

z/OS



DFSMSdfp ユーティリティー

z/OS



DFSMSdfp ユーティリティー

お願い

本書および本書で紹介する製品をご使用になる前に、393 ページの『特記事項』に記載されている情報をお読みください。

本書は、z/OS® (5694-A01) のバージョン 1 リリース 7、z/OS.e (5655-G52) のバージョン 1 リリース 7 に適用されます。改訂版などで特に断りのない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

本書は、SC26-7414-02 の改訂版です。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： SC26-7414-03
z/OS
DFSMSdfp Utilities

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.4

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1979, 2005. All rights reserved.

© Copyright IBM Japan 2006

目次

図	ix
表	xi
本書について	xiii
必要な製品知識	xiv
参照文書	xv
インターネット上の z/OS DFSMS 文書へのアクセス	xv
LookAt を使用してメッセージの説明を検索する方法	xv
表記規則	xvi
変更の要約	xix
SC26-7414-03 z/OS バージョン 1 リリース 7 の変更要約	xix
新規情報	xix
変更情報	xix
削除情報	xix
SC88-8979-02 (英文原典: SC26-7414-02) z/OS バージョン 1 リリース 3 の変更要約	xix
新規情報	xx
変更情報	xx
SC88-8979-01 (英文原典: SC26-7414-01) z/OS バージョン 1 リリース 3 の変更の要約	xx
新規情報	xx
変更情報	xx
第 1 章 はじめに	1
ユーティリティ・プログラムの機能の紹介	1
システム・ユーティリティ・プログラム	5
データ・セット・ユーティリティ・プログラム	5
制御	6
ジョブ制御ステートメント	7
ユーティリティ制御ステートメント	8
参照用の特別な工夫	10
第 2 章 IEBCOMPR (データ・セットの比較) プログラム	11
入出力	12
制御	13
ジョブ制御ステートメント	13
ユーティリティ制御ステートメント	14
IEBCOMPR の例	16
例 1: 磁気テープ上のデータ・セットを比較する	17
例 2: 磁気テープ上の順次データ・セットを比較する	17
例 3: 異なる密度で書き込まれた順次データ・セットを比較する	18
例 4: 順次データ・セットを比較する — 入力ストリームおよび磁気テープ入力	19

例 5: 2 つのジョブ・ステップ内の順次データ・セットをコピーして比較する	19
例 6: 2 つの区分データ・セットを比較する	20
例 7: 2 つのジョブ・ステップ内の区分データ・セットをコピーして比較する	20
例 8: 2 つの PDSE を比較する	21

第 3 章 IEBCOPY (ライブラリーのコピー) プログラム

ロード・モジュールをプログラム・オブジェクトに、またはその逆に変換する	24
区分データ・セットを PDSE に変換する	24
データ・セットのコピー	25
データ・セットをマージする	25
データ・セットをアンロード (バックアップ) する	25
アンロード・データ・セットをロードまたはコピーする	26
コピー、アンロード、またはロードするメンバーを選択する	28
コピー操作からメンバーを除外する	29
別名を持つメンバーをコピーする (COPY ステートメント)	29
プログラム・オブジェクトをコピーする (COPYGRP ステートメント)	32
区分データ・セットを圧縮する	33
ロード・モジュールを変更する	34
ロード・モジュールのコピーおよび再ブロック	35
IEBCOPY がテーブルおよびバッファのために仮想記憶域を使用する方法	36
制御ステートメントを指定しなければならない状態を回避する	37
入出力	38
戻りコード	38
制約事項	39
制御	41
ジョブ制御ステートメント	41
ユーティリティ制御ステートメント	47
実行する IEBCOPY 操作を決定する	48
操作の範囲	48
IEBCOPY の例	56
例 1: データ・セット全体をコピーする	57
例 2: 4 つのデータ・セットをマージする	58
例 3: データ・セットから選択したメンバーをコピーおよび置換する	60
例 4: データ・セットをアンロードおよび圧縮する	62
例 5: データ・セットをマージして、マージしたデータ・セットを圧縮する	63
例 6: 1 つの出力データ・セットへの複数のコピー操作	66

例 7: 複数の異なる出力データ・セットへの複数のコピー操作	69
例 8: データ・セットをロードする	74
例 9: 選択したメンバーをアンロード、ロード、コピー、およびマージする	75
例 10: ロード・モジュールを同じ場所に変更する	77
例 11: COPYMOD を使用してロード・モジュールを置換する	77
例 12: ロード・ライブラリーを再ブロックして、複数の異なる装置タイプに配布する	78
例 13: 区分データ・セットを PDSE に変換する	79
例 14: PDSE から PDSE にグループをコピーする	80
例 15: グループを PDSE から PDSE に置換付きでコピーする	81
例 16: 選択したグループを PDSE から PDSE にコピーする	81

第 4 章 IEBDG (テスト・データ・ジェネレーター) プログラム 83

パターンの選択	83
IBM 提供のパターン	83
ユーザー指定のパターン	84
レコード内のフィールドを変更する	85
入出力	86
制御	87
ジョブ制御ステートメント	87
ユーティリティ制御ステートメント	89
IEBDG の例	102
例 1: 順次データ・セットからコピーされたレコードに 2 進ゼロを入れる	103
例 2: 10 バイトの英字パターンのリップル	103
例 3: ユーティリティ制御ステートメントから出力レコードを作成する	104
例 4: 出力メンバーの基礎としてメンバーおよび入力レコードを使用する	105
例 5: 3 つの出力データ・セットにレコードを作成して、それらを 3 つの区分データ・セット・メンバーに書き込む	107
例 6: 独自のパターンを使用してレコードを作成する	109

第 5 章 IEBEDIT (ジョブ・ストリームの編集) プログラム 111

入出力	111
制御	112
ジョブ制御ステートメント	112
ユーティリティ制御ステートメント	113
IEBEDIT の例	114
例 1: 1 つのジョブをコピーする	115
例 2: 3 つのジョブからのステップをコピーする	115
例 3: 1 つのジョブからのステップを含め、別のジョブからのステップを除外する	116
例 4: JOBA および JOB STEPF のステートメントをコピーする	117
例 5: 入力データ・セット全体をコピーする	117

例 6: 新しい区切り文字を含めるためにデータ・セット全体をコピーする	118
-----------------------------------------------	-----

第 6 章 IEBGENER (データ・セットの順次コピー / 順次生成) プログラム . . . 119

バックアップ・コピーの作成	119
順次入力からの区分データ・セットまたは PDSE の生成	120
区分データ・セットまたは PDSE にメンバーを追加する	121
編集したデータ・セットの生成	121
論理レコード長の変更	123
2 バイト文字セット・データでの IEBGENER の使用	123
入出力	123
制御	124
ジョブ制御ステートメント	124
ユーティリティ制御ステートメント	129
IEBGENER の例	138
例 1: 順次データ・セットを印刷する	138
例 2: 順次入力から区分データ・セットを作成する	139
例 3: 順次入力を区分メンバーに変換する	140
例 4: インストリーム入力、順次データ・セットをテープ・ボリュームにコピーする	141
例 5: 非ブロック化ディスク・ファイルから磁気テープにブロック化コピーを生成する	141
例 6: ラベルのある順次入力データ・セットの編集およびコピー	142
例 7: 順次 z/OS UNIX ファイルを順次データ・セットに編集およびコピーする	143
例 8: 2 バイト文字セット・データを編集する	145

第 7 章 IEBIMAGE (プリンター・イメージの作成) プログラム 147

SYS1.IMAGELIB データ・セットのストレージ要件	147
SYS1.IMAGELIB データ・セットの維持	149
モジュールの一般構造	150
モジュールの命名規則	150
IEBIMAGE の使用	151
用紙制御バッファ・モジュールの作成	151
コピー変更モジュールの作成	156
文字配列テーブル・モジュールの作成	157
図形文字変更モジュールの作成	161
ライブラリー文字セット・モジュールの作成	163
入出力	166
制御	166
ジョブ制御ステートメント	166
ユーティリティ制御ステートメント	167
FCB ステートメント	169
COPYMOD ステートメント	173
TABLE ステートメント	176
GRAPHIC ステートメント	178
CHARSET ステートメント	180
INCLUDE ステートメント	183
NAME ステートメント	184

OPTION ステートメント	185
IEBIMAGE の例	187
例 1: 新しい 3800 用紙制御バッファ・モジュールを作成する	189
例 2: 3800 用紙制御バッファ・モジュールを置換する	189
例 3: 3800 用紙制御バッファ・モジュールを置換する	190
例 4: 新しい 3800 用紙制御バッファ・モジュールを作成する	191
例 5: 3800 用紙制御バッファ・モジュール STD3 を置換する	191
例 6: その他の ISO 用紙サイズ用に新しい 3800 用紙制御バッファ・モジュールを作成する	192
例 7: 4248 用紙制御バッファ・モジュールを作成する	193
例 8: 新しいコピー変更モジュールを作成する	194
例 9: 既存のコピー変更モジュールから新しいコピー変更モジュールを作成する	195
例 10: 文字配列テーブル・モジュールに新しい文字を追加する	196
例 11: 既存の文字配列テーブル・モジュールから新しい文字配列テーブル・モジュールを作成する	196
例 12: 文字配列テーブル・モジュールの図形文字を作成する	197
例 13: 文字配列テーブル・モジュールから図形参照を削除する	198
例 14: World Trade National Use Graphics 図形文字変更モジュールをリスト表示する	199
例 15: 文字変更モジュール World Trade GRAFMOD から図形文字変更モジュールを作成する	199
例 16: 新しい図形文字変更モジュールを作成して、文字配列テーブルを変更する	200
例 17: 複数のソースから図形文字変更モジュールを作成する	201
例 18: 図形文字変更モジュールで文字を定義して使用する	202
例 19: ライブラリー文字セット・モジュールをリスト表示する	204
例 20: ライブラリー文字セット・モジュールを作成する	205
例 21: ライブラリー文字セット・モジュールを作成して、文字配列テーブルを変更する	205
例 22 複数のソースからライブラリー文字セット・モジュールを作成する	207

第 8 章 IEBISAM プログラム 209

第 9 章 IEBTPCH (印刷 / 穿孔) プログラム 211

データ・セット全体または選択メンバーの印刷と穿孔	211
編集したデータ・セットの印刷と穿孔	212
2 バイト文字セット (DBCS) データの印刷と穿孔	212

選択レコードの印刷と穿孔	213
区分ディレクトリーの印刷と穿孔	213
ディスクまたは磁気テープへの印刷と穿孔	213
入出力	213
制御	214
ジョブ制御ステートメント	214
ユーティリティー制御ステートメント	215
IEBTPCH の例	226
例 1: 区分データ・セットを印刷する	227
例 2: 順次データ・セットを穿孔する	227
例 3: カード・デッキを複写する	228
例 4: 順次データ・セットをデフォルト・フォーマットで印刷する	228
例 5: 順次データ・セットをユーザー指定に従って印刷する	229
例 6: 3 つのレコード・グループを印刷する	229
例 7: 以前にフォーマット設定したデータ・セットを印刷する	231
例 8: 区分データ・セットのディレクトリーを印刷する	231
例 9: 区分データ・セットの選択レコードを印刷する	232
例 10: 区分データを 16 進値に変換してから印刷する	232
例 11: DBCS データが入っているメンバーを印刷する	233

第 10 章 IEBUPDTE (データ・セットの更新) プログラム 235

データ・セットのライブラリーの作成と更新	235
既存データ・セットの変更	235
データ・セットの編成の変更	236
入出力	236
制御	236
ジョブ制御ステートメント	236
ユーティリティー制御ステートメント	239
IEBUPDTE の例	251
例 1: 2 つのプロシージャを SYS1.PROCLIB に入れる	252
例 2: 3 つのメンバーからなるライブラリーを作成する	253
例 3: SYS1.MACLIB をソースにして新しいライブラリーを作成する	254
例 4: ライブラリー・メンバーを更新する	255
例 5: 新しいマスター・データ・セットの作成と選択レコードの削除	256
例 6: ライブラリー・メンバーの作成と更新	256
例 7: ライブラリー・メンバーにレコードを挿入する	258
例 8: レコードの番号を変更してライブラリー・メンバーに挿入する	259
例 9: カード入力から順次データ・セットを作成する	261
例 10: 順次データ・セットをボリューム間でコピーする	262
例 11: 新しい区分データ・セットを作成する	262

第 11 章 IEHINITT (テープの初期化) プログラム 265

標準ラベル・セットを磁気テープ上に配置する	267
DFSMSrmm の使用	268
入出力	269
制御	269
ジョブ制御ステートメント	269
ユーティリティー制御ステートメント	272
IEHINITT の例	275
例 1: EBCDIC ラベルを 3 つのテープに書き込む	275
例 2: ISO/ANSI ラベルをテープに書き込む	276
例 3: 通し番号の 2 つのグループを 6 つのテープ・ボリュームに付ける	276
例 4: 通し番号を 8 つのテープ・ボリュームに付ける	277
例 5: EBCDIC ラベルを異なる密度で書き込む	277
例 6: 通し番号を 2 つの密度でテープ・ボリュームに書き込む	277
例 7: ISO/ANSI ラベルにアクセス・コードを指定して書き込む	278
例 8: 取り外しや再装てんを行わないラベル付けに従って、磁気テープに書き込む	278

第 12 章 IEHLIST (システム・データのリスト表示) プログラム 281

区分データ・セットまたは PDSE のディレクトリの一のリスト表示	281
編集フォーマット	281
非編集 (ダンプ) フォーマット	282
ボリューム目録のリスト表示	283
編集フォーマット	283
入出力	286
制御	286
ジョブ制御ステートメント	286
ユーティリティー制御ステートメント	288
IEHLIST の例	291
例 1 DUMP および FORMAT を使用して、区分化されたディレクトリをリスト表示する	291
例 2 索引なしボリューム目録をリスト表示する	292

第 13 章 IEHMOVE (システム・データの移動) プログラム 293

ボリューム・サイズの互換性についての考慮事項	294
移動またはコピーされるデータ・セットのためのスペース割り振り	295
データ・セットの再ブロック	297
RACF における IEHMOVE の使用	298
データ・セットの移動またはコピー	298
順次データ・セット	299
区分データ・セット	299
BDAM データ・セット	302
マルチボリューム・データ・セット	303
アンロードされたデータ・セット	304
移動不能データ・セット	304

