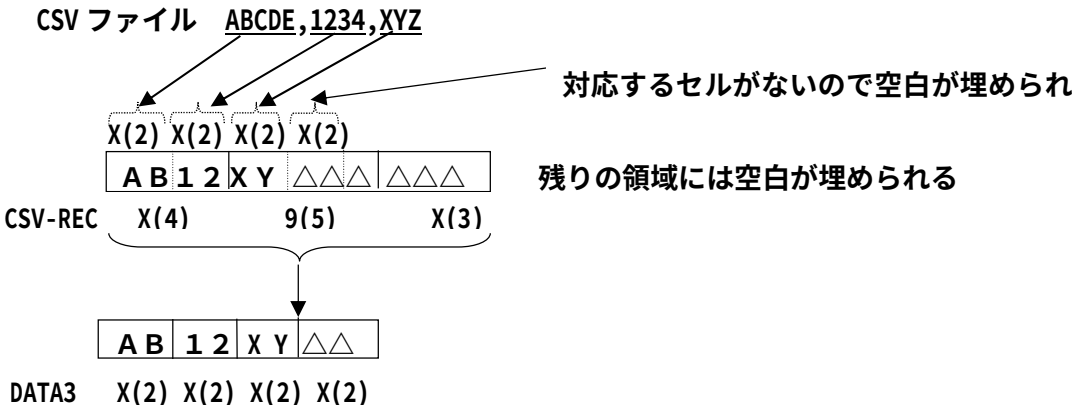
• READ CSV-FILE INTO DATA2

※ “△” は空白を表す



• READ CSV-FILE INTO DATA3

※ “△” は空白を表す



(3) データのないセルが含まれる行の READ 文の例

ここでは行順ファイル (CSV 形式) の 1 レコードのデータ値を以下とする。

CSV ファイルの 1 レコードの内容

,,XYZ

- READ CSV-FILE INTO DATA1

※ “△” は空白を表す

CSV ファイル

,

,XYZ

△	△	△	△	0	0	0	0	0	X	Y	Z

データのないセルを読み込むと
基本項目の属性にあわせて空白
または 0 が埋められる

CSV-REC

X(4)

9(5)

X(3)

(4) FROM 句なしの WRITE 文の例

- WRITE CSV-REC

CSV-REC

X(4)	9(5)	X(3)
ABCD	01234	XYZ

CSV ファイル “ABCD”, “01234”, “XYZ”

(5) FROM 句ありの WRITE 文の例

- WRITE CSV-REC FROM DATA1

A	B	C	D	E	F	1	2	3	4	5	X	Y	Z
X(6)						9(5)					X(3)		

CSV-REC

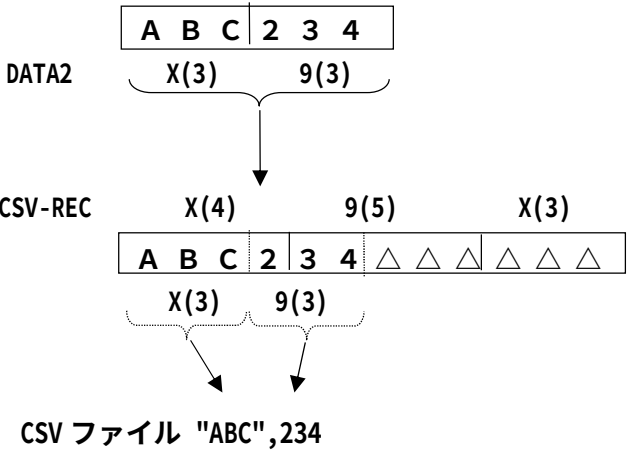
X(4)				9(5)					X(3)		
A	B	C	D	E	F	1	2	3	4	5	X
X(6)						X(5)			X(1)		

DATA1 の属性でセルに分けて書き出す
ただし、レコード長は CSV-REC のまま

CSV ファイル "ABCDEF", "12345", "X"

• WRITE CSV-REC FROM DATA2

※ “△” は空白を表す



DATA2 の属性でセルに分けて書き出す

3.8.4 行順ファイル(CSV 形式)の使用上の留意点

ここでは行順ファイル (CSV 形式) を使用する場合の注意事項を説明する。

1. セルと対応する基本項目の属性が日本語項目、または編集項目である場合、英数字項目と同様の入出力処理が行われる。

[Standard] [Enterprise for Windows] ただし、-CF オプションを指定すると、基本項目の属性が日本語項目、または日本語編集項目のとき、必要に応じて補う空白は日本語空白となる。

2. 基本項目の属性が外部 10 進項目のとき、対応するセルのデータの長さが 512 バイトを超えると、それ以降のデータは切り捨てられる。
3. 数字項目に不正文字が含まれている場合、実行時エラーを出力し、アプリケーションを強制終了する。
4. セルと対応する基本項目の属性が数字項目であり、PICTURE 句の文字列が文字 'P' を含むときの動作は次のようになる。

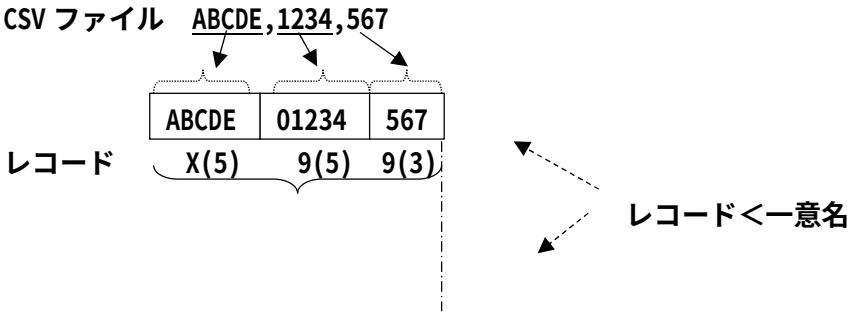
- (a) READ 文実行時は、小数点位置を合わせてセルから値が入力されるが、このとき PICTURE 文字列の文字 'P' の桁位置に対応する値は無視され、参照時にはゼロとして扱われる。

セルのデータ	PICTURE 句の記述	参照時の値
123	→ 9(2)P	120
0.123	→ P9(2)	0.023

- (b) WRITE 文実行時は、PICTURE 文字列の文字 'P' の桁位置に対応する値にゼロを補って出力される。

PICTURE 句の記述	参照時の値	セルのデータ
9(2)P	120	→ 120
P9(2)	0.023	→ 0.023

5. READ 文の INTO 句または WRITE 文の FROM 句の一意名長がレコード長よりも長く、一意名に従属する任意の基本項目の途中で入出力するデータがレコード長分に達する場合、その基本項目のデータは英数字項目として入出力される。



A B C D E	0 1 2 3 4	5 6 7 △△△	←----- 英数字項目として後方空白詰

一意名 X(5) 9(5) 9(6)

6. 区切り記号、空白およびダブルクォート(")は次のよう扱う。行順ファイル(CSV形式)で取り扱う入力ファイルを作成する場合や、出力ファイルを COBOL 以外のアプリケーションで利用する場合は、次の点に考慮が必要である。
- (a) 入力時に、1レコード中のデータの区切り記号の前後に空白がある場合、それぞれそのまま空白として扱う。空白詰めは行われない。
 - (b) 入力時に、CSV ファイルのセルの内容がダブルクォート(")で囲まれない数値以外のデータの場合、ダブルクォート(")で囲まれた場合と同じ値を CSV ファイルのセルと対応するデータ項目に読み込む。
 - (c) 入力時に、CSV ファイルのセルの先頭がダブルクォート(")の場合、対応する終わりを示すダブルクォート(")までを1つの CSV ファイルのセルとして処理する。終わりを示すダブルクォート(")より前に出現する区切り記号は、CSV ファイルのセルの値として扱う。
 - (d) 出力時に、CSV ファイルのセルと対応するデータ項目の値が数値以外の場合、データ項目の値をダブルクォート(")で囲んで CSV ファイルに出力する。
 - (e) 出力時に、CSV ファイルのセルと対応するデータ項目中にダブルクォート(")が存在する場合、ダブルクォート(")をそのまま CSV ファイルに出力する。

3.9 大容量記憶ファイル(相対ファイル)

相対ファイル(**relative file**) とはファイル中の特定のレコードを相対キーで示される相対キー番号によって順呼び出し, 乱呼び出しまたは動的呼び出しによって直接読み書きなどの処理ができるファイル編成である。

3.9.1 入出力文

相対ファイルで利用できる文には, 次の文がある。

- OPEN 文
- CLOSE 文
- READ 文
- WRITE 文
- REWRITE 文
- START 文
- DELETE 文
- USE 文(「3.12 入出力状態」を参照)

なお, これらの文はファイルの呼び出し法および OPEN 文のモードによって利用できる文が決まる。

以降に, 各文の使用方法および動作を説明する。

(1) OPEN 文(相対)

相対ファイルの OPEN 文は, ファイルを利用できる状態(**OPEN**)にする。ただし最初のレコードを読んだり書いたりしない。

また, OPEN 文のモードによって利用できる入出力文を次の表に示すように定める。

表 3-8 相対ファイルの OPEN 文のモードと入出力文の関係

OPEN 文の モード	入力 モード (INPUT)	出力 モード (OUTPUT)	入出力両用 モード (I-O)	拡張 モード (EXTEND)	説 明

文							
ファイルの呼び出し法	順呼び出し	READ 文	○	—	○	—	ファイルの最初から物理順にレコードを順次処理する。
		WRITE 文	—	○	—	○	
		REWRITE 文	—	—	○	—	
		START 文	○	—	○	—	
		DELETE 文	—	—	○	—	
	乱呼び出し	READ 文	○	—	○	—	キーを指定しレコードを乱呼び出しに処理する。
		WRITE 文	—	○	○	—	
		REWRITE 文	—	—	○	—	
		DELETE 文	—	—	○	—	
	動的呼び出し	READ 文	○	—	○	—	ファイル内のレコードを順次処理する。またキーを指定して乱呼び出しに処理する。
		WRITE 文	—	○	○	—	
		REWRITE 文	—	—	○	—	
		START 文	○	—	○	—	
		DELETE 文	—	—	○	—	
						最後のレコードの直後に位置付けた後レコードを書き出す。	
						ファイル内のレコードを置換, 削除および挿入する。	
						ファイルを作成する。	
						ファイル内の各レコードを読み込む。	

○:使用可能 —:使用不可

備考 -Qo オプション指定ありのとき, 大容量記憶ファイル(COBOL DB アクセス形式の索引ファイルを除く)の先行する入出力文で入出力状態 35 の永続誤りが発生した場合であっても, そのファイルに対して OUTPUT 指定の OPEN 文を実行できる。

-Qp オプション指定なしのとき, SHARED-MODE または PROTECTED-MODE 指定の相対ファイルに対して, ファイル共用機能は使用できない。

[Enterprise for Linux] 入力モード(INPUT)で開いた場合はファイル共用されるが, レコード排他は働かない。

(2) CLOSE 文(相対)

相対ファイルの CLOSE 文は標準ファイル閉じ (CLOSE) を実行する。OPEN したファイルは、必ず CLOSE しなければならない。LOCK 指定があれば、この実行単位ではファイルを再び開くことはできない。

[Standard] -QI オプション指定ありのとき、LOCK 指定をメモ扱いとする。

(3) READ 文(相対)

相対ファイルの READ 文は、入力または入出力両用モードで OPEN したファイルに使用する。その動作は呼び出し法によって異なる。たとえば、順呼び出しおよび動的呼び出しの順次処理においては、RELATIVE KEY データ項目の値に関係なく相対ファイルの順にレコードを読み込む。また乱呼び出しおよび動的呼び出しで任意の順序で処理する場合は、RELATIVE KEY データ項目には相対キー番号を設定し、指定した相対キー番号のレコードを読み込む。

例 ①順呼び出し

```

      :
      READ D-FILE AT END GO TO END-RTN.
      READ D-FILE AT END GO TO END-RTN.

```

②乱呼び出し

```

      :
      MOVE 35 TO REL-KEY.
      READ D-FILE INVALID KEY GO TO ERR-RTN.
      MOVE 10 TO REL-KEY.
      READ D-FILE INVALID KEY GO TO ERR-RTN.

```

③動的呼び出し

```

      :
      MOVE 30 TO REL-KEY.
      READ D-FILE INVALID KEY GO TO ERR-RTN.
      READ D-FILE NEXT AT END GO TO END-RTN.
      MOVE 25 TO REL-KEY.
      READ D-FILE INVALID KEY GO TO ERR-RTN.

```

(4) WRITE 文(相対)

相対ファイルの WRITE 文は、出力、入出力両用または拡張モードで OPEN したファイルに使用する。その動作は、呼び出し法および OPEN 文のモードによって異なる。たとえば出力モードの順呼び出しでは、レコードを順に出力し、出力モードの乱呼び出しでは、ランダムにレコードを出力する。入出力両用モードの乱呼び出しでは、レコードの挿入を行う。また、順呼び出し以外の場合は、RELATIVE KEY データ項目には、相対キー番号を設定しなければならない。

例 ①順呼び出し

```

      :
WRITE D-REC INVALID KEY GO TO ERR-RTN.
WRITE D-REC INVALID KEY GO TO ERR-RTN.

```

②乱呼び出し／動的呼び出し

```

      :
MOVE 20 TO REL-KEY.
WRITE D-REC INVALID KEY GO TO ERR-RTN.
MOVE 13 TO REL-KEY.
WRITE D-REC INVALID KEY GO TO ERR-RTN.

```

(5) REWRITE 文(相対)

相対ファイルの REWRITE 文は、入出力両用モードで OPEN したファイルにだけ使用する。その動作は呼び出し法によって異なる。たとえば順呼び出しでは、直前の成功した READ 文で読んだレコードを置換する。また、乱呼び出しおよび動的呼び出しの場合は、RELATIVE KEY データ項目で指定した相対キー番号のレコードを置換する。

(6) START 文(相対)

相対ファイルの START 文は、入力または入出力両用モードで OPEN されたファイルで、かつ順呼び出しおよび動的呼び出しの場合に使用できる。その動作は START 文の KEY 句で指定された条件を満足するレコードに位置付けする。

(7) DELETE 文(相対)

相対ファイルの DELETE 文は、入出力両用モードで OPEN されたファイルにだけ使用できる。その動作は呼び出し法によって異なる。

順呼び出しでは、直前に READ 文で読まれたレコードが削除され、また乱呼び出しおよび動的呼び出しの場合は、RELATIVE KEY データ項目で指定された相対キー番号のレコードが削除される。

3.9.2 相対ファイルの入出力文の動作

(1) READ 文

READ 文の書き方を次に示す。

書き方 1

- 動的呼び出しの場合

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]

- 順呼び出しの場合

上記の READ 文の書き方の NEXT 句を省略してもよい。

書き方 2

READ ファイル名 1 RECORD [INTO 一意名 1] [INVALID KEY 無条件文 1]

[NOT INVALID KEY 無条件文 2]

呼び出し法	READ 文の書き方	動作番号
順呼び出し	書き方 1	1
乱呼び出し	書き方 2	2
動的呼び出し	書き方 1	1
	書き方 2	2

(2) WRITE 文

呼び出し法	OPEN 文のモード	動作番号
順呼び出し	出力または拡張モード	3
乱／動的呼び出し	出力モード	4
	入出力両用モード	4

(3) REWRITE 文

呼び出し法	直前の入出力文	動作番号
順呼び出し	READ	5

	READ 以外	エラー
乱／動的呼び出し	-----	6

(4) START 文

呼び出し法	動作番号
順／動的呼び出し	7

(5) DELETE 文

呼び出し法	直前の入出力文	動作番号
順呼び出し	READ	8
	READ 以外	エラー
乱／動的呼び出し	-----	9

次に動作について説明する。なお、前表の 1～9 の動作番号が表 3-9 の番号 1～9 の動作番号に対応する。

(6) 各文の動作

表 3-9 大容量記憶ファイル(相対ファイル)に対する各文の動作

動作番号	動 作	状態キー
1	• RELATIVE KEY データ項目の値に関係なく、物理的な順序で次のレコードを読み込む (削除レコードは読み飛ばす)。 • 直前が START 文の場合には、位置づけされたレコードが読まれる。START 文が不成功の場合は、結果は保証されない。	00
	• RELATIVE KEY 句の指定があれば、読み込んだレコードの相対キー番号を RELATIVE KEY 句で指定されたデータ項目に格納する。相対キー番号がデータ項目に納まらないとき、状態キーには"14"を設定する。	14
	• すべてのレコードが処理されている場合には、AT END 条件となる。FILE STATUS 句の指定があると、状態キーには"10"を設定する。	10
2	• RELATIVE KEY データ項目で指定されたレコードが読まれる。	00

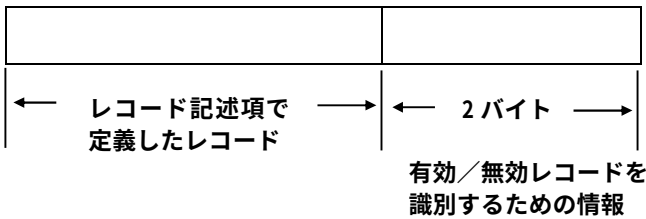
	RELATIVE KEY データ項目で指定されたレコードが、削除レコードなどの理由で見つからない場合には、INVALID KEY 条件となる。レコードが見つからない場合とは、指定のキーが最終レコードよりも大きいとき、または指定キーが削除レコードのときである。 FILE STATUS 句の指定があると、状態キーには"23"を設定する。	23
3	- RELATIVE KEY データ項目の値に関係なくレコードがファイルの最初から順次書き出し、書き出した順番に相対キー番号が 1, 2, 3, ..., n とつけられる。 - RELATIVE KEY 句の指定があれば、データ項目には、書き出されたレコードの相対キー番号を格納する。	00
	ファイルがいっぱいの場合は、ファイルオーバーフローとなる。 FILE STATUS 句の指定があると、状態キーには"24"を設定する。	24
4	RELATIVE KEY データ項目で指定された位置へレコードを書き出す。なお指定のキーが最終レコードより大きいキーのとき、最終レコードから指定キーまで削除レコードを書き、指定のキーにレコードを書き出す。	00
	RELATIVE KEY データ項目で指定したレコードがすでにファイル上に存在しているときは、INVALID KEY 条件となる。レコードがすでに存在している場合とは、指定キーが最終レコードよりも小さいときである。 FILE STATUS 句の指定があると、状態キーには"22"を設定する。	22
	RELATIVE KEY データ項目で指定した相対キー番号がファイルの外を指しているときは INVALID KEY 条件となる。 FILE STATUS 句の指定があると、状態キーには"24"を設定する。	24
5	RELATIVE KEY データ項目の値に関係なく、直前の READ 文で読まれたレコードが新しいレコードと置換される。	00
6	RELATIVE KEY データ項目で指定されたレコードが新しいレコードと置換される。	00
	指定されたレコードが見つからない場合には、INVALID KEY 条件となる。レコードが見つからない場合とは、指定キーが最終レコードよりも大きいとき、または指定キーが有効レコードでないときである。FILE STATUS 句の指定があると状態キーには"23"を設定する。	23
7	KEY 句で指定された条件を満足するレコードに位置づけされる。	00
	指定されたレコードが削除レコードなどの理由で見つからない場合には、INVALID KEY 条件となる。 FILE STATUS 句の指定があると、状態キーには"23"を設定する。	23
8	RELATIVE KEY データ項目の値に関係なく直前に読まれたレコードを論理的に削除する。	00

9	• RELATIVE KEY データ項目で指定されたレコードを削除する。	00
	• 指定されたレコードが見つからない場合には, INVALID KEY 条件となる。レコードが見つからない場合とは, 指定キーが最終レコードよりも大きいとき, または指定キーが有効レコードでないときである。 FILE STATUS 句の指定があると状態キーには"23"を設定する。	23

3.9.3 相対ファイルのファイル構成

【Enterprise】本 COBOL の相対ファイルは, 独自形式のファイル構成を持つ。相対ファイルのレコードは, 利用者がレコード記述項で定義したレコードの他に, 相対ファイルのレコードとして特別なフィールドを利用者宣言のレコードの後に付与してファイル上に格納する。このフィールドは, 相対レコードが有効レコードか無効レコードかを識別するための情報である。

ファイル上での相対レコードの構成



有効レコード:16 進 3131

無効レコード:16 進 3030

【Standard】本 COBOL の相対ファイルは, IFAS ファイルの場合は, IFAS ファイルのファイル構成を持つ。COBOL 独自形式の場合は, 独自形式のファイル構成を持つ。ただし, SE 動作モードは, 独自形式のファイル構成を持つ。

3.10 大容量記憶ファイル(索引ファイル)

3.10.1 入出力文

索引ファイルで利用できる文には, 次の文がある。

- OPEN 文
- CLOSE 文

- READ 文
- WRITE 文
- REWRITE 文
- START 文
- DELETE 文
- [Enterprise for Linux] UNLOCK 文
- USE 文(「3.12 入出力状態」を参照)

なお、これらの文はファイルの呼び出し法および OPEN 文のモードによって使用できるものがある。

以降に、各文の使用方法および動作を説明する。

(1) OPEN 文(索引)

索引ファイルの OPEN 文は、ファイルを利用できる状態にし、ファイル内の先頭にあるレコードに位置づける。しかし、最初のレコードを読んだり書いたりしない。

次に索引ファイルの OPEN 文のモードおよび呼び出し法によって使用できる入出力文を定める。

表 3-10 索引ファイルの OPEN 文のモードと入出力文の関係

OPEN 文の モード			入力 モード (INPUT)	出力 モード (OUTPUT)	入出力両用 モード (I-O)	拡張 モード (EXTEND)	説 明
文							
フ ア イ ル の 呼 び 出 し 法	順 呼 び 出 し	READ	○	—	○	—	ファイル内のレコードをキーの順番に順次処理をする。
		WRITE	—	○	—	○	
		REWRITE	—	—	○	—	
		START	○	—	○	—	
		DELETE	—	—	○	—	
		UNLOCK	○	○	○	○	
	乱 呼 び 出 し	READ	○	—	○	—	参照キーを指定して、ファイル内のレコードを乱呼び出しにて処理する。
		WRITE	—	○	○	—	
		REWRITE	—	—	○	—	
		DELETE	—	—	○	—	
		UNLOCK	○	○	○	○	
	動 的 呼 び 出 し	READ	○	—	○	—	ファイル内のレコードを順番に順次処理する。または参照キーを指定して乱呼び出しに処理する。
		WRITE	—	○	○	—	
		REWRITE	—	—	○	—	
		START	○	—	○	—	
		DELETE	—	—	○	—	
		UNLOCK	○	○	○	○	
							レコードを追加して書き出す。
							ファイル内のレコードを置換, 削除および挿入する。
							ファイルを作成する。
			ファイル内の各レコードをキーに従って読み込む。				

○:使用可能 ―:使用不可

備考 -Qo オプション指定ありのとき、大容量記憶ファイル(COBOL DB アクセス形式の索引ファイルを除く)の先行する入出力文で入出力状態 35 の永続誤りが発生した場合であっても、そのファイルに対して OUTPUT 指定の OPEN 文を実行できる。

[Enterprise for Linux]

OPEN 文の WITH LOCK 句は排他で開くことを明に指定する。

この句が指定されていない場合の共用／排他はオープンモードにより決まる。

ただし、ファイル管理記述項に LOCK MODE 句を指定することにより、WITH LOCK 句の指定がないときの共用／排他の既定値を変更することが可能である。

WITH LOCK 句指定なし時のファイルの共用／排他

オープンモード	既定値	備考
INPUT	共用オープン(注 1)	LOCK MODE 句で排他(EXCLUSIVE)オープンに変更されない
I-O	排他オープン	LOCK MODE 句で共用オープン(注 2)に変更可能
OUTPUT	排他オープン	—
EXTEND	排他オープン	LOCK MODE 句で共用オープン(注 2)に変更可能

注意 注(1)レコードロックは発生しないのでレコードロック制御不要

注(2)レコードロック制御は自動的(AUTOMATIC)に行われるモードと手動(MANUAL)で行うモードを選択可能

(2) CLOSE 文(索引)

索引ファイルの CLOSE 文は標準ファイル閉じ(CLOSE)を実行する。OPEN したファイルは、必ず CLOSE しなければならない。[Enterprise for Linux] また、そのときに CLOSE 対象のファイルが保持しているレコードロックはすべて解放される。

LOCK 指定があれば、この実行単位ではファイルを再び開くことはできない。

[Standard] -Ql オプション指定ありのとき、LOCK 指定をメモ扱いとする。

(3) READ 文(索引)

索引ファイルの READ 文は、入力(INPUT)または入出力両用モード(I-O)で OPEN したファイルに使用する。その動作は呼び出し法によって異なる。たとえば、順呼び出しおよび動的呼び出しの順次処理では、参照キーの値に関係なく索引ファイルの順にレコードを読み込む。乱呼び出しおよび動的呼び出しでかつ任意の順序で処理する場合は、参照キーを与えなければならない。

次に READ 文の例を示す。

例 ①順呼び出し


```
      :  
      READ IS-FILE AT END GO TO END-RTN.
```

②乱呼び出し

```
      :  
      MOVE A TO IDX-KEY.  
      READ IS-FILE INVALID KEY GO TO ERR-RTN.  
      MOVE B TO IDX-KEY.  
      READ IS-FILE INVALID KEY GO TO ERR-RTN.
```

③動的呼び出し

```
      :  
      MOVE C TO IDX-KEY.  
      READ IS-FILE INVALID KEY GO TO ERR-RTN.  
      READ IS-FILE NEXT AT END GO TO END-RTN.  
      MOVE D TO IDX-KEY.  
      READ IS-FILE INVALID KEY GO TO ERR-RTN.
```

[Enterprise for Linux] WITH LOCK 句および WITH NO LOCK 句は読み込んだレコードをロックするか否かを明に指定する。実際にレコードロックが発生するかどうかはファイルの共用／排他およびオープンモードによって決まる。

オープンモードと WITH LOCK 句, WITH NO LOCK 句の指定有無の組み合わせ

オープンモード		WITH LOCK 指定	WITH NO LOCK 指定	WITH 句なし
INPUT	共用モード(注 1)	Non locked	Non locked	Non locked
	排他モード(注 2)	Non locked	Non locked	Non locked
I-O	共用モード(注 3)			
	AUTOMATIC	locked	Non locked	locked
	MANUAL	locked	Non locked	Non locked
	排他モード(注 2)	Non locked	Non locked	Non locked

注意 注(1)ファイル管理記述項で LOCK MODE 句指定なしまたは AUTOMATIC/MANUAL が指定されており, WITH LOCK 句の指定がない OPEN 文でオープンされたファイルの場合

注(2)ファイル管理記述項の LOCK MODE 句指定で EXCLUSIVE が指定されている, または WITH LOCK 句指定の OPEN 文でオープンされているファイルの場合

注(3)ファイル管理記述項で LOCK MODE 句で AUTOMATIC/MANUAL が指定されており, WITH LOCK 句の指定がない OPEN 文でオープンされたファイルの場合

(4) WRITE 文(索引)

索引ファイルの WRITE 文は, 出力, 入出力両用または拡張モードで OPEN されたファイルに使用できる。その動作は, OPEN 文のモードによって異なり, 出力モードでは新しくデータを作成し, 入出力両用モードではレコードの挿入を行う。また参照キーは必ず指定しなければならない。

[Enterprise for Linux] ファイル管理記述項の LOCK MODE 句で AUTOMATIC が指定されているファイルを共用している場合, WRITE 文はそのときに保持しているレコードロックを解放する。ファイル管理記述項の LOCK MODE 句で MANUAL が指定されている場合には, WRITE 文はいずれかのレコードロックも解放しない。

次に WRITE 文の例を示す。

例 ①出力モード(順, 乱, 動的)

```

      :
MOVE  E  TO  REL-KEY.
WRITE IS-REC INVALID KEY GO TO ERR-RTN.
MOVE  F  TO  REL-KEY.

```

```
WRITE IS-REC INVALID KEY GO TO ERR-RTN.
```

②入出力両用モード(乱, 動的)

```

      :
      MOVE G TO REL-KEY.
      WRITE IS-REC INVALID KEY GO TO ERR-RTN.
      MOVE H TO REL-KEY.
      WRITE IS-REC INVALID KEY GO TO ERR-RTN.
```

(5) REWRITE 文(索引)

索引ファイルの REWRITE 文は, 入出力両用モード(I-O)で OPEN されたファイルについてだけ使用でき, その動作は呼び出し法によって異なる。たとえば順呼び出しでは, 直前に READ 文で読まれたレコードを新しいレコードに置換する。また, 乱呼び出しおよび動的呼び出しの場合は, 参照キーで指定されたレコードを置換する。

[Enterprise for Linux] ファイル管理記述項の LOCK MODE 句で AUTOMATIC が指定されているファイルを共用している場合, REWRITE 文はそのときに保持しているレコードロックを解放する。ファイル管理記述項の LOCK MODE 句で MANUAL が指定されている場合には REWRITE 文はいずれのレコードロックも解放しない。

(6) START 文(索引)

索引ファイルの START 文は, 入力(INPUT)または入出力両用モード(I-O)で OPEN されたファイルでかつ順呼び出しまたは動的呼び出しの場合に使用できる。索引ファイルの START 文は, 参照キーで指定したレコードに位置付けする。

[Enterprise for Linux] START 文はいずれのレコードロックも解放しない。

(7) DELETE 文(索引)

索引ファイルの DELETE 文は, 入出力両用モード(I-O)で OPEN されたファイルにだけ使用でき, その動作は呼び出し法によって異なる。

順呼び出しでは, 直前に READ 文で読まれた参照キーのレコードを削除し, 乱呼び出しおよび動的呼び出しの場合は, 参照キーで指定されたレコードを削除する。

[Enterprise for Linux] ファイル管理記述項の LOCK MODE 句で AUTOMATIC が指定されているファイルを共用している場合, DELETE 文はそのときに保持しているレコードロックを解放する。ファイル管理記述項の LOCK MODE 句で MANUAL が指定されている場合には, DELETE 文はいずれのレコードロックも解放しない。ただし, 削除されたレコードに関連するレコードロックは解放される。

(8) [Enterprise for Linux] UNLOCK 文(索引)

UNLOCK 文は、UNLOCK 文対象のファイルに関するレコードロックをすべて解放する。
たとえ、レコードロックがなくても成功する。

3.10.2 索引ファイルの入出力文の動作

(1) READ 文

READ 文の書き方を次に示す。

書き方 1

- 動的呼び出しの場合

READ ファイル名 1 [NEXT] RECORD [INTO 一意名 1]

[WITH { [NO] LOCK
UNLOCK }]

[AT END 無条件文 1]

[NOT AT END 無条件文 2]

- 順呼び出しの場合

上記の READ 文の書き方の NEXT 句を省略してもよい。

書き方 2

READ ファイル名 1 RECORD [INTO 一意名 1]

[KEY IS データ名]

[WITH { [NO] LOCK
UNLOCK }]

[INVALID KEY 無条件文 1]

[NOT INVALID KEY 無条件文 2]

呼び出し法	READ 文の書き方	動作番号
順呼び出し	書き方 1	1
乱呼び出し	書き方 2	2
動的呼び出し	書き方 1	1
	書き方 2	2

(2) WRITE 文

呼び出し法	OPEN 文のモード	動作番号
順呼び出し	出力または拡張モード	3
乱／動的呼び出し	出力モード	4
	入出力両用モード	5

(3) REWRITE 文

呼び出し法	直前の入出力文	動作番号
順呼び出し	READ	6
	READ 以外	エラー
乱／動的呼び出し	-----	7

(4) START 文

呼び出し法	動作番号
順／動的呼び出し	8

(5) DELETE 文

呼び出し法	直前の入出力文	動作番号
順呼び出し	READ	9
	READ 以外	エラー
乱／動的呼び出し	-----	10

(6) [Enterprise for Linux] UNLOCK 文

呼び出し法	動作番号
順／乱／動的呼び出し	11

次に動作について説明する。なお、(1)READ 文、(2)WRITE 文、(3)REWRITE 文、(4)START 文、(5)DELETE 文、(6)UNLOCK 文の各表の中にある動作番号 1～11 が、表 3-11 の番号 1～11 の動作番号に対応する。

(7) 各文の動作

表 3-11 大容量記憶ファイル(索引ファイル)に対する各文の動作

動作番号	動 作	状態キー
1	- 参照キーの値に関係なく直前に読まれたレコードより大きいキーをもつレコードを読み込む。 - 直前が START 文の場合には、位置づけされたレコードを読み込む。START 文が不成功の場合は、結果は保証されない。	00
	副レコードをもつ索引ファイルのとき読み込んだ副レコードキーと等しい参照キーをもつレコードが続いている。	02
	すべてのレコードが処理されている場合には、AT END 条件となる。FILE STATUS 句の指定があると、状態キーには"10"を設定する。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE 句が AUTOMATIC の共用モードで動作しているファイルの場合には、WITH NO LOCK 句指定でないかぎり読み込んだレコードを自動的にロックする。入出力両用モードで開かれた LOCK MODE 句が MANUAL の共用モードで動作しているファイルの場合には、WITH LOCK 句指定のときのみ読み込んだレコードをロックする。	10
	[Enterprise for Linux] 他の実行単位により読み込み対象のレコードがロックされている場合はその他の誤りとなる。FILE STATUS 句の指定があると、状態キーには '92' を設定する。	92
2	参照キーで指定されたキーをもつレコードを読み込む。	00
	副レコードを持つ索引ファイルのとき読み込んだ副レコードキーと等しい参照キーを持つレコードが続いている。	02
	指定されたキーをもつレコードが見つからない場合は、INVALID KEY 条件になる。FILE STATUS 句の指定があると状態キーには"23"を設定する。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE 句が AUTOMATIC の共用モードで動作しているファイルの場合には、WITH NO LOCK 句指定でないかぎり読み込んだレコードを自動的にロックする。入出力両用モードで開かれた LOCK MODE 句が MANUAL の共用モードで動作しているファイルの場合には、WITH LOCK 句指定のときのみ読み込んだレコードをロックする。	23
	[Enterprise for Linux] 他の実行単位により読み込み対象のレコードがロックされている場合はその他の誤りとなる。FILE STATUS 句の指定があると、状態キーには '92' を設定する。	92

3	- 出力モードのときレコードをファイルの最初から順番に書き出す。参照キーの指定が必要であり参照キーの値は昇順に与えなければならない。 - 拡張モードのときファイルにレコードを追加する。参照キーの値はファイル中にある最大の参照キーの値より大きくなければならない。	00
	WRITE 文で書き出したレコードの副レコードキーの値と同じ参照キーの値を持つレコードが、すでにファイルに存在している。	02
	指定レコードのキーが昇順となっていない場合は、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"21"を設定する。	21
	二重キーをエラーとするファイルにおいては、直前と同じキーを WRITE しようとした場合は、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"22"を設定する。	22
	ファイルがオーバーフローすると INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"24"を設定する。 この状態が起きた場合、ファイルを CLOSE しファイルを作り直す必要がある。	24
4	レコードをファイルの先頭から順番に書き出す。参照キーの順番はどんな順番でもよい。	00
	WRITE 文で書き出したレコードの副レコードキーの値と同じ参照キーの値を持つレコードが、すでにファイルに存在している。	02
	二重キーをエラーとするファイルにおいては、直前と同じキーを WRITE しようとした場合は、INVALID KEY 条件となる。	22
	ファイルがオーバーフローすると INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"24"を設定する。 この状態が起きた場合、ファイルを CLOSE しファイルを作り直す必要がある。	24
5	参照キーで指定した位置にレコードを追加する。	00
	WRITE 文で書き出したレコードの副レコードキーの値と同じ参照キーの値を持つレコードが、すでにファイルに存在している。	
	二重キーをエラーとするファイルにおいては、直前と同じキーを WRITE しようとした場合は、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"22"を設定する。	22

	ファイルがオーバーフローすると INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"24"を設定する。 この状態が起きた場合、ファイルを CLOSE しファイルを作り直す必要がある。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE が AUTOMATIC の共用モードで動作しているファイの場合で自動的にロックされたレコードが存在する場合には、そのレコードは解放される。	24
6	直前の READ 文で読み込んだレコードを新しいレコードで置換する。	00
	REWRITE 文で置換したレコードの副レコードキーの値と同じ参照キーの値を持つレコードが、すでにファイルに存在している。	02
	指定されたキーを持つレコードが見つからない場合には、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"21"を設定する。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE が AUTOMATIC の共用モードで動作しているファイの場合で自動的にロックされたレコードが存在する場合には、そのレコードは解放される。	21
7	参照キーで指定されたキーをもつレコードを新しいレコードで置換する。	00
	REWRITE 文で置換したレコードの副レコードキーの値と同じ参照キーの値を持つレコードが、すでにファイルに存在している。	02
	指定されたキーを持つレコードが見つからない場合には、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"23"を設定する。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE が AUTOMATIC の共用モードで動作しているファイの場合で自動的にロックされたレコードが存在する場合には、そのレコードは解放される。	23
8	KEY 句で指定された条件を満足するレコードに位置づける。	00
	指定されたキーを持つレコードが見つからない場合には、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"23"を設定する。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE が AUTOMATIC の共用モードで動作しているファイの場合で自動的にロックされたレコードが存在する場合には、そのレコードは解放される。	23
	[Enterprise for Linux] 他の実行単位でレコードがロックされている場合、状態キーには"92"を設定する。	92

9	参照キーの値に関係なく直前に読み込んだレコードを論理的に削除する。 [Enterprise for Linux] 入出力両用モードで開かれた共用モードで動作している場合には、LOCK MODE 句で AUTOMATIC/MANUAL いずれの場合もそのときロックされているレコードロックを解放する。	00
10	参照キーで指定されたキーをもつレコードを削除する。	00
	指定されたキーをもつレコードが見つからない場合には、INVALID KEY 条件となる。FILE STATUS 句の指定があると状態キーには"23"を設定する。 [Enterprise for Linux] 入出力両用モードで開かれた LOCK MODE が AUTOMATIC の共用モードで動作しているファイルの場合で自動的にロックされたレコードが存在する場合には、そのレコードは解放される。削除されたレコードに関するレコードロックは無条件に解放される。	23
11	[Enterprise for Linux] そのときにロックされているレコードロックを解放する。レコードロックが実際に存在しない場合も成功する。	00
	[Enterprise for Linux] OPEN されていないファイルに対して UNLOCK 文が実行された。	93

3.10.3 二重キー

[Enterprise for Windows] [Standard] 大容量記憶ファイル(索引ファイル)の二重キーについて、以下に説明する。

索引ファイルにおいて、二重キーは次のように扱われる。

本システムでは、索引ファイルの副レコードキーにかぎり、ファイル内に二重キーをもつレコードを存在させることができる。二重キーをもつレコードの存在許可または禁止は、ファイルを COBOL アプリケーションで作成するときの ALTERNATE RECORD KEY 句(WITH DUPLICATES 指定)で指定する。

また、作成時、索引ファイルの指定とアクセス時のソースプログラム上の記述(WITH DUPLICATES 句の有無)は、同じにしなければならない。

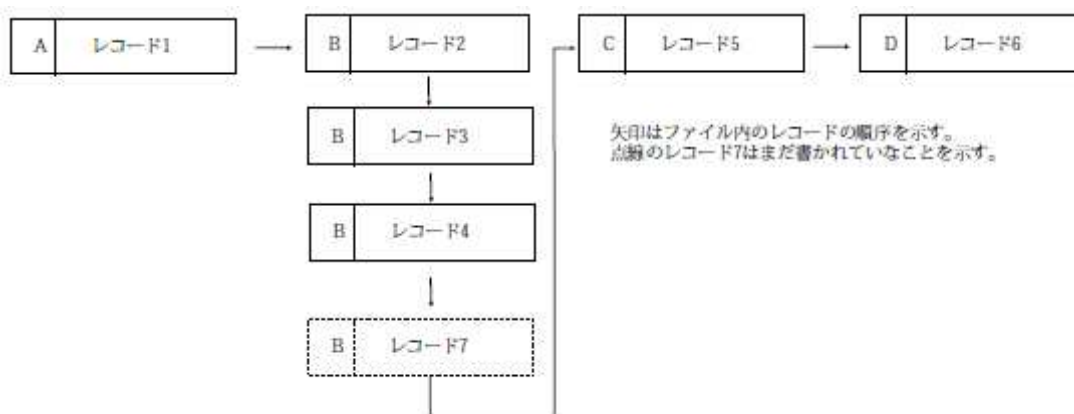
ファイル内に二重キーの存在を許可したときの手続き文と動作の関係を次の表に示す。

表 索引ファイルにおける二重キーの取り扱い

手続き文	動 作
書き方1の READ 文	現在読み込まれているレコードの次のレコードまたは位置づけられているレコードが読み込まれる。
書き方2の READ 文	二重キーをもついくつかのレコードのうち、最初のレコードが読み込まれる。
WRITE 文	二重キーをもつレコードは前に書かれた同じキーの値をもつレコードの最後に書き出される。
REWRITE 文	成功した READ 文の直後のときはその文によって読み込まれたレコードが置換される。 直前が成功した READ 文でないとき、または、キーを直前の READ 文の値から変更したときは二重キーをもついくつかのレコードのうち、最初のレコードが置換される。
DELETE 文	成功した READ 文の直後のときはその文によって読み込まれたレコードが削除される。 直前が成功した READ 文でないとき、または、キーを直前の READ 文の値から変更したときは二重キーをもついくつかのレコードのうち、最初のレコードが削除される。
START 文	二重キーをもついくつかのレコードのうち、最初のレコードに位置付けられる。

また、ファイル内の二重キーの存在を許可したときの手続き文と動作例を示す。

(例) ファイル内のレコード



上のようなレコードが記録されたファイルに対し、動的呼び出し法で次の動作を行ったとする。

(1) MOVE "B" TO R-KEY.

START MFILE KEY EQUAL R-KEY INVALID ～. … レコード2に位置付けられる。

READ MFILE NEXT AT END ～. … レコード2が読み込まれる。

READ MFILE NEXT AT END ～. … レコード3が読み込まれる。

- (2) MOVE "B" TO R-KEY.
 READ MFILE INVALID ～. レコード2が読み込まれる。
- (3) MOVE "B" TO R-KEY.
 REWRITE MREC INVALID ～. レコード2が置換される。
- (4) MOVE "B" TO R-KEY.
 DELETE MFILE INVALID ～. レコード2が削除される。
- (5) MOVE "B" TO R-KEY.
 WRITE MREC INVALID ～. レコード7が書き込まれる。
- (6) MOVE "B" TO R-KEY.
 READ MFILE INVALID ～. レコード2が読み込まれる。
 READ MFILE NEXT AT END ～. レコード3が読み込まれる。
 REWRITE MREC INVALID ～. レコード3が置換される。
 READ MFILE NEXT AT END ～. レコード4が読み込まれる。
 DELETE MFILE INVALID ～. レコード4が削除される。
 また、呼び出し法で次の動作を行ったとする。
- (7) MOVE "B" TO R-KEY.
 START MFILE KEY EQUAL R-KEY INVALID ～. ... レコード2が位置付けられる。
 READ MFILE INVALID ～. レコード2が読み込まれる。
 READ MFILE NEXT AT END ～. レコード3が読み込まれる。
 REWRITE MREC INVALID ～. レコード3が置換される。
 READ MFILE NEXT AT END ～. レコード4が読み込まれる。
 DELETE MFILE INVALID ～. レコード4が削除される。

3.10.4 索引ファイルのファイル構成および使用上の留意点

[Enterprise] 本 COBOL の索引ファイルは、ISAM を使用しているために ISAM のファイル構成をもつ。また、使用する場合は、下記の留意点を守る必要がある。

- ・ 索引ファイルを出力モードで作成すると、レコードを格納するデータファイルとレコードを検索するためのキーを格納するキー定義ファイルとキーファイルを作成する。

- ・ データファイルにはファイル名の後に".DAT"を付与し、キー定義ファイルにはファイル名の後に".DEF"を付与して作成される。キーファイルはキーの種類により拡張子が変わる。主キーファイルにはファイル名の後に".K01"を付与して作成される。副キーファイルにはファイル名の後ろに".Kxx" を付与して作成される。xx は副キーが定義された数により、02～99 の数字となる。

[Standard] 本 COBOL の索引ファイルは、IFAS ファイルの場合は、IFAS ファイルのファイル構成を持つ。ISAM 形式の場合は、ISAM 形式のファイル構成を持つ。ただし、SE 動作モードは、ISAM を

使用しているために ISAM のファイル構成を持つ。いずれも ISAM 形式の場合は、Enterprise Edition 版の留意点を参照。

レコード件数 0 の ISAM 形式の索引ファイルに対する OPEN EXTEND 文は、入出力状態 30 のファイル入出力エラーになるので注意が必要である。

3.11 APPLY 句によるファイルの共用／排他(互換機能)

APPLY 句によるファイルの共用／排他を次のように制御する。

[Enterprise] -MT オプション指定ありの場合、ファイルの排他／共用はサポートしない。

(1) OPEN 時のチェック

[Enterprise for Windows] 順ファイル、行順ファイル、相対ファイルおよび索引ファイルのファイル共用／排他制御は、OPEN 文のオープンモードと APPLY 句の入出力技法により OPEN 時にチェックする。

(a) 順ファイル(COBOL 独自形式),相対ファイル(COBOL 独自形式)

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用 モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
	APPLY 句 入出力技法	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、標準動作モードでは未サポート。

(b) 索引ファイル(ISAM 形式)

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用 モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
	OPEN 文 のモード	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	×
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	×
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、標準動作モードでは未サポート。

(c) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp あり かつ -CF/-CU/-Cj オプションなし

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
	OPEN 文 のモード	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード	EXCLUSIVE	×	×	×	×	×	×	×	×

(INPUT)	SHARE D	×	○	×	◎	×	×	×	○
入出力 両用モ ード (I-O)	EXCLU SIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	◎	×	×	×	○
出力モ ード (OUTPUT)	EXCLUS IVE	×	×	×	×	×	×	×	×
	SHARE D	×	×	×	×	×	×	×	×
拡張モ ード (EXTEN D)	EXCLUSI VE	×	×	×	×	×	×	×	×
	SHA RED	×	○	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

◎:ファイルは共用され、レコード排他が働く。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、標準動作モードでは未サポート。

(d) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp あり かつ -CF/-CU/-Cj オプションあり

先 行 後行	OPEN 文 のモー ド	入力モード (INPUT)		入出力両用モー ド (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
OPEN 文 のモー ド	APPLY 句 入出力 技法	EXC LUSIVE	SHARED	EXCLUS IVE	SHAR ED	EXCLUSIVE	SHA R ED	EXCL USIVE	SHARED
入力モ ード (INPUT)	EXCLUSI VE	×	×	×	×	×	×	×	×
	SHARE D	×	○	×	○	×	×	×	○
入出力 両用モ ード (I-O)	EXCLU SIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○
出力モ ード (OUTPU T)	EXCLUS IVE	×	×	×	×	×	×	×	×
	SHARE D	×	×	×	×	×	×	×	×
拡張モ ード (EXTEN D)	EXCLUSI VE	×	×	×	×	×	×	×	×
	SHA RED	×	○	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、標準動作モードでは未サポート。

(e) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp なし

先行 後行	OPEN 文 のモー ド	入力モード (INPUT)		入出力両用モー ド (I-0)		出力モード (OUTPUT)		拡張モード (EXTEND)	
OPEN 文 のモー ド	APPLY 句 入出力 技法	EXC LUSIVE	SHARED	EXCLUS IVE	SHAR ED	EXCLUSIVE	SHA R ED	EXCL USIVE	SHARED
入力モー ード (INPUT)	EXCLUSI VE	×	×	×	×	×	×	×	×
	SHARE D	×	×	×	×	×	×	×	×
入出力 両用モー ード (I-0)	EXCLU SIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
出力モー ード (OUTPU T)	EXCLUS IVE	×	×	×	×	×	×	×	×
	SHARE D	×	×	×	×	×	×	×	×
拡張モー ード (EXTEN D)	EXCLUSI VE	×	×	×	×	×	×	×	×
	SHA RED	×	×	×	×	×	×	×	×

備考(1) ×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、標準動作モードでは未サポート。

(f) 行順ファイル(CSV あり)(COBOL 独自形式)

先 行 後行	OPEN 文 のモード	入力モード (INPUT)		出力モード (OUTPUT)		拡張モード (EXTEND)	
	APPLY 句 入出力技 法	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力 モー ド (INPUT)	EXCLUSIVE	×	×	×	×	×	×
	SHARED	×	○	×	×	×	○
出力モー ド (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×
拡張モー ド (EXTEND)	EXCLUSIVE	×	×	×	×	×	×
	SHARED	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、標準動作モードでは未サポート。

備考(3) 行順ファイル(CSV あり)は I-O オープンできない。

【Enterprise for Linux】索引ファイル、行順ファイルのファイル共用／排他制御は、OPEN 文のオープンモードと APPLY 句の入出力技法により OPEN 時にチェックする。ただし、順ファイル、相対ファイルは共用されない。入力モード (INPUT) で開いた順ファイル、相対ファイルはファイルは共用されるが、レコード排他は働かない。

(a) 索引ファイル (ISAM 形式)

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用 モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
		EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
OPEN 文 のモード	APPLY 句 入出力技 法								
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	◎	×	◎	×	×	×	×
入出力両用 モード (I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	◎	×	◎	×	×	×	×
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×

備考 ◎: ファイルは共用され、レコード排他が働く。

×: ファイルは共用されない。後行 OPEN 文が異常となる (状態キーは 91 または 30)。

(b) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp あり かつ -CU/-Cj オプションなし

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用 モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
OPEN 文 のモード	APPLY 句 入出力技 法	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	◎	×	×	×	○
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	◎	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

◎:ファイルは共用され、レコード排他が働く。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、未サポート。

(c) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp あり かつ -CU/-Cj オプションあり

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用 モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
OPEN 文 のモード	APPLY 句 入出力技 法	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	○	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、未サポート。

(d) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp なし

先行 後行	OPEN 文 のモード	入力モード (INPUT)		入出力両用 モード (I-O)		出力モード (OUTPUT)		拡張モード (EXTEND)	
	OPEN 文 のモード	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード (INPUT)	EXCLUSIVE	○	○	×	×	×	×	×	×
	SHARED	○	○	×	×	×	×	×	×
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、未サポート。

(e) 行順ファイル(CSV あり)(COBOL 独自形式) かつ -Qq あり

先行 後行	OPEN 文 のモード	入力モード (INPUT)		出力モード (OUTPUT)		拡張モード (EXTEND)	
	OPEN 文 のモード	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED	EXCLUSIVE	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×
	SHARED	×	○	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×
	SHARED	×	○	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、未サポート。

備考(3) 行順ファイル(CSV あり)は I-O オープンできない。

【Standard】ファイル共用／排他制御は、OPEN 文のオープンモードと APPLY 句の入出力技法により次のファイルの OPEN 時にチェックする。

(a) 順ファイル(COBOL 独自形式), 相対ファイル(COBOL 独自形式), 行順ファイル(CSV なし)(COBOL 独自形式) -Qp あり かつ -CF/-Cj オプションあり

先行 後行	OPEN 文 のモード	入力モード (INPUT)			入出力両用モード (I-O)			出力モード (OUTPUT)			拡張モード (EXTEND)		
	OPEN 文 のモード	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	○	○	×	×	×	×	×	×	×	×	×
	SHARED	×	○	○	×	○	○	×	×	×	×	○	○
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	○	×	×	×	×	×	×	×	×	×
	SHARED	×	×	○	×	×	○	×	×	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	○	×	×	×	×	×	×	×	×	×
	SHARED	×	×	○	×	×	○	×	×	×	×	×	○

備考 ○: ファイルは共用されるが、レコード排他は働かない。

×: ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

(b) 順ファイル(IFAS FILE 形式),行順ファイル(CSV なし)(COBOL 独自形式) -Qp あり かつ
 -CF/-Cj オプションなし,相対ファイル(IFAS FILE 形式),索引ファイル(IFAS FILE 形式, COBOL
 DB アクセス形式)

先行 後行	OPEN 文 のモード	入力モード (INPUT)			入出力両用モード (I-O)			出力モード (OUTPUT)			拡張モード (EXTEND)		
	OPEN 文 のモード	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	○	○	×	×	×	×	×	×	×	×	×
	SHARED	×	○	○	×	◎	◎	×	×	×	×	○	○
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	○	×	×	×	×	×	×	×	×	×
	SHARED	×	×	○	×	×	◎	×	×	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	○	×	×	×	×	×	×	×	×	×
	SHARED	×	×	○	×	×	○	×	×	×	×	×	○

備考 ◎:ファイルは共用され,レコード排他が働く。

○:ファイルは共用されるが,レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

(c) 行順ファイル(CSV なし)(COBOL 独自形式) -Qp なし

先行 後行	OPEN 文 のモード	入力モード (INPUT)			入出力両用モード (I-O)			出力モード (OUTPUT)			拡張モード (EXTEND)		
	OPEN 文 のモード	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×

備考 ×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

(d) 索引ファイル(ISAM 形式)

先行 後行	OPEN 文 のモード	入力モード (INPUT)			入出力両用モード (I-O)			出力モード (OUTPUT)			拡張モード (EXTEND)		
	OPEN 文 のモード	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	○	○	×	○	○	×	×	×	×	×	×
	SHARED	×	○	○	×	○	○	×	×	×	×	×	×
入出力両用 モード(I-O)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	○	○	×	○	○	×	×	×	×	×	×
	SHARED	×	○	○	×	○	○	×	×	×	×	×	×
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×	×	×	×

備考 ○: ファイルは共用されるが, レコード排他は働かない。

×: ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

(e) 行順ファイル(CSV あり)(COBOL 独自形式)

先行 後行	OPEN 文 のモード	入力モード (INPUT)			出力モード (OUTPUT)			拡張モード (EXTEND)		
	OPEN 文 のモード	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED	EXCLUSIVE	PROTECTED	SHARED
入力モード (INPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×
	SHARED	×	×	○	×	×	×	×	×	○
出力モード (OUTPUT)	EXCLUSIVE	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×
	SHARED	×	×	×	×	×	×	×	×	×
拡張モード (EXTEND)	EXCLUSIVE	×	×	×	×	×	×	×	×	×
	PROTECTED	×	×	×	×	×	×	×	×	×
	SHARED	×	×	○	×	×	×	×	×	○

備考(1) ○:ファイルは共用されるが、レコード排他は働かない。

×:ファイルは共用されない。後行 OPEN 文が異常となる(状態キーは 91)。

備考(2) PROTECTED-MODE 指定は、ファイル共用/排他は対応しない。

備考(3) 行順ファイル(CSV あり)は I-O オープンできない。

(2) レコード排他制御

ファイルを共用したときは、ファイル内の同一レコードの共用要求がある場合に矛盾なくそれらの処理を完結させるために、レコードの排他制御を次のように行う。

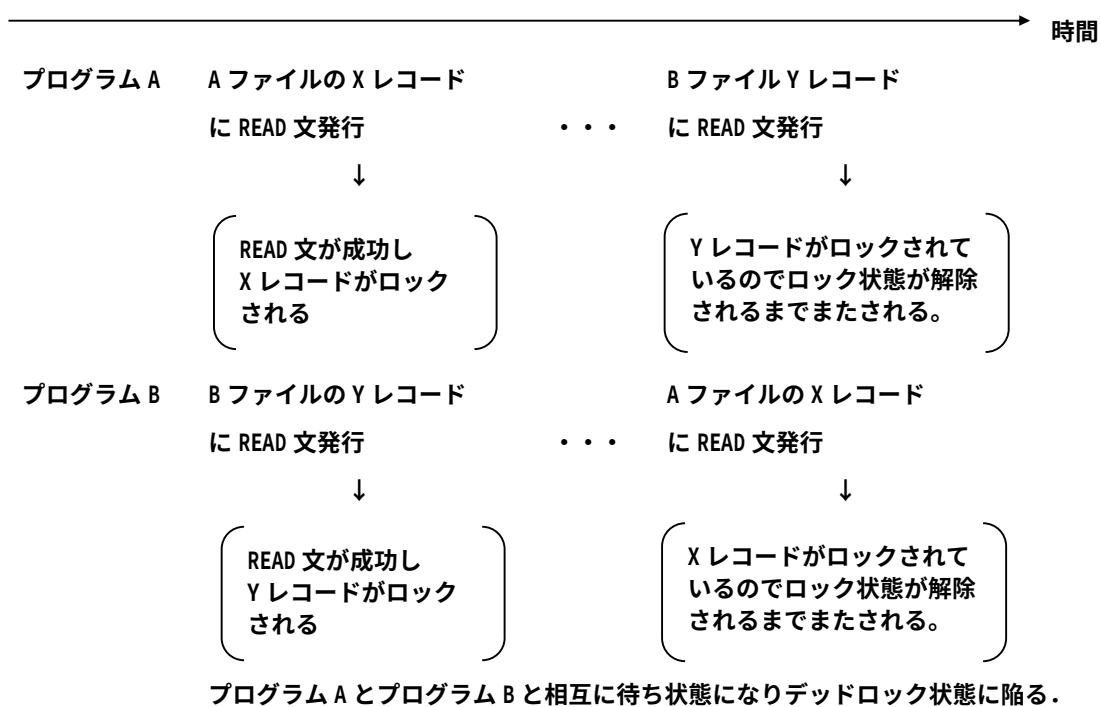
(ア) READ 文の場合にアクセスするレコードをロックし、次の入出力文が発行されるまでロック状態となる。

(イ) READ 文以外の入出力文の場合に文中でアクセスするレコードを文の実行中のみロックし文の終了後にロックを解除する。

(ウ) ロック状態のレコードを他のプログラムからアクセスした場合には、ロック状態が解除されるまで待たされる。

複数のファイルでファイル共用する場合に、デッドロックを起こすことがあるので注意が必要である。

例 デッドロック発生のケース



デッドロックを防ぐには、複数ファイルのファイル共用を止める。または、複数ファイルでファイル共用をする場合には、すでにロック状態のレコードがある時にロック状態を解除してから別の共用ファイルにアクセスを行う。

【Standard】大容量記憶ファイル共用時のレコードのロックおよびアンロックと WITH UNLOCK 句について、以下に説明する。

1. レコードのロックおよびアンロック

レコード排他制御は、レコードのロックとアンロックという 2 つの状態によって制御される。

ロック状態とは、そのレコードが他のプログラムからの呼び出しを禁止されている状態をいい、アンロック状態とは、そのレコードの呼び出しが他のプログラムに解放されている状態をいう。あるレコードがプログラムから呼び出されると、システムは自動的にそのレコードをロック状態にし、そのレコードに対する処理が終わるまで他のプログラムからの呼び出しを禁止する。実際には、ロックされたレコードを他のプログラムが呼び出しにいくと、システムはそのレコードに対する処理が終了するまで他のプログラムからの呼び出しを待たせる。

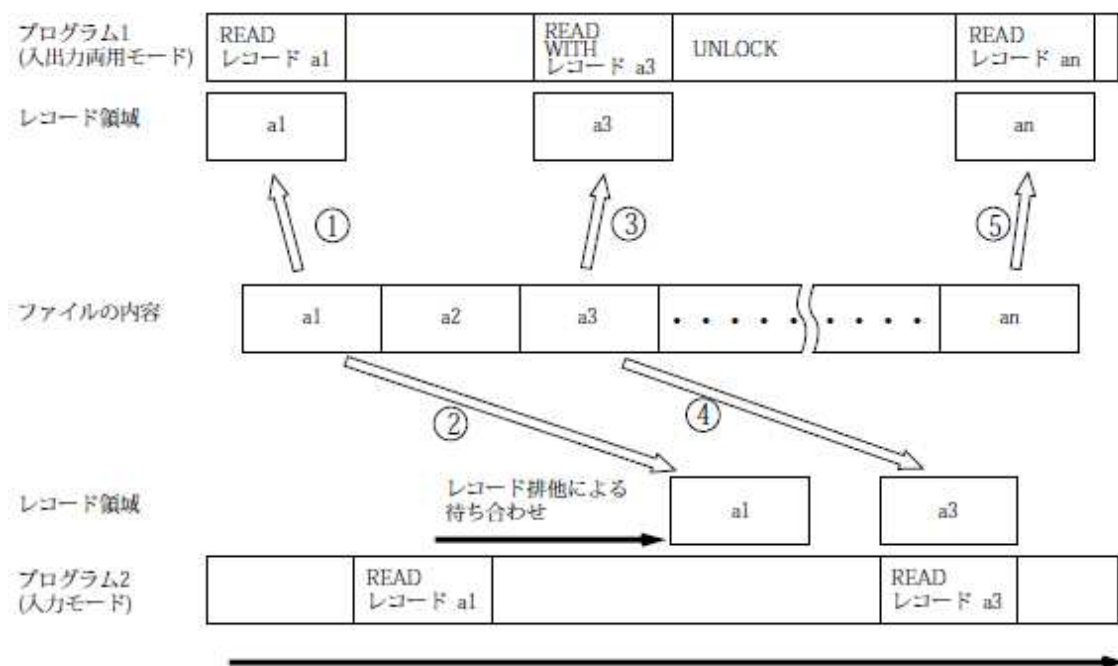
次に、レコードがロック、アンロックされる条件を次の表に示す。

表 レコードのロックとアンロック

ファイル編成	条件	
	ロックされる条件	アンロックされる条件
順ファイル	WITH UNLOCK 句のない READ 文を実行し正常に終わったとき	• 次の READ 文, REWRITE 文を実行したとき • ファイルがクローズされたとき • プログラムが正常／異常終了したとき
相対ファイル 索引ファイル	WITH UNLOCK 句のない READ 文または START 文を実行し正常に終わったとき	• 次の READ 文, REWRITE 文, DELETE 文, WRITE 文, START 文を実行したとき • ロックされたレコードに REWRITE 文, DELETE 文を実行したとき • ファイルがクローズされたとき • プログラムが正常／異常終了したとき

2. ファイルの共用時のレコード排他制御と WITH UNLOCK 句

READ 文における WITH UNLOCK 句の指定は、呼び出したレコードに対して上記のレコードのロックを行わないようにすることを意味する。



次に順を追って説明する。

- ① プログラム 1 が WITH UNLOCK 句のない READ 文によってレコード a1 を読み込む。レコード a1 はロック状態となる。
- ② プログラム 2 が同じくレコード a1 を読み込もうとするが、レコード排他制御によって読み込みは待たされる。
- ③ プログラム 1 が次のレコード a3 を読み込む。レコード a1 はアンロック状態となる。WITH UNLOCK 句の指定があるので、レコード a3 はアンロック状態のままである。読み込みを待たされていたプログラム 2 がレコード a1 を読み込む。
- ④ プログラム 2 がレコード a3 を読み込む。このときレコード a3 はロック状態ではないので、待たされることなく読み込まれる (⑤の動作が終わるまで待たされることはない)。

3.12 入出力状態

ファイル管理記述項中に **FILE STATUS**(ファイルの状態)句を書くと、**OPEN** 文、**CLOSE** 文、**READ** 文、**WRITE** 文、**REWRITE** 文、**DELETE** 文または **START** 文を実行中その入出力操作の状態を示す値を、**FILE STATUS** 句で指定した 2 桁のデータ項目中に設定する。この 2 桁のデータ項目の値のことを状態キーという。

3.12.1 状態キー1

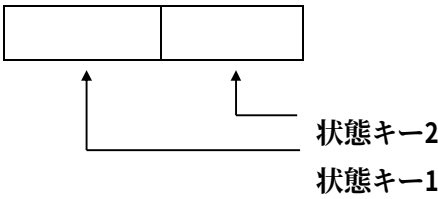
FILE STATUS データ項目の左端の 1 桁を状態キー1と呼ぶ。状態キー1は、入出力操作の結果を次の状態コードで示します。

“0”	正常終了状態	入出力文は正常に完了した。
“1”	AT END 状態	書き方 1 の READ 文が完全に実行されなかった。すなわち、ファイル中に次に読み出すべきレコードが存在しなかった。
“2”	INVALID KEY 状態	次のいずれかの理由により入出力文が完全に実行されなかった。 • 順序エラー • 二重キーエラー • レコード未発見 • 境界侵犯
“3”	永続誤り状態	ファイルのオーバフロー、物理入出力エラー、ファイルの矛盾が発生した。
“4”	論理誤り状態	文の実行順序の論理誤りが発生した。
“9”	その他の誤り状態	COBOL 特有の障害が発生した。

3.12.2 状態キー2

FILE STATUS データ項目の右端の 1 桁を状態キー2 と呼ぶ。状態キー2 は、状態キー1 で示された状態より詳しい情報を示す。

3.12.3 状態キー1 と状態キー2 の組み合わせ



状態キー1……終了状態の大分類を示す。
状態キー2……状態キー1と組み合わせて状態キー1 の小分類を示す。

表 3-12 状態キー1 と状態キー2 の組み合わせ

ファイル	状態 キー1	状態 キー2	内 容
大容量記憶ファイル プリンタファイル	0	0	正常終了，入出力動作が正常に終了したことを示す。
	3	0	物理入出力エラー発生を示す。詳細情報は B_GERRNO システムサブルーチンで参照できる。
	3	9	ファイル固有属性と、そのファイルに対してソース単位が指定した属性との間に不整合が検出されたため、OPEN 文が不成功に終わった。次のいずれかである。 ・本システムでは、ファイル編成、アクセスモード、または最大レコード長が正しくない場合に、本状態となる。 ・EXTERNAL ファイルの属性が不一致である。
	4	1	開かれている状態のファイルに OPEN 文を実行しようとした(二重オープン)。
	4	2	開かれていない状態のファイルに CLOSE 文を実行しようとした(ノットオープン)。

	4	8	正しい OPEN モードで開いていないファイル結合子に対して WRITE 文を実行しようとした。次のいずれかである。 ・順呼び出し法の場合、ファイル結合子が拡張モードや出力モードで開かれていない。 ・動的呼び出し法や乱呼び出し法の場合、ファイル結合子が入出力両用モードや出力モードで開かれていない。
	9	0	その他のエラーが発生した。 [Enterprise for Linux] 相対ファイルでは、この入出力状態は設定しない。
	9	5	順／行順／プリンタファイルのコード変換中にコード変換が不成功となった。 [Enterprise] 翻訳時のオプション指定によりプログラムの固有文字集合を Unicode とすると、コード変換が発生する入出力動作でこの入出力状態が発生する。 [Standard] [Enterprise for Windows] 翻訳時のオプション指定によりプログラムの固有文字集合を JIPS(J)または JIS2004 とすると、コード変換が発生する入出力動作でこの入出力状態が発生する。
	9	N	内部エラー。

ファイル	状態 キー1	状態 キー2	内 容
大容量記憶 ファイル	0	2	次のいずれかである。 ・READ 文で読み込んだ副レコードキーと等しい副レコードキーをもつレコードが続いている(索引ファイルのみ)。 ・WRITE 文または REWRITE 文によって書かれたレコードの副レコードの値と等しいレコードがすでにファイルに存在している(索引ファイルのみ)。
	0	4	次のいずれかである。 ・読み込んだレコードの文字位置の個数がレコード記述項で指定した最大の大きさより大きい、または指定された最小の大きさより小さい(行順ファイル)。 ・1つのセルの文字列長が対応する基本項目の項目長よりも長い、または数値データの桁あふれが発生する、または1レコード中のセルの個数と基本項目の個数が等しくない(行順ファイル CSV 形式)。
	0	5	OPEN 文は正常終了した。ただし、参照した不定ファイルは存在しなかった。
	1	0	AT END 条件、READ 文において AT END 条件の発生を示す。

	1	4	RELATIVE KEY 句の指定があれば、読み込まれたレコードの相対キー番号が RELATIVE KEY 句で指定されたデータ項目に格納される。 相対キー番号がデータ項目に納まらないとき、状態キーには '14' が格納される(相対ファイル)。
	2	1	順呼び出しにおいて、順序誤りがあった。すなわち、成功した READ 文と次の REWRITE 文の間で主レコードキーの値が変更されたか、WRITE 文の実行時に主レコードキーの値が昇順になっていない。
	2	2	INVALID KEY 条件(二重キー登録) 相対ファイルにおいて、すでにファイル内に存在するキーと同じキーのレコードを出力しようとした場合。
	2	3	INVALID KEY 条件(指定レコード未発見) 指定された参照キーをもつレコードがファイル内に見つからなかった場合(相対ファイル)。
	2	4	INVALID KEY 条件(境界侵犯) 確保したファイルの領域を超えてレコードを入出力しようとした場合(相対ファイル)。
	3	4	ファイルオーバーフロー ファイル領域がデータで満たされている状態にもかかわらず WRITE 文によってデータを追加しようとした場合(順呼び出しのファイルのみ)。
	3	5	不定ファイルでないファイルが存在しないとき。OPEN 文は不成功。 [Enterprise for Windows] [Standard] OS が許していない文字を含むファイル名を指定した場合またはファイル名が長すぎる場合、OPEN 文の入出力状態が 35 または 99 のいずれかになる。いずれの値となるかは文字や文字の出現位置、ファイルエンジンなどによって異なる場合がある。 OS が許していない文字 : ¥ / : * ? " < >
	3	7	[Enterprise for Windows] [Standard] あるファイルに OPEN 文を実行しようとしたが、OPEN 文に指定した OPEN モードはそのファイルには使えない。次のいずれかである。 ・EXTEND 指定や OUTPUT 指定を書いたが、そのファイルでは書き込み操作ができない。 ・I-O 指定を書いたが、入出力両用モードで開いたときにそのファイル編成で行ってもよい入出力操作が、そのファイルでは使えない。 ・INPUT 指定を書いたが、そのファイルでは読み取り操作ができない。
	3	8	ロックして閉じたファイルを OPEN しようとした。

	4	3	順呼び出し法の大容量記憶ファイルに対して、あるファイル結合子を介して REWRITE 文や DELETE 文を実行する前に、同じファイル結合子を介して関連ファイルに対して実行された最後の入出力文が、成功完了した READ 文ではない。
	4	4	次のいずれかである。 ・関連するファイル名の RECORD IS VARYING 句によって許される最大レコードより大きいか、最小レコードより小さいレコードを、書き出しまたは書き換えしようとした。 ・順ファイルに対してレコードを書き換えようとしたが、そのレコードが書き換えられるレコードと同じ大きさでない。 ・書き方 3(固定長または可変長)の RECORD 句に対して本システムの規定によりレコード形式が可変長である場合に、固定長または可変長形式の RECORD 句で指定された最大レコードより大きいか、最小レコードより小さいレコードを、書き出そうまたは書き換えようとした。
	4	6	入力モードや入出力両用モードで開かれたファイル結合子に順呼び出しの READ 文を実行しようとしたが、有効な次のレコードが確定しない。次のいずれかである。 ・そのファイル結合子を参照する先行する START 文が不成功だった。 ・そのファイル結合子を参照する先行する READ 文が不成功だった。
	4	7	入力モードまたは入出力両用モード以外で開かれたファイルに READ 文または START 文を実行しようとした。
	4	9	入出力両用モードで開かれていないファイルに DELETE 文または REWRITE 文を実行しようとした。
	9	1	ファイルが他から使用されている。
	9	2	[Enterprise for Linux] 他の実行単位がレコードをロックしている。または、その他の理由でタイムアウトが発生したため、 READ 文が実行できなかった。または、レコードがロックされていたため、 REWRITE 文または DELETE 文が実行できなかった。
	9	3	[Enterprise for Linux] 開かれていないファイルに UNLOCK 文を実行しようとした。
	9	9	[Enterprise for Windows] [Standard] OS が許していない文字を含むファイル名を指定した場合またはファイル名が長すぎる場合、 OPEN 文の入出力状態が 35 または 99 のいずれかになる。いずれの値となるかは文字や文字の出現位置、ファイルエンジンなどによって異なる場合がある。 OS が許していない文字 : ¥ / : * ? " < >

プリンタファイル	9	1	[Standard] ファイルビジー(アプリケーション実行環境(プリンタ機能)使用時)。
	9	7	[Standard] 紙切れ(アプリケーション実行環境(プリンタ機能)使用時)。
	9	9	[Standard] プリンタファイルに対するシステムエラー。ファイル名が不正である。

備考 表 3-12 に記述されていない状態キーについては、次の説明書を参照。

『ISAM 利用の手引』(索引ファイルのとき)

[Standard] 『IFAS FILE, IFASPRO RDB 利用の手引』

(相対ファイルまたは索引ファイルのとき),

『COBOL DB アクセス for Oracle 説明書』(索引ファイルのとき)

3.12.4 入出力状態の検査

ファイル入出力を行っているプログラムでは、利用者はその入出力結果を必ず検査しなければならない。特定の入出力文によって起こりうる状態は、ファイル編成、呼び出し法および OPEN 文のモードによって一意に定まるので、利用者はその起こりうる状態を調べればよい。また、ある事象が発生した時点での COBOL プログラムの制御の動きは、そのファイルに対する USE 文の指定の有無や入出力文の書き方によって左右されるので、利用者は適切な場所に入出力状態を検査する手続きを挿入しなければならない。プログラムの書き方によっては、入出力状態を二重に検査したり、どこでも検査されないことになったりするかもしれないので十分注意しなければならない。プログラムの制御の動きは次の条件により決まる。

- USE 手続き の有無
- 入出力文の AT END 句または INVALID KEY 句の有無

(1) USE 手続きと AT END 句または INVALID KEY 句の指定されている場合

実行されるもの	該当する状態キー
USE 手続き	30, 34, 35, 37, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 91, 9N [Enterprise] 90, 95 [Enterprise for Linux] 92, 93

AT END 句	10
INVALID KEY 句	21, 22, 23, 24
いずれも実行されず次の文へと進む	00, 02, 04, 05

- (2) USE 手続きだけが指定されている場合 (入出力文に AT END 句または, INVALID KEY 句が指定されていない場合)

実行されるもの	該当する状態キー
USE 手続き	10, 14, 21, 22, 23, 24, 30, 34, 35, 37, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 91, 97, 99, 9N [Enterprise] 90, 95 [Enterprise for Linux] 92, 93
実行されず次の文へと進む	00, 02, 04, 05

- (3) AT END 句または INVALID KEY 句だけが指定されている場合 (USE 手続き指定なし)

実行されるもの	該当する状態キー
AT END 句	10
INVALID KEY 句	21, 22, 23, 24
メッセージ表示後プログラム異常終了	30, 34, 35, 37, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 91, 9N 注(1) [Enterprise] 90, 95 [Enterprise for Linux] 92, 93
上記いずれも実行されず次の文へと進む	00, 02, 04, 05 30, 34, 35, 37, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 91, 9N 注(2) [Enterprise] 90, 95 [Enterprise for Linux] 92, 93

注意 (注 1) FILE STATUS 句が指定されていない場合のファイルオーバーフロー, 永続誤り, 論理誤り, その他誤り。

(注 2) FILE STATUS 句が指定されている場合のファイルオーバーフロー、永続誤り、論理誤り、その他誤り。

3.13 コード変換機能

[Standard] [Enterprise for Windows] 本コンパイラでは、-CF オプションを指定すると、CODE-SET 句の符号名で参照する文字集合が、外部記憶媒体上のデータ表現に使用される。

ファイル入出力文実行時に、ファイル記述項に従属するレコード記述中の基本項目の属性に合わせて、外部記憶媒体上の文字集合とプログラムの固有文字集合との間でデータ表現の変換が行われる。

データ表現の変換は次のように行われる。

CODE-SET 句で参照 される文字集合	レコード記述中の基本項目の属性	
	文字列	日本語文字列
UTF-8	JIS $\longleftrightarrow$ UTF-8 変換	JIPS(J) $\longleftrightarrow$ UTF-8 変換
UTF-16LE	JIS $\leftarrow \rightarrow$ UTF-16 Little Endian 変換	JIPS (J) $\leftarrow \rightarrow$ UTF-16 Little Endian 変換
上記以外	無変換	無変換

-CF オプションが指定されているとき、CODE-SET UTF-8 または UTF-16LE 指定のない行順ファイル(CSV 形式)は使用できない。

-CF オプションが指定されているとき、行順ファイルまたは行順ファイル(CSV 形式)にのみ CODE-SET UTF-8 または CODE-SET UTF-16LE を指定できる。

-CF オプションが指定されていないとき、CODE-SET 句はメモである。

[Enterprise] 本コンパイラでは、-CU オプションを指定すると、CODE-SET 句の符号系名で参照する文字集合が、外部記憶媒体上のデータ表現に使用される。

ファイル入出力文実行時に、ファイル記述項に従属するレコード記述中の基本項目の属性に合わせて、外部記憶媒体上の文字集合とプログラムの固有文字集合との間でデータ表現の変換が行われる。

データ表現の変換は次のように行われる。

CODE-SET 句で参照 される文字集合	レコード記述中の基本項目の属性	
	文字列	日本語文字列
STANDARD-2	無変換	シフト JIS $\longleftrightarrow$ UTF-16Little Endian 変換
UTF-8	無変換	UTF-8

		←→UTF-16Little Endian 変換
UCS-2	UTF-16Little Endian ←→UTF-8 変換	無変換
上記以外	無変換	無変換

-CU オプションが指定されていると、行順ファイル(CSV 形式)は使用できない。

-CU オプションが指定されていないとき、CODE-SET 句はメモである。

3.13.1 CODE-SET 句

ファイル記述項で CODE-SET 句を指定することで、外部記憶媒体上の文字集合を指定することができる。

記述形式は、次のとおりである。

CODE-SET IS 符号系名 1

[Enterprise] -CU オプションが指定されているとき、以下の規則を適用する。

符号系名 1 が特殊名段落において STANDARD-2 に対応づけられている場合、入出力文の実行時に、外部記憶媒体上の文字集合をシフト JIS として、レコード記述中の基本項目の属性に合わせてコード変換が行われる。

符号系名 1 が特殊名段落において UTF-8 に対応づけられている場合、入出力文の実行時に、外部記憶媒体上の文字集合を UTF-8 として、レコード記述中の基本項目の属性に合わせてコード変換が行われる。このとき、外部記憶媒体上のファイルの先頭にはバイトオーダーマーク (BOM) が指定されてはならない。

符号系名 1 が特殊名段落において UCS-2 に対応づけられている場合、入出力文の実行時に、外部記憶媒体上の文字集合を UTF-16LE として、レコード記述中の基本項目の属性に合わせてコード変換が行われる。このとき、外部記憶媒体上のファイルの先頭にはバイトオーダーマーク (BOM) が指定されていなければならない。

符号系名 1 が特殊名段落において、STANDARD-2、UTF-8、UCS-2 以外に対応づけられている場合、符号系名 1 は NATIVE に対応づけられているものとみなす。このとき、外部記憶媒体上の文字集合は、プログラムの固有文字集合と同じであるとみなされ、コード変換は行われない。

CODE-SET 句は、既存のシフト JIS 形式ファイルから Unicode 形式ファイルへ変換したい場合や COBOL プログラム以外からも使用するファイルを扱う場合に指定するとよい。COBOL プログラム内だけで利用する場合や、コード変換の必要がない場合は、CODE-SET 句を指定しないほうが実行効率がよくなる。

同時に `-CC $m$` オプションが指定されていると、`CODE-SET` 句指定のない順／行順／プリンタ／`SYSIN`／`SYSOUT` ファイルに、暗に `CODE-SET` 句が指定されているものとして処理を行う。

m には `{0 | 12}` のうち 1 つが指定できる。`-CC $m$` オプションを指定すると、`CODE-SET` 句の符号系名が、特殊名段落において以降に対応づけられているものとして扱う。

-CC m 指定	特殊名段落の指定
-CC0	UTF-8
-CC1	UCS-2
-CC2	STANDARD-2

`CODE-SET` 句が明に指定されているファイルには、本オプションは影響しない。

[Standard] [Enterprise for Windows] `-CF` オプションが指定されているとき、以下の記述形式が指定できる。

<code>CODE-SET IS</code>	符号系名1 UTF-8 UTF-16LE

UTF-8 が指定されている場合、入出力文の実行時に、外部記憶媒体上の文字集合を UTF-8 として、レコード記述中の基本項目の属性に合わせてコード変換が行われる。このとき、外部記憶媒体上のファイルの先頭にはバイトオーダーマーク (BOM) が指定されていなければならない。

UTF-16LE が指定されている場合、入出力文の実行時に、外部記憶媒体上の文字集合を UTF-16LE として、レコード記述中の基本項目の属性に合わせてコード変換が行われる。このとき、外部記憶媒体上のファイルの先頭にはバイトオーダーマーク (BOM) が指定されていなければならない。

符号系名 1 が指定されている場合、メモとして扱い、コード変換は行われない。

`CODE-SET` 句を指定できるのは、行順ファイルおよび行順ファイル (CSV 形式) のみである。

また、行順ファイル (CSV 形式) の場合は、必ず `CODE-SET` 句を指定しなければならない。

3.13.2 READ 文

READ文の記述形式は、次のとおりである。

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**]

CODE-SET句が指定されたファイルのREAD文実行時にコード変換が行われる場合、次のとおりである。

- (a) INTO句が指定されていないとき、ファイル名に関連づけられたレコード記述中の基本項目の属性に従ってコード変換が行われる。レコード記述中にREDEFINES句が記述されている場合、REDEFINES句を含む記述項およびREDEFINES句に従属する項目については、変換のための属性として扱われない。
ファイル名に関連づけられたレコード記述が複数ある場合、先頭のレコード記述の基本項目の属性に従ってコード変換が行われる。
- (b) INTO句が指定されているとき、INTO句の一意名中の基本項目の属性に従ってコード変換が行われる。一意名中にREDEFINES句が記述されている場合、REDEFINES句を含む記述項およびREDEFINES句に従属する項目については、変換のための属性として扱われない。
- (c) コード変換後のレコードの大きさが、レコード記述とあっていないとき、レコード記述の項目の大きさに合わせて、データの切捨て、空白づめが行われる。

[Enterprise] 例

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SPECIAL-NAMES.

ALPHABET 符号系名 IS ～.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT FILE1 ASSIGN TO F1-MSD

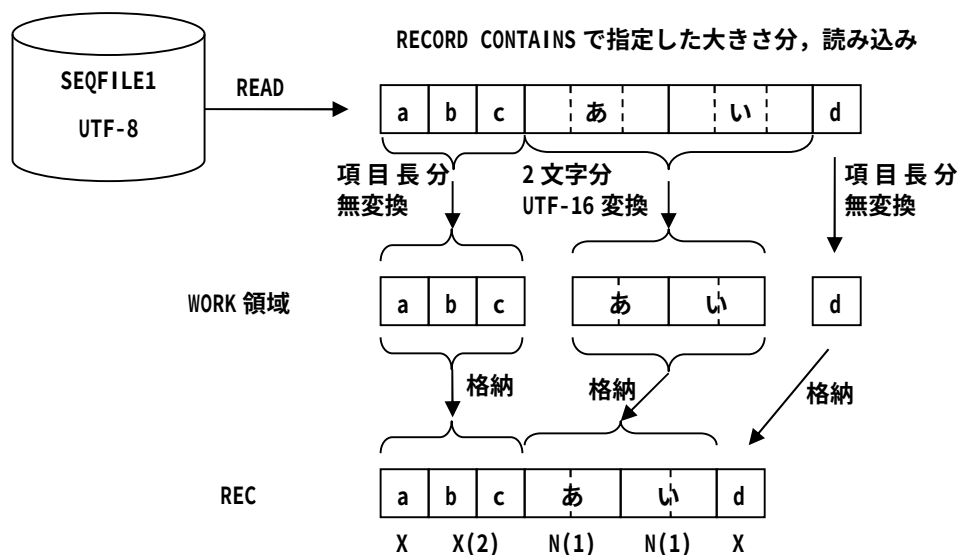
ORGANIZATION SEQUENTIAL

FILE STATUS IS FSTS.

```
DATA DIVISION.
FILE SECTION.
FD  FILE1
    LABEL RECORD STANDARD
    VALUE OF IDENTIFICATION "SEQFILE1"
    CODE-SET IS 符号系名.
01  REC.
    02 RECX1  PIC X(1).
    02 RECX2  PIC X(2).
    02 RECN1  PIC N(1).
    02 RECN2  PIC N(1).
    02 RECX3  PIC X(1).
PROCEDURE DIVISION.
PROG.
    OPEN INPUT FILE1.
    READ FILE1 AT END DISPLAY "AT END".
```

■正常なとき

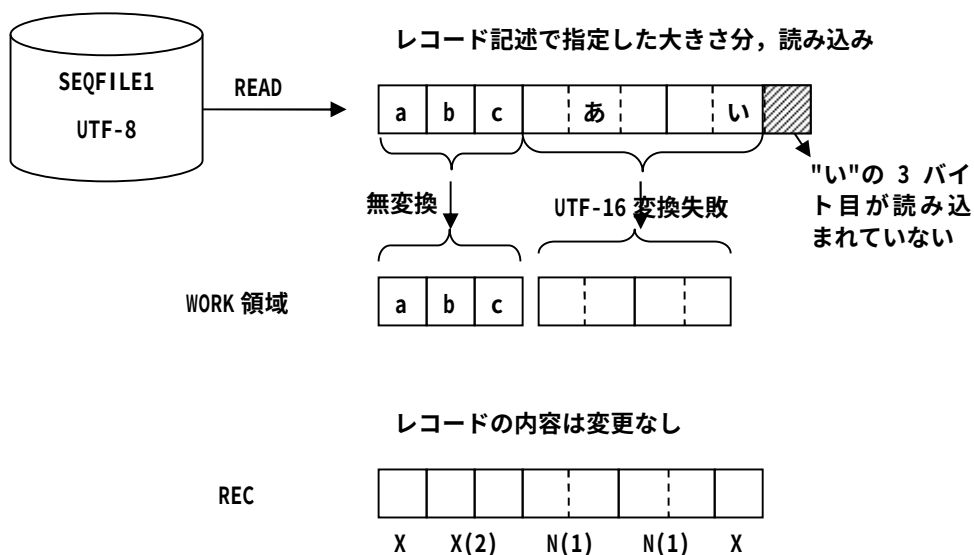
ALPHABET 句で符号系名が UTF-8 に関連づけられており,
RECORD CONTAINS 10 CHARACTERS が指定されている場合



FSTS = "00"が設定される。

■データ変換が不正な場合

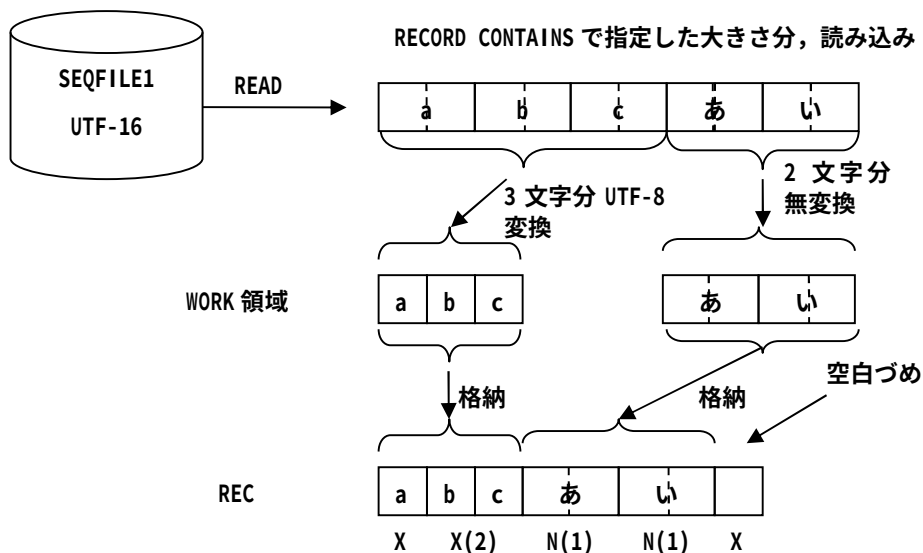
ALPHABET 句で符号系名が UTF-8 に関連づけられており,
RECORD CONTAINS 指定なしの場合



FSTS = "95"が設定される。

■データの空白詰めが行われる場合

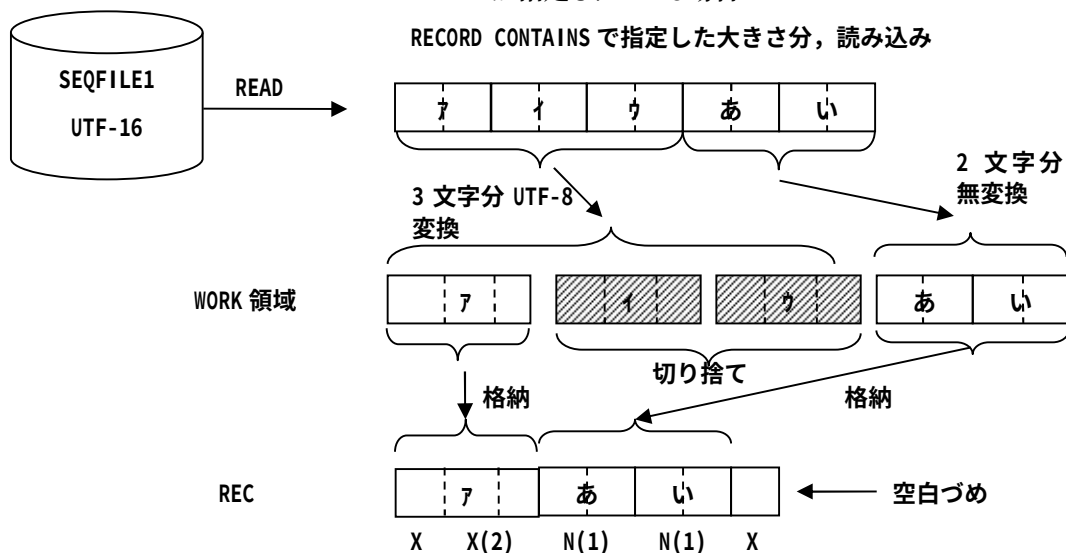
ALPHABET 句で符号系名が UTF-16 に関連づけられており、
RECORD CONTAINS 10 CHARACTERS が指定されている場合



FSTS = "04"が設定される。

■データの切り捨てが行われる場合

ALPHABET 句で符号系名が UTF-16 に関連づけられており、
RECORD CONTAINS 10 CHARACTERS が指定されている場合



FSTS = "04"が設定される。

[Standard] [Enterprise for Windows] -CF オプションありのとき, 次のとおり READ 文実行時にコード変換が行われる。

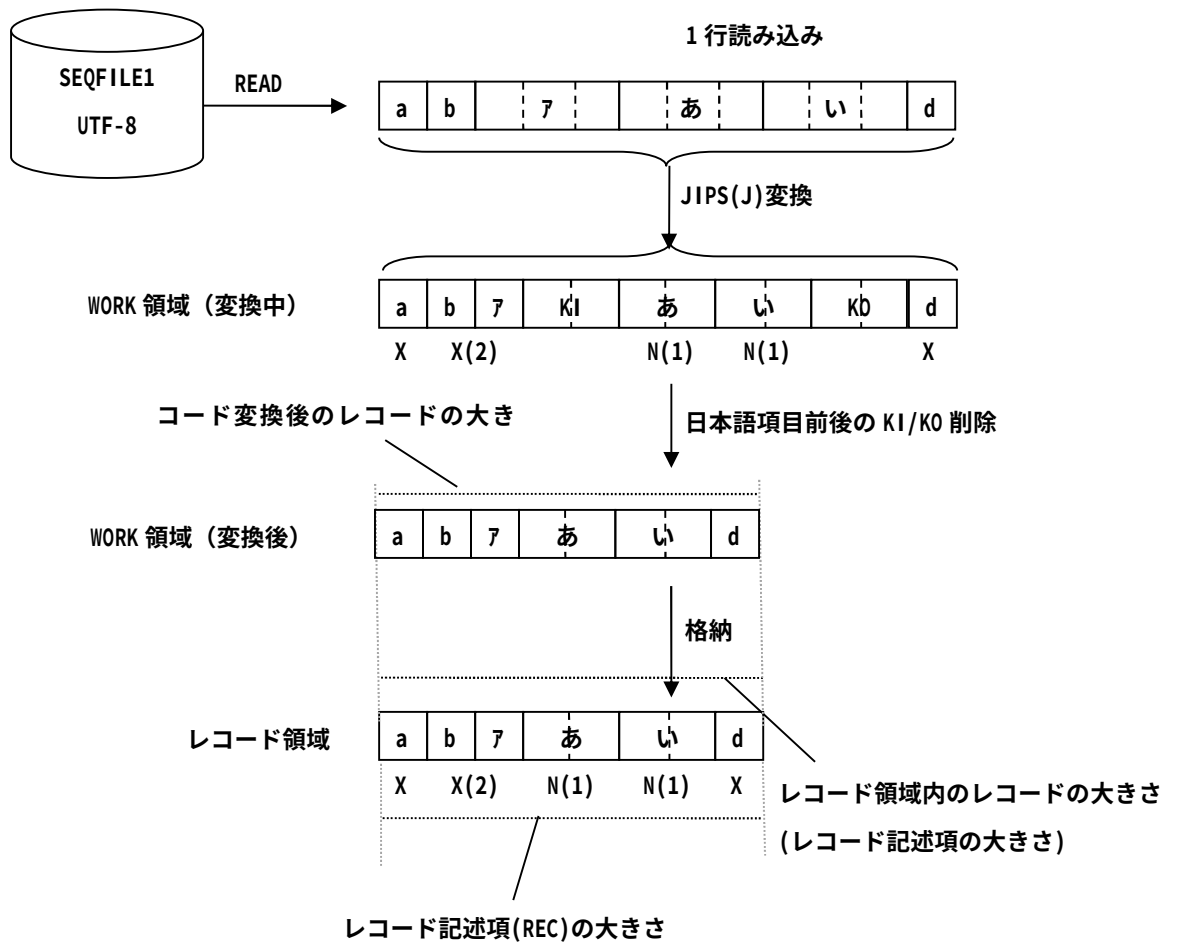
例

```

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT FILE1 ASSIGN TO F1-MSD
        ORGANIZATION LINE SEQUENTIAL
        FILE STATUS IS FSTS.
DATA DIVISION.
FILE SECTION.
FD FILE1
    RECORD句
    LABEL RECORD STANDARD
    VALUE OF IDENTIFICATION "SEQFILE1"
    CODE-SET句 .
01 REC.
    02 RECX1 PIC X(1).
    02 RECX2 PIC X(2).
    02 RECN1 PIC N(1).
    02 RECN2 PIC N(1).
    02 RECX3 PIC X(1).
PROCEDURE DIVISION.
PROG.
    OPEN INPUT FILE1.
    READ FILE1 AT END DISPLAY "AT END".
    
```

■ 正常なとき

-CF オプション指定時, 行順ファイルに CODE-SET 句で UTF-8 が指定されており,
RECORD CONTAINS 句が指定されていない場合

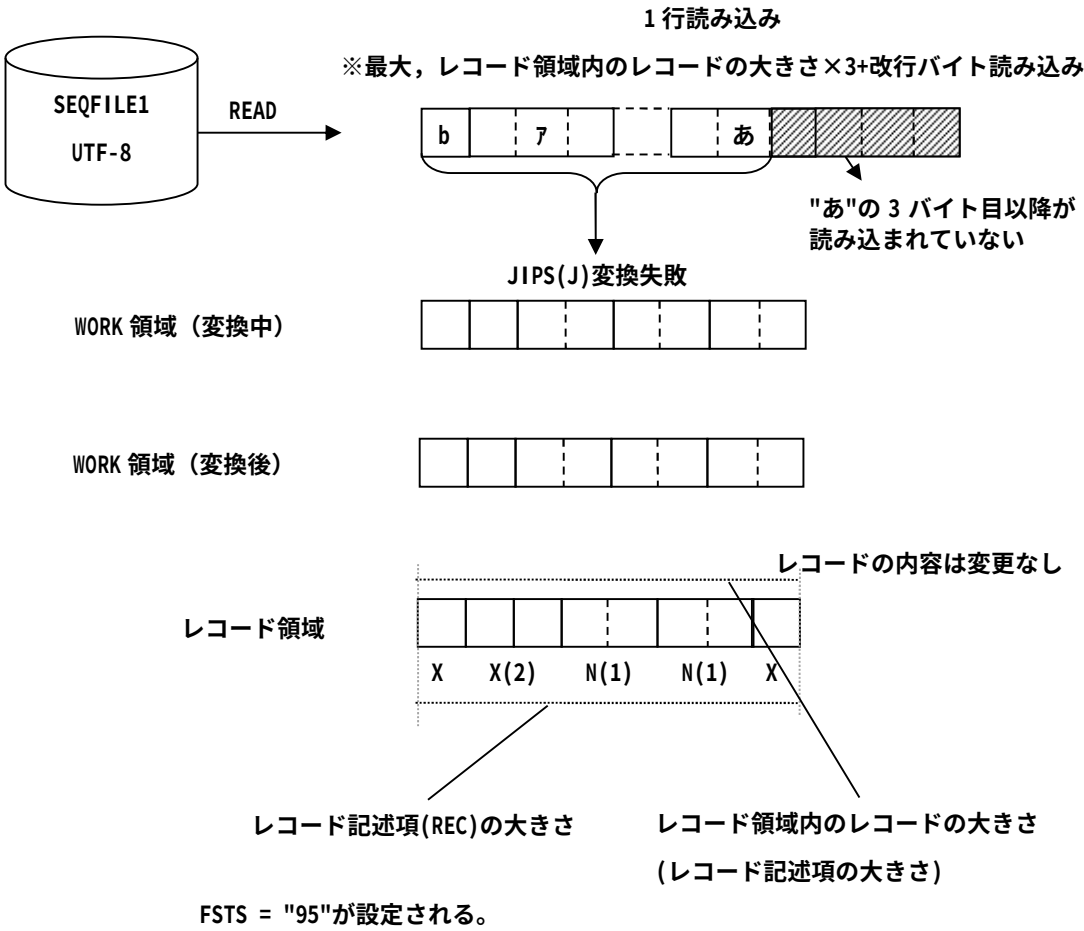


FSTS = "00"が設定される。

日本語項目にはシフトコード (KI/KO) はつかない

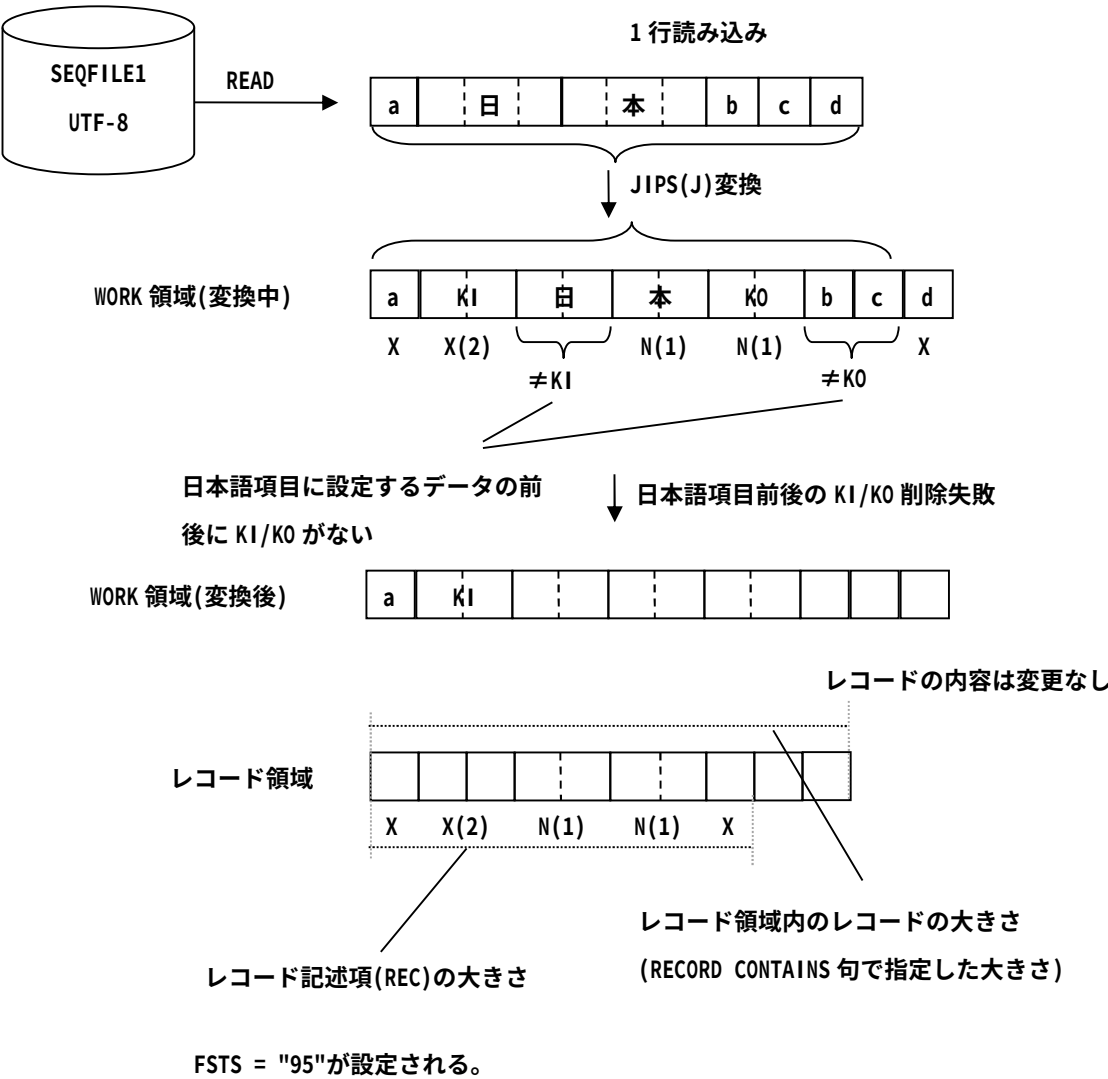
■データ変換が不正な場合

-CF オプション指定時, 行順ファイルに CODE-SET 句で UTF-8 が指定されており,
RECORD CONTAINS 句が指定されていない場合



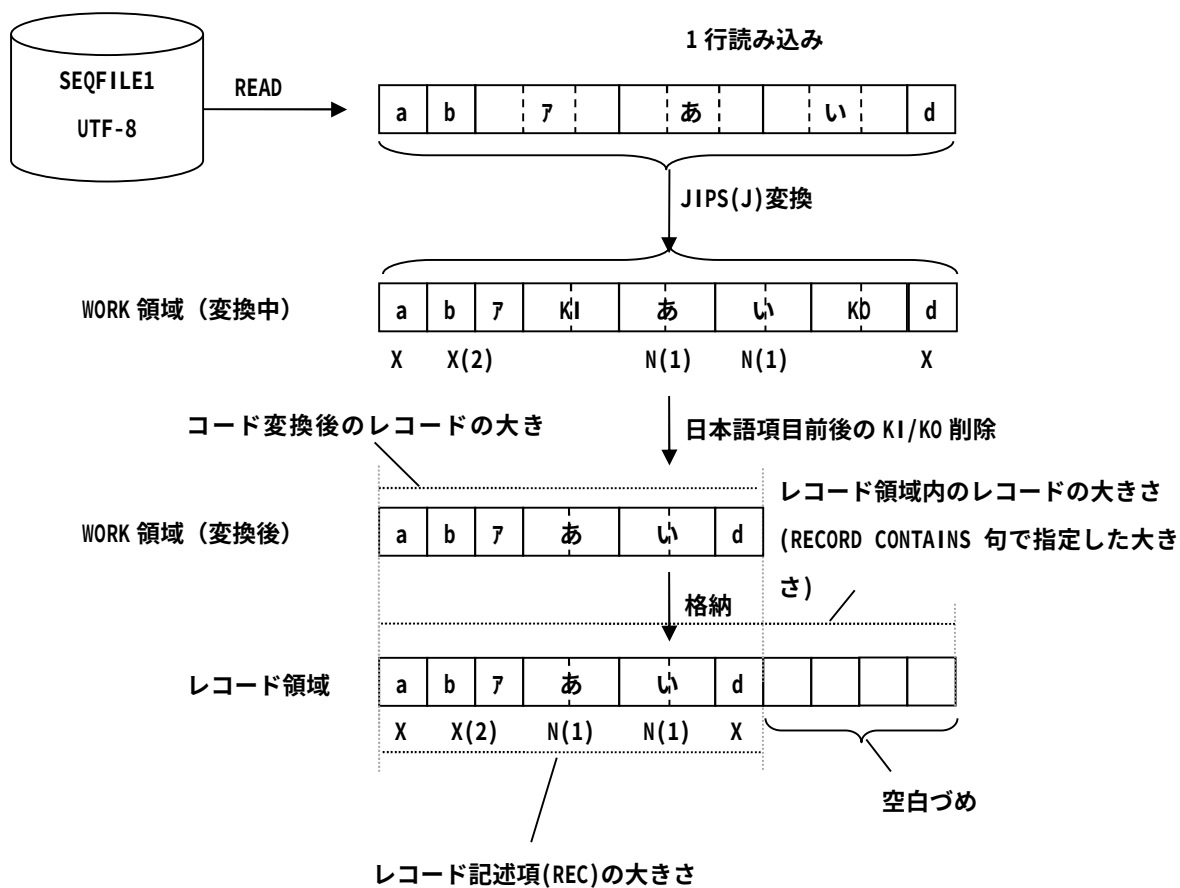
■レコードの基本項目の属性とデータが不一致の場合

-CF オプション指定時, 行順ファイルの CODE-SET 句で UTF-8 が指定されており,
RECORD CONTAINS 10 CHARACTERS が指定されている場合



■データの空白づめが行われる場合

-CF オプション指定時, 行順ファイルの CODE-SET 句で UTF-8 が指定されており,
RECORD CONTAINS 12 CHARACTERS が指定されている場合

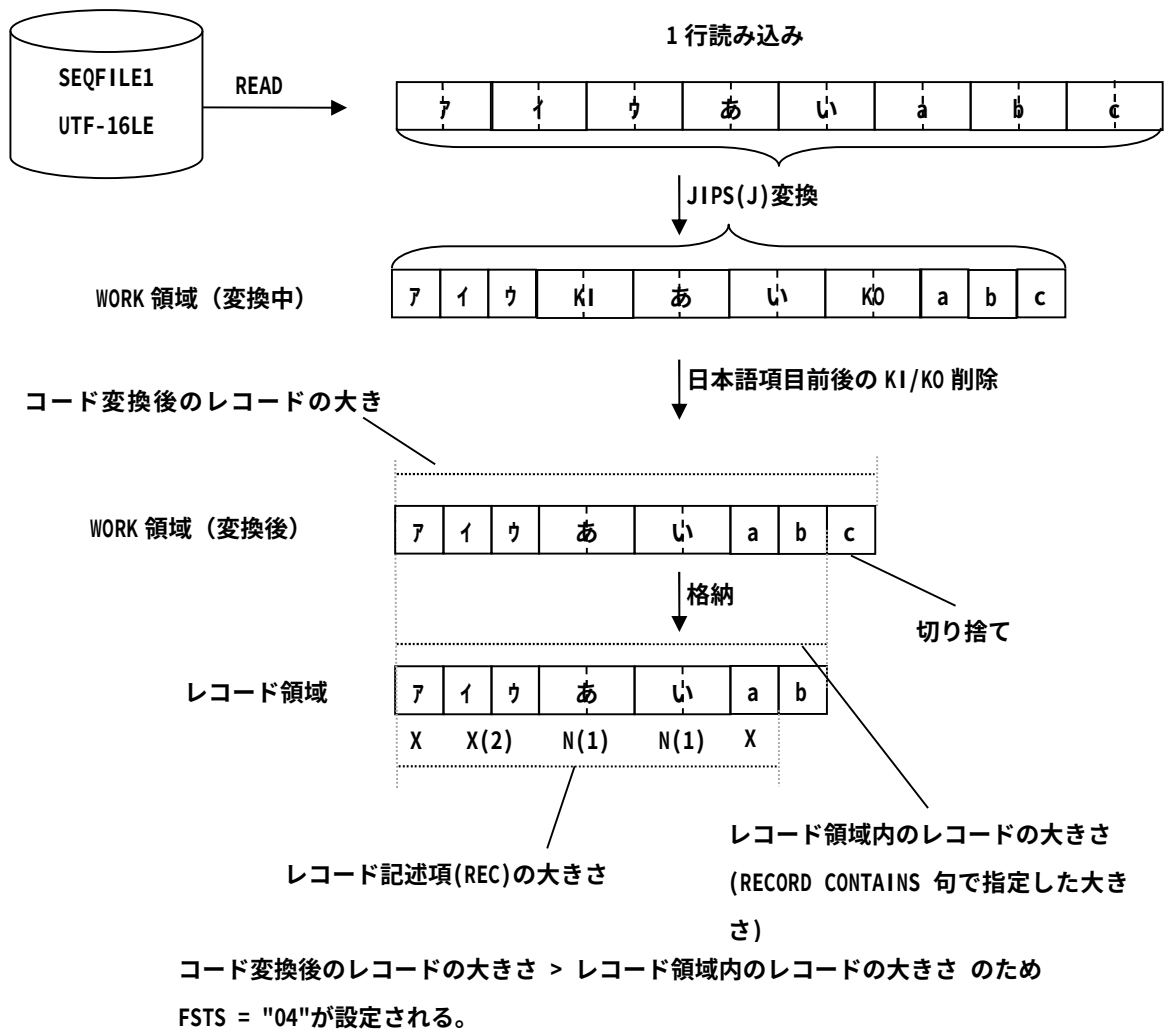


コード変換後のレコードの大きさ < レコード領域内のレコードの大きさ のため
FSTS = "04"が設定される。

日本語項目にはシフトコード (KI/KO) はつかない

■データの切り捨てが行われる場合

-CF オプション指定時, 行順ファイルの CODE-SET 句で UTF-16LE が指定されており,
RECORD CONTAINS 9 CHARACTERS が指定されている場合



3.13.3 REWRITE 文

REWRITE 文の記述形式は、次のとおりである。

REWRITE レコード名1 [**FROM** 一意名1]
[**END-REWRITE**]

CODE-SET句が指定されたファイルのREWRITE文実行時にコード変換が行われる場合、次のとおりである。

- (a) FROM句が指定されていないとき、レコード名で指定されたレコード記述中の基本項目の属性に従ってコード変換が行われる。レコード記述中にREDEFINES句が記述されている場合、REDEFINES句を含む記述項およびREDEFINES句に従属する項目については、変換のための属性として扱われない。
- (b) FROM句が指定されているとき、FROM句の一意名中の基本項目の属性に従ってコード変換が行われる。一意名中にREDEFINES句が記述されている場合、REDEFINES句を含む記述項およびREDEFINES句に従属する項目については、変換のための属性として扱われない。
- (c) コード変換後のレコードの大きさが、直前のREAD文で読み込まれたコード変換前のレコードと同じ大きさでないとき、状態キーに"44"が設定される。

3.13.4 WRITE 文

WRITE 文の記述形式は、次のとおりである。

WRITE レコード名1 [**FROM** 一意名1]
[**END-WRITE**]

CODE-SET句が指定されたファイルのWRITE文実行時にコード変換が行われる場合、次のとおりである。

- (a) FROM句が指定されていないとき、レコード名で指定されたレコード記述中の基本項目の属性に従ってコード変換が行われる。レコード記述中にREDEFINES句が記述されている場合、REDEFINES句を含む記述項およびREDEFINES句に従属する項目については、変換のための属性として扱われない。
- (b) FROM句が指定されているとき、FROM句の一意名中の基本項目の属性に従ってコード変換が行われる。一意名中にREDEFINES句が記述されている場合、REDEFINES句を含む

記述項および REDEFINES 句に従属する項目については、変換のための属性として扱われない。

3.14 GLOBAL 句付きのファイル

ファイル記述項に GLOBAL 句を書く場合、ファイル管理記述項およびファイル記述項のデータ名指定で指定可能なデータ名が定義された節に注意が必要である。

ファイル管理記述項およびファイル記述項のデータ名を指定する句に対して、データ名が指定可能な場合 (○) と指定不可な場合 (×) を次に示す。

データ名を指定する句			データ名が定義された節				
			ファイル節	作業場所節	局所記憶節	連絡節	ベースデータ節
ファイル管理記述項	SELECT ～ ASSIGN	データ名 1	×	○	×	×	×
	SELECT BASED ～ ASSIGN	データ名 1	×	○	×	○ (*)	○ (*)
	FILE STATUS	データ名 1	×	○	○ (*)	○ (*)	○ (*)
		データ名 2	×	○	×	○ (*)	○ (*)
	RECORD KEY	データ名 1	○	×	×	×	×
	ALTERNATE RECORD KEY	データ名 1	○	×	×	×	×
	RELATIVE KEY	データ名 1	○	○	○ (*)	○ (*)	×
ファイル記述項	VALUE OF	データ名 1	×	○	×	×	×
	LINAGE	データ名 1	○	○	○ (*)	○ (*)	×
		データ名 2	○	○	○ (*)	○ (*)	×
		データ名 3	○	○	○ (*)	○ (*)	×
		データ名 4	○	○	○ (*)	○ (*)	×
	RECORD VARYING	データ名 1	○	○	○	○	○

	DEPENDING ON				(*)	(*)	(*)

(*) ファイル記述項に GLOBAL 句を指定した場合は、局所記憶節/連絡節/ベースデータ節のデータ名は指定できない。ただし、翻訳時に COBOL85 互換モードオプション(-Qa/-Qb オプション)を指定することで連絡節/ベースデータ節のデータ名を指定できる。

3.15 内部ファイル ID 置換機能

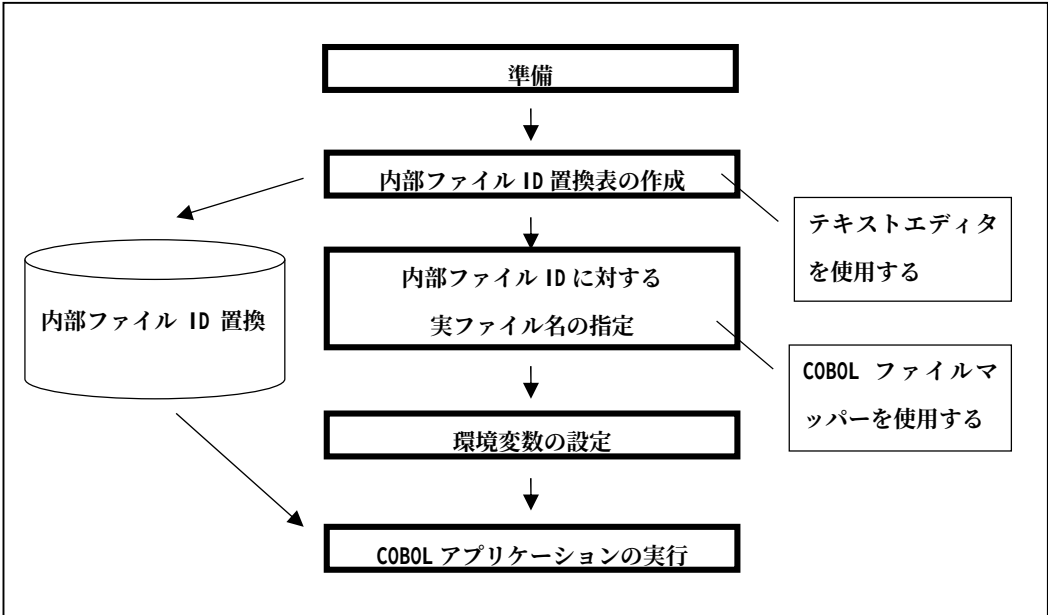
【Standard】本機能は Standard Edition 版で利用できる。SE 動作モードでは使用できない。本機能を使用するためには、COBOL A-VX 互換機能／カンパニ ID オプションのインストールが必要である。

内部ファイル ID 置換機能とは、FD 句の VALUE OF IDENTIFICATION で指定された「内部ファイル ID」を、実行時に置換することができる機能である。本機能を利用して、A-VX で提供しているカンパニ ID 毎のローカルファイル機能を実現できる。A-VX OCF のカンパニ ID を利用している環境では、ファイルアクセスは、グローバル、ローカルの概念の理解が必要である。

用語	説明
ローカルファイル	カンパニ ID 毎に完全に独立したファイルとして利用可能 (同名ファイルであっても、カンパニ ID が異なれば別ファイルとなる)。あるカンパニ ID のオペレータが実行するアプリケーションやユーティリティからは、別のカンパニ ID のローカルファイルへのアクセスはできず、ファイルの存在自体も知りえない。
グローバルファイル	どのカンパニ ID からでもアクセス可能なファイルとして利用可能。アプリケーションやユーティリティからのファイルオープン時、ローカルファイルが見つからなければ、同名のグローバルファイルを検索し、ファイルが存在すればそのグローバルファイルへのアクセスに切り替える。

3.15.1 使用方法

カンパニ ID によるファイルアクセスを移行するために必要な作業を示す。



1. 準備

(a) ファイルや表はローカルファイル単位で分割する。

A-VX にてカンパニ ID を利用しているファイルやデータベースは、カンパニ ID 毎にファイルや表を作成する。

(b) ローカルファイル単位で分割するファイルや表の名前を決める。

たとえば、カンパニ ID と内部ファイル ID を連結する。A-VX の構成が以下の場合、

カンパニ ID 内部ファイル ID	XX	YY	ZZ
FILE-A	ローカル	ローカル	ローカル
FILE-B	ローカル	ローカル	ローカル
FILE-C	グローバル		

FILE-A は、XX-FILE-A、YY-FILE-A、ZZ-FILE-A の 3 つに分割する。

FILE-B は、XX-FILE-B、YY-FILE-B、ZZ-FILE-B の 3 つに分割する。

FILE-C は、グローバルファイルなので分割しない。

内部ファイル ID 置換機能で使用する内部ファイル ID にはコンマ、および、前方と後方の半角空白は指定できない。

2. 内部ファイル ID 置換表の作成

カンパニ ID 毎にローカルファイル用の内部ファイル ID 置換表を用意する。

テキストエディタで作成する。ファイル名は、動作 OS で使用可能なファイル名であればよい。

記述方法は「3.15.2 内部ファイル ID 置換表」を参照。

例では、3つの内部ファイル ID 置換表を作成する。

XX.ini

```
# XX 用
FILE-A,XX-FILE-A
FILE-R,XX-FILE-R
```

YY.ini

```
# YY 用
FILE-A,YY-FILE-A
FILE-R,YY-FILE-R
```

ZZ.ini

```
# ZZ 用
FILE-A,ZZ-FILE-A
FILE-R,ZZ-FILE-R
```

FILE-C は、グローバルファイルなので内部ファイル ID 置換表には定義していないことに注意。

3. 内部ファイル ID に対する実ファイル名の指定

一般ファイルの場合、IFAS FILE として COBOL ファイルマッパーで指定する。マシン単位の指定でよい。

例の FILE-A が一般ファイルであったとき次のように指定する。

内部ファイル ID	デバイス	ファイル種別	実ファイル名
XX-FILE-A	指定なし	IFAS FILE	C:¥DATA¥XXFILE1
YY-FILE-A	指定なし	IFAS FILE	C:¥DATA¥YYFILE1
ZZ-FILE-A	指定なし	IFAS FILE	C:¥DATA¥ZZFILE1

RDB ファイルの IFASPRO RDB を利用する場合、COBOL ファイルマッパーで指定する。一般ファイルと同様。

RDB ファイルの COBOL DB アクセスを利用する場合、COBOL DB アクセスの表対応テーブルで指定する。

例の FILE-B, FILE-C が RDB であったとき次のように指定する。

COBOL 表名(*1)	データベース名	テーブル名	サーバ名
XX-FILE-B	SAMPLEDB	XX_FILE_B	server001
YY-FILE-B	SAMPLEDB	YY_FILE_B	server001
ZZ-FILE-B	SAMPLEDB	ZZ_FILE_B	server001
FILE-C	SAMPLEDB	FILE_C	server001

(*1)内部ファイル ID に相当

FILE-C は、グローバルファイルであることに注意。

4. 環境変数の設定

環境変数 COB_CID_INIFILE に走行したいカンパニ ID の内部ファイル ID 置換表のファイル名をフルパスで設定する。

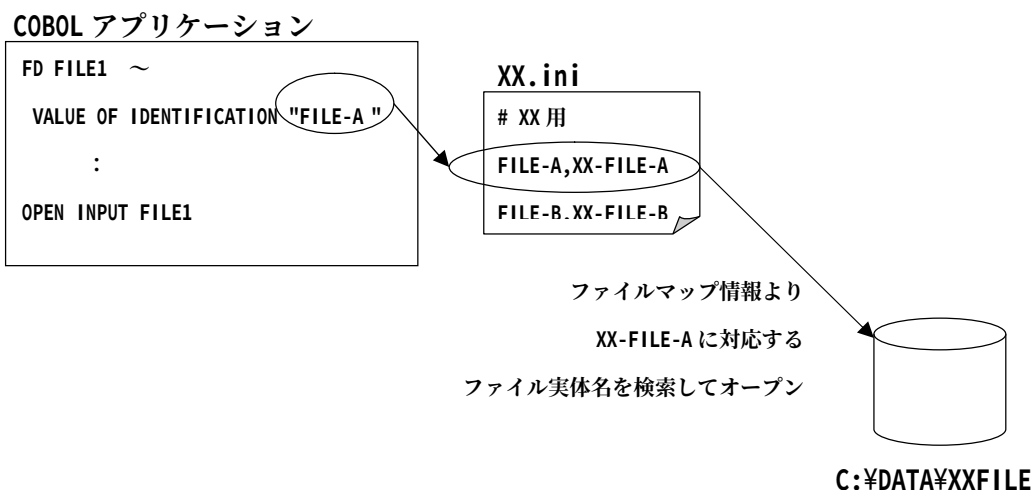
たとえば、COB_CID_INIFILE=C:¥INI¥XX.ini のように指定する。

COBOL WS エミュレータで COBOL アプリケーションを実行する場合は「WS-EML ユーザ定義ファイル」でユーザ毎に環境変数の設定が可能なので、実行ユーザが所属するカンパニ ID の内部ファイル ID 置換表を指定する。

5. COBOL アプリケーションの実行

COBOL は、ファイルのオープン時、「内部ファイル ID 置換表」の指定があれば、それに従って内部ファイル ID を変更して、ファイルをオープンする。

たとえば、あるプログラムで内部ファイル ID が FILE-A のファイルを使用し、内部ファイル ID 置換表として XX.ini を設定している場合



6. ツールや関連製品と組み合わせた COBOL アプリケーションの実行

ファイルマップ機能を使用している場合は、次のとおりである。

ファイルマップ機能は内部ファイル ID と実ファイル名を結びつける。内部ファイル ID 置換表に従い置換した内部ファイル ID にて実ファイル名を検索する。ファイルマップ機能に関連付けられている場合、COBOL ファイルマッパーの指定に従い、実ファイル名やファイル種別等を

決定する。ファイルマップ機能に関連付けられていない場合、置換した内部ファイル ID を実ファイル名として扱う。

IFAS FILE の環境設定ファイル(IFASCLT.INI)を使用している場合は、次のとおりである。
ファイルマップ機能で指定された実ファイル名を IFAS FILE/IFASPRO RDB にそのまま通知する。したがって、IFAS FILE の環境設定ファイルの指定は、ファイルマップ指定を処理後、IFAS FILE/IFASPRO RDB にて処理される。

COBOL DB アクセスの表対応テーブルを使用している場合は、次のとおりである。
ファイルマップ機能で指定された表名は、COBOL DB アクセスにそのまま通知する。したがって、DB アクセスの表対応テーブルの指定は、ファイルマップ指定を処理後、COBOL DB アクセスにて処理される。

3.15.2 内部ファイル ID 置換表

「内部ファイル ID 置換表」はテキスト形式のファイルであり、内部ファイル ID の置換を指示する。
「内部ファイル ID 置換表」の各行は次の形式のいずれかでなければならない。

・コメント行

書式	# で始まる行
説明	・任意の文字列を記載できる。 ・行頭から連続する半角空白は挿入可。以外の文字は不可 (コメント行として扱わない)。
指定例	#カンパニ ID 用テーブル

・置換指示行

書式	旧内部ファイル ID,新内部ファイル ID
説明	・内部ファイル ID に対する新たな内部ファイル ID を示す行。 ・1 行に旧内部ファイル ID と新内部ファイル ID を 1 対 1 で指定する。 1 行に 1 組を指定可能。1 行に複数組は定義できない。 ・行頭, コンマの前後, 行末の連続する半角空白は挿入可。挿入された半角空白は無視する。以外の文字は内部ファイル ID として処理する。 ・内部ファイル ID は最大 200 バイト (ファイルマップ機能に準ずる)。旧内部ファイルおよび新内部ファイル ID のいずれかが 200 バイトを超えた行は無効とする。 ・内部ファイル ID としての妥当性検査は行わない。適正でない ID は COBOL ソース記

	述の内部ファイル ID として検索できない結果となる。 ・旧内部ファイル ID または新内部ファイル ID が未指定の行は無効とする (たとえばコンマだけの行は無効)。 ・2 個以上のコンマが記述された行は無効とする。 ・旧内部ファイル ID の重複指定不可。 ・日本語が指定可。全角空白は無視せず有効文字とする。
指定例	INF1,NEW_INF1 INF1 という内部ファイル ID を NEW_INF1 という内部ファイル ID に置き換えることを指示する。

使用可能な文字コードは SJIS コードである。

半角および全角の英小文字は英大文字として扱う (COBOL ファイルマッパーの動作に合わせる)。

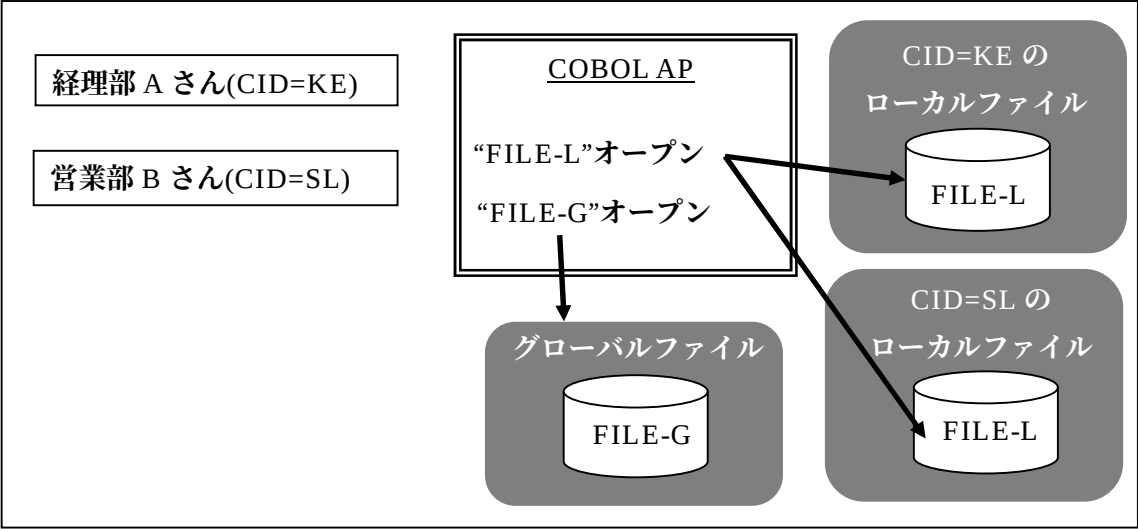
アプリケーション実行中に内部ファイル ID 置換表が更新されても反映しない。初回読み込み時の情報で処理する。

内部ファイル ID 置換表を変更するには、CS_CIDTBLCLEAR システムサブルーチンを呼び出してメモリ上に配置した内部ファイル ID 置換表を解放し、ファイルから再び読み込む必要がある。

1 行は最大 500 バイト(半角空白, 半角#, コンマ, 改行コード含む)とする。500 バイトを超えた場合は無効行とする。

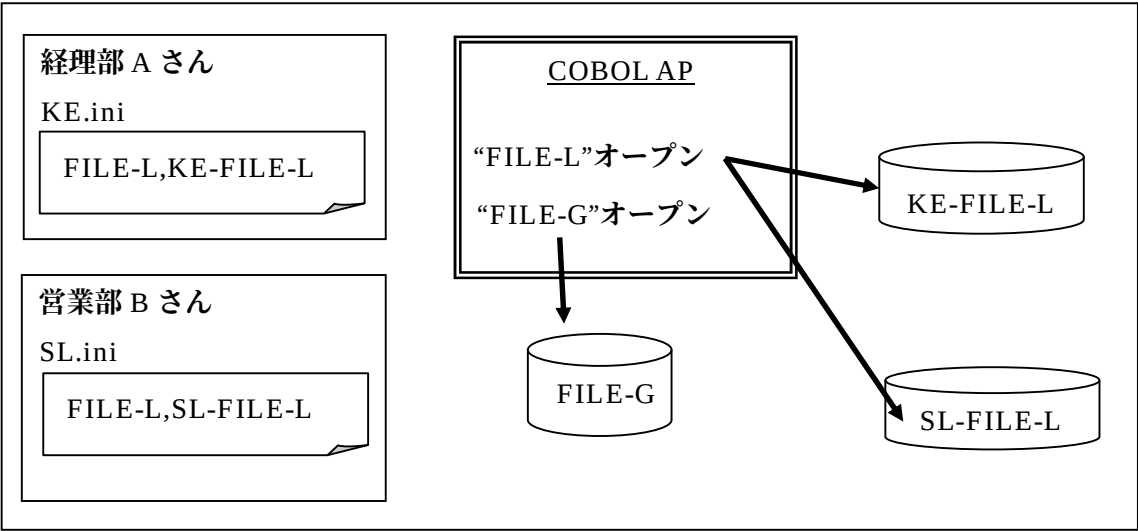
3.15.3 使用例

A-VX のシステムとして、カンパニ ID が「KE」の経理部 A さん、「SL」の営業部 B さんを例に示す。



ローカルファイル FILE-L とグローバルファイル FILE-G をオープンするアプリケーションを A さんと B さんが実行した場合、ローカルファイル FILE-L については A さんと B さんは異なる実体にアクセスし、グローバルファイル FILE-G は同じ実体にアクセスする。

このシステムは次の構成で実現できる。



A さんは環境変数 COB_CID_INIFILE に KE.ini を指定しアプリケーションを実行する。B さんは環境変数 COB_CID_INIFILE に SL.ini を指定しアプリケーションを実行する。

3.15.4 注意制限事項

- A-VX の場合, カンパニ ID なし のとき, すべてのローカルファイルを参照することができるが, 本システムの内部ファイル ID 置換機能では, 同様の機能を実現することはできない。環境変数 COB_CID_INIFILE にカンパニ ID ごとの内部ファイル ID 置換表を繰り返し設定して実行する必要がある。
- 内部ファイル ID 置換表に登録する内部ファイル ID (新旧共) にはコンマおよび前方と後方の半角空白は使用できない。
- VALUE OF 句が指定されない場合, ASSIGN 句に指定された内部ファイル名または実ファイル名を内部ファイル ID とみなす。

第4章 拡張ファイル入出力機能

4.1 概説

本章では、COBOL で使用できるファイル形式、ファイル編成およびレコード形式等について説明し、さらに、原始プログラム上でのその定義方法、使用の際の注意などについても詳細に説明してある。

4.2 ファイルの種類

次のファイルを使用することができる。

- 【Enterprise】RAF ファイル
- SYSIN／SYSOUT ファイル

これらのファイルの指定は、ファイル管理段落 (FILE-CONTROL 段落) の SELECT 句で行う。

4.3 ファイルの定義

COBOL で入出力を行うためには、まずそのファイルの属性をファイル管理記述項およびファイル記述項で定義しなければならない。ファイルが定義されると、そのファイルのコントロール情報をもつファイルテーブルが作成される。

4.3.1 ファイル識別リテラル

原始プログラムで定義したファイルを実行時に使用する実ファイルに対応させるための名前である。

SELECT ファイル名 1 ASSIGN TO ファイル識別リテラル 1

ファイル識別リテラル 1 は実行可能プログラム内では一意であり、実ファイルと 1 対 1 に対応する。

ファイル識別リテラル 1 は英字で始まる 8 文字以内の英数字でなければならない。

4.3.2 ファイルの装置およびファイル指定

ファイル管理記述項の **ASSIGN** 句によって、ファイルの装置およびファイルを指定する。形式は、次のとおりである。

<u>FILE-CONTROL.</u>	
<u>SELECT</u> ファイル名 1 <u>ASSIGN</u>	TO 内部ファイル識別名 1 TO 定数 1 <u>USING</u> {データ名 1} ...
<u>FILE SECTION.</u>	
<u>FD</u> ファイル名 1	
<u>LABEL RECORD IS STANDARD</u>	
<u>VALUE OF IDENTIFICATION</u>	データ名 2 定数 2

- 内部ファイル識別名 1 の指定
次の形式で指定する。

指定形式	識別名 /装置名	意味	備考
ファイル識別リテラル-装置名	PRU PRINTER SYSOUT	SYSOUT ファイル	
	SYSIN	SYSIN ファイル	
データ名 4-装置名	PRU PRINTER SYSOUT	SYSOUT ファイル	BASED 句指定の 場合
	SYSIN	SYSIN ファイル	
【Enterprise】 ファイル識別リテラル-識別名	RAF	RAF ファイル	
【Enterprise】	RAF	RAF ファイル	BASED 句指定の

データ名 4-識別名			場合

ファイル識別リテラルには、実ファイル名に対応する識別名を指定する。

データ名 4 には、ファイルアドレスを格納するデータポインタデータ名を指定する。

- 定数 1 の指定

英数字定数は次の形式で指定する。

指定形式	識別名 /装置名	意味	備考
【Enterprise】 “実ファイル名-識別名”	RAF	RAF ファイル	
“実ファイル名-装置名”	PRU PRINTER SYSOUT	SYSOUT ファイル	
	SYSIN	SYSIN ファイル	

実ファイル名には、デバイスタイプ上のファイル名を指定する。

(オペレーティングシステムで許されるパス名を含むファイル名が指定可能である。ただし、パス名やファイル名にハイフンを含む名前は指定できない。)

- USING データ名 1 の指定

次の形式で指定する。

指定形式	識別名 /装置名	意味	備考
【Enterprise】 データ名 6-識別名	RAF	RAF ファイル	
データ名 6-装置名	PRU PRINTER SYSOUT	SYSOUT ファイル	

データ名 6 は、デバイスタイプ上の実ファイル名を指定する。

4.4 ファイルの割り当て

プログラム内で定義したファイル識別リテラルに対して実際に使用するファイルの割り当てを行う必要がある。以降にその具体的利用方法を示す。

(1) SYSOUT ファイルの割り当て

原始プログラムの **SELECT** 句で次のように記述した場合、環境変数を指定することにより、出力ファイルを指定可能である。

SELECT ファイル名 1 **ASSIGN TO** ファイル識別リテラル 1-SYSOUT

- (a) 【Enterprise for Windows】【Standard】ファイル識別リテラル 1-SYSOUT を指定した場合、環境変数で次の設定を行う。

set ファイル識別リテラル 1_SYSOUT=出力ファイル名

例

SELECT 句の記述:

SELECT PR-FILE **ASSIGN TO** OUTFILE-SYSOUT

環境変数の形式:

set OUTFILE_SYSOUT=C:¥tmp¥OUTFILE1

↑

①

↑

②

説明

① ファイル識別リテラルに” _SYSOUT” を付加して指定する。

② C:¥tmp¥OUTFILE1 を出力ファイルの実体として指定する。

【Enterprise for Linux】ファイル識別リテラル 1-SYSOUT を指定した場合、環境変数で次の設定を行う。

export ファイル識別リテラル 1_SYSOUT=出力ファイル名

例

SELECT 句の記述:

SELECT PR-FILE **ASSIGN TO** OUTFILE-SYSOUT

環境変数の形式:

export OUTFILE_SYSOUT=/tmp/OUTFILE1

↑

①

↑

②

説明

- ① ファイル識別リテラルに” _SYSOUT” を付加して指定する。
- ② /tmp/OUTFILE1 を出力ファイルの実体として指定する。

(b) ファイル識別リテラル 1-SYSOUT を指定し、環境変数の設定がない場合、出力先は実行可能ファイルを実行したディレクトリ配下となり、出力ファイル名は次の形式となる。

- ・ プログラム定義の場合:PROGRAM-ID のプログラム名 + [pid] + “.out”
- ・ 利用者定義関数の場合:FUNCTION-ID の利用者定義関数名 + [pid] + “.out”
- ・ クラス定義の場合:CLASS-ID のクラス名 + [pid] + “.out”

ここで、[pid]は、プロセス ID を表す。

(2) SYSIN ファイルの割り当て

原始プログラムの SELECT 句で次のように記述した場合、環境変数を指定することにより、入力ファイルを指定可能である。

SELECT ファイル名 1 **ASSIGN TO** ファイル識別リテラル 1-SYSIN

(a) 【Enterprise for Windows】【Standard】ファイル識別リテラル 1-SYSIN を指定した場合、環境変数で次の設定を行う。

set ファイル識別リテラル 1-SYSIN=入力ファイル名

例

SELECT 句の記述:

SELECT IN-FILE ASSIGN TO INFILE-SYSIN

環境変数の形式:

set INFILE_SYSIN=C:¥tmp¥INFILE1

↑

①

↑

②

説明

- ① ファイル識別リテラルに” _SYSIN” を付加して指定する。
- ② C:¥tmp¥INFILE1 を入力ファイルの実体として指定する。

【Enterprise for Linux】ファイル識別リテラル 1-SYSIN を指定した場合、環境変数で次の設定を行う。

export ファイル識別リテラル 1_SYSIN=入力ファイル名

例

SELECT 句の記述:

SELECT IN-FILE ASSIGN TO INFILE-SYSIN

環境変数の形式:

```
export INFILE_SYSIN=/tmp/INFILE1
```



①

②

說明

- ① ファイル識別リテラルに”_SYSIN”を付加して指定する。
- ② /tmp/INFILE1を入力ファイルの実体として指定する。

(b) ファイル識別リテラル **1-SYSIN** を指定し、環境変数の設定がない場合の既定の入力ファイル名は次の形式となる。

- ・ プログラム定義の場合:PROGRAM-ID のプログラム名 + “.in”
- ・ 利用者定義関数の場合:FUNCTION-ID の利用者定義関数名 + “.in”
- ・ クラス定義の場合:CLASS-ID のクラス名 + “.in”

(3) 【Enterprise】RAF ファイルの割り当て

原始プログラムの **SELECT** 句で次のような記述をした場合、RAF ファイルに割り当て可能である。

SELECT ファイル名 1 ASSIGN TO ファイル識別リテラル 1-RAF

または

SELECT ファイル名 1 ASSIGN TO 定数 1

SELECT ファイル名 1 ASSIGN USING データ名 1-RAF

- ・ ファイル識別リテラル 1-RAF

refam コマンドを利用する場合

- ・ 定数 1

実ファイル名を直接記述する場合

(例) 【Enterprise for Linux】 “/home/RAF/INFILE-RAF”

【Enterprise for Windows】“C:¥tmp¥INFILE-RAF”

- ・ データ名 1-RAF

データ名に実ファイル名を指定し，入出力を行う場合

- (a) 実ファイル名にはディスク上の **RAF** ファイル名を指定する。
- (b) **RAF** ファイルは標準ラベルをもっている。すなわち，**LABEL RECORD** 句では **STANDARD** 以外の指定はできない。
- (c) **RAF** ファイルに対し，**refam** コマンドを利用する場合の **ASSIGN** 句の例を次に示す。

例

SELECT 句の記述：

SELECT OUTFILE ASSIGN TO OUTLIT-RAF

環境変数の形式：

【Enterprise for Linux】**rasetenv define -literal OUTLIT /MYJOB/MYFILE**

↑

①

↑

②

【Enterprise for Windows】**rasetenv define -literal OUTLIT D:¥MYJOB¥MYFILE**

↑

①

↑

②

説明

- ① ファイル識別リテラルはプログラム上の **SELECT** 句のファイル識別リテラルと同じものを指定する。
- ② 実ファイル名にはディスク上の **RAF** ファイル名を指定する。この場合は，**/MYJOB/MYFILE** というファイル名で確保されたファイルである。

4.5 エラー処理

COBOL プログラムで入出力エラーが発生した場合のプログラムの動作は、プログラム中でのエラー処理機能の指定の仕方や発生したエラーの種類などによって異なる。プログラム中で指定できるエラー処理の機能としては、**FILE STATUS** 句、**AT END** 句、**INVALID KEY** 句および入出力誤り手続きがある。

4.5.1 FILE STATUS 句

COBOL では、ファイルの種類に関係なく、ファイル管理記述項中に **FILE STATUS** 句を書くことによって、**OPEN** 文、**CLOSE** 文、**READ** 文、**WRITE** 文および **REWRITE** 文の各入出力動作の状態を示す値（入出力状態値および詳細情報値）を知ることができる。

FILE STATUS 句の記述は次のようになる。

FILE STATUS IS データ名 1 [, データ名 2]

- (1) データ名 1 は、2 桁の英数字データ項目として定義しなければならない。
データ名 1 には、入出力状態値が設定される。
- (2) データ名 2 は、5 桁以上の数字データ項目として定義しなければならない。
データ名 2 には、詳細情報値が設定される。
装置名が **PRU**、**PRINTER**、**SYSOUT** または **SYSIN** の場合のみ使用できる。
【Enterprise】識別名が **RAF** の場合も使用できる。
- (3) 詳細情報を必要としない場合は、データ名 2 を省略してもよい。

FILE STATUS 句で指定されたデータ名に、そのファイルに対する入出力文の実行中、また入出力誤り手続きの実行前に、各入出力動作の状態を示す値が設定される。したがって、利用者は各入出力文の実行後、**FILE STATUS** 句のデータ名 1 の値を調べることで後続の一連の処理手続きを決定することができる。

FILE STATUS 句を指定してもエラー発生時に、入出力状態を判定せず後続の一連の文を実行した場合は、エラーを正しく検出できずその後の動作も保証されない。したがって、入出力文の直後または入出力誤り手続きの中で入出力状態値を判定し、後続の処理を行わなければならない。

また、**FILE STATUS** 句にデータ名 2 を指定することにより、入出力動作の詳細な情報を得ることができる。データ名 1 の値が” 90” の場合は、データ名 2 を参照しないと正しい情報は得られな

い。

また、データ名 1 の値が” 00” の場合でも、データ名 2 には入出力動作の詳細な情報が設定される。

4.5.2 入出力状態値と意味

表 4-1 では入出力状態値および詳細情報値と意味を述べてある。

詳細情報値は、各ファイルシステムのリターンコードの右 2 バイトが 10 進数に変換されて詳細情報値が設定される項目に格納される。表 4-1 は、装置名が PRU, PRINTER, SYSOUT または SYSIN の場合に設定される入出力状態値および詳細情報値を表している。

【Enterprise】識別名が RAF の場合に設定される入出力状態値および詳細情報値も表している。

表 4-1 入出力状態値および詳細情報値と意味

大分類	入出力 状態値	詳細 情報値	リターン コード	意味
成功	00	0	DONE	入出力文は正常に完了した。
		18	ALREADY	【Enterprise】RAF ファイルがすでに存在していたため、そのファイルに対する自動作成は実行されなかった。
		22	DUMMY	【Enterprise】RAF ファイルをダミーファイルとしてオープンした。
		7169	NOTOPEN	SELECT 句または FD 句で EXTERNAL 句が指定されたファイルで、既に CLOSE 文で閉じられているのに再び CLOSE 文を実行した。
		7175	OPEN	SELECT 句または FD 句で EXTERNAL 句が指定されたファイルで、既に OPEN 文、ACCEPT 文または DISPLAY 文で開かれているのに再び OPEN を実行した。
	04	0	DONE	READ 文の実行は成功した。しかし、読み込んだレコードが、最小のレコード長より小さい、または最大のレコード長より大きい。

大分類	入出力 状態値	詳細 情報値	リターン コード	意味
成功	05	0	DONE	OPEN 文の実行は成功した。しかし、OPEN 文を実行したとき、参照した不定ファイルは存在しない。OPEN 文のモードが入出力または拡張ならば、ファイルが生成される。
ファイル終了 条件 (不成功)	10	39	DATALIM	- 順呼び出しの順次処理において書き方 1 の READ 文実行時にファイルの終わりが検出された。 - 順呼び出しの READ 文が、存在しない不定入力ファイルに対して初めて行われた。 備考 AT END 状態(リターンコード DATALIM)が検出された後さらに、そのファイルに対する順次処理の READ 文を実行した場合、異常終了(入出力状態値は 46, 詳細情報値は 7179, リターンコード EXHAUST)する。
永続誤り条件 (不成功)	30	1153	IOFAIL	ソフトウェア側での入出力障害を検出した。または継続不可能とソフトウェア側で判断した。
		1159	ABNCCAD	1. 判別不可能な入出力終了コードが戻された。
		7685	UNRECIO	【Enterprise】RAF ファイルで、回復不可能な入出力エラーが発生した。
	34	4648	FILEOV	出力モードで開かれた順ファイルの WRITE 文において、ファイルの領域を超えてレコードを書き出そうとしている。
	35	4103	EFNUNKN	不定ファイルでないファイル (non optional file) が存在しないとき、INPUT 句、I-O 句または EXTEND 句のある OPEN 文を実行しようとした。
	37	5219	PMDVIOL	【Enterprise】OPEN 文を実行しようとしたが、指定された RAF ファイルが OPEN 文に書かれたモードを使えない。
	38	2313	IFNNASG	LOCK 指定で CLOSE したファイルに対して、OPEN 文を実行しようとした。

大分類	入出力 状態値	詳細 情報値	リターン コード	意味
永続誤り条件 (不成功)	39	1120 6151	CONFLICT LNERR	固定ファイル属性とプログラム中に指定した属性の間で矛盾が検出されたため、OPEN 文が不成功となった。 ・ファイル編成が正しくない。 ・アクセスモードが正しくない。 ・最大レコード長が等しくない。ただし、内部ファイル名が SIN および SPR、装置名が SYSIN および SYSOUT のファイルに対しては返却されない。 【Enterprise】RAF ファイルで、レコード長検証機能を無効にした場合は返却されない。
論理誤り条件 (不成功)	41	7175	OPEN	ファイルが既に OPEN されているのに、 OPEN 文を実行しようとした。
	42	7169	NOTOPEN	開かれていない状態のファイルに CLOSE 文を実行しようとした。
	43	7172	NOCURREC	【Enterprise】 1. RAF ファイルで、READ 文を実行せずに REWRITE 文を実行した。 2. RAF ファイルで、削除レコードに対して REWRITE 文を実行した。

大分類	入出力 状態値	詳細 情報値	リターン コード	意味
論理誤り条件 (不成功)	44	6151	LNERR	順ファイルに対して 1. 関連するファイル名の RECORD IS VARYING 句 (レコードは可変) によって許される最大のレコードより大きい, または最小のレコードより小さいレコードを, 書き出そうとした。 2. WRITE 文の実行において, RECORD DEPENDING ON 句のデータ名の値が, RECORD 句とレコード記述項によって決定された最大レコード長より大きい, または最小レコード長より小さい。 3. レコードを書き換えようとしたが, レコードが書き換えられるレコードと同じ大きさでない。
	46	7179	EXHAUST	入力モードまたは入出力両用モードで開かれたファイルに順呼び出しの READ 文を実行しようとしたが, 有効な次のレコードがない。次の場合のいずれかである。 1. 先行する READ 文がファイル終了条件を引き起こした。 2. ファイルの終わりが検出されたのに, 順次処理の READ 文を実行した。

大分類	入出力 状態値	詳細 情報値	リターン コード	意味
論理誤り条件 (不成功)	47	7173	WRONGPMD	入力モードまたは入出力両用モードで開かれていないファイルに, READ 文または START 文を実行しようとした。
	48	7173	WRONGPMD	入出力両用モード, 出力モードまたは拡張モードで開かれていないファイルに, WRITE 文を実行しようとした。
	49	7173	WRONGPMD	【Enterprise】入出力両用モードで開かれていない RAF ファイルに, REWRITE 文を実行しようとした。
その他の誤り (不成功)	90	512	BUSY	【Enterprise】他プロセスが共用できないモードでオープン中の RAF ファイルをオープンしようとした。
		1792	NOWAIT	【Enterprise】他のプロセスが同時に同一の RAF ファイルにアクセスを行っている。
		4103	EFNUNKN	指定された媒体上に割り当てられた外部ファイルが存在しない。
		5218	SHLVVIOL	【Enterprise】RAF ファイルに対し, OPEN 文で, 許されない処理モードを指定した。
		7169	NOTOPEN	ファイルが OPEN されていない。
		7178	NOTOBS	【Enterprise】RAF ファイルの保存期間が満了していない。
		上記 以外	上記 以外	【Enterprise】RAF ファイルの詳細情報値は, refam の戻り値を意味する。 『refam 利用の手引』を参照。
	95	12288	ILLCODE	コード変換が失敗した。
	96	2313	IFNNASG	【Enterprise】RAF ファイルが定義されていない。

4.5.3 入出力動作状態の調べ方

FILE STATUS 句の指定をすることによって入出力動作の状態を知ることができる。ここでは、入出力動作の状態の調べ方を例にあげて説明する。

```
FILE STATUS IS STATUS-1, STATUS-2
```

上記のように記述されているとき

(1) USE 文が指定されていない場合

```

      READ IS-F AT END GO TO ATEND-RTN.
      ①IF STATUS-1 = "90" GO TO ERR-RTN.
        :
        :
      ATEND-RTN.
        DISPLAY "AT END" .
        :
        :
      ERR-RTN.
        ②DISPLAY "IS-F ERROR STATUS-2 =" STATUS-2.
        :
        :
```

説明

- ① READ 文が正常終了したかどうか入出力状態値を調べる。
- ② 詳細情報値を表示して異常終了の原因を調べる。

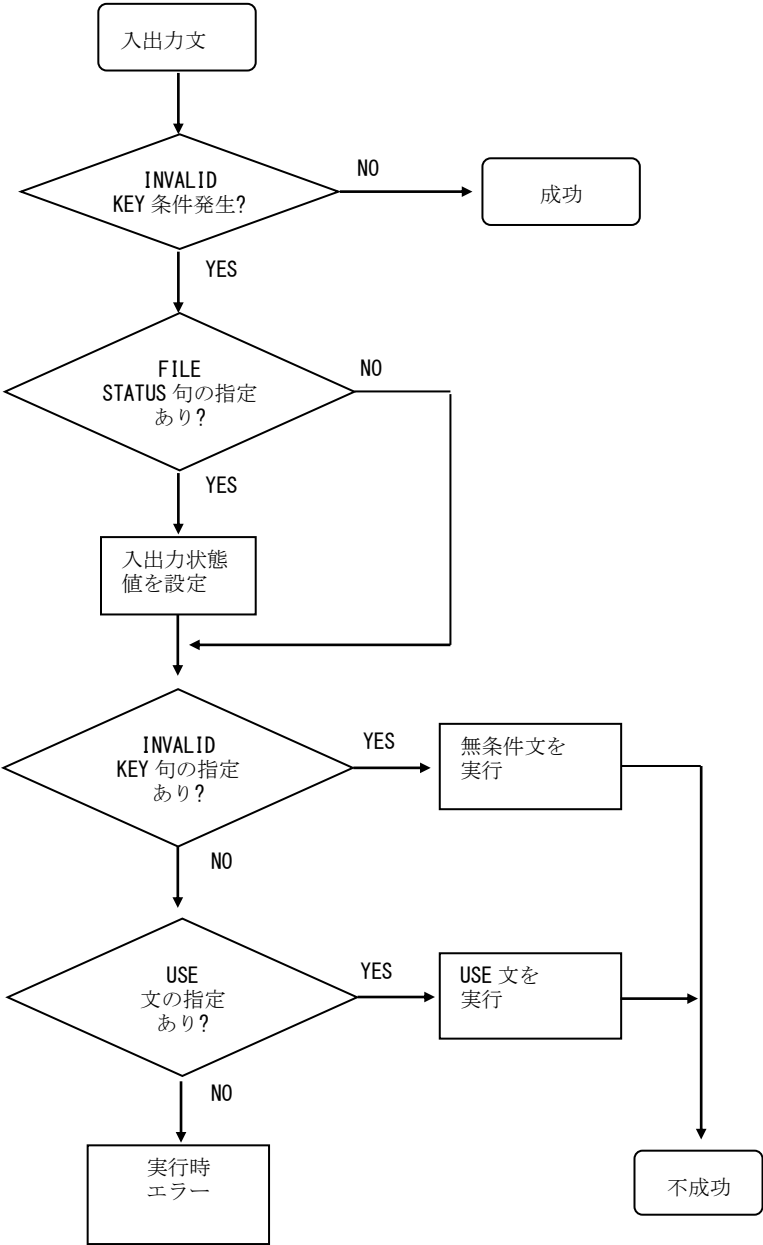
(2) USE 文が指定された場合

```
PROCEDURE DIVISION.  
DECLARATIVES.  
USE-1 SECTION.  
    USE AFTER STANDARD ERROR PROCEDURE ON IS-FILE.  
USE-RTN.  
    ①IF STATUS-1 = "90" GO TO ERR-RTN.  
    ②IF STATUS-1 = "10" ~  
        :  
        :  
ERR-RTN.  
    ③DISPLAY "ERROR STATUS-2 =" STATUS-2.  
        :  
        :  
END DECLARATIVES.  
USE-2 SECTION.  
    :  
    :  
    READ IS-FILE.  
    :  
    :
```

説明

- ① STATUS-2 に詳細情報が格納されているかどうか調べる。
- ② 起こり得る AT END 状態などを調べる。
- ③ 詳細情報の値を表示して異常終了の詳細な原因を調べる。

図 4-1 入出力文の終了状態の調べ方



4.5.4 AT END 句

READ 文を実行中にファイルの終わりが検出された場合など AT END 条件が起こることがある。その AT END 条件について述べる。

(1) AT END 条件の起こる原因

次のような場合に AT END 条件が起こる。

(a) 大容量記憶ファイルの場合

順呼び出しまたは動的呼び出しの順次処理で、ファイルの終わりが検出されたとき。

(b) SYSIN ファイルの場合

READ 文実行中に、SYSIN ファイルの終わりが検出されたとき。

(2) AT END 条件の動作

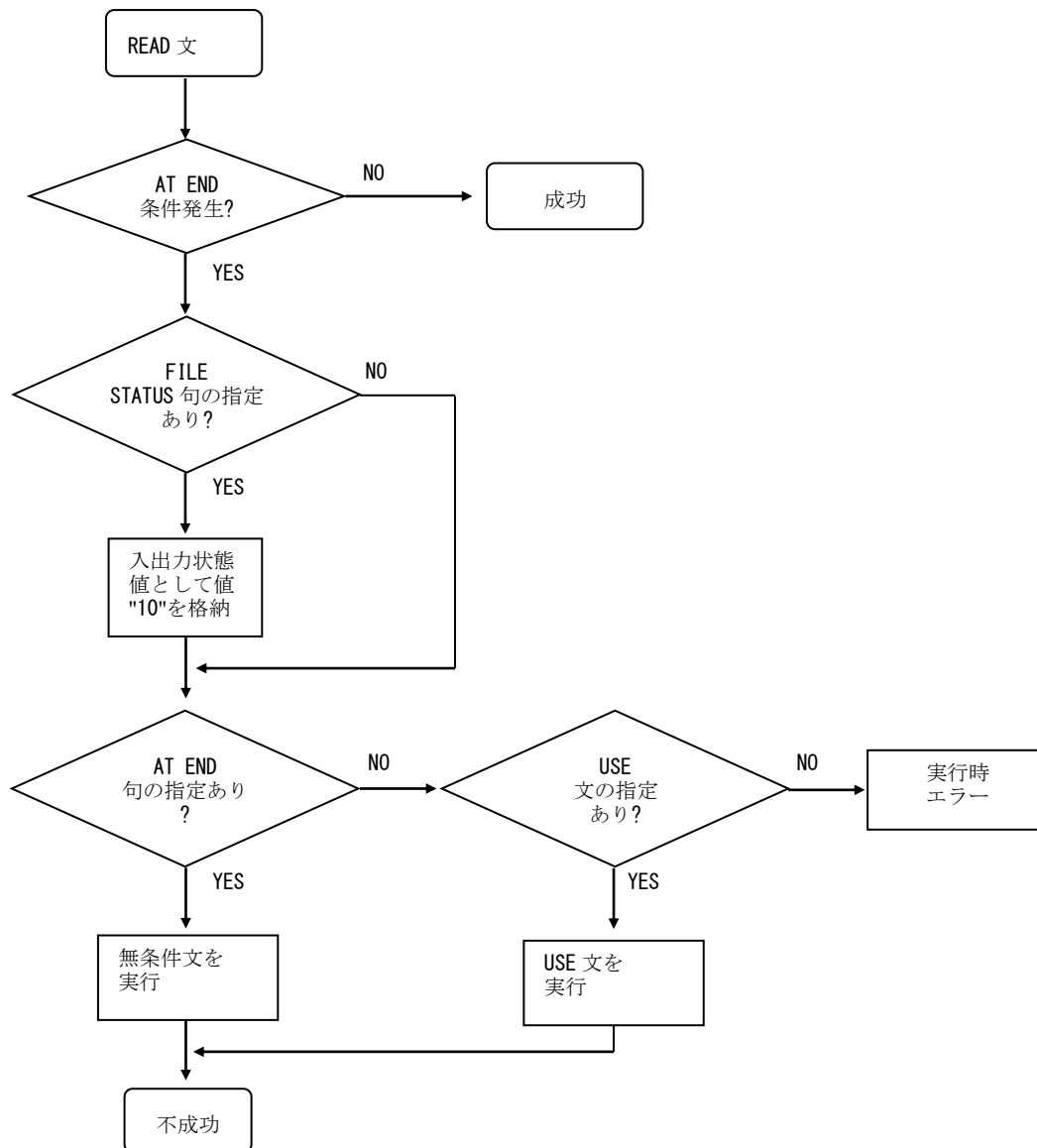
AT END 条件が発生した場合の動作は図 4-2 のようになる。

(3) AT END 条件後の READ 文

AT END 条件が起こった後は、次のいずれかの処理を実行しなければ、そのファイルに対する順次処理の READ 文(書き方 1)は使用できない。

(a) ファイルを閉じ、その後で再び開く。

図 4-2 AT END 条件の動作



4.5.5 入出力誤り手続き

入出力文実行中に入出力誤りが発生したり、ユーザラベル処理の要求があった場合に **USE** 文が実行されるが、ここでは **USE** 文が実行される条件について述べる。ただし、ユーザラベルに関してはここではふれない。

(1) **USE** 文が実行される条件

次の場合に **USE** 文が実行される。

- (a) **AT END** 条件が発生し、**AT END** 句の指定がない場合。
- (b) 入出力文 (**OPEN** , **CLOSE** 文を含む) が異常終了 (入出力状態値が "00" , "02" , "04" , "05" 以外のとき) したとき。

(2) **USE** 文への制御

- (a) ファイル名が指定されている場合は、そのファイルに対して前記 (1) の条件が発生した時点で、**FILE STATUS** 句に入出力状態値および詳細情報値が設定され (**FILE STATUS** 句の指定があるとき)、その **USE** 手続きが実行される。
- (b) **INPUT** (入力)、**I-O** (入出力)、**OUTPUT** (出力) および **EXTEND** (拡張) 句が指定されている場合は、前記 (1) の条件が発生した時点で、**FILE STATUS** 句に入出力状態値および詳細情報値が設定され (**FILE STATUS** 句の指定があるとき)、そのファイルの開かれていたモードに従って、対応する **USE** 手続きが実行される。

例

PROCEDURE DIVISION.

DECLARATIVES.

USE-INPUT SECTION. USE AFTER STANDARD ERROR

 PROCEDURE ON INPUT.

USE-RTN.

① IF FILE-ID = "A"

 DISPLAY "FILE-A ERROR "

 "STATUS-1 =" STATUS-1A

 ELSE DISPLAY "FILE-B ERROR "

 "STATUS-1 =" STATUS-1B.

END DECLARATIVES.

:

 OPEN INPUT FILE-A.

```

        MOVE "A"    TO FILE-ID.
②  READ FILE-A.
        :
        OPEN INPUT FILE-B.
        MOVE "B"    TO FILE-ID.
③  READ FILE-B.
        :

```

説明

- ① **FILE-A** または **FILE-B** どちらで入出力エラーとなったのかを調べ、該当するファイルの入出力状態値を **SYSOUT** ファイルに出力する。
- ② 入出力エラーとなると、**INPUT** 句指定のある **USE** 文(節名=**USE-INPUT**)を実行する。
- ③ 入出力エラーとなると、②と同じように **INPUT** 句指定のある **USE** 文(節名=**USE-INPUT**)を実行する。
- (c) 前記 (2) (b) でファイルが正常に開かれていない、または閉じた状態であるならば、**INPUT**, **OUTPUT** および **EXTEND** 句が指定されている **USE** 手続きはすべて指定がないものとして扱われる。したがって、この場合 **FILE STATUS** 句およびファイル名指定の **USE** 手続きがなければ、実行時エラーメッセージが出力され、そのタスクはアボートする。
- (d) **INPUT**, **OUTPUT** および **EXTEND** 句が指定されている **USE** 手続きと、ファイル名指定の **USE** 手続きがともに指定されているファイルは、ファイル名指定の **USE** 手続きが実行され、**INPUT**, **OUTPUT** および **EXTEND** 句指定の **USE** 手続きは無視される。

例

```

PROCEDURE DIVISION.
DECLARATIVES.
①  USE-FILE SECTION. USE AFTER STANDARD ERROR
    PROCEDURE ON FILE-1.
USE-RTN1.
(FILE-1 ファイルのエラー処理)
②  USE-INPUT SECTION. USE AFTER STANDARD ERROR
    PROCEDURE ON INPUT.
USE-RTN2.
(入力モードで開かれているファイルのエラー処理)
END DECLARATIVES.
        :
        OPEN INPUT FILE-1.

```


② READ FILE-1.

:

説明

- ① FILE-1 に入出力エラーが起こると、この USE 手続きが実行される。
- ② 入力モードで開かれているすべてのファイルに入出力エラーが起こると、この USE 手続きが実行される。
- ③ 入出力エラーが起こると、このファイルは入力モードで開かれているが、ファイル名指定の USE 手続きが指定されているので、①の USE 文を実行する。

4.5.6 入出力誤り状態

入出力文の実行が不成功(入出力状態値が 30, 34, 35, 37, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 90, 95, 96 のとき)に終わり、FILE STATUS 句も、入出力誤り用の USE 文もない場合、標準エラー出力にその誤りに関する情報を出力し、実行は異常終了する。

この情報から内部ファイル識別名、エラーの種類およびエラーが発生した入出力文の行番号を知ることができる。またそのステップの終了コードとしてそのエラーを意味するシンボリック文字が表示される。

4.6 レコード形式

COBOL で扱えるレコード形式は、固定長レコード形式および可変長レコード形式の 2 つに大別できる。なお、次の説明を明確にするために、ここで使用しているブロックおよびレコードの用語について定義しておく。

(1) ブロック

論理的に処理されるデータ(レコード)の集合を示し、1 回の入出力文(周辺装置に対して与える文)で読み書きされる外部記憶媒体上の物理単位である。

(2) レコード

情報の論理単位を示し、データファイルとして使われるときは、論理情報の最大単位を示す。また、レコードはいくつかの基本項目から構成され、また、基本項目がいくつか組み合わされて集団項目を構成し、さらに、集団項目がいくつか集まってレコードとなる。

レコードはデータ管理の論理レコードを意味する。

ファイル中のすべてのレコードの桁数が等しいときには固定長、異なるときには可変長のレコー

ドという。

レコードは COBOL の入出力文 (READ, WRITE 文等) で処理される単位である。次に、固定長レコード形式および可変長レコード形式の詳細と、原始プログラム上での記述方法について説明する。

4.6.1 固定長レコード

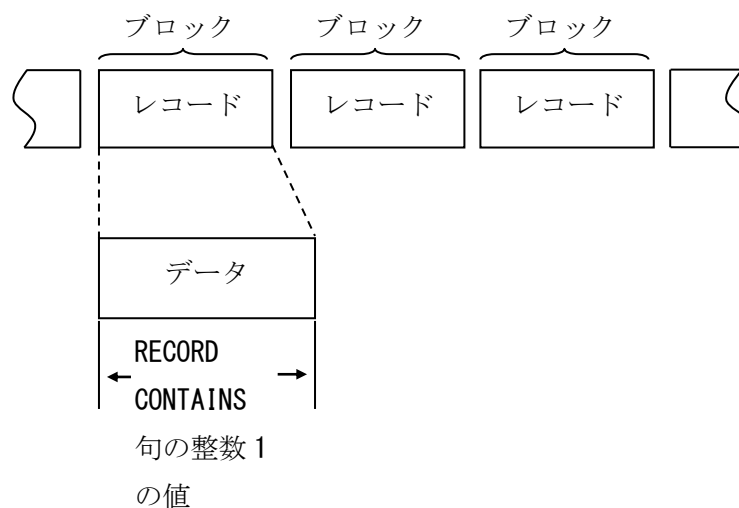
COBOL で扱う固定長レコードは、ブロック化していない固定長レコード (固定長非ブロック化レコード) と、ブロック化された固定長レコード (固定長ブロック化レコード) に分けられる。

(1) 固定長非ブロック化レコードファイル (F 形式)

固定長非ブロック化レコード形式とは、ファイル中の各レコードの桁数がすべて等しく、かつ、1 つのレコードからブロックが構成されているものをいう。このレコード形式は、すべてのファイルにおいて使用可能である。

(a) 形式

図 4-3 固定長非ブロック化レコード形式



(b) 記述様式

固定長非ブロック化レコードファイルを定義するためには、次の点に注意しなければならない。

- i. **BLOCK CONTAINS** 句
BLOCK CONTAINS 句は省略するか、または次のように記述する。

BLOCK CONTAINS 1 RECORDS

ii. RECORD 句

次の場合、レコード形式は固定長レコードとなる。

- ・RECORD 句の書き方 1 を記述したとき
- ・RECORD 句を省略, または RECORD 句の書き方 3 を記述して,
レコード記述項が単一レコードで, かつ固定長のとき

例 RECORD CONTAINS 80 CHARACTERS (書き方 1)

...

01 RECORD1 PIC X(20).

01 RECORD2 PIC X(80).

レコードの入出力は最大レコード長 (80 バイト) で行われる。

iii. レコード記述

RECORD 句の書き方 1 を記述した場合, 01 レコードを 2 個以上記述したいときには, 各 01 レコードの桁数はすべて等しくなくてもよい。

RECORD 句を省略, または RECORD 句の書き方 3 を記述した場合は, 01 レコードは単一で, 固定長でなければならない。

固定長非ブロック化レコードファイルの記述形式は, 次のようになる。

SELECT ファイル名 1 ~

:

FD ファイル名 1

[BLOCK CONTAINS 1 RECORDS]

RECORD CONTAINS 整数 1 CHARACTERS

[LABEL RECORD 句]

[DATA RECORD 句]

01 データ名 1

:

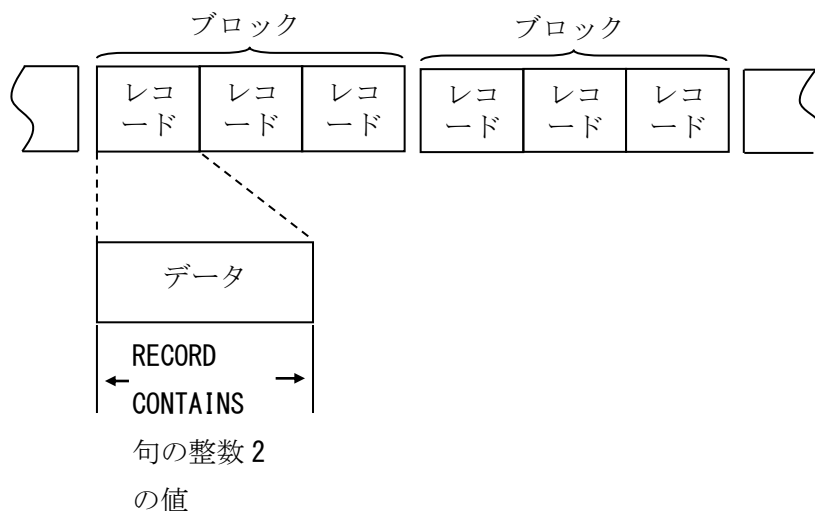
(2) 固定長ブロック化レコードファイル (FB 形式)

固定長ブロック化レコード形式とは, ファイル中の各レコードの桁数がすべて等しく, かつ 2 個以上の同じレコード数のレコードからブロックが構成されているものをいう。ファイル中の各ブロックの長さも一定である。

【Enterprise】この形式は, RAF ファイルで使用可能である。

(a) 形式

図 4-4 固定長ブロック化レコード形式



(b) 記述様式

固定長ブロック化レコードファイルを定義するためには次の点に注意しなければならない。

i. BLOCK CONTAINS 句

BLOCK CONTAINS 句は必要であり、次のように記述する。

BLOCK CONTAINS 整数 1 RECORDS

整数 1 は 2 以上でブロック内のレコード数を指定する。

ii. RECORD 句

次の場合、レコード形式は固定長レコードとなる。

・RECORD 句の書き方 1 を記述したとき

・RECORD 句を省略、または RECORD 句の書き方 3 を記述して、

レコード記述項が単一レコードで、かつ固定長のとき

例 RECORD CONTAINS 80 CHARACTERS (書き方 1)

...

01 RECORD1 PIC X(20).

01 RECORD2 PIC X(80).

レコードの入出力は最大レコード長 (80 バイト) で行われる。

iii. レコード記述

RECORD 句の書き方 1 を記述した場合、01 レコードを 2 個以上記述したいときには、

各 01 レコードの桁数はすべて等しくなくてもよい。

RECORD 句を省略, または RECORD 句の書き方 3 を記述した場合は, 01 レコードは単一で, 固定長でなければならない。

固定長ブロック化レコードファイルの記述形式は, 次のようになる。

SELECT ファイル名 1 ～

:

FD ファイル名 1

BLOCK CONTAINS 整数 1 RECORDS

RECORD CONTAINS 整数 2 CHARACTERS

[LABEL RECORD 句]

[DATA RECORD 句]

01 データ名 1

:

4.6.2 可変長レコード

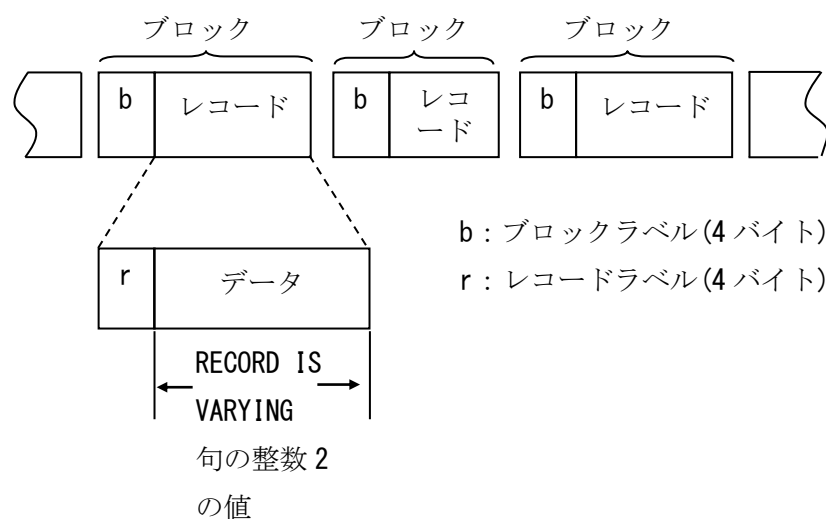
COBOL で扱う可変長レコードは、ブロック化されていない可変長レコード(可変長非ブロック化レコード)とブロック化された可変長レコード(可変長ブロック化レコード)に分けられる。

(1) 可変長非ブロック化レコードファイル(V 形式)

可変長非ブロック化レコード形式とは、ファイル中の各レコードの桁数が一定でなく、かつ、1 つのレコードからブロックが構成されているものをいう。各レコードの最初には、レコードの長さを示すレコードラベルがある。このレコードラベルはプログラムで参照することはできずデータ管理で自動的に処理される。またファイル中の各ブロックの長さも一定ではない。

(a) 形式

図 4-5 可変長非ブロック化レコード形式



(b) 記述様式

可変長非ブロック化レコードファイルを定義する際には次の点に注意しなければならない。

i. BLOCK CONTAINS 句

BLOCK CONTAINS 句は省略するか、または次のように記述する。

BLOCK CONTAINS 1 RECORDS

ii. RECORD 句

次の場合、レコード形式は可変長レコードとなる。

・RECORD 句の書き方 2 を記述したとき

・RECORD 句を省略, または RECORD 句の書き方 3 を記述して,
レコード記述項が複数レコード, または可変長集団項目のとき

例 RECORD IS VARYING IN SIZE

FROM 整数 1 TO 整数 2 CHARACTERS

DEPENDING ON データ名 1 (書き方 2)

...

01 RECORD1 PIC X(80).

または,

RECORD CONTAINS 整数 3 TO 整数 4 (書き方 3)

RECORD 句の書き方 2 を記述した場合, 整数 1 と整数 2 は, 最小のレコードの文字数と最大のレコードの文字数を表す (レコードラベルの桁数は加えない)。

DEPENDING ON 句が指定された場合には, WRITE 文の実行時にデータ名の内容で示された桁数のレコードが書き出され, READ 文の実行時には, 読み込まれたレコードの桁数がデータ名に格納される。このとき, レコードラベルの桁数は READ 文および WRITE 文で考慮しなくてよい。データ名 1 は符号なし整数項目であれば, その用途はなんでもよく, 自動的にレコードの桁数をデータ名の用途に合った形で変換して格納する。

また, RECORD 句の書き方 3 を記述した場合, 整数 3 と整数 4 は, 最小のレコードの文字数と最大のレコードの文字数を表す (レコードラベルの桁数は加えない)。

各 01 レコードの桁数は整数 3 と整数 4 の範囲内であれば, 等しくなくてもよい。

RECORD 句を省略した場合は, 各 01 レコードの桁数は等しくなくてもよい。

iii. レコード記述

RECORD 句の書き方 3 を記述した場合, または RECORD 句を省略した場合は, 桁数の異なった 01 レコードを 2 個以上指定しなければならない。可変長非ブロック化レコードファイルの記述形式は次のようになる。

SELECT ファイル名 1 ～

:

FD ファイル名 1

[BLOCK CONTAINS 1 RECORDS]

RECORD IS VARYING IN SIZE

[[FROM 整数 1] [TO 整数 2] CHARACTERS]

[DEPENDING ON データ名 1]

[LABEL RECORD 句]

[DATA RECORD 句]

01 データ名 2

:

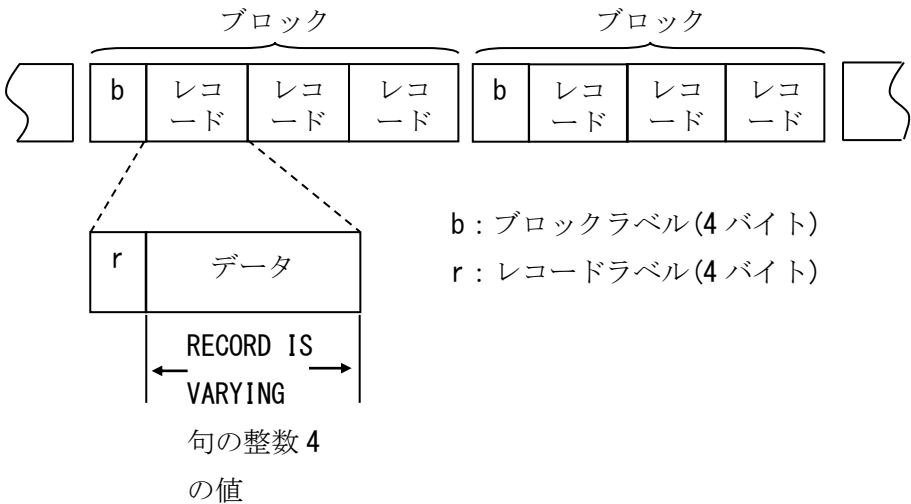
(2) 【Enterprise】可変長ブロック化レコードファイル (VB 形式)

可変長ブロック化レコード形式とは、ファイル中の各レコードの桁数が一定でなく、かつ、複数個 (1 個でもよい) のレコードからブロックが構成されているものをいう。各レコードの最初には、レコードの長さ (レコードラベルを含む) を示すレコードラベルがあり、各ブロックの最初には、ブロックの長さ (ブロックラベルをも含む) を示すブロックラベルがある。このブロックラベルや、レコードラベルはプログラムで参照することはできずデータ管理で自動的に処理される。また、レコードラベルおよびブロックラベルの桁数等について、プログラム上で考慮する必要はない。この形式は、RAF ファイルで使用可能である。

(a) 形式

可変長ブロック化レコードの形式を図 4-6 に示す。

図 4-6 可変長ブロック化レコード形式



(b) 記述様式

可変長ブロック化レコードファイルを定義する際には、次の点に注意しなければならない。

i. BLOCK CONTAINS 句

BLOCK CONTAINS 句が必要であり、次のように記述する。

BLOCK CONTAINS [整数 1 TO] 整数 2 { RECORD
CHARACTERS }

CHARACTERS という語を使ったときの整数 1 および整数 2 は、最小のブロックおよび最大

のブロックの大きさを格納するのに必要なバイト数を指定する。この場合、レコードラベルのことも考慮しなければならない。

RECORDS という語を使ったときの整数 2 は、最大のブロック内のレコード数を指定する。ただし、レコード数を指定した場合でも、各ブロック内のレコード数は一定にはならず、コンパイラは **CHARACTERS** の場合と同じに扱う。なお、整数 1 はメモである。

ii. **RECORD** 句

次の場合、レコード形式は可変長レコードとなる。

- ・**RECORD** 句の書き方 2 を記述したとき
- ・**RECORD** 句を省略、または **RECORD** 句の書き方 3 を記述して、
レコード記述項が複数レコード、または可変長集団項目のとき

例 **RECORD IS VARYING IN SIZE**

FROM 整数 1 **TO** 整数 2 **CHARACTERS**
DEPENDING ON データ名 1 (書き方 2)

...

01 **RECORD1** **PIC** X(80).

または、

RECORD CONTAINS 整数 3 **TO** 整数 4 (書き方 3)

RECORD 句の書き方 2 を記述した場合、整数 1 と整数 2 は、最小のレコードの文字数と最大のレコードの文字数を表す (レコードラベルの桁数は加えない)。**DEPENDING ON** 句が指定された場合には、**REWRITE** 文および **WRITE** 文の実行時にデータ名の内容で示された桁数のレコードが書き出され、**READ** 文の実行時には、読み込まれたレコードの桁数がデータ名に格納される。このとき、レコードラベルの桁数は **READ** 文、**REWRITE** 文および **WRITE** 文で考慮しなくてよい。データ名 1 は符号なし整数項目であれば、その用途はなんでもよく、自動的にレコードの桁数をデータ名の用途に合った形で変換して格納する。

また、**RECORD** 句の書き方 3 を記述した場合、整数 3 と整数 4 は、最小のレコードの文字数と最大のレコードの文字数を表す (レコードラベルの桁数は加えない)。各 01 レコードの桁数は整数 3 と整数 4 の範囲内であれば、等しくなくてもよい。

RECORD 句を省略した場合は、各 01 レコードの桁数は等しくなくてもよい。

iii. レコード記述

RECORD 句の書き方 3 を記述した場合、または **RECORD** 句を省略した場合は、桁数の異なった 01 レコードを 2 個以上指定しなければならない。可変長ブロック化レコードファイルの記述形式は次のようになる。

SELECT ファイル名 1 ~

```

:
FD ファイル名 1
  [BLOCK CONTAINS [整数 1 TO] 整数 2 {RECORDS|CHARACTERS}
    RECORD IS VARYING IN SIZE
    [[FROM 整数 3] [TO 整数 4] CHARACTERS]
    [DEPENDING ON データ名 1]
  [LABEL RECORD 句]
  [DATA RECORD 句]
01 データ名 2
:

```

4.7 レコード長／ブロック長

SYSIN, **SYSOUT**, および **RAF** ファイルのレコード長およびブロック長は、次のように計算されてファイルテーブルに格納される。

(1) レコード長

通常 **RECORD CONTAINS** 句で指定した値がとられる。

(2) ブロック長

(a) 固定長非ブロック化レコード

レコード長と同じ値がとられる。

(b) 固定長ブロック化レコード (**RAF** ファイルの場合)

(レコード長) × (**BLOCK CONTAINS 1 RECORD**)

(c) 可変長非ブロック化レコード

レコード長 (**SYSIN**, **SYSOUT** ファイルの場合)

レコード長+4 (**RAF** ファイルの場合)

(d) 可変長ブロック化レコード (**RAF** ファイルの場合)

BLOCK CONTAINS 句を **RECORDS** で指定した場合

(レコード長+4) × (**BLOCK CONTAINS 1 RECORD**)

BLOCK CONTAINS 句を **CHARACTERS** で指定した場合

BLOCK CONTAINS で指定した値

【Enterprise】ブロック長は **RAF** ファイルを作成するときの既定値として扱う。

既に存在する場合はそのファイルのブロック長に従う。

4.8 RAF ファイル

【Enterprise】RAF ファイルについて、以下に説明する。

COBOL で扱うファイル編成は、順ファイルだけである。

以降で、RAF ファイルの形式、プログラムでの定義方法、および入出力文等について詳細に説明する。

4.8.1 削除レコード

RAF ファイルでは、削除レコードは実際にファイルから物理的にレコードを消してしまうのではなく、レコードの先頭が `FF16` かどうかという論理的削除という方法を使用している。ただし、ファイルを削除レコードが存在しない (NODELR 指定) ファイルとして作成した場合は、レコードの先頭が `FF16` であっても有効なレコードとみなされる。

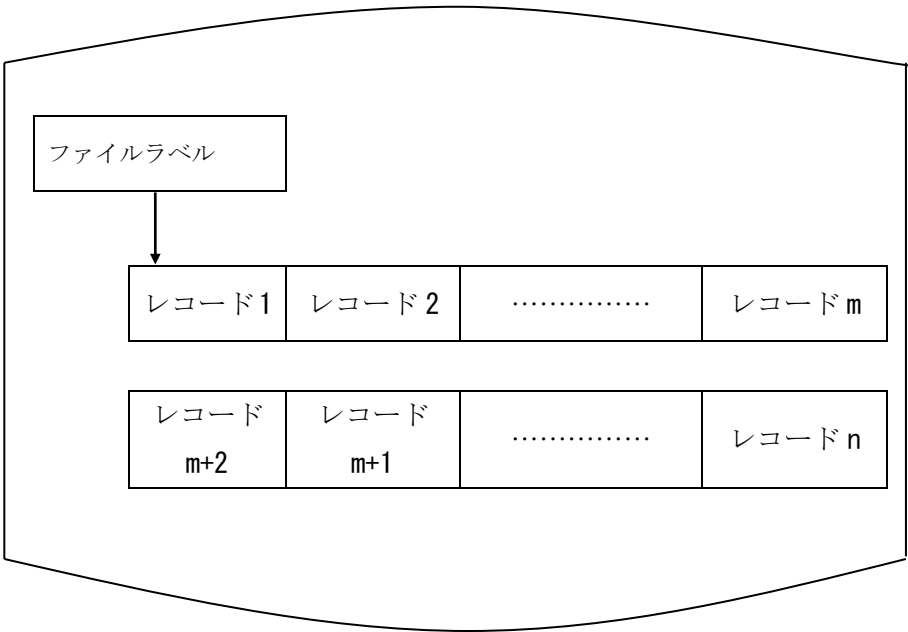
4.8.2 順ファイル

順ファイルは、各レコードを物理的な順序に従って呼び出す。したがって、データの処理はファイルが作成されたときと同じ順序で行われる。この編成は、ファイルのすべてのレコードを順番に頭から処理する業務に適している。

(1) ファイルの構造

順ファイルはディスク上に作成され、その構造はファイルの最初から順番にレコードが格納されている。

図 4-7 順ファイルの構造



(2) レコードの形式

順ファイルのレコード形式は、「4.6 レコード形式」の項で述べたすべての形式が可能である。すなわち、次の 4 つである。

- ・ 固定長非ブロック化レコード形式
- ・ 固定長ブロック化レコード形式
- ・ 可変長非ブロック化レコード形式
- ・ 可変長ブロック化レコード形式

各レコード形式を定義する際の記述様式については、「4.6 レコード形式」を参照。

(3) 順ファイルの定義

順ファイルのファイル管理記述項は、次のようになる。

```
SELECT ファイル名
      ASSIGN TO ファイル識別リテラル 1 [-装置名]
      [ORGANIZATION IS SEQUENTIAL]
      [ACCESS MODE IS SEQUENTIAL]
      [FILE STATUS IS データ名 1 [, データ名 2] ]
```

説明

- ・ **ORGANIZATION** 句で、**SEQUENTIAL**と指定するか、または**ORGANIZATION** 句を指定しないと順ファイルとなる。
- ・ **FILE STATUS** 句で指定したデータ項目には、入出力文が実行されたときの状態が格納される。

(4) 順ファイルの入出力文

順ファイルで利用できる入出力文には、次のものがある。

- ・ **OPEN** 文
- ・ **CLOSE** 文
- ・ **READ** 文
- ・ **WRITE** 文
- ・ **REWRITE** 文
- ・ **USE** 文(「4.5.5 入出力誤り手続き」を参照)

(a) OPEN 文

OPEN 文はファイルを使用可能にする。しかし、最初のレコードを呼んだり書いたりはない。また、OPEN 文のモードによって使用できる入出力文は限られる(表 4-2 参照)。

(b) CLOSE 文

標準のファイル閉じが実行される。

(c) READ 文

READ 文は、OPEN 文のモードが入力モード (INPUT)、入出力両用モード (I-O) で OPEN されたファイルだけに使用できる。その動作は、データを順次読み込んでくる。すなわち次のようになる。

```

R E A D          n 番目のレコードを読む。
  ↓
R E A D          n + 1 番目のレコードを読む。
  ↓
R E A D          n + 2 番目のレコードを読む。
  :
```

また READ 文では、さらに次のようなことが行われる。

- ・ READ 文実行中データの終わりが見つかり、AT END 句の指定があればその無条件文を実行する。AT END 句がない場合は、そのファイルに関する USE 手続きが実行される。
- ・ 可変長レコード形式の場合で、DEPENDING ON データ名の指定があると、データ名で示すデータ項目に、読み込んだレコードの長さを格納する。

(d) WRITE 文

WRITE 文は、OPEN 文のモードが出力 (OUTPUT) および拡張 (EXTEND) で OPEN されたファイルにだけ使用できる。そしてその動作は磁気テープファイルの WRITE 文と同様であり、データを順次書き込んでいく。すなわち、次のようになる。

```

W R I T E        n 番目のレコードを書き出す。
  ↓
W R I T E        n + 1 番目のレコードを書き出す。
  ↓
W R I T E        n + 2 番目のレコードを書き出す。
```

:

ただし、拡張モード (EXTEND) で OPEN されている場合は、OPEN 文でファイルの最後のレコードに位置づけられ、その後にレコードが書き出される。また、WRITE 文では、さらに次のようなことが行われる。

WRITE 文実行中にファイルの終わり (注 1) が検出された場合、FILE STATUS 句の指定があると入出力状態値が設定される項目には”34”が格納される。また、そのファイルに関する USE 文があるとその USE 手続きが実行される。WRITE 文が失敗した場合は 01 レコードにデータを移送することもできなくなる。

注意 ファイルの終わりは、あらかじめアロケートされている領域を超えた時に検出される。

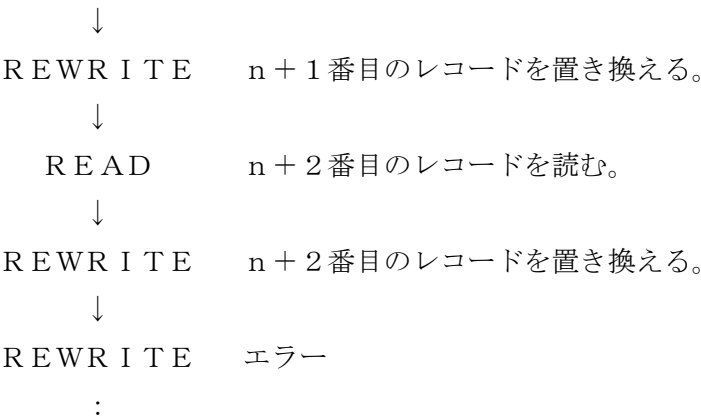
表 4-2 順ファイルの OPEN 文のモードと使用可能な文

OPEN 文 のモード	文			用途
	READ	WRITE	REWRITE	
入力モード (INPUT)	○	×	—	ファイルの最初からレコードを読み込む。
出力モード (OUTPUT)	×	○	—	ファイルの最初からレコードを書き出す。
入出力両用モード (I-O)	○	×	○	ファイル中のレコードを読んだり、また読み込んだレコードを置き換える。書き換えは、READ→REWRITE の順で読み込んだレコードを置き換える。
拡張モード (EXTEND)	×	○	×	ファイルの最後にレコードを追加する。

(e) REWRITE 文

REWRITE 文は、入出力両用モード (I-O) で OPEN されたファイルだけに使用可能である。REWRITE 文は、直前に READ 文で読み込まれたレコードを新しいレコードと置き換える。すなわち、REWRITE 文を実行する前には、READ 文が実行されていなければならない。また REWRITE 文で置き換えるレコードの長さは読み込んだレコードの長さと同じでなければならない。

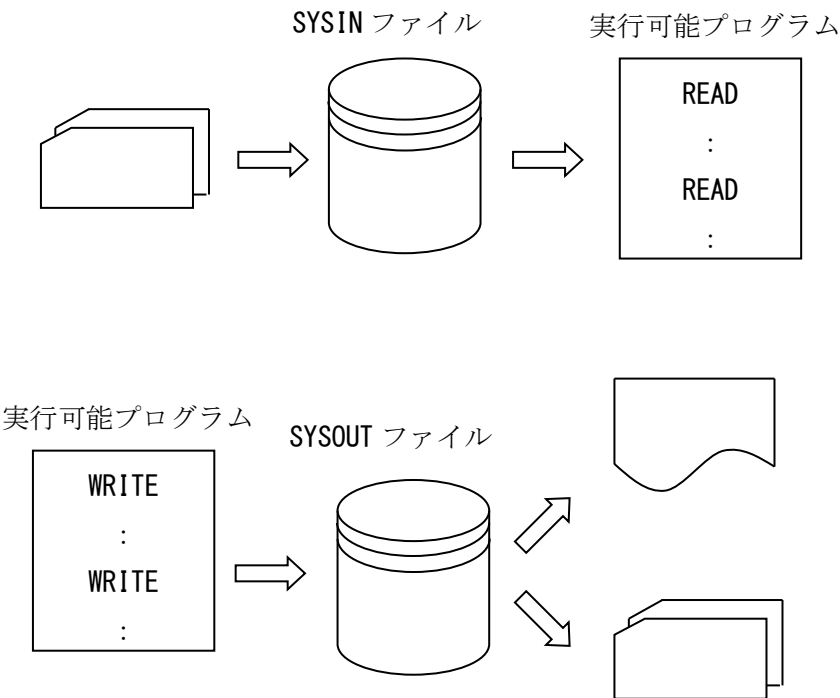
READ n 番目のレコードを読む。
↓
READ n + 1 番目のレコードを読む。



4.9 SYSIN/SYSOUT ファイル

高度の管理能力をもつ当オペレーティングシステムの下では, システム資源を有効に活用し, 最高のスループットを実現するために, 大記憶装置上の **SYSIN**ファイルおよび **SYSOUT**ファイルを使用する方がよい。

図 4-6 SYSIN/SYSOUT ファイルの流れ



4.9.1 SYSIN ファイル

COBOL では、SYSIN ファイルからデータを入力することができる。ここでは、SYSIN ファイルを使用する場合のプログラム上の定義等について述べる。

4.9.1.1 レコード形式

SYSIN ファイルは、改行コードで区切られたデータを持つファイルとして、オペレーティングシステム上のファイルシステム上に生成されたファイルである。

4.9.1.2 SYSIN ファイルの定義

SYSIN ファイルからデータを入力する場合、ファイル管理記述項で次のように記述しなければならない。

```
SELECT ファイル名 1  
    ASSIGN TO 内部ファイル名-SYSIN ...  
    [FILE STATUS IS データ名 1 [, データ名 2]]
```

説明

- ① SYSIN ファイルは順ファイルに属するため、ORGANIZATION 句は省略するか、または ORGANIZATION IS SEQUENTIAL と記述する。
- ② FILE STATUS 句で指定したデータ項目には、OPEN、CLOSE および READ 文が実行されたときの状態が格納される。
- ③ 上記以外に、次の句も指定できるがメモとして扱われる。
 - RESERVE 句…ファイルに割り当てられるバッファの個数を指定する。

4.9.1.3 LABEL RECORD 句

SYSIN ファイルの場合、LABEL RECORD 句は意味がない。すなわち、LABEL RECORD 句では OMITTED, STANDARD どちらを指定してもよい。

4.9.1.4 SYSIN ファイルの入出力文

SYSIN ファイルに対する入出力文には、次のようなものがある。

- OPEN 文
- CLOSE 文
- READ 文
- USE 文(「4.5.5 入出力誤り手続き」を参照)

FILE STATUS 句を指定することによって、OPEN、CLOSE および READ 文の実行状態が FILE STATUS 句のデータ名に格納されるため、FILE STATUS 句のデータ名を調べることによって実行状態を知ることができる。

入出力エラーに関しては表 4-1 を参照。

表 4-2 に、SYSIN ファイルの OPEN、CLOSE、READ および USE 文の動作を示す。

表 4-2 SYSIN ファイルの文の動作

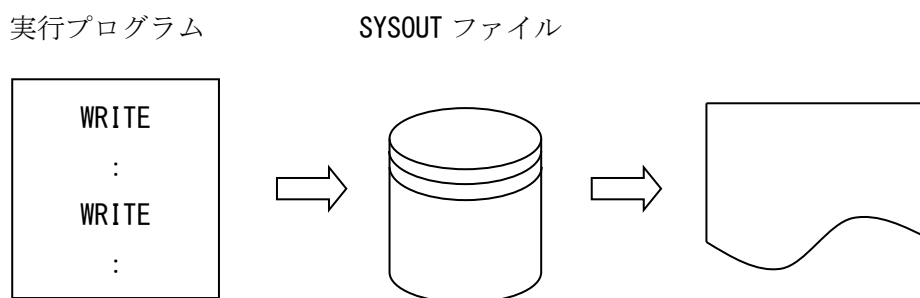
文	動 作
OPEN	• OPEN 文のモードは入力 (INPUT) だけが許される。 • SYSIN ファイルを使用できるようにする。
CLOSE	• 標準のファイル閉じを実行する。
READ	• SYSIN ファイルからデータの読み込みを行う。 • READ 文実行中データの終わりが見つかると AT END 条件になる。
USE	• OPEN、CLOSE および READ 文で入出力エラーが発生したり、入出力文の実行順序を間違ったりしたときに実行される。

4.9.1.5 SYSIN ファイル使用上の注意

- (1) SYSIN ファイル中のデータの大きさがファイルのレコード長より大きい場合、レコードの大きさを超えるデータは切り捨てられる。

4.9.2 SYSOUT ファイル

プログラムで印刷する場合、直接低速なプリンタを制御することは少なくなっている。本 COBOL では、大記憶装置上のファイルに出力し、システムが提供する制御プログラムにて、別途印刷する機能を提供する。このファイルを **SYSOUT** ファイルと呼び、次から説明する。



4.9.2.1 レコード形式

SYSOUT ファイルは、オペレーティングシステムのファイルシステム上に印刷データと制御データを持つファイルとして生成される。制御データの詳細は、「4.8.2.2 制御データ」を参照。

レコードを記述するとき、制御データを意識する必要はない。COBOL 実行時ルーチンが自動的に付加を行う。

記述方法については、「4.6 レコード形式」を参照。

環境変数 `COB_SYSOUT_FORMAT` により、**SYSOUT** ファイルのレコード形式を変更することができる。指定値については、「付録 D 環境変数」を参照。

備考 環境変数 `COB_SYSOUT_FORMAT` が `SSF` 指定のときのみ、`-CA` オプションを指定して、`APPLY SHIFT-CODE` 句で `WITH PPR-CONTROL-2` 指定ありの **SYSOUT** ファイルの印刷データに、プリンタ制御コードを埋め込む機能を使用できる。

`PRN` 指定または `TXT` 指定のとき、`APPLY SHIFT-CODE` 句で `WITH PPR-CONTROL-2` 指定ありの **SYSOUT** ファイルに対して、`CHARACTER TYPE` 句および `COLUMN NUMBER` 句を指定しても無効である。

【Enterprise】環境変数 `COB_SYSOUT_FORMAT` が `SSF` 指定(既定値)のときのみ、`-CP` オプションを指定して、`APPLY SHIFT-CODE` 句で `WITH PPR-CONTROL-3` 指定ありの **SYSOUT** ファイルの印刷データに、`KLP` 用プリンタ制御コードを埋め込む機能を使用できる。

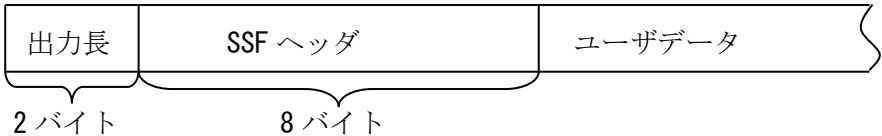
`PRN` 指定または `TXT` 指定のとき、`APPLY SHIFT-CODE` 句で `WITH PPR-CONTROL-3` 指定ありの **SYSOUT** ファイルに対して、`KLP` 用プリンタ制御コードを埋め込む機能は使用できない。

【Enterprise for Windows】環境変数 COB_SYSOUT_FORMAT が SSF 指定 (既定値) のときのみ, -Fy オプションを指定して, APPLY SHIFT-CODE 句で WITH PPR-CONTROL-1 指定ありの SYSOUT ファイルの印刷データに, シフトコードおよびプリンタ制御コードを埋め込む機能可以使用できる。

PRN 指定または TXT 指定のとき, APPLY SHIFT-CODE 句で WITH PPR-CONTROL-1 指定ありの SYSOUT ファイルに対して, シフトコードおよびプリンタ制御コードを埋め込む機能は使用できない。

4.9.2.2 制御データ

SYSOUT ファイルに対する出力を行った場合, 出力するレコードの前に 10 バイトの制御データが付加されて出力される。制御データは, 2 バイトのレコード長と 8 バイトの制御用レコードヘッダ (SSF ヘッダ) からなり, 2 バイトのレコード長には, 制御データ長 (10 バイト) と出力するデータ長の合計バイト数が格納され, SSF には改行制御情報等が格納される。



4.9.2.3 SSF ヘッダ

SSF ヘッダは 8 バイトからなり, 先頭 6 バイトには固定的に以降に示すデータが格納される。また, 7 バイト目と 8 バイト目には改行制御情報が格納され, WRITE 文における ADVANCING 句の書き方によって格納される値が異なる。改行制御情報について, 表 4-3 を参照。

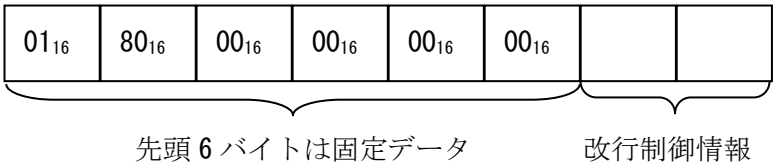


表 4-3 改行制御情報

ADVANCING	改行制御文字		機能
	第 7 バイト目	第 8 バイト目	
PAGE	E3 ₁₆	40 ₁₆	改ページする。
一意名	D9 ₁₆	一意名の内容	一意名の内容だけ相対改行する。

定数	D9 ₁₆	定数	定数だけ相対改行する。
呼び名	C1 ₁₆	LN 番号 - 1 (00 ₁₆ ～FE ₁₆)	LN 番号に絶対行改行する。

LN 番号は、機能名 LN001～LN255 の 001～255 の値を表す。

【Enterprise for Windows】 呼び名指定は、-Fv オプションを指定した場合のみ有効となる。

4.9.2.4 SYSOUT ファイルの定義

SYSOUT ファイルを定義するときのファイル管理記述項は、次のように記述しなければならない。

SELECT ファイル名 1

ASSIGN TO 内部ファイル名-SYSOUT …

[FILE STATUS IS データ名 1 [,データ名 2].]

説明

- ① SYSOUT ファイルは順ファイルに属するため、**ORGANIZATION** 句は省略するか、または **ORGANIZATION IS SEQUENTIAL** と記述する。
- ② **FILE STATUS** 句で指定したデータ項目には、**OPEN**、**CLOSE** および **WRITE** 文が実行されたときの状態が格納される。
- ③ 上記以外に、次の句も指定できるがメモとして扱われる。
 - **RESERVE** 句…ファイルに割り当てられるバッファの個数を指定する。
- ④ 装置名 **SYSOUT** はシステムファイルを示し、装置名 **PRU** または **PRINTER** はプリンタ装置を意味するが、本コンパイラでは同義とし、**SYSOUT** ファイルと総称する。

4.9.2.5 LABEL RECORD 句

SYSOUT ファイルの場合、**LABEL RECORD** 句は意味がない。すなわち、**LABEL RECORD** 句では **OMITTED**、**STANDARD** どちらを指定してもよい。

4.9.2.6 SYSOUT ファイルの入出力文

SYSOUT ファイルに対する入出力文には、次のようなものがある。

- OPEN 文
- CLOSE 文
- WRITE 文
- USE 文(「4. 5. 5 入出力誤り手続き」を参照)

FILE STATUS 句を指定することによって、OPEN、CLOSE および WRITE 文の実行状態が FILE STATUS 句のデータ名に格納されるため、FILE STATUS 句のデータ名を調べることによって実行状態を知ることができる。

入出力エラーに関しては表 4-1 を参照。

表 4-4 に、SYSOUT ファイルの OPEN、CLOSE、WRITE および USE 文の動作を示す。

表 4-4 SYSOUT ファイルの文の動作

文	動 作
OPEN	• OPEN 文のモードは出力 (OUTPUT) および拡張 (EXTEND) が許される。 • SYSOUT ファイルを使用できるようにする。
CLOSE	• 標準のファイル閉じを実行する。
WRITE	• SYSOUT ファイルに対して次の動作ができる。 1. 改行する前にレコードを印字する。 (BEFORE 句指定) 2. 改行した後レコードを印字する。 (AFTER 句指定) 3. PAGE 句の指定によって改ページさせる。そのとき、BEFORE 句があると、改ページする前にレコードを印刷し、AFTER 句の指定があると、改ページした後レコードを印刷する。 4. END-OF-PAGE 句の指定がある場合、印字位置が脚書き領域にきたときに、WRITE 文実行後、END-OF-PAGE 句の無条件文を実行させる (ページの終わり条件)。 5. LINAGE 句の指定されている場合、印刷ページ本体を超えたときは、自動的に改ページされる。このとき、BEFORE 句および AFTER 句の指定によってレコードが改ページ前に印刷されるか、改ページ後印刷されるかが決まる。END-OF-PAGE 句があると、その後無条件文が実行される (ページあふれ条件)。

USE	・ OPEN, CLOSE および WRITE 文で入出力エラーが発生したり, 入出力文の実行順序を間違ったりしたときに実行される。

4.9.2.7 SYSOUT ファイル使用上の注意

- (1) LINAGE 句付きの SYSOUT ファイルに関しては, 「4. 8. 3 LINAGE 句付きのファイル」を参照。
- (2) WRITE ADVANCING n LINES で n として用紙の形式で定義した印刷可能行数より大きい値を指定することはできない。
- (3) -Cj オプション指定がある場合, SYSOUT ファイルのレコード名または FROM 句の一意名に従属する基本項目に字類が日本語の項目があれば, プログラムの固有文字集合からシフト JIS へコード変換が行われる。

字類が日本語の項目以外であれば, SYSOUT ファイルの外部記憶媒体上の文字集合とプログラムの固有文字集合が同じとみなし, コード変換は行われない。

備考 英数字項目にシフトコード (KI/KO) で囲まれた日本語文字列があっても, プログラムの固有文字集合からシフト JIS へコード変換は行われない。

4.9.3 LINAGE 句付きのファイル

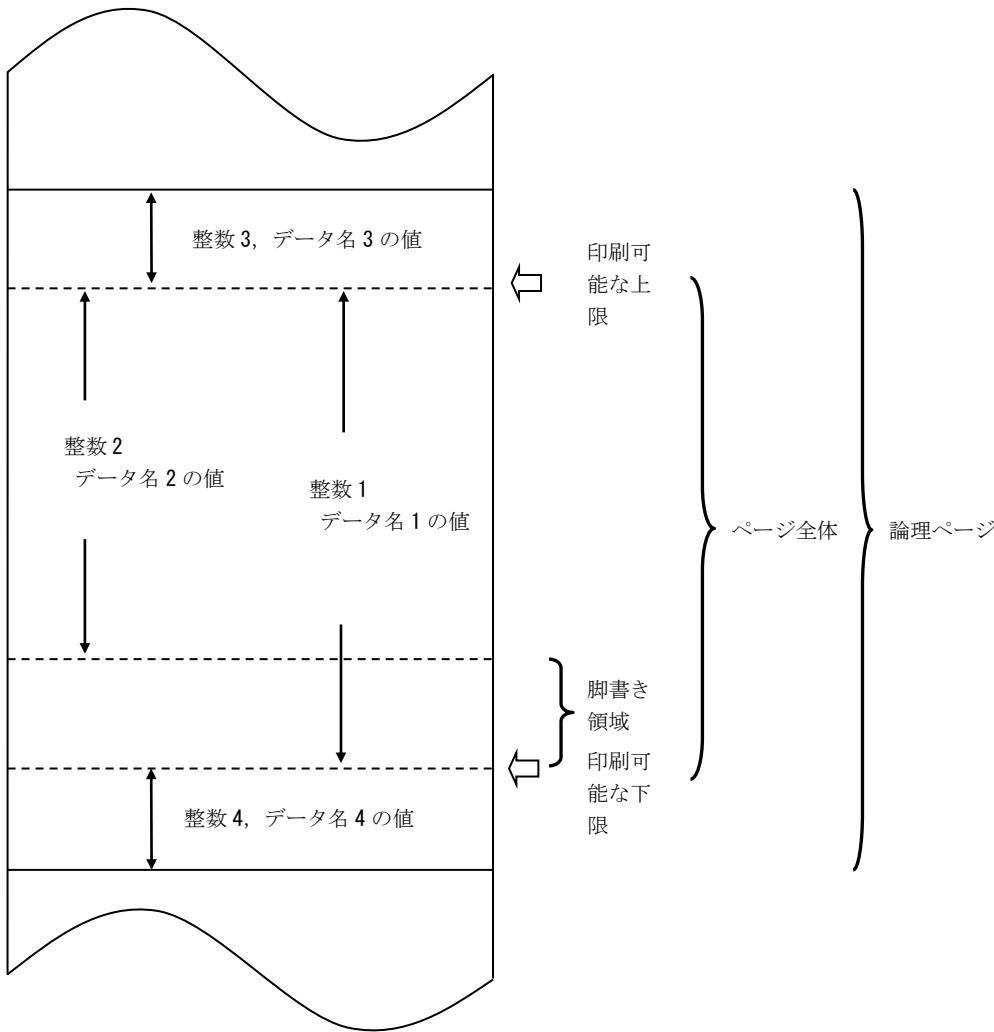
SYSOUT ファイルを使用する場合, ファイル管理記述項内で LINAGE 句を指定することによって, 論理ページを定義することができる (以降, LINAGE 句が指定されたファイルのことを LINAGE ファイルと呼ぶ)。

(1) LINAGE 句の記述形式

LINAGE IS { データ名 1
整数 1 } **LINES**
[, **WITH FOOTING AT** { データ名 2
整数 2 }]
[, **LINES AT TOP** { データ名 3
整数 3 }]
[, **LINES AT BOTTOM** { データ名 4
整数 4 }]

論理ページの形式は、上記の **LINAGE** 句指定によって図 4-7 のようになる。**LINAGE** 句の詳細は、『COBOL 言語説明書』を参照。

図 4-7 論理ページの形式



(2) LINAGE ファイルの改行制御

LINAGE ファイルの論理ページの形式は、SYSOUT ファイルを初期設定する際に与えた用紙の形式とは独立なものである。LINAGE ファイルはハードウェアとは別に独自の行数カウントを行っており、ハードウェアとは独立にページの制御を行っている。たとえば、PAGE 句付きの WRITE 文を実行した場合、ハードウェアに対し改ページ要求を出すのではなく、独自に持っている行カウンタ (LINAGE-COUNTER) から次ページまでの行数を計算し、それだけの改行をハードウェアに要求する。

しかし、ハードウェアを初期設定した際の用紙の形式は無視できるものではなく、もし初期設定した際の用紙の形式に印刷不能領域があった場合には、ハードウェアはハードウェアが認識しているページの切れ目の前後で改行を行ってしまい、LINAGE ファイルは正しい印刷ができなくなってしまう。

またハードウェアは用紙がなくなった場合、ハードウェアが認識しているミシン目の位置で印刷を中断するが、これが必ずしも LINAGE ファイルの論理ページの終わりとは一致しない。

(3) LINAGE ファイル使用上の注意

LINAGE ファイルを使用する場合には、次の点に注意しなければならない。

- (a) 帳票定義の論理ページの大きさと、LINAGE ファイルの論理ページの大きさを一致させる。
- (b) 同一内部ファイル名を他のプログラムでも使用する場合はファイルを閉じ、呼ばれたプログラムで開かなければならない。すなわち、プログラム間で LINAGE-COUNTER を共用できない。
- (c) OPEN 文が実行されると改ページ動作が行われる。
- (d) 内部ファイルが SPR の場合、SYSOUT に対する DISPLAY 文を使用すると、LINAGE-COUNTER やページ制御が乱れる。
- (e) 論理ページサイズ (1 ページの行数) は、OPEN 文および WRITE 文の実行時に、LINAGE 句の指定値により決定する。『COBOL 言語説明書』「9. 14. 24. LINAGE 句」を参照。

【Standard】 -Qg オプションを指定することにより、OPEN 文の実行時にのみ、論理ページサイズを決定する。

4.10 小入出力技法

ACCEPT 文および DISPLAY 文では、形式の不定な小容量のデータを入出力する。さらに、ACCEPT 文においては、現在の時刻または日付を入力することができる。

4.10.1 ACCEPT 文

ここでは、ACCEPT 文について記述する。

【Enterprise for Windows】【Standard】改行文字:16 進 0DOA

【Enterprise for Linux】改行文字:16 進 0A

(1) 入力装置

ACCEPT 文によってデータが読み込まれる装置には、次のものがある。

(a) SYSIN ファイル

次のような記述があると、SYSIN ファイルから入力される。

- i. ACCEPT 一意名.
- ii. ACCEPT 一意名 FROM SYSIN.
- iii. SPECIAL-NAMES. SYSIN IS 呼び名.

ACCEPT 一意名 FROM 呼び名.



【Enterprise】ただし、i. は、小入出力先の既定値を ACOS-4 互換とする -Fj オプションを指定した場合。

SYSIN ファイルは、環境変数 COB_SYSIN に次の形式で指定する。

【Enterprise for Windows】【Standard】

set COB_SYSIN=入力ファイル名

【Enterprise for Linux】

export COB_SYSIN=入力ファイル名

環境変数 COB_SYSIN の指定がない場合の既定の入力ファイル名は次の形式となる。

- ・ プログラム定義の場合:PROGRAM-ID のプログラム名 + “.in”
- ・ 利用者定義関数の場合:FUNCTION-ID の利用者定義関数名 + “.in”

- ・ クラス定義の場合: **CLASS-ID** のクラス名 + “.in”

(2) データの読み込み方

(a) **SYSIN** ファイルからの入力

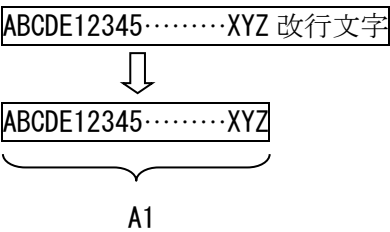
SYSIN ファイルの外部記憶媒体上の文字集合は、プログラムの固有文字集合と同じとみなす。つまり、**-CU** オプションの指定がない場合は、シフト **JIS** であり、**-CU** オプションの指定がある場合は、**UTF-8** である。したがって、**SYSIN** ファイルからの入力において、コード変換は行われない。

【Standard】**-CU** オプションは未サポート。

【Standard】【Enterprise for Windows】本システムでは、**-CF** オプションの指定がある場合、**JIS** となる。

- i. 受け取り側データ項目の桁数と、入力データの桁数が同じ場合は次のようになる。

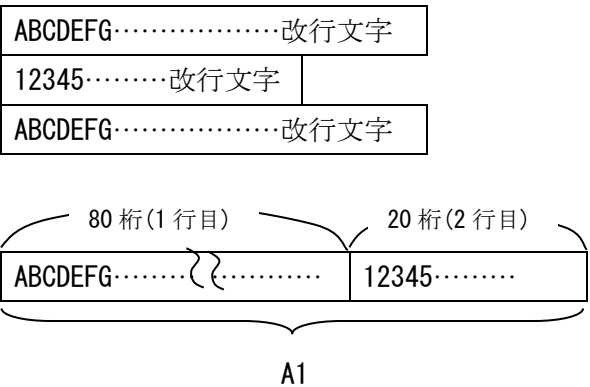
```
01 A1 PIC X(80).  
ACCEPT A1.
```



この後 **ACCEPT** 文が実行されると次の行が読まれる。

- ii. 受け取り側データ項目の桁数が入力データの桁数より大きい場合

```
01 A1 PIC X(100).  
ACCEPT A1.
```



受け取り側データ項目がいっぱいになるまで、80 桁単位で読み込まれ左側から格納される。そして、読み込んだ入力データのうち、受け取り側データ項目に格納できなかった文字は無視される。最後のデータが読み込まれても、受け取り側項目がいっぱいでない場合は空白で埋められる。

【Standard】 -Qi オプションを指定することにより、改行文字を入力したとき終了する。
受け取り側データ項目の残りの部分は空白で埋める。

iii. 受け取り側データ項目が入力データの桁数より小さい場合

01 PIC X(5).

ACCEPT A1.

格納できなかった残りの文字は無視される(左詰め)。

ABCDEFGH 改行文字



ABCDE



A1

iv. データをすべて読み終わった後さらに ACCEPT 文を実行した場合、異常終了する。

(3) 入力データの桁数

ACCEPT 文で読み込むことのできるデータの桁数に制限はなく、受け取り側データ項目の桁数で決まる。ただし、SYSIN ファイルの場合には、80 桁×n の単位で読み込まれる。

(4) 日付と時間の入力

現在の日付と時間は、オペレーティングシステムによって一貫して管理されており、実行用プログラムでそれを参照することができる。

(a) DATE(年月日)の入力

ACCEPT 一意名 FROM DATE.

と記述することによって、yyymmdd 形式の日付が一意名の領域に転記される。

1987 年 9 月 1 日 $\longrightarrow$ 870901

(b) DAY(年日)の入力

ACCEPT 一意名 FROM DAY.

と記述することによって、yyddd 形式の日付が一意名の領域に転記される。

1987 年 9 月 1 日 $\longrightarrow$ 87244

(c) TIME(時刻)の入力

ACCEPT 一意名 FROM TIME

の記述によって、hhmmssstt 形式で、現在の時刻を一意名の領域に転記する。ここで、hh は時間、mm は分、ss は秒、tt は 1/100 秒を表す。

午後 2 時 41 分 35 秒 06 $\Longrightarrow$ 14413506

(5) 入力操作

(a) SYSIN ファイルからの入力

ACCEPT 文による SYSIN ファイルからの入力操作は、READ 文による SYSIN ファイルの入力と同じである。

4.10.2 DISPLAY 文

ここでは、DISPLAY 文について述べてある。

【Enterprise for Windows】【Standard】改行文字:16 進 0D0A

【Enterprise for Linux】改行文字:16 進 0A

(1) 出力装置

DISPLAY 文によってデータが出力される装置には次のものがある。

(a) SYSOUT ファイル

次のような記述があると、SYSOUT ファイルに出力される。

- i. **DISPLAY** $\left\{ \begin{array}{l} \text{一意名 1} \\ \text{定数 1} \end{array} \right\} \dots$
- ii. **DISPLAY** $\left\{ \begin{array}{l} \text{一意名 1} \\ \text{定数 1} \end{array} \right\} \dots \text{UPON SYSOUT}$
- iii. **SPECIAL-NAMES. SYSOUT IS 呼び名 1**
 DISPLAY $\left\{ \begin{array}{l} \text{一意名 1} \\ \text{定数 1} \end{array} \right\} \dots \text{[UPON 呼び名 1]}$

【Enterprise】ただし、i. は、小入出力先の既定値を ACOS-4 互換とする -Fj オプションを指定した場合。

SYSOUT ファイルは、環境変数 COB_SYSOUT に次の形式で指定する。

【Enterprise for Windows】【Standard】

set COB_SYSOUT=出力ファイル名

【Enterprise for Linux】

export COB_SYSOUT=出力ファイル名

環境変数 COB_SYSOUT の指定がない場合、出力先は実行可能ファイルを実行したディレクトリ配下となり、出力ファイル名は次の形式となる。

- ・ プログラム定義の場合:PROGRAM-ID のプログラム名 + “_” + [pid] + “.out”
- ・ 利用者定義関数の場合:FUNCTION-ID の利用者定義関数名 + “_” + [pid] +

“.out”

- ・ クラス定義の場合:CLASS-ID のクラス名 + “_” + [pid] + “.out”

ここで, [pid]は, プロセス ID を表す。

(b) データの出力形式

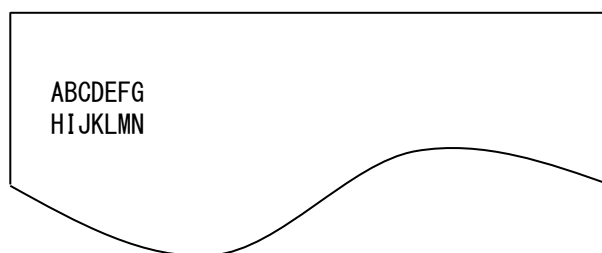
DISPLAY 文で出力できるデータの桁数および出力形式は出力装置によって異なり, 次のようになる。

- ・ SYSOUT ファイルへ出力

最大印刷文字数には制限はない。1 行に収まらない場合は複数行に分割されて出力される。1 行に印刷する文字数は 132 文字である。印刷は送り出し側データ項目の左から出力される。改行数は印刷後に 1 行改行である。

DISPLAY “ABCDEFGG” .