カタログ式データ・セットのグループの移動またはコピー	304
データ・セットのボリュームの移動またはコピー	305
入出力	306
制御	307
ジョブ制御ステートメント	307
ユーティリティー制御ステートメント	312
MOVE DSNAME および COPY DSNAME ステートメント	312
MOVE DSGROUP および COPY DSGROUP ステートメント	315
MOVE PDS および COPY PDS ステートメント	317
MOVE VOLUME および COPY VOLUME ステートメント	319
INCLUDE ステートメント	321
EXCLUDE ステートメント	322
SELECT ステートメント	322
REPLACE ステートメント	323
IEHMOVE の例	323
例 1: 複数の順次データ・セットをディスク・ボリュームから別々のボリュームに移動する	324
例 2: 区分データ・セットをディスク・ボリュームに移動してマージする	325
例 3: データ・セットのボリュームをディスク・ボリュームに移動する	326
例 4: 区分データ・セットを割り振りスペースに移動する	326
例 5: 区分データ・セット・ボリュームを移動してアンロードする	327
例 6: 順次データ・セットをラベルなしテープ・ボリュームにアンロードする	328
例 7: アンロードされた順次データ・セットをラベル付きテープからロードする	328
例 8: カタログ式データ・セットのグループを移動する	329

第 14 章 IEHPROGM (プログラム維持) プログラム 331

データ・セットまたはメンバーのスクラッチまたは名前変更	331
データ・セット・パスワードの維持	333
データ・セット・パスワードの追加	335
データ・セット・パスワードの置換	336
データ・セット・パスワードの削除	336
パスワード項目のリスト表示	336
入出力	337
制御	337
ジョブ制御ステートメント	337
ユーティリティー制御ステートメント	339
IEHPROGM の例	347
例 1: 一時システム・データ・セットのスクラッチ	347
例 2: 2 つのデータ・セットのスクラッチとアンカタログ	348
例 3: マルチボリューム・データ・セット・カタログの名前変更	348

例 4: 3 つのデータ・セットのアンカカタログ . . .	349
例 5: データ・セットの名前変更と新しいパスワードの定義	349
例 6: パスワード情報のリスト表示と置換	350
例 7: 区分データ・セットのメンバーの名前変更	350
第 15 章 IFHSTATR (ESV データのリスト表示) プログラム	353
ライブラリー内の磁気テープの品質評価	354
入出力	354
制御	356
IFHSTATR の例	356
付録 A. アプリケーション・プログラムからユーティリティ・プログラムを起動する	357
パラメーター・リストの作成	358
オプション・リスト	358
DD 名リスト	359
ページ・ヘッダー・パラメーター	361
戻りコード	362
IEBCOMPR の戻りコード	362
IEBCOPY の戻りコード	362
IEBDG の戻りコード	362
IEBEDIT の戻りコード	363
IEBGENER の戻りコード	363
IEBIMAGE の戻りコード	363
IEBPTPCH の戻りコード	364
IEBUPDTE の戻りコード	364
IEHINITT の戻りコード	365
IEHLIST の戻りコード	365
IEHMOVE の戻りコード	365
IEHPROGM の戻りコード	366
付録 B. アンロード区分データ・セット・フォーマット	367
はじめに	367

アンロード・データ・セット内のレコード	367
アンロード・データ・セットの各種フォーマット	368
詳細レコードの説明	369
付録 C. ユーティリティ・プログラムのユーザー出口を指定する	377
全体的な指針	377
ユーティリティ・プログラムから出口ルーチンへ入るときのレジスターの内容	378
プログラミングに関する注意点	378
出口ルーチンからの戻り	378
ラベル処理ルーチンに渡されるパラメーター	381
非ラベル処理ルーチンに渡されるパラメーター	382
ユーザー・ラベルの処理	382
ユーザー・ラベルをデータ・セット記述子として処理する	383
合計ルーチンへの出口	384
ユーザー・ラベルをデータとして処理する	385
IEBDG での出口ルーチンの使用	385
付録 D. IEHLIST の VTOC リスト表示	387
IEHLIST のフォーマット済み VTOC リスト表示の各フィールドの説明	388
付録 E. アクセシビリティ	391
支援機能の使用	391
ユーザー・インターフェースのキーボード・ナビゲーション	391
z/OS 情報	391
特記事項	393
プログラミング・インターフェース情報	394
商標	394
用語集	395
索引	407



1. IEBCOMPR を使用して比較できる区分ディレクトリー	12
2. IEBCOMPR を使用して比較できない区分ディレクトリー	12
3. 区分データ・セットをコピーする — フル・コピー	58
4. 3 つの入力区分データ・セットからコピーする	59
5. メンバー・レベルで置換を指定した選択的なコピー	61
6. “置換”を指定したフル・コピーの後、同所圧縮を行う	64
7. 複数のコピー操作 / コピー・ステップ	68
8. 複数のコピー操作 / ジョブ・ステップ内のコピー・ステップ	72
9. IEBDG のアクション	86
10. 入力レコードからフィールドを選択して出力レコードで使用する	90
11. IEBDG を使用して出力レコード内にフィールドを配置するときのデフォルト	97
12. 出力位置の指定されたフィールドの配置	97
13. 一部の出力位置だけが指定されたフィールドの配置	98
14. ユーティリティー制御ステートメントを使用して出力レコードを作成する	99
15. ジョブ・ステップが終了したときの出力レコード	105
16. ジョブ・ステップが終了したときの出力区分メンバー	106
17. ジョブ・ステップが完了したときの区分データ・セット・メンバー	108
18. ジョブ・ステップが終了したときの出力レコードの内容	109
19. IEBGENER による、順次入力からの区分データ・セットまたは PDSE の作成	120
20. IEBGENER による、区分データ・セットまたは PDSE へのメンバーの追加	121
21. IEBGENER による、順次データ・セットの編集	122
22. 順次データ・セットを編集してコピーする方法	144
23. 3800 の一般的なモジュール・ヘッダー	150
24. 3800 FCB モジュールの構造	152
25. 4248 FCB モジュールの構造	153
26. 4248 FCB モジュールの制御バイト	153
27. 4248 FCB モジュールのデータ・バイト	154
28. IEBIMAGE による用紙制御バッファ・モジュールのリスト表示	155
29. コピー変更モジュールの構造	156
30. IEBIMAGE によるコピー変更モジュールの 3 つのセグメントのリスト表示	157
31. 文字配列テーブル・モジュールの構造	159
32. 図形文字変更モジュール	159
33. IEBIMAGE による文字配列テーブル・モジュールのリスト表示	160
34. 1 文字のための 3800 図形文字変更モジュールの構造	162
35. IEBIMAGE による図形文字変更モジュールの 2 つのセグメントのリスト表示	163
36. 1 文字のための 3800 モデル 3 ライブラリー文字セット・モジュールの構造	164
37. IEBIMAGE によるライブラリー文字セットの 2 つのセグメントのリスト表示	165
38. INITT ステートメントの指定内容と初期ポリューム・ラベル情報の出力結果	272
39. 区分ディレクトリーの編集された項目のサンプル	282
40. 区分データ・セットまたは PDSE ディレクトリーの非編集リスト・フォーマット	283
41. IEHMOVE コピー操作の前後の区分データ・セット	301
42. IEHMOVE を使った 2 つのデータ・セットのマージ	301
43. IEHMOVE を使った 3 つのデータ・セットのマージ	302
44. データ・セットの保護状況とそのパスワードとの関係	334
45. パスワード項目のリスト表示	336
46. SMF タイプ 21 (ESV) レコード・フォーマット	353
47. IFHSTATR による出力のサンプル	355
48. ディレクトリー・レコードのレイアウト	372
49. 属性レコードのレイアウト	373
50. 注釈リスト・レコードのレイアウト	374
51. メンバー・データ・レコードのレイアウト	375
52. メンバー・データ・ブロックのレイアウト	375
53. ファイル終わりブロックのレイアウト	376
54. OPEN、EOV、CLOSE の実行時にシステムが行う処置	384
55. ユーザー合計ルーチンの戻りコード	384
56. IEBDG ユーザー出口の戻りコード	386
57. IEHLIST の出力例 — VTOC (拡張フォーマット順次データ・セット)	387
58. IEHLIST の出力例 — VTOC (順次データ・セット、区分データ・セット、PDSE)	388

表

1. タスクおよびユーティリティー・プログラム	1	34. IEBUPDTE のユーティリティー制御ステートメント	239
2. システム・ユーティリティー・プログラム	5	35. 機能ステートメントの NEW、MEMBER、および NAME パラメーター	245
3. データ・セット・ユーティリティー・プログラム	6	36. IEBUPDTE の例の一覧表	251
4. 例の一覧表	10	37. 番号変更されるシーケンス番号の例	258
5. IEBCOMPR で使用するジョブ制御ステートメント	13	38. シーケンスが変更されたシーケンス番号	260
6. IEBCOMPR で使用するユーティリティー制御ステートメント	14	39. IEHINITT で使用するジョブ制御ステートメント	269
7. LABEL ステートメントの構文	15	40. IEHINITT の例の一覧表	275
8. IEBCOMPR の例の一覧表	16	41. IEHLIST で使用するジョブ制御ステートメント	286
9. IEBCOPY で使用するジョブ制御ステートメント	41	42. IEHLIST で使用するユーティリティー制御ステートメント	288
10. IEBCOPY で使用するユーティリティー制御ステートメント	47	43. IEHLIST の例の一覧表	291
11. 1 つのジョブ・ステップ内の複数のコピー操作	48	44. 移動操作およびコピー操作 - サイズに関してソース・ボリュームと互換性のある DASD 受け取り側ボリューム	294
12. IEBCOPY の例の一覧表	56	45. 移動操作およびコピー操作 - サイズに関してソース・ボリュームと互換性のない DASD 受け取り側ボリューム	295
13. IBM 提供のテスト・データ・パターン	84	46. 移動操作およびコピー操作 - 非 DASD の受け取り側ボリューム	295
14. IEBDBG で使用するジョブ制御ステートメント	87	47. 順次データ・セットの移動およびコピー	299
15. EXEC ステートメントの構文	88	48. 区分データ・セットの移動およびコピー	300
16. IEBDBG で使用するユーティリティー制御ステートメント	89	49. カタログ式データ・セットのグループの移動およびコピー	305
17. 互換性のある IEBDG 操作	95	50. データ・セットのボリュームの移動およびコピー	306
18. IEBDBG の例の一覧表	102	51. IEHMOVE で使用するジョブ制御ステートメント	307
19. IEBEDIT で使用するジョブ制御ステートメント	112	52. IEHMOVE で使用するユーティリティー制御ステートメント	312
20. IEBEDIT の例の一覧表	115	53. IEHMOVE の例の一覧表	324
21. IEBGENER で使用するジョブ制御ステートメント	124	54. IEHPROGM で使用するジョブ制御ステートメント	337
22. 出力 DD ステートメントの影響	129	55. IEHPROGM で使用するユーティリティー制御ステートメント	339
23. IEBGENER で使用するユーティリティー制御ステートメント	130	56. IEHPROGM の例の一覧表	347
24. IEBGENER の例の一覧表	138	57. IFHSTATR で使用するジョブ制御ステートメント	356
25. 各装置のトラック (T) ごとのメンバー	148	58. COPYR1 記述子レコードの内容	369
26. IEBIMAGE で使用するジョブ制御ステートメント	167	59. COPYR2 記述子レコードの内容	371
27. IEBIMAGE で使用するユーティリティー制御ステートメント	168	60. ユーティリティーに指定できるユーザー出口ルーチン	377
28. IEBIMAGE によるコピー変更モジュールのオーバーラン注釈付きのリスト表示	187	61. ユーザー出口ルーチンが出す戻りコード	379
29. IEBIMAGE の例の一覧表	187	62. 非ラベル処理出口ルーチンのパラメーター・リスト	382
30. IEBPTPCH で使用するジョブ制御ステートメント	214		
31. IEBPTPCH のユーティリティー制御ステートメント	216		
32. IEBPTPCH の例の一覧表	226		
33. IEBUPDTE で使用するジョブ制御ステートメント	236		

本書について

本書の目的は、システム・プログラマーおよびアプリケーション・プログラマーが、z/OS™ DFSMS ユーティリティ・プログラムを使用して、システム、ユーザー・データ、およびデータ・セットを操作するのを支援することです。大部分のプログラムは、以下のパターンで解説されます。

1. プログラムによって実現できる機能の紹介および解説。この解説には通常、プログラムの使用法の概説、用語の定義、および図が含まれます。
2. ユーティリティによってサポートされる機能、および各機能の目的。
3. プログラムによって使用される入力、および生成される出力。
4. ジョブおよびユーティリティ制御ステートメントによるプログラムの制御。ジョブ制御ステートメントについては、ユーティリティ・プログラムでの使用がそのプログラム特有のものである場合だけが解説されます。ユーティリティ制御ステートメントについては、すべてが解説されます。
5. ジョブおよびユーティリティ制御ステートメントを含む、プログラムの使用の例。

以下のユーティリティの使用は勧められていません。

- ICAPRTBL—3211 印刷装置がサポートされなくなりました。
- IEBISAM は現在提供していません。その代わりに、VSAM を使用する必要があります。ISAM データ・セットを VSAM キー順データ・セットに変換することに関する情報は、z/OS DFSMS データ・セットの使用法、および z/OS DFSMS カタログのためのアクセス方式サービス・プログラムの REPRO コマンドの項を参照してください。
- IEHMOVE—DFSMSdss™ および IEBCOPY をお勧めします。
- IEHPROGM—カタログおよび削除の全機能については IDCAMS をお勧めします。
- IEHATLAS—IEHATLAS プログラムは現在提供されていません。装置サポート機能 (ICKDSF) を代わりにご使用ください。