DISPLAY “HIJKLMN” .



SYSOUT ファイルの外部記憶媒体上の文字集合は, プログラムの固有文字集合と同じとみなす。

【Enterprise】-CU オプション指定がある場合は, 環境変数 COB_SYSOUT_CODE により, 実行時に外部記憶媒体上の文字集合を指定することができる。このとき, プログラムの固有文字集合から外部記憶媒体上の文字集合へ一意名/定数の属性に従ってコード変換が行われる。ただし, SYSOUT のレコード形式が SSF の場合は, COB_SYSOUT_CODE の指定にかかわらず, 外部記憶媒体上の文字集合はシフト JIS として扱う。

環境変数 COB_SYSOUT_CODE は, アプリケーションの実行前に設定しておかなければならない。設定可能な値については, 「付録 D 環境変数」を参照。

【Standard】【Enterprise for Windows】-CF オプション指定がある場合は, 出力データ全体に対して, プログラムの固有文字集合からシフト JIS へコード変換が行われる。

-Cj オプション指定がある場合は, 出力データ全体に対して, プログラムの固有文字

集合からシフト JIS ヘコード変換が行われる。

備考 一意名または定数の字類が日本語以外でシフトコード (KI/KO) で囲まれた日本語文字列があるか、字類が日本語であれば、プログラムの固有文字集合からシフト JIS ヘコード変換が行われる。

なお、最後の一意名および定数の後には、改行文字を出力する。

4.11 その他の補足事項

ここまでは今まで述べた入出力技法に対する補足説明を行う。

4.11.1 EXTERNAL 句付きのファイル

SELECT 句または FD 句に EXTERNAL 句を書くことができる。EXTERNAL 句付きのファイルは他のプログラムとファイルを共用することを意味する。すなわち、複数のプログラムをリンクし、あるプログラムで OPEN し、他のプログラムで READ 文、WRITE 文を実行するということができる。ただし、次の点に注意しなければならない。

- 装置名 SYSIN または SYSOUT のファイル管理記述項の ASSIGN 句に定数指定またはデータ名指定をした場合は共用されない。
- 【Enterprise】識別名 RAF のファイル管理記述項の ASSIGN 句に定数指定またはデータ名指定をした場合は共用されない。
- SELECT 句、FD 句の記述は次のものを除いてすべて同一にしなければならない。
 - FILE STATUS 句の有無
 - DATA RECORD 句
 - ファイル名、データ名、レコード名

4.11.2 BASED 句付きのファイル

BASED 句付きのファイルは SELECT 句に BASED 句を書くことによって定義される。

BASED 句付きのファイルは、複数のプログラム間で同一内部ファイル名を共有し、かつその内部ファイル名が翻訳時に決まらず実行時に他のファイルから、動的に設定されることを示す。BASED 句付きのファイルはファイルの制御構造の実体はそのプログラムにはなく、活性化する側の実行時要素から制御構造へのポインタがパラメータ等で渡される。

このファイルは、複数のプログラムで共用されることを前提とするので、EXTERNAL 句付きのファイルと同じ扱いとなる。すなわち、複数のプログラムをリンクし、あるプログラムで OPEN し、他のプログラムで READ 文、WRITE 文等を実行するということができる。ただし、次の点に注意しなければならない。

- このプログラムの実行時に他の **SELECT** 句に **EXTERNAL** 句付きのファイルまたは **BASED** 句付きのファイルのアドレスを **SET** 文によりデータ名 (データポインタデータ項目) に設定してから使用する。
- ファイルの属性は次のものを除いてその実体を有するファイルの属性と同じでなければならない。
 - － **FILE STATUS** 句の有無
 - － **DATA RECORD** 句
 - － ファイルに関する記述において、利用者の定義する語
- **BASED** 句付きのファイルが、その実体を有するファイルと共有するのは、ファイル制御ブロックだけであり、レコード領域、入出力状態値が設定される領域、行数カウンタ等は個々のプログラムに存在する。
- 整列用ファイルを指定することはできない。
- **BASED** 句付きのファイルに対し、**OPEN** 文および **CLOSE** 文を実行できない。
- 行順ファイル (CSV 形式) を指定することはできない。

(1) **BASED** 句付きのファイルの定義

BASED 句付きのファイルのファイル管理記述項の記述は、次のようになる。

```
SELECT BASED ファイル名 1 ASSIGN TO
      データ名 1 {‐ 装置名}
```

説明

- ファイル名 1 は整列用ファイル記述項の名前として使用してはならない。
- データ名 1 はプログラム内でデータポインタデータ項目として定義されていなければならない。

(2) ファイルのアドレスをデータ名に設定する方法

SELECT 句に **EXTERNAL** 句付きのファイルまたは **BASED** 句付きのファイルのアドレスを活性化される側の実行時要素のデータ名に設定するには次の順序で行う。

- (a) 活性化する側の実行時要素中の **SET** 文で、ポインタ名にファイルのアドレスを設定する。
- (b) **CALL** 文のパラメータ等によりポインタ名を活性化される側の実行時要素に渡す。
- (c) 活性化される側の実行時要素中でポインタ名とデータ名を関連づける。

SET 文の記述は、次のようになる。

```
SET データポインタデータ項目 1 TO ADDRESS OF ファイル名 1
```

(3) COBOL プログラム間での使い方

COBOL プログラム間での **BASED** 句付きファイルの使い方の例を次に示す。

(a) 連絡節を使用してファイルのアドレスを渡す例

MAINPROG のプログラム

```
SELECT EXTERNAL F1 ASSIGN TO ファイル識別リテラル-SYSIN
      :
WORKING-STORAGE SECTION.
01 FILEPTR USAGE POINTER.
      :
OPEN INPUT F1.
SET FILEPTR TO ADDRESS OF F1.
CALL "SUBPROG" USING FILEPTR.
      :
```

SUBPROG のプログラム

```
SELECT BASED F2 ASSIGN TO FILEPTR-SYSIN.
      :
LINKAGE SECTION.
01 FILEPTR USAGE POINTER.
      :
PROCEDURE DIVISION USING FILEPTR.
      :
READ F2
      AT END ~
      :
```

4.11.3 READ INTO 文

READ 文に INTO 句を設定した場合の入力領域から一意名のデータ領域への転記方法は次のとおりである。

(1) 固定長レコード形式の場合

プログラムで固定長レコードと定義されているファイルなら 01 レコードから一意名のデータ領域へ CORRESPONDING 句のない MOVE 文の規則に従って転記される。

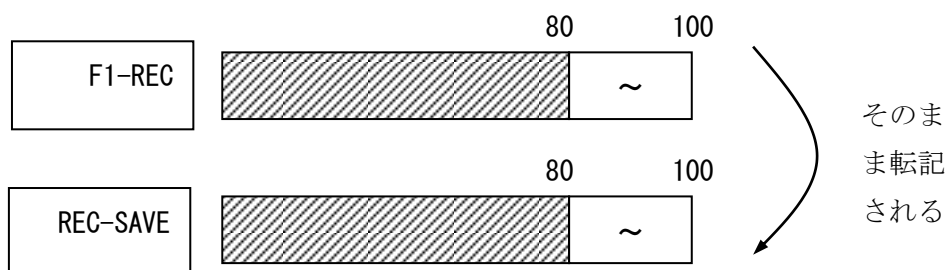
例

```
SELECT F1 ASSIGN TO F1-SYSIN.
FD F1 ~
01 F1-REC PIC X(100).
WORKING-STORAGE SECTION.
01 REC-SAVE PIC X(100).
:
```

```
READ F1 INTO REC-SAVE ~
```

説明

実際に読み込まれたレコード長を 80 とすると上記の結果, F1-REC, REC-SAVE は次のようになる。



備考 斜線部分は実際に読み込まれたレコードである。

(2) 可変長レコード形式の場合

プログラムで可変長レコード形式と定義されているファイルは、現在のレコードが 01 レコードから一意名のデータ領域へ、送り出し側の長さを実際に読み込んだレコード長で集団項目転記される。

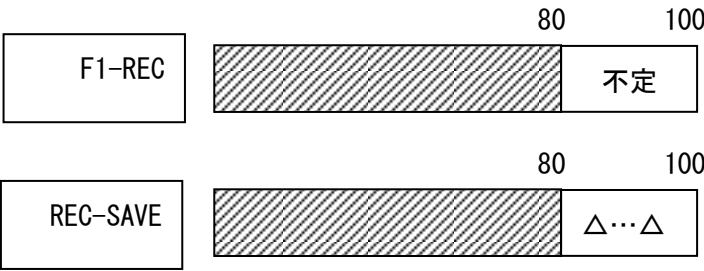
例

```
SELECT F1 ASSIGN TO F1-SYSIN.  
FD F1 ~  
RECORD VARYING ~  
01 F1-REC PIC X(100).  
WORKING-STORAGE SECTION.  
01 REC-SAVE PIC X(100).  
:
```

```
READ F1 INTO REC-SAVE ~
```

説明

実際に読み込まれたレコード長を 80 とすると上記の結果、F1-REC、REC-SAVE は次のようになる。



備考 斜線部分は実際に読み込まれたレコードである。

第5章 中核に関する補足説明

本章では、中核に属する句や文のうち特に重要と思われるものや特にわかりにくいと思われるものについて、言語説明に対する補足説明を行う。したがって、それぞれの句や文に対する詳細な規則については、『COBOL 言語説明書』を参照。

5.1 言語の概念

5.1.1 英大文字と英小文字の使い方

COBOL プログラムの語は、英大文字と英小文字どちらで記述してもかまわない。次に示す場所以外で、英小文字は対応する英大文字と等価に扱われる。

英小文字が英大文字と不等価となる場所

1. 注記行
2. 注記項
3. 英数字定数
4. プログラム名 (PROGRAM-ID 段落で指定される) (コンパイルオプション指定により、等価とすることが可能)
5. インラインコメント

英小文字を使用したプログラム例を以下に示す。

```

.....
DATA DIVISION.
*
working-storage section.
77 D-Line      pic 9.
77 Denpyo-no   pic 9(5) usage comp-3 value zeros.
01 IN-data.
   02 IN-YY    pic 99.
   02 IN-MM    pic 99.
   02 IN-DD    pic 99.
.....
PROCEDURE DIVISION.
*****

```

```

proc-0.
.....
    move 1 to D-Line.
    move zero to Gokei.
proc-4.
    if D-Line = 1  display Scr-11
       else       display Scr-12.
.....
    
```

これは英大文字のみを使用した以下のプログラムと等価である。

```

.....
DATA DIVISION.
*
WORKING-STORAGE SECTION.
77 D-LINE      PIC 9.
77 DENPYO-NO   PIC 9(5) USAGE COMP-3 VALUE ZEROS.
01 IN-DATA.
   02 IN-YY    PIC 99.
   02 IN-MM    PIC 99.
   02 IN-DD    PIC 99.
.....
PROCEDURE DIVISION.
*****
PROC-0.
.....
    MOVE 1 TO D-LINE.
    MOVE ZERO TO GOKEI.
PROC-4.
    IF D-LINE = 1  DISPLAY SCR-11
       ELSE       DISPLAY SCR-12.
.....
    
```

5.1.2 句読文字と分離符の使い方

COBOL プログラムの語はすべて分離符で区切られる。言いかえるならば COBOL プログラムでは、語と語の間には必ず分離符を置かなければならない。

分離符は 1 つまたは複数個の句読文字からなる。句読文字には次のものがある。

句読文字	,	コンマ
	;	セミコロ
	.	終止符
	"	引用符
	'	引用符(アポストロフィ)
	(	左かっこ
	)	右かっこ
	=	等号
	:	コロ
	△	空白

これらの文字が句読文字とみなされるのは、次に示す場所以外で使用されている場合だけである。

- (1) 注記行……………注記行内の文字はすべて注記とみなされる。
- (2) 注記項……………見出し部の段落で利用できる注記項では、その中のすべての文字が注記とみなされる。
- (3) 英数字定数……………英数字定数中の文字はすべて定数の一部とみなされる。
- (4) PICTURE 句の文字列…PICTURE 句の文字列中の文字はすべて文字列の一部とみなされる。
- (5) 行内注記……………行内注記中の文字はすべて注記とみなされる。

分離符の書き方の規則は、次のとおりである。

- (1) 句読文字空白を分離符として用いる場合は、空白は 1 つでもそれ以上書いてもよい。

MOVE _{△△} A _△ TO _△ B	}	これらは等価である。
MOVE _{△△△△△} A _{△△△} TO _{△△△} B		

- (2) 句読文字のコンマ、セミコロンおよび終止符を分離符として用いる場合は、これらの後に空白を置かなければならない。ただし、これらの前の空白はあってもなくてもよい。

MOVE _{△△} A _△ TO _△ B. DISPLAY _△ B.	誤り
MOVE _{△△} A _△ TO _△ B. △DISPLAY _△ B.	正しい
ADD A, B, C _△ TO _△ D.	誤り
ADD A, △B, △C _△ TO _△ D.	正しい

- (3) 句読文字の左かっこおよび右かっこを分離符として用いる場合は、必ず左かっこと右かっこを対にして用いなければならない。これらは添字付け、部分参照、関数一意名、算術式および条件の中だけで用いることができる。左かっこおよび右かっこの前後には空白があってもなくてもよい。

A _△ (_△ 1 _△) △	}	これらは等価である。
A(_△ 1 _△)		
A(1)		
IF (A _△ = _△ B) AND (C _△ = _△ D)	}	これらは等価である。
IF _△ (_{△△} A _△ = _△ B _△) _△ AND _△ (_△ C _△ = _△ D _△)		

01 _△ A _△ PIC _△ 9(5)	正しい
01 _△ A _△ PIC _△ 9 _△ (5)	誤り

- (4) 句読文字の引用符を分離符として用いる場合、始めの引用符の前には空白か左かっこがなければならない。また、終わりの引用符の後には空白、コンマ、セミコロン、終止符または右かっこのいずれかがなければならない。引用符は、英数字定数を区切るために必ず対にして書かなければならない。ただし、英数字定数が継続している場合はこのかぎりではない。

MOVE _△ "A" _△ TO _△ C.	正しい
MOVE" A" _△ TO _△ C.	誤り
IF ("A" _△ = _△ C) GO _△ TO _△ X.	正しい
DISPLAY _△ "A", _△ X, _△ "D"	正しい

5.1.3 英数字定数と 16 進英数字定数

本 COBOL においては、プログラム中で英数字定数をとる方法として、定数としてとりたい文字を直接書く方法と、16 進で表す方法の 2 通りがある。本 COBOL では前者を英数字定数と呼び、後者を 16 進英数字定数と呼ぶ。16 進英数字定数は、キーボード上に存在しない文字をとりたい場合、およびプリンタの文字セットとして存在しない文字をとりたい場合に有効である。

(1) 英数字定数

英数字定数は引用符で囲んで指定する。本 COBOL では 256 桁の英数字定数まで指定することができる。英数字定数として引用符をとりたい場合は、2 個の引用符を続けて書く。

"ABC"	指定できない。
'ABC'	指定できない。
"AB""CD"	値 AB"CD がとられる。
"AB""""CD"	値 AB""CD がとられる。
'AB ' 'C'	値 AB 'C がとられる。

(2) 16 進英数字定数

16 進英数字定数は、英数字定数のなかでさらに引用符で囲んで、16 進数の列として定義する。

16 進英数字定数では、実際にとりたい文字の主記憶上のコードを 16 進数で指定する。16 進数は、次の図に従って指定する。

16 進数	対応するビット列	16 進数	対応するビット列
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

主記憶上の文字は 8 ビットで表される。したがって、これを 16 進数で表すには 2 桁で指定する。すなわち、上位 4 ビットを上位 1 桁の 16 進数で指定し、下位 4 ビットを下位 1 桁で指定する。

文字	主記憶中のビット列	16 進数
1	00110001	31
A	01000001	41
△	00100000	20

16 進英数字定数は、前述したように英数字定数内でさらに引用符で囲んで指定する。引用符は継続行がないかぎり、必ず対になっていなければならない。

16 進英数字定数	実際にとられる値
""31""	1
"A"31"B"	A1B
""414243""	ABC
"8""3"41""	8"3A
""""41""""	"A"

5.1.4 部分参照

ここでは、部分参照を使用しない場合と使用した場合のコーディング方法を対比して示すことにより、部分参照について説明する。

部分参照を使用しないと、以下の例 (1) ～ (3) のように、あるデータの一部分を参照する場合は、再定義するか、添字付けをするか、文字列操作でデータを取り出すかしなければならない。

部分参照を使用すると、データの一部分を、最左端文字位置および長さを指定することにより、以下のよう容易に参照できる。

例 (1) データ A の斜線部分が '*' であるかを検査する。(開始位置と長さがともに固定)

データ A	1	2	3	4	5	6	7	8	9	10	11	12	13	14

・部分参照を使用しない場合

```
01 A PIC X(14).
01 A-R REDEFINES A.
02 A-R1 PIC X(5).
02 A-R2 PIC X(7).
02 A-R3 PIC X(2).
```

```
IF A-R2 = ALL '*'
.....
```

- ・部分参照を使用した場合

```
01 A PIC X(14).
```

```
IF A(6:7) = ALL '*'
```

```
.....
```

例(2) データ A の斜線部分が '*' であるかを検査する。(開始位置が可変)

データA	1	2	3	4	5	6	7	8	9	10	11	12	13	14

- ・部分参照を使用しない場合

```
01 A PIC X(14).
```

```
01 B PIC X(14).
```

```
01 C PIC X(14) VALUE ALL '*'.
```

```
01 D PIC X(14).
```

```
01 I PIC 9(2).
```

```
UNSTRING A INTO B WITH POINTER I.
```

```
UNSTRING C INTO D WITH POINTER I.
```

```
IF B = D
```

```
.....
```

- ・部分参照を使用した場合

```
01 A PIC X(14).
```

```
01 I PIC 9(2).
```

```
IF A(I:) = ALL '*'
```

```
.....
```

例(3) データ A の斜線部分が '*' であるかを検査する。(開始位置, 長さともに可変)

データA	1	2	3	4	5	6	7	8	9	10	11	12	13	14

- ・部分参照を使用しない場合

```
01 A PIC X(14).
```

```
01 A-R REDEFINES A.
```

```
02 A-RI PIC X OCCURS 14 TIMES INDEXED BY IND1.
```

```
01 B.
```

```

02 B1    PIC X OCCURS 1 TO 14 TIMES
        DEPENDING ON J INDEXED BY IND2.
01 I     PIC 9(2).
01 J     PIC 9(2).

```

```

SET IND1 TO I.
SET IND2 TO 1.
PERFORM J TIMES
    MOVE A-R1 ( IND1 ) TO B1 ( IND2 )
    SET IND1 IND2 UP BY 1
END-PERFORM.
IF B = ALL '*'
    .....

```

・部分参照を使用した場合

```

01 A     PIC X(14).
01 I     PIC 9(2).
01 J     PIC 9(2).

```

```

IF A(I:J) = ALL '*'
    .....

```

5.2 データ部

ここでは、PICTURE 句、REDEFINES 句および RENAMES 句について説明する。

5.2.1 PICTURE 句

PICTURE 句は、基本項目の一般的な性質と編集の形式を指定する。また、基本項目の項類も指定する。

次に、PICTURE 句の正しい書き方と、編集結果についていくつかの例をあげる。

表 5-1 PICTURE 句と編集方法

PICTURE 句の形式	送り出し側の値	結果	備考
999.999	12	000.012	
999.999	1234	001.234	
99B99B99	123456	12^34^56	
99/99/99	123456	12/34/56	
999.999	12.34	012.340	
+9999	123	+0123	
-9999	123	^0123	+0123 ではない。
-9999	-123	-0123	
9999+	-123	0123-	
¥9999	123	¥0123	
¥99.999	123	¥00.123	
99.999CR	+123	00.123^^	
99.999CR	-123	00.123CR	
¥99.999DB	-123	¥00.123DB	
+++++	123	^+123	
+++++	1234	+1234	
+++++	12345	+2345	
¥¥.¥¥¥	123	^^¥123	¥.123 ではない。
¥¥.¥¥¥	1234	¥1.234	
¥¥.B¥¥¥	123	^^^¥123	¥.^123 ではない。
¥¥.B¥¥¥	1234	¥1.^234	
¥¥.¥¥¥CR	-1234	¥1.234CR	
-.---.---9	-123456	^-123.456	
¥¥.B¥B	123	¥2.^3^	
¥¥.¥¥¥	0	^^^^^^	
¥¥.¥¥9	0	^^^^¥0	
¥++.^++	.123	¥^+.123	
++.+++	.012	^+.012	.+12 ではない。
++.+++	0	^^^^^^	実際の小数点も^になる。
---.---.999	-123	^^^^^^^-123	-.123 ではない
---.---.999	1234	^^^^^^1.234	
ZZZ.ZZ9	1234	^^1.234	
+ZZZ.ZZ9	123	+^^^^123	
ZZZ.ZZ9	0	^^^^^^0	
ZZZ.ZZZ	0	^^^^^^	
ZZZ.ZZZ	.120	^^^.120	
ZZZ.ZZZ	.012	^^^.012	^.12 ではない。
ZZZ.ZZZ	0	^^^^^^	^^^ ^^ ではない。
.	1234	**1.234	
.	0	*****	***.*** ではない。
.	1.234	**1.234	
.	0	***.***	***** ではない。
BB99	1234	***12^34	
BB99	1	*****01	
BB99	123456	12^34^56	
ZZ9.999	0	^^0.000	
ZZZ.999	0	^^^.000	
ZZZ.999	.012	^^^.012	

5.2.2 REDEFINES 句

REDEFINES 句は、同一の記憶領域を異なるデータ項目で再定義して使用するために用いられる。データ項目を再定義することによって、次のようなことを行うことができる。

- 同一領域の再利用
- データの再定義
- 表の初期値の定義

(1) 同一領域の再利用

プログラム中で排他的に使用される作業領域は別々の領域として定義せず、REDEFINES 句を用いて同一領域として定義することができる。たとえば、更新処理と入力処理において作業領域が排他的に使用されるような場合、次のように記述する。

```
01 UPDATE-WORK-AREA.
```

} 更新処理用作業領域に関する記述

```
01 INPUT-WORK-AREA
```

```
    REDFINES UPDATE-WORK-AREA.
```

} 入力処理用作業領域に関する記述

各処理において作業領域の初期化が必要であれば、MOVE 文などを用いて各処理の最初で作業領域の初期化を行う必要がある。

(2) データの再定義

同一のデータを異なる形式で処理したい場合、REDEFINES 句は非常に有効である。たとえば、10 桁のデータを 2 桁、3 桁、5 桁の 3 つの別々のデータとして処理したい場合、ある項類のデータを異なる項類のデータとして処理したい場合に使用される。この例を次に示す。

```

01 INPUT-RECORD.
  02 CODE-DATA PIC X (10).
  02 CODE-DETAIL REDEFINES CODE-DATA.
    03 CODE1 PIC X(2).
    03 CODE2 PIC X(3).
    03 CODE3 PIC 9(5).
  02 INPUT-DATA PIC X(50).

```

再定義する側とされる側のレベル番号は同じでなければならず、**66** や **88** であってはならない。また、ファイル節の **01** レベルの記述項にこの句を書いてはならない。なぜなら、ファイル記述項 (**FD**) に従属するすべてのレコード領域は、自動的に再定義されるからである。

さらに留意すべき点としては、同じ領域を重複して再定義する場合である。このとき **REDEFINES** 句を含む記述項は、再定義される領域の記述項の直後に連続していなければならず、この間に新しい領域を定義する記述項があってはならない。また、すべての **REDEFINES** 句は領域を最初に定義した記述項のデータ名を使わなければならない。

例	[正]	[誤]
	01 A	01 A
	: .	:
	01 B REDEFINES A.	01 B REDEFINES A.
	:	:
	01 C REDEFINES A.	01 C REDEFINES B. [誤]
	:	:
	01 D REDEFINES A.	01 D
	:	:
	01 E	01 E REDEFINES A. [誤]
	:	:

(3) 表の初期値の定義

COBOL の規則では、**OCCURS** 句が書いてあるデータ記述項またはそれに属するデータ記述項に **VALUE** 句を書くと、関連する各要素に同じ値が設定される。各要素の値が異なる表の初期値をとるには、あらかじめ初期値をとったデータ項目を再定義して使うか、集団項目レベルで **VALUE** 句を指定するか、いずれかの方法をとらなければならない。「第 6 章 表操作機能」を参照。

(4) 再定義における桁数の規則

データ項目を再定義する場合、再定義するデータ項目の桁数は再定義されるデータ項目の桁数

と等しいかまたは小さくなければならない。ただし、この規則は 01 レベルのデータ項目には適用されない。ここでデータ項目の桁数とは、そのデータ項目が主記憶上で占めるバイト数である。したがって、COMP-3 項目および SYNCHRONIZED 句を含んでいるような場合は注意を要する。

5.2.3 RENAME 句

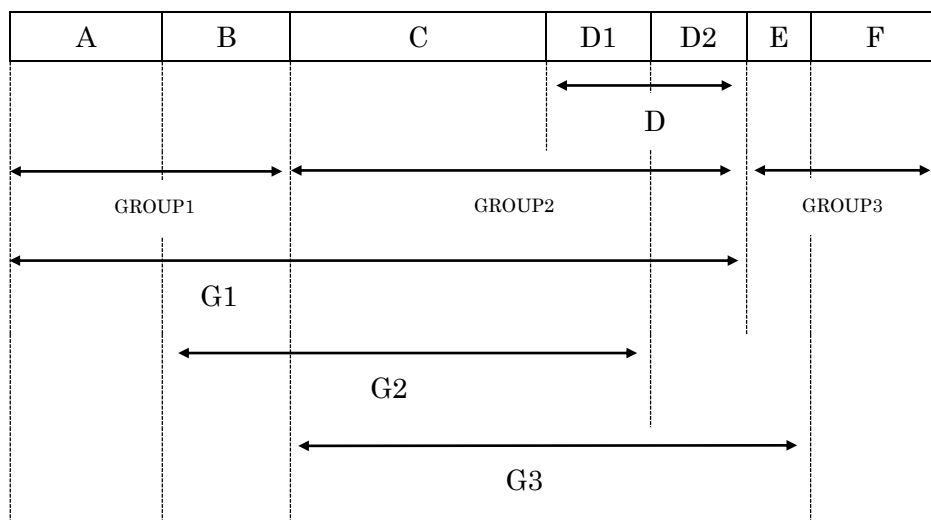
プログラムの流れやレコードの形式によって、あるレコード中のひとかたまりの基本項目、または 1 つ以上の集団項目にまたがるひとかたまりのデータを抽出する問題にぶつかる場合がある。この場合、3 つの解決法がある。すなわち、抽出したい項目すべてを MOVE 文を用いて処理する方法、REDEFINES 句を用いる方法、および RENAME 句を用いる方法がある。

MOVE 文を用いる方法は、実行用プログラムの効率低下を招く。また、REDEFINES 句では集団項目にまたがるデータを抽出することはできない。このような処理において RENAME 句が有効になる。RENAME 句ではレコード全体のデータ記述を組みかえることなく、集団項目にまたがるデータを 1 つにまとめることができる。次に、その例を示す。

```

01 INPUT-DATA.
  02 GROUP1.
    03 A PIC X(3).
    03 B PIC X(2).
  02 GROUP2.
    03 C PIC X(5).
    03 D.
      04 D1 PIC X.
      04 D2 PIC X.
  02 GROUP3.
    03 E PIC X.
    03 F PIC X(2).
66 G1 RENAME GROUP1 THROUGH GROUP2.
66 G2 RENAME B THROUGH D1.
66 G3 RENAME GROUP2 THROUGH E.
```

上記の例の場合、RENAME 句で再命名したデータ名で参照できる範囲は次の図のようになる。



RENAMES 句を用いる場合、次のことに注意しなければならない。

- (1) **RENAMES** 句で基本項目 1 個だけを指定した場合、再命名した項目は指定した基本項目と全く同じ性質をもつ基本項目として取り扱われる。
- (2) **RENAMES** 句で、集団項目または基本項目や集団項目にまたがるグループを指定した場合、再命名した項目は集団項目として取り扱われる。この場合、再命名した名前はひとまとまりのデータに対してつけられた名前であり、通常のコレクション項目につけられたデータ名とは異なる。
すなわち、再命名した名前を修飾名として用いたり、**CORRESPONDING** 句指定がある文の作用対象として用いることはできない。
- (3) 再命名される側のレベル番号は **01, 66, 77** および **88** であってはならない。あるレコードに関する **RENAMES** 句は、そのレコードの最後のデータ記述項の直後にまとめて書かなければならない。
すなわち、再命名される領域は **RENAMES** 記述項に先行していなければならない。

例	[正]		[誤]
	01 A		01 A
	02 B ～.		02 B ～.
	02 C ～.		02 C ～.
	02 D ～.		66 D RENAMES E. [誤]
	02 E ～.		02 E ～.
	66 F RENAMES D.		02 F ～.
	66 G RENAMES E.		66 G RENAMES E.
	66 H RENAMES		66 H RENAMES
	D THRU E.		C THRU B. [誤]

5.2.4 VALUE 句での注意事項

データに **VALUE** 句によって初期値を与える場合、**PICTURE** 句で指定した桁数よりも大きい値を指定すると、データへ割り当てられる値は不定となるため、このような使用法は避けるべきである。

なお、桁落ちが発生する場合、コンパイラが警告メッセージを出力する。

また、**VALUE** 句のないデータの初期値は不定であり、項目の属性や使用する文によっては、実行時不正10進エラーとなることがある。**VALUE** 句のないデータに初期値を設定するには、**-v** オプションを指定する。

5.2.5 数字と符号の重ね合わせに関する注意事項

数字と符号の重ね合わせが行われた外部10進数の値を意識してプログラミングしていると、**-CS** オプション指定時は本COBOL固有の表現形式(既定値)と異なる値であるため、期待どおりの動作が行われない場合がある。したがって、数字と符号の重ね合わせ桁を英数字として扱うような使用法は避けるべきである。

たとえば、符号付き外部10進数を含む集団項目に対して、数字と符号の重ね合わせ桁を部分参照したり、文字列操作文の作用対象として使用する場合などである。**-CS** オプションは、動作に影響がないことを十分確認の上、使用すべきである。

5.3 手続き部

ここでは、COBOL 文のなかで特にわかりにくい文について、例題を中心にして詳しく説明する。

5.3.1 算術文における中間結果

COBOL プログラムにおいて、ある特定の算術文を使用した場合、COBOL コンパイラは、その算術文を実行するのに必要な中間結果格納領域を自動的に生成し、その中間結果格納領域（以下、TS : Temporary Storage）を使用したオブジェクトコードを出力する。したがって、プログラマは TS の生成規則をよく知っていないと期待した結果が得られないことがあるので注意を要する。

基本的には、「算術式中に除算が含まれるとき、小数部の桁数は DMAX（後述する）までしか計算されない」という点を注意すれば、次に説明することを必ずしも完全に知っておく必要はない。

次に、TS を使用する算術文を示す。

- COMPUTE 文
- EVALUATE／IF／PERFORM／SEARCH 文の条件で算術式を使用している場合
- ADD／SUBTRACT／MULTIPLY／DIVIDE 文において
 - (1) 受け取り側データ項目に対して ROUNDED の指定がある場合
 - (2) GIVING 句がある場合
 - (3) ADD／SUBTRACT 文において、加数／減数を表すデータ項目が複数個指定されている場合

上記いずれか 1 つでもあてはまる場合に TS が使用される。

TS の具体的な使用例を次の表に示す。IRi は TS のことを示す。

COBOL プログラム	中間オブジェクトコード ^注
MULTIPLY A BY B ROUNDED	MULTIPLY A BY B GIVING IR1 ADD ZERO IR1 GIVING B ROUNDED
DIVIDE A BY B GIVING X Y	DIVIDE A BY B GIVING IR1 MOVE IR1 TO X MOVE IR1 TO Y
ADD A B C TO X	ADD A B GIVING IR1 ADD IR1 C GIVING IR2 ADD IR2 TO X
COMPUTE X = A + B / C	DIVIDE B BY C GIVING IR1 ADD A IR1 GIVING IR2 MOVE IR2 TO X
IF A + B = X / Y	ADD A B GIVING IR1 DIVIDE X BY Y GIVING IR2 IF IR1 = IR2

注意 COBOL コンパイラは、COBOL プログラムの文を上記の中間オブジェクトコードに示したような COBOL 文に変換し、さらにこれを実際のオブジェクトコードに変換する。

(1) 用語の定義

次の説明の都合上、次の用語を定義しておく。

- IR／IR_i : TSを表す。
- OP₁／OP₂ : コンパイラが内部で生成した算術文の作用対象を示す。これ
 には、実際に利用者が定義した一意名のととき、コンパイラが
 生成した TS の場合がある。
- IP (X) : 作用対象 X における整数部の桁数を表す^{注(1)}。
- DP (X) : 作用対象 X における小数部の桁数を表す^{注(1)}。
- DMAX : 算術式で使用されている全作用対象の DP の最大値^{注(2)}。

ただし

- COMPUTE 文の場合は、受け取り側も含める。
 - COMPUTE 文の受け取り側に ROUNDED 指定があれば、さらに DMAX に 1 を加える。
- 【 Enterprise -Fw オプションを指定した場合は、以下の条件となる。

for Windows】 COMPUTE 文の受け取り側項目に **ROUNDED** が指定され、かつその
ACOS-4 互換 受け取り側項目の小数部桁数が **DMAX** である場合、さらに **DMAX**
に 1 を加える。ただし、**-Fr** オプションが同時指定された場
合、中間結果格納領域の小数部桁数は固定値となり、**DMAX** は
使用されない。

- 比較条件中の算術式の場合には、両辺の作用対象をすべて含める。条件が複合条件になっ
ている場合には、おのこの単純条件ごとに **DMAX** が決められる。左辺が省略されている場合に
は、まず左辺が補われ、その後 **DMAX** が決定される。

注意

注(1) **PICTURE** 句の文字列に“P”を含む場合 **IP(X)**、**DP(X)** は次のようになる。

文字列の右側に“P”がある場合

IP(X) : “P”を含む整数部の桁数

DP(X) : $(-1) \times$ “P”の個数

例 **S99PPP**

IP(X) = 5

DP(X) = -3

文字列の左側に“P”がある場合

IP(X) : $(-1) \times$ “P”の個数

DP(X) : “P”を含む小数部の桁数

例 **SPPP99**

IP(X) = -3

DP(X) = 5

注(2) **DMAX** の値が負になる場合は、0 に改める。

(2) 中間結果格納領域の生成規則

算術文や算術式の書き方によって、次に示すような **TS** が自動的に生成される。

(a) **ADD** / **SUBTRACT** 文

ADD OP_1 OP_2 **GIVING** IR_i

SUBTRACT OP_1 **FROM** OP_2 **GIVING** IR_i

において、**TS** は次のようにとられる。

i. 算術式中のとき

$IP(IR_i) = \max\{IP(OP_1), IP(OP_2)\} + 1$

$DP(IR_i) = \max\{DP(OP_1), DP(OP_2)\}$

i i. ADD/SUBTRACT 文のとき

$$IP(IRi) = \min\{\max\{IP(OP_1), IP(OP_2)\} + 1, 30 - DP(IRi)\}$$

$$DP(IRi) = \max\{DP(OP_1), DP(OP_2)\}$$

(b) MULTIPLY 文

MULTIPLY OP_1 BY OP_2 GIVING IRi

において, TS は次のようにとられる。

$$IP(IRi) = IP(OP_1) + IP(OP_2)$$

$$DP(IRi) = DP(OP_1) + DP(OP_2)$$

(c) DIVIDE 文

DIVIDE OP_1 BY OP_2 GIVING IRi

において, TS は次のようにとられる。

i. 算術式中のとき

$$IP(IRi) = IP(OP_1) + DP(OP_2)$$

$$DP(IRi) = \max\{DMAX, [DP(OP_1) - DP(OP_2)]\}$$

i i. DIVIDE 文のとき

$$IP(IRi) = IP(OP_1) + DP(OP_2)$$

$$DP(IRi) = \max\{\text{GIVING 句の作用対象の DP}\} + e$$

e: **ROUNDED** 指定ありのとき 1, 指定なしのとき 0

各演算中の演算用作業域は, 整数部桁数 30 桁, 小数部桁数 30 桁の上で演算する。その結果, 上記の (a), (b) および (c) の IP が 30 桁を超える場合, または DP が 30 桁を超える場合, 桁落ちが発生するので注意されたい。

なお, COBOL コンパイラでは, 算術式中で使用する 1 つの TS の最大桁数は 30 としているので, 上記 (a), (b) および (c) の $IP(IRi)$, $DP(IRi)$ を求める際 $IP(IRi) + DP(IRi)$ の値が 30 を超えるときには, 次のように算術式中で使用する TS の桁落ちが生じるので注意されたい。

注意

(注 1) $DP(IRi) > DMAX$, かつ $IP(IRi) + DMAX \leq 30$ のとき整数部は $IP(IRi)$ がとられ, 小数部は $30 - IP(IRi)$ がとられる。すなわち, 小数部に桁落ちが生じる可能性がある。

(注 2) $DP(IRi) > DMAX$, かつ $IP(IRi) + DMAX > 30$ のとき, 整数部は $30 - DMAX$ がとられ, 小数部は $DMAX$ がとられる。すなわち, 小数部, 整数部ともに桁落ちが生じる可能性がある。

(注 3) $DP(IRi) \leq DMAX$ のとき, 整数部は $30 - DP(IRi)$ がとられ, 小数部は $DP(IRi)$ がとられる。すなわち, 整数部の桁落ちが生じる可能性がある。

(d) べき乗 (**))

 $OP_1 ** OP_2$

において, TS は次のようにとられる。

OP₂ が定数で, 値=0 のとき,

IP(IR_i) = 1

DP(IR_i) = 0

i. OP₂ が定数または一意名のとき

DP(IR_i) = MAX {DMAX, DP(OP₁)}

IP(IR_i) = 30 - DP(IR_i)

注意 TS の整数部の桁落ちが生じる場合には, 実行時, 中間結果において桁あふれ (size error) 条件の生じる可能性がある。桁あふれ条件が中間結果において生じた場合, もし ON SIZE ERROR 句の指定がある場合には, SIZE ERROR 指定の無条件文が実行され, SIZE ERROR 句の指定がなかった場合の結果は不定であり, NOT ON SIZE ERROR 句の指定がある場合にはその無条件文が実行される。TS の大きさが 30 桁を超える場合には, 翻訳時にその旨の診断メッセージ (warning) が出力される。演算用作業域での桁落ちについては, 上記の事項は適用されない。また, 結果は不定となる。

(3) 生成規則の具体例

次に, 実際に利用者が書いた算術文がどのようにして演算されるかを, いくつかの実例を示して説明する。次の例題で使用しているデータ名の名前とその形式は, 次のとおりとする。

A: S9 (5)	F: S9 (9) V9 (1)
B: S9 (3) V9 (7)	G: S9 (4) V9 (8)
C: S9 (4)	X: S9 (8) V9 (8)
D: S9 (3) P (4)	Y: S9 (3) V9 (4)
E: SP (8) 9 (2)	Z: S9 (7)

(a) 算術文

ADD A B GIVING X Y

↓

ADD A B GIVING IR₁

MOVE IR₁ TO X Y

ここで, IR₁ の形式は S9 (6) V9 (7) となる。

ADD 1000, D GIVING Z

↓

ADD 1000, D GIVING IR₁

MOVE IR₁ TO Z

ここで, IR₁ の形式は S9 (8) となる。

SUBTRACT D FROM E GIVING X Y

↓

SUBTRACT D FROM E GIVING IR₁

MOVE IR₁ TO X Y

ここで、IR₁ の形式は S9(8)V9(10) となる。

COMPUTE X = B * B * G / E

↓

MULTIPLY B BY B GIVING IR₁

MULTIPLY IR₁ BY G GIVING IR₂

DIVIDE IR₂ BY E GIVING IR₃

MOVE IR₃ TO X.

ここで、IR₁～IR₃ の形式は次のとおりである。

IR₁ : S9(6)V9(14)

IR₂ : S9(10)V9(20)

IR₃ : S9(20)V9(10)

(b) IF 文

IF A + B / C = G * D / A

↓

DIVIDE B BY C GIVING IR₁

ADD A IR₁ GIVING IR₂

MULTIPLY G BY D GIVING IR₃

DIVIDE IR₃ BY A GIVING IR₄

IF IR₂=IR₄

ここで、IR₁～IR₄ の形式は次のとおりである。

IR₁ : S9(7)V9(8)

IR₂ : S9(8)V9(8)

IR₃ : S9(11)V9(4)

IR₄ : S9(11)V9(8)

(4) 算術演算に関する注意事項

算術演算の精度の問題は非常に重要で、中間結果で定まる TS の大きさを考慮しないと期待した結果が得られないことがあるので注意されたい。

(a) 除算が含まれているとき；例で示す。

例 i. 77 A PIC 99 VALUE 10.

:

IF A / 3 > 3.33 GO TO WHERE-TO

において、 $A/3=3.33\cdots$ が代数的に正しい結果であるから上記の条件は真になるように思われる。しかし、(2)の除算の中間結果格納領域の生成規則に従って計算すると、 $A/3=3.33$ となり、上記の条件は偽となる。

例 ii. 77 A PIC 9(3)
 77 B PIC 9(3) VALUE 45.
 77 C PIC 9(3) VALUE 50.

：

 COMPUTE A = B / C * 100.

において、計算結果は **A=0** となる。もし、これが % を求める式であるならば、上記のような書き方では期待した結果は得られない。% を求めるためには、**COMPUTE A = B * 100 / C** のようにするか、B のデータ定義を **PIC 9(3)V9(2)** のようにしておくなどの考慮が必要である。

(b) べき乗が含まれているとき; **A ** B** において、

- i. B が非整数(小数部をもつ数字項目)のときは内部で型変換を行い浮動小数点演算を行うので、演算結果は近似値となる。
 ただし、**A < 0** ならば、桁あふれ条件となる。
- ii. **A=0** かつ **B=0** ならば、桁あふれ条件となる。
- iii. **B=0** ならば、A がどんな値でも答は 1 になる。
 ただし、(4) (b) ii. の場合は除く。
- iv. **A=0** ならば、B が正の場合は 0 になる。
 B が 0 または負の場合は桁あふれ条件となる。
- v. B が負のときは、内部で型変換を行い浮動小数点演算を行うので、演算結果は近似値となる。
- vi. A が非整数(小数部をもつ数字項目)のときは内部で型変換を行い浮動小数点演算を行うので、演算結果は近似値となる。

【Enterprise】 -Fr オプションに小数部桁数を指定することにより、
ACOS-4 互換 中間結果格納領域を固定化することができる。
 決定規則は次のとおりである。

整数部桁数 +Fr で指定された小 数部桁数	中間結果の桁数	
	整数部桁数	小数部桁数
≤ 30	整数部桁数	Fr で指定された 小数部桁数
> 30	30 - Fr で指定された 小数部桁数	Fr 指定された 小数部桁数

5.3.2 CORRESPONDING 指定

入力レコード内のすべての項目をプリントレコード内へ移送したり, 旧マスタファイルのレコードの内容を新マスタファイルのレコードに移送したりするのに, 個々の項目ごとに **MOVE** 文を書くのは煩雑である。

COBOL には, このようなコーディングの手間を省くために **CORRESPONDING** 句が用意されている。

```

01 INPUT-REC.
    02 SHAINBANGO PIC 9(6).
    02 SHIMEI      PIC X(20).
    02 HONKYU      PIC 9(6).
    02 KAKYU       PIC 9(6).
    02 SHIGOTOKYU PIC 9(6).
    02 FUYOKYU     PIC 9(6).
    02 KYUYO       PIC 9(7).
01 PRINT-REC.
    02 SHAINBANGO PIC 9(6).
    02 FILLER      PIC X(5) VALUE SPACE.
    02 SHIMEI      PIC X(20).
    02 FILLER      PIC X(5) VALUE SPACE.
    02 HONKYU      PIC ZZZ, ZZ9.
    02 FILLER      PIC X(5) VALUE SPACE.
    02 KAKYU       PIC ZZZ, ZZ9.
    02 FILLER      PIC X(5) VALUE SPACE.
    02 SHIGOTOKYU PIC ZZZ, ZZ9.
    02 FILLER      PIC X(5) VALUE SPACE.
    02 FUYOKYU     PIC ZZZ, ZZ9.
    02 FILLER      PIC X(5) VALUE SPACE.
    02 KYUYO       PIC Z, ZZZ, ZZ9.

```

上述の例で示した **INPUT-REC** の個々の項目を **PRINT-REC** の対応する項目に移送するためには, 通常次のように書かなければならない。

```

MOVE SHAINBANGO IN INPUT-REC TO SHAINBANGO IN PRINT-REC.
MOVE SHIMEI IN INPUT-REC TO SHIMEI IN PRINT-REC.
MOVE HONKYU IN INPUT-REC TO HONKYU IN PRINT-REC.
      :
MOVE KYUYO IN INPUT-REC TO KYUYO IN PRINT-REC.
    
```

これを, **CORRESPONDING** 句を用いることにより, 次のように書くことができる。

```

MOVE CORRESPONDING INPUT-REC TO PRINT-REC.
    
```

CORRESPONDING 句は **MOVE** 文だけでなく, **ADD** および **SUBTRACT** 文でも用いることができる。

(1) **CORRESPONDING** 句の作用対象

CORRESPONDING 句の作用対象をそれぞれ d_1 および d_2 とした場合, 次の規則に従っていなければならない。

- (a) d_1 および d_2 は, 集団項目の名前でなければならない。
- (b) d_1 および d_2 のデータ記述中に **REDEFINES** 句や **OCCURS** 句を含んでもよい。
- (c) d_1 および d_2 は **REDEFINES** 句や **OCCURS** 句を含むデータ項目に従属していてもよい。
- (d) d_1 および d_2 が 66 レベル番号をもつデータ項目であってはならない。また, d_1 および d_2 のデータ記述中に **USAGE IS INDEX** 句があってはならない。

次に, **CORRESPONDING** 句の正しい例を示す。

```

02 d1 OCCURS 10 TIMES.
   03 e1 ...
   03 e2 ...
02 d2.
   03 e1 ...
   03 e2 ...

MOVE CORRESPONDING d1 (3) TO d2.
MOVE CORRESPONDING d2 TO d1 (5) .
MOVE CORRESPONDING d1 (3) TO d1 (5) .
    
```

または

02 e₁ OCCURS 10 TIMES.

03 e₂

03 d₁ REDEFINES e₂.

04 e₃

04 e₄

02 d₂ OCCURS 10 TIMES.

03 e₃

03 e₄

MOVE CORRESPONDING d₁ (n₁) TO d₂ (m₁) .

MOVE CORRESPONDING d₂ (m₁) TO d₁ (n₁)

(2) CORRESPONDING 句の動作

MOVE, ADD および SUBTRACT 文において, CORRESPONDING 句が指定されていると, d₁ および d₂ の中に次の規則をすべて満足するデータ項目がある場合, それら是对応すると呼び, 対応するデータ項目同士で MOVE, ADD または SUBTRACT 文が実行される。

- (a) d₁ に従属するデータ項目と d₂ に従属するデータ項目が同じ名前をもち (FILLER は除く), 両方のデータ項目の修飾語が d₁ または d₂ に至る直前まで同じであること。
- (b) MOVE 文のときは, 少なくとも一方のデータ項目が基本項目であり, ADD, SUBTRACT 文のときは, 両方のデータ項目が数字基本項目であること。
- (c) レベル番号 66 をもつデータ項目または REDEFINES 句, OCCURS 句もしくは USAGE IS INDEX 句が指定されているデータ項目ではないこと。

次にいくつかの例を示す。

例(1) 01 d ₁ . 02 e ₁ PIC 9(5). 02 e ₂ PIC X(5). 02 e ₃ . 03 e ₄ PIC 9(3). 03 e ₅ PIC 9(4).	01 d ₂ . 02 e ₂ PIC X(5). 02 e ₃ . 03 e ₄ PIC X(2). 03 e ₅ PIC 9(4). 02 e ₆ . 03 e ₁ PIC 9(5).

上記の例において、MOVE CORRESPONDING d₁ TO d₂は次の 3 つの MOVE 文を書いたのと同じになる。

MOVE e₂ OF d₁ TO e₂ OF d₂

MOVE e₄ OF d₁ TO e₄ OF d₂

MOVE e₅ OF d₁ TO e₅ OF d₂

また、ADD CORRESPONDING d₁ TO d₂は次の 1 つの ADD 文を書いたのと同じになる。

ADD e₅ OF d₁ TO e₅ OF d₂

例(2) 01 d ₁ . 02 e ₁ PIC 9(5). 02 e ₂ REDEFINES e ₁ PIC X(5). 02 e ₃ USAGE INDEX. 02 e ₄ . 03 e ₅ . 04 e ₆ PIC X(5).	01 d ₂ . 02 e ₁ PIC 9(5). 02 e ₂ PIC X(5). 02 e ₃ PIC X(6). 02 e ₄ . 03 e ₆ PIC X(5).

上記の例において、MOVE CORRESPONDING d₁ TO d₂を書くと、次の 1 つの MOVE 文を書いたのと同じである。

MOVE e₁ OF d₁ TO e₁ OF d₂

5.3.3 COMBINE 文

論理演算を受け取る一意名が可変繰り返し項目を含む英数字集団項目のとき、一意名の大きさはその最大長が取られる。

【Standard】 **-Qd** オプションを指定することにより、
一意名が可変繰り返し項目を含む英数字集団項目のとき、すべての一意名の大きさはその最大長が取られる。

5.3.4 IF 文と条件式

条件式(**conditional expression**)は、条件の真理値に従って実行用プログラムの 2 つの行き先のうち、いずれかを選択する場合に用いる。条件式は、**EVALUATE** 文、**IF** 文、**PERFORM** 文および **SEARCH** 文で利用できる。条件式には単純条件と複合条件がある。また複合条件は多くの条件からなるがこの節では特に **IF** 文と比較条件に関して詳しく説明する。

(1) 比較条件

比較条件は、2 つの作用対象の大きさを比較するのに使用する。比較条件の結果は、両作用対象の項類や用途によって表 5-2 に示すようになる。

表中、各記号は次の意味をもつ。

AN(文字比較)

(a) 作用対象の桁数が等しい場合。

両方の作用対象の対応する文字位置の文字を、最左端から最右端へ向かって等しくない文字が現れるか、または最右端に達するまで 1 組ずつ比較する。それぞれの比較結果が最右端に至るまですべて等しいとき、これら 2 つの作用対象は等しいものとされる。

等しくない文字の組がはじめて現れたとき、大小順序でより高い位置の文字を含む作用対象の方をより大きいものとする。

(b) 作用対象の桁数が異なる場合。

短い方の作用対象の右側に、長い方の作用対象の桁数に等しくなるまで空白があるものとみなして、(a)と同じように比較を行う。

【Enterprise】 **-Fd** オプションを指定することにより、
ACOS-4 互換 整数でない外部 10 進数と表意定数および英数字定数を比較することができる。

また、算術式と一意名 (英数字/英数字編集) の比較を文字比較とする。

AN^A (文字比較—外部 10 進数)

外部 10 進数の絶対値 (符号なしはそのまま, 符号付きなら絶対値) と他方の作用対象の桁数の間で文字比較を行う。

N (数字比較)

代数的な値に基づいて大小比較を行う。

必要ならば, 小数点の桁合わせが行われる。

N^C (数字比較—内部表現形式の変換)

内部表現形式の変換 (10 進 ⇄ 2 進, 外部 10 進 ⇄ 内部 10 進) を行った後, 数字比較を行う。

F (表意定数比較)

作用対象の桁数だけ表意定数が内部的に作られ, それとの文字比較が行われる。

I (出現番号の比較)

数字項目と指標名の間で出現番号の比較を行う。

ID (指標データの比較)

指標の値の比較を行う。

指標の項目長は, 16 バイトである。

-Qc オプションを指定することにより, 指標名や指標データ項目の項目長を 8 バイトにすることができる。

G (日本語比較)

(a) 作用対象の桁数が等しい場合。

両方の作用対象の対応する文字位置の文字を, 最左端から最右端へ向かって等しくない文字が現れるか, または最右端に達するまで 1 組ずつ比較する。それぞれの比較結果が最右端に至るまですべて等しいとき, これら 2 つの作用対象は等しいものとされる。

等しくない文字の組がはじめて現れたとき, 大小順序でより高い位置の文字を含む作用対象の方をより大きいものとする。

(b) 作用対象の桁数が異なる場合。

短い方の作用対象の右側に, 長い方の作用対象の桁数に等しくなるまで日本語文字の空白があるものとみなして, (a) と同じように比較を行う。

G^F (日本語表意定数比較)

作用対象の桁数だけ表意定数が内部的に作られ, それとの日本語比較が行われる。

P (ポインタ比較)

- (a) 一意名 が指定された場合、内部変換なしにそのまま比較される。
- (b) **ADDRESS OF** 一意名 が指定された場合、一意名 に対応するデータ領域の左端の実効アドレスが求められた後、内部変換なしにそのまま比較される。
- (c) **NULL** が指定されると、空のポインタ値と比較される。空のポインタ値とは、8 バイトの全ビットが”0”で表される値である。

B (ブーリアン比較)

- (a) 作用対象の長さが等しい場合。
両方の作用対象のブール文字位置を最左端から最右端へ向かって等しくない文字の組が現れるか、または最右端に達するまで一組ずつ比較する。それぞれの比較が、最右端に至るまですべて等しいとき、これらの 2 つの作用対象は、等しいものとみなす。
- (b) 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端に、長い方の作用対象の長さに等しくなるまでブール文字“0”があるものとみなして、(a)と同じように比較を行う。

B^Z (ブーリアン比較-表意文字 ZERO)

作用対象の桁数だけブール文字の“0”が内部的に作られ、それとのブーリアン比較が行われる。

B^A (ALL ブール定数比較)

作用対象の桁数だけ表意定数で指定された値が左詰めで繰り返し取られ、それとのブーリアン比較が行われる。

O (オブジェクト比較)

両方の作用対象の指すオブジェクトが同じ場合、等しいものとみなす。**NULL** が指定されると、既定義オブジェクト参照 **NULL** と比較される。
既定義オブジェクト参照 **NULL** とは、8 バイトの全ビットが”0”で表される値である。

表 5-2 比較の規則

第1作用対象 (左辺)	第2作用対象 (右辺)	集 団 項 目	英 字	英 数 字	英 数 字 編 集	ブ ー ル	日 本 語	日 本 語 編 集	内部10進		外部10進				2進数		数 字 編 集	指 標 名	指 標 デ ー タ 項 目	オ ブ ジ ェ ク ト	ポ イ ン タ 名	デ ー タ ポ イ ン タ デ ー タ 項 目	N U L L	表意定数						数字定数		英 数 字 定 数	ブ ー ル 定 数	日 本 語 定 数	
									整 数	非 整 数	整 数		非 整 数		整 数	非 整 数								Z E R O	S P A C E	A L L ブ ー ル 定 数	A L L 日 本 語 定 数	そ の 他	整 数	非 整 数					
											符 号 あ り	符 号 な し	符 号 あ り	符 号 な し																					
集団項目		AN	AN	AN	AN	—	AN	AN	—	—	AN	AN	—	—	—	—	AN	—	—	—	—	—	—	F	F	—	F	F	AN	—	AN	—	—	AN	
英字		AN	AN	AN	AN	—	—	—	—	—	AN ^A	AN	—	—	—	—	AN	—	—	—	—	—	—	—	F	F	—	—	F	AN ^A	—	AN	—	—	—
英数字		AN	AN	AN	AN	—	—	—	—	—	AN ^A	AN	—	—	—	—	AN	—	—	—	—	—	—	—	F	F	—	—	F	AN ^A	—	AN	—	—	—
英数字編集		AN	AN	AN	AN	—	—	—	—	—	AN ^A	AN	—	—	—	—	AN	—	—	—	—	—	—	—	F	F	—	—	F	AN ^A	—	AN	—	—	—
ブール		—	—	—	—	B	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	B ^Z	—	B ^A	—	—	—	—	—	B	—	—
日本語		AN	—	—	—	—	G	G	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	G ^F	—	G ^F	—	—	—	—	—	G	—
日本語編集		AN	—	—	—	—	G	G	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	G ^F	—	G ^F	—	—	—	—	—	G	—
内部10進	整数	—	—	—	—	—	—	N	N	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	—	I	—	—	—	—	—	N	—	—	—	—	N	N	—	—	—	—	
	非整数	—	—	—	—	—	—	N	N	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	—	—	—	—	—	—	—	N	—	—	—	—	N	N	—	—	—	—	
外部10進	整数	符号あり	AN	AN ^A	AN ^A	AN ^A	—	—	—	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	AN ^A	I	—	—	—	—	—	N	F	—	—	F	N	N	AN ^A	—	—	—	
		符号なし	AN	AN	AN	AN	—	—	—	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	AN	I	—	—	—	—	—	N	F	—	—	F	N	N	AN	—	—		
	非整数	符号あり	—	—	—	—	—	—	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	—	—	—	—	—	—	—	N	—	—	—	—	N	N	—	—	—	—	
		符号なし	—	—	—	—	—	—	—	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	—	—	—	—	—	—	—	N	—	—	—	—	N	N	—	—	—	—	
2進数	整数	—	—	—	—	—	—	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	—	I	—	—	—	—	—	N	—	—	—	—	N	N	—	—	—	—	
	非整数	—	—	—	—	—	—	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	N ^C	—	—	—	—	—	—	—	N	—	—	—	—	N	N	—	—	—	—	
数字編集		AN	AN	AN	AN	—	—	—	—	—	AN ^A	AN	—	—	—	—	AN	—	—	—	—	—	—	—	F	F	—	—	F	AN ^A	—	AN	—	—	—
指標名		—	—	—	—	—	—	—	I	—	I	I	—	—	I	—	I	ID	ID	—	—	—	—	—	I	—	—	—	I	—	—	—	—	—	
指標データ項目		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
オブジェクト		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	—	—	—	0	—	—	—	—	—	—	—	—	—	—	—	
ポインタ名		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	P	—	P	P	—	—	—	—	—	—	—	—	—	—	—	
データポインタデータ項目		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	P	P	P	—	—	—	—	—	—	—	—	—	—	—	
NULL		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	P	P	P	—	—	—	—	—	—	—	—	—	—	—	—	
表意定数	ZERO	F	F	F	F	B ^Z	—	—	N	N	N	N	N	N	N	N	F	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	SPACE	F	F	F	F	—	G ^F	G ^F	—	—	F	F	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	ALLブール定数	—	—	—	—	B ^A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	ALL日本語定数	F	—	—	—	—	G ^F	G ^F	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
数字定数	その他	F	F	F	F	—	—	—	—	—	F	F	—	—	—	—	F	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	整数	AN	AN ^A	AN ^A	AN ^A	—	—	—	N	N	N	N	N	N	N	N	AN ^A	I	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	非整数	—	—	—	—	—	—	—	N	N	N	N	N	N	N	N	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
英数字定数		AN	AN	AN	AN	—	—	—	—	—	AN ^A	AN	—	—	—	—	AN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ブール定数		—	—	—	—	B	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
日本語定数		AN	—	—	—	—	G	G	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

(2) 字類条件

字類条件は、作用対象が数字、英字、英小文字、英大文字または環境部の特殊名段落で定義した **CLASS** 句によって指定された文字集合中の文字だけを含まかどうかを調べるのに使用する。

【Enterprise】 -Fe オプションを指定することにより、

ACOS-4 互換 **NUMERIC** 検査の内容を変更することができる。

- ・偶数桁の内部 10 進数の最左端上位 4 ビットが 0 以外の場合、偽とする
(-CN オプションが指定されていない時)。
- 不正 10 進エラーにはならない。
- ・外部 10 進数の独立符号部が空白の場合、真とする。
不正 10 進エラーにはならない。

【Standard】 -Qh オプションを指定することにより、

NUMERIC 検査の内容を変更することができる。

- ・偶数桁の内部 10 進数の最左端上位 4 ビットが 0 以外の場合、偽とする。
(-CN オプションが指定されていない時)。
- ・外部 10 進数の独立符号が空白の場合、偽とする。

5.3.5 INSPECT 文

INSPECT 文は、文字列中の特定の文字または文字列の出現回数を数えたり、別の文字または文字列で置き換えたりするときに用いる。また、そのときに数えたり置き換えたりする文字列の範囲を規定することもできる。

(1) 文字列検査の対象になる範囲

INSPECT 文によって文字列の出現回数のカウントや文字列の置き換えの対象になる範囲を定めるために、**BEFORE** 句と **AFTER** 句がある。

BEFORE 句を指定すると、**BEFORE** 句で指定した文字列(最初に見つかったもの)より前にある文字だけが検査の対象になる。**BEFORE** 句で指定した文字列が見つからない場合は、すべての文字が検査の対象になる。

AFTER 句を指定すると、**AFTER** 句で指定した文字列(最初に見つかったもの)より後にある文字だけが検査の対象になる。**AFTER** 句で指定した文字列が見つからない場合、文字列の検査はされない。

次に、いくつかの例を示す。例中実線で囲んだ部分が検査の対象になる。

```

INSPECT word ~ BEFORE "*"
word の内容      ABCD*EFG
word の内容      ABCDEFGH
word の内容      *ABCDEFG      検査されない

INSPECT word ~ AFTER "/"
word の内容      /*XYZ123
word の内容      AB/C//*Z
word の内容      AB/CDEF      検査されない

```

BEFORE 句および AFTER 句のいずれも指定していない場合は、すべての文字が検査の対象になる。

BEFORE 句および AFTER 句をともに指定する場合は、AFTER 句で決められる検査対象でかつ BEFORE 句で決められる検査対象が、INSPECT 命令での検査対象になる。次に、いくつかの例を示す。例中実線で囲んだ部分が検査の対象になる。

```

INSPECT word ~ BEFORE "*" AFTER "/"
word の内容      AB/CDE*FG
word の内容      AB/CDEFGH
word の内容      AB*CDE/FG

```

(2) TALLYING 指定

文字や文字列の出現回数を数える方法には、次に示す 3 つの指定方法がある。

(a) ALL 指定

この指定では、対象となる範囲内に現れる指定された文字列の出現回数を数える。

(b) LEADING 指定

この指定では、対象となる範囲内で、指定された文字列以外の文字列が出てくるまでに表れた指定された文字列の出現回数を数える。

(c) CHARACTERS 指定

この指定では、対象となる範囲の桁数を調べる。

いずれの指定においても、同一の文字が 2 回以上カウントされることはない。文字列の検査が終わると、出現回数が指定したカウンタに加えられる。したがって、実際の出現回数を調べたい場合は、INSPECT 文の実行に先立ってカウンタをゼロクリアしておかなければならない。

次に、いくつかの例を示す。例中カウンタはゼロクリアされているものとする。

```

INSPECT word TALLYING count1 FOR ALL "*/"
word の内容:*//*/*XYZ count1:2
word の内容:ABCD/*XY/ count1:0

```

INSPECT word TALLYING count1 FOR LEADING “*” AFTER “/”

word の内容:**/**/** count1:0

word の内容:**XYZ123/ count1:0

word の内容:/**XYZ/* count1:3

INSPECT word TALLYING count1 FOR ALL “AA” BEFORE “XYZ”

word の内容:AAABAAXYZ count1:2

word の内容:123AABAAAA count1:3

word の内容:XYZ12AA345 count1:0

INSPECT word TALLYING count1 FOR CHARACTERS BEFORE “*”

word の内容:*123456789 count1:0

word の内容:1234*567*8 count1:4

word の内容:1234567890 count1:10

(3) REPLACING 指定

文字や文字列を指定した文字や文字列で置き換える方法には、次の 4 つの指定方法がある。

(a) ALL 指定

この指定では、対象となる範囲内で、指定された文字列と合致した文字列を同一桁数の文字列と置き換える。

(b) LEADING 指定

この指定では、対象となる範囲内で、指定された文字列以外が現れるまでの文字列を指定した同一桁数の文字列と置き換える。

(c) FIRST 指定

この指定では、対象となる範囲内で、最初に現れた指定された文字列を同一桁数の文字と置き換える。

(d) CHARACTERS 指定

対象となる範囲内のすべての文字を指定された文字と置き換える。

次に、いくつかの例を示す。

INSPECT word REPLACING ALL “X” BY “A”

word(実行前):XAXAXAXAXA word(実行後):AAAAAAAAAA

INSPECT word REPLACING LEADING “X*” BY “ZZ”

word(実行前):X*X*/XX word(実行後):ZZZZ/*XX

word(実行前):ZX*Z*/12 word(実行後):ZX*Z*/12

INSPECT word REPLACING FIRST “Z” BY ZERO AFTER SPACE

word(実行前): ZZZ/ 0ZZ word(実行後): ZZZ/ *0Z

word(実行前): ZZZZZZZZ word(実行後): ZZZZZZZZ

word(実行前): △****ZZZ word(実行後): ****0ZZ

INSPECT word REPLACING CHARACTERS BY “Z” AFTER “*”

word(実行前): *1234567 word(実行後): *ZZZZZZZ

word(実行前): 12345678 word(実行後): 12345678

5.3.6 MOVE 文

(1) 転記の規則

MOVE 文は、データを編集規則に従っていくつかのデータ領域に移す。次の表に MOVE 文における作用対象の組み合わせとその規則を示す。

表 5-3 転記の規則詳細

受け取り側項目 送し出し側項目		集団項目	英字	英数字	英数字編集	ブール	日本語	日本語編集	内部10進		外部10進		2進数		数字編集
									整数	非整数	整数	非整数	整数	非整数	
集団項目		AN	AN	AN	AN	—	AN	AN	AN	AN	AN	AN	AN	AN	AN
英字		AN	AN	AN	ANE	—	G ^U	GE ^U	—	—	—	—	—	—	—
英数字		AN	AN	AN	ANE	BO ^I	G ^U	GE ^U	N ^P	N ^P	N	N	N ^B	N ^B	NE
英数字編集		AN	AN	AN	ANE	—	G ^U	GE ^U	—	—	—	—	—	—	—
ブール		—	—	AN ⁹	ANE ⁹	BO	—	—	—	—	—	—	—	—	—
日本語		AN	—	—	—	—	G	GE	—	—	—	—	—	—	—
日本語編集		AN	—	—	—	—	G	GE	—	—	—	—	—	—	—
内部10進	整数	AN	—	AN ^{UA}	ANE ^U	—	G ^{UAC}	GE ^{UA}	N	N	N ^U	N ^U	N ^B	N ^B	NE ^U
	非整数	AN	—	—	—	—	—	—	N	N	N ^U	N ^U	N ^B	N ^B	NE ^U
外部10進	整数	AN	—	AN ^A	ANE ^A	—	G ^{AC}	GE ^{AC}	N ^P	N ^P	N	N	N ^B	N ^B	NE
	非整数	AN	—	—	—	—	—	—	N ^P	N ^P	N	N	N ^B	N ^B	NE
2進数	整数	AN	—	AN ^{UA}	ANE ^U	—	G ^{UAC}	GE ^{UA}	N ^U	N ^U	N ^U	N ^U	N	N	NE ^U
	非整数	AN	—	—	—	—	—	—	N ^U	N ^U	N ^U	N ^U	N	N	NE ^U
数字編集		AN	—	AN	ANE	—	G ^U	GE ^{AC}	N ^{RP}	N ^{RP}	N ^R	N ^R	N ^{RB}	N ^{RB}	NE ^R
表意定数	ZERO	F	—	F	ANE ^F	BO ^F	(注1)	(注2)	N ^F	N ^F	N ^F	N ^F	N ^F	N ^F	NE ^F
	SPACE	F	F	F	ANE ^F	—	F	GE ^F	—	—	—	—	—	—	—
	ALL英数字定数	F	F	F	ANE ^F	—	G ^{UF}	GE ^{UF}	N ^F	N ^F	N ^F	N ^F	N ^F	N ^F	NE ^F
	ALLブール定数	—	—	F	ANE ^F	BO ^F	—	—	—	—	—	—	—	—	—
	ALL日本語定数	F	—	—	—	—	F	GE ^F	—	—	—	—	—	—	—
数字定数	その他	F	—	F	ANE ^F	—	(注1)	(注2)	—	—	—	—	—	—	—
	整数	AN	—	AN ^A	ANE ^A	—	G ^{AC}	GE ^{AC}	N ^P	N ^P	N	N	N ^B	N ^B	NE
	非整数	AN	—	—	—	—	—	—	N ^P	N ^P	N	N	N ^B	N ^B	NE
英数字定数		AN	AN	AN	ANE	BO ^I	G ^U	GE ^U	N ^P	N ^P	N	N	N ^B	N ^B	NE
ブール定数		—	—	AN ⁹	ANE ⁹	BO	—	—	—	—	—	—	—	—	—
日本語定数		AN	—	—	—	—	G	GE	—	—	—	—	—	—	—

注意 (注 1)【Enterprise】— (転記不可)

【Standard】F (表意定数転記)

(注 2)【Enterprise】— (転記不可)

【Standard】GE^{CF} (日本語編集転記: 表意定数)

表中, 各記号は次の意味をもつ。

AN (英数字転記)

- (a) 送り出し側項目が受け取り側項目より大きい場合
送り出し側項目のデータは左側から 1 文字ずつ, 左詰めで受け取り側項目に送られ, 余分な桁は切り捨てられる。
- (b) 送り出し側項目が受け取り側項目より小さい場合
送り出し側項目の右側に, 受け取り側項目と等しくなるだけ空白があるものとして, 受け取り側項目に転記される。
- (c) 項目の桁数が等しい場合
送り出し側項目のデータがそのまま受け取り側に転記される。
- (d) 受け取り側項目に **JUSTIFIED** 句の指定がある場合は, データは右詰めで転記される。詳細については『COBOL 言語説明書』「9. 14. 21 JUSTIFIED 句」を参照。

AN^A (英数字転記—外部 10 進)

送り出し側項目に外部 10 進数が格納されているとみなして受け取り側項目に左詰めで転記される。送り出し側項目が符号付き数字項目のときは絶対値が転記される。その他は英数字転記と同じである。

AN^{UA} (英数字転記—内部 10 進)

内部 10 進数が外部 10 進数に変換され, その絶対値が受け取り側項目に左詰めで転記される。その他は英数字転記と同じである。

AN^{DA} (英数字転記—2 進数)

2 進数が外部 10 進数 (単精度の場合は 5 桁, 倍精度の場合は 10 桁) に変換され, その絶対値が受け取り側項目に左詰めで転記される。その他は英数字転記と同じである。

AN⁹ (英数字転記—ブール文字→0 と 1 の文字に変換)

ブール文字が“0”のとき, 外部 10 進形式の文字“0”に, またブール文字が“1”のとき, 外部 10 進形式の文字“1”に変換されてから, 受け取り側項目に左詰めで転記される。他は英数字転記と同じである。

ANE, ANE^{UA}, ANE^A, ANE⁹, ANE^{DA}, ANE^F (英数字編集転記)

送り出し側項目が, 受け取り側項目の有効桁と同じ桁数の英数字項目に, それぞれ **AN**, **AN^{UA}**, **AN^A**, **AN⁹**, **AN^{DA}**, **F** の規則に従って転記され, それが英数字編集項目に転記されたかのように実行される。編集方法については、『COBOL 言語説明書』「9. 14. 30 PICTURE 句」を参照。

N(数字転記)

転記の際に小数点の桁合わせや符号の処理などが行われる。

- (a) 受け取り側が符号付き数字項目のときは、送り出し側項目の符号が受け取り側項目の符号になる。このとき、必要ならば符号の表現形式の変換が行われる。送り出し側項目が符号なしの場合は、符号がついていないとみなして転記される。
- (b) 受け取り側項目が符号なし数字項目のときは、送り出し側項目の絶対値が送られ、符号はつけられない。

ACOS-4 互換 -Cr オプションを指定することにより、
用途が表示用で同桁同符号の数字項目同士の場合、英数字転記を行う。

【 Enterprise 】 -Fs オプションを指定することにより、
ACOS-4 互換 英数字(英数字項目、英数字定数、ALL 英数字定数)から外部 10 進数への数字転記において、ゾーン部分のビットの補正を行わない。

N^P(数字転記:外部 10 進→内部 10 進変換)

送り出し側項目が内部 10 進数に変換され、数字転記が行われる。

N^U(数字転記:内部 10 進→外部 10 進変換)

送り出し側項目が外部 10 進数に変換され、数字転記が行われる。

N^B(数字転記:10 進→2 進変換)

送り出し側項目が 2 進数に変換され、数字転記が行われる。

N^D(数字転記:2 進→10 進変換)

送り出し側項目が 10 進数に変換され、数字転記が行われる。

N^F(数字転記:表意定数)

- (a) ZERO の場合は、受け取り側項目の用途に応じて値ゼロが転記される。
- (b) 表意定数の ALL 定数でかつ定数が数字だけの場合は、次のようになる。
 - i. 受け取り側項目が外部 10 進、内部 10 進項目の場合は、整数部に左詰めで指定された定数を整数部の桁数分だけ繰り返したものが転記される。

MOVE ALL "123" TO A

A が 9 (5) のとき 12312

A が 9 (6) のとき 123123

A が 9 (3) V9 (4) のとき 1230000

- ii. 受け取り側項目が 2 進数の場合は、ALL“定数”がいったん整数項目(単精度の場合は 5

桁、倍精度の場合は 10 桁)に転記され、それが 2 進数に変換されたものが転記される。

N^R , N^{RB} , N^{RP} , NE^R

送り出し側項目の内容が逆編集により編集されない数値に変換され、その内容がそれぞれ N , N^B , N^P , NE の規則に従って転記される。

NE (数字編集転記)

送り出し側項目の内容が、受け取り側項目の編集指定に従って編集されて転記される。編集方法については、『COBOL 言語説明書』「9. 14. 30 PICTURE 句」を参照。

F (表意定数転記)

表意定数で指定された値が、受け取り側項目の桁数に等しくなるまで左詰めで繰り返しとられ、それが転記される。

表意定数 **SPACE** が日本語項目、日本語編集項目に転記されたときは、日本語文字の空白が受け取り側項目の桁数に等しくなるまで繰り返し転記される。

送り出し側項目が **ALL** ブール定数で、受け取り側項目が英数字項目の場合は、ブール文字の“0”が外部 10 進形式の文字“0”に、またブール文字の“1”が外部 10 進形式の文字“1”に変換され、その値が受け取り側項目の桁数に等しくなるまで左詰めで繰り返し取られ、それが転記される。

【Standard】表意定数 **ZERO** が日本語項目、日本語編集項目に転記されたときは、日本語文字のゼロが受け取り側項目の桁数に等しくなるまで繰り返し転記される。

【Standard】表意定数 **HIGH-VALUE**, **LOW-VALUE** が日本語項目、日本語編集項目に転記されたときは、日本語文字の空白が受け取り側項目の桁数に等しくなるまで繰り返し転記される。

【Standard】表意定数 **QUOTE** が日本語項目、日本語編集項目に転記されたときは、日本語文字の二重引用符が受け取り側項目の桁数に等しくなるまで繰り返し転記される。

【Standard】表意定数の記号文字が日本語項目、日本語編集項目に転記されたときは、日本語文字に変換され受け取り側項目の桁数に等しくなるまで繰り返し転記される。日本語変換できない場合は、16 進文字の 0 が受け取り側項目の桁数に等しくなるまで繰り返し転記される。

G (日本語転記)

(a) 送り出し側項目が受け取り側項目より大きい場合

送り出し側項目のデータは左側から 1 文字ずつ、左詰めで受け取り側項目に送られ、余分な桁は切り捨てられる。

(b) 送り出し側項目が受け取り側項目より小さい場合

送り出し側項目の右側に、受け取り側項目と等しくなるだけ日本語文字の空白があるものとして、受け取り側項目に転記される。

(c) 項目の桁数が等しい場合

送り出し側のデータがそのまま受け取り側項目に転記される。

- (d) 受け取り側項目に **JUSTIFIED** 句の指定がある場合
データは右詰めで転記される。

G^C, **G^{AC}** (日本語転記:コード変換)

送り出し側項目のデータが日本語に変換され、その値が受け取り側項目に左詰めで転記される。

送り出し側項目が符号付きの数字項目のとき、その絶対値が日本語に変換され転記される。
ほかには、日本語転記と同じである。

G^{UAC} (日本語転記:内部 10 進→外部 10 進変換)

内部 10 進数が外部 10 進数に変換されるほかには、日本語転記(コード変換)と同じである。

G^{DAC} (日本語転記:2 進→外部 10 進変換)

2 進数が外部 10 進数に変換されるほかには、日本語転記(コード変換)と同じである。

G^{CF} (日本語転記:表意定数)

表意定数で指定された値が日本語に変換され、受け取り側項目の桁数に等しくなるまで左詰めで繰り返しのられ、それが転記される。

GE (日本語編集転記)

送り出し側項目の内容が、受け取り側項目の編集指定に従って左詰めで転記される。

GE^C, **GE^{AC}**, **GE^{UAC}**, **GE^{DAC}**, **GE^{CF}**, **GE^F**

送り出し側項目が、受け取り側の有効桁と同じ桁数の日本語項目にそれぞれ、**G^C**, **G^{AC}**, **G^{UAC}**, **G^{DAC}**, **G^{CF}**, **F** の規則に従って転記され、それが日本語編集項目に転記されたかのように実行される

【Enterprise】 -CU オプションを指定することにより、

備考 UTF-8 1 バイト文字から UCS-2(全角)へコード変換が行なわれる。

B0(ブール転記)

- (a) 送り出し側項目の桁数が受け取り側項目の桁数より大きい場合
送り出し側項目のブール文字が左端から 1 文字ずつ、左詰めで受け取り側項目に送られ、余分な文字は切り捨てられる。
- (b) 送り出し側項目の桁数が受け取り側項目の桁数より小さい場合
送り出し側項目の右側に、受け取り側項目の桁数と等しくなるだけブール文字の“0”があるものとして、受け取り側項目に転記される。
- (c) 項目の桁数が等しい場合

送り出し側のデータがそのまま受け取り側項目に転記される。

- (d) 受け取り側項目に **JUSTIFIED** 句の指定がある場合
データは右詰めで転記される。

B0¹ (ブール転記:0 と 1 の文字→ブール文字)

英数字項目の左端から文字が“0”のとき、ブール文字“0”に、また文字が“1”のとき、ブール文字“1”に変換され、受け取り側項目に左詰めで転記される。他はブール転記と同じである。0, 1 以外の文字は、その文字の最右端のビットが転記される。

B0^F (ブール転記:表意定数)

- (a) **ZERO** の場合は、ブール文字の“0”が受け取り側項目の桁数が等しくなるまで左詰めで繰り返し取られ、それが転記される。
- (b) 表意定数 **ALL** ブール定数の場合は、表意定数で指定された値が受け取り側項目の桁数が等しくなるまで左詰めで繰り返しとられ、それが転記される。

5.3.7 STRING 文

STRING 文は、複数個のデータ項目の内容の一部または全部をつなぎ合わせて 1 つのデータ項目に移す。

STRING 文は、次のような機能をもっている。

- (1) いくつかの項目の内容を 1 つのデータ項目にまとめる。このとき送り出し側のデータは、**STRING** 文で指定された順に受け取り側項目に移送される。

```
STRING a b c d DELIMITED BY SIZE INTO r.
```

a の内容:XYZX

b の内容:12

c の内容:AB

d の内容:ZZZ

r (実行前) の内容:*****

r (実行後) の内容:XYZX12ABZZZ****

上記の例でも明らかなように、受け取り側項目で実際にデータが送られてこない文字位置の内容はもとの内容が残っている。

- (2) ある送り出し側項目の内容を受け取り側項目に送る際に、その送り出し側項目の全部を送るかまたは一部を送るのかを指定することができる。これは **DELIMITED BY** 句で指定する。**DELIMITED BY**

句で **SIZE** を指定すると、送り出し側項目の全部の内容が送られる。定数または一意名を指定すると、送り出し側項目中で指定または一意名で指定された文字列が見つかる直前までの内容が送られる。指定された文字列が見つからない場合は全部の内容が送られる。**DELIMITED BY** 句は個々の送り出し側項目ごとに指定できる。

```
STRING a b DELIMITED BY "*" c d DELIMITED
      BY "ZZ" e f DELIMITED BY SIZE INTO r.
```

a の内容:123* 45

b の内容:*ZEZZGH

c の内容:XYZHZZ

d の内容:HIJKLMZ

e の内容:1234

f の内容:END

r (実行前) の内容:*****

r (実行後) の内容:123XYZHHIJKLMZ1234END**

上記例でも明らかなように、**DELIMITED BY** 句で指定された文字が送り出し側項目の左端にある場合、その送り出し側項目の内容はまったく移送されない。

- (3) 最初の送り出し側項目の内容を受け取り側項目に送るときに、受け取り側項目のどの位置からデータを格納するかを指定することができます。これは、**POINTER** 句で指定する。**POINTER** 句の指定がない場合、データは受け取り側項目の左端から格納される。**POINTER** 句は、格納開始位置を指定するだけでなく、**STRING** 文の実行によってデータがどこまで格納されたか(実際には最終格納データの次の文字位置)をプログラムに知らせる。すなわち、利用者は **POINTER** 句で指定した一意名の内容を調べることによってこれを知ることができる。

```
STRING a b c d DELIMITED BY "*" INTO r WITH POINTER p.
```

a の内容:AB*XY

b の内容:CDEF*

c の内容:*XYZ

d の内容:GHIJK

このとき、p の内容によって次のように実行される。

p (実行前) の内容:1

r (実行前) の内容:*****

p (実行後) の内容:12

r (実行後) の内容:ABCDEFGHIJK****

p (実行前)の内容:3
 r (実行前)の内容:*****
 p (実行後)の内容:14
 r (実行後)の内容:**ABCDEFGHJK**

(4) 受け取り側項目がオーバーフローしたときの処置を、**OVERFLOW** 句を使って指定することができる。受け取り側項目がオーバーフローしたとみなされるのは次の場合である。

(a) **POINTER** 句で指定した値が、1 より小さいかまたは受け取り側項目の桁数より大きい場合。

この場合、受け取り側項目の内容は変更されない。

(b) 受け取り側項目がいっぱいになったのに、まだ送り出し側項目が残っている場合。

この場合、受け取り側項目にはデータのオーバーフローが生じる直前に送られたデータまでが格納されている。また、このとき **POINTER** 句で指定されたデータ項目には、受け取り側項目の桁数に 1 を加えた値が格納される。

5.3.8 UNSTRING 文

UNSTRING 文は、送り出し側項目のひとつながりのデータを切り離して複数個の受け取り側データ項目に入れる。

UNSTRING 文は、次のような機能をもっている。

(1) 1 つのデータ項目の内容を複数個のデータ項目に分解する。このとき、送り出し側項目のデータは **UNSTRING** 文で指定された順に受け取り側項目に移送される。この移送は、送り出し側項目の内容が順次受け取り側項目で指定された桁数だけ移送される。**UNSTRING** 文では次のような移送が行われる。このとき、送り出し側項目は常に英数字項目とみなす。

(a) 受け取り側項目が数字項目のとき

受け取り側項目がゼロで満たされ、その後送り出し側項目の内容がそのまま右詰めで移送される。

123	→	9 (5)	結果 :	00123
12A	→	9 (5)	結果 :	0012A
A12	→	S9 (5)	結果 :	00A12
123+	→	9 (5)	結果 :	0123+
-123	→	S9 (5)	結果 :	0-123
12345	→	9 (3)	結果 :	345
ABCDE	→	S9 (3)	結果 :	CDE

したがって、**UNSTRING** 文で受け取り側項目に数字項目を指定した場合、直接算術演算などに使用するのは危険である。この場合、受け取り側項目の内容が数字か否かを検査してから使用すべきである。

- (b) 受け取り側項目が英字項目のとき
MOVE 文の転記の規則に従う。
- (c) 受け取り側項目が英数字項目のとき
MOVE 文の転記の規則に従う。

```
UNSTRING a INTO b c d e f g.
a : X(20)                ABCDE/*△123A123ABC*/

b : A(5)                 結果 ABCDE
c : X(3)                 結果 /*△
d : S9(4)                結果 123A
e : 9(3)                 結果 123
f : X(10) JUST RIGHT     結果 △△△△△ABC*/
g : A(3)                 結果 以前の内容が残る
```

上記の例でも明らかなように、送り出し側項目の移送がすべて終了したとき **UNSTRING** 文は終了し、余った受け取り側項目には以前の内容が残る(受け取り側項目 g)。

- (2) 1 つのデータ項目の内容を複数個のデータ項目に分解する際、受け取り側項目の桁数によって分解するのではなく、送り出し側項目内のある文字列を区切り記号として使用し、その区切り記号ごとに分解することができる。この区切り記号は **DELIMITED BY** 句で指定する。**DELIMITED BY** 句で定数または一意名によって区切り記号が指定されると、送り出し側項目の内容が、区切り記号が現れるまで調べられる(以降桁調べという)。この桁調べの結果見つけられた区切り記号の直前までの送り出し側項目の内容が、1 つの受け取り側項目に対する移送データとなり移送される。桁調べによって 1 つの受け取り側項目へのデータ移送が終了すると、次の桁調べが送り出し側項目内の区切り記号の直後の桁位置からはじめられる。この動作は送り出し側項目の終わりに達するか、受け取り側項目がなくなるまで(桁あふれ条件:以降で説明)繰り返される。桁調べによって区切り記号が見つからなかった場合、送り出し側項目の終わりまでが受け取り側項目に対する移送データとなり移送され、**UNSTRING** 文は終了する。

```
UNSTRING a DELIMITED BY "*" INTO b c d e f g.
a : X(20)    内容 ABCD*1234***1ZZZ1234

      ┌──┴──┐
      │  III  │
      └──┴──┘
      区 切 り 記 号 "*"

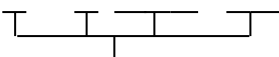
```

b : A(10) 結果 ABCD△△△△△
 c : 9(2) 結果 34
 d : X(5) 結果 △△△△△
 e : 9(3) 結果 000
 f : X(5) 結果 1ZZZ1
 g : A(3) 結果 以前の内容が残る

上記の例でも明らかなように送り出し側項目に区切り記号が連続して現れると、現在の受け取り側項目は、そのデータ記述に従って空白かゼロで満たされる(受け取り側項目 **d**, **e**)。また、区切り記号自身は移送されない。

- (3) 送り出し側項目の内容を区切り記号を指定して複数個の受け取り側項目に移送する際、送り出し側項目内に区切り記号が連続して現れた場合、これらの連続した区切り記号を 1 回の区切り記号の出現として処理することができる。これは、**DELIMITED BY ALL** 句を指定することによって行われる。

```
01 delim    PIC X(02) VALUE "*/".
UNSTRING a DELIMITED BY ALL delim INTO b c d e f.
```

a : X(20) 内容 */ABC*/**/*/*12*/*/
 
 区切り記号 ALL" */"

b : X(3) 結果 △△△
 c : A(3) 結果 ABC
 d : X(3) 結果 *△△
 e : 9(4) 結果 0012
 f : 9(3) 結果 以前の内容が残る

- (4) 送り出し側項目の内容を区切り記号で分解する際、1 つの送り出し側項目に対して、複数個の区切り記号を指定することができる。これは、**DELIMITED BY** 句に **OR** 句を使用して区切り記号を指定する。複数個の区切り記号が指定された場合、送り出し側項目内で最初に見つけられた区切り記号が、その時点の区切り記号となる。送り出し側項目の内容と区切り記号との比較は、送り出し側項目の左端または直前に見つけれられた区切り記号の次の桁から、**DELIMITED BY** 句に書かれた順番に比較される。

```
UNSTRING a DELIMITED BY ALL "△/" OR
SPACE OR "/△" INTO b c d e f g h.
```


a : X(20) 内容 AB△/CD△△/△/△///△△/△B

b : X(2) 結果 AB

c : A(2) 結果 CD

d : 9(2) 結果 00

e : X(3) 結果 /△△

f : A(2) 結果 △△

g : X(2) 結果 △△

h : A(3) 結果 B△△

(5) **DELIMITED BY** 句に区切り記号を指定して分解を行う場合、各受け取り側 項目に対し桁調べによってどのような区切り記号がとられたか、また何桁桁調べされたかを知ることができる。これは、**DELIMITER IN** 句および **COUNT IN** 句を指定することによって行われる。**DELIMITER IN** 句を指定すると桁調べの結果、とられた区切り記号がこの句で指定された一意名項目に、英数字基本項目間の **MOVE** の規則に従って移送される。このとき、見つけられた区切り記号に **ALL** 句の指定があると、区切り記号として移送されるのは送り出し側項目内の反復されている区切り記号全部ではなく、1 回分の区切り記号だけです。また、桁調べが送り出し側項目の終わりによって終了した場合、この一意名項目は空白で満たされる。

COUNT IN 句を指定した場合、桁調べされた桁数(区切り記号の桁数を除く)に等しい値が、**MOVE** の規則に従ってこの句に指定された一意名に移送される。

DELIMITER IN 句、**COUNT IN** 句は各受け取り側項目ごとに指定する。

UNSTRING a DELIMITED BY ALL "*" OR "DEL"

INTO

b **DELIMITER IN** delim-1 **COUNT IN** count-1

c **DELIMITER IN** delim-2 **COUNT IN** count-2

d **DELIMITER IN** delim-3 **COUNT IN** count-3

e **DELIMITER IN** delim-4 **COUNT IN** count-4

f **DELIMITER IN** delim-5 **COUNT IN** count-5

g **DELIMITER IN** delim-6 **COUNT IN** count-6

a : X(23) 内容 AB*CD***DEL123*****AB45

b : A(2) , 結果 AB	delim-1 : X(1) *	count-1 : 9 2
c : X , 結果 C	delim-2 : X(3) *△△	count-2 : 999 002
d : 9(3) , 結果 000	delim-3 : X(5) DEL△△	count-3 : S99 +00
e : 9(5) , 結果 00123	delim-4 : X(3) *△△	count-4 : 9 3
f : X(5) , 結果 AB45△	delim-5 : X(3) △△△	count-5 : 999 004
g : A(3) , 結果 以前の内容	delim-6 : 以前の内容	count-6 : 以前の内容

上記の例でも明らかなように、送り出し側項目に区切り記号が連続して現れると、**COUNT IN** 句の値はゼロとなる。送り出し側項目の分解が終了した後、残りの各受け取り側項目の **DELIMITER**, **COUNT IN** 項目の内容は以前の内容が残る。

- (6) ある送り出し側項目の内容を複数の受け取り側項目に分解し移送する際、送り出し側項目のどの位置から分解をはじめるかを指定することができる。これは **POINTER** 句で指定する。**POINTER** 句の指定がない場合、分解は送り出し側項目の左端から始められる。**POINTER** 句は送り出し側項目の分解開始位置を指定するだけでなく、**UNSTRING** 文の実行によって送り出し側項目の内容をどこまで処理したか(実際には最後に桁調べが行われた次の文字位置)をプログラムに知らせる。すなわち、利用者は **POINTER** 句で指定した一意名の内容を調べることによって送り出し側項目をどこまで処理したかを知ることができる。

UNSTRING a DELIMITED BY "*" INTO b c d

WITH POINTER p.

a : X(20) 内容 : ABCD*1234*GH*JK*LMNO

このとき p の内容によって次のように処理される。

p(実行前)の内容 : 12

b : X(4) 結果 H△△△

c : X(4) 結果 JK△△

d : X(4) 結果 LMNO

p(実行後)の内容 : 21

p(実行前)の内容 : 6

b : X(4) 結果 1234

c : X(4) 結果 GH△△

d : X(4) 結果 JK△△

p(実行後)の内容 : 17

- (7) 利用者は **UNSTRING** 文の実行によって送り出し側項目の内容が何個の受け取り側項目に分解されたかを知ることができる。これは **TALLYING IN** 句を使用することによって行われる。**TALLYING IN** 句を指定すると、**TALLYING IN** 句で指定された一意名の値に **UNSTRING** 文の実行の結果、作用を受けた受け取り側項目の個数が加算される。この結果、利用者は **UNSTRING** 文を複数回実行しても、現在実行した **UNSTRING** 文で作用を受けた受け取り側項目の個数、ならびに現在まで作用を受けた受け取り側項目の合計数を知ることができる。

UNSTRING a DELIMITED BY “*” INTO b c d

e f g TALLYING IN talin.

a: X(13) 内容 : AB*CD**123456



区切り記号 “*”

talin(実行前)の内容 : 5

b : A(2) 結果 AB

c : X(4) 結果 CD

d : X(3) 結果 △△△

e : 9(6) 結果 123456

f : 以前の内容

g : 以前の内容

talin(実行後)の内容 : 9

【Standard】 -Qj オプションを指定することにより、
TALLYING IN 句で指定された一意名が **PICTURE** 文字列に“P”を含む整数
 項目である場合、計数結果を正しく加算することができる。

- (8) 桁あふれ条件が起こったときの処置を **OVERFLOW** 句を使って指定することができる。**OVERFLOW** 句が指定されていると、その無条件文が実行される。**OVERFLOW** 句が指定されていなければ、制御は次の実行完結文に移る。桁あふれ条件が起こるのは次の場合である。

- (a) **UNSTRING** 文を開始したとき、**POINTER** 句の値が 1 より小さいかまたは送り出し側項目の桁数より大きいとき、**UNSTRING** 文は何も実行されない。
- (b) **UNSTRING** 文の実行中、すべての受け取り側項目が作用を受けたのに、送り出し側項目にまだ桁調べされない文字が残っているとき、桁あふれ条件が起こり、**UNSTRING** 文は終了する。

5.4 小入出力機能

ACCEPT 文および DISPLAY 文を使用することによって、形式の不定な小容量のデータを標準入力から入力または標準出力へ出力できる。さらに ACCEPT 文では、現在の時間または日付を入力できる。

5.4.1 ACCEPT 文

ACCEPT 文には次の 2 つの形式がある。

書き方 1

ACCEPT 一意名 1 [**FROM** 呼び名]

呼び名は特殊名段落で STDIN に関連付けられているものとする。

書き方 2

ACCEPT 一意名 2 **FROM** $\left\{ \begin{array}{l} \text{DATE} \\ \text{DAY} \\ \text{DAY-OF-WEEK} \\ \text{TIME} \end{array} \right\}$

書き方 1 の形式は標準入力からデータを入力する。入力されたデータは一意名 1 の項類に関係なく左詰めで一意名 1 に送られる。このとき符号の処理および必要なゼロ詰は行われない。標準入力がキーボードの場合には、Enter キーで入力は終了する。キーボードからの入力の途中に入力データの修正をする場合、Back space キーにて修正文字に位置づけた後、修正文字以降を再入力する。

データの長さが一意名 1 よりも小さいときは、残りの部分は一意名 1 の項類により英字の空白または日本語の空白で埋められる。データの長さが一意名 1 よりも大きいときは、一意名 1 分送られ、残りのデータは捨てられる。

小入力の文字コードに影響のある環境変数やオプションを指定しないとき、入力データはシフト JIS として扱われ、コード変換なしに一意名に送られる。

【Enterprise for Windows】-CU オプションを指定すると、標準入力に関連づけられているとき、入力データはシフト JIS として、また CONSOLE に関連づけられているとき、入力データは UTF-16LE として扱われ、一意名の属性に合わせてコード変換が行われた後、一意名に送られる。

【Enterprise for Linux】-CU オプションを指定すると、標準入力または CONSOLE に関連づけられているとき、入力データはシフト JIS として扱われ、一意名の属性に合わせてコード変換が行われた後、一意名に送られる。-CU または -Cj オプションを指定せず、環境変数 LANG に ja_JP.UTF-8、環境変数

COB_CODECHG に ON を指定すると、標準入力または CONSOLE に関連づけられているとき、入力データは UTF-8 として扱われ、一意名の属性が日本語項目または日本語編集項目のときはシフト JIS へコード変換が行われた後、一意名に送られる。一意名の属性がこれ以外のときはコード変換なしに一意名に送られる。

【Standard】-CF オプションが指定された対話型アプリケーションで、標準入力または CONSOLE に関連づけられているとき、一意名 1 が日本語項目または日本語編集項目のときは日本語データのみが、日本語項目でも日本語編集項目でもないときは英数字データのみが入力できる。日本語の入力データは UTF-16LE として扱われ、JIPS (J) へコード変換が行われた後、一意名に送られる。英数字の入力データはそのまま一意名に送られる。

【Standard】【Enterprise for Windows】-CF オプションが指定されたバッチ型アプリケーションで、標準入力または CONSOLE に関連づけられているとき、入力データはシフト JIS として扱われ、一意名の属性に合わせてコード変換が行われた後、一意名に送られる。

日本語項目への入力時、入力データに半角文字を含む場合（入力データのバイト数でチェック）、実行時エラーメッセージを出力後、異常終了する。ただし、UTF-8 ロケールの自動コード変換時は除く。

【 Enterprise 】 -Fj オプションを指定することにより、
ACOS-4 互換 SYSIN からデータを入力する。

書き方 2 の形式は日付（日付の年はシステム生成時の指定に従って西暦または年号の年を表す）または時間（24 時間制）を入力する。

DATE を指定したとき、システム生成時の指定とは無関係に常に yymmdd 形式の日付が入れられる。

DAY を指定したとき、yyddd 形式の日付が入れられる。

DAY-OF-WEEK を指定したとき、曜日を示す内容が入れられる。1 は月曜を、2 は火曜を…7 は日曜を示す。

TIME を指定したとき、hhmmssstt 形式の時間が入れられる。

一意名 2 で示されたデータ項目には、MOVE 文の規則に従って転記される。

業務日付の取り込み

【Enterprise】業務日付は書き方 2 の ACCEPT 文により得られ、環境変数 COB_JOBDATE で業務日付が 5 桁、6 桁、8 桁のとき、受け取り側に合わせて取り込まれる。その他の桁数のとき、その内容（DATE 指定なら左端 6 桁、DAY 指定なら左端 5 桁）がそのまま取り込まれる。

- (1) 業務日付が 5 桁の「04209」のとき、「YYDDD」形式の DAY とみなし
 - (a) DATE (年月日) 指定では「YMMDD」形式の 2004 年 7 月 27 日に変換して「040727」を取り込む。
 - (b) DAY (年日) 指定ではそのまま取り込む。
- (2) 業務日付が 6 桁の「040727」のとき、「YMMDD」形式の DATE とみなし
 - (a) DATE (年月日) 指定ではそのまま取り込む。
 - (b) DAY (年日) 指定では「YYDDD」形式の 2004 年 209 日に変換して「04209」を取り込む。

(3) 業務日付が 8 桁の「20040727」のとき、「YYYYMMDD」形式の DATE とみなし

(a) DATE (年月日) 指定では「YYMMDD」形式の 2004 年 7 月 27 日に変換して「040727」を取り込む。

(b) DAY (年日) 指定では「YYDDD」形式の 2004 年 209 日に変換して「04209」を取り込む。

たとえば、プログラム A がジョブ X で 2004 年 7 月 27 日に実行された場合の DAY-AREA および DATE-AREA の内容は次のとおりである。

	DATE-AREA	DAY-AREA
最初のステップ	040321	04081
後のステップ	040727	04209

プログラム A

```

      :
WORKING-STORAGE SECTION.
77 DAY-AREA PIC 9(5).
77 DATE-AREA PIC 9(6).
      :
PROCEDURE DIVISION.
P.
    ACCEPT DATE-AREA FROM DATE.
    ACCEPT DAY-AREA FROM DAY.
    STOP RUN.

```

【Enterprise for Windows】ジョブ X (システム標準のバッチファイルの場合)

```

set COB_JOBDATE=040321
A
set COB_JOBDATE=
A

```

【Enterprise for Linux】ジョブ X (シェルフファイルの場合)

```

export COB_JOBDATE=040321
A
export COB_JOBDATE
A

```

【Standard】業務日付は書き方 2 の ACCEPT 文により得られ、環境変数 COB_JOBDATE で業務日付が指定されていると、その内容 (DATE 指定なら左端 6 桁, DAY 指定なら左端 5 桁) がそのまま取り込まれる。

たとえば、プログラム A がジョブ X で 2004 年 7 月 27 日に実行された場合の DAY-AREA および DATE-AREA の内容は次のとおりである。

	DATE-AREA	DAY-AREA
最初のステップ	040321	04032 注(1)
後のステップ	040727	04209

プログラム A

```

      :
WORKING-STORAGE SECTION.
77 DAY-AREA PIC 9(5).
77 DATE-AREA PIC 9(6).
      :
PROCEDURE DIVISION.
P.
    ACCEPT DATE-AREA FROM DATE.
    ACCEPT DAY-AREA FROM DAY.
    STOP RUN.

```

【Standard】ジョブ X (システム標準のバッチファイルの場合)

```

set COB_JOBDATE=040321
A
set COB_JOBDATE=
A

```

注意 (注 1) 環境変数 COB_JOBDATE で指定された内容の左端 5 桁が取られる。

5.4.2 DISPLAY 文

DISPLAY 文の形式は次のとおりである。

$$\text{DISPLAY} \left\{ \begin{array}{l} \text{一意名 1} \\ \text{定数 1} \end{array} \right\} \dots [\text{UPON 呼び名 1}] [\text{WITH NO ADVANCING}]$$

呼び名は特殊名段落で **STDOUT**, **STDERR** に関連付けられていなければならない。

DISPLAY 文は標準出力および標準エラー出力ヘデータを出力する文である。なお、DISPLAY 文の最後の一意名および定数の後には、改行文字(`\n`)を出力する。

小出力の文字コードに影響のある環境変数やオプションを指定しないとき、出力データはシフト JIS として扱われ、コード変換なしにデータを出力する。

【Enterprise】-CU オプションを指定すると、次の動作となる。

- ・【Enterprise for Windows】標準出力に関連付けられているとき、出力データはシフト JIS として、また **CONSOLE** に関連付けられているとき、出力データは **UTF-16LE** として扱われ、一意名または定数の属性に合わせてコード変換が行われた後、出力される。
- ・【Enterprise for Linux】標準出力または **CONSOLE** に関連付けられているとき、出力データはシフト JIS として扱われ、一意名または定数の属性に合わせてコード変換が行われた後、出力される。
- ・標準エラー出力は、常にシフト JIS にコード変換され出力される。
- ・コード変換に失敗した場合、実行時エラーメッセージを表示し、終了する。

環境変数 **COB_CODEERR** を指定すると、コード変換に失敗した場合の動作を変更することができる。詳細については「付録 D 環境変数」を参照。

【Standrd】-CF オプションが指定された対話型アプリケーションで、標準出力および **CONSOLE** に関連付けられているとき、出力データは **UTF-16LE** として一意名または定数の属性に合わせてコード変換が行われた後、出力される。

【Standrd】【Enterprise for Windows】-CF オプションが指定されたバッチ型アプリケーションで、標準出力および **CONSOLE** に関連付けられているとき、出力データはシフト JIS として一意名または定数の属性に合わせてコード変換が行われた後、出力される。

【Enterprise for Linux】-CU または -Cj オプションを指定せず、環境変数 **LANG** に **ja_JP.UTF-8**、環境変数 **COB_CODECHG** に **ON** を指定すると、標準出力、標準エラー出力または **CONSOLE** に関連づけられているとき、出力データは **UTF-8** として、日本語定数や日本語項目または日本語編集項目の一意名をシフト JIS からコード変換して出力する。コード変換に失敗した場合、実行時エラーメッセージを表示し、終了する。環境変数 **COB_CODEERR** を指定すると、コード変換に失敗した場合の動作を変更することができる。詳細については「付録 D 環境変数」を参照。

【Enterprise】-Fj オプションを指定することにより、

ACOS-4 互換 **SYSOUT** ヘデータを出力する。

5.4.3 対話型アプリケーションまたは COBOL WS エミュレータ配下での小入出力機能の注意事項

小入出力機能の **ACCEPT** 文, **DISPLAY** 文での日本語項目の扱いは, 以下のとおりである。

	COBOL WSエミュレータ配下でない		COBOL WSエミュレータ配下	
	バッチ型アプリケーション	対話型アプリケーション	バッチ型アプリケーション	対話型アプリケーション
日本語項目	○	×	×	×

【Standard】-CF オプション指定時の小入出力機能の **ACCEPT** 文, **DISPLAY** 文の扱いは以下のとおりである。

COBOL WS エミュレータ配下でない		COBOL WS エミュレータ配下	
バッチ型アプリケーション	対話型アプリケーション	バッチ型アプリケーション	対話型アプリケーション
○	○	×	×

5.5 特殊レジスタ EXCEPTION-CODE

特殊レジスタ **EXCEPTION-CODE** は, 暗に **COMPUTATIONAL-2** の数字項目として定義されており, 例外処理機能の開始後, 発生した **COBOL** 独自の規格の例外コードを利用者に通知する。例外処理節以外で参照しても不定となる。

設定される例外コードの内容については, 「付録 E 例外コード一覧」を参照。

5.6 特殊レジスタ EXCEPTION-LINE

特殊レジスタ **EXCEPTION-LINE** は, 暗に **COMPUTATIONAL-2** の数字項目として定義されており, 例外処理節の開始後, **COBOL** 独自の例外が発生した行番号情報を利用者に通知する。例外処理節以外で参照しても不定となる。

5.7 特殊レジスタ COMPLETION-CODE

特殊レジスタCOMPLETION-CODEは、暗にCOMPUTATIONAL-2の数字項目として定義されており、プログラム開始時に0が設定される。ユーザは、0から255までの任意の値に変更ができ、プログラム終了時にこの特殊レジスタに設定されている値がシステムに渡される。

なお、プログラムが異常終了した場合は、COBOLプログラムが常に次の値を設定する。

【Enterprise】 1

【Standard】 255

注意 【Enterprise】プログラムが異常終了した場合には常に 1 が設定されるため、ユーザが任意の値として 1 を返却した場合、上位のシェル等では異常終了なのかユーザ値なのかの判別がつかない。ユーザが設定する任意の値には1を使用しないことが好ましい。

【Standard】プログラムが異常終了した場合には常に 255 が設定されるため、ユーザが任意の値として 255 を返却した場合、上位のバッチファイル等では異常終了なのかユーザ値なのかの判別がつかない。ユーザが設定する任意の値には 255 を使用しないことが好ましい。

【Enterprise for Windows】【Standard】使用例

バッチファイルの場合での例を示す。

バッチファイル例

```

cmgcdtest.exe
IF %ERRORLEVEL% EQU 10 GOTO LV10
IF %ERRORLEVEL% EQU 20 GOTO LV20
IF %ERRORLEVEL% EQU 30 GOTO LV30
IF %ERRORLEVEL% EQU 1 GOTO LV1
GOTO LVOTHER

:LV1
ECHO "COMPLETION-CODE=1"
GOTO END

:LV10
ECHO "COMPLETION-CODE=10"
GOTO END

:LV20
ECHO "COMPLETION-CODE=20"
```

```

      GOTO END
:LV30
      ECHO "COMPLETION-CODE=30"
      GOTO END
:LVOTHER
      ECHO "COMPLETION-CODE OTHER=%ERRORLEVEL%"

:END

```

次のプログラムを翻訳し、`cmpcdtest`というファイル名の実行形式を生成する。

```

000010  IDENTIFICATION      DIVISION.
000020  PROGRAM-ID.          CMPCDTEST.
000030  ENVIRONMENT          DIVISION.
000040  CONFIGURATION          SECTION.
000050  SOURCE-COMPUTER.      Express5800.
000060  OBJECT-COMPUTER.     Express5800.
000070  DATA                DIVISION.
000080  WORKING-STORAGE       SECTION.
000090  77  CMP-CD           PIC 999.
000100*
000110  PROCEDURE            DIVISION.
000120  MAIN-PROC.
000130      DISPLAY "INPUT COMPLETION-CODE".
000140      ACCEPT CMP-CD.
000150      MOVE CMP-CD TO COMPLETION-CODE.
000160      STOP RUN.

```

バッチファイルを起動すると、COBOLで生成したアプリケーションプログラムが起動される。画面から入力された値が、バッチファイルではERRORLEVELで参照できる戻り値として返却される。

【Enterprise for Linux】使用例

シェルフファイルの場合での例を示す。

シェルフファイル例

```
#!/bin/bash
```

```
cmpcdtest
```

```

ret=$?
if [ $ret = 10 ]
then
    echo "COMPLETION-CODE=10"
elif [ $ret = 20 ]
then
    echo "COMPLETION-CODE=20"
elif [ $ret = 30 ]
then
    echo "COMPLETION-CODE=30"
elif [ $ret = 1 ]
then
    echo "COMPLETION-CODE=1"
else
    echo "COMPLETION-CODE OTHER=$ret"
fi

```

次のプログラムを翻訳し, `cmpcdtest` というファイル名の実行形式を生成する。

```

000010  IDENTIFICATION      DIVISION.
000020  PROGRAM-ID.          CMPCDTEST.
000030  ENVIRONMENT         DIVISION.
000040  CONFIGURATION         SECTION.
000050  SOURCE-COMPUTER.     Express5800.
000060  OBJECT-COMPUTER.    Express5800.
000070  DATA              DIVISION.
000080  WORKING-STORAGE      SECTION.
000090  77  CMP-CD          PIC 999.
000100*
000110  PROCEDURE           DIVISION.
000120  MAIN-PROC.
000130      DISPLAY "INPUT COMPLETION-CODE".
000140      ACCEPT CMP-CD.
000150      MOVE CMP-CD TO COMPLETION-CODE.
000160      STOP RUN.

```

シェルフファイルを起動すると, COBOLで生成したアプリケーションプログラムが起動される。画面から入力された値が, シェルフファイルでは\$?で参照できる戻り値として返却される。

5.8 特殊レジスタ FUNCTION-KEY

【Enterprise】特殊レジスタ FUNCTION-KEY は未サポート。

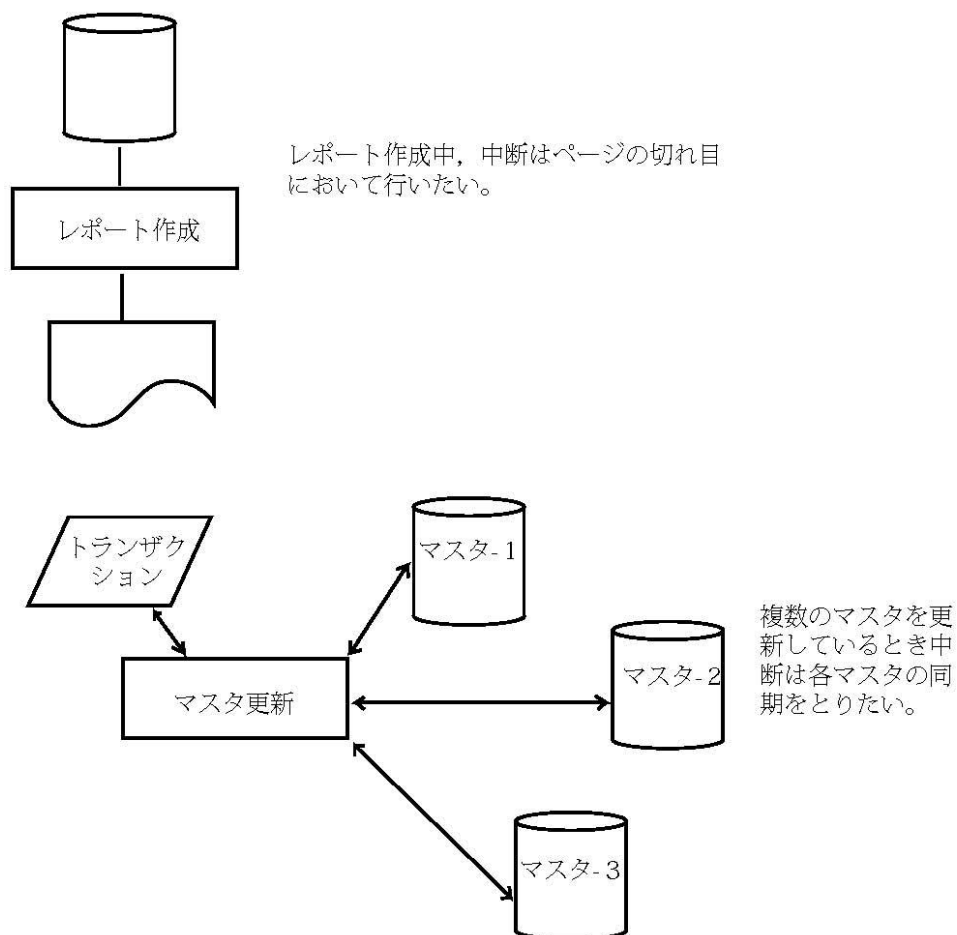
【Standard】特殊レジスタ FUNCTION-KEY は、暗に用途が表示用の 1 桁の英数字項目として定義されており、画面から入力された最後の終了キー、プログラムファンクションキーまたは画面入出力文の終了状態を利用者に通知する。バッチ型アプリケーションでは特殊レジスタ FUNCTION-KEY は未サポート。

表 5-4 FUNCTION-KEY に設定される値

FUNCTION-KEY の値 (16 進)	対応するキー	備考
42	[F1]	プログラムファンクションキー
43	[F2]	
44	[F3]	
45	[F4]	
46	[F5]	
47	[F6]	
48	[F7]	
49	[F8]	
4A	[F9]	
4B	[Shift]+[F10]	
4C	[Shift]+[F1]	
4D	[Shift]+[F2]	
88	[Shift]+[F3]	
89	[Shift]+[F4]	
8A	[Shift]+[F5]	
FF	どのキーも押されていない	初期値

FUNCTION-KEY の使用方法は、次のとおりである。バッチプログラムなどで処理を途中で中断したい場合、強制終了では、次のようなことが考えられる。

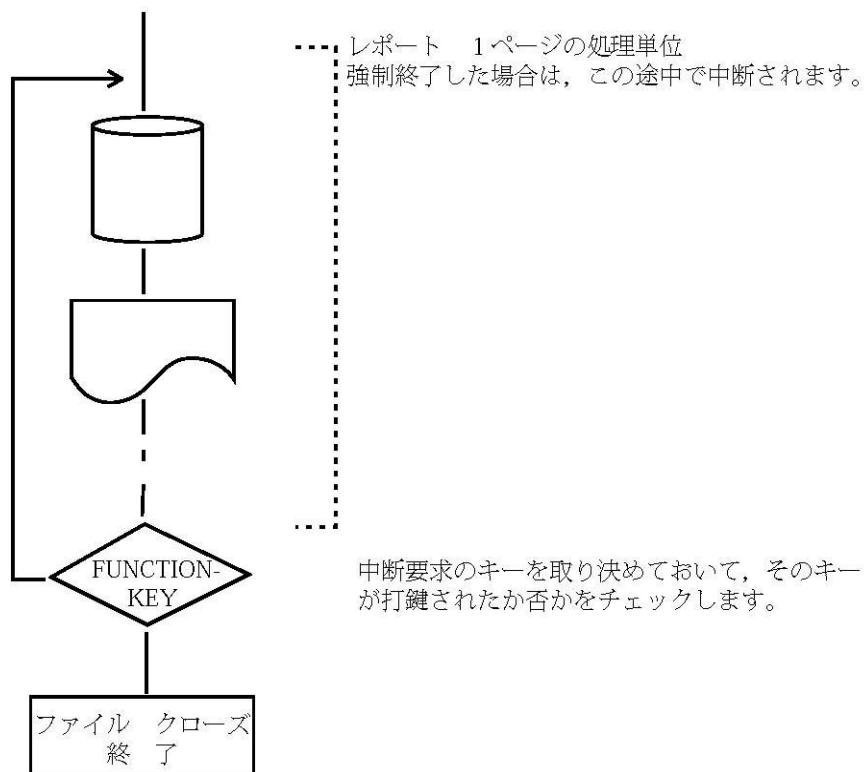
[例]



なお、中断した後のプログラム続行は、すべての利用者を考慮した上でプログラミングしておかなければならない。

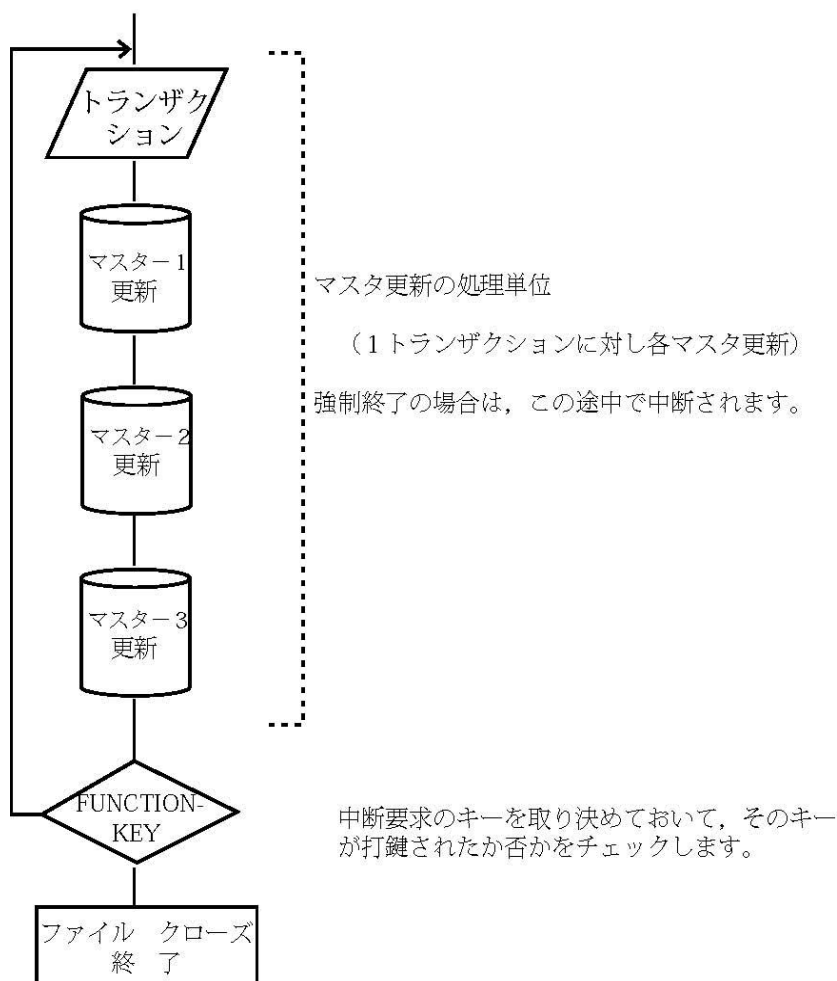
[例]

レポート作成中断の場合



[例]

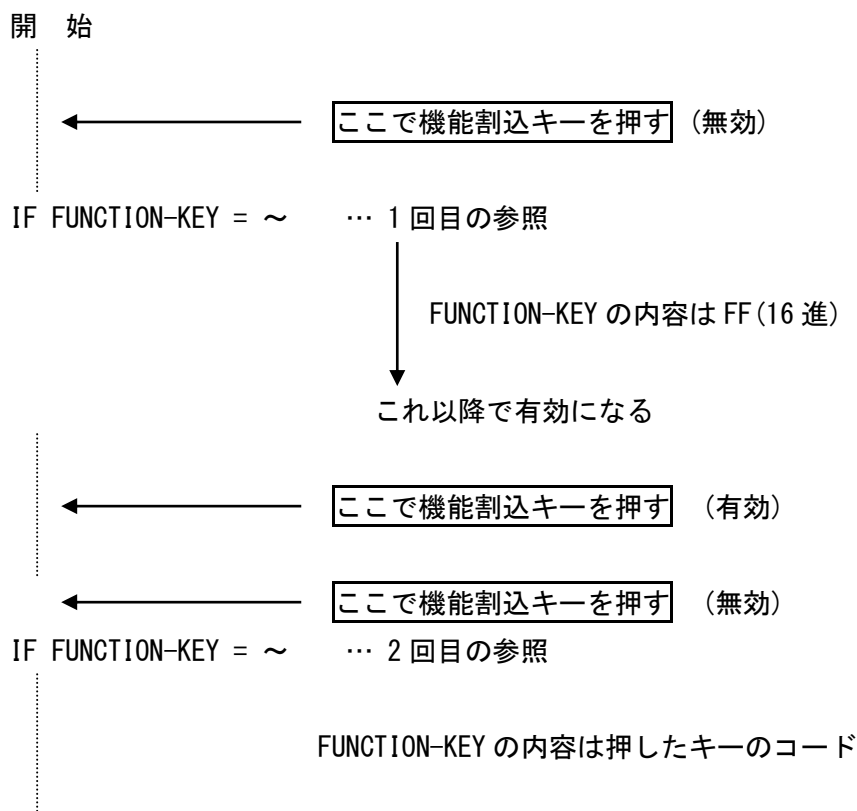
マスタ更新の場合



使用上の注意事項

1. **FUNCTION-KEY** を参照すると、その値は FF (16 進) に設定しなおされる。
2. アプリケーションが複数動作している場合は、ウィンドウがアクティブのときだけ本機能を使用できる。
3. **FUNCTION-KEY** を初めて参照したときは、FF (16 進) が設定され、これ以降、表 5-4 のプログラムファンクションキーを押すと、**FUNCTION-KEY** でそのコードを調べることができる。
すなわち、1 回目の参照後に本機能が有効になる。
4. プログラムファンクションキーを押して **FUNCTION-KEY** に値が設定された場合、**FUNCTION-KEY** を参照するまでその値は有効である。すなわち、いったん **FUNCTION-KEY** に値が設定されてから **FUNCTION-KEY** を参照するまでに押されたプログラムファンクションキーは無効となる。
5. **ACCEPT** 文の実行時に表 5-4 のプログラムファンクションキーを押した場合は、通常の入力終了キーとなる。

注意事項 3 をプログラムの流れと対応させて説明すると、次のようになる。



5.9 特殊レジスタ RECORD-LENGTH

【Enterprise】特殊レジスタ RECORD-LENGTH は未サポート。

【Standard】特殊レジスタ RECORD-LENGTH は、暗に COMP-2 項目として定義されており、行順ファイルに対する最後の READ 文が入力した文字位置の個数を利用者に通知する。READ 文以外では使用しない。

5.10 特殊レジスタ RDB-STATUS

【Enterprise】特殊レジスタ RDB-STATUS は未サポート。

【Standard】特殊レジスタ RDB-STATUS は、暗に用途が表示用の 2 桁の英数字項目として定義されており、COMMIT 文および ROLLBACK 文の実行結果を利用者に通知する。

5.11 特殊レジスタ RETURN-CODE

【Enterprise】特殊レジスタ RETURN-CODE は未サポート。

【Standard】特殊レジスタ RETURN-CODE は、暗に COMPUTATIONAL-1 の数字項目として定義されており、プログラム返却値を利用者に通知する。

5.12 外部 SWITCH 機能

COBOL アプリケーションで外部スイッチを使用する場合、コマンドラインツールを使用して SWITCH 状態を保持するためのファイルを作成し、その名前を環境変数 COB_SWITCH_FILE に設定する必要がある。

5.12.1 コマンドラインツール

プロセスツリーの最上位プロセスであるバッチファイルまたはシェルフファイルにおいてファイルの作成・削除を行う機能、任意のバッチファイルまたはシェルフファイルにおいて SWITCH 状態を設定・参照する機能を次のコマンドラインツールとして提供する。

機能	コマンド名
SWITCH 状態保持ファイルの作成	cobinitcv
SWITCH 状態の設定	cobsetcv
SWITCH 状態の参照	cobreadcv
SWITCH 状態保持ファイルの削除	cobtermcv

5.12.1.1 cobinitcv

機能

cobinitcv は、SWITCH 状態保持ファイルを作成する。

構文

cobinitcv [ディレクトリ名]

返却値

正常時は、作成した SWITCH 状態保持ファイルに対するフルパスを標準出力に出力する。異常を検出しファイルが作成できない場合は、空文字列を標準出力に出力する。

説明

一意な SWITCH 状態保持ファイルを作成し、作成したファイルのフルパスを標準出力に出力する。

引数を指定した場合、引数はディレクトリ名と解釈する。

次の順で、SWITCH 状態保持ファイルの作成を試みる。

- 【Enterprise for Windows】【Standard】環境変数 TMP
- 【Enterprise for Linux】環境変数 TMPDIR
- 引数に指定したディレクトリ
- . (カレントディレクトリ)

異常を検出しファイルが作成できない場合は、空文字列を標準出力に出力する。

cobinitcv は、環境変数には値を設定しない。環境変数への設定は、バッチファイルの **set** コマンド、またはシェルファイルの **export** コマンドと組み合わせて実現する。

5.12.1.2 cobsetcv

機能

cobsetcv は、SWITCH 状態を設定する。

構文

```
cobsetcv -s {switch=BBBBBBBB|switchN=B}
```

B:0 または 1

N:1, 2, ..., 8

返却値

正常時は、引数で指定されたBBBBBBBBまたはBを標準出力に出力する。異常時は空文字列を標準出力に出力する。

説明

環境変数 COB_SWITCH_FILE で指定されたファイルを SWITCH 状態保持ファイルとみなし、このファイルの内容によって SWITCH 状態を変更する。

引数に switch=BBBBBBBB の形式で指定した場合、SWITCH 状態全体をBBBBBBBBで指定された値に変更する。この場合、BBBBBBBB は左から SWITCH1, SWITCH2, ..., SWITCH8 の状態に相当する。BBBBBBBB が 8 桁に満たない場合、左側に 0 を補い動作する。

引数に switchN=B の形式で指定した場合、指定された SWITCH の状態を B で指定された状態に変更する。

環境変数 COB_SWITCH_FILE が定義されていない場合や環境変数 COB_SWITCH_FILE で指定したファイルが存在しない場合は異常終了する。

5.12.1.3 cobreadcv

機能

cobreadcv は、SWITCH 状態を参照する。

構文

```
cobreadcv -s { switch|switchN }  
N:1, 2, ..., 8
```

返却値

正常時は、引数で指定された SWITCH 状態の値を標準出力に出力する。異常時は、空文字列を標準出力に出力する。

説明

指定された SWITCH 全体の状態または特定の SWITCH の状態を取得し、結果を標準出力に出力する。取得に失敗した場合には、空文字列を標準出力に出力する。

5.12.1.4 cobtermcv

機能

cobtermcv は、SWITCH 状態保持ファイルを削除する。

構文

```
cobtermcv
```

返却値

正常時は、空文字列を標準出力に出力する。異常時は、ファイルが削除できなかった場合はファイルのフルパスを標準出力に出力する。これ以外の場合は、-(半角ハイフン 1 文字)を標準出力に出力する。

他のコマンドラインツールの返却値と異なるため、注意されたい。

説明

環境変数 COB_SWITCH_FILE で指定されたファイルを削除する。

cobtermcv は、環境変数には値を設定しない。環境変数への設定は、バッチファイルの set コマンド、またはシェルファイルの export コマンドと組み合わせて実現する。

5.12.2 使用例

【Enterprise for Windows】【Standard】バッチファイルにおける外部スイッチ機能の使用例を示す。

```
FOR /F %%w IN ('cobinitcv') DO SET COB_SWITCH_FILE=%%w
cobsetcv -s switch=01001110
COBOLAP1
FOR /F %%w IN ('cobreadcv -s switch7') DO SET SW7=%%w
IF %SW7% EQU 1 (COBOLAP2) ELSE (COBOLAP3)
cobtermcv
IF %ERRORLEVEL% EQU 0 (SET COB_SWITCH_FILE=)
```

【Enterprise for Linux】シェルフファイルにおける外部スイッチ機能の使用例を示す。

```
w=`cobinitcv`
export COB_SWITCH_FILE=$w
cobsetcv -s switch=01001110
COBOLAP1
w=`cobreadcv -s switch7`
export SW7=$w
if [ $SW7 = 1 ]
then
    COBOLAP2
else
    COBOLAP3
fi
cobtermcv
if [ $? = 0 ]
then
    export COB_SWITCH_FILE=
fi
```

5.12.3 COBOL アプリケーション実行環境下の使用

【Enterprise】外部スイッチ状態の参照／設定は未サポート。

COBOL アプリケーション実行環境 ジョブ管理機能と組み合わせて使用することにより, A-VX 互換の外部スイッチ状態の参照／設定を行うことができる。

[プログラム例]

```

000010 IDENTIFICATION      DIVISION.
000020 PROGRAM-ID.          SWITCH01.
000030 ENVIRONMENT          DIVISION.
000040 CONFIGURATION          SECTION.
000050 SOURCE-COMPUTER.       Express5800.
000060 OBJECT-COMPUTER.     Express5800.
000070 SPECIAL-NAMES.
000080     SWITCH-1 IS SW1
000090         ON IS SW1ON
000100         OFF IS SW1OFF
000110     SWITCH-2
000120         ON STATUS IS SW2ON.
000130 PROCEDURE              DIVISION.
000140 SWEXAMPLE.
000150     IF SW2ON
000160         THEN SET SW1 TO ON
000170         ELSE SET SW1 TO OFF
000180     END-IF.
000190     STOP RUN.

```

このプログラムから生成した COBOL アプリケーションを SWITCH01.EXE とする。

COBOL アプリケーション実行環境 ジョブ管理機能と組み合わせて使用する場合の例を以下に記述する。

指定方法については, COBOL アプリケーション実行環境 ジョブ管理機能の説明書を参照。

バッチファイル内でのスイッチ操作は次のコマンドで行うことができる。

5.12.3.1 外部スイッチの設定(SETSSW)

機能

外部スイッチの ON/OFF を設定する。

書式

SETSSW SSWn = ON または OFF

または

SETSSW ALL = S₁ S₂ S₃ S₄ S₅ S₆ S₇ S₈

パラメータ

SSWn	n は更新する外部スイッチ番号を示す。 1 から 8 までの整数で指定する。
ON	外部スイッチを ON にする
OFF	外部スイッチを OFF にする
ALL	SSW1～SSW8 までのすべての外部スイッチ状態を一括変更する場合 に, 本パラメータを指定する。
S ₁ ～S ₈	SSW1～SSW8 までの外部スイッチを 1 または 0 で指定する。 1: 外部スイッチを ON にする 0: 外部スイッチを OFF にする

実行結果

本モジュールの実行結果は, ERRORLEVEL に反映される。

- 1: 外部スイッチの設定に成功した
- 255: 外部スイッチの設定に失敗した

5.12.3.2 外部スイッチの設定(CHKSSW)

機能

外部スイッチを参照する。

書式

CHKSSW SSWn = ON または OFF

または

CHKSSW ALL = S₁ S₂ S₃ S₄ S₅ S₆ S₇ S₈

パラメータ

SSWn	n は更新する外部スイッチ番号を示す。 1 から 8 までの整数で指定する。
ON	外部スイッチが ON であるか比較する
OFF	外部スイッチが OFF であるか比較する
ALL	SSW1～SSW8 までのすべての外部スイッチを一括比較する場合に、本パラメータを指定する。
S ₁ ～S ₈	SSW1～SSW8 までの外部スイッチ状態を 1 または 0 で指定する。 1: 外部スイッチが ON であるか比較する 0: 外部スイッチを OFF であるか比較する

実行結果

本モジュールの実行結果は、ERRORLEVEL に反映される。

- 1: 外部スイッチの比較結果は「真」である
- 0: 外部スイッチの比較結果は「偽」である
- 255: 外部スイッチの参照に失敗した

[バッチファイルの例 1]

バッチファイルは、コマンドプロンプトから APECMD. EXE によって起動する。

C:¥> APECMD SSW. BAT

SSW. BAT の内容	バッチファイルの説明
SETSSW SSW2 = ON	SWITCH-2 に ON を設定します。
APECMD SWITCH01. EXE	SWITCH01. EXE を起動します。 APECMD START /W SWITCH01. EXE SWITCH01. EXE では、SWITCH-2 が ON の場合、SWITCH-1 を ON に設定しますので、ここでは SWITCH-1 が ON の状態でアプリケーションが終了します。
CHKSSW SSW1 = ON	SWITCH-1 が ON に設定されているかを比較します。
IF ERRORLEVEL 1 GOTO TRUE ECHO FALSE GOTO END :TRUE ECHO TRUE :END	ERRORLEVEL が 1 であれば、TRUE へ制御が移ります。 FALSE を表示します。 TRUE を表示します。

[バッチファイルの例 2]

COBOL WS エミュレータからバッチファイルを起動する場合は、コマンドプロンプトから NTCMD. EXE によって

起動する。

C:¥> NTCMD SSW.BAT

SSW.BAT の内容	バッチファイルの説明
SETSSW SSW2 = ON	SWITCH-2 に ON を設定します。
EMLCMD SWITCH01.EXE	SWITCH01.EXE を起動します。 SWITCH01.EXE では、SWITCH-2 が ON の場合、SWITCH-1 を ON に設定しますので、ここでは SWITCH-1 が ON の状態でアプリケーションが終了します。
CHKSSW SSW1 = ON	SWITCH-1 が ON に設定されているかを比較します。
IF ERRORLEVEL 1 GOTO TRUE ECHO FALSE GOTO END :TRUE ECHO TRUE :END	ERRORLEVEL が 1 であれば、TRUE へ制御が移ります。 FALSE を表示します。 TRUE を表示します。

第6章 表操作機能

6.1 概 説

事務処理では、表になったデータを使うことが多い。表になったデータは、そのデータを何回も繰り返し書くことによって記述できるが、これは次の理由から不十分である。

- (1) 文書化の観点からいえば、データが表になっていることがはっきりしない。
- (2) 表中のどの要素を取り出すかが実行時に定まる場合、指定されたデータを取り出すことは困難である。

このことから、COBOL においては表を定義する機能や表中の特定の要素を参照する機能が用意されている。表を定義するために OCCURS(反復)句がある。OCCURS 句を用いることによって、指定した項目を指定した回数だけ繰り返してとることができる。

また、OCCURS 句で定義した表の要素を参照するためには、表の名前と表内で何番目の要素であるかを指定する。この指定方法を添字付け(subscripting)という。

6.2 表の定義

表を定義するためには OCCURS 句を用いる。本 COBOL では、1 次元から 7 次元までの表を定義することができる。その例を次に示す。

〈1 次元の表の定義例〉

```
01 TABLE1.  
  02 TABLE-ELEMENTS OCCURS 20 TIMES.  
    03 DOG PIC X(2).  
    03 FOX PIC 9.
```

これによって、表は次のように生成される。

TABLE1							
t ₁		t ₂		t ₃		t ₄	
d ₁	f ₁	d ₂	f ₂	d ₃	f ₃	d ₄	f ₄

t : TABLE-ELEMENTS d : DOG f:FOX

〈2 次元の表の定義例〉

```
01 TABLE1.  
  02 TABLE-ELEMENTS OCCURS 20 TIMES.  
    03 DOG OCCURS 2 TIMES.  
      05 EASY PIC X(2).  
      05 FOX  PIC 9.
```

これによって、表は次のように生成される。

TABLE1							
t ₁				t ₂			
d _{1,1}		d _{1,2}		d _{2,1}		d _{2,2}	
e _{1,1}	f _{1,1}	e _{1,2}	f _{1,2}	e _{2,1}	f _{2,1}	e _{2,2}	f _{2,2}

〈 3 次元の表の定義例〉

```
01 TAB.  
  02 TABLE1 OCCURS 2 TIMES.  
    03 TABLE-ELEMENTS OCCURS 3 TIMES.  
      04 DOG OCCURS 2 TIMES.  
        05 EASY PIC X(2).  
        05 FOX  PIC 9.
```

これによって、表は次のように生成される。

TAB											
T1											
t _{1,1}				t _{1,2}				t _{1,3}			
d _{1,1,1}		d _{1,1,2}		d _{1,2,1}		d _{1,2,2}		d _{1,3,1}		d _{1,3,2}	
e _{1,1,1}	f _{1,1,1}	e _{1,1,2}	f _{1,1,2}	e _{1,2,1}	f _{1,2,1}	e _{1,2,2}	f _{1,2,2}	e _{1,3,1}	f _{1,3,1}	e _{1,3,2}	f _{1,3,2}

T2					
t _{2,2}			t _{2,3}		
d _{2,2,3}			d _{2,3,1}		d _{2,3,2}
e _{2,2,3}	f _{2,2,3}		e _{2,3,1}	f _{2,3,1}	e _{2,3,2} f _{2,3,2}

T:TABLE t:TABLE-ELEMENTS d:DOG
e:EASY f:FOX

〈可変繰返し項目の定義例〉

```
01 TAB-SIZE PIC 9(2).  
01 TAB.  
    02 TABLEI OCCURS 2 TO 10 TIMES DEPENDING ON TAB-SIZE.  
        03 DOG PIC X(2).  
        03 EASY PIC 9(2).
```

これによって、表は次のように定義される。

TAB-SIZE の値が 3 のとき

TAB											
t ₁		t ₂		t ₃		t ₄		t ₉		t ₁₀	
d ₁	e ₁	d ₂	e ₂	d ₃	e ₃	d ₄	e ₄	d ₉	e ₉	d ₁₀	e ₁₀

この範囲が参照できる。 t:TABLEI d:DOG e:EASY

TAB-SIZE の値は、2 以上 10 次の範囲内で自由に変えることができる。

6.3 表の初期値

COBOL の規則では、OCCURS 句を含む記述項およびそれに従属する記述項に VALUE 句を書くことができる。しかし、その場合、指定した VALUE の値が各要素に繰り返し割り当てられるので、各要素に異なる値の初期値を定義するためには、次のいずれかの方法をとる必要がある。

- (1) 集団項目レベルで VALUE 句を定義する方法。
- (2) 初期値がとってある項目を再定義して使用する方法。

次に、上記 2 つの方法の例を示す。

〈集団項目レベルで VALUE 句を定義する方法〉

```
01 TABLEI VALUE "1DOG2CAT3COW4ELE5HOR".  
02 ANIMAL-TBL OCCURS 5 TIMES.  
    03 ANIMAL-KEY    PIC X.  
    03 ANIMAL-NAME   PIC X(3).
```

〈REDEFINES 句を使用する方法〉

```
01 CONST PIC X(20) VALUE "1DOG2CAT3COW4ELE5HOR".  
01 TABLEI REDEFINES CONST.  
    02 ANIMAL-TBL OCCURS 5 TIMES.  
        03 ANIMAL-KEY    PIC X.  
        03 ANIMAL-NAME   PIC X(3).
```

6.4 表の参照

表の要素を参照するためには、指標名を使わない添字付けと指標名を使った添字付けの 2 つがある。これらは次に示すような特徴をもっている。

6.4.1 指標名を使わない添字付け

添字付けでは、表内の要素を参照するために、ある要素が表内で何番目にあるか(これを出現番号と呼ぶ)を表のデータ名の後にかっこで囲み、整数で指定する。たとえば、1 次元の表の定義例において特定の要素を参照したい場合には、次のように書く。

DOG(3) 表中の d_3 を指す。
TABLE-ELEMENTS(16) 表中の t_{16} を指す。

このときかっこ内の数字を添字という。添字は、整数であってもデータ名であってもよい。データ名の場合は、その内容が出現番号とみなされる。

1 次元の表では添字は 1 個でよいが、2 次元の表では 2 個、3 次元の表では 3 個必要である。すなわち、2 次元および 3 次元の表中の要素を見る場合は、次のように書く。

2 次元の表の定義例において

EASY(2, 1) 表中の $e_{2,1}$ を指す。
DOG (20, 2) 表中の $d_{20,2}$ を指す。

3 次元の表の定義例において

EASY(1, 2, 2) 表中の $e_{1,2,2}$ を指す。
FOX(2, 2, 2) 表中の $f_{2,2,2}$ を指す。

ここで明らかのように、添字は外側の大きな表から内側の表(大, 中, 小)の順番で書く。添字の書き方には、次のような種々の書き方がある。

EASY (X, Y, Z) X, Y, Z はデータ名である。
EASY (X, 1, Z)
EASY (1, 2, Z)

6.4.2 指標名を使った添字付け

表内の要素を参照するもう 1 つの方法に指標名を使った添字付けがある。この方法を用いるためには、表を OCCURS 句を使って定義するときに、その表に対して指標名(index name)を割り当てる。そして、その指標名を用いて特定の要素を参照する。指標名を定義するためには INDEXED BY 句を用いる。すなわち、次のように書く。

```
02 TABLE1 PIC X(2) OCCURS 10 TIMES
    INDEXED BY IX1.
```

そして、この表を参照するときには、INDEXED BY 句で指定した指標名を、データ名を使った添字付けの場合と同様の方法で表の名前に続けて書く。すなわち

```
TABLE1(IX1)
```

のように書く。

指標名を使った添字付けを用いて表を参照するときは、使用する指標名にあらかじめ参照したい要素の出現番号を格納しておく必要がある。これには SET 文を用いる。次に、SET 文を用いた場合の例を示す。

表の定義

```
02 TABL OCCURS 10 TIMES INDEXED BY X1.
03 CODE1 PIC 9(5).
03 CODE2 PIC X(5).
```

次の順番で文を実行するとすれば、

```
SET X1 TO 3.
MOVE CODE1(X1) TO Y. 3 番目の要素が移送される。
SET X1 UP BY 1.
MOVE CODE1(X1) TO Y. 4 番目の要素が移送される。
SET X1 DOWN BY 2.
MOVE CODE1(X1) TO Y. 2 番目の要素が移送される。
```

6.4.3 相対添字付け

添字付けにおいて、データ名、指標名の次に算術演算子(+, -)を書き、続けて整数を書くことによって、データ名、指標名で示される要素の前後の要素を参照することができる。これを相対添字付けとよぶ。次に、その例を示す。

指標名を使わない相対添字付けの例

```
MOVE 3 TO X1.
MOVE GRPCODE(X1) TO Y.           3 番目の要素が移送される。
MOVE GRPCODE(X1 + 3) TO Y. 6 番目の要素が移送される。
MOVE GRPCODE(X1 - 1) TO Y. 2 番目の要素が移送される。
```

指標名を使った相対添字付けの例

```
SET X1 TO 5.
MOVE GRPCODE(X1) TO Y.           5 番目の要素が移送される。
MOVE GRPCODE(X1 + 3) TO Y. 8 番目の要素が移送される。
MOVE GRPCODE(X1 - 1) TO Y. 4 番目の要素が移送される。
```

6.4.4 添字付けにおけるアドレス計算

COBOL 文の作用対象として、添字付けが使用されていると、COBOL コンパイラは、その文の実行に先立って添字付けで指定された要素のアドレスを計算するためのオブジェクトを自動的に生成する。このアドレス計算は、次のように行われる。

(1) 指標名を使わない添字付けにおけるアドレス計算

アドレス計算の方法を説明するために、次のような表を考える。

```
01 TABL.
  02 A OCCURS 10 TIMES INDEXED BY X1.
  03 B OCCURS 10 TIMES INDEXED BY Y1.
    04 C PIC X(5).
    04 D PIC X(10).
```

ここで、C(X, Y)のアドレス計算は次のように行われる。

$(C \text{ の第 } 1 \text{ 要素のアドレス} + [(X-1) \times (\text{表 A の要素の桁数})] + [(Y-1) \times (\text{表 B の要素の桁数})])$

たとえば、C の第 1 要素のアドレスを 100, X の値を 3, Y の値を 5 とすれば

$100 + [(3-1) \times 150] + [(5-1) \times 15]$

で計算される。

また、この計算は添字付けされている作用対象が参照されるたびに行われる。

(2) 指標名を使った添字付けにおけるアドレス計算

指標名を使った添字付けにおけるアドレス計算は、添字付けのたびに行われるのではなく、SET 文の実行時に行われる。その例を(1)で用いた表をもとにして、次に示す。

SET X1 TO X.

$[(X-1) \times \text{表 A の要素の桁数}]$ が計算されて指標名 X1 に格納される。

SET Y1 TO Y.

$[(Y-1) \times \text{表 B の要素の桁数}]$ が計算されて指標名 Y1 に格納される。

C(X1, Y1)

$(C \text{ の第 } 1 \text{ 要素のアドレス}) + (\text{指標名 X1 の値}) + (\text{指標名 Y1 の値})$ 。

説明で明らかなように、指標を使うか否かの大きな違いは、指標名を使わない添字付けは添字付けされる時点でアドレス計算が行われるが、指標名を使う場合は SET 文の実行時に行われ、添字付けされる時点では SET 文によって計算された値を使って計算するところにある。

(3) 定数による添字付けと指標の設定

添字に定数が使われたり、SET 文の作用対象に定数が使われている場合は、翻訳時に上述した計算が行われるため、データ名を使用するよりも早い実行速度が得られる。

たとえば、(1)に示した表において

C(3, 4)とすれば

$(C \text{ の第 } 1 \text{ 要素のアドレス}) + [(3-1) \times (\text{表 A の要素の桁数})] + (4-1) \times [(\text{表 B の要素の桁数})]$

SET X1 TO 3 とすれば

$(3-1) \times (\text{表 A の要素の桁数})$

SET X1 UP BY 4 とすれば

$[(\text{指標名 X1 の値}) + 4] \times \text{表 A の要素の桁数}$

のような計算が翻訳時に行われる。厳密に言えば、SET X1 UP BY 4 の文においては、 $[4 \times (\text{表 A の要素の桁数})]$ の計算だけが翻訳時に行われる。

6.4.5 指標名を使わない添字付けと指標を使った添字付けの効率

一般に表を参照するためには、指標を用いた方が効率がよい。次に両者のどちらが効率がよいかをいくつかの例を用いて説明する。詳細については、「6.4.4 添字付けにおけるアドレス計算」を参照。

(1) 表を順番に処理するような場合は、指標名の方が効率がよい。

指標名を使わない添字付け

LOOP. ADD 1 TO X.

IF KEY-CODE(X) = 103 毎回アドレス計算が行われる。

MOVE KEY-VALUE(X) TO Z 毎回アドレス計算が行われる。

ELSE GO TO LOOP.

指標を使った添字付け

LOOP. SET X1 UP BY 1. アドレス計算は翻訳時に行われる。

IF KEY-CODE(X1) = 103 指標名 X1 の値が用いられる。

MOVE KEY-VALUE(X1) TO Z

ELSE GO TO LOOP.

(2) 表の要素の出現番号を増減させる値が定数の場合は、指標名の方が効率がよい。

指標名を使わない添字付け

SUBTRACT 5 FROM X.

MOVE KEY-VALUE(X) TO Z. 毎回アドレス計算が行われる。

指標名を使った添字付け

SET X1 DOWN BY 5.	アドレス計算は翻訳時に行われる。
MOVE KEY-VALUE(X1) TO Z.	指標名 X1 の値が用いられる。

- (3) 表の要素の出現番号を増減させる値が変化(すなわちデータ名のとき)する場合は、指標名を使っても使わなくてもほとんど変わらない。

指標名を使わない添字付け

ADD Y TO X.	
MOVE KEY-VALUE(X) TO Z.	毎回アドレス計算が行われる。

指標名を使った添字付け

SET X1 UP BY X.	毎回アドレス計算が行われる。
MOVE KEY-VALUE(X1) TO Z.	指標名 X1 の値が用いられる。

- (4) 1 つのデータで複数の表の要素を参照するような場合は、指標名を使わない方がよい。

指標名を使わない添字付け

```
ADD Y TO X.  
MOVE A(X) TO B(X) C(X) D(X).  
SUBTRACT 1 FROM X.  
ADD A(X) TO B(X).
```

指標名を使った添字付け

```
SET X1 X2 X3 X4 UP BY Y.  
MOVE A(X1) TO B(X2) C(X3) D(X4)  
SET X1 X2 X3 X4 DOWN BY 1.  
ADD A(X1) TO B(X2).
```

すなわち、異なる表に同一の指標名を割り当てることはできないので、常に必要な数だけの指標名の値を更新しなければならない。

(5) 出現番号の値を頻繁に参照するような場合は、指標名を使わない方が効率がよい。

指標名を使わない添字付け

```
LOOP. ADD Y TO X.  
      IF A(X) = 103          毎回アドレス計算が行われる。  
      GO TO EXIT1.  
      IF X = M GO TO EXIT2  
      ELSE GO TO LOOP.
```

指標名を使った添字付け

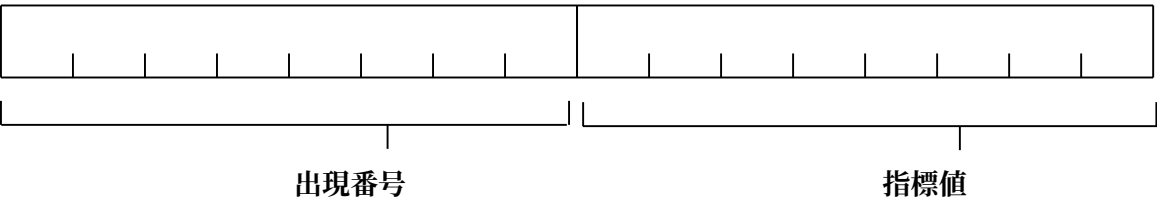
```
LOOP. SET X1 UP BY Y.        毎回アドレス計算が行われる。  
      IF A(X1) = 103  
      GO TO EXIT1.  
      IF X1 = M GO TO EXIT2   必要なデータ変換が行われ、  
      ELSE GO TO LOOP.        比較される。
```

6.5 指標名と指標データ項目

指標名の値を退避したり、復帰したりするための作業領域として、指標データ項目がある。指標データ項目の形式は指標名の形式と同じであり、退避や復帰のときはどの変換も行われずそのままの値が指標データ項目に格納される。

6.5.1 指標名と指標データ項目の内部表現形式

指標名と指標データ項目の内部表現形式



出現番号は COMP-5 形式の 8 バイト 2 進項目 (COMP-5 PIC S9(18)) で格納し、指標値は COMP-5 進形式の 8 バイト 2 進項目 (COMP-5 PIC S9(18)) で出現番号とその表の要素のバイト数を乗じた値が格納される。

-Qc オプションを指定した場合は、出現番号は COMP-2 形式の 4 バイト 2 進項目で格納し、指標値は COMP-2 形式の 4 バイト 2 進項目で出現番号－1 とその表の要素のバイト数を乗じた値が格納される。

6.6 SEARCH 文

SEARCH 文は、表の中から指定したデータを探し出す。この文を使用することによって、表中の希望するデータを複雑なアルゴリズムを考慮することなく、効率よく取り出すことができる。

SEARCH 文の表引きは、見つけられたデータがある表の要素の出現番号を、表に対応する指標名に設定することにより行われる。したがって、SEARCH 文の作用対象となる表の定義で、OCCURS 句は INDEXED BY 句を伴う必要がある。

SEARCH 文の表引きには、逐次表引きを行う SEARCH 文と 2 分表引きを行う SEARCH ALL 文とがある。

6.6.1 SEARCH(逐次表引き)文

SEARCH 文は、表が小さいかまたは表中のデータが不規則に並んでいる場合に有効である。

(1) SEARCH 文は、指定された表引き用指標名が指す要素から OCCURS 句で指定した反復回数の最大値 (DEPENDING 句があるときはそのデータ名 1 の値) が指す要素までを 1 項目ずつ順番に走査する。

表引き用指標名は次の 2 つの指定方法がある。

(a) VARYING 句の指標名を使用する場合

- VARYING 句に SEARCH 文の作用対象である表の指標名を指定する。

この場合、VARYING 句で指定した指標名が指す要素から OCCURS 句で指定した反復回数の最大値 (DEPENDING 句があるときはそのデータ名 1 の値) が指す要素までが表引きの範囲となる。

(b) SEARCH 文の作用対象である表の最初の指標名を使用する場合

- VARYING 句を省略する。
- VARYING 句で他の表の指標名を指定する。
- VARYING 句で指標データ項目または整数基本項目を指定する。

この場合、INDEXED BY 句で与えられた最初の指標名が指す表の要素から OCCURS 句で指定した反復回数の最大値(DEPENDENT 句があるときはそのデータ名 1 の値)が指す表の要素までが表引きの範囲となる。

SEARCH 文は、この表引き範囲の要素を逐次参照可能とする。すなわち、表引き用指標名の出現番号は逐次 1 ずつ増やされる。したがって、どの条件も満たされずに SEARCH 文が終わった場合は、表引き用指標名には最高出現番号(OCCURS 句で指定した反復回数の最大値)に 1 加えた出現番号が設定される。

条件の 1 つが満たされると表引きは終わり、その条件に伴う無条件文または次の実行文に制御を移す。表引き用指標名には、この条件を満足させた要素の出現番号が設定される。

表引き用指標名の初期値が 1 より小さい場合または表の最高出現番号より大きい場合、表引きは直に終わり、制御は AT END 無条件文または次の実行文に移る。このとき表引き用指標名にはもとの値が残る。

SEARCH 文に VARYING 句の指定があり、かつその項目を表引き用指標として使用しない場合、その項目も表引き操作に従って逐次 1 ずつ増やされる。

例 01 SEARCH-TABLE.

```
02 SCH-DATA PIC X(12) VALUE
    "NX  COBOL85".
02 SCH-TABLE REDEFINES SCH-DATA
    PIC X OCCURS 12 TIMES INDEXED
    BY INDEX-1 INDEX-2.
```

:

PROCEDURE DIVISION.

:

```
SET INDEX-1 TO 3.
```

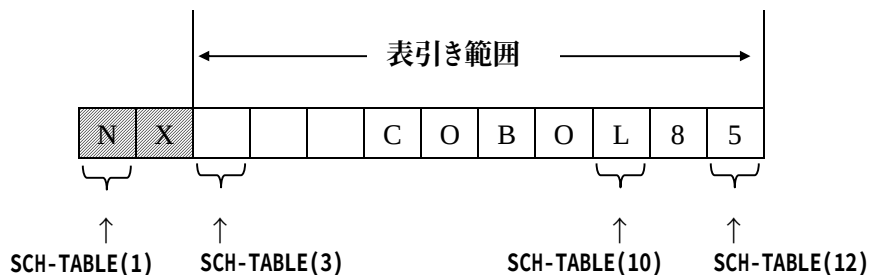
```
SET INDEX-2 TO 6.
```

- ① SEARCH SCH-TABLE
- ② SEARCH SCH-TABLE VARYING 他表の指標名
- ③ SEARCH SCH-TABLE VARYING 整数基本項目
- ④ SEARCH SCH-TABLE VARYING 指標データ項目
- ⑤ SEARCH SCH-TABLE VARYING INDEX-2

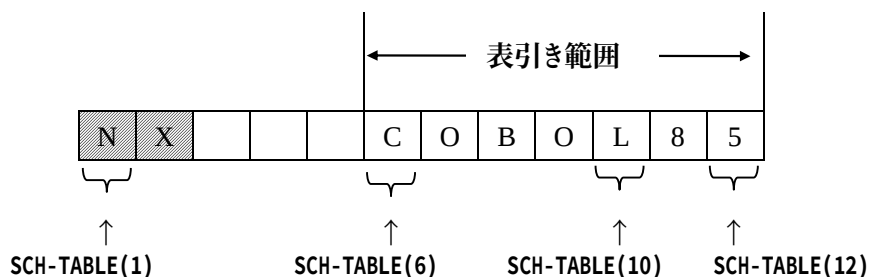
```
AT END GO TO ERROR-RTN.
```

:

上記 SEARCH 文の①, ②, ③, ④では, 表引き用指標として, INDEX-1 を使用し, 表引きの範囲は次に示す斜線以外の部分となる。



上記 SEARCH 文⑤では, 表引き用指標名として INDEX-2 を使用し, 表引きの範囲は次に示す斜線以外の部分となる。



さらに, この SEARCH 文で表中の"L"を探した場合, 表引き用指標名および VARYING 句の項目は次のようになる。

①, ②, ③, ④の SEARCH 文の表引き用指標名は

実行前の INDEX-1 の値: 3

実行後の INDEX-1 の値: 10

となり, さらに②, ③, ④の VARYING の項目にはその項目の形式に従って初期値+7 の値が設定される。

⑤の SEARCH 文の表引き用指標名は

実行前の INDEX-2 の値: 6

実行後の INDEX-2 の値: 10

となる。

(2) SEARCH 文では, WHEN 句によって複数個の条件を指定することができる。逐次表引き操作では, 1つの表の要素に対して WHEN 句の指定された順にその条件が満たされたかを検査する。表の要素がどの条件も満足しない場合, 表引きは次の要素に移る。

条件が満たされた時点で表引きは終わり、制御はその WHEN 句の無条件文または次の実行文に移る。

例 前記の表引きにおいて、(1)の SEARCH 文で WHEN 句を次のように書いた場合

(a) WHEN SCH-TABLE(INDEX-1) = "8" 無条件文 1

(b) WHEN SCH-TABLE(INDEX-1) = "5" 無条件文 2

(c) WHEN SCH-TABLE(INDEX-1) = "C" 無条件文 3

(d) WHEN SCH-TABLE(INDEX-1) = "B" 無条件文 4

"C"が最初に見つけれ、(c)の WHEN 句で表引きは終わり、無条件文 3 が実行される。

(3) SEARCH 文の表引きは、表引き用指標を逐次 1 ずつ加えていくことによって行われる。しかし、実際に検査される項目は WHEN 句で指定する条件の項目である。したがって、表の表引きを行う場合、WHEN 句に指定する条件の表を参照するデータ名は、その表の名前またはその表に從属するデータ項目の名前を使用し、表引き用指標名で添字付けすべきである。しかし、本 COBOL の SEARCH 文では WHEN 句の条件に対して何も規定していない。したがって、条件の項目に対し、表のデータ名に表引き用指標名以外のその表の指標名で添字付けしたり、他の表の指標名で添字付けすることもできる。この方法を使用することによって、異なる 2 つの表を同時に表引きしたり、2 次元の表を対角線上で表引きすることも可能である。しかし、この方法を使用する場合、実際には SEARCH 一意名で指定した表の表引きをすることにはならないので十分注意が必要である。

例 01 TABLE1.

02 SCH-TABLE1 PIC X OCCURS 10
TIMES INDEXED BY INDEX-1.

01 TABLE2.

02 SCH-TABLE2 PIC X OCCURS 10
TIMES INDEXED BY INDEX-2.

:

PROCEDURE DIVISION.

:

SET INDEX-1 INDEX-2 TO 1.

SEARCH SCH-TABLE1 VARYING INDEX-2

WHEN SCH-TABLE1(INDEX-1) = "P"

GO TO TABLE1-FOUND

WHEN SCH-TABLE2(INDEX-2) = "P"

GO TO TABLE2-FOUND.

TABLE1, TABLE2 に次のデータが入っているとします。

TABLE1										
	A	B	C	D	E	F	G	H	I	J
SCH-TABLE1	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	↑	↑	↑	↑	↑	↑				
	①	②	③	④	⑤	⑥	表引き操作順(⑥の P を見つけたとき終了)			
TABLE2	↓	↓	↓	↓	↓	↓				
	K	L	M	N	O	P	Q	R	S	T
SCH-TABLE2	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)

この SEARCH 文の表引き用指標名は INDEX-1 であり、表引き範囲は SCH-TABLE1 の出現番号 1 から 10 までである。しかし、VARYING 句で SCH-TABLE2 の指標名 INDEX-2 を指定しており、さらに WHEN 句の条件で SCH-TABLE2 をその指標名で添字付けして参照している。この結果、TABLE1, TABLE2 の両者を同時に表引きすることになる。

この SEARCH 文は、WHEN 句の SCH-TABLE2 (INDEX-2) によって、TABLE2 の "P" を見つけたとき条件を満足するため終わり、TABLE2-FOUND への GOTO 文が実行される。このとき、INDEX-1, INDEX-2 には出現番号 6 が設定される。もしこの表引きでどの条件も満足されない場合、SCH-TABLE1 の最高出現番号 10 を超えた時点、すなわち SCH-TABLE2 の "T" の項目を検査した後終了する。したがって、表の大きさが異なる場合、表引きの範囲は SEARCH 一意名-1 で指定した表の大きさによって決まる。

(4) SEARCH 文によって 2 次元、3 次元の表の全要素を表引きするためには SEARCH 文を何度か実行しなければならない。

例 2 次元の表を引く場合

```

01  SEARCH-TABLE.
    02 TABLE1 OCCURS 5 TIMES INDEXED
        BY INDEX-1.
    03  KEY1 PIC 9.
    03  TABLE2 OCCURS 10 TIMES
        INDEXED BY INDEX-2.
    04  KEY2 PIC 99.
        :
PROCEDURE DIVISION.
        :
    SET INDEX-1 TO 1.
SEARCH-RTN.
    SET INDEX-2 TO 1.
    SEARCH TABLE2
        WHEN KEY2(INDEX-1, INDEX-2)
            = データ項目 GO TO FOUND-RTN.
    IF INDEX-1 = 5 GO TO ERR-RTN ELSE
        SET INDEX-1 UP BY 1
        GO TO SEARCH-RTN.
FOUND-RTN.
        :

```

2次元目の表引きが1つ終わるごとに1次元目の表の指標名 INDEX-1 が1ずつ加えられ、2次元目の表のすべてが表引きされる。

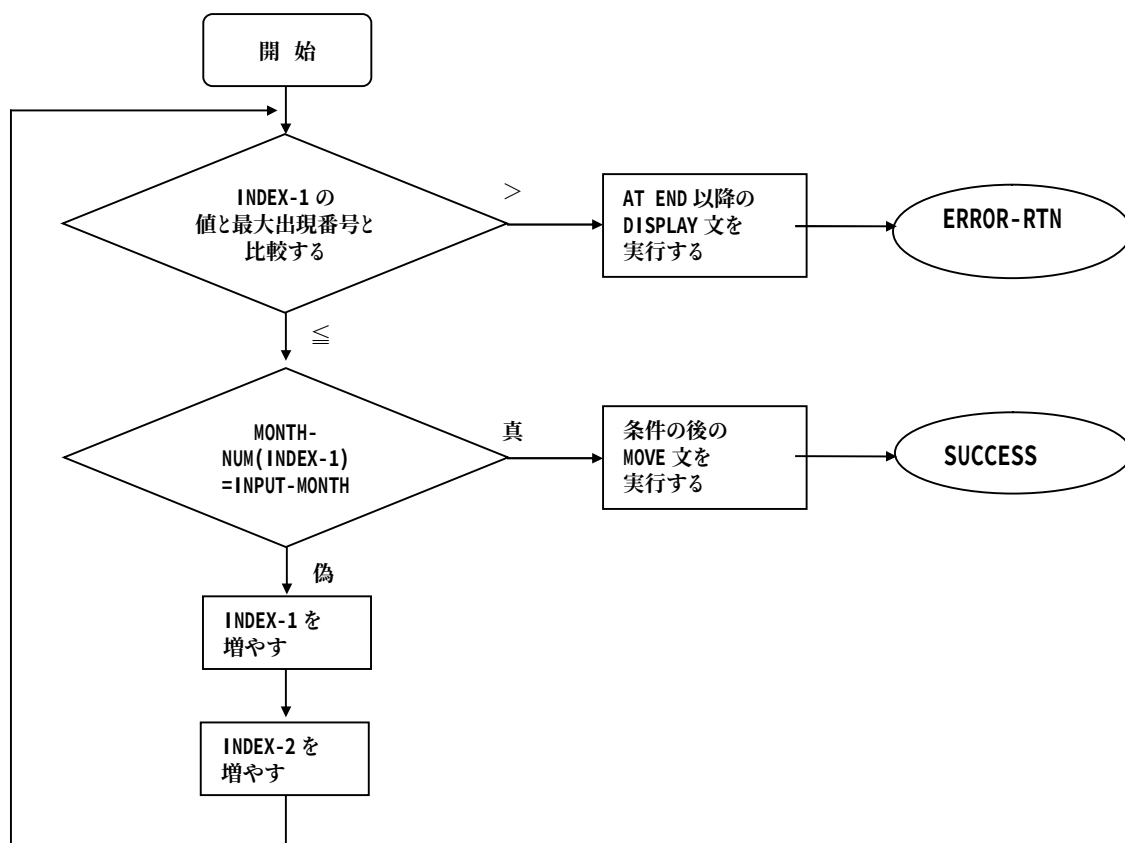
(5) SEARCH 文のプログラム例を示す。

```

01 INDEX-TABLE.
  02 CALENDER.
    03 QUARTER-1 PIC X(21) VALUE
      "JAN0131FEB0228MAR0331".
    03 QUARTER-2 PIC X(21) VALUE
      "APR0430MAY0531JUN0630".
    03 QUARTER-3 PIC X(21) VALUE
      "JUL0731AUG0831SEP0930".
    03 QUARTER-4 PIC X(21) VALUE
      "OCT1031NOV1130DEC1231".
  02 CAL-TABLE REDEFINES CALENDER
    OCCURS 12 TIMES INDEXED BY INDEX-1.
  03 CAL-ITEM.
    04 MONTH      PIC X(3).
    04 MONTH-NUM  PIC 99.
    04 MAX-DAYS   PIC 99.
01 JULIAN-VALUES.
  02 JULIANS PIC X(36) VALUE
    "001032060091121152182213244274305335".
  02 JULIAN-TABLE REDEFINES JULIANS
    OCCURS 12 TIMES INDEXED BY INDEX-2.
  03 FIRST-JULIAN PIC 999.
    :
PROCEDURE DIVISION.
    :
SEARCH-RTN.
  SET INDEX-1 INDEX-2 TO 1.
  SEARCH CAL-TABLE VARYING INDEX-2
    AT END DISPLAY "BAD INPUT MONTH"
      GO TO ERROR-RTN
    WHEN MONTH-NUM(INDEX-1) = INPUT-MONTH
      MOVE MONTH(INDEX-1) TO REPORT-MONTH
      MOVE FIRST-JULIAN(INDEX-2) TO BEGIN-JULIAN-DATE
      GO TO SUCCESS.
    :

```

前記の例における SEARCH 文の流れ図は次のようになる。



この例において、INPUT-MONTH の値が"5"であれば、SEARCH 文実行後の指標名 INDEX-1 と INDEX-2 には出現番号"5"が設定されている。さらに、REPORT-MONTH には"MAY"が、BEGIN-JULIAN-DATE には"121"が転記されている。制御は手続き名 SUCCESS に移る。

INPUT-MONTH の値が"13"以上であれば、AT END の後の DISPLAY "BAD INPUT MONTH"を実行し、制御は手続き名 ERROR-RTN に移る。

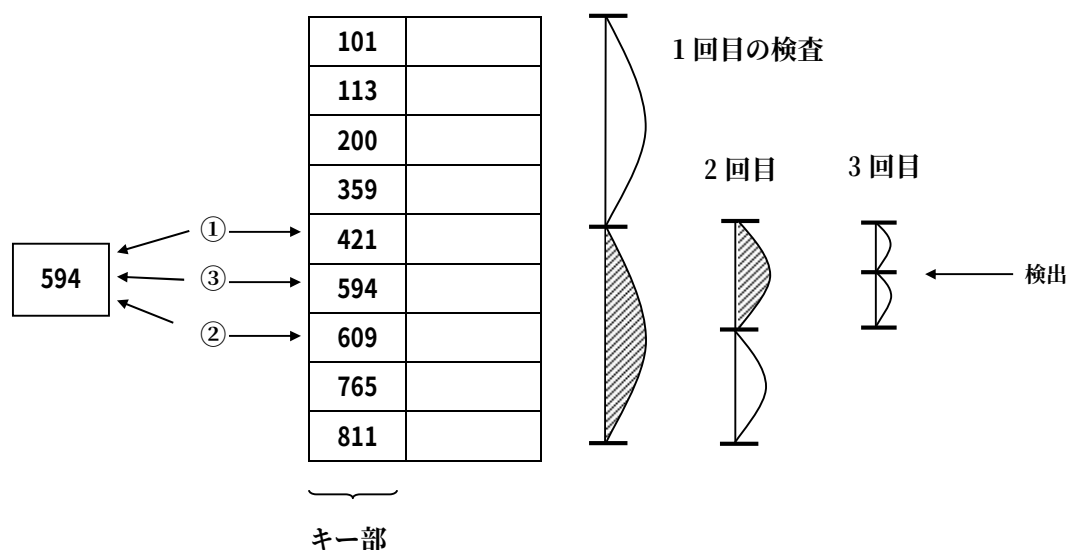
6.6.2 SEARCH ALL(2 分表引き)文

SEARCH ALL 文は、表が大きく、かつ表中のデータが規則的(昇順または降順)に並んでいる場合に有効である。

(1) SEARCH ALL 文の表引きには、2 分探索法(バイナリサーチ)を用いる。2 分探索法とは、表の

要素が規則的に並んでいることを利用して表引きの範囲を 2 分割しながらせばめていき、条件を満たす要素を検出する方法である。たとえば、ある表に 900 個の要素があるとする、**SEARCH**(逐次表引き)文では平均 450 回の検査が必要であるが、**SEARCH ALL**(2 分表引き)文では $2^9 < 900 < 2^{10}$ であるから最大 10 回の検査で済む。

例 次の表でキーが 594 の要素を探索する場合



上図の①, ②, ③は何回目の検査かを示し、斜線の部分は各回の検査で絞られた表引きの範囲を示す。

- (2) **SEARCH ALL** 文で表引きを行う場合、その作用対象の表の定義で **OCCURS** 句は **KEY IS** 句を伴わなければならない。**KEY IS** 句で指定するデータ名は、この **OCCURS** 句を含む記述項の名前かまたは、これに従属している必要があり、その値は **KEY IS** 句の指定に従って昇順(**ASCENDING**)または降順(**DESCENDING**)に並んでいる必要がある。

表引きには、対応する表の **INDEXED BY** 句で指定した最初の指標名が使用され、その指標名の当初の値は無視される。したがって、表引きの範囲は常に表の最初の要素から **OCCURS** 句で指定した反復回数の最大値(**DEPENDENT** 句があるときはそのデータ名 1 の値)が指す要素まで、すなわち表全体である。表引きは、すべての条件を満たす要素で終わり、その要素の出現番号が表引き用指標名に設定される。制御は、その条件に伴う無条件文または次の実行文に移る。どの条件も満足せずに **SEARCH ALL** 文が終わった場合、表引き用指標名の値は不定である。

例(1)

01 **SEARCH-TABLE.**

```
02 SCH-DATA PIC X(12) VALUE
    "ABCDE2345678".
02 SCH-TABLE REDEFINES SCH-DATA
    OCCURS 12 TIMES
    ASCENDING KEY IS AS-KEY
    INDEXED BY INDEX-1 INDEX-2.
03 AS-KEY PIC X.
    :
PROCEDURE DIVISION.
    :
    SEARCH ALL SCH-TABLE
        AT END GO TO ERROR-RTN
        WHEN AS-KEY(INDEX-1) = データ項目
            NEXT SENTENCE.
    :
```

前記の SEARCH ALL 文では、表引き用指標名として INDEX-1 が使用され、表全体を 2 分探索法で走査する。データ項目に"3"が指定されていた場合次のようになる。

SEARCH-TABLE											
A	B	C	D	E	2	3	4	5	6	7	8
SCH-TABLE(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
走査順						①	③		②		
INDEX-1 の値						6	7		9		

したがって、INDEX-1 には出現番号"7"が設定され、次の実行文が実行される。

- (3) SEARCH ALL 文の WHEN 句で表を参照する場合、そのデータ名は OCCURS 句の KEY IS 句で指定されているか、それに関する条件名でなければならない。KEY IS 句では、複数のデータ名を指定することができる。このデータ名は、KEY IS 句に指定された順に大分類→小分類のような分類順を示す。WHEN 句で KEY IS 句のデータ名を参照する場合またはそのデータ名に関する条件名を書く場合、KEY IS 句で指定した順序に従ってそのデータ名の前に定義したデータ名もすべて書いておく必要がある。WHEN 句に複数の条件を指定した場合、すべての条件を満足する要素で表引きは終わる。

例(2)

```
01 SEARCH-TABLE.
02 SCH-DATA PIC X(20) VALUE
    "A4A3A1B5B1C6D5D3D2D1".
02 SCH-TABLE REDEFINES SCH-DATA
    OCCURS 10 TIMES
    ASCENDING KEY IS AS-KEY
    DESCENDING KEY IS DS-KEY
    INDEXED BY INDEX-1.

03 AS-KEY PIC X.
03 DS-KEY PIC 9.
    :
PROCEDURE DIVISION
    :
    SEARCH ALL SCH-TABLE
    AT END GO TO ERROR-RTN
    WHEN AS-KEY(INDEX-1) = データ項目 1
    AND DS-KEY(INDEX-1) = データ項目 2
    GO TO FOUND-RTN.
```

前記の SEARCH ALL 文の表引き用指標名は INDEX-1 である。AS-KEY 項目の値は昇順に、DS-KEY 項目の値は AS-KEY 項目のもとで降順に並んでいなければならない。
表の "A1" を表引きすると次のようになる。

データ項目 1 = "A"
データ項目 2 = "1"

SEARCH-TABLE																			
A	4	A	3	A	1	B	5	B	1	C	6	D	5	D	3	D	2	D	1
AS-KEY	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)									
DS-KEY		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)								
検査順		②	③	④	⑤						①								
INDEX-1 の値		2		3						5									

したがって、INDEX-1 には出現番号"3"が設定され、FOUND-RTN への GO TO 文が実行される。

もしすべての条件を満足する要素がない場合、ERROR-RTN への GO TO 文が実行される。

- (4) SEARCH ALL 文で 2 次元または 3 次元の表の全要素を表引きするためには、SEARCH ALL 文を何度か実行する必要がある。この操作は、SEARCH (逐次表引き) 文と同様である。詳細は、「6.6.1 SEARCH(逐次表引き)文(4)」を参照。
- (5) SEARCH ALL 文の表引きにおいて、その表の要素の内容が一意でない場合には SEARCH ALL 文によって見つけられる要素の出現番号は不定である。
- (6) SEARCH ALL 文のプログラム例を示す。

01 INDEX-TABLE.

02 CALENDER.

03 QUATER-1 PIC X(21) VALUE
"JAN0131FEB0228MAR0331".

03 QUATER-2 PIC X(21) VALUE
"APR0430MAY0531JUN0630".

03 QUATER-3 PIC X(21) VALUE
"JUL0731AUG0831SEP0930".

03 QUATER-4 PIC X(21) VALUE
"OCT1031NOV1130DEC1231".

02 CAL-TABLE REDEFINES CALENDER

OCCURS 12 TIMES ASCENDING KEY IS
MONTH-NUM INDEXED BY INDEX-1.

03 CAL-ITEM.

04 MONTH PIC X(3).

04 MONTH-NUM PIC 99.

04 MAX-DAYS PIC 99.

01 JULIAN-VALUES.

02 JULIANS PIC X(36) VALUE
"001032060091121152182213244274305335".

02 JULIAN-TABLE REDEFINES JULIANS

OCCURS 12 TIMES INDEXED BY INDEX-2.

03 FIRST-JULIAN PIC 999.

01 REPORT-MONTH PIC X(03).

01 BEGIN-JULIAN PIC 9(03).

:


```

PROCEDURE DIVISION.
:
SEARCH-RTN.
    SEARCH ALL CAL-TABLE
    AT END DISPLAY "BAD INPUT MONTH"
    GO TO ERROR-RTN
    WHEN MONTH-NUM(INDEX-1) = INPUT-MONTH
    MOVE MONTH(INDEX-1) TO REPORT-MONTH
    SET INDEX-2 TO INDEX-1
    MOVE FIRST-JULIAN(INDEX-2) TO BEGIN-JULIAN
    GO TO SUCCESS.
:

```

前記の例で、INPUT-MONTH の値が"5"であれば SEARCH ALL 文実行後 の指標名 INDEX-1 には、出現番号として"5"が設定される。さらに、REPORT-MONTH には"MAY"を、INDEX-2 には INDEX-1 を設定し、BEGIN-JULIAN には"121"が転記される。制御は、手続き名 SUCCESS に移る。

INPUT-MONTH の値が"13"以上であれば DISPLAY "BAD INPUT MONTH"を実行し、制御は手続き名 ERROR-RTN に移る。

6.7 表の参照時の注意事項

表を参照する場合には、次の点に注意すること。

相対添字付け参照による表の参照において、指標名またはデータ名の出現番号は 1 以上、OCCURS で指定した繰り返し数以下で使用すべきである。この範囲でないと、指標名またはデータ名に続く＋整数または－整数を加算した結果の出現番号が、表の要素を示す値でも、正しく参照されない場合がある。

例

```

:
DATA DIVISION
:
01 TABLE-1.
    02 TABLE1 PIC X(100) OCCURS 1 TO 100 DEPENDING ON TBL-SIZE
    INDEXED BY IX.

```

```
      :  
PROCEDURE DIVISION  
      :  
      SET  IX  TO  101  
      MOVE TABLE1(IX - 10) TO TABLE1(IX - 50)  
      :  
      :      好ましくない
```

The diagram illustrates a problem with the COBOL code. Two arrows originate from the text '好ましくない' (Not preferred) and point to the expressions 'IX - 10' and 'IX - 50' within the MOVE statement. This indicates that using arithmetic expressions to calculate table indices is not the preferred method.

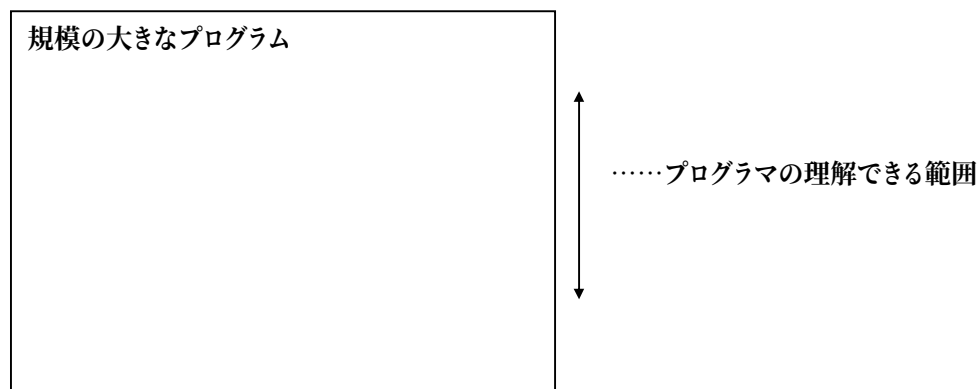
第7章 プログラム間連絡機能

7.1 概 説

本章では、複数の COBOL 原始プログラムが互いに連絡し合うために提供するプログラム間連絡機能について説明する。

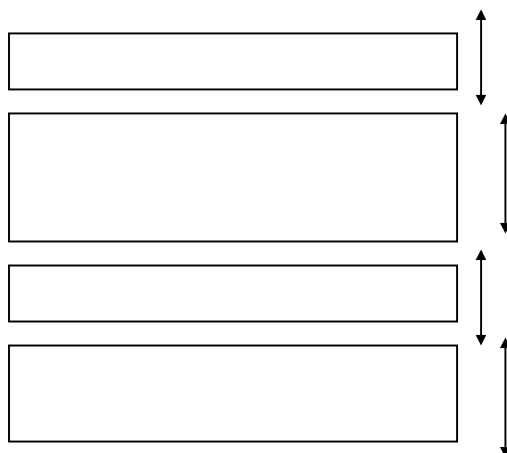
7.2 プログラムの分割

単純な問題をプログラミングする場合、プログラミングの規模も小さくプログラムを分割する必要はないが、プログラムの規模が大きくなるに従って、プログラム開発の生産性が低下し、また正しく動作するプログラムの作成が困難となる。



これは、プログラマには 1 度で理解できる範囲があり、この範囲を大きく超えるような規模のプログラムは十分な理解がむずかしいためである。

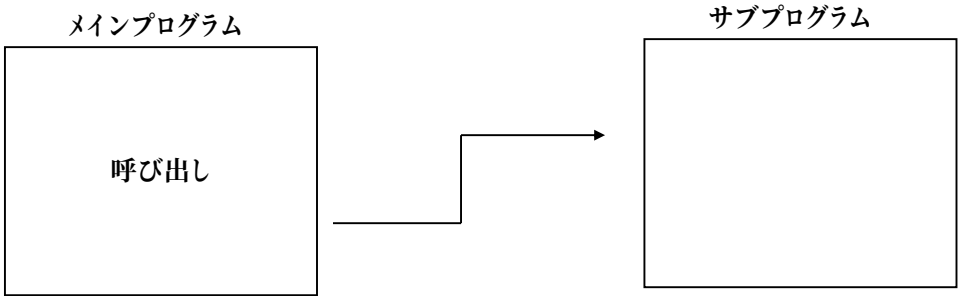
このために、プログラムを適当な大きさに分割する必要がある。



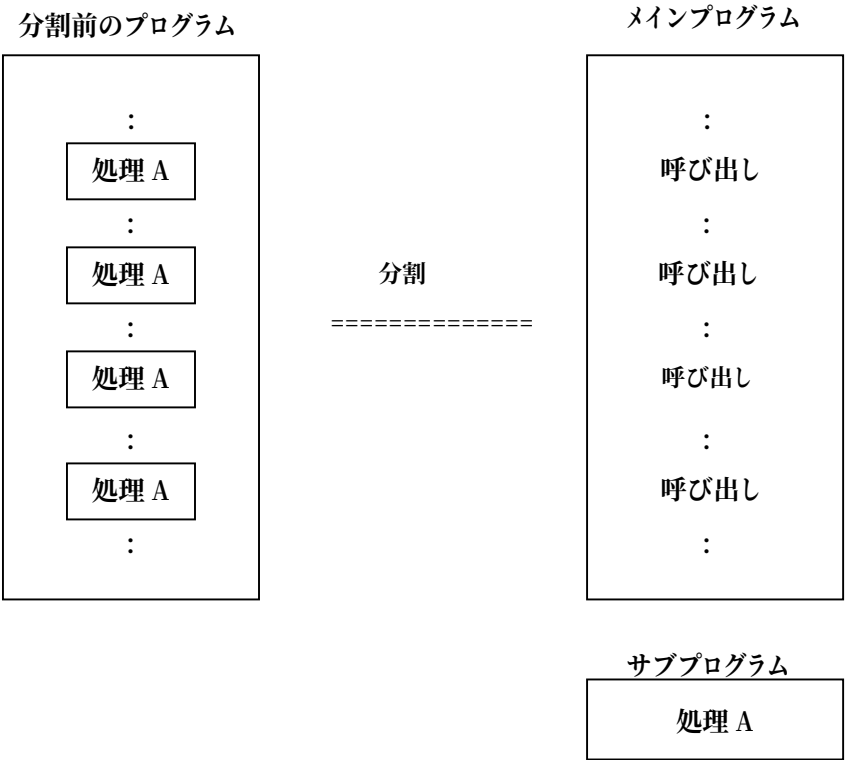
なお、プログラムを分割する場合の留意点として、分割したプログラムが単独でも理解できるようにすること、分割した各プログラムの規模がプログラマの理解できる範囲を超えないようにすることが重要である。

分割したプログラムのうちで、別のプログラムを呼び出すものをメインプログラムという。

また、呼び出されるプログラムをサブプログラムという。



また、プログラムの中で何度も繰り返される処理をサブプログラムとして分割した場合は、プログラム全体の規模を小さくすることができる。



特に、処理Aを行うサブプログラムがすでに作成／デバッグ済みであれば、これを流用することもできる。

[Enterprise] なお、本コンパイラでは、メインプログラムとなるプログラムは、翻訳時に-M オプションを指定して翻訳しなければならない。逆にサブプログラムは、-M オプションを指定してはならない。

[Standard] なお、本コンパイラでは、メインプログラムとなるプログラムは、翻訳時に-M または-Mw オプションを指定して翻訳しなければならない。逆にサブプログラムは、-M または-Mw オプションを指定してはならない。

7.3 プログラムの入れ子

COBOL 原始プログラムは、別の COBOL 原始プログラムを含むことができる。プログラムの入れ子の例を図 7-1 に示す。

図 7-1 プログラムの入れ子

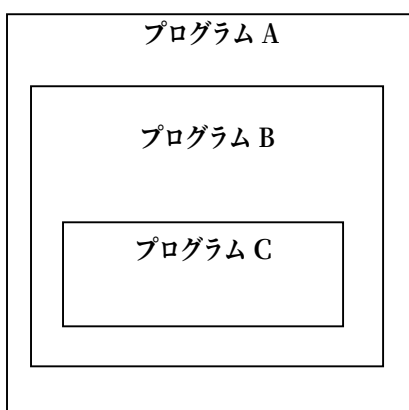
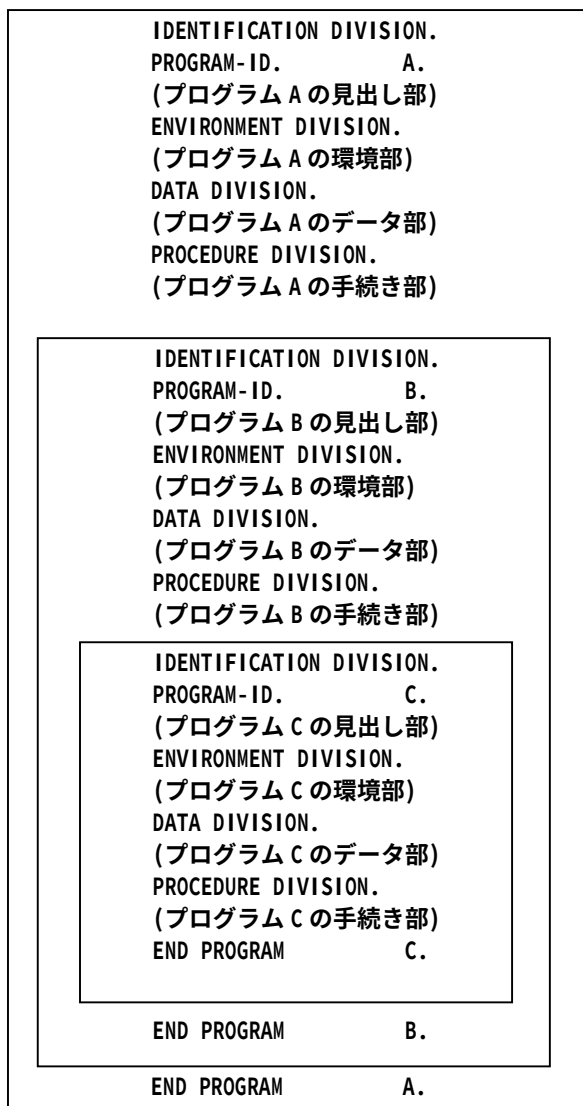


図 7-1 の例で、プログラム A とプログラム B の包含関係を ”プログラム B がプログラム A に直接的に含まれる” といい、プログラム A とプログラム C の関係を ”プログラム C がプログラム A に間接的に含まれる” という。

図 7-1 のプログラムの入れ子の例に対応する COBOL の記述を図 7-2 に示す。

図 7-2 プログラムの入れ子の COBOL 記述



特に、入れ子の包含関係で最も外側にあるプログラム(図 7-2 の例では、プログラム A)を翻訳単位のプログラムという。COBOL 原始プログラムを翻訳して得られる目的プログラムの個数は、入れ子のプログラムの有無／個数に関係なく常に翻訳単位のプログラムに対して 1 個である。

7.3.1 局所名と全域名

定義したプログラムだけから参照が許されるような名前を局所名といい、定義したプログラムの他にそのプログラムに含まれるプログラムからも参照が許されるような名前を全域名という。

COBOL の利用者語のうち、ファイル名、条件名、データ名、指標名、レコード名が全域名となる場合を表 7-1 に示す。

COBOL の利用者語のうち、翻訳単位のプログラムの構成節で定義された符号系名、字類名、呼び名、記号文字およびスイッチ状態の条件名は、その翻訳単位のプログラムとこのプログラムに含まれるすべてのプログラムから参照できる。

COBOL の利用者語のうち、登録集名および原文名は、すべてのプログラムから参照できる。

表 7-1 全域名となる場合

ファイル名	・ そのファイル記述項に GLOBAL 句が書かれた場合
条件名	・ 関連する条件変数のデータ名が全域名の場合
データ名	・ そのデータ記述項(01 レベル)に GLOBAL 句が書かれた場合 ・ GLOBAL 句が書かれたデータ記述項に従属している場合
レコード名	・ そのレコード記述項(01 レベル)に GLOBAL 句が書かれた場合 ・ 関連するファイル名が全域名の場合(ファイル節のレコード名のとき)
指標名	・ GLOBAL 句が書かれたデータ記述項に従属している場合

COBOL の利用者語のうち、段落名および節名は、その名前を定義したプログラムだけから参照できる。

また、特殊レジスタが全域名となる場合を表 7-2 に示す。

表 7-2 特殊レジスタが全域名となる場合

LINAGE-COUNTER	・ LINAGE 句を書いたファイルのファイル名が全域名の場合

7.3.2 内部項目と外部項目

定義したプログラム内に領域が確保されるような作用対象(データ項目)を内部項目といい、定義したプログラム内に領域を確保せずそのプログラムが属する実行単位で 1 つだけ領域が確保されるような作用対象を外部項目という。

作業場所節で定義されたレコード領域および指標名が外部項目となる場合を表 7-3 に示す。

表 7-3 レコード領域および指標名が外部項目となる場合

作業場所節で定義されたレコード領域	・ そのデータ記述項(01 レベル)に EXTERNAL 句が書かれた場合
指標名	・ 関連するレコード領域が外部項目の場合

整列併合用ファイルのレコード領域は、常に内部項目となる。
特殊レジスタも常に内部項目である。

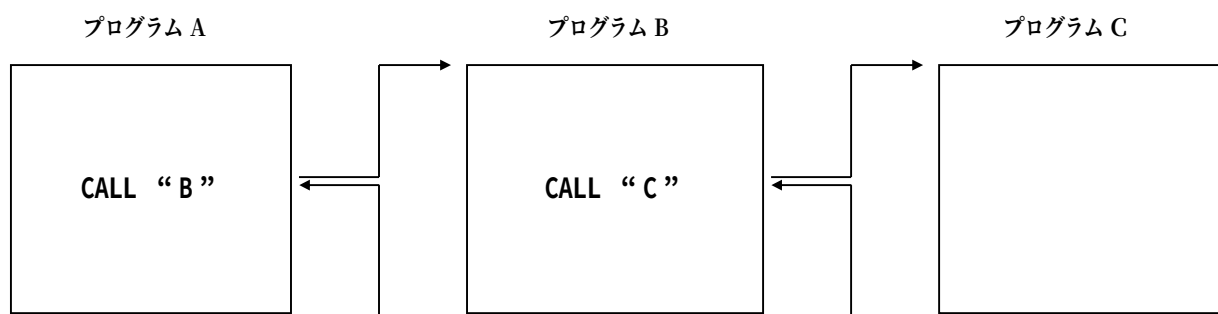
7.3.3 COBOL の利用者語の有効範囲

COBOL の利用者語の定義に関する規則を、図 7-2 の例で説明する。プログラム B では、修飾で一意にできる条件名、データ名、レコード名を除く名前はただ 1 種類の利用者語の名前として定義できるだけだが、プログラム A やプログラム C では、その名前を別の利用者語として定義できる。ただし、翻訳単位のプログラムであるプログラム A の構成節で符号系名、字類名、呼び名、記号文字およびスイッチ状態の条件名として定義された名前は、すべてのプログラムで別の利用者語として定義することはできない。

COBOL の利用者語のうち、ファイル名、条件名、データ名、指標名およびレコード名の参照に関する規則を、図 7-2 で説明する。プログラム A は翻訳単位のプログラムなので、プログラム A で参照できる名前は、プログラム A で定義された名前だけである。プログラム B で参照できる名前は、プログラム B で定義された名前とプログラム A で全域名として定義された名前である。この場合、まずプログラム B で定義された名前を探し、見つからないときだけプログラム A で全域名として定義された名前を探す。プログラム C で参照できる名前は、プログラム C で定義された名前とプログラム A およびプログラム B で全域名として定義された名前である。この場合、まずプログラム C で定義された名前を探し、見つからないときだけプログラム B で全域名として定義された名前を探す。それでも見つからなければプログラム A で全域名として定義された名前を探す。

7.4 サブプログラムの呼び出し

メインプログラムからサブプログラムを呼び出す文として、CALL 文がある。呼び出されたサブプログラムがさらに別のサブプログラムを呼び出していてもよい。ただし、プログラムが自分自身を間接的に呼び出すこと(再帰呼び出しという)はできない。サブプログラムの呼び出しの例を次に示す。



メインプログラムとサブプログラムの関係により、サブプログラムの呼び出しには次の種類がある。

- 含まれるプログラムの呼び出し
- 共通プログラムの呼び出し
- 外部プログラムの呼び出し

CALL 文で指定した名前がメインプログラムに直接含まれるプログラムの名前であれば、その CALL 文は、含まれるプログラムの呼び出しとなる。CALL 文で指定した名前がメインプログラムに直接的に含まれるプログラムの名前ではなく呼び出し可能な共通プログラムの名前であれば、その CALL 文は共通プログラムの呼び出しとなる。上記のいずれかの場合でもなければ、その CALL 文は外部プログラムの呼び出しとなる。

7.4.1 含まれるプログラムの呼び出し

プログラムの入れ子において、メインプログラムに直接的に含まれるプログラムをサブプログラムとして呼び出すことができる。含まれるプログラムの呼び出しの例を図 7-3 に示す。

図 7-3 含まれるプログラムの呼び出しの例

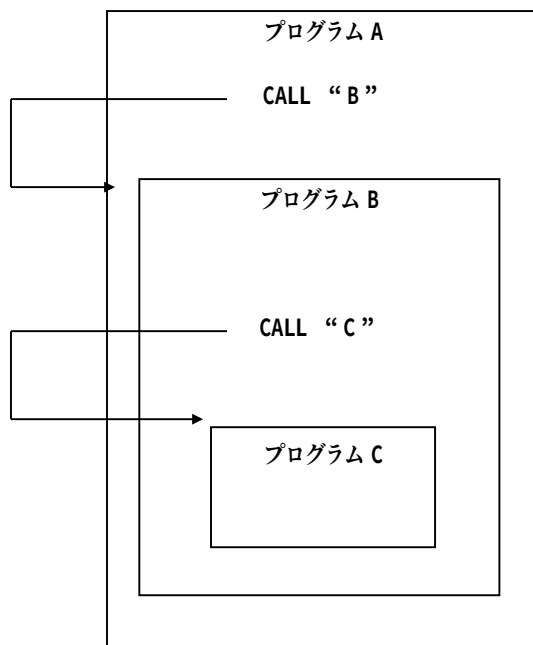


図 7-3 の例で、プログラム A はプログラム B だけを呼び出すことができる。プログラム B はプログラム C だけを呼び出すことができる。プログラム C はどのプログラムも呼び出すことができない。

7.4.2 共通プログラムの呼び出し

手続き部の PROGRAM-ID 段落に COMMON 句を書いたプログラムを共通プログラムという。共通プログラムは、翻訳単位中で共通に呼び出すことのできるサブプログラムとなる。

共通プログラムの呼び出しの例を図 7-4 に示す。

図 7-4 共通プログラムの呼び出しの例

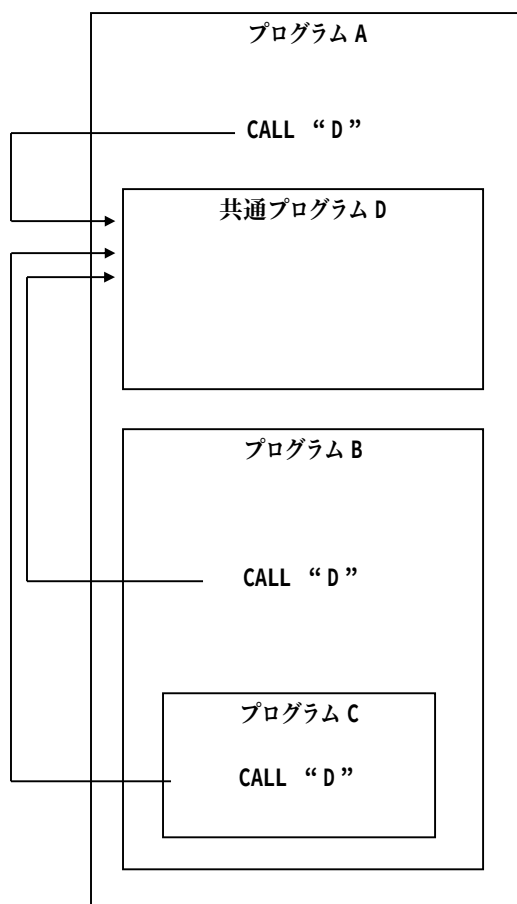


図 7-4 でプログラム A が翻訳単位のプログラムである場合、共通プログラム D は翻訳単位中のすべてのプログラムから呼び出すことができる。そうでなければ、共通プログラム D はプログラム A とプログラム A に含まれるプログラムから呼び出すことができる。

7.4.3 外部プログラムの呼び出し

メインプログラムは別の翻訳単位のプログラムをサブプログラムとして呼び出すことができる。外部プログラムの呼び出しの例を図 7-5 に示す。

図 7-5 外部プログラムの呼び出し例

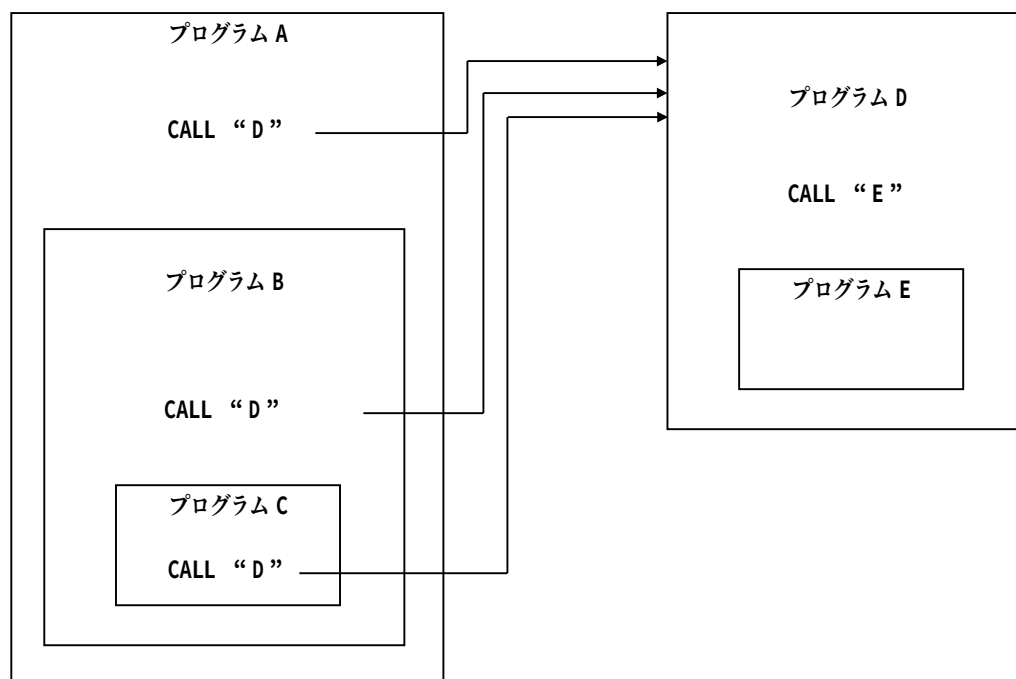


図 7-5 の例で翻訳単位のプログラム D は、別翻訳単位のプログラム A とプログラム A に含まれるすべてのプログラムから呼び出すことができる。ただし、プログラム E は、別の翻訳単位のプログラム A とプログラム A に含まれるプログラムから直接呼び出すことができない。

7.4.4 プログラム原型を利用した呼び出し

プログラム原型を用いることで、CALL 文でのサブプログラムの呼び出しにおいて、引数および返却項目が適合しているかを翻訳時に検査できる。

下記がプログラム原型を記載したプログラム例である。

***>プログラム原型**

IDENTIFICATION DIVISION.

PROGRAM-ID. SUBPROG IS PROTOTYPE.

DATA DIVISION.

LINKAGE SECTION.

01 PARAM USAGE COMP-2.

01 RESULT PIC S9(9) USAGE BINARY.

```
PROCEDURE DIVISION USING PARAM RETURNING RESULT.
END PROGRAM SUBPROG.
```

*>メインプログラム

```
IDENTIFICATION DIVISION.
PROGRAM-ID. MPROG.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    PROGRAM SUBPROG.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  INPVAL PIC 9(4) USAGE DISPLAY.
01  PARAM USAGE COMP-2.
01  RESULT PIC S9(9) USAGE BINARY.
PROCEDURE DIVISION.
主処理.
    DISPLAY "値を入力してください.".
    ACCEPT INPVAL.
    MOVE INPVAL TO PARAM.
    DISPLAY PARAM.
    CALL "SUBPROG" USING PARAM RETURNING RESULT.
    DISPLAY RESULT.
END PROGRAM MPROG.
```

上記のプログラムは、SUBPROG のプログラム原型と、その SUBPROG を呼び出すメインプログラムから構成されている。メインプログラムのリポジトリ段落に SUBPROG が指定されていることに注意されたい。この指定により、翻訳時に SUBPROG を呼び出す CALL 文の引数および返却値が、呼び出し側と定義側でデータの型が一致しているかどうかの検査が行われる。

なお、プログラム原型をあらかじめ翻訳し、リポジトリ情報を生成しておけば、必ずしもプログラム原型を記載する必要はない。翻訳時に外部リポジトリ情報を指定することで、適合検査が行われる。ただし、プログラムの可読性を高めるためには、プログラム原型を記載しておくのが望ましい。

7.4.5 プログラムの再帰呼び出し

プログラム定義時に再帰属性(RECURSIVE)を付加することで、自分自身のプログラムを直接または間接的に再帰呼び出しすることができる。

再帰呼び出しを可能にするには、プログラム共通のデータを作業場所節に定義し、呼び出しごとに用途が異なるデータを局所記憶節に定義する必要がある。

以降に、再帰プログラムの例を示す。

FACTORIALX は、仮パラメータ PAR1 で与えた整数 N の階乗を計算して、その値を返却値 RET1 に設定するサブプログラムである。プログラム名段落で RECURSIVE を指定することで再帰呼び出し可能であることを示し、手続き中では自分自身のプログラムを CALL 文にて呼び出している。また、FACTORIALP は、このサブプログラムを呼び出すメインプログラムであり、サブプログラム FACTORIX を呼び出して 5 の階乗の計算を行っている。

```
*>再帰呼び出し可能プログラム
IDENTIFICATION DIVISION.
PROGRAM-ID.    FACTORIALX IS RECURSIVE.
DATA          DIVISION.
LOCAL-STORAGE SECTION.
01 WKPAR1     PIC S9(3).
01 COUNT1     PIC S9(3).
LINKAGE       SECTION.
01 PAR1       PIC S9(3).
01 RET1       PIC S9(3).
PROCEDURE DIVISION USING PAR1 RETURNING RET1.
HAJIME.
    IF PAR1 = 0 THEN
        MOVE 1 TO RET1
        EXIT PROGRAM
    END-IF.
    COMPUTE WKPAR1 = PAR1 - 1.
    CALL "FACTORIALX" USING WKPAR1 RETURNING COUNT1.
    COMPUTE RET1 = PAR1 * COUNT1.
END PROGRAM FACTORIALX.
```

```

*>メインプログラム：5 の階乗を計算して表示する
IDENTIFICATION DIVISION.
PROGRAM-ID.    FACTORIALP.
DATA          DIVISION.
WORKING-STORAGE SECTION.
01  PAR1      PIC S9(3) VALUE 5.
01  RESULT1   PIC S9(3) VALUE 0.
PROCEDURE     DIVISION.
HAJIME.
    CALL "FACTORIALX" USING PAR1 RETURNING RESULT1.
    DISPLAY RESULT1.
    STOP RUN.
END PROGRAM FACTORIALP.