これらのユーティリティに関して提供される情報は、互換性のためだけのものです。

特定用途のユーティリティのいくつかについては、本書では解説していません。以下のリストに、それらの名前と機能、およびそれについての解説を含む資料名を示します。

ユーティリティー	機能	参照
IDCAMS	ユーザーが VSAM データ・セットの定義、操作、削除、統合カタログ機能カタログの定義、操作、および SAM と ISAM データ・セットのコピー、印刷、または VSAM データ・セットへの変換を実行できるようにします。	<i>z/OS DFSMS</i> カタログのためのアクセス方式サービス・プログラム, SC88-9109
装置サポート機能 (ICKDSF)	DASD ボリュームの初期化および維持に使用します。	<i>Device Support Facilities User's Guide and Reference</i> , GC35-0033
DFSMSdss	ダンプ/リストアおよびフリー・スペースのフラグメント化の低減などの DASD ユーティリティー機能。	<i>z/OS DFSMSdss</i> ストレージ管理ガイド, SC88-8983 <i>z/OS DFSMSdss</i> ストレージ管理リファレンス, SC88-8999
オフライン IBM® 3800 ユーティリティー (CIPOPS)	IBM 3800 テープ印刷サブシステム・フィーチャーと共に使用するプリンター・ライブラリー機能。	<i>Offline 3800 Utility</i> (SH20-9138)
IEFBR14	戻りコード 0 を戻すだけでその他の処置は実行しませんが、ジョブ・スケジューラーは構文エラーの有無に関して JCL ステートメントを検査し、データ・セットのためにスペースを割り振り、後処理を実行します。	<i>z/OS MVS JCL</i> ユーザーズ・ガイド, SA88-8570
AMASPZAP (スーパーザップ)	ディスク・データ・セットの検査および変更で使用されます。	<i>z/OS MVS</i> 診断: ツールと保守援助プログラム, GA88-8561
IPCS (対話式プログラム制御システム)	システム・ダンプおよび GTF トレースを使用して、システムおよびアプリケーション上の問題を診断、分析、および印刷するシステム。	<i>z/OS MVS IPCS</i> ユーザーズ・ガイド, SA88-8568
EDGINERS	磁気テープ・ボリューム上に IBM および ISO/ANSI 標準ラベルを書くとともに、磁気テープの自動消去およびラベル付けを行います。	<i>z/OS DFSMSrmm</i> インプリメンテーションとカスタマイズのガイド, SC88-8977
SuperC	データ・セット間の内容を比較して、その違いを報告します。	<i>HLASM Toolkit Feature User's Guide</i> , GC26-8710 <i>z/OS ISPF</i> ユーザーズ・ガイド 第 1 巻, SC88-8965

必要な製品知識

本書を効果的に使用するには、以下の知識が必要です。

- ご使用のシステムでテープを使用するアプリケーション
- DFSMS
- MVS™ 内での割り振り方式
- ジョブ制御言語 (JCL)
- データ管理
- システム管理機能 (SMF)
- テープおよび DASD ハードウェア

- テープのマウント管理

以下の参照資料に書かれている情報も理解しておいてください。

参照文書

本書では、以下の資料を参照しています。

資料名	資料番号
<i>z/OS MVS Programming: Authorized Assembler Services Guide</i>	SA22-7608
<i>z/OS MVS プログラミング: アセンブラー・サービス 解説書 ABE-HSP</i>	SA88-8578
<i>IBM 3800 Printing Subsystem Programmer's Guide</i>	GC26-3846
<i>IBM 3800 Printing Subsystem Model 3 Programmer's Guide: Compatibility</i>	SH35-0051
<i>z/OS MVS JCL 解説書</i>	SA88-8569
<i>z/OS MVS JCL ユーザーズ・ガイド</i>	SA88-8570
<i>z/OS DFSMS カタログのためのアクセス方式サービス・プログラム</i>	SC88-9109
<i>z/OS DFSMS Installation Exits</i>	SC26-7396
<i>z/OS MVS プログラム管理: ユーザーズ・ガイドおよび解説書</i>	SA88-86880
<i>z/OS DFSMS Macro Instructions for Data Sets</i>	SC26-7408
<i>z/OS DFSMS カタログの管理</i>	SC88-8978
<i>z/OS DFSMSdfp ストレージ管理リファレンス</i>	SC88-8974
<i>z/OS DFSMSdfp 拡張サービス</i>	SD88-6462
<i>z/OS DFSMS データ・セットの使用法</i>	SC88-9114
<i>z/OS DFSMS Using Magnetic Tapes</i>	SC26-7412
<i>Reference Manual for the IBM 3800 Printing Subsystem Model 1</i>	GA26-1635
<i>z/OS MVS 診断: ツールと保守援助プログラム</i>	GA88-8561
<i>z/OS TSO/E Programming Services</i>	SA22-7789
<i>z/OS MVS プログラミング: アセンブラー・サービス ガイド</i>	SA88-8577
<i>z/OS MVS システム・メッセージ 第 7 巻 (IEB-IEE) (IEB-IEE)</i>	SA88-8603
<i>z/OS MVS システム・メッセージ 第 8 巻 (IEF-IGD) (IEF-IGD)</i>	SA88-8604

インターネット上の z/OS DFSMS 文書へのアクセス

CD-ROM でソフトコピー文書を提供しているほかに、IBM では、インターネット上でもライセンス不要な z/OS ソフトコピー文書にアクセスできるようにしています。z/OS 資料を表示、検索、および印刷するためには、以下の z/OS Internet Library に進みます。

www.ibm.com/servers/eserver/zseries/zos/bkserv/

LookAt を使用してメッセージの説明を検索する方法

LookAt は、お客様が受け取る IBM のほとんどのメッセージや、数々のシステムの異常終了およびコードに関する説明を検索できる、オンライン機能です。LookAt では、通常、該当メッセージの説明がただちに表示されるため、従来の方法よりも短時間で、必要な情報を検索することができます。

お客様は LookAt を使用して、以下のロケーションから z/OS[®]、z/VM[®]、および VSE の要素や機能についての IBM のメッセージの説明を検索することができます。

- インターネット。お客様は、 <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/> にある LookAt Web サイトから、直接、IBM メッセージの説明をご利用できます。
- ご使用の z/OS TSO/E ホスト・システム。お客様は、z/OS または z/OS.e にコードをインストールして、TSO/E コマンド行 (例えば、TSO/E プロンプト、ISPF、または OMVS を実行する z/OS UNIX[®] システム・サービス) から LookAt を使用し、IBM メッセージの説明をご利用できます。
- ご使用の Windows[®] ワークステーション。お客様はコードをインストールして、Windows DOS コマンド行から LookAt を使用し、「z/OS Collection」(SK3T-4269) 上の IBM メッセージの説明をご利用できます。
- ご使用のワイヤレス・ハンドヘルド・デバイス。お客様は、無線アクセスとインターネット・ブラウザを備えたハンドヘルド・デバイス (例えば、ポケット PC 用 Internet Explorer、Palm OS 用 Blazer または Eudora、あるいは Linux ハンドヘルド・デバイス用 Opera など) から LookAt Mobile Edition をご利用できます。LookAt Web サイトから、LookAt Mobile Edition へリンクしてください。

ご使用の「z/OS Collection」(SK3T-4269) のディスク、または LookAt Web サイトからホスト・システム、または Windows ワークステーションに LookAt をインストールするコードを入手できます (「ダウンロード」を選択して、要求に適合するプラットフォーム、リリース、コレクション、およびロケーションを選択します)。より詳細な情報については、ダウンロード・プロセス中に参照可能な LOOKAT.ME ファイルをご覧ください。

表記規則

一律の表記によってユーティリティー制御ステートメントの構文が説明されます。この表記は言語の一部ではなく、ステートメントの構文を説明する方法の一つにすぎません。本書にあるステートメント構文の解説では、以下の規則が使用されます。

- [] 大括弧で囲まれているのはオプションの項目です。その項目は指定可能ですが、指定する必要はありません。以下に例を示します。

[length]
[MF=E]

- | OR 記号 (垂直のバー) は代替項目を区分します。示されているデフォルトを使用するのでなければ、それらの項目のうちの 1 つだけを指定しなければなりません。以下に例を示します。

[REREAD|LEAVE]
[length|S']

- { } 中括弧で囲まれているのは代替項目です。それらの項目のうちの 1 つだけを使用しなければなりません。以下に例を示します。

BFTEK={SIA}
{KID}
{address|SIO}

代替項目は中括弧の垂直スタックによって示されることもあります。以下に例を示します。

MACRF={{(R[CIP])}{(W[CIPIL])} {(R[C],W[C])}}

上記の例では、垂直スタックから 1 つの項目だけを選択しなければなりません。

... 省略符号は、その省略符号の直前の項目を繰り返し指定できることを示します。たとえば、以下のようになります。

(dcbaddr,[options]),. . .)

‘ ’ ‘ ’ は、次のパラメーターの前にブランク (空のスペース) が必要であることを示します。

英大文字の太字体

大文字の太字体は、示されたとおりにコーディングしなければならない項目を示します。これらの項目はキーワードとそれに続く句読点記号 (コンマ、括弧、および等号) で構成されています。以下に例を示します。

- **CLOSE , , , ,TYPE=T**
- **MACRF=(PL,PTC)**

下線のある英大文字の太字体

下線のある大文字の太字体は、代替項目のどれも指定しない場合に使用されるデフォルトを示します。以下に例を示します。

- **[EROPT={ACC|SKP|ABE}]**
- **[BFALN={FID}]**

小文字のイタリック

小文字のイタリック体 は、通常パラメーターごとに説明される仕様および制限に応じてユーザーが指定する値を示します。以下に例を示します。

- *number*
- *image-id*
- *count*

keyword=device=list

keyword は、VOL、FROM、または TO に置き換えられます。

device は 3380 などの総称名、または SYSDA などの非公式名に置き換えられます。

DASD では、*list* はコンマで区切られた 1 つ以上のボリューム通し番号に置き換わります。複数のボリューム通し番号がある場合、*list* フィールドの全体を小括弧で囲まなければなりません。

磁気テープの場合、*list* は 1 つまたは複数の『ボリューム通し番号、データ・セットのシーケンス番号』の対によって置き換えられます。それぞれの対はコンマによって次の対と区分されます。複数の対がある場合、*list* フィールドの全体を FROM=3480=(tapeA,1,tapeB,1) のように小括弧で囲まなければなりません。

必須のキーワードおよびシンボル

ここに示されているフォーマット (使用されている強調表示のタイプに注目) で示された項目は、示されたとおりにコーディングしてください。これらの

項目はキーワードとそれに続く句読点記号 (コンマ、括弧、および等号) で構成されています。以下に例を示します。

- **CLOSE , , ,TYPE=T**
- **MACRF=(PL,PTC)**

注: この項目のタイプを識別するために使用される形式 (強調表示のタイプ) は、ソフトコピー・ブックを表示するために使用するディスプレイ装置によって異なります。発行されている本書のハードコピー版は、このタイプの項目を大文字の太字体で表示します。

デフォルト値

ここに示されているフォーマット (使用されている強調表示のタイプに注目) は、代替項目のどれも指定しない場合に使用されるデフォルト値を示します。以下に例を示します。

- **[EROPT={ACCSKPIABE}]**
- **[BFALN={FID}]**

注: この項目のタイプを識別するために使用される形式 (強調表示のタイプ) は、ソフトコピー・ブックを表示するために使用するディスプレイ装置によって異なります。発行されている本書のハードコピー版は、このタイプの値を下線を付けた大文字の太字体で表示します。

ユーザーの指定値

ここに示されているフォーマット (使用されている強調表示のタイプに注目) は、通常パラメーターごとに説明される仕様および制限に応じてユーザーが指定する値を示します。以下に例を示します。

- *number*
- *image-id*
- *count*

注: この項目のタイプを識別するために使用される形式 (強調表示のタイプ) は、ソフトコピー・ブックを表示するために使用するディスプレイ装置によって異なります。発行されている本書のハードコピー版は、このタイプの値を小文字のイタリック体で表示します。

変更の要約

本書には、用語、細かな修正、および編集上の変更が含まれています。テキストおよび図に対する技術的な変更または追加は、変更個所の左側に縦線で示しています。

本書は、オンライン *z/OS LibraryCenter* (例として) で以下のタイプの拡張検索に対して使用可能となりました。

SC26-7414-03 *z/OS* バージョン 1 リリース 7 の変更要約

本書には、*z/OS* バージョン 1 リリース 3 *DFSMSdfp* ユーティリティー (SC88-8979-02) にすでに記載されている情報も含まれています。

以下のセクションで、その情報に対する変更内容を要約します。

新規情報

この版には、次の新規情報が含まれています。

- ラージ・フォーマットの順次データ・セットに対するサポートが追加されました。特に、*IEBGENER* に対する入力と出力データ・セットとして、および *IEBCOPY* に対するアンロード・データ・セットとしてのサポートです。ラージ・フォーマットのデータ・セットは *IEHMOVE* と一緒に使用しないでください。

変更情報

- *ISAM* データ・セットを *VSAM* に変換することは、推奨ではなく要件に変わりました。
- この版には、さまざまな保守内容と改善された変更点が盛り込まれています。

削除情報

- *IEBISAM* プログラムに関する情報が削除されました。本書全体を通して他の *ISAM* 固有情報も同様に削除されました。

IEHATLAS プログラムは現在提供していません。装置サポート機能 (*ICKDSF*) を代わりにご使用ください。

SC88-8979-02 (英文原典: SC26-7414-02) *z/OS* バージョン 1 リリース 3 の変更要約

本書には、*z/OS* バージョン 1 リリース 1 *DFSMSdfp* ユーティリティー (SC88-8979-01) にすでに記載されている情報も含まれています。

以下のセクションで、その情報に対する変更内容を要約します。

新規情報

この版には、次の新規情報が含まれています。

- 「12 章 IEHINITT」では、許可ユーザーだけが IEHINITT プログラムを使用できることを確実にする方法を示します。その方法を行うコマンド・シーケンスが例示されています。
- このセクションでは、インストール・システム出口を使用し、初期化要求を調べて、任意のボリュームを初期化するかどうかを決定することについても、説明します。

変更情報

この版には、さまざまな保守内容と改善された変更点が盛り込まれています。

SC88-8979-01 (英文原典 : SC26-7414-01) z/OS バージョン 1 リリース 3 の変更の要約

本書には、z/OS バージョン 1 リリース 1 DFSSMSdfp ユーティリティ (SC88-8979-00) にすでに記載されている情報も含まれています。

以下のセクションで、その情報に対する変更内容を要約します。

新規情報

この版には次の情報を含んでいます。129 ページの表 22 は、SYSUT2 DD ステートメントの RECFM、LRECL、および BLKSIZE の可用性の影響を示しています。

変更情報

次の情報が、本書で変更されています。「第 6 章、IEBGENER」では、デフォルトのブロック・サイズ値をシステム決定ブロック・サイズ (SDB) パラメーターに設定する方法を説明しています。

第 1 章 はじめに

DFSMS には、データの編成および維持に役立つユーティリティー・プログラムが備わっています。ユーティリティーとは、共通に必要とされる機能を実行する小さなプログラムのことです。必要な機能を実行するプログラムを見つけるために、『ユーティリティー・プログラムの機能の紹介』が役立ちます。

ユーティリティー・プログラムの機能の紹介

表 1 は、DFSMS ユーティリティー・プログラムを使用して実行できるタスクのリストです。『タスク』欄は、実行するタスクを示します。『オプション』欄は、そのタスクをより具体的に定義します。『1 ユーティリティー』欄は、そのタスクに特に適合するユーティリティーを示します。『2 次ユーティリティー』欄は、そのタスクを実行するために使用できる他のユーティリティーを示します。

表 1. タスクおよびユーティリティー・プログラム

タスク	オプション	1 次 ユーティリティー	2 次 ユーティリティー
追加	区分データ・セットのメンバー	IEBUPDATE, IEBGNER	IEBDG
	パスワード	IEHPROGM	
同じ場所での変更	ロード・モジュール	IEBCOPY	
カタログ	カタログ内のデータ・セット	IEHPROGM	
変更	データ・セット編成	IEBUPDTE	IEBGNER, IEBTPCH
	論理レコード長	IEBGNER	
比較	z/OS UNIX システム・サービス (z/OS UNIX) ファイル (たとえば HFS ファイル)	IEBCOMPR	
	区分データ・セット	IEBCOMPR	
	順次データ・セット	IEBCOMPR	
	PDSE	IEBCOMPR	
圧縮	区分データ・セット	IEBCOPY	
同じ場所での圧縮	区分データ・セット	IEBCOPY	
区分データ・セットへの変換	プログラム・オブジェクトを含むアンロードされた PDSE は PDS にロードすることができません。データ・オブジェクトを含むアンロードされた PDSE は PDS にロードできますが、すべての拡張属性が失われます。	IEBCOPY	
	順次データ・セット	IEBGNER	IEBUPDTE
	PDSE	IEBCOPY	

表 1. タスクおよびユーティリティー・プログラム (続き)

タスク	オプション	1 次 ユーティリティー	2 次 ユーティリティー
PDSE への変換	区分データ・セット	IEBCOPY	
	区分データ・セットまたは PDSE のアンロードされたコピー	IEBCOPY	
	順次データ・セット	IEBGENER	IEBUPDTE
順次データ・セットへの変換	区分データ・セットまたは PDSE	IEBGENER	IEBUPDTE
コピー	ロード・モジュールまたはロード・モジュール・ライブラリー	IEBCOPY	
	区分データ・セット	IEBCOPY	IEHMOVE
	データ・セットのボリューム (磁気テープまたはディスク上の)	IEHMOVE	
	ジョブ・ステップ	IEBEDIT	
	区分データ・セットの選択されたメンバー	IEBCOPY	IEHMOVE
	順次データ・セット	IEBGENER	IEHMOVE, IEBUPDTE, IEBPTPCH
	PDSE	IEBCOPY	
	PDSE メンバーのグループ	IEBCOPY	
	PDSE の選択されたメンバー	IEBCOPY	
	作成	区分データ・セットまたは PDSE のバックアップ・コピー	IEBCOPY
文字配列テーブル・モジュール		IEBIMAGE	
コピー変更モジュール		IEBIMAGE	
3800 または 4248 用紙制御バッファ・モジュール		IEBIMAGE	
図形文字変更モジュール		IEBIMAGE	
ライブラリー文字セット・モジュール		IEBIMAGE	
区分メンバーのライブラリー		IEBGENER	IEBUPDTE
区分データ・セットまたは PDSE のメンバー		IEBGENER	IEBDG, IEBUPDTE
順次出力データ・セット		IEBDG	IEBGENER, IEBPTPCH
索引順次データ・セット		IEBDG	
出力ジョブ・ストリーム		IEBEDIT	
削除	データ・セットまたは区分データ・セットのメンバー	IEHPROGM	
	パスワード	IEHPROGM	
	カタログ項目	IEHPROGM	
	区分データ・セットのレコードまたは PDSE のメンバー	IEBUPDTE	
区分データ・セットまたは PDSE の編集および変換	順次データ・セット	IEBGENER	IEBUPDTE

表1. タスクおよびユーティリティー・プログラム (続き)

タスク	オプション	1 次 ユーティリティー	2 次 ユーティリティー
編集およびコピー	ジョブ・ストリーム	IEBEDIT	
	順次データ・セット	IEBGENER	IEBUPDTE, IEBTPCH
編集およびリスト	ボリューム別エラー統計 (ESV) レコード	IFHSTATR	
編集および印刷	順次データ・セット	IEBTPCH	IEBGENER
編集および穿孔	順次データ・セット	IEBTPCH	IEBGENER
入力	プロシージャ・ライブラリーに入るプロシージャ	IEBUPDTE	
除外	コピー操作からの区分データ・セット・メンバー	IEBCOPY	IEHMOVE
	コピー操作からの PDSE メンバー	IEBCOPY	
拡張	区分データ・セットまたは PDSE	IEBCOPY	
	順次データ・セット	IEBGENER	
生成	テスト・データ	IEBDG	
組み込み	メンバーまたは順次データ・セットの変更	IEBUPDTE	
	コピー操作からの区分データ・セット・メンバー	IEBCOPY	IEHMOVE
	コピー操作からの PDSE メンバー	IEBCOPY	
指示	シフトイン・シフトアウト文字で囲んだ 2 バイト文字 セット	IEBGENER	IEBTPCH
レコードの挿入	区分データ・セットまたは PDSE への挿入	IEBUPDTE	
ラベル付け	磁気テープ・ボリューム	IEHINITT	
リスト表示	パスワード入力	IEHPROGM	
	ボリューム目録	IEHLIST	
	未使用のディレクトリー・ブロックおよびトラックの 数	IEHLIST	IEBCOPY
	区分データ・セットまたは PDSE ディレクトリー	IEHLIST	IEHPROGM
	CVOL 項目	IEHLIST	
ロード	アンロードされた区分データ・セットを区分データ・ セットに	IEBCOPY	
	アンロードされるデータ・セット	IEHMOVE	
	アンロードされる区分データ・セットを PDSE に (ロ ードしないモジュールのみ)	IEBCOPY	
	アンロードされる PDSE を区分データ・セットに (ロ ードしないモジュールのみ)	IEBCOPY	
	アンロードされる PDSE を PDSE に	IEBCOPY	
マージ	区分データ・セット	IEBCOPY	IEHMOVE
	PDSE	IEBCOPY	
	区分データ・セットおよび PDSE	IEBCOPY	
変更	区分データ・セット、順次データ・セット、または PDSE	IEBUPDTE	

表1. タスクおよびユーティリティ・プログラム (続き)

タスク	オプション	1 次 ユーティリティ	2 次 ユーティリティ
移動	データ・セットのボリューム	IEHMOVE	
	区分データ・セット	IEHMOVE	
	順次データ・セット	IEHMOVE	
レコード数	区分データ・セットまたは PDSE の新規または今までのメンバーの	IEBUPDTE	
パスワード保護	パスワードの追加	IEHPROGM	
	パスワードの削除	IEHPROGM	
	パスワードのリスト	IEHPROGM	
	パスワードの置換	IEHPROGM	
印刷	順次データ・セット	IEBTPCH	IEBGENER, IEBUPDTE
	区分データ・セットまたは PDSE	IEBTPCH	
	選択レコード	IEBTPCH	
	2 バイト文字セットおよび 1 バイト文字セットの混合 ストリング	IEBTPCH	IEBGENER
	2 バイト文字セット・データ	IEBTPCH	IEBGENER
穿孔	区分データ・セット・メンバー	IEBTPCH	
	順次データ・セット	IEBTPCH	
	選択レコード	IEBTPCH	
	2 バイト文字セットおよび 1 バイト文字セットの混合 ストリング	IEBTPCH	IEBGENER
	2 バイト文字セット・データ	IEBTPCH	IEBGENER
再ブロック	ロード・モジュール	IEBCOPY	
	区分データ・セットまたは PDSE	IEBCOPY	
	順次データ・セット	IEBGENER	IEBUPDTE
再作成	区分データ・セットまたは PDSE	IEBCOPY	
名前変更	区分データ・セットまたは PDSE のメンバー	IEBCOPY	IEHPROGM
	順次データ・セット、区分データ・セット、または PDSE	IEHPROGM	
	区分データ・セットの移動またはコピーしたメンバー	IEHMOVE	
番号変更	論理レコード	IEBUPDTE	
除去	両端のシフトアウト/シフトイン文字を除去することにより 2 バイト文字セットであることを示す	IEBGENER	
置換	パスワード	IEHPROGM	
	論理レコード	IEBUPDTE	
	区分データ・セットまたは PDSE のメンバー内のレ コード	IEBUPDTE	
	PDSE の選択されたメンバー	IEBCOPY	IEBUPDTE
	区分データ・セットの選択されたメンバー	IEBCOPY	IEBUPDTE, IEHMOVE

表1. タスクおよびユーティリティー・プログラム (続き)

タスク	オプション	1 次 ユーティリティー	2 次 ユーティリティー
スクラッチ	データ・セット	IEHPROGM	
アンカタログ	データ・セット	IEHPROGM	
アンロード	区分データ・セット	IEBCOPY	IEHMOVE
	順次データ・セット	IEHMOVE	
	PDSE	IEBCOPY	
同じ場所での更新	区分データ・セットまたは PDSE	IEBUPDTE	

システム・ユーティリティー・プログラム

システム・ユーティリティー・プログラムを使用して、データ・セット名、カタログ項目、ボリューム・ラベルなど、データ・セットおよびボリュームに関連した情報をリスト表示または変更することができます。システム・ユーティリティー・プログラムが実行できる機能のほとんどは、IDCAMS、ISMF、または DFSMSrmm などの他のプログラムを使用することにより、より効果的に実行できます。

表2 に、システム・ユーティリティー・プログラムおよびその目的をリストアップします。

表2. システム・ユーティリティー・プログラム

システム・ ユーティリティー	代替プログラム	目的
*IEHINITT	DFSMSrmm EDGINERS	テープ・ボリュームに標準ラベルを書き込む
IEHLIST	ISMF、PDF 3.4	システム制御データをリスト表示する
*IEHMOVE	DFSMSdss, IEBCOPY	データの集合を移動またはコピーする
IEHPROGM	アクセス方式サー ビス、PDF 3.2	システム制御データを作成および維持する
*IFHSTATR	DFSMSrmm, EREP	IFASMFDP テープからテープ・エラーに関する情報の選択、フォーマット、および書き込みを行う

* これらのプログラムが提供する機能は、より新しいアプリケーション (ISMF、DFSMSrmm、DFSMSdss など) によって、さらに効率的に実行されます。IBM は、サポートされる 古いシステム・レベルとの互換性を保って、これらのプログラムの出荷を継続します。

データ・セット・ユーティリティー・プログラム

データ・セット・ユーティリティー・プログラムを使用すると、データ・セットまたはレコード・レベルでデータの再編成、変更、または比較を行うことができます。これらのプログラムは、JCL ステートメントおよびユーティリティー制御ステートメントによって制御されます。

これらのユーティリティーを使用すると、プログラムへの入力として供給される、区分、順次、または索引順次データ・セット、または拡張区分データ・セット (PDSE) の操作が可能になります。操作可能なデータの範囲は、論理レコード内のフィールドからデータ・セット全体にまで及びます。

本書に含まれるデータ・セット・ユーティリティーを VSAM データ・セットと共に使用することはできません。VSAM データ・セットに関する情報は、*z/OS DFSMS データ・セットの使用法* にあります。

表 3 に、データ・セット・ユーティリティー・プログラムおよびその使用法をリストアップします。

表 3. データ・セット・ユーティリティー・プログラム

データ・セット・ユーティリティー	用途
*IEBCOMPR, SuperC, (PDF 3.12)	順次データ・セット、区分データ・セット、または PDSE 内のレコードを比較。
IEBCOPY	区分データ・セットまたは PDSE のコピー、圧縮、またはマージ。RLD カウント情報をロード・モジュールに追加。コピー操作で指定したメンバーを選択または除外。区分データ・セットまたは PDSE の選択されたメンバーを名前変更または置換。
IEBDG	パターン化されたデータから構成されるテスト・データ・セットを作成。
IEBEDIT	ジョブ・ステップおよび関連した JOB ステートメントの選択的コピー
IEBGENER または ICEGENER	順次データ・セットからのレコードのコピー、または順次編成から区分編成へのデータ・セットの変換
IEBIMAGE	IBM 3800 印刷サブシステム、IBM 3262 モデル 5、または 4248 印刷装置で使用できるように、モジュールを変更、印刷、またはリンクする
IEBPTPCH または PDF 3.1 または 3.6	順次データ・セットまたは区分データ・セット内のレコードの印刷または穿孔
IEBUPDTE	順次データ・セット、区分データ・セット、または PDSE に変更内容を編入

* これらのプログラムが提供する機能は、より新しいアプリケーション (ISMF、DFSMSrmm、DFSMSdss など) によって、さらに効率的に実行されます。IBM は、サポートされる古いシステム・レベルとの互換性を保って、これらのプログラムの出荷を継続します。

制御

システムおよびデータ・セット・ユーティリティー・プログラムは、ジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。ユーティリティー・プログラムを使用するために必要なジョブ制御ステートメントおよびユーティリティー制御ステートメントは、各ユーティリティー・プログラムについての主な解説で示されています。

ジョブ制御ステートメント

以下の方法で、システムまたはデータ・セット・ユーティリティー・プログラムを開始できます。

- ジョブ制御ステートメントをファイルに入れて、TSO SUBMIT コマンドなどによりファイルを JES に渡して実行する。
- プロシージャ・ライブラリー内のジョブ制御ステートメントを入れて、MVS™ オペレーターの START コマンドで実行するか、EXEC ジョブ制御ステートメントでそれらを JOB に組み込む。
- TSO CALL コマンドを使用する。
- CALL、LINK、または ATTACH マクロを使用する別のプログラムを使用する。

本書に示す JCL の例はほとんど、アンカタログされたデータ・セットの位置決めで使用されるパラメーターを指定します。カタログ式データ・セットでは、UNIT および VOL=SER パラメーターは不要です。本書で説明するユーティリティーと共に使用可能な SMS 管理のデータ・セットの割り振りの詳細については、*z/OS DFSMS* データ・セットの使用法 を参照してください。