```

7.5 プログラム間のデータの受渡し

メインプログラムとサブプログラムの間のデータの受け渡しには、次の 3 種類がある。

- 引数による受け渡し
- 外部項目による受け渡し
- 全域名による受け渡し

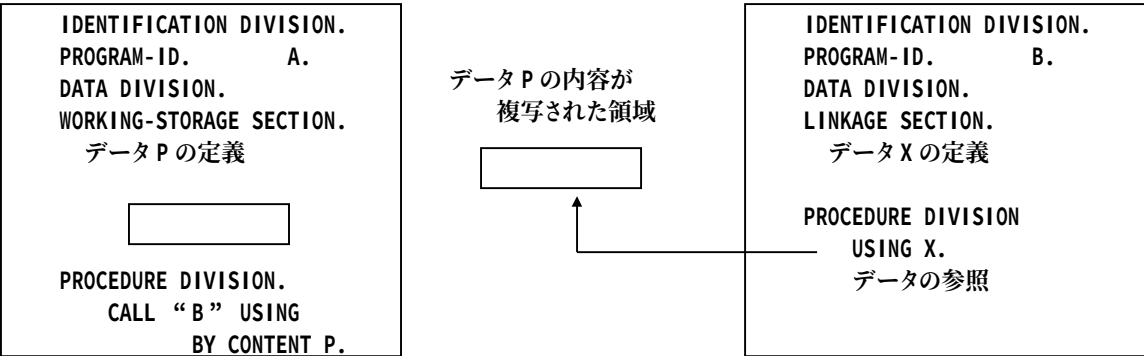
7.5.1 引数によるデータの受渡し

サブプログラムのデータ部に連絡節を定義し、CALL 文の USING 句とサブプログラムの手続き部の見出しの USING 句の両方に引数を書くことで、データを受け渡すことができる。ただし、この方法で受け渡すことができるのは、メインプログラム側のデータ部のファイル節、作業場所節、局所記憶節、連絡節またはベースデータ節で定義されたレベル番号 01～49, 77 のデータ項目に限られる。引数によって受け渡されるデータは、名前ではなく USING 句に書かれた順序で対応付けられる。対応付けられたデータのデータ記述は、メインプログラムとサブプログラムで同一でなければならない。CALL 文の USING 句の引数の個数は、サブプログラムの手続き部の USING 句の引数の個数と同一でなければならない。

また、CALL 文の USING 句の引数に BY CONTENT 句を書くことができる。この場合、その引数によって受け渡されるデータをサブプログラムで変更してもメインプログラム側のデータ項目の内容は

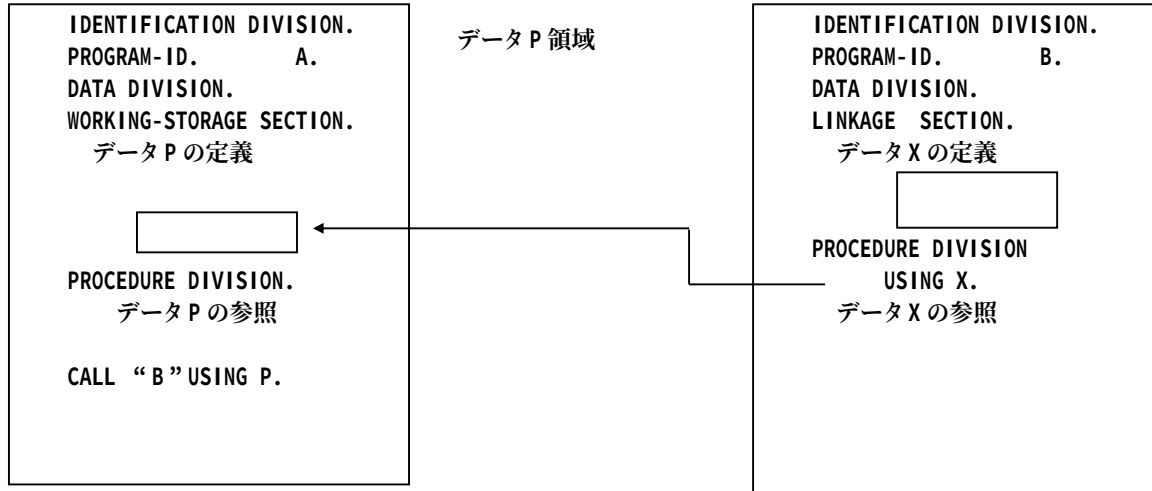
変更されない。BY CONTENT 句を書いた引数の受け渡しの例を図 7-6 に示す。

図 7-6 BY CONTENT 句を書いた引数による受け渡しの例



CALL 文の USING 句の引数に BY REFERENCE 句を書くことができる。この場合、その引数によって受け渡されるデータをサブプログラムで変更するとメインプログラム側のデータ項目の内容も変更される。BY REFERENCE 句を書いた引数の受け渡しの例を図 7-7 に示す。

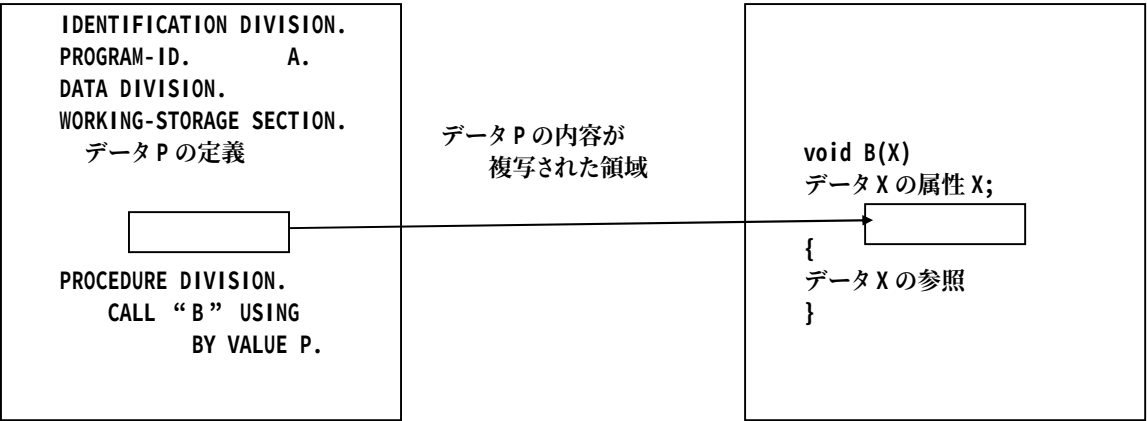
図 7-7 BY REFERENCE 句を書いた引数による受け渡しの例



CALL 文の USING 句の引数に BY VALUE 句を書くことができる。この場合、サブプログラムの見出し部の USING 句にも BY VALUE 句が指定されていなければならない。BY VALUE 句は、その引数を値で渡すことを指定する。サブプログラム側でその引数の内容を変更してもメインプログラム側には何ら影響を及ぼさない。

BY VALUE 句を書いた引数の受け渡しの例を図 7-8 に示す。

図 7-8 BY VALUE 句を書いた引数による受け渡しの例

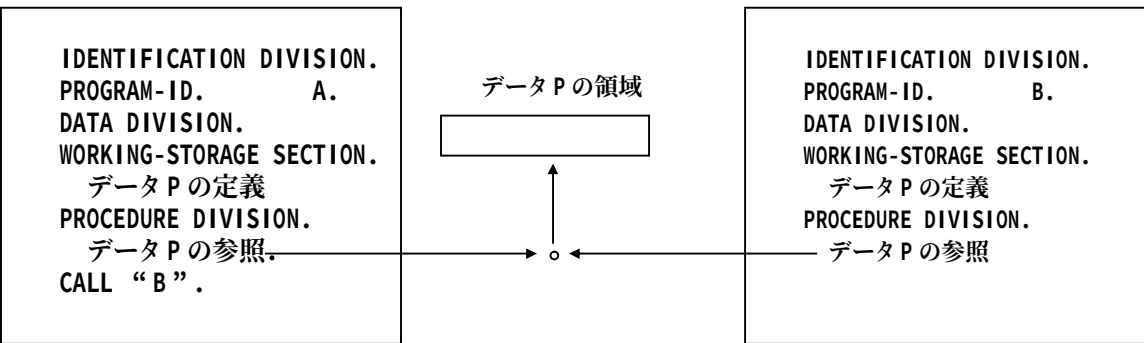


注意 入れ子プログラムに対して BY VALUE 句による引数の受け渡しをすることはできない。

7.5.2 外部項目による受け渡し

メインプログラムとサブプログラムの両方で外部項目として定義することで、データを受け渡すことができる。ただし、この方法で受け渡すことができるのは、外部ファイル結合子、作業場所節で定義された外部レコード領域、外部レコード領域に関連付けられた指標領域に限られる。外部項目による受け渡しの例を図 7-9 に示す。

図 7-9 外部項目による受け渡しの例



本コンパイラでは、作業場所節またはファイル節の外部レコードを、実行ファイル内で受け渡し可能とするか、実行単位全体で受け渡し可能とするかを選択することができる。

複数のプログラムで同じファイル識別リテラルを持つ SYSIN ファイルまたは SYSOUT ファイルの定義がある場合の指定は、次のとおりである。

- ・EXTERNAL 指定のある場合は、実行ファイル内で受け渡し可能とするか、実行単位全体で受け渡し可能とするかを指定しなければならない。

- ・EXTERNAL 指定のない場合は、実行ファイル内、実行単位全体とも受け渡し可能と指定してはならない。

実行ファイル内または実行単位全体で受け渡し可能と指定するプログラム中で EXTERNAL 指定のない SYSIN ファイルまたは SYSOUT ファイルの定義を行う必要がある場合は、それぞれ異なるファイル識別リテラルとして定義し直すか、データ名 1 指定、定数 4 指定に変更しなければならない。

データ名 1 指定、定数 4 指定については、『COBOL 言語説明書』「8.4.2 ファイル管理記述項」を参照。

【Enterprise】複数のプログラムで同じファイル識別リテラルを持つ RAF ファイルの定義がある場合の指定は、次のとおりである。

- ・EXTERNAL 指定のある場合は、実行ファイル内で受け渡し可能とするか、実行単位全体で受け渡し可能とするかを指定しなければならない。

- ・EXTERNAL 指定のない場合は、実行ファイル内、実行単位全体とも受け渡し可能と指定してはならない。

【Enterprise】実行ファイル内または実行単位全体で受け渡し可能と指定するプログラム中で EXTERNAL 指定のない RAF ファイルの定義を行う必要がある場合は、それぞれ異なるファイル識別リテラルとして定義し直すか、データ名 1 指定、定数 4 指定に変更しなければならない。

データ名 1 指定、定数 4 指定については、『COBOL 言語説明書』「8.4.2 ファイル管理記述項」を参照。

それぞれの利用方法について、次に説明する。

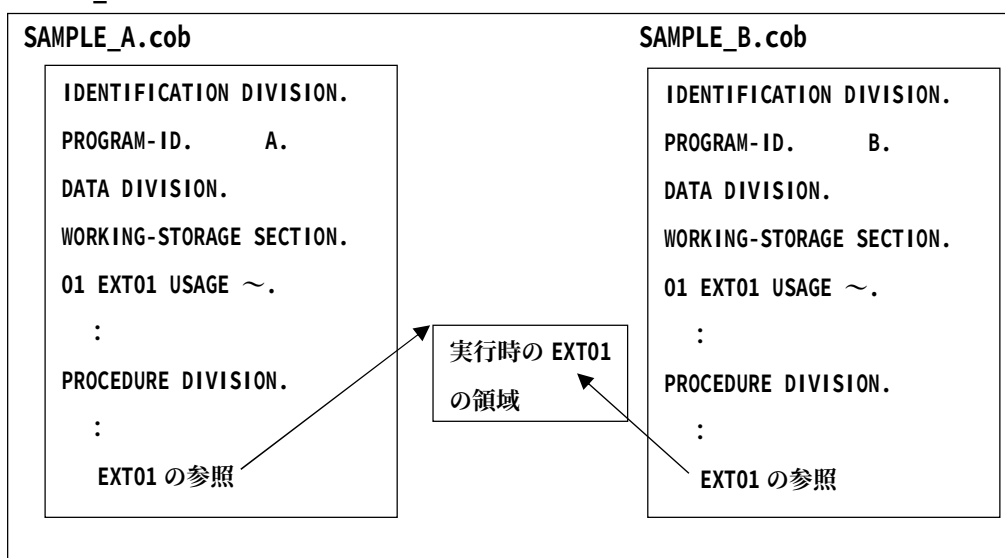
(1) 実行ファイル内で受け渡し可能とする場合

外部項目の実体は1つでなければならない。

このため、翻訳時に外部項目の実体を1つにする方法を以下に示す。

- **【Enterprise for Windows】【Standard】リンク時に1つにする方法**

外部項目の定義および参照を指定できない場合、翻訳は通常どおり行い、リンク時にリンカオプション「/FORCE」を指定する。

SAMPLE_A.exe**<生成例>**

COBOL の翻訳時、オブジェクト生成までは通常のアプリケーション生成方法と同様。

```
> cob -M -c SAMPLE_A.cob SAMPLE_B.cob
```

【Standard】または

```
> cob -Mw -c SAMPLE_A.cob SAMPLE_B.cob
```

リンク時にリンカオプション「/FORCE」の指定を行う。

```
> cob SAMPLE_A.obj SAMPLE_B.obj SAMPLE_C.lib -wl:/FORCE
```

または

```
> link /FORCE SAMPLE_A.obj SAMPLE_B.obj
```

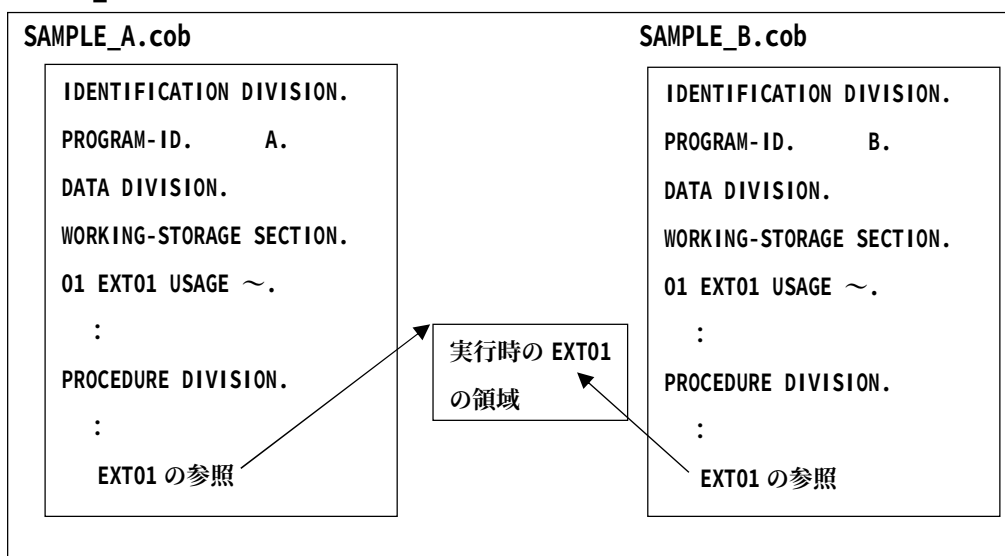
注意

- ・「/FORCE」オプションは、未定義のシンボルや重複定義のシンボルがある場合も、強制的に実

行可能ファイルを生成する。COBOL プログラムで EXTERNAL 句を指定したデータ以外に、未定義のシンボルや重複定義のシンボルがある場合に、動作不正を伴う場合がある。

- ・ SAMPLE_A.cob と SAMPLE_B.cob がそれぞれ実行ファイルとなる場合 (たとえば, SAMPLE_A.exe と SAMPLE_B.dll のような場合) は, データの受け渡しはできない。
 - ・ SAMPLE_A.cob と SAMPLE_B.cob が同一実行ファイルの DLL (たとえば, A.dll) となる場合も, データの受け渡しが可能。
- ・ **[Enterprise for Linux] リンク時に 1 つにする方法**
外部項目の定義および参照を指定できない場合, 翻訳は通常どおり行い, リンク時にリンカオプション「`--allow-multiple-definition`」を指定する (リンカオプション「`-z muldefs`」も同義である)。

SAMPLE_A



<生成例>

COBOL の翻訳時, オブジェクト生成までは通常のアプリケーション生成方法と同様。

```
> cob -M -c SAMPLE_A.cob SAMPLE_B.cob
```

[Standard] または

```
> cob -Mw -c SAMPLE_A.cob SAMPLE_B.cob
```

リンク時にリンカオプション「`--allow-multiple-definition`」の指定を行う。

```
> cob SAMPLE_A.o SAMPLE_B.o SAMPLE_C.a -Wl,--allow-multiple-definition
```

または

```
> cob SAMPLE_A.o SAMPLE_B.o SAMPLE_C.a -Wl,-z,muldefs
```

または

```
> gcc SAMPLE_A.o SAMPLE_B.o SAMPLE_C.a -Wl,--allow-multiple-definition
```

注意

- ・「**--allow-multiple-definition**」オプションは、重複定義のシンボルがある場合も、強制的に実行可能ファイルを生成する。COBOL プログラムで **EXTERNAL** 句を指定したデータ以外に、重複定義のシンボルがある場合に、動作不正を伴う場合がある。
- ・**SAMPLE_A.cob** と **SAMPLE_B.cob** がそれぞれ実行ファイルとなる場合(たとえば、実行ファイル **SAMPLE_A** と 共有ライブラリ **SAMPLE_B.so** のような場合)は、**--allow-multiple-definition** オプションを指定しない場合もデータの受け渡しが可能である。
- ・**SAMPLE_A.cob** と **SAMPLE_B.cob** が同一実行ファイルの共有ライブラリ(たとえば、**SAMPLE_A.so**)となる場合も、データの受け渡しが可能である。
- ・ **[Enterprise for Windows] [Standard]** 翻訳時に定義および参照を指定し、1 つにする方法
外部項目の定義および参照を指定する場合、外部項目の実体を定義するプログラムに対しては **-Sx** オプションを、外部項目の実体を参照するプログラムに対しては **-Si** オプションを翻訳時に指定する。

<生成例>

外部項目をプログラム A(**SAMPLE_A.cob**)で定義するとした場合、プログラム A の翻訳時には **-Sx** オプションを、外部項目を参照するプログラム B (**SAMPLE_B.cob**) の翻訳時には、**-Si** オプションを指定する。そして、生成されたオブジェクトファイル (**SAMPLE_A.obj**, **SAMPLE_B.obj**) を結合する。

```
cob -M -c -Sx SAMPLE_A.cob
cob -c -Si SAMPLE_B.cob
cob -M SAMPLE_A.obj SAMPLE_B.obj
```

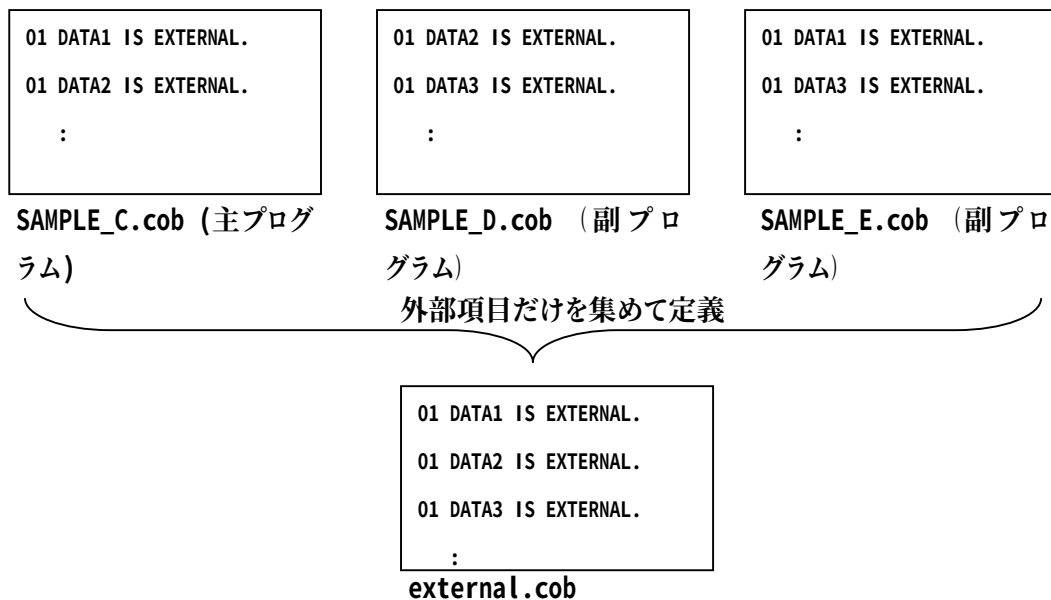
[Standard] または

```
cob -Mw -c -Sx SAMPLE_A.cob
cob -c -Si SAMPLE_B.cob
cob -Mw SAMPLE_A.obj SAMPLE_B.obj
```

しかし、このように個々のプログラムの翻訳に **-Sx -Si** オプションをつけるようにすることは運用が煩雑である。このような場合は、外部項目のデータ定義だけを集めたプログラムを作成し、このプログラムだけを **-Sx** 指定で翻訳して実体を定義する。そして、元のプログラムはすべて **-Si** 指定で翻

訳すればよい。

以降に例を示す。



上記は、元々SAMPLE_C.cob, SAMPLE_D.cob, SAMPLE_E.cob の中に定義されていた外部項目 DATA1, DATA2, DATA3 を集めて external.cob というプログラムを作成したものである。こうすると、下記のように、external.cob のみは -Sx を指定して翻訳し、元のプログラム SAMPLE_C.cob, SAMPLE_D.cob, SAMPLE_E.cob は、一律 -Si を指定して翻訳すればよい。

```
cob -c -Sx external.cob
```

```
cob -M -Si SAMPLE_C.cob SAMPLE_D.cob SAMPLE_E.cob external.obj
```

- ・ [Enterprise for Linux] 翻訳時に定義および参照を指定し、1つにする方法

外部項目の定義および参照を指定する場合、外部項目の実体を定義するプログラムに対しては -Sx オプションを、外部項目の実体を参照するプログラムに対しては -Si オプションを翻訳時に指定する。

<生成例>

外部項目をプログラム A (SAMPLE_A.cob) で定義するとした場合、プログラム A の翻訳時には -Sx オプションを、外部項目を参照するプログラム B (SAMPLE_B.cob) の翻訳時には、-Si オプションを指定する。そして、生成されたオブジェクトファイル (SAMPLE_A.o, SAMPLE_B.o) を結合する。

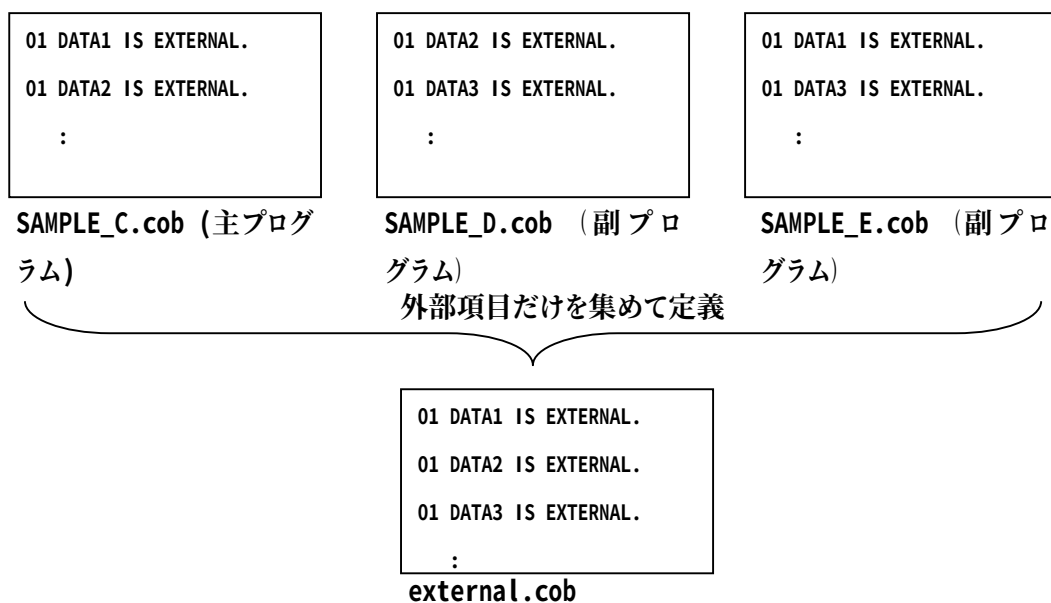
```
cob -M -c -Sx SAMPLE_A.cob
```

```
cob -c -Si SAMPLE_B.cob
```

```
cob -M SAMPLE_A.o SAMPLE_B.o
```

しかし、このように個々のプログラムの翻訳に **-Sx -Si** オプションをつけるようにすることは運用が煩雑である。このような場合は、外部項目のデータ定義だけを集めたプログラムを作成し、このプログラムだけを **-Sx** 指定で翻訳して実体を定義する。そして、元のプログラムはすべて **-Si** 指定で翻訳すればよい。

以降に例を示す。



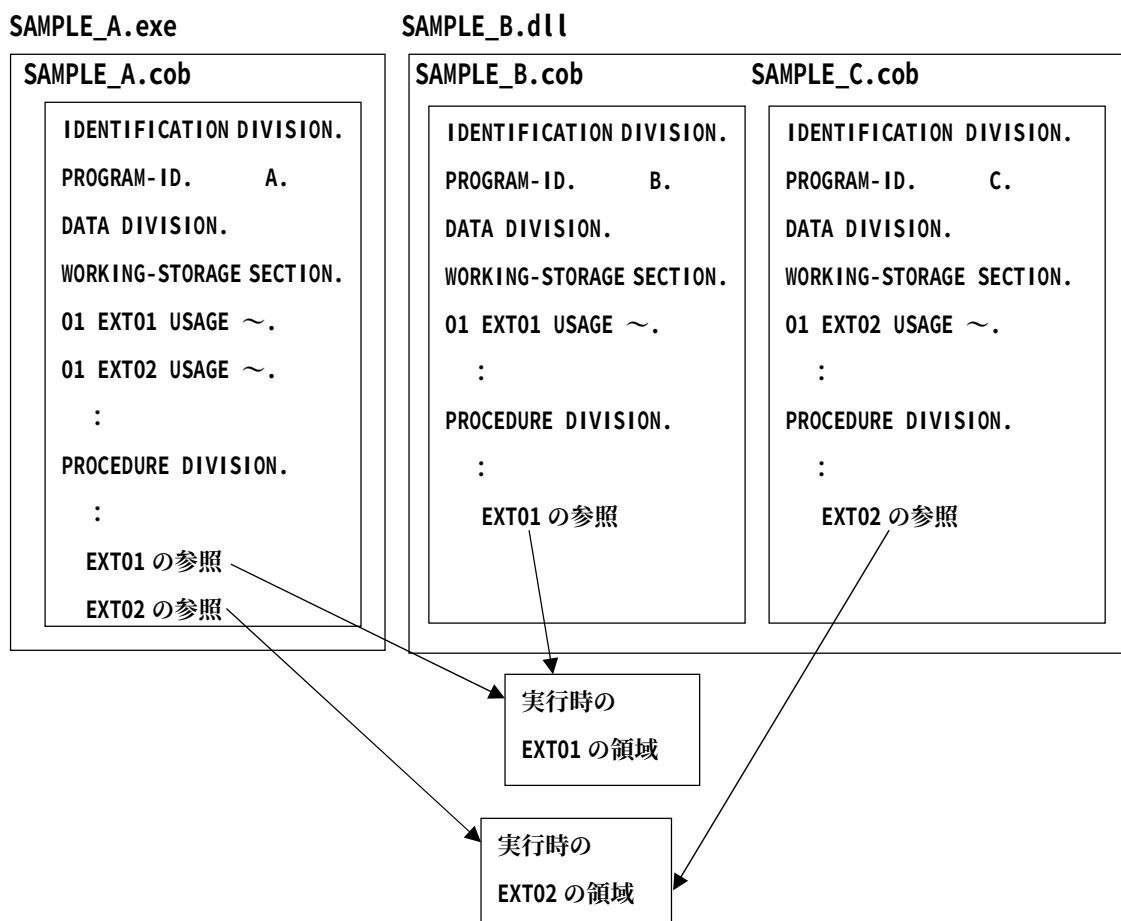
上記は、元々 **SAMPLE_C.cob**, **SAMPLE_D.cob**, **SAMPLE_E.cob** の中に定義されていた外部項目 **DATA1, DATA2, DATA3** を集めて **external.cob** というプログラムを作成したものである。こうすると、下記のように、**external.cob** のみは **-Sx** を指定して翻訳し、元のプログラム **SAMPLE_C.cob**, **SAMPLE_D.cob**, **SAMPLE_E.cob** は、一律 **-Si** を指定して翻訳すればよい。

```
cob -c -Sx external.cob
```

```
cob -M -Si SAMPLE_C.cob SAMPLE_D.cob SAMPLE_E.cob external.o
```

(2) [Enterprise for Windows] [Standard] 実行単位全体で受け渡し可能とする場合

COBOL プログラムの構成として、外部定義とするデータ宣言と、外部参照とするデータ宣言をファイル単位に分けてコーディングすること。



<生成例>

COBOL 翻訳時に、外部定義とするデータ宣言を行ったファイルと、外部参照とするデータ宣言を行ったファイルの翻訳単位を分ける。

なお、DLL の生成方法については、「7.11. 動的リンク機能」を参照。

```
> cob -G -Sx SAMPLE_B.cob SAMPLE_C.cob
```

```
> cob -M -Si SAMPLE_A.cob SAMPLE_B.lib
```

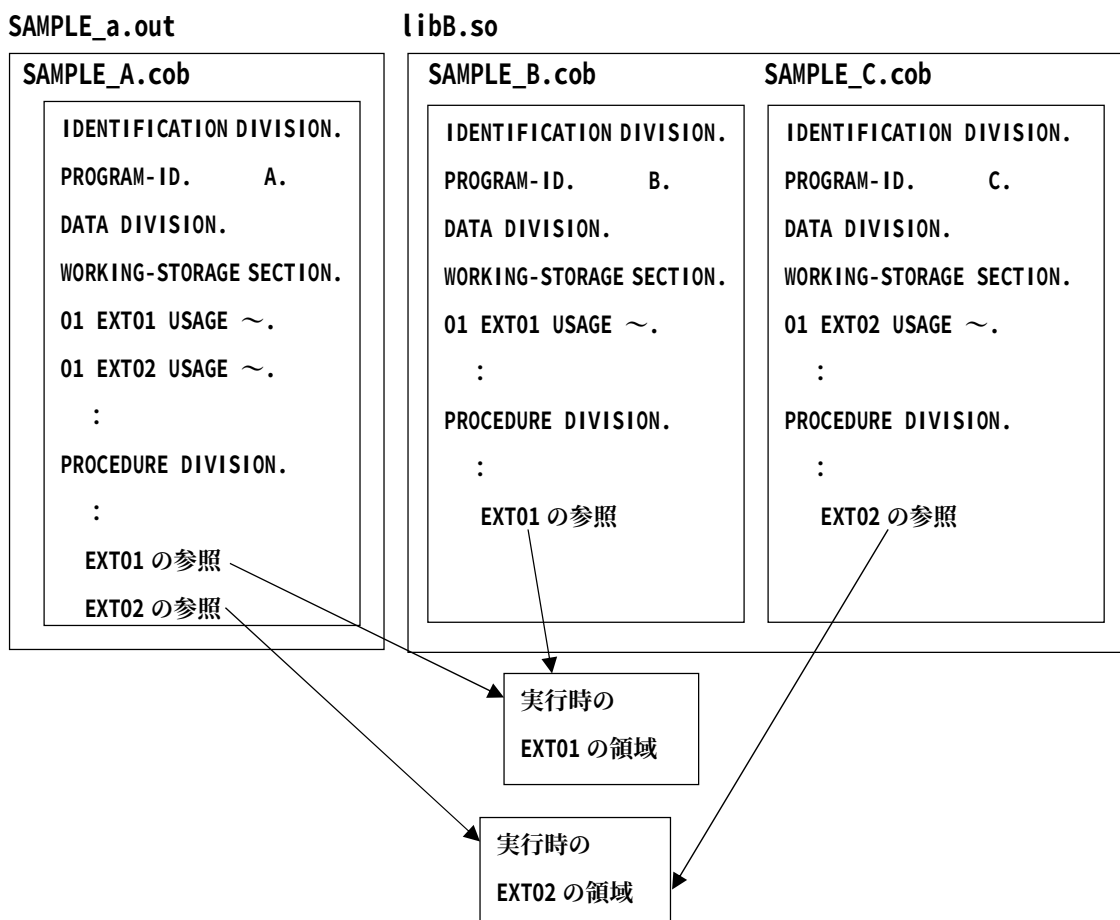
生成された **SAMPLE_A.exe** と **SAMPLE_B.dll**(**SAMPLE_B.obj**, **SAMPLE_C.obj**) のすべてで **EXTERNAL** 句指定データの受け渡しが可能である。

注意

- SAMPLE_A.cob, SAMPLE_B.cob, SAMPLE_C.cob を1つの実行ファイルとした場合もデータの受け渡しは可能。
- SAMPLE_A.cob, SAMPLE_B.cob, SAMPLE_C.cob がそれぞれ別々の実行ファイルの場合もデータの受け渡しは可能。
- SAMPLE_A.cob が定義で SAMPLE_B.cob, SAMPLE_C.cob が参照の場合、ビルド順の考慮が必要である。
- 実行ファイルの生成には、必ずインポートライブラリのリンクが必要である。よって、DLL の動的呼び出しを利用している場合も、インポートライブラリのリンクが必要なため、DLL のロードタイミングが変わる。
- 他言語で定義を行い、COBOL プログラムがすべて参照となる場合、初期値は他言語側の定義に従う。

(3) [Enterprise for Linux] 実行単位全体で受け渡し可能とする場合

COBOL プログラムの構成として、外部定義とするデータ宣言と、外部参照とするデータ宣言をファイル単位に分けてコーディングすること。



＜生成例＞

COBOL 翻訳時に、外部定義とするデータ宣言を行ったファイルと、外部参照とするデータ宣言を行ったファイルの翻訳単位を分ける。

なお、共有ライブラリ生成方法については、「7.11. 動的リンク機能」を参照。

```
> cob -o libB.so -G -Sx SAMPLE_B.cob SAMPLE_C.cob
```

```
> cob -o SAMPLE_a.out -M -Si SAMPLE_A.cob -L. -lB
```

生成された SAMPLE_a.out と SAMPLE_libB.so(SAMPLE_B.o, SAMPLE_C.o) のすべてで EXTERNAL 句指定データの受け渡しが可能である。

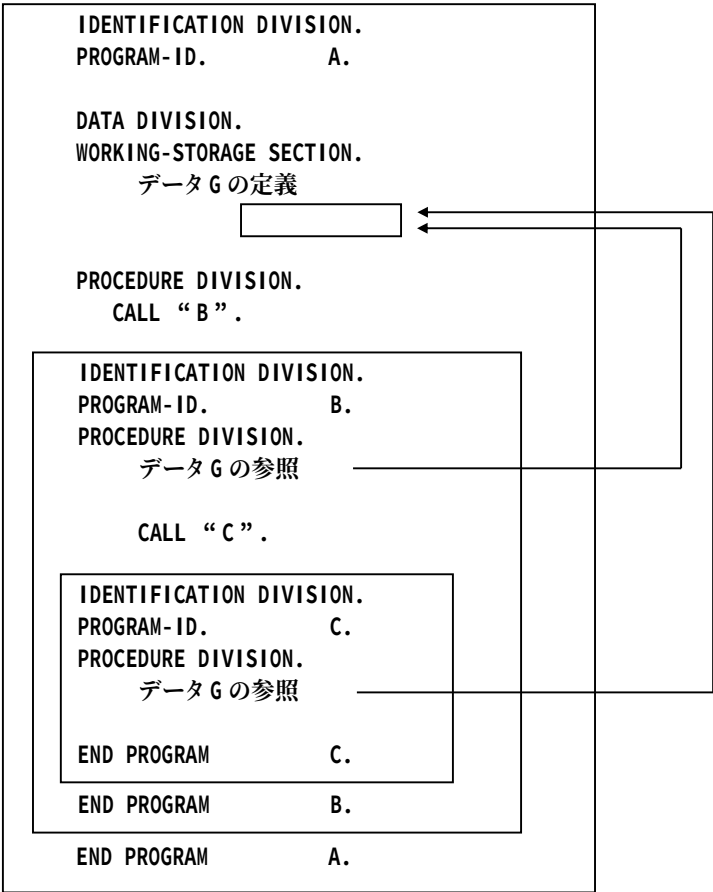
注意

- SAMPLE_A.cob, SAMPLE_B.cob, SAMPLE_C.cob を1つの実行ファイルとした場合もデータの受け渡しは可能。
- SAMPLE_A.cob, SAMPLE_B.cob, SAMPLE_C.cob がそれぞれ別々の実行ファイルの場合もデータの受け渡しは可能。
- 実行ファイルの生成には必ず共有ライブラリのリンクが必要です。

7.5.3 全域名による受け渡し

メインプログラムとサブプログラムの両方で参照できる全域名で、データを受け渡すことができる。ただし、この方法で受け渡すことができるのは、全域名となったファイル名、条件名、データ名、レコード名、指標名および特殊レジスタ LINAGE-COUNTER に限られる。全域名による受け渡しの例を図 7-10 に示す。

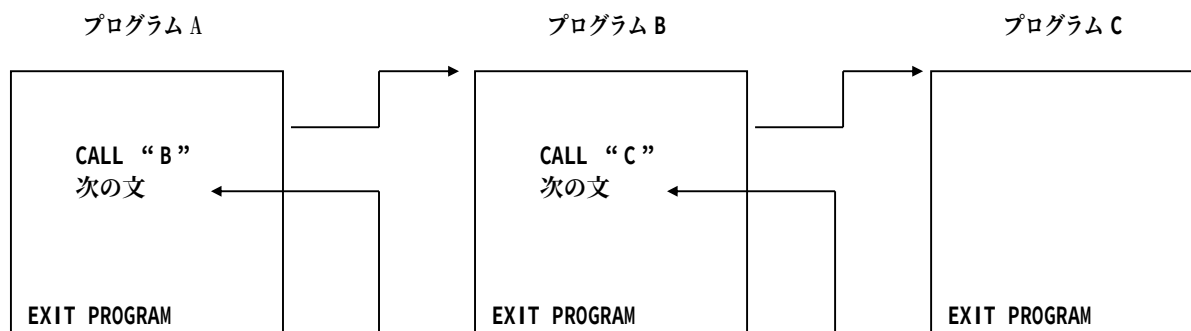
図 7-10 全域名による受け渡しの例



7.6 サブプログラムからの復帰

サブプログラムからメインプログラムへ復帰する文として `EXIT PROGRAM` 文がある。`EXIT PROGRAM` 文を実行すると、このサブプログラムを呼び出した `CALL` 文の直後の文に制御が移る。ただし、`CALL` 文で呼び出されていないプログラムで `EXIT PROGRAM` 文を実行した場合は、その `EXIT PROGRAM` 文の直後の文に制御が移る。`EXIT PROGRAM` 文によるサブプログラムからの復帰の例を図 7-11 に示す。

図 7-11 サブプログラムからの復帰の例



7.7 宣言手続きの起動

入出力文の実行結果が誤りのときに起動される宣言手続きを記述する文として **USE** 文がある。**USE** 文には **GLOBAL** 句を書くことができる。**GLOBAL** 句が書かれた宣言手続きは、この宣言手続きを定義したプログラムだけでなくそのプログラムに含まれるプログラムからも起動され得る。宣言手続きの起動の例を図 7-12、図 7-13、図 7-14 に示す。

図 7-12 宣言手続きの起動の例 1

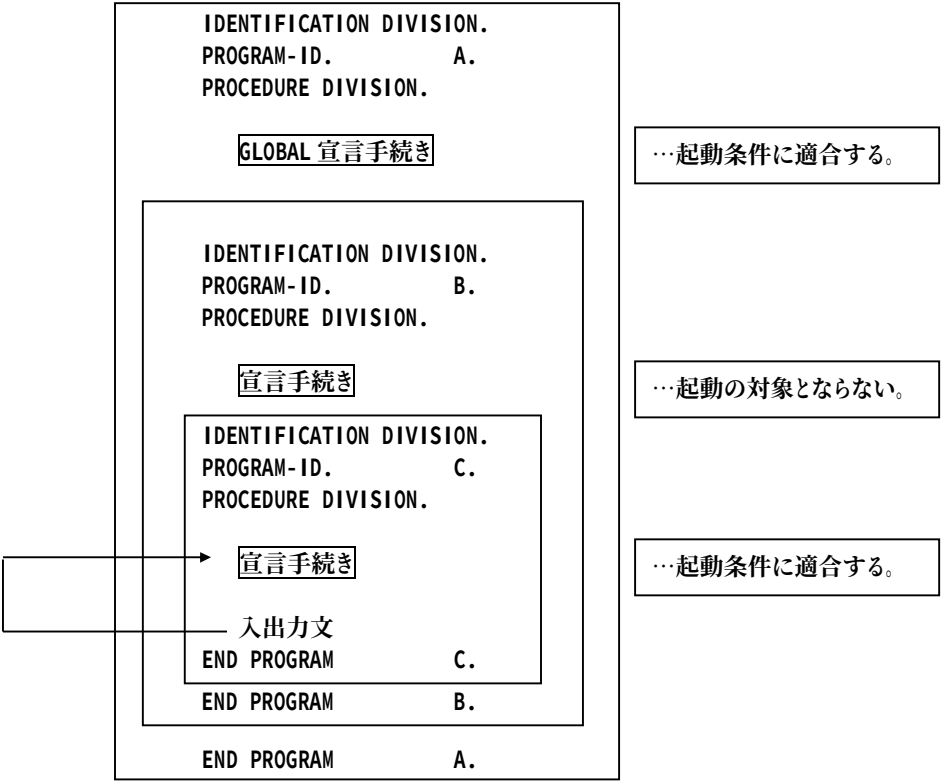


図 7-12 の例で、プログラム B の宣言手続きには **GLOBAL** 句がないので、プログラム C の入出力文から起動される対象とならない。起動され得る宣言手続きが複数ある場合は、入れ子の包含関係において入出力文を記述したプログラムに最も近いプログラムの宣言手続きが選ばれるので、本例ではプログラム C の宣言手続きが起動される。

図 7-13 宣言手続きの起動の例 2

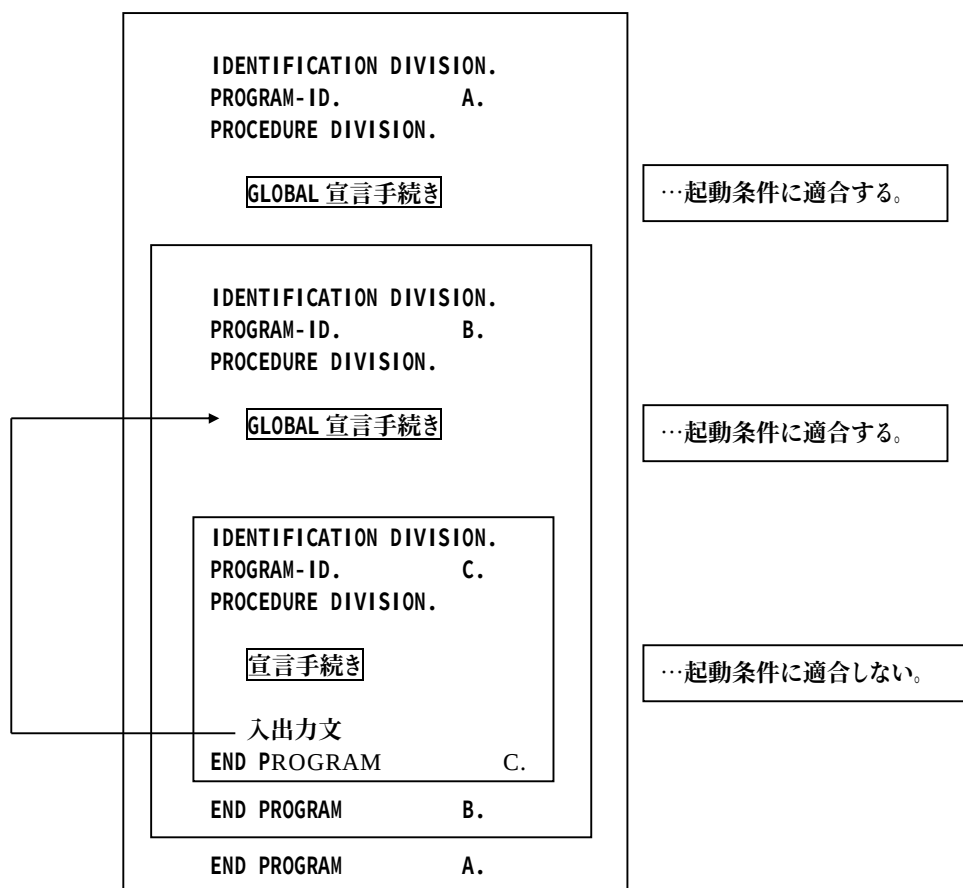


図 7-13 の例で、プログラム C の宣言手続きは起動条件に適合していない場合、プログラム C の入出力文から起動される対象とならない。起動され得る宣言手続きが複数ある場合は、入れ子の包含関係において入出力文を記述したプログラムに最も近いプログラムの宣言手続きが選ばれるので、本例ではプログラム B の宣言手続きが起動される。

図 7-14 宣言手続きの起動の例 3

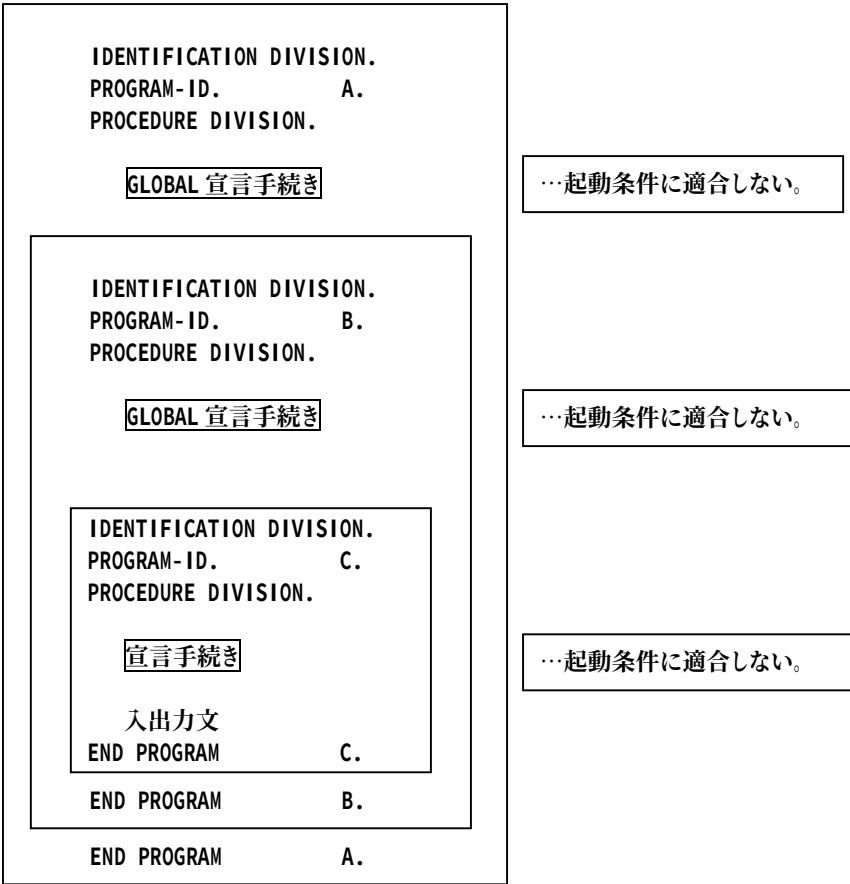


図 7-14 の例では、どのプログラムの宣言手続きにも起動条件が適合していない場合、プログラム C の入出力文から起動される対象とならない。起動され得る宣言手続きが何も実行しないので、本例では起動される宣言手続きがない。

7.8 プログラムの初期化

プログラムが実行単位中で最初に呼び出されたときの状態をプログラムの初期状態といい、次の 4 つの状態が保証される。

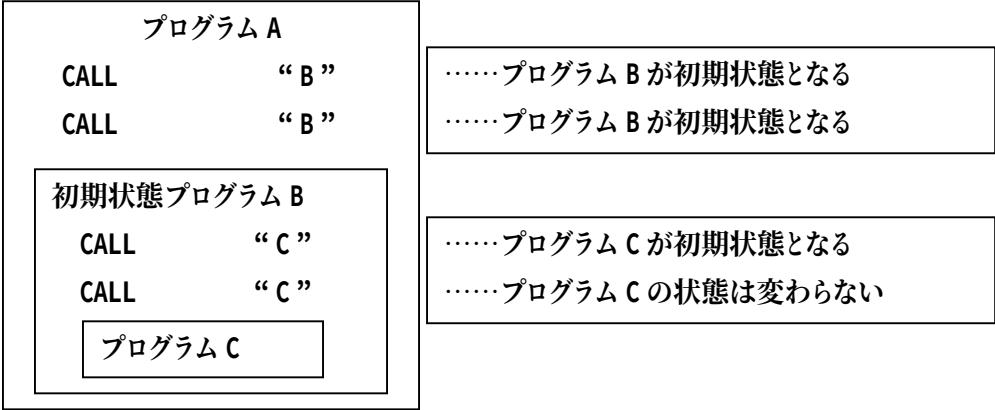
- (1) 作業場所節の VALUE 句を書いた項目がその値で初期化された状態。
- (2) 内部ファイル結合子をもつファイルが開かれていない状態。
- (3) PERFORM 文の復帰機構が初期化された状態。
- (4) ALTER 文で変更される前の GO TO 文の状態。

プログラムの任意の時点で初期状態にするためには、そのプログラムを初期状態プログラムとして定義するか、そのプログラムを指定した CANCEL 文を実行する。

7.8.1 初期状態プログラム

手続き部の PROGRAM-ID 段落に INITIAL 句を書いたプログラムを初期状態プログラムという。初期状態プログラムは、他のプログラムから呼び出される度に初期状態となる。また、初期状態プログラムに含まれるプログラムは、初期状態プログラムが呼び出された後でそのプログラムが最初に呼び出されたときに初期状態となる。初期状態プログラムの呼び出しの例を図 7-15 に示す。

図 7-15 初期状態プログラムの呼び出し例



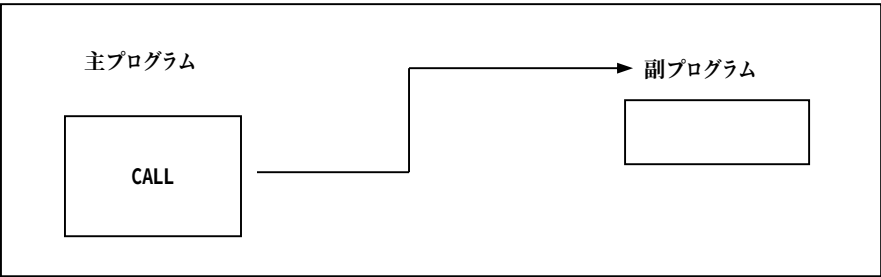
7.9 CANCEL 文

指定したプログラムを取り消す文として CANCEL 文がある。取り消されたプログラムは、次に呼び出されるとき初期状態となる。CANCEL 文で指定したプログラムを含むプログラムも同時に取り消される。CANCEL 文で取り消すことができるのは、CALL 文で呼び出すことができるプログラムに限られる。

7.10 静的リンクと動的リンク

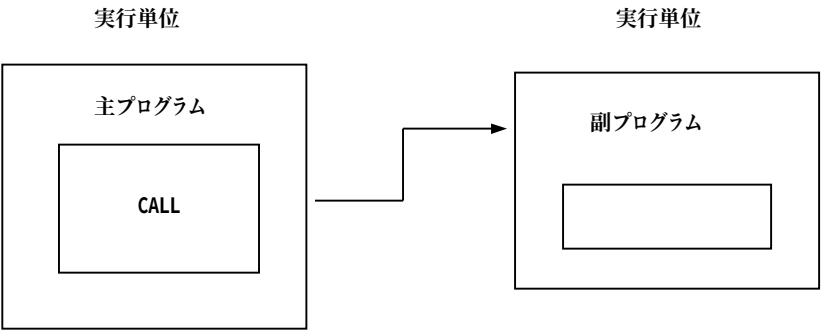
7.4.3 の外部プログラム呼び出しで、メインプログラムとサブプログラムが同一実行単位に含まれている場合を静的リンクという。静的リンクの例を図 7-16 に示す。

図 7-16 静的リンクの例



また、サブプログラムが独立した実行単位になり、実行時に主記憶にロードされる場合を動的リンクという。動的リンクの例を図 7-17 に示す。

図 7-17 動的リンクの例



ここで、呼び出し時間の短縮が重要であれば、静的リンクを適用する。逆に主記憶の占有量を少なくすることが重要であれば、動的リンクを適用する。

外部プログラムの呼び出しが静的呼び出し／動的呼び出しのどちらになるかは、CALL 文の書き方によって決まる。静的呼び出し／動的呼び出しの決定規則を表 7-4 に示す。

表 7-4 静的呼び出し／動的呼び出しの決定規則

定数指定の CALL 文	CALL 一意名
外部プログラムの呼び出しが静的呼び出しとなる	外部プログラムの呼び出しが動的呼び出しとなる

なお、-CY オプションを指定しなければ、静的リンクした外部プログラムは動的呼び出しすることができない。

7.11 動的リンク機能

7.11.1 Windows 版の動的リンク機能

7.11.1.1 COBOL プログラムの DLL 化

COBOL で記述したサブプログラムを翻訳時に-G オプションを指定することにより DLL にすることができる。DLL は複数のプロセスで共有することができるため、システム全体のメモリ効率を上げることができる。また、サブプログラムを修正するときも、DLL を再構築するだけでよく、実行形式を再リンクする必要はない。

1つのプログラムを1つのDLLにすることができるだけでなく、複数のプログラムを1つのDLLにすることができる。このオプションを指定すると、DLL ファイル(.dll)を生成すると同時にインポートライブラリ(.lib) およびエクスポートファイル(.exp)を生成する。

インポートライブラリは COBOL の実行ファイル生成時、動的リンクをするために必要なファイルである。エクスポートファイルは DLL 生成時に生成されるが通常は使用せず、相互インポートを行っている場合に使用する。

-o オプションを指定しなければ、ダイナミックリンクライブラリファイルのファイル名は既定の名前（注 1）となる。

DLL の詳細については、C/C++コンパイラ（注 2）のドキュメントを参照。

注意

（注 1）コマンドラインで指定した最初のファイル(原始プログラムファイルまたはオブジェクトファイル)のベース名に、拡張子 .dll .lib .exp を付けた名前

（注 2）Visual C++

7.11.1.2 DLL を使用する実行ファイル生成例

以降に DLL の生成方法を説明し、また DLL を用いた実行形式の生成方法および実行形式を説明する。

(1) プログラムの記述例

DLL 中のサブプログラムを呼び出すには、CALL(呼ぶ)文を使用する。

(a) プログラムで呼ばれるサブプログラムを英数字定数で指定した場合

```
CALL "SUB1" USING ~
```

(b) プログラムで呼ばれるサブプログラムを一意名で指定した場合

一意名によるサブプログラムの呼び出しを行うとき、-CY オプションを指定しないなら、DLL に格納しなければならない。

```
MOVE "SUB1" TO SUB1.  
CALL SUB1 USING ~
```

(2) DLL の生成方法

(a) DLL 内のオブジェクトがすべて COBOL の場合

① DLL 生成

```
cob -G SUB1.COB SUB2.COB
```

これを実行すると、インポートライブラリ SUB1.LIB、DLL ファイル SUB1.DLL およびエクスポートファイル SUB1.EXP が生成される。SUB1.DLL には、SUB1.COB と SUB2.COB の 2 つのプログラムが含まれる。

生成する DLL ファイル名を変更したい場合は、

```
cob -G -oSUBPROG SUB1.COB SUB2.COB
```

を実行する。この場合、生成されるファイル名はそれぞれ SUBPROG.LIB／SUBPROG.DLL／SUBPROG.EXP となる。

② 実行ファイル生成

1. プログラムで呼ばれるサブプログラムを英数字定数で指定した場合

```
cob -M MAIN.COB SUBPROG.LIB  
[Standard] または  
cob -Mw MAIN.COB SUBPROG.LIB
```

インポートライブラリを指定しリンクすることで、SUBPROG.DLL を動的リンクする実行ファイル MAIN.EXE が生成される。

2. プログラムで呼ばれるサブプログラムを一意名で指定した場合

```
cob -M MAIN.COB  
[Standard] または
```

```
cob -Mw MAIN.COB
```

インポートライブラリの指定は不要である。

(b) DLL 内のオブジェクトに COBOL と C が混在する場合

① オブジェクト生成

```
cob -G -c SUB1.COB SUB2.COB
cl /LD /c SUB3.C
```

COBOL のプログラムと C のプログラムをそれぞれ翻訳し、DLL 用のオブジェクトを生成する。C のプログラムを COBOL コンパイラで同時に翻訳した場合、COBOL の中間ソースと同じオプションで翻訳されるため、C のプログラムは必ず C/C++コンパイラで翻訳する。

注意

```
cob -G SUB1.COB SUB2.COB SUB3.C
```

COBOL のプログラムと C のプログラムを COBOL コンパイラで同時に翻訳した場合、C のプログラムの翻訳には次のオプションを指定するので注意が必要である。

```
cl /c /MD /DA_CBL_64 /I "(COBOL インストールディレクトリ)\include" SUB3.C
```

② DLL 生成

```
cob -G SUB1.OBJ SUB2.OBJ SUB3.OBJ
```

オブジェクトファイルを指定し、リンクを行う。この場合、生成されるファイル名はそれぞれ SUB1.LIB／SUB1.DLL／SUB1.EXP となる。

③ 実行ファイル生成

「(a) DLL 内のオブジェクトがすべて COBOL の場合」の「② 実行ファイル生成」と同様である。

7.11.1.3 DLL を使用する実行ファイルの実行方法

(1) プログラムで活性化される側の実行時要素を英数字定数で指定した場合

実行ファイル生成時にリンクしたインポートライブラリに対応する DLL を「7.11.1.4 DLL の検索」で記述されている検索パスに配置する。

その後、生成した実行ファイルを起動する。

(2) プログラムで活性化される側の実行時要素を一意名で指定した場合

プログラムを実行する環境で、環境変数 COB_LIBRARY に DLL 名をフルパスで指定するか、ファイル名のみを指定する。ファイル名のみを指定する場合、DLL を「7.11.1.4 DLL の検索」で記述されている検索パスに配置する。

環境変数 COB_LIBRARY を指定した後、実行形式を起動する。

```
set COB_LIBRARY=C:\sample\SUBPROG.DLL;
MAIN.EXE
```

複数の DLL を指定する場合は、区切り記号(注 1)を用いる。

```
set COB_LIBRARY=C:\sample\sample.dll;C:\sample2\SUBPROG.DLL;
```

注意 (注 1) 区切り記号は[;]

7.11.1.4 DLL の検索

DLL の検索は、ロード時に次の順に行われる。検索時に DLL が見つからない場合は、アプリケーションエラーになる。

(1) プログラムで活性化される側の実行時要素を英数字定数で指定した場合

- 実行可能ファイルのあるディレクトリ
- Windows のシステムディレクトリ (%SystemRoot%\SYSTEM32)
- Windows ディレクトリ (%SystemRoot%)
- カレントディレクトリ
- 環境変数 PATH に登録されているディレクトリ

(2) プログラムで活性化される側の実行時要素を一意名で指定した場合

- 環境変数 COB_LIBRARY で指定した順番

なお、ファイル名のみ指定した場合、その DLL の特定は、「(1) プログラムで活性化される側の実行時要素を英数字定数で指定した場合」に記載の順番に行われる。

DLL が見つからない場合、後続の DLL の検索は行わず実行時エラーが発生する。

ただし、環境変数 COB_LIBERR=CONTINUE が指定されている場合は、見つからない DLL は無視し、後続の DLL の検索を継続する。

注意 一意名で指定した外部プログラムがメインプログラムと同一の実行単位に含まれる場合、-CY オプションを指定すると、DLL の検索は行われず、メインプログラムと同一の実行単位の外部プログラムが呼び出される。

7.11.1.5 注意事項

動的リンク機能を使用するにあたって、次のような注意事項がある。

- (1) 複数の COBOL プログラムが DLL とそれ以外とに別れ、かつ同名のグローバルデータ(EXTERNAL 句付きデータ)を有する場合、そのデータは別々の実体となるためデータ共有は行なえない。
- (2) SELECT 句に EXTERNAL 句が指定されたファイルは、各 DLL 間で共有可能となる。FD 句に EXTERNAL 句が指定されたファイルは、SELECT 句に EXTERNAL 句が指定されたものと同様の範囲のみ各 DLL 間で共有可能である。
- (3) 従来製品および本製品の旧バージョンの COBOL コンパイラで生成した 32 ビットコードの DLL を本コンパイラでリンクすることはできない。また、本コンパイラで生成した DLL を従来製品および本製品の旧バージョンの COBOL コンパイラでリンクすることはできない。

7.11.2 Linux 版の動的リンク機能

7.11.2.1 COBOL プログラムの共有ライブラリ化

COBOL で記述したサブプログラムを翻訳時に -G オプションを指定することにより共有ライブラリにすることができる。共有ライブラリは複数のプロセスで共有することができるため、システム全体のメモリ効率を上げることができる。また、サブプログラムを修正するときも、共有ライブラリを再構築するだけでよく、実行形式を再リンクする必要はない。

1つのプログラムを1つの共有ライブラリにすることができるだけでなく、複数のプログラムを1つの共有ライブラリにすることができる。このオプションを指定すると、共有ライブラリファイルを生成する。

-o オプションを指定しなければ、共有ライブラリのファイル名は既定の名前 **a.out** となる。共有ライブラリの詳細については、gcc(1)のドキュメントを参照。

7.11.2.2 共有ライブラリを使用する実行ファイル生成例

以降に共有ライブラリの生成方法を説明し、また共有ライブラリを用いた実行形式の生成方法および実行形式を説明する。

(1) プログラムの記述例

共有ライブラリ中のサブプログラムを呼び出すには、CALL(呼ぶ)文を使用する。

(a) プログラムで呼ばれるサブプログラムを英数字定数で指定した場合

```
CALL "SUB1" USING ~
```

(b) プログラムで呼ばれるサブプログラムを一意名で指定した場合

一意名によるサブプログラムの呼び出しを行うとき、-CY オプションを指定しないなら、共有ライブラリに格納しなければならない。

```
MOVE "SUB1" TO SUB1.  
CALL SUB1 USING ~
```

(2) 共有ライブラリの生成方法

(a) 共有ライブラリ内のオブジェクトがすべて COBOL の場合

① 共有ライブラリ生成

1. 英数字定数指定で呼び出されるサブプログラムを含む場合

```
cob -G -olibSUB1.so SUB1.cob SUB2.cob
```

共有ライブラリの名前は **lib** で始まる名称にする必要がある。

2. 一意名指定で呼び出されるサブプログラムのみの場合

```
cob -G -oSUBPROG.so SUB1.cob SUB2.cob
```

共有ライブラリの名前は **lib** で始まる名前でもよい。

-o オプションを指定しない場合、生成される共有ライブラリの名前は **a.out** となる。

② 実行ファイル生成

1. プログラムで呼ばれるサブプログラムを英数字定数で指定した場合

```
cob -M MAIN.cob -Wl,-lSUB1,-L/
```

リンクにカレントディレクトリにある共有ライブラリ **libSUB1.so** をリンクすることを示している。

2. プログラムで呼ばれるサブプログラムを一意名で指定した場合

```
cob -M MAIN.cob
```

実行可能プログラム作成時にリンクする共有ライブラリを指定する必要はない。

(b) 共有ライブラリ内のオブジェクトに COBOL と C が混在する場合

① オブジェクト生成

```
cob -G -c SUB1.cob SUB2.cob
gcc -fPIC -c SUB3.c
```

COBOL のプログラムと C のプログラムをそれぞれ翻訳し、共有ライブラリ用のオブジェクトを生成する。C のプログラムを COBOL コンパイラで同時に翻訳した場合、COBOL の中間ソースと同じオプションで翻訳されるため、C のプログラムは必ず C/C++コンパイラで翻訳する。

注意

```
cob -G SUB1.cob SUB2.cob SUB3.c
```

COBOL のプログラムと C のプログラムを COBOL コンパイラで同時に翻訳した場合、C のプログラムの翻訳には次のオプションを指定するので注意が必要である。

```
gcc -c -fPIC -DA_CBL_LINUX -DA_CBL_64 -I (COBOL インストールディレクトリ)/include SUB3.c
```

②共有ライブラリ生成

```
cob -olibSUB1.so -G SUB1.o SUB2.o SUB3.o
```

オブジェクトファイルを指定し、リンクを行う。この場合、生成されるファイル名は **libSUB1.so** となる。

生成されるファイル名は「(a) 共有ライブラリ内のオブジェクトがすべて COBOL の場合」の「①共有ライブラリ生成」に記載のとおり、呼び出し方法にあった名前をつける必要がある。

③実行ファイル生成

「(a) 共有ライブラリ内のオブジェクトがすべて COBOL の場合」の「② 実行ファイル生成」と同様である。

7.11.2.3 共有ライブラリを使用する実行ファイルの実行方法

(1) プログラムで活性化される側の実行時要素を英数字定数で指定した場合

実行ファイル生成時にリンクしたインポートライブラリに対応する共有ライブラリを「7.11.2.4 共有ライブラリの検索」の(1)の検索パスに配置する。

その後、生成した実行ファイルを起動する。

(2) プログラムで活性化される側の実行時要素を一意名で指定した場合

プログラムを実行する環境で、環境変数 `COB_LIBRARY` に共有ライブラリ名をフルパスで指定するか、ファイル名のみを指定する。ファイル名のみを指定する場合、共有ライブラリを「7.11.2.4 共有ライブラリの検索」の(1)の検索パスに配置する。

環境変数 `COB_LIBRARY` を指定した後、実行形式を起動する。

```
export COB_LIBRARY=/sample/SUBPROG.so
a.out
```

複数の共有ライブラリを指定する場合は、区切り記号コロン(:) を用いる。

① 共有ライブラリのフルパスを指定する方法

```
export COB_LIBRARY=/sample1/sample1.so:/sample2/sample2.so
```

② 共有ライブラリのファイル名のみ指定する方法

(OS 標準のライブラリ検索パスと組み合わせる)

```
export LD_LIBRARY_PATH=/sample1:/sample2
export COB_LIBRARY=sample1.so:sample2.so
```

7.11.2.4 共有ライブラリの検索

共有ライブラリの検索は、ロード時に次の順に行われる。検索時に共有ライブラリが見つからない場合は、アプリケーションエラーになる。

(1) プログラムで活性化される側の実行時要素を英数字定数で指定した場合

共有ライブラリの検索順序は次のとおりである。

- ユーザの `LD_LIBRARY_PATH` 環境変数内のコロンで区切られたディレクトリリスト
- `/etc/ld.so.cache` 内に指定されたライブラリリスト
- `/lib`, 次に `/usr/lib`

なお、プログラム翻訳時にリンカオプション「`-Wl, arglist`」を指定した場合は、指定順に共有ライブラリを検索する。

注意

XMLパーサ用ライブラリの検索順序も上記と同じである。

(2) プログラムで活性化される側の実行時要素を一意名で指定した場合

- 環境変数 `COB_LIBRARY` で指定した順番

なお、ファイル名のみ指定した場合、その共有ライブラリの特定は、「(1) プログラムで活性化される側の実行時要素を英数字定数で指定した場合」の規則に準じる。

共有ライブラリが見つからない場合、後続の共有ライブラリの検索は行わず実行時エラーが発生する。

ただし、環境変数COB_LIBERR=CONTINUEが指定されている場合は、見つからない共有ライブラリは無視し、後続の共有ライブラリの検索を継続する。

注意

一意名で指定した外部プログラムがメインプログラムと同一の実行単位に含まれる場合、-CY オプションを指定すると、共有ライブラリの検索は行われず、メインプログラムと同一の実行単位の外部プログラムが呼び出される。

7.11.2.5 注意事項

動的リンク機能を使用するにあたって、次のような注意事項がある。

- (1) 複数の COBOL プログラムが共有ライブラリとそれ以外とに別れ、かつ同名のグローバルデータ(EXTERNAL 句付きデータ)を有する場合、そのデータは別々の実体となるためデータ共有は行なえない。
- (2) SELECT 句に EXTERNAL 句が指定されたファイルは、各共有ライブラリ間で共有可能となる。FD 句に EXTERNAL 句が指定されたファイルは、SELECT 句に EXTERNAL 句が指定されたものと同様の範囲のみ各共有ライブラリ間で共有可能である。
- (3) 既存の COBOL コンパイラで生成した共有ライブラリを本コンパイラでリンクすることはできない。また、本コンパイラで生成した共有ライブラリを既存の COBOL コンパイラでリンクすることはできない。
- (4) 実行可能プログラムに静的リンクした外部プログラムを一意名呼び出しするためには、cob コマンドで実行可能プログラム作成時(リンクまで実行時)に-CY オプションを指定する必要がある。実行可能プログラムの作成(リンク)をgcc(1)コマンドで行う場合は-rdynamic オプションを指定する。ld(1)コマンドで行う場合は-export-dynamic オプションを指定する。
- (5) CALL 一意名の呼び出しが意図しない動作となる場合は、同名の別プログラムが呼び出されている可能性がある。

まずは、そのプログラムが呼び出されたとき、プログラム内で確実に実行される手続きの先頭に DISPLAY 文を埋め込み、その表示が行われているかを確認するなど修正動作が行われることを確認する。修正動作が行われている場合、意図したプログラムが呼び出されている。修正動作が行われていない場合、同名の別プログラムが呼び出されている可能性があるため、以下の確認を行う。

- ・ 環境変数 COB_LIBRARY に指定したライブラリの中に同名のプログラムを含むものがないか

- 環境変数 COB_LIBRARY にパス指定なしにライブラリを指定している場合、環境変数 LD_LIBRARY_PATH に指定したパス中に同名のライブラリがないか、またその中に同名のプログラムを含むものがないか
- (4)のいずれかの方法で実行可能プログラムに静的リンクした外部プログラムを一意名呼び出しできるように指定している場合、実行可能プログラム内に同名のプログラムがないか

備考

動的呼び出し可能になるように作成した実行可能プログラムや共有ライブラリは、COBOL プログラムだけでなく、内部で利用している他の関数も動的呼び出し可能となる。このためこれら関数と同名の COBOL プログラムがあると、意図せず内部で利用している関数が呼び出される可能性がある。OS 標準の nm(1) コマンド等を使って実行可能プログラムや共有ライブラリの関数名を確認し、呼び出そうとしているプログラム名と同名の関数が存在しないことを確認する必要がある。

7.12 C/C++プログラムとの連絡

COBOL のプログラム、関数、メソッドから、C/C++ の関数を CALL 文で呼び出すことができる。また、そのときの戻り値(関数値)を受け取ることもできる。

さらに、C/C++ の関数から、関数呼び出しにより COBOL で作成したプログラムおよび関数を呼び出すことができる。

なお、COBOL で作成したクラスを C++ から利用する場合、「9 章 オブジェクト指向機能」を参照。

ただし、他言語プログラムの呼び出しおよび他言語プログラムからの呼び出しを行うアプリケーションを生成する場合には、外部手続き呼び出し規約を次のようにしなければならない。

[Enterprise for Windows] [Standard]

`__fastcall`

[Enterprise for Linux]

x64 System V ABI (Application Binary Interface)

上記以外の呼び出し規約で作成した外部手続きを呼び出す場合は、COBOL と他言語の間に C/C++ でラップ関数を用意する必要がある。

また、データのやりとりをする場合には、データの内部表現が一致していなければならない。

本コンパイラでは、呼び出しができる言語処理系として、次のものを規定している。

その他の言語およびバージョンについては、保証のかぎりではない。

[Enterprise for Windows] [Standard]

Microsoft 社製 Visual C++ 2015

Microsoft 社製 Visual C++ 2017

Microsoft 社製 Visual C++ 2019

Microsoft 社製 Visual C++ 2022

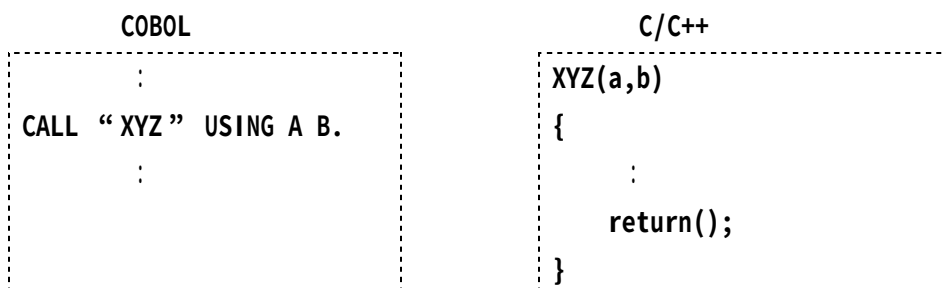
[Enterprise for Linux] gcc

ここでは COBOL と C/C++ をリンクする上での制限および注意事項について説明する。

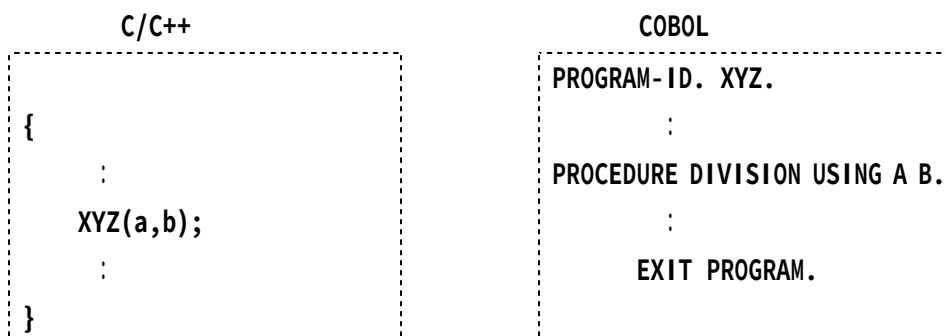
7.12.1 制御の移行方法

COBOL の CALL 文, C/C++ の関数呼び出しによって COBOL と C/C++ のリンクが可能となる。

(1) COBOL から C/C++ の呼び出し



(2) C/C++ から COBOL の呼び出し



COBOL から C/C++ の呼び出しにおいては, CALL 文で英小文字の定数を指定することでその名前に英小文字を含む関数を呼び出すことができる。

C/C++ から COBOL の呼び出しにおいては, 英小文字を含むプログラム名および利用者定義関数名を指定することができない。なぜならば, プログラム名段落または関数名段落にて, プログラム名

または関数名を英小文字で指定しても、対応する英大文字と等価とみなされるためである。ただし、**-Cs** オプションを指定した場合は、英小文字で指定したプログラム名または利用者定義関数名を英小文字のまま使用できる。また、PROGRAM-ID 段落でプログラム名に AS 定数を指定し、AS 定数中に英小文字を指定することでも英小文字のまま使用できる。

CALL 文で 31 文字を超える定数を指定しても、先頭 31 文字だけが有効である。

PROGRAM-ID. abcdef.	ABCDEF と指定したのと同じ。
PROGRAM-ID. A1234567890123456789012345678901.	致命的エラーであり、オブジェクトは生成されない。
CALL “ A1234567890123456789012345678901 ”	呼び出す関数名は、 A123456789012345678901234567890 と なる。

31 文字を超えるプログラム名で呼び出す場合は、呼び出されるプログラム名には AS 定数を指定しなければならない。

7.12.2 引数の引き渡し方法

COBOL と C/C++との引数の引き渡しについて特に注意しなければならない点がある。

COBOL コンパイラと C/C++コンパイラの引数の引き渡し方法は次のとおりである。

① COBOL コンパイラの引数の引き渡し方法

- CALL ～ BY REFERENCE 句指定時
指定した一意名のアドレスが渡される。
- CALL ～ BY CONTENT 句指定時
指定した一意名が、コンパイラが生成した同一桁数、同一属性の領域に転記されその生成領域のアドレスが渡される。
- CALL ～ BY VALUE 句指定時

指 定	動 作
一意名	指定した一意名の値が渡される
ADDRESS OF 一意名	指定した一意名のアドレスが渡される
NULL	NULL 値が渡される
LENGTH OF 一意名	指定した一意名の長さが COMP-2 項目として渡される

② C/C++コンパイラの引数の引き渡し方法

C/C++コンパイラでの引数の引き渡しおよび受け取りは、配列データの場合はアドレス、配列を除くデータの場合は値である。

前記から、引数の引き渡しに関しては次のような注意が必要である。

③ COBOL から C/C++への引数の引き渡し

- アドレス渡しの場合

(CALL ~ BY REFERENCE 句, CALL ~ BY CONTENT 句,

CALL ~ BY VALUE 句の ADDRESS OF 指定, CALL ~ BY VALUE 句の NULL 指定)

受け取り側である C/C++側は、引数が配列以外のデータの場合、ポインタで受け取る必要がある。

C/C++側で配列以外のデータで受け取る方法

COBOL

```
01 A COMP-1. *> 2 バイト 2 進
01 B COMP-2. *> 4 バイト 2 進
01 C PIC X.  *> 1 バイト英数字

CALL "XYZ" USING BY REFERENCE A B C.
```

C/C++

```
XYZ(a,b,c)
short  *a;
int    *b;
char   *c;
```

C/C++プログラムにて配列データで受け取る方法

COBOL

```
01 A PIC 9(2). *> 外部 10 進(2 バイト)
01 C PIC X(2). *> 2 バイト英数字

CALL "XYZ" USING BY REFERENCE A C.
```

C/C++

```
XYZ(a,c)
char  a[2];
char  c[2];
```

- 値渡しの場合

(CALL ~ BY VALUE 句の一意名指定, CALL ~ BY VALUE 句の LENGTH OF 指定)

受け取り側である C/C++側は, 配列以外, ポインタ以外のデータで受け取る必要がある。

COBOL <pre> 01 A COMP-1. 2 バイト 2 進 01 B PIC 9(2). 外部 10 進(2 バイト) 01 C PIC X(2). 2 バイト英数字 CALL "XYZ" USING BY VALUE A BY VALUE LENGTH OF B BY VALUE C.</pre>	C/C++ <pre> XYZ(a,b,c) short a; int b; struct c1{ char c2[2]; } c;</pre>

④ C から COBOL への引数の引き渡し

- アドレス受け取り (PROCEDURE ～ BY REFERENCE) の場合
呼び出し側である C 側は、配列以外のデータの場合にはポインタを渡す必要がある。

C/C++ <pre> short a; int b; char c[2]; XYZ(&a,&b,&c);</pre>	COBOL <pre> PROGRAM-ID. XYZ. LINKAGE SECTION. 01 A COMP-1. 01 B COMP-2. 01 C PIC X(2).</pre>

PROCEDURE DIVISION USING BY REFERENCE A B C.

- ・ 値受け取りはできない。

7.12.3 引数の受け渡し詳細

COBOL と C/C++ 間でのデータの対応を示す。

7.12.3.1 COBOL から C/C++ を呼び出す場合

(1) CALL ～ BY REFERENCE 句, CALL ～ BY CONTENT 句指定時

COBOL	C/C++
USAGE COMP-1 USAGE BINARY PIC S9(n) 注(8) USAGE COMP-5 PIC S9(n) [Enterprise] ただし n=3,4 [Standard] ただし n=1～4	short *
USAGE COMP-2 USAGE BINARY PIC S9(n) 注(8) USAGE COMP-5 PIC S9(n) ただし $5 \leq n \leq 9$	[Enterprise for Windows] [Standard] int * / long * [Enterprise for Linux] int *
USAGE BINARY PIC S9(n) 注(8) USAGE COMP-5 PIC S9(n) ただし $n \geq 10$	[Enterprise for Windows] [Standard] long long * [Enterprise for Linux] long *
USAGE BINARY PIC 9(n) 注(8) USAGE COMP-5 PIC 9(n) [Enterprise] ただし n=3,4 [Standard] ただし n=1～4	unsigned short *

USAGE BINARY PIC 9(n) 注(8) USAGE COMP-5 PIC 9(n) ただし $5 \leq n \leq 9$	[Enterprise for Windows] [Standard] unsigned int * / unsigned long * [Enterprise for Linux] unsigned int *
USAGE BINARY PIC 9(n) 注(8) USAGE COMP-5 PIC 9(n) ただし $n \geq 10$	[Enterprise for Windows] [Standard] unsigned long long * [Enterprise for Linux] unsigned long *
USAGE DISPLAY PIC S9(n) n=1 注(6) USAGE BINARY PIC S9(m) 注(8) USAGE COMP-5 PIC S9(m) ただし m=1,2	char *
USAGE DISPLAY ただし PIC X(n) n=1 または PIC 9(n) n=1 注(6) USAGE BINARY PIC 9(m) 注(8) USAGE COMP-5 PIC 9(m) ただし m=1,2	unsigned char*
—	float * / double *
USAGE DISPLAY ただし PIC X(n) n≠1 または PIC 9(n) n≠1 または PIC S9(n) n≥1 注(7)	unsigned char A[n]
USAGE COMP-3 PIC S9(n)	char A[n/2+1] 注(1)
USAGE POINTER	ポインタ
USAGE OBJECT REFERENCE	class A_CBL_Object* 注(2)
集団項目	構造体 注(3) struct {...} *
表	配列 注(4)

(2) CALL ～ BY VALUE 句指定

- ① 一意名指定(CALL ～) 注(4)

COBOL	C/C++
USAGE COMP-1 USAGE BINARY PIC S9(n) 注(8) USAGE COMP-5 PIC S9(n) [Enterprise] ただし n=3,4 [Standard] ただし n=1~4	short
USAGE COMP-2 USAGE BINARY PIC S9(n) 注(8) USAGE COMP-5 PIC S9(n) ただし $5 \leq n \leq 9$	[Enterprise for Windows] [Standard] int / long [Enterprise for Linux] int
USAGE BINARY PIC S9(n) 注(8) USAGE COMP-5 PIC S9(n) ただし $n \geq 10$	[Enterprise for Windows] [Standard] long long [Enterprise for Linux] long
USAGE BINARY PIC 9(n) 注(8) USAGE COMP-5 PIC 9(n) [Enterprise] ただし n=3,4 [Standard] ただし n=1~4	unsigned short
USAGE BINARY PIC 9(n) 注(8) USAGE COMP-5 PIC 9(n) ただし $5 \leq n \leq 9$	[Enterprise for Windows] [Standard] unsigned int / unsigned long [Enterprise for Linux] unsigned int
USAGE BINARY PIC 9(n) 注(8) USAGE COMP-5 PIC 9(n) ただし $n \geq 10$	[Enterprise for Windows] [Standard] unsigned long long [Enterprise for Linux] unsigned long
USAGE DISPLAY PIC S9(n) n=1 注(6) USAGE BINARY PIC S9(m) 注(8) USAGE COMP-5 PIC S9(m) ただし m=1,2	char
USAGE DISPLAY	unsigned char

ただし PIC X(n) n=1 または PIC 9(n) n=1 注(6) USAGE BINARY PIC 9(m) 注(8) USAGE COMP-5 PIC 9(m) ただし m=1,2	
—	float / double
USAGE DISPLAY ただし PIC X(n) n≠1 または PIC 9(n) n≠1 または PIC S9(n) n≥1 注(7)	構造体 struct { char A[n]; }
USAGE COMP-3 PIC S9(n)	構造体 注(1) struct { char A[n/2+1]; }
USAGE POINTER	ポインタ
USAGE OBJECT REFERNECE	class A_CBL_Object 注(2)
集団項目 表	構造体 注(3) struct {...} 注(4)

② ADDRESS OF 指定(CALL ～)

COBOL	C/C++
ADDRESS OF 一意名	BY REFERENCE 句, BY CONTENT 句 指定の場合と同じ

③ NULL 指定(CALL ～)

COBOL	C/C++
NULL	ポインタ class A_CBL_Object 注(2)

④ LENGTH OF 指定(CALL ～)

COBOL	C
LENGTH OF 一意名	int / long / long long

7.12.3.2 C/C++ から COBOL を呼び出す場合

(1) PROCEDURE ～ BY REFERENCE 句指定時

C/C++	COBOL
short *	USAGE COMP-1 USAGE BINARY PIC S9(4) 注(8) USAGE COMP-5 PIC S9(4)
[Enterprise for Windows] [Standard] int * / long * [Enterprise for Linux] int *	USAGE COMP-2 USAGE BINARY PIC S9(9) 注(8) USAGE COMP-5 PIC S9(9)
[Enterprise for Windows] [Standard] long long * [Enterprise for Linux] long *	USAGE BINARY PIC S9(18) 注(8) USAGE COMP-5 PIC S9(18)
unsigned short *	USAGE BINARY PIC 9(4) 注(8) USAGE COMP-5 PIC 9(4)
[Enterprise for Windows] [Standard] unsigned int * / unsigned long * [Enterprise for Linux] unsigned int *	USAGE BINARY PIC 9(9) 注(8) USAGE COMP-5 PIC 9(9)
[Enterprise for Windows] [Standard] unsigned long long * [Enterprise for Linux] unsigned long *	USAGE BINARY PIC 9(18) 注(8) USAGE COMP-5 PIC 9(18)
char *	USAGE DISPLAY PIC S9(1) 注(6) USAGE BINARY PIC S9(2) 注(8) USAGE COMP-5 PIC S9(2)
unsigned char *	USAGE DISPLAY PIC X(1) USAGE DISPLAY PIC 9(1) USAGE BINARY PIC 9(2) 注(8)

	USAGE COMP-5 PIC 9(2)
float * / double *	—
char A[n]	USAGE DISPLAY PIC S9(n) $n \geq 1$ 注(7)
unsigned char A[n]	USAGE DISPLAY ただし PIC X(n) $n \neq 1$ または PIC 9(n) $n \neq 1$
char A[n/2+1] 注(1)	USAGE COMP-3 PIC S9(n)
ポインタ	USAGE POINTER
構造体 注(3) struct {...} *	集団項目
配列 注(4)	表

(2) PROCEDURE ～ BY VALUE 句指定時

C/C++	COBOL
short	USAGE COMP-1 USAGE BINARY PIC S9(4) 注(7) USAGE COMP-5 PIC S9(4)
[Enterprise for Windows] [Standard] int / long [Enterprise for Linux] int	USAGE COMP-2 USAGE BINARY PIC S9(9) 注(7) USAGE COMP-5 PIC S9(9)
[Enterprise for Windows] [Standard] long long [Enterprise for Linux] long	USAGE BINARY PIC S9(18) 注(7) USAGE COMP-5 PIC S9(18)
unsigned short	USAGE BINARY PIC 9(4) 注(7) USAGE COMP-5 PIC 9(4)
[Enterprise for Windows] [Standard]	USAGE BINARY PIC 9(9) 注(7) USAGE COMP-5 PIC 9(9)

unsigned int / unsigned long [Enterprise for Linux] unsigned int	
[Enterprise for Windows] [Standard] unsigned long long [Enterprise for Linux] unsigned long	USAGE BINARY PIC 9(18) 注(7) USAGE COMP-5 PIC 9(18)
char	USAGE DISPLAY PIC S9(1) 注(5) USAGE BINARY PIC S9(2) 注(7) USAGE COMP-5 PIC S9(2)
unsigned char	USAGE DISPLAY PIC X(1) USAGE DISPLAY PIC 9(1) USAGE BINARY PIC 9(2) 注(7) USAGE COMP-5 PIC 9(2)
float / double	—
char A[n]	USAGE DISPLAY PIC S9(n) $n \geq 1$ 注(6)
unsigned char A[n]	USAGE DISPLAY ただし PIC X(n) $n \neq 1$ または PIC 9(n) $n \neq 1$
char A[n/2+1] 注(1)	USAGE COMP-3 PIC S9(n)
ポインタ	USAGE POINTER
class A_CBL_Object 注(2)	USAGE OBJECT REFERENCE
構造体 注(3) struct {...}	集団項目
配列 注(4)	表

注意

- 注(1) COBOL の 10 進データを扱うためには、C 側で COBOL のデータ表現形式を考慮したデータ操作が必要である。
- 注(2) オブジェクト参照項目 (USAGE OBJECT REFERENCE) は、指定されたクラス名に従う。また、BY VALUE 句および BY CONTENT 句指定の場合のみ指定できる。
- 注(3) 集団項目と構造体の相違
集団項目と構造体を対応させるためには、表に示したとおりのデータの相違の

他にデータの境界の相違を考慮する必要がある。

COBOL コンパイラでは集団項目内の各データ領域は連続して割り付けられるが、C/C++コンパイラではデータの型によって境界が決定されるため、SYNC 指定またはダミーデータを詰めることによって COBOL 記述で C のデータの境界に合わせる必要がある。

表 7-5 データの境界

境界	COBOL	C
8 バイト	SYNC 指定あり USAGE BINARY PIC S9(n) USAGE COMP-5 PIC S9(n) ただし $n \geq 10$	[Enterprise for Windows] [Standard] long long [Enterprise for Linux] long
	ポインタ	ポインタ
4 バイト	SYNC 指定あり COMP-2	[Enterprise for Windows] [Standard] int / long [Enterprise for Linux] int
2 バイト	SYNC 指定あり COMP-1	short
1 バイト	その他	その他

注(4)表／配列の構造

表／配列の構造は、COBOL と C は同じである。ただし、集団項目および構造体を要素とする表／配列を対応させるためには、要素の相違を考慮する必要がある。C/C++コンパイラでは、要素の境界がその要素に含まれるデータの最大境界で割り付けられるため、要素と要素の間にダミーが空く場合があり COBOL 記述でダミーを詰める必要がある。

注(5)BY VALUE 一意名指定時、一意名に可変長項目を指定した場合は、引き渡される値は可変だが、項目長はその可変長項目の最大長である。したがって、C 側で可変長項

目を受け取る場合にはその可変長項目の最大長分の引数を指定する必要がある。

注(6)SIGN(符号)句に SEPARATE CHARACTER の指定がない場合のみ適応する。なお、COBOL の 10 進データを扱うためには、C 側で COBOL のデータ表現形式を考慮したデータ操作が必要である。

注(7)SIGN (符号)句に SEPARATE CHARACTER の指定がある場合は $n \geq 1$, SEPARATE CHARACTER の指定がない場合は $n \geq 2$ となる。なお、COBOL の 10 進データを扱うためには、C 側で COBOL のデータ表現形式を考慮したデータ操作が必要である。

注(8)BINARY 項目の場合、低位アドレスのバイト位置が上位桁を表すようにメモリの正順で格納されるため、C 側で COBOL のデータ表現形式を考慮したデータ操作が必要である。

7.12.4 戻り値の受け取り方法

COBOL から C を呼び出して関数の戻り値を受け取る場合、戻り値は RETURNING 句によって受け取ることができる。また、C/C++ からプログラムの外部公開名を指定して戻り値を受け取ることもできる。

戻り値のデータの対応を以下に示す。

COBOL			C/C++
項目	PICTURE 句	USAGE 句	
[Enterprise] 1 バイト 2 進数 (符号あり)	S9(n) n=1,2	BINARY COMP-5	char
[Enterprise] 1 バイト 2 進数 (符号なし)	9(n) n=1,2	BINARY COMP-5	unsigned char
2 バイト 2 進数 (符号あり)	S9(n) n=3,4 [Standard] n=1,2 の場合もある。	COMP-1 BINARY COMP-5	short
2 バイト 2 進数 (符号なし)	9(n) n=3,4 [Standard] n=1,2 の場合もある。	BINARY COMP-5	unsigned short
4 バイト 2 進数		COMP-2	int

(符号あり)	S9(n) $5 \leq n \leq 9$	BINARY COMP-5	
4 バイト 2 進数 (符号なし)	9(n) $5 \leq n \leq 9$	BINARY COMP-5	unsigned int
8 バイト 2 進数 (符号あり)	S9(n) $10 \leq n \leq 18$	BINARY COMP-5	[Enterprise for Windows] [Standard] long long [Enterprise for Linux] long
8 バイト 2 進数 (符号なし)	9(n) $10 \leq n \leq 18$	BINARY COMP-5	[Enterprise for Windows] [Standard] unsigned long long [Enterprise for Linux] unsigned long
外部 10 進数	9(n) $n \neq 1$ S9(n)	DISPLAY	struct { char A[n]; }
	S9(n) $n=1$ 9(n)		char
内部 10 進数	S9(n)	COMP-3	struct { char A[n/2+1]; }
ポインタ		POINTER	void*
指標データ項目		INDEX	
オブジェクト 参照項目		OBJECT REFERENCE	class A_CBL_Object*

7.12.5 外部データによるデータ授受

COBOL と C で同一の外部データを参照する場合は、英小文字を含む外部名を指定することができない。なぜならば、EXTERNAL 句を記述した外部データ名として英小文字を指定しても対応する英大文字と等価とみなされるためである。ただし、-Cs オプションを指定した場合は、EXTERNAL 句を記述した外部データ名として英小文字を指定すると英小文字のまま使用できる。

例 外部データ ERRNO の参照

COBOL	C
<pre>01 ERRNO COMP-2 EXTERNAL. : :</pre>	<pre>extern int ERRNO; ACCERR { ERRNO = 1; return(); }</pre>

7.12.6 その他

COBOL と C のリンクにおいてメインプログラムは、COBOL と C のどちらであってもよい。

[Enterprise] COBOL がメインプログラムの場合は、-M オプションを指定して翻訳する。

[Standard] COBOL がメインプログラムの場合は、-M または -Mw オプションを指定して翻訳する。

7.13 C/C++プログラムとのリンク時の注意事項

本コンパイラ では、COBOL アプリケーションの終了時、COBOL アプリケーションが使用しているリソース(ファイル、メモリ等)を安全に解放するための終了処理が組み込まれている。

COBOL アプリケーションの実行時には、不正なメモリアクセスやゼロ除算などによるハードウェア例外や不正10 進データ検出等の実行時エラーによるソフトウェア例外が発生する可能性があり、その場合にも前出の終了処理を行った後にCOBOL アプリケーションを終了させる。

[Enterprise for Windows] [Standard] 特にハードウェア例外発生時の終了処理は、オペレーティングシステムの構造化例外処理を使用して実現しており、かつCOBOL プログラムがメインプログラムである必要がある。

構造化例外処理についての詳細は、Visual C++の説明書を熟読されたい。

以上より、他言語とのリンク時には、この終了処理を意識したプログラムを作成する必要がある。

[Enterprise for Linux] 特にハードウェア例外発生時の終了処理は、オペレーティングシステムのシグナルを使用して実現しており、かつCOBOL プログラムがメインプログラムである必要がある。

シグナルについての詳細は、gccの説明書を熟読されたい。

以上より、他言語とのリンク時には、この終了処理を意識したプログラムを作成する必要がある。

7.13.1 全般的な注意

- (1) [Enterprise for Windows] [Standard] C プログラムとのリンクで C プログラムのソースファイル名の拡張子が.CPP の場合は C プログラムの定義に「extern "C"」を記述する。「extern "C"」の詳細については Visual C++の説明書を参照。

[Enterprise for Linux] C プログラムとのリンクで C プログラムのソースファイル名の拡張子が.CPP の場合は C プログラムの定義に「extern "C"」を記述する。「extern "C"」の詳細については gcc の説明書を参照。

- (2) [Enterprise for Windows] [Standard] C プログラムは [DLL を使ったマルチスレッド] のオブジェクトとなるように、Visual C++コンパイラに /MD オプションを指定すること。詳細は Visual C++の説明書を参照。

- (3) [Enterprise for Windows] [Standard] クリティカルエラー(致命的 I/O エラー)について

- (a) COBOL がメインプログラムの場合

クリティカルエラーはCOBOL側で処理する。プロセス開始時にクリティカルエラーモードを設定する。

```
SetErrorMode(SEM_FAILCRITICALERRORS | SEM_NOOPENFILEERRORBOX);
```

- (b) COBOL から他言語を呼ぶ場合

クリティカルエラーはCOBOL側で処理する。したがって、他言語側でシステムエラーとして通知してほしい場合は、SetErrorMode(0)でCOBOL側での設定をクリアする必要がある。ただし、その際にSetErrorMode(0)から返されるクリティカルエラーモードを保存しておきCOBOL 側への復帰前にそのクリティカルエラーモードを復旧すること。

- (c) 他言語から COBOL が呼ばれる場合

クリティカルエラーはCOBOL側で処理する。呼び出されたCOBOL側の処理開始時にクリティカルエラーモードを設定し、他言語側への復帰前にクリティカルエラーモードを復旧する。

(d) マルチスレッド機能を使用する場合

呼び出されたCOBOL側でクリティカルエラーモードを設定しない。他言語側で設定されたクリティカルエラーモードで動作する。COBOLがメインプログラムの場合は(a)を参照。

(4) [Enterprise for Linux] 他言語プログラムとのリンク時、他言語プログラム側でストリーム (stdout/stderr/stdin) に対してワイド文字系のライブラリ関数は使用できない。

(5) [Enterprise for Windows] [Standard] 他言語モジュールを CALL 一意名呼び出しする場合、呼び出す関数はエクスポートしておかなければならない。「関数のエクスポート」の詳細については Visual C++の説明書を参照。

7.13.2 COBOL がメインプログラムの場合

(1) いかなる場合でも、COBOL から呼び出される他言語モジュールの中でプログラムを終了させてはならない。プログラムを終了させる必要が生じた場合、他言語側で必要な処理をすべて済ませた上で、必ず COBOL の STOP RUN 文を実行しなければならない。COBOL によって変更されたシステム状態からの復旧は STOP RUN 文において行われる。

この注意を守らない場合、プログラム終了後のシステムの安全性に関しては保証しない。

(2) [Standard] COBOL で作成した対話型アプリケーションのプログラムをメインプログラムとする場合、COBOL が暗黙的に提供しているウィンドウプロシージャを使用するため、他言語で作成したサブプログラム内では、以下の処理は行えない。

- Windows から通知されるメッセージに対する処理
- COBOL 提供のウィンドウプロシージャの処理を阻害するような処理 (メッセージ送信や COBOL メインウィンドウに対する入出力など)

(3) [Enterprise for Windows] シグナルハンドラの設定は有効となる。COBOL 側で COBOL に必要な終了処理を行うハンドラルーチンは登録されている。他言語側でハンドラルーチンを登録する場合は必ず次のことを厳守する必要がある。

- ハンドラルーチン内でアプリケーションを終了させない
- 戻り値を FALSE にする

[Standard] シグナルハンドラの設定は COBOL WS エミュレータ配下でないバッチ型アプリケーションで有効となる。COBOL 側で COBOL に必要な終了処理を行うハンドラルーチンは登録されている。他言語側でハンドラルーチンを登録する場合は必ず次のことを厳守する必要がある。

- ハンドラルーチン内でアプリケーションを終了させない
- 戻り値を FALSE にする

[Enterprise for Linux] シグナルハンドラの設定は有効となる。COBOL 側で COBOL に必要な終了処理を行うハンドラルーチンは登録されている。他言語側でハンドラルーチンを登録することはできない。COBOL が登録するシグナルの詳細は、「A.2.6 COBOL が登録するシグナル」を参照。

- (4) 正常, 異常を問わず, COBOL で作成したプログラムが終了する場合, 他言語モジュール側の終了処理を呼び出すことはない。

プログラム終了時に他言語側でどうしても行いたい終了処理がある場合, COBOLはその登録手段を提供する。

以降にサンプルコードを示す。サンプルコードはC 記述になっている。

```
#include "CSFINISH.h"
void Emergency_recovery(void);
void c_sub(void)
{
    /******
    /* 最初の文          */
    /******
    CSSETREGIST(Emergency_recovery);
    :
}
void Emergency_recovery(void)
{
    必要な終了処理
}
```

[CSSETREGIST 利用規約]

- 最初に呼び出される他言語モジュールの最初の文で登録する。
- 登録したモジュール内でプログラムを終了させてはならない。
- CSSETREGIST関数の引数にNULL('00H')を設定することで, 任意のタイミングで登録をキャンセルすることができる。
- 'CSFINISH.h'に必要な型定義が記述されているので, 利用されたい。
- 登録したモジュールからCOBOLのモジュールを呼び出してはならない。

- (5) [Enterprise for Windows] [Standard] COBOL から呼び出される他言語モジュールの中で, 動的

リンク機能を使用し生成した DLL を使用する場合、インポートライブラリをリンクし使用しなければならない。この注意を守らない場合、アプリケーションの動作は保証できない。

- (6) [Standard] ライブラリ CSGETWINHANDLE.LIB を使用することで、COBOL メインウィンドウのウィンドウハンドルを取得できるので、C 言語等により個別のカスタマイズを行うことができる。

[CSGETWINHANDLE利用規約]

- a. COBOLメインウィンドウのウィンドウハンドルが必要な場合、本関数を実行すること。
- b. CSGETWINHANDLE.LIBをCOBOL対話型アプリケーションのリンク時に指定すること。
- c. 'CSLNK.h'が提供されているので、利用されたい。また 'CSLNK.h'をインクルードする前に 'WINDOWS.h'を必ずインクルードすること。

7.13.3 COBOL がサブプログラムの場合

1. [Standard] 他言語で作成したプログラムをメインプログラムとする場合、ウィンドウプロシージャなどは他言語側で独自に作成する必要がある。また、COBOL で作成したサブプログラム内では、以下の機能制限がある。

- ・ 画面入出力

	COBOL WSエミュレータ配下 でない		COBOL WSエミュレータ配 下	
	バッチ型ア プリケーシ ョン	対話型アプ リケーショ ン	バッチ型ア プリケーシ ョン	対話型アプ リケーショ ン
小入出力 (呼び名：SYSIN/SYSOUT)	○	○	○	×
データ例外表示 (呼び名：SYSOUT)				
小入出力 (呼び名：CONSOLE/STDIN/ STDOUT/STDERR)	○	×	○	×
データ例外表示 (呼び名：CONSOLE/ STDOUT/STDERR)				
STOP定数	○	○	○	×
画面制御機能	×	×	×	×

- レジストリによるカスタマイズ
アプリケーションカスタマイザで既定値として登録されたプリンタ、ファイル、CSV、フォントの一部、サウンドの一部、その他の設定情報のみ有効となる。

2.他言語で作成したプログラムをメインプログラムとする場合、各種登録手順を提供する。
以降にサンプルコードを示す。サンプルコードはC 記述になっている。

- [Enterprise for Windows] [Standard] バッチ型アプリケーションの場合

```
#include <windows.h>
#include "CSHANDLER.h"
#include "CSFINISH.h"

void c_abort(int, int, char *);
extern void CBLMODULE(void); /* COBOLサブプログラム */

void
main(
    int argc,
    char **argv)
{
    CSSETABORT(c_abort);
    SetConsoleCtrlHandler((PHANDLER_ROUTINE)CSCBLHANDLER, TRUE);
    __try{
        CBLMODULE(); /* COBOL サブプログラム */
    }
    __finally{
        CSFINISH(); /* COBOL 側の終了処理 */
    }
}

void
c_abort(
    int cobolerrno,
    int cobolline,
    char *cobolprogid)
{
    他言語側で必要な終了処理

```



```

    CSFINISH();
}

```

- [Enterprise for Linux] バッチ型アプリケーションの場合

```

#include <signal.h>
#include "CSHANDLER.h"
#include "CSFINISH.h"

void c_abort(int, int, char *);
extern void CBLMODULE(void); /* COBOL サブプログラム */

```

```

void
main(
    int argc,
    char **argv)
{
    struct sigaction sigact;
    CSSETABORT(c_abort);

```

シグナル処理

```

    CBLMODULE(); /* COBOL サブプログラム */
    CSFINISH(); /* COBOL 側の終了処理 */
}

void
c_abort(
    int cobolerrno,
    int cobolline,
    char *cobolprogid)
{
    他言語側で必要な終了処理
    CSFINISH();
}

```

- [Standard] 対話型アプリケーションの場合
- ```

#include <WINDOWS.H>

```

```

#include "CSLNK.H"
#include "CSFINISH.H"

void c_abort(int, int, char *);
extern void CBLMODULE(void); /* COBOL サブプログラム */

static HWND hwnd;

int APIENTRY WinMain(
 HINSTANCE hInstance,
 HINSTANCE hPrevInstance,
 LPSTR lpCmdLine,
 int nCmdShow)
{
 ウィンドウクラスの登録

 hwnd = CreateWindow("SAMPLE", "SAMPLE",
 WS_OVERLAPPEDWINDOW | WS_HSCROLL | WS_VSCROLL,
 CW_USEDEFAULT, CW_USEDEFAULT,
 CW_USEDEFAULT, CW_USEDEFAULT,
 NULL, NULL, hInstance, NULL);

 ウィンドウの表示

 CSSETWINHANDLE(hwnd);

 CSSETABORT(c_abort);

 __try {
 メッセージループ
 }
 __finally{
 CSFINISH();

 他言語側で必要な終了処理
 }
}

```

```
}
```

```
void c_abort(
 int cobolerrno,
 int cobolline,
 char *cobolprogid)
{
 他言語側で必要な終了処理
 CSFINISH();
}
```

ウィンドウプロシジャー

```
{
 COBOL サブプログラムを呼び出すメッセージが通知された
 CBLMODULE(); /* COBOL サブプログラムの呼び出し */
}
```

- (1) COBOL がサブプログラムであるアプリケーションの場合、アプリケーションが正常終了および COBOL のサブプログラム内でハードウェア例外が発生した場合、COBOL が使用しているリソースの解放処理を行う終了処理が実行されないため、他言語側より呼び出せる COBOL 側の終了処理 (CSFINISH 関数) を提供しているので、必ず呼び出す必要がある。

**[CSFINISH利用規約]**

- a. 引数、返却値ともない。
- b. COBOLサブプログラムから呼び出してはならない。  
COBOLがメインプログラムである場合にも、呼び出しは不要である。
- c. CSFINISH 関数を呼び出してCOBOL側の終了処理を実行したあとでCOBOLサブプログラムを呼び出してはならない。
- d. 'CSFINISH.h'を提供しているので、利用されたい。

[Enterprise for Linux] COBOL 側の終了処理として、COBOL85 for IPF でサポートしている COBOL 側の終了処理 A\_CBL\_FINISH 関数が使用可能であるが、CSFINISH 関数との同時使用は不可である。なお、A\_CBL\_FINISH 関数の利用規約は CSFINISH 関数と同様である。

- (2) COBOL が実行時エラーで終了した場合に、他言語のメインプログラム側で何らかの終了処理を行う必要がある場合に備えて、COBOL は異常終了時に実行される他言語のメインプログラム側の終了処理登録手段 (CSSETABORT 関数) を提供しているので、必ず登録する必要がある。

**[CSSETABORT利用規約]**

- a. 他言語のメインプログラムでCOBOLサブモジュールの呼び出し前までに設定する。
- b. 他言語のメインプログラムは、COBOL異常終了時には速やかに他言語側で必要な終了処理およびCSFINISHの実行を行った上でプログラムを終了させる。COBOLサブモジュールに制御を戻すことはできない。
- c. CSSETABORT関数の引数にNULL('00H')を設定することで、任意のタイミングで登録をキャンセルすることができる。
- d. 本関数による処理登録が行われなかった場合、COBOLはCOBOL異常終了処理を行った後、プログラムを終了させる。
- e. 'CSFINISH.h'が提供されているので、利用されたい。
- f. 登録した他言語モジュールからCOBOLサブモジュールを呼び出す場合、最後にCSFINISHを実行する必要がある。
- g. 登録したモジュールがCOBOLサブモジュールの場合、COBOL異常終了処理を行った後、他言語サブモジュールを呼び出してCSFINISHを実行する必要がある。COBOL異常終了処理中にさらに実行時エラーが発生すると呼び出しが繰り返されることになるため、COBOL異常終了処理の前に、CSSETABORT(NULL)を実行する他言語サブモジュールを呼び出してCSSETABORT関数の登録をキャンセルするか、2回目の呼び出し時には何も

実行しないでCOBOLサブモジュールから処理を戻すこと。

COBOLから呼ばれるメインプログラム側のエラー処理ルーチンには、3つの引数が渡される。

- COBOL 実行時エラーメッセージ番号(int)
- COBOL 実行時エラー発生行番号(int)
- プログラム名(char \*)

- (3) 他言語で生成したバッチ型アプリケーションのプログラムから COBOL で生成したサブプログラムを呼ぶ場合、アプリケーションに対して強制終了のシグナル発行がおこなわれた場合の COBOL 側の終了処理(CSCBLHANDLER 関数)を提供しているので、必ず登録する必要がある。

[Enterprise for Linux] 他言語で生成したバッチ型アプリケーションのプログラムから COBOL で生成したサブプログラムを呼ぶ場合、アプリケーションに対して強制終了のシグナル発行がおこなわれた場合の COBOL 側の終了処理として他言語側でハンドラルーチンを登録しない場合は CSCBLHANDLER 関数を提供しているので必ず登録する必要がある。他言語側でハンドラルーチンを登録する場合は CSCBLABORT 関数を提供しているので必ず呼び出す必要がある。

#### [CSCBLHANDLER利用規約]

- a. [Enterprise for Windows] [Standard] 他言語のメインプログラムでCOBOLサブモジュールの呼び出し前までにSetConsoleCtrlHandler()にて登録する。  
[Enterprise for Linux] 他言語のメインプログラムでCOBOLサブモジュールの呼び出し前までにsigaction()にて登録する。
- b. [Enterprise for Windows] [Standard] その他に他言語側でハンドラルーチンを登録する場合は必ず次のことを厳守すること。
  - ハンドラルーチン内でアプリケーションを終了させない
  - 戻り値をFALSE にする

[Enterprise for Linux] CSCBLHANDLER関数内でプロセス終了を行うため、その他に他言語側でハンドラルーチンを登録することはできない。

その他に他言語側でハンドラルーチンを登録する場合は、CSCBLABORT関数を他言語側で呼び出す必要がある。

- c. [Standard] COBOL WSエミュレータ配下ではハンドラ設定は無効となる。
- d. 'CSHANDLER.h'が提供されているので、利用されたい。  
[Enterprise for Windows] [Standard] また、'CSHANDLER.h'をインクルードする前に、'WINDOWS.H'を必ずインクルードすること。

#### [CSCBLABORT利用規約]

- a. 引数はない。返却値は、異常終了時の終了コード(int)である。
- b. COBOLサブプログラムから呼び出してはならない。

COBOLがメインプログラムである場合にも、呼び出しは不要である。

- c. CSCBLABORT関数を呼び出してCOBOL側の終了処理を実行したあとでCOBOLサブプログラムを呼び出してはならない。
- d. CSCBLABORT関数では、プロセス終了は行わない。
- e. 'CSHANDLER.h' を提供しているので、利用されたい。

- (4) [Standard] 他言語で生成した対話型アプリケーションのプログラムから COBOL で生成したサブプログラムを呼ぶ場合、必ず他言語側で取得したウィンドウハンドルを COBOL サブプログラム側に通知する必要がある。COBOL ではウィンドウハンドルの通知手段 (CSSETWINHANDLE 関数) を提供しているので、必ず通知する必要がある。

#### [CSSETWINHANDLE利用規約]

- a. 他言語のメインプログラムでCOBOLサブモジュールの呼び出し前までに設定する。
- b. ウィンドウハンドルを持たないアプリケーションの場合には、CSSETWINHANDLE関数の引数にNULL( '00H ')を設定すること。
- c. 'CSLNK.h' が提供されているので、利用されたい。また 'CSLNK.h' をインクルードする前に 'WINDOWS.h' を必ずインクルードすること。

## 7.14 CALL 一意名呼び出し時の詳細エラー

[Enterprise for Windows] [Standard] CALL 一意名呼び出し実行時、「COB008 CALL で呼び出すエントリが見つかりません。」エラーが発生することがある。

本エラーは通常、以下の場合に発生する。

- 一意名に指定したサブプログラム名 (エントリ名) の途中で空白文字がある。
- 環境変数 COB\_LIBRARY に指定した DLL が存在しない。または、環境変数 COB\_LIBRARY に指定した DLL がロードする DLL が存在しない。

環境変数 COB\_LIBERR=CONTINUE を指定していない場合、一意名に指定したサブプログラムを含む DLL だけでなく、環境変数 COB\_LIBRARY で、その DLL より前に指定した DLL が存在しない場合にも発生する。

- 一意名に指定したサブプログラム名 (エントリ名) が実行可能プログラムにも環境変数 COB\_LIBRARY で指定した DLL にもない。その理由として以下が考えられる。
  - 一意名に指定したサブプログラム名 (エントリ名) が誤っている。
  - 一意名に指定したサブプログラム名 (エントリ名) が実行可能プログラム内にあるが、実行可能プログラム作成時にコンパイルオプション-CY を指定していない。
  - 環境変数 COB\_LIBRARY に必要な DLL の指定が漏れている。

また、プログラムの作りによっては以下の場合にも発生することがある。

- ・ 環境変数 COB\_LIBRARY に指定した DLL がアンロードされた。
- ・ コンパイラ内部で保持しているハンドル情報を破壊した。

このようにエラー発生原因が複数あるため、原因を特定するためには 1 つ 1 つ確認していく必要があり、多くの工数が掛かる。

環境変数 COB\_CALLERR=ON を指定しておく、本実行時エラー発生時、詳細情報を標準エラー出力に出力することができるため、エラー原因の特定が容易になる場合がある。

ここでは、出力した詳細情報から原因を特定する方法について説明する。

詳細情報はエラーの発生した箇所によって大きく 3 つの形式がある。

[形式 1] 一意名に指定したサブプログラム名 (エントリ名) の途中に空白文字がある

| 出力フォーマット                                                              | 備考 |
|-----------------------------------------------------------------------|----|
| COBERROR: IDENT NAME INVALID                                          | ①  |
| IDENT NAME DUMP SIZE=32                                               |    |
| 43 42 4c 5f 46 49 4c 45 31 20 20 20 20 43 20 20      CBL_FILE1      C | ②  |
| 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20                       |    |
| WORK IDENT NAME DUMP INVALID POS=14                                   |    |
| 43 42 4c 5f 46 49 4c 45 31 20 20 20 20 43 20 20      CBL_FILE1      C |    |
| 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20                       |    |
| Environment List                                                      |    |
| =::=::¥                                                               |    |
| =C:=C:¥Program Files¥Microsoft Visual Studio 11.0¥VC                  |    |
| =D:=D:¥TestEnv¥tmp                                                    |    |
| ALLUSERSPROFILE=C:¥ProgramData                                        |    |
| ...                                                                   |    |

**備考**

- ① COBERROR:に IDENT NAME INVALID が設定されている場合、一意名の途中に空白文字があることを表す。
- ② 一意名に設定されている値のダンプである。16進形式と文字列形式で表示している。これによりどこがおかしいかを判断することができる。上記例では、14 文字目に"C"という文字があるため、10～13 文字目の空白文字が一意名途中の空白として認識されている。

[形式 2] 環境変数 COB\_LIBRARY に指定した DLL 名が誤っている

| 出力フォーマット                                                                                       | 備考              |
|------------------------------------------------------------------------------------------------|-----------------|
| COBERROR:LOAD LIBRARY FAILED                                                                   | ①               |
| ERRCODE=126                                                                                    | ②               |
| MODULE NAME DUMP SIZE=27                                                                       |                 |
| 44 3a 5c 41 50 50 5c 54 50 45 44 53 45 5c 62 69<br>6e 5c 74 70 6c 69 62 2e 64 6c 6c            | ③               |
| D:¥APP¥TPBASE¥bin¥tplib.dll                                                                    |                 |
| use_handle            save_handle            new_handle            save_err    so_name         |                 |
| 00007FFFF7E1FDE0    00007FFFF7E1FDE0    00007FFFF7E1FDE0    127        -----                   |                 |
| 000000CC1C7AE390    000000CC1C7AE390    000000CC1C7AE390    127        F:¥GWPRD¥GCOS¥VIS1¥tpp¥ |                 |
|                                                                                                | APMS_COMMON.dll |
| 0000000000000000    -----            -----        0            D:¥APP¥TPBASE¥bin¥tplib         |                 |
|                                                                                                | .dll            |
| Environment List                                                                               |                 |
| =::=::¥                                                                                        |                 |
| =C:=C:¥Program Files¥Microsoft Visual Studio 11.0¥VC                                           |                 |
| =D:=D:¥TestEnv¥tmp                                                                             |                 |
| ALLUSERSPROFILE=C:¥ProgramData                                                                 |                 |
| ...                                                                                            |                 |

**備考**

- ① COBERROR:に LOAD LIBRARY FAILED が設定されている場合、環境変数 COB\_LIBRARY に指定した DLL のロードに失敗したことを表す。
- ② ERRCODE の値は、DLL のロードに失敗した原因を表す Windows システムエラーコードである。  
126 は、モジュール(DLL)が見つからなかったことを示すコード値である。  
その他のエラーコードについては、OS の説明書を参照。
- ③ 失敗の原因となった DLL 名を 16 進コードおよび文字列で表示する。  
これらを確認することで、COB\_LIBRARY に指定した DLL 名やパスの指定が誤っているのかどうかを判断することが可能である。

環境変数 COB\_LIBERR=CONTINUE を指定している場合は、本形式のエラーは発生しない。



[形式 3] 一意名に指定したサブプログラム名 (エントリ名) が実行可能プログラムにも環境変数 COB\_LIBRARY で指定した DLL にもない

| 出力フォーマット                                             |                  |                  |                  |                                            | 備考 |
|------------------------------------------------------|------------------|------------------|------------------|--------------------------------------------|----|
| COBERROR:GET PROC ADDRESS FAILED                     |                  |                  |                  |                                            | ①  |
| IDENT NAME DUMP SIZE=30                              |                  |                  |                  |                                            |    |
| 53 51 4C 5F 52 54 4E 5F 47 57 5F 54 31 50 41 54      |                  |                  | SQL_RTN_GW_T1PAT |                                            |    |
| 4E 20 20 20 20 20 20 20 20 20 20 20 20 20 20         |                  |                  | N                |                                            |    |
| ENTRY NAME DUMP SIZE=17                              |                  |                  |                  |                                            | ②  |
| 53 51 4C 5F 52 54 4E 5F 47 57 5F 54 31 50 41 54      |                  |                  | SQL_RTN_GW_T1PAT |                                            |    |
| 4E                                                   |                  |                  | N                |                                            |    |
| use_handle                                           | save_handle      | new_handle       | save_err         | so_name                                    | ③  |
| 00007FFFF7E1FDE0                                     | 00007FFFF7E1FDE0 | 00007FFFF7E1FDE0 | 127              | -----                                      |    |
| 000000CC1C7AE390                                     | 000000CC1C7AE390 | 0000000000000000 | 127              | F:¥GWPRD¥GCOS¥VIS1¥tpp¥<br>APMS_COMMON.dll |    |
| Environment List                                     |                  |                  |                  |                                            |    |
| =::=:¥                                               |                  |                  |                  |                                            |    |
| =C:=C:¥Program Files¥Microsoft Visual Studio 11.0¥VC |                  |                  |                  |                                            |    |
| =D:=D:¥TestEnv¥tmp                                   |                  |                  |                  |                                            |    |
| ALLUSERSPROFILE=C:¥ProgramData                       |                  |                  |                  |                                            |    |
| ...                                                  |                  |                  |                  |                                            |    |

**備考**

- ① COBERROR: に GET PROC ADDRESS FAILED が設定されている場合、サブプログラム名の関数アドレス取得に失敗したことを示す。
- ② 関数アドレス取得に失敗したサブプログラム名が 16 進ダンプおよび文字列として表示される。ここに表示されるサブプログラム名が意図した名前でない場合、一意名に指定したサブプログラム名が不正であると考えられる。
- ③ so\_name 欄には、サブプログラム名を検索した対象のモジュール名が表示される。先頭行の (-----) は実行可能プログラム自身であることを表す。  
save\_err 欄には、so\_name 欄に表示したモジュールを検索した結果を表す Windows システムエラーコードが表示される。  
127 は、サブプログラムが見つからなかったことを示すコード値である。

その他のエラーコードについては、OS の説明書を参照。

`so_name` 欄に表示したすべてのモジュールの `save_err` の値が 127 の場合、実行可能プログラム自身または環境変数 `COB_LIBRARY` で指定した DLL 中にサブプログラムが見つからなかったことを表す。

このとき、サブプログラムが実行可能プログラム自身に含まれている場合は、サブプログラムのコンパイル時に `-CY` オプションが指定されていないことが考えられる。

また、サブプログラムを含んでいる DLL が `so_name` 欄に表示されているかを確認することで環境変数 `COB_LIBRARY` に必要な DLL の指定が漏れているかどうかを判断することができる。

環境変数 `COB_LIBRARY` に必要な DLL が指定されており(`so_name` 欄に表示されており)、`save_err` の値が 127 の場合、`use_handle`、`save_handle`、`new_handle` の状態により原因を推測できる。

- `use_handle` と `save_handle` が等しく、`new_handle` と異なる場合、`so_name` 欄の DLL がアンロードされたことを示す。
- `save_handle` と `new_handle` が等しく、`use_handle` と異なる場合、コンパイラ内部で保持しているハンドル情報が壊された可能性がある。

[Enterprise for Linux] CALL 一意名呼び出し実行時、「COB008 CALL で呼び出すエントリが見つかりません。」エラーが発生することがある。

本エラーは通常、以下の場合に発生する。

- 一意名に指定したサブプログラム名 (エントリ名) の途中で空白文字がある。
- 環境変数 `COB_LIBRARY` に指定した共有ライブラリ名が誤っている。  
環境変数 `COB_LIBERR=CONTINUE` を指定していない場合に発生する。一意名に指定したサブプログラムを含む共有ライブラリ名だけでなく、その共有ライブラリ名より前に指定した共有ライブラリ名が誤っている場合にも発生する。
- 一意名に指定したサブプログラム名 (エントリ名) が実行可能プログラムにも環境変数 `COB_LIBRARY` で指定した共有ライブラリにもない。その理由として以下が考えられる。
  - 一意名に指定したサブプログラム名 (エントリ名) が誤っている。
  - 一意名に指定したサブプログラム名 (エントリ名) が実行可能プログラム内にあるが、実行可能プログラム作成時にコンパイルオプション `-CY` を指定していない。
  - 環境変数 `COB_LIBRARY` に必要な共有ライブラリ名の指定が漏れている。

また、プログラムの作りによっては以下の場合にも発生することがある。

- 環境変数 `COB_LIBRARY` に指定した共有ライブラリがアンロードされた。
- コンパイラ内部で保持しているハンドル情報を破壊した。

このようにエラー発生原因が複数あるため、原因を特定するためには 1 つ 1 つ確認していく必要が

あり、多くの工数が掛かる。  
環境変数 COB\_CALLERR=ON を指定しておく、本実行時エラー発生時、詳細情報を標準エラー出力に出力することができるため、エラー原因の特定が容易になる場合がある。

ここでは、出力した詳細情報から原因を特定する方法について説明する。  
詳細情報はエラーの発生した箇所によって大きく 3 つの形式がある。

[形式 1] 一意名に指定したサブプログラム名 (エントリ名) の途中に空白文字がある

| 出力フォーマット                                                              | 備考 |
|-----------------------------------------------------------------------|----|
| COBERROR: IDENT NAME INVALID                                          | ①  |
| IDENT NAME DUMP SIZE=32                                               |    |
| 53 55 42 5F 50 52 4F 47 31 20 20 20 20 43 20 20      SUB_PROG1      C | ②  |
| 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20                       |    |
| WORK IDENT NAME DUMP INVALID POS=14                                   |    |
| 53 55 42 5F 50 52 4F 47 31 20 20 20 20 43 20 20      SUB_PROG1      C |    |
| 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20                       |    |
| Environment List                                                      |    |
| TERM=xterm                                                            |    |
| SHELL=/bin/bash                                                       |    |
| HISTSIZE=1000                                                         |    |
| ...                                                                   |    |

- 備考**
- ① COBERROR: に IDENT NAME INVALID が設定されている場合、一意名の途中に空白文字があることを表す。
  - ② 一意名に設定されている値のダンプである。16 進形式と文字列形式で表示している。これによりどこがおかしいかを判断することができる。上記例では、14 文字目に "C" という文字があるため、10～13 文字目の空白文字が一意名途中の空白として認識されている。

[形式 2] 環境変数 COB\_LIBRARY に指定した共有ライブラリ名が誤っている

| 出力フォーマット                                                                                  | 備考 |
|-------------------------------------------------------------------------------------------|----|
| COBERROR:LOAD LIBRARY FAILED                                                              | ①  |
| ERRINFO=/home/cbltest/dummy.so: cannot open shared object file: No such file or directory | ②  |
| MODULE NAME DUMP SIZE=22                                                                  |    |
| 2F 68 6F 6D 65 2F 63 62 6C 74 65 73 74 2F 64 75 /home/cbltest/du                          | ③  |
| 6D 6D 79 2E 73 6F mmy.so                                                                  |    |
|                                                                                           |    |
| use_handle save_handle new_handle save_err so_name                                        |    |
| 00007FFFF7E1FDE0 00007FFFF7E1FDE0 00007FFFF7E1FDE0 (*1) -----                             |    |
| 000000CC1C7AE390 000000CC1C7AE390 000000CC1C7AE390 (*2) /home/cbltest/bin/SUB1.so         |    |
| 00000000000000000 ----- 0 /home/cbltest/dummy.so                                          |    |
| (*1) ./sample.out: undefined symbol: Sub1                                                 |    |
| (*2) /home/cbltest/bin/SUB1.so: undefined symbol: Sub1                                    |    |
|                                                                                           |    |
| Environment List                                                                          |    |
| TERM=xterm                                                                                |    |
| SHELL=/bin/bash                                                                           |    |
| HISTSIZE=1000                                                                             |    |
| ...                                                                                       |    |

**備考**

- ①COBERROR:に LOAD LIBRARY FAILED が設定されている場合、環境変数 COB\_LIBRARY に指定した共有ライブラリのロードに失敗したことを表す。
- ②ERRINFO の値は、共有ライブラリのロードに失敗した原因を表す文字列である。
- ③失敗の原因となった共有ライブラリ名を 16 進コードおよび文字列で表示する。  
これらを確認することで、COB\_LIBRARY に指定した共有ライブラリ名やパスの指定が誤っているのかどうかを判断することが可能である。

環境変数 COB\_LIBERR=CONTINUE を指定している場合は、本形式のエラーは発生しない。

[形式 3] 一意名に指定したサブプログラム名 (エントリ名) が実行可能プログラムにも環境変数 COB\_LIBRARY で指定した共有ライブラリにもない

| 出力フォーマット                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 備考                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| <pre>COBERROR:GET PROC ADDRESS FAILED IDENT NAME DUMP SIZE=30 53 55 42 5F 50 52 4F 47 52 41 4D 5F 45 4E 54 52 59 20 20 20 20 20 20 20 20 20 20 20 20 20  ENTRY NAME DUMP SIZE=17 53 55 42 5F 50 52 4F 47 52 41 4D 5F 45 4E 54 52 59  use_handle      save_handle      new_handle      save_err  so_name 00007FFFF7E1FDE0 00007FFFF7E1FDE0 00007FFFF7E1FDE0 (*1)  ----- 000000CC1C7AE390 000000CC1C7AE390 0000000000000000 (*2)  /home/cbltest/bin/SUB1.so (*1) ./sample.out: undefined symbol: SUB_PROGRAM_ENTRY (*2) /home/cbltest/bin/SUB1.so: undefined symbol: SUB_PROGRAM_ENTRY  Environment List TERM=xterm SHELL=/bin/bash HISTSIZE=1000 ...</pre> | <div>①</div> <div>②</div> <div>③</div> |

備考

- ①COBERROR: に GET PROC ADDRESS FAILED が設定されている場合、サブプログラム名の関数アドレス取得に失敗したことを示す。
- ②関数アドレス取得に失敗したサブプログラム名が 16 進ダンプおよび文字列として表示される。ここに表示されるサブプログラム名が意図した名前でない場合、一意名に指定したサブプログラム名が不正であると考えられる。
- ③so\_name 欄には、サブプログラム名を検索した対象のモジュール名が表示される。先頭行の (-----)は実行可能プログラム自身であることを表す。  
save\_err 欄には、so\_name 欄に表示したモジュールを検索した結果または結果記載箇所との対応を示す注釈番号が表示される。「0」は検索が成功したことを表し、「(\*n)」は注釈番号を表す。
- ④③の save\_err 欄の注釈番号に対応し、so\_name 欄に表示したモジュールを検索した際に発生した

エラーの原因を表す文字列を表示する。so\_name 欄に表示したすべてのモジュールのエラー原因を表す文字列が「undefined symbol」の場合、実行可能プログラム自身または環境変数 COB\_LIBRARY で指定した共有ライブラリ中にサブプログラムが見つからなかったことを表す。

このとき、サブプログラムが実行可能プログラム自身に含まれている場合は、実行可能プログラム作成時に -CY オプションが指定されていないことが考えられる。

また、サブプログラムを含んでいる共有ライブラリが so\_name 欄に表示されているかを確認することで環境変数 COB\_LIBRARY に必要な共有ライブラリの指定が漏れているかどうかを判断することができる。

環境変数 COB\_LIBRARY に必要な共有ライブラリが指定されており(so\_name 欄に表示されており)、当該モジュールのエラー原因を表す文字列が「undefined symbol」の場合、use\_handle, save\_handle, new\_handle の状態により原因を推測できる。

・use\_handle と save\_handle が等しく、new\_handle と異なる場合、so\_name 欄の共有ライブラリがアンロードされたことを示す。

・save\_handle と new\_handle が等しく、use\_handle と異なる場合、コンパイラ内部で保持しているハンドル情報が壊された可能性がある。

## 第8章 利用者定義関数

### 8.1 概説

利用者定義関数とは、利用者が自由に定義することのできる関数のことである。

利用者は、目的に応じて関数を設計し、引数、返却値および関数内の処理内容を記述する。

一度定義された利用者定義関数は、いずれのプログラム、クラス、関数等からも参照ができる。別翻訳群として翻訳された利用者定義関数も、外部リポジトリ機能を利用することにより参照が可能である。

また、利用者定義関数は常に再帰呼び出し可能である。

### 8.2 利用者定義関数の定義

利用者定義関数の定義のしかた (概要) は次のとおりである。

詳細な書き方については、『COBOL 言語説明書』を参照。

```
IDENTIFICATION DIVISION.
FUNCTION-ID. SAMPLE1.
DATA DIVISION.
:
LINKAGE SECTION.
:
 USING に指定されている仮パラメータ, および RETURNING に指定
 されている返却項目の定義
:
PROCEDURE DIVISION
 USING 仮パラメータ... RETURNING 返却項目.
HAJIME.
:
 関数手続き本体
:
 MOVE ～ TO 返却項目

END FUNCTION SAMPLE1.
```

利用者定義関数の実行結果は、返却項目に格納される。返却項目の項類により利用者定義関数の型(関数値)が決まるため、RETURNING 句は必ず指定しなければならない。

また、引数の型や返却項目の型だけを宣言したものを関数原型という。関数原型の定義のしかたは次のとおりである。

```
IDENTIFICATION DIVISION.
FUNCTION-ID. SAMPLE1 IS PROTOTYPE.
DATA DIVISION.
LINKAGE SECTION.
:
: USING に指定されている仮パラメータ, 返却項目の定義
:
PROCEDURE DIVISION
 USING 仮パラメータ ... RETURNING 返却項目.
END FUNCTION SAMPLE1.
```

### 8.3 利用者定義関数の呼び出し

利用者定義関数は、関数一意名を指定して呼び出す。

関数一意名は、式や文の一部として使うことができるため、CALL 文によるサブプログラムの呼び出しに比べて簡潔に記述できる。

MOVE 文での関数呼び出し例を次に示す (FUNC1 が関数一意名で引数が PAR1, PAR2 の例)。

```
MOVE FUNCTION FUNC1 (PAR1 PAR2) TO A
```

利用者定義関数の実行結果を得ることは、内側にプログラム定義を含まない最外側のプログラム定義(サブプログラム)を呼び出して、実行結果を得ることと同じである。

前記の例の場合、関数 FUNC1 の手続きが実行され、その結果の値が A 項目に転記される。

その転記規則は、MOVE の転記規則に従う。

また、関数を呼び出す場合には、関数呼び出しをするプログラム定義やクラス定義のリポジトリ段落に、その関数のリポジトリ情報を指定して、リポジトリが参照可能となっていなければならない。

また、利用者定義関数の定義自体は、別翻訳群で翻訳されていてもよいし、同じ翻訳群の中で



定義されていてもよい。ただし、同じ翻訳群の中で定義する場合には、その関数呼び出しが記述されているプログラム定義やクラス定義より前に関数定義または関数原型定義を記述していなければならない。

## 8.4 データの受け渡し

利用者定義関数へデータを引き渡すには、関数一意名の引数を使用する。引数は関数一意名の評価開始時点で評価される。

(1) 引数の受け渡しは、次のように決まる。

関数一意名の引数には、BY REFERENCE(参照渡し)、BY CONTENT(内容渡し)、BY VALUE(値渡し)を指定する必要はない。利用者定義関数の仮パラメータの指定により次のように想定される。

- (a) BY REFERENCE(参照渡し)が想定されるのは、対応する仮パラメータに BY REFERENCE が指定または想定され、引数が受け取り側として許される一意名であり、オブジェクトプロパティまたはオブジェクトデータ項目以外のものであるときである。
- (b) BY CONTENT(内容渡し)が想定されるのは、対応する仮パラメータに BY REFERENCE が指定または想定され、引数が定数、算術式、オブジェクトプロパティ、オブジェクトデータ項目、または受け取り側として許されない一意名のときである。
- (c) BY VALUE(値渡し)が想定されるのは、対応する仮パラメータに BY VALUE 指定があるときである。BY VALUE が指定できる字類は数字、オブジェクト、またはポインタのみである。

なお、表意定数については、下記のとおりである。

- (a) BY REFERNECE の場合、記述できません。
- (b) BY CONTENT の場合、すべて記述できる。
- (c) BY VALUE の場合、ALL 指定のない ZERO (ZEROS, ZEROES) だけである。

引数の種類と想定される引数の渡し方を次の表 8-1 に示す。

表 8-1 引数の種類と想定される引数の渡し方

| 引数の種類 |                   | 想定される引数の渡し方 |      |     |
|-------|-------------------|-------------|------|-----|
|       |                   | 参照渡し        | 内容渡し | 値渡し |
| 一意名   | 集団項目              | ○           | —    | ×   |
|       | 英字項目              | ○           | —    | ×   |
|       | 英数字項目             | ○           | —    | ×   |
|       | 英数字編集項目           | ○           | —    | ×   |
|       | 数字編集項目            | ○           | —    | ×   |
|       | ブールデータ項目          | ×           | ×    | ×   |
|       | 指標データ項目           | ○           | —    | ×   |
|       | 日本語項目             | ○           | —    | ×   |
|       | 日本語編集項目           | ○           | —    | ×   |
|       | 数字項目              | ○           | —    | ○   |
|       | オブジェクト参照          | —           | ○    | ○   |
|       | データポインタ項目         | ○           | —    | ○   |
|       | 指標名               | ×           | ×    | ×   |
|       | ポインタ名             | ○           | —    | ○   |
|       | 可変繰返しデータ項目        | ○           | —    | ×   |
|       | 関数一意名             | —           | ○    | ○*1 |
|       | 修飾付き添字付きデータ名      | ○           | —    | ○*1 |
|       | 部分参照              | ×           | ×    | ×   |
|       | オブジェクトビュー         | ×           | ×    | ×   |
|       | 既定義オブジェクト参照 NULL  | —           | ○    | ○   |
|       | 既定義オブジェクト参照 SELF  | —           | ○    | ○   |
|       | 既定義オブジェクト参照 SUPER | ×           | ×    | ×   |
|       | オブジェクトプロパティ       | ×           | ○    | ×   |
|       | 既定義アドレス NULL      | —           | ○    | ○   |
|       | 修飾付き行数カウンタ        | ○           | —    | ○   |
| 定数    | 英数字定数             | —           | ○    | ×   |
|       | 数字定数              | —           | ○    | ○   |
|       | ブール定数             | —           | ○    | ×   |
|       | 日本語定数             | —           | ○    | ×   |

|      |      |   |   |   |
|------|------|---|---|---|
|      | 表意定数 | — | ○ | × |
| 算術式  |      | — | ○ | ○ |
| ブール式 |      | × | × | × |

記号の意味

○:指定可能

×:指定不可(翻訳エラー)

—:想定外(存在しない)

\*1:数字字類のみ

(2) 引数が参照渡しの場合、引数の項類と適合規則の組み合わせは、次のとおりである。

| <div>仮パラメータ</div> <div>引数</div> | 集団項目 | 英字項目 | 英数字項目 | 英数字編集項目 | 数字編集項目 | ブール項目 | 指標データ項目 | 日本語項目 | 日本語編集項目 | 数字項目 | オブジェクト参照 | データポインタデータ項目 |
|---------------------------------|------|------|-------|---------|--------|-------|---------|-------|---------|------|----------|--------------|
| 集団項目                            | *1   | —    | *1    | *1      | *1     | —     | —       | —     | —       | —    | —        | —            |
| 英字項目                            | —    | *2   | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 英数字項目                           | *1   | —    | *2    | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 英数字編集項目                         | *1   | —    | —     | *2      | —      | —     | —       | —     | —       | —    | —        | —            |
| 数字編集項目                          | *1   | —    | —     | —       | *3     | —     | —       | —     | —       | —    | —        | —            |
| ブール項目                           | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 指標データ項目                         | —    | —    | —     | —       | —      | —     | *2      | —     | —       | —    | —        | —            |
| 日本語項目                           | —    | —    | —     | —       | —      | —     | —       | *2    | —       | —    | —        | —            |
| 日本語編集項目                         | —    | —    | —     | —       | —      | —     | —       | —     | *2      | —    | —        | —            |
| 数字項目                            | —    | —    | —     | —       | —      | —     | —       | —     | —       | *2   | —        | —            |
| オブジェクト参照<br>非限定オブジェクト参照         | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | *4       | —            |
| オブジェクト参照<br>クラス名指定              | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | *5       | —            |
| オブジェクト参照<br>ACTIVE-CLASS 指定     | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | *6       | —            |
| データポインタデータ項目                    | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | *7           |

記号の意味

\*1: 仮パラメータは対応する引数と同じかまたはより少ないバイト数を持つように記述されなければならない。

可変繰返しデータ項目は最大長として扱う。

\*2: 仮パラメータおよび引数の BLANK WHEN ZERO 句, JUSTIFIED 句, PICTURE 句, USAGE 句, およ

び SIGN 句の指定内容が同じでなければならない。

- \*3: 仮パラメータおよび引数の BLANK WHEN ZERO 句, JUSTIFIED 句, PICTURE 句, USAGE 句, および SIGN 句の指定内容が同じでなければならない。

PICTURE 記号のピリオドどうしが一致するのは, DECIMAL-POINT IS COMMA 句が, 活性化する側とされる側の両方の実行時要素中で有効, または両方で無効な場合だけとする。

PICTURE 記号のコンマどうしが一致するのは, DECIMAL-POINT IS COMMA 句が活性化する側とされる側の両方の実行時要素中で有効, または両方で無効な場合だけとする。

- \*4: 引数, 仮パラメータとも非限定オブジェクト参照でなければならない。

- \*5: 引数, 仮パラメータとも同じクラス名を指定して, FACTORY 指定の有無が同じでなければならない。

- \*6: 引数, 仮パラメータとも ACTIVE-CLASS を指定して, FACTORY 指定の有無が同じで, かつ活性化されるべきメソッドを呼び出すには, 既定義オブジェクト参照の SELF や SUPER を用いるか, または ACTIVE-CLASS 指定つきで記述されたオブジェクト参照を用いなければならない。

- \*7: 引数, 仮パラメータとも字類がポインタで項類が同じでなければならない。

―: 適合エラー

(3) 引数が内容渡しの場合, 引数の項類と適合規則の組み合わせは, 次のとおりである。

| <div>仮パラメータ</div> <div>引数</div> | 集団項目 | 英字項目 | 英数字項目 | 英数字編集項目 | 数字編集項目 | ブール項目 | 指標データ項目 | 日本語項目 | 日本語編集項目 | 数字項目 | オブジェクト参照 | データポインタデータ項目 |
|---------------------------------|------|------|-------|---------|--------|-------|---------|-------|---------|------|----------|--------------|
| 集団項目                            | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 英字項目                            | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 英数字項目                           | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 英数字編集項目                         | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 数字編集項目                          | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| ブール項目                           | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 指標データ項目                         | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 日本語項目                           | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |
| 日本語編集項目                         | ―    | ―    | ―     | ―       | ―      | ―     | ―       | ―     | ―       | ―    | ―        | ―            |

|                             |   |   |   |   |   |   |   |   |   |   |    |   |
|-----------------------------|---|---|---|---|---|---|---|---|---|---|----|---|
| 数字項目                        | — | — | — | — | — | — | — | — | — | — | —  | — |
| オブジェクト参照<br>非限定オブジェクト参照     | — | — | — | — | — | — | — | — | — | — | *1 | — |
| オブジェクト参照<br>クラス名指定          | — | — | — | — | — | — | — | — | — | — | *2 | — |
| オブジェクト参照<br>ACTIVE-CLASS 指定 | — | — | — | — | — | — | — | — | — | — | *3 | — |
| データポインタデータ項目                | — | — | — | — | — | — | — | — | — | — | —  | — |

## 記号の意味

\*1: 「SET 仮パラメータ TO 引数」が実行可能な組み合わせでなければならない。

\*2: 次の条件をすべて満たさなければならない。

- ・活性化されるべきメソッドを呼び出すには、そのクラス名を用いるか、またはそのクラス名で記述されたオブジェクト参照を用いるか、しなければならない。
- ・「SET オブジェクト参照 TO 引数」が実行可能な組み合わせでなければならない。
- ・オブジェクト参照は、クラス名が指定され、FACTORY 指定の有無が仮パラメータと同じもの。
- ・SET 文の送り出し側の一意名が指定されて、かつ受け取り側の一意名がクラス名付きで記述されている場合、送り出し側はクラス名付きで記述されたオブジェクト参照で、次の規則に従うものでなければならない。
- ・受け取り側の一意名の記述に指定されたクラス名は、送り出し側の一意名に指定されたものと同じクラスか、またはそのサブクラスを指すもの。
- ・FACTORY 指定の有無は、送り出し側の一意名の記述と同じでなければならない。

\*3: 次の条件をすべて満たさなければならない。

- ・活性化されるべきメソッドを呼び出すには、既定義オブジェクト参照の SELF や SUPER を用いるか、または ACTIVE-CLASS 指定つきで記述されたオブジェクト参照を用いなければならない。
- ・「SET オブジェクト参照 TO 引数」が実行可能な組み合わせでなければならない。
- オブジェクト参照は、ACTIVE-CLASS 付きで FACTORY 指定有無が仮パラメータと同じもの。

―: 適合エラー

(4) 引数が値渡しの場合、引数の項類と適合規則の組み合わせは、次のとおりである。

| <div>仮パラメータ</div> <div>引数</div> | 集団項目 | 英字項目 | 英数字項目 | 英数字編集項目 | 数字編集項目 | ブール項目 | 指標データ項目 | 日本語項目 | 日本語編集項目 | 数字項目 | オブジェクト参照 | データポインタデータ項目 |
|---------------------------------|------|------|-------|---------|--------|-------|---------|-------|---------|------|----------|--------------|
| 集団項目                            | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 英字項目                            | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 英数字項目                           | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 英数字編集項目                         | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 数字編集項目                          | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| ブール項目                           | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 指標データ項目                         | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 日本語項目                           | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 日本語編集項目                         | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| 数字項目                            | —    | *1   | *1    | *1      | *1     | —     | —       | *1    | *1      | *2   | —        | —            |
| オブジェクト参照<br>非限定オブジェクト参照         | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | —            |
| オブジェクト参照<br>クラス名指定              | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | *3       | —            |
| オブジェクト参照<br>ACTIVE-CLASS 指定     | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | *4       | —            |
| データポインタデータ項目                    | —    | —    | —     | —       | —      | —     | —       | —     | —       | —    | —        | *5           |

記号の意味

\*1: 「MOVE 引数 TO 仮パラメータ」に適合するもの。

\*2: 「COMPUTE 仮パラメータ = 引数」に適合するもの。

\*3: 仮パラメータが ACTIVE-CLASS 指定のオブジェクト参照以外の場合、  
「SET 仮パラメータ TO 引数」に適合するもの。

仮パラメータが **ACTIVE-CLASS** 指定のオブジェクト参照の場合、  
次のすべてに適合するもの。

- ・活性化されるべきメソッドを呼び出すには、そのクラス名を用いるか、またはそのクラス名で記述されたオブジェクト参照を用いるか、しなければならない。
- ・「**SET** オブジェクト参照 **TO** 引数」に適合するもの。  
オブジェクト参照は、クラス名が指定され、**FACTORY** 指定の有無が仮パラメータと同じもの。
- ・**SET** 文の送り出し側の一意名が指定されて、かつ受け取り側の一意名がクラス名付きで記述されている場合、送り出し側はクラス名付きで記述されたオブジェクト参照で、次の規則に従うものでなければならない。
  - ・受け取り側の一意名の記述に指定されたクラス名は、送り出し側の一意名に指定されたものと同じクラスか、またはそのサブクラスを指すもの。
  - ・**FACTORY** 指定の有無は、送り出し側の一意名の記述と同じでなければならない。

\*4: 仮パラメータが **ACTIVE-CLASS** 指定のオブジェクト参照以外の場合、

「**SET** 仮パラメータ **TO** 引数」に適合するもの。

仮パラメータが **ACTIVE-CLASS** 指定のオブジェクト参照の場合、  
次のすべてに適合するもの。

- ・活性化されるべきメソッドを呼び出すには、既定義オブジェクト参照の **SELF** や **SUPER** を用いるか、または **ACTIVE-CLASS** 指定つきで記述されたオブジェクト参照を用いなければならない。かつ、
- ・「**SET** オブジェクト参照 **TO** 引数」に適合するもの。  
オブジェクト参照は、**ACTIVE-CLASS** 付きで **FACTORY** 指定有無が仮パラメータと同じもの。

\*5: 「**SET** 仮パラメータ **TO** 引数」に適合するもの。

—: 適合エラー



## 8.5 再帰呼び出し

利用者定義関数は常に再帰呼び出し可能である。

よって、利用者定義関数の手続き内において、自分自身を呼び出すことができる。

次の例は、数字  $n$  の階乗を求める関数 FACTORIALX を利用者定義関数として定義した例である。

```
FACTORIALX(n): n>1 のとき n * FACTORIALX(n-1)
 n=1 のとき 1
```

**\*> 利用者定義関数の定義**

```
IDENTIFICATION DIVISION.
FUNCTION-ID. FACTORIALX.
DATA DIVISION.
LOCAL-STORAGE SECTION.
01 COUNT1 PIC S9(3).
LINKAGE SECTION.
01 PAR1 PIC S9(3).
01 RET1 PIC S9(3).
PROCEDURE DIVISION
 USING BY VALUE PAR1 RETURNING RET1.

HAJIME.

 IF PAR1 = 0 THEN
 MOVE 1 TO RET1
 EXIT FUNCTION
 END-IF.

 COMPUTE COUNT1 = FUNCTION FACTORIALX (PAR1 - 1).
 COMPUTE RET1 = PAR1 * COUNT1.

END FUNCTION FACTORIALX.
```

**\*> メインプログラム**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PROG2.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.

 FUNCTION FACTORIALX.
```

```

DATA DIVISION.
WORKING-STORAGE SECTION.
01 PAR1 PIC S9(3) VALUE 2.
01 RESULT1 PIC S9(3) VALUE 0.
PROCEDURE DIVISION.
HAJIME.
 MOVE FUNCTION FACTORIALX (PAR1) TO RESULT1.
 STOP RUN.
END PROGRAM PROG2.

```

## 8.6 利用者定義関数の利用方法

〔Enterprise for Linux〕本節の以降の説明文では、オブジェクトファイル名の拡張子「.obj」を「.o」、ディレクトリのパス区切り文字「¥」を「/」に読み替えること。

利用者定義関数は、その利用者定義関数を使うプログラム定義やクラス定義などと同じ翻訳群（同一翻訳群）の中に記述してもよいし、1つの利用者定義関数の定義だけで1翻訳群として記述してもよい。

しかし、利用面から考えると、1つの利用者定義関数の定義だけで1翻訳群としておくのが望ましい。

それは、同一翻訳群の中に利用者定義関数を記述すると、生成されるオブジェクトファイル名が、本コンパイラの命名規則により「ソースファイル名\_利用者定義関数名.obj」となり、他の翻訳群から利用しにくいためである（同一翻訳群を翻訳する場合のオブジェクトファイル名の生成規則については、「1.5 プログラムのリンク」を参照）。

次に、FUNC1（ソースファイル名は func1.cob）という関数が別翻訳群で定義されており、その関数呼び出しが、PROG1（ソースファイル名は PROG1.cob）で行われている場合の例とその翻訳方法を示す。

```

ソースファイル名: func1.cob
*> 関数定義の定義
IDENTIFICATION DIVISION.
FUNCTION-ID. FUNC1.
DATA DIVISION.
LINKAGE SECTION.

```

```

01 PAR1 PIC S9(3).
01 PAR2 PIC S9(3).
01 RET1 PIC S9(3).
PROCEDURE DIVISION
 USING PAR1 PAR2 RETURNING RET1.

HAJIME.

 COMPUTE RET1 = PAR1 + PAR2.
 EXIT FUNCTION.
END FUNCTION FUNC1.

```

前記のプログラムはあらかじめ次の記述にて翻訳されていると想定する。

```
cob -c -CG func1.cob
```

これにより、カレントディレクトリ(¥func1\_def とする)に関数 FUNC1 の実態である func1.obj と FUNC1 のリポジトリ情報が生成される。(-CG オプションは、リポジトリ情報を生成するオプション)

ソースファイル名:PROG1.cob

**\*> 関数原型名の宣言および関数一意名の指定**

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PROG1.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.

 FUNCTION FUNC1.

DATA DIVISION.
WORKING-STORAGE SECTION.
01 ARG1 PIC S9(3) VALUE 1.
01 ARG2 PIC S9(3) VALUE 2.
01 RES1 PIC S9(3) VALUE 0.
PROCEDURE DIVISION.
HAJIME.

 MOVE FUNCTION FUNC1 (ARG1 ARG2) TO RES1.
 STOP RUN.
END PROGRAM PROG1.

```

前記のプログラム中には関数原型の記述はないが、リポジトリ段落に **FUNC1** が指定されているため、コンパイラは、外部リポジトリから **FUNC1** に関するリポジトリ情報を参照する。その結果、コンパイラは、関数 **FUNC1** の引数や返却値のデータ型を知ることができるので、問題なくプログラムを翻訳できる。

実行形式を生成するためには、次のように翻訳コマンドを指定する。

```
cob -M -R¥func1_def PROG1.cob ¥func1_def¥func1.obj
```

ここで、関数 **FUNC1** のリポジトリ情報は、¥func1\_def ディレクトリ配下に存在することを想定している。( -R オプションは、外部リポジトリが存在するディレクトリを指定するオプション)

このように、前もって翻訳済みの関数の呼び出しを行う場合には、その関数の外部リポジトリ情報を参照すればよく、関数原型定義も必要ない。ただし、呼び出す関数の理解が深まりやすくなることから、関数原型定義は記載するのが望ましい。

**PROG1.cob** に関数原型定義を記載した例 (推奨例) を次に示す。

ソースファイル名: **PROG1.cob** : 推奨例

\*> 関数原型定義

```
IDENTIFICATION DIVISION.
FUNCTION-ID. FUNC1 IS PROTOTYPE.
DATA DIVISION.
LINKAGE SECTION.
01 PAR1 PIC S9(3).
01 PAR2 PIC S9(3).
01 RET1 PIC S9(3).
PROCEDURE DIVISION
 USING PAR1 PAR2 RETURNING RET1.
END FUNCTION FUNC1.
```

\*> プログラム本体

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PROG1.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
 FUNCTION FUNC1.
DATA DIVISION.
```

```
WORKING-STORAGE SECTION.
01 ARG1 PIC S9(3) VALUE 1.
01 ARG2 PIC S9(3) VALUE 2.
01 RES1 PIC S9(3) VALUE 0.
PROCEDURE DIVISION.
HAJIME.
 MOVE FUNCTION FUNC1 (ARG1 ARG2) TO RES1.
 STOP RUN.
END PROGRAM PROG1.
*> 関数定義の定義
IDENTIFICATION DIVISION.
FUNCTION-ID. FUNC1.
DATA DIVISION.
LINKAGE SECTION.
01 PAR1 PIC S9(3).
01 PAR2 PIC S9(3).
01 RET1 PIC S9(3).
PROCEDURE DIVISION
 USING PAR1 PAR2 RETURNING RET1.
HAJIME.
 COMPUTE RET1 = PAR1 + PAR2.
 EXIT FUNCTION.
END FUNCTION FUNC1.
```

## 第9章 オブジェクト指向機能

### 9.1 オブジェクト指向の概念

オブジェクト指向プログラミングでは、業務システムを、相互に作用し合うソフトウェアオブジェクトの集まりとして開発し、実装する。

ソフトウェアオブジェクトは、ある問題領域に関係する、あらゆる概念をモデル化するのに使用できる。たとえば、銀行口座、従業員、部品、工程、プログラム、フィールド、ファイル、データ構造などを表現できる。

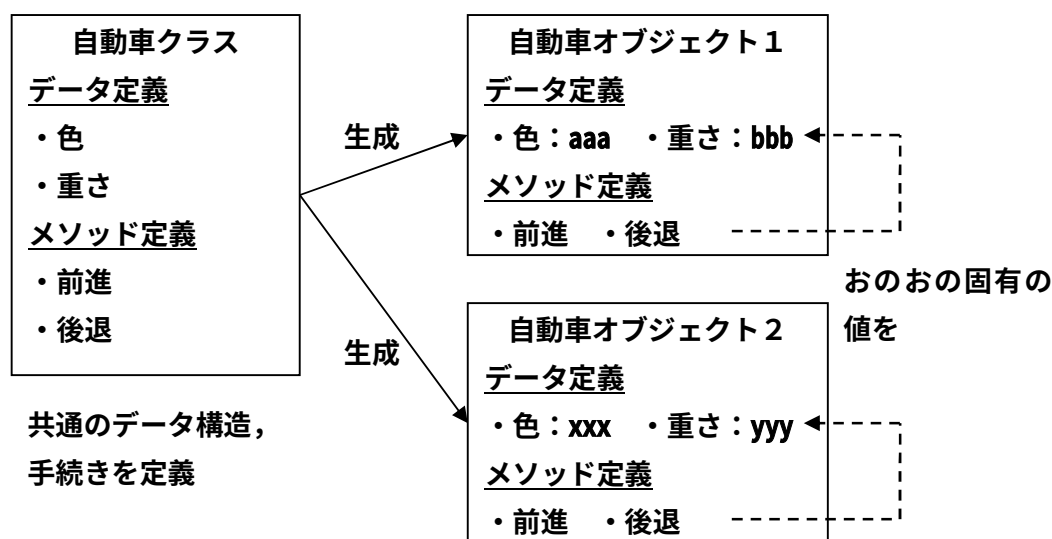
ソフトウェアオブジェクトは、自動車のような日常生活の大半のものと同様に、固有性があり、何らかの属性 (オブジェクトデータ) と振る舞い (メソッド) をもっている。たとえば、自動車をモデル化するのに使われるソフトウェアオブジェクトであれば、色、重さ、長さのような属性、そして、前進、後退のような振る舞いが考えられる。

したがって、ソフトウェアオブジェクトとは、固有性、特定のデータ構造、および特定の手続き (すなわちプログラムコード) をもつ実体であるといっていよい。プログラムコードは、小さなモジュールの集まりにまとめられる。

#### 9.1.1 クラスとオブジェクト

業務システムに存在し得る何百から何千の異なったソフトウェアオブジェクトを扱いやすくするために、オブジェクトはクラスにまとめられる。

以降に例を示す。

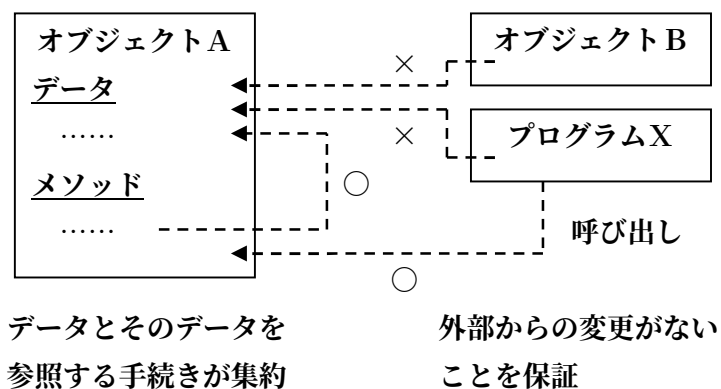


クラスは、共通のデータ構造をもち、同じメソッドを使うオブジェクトのグループである。クラスの各オブジェクトは、そのクラスで定義されたデータ構造に従い、固有のデータ値の組をもっている。データ構造およびメソッドはクラスに対して一度だけ定義されるが、そのクラスのすべてのオブジェクトに共有され、すべてのオブジェクトに使用される。

### 9.1.2 カプセル化

おののオブジェクトのデータは、そのオブジェクトのメソッドからしかアクセスできないため、各オブジェクトのデータを隠す (カプセル化する) ことを可能にしている。

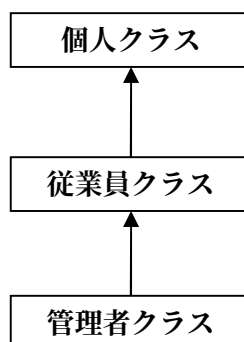
カプセル化により、データとそのデータを参照する手続きが、プログラム部品単位で独立性を持ち、保守性、再利用性が保たれる。



### 9.1.3 継承

継承とは、あるクラスを拡張して新しいクラスを定義する機構である。継承関係にあるクラス群は、階層構造をなす。

以降に例を示す。



従業員クラスは個人クラスを継承したクラスであり、管理者クラスは従業員クラスを継承したクラスである。

あるクラスの継承元クラスは、上位クラス、すなわち祖先であり、継承先クラスは、下位クラス、すなわち子孫である。下位クラスは、祖先のすべての能力を包含しており、そこにさらに能力を追加したり、祖先の能力を上書き（オーバーライド）したりしてもよい。たとえば、管理者クラスのオブジェクトでは、上位クラスである個人クラスおよび従業員クラスに記述されたすべてのオブジェクトデータおよびメソッドの定義と、それに加えて、管理者クラスだけに記述されたオブジェクトデータおよびメソッドの定義が、管理者クラスのオブジェクトを生成するときに使用される。

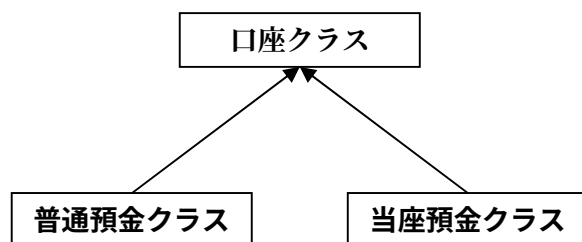
継承は、2つのクラスが“AはBの一種である”という関係であることを表し、上位のクラスを特化する手段である。例では、管理者は従業員“の一種である”し、従業員は個人“の一種である”。すべての下位クラスのオブジェクトは、上位クラスのオブジェクトであるかのように使用できる。管理者クラスのオブジェクトは、従業員クラスのオブジェクトであるかのように使用できるし、従業員クラスのオブジェクトは、個人クラスのオブジェクトであるかのように使用できる。

### 9.1.4 ポリモルフィズム



ポリモルフィズム (多態性) とは、異なるクラスに属するオブジェクト間の共通処理を可能にし、共通の操作に対して、オブジェクトごとの属するクラスに応じた振る舞いが実行時適宜選択される機能である。

以降に例を示す。



普通預金クラスは口座クラスを継承しており、普通預金クラスのオブジェクトは口座クラスのオブジェクトとしても振る舞うことができる。同様に、当座預金クラスは口座クラスを継承しており、当座預金クラスのオブジェクトは口座クラスのオブジェクトとしても振る舞うことができる。

口座クラスのメソッド、たとえば「預け入れ」は、普通預金クラスおよび当座預金クラスにも継承されている。したがって、口座クラスのオブジェクトに対して「預け入れ」操作を行う処理は、普通預金クラスのオブジェクトおよび当座預金のクラスのオブジェクトに対して共通に適用することができる。

また、各継承先クラスで継承したメソッドの実装を上書き (オーバーライド) することにより、共通処理の振る舞いを、オブジェクトの属するクラスに応じて異なったものにすることができる。

## 9.2 COBOL におけるオブジェクト指向

### 9.2.1 クラス定義

クラス定義の一般形式は次のとおりである。

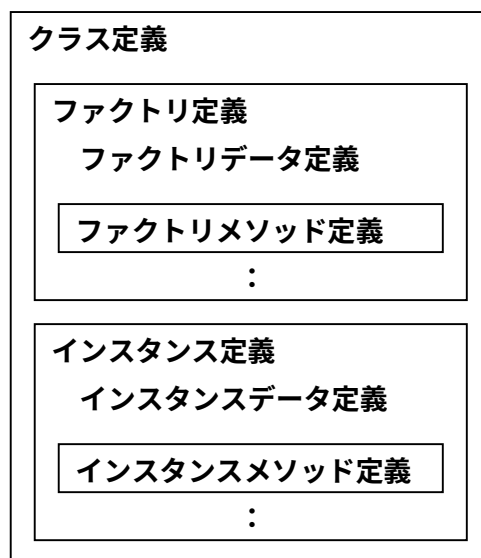
```
IDENTIFICATION DIVISION.
CLASS-ID. クラス名 1 [AS 定数 1] [INHERITS FROM クラス名 2].
[環境部]
[ファクトリ定義]
[インスタンス定義]
END CLASS クラス名 1.
```

- クラス名 1 は、このクラス定義で宣言されるクラスの名前を指定する。
- クラス名 2 は、クラス名 1 の継承元となるクラスの名前を指定する。  
クラス名 2 は、リポジトリ段落で指定する必要がある。

```
例 IDENTIFICATION DIVISION.
CLASS-ID. HUTUU_YOKIN INHERITS FROM KOUZA.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS KOUZA.
...
```

COBOL では、1つのクラスに対し、ファクトリオブジェクトとインスタンスオブジェクトという2種類のオブジェクトを定義することができる。

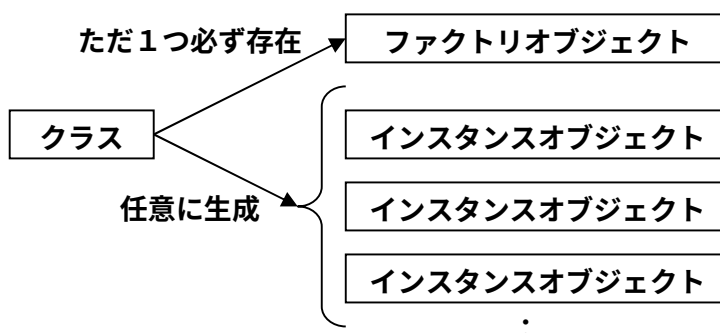
定義は、以降に示すような入れ子構造として記述する。



ファクトリオブジェクトは、各クラスに対してただ1つあり、クラスの新しいインスタンスオブジェクトを生成したり(コンストラクタ)、そのクラスのすべてのインスタンスオブジェクトに関するデータを管理したりする機能を請け負う。あるクラスのすべてのインスタンスオブジェクトは、そのクラスのファクトリオブジェクトによって生成される。

ファクトリオブジェクトは、実行単位の開始時に暗黙に生成され、実行単位の終了時に暗黙に削除される。実行中、ユーザの手続き記述によって生成、削除することはできない。

インスタンスオブジェクトは、実行中、ユーザの手続き記述によって任意に生成、削除される。おののインスタンスオブジェクトは固有のデータ値を持つ。



### 9.2.2 ファクトリ定義

ファクトリ定義は、ファクトリオブジェクトを定義する。  
そのファクトリ定義の一般形式は次のとおりである。

```
IDENTIFICATION DIVISION.
FACTORY.
[環境部]
[データ部]
[手続き部]
END FACTORY.
```

- ファクトリデータは、ファクトリ定義のデータ部にて定義する。  
ファクトリデータには、クラスで共通のファイルとクラスで共通のデータがあり、それぞれファイル節、作業場所節で定義する。  
つまり、データ部にはファイル節、作業場所節のみ記述することができ、それ以外の節（局所記憶節、連絡節など）は記述できない。  
データの生存期間はオブジェクトの生存期間と等しい。すなわち、ファクトリデータは、実行単位の開始から終了まで保持され、前のメソッド呼び出しのときの値は次のメソッド呼び出しのときに継続して使用できる。
- ファクトリメソッドは、手続き部のメソッド定義にて定義する。  
メソッド定義は複数記述できる。

```
PROCEDURE DIVISION.
[メソッド定義] …
```

### 9.2.3 インスタンス定義

インスタンス定義は、インスタンスオブジェクトを定義する。  
そのインスタンス定義の一般形式は次のとおりである。

IDENTIFICATION DIVISION.

OBJECT.

[環境部]

[データ部]

[手続き部]

END OBJECT.

- インスタンスデータは、インスタンス定義のデータ部にて定義する。  
インスタンスデータには、インスタンス固有のファイルとインスタンス固有のデータがあり、それぞれファイル節、作業場所節で定義する。  
つまり、データ部にはファイル節、作業場所節のみ記述することができ、それ以外の節（局所記憶節、連絡節など）は記述できない。  
データの生存期間はオブジェクトの生存期間と等しい。すなわち、インスタンスデータは、インスタンスオブジェクトが生成されてから削除されるまで保持され、前のメソッド呼び出しのときの値は次のメソッド呼び出しのときに継続して使用できる。
- インスタンスメソッドは、手続き部のメソッド定義にて定義する。  
メソッド定義は複数記述できる。

PROCEDURE DIVISION.

[メソッド定義] …

## 9.2.4 メソッド定義

COBOL のメソッド定義には、一般的な用途のメソッドとプロパティメソッドの 2 種類がある。本節では、一般的な用途のメソッド定義について説明し、プロパティメソッドについては 9.2.7 オブジェクトプロパティ で説明する。

一般的な用途のメソッド定義の一般形式は次のとおりである。一般形式は、ファクトリメソッドでもインスタンスメソッドでも共通である。

IDENTIFICATION DIVISION.

```
METHOD-ID. メソッド名 1 [AS 定数 1] [OVERRIDE].
```

```
[環境部]
```

```
[データ部]
```

```
[手続き部]
```

```
END METHOD メソッド名 1.
```

- メソッド名 1 は、このメソッド定義で宣言されるメソッドの名前を指定する。
- データ部には局所記憶節、連絡節のみ記述することができる。それ以外の節（ファイル節、作業場所節など）は記述できない。

データの生存期間は、一回のメソッド呼び出しの、手続き開始から終了までである。すなわち、値はメソッド呼び出し毎に初期化され、前のメソッド呼び出しのときの値は次のメソッド呼び出しのときには残っていない。

メソッド定義のデータ部に記述するのは、メソッドの手続き実行に必要な補助的なデータである。オブジェクトデータ（ファクトリデータ／インスタンスデータ）の定義は、ひとつ外側のソース単位であるファクトリ定義／インスタンス定義のデータ部で行う。

- 手続き部にはメソッドを呼び出した時に実行する手続きを記述する。記述は、プログラム定義の場合の手続き部と同様に行うことができる。
- OVERRIDE** 指定は、上位クラスから継承したメソッドの内容を上書き（オーバーライド）する。

#### 例 ①オーバーライドしない場合

```
IDENTIFICATION DIVISION.
```

```
CLASS-ID. KOUZA.
```

```
IDENTIFICATION DIVISION.
```

```
FACTORY.
```

```
PROCEDURE DIVISION.
```

```
IDENTIFICATION DIVISION.
```

```
METHOD-ID. DISP_CNAME.
```

```
PROCEDURE DIVISION.
```

```
BEGIN.
```

```
 DISPLAY "口座クラス".
```

```
END METHOD DISP_CNAME.
```

```
END FACTORY.
```

```
END CLASS KOUZA.
```

```
IDENTIFICATION DIVISION.
```

```
CLASS-ID. HUTUU_YOKIN INHERITS FROM KOUZA.
```

```
ENVIRONMENT DIVISION.
```

```
CONFIGURATION SECTION.
```

```
REPOSITORY.
```

```
 CLASS KOUZA.
```

```
END CLASS HUTUU_YOKIN.
```

HUTUU\_YOKIN クラスにはメソッドを一切定義していないが、KOUZA クラスを継承しているため、KOUZA クラスのファクトリメソッドである DISP\_CNAME メソッドは、HUTUU\_YOKIN クラスにも継承されている。したがって、DISP\_CNAME メソッドは、KOUZA クラスのインスタンスからも、HUTUU\_YOKIN クラスのインスタンスからも呼び出すことができ、どちらの場合も「口座クラス」という文字列が出力される。

## ②オーバーライドする場合

KOUZA クラスの定義は、①オーバーライドしない場合 と同じものとし、HUTUU\_YOKIN クラスのみを示す。

```
IDENTIFICATION DIVISION.
```

```
CLASS-ID. HUTUU_YOKIN INHERITS FROM KOUZA.
```

```
ENVIRONMENT DIVISION.
```

```
CONFIGURATION SECTION.
```

```
REPOSITORY.
```

```
 CLASS KOUZA.
```

```
 IDENTIFICATION DIVISION.
```

```
 FACTORY.
```

```
 PROCEDURE DIVISION.
```

```
 IDENTIFICATION DIVISION.
```

```
 METHOD-ID. DISP_CNAME OVERRIDE.
```

```
 PROCEDURE DIVISION.
```

```
 BEGIN.
```

```
 DISPLAY "普通預金クラス".
```

```
END METHOD DISP_CNAME.
```

```
END FACTORY.
```

```
END CLASS HUTUU_YOKIN.
```

HUTUU\_YOKIN クラスで、KOUZA クラスから継承した DISP\_CNAME メソッドの内容を上書き (オーバーライド) している。DISP\_CNAME メソッドは、KOUZA クラスのインスタンスからも、HUTUU\_YOKIN クラスのインスタンスからも呼び出すことができ、KOUZA クラスのインスタンスから呼び出した場合は「口座クラス」という文字列が、HUTUU\_YOKIN クラスのインスタンスから呼び出した場合は「普通預金クラス」という文字列が出力される。

もし、KOUZA クラスに他にもメソッドがあり、そのメソッドはオーバーライドしていなかった場合には、HUTUU\_YOKIN クラスでは、オーバーライドしていないメソッドに関しては KOUZA クラスと共通の手続きが、オーバーライドしているメソッドに関しては HUTUU\_YOKIN クラスで上書きした手続きが用いられる。

すなわち、継承とオーバーライドによって、あるクラスの部分的な拡張、特殊化を、共通部分を重複して記述することなしに行うことができる。

継承元クラスで定義されていないメソッド名に対して、**OVERRIDE** 指定を行うことはできない。継承先クラスのみが持つ独自のメソッドを新規に定義することになる。

継承元クラスで定義されているメソッド名に対しては、**OVERRIDE** 指定が必須である。(この仕様は、継承元クラスが持つメソッドの見落とし等により、意図せず名前が重複して、継承元のメソッドを上書きしてしまうことの防止につながる。)



### 9.2.5 オブジェクト参照

オブジェクトはそれぞれが別個のものなので、生成される時に実行時システムが作る各オブジェクトへの一意な参照値を持つ。オブジェクトの参照値は、そのオブジェクトへのポインタとして使える。すなわち、特定のオブジェクトを見分けるキーと考えられる。USAGE 句は、オブジェクトの参照値を保持するオブジェクト参照というデータ項目を定義する手段を提供する。

例 DATA DIVISION.

WORKING-STORAGE SECTION.

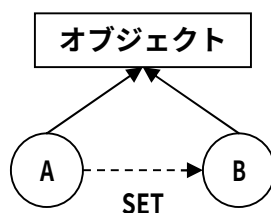
01 A USAGE OBJECT REFERENCE.

01 B USAGE OBJECT REFERENCE.

オブジェクト参照項目はプログラム定義や関数定義を含むあらゆるソース単位のデータ部に定義することができる。

オブジェクト参照項目として定義されたデータ項目は、次のように SET 文を使って、オブジェクト参照として定義された別のデータ項目と等しい値に設定できる。このとき、オブジェクトの実体が複製されるわけではない。

例 SET B TO A.



SET {一意名 3} ... TO {一意名4  
クラス名1}

- 一意名 3 は、受け取り側のオブジェクト参照項目を指定する。
- 一意名 4 は、送り出し側のオブジェクト参照項目を指定する。
- クラス名 1 を指定した場合、クラス名 1 にただ1つ属するファクトリオブジェクトが指定される。

## (1) 非限定オブジェクト参照

[USAGE IS] OBJECT REFERENCE

**OBJECT REFERENCE** 指定に何も指定しないことで、データ項目は非限定オブジェクト参照項目になる。非限定オブジェクト参照項目は、いかなるオブジェクトでも参照できる。

## (2) ファクトリオブジェクト参照／インスタンスオブジェクト参照

[USAGE IS] OBJECT REFERENCE [FACTORY OF] クラス名 1

クラス名を付けて記述されたオブジェクト参照を使うことで、たとえば従業員クラスのオブジェクトが、口座クラスのような関係のないクラスのオブジェクトとして使われることがないことを、翻訳時に保証することができる。

**FACTORY** 指定があるならば、このデータ項目の指すオブジェクトは指定クラスまたはその下位クラスのファクトリオブジェクトでなければならない。

**FACTORY** 指定がないならば、このデータ項目の指すオブジェクトは指定クラスまたはその下位クラスのインスタンスオブジェクトでなければならない。

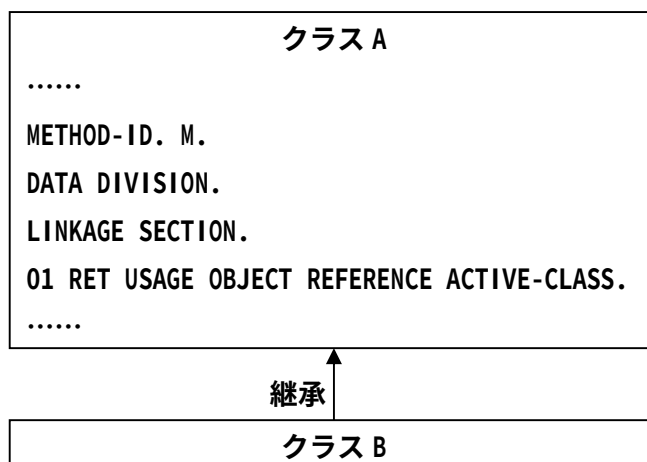
## (3) ACTIVE-CLASS

[USAGE IS] OBJECT REFERENCE [FACTORY OF] ACTIVE-CLASS

**ACTIVE-CLASS** 指定を伴うデータ項目は、クラス定義配下 (ファクトリ定義、インスタンス定義、メソッド定義) のデータ部でのみ定義できる。**ACTIVE-CLASS** 指定を書くと、メソッドが呼び出されるときに使われたオブジェクトのクラスと同じクラスのオブジェクトだけを参照することに使える。

例 次の例において、クラス A のオブジェクトからメソッド M を呼び出した場合、オブジェクト参照 RET はクラス A のオブジェクトを参照するデータ項目となり、クラス B のオブジェクトからメソッド M を呼び出した場合、オブジェクト参照 RET はクラス B のオブジェクトを参照するデ

一タ項目となる。



ACTIVE-CLASS へのオブジェクト参照は、返却項目の場合に特別な重要性を持つ。この記述は、9.2.8 標準クラス に記載する New メソッドを定義するのに必要であり、また、この機能によって、クラス階層の定義に適合しながらクラスの新しいインスタンスオブジェクトを生成するファクトリメソッドを書くことができる。

#### (4) 既定義オブジェクト参照

既定義オブジェクト参照は、暗黙的に生成されるデータ項目であり、一意名 NULL, SELF, SUPER のいずれかで参照される。

##### (ア) NULL

オブジェクト参照値 NULL は、どんなオブジェクトを指すものでもないことを示す。

##### (イ) SELF

##### (ウ) SUPER

SELF および SUPER は、メソッドの手続き中でのみ参照することができ、メソッドを呼び出すために使用されたオブジェクトを指す。特に SUPER は、メソッド内で上位クラスのメソッドを呼び出すためのオブジェクトとしてだけ指定することができ、メソッドの解決において、メソッド呼び出しを含んでいるクラスおよびそのあらゆるクラス下位クラスにおけるメソッド定義(オーバーライド)をすべて無視する。

## 9.2.6 メソッド呼び出し

メソッドの呼び出しは INVOKE 文で行う。INVOKE 文の一般形式は次のとおりである。

|                                                              |
|--------------------------------------------------------------|
| <b>INVOKE</b> {一意名1<br>クラス名1} 定数 1 [USING 指定] [RETURNING 指定] |
|--------------------------------------------------------------|

- 一意名 1 は、呼び出しに用いるオブジェクト参照項目を指定する。  
インスタンスオブジェクト参照を指定した場合、インスタンスメソッドを呼び出す。  
ファクトリオブジェクト参照を指定した場合、ファクトリメソッドを呼び出す。  
本コンパイラでは、非限定オブジェクト参照を指定することはできない。
- クラス名 1 を指定した場合、クラス名 1 のファクトリオブジェクト参照が指定される。
- 定数 1 は、呼び出すメソッド名を指定する。

### 9.2.7 オブジェクトプロパティ

オブジェクトプロパティは、オブジェクトデータの設定、参照に特化した、特殊構文によるメソッド呼び出しを提供する機能である。

例 あるファクトリないしインスタンスが持つオブジェクトデータ NUM を設定、参照することを考える。  
オブジェクトデータはカプセル化されているため、NUM を設定、参照するためのメソッドを定義する必要がある。

#### ①オブジェクトプロパティを利用しない場合

DATA DIVISION.

WORKING-STORAGE SECTION.

01 NUM PIC 9(3).

PROCEDURE DIVISION.

IDENTIFICATION DIVISION.

METHOD-ID. SET\_NUM.

DATA DIVISION.

LINKAGE SECTION.

01 IN\_NUM PIC 9(3).

PROCEDURE DIVISION USING IN\_NUM.

```

BEGIN.
 MOVE IN_NUM TO NUM.

END METHOD SET_NUM.

```

```

IDENTIFICATION DIVISION.
METHOD-ID. GET_NUM.

DATA DIVISION.
LINKAGE SECTION.
01 OUT_NUM PIC 9(3).

PROCEDURE DIVISION RETURNING OUT_NUM.
BEGIN.
 MOVE NUM TO OUT_NUM.

END METHOD GET_NUM.

```

SET\_NUM は、引数として受け取った値をオブジェクトデータに設定するメソッドである。GET\_NUM は、オブジェクトデータの値を返却項目とするメソッドである。

オブジェクトを利用する側の記述は次のようになる。

```

ENVIRONMENT DIVISION.
CLASS

PROCEDURE DIVISION.
.....

 INVOKE OBJREF "SET_NUM" USING 123.

 INVOKE OBJREF "GET_NUM" RETURNING TMP.
 DISPLAY TMP.

```

②オブジェクトプロパティを利用する場合  
クラス側の記述は次のようになる。

```
DATA DIVISION.
WORKING-STORAGE SECTION.
01 NUM PIC 9(3) PROPERTY.
```

利用する側の記述は次のようになる。

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
 CLASS
 PROPERTY NUM.

PROCEDURE DIVISION.
.....

 MOVE 123 TO NUM OF OBJREF.

 DISPLAY NUM OF OBJREF.
```

オブジェクトプロパティを設定、参照するための機構は、プロパティメソッド (SET プロパティメソッド / GET プロパティメソッド) である。プロパティメソッドは暗黙的に定義する方法と明示的に定義する方法がある。

#### (1) 暗黙的プロパティメソッド

オブジェクトデータの定義に **PROPERTY** 句を指定することで、そのデータに引数値を設定する SET プロパティメソッド、そのデータの値を返却する GET プロパティメソッドが暗黙的に定義され、そのデータ項目がオブジェクトプロパティとして使用できるようになる。

$$\underline{\text{PROPERTY}} \left[ \text{WITH } \underline{\text{NO}} \left\{ \begin{array}{c} \underline{\text{GET}} \\ \underline{\text{SET}} \end{array} \right\} \right]$$

- NO GET 指定を行うと、SET プロパティメソッドのみが定義される。
- NO SET 指定を行うと、GET プロパティメソッドのみが定義される。

#### (2) 明示的プロパティメソッド

メソッド定義で次の一般形式を用いることにより、プロパティメソッドを明示的に定義することができる。

```
IDENTIFICATION DIVISION.
```

```
METHOD-ID. {GET
SET} PROPERTY プロパティ名 1 [OVERRIDE].
```

```
[環境部]
```

```
[データ部]
```

```
[手続き部]
```

```
END METHOD.
```

- GET を指定すると、このメソッドは GET プロパティメソッドとなり、引数を持たず、返却項目を持つメソッドとしなければならない。手続き内容は自由に記述してよい。
- SET を指定すると、このメソッドは SET プロパティメソッドとなり、引数をただ1つ持ち、返却項目を持たないメソッドとしなければならない。手続き内容は自由に記述してよい。
- プロパティ名 1 は、このメソッド定義で宣言されるプロパティ名を指定する。  
同じプロパティに対し、プロパティメソッドの暗黙的定義と明示的定義を同時に行うことはできない。すなわち、プロパティ名 1 がオブジェクトデータの名前である場合、データ定義に PROPERTY 句が指定されていない。

明示的定義の場合、暗黙的定義よりもプロパティメソッドの機能を自由に拡張することが可能である。

例 DATA DIVISION.

```
WORKING-STORAGE SECTION.
```

```
01 NUM PIC 9(3).
```

```
PROCEDURE DIVISION.
```

```
IDENTIFICATION DIVISION.
```

```
METHOD-ID. SET PROPERTY NUM.
```

```
DATA DIVISION.
```

```
LINKAGE SECTION.
```

```
01 IN_NUM PIC 9(3).
```

```

PROCEDURE DIVISION USING IN_NUM.
BEGIN.
 IF 50 <= IN_NUM AND IN_NUM <= 100 THEN
 MOVE IN_NUM TO NUM
 END-IF.

END METHOD.

```

この例では、受け取った値が 50 以上 100 以下のときのみオブジェクトデータ NUM を変更しており、設定できる値の範囲を保証している。

暗黙的定義と異なり、宣言するプロパティは必ずしも1つのオブジェクトデータに紐付ける必要はなく、プロパティ名はオブジェクトデータの名前でなくてもよい。

例 DATA DIVISION.

```
WORKING-STORAGE SECTION.
```

```
01 NUM1 PIC 9(3).
```

```
01 NUM2 PIC 9(3).
```

```
01 NUM3 PIC 9(3).
```

```
PROCEDURE DIVISION.
```

```
IDENTIFICATION DIVISION.
```

```
METHOD-ID. GET PROPERTY TOTAL.
```

```
DATA DIVISION.
```

```
LINKAGE SECTION.
```

```
01 OUT_TOTAL PIC 9(4).
```

```
PROCEDURE DIVISION RETURNING OUT_TOTAL.
```

```
BEGIN.
```

```
 COMPUTE OUT_TOTAL = NUM1 + NUM2 + NUM3
```

```
END METHOD.
```

この例では、オブジェクトデータ NUM1, NUM2, NUM3 の総和を表すプロパティ TOTAL を定義



している。

オブジェクトプロパティを利用する側の一般形式を以降に示す。プロパティメソッドが暗黙的定義か明示的定義かに依らず、利用する側の記述は同じになる。

プロパティ名 1 OF  $\left\{ \begin{array}{l} \text{一意名1} \\ \text{クラス名1} \end{array} \right\}$

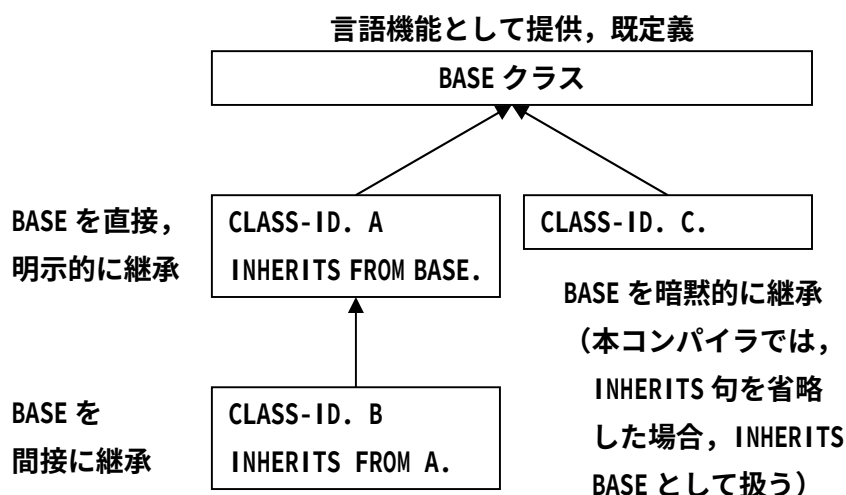
- プロパティ名 1 は、設定、参照するプロパティ名を指定する。  
プロパティ名 1 は、リポジトリ段落で指定する必要がある。
- 一意名 1 は、プロパティを持つオブジェクト参照項目を指定する。
- クラス名 1 を指定した場合、クラス名 1 のファクトリオブジェクト参照が指定される。

オブジェクトプロパティは、単なるデータ項目であるかのように式の中で使用できるが、実際にはメソッド呼び出しを隠蔽したものであり、メソッド呼び出しおよびメソッド内の手続き実行を伴っている。

オブジェクトプロパティが受け取り側項目として記述された場合は SET プロパティメソッドが、送り出し側項目として記述された場合は GET プロパティメソッドが、呼び出される。

## 9.2.8 標準クラス

標準クラス **BASE** は、言語機能として提供する既定義のクラスである。**BASE** クラスは、すべてのクラスが直接的または間接的に、明示的にまたは暗黙的に継承する、クラス階層の頂点となるクラスである。



**BASE** クラスは次のメソッドを提供する。**BASE** クラスはすべてのクラスの上位クラスであるため、これらのメソッドはすべてのクラスに備わっているメソッドとなる。

- **New** メソッド
- **FactoryObject** メソッド
- **CSRelease** メソッド
- **CSReleaseSelf** メソッド
- **CSReleaseInThread** メソッド
- **CSReleaseAll** メソッド

本コンパイラでは、標準クラスのメソッドをオーバーライドすることはできない。また、メソッド名は大文字小文字を区別しないため、"NEW", "new", "nEw"等のメソッド名を持つメソッドは同一視されるため新たに定義することはできない。

なお、**CSRelease** メソッド、**CSReleaseSelf** メソッド、**CSReleaseInThread** メソッド、**CSReleaseAll** メソッドは、本コンパイラ独自追加のメソッドである。

**(1) New メソッド**

New メソッドは、クラスの新しいインスタンスオブジェクトを生成するためのファクトリメソッドである。

引数は持たず、新規生成したインスタンスオブジェクトの参照を返却する。返却項目のオブジェクト参照は ACTIVE-CLASS であり、New メソッドを呼び出したクラスのインスタンスが生成される。

**例 01 OBJREF USAGE OBJECT REFERENCE KOUZA.**

.....

**INVOKE KOUZA "New" RETURNING OBJREF.**

インスタンス生成と同時に何らかの初期化処理や共通処理を行い場合は、New メソッドの呼び出し、初期化処理、共通処理を一括して行うファクトリメソッド (コンストラクタ) を定義するとよい。クラスを利用する側は、直接 New メソッドを呼ぶのではなく、そのファクトリメソッドを呼ぶようにする。

**例 IDENTIFICATION DIVISION.**

**FACTORY.**

**DATA DIVISION.**

**WORKING-STORAGE SECTION.**

**01 KOUZA\_SOUSUU PIC 9(5) VALUE ZERO.**

**PROCEDURE DIVISION.**

**IDENTIFICATION DIVISION.**

**METHOD-ID. NEW\_KOUZA.**

**DATA DIVISION.**

**LINKAGE SECTION.**

**01 OUT\_OBJREF USAGE OBJECT REFERENCE ACTIVE-CLASS.**

**PROCEDURE DIVISION RETURNING OUT\_OBJREF.**

**BEGIN.**

**INVOKE SELF "New" RETURNING OUT\_OBJREF.**

```

 ADD 1 TO KOUZA_SOUSUU.
 MOVE KOUZA_SOUSUU TO ID OF OUT_OBJREF.

END METHOD NEW_KOUZA.

```

この例では、新しいインスタンスを生成すると同時に、インスタンスのプロパティ「ID」に通し番号を設定するファクトリメソッドを定義している。このクラスを利用する側は、次のようになる。

```

01 OBJREF USAGE OBJECT REFERENCE KOUZA.
.....

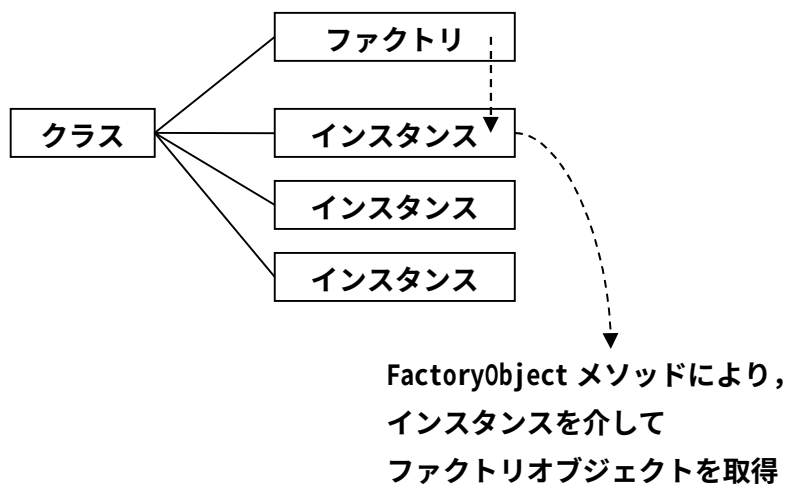
INVOKE KOUZA "NEW_KOUZA" RETURNING OBJREF.

```

## (2) FactoryObject メソッド

**FactoryObject** メソッドはインスタンスメソッドである。

引数は持たず、呼び出したインスタンスが属するクラスのファクトリオブジェクト参照を返却する。



```

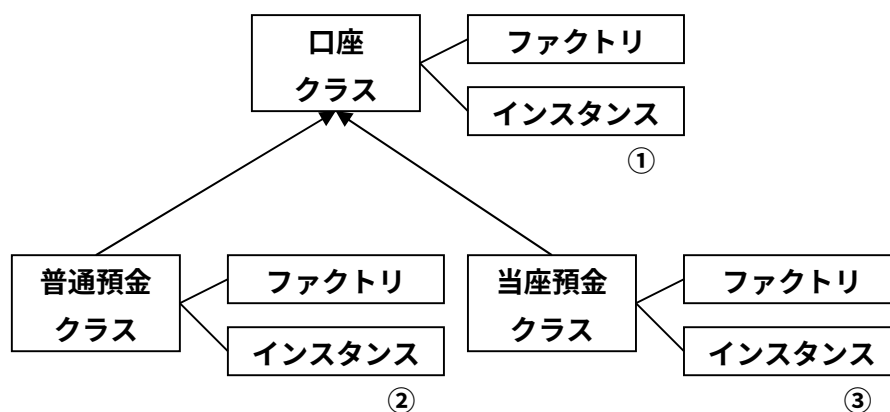
例 01 OBJ USAGE OBJECT REFERENCE KOUZA.
01 FOBJ USAGE OBJECT REFERENCE FACTORY OF KOUZA.
.....

INVOKE OBJ "FactoryObject" RETURNING FOBJ.
INVOKE FOBJ "(ファクトリメソッド名)".

```

この例では、OBJ の属するクラスが、たとえば KOUZA クラスの下位クラスである HUTUU\_YOKIN クラスに属すると分かっているのであれば、FactoryObject メソッドを用いず「INVOKE HUTUU\_YOKIN "(ファクトリメソッド名)"とすればよい。

FactoryObject メソッドは、OBJ の属するクラスが、たとえば HUTUU\_YOKIN クラスであるか TOUZA\_YOKIN クラスであるか分からない時に役に立つ。



OBJ は①，②，③のいずれも指すことができるため、  
対応するファクトリオブジェクトを実行時に取得するには  
FactoryObject メソッドを利用する

### (3) CSRelease メソッド

CSRelease メソッドは、クラスのインスタンスオブジェクトを削除するためのファクトリメソッドである。

削除するオブジェクト参照を引数にとり、削除結果 (失敗:0, 成功:1) を示す COMP-2 項目を返却する。

例 INVOKE KOUZA "CSRelease" USING OBJREF RETURNING RESULT.

### (4) CSReleaseSelf メソッド

CSReleaseSelf メソッドはインスタンスメソッドである。

引数は持たず、呼び出したインスタンス自身を削除し、削除結果 (失敗:0, 成功:1) を示す COMP-2 項目を返却する。

例 INVOKE OBJREF "CSReleaseSelf" RETURNING RESULT.

### (5) CSReleaseInThread メソッド

CSReleaseInThread メソッドは、呼び出されたスレッド内で生成したすべてのオブジェクトを一

括削除するためのファクトリメソッドである。

引数は持たず、削除結果 (失敗:0, 成功:1) を示す COMP-2 項目を返却する。

**例 INVOKE BASE "CSReleaseInThread" RETURNING RESULT.**

このメソッドは、どのクラスのファクトリオブジェクトから呼び出されたかに関係なく作用する。すなわち、上の例で **INVOKE KOUZA "CSReleaseInThread"...** と記述しても同じ動作となる。(削除対象が KOUZA クラスのオブジェクトだけになることはない。) したがって、混乱を招かないよう **BASE** クラスを指定する方がよい。

また、削除対象が他のスレッドからも参照されているオブジェクトであるかに依らずに削除を行う。

#### (6) CSReleaseAll メソッド

**CSReleaseAll** メソッドは、プロセス内で生成したすべてのオブジェクトを一括削除するためのファクトリメソッドである。

引数は持たず、削除結果 (失敗:0, 成功:1) を示す COMP-2 項目を返却する。

**例 INVOKE BASE "CSReleaseAll" RETURNING RESULT.**

このメソッドは、どのクラスのファクトリオブジェクトから呼び出されたかに関係なく作用する。すなわち、上の例で **INVOKE KOUZA "CSReleaseAll"...** と記述しても同じ動作となる。(削除対象が KOUZA クラスのオブジェクトだけになることはない。) したがって、混乱を招かないよう **BASE** クラスを指定する方がよい。

本コンパイラでは、あるオブジェクトが実行単位のその後の実行に関与しないとする判断は、手続き中で **CSRelease** メソッド、**CSReleaseSelf** メソッド、**CSReleaseInThread** メソッド、**CSReleaseAll** メソッドを呼び出すことで明示的に行う。

オブジェクトの生存期間を管理するうえでの注意事項を以降に示す。

削除済みのオブジェクト参照に対してメソッドを呼び起こしたり、二重に削除を行ったりしたときの動作は不定である。(一般に、不正アクセスの結果、データ破壊やクラッシュを招く可能性がある。) オブジェクトを削除した際には合わせて、削除したオブジェクトを参照していた項目に **NULL** 値を設定しておくといふ。**NULL** オブジェクト参照に対するメソッド呼び出しは、実行時エラーとして実行単位内で処理される。

生成したオブジェクトの削除漏れは、一般にメモリリークとなる。削除漏れが起きやすいと思われるケースを以降に示す。

- 実行時エラーの発生によって異常終了する場合  
USE 文で指定した宣言手続きの中で、オブジェクトを削除することができる。名前の有効範囲外（入れ子プログラムの中など）のオブジェクト参照項目は、CSRelease メソッド、CSReleaseSelf メソッドの呼び出しに指定できないため、CSReleaseInThread メソッド、CSReleaseAll メソッドを用いることで削除できる。
- CANCEL 文によってオブジェクト参照項目を持つプログラムが初期化される場合  
元々指していたオブジェクトが参照できなくなるため、事前に CSRelease メソッド、CSReleaseSelf メソッドによって削除しておくか、後で CSReleaseInThread メソッド、CSReleaseAll メソッドによって削除する必要がある。  
プログラム定義に INITIAL 句が指定されている場合も同様である。

### 9.2.9 オブジェクト指向での適合

オブジェクトの適合とは、オブジェクトが持つインタフェース（オブジェクトに対して可能な操作、振る舞い）の整合性に関する概念であり、オブジェクト自身が属するクラスのインタフェースとは異なるインタフェースに従った使い方が許されるようにする機能である。

具体的には、下位クラスのオブジェクトは上位クラスのオブジェクトとしても使用できる。これは、下位クラスのオブジェクトが、上位クラスのオブジェクトが持つメソッドをすべて備えており、上位クラスのオブジェクトの能力を包含しているからである。

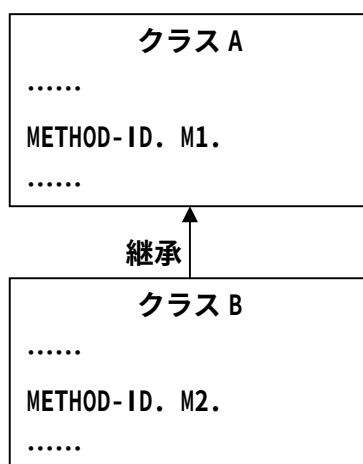
適合検査は、SET 文や引数、返却項目を介したオブジェクト参照項目間の受け渡しにおいて、送り出し側と受け取り側のインタフェースの包含関係を検査するものである。基本的に適合検査は翻訳時に行うが、オブジェクトビューに対する適合検査は実行時に行う。

オブジェクトビューは、オブジェクト参照を指定どおりの記述を持つかのように扱う機能である。

|                                                     |
|-----------------------------------------------------|
| 一意名 1 <u>AS</u> { [FACTORY OF] クラス名1 }<br>UNIVERSAL |
|-----------------------------------------------------|

- 一意名 1 の参照は翻訳時に、AS 指定のとおり記述されたかのように扱われる。  
一意名 1 が参照するオブジェクトが、実際に AS 指定の記述に適合するかは、実行時に検査が行われる。適合しない場合、実行時エラーとなる。
- UNIVERSAL を指定すると、一意名 1 はいかなる選択指定もつかない USAGE OBJECT REFERENCE と記述された扱い (非限定オブジェクト参照) となる。

例 クラス B はクラス A を継承しており、クラス A にはメソッド M1 が、クラス B にはメソッド M2 が定義されているものとする。



```
01 OBJREF_A USAGE OBJECT REFERENCE A.
```

```
01 OBJREF_B USAGE OBJECT REFERENCE B.
```

```
...
```

```
SET OBJREF_A TO OBJREF_B. ... ①
```

```
SET OBJREF_B TO OBJREF_A. ... ②
```

```
INVOKE OBJREF_A "M1". ... ③
```

```
INVOKE OBJREF_A "M2". ... ④
```

```
INVOKE OBJREF_B "M1". ... ⑤
```

```
INVOKE OBJREF_B "M2". ... ⑥
```

- ① 記述可能である。OBJREF\_A はクラス A のオブジェクトもクラス B のオブジェクトも扱うことができる。
- ② 翻訳エラーとなる。OBJREF\_B はクラス B のオブジェクトしか扱えないが、OBJREF\_A はクラス A のオブジェクトを参照している可能性がある。  
実際の実行時に OBJREF\_A がクラス B のオブジェクトを参照しているならば、この SET 文は



実行上問題ないが、翻訳時の適合検査は静的な解釈に基づいて行われる。

オブジェクトビューを利用し、`SET OBJREF_B TO OBJREF_A AS B.` とすることで、実行時に適合検査を行うように変更できる。

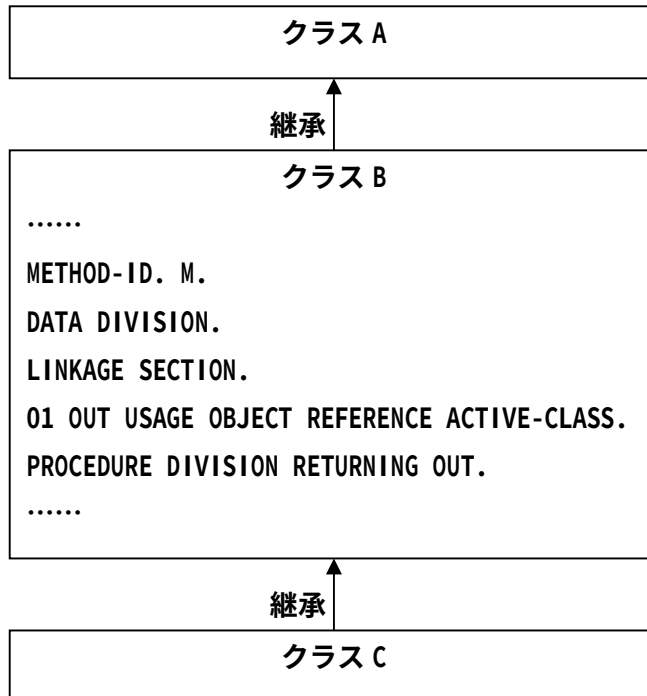
- ③ 記述可能である。OBJREF\_A はクラス A のオブジェクトもクラス B のオブジェクトも扱うことができるが、いずれのクラスもメソッド M1 を備えている。
- ④ 翻訳エラーとなる。メソッド M2 はクラス B のみが持つメソッドであるが、OBJREF\_A はクラス A のオブジェクトを参照している可能性がある。

実際の実行時に OBJREF\_A がクラス B のオブジェクトを参照しているならば、この INVOKE 文は実行上問題ないが、翻訳時の適合検査は静的な解釈に基づいて行われる。

オブジェクトビューを利用し、`INVOKE OBJREF_A AS B "M2".` とすることで、実行時に適合検査を行うように変更できる。

- ⑤ 記述可能である。クラス B はクラス A のメソッド M1 を継承しており、クラス A のインタフェースを包含している。
- ⑥ 記述可能である。

例 クラス B はクラス A を、クラス C はクラス B を継承しており、クラス B のメソッド M は、`ACTIVE-CLASS` のオブジェクト参照を返却するものとする。



```
01 OBJREF_A USAGE OBJECT REFERENCE A.
```

```
01 OBJREF_B USAGE OBJECT REFERENCE B.
```

01 OBJREF\_C USAGE OBJECT REFERENCE C.

01 OBJREF\_B2 USAGE OBJECT REFERENCE B.

...

|                                          |       |
|------------------------------------------|-------|
| INVOKE OBJREF_B2 "M" RETURNING OBJREF_A. | ... ① |
| INVOKE OBJREF_B2 "M" RETURNING OBJREF_B. | ... ② |
| INVOKE OBJREF_B2 "M" RETURNING OBJREF_C. | ... ③ |

- ① 記述可能である。メソッド M は、OBJREF\_B2 が扱えるクラス B またはクラス C のオブジェクトから呼び出されるため、ACTIVE-CLASS の返却項目もクラス B またはクラス C のオブジェクトとなる。OBJREF\_A はいずれも扱える。
- ② ①と同様に、記述可能である。
- ③ 翻訳エラーとなる。ACTIVE-CLASS がクラス B の場合、OBJREF\_C で受けることはできない。実際の実行時に OBJREF\_B2 がクラス C のオブジェクトを参照しているならば、この INVOKE 文は実行上問題ないが、翻訳時の適合検査は静的な解釈に基づいて行われる。オブジェクトビューを利用し、INVOKE OBJREF\_B2 AS C "M" RETURNING OBJREF\_C. とすることで、実行時に適合検査を行うように変更できる。

## 9.2.10 オブジェクト指向でのファイル

### (1) インスタンスオブジェクト中のファイル

あるインスタンスオブジェクト中にファイルが指定されている、とは、インスタンス定義中にファイル管理段落とファイル節とがあることを意味する。1つ以上のインスタンスメソッド中に OPEN, CLOSE, READ, WRITE などのファイル入出力文があることになる。ファイルに関連するレコードには、すべてのインスタンスメソッドからアクセス可能である。

インスタンス定義中にファイルの指定のあるクラスが継承された場合は、直接または間接の子孫のそれぞれもまた、そのファイルの指定を継承する。これらの下位クラスのおのおののインスタンスは、EXTERNAL 句の指定がファイルにないかぎり、各自に固有なファイル結合子をもつ。動的ファイル割り当てをファイル管理記述項中に指定すると、あるクラスをある構造の論理ファイルを定義するのに用いることができ、各インスタンスは、それぞれ異なる物理ファイルをインスタンス固有のファイル結合子と関連付けて、その物理ファイルに対して入出力を行うことができる。

### (2) ファクトリオブジェクト中のファイル

ファクトリオブジェクト中にファイルが指定されている、とは、ファクトリ定義中にファイル管理段落とファイル節とがあることを意味する。1つ以上のファクトリメソッド中に OPEN, CLOSE, READ, WRITE などのファイル入出力文があることになる。ファイルに関連するレコードには、すべての

ファクトリメソッドからアクセス可能である。

ファクトリ定義中にファイルの指定のあるクラスが継承された場合は、直接または間接の子孫のそれぞれもまた、そのファイルの指定を継承する。これらの下位クラスのファクトリは、EXTERNAL 句の指定がファイルにないかぎり、各自に固有なファイル結合子をもつ。

## 9.3 プログラム例

1 個のメインプログラムと Account (口座) クラスとで構成された、非常に単純な銀行業務システムの例を示す。ただし、COBOL のオブジェクト指向の全機能をここで解説しているわけではない。

### (1) メインプログラム

大半の場合オブジェクト指向業務適用システムには処理を開始するために、従来の書き方のプログラムが 1 個ある。この例の銀行業務システムでは BANKMAIN がこの役割を果たしている。

```
IDENTIFICATION DIVISION.
PROGRAM-ID. BANKMAIN.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
 CLASS Account.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 an-object USAGE OBJECT REFERENCE Account.
PROCEDURE DIVISION.
go-now.
 INVOKE Account "newAccount" RETURNING an-object.
 INVOKE an-object "displayUI".
 INVOKE Account "CSRelease" USING an-object.
 SET an-object to NULL.
END PROGRAM BANKMAIN.
```

### (2) 口座クラス

このクラスにはファクトリメソッドが 3 個ある。

- newAccount は、口座オブジェクトの新しいインスタンス 1 個を創生する。
- addAccount は、1 を「口座の総数」の値に加える。

- **removeAccount** は, 1 を「口座の総数」の値から引く。  
また, このクラスにはインスタンスメソッドが 5 個ある。
- **displayUI** は, 口座残高の値の表示その他の機能を, 利用者の要求に基づいて行う。
- **balance** は, 口座の残高を照会する。
- **deposit** は, ある金額を, 口座の現残高に加える。
- **withdraw** は, ある金額を, 口座の現残高から引く。
- **initializeAccount** は, 初期値をインスタンスデータに転記する。

IDENTIFICATION DIVISION.

CLASS-ID. Account.

IDENTIFICATION DIVISION.

FACTORY.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 number-of-accounts PIC 9(5) VALUE ZERO. \*> [ 口座の総数 ]

PROCEDURE DIVISION.

IDENTIFICATION DIVISION.

METHOD-ID. newAccount. \*> [「口座を新設」する方法]

DATA DIVISION.

LOCAL-STORAGE SECTION.

LINKAGE SECTION.

01 an-object USAGE OBJECT REFERENCE ACTIVE-CLASS.

PROCEDURE DIVISION RETURNING an-object.

begin-here.

    INVOKE SELF "new" RETURNING an-object.

    INVOKE an-object "initializeAccount" USING BY CONTENT number-of-accounts.

    EXIT METHOD.

END METHOD newAccount.

IDENTIFICATION DIVISION.

METHOD-ID. addAccount. \*> [「口座を追加」する方法]

PROCEDURE DIVISION.

method-start.

    ADD 1 TO number-of-accounts.

    EXIT METHOD.

END METHOD addAccount.

```

IDENTIFICATION DIVISION.
METHOD-ID. removeAccount. *> [「口座を削除」する方法]
PROCEDURE DIVISION.
main-entry.
 SUBTRACT 1 FROM number-of-accounts.
 EXIT METHOD.
END METHOD removeAccount.
END FACTORY.

```

```

IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 account-balance PIC S9(9)V99. *> [口座残高]
01 account-number PIC X(5). *> [口座番号]
01 the-date PIC 9(8). *> [日付]
PROCEDURE DIVISION.
IDENTIFICATION DIVISION.
METHOD-ID. displayUI. *> [「利用者インタフェースの表示」方法]
DATA DIVISION.
LOCAL-STORAGE SECTION.
01 in-data.
 03 action-type PIC X. *> [取り扱い種別]
 03 in-amount PIC S9(9)V99. *> [取り扱い金額]
PROCEDURE DIVISION.
method-start.
 DISPLAY "預け入れ(D),残高照会(B),引き出し(W)のどれかを入れてください".
 ACCEPT in-data.
 EVALUATE action-type
 WHEN "D"
 PERFORM get-amount
 INVOKE SELF "deposit" USING in-amount *> [「預け入れ」]
 WHEN "W"
 PERFORM get-amount
 INVOKE SELF "withdraw" USING in-amount *> [「引き出し」]

```

```

 WHEN "B"
 INVOKE SELF "balance" *> [「残高照会」]
 WHEN OTHER
 DISPLAY "正しい取り扱い種別を入れてください。"
 EXIT METHOD
END-EVALUATE.
EXIT METHOD.

get-amount.
 DISPLAY "金額を 9(9).99 で入れてください".
 ACCEPT in-amount.
END METHOD displayUI.

```

```

IDENTIFICATION DIVISION.
METHOD-ID. balance. *> [「残高照会」の方法]
DATA DIVISION.
LOCAL-STORAGE SECTION.
01 display-balance PIC ¥ZZZ,ZZZ,ZZ9.99B-. *> [残高の表示]
PROCEDURE DIVISION.
disp-balance.
 MOVE account-balance TO display-balance.
 DISPLAY "あなたの口座残高は: " display-balance.
 EXIT METHOD.
END METHOD balance.

```

```

IDENTIFICATION DIVISION.
METHOD-ID. deposit. *> [「預け入れ」の方法]
DATA DIVISION.
LINKAGE SECTION.
01 in-deposit PIC S9(9)V99. *> [預け入れ額]
PROCEDURE DIVISION USING in-deposit.
make-deposit.
 ADD in-deposit TO account-balance.
 EXIT METHOD.
END METHOD deposit.

```

```

IDENTIFICATION DIVISION.
METHOD-ID. withdraw. *> [「引き出し」の方法]
DATA DIVISION.
LINKAGE SECTION.
01 in-withdraw PIC S9(9)V99. *> [引き出し額]
PROCEDURE DIVISION USING in-withdraw.
withdraw-start.
 IF account-balance >= in-withdraw
 SUBTRACT in-withdraw FROM account-balance
 ELSE
 DISPLAY "あなたの残高は不十分です"
 END-IF.
EXIT METHOD.
END METHOD withdraw.

```

```

IDENTIFICATION DIVISION.
METHOD-ID. initializeAccount. *> [「口座を初期化」する方法]
DATA DIVISION.
LINKAGE SECTION.
01 new-account-number PIC 9(5). *> [新たな口座番号]
PROCEDURE DIVISION USING new-account-number.
Begin-initialization.
 MOVE ZERO TO account-balance.
 MOVE new-account-number TO account-number.
 MOVE FUNCTION CURRENT-DATE (1:8) TO the-date.
 EXIT METHOD.
END METHOD initializeAccount.

```

```

END OBJECT.

```

```

END CLASS Account.