JCL ステートメントのコーディングの詳細については、*z/OS MVS JCL 解説書* を参照してください。

データ・セットの共用

VSAM データ・セットまたは PDSE の場合を除き、複数のジョブまたはユーザーが同時にデータ・セットを更新すると、そのデータ・セットを損傷する危険があります。一部のデータ・セット、特にシステム・データ・セット（『SYS1』によって識別される）は常に使用できます。共用データ・セットを安全に更新するには、1人のユーザーを除いてすべてのユーザーがデータ・セットの更新を中止しなければなりません。更新が終了した後、すべてのユーザーはデータ・セットに再アクセスする必要があります。データ・セットへの再アクセスはデータ・セットを使用するプログラムの機能で、データ・セットのクローズと再オープン、およびデータ・セットの再割り当てを含むことがあります。これを実行できないプログラムもあるので、共用データ・セットを常に安全に更新できるとは限りません。残念ながら、この逐次化のメカニズムは、データ・セットが異なるボリューム上にあれば、名前が同じでも両者を区別します。

DD ステートメント上の DISP パラメーター、TSO ALLOCATE コマンド、または動的割り振り用の同等のテキスト・ユニット (SVC 99) を使用すれば、データ・セットをロックして更新できるようにすることができます。PDSE 以外のデータ・セットを更新するたびに、DISP=OLD または DISP=MOD を指定します。

JCL に DISP=SHR をコーディングするか、TSO ALLOCATE コマンドまたは動的割り振りに同等のパラメーターをコーディングすると、更新中のデータ・セットは、PDSE でない限り、他のユーザーによって同時に更新されて、使用できなくなる可能性があることに注意してください。ただし、PDS の場合は、例外です。GRSplex 内の第 2 のプログラムが出力のために共用 PDS をオープンしようとする（同じ場所での更新オプションでない場合）、そのプログラムは、213 ABEND を通知されます。このような共用プロトコル違反の検出は、名前は同じだが異なるボリューム上にある 2 つのデータ・セットや、名前とボリューム通し番号は同じだ

が、異なるシステム上にある 2 つのデータ・セットの場合であっても有効です。これらの 2 つの場合には、ABEND 213 が出されることはありません。

この問題が報告されているのは、プログラム管理 (バインダー、リンケージ・エディター)、MVS 割り振り (JCL、SVC 99)、および ISPF/PDF などの一部のコンポーネントに関してです。各コンポーネントにはそれぞれ独自のインターロックがあり、そのどれも他のすべてのインターロックを認識することはできません。したがって、DISP=SHR を指定したデータ・セット割り振りを更新する上で完全に安全な方法はありません。

区分データ・セットの場合はディレクトリーおよび個々のメンバーがあるため、共用の仕方はさらに複雑になります。一般に、これらのサブパーツは、システム内で保護されています (ISPF/PDF では、他の ISPF/PDF ユーザーがメンバーおよびディレクトリーの変更を試みても、適切に保護されます)。

PDSE (拡張区分データ・セット) は共用問題を回避するように設計されています。区分データ・セットの代わりとしてそれを使用することを検討してください。

1 つのボリュームが複数の異なるオペレーティング・システム間で共有される場合 (MVS システムと VM システムの間で DASD を共有する場合など)、DD ステートメントまたは TSO ALLOC コマンドを使用しても、2 つの異なるシステムが、ボリュームを同時に更新できないようにすることは不可能な場合があります。データ・セットの共用に関する詳細に関しては、*z/OS DFSMS データ・セットの使用法* を参照してください。

ユーティリティー制御ステートメント

ユーティリティー制御ステートメントによって、ユーティリティー・プログラムが実行する特定の機能を識別すること、および必要な場合、処理する特定のボリュームまたはデータ・セットを識別することができます。特定のユーティリティーが使用するユーティリティー制御ステートメントは、そのユーティリティーに関する章で説明されています。

ユーティリティー制御ステートメントは通常、入力ストリームに含まれます。しかし、順次データ・セット、区分データ・セットあるいは PDSE のメンバー、または *z/OS UNIX* システム・サービス (*z/OS UNIX*) ファイル (たとえば HFS ファイル) に入れられることもあります。どちらの場合でも、データ・セットには論理レコード長が 80 の固定長レコードまたは固定長ブロック・レコードが必要です。*z/OS UNIX* ファイルの場合は、レコードの長さは、さまざまです。DD ステートメントに FILEDATA=TEXT とコーディングします。

例外: ユーティリティーによっては、この規則に例外が生じます。

ユーティリティー・プログラムの制御ステートメントには、以下の標準フォーマットがあります。

ラベル 操作 オペランド 注釈

ラベル は制御ステートメントをシンボルによって識別するもので、システム・ユーティリティー・プログラム IEHINIT の場合以外は省略できます。これを含める場合、名前はステートメントの先頭位置から開始して、その後には 1 つ以上の空白がなければなりません。ラベルは、1 から 8 文字の英数字を含むことができます。

IEBUPDTE 制御ステートメントはこの規則に対する例外です。その場合、位置 1 および 2 の『./』で開始して、オプション・ラベルは位置 3 から始まります。

操作 は、制御ステートメントのタイプを示します。その前後には 1 つ以上のブランクが必要です。

オペランド は、コンマで区切られた 1 つ以上のキーワード・パラメーターから構成されます。オペランド欄の前後には 1 つ以上のブランクが必要です。コンマ、括弧、およびブランクは、区切り文字としてのみ使用できます。

注釈 はユーティリティー・ステートメント内に書き込むことができますが、1 つ以上のブランクによってオペランド欄の最後のパラメーターから区分しなければなりません。

ユーティリティー制御ステートメントの継続

ユーティリティー制御ステートメントは、1 桁目から 71 桁目の間に入力されます。71 文字を超えるステートメントは、1 つ以上の追加のレコード上で継続します。継続を示すため、72 桁目に非ブランク文字を入れてください。

例外: ユーティリティーによっては、この規則に例外が生じます。この場合、ユーティリティー・ステートメントは、71 桁目、または任意のコンマの後で中断できます。

ユーティリティー制御ステートメントの継続部分は、次のレコードの 16 桁目から開始しなければなりません。

例外: IEBTPCH および IEBGENER ユーティリティー・プログラムでは、これらの要求に対していくつかの例外が認められています (該当するプログラムの解説を参照してください)。

制約事項

- ユーティリティー制御ステートメントでは、アンパーサンドで始まる一時データ・セット名はサポートされていません。システムがデータ・セット用に生成した完全な名前をコーディングすることができます (たとえば、`DSNAME=SYS95296.T000051.RP001.JOBTEMP.TEMPMOD` など)。ただし、データ・セットを識別できるのが、DD ステートメント上または動的割り振りの同等のステートメントに限られるユーティリティーの場合は、アンパーサンドで始まる一時データ・セット名を使用することができます。
- 本書で解説されるユーティリティー・プログラムは通常、VSAM データ・セットをサポートしません。一部の例外については、各種プログラムの解説を参照してください。
- ASCII テープ・データ・セットは、`LABEL=(,AL)` または `OPTCD=Q` によって識別します。

テープ上で 2 進 (非テキスト) データを読み書きするユーティリティーは、ASCII テープをサポートしていません。これは、ASCII テープの場合、すべてのデータがテキストでなければならないからです。z/OS DFSMS データ・セットの使用法を参照してください。ASCII テープをサポートしないユーティリティーには、IEBCOPY、IEHMOVE、および IFHSTATR などがあります。

参照用の特別な工夫

必要なユーティリティー・プログラムを見つけたり、参照のために適切なプログラムの例を見つけたりするために、本書では、参照に役立つ 2 つの特別な工夫を凝らしています。

適切なユーティリティー・プログラムを見つけるために、ユーティリティー・プログラムの機能の紹介にある 1 ページの表 1 を参照してください。

適切な例を見つけるには、各プログラム例の前にある図（「例の一覧表」と呼ばれる）を参照してください。表 4 は、IEBCOPY の例の一覧表の一部を示しています。この図は、IEBCOPY 例 1 が区分データ・セットを 1 つのディスク・ボリュームから他のディスク・ボリュームにコピーする例であること、および例 2 が 3 つの入力区分データ・セットから既存の出力区分データ・セットにコピーする例であることを示しています。

表 4. 例の一覧表

操作	装置	注釈	例
COPY	ディスク	フル・コピー。入力および出力データ・セットは区分されています。	1
COPY	ディスク	複数の入力区分データ・セット。固定長ブロックおよび固定長レコード・フォーマット。	2

第 2 章 IEBCOMPR (データ・セットの比較) プログラム

推奨: SuperC ユーティリティーを IEBCOMPR の代わりに使用してください。SuperC は ISPF/PDF および高水準アセンブラー・ツールキット・フィーチャーの一部です。SuperC は、バッチでもフォアグラウンドでも実行可能で、その報告書はそれ以上に有用です。

IEBCOMPR は 2 つの順次データ・セット、2 つの区分データ・セット、または 2 つの拡張区分データ・セット (PDSE) を論理レコード・レベルで比較して、バックアップ・コピーを検査するためのデータ・セット・ユーティリティーです。ブロック化または非ブロック化データ・セットまたはメンバーからの固定、可変、または不定形式レコードも比較できます。しかし、ロード・モジュールの比較には IEBCOMPR を使用しないでください。

以下の両方の条件を満たした場合に、2 つの順次データ・セットは等しい、つまり同一であると見なされます。

- 両方のデータ・セットに同じ数のレコードがある。
- 対応するレコードおよびキーが等しい。

以下のすべての条件を満たした場合に、2 つの区分データ・セットまたは 2 つの PDSE は等しいと見なされます。

- 対応するメンバーが同じ数のレコードを含んでいる。
- 注釈リストが対応するメンバー内の同じ位置にある。
- 対応するレコードおよびキーが等しい。
- 対応するディレクトリー・ユーザー・データ・フィールドが等しい。

ある特定のタイプのデータ・セットに関して上記のすべての条件が合致するのではなく、それらのデータ・セットは等しくないと見なされます。レコードが等しくない場合、レコードおよびブロック番号、データ・セットを定義する DD ステートメントの名前、および等しくないレコードがメッセージ・データ・セットにリスト表示されます。等しくない比較が連続して 10 回続くと、エラー状態を処理するルーチンが備わっていない限り、ジョブ・ステップは停止します。

異なるタイプの装置上にあるロード・モジュール区分データ・セットを比較することはできません。ほとんどの場合、データ・セットを比較しても等しいとは見なされません。

区分データ・セットまたは PDSE を比較できるのは、1 つまたは両方のディレクトリー内のすべての名前が他方のディレクトリーに対応する項目を持つ場合だけです。それらの項目および対応するユーザー・データによって示されるメンバーが比較されます。

12 ページの図 1 は 2 つの区分データ・セットのディレクトリーを示しています。ディレクトリー 2 にはディレクトリー 1 内のすべての名前に対応する項目が含まれるので、データ・セットを比較することができます。

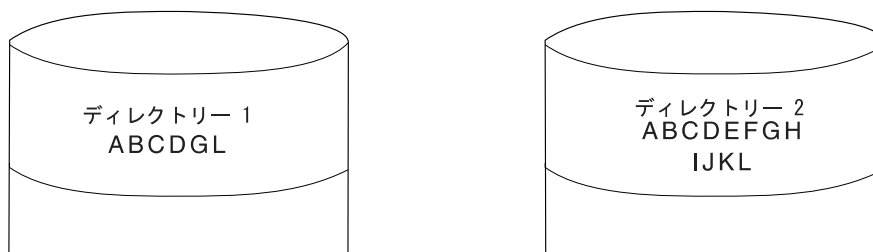


図1. IEBCOMPR を使用して比較できる区分ディレクトリー

図2 は 2 つの区分データ・セットのディレクトリーを示しています。各ディレクトリーには他方のディレクトリーに対応する項目のない名前が含まれています。そのため、データ・セットを比較することはできず、ジョブ・ステップは停止します。

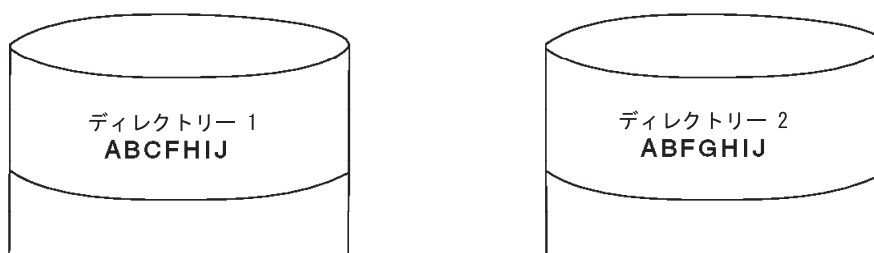


図2. IEBCOMPR を使用して比較できない区分ディレクトリー

オプションのユーザー・ルーチンがユーザー・ラベルの生成、エラー状態の処理、およびソース・レコードの変更を行うためのユーザー出口が備わっています。ユーザー・ルーチンの使用時に適用するリンケージ規則についての説明は、377 ページの『付録 C. ユーティリティ・プログラムのユーザー出口を指定する』を参照してください。

IEBCOMPR をアプリケーション・プログラムまたは TSO から起動した場合、IEBCOMPR を呼び出す前に、動的割り振り (SVC 99) または TSO ALLOCATE コマンドを呼び出すことによって、データ・セットを動的に割り振ることができます。

入出力

IEBCOMPR は以下の入力を使用します。

- 比較する 2 つの順次データ・セット、区分データ・セット、PDSE、または USS ファイル。
- ユーティリティ制御ステートメントを含む制御データ・セット。入力データ・セットが区分または PDSE である場合、またはユーザー・ルーチンが使用される場合にこのデータ・セットが必要です。

IEBCOMPR は出力として、情報メッセージ (ユーティリティ制御ステートメントの内容など)、比較の結果、およびエラー・メッセージを生成します。

IEBCOMPR の戻りコードについては、付録 Aを参照してください。

制御

IEBCOMPR はジョブおよびユーティリティー制御ステートメントによって制御されます。ジョブ制御ステートメントは IEBCOMPR の処理および IEBCOMPR によって使用および生成されたデータ・セットの定義に必要です。ユーティリティー制御ステートメントを使用して、入力データ・セットの編成 (順次、区分、または PDSE) を指定すること、供給されるユーザー・ルーチンを識別すること、およびユーザー・ラベルをデータとして処理することを指定することができます。

ジョブ制御ステートメント

表 5 は、IEBCOMPR で使用するジョブ制御ステートメントをまとめたものです。

表 5. IEBCOMPR で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEBCOMPR)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。
SYSPRINT DD	IEBCOMPR が生成するメッセージに使用される順次データ・セットを定義します。このデータ・セットは、システム出力装置 (SYSOUT)、テープ・ボリューム、直接アクセス・ボリューム、TSO 端末、またはダミー (DUMMY DD) に書き込むことができます。
SYSUT1 DD	比較する入力データ・セットまたは USS ファイルを定義します。
SYSUT2 DD	比較する入力データ・セットまたは USS ファイルを定義します。
SYSIN DD	制御データ・セットを定義するか、または入力データ・セットが順次でユーザー・ルーチンが使用されない場合は DUMMY を指定します。制御データ・セットは通常、入力ストリーム内に置かれます。しかし、区分メンバーのライブラリーにあるメンバーとして定義することも可能です。

入力データ・セットの 1 つまたは両方は、先行するジョブ・ステップから渡すことができます。

異なる装置タイプ上に存在するデータ・セットを比較することができます。しかし、異なる装置タイプ上に存在するロード・モジュール・ライブラリーを比較することはできません。さらに、異なる密度で記述された順次データ・セットを比較することも可能です。

IEBCOMPR を使用するたびに SYSPRINT DD ステートメントが存在しなければなりません。SYSPRINT DD ステートメントで指定されるブロック・サイズは、121 の倍数でなければなりません。

SYSIN DD ステートメントが必要です。SYSIN DD ステートメントで指定されるブロック・サイズは、80 の倍数でなければなりません。

それぞれの入力データ・セットの論理レコード長は同じでなければなりません。同じでない場合、2 つのデータ・セットの比較結果は等しくないことになります。入

カデータ・セットのブロック・サイズは異なっていても構いません。固定長ブロック (FB) データ・セットでは、ブロック・サイズは論理レコード長の倍数でなければなりません。ブロック・サイズは 32760 バイトを超えてはなりません。

ユーティリティー制御ステートメント

表 6 には、IEBCOMPR を制御するのに使われるユーティリティー制御ステートメントが示されています。

表 6. IEBCOMPR で使用するユーティリティー制御ステートメント

ステートメント	用途
COMPARE	データ・セットの編成を示します。
EXITS	使用するユーザー出口ルーチンを識別します。
LABELS	ユーザー・ラベルが IEBCOMPR によってデータとして扱われるかどうかを示します。

ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

COMPARE Statement

COMPARE ステートメントを使用して、比較したいデータ・セットの編成を指定できます。

COMPARE ステートメントを含める場合、それを最初のユーティリティー制御ステートメントとしなければなりません。EXITS または LABELS ステートメントを使用する場合、または入力データ・セットが区分データ・セットか PDSE である場合、COMPARE が必要です。

COMPARE ステートメントの構文は次のとおりです。

[label]	COMPARE	TYPORG={PS PO}
---------	---------	----------------

各項目の意味は以下のとおりです。

TYPORG={PS|PO}

入力データ・セットの編成を示します。コーディングできる値は以下のとおりです。

PS 入力データ・セットが順次データ・セットであることを指定します。これがデフォルトです。

PO

入力データ・セットが区分データ・セットまたは PDSE であることを指定します。

EXITS ステートメント

EXITS ステートメントを使用して、使用したい出口ルーチンを指定できます。出口ルーチンを使用する場合、EXITS ステートメントが必要です。複数の EXITS ステートメントを使用した場合、IEBCOMPR が使用するのは最後の EXITS ステート

メントだけです。その他はすべて無視されます。ユーザー・ラベルをデータ・セット記述子として処理することに関する説明は、382 ページの『ユーザー・ラベルの処理』を参照してください。

EXITS ステートメントの構文は次のとおりです。

[label]	EXITS	[INHDR=routinename] [,INTLR=routinename] [,ERROR=routinename] [,PRECOMP=routinename]
---------	-------	-----------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

INHDR=routinename

ユーザー入力ヘッダー・ラベルを処理するルーチンの名前を指定します。

INTLR=routinename

ユーザー入力トレーラー・ラベルを処理するルーチンの名前を指定します。

ERROR=routinename

等しくない比較の後にエラー処理のため制御を受け継ぐルーチンの名前を示します。このパラメーターが省略されていて、IEBCOMPR が順次データ・セットを比較しているときに等しくない比較が連続して 10 回生じた場合、処理は停止します。入力データ・セットが区分または PDSE の場合、次のメンバーから処理が続行します。

PRECOMP=routinename

入力データ・セットを比較する前に、その片方または両方からの論理レコード (可変長スパン (VS) または可変長ブロック・スパン (VBS) レコードが 32k バイトより長い場合は物理ブロック) を処理するルーチンの名前を指定します。

LABELS ステートメント

LABELS ステートメントを使用して、IEBCOMPR がラベル・データとして処理するようにするかどうかを指定します。このオプションについての説明は、382 ページの『ユーザー・ラベルの処理』を参照してください。

複数の LABELS ステートメントを使用した場合、IEBCOMPR が使用するのは最後の LABELS ステートメントだけです。その他はすべて無視されます。

表 7 は、LABELS ステートメントの構文規則を示しています。

表 7. LABEL ステートメントの構文

[label]	LABELS	[DATA={YES NO ALL ONLY}]
---------	--------	--------------------------

DATA={YES|NO|ALL|ONLY}

ユーザー・ラベルをデータとして扱うかどうかを指定します。コーディングできる値は以下のとおりです。

YES

ユーザーのラベル処理ルーチンによってリジェクトされないユーザー・ラベ

ルをすべてデータとして処理することを指定します。ラベルをデータとして処理する場合、その処理は標準の戻りコードに従って停止します。デフォルトは YES です。

NO

ユーザー・ラベルをデータとして扱わないことを指定します。

ALL

すべてのユーザー・ラベルをデータとして扱うことを指定します。戻りコードが 16 の場合、IEBCOMPR はユーザー・ラベルのグループの残りに対する処理を完了してからジョブ・ステップを終了します。

ONLY

ユーザー・ヘッダー・ラベルだけをデータとして扱うことを指定します。ユーザー・ヘッダー・ラベルは戻りコードに関係なく、データとして処理されます。ジョブは OPEN ルーチンから戻った時点で終了します。

要件: 標準外ラベル (NSL) のある入出力データ・セットを処理するときに IBM 標準ユーザー・ラベル (SUL) 出口を非アクティブにするためには、LABELS DATA=NO を指定しなければなりません。

IEBCOMPR の例

表 8 の各例では、IEBCOMPR の使用方法の一部を示します。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

実際の装置名または装置番号を指定するべきところで **disk** または **tape** を使用している例は、使用する前に変更する必要があります。実際の装置名または装置番号は、システムに対する装置定義の方法によって異なります。

表 8. IEBCOMPR の例の一覧表

操作	データ・セット編成	装置	注釈	例
COMPARE	区分	ディスク	ユーザー・ルーチンなし。ブロック化入力。	6
COMPARE	PDSE	ディスク	ユーザー・ルーチンなし。SMS 管理のデータ・セット。	8
COMPARE	順次	9 トラックの磁気テープ	ユーザー・ルーチンなし。ブロック化入力。	1
COMPARE	順次	7 トラックの磁気テープ	ユーザー・ルーチンなし。ブロック化入力。	2
COMPARE	順次	7 トラックの磁気テープおよび 9 トラックの磁気テープ	ユーザー・ルーチン。ブロック化入力。異なる密度の磁気テープ。	3
COMPARE	順次	システム入力ストリーム、9 トラックの磁気テープ	ユーザー・ルーチンなし。ブロック化入力。	4

表 8. IEBCOMPR の例の一覧表 (続き)

操作	データ・セット編成	装置	注釈	例
COPY (IEBGENER を使 用) および COMPARE	順次	ディスクまたは 磁気テープ	ユーザー・ルーチンなし。ブロック化 入力。2 つのジョブ・ステップ。デー タ・セットは 2 番目のジョブ・ステッ プに渡されます。	5
COPY (IEBCOPY を使用) および COMPARE	区分	ディスク	ユーザー・ルーチン。ブロック化入 力。2 つのジョブ・ステップ。デー タ・セットは 2 番目のジョブ・ステッ プに渡されます。	7

例 1: 磁気テープ上のデータ・セットを比較する

この例では、9トラックのテープ・ボリューム上にある 2 つの順次データ・セットを比較します。

```
//TAPETAPE JOB ...
//          EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=SET1,UNIT=tape,LABEL=(,NL),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000),
//          DISP=(OLD,KEEP),VOLUME=SER=001234
//SYSUT2   DD DSN=SET2,UNIT=tape,LABEL=(,NL),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1040),
//          DISP=(OLD,KEEP),VOLUME=SER=001235
//SYSIN    DD DUMMY
/*
```

各ジョブ制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ラベルのない 9トラックのテープ・ボリューム上にある入力データ・セット (SET1) を定義しています。
- SYSUT2 DD では、ラベルのない 9トラックのテープ・ボリューム上にある入力データ・セット (SET2) を定義しています。
- SYSIN DD では、ダミー・データ・セットを定義しています。使用されるユーザー・ルーチンがなく、入力データ・セットが順次編成であるため、ユーティリティ制御ステートメントは必要ありません。

例 2: 磁気テープ上の順次データ・セットを比較する

この例では、7トラックのテープ・ボリューム上にある 2 つの順次データ・セットを比較します。

```
//TAPETAPE JOB ...
//          EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=SET1,LABEL=(2,SUL),DISP=(OLD,KEEP),
//          VOL=SER=001234,DCB=(DEN=2,RECFM=FB,LRECL=80,
//          BLKSIZE=2000,TRTCH=C),UNIT=tape
//SYSUT2   DD DSN=SET2,LABEL=(,SUL),DISP=(OLD,KEEP),
//          VOL=SER=001235,DCB=(DEN=2,RECFM=FB,LRECL=80,
//          BLKSIZE=2000,TRTCH=C),UNIT=tape
//SYSIN    DD *
//          COMPARE TYPORG=PS
//          LABELS DATA=ONLY
/*
```


各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、ラベルのある 7 トラックのテープ・ボリュウム上にある入力データ・セット (SET1) を定義しています。ブロック化データ・セットの初期の書き込み密度は 800 ビット / インチ (DEN=2) で、データ・コンバーターはオン (TRTCH=C) です。
- **SYSUT2 DD** では、ラベルのある 7 トラックのテープ・ボリュウム上にある最初または唯一の入力データ・セット (SET2) を定義しています。ブロック化データ・セットの初期の書き込み密度は 800 ビット / インチ (DEN=2) で、データ・コンバーターはオン (TRTCH=C) です。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **COMPARE TYPORG=PS** は、入力データ・セットの編成方式が順次であることを指定します。
- **LABELS DATA=ONLY** は、ユーザー・ヘッダー・ラベルがデータとして扱われ、比較されるように指定します。磁気テープ上の他のすべてのラベルは無視されます。

例 3: 異なる密度で書き込まれた順次データ・セットを比較する

この例では、異なる磁気テープ装置上に異なる密度で書き込まれた 2 つの順次データ・セットを比較します。

```
//TAPETAPE JOB ...
//          EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SET1,LABEL=(,SUL),DISP=(OLD,KEEP),
//          VOL=SER=001234,DCB=(DEN=1,RECFM=FB,LRECL=80,
//          BLKSIZE=320,TRTCH=C),UNIT=tape
//SYSUT2 DD DSN=SET2,LABEL=(,SUL),DISP=(OLD,KEEP),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=640),
//          UNIT=tape,VOLUME=SER=001235
//SYSIN DD *
          COMPARE TYPORG=PS
          EXITS INHDR=HDRS,INTLR=TLRS
          LABELS DATA=NO
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、ラベルのある 7 トラックのテープ・ボリュウム上にある最初または唯一の入力データ・セット (SET1) を定義しています。ブロック化データ・セットの初期の書き込み密度は 556 ビット / インチ (DEN=1) で、データ・コンバーターはオン (TRTCH=C) です。
- **SYSUT2 DD** では、ラベルのあるテープ・ボリュウム上にある最初または唯一のブロック化データ・セット (SET2) を定義しています。この例では、SYSUT2 が 9 トラックの磁気テープ装置にあると想定しています。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **COMPARE TYPORG=PS** は、入力データ・セットの編成方式が順次であることを指定します。
- **EXITS** は、ユーザー入力ヘッダー・ラベルおよびトレーラー・ラベルの処理に使用するルーチンの名前を識別します。
- **LABELS DATA=NO** は、データ・セットごとのユーザー入力ヘッダーおよびトレーラー・ラベルを比較しないように指定します。

例 4: 順次データ・セットを比較する — 入力ストリームおよび磁気テープ入力

この例では、2 つの順次データ・セット (入力ストリームおよび磁気テープ入力) を比較します。

```
//CARDTAPE JOB ...
//          EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT2   DD UNIT=tape,VOLUME=SER=001234,LABEL=(,NL),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000),
//          DISP=(OLD,KEEP)
//SYSUT1   DD DATA
```

(入力データ・セット)

/*

各制御ステートメントについての説明は以下のとおりです。

- SYSIN DD では、ダミー制御データ・セットを定義しています。ユーザー・ルーチンがなく、入力データ・セットが順次編成であるため、ユーティリティー制御ステートメントは必要ありません。
- SYSUT2 DD では、ラベルのないテープ・ボリューム上にある入力データ・セットを定義しています。
- SYSUT1 DD では、システム入力ストリーム・データ・セットを定義しています。

例 5: 2 つのジョブ・ステップ内の順次データ・セットをコピーして比較する

この例では、2 つのジョブ内の順次ディスクまたは磁気テープ・データ・セットをコピーして比較します。