```

## 9.4 C++からの利用方法

C++から COBOL で定義したクラスを利用する方法を以下に示す。

### (1) COBOL で定義したクラスやメソッドの定義を参照するには

まず、COBOL で定義したクラスを C++から呼び出すためには、C++のクラスやメソッドから COBOL で定義したクラスやメソッドの定義を参照できるようにする必要がある。

そのために -CG オプションを指定して、リポジトリを生成する。

リポジトリを生成した際にクラス名を含む .rph を拡張子に持つファイルが生成されるため、そのファイルを参照する C++ のソースファイルにインクルードしてください。

また、COBOL で定義したクラスは、標準クラス `BASE` を継承しているため、必ず `'CSINTERFACE.h'` をインクルードしてください。

64bit のアプリケーションを作成する場合は、`"CSINTERFACE.h"` をインクルードした C++ ソースファイルをコンパイルする際に `A_CBL_64` マクロを定義する必要があります。C++ ソースファイルをコンパイルするコンパイラ (注 1) のオプションに `-DA_CBL_64` を追加してください。

### 注意

(注 1) [Enterprise for Windows] [Standard] 使用する c コンパイラは Visual C++ コンパイラである。

[Enterprise for Linux] 使用する c コンパイラは gcc(1) コンパイラである。

### (2) COBOL で定義したクラスやメソッドを利用するには

次に、C++からの呼び出しについて説明する。

C++で COBOL で定義したクラスのインスタンスを生成するには、COBOL のファクトリオブジェクトにある `New` メソッドを呼び出してください。

(COBOL で定義したクラスに対して、`New` メソッドを呼び出すことはできません)

例) COBOL で `Account` クラスを定義している場合

```
Account* obj = (Account*)Account::A_CBL_FactoryObject()->New(NULL);
```

生成したインスタンスを用いて、インスタンスメソッドを呼び出すには、生成したインスタンスを指定して呼び出してください。



例) Account クラスのインスタンスメソッドである **displayUI** メソッドを呼び出す場合

```
obj->displayUI();
```

ファクトリメソッドを呼び出すには、クラスのファクトリオブジェクトを指定して呼び出してください。

例) Account クラスのファクトリメソッドである **newAccount** メソッドを呼び出す場合

```
Account::A_CBL_FactoryObject()->newAccount();
```

なお、プロパティメソッドは、下記の命名規則に従った名前のメソッドであるので、その名前で呼び出してください。

**GET** プロパティメソッド・・・**A\_CBL\_GET\_**プロパティ名

**SET** プロパティメソッド・・・**A\_CBL\_SET\_**プロパティ名

COBOL で定義したクラスのインスタンスを解放するには、COBOL のファクトリメソッドにある **CSRelease** メソッドを呼び出してください。

(COBOL で定義したクラスに対して、**delete** メソッドを呼び出すことはできません)

例) Account クラスのインスタンスを解放する場合

```
Account::A_CBL_FactoryObject()->CSRelease(obj);
```

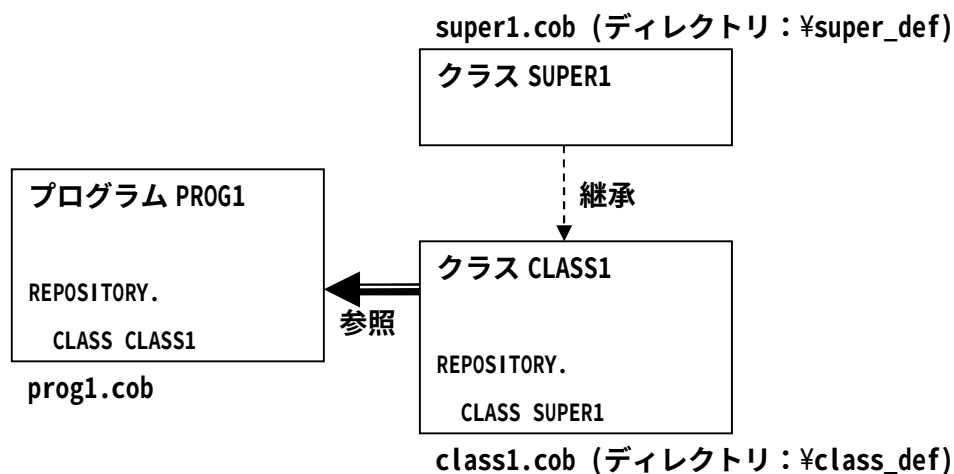
## 9.5 オブジェクト指向機能の利用方法

【Enterprise for Linux】本節の以降の説明文では、オブジェクトファイル名の拡張子「.obj」を「.o」、ディレクトリのパス区切り文字「¥」を「/」に読み替えること。

クラス定義は、そのクラスを使うプログラム定義やクラス定義などと同じ翻訳群（同一翻訳群）の中に記述してもよいし、1つのクラスの定義だけで1翻訳群として記述してもよい。

しかし、利用面から考えると、1つのクラスの定義だけで1翻訳群としておくのが望ましい。

それは、同一翻訳群の中にクラスを記述すると、生成されるオブジェクトファイル名が、本コンパイラの命名規則により「ソースファイル名\_利用者定義関数名.obj」となり、他の翻訳群から利用しにくいためである（同一翻訳群を翻訳する場合のオブジェクトファイル名の生成規則については、「1.5 プログラムのリンク」を参照）。



以下、クラス SUPER1（ソースファイル名は super1.cob）から継承された CLASS1（ソースファイル名は class1.cob）というクラスが、それぞれ別翻訳群で定義されており、さらに、プログラム PROG1（ソースファイル名は prog1.cob）において、クラス CLASS1 が参照されている場合の翻訳方法を示す。

まず、継承元クラス SUPER1 は、下記のように翻訳する。

```
cob -c -CG super1.cob
```

これにより、カレントディレクトリ（¥super\_def とする）にクラス SUPER1 の実態である super1.obj と SUPER1 のリポジトリ情報が生成される。（-CG オプションは、リポジトリ情報を生成するオプション）

同様に、クラス CLASS1 は、クラス定義内のリポジトリ段落に継承元クラスである SUPER1 を指定した上で、下記のように翻訳する。

```
cob -c -CG -R¥super_def class1.cob
```

ここで、クラス CLASS1 の継承元であるクラス SUPER1 のリポジトリ情報は、¥super\_def ディレクトリ配下に存在することを想定している。(-R オプションは、外部リポジトリが存在するディレクトリを指定するオプション)

これにより、カレントディレクトリ(ここでは ¥class\_def とする)にクラス CLASS1 の実態である class1.obj と CLASS1 のリポジトリ情報が生成される。

クラス CLASS1 には、クラス定義中にリポジトリ段落に SUPER1 が指定されているため、コンパイラは、外部リポジトリから SUPER1 に関するリポジトリ情報を参照する。その結果、コンパイラは、クラス SUPER1 のメソッドの引数や返却値のデータ型を知ることができるので、問題なくクラス CLASS1 を翻訳できる。

プログラム PROG1 のリポジトリ段落には、直接参照しているクラス CLASS1 のみを指定する。クラス CLASS1 の継承元である SUPER1 については、プログラム PROG1 で直接参照していないかぎり、リポジトリ段落には記述する必要はない。

プログラム PROG1 の実行形式を生成するためには、下記のように翻訳コマンドを指定する。

```
cob -M -R¥class_def PROG1.cob ¥super_def¥super1.obj ¥class_def¥class1.obj
```

ここで、クラス CLASS1 のリポジトリ情報が、¥class\_def ディレクトリ配下に存在することを想定している(-R オプション)。また、継承元クラスのリポジトリ参照ディレクトリを指定する必要はないが、オブジェクトファイル(super1.obj)の指定は必須となる。

このように、クラスを参照しているクラスやプログラムの翻訳を行う場合には、リポジトリ段落に参照しているクラスの指定と、-R オプションによる外部リポジトリの指定、および参照しているクラスが直接的または間接的に参照しているすべてのクラスのオブジェクトファイルを指定する必要がある。

なお、翻訳指定を簡易化するために、オブジェクトファイルは、まとめて1つのアーカイブファイル、共有ライブラリ化しておくのが望ましい。たとえば、super1.obj および class1.obj をまとめて class.a (または class.dll)とした場合には、次のように指定することで翻訳ができる。

```
cob -M -R¥class_def PROG1.cob -lclass
```

## 9.6 オブジェクト指向機能利用時のメモリ容量

【Enterprise for Linux】本節の以降の説明文では、オブジェクトファイル名の拡張子「.obj」を「.o」、ディレクトリのパス区切り文字「¥」を「/」に読み替えること。

クラスを生成したときにどのようにメモリが増えるかは、システム設計をする上で非常に重要である。ここでは、本コンパイラがどのようにメモリの確保をしているのか、その概要を説明する。

なお、説明の便宜上、クラス **c** (ファイル名を **c.cob** とする) の中に定義されるファクトリ、インスタンスのそれぞれのデータ部、手続き部の大きさを次のように表現することとする。

- c.f.wss** : ファクトリ内の作業場所節のデータの大きさ
- c.f.lss(n)** : ファクトリ内の **n** 番目のメソッドの局所記憶節データの大きさ
- c.f.prc(n)** : ファクトリ内の **n** 番目のメソッドの手続き部の大きさ
- c.i.wss** : インスタンス内の作業場所節のデータの大きさ
- c.i.lss(n)** : インスタンス内の **n** 番目のメソッドの局所記憶節データの大きさ
- c.i.prc(n)** : インスタンス内の **n** 番目のメソッドの手続き部の大きさ

また、クラスを参照しているプログラム **p** (ファイル名を **p.cob** とする) のデータ部、手続き部の大きさを次のように表現する。

- p.wss** : 作業場所節のデータの大きさ
- p.lss** : 局所記憶節のデータの大きさ
- p.prc** : 手続き部の大きさ

さらに、下記翻訳コマンドにより、**p.exe** という実行形式が生成されているものとする。

```
cob -op.exe -M p.cob c.obj
```

**p.exe** は、プログラム **p** のデータおよび手続き、およびクラス **c** のファクトリおよびインスタンスのデータやメソッドなどの手続きから構成されている。**p.exe** を実行したとき、データ部や手続き部のデータは次のようにメモリ空間に割り付けられる。

### (1) p.exe 起動時

p.exe が最初に起動されたときにメモリ上に確保されるデータは、プログラム p のデータ、およびクラス c のファクトリオブジェクト分のデータのみであり、インスタンスオブジェクトのデータは確保されない。

また、手続き部（プログラムの手続き部およびファクトリオブジェクトのメソッド、インスタンスオブジェクトのメソッド）は、起動時にすべてメモリ上に確保される。

データ部のメモリの大きさ:  $p.wss + p.lss + c.f.wss$

手続き部のメモリの大きさ:  $p.prc + \sum c.f.prc(i) + \sum c.i.prc(i)$

※ここで  $\sum$  は総和を示す

p.exe 終了時、確保されているメモリはすべて解放される。

## (2) インスタンス生成時

インスタンスを1つ生成すると、インスタンスオブジェクトの作業場所節の大きさだけメモリが増加する。複数生成すると、その分メモリが増加する。

データ部の増加メモリ大きさ:  $c.i.wss \times \text{インスタンス生成数}$

手続き部の増加メモリ大きさ: 0

インスタンスが削除されると、メモリ容量はその分減少する。

## (3) ファクトリオブジェクトのメソッド実行時

実行したファクトリオブジェクトのメソッドの局所記憶域の大きさの分、メモリが増加する。メソッド実行終了時には、その分のメモリは解放される。

データ部の増加メモリ大きさ:  $c.f.lss$

手続き部の増加メモリ大きさ: 0

## (4) インスタンスオブジェクトのメソッド実行時

実行したインスタンスオブジェクトのメソッド n の局所記憶域の大きさの分、メモリが増加する。メソッド実行終了時には、その分のメモリは解放される。

データ部の増加メモリ大きさ:  $c.w.lss(n)$

手続き部の増加メモリ大きさ: 0

## 9.7 注意事項

### (1) メソッド名の命名規則について

**CSRelease** メソッド, **CSReleaseSelf** メソッド, **CSReleaseInThread** メソッド, **CSReleaseAll** メソッドは, 本コンパイラ独自追加のメソッドである。

したがって, 利用者は同綴り (大文字小文字の区別なし) のメソッドを個別に定義することはできず, 定義した場合には翻訳エラーとなる。

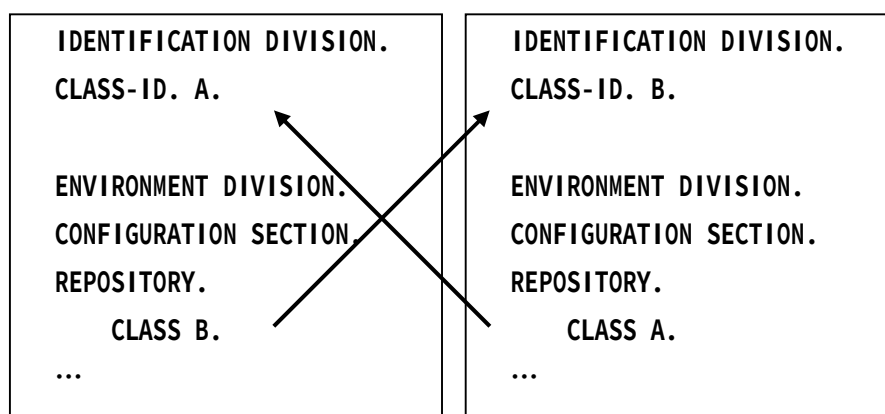
なお, 本コンパイラでは今後も独自追加メソッドを「CS」から始まる名前で追加する可能性がある。したがって, 利用者は「CS」から始まるメソッドは定義しないようにするのが望ましい。

### (2) 非限定オブジェクト参照について

本コンパイラでは, 非限定オブジェクト参照の定義は可能であるが, 利用できる範囲を制限している。したがって, 特別な理由がないかぎり, 非限定オブジェクト参照は利用しないのが望ましい。

### (3) クラスの相互参照について

本コンパイラでは, 次のようなクラス相互参照 (クラス A はクラス B のオブジェクト参照をオブジェクトデータとして持ち, クラス B はクラス A のオブジェクト参照をオブジェクトデータとして持つなど) は記述できない。



相互参照を記述した場合, アプリケーション作成時に使用する C/C++コンパイラにおいて翻訳エラーとなる。

### (4) クラスを含む DLL の生成について

[Enterprise for Windows] [Standard] 本コンパイラでは, クラスを含む DLL を生成する場合,

そのクラスの祖先クラスについても同じ DLL に含めるか、別途 DLL を生成するとともに、インポートライブラリを生成する必要がある。(既に DLL 化されている BASE クラスは除く。)

クラスを含む DLL を利用したアプリケーションを生成する際に、下位クラスしか DLL に含まれていない場合、リンクエラーが発生する。

**[Enterprise for Linux]** 本コンパイラでは、クラスを含む共有ライブラリを生成する場合、そのクラスの祖先クラスについても同じ共有ライブラリに含めるか、別途共有ライブラリを生成する。(既に共有ライブラリ化されている BASE クラスは除く。)

クラスを含む共有ライブラリを利用したアプリケーションを生成する際に、下位クラスしか共有ライブラリに含まれていない場合、リンクエラーが発生する。

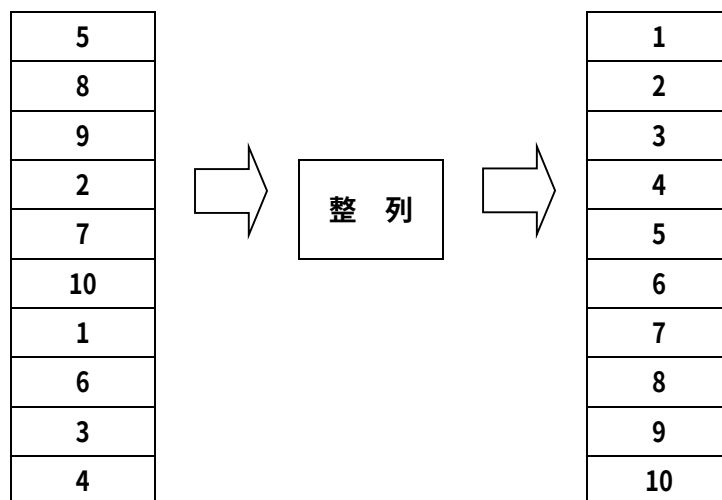


## 第10章 整列併合機能

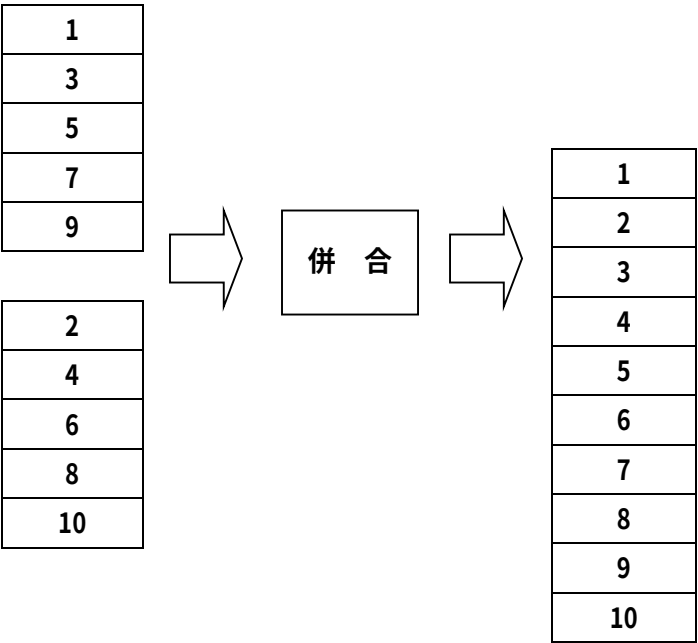
### 10.1 概 説

整列(sorting)機能とは、ファイル中の入力データを指定した項目によって、順序よく並べ替えることである。たとえば、売上傳票の並べ方は、日付順、得意先別、売上の多い順などいろいろな順序が考えられるが、あるときは品物単位の売上高に関する報告書が必要となるかもしれない。この場合、入力データとしての売上傳票が品物別に整列されているか否かによって、処理効率が大きいに左右される。このように、ファイルの中での入力データは、単にある一定の順序で収まっているだけで、他の順序にするためには並べ替えが必要である。この並べ替えの操作を整列操作という。

整列機能では、先に述べたような日付や得意先、品物単位などの整列の順序を決める項目を整列用キーといい、ファイル内のレコードを昇順または降順に並べ替えることができる。さらに、整列操作においてレコードの追加、削除、変更、その他の修正処理を入力手続きまたは出力手続き内で行うことができる。

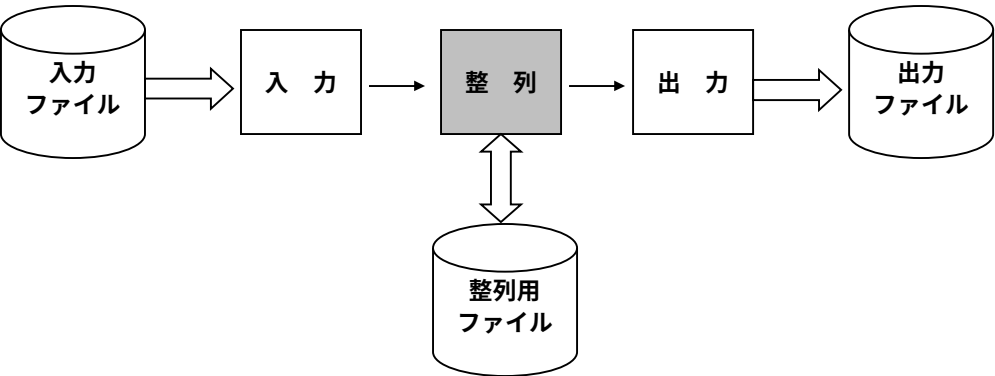


また、併合機能では、一定の規則で整列されたいくつかのファイルを 1 個のファイルとしてまとめることができる。さらに併合操作においてレコードの追加、削除、変更、その他の修正処理を出力手続き内で行うことができる。

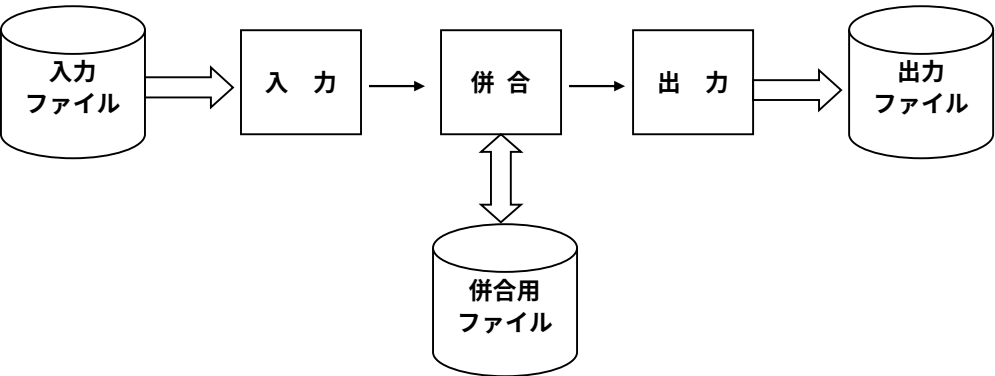


## 10.2 整列併合用ファイルの設定

ファイルの整列を目的とするプログラムの整列操作と関連ファイルに注目して流れ図を示すと次のようになる。



同様にファイルの併合を目的とするプログラムについて流れ図を示すと次のようになる。



整列用ファイルとは、整列操作時に作業用として一時的に使用するファイルである。併合用ファイルとは、併合操作時に必要なキー項目の位置や表現形式などを指定する目的だけで記述する仮想ファイルである。したがって、整列および併合用ファイルの扱いは他の通常のファイルと次の点において異なる。

- SELECT 記述項では、ASSIGN 句以外の句を記述してはならない。
- ファイル記述項は、レベル指示語 SD で始める。
- 整列併合用ファイル記述項で指定できるのは、RECORD 句と DATA RECORDS 句だけである。
- OPEN 文で開いたり、CLOSE 文で閉じてはならない。
- 整列用ファイルへの書き込みは RELEASE 文で行う。
- 整列および併合用ファイルの読み込みは RETURN 文で行う。

前記の相違点から推察できるように、整列および併合用ファイルを通常のファイル同様にファイル管理段落やファイル節で記述するのは、単に構文上の必要性からにすぎない。より重要なことは、整列および併合用ファイルのレコード記述項に SORT 文または MERGE 文で指定するキー項目が、すべて含まれていなければならないことである。また、これらのキー項目はすべてのレコードにおいて表現形式を含めて同じ大きさであり、さらに相対的に同じ位置になければならない。

次にファイル管理段落およびファイル節の記述例を示す。

```
FILE-CONTROL.
 SELECT SORT-F ASSIGN TO "SORTWK-DFS".
 ;
DATA DIVISION.
FILE SECTION.
SD SORT-F
 DATA RECORDS ARE REC1 REC2.
01 REC1.
 ;
01 REC2.
 ;
```

### 10.3 整列併合機能

COBOL で提供する整列機能は、簡単に記述すると次のようになる。

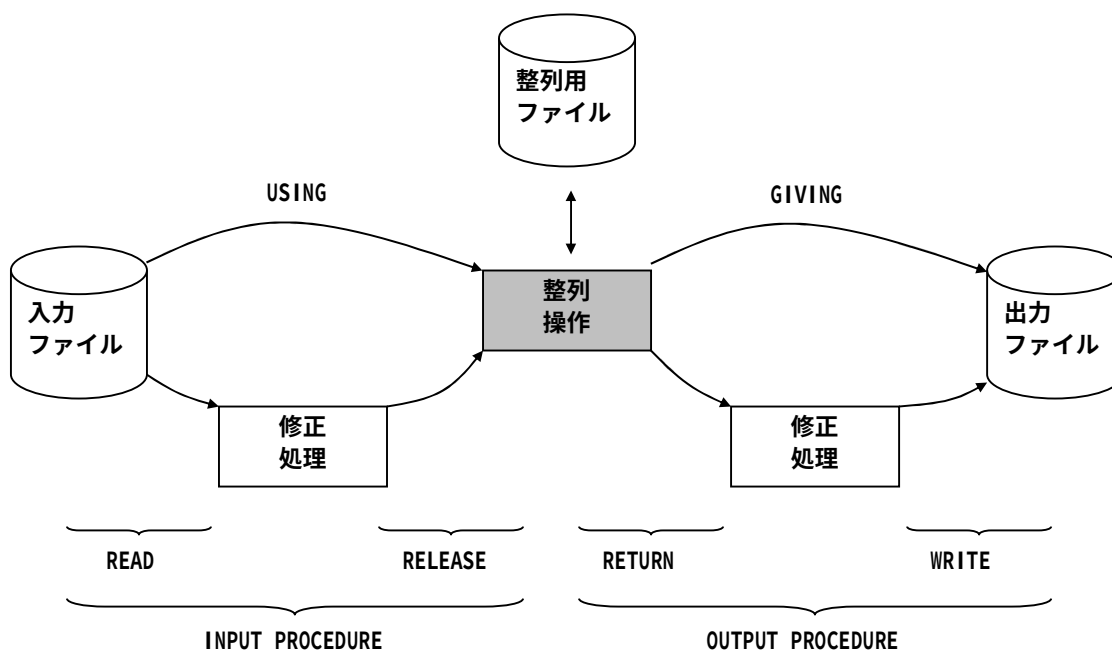
- (1) 整列すべきデータの入っているファイルからまたはそのプログラム内でデータを作り出し、整列用ファイルにレコードを引き渡す。この間に必要があればレコードの編集を行う。
- (2) 次に、整列用ファイルに渡されたレコードを指定された整列用キーに従って昇順または降順に並べ替える。
- (3) 整列されたレコード順に受け取り、必要があれば編集して出力用ファイルに出力する。

COBOL での併合機能は、簡単に記述すると次のようになる。

- (1) 併合するデータの入ったファイルから、併合キーによって昇順または降順に、併合用ファイルにレコードを引き渡す。
- (2) 併合されたレコード順に受け取り、必要があれば編集して出力用ファイルに出力する。

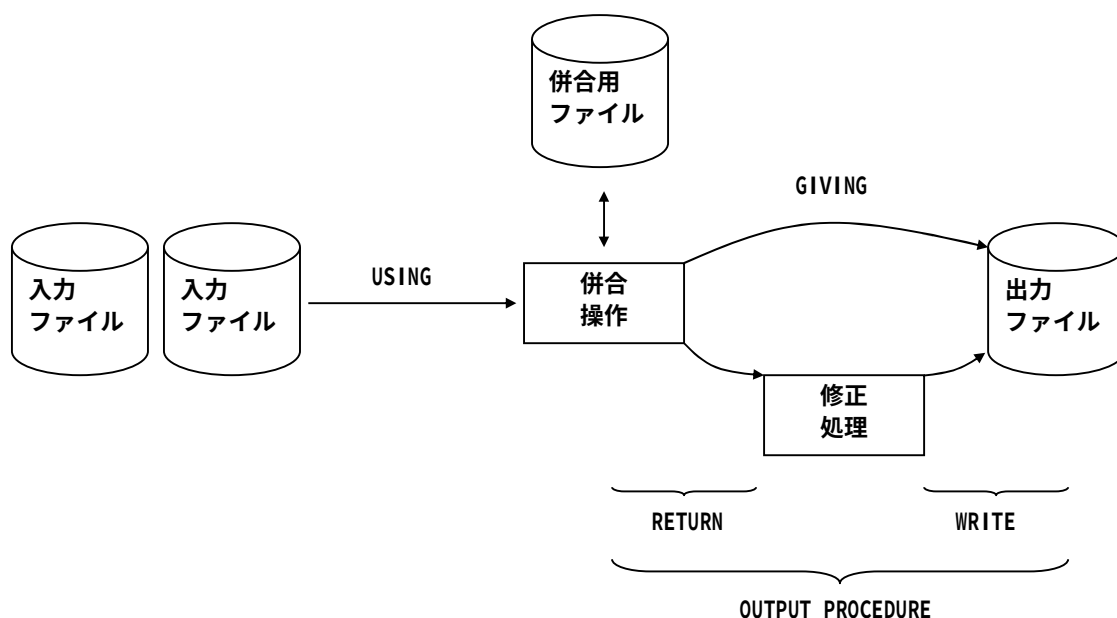
整列用ファイルと呼び出す文として SORT, RELEASE, RETURN 文がある。SORT 文と SORT 文の句および RELEASE 文, RETURN 文の相互関係を図示する。

図 10-1 整列用文の相互関係



同様に併合を目的とする場合、MERGE 文と MERGE 文の句および RETURN 文の相互関係を図示する。

図 10-2 併合用文の相互関係



整列併合用ファイルは通常のファイルと異なり OPEN 文で開いたり、CLOSE 文で閉じたりしてはならない。

### 10.3.1 入力ファイルと出力ファイル

入力ファイル とは、次のファイルをいう。

- (1) SORT 文または MERGE 文の USING 句で指定されたファイル。
- (2) SORT 文の INPUT PROCEDURE 句で指定された入力手続き中で読まれるファイル。ただし、入力手続き中では必ずしもファイルからレコードが読み込まれるとはかぎらず、入力ファイルがない場合もありうる。

出力ファイル とは、次のファイルをいう。

- (1) SORT 文または MERGE 文の GIVING 句で指定されたファイル。
- (2) SORT 文または MERGE 文の OUTPUT PROCEDURE 句で指定された出力手続き中で書かれるファイル。ただし、出力手続き中では必ずしもファイルにレコードが書かれるとはかぎらず、出力ファ

イルがない場合もありうる。

整列併合処理の入力ファイルや出力ファイルとして用いられるファイルの編成または媒体などは任意でよい。ただし、`USING` 句や `GIVING` 句で指定するファイルの呼び出し法は順呼び出しでなければならない。

### 10.3.2 レコードの引き渡しとレコードの受け取り

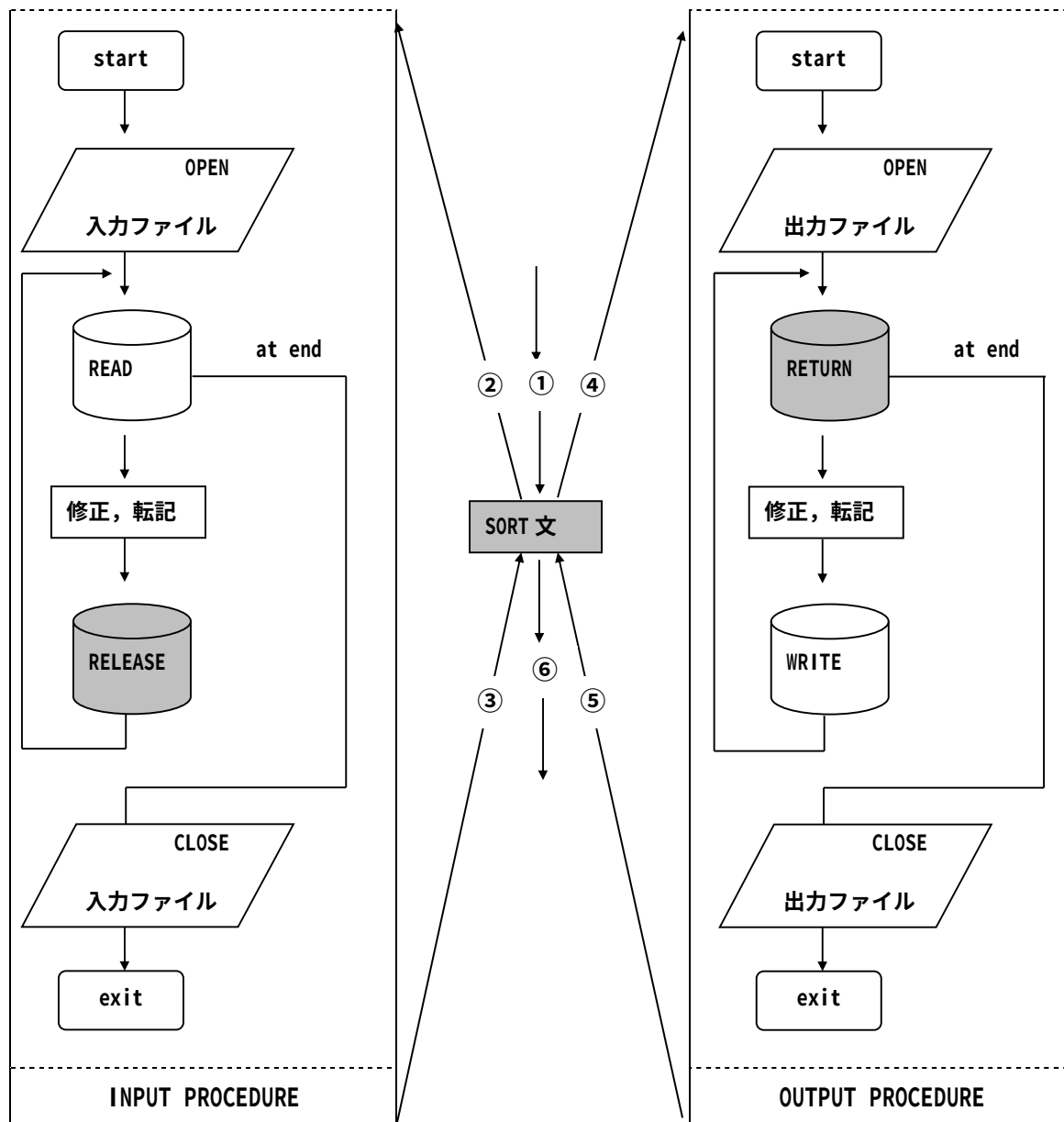
COBOL の実行用プログラムから整列プログラムへのレコード引き渡しは、`RELEASE` 文によって行う。また、整列併合プログラムからのレコードの受け取りは `RETURN` 文によって行う。これらの文は、`SORT` 文で `INPUT PROCEDURE` 句や `SORT` 文および `MERGE` 文で `OUTPUT PROCEDURE` 句を指定する場合には、おのおのに対応する入力手続きや出力手続きの中でのみ実行することができる。

`SORT` 文および `MERGE` 文で `USING` 句や `GIVING` 句を指定する場合には、コンパイラが入力手続きや出力手続きを生成する。

入力手続きや出力手続き(コンパイラが生成する入力手続き、出力手続きも含めて)は、`SORT` 文および `MERGE` 文の中から `PERFORM` 文によって制御を渡される。入力手続きや出力手続きの最後には、`PERFORM` 文の復帰機構が生成される。

次に入力手続きの基本構造と `SORT` 文の対応を図示する。

図 10-3 SORT 文と入力, 出力手続の関係



なお、整列操作の入力処理において整列すべきレコードが 1 件もなかった場合、出力処理は次のいずれかとなる。

- (1) OUTPUT PROCEDURE 句指定のときは、出力手続き中の最初の RETURN 文実行時に AT END 条件となる。
- (2) GIVING 句指定のときは、出力ファイルに対して OPEN(OUTPUT)および CLOSE 文のみ実行される。

ここで、整列操作の入力処理において整列すべきレコードが 1 件もなかった場合とは次のいずれかの場合である。

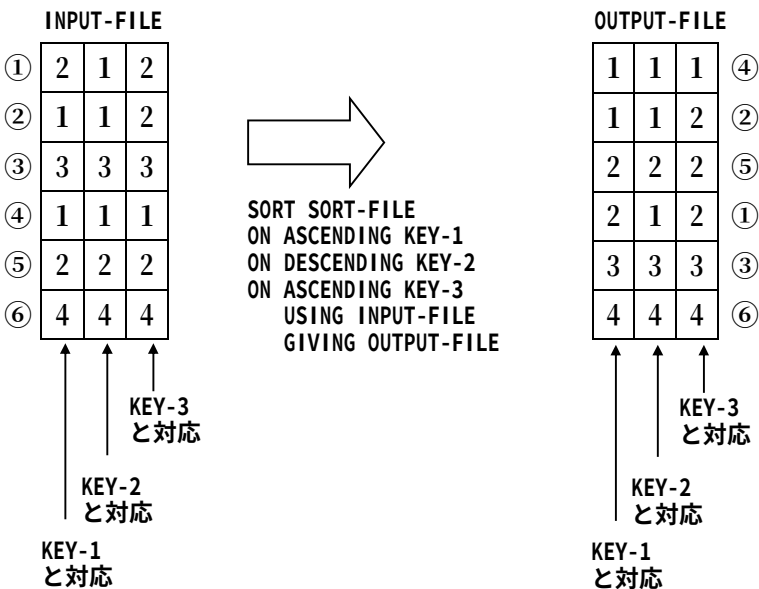
- (1) INPUT PROCEDURE 句指定のときは、入力手続き中で RELEASE 文が 1 回も実行されなかった。
- (2) USING 句指定のときは、入力ファイル中に 1 つのレコードも含まれていなかった。

10.3.3 キー項目の宣言と評価

ファイル内のデータを一定順序になるよう並べ替える場合、レコードのどの情報をもとにし、昇順 (ascending order) または降順 (descending order) で行うか指定する。この情報を整列用キーといい、プログラムでは SORT 文および MERGE 文で、次のように指定する。なお、キー項目の位置や大きさは、整列併合用ファイル記述項で定義する。

ON { ASCENDING  
DESCENDING } KEY キー項目

下の例では、KEY-2 で降順、KEY-1、KEY-3 で昇順に整列することとする。



キーの優先順位は、SORT 文または MERGE 文で記述した順序になる。すなわち、KEY-1 で昇順に並べ KEY-1 の値が同じレコードの場合は、KEY-2 で降順に並べている。さらに同じものがある場合



は、KEY-3 で昇順に並べ替える。それでも同じものがある場合は、引き渡される順序とは無関係に出力される。

整列キーおよび併合キーは、英数字項目や数字項目などによって、次のように整列優先順位が決められる。

- 英数字項目の場合

整列キーおよび併合キーの大小順序は、次の優先順位で決められ、この大小順序に従って整列する。

- (1) SORT 文または MERGE 文で、COLLATING SEQUENCE(文字照合順序)句で指定された順
- (2) PROGRAM COLLATING SEQUENCE 句で指定した順
- (3) プログラムの固有文字集合の大小順序

- 数字項目の場合

常に、数値としての大小順序に従って整列する。

- 日本語項目、日本語編集項目の場合

シフト JIS コード体系に従って整列する。

[Enterprise] ただし、-CU オプションを指定した場合は、UTF-16LE コード体系に従って整列する。

[Standard] [Enterprise for Windows] また、-CF オプションを指定した場合は、JIPS(J)コード体系に従って整列する。

- 同じキーデータが 2 つ以上ある場合

整列キーの中に同一のキーデータが 2 つ以上ある場合には、大小順序は不定となる。同一キーデータの順序を入力された順に出力するには、SORT 文または MERGE 文で DUPLICATES 句を指定する。

## 10.4 整列併合処理

整列機能は、整列の作業用に記述されたファイル(整列用ファイル)のレコードを並べ替える(整列)機能である。これは SORT 文の中に、整列用ファイルへのデータの引き渡し方、整列用キー、整列されたレコードの整列用ファイルからの受け取り方を指定することによって、整列処理が実行される。

なお、ここでは整列機能をどのように利用するかを理解しやすくするために、簡単な例をあげて説明する。

例「ある会社で入社試験を実施し、得点順に並べ替えた情報を必要としている場合」

入力ファイル形式:磁気ディスク, 順ファイル

レコードの大きさ:80桁

レコード内容

| 1    | 11 | 12  | 31 | 32  | 36 | 37    | 80 |
|------|----|-----|----|-----|----|-------|----|
| 受験番号 |    | 氏 名 |    | 得 点 |    | その他情報 |    |

出力ファイル形式:磁気ディスク, 順ファイル

レコードの内容およびレコードの大きさは入力ファイルと同じ

プログラム例

```

1 000010 IDENTIFICATION DIVISION.
2 000020 PROGRAM-ID. SORT01.
3 000030 ENVIRONMENT DIVISION.
4 000040 CONFIGURATION SECTION.
5 000050 SOURCE-COMPUTER. PC9800.
6 000060 OBJECT-COMPUTER. PC9800.
7 000070 INPUT-OUTPUT SECTION.
8 000080 FILE-CONTROL.
9 000090 SELECT INFILE ASSIGN TO "INF-UFS".
10 000100 SELECT SORTWORK ASSIGN TO "SORTWORK-UFS".
11 000110 SELECT OUTFILE ASSIGN TO "OUTF-UFS".
12 000120 DATA DIVISION.
13 000130 FILE SECTION.
14 000140 FD INFILE LABEL RECORD IS STANDARD.
15 000150*
16 000160 01 INREC PICTURE X(80).
17 000170*
18 000180 SD SORTWORK.
19 000190 01 SRTREC.
20 000200 02 FILLER PICTURE X(31).
21 000210 02 TOKUTEN PICTURE 9(5).
22 000220 02 FILLER PICTURE X(44).
23 000230*

```

```

24 000240 FD OUTFILE LABEL RECORD IS STANDARD.
25 000250*
26 000260 01 OUTREC PICTURE X(80).
27 000270*
28 000280 PROCEDURE DIVISION.
29 000290 SORTEXAMPLE.
30 000300 SORT SORTWORK
31 000310 ON DESCENDING KEY TOKUTEN
32 000320 USING INFILE
33 000330 GIVING OUTFILE.
34 000340 STOP RUN.

```

この例では、入力ファイル(INFILE)から単純に整列用ファイル(SORTWORK)にレコードを順に渡し、指定されたキー(TOKUTEN)に従って大きいものから順にレコードを並べ替えている。そして、出力ファイル(OUTFILE)として指定された順ファイルに出力している。

ここでは整列用ファイルと入力、出力ファイルのレコードの大きさは同じで、レコード内の各項目の並べ替えや編集は行わず、整列用キーも 1 個である。しかし、通常の整列では、入力ファイルから整列用ファイルにレコードを引き渡すときや整列用ファイルからデータを受け取った後に編集することが多く、また整列用キーも複数個あることが常である。

### 10.4.1 SORT 文

**SORT 文**で指定するキーは、整列用ファイルのレコードに属する項目でなければならない。

キーは、優先度の高い順に左から右へ並べる。昇順に整列するのであれば **ASCENDING 句**、降順に整列するのであれば **DESCENDING 句**をそれぞれのキーに指定する。

**USING 句**では、入力ファイルを指定する。

**USING 句**のある **SORT 文**を実行するときに、入力ファイルは開かれていてはならない。

コンパイラは入力ファイルを開き、全部の入力レコードを整列用ファイルへ引き渡し、入力ファイルを閉じる文を実行用プログラムに含ませる。整列用ファイルは入力ファイルより等しいか大きくなければならない。

**GIVING 句**では、出力ファイルを指定する。

**GIVING 句**のある **SORT 文**を実行するときに、出力ファイルは開かれていてはならない。

コンパイラは出力ファイルを開き、全部の整列済みレコードを整列用ファイルから出力ファイルへ書き移し、出力ファイルを閉じる文を実行用プログラムに含ませる。出力ファイルは整列用ファイルより等しいか大きくなければならない。

次に、USING 句と GIVING 句指定のある SORT 文を含むプログラム例を示す。

```
FILE-CONTROL.
 SELECT INFILE ASSIGN TO "INF-UFS".
 SELECT OUTFILE ASSIGN TO "OUTF-UFS".
 SELECT SORTWORK ASSIGN TO "SORTWORK-UFS".
 ;
FILE SECTION.
FD INFILE ...
01 INREC.
 ;
FD OUTFILE ...
01 OUTREC.
 ;
SD SORTWORK ...
01 SRTREC.
 ;
PROCEDURE DIVISION.
SORT-CALL.
 SORT SORTWORK ON ...
 USING INFILE GIVING OUTFILE.
SORT-FINAL.
 STOP RUN.
```

コンパイラは、USING 句に対しては入力手続きを GIVING 句に対しては出力手続きをそれぞれ生成する。したがって、上記の SORT 文は結果的には次のような入力手続きと出力手続きをもつ SORT 文と同じ機能をもつことになる。

```
PROCEDURE DIVISION.
DUMMY SECTION.
SORT-CALL.
 SORT SORTWORK ON ...
 INPUT PROCEDURE SEC-X.
 OUTPUT PROCEDURE SEC-Y.
SORT-FINAL.
 STOP RUN.
SEC-X SECTION.
X-1. OPEN INPUT INFILE.
X-2. READ INFILE AT END GO TO X-3.
 MOVE INREC TO SRTREC.
 RELEASE SRTREC.
 GO TO X-2.
X-3. CLOSE INFILE.
SEC-Y SECTION.
Y-1. OPEN OUTPUT OUTFILE.
Y-2. RETURN SORTWORK AT END GO TO Y-3.
 MOVE SRTREC TO OUTREC.
 WRITE OUTREC.
 GO TO Y-2.
Y-3. CLOSE OUTFILE.
```

#### 10.4.2 RELEASE 文

RELEASE 文は、入力手続き内で使わなければならない。WRITE 文に相当し、整列用ファイルに対する書き込み文である。RELEASE 文実行後は、整列用ファイルのレコード処理領域の内容は不定である。

### 10.4.3 RETURN 文

RETURN 文は、出力手続き内で使わなければならない。READ 文に相当し、整列用または併合用ファイルからレコードを読み込む文である。AT END 句の無条件文に制御が移ったときには、レコード処理領域の内容は不定である。

### 10.4.4 MERGE 文

MERGE 文で指定するキーは、併合用ファイルのレコードに属する項目でなければならない。

キーは、優先度の高い順に左から右へ並べる。昇順に整列するのであれば ASCENDING 句、降順に整列するのであれば DESCENDING 句をそれぞれのキーに指定する。

USING 句では、併合すべき入力ファイルを 2 個以上指定する。入力ファイルは、64 個まで指定することができる。MERGE 文を実行するとき、入力ファイルは開かれていてはならない。

コンパイラは全部の入力ファイルを開き、全部のレコードを自動的に併合用ファイルへ引き渡し、入力ファイルを閉じる文を実行用プログラムに含ませる。入力ファイルと併合用ファイルとのレコード記述は等しくなければならない。

GIVING 句では、出力ファイルを指定する。

GIVING 句のある MERGE 文を実行するとき、出力ファイルは開かれていてはならない。

コンパイラは出力ファイルを開き、全部の併合済みのレコードを併合用ファイルから自動的に出力ファイルへ引き渡し、出力ファイルを閉じる文を実行可能プログラムに含ませる。出力ファイルと併合用ファイルのレコード記述は等しくなければならない。

次に、USING 句と GIVING 句指定のある MERGE 文を含むプログラム例を示す。

```
FILE-CONTROL.
 SELECT INFILE1 ASSIGN TO "INF1-UFS".
 SELECT INFILE2 ASSIGN TO "INF2-UFS".
 SELECT MERGEWORK ASSIGN TO "MERGEWORK-UFS".
 SELECT OUTFILE ASSIGN TO "OUTF-UFS".
 ;
FILE SECTION.
FD INFILE1 ...
01 INREC1.
 ;
FD INFILE2 ...
01 INREC2.
 ;
SD MERGEWORK ...
01 MERGEREC.
 ;
FD OUTFILE ...
01 OUTREC.
 ;
PROCEDURE DIVISION.
MERGE-CALL.
 MERGE MERGEWORK ON ...
 USING INFILE1 INFILE2
 GIVING OUTFILE.
SORT-FINAL.
 STOP RUN.
```

### 10.4.5 入力手続き

入力ファイルから整列用ファイルにレコードを引き渡すときに編集する場合(レコード内のある項目を変更する, または特定のレコードを整列用ファイルに渡さない場合など), またはプログラム内でデータを作り出す場合には入力手続きを指定する。プログラムでは SORT 文の INPUT PROCEDURE 句で次のように指定する。

$$\text{INPUT PROCEDURE IS 手続き名 1} \left[ \left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{手続き名 2} \right]$$

実行時の制御の流れは, SORT 文に制御が移ってくれば整列が行われる前に手続き名 1 に制御が移り, それから手続き名 1 の最後の文を実行後(THROUGH 句指定なし)または手続き名 2 の最後の文を実行後(THROUGH 句指定あり), SORT 文に制御が戻り, 整列操作が開始される。

整列用ファイルへのレコードの引き渡しは RELEASE 文で行う。

USING 句指定のときは, 入力ファイルの OPEN, READ および CLOSE 処理は自動的に行われるが, 入力手続き指定のときは入力ファイルの OPEN, CLOSE などをプログラムで記述する必要がある。

### 10.4.6 出力手続き

整列用ファイルから整列されたレコードを受け取り, または, 併合用ファイルから併合されたレコードを受け取り, そのまま出力ファイルに出力しないときには出力手続きを指定する。

プログラムでは, SORT 文または MERGE 文の OUTPUT PROCEDURE 句で, 次のように指定する。

$$\text{OUTPUT PROCEDURE IS 手続き名 1} \left[ \left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{手続き名 2} \right]$$

実行時の制御の流れは, 整列併合終了後, 手続き名 1 に制御が移り, それから手続き名 1 の最後の文を実行後(THROUGH 句指定なし)または手続き名 2 の最後の文を実行後(THROUGH 句指定あり), SORT 文または MERGE 文の次の文に制御が移る。整列用ファイルから整列されたレコードを受け取るとき, または併合用ファイルから併合されたレコードを受け取るときは, RETURN 文によって行う。

また GIVING 句指定のときは出力ファイルの OPEN, WRITE および CLOSE 処理は自動的に行われるが, 出力手続きを指定したときは出力ファイルの OPEN, WRITE, CLOSE などはプログラムで記述する必要がある。

次に, 入力手続きまたは出力手続きを使用したプログラムの例を次に示す。



例 「ある会社で入社試験を実施し、得点順に並べ替えた情報を必要としている場合」

前記の例を次のように変更する。

入力ファイル形式

- 磁気ディスク、順ファイル
- レコードの大きさ: 80 桁
- レコード内容

|      |     |     |    |    |       |
|------|-----|-----|----|----|-------|
| 1    | 11  | 12  | 31 |    |       |
| 受験番号 | 氏 名 | 得 点 |    |    | その他情報 |
|      |     | 数学  | 英語 | 国語 |       |

出力ファイル形式

- ラインプリンタ
- レコードの大きさ: 120 桁
- 出力内容

|          |                                         |
|----------|-----------------------------------------|
| ページ見出し 1 | △△△△△△△入 社 試 験 成 績 順 位 表                |
| ページ見出し 2 | △ 番号 △△△△受験番号△△△ ~ 氏 名 ~ △数学△英語△国語△△合計  |
| 明細       | △△×××△△×××××××××××××××××××△△×△△×△△×△△×× |
| 50 桁／ページ | :                                       |
|          | :                                       |
|          | :                                       |

処理概略

入力ファイルから得られる情報によって、総得点の高いものから低いものの順に並べ換え、出力リストに印字する。総得点が同じときは数学の点の高いものを先に出力する。

## プログラム例

```

1 000010 IDENTIFICATION DIVISION.
2 000020 PROGRAM-ID. SORT02.
3 000030*
4 000040* 整 列 機 能 の 例 題
5 000050*
6 000060 ENVIRONMENT DIVISION.
7 000070 CONFIGURATION SECTION.
8 000080 SOURCE-COMPUTER. PC9800.
9 000090 OBJECT-COMPUTER. PC9800.
10 000100 INPUT-OUTPUT SECTION.
11 000110 FILE-CONTROL.
12 000120*
13 000130 SELECT INFILE ASSIGN TO "SEISEKIFILE-UFS".
14 000140 SELECT SORTWORK ASSIGN TO "SORTWK-UFS".
15 000150 SELECT OUTFILE ASSIGN TO "LPFILE-SYSOUT".
16 000160*
17 000170 DATA DIVISION.
18 000180*
19 000190 FILE SECTION.
20 000200*
21 000210* INPUT-FILEのレコードレイアウト
22 000220*
23 000230 FD INFILE LABEL RECORD STANDARD.
24 000240 01 INRECORD.
25 000250 03 JUKEN-NO PIC 9(11).
26 000260 03 SHIMEI PIC X(20).
27 000270 03 TOKUTEN.
28 000280 05 SUGAKU PIC 9(03).
29 000290 05 EIGO PIC 9(03).
30 000300 05 KOKUGO PIC 9(03).
31 000310 03 FILLER PIC X(40).
32 000320*
33 000330* SORT-WORK-FILE の ソート記述項

```

```

34 000340*
35 000350 SD SORTWORK.
36 000360 01 SORTRECORD.
37 000370 03 S-JUKEN-NO PIC 9(11).
38 000380 03 S-SHIMEI PIC X(20).
39 000390 03 S-TOKUTEN.
40 000400 05 S-SUGAKU PIC 9(03).
41 000410 05 S-EIGO PIC 9(03).
42 000420 05 S-KOKUGO PIC 9(03).
43 000440 05 S-TOTAL PIC 9(04).
44 000440*
45 000450* プリンタの出力レコードバッファ
46 000460*
47 000470 FD OUTFILE LABEL RECORD OMITTED.
48 000480 01 OUTRECORD.
49 000490 03 PRINTDATA PIC X(130).
50 000500*
51 000510*
52 000520 WORKING-STORAGE SECTION.
53 000530*
54 000540* プリンタ出力用の明細行
55 000550*
56 000560 01 EDITRECORD.
57 000570 03 FILLER PIC XX VALUE SPACE.
58 000580 03 P-NO PIC ZZ9.
59 000590 03 FILLER PIC XX VALUE SPACE.
60 000600 03 P-JUKEN-NO PIC 9(11).
61 000610 03 FILLER PIC XX.
62 000620 03 P-SHIMEI PIC X(20).
63 000630 03 FILLER PIC XX VALUE SPACE.
64 000640 03 P-SUGAKU PIC ZZ9.
65 000650 03 FILLER PIC XX VALUE SPACE.
66 000660 03 P-EIGO PIC ZZ9.
67 000670 03 FILLER PIC XX VALUE SPACE.
68 000680 03 P-KOKUGO PIC ZZ9.
69 000690 03 FILLER PIC XX VALUE SPACE.

```

```

70 000700 03 P-TOTAL PIC ZZZ9.
71 000710 03 FILLER PIC X(59) VALUE SPACE.
72 000720*
73 000730* 帳票見出し1
74 000740*
75 000750 01 HEADING-1.
76 000760 03 FILLER PIC X(7) VALUE SPACE.
77 000770 03 FILLER PIC N(17) VALUE
78 000780 NC"入 社 試 験 成 績 順 位 表".
79 000790 03 FILLER PIC X(79) VALUE SPACE.
80 000800*
81 000810* 帳票見出し2
82 000820*
83 000830 01 HEADING-2.
84 000840 03 FILLER PIC X(1) VALUE SPACE.
85 000850 03 FILLER PIC N(2) VALUE NC"番号".
86 000860 03 FILLER PIC X(4) VALUE SPACE.
87 000870 03 FILLER PIC N(4) VALUE NC"受験番号".
88 000880 03 FILLER PIC X(3) VALUE SPACE.
89 000890 03 FILLER PIC N(10) VALUE NC"氏名".
90 000900 03 FILLER PIC X VALUE SPACE.
91 000910 03 FILLER PIC N(2) VALUE NC"数学".
92 000920 03 FILLER PIC X VALUE SPACE.
93 000930 03 FILLER PIC N(2) VALUE NC"英語".
94 000940 03 FILLER PIC X VALUE SPACE.
95 000950 03 FILLER PIC N(2) VALUE NC"国語".
96 000960 03 FILLER PIC X(2) VALUE SPACE.
97 000970 03 FILLER PIC N(2) VALUE NC"合計".
98 000980 03 FILLER PIC X(59) VALUE SPACE.
99 000990*
100 001000* PROGRAM WORK AREA
101 001010*
102 001020 77 W-NO PIC 99.
103 001030 77 W-LINE-NO PIC 99.
104 001040*
105 001050*

```

```

106 001060 PROCEDURE DIVISION.
107 001070*
108 001080*****LEVEL 0*****
109 001090*
110 001010 SORT-MAIN SECTION.
111 001110 EXAMPLE.
112 001120*
113 001130 SORT SORTWORK
114 001140 ON DESCENDING KEY S-TOTAL S-SUGAKU
115 001150 INPUT PROCEDURE IS INPUT-OWN
116 001160 OUTPUT PROCEDURE IS OUTPUT-OWN.
117 001170 STOP RUN.
118 001180*
119 001190*****LEVEL 1*****
120 001200*
121 001210* 入力手続き
122 001220*
123 001230 INPUT-OWN SECTION.
124 001240 IN-OPEN.
125 001250 OPEN INPUT INFILE.
126 001260 IN-MAIN.
127 001270 READ INFILE AT END
128 001280 GO TO IN-CLOSE
129 001290 END-READ.
130 001300*
131 001310 MOVE JUKEN-NO TO S-JUKEN-NO.
132 001320 MOVE SHIMEI TO S-SHIMEI.
133 001330 MOVE TOKUTEN TO S-TOKUTEN.
134 001340 ADD SUGAKU EIGO KOKUGO GIVING S-TOTAL.
135 001350*
136 001360 RELEASE SORTRECORD.
137 001370 GO TO IN-MAIN.
138 001380 IN-CLOSE.
139 001390 CLOSE INFILE.
140 001400*
141 001410* 出力手続き

```

```

142 001420*
143 001430 OUTPUT-OWN SECTION.
144 001440 OUT-OWN.
145 001450 OPEN OUTPUT OUTFILE.
146 001460 PERFORM PAGE-HEAD-RTN THRU PAGE-HEAD-EXT.
147 001470 MOVE ZERO TO W-NO.
148 001480 MOVE ZERO TO W-LINE-NO.
149 001490*
150 001500 OUT-MAIN.
151 001510 RETURN SORTWORK AT END GO TO OUT-CLOSE.
152 001520 ADD 1 TO W-LINE-NO.
153 001530*
154 001540 IF W-LINE-NO > 50
155 001550 MOVE ZERO TO W-LINE-NO
156 001560 PERFORM PAGE-HEAD-RTN THRU PAGE-HEAD-EXT
157 001570 END-IF.
158 001580*
159 001590 ADD 1 TO W-NO.
160 001600 MOVE SPACE TO OUTRECORD.
161 001610 MOVE W-NO TO P-NO.
162 001620 MOVE S-JUKEN-NO TO P-JUKEN-NO.
163 001630 MOVE S-SHIMEI TO P-SHIMEI.
164 001640 MOVE S-SUGAKU TO P-SUGAKU.
165 001650 MOVE S-EIGO TO P-EIGO.
166 001660 MOVE S-KOKUGO TO P-KOKUGO.
167 001670 MOVE S-TOTAL TO P-TOTAL.
168 001680*
169 001690 WRITE OUTRECORD FROM EDITRECORD
170 001700 AFTER ADVANCING 1 LINE
171 001710 END-WRITE.
172 001720*
173 001730 GO TO OUT-MAIN.
174 001740 OUT-CLOSE.
175 001750 CLOSE OUTFILE.
176 001760*
177 001770*****LEVEL 3*****

```

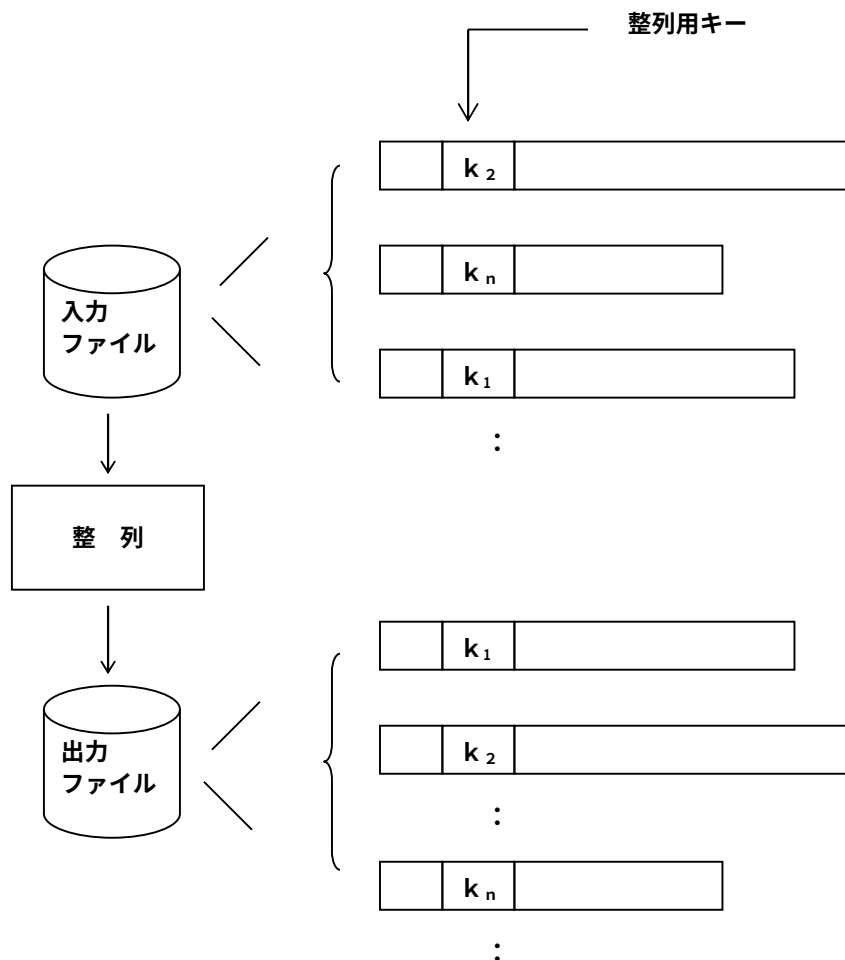
```
178 001780*
179 001790 LEVEL2 SECTION.
180 001800 PAGE-HEAD-RTN.
181 001810 WRITE OUTRECORD FROM HEADING-1
182 001820 AFTER ADVANCING PAGE
183 001830 END-WRITE.
184 001840*
185 001850 WRITE OUTRECORD FROM HEADING-2
186 001860 AFTER ADVANCING 2 LINES
187 001870 END-WRITE.
188 001880*
189 001890 MOVE SPACE TO OUTRECORD.
190 001900 WRITE OUTRECORD
191 001910 AFTER ADVANCING 2 LINES
192 001920 END-WRITE.
193 001930 PAGE-HEAD-EXT.
194 001940 EXIT.
```

## 10.5 可変長レコードの整列併合

次に、整列併合するレコードの長さが異なる場合の記述方法について述べる。

COBOL では、整列併合用のファイルが可変長レコード形式のとき、**USING** 句指定で入力したレコードの長さで整列併合用ファイルにレコードが渡される。また、**GIVING** 句指定の出力ファイルが可変長レコード形式のとき、整列併合用ファイルより入力したレコードの長さで、出力ファイルに出力される。有効なレコード分だけを整列併合用ファイルに渡すことにより、整列併合用ファイルを有効に使い、また、実行効率を上げることができる。

例 各レコードの長さが異なるレコードを入力ファイルより入力して並べ替え, もとの長さのまま出力ファイルに書き出す。



入力ファイル, 出力ファイル, 整列併合用ファイルが可変長レコード形式の例を次に示す。

IDENTIFICATION DIVISION.

PROGRAM-ID. EXAMPLE.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. Express5800.

OBJECT-COMPUTER. Express5800.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT INFILE ASSIGN TO "TPFILE-RAF".

SELECT SORTWORK ASSIGN TO "SRTWRK-RAF".



```

 SELECT OUTFILE ASSIGN TO "LPFILE-RAF".
*
DATA DIVISION.
FILE SECTION.
FD INFILE LABEL RECORDS ARE STANDARD
 RECORD VARYING 50 TO 100.
01 INRECORD-1.
 02 INRECORD-11 PIC X(100).
01 INRECORD-2.
 02 INRECORD-22 PIC X(50).
SD SORTWORK
 RECORD VARYING 8 TO 100.
01 SRTRECORD.
 02 SRTREC-1 PIC X(5).
 02 SRT-KEY PIC 9(3).
 02 FILLER PIC X(92).
FD OUTFILE LABEL RECORD IS STANDARD
 RECORD VARYING 8 TO 100.
01 OUTREC-1.
 02 OUTREC-12 PIC X(100).
WORKING-STORAGE SECTION.
 :
PROCEDURE DIVISION.
 :
SAMPLE.
 SORT SORTWORK
 ON DESCENDING KEY SRT-KEY
 USING INFILE
 GIVING OUTFILE.

```

## 10.6 整列併合プログラムの実行

次の説明の都合上, 整列併合機能を含む COBOL 実行用プログラムを簡単に整列併合プログラムと呼ぶ。

### 10.6.1 整列併合用ファイル

整列併合プログラムは, 整列併合用ファイルを使用する。

整列併合用ファイルは, システムで自動的に一時ファイルを確保するので利用者は意識する必要がない。一時ファイルは SORT 文または MERGE 文の開始時に確保され, SORT 文または MERGE 文の終了時に解放される。

### 10.6.2 注意事項

(1) 本コンパイラは, SORT 文, MERGE 文の実行に, SORTKIT を利用する。SORTKIT を利用した場合, 整列併合機能実行時に, メモリの大きさ等の変更が可能となる。

なお, 変更方法, 諸元等の詳細については次の関連説明書を参照。

【Enterprise for Windows】関連説明書: 『SORTKIT/Enterprise 説明書』

【Standard】関連説明書: 『SORTKIT/Enterprise 説明書』

【Enterprise for Linux】関連説明書: 『SORTKIT 利用の手引』

## 第11章 翻訳指示機能

### 11.1 概説

翻訳指示機能は、翻訳時に登録集の原文 (text)を原始プログラム中に複写する機能を提供する。このとき複写された原文は、原始プログラムの一部として書かれているものと同じものとして扱われる。

利用者はこの機能を使うことにより、次のような利点が得られる。

- (1) 原始プログラムで共通な部分(たとえば見出し部とか環境部の一部)を登録しておくことにより、コーディングの分量を減らすことができる。
- (2) 共通処理(たとえばエラー処理ルーチンとかある一定の計算または処理ルーチン)を登録しておくことにより、原始プログラムの標準化を容易に行うことができる。
- (3) 複数の原始プログラムから参照するファイルの記述(SELECT 句, ファイル記述項, ファイルの 01 レコード記述など)を登録しておくことによって、ファイルの仕様変更に対する修正を登録集の原文だけに留めることができ、原始プログラム間のインタフェースミスを防ぐこともできる。

[Enteprise] 登録集原文を原始プログラム中に複写するために COPY 文があり、登録集の原文名に ".cob", ".cbl"または ".cbf" を付与したファイル名でエディタを使用して作成、原始プログラムと同様に登録集原文の格納場所に登録する。

[Standard] 登録集原文を原始プログラム中に複写するために COPY 文があり、登録集の原文名に ".cob", ".cbl", ".cbf"または ".cbx" を付与したファイル名でエディタを使用して作成、原始プログラムと同様に登録集原文の格納場所に登録する。

### 11.2 COPY 文

登録集原文を原始プログラム中に複写するために、翻訳指示文として COPY 文がある。

COPY 文は原始プログラム中で文字列または分離符が書ける所ならどこに書いてもよい。ただし、登録集原文の中で COPY 文を書くことはできない。

COPY 文を書くと COPY 文全体が指定された登録集原文に置き換わり、それが原始プログラムに書かれているのと同様に翻訳される。

### 11.2.1 登録集原文名の指定

COPY 文に登録集原文名を記述すると、その登録集原文が原始プログラム内に複写される。

COPY 文で指定された登録集原文名は、次の検索規則に従って登録集のディレクトリ内が検索され、原始プログラム内に複写される。

#### (1) ディレクトリに対する検索規則

登録集原文は、次の順序に従い、ファイルの検索を実行する。

なお、登録集原文が見つかった場合は、その時点で検索は終了し、登録集原文の内容が原始プログラム内に複写される。

(a) -I オプションで指定したディレクトリ内を検索する。

複数指定の場合は、指定した順に検索を行う。

(b) 環境変数 COB\_COPY で指定したディレクトリ内を検索する。

複数指定の場合は、指定した順に検索を行う。

(c) 原始プログラムファイルのディレクトリ内を検索する。

いずれのディレクトリ中にも該当するファイルがない場合は、致命的エラーとなる。

ファイル拡張子の異なる同じ名前の登録集原文が記述可能である。この場合は、次の検索規則に従って登録集原文が決定され、原始プログラム内に複写される。

#### (2) ファイルに対する検索規則

COPY 文に指定する登録集原文名は、拡張子を含まない形式で指定する。

ただし登録集原文は拡張子を指定して作成するため、同一の登録集原文名をもつ形式の異なるファイルを同時に作成することができるため、以降に登録集原文名の検索順を示す。

(a) [Enteprise] 原始プログラムのファイル拡張子が “.cob” または “.cbl” の場合登録集原文名に “.cob” “.cbl” または “.cbf” を付与したファイルを検索する。 “.cob” “.cbl” “.cbf” の順に検索する。したがって、登録集原文は、 “登録集原文名.cob” “登録集原文名.cbl” または “登録集原文名.cbf” のファイルとして作成する必要がある。

[Standard] 原始プログラムのファイル拡張子が “.cob” “.cbl” または “.cbx” の場合登録集原文名に “.cob” “.cbl” “.cbx” または “.cbf” を付与したファイルを検索する。 “.cob” “.cbl” “.cbx” “.cbf” の順に検索する。したがって、登録集原文は、 “登録集原文名.cob” “登録集原文名.cbl” “登録集原文名.cbx” または “登録集原文名.cbf” のファイルとして作成する必要がある。

(b) [Enteprise] 原始プログラムのファイル拡張子が “.cbf” の場合

登録集原文名に “.cbf” または “.cob” “.cbl” を付与したファイルを検索する。 “.cbf” “.cob” “.cbl” の順に検索する。したがって、登録集原文は、 “登録集原文

名.cbf”または”登録集原文名.cob” ”登録集原文名.cbl”のファイルとして作成する必要がある。

[Standard] 原始プログラムのファイル拡張子が”.cbf”の場合

登録集原文名に”.cbf”または”.cob” ”.cbl” ”.cbx”を付与したファイルを検索する。”.cbf” ”.cob” “.cbl” “.cbx”の順に検索する。したがって、登録集原文は、”登録集原文名.cbf”または”登録集原文名.cob” ”登録集原文名.cbl” ”登録集原文名.cbx”のファイルとして作成する必要がある。

#### 補足

運用時には、特別な理由がない場合、登録集原文は1つの登録集原文格納場所に格納することを推奨する

(拡張子を除いた、登録集原文名で一意のファイルになるような管理を行う)

## 11.2.2 REPLACING 句

COPY 文に REPLACING 句が書いてある場合、登録集原文がそのまま原始プログラムに複写されるのではなく、REPLACING 句で指定された置き換えが行われ、複写される。置き換え処理は『COBOL 言語説明書』の COPY 文で述べた規則に従って行われる。置き換えの例を次に示す。

### (1) 仮原文による置き換え

#### (a) 仮原文の置き換えでは

- i. 仮原文 1 または登録集原文中の分離符のコンマまたはセミコロンは1つの空白として処理する。
- ii. 登録集原文および仮原文 2 中の注記行は原始プログラムにそのまま複写される。
- iii. 仮原文 1 の文と同じ登録集原文中の文が仮原文 2 と置き換えられる。

例

登録集原文 A

```
ADD 1 TO A.
MOVE A TO B. GO TO ~ .
```

原始プログラム

```

:
COPY A REPLACING
== MOVE A TO B. == BY
== ADD A B GIVING C. ==
:
```

置き換え後の原始プログラム

```

 :
 ADD 1 TO A.
 ADD A B GIVING C. GO TO ~ .
 :

```

(2) 仮文字列による置き換え

(a) 仮文字列の置き換えでは

- i. 利用者語だけが置き換えの対象となり, 予約語や定数は置き換えの対象とならない。
- ii. 利用者語の中で仮文字列 1 と同じ文字列部分が仮文字列 2 と置き換えられる。

例

登録集原文 A

```

02 JOSU-PICT PICTURE X(7) VALUE "PICTURE".
02 PICTAREA PICTURE X(30).
02 NUM123P PICTURE 9(3) VALUE 123.

```

原始プログラム

```

 :
 COPY A REPLACING //PICT// BY //XXX//
 //123// BY //999//.
 :

```

置き換え後の原始プログラム

```

:
02 JOSU-XXX PICTURE X(7) VALUE "PICTURE".
02 XXXAREA PICTURE X(30).
02 NUM999P PICTURE 9(3) VALUE 123.
:
```

### 11.3 登録集原文

登録集原文は、原始プログラムと同様にエディタで作成する。

登録集原文の単位に制限はなく、小は語から大はプログラム全体にいたるまでどのような形式でもよい。

### 11.4 デバッグ行と COPY 文

デバッグ行に COPY 文が書いてある場合、または登録集原文中にデバッグ行が使用されている場合は、次のように扱う。

- (1) COPY 文がデバッグ行に書いてあり、デバッグ行が注記行として扱われた場合、COPY 文そのものも注記行として扱うため、登録集原文の複写は行わない。登録集原文(ファイル)はなくてもよい。
- (2) COPY 文がデバッグ行に書いてあり、デバッグ行を注記行として扱わない場合、複写した原文もすべてデバッグ行とみなす。ただし、複写した原文の行が注記行(標識領域に＊または／の書かれている行)の場合は、そのまま注記行として扱う。
- (3) 登録集原文内でデバッグ行が書いてある場合、それはデバッグ行とみなす。

### 11.5 選択行と COPY 文

選択行に COPY 文が書いてある場合、または登録集原文中に選択行が使用されている場合は、次のように扱う。

- (1) COPY 文が選択行に書いてあり、選択行が無効な場合、COPY 文そのものも注記行として扱うため、登録集原文の複写は行わない。登録集原文(ファイル)はなくてもよい。
- (2) COPY 文が選択行に書いてあり、選択行が有効な場合、登録集原文を複写する。
- (3) 登録集原文内で選択行が書いてある場合、それは選択行とみなす。



## 第12章 例外処理機能

### 12.1 概説

例外処理機能は、実行用プログラムの実行時に不正 10 進データ例外や添字の範囲外エラー、またはファイル入出力エラー等が発生したときに、そのプログラムの実行をただちに終了させず、エラー処理としてユーザが記述した処理を実行する機能を持っている。

たとえば、オンライントランザクションによるオンラインプログラム等のデバッグ時においては、入力データ等の誤りによって、数字データ項目に数字以外のデータが入力されることがある。このようなとき、数字比較、数字転記および算術演算等の実行中、不正 10 進データ例外が発生し、例外処理機能を使用していないプログラムはただちに強制終了する。しかし、オンライントランザクション配下においては、そのプロセスが異常終了したことをシステムに知らせる等の処理が必要であり、例外処理機能を使用することによって、それらの処理を実行させることができる。

なお、本章でいう例外とは、オペレーティングシステムシステムのシグナルによって発生する例外ではなく、COBOL 実行環境が検出する COBOL 独自の例外であることに注意されたい。

## 12.2 例外手続き

実行用プログラムの実行時に例外が発生したとき、制御を利用者の指定した例外節に移し、例外節を実行後に強制終了する。

例外節は、宣言部分内に USE(例外節で使用)文および必要な例外手続きを書くことによって指定する。なお、入れ子プログラムにおいて、含まれるプログラムには例外手続きは記述できない。

DECLARATIVES.

[例外節以外の節]

:

節名 SECTION.

USE [GLOBAL] AFTER STANDARD { ERROR  
EXCEPTION }

PROCEDURE ON EXCEPTION

例外処理手続き

[例外節以外の節]

:

END DECLARATIVES.

例外手続きには、例外節に対して暗に GO TO 文が実行されたかのように制御が渡されるため、EXIT PROGRAM 文以外の文ならどの文を書いてもよいが、例外手続きの実行中、再度不正 10 進データ例外等の例外が発生すると、そのプログラムはただちに強制終了するので注意されたい。

さらに、例外節中で次の特殊レジスタにより、例外の発生した行番号および例外の種類を知ることができる。

**(1) EXCEPTION-CODE**

特殊レジスタ **EXCEPTION-CODE** は、暗に **COMPUTATIONAL-2** の数字項目として定義されている。この特殊レジスタには、発生した例外コードが設定されており、例外が発生したときの例外節中で参照できる。

設定される例外コードの内容については、「付録 E 例外コード一覧」を参照。

**(2) EXCEPTION-LINE**

特殊レジスタ **EXCEPTION-LINE** は、暗に **COMPUTATIONAL-2** の数字項目として定義されている。この特殊レジスタは、例外が発生した行番号が設定されており、例外が発生したときの例外節中で参照できる。

なお、設定される行番号は **-Ce** オプション指定時は外部行番号、**-Ce** オプションの指定がないときは内部行番号(コンパイラが生成した一連番号)であることに注意されたい。

## 12.3 例外の説明

次に主な例外について説明する。

### (1) 不正 10 進データ例外(例外コード:094,010)

不正 10 進データ例外(illegal decimal data exception)は、記憶域内のデータが参照されるとき次のような条件で検出される。

#### (a) 内部 10 進データ(パック形式データ)

- 符号位置の内容が不正( $A_{16} \sim F_{16}$  以外)である。
- 数字位置の内容が不正( $0 \sim 9$  以外)である。
- 桁数が偶数の時、最左端バイトの上位 4 ビットが 0 でない  
(-CN オプションが指定されていない時)。

#### (b) 外部 10 進データ(アンパック形式データ)

##### i. 重ね符号(符号部が独立していない)の場合

- 最右端(SIGN LEADING 句が指定されているときは最左端)バイトの内容が不正(表 12-1 以外の値が設定されている)である。
- 数字位置の内容が不正( $0 \sim 9$  以外)である。

表 12-1 最右端(最左端)バイトの正しい内容

| 最右端(最左端)バイト<br>の値 | 符 号              |                  |                  |                  |                  |                  |
|-------------------|------------------|------------------|------------------|------------------|------------------|------------------|
|                   | 符号なし             | 正符号付き            |                  |                  | 負符号付き            |                  |
|                   | SDn              | SDn              | SDn              | SDn              | SDn              | SDn              |
| 0                 | 30 <sub>16</sub> | 7B <sub>16</sub> | 78 <sub>16</sub> | 24 <sub>16</sub> | 7D <sub>16</sub> | 7A <sub>16</sub> |
| 1                 | 31 <sub>16</sub> | 41 <sub>16</sub> | 7E <sub>16</sub> | 9F <sub>16</sub> | 4A <sub>16</sub> | A0 <sub>16</sub> |
| 2                 | 32 <sub>16</sub> | 42 <sub>16</sub> | CD <sub>16</sub> | 53 <sub>16</sub> | 4B <sub>16</sub> | E0 <sub>16</sub> |
| 3                 | 33 <sub>16</sub> | 43 <sub>16</sub> | CE <sub>16</sub> | 54 <sub>16</sub> | 4C <sub>16</sub> | E1 <sub>16</sub> |
| 4                 | 34 <sub>16</sub> | 44 <sub>16</sub> | CF <sub>16</sub> | 55 <sub>16</sub> | 4D <sub>16</sub> | E2 <sub>16</sub> |
| 5                 | 35 <sub>16</sub> | 45 <sub>16</sub> | D0 <sub>16</sub> | 56 <sub>16</sub> | 4E <sub>16</sub> | E3 <sub>16</sub> |
| 6                 | 36 <sub>16</sub> | 46 <sub>16</sub> | D1 <sub>16</sub> | 57 <sub>16</sub> | 4F <sub>16</sub> | E4 <sub>16</sub> |
| 7                 | 37 <sub>16</sub> | 47 <sub>16</sub> | D2 <sub>16</sub> | 58 <sub>16</sub> | 50 <sub>16</sub> | E5 <sub>16</sub> |
| 8                 | 38 <sub>16</sub> | 48 <sub>16</sub> | D3 <sub>16</sub> | 59 <sub>16</sub> | 51 <sub>16</sub> | E6 <sub>16</sub> |
| 9                 | 39 <sub>16</sub> | 49 <sub>16</sub> | D4 <sub>16</sub> | 5A <sub>16</sub> | 52 <sub>16</sub> | E7 <sub>16</sub> |

本コンパイラ固有の表現形式は上記のとおりである。

-CS オプション指定により、Pro\*COBOL 表現形式に変更することができる。その場合の数字と符号の重ね合わせを次に示す。

| 最右端(最左端)<br>バイトの値 | 符 号              |                  |                  |
|-------------------|------------------|------------------|------------------|
|                   | 符号なし             | 正符号付き            | 負符号付き            |
|                   | SDn              | SDn              | SDn              |
| 0                 | 30 <sub>16</sub> | 30 <sub>16</sub> | 70 <sub>16</sub> |
| 1                 | 31 <sub>16</sub> | 31 <sub>16</sub> | 71 <sub>16</sub> |
| 2                 | 32 <sub>16</sub> | 32 <sub>16</sub> | 72 <sub>16</sub> |
| 3                 | 33 <sub>16</sub> | 33 <sub>16</sub> | 73 <sub>16</sub> |
| 4                 | 34 <sub>16</sub> | 34 <sub>16</sub> | 74 <sub>16</sub> |
| 5                 | 35 <sub>16</sub> | 35 <sub>16</sub> | 75 <sub>16</sub> |
| 6                 | 36 <sub>16</sub> | 36 <sub>16</sub> | 76 <sub>16</sub> |
| 7                 | 37 <sub>16</sub> | 37 <sub>16</sub> | 77 <sub>16</sub> |
| 8                 | 38 <sub>16</sub> | 38 <sub>16</sub> | 78 <sub>16</sub> |
| 9                 | 39 <sub>16</sub> | 39 <sub>16</sub> | 79 <sub>16</sub> |

ii. 独立符号の場合

- 符号位置の内容が不正(2B<sub>16</sub>, 2D<sub>16</sub> 以外)である。
- 数字位置の内容が不正(0～9 以外)である。

|    |                                                                                                                                                                                                                                                                         |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 備考 | <p>翻訳時に-Cd オプション指定で翻訳された実行可能プログラムにおいては不正 10 進データ例外は発生しない。</p> <p>外部 10 進データ項目に、文字データが上記(b)以外の条件で格納されていると、不正 10 進データ例外は発生しない。すなわち、10 進データとして取り扱われる。したがって、字類条件を使用して、数字データか否かの検査をした方が望ましい。</p> <p>また、外部 10 進データの最右端バイトが 16 進の "20" の場合、不正 10 進データ例外は発生せず、値ゼロとして取り扱われる。</p> |
| 例  | 外部 10 進データ(文字):12C△                                                                                                                                                                                                                                                     |

|                           |                                                                                                                                                                                                       |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | 内部表現形式 (16 進) : 31324320<br>上記の外部 10 進データは 10 進数値, +1230 と<br>して取り扱われる。                                                                                                                               |
| [Enterprise]<br>ACOS-4 互換 | -Fe オプションを指定することにより,<br>NUMERIC 検査の内容を変更することができる。<br>・偶数桁の内部 10 進数の最左端上位 4 ビットが 0<br>以外の場合, 偽とする (-CN オプションが指定さ<br>れていない時)。不正 10 進エラーにはならな<br>い。<br>・外部 10 進数の独立符号部が空白の場合, 真と<br>する。不正 10 進エラーにはならない。 |

## (2) SORT/MERGE 例外 (例外コード: 401~409)

SORT/MERGE 文において発生する例外である。409 の例外は, 関連製品で発生する例外である。

それぞれの例外コードの内容については, 「付録 E 例外コード一覧」を参照。

## (3) 添字／可変繰り返し数／部分参照例外 (例外コード: 002, 003)

一意名に指定されている添字の値が, 最大繰り返し数を超過している, または部分参照において最左端文字位置, 長さが不正である場合に発生する例外である。

それぞれの例外コードの内容については, 「付録 E 例外コード一覧」を参照。

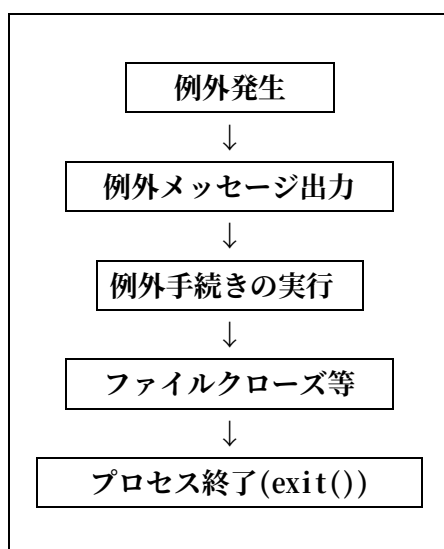
## 12.4 例外発生時の処理の流れ

例外が発生したプログラムに例外節があれば、暗に GO TO 文が実行されたかのように、例外節に制御が渡される。

例外節は、実行用プログラムごとに記述しなければならない。

たとえば、活性化する側の実行時要素に例外節が記述されているが、活性化される側の実行時要素に例外節が記述されていないと、その活性化される側の実行時要素で例外が発生しても、活性化する側の実行時要素で記述した例外節に制御は渡らない。

また、例外が発生した場合には、通常次の順序で標準エラー処理が行われる。



### (1) 例外メッセージ出力

例外が発生すると、ただちにその例外を示すメッセージが出力される。メッセージ出力後は確認入力待ちになる。

### (2) 例外手続きの実行

例外メッセージ出力後、例外節の手続きが実行される。この手続きの中で再度例外が発生した場合には、ただちにプロセスが終了(exit())する。

### (3) ファイルクローズ等

そのときに COBOL 実行環境が確保しているメモリの解放および COBOL の OPEN 文によって開かれたままとなっているファイルのクローズを行う。

### (4) プロセス終了(exit())

プロセスを終了する。

## 12.5 例外メッセージ制御

例外メッセージの出力は、環境変数 `COB_RUNERR` により抑制することができる。

オンライントランザクション配下で動作する実行用プログラムで、エラーメッセージの出力を抑制しておく必要がある場合、次のように設定する。

**[Enterprise for Windows] [Standard] バッチファイルの設定例**

```
set COB_RUNERR=NO_DISPLAY
```

**[Enterprise for Linux] シェルファイルの設定例(bashの場合)**

```
export COB_RUNERR=NO_DISPLAY
```

例外メッセージの出力後の確認入力待ちを行わないようにする必要がある場合、次のように設定する。

**[Enterprise for Windows] [Standard] バッチファイルの設定例**

```
set COB_RUNERR=NO_PAUSE
```

**[Enterprise for Linux] シェルファイルの設定例(bashの場合)**

```
export COB_RUNERR=NO_PAUSE
```



## 第13章 システムサブルーチン

### 13.1 概説

本システムでは、言語機能で提供する各種機能のほかに CALL 文によって呼び出しが可能な次の 13 種類のシステムサブルーチンを提供する。

- (1) コマンド行の引数の取り込み
- (2) 環境変数の取り込み
- (3) 環境変数の設定
- (4) エラーコード取得
- (5) シェルコマンド実行
- (6) 4 桁での西暦取得
- (7) コード変換
- (8) 基本多言語面以外の文字検査
- (9) プリンタファイル
- (10) 別プログラムの起動
- (11) ファイル入出力
- (12) 内部ファイル ID 置換表の解放
- (13) ファイル入出力状態の永続誤り解除

本システムでは、システムサブルーチンの呼び出しについて、CALL 文の規則に準じる。  
各システムサブルーチンの規則は、CALL 文との差分のみ記載する。

### 13.2 システムサブルーチンの詳細

#### 13.2.1 コマンド行の引数の取り込み

機 能

コマンド行の引数を取り込む。

一般形式

**CALL "B\_CMOPT"**

**USING** データ名 1 データ名 2

## 規 則

- (1) データ名 1 およびデータ名 2 は、次の形式で定義する。  
     01 データ名 1   COMP-2  
     01 データ名 2   PIC   X(256).
- (2) データ名 1 はコマンド行の引数の番号を指定する。
- (3) データ名 2 はデータ名 1 で指定された番号の、コマンド行の引数の文字列を返却する。
- (4) データ名 1 に 0 が指定されたときは、データ名 1 に起動したコマンド行の引数の個数、データ名 2 に起動したコマンド名を返却する。
- (5) データ名 1 に指定されたコマンド行の引数の番号に対応する引数が見つからないとき、またはデータ名 1 に指定されたコマンド行の引数の番号に対応する引数のコード変換に失敗したとき、またはデータ名 1 に指定された番号が負数のときはデータ名 1 に 0 が返却される。  
     コード変換に失敗したとき、データ名 2 には変換に失敗した文字を 00<sub>16</sub> に変換した文字列が設定される。
- (6) 本システムサブルーチンを用いるときは、メインプログラムは COBOL 言語で記述されていなければならない。

## 13.2.2 環境変数の取り込み

## 機 能

指定した環境変数の値を取り込む。

## 一般形式

```
CALL "B_GETENV"

USING データ名 1 データ名 2
```

## 規 則

- (1) データ名 1 およびデータ名 2 は、次の形式で定義する。  
     01 データ名 1   PIC   X(256).  
     01 データ名 2   PIC   X(256).
- (2) データ名 1 は環境変数を指定する。
- (3) データ名 2 は、データ名 1 に指定した環境変数に対する値を返却する。対応する環境変数がない場

合は、データ名 2 に表意定数 SPACE を返却する。

(4) データ名 1 は、左詰めにデータが格納されていなければならない。

データ名 1 の左端より順にスキャンして、最初の SPACE の前の文字までを環境変数として扱う。また、データ名 1 の先頭が SPACE のときは、環境変数の取り込みの処理はされない。

### 13.2.3 環境変数への設定

#### 機 能

指定した環境変数へ値を設定する。

#### 一般形式

```
CALL "B_PUTENV"
 USING データ名 1 データ名 2
```

#### 規 則

- (1) データ名 1 およびデータ名 2 は、次の形式で定義する。  
 01 データ名 1 PIC X(256).  
 01 データ名 2 PIC X(256).
- (2) データ名 1 は、環境変数を指定する。
- (3) データ名 2 は、データ名 1 に指定した環境変数に対する値を指定する。
- (4) 環境変数を格納する領域を確保できないときは、データ名 2 に表意定数 SPACE を返却する。
- (5) 次に記す規則はデータ名 2 にも当てはまる。  
 データ名 1 は、左詰めにデータが格納されていなければならない。  
 データ名 1 の左端より、順にスキャンして、最初の SPACE の前の文字までをデータとして扱う。また、データ名 1 の先頭が SPACE のときは、環境変数への設定の処理はされない。
- (6) 本システムサブルーチンによって変更された既存の環境変数の値は、本システムサブルーチンが実行されるプロセス内およびそのプログラムから起動するプログラム内でのみ取り出し可能である。
- (7) 本システムサブルーチンによって新しく作成された環境変数および設定された値は、サブルーチンが実行されるプロセス内およびそのプログラムから起動するプログラム内でのみ取り出し可能である。
- (8) COBOL WS エミュレータ経由で実行する場合、CBLSPAWN システムサブルーチンで起動したプログラム内では取り出すことができない。環境変数の引継ぎはできない。

## 13.2.4 エラーコード取得

### 機 能

詳細な情報のあるエラーに関して、その情報を取得する。

### 一般形式

**CALL "B\_GERRNO" USING データ名 1**

### 規 則

- (1) データ名 1 は、COMP-2 でなければならない。
- (2) データ名 1 には、詳細エラーコードが返却される。

詳細エラーコードとファイルの種別を以下に示す。

| ファイルの種別                               | 詳細エラーコード返却 | 詳細エラーコードの意味                                                                                                                                                                                                     |
|---------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 順ファイル<br>(COBOL 独自形式)                 | ○          | [Enterprise for Windows]<br>[Standard]                                                                                                                                                                          |
| 行順ファイル<br>(COBOL 独自形式)                | ○          | Visual C++ のエラーコードが通知される                                                                                                                                                                                        |
| 相対ファイル<br>(COBOL 独自形式)                | ○          | [Enterprise for Linux]<br>errno が返却される                                                                                                                                                                          |
| [Standard]<br>順ファイル<br>(IFAS ファイル形式)  | ○          | IFAS ファイルの説明書を参照                                                                                                                                                                                                |
| [Standard]<br>相対ファイル<br>(IFAS ファイル形式) | ○          |                                                                                                                                                                                                                 |
| [Standard]<br>索引ファイル<br>(IFAS ファイル形式) | ○          |                                                                                                                                                                                                                 |
| 索引ファイル<br>(ISAM 形式)                   | ○          | [Enterprise for Windows]<br>[Standard]<br>Visual C++ のエラーコードおよび<br>ISAM のエラー番号が通知される。<br>ISAM のエラー番号は『ISAM 利用<br>の手引』を参照<br>[Enterprise for Linux]<br>errno および iserrno が返却され<br>る。iserrno は『ISAM 利用の手<br>引』を参照 |

|                                                                  |   |                               |
|------------------------------------------------------------------|---|-------------------------------|
| [Standard]<br>索引ファイル<br>(COBOL DB アクセス使用)                        | × | —                             |
| [Standard]<br>RDB ファイル<br>(COBOL DB アクセス使用)                      | × | —                             |
| [Standard]<br>RDB ファイル<br>(IFASPRO RDB 使用)                       | ○ | IFASPRO RDB の説明書を参照           |
| [Enterprise for Windows] [Standard]<br>プリンタファイル<br>(プリントマネージャ経由) | ○ | —                             |
| [Standard]<br>プリンタファイル<br>(COBOL アプリケーション実行環境使用)                 | ○ | COBOL アプリケーション実行環境の<br>説明書を参照 |

順ファイル、相対ファイル、索引ファイルのファイルステータス ” 30 ” のエラーが発生した場合、または SORTKIT 利用時に SORTKIT よりエラーが返却された場合、次の値が返却される。

データ名 1 に返却される詳細エラーコードは、コードの重複を避けるため、次のような規則で設定される。

|                            |                                          |
|----------------------------|------------------------------------------|
| 既に永続誤りのエラーが発生しているファイルへの IO | 00000                                    |
| EXTERNAL／BASED 項目の属性エラー    | 30000 番台                                 |
| SORTKIT のエラー               | 40000 番台<br>(SORTKIT のエラーに<br>40000 を加算) |
| IFAS のエラー                  | 50000 番台                                 |
| errno                      | 60000 番台<br>(errno の値に 60000 を加<br>算)    |
| 相対ファイル(COBOL 独自形式)         | errno の値を返却する<br>(加算なし)                  |
| 索引ファイル(ISAM 形式)            | errno および iserrno の<br>値を返却する<br>(加算なし)  |
| GetLastError()の返却値         | その他                                      |

EXTERNAL／BASED 項目の属性エラーコードの意味は次のとおり。

|       |                                             |
|-------|---------------------------------------------|
| 30701 | ASSIGN 句で指定したデータポインタデータ項目にファイルのアド<br>レスが未設定 |
| 30702 | ファイル未オープン                                   |

|       |                                                                    |
|-------|--------------------------------------------------------------------|
| 30703 | レコード長不一致                                                           |
| 30705 | デバイスタイプ不一致                                                         |
| 30706 | 共用排他種別不一致                                                          |
| 30707 | アクセスモード不一致                                                         |
| 30708 | レコード形式不一致,<br>OPTIONAL 指定不一致,<br>LINAGE 指定不一致,<br>SHIFT-CODE 指定不一致 |
| 30709 | ファイル編成不一致                                                          |
| 30710 | 最小レコード長不一致                                                         |
| 30711 | 最大レコード長不一致                                                         |
| 30714 | 副キー指定不一致                                                           |
| 30715 | DUPLICATE 指定不一致                                                    |
| 30722 | 論理ページの構成不一致                                                        |
| 30724 | CODE-SET 指定不一致                                                     |

索引ファイル(ISAM)のエラーコードの意味は次のとおり。

|       |                            |
|-------|----------------------------|
| 30802 | 既に永続誤りのエラーが発生しているファイルへの IO |
| 30803 | コンパイラ生成情報が破壊されている          |

SORTKIT については、次の関連説明書を参照。

- [Enterprise for Windows] 関連説明書:『SORTKIT/Enterprise Ver6.0 説明書』
- [Standard] 関連説明書:『SORTKIT/Enterprise 説明書』
- [Enterprise for Linux] 関連説明書:『SORTKIT 利用の手引』

IFAS ファイルのステータスについては、次の関連説明書を参照。

- [Enterprise] 関連説明書:『IFASPRO Ver5.0 利用の手引』
- [Standard] 関連説明書:『IFAS FILE、IFASPRO RDB 利用の手引』

GetLastError()で返却されるシステムエラーコードおよび errno については、次を参照。

- [Enterprise for Windows] [Standard] Visual C++の説明書
- [Enterprise for Linux] Red Hat Enterprise Linux で提供されている公開情報

(3) 本システムサブルーチンは、詳細な情報のあるエラーが発生した直後に使用すること。他の状態で使用した場合には、返却される値は不定である。

使用例

|            |
|------------|
| COBOL 入出力文 |
|------------|

```
IF 入出力状態 = "30"
 THEN CALL "B_GERRNO" USING ERRNO
 ERRNO による処理
 ;
ELSE
 ;
```



### 13.2.5 シェルコマンドの実行

#### 機 能

シェルコマンド(バッチファイル)の発行を行う。

#### 一般形式

**CALL "B\_SYSTEM" USING データ名 1 データ名 2**

#### 規 則

- (1) データ名 1 は、256 桁の英数字項目で定義しなければならない。
- (2) データ名 1 には、端末からの入力と同様なコマンドおよびパラメータを指定する。
- (3) データ名 2 は、次の形式で定義する。

[Enterprise]

01 データ名 2 USAGE COMP-2.

[Standard]

01 データ名 2.

02 データ名 21 USAGE COMP-2.

02 データ名 22 USAGE COMP-2.

- (4) データ名 2 には、データ名 1 で指定した内容を実行した戻り値が設定される。

[Standard] 詳細は次のとおりである。

| データ名 21 | データ名 22                         | 詳 細                      |
|---------|---------------------------------|--------------------------|
| 0       | データ名 1 で指定されるコマンド内で戻り値として設定された値 | シェルコマンド(バッチファイル)は正常に終了した |
| 1       | 0                               | シェルコマンド(バッチファイル)の起動に失敗した |

#### 注意

[Standard] データ名 1 には、直接的にも間接的にも COBOL で作成したプログラムを指定してはならない。

[Standard] データ名 1 にオペレーティングシステムの内部コマンドを指定する場合は、コマンドインタプリタも指定する必要がある。

例 dir 内部コマンドを指定する場合

MOVE "cmd.exe /C dir" TO データ名 1.

### 13.2.6 4 桁での西暦取得

#### 機 能

西暦 4 桁を取得する。

#### 一般形式

**CALL "B\_SYSDATE" USING データ名 1**

#### 規 則

- (1) データ名 1 は、用途が表示用と指定された符号を伴わない 8 桁の数字(整数)項目を指定しなければならない。
- (2) 日付は、yyyymmdd の形式でデータ名 1 に格納される。ここで、yyyy は西暦 4 桁、mm は月、dd は日を意味する。

## 13.2.7 コード変換

## 機 能

CALL 文の定数 1 にコード変換のシステムサブルーチンを指定することにより、任意のデータのコード変換を行うことができる。

- EBCDIC カタカナ $\longleftrightarrow$ JIS コード変換

| サブルーチン名 | 機能                                  |
|---------|-------------------------------------|
| C-C1EJ  | EBCDIC カタカナ $\rightarrow$ JIS コード変換 |
| C-C1JE  | JIS $\rightarrow$ EBCDIC カタカナコード変換  |

- シフト JIS $\longleftrightarrow$ JIPS(J)コード変換

| サブルーチン名 | 機能                            |
|---------|-------------------------------|
| C_CSJ   | シフト JIS コードから JIPS(J) コードへの変換 |
| C_CJS   | JIPS(J) コードからシフト JIS コードへの変換  |

- シフト JIS $\longleftrightarrow$ UNICODE(UTF-8)コード変換

| サブルーチン名 | 機能                                   |
|---------|--------------------------------------|
| C_CSU8  | シフト JIS コードから UNICODE(UTF-8) コードへの変換 |
| C_CU8S  | UNICODE(UTF-8) コードからシフト JIS コードへの変換  |

- シフト JIS $\longleftrightarrow$ UNICODE(UTF-16 Little Endian)コード変換

| サブルーチン名  | 機能                                                  |
|----------|-----------------------------------------------------|
| C_CSU16L | シフト JIS コードから UNICODE(UTF-16 Little Endian) コードへの変換 |
| C_CU16LS | UNICODE(UTF-16 Little Endian) コードからシフト JIS コードへの変換  |

- UNICODE(UTF-8) $\longleftrightarrow$ UNICODE(UTF-16 Little Endian)コード変換

| サブルーチン名   | 機能                                                         |
|-----------|------------------------------------------------------------|
| C_CU8U16L | UNICODE(UTF-8) コードから UNICODE(UTF-16 Little Endian) コードへの変換 |

|           |                                                               |
|-----------|---------------------------------------------------------------|
| C_CU16LU8 | UNICODE(UTF-16 Little Endian) コードから<br>UNICODE(UTF-8) コードへの変換 |
|-----------|---------------------------------------------------------------|

- UNICODE(UTF-16 Little Endian)↔UNICODE(UTF-16 Big Endian)コード変換

| サブルーチン名     | 機能                                                                         |
|-------------|----------------------------------------------------------------------------|
| C_CU16LU16B | UNICODE(UTF-16 Little Endian) コードから<br>UNICODE(UTF-16 Big Endian) コードへの変換  |
| C_CU16BU16L | UNICODE(UTF-16 Big Endian) コード から<br>UNICODE(UTF-16 Little Endian) コードへの変換 |

[Standard] [Enterprise for Windows] -CF オプションを指定した場合, C\_CU8U16L, C\_CU16LU8, C\_CU16LU16B, C\_CU16BU16L は使用できない。

[Standard] [Enterprise for Windows] -CF オプションを指定した場合, 「FontAvenue UniAssist コード変換」 使用時に次のコード変換システムサブルーチンを使用することができる。

- シフト JIS↔JIPS(J)コード変換

| サブルーチン名  | 機能                            |
|----------|-------------------------------|
| C_CEX_SJ | シフト JIS コードから JIPS(J) コードへの変換 |
| C_CEX_JS | JIPS(J) コードからシフト JIS コードへの変換  |

- シフト JIS↔UNICODE(UTF-8)コード変換

| サブルーチン名   | 機能                                   |
|-----------|--------------------------------------|
| C_CEX_SU8 | シフト JIS コードから UNICODE(UTF-8) コードへの変換 |
| C_CEX_U8S | UNICODE(UTF-8) コードからシフト JIS コードへの変換  |

- シフト JIS↔UNICODE(UTF-16 Little Endian)コード変換

| サブルーチン名     | 機能                                                  |
|-------------|-----------------------------------------------------|
| C_CEX_SU16L | シフト JIS コードから UNICODE(UTF-16 Little Endian) コードへの変換 |
| C_CEX_U16LS | UNICODE(UTF-16 Little Endian) コードからシフト JIS コードへの変換  |

- JIPS(J)↔UNICODE(UTF-8)コード変換

| サブルーチン名   | 機能                                 |
|-----------|------------------------------------|
| C_CEX_JU8 | JIPS(J)コードから UNICODE(UTF-8)コードへの変換 |
| C_CEX_U8J | UNICODE(UTF-8)から JIPS(J)コードへの変換    |

- JIPS(J)↔UNICODE(UTF-16 Little Endian)コード変換

| サブルーチン名 | 機能 |
|---------|----|
|---------|----|

|             |                                                   |
|-------------|---------------------------------------------------|
| C_CEX_JU16L | JIPS(J)コードから UNICODE(UTF-16 Little Endian)コードへの変換 |
| C_CEX_U16LJ | UNICODE(UTF-16 Little Endian)から JIPS(J)コードへの変換    |

## 一般形式

CALL {
 “C-C1EJ”  
 “C-C1JE”  
 “C-CSJ”  
 “C-CJS”  
 “C-CSU8”  
 “C-CU8S”  
 “C-CSU16L”  
 “C-CU16LS”  
 “C-CU8U16L”  
 “C-CU16LU8”  
 “C-CU16LU16B”  
 “C-CU16BU16L”
 } USING データ名 1 データ名 2 データ名 3

[Standard] -CF オプションを指定した場合、次のコード変換システムサブルーチンを使用できる。

CALL {
 “C\_CEX\_SJ”  
 “C\_CEX\_JS”  
 “C\_CEX\_SU8”  
 “C\_CEX\_U8S”  
 “C\_CEX\_SU16L”  
 “C\_CEX\_U16LS”  
 “C\_CEX\_JU8”  
 “C\_CEX\_U8J”  
 “C\_CEX\_JU16L”  
 “C\_CEX\_U16LJ”
 } USING データ名 1 データ名 2 データ名 3

## 構文規則

- (1) データ名 1 は、変換するデータ項目を指定しなければならない。
- (2) データ名 2 は、変換結果を格納する際に十分な大きさであるデータ項目を指定しなければならない。
- (3) データ名 1、データ名 2 およびデータ名 3 は、ファイル節、作業場所節、局所記憶節、連絡節およびベースデータ節中のデータ項目として定義され、レベル番号は 01 か 77 でなければならない。各データ名中は、ファイル節で定義されたデータ項目であれば修飾してもよい。
- (4) データ名 1 およびデータ名 2 の属性は、C-C1EJ または C-C1JE サブルーチンを使用する場合、データ名 1 は英字項目、英数字項目、英数字編集項目、数字編集項目または集団項目、データ名 2 は英数字項目または集団項目でなければならない。  
C-CSJ、C-CJS または C-CU16LU16B、C-CU16BU16L サブルーチンを使用する場合、データ名 1 は日本語項目または日本語編集項目、データ名 2 は日本語項目でなければならない。

C\_CSU16L, C\_CU8U16L サブルーチンを使用する場合、データ名 1 は英字項目、英数字項目、英数字編集項目または数字編集項目、データ名 2 は日本語項目でなければならない。C\_CU16LS, C\_CU16LU8 サブルーチンを使用する場合、データ名 1 は日本語項目または日本語編集項目、データ名 2 は英数字項目または集団項目でなければならない。C\_CSU8, C\_CU8S サブルーチンを使用する場合、データ名 1 は英字項目、英数字項目、英数字編集項目または数字編集項目、データ名 2 は英数字項目または集団項目でなければならない。

集団項目には、可変繰り返しデータ項目が含まれてはならない。

[Standard] -CF オプションを指定した場合、次の規則も適用される。

C\_CEX\_SJ, C\_CEX\_JS, C\_CEX\_JU16L, C\_CEX\_U16LJ サブルーチンを使用する場合、データ名 1 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。また、データ名 2 は日本語項目でなければならない。C\_CEX\_SU16L, C\_CEX\_U8J サブルーチンを使用する場合、データ名 1 は英字項目、英数字項目、英数字編集項目または数字編集項目でなければならない。また、データ名 2 は日本語項目でなければならない。C\_CEX\_U16LS, C\_CEX\_JU8 サブルーチンを使用する場合、データ名 1 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。また、データ名 2 は英数字項目または集団項目でなければならない。C\_CEX\_SU8, C\_CEX\_U8S サブルーチンを使用する場合、データ名 1 は英字項目、英数字項目、英数字編集項目または数字編集項目でなければならない。また、データ名 2 は英数字項目または集団項目でなければならない。

- (5) C-C1EJ または C-C1JE サブルーチンを使用する場合、データ名 1 は、1～256 文字の英字項目、英数字項目、英数字編集項目、数字編集項目または集団項目、データ名 2 は、1～256 文字の英数字項目または集団項目でなければならない。

[Enterprise] C\_CSJ, C\_CJS サブルーチンを使用する場合、データ名 1 は、1～128 文字の日本語項目または日本語編集項目、データ名 2 は、1～128 文字の日本語項目でなければならない。

[Standard] C\_CSJ, C\_CJS サブルーチンを使用する場合、データ名 1 およびデータ名 2 は、1 文字の日本語項目として定義する。

[Standard] -CF オプションを指定した場合、次の規則も適用される。

データ名 1 は C\_CEX\_SJ, C\_CEX\_JS, C\_CEX\_U16LS, C\_CEX\_JU8, C\_CEX\_JU16L, C\_CEX\_U16LJ サブルーチンを使用する場合は 1～128 桁、C\_CEX\_SU8, C\_CEX\_U8S, C\_CEX\_SU16L, C\_CEX\_U8J サブルーチンを使用する場合は 1～256 桁でなければならない。また、データ名 2 は C\_CEX\_SJ, C\_CEX\_JS, C\_CEX\_U8J, C\_CEX\_U16LJ サブルーチンを使用する場合は 1～128 桁でなければならない。

- (6) データ名 3 は 2 文字または 3 文字の英数字項目または集団項目でなければならない。

#### サブルーチンパラメータの字類

| サブルーチン名  | データ名 1                                          | データ名 2                    | データ名 3                  |
|----------|-------------------------------------------------|---------------------------|-------------------------|
| "C-C1EJ" | 英字項目、英数字項目、英数字編集項目、数字編集項目または集団項目<br>(最大 256 文字) | 英数字項目、<br>集団項目(最大 256 文字) | 英数字項目、<br>集団項目(2, 3 文字) |
| "C-C1JE" | 英字項目、英数字項                                       | 英数字項目、                    | 英数字項目、                  |

|               |                                                      |                   |                         |
|---------------|------------------------------------------------------|-------------------|-------------------------|
|               | 目, 英数字編集項目,<br>数字編集項目または集<br>団項目<br>(最大 256 文字)      | 集団項目(最大 256 文字)   | 集団項目(2, 3 文字)           |
| "C_CSJ"       | 日本語項目または日本<br>語編集項目<br>(最大 128 文字)                   | 日本語項目(最大 128 文字)  | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CJS"       | 日本語項目または日本<br>語編集項目<br>(最大 128 文字)                   | 日本語項目(最大 128 文字)  | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CSU8"      | 英字項目, 英数字項<br>目, 英数字編集項目ま<br>たは数字編集項目<br>(最大 256 文字) | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CU8S"      | 英字項目, 英数字項<br>目, 英数字編集項目ま<br>たは数字編集項目<br>(最大 256 文字) | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CSU16L"    | 英字項目, 英数字項<br>目, 英数字編集項目ま<br>たは数字編集項目<br>(最大 256 文字) | 日本語項目             | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CU16LS"    | 日本語項目または日本<br>語編集項目<br>(最大 128 文字)                   | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CU8U16L"   | 英字項目, 英数字項<br>目, 英数字編集項目ま<br>たは数字編集項目<br>(最大 256 文字) | 日本語項目             | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CU16LU8"   | 日本語項目または日本<br>語編集項目<br>(最大 128 文字)                   | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CU16LU16B" | 日本語項目または日本<br>語編集項目<br>(最大 128 文字)                   | 日本語項目             | 英数字項目,<br>集団項目(2, 3 文字) |
| "C_CU16BU16L" | 日本語項目または日本<br>語編集項目                                  | 日本語項目             | 英数字項目,<br>集団項目(2, 3 文字) |

|                             |                                                 |                   |                         |
|-----------------------------|-------------------------------------------------|-------------------|-------------------------|
|                             | (最大 128 文字)                                     |                   |                         |
| [Standard]<br>"C_CEX_SJ"    | 日本語項目または日本語編集項目<br>(最大 128 文字)                  | 日本語項目(最大 128 文字)  | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_JS"    | 日本語項目または日本語編集項目<br>(最大 128 文字)                  | 日本語項目(最大 128 文字)  | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_SU8"   | 英字項目, 英数字項目,<br>英数字編集項目または数字編集項目<br>(最大 256 文字) | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_U8S"   | 英字項目, 英数字項目,<br>英数字編集項目または数字編集項目<br>(最大 256 文字) | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_SU16L" | 英字項目, 英数字項目,<br>英数字編集項目または数字編集項目<br>(最大 256 文字) | 日本語項目             | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_U16LS" | 日本語項目または日本語編集項目<br>(最大 128 文字)                  | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_JU8"   | 日本語項目または日本語編集項目<br>(最大 128 文字)                  | 英数字項目,<br>固定長集団項目 | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_U8J"   | 英字項目, 英数字項目,<br>英数字編集項目または数字編集項目<br>(最大 256 文字) | 日本語項目(最大 128 文字)  | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_JU16L" | 日本語項目または日本語編集項目<br>(最大 128 文字)                  | 日本語項目             | 英数字項目,<br>集団項目(2, 3 文字) |
| [Standard]<br>"C_CEX_U16LJ" | 日本語項目または日本語編集項目<br>(最大 128 文字)                  | 日本語項目(最大 128 文字)  | 英数字項目,<br>集団項目(2, 3 文字) |

一般規則



- (1) 各コード変換システムサブルーチンは、データ名 1 で示されるデータ(入力データと呼ぶ)を(2)の変換規則に従ってコード変換し、データ名 2 で示される領域に格納する。

データ名 2 に JUSTIFIED 句の指定がある場合には右詰めで格納し、JUSTIFIED 句の指定がなければ左詰めで格納する。

変換後のデータの長さ(文字数)よりデータ名 2 で指定されたデータ項目の文字数が長い場合、データ名 2 の残りの部分には変換後のデータのコード体系に基づく空白文字が詰められる。

変換後のデータの長さ(文字数)よりデータ名 2 で指定されたデータ項目の文字数が短い場合、変換後のデータの残りの部分は切り捨てられる。

- (2) 各コード変換システムサブルーチンは、入力データを次の変換規則に従ってコード変換する。

(a) EBCDIC カタカナ $\longleftrightarrow$ JIS コード変換

- C-C1EJ(EBCDIC カタカナ $\rightarrow$ JIS コード変換)

入力データ中の各文字は、EBCDIC カタカナ $\rightarrow$ JIS のコード変換規則に従って変換される。

- C-C1JE(JIS $\rightarrow$ EBCDIC カタカナコード変換)

入力データ中の各小文字は、JIS $\rightarrow$ EBCDIC カタカナのコード変換規則に従って変換される。

(b) シフト JIS $\longleftrightarrow$ JIPS(J)コード変換

- C-CSJ(シフト JIS コードの日本語項目を JIPS(J)コードへ変換)

入力データ中の各文字は、シフト JIS $\rightarrow$ JIPS(J)のコード変換規則に従って変換される。

- C-CJS(JIPS(J)コードの日本語項目をシフト JIS コードへ変換)

入力データ中の各文字は、JIPS(J) $\rightarrow$ シフト JIS のコード変換規則に従って変換される。

(c) シフト JIS $\longleftrightarrow$ UNICODE(UTF-8)コード変換

- C-CSU8(シフト JIS コードの日本語項目を UNICODE(UTF-8)コードへ変換)

入力データ中の各文字は、シフト JIS $\rightarrow$ UNICODE(UTF-8)のコード変換規則に従って変換される。

- C-CU8S(UNICODE(UTF-8)コードの項目をシフト JIS コードへ変換)

入力データ中の各文字は、UNICODE(UTF-8) $\rightarrow$ シフト JIS のコード変換規則に従って変換される。

(d) シフト JIS $\longleftrightarrow$ UNICODE(UTF-16 Little Endian)コード変換

- C-CSU16L(シフト JIS コードの英数字項目を UNICODE(UTF-16 Little Endian)コードへ変換)

入力データ中の各文字は、シフト JIS $\rightarrow$ UNICODE(UTF-16 Little Endian)のコード変換規則に従って変換される。

- C-CU16LS(UNICODE(UTF-16 Little Endian)コードの日本語項目をシフト JIS コードへ変換)

入力データ中の各文字は、UNICODE(UTF-16 Little Endian) $\rightarrow$ シフト JIS のコード変換規則に従って変換される。

(e) UNICODE(UTF-8) $\longleftrightarrow$ UNICODE(UTF-16 Little Endian)コード変換

- **C\_CU8U16L(UNICODE(UTF-8)コードの項目を UNICODE(UTF-16 Little Endian)コードへ変換)**  
入力データ中の各文字は、UNICODE(UTF-8)→UNICODE(UTF-16 Little Endian)のコード変換規則に従って変換される。
- **C\_CU16LU8(UNICODE(UTF-16 Little Endian)コードの日本語項目を UNICODE(UTF-8)コードへ変換)**  
入力データ中の各文字は、UNICODE(UTF-16 Little Endian)→UNICODE(UTF-8)のコード変換規則に従って変換される。

**(f) UNICODE(UTF-16 Little Endian)↔UNICODE(UTF-16 Big Endian)コード変換**

- **C\_CU16LU16B(UNICODE(UTF-8)コードの項目を UNICODE(UTF-16 Big Endian)コードへ変換)**  
入力データ中の各文字は、UNICODE(UTF-16 Little Endian)→UNICODE(UTF-16 Big Endian)のコード変換規則に従って変換される。
- **C\_CU16BU16L(UNICODE(UTF-16 Big Endian)コードの日本語項目を UNICODE(UTF-16 Little Endian)コードへ変換)**  
入力データ中の各文字は、UNICODE(UTF-16 Big Endian)→UNICODE(UTF-16 Little Endian)のコード変換規則に従って変換される。

**(g) [Standard] シフト JIS↔JIPS(J)コード変換**

- **C\_CEX\_SJ(シフト JIS コードの日本語項目を JIPS(J)コードへ変換)**  
入力データ中の各文字は、シフト JIS→JIPS(J)のコード変換規則に従って変換される。
- **C\_CEX\_JS(JIPS(J)コードの日本語項目をシフト JIS コードへ変換)**  
入力データ中の各文字は、JIPS(J)→シフト JIS のコード変換規則に従って変換される。

**(h) [Standard] シフト JIS↔UNICODE(UTF-8)コード変換**

- **C\_CEX\_SU8(シフト JIS コードの英数字項目を UNICODE(UTF-8)コードへ変換)**  
入力データ中の各文字は、シフト JIS→UNICODE(UTF-8)のコード変換規則に従って変換される。
- **C\_CEX\_U8S(UNICODE(UTF-8)コードの英数字項目をシフト JIS コードへ変換)**  
入力データ中の各文字は、UNICODE(UTF-8)→シフト JIS のコード変換規則に従って変換される。

**(i) [Standard] シフト JIS↔UNICODE(UTF-16 Little Endian)コード変換**

- **C\_CEX\_SU16L(シフト JIS コードの英数字項目を UNICODE(UTF-16 Little Endian)コードへ変換)**  
入力データ中の各文字は、シフト JIS→UNICODE(UTF-16 Little Endian)のコード変換規則に従って変換される。
- **C\_CEX\_U16LS(UNICODE(UTF-16 Little Endian)コードの日本語項目をシフト JIS コードへ変換)**  
入力データ中の各文字は、UNICODE(UTF-16 Little Endian)→シフト JIS のコード変換規則に従って変換される。

## (j) [Standard] JIPS(J) ↔ UNICODE(UTF-8)コード変換

- C\_CEX\_JU8(JIPS(J)コードの日本語項目を UNICODE(UTF-8)コードへ変換)  
入力データ中の各文字は、JIPS(J) → UNICODE(UTF-8)のコード変換規則に従って変換される。
- C\_CEX\_U8J(UNICODE(UTF-8)の英数字項目を JIPS(J)コードへ変換)  
入力データ中の各文字は、UNICODE(UTF-8) → JIPS(J)のコード変換規則に従って変換される。

## (k) [Standard] JIPS(J) ↔ UNICODE(UTF-16 Little Endian)コード変換

- C\_CEX\_JU16L(JIPS(J)コードの日本語項目を UNICODE(UTF-16 Little Endian)のコードへ変換)  
入力データ中の各文字は、JIPS(J) → UNICODE(UTF-16 Little Endian)のコード変換規則に従って変換される。
- C\_CEX\_U16LJ(UNICODE(UTF-16 Little Endian)コードの日本語項目を JIPS(J)のコードへ変換)  
入力データ中の各文字は、UNICODE(UTF-16 Little Endian) → JIPS(J)のコード変換規則に従って変換される。

(3) データ名 3 には、コード変換システムサブルーチンの実行状態を示す状態キーの内容が格納される。

状態キーは 3 文字から成り、左の文字から順に状態キー 1、状態キー 2、状態キー 3 と呼ぶ。状態キーの内容は、コード変換サブルーチンを実行時、データ名 3 で指定されたデータ項目の文字数によって次のように異なる。

## (a) データ名 3 が 2 文字の場合

状態キー 1 の内容が 0 の時、状態キー 1 および状態キー 2 の内容がデータ名 3 に格納される。

状態キー 1 の内容が 9 の時、標準エラー出力に COB201 のメッセージが表示され、そのプログラムは強制的に打ち切られる(詳細は「付録 C 実行時メッセージ一覧」を参照)。

状態キー "00" は正常終了を表す。

[Standard] C\_CSJ, C\_CJS サブルーチンの場合、状態キー "01" は変換前データに変換できない文字が検出されたことを表す。

## (b) データ名 3 が 3 文字の場合

状態キーの内容がそのままデータ名 3 に格納される。

すなわち、状態キー 1 の内容が 9 の時も、そのまま処理が続行される。

状態キーの種類および意味は次のとおり。

| 状態キー |   |   | 意味    |
|------|---|---|-------|
| 1    | 2 | 3 |       |
| 0    | 0 | △ | 正常終了。 |

|   |   |   |                                                                                                                                                                                                                                                                                                                                                                         |
|---|---|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | 0 | 4 | <p>データ名2の領域に収まらない部分で桁落ちが発生した。</p> <p>この状態は、C-C1EJ, C-C1JE, C-CSJ, C-CJS 以外のコード変換サブルーチンの実行時に発生する。</p>                                                                                                                                                                                                                                                                    |
| 9 | 1 | 2 | <p>入力データ中に変換不能文字が検出された。</p> <p>変換不能文字は、1 バイトコードの時 00(16)に、2 バイトコードの時 0000(16)に変換される。</p> <p>この状態は、C-C1EJ および C-C1JE 以外のコード変換サブルーチンの実行時に発生する。</p> <p>[Standard] C_CEX_SJ, C_CEX_JS, C_CEX_SU8, C_CEX_U8S, C_CEX_SU16L, C_CEX_U16LS, C_CEX_JU8, C_CEX_U8J, C_CEX_JU16L, C_CEX_U16LJ サブルーチンの場合、変換不能文字は、0xFFFD に変換される。なお、変換先が UNICODE の場合は、変換先のコード体系に応じて U+FFFD が設定される。</p> |

(4) [Standard] データ名 1 に日本語文字以外を含む場合、コード変換結果は保証しない。

## 13.2.8 基本多言語面以外の文字検査

### 機能

データ項目中に Unicode の基本多言語面の文字以外が含まれているかを検査し、最初に見つかった文字位置を返す。

### 一般形式

**CALL** "C\_NOBMP" **USING** データ名 1 データ名 2

### 構文規則

- (1) データ名 1, データ名 2 は、ファイル節, 作業場所節, 局所記憶節, 連絡節およびベースデータ節中のデータ項目として定義されていなければならない。また、ファイル節で定義されている場合は修飾してもよい。
- (2) データ名 1 は、英数字基本項目または日本語項目でなければならない。
- (3) データ名 2 は、COMP-2 でなければならない。

### 一般規則

- (1) データ名 1 は、検査対象となるデータ項目を指定する。
- (2) データ名 1 が英数字基本項目として定義されているとき、データ項目に UTF-8 の文字が格納されているものとして、Unicode の基本多言語面の文字以外が含まれているかを検査する。
- (3) データ名 1 が日本語項目として定義されているとき、データ項目にメモリの逆順に配置された UTF-16 の文字が格納されているものとして、Unicode の基本多言語面の文字以外が含まれているかを検査する。
- (4) データ名 1 の先頭から順番に検査し、Unicode の基本多言語面の文字以外が見つかったとき、データ名 2 にデータ名 1 中の該当する文字位置が設定される。データ名 1 中に該当する文字が複数ある場合、データ名 2 には最初に見つかった文字位置が設定される。データ名 1 中に該当する文字が見つからないと、データ名 2 には 0 が設定される。

### 注意

本システムサブルーチンを使用すると、データ項目中に、本システムで対応していない以下の文字が含まれていないか検査することができる。

- ・Unicode で定義されている基本多言語面以外の文字
- ・Unicode で定義されていないコード範囲の文字
- ・サロゲートペアを表現するために使用されるコード範囲の文字 (UTF-16 のみ)

たとえば、日本語項目 1 文字は 2byte だが、サロゲートペアは 1 文字が 4byte で表現されるため、サロゲートペアの文字が含まれる場合、コード変換で意図しない桁落ちが発生する可能性がある。

## 13.2.9 プリンタファイル

[Standard] 本機能は Standard Edition 版および SE 動作モードにおいて、使用できる。

### 13.2.9.1 プリンタファイルのシンボリック制御

#### 機 能

プリンタファイルに対してシンボリックに ESC/CEX シーケンスを出力する。

なお、本システムサブルーチンは「COBOL アプリケーション実行環境 (プリンタ機能)」使用時のみ使用できる。

#### 一般形式

**CALL "CBLSYMPR" USING** ファイル名

#### 規 則

- (1) ファイル名は、EXTERNAL 句の指定のない、既にオープンが成功しているプリンタファイルを指定する。
- (2) 本システムサブルーチンを実行すると、指定されたファイルの 01 レコード領域の内容を ESC/CEX シーケンスとして、指定されたプリンタファイルに出力する。  
ESC/CEX シーケンスについては、『COBOL アプリケーション実行環境説明書』を参照。
- (3) 本システムサブルーチンの実行結果は、FILE STATUS データ項目に通知される。  
FILE STATUS データ項目に通知される値を以下に示す。

| FILE STATUS データ項目 |        | 意 味                                                                                                |
|-------------------|--------|----------------------------------------------------------------------------------------------------|
| 状態キー 1            | 状態キー 2 |                                                                                                    |
| 0                 | 0      | 正常終了                                                                                               |
| 3                 | 0      | プリンタの準備ができていない                                                                                     |
| 4                 | 0      | WID, WD1, WDC の場合<br>ID 書き込み時, ID を正しく書くことができない<br>RID, RD1, RDC の場合<br>ID 読み込み時, ID を正しく読むことができない |
| 4                 | 8      | ファイルがオープンされていないファイルである                                                                             |
| 9                 | 1      | ファイルに EXTERNAL 句の指定がある, または, プリンタファイルでない                                                           |
| 9                 | 7      | プリンタ用紙切れが発生した                                                                                      |

|   |   |                                                                          |
|---|---|--------------------------------------------------------------------------|
| 9 | 9 | シーケンスが未定義, またはパラメータエラーが発生した<br>または「COBOL アプリケーション実行環境 (プリンタ機能)」が使用されていない |
|---|---|--------------------------------------------------------------------------|

**注意**

FILE STATUS 句が指定されていないファイルに対しては, 上記の FILE STATUS は通知されない。また, 実行時エラーメッセージも出力されない。

## 13.2.9.2 プリンタファイルの装置アドレス変更

**機 能**

原始プログラムのファイル管理記述項で記述した, プリンタファイルの SELECT 句の装置指定およびフォーム名を, 実行時に変更する。

**一般形式**

**CALL "CBLASGN" USING** ファイル名 データ名 1 データ名 2

**規 則**

- (1) ファイル名は, SELECT 句の装置指定のデバイスタイプが PRN であり, オープンされていないプリンタファイルを指定する。
- (2) データ名 1, データ名 2 は次の規則に従っていなければならない。
  - (a) データ名 1: 装置指定  
装置指定を定義する 6 桁の英数字項目を定義する。
  - (b) データ名 2: フォーム名指定  
フォーム名を指定する 8 桁の英数字項目を定義する。
- (3) ファイル名で指定したファイルの SELECT 句の装置指定を, データ名 1 に変更する。
- (4) フォーム名をデータ名 2 に変更する。
- (5) データ名 1 は, SELECT 句の装置指定から変更する装置指定のデバイスタイプ(3 桁)と, デバイスアドレス(3 桁)を指定する。

| SELECT 句の装置指定 | 変更できる装置指定 |
|---------------|-----------|
| PRN [nnn]     | PRN [mmm] |

- (6) データ名 2 は, フォーム名を指定する。空白にした場合はフォーム名が指定されないものとみなす。
- (7) 本システムサブルーチンを実行後, そのファイルに対する OPEN 文を実行 (正常終了) したときに装置

指定などが変更される。以後、そのファイルに対して再び本システムサブルーチンを実行し、OPEN 文を実行 (正常終了) するまで有効となる。

ただし、「COBOL アプリケーション実行環境 (プリンタ機能)」を使用しないときに、フォーム名指定を本システムサブルーチンで実行すると、その指定は改ページ毎に有効となる。

## 注意

EXTERNAL 句が指定されているファイルに対しては、装置指定の変更は、CBLASGN を呼び出したプログラムでの OPEN 文のみで、有効となる。

### 13.2.9.3 プリンタファイルの強制クローズ

## 機能

プリンタファイルを強制クローズし、プリンタの解放を行う。

## 一般形式

**CALL "CBLCLSPR" USING ファイル名**

## 規則

- (1) ファイル名は、すでにオープンが成功しているプリンタファイルを指定する。
- (2) 通常 CLOSE 文でエラーが発生した場合、アプリケーションはリカバリのためのエラーステータス (注 1) を返却しプリンタを解放しない。このときエラーを取り除くまでプリンタは解放されず、アプリケーションを終了させなければ他のアプリケーションからの使用 (注 2) もできないままとなる。  
このような場合に、本システムサブルーチンを実行すると、指定されたプリンタファイルを強制的にクローズしてアプリケーションのプロセスを終了せずにプリンタを解放する。

## 注意

- (注 1) プリンタの未準備、用紙切れ、プリンタ異常など。  
(注 2) SHARED-MODE を指定してプリンタの共用を行った場合。

- (3) 本システムサブルーチンの実行結果は、FILE STATUS データ項目に通知される。

FILE STATUS データ項目に通知される値を以下に示す。

| FILE STATUS データ項目 |        | 意 味                    |
|-------------------|--------|------------------------|
| 状態キー 1            | 状態キー 2 |                        |
| 0                 | 0      | 正常終了                   |
| 4                 | 8      | ファイルがオープンされていないファイルである |



|   |   |                  |
|---|---|------------------|
| 9 | 1 | ファイルがプリンタファイルでない |
|---|---|------------------|

## 13.2.9.4 SPOOL 付加情報の設定

## 機 能

直接スプールファイル出力を指定したプリンタファイルに対して、SPOOL 付加情報を設定する。

なお、本システムサブルーチンは「COBOL アプリケーション実行環境 (プリンタ機能)」使用時のみ使用できる。

## 一般形式

**CALL "CBLSP00L" USING ファイル名 データ名**

## 規 則

(1) ファイル名は、直接スプールファイル出力を指定したプリンタファイルを指定する。

(2) データ名は次の規則に従うこと。

データ名: SPOOL 付加情報

SPOOL 付加情報を定義する。

(3) データ名は、以下のように定義する。

01 データ名.

02 データ名 11 PIC 9(2).

02 FILLER PIC X.

02 データ名 12 PIC X.

02 データ名 13 PIC X.

02 データ名 14 PIC X.

02 データ名 15 PIC X(2).

02 FILLER PIC X(2).

02 データ名 16 { PIC X(16).  
PIC N(8). }

02 FILLER PIC X(2).

(a) データ名 11 からデータ名 14 までは、以下のように指定する。

| データ名    | 意 味    | 指定値  | 内 容         |
|---------|--------|------|-------------|
| データ名 11 | コピー部数  | 1~99 | コピー部数       |
| データ名 12 | 印字開始方法 | Y    | 自動的に印字を開始する |
|         |        | N    | 自動的に印字は行わない |

|         |        |   |                   |
|---------|--------|---|-------------------|
| データ名 13 | テスト印字  | Y | 印字開始前にテスト印字を行う    |
|         |        | N | 印字開始前にテスト印字を行わない  |
| データ名 14 | レポート保存 | Y | 印字終了後, レポートを保存する  |
|         |        | N | 印字終了後, レポートは保存しない |

(b) データ名 15 に通知されるステータスは, 以下のとおり。

| データ名 15 に通知される値 | 意 味                                    |
|-----------------|----------------------------------------|
| 00              | 正常終了                                   |
| 90              | データ名 11～データ名 14 の内容が間違っている             |
| 91              | 指定されたファイルが直接スプールファイル出力を指定したプリンタファイルでない |
| 9N              | 内部エラーが発生した                             |

(c) データ名 16 は, 以下のよう指定する。

| データ名    | 意 味  | 指定値 | 内 容                                                                                                                                                                 |
|---------|------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| データ名 16 | コメント | 任意  | SP00L に付加するコメントを格納する<br>英数字項目では 16 桁, 日本語項目では 8 桁定義する<br>この領域が空白のとき, SP00L にコメントは付加されない<br>本システムでは, プログラムの固有文字集合を JIS2004 対応とする翻訳オプション指定時, データ名 16 を日本語項目で定義してはならない |

(4) 本システムサブルーチンを実行すると, そのファイルに対する OPEN 文を実行 (正常終了) したときに付加情報が設定される。以後, そのファイルに対して再び本システムサブルーチンを実行し, OPEN 文を実行 (正常終了) するまで付加情報は変更されない。

## 13.2.9.5 プリンタ印字モードの指定

## 機 能

印刷機能における印字モードを実行時に変更する。

## 一般形式

**CALL "CBLPMODE" USING データ名 1 データ名 2**

## 規 則

(1) データ名 1, データ名 2 は次の規則に従うこと。

(a) データ名 1: 印字モード

1 桁の英数字項目として定義する。

(b) データ名 2: エラーステータス

2 桁の英数字項目として定義する。

(2) データ名 1 は、以下のように指定する。

| 値 | 意 味               |
|---|-------------------|
| A | PRNAMR 経由の印字となる   |
| G | プリントマネージャ経由の印字となる |

(3) データ名 2 には、以下値が設定される。

| 値  | 意 味                                       |
|----|-------------------------------------------|
| 00 | 正常終了した                                    |
| 30 | すでにプリンタモードは設定されている                        |
| 39 | PRNAMR 経由の印字の指定が行われているが、PRNAMR 経由では実行できない |
| 90 | パラメータ不正                                   |

(4) 本システムサブルーチンは、プリンタファイルをオープンする前に呼び出すこと。

## 13.2.10 別プログラムの起動

[Standard] 本機能は Standard Edition 版および SE 動作モードにおいて、使用できる。

### 13.2.10.1 別プログラムの起動 (CBLRUN)

#### 機 能

別プログラムを起動する。

なお、本システムサブルーチンは「COBOL アプリケーション実行環境 (ジョブ管理機能)」使用時のみ使用できる。

#### 一般形式

**CALL "CBLRUN" USING データ名 1 データ名 2**

#### 規 則

- (1) データ名 1、データ名 2 は次の規則に従うこと。
  - (a) データ名 1: 実行形式ファイル / パラメータ  
512 桁以下のデータ項目として定義する。
  - (b) データ名 2: 桁数  
4 桁の符号なし外部 10 進項目として定義する。
- (2) 本システムサブルーチンを実行すると、本システムサブルーチンを実行したプログラムは終了し、次のプログラムを起動する。
- (3) データ名 1 の内容は、次のいずれかの形式でなければならない。  
 なお、name は必要に応じてディレクトリ指定を含み、実行形式ファイル名を意味する。
  - ・パラメータを指定しない場合  

Name
  - ・パラメータを指定する場合  

Name パラメータ 1 パラメータ 2 …
- (4) データ名 2 には、データ名 1 に指定した文字列の桁数を指定する。
- (5) 本システムサブルーチンを実行する前に、プログラム内でオープンしたファイルは、すべてクローズしておかなければならない。
- (6) データ名 2 の内容が不正であった場合、要求されたプログラムは起動されず、要求元プログラムの次の手続き文に戻る。

## 13.2.10.2 別プログラムの起動 (CBLSPAWN)

## 機 能

自プログラムを終了することなく、要求されたプログラムを画面切り離し状態で起動する。

なお、本システムサブルーチンは「COBOL アプリケーション実行環境 (ジョブ管理機能)」使用時のみ使用できる。

## 一般形式

**CALL "CBLSPAWN" USING データ名 1 データ名 2 データ名 3**

## 規 則

(1) データ名 1、データ名 2、データ名 3 は次の規則に従うこと。

(a) データ名 1: 実行形式ファイル／パラメータ

512 桁以下のデータ項目として定義する。

(b) データ名 2: 桁数

4 桁の符号なし外部 10 進項目として定義する。

(c) データ名 3: 実行モード／実行結果

12 桁の英数字項目として定義する。

(2) 本システムサブルーチンを実行したプログラムは、終了することなく要求されたプログラムを起動する。  
起動されたプログラムは画面切り離し状態になる。

(3) データ名 1 の内容は、次のいずれかの形式でなければならない。

なお、name は必要に応じてディレクトリ指定を含み、実行形式ファイル名を意味する。

・パラメータを指定しない場合

|      |
|------|
| Name |
|------|

・パラメータを指定する場合

|      |         |         |     |
|------|---------|---------|-----|
| Name | パラメータ 1 | パラメータ 2 | ... |
|------|---------|---------|-----|

(4) データ名 2 には、データ名 1 に指定した桁数を指定する。

(5) データ名 3 は、以下の形式で定義する。

01 データ名 3.

02 データ名 31 PIC X.

02 データ名 32 PIC X.

02 データ名 33 PIC X(2).

02 FILLER PIC X(8).

## (a) データ名 31

起動元の画面の扱いを指定する。

"F" : 起動したプログラムの画面を起動元の画面の前に出す。

"B" : 起動したプログラムの画面を起動元の画面の後ろに出す。

(COBOL WS エミュレータ使用時のみ)

## (b) データ名 32

起動元のプログラムの実行モードを指定する。

"D" : 起動したプログラムの終了に関係なく進行する。

"W" : 起動したプログラムが終了するまで停止する。

## (c) データ名 33

実行結果が通知される。値と内容は以下のとおり。

| 値  | 意 味                        |
|----|----------------------------|
| 00 | 正常終了した                     |
| 10 | 最大プロセスを超えるプロセスを起動した        |
| 23 | データ名 1 で指定したパス名が見つからない     |
| 90 | 引数のパラメータエラーが発生した           |
| 91 | 起動できない (ジョブ管理が資源を確保できなかった) |

## 13.2.10.3 別プログラムの起動 (CBLCHAINR)

## 機 能

自プログラムを終了させて、新たに要求されたプログラムを起動する。起動されたプログラムの終了後、再び元のプログラムを起動する。

なお、本システムサブルーチンは「COBOL アプリケーション実行環境 (ジョブ管理機能)」使用時のみ使用できる。

## 一般形式

**CALL "CBLCHAINR" USING データ名 1 データ名 2 データ名 3**

## 規 則

(1) データ名 1, データ名 2, データ名 3 は次の規則に従うこと。

(a) データ名 1: 実行形式ファイル / パラメータ

512 桁以下のデータ項目として定義する。

(b) データ名 2: 桁数

4 桁の符号なし外部 10 進項目として定義する。

(c) データ名 3: 実行モード／実行結果

10 桁の英数字項目として定義する。

(2) 本システムサブルーチンを実行したプログラムは終了し、新たなプログラムが起動される。起動されたプログラムが終了した後、本システムサブルーチンを実行したプログラムが再び起動される。

(3) データ名 1 の内容は、次のいずれかの形式でなければならない。

なお、name は必要に応じてディレクトリ指定を含み、実行形式ファイル名を意味する。

・パラメータを指定しない場合

|      |
|------|
| Name |
|------|

・パラメータを指定する場合

|      |         |         |     |
|------|---------|---------|-----|
| Name | パラメータ 1 | パラメータ 2 | ... |
|------|---------|---------|-----|

(4) データ名 2 には、データ名 1 に指定した桁数を指定する。

(5) データ名 3 は、以下の形式で定義する。

01 データ名 3.

02 データ名 31 PIC X(2).

02 FILLER PIC X(8).

(a) データ名 31

実行結果が通知される。値と内容は以下のとおり。

| 値  | 意 味                        |
|----|----------------------------|
| 00 | 正常終了した                     |
| 10 | 最大プロセスを超えるプロセスを起動した        |
| 23 | データ名 1 で指定したパス名が見つからない     |
| 90 | 引数のパラメータエラーが発生した           |
| 91 | 起動できない (ジョブ管理が資源を確保できなかった) |

(6) 本システムサブルーチンを実行する前に、プログラム内でオープンしたファイルは、すべてクローズしておかなければならない。

#### 13.2.10.4 別プログラムの起動 (CBLCHAINC)

#### 機 能

自プログラムを終了させて、起動されたプログラムに制御が渡され、起動されたプログラムの終了後、元のプログラムの次の手続き文に戻る。

なお、本システムサブルーチンは「COBOL アプリケーション実行環境 Server」使用時のみ、COBOL WS エミューレータ配下で使用できる。「COBOL アプリケーション実行環境 Client」では使用できない。

## 一般形式

**CALL "CBLCHAINC" USING データ名 1 データ名 2 データ名 3**

## 規 則

- (1) データ名 1, データ名 2, データ名 3 は次の規則に従うこと。
  - (a) データ名 1: 実行形式ファイル／パラメータ  
512 桁以下のデータ項目として定義する。
  - (b) データ名 2: 桁数  
4 桁の符号なし外部 10 進項目として定義する。
  - (c) データ名 3: 実行モード／実行結果  
10 桁の英数字項目として定義する。
- (2) 本システムサブルーチンを実行したプログラムは終了し、新たなプログラムが起動される。起動されたプログラムが終了した後、本システムサブルーチンを実行した次の手続き文から処理が開始される。
- (3) データ名 1 の内容は、次のいずれかの形式でなければならない。

なお、name は必要に応じてディレクトリ指定を含み、実行形式ファイル名を意味する。

- ・パラメータを指定しない場合

|      |
|------|
| Name |
|------|

- ・パラメータを指定する場合

|      |         |         |     |
|------|---------|---------|-----|
| Name | パラメータ 1 | パラメータ 2 | ... |
|------|---------|---------|-----|

- (4) データ名 2 には、データ名 1 に指定した桁数を指定する。
- (5) データ名 3 は、以下の形式で定義する。

01 データ名 3.

02 データ名 31 PIC X(2).

02 FILLER PIC X(8).

(a) データ名 31

実行結果が通知される。値と内容は以下のとおり。

| 値  | 意 味                        |
|----|----------------------------|
| 00 | 正常終了した                     |
| 10 | 最大プロセスを超えるプロセスを起動した        |
| 23 | データ名 1 で指定したパス名が見つからない     |
| 90 | 引数のパラメータエラーが発生した           |
| 91 | 起動できない (ジョブ管理が資源を確保できなかった) |



### 13.2.11 ファイル入出力用システムサブルーチン

[Enterprise] 本機能は Enterprise Edition 版において、使用できる。ファイル入出力用システムサブルーチンには以下の表 13-1 に示す 4 種類のサブルーチンがある。利用者は、これらのサブルーチンを CALL 文を用いて呼び出すことができる。

表 13-1 ファイル入出力システムサブルーチン一覧

| システムサブルーチン名  | 機能                                  |
|--------------|-------------------------------------|
| A_CBL_START  | RAF 順ファイルにおいて、特定のレコードに位置付ける。        |
| A_CBL_NOTE   | RAF 順ファイルにおいて、現在確立されているレコードアドレスを得る。 |
| A_CBL_CLEAN  | バッファ内に残っている更新データを強制的にディスクに書き出す。     |
| A_CBL_STARTB | RAF 順ファイルにおいて、第 1 レコードに位置付ける。       |

#### 13.2.11.1 A\_CBL\_START

##### 機 能

RAF 順ファイルにおいて、特定レコードに位置付ける手段を与える。

##### 一般形式

**CALL "A\_CBL\_START"**

**USING** ファイル名 1 一意名 1 [ 一意名 2 ]

##### 規 則

- (1) CALL 文を実行する前に、そのファイルを入力または入出力両用モードの OPEN 文で開いておかなければならない。
- (2) ファイルは RAF 順ファイルとして定義されていなければならない。
- (3) 一意名 1 は 8 バイト以上の書き込み可能なエリアとして定義されていて、本システムサブルーチンを実行する前に、2 進形式(メモリ逆順の 4 バイトの符号なし 2 進数)+2 進形式(メモリ逆順の 4 バイトの符号なし 2 進数)のレコードアドレスを左詰めで格納していなければならない。

レコードアドレスの形式として

BLKDD

LREC

が利用できるが、以下のことには注意されたい。

(a) ファイルの媒体やレコード形式により使用可能なアドレス形式に制限がある。

(b) COBOL の既定値はレコード形式により異なる。

固定長レコード（レコード形式=F または FB）のときは、LREC 形式。

可変長レコード（レコード形式=V または VB）のときは、BLKDD 形式。

- (4) 一意名 2 は 2 バイト以上の書き込み可能なエリアとして定義されていなければならない。このエリアには、本システムサブルーチンの実行状態を示す値が 2 進形式 (COMP-1) で設定される。設定される値は表 13-2 を参照。
- (5) 一意名 2 が省略された場合、本システムサブルーチンの実行でエラー状態が検出されると実行時エラーメッセージが表示され、そのタスクは異常終了 (リターンコードは表 13-2 で示される) する。
- (6) 本システムサブルーチンの直後に実行された READ 文は、本システムサブルーチンで位置付けられたレコードを読み込む。

#### 13.2.11.2 A\_CBL\_NOTE

### 機 能

RAF 順ファイルにおいて、現在確立されているレコードアドレスをプログラムに知らせる手段を与える。

### 一般形式

**CALL "A\_CBL\_NOTE"**

**USING** ファイル名 1 一意名 1 [ 一意名 2 ]

### 規 則

- (1) CALL 文を実行する前に、そのファイルを OPEN 文で開いておかなければならない。
- (2) ファイルは RAF 順ファイルとして定義されていなければならない。
- (3) 一意名 1 には現在確立されているレコードアドレス (直前の入出力文で処理されたレコードのアドレス) が左詰めで格納される。したがって、一意名 1 のエリアは、使用しているレコードアドレスを 2 進形式 (メモリ逆順の 4 バイトの符号なし 2 進数) + 2 進形式 (メモリ逆順の 4 バイトの符号なし 2 進数) で格納できる大きさでなければならない。

レコードアドレスの形式として

BLKDD

**LREC**

が利用できるが、以下のことに注意されたい。

(a) ファイルの媒体やレコード形式により使用可能なアドレス形式に制限がある。

(b) COBOL の既定値はレコード形式により異なる。

固定長レコード（レコード形式=F または FB）のときは、LREC 形式。

可変長レコード（レコード形式=V または VB）のときは、BLKDD 形式。

(4) 一意名 2 は 2 バイト以上の書き込み可能なエリアとして定義されていなければならない。このエリアには、本システムサブルーチンの実行状態を示す値が 2 進形式 (COMP-1) で設定される。設定される値は表 13-2 を参照。

(5) 一意名 2 が省略された場合、本システムサブルーチンの実行でエラー状態が検出されると実行時エラーメッセージが表示され、そのタスクは異常終了 (リターンコードは表 13-2 で示される) する。

(6) ファイルの OPEN 直後に、本システムサブルーチンを実行してはならない。

表 13-2 A\_CBL\_START と A\_CBL\_NOTE のリターンコード

| 一意名-2 の値 | リターンコード  | 意味                           |
|----------|----------|------------------------------|
| 0        | DONE     | 正常終了。                        |
| 1792     | NOWAIT   | 他のプロセスと同時にアクセスすることが認められていない。 |
| 2052     | ADDROUT  | レコードアドレスがファイル外を指している。        |
| 2565     | FUNCNAV  | RAF 順ファイル以外である。              |
| 6177     | FLILERR  | ファイル識別リテラルが正しいファイル識別リテラルでない。 |
| 6179     | ADFTERR  | アドレス形式が正しくない。                |
| 7169     | NOTOPEN  | ファイルが OPEN されていない。           |
| 7172     | NOCURREC | 現在位置づけられているレコードがない。          |
| 7173     | WRONGPMD | OPEN 文のモードが正しくない。            |

## 13.2.11.3 A\_CBL\_CLEAN

**機 能**

バッファ内に残っている更新データを強制的に RAF 順ファイルに書き込み、更新処理と RAF 順ファイルの同期をとる。

さらに以後同一ブロックの読み込み要求がきた場合、バッファ内にあるブロックを使用せず、新たに RAF 順ファイルよりブロックの読み込みを行う。

## 一般形式

**CALL "A\_CBL\_CLEAN"****USING** ファイル名 1 [ 一意名 1 ]

## 規 則

- (1) ファイルは RAF 順ファイルとして定義されていなければならない。
- (2) CALL 文を実行する前に、そのファイルを OPEN 文で開いておかなければならない。
- (3) 一意名 1 は 2 バイト以上の書き込み可能なエリアとして定義されていなければならない。このエリアには、本システムサブルーチンの実行状態を示す値が 2 進形式 (COMP-1) で設定される。設定される値は表 13-3 を参照。
- (4) 一意名 1 が省略された場合、本システムサブルーチンの実行でエラー状態が検出されると実行時エラーメッセージが表示され、そのタスクは異常終了 (リターンコードは表 13-3 で示される) する。

表 13-3 A\_CBL\_CLEAN のリターンコード

| 一意名 1 の値 | リターンコード  | 意味                           |
|----------|----------|------------------------------|
| 0        | DONE     | 正常終了。                        |
| 512      | BUSY     | 他のチャンネルにより装置が使用されている。        |
| 1153     | IOFAIL   | 入出力エラーが発生した。                 |
| 1159     | ABNCCAD  | 判別不可能な入出力終了コードが戻された。         |
| 1160     | SRCHFAIL | 入出力実行中レコードの位置づけに失敗した。        |
| 1792     | NOWAIT   | 他のプロセスと同時にアクセスすることが認められていない。 |
| 2562     | DVNAV    | 装置が使用不能である。                  |
| 2565     | FUNCNAV  | RAF 順ファイル以外である。              |
| 6177     | FLILERR  | ファイル識別リテラルが正しいファイル識別リテラルでない。 |
| 7169     | NOTOPEN  | ファイルが OPEN されていない。           |
| 7682     | DATAERR  | データの入出力が正常に終了しなかった。          |
| 7685     | UNRECIO  | 回復不可能な入出力エラーが発生した。           |
| 7686     | STANDBY  | 装置が STANDBY 状態である。           |
| 7687     | OFFLINE  | 装置がオフライン状態である。               |

## 13.2.11.4 A\_CBL\_STARTB

## 機 能

RAF 順ファイルにおいて、第 1 レコードに位置付ける手段を与える。

## 一般形式

**CALL "A\_CBL\_STARTB"**

**USING** ファイル名 1 [ 一意名 1 ]

## 規 則

- (1) CALL 文を実行する前に、そのファイルを入力または入出力両用モードの OPEN 文で開いておかなければならない。
- (2) ファイルは RAF 順ファイルとして定義されていなければならない。
- (3) 一意名 1 は 2 バイト以上の書き込み可能なエリアとして定義されていなければならない。このエリアには、本システムサブルーチンの実行状態を示す値が 2 進形式 (COMP-1) で設定される。設定される値は表 13-4 を参照。
- (4) 一意名 1 が省略された場合、本システムサブルーチンの実行でエラー状態が検出されると実行時エラーメッセージが表示され、そのタスクは異常終了 (リターンコードは表 13-4 で示される) する。
- (5) 本システムサブルーチンの直後に実行された書き方 1 の READ 文は、ファイルの第 1 レコードを読み込む。  
ファイルの第 1 レコードが削除レコードなら、最初の有効レコードが読み込まれる。
- (6) 本システムサブルーチンの実行が不成功に終わった時は、現在のレコード指示子の指している位置は不定となる。

表 13-4 A\_CBL\_STARTB のリターンコード

| 一意名-1 の値 | リターンコード | 意味                        |
|----------|---------|---------------------------|
| 0        | DONE    | 正常終了。                     |
| 22       | DUMMY   | ダミーファイルが割り当てられている。        |
| 512      | BUSSY   | 他のチャンネルにより装置が使用されている。     |
| 1079     | RECNOFD | 指定されたレコードが存在しない。          |
| 1153     | IOFAIL  | 入出力エラーが発生したにもかかわらず、入出力要求が |

|      |          |                              |
|------|----------|------------------------------|
|      |          | 出された。                        |
| 1159 | ABNCCAD  | 判別不可能な入出力終了コードが戻された。         |
| 1160 | SRCHFAIL | 入出力実行中レコードの位置付けに失敗した。        |
| 1792 | NOWAIT   | 他のプロセスと同時にアクセスすることが認められていない。 |
| 2562 | DVNAV    | 装置が使用不能である。                  |
| 2565 | FUNCNAV  | RAF 順ファイル以外である。              |
| 7169 | NOTOPEN  | ファイルが OPEN されていない。           |
| 7173 | WRONGPMD | OPEN 文のモードが正しくない。            |
| 7682 | DATAERR  | データの入出力が正常に終了しなかった。          |
| 7685 | UNRECIO  | 回復不可能な入出力エラーが発生した。           |
| 7686 | STANDBY  | 装置が STANDBY 状態である。           |
| 7687 | OFFLINE  | 装置がオフライン状態である。               |

## 13.2.12 内部ファイル ID 置換表の解放

[Standard] 本機能は Standard Edition 版において、使用できる。SE 動作モードでは使用できない。

### 機能

メモリ上に配置した「内部ファイル ID 置換表」を任意のタイミングで解放し、再び読み込み可能な状態にする。

### 一般形式

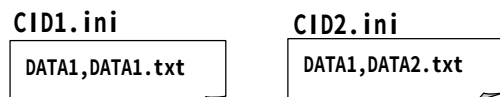
**CALL "CS\_CIDTBLCLEAR"**

### 規 則

- (1) 本システムサブルーチンを実行すると、次のファイルのオープン時、環境変数 COB\_CID\_INIFILE に指定された内部ファイル ID 置換表を読み込む。

以降にサンプルコードを示す。

### 内部ファイル ID 置換表



### COBOL プログラム

```

IDENTIFICATION DIVISION.
PROGRAM-ID. CID01.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

 SELECT OUT1-FILE ASSIGN TO DF-MSD
 ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.
FILE SECTION.

FD OUT1-FILE
 VALUE OF IDENTIFICATION "DATA1".

01 DATA-REC PIC X(80).
WORKING-STORAGE SECTION.
01 PUTENV_DAT1 PIC X(256).

```

```

01 PUTENV_DAT2 PIC X(256).
PROCEDURE DIVISION.
HAJIME.

*** 1つ目の内部ファイル置換表 ***
MOVE "COB_CID_INIFILE" TO PUTENV_DAT1.
MOVE "C:¥Users¥cbltest¥CID1.ini" TO PUTENV_DAT2.
CALL "B_PUTENV" USING PUTENV_DAT1 PUTENV_DAT2.
MOVE ALL "A" TO DATA-REC.
PERFORM WRITE-RTN.

*** システムサブルーチン呼び出し ***
CALL "CS_CIDTBLCLEAR".

*** 2つ目の内部ファイル置換表 ***
MOVE "COB_CID_INIFILE" TO PUTENV_DAT1.
MOVE "C:¥Users¥cbltest¥CID2.ini" TO PUTENV_DAT2.
CALL "B_PUTENV" USING PUTENV_DAT1 PUTENV_DAT2.
MOVE ALL "B" TO DATA-REC.
PERFORM WRITE-RTN.
STOP RUN.

WRITE-RTN.
OPEN OUTPUT OUT1-FILE.
WRITE DATA-REC.
CLOSE OUT1-FILE.
END PROGRAM CID01.

```



## 13.2.13 ファイル入出力状態の永続誤り解除

### 機能

大容量記憶ファイルの永続誤りのファイル入出力状態値をクリアする。

### 一般形式

**CALL "B\_CLEARERR" USING** ファイル名

### 規 則

- (1) ファイル入出力状態の永続誤り解除対象のファイルは次である。
  - (a) COBOL 独自 (順／行順／相対), ISAM (索引)
  - (b) [Standard] IFAS FILE (順／相対／索引), COBOL DB アクセス (索引)  
 ただし, IFAS FILE (順／相対／索引), COBOL DB アクセス (索引) は SE 動作モードでは使用できない。

### 注意事項

- (1) ファイル入出力状態の永続誤りをクリアした後, 同一ファイルに対する入出力処理を続行する場合, エラーリカバリー処理を実行せずに, 永続誤りの原因となったファイル入出力文を再び実行してはならない。

### 使用例

```

COBOL 入出力文を実行
IF FILE1-STS = "30"
 CALL "B_CLEARERR" USING FILE1
 エラーリカバリー処理を実行
 OPEN OUTPUT FILE1
 IF FILE1-STS NOT = "00"
 DISPLAY "ファイルのオープンに失敗しました。"
 DISPLAY "FILE1-STS=" FILE1-STS
 END-IF
ELSE
 IF FILE1-STS NOT = "00"
 DISPLAY "ファイルのオープンに失敗しました。"
 DISPLAY "FILE1-STS=" FILE1-STS
 END-IF
END-IF.
```

## 第14章 プログラミング技法

### 14.1 概説

COBOL プログラムの目的は最少量の主記憶を用い、最も短い時間で実行できるような COBOL プログラムを作成することである。本章では、その目的にあうようにするためにはどんなところに気をつけたらよいかを説明する。ただし、COBOL プログラムのそれぞれの実行環境によって達成される最適化が制限または緩和されるかもしれない。

たとえば

- (1) 上記 2 つの目的を常に達成できるとはかぎらない。すなわち、通常主記憶を増せば実行時間は早くなり、逆に主記憶を少なくしようとすれば実行時間が遅くなる場合が多い。
- (2) ハードウェアの機器構成やシステムの負荷の度合によって、プログラムの最適化が制限または緩和されることがある。
- (3) プログラムの最適化よりも、プログラムの標準化、文書化、わかりやすさなど優先しなければならない場合がある。
- (4) プログラムの最適化がコード体系やシステム設計の変更にまでおよび、それを達成するとなると大幅な改造が必要となる場合がある。

本章で述べるプログラミング技法は、効率のよい COBOL プログラムを書く際の手助けとなることを意図したものであるが、上述したようにそのプログラムの実行環境にあわせて使用する必要がある。

### 14.2 実行用プログラムの効率化

実行用プログラムの効率をあげるための技法を、次の項目に分類して記述する。

- (1) ファイルの入出力処理を早くするための技法
- (2) プログラムが占有する主記憶を少なくするための技法
- (3) データの処理を早くするための技法

### 14.2.1 ファイルの入出力処理を早くするための技法

入出力処理時間を早くするための技法には、次のようなものがある。

(1) ファイルの大きさ(レコードの大きさ)を小さくする

レコード中のデータは内部 10 進数(COMP-3)項目や 2 進数(BINARY, COMP-1, COMP-2, COMP-5)項目を用いることにより、小さくする。ただし、この方法を用いる場合は次のことに注意しなければならない。すなわち、内部表現形式が異なる項目間の移送や演算は処理効率低下の原因になることがある。

通常、この方法はデータがいくつかのプログラムで加工されて最終結果を得られるような場合、最初のプログラムの入力形式は外部 10 進数(DISPLAY)項目を用い、その出力として COMP-3 項目、BINARY 項目や COMP-1, COMP-2, COMP-5 項目を用い、中間のプログラムはすべてその形式で使用し、最後のプログラムでまた外部 10 進数項目に戻すような場合に使用する。

### 14.2.2 プログラムが占有する主記憶を少なくするための技法

プログラムの主記憶を少なくするための技法には、次のようなものがある。また、ここでは特に述べていないが、データの処理を早くするための技法のうちの多くはプログラムの主記憶の縮小にもつながる。これについては、「14.2.3 データの処理を早くするための技法」を参照。

(1) REDEFINES 句を使用する

同時に使用しない作業領域は、できるだけ REDEFINES 句を用いて再定義し、同一の領域を共有する。詳細については、「5.2.2 REDEFINES 句」を参照。

(2) 共通の手続きのサブルーチン化

プログラム中に共通の手続きがある場合は、それをできるだけサブルーチン化し、PERFORM 文でそのサブルーチンを使用するようにすれば、手続き部の主記憶を少なくすることができる。ただし、この方法は共通化されるサブルーチンが小さい場合は、逆に主記憶の増大および実行効率の低下を招くので注意されたい。

(3) BINARY, COMP-1, COMP-2, COMP-5 項目を使用する

外部 10 進数項目を使用するよりも、BINARY, COMP-1, COMP-2, COMP-5 項目を使用した方がデータ部の主記憶を少なくすることができる。ただし、内部表現形式が異なるデータ項目間の移送、演算などは実行効率の低下を招くので注意されたい。

(4) -Ci オプションを使用する

アプリケーションを構成するすべてのプログラムが、INITIAL 句および CANCEL 文を使用しない場合、COBOL プログラムの翻訳時に -Ci オプションを指定する。これにより、INITIAL 句および CANCEL 文のためにコンパイラが確保しているデータバックアップ領域の確保を行なわないため、データ部の主記憶を少なくすることができる。

INITIAL 句および CANCEL 文を使用するアプリケーションに対して -Ci オプションの指定を行なった場合の動作は保証できないため注意されたい。

### 14.2.3 データの処理を早くするための技法

実行時のデータ処理の効率をあげるためには、次に示すような技法がある。

#### (1) INITIALIZE 文について

集団項目に対する INITIALIZE 文は、データ項目の属性など意識することなく、初期化処理が行え、便利な文である。しかし、裏を返せば、集団項目のすべてのデータを初期化するため、集団項目が大きかったり、表定義を含んでいると、1 つの文を実行するのに多大な時間を要する。安易に INITIALIZE 文を使用するのは避け、必要最小限の項目のみを MOVE 文を使用して初期化することが望ましい。

#### (2) ループカウンタなど内部的に使用する数字項目は 2 進数 (BINARY, COMP-1, COMP-2, COMP-5) 項目を使用した方が実行時間を短縮できる。また、2 進数項目の場合は SYNC 句を指定してけたづめを行うことが望ましい。

#### (3) 表操作における効率向上

(a) 添字の値が一定の場合は、添字に数字定数を用いる。数字定数を用いると、実際の番地計算は翻訳時に行われるので実行時間が短縮される。これは主記憶の減少にもつながる。

(b) 添字付けより指標付けの方が実行時間が短縮される。また、主記憶も少なくて済む。しかし、指標付けは SET 文が必要で、添字に対する ADD/MOVE 文より効果が悪い。したがって、表の要素の参照が多いときには指標付けが効率よく、値の設定が多いときには、添字付けの方がむしろ効率がよい。

(c) 必要な添字／指標を減らすため、表の一要素はなるべくまとめた方がよい。

|   |    |    |        |    |     |   |    |    |        |     |     |
|---|----|----|--------|----|-----|---|----|----|--------|-----|-----|
| 例 | 02 | A1 | OCCURS | 10 | ... | } | 02 | A  | OCCURS | 10  | ... |
|   | 02 | A2 | OCCURS | 10 | ... |   |    | 03 | A1     | ... |     |
|   |    |    |        |    |     |   |    | A2 | ...    |     |     |

こうすることにより、A1 と A2 はともに同じ添字／指標で参照でき、たとえば SET, SET UP 文で節約できる。

(d) 表を表引きする場合、表全体を表引きする必要のないときは表の必要部分の範囲のみを表引きすることにより実行時間が短縮される。表引き範囲の変更は、表引き用指標および OCCURS 句の DEPENDING データ名に表の要素の出現番号を設定することにより行われる。ただし SEARCH ALL 文では、表の範囲の上限のみを指定できる。表操作の詳細については、「第 6 章 表操作機能」を参照。

(4) 英字、英数字項目の移送および比較では、両作用対象の桁数を等しくした方が効率がよい。

(5) データ項目の大きさは、実行効率と主記憶との観点から、必要最小限の大きさにしなければならない。

ただし、プログラムの拡張性やデータ件数の増大の可能性なども考慮して決めなければならない。

- (6) 文はできるだけ単純な文を使用した方がよい。たとえば、複雑な数式や複数の答えがある算術文を使用すると、演算のために中間領域が必要になり、実行速度は低下し、主記憶は増大する。よって、文の使いやすさと効率の低下の両観点から考慮し決めなければならない。
- (7) 連絡節(LINKAGE SECTION)内の 01, 77 項目の数を少なくするようにする。

例 LINKAGE SECTION.

|           |   |           |
|-----------|---|-----------|
| 77 A1 ... | } | 01 A.     |
| 77 A2 ... |   | 02 A1 ... |
|           |   | 02 A2 ... |

連絡節の 01, 77 項目の数を減らすことは、CALL 文の USING 句に指定する作用対象の数を減らすことにもなり、CALL 文の効率向上にもつながる。

また、連絡節のデータは頻繁に参照しないほうがよい。

#### (8) IF 文について

##### (a) IF 文において

IF 条件 GO TO T ELSE GO TO F.

は効率上好ましくない。なるべく、

IF 条件 真のときの処理 ELSE 偽のときの処理

のように書いた方がよい。

- (b) IF 文による判定がいくつか続くとき、真になる確率の大きい方から判定すべきである。またそのとき、無駄な判定を繰り返すことは避ける。

例 IF A=1 MOVE X1 TO Y.

IF A=2 MOVE X2 TO Y.

↓

IF A=1 MOVE X1 TO Y GO TO OWARI.

IF A=2 MOVE X2 TO Y GO TO OWARI.

:

または

IF A=1 MOVE X1 TO Y

ELSE IF A=2 MOVE X2 TO Y.

- (c) 判定が A=1, A=2, A=3, ...などと規則的なときは、

GO TO ... DEPENDING ON A

の使用も考える。約 5 個以上の判定なら

GO TO ... DEPENDING の方が効率がよい。

- (9) ACCEPT~FROM DATE/ DAY/ TIME 文は実行に若干の時間を要するので、必要最小限の実行回数にとどめることが望ましい。

また, OPEN 文, CLOSE 文についても同じことが言える。

- (10) 小数部付きの 2 進数 (BINARY/COMP-5) 項目を使用する場合は, 小数部桁数を一致させることで実行時間を短縮できる。

## 第15章 翻訳リスト

### 15.1 概説

本 COBOL は、原始プログラムの翻訳時、次のような翻訳リストを出力する。

- (1) パラメータリスト
- (2) 原始プログラムリスト
- (3) クロスリファレンスリスト
- (4) 登録集参照リスト
- (5) 診断メッセージリスト
- (6) データマップリスト
- (7) [Enterprise for Linux] 用語の展開リスト
- (8) [Enterprise for Linux] 用語名リスト
- (9) [Enterprise for Linux] 用語値名参照リスト

これらのリストは翻訳時のオプションの指定によって出力することができる。

| 翻訳リスト                               | オプション                                                   |
|-------------------------------------|---------------------------------------------------------|
| パラメータリスト                            | -Hp                                                     |
| 原始プログラムリスト                          | -Hs, -Hsc (COPY 展開処理を行ったソースを出力)<br>-Hi (診断メッセージを埋め込み出力) |
| クロスリファレンスリスト                        | -Hx                                                     |
| 登録集参照リスト                            | -Hx                                                     |
| 診断メッセージリスト                          | 診断メッセージがあれば必ず出力される                                      |
| データマップリスト                           | -Hd                                                     |
| [Enterprise for Linux]<br>用語の展開リスト  | -Hf (原始プログラムリストに用語データ項目を展開)                             |
| [Enterprise for Linux]<br>用語名リスト    | -Hm                                                     |
| [Enterprise for Linux]<br>用語値名参照リスト | -He                                                     |

[Enterprise] 診断メッセージのみを出力する場合には、標準出力に出力する。診断メッセージ以外のリストを出力する場合には、ソースファイル名の拡張子 “.cob”, “.cbl” または “.cbf” を除いたファイル

名の最後に **”.list ”**を付与したファイルに、診断メッセージとともに出力する。

ただし、診断メッセージを埋め込み出力のオプションがある場合、原始プログラムリストに診断メッセージを埋め込み出力して、診断メッセージリストは出力しない。

[Standard] 診断メッセージのみを出力する場合には、標準出力に出力する。診断メッセージ以外のリストを出力する場合には、ソースファイル名の拡張子 **”.cob ”**、**”.cbl ”**、**”.cbf ”**または **”.cbx ”**を除いたファイル名の最後に **”.list ”**を付与したファイルに、診断メッセージとともに出力する。

ただし、診断メッセージを埋め込み出力のオプションがある場合、原始プログラムリストに診断メッセージを埋め込み出力して、診断メッセージリストは出力しない。



## 15.2 パラメータリスト

パラメータリストは、翻訳時のオプションパラメータの情報を含んでいる。

| 項目   | 出力項目              | 内容                                                          |
|------|-------------------|-------------------------------------------------------------|
| (1)  | 製品名とバージョン情報       | 製品名とバージョン情報を出力する。                                           |
| (2)  | 原始プログラム名          | 原始プログラム名と作成日付を出力する。                                         |
| (3)  | 翻訳日付              | 原始プログラムが翻訳された日付を出力する。                                       |
| (4)  | 実行可能プログラムファイル名    | 実行可能プログラムファイル名を出力する。<br>翻訳のみを行う場合 (-c オプションと同期)、この項目は出力しない。 |
| (5)  | 登録集原文を検索するディレクトリ名 | 登録集原文を検索するディレクトリ名を出力する。                                     |
| (6)  | OS 名              | OS 情報を出力する。                                                 |
| (7)  | ユーザ名              | ユーザ名を出力する。                                                  |
| (8)  | ドメイン名             | [Enterprise for Windows] [Standard]<br>ドメイン名を出力する。          |
| (9)  | オプション             | 指定されたオプションを出力する。                                            |
| (10) | グループ名             | [Enterprise for Linux]<br>グループ名を出力する。                       |
| (11) | ホスト名              | [Enterprise for Linux]<br>ホスト名を出力する。                        |

**【Enterprise for Windows】【Standard】**  
cobol compiler parameter list

```
(1) compiler version : NEC COBOL Ver1.0
(2) source file : SAMPLE.COB (Mon May 07 11:43:44 2012)
(3) compile date and time : Mon May 07 11:50:45 2012
(4) executable file : SAMPLE.EXE
(5) directories for copy : C:¥home¥cobol¥business¥copy
(6) operating system : Microsoft Windows Server 2008 R2 (x64) Service Pack 1
(7) user name : business
(8) domain name : cobol
(9) compile options : -Hdps -M
```

**【Enterprise for Linux】**  
cobol compiler parameter list

```
(1) compiler version : NEC COBOL Ver1.0
(2) source file : SAMPLE.cob (Thu Jan 31 13:07:38 2013)
```

```

(3) compile date and time : Fri Feb 1 14:24:17 2013
(4) executable file : a.out
(5) directories for copy : ./
(6) operating system : Linux 2.6.32-220.el6.x86_64
(7) user name : business
(10) group name : cobol
(11) host name : server.nec.co.jp
(9) compile options : -Hdps -M

```

### 15.3 原始プログラムリスト

原始プログラムリストは、原始プログラムそのもののリストである。

| 項目  | 出力項目           | 内容                                                                                                                                                                                                                                                                              |
|-----|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | 原文識別           | ソース原文のとき、空白を出力する。<br>登録集原文のとき、"L"を出力する。<br>デバッグ行または選択行が無効のとき、"D"を出力する。                                                                                                                                                                                                          |
| (2) | 内部行番号          | コンパイラが自動的に付与する行番号。                                                                                                                                                                                                                                                              |
| (3) | 原始プログラムのネストレベル | 原始プログラムのネストレベルを出力する。                                                                                                                                                                                                                                                            |
| (4) | 外部行番号          | 【Enterprise】原文のファイル拡張子が".cob"または".cbl"のとき、一連番号領域の行番号を出力する。<br>【Standard】原文のファイル拡張子が".cob" ".cbl"または".cbx"のとき、一連番号領域の行番号を出力する。<br>原文のファイル拡張子が".cbf"のとき、空白を出力する。                                                                                                                  |
| (5) | 標識             | 【Enterprise】原文のファイル拡張子が".cob"または".cbl"のとき、標識領域の固定標識文字を出力する。<br>【Standard】原文のファイル拡張子が".cob" ".cbl"または".cbx"のとき、標識領域の固定標識文字を出力する。<br>原文のファイル拡張子が".cbf"のとき、空白を出力する。                                                                                                                |
| (6) | 原始プログラムイメージ    | 【Enterprise】原文のファイル拡張子が".cob"または".cbl"のとき、A 領域および B 領域の原文を出力する。<br>【Standard】原文のファイル拡張子が".cob" ".cbl"または".cbx"のとき、A 領域および B 領域の原文を出力する。<br>原文のファイル拡張子が".cbf"のとき、プログラム原文領域の原文を出力する。<br>本項目は英数字 80 文字である。これを超える場合は、ソース原文の正書法における行の継続に関する規則に従って折り返し出力する。<br>継続行には固定標識または浮動標識を出力する。 |

|     |          |                                                                                                                                                     |
|-----|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|     |          | 診断メッセージを埋め込み出力のオプションがある場合、<br>本項目に診断メッセージを埋め込む。                                                                                                     |
| (7) | 識別       | [Enteprirse] 原文のファイル拡張子が".cbl"のとき、識別領域の文字を出力する。<br>[Standard] 原文のファイル拡張子が".cbl"または".cbx"のとき、識別領域の文字を出力する。<br>原文のファイル拡張子が".cob"または".cbf"のとき、空白を出力する。 |
| (8) | 登録集原文の名前 | 登録集原文名を出力する。                                                                                                                                        |

&lt;固定形式正書法（診断メッセージなしまたは診断メッセージを埋め込み出力しない）の場合&gt;

| (1) | (2)  | (3) | (4)    | (5) | (6)                                              | (7) | (8)       |
|-----|------|-----|--------|-----|--------------------------------------------------|-----|-----------|
|     | LINE | PN  | SEQ-NO |     | ***** SOURCE PROGRAM LIST *****                  |     | TEXT.NAME |
|     | 1    | 1   | 000010 |     | IDENTIFICATION DIVISION.                         |     |           |
|     | 2    | 1   | 000020 |     | PROGRAM-ID. SAMPLE.                              |     |           |
|     | 3    | 1   | 000030 |     | ENVIRONMENT DIVISION.                            |     |           |
|     | 4    | 1   | 000040 |     | CONFIGURATION SECTION.                           |     |           |
|     | 5    | 1   | 000050 |     | SOURCE-COMPUTER. PC.                             |     |           |
|     | 6    | 1   | 000060 |     | OBJECT-COMPUTER. PC.                             |     |           |
|     | 7    | 1   | 000070 |     | INPUT-OUTPUT SECTION.                            |     |           |
|     | 8    | 1   | 000080 |     | FILE-CONTROL.                                    |     |           |
|     | 9    | 1   | 000090 |     | COPY TMSEL.                                      |     | TMSEL     |
| L   | 10   | 1   | 123456 |     | SELECT TOKUI-M ASSIGN TO "TOKUI-UFS".            |     |           |
|     | 11   | 1   | 000100 |     | COPY SMSEL.                                      |     | SMSEL     |
| L   | 12   | 1   | 123456 |     | SELECT EXTERNAL SYOHIN-M ASSIGN TO "SYOHIN-UFS". |     |           |
|     | 13   | 1   | 000110 |     | DATA DIVISION.                                   |     |           |
|     | 14   | 1   | 000120 |     | FILE SECTION.                                    |     |           |
|     | 15   | 1   | 000130 |     | COPY TMFD.                                       |     | TMFD      |
| L   | 16   | 1   | 123456 |     | FD TOKUI-M IS GLOBAL                             |     |           |
| L   | 17   | 1   |        |     | LABEL RECORD IS STANDARD.                        |     |           |
| L   | 18   | 1   |        |     | 01 TOKUI-R.                                      |     |           |
| L   | 19   | 1   |        |     | 02 T-CODE PICTURE X(6).                          |     |           |
| L   | 20   | 1   |        |     | 02 T-NAME PICTURE X(20).                         |     |           |

COBOL プログラミングの手引 — 第 15 章翻訳リスト

|   |    |   |        |                           |      |
|---|----|---|--------|---------------------------|------|
| L | 21 | 1 |        | 02 FILLER PICTURE X(14).  |      |
|   | 22 | 1 | 000140 | COPY SMFD.                | SMFD |
| L | 23 | 1 | 123456 | FD SYOHIN-M IS GLOBAL     |      |
| L | 24 | 1 |        | LABEL RECORD IS STANDARD. |      |
| L | 25 | 1 |        | 01 SYOHIN-R.              |      |
| L | 26 | 1 |        | 02 S-CODE PICTURE X(6).   |      |
| L | 27 | 1 |        | 02 S-NAME PICTURE X(20).  |      |
| L | 28 | 1 |        | 02 S-TANKA PICTURE 9(5).  |      |
| L | 29 | 1 |        | 02 FILLER PICTURE X(9).   |      |
|   | 30 | 1 | 000150 | WORKING-STORAGE SECTION.  |      |
|   | 31 | 1 | 000160 | 01 D-LINE PIC 9.          |      |

<自由形式正書法（診断メッセージなしまたは診断メッセージを埋め込み出力しない）の場合>

| (1)  | (2) | (3)    | (4)                             | (5) | (6) | (7)       | (8) |
|------|-----|--------|---------------------------------|-----|-----|-----------|-----|
| LINE | PN  | SEQ-NO | ***** SOURCE PROGRAM LIST ***** |     |     | TEXT.NAME |     |
| 1    | 1   |        | IDENTIFICATION DIVISION.        |     |     |           |     |
| 2    | 1   |        | PROGRAM-ID. SAMPLE.             |     |     |           |     |
| 3    | 1   |        | ENVIRONMENT DIVISION.           |     |     |           |     |
| 4    | 1   |        | CONFIGURATION SECTION.          |     |     |           |     |
| 5    | 1   |        | SOURCE-COMPUTER. PC.            |     |     |           |     |
| 6    | 1   |        | OBJECT-COMPUTER. PC.            |     |     |           |     |
| 7    | 1   |        | INPUT-OUTPUT SECTION.           |     |     |           |     |
| 8    | 1   |        | FILE-CONTROL.                   |     |     |           |     |
| 9    | 1   |        | COPY TMSSEL.                    |     |     | TMSSEL    |     |

COBOL プログラミングの手引 ― 第 15 章翻訳リスト

|   |    |   |                                                  |       |
|---|----|---|--------------------------------------------------|-------|
| L | 10 | 1 | SELECT TOKUI-M ASSIGN TO "TOKUI-UFS".            |       |
|   | 11 | 1 | COPY SMSEL.                                      | SMSEL |
| L | 12 | 1 | SELECT EXTERNAL SYOHIN-M ASSIGN TO "SYOHIN-UFS". |       |
|   | 13 | 1 | DATA DIVISION.                                   |       |
|   | 14 | 1 | FILE SECTION.                                    |       |
|   | 15 | 1 | COPY TMFD.                                       | TMFD  |
| L | 16 | 1 | FD TOKUI-M IS GLOBAL                             |       |
| L | 17 | 1 | LABEL RECORD IS STANDARD.                        |       |
| L | 18 | 1 | 01 TOKUI-R.                                      |       |
| L | 19 | 1 | 02 T-CODE PICTURE X(6).                          |       |
| L | 20 | 1 | 02 T-NAME PICTURE X(20).                         |       |
| L | 21 | 1 | 02 FILLER PICTURE X(14).                         |       |
|   | 22 | 1 | COPY SMFD.                                       | SMFD  |
| L | 23 | 1 | FD SYOHIN-M IS GLOBAL                            |       |
| L | 24 | 1 | LABEL RECORD IS STANDARD.                        |       |
| L | 25 | 1 | 01 SYOHIN-R.                                     |       |
| L | 26 | 1 | 02 S-CODE PICTURE X(6).                          |       |
| L | 27 | 1 | 02 S-NAME PICTURE X(20).                         |       |
| L | 28 | 1 | 02 S-TANKA PICTURE 9(5).                         |       |
| L | 29 | 1 | 02 FILLER PICTURE X(9).                          |       |
|   | 30 | 1 | WORKING-STORAGE SECTION.                         |       |
|   | 31 | 1 | 01 D-LINE PIC 9.                                 |       |

## &lt;固定形式正書法（診断メッセージを埋め込み出力する）の場合&gt;

| (1) | (2)  | (3) | (4)    | (5) | (6)                             | (7) | (8)       |
|-----|------|-----|--------|-----|---------------------------------|-----|-----------|
|     | LINE | PN  | SEQ-NO |     | ***** SOURCE PROGRAM LIST ***** |     | TEXT.NAME |
|     | 1    | 1   | 000010 |     | IDENTIFICATION DIVISION         |     |           |
|     | 2    | 1   | 000020 |     | PROGRAM-ID. SAMPLE1.            |     |           |
|     |      |     |        |     | > W D0016 必要な終止符がありません。         |     |           |
|     | 3    | 1   | 000030 |     | DATA DIVISION.                  |     |           |
|     | 4    | 1   | 000040 |     | WORKING-STORAGE SECTION.        |     |           |
|     | 5    | 1   | 000050 |     | 01 ANS PIC X(10).               |     |           |
|     | 6    | 1   | 000060 |     | PROCEDURE DIVISION.             |     |           |
|     | 7    | 1   | 000070 |     | STEP01.                         |     |           |
|     | 8    | 1   | 000080 |     | COPY COPY1.                     |     | COPY1     |
| L   | 9    | 1   | 000010 |     | ACCEPT ANS2.                    |     |           |
|     |      |     |        |     | > F E0106 `ANS2`は定義されていません。     |     |           |
| L   | 10   | 1   | 000020 |     | ACCEPT ANS.                     |     |           |
|     | 11   | 1   | 000090 |     | STEP99.                         |     |           |
|     | 12   | 1   | 000100 |     | STOP RUN.                       |     |           |

<自由形式正書法（診断メッセージを埋め込み出力する）の場合>

| (1) | (2)  | (3) | (4)    | (5) | (6)                             | (7) | (8)       |
|-----|------|-----|--------|-----|---------------------------------|-----|-----------|
|     | LINE | PN  | SEQ-NO |     | ***** SOURCE PROGRAM LIST ***** |     | TEXT.NAME |
|     | 1    | 1   |        |     | IDENTIFICATION DIVISION         |     |           |
|     | 2    | 1   |        |     | PROGRAM-ID. SAMPLE1.            |     |           |
|     |      |     |        |     | > W D0016 必要な終止符がありません。         |     |           |
|     | 3    | 1   |        |     | DATA DIVISION.                  |     |           |
|     | 4    | 1   |        |     | WORKING-STORAGE SECTION.        |     |           |
|     | 5    | 1   |        |     | 01 ANS PIC X(10).               |     |           |
|     | 6    | 1   |        |     | PROCEDURE DIVISION.             |     |           |
|     | 7    | 1   |        |     | STEP01.                         |     |           |
|     | 8    | 1   |        |     | COPY COPY1.                     |     | COPY1     |
| L   | 9    | 1   |        |     | ACCEPT ANS2.                    |     |           |
|     |      |     |        |     | > F E0106 `ANS2`は定義されていません。     |     |           |
| L   | 10   | 1   |        |     | ACCEPT ANS.                     |     |           |
|     | 11   | 1   |        |     | STEP99.                         |     |           |
|     | 12   | 1   |        |     | STOP RUN.                       |     |           |



## 15.4 クロスリファレンスリスト

クロスリファレンスリスト(相互参照表)は、原始プログラムで使われているデータ名や手続き名の定義および参照に関する情報を含んでいる。

| 項目  | 出力項目   | 内容                                                                                                                                                                          |
|-----|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | 利用者定義語 | 利用者の定義した名前をシフト JIS コードに基づいて分類された順に出力する。                                                                                                                                     |
| (2) | 定義行番号  | 利用者定義語が定義されている行番号を出力する。<br>固定形式正書法の原始プログラムの場合、-Ce オプションありであれば、外部行番号で表示する。-Ce オプションなし(既定値)であれば、内部行番号で表示する。<br>自由形式正書法の原始プログラムの場合、常に内部行番号で表示する。<br>S.R. は特殊レジスタおよび暗黙の定義を意味する。 |
| (3) | 参照行番号  | 利用者定義語が参照されている行番号を出力する。<br>固定形式正書法の原始プログラムの場合、-Ce オプションありであれば、外部行番号で表示する。-Ce オプションなし(既定値)であれば、内部行番号で表示する。<br>自由形式正書法の原始プログラムの場合、常に内部行番号で表示する。                               |

| *** (1) SYMBOL *** | (2) DEF.NO | (3) REF.NO | ***** CROSS REFERENCE LIST ***** |
|--------------------|------------|------------|----------------------------------|
| COMPLETION-CODE    | S.R.       |            |                                  |
| D-LINE             |            | 31         |                                  |
| DD                 |            | 77         |                                  |
| DEMPYO-NO          |            | 32         |                                  |
| EXCEPTION-CODE     | S.R.       |            |                                  |
| EXCEPTION-LINE     | S.R.       |            |                                  |
| IN-DATE            |            | 33         | 57                               |
| IN-DD              |            | 36         |                                  |
| IN-MM              |            | 35         |                                  |
| IN-YY              |            | 34         |                                  |
| INIT-ROUTINE       |            | 55         | 65                               |
| INIT-ROUTINE-END   |            | 68         |                                  |
| KINGAKU            |            | 38         |                                  |
| LOOP-T-CODE        |            | 61         | 63                               |
| MAIN               |            | 40         |                                  |
| MM                 |            | 76         | 62                               |
| NOT-FOUND          |            | 66         |                                  |
| OPEN-FILE          |            | 79         |                                  |
| S-CODE             |            | 26         |                                  |
| S-NAME             |            | 27         |                                  |
| S-TANKA            |            | 28         |                                  |
| SURYO              |            | 37         |                                  |

|          |    |    |    |    |
|----------|----|----|----|----|
| SYOHIN-M | 23 |    |    |    |
| SYOHIN-R | 25 |    |    |    |
| T-CODE   | 19 | 63 |    |    |
| T-CODE   | 53 | 60 | 63 |    |
| T-NAME   | 20 | 64 |    |    |
| TOKUI-M  | 16 | 62 | 69 | 80 |
| TOKUI-R  | 18 |    |    |    |
| WK-DATA  | 49 | 56 | 57 |    |
| WK-DD    | 52 |    |    |    |
| WK-MM    | 51 |    |    |    |
| WK-YY    | 50 |    |    |    |
| YY       | 75 |    |    |    |
| YYMMDD   | 74 |    |    |    |

## ■ リポジトリ段落で指定された各指定子の定義行番号・参照行番号出力

### ・定義行番号

1. リポジトリ段落の指定子が、リポジトリ段落が含まれる定義名を指す場合、または、コンパイラが暗黙的に定義名の指定子を生成する場合は、見出し部の定義行番号を出力する。
2. リポジトリ段落に **BASE** クラスの記述がある場合は、その行番号を定義行番号として出力する。リポジトリ段落に **BASE** クラスの記述がない場合は、暗黙の定義として定義行番号に"**S.R**"を出力する。
3. 上記以外のリポジトリ段落の各指定子は、その行番号を定義行番号として出力する。

※ 1.の対象となる指定子は、クラス指定子/プログラム指定子/関数指定子とする。  
プロパティ指定子は常にリポジトリ段落の行番号を定義行番号として出力する。

### ・参照行番号

手続き中にリポジトリ段落の指定子を参照する記述がある場合、その指定子を含む文の行番号を参照行番号として出力する。

※ たとえば、手続き中に以下の記述がある場合のクラス名の定義行番号および参照行番号は、次のとおりである。

[**INVOKE** クラス名 "メソッド名"]記述の場合

- ・クラス名が自クラスの場合 : 定義行番号=見出し部の行番号  
参照行番号=**INVOKE** 文の行番号
- ・クラス名が自クラス以外の場合 : 定義行番号=リポジトリ段落の行番号  
参照行番号=**INVOKE** 文の行番号

[**INVOKE BASE** "メソッド名"]記述の場合

- ・**BASE** が明示的な定義の場合 : 定義行番号=リポジトリ段落の行番号  
参照行番号=**INVOKE** 文の行番号
- ・**BASE** が暗黙的な定義の場合 : 定義行番号=**S.R**  
参照行番号=**INVOKE** 文の行番号

## ■ オブジェクト参照の定義行番号・参照行番号出力

### ・定義行番号

データ部に **USAGE OBJECT REFERENCE** で定義したデータ名(オブジェクト参照項目)の定義行番号を出力する。

・参照行番号

手続き中にオブジェクト参照項目のデータ名を参照する記述がある場合、そのデータ名を含む文の行番号を参照行番号として出力する。

※「USAGE OBJECT REFERENCE クラス名」記述のオブジェクト参照項目の場合

- ・データ名の定義行番号および手続き中の参照行番号を出力する。
- ・クラス名に関する参照行番号は出力しない。

## 15.5 登録集参照リスト

登録集参照リストは、原始プログラムで COPY 文により複写された原文に関する情報を含んでいる。

| 項目  | 出力項目         | 内容                                                                                                                                                               |
|-----|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | 登録集原文名       | 複写された登録集の原文名を出力する。                                                                                                                                               |
| (2) | 登録集参照行<br>番号 | 登録集参照の COPY 文が書かれている行番号を出力する。<br>固定形式正書法の原始プログラムの場合、-Ce オプションありであれば、<br>外部行番号で表示する。-Ce オプションなし(既定値)であれば、内部行<br>番号で表示する。<br>自由形式正書法の原始プログラムの場合は、常に内部行番号で表示す<br>る。 |

(1)  
T-NAME  
SMFD  
SMSEL  
TMFD  
TMSEL

(2)  
REF.NO \*\*\*\*\* COPY REFERENCE LIST \*\*\*\*\*  
22  
11  
15  
9

## 15.6 診断メッセージリスト

診断メッセージは翻訳の結果、診断メッセージがあるとき出力されるリストで原始プログラムのどの行に誤りがあるかを示している。

| 項目  | 出力項目      | 内容                                                                                                                                                 |
|-----|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | 行番号       | エラーのある原始プログラムリスト上の行番号を出力する。<br>固定形式正書法の原始プログラムの場合、-Ce オプションありであれば、外部行番号で表示する。-Ce オプションなし(既定値)であれば、内部行番号で表示する。<br>自由形式正書法の原始プログラムの場合は、常に内部行番号で表示する。 |
| (2) | 登録集原文名    | 登録集原文内でエラーが発生した場合、その登録集原文名を出力する。                                                                                                                   |
| (3) | 登録集原文内行番号 | 登録集原文内でエラーが発生した場合、登録集原文内の相対行番号を出力する。登録集原文がなければ(1)の行番号と同じである。                                                                                       |
| (4) | エラーレベル    | エラーのレベルを出力する。<br>F:致命的エラー<br>W:警告エラー<br>O:注意                                                                                                       |
| (5) | エラー番号     | エラーメッセージの番号を出力する。                                                                                                                                  |
| (6) | エラー本文     | エラーメッセージの本文を出力する。                                                                                                                                  |

```

(1) (2) (3) (4) (5) (6)
LINE T-NAME T-LINE L DIAGNO ***** DIAGNOSTIC MESSAGE LIST *****

40 25 W E0009 必要な終止符がありません。
41 26 F S8251 ` ` が誤っています。
42 27 F J0204 ADD 文の作用対象が誤っています。
43 28 F J0201 MOVE 文の送り出し側が誤っています。
44 29 W J0216 MOVE 文で右の桁落ちが発生します。
66 51 F J0055 FROM または INTO 句の作用対象が誤っています。

```

## 15.7 データマップリスト

データマップリストは、原始プログラムで定義しているデータの性質や構造、アドレスに関する情報を含んでいる。

| 項目   | 出力項目                  | 内容                                                                                                                                                              |
|------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1)  | 行番号                   | 原始プログラムの行番号。<br>この行番号は、他の翻訳リストにおいても使用される。<br>固定形式正書法の原始プログラムの場合、-Ce オプションありであれば、外部行番号で表示する。-Ce オプションなし(既定値)であれば、内部行番号で表示する。<br>自由形式正書法の原始プログラムの場合、常に内部行番号で表示する。 |
| (2)  | レベル指示語<br>レベル番号       | レベル指示語(FD,SD)と、レベル番号(01,02～49,66,77,88)を示す。                                                                                                                     |
| (3)  | 利用者定義語                | 利用者定義語を示す。                                                                                                                                                      |
| (4)  | PICTURE 文字列または編集項目の種類 | 編集項目以外の PICTURE 文字列または編集項目の種類、および桁数を表示する。<br>ただし、PICTURE 文字列は利用者定義そのものではなく正規化した表現となる。                                                                           |
| (5)  | 外部属性                  | 各データ項目が外部属性(EXTERNAL)であるかどうかを示す。                                                                                                                                |
| (6)  | 全域属性                  | 各データ項目が全域属性(GLOBAL)であるかどうかを示す。                                                                                                                                  |
| (7)  | 用途                    | 各データ項目の明示的または暗黙的な用途(USAGE)を示す。                                                                                                                                  |
| (8)  | 割り付けアドレス              | 各データ項目の割り付けアドレスを、16 進 8 桁とカッコ内 10 進 10 桁の区分番号と、16 進 8 桁とカッコ内 10 進 10 桁の区分内相対アドレスで示す。                                                                            |
| (9)  | 項目長                   | 各データ項目の桁数を、16 進 8 桁とカッコ内 10 進 10 桁で示す。                                                                                                                          |
| (10) | 名前                    | プログラム名、関数名、クラス名、メソッド名を示す。                                                                                                                                       |

| (1)<br>LINE                 | (2)<br>LEVEL | (3)<br>SYMBOL | (4)<br>PICTURE<br>/COMMENT | (5)<br>EX | (6)<br>GL | (7)<br>USAGE | ADDRESS   | (8)          | LENGTH | (9)           |
|-----------------------------|--------------|---------------|----------------------------|-----------|-----------|--------------|-----------|--------------|--------|---------------|
| ***** SAMPLE ***** (10)     |              |               |                            |           |           |              |           |              |        |               |
| * FILE SECTION *            |              |               |                            |           |           |              |           |              |        |               |
| 16                          | FD           | TOKUI-M       |                            |           | G         |              |           |              |        |               |
| 18                          | 01           | TOKUI-R       | X(0000000040)              |           | G         | DISPLAY      | 00000001( | 1) 00000018( | 24)    | 00000028( 40) |
| 19                          | 02           | T-CODE        | X(0000000006)              |           | G         | DISPLAY      | 00000001( | 1) 00000018( | 24)    | 00000006( 6)  |
| 20                          | 02           | T-NAME        | X(0000000020)              |           | G         | DISPLAY      | 00000001( | 1) 0000001E( | 30)    | 00000014( 20) |
| 21                          | 02           | FILLER        | X(0000000014)              |           | G         | DISPLAY      | 00000001( | 1) 00000032( | 50)    | 0000000E( 14) |
| :                           |              |               |                            |           |           |              |           |              |        |               |
| * WORKING-STORAGE SECTION * |              |               |                            |           |           |              |           |              |        |               |
| :                           |              |               |                            |           |           |              |           |              |        |               |
| * LOCAL-STORAGE SECTION *   |              |               |                            |           |           |              |           |              |        |               |
| :                           |              |               |                            |           |           |              |           |              |        |               |
| * LINKAGE SECTION *         |              |               |                            |           |           |              |           |              |        |               |
| :                           |              |               |                            |           |           |              |           |              |        |               |
| * BASED SECTION *           |              |               |                            |           |           |              |           |              |        |               |
| :                           |              |               |                            |           |           |              |           |              |        |               |
| * PROCEDURE NAME *          |              |               |                            |           |           |              |           |              |        |               |

局所記憶節が記載された場合、データマップリストへ「\* LOCAL-STORAGE SECTION \*」見出しに続けて、局所記憶節で定義したデータ項目を出力する。

割り付けアドレスのブロック番号には、一連番号を設定する(1 基点)。

| (1)<br>LINE               | (2)<br>LEVEL | (3)<br>SYMBOL | (4)<br>PICTURE<br>/COMMENT | (5)<br>EX | (6)<br>GL | (7)<br>USAGE | ADDRESS   | (8)          | LENGTH | (9)           |
|---------------------------|--------------|---------------|----------------------------|-----------|-----------|--------------|-----------|--------------|--------|---------------|
| * LOCAL-STORAGE SECTION * |              |               |                            |           |           |              |           |              |        |               |
| 31                        | 01           | DATA1         | COMP-1                     |           |           | DISPLAY      | 00000001( | 1) 00000000( | 0)     | 00000002( 2)  |
| 32                        | 01           | DATA3         | X(20)                      |           |           | DISPLAY      | 00000002( | 2) 00000000( | 0)     | 00000014( 20) |
| 33                        | 02           | DATA3-1       | S9(10)                     |           |           | DISPLAY      | 00000002( | 2) 00000000( | 0)     | 0000000A( 10) |
| 34                        | 02           | DATA3-2       | X(10)                      |           |           | DISPLAY      | 00000002( | 2) 0000000A( | 10)    | 0000000A( 10) |

クラス定義の場合、ファクトリやインスタンス、どのメソッドのデータかを判別するための見出しを出力する。  
また、字類がオブジェクトの場合、用途に OBJECT を出力する。

| (1)<br>LINE | (2)<br>LEVEL | (3)<br>SYMBOL               | (4)<br>PICTURE<br>/COMMENT | (5)<br>EX | (6)<br>GL | (7)<br>USAGE | ADDRESS   | (8)          | LENGTH | (9)           |
|-------------|--------------|-----------------------------|----------------------------|-----------|-----------|--------------|-----------|--------------|--------|---------------|
| 1           |              | ***** SAMPLE_CLASS *****    |                            |           |           |              |           |              |        |               |
|             |              | *** FACTORY ***             |                            |           |           |              |           |              |        |               |
|             |              | * FILE SECTION *            |                            |           |           |              |           |              |        |               |
| 16          | FD           | TOKUI-M                     |                            |           |           |              |           |              |        |               |
| 18          | 01           | TOKUI-R                     | X(0000000040)              |           |           | DISPLAY      | 00000001( | 1) 00000018( | 24)    | 00000028( 40) |
|             |              | :                           |                            |           |           |              |           |              |        |               |
|             |              | * WORKING-STORAGE SECTION * |                            |           |           |              |           |              |        |               |
| 31          | 01           | D-LINE                      | 9(01)                      |           |           | DISPLAY      | 00000001( | 1) 00000170( | 368)   | 00000001( 1)  |
|             |              | :                           |                            |           |           |              |           |              |        |               |
|             |              | *** OBJECT ***              |                            |           |           |              |           |              |        |               |
|             |              | ** SAMPLE_METHOD **         |                            |           |           |              |           |              |        |               |
|             |              | * LOCAL-STORAGE SECTION *   |                            |           |           |              |           |              |        |               |
| 45          | 01           | WORK01                      |                            |           |           | OBJECT       | 00000001( | 1) 00000170( | 368)   | 00000004( 4)  |



## 15.8 用語の展開リスト

用語の展開リストは、原始プログラム中に用語名の属性を展開する。

| 項目  | 出力項目          | 内容                                               |
|-----|---------------|--------------------------------------------------|
| (1) | 行番号           | 原始プログラムの行番号。                                     |
| (2) | レベル番号         | 用語データ項目のレベル番号。                                   |
| (3) | 値付き用語データ項目の識別 | 値付き用語データ項目であるか否かを表す。<br>値付き用語データ項目であれば、"V"を表示する。 |
| (4) | 用語データ項目名      | 用語データ項目の名前。                                      |
| (5) | PIC 情報        | 用語データ項目の PIC 情報。                                 |
| (6) | 用途識別          | 用語データ項目の項類。                                      |
| (7) | 用語名           | プログラムで参照している用語名。                                 |

| (1)  | (2) | (3) | (4)            | (5)          | (6)    | (7)        |
|------|-----|-----|----------------|--------------|--------|------------|
| LINE | LN  | V   | TERM-DATA-NAME | PIC          | USAGE  | TERM-NAME  |
| 31   | 01  | *   | T_YOGO_09      | X(00005)     |        | T_YOGO_09  |
| 32   | 01  |     | A1             | S9(09)V9(09) | BINARY | T_YOGO_06C |

## 15.9 用語名リスト

用語名リストは、原始プログラム内で用語機能を使用している場合に、使用している用語名の情報を出力する。

| 項目  | 出力項目          | 内容                                               |
|-----|---------------|--------------------------------------------------|
| (1) | 値付き用語データ項目の識別 | 値付き用語データ項目であるか否かを表す。<br>値付き用語データ項目であれば、"V"を表示する。 |
| (2) | 用語名           | プログラムで参照している用語名。                                 |
| (3) | 用語データ項目名      | 用語データ項目の名前。                                      |
| (4) | 定義行           | 定義行番号。                                           |
| (5) | 属性名           | プログラムで参照している属性名。                                 |

・引用用語名がないとき

| (1)(2)      | (3)            | (4)    | (5)            |
|-------------|----------------|--------|----------------|
| V TERM-NAME | TERM-DATA-NAME | DEF.NO | ATTRIBUTE-NAME |
| * T_YOGO_09 | T_YOGO_09      | 31     |                |

・引用用語名があるとき

| (1) (2)      | (3)            | (4)    | (5)              |
|--------------|----------------|--------|------------------|
| V TERM-NAME  | TERM-DATA-NAME | DEF.NO | ATTRIBUTE-NAME   |
| T_YOGO_06B   |                |        | T_YOGO_06B_AT_01 |
| > T_YOGO_06C | A1             | 32     | T_YOGO_06C_AT_02 |

## 15.10 用語値名参照リスト

用語値名参照リストは、原始プログラム内で用語機能を使用している場合に、手続き部で参照している用語値名の情報を出力する。

・前半部（用語値名－用語名－参照行情報）

| 項目  | 出力項目 | 内容                |
|-----|------|-------------------|
| (1) | 用語値名 | プログラムで参照している用語値名。 |
| (2) | 用語名  | 対応する用語名。          |
| (3) | 参照行  | 用語値名の参照行番号。       |

・後半部（用語値名－用語名－用語値情報）

| 項目  | 出力項目 | 内容                |
|-----|------|-------------------|
| (4) | 用語値名 | プログラムで参照している用語値名。 |
| (5) | 用語名  | 対応する用語名。          |
| (6) | 用語値  | 用語値の情報。           |

・前半部（用語値名－用語名－参照行情報）

| (1)<br>TERM-VALUE-NAME | (2)<br>TERM-NAME | (3)<br>REF.NO |   |   |  |
|------------------------|------------------|---------------|---|---|--|
| T_YOGO_01V_1           | T_YOGO_01        | 3             | 4 | 5 |  |

・後半部（用語値名－用語名－用語値情報）

| (4)<br>TERM-VALUE-NAME | (5)<br>TERM-NAME | (6)<br>TERM-VALUE              |
|------------------------|------------------|--------------------------------|
| T_YOGO_01V_1           | T_YOGO_01        | "ABCDEF", "BCDEFG", "1234567", |

## 第16章 データ例外処理機能

### 16.1 概説

データ例外処理機能は、実行用プログラムの実行時に不正 10 進データ例外が発生したとき、そのプログラムの実行をただちに終了せず、エラー処理としてユーザが記述した処理を実行する機能である。

たとえば、オンラインプログラム等のデバッグ時においては、入力データ等の誤りによって、数字データ項目に数字以外のデータが入力されることがある。このようなとき、数字比較、数字転記および算術演算等の実行時に不正 10 進データ例外が発生する。データ例外処理機能を使用していないプログラムはただちに強制終了する。しかし、オンラインプログラムの場合は、そのプロセスが異常終了したことをシステムに知らせる等の処理が必要であり、データ例外処理機能を使用することによって、それらの処理を実行させることができる。

また、このとき必要ならば、データ例外の発生原因となったデータ項目の内容を表示したり、そのデータ項目をゼロで初期化して再度例外の発生した文から実行させたりすることができる。

したがって、プログラムの実行後、表示された情報によって、プログラムのどの文で、またどんなデータによって不正 10 進データ例外が発生したか、調べることができる。

### 16.2 データ例外手続き

実行用プログラムの実行時に不正 10 進データ例外が発生したとき、利用者の指定したデータ例外節に制御を移し、データ例外手続きを実行後、再び不正 10 進データ例外の発生した文に制御を戻す。

データ例外節は、宣言部分内に USE (データ例外節で使用) 文および必要なデータ例外手続きを書くことによって指定する。また、データ例外節の有無は、静的リンクしたメインプログラムとサブプログラムの間で統一しなければならない。なお、入れ子プログラム中にはデータ例外手続きは記述できない。

**DECLARATIVES.**

[データ例外節以外の節]

:

節名 SECTION.

USE AFTER STANDARD

|   |           |
|---|-----------|
| { | ERROR     |
|   | EXCEPTION |

PROCEDURE ON DATA ERROR.

**データ例外手続き**

[データ例外節以外の節]

:

**END DECLARATIVES.**

データ例外手続きには、データ例外節に対して暗に CALL 文が実行されたかのように制御が渡されるため、EXIT PROGRAM 文以外の文ならどの文を書いてもよいが、データ例外手続きの実行中、再度不正 10 進データ例外が発生すると、そのプログラムはただちに強制終了するので注意されたい。

データ例外手続き内だけで書けるものとして、次の文がある。

**(1) DISPLAY (データ例外表示) 文**

この文は、データ例外の発生原因となったデータ項目の内容を表示する。

データは、各文字を 2 桁の 16 進文字の形に変換して表示する。

DISPLAY 文の記述は、次のようになる。

**DISPLAY ERROR [UPON 呼び名].**

**(2) MOVE (データ例外初期化) 文**

この文は、データ例外の発生原因となったデータ項目に対して、表意定数 ZERO を送り出し側項目とする MOVE 文が実行されたかのようにゼロで初期化する。

MOVE 文の記述は、次のようになる。

**MOVE ZERO TO ERROR.**

**(3) ALTER (データ例外変更) 文**

本 COBOL では、メモ扱いである。この文の有無に関係なく、データ例外手続きの実行後、常にデータ例外の発生した文に制御が移る。

ALTER 文の記述は、次のようになる。

**ALTER EXIT TO [PROCEED TO] ERROR.**

**16.3 不正 10 進データ例外と COBOL 文**

実行用プログラムの実行時、入力データ等の誤りによって数字データ項目に数字以外のデータが入力されると、演算、数字比較、数字転記等において、不正 10 進データ例外が発生する。したがって、利用者は入力データ等の誤りによって、数字以外のデータが格納される可能性のある項目に対しては、字類条件を使用して数字データか否かの検査をした後に演算、数字比較、数字転記などの文を実行した方がよい。特に、オンラインプログラムにおいては、細心の注意が必要である。

**(1) 不正 10 進データ例外の発生条件 (例外コード:094,010)**



不正 10 進データ例外 (illegal decimal data exception) は、記憶域内のデータが参照されるとき、次のような条件で検出される。

(a) 内部 10 進データ(パック形式データ)

- 符号位置の内容が 10 進数字 (0～9) である。
- 数字位置の内容が A<sub>16</sub>～F<sub>16</sub> である。
- 桁数が偶数の時、最左端バイトの上位 4 ビットが 0 でない (-CN オプションが指定されていない時)。

(b) 外部 10 進データ(アンパック形式データ)

i. 重ね符号(符号部が独立していない)の場合

- 最右端(SIGN LEADING 句が指定されているときは最左端)バイトの内容が不正(表 16-1 以外の値が設定されている)である。
- 数字位置の内容が A<sub>16</sub>～F<sub>16</sub> である。

表 16-1 最右端(最左端)バイトの正しい内容

| 最右端(最左端)バイトの値 | 符 号              |                  |                  |                  |                  |                  |
|---------------|------------------|------------------|------------------|------------------|------------------|------------------|
|               | 符号なし             | 正符号付き            |                  |                  |                  | 負符号付き            |
|               | SD <sub>n</sub>  | SD <sub>n</sub>  | SD <sub>n</sub>  | SD <sub>n</sub>  | SD <sub>n</sub>  | SD <sub>n</sub>  |
| 0             | 30 <sub>16</sub> | 7B <sub>16</sub> | 78 <sub>16</sub> | 24 <sub>16</sub> | 7D <sub>16</sub> | 7A <sub>16</sub> |
| 1             | 31 <sub>16</sub> | 41 <sub>16</sub> | 7E <sub>16</sub> | 9F <sub>16</sub> | 4A <sub>16</sub> | A0 <sub>16</sub> |
| 2             | 32 <sub>16</sub> | 42 <sub>16</sub> | CD <sub>16</sub> | 53 <sub>16</sub> | 4B <sub>16</sub> | E0 <sub>16</sub> |
| 3             | 33 <sub>16</sub> | 43 <sub>16</sub> | CE <sub>16</sub> | 54 <sub>16</sub> | 4C <sub>16</sub> | E1 <sub>16</sub> |
| 4             | 34 <sub>16</sub> | 44 <sub>16</sub> | CF <sub>16</sub> | 55 <sub>16</sub> | 4D <sub>16</sub> | E2 <sub>16</sub> |
| 5             | 35 <sub>16</sub> | 45 <sub>16</sub> | D0 <sub>16</sub> | 56 <sub>16</sub> | 4E <sub>16</sub> | E3 <sub>16</sub> |
| 6             | 36 <sub>16</sub> | 46 <sub>16</sub> | D1 <sub>16</sub> | 57 <sub>16</sub> | 4F <sub>16</sub> | E4 <sub>16</sub> |
| 7             | 37 <sub>16</sub> | 47 <sub>16</sub> | D2 <sub>16</sub> | 58 <sub>16</sub> | 50 <sub>16</sub> | E5 <sub>16</sub> |
| 8             | 38 <sub>16</sub> | 48 <sub>16</sub> | D3 <sub>16</sub> | 59 <sub>16</sub> | 51 <sub>16</sub> | E6 <sub>16</sub> |
| 9             | 39 <sub>16</sub> | 49 <sub>16</sub> | D4 <sub>16</sub> | 5A <sub>16</sub> | 52 <sub>16</sub> | E7 <sub>16</sub> |

本コンパイラ固有の表現形式は上記のとおりである。

-CS オプション指定により、Pro\*COBOL 表現形式に変更することができる。その場合の数字と符号の重ね合わせを次に示す。

| 最右端(最左端)バイトの値 | 符 号              |                  |                  |
|---------------|------------------|------------------|------------------|
|               | 符号なし             | 正符号付き            | 負符号付き            |
|               | SD <sub>n</sub>  | SD <sub>n</sub>  | SD <sub>n</sub>  |
| 0             | 30 <sub>16</sub> | 30 <sub>16</sub> | 70 <sub>16</sub> |
| 1             | 31 <sub>16</sub> | 31 <sub>16</sub> | 71 <sub>16</sub> |
| 2             | 32 <sub>16</sub> | 32 <sub>16</sub> | 72 <sub>16</sub> |
| 3             | 33 <sub>16</sub> | 33 <sub>16</sub> | 73 <sub>16</sub> |
| 4             | 34 <sub>16</sub> | 34 <sub>16</sub> | 74 <sub>16</sub> |
| 5             | 35 <sub>16</sub> | 35 <sub>16</sub> | 75 <sub>16</sub> |
| 6             | 36 <sub>16</sub> | 36 <sub>16</sub> | 76 <sub>16</sub> |
| 7             | 37 <sub>16</sub> | 37 <sub>16</sub> | 77 <sub>16</sub> |
| 8             | 38 <sub>16</sub> | 38 <sub>16</sub> | 78 <sub>16</sub> |
| 9             | 39 <sub>16</sub> | 39 <sub>16</sub> | 79 <sub>16</sub> |

ii. 独立符号の場合

- 符号位置の内容が不正 ( $2B_{16}$ ,  $2D_{16}$  以外) である。
- 数字位置の内容が不正 (0~9 以外) である。

iii. 符号なしの場合

- 数字位置の内容が不正 (0~9 以外) である。
- 最右端バイトの符号部はチェックしない。

**備考** 翻訳時に -Cd オプション指定で翻訳された実行可能プログラムにおいては不正 10 進データ例外は発生しない。

外部 10 進データ項目に、文字データが上記 (b) 以外の条件で格納されていると、不正 10 進データ例外は発生しない。すなわち、10 進データとして取り扱われる。したがって、字類条件を使用して、数字データか否かの検査をした方が望ましい。また、外部 10 進データの最右端バイトが 16 進の "20" の場合、不正 10 進データ例外は発生せず、値ゼロとして取り扱われる。

**例** 外部 10 進データ (文字) : 12C△

内部表現形式 (16 進) : 31324320

上記の外部 10 進データは 10 進数値、+1230 として取り扱われる。

**【Enterprise ACOS-4 互換】** -Fe オプションを指定することにより、NUMERIC 検査の内容を変更することができる。

- ・偶数桁の内部 10 進数の最左端上位 4 ビットが 0 以外の場合、偽とする (-CN オプションが指定されていない時)。不正 10 進エラーにはならない。
- ・外部 10 進数の独立符号部が空白の場合、真とする。不正 10 進エラーにはならない。

-Ft オプションを指定することにより、符号なし外部 10 進数の最右端バイトの符号部 (S) のチェックを行う。

**【Enterprise for Windows ACOS-4 互換】** -Fz オプションを指定することにより、データ例外が発生する手続き文を ACOS-4 互換とする。

次の手続き文について、不正 10 進チェックを行う。

- (1) DISPLAY 文の作用対象が数字項目のとき
- (2) COMBINE 文の SHIFTED 句の作用対象が外部 10 進項目のとき
- (3) SORT 文または MERGE 文の KEY 句の作用対象が数字項目のとき

## 16.4 メッセージの出力形式

データ例外手続き内で、DISPLAY (データ例外表示) 文を実行すると、次に示すメッセージが表示される。

COBxxx 不正 1 0 進データを検出しました プログラム名=zzzzz 行番号=nnnnnn  
Data: XX... [データ項目の大きさ分出力]

### 説明

- (1) zzzzz にはプログラム名が設定される。
- (2) nnnnnn には内部行番号を設定する。ただし、-Ce オプション指定ありの場合は外部行番号を設定する。
- (3) XX... には不正 10 進データ例外を起こしたデータ項目の内容が 16 進表記で設定される。

(4) [Enterprise for Linux] 以下の条件をすべて満たす場合は英文のエラーメッセージが表示される。

- 環境変数 LANG=ja\_JP.SJIS でない
- 環境変数 LANG=ja\_JP.UTF-8 でないか環境変数 COB\_CODECHG=ON でない、または呼び名が標準出力、標準エラー出力または CONSOLE に関連づけられていない

## 16.5 COBOL データ例外処理

不正 10 進データ例外が発生したプログラムにデータ例外節があれば、暗に CALL 文が実行されたかのように、データ例外節に制御が渡される。

データ例外節は、各実行用プログラムごとに記述しなければならない。

たとえば、**活性化する側の実行時要素**にデータ例外節が記述されているが、**活性化される側の実行時要素**にデータ例外節が記述されていないと、その活性化される側の実行時要素で不正 10 進データ例外が発生しても、活性化する側の実行時要素で記述したデータ例外節に制御は渡らない。

不正 10 進データ例外以外の例外が発生した場合は、データ例外節に制御は渡らない。

## 16.6 データ例外発生原因の調べ方

データ例外処理機能を使用した COBOL 実行用プログラムの実行中、不正 10 進データ例外が発生したとき、実行時エラーメッセージを出力して処理を続行することによって、プログラム実行後、どのデータ項目を参照したとき、不正 10 進データ例外が発生したのか調べることができる。

# 第17章 利用者制御変数処理機能

## 17.1 概説

利用者制御変数を使用する COBOL プログラムの実行時には、最上位のプロセスにおいて、後述するコマンドラインツールで利用者制御変数保持ファイルを作成し、そのファイル名を環境変数 COB\_UC\_FILE に設定しなければならない。また、終了時には使用したファイルを削除する必要がある。

環境変数 COB\_UC\_FILE が定義されていない場合や環境変数 COB\_UC\_FILE で指定したファイルが存在しない場合は、COBOL プログラムは利用者制御変数を使用する際に実行時エラーメッセージを出力し、異常終了する。

以降では、バッチファイルまたはシェルフファイルの使用を想定して説明している。

## 17.2 環境変数

利用者制御変数保持ファイル名を設定する環境変数 COB\_UC\_FILE に、後述するコマンドラインツールで作成した利用者制御変数保持ファイル名を設定する。

### 設定方法

[Enterprise for Windows] [Standard] set COB\_UC\_FILE=ファイル名

[Enterprise for Linux] export COB\_UC\_FILE=ファイル名

## 17.3 コマンドラインツール

プロセスツリーの最上位プロセスであるバッチファイルまたはシェルフファイルにおいてファイルの作成・削除を行う機能、任意のバッチファイルまたはシェルフファイルにおいて利用者制御変数を設定・参照する機能を次のコマンドラインツールとして提供する。

| 機能               | コマンド名     |
|------------------|-----------|
| 利用者制御変数保持ファイルの作成 | cobinituc |
| 利用者制御変数の設定       | cobsetuc  |
| 利用者制御変数の参照       | cobreaduc |
| 利用者制御変数保持ファイルの削除 | cobtermuc |

### 17.3.1 cobinituc

#### 機能

cobinituc は、利用者制御変数保持ファイルを作成する。

#### 構文

cobinituc [ディレクトリ名]

#### 返却値

正常時は、作成した利用者制御変数保持ファイルに対するフルパスを標準出力に出力する。異常を検出しファイルが作成できない場合は、空文字列を標準出力に出力する。

**[Enterprise for Windows] [Standard] 説明**

一意な利用者制御変数保持ファイルを作成し、作成したファイルのフルパスを標準出力に出力する。

引数を指定した場合、引数はディレクトリ名と解釈する。

次の順で、利用者制御変数保持ファイルの作成を試みる。

- 環境変数 TMP
- 引数に指定したディレクトリ
- . (カレントディレクトリ)

異常を検出しファイルが作成できない場合は、空文字列を標準出力に出力する。

**cobinituc** は、環境変数には値を設定しない。環境変数への設定はバッチファイルの **set** コマンドと組み合わせて実現する。

**[Enterprise for Linux] 説明**

一意な利用者制御変数保持ファイルを作成し、作成したファイルのフルパスを標準出力に出力する。

引数を指定した場合、引数はディレクトリ名と解釈する。

次の順で、利用者制御変数保持ファイルの作成を試みる。

- 環境変数 TMPDIR
- 引数に指定したディレクトリ
- . (カレントディレクトリ)

異常を検出しファイルが作成できない場合は、空文字列を標準出力に出力する。

**cobinituc** は、環境変数には値を設定しない。環境変数への設定はシェルフファイルの **export** コマンドと組み合わせて実現する。

### 17.3.2 cobsetuc

#### 機能

**cobsetuc** は、利用者制御変数を設定する。

#### 構文

```
cobsetuc ucname { ITGmmm=no | CHARnnn=st }
```

**ucname**: 利用者制御変数名

**mmm**: 001 ~ 009

**nnn**: 001 ~ 255

**no** : **mmm** に指定した桁数の数値

**st** : **nnn** に指定した長さの文字

#### 返却値

正常時は、引数で指定された **no** または **st** を標準出力に出力する。異常時は空文字列を標準出力に出力する。

#### 説明

環境変数 **COB\_UC\_FILE** で指定されたファイルを利用者制御変数保持ファイルとみなし、このファイルでの利用者制御変数を変更する。

引数に指定した利用者制御変数の値を指定された値に変更する。

環境変数 **COB\_UC\_FILE** が定義されていない場合や環境変数 **COB\_UC\_FILE** で指定したファイルが存在しない場合は異常終了する。

### 17.3.3 cobreaduc

#### 機能

**cobreaduc** は、利用者制御変数を参照する。

#### 構文

```
cobreaduc ucname { ITGmmm | BARnnn }
 ucname: 利用者制御変数名
 mmm: 001 ~ 009
 nnn: 001 ~ 255
```

#### 返却値

正常時は、引数で指定された利用者制御変数の値を標準出力に出力する。異常時は、空文字列を標準出力に出力する。

#### 説明

指定された利用者制御変数の状態を得る。

環境変数 **COB\_UC\_FILE** が定義されていない場合や環境変数 **COB\_UC\_FILE** で指定したファイルが存在しない場合は異常終了する。

### 17.3.4 cobtermuc

#### 機能

**cobtermuc** は、利用者制御変数保持ファイルを削除する。

#### 構文

**cobtermuc**

#### 返却値

正常時は、空文字列を標準出力に出力する。異常時は、ファイルが削除できなかった場合はファイルのフルパスを標準出力に出力する。これ以外の異常が発生した場合は、- (半角ハイフン 1 文字) を標準出力に出力する。

他のコマンドラインツールの返却値と異なるため、注意されたい。

#### [Enterprise for Windows] [Standard] 説明

環境変数 **COB\_UC\_FILE** で指定されたファイルを削除する。

**cobtermuc** は、環境変数には値を設定しない。環境変数への設定はバッチファイルの **set** コマンドと組み合わせて実現する。

#### [Enterprise for Linux] 説明

環境変数 **COB\_UC\_FILE** で指定されたファイルを削除する。

**cobtermuc** は、環境変数には値を設定しない。環境変数への設定はシェルフファイルの **export** コマンドと組み合わせて実現する。



## 17.4 使用例

[Enterprise for Windows] [Standard] バッチファイルにおける利用者制御変数の使用例を示す。

```
FOR /F %%w IN ('cobinituc') DO SET COB_UC_FILE=%%w
cobsetuc username1 ITG001=7
COBOLAP1
FOR /F %%w IN ('cobreaduc username1 ITG001') DO SET UC1=%%w
IF %UC1% EQU 7 (COBOLAP2) ELSE (COBOLAP3)
cobtermuc
IF %ERRORLEVEL% EQU 0 (SET COB_UC_FILE=)
```

[Enterprise for Linux] シェルファイルにおける利用者制御変数の使用例を示す。

```
w=`cobinituc`
export COB_UC_FILE=$w
cobsetuc UNAME1 ITG001=7
COBOLAP1
w=`cobreaduc UNAME1 ITG001`
if [$w -eq 7]
then
 COBOLAP2
else
 COBOLAP3
fi
cobtermuc
if [$? = 0]
then
 export COB_UC_FILE=
fi
```

## 第18章 日本語情報処理機能

### 18.1 概説

日本語情報処理機能は、英字および数字データと同様に、漢字、ひらがなおよびカタカナを含む日本語データを容易に取り扱うことができる拡張機能である。

日本語情報処理機能には、日本語の定数、日本語データ項目の定義、日本語データの操作（初期化、比較、転記、整列併合、連結、分解、文字列検査）および日本語データ入力、表示などの機能がある。また、簡単な日本語データの印刷であれば、書式オーバーレイ制御印刷機能を使用せずに印刷することができる。

### 18.2 ファイル入出力機能

#### 18.2.1 APPLY SHIFT-CODE 句

WRITE 文を実行すると、編集用レコード領域の左隣に、ASA コード(1 バイト)を付加して出力される。このとき、ASA コードエリアには、改行制御情報が格納される。

直接プリンタファイルへ出力すると、シフトコードやプリンタ制御コードが文字化けするため、本機能を使用するときは、必ず LABEL RECORD STANDARD を指定し、ファイルに出力後、SpoolServer 経由で印字しなければならない。

APPLY SHIFT-CODE 句を指定する場合、外部記憶媒体上の文字集合はシフト JIS として扱われる。  
-CU オプションが指定された場合は、プログラムの固有文字集合からシフト JIS へコード変換が行われ、出力される。

【Standard】【Enterprise for Windows】-CF オプションを指定した場合は、コード変換は行われず出力される。

-Cy オプション指定なしのとき、プリンタファイルの APPLY SHIFT-CODE 句は有効である。プリンタファイルの印刷データに、プリンタ制御コードを埋め込む機能を使用できる。  
-Cy オプション指定ありのとき、APPLY SHIFT-CODE 句はメモ扱いとなり、指定しても無効である。  
APPLY SHIFT-CODE 句で WITH PPR-CONTROL-1 または WITH PPR-CONTROL-2 指定ありのプリンタファイルに対して、CHARACTER TYPE 句および COLUMN NUMBER 句を指定しても無効である。

18.2.1.1 改行制御

ASA コードエリアには改行制御情報が格納されるが、WRITE 文の ADVANCING 句の書き方によって、それぞれの値が格納される。

改行制御情報の具体的な値については、次の表の改行制御文字を参照。

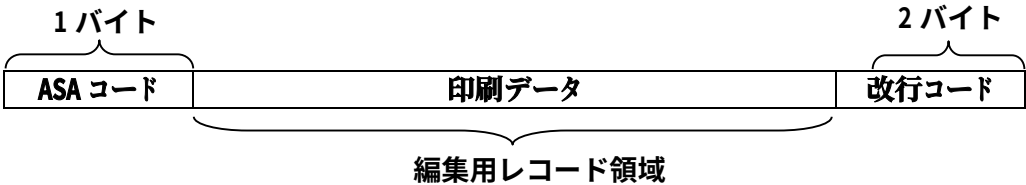
| ADVANCING | ASA 制御文字                 | 機能             |
|-----------|--------------------------|----------------|
| 指定なし      | △:1 行改行後印刷               | 1 行改行後印字する     |
| 一意名       | +:改行なし<br>△:1 行改行後印刷     | 一意名の内容だけ相対改行する |
| 整数        | 0:2 行改行後印刷<br>-:3 行改行後印刷 | 整数だけ相対改行する     |
| PAGE      | 1:改頁                     | 改ページする         |
| 呼び名       | 使用不可                     | —              |

記号の説明:△は ANK スペースを表す。

3 行以上の改行の場合は、3 行改行、2 行改行および 1 行改行のダミーレコードを組み合わせで出力する。BEFORE 指定のとき、“+”で改行なしの印刷データを出力した後、ダミーレコード (ASA 制御文字のみで印刷データなしのレコード) を ADVANCING 句の指定によって何度か出力する。

18.2.1.2 レコード形式

利用者は、ASA コードエリアを意識する必要はない。WRITE 文の実行時、編集用レコード領域の左端 1 バイトに ASA コードエリア、右端 2 バイトに改行コードを付加する。  
WRITE 文の改行／改頁の指定内容が ASA コードエリアへ設定される。



改行コード:0D0A (16 進表記)

18.2.1.3 レコード出力例

改行制御指定があるときのレコード出力内容を次に示す。

**例 1** AFTER ADVANCING 5 LINES 指定のとき

|   |       |      |
|---|-------|------|
| — | OD0A  |      |
| 0 | 印刷データ | OD0A |

**例 2** BEFORE ADVANCING 5 LINES 指定のとき

|   |      |       |      |
|---|------|-------|------|
| + |      | 印刷データ | OD0A |
| — | OD0A |       |      |
| 0 | OD0A |       |      |

## 第19章 64 ビットアプリケーションの生成方法

### 19.1 Windows 版の場合

[Enterprise for Windows] [Standard] 本機能は Enterprise Edition for Windows 版および Standard Edition 版において、使用できる。

#### 19.1.1 アプリケーションの生成と実行の環境

本コンパイラの動作 OS と生成可能なアプリケーションを次に示す。

| 開発 OS  | 生成アプリケーション         |
|--------|--------------------|
| 64 ビット | 64 ビットアプリケーション(*1) |

(\*1) 関連製品との組み合わせが必要な機能に制限がある。詳細は「第 21 章 関連製品連携」参照。

本コンパイラで生成した 64 ビットアプリケーションは、実行マシンが 64 ビット OS であれば動作可能である。

なお、COBOL アプリケーションの中から他言語で作成したプログラムを呼び出している場合には、そのプログラムも COBOL アプリケーションに合わせて 64 ビットアプリケーションでなければならない。これは、他言語から COBOL アプリケーションを呼び出す場合の他言語についても同様である。

#### 19.1.2 翻訳環境の設定

アプリケーションを生成するには、

- ① Visual Studio コマンドプロンプト上で COBOL コマンドを起動して生成する
- ② COBOL 開発環境を使って生成する

の 2 通りの方法がある。

以下、①の生成方法についての注意点を記載する。②の COBOL 開発環境を使って生成する方法については『COBOL 開発環境 利用の手引』を参照。

① **Visual Studio** コマンドプロンプト上で COBOL コマンドを起動して生成する場合

**Visual Studio** をインストールしている場合、環境設定は不要である。

ただし、使用する **Visual Studio** コマンドプロンプトが異なるので、注意してください。

・64 ビット OS 上で 64 ビットアプリケーションを生成する場合

**Windows Server 2019:**

**Visual Studio 2019**

[スタート]→[Visual Studio 2019]→[x64 Native Tools Command Prompt for VS 2019]

**Visual Studio 2022**

[スタート]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

**Windows Server 2022:**

**Visual Studio 2022**

[スタート]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

**Windows 10:**

**Visual Studio 2015**

[スタート]→[Visual Studio 2015]→[Visual Studio Tools]→[VS2015 x64 Native Tools コマンド プロンプト]

**Visual Studio 2017**

[スタート]→[Visual Studio 2017]→[VS 2017 用 x64 Native Tools コマンド プロンプト]

**Visual Studio 2019**

[スタート]→[Visual Studio 2019]→[x64 Native Tools Command Prompt for VS 2019]

**Visual Studio 2022**

[スタート]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

**Windows 11:**

**Visual Studio 2022**

[スタート]→[すべてのアプリ]→[Visual Studio 2022]→[x64 Native Tools Command Prompt for VS 2022]

### 19.1.3 コマンドの指定方法

本コンパイラでアプリケーションを生成するには、Visual Studio コマンドプロンプトで、COBOL コンパイラを実行する。

COBOL ソースファイル名(メインプログラム)が、「sample.cob」である場合の、64 ビットの例を次に記載する。

・「sample.cob」である場合の例を次に記述する。

```
> cob -M sample.cob
```

### 19.1.4 64 ビットでの言語仕様

64 ビットでの、言語仕様を示す。

・項目長

| 用途               | [Enterprise for Windows]          | [Standard]       |
|------------------|-----------------------------------|------------------|
| USAGE IS DISPLAY |                                   |                  |
| 英字項目・英数字項目       | 1 バイト/1 文字                        | 1 バイト/1 文字       |
| 外部 10 進項目        | 1 バイト/1 桁                         | 1 バイト/1 桁        |
| 数字編集項目           | 1 バイト/1 桁                         | 1 バイト/1 桁        |
| 英数字編集項目          | 1 バイト/1 桁                         | 1 バイト/1 桁        |
| 日本語項目            | 2 バイト/1 文字                        | 2 バイト/1 文字       |
| 日本語編集項目          | 2 バイト/1 文字                        | 2 バイト/1 文字       |
| USAGE IS BINARY  |                                   |                  |
| 1～2 桁            | 1 バイト<br>(-Fa オプション指定時,<br>2 バイト) | 2 バイト            |
| 3～4 桁            | 2 バイト                             | 2 バイト            |
| 5～9 桁            | 4 バイト                             | 4 バイト            |
| 10～18 桁          | 8 バイト                             | 8 バイト            |
| USAGE IS COMP    | USAGE IS BINARY<br>(-Fc オプション指定時, | USAGE IS DISPLAY |

|                                            |                                                                                                                         |                                                                                                                         |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
|                                            | DISPLAY)                                                                                                                |                                                                                                                         |
| USAGE IS COMP-1                            | 2 バイト                                                                                                                   | 2 バイト                                                                                                                   |
| USAGE IS COMP-2                            | 4 バイト                                                                                                                   | 4 バイト                                                                                                                   |
| USAGE IS COMP-3<br>USAGE IS PACKED-DECIMAL | 1 バイト/2 桁                                                                                                               | 1 バイト/2 桁                                                                                                               |
| USAGE IS INDEX                             | 16 バイト<br>(出現番号 COMP-5 PIC S9(18)<br>+ 指標値 COMP-5 PIC S9(18))<br>(-Qc オプション指定時, 8 バイト<br>(出現番号 COMP-2<br>+ 指標値 COMP-2)) | 16 バイト<br>(出現番号 COMP-5 PIC S9(18)<br>+ 指標値 COMP-5 PIC S9(18))<br>(-Qc オプション指定時, 8 バイト<br>(出現番号 COMP-2<br>+ 指標値 COMP-2)) |
| USAGE IS COMP-5                            |                                                                                                                         |                                                                                                                         |
|                                            | 1 バイト<br>(-Fa オプション指定時,<br>2 バイト)                                                                                       | 2 バイト                                                                                                                   |
| 1～2 桁                                      |                                                                                                                         |                                                                                                                         |
| 3～4 桁                                      | 2 バイト                                                                                                                   | 2 バイト                                                                                                                   |
| 5～9 桁                                      | 4 バイト                                                                                                                   | 4 バイト                                                                                                                   |
| 10～18 桁                                    | 8 バイト                                                                                                                   | 8 バイト                                                                                                                   |
| USAGE IS POINTER                           | 8 バイト                                                                                                                   | 8 バイト                                                                                                                   |
| USAGE IS BIT                               | 1 ビット                                                                                                                   | 1 ビット                                                                                                                   |
| USAGE IS OBJECT REFERENCE                  | 8 バイト                                                                                                                   | 8 バイト                                                                                                                   |

・データ割付境界合わせ(バウンダリ)

| 項目名                                  | [Enterprise for Windows]             |                                      | [Standard]     |                |
|--------------------------------------|--------------------------------------|--------------------------------------|----------------|----------------|
|                                      | SYNC 句なし                             | SYNC 句あり                             | SYNC 句なし       | SYNC 句あり       |
| 外部 10 進数<br>DISPLAY<br>COMP          | 1 バイト<br>1 バイト<br>(-Fc オプション<br>指定時) | 1 バイト<br>1 バイト<br>(-Fc オプション<br>指定時) | 1 バイト<br>1 バイト | 1 バイト<br>1 バイト |
| 内部 10 進数<br>PACKED-DECIMAL<br>COMP-3 | 1 バイト<br>1 バイト                       | 1 バイト<br>1 バイト                       | 1 バイト<br>1 バイト | 1 バイト<br>1 バイト |
| 2 進数 (1 バイト)<br>BINARY(1～2 桁)        | 1 バイト                                | 1 バイト                                |                |                |



|                                                                         |                                                                                                                                                                               |                                                              |                                                                                                                                                                                   |                                              |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| COMP-5(1～2 桁)<br>COMP(1～2 桁)<br><br>(-Fa オプション指定時,<br>2 進数(2 バイト))      | 1 バイト<br><br>1 バイト<br>(-Fc オプション<br>指定なし)                                                                                                                                     | 1 バイト<br><br>1 バイト<br>(-Fc オプション<br>指定なし)                    |                                                                                                                                                                                   |                                              |
| 2 進数 (2 バイト)<br>BINARY(3～4 桁)<br>COMP-5(3～4 桁)<br>COMP-1<br>COMP(3～4 桁) | 1 バイト<br><br>1 バイト<br><br>1 バイト<br><br>1 バイト                                                                                                                                  | 2 バイト<br><br>2 バイト<br><br>2 バイト<br><br>2 バイト                 | 1 バイト<br><br>1 バイト<br><br>1 バイト<br><br>— — —                                                                                                                                      | 2 バイト<br><br>2 バイト<br><br>2 バイト<br><br>— — — |
| 2 進数 (4 バイト)<br>BINARY(5～9 桁)<br>COMP-5(5～9 桁)<br>COMP-2<br>COMP(5～9 桁) | 1 バイト<br><br>1 バイト<br><br>1 バイト<br><br>1 バイト                                                                                                                                  | 4 バイト<br><br>4 バイト<br><br>4 バイト<br><br>4 バイト                 | 1 バイト<br><br>1 バイト<br><br>1 バイト<br><br>— — —                                                                                                                                      | 4 バイト<br><br>4 バイト<br><br>4 バイト<br><br>— — — |
| 2 進数 (8 バイト)<br>BINARY(10～18 桁)<br>COMP-5(10～18 桁)<br>COMP(10～18 桁)     | 1 バイト<br><br>1 バイト<br><br>1 バイト                                                                                                                                               | 8 バイト<br><br>8 バイト<br><br>8 バイト<br>(-Fa オプション<br>指定時, 4 バイト) | 1 バイト<br><br>1 バイト<br><br>— — —                                                                                                                                                   | 8 バイト<br><br>8 バイト<br><br>— — —              |
| INDEX                                                                   | 1 バイト                                                                                                                                                                         | 指定不可                                                         | 1 バイト                                                                                                                                                                             | 指定不可                                         |
| POINTER                                                                 | 8 バイト                                                                                                                                                                         | 指定不可                                                         | 8 バイト                                                                                                                                                                             | 指定不可                                         |
| BIT                                                                     | ・基本ビットデータ項目<br>であって同じレベルの<br>基本ビットデータ項目<br>やビット集団項目の直<br>後にあるもの、および<br>ビット集団項目であっ<br>て同じレベルのビット<br>集団項目や基本ビット<br>データ項目の直後に<br>あるものは、1 ビット<br>・それ以外は、1 バイト<br>・01/77 のときは 8 バイ | 1 バイト<br>01/77 のときは<br>8 バイト                                 | ・基本ビットデータ項目<br>であって同じレベルの<br>基本ビットデータ項目<br>やビット集団項目の直<br>後にあるもの、および<br>ビット集団項目であっ<br>て同じレベルのビット<br>集団項目や基本ビット<br>データ項目の直後に<br>あるものは、1<br>ビット<br>・それ以外は、1 バイト<br>・01/77 のときは 8 バイト | 1 バイト<br>01/77 のときは<br>8 バイト                 |

|                                                    |       |       |       |       |
|----------------------------------------------------|-------|-------|-------|-------|
|                                                    | ト     |       |       |       |
| OBJECT REFERENCE                                   | 8 バイト | 8 バイト | 8 バイト | 8 バイト |
| OBJECT REFERENCE<br>[ FACTORY OF ]<br>ACTIVE-CLASS | 8 バイト | 8 バイト | 8 バイト | 8 バイト |
| OBJECT REFERENCE<br>[ FACTORY OF ] クラス名 1          | 8 バイト | 8 バイト | 8 バイト | 8 バイト |

## 19.2 Linux 版の場合

[Enterprise for Linux] 本機能は、Enterprise Edition for Linux 版において、使用できる。

### 19.2.1 アプリケーションの生成と実行の環境

本コンパイラの動作 OS と生成可能なアプリケーションを次に示す。

|        |                    |
|--------|--------------------|
| 開発 OS  | 生成アプリケーション         |
| 64 ビット | 64 ビットアプリケーション(*1) |

(\*1) 関連製品との組み合わせが必要な機能に制限がある。詳細は「第 21 章 関連製品連携」参照。

本コンパイラで生成した 64 ビットアプリケーションは、実行マシンが 64 ビット OS であれば動作可能である。

なお、COBOL アプリケーションの中から他言語で作成したプログラムを呼び出している場合には、そのプログラムも COBOL アプリケーションに合わせて 64 ビットアプリケーションでなければならない。これは、他言語から COBOL アプリケーションを呼び出す場合の他言語についても同様である。

### 19.2.2 翻訳環境の設定

アプリケーションを生成するには、端末上で COBOL コマンドを起動して生成する。

### 19.2.3 コマンドの指定方法

本コンパイラでアプリケーションを生成するには、端末で COBOL コンパイラを実行する。

COBOL ソースファイル名(メインプログラム)が、「sample.cob」である場合の例を次に記載する。

・64 ビットアプリケーションを生成する場合

> cob -M sample.cob

### 19.2.4 64 ビットでの言語仕様

64 ビットでの、言語仕様を示す。

・項目長

| 用途               | 64 ビット                           |
|------------------|----------------------------------|
| USAGE IS DISPLAY |                                  |
| 英字項目・英数字項目       | 1 バイト/1 文字                       |
| 外部 10 進項目        | 1 バイト/1 桁                        |
| 数字編集項目           | 1 バイト/1 桁                        |
| 英数字編集項目          | 1 バイト/1 桁                        |
| 日本語項目            | 2 バイト/1 文字                       |
| 日本語編集項目          | 2 バイト/1 文字                       |
| USAGE IS BINARY  |                                  |
| 1～2 桁            | 1 バイト<br>(-Fa オプション指定時<br>2 バイト) |
| 3～4 桁            | 2 バイト                            |
| 5～9 桁            | 4 バイト                            |
| 10～18 桁          | 8 バイト                            |
| USAGE IS COMP    | USAGE IS BINARY<br>(-Fc オプション指定時 |

|                                            |                                                                                                                               |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
|                                            | DISPLAY)                                                                                                                      |
| USAGE IS COMP-1                            | 2 バイト                                                                                                                         |
| USAGE IS COMP-2                            | 4 バイト                                                                                                                         |
| USAGE IS COMP-3<br>USAGE IS PACKED-DECIMAL | 1 バイト/2 桁                                                                                                                     |
| USAGE IS INDEX                             | 16 バイト<br>(出現番号 COMP-5 PIC<br>S9(18)<br>+指標値 COMP-5 PIC<br>S9(18))<br>(-Qc オプション指定時<br>8 バイト<br>(出現番号 COMP-2<br>+指標値 COMP-2)) |
| USAGE IS COMP-5                            |                                                                                                                               |
| 1～2 桁                                      | 1 バイト<br>(-Fa オプション指定時<br>2 バイト)                                                                                              |
| 3～4 桁                                      | 2 バイト                                                                                                                         |
| 5～9 桁                                      | 4 バイト                                                                                                                         |
| 10～18 桁                                    | 8 バイト                                                                                                                         |
| USAGE IS POINTER                           | 8 バイト                                                                                                                         |
| USAGE IS BIT                               | 1 ビット                                                                                                                         |
| USAGE IS OBJECT REFERENCE                  | 8 バイト                                                                                                                         |

## ・データ割付境界合わせ(バウンダリ)

| 項目名                                            | 64 ビット   |          |
|------------------------------------------------|----------|----------|
|                                                | SYNC 句なし | SYNC 句あり |
| 外部 10 進数 DISPLAY                               | 1 バイト    | 1 バイト    |
| COMP                                           |          |          |
| 内部 10 進数<br>PACKED-DECIMAL COMP-3              | 1 バイト    | 1 バイト    |
| 2 進数 (1 バイト)<br>BINARY(1～2 桁)<br>COMP-5(1～2 桁) | 1 バイト    | 1 バイト    |

|                                                                  |                                                                                     |                                     |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------|
| COMP(1～2 桁)                                                      | 1 バイト                                                                               | 1 バイト                               |
| 2 進数 (2 バイト)<br>(-Fa オプション指定時)<br>BINARY(1～2 桁)<br>COMP-5(1～2 桁) | 1 バイト                                                                               | 2 バイト                               |
| 2 進数 (2 バイト)<br>BINARY(3～4 桁)<br>COMP-1                          | 1 バイト                                                                               | 2 バイト                               |
| COMP(3～4 桁)                                                      | 1 バイト                                                                               | 2 バイト                               |
| COMP-5(3～4 桁)                                                    | 1 バイト                                                                               | 2 バイト                               |
| 2 進数 (4 バイト)<br>BINARY(5～9 桁)<br>COMP-2                          | 1 バイト                                                                               | 4 バイト                               |
| COMP(5～9 桁)                                                      | 1 バイト                                                                               | 4 バイト                               |
| COMP-5(5～9 桁)                                                    | 1 バイト                                                                               | 4 バイト                               |
| 2 進数 (8 バイト)<br>BINARY(10～18 桁)                                  | 1 バイト                                                                               | 8 バイト<br>(-Fa オプション<br>指定時 4 バイト)   |
| COMP(10～18 桁)                                                    | 1 バイト                                                                               | 8 バイト<br>(-Fa オプション<br>指定時 4 バイト)   |
| COMP-5(10～18 桁)                                                  | 1 バイト                                                                               | 8 バイト<br>(-Fa オプション<br>指定時 4 バイト)   |
| INDEX                                                            | 1 バイト                                                                               | 指定不可                                |
| POINTER                                                          | 8 バイト<br>(-Fb オプション指<br>定時, 1 バイト)                                                  | 8 バイト<br>(-Fb オプション<br>指定時のみ可<br>能) |
| BIT                                                              | 基本ビットデータ<br>項目であって同じ<br>レベルの基本ビット<br>データ項目やビット<br>集団項目の直後<br>にあるもの, および<br>ビット集団項目で | 1 バイト<br>01/77 のときは 8<br>バイト        |

|                                                    |                                                                                                        |       |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------|
|                                                    | あって同じレベル<br>のビット集団項目<br>や基本ビットデー<br>タ項目の直後にあ<br>るものは、1 ビット<br>それ以外は、1 バイ<br>ット<br>01/77 のときは 8 バ<br>イト |       |
| OBJECT REFERENCE                                   | 8 バイト                                                                                                  | 8 バイト |
| OBJECT REFERENCE<br>[ FACTORY OF ]<br>ACTIVE-CLASS | 8 バイト                                                                                                  | 8 バイト |
| OBJECT REFERENCE<br>[ FACTORY OF ] クラス名 1          | 8 バイト                                                                                                  | 8 バイト |

### 19.2.5 注意事項

データポインタデータ項目の ACOS-4 互換オプション: -Fb

各オプションについての詳細は、「F.1 翻訳オプションによる仕様差異」を参照。

## 第20章 マルチスレッドプログラミング

【Enterprise】本機能は Enterprise Edition for Windows/Enterprise Edition for Linux 版において、使用できる。

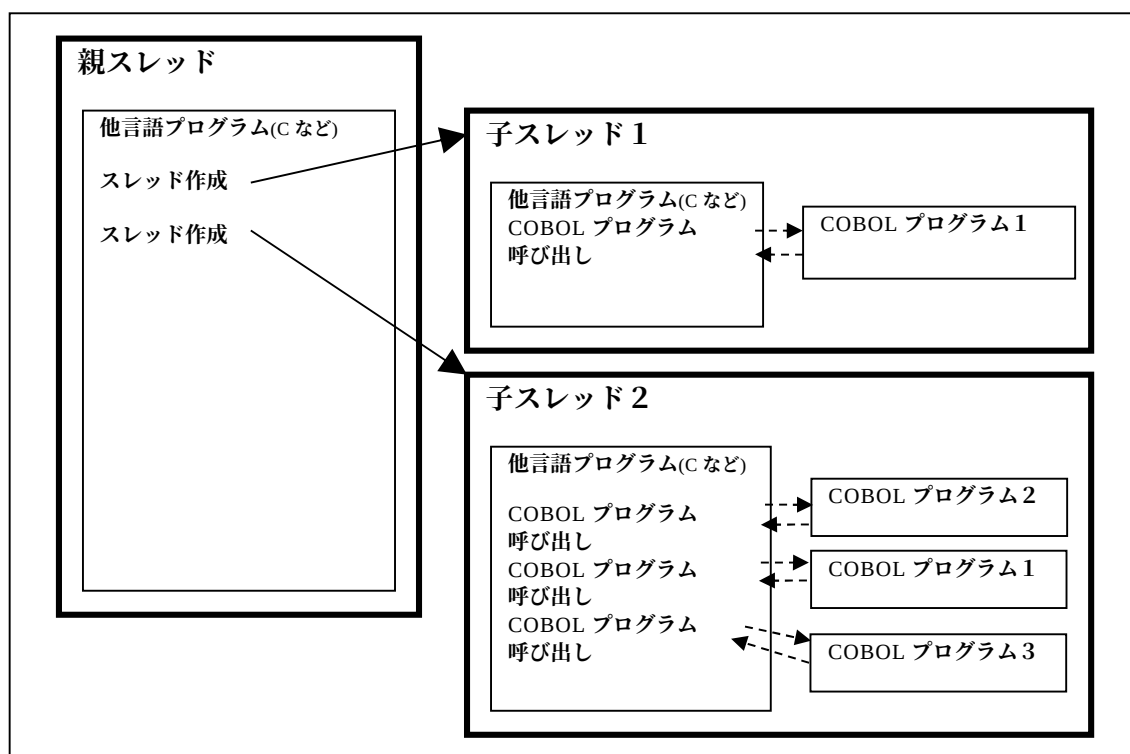
1つのプロセスの中で複数のスレッドを並列に処理することにより、アプリケーションの性能向上やシステムリソースの有効利用を実現できる。ただし、同じプロセスの他スレッドと共有する可能性があるリソースを認識し、慎重にプログラミングする必要がある。本章では、本コンパイラが提供するマルチスレッド機能の提供範囲や注意事項について説明する。

### 20.1 COBOL の提供するマルチスレッド機能

本 COBOL のマルチスレッド機能は COBOL プログラムがマルチスレッド配下で動作することを保証する。COBOL 言語によるスレッドの作成や終了、同期機構を提供するものではない。

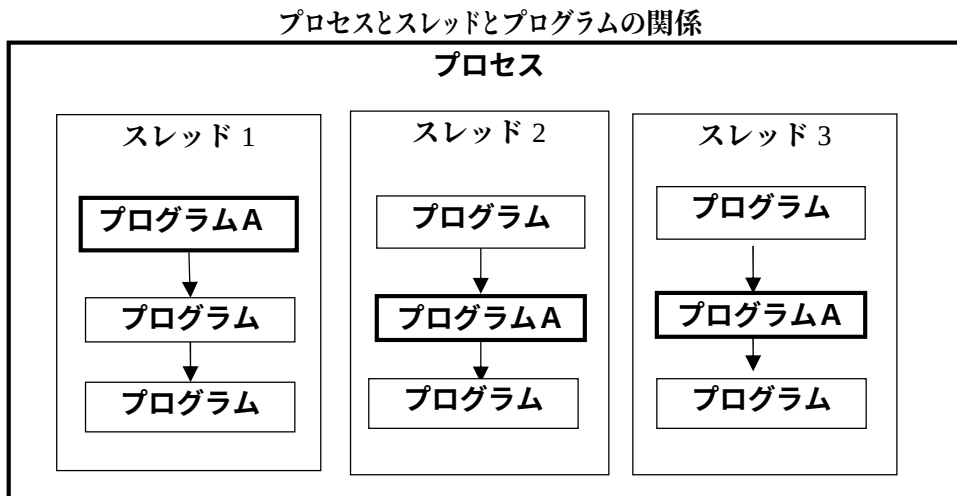
C などの他言語プログラムやシステムを構成する実行基盤が作成したスレッドから COBOL プログラムを呼び出した時、そのスレッド上で COBOL プログラムは動作し、処理が終了すると呼び出し元に制御を戻す。

COBOL プログラムがマルチスレッド配下で動作する場合の典型的な利用イメージを次に示す。



マルチスレッド機能を使用する時、プロセスとスレッドとプログラムの関係について理解する必要がある。

プロセスはオペレーションシステムからCPUやメモリ空間などの割り当てを受けて動作する実行単位である。プロセスは1つ以上のスレッドで構成され、複数のスレッドを同一メモリ空間上で実行する。スレッドはCPU利用の単位であり、それぞれ一連のプログラムを順に実行する。



各スレッドは、同一メモリ空間上で実行されるため、マルチスレッドプログラムを作成する場合には、スレッド間でデータが共有されることを考慮する必要がある。本 COBOL では、この図のプロセスとスレッドとプログラムの関係に示すプロセスは言語説明書の用語“実行単位”、この図のプロセスとスレッドとプログラムの関係に示すプログラムは“実行時要素”（プログラムや関数やメソッド）に相当する。共有する可能性がある COBOL データの一部を次に記す。

- ・ 作業場所節に定義したデータ  
実行時要素に1つだけ領域を確保し、実行時要素の終了後も値を保持する。前図では、スレッド1のプログラムAが終了後、スレッド2のプログラムAが開始したとすると、スレッド1の処理結果をスレッド2が参照できる。
- ・ 外部データ  
実行単位（プロセス）に1つだけ領域を確保し、任意の実行時要素からアクセスできる。前図では、スレッド1, 2, 3のすべてのプログラムからアクセスできる。
- ・ 外部ファイル  
実行単位（プロセス）に1つだけ管理用領域を確保し、任意の実行時要素から読み込み、書き込みが可能である。

COBOL データについて詳細は「20.4.1 COBOL データの有効範囲と生存期間」を参照。

## 20.2 アプリケーションの生成と実行の環境

マルチスレッド機能は-MT オプションを指定することにより有効となる。  
-MT オプションを指定すると、コンパイラは、命令コードおよびコンパイラ自体が生成する命令コードやデータを、マルチスレッドで動作させても支障が出ないようにアプリケーションを生成する。  
本コンパイラの開発 OS と生成可能なアプリケーションを次に示す。

| 動作 OS  | オプション | 生成アプリケーション            |
|--------|-------|-----------------------|
| 64 ビット | -MT   | 64 ビットマルチスレッドアプリケーション |

関連製品との組み合わせが必要な機能に制限がある。詳細は「第 21 章 関連製品連携」を



参照。

なお、-MT オプションを指定して翻訳されたアプリケーションを、シングルスレッドで動作させることはとくに支障はない。

## 20.3 用語の説明

本章で使用する用語を説明する。

| 用語      | 説明                                                                 |
|---------|--------------------------------------------------------------------|
| メインスレッド | プログラム実行により作成されるスレッドを指す。COBOL がサポートするアプリケーション構成では、スレッドを作成するスレッドを指す。 |
| サブスレッド  | メインスレッドより作成されたスレッドを指す。                                             |
| 外部ファイル  | ファイル管理記述項またはファイル記述項に EXTERNAL 句の記述があるファイルを指す。                      |
| 内部ファイル  | ファイル管理記述項またはファイル記述項に EXTERNAL 句の記述がないファイルを指す。                      |

## 20.4 マルチスレッドプログラミングの留意事項

### 20.4.1 COBOL データの有効範囲と生存期間

複数のスレッドが1つのデータに対して同時にアクセスすることにより知らない間にデータの整合性を破壊することがないように、COBOL データの有効範囲と生存期間を理解する必要がある。

データの有効範囲は次のように分類できる。

外部項目 : 実行単位中の任意の実行時要素がアクセスできる。

内部項目 : データを記述している実行時要素のみアクセスできる。

生存期間は次のように分類できる。

プロセス単位: プロセス起動時に領域確保。プロセス終了時に破棄。

スレッド単位: スレッド作成 (アタッチ) 時に領域確保。

スレッド終了 (デタッチ) 時に破棄。

呼び出し単位: プログラムや関数やメソッドの呼び出し時に領域確保。

呼び出し元に戻る時に破棄。

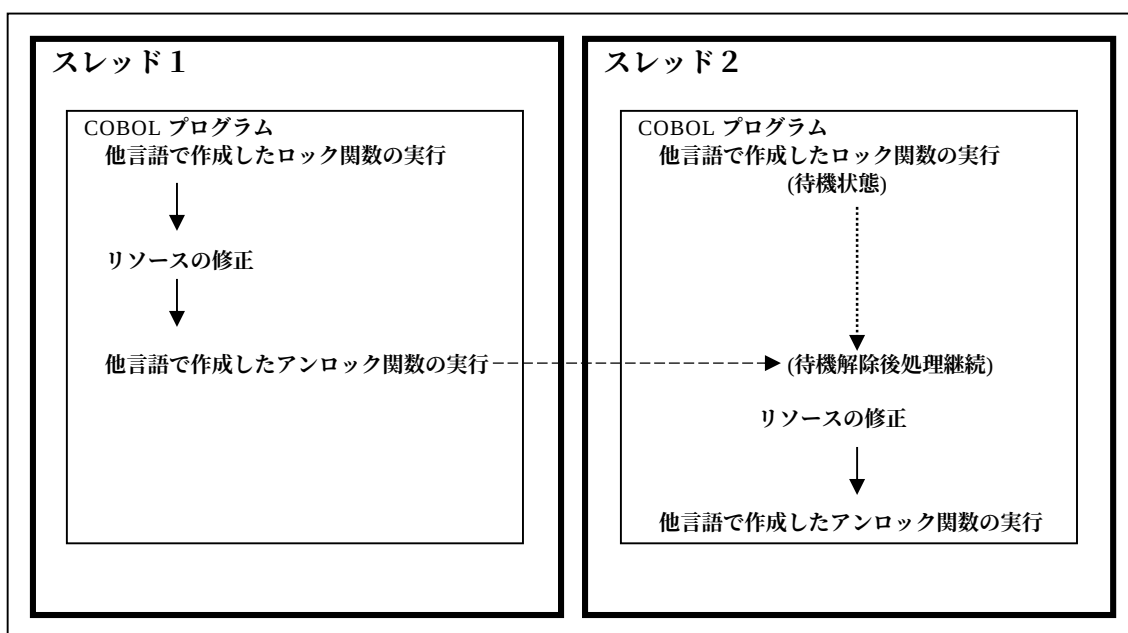
内部項目となるデータは、さらに自動項目、静的項目に分類できる。自動項目の生存期間は“呼び出し単位”となり、静的項目の生存期間は“プロセス単位”となる。

生存期間が“プロセス単位”となるデータを使用する場合、複数スレッドからの同時アクセスの可能性について十分に確認を行い、必要に応じて次のような処置を検討すること。

- 1つのデータが同時にアクセスされないようアプリケーションを構成する。  
たとえば、同一ファイルを複数スレッドからアクセスする処理は避け、ある特定のスレッドからのみアクセスするよう設計する (ログ出力スレッド、データベースアクセススレッドなど)。
- 作業場所節のデータではなく局所記憶節のデータを利用する。  
計算時の一時作業用データ等は生存期間を“呼び出し単位”に変更することで、他スレ

ッドの影響を完全になくすることができる。ただし、プログラムや関数やメソッドの処理が終了するとデータは破棄され、保持できない。呼び出し後も値を保持しておく必要があるデータにはこの処置は不適切である。

- C など他言語の同期機構を使用して別々のスレッドからのアクセスを制御する。  
COBOL から他言語の関数を呼び出すことができるため、必要に応じて同期制御関数を他言語で作成し、必要な箇所で実行する。たとえば、ミューテックスを使用したロック関数とアンロック関数を用意する。データの整合性を保つために、ロック関数を実行後、リソースを修正し、その後、アンロック関数を実行する。あるスレッドがデータを修正している間は、他スレッドのロック関数は待機状態になり、修正したスレッドがアンロック関数を実行すると、待機状態のスレッドは処理を継続できる。



次に COBOL データの有効範囲と生存期間を示す。

表 20-1 COBOL データの有効範囲と生存期間

| COBOL データ                |                                                     | マルチスレッド動作可能なアプリケーション<br>(-MT 指定あり) |        | シングルスレッドのみ動作可能なアプリケーション<br>(-MT 指定なし) |                |
|--------------------------|-----------------------------------------------------|------------------------------------|--------|---------------------------------------|----------------|
|                          |                                                     | 有効範囲                               | 生存期間   | 有効範囲                                  | 生存期間           |
| 作業場所節に記述されたデータ項目         | EXTERNAL 句の記述があるデータ(*1)                             | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
|                          | EXTERNAL 句の記述がないデータ                                 | 内部項目<br>静的                         | プロセス単位 | 内部項目<br>静的                            | プロセス単位         |
| 局所記憶節に記述されたデータ項目         |                                                     | 内部項目<br>自動                         | 呼び出し単位 | 内部項目<br>自動                            | 呼び出し単位         |
| 連絡節に記述されたデータ項目           |                                                     | 内部項目<br>自動                         | 呼び出し単位 | 内部項目<br>自動<br>(*2)                    | 呼び出し単位<br>(*2) |
| ベースデータ節に記述されたポインタ名(*6)   |                                                     | 内部項目<br>自動                         | 呼び出し単位 | 内部項目<br>自動<br>(*3)                    | 呼び出し単位<br>(*3) |
| ファイル                     | ファイル記述項に EXTERNAL 句の記述があるファイル結合子とレコード領域             | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
|                          | ファイル管理記述項に EXTERNAL 句の記述があるファイル結合子(*7)              | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
|                          | ファイル記述項およびファイル管理記述項に EXTERNAL 句の記述がないファイル結合子とレコード領域 | 内部項目<br>静的                         | プロセス単位 | 内部項目<br>静的                            | プロセス単位         |
| 特殊レジスタ EXCEPTION-CODE    |                                                     | 内部項目<br>(*4)                       | スレッド単位 | 内部項目<br>(*4)                          | プロセス単位         |
| 特殊レジスタ EXCEPTION-LINE    |                                                     | 内部項目<br>(*4)                       | スレッド単位 | 内部項目<br>(*4)                          | プロセス単位         |
| 特殊レジスタ COMPLETION-CODE   |                                                     | 外部項目                               | スレッド単位 | 外部項目                                  | プロセス単位         |
| 特殊レジスタ<br>LINAGE-COUNTER | ファイル記述項に EXTERNAL 句の記述がある場合                         | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
|                          | ファイル記述項に EXTERNAL 句の記述がない場合                         | 内部項目<br>静的                         | プロセス単位 | 内部項目<br>静的                            | プロセス単位         |
| システム入力ファイルとシステム出力ファイル    |                                                     | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
| 外部スイッチ                   |                                                     | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
| 利用者制御変数                  |                                                     | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
| クラス                      | ファクトリオブジェクト                                         | 外部項目                               | プロセス単位 | 外部項目                                  | プロセス単位         |
|                          | インスタンスオブジェクト                                        | (*5)                               | (*5)   | (*5)                                  | (*5)           |

(\*1) 翻訳時に定義/参照オプション(-Sx/-Si オプション)の指定がない場合、exe ファイルや dll ファイルなど実行ファイル毎に領域を確保するため、有効範囲は実行ファイル内となる。  
-Sx/-Si オプションを指定した場合、外部データの領域は実行単位でひとつだけ確保するため、実行ファイル間で情報を受け渡すことが可能となる。しかし、同時アクセスの可能性が高くなるためデータの整合性確保の設計は十分に行う必要がある。

(\*2) 連絡節に GLOBAL 句の記述を可能とするオプション(-Qa オプション)指定時、有効範囲は“静的項目”、生存期間は“プロセス単位”となる。-Qa オプションはマルチスレッド機能と同時指定

できない。

- (\*3) ベースデータ節に GLOBAL 句の記述を可能とするオプション(-Qb オプション)指定時、‘静的項目’；生存期間は‘プロセス単位’となる。-Qb オプションはマルチスレッド機能と同時指定できない。
- (\*4) 例外処理節のみで参照可能。
- (\*5) 有効範囲はその実行インスタンスのオブジェクト参照値を格納したオブジェクト参照データ項目の有効範囲となる。オブジェクトの生存期間は、その生成で始まり、削除の時点で終わる。
- (\*6) ベースデータ節に記述されたデータ項目については、ポインタによって関係づけるデータ領域の生存期間を意識する必要がある。
- (\*7) ファイル管理記述項に EXTERNAL 句の記述があるレコード領域は、ファイル記述項およびファイル管理記述項に EXTERNAL 句の記述がないファイル結合子とレコード領域と等しく、有効範囲は内部項目・静的となる。

## 20.4.2 マルチスレッドオプション

マルチスレッド機能を使用する場合、-MT オプションを指定して COBOL ソースを翻訳する。

同じ実行単位内では、-MT オプションを指定して翻訳したプログラムと指定しないで翻訳したプログラムを混在することはできない。

マルチスレッド機能を使用すると、内部データ配置の変更などにより性能に悪影響を与える可能性がある。また、一部機能を使用することができない。次に機能差異を記す。

| 機能                                    | マルチスレッド動作可能なアプリケーション<br>(-MT オプション指定あり)                                                                        | シングルスレッドのみ動作可能なアプリケーション<br>(-MT オプション指定なし)                                                   |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| プログラム名段落                              | RECURSIVE 指定あり(暗黙的に RECURSIVE)とみなす。                                                                            | COBOL プログラムの記述どおりの指定となる。                                                                     |
| GLOBAL 句                              | 局所記憶節、連絡節、ベースデータ節に GLOBAL 句を記述すると翻訳エラーとなる。                                                                     | 局所記憶節、連絡節、ベースデータ節に GLOBAL 句の記述不可。ただし、翻訳時に -Qa, -Qb オプション指定ありの場合、連絡節・ベースデータ節に GLOBAL 句の記述が可能。 |
| CALL 一意名                              | 翻訳エラーとなる。                                                                                                      | 制限なし。                                                                                        |
| CANCEL 文                              | 翻訳エラーとなる。                                                                                                      | 制限なし。                                                                                        |
| STOP 文                                | STOP RUN 文、および、-FI オプション指定時に使用可能な STOP ABORT 文は、翻訳エラーとなる。ただし、メインプログラムの場合は STOP RUN 文/STOP ABORT 文は翻訳エラーにはならない。 | 制限なし。                                                                                        |
| 実行時エラー発生時の動作                          | 構造化例外をスローする。<br>「20.4.3 実行時エラー発生時の動作」参照。                                                                       | プロセスが終了する。                                                                                   |
| 実行時エラーメッセージの一時停止解除機能(環境変数 COB_RUNERR) | 一時停止しない。「20.4.4 環境変数」を参照。                                                                                      | 環境変数 COB_RUNERR に NO_PAUSE 指定時にかぎり、一時停止しない。通常は実行時メッセージを出力後、一時停止しリターンキーの入力を待                  |

|          |           |       |
|----------|-----------|-------|
|          |           | つ。    |
| 相対ファイル   | 翻訳エラーとなる。 | 制限なし。 |
| 索引ファイル   | 翻訳エラーとなる。 | 制限なし。 |
| RAF ファイル | 翻訳エラーとなる。 | 制限なし。 |

### 20.4.3 実行時エラー発生時の動作

マルチスレッド動作可能なアプリケーション(-MT オプション指定あり)において、メインスレッドで動作するプログラムで実行時エラーが発生した場合、シングルスレッドのみ動作可能なアプリケーション(-MT オプション指定なし)と同じく、実行時エラーを出力してプロセスが終了する。サブスレッドで動作するプログラムで実行時エラーが発生した場合、プロセス終了もスレッド終了も行わず、Windows の構造化例外をスローする。実行時エラーが発生したスレッドは速やかに終了処理を実施しスレッドを終了すること。COBOL が発行する構造化例外コードは0 (ゼロ) である。

### 20.4.4 環境変数

環境変数はプロセス単位で使用可能である。設定値は全スレッドに同一の影響を与える。特にファイル名を指定する環境変数 COB\_SYSIN, COB\_SYSOUT においては複数スレッドから参照できてしまうため、結果的に同一ファイルを利用することになる可能性があるので注意すること。

マルチスレッド動作可能なアプリケーション(-MT オプション指定あり)において、環境変数 COB\_RUNERR 設定なしの場合、実行時エラーの動作が異なる。メインスレッドで動作する COBOL プログラムの場合はエラーメッセージ表示後リターンキーの入力待ちとなる(シングルスレッドのみ動作可能なアプリケーションと同じ)。サブスレッドで動作する COBOL プログラムの場合はエラーメッセージ表示後リターンキーの入力待ちを行わない(COB\_RUNERR=NO\_PAUSE と同じ)。

| 環境変数                  | メインスレッドで動作している場合             | サブスレッドで動作している場合              |
|-----------------------|------------------------------|------------------------------|
| COB_RUNERR 設定なし       | エラーメッセージ表示後、リターンキーの入力待ちとなる   | エラーメッセージ表示後、リターンキーの入力待ちを行わない |
| COB_RUNERR=NO_PAUSE   | エラーメッセージ表示後、リターンキーの入力待ちを行わない | エラーメッセージ表示後、リターンキーの入力待ちを行わない |
| COB_RUNERR=NO_DISPLAY | エラーメッセージの表示とリターンキーの入力待ちを行わない | エラーメッセージの表示とリターンキーの入力待ちを行わない |

### 20.4.5 システムサブルーチン

本コンパイラが提供しているシステムサブルーチンのうち、マルチスレッド機能を利用する場合に注意が必要なシステムサブルーチンは、環境変数に関するシステムサブルーチンである。

環境変数の取り込み(B\_GETENV)は環境変数への設定(B\_PUTENV)とセットで利用されることを想定している。複数スレッドからの利用は可能であるが、プロセス単位で1つの環境変数を操作することに注意すること。

## 20.4.6 ファイル

[Enterprise] -MT オプション指定ありの場合、ファイルの排他／共用はサポートしない。

### 20.4.6.1 外部ファイルの操作

マルチスレッド機能を使用する場合、外部ファイルの OPEN/CLOSE は、必ずメインスレッドで行う必要がある。OPEN はスレッド作成前に行い、作成したスレッドの終了を待ち合わせた後に CLOSE を行うこと。

- メインプログラムが COBOL の場合  
外部ファイルの OPEN および CLOSE 処理は、必ずメインプログラムで行うこと。
- メインプログラムが他言語の場合  
外部ファイルの OPEN を行う COBOL プログラム、CLOSE を行う COBOL プログラムをそれぞれ用意し、メインプログラムから呼び出すこと。

外部ファイルのステータスはプロセスで一意となる為、複数スレッドから同一ファイルに対する処理を行う場合は注意が必要である。

ファイルのステータスが正常・異常にかかわらず、メインプログラムでは最後に外部ファイルを必ず CLOSE すること。

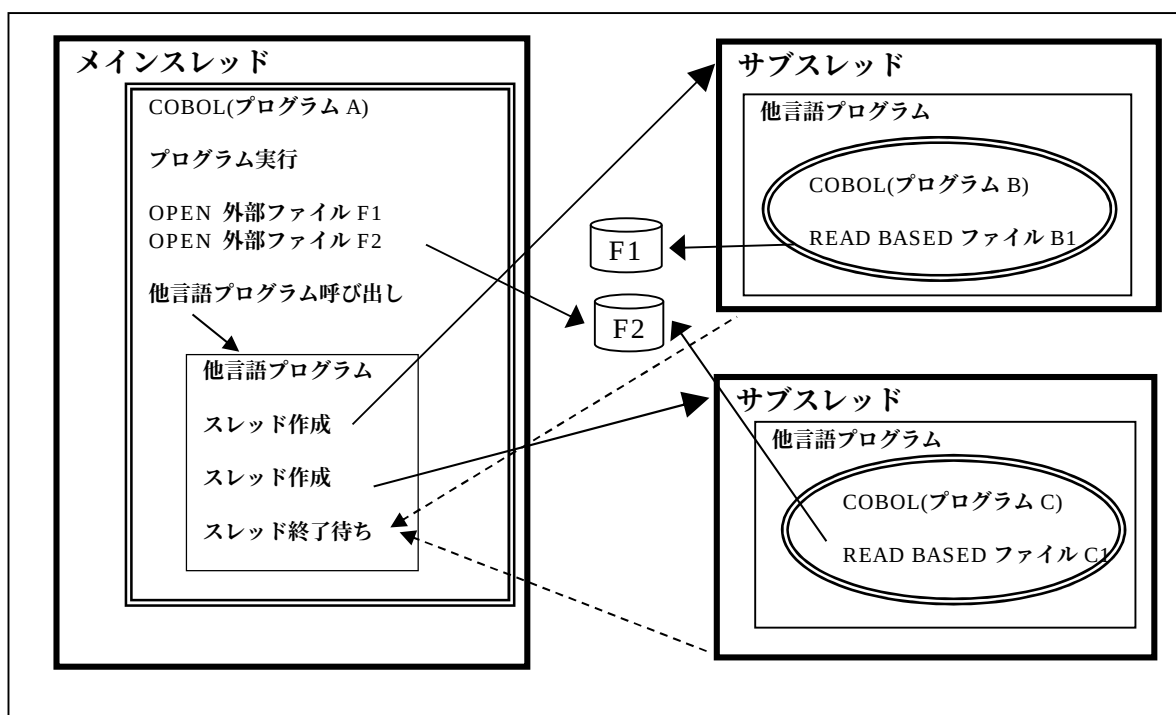
### 20.4.6.2 内部ファイルの操作

内部ファイルの OPEN/CLOSE は、必ず実行時要素内で行う必要がある。仮に OPEN されたままとなった場合、COBOL で暗黙に CLOSE する事は無く、プロセス終了時まで OPEN 状態となる。FILE STATUS 句によるファイル I/O の結果確認や取り扱いデータ値の妥当性確認、例外節の組み込みなどを利用してエラー処理を適切に実装し、正常時だけでなく異常時の終了処理としても内部ファイルを CLOSE すること。

### 20.4.7 BASED 指定ファイルの使用

複数のスレッドから BASED 指定ファイルを使用して外部ファイルにアクセスする場合、スレッド毎に別のサブプログラムを使用してアクセスすること（同一サブプログラムを複数スレッドで使用してはならない）。

●外部ファイル F1 をサブスレッド（プログラム B）から READ、外部ファイル F2 をプログラム C から READ する場合の例（F1, F2 は同一ファイルでもよい）



### 20.4.8 ポインタ

関係づけるデータ領域の生存期間について注意すること。

たとえば、ベースデータ節に記述したデータ項目に局所記憶節のデータ項目を関係づけた場合、生存期間が“呼び出し単位”となるので複数スレッドからの同時アクセスの可能性はない。一方、ベースデータ節に記述したデータ項目に作業場所節のデータ項目を関係づけた場合、生存期間が“プロセス単位”となるので複数スレッドからの同時アクセスの可能性はあるため、排他制御処理を組み込むなど検討が必要である。

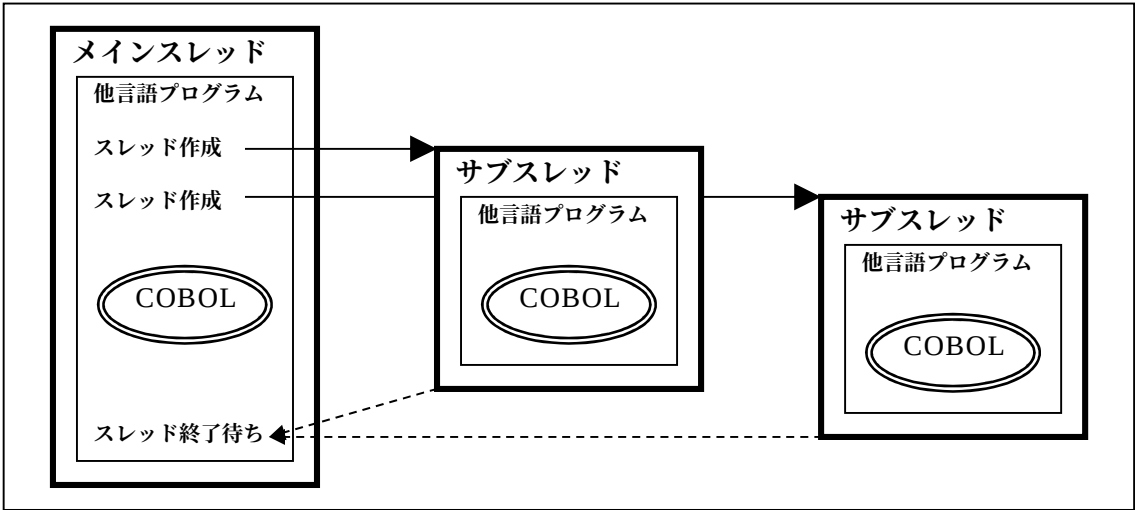
## 20.5 マルチスレッド対応時の他言語連携の留意事項

COBOL はスレッド作成機能を提供していないため、マルチスレッド機能を使用する場合、他言語連携は必須となる。COBOL と他言語の連携について「7.12 C/C++プログラムとの連絡」および「7.13 C/C++プログラムとのリンク時の注意事項」も合わせて参照。

### 20.5.1 プログラム構成

COBOL プログラムと他言語プログラムの組合せに関して、次の2つの構成に対しマルチスレッド機能をサポートする。

- 他言語プログラムやシステム実行基盤が作成したスレッドから COBOL プログラムを実行する構成



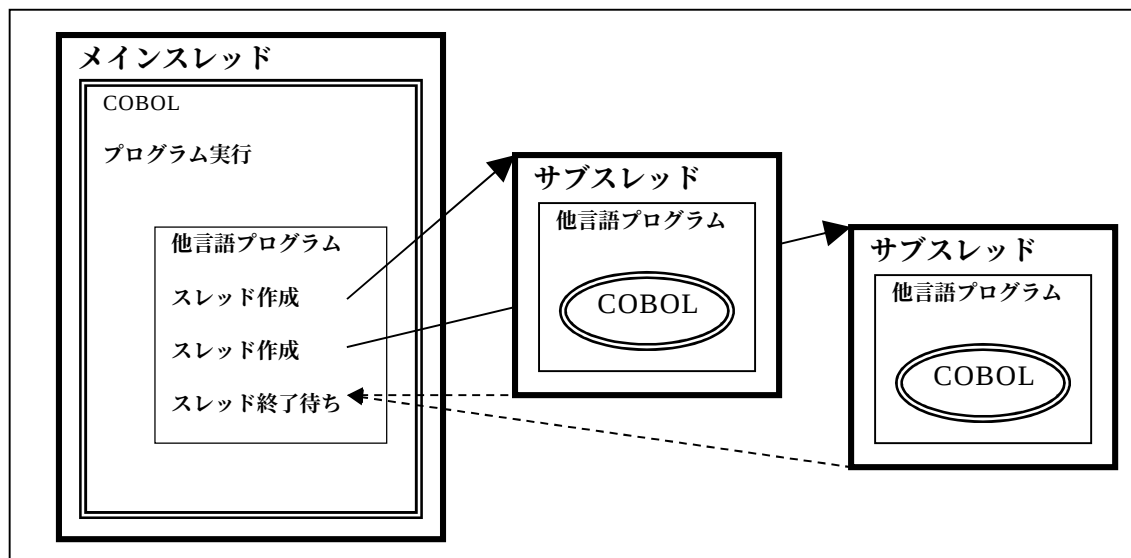
メインスレッドとなる他言語プログラムにてスレッドを作成する。スレッドの作成関数の引数に COBOL プログラムを指定してはならない。サブスレッドは他言語プログラムから始まり、COBOL が提供する初期処理関数と終了処理関数を実行する必要がある。マルチスレッド機能として COBOL が提供する関数を記す。詳細は「20.5.3 サブスレッドから COBOL プログラムを実行する場合」を参照。

| 提供関数            | 機能                   | 参照先                        |
|-----------------|----------------------|----------------------------|
| CSCHKCOBRTPINIT | サブスレッド初期処理           | (3) [CSCHKCOBRTPINIT 利用規約] |
| CSGETTHREADCODE | COMPLETION-CODE 値の取得 | (4) [CSGETTHREADCODE 利用規約] |
| CSFINISH_MT     | サブスレッド終了処理           | (5) [CSFINISH_MT 利用規約]     |
| CSGETCOBERR     | エラー情報の取得             | (6) [CSGETCOBERR 利用規約]     |

メインスレッドおよびサブスレッドで COBOL プログラムを実行することができる。メインスレッドで COBOL プログラムを実行する場合の注意事項については「7.13.3 COBOL がサブプログラムの場合」を参照。サブスレッドから COBOL プログラムを実行する場合の注意事項については「20.5.3 サブスレッドから COBOL プログラムを実行する場合」を参照。



●COBOL のメインプログラムから他言語でスレッドを作成し、サブスレッドから COBOL プログラムを実行する構成



メインスレッドとなる COBOL プログラムからスレッドを作成する他言語プログラムを実行する。他言語プログラムはスレッドの作成後、すべてのスレッドの終了を待ち、COBOL プログラムに制御を戻す必要がある。複数スレッド実行中にメインスレッドで COBOL プログラムを実行してはいけない。メインスレッドで他言語プログラムを実行する場合の注意事項について「7.13.2 COBOL がメインプログラムの場合」を参照。また、「20.5.2 COBOL プログラムからのスレッド作成」を合わせて参照。サブスレッドで COBOL プログラムを実行する場合の注意事項については「20.5.3 サブスレッドから COBOL プログラムを実行する場合」を参照。

## 20.5.2 COBOL プログラムからのスレッド作成

スレッド作成機能を持つ他言語プログラムを呼び出すことで、COBOL プログラムが動作するスレッドの子スレッドを作成することが可能である。このとき、COBOL プログラムは子スレッドに関して一切関知しない。たとえば、親スレッドの COBOL プログラムが終了する際、子スレッドも同時に終了させるような機能はないため、(再度他言語プログラムを介して) 子スレッドを終了させる必要がある。

COBOL メインプログラムは、作成したスレッドと並列処理を行ってはならない。スレッド作成用の他言語プログラムでは、必要な数だけスレッドを作成した後、速やかに WAIT 処理を行い、作成したスレッドの終了を待ち合わせる必要がある。スレッド作成を行う他言語プログラムには、COBOL メインプログラムに対して異常を表すステータスを返却できる仕組みを盛り込むこと。

C などの他言語モジュール側終了処理を登録する CSSETREGIST 関数はサブスレッドには影響しない。CSSETREGIST 関数で登録された関数はメインスレッドの COBOL メインプログラム終了時に実行する。また、CSSETREGIST 関数はメインスレッドから実行すること。サブスレッドから実行した場合、アプリケーションの動作は保証できない。

### 20.5.3 サブスレッドから COBOL プログラムを実行する場合

サブスレッドから COBOL プログラムを実行する場合、COBOL が提供する初期処理関数と終了処理関数をサブスレッド上で呼び出す必要がある。

次にサンプルコードを示す。C メインプログラムは 2 つのスレッドを作成し、各スレッドから COBOL プログラムを実行する。同期制御用のロック関数とアンロック関数を定義する。COBOL プログラムは、ロック関数を実行し、外部データを操作後、アンロック関数を実行する。

#### C メインプログラム

```
#include <windows.h>
#include <process.h>
#include <stdio.h>
#include "CSFINISH_MT.H"

extern void CBLPROG(void*);
unsigned __stdcall subthread(void* params);
int my_errorfunc(DWORD dwcode);
HANDLE mutex;
void mt_lock();
void mt_unlock();

int main()
{
 HANDLE thread[2];
 DWORD endcode[2];
 int dat1 = 1;
 int dat2 = 2;
 int i;

 /* EXTERNAL ファイルを使用している場合
 /* OPEN 用サブプログラム(ユーザが COBOL で作成)

 mutex = CreateMutex(NULL, FALSE, NULL);

 thread[0] = (HANDLE)_beginthreadex(NULL, 0, &subthread, &dat1, 0, NULL); /* スレッド作成 */
 thread[1] = (HANDLE)_beginthreadex(NULL, 0, &subthread, &dat2, 0, NULL); /* スレッド作成 */

 WaitForMultipleObjects(2, thread, TRUE, INFINITE);
 for (i = 0; i < 2; i++) {
 GetExitCodeThread(thread[i], &(endcode[i]));
 if (endcode[i] != 0) {
 /* スレッド i の異常終了処理。ここでプロセスを終了させる場合、
 /* ハンドルのクローズや EXTERNAL ファイルのクローズを行う

 }
 }
 CloseHandle(thread[0]);
 CloseHandle(thread[1]);
 CloseHandle(mutex);

 /* EXTERNAL ファイルを使用している場合
 /* CLOSE 用サブプログラム(ユーザが COBOL で作成)
 return(0);
}
unsigned __stdcall subthread(void* params)
{
 int compcode;

 if (CSCHKCOBRTPINIT() == 0) {
 return(99999);
 }
}
```

```

 }
 __try{
 CBLPROG(params);
 }
 __except(my_errorfunc(GetExceptionCode())) {
 return(99998);
 }
 compcode = CSGETTHREADCODE();
 CSFINISH_MT();
 return(0);
}
int my_errorfunc(DWORD dwcode)
{
 COBOL_RUNERR coberrinfo;
 CSGETCOBERR(&coberrinfo);
 CSFINISH_MT();
 switch(dwcode){
 case 0:
 return(EXCEPTION_EXECUTE_HANDLER);
 default :
 return(EXCEPTION_CONTINUE_SEARCH);
 }
}
void mt_lock()
{
 WaitForSingleObject(mutex, INFINITE);
}
void mt_unlock()
{
 ReleaseMutex(mutex);
}

```

/\* COBOL プログラム実行 \*/

/\* 構造化例外処理 \*/

/\* 必要に応じて COMPLETION-CODE 取得 \*/

/\* COBOL サブスレッド用終了処理 \*/

/\* フィルタ関数 \*/

/\* エラー情報構造体 \*/

/\* 必要に応じてエラー情報を取得 \*/

/\* COBOL サブスレッド終了処理 \*/

/\* COBOL ランタイムエラー \*/

/\* 自例外ハンドラで処理する \*/

/\* 以外のエラー \*/

/\* 別の例外ハンドラに処理を任せる \*/

/\* ロック関数 \*/

/\* アンロック関数 \*/

### COBOL プログラム

```

000010 IDENTIFICATION DIVISION .
000020 PROGRAM-ID. CBLPROG.
000030 DATA DIVISION.
000040 WORKING-STORAGE SECTION.
000050 01 WSD EXTERNAL PIC 99 USAGE BINARY.
000060 01 P USAGE COMP-2 VALUE 1.
000070 LINKAGE SECTION.
000080 01 PARAM1 PIC 9 USAGE BINARY.
000090 PROCEDURE DIVISION USING BY REFERENCE PARAM1.
000100 L1.
000110 CALL "mt_lock".
000120 DISPLAY "[" PARAM1 "]"start.".
000130 CALL "mt_unlock".
000140
000150 PERFORM 10 TIMES
000160 CALL "mt_lock"
000170 COMPUTE WSD = WSD + 1
000180 DISPLAY "[" PARAM1 "]" WSD
000190 CALL "mt_unlock"
000200 END-PERFORM.
000210
000220 CALL "mt_lock".
000230 DISPLAY "[" PARAM1 "]"end.".
000240 CALL "mt_unlock".
000250
000260 END PROGRAM CBLPROG.

```

- (1) スレッドの作成関数の引数に COBOL プログラムを指定してはならない。
- (2) マルチスレッドで使用する他言語では、CSFINISH.H ではなく CSFINISH\_MT.H をインクルードす

ること。

- (3) サブスレッドの最初の処理として、スレッド初期処理(CSCHKCOBRTPINIT 関数)を提供しているので、必ず呼び出す必要がある。

[CSCHKCOBRTPINIT利用規約]

- 引数はない。初期処理に成功した場合1を、失敗した場合0を返却する。
  - メインスレッドから呼び出してはならない。
  - 本関数が異常値を返却した場合、速やかにスレッドを終了すること。この規則に従わずCOBOLプログラムを呼び出した場合、メッセージを出力後プログラムは異常終了する。なお、本関数が異常値を返却した場合、それ以降に作成されるスレッドの初期処理はすべて失敗するため、致命的エラーとしてプロセスを終了すること。
  - 'CSFINISH\_MT.H'を提供しているので、利用されたい。
- (4) サブスレッドにて COBOL プログラムを実行後、COBOL プログラムが設定した COMPLETION-CODE の値を確認する場合、COMPLETION-CODE 取得処理(CSGETTHREADCODE 関数)を提供しているので、必要に応じて呼び出すこと。

[CSGETTHREADCODE利用規約]

- 引数はない。COMPLETION-CODEの値を返却する。
  - メインスレッドから呼び出してはならない。
  - CSFINISH\_MT 関数実行後に本関数を呼び出してはならない。
  - 'CSFINISH\_MT.H'を提供しているので、利用されたい。
- (5) COBOL がスレッド毎に使用しているリソースの解放処理を行うため、スレッド終了処理(CSFINISH\_MT 関数)を提供しているので、必ず呼び出す必要がある。

[CSFINISH\_MT利用規約]

- 引数、返却値ともない。
  - メインスレッドから呼び出してはならない。
  - CSFINISH\_MT 関数を呼び出してCOBOL側のスレッド終了処理を実行したあとでCOBOL プログラムを呼び出してはならない。
  - 'CSFINISH\_MT.H'を提供しているので、利用されたい。
- (6) サブスレッドの COBOL プログラムで実行時エラーが発生した場合、COBOL は構造化例外をスローする。COBOL プログラム呼び出し箇所には、構造化例外処理を実装し、例外を受け取るために\_\_finallyではなく、\_\_exceptを使用する。その際\_\_exceptに設ける例外フィルタ関数の返却値は、ユーザのアプリケーション構成により選択する。また、例外フィルタ関数で COBOL 提供関数を実行する場合、マルチスレッド版(CSFINISH\_MT 関数や CSGETTHREADCODE 関数)を使用すること。

実行時エラーの内容に応じて処理を切り分ける場合は、エラー情報取得処理(CSGETCOBERR 関数)を提供しているので、必要に応じて呼び出すこと。

また、情報の取得にはエラー情報構造体(COBOL\_RUNERR)を宣言する必要がある。エラー情報構造体は次の3つの要素を持つ。

- ・実行時エラー番号
- ・実行時エラー発生行番号
- ・実行時エラー発生プログラム名

詳細は、CSFINISH\_MT.Hを参照。

[CSGETCOBERR 利用規約]

- エラー情報構造体へのポインタを引数に指定する。  
失敗した場合参照先に NULL を設定し返却する。
- メインスレッドから呼び出してはならない。
- CSFINISH\_MT 関数実行後に本関数を呼び出してはならない。
- 'CSFINISH\_MT.H'を提供しているので、利用されたい。

- (7) マルチスレッド動作可能なアプリケーションにおいて、COBOL プログラムを動的ロードする(非推奨)プログラムを作成する場合、スレッド作成より前に COBOL プログラムをロードすること。スレッド作成までに COBOL ランタイムがロードされていない場合は、作成したスレッドで呼び出す COBOL プログラムが正常に動作しない可能性がある。
- (8) スレッドを作成したプログラムは、基本的にすべてのスレッド終了を待ち合わせる。ただし、他言語がメインプログラムの場合で、メインプログラム内でプロセス終了を行っておらず、各スレッドに対して相互に関連性がない場合は、このかぎりではない。
- (9) サブスレッドで CSFINISH 関数を呼び出してはいけない。サブスレッドでは CSFINISH\_MT 関数を呼び出すこと。メインスレッド上で COBOL プログラムを実行しない場合、メインスレッドで CSFINISH 関数を実行する必要はない。
- (10) CSSETABORT 関数はサブスレッドには影響しない。CSSETABORT 関数で登録された関数はメインスレッド上で動作する COBOL プログラム異常終了時に実行する。メインスレッド上で COBOL プログラムを実行しない場合は CSSETABORT 関数を登録する必要はない。
- (11) SetConsoleCtrlHandler()で登録した関数の実行は、スレッドを意識せずに行われ、実行スレッドは強制終了のタイミングにより異なる。よって強制終了ハンドラを登録する場合は注意すること。

## 第21章 関連製品連携

### 21.1 概説

本製品と組み合わせて使用可能なオプション製品や関連製品は次のとおり。

#### [Enterprise for Windows]

| 機能                | 関連製品                                 |
|-------------------|--------------------------------------|
| RAF 順ファイル         | refam VX                             |
| 索引ファイル            | ISAM                                 |
| ソートマージ            | SORTKIT/Enterprise                   |
| COBOL/S           | COBOL/S for Windows                  |
| GMP マクロ           | GMP マクロ for Windows                  |
| 帳票作成・印刷           | SpoolServer                          |
|                   | さばける君                                |
| データベース接続          | Oracle Programmer<br>(Pro*COBOL)     |
|                   | COBOL SQL アクセス                       |
| オンライントランザクションシステム | TPBASE                               |
| ジョブ管理             | WebSAM JobCenter                     |
| 日本語文字拡張           | FontAvenue UniAssist コード変換           |
|                   | FontAvenue UniAssist クライアント PC 用フォント |
| COBOL プログラムの標準部品  | COBOL 拡張部品                           |

#### [Standard]

| 機能       | 関連製品                                                                             |
|----------|----------------------------------------------------------------------------------|
| 順ファイル    | IFAS FILE(Standard Edition 版のとき)                                                 |
| 相対ファイル   | IFAS FILE(Standard Edition 版のとき)                                                 |
| 索引ファイル   | IFAS FILE(Standard Edition 版のとき)<br>COBOL DB アクセス(Standard Edition 版のとき)<br>ISAM |
| ソートマージ   | SORTKIT                                                                          |
| 帳票作成・印刷  | VISUALFORMS                                                                      |
|          | WebSAM Rakuform                                                                  |
| データベース接続 | COBOL DB アクセス for Oracle                                                         |
|          | COBOL DB アクセス for SQL Server                                                     |
|          | IFASPRO RDB <sup>(注 1)</sup>                                                     |
|          | Oracle Programmer<br>(Pro*COBOL)                                                 |

|                   |                                                |
|-------------------|------------------------------------------------|
|                   | COBOL SQL アクセス                                 |
| オンライントランザクションシステム | TPBASE <sup>(注 2)</sup>                        |
| ジョブ管理             | WebSAM JobCenter                               |
| ジョブ管理機能, プリンタ管理機能 | COBOL アプリケーション実行環境                             |
| システムサブルーチン        | COBOL 拡張システムサブルーチン                             |
| 日本語文字拡張           | 日本語文字拡張セット                                     |
|                   | G1 集合開発支援パッケージ                                 |
|                   | G1 集合実行支援パッケージ                                 |
|                   | FontAvenue UniAssist コード変換                     |
|                   | FontAvenue UniAssist クライアント PC 用フォント           |
| GUI 画面            | COBOL GUI <sup>(注 1)</sup>                     |
| 内部ファイル ID 置換機能    | COBOL A-VX 互換機能／カンパニ ID オプション <sup>(注 1)</sup> |

**注意**

(注 1) Standard Edition 版では使用できるが, SE 動作モードでは使用できない。

(注 2) SE 動作モードでは使用できるが, Standard Edition 版では使用できない。

**[Enterprise for Linux]**

| 機能                | 関連製品                          |
|-------------------|-------------------------------|
| RAF 順ファイル         | refam/E<br>refam VX           |
| 索引ファイル            | ISAM                          |
| ソートマージ            | SORTKIT/Enterprise            |
| COBOL/S           | COBOL/S for Linux             |
| GMP マクロ           | GMP マクロ for Linux             |
| データベース接続          | Oracle Programmer (Pro*COBOL) |
|                   | COBOL SQL アクセス                |
| オンライントランザクションシステム | TPBASE                        |
| COBOL プログラムの標準部品  | COBOL 拡張部品                    |

## 21.2 アプリケーション生成時の注意

オプション製品や関連製品との組み合わせでアプリケーション生成する際の注意点を示す。

[Enterprise for Windows]

| 関連製品                          | 生成するアプリケーション                                                                                                                                                   |                                             |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
|                               | 非マルチスレッド                                                                                                                                                       | マルチスレッド                                     |
| refam VX                      | 特になし                                                                                                                                                           | マルチスレッドでの動作は不可                              |
| ISAM                          | 特になし                                                                                                                                                           | マルチスレッドでの動作は不可                              |
| COBOL/S for Windows           | COBOL85 規格の範囲の機能のみ使用可能。                                                                                                                                        |                                             |
| GMP マクロ for Windows           |                                                                                                                                                                |                                             |
| SpoolServer                   | COBOL アプリケーションで作成したスプールファイルを、COBOL アプリケーションとは非同期に各製品を経由して印刷処理を行う。よって、特に注意事項はない。                                                                                |                                             |
| さばける君                         |                                                                                                                                                                |                                             |
| Oracle Programmer (Pro*COBOL) | ・ Pro*COBOL のオプションに comp1=integer の指定不可。ホスト変数に COMP-1 および COMP-2 を使用する場合は、「ホスト変数の同値化」を利用したコーディングを行う。<br>・ホスト変数に重ね符号付き外部 10 進数項目を使用される場合、コンパイルオプションに -CS を指定する。 | 生成不可                                        |
| TPBASE                        | 特になし                                                                                                                                                           | マルチスレッドでの動作は不可(マルチスレッドで生成したアプリケーションの単独動作は可) |
| WebSAM JobCenter              | 特になし                                                                                                                                                           | マルチスレッドでの動作は不可(マルチスレッドで生成したアプリケーションの単独動作は可) |
| COBOL SQL アクセス                | コンパイルオプションに -CF/-Cj/-CU を指定してはならない                                                                                                                             | マルチスレッドでの動作は不可                              |
| COBOL 拡張部品                    | COBOL85 規格の範囲の機能のみ使用可能。                                                                                                                                        |                                             |



## [Standard]

| 関連製品                                  | 生成するアプリケーション                                                                                                                                                                                                                                   |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IFAS FILE                             | 特になし                                                                                                                                                                                                                                           |
| ISAM                                  | 特になし                                                                                                                                                                                                                                           |
| SORTKIT<br>( COBOL ソート・マージ<br>Client) | クライアント OS でソート・マージのある COBOL アプリケーションを実行する場合にのみ使用可能。バッチなどから SORT コマンド実行, またはサーバ OS でソート・マージのある COBOL アプリケーションを実行する場合は, SORTKIT/Enterprise をご利用ください。                                                                                             |
| SORTKIT<br>(SORTKIT/Enterprise)       | 特になし                                                                                                                                                                                                                                           |
| VISUALFORMS                           | 特になし                                                                                                                                                                                                                                           |
| WebSAM Rakuform                       | 特になし                                                                                                                                                                                                                                           |
| COBOL DB アクセス for<br>Oracle           | 特になし                                                                                                                                                                                                                                           |
| COBOL DB アクセス for SQL<br>Server       | 特になし                                                                                                                                                                                                                                           |
| IFASPRO RDB                           | 特になし                                                                                                                                                                                                                                           |
| Oracle Programmer<br>(Pro*COBOL)      | <ul style="list-style-type: none"> <li>・Pro*COBOL のオプションに comp1=integer の指定不可。ホスト変数に COMP-1 および COMP-2 を使用する場合は, 「ホスト変数の同値化」を利用したコーディングを行う。</li> <li>・ホスト変数に重ね符号付き外部 10 進数項目を使用される場合, コンパイルオプションに -CS を指定する。</li> </ul>                        |
| COBOL SQL アクセス                        | コンパイルオプションに -CF/-Cj を指定してはならない                                                                                                                                                                                                                 |
| WebSAM JobCenter                      | 特になし                                                                                                                                                                                                                                           |
| COBOL アプリケーション実行<br>環境                | <ul style="list-style-type: none"> <li>・本製品の利用により, 外部スイッチ (SSW)/COMPLETION-CODE/STOP RUN 時の別プログラム呼び出し, ならびにスプール機能が使用可能となる。</li> <li>ただし, #SFGEN のサポートはない。</li> <li>・COBOL WS エミュレータで画面制御機能を使用する場合, COBOL プログラムの翻訳時に -Mw オプションを指定する。</li> </ul> |
| COBOL 拡張システムサブルーチン                    | 特になし                                                                                                                                                                                                                                           |
| 日本語文字拡張セット                            | 特になし                                                                                                                                                                                                                                           |
| G1 集合開発支援パッケージ                        | 特になし                                                                                                                                                                                                                                           |
| G1 集合実行支援パッケージ                        | 特になし                                                                                                                                                                                                                                           |
| COBOL GUI                             | <ul style="list-style-type: none"> <li>・COBOL GUI アプリケーションを作成する場合, COBOL プログラムの翻訳時に -Pg オプションを指定する。</li> <li>・COBOL GUI の画面節移行機能を使用する場合, COBOL プログラムの翻訳時に -P2 オプションを指定</li> </ul>                                                              |

|                               |      |
|-------------------------------|------|
|                               | する。  |
| COBOL A-VX 互換機能／カンパニ ID オプション | 特になし |

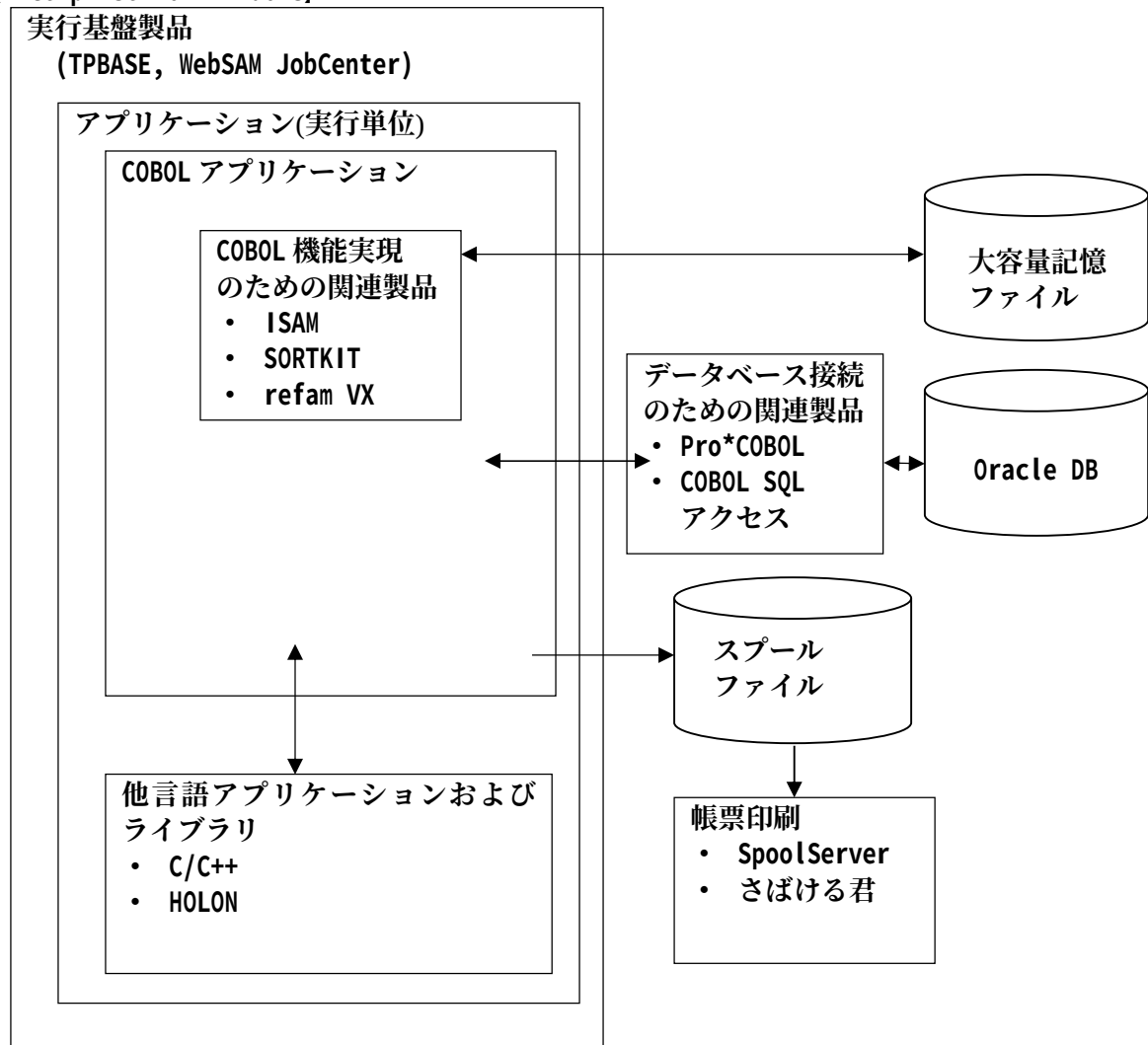
## [Enterprise for Linux]

| 関連製品                                   | 生成するアプリケーション                                                                                                                                                     |                                               |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
|                                        | 非マルチスレッド                                                                                                                                                         | マルチスレッド                                       |
| refam/E<br>refam VX                    | 特になし                                                                                                                                                             | マルチスレッドでの動作は不可                                |
| ISAM                                   | 特になし                                                                                                                                                             | マルチスレッドでの動作は不可                                |
| COBOL/S for Linux<br>GMP マクロ for Linux | COBOL85 規格の範囲の機能のみ使用可能。                                                                                                                                          |                                               |
| SORTKIT/Enterprise                     | 特になし                                                                                                                                                             | マルチスレッドで利用の場合、COBOL プログラムの翻訳時に -MT オプションを指定する |
| Oracle Programmer<br>(Pro*COBOL)       | ・Pro*COBOL のオプションに comp1=integer の指定不可。<br>ホスト変数にCOMP-1 および COMP-2 を使用する場合は、「ホスト変数の同値化」を利用したコーディングを行う。<br>・ホスト変数に重ね符号付き外部 10 進数項目を使用される場合、コンパイルオプションに -CS を指定する。 | 生成不可                                          |
| COBOL SQL アクセス                         | ・コンパイルオプションに -Cj/-CU を指定してはならない。                                                                                                                                 | マルチスレッドでの動作は不可                                |
| TPBASE                                 | 特になし                                                                                                                                                             | マルチスレッドでの動作は不可(マルチスレッドで生成したアプリケーションの単独動作は可)   |
| COBOL 拡張部品                             | COBOL85 規格の範囲の機能のみ使用可能。                                                                                                                                          |                                               |

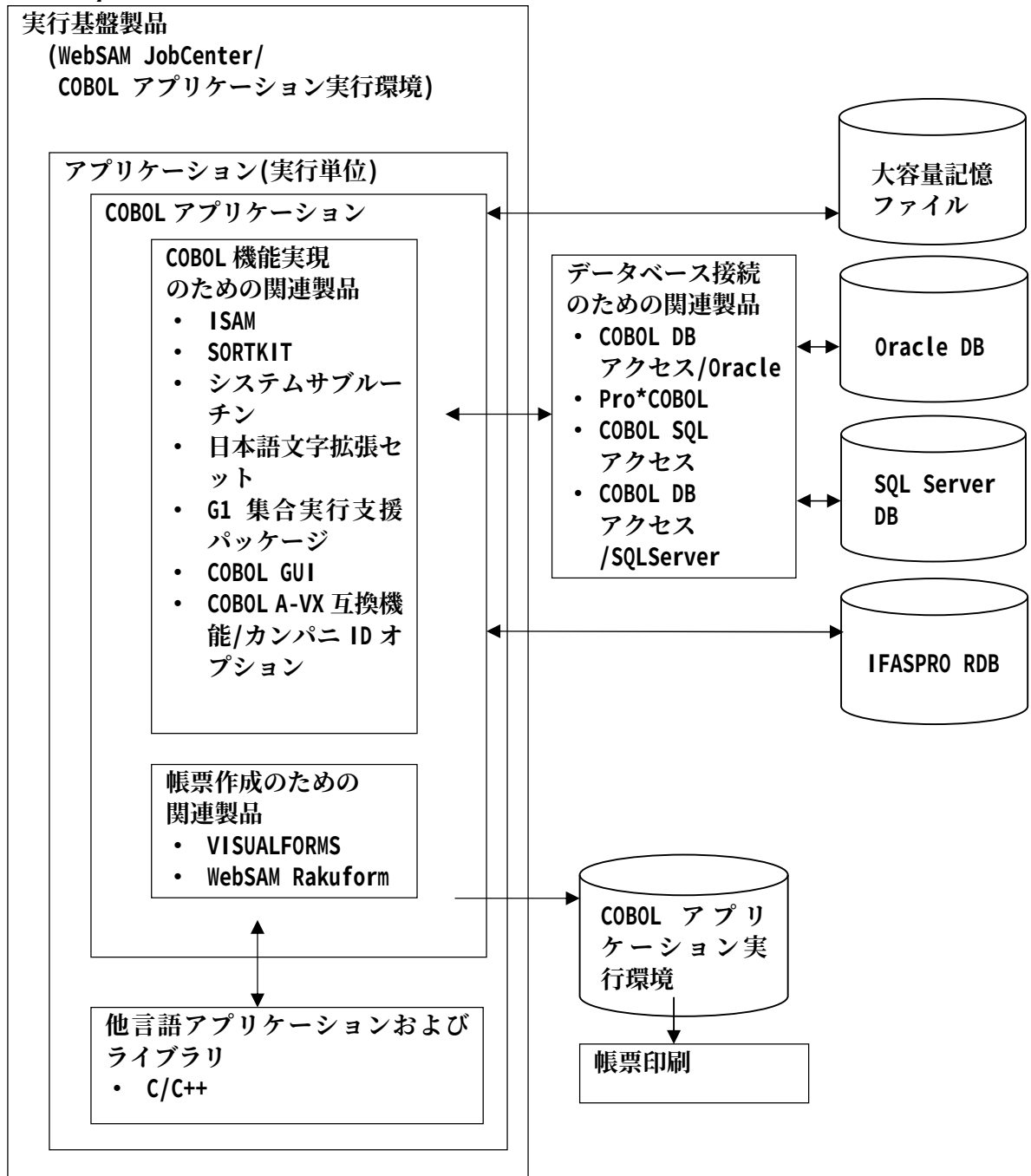
## 21.3 動作環境(実行環境)

オプション製品や関連製品との組み合わせでの動作構成を示す。

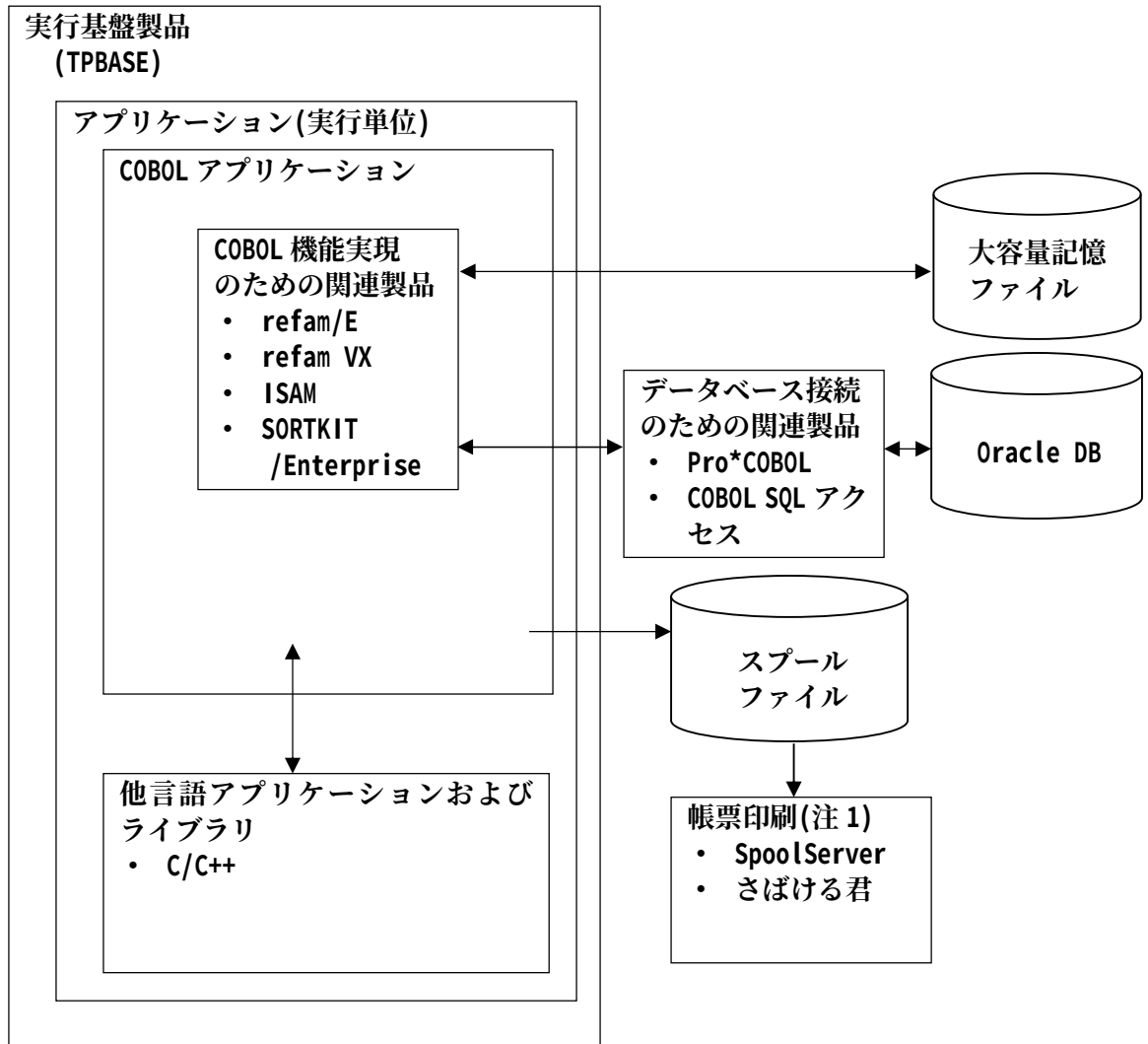
[Enterprise for Windows]



[Standard]



[Enterprise for Linux]



**注意** (注 1)帳票の印刷は, Windows 上の SpoolServer やさばける君で行う。

オプション製品や関連製品との組み合わせて実行可能な動作環境を示す。

[Enterprise for Windows]

| 関連製品                | 説明                                | 64 ビット OS                             |     |
|---------------------|-----------------------------------|---------------------------------------|-----|
|                     |                                   | シングル                                  | マルチ |
| refam VX            | RAF 順ファイル機能の提供                    | ○                                     | ×   |
| ISAM                | 索引ファイル機能の提供                       | ○                                     | ×   |
| SORTKIT/Enterprise  | 整列併合機能の提供                         | ○                                     | ○   |
| TPBASE              | オープン環境下における集中型のオンラインランザクション処理システム | ○                                     | ×   |
| Oracle Pro*COBOL    | 埋め込み SQL 解釈するプリコンパイラ              | ○                                     | ×   |
| COBOL SQL アクセス      | 埋め込み SQL 解釈するプリコンパイラ              | ○                                     | ×   |
| COBOL/S for Windows | 既存 COBOL/S 資産の移行時に使用するプリコンパイラ     | COBOL ソースを生成するプリコンパイラのため、実行動作には関係しない。 |     |
| GMP マクロ for Windows | ACOS-4 互換の GMP マクロを使用可能とする製品      | COBOL ソースを生成するプリコンパイラのため、実行動作には関係しない。 |     |

## [Standard]

| 関連製品                          | 説明                            | 64 ビット OS    |
|-------------------------------|-------------------------------|--------------|
| ISAM                          | 索引ファイル機能の提供                   | ○            |
| SORTKIT                       | 整列併合機能の提供                     | ○            |
| COBOL DB アクセス for Oracle      | データベース接続機能の提供                 | ○            |
| COBOL DB アクセス for SQL Server  | データベース接続機能の提供                 | ○            |
| IFASPRO RDB                   | データベース接続機能の提供                 | ○            |
| Oracle Pro*COBOL              | 埋め込み SQL 解釈するプリコンパイラ          | ○            |
| COBOL SQL アクセス                | 埋め込み SQL 解釈するプリコンパイラ          | ○            |
| COBOL アプリケーション実行環境            | ジョブ管理機能, プリンタ管理機能の提供          | ○            |
| COBOL 拡張システムサブルーチン            | システムサブルーチンの提供                 | ○            |
| 日本語文字拡張セット                    | Express 外字と Windows 外字の直接相互変換 | ○            |
| G1 集合開発支援パッケージ                | G1 集合文字機能の提供                  | 実行動作には関係しない。 |
| G1 集合実行支援パッケージ                | G1 集合文字機能の提供                  | ○            |
| COBOL GUI                     | COBOL GUI アプリケーションの提供         | ○            |
| COBOL A-VX 互換機能／カンパニ ID オプション | 内部ファイル ID 置換機能の提供             | ○            |

## [Enterprise for Linux]

| 関連製品                             | 説明                                             | 64 ビット OS                                 |     |
|----------------------------------|------------------------------------------------|-------------------------------------------|-----|
|                                  |                                                | シングル                                      | マルチ |
| refam/E<br>refam VX              | RAF順ファイル機能<br>の提供                              | ○                                         | ×   |
| ISAM                             | 索引ファイル機能<br>の提供                                | ○                                         | ×   |
| SORTKIT/Enterprise               | 整列併合機能の提供                                      | ○                                         | ○   |
| TPBASE                           | オープン環境下における<br>集中型のオンライントラ<br>ンザクション処理システ<br>ム | ○                                         | ×   |
| Oracle Programmer<br>(Pro*COBOL) | 埋め込みSQL 解釈するプ<br>リコンパイラ                        | ○                                         | ×   |
| COBOL SQL アクセス                   | 埋め込みSQL 解釈するプ<br>リコンパイラ                        | ○                                         | ×   |
| COBOL/S for Linux                | 既存 COBOL/S 資産の移行<br>時に使用するプリコンパ<br>イラ          | COBOL ソースを生成するプリコンパイラ<br>のため、実行動作には関係しない。 |     |
| GMP マクロ for Linux                | ACOS-4 互換のGMP マクロ<br>を使用可能とする製品                | COBOL ソースを生成するプリコンパイラ<br>のため、実行動作には関係しない。 |     |



## 21.4 アプリケーション実行時の注意事項

【Standard】本章では,Standard Edition 版および SE 動作モードで関連製品を使用する際の注意事項を記載する。

### 21.4.1 プログラム名段落の INITIAL 句,COMMON 句

関連製品を使用する場合,プログラム名段落の INITIAL 句,COMMON 句は使用不可である。

### 21.4.2 データ部の諸元

関連製品を使用する場合,集団項目の文字数,英字／英数字項目の文字数,日本語項目の文字数の桁数は以下のとおりである。

| 要素           |                     | 諸元              |            |
|--------------|---------------------|-----------------|------------|
|              |                     | 関連製品なし          | 関連製品あり     |
| 集団項目の文字数     | ファイル節のとき            | 1～65535 文字      | 1～65535 文字 |
|              | その他の節のとき            | 1～2147483647 文字 | 1～65535 文字 |
| 英字／英数字項目の文字数 | ファイル節のとき            | 1～65535 文字      | 1～32767 文字 |
|              | その他の節のとき            | 1～2147483647 文字 | 1～32767 文字 |
|              | 画面節のとき              | 255 文字          | 255 文字     |
| 日本語項目の文字数    | ファイル節のとき            | 1～32767 文字      | 1～16383 文字 |
|              | その他の節／EXTERNAL 句のとき | 1～1073741823 文字 | 1～16383 文字 |
|              | 画面節のとき              | 127 文字          | 127 文字     |

関連製品を使用する場合,LINAGE 句に指定できる 1 論理ページの行数,プリンタファイルにおける 1 レコードの最大長の実行時の諸元は,関連製品に依存する。

### 21.4.3 手続き部の諸元

関連製品を使用する場合,PROCEDURE USING のデータ名の個数,CALL 文のパラメータの個数,1 回の WRITE 文での改行数は以下のとおりである。

| 要素                       | 諸元      |        |
|--------------------------|---------|--------|
|                          | 関連製品なし  | 関連製品あり |
| PROCEDURE USING のデータ名の個数 | 1024 個  | 250 個  |
| CALL 文のパラメータの個数          | 1024 個  | 253 個  |
| 1 回の WRITE 文での改行数        | 0～32767 | 0～255  |

## 第22章 画面制御機能

**【Standard】**本機能は Standard Edition 版および Enterprise Edition for Windows 版の SE 動作モードにおいて、使用できる。

## 22.1 概説

本システムにおいて、ディスプレイステーションの画面を使用して、データエントリまたは問い合わせなどの業務を行う場合、プログラムの作成者が操作員の使いやすい画面を作成し、効率のよいプログラミングができるように、本 COBOL では拡張機能として画面制御機能を設けている。

この画面制御機能は、データ部の画面節(画面記述項、画面データ記述項)と手続き部の画面入出力文(ACCEPT、DISPLAY 文など)で構成される。

本章では、画面制御機能に属する句や文のうち重要なものや解りにくいものについて、言語説明に対する補足説明を行う。したがって、それぞれの句や文に対する詳細な規則については、『COBOL 言語説明書』を参照。

### 22.1.1 画面の構成

COBOL で画面にデータを表示したり、キーボードからデータを入力する場合、以下の 3 つの方法がある。

## (1) 画面制御機能を使用する方法

### 図 22-1 画面の構成(画面制御機能)

[illegible]

データ部の画面節で、画面の構成を定義しておき、画面操作の ACCEPT, DISPLAY 文を使用して任意の位置にデータを表示したり、入力することができる。  
画面を使用してデータエントリや問い合わせなどの業務を行う場合、操作員の使用しやすい画面を作成するのに適している。

(2) ガイド領域に表示する方法

GDD に関係付けられた呼び名を UPON 句に指定した DISPLAY 文や定数指定の STOP 文で、ガイド領域にデータを表示することができる。  
その表示箇所は、画面制御機能の有無に関係なく、以下のように表示される。

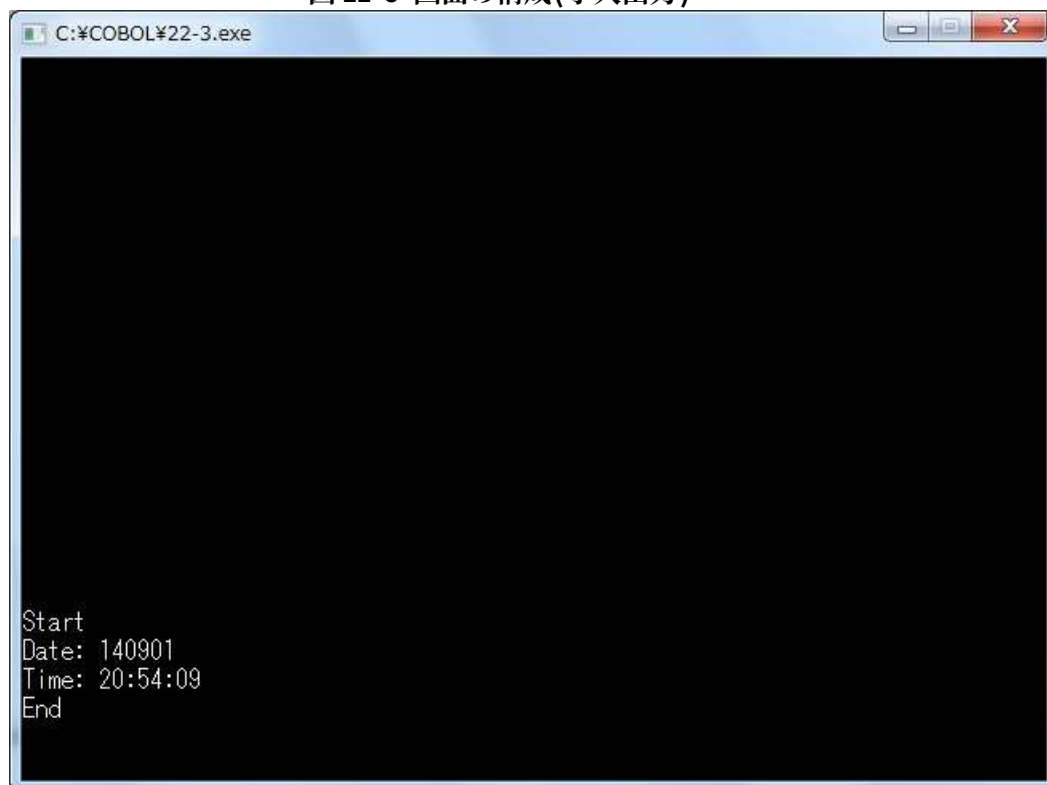
図 22-2 画面の構成(ガイド領域)



(3) 小入出力文を使用する方法

詳細は、「5.4 小入出力機能」を参照。

図 22-3 画面の構成(小入出力)



## 22.1.2 カスタマイズ機能

画面制御機能では、操作員がキーボードから各種データを打鍵し、その結果がプログラムに通知される。その中には入力終了キーなどとして、COBOL コンパイラが事前に規定しているものがある。しかし、その規定はすべての操作員が望むキー操作とは一致するとは限らない。また、事前に規定されているキー操作のうち一部のキーは使用できないキーとしたい(使用を規制したい)場合も考えられる。

そこで、本 COBOL ではキー操作の変更を可能にするカスタマイザを用意している。カスタマイザの操作法は、『COBOL ユーザーズガイド』を参照。

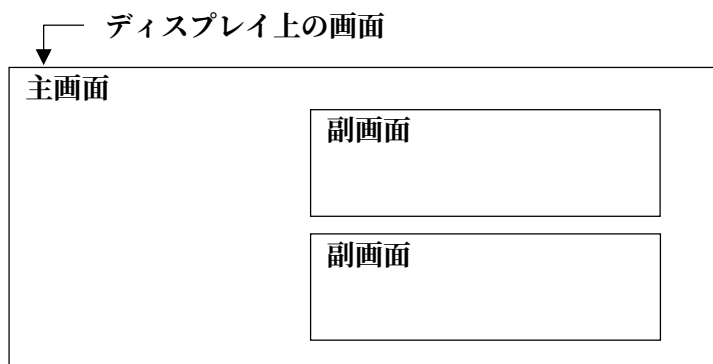
また、『COBOL 言語説明書』と以降で説明するキー操作に関する記述は、COBOL コンパイラにより事前に規定したものをもとにしている。カスタマイザにより変更した場合は、変更事項に置き換えて読むこと。

## 22.1.3 副画面機能

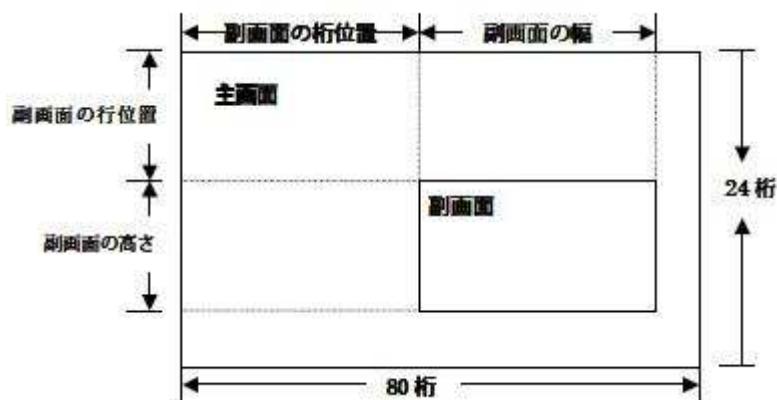
ディスプレイステーション上の画面に対応する画面を主画面と呼ぶ。

副画面は主画面上で任意の位置、任意の大きさで領域を定義できる。また、副画面の範囲内でデータを入出力することができる。この機能を副画面機能と呼ぶ。

副画面はなくてもよく、また、複数の副画面を定義できる。

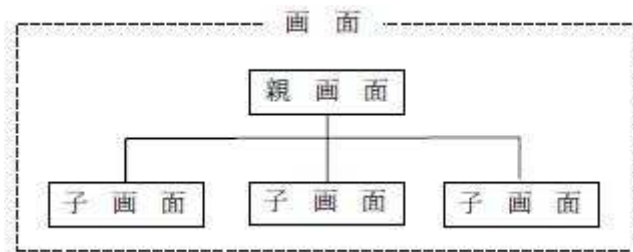


主画面に対して、副画面の大きさと表示する位置は以下のように定める。



## 22.1.3.1 親画面と子画面

親画面と子画面の関係は、画面の相対的な関係である。  
主画面を親画面とし、副画面を子画面とする。



本システムでは、副画面を親画面とした子画面は定義不可である。(副画面のネストは不可)

## 22.1.3.2 主画面

- (1) 画面の表示位置はカスタマイズ機能により変更することができる。  
画面の左上が基点 (1 行 1 桁) である。
- (2) 画面の大きさはカスタマイズ機能により変更することができる。既定値は 24 行 80 桁である。  
『COBOL 言語説明書』と以降の説明にある画面の大きさの記述は既定値をもとにしている。カスタマイズにより変更した場合は、変更事項に置き換えて読むこと。
- (3) 画面の入出力
  - 画面に対して、BOX, 色等を指定したデータの入出力ができる。
  - 画面をはみ出した位置からの入出力はできない。
  - 表示データが画面の右端を越えると次の行に表示される。  
ただし、最下行では右端を越えると桁落ちする。
  - 画面の左端へ BOX 指定のデータを表示すると BOX の左縦線は表示されない。
  - 画面の右端へ BOX 指定のデータを表示すると BOX の右縦線は表示されない。

## 22.1.3.3 副画面

- (1) 画面の表示位置を変更できる。
- (2) 画面の大きさを変更できる。  
画面の高さは 1 から 24 行、幅は 1 から 80 桁の範囲で変更できる。
- (3) 画面は OPEN 文により表示域が使用可能になり、CLOSE 文により表示域は OPEN 文の前の状態に戻される。
- (4) 枠が指定できる。
  - 枠を指定すると、指定した画面の外側に縦線を表示するため、左右 1 桁ずつ大きな領域が表示域として使用される。  
枠の指定があるとき、画面開始位置 (表示域の左端) が主画面の 1 桁目であってはならない。  
枠の指定があるとき、画面終了位置 (表示域の右端) が主画面の 80 桁目であってはならない。
- (5) 画面の入出力
  - 画面に対して、BOX, 色等を指定したデータの入出力ができる。
  - 画面をはみ出した位置からの入出力はできない。

- 表示データが画面の右端を越えてはならない。
  - 画面の左端へ BOX 指定のデータの表示は枠指定の場合は可能である。  
枠指定のない場合は、エラーとなる。  
画面の右端へ BOX 指定のデータの表示は枠指定の場合は可能である。  
枠指定のない場合は、エラーとなる。
  - ACCEPT, DISPLAY 文で BEFORE 句と AFTER 句は使用できない。
- (6) 画面の消去に CLEAR 句は使用できない。
- (7) 同時に OPEN できる副画面の数は、16 個までである。

#### 22.1.3.4 親画面と子画面の関係

- (1) 画面の領域に関して
- 子画面の位置は親画面の領域内を、1 相対の行と桁で表す。
  - 子画面は、親画面の領域内でなければならない。
  - 親画面の左端、または、右端に重なる子画面は、枠指定を指定してはならない。
- (2) 画面の入出力
- 親画面でデータを入出力する時、すでに表示している子画面の表示領域に重なってはならない。

## 22.2 画面記述項

画面記述項は、画面管理システムと利用者プログラム間の情報交換領域を定義するもので次の項目より構成されている。

### 書き方 1

```
SD 画面記述名 1
END STATUS IS データ名 1
[LINE POSITION IS データ名 2]
[COLUMN POSITION IS データ名 3]
[BTAB POSITION IS データ名 4].
```

## 書き方 2

**SD** 画面記述名 1**END STATUS** IS データ名 1**[LINE POSITION IS データ名 2]****[COLUMN POSITION IS データ名 3]****[BTAB POSITION IS データ名 4]**

**PARENT WINDOW IS**      { 画面記述名 2  
                                 OMITTED }

**[WINDOW STATUS IS データ名 5]**

**WINDOW HIGHT IS**      { 整数 1  
                                 データ名 6 } { LINE  
                                                 LINES }

**WINDOW WIDTH IS**      { 整数 2  
                                 データ名 7 } { COLUMN  
                                                 COLUMNS }

**DISPLAY LINE NUMBER IS**      { 整数 3  
                                                 データ名 8 }

**DISPLAY COLUMN NUMBER IS**      { 整数 4  
                                                 データ名 9 }

**[BORDER].**

データ名 1 は 2 桁の英数字項目として、データ名 2 とデータ名 3 は 4 桁の数字項目として、データ名 4 は 8 桁の数字項目として自動的に確保されるので定義する必要はない。

(1) 入力終了状態(EN D STATUS)

画面 ACCEPT 文の入力終了状態を通知する。入力操作で入力終了として打鍵した入力終了キーを示す値が通知される。

入力終了状態の設定値を記す。アプリケーションカスタマイザにて変更可能である。



表 22-1 END STATUS データ項目に設定される値

|                | 値  | 対応するキー              | 備考       |
|----------------|----|---------------------|----------|
| 通常の入力終了キー      | 00 | 終了キーなし              | 桁数分の入力終了 |
|                | 01 | [Enter]             |          |
|                | 02 | [Shift]+[F8], [F11] |          |
|                | 03 | [Shift]+[F9], [F12] |          |
|                | 04 | [F10]               |          |
|                | 05 | [Shift]+[Enter]     |          |
|                | 06 | [Tab]               |          |
|                | 07 | ↑                   |          |
|                | 08 | ↓                   |          |
|                | 09 | [Back space]        |          |
| プログラムファンクションキー | P0 | [End]               |          |
|                | P1 | [F1]                |          |
|                | P2 | [F2]                |          |
|                | P3 | [F3]                |          |
|                | P4 | [F4]                |          |
|                | P5 | [F5]                |          |
|                | P6 | [F6]                |          |
|                | P7 | [F7]                |          |
|                | P8 | [F8]                |          |
|                | P9 | [F9]                |          |
|                | PA | [Shift]+[F10]       |          |
|                | PB | [Shift]+[F1]        |          |
|                | PC | [Shift]+[F2]        |          |
|                | PD | [Shift]+[F3]        |          |
|                | PE | [Shift]+[F4]        |          |
|                | PF | [Shift]+[F5]        |          |
|                | PG | [Shift]+[F6]        |          |
| 機能キー           | FW | [Shift]+[Page Down] |          |
|                | BW | [Shift]+[Page Up]   |          |
|                | FM | [Ctrl]+[F10]        |          |
|                | UP | [Page Down]         |          |
|                | DW | [Page Up]           |          |

## (2) 行位置(LINE POSITION)

画面 ACCEPT/DISPLAY 文により画面に対し直前に行われた入出力領域(フィールド)の先頭の文字位置の行の値が通知される。

## (3) 桁位置(COLUMN POSITION)

画面 ACCEPT/DISPLAY 文により画面に対し直前に行われた入出力領域(フィールド)の先頭の文字位置の桁の値が通知される。

## (4) BTAB 位置(BTAB POSITION)

- 画面 ACCEPT 文を **Back space** キーを用いて終了したとき、最初に **Back space** キー

- ーを押した入力領域(フィールド)の先頭の行位置と桁位置が通知される。ACCEPT 文が **Back space** キーで終了しなかったときは、値は通知されない。
- a. BTAB POSITION の値がゼロのとき、  
その ACCEPT 文で扱われる INITIAL FILLER CHARACTER 句の指定は、ソース上の指定どおりに扱われる。
  - b. BTAB POSITION の値がゼロでないとき、  
その ACCEPT 文で扱われる INITIAL FILLER CHARACTER 句の指定は、NO IFC 句が指定されたかのように扱われる。
- 画面 ACCEPT 文で、先頭の行位置と桁位置が BTAB POSITION の値と同じ入力領域への入力処理が行われた場合に、BTAB POSITION の値は初期値に戻る。
  - **Tab** キーによるスキップ操作について  
**Tab** キーによるスキップ操作は INITIAL FILLER CHARACTER 句の指定と密接に関連しているが、BTAB POSITION との関係もある。
    - a. INITIAL FILLER CHARACTER 句に定数指定のとき  
BTAB POSITION の値がゼロのとき、INITIAL FILLER CHARACTER 句で指定された文字が存在するため、スキップ操作は無効となる。  
BTAB POSITION の値がゼロでないとき、先頭の行位置と桁位置が BTAB POSITION の値と異なれば、スキップ操作は有効となるが、同じならば、スキップ操作は無効となる。
    - b. INITIAL FILLER CHARACTER 句に NO 指定のとき  
入力項目の占める領域に INITIAL FILLER CHARACTER 句で指定された文字が存在しないため、スキップ操作は常に有効となる。

副画面を使用するときは、副画面の定義を次の項目により行う。

#### (1) 画面状態(WINDOW STATUS)

- 副画面の状態を収めるデータ項目を指定する。
  - 2 文字の概念上の存在である。OPEN 文、CLOSE 文、ACCEPT 文、DISPLAY 文の実行中に、その入出力動作の状態を示す値を設定する。入出力状態の値は、その画面に対する画面記述項の WINDOW STATUS(画面状態)句を使用することにより COBOL プログラムで使用可能となる。
  - 入出力状態の値のいくつかは、重大誤り状態を示す。これらは、値 4 の桁で始まる状態である。入出力操作に対する入出力状態の値が、その誤り状態を示すならば、標準の入出力制御システム誤りの処理を行う。
  - 入出力状態は、入出力動作完了後の次の条件のうちの 1 つを示す。
    - a. 成功。入出力文(Input-output statement)の実行が成功した。
    - b. 論理誤り。入出力文の実行が、その画面上の妥当でない順序の入出力動作または利用者が定めた境界を超えたことによって不成功になった。
- WINDOW STATUS(画面状態)で、副画面に対する画面文の実行結果を知ることができる。

表 22-2 WINDOW STATUS データ項目に設定される値

| 値  | 説明   |                                 |
|----|------|---------------------------------|
| 00 | 成功   | 入出力文の実行が成功した。                   |
| 41 | 論理誤り | 開かれている状態の画面に OPEN 文を実行しようとした。   |
| 42 |      | 開かれていない状態の画面に CLOSE 文を実行しようとした。 |
| 43 |      | OPEN 文を実行しようとしたが、親画面が開かれていない。   |
| 44 |      | 副画面の表示域が親画面の表示域を越えた。            |

|    |  |                                      |
|----|--|--------------------------------------|
| 45 |  | 同時に開ける副画面数が最大値を越えた。                  |
| 46 |  | 開かれていない副画面に DISPLAY 文／ACCEPT 文を実行した。 |
| 47 |  | DISPLAY 文／ACCEPT 文の表示域が副画面の表示域を越えた。  |

## (2) 画面の高さ(WINDOW HEIGHT)

- 副画面の高さを指定する。
- 値は、1 から 24 の範囲内でなければならない。

## (3) 画面の幅(WINDOW WIDTH)

- 副画面の幅を指定する。
- 値は、1 から 80 の範囲内でなければならない。

## (4) 表示する行位置(DISPLAY LINE NUMBER)

- 副画面の左上を表示する行位置を親画面内の行指定で指定する。
- 値は、1 から 24 の範囲内でなければならない。

## (5) 表示する桁位置(DISPLAY COLUMN NUMBER)

- 副画面の左上を表示する桁位置を親画面内の桁指定で指定する。
- 値は、1 から 80 の範囲内でなければならない。

## (6) 枠(BORDER)

- 副画面を枠で囲む。

## 22.3 画面データ記述項

画面データ記述項は、画面の構成および属性を記述する一群のデータ記述項で構成する。各データ記述項はレベル番号とデータ名および必要に応じて一群の独立した句とで構成される。

画面データ記述項の書き方には次の5種類がある。

- レベル番号 01 の集団記述項を指定する。(書き方 1)
- レベル番号 02～48 の集団記述項を指定する。(書き方 2)
- レベル番号 02～49 の基本記述項を指定する。(書き方 3, 4, 5)

### 22.3.1 一般形式

書き方 1(レベル番号 01 の集団記述項)

01 データ名 1

$$\left[ \begin{array}{l} \underline{\text{LINE NUMBER IS}} \quad \left\{ \begin{array}{l} \text{データ名 2} \quad [\underline{\text{PLUS}} \text{ 整数 1}] \\ [\underline{\text{PLUS}} \text{ 整数 2}] \end{array} \right\} \\ \\ \left[ \begin{array}{l} \underline{\text{CLEAR}} \quad \left\{ \begin{array}{l} \underline{\text{SCREEN}} \\ \left\{ \begin{array}{l} \underline{\text{LINE}} \\ \underline{\text{DATA}} \end{array} \right\} \end{array} \right\} \quad \left[ \underline{\text{TO}} \quad \left\{ \begin{array}{l} \text{データ名 3} \quad [\underline{\text{PLUS}} \text{ 整数 3}] \\ [\underline{\text{PLUS}}] \text{ 整数 4} \end{array} \right\} \right\} \end{array} \right] \end{array} \right] .$$

書き方 2(レベル番号 02～48 の集団記述項)

レベル番号 [データ名 1]

$$\left[ \begin{array}{l} \underline{\text{LINE NUMBER IS}} \quad \left\{ \begin{array}{l} \text{データ名 2} \quad [\underline{\text{PLUS}} \text{ 整数 1}] \\ [\underline{\text{PLUS}} \text{ 整数 2}] \end{array} \right\} \\ \\ \left[ \begin{array}{l} \underline{\text{CLEAR}} \quad \left\{ \begin{array}{l} \underline{\text{SCREEN}} \\ \left\{ \begin{array}{l} \underline{\text{LINE}} \\ \underline{\text{DATA}} \end{array} \right\} \end{array} \right\} \quad \left[ \underline{\text{TO}} \quad \left\{ \begin{array}{l} \text{データ名 3} \quad [\underline{\text{PLUS}} \text{ 整数 3}] \\ [\underline{\text{PLUS}}] \text{ 整数 4} \end{array} \right\} \right\} \end{array} \right] \end{array} \right] .$$

書き方 3(レベル番号 02～49 の基本記述項)

レベル番号 [データ名 1]

$$\left[ \begin{array}{l} \text{LINE NUMBER IS} \left\{ \begin{array}{l} \text{データ名 2} \text{ [PLUS 整数 1]} \\ \text{[PLUS 整数 2]} \end{array} \right\} \end{array} \right]$$

$$\left[ \begin{array}{l} \text{CLEAR} \left\{ \begin{array}{l} \text{SCREEN} \\ \left\{ \begin{array}{l} \text{LINE} \\ \text{DATA} \end{array} \right\} \end{array} \right\} \left[ \text{TO} \left\{ \begin{array}{l} \text{データ名 3} \text{ [PLUS 整数 3]} \\ \text{[PLUS] 整数 4} \end{array} \right\} \right] \end{array} \right]$$

$$\left[ \begin{array}{l} \text{COLUMN NUMBER IS} \left\{ \begin{array}{l} \text{データ名 4} \text{ [PLUS 整数 5]} \\ \text{[PLUS 整数 6]} \end{array} \right\} \end{array} \right]$$

$$\left[ \begin{array}{l} \left\{ \begin{array}{l} \text{PICTURE} \\ \text{PIC} \end{array} \right\} \text{ IS 文字列} \left\{ \begin{array}{l} \text{FROM 一意名 1} \\ \text{INTO 一意名 2} \\ \text{USING 一意名 3} \end{array} \right\} \\ \left[ \left\{ \begin{array}{l} \text{PICTURE} \\ \text{PIC} \end{array} \right\} \text{ IS 文字列} \right] \text{ VALUE IS 定数 1} \end{array} \right]$$

[BLANK WHEN ZERO]

$$\left[ \left\{ \begin{array}{l} \text{JUSTIFIED} \\ \text{JUST} \end{array} \right\} \text{ RIGHT} \right]$$

$$\left[ \text{SHIFTED} \left[ \begin{array}{l} \text{RIGHT} \\ \text{LEFT} \end{array} \right] \right]$$

[BOX]  
[OVER LINE]  
[UNDER LINE]  
[BUZZER]

$$\left[ \left\{ \begin{array}{l} \text{[SECRET] [BLINK] [HIGH INTENSITY] [REVERSE]} \\ \text{PREVIOUS ATTRIBUTE} \end{array} \right\} \right]$$

$\left( \begin{array}{c} \text{COLOR IS} \end{array} \left\{ \begin{array}{c} \text{RED} \\ \text{GREEN} \\ \text{YELLOW} \\ \text{BLUE} \\ \text{MAGENTA} \\ \text{CYAN} \\ \text{WHITE} \end{array} \right\} \right)$

[WAIT FOR CONFIRMATION]

$\left( \begin{array}{c} \text{CHECK WITH} \end{array} \left\{ \begin{array}{c} \text{OVERFLOW} \\ \text{LENGTH} \\ \text{OMISSION} \\ \text{MODULUS 定数 2} \\ \text{BTAB KEY} \\ \text{ADVANCE KEY} \end{array} \right\} \dots \right)$

$\left( \begin{array}{c} \text{NO} \left\{ \begin{array}{c} \text{INITIAL FILLER CHARACTER} \\ \text{IFC} \end{array} \right\} \\ \left\{ \begin{array}{c} \text{INITIAL FILLER CHARACTER} \\ \text{IFC} \end{array} \right\} \text{ IS 定数 3} \end{array} \right)$

$\left( \begin{array}{c} \text{FILL CHARACTER IS} \end{array} \left\{ \begin{array}{c} \text{ZERO} \\ \text{SPACE} \\ \left\{ \begin{array}{c} \text{INITIAL FILLER CHARACTER} \\ \text{IFC} \end{array} \right\} \end{array} \right\} \right)$

$\left( \begin{array}{c} \text{INPUT MODE IS} \end{array} \left\{ \begin{array}{c} \text{MAT} \\ \text{HANDSCANNER} \\ \text{KANA-SHIFT} \\ \text{ID-CARD-READER} \\ \text{IO-OCR} \end{array} \right\} \right)$

[USAGE IS MIXED] .

書き方 4 (レベル番号 02～49 の基本記述項)

レベル番号 [データ名 1]

$$\left[ \begin{array}{l} \underline{\text{LINE NUMBER IS}} \left\{ \begin{array}{l} \text{データ名 2} \quad [\underline{\text{PLUS}} \text{ 整数 1}] \\ [\underline{\text{PLUS}} \text{ 整数 2}] \end{array} \right\} \\ \\ \underline{\text{CLEAR}} \left\{ \begin{array}{l} \underline{\text{SCREEN}} \\ \left\{ \begin{array}{l} \underline{\text{LINE}} \\ \underline{\text{DATA}} \end{array} \right\} \end{array} \right\} \left[ \underline{\text{TO}} \left\{ \begin{array}{l} \text{データ名 3} \quad [\underline{\text{PLUS}} \text{ 整数 3}] \\ [\underline{\text{PLUS}}] \text{ 整数 4} \end{array} \right\} \right] \end{array} \right]$$

[COLUMN NUMBER IS 整数 5]  
[OVER LINE TO 整数 6]  
[UNDER LINE TO 整数 7]  
[VERTICAL LINE AT COLUMN {整数 8} ... ]  
[OCCURS 整数 9 TIMES].

書き方 5 (レベル番号 02～49 の基本記述項)

レベル番号 [データ名 1]

$$\left[ \begin{array}{l} \underline{\text{LINE NUMBER IS}} \left\{ \begin{array}{l} \text{データ名 2} \quad [\underline{\text{PLUS}} \text{ 整数 1}] \\ [\underline{\text{PLUS}} \text{ 整数 2}] \end{array} \right\} \\ \\ \underline{\text{CLEAR}} \left\{ \begin{array}{l} \underline{\text{SCREEN}} \\ \left\{ \begin{array}{l} \underline{\text{LINE}} \\ \underline{\text{DATA}} \end{array} \right\} \end{array} \right\} \left[ \underline{\text{TO}} \left\{ \begin{array}{l} \text{データ名 3} \quad [\underline{\text{PLUS}} \text{ 整数 3}] \\ [\underline{\text{PLUS}}] \text{ 整数 4} \end{array} \right\} \right] \end{array} \right]$$

### 22.3.2 書き方 1、書き方 2 と書き方 3、4 または 5 の組み合わせ

書き方 1、書き方 2 と書き方 3、4 または 5 は次のように組み合わせて記述しなければならない。

|   |               |   |   |
|---|---------------|---|---|
| [ | 書き方 1         | ] | … |
|   | 書き方 2         |   |   |
| [ | 書き方 3、4 または 5 | ] |   |
|   | 書き方 2         |   |   |
|   | 書き方 3、4 または 5 |   |   |

1 つの画面データ記述項は、まず書き方 1 が最初に記述される必要がある。

1 つの画面データ記述項には、少なくとも 1 つの書き方 3、4 または 5 (基本記述項) が記述されなければならない。

書き方 1 と書き方 3、4 または 5 の間に書き方 2 は複数個記述してもかまわない。

書き方 1、書き方 2 と書き方 3、4 または 5 の組み合わせにおいて LINE NUMBER 句の指定は、上位のレベル(書き方 1 または書き方 2)で指定されている場合、それに従属する下位のレベルで指定することはできない。LINE NUMBER 句のない下位のレベルの行位置はその上位のレベルで指定されたものが有効となる。

### 22.3.3 画面データ記述項の各句の分類

画面データ記述項の各句は、画面上の位置、表示形式、動作などを定義する。それら各種指定は以下のように分類されている。

| 分 類      | 説 明                                                                   | 属する句                                                                   |
|----------|-----------------------------------------------------------------------|------------------------------------------------------------------------|
| 行指定      | 画面最上行を 1 とした場合の行位置を指定する。                                              | LINE NUMBER 句                                                          |
| 桁指定      | 画面最左端を 1 とした場合の桁位置を指定する。                                              | COLUMN NUMBER 句                                                        |
| クリア指定    | 画面上に表示されているデータやけい線の消去を指定する。                                           | CLEAR 句                                                                |
| 入力チェック指定 | 指定された項目に対する入力処理で行うチェックを指定する。                                          | CHECK 句                                                                |
| 動作指定     | 指定された項目の入力または出力に伴う画面上の動作を指定する。<br>この指定は指定された項目の入力または出力時点で 1 回のみ有効である。 | BUZZER 句, SECRET 句,<br>WAIT FOR CONFIRMATION 句                         |
| けい線指定    | 指定された項目にけい線を指定する。                                                     | BOX 句, OVER LINE 句,<br>UNDER LINE 句                                    |
|          | 入力または出力項目とは独立してけい線を指定する。                                              | OVER LINE TO 句,<br>UNDER LINE TO 句,<br>VERTICAL LINE AT 句,<br>OCCURS 句 |



|            |                                    |                                                                               |
|------------|------------------------------------|-------------------------------------------------------------------------------|
| 色指定        | 指定された項目の画面上の各種属性を指定する。             | BLINK 句, COLOR 句, PREVIOUS ATTRIBUTE 句, REVERSE 句                             |
| PICTURE 指定 | 画面上に占める桁数や表現形式などを指定する。             | FROM 句, INTO 句, PICTURE 句, USAGE 句, USING 句, VALUE 句                          |
| データ変換指定    | 指定された項目の入力または出力に伴うデータに施される処理を指定する。 | BLANK WHEN ZERO 句, FILL 句, INITIAL FILLER CHARACTER 句, JUSTIFIED 句, SHIFTED 句 |

## 22.4 行指定

行指定は、画面データ記述項において次の形式で指定する。

$$\left[ \underline{\text{LINE NUMBER IS}} \left\{ \begin{array}{l} \text{データ名 1} \text{ [PLUS 整数 1]} \\ \text{[PLUS 整数 2]} \end{array} \right\} \right]$$

| 指定方法    | 書き方                    | 説 明                                                                  |
|---------|------------------------|----------------------------------------------------------------------|
| 絶対行番号指定 | 整数 2                   | 整数 2(1～24 行)で指定された値を行の値として用いる。                                       |
|         | データ名 1                 | データ名 1(整数基本項目)で示される項目の内容を行の値として用いる。                                  |
|         | データ名 1<br>PLUS<br>整数 1 | データ名 1(整数基本項目)で示される項目の内容に整数 1を加えた値を行の値として用いる。データ名 1で示される項目の内容は変わらない。 |
| 相対行番号指定 | PLUS<br>整数 2           | 直前に入出力が行われた項目の先頭の行位置に整数 2を加えた値を行の値として用いる。                            |
| 省略      | LINE NUMBER 句を記述しない。   | 上位のレベルで指定されている LINE NUMBER 句の指定が有効となる(上位のレベルで必ず指定されていなければならない)。      |

### 注意

データ名の値はゼロや負の値であってはならない。また、入出力フィールドが画面内からはみ出すような指定をしてはならない。値が不正な場合、実行時エラーとなる。

## 22.5 桁指定

桁指定は、画面データ記述項において次の形式で指定する。

$$\left[ \text{COLUMN NUMBER IS} \left\{ \begin{array}{l} \text{データ名 1 [PLUS 整数 1]} \\ \text{[PLUS 整数 2]} \end{array} \right\} \right]$$

| 指定方法    | 書き方                    | 説明                                                                      |
|---------|------------------------|-------------------------------------------------------------------------|
| 絶対桁番号指定 | 整数 2                   | 整数 2(1～80 桁)で指定された値を桁の値として用いる。                                          |
|         | データ名 1                 | データ名 1(整数基本項目)で示される項目の内容を桁の値として用いる。                                     |
|         | データ名 1<br>PLUS<br>整数 1 | データ名 1(整数基本項目)で示される項目の内容に整数 1 を加えた値を桁の値として用いる。データ名 1 で示される項目の内容は変わらない。  |
| 相対桁番号指定 | PLUS<br>整数 2           | 直前に入出力が行われた項目の先頭の桁位置に整数 2 を加えた値を桁の値として用いる。                              |
| 省略      | COLUMN NUMBER 句を記述しない。 | 同一行で直前の記述項の領域の直後の桁位置が指定されているとみなす。<br>ある行の最初の記述項で COLUMN 句がないときは 1 とみなす。 |

### 注意

データ名の値はゼロや負の値であってはならない。また、入出力フィールドが画面内からはみ出すような指定をしてはならない。値が不正な場合、実行時エラーとなる。

## 22.6 クリア指定

クリア指定は、画面データ記述項において次の形式で指定する。



### (1) クリアの指定と動作

| 指定        | 動 作                                                                                                  |
|-----------|------------------------------------------------------------------------------------------------------|
| SCREEN 指定 | 全画面(1～24 行)のデータおよびけい線指定、色指定を消去する。                                                                    |
| LINE 指定   | CLEAR 句が指定されている画面データ記述項の行のデータおよびけい線指定、色指定を消去する。<br>また TO 句以下を指定すると、この記述項の行から消去する終了行を指定することができる。      |
| DATA 指定   | CLEAR 句が指定されている画面データ記述項の行のデータを消去する。けい線指定、色指定は消去されない。<br>また TO 句以下を指定すると、この記述項の行から消去する終了行を指定することができる。 |

### (2) 終了行の指定

TO 以下の句は、指定の方法によりおのおの次のように消去する終了行が決定される。

| 指定方法    | 書き方                    | 説 明                                                                                                                                                                                              |
|---------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 絶対行番号指定 | 整数 2                   | <ul style="list-style-type: none"><li>整数 2(1～24 行)で指定された行を終了行とする。</li><li>整数 2 の値は CLEAR 句の指定された画面データ記述項の行の値以上でなければならない。</li></ul>                                                               |
|         | データ名 1                 | <ul style="list-style-type: none"><li>データ名 1(整数基本項目)で示される項目の内容を終了行とする。</li><li>データ名 1 の値は CLEAR 句の指定された画面データ記述項の行の値以上でなければならない。</li></ul>                                                        |
|         | データ名 1<br>PLUS<br>整数 1 | <ul style="list-style-type: none"><li>データ名 1(整数基本項目)で示される項目の内容に整数 1 を加えた値を終了行とする。</li><li>データ名 1 で示される項目の内容は変わらない。</li><li>データ名 1 の値と整数 1 を加えた値は、CLEAR 句の指定された画面データ記述項の行の値以上でなければならない。</li></ul> |
| 相対行番号指定 | PLUS<br>整数 2           | <ul style="list-style-type: none"><li>CLEAR 句の指定された画面データ記述項の行の値に整数 2 を加えた値を終了行とする。</li></ul>                                                                                                     |
| 省略      | TO 句以下を記述しない。          | <ul style="list-style-type: none"><li>CLEAR 句の指定された画面データ記述項の行のみ 1 行をクリアする。</li></ul>                                                                                                             |

#### 注意

- TO 句以下の指定で、整数 2、データ名 1、またはデータ名 1 PLUS 整数 1 で指定された値が、画面データ記述項の行の値より小さい、またはゼロや負の値のとき、画面データ記

述項の行のみをクリアする。

## 22.7 入力チェック指定

入力時における各種チェックは、画面データ記述項において、次の形式で指定する。

**CHECK WITH** { **OVERFLOW**  
**LENGTH**  
**OMISSION**  
**MODULUS** 定数  
**BTAB KEY**  
**ADVANCE KEY** } ...

MODULUS 指定は、16 桁以下の数字項目に対してのみ指定できる。

### (1) チェックの指定と機能

| 指 定                            | 機 能                                                                                                                                          | 指定しないときの動作                                                    |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| OVERFLOW                       | 入力項目の画面上に占める桁数を超えるデータの <input type="text"/> を入力時エラー (OVERFLOW エラー) とし、終了キーの打鍵が必要となる。<br>画面上に占める桁数のすぐ右の桁で値が入力されたとき発生する。                       | 画面上に占める桁数を超えるデータの <input type="text"/> の入力は次の入力項目に対するデータとみなす。 |
| LENGTH                         | 入力項目の画面上に占める桁数と同じ桁のデータが入力されないと入力時エラー (LENGTH エラー) とし、同じ桁のデータ打鍵が必要となる。<br>COBOL WS エミュレータ配下の場合は、日本語項目および英数字、日本語混在指定の英数字項目に対する LENGTH 指定を無視する。 | 画面上に占める桁数未満のデータを入力した後、終了キーによって正常な入力となる。                       |
| OMISSION                       | 入力項目に対し終了キーだけの打鍵を入力時エラー (OMISSION エラー) とし、有効データの打鍵が必要となる。                                                                                    | 終了キーだけの打鍵を正常な入力終了として処理する。                                     |
| MODULUS<br>定数<br>(定数は10 または11) | 数字入力項目に対し、自己検査を行い、エラーのとき入力時エラー (CHKDGT エラー) とし、正しいチェックデジットデータの打鍵が必要となる。                                                                      | 自己検査は行われない。                                                   |
| BTAB                           | 入力項目に対し、 <b>Back space</b> キーの打鍵を入力時エラー (BTAB エラー) とし禁止する。                                                                                   | <b>Back space</b> キーの打鍵をエラーとせず、 <b>Back space</b> キーの動作をする。   |
| ADVANCE                        | 入力項目に対し、 <b>F10</b> キーの打鍵を入力時エラー (ADVANCE エラー) とし禁止する。COBOL WS エミュレータ配下の場合は、日本語項目および英数字、日本語混在指定の英数字項目に対する ADVANCE 指定を無視する。                   | <b>F10</b> キーの打鍵をエラーとしない。                                     |

上記表中機能にて示したエラー表示は、24 行(最下行)の 1 行下に以下の形式で表示されるため **Esc** キーによりエラー状態を解除し再入力すること。

| エラーの種類       | COBOL WS エミュレータ配下<br>でない表示文字列 | COBOL WS エミュレータ配下<br>の表示文字列 |
|--------------|-------------------------------|-----------------------------|
| OVERFLOW エラー | OVERFLOW                      | OVFLOW                      |
| LENGTH エラー   | LENGTH                        | LENGTH                      |
| OMISSION エラー | OMISSION                      | OMSSON                      |
| CHKDGT エラー   | CHKDGT                        | CHKDGT                      |
| BTAB エラー     | BTAB                          | ?                           |
| ADVANCE エラー  | ADVANCE                       | ?                           |

また、MODULUS 定数以外の指定は、指定された項目の画面表示から値を取り込む終了キー打鍵時は無効となる。どのキーがそれに該当するかは、「22.15 入力終了条件／状態」を参照。ただし、CHKDGT エラーは有効である。

(2)CHECK 句の指定の組み合わせ指定による動作

| 項目<br>打鍵データ                    | PICTURE 9(5)        |                     |        |                                |                              |
|--------------------------------|---------------------|---------------------|--------|--------------------------------|------------------------------|
|                                | Enter               | 123 Enter           | 12345  | 12345 Enter                    | 123456                       |
| CHECK 句指定なし                    | 正常終了                | 正常終了                | 正常終了   | 正常終了<br>(Enter は次の項目に対する打鍵となる) | 正常終了<br>('6' は次の項目に対する打鍵となる) |
| OMISSION                       | 省略エラー<br>(OMISSION) | 正常終了                | 正常終了   | 正常終了<br>(Enter は次の項目に対する打鍵となる) | 正常終了<br>('6' は次の項目に対する打鍵となる) |
| OVERFLOW                       | 正常終了                | 正常終了                | 終了キー待ち | 正常終了                           | オーバーフローエラー<br>(OVERFLOW)     |
| LENGTH                         | レンジスエラー<br>(LENGTH) | レンジスエラー<br>(LENGTH) | 正常終了   | 正常終了<br>(Enter は次の項目に対する打鍵となる) | 正常終了<br>('6' は次の項目に対する打鍵となる) |
| OMISSION<br>OVERFLOW           | 省略エラー<br>(OMISSION) | 正常終了                | 終了キー待ち | 正常終了                           | オーバーフローエラー<br>(OVERFLOW)     |
| OMISSION<br>LENGTH             | 省略エラー<br>(OMISSION) | レンジスエラー<br>(LENGTH) | 正常終了   | 正常終了<br>(Enter は次の項目に対する打鍵となる) | 正常終了<br>('6' は次の項目に対する打鍵となる) |
| OVERFLOW<br>LENGTH             | レンジスエラー<br>(LENGTH) | レンジスエラー<br>(LENGTH) | 終了キー待ち | 正常終了                           | オーバーフロー<br>(OVERFLOW)        |
| OMISSION<br>OVERFLOW<br>LENGTH | 省略エラー<br>(OMISSION) | レンジスエラー<br>(LENGTH) | 終了キー待ち | 正常終了                           | オーバーフロー<br>(OVERFLOW)        |
| 正常終了時の<br>入力項目の内容              | 00000               | 00123               | 12345  | 12345                          | 12345                        |

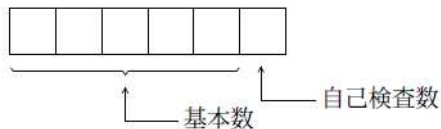
(3)自己検査の行われ方と自己検査数の算出法

MODULUS 定数を指定すると、次に説明する自己検査が行われる。

a. 項目と自己検査数の関係

自己検査を指定された項目では、右端桁の数値を自己検査数といい、右端桁を除いた数値を基本数という。

画面データ記述項の PICTURE 記述で 9(6) と指定した場合





## b. 自己検査の方法

基本数から一定の算出法(モジュラス 10 とモジュラス 11 によって異なる)によって算出した数と自己検査数として右端に入力された値が等しいか否かを調べる。

## c. モジュラス 10 のときの算出法

- i. 基本数の右端桁およびそれと 1 つおきの各桁(基本数の 1 位桁, 100 位桁, 10000 位桁, ……)に '2' を乗じる。
- ii. そのおのおのの積を構成する数字と基本数の中で '2' を乗じなかった数の和を求める。
- iii. その和を, それを含むより大きな 10 の整数倍の最小値から引く, 求められた差が自己検査数である。
- ii. の和が, たとえば 30 のとき自己検査数は,  
 $30 - 30 = 0$  で 0 となる。

例 基本数を 8765 とすると自己検査数は 0 となる。

$$\text{i. の計算} \quad \begin{array}{r} 5 \times 2 = 10 \\ 6 \quad 6 \\ 7 \times 2 = 14 \\ 8 \quad 8 \end{array}$$

$$\text{ii. の計算} \quad 1 + 0 + 6 + 1 + 4 + 8 = 20$$

$$\text{iii. の計算} \quad 20 - 20 = 0$$

この例では 6 桁の項目に 87650 が入力されれば正常に入力できる。

## d. モジュラス 11 のときの算出法

- i. 基本数の各桁に加重係数(基本数の 1 位桁から 2~7 を繰り返す)を乗じる。
- ii. 積の和を求める。
- iii. 和を 11 で割る。
- iv. 余りを 11 から引く, 求められた差が自己検査数である。

例 基本数を 98765 とすると自己検査数は 4 となる。

$$\text{i. の計算} \quad \begin{array}{r} 5 \times 2 = 10 \\ 6 \times 3 = 18 \\ 7 \times 4 = 28 \\ 8 \times 5 = 40 \\ 9 \times 6 = 54 \end{array}$$

$$\text{ii. の計算} \quad 10 + 18 + 28 + 40 + 54 = 150$$

$$\text{iii. の計算} \quad 150 \div 11 = 13 \text{ 余り } 7$$

$$\text{iv. の計算} \quad 11 - 7 = 4$$

## 備考

- (1) モジュラス 11 の自己検査では, 自己検査数が 11 となることがあるが, そのような基本数を使用することはできない(打鍵のときに, すべて MODULUS11 エラーとなる)。
- (2) モジュラス 11 の自己検査では, 自己検査数が 10 となることがあるが, このとき自己検査数は 0 となる。

例 基本数を 11345 とすると自己検査数は 0 となる。

$$\text{i. の計算} \quad \begin{array}{r} 5 \times 2 = 10 \\ 4 \times 3 = 12 \\ 3 \times 4 = 12 \end{array}$$

|         |                             |
|---------|-----------------------------|
|         | $1 \times 5 = 5$            |
|         | $1 \times 6 = 6$            |
| ii.の計算  | $10 + 12 + 12 + 5 + 6 = 45$ |
| iii.の計算 | $45 \div 11 = 4$ 余り 1       |
| iv.の計算  | $11 - 1 = 10$               |

## 22.8 動作指定

入力または出力項目の動作指定を次の中から指定できる。組み合わせて指定することもできる。

| 動作指定                  | 指定可能項目    | 機 能                                                                      |
|-----------------------|-----------|--------------------------------------------------------------------------|
| BUZZER                | 入力および出力項目 | 指定された項目の入力／表示前にブザーを鳴動させる。                                                |
| SECRET                | 入力項目      | 入力項目に対する打鍵データの表示を行わないようにする。<br>データの表示は行わないが、値を取り込む終了キーを押すと、打鍵データが取り込まれる。 |
| WAIT FOR CONFIRMATION | 出力項目      | 画面にデータが表示された後、表示確認の操作として、 <b>Esc</b> キーの打鍵を待たせる。                         |

## 22.9 けい線指定

### 22.9.1 指定された項目に対するけい線指定

入力または出力項目に対して、けい線指定を次の中から指定できる。

| けい線指定      | 指定可能項目    | 機 能              |
|------------|-----------|------------------|
| BOX        | 入力および出力項目 | 指定された項目を枠で囲む。    |
| OVER LINE  | 入力および出力項目 | 指定された項目に上線を表示する。 |
| UNDER LINE | 入力および出力項目 | 指定された項目に下線を表示する。 |

### 22.9.2 指定された項目とは独立したけい線指定

入力または出力項目とは独立して、けい線(オーバーライン、アンダーラインおよびバーチカルライン)を次の形式で指定する。

[OVER LINE TO 整数 1]  
[UNDER LINE TO 整数 2]  
[VERTICAL LINE AT COLUMN {整数 3} … ]  
[OCCURS 整数 4 TIMES].

OVER LINE TO 句、UNDER LINE TO 句または VERTICAL LINE AT 句のうち少なくとも 1 つは必ず指定しなければならない。

#### (1) けい線指定と機能

| けい線指定              | 機 能                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| OVER LINE TO 句     | <ul style="list-style-type: none"><li>COLUMN NUMBER 句で指定された桁から整数 1 で指定された桁までオーバーラインを表示する。</li><li>整数 1 の値は COLUMN NUMBER 句で指定した値より大きくなければならない。</li></ul> |
| UNDER LINE TO 句    | <ul style="list-style-type: none"><li>COLUMN NUMBER 句で指定された桁から整数 2 で指定された桁までアンダーラインを表示する。</li><li>整数 2 の値は COLUMN NUMBER 句で指定した値より大きくなければならない。</li></ul> |
| VERTICAL LINE AT 句 | <ul style="list-style-type: none"><li>整数 3…で指定される桁位置にバーチカルラインを表示する。</li><li>整数 3…の値は、COLUMN NUMBER 句で指定する値以上でなければならない。</li></ul>                         |

#### (2) 繰り返しの指定

OCCURS 句を指定することによって、OVER LINE TO 句、UNDER LINE TO 句および VERTICAL LINE AT 句で指定したけい線を複数行に表示することができる。

| OCCURS 句の指定有無 | 説 明                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------|
| 指定する          | OVER LINE TO 句、UNDER LINE TO 句および VERTICAL LINE AT 句で指定したけい線を、LINE NUMBER 句で指定された行から整数 4 で指定された行数分表示する。 |

|       |                                  |
|-------|----------------------------------|
| 指定しない | LINE NUMBER 句で指定された行だけにけい線を表示する。 |
|-------|----------------------------------|

**(3) けい線指定による表の作成**

けい線指定を利用して、画面上に簡単に表を作成することができる。

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

このような表を作成する場合には、次のように記述する。

**a. 記述方法 1**

```

01 TABLE1.
 02 LINE 2 COLUMN 1
 OVER LINE TO 20
 VERTICAL LINE AT 1 7 13 20
 OCCURS 5.
 02 LINE 7 COLUMN 1
 OVER LINE TO 20
 UNDER LINE TO 20
 VERTICAL LINE AT 1 7 13 20.

```

オーバーラインとバーチカルラインの組み合わせで 5 行繰り返し、最下行はオーバーライン、アンダーラインおよびバーチカルラインを組み合わせで指定し表を完成させる。

**b. 記述方法 2**

```

01 TABLE1.
 02 LINE 2 COLUMN 1
 OVER LINE TO 20
 UNDER LINE TO 20
 VERTICAL LINE AT 1 7 13 20.
 02 LINE 3 COLUMN 1
 UNDER LINE TO 20
 VERTICAL LINE AT 1 7 13 20
 OCCURS 5.

```

表の最上行をオーバーライン、アンダーラインおよびバーチカルラインの組み合わせで指定し、2 行目以降をアンダーラインとバーチカルラインの組み合わせで 5 行繰り返して表を完成させる。

**(4) 注意事項**

- a. けい線を使用して長方形を作る場合、オーバーラインまたはアンダーラインはバーチカルラインと同じ行で組み合わせで指定する。

正しい記述

```
01 TABLE1.
 02 LINE 2 COLUMN 1
 OVER LINE TO 10
 VERTICAL LINE AT 1 10.
 02 LINE 3 COLUMN 1
 UNDER LINE TO 10
 VERTICAL LINE AT 1 10.
```



誤った記述

```
01 TABLE1.
 02 LINE 2 COLUMN 1
 OVER LINE TO 10
 VERTICAL LINE AT 1 10.
 02 LINE 3
 VERTICAL LINE AT 1 10.
 02 LINE 4 COLUMN 1
 OVER LINE TO 10.
```



長方形の下横線を、4 行目のオーバーラインを使用して作成しようとすると、図のように両端がでる。

## 22.10 色指定

指定された項目の画面上の各種属性を指定する。

### 22.10.1 COLOR 句

COLOR 句の指定と表示色

| COLOR 句の指定 | 表示色      |
|------------|----------|
| RED        | 赤        |
| GREEN      | 緑        |
| YELLOW     | 黄        |
| BLUE       | 青        |
| MAGENTA    | マゼンタ(赤紫) |
| CYAN       | シアン(水色)  |
| WHITE      | 白        |

### 22.10.2 その他の色指定

| 色指定                | 指定可能項目    | 機 能                                                                                    |
|--------------------|-----------|----------------------------------------------------------------------------------------|
| BLINK              | 入力および出力項目 | COBOL WS エミュレータで起動する場合のみ有効である。<br>それ以外の場合は、メモとみなす。                                     |
| HIGH INTENSITY     | 入力および出力項目 | 本 COBOL ではメモ扱い。                                                                        |
| PREVIOUS ATTRIBUTE | 入力および出力項目 | 指定された項目以前に設定された REVERSE 指定、SECRET 指定をそのまま受け継ぐ。<br>CLEAR 句指定により消去されたときは、消去された後の状態を引き継ぐ。 |
| REVERSE            | 入力および出力項目 | 指定された項目をリバース(反転)させる。                                                                   |

22.11 PICTURE 指定

PICTURE(形式)記述は、入力領域および出力領域の大きさ、属性および表示する定数を与えることができる。さらに、実際に入出力データの対象としたい領域を指定する。

書き方 1(入力または出力項目の場合)

PICTURE

PIC

IS 文字列

FROM 一意名 1

INTO 一意名 2

USING 一意名 3

[[USAGE IS] MIXED]

書き方 2(定数を出力する場合)

PICTURE

PIC

IS 文字列

VALUE IS 定数

[[USAGE IS] MIXED]

書き方 1 において入力項目の場合、PICTURE 句が数字項目、日本語項目および日本語編集項目以外の場合、同じ桁数の英数字項目として取り扱われる。日本語編集項目のときは、同じ桁数の日本語項目として取り扱われる。

書き方 2 において PICTURE 句を省略できるのは、VALUE 句の定数が英数字定数または日本語定数のときだけである。

この場合次の PICTURE 記述が指定されたものとみなして処理する。

PICTURE IS X(m) VALUE IS 英数字定数.  
PICTURE IS N(n) VALUE IS 日本語定数.  
(ここで m は英数字定数の桁数, n は日本語定数の桁数を意味する)

(1)FROM, INTO, USING, VALUE 句の画面手続き文における動作

| 句     | 手続き文                                                       |                                                             |
|-------|------------------------------------------------------------|-------------------------------------------------------------|
|       | ACCEPT                                                     | DISPLAY                                                     |
| FROM  | 動作の対象とならない。                                                | 手続き文にて指定された画面データ記述項の領域へ FROM 句で指定された一意名からデータを転記した後、画面へ出力する。 |
| INTO  | 手続き文にて指定された画面データ記述項の領域の入力領域に入力後 INTO 句で指定された一意名へデータを転記する。  | 動作の対象とならない。                                                 |
| USING | 手続き文にて指定された画面データ記述項の領域の入力領域に入力後 USING 句で指定された一意名へデータを転記する。 | 手続き文にて指定された画面データ記述項の領域へ USING 句で指定された一意名からデータを転記した後、画面      |



|       |             |                                           |
|-------|-------------|-------------------------------------------|
|       |             | へ出力する。                                    |
| VALUE | 動作の対象とならない。 | 手続き文にて指定された画面データ記述項の領域に初期設定された定数を画面へ出力する。 |

(2)英数字・日本語混在指定

英数字基本項目に、英数字・日本語混在データを格納するときには、USAGE IS MIXED 句を指定する。その際、日本語 1 文字を英数字 2 文字として扱う。

### 22.11.1 数字項目に対する入力／出力時の画面上の桁数

画面データの記述項で数字項目を定義し、画面上で入力／出力を行うとき、使用する画面上の桁数は次のようになる。

| PICTURE    | 動 作           |           |                                                    |
|------------|---------------|-----------|----------------------------------------------------|
|            | 入 力           | 出 力       | 説 明                                                |
| 9(n)       | n             | n         |                                                    |
| S9(n)      | n + 1         | n + 1     | 入出力のとき、n + 1 桁目に符号桁が確保される。<br>正のとき…"△"<br>負のとき…"—" |
| 9(n)V9(m)  | n + m + 1     | n + m     | 入力のとき、n + 1 桁目に小数点桁が確保される。<br>出力のとき、小数点桁は確保されない。   |
| S9(n)V9(m) | n + m + 1 + 1 | n + m + 1 | 入力のとき、小数点桁と符号桁が確保される。<br>出力のとき、符号桁のみが確保される。        |

入力項目の画面上の桁数が、実際のデータの桁数より増える場合(符号／小数点が付く場合)、符号や小数点を指定せずに終了すると、余分なデータ入力の桁落ちや LENGTH エラーとなる。たとえば、S9(6)V9(2)の項目に対し 7 桁以上の数字を入力し Enter キーを押すと上位桁が取り除かれ、下 6 桁を表示する。

- カスタマイズにて「数字項目入力時の桁数チェック方式を変更する」を指定することで次の入力チェックを行う。Enter キーまたは小数点の打鍵時、整数部の桁数が PICTURE 句に記述した整数部の桁数より大きい場合、入力時エラー(?エラー)とする。
- Enter キーの打鍵時、小数部の桁数が PICTURE 句に記述した小数部の桁数より大きい場合、入力時エラー(?エラー)とする。
- 整数部のみの場合、小数点の打鍵を入力時エラー(?エラー)とする。
- 符号なしの場合、符号文字[+]と[-]の入力を入力時エラー(NUMERIC エラー)とする。

エラー表示は 24 行(最下行)の 1 行下に表示する。 **Esc** キーによりエラー状態を解除し再入力すること。

例 画面データ記述

01 A.

02 B LINE NUMBER IS 5  
 COLUMN NUMBER IS 10  
 PICTURE 文字列 USING X.

| 動作 | 文 字 列           | データ            | 画面上の表示                                                                    | 画面上<br>の桁数 |
|----|-----------------|----------------|---------------------------------------------------------------------------|------------|
| 入力 | 9 (5)           | 1 2 3 4 5      | <span style="border: 1px solid black; padding: 2px;">1 2 3 4 5</span>     | 5          |
|    | S 9 (5)         | 1 2 3 (正)      | <span style="border: 1px solid black; padding: 2px;">0 0 1 2 3 △</span>   | 6          |
|    |                 | - 1 2 3 4      | <span style="border: 1px solid black; padding: 2px;">0 1 2 3 4 -</span>   | 6          |
|    | 9 (3) V 9 (2)   | 1 2 3. 4 5     | <span style="border: 1px solid black; padding: 2px;">1 2 3 . 4 5</span>   | 6          |
|    | S 9 (3) V 9 (2) | 1 2 3. 4 5 (正) | <span style="border: 1px solid black; padding: 2px;">1 2 3 . 4 5 △</span> | 7          |
|    |                 | - 1 2 3. 4 5   | <span style="border: 1px solid black; padding: 2px;">1 2 3 . 4 5 -</span> | 7          |
| 出力 | 9 (5)           | 1 2 3          | <span style="border: 1px solid black; padding: 2px;">0 0 1 2 3</span>     | 5          |
|    | S 9 (5)         | + 1 2 3 4      | <span style="border: 1px solid black; padding: 2px;">0 1 2 3 4 △</span>   | 6          |
|    |                 | - 1 2 3 4      | <span style="border: 1px solid black; padding: 2px;">0 1 2 3 4 -</span>   | 6          |
|    | 9 (3) V 9 (2)   | 1 2 3. 4 5     | <span style="border: 1px solid black; padding: 2px;">1 2 3 4 5</span>     | 5          |
|    | S 9 (3) V 9 (2) | + 1 2 3. 4 5   | <span style="border: 1px solid black; padding: 2px;">1 2 3 4 5 △</span>   | 6          |
|    |                 | - 1 2 3. 4 5   | <span style="border: 1px solid black; padding: 2px;">1 2 3 4 5 -</span>   | 6          |

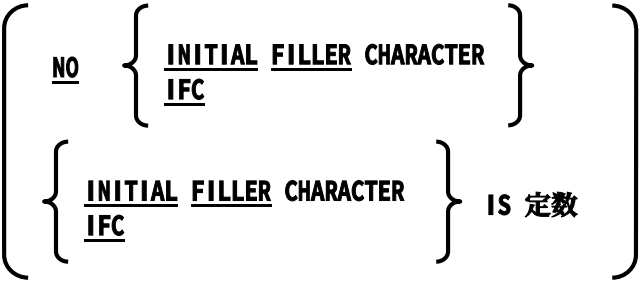
## 22.12 データ変換指定

データ変換指定には、次のものがある。

- 初期表示文字の指定(INITIAL FILLER CHARACTER 句)
- 補充文字の指定(FILL 句)
- 桁合わせの指定(SHIFTED 句)
- ゼロならば空白指定(BLANK WHEN ZERO 句)
- 桁寄せ指定(JUSTIFIED 句)

### (1) 初期表示文字の指定(INITIAL FILLER CHARACTER 句)

初期表示文字とは、入力領域に表示される文字で、入力領域の位置に入力する桁数だけ表示され、打鍵と同時に打鍵した文字と置き換えられる文字をいう。また指定により初期表示文字を表示しなくすることもできる。



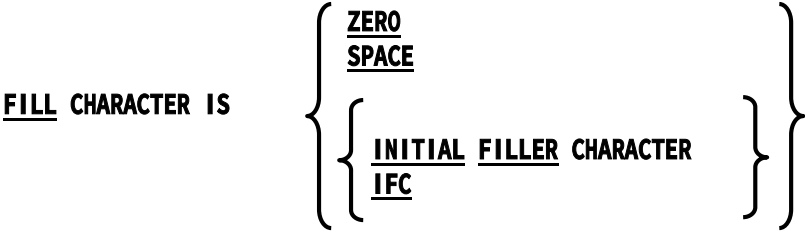
#### a. 定数の値による動作の違い

| INITIAL FILLER CHARACTER の指定 | 初期表示文字および動作説明                                                                                                                                                                                                            |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 句を記述しない                      | 数字項目以外の場合、^ (アクセサンシルコンフлекс)が初期表示文字として表示される。<br>数字項目の場合、_(アンダーライン)が初期表示文字として表示される。<br><b>Tab</b> キーによるスキップ操作は行えない。                                                                                                       |
| NO 指定                        | 初期表示文字は表示されず、入力領域には以前の画面上のデータが残っている。<br>この時、 <b>Back space</b> キーが押された状態でも <b>Tab</b> キーによるスキップ操作は行える。                                                                                                                   |
| 定数指定                         | 指定された定数を初期表示文字として表示する。<br><b>Tab</b> キーによるスキップ操作は行えない。<br>COBOL WS エミュレータ配下の場合は、日本語項目および英数字、日本語混在指定の英数字項目に対する英数字定数指定を無視する。<br>日本語項目には 2 バイトの^ (アクセサンシルコンフлекс)を初期表示する。英数字、日本語混在指定の英数字項目には1バイトの^ (アクセサンシルコンフлекс)を初期表示する。 |

スキップ操作が行えない項目で **Tab** キーを打鍵すると、COBOL WS エミュレータ配下でない場合 SKIP エラー、COBOL WS エミュレータ配下の場合ステーション入力ミス ‘?’ となるので、**Esc** キーによりエラー状態を解除し再入力すること。

(2)補充文字の指定(FILL 句)

補充文字は、入力されるデータの桁数が画面データ記述項の PICTURE 句で指定した入力領域の桁数よりも少ないときに、入力領域の残りの部分に詰める文字のことをいう。



a. 指定と動作

| FILL 句の指定                           | 動 作                                                                                                        |
|-------------------------------------|------------------------------------------------------------------------------------------------------------|
| 句を記述しない                             | 数字項目以外の場合、SPACE が指定されたものとみなして空白を詰める。<br>数字項目の場合、ZERO が指定されたものとみなして 0(ゼロ)を詰める。                              |
| ZERO                                | 0(ゼロ)を詰める。                                                                                                 |
| SPACE                               | 空白を詰める。<br>数字項目に FILL SPACE が指定されると、画面上は空白が表示されるが、値としては 0(ゼロ)がとられる。                                        |
| INITIAL FILLER CHARACTER<br>または IFC | 初期表示文字(INITIAL FILLER CHARACTER)句で指定された文字を詰める。日本語項目には 2 バイトの空白文字を補充する。英数字、日本語混在指定の英数字項目には 1 バイトの空白文字を補充する。 |

COBOL WS エミュレータ配下の場合、日本語項目および英数字、日本語混在指定の英数字項目に対する IFC 指定を無視する。

(3)桁合わせの指定(SHIFTED 句)

入力動作を行ったとき入力されたデータを入力領域に右詰めに格納するか、左詰めに格納するかを指定する。



## a. 指定と入力領域における動作

| SHIFTED 句の指定                                       | 入力領域の項目 |                            |                                      |                                           |
|----------------------------------------------------|---------|----------------------------|--------------------------------------|-------------------------------------------|
|                                                    | 数字項目以外  | 数字項目                       |                                      |                                           |
|                                                    |         | 想定小数点なし<br>(PIC 文字列に V なし) | 想定小数点あり (PIC 文字列に V あり)              |                                           |
|                                                    |         |                            | 入力データに小数点<br>(.)あり                   | 入力データに小数点(.)なし                            |
| 句を記述しない<br>(SHIFTED 句を書かない)                        | 左詰め     | 右詰め                        | 想定小数位置 (V) と<br>入力データの小数点<br>による桁合わせ | 入力データを整数<br>部分に対するデー<br>タとみなす<br>(小数部はゼロ) |
| SHIFTED 句を書いて RIGHT /<br>LEFT 指定を省略または RIGHT<br>指定 | 右詰め     | 右詰め                        | 想定小数位置 (V) と<br>入力データの小数点<br>による桁合わせ | PIC 文字列の V を無<br>視して小数部の右<br>まで詰める        |
| LEFT 指定                                            | 左詰め     | 左詰め                        | 無視される<br>(SHIFTED 句を省略し<br>たときに同じ)   | 無視される<br>(SHIFTED 句を省略<br>したときに同じ)        |

**注意** COBOL WS エミュレータ配下の場合は、日本語項目および英数字、日本語混在指定の英数字項目に対する RIGHT 指定を無視する。左詰めに格納する。

## (4) ゼロならば空白指定 (BLANK WHEN ZERO 句)

BLANK WHEN ZERO 句の指定は PICTURE 記述の FROM 句で指定された一意名の内容が画面データ記述項の出力領域へ転記されるときに有効となる。これらの指定の動作は、通常の転記 (MOVE) 文と同じである。

## (5) 桁寄せ指定 (JUSTIFIED 句)

JUSTIFIED 句の指定は、PICTURE 記述の FROM 句または USING 句で指定された一意名の内容が画面データ記述項の出力領域へ転記されるときに有効となる。これらの指定の動作は、通常の転記 (MOVE) 文と同じである。

## 22.13 キー入力時の動作

データ入力時に関する注意すべき動作を以下にまとめる。

- 各入力項目の始めは、上書きモードに設定される。
- 各項目内でのカーソル移動は、英数字、日本語を問わず、1 文字単位で移動する。
- 各項目内での文字の削除は、英数字、日本語を問わず、1 文字単位で削除される。
- 各項目内での文字の挿入は、英数字、日本語を問わず、1 文字単位で挿入される。
- キーボードから入力可能な値は、その項目属性(PICTURE 句や USAGE 句の指定)による。
 

|           |                        |
|-----------|------------------------|
| 英字        | 1 バイトで表される表示可能な文字      |
| 英数字       | 1 バイトで表される表示可能な文字      |
| 数字        | '0'から'9', 小数点文字, 正負の符号 |
| 日本語       | 2 バイトで表されるすべての文字       |
| 英数字・日本語混在 | 英数字と日本語の条件を満たす文字       |
- カーソルの移動可能範囲は、項目の入力桁数のすぐ右の 1 桁までである。
- 挿入モードで文字が入力されたとき、桁あふれ条件に当てはまらないときは、あふれる文字は項目の右端から文字単位で文字落ちする。
- 文字の削除や文字落ちによる右端の空き領域には、空白が補われる。
- 上書きモードで、日本語文字を英数字で置き換えるとき、日本語文字は 2 文字の空白のように扱われ、上書きされなかった 1 バイトは空白となる。

## 22.14 データの修正方法

画面制御機能を使用してデータの入力を行っているとき、打鍵したデータを修正したい場合がある。その修正したい箇所によって次の4種類に分けられる。

- 入力中の項目
- 実行中の手続き文の中で、すでに入力した項目
- 以前の手続き文で入力終了した項目(画面上に残っている場合)
- 以前の手続き文で入力終了した項目(画面上に残っていない場合)

### (1) データ修正に使用するキー

データ修正のために使用するキーとして次のものがある。

| 機能キー       | 動 作                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Home       | <ul style="list-style-type: none"> <li>• カーソルがその項目の左端に戻り、打鍵してあったデータは消される。</li> </ul>                                                                                                                                                                                                                                                                                        |
| Back space | <ul style="list-style-type: none"> <li>• カーソルが項目の途中(左端以外)にあるときは、その項目の左端にカーソルが戻る。</li> <li>• カーソルが項目の左端にあるときに押すとその実行されている文で直前の入力項目の左端にカーソルが移動する。このとき <b>Back space</b> キーの押された項目が、実行されている文の最初の入力項目のときは、その文は終了し END STATUS データ項目には <b>Back space</b> キーを示す'09'が通知される。</li> <li>• カーソルが移動または手続き文が終了するだけで画面上に表示されているデータは変化しない。</li> </ul>                                       |
| →          | <ul style="list-style-type: none"> <li>• カーソルが右方向へ1桁移動する。項目の右端の右隣よりは進まない。</li> <li>• カーソルが移動するのみで、画面上に表示されているデータは変化しない。</li> <li>• 入力動作が終了することはない。</li> </ul>                                                                                                                                                                                                               |
| ←          | <ul style="list-style-type: none"> <li>• カーソルが左方向へ1桁移動する。項目の左端にカーソルがあるときは押しても動作しない。</li> <li>• カーソルが移動するのみで、画面上に表示されているデータは変化しない。</li> <li>• 入力動作が終了することはない。</li> </ul>                                                                                                                                                                                                     |
| Tab        | <ul style="list-style-type: none"> <li>• 画面上の入力項目に表示されているデータを入力とみなして入力する。</li> <li>• <b>Back space</b> キーが打鍵されていない状態または INITIAL FILLER CHARACTER 句で NO を指定せず定数を指定したとき押すと、COBOL WS エミュレータ配下でない場合 SKIP エラー、COBOL WS エミュレータ配下の場合ステーション入力ミス '?' が表示され、<b>Tab</b> キーは押せない(すなわち、初期表示文字の表示されている項目では打鍵できない)。</li> <li>• SHIFTED 句の RIGHT 指定と CHECK 句の OVERFLOW 指定は無視される。</li> </ul> |
| Delete     | <ul style="list-style-type: none"> <li>• カーソル位置の文字を1文字削除し、以後の文字を項目内で左に詰め、項目の右端には空白を埋める。</li> <li>• 入力動作が終了することはない。</li> </ul>                                                                                                                                                                                                                                               |
| Insert     | <ul style="list-style-type: none"> <li>• このキーを押すと挿入モードになる。データを入力するとカーソル位置の文字も含めて、それより右の文字は1文字右に移動し、入力データが挿入される。カーソルも1桁右へ移動する(挿入モードか上書きモードかを 24 行(最下行)の1行下の左端に表示している。)</li> </ul>                                                                                                                                                                                            |

- 入力動作が終了することはない。
- 挿入モードから上書きモードに戻すときは、再度、本キーを押す。

## (2) 入力中の項目の修正

X(5)に対する入力に対し‘ABC’を打鍵するのを誤って‘ANC’を押してしまった。

画面上は、ANC となっている。

↑  
カーソル位置

修正するには、次のような方法がある。

## 方法 1.

Back space を押し ‘ABC’ を押し直す。

## 方法 2.

Back space → を押し ‘B’ を押し直し、→ を押す。

## 方法 3.

← ← を押し ‘B’ を押し直し、→ を押す。

## 方法 4.

Home を押し ‘ABC’ を押し直す。

## (3) 実行中の手続き文の中で、すでに入力した項目の修正

9(5) X(3) 9(4)に対する入力に対し‘123’ ‘AB’ を打鍵するのを誤って‘153’ ‘AB’ を押してしまった。

画面上は 00153 AB△ ----- となっている。

↑  
カーソル位置

修正するには、次のような方法がある。

## 方法 1.

Back space Back space を押し(カーソルは最初の項目の左端に戻る)その後 ‘123’ を押し、X(3)の項目の入力をスキップするため、Tab を押す。

## 方法 2.

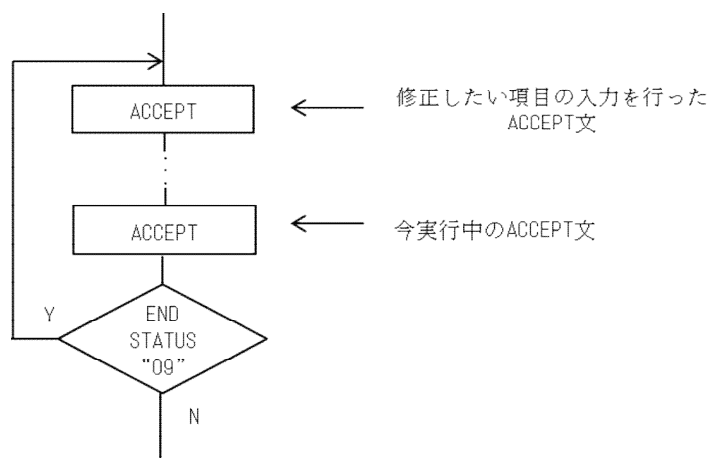
Back space Back space を押し、→ → → を押し、‘2’ Tab Tab と押す。

## (4) 以前の手続き文で入力終了した項目(画面上に残っている場合)の修正

修正するため、カーソルを戻して修正する可能性のある一連の入力項目は、1回のACCEPT文によって実行すべきである。

しかし、どうしても処理上1回のACCEPT文で実行できないときは、数回のACCEPT文に分割することになる。このとき以前に終了したACCEPT文で押したデータを修正するには、今実行中のACCEPT文を1度 Back space キー打鍵状態で終了させる(END STATUS データ項目の値が‘09’)。その後修正したい項目の入力を行ったACCEPT文を再度実行し、修正しACCEPT文を終了させた後、1度 Back space キー打鍵状態で終了させたACCEPT文に戻り処理を続ける。





#### (5) 以前の手続き文で入力終了した項目(画面上に残っていない場合)の修正

修正したい項目が画面上に表示されていないとき、または入力動作終了時と異なる状態で画面上に表示されている(入力動作終了直後に数字編集を行って再表示したような場合)とき、修正するためには、おのおの入力項目の PICTURE 記述に対し次に示す PICTURE 記述にて出力した後、入力を行えば、1.の機能キーが使用できる。

| 字類  | 入力の PICTURE 指定 | 出力の PICTURE 指定                                  |
|-----|----------------|-------------------------------------------------|
| 英数字 | X(n)           | X(n)                                            |
| 英字  | A(n)           | A(n)                                            |
| 数字  | [S]9(n)        | 9(n)[-]<br>Z(n-1)9[-]<br>Z(n)[-]                |
|     | [S]9(m)V9(n)   | 9(m).9(n)[-]<br>Z(m-1)9.9(n)[-]<br>Z(m).9(n)[-] |

m, n は桁数を表す。

角カッコ[]は符号の有無を表し、入力が'S'があれば、'-'が必要なことを表している。

#### 例

'S9(3)V9(2)'に対し修正したいデータを出力するには、'9(3).9(2)-'、'Z(2)9.9(2)-'または'Z(3).9(2)-'にて行える。

#### 初期表示文字を出力しない方法

このような処理を行うには、初期表示文字を出力しないようにするため、次のいずれかの方法でプログラミングしなければならない。

- 入力項目の画面データ記述項の INITIAL FILLER CHARACTER 句で NO を指定する。
- BTAB POSITION データ項目(画面記述項)の内容に 0(ゼロ)以外の値を設定し、**Back space** キーを押した状態として、ACCEPT 文を実行する。

ただし、BTAB POSITION データ項目には画面上の入力項目の開始行や桁より大きな値を設定しなければならない。また、このような修正処理が終了したときは、BTAB POSITION データ項目に 0 を設定し画面手続き文を実行し、**Back space** キーを押した状態を解除しておかなければならない。

## 22.15 入力終了条件／状態

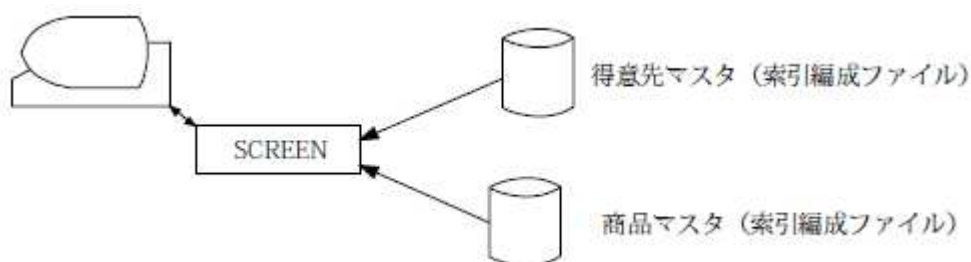
ACCEPT 文の入力項目に対する入力は、次の条件で終了し、終了したときの入力項目の状態は次に示すようになる。

| 使用キー                                                 | 終了の対象                           | 項目に対する動作／入力結果       |                                 |                                           |
|------------------------------------------------------|---------------------------------|---------------------|---------------------------------|-------------------------------------------|
|                                                      |                                 | 入力前の項目              | 入力項目                            | 入力後の項目                                    |
| Enter<br><div>Shift + F8</div> <div>Shift + F9</div> | 打鍵した項目のみ                        | すでに入力済み<br>で影響を受けない | カーソルの左から<br>値を取り込む              | 続いて動作する                                   |
| <div>Tab</div>                                       |                                 |                     | 画面表示から値を<br>取り込む                |                                           |
| <div>Shift</div> + Enter                             |                                 |                     | ACCEPT 文全体                      | カーソルの左から<br>値を取り込む                        |
| <div>F10</div>                                       | 値は取らない(入<br>力項目の内容は<br>以前のまま残る) |                     |                                 |                                           |
| その他の終了キー                                             |                                 |                     |                                 |                                           |
| <div>Back space</div><br>(最初の入力項目の先<br>頭での押下)        | ACCEPT 文全体                      | -----               | 入力されない(入<br>力項目の内容は<br>以前のまま残る) | 手続き文が終了<br>するため動作しない(入力項目の内容<br>は以前のまま残る) |

## 22.16 画面制御機能のプログラム例

次に今まで述べてきた画面制御機能を実際にプログラミングした場合、どのような例を掲げて説明する。あくまでも例として作成したプログラムであり、特に画面に対する操作部分のみをプログラミングしているため、実際の業務プログラムでは、マスタファイルの更新、伝票の印字、打鍵ミスに対する修正処理などが追加される。

### (1) プログラム処理概要



- 伝票 NO はプログラム内で加算し表示する。
- 日付, 得意先コードを入力し得意先マスタと突き合わせる。
- 伝票の明細行番号(1~5)はプログラムより表示する。
- 商品コード, 数量を入力し, 商品マスタと突き合わせる。
- 商品マスタの単価と数量を掛けて金額を計算する。
- 商品名, 単価, 金額を表示する。
- 1 伝票終了時(明細行 5 行または **F10** キーを押す)には, 1 伝票の金額の合計を表示しオペレータに確認させる。
- プログラムの終了は, 日付入力時 **F1** キーを押す。

## (2)画面レイアウト

行  
↓

|    |                  |                              |                |        |              |
|----|------------------|------------------------------|----------------|--------|--------------|
| 1  | 伝票NO. XXXXX      |                              | XX 年 XX 月 XX 日 |        |              |
| 3  | 得意先コード<br>XXXXXX | 得意先名<br>XXXXXXXXXXXXXXXXXXXX |                |        |              |
| 6  | 商品コード            | 商品名                          | 数 量            | 単 価    | 金 額          |
| 8  | 1 XXXXX          | XXXXXXXXXXXXXXXXXXXX         | XXXXX          | XXXXXX | XXXXXXXXXXXX |
| 9  | 2 XXXXX          | XXXXXXXXXXXXXXXXXXXX         | XXXXX          | XXXXXX | XXXXXXXXXXXX |
| 10 | 3 XXXXX          | XXXXXXXXXXXXXXXXXXXX         | XXXXX          | XXXXXX | XXXXXXXXXXXX |
| 11 | 4 XXXXX          | XXXXXXXXXXXXXXXXXXXX         | XXXXX          | XXXXXX | XXXXXXXXXXXX |
| 12 | 5 XXXXX          | XXXXXXXXXXXXXXXXXXXX         | XXXXX          | XXXXXX | XXXXXXXXXXXX |
| 18 |                  |                              |                | 合計     | XXXXXXXXXXXX |
| 22 | 確認 X             |                              |                |        |              |

```

000001 IDENTIFICATION DIVISION.
000002 PROGRAM-ID. PROG01.
000003 ENVIRONMENT DIVISION.
000004 CONFIGURATION SECTION.
000005 SOURCE-COMPUTER. Express5800.
000006 OBJECT-COMPUTER. Express5800.
000007 INPUT-OUTPUT SECTION.
000008 FILE-CONTROL.
000009 SELECT TOKUI-M ASSIGN TO TOKUI-MSD
000010 ORGANIZATION IS INDEXED
000011 ACCESS MODE IS RANDOM
000012 RECORD KEY IS T-CODE.
000013 SELECT SYOHIN-M ASSIGN TO SYOHIN-MSD
000014 ORGANIZATION IS INDEXED
000015 ACCESS MODE IS RANDOM
000016 RECORD KEY IS S-CODE.
000017*
000018 DATA DIVISION.
000019 FILE SECTION.
000020 FD TOKUI-M
000021 LABEL RECORD IS STANDARD
000022 VALUE OF IDENTIFICATION "TOKUI".
000023 01 TOKUI-R.
000024 02 T-CODE PICTURE X(6).
000025 02 T-NAME PICTURE X(20).
000026 02 FILLER PICTURE X(14).
000027*
000028 FD SYOHIN-M
000029 LABEL RECORD IS STANDARD
000030 VALUE OF IDENTIFICATION "SYOHIN".
000031 01 SYOHIN-R.
000032 02 S-CODE PICTURE X(6).
000033 02 S-NAME PICTURE X(20).
000034 02 S-TANKA PICTURE 9(5).
000035 02 FILLER PICTURE X(9).
000036*
000037 WORKING-STORAGE SECTION.
000038 77 D-LINE PICTURE 9.
000039 77 DENPYO-NO PICTURE 9(5) USAGE COMP-3 VALUE ZEROS.
000040 01 IN-DATE.
000041 02 IN-YY PICTURE 99.
000042 02 IN-MM PICTURE 99.
000043 02 IN-DD PICTURE 99.
000044 77 SURYO PICTURE 9(5).
000045 77 KINGAKU PICTURE 9(9).
000046 77 GOKEI PICTURE 9(9).
000047 77 KAKUNIN PICTURE X.
000048 SCREEN SECTION.
000049 SD SCR-D
000050 END STATUS IS END-STs.
000051 01 SCR-01
000052 CLEAR SCREEN.
000053 02
000054 LINE NUMBER IS 1 COLUMN NUMBER IS 2
000055 VALUE IS NC"伝票NO. ".
000056 02

```

```

000057 LINE NUMBER IS 3.
000058 03
000059 COLUMN NUMBER IS 2
000060 VALUE IS NC"得意先コード".
000061 03
000062 COLUMN NUMBER IS 20
000063 VALUE IS NC"得意先名".
000064 02
000065 LINE NUMBER IS 6.
000066 03
000067 COLUMN NUMBER IS 4 VALUE IS
000068 NC"商品コード 商品名".
000069 03
000070 COLUMN NUMBER IS 44
000071 VALUE IS
000072 NC"数 量 単 価 金 額".
000073*
000074 01 SCR-02.
000075 02
000076 LINE NUMBER IS 1 COLUMN NUMBER IS 12
000077 PICTURE IS 9(5) FROM DENPYO-NO.
000078 02
000079 LINE NUMBER IS 1 COLUMN NUMBER IS 64
000080 VALUE IS NC"年 月 日".
000081 02
000082 LINE NUMBER IS 1.
000083 03
000084 COLUMN NUMBER IS 62
000085 PICTURE IS 99 INTO IN-YY.
000086 03
000087 COLUMN NUMBER IS PLUS 4
000088 PICTURE IS 99 INTO IN-MM.
000089 03
000090 COLUMN NUMBER IS PLUS 4
000091 PICTURE IS 99 INTO IN-DD
000092 CHECK WITH OVERFLOW.
000093 01 SCR-04.
000094 02 SCR-05
000095 LINE NUMBER IS 4.
000096 03
000097 COLUMN NUMBER IS 4
000098 PICTURE IS 9(6) INTO T-CODE
000099 CHECK WITH OVERFLOW LENGTH.
000100 03
000101 COLUMN NUMBER IS 20
000102 PICTURE IS X(20) FROM T-NAME.
000103 01 SCR-10.
000104 02 SCR-11
000105 LINE NUMBER IS D-LINE PLUS 7
000106 CLEAR DATA TO 24.
000107 03 SCR-12
000108 COLUMN NUMBER IS 2
000109 PICTURE IS 9 FROM D-LINE.
000110 02 SCR-13
000111 LINE NUMBER IS D-LINE PLUS 7.
000112 03

```

```

000113 COLUMN NUMBER IS 4
000114 PICTURE IS X(6) INTO S-CODE
000115 CHECK WITH OVERFLOW LENGTH.
000116 03
000117 COLUMN NUMBER IS 20
000118 PICTURE IS X(20) FROM S-NAME.
000119 03
000120 COLUMN NUMBER IS 45
000121 PICTURE IS 9(5) INTO SURYO
000122 CHECK WITH OVERFLOW.
000123 03
000124 COLUMN NUMBER IS 54
000125 PICTURE IS ZZ,ZZ9 FROM S-TANKA.
000126 03
000127 COLUMN NUMBER IS 67
000128 PICTURE IS ZZZ,ZZZ,ZZ9 FROM KINGAKU.
000129 01 SCR-90.
000130 02
000131 LINE NUMBER IS 18.
000132 03
000133 COLUMN NUMBER IS 55
000134 VALUE IS NC"合計".
000135 03
000136 COLUMN NUMBER IS 67
000137 PICTURE IS ZZZ,ZZZ,ZZ9 FROM GOKEI.
000138 02
000139 LINE NUMBER IS 22.
000140 03
000141 COLUMN NUMBER IS 55
000142 VALUE IS NC"確認".
000143 03
000144 PICTURE IS X INTO KAKUNIN.
000145 PROCEDURE DIVISION.
000146 PROC-0.
000147 OPEN INPUT TOKUI-M SYOHIN-M.
000148 DISPLAY SCR-01.
000149 PRC-2.
000150 ADD 1 TO DENPYO-NO.
000151 PRC-3.
000152 DISPLAY SCR-02.
000153 ACCEPT SCR-02.
000154 IF END-STS = "P1" GO TO PRC-9.
000155 ACCEPT SCR-04.
000156 READ TOKUI-M INVALID KEY STOP "ERROR 1"
000157 GO TO PRC-3.
000158 DISPLAY SCR-04.
000159 MOVE 1 TO D-LINE.
000160 MOVE ZERO TO GOKEI.
000161 PRC-4.
000162 IF D-LINE = 1 DISPLAY SCR-11
000163 ELSE DISPLAY SCR-12.
000164 ACCEPT SCR-13.
000165 IF END-STS = "04" GO TO PRC-8.
000166 READ SYOHIN-M INVALID KEY STOP "ERROR 2"
000167 GO TO PRC-4.
000168 MULTIPLY SURYO BY S-TANKA GIVING KINGAKU.

```

```
000169 ADD KINGAKU TO GOKEI.
000170 DISPLAY SCR-13.
000171 IF D-LINE = 5 GO TO PRC-8.
000172 ADD 1 TO D-LINE.
000173 GO TO PRC-4.
000174 PRC-8.
000175 DISPLAY SCR-90.
000176 ACCEPT SCR-90.
000177 GO TO PRC-2.
000178 PRC-9.
000179 CLOSE TOKUI-M SYOHIN-M.
000180 STOP RUN.
```



## 第23章 データベース機能

【Standard】本機能は Standard Edition 版および Enterprise Edition for Windows 版の SE 動作モードにおいて、使用できる。

### 23.1 概説

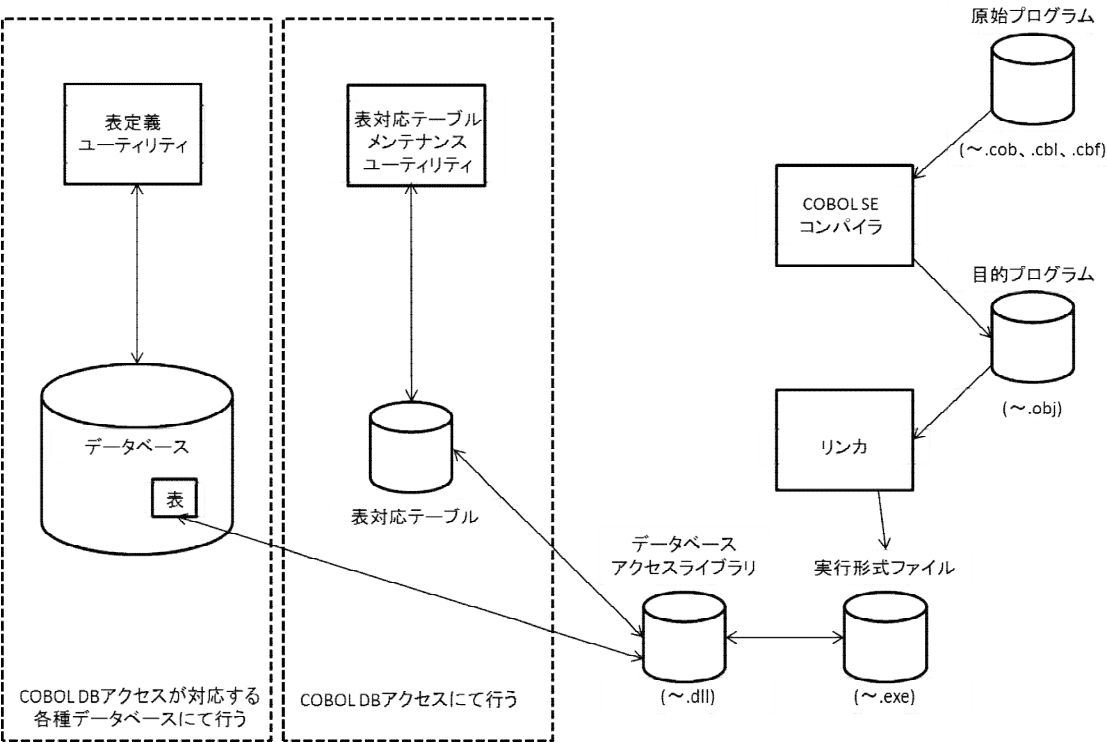
データベース機能は、本 OS の RDB(リレーショナル型データベース)上の関係表を従来からのファイル入出力と同じ概念で、アクセスする機能である。

関係表をアクセスするには、従来のファイルと同様に、ファイル定義やレコード記述項の定義を行い、入出力文を利用する。また、本 OS の RDB をより高度に利用するために、従来の入出力文を拡張した文が用意されている。

データベース機能を使用するためには、対応する別製品(COBOL DB アクセス, IFASPRO RDB)が必要である。

### 23.1.1 COBOL で使用するまでの手順

COBOL を使用して、ファイルに対するアクセスを行う場合の手順は、次のとおりである。



### 23.2 RDB ファイルの装置指定

RDB ファイルの装置指定は、`ASSIGN` 句に `RDB` を指定し、`VALUE OF` 句に表名を指定する。形式は次のとおりである。

**FILE—CONTROL.**

**SELECT** ファイル名 1 **ASSIGN TO** 内部ファイル名—RDB

**FD** ファイル名 1

**LABEL RECORD IS STANDARD**

**VALUE OF** { **IDENTIFICATION IS** { データ名 1  
定数 1 } } ...

- 内部ファイル名は、8 文字以内の英字から始まる英字、数字および下線で指定する。
- VALUE OF 句で、表定義ユーティリティによって定義した表名を指定する。
- ファイルマップ機能を用いた場合については「3.1 ファイルの装置およびファイル指定」の「2. ファイルマップ機能」を参照。

## 23.3 DB アクセス文

### 23.3.1 入出力文と拡張入出力文

RDB のアクセスは、通常のファイル入出力と同じ入出力文を利用する。さらに、RDB をより高度に利用するための拡張入出力文があるため、表 23-1 を参照。

表 23-1 従来の入出力文と拡張入出力文

| 言語記述          | 基本形 (従来の入出力文)                                                                                                                                                                                                                                                                                   | 拡張形 (DB アクセスだけの入出力文)                                                                       | 説 明                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| ファイル管理<br>記述項 | SELECT ファイル名<br>ASSIGN TO<br>内部ファイル名-MSD                                                                                                                                                                                                                                                        | SELECT ファイル名<br>ASSIGN TO<br>内部ファイル名-RDB                                                   | DB アクセスのときは、装置指定を ‘RDB’ とする。                                |
|               | RECORD KEY IS データ名                                                                                                                                                                                                                                                                              | RECORD KEY IS データ名 …<br>[WITH DUPLICATES]                                                  | RDB 索引ファイルでは、主レコードキーにも二重キーが許される。                            |
|               | ALTERNATE RECORD KEY IS<br>データ名<br>[WITH DUPLICATES]                                                                                                                                                                                                                                            | ALTERNATE RECORD KEY IS<br>データ名 …<br>[WITH DUPLICATES]                                     |                                                             |
| CLOSE 文       | CLOSE ファイル名<br>…                                                                                                                                                                                                                                                                                | 同左                                                                                         |                                                             |
| COMMIT 文      | なし                                                                                                                                                                                                                                                                                              | COMMIT                                                                                     |                                                             |
| DELETE 文      | DELETE ファイル名 RECORD<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2]                                                                                                                                                                                                                   | DELETE [CURRENT]<br>ファイル名 RECORD<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2] | CURRENT 句の指定により、乱および動的呼び出し法においても、現在のレコード指示子で示されるレコードを削除できる。 |
| OPEN 文        | OPEN<br><br><div style="display: inline-block; vertical-align: middle;"> <math>\left\{ \begin{array}{l} \text{INPUT \{ファイル名 1\}} \\ \dots \\ \text{OUTPUT \{ファイル名 2\}} \\ \dots \\ \text{I-O \{ファイル名 3\}} \\ \dots \\ \text{EXTEND \{ファイル名 3\}} \\ \dots \end{array} \right.</math> </div><br>… | 同左                                                                                         |                                                             |

|            |                                                                                                                                      |                                                                                                                                        |                                                                                                                                           |
|------------|--------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| READ 文     | 書き方 1<br>READ ファイル名 [NEXT]<br>RECORD [INTO 一意名]<br>[WITH UNLOCK]<br>[AT END 無条件文 1]<br>[NOT AT END 無条件文 2]                           | 書き方 1<br>READ ファイル名<br>{ NEXT<br>PRIOR } RECORD<br>[INTO 一意名]<br>[WITH UNLOCK]<br>[AT END 無条件文 1]<br>[NOT AT END<br>無条件文 2]            | PRIOR 句の指定により、<br>レコードを逆順に読み<br>込むことができる。                                                                                                 |
|            | 書き方 2<br>READ ファイル名 RECORD<br>[INTO 一意名]<br>[KEY IS データ名]<br>[WITH UNLOCK]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2] | 書き方 2<br>READ ファイル名 RECORD<br>[INTO 一意名]<br>[KEY IS データ名 …]<br>[WITH UNLOCK]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2] |                                                                                                                                           |
| REWRITE 文  | REWRITE レコード名<br>[FROM 一意名]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2]                                                | REWRITE [CURRENT]<br>レコード名 [FROM 一意名]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2]                                        | CURRENT 句の指定によ<br>り、乱および動的呼び<br>出し法においても、現<br>在のレコード指示子で<br>示されるレコードを書き<br>換えることができる。し<br>たがって、このときにレ<br>コードキーの値を変更<br>して書き換えることがで<br>きる。 |
| ROLLBACK 文 | なし                                                                                                                                   | ROLLBACK                                                                                                                               |                                                                                                                                           |
| SCRATCH 文  | なし                                                                                                                                   | SCRATCH ファイル名                                                                                                                          | SELECT 文で選択した<br>部分集合を解放する。                                                                                                               |
| SELECT 文   | なし                                                                                                                                   | SELECT ファイル名<br>[WHERE レコード選択式]<br>[COUNT IN 一意名]<br><br>[ORDER { BY { ASCENDING<br>DESCENDING } }<br><br>KEY データ名 1 … } … ]           | レコード選択式で示さ<br>れる条件を満たすレコ<br>ードを RDB から抽出し、<br>部分集合を作る。                                                                                    |

|         |                                                                                                                             |                                                                                                                                      |                                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| START 文 | 書き方 1<br>START ファイル名<br>[KEY IS ~ データ名]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2]                           | 書き方 1<br>START ファイル名<br>[KEY IS ~ データ名 ...]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2]                                |                                                                                                                   |
| START 文 | 書き方 2<br><br>なし                                                                                                             | 書き方 2<br>START ファイル名<br>{ FIRST } RECORD<br>{ LAST }<br>[KEY IS データ名 ... ]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2] | 書き方 2により, 表内の最初 (FIRST), または最後 (LAST) のレコードに位置づけることができる。また, 書き方 2 の START 文は RDB 順ファイルに対しても使用することができる。KEY 句は書けない。 |
| USE 文   | USE AFTER STANDARD<br>{ EXCEPTION }<br>{ ERROR } PROCEDURE ON<br><br>{<br>ファイル名...<br>INPUT<br>OUTPUT<br>I-O<br>EXTEND<br>} | 同左                                                                                                                                   |                                                                                                                   |
| WRITE 文 | WRITE レコード名<br>[FROM 一意名]<br>[INVALID KEY<br>無条件文 1]<br>[NOT INVALID KEY<br>無条件文 2]                                         | 同左                                                                                                                                   | RDB 順ファイルに対して INVALID KEY 句が書ける (必ず書かねばならない)。                                                                     |

## 23.4 RDB とファイル編成

本 COBOL では、RDB を通常のファイルの入出力と同様にアクセスできるよう、言語記述上で 3 つのファイル編成 (RDB 順ファイル、RDB 相対ファイル、RDB 索引ファイル) として扱う。どのファイル編成を用いて RDB をアクセスするかは、アプリケーションの都合により決定し、ファイル管理記述項で指定する。

### FILE—CONTROL.

SELECT ファイル名 1 ASSIGN TO 内部ファイル名—RDB

ORGANIZATION IS       $\left\{ \begin{array}{ll} \underline{\text{SEQUENTIAL}} & \text{(順ファイル)} \\ \underline{\text{RELATIVE}} & \text{(相対ファイル)} \\ \underline{\text{INDEXED}} & \text{(索引ファイル)} \end{array} \right\}$

- RDB 順ファイル  
RDB 上のデータを、物理格納順に順次アクセスするときに利用する。
- RDB 相対ファイル  
RDB 上のデータの物理格納位置を、RELATIVE KEY データ項目で指定してアクセスする。
- RDB 索引ファイル  
RDB 上のデータを、レコードキーを指定してアクセスする。レコードキー項目は、複数定義できるため、もっとも多様なアクセスを行うことが可能である。

## 23.5 入出力文

RDB ファイルに対する入出力文は、ファイル入出力機能の入出力文と同様に行う。ここでは、DB アクセスのために拡張された入出力文について説明する。

### (1) 順呼び出し

最初(最後)から順に読み込んでいく。また、読み込んだレコードの次の(前の)レコードを読み込む。

### (2) 乱呼び出し

ファイルの好きな場所から読み込む。その際は必ずどのレコードを読み込むか指定しなければ

ばならない。指定方法はファイルの種類によって異なる。

### (3) 動的呼び出し

乱呼び出しをしてから順呼び出しをすることである。ファイルのある場所を読み込んでから、降順または昇順に読み込んでいく。

## 23.5.1 READ 文

RDB 順ファイル, RDB 相対ファイル, RDB 索引ファイル内の READ 文について説明する。

### (1) RDB 順ファイル

順ファイルの READ 文は, 入力または入出力両用モードで OPEN されたファイルに使用し, その動作は, データを RDB 上の物理順, または逆物理順に読む。

例

|                                   |                 |
|-----------------------------------|-----------------|
| READ D-FILE NEXT AT END GO TO ～.  | n 番目のレコードを読む。   |
| ↓                                 |                 |
| READ D-FILE NEXT AT END GO TO ～.  | n+1 番目のレコードを読む。 |
| ↓                                 |                 |
| READ D-FILE PRIOR AT END GO TO ～. | n 番目のレコードを読む。   |
| ↓                                 |                 |
| READ D-FILE PRIOR AT END GO TO ～. | n-1 番目のレコードを読む。 |

READ 文実行中にデータの終端または始端が見つかると, AT END 条件になる。

### (2) RDB 相対ファイル

相対ファイルの READ 文は, 入力または入出力両用モードで OPEN されたファイルに使用し, その動作は, 呼び出し法によって異なる。

順呼び出しや動的呼び出しの順次処理では, RELATIVE KEY データ項目の値に関係なく呼び出す。乱呼び出しや動的呼び出しで任意の順に処理するときは, RELATIVE KEY データ項目を指定する。

例

#### a. 順呼び出し

|                                         |                 |
|-----------------------------------------|-----------------|
| READ D-FILE AT END GO TO END-RTN.       | n 番目のレコードを読む。   |
| READ D-FILE AT END GO TO END-RTN.       | n+1 番目のレコードを読む。 |
| READ D-FILE PRIOR AT END GO TO END-RTN. | n 番目のレコードを読む。   |

#### b. 乱呼び出し



```

|
MOVE 35 TO REL-KEY.
READ D-FILE INVALID KEY GO TO ERR-RTN. 35 番目のレコードを読む。
MOVE 10 TO REL-KEY.
READ D-FILE INVALID KEY GO TO ERR-RTN. 10 番目のレコードを読む
|

```

### c. 動的呼び出し

```

|
MOVE 30 TO REL-KEY.
READ D-FILE INVALID KEY GO TO ERR-RTN. 30 番目のレコードを読む。
READ D-FILE PRIOR AT END GO TO END-RTN. 29 番目のレコードを読む。
MOVE 25 TO REL-KEY.
READ D-FILE INVALID KEY GO TO ERR-RTN. 25 番目のレコードを読む。
READ D-FILE NEXT AT END GO TO END-RTN. 26 番目のレコードを読む。
|

```

## (3) RDB 索引ファイル

索引ファイルの READ 文は、入力または入出力両用モードで OPEN されたファイルに使用し、その動作は、呼び出し法によって異なる。

順呼び出しや動的呼び出しの順次処理では、レコードキーの順に読み込まれ、乱呼び出しや動的呼び出しで任意の順に処理するときは、レコードキーを指定する。

### 例

#### a. 順呼び出し

```

START IS-FILE FIRST KEY IS REC-KEY-1 REC-KEY-1 で最小のキー値を
INVALID KEY GO TO ERR-RTN. 持つレコードに位置づける。
READ IS-FILE AT END GO TO END-RTN. 位置づけされたレコードを読む。
READ IS-FILE AT END GO TO END-RTN. REC-KEY-1 の次のキー値を持つ
 レコードを読む。

```

#### b. 乱呼び出し

```

MOVE A TO REC-KEY-1
READ IS-FILE KEY IS REC-KEY-1 REC-KEY-1 のキー値が A のレコードを読む。
INVALID KEY GO TO ERR-RTN.
MOVE B TO REC-KEY-2.
READ IS-FILE KEY IS REC-KEY-2 REC-KEY-2 のキー値が B のレコードを読む。
INVALID KEY GO TO ERR-RTN.

```

c. 動的呼び出し

MOVE A TO REC-KEY-1.

READ IS-FILE KEY IS REC-KEY-1      REC-KEY-1 のキー値が A のレコードを読む。

INVALID KEY GO TO ERR-RTN.

READ IS-FILE NEXT AT END GO TO END RTN. REC-KEY-1 の次のキー値を持つレコードを読む。

## 23.5.2 WRITE 文

RDB 順ファイル, RDB 相対ファイル, RDB 索引ファイル内の WRITE 文について説明する。

### (1) RDB 順ファイル

WRITE 文は, 出力および拡張モードで OPEN されたファイルに使用する。

RDB 順ファイルに対して WRITE 文実行の結果, 二重キーが発生する可能性があるときは INVALID KEY 句を書く。

RDB ファイルでは, 順ファイルアクセスでも索引キーに対する索引の保守が行われる。このため, 表定義されたプライマリキーやセカンダリキーが二重キーを許さなければ, 二重キーエラーが発生する。

### (2) RDB 相対ファイル

WRITE 文は, 出力または入出力両用モードで OPEN されたファイルに使用し, その動作は, 呼び出し法や OPEN 文のモードによって異なる。

出力モードの順呼び出しでは, レコードを順次作成し, また, 出力モードの乱呼び出しや動的呼び出しでは, ランダムにレコードを作成する。入出力モードの乱呼び出しでは, レコードの挿入を行う。

なお, RELATIVE KEY データ項目は, 順呼び出し以外では必ず指定しなければならない。

### (3) RDB 索引ファイル

WRITE 文は, 出力または入出力両用モードで OPEN されたファイルに使用し, その動作は, OPEN 文のモードによって異なる。

出力モードでは, 新しくデータを作成し, 入出力両用モードではデータの挿入を行う。

なお, レコードキーは, 必ず指定しなければならない。

### 23.5.3 REWRITE 文

RDB 順ファイル, RDB 相対ファイル, RDB 索引ファイル内の REWRITE 文について説明する。

#### (1) RDB 順ファイル

REWRITE 文は, 入出力両用モードで OPEN されたファイルに使用する。

この REWRITE 文は, 直前の READ 文で読み込まれたレコードを, 新しいレコードに置き換える。

REWRITE 文を実行する前には, READ 文が実行されていなければならない。

例

```

READ n 番目のレコードを読む。
↓
READ n+1 番目のレコードを読む。
↓
REWRITE n+1 番目のレコードを置き換える。
↓
REWRITE エラー(命令実行順序エラー)

```

また, WRITE 文と同じように, 二重キーが発生する可能性があるときは, INVALID KEY 句を書く。

#### (2) RDB 相対ファイル

REWRITE 文は, 入出力両用モードで OPEN されたファイルに使用し, その動作は, 呼び出し法によって異なる。

順呼び出しでは, 直前に READ 文で読み込まれたレコードを置き換え, 乱呼び出しや動的呼び出しでは, CURRENT 句がなければ RELATIVE KEY データ項目で指定したレコードを置き換える。また CURRENT 句があれば, 直前の READ 文で読み込んだレコードを置き換えることができる。

#### (3) RDB 索引ファイル

REWRITE 文は, 入出力両用モードで OPEN されたファイルに使用し, その動作は, 呼び出し法によって異なる。

順呼び出しでは, 直前に READ 文で読み込まれたレコードを置き換え, 乱呼び出しや動的呼び出しでは, CURRENT 句がなければレコードキーで指定したレコードを置き換える。また, CURRENT 句があれば, 直前の READ 文で読み込んだレコードを置き換えることができる。このときにレコードキーの値を変更することができる。

例

a. 順呼び出し

|

```

|
READ IS-FILE AT END GO TO END-RTN.
REWRITE OS-REC INVALID KEY GO TO ERR-RTN.
|

```

b.乱呼び出し/動的呼び出し

```

MOVE "A" TO REC-KEY-1.
READ IS-FILE INVALID KEY GO TO ERR-RTN.
MOVE "B" TO REC-KEY-1.
REWRITE CURRENT IS-REC
 INVALID KEY GO TO ERR-RTN.

```

レコードキー"A"をもつレコードは存在なくなり、レコードキー"B"をもつレコードが書かれる。

## 23.5.4 START 文

RDB 順ファイル, RDB 相対ファイル, RDB 索引ファイル内の **START** 文について説明する。  
 下記に, **START** 文の書き方について示す。

### 書き方 1

**START** ファイル名 1

|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| $\left( \begin{array}{c} \text{KEY} \end{array} \right.$ | $\left\{ \begin{array}{l} \text{IS } \underline{\text{EQUAL}} \text{ TO} \\ \text{IS } = \\ \text{IS } \underline{\text{GREATER}} \text{ THAN} \\ \text{IS } > \\ \text{IS } \underline{\text{NOT}} \text{ } \underline{\text{LESS}} \text{ THAN} \\ \text{IS } \underline{\text{NOT}} < \\ \text{IS } \underline{\text{GREATER}} \text{ THAN } \underline{\text{OR}} \text{ } \underline{\text{EQUAL}} \text{ TO} \\ \text{IS } > = \end{array} \right.$ | $\left. \begin{array}{c} \text{データ名 1} \end{array} \right)$ |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |
|                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                             |

[**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  $\left[ \begin{array}{c} \underline{\text{FIRST}} \\ \underline{\text{LAST}} \end{array} \right]$  RECORD  
**[KEY IS** データ 1 ... **]**  
**[INVALID KEY** 無条件文 1]  
**[NOT INVALID KEY** 無条件文 2]  
**[END—START].**

## (1) RDB 順ファイル

START 文は、入力または入出力両用モードで OPEN されたファイルに使用し、その動作は、ファイル中の最初か最後のレコードに位置づけられる。ただし、順ファイルでは、書き方 2 の START 文だけが使用でき、KEY 句を書くことはできない。

例

START S-FILE FIRST INVALID KEY～ ——1 番目のレコードに位置づける。  
 READ S-FILE NEXT AT END～ ————1 番目のレコードを読み込む。  
 READ S-FILE NEXT AT END～ ————2 番目のレコードを読み込む。  
 START S-FILE LAST INVALID KEY ——最後のレコードに位置づける。  
 READ S-FILE PRIOR AT END～ ————最後のレコードを読み込む。

## (2) RDB 相対ファイル

START 文は、入力または入出力両用モードで OPEN されたファイルで、かつ順呼び出しや動的呼び出しの場合に使用する。

その動作は、書き方 1 では KEY 句で指定された条件を満たすレコードに位置づけ、書き方 2 ではファイル中の最小の相対レコード番号をもつレコード (FIRST 指定)、または最大の相対レコード番号をもつレコード (LAST 指定) に位置づけられる。

## (3) RDB 索引ファイル

START 文は、入力または入出力両用モードで OPEN されたファイルで、かつ順呼び出しや動的呼び出しの場合に使用できる。その動作は、書き方 1 では KEY 句で指定した条件を満たすレコードに位置づけ、書き方 2 では KEY 句で指定したレコードキーのなかで最小の値をもつレコード (FIRST 指定)、または最大の値をもつレコード (LAST 指定) に位置づけられる。

## 23.5.5 DELETE 文

RDB 順ファイル、RDB 相対ファイル、RDB 索引ファイルの DELETE 文について説明する。

## (1) RDB 順ファイル

DELETE 文は、入出力両用モードで OPEN されたファイルに使用し、直前の READ 文で読み込まれたレコードを削除する。DELETE 文を実行する前には、必ず READ 文が実行されていなければならない。

例

```

READ n 番目のレコードを読む。
↓
READ n+1 番目のレコードを読む。
↓
DELETE n+1 番目のレコードを削除する。
↓
DELETE エラー(命令実行順序エラー)

```

## (2) RDB 相対ファイル

DELETE 文は、入出力両用モードで OPEN されたファイルに使用し、その動作は、呼び出し法によって異なる。

順呼び出しでは直前に READ 文で読み込まれたレコードを削除し、乱呼び出しや動的呼び出しでは、CURRENT 句がなければ RELATIVE KEY データ項目で指定したレコードを削除する。また、CURRENT 句があれば、直前の READ 文で読み込んだレコードを削除することができる。

## (3) RDB 索引ファイル

DELETE 文は、入出力両用モードで OPEN されたファイルに使用し、その動作は、呼び出し法によって異なる。

順呼び出し法では直前に READ 文で読み込まれたレコードを削除し、乱呼び出しや動的呼び出しでは、CURRENT 句がなければレコードキーで指定したレコードを削除する。また、CURRENT 句があれば、直前の READ 文で読み込んだレコードを削除することができる。

## 23.5.6 SELECT 文と SCRATCH 文

SELECT 文と SCRATCH 文は、入力および入出力両用モードで OPEN されたファイルで、かつ呼び出し方法が順呼び出しや動的呼び出しの場合に使用する。

SELECT 文の動作は、レコード選択式の条件を満たすレコードを、RDB ファイル上から選択抽出し、その部分集合を作成する。

また、COUNT IN 句で指定されたレコード件数を通知したり、ORDER BY 句で指定したデータ名をキーにして、昇順または降順に並べ換えることができる。

なお、SCRATCH 文は、選択した部分集合を解放し、SCRATCH 文を実行すると、レコード指示子の指す位置は不定となる。

例

```

SELECT MST-F ASSIGN TO MASTER-RDB
 ORGANIZATION INDEXED
 ACCESS MODE SEQUENTIAL
 RECORD KEY A WITH DUPLICATES
 ALTERNATE RECORD KEY X WITH DUPLICATES
 FILE STATUS F-STS.

```

```

 :

```

```

FD MST-F
 LABEL RECORD STANDARD
 VALUE OF IDENTIFICATION "SHOMST".

```

```

01 MST-R
 02 A PIC X(20).
 02 B PIC N(30).
 02 C PIC 9(10).
 02 X PIC X(2).
 02 T PIC 9(3).

```

```

 :

```

```

77 R-CNT PIC 9(5).

```

#### 手続き部

```

OPEN INPUT MST-F.
SELECT MST-F WHERE T IS NOT < 100
 COUNT IN R-CNT
 ORDER BY ASCENDING KEY X. — a

```

```

P-1.

```

```

 READ MST-F AT END GO TO P-2. ——— b

```

```

 :

```

```

 GO TO P-1.

```

```

P-2.