```
//TAPETAPE JOB ...
//STEP A    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=WAREHOUS.COPYSET1,DISP=(OLD,PASS),
//SYSUT2   DD DSN=WAREHOUS.COPYSET2,DISP=(,PASS),LABEL=(,SL),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=640),
//          UNIT=tape,VOLUME=SER=001235
//SYSIN    DD DUMMY
//STEP B    EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=*.STEP A.SYSUT1,DISP=(OLD,KEEP)
//SYSUT2   DD DSN=*.STEP A.SYSUT2,DISP=(OLD,KEEP)
//SYSIN    DD DUMMY
```

最初のジョブ・ステップはデータ・セットをコピーして、元のデータ・セットおよびコピーしたデータ・セットを 2 番目のジョブ・ステップに渡します。2 番目のジョブ・ステップはその 2 つのデータ・セットを比較します。

IEBCOMPR ジョブ・ステップの各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、先行するジョブ・ステップ (COPYSET1) から渡される入力データ・セットを定義しています。データ・セットはラベル付きテープ・ボリューム上にあります。

- SYSUT2 DD では、先行するジョブ・ステップ (COPYSET2) から渡される入力データ・セットを定義しています。先行するジョブ・ステップで作成されたデータ・セットは、ラベル付きテープ・ボリューム上にあります。
- SYSIN DD では、ダミー制御データ・セットを定義しています。入力が順次で、ユーザー出口がないため、ユーティリティー制御ステートメントは必要ありません。

例 6: 2 つの区分データ・セットを比較する

この例では、2 つの区分データ・セットを比較します。

```
//DISKDISK JOB ...
//STEP1 EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=PDSSET1,UNIT=disk,DISP=SHR,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000),
//          VOLUME=SER=111112
//SYSUT2 DD DSNAME=PDSSET2,UNIT=disk,DISP=SHR,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000),
//          VOLUME=SER=111113
//SYSIN DD *
          COMPARE TYPORG=PO
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、入力区分データ・セット、PDSSET1 を定義しています。ブロッック化データ・セットはディスク・ボリューム上にあります。
- SYSUT2 DD では、入力区分データ・セット、PDSSET2 を定義しています。ブロッック化データ・セットはディスク・ボリューム上にあります。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- COMPARE TYPORG=PO は、入力データ・セットが区分編成であることを指定します。

例 7: 2 つのジョブ・ステップ内の区分データ・セットをコピーして比較する

この例では、2 つのジョブ内の区分データ・セットをコピーして比較します。

```
//DISKDISK JOB ...
//STEPA EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=MAINDB.LOG.OLDSET,DISP=SHR
//SYSUT2 DD DSNAME=NEWMEMS,UNIT=disk,DISP=(,PASS),
//          VOLUME=SER=111113,SPACE=(TRK,(5,5,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=640)
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN DD *
          COPY OUTDD=SYSUT2,INDD=SYSUT1
          SELECT MEMBER=(A,B,D,E,F)
/*
//STEPB EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=OLDSET,DISP=(OLD,KEEP)
//SYSUT2 DD DSNAME=NEWMEMS,DISP=(OLD,KEEP)
//SYSIN DD *
          COMPARE TYPORG=PO
          EXITS ERROR=SEEERROR
/*
```

最初のジョブ・ステップはデータ・セットをコピーして、元のデータ・セットおよびコピーしたデータ・セットを 2 番目のジョブ・ステップに渡します。2 番目のジョブ・ステップはその 2 つのデータ・セットを比較します。

IEBCOMPR ジョブ・ステップの各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、先行するジョブ・ステップから渡されるブロック化入力データ・セット (MAINDB.LOG.OLDSET) を定義しています。データ・セットはディスクまたはテープ・ボリューム上にあります。
- **SYSUT2 DD** では、先行するジョブ・ステップから渡されるブロック化入力データ・セット (MAINDB.LOG.NEWMEMS) を定義しています。このデータ・セットはディスク・ボリューム上にあります。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **COMPARE TYPORG=PO** は、区分編成を指定します。
- **EXITS** は、ユーザー・エラー・ルーチン **SEEERROR** を使用するように指定します。

入力データ・セット名は同一ではないため、データ・セットをデータ・セット名によって検索できます。

例 8: 2 つの PDSE を比較する

この例では、2 つの PDSE を比較します。

```
//DISKDISK JOB ...
//          EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=PDSE1,DISP=SHR
//SYSUT2   DD DSN=PDSE2,DISP=SHR
//SYSIN    DD *
           COMPARE TYPORG=PO
/*
```

PDSE は、もはや SMS によって管理される必要はありません。これらの PDSE はカタログ化されているので、UNIT パラメーターまたは VOLUME パラメーターの指定は不要です。

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** および **SYSUT2 DD** では、入力 PDSE として PDSE1 および PDSE2 を定義しています。DCB 値が指定されていないので、データ・セットを作成するために指定された DCB 値が使用されます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **COMPARE TYPORG=PO** は、入力データ・セットが PDSE であることを指定します。

第 3 章 IEBCOPY (ライブラリーのコピー) プログラム

IEBCOPY は 1 つまたは複数の区分データ・セット、または拡張区分データ・セット (PDSE) の間で、メンバーの全体または一部をコピーまたはマージするために使用するデータ・セット・ユーティリティです。さらに IEBCOPY を使用して区分データ・セットのバックアップを順次データ・セットに作成 (アンロード・データ・セットまたは PDSU と呼ばれる) すること、およびバックアップから区分データ・セットにメンバーをコピーすることができます。

IEBCOPY を使用して、以下のタスクを実行できます。

- 区分データ・セットまたは PDSE のコピーを作成すること。
- 区分データ・セットをマージすること (アンロード時を除く)。
- バックアップまたは移送のために、区分データ・セットまたは PDSE の順次形式のものを作成すること。
- PDSU から区分データ・セットまたは PDSE に 1 つまたは複数のメンバーを再ロードすること。(アンロードされたロード・モジュールは、この処理では、プログラム・オブジェクトに変換できないことに注意してください。)
- コピー、ロード、またはアンロードする区分データ・セットまたは PDSE の特定のメンバーを選択すること。
- 区分データ・セットまたは PDSE のメンバーを置き換えること。
- コピーするときに、区分データ・セットまたは PDSE の選択されたメンバーを名前変更すること。
- コピー、アンロード、またはロードするデータ・セットからメンバーを除外すること。(COPYGRP の場合を除く。)
- 区分データ・セットをその場所で圧縮すること。
- MVS プログラム取り出しによりロード・モジュールをアップグレードして、より高速のロードを可能にすること。
- ロード・モジュールをコピーおよび再ブロックすること。
- 区分データ・セットを PDSE にコピーするとき、区分データ・セット内のロード・モジュールを PDSE 内のプログラム・オブジェクトに変換すること。
- 区分データ・セットを PDSE に、または PDSE を区分データ・セットに変換すること。
- PDSE データ・セットから、または PDSE データ・セットに、メンバーおよび別名をグループとして一緒にコピーすること (COPYGRP)。

さらに、出力データ・セットが区分データ・セットの場合、IEBCOPY はメンバー・レコードが使用できる未使用のディレクトリー・ブロック数および未使用のトラック数を自動的にリストします。

注： 7 ページの『ジョブ制御ステートメント』に共用区画データ・セットについての重要な情報が記載されています。

ロード・モジュールをプログラム・オブジェクトに、またはその逆に変換する

プログラム・オブジェクトはロード・モジュールが PDSE にコピーされるときに自動的に作成されます。同様に、プログラム・オブジェクトは区分データ・セットにコピーされるときに自動的にロード・モジュールに変換されて戻されます。一部のプログラムは、ロード・モジュール内に存在しないプログラム・オブジェクトの機能を使用するため、ロード・モジュールに変換できないことに注意してください。プログラム・オブジェクトとロード・モジュールの相違点の詳細については、*z/OS MVS プログラム管理: ユーザーズ・ガイド*および*解説書*を参照してください。

PDSE または区分データ・セットをロードまたはアンロードする場合、IEBCOPY はプログラム・オブジェクトとロード・モジュールとの間で直接の変換を行うことはできません。ロード操作によって実行できるのは、ロード・モジュールを区分データ・セットに再ロードすること、またはプログラム・オブジェクトを PDSE に再ロードすることだけです。区分データ・セットをアンロードするとき、ロード・モジュールはアンロード・データ・セット内にだけ入れられます。同様に、PDSE をアンロードするとき、プログラム・オブジェクトはアンロード・データ・セット内にだけ入れられます。

したがって、アンロードされたロード・モジュールをプログラム・オブジェクトに変換するには、ロード・モジュールを区分データ・セットに再ロードしてから、その区分データ・セットを PDSE にコピーします。プログラム・オブジェクトをアンロードされたロード・モジュールに変換するには、PDSE を区分データ・セットにコピーしてから、その区分データ・セットをアンロードします。

区分データ・セットにロード・モジュールおよびデータ・メンバーの両方が含まれる場合、上で説明した方法で区分データ・セットをプログラム・オブジェクト用およびデータ・メンバー用の 2 つの異なる PDSE に変換しなければなりません。

区分データ・セットを PDSE に変換する

IEBCOPY を使用して区分データ・セットを PDSE に変換できます。

区分データ・セットを PDSE に変換するには、PDSE を作成して区分データ・セットをその新しい PDSE にコピーします。これは IEBCOPY の 1 回の使用で実現できます。

以下の特徴を持つ区分データ・セットを変換することはできません。

- ロード・モジュールと非ロード・モジュールが同じ区分データ・セット内にあるもの。区分データ・セットの個々のメンバーは、それらを 2 つの別個の PDSE (データ用およびプログラム・オブジェクト用) にコピーすることによって変換できます。
- 注釈リスト。注釈リストを含むロード・モジュールは、自動的にプログラム・オブジェクトに変換されるので PDSE に入れることができます。
- メンバー内にゼロ以外の長さのキーがあるもの。

データ・セットのコピー

IEBCOPY を使用して、区分データ・セットを 1 つの直接アクセス・ボリュームから別の直接アクセス・ボリュームにコピーできます。さらに、データ・セットの名前を変更すれば、データ・セットが存在するそのボリュームにコピーすることもできます。(データ・セット名を変更しない場合、IEBCOPY はその要求を同所圧縮と解釈します。この場合、SELECT / EXCLUDE ステートメントはコーディングされていないものと仮定します。)

IEBCOPY を使用して PDSE を PDSE にコピーする場合、ボリュームからボリュームへのコピーでもそれが存在するボリュームへのコピーでも、すべての DDM 属性も同様にコピーされます。

区分データ・セットにコピーされるメンバーは物理的に再配列されるのではありません。メンバーは、元のデータ・セットと物理的に同じ順序になるようにコピーされます。

データ・セットをマージする

データ・セットのマージは、追加のメンバーを既存の区分データ・セットにコピーまたはロードすることによって行えます。マージ操作 (出力データ・セットのディレクトリーの順序付け) は、IEBCOPY によって自動的に実行されます。

区分データ・セット用のディレクトリー・スペースを拡張する

IEBCOPY は、区分データ・セット内のディレクトリー・ブロックの数を増加させることができません。(PDSE ディレクトリーは必要に応じて自動的に拡張します。) マージ先の出力区分データ・セットに十分なディレクトリー・ブロックがあるかどうか分からない場合、マージ操作の前に出力データ・セットのディレクトリー・スペースを拡張してください。

IEHLIST を使用して、区分データ・セット内に残っているディレクトリー・スペースを判別します。さらにブロックが必要な場合、以下の手順によりほとんどのデータ・セットを拡張できます。

1. データ・セットを名前変更する。
2. 新規のデータ・セットに十分なスペースおよびディレクトリー・ブロックを割り振る。
3. 名前変更したデータ・セットを新規に割り振ったデータ・セットにコピーする。
4. 名前変更したデータ・セットを削除する (またはバックアップとして保管する)。

データ・セットをアンロード (バックアップ) する

IEBCOPY を使用して、区分データ・セットを DASD、磁気テープ、または QSAM によってサポートされる他の装置上の順次データ・セットにコピー (アンロード) することにより、その区分データ・セットのバックアップ・コピーを作成できます。

出力データ・セットに物理順次編成 (DSORG=PS) を指定すると、IEBCOPY はアンロード・データ・セットを作成します。区分データ・セットを作成するには、DSORG=PO および DSNTYPE=PDS または DSNTYPE=LIBRARY を指定します。

要件: DSORG を明示的に指定しない場合、ACS ルーチンまたは他の JCL パラメーターによって意図されなかった設定になってしまうことがあります。たとえば、

LIKE= または DCB= またはモデル DSCB を指定して DSORG を省略した場合、参照先オブジェクトにある DSORG が暗黙的に DD ステートメントに追加されます。参照先オブジェクトから何が取得されるかが不明の場合は、必ず DSORG を指定してください。

重要: アンロード・データ・セットの DCB パラメーターを IEBCOPY による作成後に変更しないでください。変更した場合、IEBCOPY による再ロードができなくなる場合があります。

IEBCOPY の 1 回の実行で複数の区分データ・セットを同じテープ・ボリュームにアンロードするには、複数のコピー操作を使用して、磁気テープ上の連続したファイル上に複数の順次データ・セットを割り振る必要があります。

IEBCOPY を使用すると、PDSU を PDS に再ロードしてからその PDS をアンロードして新規の PDSU を作成するのではなく、直接 PDSU ディレクトリーにコピーできます。選択的なコピーが必要であれば、IEBGENER を使用して PDSU を新規の PDSU にコピーする方が高速に行えます。

アンロード・データ・セットを作成できるのは COPY 操作だけです。COPYMOD では作成できません。

ディレクトリー情報を区分データ・セットと PDSE との間でコピーする

PDSE ディレクトリーには、今まで区分データ・セットのディレクトリー項目に含まれていたものに加えて、属性を含めることができます。

一部の PDSE 拡張属性はアンロード・データ・セットに記録されて、ターゲットが PDSE のときに再ロードされます。

プログラム・オブジェクトを含むアンロードされた PDSE を区分データ・セットに再ロードする場合、エラー・メッセージが出されて操作は失敗します。

区分データ・セット・ディレクトリーにユーザー・データとして保管されている情報 (PDF 統計値など) は PDSE ディレクトリーに移動されて、変更されることなく、PDSE ディレクトリーから区分データ・セット・ディレクトリーに戻されます。

アンロード・データ・セットをロードまたはコピーする

IEBCOPY を使用して、順次 (アンロード) データ・セットを区分データ・セットにコピーすることにより、区分データ・セットのアンロードされたコピーから区分データ・セットを再作成することができます。プログラム・オブジェクトを含まない区分データ・セットはアンロードされた PDSE からロードすることができ、ロード・モジュールを含まない PDSE はアンロードされた区分データ・セットからロードすることができます。

注: ロード操作によってアンロードされたロード・モジュールをプログラム・オブジェクトに変換したり、プログラム・オブジェクトをアンロードされたロード・モジュールに変換したりすることはできません。