```

```

 SCRATCH MST-F. ————— c

```

```

 :

```

- a. SELECT 文の実行により`SHOMST'の部分集合が選択され、ORDER BY ASCENDING KEY 句の指定で、項目X の昇順に並べ変わる。また、COUNT IN 句のR-CNT の値は、3 となる。
- b. READ 文で選択された部分集合内のレコードを、順次読み込む。選択された部分集合が0 件のとき、最初のREAD 文でAT END 条件となる。
- c. 選択した部分集合を解放し、レコード指示子は不定となる。

## 23.5.7 COMMIT 文と ROLLBACK 文

COMMIT(静止点の確立)文と ROLLBACK(静止点への復帰)文は、OPEN 文の指定やファイルの呼び出し方法に関わらず使用できる。

COMMIT 文の動作は、データベースの更新要求を有効にし、レコードの排他制御を行っている場合はすべてのロックを解放するとともに、静止点を確立する。

ROLLBACK 文の動作は、直前の静止ポイントから現在まで行われたデータベースの更新を無効にする。

例

```

1 COMMIT データベースの更新要求を有効にする
 ↓
2 WRITE データベースを更新する
 ↓
3 ROLLBACK 2のデータベースの更新を無効とし、データベースを1の状態にする

```

複数の RDB エンジンと同時に使用している場合は、以下の順で COMMIT/ROLLBACK 文を発行する。

(1) COBOL DB アクセス

(2) IFASPRO RDB

これは、COMMIT/ROLLBACK 文の異常終了時の RDB-STATUS の返却に影響する。

## 23.6 RDB-STATUS

### (1) 概要

通常 COMMIT/ROLLBACK 文の異常終了時は、実行時エラーメッセージを出力しアプリケーションが異常終了する。

COMMIT/ROLLBACK 文において通常ファイルのようにステータスを取得し、アプリケーション内での制御を可能にする方法が RDB-STATUS である。

RDB-STATUS は 2 桁の英数字基本項目で構成され、COMMIT 文、ROLLBACK 文の実行結果のみが通知される。

### (2) 取得方法

RDB-STATUS は、アプリケーションカスタマイザの指定または環境変数 COB\_RDB\_STATUS の設定により取得が可能となる。

アプリケーションカスタマイザの指定または環境変数 COB\_RDB\_STATUS の設定で、RDB-STATUS へ通知する設定が行われているとき、プログラム上の RDB-STATUS に実行結果



が通知される。

### (3) RDB-STATUS の返却

RDB-STATUS へ通知する設定が行われていないとき、COMMIT/ROLLBACK 文 が異常終了した場合は、プログラム上の RDB-STATUS に実行結果は通知されず、実行時エラーメッセージ出力後、異常終了する。

RDB-STATUS へ通知する設定が行われているとき、COMMIT/ROLLBACK の異常終了した場合は、実行時エラーメッセージ出力後、プログラム上の RDB-STATUS に実行結果を通知する。

COMMIT/ROLLBACK の異常終了後は、アプリケーションを終了させるまで RDB に関するすべての文は 99 エラーで返却される。

### (4) 複数 RDB エンジン使用時

アプリケーションカスタマイザの「ファイル」タブ「RDB-STATUS への通知動作を指定する」に「通知する」の指定があり、複数の RDB エンジンを1アプリケーション上で同時に使用しているときに、COMMIT/ROLLBACK 文が異常終了した場合は、最初に発生した RDB-STATUS の内容を返却する。

## 23.7 入出力状態

ファイル管理記述項中に FILE STATUS(ファイル状態)句を書くと各入出力文の入出力操作の状態を示す値が、FILE STATUS 句で指定した 2 桁のデータ項目中に設定される。

また、アプリケーションカスタマイザの「RDB-STATUS への通知動作を指定する」にて「通知する」を指定することにより、COMMIT 文または ROLLBACK 文の入出力状態を示す値が、特殊レジスタ RDB-STATUS に設定される。ただし、COMMIT 文または ROLLBACK 文の実行結果により USE 手続きが実行されることはない。

ファイル管理記述項中に FILE STATUS(ファイル状態)句を書いてあっても、プログラムに入出力状態を示す値を通知しても復旧不可な状態が発生した場合や、ファイル管理記述項中に FILE STATUS(ファイル状態)句を書かなかった場合には、実行時メッセージが表示される。その場合に、メッセージ中に使用している RDB エンジンが検出したエラーの詳細ステータスコードが設定される。その詳細ステータスコードの意味は、使用している RDB エンジン側のマニュアルを参照。OS 互換モードの違いにより入出力状態値および動作の差異がある。OS 互換モードが A-VXII のとき、または A-VX5 のときの詳細仕様については、IFASPRO RDB および COBOL DB アクセスのマニュアルを参照。

RDB 順ファイルの入出力状態については表 23-2、RDB 相対ファイルの入出力状態については表 23-3、RDB 索引ファイルの入出力状態については表 23-4 を参照。また、使用している RDB エンジンに付属する説明も合わせて参照。

表 23-2 RDB 順ファイルの入出力状態

| 入出力<br>状態 | 内容                                                                                                                                                                                          | 文               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 00        | 入出力文の実行は正常終了した。                                                                                                                                                                             | すべて             |
| 05        | OPEN 文の実行は成功した。しかし、OPEN 文を実行したとき、参照した不定ファイル(optional file)は存在しない。                                                                                                                           | OPEN            |
| 10        | READ 文が行われたが、次のレコードが存在しない。次の場合のいずれかである。<br><br>1. ファイルの終わりに達した。<br>2. 部分集合の終わりに達した。<br>3. READ 文が、存在しない不定入力ファイルに対して初めて行われた。                                                                 | READ            |
| 22        | 重複キーがあった。重複キーを引き起こすような REWRITE 文または WRITE 文を実行しようとした。                                                                                                                                       | REWRITE, WRITE  |
| 23        | レコード未発見を示す。指定したキーを持つレコードがファイル中に見つからなかった。                                                                                                                                                    | SELECT, START   |
| 30        | 永続誤りが存在する。                                                                                                                                                                                  | すべて             |
| 34        | 区域外書出しによる永続誤りが存在する。COBOL の外部で定義されたファイルの区域外にレコードを書こうとした。                                                                                                                                     | REWRITE, WRITE  |
| 35        | 不定ファイルでないファイル(non-optional file) が存在しない(表が定義されていない)とき、OPEN を実行しようとしたことによる永続誤りが存在する。                                                                                                         | OPEN            |
| 37        | オープンモード指定誤りが存在する。仮想表に対して OUTPUT/EXTEND オープンを実行しようとしたことによる永続誤りが存在する。                                                                                                                         | OPEN            |
| 39        | 固定ファイル属性とプログラム中に指定した属性の間で矛盾が検出されたため、OPEN 文が不成功となった。                                                                                                                                         | OPEN            |
| 41        | 開かれている状態のファイルに OPEN 文を実行しようとした。                                                                                                                                                             | OPEN            |
| 42        | 開かれていない状態のファイルに CLOSE 文を実行しようとした。                                                                                                                                                           | CLOSE           |
| 43        | 大容量記憶ファイルで、DELETE 文または REWRITE 文を実行する前に、関連するファイルに対して実行された最後の入出力文が、成功した READ 文でない。または WITH UNLOCK 句を指定した READ 文である。                                                                          | DELETE, REWRITE |
| 46        | 入力モードまたは入出力両用モードで開かれたファイルに READ 文を実行しようとしたが、有効な次のレコードがない。次の場合のいずれかである。<br><br>1. 先行する READ 文が、ファイル終了条件以外より不成功であった。<br>2. 先行する READ 文が、ファイル終了条件を引き起こした。<br>3. 先行するその他の文により、ファイル位置指示子が不定となった。 | READ            |

|    |                                                                                                                                                                          |                                     |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| 47 | 入力モードまたは入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルまたは未オープンファイルに、READ 文、SELECT 文または START 文を実行しようとした。                                                                           | READ, SELECT, START                 |
| 48 | 出力モードまたは拡張モードで開かれていない(開かれた状態でない場合も含む)ファイルに、WRITE 文を実行しようとした。または結合型仮想表に対して、WRITE 文を実行しようとした。                                                                              | WRITE                               |
| 49 | 入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、DELETE 文または REWRITE 文を実行しようとした。または結合型仮想表に対して、DELETE 文または REWRITE 文を実行しようとした。                                                       | REWRITE, DELETE                     |
| 91 | ファイルが他から排他的に使用されている(IFASPRO RDB 使用時)。                                                                                                                                    | OPEN                                |
| 92 | (IFASPRO RDB 使用時)<br>1.READ または START 文において、指定キーが存在しない、またはキー情報が不正である。<br>2.OUTPUT/EXTEND モードで OPEN されたファイルに対して SELECT 文を実行しようとした。<br>3.SELECT 文を行わずに SCRATCH 文を実行しようとした。 | READ, START, SCRATCH, SELECT        |
| 99 | (IFASPRO RDB 使用時)<br>1. IFASPRO RDB でエラーが発生している。詳細は IFASPRO RDB のマニュアルを参照。                                                                                               | すべて                                 |
| 9F | フェールオーバーが発生した(IFASPRO RDB 使用時)。                                                                                                                                          | すべて                                 |
| 9J | デッドロック状態となった。ROLLBACK 文が自動的に行われた。                                                                                                                                        | DELETE, READ, REWRITE, START, WRITE |
| 9K | 1.ロールバックログ収集エラーが発生した(IFASPRO RDB 使用時)。<br>2.COMMIT/ROLLBACK が異常終了した(COBOL DB アクセス使用時)。                                                                                   | すべて                                 |
| 9M | システム表のアクセスエラー(DB 破壊)がある(IFASPRO RDB 使用時)。                                                                                                                                | すべて                                 |
| 9N | ログ領域がオーバーフローした(COBOL DB アクセス使用時)。                                                                                                                                        | DELETE, REWRITE, WRITE              |
| 9O | ログ領域がオーバーフローした(IFASPRO RDB 使用時)。                                                                                                                                         | DELETE, REWRITE, WRITE              |
| 9P | ページまたはレコードのロックテーブルがオーバーフローした。                                                                                                                                            | DELETE, READ, REWRITE, WRITE        |
| 9T | ログファイルの I/O エラーがある(IFASPRO RDB 使用時)。                                                                                                                                     | すべて                                 |

表 23-3 RDB 相対ファイルの入出力状態

| 入出力<br>状態 | 内容                                                                                                                                                             | 文               |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 00        | 入出力文の実行は正常終了した。                                                                                                                                                | すべて             |
| 05        | OPEN 文の実行は成功した。しかし、OPEN 文を実行したとき、参照した不定ファイル(optional file)は存在しない。                                                                                              | OPEN            |
| 10        | 順呼び出しの READ 文が行われたがファイルの終わりに達したが、次の論理レコードがファイル中に存在しない。次の場合のいずれかである。<br><br>1. ファイルの終わりに達した。<br>2. 部分集合の終わりに達した。<br>3. 順呼び出しの READ 文が、存在しない不定入力ファイルに対して初めて行われた。 | READ            |
| 14        | 順呼び出しの READ 文が行われたが、相対レコード番号の有効桁数とそのファイルに指定された相対キー項目の大きさより大きい。                                                                                                 | READ            |
| 22        | 重複キーがあった。すでにレコードがある位置に重ねて WRITE 文でレコードを書こうとした。                                                                                                                 | REWRITE, WRITE  |
| 23        | 次の場合のいずれかである。<br>1. ファイル中に存在しないレコードを乱呼び出ししようとした。<br>2. START 文または乱呼び出しの READ 文を、存在しない不定入力ファイルに実行しようとした。                                                        | READ, START     |
| 24        | 次の場合のいずれかである。<br>1. COBOL の外部で定義された相対ファイルの区域外にレコードを書こうとした。<br>2. 順呼び出しの WRITE 文が行われたが、相対レコード番号の有効桁数とそのファイルに指定された相対キー項目の大きさより大きい。                               | REWRITE, WRITE  |
| 30        | 永続誤りが存在する。その入出力状態に関しては、これ以上情報がない。                                                                                                                              | すべて             |
| 35        | 不定ファイルでないファイル(non optional file)が存在しない(表が定義されていない)とき、OPEN 文を実行しようとしたことによる永続誤りが存在する。                                                                            | OPEN            |
| 37        | オープンモード指定誤りが存在する。仮想表に対して OUTPUT/EXTEND オープンを実行しようとしたことによる永続誤りが存在する。                                                                                            | OPEN            |
| 39        | 固定ファイル属性とプログラム中に指定した属性の間で矛盾が検出されたため、OPEN 文が不成功となった。                                                                                                            | OPEN            |
| 41        | 開かれている状態のファイルに OPEN 文を実行しようとした。                                                                                                                                | OPEN            |
| 42        | 開かれていない状態のファイルに CLOSE 文を実行しようとした。                                                                                                                              | CLOSE           |
| 43        | 順呼び出しで、CURRENT 句付きの DELETE 文または CURRENT 句                                                                                                                      | DELETE, REWRITE |

|    |                                                                                                                                                                                                                                       |                                     |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
|    | 付きの REWRITE 文を実行する前に、そのファイルに対して実行された最後の入出力文が、成功した READ 文でない。または WITH UNLOCK 句を指定した READ 文である。                                                                                                                                         |                                     |
| 46 | 入力モードまたは入出力両用モードで開かれたファイルに順呼び出しの READ 文(NEXT/PRIOR 指定)を実行しようとしたが、有効な次のレコードがない。次の場合のいずれかである。<br>1. 先行する START 文が、不成功であった。<br>2. 先行する READ 文が、ファイル終了条件を引き起こした。<br>3. 先行する READ 文が、ファイル終了条件を引き起こした。<br>4. 先行するその他の文により、ファイル位置指示子が不定となった。 | READ                                |
| 47 | 入力モードまたは入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、READ 文、SELECT 文または START 文を実行しようとした。                                                                                                                                                    | READ, SELECT, START                 |
| 48 | 入出力両用モード、出力モードまたは拡張モードで開かれていない(開かれた状態でない場合も含む)ファイルに、WRITE 文を実行しようとした。または結合型仮想表に対して、WRITE 文を実行しようとした。                                                                                                                                  | WRITE                               |
| 49 | 入出力両用モード、出力モードまたは拡張モードで開かれていない(開かれた状態でない場合も含む)ファイルに、WRITE 文を実行しようとした。または結合型仮想表に対して、WRITE 文を実行しようとした。                                                                                                                                  | WRITE                               |
| 91 | ファイルが他から排他的に使用されている(IFASPRO RDB 使用時)。                                                                                                                                                                                                 | OPEN                                |
| 92 | (IFASPRO RDB 使用時)<br>1. READ または START 文において、指定キーが存在しない、またはキー情報が不正である。<br>2. OUTPUT/EXTEND モードで OPEN されたファイルに対して SELECT 文を実行しようとした。<br>3. SELECT 文を行わずに SCRATCH 文を実行しようとした。                                                           | READ, START, SCRATCH, SELECT        |
| 99 | (IFASPRO RDB 使用時)<br>1. IFASPRO RDB でエラーが発生している。詳細は IFASPRO RDB のマニュアルを参照。                                                                                                                                                            | すべて                                 |
| 9F | フェールオーバーが発生した(IFASPRO RDB 使用時)。                                                                                                                                                                                                       | すべて                                 |
| 9J | デッドロック状態となった。ROLLBACK 文が自動的に行われた。                                                                                                                                                                                                     | DELETE, READ, REWRITE, START, WRITE |
| 9K | 1. ロールバックログ収集エラーが発生した(IFASPRO RDB 使用時)。<br>2. COMMIT/ROLLBACK が異常終了した(COBOL DB アクセス使用時)。                                                                                                                                              | すべて                                 |
| 9M | システム表のアクセスエラー(DB 破壊)がある(IFASPRO RDB 使用時)。                                                                                                                                                                                             | すべて                                 |
| 9N | ログ領域がオーバーフローした(COBOL DB アクセス使用時)。                                                                                                                                                                                                     | DELETE, REWRITE, WRITE              |

|    |                                       |                              |
|----|---------------------------------------|------------------------------|
| 90 | ログ領域がオーバーフローした (IFASPRO RDB 使用時)。     | DELETE, REWRITE, WRITE       |
| 9P | ページまたはレコードのロックテーブルがオーバーフローした。         | DELETE, READ, REWRITE, WRITE |
| 9T | ログファイルの I/O エラーがある (IFASPRO RDB 使用時)。 | すべて                          |

表 23-4 RDB 索引ファイルの入出力状態

| 入出力<br>状態 | 内容                                                                                                                                                                                                   | 文                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| 00        | 入出力文の実行は正常終了した。                                                                                                                                                                                      | すべて                  |
| 02        | 入出力文の実行は正常終了したが、重複キーが検出された。次のいずれかの場合である。<br>1. READ 文で入力したレコードの参照キーの値が、その索引における次のレコードの参照キーと等しい。<br>2. REWRITE 文または WRITE 文によって書かれたレコードの副レコードキーの値が、既にファイルに存在している。ただし、この副レコードキーは重複した値が許されているので、誤りではない。 | READ, REWRITE, WRITE |
| 05        | OPEN 文の実行は成功した。しかし、OPEN 文を実行したとき、参照した不定ファイル(optional file)は存在しない。                                                                                                                                    | OPEN                 |
| 10        | 順呼び出し READ 文が行われたが、次の論理レコードがファイル中に存在しない。<br>次のいずれかの場合である。<br>1. ファイルの終わりに達した。<br>2. 部分集合の終わりに達した。<br>3. 順呼び出しの READ 文が、存在しない不定入力ファイルに初めて行われた。                                                        | READ                 |
| 21        | 順呼び出しにおいて、順序誤りがあった。すなわち、成功した READ 文と次の REWRITE 文の間で主レコードキーの値が変更されたか、または、WRITE 文を実行時に主レコードキーの値が昇順になっていない。                                                                                             | WRITE                |
| 22        | REWRITE 文または WRITE 文で RDB 索引ファイルに書こうとしたレコードの主レコードキーまたは DUPLICATES 句が指定されていない副レコードキーの値が、すでにファイル中に存在している。                                                                                              | REWRITE, WRITE       |
| 23        | 次の場合のいずれかである。<br>1. ファイル中に存在しないレコードを乱呼び出ししようとした。<br>2. START 文または乱呼び出しの READ 文を存在しない不定入力ファイルに実行しようとした。                                                                                               | READ, START          |
| 24        | COBOL の外部で定義された RDB 索引ファイルの区域外にレコードを書こうとした。                                                                                                                                                          | WRITE, REWRITE       |
| 30        | 永続誤りが存在する。                                                                                                                                                                                           | すべて                  |

|    |                                                                                                                                                                                                                                 |                             |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| 35 | 不定ファイルでないファイル(non optional file)が存在しない(表が定義されていない)とき、OPEN 文を実行しようとしたことによる永続誤りが存在する。                                                                                                                                             | OPEN                        |
| 37 | オープンモード指定誤りが存在する。仮想表に対して OUTPUT/EXTEND オープンを実行しようとしたことによる永続誤りが存在する。                                                                                                                                                             | OPEN                        |
| 39 | 固定ファイル属性とプログラム中に指定した属性の間で矛盾が検出されたため、OPEN 文が不成功となった。                                                                                                                                                                             | OPEN                        |
| 41 | 開かれている状態のファイルに OPEN 文を実行しようとした。                                                                                                                                                                                                 | OPEN                        |
| 42 | 開かれていない状態のファイルに CLOSE 文を実行しようとした。                                                                                                                                                                                               | CLOSE                       |
| 43 | 順呼び出しで、CURRENT 句付きの DELETE 文または CURRENT 句付きの REWRITE 文を実行する前に、そのファイルに対して実行された最後の入出力文が、成功した READ 文でない。または WITH UNLOCK 句を指定した READ 文である。                                                                                          | DELETE, REWRITE             |
| 46 | 入力モードまたは入出力両用モードで開かれたファイルに順呼び出しの READ 文を実行しようとしたが、有効な次のレコードがない。<br>次の場合のいずれかである。<br>1. 先行する START 文が、不成功であった。<br>2. 先行する READ 文が、ファイル終了条件以外により不成功であった。<br>3. 先行する READ 文が、ファイル終了条件を引き起こした。<br>4. 先行するその他の文により、ファイル位置指示子が不定となった。 | READ                        |
| 47 | 入力モードまたは入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、READ 文、SELECT 文または START 文を実行しようとした。                                                                                                                                              | READ, SELECT, START         |
| 48 | 入出力両用モード、出力モードまたは拡張モードで開かれていない(開かれた状態でない場合も含む)ファイルに、WRITE 文を実行しようとした。または結合型仮想表に対して、WRITE 文を実行しようとした。                                                                                                                            | WRITE                       |
| 49 | 入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、DELETE 文または REWRITE 文を実行しようとした。または結合型仮想表に対して、DELETE 文または REWRITE 文を実行しようとした。                                                                                                              | DELETE, REWRITE             |
| 91 | ファイルが他のアプリケーションで排他で使用されている(IFASPRO RDB 使用時)。                                                                                                                                                                                    | OPEN                        |
| 92 | (IFASPRO RDB 使用時)<br>1. READ または START 文において、指定キーが存在しない、またはキー情報が不正である。<br>2. OUTPUT/EXTEND モードで OPEN されたファイルに対して SELECT 文を実行しようとした。<br>3. SELECT 文を行わずに SCRATCH 文を実行しようとした。                                                     | READ, STAR, SELECT, SCRATCH |
| 99 | (IFASPRO RDB 使用時)<br>1. IFASPRO RDB でエラーが発生している。詳細は IFASPRO RDB                                                                                                                                                                 | すべて                         |

|    |                                                                                          |                                           |
|----|------------------------------------------------------------------------------------------|-------------------------------------------|
|    | のマニュアルを参照。                                                                               |                                           |
| 9F | フェールオーバーが発生した(IFASPRO RDB 使用時)。                                                          | すべて                                       |
| 9J | デッドロック状態となった。ROLLBACK 文が自動的に行われた。                                                        | DELETE, READ,<br>REWRITE, START,<br>WRITE |
| 9K | 1. ロールバックログ収集エラーが発生した(IFASPRO RDB 使用時)。<br>2. COMMIT/ROLLBACK が異常終了した(COBOL DB アクセス使用時)。 | すべて                                       |
| 9M | システム表のアクセスエラー(DB 破壊)がある(IFASPRO RDB 使用時)。                                                | すべて                                       |
| 9N | ログ領域がオーバーフローした(COBOL DB アクセス使用時)。                                                        | DELETE, REWRITE,<br>WRITE                 |
| 9O | ログ領域がオーバーフローした(IFASPRO RDB 使用時)。                                                         | DELETE, REWRITE,<br>WRITE                 |
| 9P | ページまたはレコードのロックテーブルがオーバーフローした。                                                            | DELETE, READ,<br>REWRITE, WRITE           |
| 9T | ログファイルの I/O エラーがある(IFASPRO RDB 使用時)。                                                     | すべて                                       |



## 23.8 入出力状態の検査

ファイル入出力を行っているプログラムでは、プログラマはその入出力結果を必ず検査しなければならない。特定の入出力文によって起こりうる状態は、ファイル編成、呼び出し法および OPEN 文のモードによって一意に定まるので、プログラマはその起こりうる状態を調べるだけですむ。また、ある事象が発生した時点での COBOL プログラムの制御の動きは、そのファイルに対する USE 文の指定の有無や入出力文の書き方によって左右されるので、プログラマは適切な場所に入出力状態を検査する手続きを挿入しなければならない。プログラムの書き方によっては、入出力状態を二重に検査したり、どこでも検査されないことになるかもしれないので十分注意しなければならない。

プログラムの制御の動きは次の条件により決まる。

- USE 手続き の有無
- 入出力文の AT END 句または INVALID KEY 句の有無

(1)USE 手続きと AT END 句または INVALID KEY 句の指定されている場合

| 実行されるもの               | 該当する状態キー                                                                               |
|-----------------------|----------------------------------------------------------------------------------------|
| USE 手続き               | 30, 34, 35, 37, 39, 41, 42, 43, 46, 47, 48, 49, 91, 92, 99, 9F, 9J, 9M, 9N, 9P, 9T, 9K |
| AT END 句              | 10, 14                                                                                 |
| INVALID KEY 句         | 21, 22, 23, 24                                                                         |
| いずれも実行されず<br>次の命令へと進む | 00, 02, 05                                                                             |

FILE STATUS 句の有無は関係ない。

(2)USE 手続きだけが指定されている場合(入出力文に AT END 句または, INVALID KEY 句が指定されていない場合)

| 実行されるもの               | 該当する状態キー                                                                                                       |
|-----------------------|----------------------------------------------------------------------------------------------------------------|
| USE 手続き               | 10, 14, 21, 22, 23, 24, 30, 34, 35, 37, 39, 41, 42, 43, 46, 47, 48, 49, 91, 92, 99, 9F, 9J, 9M, 9N, 9P, 9T, 9K |
| いずれも実行されず<br>次の命令へと進む | 00, 02, 05                                                                                                     |

FILE STATUS 句の有無は関係ない。

(3)AT END 句または INVALID KEY 句だけが指定されている場合(USE 手続き指定なし)

| 実行されるもの                        | 該当する状態キー                                                                                           |
|--------------------------------|----------------------------------------------------------------------------------------------------|
| AT END 句                       | 10, 14                                                                                             |
| INVALID KEY 句                  | 21, 22, 23, 24                                                                                     |
| メッセージ表示後プログラム異常終了<br>(注 1)     | 30, 34, 35, 37, 39, 41, 42, 43, 46, 47, 48, 49, 91, 92, 99, 9F, 9J, 9M, 9N, 9P, 9T, 9K             |
| いずれも実行されず<br>次の命令へと進む<br>(注 2) | 00, 02, 05, 30, 34, 35, 37, 39, 41, 42, 43, 46, 47, 48, 49, 91, 92, 99, 9F, 9J, 9M, 9N, 9P, 9T, 9K |

**注意**

(注 1) FILE STATUS 句が指定されていない場合。

(注 2) FILE STATUS 句が指定されている場合。

# 第24章 実行トレース機能

## 24.1 概説

実行トレース機能の使い方について説明する。

実行トレース機能を使用することにより、COBOL アプリケーションのおおよその動作の流れを知ることができ、デバッグ時に障害発生箇所などを絞り込む有効な手段として用いることができる。

実行トレース機能は、次の情報を CSV ファイルに出力することができる。

- (1) トレース情報
- (2) レコードダンプ
- (3) カバレッジ情報
- (4) データ値

各実行トレース情報の出力は、翻訳オプションと環境変数の組合せで決定する。

環境変数 COB\_TRACE および COB\_TRACE\_LEVEL は、トレース情報、レコードダンプ、カバレッジ情報、データ値を出力するために設定する。

| 出力情報       | 環境変数<br>COB_TRACE |    | 環境変数<br>COB_TRACE_LEVEL |   |   |   |   |
|------------|-------------------|----|-------------------------|---|---|---|---|
|            | なし                | あり | 0                       | 1 | 2 | 3 | 他 |
| (1)トレース情報  | ×                 | ○  | ○                       | ○ | ○ | ○ | ○ |
| (2)レコードダンプ | ×                 | ○  | ×                       | ○ | × | ○ | × |
| (3)カバレッジ情報 | ×                 | ○  | ×                       | × | ○ | ○ | × |
| (4)データ値(*) | ×                 | ○  | ○                       | ○ | ○ | ○ | ○ |

(\*) 翻訳時にオプション指定で作成された実行可能プログラムのみ

環境変数 COB\_TRACE\_DATA は、データ値を出力するために設定する。詳細は、「24.5 データ値出力の方法」を参照。

環境変数 COB\_TRACE\_LINE および COB\_TRACE\_LABEL は、トレース情報の情報量を小さくするために設定する。詳細は、「24.4 出力範囲指定の方法」を参照。

## 24.2 トレース情報ファイルの出力方法

環境変数 COB\_TRACE, COB\_TRACE\_LEVEL, COB\_TRACE\_DATA, COB\_TRACE\_LINE, COB\_TRACE\_LABEL を使って、実行トレースおよびレコードダンプを出力する。

| 環境変数                    |                 | 説明                                                                                                                                                                                                                                                       |
|-------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB_TRACE               | COB_TRACE_LEVEL |                                                                                                                                                                                                                                                          |
| なし                      |                 | トレース情報を出力しない。                                                                                                                                                                                                                                            |
| トレース情報<br>ファイル名(フルパス付き) | 0               | 指定ディレクトリに実行トレース（カバレッジ情報なし）のみ出力する。<br>レコードダンプは出力しない。                                                                                                                                                                                                      |
|                         | 1               | 指定ディレクトリに実行トレース（カバレッジ情報なし）とレコードダンプ（レコードダンプ、キー情報（位置、長さ、ダンプ）、副キー情報（個数、位置、長さ、ダンプ））を出力する。                                                                                                                                                                    |
|                         | 2               | カバレッジ情報出力が有効な場合、指定ディレクトリに実行トレース（カバレッジ情報あり）のみ出力する。<br>レコードダンプは出力しない。<br>実行トレース(カバレッジ情報あり)に通過時刻情報を出力する。<br>カバレッジ情報出力が無効な場合、指定ディレクトリに実行トレース（カバレッジ情報なし）のみ出力する。<br>レコードダンプは出力しない。                                                                             |
|                         | 3               | カバレッジ情報出力が有効な場合、指定ディレクトリに実行トレース（カバレッジ情報あり）とレコードダンプ（レコードダンプ、キー情報（位置、長さ、ダンプ）、副キー情報（個数、位置、長さ、ダンプ））を出力する。<br>実行トレース(カバレッジ情報あり)とレコードダンプに通過時刻情報を出力する。<br>カバレッジ情報出力が無効な場合、指定ディレクトリに実行トレース（カバレッジ情報なし）とレコードダンプ（レコードダンプ、キー情報（位置、長さ、ダンプ）、副キー情報（個数、位置、長さ、ダンプ））を出力する。 |
|                         | その他             | 指定ディレクトリに実行トレース（カバレッジ情報なし）のみ出力する。<br>レコードダンプは出力しない。                                                                                                                                                                                                      |

| 環境変数                    |                          | 説明                                                   |
|-------------------------|--------------------------|------------------------------------------------------|
| COB_TRACE               | COB_TRACE_DATA           |                                                      |
| なし                      |                          | トレース情報を出力しない。                                        |
| トレース情報<br>ファイル名(フルパス付き) | なし                       | 指定ディレクトリに COB_TRACE_LEVEL に応じたトレース情報を出力する。           |
|                         | データ項目指定<br>ファイル名(フルパス付き) | 指定ディレクトリに COB_TRACE_LEVEL に応じたトレース情報と指定したデータの値を出力する。 |
|                         | その他                      | 実行時エラーを出力し、実行を終了する。                                  |

| 環境変数                      |                           | 説明                                                                                                      |
|---------------------------|---------------------------|---------------------------------------------------------------------------------------------------------|
| COB_TRACE_LINE            | COB_TRACE_LABEL           |                                                                                                         |
| 部分情報出力指定<br>ファイル名(フルパス付き) |                           | ソース単位名と行番号で指定した実行トレースとレコードダンプを出力する。<br>COB_TRACE_LEVEL の指定に関わりなく部分情報を出力する。<br>COB_TRACE_LABEL と同時指定できる。 |
|                           | 部分情報出力指定<br>ファイル名(フルパス付き) | ソース単位名と手続き名で指定した実行トレースとレコードダンプを出力する。<br>COB_TRACE_LEVEL の指定に関わりなく部分情報を出力する。<br>COB_TRACE_LINE と同時指定できる。 |

## 24.3 トレース情報ファイルの出力内容

トレース情報ファイルは CSV 形式ファイルとする。文字コードはシフト JIS である。

トレース情報ファイルには次の情報を出力する。

表 24-1 トレース情報ファイルの出力内容

| 項目           | 設定値                                                                                                                                                          | 説明                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ソース単位        | 次のいずれかである。<br>・PROGRAM-ID 段落のプログラム名<br>・FUNCTION-ID 段落の関数名<br>・CLASS-ID 段落のクラス名                                                                              | ・PROGRAM-ID 段落のプログラム名は入れ子プログラム内でも最外側のプログラム名を出力する (内側のプログラム名は出力しない)<br>・AS 定数指定ありのとき、プログラム名/関数名/クラス名を出力する                                                                |
| 行番号<br>(注 1) | 次のいずれかである。<br>・内部行番号 (10 桁)<br>コンパイラが自動付与した値<br>・外部行番号 (6 桁)<br>原始プログラムの指定値                                                                                  | ・既定値は内部行番号。<br>・-Ce オプションありのときは外部行番号を出力する<br>・固定形式の原始プログラムで外部行番号を省略したときは「999999」を出力する<br>・自由形式の原始プログラムのときは内部行番号を出力する<br>・カバレッジ情報出力ありのときは内部行番号                           |
| 通過時刻         | ・YYYYMMDDhhmmssxxxzzz 形式で通過時刻を出力する。<br>YYYY:年(4 桁)<br>MM :月(2 桁)<br>DD :日(2 桁)<br>hh :時(2 桁)<br>mm :分(2 桁)<br>ss :秒(2 桁)<br>xxx :ミリ秒(3 桁)<br>zzz :マイクロ秒(3 桁) | ・COBOL Assistant Option インストール済み、かつカバレッジ情報と部分情報出力オプション指定あり、かつ COB_TRACE_LEVEL 環境変数が 2 または 3 指定の場合、実行トレース(カバレッジ情報あり)およびレコードダンプにトレース情報出力時の通過時刻を出力する<br>・その他の場合は通過時刻を出力しない |
| 手続き文         | 次のいずれかである。<br>・手続き文の名前情報<br>・カバレッジ情報 (条件文の分岐情報)                                                                                                              | ・詳細は表 24-2 を参照<br>・手続き文が特定できないときは「***」を出力する<br>・キー情報行 (注 2 の改行された行) の場合、手続き文の後ろに「(KEY)」を付加する                                                                            |

|                 |                                                                                                                                                                                                                                                |                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 手続き名            | <ul style="list-style-type: none"> <li>・PERFORM 文が参照する利用者定義の手続き名を出力する。</li> <li>・THRU 指定ありのとき、手続き名 1 は「手続き名 1→」、手続き名 2 は「→手続き名 2」を出力する。</li> <li>・THRU 指定なしのとき、手続き名 1 は「手続き名 1」を出力する。</li> <li>・カバレッジ情報の利用者定義の手続き名 (節名, 段落名) を出力する。</li> </ul> | <ul style="list-style-type: none"> <li>・PERFORM 文 (そと PERFORM) のとき出力する</li> <li>・カバレッジ情報出力ありのときは利用者定義の手続き名を出力</li> <li>・その他のときは出力しない</li> </ul>          |
| 外部公開名           | <ul style="list-style-type: none"> <li>・CALL 文が参照するプログラム名を出力する。システムサブルーチンの呼び出しであればシステムサブルーチン名を出力する。</li> <li>・INVOKE 文が参照するメソッド名を出力する。</li> </ul>                                                                                              | <ul style="list-style-type: none"> <li>・CALL 文または INVOKE 文のとき出力する</li> <li>・その他のときは出力しない</li> </ul>                                                      |
| クラス名            | <ul style="list-style-type: none"> <li>・INVOKE 文が参照するクラス名を出力する。</li> </ul>                                                                                                                                                                     | <ul style="list-style-type: none"> <li>・INVOKE 文のとき出力する</li> <li>・その他のときは出力しない</li> <li>※ポリモルフィズムを用いている (参照され得るクラスが複数ある) 場合、その中で最上位のクラス名を出力する</li> </ul> |
| 実行時エラー (注 1)    | COB エラー番号とメッセージ文を出力する                                                                                                                                                                                                                          | <ul style="list-style-type: none"> <li>・実行時エラーのとき引用符「"」で囲んで出力する</li> <li>・正常実行のときは出力しない</li> </ul>                                                       |
| 処理モード           | 次のいずれかである。 <ul style="list-style-type: none"> <li>・INPUT</li> <li>・OUTPUT</li> <li>・I-O</li> <li>・EXTEND</li> </ul>                                                                                                                            | <ul style="list-style-type: none"> <li>・OPEN 文 (ファイル) のとき出力する</li> <li>・その他のときは出力しない</li> </ul>                                                          |
| ファイル名           | ファイル管理記述項 SELECT 句の一般形式のファイル名 1, ファイル名 2, ファイル名 3, ファイル名 4 のいずれかを出力する。                                                                                                                                                                         | <ul style="list-style-type: none"> <li>・本 COBOL では「ファイル名」を出力しない</li> </ul>                                                                               |
| 実ファイル名 / 表名     | 実ファイル名または表名を出力する。ただし、「IFAS FILE の環境設定ファイル (IFASCLT.INI)」で実ファイル名 / 表名を割り当てているとき、最終的な実ファイル名は出力不可である。                                                                                                                                             | <ul style="list-style-type: none"> <li>・OPEN 文 (ファイル), READ 文, WRITE 文, DELETE 文, REWRITE 文, START 文のとき引用符「"」で囲んで出力する</li> <li>・その他のときは出力しない</li> </ul>  |
| ファイルステータス       | 英数字 2 文字で出力する                                                                                                                                                                                                                                  | <ul style="list-style-type: none"> <li>・FILE STATUS 句ありのファイル入出力文のとき出力する</li> <li>・その他のときは出力しない</li> </ul>                                                |
| レコードダンプ (注 1)   | 16 進形式で出力する                                                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>・READ 文, WRITE 文, DELETE 文, REWRITE 文のとき、手続き実行後の内容を出力する</li> <li>・その他のときは出力しない</li> </ul>                         |
| キー情報 (個数) (注 2) | 10 進数形式で出力する                                                                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>・索引キー (合成キー指定なしの主キー) または相対キーありのとき、1 を出力する</li> <li>・索引キー (合成キー指定ありの主キ</li> </ul>                                  |

|                                   |                                          |                                                                                                                 |
|-----------------------------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
|                                   |                                          | 一)ありのとき, 合成キーの個数を出力する(2 以上)<br>・その他のときはゼロを出力する                                                                  |
| キー情報<br>(位置)<br>(注 2)(注 3)        | レコード領域先頭からのバイト数を 10 進数形式で出力する(起点はゼロ)     | ・索引キー(主キー)のとき, 出力する<br>・その他のときは出力しない                                                                            |
| キー情報<br>(長さ)<br>(注 2)(注 3)        | 10 進数形式のバイト数で出力する                        | ・相対キーまたは索引キー(主キー)のとき, 出力する<br>・その他のときは出力しない                                                                     |
| キー情報<br>(ダンプ)<br>(注 2)(注 3)       | 索引キー(主キー)は 16 進形式,<br>相対キーは 10 進数形式で出力する | ・ READ 文, WRITE 文, DELETE 文, REWRITE 文, START 文のとき, 手続き実行前の相対キーまたは索引キー(主キー)を出力する。キーなしのときは出力しない<br>・その他のときは出力しない |
| 副キー情報<br>(個数)<br>(注 2)            | 10 進数形式で出力する                             | ・索引キー(副キー)ありのとき, 出力する<br>・その他のときはゼロを出力する                                                                        |
| 副キー情報<br>(合成キーの個数)<br>(注 2)(注 4)  | 10 進数形式で出力する                             | ・索引キー(合成キー指定ありの副キー)ありのとき, 合成キーの個数を出力する(2 以上)<br>・その他のときはゼロを出力する                                                 |
| 副キー情報<br>(位置)<br>(注 2)(注 4)(注 5)  | レコード領域先頭からのバイト数を 10 進数形式で出力する(起点はゼロ)     | ・索引キー(副キー)ありのとき出力する<br>・その他のときは出力しない<br>※手続き実行前の索引キー(副キー)を出力する<br>※副キー情報(位置, 長さ, ダンプ)は副キー個数分, 繰り返し出力する          |
| 副キー情報<br>(長さ)<br>(注 2)(注 4)(注 5)  | 10 進数形式のバイト数で出力する                        |                                                                                                                 |
| 副キー情報<br>(ダンプ)<br>(注 2)(注 4)(注 5) | 16 進形式で出力する                              |                                                                                                                 |
| データ項目の値<br>(注 6)                  | 16 進形式で出力する                              | ・データ値出力が有効のとき<br>・その他のときは出力なし                                                                                   |

**注意**

- (注 1) システムサブルーチン内で実行時エラーが発生した場合は, 実行時エラー情報を改行して出力する。  
 また, その際行番号が特定できない場合は, 行番号情報が 0 となる。  
 (直前の, そのシステムサブルーチンを呼び出した CALL 文に対する出力には, 呼び出し元の行番号情報を出力する。)
- (注 2) 環境変数 COB\_TRACE\_LEVEL=1 または 3 指定のとき, 主キーまたは副キー指定があれば, 主キー情報または副キー情報を改行して出力する。  
 改行して出力する場合, データ値出力が無効であれば, 共通分情報とキー情報の項目数を一致させるよう, 共通分情報の行にカンマ区切りを追加する。データ値出力が有効であれば, 共通分情報の行に出力するデータ項目の値の位置が改行をしない行と一致するよう, カンマ区切りの追加は行わない。  
 環境変数 COB\_TRACE\_LEVEL がその他を指定または未定義のとき, 出力しない。

キー情報(個数)と副キー情報(個数)は常に出力する。環境変数 COB\_TRACE\_LEVEL がその  
他を指定または未定義の場合も、キー情報(個数)と副キー情報(個数)には「0」を出力す  
る。

(注 3) 主キー(合成キー指定なし)および相対キーなしのとき、出力しない。  
相対キーありのとき、出力する。

主キー(合成キー指定あり)のとき、指定個数分、繰り返し出力する。  
その他は出力しない。

(注 4) 副キーの個数分、繰り返し出力する。

(注 5) 副キー(合成キー指定あり)のとき、指定個数分、繰り返し出力する。  
その他は出力しない。

(注 6) 出力データ数分、データ項目指定ファイルで指定した順番に、繰り返し出力する。(注 2)の  
改行して出力する場合については、共通分情報の行にのみ出力する。

表 24-2 手続き文の出力情報

| 手続き文            | 出力情報            |
|-----------------|-----------------|
| ACCEPT 文        | ACCEPT (注 1)    |
| ADD 文           | なし              |
| ALTER 文         | なし              |
| CALL 文          | CALL            |
| CANCEL 文        | なし              |
| CLOSE 文(ファイル機能) | CLOSE           |
| CLOSE 文(画面制御機能) | CLOSE(画面) (注 1) |
| COMBINE 文       | なし              |
| COMMIT 文        | COMMIT          |
| COMPUTE 文       | なし              |
| CONTINUE 文      | なし              |
| DELETE 文        | DELETE          |
| DISPLAY 文       | DISPLAY (注 1)   |
| DIVIDE 文        | なし              |
| EVALUATE 文      | なし              |
| EXIT 文          | なし              |
| GO TO 文         | なし              |
| GOBACK 文        | なし              |
| IF 文            | なし              |
| INITIALIZE 文    | なし              |
| INSPECT 文       | なし              |
| INVOKE 文        | INVOKE          |
| MERGE 文         | MERGE (注 2)     |
| MOVE 文          | なし              |
| MULTIPLY 文      | なし              |
| OPEN 文(ファイル機能)  | OPEN            |
| OPEN 文(画面制御機能)  | OPEN(画面) (注 1)  |
| PERFORM 文       | PERFORM         |
| READ 文          | READ            |
| RELEASE 文       | RELEASE         |
| RETURN 文        | RETURN          |
| REWRITE 文       | REWRITE         |



|                               |                         |
|-------------------------------|-------------------------|
| ROLLBACK 文                    | ROLLBACK                |
| SCRATCH 文                     | SCRATCH                 |
| SEARCH 文                      | なし                      |
| SELECT 文                      | SELECT                  |
| SET 文                         | なし                      |
| SORT 文                        | SORT (注 2)              |
| START 文                       | START                   |
| STOP 文                        | なし                      |
| STRING 文                      | なし                      |
| SUBSTRING 文                   | なし                      |
| SUBTRACT 文                    | なし                      |
| UNLOCK 文                      | UNLOCK                  |
| UNSTRING 文                    | なし                      |
| USE 文                         | なし                      |
| WRITE 文                       | WRITE                   |
| ADD 文の ON SIZE ERROR          | ON SIZE ERROR (注 3)     |
| ADD 文の NOT ON SIZE ERROR      | NOT ON SIZE ERROR (注 3) |
| CALL 文の ON OVERFLOW           | ON OVERFLOW (注 3)       |
| CALL 文の ON EXCEPTION          | ON EXCEPTION (注 3)      |
| CALL 文の NOT ON EXCEPTION      | NOT ON EXCEPTION (注 3)  |
| COMPUTE 文の ON SIZE ERROR      | ON SIZE ERROR (注 3)     |
| COMPUTE 文の NOT ON SIZE ERROR  | NOT ON SIZE ERROR (注 3) |
| DELETE 文の INVALID KEY         | INVALID KEY (注 3)       |
| DELETE 文の NOT INVALID KEY     | NOT INVALID KEY (注 3)   |
| DIVIDE 文の ON SIZE ERROR       | ON SIZE ERROR (注 3)     |
| DIVIDE 文の NOT ON SIZE ERROR   | NOT ON SIZE ERROR (注 3) |
| EVALUATE 文の WHEN              | WHEN (注 3)              |
| IF 文の THEN                    | THEN (注 3)              |
| IF 文の ELSE                    | ELSE (注 3)              |
| MULTIPLY 文の ON SIZE ERROR     | ON SIZE ERROR (注 3)     |
| MULTIPLY 文の NOT ON SIZE ERROR | NOT ON SIZE ERROR (注 3) |
| READ 文の AT END                | AT END (注 3)            |
| READ 文の NOT AT END            | NOT AT END (注 3)        |
| READ 文の INVALID KEY           | INVALID KEY (注 3)       |
| READ 文の NOT INVALID KEY       | NOT INVALID KEY (注 3)   |
| RETURN 文の AT END              | AT END (注 3)            |
| RETURN 文の NOT AT END          | NOT AT END (注 3)        |
| REWRITE 文の INVALID KEY        | INVALID KEY (注 3)       |
| REWRITE 文の NOT INVALID KEY    | NOT INVALID KEY (注 3)   |
| SEARCH 文の AT END              | AT END (注 3)            |
| SEARCH 文の WHEN                | WHEN (注 3)              |
| START 文の INVALID KEY          | INVALID KEY (注 3)       |
| START 文の NOT INVALID KEY      | NOT INVALID KEY (注 3)   |
| STRING 文の ON OVERFLOW         | ON OVERFLOW (注 3)       |
| STRING 文の NOT ON OVERFLOW     | NOT ON OVERFLOW (注 3)   |
| SUBTRACT 文の ON SIZE ERROR     | ON SIZE ERROR (注 3)     |

|                               |                          |
|-------------------------------|--------------------------|
| SUBTRACT 文の NOT ON SIZE ERROR | NOT ON SIZE ERROR (注 3)  |
| UNSTRING 文の ON OVERFLOW       | ON OVERFLOW (注 3)        |
| UNSTRING 文の NOT ON OVERFLOW   | NOT ON OVERFLOW (注 3)    |
| WRITE 文の INVALID KEY          | INVALID KEY (注 3)        |
| WRITE 文の NOT INVALID KEY      | NOT INVALID KEY (注 3)    |
| WRITE 文の AT END-OF-PAGE       | AT END-OF-PAGE (注 3)     |
| WRITE 文の NOT AT END-OF-PAGE   | NOT AT END-OF-PAGE (注 3) |

**注意**

- (注 1) COBOL GUI アプリケーションでは手続き文の出力情報を出力しない。
- (注 2) SORT 文/MERGE 文と内部展開手続き (OPEN 文, CLOSE 文, RELEASE 文, RETURN 文, PERFORM 文) のトレース情報を出力する。
- (注 3) カバレッジ情報と部分情報を出力するオプションありのとき、トレース情報に条件文の分岐情報を出力する。

## 24.4 出力範囲指定の方法

部分情報出力指定ファイルは、テキスト形式であり、実行トレースおよびレコードダンプの出力範囲を指定する。

指定方法には行番号指定と手続き名指定がある。

部分情報出力指定ファイル名は、動作 OS で使用可能な文字であればよい。

部分情報出力指定ファイルの指定内容は、SJIS コードである。

COBOL アプリケーション実行中に部分情報出力指定ファイルの指定内容が更新されても反映しない。部分情報出力指定ファイルの初回読み込み時の情報で処理する。

行番号を指定してトレース情報を出力したい場合は、環境変数 COB\_TRACE\_LINE に部分情報出力指定ファイル名をフルパスで設定する。

手続き名を指定してトレース情報を出力したい場合は、環境変数 COB\_TRACE\_LABEL に部分情報出力指定ファイル名をフルパスで設定する。

たとえば、COB\_TRACE\_LINE=C:\¥INI¥XX.ini のように設定する。

部分情報出力指定ファイルの各行は次の形式でなければならない。

## ・行番号指定

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 書式  | ソース単位名[,開始行番号[:終了行番号]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 説明  | <p>トレース情報の出力範囲を行番号で指定する。<br/>         改行して繰り返し指定可能である。<br/>         部分情報出力指定ファイル中に記載のないソース単位はトレース出力の対象としない。<br/>         行番号はコンパイラが自動的に付与する内部行番号を指定する。<br/>         行番号の指定が次のいずれかの場合、動作は保証しない。</p> <ul style="list-style-type: none"> <li>・該当する行番号がない。</li> <li>・開始行番号より終了行番号が小さい。</li> <li>・行番号の指定範囲が重なっている。</li> <li>・外部行番号を指定している。</li> <li>・-Ce オプションあり (外部行番号を使用) を指定している。</li> <li>・-Ce オプションあり (外部行番号を使用) の場合、コマンドエラーにはしない。</li> </ul> <p>ソース単位名<br/>         出力するソース単位名を指定する。</p> <ul style="list-style-type: none"> <li>・PROGRAM-ID 段落のプログラム名</li> <li>・FUNCTION-ID 段落の関数名</li> <li>・CLASS-ID 段落のクラス名</li> </ul> <p>PROGRAM-ID 段落のプログラム名には入れ子プログラム内でも最外側のプログラム名を指定する (内側のプログラム名を指定しない)。<br/>         AS 定数指定ありのとき、プログラム名/関数名/クラス名を指定する。</p> <p>開始行番号<br/>         トレース情報の開始行番号を指定する。<br/>         省略時はソース単位のすべてのトレース情報を出力する。</p> <p>終了行番号<br/>         トレース情報の終了行番号を指定する。<br/>         開始行番号ありのときに指定できる。<br/>         省略時はソース単位の開始行番号以降のトレース情報を出力する。</p> |
| 指定例 | <p>PROG1,128:256<br/>         PROG2,100<br/>         PROG3</p> <p>・ソース単位名 PROG1 の行番号 128～256, ソース単位名 PROG2 の行番号 100 以降, ソース単位名 PROG3 の全行番号のトレース情報を出力する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## ・手続き名指定

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 書式  | ソース単位名[,開始手続き名[:終了手続き名]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 説明  | <p>トレース情報の出力範囲を手続き名で指定する。<br/>改行して繰り返し指定可能である。<br/>部分情報出力指定ファイル中に記載のないソース単位はトレース出力の対象としない。<br/>手続き名は節名、段落名のいずれかを指定する。<br/>手続き名の指定が次のいずれかである場合、部分情報出力の動作は保証しない。</p> <ul style="list-style-type: none"> <li>・該当する手続き名がないか、一意でない。</li> <li>・開始手続き名が終了手続き名より前に定義されていない。</li> <li>・手続き名の指定範囲が重なっている。</li> <li>・-Ce オプション (外部行番号を使用) を指定している。 <ul style="list-style-type: none"> <li>-Ce オプションあり (外部行番号を使用) の場合、コマンドエラーにはしない。</li> </ul> </li> <li>・プログラム定義の外側プログラムと内側プログラムで同じ手続き名を指定</li> <li>・プログラム定義の内側プログラム間で同じ手続き名を指定</li> <li>・クラス定義のメソッド定義間で同じ手続き名を指定</li> </ul> <p>ソース単位名<br/>出力するソース単位名を指定する。</p> <ul style="list-style-type: none"> <li>・PROGRAM-ID 段落のプログラム名</li> <li>・FUNCTION-ID 段落の関数名</li> <li>・CLASS-ID 段落のクラス名</li> </ul> <p>PROGRAM-ID 段落のプログラム名には入れ子プログラム内でも最外側のプログラム名を指定する (内側のプログラム名を指定しない)。<br/>AS 定数指定ありのとき、プログラム名/関数名/クラス名を指定する。</p> <p>開始手続き名<br/>トレース情報の開始手続き名を指定する。<br/>省略時はソース単位のすべてのトレース情報を出力する。</p> <p>終了手続き名<br/>トレース情報の終了手続き名を指定する。<br/>開始手続き名ありのときに指定できる。<br/>省略時はソース単位の開始手続き名以降のトレース情報を出力する。</p> |
| 指定例 | <p>PROG1,LABEL1:LABEL6<br/>PROG2,LABEL8<br/>PROG3</p> <p>・ソース単位名 PROG1 の手続き名 LABEL1～LABEL6, ソース単位名 PROG2 の手続き名 LABEL8 以降, ソース単位名 PROG3 の全手続き名のトレース情報を出力する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

## 24.5 データ値出力の方法

【Standard】【Enterprise for Windows】本機能は Standard Edition 版および Enterprise for Windows 版において、使用できる。

本機能は、アプリケーションのデバッグ支援として、アプリケーション実行時のデータ内容を 16 進数で出力する機能である。

本機能は、実行トレース機能と組合せて使用し、トレース情報出力と同じタイミングでデータ内容を出力する。

本機能を使用するためには、COBOL Assistant Option のインストールが必要である。

### 24.5.1 データ値出力の概要

本機能は翻訳時にデータ情報ファイルを生成し、アプリケーション実行時にデータ情報ファイルを参照することで、データ項目指定ファイルに指定した対象データをトレース情報ファイルに付加して出力する。

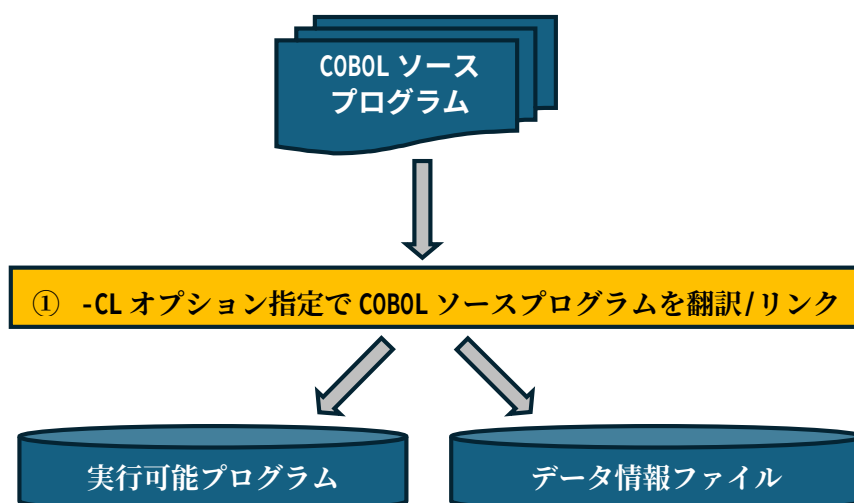
出力対象の手続き文は、実行トレース機能に従う。詳細は「表 24-2.手続き文の出力情報」を参照。

データ値出力の概要を以下に説明する。

【翻訳時】

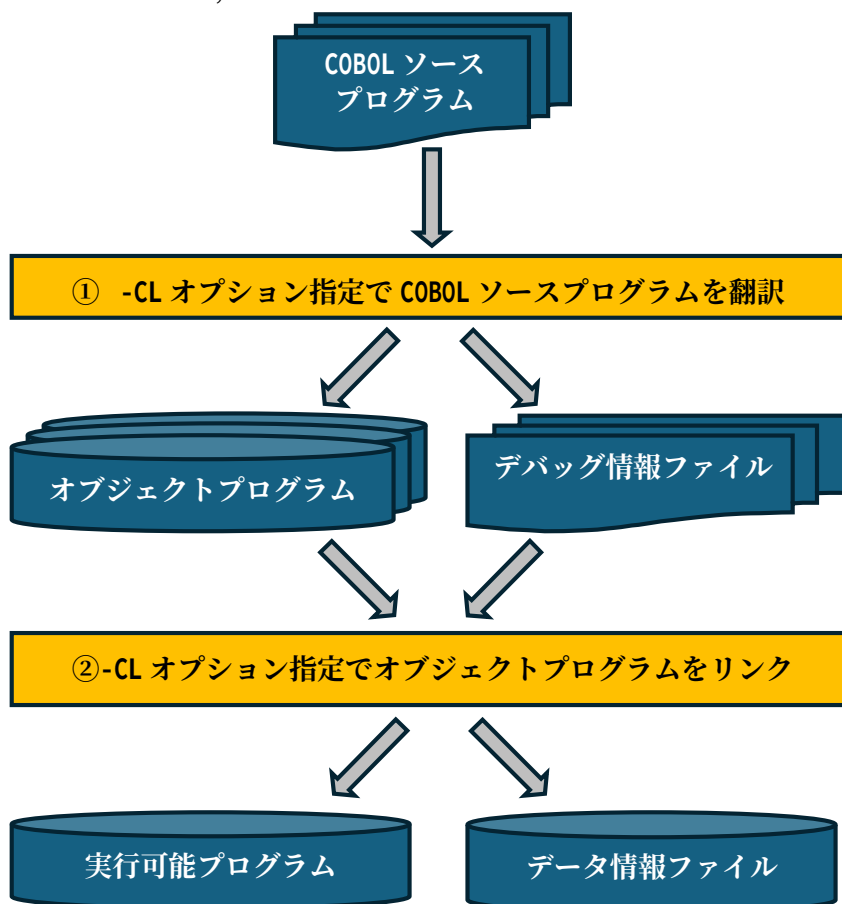
a) COBOL の翻訳とリンクを同時に行う場合

cob コマンドの -CL オプション指定で COBOL ソースプログラムを翻訳/リンクする。



b) COBOL の翻訳とリンクを別で行う場合

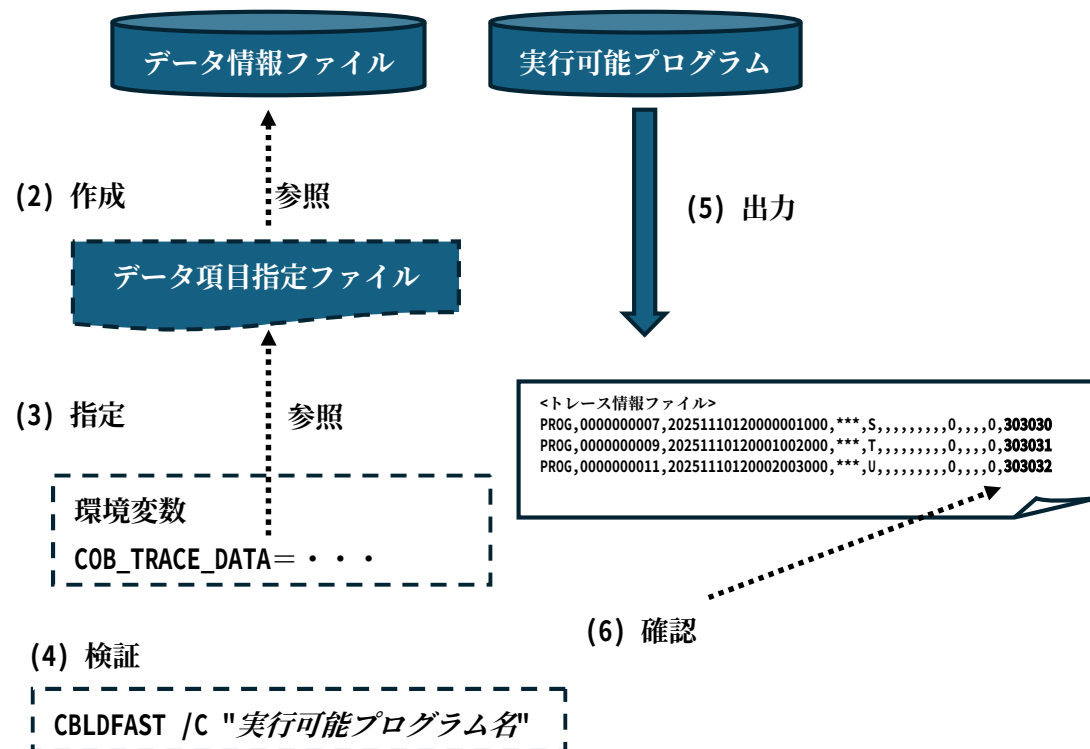
cob コマンドの-CL オプション指定で COBOL ソースプログラムをコンパイル後、オブジェクトプログラムとデバッグ情報ファイルを使用して、cob コマンドに-CL オプションを指定し、リンクを行う。



【実行時】

- (1) アプリケーションとデータ情報ファイルを実行環境に配置する
- (2) データ項目指定ファイルを作成する
- (3) 環境変数 COB\_TRACE\_DATA にデータ項目指定ファイルのフルパスを指定
- (4) COBOL 製品添付のデータ項目指定ファイル作成支援ツールを実行し、各設定に問題がないかを確認する
- (5) COBOL アプリケーションを実行し、トレース情報ファイルを出力する
- (6) トレース情報ファイルに出力されたデータの値を確認する

## (1) 配置



## 24.5.2 データ値出力の方法

## 【準備】

本機能を使用するには、次の準備を行う必要がある。

## [翻訳時]

実行可能プログラムをトレース情報にデータ項目の値を出力する翻訳オプション (-CL) を使用して作成する。

※オブジェクトプログラムの作成と実行可能プログラムの作成（リンク）を分けて行う場合、-CL オプションはオブジェクトプログラムを作成するときも、実行可能プログラムを作成するときも、指定する必要がある。

-CL オプションを指定して作成していない実行可能プログラムの場合、該当実行可能プログラム内のいずれのデータ項目の値も出力することはできない。

-CL オプションを指定して作成していないオブジェクトプログラムが実行可能プログラムに含まれる場合、該当オブジェクトプログラム内のデータ項目の値は出



力することはできない。

本オプションを指定すると、実行可能プログラム、オブジェクトプログラムに加えて、次のファイルが生成される。

|                                 |              |
|---------------------------------|--------------|
| <実行可能プログラム名>.cbldatainfo        | … データ情報ファイル  |
| <実行可能プログラム名>_debuginfo.csv      | … デバッグ情報ファイル |
| <オブジェクトプログラム名>_objdebuginfo.csv | … デバッグ情報ファイル |

#### 備考

- ・データ情報ファイルおよびデバッグ情報ファイルのファイル名は変更してはならない。
- ・デバッグ情報ファイルはデータ情報ファイルの作成に必要となるファイルであり、実行時は使用しない。

#### i) 実行可能プログラム(EXE)を cob コマンドで作成する場合

##### 例 1:

```
cob -M -CL sample1.cob sample2.cob
sample1.exe … 実行可能プログラム
sample1.cbldatainfo … データ情報ファイル
sample1_debuginfo.csv … デバッグ情報ファイル
```

#### ii) 実行可能プログラム(DLL)を cob コマンドで作成する場合

##### 例 2:

```
cob -G -CL sample1.cob sample2.cob
sample1.dll … 実行可能プログラム
sample1.cbldatainfo … データ情報ファイル
sample1_debuginfo.csv … デバッグ情報ファイル
```

#### iii) 実行可能プログラム(EXE)を翻訳とリンクに分けて作成する場合

##### 例 3:

```
cob -M -c -CL sample1.cob sample2.cob
sample1.obj … オブジェクトプログラム
sample1_objdebuginfo.csv … デバッグ情報ファイル (オブジェクト単位)
sample2.obj … オブジェクトプログラム
sample2_objdebuginfo.csv … デバッグ情報ファイル (オブジェクト単位)
cob -M -CL sample1.obj sample2.obj
```

|                       |              |
|-----------------------|--------------|
| sample1.exe           | … 実行可能プログラム  |
| sample1.cbldatainfo   | … データ情報ファイル  |
| sample1_debuginfo.csv | … デバッグ情報ファイル |

## iv) 実行可能プログラム(DLL)を翻訳とリンクに分けて作成する場合

例 4:

```
cob -c -CL sample1.cob sample2.cob
sample1.obj … オブジェクトプログラム
sample1_objdebuginfo.csv … デバッグ情報ファイル (オブジェクト単位)
sample2.obj … オブジェクトプログラム
sample2_objdebuginfo.csv … デバッグ情報ファイル (オブジェクト単位)

cob -G -CL sample1.obj sample2.obj
sample1.dll … 実行可能プログラム
sample1.cbldatainfo … データ情報ファイル
sample1_debuginfo.csv … デバッグ情報ファイル
```

## 備考

実行可能プログラム(EXE/DLL)を翻訳とリンクに分けて作成する場合、リンク時にデバッグ情報ファイル (オブジェクト単位) はオブジェクトプログラムと同じフォルダに格納する必要がある。

## v) 開発環境を使用する場合

COBOL 開発環境のプロジェクトファイル (\*.neccbproj) が保存されているフォルダには、makefile が自動生成される。この makefile を以下の手順で修正し、ビルドを行うことで実行可能プログラムと対応のデータ情報ファイルを作成する。

- (1) 対象の makefile を任意のエディターで開く。
- (2) 変数定義部に以下のオプションを追加する。
  - ・ COBOPT 変数に -CL を指定
  - ・ LKOPT 変数に -CL を指定
- (3) 【Enterprise for Windows】オブジェクトファイル移動命令の直後にデバッグ情報ファイル移動命令を追加する。

例: move /Y "test\*.obj" "Release¥"

move /Y "test\*\_objdebuginfo.csv" "Release¥" ※デバッグ情報ファイル移動

- (4) Release フォルダ内の既存の実行可能プログラムを削除する。
- (5) コマンドプロンプトで nmake コマンドを実行し、実行可能プログラムとデータ情

報ファイルを作成する。

[実行時]

- ① 実行可能プログラムとデータ情報ファイルを実行環境に配置する。
- ② 実行環境で実行トレース機能の環境変数 COB\_TRACE を設定する。  
用途に応じて環境変数 COB\_TRACE\_LEVEL, COB\_TRACE\_LINE および COB\_TRACE\_LABEL を設定する。
- ③ 環境変数 COB\_TRACE\_DATA にデータ項目指定ファイルをフルパスで設定する。  
データ項目指定ファイルは、テキスト形式であり、データ情報ファイル名、オブジェクトファイル名およびデータ項目を指定する。  
環境変数 COB\_TRACE\_DATA に指定されたデータ項目指定ファイルが存在しない、形式が無効、またはデータ情報ファイルに指定したデータ項目が見つからない場合、トレース情報ファイルに指定誤りのエラーメッセージを出力した後、実行時エラー「COB997 データ値出力の指定が正しくありません。」を出力する。

データ項目指定ファイルの各行は、次の形式でなければならない。

・1行目：データ情報ファイル名

|    |                                                                                                                                                                                          |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 書式 | データ情報ファイル名                                                                                                                                                                               |
| 説明 | -CL オプション指定で出力されたデータ情報ファイル名をフルパスで指定する。<br><実行可能プログラム名>.cbldatainfo<br><br>※実行可能プログラムとデータ情報ファイルは、同時に作成された組み合わせでなければならない。<br>指定されたデータ情報ファイルが実行可能プログラムと異なる翻訳（リンク）で作成されていた場合、データ項目の値は出力されない。 |

・2行目：オブジェクトファイル名

|    |                                                         |
|----|---------------------------------------------------------|
| 書式 | オブジェクトファイル名                                             |
| 説明 | データ項目が含まれるオブジェクトプログラムのファイル名を指定する。<br><オブジェクトプログラム名>.obj |

・3行目以降：データ項目（出力するデータ項目の個数分、1行に1データ項目ずつ指定）

|    |                                                                                                                                                                                                          |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 書式 | データ項目[,開始位置:長さ]<br># コメント                                                                                                                                                                                |
| 説明 | データ項目には、作業場所節に定義されたデータ項目を指定する。 <ul style="list-style-type: none"><li>・01/77 レベルの場合<br/>データ項目を指定する</li><li>・02 レベル～49 レベルの場合<br/>01 レベルのデータ項目と 02 レベル以降のデータ項目（FILLER 項目、無名項目は除く）をピリオドで結合して指定する。</li></ul> |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <p>※必ず完全修飾で指定する必要がある。</p> <p><b>備考</b><br/> データ項目指定ファイル作成支援ツールを使ってデータ情報ファイルに存在するデータ項目の完全修飾名一覧を作成することができる。詳細は、ユーザズガイド「5.5 データ項目指定ファイル作成支援ツール」を参照。</p> <p>英数字の場合は、プログラムが小文字で記述されている場合、大文字で指定する。<br/> 日本語の場合は、シフト JIS コードで指定する。</p> <p>データ項目に対して、開始位置と長さ（いずれもバイト単位）を指定することができる。<br/> 開始位置は1オリジンで指定する。<br/> 開始位置と長さの指定がデータの範囲外、もしくは指定が無効な場合は、トレース情報ファイルに指定誤りのエラーメッセージを出力する。</p> <p>次のデータ項目は指定できない。</p> <ul style="list-style-type: none"> <li>・レベル番号が 66, 78 または 88 のデータ項目</li> <li>・EXTERNAL 指定のデータ項目</li> <li>・ポインタ名</li> <li>・指標名</li> <li>・特殊レジスタ</li> <li>・手続き部で参照されていないデータ項目</li> <li>・クラス定義中のデータ項目</li> </ul> <p>指定した場合、トレース情報ファイルに指定誤りのエラーメッセージを出力する。</p> <p>先頭位置に#を記述すると、コメント行となる。</p> <p>指定できるデータ項目数ならびにデータ項目のサイズに制限はないが、出力するサイズに比例してトレース情報ファイルのサイズ増加や性能への影響が発生する。目安は「24.6 注意事項」を参照。</p> |
| 指定例 | <p>【01 レベルのデータ項目の場合】<br/> [プログラム記述]<br/> 01 DATA1 PIC X(240).<br/> [指定方法]<br/> DATA1,101:10<br/> [説明]<br/> DATA1 の 101 桁目から 10 桁をトレース情報に出力する。</p> <p>【03 レベルのデータ項目の場合】<br/> [プログラム記述]<br/> 01 DATA2.<br/> 02 DATA2-1.<br/> 03 DATA2-A PIC 9(3).<br/> 03 DATA2-B PIC X(5).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

|  |                                                                                                                                                                                                                                                                                                                                                                                               |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <pre> 02 DATA2-2. 03 DATA2-A PIC 9(3). 03 DATA2-B PIC X(5). </pre> <p>[指定方法]<br/>DATA2.DATA2-1.DATA2-A</p> <p>[説明]<br/>01 レベルの DATA2 と 02 レベルの DATA2-1 と 03 レベルの DATA2-A をピリオドで結合して完全修飾で指定する。</p> <p>【FILLER 項目を含む場合】</p> <p>[プログラム記述]<br/> <pre> 01 DATA2. 02 FILLER. 03 DATA2-A PIC 9(3). </pre> </p> <p>[指定方法]<br/>DATA2.DATA2-A</p> <p>[説明]<br/>FILLER 項目、無名項目を除くデータ項目をピリオドで結合して指定する。</p> |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

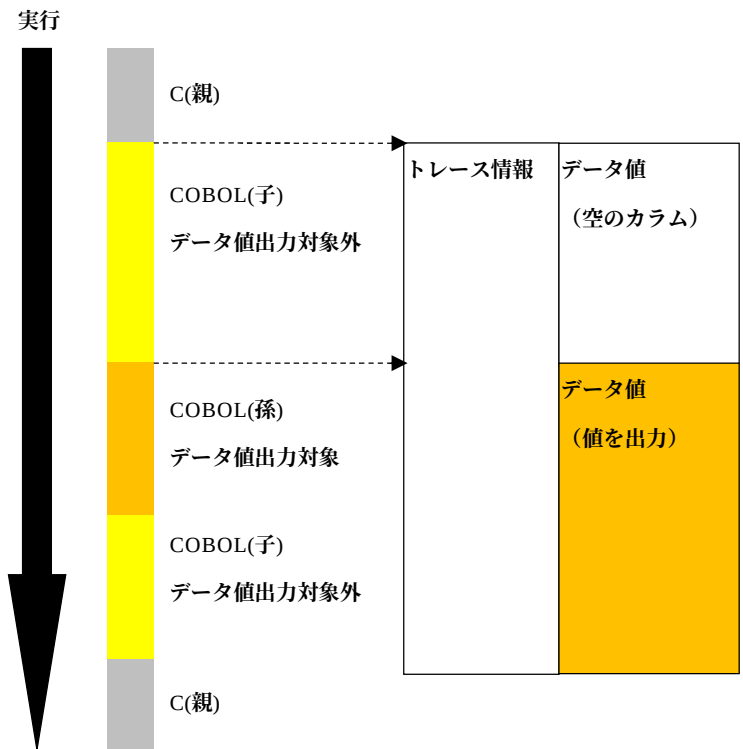
- ④ データ項目指定ファイル作成支援ツールを実行し、各設定に問題ないか確認する。
- ⑤ アプリケーションを実行し、トレース情報ファイルを出力する。

#### 備考

データ値出力対象のプログラムが初回実行されるまで、データ値は出力されない（空のカラムが出力される）。データ値出力対象のプログラムが初回実行されて以降、データ値が出力されるようになる。

#### (例)

C(親)→COBOL(子)→COBOL(孫)という関係を持つ実行可能プログラムを使用し、COBOL(孫)がデータ値出力対象の場合



データ値出力対象のプログラムが DLL 内にあり、DLL がアンロードされた場合、以降データ値は出力されない（空のカラムが出力される）。

さらに DLL がリロードされた場合、データ値出力対象のプログラムが最初に再び呼び出されて以降、データ値が出力される。

必要な場合は、トレース情報ファイルの出力範囲やデータ項目を変更し、アプリケーションの実行を繰り返すことで、プログラムの問題箇所を特定する。

### 24.5.3 データ値出力の内容

データ値出力は、トレース情報ファイルの先頭にデータ項目指定ファイルの内容を出力した後、トレース情報を出力する。トレース情報ファイルのレコード末尾に、データ項目指定ファイルに指定したデータ項目の値を 16 進数で出力する。

| データ項目指定ファイルの内容 |      |
|----------------|------|
| トレース情報         | データ値 |

**備考**

レコードダンプが有効な場合、入出力文のキー情報は改行を伴って出力されるが、データ値はキー情報行には出力されない。

キー情報行では、手続き文の後ろに"(KEY)"を付加しているため、この"(KEY)"を含む行を除去（またはフィルタリング）することで、データ値のみが含まれる行を取り出すことができる。

**<出力例>**

以下プログラムのデータ A をトレース情報として参照する場合

**[プログラム]**

```
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. PROG.
000030 DATA DIVISION.
000040 WORKING-STORAGE SECTION.
000050 01 A PIC 9(3) VALUE ZERO.
000060 PROCEDURE DIVISION.
000070 S.
000080 MOVE 1 TO A.
000090 T.
000100 ADD 1 TO A.
000110 U.
000120 STOP RUN.
```

**[環境変数]**

```
COB_TRACE=D:¥tmp¥trace¥trace.txt
COB_TRACE_LEVEL=2
COB_TRACE_DATA=D:¥tmp¥trace¥trace_data.txt
```

**[データ項目指定ファイル]**

```
D:¥tmp¥debug¥prog.cbldatainfo
prog.obj
A
```

**[トレース情報ファイル内容] 【正常時】**

```
D:¥tmp¥debug¥prog.cbldatainfo
prog.obj
```

A

PROG,0000000007,YYYYMMDDhhmmssxxxxzzz,\*\*\*,S,,,,,,,,,0,,,,,0,303030

PROG,0000000009,YYYYMMDDhhmmssxxxxzzz,\*\*\*,T,,,,,,,,,0,,,,,0,303031

PROG,0000000011,YYYYMMDDhhmmssxxxxzzz,\*\*\*,U,,,,,,,,,0,,,,,0,303032

| カラム位置 | 出力内容          | 補足説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | プログラム名        | プログラムの場合                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2     | 行番号           | <p>行番号が内部行番号（既定値）の場合は、登録集展開ありの翻訳リスト（原始プログラムリスト）と合わせて確認する必要がある。カバレッジ情報ありの場合は、内部行番号となる。</p> <p style="text-align: center;">＜原始プログラムリスト＞</p> <pre> LINE      PN SEQ-NO ***** SOURCE PROGRAM LIST   1  1 000010 IDENTIFICATION  DIVISION.   2  1 000020 PROGRAM-ID.      PROG.   3  1 000030 DATA              DIVISION.   4  1 000040 WORKING-STORAGE  SECTION.   5  1 000050 01  A              PIC 9(3)   6  1 000060 PROCEDURE        DIVISION.   7  1 000070 S.   8  1 000080      MOVE 1 TO A.   9  1 000090 T.  10  1 000100      ADD  1 TO A.  11  1 000110 U.  12  1 000120      STOP RUN. </pre> <p>行番号が外部行番号（-Ce オプション指定）の場合は、原始プログラムに記述された外部行番号となる。省略した場合は、「999999」となる。</p> |
| 5     | 手続き名          | PERFORM 文またはカバレッジ情報ありの場合                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 19    | 指定したデータ A の内容 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

上記例では、1行目から3行目までがデータ項目指定ファイルの内容、4行目から6行目がトレース情報である。

次に、各カラムの主な情報について説明する。各カラムの詳細は、『表 24-1 トレース情報ファイルの出力内容』を参照のこと。

#### 備考

環境変数 COB\_TRACE\_LEVEL が 0 または 1 の場合、上記例ではトレース情報は出力されない。



上記例では、データ A の値が 7 行目の手続き名(S)で 000(16 進：303030)⇒9 行目の手続き名(T)で 001(16 進：303031)⇒11 行目の手続き名(U)で 002(16 進：303032)と変化していることがわかる。

[トレース情報ファイル内容] 【異常時】

D:¥tmp¥debug¥prog.cbldatainfo

prog.obj

A

データ項目がデータ情報ファイルから見つかりません。(データ項目=A)

PROG,0000000007,\*\*\*,S,,,, COB997 データ値出力の指定が正しくありません。,,,,,0,,,,0,

## 24.5.4 データ値出力のエラーメッセージ

環境変数 COB\_TRACE\_DATA に指定されたデータ項目指定ファイルが存在しない、形式が無効、またはデータ情報ファイルに指定したデータ項目が見つからない場合、トレース情報ファイルに以下の指定誤りのエラーメッセージを出力する。

|       |                                                    |                                                                                                                             |
|-------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| メッセージ | 環境変数 COB_TRACE_DATA で設定されたデータ項目指定ファイルの指定が正しくありません。 |                                                                                                                             |
|       | 原因                                                 | (1) データ項目指定ファイルがフルパスで指定されていない。<br>(2) データ項目指定ファイルは存在しない<br>(3) データ項目指定ファイルのアクセス権がない                                         |
| メッセージ | データ情報ファイルの指定が正しくありません。                             |                                                                                                                             |
|       | 原因                                                 | (1) データ項目指定ファイルでデータ情報ファイルがフルパスで指定されていない。<br>(2) データ項目指定ファイルで指定されたデータ情報ファイルは存在しない<br>(3) データ項目指定ファイルで指定されたデータ情報ファイルのアクセス権がない |
| メッセージ | データ項目の指定が誤っています。(データ項目=zzzzzz)                     |                                                                                                                             |
|       | 原因                                                 | (1) データ項目指定ファイルで指定されたデータ項目の形式が正しくない<br>(2) データ項目に指定した開始位置と長さがデータ項目の範囲を超えている                                                 |
| メッセージ | データ項目がデータ情報ファイルから見つかりません (データ項目=zzzzzz)            |                                                                                                                             |
|       | 原因                                                 | 指定したデータ項目の情報がデータ情報ファイルから見つからない                                                                                              |
| メッセージ | データ情報ファイルが正しく読み取れません。                              |                                                                                                                             |
|       | 原因                                                 | (1) データ情報ファイルが壊れている<br>(2) データ情報ファイルではない                                                                                    |

## 24.6 注意事項

1. [Enterprise] マルチスレッドアプリケーションにおいて本機能を使用する場合、次の注意事項がある。
  - ・同じプロセスのスレッド毎にトレース情報ファイルを出力する。
  - ・そと PERFORM 文, CALL 文, INVOKE 文のトレース情報は出力しない。
2. 実行トレース機能を使用する場合、トレース情報ファイルの出力先には十分な空き容量が必要である。  
例: 4096 バイトのレコードを 10 万件書き込む場合、トレース情報ファイルのサイズは約 1G バイトである。
3. 別プログラム起動のシステムサブルーチン CBLRUN, CBLCHAINR, CBLCHAINC, CBLSPAWN を呼び出す CALL 文で実行時エラー発生時のトレース情報は、部分情報出力機能の指定範囲にかかわらず出力対象となります。
4. 本機能は、お客様アプリケーションにあるデータの値をトレース情報ファイルに出力するため、情報漏洩に繋がる可能性がある。個人情報や秘密情報を取り扱う際、本機能を有効にしたオプションで作成した実行可能プログラムは使用しないこと。

データ値出力を使用する場合の注意事項は以下のとおり。

5. 1つの実行可能プログラム (exe/dll) に同名のオブジェクトプログラム (obj) が含まれている場合、データ値が正しく出力されない可能性がある。
6. COBOL プログラム上の一意ではないデータ項目をデータ情報指定ファイルに記述した場合、データ値が正しく出力されない可能性がある。
7. COBOL ソースプログラムのファイル名に"," (カンマ) を使用した場合、データ値が正しく出力されない。
8. データ値出力を使用する場合、トレース情報ファイルの出力先には注意事項 4 で示した容量に対してさらに、十分な空き容量が必要である。  
例: 4,096 バイトのデータ値出力を 10 万件書き込む場合、トレース情報ファイルのサイズは約 0.4G バイト増加する。
9. FILLER または無名項目は、データ項目 (完全修飾) の一覧の出力において、上位項目と同じ完全修飾名が出力されますが、FILLER または無名項目のデータ値を出力することはできません。データ値出力においてこの完全修飾名を指定した場合、上位項目を指定したものとみなされます。

## 第25章 用語機能

〔Enterprise for Linux〕本システムの拡張機能であり、用語機能を使用する翻訳オプションを指定したときのみ使用できる。

### 25.1 概説

用語機能について、その記述方法および、操作法について述べる。なお、用語機能を利用するためには、あらかじめ用語辞書を作成しておかなければならない。用語辞書の作成方法については、『COBOL 用語エディタ リファレンスマニュアル』を参照。

### 25.2 用語機能

#### 25.2.1 用語データ項目の定義と参照

COBOL プログラム中で用語データ項目を定義するには、データ項目に対する **USAGE** 句で用語名を指定することによって行う。用語名を指定する **USAGE** 句の書き方は次のとおりである。

$$\left\{ \begin{array}{l} \text{USAGE IS 用語名} \\ \text{[USAGE IS] TERM} \end{array} \right\} \quad [\text{WITH 属性名}]$$

用語名は、用語辞書に対するキーとなる名前である。TERM 指定の場合は、そのデータ記述項で記述されたデータ名が用語名となる。

用語が複数属性のとき、WITH 句で属性名を指定する。

COBOL プログラム中で前記のような形式の **USAGE** 句を指定することにより、対応するデータ項目が用語辞書で定義された各種情報 (属性情報、用語値情報) を有する一つのデータ項目 (用語データ項目) として定義される。用語データ項目は、この項目に対する 88 条件名の宣言を除いて、この項目の後に従属するデータ項目を宣言することはできない。

## 25.2.2 用語値名

用語値名は、用語データ項目に対する用語辞書中の用語値情報によって定義される名前である。用語値情報はデータ項目の取り得る値と、その値に対して COBOL プログラム中で値を参照しようとする場合の基本型の利用語値名、およびデータ項目の特定の値または値の組にあるかどうかを調べるための複合型の利用語値名の情報よりなる。

用語値名は、その用語値名を有する値付き用語データ項目と対にして、初期値の設定や条件名を定義する VALUE 句、MOVE 命令および比較条件などの特定の構文においてのみ用いることができる。

複合型の利用語値名は、その複合型の利用語値名を有する値付き用語データ項目に対して特定の値、または値の組にあるかどうか調べる比較条件としてのみ用いることができる。

## 25.2.3 用語機能の全般規定

COBOL プログラム中で用語データ項目を定義したり、用語値名を使用する場合の全般的な規定を示す。おのおのの詳細については、『COBOL 言語説明書』を参照。

1. 用語データ項目は、データ部のすべての節において定義することができる。

用語データ項目は、手続き部での参照も含めて用語辞書に定義された属性に対応する COBOL の属性を持つ一つのデータ項目として扱われる。すなわち、一般の COBOL の規則にしたがっているかぎり、用語データ項目は対応する COBOL のデータ項目が使用できる場所ならどこでも使用することができる。

2. 値付き用語データ項目および用語値名は、COBOL プログラム内の特定な場所および特定な用法で使用する事ができる。これらの項目を使用する場合の一般的規則は次のとおりである。
  - a. 用語値名は、次に示す場所だけで使用できる。
    - VALUE 句
    - MOVE 命令の送り出し側項目
    - 条件式の比較条件 (IF, PERFORM, SEARCH, EVALUATE)
    - EVALUATE 命令の選択対象
  - b. 値付き用語データ項目は、項目の値の変更を伴わないかぎり、対応する COBOL データ項目が使用可能な場所ならどこでも使用できる。ただし、値付き用語データ項目を部分参照した場合の実行結果は利用者の責任である。

## 25.2.4 データ記述項の骨組み

用語データ項目を定義するためのデータ記述項, および用語データ項目に対する条件名を指定するための記述項を以下に示す。

書き方 1

レベル番号 [データ名 1]

[REDEFINES データ名 2]

[BASED ON ポインタ名]

{ USAGE IS 用語名  
[USAGE IS] TERM }

[WITH 属性名]

[PREFIX IS (接頭語名 1 [, 接頭語名 2])]

[VALUE { 定数  
用語値名 } ]

[OCCURS { 整数 1 TO 整数 2 TIMES DEPENDING ON データ名 3  
整数 2 TIMES } ]

[ { ASCENDING  
DESCENDING } KEY IS データ名 4]

INDEXED BY {指標名 1} ...]

[ { SYNCHRONIZED  
SYNC } { LEFT  
RIGHT } ]

[ { JUSTIFIED  
JUST } RIGHT]

[BLANK WHEN ZERO].

## 書き方 2

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

## 書き方 3

88 条件名  $\left\{ \begin{array}{l} \underline{\text{VALUE}} \text{ IS} \\ \underline{\text{VALUES}} \text{ ARE} \end{array} \right\} \{ \text{用語値名 1} \text{ [ } \left\{ \begin{array}{l} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right\} \text{ 用語値名 2} ] \} \dots$

## 書き方 4

66 データ名 5 RENAMES データ名 6 [  $\left\{ \begin{array}{l} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right\}$  データ名 7 ].

## データ記述項に関する一般的規則

1. 用語データ項目に対して条件名を指定する場合、書き方 2 または書き方 3 を用いる。値付き用語データ項目のときは書き方 3 を用いる。
2. 書き方 1 のデータ記述項に従属するデータ項目を宣言することはできない。
3. 書き方 1 で各句の順序は任意である。ただし、USAGE 句内の各句は一般形式の順序に記述しなければならない (USAGE 句→WITH 句→PREFIX 句)。
4. 書き方 4 の RENAMES 句のデータ名 6 およびデータ名 7 は、値付き用語データ項目であってはならない。
5. データ名を省略するときは、USAGE 句で TERM を指定してはならない。

## 25.2.5 USAGE 句

### 機 能

USAGE (用途) 句は、データ項目に対する用語名を指定することによって用語データ項目を定義し、COBOL プログラム内でのデータ項目の使用法を規定する。

### 一般形式

レベル番号 [データ名 1]

$$\left\{ \begin{array}{l} \text{USAGE IS 用語名} \\ \text{[USAGE IS] TERM} \end{array} \right\} \text{ [WITH 属性名]}$$

[PREFIX IS (接頭語名 1 [, 接頭語名 2])]

**備考** 一般形式中のレベル番号、データ名 1 は書き方を明確にするために示してあるだけで、レベル番号とデータ名 1 は、USAGE 句の一部ではない。

### 構文規則

1. 用語名は、用語辞書に登録されていなければならない。
2. この USAGE 句を指定したデータ記述項に従属するデータ項目を宣言することはできない。
3. 属性名は、用語名に対応する別の属性として用語辞書に登録されていなければならない。
4. TERMを指定したとき、用語名はデータ名 1 (用語データ項目名) と同じ名前が指定されたものと見なされる。データ名 1 は、REPLACE PREFIX 命令で指定した接頭語名が付加されても、接頭語名を除いた名前 (データ名 1) が用語名である。
5. データ名を省略するときは、USAGE 句で TERM を指定してはならない。
6. 用語名、属性名、および接頭語名は COBOL の利用者語の規則にしたがっていなければならない。ただし、他の利用者語との間で一意にする必要はない。また、接頭語名の最後がハイフンであってもよい。
7. 接頭語名 2 を指定した場合、接頭語名 1、接頭語名 2 のどちらか一方は日本語でなければならない。ただし、接頭語名 1、接頭語名 2 の両方ともが日本語であってはならない。

### 一般規則

1. この USAGE 句を指定することにより、用語名 (属性名の指定があれば、属性名) に対応する用語辞書で定義された属性 (用途) をもつ一つのデータ項目 (用語データ項目) が定義される。
2. 用語データ項目の内部表現形式は、別途示す対応表にしたがって COBOL のデータ項目と

見なした場合の内部表現形式と同一である。

3. PREFIX 句の指定があるとき、用語名により展開された従属項目 (要素名) のデータ名の左端に接頭語名を付加する。この時、接頭語名を付加した後の名前がデータ項目名になる。  
一般形式のデータ名 1 には接頭語名は付加されない。また、用語値名にも接頭語名は付加されない。
4. 接頭語名が日本語のときは日本語のデータ名に、接頭語名が日本語以外のときは日本語以外のデータ名に接頭語名が付加される。用語の辞書での暗黙の接頭語の付加は用語辞書の仕様にしたがう。
5. 接頭語名を付加した後の名前は COBOL の利用者語の規則に反してはならない。接頭語名を付加した後の名前の長さが大きい場合は翻訳時エラーとなるが、接頭語を付加した後の名前が予約語と同じであっても、その名前がデータ名と見なされるので注意が必要である。

例 <登録情報>

| 用語   | 属性          |
|------|-------------|
| 商品情報 | 01 品番       |
|      | 02 型番 9(5)  |
|      | 02 カラー X(5) |
|      | 01 定価 Z(9)9 |
|      | 01 在庫 Z(5)9 |

<用語の定義と従属項目の展開イメージ>

例 1 01 XX支店商品情報 USAGE IS 商品情報 PREFIX(商品).



01 XX支店商品情報.  
02 商品品番.  
03 商品型番 PIC 9(5).  
03 商品カラー PIC X(5).  
02 商品定価 PIC Z(9)9.  
02 商品在庫 PIC Z(5)9.

例 2 01 マスタファイル.

～

05 商品情報 USAGE IS TERM.  
88 品切れ VALUE 0.



01 マスタファイル.  
～  
05 商品情報.



06 品番.  
07 型番 PIC 9(5).  
07 カラー PIC X(5).  
06 定価 PIC Z(9)9.  
06 在庫 PIC Z(5)9.  
88 品切れ VALUE 0.

例2で、用語の直後に書かれた条件名「品切れ」は、用語に従属する項目の内の最後の項目「在庫」に対する条件名となる。

25.2.6 手続き部

『COBOL 言語説明書』の「付録 E. 用語データ項目の属性対応表」に示す一つの COBOL データ項目と同じとして扱われ、COBOL データ項目が使用できる場所ならばどこでも使用できる。以降に、用語データ項目の比較可能な組合せ、転記可能な組合せを示す。その他の命令での使用方法、注意については、『COBOL 言語説明書』の用語機能の章を参照。

表 25-1 比較可能な組合せ

| 左辺   |    |       | 基本項目       |       |       | 用語値名       | 定数 | 集団項目 |
|------|----|-------|------------|-------|-------|------------|----|------|
|      |    |       | 用語の型       |       | 既定義の型 |            |    |      |
|      |    |       | 用語値付き      | 用語値なし |       |            |    |      |
| 基本項目 | 用語 | 用語値付き | ○          | ○     | ○     | ○<br>(注 1) | ○  | ○    |
|      | の型 | 用語値なし | ○          | ○     | ○     | ×          | ○  | ○    |
|      |    | 既定義の型 | ○          | ○     | ○     | ×          | ○  | ○    |
| 用語値名 |    |       | ○<br>(注 1) | ×     | ×     | ×          | ×  | ×    |
| 定数   |    |       | ○          | ○     | ○     | ×          | ×  | ○    |
| 集団項目 |    |       | ○          | ○     | ○     | ×          | ○  | ○    |

**注意** (注 1)用語値名がその用語データ項目の用語値情報として含まれていなければならない。

表 25-2 転記可能な組合せ

| 受け取り側項目<br><br>送り出し側項目 |      |       | 基本項目       |       |       | 集団項目 |
|------------------------|------|-------|------------|-------|-------|------|
|                        |      |       | 用語の型       |       | 既定義の型 |      |
|                        |      |       | 用語値付き      | 用語値なし |       |      |
| 基本項目                   | 用語の型 | 用語値付き | ○          | ○     | ○     | ○    |
|                        |      | 用語値なし | ○          | ○     | ○     | ○    |
|                        |      | 既定義の型 | ○          | ○     | ○     | ○    |
| 用語値名                   |      |       | ○<br>(注 1) | ×     | ×     | ×    |
| 定数                     |      |       | ○          | ○     | ○     | ○    |
| 集団項目                   |      |       | ○          | ○     | ○     | ○    |

**注意** (注 1) 用語値名が用語データ項目の用語値情報として含まれていなければならない。

## 25.3 用語機能の例

用語データ項目の使い方について、簡単な例をあげて説明する。

ある会社の創立時からの社員情報ファイルを読み込み、昭和生まれの社員だけを選んで、その社員番号と生年月日をリストアップする。

この社員情報ファイルのレコードの一部は、他のファイルにも使われている汎用的な形式であるため、用語として用語辞書に登録する。

### <登録情報>

| 用語名 | 属性   | 用語値名:用語値                     |
|-----|------|------------------------------|
| 年号  | 9(2) | 明治:1<br>大正:2<br>昭和:3<br>平成:4 |
| 日付  | 9(6) | なし                           |

| 用語名  | 属性                     |
|------|------------------------|
| 個人情報 | 01 社員番号 9(5)           |
|      | 01 名前 N(15)            |
|      | 01 生年月日                |
|      | 02 年号1 USAGE 年号 } ...① |
|      | 02 日付1 USAGE 日付 }      |
|      | 01 住所 N(15)            |

①は、用語名「個人情報」の中の項目、「年号1」、「日付1」が、さらに用語名「年号」、「日付」によって定義されていることを示す。すなわち、用語中の属性情報をさらに用語によって定義することが可能である。これにより、年号、日付の共通化(用語化)が図られる。

次にプログラムの例を示す。

```

IDENTIFICATION DIVISION.
*
PROGRAM-ID. EXAMPLE.
*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. NX7700i.
OBJECT-COMPUTER. NX7700i.
*
INPUT-OUTPUT SECTION.
FILE-CONTROL.
 SELECT INFILE ASSIGN TO "INFILE-UFS".
 SELECT OUTFILE ASSIGN TO SPR-SYSOUT.
*
DATA DIVISION.
FILE SECTION.
FD INFILE LABEL RECORD IS STANDARD
 VALUE OF IDENTIFICATION IS "INFILE".
01 入力レコード.
 02 一連番号 PIC 9(10).
 02 個人情報 USAGE TERM. ②
 02 詳細情報 PIC N(100).
FD OUTFILE LABEL RECORD IS STANDARD.
01 出力レコード.
 02 社員番号 PIC 9(5).
 02 生年月日.
 03 年号2 USAGE 年号. } ③
 03 日付2 USAGE 日付. }
WORKING-STORAGE SECTION.
01 END-SW PIC 9(2).
*
*
```

```

PROCEDURE DIVISION.
OPEN-FILE.
 MOVE 0 TO END-SW.
 OPEN INPUT INFILE.
 OPEN OUTPUT OUTFILE.
 PERFORM UNTIL END-SW = 1
 READ INFILE
 AT END MOVE 1 TO END-SW
 EXIT PERFORM
 END-READ
 IF 年号1 = 昭和 ④
 THEN MOVE 社員番号 OF 入力レコード TO 社員番号 OF 出力レコード
 MOVE 生年月日 OF 入力レコード TO 生年月日 OF 出力レコード
 WRITE 出力レコード
 END-IF
END-PERFORM.
CLOSE INFILE OUTFILE.
STOP RUN.

```

- ② 用語辞書から「個人情報」という用語名で登録された属性が展開される。  
 次のようにコーディングしたことと同じである。

```

02 個人情報.
 03 社員番号 PIC 9(5).
 03 名前 PIC N(15).
 03 生年月日.
 04 年号1 PIC 9(2).
 04 日付1 PIC 9(6).
 03 住所 PIC N(15).

```

- ③ 用語辞書から「年号」、「日付」という用語名で登録された属性が展開される。次のようにコーディングしたことと同じである。

03 年号 2 PIC 9(2).

03 日付 2 PIC 9(6).

- ④ 用語値名「昭和」は、値 3 として展開される。次のようにコーディングしたことと同じである。

IF 年号 1 = 3

## 第26章 ファイル入出力レコード情報生成機能

### 26.1 概説

ファイル入出力レコード情報生成機能は、異なるファイルシステムへのデータ移行を支援するために COBOL プログラムのファイル節のレコード記述項からファイル入出力レコード情報を生成する機能である。

ファイル入出力レコード情報には、次の形式および用途の異なる 2 種類のファイルがある。

- ・ [Standard] [Enterprise for Windows] RFD(レコードフォーマット定義)形式  
ファイルメンテナンスキット, IFAS FILE, IFASPRO RDB の機能・ツールが扱える形式である。  
RFD 形式のファイルの表示(確認)・編集, RFD 形式のファイルに基づいたデータ変換や表定義を行うことができる。詳細は各製品に付属の説明書を参照。
- ・ CSV 形式  
特定の製品・機能に依存しない形式である。  
CSV 形式が扱える機能・ツールによるファイルの表示(確認)・編集, CSV 形式のファイルに基づいてデータ変換や表定義を行うプログラムを作成することもできる。

ファイル入出力レコード情報を生成するファイルは次のとおりである。

- ・ 大容量記憶ファイルの順編成, CSV 指定なし行順編成, 相対編成, 索引編成
  - ・ プリンタファイル, SYSIN ファイル, SYSOUT ファイル
  - ・ [Standard] RDB ファイルの順編成, 相対編成, 索引編成
- なお、次のファイルについては、ファイル入出力レコード情報を生成しない。
- ・ 大容量記憶ファイルの CSV 指定あり行順編成
  - ・ 整列併合用ファイル記述項のファイル
  - ・ [Enterprise] RAF ファイル

### 26.2 ファイル入出力レコード情報の出力方法

ファイル入出力レコード情報生成の翻訳オプションを指定したとき、ファイル記述項のファイル名毎にファイル入出力レコード情報の生成ファイルを出力する。

## 26.3 ファイル入出力レコード情報の出力内容

ファイル入出力レコード情報はテキストファイルであり、文字コードはシフト JIS である。

ファイル入出力レコード情報ファイルの構成は次のとおりである。

- (1) プログラム名レコード
- (2) ファイル名レコード
- (3) ヘッダレコード
- (4) フォーマット情報レコード (注 2)
- (5) フィールド情報レコード (注 1) (注 2)
- (6) 合成キー情報レコード (注 4) (注 5)
- (7) 合成キー構成項目レコード (注 3) (注 4) (注 5)

### 注意

- (注 1) フィールド情報レコードはフィールド個数分繰り返し出力する。
- (注 2) マルチレコードフォーマットの場合、フォーマット情報レコードとフィールド情報レコードを繰り返し出力する。
- (注 3) 合成キー構成項目レコードは合成キー情報レコードを構成するデータ項目の個数分繰り返し出力する。
- (注 4) 合成キーが複数の場合、合成キー情報レコードと合成キー構成項目レコードを繰り返し出力する。
- (注 5) [Standard] 合成キーは RDB ファイル固有仕様である。

各レコードの内容を次に説明する。

### (1) プログラム名レコード

| 項目                  | 説明                                                                                                                                                                                                |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (a) プログラム名          | <ul style="list-style-type: none"> <li>・ [Standard] [Enterprise for Windows] RFD 形式のファイル入出力レコード情報ファイル<br/>設定値: "CBL2RFD V3" 固定</li> <li>・ CSV 形式のファイル入出力レコード情報ファイル<br/>設定値: "COBOL" 固定</li> </ul> |
| (b) 拡張 RFD フォーマット指定 | <ul style="list-style-type: none"> <li>・ [Standard] [Enterprise for Windows] RFD 形式のファイル入出力レコード情報ファイル<br/>設定値: "EXTENDED" 固定</li> <li>・ CSV 形式のファイル入出力レコード情報ファイル<br/>設定なし</li> </ul>              |

### (2) ファイル名レコード

| 項目        | 説明                                                       |
|-----------|----------------------------------------------------------|
| (a) ファイル名 | <ul style="list-style-type: none"> <li>・ 設定なし</li> </ul> |

## (3) ヘッダレコード

| 項目               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |     |          |          |         |         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|----------|----------|---------|---------|
| (a) ファイル編成       | <ul style="list-style-type: none"> <li>・【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル <ul style="list-style-type: none"> <li>編成            設定値</li> <li>順編成        "SEQUENTIAL"</li> <li>相対編成     "RELATIVE"</li> <li>索引順編成   "INDEX"</li> <li>行順編成     "TEXT"</li> </ul> </li> <li>・ CSV 形式のファイル入出力レコード情報ファイル <ul style="list-style-type: none"> <li>編成            設定値</li> <li>順編成        "SEQUENTIAL"または設定なし</li> <li>相対編成     "RELATIVE"</li> <li>索引順編成   "INDEX"</li> <li>行順編成     "TEXT"</li> </ul> </li> </ul> <p><b>備考</b></p> <ul style="list-style-type: none"> <li>・ SYSIN ファイルは順編成であるが、改行コードで区切るレコード形式が順編成と異なるため、設定なしにする。</li> <li>・ SYSOUT ファイル、プリンタファイルは順編成であるが、制御データ付きのレコード形式が順編成と異なるため、設定なしにする。</li> </ul> |    |     |          |          |         |         |
| (b) コード種別        | <ul style="list-style-type: none"> <li>・ 設定値: "SJIS"固定</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |    |     |          |          |         |         |
| (c) 最大レコード長      | <ul style="list-style-type: none"> <li>・ 設定値: 1~65535(バイト数)</li> <li>・ RECORD 句(書き方 1,3,4,5)指定あり, かつ RECORD 句の最大値がレコード記述項のレコード定義長より大きい, かつ行順ファイルではない場合 <ul style="list-style-type: none"> <li>・ 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル <ul style="list-style-type: none"> <li>RECORD 句(書き方 1,3,4,5)の最大値を設定</li> </ul> </li> <li>・ CSV 形式のファイル入出力レコード情報ファイル <ul style="list-style-type: none"> <li>-Zc オプション指定時 <ul style="list-style-type: none"> <li>RECORD 句(書き方 1,3,4,5)の最大値を設定</li> </ul> </li> <li>-Zx オプション指定時 <ul style="list-style-type: none"> <li>レコード記述項のレコード定義長を設定</li> </ul> </li> </ul> </li> </ul> </li> </ul>                                                                                       |    |     |          |          |         |         |
| (d) フォーマット区分     | <table border="1"> <thead> <tr> <th>区分</th><th>設定値</th></tr> </thead> <tbody> <tr> <td>シングルレコード</td><td>"SINGLE"</td></tr> <tr> <td>マルチレコード</td><td>"MULTI"</td></tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 区分 | 設定値 | シングルレコード | "SINGLE" | マルチレコード | "MULTI" |
| 区分               | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |     |          |          |         |         |
| シングルレコード         | "SINGLE"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |     |          |          |         |         |
| マルチレコード          | "MULTI"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |     |          |          |         |         |
| (e) レコードフォーマット個数 | <ul style="list-style-type: none"> <li>・ 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル <ul style="list-style-type: none"> <li>設定値: 1~99</li> </ul> </li> <li>・ CSV 形式のファイル入出力レコード情報ファイル <ul style="list-style-type: none"> <li>設定値: 1 以上(上限値は設けない)</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |     |          |          |         |         |



|             |                |
|-------------|----------------|
| (f) バージョン情報 | ・ 設定値: "001"固定 |
| (g) エンジン名   | ・ 設定なし         |

## (4) フォーマット情報レコード

| 項目               | 説明                                                                                                                                                                                                  |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (a) レコードフォーマット番号 | <ul style="list-style-type: none"> <li>・ 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル<br/>設定値: 0~98(0 起点の一連番号)</li> <li>・ CSV 形式のファイル入出力レコード情報ファイル<br/>設定値: 0 以上(上限値は設けない)</li> </ul> |
| (b) フィールド個数      | <ul style="list-style-type: none"> <li>・ レコード有効/無効情報が"E"のフィールド個数を設定<br/>設定値: 1~65535</li> </ul>                                                                                                     |

## (5) フィールド情報レコード

・レコード有効/無効情報が"E"または"R","N"の場合

| 項目               | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|-----------|------|-------------|-------|-----------|------|-----------|-------|----------|------|--------|-------|---------------|------|---------------|-------|
| (a) レコード有効/無効情報  | <table border="1"> <thead> <tr> <th>状態</th><th>設定値</th></tr> </thead> <tbody> <tr> <td>有効</td><td>"E"</td></tr> <tr> <td>REDEFINES 句</td><td>"R"</td></tr> <tr> <td>RENAMES 句</td><td>"N"</td></tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                  | 状態 | 設定値 | 有効        | "E"  | REDEFINES 句 | "R"   | RENAMES 句 | "N"  |           |       |          |      |        |       |               |      |               |       |
| 状態               | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 有効               | "E"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| REDEFINES 句      | "R"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| RENAMES 句        | "N"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| (b) 項目名          | <ul style="list-style-type: none"> <li>・ 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル<br/>最大 30 桁(バイト数)の項目名を設定<br/>項目名が FILLER または省略した場合は"FILLER"を設定</li> <li>・ CSV 形式のファイル入出力レコード情報ファイル<br/>最大 31 文字(英数字 31 バイト, 日本語 31 文字)の項目名を設定<br/>項目名が FILLER または省略した場合は"FILLER"を設定</li> </ul>                                                                                                                                                                                                                                                                     |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| (c) フィールド開始位置    | <ul style="list-style-type: none"> <li>・ 各フィールドのレコード内開始位置を設定<br/>(0 起点のバイト位置)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| (d) フィールド長       | <ul style="list-style-type: none"> <li>・ 各フィールドのフィールド長を設定<br/>(バイト数またはビット数)</li> </ul> <p><b>備考</b><br/>ブールデータの場合はビット数</p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| (e) フィールド属性(3 桁) | <ul style="list-style-type: none"> <li>・ 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル</li> </ul> <table border="1"> <thead> <tr> <th>属性</th><th>設定値</th></tr> </thead> <tbody> <tr> <td>英数字(編集なし)</td><td>"X "</td></tr> <tr> <td>英数字(編集あり)</td><td>"XH "</td></tr> <tr> <td>日本語(編集なし)</td><td>"N "</td></tr> <tr> <td>日本語(編集あり)</td><td>"NH "</td></tr> <tr> <td>英字(編集なし)</td><td>"A "</td></tr> <tr> <td>数字編集項目</td><td>"9H "</td></tr> <tr> <td>外部 10 進(符号なし)</td><td>"9 "</td></tr> <tr> <td>外部 10 進(左端符号)</td><td>"9L "</td></tr> </tbody> </table> | 属性 | 設定値 | 英数字(編集なし) | "X " | 英数字(編集あり)   | "XH " | 日本語(編集なし) | "N " | 日本語(編集あり) | "NH " | 英字(編集なし) | "A " | 数字編集項目 | "9H " | 外部 10 進(符号なし) | "9 " | 外部 10 進(左端符号) | "9L " |
| 属性               | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 英数字(編集なし)        | "X "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 英数字(編集あり)        | "XH "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 日本語(編集なし)        | "N "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 日本語(編集あり)        | "NH "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 英字(編集なし)         | "A "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 数字編集項目           | "9H "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 外部 10 進(符号なし)    | "9 "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |
| 外部 10 進(左端符号)    | "9L "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |     |           |      |             |       |           |      |           |       |          |      |        |       |               |      |               |       |

|               | <p>外部 10 進(左端分離符号) "9LS"<br/> 外部 10 進(右端符号) "9T "<br/> 外部 10 進(符号右端分離) "9TS"<br/> 内部 10 進(符号なし) "P "<br/> 内部 10 進(符号付き) "PT "<br/> 2 進(単精度) "B1 "<br/> 2 進(倍精度) "B2 "<br/> 2 進(LittleEndian:符号なし) "B5 "<br/> 2 進(LittleEndian:符号付き) "B5S"<br/> 2 進(BigEndian:符号なし) "BB "<br/> 2 進(BigEndian:符号付き) "BBS"<br/> 集団項目 "GRP"<br/> 指標データ項目 "IND"<br/> ポインタデータ項目 "PTR"</p> <ul style="list-style-type: none"> <li>• CSV 形式のファイル入出力レコード情報ファイル</li> <li>RFD 形式のファイル入出力レコード情報ファイルのフィールド属性に加えて、次のフィールド属性も設定する。</li> </ul> <table> <tr> <th>属性</th><th>設定値</th></tr> <tr> <td>内部 10 進(符号なし)</td><td>"P2 " (注 1)</td></tr> <tr> <td>ブールデータ</td><td>"BIT"</td></tr> </table> <p><b>注意</b><br/> (注 1) -CN オプションを指定すると、符号領域なしの偶数桁の符号なし内部 10 進になる。</p> | 属性 | 設定値 | 内部 10 進(符号なし) | "P2 " (注 1) | ブールデータ | "BIT" |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|---------------|-------------|--------|-------|
| 属性            | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |               |             |        |       |
| 内部 10 進(符号なし) | "P2 " (注 1)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |     |               |             |        |       |
| ブールデータ        | "BIT"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |     |               |             |        |       |
| (f) 整数部桁数     | <ul style="list-style-type: none"> <li>• フィールド属性が外部 10 進、内部 10 進、2 進の場合、整数部の桁数を設定 <ul style="list-style-type: none"> <li>• 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル<br/> PICTURE 文字列に記号「P」がある場合、整数部桁数に PICTURE 文字列の記号「9」の数を設定</li> </ul> </li> <li>• CSV 形式のファイル入出力レコード情報ファイル<br/> PICTURE 文字列の左端に記号「P」がある場合、整数部桁数に"0"を設定。<br/> PICTURE 文字列の右端に記号「P」がある場合、整数部桁数に PICTURE 文字列の記号「9」と記号「P」の合計個数を設定</li> <li>• フィールド属性が英数字、日本語、英字の場合、"0"を設定</li> </ul>                                                                                                                                                                                                                                                                 |    |     |               |             |        |       |
| (g) 小数部桁数     | <ul style="list-style-type: none"> <li>• フィールド属性が外部 10 進、内部 10 進、2 進の場合、小数部の桁数を設定 <ul style="list-style-type: none"> <li>• 【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル<br/> PICTURE 文字列に記号「P」がある場合、小数部桁数に"0"を設定</li> </ul> </li> <li>• CSV 形式のファイル入出力レコード情報ファイル<br/> PICTURE 文字列の左端に記号「P」がある場合、小数部桁数に PICTURE 文字列の記号「P」と記号「9」の合計個数を設定。<br/> PICTURE 文字列の右端に記号「P」がある場合、小数部桁数に"0"を設定</li> <li>• フィールド属性が英数字、日本語、英字の場合、"0"を設定</li> </ul>                                                                                                                                                                                                                                                                                  |    |     |               |             |        |       |

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----|---------------|-----|---------------|-----|---------------|-----|---------------|-----|---------|-----|-------------|-----|
| (h) 編集付加情報       | <ul style="list-style-type: none"> <li>省略時は"NN"と同意</li> <li>BLANK WHEN ZERO 句の指定がある場合, "YZ"を設定する。</li> <li>属性が英数字(編集あり),日本語(編集あり),数字(編集あり),数字(PICTURE 文字列に S, V または P 文字あり)の場合, "YD"を設定する。</li> <li>1 桁目は編集情報の有無を設定<br/> <table> <tr> <td>編集情報有無</td><td>設定値</td></tr> <tr> <td>編集情報あり</td><td>"Y"</td></tr> <tr> <td>編集情報なし</td><td>"N"</td></tr> </table> </li> <li>2 桁目は出力編集情報の内容を設定<br/> <table> <tr> <td>出力編集情報</td><td>設定値</td></tr> <tr> <td>なし</td><td>"N"</td></tr> <tr> <td>ゼロサプレス</td><td>"Z"</td></tr> <tr> <td>PICTURE 句参照</td><td>"D"</td></tr> </table> </li> </ul> | 編集情報有無 | 設定値 | 編集情報あり        | "Y" | 編集情報なし        | "N" | 出力編集情報        | 設定値 | なし            | "N" | ゼロサプレス  | "Z" | PICTURE 句参照 | "D" |
| 編集情報有無           | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 編集情報あり           | "Y"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 編集情報なし           | "N"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 出力編集情報           | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| なし               | "N"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| ゼロサプレス           | "Z"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| PICTURE 句参照      | "D"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (i) 単一キー情報       | <ul style="list-style-type: none"> <li>索引ファイルのキー情報を設定<br/> <table> <tr> <td>キーの種別</td><td>設定値</td></tr> <tr> <td>主キー項目(二重キーなし)</td><td>"M"</td></tr> <tr> <td>主キー項目(二重キーあり)</td><td>"N"</td></tr> <tr> <td>副キー項目(二重キーなし)</td><td>"U"</td></tr> <tr> <td>副キー項目(二重キーあり)</td><td>"D"</td></tr> <tr> <td>キー項目でない</td><td>" "</td></tr> </table> </li> </ul>                                                                                                                                                                                                                            | キーの種別  | 設定値 | 主キー項目(二重キーなし) | "M" | 主キー項目(二重キーあり) | "N" | 副キー項目(二重キーなし) | "U" | 副キー項目(二重キーあり) | "D" | キー項目でない | " " |             |     |
| キーの種別            | 設定値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 主キー項目(二重キーなし)    | "M"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 主キー項目(二重キーあり)    | "N"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 副キー項目(二重キーなし)    | "U"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| 副キー項目(二重キーあり)    | "D"                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| キー項目でない          | " "                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (j) 入カチェック/編集情報  | <ul style="list-style-type: none"> <li>設定なし</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (k) 既定値/有効値リスト   | <ul style="list-style-type: none"> <li>設定なし</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (l) PICTURE 情報   | <ul style="list-style-type: none"> <li>編集付加情報が"YZ"または"YD"である場合 (フィールド属性が英数字(編集あり),日本語(編集あり),数字(編集あり),数字(PICTURE 文字列に S, V または P 文字あり)), PICTURE 句に指定した編集の形式をそのままの形式または配列形式で設定。省略可能な想定小数点は出力しない</li> <li>編集付加情報が"YZ"または"YD"でない場合, 省略か空白を設定</li> </ul>                                                                                                                                                                                                                                                                                                                               |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (m) JUSTIFIED 情報 | <ul style="list-style-type: none"> <li>JUSTIFIED 句がある場合, "J"を設定</li> <li>JUSTIFIED 句がない場合, 省略または空白を設定</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (n) レベル番号        | <ul style="list-style-type: none"> <li>項目のレベル番号を設定</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |     |               |     |               |     |               |     |               |     |         |     |             |     |
| (o) インデックス名      | <ul style="list-style-type: none"> <li>設定なし</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |        |     |               |     |               |     |               |     |               |     |         |     |             |     |

・レコード有効/無効情報が"S"または"F"の場合

| 項目              | 説明                                                                                                                                                                                                                               |    |     |           |     |           |     |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|-----------|-----|-----------|-----|
| (a) レコード有効/無効情報 | <table> <tr> <td>状態</td><td>設定値</td></tr> <tr> <td>OCCURS 開始</td><td>"S"</td></tr> <tr> <td>OCCURS 終了</td><td>"F"</td></tr> </table>                                                                                             | 状態 | 設定値 | OCCURS 開始 | "S" | OCCURS 終了 | "F" |
| 状態              | 設定値                                                                                                                                                                                                                              |    |     |           |     |           |     |
| OCCURS 開始       | "S"                                                                                                                                                                                                                              |    |     |           |     |           |     |
| OCCURS 終了       | "F"                                                                                                                                                                                                                              |    |     |           |     |           |     |
| (b) 繰り返し下限値     | <ul style="list-style-type: none"> <li>OCCURS 句の繰り返しの下限値を設定</li> <li>レコード有効/無効情報が"F"(OCCURS 終了)の場合, 設定なし</li> </ul>                                                                                                              |    |     |           |     |           |     |
| (c) 繰り返し上限値     | <ul style="list-style-type: none"> <li>【Standard】【Enterprise for Windows】RFD 形式のファイル入出力レコード情報ファイル<br/>OCCURS 句の繰り返しの上限値を設定<br/>設定値: 1~32767<br/>レコード有効/無効情報が"F"(OCCURS 終了)の場合, 設定なし</li> <li>CSV 形式のファイル入出力レコード情報ファイル</li> </ul> |    |     |           |     |           |     |

|             |                                                                           |
|-------------|---------------------------------------------------------------------------|
|             | OCCURS 句の繰り返しの上限値を設定<br>設定値：1～65535<br>レコード有効/無効情報が"F"(OCCURS 終了)の場合、設定なし |
| (d) 繰り返し項目長 | ・ OCCURS 句の繰り返し項目長をバイト数で設定<br>・ レコード有効/無効情報が"F"(OCCURS 終了)の場合、設定なし        |

## (6) 合成キー情報レコード

| 項目          | 説明                                                                                                                                                                                                                                                                                                     |       |     |             |     |             |     |             |     |             |     |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|-----|-------------|-----|-------------|-----|-------------|-----|
| (a) 識別情報    | ・ 設定値："G"固定                                                                                                                                                                                                                                                                                            |       |     |             |     |             |     |             |     |             |     |
| (b) キー情報    | ・ 合成キーのキー情報(キー種別)を設定<br><table border="1"> <thead> <tr> <th>キーの種別</th><th>設定値</th></tr> </thead> <tbody> <tr> <td>主キー(二重キーなし)</td><td>"M"</td></tr> <tr> <td>主キー(二重キーあり)</td><td>"N"</td></tr> <tr> <td>副キー(二重キーなし)</td><td>"U"</td></tr> <tr> <td>副キー(二重キーあり)</td><td>"D"</td></tr> </tbody> </table> | キーの種別 | 設定値 | 主キー(二重キーなし) | "M" | 主キー(二重キーあり) | "N" | 副キー(二重キーなし) | "U" | 副キー(二重キーあり) | "D" |
| キーの種別       | 設定値                                                                                                                                                                                                                                                                                                    |       |     |             |     |             |     |             |     |             |     |
| 主キー(二重キーなし) | "M"                                                                                                                                                                                                                                                                                                    |       |     |             |     |             |     |             |     |             |     |
| 主キー(二重キーあり) | "N"                                                                                                                                                                                                                                                                                                    |       |     |             |     |             |     |             |     |             |     |
| 副キー(二重キーなし) | "U"                                                                                                                                                                                                                                                                                                    |       |     |             |     |             |     |             |     |             |     |
| 副キー(二重キーあり) | "D"                                                                                                                                                                                                                                                                                                    |       |     |             |     |             |     |             |     |             |     |
| (c) キーの項目数  | ・ 設定値：2～16                                                                                                                                                                                                                                                                                             |       |     |             |     |             |     |             |     |             |     |
| (d) インデックス名 | ・ 設定なし                                                                                                                                                                                                                                                                                                 |       |     |             |     |             |     |             |     |             |     |

## (7) 合成キー構成項目レコード

| 項目            | 説明                                                                                                                                       |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|
| (a) 識別情報      | ・ 設定値："H"固定                                                                                                                              |
| (b) 項目名       | ・ RFD 形式のファイル入出力レコード情報ファイル<br>最大 30 桁(バイト数)の合成キーの項目名を設定<br>・ CSV 形式のファイル入出力レコード情報ファイル<br>最大 31 文字(英数字 31 バイト, 日本語 31 文字)の<br>合成キーの項目名を設定 |
| (c) フィールド開始位置 | ・ 合成キーのレコード内開始位置を設定(0 起点のバイト位置)                                                                                                          |
| (d) フィールド長    | ・ 合成キーの項目長を設定(バイト数)                                                                                                                      |
| (e) フィールド属性   | ・ 合成キーのフィールド属性を設定<br>設定値はフィールド情報レコードのフィールド属性を参照                                                                                          |
| (f) 繰り返し番号    | ・ 設定なし                                                                                                                                   |

## 26.4 注意事項

本機能を使用する場合、次の注意事項がある。

- ・ [Standard] [Enterprise for Windows] RFD 形式のファイル入出力レコード情報ファイルのフィールド長とフィールド属性は、ファイルメンテナンスキット, IFAS FILE, IFASPRO RDB のサポート範囲である。サポート範囲外の記述をしないよう注意が必要。

## 第27章 データレコード情報出力機能

### 27.1 概説

COBOL 原始プログラムの PROCEDURE DIVISION USING データ名および RETURNING データ名の定義内容を見える化して、CSV 形式のデータレコード情報を出力する機能である。

本機能を使用するためには、COBOL Assistant Option のインストールが必要である。

### 27.2 データレコード情報の出力方法

データレコード情報出力の翻訳オプションを指定したとき、PROCEDURE DIVISION USING データ名および RETURNING データ名に対して、データレコード情報ファイルを出力する。

## 27.3 データレコード情報の出力内容

データレコード情報はテキストファイルであり、文字コードはシフト JIS である。

データレコード情報ファイルの出力内容を次のとおりである。

| 項目名        | 説明                                                                                                                                                                                           |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME       | 外部公開名を設定する。<br><ul style="list-style-type: none"> <li>・ [最外側の] プログラム定義の場合<br/>PROGRAM-ID のプログラム名。AS 指定があれば定数。</li> <li>・ 関数定義の場合<br/>FUNCTION-ID の利用者関数名。AS 指定があれば定数。</li> </ul>             |
| CALL       | 外部手続き呼び出し規約を設定する。<br>【Enterprise for Windows】【Standard】<br>__fastcall<br>【Enterprise for Linux】<br>x64 System V ABI (Application Binary Interface)                                           |
| RETURN     | PROCEDURE DIVISION RETURNING 返却項目の情報を設定する。<br><ul style="list-style-type: none"> <li>・ 返却項目なしの場合<br/>void</li> <li>・ 返却項目ありの場合<br/>「表 27-1 データ名の詳細」を参照する。</li> </ul>                         |
| RETURN-SUB | 返却項目の従属項目の情報を設定する。<br><ul style="list-style-type: none"> <li>・ 返却項目が基本項目で従属項目がない場合<br/>RETURN-SUB レコードは出力しない</li> <li>・ 返却項目が集団項目で従属項目がある場合<br/>「表 27-1 データ名の詳細」を参照する。</li> </ul>            |
| PARAM-CNT  | PROCEDURE DIVISION USING 仮パラメータ情報の個数を設定する。                                                                                                                                                   |
| PARAM-TYPE | 仮パラメータ情報のタイプを設定する。<br>0 : BY REFERENCE<br>1 : BY VALUE                                                                                                                                       |
| PARAM      | 仮パラメータ情報の詳細を設定する。<br>「表 27-1 データ名の詳細」を参照する。                                                                                                                                                  |
| PARAM-SUB  | 仮パラメータ情報の従属項目の情報を設定する。<br><ul style="list-style-type: none"> <li>・ 仮パラメータ情報が基本項目で従属項目がない場合<br/>PARAM-SUB レコードは出力しない</li> <li>・ 仮パラメータ情報が集団項目で従属項目がある場合<br/>「表 27-1 データ名の詳細」を参照する。</li> </ul> |

### 備考

- ・ COBOL 原始プログラムに複数翻訳単位を含む場合は、NAME から PARAM-SUB レコードまで繰り返し出力する。
- ・ 仮パラメータ情報の個数分、PARAM-TYPE から PARAM-SUB レコードまで繰り返し出力する。  
 仮パラメータなしの場合は PARAM-TYPE から PARAM-SUB レコードは出力しない。

表 27-1 データ名の詳細

| 項目名    | 説明                                                                                                                                                                                 |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| データ名   | データ名を" "で囲み設定する。<br>無名項目の場合は"" (空文字列)を設定する。<br>データ名が集団項目の従属項目の場合、完全修飾 (例 A.B) にて設定する。<br>データ名が無名項目を途中に含む場合、無名項目に対する「.」を設定しない。                                                      |
| データレベル | 連絡節にて定義されたデータのレベル番号を 2 桁で設定する。                                                                                                                                                     |
| 定義情報   | 連絡節内でデータ名が一意であるかどうかを設定する。<br>0：一意でない<br>1：一意である                                                                                                                                    |
| データ属性  | 「表 27-2 データ属性」を参照する。                                                                                                                                                               |
| データ長   | データの長さ(メモリ上のバイト数)を設定する。<br>ブールデータ項目の場合、ビット数を設定する。                                                                                                                                  |
| 符号情報   | 符号編集用の文字が含まれているか否か、含まれている場合は、その種別を設定する。<br>外部 10 進[USAGE DISPLAY]以外で、符号編集用文字がある場合「あり」とする。<br>「表 27-3 符号情報」を参照する。                                                                   |
| 整数部の桁数 | 整数部の桁数を設定する。<br>固定長集団項目、可変長集団項目、英数字編集項目、数字編集項目、日本語編集項目、日本語項目、英字項目、英数字項目、ポインタデータ項目、オブジェクト参照項目、ブールデータ項目、指標データ項目の場合は 0 を設定する。<br>単精度固定 2 進数の場合は 05 を設定する。<br>倍精度固定 2 進数の場合は 10 を設定する。 |
| 小数部の桁数 | 小数部の桁数を設定する。<br>固定長集団項目、可変長集団項目、英数字編集項目、数字編集項目、日本語編集項目、日本語項目、英字項目、英数字項目、単精度固定 2 進数、倍精度固定 2 進数、ポインタデータ項目、オブジェクト参照項目、ブールデータ項目、指標データ項目の場合は 0 を設定する。                                   |
| 相対アドレス | レベル番号 01, 77 先頭からのバイト位置を設定する。<br>OCCURS 繰り返し項目の場合は 1 要素目のバイト位置を設定する。                                                                                                               |
| 次元数    | 次元数を設定する。<br>最大 7 次元で、繰り返しなしの場合は 0 を設定する。                                                                                                                                          |
| 繰り返し数  | OCCURS 句の繰り返し数を設定する。                                                                                                                                                               |
| 繰り返し長  | 繰り返し項目の長さ(メモリ上のバイト数)を設定する。                                                                                                                                                         |

表 27-2 データ属性

| 値  | データ属性                                                          |
|----|----------------------------------------------------------------|
| 01 | 固定長集団項目                                                        |
| 02 | 可変長集団項目                                                        |
| 03 | 英数字編集項目                                                        |
| 04 | 数字編集項目                                                         |
| 05 | 日本語編集項目                                                        |
| 06 | 日本語項目                                                          |
| 07 | 英字項目                                                           |
| 08 | 英数字項目                                                          |
| 10 | 外部 10 進数[USAGE DISPLAY]<br>外部 10 進数[USAGE COMP] (-Fc オプション指定時) |
| 11 | 内部 10 進数[USAGE COMP-3/USAGE PACKED DECIMAL]                    |
| 12 | 単精度固定 2 進数[USAGE COMP-1]                                       |
| 13 | 倍精度固定 2 進数[USAGE COMP-2]                                       |
| 17 | 2 進数[USAGE BINARY/USAGE COMP]                                  |
| 18 | 2 進数[USAGE COMP-5]<br>2 進数[USAGE COMP] (-C5 オプション指定時)          |
| 20 | ポインタデータ項目[USAGE POINTER]                                       |
| 21 | オブジェクト参照項目[USAGE OBJECT REFERENCE]                             |
| 22 | ブールデータ項目[USAGE BIT]                                            |
| 23 | 指標データ項目[USAGE INDEX]                                           |

表 27-3 符号情報

| 値 | 符号編集用文字の有無／種別 |
|---|---------------|
| 0 | なし            |
| 1 | あり            |
| 2 | 重ね符号あり(右端)    |
| 3 | 分離符号あり(右端)    |
| 4 | 重ね符号あり(左端)    |
| 5 | 分離符号あり(左端)    |



# 付録

## A.制限事項/注意事項

### A.1 翻訳時の諸元

次に本 COBOL の翻訳時の諸元を記述する。

諸元には、本 COBOL 独自の諸元と、言語説明書に記載されている諸元がある。なお、ここで記述している項目の内翻訳時にメッセージを出力しないものもあるので注意が必要である。

#### A.1.1 各部に共通なもの

翻訳可能行数や定義可能な利用者定義語の個数は記述内容やコンパイラ起動時のメモリ容量による。

- |     |                                                        |                                          |
|-----|--------------------------------------------------------|------------------------------------------|
| (1) | 翻訳可能行数                                                 | 制限なし                                     |
| (2) | 定義可能な利用者定義語の個数                                         | 約 550,000 個                              |
| (3) | COBOL ソースファイルの 1 行の最大桁数                                | 255 桁                                    |
| (4) | 利用者定義語の文字数                                             | 31 文字以内                                  |
| (5) | プログラム名, プログラム原型名, クラス名, 利用者定義関数名, 関数原型名, メソッド名, プロパティ名 | 31 文字以内                                  |
| (6) | 外部属性をもつデータ名の文字数                                        | 256 文字以内                                 |
| (7) | 外部属性をもつファイルのファイル名長                                     | 最大 256 文字<br>(ただし、実行時に有効なファイル名長は、246 文字) |
| (8) | 翻訳単位毎のプログラムの個数                                         | 1023 個                                   |

#### A.1.2 データ部に関するもの

1 つの翻訳単位に定義できる文字数はコンパイラ起動時のメモリ容量による。また実行時には、定義したデータ以上のメモリ容量が必要である。

- |     |                  |             |
|-----|------------------|-------------|
| (1) | 外部名の個数           | 最大 65,535 個 |
| (2) | ポインタ名の個数         | 最大 65,535 個 |
| (3) | 指標名の個数           | 最大 16,383 個 |
| (4) | 索引ファイルの参照キー項目の桁数 |             |
|     | IFAS FILE 形式のとき  | 256 桁       |

|      |                                     |                 |
|------|-------------------------------------|-----------------|
|      | COBOL DB アクセス形式のとき                  | 256 桁           |
|      | ISAM 形式のとき                          | 255 桁           |
| (5)  | ALTERNATE RECORD KEY 句の数            |                 |
|      | [Enterprise for Windows]            | 制限なし            |
|      | [Standard]                          | 最大 7 個          |
| (6)  | SORT/MERGE 文のキー項目の個数                | 最大 64 個         |
| (7)  | SORT/MERGE 文のキー項目の合計文字数             | 4096 文字         |
| (8)  | PICTURE 句の文字列                       | 最大 30 文字        |
| (9)  | 英数字定数の文字数                           | 最大 256 文字       |
| (10) | 数字定数の桁数                             | 最大 18 桁         |
| (11) | 日本語定数の文字数                           | 最大 128 文字       |
| (12) | 数字項目の桁数                             | 1～18 桁          |
|      | [Standard] 画面節のとき                   | 18(小数部 15)桁     |
|      | [Standard] 画面節で CHECK MODULUS ありのとき | 16 桁            |
| (13) | 英字/英数字項目の文字数                        |                 |
|      | ファイル節のとき                            | 1～65535 文字      |
|      | その他の節のとき                            | 1～2147483647 文字 |
|      | [Standard] 画面節のとき                   | 255 文字          |
| (14) | 日本語項目の文字数                           |                 |
|      | ファイル節のとき                            | 1～32767 文字      |
|      | その他の節/EXTERNAL 句のとき                 | 1～1073741823 文字 |
|      | [Standard] 画面節のとき                   | 127 文字          |
| (15) | 集団項目の文字数                            |                 |
|      | ファイル節のとき                            | 1～65535 文字      |
|      | その他の節のとき                            | 1～2147483647 文字 |
| (16) | 数字編集項目の数字の桁位置                       | 1～18 桁          |
| (17) | 数字編集項目の数字/挿入文字位置の合計                 | 最大 31 文字        |
| (18) | 英数字編集の文字数                           | 1～256 文字        |
|      | [Standard] 画面節のとき                   | 255 文字          |
| (19) | 日本語編集項目の文字数                         | 1～128 文字        |
|      | [Standard] 画面節のとき                   | 127 文字          |
| (20) | ブール定数およびブール項目の文字数                   | 最大 64 文字        |
| (21) | 表の次元数                               | 7 次元            |
| (22) | 1つの表で指定できるキー項目の数                    | 制限なし            |
| (23) | 表の繰り返し数(OCCURS 句の整数)                |                 |
|      | ファイル節のとき                            | 最大 65535        |

|      |                                       |                  |
|------|---------------------------------------|------------------|
|      | その他の節のとき .....                        | 最大 2147483647    |
| (24) | 表の一要素の大きさ                             |                  |
|      | ファイル節のとき .....                        | 最大 65535 文字      |
|      | その他の節のとき .....                        | 最大 2147483647 文字 |
| (25) | 相対添字の大きさ .....                        | 最大 2147483647    |
| (26) | 連絡節(LINKAGE SECTION)の 01, 77 項目 ..... | 1025 個           |
| (27) | ファイル節に定義可能な項目の大きさ .....               | 最大 65535         |
| (28) | 作業場所節に定義可能な項目の大きさ .....               | 最大 2147483647    |
| (29) | 連絡節に定義可能な項目の大きさ .....                 | 最大 2147483647    |
| (30) | ベースデータ節に定義可能な項目の大きさ .....             | 最大 2147483647    |
| (31) | 局所記憶節に定義可能な項目の大きさ .....               | 最大 2147483647    |
| (32) | 論理ページの行数 .....                        | 最大 32767 行       |
| (33) | プリンタファイルにおける 1 レコードの最大長               |                  |
|      | APPLY SHIFT-CODE 指定なしのとき .....        | 65535 バイト        |
|      | APPLY SHIFT-CODE 指定ありのとき .....        | 512 バイト          |

## A.1.3 手続き部に関するもの

文あたりに指定できる作用対象の個数はコンパイラ起動時のメモリ容量による。

|      |                                                                         |             |
|------|-------------------------------------------------------------------------|-------------|
| (1)  | PROCEDURE DIVISION USING のデータ名 .....                                    | 1024 個      |
| (2)  | 節名, 段落名およびコンパイラ内部生成ラベルの合計数 .....                                        | 最大 65534 個  |
| (3)  | 文あたりに指定できる作用対象の個数 (DISPLAY 文, INSPECT 文, STRING 文, UNSTRING 文を除く) ..... | 制限なし        |
| (4)  | 算術式中の ” ( “ のネスト個数 .....                                                | 最大 60 個     |
| (5)  | べき乗の指数の桁数 .....                                                         | 最大 18 桁     |
| (6)  | CALL 文のパラメータ数 .....                                                     | 1024 個      |
| (7)  | INSPECT 文の一意名 3 から一意名 7 の最大桁数 .....                                     | 最大 256 桁    |
| (8)  | INSPECT 文の作用対象の個数 .....                                                 | 最大 255 個    |
| (9)  | DISPLAY 文の作用対象(一意名, 定数)の個数 .....                                        | 最大 255 個    |
| (10) | STRING 文の作用対象の個数 .....                                                  | 最大 255 個    |
| (11) | UNSTRING 文の作用対象 .....                                                   | 最大 255 個    |
| (12) | 1 回の WRITE 文での改行数 .....                                                 | 0~32767     |
| (13) | 修飾語の数 .....                                                             | 最大 50 個     |
| (14) | ACCEPT 文(小入力)の作用対象の文字数                                                  |             |
|      | [Standard] .....                                                        | 最大 32767 文字 |

(15) DISPLAY 文(小出力)の作用対象の文字数

[Standard] .....最大 32767 文字

A.1.4 オブジェクト指向機能に関するもの

(1) クラス継承のネスト個数

[Enterprise for Windows] [Standard] ..... 最大 512 個

(Visual C++の定量制限に依存するため、最大個数まで使用できない場合がある)

[Enterprise for Linux] ..... 最大 64 個

(2) プロパティ OF 一意名の OF 修飾の個数 ..... 最大 50 個

(3) クラス内メソッドの個数..... 1023 個

A.1.5 リポジトリ機能に関するもの

ファイル名に使用可能な文字数は、OS のファイル名長制限による。

(1) 外部リポジトリファイルのファイル名..... 最大 256 文字

A.2 注意事項

(1) 本コンパイラで使用する各種ファイルのファイル名に使用可能な文字数は、OS のファイル名長制限(255 バイト)に準じる。

(2) 外部公開名として使用できない文字や文字列がある。詳細は「A.2.1 外部公開名に使用可能な文字」を参照。

(3) 外部公開名に使用可能な文字数に制限がある。詳細は「A.2.2 外部公開名に使用可能な文字数」を参照。

(4) プログラム実行に必要な内部データや局所記憶節に定義したデータ項目等スタック領域に配置されるデータがあり、プログラムの書き方により実行時にスタックオーバーフローが発生する場合がある。原因および対処の詳細は「A.2.3 スタックオーバーフロー発生時の対応」を参照。

また、スタックの大きさの上限は、アプリケーションが実行される実行環境(OS 等)のスタックの大きさの制約に準じる。

(5) 既存プログラムの利用者定義語が新規予約語に該当する場合、利用者定義語を修正すること。修正が不可能な場合、-U オプションにその利用者定義語を指定し、翻訳することで、利用者定義語として使用可能になる。

(6) 1 つの節や段落内に数千ステップの文を連続して記述している場合やデータ部の1つの節に数千個の VALUE 句指定のあるデータ項目を記述している場合に、翻訳時に-02 オプショ

ンを指定すると翻訳時間が極端に遅くなる場合がある。この場合の対処は「A.2.5 Visual C++からのエラー報告への対応」の「(1) Visual C++コンパイラのコンパイルスタックオーバーフローへの対応」を参照。

- (7) [Standard] COBOL GUI アプリケーションを作成するオプション(-Pg)を指定すると、EXTERNAL 句の記述されたデータ名の末尾が「\_PARAM」であるとき、データの参照となる。  
データの定義とするには、次のいずれかの方法で対処する。
- (a) EXTERNAL 句の記述されたデータ名の末尾が「\_PARAM」とならないように名前を変更する。
  - (b) GUI 部分とインタフェースをとらない COBOL プログラムにて、-Pg オプションを指定せず、-Sx オプションを指定する。
- (8) [Standard] EXTERNAL 句の記述されたデータを実行ファイル間で共有するためのオプション (-Sx と -Si) を指定したとき、または、COBOL GUI アプリケーションを作成するオプション (-Pg) を指定したとき、LNK4217 リンク警告エラーが発生する。  
この警告エラーは実行動作に影響しないため、無視してかまわないが、出力を抑止するには、-Wl:/IGNORE:4217 オプションを指定する。
- (9) [Standard] COBOL アプリケーション実行環境(プリンタ機能)使用时には、状態キー"91"および"97"は設定しない。

#### A.2.1 外部公開名に使用できない文字

外部公開名に使用可能な文字は、基本文字、基本数字および下線文字の組み合わせ文字からなり、かつ語頭は基本文字(A～Z, a～z)または下線文字でなければならない。

さらに、本コンパイラでは、AS 指定のない最も外側のプログラム名、AS 指定のないクラス名、AS 指定のない関数原型名、AS 指定のないメソッド名、AS 指定のないプログラム原型名、プロパティ名、AS 指定のない利用者定義関数名、AS 指定のない EXTERNAL 句の指定されたデータ名またはファイル名、利用者制御変数名、内部ファイル識別名のファイル識別リテラル、最も外側のプログラム名の AS 指定中、クラス名の AS 指定中、関数原型名の AS 指定中、メソッド名の AS 指定中、プログラム原型名の AS 指定中、利用者定義関数名の AS 指定中、REPOSITORY 段落の記述項の AS 指定中、EXTERNAL 句の AS 指定中、CALL 文中のプログラム名、CANCEL 文中のプログラム名、INVOKE 文中のメソッド名に、次の文字列は使用できない。

誤って使用した場合、アプリケーションが誤動作する可能性がある。COBOL が提供するシステムサブルーチンを使用する場合は使用可能である。

- A\_CBL で始まる文字列
- B\_で始まる文字列
- C\_で始まる文字列

- E\_CBR で始まる文字列
- -(ハイフン)を含む文字列
- CBL で始まる文字列
- SYS で始まる文字列
- CS で始まる文字列
- [Standard] \_PARAM で終わる文字列 (EXTERNAL 句が記述されたデータ名)
- 次の文字列

SIN\_FP, SPR\_FP

BASE, CVCsExitFlag, CVAPTYPE, CvhCobolThread, CVCobolThreadId,  
SSFUNTION, CWWINCOM, CWWINVARIABLE,

[Standard] TP\_send, TP\_receive

TP\_send, TP\_receive は, SE 動作モードでは, 使用できる。

C/C++コンパイラの予約語は使用しないでください。使用できない予約語の詳細は,  
C/C++コンパイラのドキュメントを参照。

[Enterprise for Windows] [Standard] C/C++コンパイラ:Visual C++

[Enterprise for Linux] C/C++コンパイラ:gcc(1)

また, プログラム原型や関数原型, プログラム定義, 関数定義およびクラス定義の外  
部公開名に使用可能な文字は, ファイル名に使用可能な文字に制限される。

#### A.2.2 外部公開名に使用可能な文字数

外部公開名に指定可能な文字の長さは最大 256 文字である。

ただし, プログラム原型や関数原型, プログラム定義, 関数定義およびクラス定義の外  
部公開名の文字数は, 外部リポジトリファイルのファイル名長に準じて制限される。

外部リポジトリファイルのファイル名は, 「種別を表すキーワード\_外部公開名.拡張子」  
で生成される。また, 外部公開名が英小文字の場合 1 文字に付きカレット(^)1 文字を付  
与する。これらの付加情報分の文字数が内部的に使用されるため, 使用可能な文字数  
の考慮が必要である。

(外部リポジトリファイルのファイル名の命名規則については, 「2.3 外部リポジトリのフ  
ァイル名生成規則」を参照。)

次に例を示す。

(例)カレントディレクトリのフルパスの長さが 200 バイトのプログラム定義の場合, ファイ  
ル名に使用できる文字数の算出方法

種別を表すキーワード ..... 5 バイト  
拡張子..... 4 バイト(.rpg)

255(Windows のファイル名長の最大長)

- 200(カレントディレクトリのパス長)
  - 5(種別を表すキーワード ” PROG\_ ” )
  - 4(拡張子 ” .rpg ” )
- = 46

外部公開名をすべて英大文字で指定する場合..... 46 バイト

外部公開名をすべて英小文字で指定する場合..... 23 バイト

### A.2.3 スタックオーバーフロー発生時の対応

アプリケーションの構成によっては、デフォルトの大きさ(\*)以上のスタックを使用する場合があります。その場合、スタックオーバーフローによりアプリケーションがアボートする。

(\*) [Enterprise for Windows] [Standard] 1MB

[Enterprise for Linux] 10MB(システム設定やユーザー設定により変更可能)

スタックオーバーフローによるアボートは、下記の様な構成のアプリケーションで発生する場合があります。

- プログラム呼び出しのネストが深い。
  - 1プログラムあたりの段落が多い。
  - 組込み関数の使用箇所が多い。
  - 連絡節の項目が多い。
  - 1プログラムあたりの局所記憶節の項目数や桁数が多い。
- (上記項目は、COBOL の原始プログラムの記述に依存して変わる。)

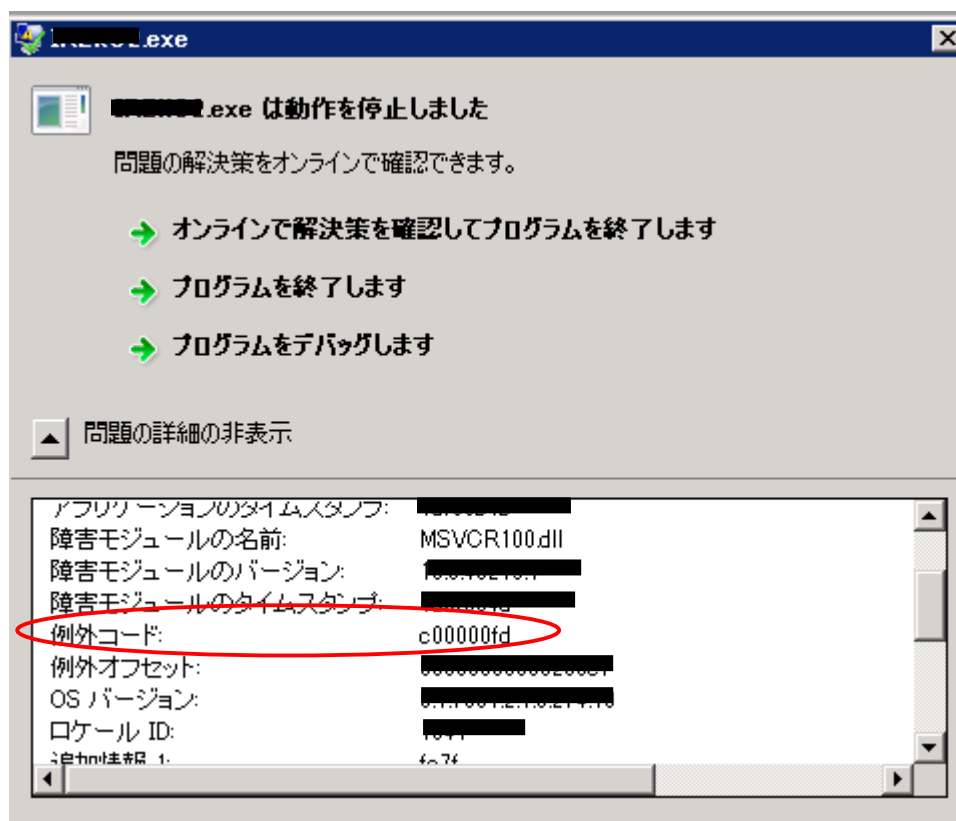


スタックオーバーフローが発生した際の判別方法と、対処方法を以降に記載する。

【Enterprise for Windows】【Standard】確認方法および対処方法は、以下の(1)(2)(3)を参照。

#### (1) スタックオーバーフロー発生の確認方法

アプリケーションがアボートした際に、アボートダイアログで問題の詳細を表示し例外コードを確認する。例外コードが「C00000fd」の場合、スタックオーバーフローが発生している。



## (2) 対処方法

### (a) 必要なスタックの大きさを算出する

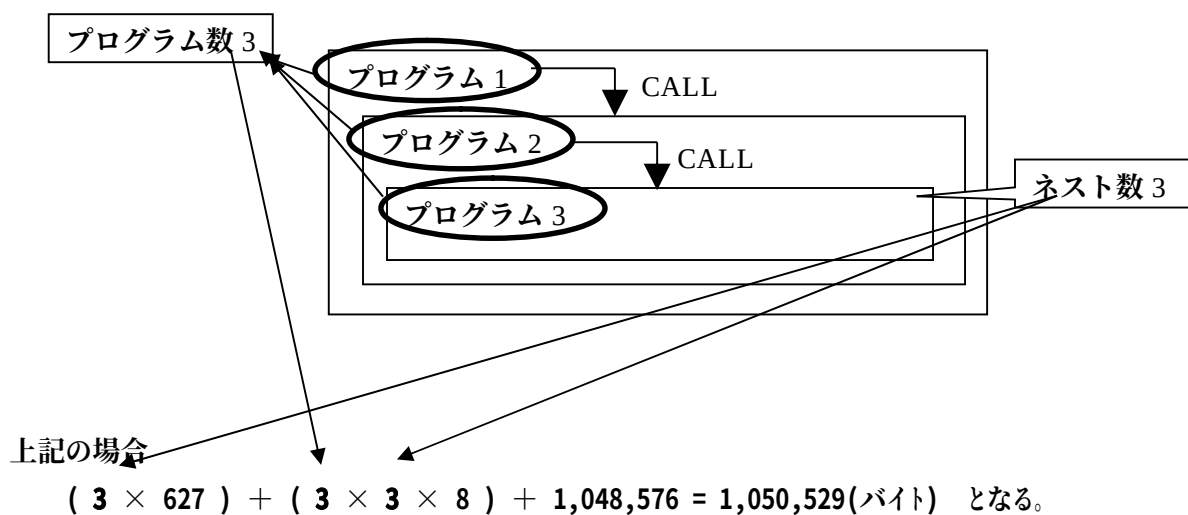
$(\text{プログラム呼び出しの最大ネスト数}(*1) \times 627) + (\text{プログラム数}(*2) \times \text{プログラム呼び出しの最大ネスト数} \times 8) + 1,048,576$

(\*1) プログラム呼び出しの最大ネスト数

関数 CALL の最も深い時の深さを指定する。

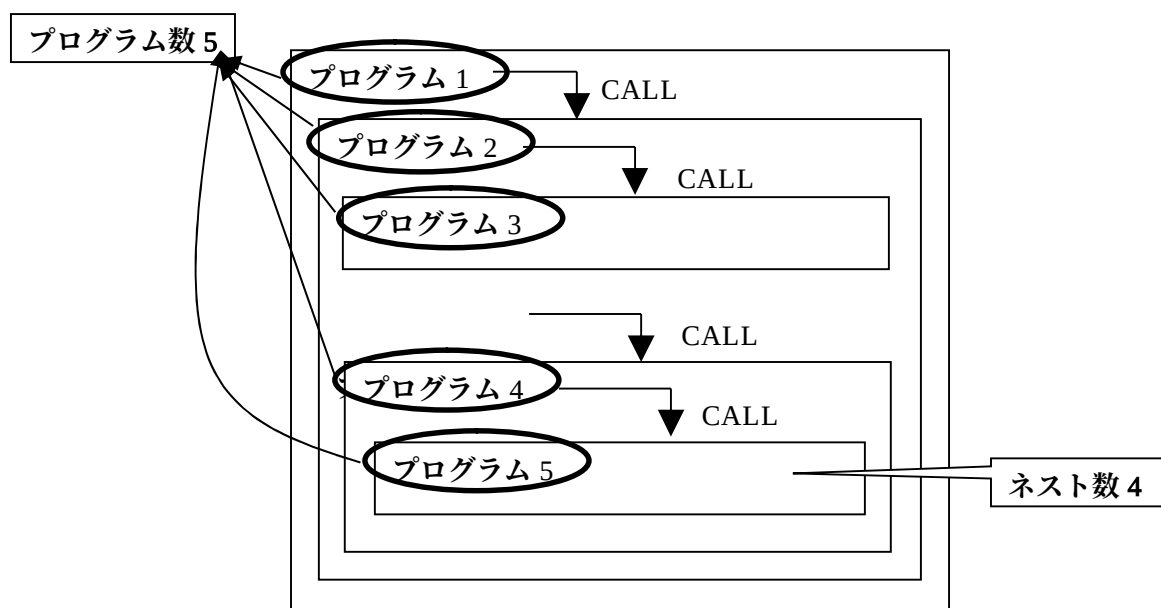
(\*2) プログラム数

最外側プログラム内の全プログラム数を指定する。



※参考例の為、数値が低くなっている。

最大ネスト数のため、次のような構成の場合は注意が必要である。



スタックの大きさの変更は、拡張子.exe のファイルに対して適用される。拡張子.dll ごとに  
変更することはできない。拡張子.dll を利用する拡張子.exe 単位に適用されるため、この  
ようなファイル構成のアプリケーションの場合、必要スタックの大きさは次のように算出する  
必要がある。

(例) ファイル構成が A.exe, B.dll, C.dll で構成されたアプリケーションの場合

A.exe に必要なスタックの大きさの算出結果……………600KB

B.dll に必要なスタックの大きさの算出結果……………500KB

C.dll に必要なスタックの大きさの算出結果……………500KB

必要スタックの大きさ: 600KB(A.exe) + 500KB(B.dll) + 500KB(C.dll) = 1.6MB

(b) スタックの大きさを変更する

- COBOL 翻訳時に指定する方法

COB コマンドの翻訳オプションに「-Wl:/STACK:xxxxxxx」(\*1)を指定する。

- 実行形式ファイルに直接指定する方法(要 VisualStudio 開発環境)  
下記コマンドを実行する。

```
> editbin /STACK:xxxxxxx yyyy.exe (*1)(*2)
```

(\*1) xxxxxx にはスタックの大きさを 10 進数で指定する。単位はバイト。

(\*2) yyyy には翻訳済みのモジュール名を指定する。

(3) 特定項目使用に対する補正

特定の項目が多いプログラムの場合、(2)(a)で算出した値でスタックの大きさを拡張しても現象が改善されない場合がある。その際は、下記式で算出できる値を加算して領域を確保する。

(各項目の計算範囲は、最外側プログラム内の全プログラム。)

- 1 プログラムあたりの段落が多い。

全段落数×4 バイト

- 組込み関数の使用箇所が多い。

組込み関数使用箇所×1024 バイト

- 連絡節の項目が多い。

(連絡節の項目数+1)×8 バイト

- 1 プログラムあたりの局所記憶節の項目数や桁数が多い。

データマップリストを出力し、LOCAL-STORAGE SECTION で定義したデータの LENGTH を合計したバイト数 + 1MB を、スタックの大きさに加算する。

データマップリストの出力方法は「第 15 章 翻訳リスト」を参照。

**注意**

・該当プログラムが再帰呼び出しの対象となっている場合は、上記で算出した値に再帰回数を乗算する事。

・再帰するプログラムが複数に渡る場合は、それぞれのプログラムのスタックの大きさも算出する事。

・ネストがある場合は、(2)(a)に記載した計算式も併用し大きさを算出する事。

・プロセス全体のスタックの大きさとして指定できる上限は 2GB となっています。マ

ルチスレッドプログラムの場合は、スレッド毎にスタック領域を確保する為、スレッドを多く生成するプログラムは注意が必要。

[Enterprise for Linux] 確認方法および対処方法は、以下の(4)(5)(6)を参照。

#### (4) スタックオーバーフロー発生の確認方法

スタックオーバーフローが発生した場合、アプリケーションは「セグメンテーション違反です」または「Segmentation fault」のメッセージを出力してアボートする。

(5)の対処方法に従い算出した必要なスタックの大きさが現在設定されているスタックの大きさを越えている場合には、(5)(b)に従い適切なスタックサイズへ変更する必要がある。

現在の設定値は、`ulimit -s` で確認可能である。

#### (5) 対処方法

##### (a) 必要なスタックの大きさを算出する

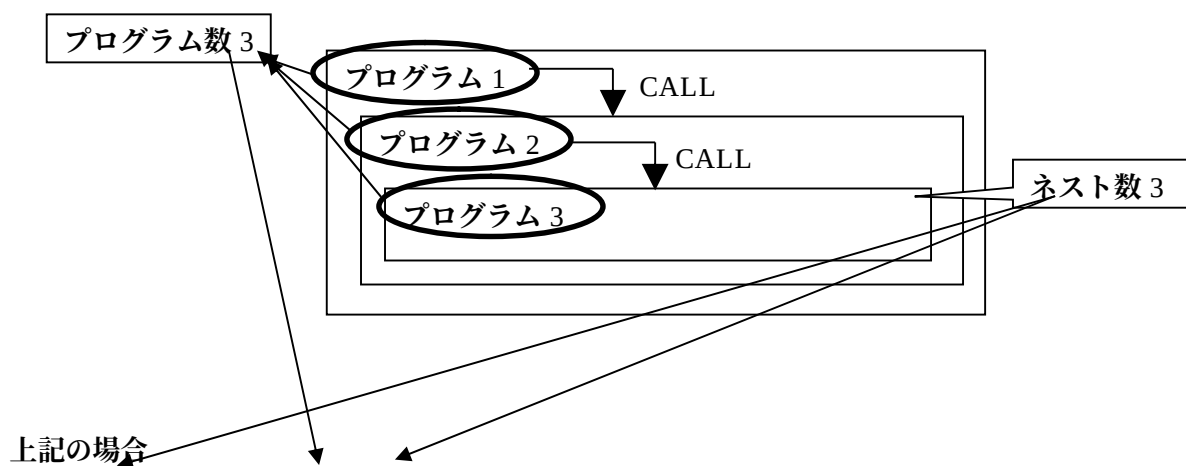
$(\text{プログラム呼び出しの最大ネスト数}(*1) \times 627) + (\text{プログラム数}(*2) \times \text{プログラム呼び出しの最大ネスト数} \times 8) + 1,048,576$

(\*1) プログラム呼び出しの最大ネスト数

関数 CALL の最も深い時の深さを指定する。

(\*2) プログラム数

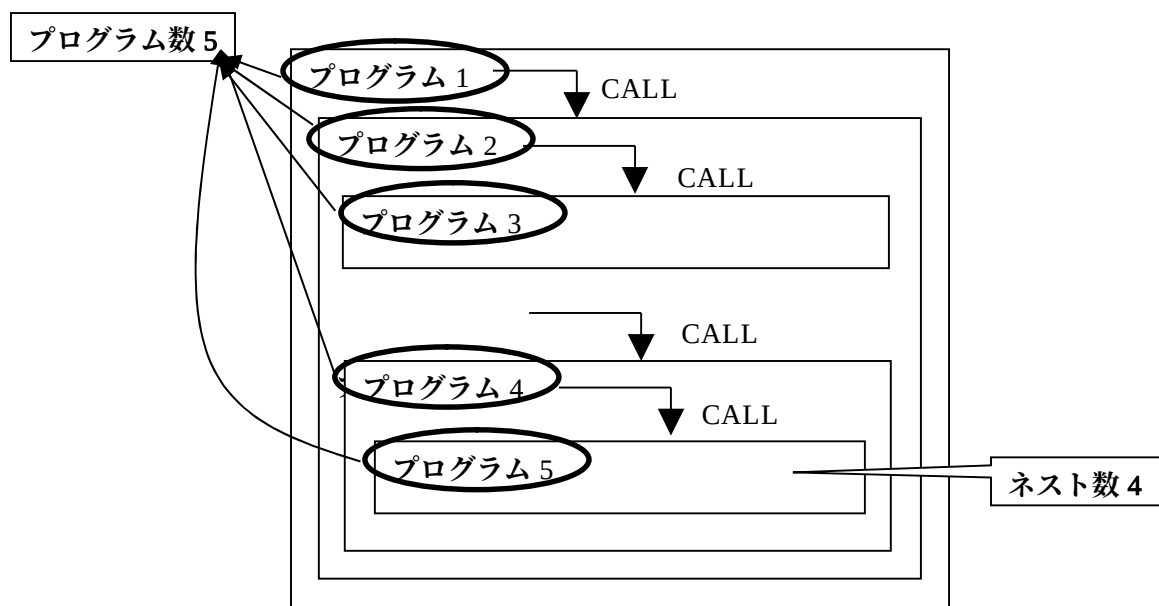
最外側プログラム内の全プログラム数を指定する。



$(3 \times 627) + (3 \times 3 \times 8) + 1,048,576 = 1,050,529(\text{バイト})$  となる。

※参考例の為、数値が低くなっている。

最大ネスト数のため、次のような構成の場合は注意が必要である。



スタックの大きさの変更は、プロセスに対して適用される。したがって、実行形式ファイルの他、共有ライブラリを利用するような構成のアプリケーションの場合、必要スタックの大きさは次のように共有ライブラリで使用するスタックの大きさも算出する必要がある。

(例) ファイル構成が A, B.so, C.so で構成されたアプリケーションの場合

|      |                        |       |
|------|------------------------|-------|
| A    | に必要なスタックの大きさの算出結果…………… | 600KB |
| B.so | に必要なスタックの大きさの算出結果…………… | 500KB |
| C.so | に必要なスタックの大きさの算出結果…………… | 500KB |

必要スタックの大きさ:  $600\text{KB}(A) + 500\text{KB}(B.so) + 500\text{KB}(C.so) = 1.6\text{MB}$

#### (b) スタックの大きさを変更する

Linux においてスタックの大きさを変更する場合、いくつかの方法があるが、本章では `ulimit` コマンドを使用する方法を示す。

`ulimit` は `sh (bash)` の内部コマンドである。スタックの大きさの変更方法の詳細については、Linux オペレーティングシステムのマニュアルやドキュメントを参照。

```
> ulimit -s xxxxx
```

(\*) xxxxxx はスタックサイズを指定する。単位は KB。

ulimit コマンドで設定したスタックサイズは、実行したシェルの子プロセスでのみ有効となる。ログアウトした後、ログインし直した際は、デフォルトの大きさに戻するため注意が必要である。

#### (6) 特定項目使用に対する補正

特定の項目が多いプログラムの場合、(5)(a)で算出した値でスタックの大きさを拡張しても現象が改善されない場合がある。その際は、下記式で算出できる値を加算して領域を確保する。

(各項目の計算範囲は、最外側プログラム内の全プログラム。)

- 1 プログラムあたりの段落が多い。  
全段落数  $\times$  4 バイト
- 組込み関数の使用箇所が多い。  
組込み関数使用箇所  $\times$  1024 バイト
- 連絡節の項目が多い。  
(連絡節の項目数+1)  $\times$  8 バイト
- 1 プログラムあたりの局所記憶節の項目数や桁数が多い。  
データマップリストを出力し、LOCAL-STORAGE SECTION で定義したデータの LENGTH を合計したバイト数 + 1MB を、スタックの大きさに加算する。  
データマップリストの出力方法は「第 15 章 翻訳リスト」を参照。

#### 注意

- ・該当プログラムが再帰呼び出しの対象となっている場合は、上記で算出した値に再帰回数を乗算する事。
- ・再帰するプログラムが複数に渡る場合は、それぞれのプログラムのスタックの大きさも算出する事。
- ・ネストがある場合は、(5)(a)に記載した計算式も併用し大きさを算出する事。

## A.2.4 実行ファイルの大きさの制限

[Enterprise for Windows] [Standard] 実行ファイルの大きさの制限は、次を参照。

Windows アプリケーションでは、実行ファイル (.exe, .dll) のフォーマットである PE (Portable Executable) の制約により、イメージの大きさが最大許容の大きさである 2GB を超える場合にリンク時エラー (LNK1248) が発生し、実行ファイルの生成に失敗する。

最大許容の大きさは 2GB である。

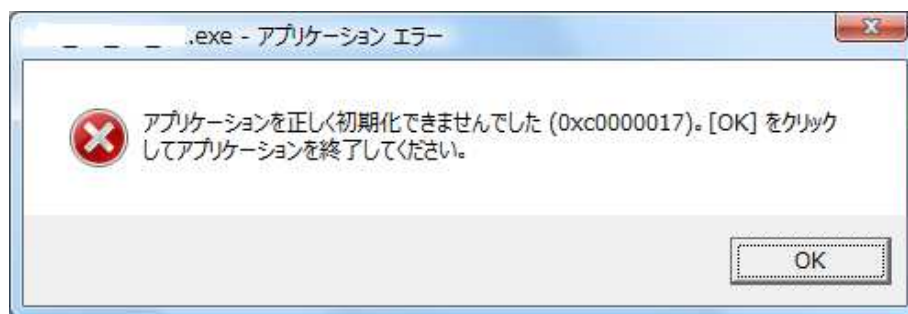
「イメージの大きさ」はプログラムコードとすべての静的データの合計であり、COBOL 記述では以下のものが該当する。実行単位を構成するすべてのプログラムの合計である。

- (1) プログラムコード
- (2) データ部 (data division) に記述したデータ群  
作業場所節, ファイル節, 連絡節, ベースデータ節のデータ項目
- (3) その他 本コンパイラが自動生成する領域  
プログラム制御用データ, ランタイム制御用データ, 定数データなどが実行ファイル内で静的データとして生成される。本データは COBOL の原始プログラムの記述に依存して大きさが変わる。

連絡節, ベースデータ節のデータ項目, および各節に定義した指標データ項目, データポインタデータ項目は, それぞれ 16 バイト, 8 バイトに拡張される。また, 作業場所節に定義した集団項目等において 8 バイト境界に合わせるため, 確保されるデータの大きさが増加する場合がある。COBOL85 で LNK1248 エラーとならず, 本コンパイラで LNK1248 エラーとなる場合は, これらデータの大きさの増加が原因の可能性あり。

DLL (ランタイム DLL を含む) をリンクしている場合, 翻訳 (リンク) 時ではなく, 実行時にイメージの大きさが計算され, 最大許容の大きさである 2GB を超える場合は以下のような実行時エラーが発生する。





リンク時エラー (LNK1248), または実行時エラーが発生した場合, COBOL 原始プログラムの記述を見直し, プログラムコードや定義データを削減する必要がある。

なお, 「7.5.2 外部項目による受け渡し」に記載されている「/FORCE」オプションを使用して, **EXTERNAL** 句指定データの受け渡しを行う場合, 本コンパイラにおいては受け渡し対象のデータがリンク時にそれぞれ固有のメモリに割り当てられるよう判断される。そのため, 「イメージの大きさ」にはそれぞれの受け渡し対象のデータ領域の合計が加算される。

※実行時は同一の領域を参照するため, 対象データは共有される。

【Enterprise for Linux】実行ファイルの大きさの制限は, 次を参照。

COBOL が生成する 64 ビットの実行ファイル (a.out, .so) は, Intel64 アーキテクチャのスモールモデルであるため, イメージの大きさ (コードとデータ) はアドレス空間が 2GB に制約される。イメージの大きさが最大許容の大きさである 2GB を超える場合, リンク時にエラーメッセージが表示され, 実行ファイルの生成に失敗する。

```
relocation truncated to fit: R_X86_64_32S against
collect2: ld returned 1 exit status
```

「イメージの大きさ」はプログラムコードとすべての静的データの合計であり, COBOL 記述では以下のものが該当する。実行単位を構成するすべてのプログラムの合計である。

- (1) プログラムコード
- (2) データ部 (data division) に記述したデータ群  
作業場所節, ファイル節, 連絡節, ベースデータ節のデータ項目
- (3) その他 本コンパイラが自動生成する領域  
プログラム制御用データ, ランタイム制御用データ, 定数データなどが実行ファイル内で静的データとして生成される。本データは COBOL の原始プログラムの記述に依存して大きさが変わ

る。

リンク時にエラーが発生した場合、COBOL 原始プログラムの記述を見直し、プログラムコードや定義データを削減する必要がある。

#### A.2.5 C++コンパイラからのエラー報告への対応

[Enterprise for Windows] [Standard] 本項目は、Windows 版でのみ有効である。

Visual C++の制約によりプログラムの書き方によって Visual C++コンパイラのコンパイルスタックオーバーフローや内部エラー (C1001) が Visual C++から報告される場合がある。

##### (1) Visual C++コンパイラのコンパイルスタックオーバーフローへの対応

1つの節や段落内に数万ステップの文を連続して記述している場合やデータ部の1つの節内に数万個の VALUE 句指定ありのデータ項目を記述している場合に、-02 オプションを指定すると Visual C++コンパイラの制約により Visual C++コンパイラで次のエラーが発生する場合がある。

～: fatal error

C1063: コンパイラの制限 :コンパイラ スタックのオーバー フローが発生しました。

～¥cl.exe での内部コンパイラ エラーです。エラー報告をマイクロソフトに送信

本メッセージが出力された場合の対処方法を以下に示す。

- 1つの節や段落内に数万ステップの文を連続して記述している場合、プログラムを細かく分割することや段落を細かく分割し PERFORM 文で実行するようにする。
- データ部の1つの節内に数万個の VALUE 句指定ありのデータ項目を記述している場合、他の節に記載することが可能なデータ項目を他の節へ記載することやデータ項目の初期化を MOVE 文等で行うことで1つの節内に記載する VALUE 指定ありデータ項目の個数を小さくする。

## (2) Visual C++コンパイラのコンパイラ内部エラー (C1001) への対応

CALL 文に多数の引数を指定した場合に -02 オプションを指定すると、Visual C++コンパイラの制約により Visual C++コンパイラで次のエラーが発生する場合があります。

~¥xxxxxxx.cpp(xxxx) : fatal error C1001: コンパイラで内部エラーが発生しました。 (コンパイラ ファイル 'msc1.cpp', 行 xxxx)  
この問題を回避するには、上記の場所付近のプログラムを単純化するか変更し

本メッセージが出力された場合には、何個かの引数を集団項目にまとめるなどにより CALL 文に指定する引数の個数を小さくすることが必要である。

## (3) Visual C++コンパイラのコンパイラ内部エラー (C1060) への対応

1つの手続き部内の手続き名の合計数(節名と段落名に COBOL コンパイラが内部生成したラベルを加えた数)が 1 万個を超える場合、Visual C++コンパイラの制約により「fatal error C1060: ヒープの領域を使い果たしました。」が発生する場合があります。

本メッセージが出力された場合には、次のいずれかの手段で回避する必要がある。

- ・仮想メモリ量を増やす
- ・プログラムを分割するなどにより手続き部内の手続き名の合計数を減らす

### A.2.6 COBOL が登録するシグナル

[Enterprise for Linux] 本項目は、Linux 版のみの注意事項である。

シグナルとは、ハードウェア、およびソフトウェアで発生する割り込みの事である。

COBOL がメインプログラムの場合、COBOL コンパイラが生成する実行可能プログラムは次のシグナルを登録する。登録には sigaction()を使用する。

SIGHUP/SIGINT/SIGKILL/SIGPIPE/SIGTERM/SIGUSR1/SIGUSR2/SIGPROF/SIGVTALRM/SIGSTKFLT/SIGIO/SIGPWR

COBOL が上記シグナルを登録した後で COBOL 以外のプログラムで上記のシグナルを登録した場合、異常終了時の結果は保証しない。

## A.2.7 オプションに使用できない文字

次のオプションを指定する場合は、使用できない文字があるため注意すること。

[Enterprise for Windows] -o オプション

[Standard] -o オプション, -Yf オプション, Yi オプション, -Yo オプション

| 文字<br>オプション | /<br>(*1) | ¥<br>(*1) | "<br>(*2) | :<br>(*3) | ;<br>(*4) | < | > | ? | * |   | 最後の<br>ピリオド |
|-------------|-----------|-----------|-----------|-----------|-----------|---|---|---|---|---|-------------|
| -o          | ○         | ○         | ○         | ○         | ×         | × | × | × | × | × | ×           |
| -Yf         | ○         | ○         | ○         | ○         | ○         | × | × | × | × | × | ×           |
| -Yi         | ○         | ○         | ○         | ○         | ○         | × | × | × | × | × | ×           |
| -Yo         | ○         | ○         | ○         | ○         | ○         | × | × | × | × | × | ×           |

[記号の意味] ○:使用可 ×:使用不可

- \*1 「/」, 「¥」 使用時は、おのこのの前に記述しているフォルダが存在していること  
(「/」, 「¥」 がルートを示す場合を除く)
- \*2 「:」は、ドライブ名の指定として使用すること
- \*3 「"」は、フォルダの前後に2つ記述すること
- \*4 [Standard] -q オプション指定時は、使用可能

## B 診断メッセージ

### B.1 原始プログラムの診断メッセージ

COBOL 文法規則に従って作成された原始プログラムを翻訳するとき、コンパイラが文法規則に違反している記述を検出すると、標準出力に診断メッセージを出力する。

【Enterprise for Linux】システム環境変数 `LANG=ja_JP.SJIS` でない場合、英文の診断メッセージを出力する。

次に診断メッセージ本文と原因を記述する。

記述する形式は次のとおりである。

| L        | FFNNN | 診断メッセージ本文 |
|----------|-------|-----------|
| 原因       |       |           |
|          | (1)   |           |
|          | (2)   |           |
| 埋め込み情報   |       |           |
|          | (1)   |           |
|          | (2)   |           |
| コンパイラの処理 |       |           |

【Enterprise for Windows】【Standard】なお、LNK で始まるエラー番号はリンカが出力するメッセージである。詳細は Visual C++ 2015, Visual C++ 2017, Visual C++ 2019 または Visual C++ 2022 のドキュメントを参照。

【Enterprise for Linux】なお、ld: で始まるメッセージはリンカが出力するメッセージである。詳細は gcc のドキュメントを参照。

## 説 明

**L** (エラーレベル)

診断メッセージのエラーレベルを示す。

|   |                 |                                                     |
|---|-----------------|-----------------------------------------------------|
| { | F (Fatal)       | 致命的エラー (オブジェクトは生成されない)                              |
|   | W (Warning)     | 警告エラー (オブジェクトは生成できる,<br>原因およびコンパイラの処理を参照すべきことを意味する) |
|   | O (Observation) | 注意                                                  |

**FFNNN** (診断メッセージ番号)

診断メッセージの番号を示す。

この付録 B は、この診断メッセージ番号順に記述している。

**S\*241**～**S\*252** の '**\***' には発生場所を表す数字が入ります。

0 (見出し部の誤り)

1 (環境部の誤り)

2 (データ部の誤り)

8 (手続き部の誤り)

**診断メッセージ本文**

診断メッセージの本文である。本文中の '**?**' は、ある語などが埋め込まれることを示している。埋め込まれる情報は、埋め込み情報に書かれている (本文中に複数 '**?**' が存在するときは、左から埋め込み情報 1, 埋め込み情報 2 という)。

W A0001      翻訳中の定義と外部リポジトリの内容に差分があります。  
              【Enterprise for Linux】DIFFERENCE EXISTS BETWEEN EXTERNAL REPOSITORY  
              AND CURRENT DEFINITION.

【原因】      翻訳中の定義と外部リポジトリの内容に差分がありました。  
【コンパイラの処理】 そのまま翻訳を続行する。

F C0001      ソース単位の構成が誤っています。  
              【Enterprise for Linux】THE OBLIGATORY DIVISION HEADER IS MISSING OR  
              INCORRECT.

【原因】      部や節やソース単位終了標が正しい順序でない。  
【コンパイラの処理】 構文解析等の以降の処理を一切行わない。

W C0003      必要な終止符がありません。  
              【Enterprise for Linux】THE TERMINATOR `PERIOD` IS MISSING.

【原因】      必要な終止符がない。  
【コンパイラの処理】 終止符を補う。

F C0004      見出し部の段落見出しがありません。  
              【Enterprise for Linux】THE PROGRAM-ID PARAGRAPH IS MISSING.

【原因】      見出し部の段落見出しがない。

F C0005      ソース単位名が誤っています。  
              【Enterprise for Linux】THE PROGRAM NAME IS INCORRECT.

【原因】      (1) 見出し部の段落見出し, またはソース単位終了標のソース単位名が利  
                  用者定義語でない。  
              (2) ソース単位終了標のソース単位名が, 見出し部の段落見出しで記述し  
                  たソース単位名と一致しない。  
【コンパイラの処理】 (1) の場合: 不正な文字列をそのままソース単位名とみなして処理を続  
                          行する。  
                          (2) の場合: 処理中のソース単位が終了したとみなして処理を続行す

る。

F C0006 後の行の指定が誤っています。  
【Enterprise for Linux】THE CONTINUATION LINE IS MISPLACED.

- 〔原因〕
- (1) ソース単位の始まりが、後の行である。
  - (2) 登録集原文の始まりが、後の行である。
  - (3) 注記項に固定継続標識がある。

W C0007 デバッグ行は指定できません。  
【Enterprise for Linux】THE DEBUG LINE IS FOLLOWED BY OBJECT-COMPUTER PARAGRAPH.

- 〔原因〕 実行用計算機 (OBJECT-COMPUTER) 段落より前に、デバッグ行がある。  
〔コンパイラの処理〕 注記行とみなす。

W C0008 文法的に正しくない文字があります。  
【Enterprise for Linux】THERE IS A CHARACTER WHICH IS NOT RIGHT GRAMMATICALLY.

- 〔原因〕
- (1) 文字列が分離符によって正しく区切られていない。
  - (2) 利用者定義語の文字列が英数字と日本語が混在している。
- 〔コンパイラの処理〕
- (1) 分離符があるものとみなす。
  - (2) 英数字と日本語の利用者定義語が複数指定されたとみなす。

F C0009 定数の終り方が誤っています。  
【Enterprise for Linux】THE SECOND QUOTE IS MISSING OR INCORRECT.

- 〔原因〕
- (1) 引用符の対応が、誤っている。
  - (2) 英数字定数またはブール定数の中身がない。
  - (3) 引用符 ( " ) とアポストロフィ ( ' ) の使い方が誤っている。
  - (4) 浮動定数継続標識の後の行に、定数が記述されていない。
  - (5) 登録集原文の終りに浮動定数継続標識が記述されている。

W C0010 後の行が A 領域から始まっています。  
【Enterprise for Linux】THE CONTINUATION LINE STARTS IN AREA A.



[原因] 後の行が, B 領域から始まっていない(A 領域から始まっている)。

[コンパイラの処理] B 領域から始まっているとみなす。

F C0011 後の行が全て空白です。

【Enterprise for Linux】THE CONTINUATION LINE IS BLANK.

[原因] 後の行がすべて空白である。

F C0012 標識領域に許されない文字があります。

【Enterprise for Linux】AN UNACCEPTABLE CHARACTER APPEARS IN COLUMN 7.

[原因] 標識領域に, 空白, 数字, 'ー', '\*' , '/' , 'D' 以外の文字がある。

F C0013 文字列の長さが?文字を超えています。

【Enterprise for Linux】THE STRING EXCEEDS ? CHARACTERS.

[原因] (1) 利用者定義語が 31 文字を超えている。

(2) PICTURE 句の文字列が 30 文字を超えている。

[埋め込み情報] (1) 31

(2) 30

F C0014 利用者定義語の最初または最後にハイフン(`-`)があります。

【Enterprise for Linux】THE USER DEFINED WORD IS PRECEDED BY OR FOLLOWED BY A HYPHEN.

[原因] 利用者定義語の最初／最後にハイフン(-)がある。

W C0015 利用者定義語に英字またはかな文字がありません。

【Enterprise for Linux】NO ALPHABET IS FOUND IN THE USER DEFINED WORD.

[原因] 手続き名以外の利用者定義語に, 英字またはかな文字が, 含まれていない。

[コンパイラの処理] 文字列をそのまま利用者定義語とみなして処理を行う。

F C0016      数字定数の長さが 18 を超えています。  
【Enterprise for Linux】THE NUMERIC LITERAL CONSISTS OF MORE THAN 18  
CHARACTERS.

〔原因〕      環境部, データ部の数字定数が 18 桁を超えている。

F C0017      許されない文字列があります。  
【Enterprise for Linux】AN UNACCEPTABLE CHARACTER STRING IS FOUND.

〔原因〕      利用者定義語, 数字定数, 英数字定数, 予約語, PICTURE 句の文字列の  
いずれにも属さない不正な文字列がある。

F C0019      英数字定数の長さが 256 を超えています。  
【Enterprise for Linux】THE NONNUMERIC LITERAL CONSISTS OF MORE THAN 256  
CHARACTERS.

〔原因〕      英数字定数の長さが 256 を超えている。

F C0020      16 進英数字定数が奇数個の文字になっています。  
【Enterprise for Linux】THE LITERAL CONSISTS OF ODD HEXADECIMAL  
CHARACTERS.

〔原因〕      16 進英数字定数が偶数個でない。

F C0021      英数字定数の内容が誤っています。  
【Enterprise for Linux】THE NONNUMERIC LITERAL CONTAINS INVALID  
DESCRIPTION.

〔原因〕      16 進英数字定数に指定された文字が, '0' ~ '9', 'A' ~ 'F' 以外の文字で  
ある。

F C0022      原文名が誤っています。  
【Enterprise for Linux】THE TEXT NAME IS INCORRECT.

〔原因〕      (1) COPY の原文名が, 利用者定義語でない。  
              (2) 日本語利用者定義語を指定した。

(3) 定数 1 指定および定数 2 指定がある。

F C0023 COPY 文が分離符の終止符(`. `)で終わっていません。  
【Enterprise for Linux】THE COPY STATEMENT IS NOT TERMINATED BY A PERIOD.

〔原因〕 (1) COPY 文が終止符で終わっていない。  
(2) SUPPRESS 指定がある。

〔コンパイラの処理〕 COPY 文がなかったものとみなす。

F C0025 原文名または登録集名が誤っています。  
【Enterprise for Linux】THE TEXT NAME OR THE LIBRARY NAME IS INCORRECT.

〔原因〕 (1) COPY 文に指定された原文が、登録集のファイルに登録されていない。  
または、翻訳時パラメータの登録集ファイルに対する指定を間違えた。  
(2) 指定された登録集名の登録集が存在しない。

F C0032 COPY 文の REPLACING 句が誤っています。  
【Enterprise for Linux】THE REPLACING CLAUSE CONTAINS INVALID DESCRIPTION.

〔原因〕 (1) REPLACING 句の作用対象がない。  
(2) REPLACING 句の作用対象が誤っている。  
(3) 必要語 BY がない。  
(4) 仮原文区切り記号 "==" が継続している。  
(5) 仮原文が正しく開始または終了していない。  
(6) 仮原文 1 が空または空白である。  
(7) 仮文字列が継続している。  
(8) 仮文字列中に句読文字が指定されている。  
(9) 仮文字列 1 が空または空白である。  
(10) LEADING 指定および TRAILING 指定がある。

W C0035 登録集名が誤っています。  
【Enterprise for Linux】THE LIBRARY NAME IS INCORRECT.

〔原因〕 (1) COPY の登録集名が、利用者定義語でない。  
(2) 日本語利用者定義語を指定した。

[コンパイラの処理] 登録集名の指定がなかったものとみなす。

F C0036 登録集原文に語`COPY`は指定できません。  
【Enterprise for Linux】A WORD `COPY` CANNOT BE SPECIFIED IN THE LIBRARY TEXT.

[原因] 登録集原文に COPY 文が含まれている。

F C0039 REPLACE 文の構文が誤っています。  
【Enterprise for Linux】THERE IS AN ILLEGAL SYNTAX IN THE REPLACE STATEMENT.

[原因] (1) 仮原文 1 または仮原文 2 の始まりの `==` がない。  
(2) REPLACE の直後に `OFF` がない。  
(3) 仮原文 1 と仮原文 2 の間に `BY` がない。  
(4) 仮原文 1 または仮原文 2 の終りの `==` が見つからないままソースの終りに達した。  
(5) 仮原文 1 が 1 つも原文語を含んでいない。または `;`, `,` のみから構成されている。  
(6) ALSO 指定および LAST 指定がある。  
(7) LEADING 指定および TRAILING 指定がある。

W C0041 REPLACE 文が終止符で終わっていません。  
【Enterprise for Linux】NO PERIOD FOLLOWS THE REPLACE STATEMENT.

[原因] 書き方 2 で `OFF` の後に 終止符がない。

[コンパイラの処理] 終止符があるものとみなす。

F C0042 REPLACE 文を記述する位置が誤っています。  
【Enterprise for Linux】THE POSITION THAT DESCRIBE REPLACE STATEMENT IS INVALID.

[原因] REPLACE 文を記述する位置が誤っている。

W C0046 ソース単位終了標が B 領域から書かれています。  
【Enterprise for Linux】THE END PROGRAM HEADER APPEARS IN AREA B.

[原因] 'END' が B 領域から書かれていて、その直後に 'PROGRAM' , 'CLASS' , 'FACTORY' , 'FUNCTION' , 'OBJECT' または 'METHOD' が書かれている。

[コンパイラの処理] A 領域から書かれたものとみなす。

F C0047 ソース単位終了標がありません。

**【Enterprise for Linux】THE END PROGRAM HEADER IS MISSING.**

[原因] (1) ソース単位名の記述が必要なソース単位終了標に、ソース単位名の記述がない。

(2) 入れ子機能を使用し、翻訳単位の終わりに達したのに、完結していないソース単位がある。

F C0048 含まれるプログラムに ? SECTION が指定されています。

**【Enterprise for Linux】THE ? SECTION CANNOT BE DESIGNATED IN THE PROGRAM CONTAINED WITHIN ANOTHER PROGRAM.**

[原因] 含まれるプログラムに構成節が指定されている。

[埋め込み情報] CONFIGURATION

F C0049 COMMUNICATION SECTION は 2 度以上指定できません。

**【Enterprise for Linux】THE COMMUNICATION SECTION CANNOT BE SPECIFIED MORE THAN TWICE.**

[原因] COMMUNICATION SECTION が 1 翻訳単位に 2 度以上指定されている。

F C0061 翻訳単位の始まりが登録集原文中にあります。

**【Enterprise for Linux】THE START AT EACH TRANSLATION IS IN THE LIBRARY TEXT.**

[原因] 翻訳単位の始まりが登録集原文中にあります。

[コンパイラの処理] 翻訳単位の始まり以降を無視する。

W C0110 外部行番号として無効な値が指定されています。

**【Enterprise for Linux】THE EXTERNAL LINE NUMBER IS INVALID.**

- [原因] (1) **-Ce** オプション指定でかつ外部行番号に数字以外が記述されている。  
(2) **-Ce** オプション指定でソース原文が固定形式かつ登録集原文が自由形式で記述されている。

[コンパイラの処理] 外部行番号に **999999** が指定されたとみなす。

F C0120 日本語定数に有効な文字列がありません。

**【Enterprise for Linux】THE SECOND QUOTE IS MISPLACED.**

- [原因] (1) 日本語定数を囲む引用符の間に文字列がない。

F C0121 日本語定数の文字列の長さが制限を超えています。

**【Enterprise for Linux】THE GRAPHIC LITERAL CONSISTS OF MORE THAN 128 CHARACTERS.**

- [原因] (1) 日本語 **16** 進定数の文字列の指定が **512** 文字を超えた。  
(2) 日本語コード定数の文字列の指定が **128** 文字を超えた。

F C0122      日本語定数の内容が誤っています。  
**【 Enterprise for Linux 】 THE GRAPHIC LITERAL CONTAINS INVALID DESCRIPTION.**

- [原 因]      (1) 日本語定数で指定された文字列が日本語として許されない文字列である (内部コード変換エラー)。  
 (2) 日本語 16 進定数が, 16 進 4 文字 1 組で指定されていない。  
 (3) 日本語 16 進定数に 0～9, A～Z 以外の文字が指定された。

F C0224      PICTURE 句の文字列による項目の大きさが 2147483647 を超えています。  
**【Enterprise for Linux】THE ITEM SIZE BEYOND 2147483647.**

- [原 因]      PICTURE 句の文字列による項目の大きさが 2147483647 を超えている。

F C0225      PICTURE 句の文字列による日本語項目の大きさが 1073741823 を超えています。  
**【Enterprise for Linux】THE GRAPHIC ITEM SIZE BEYOND 1073741823.**

- [原 因]      PICTURE 句の文字列による日本語項目の大きさが 1073741823 を超えている。

F C0226      PICTURE 句の文字列の繰り返し指定が誤っています。  
**【Enterprise for Linux】THE USE OF PARENTHESES IN THE PICTURE CHARACTER STRING IS ILLEGAL.**

- [原 因]      (1) PICTURE 句の文字列の最初がかっこである。  
 (2) PICTURE 句の文字列においてかっこが連続している。  
 (3) PICTURE 句の文字列において  
     (a) ‘N’ に続くかっこの中の数値が 1～31767 でない。  
     (b) ‘N’ 以外の文字に続くかっこの中の数値が 1～65535 でない。  
     (c) ‘N’ に続くかっこの中の数値が 1～1073741823 ではない。  
     (d) ‘N’ 以外の文字に続くかっこの中の数値が 1～2147483647 でない。  
 (4) PICTURE 文字 ‘.’, ‘CR’, ‘DB’, ‘S’, ‘V’ のかっこの繰り返し指定が 1 でない。  
 (5) PICTURE 句の文字列のかっこの対応が正しくない。

F C0227 PICTURE 句の文字列による項目の大きさが 65535 を超えています。

【Enterprise for Linux】THE ITEM SIZE BEYOND 65535.

〔原因〕 PICTURE 句の文字列による項目の大きさが 65535 を超えている。

F C0228 PICTURE 句の文字列による日本語項目の大きさが 32767 を超えています。

【Enterprise for Linux】THE GRAPHIC ITEM SIZE BEYOND 32767.

〔原因〕 PICTURE 句の文字列による日本語項目の大きさが 32767 を超えている。

F C0230 PICTURE 句の文字列に許されない文字があります。

【Enterprise for Linux】THE PICTURE CHARACTER STRING CONTAINS AN INVALID CHARACTER.

〔原因〕 PICTURE 句の文字列に許されない文字列がある。

F C0232 PICTURE 句の文字列が誤っています。

【Enterprise for Linux】THE PICTURE CHARACTER STRING IS ILLEGAL.

- 〔原因〕
- (1) PICTURE 句の文字列の中がすべて項目の大きさに数えない 'S', 'V' または 'P' からのみで構成されている。
  - (2) PICTURE 句の文字列中に 'A', 'N', 'X', '9', 'Z', '\*' が1つもないか, '+', '-', '¥' のいずれも2つ以上ない。
  - (3) PICTURE 文字の順序規則に違反している。

F C0233 数字または数字編集項目の数字位置が 18 を超えています。

【Enterprise for Linux】THE NUMBER OF DIGIT POSITIONS BEYOND 18.

〔原因〕 数字／数字編集項目の数字位置が 18 を超えている。

F C0234 数字編集項目の大きさが 31 を超えています。

【Enterprise for Linux】THE TOTAL LENGTH OF THE NUMERIC EDITED PICTURE CHARACTER STRING IS GREATER THAN 31.

〔原因〕 数字編集項目の大きさが 31 を超えている。



F C0235 英数字編集項目の大きさが 256 を超えています。  
【Enterprise for Linux】THE TOTAL LENGTH OF THE ALPHANUMERIC EDITED  
PICTURE CHARACTER STRING IS GREATER THAN 256.

〔原因〕 英数字編集項目の大きさが 256 を超えている。

F C0236 日本語編集項目の大きさが 128 を超えています。  
【Enterprise for Linux】THE TOTAL LENGTH OF THE GRAPHIC EDITED PICTURE  
CHARACTER STRING IS GREATER THAN 128.

〔原因〕 日本語編集項目の大きさが 128 を超えている。

F C0251 ブール定数の内容が誤っています。  
【Enterprise for Linux】THE BOOLEAN LITERAL CONTAINS INVALID  
DESCRIPTION.

〔原因〕 ブール定数の値に“0”または“1”以外の文字が含まれている。

F C0252 ブール定数の長さが 64 を超えています。  
【Enterprise for Linux】THE BOOLEAN LITERAL CONSISTS OF MORE THAN 64  
CHARACTERS.

〔原因〕 ブール定数の長さが 64 を超えている。

F C0253 ブールデータ項目の長さが 64 を超えています。  
【Enterprise for Linux】THE TOTAL LENGTH OF THE BOOLEAN DATA PICTURE  
CHARACTER STRING IS GREATER THAN 64.

〔原因〕 ブールデータ項目の長さが 64 を超えている。

W C0300 外部行番号が使用できないソース原文です。  
【Enterprise for Linux】EXTERNAL LINE NUMBER INADAPTABLE SOURCE PROGRAM.

〔原因〕 自由形式のソース原文に対して、-Ce オプションが指定されている。  
〔コンパイラの処理〕 -Ce オプションを無効とし内部行番号を使用する。

F C0301 見出し部の段落見出しが誤っています。  
【Enterprise for Linux】INCORRECT PARAGRAPH HEADER.

- [原因]
- (1) プログラム名段落  
プログラム定義中に、次のいずれかのソース単位が指定されている。
    - (a) クラス定義
    - (b) ファクトリ定義
    - (c) 関数定義
    - (d) 関数原型
    - (e) メソッド定義
    - (f) インスタンス定義
  - (2) 関数名段落  
関数定義または関数原型中に、次のいずれかのソース単位が指定されている。
    - (a) クラス定義
    - (b) ファクトリ定義
    - (c) 関数定義
    - (d) 関数原型
    - (e) メソッド定義
    - (f) インスタンス定義
    - (g) プログラム定義
    - (h) プログラム原型
  - (3) クラス名段落  
クラス定義中に、次のいずれかのソース単位が指定されている。
    - (a) クラス定義
    - (b) 関数定義
    - (c) 関数原型
    - (d) メソッド定義
    - (e) プログラム定義
    - (f) プログラム原型
  - (4) ファクトリ段落  
ファクトリ定義中に、次のいずれかのソース単位が指定されている。
    - (a) クラス定義
    - (b) ファクトリ定義
    - (c) 関数定義
    - (d) 関数原型

- (e) インスタンス定義
- (f) プログラム定義
- (g) プログラム原型
- (5) オブジェクト段落  
インスタンス定義中に、次のいずれかのソース単位が指定されている。
  - (a) クラス定義
  - (b) ファクトリ定義
  - (c) 関数定義
  - (d) 関数原型
  - (e) インスタンス定義
  - (f) プログラム定義
  - (g) プログラム原型
- (6) メソッド名段落  
メソッド定義中に、次のいずれかのソース単位が指定されている。
  - (a) クラス定義
  - (b) ファクトリ定義
  - (c) 関数定義
  - (d) 関数原型
  - (e) メソッド定義
  - (f) インスタンス定義
  - (g) プログラム定義
  - (h) プログラム原型
- (7) 最外側の見出し部の段落見出しに、次のいずれかのソース単位が指定されている。
  - (a) ファクトリ定義
  - (b) メソッド定義
  - (c) インスタンス定義

F C0302 クラス定義中に`?`は2度以上指定できません。

**【Enterprise for Linux】'?' DUPLICATES IN CLASS DEFINITION.**

- [原因] (1) クラス定義の中にファクトリ定義が2個以上記述されている。  
(2) クラス定義の中にインスタンス定義が2個以上記述されている。

- [コンパイラの処理] (1) FACTORY  
(2) OBJECT

F C0303 クラス定義中に指定したファクトリ定義とインスタンス定義の定義順が誤っています。

**【Enterprise for Linux】INCORRECT FACTORY-INSTANCE ORDER IN CLASS DEFINITION.**

〔原因〕 クラス定義中に指定した定義順がファクトリ定義、インスタンス定義でない。

F C0304 終了標が誤っています。

**【Enterprise for Linux】INCORRECT END PROGRAM HEADER.**

〔原因〕 見出し部の段落見出しの定義と終了標の定義が異なっている。

F C0305 クラス定義の書き方が誤っています。

**【Enterprise for Linux】INCORRECT CLASS DEFINITION.**

〔原因〕 クラス定義中に、データ部または手続き部が記載されている。

W C0306 行内注記の前に空白がありません。

**【Enterprise for Linux】NO SPACE APPEARS BEFORE INLINE COMMENT INDICATOR.**

〔原因〕 任意の語、定数、**PICTURE** 句の文字列または終止符と注記標識\*>の間に空白がない。

〔コンパイラの処理〕 \*>の前に空白があるものとして処理する。

F C0308 メソッド定義の書き方が誤っています。

**【Enterprise for Linux】INCORRECT METHOD DEFINITION.**

〔原因〕 ファクトリ定義またはインスタンス定義中のメソッド定義が、手続き部以外に記述されている。

W C0309 浮動デバッグ標識の直後に空白がありません。

**【Enterprise for Linux】SPACE LACKED AFTER FLOATING DEBUG INDICATOR.**

〔原因〕 浮動デバッグ標識の直後に空白がない。

〔コンパイラの処理〕 デバッグ標識の直後に空白があるものとみなして処理する。

F C0311 浮動定数継続標識の後に空白以外があります。

**【Enterprise for Linux】INCORRECT LETTER AFTER FLOATING CONTINUATION INDICATOR.**

〔原因〕 浮動定数継続標識の後に空白以外がある。

F C0312 浮動デバッグ標識の前に空白以外があります。  
**【Enterprise for Linux】SPACE LACKED BEFORE FLOATING DEBUG INDICATOR.**

〔原因〕 浮動デバッグ標識の前に空白以外がある。

F C0313 **【Standard】**画面節を記述したソース単位が、入れ子機能を使用しています。

〔原因〕 画面節を記述したソース単位は、入れ子機能を使用してはならない。

F C0317 **【Enterprise】**画面制御機能は使用できません。  
**【Enterprise for Linux】SCREEN SECTION CANNOT BE SPECIFIED.**

〔原因〕 データ部に画面節を指定している。

F D0001 内側プログラムのプログラム名段落には、?が指定できません。  
**【Enterprise for Linux】? CAN NOT ASSIGN FOR INTERNAL PROGRAM-ID PARAGRAPH.**

〔原因〕 (1) 内側プログラムのプログラム名段落に、AS 定数を指定している。

〔埋め込み情報〕 (1) AS

F D0002 プログラム名が重複しています。  
**【Enterprise for Linux】THE PROGRAM NAME IS DUPLICATED.**

〔原因〕 翻訳単位でプログラム名段落に記述したプログラム名が重複している。

F D0003 プログラム名段落に INITIAL 句を指定しているプログラムに、-Ci オプションが指定されています。  
**【Enterprise for Linux】THE -Ci COMPILE OPTION CAN NOT BE SPECIFIED FOR PROGRAM WITH INITIAL CLAUSE.**

〔原因〕 プログラム名段落に INITIAL 句があり、かつ-Ci オプションが指定されて

いる。

F D0004      **COMMON** 句の書き方が誤っています。  
【Enterprise for Linux】INCORRECT COMMON PHRASE.

〔原因〕      プログラム名段落の **COMMON** 句が最外側のプログラムに記述されている。

F D0005      **INITIAL** と **RECURSIVE** は同時に指定できません。  
【Enterprise for Linux】INITIAL AND RECURSIVE ARE NOT COMPATIBLE.

〔原因〕      プログラムを直接または間接に含んでいるプログラムが初期化プログラムである場合に **RECURSIVE** 句を指定しているまたはプログラムを直接または間接に含んでいるプログラムが再帰的プログラムである場合に **INITIAL** 句を指定している。

F D0006      ?はプログラム定義以外の要素に記述できません。  
【Enterprise for Linux】? IS DESCRIBED OUTSIDE IDENTIFICATION DIVISION.

〔原因〕      プログラム定義以外の要素に **AUTHOR**, **INSTALLATION**, **DATE-WRITTEN**, **DATE-COMPILED**, **SECURITY** が記述されている。

〔埋め込み情報〕      (1) **AUTHOR**  
                          (2) **INSTALLATION**  
                          (3) **DATE-WRITTEN**  
                          (4) **DATE-COMPILED**  
                          (5) **SECURITY**

F D0007      プログラム原型あるいは関数原型が他のソース単位の後方にあります。  
【Enterprise for Linux】FORWARD REFERENCE OF PROGRAM OR FUNCTION PROTOTYPE.

〔原因〕      ある翻訳群中で、関数原型とプログラム原型はその他すべての種類のソース単位より先行されていない。

W D0008      メインプログラムに?句が指定されています。  
【Enterprise for Linux】? PHRASE SHOULD NOT BE SPECIFIED IN MAIN PROGRAM.

〔原因〕      メインプログラムに **RECURSIVE** 句または **AS** 句が指定されている。

[埋め込み情報] (1) RECURSIVE

(2) AS

[コンパイラの処理] RECURSIVE 句または AS 句なしとして処理続行します。

F D0009 外部公開名は英字または下線で始まり、英字、数字または下線からなる英数字定数でなければなりません。

**【Enterprise for Linux】EXTERNALIZED NAME MUST BE ALPHANUMERIC CONSTANT STARTED BY ALPHABET OR UNDERSCORE AND CONSISTS OF ALPHABET/NUMERIC/UNDERSCORE.**

[原因] AS 句の定数が次のいずれかである。

(1) 英字 A～Z, 数字 0～9, 下線でない文字が含まれている。

(2) 語頭が、英字および下線でない。

F D0010 同じプロパティメソッドは既に定義されています。

**【Enterprise for Linux】THE SAME PROPERTY METHOD IS ALREADY DEFINED.**

[原因] SET 指定および GET 指定が同一のプロパティメソッドが、クラス内のファクトリ定義またはインスタンス定義で既に定義されている。

F D0011 メソッドは既に定義されています。

**【Enterprise for Linux】PROPERTY METHOD IS ALREADY DEFINED.**

[原因] メソッド名または外部公開名が同一のメソッドがクラス内で既に定義されている。

F D0013 翻訳中のクラスと同じ名前が、継承元クラス名に指定されています。

**【Enterprise for Linux】INHERITANCE OF THE SAME NAMED CLASS.**

[原因] 翻訳中のクラスと同じ名前が、継承元クラス名に指定されている。

F D0014 標準クラスのメソッド`?`はオーバーライドできません。

**【Enterprise for Linux】STANDARD CLASS ? CAN NOT BE OVERRIDE.**

[原因] 標準クラスが持つメソッドと同じ名前のメソッドを定義しようとした。

[埋め込み情報] (1) New

(2) FactoryObject

- (3) CSRelease
- (4) CSReleaseInThread
- (5) CSReleaseAll

F D0015     `?'が外部公開名として許されない利用者定義語です.  
【Enterprise for Linux】USER DEFINED WORD ? CAN NOT BE USED FOR  
EXTERNALIZED NAME.

[原因]            外部公開名として許されない利用者定義語である。  
[埋め込み情報]    利用者定義語

W D0016     必要な終止符がありません.  
【Enterprise for Linux】THE TERMINATOR `PERIOD` IS MISSING.

[原因]            必要な終止符がない。  
[コンパイラの処理] 終止符があるものとみなす。

W D0018     IDENTIFICATION の後に DIVISION がありません.  
【Enterprise for Linux】"DIVISION" IS MISSING AFTER "IDENTIFICATION".

[原因]            IDENTIFICATION の後に DIVISION がない。  
[コンパイラの処理] 書いてあるものとみなす。

W D0019     `?'の記述が誤っています.  
【Enterprise for Linux】WRONG DESCRIPTION OF ?.

[原因]            AUTHOR, INSTALLATION, DATE-WRITTEN, DATE-COMPILED, SECURITY の  
記述順序が誤っているか, 重複記述されている。

- [埋め込み情報]    (1) AUTHOR  
                  (2) INSTALLATION  
                  (3) DATE-WRITTEN  
                  (4) DATE-COMPILED  
                  (5) SECURITY

[コンパイラの処理] 記述がないものとみなす。

F D0020     見出し部の段落見出しが誤っています.  
【Enterprise for Linux】INCORRECT PARAGRAPH HEADER.



- [原因] (1) プログラム名段落
- (a) プログラム定義中に、プログラム原型が指定されている。
  - (b) プログラム原型中に、プログラム定義またはプログラム原型が指定されている。

W D1001 区分番号が 1 から 49 の整数ではありません。  
【Enterprise for Linux】THE SEGMENT NUMBER IS OUTSIDE THE RANGE (1-49).

- [原因] 区分番号が 1 から 49 の整数でない。  
[コンパイラの処理] 区分番号を 1 とする。

F D1002 ?句が二重に指定されています。  
【Enterprise for Linux】THE ? CLAUSE IS DUPLICATED.

- [原因] 句が二重に指定されている。  
[埋め込み情報] (1) ORGANIZATION  
(2) RENAME  
(3) ACCESS MODE  
(4) RECORD KEY  
(5) FILE STATUS

F D1003 符号系名または記号文字`?`は既に定義されています。  
【Enterprise for Linux】THE ALPHABET NAME OR THE SYMBOLIC CHARACTERS NAME  
`?` IS DUPLICATED.

- [原因] 符号系名, 記号文字が二重に指定されている。  
[埋め込み情報] (1) 符号系名  
(2) 記号文字

F D1004 CURRENCY SIGN 句の定数が 1 桁ではありません。  
【Enterprise for Linux】THE LITERAL DEFINED IN THE CURRENCY SIGN CLAUSE  
IS NOT ONE CHARACTER.

- [原因] CURRENCY SIGN 句の定数が 1 桁でない。

**F D1005**      **CURRENCY SIGN** 句の定数に許されない文字があります.  
**【Enterprise for Linux】AN UNACCEPTABLE CHARACTER IS DEFINED FOR THE LITERAL IN THE CURRENCY SIGN CLAUSE.**

[原 因]      **CURRENCY SIGN** 句の定数に許されない文字がある。  
 次に示す文字を除く 1 桁の英数字定数でなければならない。

- (1) 数字 0 から 9 まで
- (2) 英字 A, B, C, D, E, N, P, R, S, V, X, Z, もしくはこれらの英小文字, または空白
- (3) 特殊文字
  - + 正号
  - 負号
  - \* 星印
  - / 斜線
  - = 等号
  - , コンマ
  - ; セミコロン
  - : コロン
  - . 終止符または小数点
  - " 引用符
  - ' 引用符 (アポストロフィ)
  - ( 左かっこ
  - ) 右かっこ

**F D1006**      ファイル名 `?` は既に定義されています.  
**【Enterprise for Linux】THE FILE NAME `?` IS DUPLICATED.**

[原 因]      同一ファイル名が **SELECT** 句で既に定義されている。  
 [埋め込み情報] ファイル名

**F D1007**      ファイル識別リテラルの長さが 8 桁 を超えています.  
**【Enterprise for Linux】THE FILE IDENTIFIER EXCEEDS 8 CHARACTERS.**

[原 因]      **SELECT** 句で指定された内部ファイル名の長さが 8 文字を超えている。

**F D1008**      装置名が指定されていません。

**【Enterprise for Linux】NO DEVICE TYPE IS DESIGNATED IN THE SELECT CLAUSE.**

〔原因〕 装置名または識別名が指定されていません。

F D1009 ファイル識別リテラルに許されない文字があります。  
**【Enterprise for Linux】THE FILE IDENTIFIER CONTAINS AN UNACCEPTABLE CHARACTER.**

〔原因〕 ファイル識別リテラルに許されない文字（英数字以外）が使用されている。

F D1010 ファイル識別リテラルの最初が英字ではありません。  
**【Enterprise for Linux】THE FILE IDENTIFIER DOES NOT BEGIN WITH AN ALPHANUMERIC CHARACTER.**

〔原因〕 ファイル識別リテラルの最初が英字ではありません。

F D1013 **【Standard】SELECT BASED** 句は指定できません。

〔原因〕 **SELECT BASED** 句が指定されている。

F D1014 **【Standard】**装置の付加情報の長さが 8 を超えています。

〔原因〕 装置の付加情報の長さが 8 を超えている。

F D1015 **【Standard】**装置の付加情報に許されない文字があります。

〔原因〕 装置の付加情報に許されない文字がある。

F D1016 ? 句が二重に指定されています。  
**【Enterprise for Linux】THE `?'` CLAUSE IS DUPLICATED.**

〔原因〕 句が二重に指定されている。

〔埋め込み情報〕 **RELATIVE KEY**

F D1017 ? 句で参照されたファイル名 `?'` は定義されていません。

**【Enterprise for Linux】IN ? CLAUSE, THE FILE NAME `?' IS NOT DEFINED.**

**[原因]** SELECT 句で定義されていないファイル名が, SAME 句, APPLY EXCLUSIVE-MODE 句, APPLY SHARED-MODE 句, APPLY SHIFT-CODE 句および MULTIPLE 句で参照されている。

**[埋め込み情報]** 埋め込み情報 1

(1) SELECT

(2) SAME

(3) APPLY

(4) MULTIPLE FILE

埋め込み情報 2

対応する未定義のファイル名

**F D1018** SAME AREA 句でファイル名 `?' が二重に指定されています。

**【Enterprise for Linux】THE FILE NAME `?' REFERENCED BY SAME AREA CLAUSE IS DUPLICATED.**

**[原因]** 既に SAME AREA 句で参照されているファイル名が再び同じ句で参照されている。

**[埋め込み情報]** ファイル名

**F D1019** SAME RECORD AREA 句でファイル名 `?' が二重に指定されています。

**【Enterprise for Linux】THE FILE NAME `?' REFERENCED BY SAME RECORD AREA CLAUSE IS DUPLICATED.**

**[原因]** 既に SAME RECORD AREA で参照されているファイル名が, 再び SAME RECORD AREA 句で参照されている。

**[埋め込み情報]** ファイル名

**F D1020** APPLY 句でファイル名 `?' が二重に指定されています。

**【Enterprise for Linux】THE FILE NAME `?' REFERENCED BY APPLY CLAUSE IS DUPLICATED.**

**[原因]** 既に APPLY 句で参照されているファイル名が再び同じ句で参照されている。

**[埋め込み情報]** ファイル名

W D1021      **【Standard】**RDB ファイルに付加情報が指定されています。

[原因]            RDB ファイルにデバイスアドレス, フォーム名, またはメディア識別名が指定されている。

[コンパイラの処理]    付加情報は指定されていないものと見なす。

F D1022      **【Standard】**メディア識別名の長さが 16 を超えています。

[原因]            メディア識別名の長さが 16 を超えた。

F D1023      識別名の指定に許されない文字があります。

**【Enterprise for Linux】**THE IDENTIFICATION CONTAINS AN UNACCEPTABLE CHARACTER.

[原因]            装置名または識別名に英数字とハイフン以外がある。

W D1024      既に識別名は指定されています。無視します。

**【Enterprise for Linux】**IDENTIFIER IS ALREADY SPECIFIED. THIS CLAUSE IS IGNORED.

[原因]            識別名を複数指定した。

[コンパイラの処理]    最初の識別名を有効とする。

F D1025      `?`は 1 つのプログラムで 1 つだけしか指定できません。

**【Enterprise for Linux】**THIS SELECTED FILE IS REDUNDANTLY ASSIGNED TO `?`.

[原因]            SPR ファイルをユーザが明に複数指定した。

[埋め込み情報]      SPR

W D1026      必要な 終止符がありません。

**【Enterprise for Linux】**THE TERMINATOR `PERIOD` IS MISSING.

[原因]            文の終止符がない。

[コンパイラの処理]    終止符があるものとみなす。

F D1027      必要語`?`がありません。  
              【Enterprise for Linux】THE OBLIGATORY RESERVED WORD `?` IS MISSING.

〔原因〕      必要な予約語がない。

〔埋め込み情報〕 (1) DIVISION  
                  (2) SECTION  
                  (3) MODE  
                  (4) DEBUGGING  
                  (5) STATUS  
                  (6) ON  
                  (7) COMMA  
                  (8) DUPLICATES

F D1028      【Standard】プリンタファイルのフォーム名の長さが 123 を超えています。

〔原因〕      PRT 装置タイプでフォーム名指定の定数長が 123 を超えている。

F D1029      ?段落がありません。  
              【Enterprise for Linux】THE ? PARAGRAPH IS MISSING.

〔原因〕      段落がない。

〔埋め込み情報〕 (1) SOURCE-COMPUTER  
                  (2) OBJECT-COMPUTER  
                  (3) FILE-CONTROL

F D1030      装置名または識別名の指定が誤っています。  
              【Enterprise for Linux】THE IDENTIFIER OR DEVICE NAME CONTAINS INVALID  
DESCRIPTION.

〔原因〕      (1) ファイル識別リテラル指定で、指定できない装置名または識別名を指  
定している。  
              (2) 【Standard】装置名 PRT にデバイスアドレスが指定されている。

F D1031      ?段落の書き方が誤っています。  
              【Enterprise for Linux】THE ? PARAGRAPH CONTAINS INVALID DESCRIPTION.

[原因] 段落の書き方が誤っている。

[埋め込み情報] (1) SOURCE-COMPUTER  
(2) OBJECT-COMPUTER  
(3) I-O-CONTROL

F D1032      **【Standard】**入れ子の内側プログラム, または再帰的プログラム中に RDB 順ファイル, RDB 相対ファイル, または RDB 索引ファイルは指定できません。

[原因] (1) 入れ子の内側プログラム中に RDB 順ファイル, RDB 相対ファイル, または RDB 索引ファイルが指定されている。  
(2) 再帰的プログラム中に RDB 順ファイル, RDB 相対ファイル, または RDB 索引ファイルが指定されている。

F D1033      ?句に必要な定数がありません。  
**【Enterprise for Linux】**NO OBLIGATORY LITERAL IS FOUND IN THE ? CLAUSE.

[原因] 句に定数が指定されていない。

[埋め込み情報] (1) MEMORY SIZE  
(2) RESERVE  
(3) CURRENCY SIGN  
(4) POSITION

F D1034      ?句の書き方が誤っています。  
**【Enterprise for Linux】**THE ? CLAUSE CONTAINS INVALID DESCRIPTION.

[原因] (1) 句の書き方が誤っている。  
(2) 索引ファイルにおいて LOCK MODE 句と APPLY 句を同時に指定した。  
(3) APPLY SHIFT-CODE 句に PPR-CONTROL-1 および PPR-CONTROL-2 以外が指定された。

[埋め込み情報] (1) MEMORY SIZE  
(2) ACCESS MODE  
(3) FILE STATUS  
(4) ASSIGN  
(5) ORGANIZATION  
(6) PROGRAM COLLATING SEQUENCE

- (7) ALTERNATE RECORD
- (8) MULTIPLE FILE
- (9) CLASS
- (10) LOCK MODE
- (11) APPLY
- (12) PADDING CHARACTER
- (13) RECORD DELIMITER
- (14) SELECT

F D1036      RECORD 句または RELATIVE KEY 句の書き方が誤っています。  
【Enterprise for Linux】THE RECORD CLAUSE OR THE RELATIVE KEY CLAUSE  
CONTAINS INCORRECT DESCRIPTION.

[原因]      (1) RECORD/RELATIVE KEY 句の書き方が誤っている。

F D1037      【Standard】-Cj, -CU の何れかのオプション指定時, 装置タイプ RDB, PRT またはフ  
ォーム指定の装置タイプ PRN は指定できません。

[原因]      (1) -Cj, -CU のいずれかのオプション指定で翻訳したソース中に, 装置タ  
イプ RDB が指定されている。  
(2) -Cj, -CU のいずれかのオプション指定で翻訳したソース中に, 装置タ  
イプ PRT が指定されている。  
(3) -Cj, -CU のいずれかのオプション指定で翻訳したソース中に, フォー  
ムが指定されている装置タイプ PRN が指定されている。

F D1038      【Standard】入れ子の内側プログラムまたは, 再帰的プログラム中に装置タイプ  
PRT または, フォーム指定の装置タイプ PRN は指定できません。

[原因]      (1) 入れ子の内側プログラム中に, 装置タイプ PRT が指定されている。  
(2) 入れ子の内側プログラム中に, フォームが指定されている装置タイプ  
PRN が指定されている。  
(3) 再帰的プログラム中に, 装置タイプ PRT が指定されている。  
(4) 再帰的プログラム中に, フォームが指定されている装置タイプ PRN が  
指定されている。

F D1039      【Standard】装置タイプ PRT は, 定数指定できません。



〔原因〕 (1) 装置タイプ PRT が定数指定されている。

F D1040 必要な条件名がありません。  
【Enterprise for Linux】NO CONDITION NAME IS DEFINED IN THE SWITCH STATUS  
CONDITION.

〔原因〕 スイッチ条件名指定に条件名が指定されていない。

F D1041 必要な呼び名がありません。  
【Enterprise for Linux】NO MNEMONIC NAME IS DEFINED.

〔原因〕 (1) 機能名の呼び名がない。  
(2) スイッチ名の呼び名がない

F D1042 スイッチ名指定の書き方が誤っています。  
【Enterprise for Linux】THE SWITCH NAME CLAUSE CONTAINS INVALID  
DESCRIPTION.

〔原因〕 (1) SWITCH-n で n が 1 から 8 の範囲にない。  
(2) SWITCH-n に関連付ける呼び名, ON 句または OFF 句がない。

F D1043 符号系名指定の書き方が誤っています。  
【Enterprise for Linux】THE ALPHABET NAME CLAUSE CONTAINS INVALID  
DESCRIPTION.

〔原因〕 符号系名の指定が誤っている。

F D1044 機能名指定の書き方が誤っています。  
【Enterprise for Linux】THE FUNCTION NAME CLAUSE CONTAINS INVALID  
DESCRIPTION.

〔原因〕 (1) 機能名が次のものでない。  
(a) CONSOLE  
(b) STDIN

(c) STDOUT

(d) STDERR

(2) 機能名指定の記述場所が誤っている。

F D1045 制御変数指示語の書き方が誤っています。  
【Enterprise for Linux】THE CONTROL VARIABLE CONTAINS INVALID DESCRIPTION.

〔原因〕 ITGmmm または CHARnnn で mmm または nnn の範囲が誤っている。

F D1046 CURRENCY SIGN 句の書き方が誤っています。  
【Enterprise for Linux】THE CURRENCY SIGN CLAUSE CONTAINS INVALID DESCRIPTION.

〔原因〕 CURRENCY SIGN 句の書き方が誤っている。

F D1047 DECIMAL-POINT 句の書き方が誤っています。  
【Enterprise for Linux】THE DECIMAL-POINT CLAUSE CONTAINS INVALID DESCRIPTION.

〔原因〕 DECIMAL-POINT 句の書き方が誤っている。

F D1048 呼び名は英字または下線で始まる、英字、数字または下線からなる 8 桁までの文字でなければなりません。  
【Enterprise for Linux】CHARACTER SET OF MNEMONIC NAME MUST BE ALPHABET, NUMERIC OR `\_`, AND LESS THAN 9 CHARACTERS, FIRST OF WHICH MUST BE ALPHABET OR `\_`.

〔原因〕 (1) 呼び名が次の条件に該当する。  
(a) 英字で始まっていない。  
(b) 英字、数字または下線でない文字が含まれている。  
(c) 8 桁より大きい文字である。  
(d) 制御変数指示語、機能名またはスイッチ名である。

W D1049 語 `ALPHABET` を省略したものとみなします。  
【Enterprise for Linux】OMITTED A WORD `ALPHABET` IS ASSUMED.

【原因】 ALPHABET 句の指定で、語 'ALPHABET' を省略している。

【コンパイラの処理】 ALPHABET 句がないものとみなす。

F D1050 SELECT 句のファイル名がありません。

【Enterprise for Linux】NO FILE NAME IS DEFINED IN THE SELECT PHRASE.

【原因】 SELECT 句で指定されたファイル名がない。

F D1051 ASSIGN 句の装置指定がありません。

【Enterprise for Linux】NO DEVICE NAME IS DEFINED IN THE ASSIGN PHRASE.

【原因】 ASSIGN 句の装置指定がない。

F D1054 OF または IN の次がデータ名ではありません。

【Enterprise for Linux】THE WORD FOLLOWING THE QUALIFIER 'OF' OR 'IN' IS NOT A DATA NAME.

【原因】 修飾のための OF または IN の次に、データ名が指定されていない。

W D1055 FILE STATUS 句にデータ名 2 が指定されています。

【Enterprise for Linux】DATA-NAME-2 IS DEFINED IN THE FILE STATUS CLAUSE.

【原因】 このファイルには、FILE STATUS 句にデータ名 2 を指定できない。

【コンパイラの処理】 データ名 2 がないものとみなす。

F D1056 FILE STATUS 句にデータ名がありません。

【Enterprise for Linux】NO DATA NAME IS DEFINED IN THE FILE STATUS CLAUSE.

【原因】 FILE STATUS 句にデータ名がない。

F D1058 SAME 句にファイルが 1 つだけ指定されています。

【Enterprise for Linux】ONLY ONE FILE IS SPECIFIED IN THE SAME CLAUSE.

【原因】 (1) SAME AREA 句に、ファイル名が 2 個以上、指定されていない。

(2) SAME RECORD AREA 句に、ファイル名が 2 個以上、指定されていない。

(3) **SAME SORT AREA** 句にファイル名が 2 個以上指定されていない。

F D1059      **APPLY** 句にファイル名がありません。  
【Enterprise for Linux】NO FILE NAME IS SPECIFIED IN THE APPLY CLAUSE.

〔原因〕      **APPLY** 句にファイル名が指定されていない。

F D1063      **ADDRESS-FORMAT** 句のアドレス形式が誤っています。  
【Enterprise for Linux】INVALID ADDRESS FORMAT.

〔原因〕      **ADDRESS-FORMAT** 句に存在しないアドレス形式を指定した。

F D1064      数字定数の値が 2147483647 を超えています。  
【Enterprise for Linux】THE VALUE OF NUMERIC LITERAL EXCEEDS 2147483647.

〔原因〕      数字定数の値が 2147483647 を超えている。

F D1071      整数に符号は書けません。  
【Enterprise for Linux】THE INTEGER HAS A SIGN.

〔原因〕      整数に符号は書けない。

F D1072      定数として 0 (ゼロ) は指定できません。  
【Enterprise for Linux】A ZERO VALUE IS SPECIFIED FOR THE NUMERIC LITERAL.

〔原因〕      定数として 0 は指定できない。

F D1082      符号系名または字類名 `?' の定数句で同じ文字に対して 2 回以上の指定があります。  
THE CHARACTER REFERENCED BY THE ALPHABET NAME OR THE CLASS NAME `?' IS  
DUPLICATED.

〔原因〕      符号系名または字類名の定数句で同一文字に対して、2 回以上の指定がある。

〔埋め込み情報〕 (1) 符号系名  
                  (2) 字類名

F D1083 符号系名句, 字類名句または記号文字の数字定数または整数の値が 1 から 256 の範囲ではありません。

【Enterprise for Linux】THE NUMERIC LITERAL IN THE ALPHABET CLAUSE , THE CLASS CLAUSE OR THE SYMBOLIC CLAUSE IS OUTSIDE THE RANGE OF 1-256.

[原因] 符号系名句または字類名句の数字定数または整数の値が 1 から 256 の範囲でない。

F D1084 ALSO 句または THROUGH/THRU 句の指定の英数字定数が 1 桁ではありません。

【Enterprise for Linux】THE ALPHANUMERIC LITERAL IN THE ALSO CLAUSE OR THE THROUGH/THRU CLAUSE IS NOT ONE CHARACTER.

[原因] ALSO または THROUGH 句の指定の英数字定数が 1 桁でない。

F D1085 符号系名`?`は定義されていません。

【Enterprise for Linux】THE ALPHABET NAME `?` IS UNDEFINED.

[原因] 符号系名が定義されていない。

[埋め込み情報] 符号系名

F D1089 MULTIPLE FILE 句で POSITION 句の指定が誤っています。

【Enterprise for Linux】THE POSITION PHRASE IN THE MULTIPLE FILE CLAUSE CONTAINS INVALID DESCRIPTION.

[原因] MULTIPLE FILE 句で POSITION 句の指定が誤っている。

F D1090 MULTIPLE FILE 句の POSITION 句で指定された整数の値が 1 から 254 の範囲ではありません。

【Enterprise for Linux】THE INTEGER IN THE POSITION PHRASE IS OUTSIDE THE RANGE (1-254).

[原因] MULTIPLE FILE 句の POSITION 句で指定された整数の値が 1 以上 254 以下でない。

F D1092 ファイル名`?`は既に MULTIPLE FILE 句で指定されています。

**【Enterprise for Linux】THE FILE NAME `?' REFERENCED BY THE MULTIPLE FILE  
CLAUSE IS DUPLICATED.**

[原因] 既に **MULTIPLE FILE** 句で参照されているファイル名が再び同じ句で参照されている。

[埋め込み情報] ファイル名

**F D1094**     **SAME SORT AREA** 句でファイル名 `?' が二重に指定されています。  
**【Enterprise for Linux】THE FILE NAME `?' REFERENCED BY THE SAME SORT AREA  
CLAUSE IS DUPLICATED.**

[原因] **SAME SORT AREA** 句でファイル名が二重に指定されている。

[埋め込み情報] ファイル名

**F D1095**     **MULTIPLE FILE** 句で指定されたファイルの数が 254 を超えています。  
**【Enterprise for Linux】THE NUMBER OF FILES IN THE MULTIPLE FILE CLAUSE  
IS MORE THAN 254.**

[原因] **MULTIPLE FILE** 句で指定されたファイルの数が 254 を超えた。

**F D1097**     **PROGRAM COLLATING SEQUENCE** 句で指定されている符号系名の文字照合順序コードが誤っています。  
**【Enterprise for Linux】THE ALPHABET NAME IN THE PROGRAM COLLATING SEQUENCE  
CLAUSE IS INCORRECT.**

[原因] **PROGRAM COLLATING SEQUENCE** で指定された符号系名で定義されている文字照合順序コードが **PROGRAM COLLATING SEQUENCE** で指定できないコードである。

**F D1098**     **【Standard】**副レコードキーの数が 7 を超えています。

[原因] **【Standard】**副レコードキーの数が 7 を超えている。

**F D1099**     **【Standard】**副レコードキーの指定が誤っています。

[原因] **RDB 索引ファイルの ALTERNATE RECORD KEY** 句にデータ名が 16 を超え

て指定されている。

**F D1100**      ファイル名 `?' は APPLY SHIFT-CODE 句で指定できません。  
                 **【Enterprise for Linux】**THE FILE NAME `?' CANNOT BE SPECIFIED IN THE APPLY  
                 SHIFT-CODE CLAUSE.

〔原因〕            APPLY SHIFT-CODE 句で指定されているファイルの装置種別がプリンタ以外  
                 である。

〔埋め込み情報〕 ファイル名

〔コンパイラの処理〕 APPLY SHIFT-CODE 句の指定を無視する。

**W D1102**      ファイル識別リテラルは英字で始まる英字、数字、下線からなる利用者定義語で  
                 なければなりません。  
                 **【Enterprise for Linux】**CHARACTER SET OF FILE IDENTIFIER MUST BE  
                 ALPHABET, NUMERIC OR `\_', FIRST OF WHICH MUST BE ALPHABET.

〔原因〕            ファイル識別リテラルとして使用できない文字がある。

〔コンパイラの処理〕 指定誤りとして翻訳を続ける。

**W D1103**      これ以降のファイル識別リテラルまたはデータ名を無視します。  
                 **【Enterprise for Linux】**FILE IDENTIFIER OR DATA NAME IS IGNORED.

〔原因〕            2 個以上の内部ファイル名を指定した。

〔コンパイラの処理〕 2 個目以降の指定を無視して翻訳を続ける。

**W D1106**      SPECIAL-NAMES 段落に記述した句の順序が誤っています。  
                 **【Enterprise for Linux】**THE ORDER OF DESCRIPTION IN THE SPECIAL-NAMES  
                 PARAGRAPH IS WRONG.

〔原因〕            SPECIAL-NAMES 段落に記述した句の順序が誤っている。

〔コンパイラの処理〕 正しいものとみなす。

**W D1108**      無効となるファイル識別リテラルが指定されています。  
                 **【Enterprise for Linux】**MEANINGLESS FILE IDENTIFIER ARE SPECIFIED.

〔原因〕            無効となるファイル識別リテラルが指定されている。

[コンパイラの処理] 2 個目以降のファイル識別リテラルの指定を無視する。

F D1112 データ名または実ファイル名指定で装置名は省略できません。  
【Enterprise for Linux】DEVICE TYPE CANNOT BE OMITTED BY DATA NAME OR  
EXTERNAL FILE NAME.

[原因] データ名または実ファイル名指定で装置名を省略した。

F D1114 装置名または識別名の指定に矛盾があります。  
【Enterprise for Linux】THE IDENTIFIER OR DEVICE NAME CONFLICTS.

[原因] 装置名または識別名の指定が前の装置名または識別名の指定と矛盾している。

F D1221 定数は整数でなければなりません。  
【Enterprise for Linux】THE NUMERIC LITERAL INCLUDES A DECIMAL POINT.

[原因] 定数が整数でない。

F D1222 数字定数が 32767 を超えています。  
【Enterprise for Linux】THE VALUE OF NUMERIC LITERAL BEYOND THE LIMIT  
(32767).

[原因] 数字定数が 32767 を超えている。

F D1240 本機能は指定できません。  
【Enterprise for Linux】THIS FUNCTION CANNOT BE SPECIFIED.

[原因] (1) 【Enterprise】使用できない APPLY PROTECTED-MODE を指定している。  
(2) 【Enterprise】WITH RETURN VALUE を指定している。

F D1301 ?定義中に?は指定できません。  
【Enterprise for Linux】IN ? DEFINITION, ? CAN NOT BE SPECIFIED.

[原因] 次のいずれかの場合。  
(1) メソッド定義中に構成節が指定された。



- (2) ファクトリ定義およびインスタンス定義中に翻訳用計算機段落, 実行用計算機段落, リポジトリ段落が指定された。
- (3) クラス定義直下, メソッド定義中に入出力節が指定された。
- (4) **【Standard】**クラス定義中または関数定義中に **WITH RETURN VALUE** 句が指定された。

[埋め込み情報] 埋め込み情報 1

- (1) **METHOD**
- (2) **FACTORY**  
**OBJECT**
- (3) **CLASS**  
**METHOD**
- (4) **【Standard】CLASS**  
**FUNCTION**

埋め込み情報 2

- (1) **CONFIGURATION SECTION**
- (2) **SOURCE-COMPUTER**  
**OBJECT-COMPUTER**  
**REPOSITORY**
- (3) **INPUT-OUTPUT SECTION**
- (4) **【Standard】WITH RETURN VLAUE**

**F D1304**     ?原型中に?は指定できません.  
**【Enterprise for Linux】IN ? PROTOTYPE, ? CAN NOT BE SPECIFIED.**

[原因]       次のいずれかの場合。

- (1) プログラム原型中の環境部に実行用計算機段落または入出力節があった。
- (2) 関数原型中の環境部に実行用計算機段落または入出力節があった。

[埋め込み情報] 埋め込み情報 1

- (1) **PROGRAM**
- (2) **FUNCTION**

埋め込み情報 2

- (1) **OBJECT-COMPUTER**  
**INPUT-OUTPUT SECTION**
- (2) **OBJECT-COMPUTER**

## INPUT-OUTPUT SECTION

**F D1305**     ?原型中には **CURRENCY** 句, **DECIMAL-POINT** 句のみ指定可能です.  
**【Enterprise for Linux】ONLY CURRENCY OR DECIMAL-POINT PHRASE CAN BE SPECIFIED IN ? PROTOTYPE.**

[原因]        次のいずれかの場合。  
               (1) プログラム原型中の特殊名段落に **CURRENCY** 句, **DECIMAL-POINT** 句以外が記述された。  
               (2) 関数原型中の特殊名段落に **CURRENCY** 句, **DECIMAL-POINT** 句以外が記述された。

[埋め込み情報]    (1) **PROGRAM**  
                     (2) **FUNCTION**

**F D1306**     `?'が外部公開名として許されない利用者定義語です.  
**【Enterprise for Linux】USER DEFINED WORD ? IS CAN NOT BE USE FOR EXTERNALIZED NAME.**

[原因]        外部公開名として許されない利用者定義語である。

[埋め込み情報]    外部公開名

**F D1307**     外部公開名は英字または下線で始まり, 英字, 数字または下線からなる英数字定数でなければなりません.  
**【Enterprise for Linux】EXTERNALIZED NAME MUST BE ALPHANUMERIC CONSTANT STARTED BY ALPHABET OR UNDERSCORE AND CONSISTS OF ALPHABET/NUMERIC/UNDERSCORE.**

[原因]        **AS** 句の定数が次のいずれかである。  
               (1) 英字 **A～Z**, 数字 **0～9**, 下線でない文字が含まれている。  
               (2) 語頭が, 英字および下線でない。

**F D1308**     ?定義中の特殊名段落に?句は指定できません.  
**【Enterprise for Linux】IN SPECIAL-NAMES PARAGRAPH OF ? DEFINITION, ? PHRASE CAN NOT BE SPECIFIED.**

[原因]        ファクトリ定義, インスタンス定義中の特殊名段落に指定できない句を記載

した。

[埋め込み情報] 埋め込み情報 1

(1) FACTORY

(2) OBJECT

埋め込み情報 2

(1) ALPHABET

(2) SYMBOLIC

(3) CLASS

(4) CURRENCY SIGN

(5) DECIMAL-POINT

F D1309 ?定義中の特殊名段落にスイッチ名指定は指定できません。

【Enterprise for Linux】THE SWITCH NAME CAN NOT BE SPECIFIED IN  
SPECIAL-NAMES PARAGRAPH OF ? DEFINITION.

[原因] ファクトリ定義, インスタンス定義中の特殊名段落にスイッチ名指定を記載した。

[埋め込み情報] (1) FACTORY

(2) OBJECT

F D1310 ?定義中の特殊名段落に機能名指定は指定できません。

【Enterprise for Linux】THE FUNCTION NAME CAN NOT BE SPECIFIED IN  
SPECIAL-NAMES PARAGRAPH OF ? DEFINITION.

[原因] ファクトリ定義, インスタンス定義中の特殊名段落に機能名指定を記載した。

[埋め込み情報] (1) FACTORY

(2) OBJECT

F D1311 ?定義中の特殊名段落に装置名指定は指定できません。

【Enterprise for Linux】THE DEVICE NAME CAN NOT BE SPECIFIED IN  
SPECIAL-NAMES PARAGRAPH OF ? DEFINITION.

[原因] ファクトリ定義, インスタンス定義中の特殊名段落に装置名指定を記載した。

[埋め込み情報] (1) FACTORY

## (2) OBJECT

F D1312      ?定義中の特殊名段落に制御変数指示語は指定できません。  
              【Enterprise for Linux】THE CONTROL VARIABLE CAN NOT BE SPECIFIED IN  
              SPECIAL-NAMES PARAGRAPH OF ? DEFINITION.

【原因】          ファクトリ定義, インスタンス定義中の特殊名段落に制御変数指示語を記載した。

【埋め込み情報】    (1) FACTORY  
                      (2) OBJECT

F D1313      CSV 形式の行順ファイルが指定されているプログラムに, -CU オプションが指定されています。  
              【Enterprise for Linux】THE -CU COMPILE OPTION CAN NOT BE SPECIFIED FOR  
              PROGRAM USING CSV TYPE LINE-SEQUENTIAL FILE.

【原因】          CSV 形式の行順ファイルが指定されているプログラムを-CU オプション指定で翻訳しようとしている。

F D1314      【Standard】メインプログラムに?は指定できません。

【原因】          メインプログラム中に指定できない句が指定された。

【埋め込み情報】    (1) WITH RETURN VLAUE

F D1315      【Standard】入れ子の内側プログラムまたは, 再帰的プログラム中に?は指定できません。

【原因】          入れ子の内側プログラムまたは, 再帰的プログラム中に指定できない句が指定された。

【埋め込み情報】    (1) WITH RETURN VLAUE

F D1316      【Standard】-CG, -CH の何れかのオプション指定時, ?は指定できません。

【原因】          -CG, -CH のいずれかのオプション指定で翻訳したソース中に指定できない句が指定された。

【埋め込み情報】    (1) WITH RETURN VLAUE

F D1317      **【Enterprise】**再帰的プログラム中に RAF ファイルは指定できません。  
**【Enterprise for Linux】**RAF FILE CANNOT BE DESIGNATED IN THE PROGRAM  
CONTAINED WITHIN RECURSIVE PROGRAM.

〔原 因〕      再帰的プログラム中に RAF ファイルが指定されている。

F D1318      **【Enterprise】**識別名 RAF は、プログラム定義中にしか指定できません。  
**【Enterprise for Linux】**RAF FILE CAN SPECIFY ONLY PROGRAM DEFINITION.

〔原 因〕      識別名 RAF がプログラム定義以外で指定されている。

F D1319      **【Enterprise】**-CG, -CH, -Cj, -CU の何れかのオプション指定時、識別名 RAF は  
指定できません。  
**【Enterprise for Linux】**THE -CG, -CH, -Cj, -CU COMPILE OPTION CAN NOT BE  
SPECIFIED WITH RAF FILE.

〔原 因〕      -CG, -CH, -Cj, -CU のいずれかのオプション指定で翻訳したソース中に、識  
別名 RAF が指定されている。

F D1320      **【Enterprise】【Standard】**-Cj, -CU の何れかのオプション指定時、相対編成また  
は索引編成のファイルは使用できません。  
**【Enterprise for Linux】**THE -Cj, -CU COMPILE OPTION CAN NOT BE SPECIFIED  
WITH RELATIVE FILE OR INDEXED FILE.

〔原 因〕      **【Enterprise】【Standard】**-Cj, -CU のいずれかのオプション指定で翻訳し  
たソース中で、相対編成または索引編成のファイルを使用している。  
ただし、Standard Edition 版は対象外である。

F D1321      **【Enterprise】**RAF ファイルとリポジトリ段落は同時に記述できません。  
**【Enterprise for Linux】**RAF FILE AND REPOSITORY PARAGRAPH NOT COMPATIBLE.

〔原 因〕      RAF ファイルがリポジトリ段落と同時に記述されている。

F D1322      **【Enterprise】**-MT オプション指定時、相対編成または索引編成のファイルは使用  
できません。

**【Enterprise for Linux】WHEN -MT COMPILE OPTION IS SPECIFIED, RELATIVE FILES OR INDEX FILE CANNOT BE USED.**

[原 因]        -MT オプション指定で翻訳したソース中で、相対編成または索引編成のファイルを使用している。

F D2001        データ名が誤っています。  
**【Enterprise for Linux】THE DATA NAME IS INCORRECT.**

[原 因]        利用者定義語が不正である。

F D2002        **【Enterprise for Linux】**用語名または属性名が用語辞書に登録されていません。  
**【Enterprise for Linux】THE TERM NAME OR ATTRIBUTE NAME IS NOT REGISTERED IN THE TERM DICTIONARY.**

[原 因]        用語名または属性名が未登録である。

F D2003        **【Enterprise for Linux】**用語値名が用語データ項目の用語情報に登録されていません。  
**【Enterprise for Linux】THE TERM VALUE NAME IS NOT REGISTERSD IN THE TERM INFORMATION OF TERM DATA ITEM.**

[原 因]        用語値名が未登録である。

F D2004        **【Enterprise for Linux】**用語辞書の内容が誤っています(?)。  
**【Enterprise for Linux】THE CONTENT OF TERM DICTIONARY IS INVALID(?).**

[原 因]        用語が参照できない。

[埋め込み情報] (1)    1 (ログインできない)  
                  (2)    2 (データベースエラー)  
                  (3)    3 (メモリエラー)  
                  (4)    4 (用語辞書がない)  
                  (5)    5 (入力引数エラー)  
                  (6)    6 (用語辞書管理テーブルがない)  
                  (7)    7 (用語辞書管理テーブルは不正な値が格納されている)

- (8) 13(既定データソースが存在しない)
- (9) 14(その他エラー)
- (10) 15(用語値の長さエラー)
- (11) 16(用語値ネストオーバー)
- (12) 17(包含用語値の個数オーバー)
- (13) 18(用語値の属性種別エラー)
- (14) 19(PICTURE 情報の長さエラー)
- (15) 20(属性情報ネストオーバー)
- (16) 21(属性情報エラー)

**F D2005**      **【Standard】RDB ファイルに対するファイル記述項に?句は指定できません.**

- [原因]            (1) RDB ファイルに対するファイル記述項に **EXTERNAL** 句が指定されている。
- (2) RDB ファイルに対するファイル記述項に **GLOBAL** 句が指定されている。
- (3) RDB ファイルに対するファイル記述項に **RECORD VARYING** 句が指定されている。

- [埋め込み情報] (1) **EXTERNAL**
- (2) **GLOBAL**
- (3) **RECORD VARYING**

**W D2006**      ?互換:データ項目の大きさが異なります.

**【Enterprise for Linux】COMPATIBILITY WITH ? : IT IS NOT COMPATIBLE IN SIZE.**

[原因]            **BINARY** データ項目の大きさが異なります。

[埋め込み情報] **ACOS-4**

[コンパイラの処理] **ACOS-4** の **BINARY** データ項目の大きさとする。

**F D2007**      ?が外部公開名として許されない利用者定義語です.

**【Enterprise for Linux】`?` CANNOT BE SPECIFIED FOR EXTERNAL NAME.**

[原因]            外部公開名として許されない利用者定義語である。

[埋め込み情報] 利用者定義語

**W D2008**      **COMPAT=COMP3** 互換:この内部 10 進項目の符号部分は割り付けません.

**【Enterprise for Linux】FOR COMPAT=COMP3 COMPATIBILITY: SIGN FIELD OF THIS PACKED-DECIMAL ITEM IS NOT ALLOCATED.**

[原因] 内部 10 進項目の符号部分は割り付けない。

[コンパイラの処理] 符号部分なしの内部 10 進項目とする。

F D2009 COMPAT=COMP3 互換:符号のない内部 10 進項目の桁数(`P`を含まない)は偶数でなければなりません。

**【Enterprise for Linux】FOR COMPAT=COMP3 COMPATIBILITY: SIZE OF UNSIGNED PACKED-DECIMAL ITEM (NOT INCLUDING `P`) MUST BE AN EVEN NUMBER.**

[原因] 符号のない内部 10 進項目の桁数(`P`を含まない)は偶数でなければならない。

F D2010 CHARACTER TYPE 句が指定された基本項目は日本語項目または日本語編集項目でなければなりません。

**【Enterprise for Linux】THE ITEM WITH CHARACTER TYPE CLAUSE OR ITS SUBORDINATE ITEM MUST BE THE GRAPHIC(-EDITED) ITEM.**

[原因] CHARACTER TYPE 句が指定された基本項目は、日本語項目または日本語編集項目でなければならない。

F D2011 `TYPE`が誤っています。

**【Enterprise for Linux】THE `TYPE` IS INVALID.**

[原因] CHARACTER TYPE 句の`TYPE`がないか正しくない。

F D2012 書体名がないかまたは正しくありません。

**【Enterprise for Linux】THE CHARACTER TYPE IS INVALID.**

[原因] 書体名がないかまたは正しくない。

F D2013 従属項目の CHARACTER TYPE 句がそれを含む集団項目の CHARACTER TYPE 句と矛盾しています。

**【Enterprise for Linux】THE CHARACTER TYPE CLAUSE OF THE SUBORDINATE ITEM IS INCONSISTENT WITH THE CHARACTER TYPE CLAUSE OF THE GROUP ITEM.**



[原因] 集団項目に指定した書体名と従属項目にしていた書体名が矛盾している。

F D2014 COLUMN NUMBER 句は集団項目, OCCURS 句を指定した項目およびその従属項目に指定できません。  
【Enterprise for Linux】THE COLUMN NUMBER CLAUSE CANNOT BE SPECIFIED FOR GROUP ITEM OR THE ITEM WITH OCCURS CLAUSE, THE SUBORDINATE ITEM.

[原因] COLUMN NUMBER 句は, 集団項目, OCCURS 句が指定された項目および OCCURS 句が指定された項目に従属する項目には指定できない。

F D2015 COLUMN NUMBER 句の整数が誤っています。  
【Enterprise for Linux】THE COLUMN NUMBER CLAUSE CONTAINS INVALID INTEGER.

[原因] COLUMN NUMBER 句で指定した値が負の整数または 0 である。

F D2016 COLUMN NUMBER 句の整数が?を超えています。  
【Enterprise for Linux】THE INTEGER IN THE COLUMN NUMBER CLAUSE IS GREATER THAN ?.

[原因] (1) COLUMN NUMBER 句で指定した整数が 255 を超えている。  
(2) 【Enterprise】APPLY SHIFT-CODE 句に PPR-CONTROL-3 が指定された SYSOUT ファイルに対して, COLUMN NUMBER 句で指定した整数が 208 を超えている。

[埋め込み情報] (1) 255  
(2) 208

F D2017 データ部中の節の指定順序が正しくありません。  
【Enterprise for Linux】SECTIONS PRECEDING SYNTAX ERROR IS DETECTED, CHECK COBOL MANUAL FOR CORRECTION.

[原因] データ部中の節の指定順序が誤っている。

F D2018 ? SECTION が 2 度以上指定されています。  
【Enterprise for Linux】THE ? SECTION CANNOT BE SPECIFIED MORE THAN TWICE.

[原因] データ部中の節の指定で同じ節が 2 度以上指定されている。

[埋め込み情報] (1) FILE  
(2) WORKING-STORAGE  
(3) LOCAL-STORAGE  
(4) LINKAGE  
(5) BASED

F D2019 装置名または識別名の指定に矛盾があります。  
【Enterprise for Linux】THE IDENTIFIER OR DEVICE NAME CONFLICTS.

[原因] 装置名または識別名の指定と ORGANIZATION 句の指定に矛盾がある。

F D2020 【Standard】VALUE OF 句で指定された RDB ファイルの表名が 30 桁を超えています。

[原因] VALUE OF 句で指定された RDB ファイルの表名が 30 桁を超えている。

F D2021 ブールデータ項目が USAGE BIT ではありません。  
【Enterprise for Linux】THE USAGE OF THE BOOLEAN DATA ITEM IS NO BIT.

[原因] ブールデータ項目が USAGE BIT でない。

F D2022 ファイル節の 01 レベルの項目はブールデータ項目であってはなりません。  
【Enterprise for Linux】THE 01-LEVEL ITEM IN THE FILE SECTION CANNOT BE BOOLEAN DATA ITEM.

[原因] ファイル節の 01 レベルの項目がブールデータ項目である。

F D2023 ブールデータ項目に THROUGH/THRU 句のある VALUE 句が指定されています。  
【Enterprise for Linux】THE VALUE CLAUSE WITH THE THROUGH/THRU CLAUSE CANNOT BE SPECIFIED FOR THE BOOLEAN DATA ITEM.

[原因] ブールデータ項目に THROUGH/THRU 句のある VALUE 句が指定されている。

W D2024 OCCURS 句の指定があるブールデータ項目は SYNC 句が指定されているものとみなします。

**【Enterprise for Linux】AN IMPLICIT SYNCHRONIZED/SYNC CLAUSE IS SPECIFIED FOR THE BOOLEAN DATA ITEM WITH OCCURS CLAUSE.**

[原因] SYNC 句指定のないブールデータ項目に OCCURS 句が指定されている。  
[コンパイラの処理] SYNC 句ありとみなす。

**W D2025 【Standard】FILE STATUS 句にデータ名 2 が指定されています.**

[原因] このファイルには、FILE STATUS 句にデータ名 2 を指定できない。  
[コンパイラの処理] データ名 2 がないものと見なす。

**F D2028 OPTIONAL 句は指定できません.  
【Enterprise for Linux】THE OPTIONAL CLAUSE CANNOT BE SPECIFIED.**

[原因] **【Enterprise】**UFS ファイル以外に OPTIONAL 句を指定した。  
**【Standard】**大容量記憶ファイル, RDB ファイル以外に OPTIONAL 句を指定した。

**F D2034 RECORD DEPENDING ON 句は SHIFT-CODE 句が指定されたファイルには指定できません.  
【Enterprise for Linux】THE RECORD DEPENDING ON CLAUSE CANNOT BE SPECIFIED FOR SHIFT-CODE APPLIED FILE.**

[原因] RECORD DEPENDING ON 句が指定されているファイルを SHIFT-CODE 句に指定した。

**W D2036 WITH ? 句とファイル記述項の RECORD 句は同時に指定できません. RECORD 句は無視されます.  
【Enterprise for Linux】THE WITH ? CLAUSE AND THE RECORD CLAUSE CANNOT BE SPECIFIED AT THE SAME TIME. THE RECORD CLAUSE IS IGNORED.**

[原因] WITH 句と RECORD 句を同時に指定した。  
[埋め込み情報] (1) FLR  
(2) VLR  
(3) ULR  
[コンパイラの処理] WITH 句の指定を有効とする。

F D2037      **ALTERNATE RECORD KEY** 句はこのファイルには指定できません。  
【Enterprise for Linux】**THE ALTERNATE RECORD KEY CLAUSE CANNOT BE SPECIFIED FOR THIS FILE.**

〔原因〕      UFS ファイル以外に **ALTERNATE RECORD KEY** 句を指定した。

F D2038      **ALTERNATE RECORD KEY** 句の数が二次索引用のファイル識別リテラルまたはデータ名の数と等しくありません。  
【Enterprise for Linux】**NUMBER OF ALTERNATE RECORD KEY CLAUSE IS NOT EQUAL TO THAT OF FILE IDENTIFIER OR DATA NAME.**

〔原因〕      **ALTERNATE RECORD KEY** 句の数が二次索引の内部ファイル名またはデータ名の数が合っていない。

F D2039      乱編成は指定できません。  
【Enterprise for Linux】**THE ACCESS MODE IS RANDOM CANNOT BE SPECIFIED.**

〔原因〕      乱編成は指定できない。

F D2041      **EXTERNAL** 句は **SELECT** 句で **EXTERNAL** 句または **BASED** 句が指定されたファイルには指定できません。  
【Enterprise for Linux】**THE EXTERNAL CLAUSE CANNOT BE SPECIFIED FOR THE FILE WITH EXTERNAL OR BASED CLAUSE.**

〔原因〕      **SELECT** 句で **EXTERNAL** 句または **BASED** 句が指定されたファイルに **FD EXTERNAL** 句を指定した。

F D2042      このファイルは印刷ファイルでなければなりません。  
【Enterprise for Linux】**THIS FILE IS NOT PRINT FILE.**

〔原因〕      ファイルが印刷ファイルでない。

F D2043      **PPR-CONTROL-2** または **PPR-CONTROL-3** は指定できません。  
【Enterprise for Linux】**THE PPR-CONTROL-2 OR PPR-CONTROL-3 CANNOT BE SPECIFIED.**

[原因] APPLY 句に PPR-CONTROL-2 または PPR-CONTROL-3 を指定した。

F D2050 BASED ON 句が誤っています。  
【Enterprise for Linux】THE BASED ON CLAUSE CONTAINS INVALID DESCRIPTION.

[原因] BASED ON 句がベースデータ節のレベル番号 01 項目または 77 項目以外に指定されている。

F D2051 BASED ファイル以外に BASED ON 句が指定されています。  
【Enterprise for Linux】THE BASED ON CLAUSE IS DESIGNATED IN OTHER THAN BASED FILE.

[原因] BASED ファイル以外に BASED ON 句が指定されている。

F D2052 BASED ON 句と一緒に指定できない句が指定されています。  
【Enterprise for Linux】THE SPECIFICATION AND THE BASED ON CLAUSE CANNOT BE SPECIFIED AT THE SAME TIME.

[原因] BASED ON 句と一緒に指定できない句が指定されている。

F D2064 数字定数の値が 2147483647 を超えています。  
【Enterprise for Linux】THE VALUE OF NUMERIC LITERAL EXCEEDS 2147483647.

[原因] 数字定数の値が 2147483647 を超えている。

F D2071 整数に符号は指定できません。  
【Enterprise for Linux】THE INTEGER CONTAINS A SIGN.

[原因] 整数に符号は指定できない。

F D2072 数字定数として 0 (ゼロ) は指定できません。  
【Enterprise for Linux】ZERO VALUE IS SPECIFIED FOR THE NUMERIC LITERAL.

[原因] 数字定数として 0 は指定できない。

F D2073 レベル番号が 2 桁を超えています。  
【Enterprise for Linux】THE LEVEL NUMBER CONSISTS OF MORE THAN 2 DIGITS.

[原因] レベル番号が 2 桁を超えている。

F D2074 レベル番号が誤っています。  
【Enterprise for Linux】THE LEVEL NUMBER IS INVALID.

[原因] (1) 01～49, 66～77, 88 でない。  
(2) 集団項目の先頭が 01 でない。  
(3) ファイル節中に 77 が現れた。  
(4) レコードの最後以外に 66 が現れた。  
(5) 独立項目の後に 66 が現れた。  
(6) 【Enterprise for Windows】【Standard】定数名の後に 66 が現れた。

F D2075 LINAGE 句の整数が 32767 を超えています。  
【Enterprise for Linux】THE INTEGER IN THE LINAGE CLAUSE IS GREATER THAN 32767.

[原因] LINAGE, TOP, BOTTOM の各句の整数が 32767 を超えている。

F D2079 BLOCK CONTAINS 句の定数の指定が誤っています。  
【Enterprise for Linux】WRONG LITERAL IN THE BLOCK CONTAINS CLAUSE.

[原因] BLOCK CONTAINS 句の定数の大小関係が誤っている。

F D2080 ?項で参照されたファイルは SELECT 句で定義されていません。  
【Enterprise for Linux】THE FILE REFERENCED BY ? ENTRY IS NOT DEFINED IN THE SELECT CLAUSE.

[原因] SELECT 句で定義されていないファイルが参照された。

[埋め込み情報] (1) FD  
(2) SD

F D2081 ?項で二重に参照されています。  
【Enterprise for Linux】THE FILE NAME REFERENCED IN THE ? ENTRY IS

**DUPLICATED.**

[原因] 既にファイル記述項で参照のあるファイルが再び参照された。

[埋め込み情報] (1) FD

(2) SD

F D2082 ?句は指定できません。

**【Enterprise for Linux】THE ? CLAUSE CANNOT BE SPECIFIED.**

[原因] (1) 相対ファイル以外に ADDRESS-FORMAT 句を指定した。

(2) 順ファイル以外に FORM-MEDIA 句を指定した。

[埋め込み情報] (1) ADDRESS-FORMAT

(2) FORM-MEDIA

F D2083 ?句は既に指定されています。

**【Enterprise for Linux】THE ? CLAUSE IS DUPLICATED.**

[原因] 既に句は記述されている。

[埋め込み情報] (1) LINAGE

(2) FOOTING

(3) TOP

(4) BOTTOM

F D2084 ?句は既に指定されています。

**【Enterprise for Linux】THE ? CLAUSE IS DUPLICATED.**

[原因] 既に句は記述されている。

[埋め込み情報] (1) BLOCK CONTAINS

(2) RECORD CONTAINS

(3) LABEL RECORD

(4) INDEXED BY

(5) CODE-SET

(6) PICTURE

(7) USAGE

(8) SIGN

(9) OCCURS

- (10) SYNCHRONIZED / SYNC
- (11) JUSTIFIED / JUST
- (12) VALUE
- (13) BLANK
- (14) VALUE OF
- (15) GLOBAL
- (16) EXTERNAL
- (17) CHARACTER TYPE
- (18) COLUMN NUMBER
- (19) PROPERTY

F D2085      符号系名 `?' は定義されていません。  
              【Enterprise for Linux】THE ALPHABET NAME `?' IS NOT DEFINED.

【原因】        CODE-SET 句で参照された符号系名が定義されていない。  
【埋め込み情報】 CODE-SET 句で参照された符号系名

F D2086      ファイル記述項の後にレコード記述項がありません。  
              【Enterprise for Linux】NO DATA DESCRIPTION ENTRY IS FOUND AFTER THE FILE  
                  DESCRIPTION ENTRY.

【原因】        ファイル記述項 / 分類用ファイル記述項の後にレコード記述項がない。

W D2087      BLANK WHEN ZERO 句の ZERO が単数形ではありません。  
              【Enterprise for Linux】THE WORD `ZERO` MUST NOT BE SPELLED AS  
                  `ZEROS/ZEROES` IN THE BLANK WHEN ZERO CLAUSE.

【原因】        BLANK WHEN ZERO 句の ZERO が単数形でない。  
【コンパイラの処理】 ZERO とみなす。

W D2091      必要な終止符がありません。  
              【Enterprise for Linux】THE TERMINATOR `PERIOD` IS MISSING.

【原因】        必要な終止符がない。  
【コンパイラの処理】 あるものとみなす。



F D2092      必要語`?`がありません。  
【Enterprise for Linux】THE OBLIGATORY WORD `?` IS MISSING.

〔原因〕      必要語が記述されていない。

〔埋め込み情報〕 (1) DIVISION  
                  (2) SECTION  
                  (3) INPUT  
                  (4) LENGTH  
                  (5) KEY  
                  (6) RENAMES  
                  (7) IDENTIFICATION

F D2093      `FD`または`SD`の次に必要なファイル名がありません。  
【Enterprise for Linux】NO FILE NAME FOLLOWS THE RESERVED WORD `FD` OR  
              `SD`.

〔原因〕      FD／SD の次にファイル名がない。

F D2094      項目名句が正しくありません。  
【Enterprise for Linux】AN UNRECOGNIZABLE DATA ATTRIBUTE IS ENCOUNTERED.

〔原因〕      データ記述項の書き方が誤っている。

F D2095      ?句の書き方が誤っています。  
【Enterprise for Linux】THE ? CLAUSE CONTAINS INVALID DESCRIPTION.

〔原因〕      句の書き方が誤っている。

〔埋め込み情報〕 (1) BLOCK CONTAINS  
                  (2) RECORD CONTAINS  
                  (3) LABEL RECORD  
                  (4) DATA RECORD  
                  (5) LINAGE  
                  (6) CODE-SET  
                  (7) USAGE  
                  (8) SIGN  
                  (9) VALUE

- (10) OCCURS
- (11) REDEFINES
- (12) VALUE OF
- (13) RECORD
- (14) RECORD IS VARYING
- (15) EXTERNAL
- (16) GLOBAL
- (17) CHARACTER TYPE
- (18) COLUMN NUMBER
- (19) PROPERTY

F D2096     `?`の次に必要な整数がありません。  
【Enterprise for Linux】NO INTEGER FOLLOWS THE RESERVED WORD `?`.

〔原因〕     埋め込み情報の次に整数がない。

〔埋め込み情報〕 (1) TO  
                  (2) OCCURS  
                  (3) FROM

F D2097     ベースデータ節のレコードの数が多すぎます。  
【Enterprise for Linux】TOO MANY RECORDS ARE SPECIFIED IN THE BASE DATA SECTION.

〔原因〕     ポインタ名が 65535 を超えている。

F D2098     ベースデータ節のレコードの書き方が正しくありません。  
【Enterprise for Linux】THE RECORD DESCRIPTION ENTRY IN THE BASE DATA SECTION IS INCORRECT.

〔原因〕     ベースデータ節のレコードの書き方が誤っている。

W D2099     IS または ARE と主語の人称が誤っています。  
【Enterprise for Linux】THE RESERVED WORD IS OR ARE IS NOT COMPATIBLE WITH ITS SUBJECT.

〔原因〕     IS／ARE と主語の人称が合わない。

[コンパイラの処理] 正しいものとみなす。

W D2100 無効となる VALUE OF 句が指定されています。  
【Enterprise for Linux】MEANINGLESS VALUE OF PHRASES ARE SPECIFIED.

[原因] 無効となる VALUE OF 句が指定されている。

[コンパイラの処理] 2 個目以降の IDENTIFICATION 句を無視する。

F D2101 `?'の次に必要なデータ名がありません。  
【Enterprise for Linux】NO DATA NAME FOLLOWS THE RESERVED WORD `?'.

[原因] 次に必要なデータ名がない。

[埋め込み情報] (1) OF/IN

(2) RENAMES

(3) THROUGH/THRU

F D2102 ASCENDING KEY 句または DESCENDING KEY 句の書き方が誤っています。  
【Enterprise for Linux】THE ASCENDING KEY CLAUSE OR THE DESCENDING KEY CLAUSE CONTAINS INVALID DESCRIPTION.

[原因] ASCENDING/DESCENDING KEY 句の書き方が誤っている。

F D2103 ?句の書き方が誤っています。  
【Enterprise for Linux】THE ? CLAUSE CONTAINS INVALID DESCRIPTION.

[原因] 句の書き方が誤っている。

[埋め込み情報] INDEXED BY

F D2111 レベル番号 88 の次に必要な条件名がありません。  
【Enterprise for Linux】NO CONDITION NAME IS SPECIFIED PRECEDED BY THE LEVEL NUMBER `88`.

[原因] レベル番号 88 の次に条件名がない。

F D2112 条件名記述項の書き方が誤っています。  
【Enterprise for Linux】THE CONDITION NAME DESCRIPTION IS INCORRECT.

[原因] レベル番号 88 の記述項の書き方が誤っている。

F D2113 レベル番号の次に必要なデータ名がありません。  
【Enterprise for Linux】NO DATA NAME APPEARS PRECEDED BY LEVEL NUMBER.

[原因] レベル番号の次にデータ名がない。

F D2130 条件名の VALUE 句の定数が昇順ではありません。  
【Enterprise for Linux】THE LEFT HAND SIDE LITERAL OF THRU IS NOT SMALLER THAN THE RIGHT HAND SIDE ONE IN THE VALUE CLAUSE.

[原因] 条件名 VALUE の THROUGH の両側の定数の値の大小関係が誤っている。

F D2131 SIGN 句に関連する基本項目が符号付きではありません。  
【Enterprise for Linux】THE ELEMENTARY ITEM RELATED TO THE SIGN CLAUSE IS UNSIGNED.

[原因] (1) 集団項目レベルに SIGN 句が指定されたにもかかわらず、属する項目に符号付数字基本項目がない。  
(2) 基本項目に SIGN 句が指定されたが、その項目が符号付数字基本項目でない。

F D2132 OCCURS DEPENDING ON 句を持つ項目がレコード記述内の最後の項目ではありません。  
【Enterprise for Linux】THE OCCURS DEPENDING ON ITEM IS NOT THE LAST GROUP OR ELEMENTARY ITEM IN THE RECORD.

[原因] OCCURS(書き方 2)のある項目の後、同一レコード内に新しい領域がある。

F D2133 集団項目に続く項目のレベル番号が先行するレベル番号と一致しません。  
【Enterprise for Linux】THE LEVEL NUMBER IS NOT DEFINED BEFORE IN THE GROUP ITEM.

[原因] 集団項目の定義において、基本項目の直後の項目のレベル番号が、先行するレベル番号と一致しない。

(例)

```
01 A.
 05 B.
 07 C PIC X.
 06 D.
```

F D2134 表が 7 次元を超えています。  
【Enterprise for Linux】THE DEPTH OF DIMENSION EXCEEDS 7.

[原因] 表の次元が 7 を超えている。

F D2136 USAGE 句の指定が上位のレベルと矛盾しています。  
【Enterprise for Linux】THE USAGE CLAUSE DESIGNATION IS INCONSISTENT IN A HIGHER LEVEL DATA DESCRIPTION.

[原因] USAGE 句の指定が上位のレベルと矛盾している。

F D2137 初期値は既に上位のレベルで指定されています。  
【Enterprise for Linux】THE INITIAL VALUE DESIGNATION IS DUPLICATED WITH THE HIGHER LEVEL DESCRIPTION.

[原因] 既に上位のレベルで初期値設定用 VALUE 句が指定されているにもかかわらず再び初期値設定用 VALUE 句が指定された。

F D2138 集団項目の記述項に?句は指定できません。  
【Enterprise for Linux】THE ? CLAUSE CANNOT BE DESIGNATED IN A GROUP ITEM.

[原因] 集団項目レベルの記述項に記述できない句が書かれた。

[埋め込み情報] (1) BLANK  
(2) JUSTIFIED/JUST  
(3) PICTURE  
(4) SYNCHRONIZED/SYNC

F D2139 数字または数字編集項目以外に BLANK 句が指定されています。  
【Enterprise for Linux】THE BLANK CLAUSE IS DESIGNATED IN OTHER THAN NUMERIC OR NUMERIC EDITED ITEM.

[原因] 数字／数字編集項目以外に **BLANK** 句が指定されている。

F D2140      ゼロ抑制指定`\*`の項目に **BLANK** 句が指定されています。  
【Enterprise for Linux】THE BLANK CLAUSE IS DESIGNATED IN THE NUMERIC  
EDITED ITEM WITH `\*` SUPPRESSION.

[原因]      ゼロ抑制指定の`\*`の項目に **BLANK** 句が指定されている。

F D2141      **USAGE DISPLAY** 以外の項目に **BLANK** 句が指定されています。  
【Enterprise for Linux】THE BLANK CLAUSE IS DESIGNATED IN THE ITEM OTHER  
THAN DISPLAY USAGE.

[原因]      **USAGE DISPLAY** 以外の項目に **BLANK** 句が指定されている。

F D2143      **JUSTIFIED/JUST** 句が数字または編集項目に指定されています。  
【Enterprise for Linux】THE JUSTIFIED/JUST PHRASE CANNOT BE DESIGNATED  
IN NUMERIC AND EDITED ITEMS.

[原因]      数字, 数字編集, 英数字編集項目または日本語編集項目に **JUSTIFIED**  
句が指定された。

F D2144      **JUSTIFIED/JUST** 句が **USAGE DISPLAY** 以外の項目に指定されています。  
【Enterprise for Linux】THE JUSTIFIED/JUST PHRASE CANNOT BE DESIGNATED  
IN ITEMS OTHER THAN THE DISPLAY USAGE.

[原因]      **JUSTIFIED/JUST** 句が **USAGE DISPLAY** 以外の項目に指定されている。

F D2145      上位のレベルの記述と **JUSTIFIED/JUST** 句が矛盾しています。  
【Enterprise for Linux】THE JUSTIFIED PHRASE CONFLICTS WITH THE HIGHER  
LEVEL DESCRIPTION.

[原因]      **JUSTIFIED** 句が次の項目に指定された。  
    (a) 上位レベルに初期値 **VALUE** 句がある。  
    (b) 上位レベルが条件変数である。

F D2146 レベル番号 01 項目または 77 項目に OCCURS 句が指定されています。  
【Enterprise for Linux】THE 01-LEVEL ITEM OR THE 77-LEVEL ITEM CONTAINS  
AN OCCURS CLAUSE.

〔原因〕 レベル番号 01／77 の項目に OCCURS 句が指定されている。

F D2147 OCCURS KEY 句の指定と項目の定義が矛盾しています。  
【Enterprise for Linux】THE DESCRIPTION OF OCCURS KEY CLAUSE CONFLICTS  
WITH THE DEFINITION OF DATA ITEM.

〔原因〕 1 つの OCCURS KEY 句に対するデータ名が 2 個以上あるにもかかわらず項目が基本項目である。

F D2148 OCCURS 句の繰り返しが 0 (ゼロ) です。  
【Enterprise for Linux】THE ITERATION COUNT OF THE OCCURS CLAUSE IS ZERO.

〔原因〕 固定長繰り返し項目において、OCCURS 句の繰り返し数がゼロである。

F D2149 OCCURS 句 (書き方 2) の整数が誤っています。  
【Enterprise for Linux】THE FORMAT-2 OCCURS CLAUSE CONTAINS AN INVALID  
INTEGER.

〔原因〕 OCCURS 句の可変繰り返し指定の整数の大小関係が誤っている。

F D2150 繰り返しデータ項目に可変繰り返しデータ項目が従属しています。  
【Enterprise for Linux】THE SUBORDINATE ITEM OF OCCURS CLAUSE CANNOT  
CONTAIN THE FORMAT-2 OCCURS CLAUSE.

〔原因〕 繰り返しデータ項目に可変繰り返しデータ項目が従属している。

F D2152 USAGE INDEX, COMP-1 または COMP-2 以外の基本項目に PICTURE 句が指定されていません。  
【Enterprise for Linux】NO PICTURE CLAUSE IS DESIGNATED FOR THE  
ELEMENTARY ITEM OTHER THAN THE INDEX, COMP-1 OR COMP-2 USAGE.

〔原因〕 USAGE INDEX／COMP-1／COMP-2 以外の基本項目に、PICTURE 句の指定が

ない。

F D2153      USAGE ?の項目に PICTURE 句が指定されています。  
【Enterprise for Linux】THE ITEM OF `USAGE IS ?` CONTAINS THE PICTURE  
CLAUSE.

〔原因〕      USAGE?の項目に PICTURE 句が指定されている。

〔埋め込み情報〕 (1) INDEX  
                  (2) COMP-1/COMPUTATIONAL-1  
                  (3) COMP-2/COMPUTATIONAL-2  
                  (4) POINTER  
                  (5) OBJECT REFERENCE

F D2154      BLANK WHEN ZERO 句を指定したデータ項目に SIGN を指定しています。  
【Enterprise for Linux】THE SIGN CLAUSE CANNOT BE SPECIFIED FOR THE DATA  
ITEM WITH BLANK WHEN ZERO CLAUSE.

〔原因〕      PICTURE 文字 'S' (SIGN) のある数字項目に BLANK WHEN ZERO 句が指定さ  
れた。

F D2155      USAGE ?の PICTURE 句の指定が誤っています。  
【Enterprise for Linux】THE PICTURE CLAUSE SPECIFICATION FOR THE `USAGE  
IS ?` IS INCORRECT.

〔原因〕      USAGE が次の基本項目において PICTURE 句がないか、または項類が数  
字ではない。

〔埋め込み情報〕 (1) COMPUTATIONAL/COMP  
                  (2) COMPUTATIONAL-3/COMP-3  
                  (3) COMPUTATIONAL-5/COMP-5

F D2156      ファイル節のレベル番号 01 の項目に REDEFINES 句が指定されています。  
【Enterprise for Linux】THE REDEFINES CLAUSE IS DESIGNATED FOR THE  
01-LEVEL ITEM IN THE FILE SECTION OR COMMUNICATION SECTION.

〔原因〕      ファイル節のレベル番号 01 の項目に、REDEFINES 句が指定されている。



F D2157 再定義した記述項に初期値設定用 **VALUE** 句が指定されています。  
【Enterprise for Linux】THE INITIAL VALUE DESCRIPTION CANNOT DESIGNATE  
IN THE REDEFINING DATA ITEM.

〔原因〕 同じ記憶領域の 2 番目以降の記述項に初期値設定用の **VALUE** 句がある。

F D2159 **SIGN** 句の指定された `S` 付き数字基本項目が **USAGE DISPLAY** ではありません。  
【Enterprise for Linux】THE USAGE OF THE SIGNED NUMERIC ELEMENTARY ITEM  
IS NOT DISPLAY.

〔原因〕 **SIGN** 句の指定された `S` (符号) 付きの数字基本項目が **USAGE DISPLAY** で  
ない。

F D2161 **USAGE ?** の項目に **SYNCHRONIZED/SYNC** 句が指定されています。  
【Enterprise for Linux】THE ITEM OF `USAGE IS ?` CONTAINS THE  
**SYNCHRONIZED/SYNC** CLAUSE.

〔原因〕 指標データ項目, データポインタデータ項目に **SYNCHRONIZED/SYNC** 句が指  
定されている。

〔埋め込み情報〕 (1) **INDEX**  
(2) **POINTER**

F D2162 上位のレベルの記述と **SYNCHRONIZED/SYNC** 句が矛盾しています。  
【Enterprise for Linux】THE SPECIFICATION OF **SYNCHRONIZED/SYNC** CLAUSE  
CONFLICTS WITH THE HIGHER LEVEL DATA ITEM.

〔原因〕 次の項目に **SYNCHRONIZED** 句が指定された。  
(1) 上位レベルに初期値 **VALUE** 句のある項目  
(2) 上位レベルに条件 **VALUE** 句のある項目

F D2163 上位のレベルに **VALUE** 句の指定された項目の **USAGE** が誤っています。  
【Enterprise for Linux】THE USAGE IS ILLEGAL FOR THE HIGHER LEVEL ITEM  
CONTAINING THE **VALUE** CLAUSE.

〔原因〕 次の項目に **DISPLAY** 以外の **USAGE** が指定された。  
(1) 上位レベルに初期値 **VALUE** 句のある項目

(2) 上位レベルに条件 VALUE 句のある項目

F D2165 REDEFINES 句と初期値設定用 VALUE 句が矛盾しています。  
【Enterprise for Linux】THE SPECIFICATION OF INITIAL VALUE CLAUSE  
CONFLICTS WITH THE ITEM CONTAINING THE REDEFINES CLAUSE.

〔原因〕 次の項目に初期値 VALUE 句が書かれた。  
(1) REDEFINES 句を含む記述項  
(2) REDEFINES 句を含む記述項に従属する記述項

F D2166 初期値設定用 VALUE 句は指定できません。  
【Enterprise for Linux】THE SPECIFICATION OF INITIAL VALUE CLAUSE CANNOT  
BE DESIGNATED.

〔原因〕 ベースデータ節で初期値設定用 VALUE 句が書かれている。

F D2168 USAGE ? のデータ項目に初期値設定用 VALUE 句が指定されています。  
【Enterprise for Linux】THE INITIAL VALUE CLAUSE IS DESIGNATED FOR THE ?  
DATA ITEM.

〔原因〕 指標データ項目、オブジェクト参照項目に初期値設定用 VALUE 句が指定さ  
れている。

〔埋め込み情報〕 (1) INDEX  
(2) OBJECT REFERENCE

F D2169 USAGE ? の項目が条件変数に指定されています。  
【Enterprise for Linux】THE ITEM OF `USAGE IS ?` CANNOT BE A CONDITIONAL  
VARIABLE.

〔原因〕 指標データ項目、オブジェクト参照項目、データポインタデータ項目が条件  
変数に指定されている。

〔埋め込み情報〕 (1) INDEX  
(2) POINTER  
(3) OBJECT REFERENCE

F D2170 集団項目の VALUE 句で英数字定数以外が指定されています。

【Enterprise for Linux】THE GROUP ITEM CAN SPECIFY THE VALUE CLAUSE ONLY CONTAINING A NONNUMERIC LITERAL.

- 〔原因〕 集団項目の次の VALUE 句に対し数字定数または日本語定数が指定されている。
- (1) 条件 VALUE 句
  - (2) 初期値 VALUE 句

F D2171 項類と VALUE 句の定数が矛盾しています。  
 【Enterprise for Linux】THE LITERAL IN VALUE CLAUSE CONFLICTS WITH ITS CATEGORY.

- 〔原因〕 次の表の条件に違反している。

| 条件 | 項目                                 | 定数                               |
|----|------------------------------------|----------------------------------|
| 1  | 数字項目                               | 数字定数<br>【Standard】LENGTH OF データ名 |
| 2  | 英字項目<br>英数字項目<br>英数字編集項目<br>数字編集項目 | 英数字定数                            |
| 3  | 日本語項目<br>日本語編集項目                   | 日本語定数                            |

W D2172 VALUE 句の値で 0(ゼロ)以外の桁落ちが発生します。  
 【Enterprise for Linux】SIGNIFICANT DIGITS (OTHER THAN ZERO) SPECIFIED IN THE VALUE CLAUSE ARE TRUNCATED.

- 〔原因〕 VALUE 句の値で 0 以外の桁落ちが発生する。  
 〔コンパイラの処理〕 MOVE 文の転記の規則と同様に切り捨てが行われる。

F D2173 PICTURE 句と VALUE 句の符号の指定が矛盾しています。  
 【Enterprise for Linux】THE SIGN DESIGNATED IN THE PICTURE CLAUSE CONFLICTS WITH THE ONE DESIGNATED IN THE VALUE CLAUSE.

[原因] PIC 文字列に 'S' (符号桁) がない項目に対し, 次の VALUE 句で符号付き数字定数が指定されている。

- (1) 初期値 VALUE 句
- (2) 条件 VALUE 句

F D2174 USAGE DISPLAY 以外の項目に対する VALUE 句の定数指定が誤っています。  
【Enterprise for Linux】THE LITERAL IN VALUE CLAUSE IS INCONSISTENT WITH THE USAGE CLAUSE SPECIFICATION.

[原因] USAGE 種別と VALUE (初期値, 条件) 句が次の組み合わせ以外である。

| USAGE 種別 | 指定可能な定数                           |
|----------|-----------------------------------|
| INDEX    | 指定不可                              |
| COMP     | 数字定数または表意定数 ZERO                  |
| COMP-1   |                                   |
| COMP-2   |                                   |
| COMP-3   |                                   |
| COMP-5   |                                   |
| POINTER  | NULL<br>【Standard】ADDRESS OF データ名 |

W D2175 EXTERNAL 句が記述されたデータ項目の名前が 8 桁を超えています。  
【Enterprise for Linux】THE LENGTH OF THE DATA NAME WITH THE EXTERNAL CLAUSE IS OUTSIDE THE RANGE.

[原因] EXTERNAL 句が記述されたデータ項目の名前が 8 桁を超えている。

[コンパイラの処理] そのまま翻訳を続ける。

W D2176 VALUE 句で定数の桁落ちが発生します。  
【Enterprise for Linux】THE NONNUMERIC LITERAL DESIGNATED IN THE VALUE CLAUSE IS TRUNCATED.

[原因] 次の VALUE 句において英数字定数または日本語定数の桁落ちが発生する。

- (1) 初期値 VALUE 句

(2) 条件 **VALUE** 句

これは **PICTURE** 句で指定された文字列長（全長）より長い文字列を **VALUE** 句で指定したために起こる

[コンパイラの処理] **JUSTIFIED** 句の指定に関係なく項目長に合わせて右側が切り捨てられる。

F D2177 順編成ファイルに指定できない句があります。

**【Enterprise for Linux】INVALID CLAUSE FOR SEQUENTIAL FILE ARE SPECIFIED.**

[原因] 順ファイルに指定できない句がある。

(1) **RELATIVE KEY** 句

(2) **RECORD KEY** 句

(3) **CONTROL-AREA** 句

F D2178 順ファイルに指定された **ACCESS MODE** 句が誤っています。

**【Enterprise for Linux】THE ACCESS MODE DESIGNATED FOR THE SEQUENTIAL FILE IS NOT SEQUENTIAL.**

[原因] 順ファイルに指定された **ACCESS MODE** 句が誤っている。

(**SEQUENTIAL MODE** 以外の指定)

F D2179 相対ファイルが大記憶装置以外に割り当てられています。

**【Enterprise for Linux】THE RELATIVE FILE IS ASSIGNED TO OTHER THAN MASS STORAGE DEVICES.**

[原因] デバイス種別に対して指定できないファイル編成を指定している。

F D2180 相対編成ファイルに指定できない句があります。

**【Enterprise for Linux】INVALID CLAUSES FOR RELATIVE FILE ARE SPECIFIED.**

[原因] 相対ファイルに指定できない句がある。

(1) **RECORD KEY** 句

(2) **CONTROL-AREA** 句

F D2181 相対ファイルの乱呼び出し法または動的呼び出し法に **RELATIVE KEY** 句がありません。

**【Enterprise for Linux】THE RELATIVE KEY CLAUSE IS MISSING IN THE RELATIVE FILE OTHER THAN SEQUENTIAL ACCESS MODE.**

〔原因〕 相対ファイルの乱／動的呼び出し法に **RELATIVE KEY** 句がない。

F D2183 索引ファイルが大記憶装置以外に割り当てられています。  
**【Enterprise for Linux】THE INDEXED FILE IS ASSIGNED TO OTHER THAN MASS STORAGE DEVICES.**

〔原因〕 索引ファイルが大記憶装置以外に割り当てられている。

F D2184 索引編成ファイルに指定できない句があります。  
**【Enterprise for Linux】INVALID CLAUSES INDEXED FILE ARE SPECIFIED.**

〔原因〕 索引ファイルに指定できない句がある。  
(1) **RELATIVE KEY** 句  
(2) **CONTROL-AREA** 句

F D2185 索引ファイルに **RECORD KEY** 句が指定されていません。  
**【Enterprise for Linux】NO RECORD KEY CLAUSE IS DESIGNATED FOR THE INDEXED FILE.**

〔原因〕 索引ファイルに **RECORD KEY** 句が指定されていない。

F D2186 **RECORD** 句で指定した整数が誤っています。  
**【Enterprise for Linux】THE INTEGER IN THE RECORD CLAUSE IS INVALID.**

〔原因〕 **FD/SD** 句中の **RECORD** 句の指定で  
(1) 整数が **65535** を超えている。  
(2) 整数 2 > 整数 3 (**RECORD CONTAINS** 整数 2 **TO** 整数 3)  
(3) 整数 4 > 整数 5 (**RECORD VARYING FROM** 整数 4 **TO** 整数 5)  
(4) **RECORD** 句で整数を指定するべきところに整数が指定されていない。

F D2188 ?句の指定されたファイルのレコード中に許されない記述があります。  
**【Enterprise for Linux】THE RECORD OF THE FILE SPECIFIED IN THE ? PHRASE CONTAINS AN UNACCEPTABLE DESCRIPTION.**

- [原因] (1) **APPLY SHIFT-CODE** 句の指定されたファイルのレコード記述項に **DEPENDING ON** 句, または **SYNCHRONIZED** 句, またはプリンタファイルのときに表示用以外の項目が記述された。
- (2) **CODE-SET** 句の指定されたファイルのレコード記述項に **DEPENDING ON** 句, または **SYNCHRONIZED** 句, または表示用以外の項目, または **SIGN IS SEPARATE** 句のない符号付き数字項目が記述された。

[埋め込み情報] (1) **APPLY SHIFT-CODE**  
 (2) **CODE-SET**

F D2189 ? 句の指定されたファイルのファイル記述項に許されない記述があります.  
**【Enterprise for Linux】THE FILE DESCRIPTION SPECIFIED IN THE ? PHRASE CONTAINS AN UNACCEPTABLE DESCRIPTION.**

- [原因] **CODE-SET** 句の指定されたファイルのファイル記述項に **RECORD DEPENDING** 句, または **RECORD VARYING** 句が記述された。

[埋め込み情報] **CODE-SET**

F D2190 **LABEL RECORD** 句で **OMITTED** 以外は指定できません.  
**【Enterprise for Linux】THE LABEL RECORD CLAUSE CAN BE SPECIFIED ONLY FOR OMITTED.**

- [原因] トランザクションファイルの **LABEL RECORD** 句に **OMITTED** 以外が指定された。

F D2192 **LINAGE** の定数より大きい **FOOTING** の定数が指定されています.  
**【Enterprise for Linux】THE LITERAL DESIGNATED IN THE FOOTING PHRASE IS GREATER THAN THAT SPECIFIED IN THE LINAGE CLAUSE.**

- [原因] **LINAGE** の定数より **FOOTING** の定数が大きい。

F D2193 論理ページの大きさが **32767** を超えています.  
**【Enterprise for Linux】THE LOGICAL PAGE SIZE EXCEEDS 32767.**

- [原因] **LINAGE** 句に指定された論理ページの大きさが **32767** を超えた。

F D2194 **CODE-SET** 句の指定できないファイルです。

**【Enterprise for Linux】NO CODE-SET CLAUSE CAN BE DESIGNATED FOR THE FILE.**

- [原 因]           (1) CODE-SET 句を順ファイル以外に指定している。  
                   (2) CODE-SET 句を APPLY SHIFT-CODE 句が指定されたファイルに指定している。

F D2197       SELECT 句で定義されたファイルに対するファイル記述項がありません。  
**【Enterprise for Linux】THE FILE DESIGNATED IN THE SELECT CLAUSE IS NOT REFERENCED IN THE FILE DESCRIPTION ENTRY.**

- [原 因]       SELECT 句で定義されたファイルに対するファイル記述項がない。

F D2198       SAME AREA 句または SAME RECORD AREA 句の書き方が誤っています。  
**【Enterprise for Linux】THE SAME AREA CLAUSE OR THE SAME RECORD AREA CLAUSE CONTAINS INVALID DESCRIPTION.**

- [原 因]       SAME AREA 句, SAME RECORD AREA 句について、次の規則に違反している。  
                   (1) ある SAME AREA 句に書いたファイル名を、1 つでも SAME RECORD AREA 句に書いた場合、その SAME AREA 句の他のファイル名もすべて同じ SAME AREA 句に書かねばならない。

F D2199       APPLY 句の指定が誤っています。  
**【Enterprise for Linux】THE APPLY CLAUSE IN THE FILE IS INVALID.**

- [原 因]       APPLY 句の指定の仕方が誤っている。  
                   (1) **【Enterprise for Windows】**大容量記憶ファイル以外の順ファイルに APPLY EXCLUSIVE-MODE/SHARED-MODE が指定された。  
                   **【Standard】**CSV 形式の行順ファイルに APPLY PROTECTED-MODE が指定された。  
                   **【 Enterprise for Linux 】**順ファイルに APPLY EXCLUSIVE-MODE/SHARED-MODE が指定された。または、索引ファイルに APPLY EXCLUSIVE-MODE/SHARED-MODE と LOCK MODE が同時指定された。  
                   (2) APPLY SHIFT-CODE 句が SYSOUT ファイルならびにプリンタファイル以外に指定された。



F D2200 GLOBAL 句は SAME RECORD AREA 句を指定したファイルまたはそのレコードには指定できません。  
【Enterprise for Linux】THE GLOBAL CLAUSE CANNOT BE SPECIFIED FOR THE FILE WITH SAME RECORD AREA CLAUSE.

〔原因〕 GLOBAL 句は SAME RECORD AREA 句を指定したファイルまたはそのレコードには指定できない。

F D2201 EXTERNAL 句は SAME 句を指定したファイルには指定できません。  
【Enterprise for Linux】THE EXTERNAL CLAUSE CANNOT BE SPECIFIED FOR THE FILE WITH SAME CLAUSE.

〔原因〕 EXTERNAL 句は SAME 句を指定したファイルには指定できない。

F D2202 CODE-SET 句で指定された符号系名の符号名が誤っています。  
【Enterprise for Linux】THE ALPHABET NAME REFERENCED BY THE CODE-SET CLAUSE IS INVALID.

〔原因〕 CODE-SET 句で指定された符号系名で定義された符号名が NATIVE 以外である。

F D2203 整列用ファイル記述項に?句は指定できません。  
【Enterprise for Linux】THE ? CLAUSE CANNOT BE SPECIFIED IN THE SORT FILE DESCRIPTION ENTRY.

〔原因〕 分類用ファイル記述項で指定できない句を指定している。

〔埋め込み情報〕 (1) BLOCK  
(2) LABEL  
(3) VALUE  
(4) LINAGE  
(5) CODE-SET

F D2204 整列用ファイルに指定されたファイルが大記憶装置ではありません。  
【Enterprise for Linux】THE SORT FILE MUST BE ASSIGNED TO A MASS STORAGE

DEVICE.

〔原因〕 整列用ファイルに次に示すファイルが指定された。

- (1) プリンタファイル
- (2) SYSOUT ファイル

F D2205      **【Enterprise for Linux】【Standard】RECORD KEY 句の書き方が誤っています。  
【Enterprise for Linux】THE RECORD KEY CLAUSE IN THE FILE IS INVALID.**

〔原因〕      (1) 一般ファイルに RECORD KEY 句の WITH DUPLICATES が指定されている。  
                 (2) 一般ファイルに対する RECORD KEY 句に複数のデータ名が指定されている。  
                 (3) 一般ファイルに対する ALTERNATE RECORD 句に複数のデータ名が指定されている。

F D2206      行順ファイルの割り当てが誤っています。  
                 **【Enterprise for Linux】THE LINE SEQUENTIAL FILE IS ASSIGNED TO WRONG  
DEVICES.**

〔原因〕      UFS ファイル以外に LINE SEQUENTIAL を指定した。

F D2208      LABEL RECORD は STANDARD でなければなりません。  
                 **【Enterprise for Linux】THIS FILE MUST BE STANDARD LABEL.**

〔原因〕      LABEL RECORD 句に STANDARD 以外を指定した。

F D2209      ユーザラベルはこのファイルには指定できません。  
                 **【Enterprise for Linux】THE USER LABEL CANNOT BE SPECIFIED FOR THIS FILE.**

〔原因〕      順編成以外および装置名が SYSOUT のファイルで、LABEL RECORD 句にデータ  
                 名を指定した。

F D2218      整列用ファイルに?句が指定されています。  
                 **【Enterprise for Linux】THE ? CLAUSE CANNOT BE SPECIFIED IN THE SORT FILE.**

[原因] 整列用ファイルに次に示す句が指定された。

(1) EXTERNAL 句

(2) BASED 句

[埋め込み情報] (1) EXTERNAL

(2) BASED

F D2219 整列用ファイルが?句で指定されています。

【Enterprise for Linux】THE SORT FILE CANNOT BE SPECIFIED IN THE ? CLAUSE.

[原因] 整列用ファイルが次に示す句で指定された。

(1) MULTIPLE FILE 句

(2) EXTERNAL 句

(3) BASED 句

(4) SAME AREA 句

[埋め込み情報] (1) MULTIPLE FILE

(2) EXTERNAL

(3) BASED

(4) SAME AREA

F D2220 SAME AREA または SAME SORT AREA 句の書き方が誤っています。

【Enterprise for Linux】THE SAME AREA CLAUSE OR THE SAME SORT AREA CLAUSE  
CONTAINS INVALID DESCRIPTION.

[原因] SAME AREA/SAME SORT AREA 句の書き方が誤っている。

F D2221 定数が整数ではありません。

【Enterprise for Linux】THE NUMERIC LITERAL CONTAINS A DECIMAL POINT.

[原因] 定数が整数でない。

F D2222 【Enterprise for Windows】【Standard】定数名記述項が記述できない節です。

[原因] (1) 作業場所節以外に定数名記述項が記述してある。

F D2223 数字定数の値が 65535 を超えています。

【Enterprise for Linux】THE VALUE OF NUMERIC LITERAL EXCEEDS 65535.

[原 因] 数字定数の値が 65535 を超えている。

F D2224 【Enterprise for Windows】【Standard】定数名記述項の書き方が誤っています。

[原 因] (1) VALUE 句がない。  
(2) VALUE 句と他の句が同時に指定してある。

F D2232 プリンタファイル以外に LINAGE 句が指定されています。  
【Enterprise for Linux】THE LINAGE CLAUSE CAN BE DESIGNATED FOR ONLY THE  
PRINTER FILE.

[原 因] プリンタファイル以外に LINAGE 句の指定がある。

F D2233 VALUE OF 句が指定されていません。  
【Enterprise for Linux】NO VALUE OF CLAUSE IS FOUND.

[原 因] (1) ラベルレコードあり (LABEL RECORD STANDARD 指定) のファイルに VALUE  
OF 句が指定されていない。  
(2) 【Standard】RDB ファイルに対するファイル記述項に VALUE OF 句が指定  
されていない。

F D2235 日本語定数が THROUGH/THRU 句のある条件名に指定されています。  
【Enterprise for Linux】THE GRAPHIC LITERAL CANNOT BE SPECIFIED IN THE  
THROUGH/THRU PHRASE.

[原 因] 日本語定数が THROUGH/THRU 句のある条件名に指定されている。

F D2237 SAME SORT AREA 句に整列用ファイルが指定されていません。  
【Enterprise for Linux】NO SORT FILE IS SPECIFIED IN THE SAME SORT AREA  
CLAUSE.

[原 因] SAME SORT AREA 句に整列用ファイルが指定されていない。

F D2240 本機能は指定できません。  
【Enterprise for Linux】THIS FUNCTION CANNOT BE SPECIFIED.

- [原因] (1) **-Qa** オプション指定なしの場合、  
連絡節のデータ項目に使用できない **GLOBAL** 句を指定している。
- (2) **-Qb** オプション指定なしの場合、  
ベースデータ節のデータ項目に使用できない **GLOBAL** 句を指定している。
- (3) **-Zr** オプション指定ありの場合、  
レコード記述項に使用できないブールデータ型を指定している。  
レコード記述項に使用できない **SYNC** 句を指定している。

**F D2302** 原型のデータ部に連絡節以外の節が指定されています。  
【Enterprise for Linux】**NO SECTION EXCEPT LINKAGE CAN BE SPECIFIED IN DATA DIVISION OF PROTOTYPE.**

[原因] 原型のデータ部に連絡節以外の節が指定されている。

**F D2303** 外部公開名は英字または下線で始まり、英字、数字または下線からなる英数字定数でなければなりません。  
【Enterprise for Linux】**EXTERNALIZED NAME MUST BE ALPHANUMERIC CONSTANT STARTED BY ALPHABET OR UNDERSCORE AND CONSISTS OF ALPHABET/NUMERIC/UNDERSCORE.**

- [原因] **AS** 句の定数が次のいずれかである。
- (1) 英字 **A～Z**、数字 **0～9**、下線でない文字が含まれている。
- (2) 語頭が、英字および下線でない。

**F D2304** 環境部の情報が異なる?が既に定義されています。  
【Enterprise for Linux】**? IS ALREADY DEFINED AND HAS DIFFERENCE IN ENVIRONMENT DIVISION.**

- [原因] 次のいずれかの場合。
- (1) 翻訳中のプログラムと同一の外部公開名を持つプログラムが同じ翻訳群中で既に翻訳されており、環境部の呼び出し情報が異なっている。
- (2) 翻訳中の関数と同一の外部公開名を持つ関数が同じ翻訳群中で既に翻訳されており、環境部の呼び出し情報が異なっている。

[埋め込み情報] (1) **PROGRAM**

## (2) FUNCTION

**W D2305** ?は先行する翻訳単位のリポジトリ段落で後方参照されています。  
【Enterprise for Linux】? IS BACKWARD REFERENCED IN REPOSITORY PARAGRAPH  
OF PRIOR COMPILATION UNIT.

【原因】 リポジトリ段落にクラス、関数、プログラムのいずれかの参照を持つ翻訳単位  
より後に、その定義を持つ翻訳単位が翻訳された。

【埋め込み情報】 (1) CLASS  
(2) FUNCTION  
(3) PROGRAM

【コンパイラの処理】 外部リポジトリの情報は無視し、参照より後の翻訳単位に現われた定  
義を有効とする。

**F D2306** リポジトリ段落で宣言された指定子の情報が外部リポジトリに見つかりません。  
【Enterprise for Linux】NO SPECIFIER DEFINED IN REPOSITORY PARAGRAPH IS  
FOUND IN EXTERNAL REPOSITORY.

【原因】 リポジトリ段落に次のいずれかの指定が行われた場合。  
(1) 外部リポジトリに存在しないクラス名または外部公開名が指定されてい  
る。  
(2) 外部リポジトリに存在しない関数原型名または外部公開名が指定され  
ている。  
(3) 外部リポジトリに存在しないプログラム名または外部公開名が指定され  
ている。

**F D2307** リポジトリ情報にエラーが発生しました. CODE=?.  
【Enterprise for Linux】REPOSITORY INFORMATIONAL ERROR. CODE=?.

【原因】 リポジトリ情報を処理する際にエラーが発生した。

【埋め込み情報】 (1) 10 (外部リポジトリファイルの形式が不正である)  
(2) 12 (外部リポジトリファイルの読込に失敗した)  
(3) 17 (外部リポジトリファイル名が長すぎる)  
(4) 20 (製品構成ファイル CobolRepositorySchema.xsd が存在しない)

F D2308      ?定義中に?は指定できません.  
**【Enterprise for Linux】IN ? DEFINITION, ? CAN NOT DEFINE.**

[原因]          次のいずれかの場合。

- (1) メソッド定義中にファイル節が指定された。
- (2) メソッド定義中に作業場所節が指定された。
- (3) ファクトリ定義, およびインスタンス定義中に局所記憶節が指定された。
- (4) ファクトリ定義, およびインスタンス定義中に連絡節が指定された。
- (5) ファクトリ定義, インスタンス定義, およびメソッド定義中に **GLOBAL** 句が指定された。
- (6) ファクトリ定義, およびインスタンス定義中にベースデータ節が指定された。

[埋め込み情報]      埋め込み情報 1

- (1)    **METHOD**
- (2)    **METHOD**
- (3)    **FACTORY**  
         **OBJECT**
- (4)    **FACTORY**  
         **OBJECT**
- (5)    **FACTORY**  
         **OBJECT**  
         **METHOD**
- (6)    **FACTORY**  
         **OBJECT**

埋め込み情報 2

- (1)    **FILE SECTION**
- (2)    **WORKING-STORAGE SECTION**
- (3)    **LOCAL-STORAGE SECTION**
- (4)    **LINKAGE SECTION**
- (5)    **GLOBAL**
- (6)    **BASED SECTION**

F D2309      ?中にオブジェクト参照項目は指定できません.  
**【Enterprise for Linux】NO OBJECT-REFERENCE ITEM CAN SPECIFY IN ?.**

[原因] 次のいずれかの場合。

- (1) ファイル節にオブジェクト参照項目が指定された。
- (2) ベースデータ節にオブジェクト参照項目が指定された。

[埋め込み情報] (1) FILE SECTION  
(2) BASED SECTION

F D2310 PROPERTY 句がファクトリ定義、インスタンス定義の作業場所節以外に指定されています。

【Enterprise for Linux】PROPERTY PHRASE CAN SPECIFY IN WORKING-STORAGE SECTION OF FACTORY OR INSTANCE DEFINITION.

[原因] PROPERTY 句がファクトリ定義、インスタンス定義の作業場所節以外に指定されている。

F D2311 USAGE ?が 01 レベルの基本データ項目以外に指定されています。

【Enterprise for Linux】USAGE ? MUST BE SPECIFIED IN 01 LEVEL ELEMENTARY DATA ITEM.

[原因] 01 レベルの基本データ項目以外に対して、USAGE 句が OBJECT REFERENCE を伴って指定されている。

[埋め込み情報] OBJECT REFERENCE

F D2312 ACTIVE-CLASS がファクトリ定義の作業場所節、インスタンス定義の作業場所節、メソッド定義の連絡節、メソッド定義の局所記憶節以外で指定されています。

【Enterprise for Linux】ACTIVE-CLASS CAN SPECIFY IN WORKING-STORAGE SECTION OF FACTORY OR INSTANCE DEFINITION, LINKAGE OR LOCAL-STORAGE SECTION OF METHOD DEFINITION.

[原因] ACTIVE-CLASS がファクトリ定義の作業場所節、インスタンス定義の作業場所節、メソッド定義の連絡節、メソッド定義の局所記憶節以外で指定されている。

F D2314 継承元クラス名がリポジトリ段落に指定されていません。

【Enterprise for Linux】INHERITANCE BASE CLASS NAME IS NOT SPECIFIED IN REPOSITORY PARAGRAPH.



【原因】 継承元クラスに指定したクラス名が、リポジトリ段落中のクラス指定子に記載されていない。

F D2315 クラスの継承が循環しています。  
【Enterprise for Linux】CIRCULAR REFERENCE OF CLASS.

【原因】 (1) 継承元クラスが、直接または間接に現在翻訳中のクラスを継承している。  
(2) 継承元クラスのリポジトリ情報が不整合である。

F D2316 外部公開名が重複しています。  
【Enterprise for Linux】DUPLICATED EXTERNALIZED NAME.

【原因】 次の外部公開名が重複しています。  
・最外側のプログラム名  
・クラス名  
・関数原型名  
・プログラム原型名  
・利用者定義関数名  
・EXTERNAL 属性付きで記述された項目の  
・データ名  
・ファイル名

F D2317 リポジトリ段落中の`?`が重複しています。  
【Enterprise for Linux】? IN REPOSITORY PARAGRAPH IS DUPLICATED.

【原因】 リポジトリ段落中で、クラス名、プログラム原型名、関数原型名、プロパティ名が 2 度以上指定され、かつ、その名前に対する指定が異なっている。

【埋め込み情報】 (1) クラス名  
(2) プログラム原型名  
(3) 関数原型名  
(4) プロパティ名

F D2318 上位クラスに同名のメソッドが定義されていません。  
【Enterprise for Linux】SAME NAME METHOD IS NOT DEFINED IN SUPERCLASS.

- [原因] (1) ファクトリ定義中の **VERRIDE** 指定があるメソッド定義と同名のメソッドが, 上位クラスのファクトリ定義中に定義されていない。
- (2) インスタンス定義中の **VERRIDE** 指定があるメソッド定義と同名のメソッドが, 上位クラスのインスタンス定義中に定義されていない。

F D2319 上位クラスに同名のメソッドが定義されています。

**【Enterprise for Linux】SAME NAME METHOD IS DEFINED IN SUPERCLASS.**

- [原因] **VERRIDE** 指定がないメソッド定義と同名のメソッドが, 上位クラスに定義されている。

F D2320 **PROPERTY** 句が指定されたデータ項目と同名のプロパティが定義されています。

**【Enterprise for Linux】THE SAME NAME OF PROPERTY PHRASE SPECIFIED DATA ITEM IS ALREADY DEFINED.**

- [原因] (1) ファクトリ定義またはインスタンス定義中の作業場所節に **PROPERTY** 句付きのデータ項目が定義されており, 同じファクトリ定義またはインスタンス定義中に同名のプロパティメソッドが定義されている。
- (2) 同名の **PROPERTY** 句付きデータ項目が定義されている。

F D2321 プロパティ名は既に上位クラスで定義されています。

**【Enterprise for Linux】SAME NAME OF PROPERTY IS DEFINED IN SUPERCLASS.**

- [原因] **PROPERTY** 句が指定されたデータ項目の名前が, 既に上位クラスでプロパティ名として定義されている。

F D2322 **USAGE OBJECT REFERENCE** で指定されたクラスは, リポジトリ段落で指定されていません。

**【Enterprise for Linux】CLASS THAT SPECIFIED IN USAGE OBJECT REFERENCE IS NOT SPECIFIED IN REPOSITORY PARAGRAPH.**

- [原因] **USAGE OBJECT REFERENCE** で指定されたクラスが, リポジトリ段落に記載されていない。または, 外部リポジトリ中に見つからない。

F D2323 **【Enterprise for Windows】【Standard】**

クラスの継承が 512 段を超えています。

**【Enterprise for Linux】**

クラスの継承が 64 段を超えています。

**【Enterprise for Linux】TOO MUCH CLASS INHERITANCE.**

[原因] **【Enterprise for Windows】【Standard】**

クラスの継承の深さが上限の 512 を超えている。

**【Enterprise for Linux】**

クラスの継承の深さが上限の 64 を超えている。

F D2324 CSV 形式の行順ファイルのレコードに OCCURS DEPENDING ON 句が指定されています。

**【Enterprise for Linux】OCCURS DEPENDING ON PHRASE CAN NOT BE SPECIFIED FOR RECORD OF CSV TYPE LINE-SEQUENTIAL FILE.**

[原因] CSV 形式の行順ファイルのレコード記述項に OCCURS DEPENDING ON 句が記述された。

F D2326 **【Enterprise for Windows】【Standard】?**が作業場所節以外で指定された。

[原因] 作業場所節以外で初期値設定用 VALUE 句に ADDRESS OF 句または LENGTH OF 句が指定された。

[埋め込み情報] (1) ADDRESS OF  
(2) LENGTH OF

F D2327 **【Enterprise for Windows】【Standard】**入れ子の内側プログラムまたは、再帰的プログラム中に?は指定できません。

[原因] 入れ子の内側プログラムまたは、再帰的プログラム中の初期値設定用 VALUE 句に ADDRESS OF 句または LENGTH OF 句が指定された。

[埋め込み情報] (1) ADDRESS OF  
(2) LENGTH OF

F D2328 **【Enterprise for Windows】【Standard】**プログラム定義中以外に?は指定できません。

[原因] プログラム定義中以外の初期値設定用 VALUE 句に ADDRESS OF 句または

LENGTH OF 句が指定された。

[埋め込み情報] (1) ADDRESS OF  
(2) LENGTH OF

F D2330      **【Enterprise for Windows】【Standard】`?`**の修飾が制限を超えています。

[原因]          データの修飾が 50 を超えている。

[埋め込み情報] データ名 (修飾される語)

F D2331      **【Enterprise for Windows】【Standard】?**は OCCURS 句を指定した項目およびその従属項目に指定できません。

[原因]          OCCURS 句が指定された項目および OCCURS 句が指定された項目に従属する項目には指定できない。

[埋め込み情報] (1) ADDRESS OF  
(2) LENGTH OF

F D2332      **【Enterprise】RAF** ファイルと局所記憶節は同時に記述できません。  
**【Enterprise for Linux】RAF FILE AND LOCAL-STORAGE SECTION ARE NOT COMPATIBLE.**

[原因]          RAF ファイルと局所記憶節を同時に指定した。

F D2333      **【Enterprise】RAF** ファイルとオブジェクト参照項目は同時に記述できません。  
**【Enterprise for Linux】RAF FILE AND OBJECT-REFERENCE ITEM ARE NOT COMPATIBLE.**

[原因]          RAF ファイルとオブジェクト参照項目は同時に指定した。

F D2334      **【Standard】【Enterprise for Windows】?**句を指定する必要があります。

[原因]          (1) -CF オプション指定ありのとき、CSV 形式の行順ファイルには、CODE-SET IS UTF-8 または CODE-SET IS UTF-16LE を指定する必要があります。

[埋め込み情報] (1) CODE-SET

F D2335      **【Enterprise】識別名 RAF** はマルチスレッドで使用できません。

**【Enterprise for Linux】RAF FILE CANNOT BE USED IN MULTI-THREAD.**

[原因] -MT オプション指定ありの場合, RAF 順編成ファイルを記述している。

F D3001 **【Standard】**主画面はただ 1 つしか書けません。

[原因] 主画面は画面節の先頭にただ 1 つしか書けない。

F D3002 **【Standard】**?句は既に指定されています。

[原因] 句はすでに指定されている。

[埋め込み情報] (1) END STATUS  
(2) LINE POSITION  
(3) COLUMN POSITION  
(4) BTAB POSITION  
(5) WINDOW STATUS  
(6) WINDOW HIGHT  
(7) WINDOW WIDTH  
(8) DISPLAY LINE  
(9) DISPLAY COLUMN  
(10) BORDER  
(11) LINE  
(12) COLUMN  
(13) CLEAR  
(14) BLANK  
(15) JUSTIFIED/JUST  
(16) SHIFTED  
(17) BUZZER  
(18) SECRET  
(19) BLINK  
(20) HIGH INTENSITY  
(21) REVERSE  
(22) CHECK  
(23) CHECK OVERFLOW  
(24) CHECK LENGTH  
(25) CHECK OMISSION

- (26) CHECK MODULUS
- (27) CHECK BTAB
- (28) CHECK ADVANCE
- (29) IFC
- (30) FILL
- (31) INPUT
- (32) COLOR
- (33) WAIT
- (34) PREVIOUS ATTRIBUTE
- (35) OVER LINE
- (36) UNDER LINE
- (37) BOX
- (38) VERTICAL LINE AT
- (39) OCCURS
- (40) OVER LINE TO
- (41) UNDER LINE TO
- (42) MIXED

F D3003      **【Standard】**?句は集団項目に指定できません.

[原因]          句は集団項目に指定できない。

- [埋め込み情報]
- (1) COLUMN
  - (2) BLANK
  - (3) JUSTIFIED/JUST
  - (4) BUZZER
  - (5) SECRET
  - (6) BLINK
  - (7) HIGH INTENSITY
  - (8) REVERSE
  - (9) CHECK OVERFLOW
  - (10) CHECK LENGTH
  - (11) CHECK OMISSION
  - (12) CHECK MODULUS
  - (13) CHECK BTAB
  - (14) CHECK ADVANCE
  - (15) IFC

- (16) FILL
- (17) SHIFTED RIGHT
- (18) SHIFTED LEFT
- (19) INPUT HANDSCANNER
- (20) INPUT KANA-SHIFT
- (21) INPUT ID-CARD-READER
- (22) INPUT IO-OCR
- (23) WAIT
- (24) PREVIOUS ATTRIBUTE
- (25) OVER LINE
- (26) UNDER LINE
- (27) BOX
- (28) COLOR
- (29) VERTICAL LINE AT
- (30) OCCURS
- (31) OVER LINE TO
- (32) UNDER LINE TO

F D3004      **【Standard】**?句は指定できません.

- [原因]      (1) CLEAR SCREEN 句以外の CLEAR 句があるにもかかわらず, LINE 句が明にも暗にも指定されていない。
- (2) FILL IS INITIAL FILLER CHARACTER (IFC) 句が指定されたにもかかわらず INITIAL FILLER CHARACTER 句が指定されていない。
- (3) CLEAR 句が副画面に指定されている。
- (4) -CF オプション指定ありのとき, MIXED は指定できない。

- [埋め込み情報] (1) CLEAR
- (2) FILL
- (3) ATTRIBUTE
- (4) MIXED

F D3005      **【Standard】**数字項目以外に CHECK MODULUS 句は指定できません.

- [原因]      数字項目以外は, CHECK MODULUS 句に指定できない。

F D3006      **【Standard】**画面記述項に?句が指定されていません.

[原因] 画面記述項に必要な句が指定されていない。

- [埋め込み情報] (1) END STATUS  
(2) WINDOW HIGHT  
(3) WINDOW WIDTH  
(4) DISPLAY LINE  
(5) DISPLAY COLUMN  
(6) WINDOW STATUS

F D3007      **【Standard】**レベル番号が誤っています。

- [原因] (1) レベル番号が 01 から 49 の範囲でない。  
(2) レベル番号が整数でない。  
(3) 集団項目のすぐ次に書くレベル番号が、先の基本項目の属する集団のうちの1つのレベル番号と一致していない。

F D3008      **【Standard】**?句の整数が誤っています。

- [原因] (1) 数字定数が指定されているが整数でない。  
(2) LINE 句の整数が 1～24 の範囲でない。  
(3) CLEAR 句の整数が 1～24 の範囲でない。  
(4) COLUMN 句の整数が 1～80 の範囲でない。  
(5) OVER LINE TO 句の整数が 1～80 の範囲でない。  
(6) UNDER LINE TO 句の整数が 1～80 の範囲でない。  
(7) VERTICAL LINE AT 句の整数が 1～80 の範囲でない。  
(8) OCCURS 句の整数が 1～24 の値の範囲内でない。  
(9) WINDOW HIGHT の整数が 1～24 の範囲でない。  
(10) WINDOW WIDTH の整数が 1～80 の範囲でない。  
(11) DISPLAY LINE 句の整数が 1～24 の範囲でない。  
(12) DISPLAY COLUMN 句の整数が 1～80 の範囲でない。  
(13) 副画面の大きさが、親画面の大きさの範囲でない。

- [埋め込み情報] (1) LINE  
(2) CLEAR+  
(3) COLUMN  
(4) OVER LINE TO  
(5) UNDER LINE TO



- (6) VERTICAL LINE AT
- (7) OCCURS
- (8) WINDOW HIGHT
- (9) WINDOW WIDTH
- (10) DISPLAY LINE
- (11) DISPLAY COLUMN

F D3009      **【Standard】**CHECK MODULUS 句の定数が誤っています.

- [原因]      (1) CHECK MODULUS 句の定数が整数でない。  
(2) CHECK MODULUS 句の整数が 10 または 11 でない。

F D3010      **【Standard】**?句の定数が誤っています.

- [原因]      INITIAL FILLER CHARACTER (IFC) 句の定数が 1 文字の英数字定数でない。

- [埋め込み情報] (1) IFC  
(2) TITLE

F D3012      **【Standard】**01 記述項にデータ名がありません.

- [原因]      01 記述項にデータ名がない。

F D3013      **【Standard】**基本項目に必要な句が指定されていません.

- [原因]      (1) 書き方 3 の画面データ記述項(基本項目)に対して必要な PIC 記述が指定されていない。  
(2) 書き方 4 の画面データ記述項(基本項目)に対して必要な OVER LINE TO 句, UNDER LINE TO 句または VERTICAL LINE AT 句のいずれもが指定されていない。  
(3) 書き方 5 の画面データ記述項(基本項目)に対して必要な CLEAR 句が指定されていない。

F D3014      **【Standard】**CHECK MODULUS 句の指定された項目の桁数が 16 桁を超えています.

- [原因]      CHECK MODULUS 句の指定された項目の桁数が 16 桁を超えている。

F D3015      **【Standard】**出力領域に?句は指定できません.

[原 因]          句は出力領域に指定できない。

[埋め込み情報] (1) CHECK OVERFLOW  
(2) CHECK LENGTH  
(3) CHECK OMISSION  
(4) CHECK MODULUS  
(5) CHECK BTAB  
(6) CHECK ADVANCE  
(7) SECRET  
(8) SHIFTED RIGHT  
(9) SHIFTED LEFT  
(10) INPUT HANDSCANNER  
(11) INPUT KANA-SHIFT  
(12) INPUT ID-CARD-READER  
(13) INPUT IO-OCR  
(14) IFC  
(15) FILL

F D3016      **【Standard】**LINE 句は既に上位のレベルで指定されています.

[原 因]          LINE 句が指定された画面データ記述項を含む画面データ記述項にすでに  
LINE 句が指定されている。

F D3017      **【Standard】**画面節の後に画面記述項がありません.

[原 因]          SCREEN SECTION (画面節) の後に画面記述項がないか, 画面記述項の前に  
余分な記述がある。

W D3018      **【Standard】**必要な終止符がありません.

[原 因]          必要な終止符がない。

[コンパイラの処理]    あるものとみなす。

F D3019      **【Standard】**数字/数字編集項目以外に BLANK 句が指定されています.

[原因] 数字/数字編集項目以外に **BLANK** 句が指定されている。

F D3020     **【Standard】**JUSTIFIED/JUST 句が数字/編集項目に指定されています。

[原因] 数字項目, 数字編集項目, 英数字編集項目, 日本語編集項目に対して  
JUSTIFIED/JUST 句が指定された。

W D3023     **【Standard】**VALUE 句で値が 0 (ゼロ) 以外の桁落ちが発生します。

[原因] VALUE 句の値で 0 以外の桁落ちが発生する。  
[コンパイラの処理] MOVE 文の転記の規則と同様に切り捨てが行われる。

W D3024     **【Standard】**PICTURE 句と VALUE 句の符号の指定が矛盾しています。

[原因] PICTURE 句で符号の指定のない数字項目に対して符号付きの数字定数を  
指定した。  
[コンパイラの処理] 符号の指定を無視し, 絶対値を値とする。

W D3026     **【Standard】**VALUE 句で英数字定数/日本語定数の桁落ちが発生します。

[原因] VALUE 句で英数字定数/日本語定数の桁落ちが発生する。  
[コンパイラの処理] JUSTIFIED 句の指定に関係なく項目長に合わせて右側が切り捨てら  
れる。

F D3027     **【Standard】**`SD`の次に必要な画面記述名がありません。

[原因] SD の次に画面記述名がない。

F D3028     **【Standard】**?句の書き方が誤っています。

[原因] 主画面は画面節の先頭にただ 1 つしか書けない。  
[埋め込み情報] (1) END STATUS  
                  (2) LINE POSITION  
                  (3) COLUMN POSITION  
                  (4) BTAB POSITION

- (5) PARENT WINDOW
- (6) WINDOW STATUS
- (7) WINDOW HIGHT
- (8) WINDOW WIDTH
- (9) DISPLAY LINE
- (10) DISPLAY COLUMN
- (11) LINE
- (12) COLUMN
- (13) CLEAR
- (14) VALUE
- (15) IFC
- (16) FILL
- (17) INPUT
- (18) CHECK
- (19) COLOR
- (20) PREVIOUS ATTRIBUTE
- (21) OVER LINE
- (22) UNDER LINE
- (23) VERTICAL LINE AT
- (24) OVER LINE TO
- (25) UNDER LINE TO
- (26) OCCURS
- (27) TITLE
- (28) ATTRIBUTE
- (29) MIXED

F D3030      **【Standard】**FROM/INTO/USING 句の一意名の書き方が誤っています。

    [原因]      FROM/INTO/USING 句の一意名の書き方が誤っている。

F D3031      **【Standard】**画面節の VALUE 句に表意定数は指定できません。

    [原因]      画面節の VALUE 句に表意定数は指定できない。

F D3032      **【Standard】**PICTURE 句のない VALUE 句に英数字定数/日本語定数以外は指定できません。

〔原因〕 PICTURE 句のない VALUE 句に英数字定数/日本語定数以外は指定できない。

F D3033 【Standard】`SCREEN`の後に`SECTION`がありません。

〔原因〕 SCREEN の後に SECTION がない。

F D3034 【Standard】基本項目に対して LINE 句が上位のレベルにも指定されていません。

〔原因〕 LINE 句が書き方 3, 書き方 4 または書き方 5 の画面データ記述項(基本項目)に対して記述されてなく, かつその項目を含む書き方 2 および書き方 1 の画面データ記述項にも指定されていない。

F D3035 【Standard】画面データ記述項がありません。

〔原因〕 画面記述項の後に画面データ記述項が 1 つもない。

F D3036 【Standard】画面データ記述項の最初のレベル番号が 01 ではありません。

〔原因〕 画面記述項の直後の画面データ記述項のレベル番号が“01”でない。

F D3037 【Standard】書き方 1 の画面データ記述項に従属する記述項がありません。

〔原因〕 画面記述項の直後の画面データ記述項のレベル番号が“01”でない。

F D3038 【Standard】PICTURE 記述は既に指定されています。

〔原因〕 PICTURE 記述はすでに指定されている。

F D3039 【Standard】集団項目に PICTURE 記述は指定できません。

〔原因〕 集団項目に PICTURE 記述は指定できない。

F D3040 【Standard】レベル番号が 2 桁を超えています。

【原因】 レベル番号が 2 桁を超えている。

F D3041 【Standard】ゼロ抑制指定(`\*`)の項目に BLANK 句が指定されています。

【原因】 星印によるゼロ抑制指定(\*)の項目に BLANK ZERO 句が指定されている。

F D3042 【Standard】符号のある数字項目に BLANK 句が指定されています。

【原因】 符号のある数字項目に BLANK ZERO 句が指定されている。

F D3043 【Standard】英数字項目以外に?句が指定されています。

【原因】 (1) 英数字項目以外に INPUT KANA-SHIFT 句が指定されている。

(2) 英数字項目以外に MIXED 句が指定されている。

【埋め込み情報】 (1) INPUT KANA-SHIFT

(2) MIXED

F D3044 【Standard】項類と VALUE 句の定数が矛盾しています。

【原因】 項類と VALUE 句の定数が次の表の条件に違反している。

| 条件 | 項類      | 定数    |
|----|---------|-------|
| 1  | 数字項目    | 数字定数  |
| 2  | 英字項目    | 英数字定数 |
|    | 英数字項目   |       |
|    | 英数字編集項目 |       |
|    | 数字編集項目  |       |
| 3  | 日本語項目   | 日本語定数 |
|    | 日本語編集項目 |       |

F D3045 【Standard】?句は入力領域に対して指定できません。

【原因】 句は入力領域に指定できない。

【埋め込み情報】 (1) JUSTIFIED/JUST

(2) WAIT

F D3046 【Standard】書き方 4 の基本記述項に?句は指定できません。

[原因] 句は書き方 4 の基本記述項に指定できない。

- [埋め込み情報] (1) COLUMN  
(2) BLANK  
(3) JUSTIFIED/ JUST  
(4) SHIFTED RIGHT  
(5) SHIFTED LEFT  
(6) BUZZER  
(7) SECRET  
(8) BLINK  
(9) HIGH INTENSITY  
(10) REVERSE  
(11) CHECK OVERFLOW  
(12) CHECK LENGTH  
(13) CHECK OMISSION  
(14) CHECK MODULUS  
(15) CHECK BTAB  
(16) CHECK ADVANCE  
(17) INPUT HANDSCANNER  
(18) INPUT KANA-SHIFT  
(19) INPUT ID-CARD-READER  
(20) INPUT IO-OCR  
(21) WAIT  
(22) BOX  
(23) PREVIOUS ATTRIBUTE  
(24) OVER LINE  
(25) UNDER LINE  
(26) IFC  
(27) FILL  
(28) COLOR

F D3047      **【Standard】**書き方 3 の基本記述項に OCCURS 句または OVER LINE TO/UNDER LINE TO/VERTICAL LINE AT 句は指定できません。

[原因]      書き方 3 の基本記述項に OCCURS, OVER LINE TO/UNDER LINE TO/VERTICAL LINE AT 句は指定できない。

F D3049      **【Standard】**?句の整数が COLUMN 句の整数より小さい.

〔原因〕      句の整数が COLUMN 句の整数より小さい。

〔埋め込み情報〕 (1) VERTICAL LINE AT  
(2) OVER LINE TO  
(3) UNDER LINE TO

F D3050      **【Standard】**書き方 4 の COLUMN 句にデータ名は指定できません.

〔原因〕      書き方 4 の COLUMN 句にデータ名は指定できない。

F D3052      **【Standard】**PICTURE 句の指定が誤っています.

〔原因〕      (1) 数字項目/数字編集項目に対する PICTURE 句の文字列に“P”が指定されている。  
(2) PICTURE 句の文字列にブールデータ項目が指定されている。

F D3053      **【Standard】**画面データ項目の桁数が大きすぎます.

〔原因〕      画面データ項目の桁数が大きすぎる。

F D3054      **【Standard】**INTO/USING 句に対して編集項目は指定できません.

〔原因〕      INTO/USING 句に編集項目は指定できない。

F D3055      **【Standard】**?句は書き方 5 に指定できません.

〔原因〕      句は書き方 5 に指定できない。

〔埋め込み情報〕 (1) COLUMN  
(2) BLANK  
(3) JUSTIFIED/JUST  
(4) SHIFTED RIGHT  
(5) SHIFTED LEFT  
(6) BUZZER  
(7) SECRET



- (8) BLINK
- (9) HIGH INTENSITY
- (10) REVERSE
- (11) CHECK OVERFLOW
- (12) CHECK LENGTH
- (13) CHECK OMISSION
- (14) CHECK MODULUS
- (15) CHECK BTAB
- (16) CHECK ADVANCE
- (17) INPUT HANDSCANNER
- (18) INPUT KANA-SHIFT
- (19) INPUT ID-CARD-READER
- (20) INPUT IO-OCR
- (21) WAIT
- (22) PREVIOUS ATTRIBUTE
- (23) OVER LINE
- (24) UNDER LINE
- (25) BOX
- (26) IFC
- (27) FILL
- (28) COLOR
- (29) OCCURS
- (30) ATTRIBUTE
- (31) MIXED

F D3057      **【Standard】**BLANK WHEN ZERO 句は FROM 句以外に対して指定できません。

[原 因]      BLANK WHEN ZERO 句は FROM 句以外に対して指定できない。

F D3059      **【Standard】**PREVIOUS ATTRIBUTE 句は?句と同時に指定できません。

[原 因]      主画面は画面節の先頭にただ 1 つしか書けない。

[埋め込み情報] (1) SECRET  
(2) BLINK  
(3) HIGH INTENSITY  
(4) REVERSE

F D3060      **【Standard】**SCREEN SECTION が 2 度以上指定されています.

〔原因〕      画面節が複数指定されている。

F D3061      **【Standard】**データ部中の節の指定順序が正しくありません.

〔原因〕      データ部中の節の指定順序が誤っている。

F D3101      **【Standard】**画面機能は使えません.

〔原因〕      COBOL GUI アプリケーション作成時, 画面機能は使用できない。

F D3200      **【Standard】**副画面の定義数が制限を超えています.

〔原因〕      副画面の定義数が 999 個を超えた。

F D3238      **【Standard】**画面節の定義が誤っています.

〔原因〕      (1) 最も外側のプログラム定義以外のソース単位では画面節は記述できない。  
(2) 画面節を記述したプログラム定義のプログラム名段落に RECURSIVE 句を指定してはならない。

F D3239      **【Standard】**画面節が指定されているプログラムに, -Cj, -CU オプションが指定されています.

〔原因〕      画面節を記述した COBOL ソースに対して, -Cj, -CU オプションが指定できません。

F D3240      **【Standard】**PARENT WINDOW 句に必要な画面記述名がありません.

〔原因〕      (1) PARENT WINDOW 句の画面記述名が定義されていないか, 誤っている。

F E0001      USE 手続きに?句が複数指定されています.

**【Enterprise for Linux】**THE USE ERROR PROCEDURE WITH ? CLAUSE IS

DUPLICATED.

[原因] 指定されている **USE** 手続きが複数ある。

[埋め込み情報] (1) **INPUT**  
(2) **OUTPUT**  
(3) **I-O**  
(4) **EXTEND**

F E0003      **USE** 手続きが指定されていません。  
              【Enterprise for Linux】THE USE ERROR PROCEDURE IS NOT SPECIFIED.

[原因] 宣言部分に **USE** 手続きが指定されていない。

F E0004      関数名でなければなりません。  
              【Enterprise for Linux】FUNCTION-NAME IS REQUIRED.

[原因] 予約語「**FUNCTION**」の直後が関数名でない。

W E0005      **USE** 手続きに同じファイルが指定されています。  
              【Enterprise for Linux】THE FILE NAME IS DUPLICATED IN THE USE ERROR  
PROCEDURE.

[原因] **USE** 手続きに同じファイルが指定されている。

[コンパイラの処理] そのまま翻訳を続行する。

W E0007      区分番号の指定が誤っています。  
              【Enterprise for Linux】THE SEGMENT NUMBER IS OUTSIDE THE RANGE.

[原因] (1) 区分番号が 0 から 99 までの範囲でない(宣言節以外)。  
(2) 区分番号が 0 から 49 までの範囲でない(宣言節)。

[コンパイラの処理] 0 とみなす。

F E0008      数字定数の長さが 18 桁を超えています。  
              【Enterprise for Linux】THE NUMERIC LITERAL EXCEEDS 18 DIGITS.

[原因] 数字定数の長さが 18 桁を超えている。

W E0009      必要な終止符がありません。  
              【Enterprise for Linux】THE TERMINATOR `PERIOD` IS MISSING.

    [原因]          必要な終止符がない。  
    [コンパイラの処理]    終止符があるものとみなす。

F E0010      ?文の英数字定数に許されない文字が含まれています。  
              【Enterprise for Linux】A NONNUMERIC LITERAL IN THE ? STATEMENT CONTAINS  
              AN UNACCEPTABLE CHARACTER.

    [原因]          英数字定数に指定以外の文字が含まれている。  
    [埋め込み情報] (1) CALL  
                      (2) CANCEL

F E0012      引数の個数が誤っています。  
              【Enterprise for Linux】NUMBER OF ARGUMENT IS ILLEGAL.

    [原因]          引数の個数が定められた個数の範囲外である。

F E0013      添字に語 ALL は指定できません。  
              【Enterprise for Linux】DESCRIBING `ALL` IS NOT ALLOWED.

    [原因]          表を指定できない引数に対して表を指定した。

F E0014      この関数は指定できません。  
              【Enterprise for Linux】TYPE OF FUNCTION IS ILLEGAL.

    [原因]          その場所に対して許されない型を持つ関数を指定した。

F E0015      引数の型が誤っています。  
              【Enterprise for Linux】TYPE OF THE ARGUMENT IS ILLEGAL.

    [原因]          引数の型がその関数に対して許されない型である。

F E0016      この関数に対して部分参照は指定できません。

**【Enterprise for Linux】REFERENCE MODIFICATION IS NOT ALLOWED.**

〔原 因〕 組込み関数の時、英数字型でない関数に対して部分参照を指定した。

F E0017 引数が範囲外です。

**【Enterprise for Linux】VALUE OF THE ARGUMENT IS OUT OF BOUNDS.**

〔原 因〕 引数が定数で記述されているとき、その値が範囲外である。

F E0018 かっこの対応がとれていません。

**【Enterprise for Linux】LEFT AND RIGHT PARENTHESES ARE UNEQUAL IN NUMBER.**

〔原 因〕 関数中のかっこの対応が合わない。

F E0020 範囲終了子の記述が誤っています。

**【Enterprise for Linux】WRONG SCOPE TERMINATOR DESCRIPTION.**

〔原 因〕 (1) 対応のとれない範囲終了子を記述した。  
(2) 複数の語からなる範囲終了子に構文誤りがある。

F E0021 選択主体と選択対象の数が一致しません。

**【Enterprise for Linux】THE NUMBER OF SELECTION OBJECTS MUST BE EQUAL TO THE NUMBER OF SELECTION SUBJECTS.**

〔原 因〕 選択主体の **ALSO** の数と選択対象の **ALSO** の数が合わない。

F E0022 **EXIT PERFORM** 文はうち **PERFORM** 文の無条件文内にしか指定できません。

**【Enterprise for Linux】THE EXIT PERFORM STATEMENT MUST BE SPECIFIED IN THE IN-LINE PERFORM STATEMENT.**

〔原 因〕 **EXIT PERFORM** 文の記述がうち **PERFORM** の無条件文の中でない。

F E0024 **【Enterprise for Windows】【Standard】**定数名`?`の **LENGTH OF** 句が一意ではありません。

〔原 因〕 (1) 定数名記述項の **LENGTH OF** 句で指定したデータ名が一意でない。

(2) 定数名記述項の **LENGTH OF** 句で指定したデータ名が作業場所節またはファイル節で定義されていない。

(3) 定数名記述項の **LENGTH OF** 句で指定したデータ名の次元が **0** 以外である。

**F E0025** 例外節は既に指定されています。

**【Enterprise for Linux】THE EXCEPTION SECTION IS DUPLICATED.**

[原因] 手続き部に例外節が複数書かれている。

[コンパイラの処理] 次の文まで読みとばす。

**F E0026** 含まれるプログラムの **USE** 文に **EXCEPTION** 句が指定されています。

**【Enterprise for Linux】THE EXCEPTION CLAUSE CANNOT BE DESIGNATED IN THE PROGRAM CONTAINED WITHIN ANOTHER PROGRAM.**

[原因] 含まれるプログラムの **USE** 文に **EXCEPTION** 句が指定されている。

[コンパイラの処理] 次の文まで読みとばす。

**F E0027** 例外節内に ? 文は指定できません。

**【Enterprise for Linux】A ? STATEMENT CANNOT BE SPECIFIED WITHIN A EXCEPTION SECTION.**

[原因] **EXIT PROGRAM** または **GOBACK** 文が例外節に書かれている。

[埋め込み情報] (1) **EXIT PROGRAM**

(2) **GOBACK**

[コンパイラの処理] 次の文まで読みとばす。

**F E0028** ? 句の指定が誤っています。

**【Enterprise for Linux】THE ? CLAUSE CONTAINS INVALID DESCRIPTION.**

[原因] (1) **PROCEDURE DIVISION USING** 句にデータ名を重複して指定している。

(2) **PROCEDURE DIVISION RETURNING** 句のデータ名が **USING** 句のデータ名と重複している。

(3) **【Standard】**実行用計算機段落の **WITH RETURN VALUE** 句を指定しているのに、**PROCEDURE DIVISION RETURNING** 句を指定している。

[埋め込み情報] (1) **USING**

(2) RETURNING

F E0029     ?文は GLOBAL 指定の USE 手続き中に指定できません。  
              【Enterprise for Linux】THE ? STATEMENT CANNOT BE SPECIFIED IN A USE  
              PROCEDURE WITH GLOBAL CLAUSE.

[原因]       (1) EXIT PROGRAM 文は, GLOBAL 句が指定された宣言手続きに書いてはな  
                  らない。  
              (2) GOBACK 文は, GLOBAL 句が指定された宣言手続きに書いてはならな  
                  い。

[埋め込み情報] (1) EXIT PROGRAM  
                  (2) GOBACK

[コンパイラの処理] 次の文まで読みとばす。

F E0030     ?文は指定できません。  
              【Enterprise for Linux】THE ? STATEMENT CANNOT BE SPECIFIED.

[原因]       次の手続き文は指定できない。  
              (1) DISABLE  
              (2) ENABLE  
              (3) RECEIVE  
              (4) SEND  
              (5) GENERATE  
              (6) INITIATE  
              (7) SUPPRESS  
              (8) TERMINATE  
              (9) APPLY  
              (10) COMMIT  
              (11) ROLLBACK  
              (12) SCRATCH  
              (13) SELECT  
              (14) SUBSTRING

[埋め込み情報] (1) DISABLE  
                  (2) ENABLE  
                  (3) RECEIVE  
                  (4) SEND

- (5) GENERATE
- (6) INITIATE
- (7) SUPPRESS
- (8) TERMINATE
- (9) APPLY
- (10) COMMIT
- (11) ROLLBACK
- (12) SCRATCH
- (13) SELECT
- (14) SUBSTRING

F E0031      ?句は指定できません。  
【Enterprise for Linux】THE ? CLAUSE CANNOT BE SPECIFIED.

[原因]          次の句は指定できない。

- (1) TERMINAL

[埋め込み情報] (1) TERMINAL

F E0040      PERFORM 手続き文の AFTER 句が 7 個以上指定されています。  
【Enterprise for Linux】THE NUMBER OF AFTER CLAUSE IN THE PERFORM  
STATEMENT EXCEEDS THE LIMIT (7).

[原因]          書き方 4 の PERFORM 文に AFTER 句が 7 個以上指定されている。

F E0041      ? 文の SEQUENCE 句で符号系名の指定が正しくありません。  
【Enterprise for Linux】THE OPERAND OF THE COLLATING SEQUENCE CLAUSE IN  
THE ? STATEMENT IS ILLEGAL.

[原因]          SORT 文または MERGE 文 COLLATING SEQUENCE 句の符号系名の指定が正しくない。

- [埋め込み情報] (1) SORT
- (2) MERGE

F E0042      CANCEL 文を使用しているプログラムに、-Ci オプションが指定されています。  
【Enterprise for Linux】THE -Ci COMPILE OPTION CANNOT BE DESIGNATED IN  
THE PROGRAM WITH CANCEL STATEMENT.



[原因] CANCEL 文があり、かつ-Ci オプションが指定されている。

F E0060 データ例外節は既に指定されています。  
【Enterprise for Linux】THE DATA ERROR SECTION IS DUPLICATED.

[原因] 1つのプログラム内でデータ例外節を二度以上指定している。

F E0061 含まれるプログラムの USE 文に DATA ERROR 句が指定されています。  
【Enterprise for Linux】THE DATA ERROR CLAUSE CANNOT BE DESIGNATED IN THE PROGRAM CONTAINED WITHIN ANOTHER PROGRAM.

[原因] 含まれるプログラムの USE 文に DATA ERROR 句を指定している。

F E0062 データ例外節内に? 文は指定できません。  
【Enterprise for Linux】A ? STATEMENT CANNOT BE SPECIFIED WITHIN A DATA ERROR SECTION.

[原因] データ例外節に EXIT PROGRAM または GOBACK 文を記述している。

[埋め込み情報] (1) EXIT PROGRAM  
(2) GOBACK

F E0063 データ例外節以外に? 文は指定できません。  
【Enterprise for Linux】A ? STATEMENT CAN BE SPECIFIED ONLY FOR DATA ERROR SECTION.

[原因] データ例外節以外で ALTER (データ例外変更), DISPLAY (データ例外表示), MOVE (データ例外初期化) 文を記述している。

[埋め込み情報] (1) ALTER ERROR  
(2) DISPLAY ERROR  
(3) MOVE ERROR

F E0080 メインプログラムの PROCEDURE DIVISION に? 句が指定されています。  
【Enterprise for Linux】THE ? CLAUSE CANNOT BE DESIGNATED IN THE PROCEDURE DIVISION OF MAIN PROGRAM.

- [原因] (1) メインプログラムの **PROCEDURE DIVISION** に **USING** 句が指定されている。  
(2) メインプログラムの **PROCEDURE DIVISION** に **RETURNING** 句が指定されている。

- [埋め込み情報] (1) **USING**  
(2) **RETURNING**

W E0102      **CORRESPONDING/CORR** 指定のある文で `?` の中に対応する項目がありません。  
              【Enterprise for Linux】**NO CORRESPONDING ITEM IS FOUND IN THE GROUP ITEM**  
              **WHOSE NAME IS `?`.**

- [原因]        **CORRESPONDING** 指定のある文の作用対象の集団項目同士の中に対応する項目が 1 つもない。

[埋め込み情報] 利用者定義語 (受け取り側の集団項目名)

[コンパイラの処理] 対応する項目のない集団項目間の転記は行わない。

F E0103      `?` は **CORRESPONDING/CORR** 指定のある文の作用対象中で多重定義されています。  
              【Enterprise for Linux】**THE DATA NAME `?` IS NOT UNIQUE IN THE GROUP ITEM.**

- [原因]        **CORRESPONDING/CORR** 指定のある文の作用対象中で多重定義されている。

[埋め込み情報] 利用者定義語

F E0104      **CORRESPONDING/CORR** 指定のある文の作用対象が誤っています。  
              【Enterprise for Linux】**THE STATEMENT WITH CORRESPONDING MUST HAVE A**  
              **GROUP ITEM.**

- [原因]        **CORRESPONDING** 指定のある文の作用対象が誤っている。

- (1) 集団項目でない。  
(2) 部分参照が記述されている。  
(3) 【Standard】画面節のデータが記述されている。  
(4) 【Standard】画面記述名が記述されている。  
(5) 【Enterprise for Windows】【Standard】定数名である。

F E0106      `?` は定義されていません。  
              【Enterprise for Linux】**`?` IS UNDEFINED.**

[原因] 利用者定義語が定義されていない。

[埋め込み情報] 未定義な利用者定義語

F E0107 `?`は一意ではありません。

【Enterprise for Linux】`?` IS NOT UNIQUE.

[原因] (1) 利用者の定義する語が一意でない。

(2) 一意参照の修飾が完全でない。

[埋め込み情報] 利用者定義語

F E0108 `?`の修飾が制限を超えています。

【Enterprise for Linux】THE QUALIFIERS FOR THE DATA NAME `?` EXCEEDS THE LIMIT (50).

[原因] データの修飾が 50 を超えている。

[埋め込み情報] データ名 (修飾される語)

F E0109 `?`は二重に定義されています。

【Enterprise for Linux】THE DEFINITION OF `?` IS DUPLICATED.

[原因] (1) 利用者語が二重に定義されている。

(2) ACCEPT (利用者制御変数の取り込み) 文または DISPLAY (利用者制御変数の設定) 文で呼び名が一意でない。

[埋め込み情報] 利用者定義語

F E0110 `?`に対して修飾はできません。

【Enterprise for Linux】THE USER DEFINED WORD `?` CANNOT BE QUALIFIED.

[原因] 修飾できない利用者定義語を修飾しようとしている。

(1) ファイル名

(2) 指標名

(3) 節名

(4) 呼び名

(5) 独立項目のデータ名

(6) 階層構造の最上位のデータ名

[埋め込み情報] 利用者定義語

F E0120 INITIALIZE 文の作用対象の属性が誤っています。  
【Enterprise for Linux】THE OPERAND OF THE INITIALIZE STATEMENT IS  
INCORRECT.

[原 因] INITIALIZE 文の作用対象は次のものであってはならない。

- (1) 指標データ項目
- (2) レベル番号 66 で定義された集団項目
- (3) 88 条件名
- (4) 【Enterprise for Windows】【Standard】定数名

F E0121 INITIALIZE 文の初期化指定が誤っています。  
【Enterprise for Linux】THE OPERAND OF THE REPLACING CLAUSE CONFLICTS  
WITH THE ONE OF THE BY CLAUSE.

[原 因] (1) INITIALIZE 文の REPLACING 句の作用対象と BY 句の作用対象の属性  
が合わない。

(2) 【Enterprise for Windows】【Standard】INITIALIZE 文の REPLACING  
句に指定した定数名が転記の規則に違反している。

F E0122 INITIALIZE 文の REPLACING 句の属性が矛盾しています。  
【Enterprise for Linux】THE OPERAND OF THE REPLACING CLAUSE IS ILLEGAL.

[原 因] INITIALIZE 文の初期化作用対象の基本項目が REPLACING 句で示した項  
目で初期化ができない(転記できない)。

W E0123 初期化の対象となるものが 1 つも指定されていません。  
【Enterprise for Linux】NO INITIALIZING ITEM IS FOUND.

[原 因] INITIALIZE 文の集団項目の要素に REPLACING 句で指定した要素のものが 1 つもない。

[コンパイラの処理] この INITIALIZE 文は何も初期化しない手続き文とみなす。

F E0124 REPLACING BY 句の作用対象の属性が誤っています。  
【Enterprise for Linux】THE OPERAND OF THE REPLACING BY CLAUSE IS ILLEGAL.

[原 因] INITIALIZE 文の REPLACING BY 間に指定したものが次のものであってはならない。

- (1) 指標データ項目
- (2) レベル番号 66 で定義された集団項目
- (3) 88 条件名

W E0127 INITIALIZE 文の作用対象に可変繰返しデータ項目を含む集団項目が指定されています。

【Enterprise for Linux】THE OPERAND OF THE INITIALIZE STATEMENT IS THE GROUP ITEM INCLUDING VARIABLE OCCURRENCE DATA ITEMS.

[原 因] INITIALIZE 文の作用対象が可変繰返しデータ項目を含む集団項目である。

[コンパイラの処理] 表の最大反復回数分を初期化するものとみなす。

F E0128 SHIFT-CODE 句の指定された?文の一意名中に許されない記述があります。

【Enterprise for Linux】THE ? STATEMENT IDENTIFIER OF THE FILE SPECIFIED IN THE SHIFT-CODE PHRASE CONTAINS AN UNACCEPTABLE DESCRIPTION.

[原 因] APPLY SHIFT-CODE 句の指定されたファイルへの WRITE 文の FROM 句の一意名中に SYNCHRONIZED 句が指定された。

[埋め込み情報] WRITE

F E0129 CODE-SET 句の指定された?文の一意名中に許されない記述があります。

【Enterprise for Linux】THE ? STATEMENT IDENTIFIER OF THE FILE SPECIFIED IN THE CODE-SET PHRASE CONTAINS AN UNACCEPTABLE DESCRIPTION.

[原 因] (1) CODE-SET 句に関連するファイルの REWRITE 文の FROM 句の一意名中に DEPENDING ON 句, または SYNCHRONIZED 句, または表示用以外の項目, または SIGN IS SEPARATE 句のない符号付き数字項目が記述されている。

(2) CODE-SET 句に関連するファイルの WRITE 文の FROM 一意名中に DEPENDING ON 句, または SYNCHRONIZED 句, または表示用以外の項目, または SIGN IS SEPARATE 句のない符号付き数字項目が記述されている。

(3) CODE-SET 句に関連するファイルの READ 文の INTO 一意名中に

DEPENDING ON 句, または SYNCHRONIZED 句, または表示用以外の項目, または SIGN IS SEPARATE 句のない符号付き数字項目が記述されている。

- (4) CODE-SET 句に関連するファイルの REWRITE 文の FROM 句の一意名が再定義項目である。
- (5) CODE-SET 句に関連するファイルの WRITE 文の FROM 句の一意名が再定義項目である。
- (6) CODE-SET 句に関連するファイルの READ 文の INTO 句の一意名が再定義項目である。

[埋め込み情報] (1) REWRITE

(2) WRITE

(3) READ

F E0130      SHIFT-CODE 句の指定されたファイルへの?文のレコードが複雑です.  
 【Enterprise for Linux】THE RECORD OF ? STATEMENT OF THE FILE SPECIFIED  
 IN THE SHIFT-CODE PHRASE IS TOO COMPLEX.

[原 因]      APPLY SHIFT-CODE 句の指定されたファイルへの WRITE 文のレコードまたは一意名が複雑すぎる。

[埋め込み情報] WRITE

F E0131      CODE-SET 句の指定されたファイルへの?文のレコードが複雑です.  
 【Enterprise for Linux】THE RECORD OF ? STATEMENT OF THE FILE SPECIFIED  
 IN THE CODE-SET PHRASE IS TOO COMPLEX.

[原 因]      (1) CODE-SET 句が指定されたファイルへの READ 文または REWRITE 文または WRITE 文に指定したレコードに対するコード変換のための情報領域が 64K 以上になった。  
 (2) CODE-SET 句が指定されたファイルへの REWRITE 文または WRITE 文の FROM 句の一意名に対するコード変換のための情報領域が 64K 以上になった。

[埋め込み情報] (1) REWRITE

(2) WRITE

(3) READ

F E0132      FROM 句のある WRITE 文のレコード名と一意名のどちらか一方が集団項目でありま

せん.

【Enterprise for Linux】THE IDENTIFIER AFTER FROM PHRASE MUST NOT BE ELEMENTARY ITEM WHEN RECORD NAME IS ELEMENTARY ITEM.

〔原 因〕 APPLY SHIFT-CODE 句が指定されたファイルへの FROM 句指定ありの WRITE 文でレコード名と一意名はどちらか一方が集団項目でない。

F E0133 ?句の指定されたファイルに対する FROM 句の一意名にレベル番号 66 のデータ項目が指定されています。

【Enterprise for Linux】THE FROM IDENTIFIER OF THE ? PHRASE CANNOT BE 66 RENAMES ITEM.

〔原 因〕 WRITE 文または REWRITE 文の FROM 句の一意名にレベル番号 66 のデータ項目を指定した。

〔埋め込み情報〕 (1) CODE-SET  
(2) APPLY SHIFT-CODE

F E0188 SHIFT-CODE 指定のファイルの FROM 一意名が誤っています。

【Enterprise for Linux】THE FROM IDENTIFIER OF THE SHIFT-CODE APPLIED FILE IS WRONG.

〔原 因〕 (1) APPLY SHIFT-CODE 句が指定されたファイルに対する WRITE 文の FROM 一意名が誤っている。  
(a) OCCURS DEPENDING ON 句が指定されている。  
(b) SYNCHRONIZED 句が指定されている。  
(c) 再定義項目である。  
(d) 再定義される項目より再定義する項目が大きい。  
(e) 【Standard】用途が表示用以外の項目が指定されている。

F E0189 SHIFT-CODE 指定のファイルの FROM 一意名の長さが誤っています。

【Enterprise for Linux】THE LENGTH OF FROM IDENTIFIER OF THE SHIFT-CODE APPLIED FILE IS WRONG.

〔原 因〕 (1) APPLY SHIFT-CODE 句が指定されたファイルの FROM 一意名が次のいずれかである。  
(a) FROM 一意名の長さが、ファイルの最大レコード長より大きい。

(b) **APPLY SHIFT-CODE** 句が指定されたファイルで, **FROM** 一意名のシフトコードやプリンタ制御コードを挿入した長さが, **512** バイトを超えている。

**F E0190**      **COLUMN NUMBER** 句の整数が ? を超えています.  
【Enterprise for Linux】**THE INTEGER IN THE COLUMN NUMBER CLAUSE IS GREATER THAN ?.**

[原因]      【Enterprise】**APPLY SHIFT-CODE** 句に **PPR-CONTROL-3** が指定された **SYSOUT** ファイルに対して, **COLUMN NUMBER** 句で指定した整数が **208** を超えている。

[埋め込み情報] (1) 208

**F E0200**      プログラム定義やプログラム原型定義の引数または返却項目が矛盾しています.  
【Enterprise for Linux】**CONFLICTED ARGUMENT OR RETURNING ITEM OF PROGRAM/PROGRAM PROTOTYPE DEFINITION.**

[原因]      同名のプログラム原型定義が既に存在し, かつ仮パラメータと返却項目の記述がそれぞれ一致でない。

**F E0201**      原型の手続き部の書き方が誤っています.  
【Enterprise for Linux】**PROCEDURE DIVISION OF PROTOTYPE IS WRONG.**

[原因]      プログラム原型および関数原型の時, 手続き部の内容は手続き部見出しだけでない。

**F E0202**      プログラム原型名が誤っています.  
【Enterprise for Linux】**PROGRAM PROTOTYPE NAME IS INCORRECT.**

[原因]      (1) **CALL** 文で指定したプログラム原型名はリポジトリ段落で指定されていない。  
              (2) **CALL** 文で指定したプログラム原型名は二重定義されている。

**F E0204**      ?の指定箇所が誤っています.  
【Enterprise for Linux】**? IS SPECIFIED IN WRONG POSITION.**



- [原因] (1) EXIT PROGRAM がプログラムの手続き部以外に指定されている。  
(2) EXIT SECTION が節中以外に指定されている。  
(3) EXIT FUNCTION が関数の手続き部以外に指定されている。

- [埋め込み情報] (1) EXIT PROGRAM  
(2) EXIT SECTION  
(3) EXIT FUNCTION

F E0205 【Enterprise】

マルチスレッドで使用できない手続き文があります。

【Enterprise for Linux】NOT USABLE PROCEDURE STATEMENT IN MULTI-THREAD IS FOUND.

- [原因] (1) -MT オプション指定ありの場合、次のいずれかの手続き文を記述している。  
(a) CANCEL  
(b) CALL 一意名  
(2) -MT オプション指定ありの場合、メインプログラム以外に次のいずれかの手続き文を記述している。  
(a) STOP RUN  
(b) STOP ABORT

F E0206 関数定義や関数原型定義には、RETURNING を指定しなければなりません。

【Enterprise for Linux】RETURNING PHRASE MISSING IN FUNCTION/FUNCTION PROTOTYPE DEFINITION.

- [原因] 関数定義や関数原型定義に、RETURNING が指定されていない。

F E0207 関数定義や関数原型定義の引数または返却項目が矛盾しています。

【Enterprise for Linux】CONFLICTED ARGUMENT OR RETURNING ITEM OF FUNCTION/FUNCTION PROTOTYPE DEFINITION.

- [原因] 同じ名前の関数定義や関数原型定義が既に外部リポジトリに存在し、かつ仮パラメータと返却項目の記述がそれぞれ一致でない。

F E0209 上位クラスで定義されたメソッドと引数が適合していません。CODE=?, NO=?.

**【Enterprise for Linux】ARGUMENT DO NOT CONFORM TO METHOD DEFINED IN SUPERCLASS. CODE=?, NO=?.**

[原因] 上位クラスのメソッドと引数が不整合である。

[埋め込み情報] 埋め込み情報 1

不適合となった条件の識別番号

- (1) 1(引数の数が異なる)
- (2) 2(引数の BY REFERENCE や BY VALUE の指定が異なる)
- (3) 3(字類または項類の組み合わせが不正)
- (4) 4(同じクラス名を伴ったオブジェクト参照でない)
- (5) 5(FACTORY 指定の有無が異なる)
- (6) 6(PICTURE, USAGE, SIGN, JUSTIFIED, BLANK WHEN ZERO が異なる)
- (7) 7(通貨文字列が異なる)
- (8) 8(環境部の DECIMAL-POINT IS COMMA 指定の有無が異なる)

埋め込み情報 2

不適合となった引数が何番目かの数字

F E0210 上位クラスで定義されたメソッドと返却項目が適合していません。CODE=?.

**【Enterprise for Linux】RETURNING ITEM DO NOT CONFORM TO METHOD DEFINED IN SUPERCLASS. CODE=?.**

[原因] 上位クラスのメソッドと返却項目が不整合である。

[埋め込み情報] 不適合となった条件の識別番号

- (1) 1(返却項目の有無が異なる)
- (2) 3(字類または項類の組み合わせが不正)
- (3) 4(同じクラス名を伴ったオブジェクト, またはサブクラス名指定のオブジェクト参照でない)
- (4) 5(FACTORY 指定の有無が異なる)
- (5) 6(PICTURE, USAGE, SIGN, JUSTIFIED, BLANK WHEN ZERO が異なる)
- (6) 7(通貨文字列が異なる)
- (7) 8(環境部の DECIMAL-POINT IS COMMA 指定の有無が異なる)

F E0211 一意名?は指定できません。

**【Enterprise for Linux】IDENTIFIER ? CAN NOT SPECIFY.**

〔原因〕 次の一意名は指定できません。

- 〔埋め込み情報〕 (1) SELF  
(2) SUPER  
(3) NULL

F E0212 オブジェクトビューは指定できません。

【Enterprise for Linux】OBJECT VIEW CAN NOT SPECIFY.

〔原因〕 次の項目以外にオブジェクトビューを指定している。

- (1) INVOKE 文の呼び出し対象  
(2) INVOKE 文の USING 引数 (ただし, BY REFERENCE を除く)  
(3) SET 文の送り出し側作用対象  
(4) オブジェクトプロパティの一意名  
(5) CALL 文の書き方 3 の USING 引数 (ただし, BY REFERENCE を除く)  
(6) EVALUATE 文の選択主体と選択対象  
(7) IF 文, PERFORM 文, SEARCH 文の条件式

F E0213 関数は指定できません。

【Enterprise for Linux】FUNCTION CAN NOT SPECIFY.

〔原因〕 次の項目は指定できない。

- (1) 関数一意名

F E0215 プロパティ名 `?` は指定できません。

【Enterprise for Linux】PROPERTY-NAME ? CAN NOT SPECIFY.

〔原因〕 プロパティ名は指定できない。

〔埋め込み情報〕 利用者定義語

F E0216 `?` には返却項目はありません。

【Enterprise for Linux】? HAS NO RETURNING ITEM.

〔原因〕 返却項目なしのプログラムやメソッドの CALL 文, INVOKE 文に返却項目が指定されている。

〔埋め込み情報〕 利用者定義語

F E0217      ファクトリ/インスタンス定義の手続き部の書き方が誤っています。  
              【Enterprise for Linux】WRONG PROCEDURE DIVISION IN FACTORY/INSTANCE  
              DEFINITION.

- 【原因】            (1) ファクトリ定義, インスタンス定義の手続き部の見出しに仮パラメータまたは返却項目を指定している。  
                      (2) ファクトリ定義, インスタンス定義の場合, 手続き部に手続き部見出しとメソッド定義以外の内容を指定している。

F E0218      `?'は指定できません。  
              【Enterprise for Linux】? CAN NOT SPECIFY.

- 【原因】            次の利用者定義語は指定できない。  
                      (1) クラス名  
                      (2) プログラム原型名

【埋め込み情報】    利用者定義語

F E0220      プロパティメソッドの引数あるいは返却項目の指定が誤っています。  
              【Enterprise for Linux】WRONG ARGUMENT OR RETURNING ITEM OF PROPERTY  
              METHOD.

- 【原因】            (1) SET プロパティメソッドの定義が次のいずれかである。  
                      (a) USING 句のデータが 1 つでない。  
                      (b) RETURNING 句の指定がある。  
                      (2) GET プロパティメソッドの定義が次のいずれかである。  
                      (a) USING 句の指定がある。  
                      (b) RETURNING 句の指定がない。

F E0221      プロパティメソッドの SET と GET の定義が一致しません。  
              【Enterprise for Linux】SET/GET DEFINITION UNMATCHED IN PROPERTY METHOD.

- 【原因】            SET プロパティメソッドの USING パラメータに指定した項目のデータ記述と  
                      GET プロパティメソッドの RETURNING 指定に書いた項目のデータ記述とが同  
                      じでない。

F E0222 AS 指定が誤っています.  
【Enterprise for Linux】"AS" SPECIFICATION IS INCORRECT.

[原 因] 次の項目以外の利用者定義語を指定している。  
(1) クラス名  
(2) プログラム原型名

F E0223 SHIFT-CODE 指定のファイルのレコードの長さが誤っています.  
【Enterprise for Linux】THE LENGTH OF RECORD OF THE SHIFT-CODE APPLIED  
FILE IS WRONG.

[原 因] APPLY SHIFT-CODE 句が指定されたファイルで、レコード名のシフトコードや  
プリンタ制御コードを挿入した長さが、512 バイトを超えている。

F E0224 【Standard】画面節移行できない機能です.

[原 因] COBOL GUI の画面節移行機能使用時、ROLLING 句付き ACCEPT/DISPLAY  
文は使用できない。

F E0300 READ/WRITE 文の INTO/FROM 句の作用対象が誤っています.  
【Enterprise for Linux】THE OPERAND OF INTO/FROM CLAUSE IN READ/WRITE IS  
INVALID.

[原 因] 行順ファイル(CSV 形式)に対する READ/WRITE 文の INTO/FROM 句に次の指  
定がある。  
(a) 書き方 2 の OCCURS 句が含まれている。  
(b) 再命名項目である。  
(c) 用途が表示用以外の基本項目が含まれている。

F F0001 連絡節のレコードの数が制限を超えています.  
【Enterprise for Linux】THE LINKAGE SECTION CONTAINS MORE THAN 1025  
LINKAGE RECORDS.

[原 因] 連絡節でレベル番号"01"と"77"のデータ名の数 が 1025 個を超えた。

F F0003 【Standard】? 句のデータ名 `?` が誤っています.

- [原因] (1) データ名がファイル節, 作業場所節, 連絡節以外のデータ節で定義されている。
- (2) データ名が定義されていない。または多重に定義されている。
- (3) データ名が整数基本項目でない。
- (4) 定数名である。

- [埋め込み情報 1] (1) LINE
- (2) COLUMN
- (3) CLEAR

[埋め込み情報 2] 対応するデータ名

F F0004 `?'は?句に指定できません.  
**【Enterprise for Linux】**THE USER DEFINED WORD `?' CANNOT BE DESIGNATED  
 IN THE ? CLAUSE.

- [原因] (1) **FILE STATUS** 句で指定されたデータ名が次のものであった。
- (a) 2 桁の英数字項目, 集団項目でない。
- (b) ファイル節で定義されている。
- (c) 局所記憶節, 連絡節, ベースデータ節のデータ名であり, ファイル名と名前の有効範囲が異なる場合。
- (d) **【Enterprise for Windows】【Standard】**定数名である。
- (2) **FILE STATUS** 句で指定されたデータ名 (詳細情報値) が次のものであった。
- (a) 数字項目でない。
- (b) ファイル節, 局所記憶節で定義されている。
- (c) 連絡節, ベースデータ節のデータ名であり, ファイル名と名前の有効範囲が異なる場合。
- (d) **【Enterprise for Windows】【Standard】**定数名である。
- (3) **【Standard】WINDOW STATUS** 句 (画面状態) で指定されたデータ名が次のものであった。
- (a) 2 桁の英数字項目でない。
- (b) ファイル節, 作業場所節または連絡節以外で定義されている。
- (c) 定数名である。
- (4) **【Standard】WINDOW HIGHT** 句 (画面の高さ), **WINDOW WIDTH** 句 (画面の

幅), **DISPLAY LINE NUMBER** 句 (表示する行位置) および **DISPLAY COLUMN NUMBER** 句 (表示する桁位置) で指定されたデータ名が次のものであった。

- (a) 整数基本項目でない。
- (b) ファイル節, 作業場所節または連絡節以外で定義されている。
- (c) 定数名である。

(5) **【Enterprise for Windows】【Standard】**初期値設定用 **VALUE** 句の **ADDRESS OF** 句および **LENGTH OF** 句のデータ名が次のものであった。

- (a) 作業場所節またはファイル節以外で定義されている。
- (b) ファイル節で定義されている場合, そのファイルに **SAME RECORD AREA** 句が指定されている。
- (c) **OCCURS** 句を指定した項目およびその従属項目である。
- (d) 書き方 2 の **OCCURS** 句を含む集団項目である。
- (e) **ADDRESS OF** 句の場合, その項目に外部属性がある。
- (f) ブールデータ項目である。
- (g) オブジェクト参照項目である。
- (h) 定数名である。

[埋め込み情報 1] 利用者定義語

[埋め込み情報 2] **FILE STATUS**

**【Standard】WINDOW STATUS**

**【Standard】WINDOW HEIGHT**

**【Standard】WINDOW WIDTH**

**【Standard】DISPLAY LINE NUMBER**

**【Standard】DISPLAY COLUMN NUMBER**

**【Enterprise for Windows】【Standard】ADDRESS OF**

**【Enterprise for Windows】【Standard】LENGTH OF**

F F0005 **RENAMES** 句の作用対象同士の関係が誤っています。

**【Enterprise for Linux】THE RELATIONSHIP BETWEEN THE LEFT AND RIGHT SIDE DATA ITEMS IS INCORRECT IN THE RENAMES CLAUSE.**

[原 因] **RENAMES** 句の作用対象であるデータ名 2 とデータ名 3 の関係が誤っている。

- (1) データ名 3 がデータ名 2 に従属する項目である。
- (2) データ名 3 で指定されている領域の最初の位置が, データ名 2 で指定されている領域の最初の位置より論理的に後でない。

F F0006      **RENAMES** 句の作用対象`?`が誤っています。  
【Enterprise for Linux】THE RENAMES OPERAND `?` IS INCORRECT.

- [原因]      **RENAMES** 句の作用対象が誤っている。
- (1) 【Enterprise for Linux】作用対象がレベル番号 01, 66, 77, 88 をもつ記述項である。
  - 【Enterprise for Windows】【Standard】作用対象がレベル番号 01, 66, 77, 78, 88 をもつ記述項である。
  - (2) 作用対象が **OCCURS** 句をもつ記述項である。
  - (3) 作用対象が **OCCURS** 句をもつ記述項に従属する記述項である。
  - (4) 作用対象が可変繰返しデータ項目を含む記述項である。
  - (5) 作用対象が直前のレベル番号 01 に従属していない記述項である。
  - (6) 作用対象がオブジェクトのデータ項目を含む記述項である。
  - (7) 【Enterprise for Linux】作用対象が値付き用語データ項目である。

[埋め込み情報] 利用者定義語

F F0007      `?`は定義されていないまたは一意ではありません。  
【Enterprise for Linux】THE DATA NAME `?` IS NOT DEFINED OR NOT UNIQUE.

[原因]      利用者定義語が定義されていない, または一意でない。

[埋め込み情報] 利用者定義語

F F0008      **RECORD** 句で指定した整数が誤っています。  
【Enterprise for Linux】THE INTEGER IN THE RECORD CLAUSE IS INVALID.

- [原因]      (1) **FD/SD** 句中で, **RECORD CONTAINS** 句で指定した整数がレコードの大きさより小さい。
- (2) **FD/SD** 句中で, **RECORD VARYING** 句で指定した整数 3 がレコードの最大値より小さい
- (3) **FD/SD** 句中で, **RECORD VARYING** 句で指定した整数 2 がレコードの最小値より大きい。

F F0009      **REDEFINES** 句の作用対象`?`が定義されていません。  
【Enterprise for Linux】THE REDEFINED AREA `?` IS UNDEFINED.



【原因】 REDEFINES 句の作用対象が同じレベル番号をもつ直前の記述項のデータ名と一致しない。

【埋め込み情報】 利用者定義語

F F0010 可変繰返しデータ項目または可変繰返しデータ項目を含む項目に REDEFINES 句は指定できません。

【Enterprise for Linux】THE DATA ITEM WHICH HAS A REDEFINES CLAUSE CANNOT CONTAIN VARIABLE OCCURRENCE ITEMS.

【原因】 可変繰返しデータ項目または可変繰返しデータ項目を含む項目に REDEFINES 句は指定できない。

W F0011 REDEFINES 句の作用対象`?`より大きい項目長が指定されています。

【Enterprise for Linux】THE REDEFINING AREA IS GREATER THAN THE REDEFINED AREA `?` IN SIZE.

【原因】 再定義する項目の項目長が再定義される項目の項目長より大きい。

【埋め込み情報】 利用者定義語

【コンパイラの処理】 再定義する項目長に合わせる。

F F0012 REDEFINES 句の指定されている`?`は REDEFINES 句の作用対象に指定できません。

【Enterprise for Linux】THE `?` WITH THE REDEFINES CLAUSE CANNOT BE SPECIFIED IN THE REDEFINES CLAUSE.

【原因】 REDEFINES 句が指定されている利用者語は、REDEFINES 句の作用対象に指定できない。

【埋め込み情報】 利用者定義語

F F0013 OCCURS 句の指定されている`?`は REDEFINES 句の作用対象に指定できません。

【Enterprise for Linux】THE `?` WITH THE OCCURS CLAUSE CANNOT BE SPECIFIED IN THE REDEFINES CLAUSE.

【原因】 OCCURS 句の指定されている利用者定義語は、REDEFINES 句の作用対象にできない。

【埋め込み情報】 利用者定義語

F F0014    `?'は RECORD KEY 句に指定できません.  
**【Enterprise for Linux】**THE DATA NAME `?' CANNOT BE DESIGNATED IN THE  
 RECORD KEY CLAUSE.

- [原因]        (1) **【Enterprise】**RECORD KEY 句のデータ名が次のものである。
- (a) 当該ファイルのレコード記述項で定義されていない。
  - (b) 用途が表示用で、項類が英数字、英字、英数字編集、数字、数字編集、日本語、日本語編集の基本データ項目でないか可変繰返しデータ項目を含む集団項目である。
- (2) **【Standard】**RDB ファイル以外の RECORD KEY 句のデータ名が次のものである。
- (a) 当該ファイルのレコード記述項で定義されていない。
  - (b) 用途が表示用で、項類が英数字、英字、英数字編集、数字、数字編集、日本語、日本語編集の基本データ項目でないか可変繰返しデータ項目を含む集団項目である。
  - (c) 定数名である。
- (3) **【Standard】**RDB ファイルの RECORD KEY 句または ALTERNATE RECORD KEY のデータ名が次のものである。
- (a) 当該ファイルのレコード記述項で定義されていない。
  - (b) 243 桁以内の英数字項目または 121 桁以内の日本語項目または PIC 文字“P”を含まない数字項目でない。
  - (c) 数字項目が外部 10 進数で符号を持つ場合、符号は SIGN IS TRAILING 句の項目でない。
  - (d) 定数名である。

[埋め込み情報] 利用者定義語

F F0015    `?'は RELATIVE KEY 句に指定できません.  
**【Enterprise for Linux】**THE DATA NAME `?' CANNOT BE DESIGNATED IN THE  
 RELATIVE KEY CLAUSE.

- [原因]        (1) RELATIVE KEY 句のデータ名が次のものである。
- (a) 当該ファイルのレコード記述項で定義されている。
  - (b) ベースデータ節で定義されている。
  - (c) 局所記憶節、連絡節のデータ名であり、ファイル名と名前の有効範囲が異なる場合。

- (d) 符号なし整数基本項目でない。
- (e) PICTURE 句の文字列に P を含む項目である。
- (f) 【Enterprise for Windows】【Standard】定数名である。
- (2) 【Standard】RDB ファイルの RELATIVE 句のデータ名が次のものである。
  - (a) 作業場所節または連絡節以外で定義されている。
  - (b) 定数名である。

[埋め込み情報] 利用者定義語

F F0016     `?'は VALUE OF 句に指定できません。  
 【Enterprise for Linux】THE DATA NAME `?' CANNOT BE DESIGNATED IN THE  
 VALUE OF CLAUSE.

- [原因]     VALUE OF 句のデータ名が次のものである。
- (1) 作業場所節で定義されていない。
  - (2) 【Enterprise】256 桁以内の英数字基本項目または集団項目でない。
  - (3) 【Standard】RDB ファイル以外の場合, 256 桁以内の英数字基本項目  
 または集団項目でない。
  - (4) 【Standard】RDB ファイルの場合, 30 桁以内の英数字基本項目または  
 集団項目でない。
  - (5) 【Enterprise for Windows】【Standard】定数名である。

[埋め込み情報] 利用者定義語

F F0017     `?'は DEPENDING ON 句に指定できません。  
 【Enterprise for Linux】THE DATA NAME `?' CANNOT BE DESIGNATED PRECEDED  
 BY THE RESERVED WORD `DEPENDING ON'.

- [原因]     (1) OCCURS DEPENDING ON 句のデータ名が次のものである。
- (a) 整数基本項目でない
  - (b) OCCURS 句を含むデータ記述項により定義される最初の記憶域番  
 地から, その OCCURS 句を含むレコード記述項により定義される最後  
 の記憶域番地までの範囲にある。
  - (c) OCCURS 句が書かれている。
  - (d) EXTERNAL 句が指定されたのに, DEPENDING ON 句のデータ項目に  
 は EXTERNAL 句が記述されていない。
  - (e) GLOBAL 句が指定されたのに, DEPENDING ON 句のデータ項目には  
 GLOBAL 句が記述されていない。

- (f) 【Enterprise for Windows】【Standard】定数名である。
- (2) RECORD DEPENDING ON 句のデータ名が次のものである。
  - (a) 整数基本項目でない。
  - (b) PICTURE 句の文字列に P を含む項目である。
  - (c) データ名は整数 5 の値を格納するのに十分でない。
  - (d) 局所記憶節, 連絡節, ベースデータ節のデータ名であり, ファイル名と名前の有効範囲が異なる場合。
  - (e) 【Enterprise for Windows】【Standard】定数名である。

[埋め込み情報] 利用者定義語

F F0018     `?` は LINAGE 句に指定できません.  
 【Enterprise for Linux】THE DATA NAME `?` CANNOT BE DESIGNATED IN THE LINAGE CLAUSE.

- [原因]
- (1) LINAGE 句に指定したデータ名が符号なしの整数基本項目でない。
  - (2) データ名がベースデータ節で定義されている。
  - (3) 局所記憶節, 連絡節のデータ名であり, ファイル名と名前の有効範囲が異なる場合。
  - (4) 【Enterprise for Windows】【Standard】定数名である。

[埋め込み情報] 利用者定義語

F F0019     `?` は KEY 句に指定できません.  
 【Enterprise for Linux】THE DATA NAME `?` CANNOT BE DESIGNATED IN THE KEY CLAUSE.

- [原因]
- OCCURS 句の KEY 句のデータ名が次のものである。
- (1) 最初のデータ名 2 が, この OCCURS 句の左辺でなく, かつ, その左辺に従属する項目でもない。
  - (2) 2 番目以降のデータ名 2 が, この OCCURS 句の左辺に従属する項目でない。
  - (3) データ名 2 の次元数と, この OCCURS 句の左辺の次元数が異なっている。
  - (4) データ名 2 が日本語項目または日本語編集項目である。
  - (5) 【Enterprise for Windows】【Standard】定数名である。

[埋め込み情報] 利用者定義語

W F0020     `?'の VALUE 句の英数字定数が制限を超えています。  
【Enterprise for Linux】THE NONNUMERIC LITERAL IN THE VALUE CLAUSE OF `?'  
IS TOO LONG.

〔原因〕        集団項目に指定された VALUE 句の英数字定数の大きさがその集団項目の  
項目長より大きい。

〔埋め込み情報〕 利用者定義語

〔コンパイラの処理〕 項目長より大きい部分の右側は切り捨てられる。

F F0022     REDEFINES 句の作用対象`?'と境界が一致していません。  
【Enterprise for Linux】THE BEGINNING POSITION OF THE REDEFINES AREA `?'  
IS INCORRECT.

〔原因〕        再定義する項目の境界と再定義される項目の境界があわない。

〔埋め込み情報〕 利用者定義語

F F0023     可変繰返しデータ項目`?'は REDEFINES 句の作用対象に指定できません。  
【Enterprise for Linux】THE REDEFINES CLAUSE CONTAINS THE GROUP ITEM  
HAVING THE VARIABLE OCCURRENCE ITEM `?'.

〔原因〕        可変データ繰返し項目は REDEFINES 句の作用対象に指定できない。

〔埋め込み情報〕 利用者定義語

F F0027     BASED 句指定のあるファイルに対して VALUE OF 句で定数が指定されています。  
【Enterprise for Linux】A LITERAL IS SPECIFIED IN THE `VALUE OF` CLAUSE  
FOR A FILE DESCRIPTION WITH THE BASED CLAUSE.

〔原因〕        BASED 句指定のあるファイルに対して VALUE OF 句で定数が指定されてい  
る。

F F0029     重複するキーがあります。  
【Enterprise for Linux】THE ALTERNATE RECORD KEY AREA IS OVERLAPPED.

〔原因〕        索引ファイルのレコードキー定義で左端位置の等しいレコードキーが複数  
指定されている。

F F0032      **BLOCK CONTAINS** 句で指定した大きさが実際のレコードの大きさの整数倍ではありません。

【Enterprise for Linux】THE SIZE SPECIFIED IN THE BLOCK CONTAINS CLAUSE IS NOT INTEGER TIMES OF THE RECORD SIZE.

〔原因〕      大容量記憶ファイルまたは磁気テープファイルで **BLOCK CONTAINS** 句が **CHARACTERS** 指定のとき、指定値が最大レコードの大きさの整数倍でない。

F F0033      【Standard】**FROM** 句、**INTO** 句または **USING** 句の一意名?が誤っています。

〔原因〕      (1) 一意名がファイル節、作業場所節、連絡節以外のデータ節で定義されている。  
(2) 一意名が定義されていない。または多重に定義されている。  
(3) 【Standard】定数名である。

〔埋め込み情報〕 対応する一意名

W F0034      境界が一致しないため **SYNCHRONIZED/SYNC** 句は無視します。

【Enterprise for Linux】THE SYNCHRONIZED/SYNC CLAUSE IS IGNORED BECAUSE OF BOUNDARY UNMATCHED.

〔原因〕      再定義する項目または再定義する項目に従属する項目に **SYNC** 指定があるため再定義される項目と境界が合わない。

〔コンパイラの処理〕 **SYNC** 句がないものとして処理する。

F F0036      外部名の総数が 65535 を超えています。

【Enterprise for Linux】THE TOTAL NUMBER OF AN EXTERNAL NAME EXCEEDS 65535.

〔原因〕      次の外部的な名前の総数が 65535 を超えている。  
(a) **EXTERNAL** 句を指定しているデータ項目  
(b) **EXTERNAL** 句を指定しているファイル  
(c) 【Standard】主画面の画面記述名

F F0037      外部データの大きさが矛盾しています。

【Enterprise for Linux】THE LENGTH OF EXTERNAL DATA IS INCONSISTENT.

〔原因〕 それぞれのプログラムで定義した同一の外部データの大きさが異なっている。

W F0038 ?互換:データ項目の割り付け位置が異なります.  
【Enterprise for Linux】COMPATIBILITY WITH ?: IT IS NOT COMPATIBLE IN ALLOCATION.

〔原因〕 ACOS-4 のデータポインタデータ項目と割り付け位置が異なる。

〔埋め込み情報〕 ACOS-4

〔コンパイラの処理〕 そのまま翻訳を続ける。

F F0040 01 レベルの集団項目の大きさが 2G バイトを超えています.  
【Enterprise for Linux】A COMPILER GENERATED ELEMENTARY SEGMENT SIZE EXCEEDS 2G BYTES.

〔原因〕 01 レベルの集団項目の大きさが 2147483647 より大きい。  
(ファイル節以外の場合)

F F0041 01 レベルの集団項目の大きさが 64K バイトを超えています.  
【Enterprise for Linux】A COMPILER GENERATED ELEMENTARY SEGMENT SIZE EXCEEDS 64K BYTES.

〔原因〕 01 レベルの集団項目の大きさが 65535 より大きい。  
(ファイル節の場合)

F F0044 【Standard】合成キーの長さが 256 を超えた。

〔原因〕 合成キーで指定されたデータ項目長の和が 256 バイトを超えた。

O F0045 実行効率を低下させる記述があります.  
【Enterprise for Linux】THERE IS THE SPECIFICATION FOR DECREASE OBJECT-CODE EFFICIENCY.

〔原因〕 実行効率を高めるためにプログラムを書き直すべき記述がある。

O F0046 強制 SYNC 機能が適用されています。

**【Enterprise for Linux】THE IMPLICIT SYNCHRONIZED CLAUSE IS APPLIED.**

〔原因〕 強制 SYNC 機能が適応されている。

**F F0048** このデータ名はデータポインタデータ項目ではありません。または正しい節に書かれていません。

**【Enterprise for Linux】THIS DATA NAME IS NOT POINTER DATA ITEM, OR IS NOT DEFINED IN PROPER SECTION.**

〔原因〕 (1) データポインタデータ項目がファイル節、局所記憶節で定義された。  
(2) 連絡節、ベースデータ節のデータポインタデータ項目であり、ファイル名と名前の有効範囲が異なる場合。  
(3) **【Enterprise for Windows】【Standard】**定数名である。

**W F0049** 項類が英数字ではありません。英数字と見なして処理します。

**【Enterprise for Linux】THE CLASS IS NOT ALPHANUMERIC. ALPHANUMERIC IS ASSUMED.**

〔原因〕 (1) **FILE STATUS** 句のデータ名 1 に、2 バイトの符号なし外部 10 進数を指定した。  
(2) **RECORD KEY** 句、**ALTERNATE RECORD KEY** 句のデータ名に英数字基本項目、英字項目、数字項目、数字編集項目、日本語項目および日本語編集項目を指定した。

**F F0051** この項目が英数字項目ではありません。または正しい節に書かれていません。

**【Enterprise for Linux】THIS ITEM IS NOT ALPHANUMERIC, OR IS NOT DEFINED IN PROPER SECTION.**

〔原因〕 **CONTROL-AREA** 句で指定したデータ名の定義に誤りがある。

**F F0052** 用紙名が 6 桁以上の英数字項目ではありません。または正しい節に書かれていません。

**【Enterprise for Linux】THE FORM-MEDIA ITEM IS NOT ALPHANUMERIC ITEM EXCEEDING 6 CHARACTERS, OR IS NOT DEFINED IN PROPER SECTION.**

〔原因〕 **FORM-MEDIA** 句で指定したデータ名の定義に誤りがある。



F F0053 項目が 8 または 10 桁の英数字項目ではありません。または正しい節に書かれていません。

【Enterprise for Linux】THE ITEM IS NOT 8 OR 10 CHARACTERS ALPHANUMERIC DATA ITEM, OR IS NOT DEFINED IN PROPER SECTION.

〔原 因〕 FORM-SET 句で指定したデータ名の定義に誤りがある。

F F0054 不定長ファイルはブロッキングできません。

【Enterprise for Linux】THE ULR FORMAT MUST NOT BE BLOCKED.

〔原 因〕 WITH ULR 句指定のあるファイルに BLOCK CONTAINS 句を指定した。

F F0055 この SELECT 句には RECORD KEY 句が必要です。

【Enterprise for Linux】A RECORD KEY CLAUSE IS REQUIRED FOR SELECT CLAUSE.

〔原 因〕 索引ファイルに RECORD KEY 句の指定がない。

F F0057 `?'は ASSIGN 句に指定できません。

【Enterprise for Linux】THE DATA NAME `?' CANNOT BE DESIGNATED IN THE ASSIGN CLAUSE.

〔原 因〕 SELECT 句の ASSIGN 句のデータ名が次のものである。

- (1) 作業場所節に定義されていない。
- (2) 256 桁以内の英数字基本項目または集団項目でない。
- (3) 【Enterprise for Windows】【Standard】定数名である。

〔埋め込み情報〕 利用者定義語

F F0058 GLOBAL 句のあるレコード記述項を含むファイル記述項に GLOBAL 句の指定がありません。

【Enterprise for Linux】THERE IS NO DESIGNATION OF GLOBAL CLAUSE IN THE FILE NAME WHICH CONTAINS A RECORD DESCRIPTION ENTRY WITH GLOBAL CLAUSE.

〔原 因〕 全域属性を持つレコード記述項を 1 つでも含むファイル名が全域属性として定義されていない。

W F0068      **【Enterprise for Windows】【Standard】LENGTH OF 句で桁落ちが発生する。**

〔原因〕      LENGTH OF 句を含む記述項で桁落ちが発生している。

F F0188      SHIFT-CODE 指定のファイルのレコードでは、再定義する項目が大きくてはなりません。

**【Enterprise for Linux】REDEFINED ITEM MUST BE GREATER THAN ORIGINAL IN RECORD OF SHIFT-CODE APPLIED FILE.**

〔原因〕      APPLY SHIFT-CODE 句が指定されたファイルのレコードで、再定義される項目より再定義する項目が大きい。

F F0200      行順ファイルの USAGE 句に DISPLAY 以外は指定できません。

**【Enterprise for Linux】THE USAGE PHRASE OF RECORD DESCRIPTION ENTRY IN THE LINE SEQUENTIAL FILE MUST BE DISPLAY.**

〔原因〕      行順ファイルのレコード記述項に、用途が表示用以外のデータ項目がある。

F F0202      PROPERTY 句は、OCCURS 句を指定した項目およびその従属項目に指定できません。

**【Enterprise for Linux】THE PROPERTY CLAUSE CANNOT BE SPECIFIED FOR GROUP ITEM OR THE ITEM WITH OCCURS CLAUSE, THE SUBORDINATE ITEM.**

〔原因〕      PROPERTY 句は、OCCURS 句を指定した項目およびその従属項目に指定できない。

F F0203      REDEFINES 句の作用対象`?`が誤っています。

**【Enterprise for Linux】REDEFINES PHRASE CAN NOT BE SPECIFIED FOR ?.**

〔原因〕      REDEFINES 句の作用対象がオブジェクトのデータ項目である。

〔埋め込み情報〕 利用者定義語

F F0204      外部公開名が一意ではありません。

**【Enterprise for Linux】EXTERNALIZED NAME IS NOT IDENTIFIABLE.**

【原因】 外部公開名が一意でない。

F F0205      **【Standard】**プリンタファイルのフォーム指定が誤っています。

- 【原因】
- (1) フォーム名が 123 桁以内の英数字項目でない。
  - (2) フォーム名が作業場所節, 連絡節以外のデータ節で定義されている。
  - (3) **【Standard】**定数名である。

F F0214      CSV 形式ファイルにレベル番号 01 のレコードが複数定義されています。  
**【Enterprise for Linux】**MULTIPLE DEFINITION OF LEVEL 01 RECORDS IN CSV TYPE FILE.

【原因】 CSV 形式ファイルにレベル番号 01 のレコードが複数定義されている。

F F0215      CODE-SET 句が指定されたファイルに大きさの異なるレコードが定義されています。  
**【Enterprise for Linux】**RECORDS WITH DIFFERENT SIZES ARE DEFINED IN THE FILE IN WHICH THE CODE-SET CLAUSE IS SPECIFIED.

- 【原因】
- (1) **【Standard】【Enterprise for Windows】**-CF オプションが指定され, CODE-SET 句が指定されたファイルのファイル記述項に RECORD 句の指定がないか, 書き方 3 の RECORD 句が指定されているとき, このファイルに関連するすべてのレコード記述項の大きさが同じでない。
  - (2) **【Enterprise】**-CU オプションが指定され, CODE-SET 句が指定されたファイルのファイル記述項に RECORD 句の指定がないか, 書き方 3 の RECORD 句が指定されているとき, このファイルに関連するすべてのレコード記述項の大きさが同じでない。

F J0000      手続き文の書き方が複雑すぎて翻訳できません。  
**【Enterprise for Linux】**THE STATEMENT IS TOO COMPLEX.

- 【原因】
- (1) 算術式や条件式が複雑すぎる。
  - (2) SEARCH ALL 文の AND 句が多すぎる。
  - (3) 利用者定義関数で指定した引数の数が 1024 を超えている。
  - (4) CALL 文で指定した引数の数が 1024 を超えている。

(5) INVOKE 文で指定した引数の数が 1024 を超えている。

W J0001 ? 文の後に手続き文があります。

【Enterprise for Linux】THERE IS A STATEMENT PRECEDED BY THE ? STATEMENT.

[原因] STOP RUN, STOP ABORT, GO TO, GOBACK, EXIT PROGRAM, EXIT FUNCTION, EXIT METHOD, EXIT PARAGRAPH または EXIT SECTION 文が一連の無条件文の最後, または唯一の手続き文でない。

[埋め込み情報] (1) STOP RUN  
(2) STOP ABORT  
(3) GO  
(4) GOBACK  
(5) EXIT PROGRAM  
(6) EXIT FUNCTION  
(7) EXIT METHOD  
(8) EXIT PARAGRAPH  
(9) EXIT SECTION

[コンパイラの処理] そのまま翻訳を続行する。

W J0002 WHEN 句の条件は常に真となります。

【Enterprise for Linux】THE CONDITION OF THE WHEN PHRASE IS ALWAYS TRUE.

[原因] EVALUATE 文の WHEN 句の比較で, 選択対象の組の真理値が常に真である。

[コンパイラの処理] そのまま翻訳を続行する。

W J0003 WHEN 句の条件は常に偽となります。

【Enterprise for Linux】THE CONDITION OF THE WHEN PHRASE IS ALWAYS FALSE.

[原因] EVALUATE 文の WHEN 句の比較で, 選択対象の組の真理値が常に偽である。

[コンパイラの処理] そのまま翻訳を続行する。

F J0004 1 段落が EXIT 文のみで構成されていません。

【Enterprise for Linux】THE EXIT STATEMENT MUST BE THE ONLY STATEMENT IN THE PARAGRAPH.

[原因] EXIT 完結文には段落名を付け、この段落は EXIT 完結文だけで構成されなければならない(他の手続き文をこの段落に記述することはできない)。

F J0005 1 手続き文の演算の中間結果数が 255 個を超えています。  
【Enterprise for Linux】NUMBER OF CALCULATION INTERIM RESULT EXCEEDS 255  
IN ONE STATEMENT.

[原因] 1 手続き文中で実行される演算の中間結果数が 255 個を超えている。

F J0006 一意名の属性が誤っています。  
【Enterprise for Linux】THE IDENTIFIER IS INCORRECT.

[原因] (1) CALL 文または CANCEL 文の一意名が英数字基本項目でない。

W J0007 ? 文の一意名の長さが範囲を超えています。  
【Enterprise for Linux】THE IDENTIFIER DESIGNATED IN THE ? STATEMENT IS  
OUTSIDE THE RANGE.

[原因] 一意名で示されるプログラム名の桁落ちが発生する可能性がある。

[埋め込み情報] (1) CALL

(2) CANCEL

[コンパイラの処理] そのまま翻訳を続行する。

F J0008 PROCEDURE DIVISION のパラメータと矛盾しています。  
【Enterprise for Linux】THE PARAMETERS CONFLICT WITH THE PROCEDURE  
DIVISION.

[原因] (1) 内側プログラムの呼び出しで、CALL 文と PROCEDURE DIVISION の引数の数または長さが誤っている。

F J0009 【Standard】参照できない作用対象があります。

[原因] (1) ACCEPT, DISPLAY 文以外の作用対象に画面節で定義されたデータ名を指定している。

(2) OPEN, CLOSE 文以外の作用対象に画面節で定義されたデータ名を指

定している。

F J0010 添字付けまたは部分参照のできない一意名があります。  
**【Enterprise for Linux】DETECTED A IDENTIFIER WHOSE SUBSCRIPTING OR REFERENCE MODIFICATION IS NOT ALLOWED.**

- [原因]
- (1) 次元のない一意名に添字付けがある。
  - (2) 一意名でない語に‘(’がある。
  - (3) “USAGE IS DISPLAY”で定義したデータ名以外に部分参照を記述した。
  - (4) 部分参照できない利用者定義関数の関数一意名, オブジェクトプロパティの一意名, 組込み関数の関数一意名に部分参照を記述している。
  - (5) **【Standard】**画面記述名および画面データ名に部分参照を記述している。
  - (6) **【Enterprise for Windows】【Standard】**定数名である。

F J0011 添字付けに必要なデータ名があります。  
**【Enterprise for Linux】DETECTED A DATA ITEM REQUIRED SUBSCRIPTING.**

- [原因] 次元のあるデータに対し添字が付けられていない。

F J0012 添字付けまたは部分参照が誤っているデータ名があります。  
**【Enterprise for Linux】DETECTED A DATA ITEM WHOSE SUBSCRIPTING OR REFERENCE MODIFICATION IS INVALID.**

- [原因]
- (1) 添字に正しくない指定をしている。  
 正しい指定
    - ・ 上限を超えない整数
    - ・ 次元をもっていない整数基本項目
    - ・ データ名に属した指標名
 ただし 2 次元以上の表に対して, 1 次元目の指標と表を定義した 1 次元目の指標が一致しない場合も含む。
  - (2) 添字の相対値が以下のいずれかに該当する場合。
    - (a) +32767 を超えている。
    - (b) -32767 を超えている。

- (3) 部分参照の作用対象が以下のいずれかに該当する場合
  - (a) 一意名以外の利用者定義語である。
  - (b) 整数項目でない一意名である。
  - (c) 最左端または長さが定数で記述していて、データ項目の範囲外を参照する。
  - (d) **【Standard】**画面節で定義した一意名である。
- (4) **APPLY SHIFT-CODE** が指定されたファイルへの **WRITE** 文の **FROM** 句の一意名に部分参照が書かれている。
- (5) **CODE-SET** 句が指定されたファイルへの **WRITE** 文または **REWRITE** 文の **FROM** 句の一意名に部分参照が書かれている。
- (6) **CODE-SET** 句が指定されたファイルへの **READ** 文の **INTO** 句の一意名に部分参照が書かれている。
- (7) **【Enterprise for Windows】【Standard】**添字付けで添字に次のものを指定している。
  - ・ 符号なし数字定数以外かつ正符号付き数字定数以外を表す定数名
  - ・ 0 を表す定数名
- (8) **【Enterprise for Windows】【Standard】**部分参照の最左端または長さに次のものを指定している。
  - ・ 整数の数字定数以外を表す定数名

F J0014      **【Enterprise for Linux】**用語辞書の内容が誤っています(?)。  
**【Enterprise for Linux】**THE CONTENT OF TERM DICTIONARY IS INVALID(?).

[原因]          用語が参照できない。

- [埋め込み情報]
- (1)    1 (ログインできない)
  - (2)    2 (データベースエラー)
  - (3)    3 (メモリエラー)
  - (4)    4 (用語辞書がない)
  - (5)    5 (入力引数エラー)
  - (6)    6 (用語辞書管理テーブルがない)
  - (7)    7 (用語辞書管理テーブルは不正な値が格納されている)
  - (8)    13 (既定データソースが存在しない)
  - (9)    14 (その他エラー)
  - (10)   15 (用語値の長さエラー)
  - (11)   16 (用語値ネストオーバー)

- (12) 17(包含用語値の個数オーバー)
- (13) 18(用語値の属性種別エラー)
- (14) 19(PICTURE 情報の長さエラー)
- (15) 20(属性情報ネストオーバー)
- (16) 21(属性情報エラー)

F J0015 表の定義と次元があわないデータ名があります。  
【Enterprise for Linux】DETECTED A DATA ITEM WHOSE NUMBER OF SUBSCRIPTS  
DISAGREES WITH THE DEFINITION.

〔原因〕 添字付けにおいて、表の定義と次元が合わない。

F J0016 ALTER 文で参照される段落が 1 つの GO 文のみで構成されていません。  
【Enterprise for Linux】THE PARAGRAPH REFERRED IN AN ALTER STATEMENT MUST  
CONTAIN A SINGLE SENTENCE CONSISTING OF A GO STATEMENT.

〔原因〕 ALTER 文で参照される GO 文には段落名を付け、この段落は GO 文だけで構成されなければならない(他の手続き文をこの段落に記述することはできない)。

F J0017 ALTER 文で参照されていない GO TO 文に必要な手続き名がありません。  
【Enterprise for Linux】THE OBLIGATORY PARAGRAPH NAME IN THE GO TO  
STATEMENT IS MISSING.

〔原因〕 (1) ALTER で参照されていない段落に書かれた書き方 1 の GO TO 文で手続き名の省略が行われている。  
(2) 飛び先のない GO TO 文のある段落が ALTER で参照されていない。

W J0018 PERFORM 文で空手続きを参照しています。  
【Enterprise for Linux】THE PARAGRAPH REFERENCED BY THE PERFORM STATEMENT  
IS NULL.

〔原因〕 PERFORM 文で参照されている手続き名には実行する手続き文が存在しない(実行しても意味のない PERFORM 文である)。

〔コンパイラの処理〕 警告メッセージ出力のみでオブジェクトは生成する。



F J0020 GO TO 文の **DEPENDING ON** 句の作用対象が整数基本項目ではありません。  
【Enterprise for Linux】THE DEPENDING ITEM IS NOT A NUMERIC ELEMENTARY  
ITEM.

[原 因] GO TO 文の **DEPENDING ON** 句の作用対象が整数基本項目でない。

F J0021 作用対象が手続き名ではありません。  
【Enterprise for Linux】THE OPERAND MUST BE A PROCEDURE NAME.

[原 因] **ALTER** 文, **GO TO** 文, **SORT** 文, **MERGE** 文または **PERFORM** 文の作用対象が手  
続き名でない。

F J0022 参照できない手続き名を参照しています。  
【Enterprise for Linux】THE STATEMENT REFERENCES AN INCORRECT PROCEDURE  
NAME.

[原 因] **ALTER** 文, **GO TO** 文または **PERFORM** 文で参照できない手続き名を参照した。  
(1) 宣言部分の **GO TO**／**PERFORM** 文が手続き部分(宣言節以外)の手続き  
名を参照している。  
(2) 手続き部分の **GO TO** 文が宣言節の手続き名を参照している。  
(3) **ALTER** 文の手続き名 1 に節名が指定されている。

F J0023 **PERFORM** 文の 2 つの手続き名の定義されている節が矛盾しています。  
【Enterprise for Linux】THE TWO PROCEDURE NAMES IN THE PERFORM STATEMENT  
ARE DEFINED IN INCORRECT PLACES.

[原 因] **PERFORM** 文に **THRU** がある場合, 手続き名の一方が宣言節中の手続き名で  
あるのに, 他方の手続き名が同一の宣言節中に定義されていない。

F J0024 **PERFORM** 文の **TIMES** 句の作用対象の項類が整数ではありません。  
【Enterprise for Linux】THE DATA ITEM IN THE TIMES CLAUSE OF PERFORM  
STATEMENT IS NOT A NUMERIC INTEGER.

[原 因] **PERFORM** 文の **TIMES** 指定の作用対象が, 整数または整数基本項目でな  
い。

F J0025      **PERFORM** 文の?句の作用対象の属性が誤っています。  
【Enterprise for Linux】THE DATA ITEM IN THE ? CLAUSE OF PERFORM STATEMENT  
IS INCORRECT.

〔原因〕      **PERFORM** 文において、次の誤りがある。

- (1) **VARYING, AFTER** 句の作用対象が一意名または指標名でない。
- (2) **FROM** 句の作用対象が一意名、指標名または数字定数でない。
- (3) **BY** 句の作用対象が一意名またはゼロでない数字定数でない。

なお、上記一意名はすべて数字基本項目でなければならない。

〔埋め込み情報〕 (1) **VARYING**  
(2) **FROM**  
(3) **BY**  
(4) **AFTER**

F J0026      ?句の指標名と?句の作用対象の属性が矛盾しています。  
【Enterprise for Linux】THE INDEX NAME IN THE ? CLAUSE IS INCONSISTENT  
WITH THE DATA ITEM IN THE ? CLAUSE.

〔原因〕      **PERFORM** 文において、次の規則に反する。

- (1) **VARYING, AFTER** 句の作用対象が指標名のとき **FROM** 句、**BY** 句の一意名は整数基本項目、**FROM** 句の定数は正の整数、または **BY** 句の定数はゼロ以外の整数でなければならない。
- (2) **FROM** 句の作用対象が指標名のとき **VARYING** 句、**AFTER** 句、または **BY** 句の一意名は整数基本項目、または **BY** 句の定数はゼロ以外の整数でなければならない。

〔埋め込み情報〕

| 埋め込み情報 1       | 埋め込み情報 2                                    |
|----------------|---------------------------------------------|
| <b>VARYING</b> | <b>FROM</b><br><b>BY</b>                    |
| <b>AFTER</b>   | <b>FROM</b><br><b>BY</b>                    |
| <b>FROM</b>    | <b>VARYING</b><br><b>AFTER</b><br><b>BY</b> |

F J0027      STOP 文の定数が誤っています。  
               【Enterprise for Linux】THE LITERAL IN THE STOP STATEMENT IS INVALID.

- 〔原因〕      (1) ALL 定数が指定してある。  
               (2) 【Enterprise for Windows】【Standard】定数名の値が ALL 定数である。

F J0028      SET 文の作用対象が誤っています。  
               【Enterprise for Linux】THE OPERAND IN THE SET STATEMENT IS INVALID.

- 〔原因〕      (1) T0 の前に記述された受取り側が以下のいずれかの場合  
                   (a) 書き方 1: 指標データ項目, 整数基本項目または指標名以外のとき  
                   (b) 書き方 3: 呼び名に対応付けられたスイッチ名以外のとき  
                   (c) 書き方 4: 条件名以外のとき  
               (2) 書き方 1 または書き方 2 において T0 の後に記述された送り出し側が以下のいずれかの場合  
                   (a) 書き方 1: 指標データ項目, 整数基本項目, 指標名または正の整数以外のとき  
                   (b) 書き方 5: オブジェクト参照項目, クラス名, 既定義オブジェクト参照 SELF, 既定義オブジェクト参照 NULL 以外のとき  
                   (c) 書き方 7: ポインタ項目, ポインタ名, 既定義アドレス NULL, ADDRESS OF 一意名以外のとき  
               (3) UP/DOWN の前に記述された受取り側が以下のいずれかの場合  
                   (a) 書き方 2: 指標名以外のとき  
                   (b) 書き方 9: ポインタ項目またはポインタ名以外のとき  
               (4) UP/DOWN の後に記述された送り出し側が以下のいずれかの場合  
                   (a) 書き方 2: 整数基本項目または整数以外のとき  
                   (b) 書き方 9: 整数基本項目の正の整数以外のとき  
               (5) 書き方 1, 書き方 2 または書き方 5 において転記の規則に違反した場合

W J0029      可変繰返しデータ項目を指定することはできません。最大桁数で処理されます。  
               【Enterprise for Linux】ITEM MUST NOT BE VARIABLE LENGTH. MAXIMUM LENGTH IS ASSUMED.

[原因] コード変換システムサブルーチンの引数に可変繰返しデータ項目が指定されている。

[コンパイラの処理] 最大桁数の大きさの英数字項目とみなされる。

F J0030 B\_SYSTEM システムサブルーチンの指定が誤っています。  
【Enterprise for Linux】THE B\_SYSTEM SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

[原因] B\_SYSTEM システムサブルーチンの指定が誤っている。

F J0031 B\_CLEARERR システムサブルーチンの指定が誤っています。  
【Enterprise for Linux】THE B\_CLEARERR SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

[原因] B\_CLEARERR システムサブルーチンの指定が誤っている。

F J0032 USING 句に VALUE 句は指定できません。  
【Enterprise for Linux】THE VALUE CLAUSE IS DESIGNATED IN THE USING CLAUSE.

[原因]

- (1) 含まれるプログラムの手続き部の PROCEDURE DIVISION USING 句に VALUE 句が指定されている。
- (2) 含まれるプログラムを呼ぶ CALL 文の USING 句に VALUE 句が指定されている。

F J0033 PROCEDURE DIVISION ? 句の作用対象が誤っています。  
【Enterprise for Linux】THE OPERANDS OF ? CLAUSE IN THE PROCEDURE DIVISION ARE INCORRECT.

- [原因]
- (1) 手続き部の PROCEDURE USING 句または RETURNING 句に連絡節のレベルが 01 または 77 のデータ名以外のものが指定されている。
  - (2) 手続き部の PROCEDURE USING 句または RETURNING 句のデータ項目は REDEFINES 句を含んでいる。
  - (3) BY VALUE 句に字類が数字、オブジェクト、またはポインタ以外のデータ項目が指定されている。

- [埋め込み情報] (1) USING  
(2) RETURNING

F J0035 ?句の作用対象が誤っています.

【Enterprise for Linux】THE OPERANDS OF ? CLAUSE ARE INCORRECT.

- [原因]
- (1) CALL 文/INVOKE 文の USING 句に指定された一意名はファイル節, 作業場所節, 局所記憶節, ベースデータ節または連絡節の項類がブール以外でなければならない。
  - (2) BY CONTENT 句または BY VALUE 句でファイル名が書かれていた場合。
  - (3) CALL 文/INVOKE 文で RETURNING 句の作用対象が, 次のいずれかである。
    - (a) ファイル節, 作業場所節, 局所記憶節, ベースデータ節または連絡節で定義されていない。
    - (b) ブールデータ項目
    - (c) ポインタ名
  - (4) CALL 文の書き方 3/INVOKE 文の USING 句の作用対象が次のいずれかである。
    - (a) ブールデータ項目
    - (b) ポインタ名
    - (c) BY REFERENCE 指定のデータ名がファクトリ定義, インスタンス定義のファイル節や作業場所節で定義されたデータ項目である。
    - (d) BY VALUE 指定のデータ名の字類は数字, オブジェクトまたはポインタ以外である。
    - (e) 引数 NULL の渡し方を省略している場合, 対応する仮パラメータの渡し方が BY REFERENCE である。
  - (5) CALL 文の書き方 1と書き方 2 の USING 句の作用対象が次のいずれかである。
    - (a) ブールデータ項目
    - (b) BY REFERENCE の作用対象がオブジェクト参照項目である。
    - (c) BY VALUE 句の作用対象が可変繰り返しデータ項目である。
  - (6) 【Enterprise for Windows】【Standard】CALL 文の USING 句に ADDRESS OF と LENGTH OF の作用対象が定数名である。
- [埋め込み情報] (1) USING  
(2) RETURNING

W J0037      ALTER 文に同じ段落名を指定しています。  
【Enterprise for Linux】THE PARAGRAPH NAME IS DUPLICATED IN THE OPERAND  
OF THE ALTER STATEMENT.

〔原因〕      ALTER 文に同じ段落名を指定している。  
〔コンパイラの処理〕    そのまま翻訳を続行する。

W J0038      ALTER 文の手続き名 1 と手続き名 2 に同じ段落名が指定されています。  
【Enterprise for Linux】IN THE ALTER STATEMENT, THE PROCEDURE-NAME-1 AND  
THE PROCEDURE-NAME-2 IS DUPLICATED.

〔原因〕      ALTER 文の手続き名 1 と手続き名 2 に同じ段落名が指定されている。  
〔コンパイラの処理〕    そのまま翻訳を続行する。

F J0040      コード変換サブルーチンの指定が誤っています。  
【Enterprise for Linux】THE CODE CHANGE SUBROUTINE CONTAINS INVALID  
DESCRIPTION.

〔原因〕      (1) コード変換サブルーチンに **USING** 句が指定されていない。  
                (2) コード変換サブルーチンの **USING** 句の作用対象が誤っている。  
                    (a) 作用対象の数が 3 個でない。  
                    (b) 作用対象がファイル節, 作業場所節, 連絡節で定義されたデータ  
                        項目でない。  
                    (c) 作用対象に可変繰返しデータ項目が指定されている。  
                    (d) 作用対象が添字付けされている。  
                    (e) 作用対象の属性に誤りがある。  
                    (f) 作用対象に **BY CONTENT** 句または **BY VALUE** 句が指定されている。  
                    (g) データ名 2 に **JUSTIFIED** 句が指定されている。  
                    (h) データ名 3 の項目長が 6 以上でない。  
                (3) コード変換サブルーチンに **WITH RETURNING TO** 句が指定されている。  
                (4) プログラム内に混在できないコード変換サブルーチンが指定されてい  
                    る。

F J0041      ALTER 文で参照される GO TO 文に **DEPENDING ON** 句が指定されています。  
【Enterprise for Linux】THE GO TO STATEMENT WITH THE DEPENDING ON CLAUSE  
CANNOT BE DESIGNATED IN THE ALTER STATEMENT.

[原因] ALTER 文で参照される GO TO 文に **DEPENDING ON** 句が指定されている。

W J0043 USING 句の作用対象の整数部の桁落ちが発生します。  
**【Enterprise for Linux】MOST SIGNIFICANT DIGIT(S) OF INTEGER PART ARE POSSIBLY TRUNCATED.**

[原因] システムサブルーチンで **USING** 句の作用対象の整数部の桁落ちが発生する。

[コンパイラの処理] そのまま翻訳を続行する。

F J0044 **【Enterprise for Windows】**プログラム名が誤っています。  
**【Enterprise for Linux】【Standard】**プログラム名またはメソッド名が誤っています。  
**【Enterprise for Linux】THE PROGRAM NAME OR METHOD NAME IS INCORRECT.**

[原因] (1) **CALL** 文/**CANCEL** 文で呼び出すプログラム名が誤っています。  
 (a) **【Enterprise】C-**で始まる文字列  
 (b) **C\_**で始まる文字列  
 (c) 英字 **A～Z**, 数字 **0～9**, 下線でない文字が含まれている  
 (d) 語頭が, 英字および下線でない  
 (2) **【Enterprise for Windows】【Standard】INVOKE** 文で呼び出すメソッド名が英数字字類でない定数名である。

F J0045 ?文の **KEY** データが誤っています。  
**【Enterprise for Linux】ILLEGAL KEY DATA ITEM IN THE ? STATEMENT.**

[原因] (1) **KEY** データ名が可変長項目である。  
 (2) **KEY** データ名として指定されたデータ項目の記述項に **OCCURS** 句が指定されている。または **OCCURS** 句の指定がある項目に従属している。  
 (3) **KEY** データ項目間で記憶域の重なりがある。  
 (4) **KEY** データ項目の合計個数が **64** を超えた。  
 (5) **KEY** データ項目の合計長が **4096** を超えた。  
 (6) 整列併合用ファイルが可変長レコードを含むときキーデータ名が **ASCENDING** 句で指定されていない。またはキーデータ名のデータ項目が主レコードキー項目と同じ文字位置を占めていない。

- (7) **GIVING** 句のファイルが索引ファイルのとき最初のキーデータ名が **ASCENDING** 句で指定されていない。またはキーデータ名のデータ項目が主レコードキー項目と同じ文字位置を占めていない。

- (8) **【Enterprise for Windows】【Standard】KEY** データが定数名である。

[埋め込み情報] (1) **MERGE**

- (2) **SORT**

**F J0046**      ?文の指定するレコードサイズが範囲内にありません。

**【Enterprise for Linux】THE RECORD SIZE OF THE ? STATEMENT IS OUT OF RANGE.**

[原因]          (1) **USING** 句のファイルのレコードの大きさが整列併合用ファイルの最小レコードより小さいまたは最大レコードより大きい。

- (2) 整列併合用ファイルのレコードの大きさが **GIVING** 句のファイルの最小レコードより小さいまたは最大レコードより大きい。

[埋め込み情報] (1) **MERGE**

- (2) **SORT**

**W J0047**      **【Standard】?**句の指定が誤っています。

[原因]          (1) **RDB** ファイルの **CLOSE** 文に **NO REWIND, LOCK, REEL, UNIT, FOR REMOVAL** を指定した。

- (2) **RDB** ファイルの **OPEN** 文に **NO REWIND, LOCK** を指定した。

[埋め込み情報] (1) **NO REWIND**

- (2) **LOCK**

- (3) **REEL**

- (4) **UNIT**

- (5) **REMOVAL**

**F J0048**      レコード名の属性が誤っています。

**【Enterprise for Linux】THE OPERAND OF FROM/INTO CLAUSE IS INCORRECT.**

[原因]          (1) **INTO** 句を記述した **READ** または **RETURN** 文のファイル名に関連したレコード名が指標データ項目である。

- (2) **FROM** 句を記述した **REWRITE**, または **WRITE** 文のレコード名が指標データ項目である。



F J0049 FROM 句または INTO 句の作用対象とファイルのレコードが MOVE の転記規則に違反しています。

【Enterprise for Linux】THE RULES OF MOVE CANNOT BE APPLIED TO THE OPERAND OF FROM CLAUSE OR INTO CLAUSE AND RECORD RELATED TO FILE.

- 〔原因〕
- (1) READ 文または RETURN 文のファイル名に関連したレコード名を送り出し項目, INTO 句の作用対象を受け取り項目として転記の規則に違反している。
  - (2) REWRITE, WRITE または RELEASE 文のレコード名を受け取り項目, FROM 句の作用対象を送り項目として転記の規則に違反している。

F J0050 作用対象がファイル名ではありません。

【Enterprise for Linux】THE OPERAND MUST BE A FILE NAME.

〔原因〕 次の手続き文の作用対象がファイルでない。

- (1) CLOSE 文
- (2) DELETE 文
- (3) OPEN 文
- (4) READ 文
- (5) START 文
- (6) USE 文
- (7) SORT 文
- (8) RETURN 文
- (9) 【Standard】SCRATCH 文
- (10) 【Standard】SELECT 文

F J0052 作用対象に指定できないファイルを指定しています。

【Enterprise for Linux】AN INCORRECT FILE IS SPECIFIED IN THE OPERAND.

- 〔原因〕
- (1) READ 文において
    - (a) 【Enterprise】NEXT 指定なしの書き方 1 に動的呼び出し法のファイルを使った。
    - (b) 【Standard】NEXT /PRIOR 指定なしの書き方 1 に動的呼び出し法のファイルを使った。
    - (c) 書き方 1 の READ 文に乱呼び出し法のファイルを使った。
    - (d) 書き方 2 の READ 文に順呼び出し法のファイルを使った。

- (e) **【Enterprise】**順呼び出しファイルまたは、動的呼び出しファイル (NEXT 指定あり) に **NOT INVALID KEY** を指定した。
- (f) **【Standard】**順呼び出しファイルまたは、動的呼び出しファイル (NEXT /PRIOR 指定あり) に **NOT INVALID KEY** を指定した。
- (g) **【Enterprise】**乱呼び出し法のファイルまたは動的呼び出し法のファイル (NEXT 指定なし) に **NOT AT END** 句を指定した。
- (h) **【Standard】**乱呼び出し法のファイルまたは動的呼び出し法のファイル (NEXT/PRIOR 指定なし) に **NOT AT END** 句を指定した。

(2) **START** 文において

- (a) **【Enterprise】**順呼び出し法、または動的呼び出し法の相対ファイル、索引ファイル以外を使った。
- (b) **【Standard】**順呼び出し法、または動的呼び出し法の相対ファイル、索引ファイルまたは **RDB** 順ファイル以外を使った。
- (c) **【Standard】**書き方 2 の **START** 文に **RDB** 順ファイルを使った。

(3) **SORT** 文

- (a) **SORT** 文で整列用に用いるファイルが、整列用ファイル記述項で宣言されていない。
- (b) **SORT** 文の **USING/GIVING** 句で指定したファイルが、整列用のファイル記述項で宣言されている。
- (c) **SORT** 文の **USING/GIVING** 句で指定したファイルが乱呼び出し法である。
- (d) **GIVING** 句で指定したファイルに **OPTIONAL** 句が指定されている。
- (e) **ADDRESS FORMAT** 句に **LRR** 以外を指定した相対ファイルを指定している。
- (f) **SORT** 文の **USING/GIVING** 句で指定したファイルが行順ファイル (**CSV** 形式)である。

(4) **RETURN** 文

- (a) **RETURN** 文で用いるファイルが整列用ファイル記述項以外で宣言されている。

(5) **MERGE** 文

- (a) **MERGE** 文で整列併合用に用いるファイルは、整列併合用ファイル記述項で宣言されていない。
- (b) **MERGE** 文の **USING/GIVING** 句で指定したファイルが、整列併合用ファイル記述項で宣言されている。
- (c) **MERGE** 文の **USING/GIVING** 句で指定したファイルが、乱呼び出し法である。

- (d) **GIVING** 句で指定したファイルに **OPTIONAL** 句が指定されている。
- (e) **ADDRESS FORMAT** 句に **LRR** 以外を指定した相対ファイルを指定している。
- (f) **MERGE** 文の **USING** / **GIVING** 句で指定したファイルが行順ファイル (**CSV** 形式) である。
- (6) **DELETE** 文
  - (a) 順ファイルを指定した。
- (7) **OPEN** 文
  - (a) **BASED** 句が指定されているファイルを指定した。
- (8) **【Standard】SCRATCH** 文
  - (a) 順呼び出し/動的呼び出し法の **RDB** ファイル以外のファイルを指定した。
- (9) **【Standard】SELECT** 文
  - (a) 順呼び出し/動的呼び出し法の **RDB** ファイル以外のファイルを指定した。

F J0053      ?句の指定が誤っています。

**【Enterprise for Linux】THE ? CLAUSE CANNOT BE SPECIFIED.**

[原因]

- (1) **OPEN** 文において
  - (a) **WITH NO REWIND** 句を順ファイル以外に指定した。
  - (b) **I-O** 句を大容量記憶ファイル, フロッピーディスクファイル以外に指定した。
  - (c) **EXTEND** 句において
    - i. 順呼び出し法以外の相対ファイル, 索引ファイルに対して指定された。
    - ii. 整列用ファイルに指定された。
    - iii. **LINAGE** 句のあるプリンタファイルに指定した。
  - (d) **【Enterprise】OUTPUT** 句において **OPTIONAL** 指定ファイルを指定した。  
**【Standard】OUTPUT** 句, **I-O** 句または **EXTEND** 句において **OPTIONAL** 指定ファイルを指定した。
  - (e) プリンタファイルに **OUTPUT** 句以外を指定した。
  - (f) **I-O** 句を行順ファイル (**CSV** 形式) に指定した。
- (2) **READ** 文または **RETURN** 文において
  - (a) ファイルに従属する複数個の **01** レコード記述の桁数が, 等しくな

く、かつすべてのレコード名および一意名 1 のデータ項目が、  
 集団項目、日本語項目または、英数字基本項目以外である。

(b) **【Enterprise for Windows】【Standard】INTO** 句に定数名を指定した。

(3) **LINAGE** 句において

(a) **LINAGE** 句のあるプリンタファイルに **OPEN EXTEND** 文を記述した。

(4) **WRITE** 文において

(a) **ADVANCING** 句はプリンタファイルおよび **SYSOUT** ファイル以外に指定できない。

(5) **【Standard】**画面機能の **CLOSE** 文では

(a) 画面記述名後 **REEL/UNIT/WITH** 句がある。

(6) **【Standard】START** 文では

(a) **KEY** 句がある場合、**RDB** 相対ファイルを使用した。

(b) **RDB** ファイル以外で **KEY** 句複数データ名を指定した。

(c) **RDB** 以外の相対ファイルに対して、**KEY** 句に指定したデータ名 1 は、関連するファイル管理記述項の **RELATIVE KEY** 句に書かれたデータ項目ではない。

(d) **RDB** 以外の索引ファイルに対して、**KEY** 句に指定したデータ名 1 は、次のいずれかを満たされない。

ア. ファイル名 1 に関連する主レコードキーか副レコードキーであるデータ項目。

イ. 次の属性をもつデータ項目。

- ・ファイルのレコード内での最左端文字位置が、ファイル名 1 に関連し、**RECORD KEY** 句か、**ALTERNATE RECORD KEY** 句において定義された、主レコードキーか副レコードキーの最左端文字位置に対応している。

- ・字類、項類、用途が、そのレコードキーと同じである。

- ・長さが、そのレコードキーの長さより大きくない。

(7) **【Standard】READ** 文では

(a) ファイルに従属する複数個の **01** レコード記述の桁数が、等しくなく、かつすべてのレコード名および一意名 1 のデータ項目が、集団項目、日本語項目または、英数字基本項目以外である。

(b) 索引ファイル以外に **KEY** 句が指定された。

(c) 一般ファイルの **KEY** 句にデータ名が複数指定された。

(d) 順呼び出しの索引ファイルに **KEY** 句が指定された。

(e) 一般ファイルの索引ファイルの **KEY** データ名が、ファイル名の

RECORD KEY データ名ではない。

- [埋め込み情報] (1) OPEN 文では
- I-O
  - EXTEND
  - OUTPUT
  - INPUT
  - WITH NO REWIND
  - LOCK
- (2) READ 文または RETURN 文では
- INTO
  - KEY
- (3) LINAGE 句では
- EXTEND
- (4) WRITE 文では
- ADVANCING
- (5) 【Standard】画面機能の CLOSE 文では
- WITH NO REWIND
  - WITH LOCK
  - REEL
  - UNIT
  - REMOVAL
- (6) 【Standard】START 文では
- KEY

F J0054      AT END または INVALID KEY 句が指定されていません。  
 【Enterprise for Linux】THE RESERVED WORD `AT END` OR `INVALID KEY` MUST  
 BE SPECIFIED.

- [原因]      (1) DELETE 文においてファイル名が乱呼び出し法または動的呼び出し法のファイルでかつ USE 文が指定されていないファイルに対しては INVALID KEY 句を指定しなければならない。
- (2) READ 文においてファイル名に対して USE 手続きを指定していないときは, AT END 句または INVALID KEY 句を指定しなければならない。
- (3) REWRITE 文においてレコード名に関連したファイル名が順呼び出し法の索引ファイルまたは, 乱呼び出し法／動的呼び出し法の相対／索引ファイルで USE 手続きが指定されていない場合は INVALID KEY 句

を指定しなければならない。

- (4) **START** 文においてファイル名に対して **USE** 手続きを指定していないときは、**INVALID KEY** 句を指定しなければならない。
- (5) 相対ファイルまたは、索引ファイルの **WRITE** 文においてレコード名に関連したファイル名に対して **USE** 手続き名が指定されていないファイルに対しては **INVALID KEY** 句を指定しなければならない。

**F J0055**      **FROM** または **INTO** 句の作用対象が誤っています。  
**【Enterprise for Linux】THE OPERAND OF FROM CLAUSE OR INTO CLAUSE IS INCORRECT.**

- [原 因]
- (1) **READ** 文の **INTO** 句, **REWRITE** 文の **FROM** 句, **WRITE** 文の **FROM** 句, **RELEASE** 文の **FROM** 句, **RETURN** 文の **INTO** 句の作用対象が一意名でない。または一意名の領域とファイルの **01** レコード領域が同じである。
  - (2) ファイルが **CSV** 形式ファイル時, **READ** 文の **INTO** 句, **WRITE** 文の **FROM** 句に指定している一意名を部分参照指定している。
  - (3) **【Standard】RDB** ファイルに対する **READ** 文の **INTO** 句, **REWRITE** 文の **FROM** 句, **WRITE** 文の **FROM** 句の作用対象は局所記憶節で定義したデータ項目である。

**F J0056**      作用対象がレコード名ではありません。  
**【Enterprise for Linux】THE OPERAND MUST BE A RECORD NAME.**

- [原 因]
- 次の手続き文の作用対象がレコード名でない。
- (a) **REWRITE** 文
  - (b) **WRITE** 文
  - (c) **RELEASE** 文

**F J0058**      **RELATIVE KEY** のない相対ファイルに **START** 文は指定できません。  
**【Enterprise for Linux】THE START STATEMENT CANNOT BE USED FOR A RELATIVE FILE WITHOUT A RELATIVE KEY CLAUSE.**

- [原 因]      **RELATIVE KEY** のない相対ファイルに **START** 文は使えない。

**F J0059**      レコードに関連するファイルが誤っています。

【Enterprise for Linux】THE FILE RELATED TO THE RECORD IS INCORRECT.

- 〔原因〕
- (1) WRITE 文において
    - (a) プリンタファイル以外のファイルに対して ADVANCING 句が指定されている。
    - (b) LINAGE 句指定なしのファイルに対して AT EOP 句または NOT AT EOP 句が指定されている。
    - (c) 【Enterprise】順ファイルに対して INVALID KEY 句または NOT INVALID KEY 句が指定されている。
    - (d) 【Standard】RDB 以外の順ファイルに対して INVALID KEY 句または NOT INVALID KEY 句が指定されている。
    - (e) レコード名に関連したファイルが整列併合用ファイルである。
  - (2) RELEASE 文において
    - (a) レコード名が整列併合用ファイル記述項中のレコードでない。
  - (3) REWRITE 文において
    - (a) レコード名が行順ファイル (CSV 形式) 記述項中のレコードである。

F J0060      ADVANCING 句の作用対象が誤っています。

【Enterprise for Linux】THE OPERAND OF ADVANCING CLAUSE IS INCORRECT.

- 〔原因〕
- (1) WRITE 文の ADVANCING 句の作用対象が、次のいずれでもない。
    - (a) 整数基本項目
    - (b) 整数 (ゼロでも可)
    - (c) 表意定数の ZERO
  - (2) WRITE 文の ADVANCING 句の作用対象が呼び名であるとき、対応する機能名が次のいずれでもない。
    - (a) LN001～LN255
    - (b) CH01～CH12
    - (c) SP

F J0061      FROM 句の受け取り側の属性が誤っています。

【Enterprise for Linux】CATEGORY OF THE RECEIVING ITEM IS INCORRECT.

- 〔原因〕
- (a) ACCEPT 文の FROM 句が DATE, DAY, TIME, DAY-OF-WEEK 指定のとき、受け取り側項目に英字項目は使えない。
  - (b) FROM 句の呼び名が SYSIN に対応付けられている場合、受け取り側項

目に日本語／日本語編集項目は指定できない。

F J0062      **【Standard】**作用対象が画面節のデータ名の時 FROM 句は指定できません。

[原因]      (1) **ACCEPT** 文の作用対象が画面節のデータ名のとき FROM 句は指定できない。

F J0063      ?文の作用対象が誤っています。

**【Enterprise for Linux】**THE OPERAND OF ? STATEMENT IS INCORRECT.

[原因]      (1) **ACCEPT** 文の作用対象が指標データ項目、ブールデータ項目、オブジェクト参照項目、データポインタデータ項目である。

**【Standard】**ACCEPT 文の作用対象が画面記述項内のデータ項目である。

**【Enterprise for Windows】****【Standard】**ACCEPT 文の作用対象が定数名である。

(2) **DISPLAY** 文の作用対象が指標データ項目、オブジェクト参照項目、データポインタデータ項目、符号あり整数、整数以外の数字定数である。

**【Enterprise for Windows】****【Standard】**DISPLAY 文の作用対象が定数名である。

(3) **STOP** 文の作用対象が符号あり整数、整数以外の数字定数である。

(4) **SUBSTRING** 文において、次のいずれかである。

(a) 一意名 1 または一意名 4 がブールデータ項目、オブジェクト参照項目である。

(b) **BITS** 指定で、一意名 3, 整数 2, 一意名 6, または整数 4 が省略されており、かつ長さが 64 を超えている。

(c) **BITS** 指定で、一意名 1 および一意名 4 の大きさが、16777216 を超えている。

(d) 一意名 2, 一意名 3, 一意名 5, 一意名 6 は整数基本項目ではない。

(e) 整数 1, 整数 2, 整数 3, 整数 4 が整数以外の数字定数である。または数字定数の値が 2147483647 を超えている。

(f) 整数 1 と整数 2 は、次の条件のいずれかを満たしていない場合。  
(ここで、L は送り出し側の大きさを表し、ビット単位で扱われる場合は標準データ形式で数えた桁数を 8 倍した値である。)



- i.  $1 \leq \text{整数 } 1 \leq L$
  - ii.  $1 \leq \text{整数 } 2 \leq L$
  - iii.  $(\text{整数 } 1 + \text{整数 } 2 - 1) \leq L$  かつビット単位の場合は,
  - iv.  $\text{整数 } 2 \leq 64$
- (g) 整数 3 と整数 4 が, 次の条件のいずれかを満足しない場合。
- i.  $1 \leq \text{整数 } 3 \leq L$
  - ii.  $1 \leq \text{整数 } 4 \leq L$
  - iii.  $(\text{整数 } 3 + \text{整数 } 4 - 1) \leq L$  かつビット単位の場合は,
  - iv.  $\text{整数 } 4 \leq 64$
- (5) **【Standard】OPEN WINDOW** 直後のデータは画面記述名ではない。
- (6) **【Standard】CLOSE** 直後のデータは画面記述名ではない。
- (7) **【Standard】DISPLAY** 文の **UPON** 句の呼び名 - 1 が **GDD** に対応づけられているとき, 作用対象一意名 1 は局所記憶節で定義したデータである。
- (8) **【Standard】COBOL GUI** アプリケーション作成時, **ACCEPT** 文の作用対象の入力元が明示的または暗黙的に **CONSOLE**, **STDIN** に割り当てられている。
- (9) **【Standard】COBOL GUI** アプリケーション作成時, **DISPLAY** 文の作用対象の出力先が明示的または暗黙的に **CONSOLE**, **STDOUT**, **STDERR** に割り当てられている。

- [埋め込み情報] (a) **ACCEPT**  
 (b) **DISPLAY**  
 (c) **STOP**  
 (d) **SUBSTRING**  
 (e) **【Standard】CLOSE**  
 (f) **【Standard】OPEN**

**F J0064**      **FROM** または **UPON** 句の作用対象が誤っています。  
**【Enterprise for Linux】THE OPERAND OF FROM/UPON CLAUSE IS INVALID.**

- [原因]      (1) **ACCEPT** 文で **UPON** 句の呼び名が **CONSOLE**, **STDIN** または **SYSIN** に関連づけられていない。
- (2) **【Enterprise】DISPLAY** 文で **UPON** 句の呼び名が **CONSOLE**, **STDOUT**, **STDERR** または **SYSOUT** に関連づけられていない。
- (3) **【Standard】DISPLAY** 文で **UPON** 句の呼び名が **CONSOLE**, **STDOUT**, **GDD**, **STDERR** または **SYSOUT** に関連づけられていない。

- (4) **【Standard】DISPLAY** 文で **UPON** 句の呼び名が **GDD** に対応づけられているとき、作用対象として一意名または定数の複数を指定している。

F J0065 **【Standard】?**句は指定できません。

- [原因] (1) **DISPLAY** 文について、すべてが画面記述データ名のとき以外に **NO ACTUAL EXECUTING** 句/**BYPASS** 句/**AFTER** 句/**BEFORE** 句を指定した。  
 (2) **DISPLAY** 文について、すべてが画面記述データ名のときに **UPON** 句/**WITH NO ADVANCING** 句を指定した。  
 (3) **ACCEPT** 文/**DISPLAY** 文の作用対象に副画面の画面記述データ名がある時、**ROLLING** 句を指定した。  
 (4) ブーリアン操作機能の **COMPUTE** 文に **ON SIZE ERROR** 句を指定した。

- [埋め込み情報] (1) **WITH NO ACTUAL EXECUTING**  
 (2) **BYPASS**  
 (3) **UPON**  
 (4) **BEFORE**  
 (5) **ROLLING**  
 (6) **AFTER**  
 (7) **WITH NO ADVANCING**  
 (8) **ON SIZE ERROR**

F J0066 **【Standard】DISPLAY** 文の作用対象に画面記述データ名と一意名/定数が混在しています。

- [原因] **DISPLAY** 文の作用対象に画面記述データ名と一意名/定数が混在している。

F J0068 **【Standard】ROLLING** 句の作用対象が誤っています。

- [原因] (1) **ACCEPT** または **DISPLAY** 文の **ROLLING** 句の作用対象が誤っている。  
 (a) 整数基本項目でない。  
 (b) 符号のない整数でない。  
 (c) 整数の値が 1～255 の範囲でない。  
 (d) 整数-2 より整数-1 の方が大きい。  
 (e) 局所記憶節で定義したデータ項目である。  
 (2) 数字定数が 1 から **WINDOW HIGHT** 句の値の範囲でない。



W J0074     ?互換:転記の結果は異なる可能性があります.  
            【Enterprise for Linux】COMPATIBILITY WITH ?: IT IS NOT COMPATIBLE IN  
            RESULT OF MOVE.

[原 因]       英数字項目から数字編集項目への数字編集転記において, 送り出し側の  
                値により, ACOS-4 と異なる結果になる可能性がある。

[埋め込み情報] ACOS-4

[コンパイラの処理]   そのまま翻訳を続ける。

W J0077     呼び名 (制御変数) の大きさが受け取り項目の大きさより大きくなっています. 余分  
                な文字は桁落ちします.  
            【Enterprise for Linux】THE SIZE OF THE CONTROL VARIABLE IS LARGER THAN  
            THAT OF THE RECEIVER. TRUNCATION OCCURS.

[原 因]       ACCEPT (利用者制御変数の取り込み) 文に指定した呼び名の大きさが受け  
                取り項目の大きさよりも大きい。

[コンパイラの処理]   余分な文字は桁落ちする。

F J0086     【Standard】RDB ファイル以外に?句は指定できません。

[原 因]       当句は RDB ファイルのために拡張したものであり, 他のファイルには使えな  
                い。

[埋め込み情報] (1) CURRENT

(2) PRIOR

(3) FIRST

(4) LAST

F J0088     【Standard】データ名が日本語項目の時条件が EQUAL/=ではありません。

[原 因]       START 文の書き方 2 のデータ名が日本語項目の時, 比較条件に=以外を  
                指定している。

F J0090     選択対象の書き方が誤っています.  
            【Enterprise for Linux】THE SELECTION OBJECT CONTAINS ILLEGAL SYNTAX.

[原 因]       選択対象の書き方が誤っている。

- (1) 選択主体との組み合わせが正しくないとき
- (2) THROUGH 句の左辺と右辺の字類が異なるとき

**F J0091**      **ADVANCING PAGE** 句と **END-OF-PAGE/EOP** 句が同時に指定されています。  
**【Enterprise for Linux】BOTH OF THE ADVANCING PAGE PHRASE AND THE**  
**END-OF-PAGE/EOP PHRASE ARE SPECIFIED.**

[原 因]      1 つの **WRITE** 文に **ADVANCING PAGE** 句と **END-OF-PAGE** 句が同時に指定されている。

**W J0092**      送り出し項目の大きさが呼び名 (制御変数) の大きさより大きくなっています。余分な文字は桁落ちします。  
**【Enterprise for Linux】THE SIZE OF THE SENDER IS LARGER THAN THAT OF THE**  
**CONTROL VARIABLE. TRUNCATION OCCURS.**

[原 因]      **DISPLAY** (利用者制御変数の設定) 文に指定した送り出し項目の大きさが呼び名 (制御変数) の大きさよりも大きい。

[コンパイラの処理]      余分な文字は桁落ちする。

**W J0093**      呼び名 (文字型制御変数) が数字または数字編集項目へ転記されています。不正 10 進データでアボートする可能性があります。  
**【Enterprise for Linux】MOVING A CHARACTER TYPE CONTROL VARIABLE TO A**  
**NUMERIC (-EDITED) ITEM. ABORTION MAY OCCUR FOR ILLEGAL DECIMAL DATA.**

[原 因]      **ACCEPT** (利用者制御変数の取り込み) 文に指定した呼び名 (制御変数) が文字型で定義されており、受け取り項目が数字または数字編集項目で定義されている。

[コンパイラの処理]      そのまま翻訳を続ける。

**W J0094**      英数字項目が呼び名 (整数型制御変数) へ転記されています。不正 10 進データでアボートする可能性があります。  
**【Enterprise for Linux】MOVING AN ALPHANUMERIC ITEM TO AN INTEGER TYPE**  
**CONTROL VARIABLE. ABORTION MAY OCCUR FOR ILLEGAL DECIMAL DATA.**

[原 因]      **DISPLAY** (利用者制御変数の設定) 文に指定した項目が英数字項目、集団

項目であり呼び名 (制御変数) が整数型で定義されている。

[コンパイラの処理] そのまま翻訳を続ける。

**F J0095**      呼び名 (制御変数) と送り出し項目の組み合わせが正しくありません。  
**【Enterprise for Linux】THE CONTROL VARIABLE DOES NOT MATCH WITH THE SENDER.**

- [原因]          (1) 次のいずれかである。
- (a) **DISPLAY** (利用者制御変数の設定) 文に指定した呼び名 (制御変数) が整数型で定義されていて、送り出し項目が英字、英数字編集、日本語、日本語編集、**ZERO** 以外の表意定数または日本語定数が指定されている。
  - (b) **DISPLAY** (利用者制御変数の設定) 文に指定した呼び名 (制御変数) が文字型で定義されていて、送り出し項目が非整数の数字項目、日本語、日本語編集または日本語定数が指定されている。

**F J0096**      呼び名 (制御変数) と受け取り項目の組み合わせが正しくありません。  
**【Enterprise for Linux】THE CONTROL VARIABLE DOES NOT MATCH WITH THE RECEIVER.**

- [原因]          (1) 次のいずれかである。
- (a) **ACCEPT** (利用者制御変数の取り込み) 文に指定した呼び名 (制御変数) が整数型で定義されていて、受け取り項目が英字、日本語、日本語編集または日本語定数が指定されている。
  - (b) **ACCEPT** (利用者制御変数の取り込み) 文に指定した呼び名 (制御変数) が文字型で定義されていて、受け取り項目が日本語または日本語編集が指定されている。

**W J0098**      転記の受け取り側は最大長とみなします。  
**【Enterprise for Linux】THE MAXIMUM SIZE OF THE RECEIVING ITEM IS ASSUMED.**

- [原因]          次の一意名／レコード名が可変長で **DEPENDING ON** 句で指定したデータ項目を含む場合、最大長が対象になる。
- (1) **MOVE** 文の受け取り側の一意名
  - (2) **ACCEPT** 文の受け取り側の一意名
  - (3) **RETURN INTO** 一意名

- (4) READ INTO 一意名
- (5) RECEIVE INTO 一意名
- (6) WRITE レコード名 FROM 一意名
- (7) REWRITE レコード名 FROM 一意名
- (8) RELEASE レコード名 FROM 一意名
- (9) UNSTRING~DELIMITER IN 一意名
- (10) UNSTRING~INTO 一意名
- (11) STRING~INTO 一意名

[コンパイラの処理] 転記の受け取り側は最大長とみなす。

F J0099 複数個の一意名または定数を持つ DISPLAY 文は呼び名 (制御変数) を指定することはできません。

【Enterprise for Linux】THE DISPLAY STATEMENT SPECIFYING A CONTROL VARIABLE IS REQUIRED TO HAVE ONLY ONE IDENTIFIER/CONSTANT.

[原因] DISPLAY (利用者制御変数の設定) 文に複数個の一意名または定数を指定している。

F J0101 ?文の作用対象が誤っています。

【Enterprise for Linux】THE OPERAND OF ? STATEMENT IS INCORRECT.

- [原因]
- (1) INSPECT 文の次の作用対象が DISPLAY 項目でない。
  - (2) STRING 文の次の作用対象が DISPLAY 項目または ALL なし定数でない。
  - (3) UNSTRING 文の次の作用対象が英数字項目でない。

- [埋め込み情報]
- (1) INSPECT
  - (2) STRING
  - (3) UNSTRING

F J0102 ?句の作用対象が誤っています。

【Enterprise for Linux】THE OPERAND OF ? CLAUSE IS INCORRECT.

- [原因]
- (1) INSPECT 文において
    - (a) ALL, LEADING, BEFORE, AFTER, FIRST, BY, CONVERTING, TO 句の作用対象は次のものでなければならない。用途が表示用と定義された英字基本項目, 英数字基本項目 (桁数 256 以下), 数字基本

項目, 英数字定数, ALL なし表意定数

【Enterprise for Windows】【Standard】ALL, LEADING, BEFORE, AFTER, FIRST, BY, CONVERTING, TO 句の作用対象は次のものでなければならない。英数字定数/日本語定数/ALL なし表意定数を表す定数名

- (b) TALLYING 句の作用対象は数字基本項目でなければならない。
- (c) 書き方 2, 書き方 3 で ALL, LEADING, FIRST の作用対象の桁数と BY 句の作用対象の桁数は等しくなければならない。
- (d) 書き方 2, 書き方 3 で CHARACTERS 指定のとき BY 句, BEFORE 句, AFTER 句の作用対象の桁数は 1 桁でなければならない。
- (e) 書き方 4 で, CONVERTING 句の作用対象のデータ項目の文字数と TO 句の作用対象のデータ項目の文字数が等しくない。

(2) STRING 文において

- (a) DELIMITED BY 句の作用対象は次のものでなければならない。  
DISPLAY 項目 (数字項目の PIC 文字列に 'P' があってはならない), 英数字定数, ALL なし表意定数
- (b) INTO 句の作用対象は JUST 指定のない英数字基本項目でなければならない。
- (c) POINTER 句の作用対象が整数項目で INTO 句の作用対象の桁数 +1 の値を含むことのできるデータ項目であればよい。

(3) UNSTRING 文において

- (a) DELIMITED BY 句の作用対象は次のものでなければならない。英数字項目, 英数字定数, ALL なし表意定数  
【Enterprise for Windows】【Standard】DELIMITED BY 句の作用対象は次のものでなければならない。英数字定数/日本語定数/ALL なし表意定数を表す定数名
- (b) DELIMITER IN 句の作用対象は英数字項目でなければならない。
- (c) INTO 句の作用対象は英字項目 (PIC 文字列に 'B' があってはならない), 英数字項目 (集団項目でも可), 数字項目 (表示用 PIC 文字列に 'P' が不在の DISPLAY 項目) でなければならない。
- (d) COUNT IN 句, POINTER 句の作用対象は整数基本項目でなければならない。
- (e) POINTER 句の作用対象が, 整数項目で, INTO 句の作用対象の桁数 +1 の値を含むことのできないデータ項目である。

[埋め込み情報] (1) INSPECT 文のとき

- (a) ALL



- (b) LEADING
  - (c) BEFORE
  - (d) AFTER
  - (e) FIRST
  - (f) BY
  - (g) TALLYING
  - (h) CONVERTING
  - (i) TO
- (2) STRING 文のとき
- (a) DELIMITED
  - (b) INTO
  - (c) POINTER
- (3) UNSTRING 文のとき
- (a) DELIMITED
  - (b) DELIMITER
  - (c) INTO
  - (d) COUNT
  - (e) POINTER

W J0103     ?文の作用対象が誤っています。

【Enterprise for Linux】THE OPERAND OF THE ? STATEMENT IS INVALID.

[原因]        ANSI85 規格で禁止となった以下の記述がある。STRING 文の一意名 1 または一意名 2 が非整数の数字項目である。

[埋め込み情報] STRING

[コンパイラの処理]    そのまま翻訳を続行する。

W J0104     ?句の作用対象が誤っています。

【Enterprise for Linux】THE OPERAND OF ? PHRASE IS INVALID.

[原因]        ANSI85 規格で禁止となった以下の記述がある。

- (1) STRING 文の POINTER 句の作用対象に PIC 文字“P”がある。
- (2) UNSTRING 文の COUNT IN 句の作用対象に PIC 文字“P”がある。
- (3) UNSTRING 文の POINTER 句の作用対象に PIC 文字“P”がある。
- (4) UNSTRING 文の POINTER 句の作用対象が一意名 1 の項目長より 1 大きい値を格納するのに十分な桁数がない。

(5)UNSTRING 文の TALLYING 句の作用対象に PIC 文字“P”がある。

[埋め込み情報] POINTER, COUNT, TALLYING

[コンパイラの処理] そのまま翻訳を続ける。

F J0105 WITH 句の作用対象は符号なし 1 桁の整数基本項目でなければなりません。  
【Enterprise for Linux】THE OPERAND OF THE WITH CLAUSE MUST BE A UNSIGNED ONE-DIGIT INTEGER ITEM.

[原因] SEND 文の WITH 句の作用対象は符号なし 1 桁の整数基本項目でない。

W J0108 WITH NO ADVANCING 句は実行されません. この句は無視されます。  
【Enterprise for Linux】THE WITH NO ADVANCING CLAUSE IS NOT EXECUTED. THIS CLAUSE IS IGNORED.

[原因] 出力先が CONSOLE または SYSOUT の DISPLAY 文に WITH NO ADVANCING 句が指定されている。

[コンパイラの処理] そのまま翻訳を続行する。

F J0109 呼び名が整数型制御変数のときは英数字定数の値は数字でなければなりません。  
【Enterprise for Linux】IN CASE A CONTROL VARIABLE IS INTEGER TYPE, EVERY CHARACTER CONTAINED IN A NONNUMERIC LITERAL MUST BE NUMERIC.

[原因] 整数型制御変数に対する DISPLAY (利用者制御変数の設定) 文で, 定数が数字以外である。

F J0113 USING または GIVING 句に同じファイル名が指定されています。  
【Enterprise for Linux】THE SAME FILE NAME IS SPECIFIED IN BOTH USING AND GIVING PHRASE.

[原因] 1つの MERGE 文または SORT 文で入力ファイルと出力ファイルに同じファイル名を指定している。

F J0114 ファイル入出力用サブルーチンの指定が誤っています。  
【Enterprise for Linux】THE FILE-IO SUBROUTINE CONTAINS INVALID DESCRIPTION.

【原因】 ファイル入出力用サブルーチンの指定が誤っている。

- F J0115 CALL 文に指定したプログラム名は、`A\_CBL\_`、`B\_`、`E\_CBR\_`で始まる名前であってはなりません。  
【Enterprise for Linux】IN THE CALL STATEMENT, A PROGRAM NAME STARTING WITH `A\_CBL\_`, `B\_` OR `E\_CBR\_` IS NOT ALLOWED.

【原因】 CALL 文の定数に`A\_CBL\_`,`B\_`,`E\_CBR\_`で始まる英数字定数を指定している。

- F J0116 B\_GERRNO システムサブルーチンの指定が誤っています。  
【Enterprise for Linux】THE B\_GERRNO SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

【原因】 B\_GERRNO システムサブルーチンの指定が誤っている。

- F J0117 B\_SYSDATE システムサブルーチンの指定が誤っています。  
【Enterprise for Linux】THE B\_SYSDATE SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

【原因】  
B\_SYSDATE システムサブルーチンの指定が誤っている。

- F J0118 B\_CMOPT システムサブルーチンの指定が誤っています。  
【Enterprise for Linux】THE B\_CMOPT SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

【原因】  
B\_CMOPT システムサブルーチンの指定が誤っている。

- F J0119 B\_PUTENV システムサブルーチンの指定が誤っています。  
【Enterprise for Linux】THE B\_PUTENV SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

【原因】  
B\_PUTENV システムサブルーチンの指定が誤っている。

F J0122     B\_GETENV システムサブルーチンの指定が誤っています。  
          【Enterprise for Linux】THE B\_GETENV SYSTEM SUBROUTINE CONTAINS INVALID  
          DESCRIPTION.

〔原因〕

B\_GETENV システムサブルーチンの指定が誤っている。

W J0123     COMPUTATIONAL-5/COMP-5 項目への転記で下位の桁落ちが発生します。  
          【Enterprise for Linux】THE MOVE TO A COMPUTATIONAL-5/COMP-5 ITEM CAUSES  
          LOWER-DIGIT TRUNCATION.

〔原因〕

MOVE 文で下位の桁落ちが発生する。

〔コンパイラの処理〕

桁数の超えている部分は切り捨てられる。

W J0124     COMPUTATIONAL-5/COMP-5 項目への転記で上位の桁落ちが発生します。  
          【Enterprise for Linux】THE MOVE TO A COMPUTATIONAL-5/COMP-5 ITEM CAUSES  
          UPPER-DIGIT TRUNCATION.

〔原因〕

MOVE 文で上位の桁落ちが発生する。

〔コンパイラの処理〕

桁数の超えている部分は切り捨てられる。

F J0130     COMBINE 文の作用対象が誤っています。  
          【Enterprise for Linux】THE OPERAND OF THE COMBINE STATEMENT IS ILLEGAL.

〔原因〕

COMBINE 文の作用対象が誤っている。

- (1) 受け取り項目および被演算項目が英数字データ項目でない。
- (2) 受け取り項目および被演算項目が 256 桁を超えている。
- (3) シフト指示が符号なしの整数基本項目でない。

(4) シフト指示の定数値が 255 を超えている。

F J0142 同じ **MULTIPLE FILE TAPE** 句に指定されたファイルは使えません。  
【Enterprise for Linux】THE FILES CANNOT BE SPECIFIED IN THE SAME MULTIPLE  
FILE TAPE CLAUSE.

〔原因〕

**MERGE** 文で、同じ **MULTIPLE FILE TAPE** 句に指定されたファイルを指定している。

W J0149 **NUMERIC** 検査に指定された数字編集項目は英数字項目とみなして処理します。  
【Enterprise for Linux】THE NUMERIC EDITED DATA ITEM, WHICH IS SPECIFIED  
IN THE NUMERIC TEST, IS ASSUMED TO BE AN ALPHANUMERIC DATA ITEM.

〔原因〕

**NUMERIC** 検査に数字編集項目を指定した。

〔コンパイラの処理〕

そのまま翻訳を続行する。

F J0155 参照している連絡節のデータ名または指標名は **PROCEDURE DIVISION USING** 句  
または **RETURNING** 句に指定されていません。  
【Enterprise for Linux】THE DATA NAME OR THE INDEX NAME IS NOT DEFINED  
IN THE PROCEDURE DIVISION USING CLAUSE OR RETURNING CLAUSE.

〔原因〕

**PROCEDURE DIVISION** の **USING** 句または **RETURNING** 句に指定されていないデータ項目ま  
たは指標名を参照した。

F J0161 **SEARCH** 文の作用対象が誤っています。  
【Enterprise for Linux】THE OPERAND OF THE SEARCH STATEMENT IS INCORRECT.

〔原因〕

- (1) **SEARCH** 文の作用対象が誤っている。
- (2) 【Enterprise for Windows】【Standard】**SEARCH** 文または **SEARCH ALL** 文の次の作用  
対象が定数名である。

F J0201 MOVE 文の送り出し側が誤っています。  
【Enterprise for Linux】THE SENDING SIDE OF THE MOVE STATEMENT IS  
INCORRECT.

[原 因]

- (1) MOVE 文の送り出し側が誤っている。
- (2) 日本語編集転記のとき、送り出し側が日本語項目以外のとき 256 桁を超えている。

F J0202 MOVE 文の受け取り側が誤っています。  
【Enterprise for Linux】THE RECEIVING SIDE OF THE MOVE STATEMENT IS  
INCORRECT.

[原 因]

MOVE 文の受け取り側が誤っている。

F J0203 送り出し側と受け取り側が転記の規則に違反しています。  
【Enterprise for Linux】THE RULES FOR THE MOVE STATEMENT ARE NOT OBSERVED.

[原 因]

MOVE 文の送り出し側と受け取り側が転記の規則に違反している。

F J0204 ? 文の作用対象が誤っています。  
【Enterprise for Linux】THE OPERAND OF ? STATEMENT IS INCORRECT.

[原 因]

- (1) ADD, SUBTRACT, MULTIPLY, DIVIDE 文の作用対象が数字項類でない。ただし、  
GIVING 後の作用対象は数字編集項目であってもよい。
- (2) CALL 文または INVOKE 文の引数に算術式を指定している場合、算術式の作用対象が  
数字項類でない。

[埋め込み情報]

- (1) ADD
- (2) SUBTRACT
- (3) MULTIPLY
- (4) DIVIDE
- (5) CALL

(6) INVOKE

F J0206      COMPUTE 文の左辺が誤っています.  
              【Enterprise for Linux】THE LEFT HAND SIDE OF THE COMPUTE STATEMENT IS  
              INCORRECT.

【原 因】

COMPUTE 文の左辺が数字項目, 数字編集項目でない。

F J0207      算術式の書き方が誤っています.  
              【Enterprise for Linux】THE ARITHMETIC EXPRESSION CONTAINS ILLEGAL  
              SYNTAX.

【原 因】

算術式の書き方が誤っている。

F J0208      算術式の作用対象の項類が数字ではありません.  
              【Enterprise for Linux】THE OPERANDS OF ARITHMETIC EXPRESSION MUST BE  
              NUMERIC.

【原 因】

算術式の作用対象の項類が数字でない。

F J0209      かっこの対応がとれていません.  
              【Enterprise for Linux】LEFT AND RIGHT PARENTHESES ARE UNEQUAL IN NUMBER.

【原 因】

- (1) 算術式のカッコの対応が合わない。
- (2) 条件式のカッコの対応が合わない。

F J0210      条件式の書き方が誤っています.  
              【Enterprise for Linux】THE CONDITIONAL EXPRESSION CONTAINS ILLEGAL  
              SYNTAX.

【原 因】

条件式の書き方が誤っている。

- F J0212 比較条件の略記法が誤っています.  
【Enterprise for Linux】ABBREVIATED COMBINED RELATION CONDITION IS ILLEGAL.

【原因】

比較条件の略記法が誤っている。

- (1) 最初に省略形を使った。
- (2) 省略形の直前が比較条件でない。
- (3) 一連の省略形の間に'('または')'がある。

- F J0213 比較の規則に違反しています.  
【Enterprise for Linux】THE DESCRIPTION IS NOT OBSERVED IN THE RULES OF COMPARISON.

【原因】

次の事項で比較の規則に違反している。

- (1) 比較条件の左辺と右辺
- (2) 条件名条件の条件変数と VALUE の値

- W J0216 MOVE 文で右の桁落ちが発生します.  
【Enterprise for Linux】RIGHT HAND SIDE CHARACTER(S) ARE TRUNCATED IN THE MOVE STATEMENT.

【原因】

MOVE 文で右の桁落ちが発生する。

【コンパイラの処理】

桁数の超えている部分は切り捨てられる。

- W J0217 MOVE 文で左の桁落ちが発生します.  
【Enterprise for Linux】LEFT HAND SIDE CHARACTER(S) ARE TRUNCATED IN THE MOVE STATEMENT.

【原因】

MOVE 文で受け取り側項目に JUSTIFIED 句の指定がある場合、送り出し側項目の桁数が受



け取り側項目の桁数より大きい。

[コンパイラの処理]

桁数を超えている部分は切り捨てられる。

W J0218      **MOVE** 文で小数部の下位の桁落ちが発生します。  
              **【Enterprise for Linux】LEAST SIGNIFICANT DIGIT(S) OF DECIMAL PART ARE  
              POSSIBLY TRUNCATED.**

[原 因]

**MOVE** 文で小数部の桁落ちが発生する。

[コンパイラの処理]

小数部で桁の超えている部分は切り捨てられる。

W J0219      **MOVE** 文で整数部の上位の桁落ちが発生します。  
              **【Enterprise for Linux】MOST SIGNIFICANT DIGIT(S) OF INTEGER PART ARE  
              POSSIBLY TRUNCATED.**

[原 因]

**MOVE** 文で整数部の桁落ちが発生する。

[コンパイラの処理]

整数部で桁の超えている部分は切り捨てられる。

F J0220      **VARYING** 句の作用対象の属性が誤っています。  
              **【Enterprise for Linux】THE OPERAND OF VARYING CLAUSE IS INCORRECT.**

[原 因]

**SEARCH** 文の **VARYING** 句の作用対象が指標名, 指標データ項目, または整数基本項目でない。

F J0221      ? 句の作用対象が誤っています。  
              **【Enterprise for Linux】THE OPERAND OF ? CLAUSE IS INCORRECT.**

[原 因]

SEARCH ALL 文の WHEN 句, AND 句において

- (1) EQUAL の左辺のデータが SEARCH ALL 文の作用対象の KEY 句中に指定されていなければならない, かつ最初の指標名によって指標付けされていなければならない。
- (2) EQUAL の右辺は KEY 句中のデータであってはならず, 最初の指標名で指標付けされてはいけない。
- (3) 条件名は単一の値をもつように定義しておかなければならず, その条件変数は, KEY 句において指定されていなければならない。

[埋め込み情報]

- (1) WHEN
- (2) AND

F J0222 指定された KEY の順番に参照されていません。

【Enterprise for Linux】THE REFERENCE IS NOT THE ORDER OF THE KEY CLAUSE DEFINED.

[原因]

SEARCH ALL 文で WHEN 句, AND 句の作用対象は SEARCH ALL の作用対象の KEY 句中のデータ名, またはそれに関連した条件名が KEY 句で指定した順番どおりに参照されていない。

F J0223 WHEN 句の書き方が誤っています。

【Enterprise for Linux】THE WHEN CLAUSE IS INCORRECT.

[原因]

- (1) WHEN 句の後に AND 句, NEXT SENTENCE, 無条件文以外がある。
- (2) WHEN 句, AND 句中の算術式が途中で終わっている。

F J0225 【Standard】?句の作用対象が誤っています。

[原因]

- (1) SELECT 文の COUNT 句の作用対象が, 整数基本項目でない。
- (2) SELECT 文の ORDER 句の作用対象が,
  - ・ ファイル名に関連するレコード記述内に定義したデータ記述項で, 数字項目, 243 桁以外の英数字項目 (集団項目を含む) または 121 桁以内の日本語項目でない。
  - ・ 数字項目で, PICTURE 文字列に“P”が含まれている。

- ・ 符号付き外部 10 進数のとき、符号が右端重ね符号でない。
- ・ 2 進データ項目である。
- ・ 合計桁数は 243 桁(ただし、日本語項目のとき日本語 1 文字は 2 桁として数える)を超える。
- ・ 定数名である。

(3) SELECT 文の COUNT 句の作用対象が局所記憶節で定義したデータ項目である。

[埋め込み情報] (1) COUNT

(2) ORDER

F J0226      **【Standard】**SELECT 文の選択式が誤っています。

[原因]      SELECT 文において、選択式中の条件記述が、次のいずれかである。

- ・ ファイル名に関連するレコード記述内のデータ項目が、右辺および左辺のいずれにも書かれていない。
- ・ ファイル名に関連するレコード記述内のデータ項目が、次のものである。
  - (a) 2 進データ項目
  - (b) 英字項目
  - (c) 可変長集団項目
  - (d) PICTURE 文字列中に“P”を含む数字項目
  - (e) 符号付き外部 10 進数のとき符号が右端重ね符号以外
- ・ ファイル名に関連するレコード記述外のデータ項目が、次のものである。
  - (a) 英字項目
  - (b) 可変長集団項目
  - (c) 局所記憶節で定義したデータ項目である。
- ・ 右辺と左辺の字類および桁数が同じでない。
- ・ 右辺および左辺のデータ項目が英数字項目のときには 243 桁、または日本語項目のときには 121 文字を超えている。
- ・ 条件として正負条件、字類条件、算術式を含む条件、略記された条件などが書かれている。
- ・ 選択式が部分値条件の場合にはさらに、次の条件を満たしていない。
  - (a) 左辺のデータ項目がレコード記述内データで、かつ固定長の英数字項目または日本語項目

- (b) 右辺のデータ項目がレコード記述外データまたは英数字定数
- (c) 右辺の項目長が左辺の項目長以下
- (d) 右辺のデータ項目が局所記憶節で定義したデータ項目にならない。

F J0227 無条件文の中に NEXT SENTENCE 句があります。  
【Enterprise for Linux】THE NEXT SENTENCE PHRASE APPEARS IN THE IMPERATIVE STATEMENT.

[原 因]

無条件文中に IF 文または SEARCH 文の NEXT SENTENCE 句がある。

F J0228 ? 文は宣言部分の中には指定できません。  
【Enterprise for Linux】THE ? STATEMENT CANNOT BE SPECIFIED IN THE DECLARATIVE PROCEDURES.

[原 因]

次の手続き文は宣言文の中に指定できない。

[埋め込み情報]

- (1) MERGE
- (2) SORT

F J0229 同じ SAME 句に指定されたファイル名があります。  
【Enterprise for Linux】THE FILES IN THE SORT/MERGE STATEMENT ARE SPECIFIED IN ONE SAME SORT/SORT-MERGE CLAUSE.

[原 因]

同じ SAME SORT AREA 句または SAME SORT-MERGE AREA 句に指定されたファイル名の組を 1 つの SORT または MERGE 文で指定した。

W J0230 送り出し側の項目に数字以外の文字が含まれる場合、この MOVE 文の正しい実行結果が得られません。  
【Enterprise for Linux】RESULT OF MOVE IS INCORRECT, BECAUSE THERE IS OTHER THAN NUMERIC ITEM IN THE SENDING SIDE OF THE MOVE STATEMENT.

【原因】

受け取り側が数字項目のとき、送り出し側に次の項目を指定している。

- (1) 英数字項目
- (2) 英数字定数
- (3) ALL 定数

【コンパイラの処理】

送り出し側のデータを符号なし整数と見なして、数字転記を行う。

O J0237 実行効率を低下させる記述があります。

【Enterprise for Linux】THERE IS THE SPECIFICATION FOR DECREASE  
OBJECT-CODE EFFICIENCY.

【原因】

実行効率を低下させる記述がある。

F J0240 本機能は未サポート機能です。

【Enterprise for Linux】THIS FUNCTION IS NOT SUPPORTED.

【原因】

本機能は未サポート機能である。

W J0241 【Standard】?句は無視されます。

【原因】 DISPLAY 文に呼び名が GDD に関連付けられている UPON 句と、WITH NO  
ADVANCING 句を指定している。

【埋め込み情報】 WITH NO ADVANCING

【コンパイラの処理】 WITH NO ADVANCING 句は無視される。

W J0242 【Standard】?文の作用対象の長さが59文字を超えた、または超える可能性があります。

- 【原因】
- (1) DISPLAY 文の UPON 句の呼び名が GDD に関連付けられているとき、作用対象の桁数が59桁を超える、または超える可能性がある。
  - (2) STOP 文の定数の桁数が59桁を超えている。

- 【埋め込み情報】
- (1) DISPLAY
  - (2) STOP

[コンパイラの処理] 先頭 59 桁を有効とする。

F J0302 OUTPUT 句は指定できません。

【Enterprise for Linux】THE OUTPUT CLAUSE CANNOT BE SPECIFIED.

[原因]

順呼び出し法の乱ファイルに対する OPEN 文に OUTPUT 句を指定した。

F J0303 ? 文は指定できません。

【Enterprise for Linux】THE ? STATEMENT CANNOT BE SPECIFIED.

[原因]

順呼び出し法の乱ファイルに対する WRITE 文を指定した。

[埋め込み情報]

(1) WRITE

F J0312 ACCEPT 文の作用対象が `P` を含まない符号付き整数基本項目ではありません。

【Enterprise for Linux】THE OPERAND OF THE ACCEPT STATEMENT IS NOT SIGNED INTEGER ITEM EXCEPTED PICTURE CHARACTER `P`.

[原因]

ACCEPT 文の作用対象が `P` を含まない符号付き整数項目でない。

F J0313 データ項目が 12 桁以上の英数字項目ではありません。

【Enterprise for Linux】THE DATA ITEM IS NOT ALPHANUMERIC ITEM EXCEEDING 12 CHARACTERS.

[原因]

データ項目が 12 桁以上の英数字項目でない。

F J0314 データ項目が (13 + キーデータ項目の長さ) 桁以上の英数字項目ではありません。

【Enterprise for Linux】THE DATA ITEM IS NOT ALPHANUMERIC ITEM EXCEEDING (13 + LENGTH OF THE KEY DATA ITEM) CHARACTERS.

[原 因]

データ項目が (13 + キーデータ項目の長さ) 桁以上の英数字項目でない。

- F J0316 データ名の指定が誤っています。  
【Enterprise for Linux】THE DATA NAME IS INVALID.

[原 因]

データ名が主レコードキーまたは副レコードキーでない。

- F J0317 RANGE-NUMBER 句の作用対象が整数基本項目ではありません。  
【Enterprise for Linux】THE OPERAND OF THE RANGE-NUMBER CLAUSE IS NOT INTEGER ITEM.

[原 因]

RANGE-NUMBER 句の作用対象が整数基本項目でない。

- F J0318 RANGE-KEY または RANGE-NUMBER 句の作用対象の数が 64 個を超えています。  
【Enterprise for Linux】THE NUMBER OF DATA ITEM IN RANGE-KEY/RANGE-NUMBER CLAUSE EXCEEDS 64.

[原 因]

RANGE-KEY 句と RANGE-NUMBER 句の作用対象の数が 64 個を超えている。

- F J0319 COMMIT 文の作用対象が `P` を含まない整数基本項目ではありません。  
【Enterprise for Linux】THE OPERAND OF THE COMMIT STATEMENT IS NOT INTEGER ITEM EXCEPTED PICTURE CHARACTER `P`.

[原 因]

COMMIT 文の作用対象が `P` を含まない整数基本項目でない。

- F J0351 算術式またはブール式の書き方が誤っています。  
【Enterprise for Linux】THE ARITHMETIC EXPRESSION OR THE BOOLEAN EXPRESSION CONTAINS ILLEGAL SYNTAX.

[原 因]

算術式またはブール式の書き方が誤っている。

- F J0352      ブール式の作用対象が誤っています。  
              【Enterprise for Linux】THE OPERANDS OF BOOLEAN EXPRESSION IS INVALID.

【原 因】

ブール式の作用対象が誤っている。

- F J0353      ADDRESS OF 句の作用対象が誤っています。  
              【Enterprise for Linux】THE OPERAND OF "ADDRESS OF" CLAUSE IS INCORRECT.

【原 因】

ADDRESS OF 句の作用対象に次の項目を指定している。

- (1) オブジェクト参照項目
- (2) インスタンス定義やファクトリ定義の作業場所節やファイル節で定義された項目
- (3) ブールデータ項目

- F J0354      C\_NOBMP システムサブルーチンの指定が誤っています。  
              【Enterprise for Linux】THE ? SYSTEM SUBROUTINE CONTAINS INVALID DESCRIPTION.

【原 因】

C\_NOBMP システムサブルーチンの指定が誤っている。

- F J0362      活性化する側と活性化される側の引数の個数が異なります。  
              【Enterprise for Linux】NUMBER OF PARAMETERS FOR ACTIVATION CONFLICTS.

【原 因】

引数の個数が異なる。

- F J0363      活性化する側と活性化される側の引数が適合していません。CODE=?, NO=?。  
              【Enterprise for Linux】PARAMETERS FOR ACTIVATION DOES NOT CONFORM.  
              CODE=?, NO=?.

【原 因】

引数が不整合である。



[埋め込み情報 1]

不適合となった条件を識別できる詳細番号

- (1) 2 (字類または項類の組み合わせが不正)
- (2) 3 (仮引数を実引数よりバイト数が多い)
- (3) 4 (MOVE 文の規則に反する)
- (4) 5 (FACTORY 指定の有無が異なる)
- (5) 6 (INVOKE 文に SELF や SUPER, ACTIVE-CLASS 指定を伴ったオブジェクト参照を用いていない)
- (6) 8 (PICTURE, USAGE, SIGN, JUSTIFIED, BLANK WHEN ZERO が異なる)
- (7) 9 (通貨文字列が異なる)
- (8) 10 (環境部の DECIMAL-POINT IS COMMA 指定の有無が異なる)
- (9) 12 (SET 文の規則に反する)
- (10) 13 (COMPUTE 文の規則に反する)
- (11) 15 (引数の BY REFERENCE や BY VALUE の指定が異なる)

[埋め込み情報 2]

不適合となった引数が何番目かの数字

W J0364 返却項目を指定していません.

**【Enterprise for Linux】MISSING RETURNING ITEM.**

[原因]

活性化される側に返却項目の指定があるのに、活性化する側に返却項目の指定がない。

[コンパイラの処理]

返却項目がないものとして処理を継続する。

F J0365 活性化される側に返却項目はありません.

**【Enterprise for Linux】NO RETURNING ITEM IN ACTIVATION.**

[原因]

活性化する側に返却項目の指定があるのに、活性化される側に返却項目の指定がない。

F J0366 活性化する側と活性化される側の返却項目が適合していません. CODE=?.

**【Enterprise for Linux】RETURNING ITEM DO NOT CONFORM IN ACTIVATION.**

CODE=?.

[原因]

返却項目が不整合である。

[埋め込み情報]

不適合となった条件を識別できる詳細番号

- (1) 8 (PICTURE, USAGE, SIGN, JUSTIFIED, BLANK WHEN ZERO が異なる)
- (2) 9 (通貨文字列が異なる)
- (3) 10 (環境部の DECIMAL-POINT IS COMMA 指定の有無が異なる)
- (4) 12 (SET 文の規則に反する)
- (5) 14 (返却項目の長さが異なる)

F J0367 メソッドが見つかりません。

**[Enterprise for Linux]METHOD DOES NOT FOUND.**

[原因]

**INVOKE** 文の一意名 1 またはクラス名 1 で指定したファクトリ定義やインスタンス定義中に、指定したメソッドが見つかりません。

条件および検索規則は次のとおり。検索順に検索対象 (矢印の右) を記す。

(1) 一意名 1

(a) オブジェクト参照

① **FACTORY** クラス名 のとき

→ 指定クラスのファクトリ定義の定義中

→ 指定クラスのファクトリ定義の継承元

② クラス名 のとき

→ 指定クラスのインスタンス定義の定義中

→ 指定クラスのインスタンス定義の継承元

③ **FACTORY ACTIVE-CLASS** のとき

→ **INVOKE** 文を含むクラスのファクトリ定義の定義中

→ **INVOKE** 文を含むクラスのファクトリ定義の継承元

④ **ACTIVE-CLASS** のとき

→ **INVOKE** 文を含むクラスのインスタンス定義の定義中

→ **INVOKE** 文を含むクラスのインスタンス定義の継承元

(b) 既定義オブジェクト参照 **SELF**

① 現在のメソッドがファクトリメソッド のとき

→ 現在のメソッドと同じクラスのファクトリ定義の定義中

→ 現在のメソッドと同じクラスのファクトリ定義の継承元

- ② 現在のメソッドがインスタンスメソッドのとき
  - 現在のメソッドと同じクラスのインスタンス定義の定義中
  - 現在のメソッドと同じクラスのインスタンス定義の継承元
- (c) 既定義オブジェクト参照 **SUPER**
  - ① 現在のメソッドがファクトリメソッドのとき
    - 現在のメソッドと同じクラスのファクトリ定義の継承元
  - ② 現在のメソッドがインスタンスメソッドのとき
    - 現在のメソッドと同じクラスのインスタンス定義の継承元
- (d) 既定義オブジェクト参照 クラス名 1 **OF SUPER**
  - ① 現在のメソッドがファクトリメソッドのとき
    - 現在のクラスの **INHERITS** 句に指定されたクラス名 1 のファクトリ定義の定義中
    - 現在のクラスの **INHERITS** 句に指定されたクラス名 1 のファクトリ定義の継承元
  - ② 現在のメソッドがインスタンスメソッドのとき
    - 現在のクラスの **INHERITS** 句に指定されたクラス名 1 のインスタンス定義の定義中
    - 現在のクラスの **INHERITS** 句に指定されたクラス名 1 のインスタンス定義の継承元
- (2) クラス名 1
  - 指定クラスのファクトリ定義の定義中
  - 指定クラスのファクトリ定義の継承元

**F J0368**      **INVOKE** 文の作用対象が誤っています。  
**【Enterprise for Linux】OPERAND FOR INVOKE STATEMENT IS INCORRECT.**

[原因]

- (1) **INVOKE** 文の一意名 1 がオブジェクト参照でない。
- (2) **INVOKE** 文の一意名 1 が非限定オブジェクト参照である。

**F J0369**      **SUPER** に指定されたクラス名は継承元に存在しません。  
**【Enterprise for Linux】THE CLASS-NAME SPECIFIED "SUPER" IS NOT FOUND IN INHERITANCE BASE.**

[原因]

**SUPER** に指定されたクラス名が現在のクラスの **INHERITS** 句に指定されたクラス名ではありません。

ません。

F J0370      オブジェクトプロパティの作用対象が誤っています。  
              【Enterprise for Linux】OPERAND FOR OBJECT PROPERTY IS INCORRECT.

【原 因】

オブジェクトプロパティの作用対象が、次のいずれでもない。

- (1) オブジェクト参照項目 (非限定オブジェクト参照を除く)
- (2) 既定義オブジェクト参照 SELF
- (3) 既定義オブジェクト参照 SUPER
- (4) クラス名
- (5) プロパティ名

F J0371      オブジェクトプロパティが存在しません。  
              【Enterprise for Linux】OBJECT PROPERTY IS NOT FOUND IN REPOSITORY  
              INFORMATION.

【原 因】

リポジトリ情報中に該当するプロパティ名が存在しません。

F J0373      オブジェクトプロパティが誤っています。  
              【Enterprise for Linux】OBJECT PROPERTY IS INCORRECT.

【原 因】

オブジェクトプロパティとリポジトリ情報中の該当するプロパティメソッド情報が矛盾しています。

F J0374      オブジェクトビューの作用対象が誤っています。  
              【Enterprise for Linux】OPERAND FOR OBJECT VIEW IS INCORRECT.

【原 因】

オブジェクトビューの作用対象の字類がオブジェクトでない。

F J0375      引数が誤っています。  
              【Enterprise for Linux】PARAMETERS IS INCORRECT.

【原因】

- (1) 組込み関数の引数の型がその関数に対して許されない型である。
- (2) 利用者定義関数の引数に指定された一意名はファイル節, 作業場所節, 局所記憶節, ベースデータ節または連絡節中で定義された項類がブール以外でなければならない。
- (3) 利用者定義関数の引数が次のいずれかである。
  - (a) ブールデータ項目
  - (b) ポインタ名
  - (c) BY VALUE 指定の仮パラメータに対応する引数の字類が数字, オブジェクトまたはポインタ以外である。
  - (d) BY REFERENCE 指定の仮パラメータに対応する引数が NULL である。

F J0376 INITIALIZE 文の初期化指定が誤っています。  
 【Enterprise for Linux】ILLEGAL SET SPECIFICATION IN INITIALIZE STATEMENT.

【原因】

INITIALIZE 文の初期化作用対象の基本項目と BY 句で指定したオブジェクト参照項目が SET 文の規則に違反している。

F J0377 【Standard】?文の KEY 句の作用対象がファイルの KEY でない。

- 【原因】
- (1) 索引ファイルに対し, READ 文, START 文の KEY 句のデータ名がそのファイルに対して指定されたキーデータ名 (主レコードキー, 副レコードキーまたは部分キー (START 文の場合)) でない。
  - (2) 相対ファイルに対し, START 文の KEY 句のデータ名がそのファイルに対して指定された RELATIVE KEY データ名でない。
  - (3) READ 文, START 文の KEY 句のデータ名が定数名である。

- 【埋め込み情報】
- (1) READ
  - (2) START

W K0002 演算の中間結果の格納領域が 30 桁を超えています。  
 【Enterprise for Linux】THE INTERIM RESULT WILL EXCEED 30 BYTES.

【原因】

演算の中間結果の格納領域が 30 桁を超えている。

[コンパイラの処理]

中間結果格納領域の生成規則に基づいて処理する。

F L0001      日本語定数の内容が誤っています。  
              【Enterprise for Linux】THE GRAPHIC LITERAL CONTAINS INVALID  
              DESCRIPTION.

[原因]

- (1)    日本語定数で指定された文字列が許されない文字列である。(内部コード変換エラー)
- (2)    日本語 16 進定数の値が誤っている。(内部コード変換エラー)

W L0002      演算の中間結果の格納領域が 30 桁を超えています。  
              【Enterprise for Linux】THE INTERIM RESULT WILL EXCEED 30 BYTES.

[原因]

演算の中間結果の格納領域が 30 桁を超えている。

[コンパイラの処理]

中間結果格納領域の生成規則に基づいて処理する。

F L0003      演算の中間結果の転記桁数が 18 桁を超えています。  
              【Enterprise for Linux】THE INTERIM RESULT TRANSFER WILL EXCEED 18 BYTES.

[原因]

演算の中間結果の転記桁数が 18 桁を超えている。

F Q0002      診断メッセージが多すぎます。  
              【Enterprise for Linux】TOO MUCH DIAGNOSTIC MESSAGE.

[原因]

翻訳時の診断メッセージが 65535 個以上あり多すぎます。

W S\*241      B 領域に書くべき語が A 領域にあります。  
              【Enterprise for Linux】THE WORD MUST BEGIN WITH AREA B.

[原因]

B 領域より始まるべき語が A 領域より始まっている。

[コンパイラの処理]

そのまま翻訳を続行する。

**W S\*242**      A 領域に書くべき語が B 領域にあります.  
                  【Enterprise for Linux】THE WORD MUST BEGIN WITH AREA A.

[原 因]

A 領域に記述すべき語が B 領域から始まっている。

[コンパイラの処理]

そのまま翻訳を続行する。

**F S\*246**      見出し部の終り方が誤っています.  
                  【Enterprise for Linux】THE ENVIRONMENT DIVISION BEGINS BEFORE THE  
IDENTIFICATION DIVISION ENDS.

[原 因]

見出し部の終り方が誤っている。

**F S\*247**      環境部の終り方が誤っています.  
                  【Enterprise for Linux】THE DATA DIVISION BEGINS BEFORE THE ENVIRONMENT  
DIVISION ENDS.

[原 因]

環境部の終り方が誤っている。

**F S\*248**      データ部の終り方が誤っています.  
                  【Enterprise for Linux】THE PROCEDURE DIVISION BEGINS BEFORE THE DATA  
DIVISION ENDS.

[原 因]

データ部の終り方が誤っている。

**F S\*249**      手続き部の終り方が誤っています.

**【Enterprise for Linux】THE SOURCE PROGRAM EXHAUSTS BEFORE THE PROCEDURE DIVISION ENDS.**

〔原因〕

手続き部の終り方が誤っている。

F S\*250 定数が誤っています。

**【Enterprise for Linux】THE LITERAL IS INVALID.**

〔原因〕

(1) 構文的に定数が許されない所に定数を記述した。

F S\*251 `?'が誤っています。

**【Enterprise for Linux】THE `?' IS INCORRECT.**

〔原因〕

正当でない記述があった。

〔埋め込み情報〕

正当でない語

F S\*252 PICTURE 句の文字列が誤っています。

**【Enterprise for Linux】THE PICTURE CHARACTER STRING IS INCORRECT.**

〔原因〕

PICTURE 句の文字列の記述場所が誤っている。



## B.2 翻訳時のメッセージ

コンパイラは、入出力エラーなどが発生すると、標準エラー出力に翻訳時メッセージを出力する。次にメッセージ本文と原因を記述する。

【Enterprise for Linux】システム環境変数 `LANG=ja_JP.SJIS` でない場合、英文の翻訳時メッセージを出力する。

LCB002      必須製品が正しくインストールされていません。

【Enterprise for Linux】Necessary products are not correctly installed.

〔原因〕

コンパイラが動作するための必須製品が正しくインストールされていない。

LCB010      (形式 1)

内部コンパイラエラーが発生しました。PROCESS=?, SOURCE FILE=?, LINE=?,  
CODE=?.

【Enterprise for Linux】An inside compiler error occurred. PROCESS=?,  
SOURCE FILE=?, LINE=?, CODE=?.

(形式 2)

内部コンパイラエラーが発生しました。PROCESS=?, SOURCE FILE=?, CODE=?.

【Enterprise for Linux】An inside compiler error occurred. PROCESS=?,  
SOURCE FILE=?, CODE=?.

〔埋め込み情報〕

(形式 1) プロセス名, ソースファイル名, 内部行番号, エラーコード

(形式 2) プロセス名, ソースファイル名, エラーコード

〔原因〕

コンパイラが翻訳できない状態になったときに出力され、この時点で翻訳処理を打ち切る。

LCB011      ?

〔埋め込み情報〕

「十分なスペースがありません」等のシステムが出力するエラーメッセージ

[原 因]

システム関数またはライブラリ関数の呼び出し時にエラーが発生した。

LCB012      ?. FILE=?

[埋め込み情報]

「パーミッションがありません」等のシステムが出力するエラーメッセージ,  
ファイル名

[原 因]

ファイル入出力関連のシステム関数またはライブラリ関数の呼び出し時にエラーが発生した。

LCB013      作業ファイルの獲得に失敗しました。  
【Enterprise for Linux】It failed in the acquisition of a file to use temporarily.

[原 因]

作業ファイル名の作成に失敗した。

LCB020      ソース原文または登録集原文の文字位置が, 限界を超えています. FILE=?,  
LINE=?  
【Enterprise for Linux】The character position in source text or library text exceeds limit. FILE=?, LINE=?

[埋め込み情報]

ファイル名, 内部行番号

[原 因]

- (1) 固定形式のソース原文または登録集原文の文字位置が, 256 を超えている。
- (2) 自由形式のソース原文または登録集原文の文字位置が, 255 を超えている。

LCB025      書き方が複雑すぎて翻訳できません。  
【Enterprise for Linux】The statement is too complex to compile.

[原因]

コンパイラが翻訳できない状態になったときに出力され、この時点で翻訳処理を打ち切る。

LCB030      ‘?’ オプションの指定は誤っています。

【Enterprise for Linux】The option designation of `?'` is wrong.

[埋め込み情報]

オプション文字列

[原因]

翻訳オプションパラメータの指定が正しくない。

LCB031      ‘?’ と ‘?’ は同時に指定できません。

【Enterprise for Linux】`?'` and `?'` can't be specified at the same time.

[埋め込み情報]

オプション文字列 1, オプション文字列 2

[原因]

2 つの翻訳オプションパラメータは同時に指定できない。

LCB032      `?'` と `?'` は同時に指定できません。`?'` は無効です。

【Enterprise for Linux】`?'` and `?'` can't be specified at the same time,  
`?'` ignored.

[埋め込み情報 1]

(1) -G

(2) 【Enterprise for Windows】【Standard】-Hi

[埋め込み情報 2]

(1) -M

(2) 【Enterprise for Windows】【Standard】-Ce

[埋め込み情報 3]

(1) -M

(2) 【Enterprise for Windows】【Standard】-Ce

[原因]

- (1) 同時指定できない`-G`と`-M`のオプションを同時に指定した。
- (2) **【Enterprise for Windows】【Standard】**同時指定できない`-Hi`と`-Ce`のオプションを同時に指定した。

**LCB040**      ファイル名またはパス名の長さが?バイトを超えています. **FILE=?**  
**【Enterprise for Linux】**A file name head exceeds ? bytes. **FILE = ?**

[埋め込み情報]

ファイル名長, ファイル名

[原因]

ファイル名長が制限を超えている。(ファイル名長はディレクトリ名長も含む)

**LCB041**      **【Enterprise for Linux】**‘?’オプションに拡張子‘?’のファイルは指定できません. **FILE=?**  
**【Enterprise for Linux】**The file of the extension ‘?’ can't be specified in the ‘?’ option. **FILE = ?**

[埋め込み情報]

オプション文字列, 拡張子, ファイル名

[原因]

指定されたファイルは上書きされる可能性があるために指定できない。

**LCB050**      **【Enterprise for Linux】**用語機能のライブラリが見つかりません(?)  
**【Enterprise for Linux】**The library for Term Module is missing(?).

[埋め込み情報]

用語機能のライブラリ名

[原因]

用語機能のライブラリが見つからない。

**LCB051**      **【Enterprise for Linux】**用語機能のライブラリが誤っています(?)  
**【Enterprise for Linux】**The library for Term Module is invalid(?).

[埋め込み情報]

用語機能のライブラリ名

[原因]

指定した機能が使えない。

**LCB060**      リポジトリ情報にエラーが発生しました **CODE=?**  
                 **【Enterprise for Linux】Error in repository information. CODE=?**

[埋め込み情報]

- (1) 2 (製品構成ファイル **CLS\_BASE.rpg** が存在しない)
- (2) 10 (外部リポジトリファイルの形式が不正である)
- (3) 11 (外部リポジトリファイルの書込に失敗した)
- (4) 12 (外部リポジトリファイルの読込に失敗した)
- (5) 17 (外部リポジトリファイル名が長すぎる)
- (6) 19 (外部リポジトリファイル操作の準備に失敗した)
- (7) 20 (製品構成ファイル **CobolRepositorySchema.xsd** が存在しない)

[原因]

リポジトリ情報を処理する際にエラーが発生した。

**LCB061**      **【Enterprise for Linux】**外部リポジトリ機能のライブラリが見つかりません(?)  
                 **【Enterprise for Linux】The library for External Repository Module is**  
                 **missing (?)**.

[埋め込み情報]

外部リポジトリ機能のライブラリ名

[原因]

- (1) **XML** パーサがインストールされていない。
- (2) 共有ライブラリに関する環境設定が不正である。

**LCB062**      **【Enterprise for Linux】**外部リポジトリ機能のライブラリが誤っています(?)  
                 **【Enterprise for Linux】The library for External Repository Module is**  
                 **invalid (?)**.

[埋め込み情報]

外部リポジトリ機能のライブラリ名

[原因]

指定した機能が使えない。

LCB063      **【Standard】**`-c`と`-Yi`は同時に指定できません. `-Yi`は無効です.

[原因]

同時指定できない`-c`と`-Yi`のオプションを同時に指定した。

LCB064      **【Standard】**`-G`と`-Yi`は同時に指定できません. `-Yi`は無効です.

[原因]

同時指定できない`-G`と`-Yi`のオプションを同時に指定した。

LCB065      **【Standard】**生成オブジェクトの作成でコンパイラエラーが発生しました. CODE  
=?, SOURCE FILE=?.

[埋め込み情報]

エラーコード, ソースファイル名

[原因]

生成オブジェクトの作成に失敗した。この時点で翻訳処理を打ち切る。

LCB066      **【Standard】**`-M`または`-Mw`が指定されていません.

[原因]

`-Yr`または`-Yi`指定時に必要な`-M`または`-Mw`が指定されていません.

LCB067      **【Standard】**`-H[cdpswx]`が指定されていません.

[原因]

`-Yf`指定時に必要な`-H[cdpswx]`が指定されていません.

LCB069      デバッグ情報ファイルとオブジェクトプログラムが一致しません. FILE=?, ?

[埋め込み情報]

デバッグ情報ファイル名, オブジェクトプログラム名

[原因]

デバッグ情報ファイルとオブジェクトプログラム内容が一致しない。

# C 実行時メッセージ一覧

## C.1 実行時エラー

実行時にエラーが発生した場合は標準エラー出力にエラーメッセージが表示される。

[Enterprise for Linux] 以下の条件をすべて満たす場合は英文のエラーメッセージが表示される。

- 環境変数 LANG=ja\_JP.SJIS でない
- 環境変数 LANG=ja\_JP.UTF-8 でないか環境変数 COB\_CODECHG=ON でない

エラーメッセージ表示後、リターンキーにて処理の継続または強制終了する。なお、次の条件に合致する場合は、動作が変わる。

例外節の記述がある場合、次の条件により動作が変わる。どの場合でも、リターンキーの入力待ちは行わない。詳細は「第 12 章 例外処理機能」を参照。

| エラー番号  | 動作                    |
|--------|-----------------------|
| COB160 | メッセージ表示後、そのまま処理を継続する。 |
| 上記以外   | メッセージ表示後、例外節の処理を行う。   |

エラーメッセージの表示および表示後の動作は環境変数 COB\_RUNERR の設定の影響を受ける。

| COB_RUNERR の値 | 動作                            |
|---------------|-------------------------------|
| NO_PAUSE      | エラーメッセージ表示後リターンキーの入力待ちを行わない。  |
| NO_DISPLAY    | エラーメッセージの表示とリターンキーの入力待ちを行わない。 |

環境変数 COB\_RUNERR の設定によりエラーメッセージの表示抑制や確認入力待ちを行わないように変更できるため、エラー発生時でも処理を継続することができる。



例 1:

COByyy エラーメッセージ本文 プログラム名=zzzzz 行番号=nnnnnn

例 2:

[Standard] COBOL WS エミュレータのとき

COByyy [zzzzz/nnnnnn] エラーメッセージ本文 (埋め込み情報/ccccc)

説明

- (1) yyy にはエラーメッセージ番号が設定される。
- (2) エラーメッセージは、エラーメッセージ本文と埋め込み情報で構成される。
- (3) zzzzz にはプログラム名が設定される。
- (4) nnnnnn には内部行番号を設定する。ただし、-Ce オプション指定ありの場合は外部行番号を設定する。
- (5) ccccc にはエラーコード、システムエラー、関連製品エラー番号の詳細情報値が設定される。
- (6) エラーメッセージの形式や埋め込み情報の内容や意味はエラーメッセージ番号毎に異なる。プログラム名 (zzzzz)、行番号 (nnnnnn) 以外の埋め込み情報の詳細は表 C-1 を参照する。
- (7) [Standard] COBOL WS エミュレータでは、メッセージの長さが 59 バイトを超過する場合、エラーメッセージ本文、埋め込み情報は出力しない。サーバ側に環境変数 COB\_EVENTLOG=WSEML を指定することで、エラーメッセージ全文をイベントログに出力することができる。

次に実行時エラーを一覧で示す。

表 C-1 実行時エラー

| エラー番号  | エラーメッセージの説明                                                                                                                            |
|--------|----------------------------------------------------------------------------------------------------------------------------------------|
| COB000 | 内部エラーが発生しました。(プログラム名=zzzzz, 行番号=nnnnnn)<br>[Enterprise for Linux] INTERNAL ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)                    |
| [原因]   | 内部エラーが発生した。                                                                                                                            |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                  |
| [対策]   | 弊社担当者に問い合わせる。                                                                                                                          |
| COB002 | 添字、可変繰り返し数または部分参照の限界を超えました。(プログラム名=zzzzz, 行番号=nnnnnn)<br>[Enterprise for Linux] DATA RANGE ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)    |
| [原因]   | (1) 添字が表要素の範囲内でない。<br>(2) 部分参照の最左端または長さが一意名の領域内でない。                                                                                    |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                  |
| [対策]   | (1) 添字に指定したデータ名の値が、表要素の範囲を超えないようにする。<br>(2) 部分参照の左端文字位置、長さに指定した値が、一意名の領域外を参照しないようにする。                                                  |
| COB003 | 可変長集団項目の繰り返し数が範囲を超えました。(プログラム名=zzzzz, 行番号=nnnnnn)<br>[Enterprise for Linux] DATA RANGE ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)        |
| [原因]   | DEPENDENT ON 句のデータ名の値が、可変繰り返し数の最小値から最大値の範囲内でない。                                                                                        |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                  |
| [対策]   | DEPENDENT ON 句のデータ名の値が、可変繰り返し数の最小値から最大値の範囲外にならないようにする。                                                                                 |
| COB004 | GO TO の飛び先が設定されていません。(プログラム名=zzzzz, 行番号=nnnnnn)<br>[ Enterprise for Linux ] GO TO ADDRESS NOT FOUND(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn) |
| [原因]   | 手続き名を省略した GO TO 文の実行前に必要な ALTER 文が実行されていない。                                                                                            |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                  |
| [対策]   | GO TO 文に手続き名を指定するか、GO TO 文の実行前に必要な ALTER 文を実行する。                                                                                       |

|        |                                                                                                                                                                                                                                               |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB006 | INSPECT で長さ不一致のため置換できません。(プログラム名=zzzzz, 行番号=nnnnnn)<br>[ Enterprise for Linux ] CANNOT REPLACE ACCORDING TO THE UNMATCH OF LENGTH(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)                                                                          |
| [原因]   | (1) CONVERTING 指定の INSPECT 文において, T0 前後の作用対象の長さが異っている。<br>(2) REPLACING 指定の INSPECT 文において, BY 前後の作用対象の長さが異っている。                                                                                                                               |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                         |
| [対策]   | (1) CONVERTING 指定の INSPECT 文において, T0 前後の作用対象の長さを揃える。<br>(2) REPLACING 指定の INSPECT 文において, BY 前後の作用対象の長さを揃える。                                                                                                                                   |
| COB007 | 環境変数 COB_LIBRARY が誤っています。(プログラム名=zzzzz, 行番号=nnnnnn)<br>[ Enterprise for Linux ] ENVIRONMENT COB_LIBRARY<br>ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)                                                                                           |
| [原因]   | 環境変数 COB_LIBERR の指定内容が CONTINUE ではなく, かつ, 環境変数 COB_LIBRARY<br>の指定内容が次のいずれかである。<br>[Enterprise for Windows] [Standard]<br>(1) 先頭に区切り文字「;」がある。<br>(2) 区切り文字「;」が連続している。<br>[Enterprise for Linux]<br>(1) 先頭に区切り文字「:」がある。<br>(2) 区切り文字「:」が連続している。 |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                         |
| [対策]   | 環境変数 COB_LIBERR に CONTINUE を指定するか, 環境変数 COB_LIBRARY に正しい内容を<br>を設定する。                                                                                                                                                                         |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB008      | CALL で呼び出すエントリが見つかりません。(プログラム名=zzzzz, 行番号=nnnnnn)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|             | <p data-bbox="483 369 1532 448">[ Enterprise for Linux ] CALL ENTRY NOT FOUND(PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p data-bbox="483 459 1532 504"><b>[原因]</b></p> <p data-bbox="483 515 1532 560">(1) CALL 一意名で指定したサブプログラム名 (エントリ名) の途中で空白文字がある。</p> <p data-bbox="483 571 1532 638">(2) [Enterprise for Windows] [Standard] 環境変数 COB_LIBRARY に指定した DLL が存在しない。または環境変数 COB_LIBRARY に指定した DLL が呼び出す DLL が存在しない。</p> <p data-bbox="483 649 1532 784">[Enterprise for Linux] 環境変数 COB_LIBRARY に指定した共有ライブラリが存在しない。または環境変数 COB_LIBRARY に指定した共有ライブラリが呼び出す共有ライブラリが存在しない。</p> <p data-bbox="483 795 1532 929">(3) [Enterprise for Windows] [Standard] CALL 一意名で指定したサブプログラム名 (エントリ名) が、実行モジュール(EXE)にも、環境変数 COB_LIBRARY に指定した DLL にもない。また、プログラムの作りによっては以下の場合にも発生することがある。</p> <p data-bbox="483 940 1532 1075">[Enterprise for Linux] CALL 一意名で指定したサブプログラム名 (エントリ名) が、実行モジュールにも、環境変数 COB_LIBRARY に指定した共有ライブラリにもない。また、プログラムの作りによっては以下の場合にも発生することがある。</p> <p data-bbox="483 1086 1532 1153">(4) [Enterprise for Windows] [Standard] 環境変数 COB_LIBRARY に指定した DLL がアンロードされた。</p> <p data-bbox="483 1164 1532 1232">[Enterprise for Linux] 環境変数 COB_LIBRARY に指定した共有ライブラリがアンロードされた。</p> <p data-bbox="483 1243 1532 1288">(5) コンパイラ内部で保持しているハンドル情報を破壊した。</p> |
| <b>[処置]</b> | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>[対策]</b> | <p data-bbox="483 1355 1532 1444">(1) CALL 一意名で指定したサブプログラム名 (エントリ名) の途中で空白文字がないようにする。</p> <p data-bbox="483 1456 1532 1691">(2) [Enterprise for Windows] [Standard] 環境変数 COB_LIBERR=CONTINUE を指定していない場合、環境変数 COB_LIBRARY に一意名で指定したサブプログラムを含む DLL 名だけでなく、その DLL 名より前に指定した DLL 名も正しく設定する。環境変数 COB_LIBRARY に指定した DLL およびその DLL が呼び出す DLL がすべて DLL の検索パスに存在しているか確認する。</p> <p data-bbox="483 1702 1532 1971">[Enterprise for Linux] 環境変数 COB_LIBERR=CONTINUE を指定していない場合、環境変数 COB_LIBRARY に一意名で指定したサブプログラムを含む共有ライブラリ名だけでなく、その共有ライブラリ名より前に指定した共有ライブラリ名も正しく設定する。環境変数 COB_LIBRARY に指定した共有ライブラリおよびその共有ライブラリが呼び出す共有ライブラリがすべて共有ライブラリの検索パスに存在しているか確認する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <p>(3) [Enterprise for Windows] [Standard] CALL 一意名で指定したサブプログラム名 (エントリ名) が, 実行モジュール(EXE)や環境変数 COB_LIBRARY に指定した DLL に存在するか次の点を確認する。</p> <p>[Enterprise for Linux] CALL 一意名で指定したサブプログラム名 (エントリ名) が, 実行モジュールや環境変数 COB_LIBRARY に指定した共有ライブラリに存在するか次の点を確認する。</p> <ul style="list-style-type: none"> <li>・CALL 一意名で指定したサブプログラム名 (エントリ名) が大文字／小文字を含め, 呼び出される側のエントリ名と一致しているか。</li> <li>・CALL 一意名で指定したサブプログラム名 (エントリ名) が実行モジュール内にある場合, 実行モジュール作成時にコンパイルオプション-CY を指定しているか。</li> </ul> |
| COB010 | 不正 10 進データを検出しました。(プログラム名=zzzzz, 行番号=nnnnnn)                                                                                                                                                                                                                                                                                                                                                                                                                                |
| [原因]   | <p>[Enterprise for Linux] DATA EXCEPTION(PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>数字項目に 10 進数以外の不正なデータを検出した。</p> <p>不正なデータについては, 「16.3 不正 10 進データ例外と COBOL 文」を参照する。</p>                                                                                                                                                                                                                                                                                                   |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| [対策]   | 数字項目の内容を確認する。                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB048 | <p>利用者制御変数を設定または参照した時にエラーが発生しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <pre>[ Enterprise for Linux ] USER CONTROL VARIABLE ACCESS ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</pre> <p>[原因]</p> <p>(1) 環境変数 COB_UC_FILE が定義されていない。<br/> (2) 環境変数 COB_UC_FILE に指定した利用者制御変数保持ファイルが存在しない。<br/> (3) 環境変数 COB_UC_FILE に指定した利用者制御変数保持ファイルのアクセス時にエラーが発生した。</p> <p>[埋め込み情報]</p> <p>利用者制御変数名=uuuuu, コード=cccccc</p> <p><b>備考</b></p> <p>uuuuu: 利用者制御変数名<br/> cccccc: エラーコード(16進)</p> <p>[処置]</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>[対策]</p> <p>(1) 環境変数 COB_UC_FILE を定義する。<br/> (2) cobinituc コマンドで利用者制御変数保持ファイルの作成が成功しているか, 環境変数 COB_UC_FILE に cobinituc コマンドで作成した利用者制御変数保持ファイル名を正しく指定しているか確認する。<br/> (3) 環境変数 COB_UC_FILE に指定した制御変数保持ファイルの状態を確認する。</p> |
| COB049 | <p>利用者制御変数が存在しません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <pre>[ Enterprise for Linux ] USER CONTROL VARIABLE IS UNKNOWN(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</pre> <p>[原因]</p> <p>環境変数 COB_UC_FILE に指定したファイル中に利用者制御変数が存在しない。</p> <p>[埋め込み情報]</p> <p>利用者制御変数名=uuuuu</p> <p><b>備考</b></p> <p>uuuuu: 利用者制御変数名</p> <p>[処置]</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>[対策]</p> <p>利用者制御変数を正しく設定する。</p>                                                                                                                                                                                                                                                                                                                                                  |

|              |                                                                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB093       | 関数の引数が 0 個です。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)<br>[ Enterprise for Linux ] THE NUMBER OF FUNCTION ARGUMENT IS ZERO(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn) |
| [原因]         | 組込み関数の引数がない。                                                                                                                                           |
| [埋め込み<br>情報] | 関数名=kkkkk<br><b>備考</b><br>kkkkk: 組込み関数名                                                                                                                |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                  |
| [対策]         | 組込み関数の引数を正しく指定する。                                                                                                                                      |
| COB094       | 不正 10 進データを検出しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)<br>[Enterprise for Linux] DATA EXCEPTION(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)                       |
| [原因]         | 組込み関数において, 数字項目に 10 進数以外の不正なデータを検出した。<br>不正なデータについては, 「16.3 不正 10 進データ例外と COBOL 文」を参照する。                                                               |
| [埋め込み<br>情報] | 関数名=kkkkk<br><b>備考</b><br>kkkkk: 組込み関数名                                                                                                                |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                  |
| [対策]         | 組込み関数の数字項目の内容を確認する。                                                                                                                                    |
| COB095       | 関数の引数の値が範囲外です。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)<br>[ Enterprise for Linux ] OUT OF DATA RANGE ON FUNCTION ARGUMENT(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn) |
| [原因]         | 組込み関数の引数の値が範囲内でない。                                                                                                                                     |
| [埋め込み<br>情報] | 関数名=kkkkk<br><b>備考</b><br>kkkkk: 組込み関数名                                                                                                                |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                  |
| [対策]         | 組込み関数の引数の値を正しく指定する。                                                                                                                                    |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB100 | <p>ファイルが指定されていません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] FILE NOT ASSIGNED(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因]<br/> (1) LOCK 指定の CLOSE 文で閉じた拡張ファイルに OPEN 文を実行した。<br/> (2) [Enterprise] RAF ファイルが未定義である。</p> <p>[埋め込み情報]<br/> ファイル名<br/> <b>備考</b><br/> ファイル名: ファイル名=実ファイル名<br/> (実ファイル名の場合)<br/> ファイル識別リテラル=ファイル識別リテラル<br/> (ファイル識別リテラルの場合)<br/> 省略される場合がある</p> <p>[処置]<br/> 解除キーを押せばプログラムは異常終了する。</p> <p>[対策]<br/> (1) 拡張ファイルに再度 OPEN 文を実行する場合は LOCK 指定の CLOSE 文を実行しない。<br/> (2) [Enterprise] refam コマンドを利用して RAF ファイルを定義する。</p> |
| COB101 | <p>ファイルが開けません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] FILE COULD NOT OPEN(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因]<br/> 拡張ファイルに対する OPEN 文でエラーが発生した。</p> <p>[埋め込み情報]<br/> ファイル名, コード=cccccc<br/> <b>備考</b><br/> ファイル名: ファイル名=実ファイル名<br/> (実ファイル名の場合)<br/> ファイル識別リテラル=ファイル識別リテラル<br/> (ファイル識別リテラルの場合)<br/> 省略される場合がある<br/> cccccc: 入出力動作の状態を示す詳細情報値(16 進)</p> <p>[処置]<br/> 解除キーを押せばプログラムは異常終了する。</p> <p>[対策]<br/> COBOL プログラミングの手引「4.5.2 入出力状態値と意味」の詳細情報値または関連製品 (refam) 利用の手引にて, OPEN 文が使用する拡張ファイルの状態を確認する。</p>                          |



|                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>COB102</b></p> <p>[原因]</p> <p>[埋め込み<br/>情報]</p> <p>[処置]</p> <p>[対策]</p> | <p>ファイルが開かれていません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] FILE NOT OPENED(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>開かれていない拡張ファイルに対して, 読み込み, 書き出し, またはファイル入出力用システムサブルーチンを実行した。</p> <p>ファイル名</p> <p><b>備考</b></p> <p>ファイル名: ファイル名=実ファイル名<br/>(実ファイル名の場合)<br/>ファイル識別リテラル=ファイル識別リテラル<br/>(ファイル識別リテラルの場合)</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>読み込み, 書き出し, またはファイル入出力用システムサブルーチンを実行する前に拡張ファイルを開いておく。</p>                                                                             |
| <p><b>COB103</b></p> <p>[原因]</p> <p>[埋め込み<br/>情報]</p> <p>[処置]</p> <p>[対策]</p> | <p>ファイル入出力中にエラーが発生しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] FILE I-O ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>拡張ファイルに対する OPEN 文以外でエラーが発生した。</p> <p>ファイル名, コード=cccccc</p> <p><b>備考</b></p> <p>ファイル名: ファイル名=実ファイル名<br/>(実ファイル名の場合)<br/>ファイル識別リテラル=ファイル識別リテラル<br/>(ファイル識別リテラルの場合)<br/>省略される場合がある</p> <p>cccccc: 入出力動作の状態を示す詳細情報値(16 進)</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>COBOL プログラミングの手引「4.5.2 入出力状態値と意味」の詳細情報値または関連製品 (refam) 利用の手引にて, 入出力文が使用する拡張ファイルの状態を確認する。</p> |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB104 | <p>ファイルがオーバーフローしました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] FILE OVERFLOW(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] 拡張ファイルの領域を超えてレコードを書き出した。</p> <p>[埋め込み情報] ファイル名</p> <p><b>備考</b></p> <p>ファイル名: ファイル名=実ファイル名<br/>(実ファイル名の場合)<br/>ファイル識別リテラル=ファイル識別リテラル<br/>(ファイル識別リテラルの場合)</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 十分な空き領域を持つディスクでアプリケーションを実行する。<br/>[Enterprise] RAF ファイルの場合, 容量の大きい拡張ファイルまたは自動拡張指定で作成したファイルを使用する。</p>                                                                                                                                    |
| COB105 | <p>CSV 形式のファイル読み込み中にエラーが発生しました。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] FILE(CSV) READ<br/>ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] CSV 形式のファイルを読み込むとき, 外部 10 進数に不正なデータを検出した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] CSV 形式のファイルの内容を確認する。</p>                                                                                                                                                                                                                                                                                                               |
| COB106 | <p>[Enterprise for Windows] [Standard]<br/>ファイルパス長は、246 バイト以下である必要があります。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux]<br/>ファイルパス長は、4094 バイト以下である必要があります。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] FILE PATH IS TOO LONG (&gt;4094) (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] [Enterprise for Windows] [Standard] ファイルパス長が 246 バイトを超過した。<br/>[Enterprise for Linux] ファイルパス長が 4094 バイトを超過した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] [Enterprise for Windows] [Standard] ファイルパス長を 246 バイト以下に修正する。<br/>[Enterprise for Linux] ファイルパス長を 4094 バイト以下に修正する。</p> |

|                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>COB107</p> <p>[原因]</p> <p>[埋め込み<br/>情報]</p> <p>[処置]</p> <p>[対策]</p> | <p>[Standard] ファイルとオプション指定が不整合である (埋め込み情報, プログラム名 = zzzzz, 行番号 = nnnnnn)</p> <p>(1) COBOL DB アクセスの一般索引ファイル利用時, -Cj または -CF オプションを指定した。</p> <p>(2) プリンタファイル (PRNAMR/PRNMAN 経由) 利用時, -Cj オプションを指定した。</p> <p>オプション = 00000</p> <p><b>備考</b></p> <p>00000: 次のいずれかを設定する</p> <p>(1) -Cj</p> <p>(2) -CF</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>(1) COBOL DB アクセスの一般索引ファイル利用時, -Cj または -CF オプションを指定しない。</p> <p>(2) プリンタファイル (PRNAMR/PRNMAN 経由) 利用時, -Cj オプションを指定しない。</p>                                                                                                                                                               |
| <p>COB135</p> <p>[原因]</p> <p>[処置]</p> <p>[対策]</p>                      | <p>外部スイッチの参照でエラーが発生しました。(プログラム名 = zzzzz, 行番号 = nnnnnn)</p> <p>[ Enterprise for Linux ] EXTERNAL SWITCH REFERENCE<br/>ERROR (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>(1) 環境変数 COB_SWITCH_FILE が定義されていない。</p> <p>(2) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイルが存在しない。</p> <p>(3) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイルのオープン, リードでエラーが発生した。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>(1) 環境変数 COB_SWITCH_FILE を定義する。</p> <p>(2) cobinitcv コマンドで SWITCH 状態保持ファイルの作成が成功しているか, 環境変数 COB_SWITCH_FILE に cobinitcv コマンドで作成した SWITCH 状態保持ファイル名を正しく指定しているか確認する。</p> <p>(3) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイルの状態を確認する。</p> |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB136 | <p>外部スイッチの設定でエラーが発生しました。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] EXTERNAL SWITCH SETTING<br/>ERROR(PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因]</p> <p>(1) 環境変数 COB_SWITCH_FILE が定義されていない。<br/> (2) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイルが存在しない。<br/> (3) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイル内容が壊れた。<br/> (4) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイルのオープン、ライトでエラーが発生した。</p> <p>[処置]</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>[対策]</p> <p>(1) 環境変数 COB_SWITCH_FILE を定義する。<br/> (2) cobinitcv コマンドで SWITCH 状態保持ファイルの作成が成功しているか、環境変数 COB_SWITCH_FILE に cobinitcv コマンドで作成した SWITCH 状態保持ファイル名を正しく指定しているか確認する。<br/> (3) 環境変数 COB_SWITCH_FILE に指定した SWITCH 状態保持ファイルの状態を確認する。</p> |
| COB140 | <p>内部でのコード変換に失敗しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] COBOL CODE CONVERSION<br/>ERROR(PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因]</p> <p>(1) コード変換不能文字が存在した。<br/> (2) コード変換のとき, 内部/製品エラーが発生した。</p> <p>[埋め込み情報]</p> <p>コード=cccccc</p> <p><b>備考</b></p> <p>cccccc: 次のコード値(16進)を設定する</p> <p>(1) ffff: コード変換不能文字が存在した<br/> (2) 内部管理用のコード値: コード変換のとき, 内部/製品エラーが発生した</p> <p>[処置]</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>[対策]</p> <p>(1) 不正な文字コード値を使用しない。<br/> (2) 内部/製品エラーの場合, 弊社担当者に問い合わせる。</p>                                                                                                                                                                              |
| COB160 | <p>不正日本語データを検出しました。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] JAPANESE DATA<br/>EXCEPTION(PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因]</p> <p>日本語項目に不正なデータを検出した。</p> <p>[処置]</p> <p>解除キーを押せばプログラムの処理を継続する。</p> <p>[対策]</p> <p>正しい日本語データを入力する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                      |

|        |                                                                                                                                                                                                                                                                                                                                                                       |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB201 | <p>サブルーチン 埋め込み情報 でエラーが発生しました。(WRONG CHARACTER) (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONVERSION ERROR (WRONG CHARACTER) (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] サブルーチンで変換するデータ項目に不正なデータを検出した。</p> <p>[埋め込み情報] サブルーチン名</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 不正な文字コード値を使用しない。</p>                                                |
| COB202 | <p>サブルーチン 埋め込み情報 でエラーが発生しました。(WRONG COMBINATION) (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONVERSION ERROR (WRONG COMBINATION) (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] サブルーチンで変換するデータ項目に不正な濁点, 半濁点が存在した。</p> <p>[埋め込み情報] サブルーチン名</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 不正な濁点, 半濁点を使用しない。</p>                                       |
| COB203 | <p>サブルーチン 埋め込み情報 でエラーが発生しました。(WRONG CHARACTER AND COMBINATION) (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] CONVERSION ERROR (WRONG CHARACTER AND COMBINATION) (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] サブルーチンで変換するデータ項目に不正なデータと濁点, 半濁点が存在した。</p> <p>[埋め込み情報] サブルーチン名</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 不正な文字コード値, 濁点, 半濁点を使用しない。</p> |

|        |                                                                                                                                                                                                                                                                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB204 | <p>サブルーチン 埋め込み情報1 でエラーが発生しました。(埋め込み情報2, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] CONVERSION ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] サブルーチンで内部エラーが発生した。</p> <p>[埋め込み情報1] サブルーチン名</p> <p>[埋め込み情報2] コード=cccccc</p> <p><b>備考</b></p> <p>cccccc: コード変換失敗原因を表すコード値(16 進)</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 弊社担当者に問い合わせる。</p> |
| COB209 | <p>数字と符号の重ね合わせ形式が異なります。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONFLICT BETWEEN UNPACKED DECIMAL SIGN(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] -CS オプション指定ありのプログラムと-CS オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -CS オプション指定の有無を同じにする。</p>                                                                |
| COB210 | <p>-Ci オプション指定の有無の異なるプログラムが混在しています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONFLICT BETWEEN OPTION(Ci) (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] -Ci オプション指定ありのプログラムと-Ci オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -Ci オプション指定の有無を同じにする。</p>                                                              |
| COB211 | <p>[Enterprise] -CU オプション指定の有無の異なるプログラムが混在しています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONFLICT BETWEEN OPTION(CU) (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] -CU オプション指定ありのプログラムと-CU オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -CU オプション指定の有無を同じにする。</p>                                                 |

|        |                                                                                                                                                                                                                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB213 | <p>-CN オプション指定の有無の異なるプログラムが混在しています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONFLICT BETWEEN OPTION(CN) (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] -CN オプション指定ありのプログラムと-CN オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -CN オプション指定の有無を同じにする。</p>              |
| COB214 | <p>-Cj オプション指定の有無の異なるプログラムが混在しています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONFLICT BETWEEN OPTION(Cj) (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] -Cj オプション指定ありのプログラムと-Cj オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -Cj オプション指定の有無を同じにする。</p>              |
| COB215 | <p>[Enterprise] -MT オプション指定の有無の異なるプログラムが混在しています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] CONFLICT BETWEEN OPTION(MT) (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] -MT オプション指定ありのプログラムと-MT オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -MT オプション指定の有無を同じにする。</p> |
| COB218 | <p>[Enterprise for Windows] [Standard] -CF オプション指定の有無の異なるプログラムが混在しています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] -CF オプション指定ありのプログラムと-CF オプション指定なしのプログラムが混在している。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] -CF オプション指定の有無を同じにする。</p>                                                                          |

|                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>COB256</b></p> <p>[原因]</p> <p>[埋め込み<br/>情報]</p> <p>[処置]</p> <p>[対策]</p> | <p>[Standard] COMMIT／ROLLBACK 異常終了 (埋め込み情報, プログラム名 = zzzzzz, 行番号 = nnnnnn)</p> <p>COMMIT 文または ROLLBACK 文において, 異常が発生した。</p> <p>NNNN: XX YY [mmmmmmmm] ファイル名</p> <p><b>備考</b></p> <p>      NNNN: 関連製品名</p> <p>          DBLNK: COBOL DB アクセス</p> <p>          IFAS : IFASPRO RDB</p> <p>      XX: COBOL 文識別子</p> <p>          CM: COMMIT 文</p> <p>          RB: ROLLBACK 文</p> <p>      YY: ファイル入出力状態</p> <p>      mmmmmmmm: 詳細ステータス (16 進)</p> <p>      ファイル名: 内部ファイル ID または実ファイル名</p> <p>                  省略される場合がある</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>埋め込み情報を参考にして, COBOL DB アクセスまたは IFASPRO RDB の説明書にて, ファイルの状態を確認する。</p> |
| <p><b>COB257</b></p> <p>[原因]</p> <p>[処置]</p> <p>[対策]</p>                      | <p>外部ファイル不整合 (プログラム名 = zzzzzz, 行番号 = nnnnnn)</p> <p>EXTERNAL ファイルにおいて, ファイル種別の不整合が発生した。</p> <p>たとえば, COBOL 独自と COBOL 独自以外のファイル種別が混在する場合である。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>ファイル種別を同じにする。</p>                                                                                                                                                                                                                                                                                                                                                                                                                              |



|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB261 | <p>[Standard] [Enterprise for Windows] ファイルマップ情報不正 (埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] (1) レジストリへのアクセス中に不正データが読み込まれた。<br/>(2) システムからエラーが返却された。</p> <p>[埋め込み情報] ccccc</p> <p><b>備考</b><br/>cccccc: エラーの原因を表すコード値(16 進)<br/>00000000: レジストリに存在するはずの値名が見つからない<br/>または値名に設定された種類/データの値が不正<br/>上記以外: Windows システムのエラーコード</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] COBOL ファイルマッパーを使わず, エディタ等で編集したファイルマップ情報を登録している場合, 登録したファイルマップ情報が誤っていないか確認する。<br/>上記に該当しない場合, いったん OS をシャットダウンして, 再起動後, ファイルマップの再設定を行う。</p> |
| COB262 | <p>[Standard] 内部ファイル ID 置換処理でエラーが発生しました (埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] 内部ファイル ID 置換処理でエラーが発生した。</p> <p>[埋め込み情報] mmmmmmm</p> <p><b>備考</b><br/>mmmmmmmm: 詳細エラーコード errno (16 進)</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 埋め込み情報の詳細エラーコード errno を参考にして, Visual C++の説明書にて, エラー原因を確認する。</p>                                                                                                                                                                                                              |
| COB401 | <p>SORT/MERGE の実行順序が誤っています。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] WRONG SORT VERB SEQUENCE (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] (1) 出力手続き中に MERGE 文, RELEASE 文, または SORT 文を実行した。<br/>(2) AT END 句の無条件文を実行後, その出力手続き中に RETURN 文を実行した。<br/>(3) 入力手続き中に MERGE 文, RETURN 文または SORT 文を実行した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 入力手続き中または出力手続き中の手続き文の実行順序を修正する。</p>                                                                                                                |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB402 | <p>SORT/MERGE で使用する作業ファイルが確保できません。(プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] SORT/MERGE WORK FILE TOO SMALL (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] SORT 文, MERGE 文で内部的に使用する作業ファイルの領域を確保できない。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] メモリを確保する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| COB409 | <p>関連製品でエラーを検出しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] OTHER PRODUCT ERROR (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] 関連製品でエラーが発生した。</p> <p>[埋め込み情報] 形式 1<br/>NNNN: XX YY [mmmmmmmm] ファイル名</p> <p><b>備考</b></p> <p>NNNN: 関連製品名(かっこが付く場合がある)</p> <p>DBLNK : COBOL DB アクセス</p> <p>IFAS : IFASPRO RDB/IFAS ファイル</p> <p>SORTKIT : SORTKIT</p> <p>NECCDCNV init: FontAvenue UniAssist コード変換</p> <p>XX: COBOL 文識別子</p> <p>OP: OPEN 文</p> <p>CL: CLOSE 文</p> <p>RE: READ 文</p> <p>WR: WRITE 文</p> <p>ST: START 文</p> <p>RW: REWRITE 文</p> <p>DE: DELETE 文</p> <p>SC: SCRATCH 文</p> <p>CM: COMMIT 文</p> <p>RB: ROLLBACK 文</p> <p>SL: SELECT 文</p> <p>SORTKIT または FontAvenue UniAssist コード変換の場合は省略</p> <p>YY: ファイル入出力状態。省略される場合がある</p> <p>mmmmmmmm: 詳細ステータス</p> |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <p>COBOL DB アクセスと IFASPRO RDB/IFAS ファイルは 16 進<br/>SORTKIT は 10 進, SORTKIT のエラーに 40000 を加算<br/>FontAvenue UniAssist コード変換は 10 進<br/>ファイル名: 内部ファイル ID または実ファイル名。省略される場合がある</p> <p>・形式 2<br/>(NNNNN):SGミス 99</p> <p><b>備考</b><br/>NNNNN: 関連製品名<br/>DBLNK: COBOL DB アクセス<br/>IFAS: IFASPRO RDB/IFAS ファイル</p> <p>・形式 3<br/>aaaa</p> <p><b>備考</b><br/>aaaa: 9001 または 9002<br/>9001: COBOL アプリケーション実行環境のジョブ管理で特殊レジスタ<br/>COMPLETION-CODE の値を取得し設定するときにエラーが発生した<br/>9002: COBOL アプリケーション実行環境のジョブ管理で特殊レジスタ<br/>COMPLETION-CODE の値を COBOL アプリケーション実行環境の<br/>ジョブ管理側に通知するときにエラーが発生した</p> <p>[処置]<br/>解除キーを押せばプログラムは異常終了する。</p> <p>[対策]<br/>埋め込み情報を参考にして、関連製品のマニュアルにて、エラー原因を確認する。<br/>形式 2 の場合は、COBOL に対応したバージョンの製品をインストールする。</p> |
| COB410 | <p>[Enterprise for Windows] [Standard]<br/>DLL ロードエラーが発生しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux]<br/>共有ライブラリロードエラーが発生しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] SHARED LIBRARY LOAD ERROR.<br/>(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因]<br/>[Enterprise for Windows] [Standard] DLL ファイルが見つからない。<br/>[Enterprise for Linux] 共有ライブラリが見つからない。</p> <p>[埋め込み<br/>情報]<br/>NECCDCNV_JIS2004ToJIPS.dll : FontAvenue UniAssist コード変換</p>                                                                                                                                                                                                                    |

|      |                                                                                                                          |
|------|--------------------------------------------------------------------------------------------------------------------------|
|      | NECCDCNV_JIS2004ToShiftJIS.dll: FontAvenue UniAssist コード変換                                                               |
|      | IFASLKCB64.DLL : IFASPRO RDB/IFAS ファイル                                                                                   |
|      | CBLDBDLL.DLL : COBOL DB アクセス                                                                                             |
|      | isamnt64.dll : ISAM                                                                                                      |
|      | SORTKIT.DLL : SORTKIT(-MT 指定なし)                                                                                          |
|      | [Enterprise for Windows]                                                                                                 |
|      | libraf.dll : refam                                                                                                       |
|      | libjcmfsret.dll : refam                                                                                                  |
|      | STKT_MT.DLL : SORTKIT(-MT 指定あり)                                                                                          |
|      | [Enterprise for Linux]                                                                                                   |
|      | libraf.so : refam                                                                                                        |
|      | libisam64.so : ISAM                                                                                                      |
|      | libnsort.so : SORTKIT                                                                                                    |
|      |                                                                                                                          |
| [処置] | 解除キーを押せばプログラムは異常終了する。                                                                                                    |
| [対策] | [Enterprise for Windows] [Standard]                                                                                      |
|      | DLL 名に対応する製品がインストールされているか, DLL が配置されたディレクトリが環境変数 PATH に登録されているか確認する。                                                     |
|      | [Standard]                                                                                                               |
|      | DLL 名が意図した製品の DLL でない場合, 以下のように設定を見直す。                                                                                   |
|      | ・IFASPRO RDB 利用時, COBOL ファイルマッパーを使用する。                                                                                   |
|      | ・COBOL 独自の順ファイル利用時, -Cf オプションまたは COBOL ファイルマッパーを使用する。                                                                    |
|      | ・COBOL 独自の相対ファイル利用時, -Cg オプションまたは COBOL ファイルマッパーを使用する。                                                                   |
|      | ・ISAM の索引ファイル利用時, -Ch オプションまたは COBOL ファイルマッパーを使用する。                                                                      |
|      | <b>備考</b> オプションについては「1.4.2 翻訳用コマンドプロンプトとその使用法」,<br>COBOL ファイルマッパーについては COBOL ユーザーズガイド<br>「第 3 章 COBOL ファイルマッパーの使い方」を参照する |
|      | [Enterprise for Linux]                                                                                                   |
|      | 共有ライブラリ名に対応する製品がインストールされているか, 共有ライブラリが配置されたディレクトリが共有ライブラリの検索パスに含まれているか確認する。                                              |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB411 | <p>[Enterprise for Windows] [Standard]<br/> DLL から API 関数アドレスが取得できません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux]<br/> 共有ライブラリから API 関数アドレスが取得できません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[ Enterprise for Linux ] SHARED LIBRARY ERROR(API NOT FOUND)<br/> (PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因]<br/> [Enterprise for Windows] [Standard] DLL から API 関数アドレスが取得できない。<br/> [Enterprise for Linux] 共有ライブラリから API 関数アドレスが取得できない。</p> <p>[埋め込み<br/>情報]<br/> [Standard]<br/> PRNAMR.DLL : COBOL アプリケーション実行環境</p> <p>[Enterprise for Windows]<br/> libraf.dll : refam<br/> libjcmfsret.dll : refam<br/> STKT_MT.DLL : SORTKIT(-MT 指定あり)</p> <p>[Enterprise for Windows] [Standard]<br/> isamnt64.dll : ISAM<br/> FontAvenue UniAssist コード変換: DLL から API 関数アドレスが取得できない場合,<br/> 関数名を埋め込む。</p> <p>SORTKIT.DLL : SORTKIT(-MT 指定なし)</p> <p>[Enterprise for Linux]<br/> libraf.so : refam<br/> libisam64.so : ISAM<br/> libnsort.so : SORTKIT</p> <p>[処置]<br/> 解除キーを押せばプログラムは異常終了する。</p> <p>[対策]<br/> [Enterprise for Windows] [Standard]<br/> 製品が COBOL に対応したバージョンであるか, DLL の検索パスに同名の DLL が存在しないか確認する。</p> <p>[Enterprise for Linux]<br/> 製品が COBOL に対応したバージョンであるか, 共有ライブラリの検索パスに同名の共有ライブラリが存在しないか確認する。</p> |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB412 | <p>[Standard] [Enterprise for Windows]</p> <p>環境変数が指定されていません。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] -CF オプションを指定した場合, 環境変数 FA_JIS2004_CDMAP が定義されていない。</p> <p>[埋め込み情報] FA_JIS2004_CDMAP</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 環境変数 FA_JIS2004_CDMAP を定義して, ユーザ定義マッピングファイルが存在するフォルダ (絶対パス) を指定する。</p>                                                                                                                                              |
| COB413 | <p>[Standard] [Enterprise for Windows]</p> <p>環境変数に指定した値が不正です。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] -CF オプションを指定した場合, 環境変数 FA_JIS2004_CDMAP に指定したフォルダにユーザ定義マッピングファイルがない。</p> <p>[埋め込み情報] ユーザ定義マッピングファイル名</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 環境変数 FA_JIS2004_CDMAP にユーザ定義マッピングファイルが存在するフォルダ (絶対パス) を指定する。</p>                                                                                                                                  |
| COB502 | <p>ファイル入出力エラーが発生しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[Enterprise for Linux] FILE I/O ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>[原因] (1) 大容量記憶ファイルまたはプリンタファイルの入出力文でエラーが発生した。<br/>(2) [Standard] RDB ファイルの入出力文でエラーが発生した。</p> <p>[埋め込み情報] ・形式 1<br/>30mmnn</p> <p><b>備考</b><br/>mm: プリンタファイルの詳細ステータス 1 (16 進)<br/>nn: プリンタファイルの詳細ステータス 2 (16 進)</p> <p>・形式 2<br/>NNNN: XX [01mmmmmmmm]</p> <p><b>備考</b><br/>NNNN: 関連製品名<br/>DBLNK: COBOL DB アクセス</p> |

**IFAS: IFASPRO RDB/IFAS ファイル**

**XX: COBOL 文識別子**

**OP: OPEN 文**

**CL: CLOSE 文**

**RE: READ 文**

**WR: WRITE 文**

**ST: START 文**

**RW: REWRITE 文**

**DE: DELETE 文**

**SC: SCRATCH 文**

**CM: COMMIT 文**

**RB: ROLLBACK 文**

**SL: SELECT 文**

**mmmmmm: 詳細ステータス (16 進)**

**・形式 3**

**補足文字 XX STATUS=YY(mmmm) ファイル名**

**備考**

**補足文字: エラーの補足説明**

「表 C-3 COB502 エラーの補足文字」を参照する

省略される場合がある

**XX: COBOL 文識別子**

**OP: OPEN 文**

**CL: CLOSE 文**

**RE: READ 文**

**WR: WRITE 文**

**ST: START 文**

**RW: REWRITE 文**

**DE: DELETE 文**

**SC: SCRATCH 文**

**CM: COMMIT 文**

**RB: ROLLBACK 文**

**SL: SELECT 文**

**YY: ファイル入出力状態**

**mmmm: 詳細ステータス (10 進)**

**ファイル名: 内部ファイル ID または実ファイル名**

|                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>[処置]<br/>[対策]</p>                                                | <p>解除キーを押せばプログラムは異常終了する。<br/>埋め込み情報を参考にして、COBOL または関連製品のマニュアルにて、ファイルの状態を確認する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <p>COB506</p> <p>[原因]</p> <p>[埋め込み<br/>情報]</p> <p>[処置]<br/>[対策]</p> | <p>標準入力終了した為、小入力あるいは確認入力できません。(埋め込み情報、プログラム名=zzzzz, 行番号=nnnnnn)<br/>[Enterprise for Linux] STDIN TERMINATED (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>(1) ファイルからデータを読み込むとき、入力データの読み込みに失敗した。<br/>(2) 読み込むデータがない。</p> <p>以下の実行時にファイルからデータを読み込む場合がある。</p> <ul style="list-style-type: none"> <li>・小入力 ACCEPT 文の実行時</li> <li>・STOP 文(定数指定)の実行時(一時停止の解除)</li> <li>・DISPLAY 文(UPON GDD に対応付けられた呼び名指定)の実行時(一時停止の解除)</li> <li>・実行時エラーメッセージ表示時(一時停止の解除)</li> </ul> <p>コード=cccccc</p> <p><b>備考</b></p> <p>cccccc: 内部管理用のコード値(16 進)</p> <p>解除キーを押せばプログラムは異常終了する。<br/>データが入力ファイルに存在することを確認する。</p> |
| <p>COB601</p> <p>[原因]</p> <p>[処置]<br/>[対策]</p>                      | <p>[Standard] 画面初期化エラー (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>対話型アプリケーションまたは COBOL GUI アプリケーションの初期化処理でエラーが発生した。</p> <p>解除キーを押せばプログラムは異常終了する。<br/>弊社担当者に問い合わせる。</p>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <p>COB602</p> <p>[原因]</p> <p>[処置]<br/>[対策]</p>                      | <p>[Standard] 画面入出力エラー (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>画面入出力中にエラーが発生した。</p> <p>解除キーを押せばプログラムは異常終了する。<br/>弊社担当者に問い合わせる。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <p>COB603</p> <p>[原因]</p> <p>[処置]<br/>[対策]</p>                      | <p>[Standard] 画面OPEN実行順序エラー (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>(1) 開かれている副画面に OPEN 文を実行した。<br/>(2) 開かれていない副画面に OPEN 文を実行したが、親画面が開かれていない。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>(1) 開かれていない副画面を指定する。<br/>(2) 開かれている親画面の副画面を指定する。</p>                                                                                                                                                                                                                                                                                                                                                        |



|        |                                                                                                                                                                                          |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB604 | [Standard] 画面CLOSE実行順序エラー (プログラム名=zzzzz, 行番号=nnnnn)                                                                                                                                      |
| [原因]   | 開かれていない副画面に CLOSE 文を実行した。                                                                                                                                                                |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]   | 開かれている副画面を指定する。                                                                                                                                                                          |
| COB605 | メモリ不足エラーが発生しました。(プログラム名=zzzzz, 行番号=nnnnn)                                                                                                                                                |
| [原因]   | メモリが確保できない。                                                                                                                                                                              |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]   | ・同時実行するプログラムの数を減らして, メモリ不足を解消する。<br>・メモリを増設して, 十分な利用可能メモリを確保する。                                                                                                                          |
| COB606 | [Standard] 画面DISPLAY実行順序不正 (プログラム名=zzzzz, 行番号=nnnnn)                                                                                                                                     |
| [原因]   | (1) OPEN していない副画面に DISPLAY 文を実行した。<br>(2) NO ACTUAL EXECUTING 句の指定のある DISPLAY 文を実行した後, ACCEPT 文または NO ACTUAL EXECUTING 句の指定のない DISPLAY 文を実行せずに, OPEN 文により副画面を開いた。または, CLOSE 文により副画面を閉じた。 |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]   | (1) 画面入出力文の実行順序を見直す。<br>(2) NO ACTUAL EXECUTING 句の指定のない DISPLAY 文に変更する。                                                                                                                  |
| COB607 | [Standard] ウィンドウ表示領域不正 (プログラム名=zzzzz, 行番号=nnnnn)                                                                                                                                         |
| [原因]   | (1) 副画面の大きさ, 表示位置の指定が範囲外である。<br>(2) 子画面の表示域が親画面の表示域を超えた。<br>(3) BORDER 句の指定のある子画面の親画面に BORDER 句の指定がない。                                                                                   |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]   | (1) 副画面の大きさ, 表示位置の指定を範囲内に収める。<br>(2) 子画面の表示域を親画面の表示域に収める。<br>(3) BORDER 句の指定のある子画面の親画面に BORDER 句を指定する。                                                                                   |
| COB608 | [Standard] 画面OPEN制限超過 (プログラム名=zzzzz, 行番号=nnnnn)                                                                                                                                          |
| [原因]   | 同時に開く副画面が最大数(16 個)を超えた。                                                                                                                                                                  |
| [処置]   | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]   | 同時に開く副画面の数を 16 個以下にする。                                                                                                                                                                   |

|                                    |                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB609<br>[原因]<br><br>[処置]<br>[対策] | <p>[Standard] 画面ACCEPT入力領域不正 (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>(1) 主画面で入力域の桁が 1 未満である。</p> <p>(2) 主画面で入力域が最終行を超えている。(COBOL WS エミュレータ利用時)</p> <p>(3) 副画面で入力域の行が 1~WINDOW HEIGHT に指定した行の範囲内でない。</p> <p>(4) 副画面で入力域の桁が 1~WINDOW WIDTH に指定した桁の範囲内でない。</p> <p>(5) 副画面に枠指定がない場合, 画面の左端または右端に BOX 指定の入力域がある。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>入力領域の指定を訂正する。</p>                       |
| COB610<br>[原因]<br><br>[処置]<br>[対策] | <p>[Standard] 画面DISPLAY出力領域不正 (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>(1) 主画面で表示域の桁が 1 未満である。</p> <p>(2) 主画面で表示域が最終行を超えている。(COBOL WS エミュレータ利用時)</p> <p>(3) 副画面で表示域の行が 1~WINDOW HEIGHT に指定した行の範囲内でない。</p> <p>(4) 副画面で表示域の桁が 1~WINDOW WIDTH に指定した桁の範囲内でない。</p> <p>(5) 副画面に枠指定がない場合, 画面の左端または右端に BOX 指定の表示域がある。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>出力領域の指定を訂正する。</p>                      |
| COB611<br>[原因]<br>[処置]<br>[対策]     | <p>[Standard] 画面制御機能は使用できません (プログラム名=zzzzz)</p> <p>画面制御機能を使用できないバッチ型アプリケーションで画面制御機能を実行した。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>画面制御機能を使用する場合は, -M オプション (バッチ型アプリケーションを作成する) を -Mw オプション (対話型アプリケーションを作成する) に変更してコンパイルしなおす。</p>                                                                                                                                                    |
| COB612<br>[原因]<br>[処置]<br>[対策]     | <p>[Standard] 画面サイズエラー</p> <p>ウィンドウが画面に収まらない。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>画面に収まるように, COBOL アプリケーションカスタマイザを使用して, [画面表示フォントのサイズを変更する], [ウィンドウサイズの既定値を変更する] から表示フォント/行数/桁数を変更する。</p>                                                                                                                                                                                        |
| COB613<br>[原因]<br><br>[処置]<br>[対策] | <p>[Standard] 画面ACCEPT実行順序エラー (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>(1) OPEN していない副画面に ACCEPT 文を実行した。</p> <p>(2) NO ACTUAL EXECUTING 句の指定のある DISPLAY 文を実行した後, ACCEPT 文または NO ACTUAL EXECUTING 句の指定のない DISPLAY 文を実行しないまま, OPEN 文により副画面を開いた。または, CLOSE 文により副画面を閉じた。</p> <p>解除キーを押せばプログラムは異常終了する。</p> <p>(1) 画面入出力文の実行順序を見直す。</p> <p>(2) NO ACTUAL EXECUTING 句の指定のない DISPLAY 文に変更する。</p> |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB650 | <p>オプション指定が不整合のため小入出力機能は使用できません (プログラム名=zzzzz, 行番号=nnnnn)</p> <p><b>[原因]</b></p> <p>(1) <b>[Standard]</b> -CF オプション指定時に COBOL WS エミュレータで画面制御機能, 小入出力文, STOP 文(定数指定)を使用している。</p> <p>(2) <b>[Enterprise]</b> -CU オプション指定時に小入出力文, STOP 文(定数指定)を使用している。</p> <p>(3) -Cj オプション指定時に小入出力文, STOP 文(定数指定)を使用している。</p> <p><b>[処置]</b><br/>解除キーを押せばプログラムは異常終了する。</p> <p><b>[対策]</b></p> <p>(1) プログラム固有文字集合を変更するオプションを使用しない。</p> <p>(2) 画面制御機能, 小入出力文, STOP 文(定数指定)をファイル入出力文に置き換える。</p> |
| COB680 | <p><b>[原因]</b><br/><b>[Standard]</b> <b>[Enterprise for Windows]</b> 表示フォント不正</p> <p>COBOL アプリケーションカスタマイザで-CF オプション指定時に使用できないフォントを指定している。</p> <p><b>[処置]</b><br/>メッセージ表示後, そのまま処理を継続する。</p> <p><b>[対策]</b><br/>COBOL アプリケーションカスタマイザのフォント設定を「FA JIPS 明朝」または「FA JIPS ゴシック」のいずれかにする。</p>                                                                                                                                                                                      |
| COB699 | <p><b>[原因]</b><br/><b>[Standard]</b> STNAMR. DLLロードエラー (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>STNAMR.DLL のロードに失敗した。</p> <p><b>[処置]</b><br/>解除キーを押せばプログラムは異常終了する。</p> <p><b>[対策]</b><br/>COBOL に対応したバージョンの COBOL アプリケーション実行環境がインストールされているか確認する。</p>                                                                                                                                                                                                                               |
| COB701 | <p>オブジェクトビューによる適合に失敗しました。(プログラム名=zzzzz, 行番号=nnnnn)</p> <p><b>[原因]</b></p> <p>[ Enterprise for Linux ] OBJECTVIEW CONFORMANCE FAILED(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)</p> <p>(1) クラス名 1 に FACTORY 指定がないとき, 一意名 1 の指すオブジェクトが, クラス名 1 に属するオブジェクトや, その下位クラスに属するオブジェクトではない。</p> <p>(2) FACTORY 指定があるとき, 一意名 1 の指すオブジェクトがクラス名 1 のファクトリオブジェクトやその下位クラスのファクトリオブジェクトではない。</p> <p><b>[処置]</b><br/>解除キーを押せばプログラムは異常終了する。</p> <p><b>[対策]</b><br/>プログラムを修正する。</p>                      |

|        |                                                                                                                                                                                                                                                                                                         |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB702 | <p>NULL オブジェクト参照を使用してメソッドが呼び出されました。(プログラム名=zzzzz, 行番号=nnnnn)</p> <p>[ Enterprise for Linux ] METHOD CALL BY NULL OBJECT REFERENCE (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] オブジェクト参照項目が NULL である。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] プログラムを修正する。</p>                                  |
| COB703 | <p>インスタンスの生成に失敗しました。(プログラム名=zzzzz, 行番号=nnnnn)</p> <p>[Enterprise for Linux] NEW METHOD FAILED (PROGRAM-ID=zzzzz, STATEMENT=nnnnnn)</p> <p>[原因] New メソッドによるオブジェクト生成がリソース不足により失敗した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] (1) リソースを増設する。<br/>(2) 同時に起動するアプリケーションの数を減らす。<br/>(3) 不要なサービスを停止する。</p> |
| COB800 | <p>[Standard] コンソールへの小入出力文は使用できません (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>[原因] COBOL GUI アプリケーションのとき、明示的または暗黙的に装置名 CONSOLE, STDIN に対応付けた呼び名を指定した ACCEPT 文, 装置名 CONSOLE, STDOUT, STDERR に対応付けた呼び名を指定した DISPLAY 文は使用できない。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] COBOL GUI アプリケーションのとき、小入出力文を使用しない。</p>  |
| COB801 | <p>[Standard] 画面操作のシステムサブルーチンは使用できません (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>[原因] COBOL GUI アプリケーションのとき、次の使用できないシステムサブルーチンを実行した。<br/>・@ATRS<br/>・@BUZS</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] COBOL GUI アプリケーションのとき、使用できないシステムサブルーチンを実行しない。</p>                                                        |
| COB802 | <p>[Standard] このアプリケーションを実行することはできません (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>[原因] COBOL GUI アプリケーションが COBOL WS エミュレータ配下で実行された。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] なし。</p>                                                                                                                          |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB805 | <p>[Standard] 外部モジュール呼び出しプログラムからは使用できません (プログラム名=zzzzz, 行番号=nnnnn)</p> <p>[原因] COBOL GUI 手続き機能の「外部モジュール呼び出し」で呼び出す COBOL プログラムで画面制御機能の手続き文を実行した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 外部モジュール呼び出しから呼び出す COBOL プログラムで画面制御機能の手続き文を実行しない。</p>                                                                                                                                                                                       |
| COB918 | <p>[Standard] 小入出力異常 (埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] COBOL WS エミュレータ配下でないバッチ型アプリケーションにおいて, 小入出力文でエラーが発生した。</p> <p>[埋め込み情報] 入出力エラーの発生した装置名 (STDIN/CONSOLE/STDOUT/STDERR)を設定する。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 弊社担当者に問い合わせる。</p>                                                                                                                                                                             |
| COB919 | <p>メモリ不足 (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] 小出力において必要なメモリの確保に失敗した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] メモリを増設する。</p>                                                                                                                                                                                                                                                                                                       |
| COB920 | <p>[Standard] コンソールへの小入出力機能は使用できません (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] 他言語がメインプログラムの COBOL WS エミュレータ配下でない対話型アプリケーションにおいて, 明示的または暗黙的に装置名 CONSOLE, STDIN に対応付けた呼び名を指定した ACCEPT 文, 装置名 CONSOLE, STDOUT, STDERR に対応付けた呼び名を指定した DISPLAY 文を実行した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 他言語がメインプログラムの COBOL WS エミュレータ配下でない対話型アプリケーションにおいて, 明示的または暗黙的に装置名 CONSOLE, STDIN, STDOUT, STDERR に対応付けた呼び名を ACCEPT 文または DISPLAY 文を実行しない。</p> |
| COB980 | <p>小入出力データ不正 (プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>[原因] COBOL WS エミュレータ配下でないバッチ型アプリケーション以外において, 明示的または暗黙的に装置名 CONSOLE, STDIN に対応付けた呼び名を指定した ACCEPT 文, 装置名 CONSOLE, STDOUT, STDERR に対応付けた呼び名を指定した DISPLAY 文でコード変換エラーが発生した。</p> <p>[処置] 解除キーを押せばプログラムの処理を継続する。</p> <p>[対策] 不正な文字コード値を訂正する。</p>                                                                                                                                                 |

|              |                                                                                                                                                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB982       | COBOLエラー (埋め込み情報) (プログラム名=zzzzz, 行番号=nnnnnn)                                                                                                                                                                             |
| [原因]         | 内部エラーが発生した。                                                                                                                                                                                                              |
| [埋め込み<br>情報] | cccccc<br><b>備考</b><br>cccccc: 内部管理用のコード値(16 進)                                                                                                                                                                          |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                    |
| [対策]         | 弊社担当者に問い合わせる。                                                                                                                                                                                                            |
| COB991       | [Enterprise for Windows] インストール環境が Enterprise Edition の標準動作モードではなくプログラムを実行できません。(プログラム名=zzzzz, 行番号=nnnnnn)                                                                                                               |
| [原因]         | インストール環境が Enterprise Edition の標準動作モードではないため, Enterprise Edition の標準動作モードの COBOL アプリケーションが実行できない。                                                                                                                         |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                    |
| [対策]         | アプリケーション動作モード切替ツールで Enterprise Edition の動作モードを Enterprise Edition の標準動作モードに切り替える。                                                                                                                                        |
| COB992       | [Enterprise for Windows] [Standard] インストール環境が不正である。(プログラム名=zzzzz, 行番号=nnnnnn)                                                                                                                                            |
| [原因]         | [Enterprise for Windows] インストール環境が Enterprise Edition の SE 動作モードではないため, Enterprise Edition の SE 動作モードの COBOL アプリケーションが実行できない。<br>[Standard] インストール環境が Standard Edition ではないため, Standard Edition の COBOL アプリケーションが実行できない。 |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                    |
| [対策]         | [Enterprise for Windows] アプリケーション動作モード切替ツールで Enterprise Edition の動作モードを Enterprise Edition の SE 動作モードに切り替える。<br>[Standard] Standard Edition を再インストールする。                                                                  |

|                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>COB996</p> <p>[原因]</p> <p>[埋め込み<br/>情報]</p> <p>[処置]</p> <p>[対策]</p> | <p>[Standard] 別プログラムの起動に失敗 (埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)</p> <p>別プログラムの起動に失敗した。</p> <p>XXXXXX:YY</p> <p><b>備考</b></p> <p>XXXXXX: エラーが発生したシステムサブルーチン名</p> <p>CBLRUN</p> <p>CBLCHAINR</p> <p>YY: 実行結果(10 進)</p> <p>23: 次のいずれかである</p> <ul style="list-style-type: none"> <li>・指定のプログラムが指定のパスが示すフォルダに存在しない</li> <li>・指定のパスが示すフォルダが存在しない</li> <li>・COBOL アプリケーションの実行ユーザにプログラムやパスが示すフォルダへのアクセス許可がない</li> </ul> <p>90: データ名 1 の記述内容誤り</p> <p>91: 次のいずれかである</p> <ul style="list-style-type: none"> <li>・(COBOL WS エミュレータ配下の COBOL アプリケーションの実行を除き) CBLRUN を利用する COBOL アプリケーションと同一ユーザ/同一セッションで, COBOL アプリケーションの起動前からジョブ管理が動作していなかった</li> <li>・ジョブ管理の動作に必要なメモリ等の資源が不足のため確保できなかった</li> <li>・ジョブ管理の内部異常</li> </ul> <p>解除キーを押せばプログラムは異常終了する。</p> <p>・実行結果=23(CBLRUN のみ)</p> <p>データ名 1 に指定したプログラムやパスが示すフォルダが存在するか, COBOL アプリケーションの実行ユーザに起動プログラムや格納フォルダへのアクセス許可を付与しているか確認する。</p> <p>・実行結果=90(CBLRUN のみ)</p> <p>データ名 1 の記述内容に誤り(たとえば"の数合わないなど)がないか確認する。</p> <p>・実行結果=91(CBLRUN/CBLCHAINR 共通)</p> <p>COBOL アプリケーションと同一ユーザ/同一セッションで, COBOL アプリケーション起動前に COBOL アプリケーションのジョブ管理が動作していたか確認する。</p> <p>COBOL アプリケーション起動前に COBOL アプリケーションのジョブ管理が動作していて, このエラーが発生した場合は, いったんログオフして再ログオン後に COBOL アプリケーションのジョブ管理を実行状態にして COBOL アプリケーションを再実行する。</p> <p>COBOL WS エミュレータ配下で実行中の COBOL アプリケーションで, このエラーになった場</p> |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|              |                                                                                                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 合は, いったん COBOL WS エミュレータを操作終了して再接続後に COBOL アプリケーションを再実行する。                                                                                                                                                   |
| COB997       | データ値出力の指定が正しくありません。                                                                                                                                                                                          |
| [原因]         | COBOL 実行トレース情報ファイルのオープン, 書き込み, またはクローズに失敗した。                                                                                                                                                                 |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                        |
| [対策]         | データ値出力の指定を訂正する。                                                                                                                                                                                              |
| COB998       | トレース情報ファイルのアクセスエラーが発生しました。(埋め込み情報, プログラム名=zzzzz, 行番号=nnnnnn)                                                                                                                                                 |
|              | [ Enterprise for Linux ] TRACE FILE I/O<br>ERROR(PROGRAM-ID=zzzzz,STATEMENT=nnnnnn)                                                                                                                          |
| [原因]         | COBOL 実行トレース情報ファイルのオープン, 書き込み, またはクローズに失敗した。                                                                                                                                                                 |
| [埋め込み<br>情報] | XX[mmmmm]<br><b>備考</b><br>XX: 動作識別子<br>OP: オープン<br>WR: 書き込み<br>CL: クローズ<br>mmmmm: [Enterprise for Windows] [Standard] 詳細エラーコード(10 進, GetLastError の返却値)<br>[Enterprise for Linux] 詳細エラーコード(10 進, errno の返却値) |
| [処置]         | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                        |
| [対策]         | [Enterprise for Windows] [Standard] 埋め込み情報を参考にして, Visual C++の説明書にて, エラー原因を確認する。<br>[Enterprise for Linux] 埋め込み情報を参考にして, エラー原因を確認する。                                                                          |
| COB999       | [Enterprise for Windows] メモリ不足でエラーが発生しました。(COBOL ランタイムの初期化に失敗しました。)                                                                                                                                          |
|              | [ Enterprise for Linux ] MEMORY ALLOCATION ERROR DURING COBOL RUNTIME INITIATIZATION                                                                                                                         |
| [原因]         | マルチスレッドプログラム(-MT オプション指定あり)において, 作業領域のメモリが確保できない。                                                                                                                                                            |
| [処置]         | COBOL 実行時エラーを表示した後, 例外が発生する。                                                                                                                                                                                 |
| [対策]         | メモリを増設する。                                                                                                                                                                                                    |

**注意**

- (注 1) メッセージの(プログラム名=xxxx, 行番号=xxx)は, 「クラス」や「利用者定義関数」の場合も「プログラム名=xxxx」に「クラス名」や「利用者定義関数名」を出力する。





[Standard] [Enterprise for Windows] 次に COBOL WS エミュレータに出力する実行時エラーを一覧で示す。

表 C-2 COBOL WS エミュレータに出力する実行時エラー

| 番号  | エラーメッセージの説明                           |
|-----|---------------------------------------|
| 000 | [zzzzz/nnnnnn] 内部エラー                  |
| 002 | [zzzzz/nnnnnn] データの限界超過               |
| 003 | [zzzzz/nnnnnn] 繰り返し数の限界超過             |
| 004 | [zzzzz/nnnnnn] GO TO の飛び先未設定          |
| 006 | [zzzzz/nnnnnn] INSPECT の長さ不一致         |
| 007 | [zzzzz/nnnnnn] 環境変数 COB_LIBRARY 不正    |
| 008 | [zzzzz/nnnnnn] CALL エントリ未検出           |
| 010 | [zzzzz/nnnnnn] 不正 10 進データ検出           |
| 048 | [zzzzz/nnnnnn] 利用者制御変数エラー (埋め込み情報)    |
| 049 | [zzzzz/nnnnnn] 利用者制御変数が存在しない (埋め込み情報) |
| 093 | [zzzzz/nnnnnn] 関数の引数がない (埋め込み情報)      |
| 094 | [zzzzz/nnnnnn] 不正 10 進データ検出 (埋め込み情報)  |
| 095 | [zzzzz/nnnnnn] 関数の引数値が範囲外 (埋め込み情報)    |
| 100 | [zzzzz/nnnnnn] ファイル未指定 (埋め込み情報)       |
| 101 | [zzzzz/nnnnnn] ファイル OPEN エラー (埋め込み情報) |
| 102 | [zzzzz/nnnnnn] ファイル未 OPEN (埋め込み情報)    |
| 103 | [zzzzz/nnnnnn] ファイル入出力エラー (埋め込み情報)    |
| 104 | [zzzzz/nnnnnn] ファイル OVERFLOW (埋め込み情報) |
| 105 | [zzzzz/nnnnnn] CSV ファイル読込エラー          |
| 106 | [zzzzz/nnnnnn] ファイルパス長の限界超過           |
| 107 | [zzzzz/nnnnnn] オプション不整合 (埋め込み情報)      |
| 135 | [zzzzz/nnnnnn] 外部スイッチ参照エラー            |
| 136 | [zzzzz/nnnnnn] 外部スイッチ設定エラー            |
| 140 | [zzzzz/nnnnnn] コード変換失敗 (埋め込み情報)       |
| 160 | [zzzzz/nnnnnn] 不正日本語データ検出             |
| 201 | [zzzzz/nnnnnn] コード変換エラー (埋め込み情報)      |
| 202 | [zzzzz/nnnnnn] コード変換エラー (埋め込み情報)      |
| 203 | [zzzzz/nnnnnn] コード変換エラー (埋め込み情報)      |
| 204 | [zzzzz/nnnnnn] コード変換エラー (埋め込み情報)      |

|     |                                            |
|-----|--------------------------------------------|
| 209 | [zzzzz/nnnnnn] 重ね符号形式不正                    |
| 210 | [zzzzz/nnnnnn] -Ci オプション不整合                |
| 213 | [zzzzz/nnnnnn] -CN オプション不整合                |
| 214 | [zzzzz/nnnnnn] -Cj オプション不整合                |
| 218 | [zzzzz/nnnnnn] -CF オプション不整合                |
| 256 | [zzzzz/nnnnnn] COMMIT/ROLLBACK 異常 (埋め込み情報) |
| 257 | [zzzzz/nnnnnn] 外部ファイル不整合                   |
| 261 | [zzzzz/nnnnnn] ファイルマップ情報不正 (埋め込み情報)        |
| 262 | [zzzzz/nnnnnn] 内部ファイル ID 置換表エラー (埋め込み情報)   |
| 401 | [zzzzz/nnnnnn] 整列併合実行順序不正                  |
| 402 | [zzzzz/nnnnnn] 整列併合作業ファイル確保失敗              |
| 409 | [zzzzz/nnnnnn] 関連製品エラー (埋め込み情報)            |
| 410 | [zzzzz/nnnnnn] DLL ロードエラー (埋め込み情報)         |
| 411 | [zzzzz/nnnnnn] 関数アドレス取得失敗 (埋め込み情報)         |
| 412 | [zzzzz/nnnnnn] 環境変数取得失敗 (埋め込み情報)           |
| 413 | [zzzzz/nnnnnn] 環境変数不正 (埋め込み情報)             |
| 502 | [zzzzz/nnnnnn] ファイル入出力エラー (埋め込み情報)         |
| 506 | [zzzzz/nnnnnn] 標準入力終了 (埋め込み情報)             |
| 601 | [zzzzz/nnnnnn] 画面初期化エラー                    |
| 602 | [zzzzz/nnnnnn] 画面入出力エラー                    |
| 603 | [zzzzz/nnnnnn] 画面 OPEN 実行順序エラー             |
| 604 | [zzzzz/nnnnnn] 画面 CLOSE 実行順序エラー            |
| 605 | [zzzzz/nnnnnn] メモリ不足エラー                    |
| 606 | [zzzzz/nnnnnn] 画面 DISPLAY 実行順序不正           |
| 607 | [zzzzz/nnnnnn] ウィンドウ表示領域不正                 |
| 608 | [zzzzz/nnnnnn] 画面 OPEN 制限超過                |
| 609 | [zzzzz/nnnnnn] 画面 ACCEPT 入力領域不正            |
| 610 | [zzzzz/nnnnnn] 画面 DISPLAY 出力領域不正           |
| 611 | [zzzzz] 画面制御機能は使用できません                     |
| 612 | 画面サイズエラー                                   |
| 613 | [zzzzz/nnnnnn] 画面 ACCEPT 実行順序エラー           |
| 650 | [zzzzz/nnnnnn] オプション不整合                    |
| 680 | 表示フォント不正                                   |
| 699 | [zzzzz/nnnnnn] STNAMR.DLL ロードエラー           |
| 701 | [zzzzz/nnnnnn] オブジェクトビュー適合エラー              |

|     |                                     |
|-----|-------------------------------------|
| 702 | [zzzzz/nnnnnn] NULL オブジェクト参照        |
| 703 | [zzzzz/nnnnnn] インスタンス生成失敗           |
| 802 | [zzzzz/nnnnnn] GUI実行不可              |
| 918 | [zzzzz/nnnnnn] 小入出力異常 (埋め込み情報)      |
| 919 | [zzzzz/nnnnnn] メモリ不足                |
| 920 | [zzzzz/nnnnnn] 小入出力使用不可             |
| 980 | [zzzzz/nnnnnn] 小入出力データ不正            |
| 982 | [zzzzz/nnnnnn] COBOL エラー (埋め込み情報)   |
| 992 | [zzzzz/nnnnnn] インストール環境不正           |
| 996 | [zzzzz/nnnnnn] プログラム起動失敗 (埋め込み情報)   |
| 997 | データ項目指定誤り                           |
| 998 | [zzzzz/nnnnnn] トレースアクセスエラー (埋め込み情報) |
| 999 | メモリ不足エラー(初期化失敗)                     |

**注意**

(注 1) zzzzz(プログラム名), nnnnnn(行番号)以外の埋め込み情報の詳細は表 C-1 を参照する。

表 C-3 COB502 エラーの補足文字

| 補足文字の詳細内容   |                                                                                                                                                                                                                                      |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 設定情報不一致     |                                                                                                                                                                                                                                      |
| [原因]        | <p>入出力状態=39。</p> <p>ファイル固有属性と、そのファイルに対してソース単位が指定した属性との間に不整合が検出されたため、OPEN 文が不成功に終わった。</p> <p>次の場合である。</p> <p>指定されたファイル種別がそのファイル編成では利用できない場合</p>                                                                                       |
| [処置]        | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                |
| [対策]        | ファイルマップ機能で指定したファイル種別を見直す。                                                                                                                                                                                                            |
| キー不正        |                                                                                                                                                                                                                                      |
| [原因]        | <p>入出力状態=92。</p> <p>[Standard] IFAS FILE および IFASPRO RDB 使用時。</p> <p>(1) READ 文または START 文において、指定キーが存在しない。または、キー情報が不正である。</p> <p>(2) OUTPUT/EXTEND モードで OPEN されたファイルに対して、SELECT 文を実行した。</p> <p>(3) SELECT 文を行わずに SCRATCH 文を実行した。</p> |
| [処置]        | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                |
| [対策]        | <p>IFAS FILE, IFASPRO RDB のマニュアルを参照する。</p> <p>(1) READ または START 文において、存在するキーを正しく指定する。</p> <p>(2) OUTPUT/EXTEND モードで OPEN したファイルに対して、SELECT 文を実行しない。</p> <p>(3) SELECT 文を行わずに SCRATCH 文を実行しない。</p>                                   |
| デッドロック発生    |                                                                                                                                                                                                                                      |
| [原因]        | <p>入出力状態=9J。</p> <p>[Standard] デッドロック状態となった。ROLLBACK 文が自動的に行われた。</p>                                                                                                                                                                 |
| [処置]        | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                |
| [対策]        | IFASPRO RDB, COBOL DB アクセスのマニュアルを参照する。                                                                                                                                                                                               |
| ログ領域オーバーフロー |                                                                                                                                                                                                                                      |
| [原因]        | <p>・入出力状態=9N。</p> <p>[Standard] ログ情報がオーバーフローした(COBOL DB アクセス使用時)。</p> <p>・入出力状態=90。</p> <p>[Standard] ログ領域がオーバーフローした(IFAS FILE および IFASPRO RDB 使用時)。</p>                                                                             |
| [処置]        | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                |
| [対策]        | <p>入出力状態=9N のとき、COBOL DB アクセス のマニュアルを参照する。</p> <p>入出力状態=90 のとき、IFAS FILE, IFASPRO RDB のマニュアルを参照する。</p>                                                                                                                               |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ロックテーブルオーバーフロー |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [原因]           | 入出力状態=9P。<br>[Standard] ページまたはレコードのロックテーブルがオーバーフローした。                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [処置]           | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [対策]           | IFASPRO RDB のマニュアルを参照する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| DB破壊           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [原因]           | 入出力状態=9M。<br>[Standard] システム表のアクセスエラー(DB 破壊)がある(IFASPRO RDB 使用時)。                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| [処置]           | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [対策]           | IFASPRO RDB のマニュアルを参照する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| ロールバックログ収集エラー  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [原因]           | 入出力状態=9K。<br>(1) [Standard] ロールバックログ収集エラーが発生した(IFAS FILE および IFASPRO RDB 使用時)。<br>(2) [Standard] COMMIT/ROLLBACK が異常終了した(COBOL DB アクセス使用時)。                                                                                                                                                                                                                                                                                                                                                                                         |
| [処置]           | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [対策]           | IFAS FILE, IFASPRO RDB または COBOL DB アクセスのマニュアルを参照する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 不正状態           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [原因]           | ・入出力状態=21。<br>順呼び出しにおいて、順序誤りがあった。すなわち、成功した READ 文と次の REWRITE 文の間で主レコードキーの値が変更されたか、または WRITE 文の実行時に主レコードキーの値が昇順になっていない。<br>・入出力状態=22。<br>(1) WRITE 文または REWRITE 文で索引ファイルに書き出すレコードの主レコードキーまたは DUPLICATES 指定なしの副レコードキーの値が、すでにファイル中に存在している。<br>(2) 相対ファイルで重複キーがあった。すでにレコードがある位置に重ねて WRITE 文でレコードを書こうとした。<br>・入出力状態=23。<br>次の場合のいずれかである。<br>(1) ファイル中に存在しないレコードを乱呼び出しした。<br>(2) 存在しない不定入力ファイルに対して、START 文または乱呼び出しの READ 文を実行した。<br>・入出力状態=24。<br>(1) COBOL の外部で定義した索引ファイルの区域外にレコードを書き出した。<br>(2) 順呼び出しの WRITE 文が行われたが、相対レコード番号の有効桁数とそのファ |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [処置]  | イルに指定された相対キー項目の大きさより大きい。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [対策]  | <p>解除キーを押せばプログラムは異常終了する。</p> <p>・入出力状態＝21。</p> <p>順呼び出しにおいて、順序誤りがないことを確認する。すなわち、成功した READ 文と次の REWRITE 文の間で主レコードキーの値が変更されていないこと、かつ、WRITE 文の実行時に主レコード キーの値が昇順になっていることを確認する。</p> <p>・入出力状態＝22。</p> <p>(1) WRITE 文または REWRITE 文で索引ファイルに書き出すレコードの主レコードキーまたは DUPLICATES 指定なし副レコードキーにファイル中に存在しないキーを指定する。</p> <p>(2) 相対ファイルで重複キーを指定しないように処理を見直す。すでにレコードがある位置に重ねて WRITE 文でレコードを書き出さないように処理を見直す。</p> <p>・入出力状態＝23。</p> <p>次の場合のいずれかである。</p> <p>(1) ファイル中に存在しないレコードを呼び出さない。</p> <p>(2) 存在しない不定入力ファイルに対して、START 文または乱呼び出しの READ 文を実行しない。</p> <p>・入出力状態＝24。</p> <p>(1) COBOL の外部で定義した索引ファイルの区域外にレコードを書き出さない。</p> <p>(2) 順呼び出しの WRITE 文が行われたが、相対レコード番号の有効桁数とそのファイルに指定された相対キー項目の大きさより大きくならないように処理を見直す。</p> |
| 入出力異常 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| [原因]  | <p>入出力状態＝30。</p> <p>永続誤りが存在する。原因は詳細エラーコードにて確認が必要。詳細エラーコードの意味については、「13.2.4 エラーコード取得」を参照する。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| [処置]  | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [対策]  | 「13.2.4 エラーコード取得」のエラーコードの意味を元に、原因を特定し取り除く。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 容量不足  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| [原因]  | <p>入出力状態＝34。</p> <p>ファイルオーバーフロー。</p> <p>ファイル領域がデータで満たされている状態にもかかわらず、WRITE 文によってデータを追加した場合(順呼び出しのファイルのみ)。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| [処置]  | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [対策]  | ファイルの容量を確保する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ファイル未発見 | <p>[原因] 入出力状態=35。<br/>不定ファイルでないファイルが存在しない(表が定義されていない)とき、OPEN 文を実行したことによる永続誤りが存在する。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 存在するファイルを指定する。</p>                                                                                                                                                                                                                                                                                                                                       |
| 二重オープン  | <p>[原因] 入出力状態=41。<br/>開かれているファイルに OPEN 文を実行した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 開かれていないファイルを指定する。</p>                                                                                                                                                                                                                                                                                                                                                                        |
| 未オープン   | <p>[原因] ・入出力状態=42。<br/>開かれていないファイルに CLOSE 文を実行した。<br/>・入出力状態=48。<br/>開かれていない状態のプリンタファイルに WRITE 文を実行しようとした。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。<br/>・入出力状態=42。<br/>CLOSE 文を実行する前に OPEN 文を実行する。<br/>・入出力状態=48。<br/>WRITE 文を実行する前に OPEN 文を実行する。</p> <p>[対策] 開かれているファイルを指定する。</p>                                                                                                                                                                                                               |
| 文エラー    | <p>[原因] ・入出力状態=43。<br/>次の場合のいずれかである。</p> <p>(1) 順呼び出し法の大容量記憶ファイルに対して、あるファイル結合子を介して REWRITE 文や DELETE 文を実行する前に、同じファイル結合子を介して関連ファイルに対して実行された最後の入出力文が、成功完了した READ 文ではない。</p> <p>(2) RDB 順ファイルに対して、DELETE 文または REWRITE 文を実行する前に、関連するファイルに対して実行された最後の入出力文が、成功した READ 文でない。または WITH UNLOCK 句を指定した READ 文である。</p> <p>(3) RDB 相対ファイルまたは RDB 索引ファイルに対して、順呼び出しで、CURRENT 句付きの DELETE 文または CURRENT 句付きの REWRITE 文を実行する前に、そのファイルに対して実行された最後の入出力文が、成功した READ 文でない。または WITH UNLOCK 句を指定した READ 文である。</p> |



・入出力状態=46。

入力モードまたは入出力両用モードで開かれたファイルに、READ 文を実行したが、有効な次のレコードがない。次の場合のいずれかである。

- (1) 先行する READ 文が、ファイル終了条件以外により不成功であった。
- (2) 先行する READ 文が、ファイル終了条件を引き起こした。
- (3) 先行するその他の文により、ファイル位置指示子が不定となった。
- (4) 先行する START 文が不成功だった。

・入出力状態=47。

入力モードまたは入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、READ 文、SELECT 文または START 文を実行した。

・入出力状態=48。

次のように、正しい OPEN モードで開いていないファイル結合子に対して、WRITE 文を実行した。または結合型仮想表に対して、WRITE 文を実行した。

- (1) 順呼び出し法の場合、ファイル結合子が拡張モードや出力モードで開かれていない(開かれた状態でない場合も含む)。
- (2) 動的呼び出し法や乱呼び出し法の場合、ファイル結合子が入出力両用モードや出力モードで開かれていない(開かれた状態でない場合も含む)。

・入出力状態=49。

入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、DELETE 文または REWRITE 文を実行した。または、結合型仮想表に対して、DELETE 文または REWRITE 文を実行した。

・入出力状態=37。

RDB 順ファイル/RDB 相対ファイル/RDB 索引ファイルの場合、

オープンモード指定誤りが存在する。仮想表に対して、OUTPUT/EXTEND オープンを実行したことによる永続誤りが存在する。

[処置] 解除キーを押せばプログラムは異常終了する。

[対策] ・入出力状態=43。

- (1) 順呼び出し法の大容量記憶ファイルに対して、あるファイル結合子を介して REWRITE 文や DELETE 文を実行する前に、同じファイル結合子を介して関連ファイルに対して実行された最後の入出力文が、成功完了した READ 文になるように処理を見直す。
- (2) RDB 順ファイルに対して、DELETE 文または REWRITE 文を実行する前に、関連するファイルに対して実行された最後の入出力文が、成功した READ 文になるように処理を見直す。または WITH UNLOCK 句を指定した READ 文にならないように処理を見直す。
- (3) RDB 相対ファイルまたは RDB 索引ファイルに対して、順呼び出しで、CURRENT 句付

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>きの DELETE 文または CURRENT 句付きの REWRITE 文を実行する前に、そのファイルに対して実行された最後の入出力文が、成功した READ 文になるように処理を見直す。または WITH UNLOCK 句を指定した READ 文にならないように処理を見直す。</p> <p>・入出力状態=46。<br/>入力モードまたは入出力両用モードで開かれたファイルに READ 文を実行するとき、有効なレコードが存在するよう処理を見直す。</p> <p>・入出力状態=47。<br/>入力モードまたは入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、READ 文、SELECT 文または START 文を実行しない。</p> <p>・入出力状態=48。<br/>(1) 順呼び出し法の場合、ファイル結合子が拡張モードや出力モードで開かれていない(開かれた状態でない場合も含む)ファイルに、WRITE 文を実行しない。<br/>(2) 動的呼び出し法や乱呼び出し法の場合、ファイル結合子が入出力両用モードや出力モードで開かれていない(開かれた状態でない場合も含む)ファイルに、WRITE 文を実行しない。</p> <p>・入出力状態=49。<br/>入出力両用モードで開かれていない(開かれた状態でない場合も含む)ファイルに、DELETE 文または REWRITE 文を実行しない。または、結合型仮想表に対して、DELETE 文または REWRITE 文を実行しない。</p> <p>・入出力状態=37。<br/>オープンモード指定を修正する。または、仮想表に対して、OUTPUT/EXTEND オープンを実行しない。</p> | <p>ファイル使用中</p> <p>[原因] 入出力状態=91。<br/>[Standard] ファイルが他から排他的に使用中である。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] ファイル種別によっては、本エラーを出力せず、OPEN 可能となるまで待ち合わせる動作に変更することができる。詳細は「付録 D 環境変数」の「COB_FILE_SHARING」を参照する。</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 媒体異常

**[原因]** 入出力状態=30。

永続誤りが存在する。次のいずれかである。

(1) IFAS FILE および IFASPRO RDB 使用時, ディスク媒体が準備されていない。

(2) 他 の 場 合 , 詳 細 エ ラ ー ス テ ー タ ス に シ ス テ ム エ ラ ー コ ー ド 00021 (ERROR\_NOT\_READY)が返却された。

**[処置]** 解除キーを押せばプログラムは異常終了する。

**[対策]** 使用しているディスクが存在するかどうかをチェックする。

### アクセス不可

**[原因]** 入出力状態=37。

**[Enterprise for Windows] [Standard]** 永続誤りが存在する。あるファイルに OPEN 文を実行しようとしたが, OPEN 文に指定した OPEN モードはそのファイルには使えない。次のいずれかである。

(1) EXTEND 指定や OUTPUT 指定を書いたが, そのファイルでは書き込み操作ができない。

(2) I-O 指定を書いたが, 入出力両用モードで開いたときにそのファイル編成で行ってもよい入出力操作が, そのファイルでは使えない。

(3) INPUT 指定を書いたが, そのファイルでは読み取り操作ができない。

**[処置]** 解除キーを押せばプログラムは異常終了する。

**[対策]** ファイルのアクセス権限を見直す。

### オープン過剰

**[原因]** 入出力状態=30。

永続誤りが存在する。次の場合である。

・ 詳 細 エ ラ ー ス テ ー タ ス に シ ス テ ム エ ラ ー コ ー ド 00004 (ERROR\_TOO\_MANY\_OPEN\_FILES) が返却された。

**[処置]** 解除キーを押せばプログラムは異常終了する。

**[対策]** 同時オープンのファイル数を見直す。

|                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>レコード長矛盾</p> <p>[原因] 入出力状態=44。<br/>次のいずれかである。</p> <p>(1) RECORD IS VARYING 句で指定した最大レコードより大きい、または最小レコードより小さいレコード長で書き出したまたは書き換えを行った。</p> <p>(2) 順ファイルを書き換えるレコード長が、書き換えられるレコード長と同じでない。</p> <p>(3) 書き方 3(固定長または可変長)の RECORD 句に対して、本システムの規定によりレコード形式が可変長である場合に、固定長または可変長形式の RECORD 句で指定した最大レコードより大きい、または最小レコードより小さいレコード長で書き出したまたは書き換えを行った。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] 書き出したまたは書き換えのレコード長を見直す。</p>                                         | <p>メモリエラー</p> <p>[原因] 入出力状態=30。<br/>メモリの確保に失敗した。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] (1) リソースを増設する。<br/>(2) 同時に起動するアプリケーションの数を減らす。<br/>(3) 不要なサービスを停止する。</p> |
| <p>属性不一致</p> <p>[原因] 入出力状態=39。<br/>ファイル固有属性とソース単位で指定したファイル属性が不整合であるため、OPEN 文が不成功に終わった。<br/>次のいずれかである。</p> <p>(1) 関連製品のファイル固有属性(物理ファイルの属性)とソース単位に指定した属性との間で不整合が検出された。</p> <p>(2) EXTERNAL ファイル同士(ソース単位に指定した属性と別のソース単位に指定した属性)で、ファイル編成/アクセスモード/最大レコード長が一致しない。</p> <p>[処置] 解除キーを押せばプログラムは異常終了する。</p> <p>[対策] (1) IFAS FILE, IFASPRO RDB または COBOL DB アクセスのマニュアルを参照する。<br/>(2) EXTERNAL ファイル同士(ソース単位に指定した属性と別のソース単位に指定した属性)で、ファイル編成/アクセスモード/最大レコード長を統一する。</p> |                                                                                                                                                                        |

# ファイル名不正

[原因] ・入出力状態=30。

ファイル名不正。ファイルオープン処理中にエラーが発生した。

次のいずれかである。

- (1) 指定されたファイルが見つからない。
- (2) ファイル名または拡張子が長すぎる。
- (3) ファイル名, ディレクトリ名, またはボリュームラベルの構文が正しくない。

・入出力状態=99。

次のいずれかである。

- (1) [Standard] プリンタファイルに対するシステムエラー。ファイル名が不正である。
  - ①APPLY SHIFT-CODE が指定されている場合, 生成するファイル名がすべて空白である。
  - ②LABEL RECORD STANDARD が指定されている場合, 生成するスプールファイル名の長さが 257 より大きい, あるいは 0 である。
- (2) 本システムでは, 従来製品互換の翻訳オプションを指定することにより, 相対ファイル(COBOL 独自形式)および索引ファイル(ISAM 形式)の OPEN 文でファイル名が不正な場合, この入出力状態が発生する。翻訳オプション指定なしのときは, 入出力状態=30 である。
 

次のいずれかである。

  - ①末尾の連続する空白を除いた後のファイル名の長さが 246 より大きい。
  - ②指定されたファイルが見つからない。
  - ③ファイル名, ディレクトリ名, またはボリュームラベルの構文が正しくない。
- (3) 順ファイル, 行順ファイル, 行順ファイル(CSV 形式)でファイル名が不正な場合, この入出力状態が発生する。
 

次のいずれかである。

  - ①末尾の連続する空白を除いた後のファイル名の長さが 246 より大きい。
  - ②指定されたファイルが見つからない。
  - ③ファイル名, ディレクトリ名, またはボリュームラベルの構文が正しくない。
- (4) [Standard] 順ファイル(IFAS FILE 形式), 相対ファイル(IFAS FILE 形式), 索引ファイル(IFAS FILE 形式/COBOL DB アクセス形式)でファイル名が不正な場合, この入出力状態が発生する。
  - ①末尾の連続する空白を除いた後のファイル名の長さが 246 より大きい。

[処置] 解除キーを押せばプログラムは異常終了する。

[対策] ファイル名を修正する。

|                |                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------|
| ログファイル I/O エラー |                                                                                                                |
| 〔原因〕           | 入出力状態=9T。<br>〔Standard〕ログファイルの I/O エラーがある (IFAS FILE および IFASPRO RDB 使用時)。                                     |
| 〔処置〕           | 解除キーを押せばプログラムは異常終了する。                                                                                          |
| 〔対策〕           | IFAS FILE, IFASPRO RDB のマニュアルを参照する。                                                                            |
| フェイルオーバー発生     |                                                                                                                |
| 〔原因〕           | 入出力状態=9F。<br>〔Standard〕フェイルオーバーが発生した (IFAS FILE および IFASPRO RDB 使用時)。                                          |
| 〔処置〕           | 解除キーを押せばプログラムは異常終了する。                                                                                          |
| 〔対策〕           | IFAS FILE, IFASPRO RDB のマニュアルを参照する。                                                                            |
| プリンタ未準備        |                                                                                                                |
| 〔原因〕           | 入出力状態=30。<br>〔Standard〕プリンタの準備ができていない。                                                                         |
| 〔処置〕           | 解除キーを押せばプログラムは異常終了する。                                                                                          |
| 〔対策〕           | COBOL アプリケーション実行環境 (プリンタ機能) のマニュアルを参照して、プリンタを準備する。                                                             |
| SGミス           |                                                                                                                |
| 〔原因〕           | 入出力状態=30。<br>〔Standard〕COBOL アプリケーション実行環境 (プリンタ機能) を使用するプリンタファイルに対して、OPEN 文の実行が失敗した。<br>たとえば、プリンタ番号が未定義の場合である。 |
| 〔処置〕           | 解除キーを押せばプログラムは異常終了する。                                                                                          |
| 〔対策〕           | COBOL アプリケーション実行環境 (プリンタ機能) のマニュアルを参照して、COBOL アプリケーション実行環境設定ユーティリティ AERO でプリンタ番号を定義する。                         |
| 用紙切れ           |                                                                                                                |
| 〔原因〕           | 入出力状態=97。<br>〔Standard〕紙切れ (COBOL アプリケーション実行環境 (プリンタ機能) 使用時)。                                                  |
| 〔処置〕           | 解除キーを押せばプログラムは再開する。                                                                                            |
| 〔対策〕           | 紙を補給する。                                                                                                        |

|         |                                                                                                                                                                                          |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| アクセスエラー |                                                                                                                                                                                          |
| [原因]    | <p>入出力状態=30。</p> <ul style="list-style-type: none"> <li>・詳細ステータスを参照(COBOL アプリケーション実行環境(プリンタ機能)使用時)。</li> <li>・プリンタ出力先のフォルダやファイルに対するアクセス権限がない(COBOL アプリケーション実行環境(プリンタ機能)未使用時)。</li> </ul> |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | <p>・COBOL アプリケーション実行環境(プリンタ機能)のマニュアルを参照して、原因を特定し取り除く。(COBOL アプリケーション実行環境(プリンタ機能)使用時)</p> <p>・ファイルのアクセス権限を見直す。(COBOL アプリケーション実行環境(プリンタ機能)未使用時)</p>                                        |
| プリンタ使用中 |                                                                                                                                                                                          |
| [原因]    | <p>入出力状態=91。</p> <p>プリンタ使用中である。</p>                                                                                                                                                      |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | プリンタファイルの共用排他を見直す。                                                                                                                                                                       |
| フォームエラー |                                                                                                                                                                                          |
| [原因]    | <p>入出力状態=30。</p> <p>[Standard] フォームに関するエラーを検出した(COBOL アプリケーション実行環境(プリンタ機能)使用時)。</p>                                                                                                      |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | COBOL アプリケーション実行環境(プリンタ機能)のマニュアルを参照して、原因を特定し取り除く。                                                                                                                                        |
| 端末名取得不能 |                                                                                                                                                                                          |
| [原因]    | <p>入出力状態=30。</p> <p>[Standard] 端末名取得不能エラーを検出した(COBOL アプリケーション実行環境(プリンタ機能)使用時)。</p>                                                                                                       |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | COBOL アプリケーション実行環境(プリンタ機能)のマニュアルを参照して、原因を特定し取り除く。                                                                                                                                        |

|         |                                                                                                                                                                                          |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| プリンタ異常  |                                                                                                                                                                                          |
| [原因]    | 入出力状態=30。<br>[Standard] プリンタ異常のエラーを検出した(COBOL アプリケーション実行環境(プリンタ機能)使用時)。                                                                                                                  |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | COBOL アプリケーション実行環境(プリンタ機能)のマニュアルを参照して、原因を特定し取り除く。                                                                                                                                        |
| システムエラー |                                                                                                                                                                                          |
| [原因]    | 入出力状態=30。<br>[Standard] プリントマネージャ経由のプリンタに対して、システムエラーを検出した。                                                                                                                               |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | 「13.2.4 エラーコード取得」のエラーコードの意味を元に、原因を特定し取り除く。                                                                                                                                               |
| レジストリ不正 |                                                                                                                                                                                          |
| [原因]    | 入出力状態=30。<br>[Standard] プリントマネージャ経由のプリンタに対して、レジストリ不正エラーを検出した。                                                                                                                            |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | 通常使うプリンタまたは COBOL アプリケーションカスタマイザでメディア識別名に対応付けたプリンタへのアクセス権限があるか、プリンタドライバが正しくインストールされているかを確認する。                                                                                            |
| プリンタ未設定 |                                                                                                                                                                                          |
| [原因]    | 入出力状態=30。<br>[Standard] プリンタ未設定エラーを検出した。<br>次のいずれかである。<br>(1) 通常使うプリンタが設定されていない。<br>(2) プリンタファイル名の長さ(末尾の連続する空白を除く)が 246 より大きい。<br>(3) プリンタファイルオープン時、ファイル名、ディレクトリ名、またはボリュームラベルの構文が間違っている。 |
| [処置]    | 解除キーを押せばプログラムは異常終了する。                                                                                                                                                                    |
| [対策]    | (1) 通常使うプリンタを設定する。<br>(2) プリンタファイル名の長さ(末尾の連続する空白を除く)を 246 以下にする。<br>(3) プリンタファイルオープン時、ファイル名、ディレクトリ名、またはボリュームラベルの構文を訂正する。                                                                 |



## D 環境変数

本 COBOL では、いくつかの環境変数により COBOL コンパイラおよび COBOL オブジェクトの実行環境を変更することができる。翻訳時には、COBOL 専用ではないシステムで標準的に規定されているいくつかの環境変数を参照する。

また、COBOL オブジェクトでシステムサブルーチン B\_PUTENV および B\_GETENV を使用することにより、環境変数を設定および参照することができる。

[Standard] 次の Unicode 機能の説明は未サポート。

なお、COBOL で参照する環境変数の値に Unicode 固有文字が指定できるかどうかは、次の規則に従う。

- ・ -CU オプションの指定の有無に関わらず、翻訳時に参照する環境変数の値に Unicode 固有文字を指定することはできない。
- ・ -CU オプションを指定せず生成したアプリケーションを実行するとき、参照する環境変数の値に Unicode 固有文字を指定することはできない。
- ・ -CU オプションを指定して生成したアプリケーションを実行するとき、参照する環境変数の値に Unicode 固有文字を指定することができる。

-CU オプション指定時における環境変数ごとの Unicode 固有文字使用可否は、次の表に記載したとおりである。Windows 版と Linux 版の仕様が異なる場合、上段に Windows 版、下段に Linux 版を記載する。

## (1) ファイル入出力機能関連

| 項番 | 環境変数                                                           | 機能                                                                                                                                                                                                                                                           | Unicode<br>固有文<br>字使用<br>可否 | 備考  |
|----|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-----|
| 1  | COB_FILETAB=n<br>( $1 \leq n \leq 16$ )                        | 行順ファイルの入力において、TAB コードをいくつかの空白に拡張し、n を法とする境界に合わせる。                                                                                                                                                                                                            | 不可                          | 実行時 |
| 2  | COB_SYSIN                                                      | 小入力 ACCEPT で使用する SYSIN ファイル名をフルパスで指定する。                                                                                                                                                                                                                      | 可<br>不可                     | 実行時 |
| 3  | COB_SYSOUT                                                     | 小出力 DISPLAY で使用する SYSOUT ファイル名をフルパスで指定する。                                                                                                                                                                                                                    | 可<br>不可                     | 実行時 |
| 4  | xxx_SYSIN                                                      | SELECT 句の「xxx-SYSIN」(xxx はファイル識別リテラル)に対して実ファイルのパスを指定する。                                                                                                                                                                                                      | 可<br>不可                     | 実行時 |
| 5  | xxx_SYSOUT                                                     | SELECT 句の「xxx-SYSOUT」(xxx はファイル識別リテラル)に対して実ファイルのパスを指定する。                                                                                                                                                                                                     | 可<br>不可                     | 実行時 |
| 6  | COB_SYSOUT_FORMAT                                              | DISPLAY 文(データ例外表示)を除く SYSOUT のレコード形式を変更する。<br>SSF 指定<br>SSF ヘッダ(制御データ)付きの SYSOUT ファイル既定のレコード形式で出力する。<br>PRN 指定<br>PRN プリンタファイルのレコード形式で出力する。<br>TXT 指定<br>テキストファイルのレコード形式で出力する。右端の空白は削除する。<br>上記以外または指定なし<br>[Enterprise] SSF 指定となる。<br>[Standard] TXT 指定となる。 | 不可                          | 実行時 |
| 7  | [Enterprise for Windows]<br>[Standard]<br>COB_SYSOUT_FORMAT_4D | [Enterprise for Windows] [Standard] 小出力 DISPLAY で使用する SYSOUT ファイルのレコード形式を変更する。<br>SSF 指定<br>SSF ヘッダ(制御データ)付きの SYSOUT ファ                                                                                                                                      | 不可                          | 実行時 |

|   |                                            |                                                                                                                                                                                                                                                                                                                                          |    |     |
|---|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|   |                                            | <p>イル既定のレコード形式で出力する。</p> <p><b>PRN 指定</b><br/>PRN プリンタファイルのレコード形式で出力する。</p> <p><b>TXT 指定</b><br/>テキストファイルのレコード形式で出力する。<br/>右端の空白は削除する。</p> <p>未定義または SSF, PRN, TXT 以外の指定<br/>環境変数 COB_SYSOUT_FORMAT の指定に従う。</p>                                                                                                                          |    |     |
| 8 | COB_SYSOUT_CODE<br>= { UTF-8 UCS-2 S-JIS } | <p>小出力 DISPLAY で使用する SYSOUT ファイルの文字集合を指定する。(-CU オプション指定で生成されたアプリケーションにのみ有効)</p> <p><b>UTF-8 指定</b><br/>プログラムの固有文字集合から UTF-8 に変換して出力する。</p> <p><b>UCS-2 指定</b><br/>プログラムの固有文字集合から UTF-16LE に変換して出力する。</p> <p><b>S-JIS 指定</b><br/>プログラムの固有文字集合からシフト JIS に変換して出力する。</p> <p>上記以外または指定なし<br/>コード変換せず、そのまま出力される。</p>                           | 不可 | 実行時 |
| 9 | [Enterprise for Windows]<br>xxx_PMD        | <p>[Enterprise for Windows] SELECT 句で内部ファイル名 (xxx) を指定した大容量記憶ファイルの順／行順／CSV ファイルのオープンモードを変更して実行する。</p> <p>オープンモードの変更はプロセス毎に最初の 1 回目のみ有効である。1 度 CLOSE して再度 OPEN するとき (2 回目以降), OPEN 文の記述どおり実行する。</p> <p>プログラム中に同じ内部ファイル名を指定した別の SELECT 句がある場合, オープンモードが変更されるのは先に OPEN するファイルのみである。</p> <p><b>APPEND 指定</b><br/>最初のオープンモードが OUTPUT のとき,</p> | 不可 | 実行時 |

|    |                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |    |     |
|----|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                                    | <p>EXTEND オープンとして実行する。</p> <p>OUTPUT 指定</p> <p>最初のオープンモードが EXTEND のとき、<br/>OUTPUT オープンとして実行する。</p> <p>未定義または APPEND, OUTPUT 以外を指定<br/>環境変数 COB_FILE_APPEND の指定に従う。</p> <p>※本機能を有効にするには、V5.4 以降でコンパイルが必要です。V5.3 以前に作成したアプリケーションでは有効になりません。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |     |
| 10 | <p>[Standard]</p> <p>COB_FILE_SHARING<br/>= { WAIT   NO_WAIT }</p> | <p>ファイルの排他制御時の動作を指定する。</p> <ul style="list-style-type: none"> <li>• SYSOUT/SYSIN ファイルの場合<br/>環境変数 COB_FILE_SHARING の指定は無効である。</li> <li>• 大容量記憶ファイル/RDB ファイルの場合<br/>WAIT を指定した場合<br/>他のプログラムで排他制御を行っている<br/>ファイルに対して OPEN 文を実行した場合、<br/>OPEN 文は保留される。<br/>ファイル種別が COBOL DB アクセス, IFAS<br/>FILE, および IFASPRO RDB の場合、OS 互<br/>換モードが A-VX2 の場合のみファイル排<br/>他制御の設定が有効となり、OS 互換モー<br/>ドが A-VX5 の場合は、NO_WAIT が指定さ<br/>れた場合と同じ動作になる。</li> </ul> <p>NO_WAIT を指定した場合<br/>他のプログラムで排他制御を行っている<br/>ファイルに対して OPEN 文を実行した場合、<br/>次の様な動作となる。</p> <ul style="list-style-type: none"> <li>• FILE STATUS 句がある場合は、入<br/>出力状態 '91' を返却する。</li> <li>• FILE STATUS 句がない場合は、<br/>実行時メッセージを出力しプログ<br/>ラムが異常終了する。</li> </ul> <p>WAIT および NO_WAIT 以外の指定ありまたは<br/>指定なしの場合<br/>コンパイラ動作既定値に従い、WAIT が指</p> | 不可 | 実行時 |

|    |                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |     |
|----|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                            | <p>定されたものとみなす。</p> <p>COBOL 独自形式の相対ファイルおよび ISAM 形式の索引ファイルに対して、環境変数 COB_FILE_SHARING の指定は無効であり、常に NO_WAIT が指定されたものと見なす。</p> <ul style="list-style-type: none"> <li>プリンタファイルの場合<br/>環境変数 COB_FILE_SHARING の有無にかかわらず、次の様な動作となる。<br/>プリントマネージャ経由の場合<br/>排他制御は行わない。<br/>PRNAMR 経由のプリンタの場合<br/>他のプログラムで使用中のプリンタに対して OPEN 文を実行した場合、次の様な動作となる。 <ul style="list-style-type: none"> <li>FILE STATUS 句がある場合は、入出力状態 '91' を返却する。</li> <li>FILE STATUS 句がない場合は、実行時メッセージを出力し、解除キー待ちになる。</li> </ul> </li> </ul> |    |     |
| 11 | <p>[Standard]</p> <p>COB_RDB_STATUS<br/>= { ON   OFF }</p> | <p>特殊レジスタ RDB-STATUS に COMMIT / ROLLBACK 文の実行結果の通知方法を指定する。</p> <p>ON を指定した場合<br/>COMMIT または ROLLBACK 文の実行結果は、常に特殊レジスタ RDB-STATUS に通知される。</p> <p>OFF を指定または、環境変数 COB_RDB_STATUS を指定しなかった場合<br/>COMMIT または ROLLBACK 文の実行結果が異常終了したとき、実行時メッセージ を出力の上でプログラムが異常終了する。</p> <p>RDB-STATUS に返却される値については、各データベースエンジンのマニュアルを参照。</p>                                                                                                                                                                         | 不可 | 実行時 |
| 12 | <p>[Standard]</p> <p>COB_PRNTHR = ON</p>                   | <p>プリンタ機能の PASSTHROUGH モードの使用を設定する。PASSTHROUGH モードを使用することにより、プリンタシーケンスの発行が可能となる。</p> <p>PASSTHROUGH モード機能を使用する場合に設</p>                                                                                                                                                                                                                                                                                                                                                                          | 不可 | 実行時 |

|       |                                                            | <p>定値が ON ではない場合、環境変数の指定がないものとして動作する。</p> <p>また、レジストリと環境変数を個々にまたは同時に指定された場合は、以下の表の組み合わせで動作が決定される。</p> <table><tr><th>レジストリ</th><th>環境変数</th><th>動作</th></tr><tr><td>指定なし</td><td>ON</td><td>PASSTHROUGH</td></tr><tr><td>ON</td><td>ON</td><td rowspan="2">モード機能使用</td></tr><tr><td>ON</td><td>指定なし</td></tr><tr><td>指定なし</td><td>指定なし</td><td>PASSTHROUGH</td></tr><tr><td>OFF</td><td>指定なし</td><td rowspan="2">モード機能未使用</td></tr><tr><td>OFF</td><td>ON</td></tr></table>                                                                                                                                                              | レジストリ | 環境変数 | 動作 | 指定なし | ON   | PASSTHROUGH                                     | ON | ON   | モード機能使用 | ON | 指定なし | 指定なし | 指定なし | PASSTHROUGH | OFF                                | 指定なし | モード機能未使用 | OFF | ON |     |     |    |     |
|-------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------|----|------|------|-------------------------------------------------|----|------|---------|----|------|------|------|-------------|------------------------------------|------|----------|-----|----|-----|-----|----|-----|
| レジストリ | 環境変数                                                       | 動作                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| 指定なし  | ON                                                         | PASSTHROUGH                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| ON    | ON                                                         | モード機能使用                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| ON    | 指定なし                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| 指定なし  | 指定なし                                                       | PASSTHROUGH                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| OFF   | 指定なし                                                       | モード機能未使用                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| OFF   | ON                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| 13    | <p>[Standard]</p> <p>COB_AMRMODE</p> <p>= { ON   OFF }</p> | <p>PRNAMR 経由での印刷はプリントマネージャ経由の印刷かを指定する。</p> <p>COBOL アプリケーション実行環境がインストールされている環境の場合</p> <p>レジストリ CBLAMRMODE と環境変数 COB_AMRMODE を個々にまたは同時に指定された場合、以下の表の組み合わせで動作が決定される。</p> <table><tr><th>レジストリ</th><th>環境変数</th><th>動作</th></tr><tr><td>指定なし</td><td>指定なし</td><td rowspan="4">PRNAMR を使用する(プリントマネージャを使用しない)COBOL アプリケーション用の設定</td></tr><tr><td>ON</td><td>指定なし</td></tr><tr><td>ON</td><td>ON</td></tr><tr><td>ON</td><td>OFF</td></tr><tr><td>指定なし</td><td>OFF</td><td rowspan="4">PRNAMR を使用しない(プリントマネージャを使用する)COBOL</td></tr><tr><td>OFF</td><td>指定なし</td></tr><tr><td>OFF</td><td>ON</td></tr><tr><td>OFF</td><td>OFF</td></tr></table> | レジストリ | 環境変数 | 動作 | 指定なし | 指定なし | PRNAMR を使用する(プリントマネージャを使用しない)COBOL アプリケーション用の設定 | ON | 指定なし | ON      | ON | ON   | OFF  | 指定なし | OFF         | PRNAMR を使用しない(プリントマネージャを使用する)COBOL | OFF  | 指定なし     | OFF | ON | OFF | OFF | 不可 | 実行時 |
| レジストリ | 環境変数                                                       | 動作                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| 指定なし  | 指定なし                                                       | PRNAMR を使用する(プリントマネージャを使用しない)COBOL アプリケーション用の設定                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| ON    | 指定なし                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| ON    | ON                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| ON    | OFF                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| 指定なし  | OFF                                                        | PRNAMR を使用しない(プリントマネージャを使用する)COBOL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| OFF   | 指定なし                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| OFF   | ON                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |
| OFF   | OFF                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |      |    |      |      |                                                 |    |      |         |    |      |      |      |             |                                    |      |          |     |    |     |     |    |     |

|    |                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |  |              |    |     |
|----|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--------------|----|-----|
|    |                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |  | アプリケーション用の設定 |    |     |
|    |                                                | COBOL アプリケーション実行環境がインストールされていない環境の場合<br>本環境変数は無効である。<br>ON/OFF 以外の値を指定した場合<br>指定なしとみなす。                                                                                                                                                                                                                                                                                                                                                                |  |              |    |     |
| 14 | [Standard]<br>COB_PRNFONT<br>= フォント名[,フォントサイズ] | プリントマネージャ経由の印刷において使用する文字フォントを指定する。<br>フォント名, フォントサイズを指定した場合<br>プリントマネージャ経由の印刷において, 指定したフォント名, フォントサイズが使用される。<br>フォント名がない場合やフォント名が誤っている場合<br>指定がないものとして動作し, システムで定義されているフォントが使用される。ただし, フォント名は1字でも間違いがあると, フォント名とは認識されない。<br>フォントサイズが省略された場合または指定が誤っている場合<br>10 ポイントとする。<br>フォントサイズにフォントに対応されていない数値を指定した場合<br>近いサイズを自動的に選択する。<br><br>プリントマネージャ経由の印刷において使用する文字フォントは, アプリケーションカスタマイザによっても指定可能となる。アプリケーションカスタマイザと環境変数を同時に指定した場合, アプリケーションカスタマイザでの指定が優先とする。 |  |              | 不可 | 実行時 |
| 15 | [Standard]<br>COB_CID_INIFILE                  | 内部ファイル ID 置換表のファイル名をフルパスで指定する。                                                                                                                                                                                                                                                                                                                                                                                                                         |  |              | 不可 | 実行時 |
| 16 | [Enterprise for Windows]<br>[Standard]         | [Enterprise for Windows] [Standard] DISPLAY 文(データ例外表示)を除く SYSOUT ファイルの改                                                                                                                                                                                                                                                                                                                                                                                |  |              | 不可 | 実行時 |

|    |                                      |                                                                                                                                                                                                                                                                                                                                                                    |    |     |
|----|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    | COB_SYSOUT_NEWLINE = ON              | <p>行コードを指定する。</p> <p>ON 指定<br/>SYSOUT ファイルでは改行コード (0x0D0A) を出力する。</p> <p>未定義または ON 以外を指定<br/>SYSOUT ファイルでは改行コード (0x0A) を出力する。</p> <p>[Enterprise for Windows] 下記のすべての条件を満たす場合, 改行コード (0x0D000A00) に変更する。</p> <ul style="list-style-type: none"> <li>・標準動作モードで, -CU ありかつ CODE-SET 句の符号系名で参照される文字集合が UCS-2。</li> <li>・環境変数 COB_SYSOUT_NEWLINE に ON を指定する。</li> </ul> |    |     |
| 17 | [Enterprise]<br>COB_FILE_APPEND = ON | <p>[Enterprise] 大容量記憶ファイルの順/行順/CSV ファイルの OUTPUT オープンを EXTEND オープンで実行する。</p> <p>ON 指定<br/>OUTPUT オープンを EXTEND オープンで実行する。</p> <p>EXTEND オープンの切り換えは最初の 1 回目のみ有効である。1 度 CLOSE して再度 OPEN するとき (2 回目以降), OPEN 文の記述どおり実行する。</p> <p>未定義または ON 以外を指定<br/>OPEN 文の記述どおり実行する。</p>                                                                                              | 不可 | 実行時 |
| 18 | COB_SYSIN_READ = ON                  | <p>SYSIN ファイルから READ 文で 80 桁を超えるレコードを読み込めるようにすることを指定する。</p> <p>ON を指定した場合,<br/>最大でレコード記述項で指定した大きさ分ファイルから読み込む。</p> <p>未定義または ON 以外の指定ありの場合,<br/>最大で 80 桁分ファイルから読み込む。</p> <p>いずれの場合も, ファイルから読み込んだレコードの大きさがレコード記述項で指定した大きさに</p>                                                                                                                                       | 不可 | 実行時 |



|    |                                                                       |                                                                                                                                                                                                                                                                                                                                                                                |    |     |
|----|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                                       | <p>満たない場合は、残りのレコード領域には半角空白が埋められる。</p> <p>また、ファイルから読み込んだレコードの大きさが読み込み可能な最大の大きさを超える場合は、レコードは右側が切り捨てられ最大の大きさになる。</p>                                                                                                                                                                                                                                                              |    |     |
| 19 | <p>[Enterprise for Windows]<br/>[Standard]<br/>COB_SYSIO_NUL = ON</p> | <p>SYSIN/SYSOUT ファイルのファイル名にデバイス名 NUL を指定することを可能とする。</p> <p>・READ 文によるファイル名が NUL の SYSIN ファイル読み込み時</p> <p>ON を指定した場合、<br/>AT END になる。</p> <p>未定義または ON 以外の指定ありの場合、<br/>「COB101 ファイルが開けません」実行時エラーとなる。</p> <p>・WRITE 文によるファイル名が NUL の EXTEND 指定 SYSOUT ファイルへの書き込み時</p> <p>ON を指定した場合、<br/>エラーは発生せず、ファイル出力もしない。</p> <p>未定義または ON 以外の指定ありの場合、<br/>「COB101 ファイルが開けません」実行時エラーとなる。</p> | 不可 | 実行時 |
| 20 | <p>[Enterprise for Windows]<br/>COB_CSV_NOQUOTE = ON</p>              | <p>CSV ファイルの文字型項目出力時、ダブルクォーテーションで囲まず出力することを可能とする。</p> <p>環境変数はアプリケーションでの行順ファイル (CSV 形式) の初回オープン時に一度だけ取得され、アプリケーション内で同じ値が保証される。したがってファイル毎に異なる設定をしたり、アプリケーション実行中に設定を変更したりすることはできない。</p> <p>ON を指定した場合、<br/>CSV ファイルの文字型項目出力時、ダブルクォーテーションで囲まず出力する。</p> <p>例) セルのデータが ABC△△ の場合</p>                                                                                                | 不可 | 実行時 |

|    |                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |     |
|----|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                                                           | <p>※ ”△”は空白を表す<br/>         …,ABC△△,…</p> <p>未定義または ON 以外の指定ありの場合,<br/>         CSV ファイルの文字型項目出力時, ダブルクォーテーションで囲んで出力する。</p>                                                                                                                                                                                                                                                                                                                                                                                              |    |     |
| 21 | <p>[Enterprise for Windows]<br/>         [Standard]<br/>         COB_CSV_TRIMEND = ON</p> | <p>CSV ファイルの文字型項目出力時のセル末尾空白(1 バイト/2 バイト共に)を削除することを可能とする。<br/>         (-CF/-Cj オプション指定なしで生成されたアプリケーションにのみ有効)</p> <p>環境変数はアプリケーションでの行順ファイル (CSV 形式)の初回オープン時に一度だけ取得され, アプリケーション内で同じ値が保証される。<br/>         したがってファイル毎に異なる設定をしたり, アプリケーション実行中に設定を変更したりすることはできない。</p> <p>ON を指定した場合,<br/>         CSV ファイルの文字型項目出力時のセル末尾空白(1 バイト/2 バイト共に)を削除する。<br/>         例) セルのデータが ABC△△ の場合<br/>         ※ ”△”は空白を表す<br/>         …,"ABC",…</p> <p>未定義または ON 以外の指定ありの場合,<br/>         CSV ファイルの文字型項目出力時のセル末尾空白(1 バイト/2 バイト共に)を削除しない。</p> | 不可 | 実行時 |

## (2) その他, COBOL 専用環境変数

| 項番 | 環境変数                                                                                                                                                 | 機能                                                                                                                                                                                                                                                                                                                                                                            | Unicode<br>固有文字<br>使用可否 | 備考  |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|-----|
| 1  | COB_COPY                                                                                                                                             | <p>翻訳時に登録集原文が存在するパス名を指定する。</p> <p>検索の優先順位は-I オプション&gt;COB_COPY&gt;COBOL 原始プログラムの存在場所となる。</p> <p>[Enterprise for Windows] [Standard] 複数のパスを設定する場合, 区切り記号として“;”を指定する</p> <p>(例: set<br/>COB_COPY=C:¥path¥to¥copy;C:¥path¥to¥copy<br/>2)</p> <p>[Enterprise for Linux] 複数のパスを設定する場合, 区切り記号として“:”を指定する</p> <p>(例: export<br/>COB_COPY=/home/path/to/copy:/home/path/to/copy2)</p> | 不可                      | 翻訳時 |
| 2  | [Enterprise for Linux]<br>COB_LINK_PARAM                                                                                                             | リンカに渡すオプションを指定する                                                                                                                                                                                                                                                                                                                                                              | 不可                      | 翻訳時 |
| 3  | COB_JOBDATE                                                                                                                                          | 業務日付を設定する。                                                                                                                                                                                                                                                                                                                                                                    | 不可                      | 実行時 |
| 4  | <p>[Enterprise for Windows]<br/>[Standard]<br/>COB_LIBRARY<br/>= DLL 名[;DLL 名]…</p> <p>[Enterprise for Linux]<br/>COB_LIBRARY<br/>= so 名[:so 名]…</p> | <p>[Enterprise for Windows] [Standard] 動的リンク時にリンクを行うべき DLL 名をフルパス, またはパス指定せずファイル名のみで指定する。ファイル名のみで指定した場合の DLL 検索規則は, 「7.11.1.4 DLL の検索」を参照。</p> <p>[Enterprise for Linux] 動的リンク時にリンクを行うべき共有ライブラリ名をフルパス, またはパス指定せずファイル名のみで指定する。ファイル名のみで指定した場合の共有ライブラリ検索規則は, 「7.11.2.4 共有ライブラリの検索」を参照。</p>                                                                                       | 可<br>不可                 | 実行時 |
| 5  | COB_RUNERR                                                                                                                                           | NO_PAUSE 指定時は,                                                                                                                                                                                                                                                                                                                                                                | 不可                      | 実行時 |

|    |                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |         |     |
|----|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-----|
|    | = {NO_PAUSE   NO_DISPLAY}         | <p>実行時エラーメッセージの一時停止を行わない。連続実行に便利である。</p> <p>WS-EML 配下でないバッチ型アプリケーションでのみ有効。</p> <p>NO_DISPLAY 指定時は、</p> <p>実行時エラーメッセージを表示しない。</p> <p>未定義または NO_PAUSE, NO_DISPLAY 以外を指定時は、</p> <p>実行時エラーメッセージを出力後、確認キーの入力待ちとなり、処理を一時停止する。</p>                                                                                                                                                                                                                                                |         |     |
| 6  | COB_SWITCH_FILE<br>= ファイル名        | 外部スイッチ状態を保存するファイル名をフルパスで指定する。                                                                                                                                                                                                                                                                                                                                                                                                                                              | 可<br>不可 | 実行時 |
| 7  | COB_ABTASK                        | 1～255 の異常終了値を設定する。                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 不可      | 実行時 |
| 8  | COB_UC_FILE                       | 利用者制御変数の値を保存するファイル名をフルパスで指定する。                                                                                                                                                                                                                                                                                                                                                                                                                                             | 可<br>不可 | 実行時 |
| 9  | COB_CODEERR = { STOP   CONTINUE } | <p>-CU オプション指定時、標準出力／標準エラー出力への出力時、コード変換が失敗した場合の動作を指定する。</p> <p>【Enterprise for Linux】環境変数<br/>LANG=ja_JP.UTF-8 指定時かつ環境変数<br/>COB_CODECHG=ON 指定時、標準出力／標準エラー出力／CONSOLE への出力時、コード変換が失敗した場合の動作を指定する。</p> <p>STOP 指定(デフォルト)</p> <p>実行時エラーメッセージを表示し、終了する。</p> <p>CONTINUE 指定</p> <p>-CU オプション指定時、変換に失敗した文字を 00<sub>16</sub> に置き換えて処理を継続する。</p> <p>【Enterprise for Linux】環境変数<br/>LANG=ja_JP.UTF-8 指定時かつ環境変数<br/>COB_CODECHG=ON 指定時、変換に失敗した文字を 00<sub>16</sub> に置き換えて処理を継続する。</p> | 不可      | 実行時 |
| 10 | COB_GCMOVE = { ON   OFF }         | <p>英数字項目から日本語項目への転記で、英数字項目に濁点が含まれる場合、</p> <p>カタカナの濁点を結合する。</p> <p>(-CU オプション指定なしで生成されたアプリケ</p>                                                                                                                                                                                                                                                                                                                                                                             | 不可      | 実行時 |

|    |                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |
|----|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                                          | <p>ーションにのみ有効)</p> <p>ON 指定</p> <p>カタカナの濁点を結合する。</p> <p>OFF 指定</p> <p>カタカナの濁点を結合しない。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |     |
| 11 | <p>[Standard]</p> <p>COB_EVENTLOG = { ON   WSEML   NON_WSEML   ALL }</p> | <p>実行時エラーとなった場合の動作を指定する。<br/>環境変数 COB_RUNERR により動作が決まる。</p> <p>環境変数 COB_RUNERR に NO_DISPLAY 指定時、<br/>COB_EVENTLOG は無効である。</p> <p>環境変数 COB_RUNERR に NO_DISPLAY 未指定時、<br/>指定値が ON の場合</p> <p>実行時エラーメッセージをイベントログとして出力し、確認キーの入力待ちを行わず、処理を終了する。</p> <p>指定値が WSEML の場合</p> <p>実行時エラーメッセージをイベントログとして出力するとともに、実行時エラーメッセージを COBOL WS エミュレータに出力後、確認キーの入力待ちとなり、処理を一時停止する。<br/>(COBOL WS エミュレータ配下で動作するアプリケーションにのみ有効)</p> <p>指定値が NON_WSEML の場合</p> <p>実行時エラーメッセージをイベントログとして出力するとともに環境変数 COB_RUNERR の指定に従った動作を行う。<br/>(COBOL WS エミュレータ配下でないアプリケーションにのみ有効)</p> <p>指定値が ALL の場合</p> <p>COBOL WS エミュレータ配下のアプリケーションの場合、WSEML を指定したのと同じように動作する。<br/>COBOL WS エミュレータ配下でないアプリケーションの場合、NON_WSEML を指定したのと同じように動作する。</p> | 不可 | 実行時 |

|    |                                               |                                                                                                                                                                                                                                                                                                    |    |     |
|----|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                               | <p>未定義または指定値が ON, WSEML, NON_WSEML, ALL でない場合</p> <p>環境変数 COB_RUNERR が未定義の場合, 実行時エラーメッセージを出力後, 確認キーの入力待ちとなり, 処理を一時停止する。</p> <p>環境変数 COB_RUNERR に NO_PAUSE 指定時, 実行時エラーメッセージを出力後, 確認キーの入力待ちを行わず, 処理を続行する。</p>                                                                                        |    |     |
| 12 | <p>[Standard]</p> <p>COB_WSJISNOTCHG = ON</p> | <p>COBOL WS エミュレータ使用時に行われるデータのシフト JIS と JIPS コード変換において, JIS78/90 文字の入替処理を抑止する。</p> <p>未定義または ON 以外を指定した場合, JIS78/90 文字の入替処理を行う。</p>                                                                                                                                                                | 不可 | 実行時 |
| 13 | <p>[Standard]</p> <p>COB_KANAIME = ON</p>     | <p>アプリケーションカスタマイザの入力モードの設定がカタカナ(既定値)の場合, INPUT MODE IS KANA-SHIFT 句が指定された項目にキャレットが位置付いた際に, 日本語入力システムを起動し, 半角カタカナ文字を入力できる状態にする。</p> <p>本環境変数は COBOL WS エミュレータ配下でない対話型アプリケーションで有効である。</p> <p>ON 指定</p> <p>日本語入力システムを起動する。入力モード(カナ入力/ローマ字入力)は IME の設定に従う。</p> <p>未定義または ON 以外を指定</p> <p>動作に影響しない。</p> | 不可 | 実行時 |
| 14 | <p>[Standard]</p> <p>COB_GDDMAX = ON</p>      | <p>STOP 定数文および機能名 GDD に対応づけられた呼び名への DISPLAY 文の表示可能な文字数を拡張する。ただし, メッセージ出力中の画面ハードコピー機能は使用不可となる。</p> <p>本環境変数は COBOL WS エミュレータ配下でない対話型アプリケーションで有効である。</p> <p>ON 指定</p> <p>表示可能な文字数は 236 文字</p> <p>未定義または ON 以外を指定</p> <p>表示可能な文字数は 59 文字</p>                                                          | 不可 | 実行時 |

|    |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |     |
|----|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
| 15 | COB_LIBERR = CONTINUE | <p>[Enterprise for Windows] [Standard] 環境変数 COB_LIBRARY に指定した DLL が存在しない場合の動作を指定する。</p> <p>CONTINUE 指定</p> <p>存在しない DLL 名を無視し、後続の DLL の検索を継続する。</p> <p>未定義または CONTINUE 以外を指定</p> <p>DLL の検索を終了し、実行時エラーを出力する。DLL 名に Unicode 固有文字を含む場合は、埋め込み情報のファイル名が正しく表示されない。</p> <p>[Enterprise for Linux] 環境変数 COB_LIBRARY に指定した共有ライブラリが存在しない場合の動作を指定する。</p> <p>CONTINUE 指定</p> <p>存在しない共有ライブラリ名を無視し、後続の共有ライブラリの検索を継続する。</p> <p>未定義または CONTINUE 以外を指定</p> <p>共有ライブラリの検索を終了し、実行時エラーを出力する。</p> | 不可 | 実行時 |
| 16 | COB_CALLERR = ON      | <p>CALL 一意名の呼び出しでエントリが見つからない場合に詳細情報を出力するかを指定する。</p> <p>ON 指定</p> <p>標準エラー出力に詳細情報を出力する。</p> <p>未定義または ON 以外を指定</p> <p>詳細情報を出力しない。</p>                                                                                                                                                                                                                                                                                                                                           | 不可 | 実行時 |
| 17 | COB_TRACE = ファイル名     | <p>トレース情報の出力を指定する。</p> <p>ファイル名 (フルパス付き) 指定</p> <p>指定ディレクトリにトレース情報を出力する。</p> <p>指定ファイル名が abc.csv の場合、トレース情報ファイル名は次のとおり。</p> <p>「abc_日付_時刻_プロセス ID_スレッド ID.csv」</p> <p>※日付 (YYYYMMDD) は西暦年月日</p> <p>※時刻 (HHMMSSXX) は時分秒 1/100 秒</p> <p>※シングルスレッドアプリケーションでもスレッド</p>                                                                                                                                                                                                              | 不可 | 実行時 |

|    |                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |     |
|----|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                   | <p>IDを持っているので、そのままスレッド ID を使用する。</p> <p>上記以外または未定義<br/>トレース情報を出力しない。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |     |
| 18 | <p>COB_TRACE_LEVEL</p> <p>= { 0   1   2   3 }</p> | <p>トレース情報の出力レベルを指定する。</p> <p><b>0 指定</b><br/>実行トレース(カバレッジ情報を含まない)のみ出力する。</p> <p><b>1 指定</b><br/>実行トレース(カバレッジ情報を含まない)とレコードダンプ(レコードダンプ, キー情報(位置, 長さ, ダンプ), 副キー情報(個数, 位置, 長さ, ダンプ))を出力する。</p> <p><b>2 指定</b><br/>COBOL Assistant Option インストール済み, かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合, 実行トレース(カバレッジ情報を含む)のみ出力する。<br/>実行トレース(カバレッジ情報を含む)に通過時刻情報を出力する。<br/>その他の場合, 実行トレース(カバレッジ情報を含まない)のみ出力する。</p> <p><b>3 指定</b><br/>COBOL Assistant Option インストール済み, かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合, 実行トレース(カバレッジ情報を含む)とレコードダンプ(レコードダンプ, キー情報(位置, 長さ, ダンプ), 副キー情報(個数, 位置, 長さ, ダンプ))を出力する。実行トレース(カバレッジ情報を含む)とレコードダンプに通過時刻情報を出力する。<br/>その他の場合, 実行トレース(カバレッジ情報を含まない)とレコードダンプ(レコードダンプ, キー情報(位置, 長さ, ダンプ), 副キー情報(個数, 位置, 長さ, ダンプ))を出力する。</p> | 不可 | 実行時 |



|    |                                         |                                                                                                                                                                                                                                                                                                     |    |     |
|----|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                         | <p>上記以外または未定義</p> <p>COB_TRACE=ファイル名指定ありのとき, 実行トレースのみ出力する。</p> <p>COB_TRACE=ファイル名指定なしのとき, トレース情報は出力しない。</p>                                                                                                                                                                                         |    |     |
| 19 | COB_OBJFILE<br>= { NO_DELETE   DELETE } | <p>1 つの原始プログラムを翻訳し, リンクした場合の, 生成したオブジェクトファイルの削除動作を指定する。</p> <p>NO_DELETE を指定<br/>オブジェクトファイルを削除しない。</p> <p>未定義または DELETE を指定<br/>既存動作どおりオブジェクトファイルを削除する。</p>                                                                                                                                         | 不可 | 翻訳時 |
| 20 | COB_TRACE_LINE = ファイル名                  | <p>ソース単位名と行番号指定の部分情報出力を指定する。COB_TRACE_LABEL と同時指定できる。</p> <p>ファイル名(フルパス付き)指定</p> <p>COBOL Assistant Option インストール済み, かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合, ソース単位名と行番号指定の部分情報出力指定ファイル名をフルパスで指定すると, 指定した範囲の実行トレースとレコードダンプを出力する。</p> <p>その他の場合, 部分情報出力は無効である。</p> <p>上記以外または未定義<br/>部分情報を出力しない。</p> | 不可 | 実行時 |
| 21 | COB_TRACE_LABEL = ファイル名                 | <p>ソース単位名と手続き名指定の部分情報の出力を指定する。COB_TRACE_LINE と同時指定できる。</p> <p>ファイル名(フルパス付き)指定</p> <p>COBOL Assistant Option インストール済み, かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合, ソース単位名と手続き名指定の部分情報出力指定ファイル名をフルパスで指定すると, 指定した範囲の実行トレースとレコードダンプを出力する。</p>                                                              | 不可 | 実行時 |

|    |                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |    |     |
|----|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                                                           | <p>その他の場合, 部分情報出力は無効である。<br/>上記以外または未定義<br/>部分情報を出力しない。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |     |
| 22 | <p><b>【Enterprise for Linux】</b><br/>COB_CODECHG = ON</p> | <p>実行時エラーメッセージの出力コードおよび小入出力文の入出力コードをロケールに合わせて自動コード変換する。<br/>設定値が ON の場合<br/>環境変数 LANG が ja_JP.UTF-8 のとき,<br/>・実行時エラーメッセージを UTF-8 の日本語で出力する(トレース情報ファイルへの出力を除く)。<br/>・明示的または暗黙的に CONSOLE, STDOUT, STDERR に対応付けられた呼び名を指定した DISPLAY 文(データ例外表示)が出力する実行時エラーメッセージを UTF-8 の日本語で出力する(トレース情報ファイルへの出力を除く)。<br/>・プログラムの固有文字集合がシフト JIS のとき,<br/>- 明示的または暗黙的に CONSOLE, STDIN に対応付けられた呼び名を指定した ACCEPT 文の字類が日本語の作用対象の入力データを UTF-8 からシフト JIS に変換して入力する。<br/>- 明示的または暗黙的に CONSOLE, STDOUT, STDERR に対応付けられた呼び名を指定した DISPLAY 文の字類が日本語のデータ項目および定数の作用対象の出力データをシフト JIS から UTF-8 に変換して出力する。<br/>上記条件を満たさないとき, 小入出力文の自動コード変換は行わない。また実行時エラーメッセージの出力コードは環境変数 LANG の設定に依存する。<br/>未定義または ON 以外を指定した場合<br/>小入出力文の自動コード変換は行わない。ま</p> | 不可 | 実行時 |

|    |                           |                                                                                                                                              |    |     |
|----|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|----|-----|
|    |                           | た実行時エラーメッセージの出力コードは環境変数 LANG の設定に依存する。                                                                                                       |    |     |
| 23 | COB_TRACE_DATA<br>= ファイル名 | トレース情報に出力するデータ項目指定ファイルを絶対パスで指定する。<br>ファイル名(フルパス付き)指定<br>トレース情報にデータ項目の値を出力する。<br>上記以外<br>実行時エラーを出力し、実行を終了する。<br>未定義<br>トレース情報にデータ項目の値は出力されない。 | 不可 | 実行時 |

## (3) その他,システム環境変数

COBOL 専用ではなく標準的な環境変数であり, Unicode 固有文字使用は不可。

| 項番 | 環境変数                                                                                        | 機能                                     | 備考         |
|----|---------------------------------------------------------------------------------------------|----------------------------------------|------------|
| 1  | PATH                                                                                        | 実行可能なファイルの検索パスを指定する。複数のパスを指定することができる。  | 翻訳時<br>実行時 |
| 2  | [Enterprise for Windows]<br>[Standard] LIB<br><br>[Enterprise for Linux]<br>LD_LIBRARY_PATH | ライブラリファイルの検索パスを指定する。複数のパスを指定することができる。  | リンク時       |
| 3  | [Enterprise for Windows]<br>[Standard] TMP<br><br>[Enterprise for Linux]<br>TMPDIR          | テンポラリファイルの格納パスを指定する。パス指定は1つしかできない。     | 翻訳時        |
| 4  | [Enterprise for Windows]<br>[Standard] TEMP                                                 | TMP と同じ                                | 翻訳時        |
| 5  | [Enterprise for Linux]<br>LANG                                                              | 環境変数 LANG に指定可能なロケールは OS のサポート範囲に依存する。 | 翻訳時<br>実行時 |

## (4) その他,共用環境変数

COBOL 専用ではなく COBOL とその関連製品とで共通に利用する環境変数であり, Unicode 固有文字使用は不可。

| 項番 | 環境変数                                                       | 機能                                                                                                                                                                                       | 備考         |
|----|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1  | [Standard]<br>[Enterprise for Windows]<br>FA_JIS2004_CDMAP | -CF オプションを指定したアプリケーションの作成時および実行時に参照するユーザ定義マッピングファイルの存在するフォルダを指定する。絶対パスで指定する必要がある。<br>(例:set FA_JIS2004_CDMAP=D:¥data¥)<br>※ユーザ定義マッピングファイルの入手方法は、関連製品「FontAvenue UniAssist コード変換」の説明書を参照。 | 翻訳時<br>実行時 |

## E 例外コード一覧

以降に COBOL 独自の例外コード一覧を示す。本例外コードは、例外節内において特殊レジスタ EXCEPTION-CODE で参照することができる。

| 例外コード | 原因                                                                                                                   |
|-------|----------------------------------------------------------------------------------------------------------------------|
| 000   | 内部エラーが発生した                                                                                                           |
| 002   | 添字、可変繰り返し数または部分参照の限界を超えた                                                                                             |
| 003   | 可変長集団項目の繰り返し数が範囲を超えた                                                                                                 |
| 004   | GO TO の飛び先が設定されていない                                                                                                  |
| 006   | INSPECT で長さ不一致のため置換できない                                                                                              |
| 007   | 環境変数 COB_LIBRARY が誤っている                                                                                              |
| 008   | CALL で呼び出すエントリが見つからない                                                                                                |
| 010   | 不正 10 進データを検出した <sup>注(1)</sup>                                                                                      |
| 048   | 利用者制御変数を設定または参照した時にエラーが発生した                                                                                          |
| 049   | 利用者制御変数が存在しない                                                                                                        |
| 093   | 関数の引数が 0 個である                                                                                                        |
| 094   | 不正 10 進データを検出した <sup>注(1)</sup>                                                                                      |
| 095   | 関数の引数の値が範囲外である                                                                                                       |
| 100   | ファイルが指定されていない                                                                                                        |
| 101   | ファイルが開けない                                                                                                            |
| 102   | ファイルが開かれていない                                                                                                         |
| 103   | ファイル入出力中にエラーが発生した                                                                                                    |
| 104   | ファイルがオーバーフローした                                                                                                       |
| 105   | CSV 形式のファイル読み込み中にエラーが発生した                                                                                            |
| 106   | [Enterprise for Windows] [Standard] ファイルパス長は、246 バイト以下である必要がある<br>[Enterprise for Linux] ファイルパス長は、4094 バイト以下である必要がある |
| 107   | [Standard] ファイルとオプション指定が不整合である                                                                                       |
| 135   | 外部スイッチの参照でエラーが発生した                                                                                                   |
| 136   | 外部スイッチの設定でエラーが発生した                                                                                                   |
| 140   | 内部でのコード変換に失敗した                                                                                                       |
| 160   | 不正日本語データを検出した                                                                                                        |
| 201   | サブルーチン aaaa でエラーが発生した(WRONG CHARACTER)                                                                               |
| 202   | サブルーチン aaaa でエラーが発生した(WRONG COMBINATION)                                                                             |
| 203   | サブルーチン aaaa でエラーが発生した(WRONG CHARACTER AND COMBINATION)                                                               |
| 204   | サブルーチン aaaa でエラーが発生した                                                                                                |
| 209   | 数字と符号の重ね合わせ形式が異なる                                                                                                    |
| 210   | -Ci オプションの指定の有無が異なる                                                                                                  |
| 211   | [Enterprise] -CU オプションの指定の有無が異なる                                                                                     |
| 212   | [Enterprise for Windows] -CM オプションの指定の有無が異なる                                                                         |
| 213   | -CN オプション指定の有無の異なるプログラムが混在している                                                                                       |
| 214   | -Cj オプション指定の有無の異なるプログラムが混在している                                                                                       |
| 215   | [Enterprise] -MT オプション指定の有無の異なるプログラムが混在している                                                                          |

| 例外コード | 原因                                                                                               |
|-------|--------------------------------------------------------------------------------------------------|
| 216   | プログラムの初期化に失敗した                                                                                   |
| 256   | [Standard] COMMIT/ROLLBACK が異常終了した                                                               |
| 257   | [Standard] 外部ファイル不整合がある                                                                          |
| 261   | [Standard] ファイルマップ情報不正がある                                                                        |
| 401   | SORT/MERGE の実行順序が誤っている                                                                           |
| 402   | SORT/MERGE で使用する作業ファイルが確保できない                                                                    |
| 403   | SORT/MERGE で不正長レコードを検出した                                                                         |
| 409   | 関連製品でエラーを検出した                                                                                    |
| 410   | [Enterprise for Windows] [Standard] DLL ロードエラーが発生した<br>[Enterprise for Linux] 共有ライブラリロードエラーが発生した |
| 411   | [Enterprise for Linux] 共有ライブラリから API 関数アドレスが取得できない<br>[Standard] DLL から API 関数アドレスが取得できない        |
| 502   | ファイル入出力エラーが発生した                                                                                  |
| 506   | 標準入力終了した為、小入力あるいは確認入力ができない                                                                       |
| 508   | シェルコマンド実行ルーチン (B_SYSTEM) が異常終了した                                                                 |
| 601   | [Standard] 画面初期化エラーである                                                                           |
| 602   | [Standard] 画面入出力エラーである                                                                           |
| 603   | [Standard] 画面 OPEN 実行順序エラーである                                                                    |
| 604   | [Standard] 画面 CLOSE 実行順序エラーである                                                                   |
| 605   | メモリ不足エラーである                                                                                      |
| 606   | [Standard] 画面 DISPLAY 実行順序不正である                                                                  |
| 607   | [Standard] ウィンドウ表示領域不正である                                                                        |
| 608   | [Standard] 画面 OPEN 制限超過である                                                                       |
| 609   | [Standard] 画面 ACCEPT 入力領域不正である                                                                   |
| 610   | [Standard] 画面 DISPLAY 出力領域不正である                                                                  |
| 611   | [Standard] 画面制御機能は使用できない                                                                         |
| 612   | [Standard] 画面サイズエラーである                                                                           |
| 613   | [Standard] 画面 ACCEPT 実行順序エラーである                                                                  |
| 650   | [Standard] オプション指定が不整合のため小入出力機能は使用できない                                                           |
| 680   | [Standard] 表示フォント不正である                                                                           |
| 699   | [Standard] STNAMR DLL ロードエラーである                                                                  |
| 701   | オブジェクトビューによる適合に失敗した                                                                              |
| 702   | NULL オブジェクト参照を使用してメソッドが呼び出された                                                                    |
| 703   | インスタンスの生成に失敗した                                                                                   |
| 918   | [Standard] 小入出力異常である                                                                             |
| 919   | [Standard] メモリ不足である                                                                              |
| 920   | [Standard] コンソールへの小入出力機能は使用できない                                                                  |
| 980   | [Standard] 小入出力データ不正である                                                                          |
| 982   | [Standard] COBOL エラーである                                                                          |
| 990   | [Standard] インストール環境が XX ではなくプログラムを実行できない                                                         |
| 996   | [Standard] 別プログラムの起動に失敗した                                                                        |
| 999   | メモリ不足でエラーが発生した (COBOL ランタイムの初期化に失敗した)                                                            |

**注意** 注(1) 不正 10 進データが組込み関数内で発生した場合には 094, それ以外の場合には 010 の例外コードが使用される。

## F 翻訳オプションや環境変数による仕様差異

本コンパイラでは、オプションや環境変数の指定有無により、解釈が異なる部分があります。

### F.1 翻訳オプションによる仕様差異

以降に、オプションと言語仕様差異を示します。

| オプション             | 指定なしのとき(既定値)                                                                                                         | 指定ありのとき                                                                                                           |
|-------------------|----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| -CA               | APPLY SHIFT-CODE 句 の WITH PPR-CONTROL-2 指定ありの SYSOUT ファイルに CHARACTER TYPE 句および COLUMN NUMBER 句があってもプリンタ制御コードは埋め込まない。 | APPLY SHIFT-CODE 句 の WITH PPR-CONTROL-2 指定ありの SYSOUT ファイルに CHARACTER TYPE 句および COLUMN NUMBER 句があればプリンタ制御コードを埋め込む。 |
| 【Standard】<br>-Cf | 順編成の大容量記憶ファイルは IFAS FILE 形式である。                                                                                      | 順編成の大容量記憶ファイルは COBOL 独自形式とする。                                                                                     |
| 【Standard】<br>-Cg | 相対編成の大容量記憶ファイルは IFAS FILE 形式である。                                                                                     | 相対編成の大容量記憶ファイルは COBOL 独自形式とする。                                                                                    |
| 【Standard】<br>-Ch | 索引編成の大容量記憶ファイルは IFAS FILE 形式である。                                                                                     | 索引編成の大容量記憶ファイルは ISAM 形式とする。                                                                                       |
| -Ci               | INITIAL 句および CANCEL 文の使用が可能                                                                                          | INITIAL 句および CANCEL 文を使用しないことを前提とした最適なオブジェクトを生成                                                                   |
| -Cj               | プログラムの固有文字集合はシフト JIS                                                                                                 | プログラムの固有文字集合は JIPS(J)                                                                                             |
| -Cr               | 用途が表示用で同桁同符号の数字項目同士であっても、数字転記を行う。                                                                                    | 用途が表示用で同桁同符号の数字項目同士の場合、英数字転記を行う。                                                                                  |
| -Ct               | BINARY 項目へのデータ格納時、PICTURE 句桁数での桁落しを行う。                                                                               | BINARY 項目へのデータ格納時、データが割り付けられた記憶領域に対応した値まで格納し、PICTURE 句桁数での桁落しを行わない。                                               |
| -Cv               | CALL 文の引数の引き渡し方法を指定どおりに解釈する。                                                                                         | 定数指定の CALL 文 の BY VALUE 指定の引数を BY REFERENCE 指定とみなす。                                                               |

| オプション                                         | 指定なしのとき(既定値)                                                                                  | 指定ありのとき                                                                                                                                                                                                                   |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Cw                                           | 英数字転記の作用対象が重なっている場合、処理速度を優先した転記を行う。                                                           | 英数字転記の作用対象が重なっている場合、先頭から1バイト単位で転記を行う。                                                                                                                                                                                     |
| -CN                                           | COMP-3 項目は常に符号領域を持つ。                                                                          | 偶数桁で PICTURE 文字列に符号 S がない COMP-3 項目を符号領域なし COMP-3 項目として動作させる。                                                                                                                                                             |
| 【Enterprise】<br>-CP                           | PPR-CONTROL-3 指定の SYSOUT ファイルの印刷データに、KLP 用のプリンタ制御コードを埋め込まない。                                  | PPR-CONTROL-3 指定の SYSOUT ファイルの印刷データに、KLP 用のプリンタ制御コードを埋め込む。                                                                                                                                                                |
| -CS                                           | 数字と符号の重ね合わせが行われた外部 10 進数の符号桁の値は、本 COBOL 固有の表現形式とする。                                           | 数字と符号の重ね合わせが行われた外部 10 進数の符号桁の値を、Pro*COBOL の表現形式に変更する。                                                                                                                                                                     |
| 【Enterprise】<br>-CU                           | <ul style="list-style-type: none"> <li>プログラムの固有文字集合はシフト JIS</li> <li>CODE-SET 句はメモ</li> </ul> | <ul style="list-style-type: none"> <li>プログラムの固有文字集合は Unicode<br/>英数字項目 UTF-8<br/>日本語項目 UCS-2</li> <li>ALPHABET 句に UTF-8, UCS-2 が記述できる。</li> <li>CODE-SET 句が有効</li> <li>16 進日本語定数 NU (16 進数を UCS-2 として扱う) が指定可能</li> </ul> |
| -CX                                           | ON SIZE ERROR 句なしのゼロ除算結果は規定しない。                                                               | ON SIZE ERROR 句なしのゼロ除算結果を 0 とする。                                                                                                                                                                                          |
| 【Enterprise for Windows】<br>【Standard】<br>-CY | 実行可能プログラムに静的リンクした外部プログラムを一意名呼び出しできない。                                                         | 実行可能プログラムに静的リンクした外部プログラムを一意名呼び出しできる。                                                                                                                                                                                      |
| -Cy                                           | プリンタファイルの APPLY SHIFT-CODE 句は有効である。                                                           | プリンタファイルの APPLY SHIFT-CODE 句はメモ扱いとする。                                                                                                                                                                                     |
| -C5                                           | COMP 項目の内部表現形式を BINARY とする。                                                                   | COMP 項目の内部表現形式を COMP-5 とする。                                                                                                                                                                                               |
| 【Enterprise for Linux】<br>-Cz                 | 位置独立コード(Position Independent Code)を作成しない。                                                     | 位置独立コード(Position Independent Code)を作成する。                                                                                                                                                                                  |



| オプション                                                       | 指定なしのとき(既定値)                                                                                                                           | 指定ありのとき                                                                                                                                                                                                                    |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br><b>【Enterprise for Windows】</b><br>-CF | プログラムの固有文字集合(内部コード)はシフト JIS である。                                                                                                       | プログラムの固有文字集合(内部コード)は JIS2004 対応の JIPS(J)とする。                                                                                                                                                                               |
| <b>【Enterprise】</b><br>-Fa                                  | データ項目の大きさとアラインについて、プラットフォームの仕様に従う。<br>・1～2 桁の 2 進数を 1 バイトの連続した 8 ビットにする。<br>・2 進数に SYNC 指定がある場合、1-2 桁は 1 バイト境界、10-18 桁は 8 バイト境界に桁づめする。 | データ項目の大きさとアラインを ACOS-4 互換とする。<br>・1～2 桁の 2 進数を 2 バイトの連続した 16 ビットにする。<br>・2 進数に SYNC 指定がある場合、1-2 桁は 2 バイト境界、10-18 桁は 4 バイト境界に桁づめする。                                                                                         |
| <b>【Enterprise】</b><br>-Fb                                  | データポインタデータ項目の割り付け境界について、プラットフォームの仕様に従う。<br>・SYNC 句の指定ができない。<br>・常に、4 バイト境界に割り付ける。                                                      | データポインタデータ項目の割り付け境界を ACOS-4 互換とする。<br>・SYNC 句の指定ができる。<br>・SYNC 指定なしの場合、1 バイト境界に割り付ける。<br>・SYNC 指定ありの場合、4 バイト境界に割り付ける。                                                                                                      |
| <b>【Enterprise】</b><br>-Fc                                  | COMP 項目の内部表現形式を 2 進数とする。                                                                                                               | COMP 項目の内部表現形式を ACOS-4 互換の外部 10 進数とする。                                                                                                                                                                                     |
| <b>【Enterprise】</b><br>-Fd                                  | 比較規則について、プラットフォームの仕様に従う。<br>・非整数外部 10 進数と ZERO 以外の表意定数は比較できない。<br>・算術式と一意名(英数字/英数字編集)は比較できない。<br>・シフト JIS コード体系に基づく、日本語の大小比較を行う。       | 比較規則を ACOS-4 互換とする。<br>・非整数外部 10 進数と ZERO 以外の表意定数比較を表意定数比較とする。<br>・算術式と一意名(英数字/英数字編集)の比較を文字比較とする。<br>・JIPS(J) コード体系に基づく、日本語の大小比較を行う。<br>・非整数外部 10 進数と英数字定数の比較を文字比較とする<br>・指標名と算術式の比較を、算術結果の整数部と指標名の出現番号との I 比較(出現番号の比較)とする |

| オプション                           | 指定なしのとき(既定値)                                                                                                                  | 指定ありのとき                                                                                                                                                       |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 【Enterprise】<br>-Fe             | NUMERIC 検査について、プラットフォームの仕様に従う。<br>・偶数桁の内部 10 進数の最左端上位 4 ビットが 0x0 以外の場合、不正 10 進エラーとする。<br>・外部 10 進数の独立符号部が空白の場合、不正 10 進エラーとする。 | NUMERIC 検査を ACOS-4 互換とする。<br>・偶数桁の内部 10 進数の最左端上位 4 ビットが 0x0 以外の場合、偽とする(-CN オプションが指定されていない時)。不正 10 進エラーにはならない。<br>・外部 10 進数の独立符号部が空白の場合、真とする。不正 10 進エラーにはならない。 |
| 【Enterprise】<br>-Fj             | 小入出力先の既定値について、プラットフォームの仕様に従う。<br>ACCEPT は STDIN から入力、DISPLAY は STDOUT へ出力する。                                                  | 小入出力先の既定値を ACOS-4 互換とする。ACCEPT は SYSIN から入力、DISPLAY は SYSOUT へ出力する。                                                                                           |
| 【Enterprise】<br>-Fl             | STOP 文の ABORT 指定は使用できない。                                                                                                      | STOP 文の ABORT 指定を可能とする。<br>・異常終了時の終了値を環境変数から取得する。                                                                                                             |
| 【Enterprise】<br>-Fr             | コンパイラの間接結果格納領域の生成規則に従って、作用対象毎に自動的に桁数を決定する。                                                                                    | COMPUTE 文および条件式の間接結果格納領域を指定した小数部桁数に基づく固定値とする。                                                                                                                 |
| 【Enterprise】<br>-Fs             | 英数字(英数字項目、英数字定数、ALL 英数字定数)から外部 10 進数への数字転記において、ゾーン部分のビットの補正を行う。                                                               | 英数字(英数字項目、英数字定数、ALL 英数字定数)から外部 10 進数への数字転記において、ゾーン部分のビットの補正を行わない。                                                                                             |
| 【Enterprise】<br>-Ft             | 符号なし外部 10 進数の参照時に最右端バイトを数字と符号の重ね合わせの文字として処理しない。                                                                               | 符号なし外部 10 進数の参照時に最右端バイトを数字と符号の重ね合わせの文字として処理する。                                                                                                                |
| 【Enterprise for Windows】<br>-Fv | SYSOUT ファイルに対する WRITE 文の ADVANCING 句に機能名 LN001~LN255 に対応する呼び名を指定した場合、ADVANCING 句が指定されていないものとして動作する。                           | SYSOUT ファイルに対する WRITE 文の ADVANCING 句に機能名 LN001~LN255 に対応する呼び名を指定することで、SYSOUT ファイルの改行位置の絶対行指定を可能とする。                                                           |

| オプション                           | 指定なしのとき(既定値)                                                                                                                                                                               | 指定ありのとき                                                                                                                                                                                      |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 【Enterprise for Windows】<br>-Fw | COMPUTE 文の受け取り側項目に<br>ROUNDED が指定された場合、その受け取り側項目の小数部桁数が小数部桁数の最大値(DMAX)か否かに関わらず、小数部桁数の最大値(DMAX)に 1 を加える。                                                                                    | COMPUTE 文の受け取り側項目に<br>ROUNDED が指定され、かつその受け取り側項目の小数部桁数が小数部桁数の最大値(DMAX)である場合、小数部桁数の最大値(DMAX)に 1 を加える。                                                                                          |
| 【Enterprise】<br>-Fx             | 符号なし外部 10 進数の参照時に最右端バイトを数字と符号の重ね合わせの文字として処理しない。                                                                                                                                            | MOVE 文の受け取り側が英数字項目または日本語項目である場合、送り出し側の符号なし外部 10 進数の最右端バイトを数字と符号の重ね合わせの文字として処理しない。<br><br>MOVE 文の受け取り側が英数字項目および日本語項目でない場合、または MOVE 文以外の手続き文の場合、符号なし外部 10 進数の参照時に最右端バイトを数字と符号の重ね合わせの文字として処理する。 |
| 【Enterprise for Windows】<br>-Fy | PPR-CONTROL-1 指定の SYSOUT ファイルの印刷データに、シフトコードおよびプリンタ制御コードを埋め込まない。                                                                                                                            | PPR-CONTROL-1 指定の SYSOUT ファイルの印刷データに、シフトコードおよびプリンタ制御コードを埋め込む。                                                                                                                                |
| 【Enterprise for Windows】<br>-Fz | データ例外が発生する手続き文を<br>ACOS-4 互換としない。<br>次の手続き文について、不正 10 進チェックを行わない。<br>(1) DISPLAY 文の作用対象が数字項目のとき<br>(2) COMBINE 文の SHIFTED 句の作用対象が外部 10 進項目のとき<br>(3) SORT 文または MERGE 文の KEY 句の作用対象が数字項目のとき | データ例外が発生する手続き文を<br>ACOS-4 互換とする。<br>次の手続き文について、不正 10 進チェックを行う。<br>(1) DISPLAY 文の作用対象が数字項目のとき<br>(2) COMBINE 文の SHIFTED 句の作用対象が外部 10 進項目のとき<br>(3) SORT 文または MERGE 文の KEY 句の作用対象が数字項目のとき      |
| 【Enterprise for Windows】<br>-FA | DIVIDE 文の剰余結果を ACOS-4 互換としない。<br>剰余は商の定義と符号有無が同じ補助的な商を用いて計算する。                                                                                                                             | DIVIDE 文の剰余結果を ACOS-4 互換とする。<br>剰余は符号付きの補助的な商を用いて計算する。                                                                                                                                       |

| オプション             | 指定なしのとき(既定値)                                                                                                                                                            | 指定ありのとき                                                                                                                                                                          |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Qa               | 連絡節に GLOBAL 句は記述不可。                                                                                                                                                     | 連絡節に GLOBAL 句の記述を可能とする。                                                                                                                                                          |
| -Qb               | ベースデータ節に GLOBAL 句は記述不可。                                                                                                                                                 | ベースデータ節に GLOBAL 句の記述を可能とする。                                                                                                                                                      |
| -Qc               | 指標名や指標データ項目は連続した 16 バイトでバイト境界から割り当てられ、最初の 8 バイトは出現番号が次の 8 バイトに指標値が COMP-5 PIC S9(18) 形式で入る。                                                                             | 指標名や指標データ項目は連続した 8 バイトでバイト境界から割り当てられ、最初の 4 バイトは出現番号が次の 4 バイトに指標値が倍精度固定 2 進数形式で入る。                                                                                                |
| 【Standard】<br>-Qd | 可変繰り返し回数を評価して COMBINE 文を実行する。                                                                                                                                           | 可変繰り返し回数が最大として COMBINE 文を実行する。                                                                                                                                                   |
| -Qe               | 行順ファイルのレコード右端の空白を削除せずに書き出す。                                                                                                                                             | 行順ファイルのレコード右端の空白を削除して書き出す。                                                                                                                                                       |
| 【Standard】<br>-Qg | LINAGE 句の論理ページサイズの評価タイミングを変更しない。                                                                                                                                        | LINAGE 句の論理ページサイズの評価タイミングを変更する。                                                                                                                                                  |
| 【Standard】<br>-Qh | NUMERIC 検査の結果を変更しない。                                                                                                                                                    | NUMERIC 検査の結果を変更する。                                                                                                                                                              |
| 【Standard】<br>-Qi | SYSIN ファイルの入力終了条件を指定しない。                                                                                                                                                | SYSIN ファイルの入力終了条件を指定する。                                                                                                                                                          |
| 【Standard】<br>-Qj | UNSTRING 文の P 付き整数項目の計数結果を変更しない。                                                                                                                                        | UNSTRING 文の P 付き整数項目の計数結果を変更する。                                                                                                                                                  |
| 【Standard】<br>-Ql | CLOSE 文の WITH LOCK 句をコメントとしない。                                                                                                                                          | CLOSE 文の WITH LOCK 句をコメントとする。                                                                                                                                                    |
| -Qm               | プログラム名と同じ名前の利用者定義語を定義できない。                                                                                                                                              | プログラム名と同じ名前の利用者定義語を定義できる。                                                                                                                                                        |
| -Qo               | <ul style="list-style-type: none"> <li>•OPEN 文の永続誤り(入出力状態値 35)後、OUTPUT 指定の OPEN 文を続けて実行できない。</li> <li>•COBOL 独自相対ファイルおよび ISAM 索引ファイルのファイル名不正の入出力状態値は 30 である。</li> </ul> | <ul style="list-style-type: none"> <li>•OPEN 文の永続誤り(入出力状態値 35)後、OUTPUT 指定の OPEN 文を続けて実行できる。</li> <li>•【Standard】COBOL 独自相対ファイルおよび ISAM 索引ファイルのファイル名不正の入出力状態値は 99 である。</li> </ul> |
| -Qp               | <ul style="list-style-type: none"> <li>•SHARED-MODE または PROTECTED-MODE 指定の CSV 指定なし行順ファイルに対して、ファイル共用機能は使用できない。</li> </ul>                                               | <ul style="list-style-type: none"> <li>•SHARED-MODE または PROTECTED-MODE 指定の CSV 指定なし行順ファイルに対して、ファイル共用機能を使用可能とする。</li> </ul>                                                       |

| オプション                         | 指定なしのとき(既定値)                         | 指定ありのとき                                                                                                                                                                                                                                                                                                       |
|-------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 【Enterprise for Linux】<br>-Qq | ・CSV 指定あり行順ファイルに対して、ファイル共用機能を使用できない。 | ・CSV 指定あり行順ファイルに対して、ファイル共用機能を使用できる。                                                                                                                                                                                                                                                                           |
| -vi                           | VALUE 句が指定されていないデータ項目の初期値を保証しない。     | 初期値のない項目はすべて実行コード系に依存した” 0” で表される値を初期値として与える。                                                                                                                                                                                                                                                                 |
| -vs                           | VALUE 句が指定されていないデータ項目の初期値を保証しない。     | データ項目が英字、英数字、英数字編集、数字編集の時には、VALUE IS SPACE が記述されたかのように初期値を与える。他の項目については次のとおりである。<br>(1) データ項目の項類が数字のとき<br>VALUE IS ZERO が指定されたかのように初期値を与える。<br>(2) データポインタデータ項目のとき<br>NULL (空ポインタ)を初期値として与える。<br>(3) データ項目の項類が日本語および日本語編集項目のとき<br>実行コード系に依存した日本語文字の空白を初期値として与える。<br>(4) ブールデータ項目のとき<br>ブール文字の” 0” を初期値として与える。 |
| -vz                           | VALUE 句が指定されていないデータ項目の初期値を保証しない。     | データ項目の項類が英字、英数字、英数字編集、数字編集のときは、VALUE IS ZERO が指定されたかのように初期値を与える。他の項目については-vs 指定のときと同様である。                                                                                                                                                                                                                     |
| -vnn                          | VALUE 句が指定されていないデータ項目の初期値を保証しない。     | データ項目の項類が英字、英数字、英数字編集、数字編集のときは、VALUE IS ALL ” ” nn” ” が指定されたかのように初期値を与える。他の項目については-vs 指定の時と同様である(n は 16 進数文字[0-9, a-f, A-F]の 1 文字を表す)。                                                                                                                                                                        |

| オプション                                                             | 指定なしのとき(既定値)                                                                | 指定ありのとき                                                                                                                                                   |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-w</b> <i>c</i>                                                | 本コンパイラ既定の予約語を使用する。<br>( <b>-w0</b> が指定されたとみなす。)                             | <b>c</b> には[012]のうち 1 つを指定する。<br><b>0</b> : 本コンパイラ既定の予約語を使用する。<br><b>1</b> : COBOL85 MFA の予約語を使用する。<br><b>2</b> : A-VX COBOL85 および COBOL85 Pro の予約語を使用する。 |
| <b>-Xi</b>                                                        | <b>INITIALIZE</b> の展開順序を 2002 規格仕様とする。                                      | <b>INITIALIZE</b> の展開順序を 85 規格仕様とする。                                                                                                                      |
| <b>【Enterprise】</b><br><b>-MT</b>                                 | 複数スレッドで動作可能なスレッドセーフコードを生成しない。                                               | 複数スレッドで動作可能なスレッドセーフコードを生成する。                                                                                                                              |
| <b>【Standard】</b><br><b>-Mw</b>                                   | バッチ型アプリケーションを作成する。                                                          | 対話型アプリケーションを作成する。                                                                                                                                         |
| <b>【Standard】</b><br><b>【Enterprise for Windows】</b><br><b>-q</b> | コンパイルを行うとき、マニフェストファイルを生成しない。                                                | コンパイルを行うとき、マニフェストファイルを <b>COBOL</b> アプリケーションファイルに書き込まずに単独のファイルとして生成する。                                                                                    |
| <b>【Standard】</b><br><b>【Enterprise for Windows】</b><br><b>-p</b> | コンパイルを行うとき、マニフェストファイルを生成しない。                                                | コンパイルを行うとき、マニフェストファイルを <b>COBOL</b> アプリケーションファイルに書き込み、単独のファイルとして生成しない。                                                                                    |
| <b>【Standard】</b><br><b>-Yf</b> <i>lstfile</i>                    | 原始プログラムファイルに対応する翻訳リスト(ファイルのベース名に <b>.list</b> を付けたもの)を生成する。                 | <b>lstfile.list</b> という名前で翻訳リストを生成する。                                                                                                                     |
| <b>【Standard】</b><br><b>-Yi</b> <i>iconfile</i>                   | 既定値のアイコンを使用する。                                                              | <b>iconfile</b> という名前のアイコンを使用する。                                                                                                                          |
| <b>【Standard】</b><br><b>-Yo</b> <i>objfile</i>                    | すべての原始プログラムファイルに対応するオブジェクトファイル(各原始プログラムファイルのベース名に <b>.obj</b> を付けたもの)を生成する。 | <b>objfile.obj</b> という名前でオブジェクトファイルを生成する。                                                                                                                 |
| <b>【Standard】</b><br><b>-Yr</b> <i>registry</i>                   | レジストリ機能を利用しない。                                                              | レジストリ機能を利用して、アプリケーションごとまたは複数のアプリケーション共通に、プログラムに対する各種設定を行う。                                                                                                |

| オプション                           | 指定なしのとき(既定値)                                | 指定ありのとき                                     |
|---------------------------------|---------------------------------------------|---------------------------------------------|
| 【Standard】<br>-P1               | 画面制御機能 ACCEPT/DISPLAY 文の<br>ROLLING 句は使用可能。 | 画面制御機能 ACCEPT/DISPLAY 文の<br>ROLLING 句は使用不可。 |
| 【Standard】<br>-P2               | 画面制御機能 ACCEPT/DISPLAY 文の<br>ROLLING 句は使用可能。 | 画面制御機能 ACCEPT/DISPLAY 文の<br>ROLLING 句は使用不可。 |
| 【Enterprise<br>for Linux】<br>-T | 用語機能は使用できない。                                | 用語機能は使用できる。                                 |

F.2 環境変数による仕様差異

以降に、環境変数と言語仕様差異を示します。

| 環境変数                                       | 機能                                                                                                                                                                                                                                                         |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB_FILETAB<br>= n (1≦n≦16)                | 行順ファイルの入力において、TAB コードをいくつかの空白に拡張し、n を法とする境界に合わせる。                                                                                                                                                                                                          |
| COB_SYSOUT_FORMAT<br>= { SSF   PRN   TXT } | DISPLAY 文(データ例外表示)を除く SYSOUT のレコード形式を変更する。<br>SSF 指定<br>SSF ヘッダ(制御データ)付きの SYSOUT ファイル既定のレコード形式で出力する。<br>PRN 指定<br>PRN プリンタファイルのレコード形式で出力する。<br>TXT 指定<br>テキストファイルのレコード形式で出力する。右端の空白は削除する。<br>上記以外または指定なし<br>【Enterprise】SSF 指定となる。<br>【Standard】TXT 指定となる。 |



| 環境変数                                                                  | 機能                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Enterprise】</b><br>COB_SYSOUT_CODE<br>= { UTF-8   UCS-2   S-JIS } | 小出力 <b>DISPLAY</b> で使用する <b>SYSOUT</b> ファイルの文字集合を指定する。<br>(-CU オプション指定で生成されたアプリケーションにのみ有効)<br><b>UTF-8 指定</b><br>プログラムの固有文字集合から <b>UTF-8</b> に変換して出力する。<br><b>UCS-2 指定</b><br>プログラムの固有文字集合から <b>UTF-16LE</b> に変換して出力する。<br><b>S-JIS 指定</b><br>プログラムの固有文字集合からシフト <b>JIS</b> に変換して出力する。<br>上記以外または指定なし<br>コード変換せず、そのまま出力される。 |
| COB_RUNERR<br>= { NO_PAUSE   NO_DISPLAY }                             | <b>NO_PAUSE 指定</b><br>実行時エラーメッセージの一時停止を行わない。連続実行に便利である。 <b>WS-EML</b> 配下でないバッチ型アプリケーションでのみ有効。<br><b>NO_DISPLAY 指定</b><br>実行時エラーメッセージを表示しない。<br>未定義または <b>NO_PAUSE</b> , <b>NO_DISPLAY</b> 以外を指定時<br>実行時エラーメッセージを出力後、確認キーの入力待ちとなり、処理を一時停止する。                                                                                |
| COB_ABTASK<br>= n (1~255)                                             | 1~255 の異常終了値を設定する。                                                                                                                                                                                                                                                                                                         |

| 環境変数                                                        | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Enterprise】</b><br>COB_CODEERR<br>= { STOP   CONTINUE } | <p>–CU オプション指定時, 標準出力／標準エラー出力への出力時, コード変換が失敗した場合の動作を指定する。</p> <p><b>【Enterprise for Linux】環境変数</b><br/>           LANG=ja_JP.UTF-8 指定時かつ環境変数 COB_CODECHG=ON 指定時, 標準出力／標準エラー出力／CONSOLE への出力時, コード変換が失敗した場合の動作を指定する。</p> <p><b>STOP 指定 (デフォルト)</b><br/>           実行時エラーメッセージを表示し, 終了する。</p> <p><b>CONTINUE 指定</b><br/>           –CU オプション指定時, 変換に失敗した文字を 00<sub>16</sub> に置き換えて処理を継続する。</p> <p><b>【Enterprise for Linux】環境変数</b><br/>           LANG=ja_JP.UTF-8 指定時かつ環境変数 COB_CODECHG=ON 指定時, 変換に失敗した文字を 00<sub>16</sub> に置き換えて処理を継続する。</p> |
| COB_GCMOVE<br>= { ON   OFF }                                | <p>英数字項目から日本語項目への転記で, 英数字項目に濁点が含まれる場合, カタカナの濁点を結合する。</p> <p>(–CU オプション指定なしで生成されたアプリケーションにのみ有効)</p> <p><b>ON 指定</b><br/>           カタカナの濁点を結合する。</p> <p><b>OFF 指定</b><br/>           カタカナの濁点を結合しない。</p>                                                                                                                                                                                                                                                                                                                                       |

| 環境変数                                                                        | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br><b>COB_FILE_SHARING</b><br><b>= { WAIT   NO_WAIT }</b> | <p>ファイルの排他制御時の動作を指定する。</p> <ul style="list-style-type: none"> <li>・SYSOUT/SYSIN ファイルの場合<br/>環境変数 <b>COB_FILE_SHARING</b> の指定は無効である。</li> <li>・大容量記憶ファイル/RDB ファイルの場合<br/><b>WAIT</b> 指定<br/>他のプログラムで排他制御を行っているファイルに対して <b>OPEN</b> 文を実行した場合、<b>OPEN</b> 文は保留される。</li> <li><b>NO_WAIT</b> 指定<br/>他のプログラムで排他制御を行っているファイルに対して <b>OPEN</b> 文を実行した場合、<b>FILE STATUS</b> 句がある場合は、入出力状態“91”を返却、<b>FILE STATUS</b> 句がない場合は、実行時メッセージを出力しプログラムが異常終了する。</li> </ul> <p>上記以外または指定なし<br/>コンパイラ動作既定値に従い、<b>WAIT</b> が指定されたものとみなす。</p> <ul style="list-style-type: none"> <li>・プリンタファイルの場合<br/>環境変数 <b>COB_FILE_SHARING</b> の有無にかかわらず、次の様な動作となる。</li> </ul> <p>プリントマネージャ経由の場合<br/>排他制御は行わない。</p> <p><b>PRNAMR</b> 経由のプリンタの場合<br/>他のプログラムで使用中のプリンタに対して <b>OPEN</b> 文を実行した場合、<b>FILE STATUS</b> 句がある場合は、入出力状態“91”を返却、<b>FILE STATUS</b> 句がない場合は、実行時メッセージを出力し、解除キー待ちになる。</p> |

| 環境変数                                                  | 機能                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br>COB_RDB_STATUS<br>= { ON   OFF } | 特殊レジスタ RDB-STATUS に COMMIT / ROLLBACK 文の実行結果の通知方法を指定する。<br><b>ON 指定</b><br>COMMIT または ROLLBACK 文の実行結果は、常に特殊レジスタ RDB-STATUS に通知される。<br><b>OFF 指定または指定なし</b><br>COMMIT または ROLLBACK 文の実行結果が異常終了したとき、実行時メッセージを出力の上でプログラムが異常終了する。<br>RDB-STATUS に返却される値については、各データベースエンジンのマニュアルを参照。 |
| <b>【Standard】</b><br>COB_PRNTHR<br>= ON               | プリンタ機能の PASSTHROUGH モードの使用を設定する。PASSTHROUGH モードを使用することにより、プリンタシーケンスの発行が可能となる。<br>PASSTHROUGH モード機能を使用する場合に設定値が ON ではない場合、環境変数の指定がないものとして動作する。また、レジストリと環境変数を組み合わせたときの動作は付録 D を参照。                                                                                               |
| <b>【Standard】</b><br>COB_AMRMODE<br>= { ON   OFF }    | PRNAMR 経由での印刷はプリントマネージャ経由の印刷かを指定する。<br>COBOL アプリケーション実行環境がインストールされている環境の場合<br>レジストリと環境変数を組み合わせたときの動作は付録 D を参照。<br>COBOL アプリケーション実行環境がインストールされていない環境の場合<br>本環境変数は無効である。<br>ON/OFF 以外の値を指定した場合<br>指定なしとみなす。                                                                      |

| 環境変数                                                         | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br><b>COB_PRNFONT</b><br>=フォント名[, フォントサイズ] | <p>プリントマネージャ経由の印刷において使用する文字フォントを指定する。</p> <p>フォント名, フォントサイズを指定した場合</p> <p>プリントマネージャ経由の印刷において, 指定したフォント名, フォントサイズが使用される。</p> <p>フォント名がない場合やフォント名が誤っている場合</p> <p>指定がないものとして動作し, システムで定義されているフォントが使用される。ただし, フォント名は1字でも間違いがあると, フォント名とは認識されない。</p> <p>フォントサイズが省略された場合または指定が誤っている場合</p> <p>10 ポイントとする。</p> <p>フォントサイズにフォントに対応されていない数値を指定した場合</p> <p>近いサイズを自動的に選択する。</p> <p>プリントマネージャ経由の印刷において使用する文字フォントは, アプリケーションカスタマイザによっても指定可能となる。アプリケーションカスタマイザと環境変数を同時に指定した場合, アプリケーションカスタマイザでの指定が優先とする。</p> |

| 環境変数                                                                              | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br><b>COB_EVENTLOG</b><br>= { ON   WSEML   NON_WSEML<br>  ALL } | <p>実行時エラーとなった場合の動作を指定する。<br/>           環境変数 COB_RUNERR により動作が決まる。</p> <p>環境変数 COB_RUNERR に NO_DISPLAY 指定時、<br/>           COB_EVENTLOG は無効である。</p> <p>環境変数 COB_RUNERR に NO_DISPLAY 未指定時、<br/>           ON 指定<br/>           実行時エラーメッセージをイベントログとして出力し、確認キーの入力待ちを行わず、処理を終了する。</p> <p>WSEML 指定<br/>           実行時エラーメッセージをイベントログとして出力するとともに、実行時エラーメッセージを COBOL WS エミュレータに出力後、確認キーの入力待ちとなり、処理を一時停止する。</p> <p>指定値が NON_WSEML の場合<br/>           実行時エラーメッセージをイベントログとして出力するとともに環境変数 COB_RUNERR の指定に従った動作を行う。<br/>           (COBOL WS エミュレータ配下でないアプリケーションにのみ有効)</p> <p>指定値が ALL の場合<br/>           COBOL WS エミュレータ配下のアプリケーションの場合、WSEML を指定したのと同じように動作する。<br/>           COBOL WS エミュレータ配下でないアプリケーションの場合、NON_WSEML を指定したのと同じように動作する。</p> <p>上記以外または指定なし<br/>           環境変数 COB_RUNERR が未定義の場合、実行時エラーメッセージを出力後、確認キーの入力待ちとなり、処理を一時停止する。<br/>           環境変数 COB_RUNERR に NO_PAUSE 指定時、実行時エラーメッセージを出力後、確認キーの入力待ちを行わず、処理を続行する。</p> |

| 環境変数                                         | 機能                                                                                                                                                                                                                                                                      |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br>COB_WSJISNOTCHG<br>= ON | <b>ON 指定</b><br>COBOL WS エミュレータ使用時に行われるデータのシフト JIS と JIPS コード変換において、JIS78/90 文字の入替処理を抑止する。<br>上記以外または指定なし<br>JIS78/90 文字の入替処理を行う。                                                                                                                                       |
| <b>【Standard】</b><br>COB_KANAIME<br>= ON     | アプリケーションカスタマイザの入力モードの設定がカタカナ(既定値)の場合、INPUT MODE IS KANA-SHIFT 句が指定された項目にcaretが位置付いた際に、日本語入力システムを起動し、半角カタカナ文字を入力できる状態にする。<br>本環境変数は COBOL WS エミュレータ配下でない対話型アプリケーションで有効である。<br><b>ON 指定</b><br>日本語入力システムを起動する。入力モード(カナ入力/ローマ字入力)は IME の設定に従う。<br>上記以外または指定なし<br>動作に影響しない。 |
| <b>【Standard】</b><br>COB_CID_INIFILE         | 内部ファイル ID 置換表のファイル名をフルパスで指定する。                                                                                                                                                                                                                                          |
| <b>【Standard】</b><br>COB_GDDMAX = ON         | <b>STOP</b> 定数文および機能名 GDD に対応づけられた呼び名への DISPLAY 文の表示可能な文字数を拡張する。ただし、メッセージ出力中の画面ハードコピー機能は使用不可となる。<br>本環境変数は COBOL WS エミュレータ配下でない対話型アプリケーションで有効である。<br><b>ON 指定</b><br>表示可能な文字数は 236 文字<br>未定義または ON 以外を指定<br>表示可能な文字数は 59 文字                                             |

| 環境変数                                                                            | 機能                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Standard】</b><br><b>【Enterprise for Windows】</b><br>COB_SYSOUT_NEWLINE = ON | DISPLAY 文(データ例外表示)を除く SYSOUT ファイルの改行コードを指定する。<br>ON を指定した場合<br>SYSOUT ファイルでは改行コード(0x0D0A)を出力する。<br>未定義または ON 以外の指定ありの場合<br>SYSOUT ファイルでは改行コード(0x0A)を出力する。<br><b>【Enterprise for Windows】</b><br>下記のすべての条件を満たす場合、改行コードを 0x0D000A00 に変更する。<br>・標準動作モードで、-CU ありかつ CODE-SET 句の符号系名で参照される文字集合が UCS-2。<br>・環境変数 COB_SYSOUT_NEWLINE に ON を指定する。 |
| <b>【Enterprise】</b><br>COB_FILE_APPEND = ON                                     | 大容量記憶ファイルの順/行順/CSV ファイルの OUTPUT オープンを EXTEND オープンで実行する。<br>ON を指定した場合<br>OUTPUT オープンを EXTEND オープンで実行する。EXTEND オープンの切り換えは最初の 1 回目のみである。1 度 CLOSE して再度 OPEN するとき(2 回目以降)、OPEN 文の記述どおり実行する。<br>未定義または ON 以外の指定ありの場合<br>OPEN 文の記述どおり実行する。                                                                                                    |



| 環境変数                                                                  | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB_SYSIN_READ = ON                                                   | <p><b>SYSIN</b> ファイルから <b>READ</b> 文で 80 桁を超えるレコードを読み込めるようにすることを指定する。</p> <p><b>ON</b> を指定した場合、<br/>最大でレコード記述項で指定した大きさ分ファイルから読み込む。</p> <p>未定義または <b>ON</b> 以外の指定ありの場合、<br/>最大で 80 桁分ファイルから読み込む。</p> <p>いずれの場合も、ファイルから読み込んだレコードの大きさがレコード記述項で指定した大きさに満たない場合は、残りのレコード領域には半角空白が埋められる。</p> <p>また、ファイルから読み込んだレコードの大きさが読み込み可能な最大の大きさを超える場合は、レコードは右側が切り捨てられ最大の大きさになる。</p>                                                                                                          |
| <p>【Enterprise for Windows】<br/>【Standard】<br/>COB_SYSIO_NUL = ON</p> | <p><b>SYSIN/SYSOUT</b> ファイルのファイル名にデバイス名 <b>NUL</b> を指定することを可能とする。</p> <p>・<b>READ</b> 文によるファイル名が <b>NUL</b> の <b>SYSIN</b> ファイル読み込み時<br/><b>ON</b> を指定した場合、<br/><b>AT END</b> になる。</p> <p>未定義または <b>ON</b> 以外の指定ありの場合、<br/>「COB101 ファイルが開けません」実行時エラーとなる。</p> <p>・<b>WRITE</b> 文によるファイル名が <b>NUL</b> の <b>EXTEND</b> 指定 <b>SYSOUT</b> ファイルへの書き込み時<br/><b>ON</b> を指定した場合、<br/>エラーは発生せず、ファイル出力もしない。</p> <p>未定義または <b>ON</b> 以外の指定ありの場合、<br/>「COB101 ファイルが開けません」実行時エラーとなる。</p> |

| 環境変数              | 機能                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB_TRACE = ファイル名 | <p>トレース情報の出力を指定する。</p> <p>ファイル名 (フルパス付き) 指定</p> <p>指定ディレクトリにトレース情報を出力する。</p> <p>指定ファイル名が <b>abc.csv</b> の場合, トレース情報ファイル名は次のとおり。</p> <p>「<b>abc_日付_時刻_プロセス ID_スレッド ID.csv</b>」</p> <p>※日付 (YYYYMMDD) は西暦年月日</p> <p>※時刻 (HHMMSSXX) は時分秒 1/100 秒</p> <p>※シングルスレッドアプリケーションでもスレッド ID を持っているので, そのままスレッド ID を使用する。</p> <p>上記以外または未定義</p> <p>トレース情報を出力しない。</p> |

| 環境変数                                   | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB_TRACE_LEVEL = { 0   1<br>  2   3 } | <p>トレース情報の出力レベルを指定する。</p> <p><b>0 指定</b><br/>実行トレース(カバレッジ情報を含まない)のみ出力する。</p> <p><b>1 指定</b><br/>実行トレース(カバレッジ情報を含まない)とレコードダンプ(レコードダンプ, キー情報(位置, 長さ, ダンプ), 副キー情報(個数, 位置, 長さ, ダンプ))を出力する。</p> <p><b>2 指定</b><br/>COBOL Assistant Option インストール済み, かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合, 実行トレース(カバレッジ情報を含む)のみ出力する。<br/>その他の場合, 実行トレース(カバレッジ情報を含まない)のみ出力する。</p> <p><b>3 指定</b><br/>COBOL Assistant Option インストール済み, かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合, 実行トレース(カバレッジ情報を含む)とレコードダンプ(レコードダンプ, キー情報(位置, 長さ, ダンプ), 副キー情報(個数, 位置, 長さ, ダンプ))を出力する。<br/>その他の場合, 実行トレース(カバレッジ情報を含まない)とレコードダンプ(レコードダンプ, キー情報(位置, 長さ, ダンプ), 副キー情報(個数, 位置, 長さ, ダンプ))を出力する。</p> <p>上記以外または未定義<br/>COB_TRACE=ファイル名指定ありのとき, 実行トレース(カバレッジ情報を含まない)のみ出力する。<br/>COB_TRACE=ファイル名指定なしのとき, トレース情報は出力しない。</p> |

| 環境変数                    | 機能                                                                                                                                                                                                                                                                                                    |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COB_TRACE_LINE          | <p>ソース単位名と行番号指定の部分情報出力を指定する。COB_TRACE_LABEL と同時指定できる。</p> <p>ファイル名 (フルパス付き) 指定</p> <p>COBOL Assistant Option インストール済み、かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合、ソース単位名と行番号指定の部分情報出力指定ファイル名をフルパスで指定すると、指定した範囲の実行トレースとレコードダンプを出力する。</p> <p>その他の場合、部分情報出力は無効である。</p> <p>上記以外または未定義</p> <p>部分情報を出力しない。</p>  |
| COB_TRACE_LABEL         | <p>ソース単位名と手続き名指定の部分情報出力を指定する。COB_TRACE_LINE と同時指定できる。</p> <p>ファイル名 (フルパス付き) 指定</p> <p>COBOL Assistant Option インストール済み、かつカバレッジ情報と部分情報出力オプションを指定してオブジェクトを生成した場合、ソース単位名と手続き名指定の部分情報出力指定ファイル名をフルパスで指定すると、指定した範囲の実行トレースとレコードダンプを出力する。</p> <p>その他の場合、部分情報出力は無効である。</p> <p>上記以外または未定義</p> <p>部分情報を出力しない。</p> |
| COB_OBJFILE = NO_DELETE | <p>1 つの原始プログラムを翻訳し、リンクした場合の、生成したオブジェクトファイルの削除動作を指定する。</p> <p>NO_DELETE を指定</p> <p>オブジェクトファイルを削除しない。</p> <p>未定義または DELETE を指定</p> <p>既存動作どおりオブジェクトファイルを削除する。</p>                                                                                                                                       |

| 環境変数                                                 | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>【Enterprise for Liux】</p> <p>COB_CODECHG = ON</p> | <p>実行時エラーメッセージの出力コードおよび小入出力文の入出力コードをロケールに合わせて自動コード変換する。</p> <p>設定値が ON の場合</p> <p>環境変数 LANG が ja_JP.UTF-8 のとき、</p> <ul style="list-style-type: none"> <li>・実行時エラーメッセージを UTF-8 の日本語で出力する(トレース情報ファイルへの出力を除く)。</li> <li>・明示的または暗黙的に <b>CONSOLE</b>, <b>STDOUT</b>, <b>STDERR</b> に対応付けられた呼び名を指定した <b>DISPLAY</b> 文(データ例外表示)が出力する実行時エラーメッセージを UTF-8 の日本語で出力する(トレース情報ファイルへの出力を除く)。</li> <li>・プログラムの固有文字集合がシフト JIS のとき、 <ul style="list-style-type: none"> <li>－ 明示的または暗黙的に <b>CONSOLE</b>, <b>STDIN</b> に対応付けられた呼び名を指定した <b>ACCEPT</b> 文の字類が日本語の作用対象の入力データを UTF-8 からシフト JIS に変換して入力する。</li> <li>－ 明示的または暗黙的に <b>CONSOLE</b>, <b>STDOUT</b>, <b>STDERR</b> に対応付けられた呼び名を指定した <b>DISPLAY</b> 文の字類が日本語のデータ項目および定数の作用対象の出力データをシフト JIS から UTF-8 に変換して出力する。</li> </ul> </li> </ul> <p>上記条件を満たさないとき、小入出力文の自動コード変換は行わない。また実行時エラーメッセージの出力コードは環境変数 LANG の設定に依存する。</p> <p>未定義または ON 以外を指定した場合</p> <p>小入出力文の自動コード変換は行わない。また実行時エラーメッセージの出力コードは環境変数 LANG の設定に依存する。</p> |

| 環境変数                                                    | 機能                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Enterprise for Windows】</b><br>COB_CSV_NOQUOTE = ON | <p>CSV ファイルの文字型項目出力時, ダブルクォーテーションで囲まず出力することを可能とする。</p> <p>環境変数はアプリケーションでの行順ファイル (CSV 形式) の初回オープン時に一度だけ取得され, アプリケーション内で同じ値が保証される。したがってファイル毎に異なる設定をしたり, アプリケーション実行中に設定を変更したりすることはできない。</p> <p>ON を指定した場合,</p> <p>CSV ファイルの文字型項目出力時, ダブルクォーテーションで囲まず出力する。</p> <p>例) セルのデータが ABC△△ の場合</p> <p>※”△”は空白を表す</p> <p>..., ABC△△, ...</p> <p>未定義または ON 以外の指定ありの場合,</p> <p>CSV ファイルの文字型項目出力時, ダブルクォーテーションで囲んで出力する。</p> |

| 環境変数                                                                         | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Enterprise for Windows】</b><br><b>【Standard】</b><br>COB_CSV_TRIMEND = ON | <p>CSV ファイルの文字型項目出力時のセル末尾空白 (1 バイト/2 バイト共に) を削除することを可能とする。</p> <p>(-CF/-Cj オプション指定なしで生成されたアプリケーションにのみ有効)</p> <p>環境変数はアプリケーションでの行順ファイル (CSV 形式) の初回オープン時に一度だけ取得され、アプリケーション内で同じ値が保証される。したがってファイル毎に異なる設定をしたり、アプリケーション実行中に設定を変更したりすることはできない。</p> <p>ON を指定した場合、</p> <p>CSV ファイルの文字型項目出力時のセル末尾空白 (1 バイト/2 バイト共に) を削除する。</p> <p>例) セルのデータが ABC△△ の場合</p> <p>※” △” は空白を表す</p> <p>…, “ABC”, …</p> <p>未定義または ON 以外の指定ありの場合、</p> <p>CSV ファイルの文字型項目出力時のセル末尾空白 (1 バイト/2 バイト共に) を削除しない。</p> |

| 環境変数                                                                                                  | 機能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>【Enterprise for Windows】</b><br><b>【Standard】</b><br>COB_SYSOUT_FORMAT_4D<br>= { SSF   PRN   TXT } | <b>【Enterprise for Windows】【Standard】</b> 小出力<br>DISPLAY で使用する SYSOUT ファイルのレコード<br>形式を変更する。<br>SSF 指定<br>SSF ヘッダ(制御データ)付きの SYSOUT ファイ<br>ル既定のレコード形式で出力する。<br>PRN 指定<br>PRN プリンタファイルのレコード形式で出力す<br>る。<br>TXT 指定<br>テキストファイルのレコード形式で出力する。右<br>端の空白は削除する。<br>未定義または SSF, PRN, TXT 以外の指定<br>環境変数 COB_SYSOUT_FORMAT の指定に従う。                                                                                                                                                                               |
| <b>【Enterprise for Windows】</b><br>xxx_PMD<br>= { APPEND   OUTPUT }                                   | <b>【Enterprise for Windows】</b> SELECT 句で内部フ<br>ァイル名(xxx)を指定した大容量記憶ファイルの<br>順／行順／CSV ファイルのオープンモードを変更<br>して実行する。<br>オープンモードの変更はプロセス毎に最初の 1<br>回目のみ有効である。1 度 CLOSE して再度 OPEN<br>するとき(2 回目以降), OPEN 文の記述どおり実行<br>する。<br>プログラム中に同じ内部ファイル名を指定した別<br>の SELECT 句がある場合, オープンモードが変更<br>されるのは先に OPEN するファイルのみである。<br>APPEND 指定<br>最初のオープンモードが OUTPUT のとき,<br>EXTEND オープンとして実行する。<br>OUTPUT 指定<br>最初のオープンモードが EXTEND のとき,<br>OUTPUT オープンとして実行する。<br>未定義または APPEND, OUTPUT 以外を指定<br>環境変数 COB_FILE_APPEND の指定に従う。 |



## G Unicode 対応について

[Standard] Unicode 機能の説明は未サポート。  
ここでは Unicode を扱う上での制限および注意事項について説明する。

### G.1 翻訳時の Unicode 対応

[Enterprise for Linux] ディレクトリ名やファイル名、環境変数に Unicode 固有文字は使用できない。

-CU オプションを指定しても、翻訳時には Unicode 固有文字を扱うことはできない。  
COBOL 原始プログラムのファイル名やパス名、登録集原文名、登録集原文を検索するディレクトリなど翻訳時に参照するディレクトリ名やファイル名に Unicode 固有文字が含まれている場合、正しい文字として認識できない。そのため、存在しないファイル名／パス名が指定されたものとして扱われエラーとなる。

例 1 COBOL 原始プログラム名に Unicode 固有文字が指定されているとき  
cob -M m<sup>3</sup>.cob ( m<sup>3</sup> が Unicode 固有文字 )  
LCB012 Invalid argument. FILE =?.cob

ファイル名が不正と認識され、上記エラーとなる。

例 2 環境変数 COB\_COPY に Unicode 固有文字が指定されているとき  
set COB\_COPY=C:¥CBLCOPY¥m<sup>3</sup>  
cob -M test01.cob  
F C0025 原文名または登録集名が誤っています。

登録集原文検索ディレクトリに Unicode 固有文字が指定されているため、登録集原文検索ディレクトリを正しく検索できず、上記エラーとなる。

### G.2 Unicode のコード範囲と本 COBOL でのサポート範囲

次の図は、Unicode のコード範囲と UCS-2 のサポート範囲を示したものである。

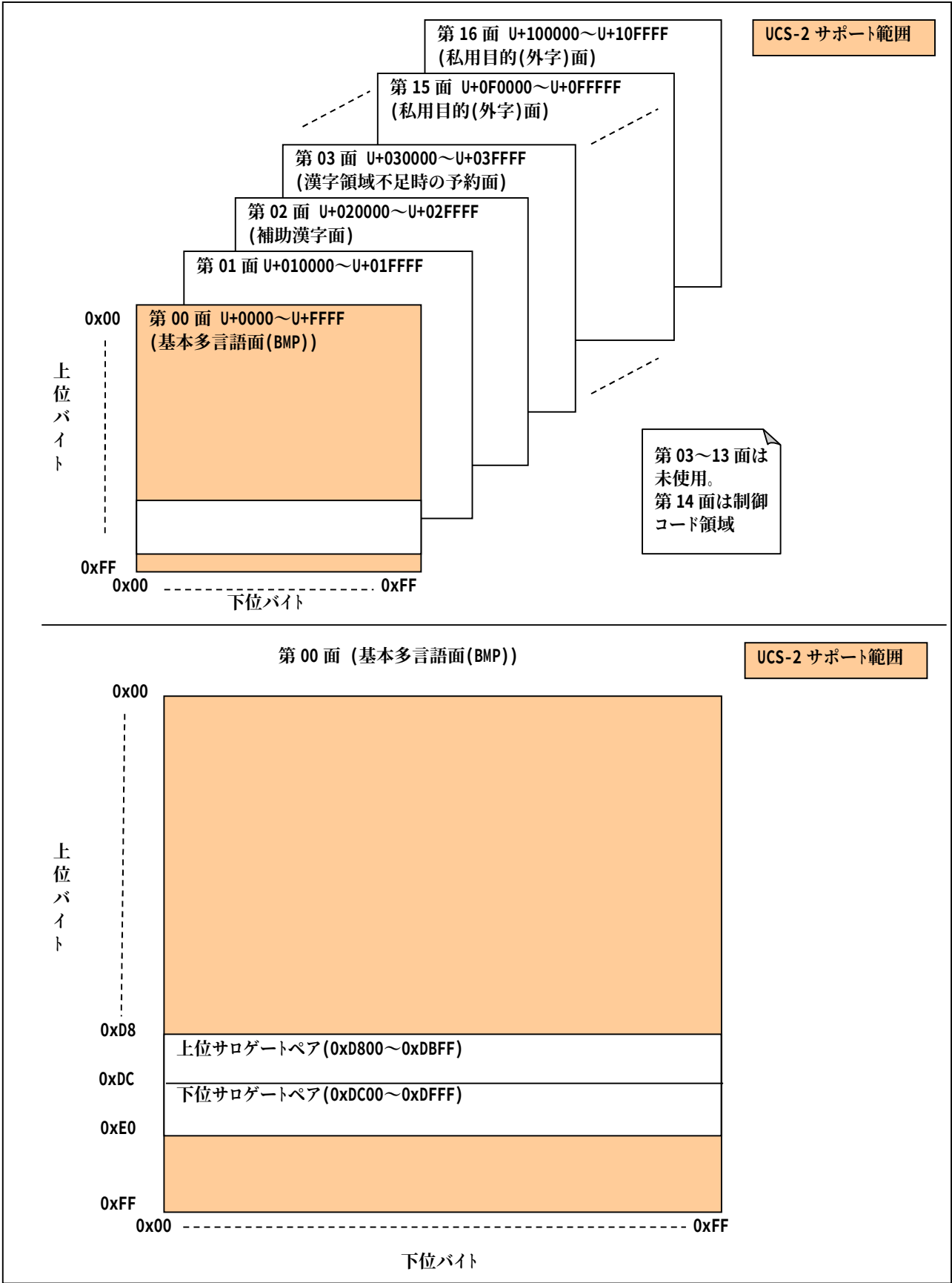
本 COBOL では、-CU オプション指定時、日本語項目を UCS-2 として扱う。したがって、日本語項目として扱える範囲は、基本多言語面(U+0000～U+FFFF)の 2 バイトであり、上位サロゲートペアと下位サロゲートペアの 4 バイトで表す第 01 面～第 16 面の範囲の文字は対象外となる。

また、-CU オプション指定時、英数字項目は UTF-8 として扱う。

UTF-8 は、Unicode の 1 文字を 1 バイト～4 バイトの可変長で表す方式である。

1 バイトで表現可能な範囲は、U+00～U+7F までの 128 文字である。

英数字項目は、1 バイトで 1 文字を表す方式であるため、UTF-8 で文字として表現できる範囲は、U+00～U+7F の範囲となる。



### G.3 文字コードに関連した制限

- ・ 索引ファイルの RECORD KEY／ALTERNATE RECORD KEY 句に日本語項目／日本語編集項目を指定できない。
- ・ 結合文字を使用することはできない。
- ・ PROGRAM COLLATING SEQUENCE に UCS-2 を指定できない。
- ・ APPLY SHIFT-CODE 指定時, シフト JIS での出力となる。
- ・ 実行時エラーメッセージは, シフト JIS での出力となる。
- ・ COBOL ソースの文字コードに Unicode は指定できない。
- ・ トレース情報ファイルの文字コードはシフト JIS であるため, 「実ファイル名／表名」項目に Unicode 固有文字があるときは「？」文字に変換する。

## H JIS2004 対応について

[Standard] [Enterprise for Windows] ここでは JIS2004 対応の制限事項および注意事項について説明する。

JIS2004 対応機能は、-CF オプション指定時、日本語項目および日本語編集項目中のデータに JIS2004 固有文字に対応した JIPS(J)コードを設定することによって、ファイルや画面の入出力データとして JIS2004 固有文字を使用できるようにする機能である。

本機能を使うには、コンパイル環境、アプリケーションの実行環境ともに以下の関連製品が必須である。

「FontAvenue UniAssist コード変換」

本 COBOL の JIS2004 サポート範囲は、上記製品の仕様に従う。

また、コンパイル、アプリケーションの実行を行う前に、環境変数 FA\_JIS2004\_CDMAP を設定しておく必要がある。環境変数の設定値については、「付録 D 環境変数」を参照。

### H.1 注意事項

[Standard] [Enterprise for Windows] 以下には JIS2004 固有文字を使用することはできない。

- ・ コンパイル時に指定するソースファイルのディレクトリ名やファイル名
- ・ 登録集原文名
- ・ 登録集原文を検索するディレクトリ名
- ・ 環境変数
- ・ 実行時に入出力するファイルのディレクトリ名やファイル名
- ・ [Standard] システムサブルーチン B\_CMOP, B\_GETENV, B\_PUTENV, B\_SYSTEM, CBLRUN, CBLSPAWN, CBLCHAINR, CBLCHAINC の引数
- ・ [Enterprise for Windows] システムサブルーチン B\_CMOP, B\_GETENV, B\_PUTENV, B\_SYSTEM の引数

また、以下の点に注意する必要がある。

- ・ 行順ファイルで CODE-SET IS UTF-8 または CODE-SET IS UTF-16LE を指定した場合、固有文字集合では 1 文字のバイト数が同じでも外部記憶媒体上ではバイト数が異なる場合があるため、このような文字を REWRITE 文で書きかえることはできない。  
たとえば、ファイルのレコードで X 項目で定義された領域に READ 文によって UTF-8 のファイルから "A" という文字が読み込まれた後、これを半角カナの "ア" に置き換えて REWRITE する場合、固有文字集合ではどちらも 1 バイトであるが、UTF-8 では "A" は 1 バイトだが、"ア" は 3 バイトであるため、置き換え前と置き換え後のレコードの長さが変わることになり、REWRITE は失敗する。  
同様に N 項目 1 文字で定義された領域に入る日本語文字でも、UTF-16LE の 2 バイトで表現する文字と 4 バイトで表現する文字とを置き換えようとすると REWRITE は失敗する。
- ・ 行順ファイルで CODE-SET IS UTF-8 または CODE-SET IS UTF-16LE を指定した場合、RECORD 句に指定した大きさがレコード記述項に指定した大きさより大きいと、入出力文の実行時にコード変換エラーが発生することがある。これはレコード記述項の大きさを超える部分 (不定) に変換不能文字が設定される場合があるためである。
- ・ [Standard] 画面制御機能にて使用できるフォントは以下のフォントに限定され、デフォルトフォントは「FA JIPS 明朝」に設定される。  
「FA JIPS 明朝」  
「FA JIPS ゴシック」  
デフォルトフォントを「FA JIPS 明朝」から「FA JIPS ゴシック」に変更するには、前もってコ

ンパイルオプション-Yr (レジストリのキー名指定) を指定して、アプリケーションを作成しておき、COBOL アプリケーションカスタマイザを用いてフォントを変更する。なおフォントの変更時、「FA JIPS 明朝」、「FA JIPS ゴシック」以外のフォントを指定した場合、アプリケーションの実行時、一番最初の画面制御機能の DISPLAY, ACCEPT 文で、番号 680 の実行メッセージを表示し、プログラムは操作員介入待ちとなる。解除キーが押された後、フォントを「FA JIPS 明朝」に変更し、処理を続行する。  
また、使用しているマシンのディスプレイ解像度によって、デフォルトフォントサイズの文字では見にくい場合があるため、必要に応じて COBOL アプリケーションカスタマイザを使用してフォントのサイズを変更する。

## H.2 制限事項

[Standard] [Enterprise for Windows] JIS2004 対応機能利用時は以下の機能は利用できない。

- ・ [Standard] COBOL WS エミュレータ利用時の ACCEPT, DISPLAY, STOP 定数文
- ・ [Standard] 一般索引ファイル (COBOL DB アクセス)
- ・ 一般索引ファイル (ISAM)
- ・ [Standard] データベース機能
- ・ 埋め込み SQL 機能 (Oracle Pro\*COBOL, COBOL SQL アクセス)
- ・ [Standard] COBOL 画面 GUI 化機能
- ・ [Enterprise for Windows] マルチスレッド

また、以下の制約がある。

- ・ [Standard] 対話型アプリケーションの小入力 ACCEPT が標準入力または CONSOLE に関連づけられているとき、英数字項目に日本語文字を入力することはできない。また、日本語項目に英数字文字を入力することはできない。
- ・ [Standard] 小出力 DISPLAY の UPON 呼び名に環境部の特殊名段落で GDD に対応づけた呼び名は指定できない。
- ・ [Standard] USAGE MIXED 句は指定できない。
- ・ -CF オプションを指定する場合は、アプリケーションを構成するすべての COBOL ソースに -CF オプションを指定する必要がある。
- ・ 日本語の大小比較は JIPS(J)コード体系に基づいて行われる。
- ・ 実行時エラーメッセージはシフト JIS での出力となる。
- ・ [Standard] 以下のすべての条件を満たす場合、プリンタファイルは日本語文字の出力はできない。英数字文字のみ出力可能。
  - -Cy オプション指定あり、または APPLY SHIFT-CODE 句指定なしプリンタファイル
  - プリントマネージャ経由の印刷

COBOL

プログラミングの手引

2012 年 5 月 初 版

2026 年 6 月 第 19 版

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

TEL (03) 3454-1111 (大代表)

(C) NEC Corporation 2012, 2026

日本電気株式会社の許可なく複製・改変などを行うことはできません。

本書の内容に関しては将来予告なしに変更することがあります。