単一の区分データ・セットを複数の (アンロードされた) 入力順次データ・セットから作成できます。

アンロードまたはロード操作では、コピー操作と同じ方法で要求が扱われます。特定のメンバーを選択して処理すること、およびそれらを名前変更することができます。

アンロード・フォーマットの区分データ・セットのレコード・フォーマットは、可変長スパンとなります。アンロード・データ・セットが続いてロードされる時、データ・セットを再ロードするとき特性の指定変更をするのでなければ、出力データ・セットの特性はアンロード操作の前と同じになります。

IEBCOPY アンロード・データ・セットの DCB パラメーター

アンロード・データ・セットは常に、順次編成の可変長スパン・レコード・フォーマット (RECFM=VS および DSORG=PS) です。

アンロード・データ・セットの論理レコード長は、入力データ・セットからのブロックおよび見出しを含むためのものです。以下の考慮事項があります。

1. LRECL は以下のうち大きい方の値となります。
 - a. 280 バイト。
 - b. 16 バイト + ブロック・サイズ + 入力データ・セットのキー長。
2. LRECL が 32760 を超える場合、それは 32760 に縮小されます。

注: アンロード・データ・セットを読み取るアプリケーションは、RECFM=VS データ・セットの場合に LRECL がこのステップによって 32760 バイトに縮小されても、アセンブルされた論理レコードの実際の長さが 32760 バイトを超過する可能性があることを認識している必要があります。

注: この入力データ・セットが PDSE で、そのブロック・サイズが 32744 よりも大きい場合、アンロード・データ・セットの LRECL が X (例えば、LRECL=32768) に設定されます。LRECL=X を使用して、32760 バイトを超える論理レコードを処理します。

3. ユーザーがここで算出された値よりも大きな LRECL を指定した場合、それはデータ・セット・ラベルに入れられます。しかし、IEBCOPY が作成する論理レコードのサイズは増加しません。

アンロード・データ・セットのブロック・サイズ (BLKSIZE) は、以下のステップによって決まります。

1. 初期ブロック・サイズはユーザーが指定するブロック・サイズに設定されます。ユーザーがブロック・サイズを指定しない場合、LRECL + 4 として算出されず。
2. ブロック・サイズが 284 より小さい場合、それは 284 に増加されます。
3. ブロック・サイズが 32760 を超える場合、それは 32760 に縮小されます。
4. その後、ブロック・サイズが出力装置で使用可能な最大ブロック・サイズと比較されます。出力装置の能力がブロック・サイズよりも小さい場合、ブロック・サイズは出力装置で使用可能な最大値に設定されます。

アンロード・データ・セットは非ブロック化されているため、ブロック・サイズを LRECL + 4 より大きな値に増加させても、物理レコードが長くなったり使用効率が向上したりすることはありません。

ブロック・サイズは最初の制御レコード (COPYR1) に保管されて、ロード時に使用されます。アンロード・データ・セットのブロック・サイズが作成後に変更される場合、IEBCOPY によって再ロードできなくなることがあります。

推奨: PDSU ブロック・サイズを PDS ブロック・サイズと同じにしないでください。同じにした場合、PDSU のスペース使用効率およびパフォーマンスは非常に悪くなります。IEBCOPY がブロック・サイズを決めるようにするか、または PDSU ブロック・サイズを PDS ブロック・サイズよりも 20 バイト大きくします。

367 ページの『付録 B. アンロード区分データ・セット・フォーマット』には、アンロードされるデータ・セット形式を記載してあります。

コピー、アンロード、またはロードするメンバーを選択する

SELECT ステートメントをコーディングしてメンバーを指名することにより、1 つまたは複数のデータ・セットから処理する特定のメンバーを選択します。または、EXCLUDE ステートメントをコーディングして処理しないメンバーを指名することにより、特定のメンバーを除くすべてのメンバーを指定します。

同一のコピー操作 (入力 DD 名の同じセット) で、SELECT および EXCLUDE ステートメントの両方を使用することはできません。

コピー操作では、メンバー名または別名として最大 8 文字を使用できます。

選択したメンバーは、SELECT ステートメントで指定したシーケンスには関係なく、昇順 (a から z へ) の照合シーケンスで検索されます。しかし、コピーは入力区分データ・セットまたは PDSE でのシーケンスと同じ物理的なシーケンスで行われます。

SELECT ステートメントで指定したメンバーが入力データ・セットで見つかり、それ以降の入力データ・セットでは検索は行われません。選択したメンバーがすべて見つかり、データ・セットがすべてでは処理されていなくても、操作は終了します。

例:

メンバー A および B が指定されて、A が 3 つの入力データ・セットの最初のもので見つかった場合、2 番目および 3 番目のデータ・セットを検索するときに A が再び検索されることはありません。

B が 2 番目の入力データ・セットで見つかった場合、2 番目の入力データ・セットが処理された後に操作は正常に終了して、3 番目の入力データ・セットが調べられることはありません。

コピー時にメンバーをリネーム可能です。ただし、入力と出力データ・セットが同じ場合は除外します。

コピー操作からメンバーを除外する

コピー、アンロード、またはロード操作に関して、1 つまたは複数の入力データ・セットからメンバーを除外することができます。除外したメンバーはコピー、アンロード、またはロード操作に関してすべての入力データ・セットで検索されて、常に除外されます。メンバーは、EXCLUDE ステートメントに先行する INDD ステートメント上で指名された入力データ・セットから除外されます。

コピー操作では、メンバー名または別名として最大 8 文字を使用できます。

制限: EXCLUDE は COPYGRP では使用できません。

置換オプションは排他的コピーまたはロードに関してデータ・セット・レベルで指定できます。その場合、入力データ・セット内の非排他的メンバーが出力データ・セットの同じ名前のメンバーに置き換わります。置換オプションの詳細については、30 ページの『データ・セット内のメンバーを置換する』を参照してください。

別名を持つメンバーをコピーする (COPY ステートメント)

ここでは、COPY ステートメントを使用して別名を持つメンバーが入った PDS、PDSU、または PDSE をコピーすることについて解説します。プログラム・オブジェクトのコピーには、COPYGRP ステートメントをお勧めします。プログラム・オブジェクトのコピーについては、32 ページの『プログラム・オブジェクトをコピーする (COPYGRP ステートメント)』を参照してください。

ヒント: COPY ステートメントを使用してプログラム・オブジェクトをコピーすると、エラーが生じることがあります。

- データ・セット全体を新規のデータ・セット (コピー操作の前にメンバーがないもの) にコピーする場合、すべてのメンバーおよびその別名がコピーされて、それらの互いに対する関係は元のデータ・セットと同じになります。
- データ・セットを別のデータ・セットにマージする場合、入力メンバーが出力メンバーを置換するように指定しているのであれば、出力データ・セット上のメンバーまたは別名は変更されません。

例:

どんな状況でも、入力データ・セットに別名 B を持つメンバー A があり、出力データ・セットに別名 B を持つメンバー C がある場合、置換を指定しなければ、メンバー A はコピーにより上書きされますが、別名 B は C への参照を継続します。置換を指定した場合、新規にマージされたデータ・セットでは B がメンバー A の別名としてコピーされます。

- 区分データ・セットから選択的にコピーする場合、コピーしたいすべての名前およびその別名を指定しなければなりません。

例:

メンバー A を選択していて、メンバー A には別名 B および C がある場合、その 3 つの名前すべてをコピーするには SELECT MEMBER=(A,B,C) を指定しなければなりません。これにより、そのメンバーに対して 1 回のコピーが行われ、3 つの名前はすべてディレクトリーに入れられてそのメンバー・データに関連付けられます。

SELECT MEMBER=(A,C) を指定した場合。これにより、そのメンバーに対して 1 回のコピーが行われ、A および C の 2 つの名前がディレクトリーに入れられてそのメンバー・データに関連付けられます。

- PDSE へコピーする場合、または PDSE からコピーする場合、メンバー名を指定しなければなりません。出力データ・セットで別名がメンバー名になることはありません。
- データ・セットを (EXCLUDE ステートメントを使用して) 排他的にコピーする場合、コピー操作からメンバー・データを除外するにはそのメンバー名だけでなくすべての別名も指定しなければなりません。

例:

メンバー A をコピー操作から除外したい場合、A に別名 B および C があれば、EXCLUDE MEMBER=(A,B,C) を指定する必要があります。

MEMBER=A だけを指定した場合、そのメンバーは出力データ・セットに別名 B および C としてコピーされます。

- メンバーの置換または名前変更に関する規則は別名およびメンバーの両方に適用されます。それらに区別はありません。

データ・セット内のメンバーを置換する

IEBCOPY の COPY または COPYGRP ステートメントを使用して、出力区分データ・セットまたは PDSE 上のメンバー置換することができます。このトピックでの解説は、COPY ステートメントのためのものです。COPYGRP を使用するメンバーの置換については、32 ページの『プログラム・オブジェクトをコピーする (COPYGRP ステートメント)』を参照してください。

COPY ステートメントを使用して、以下のタスクを実行できます。

- データ・セット・レベルでの置換を指定できます。この場合、入力データ・セットのすべてのメンバーは出力データ・セットにコピーされます。出力データ・セット上の各メンバーで、入力データ・セット上のメンバー名と同じ名前を持つものは置換されます。
- メンバー・レベルでの置換を指定できます。この場合、入力データ・セットの特定のメンバーが出力データ・セットの同名のメンバーを置換すること、および他のメンバーのコピーはそれが出力データ・セット上にすでに存在しない場合にだけ行われることを指定できます。

メンバー・レベルで置換を指定する場合、入力メンバーの名前変更も行えます。その新しい名前が出力データ・セット・ディレクトリーで検索されて、メンバーをコピーするかどうか判別されます。たとえば、メンバー A を B に名前変更して、出力データ・セット上のメンバー B を置換するようにできます。

データ・セット・レベルでの置換を指定する: 区分データ・セットをマージするか、またはアンロード・データ・セットをすでにメンバーのある区分データ・セットにロードする場合、入力および出力データ・セットに同じ名前のメンバーが存在する可能性があります。通常の処理では、それらの入力メンバーは同名の出力メンバーを置換しません。すべての入力メンバーが出力データ・セットをコピーして、同じ名前の出力メンバーを置換するように指定するには、INDD または COPY ステートメントで置換 (R) オプションを使用します。

データ・セット・レベルで置換 (R) を指定した場合、入力データは以下のように処理されます。

- フル・コピーまたはロードの処理で、入力データ・セット内のすべてのメンバーは出力区分データ・セットにコピーされます。出力区分データ・セット内にすでに名前が存在するメンバーは、入力データ・セットからコピーまたはロードされるメンバーによって置換されます。
- 選択的なコピーまたはロードの処理で、選択したすべての入力メンバーは出力データ・セットにコピーされて、同名の出力データ・セット・メンバーが置換されます。選択的なコピーを実行するときにデータ・セット・レベルで置換 (R) を指定すると、コピーしたいメンバーごとに置換 (R) を指定しなくてもすみます。
- 排他的なコピーの処理で、入力データ・セット上の除外されていないすべてのメンバーは出力区分データ・セットにコピーまたはロードされて、それらのメンバーが出力区分データ・セット上の重複名に置換されます。

メンバー・レベルでの置換を指定する: SELECT ステートメントの MEMBER オペランドにコピーするメンバーの名前を指定する場合、出力データ・セット内に同名のメンバーがある入力メンバーに対して置換 (R) オプションを指定します。このようにして、入力データ・セットから多くのメンバーをコピーしても、そのごく一部だけが出力データ・セット内のメンバーを置換することができます。

メンバー置換処理の考慮事項: フル、選択的、排他的な処理には違いがあります。これらの相違は、データ・セットまたはメンバー・レベルのどちらかで置換 (R) オプションを指定した場合で、出力データ・セットにコピーまたはロードされる入力データ・セットの一部またはフルに共通のメンバー名が含まれる場合には、覚えておく必要があります。それらの相違は、次のとおりです。

- フル・コピーまたはロードを実行するとき、出力区分データ・セットには最後にコピーされた入力データ・セット上にあった置換メンバーが含まれています。
- 選択的なコピーまたはロードを実行するとき、出力区分データ・セットには最初に検索された入力データ・セット上にあった選択された置換メンバーが含まれています。選択されたメンバーが見つかる、それは再び検索されません。そのため、それが見つかった後、選択されたメンバーはコピーまたはロードされます。同じメンバーが別の入力データ・セット上に存在する場合、それは検索されないため、コピーまたはロードされません。
- 排他的なコピーまたはロードを実行するとき、出力区分データ・セットには最後にコピーまたはロードされた入力データ・セット上にあった、除外するメンバーを除くすべてのメンバーが含まれています。

選択したメンバーを名前変更する

SELECT ステートメントを使用してメンバーを名前変更する: SELECT ステートメントを使用してメンバーを名前変更するには、古い名前、新規の名前、およびオプションで置換 (R) 標識を共に括弧に入れて SELECT ステートメントの MEMBER パラメーターのオペランドとします。MEMBER パラメーター・オペランドには、「古い名前 / 新規の名前 / 置換」の組を必要な数だけ指定できます。

置換 (R) オプションは、新規の名前が出力データ・セット内のメンバー名と一致する場合に使用しなければなりません。使用しない場合、そのメンバーはコピーされません。INDD パラメーターの使用により置換がすべてのメンバーに対してグロー

バルに指定されても、または MEMBER パラメーターの使用により個別のメンバーに対して指定されても、それは同様です。

選択したメンバーは入力データ・セット・ディレクトリー内で名前変更されません。それらは単に新しい名前でも出力データ・セットに追加されます。

プログラム・オブジェクトをコピーする (COPYGRP ステートメント)

COPYGRP ステートメントを使用して、プログラム・オブジェクトおよびその別名を PDSE データ・セットに、またはそこからコピーすることをお勧めします。プログラム・オブジェクトの別名は、最大 8 文字です。COPYGRP ステートメントを使用すると、それらの長い別名がメンバーと共に確実にコピーされます。

グループのコピー、アンロード、またはロードを開始するには、COPYGRP ステートメントを使用します。1 つのグループは、1 つのメンバーおよびそのすべての別名から構成されます。COPYGRP はグループを単一のエンティティーとして扱います。

COPYGRP は、入力データ・セットまたは出力データ・セット、もしくはその両方が PDSE の場合にデータ・セットをコピーするために使用できます。

PDSE から PDSE へ
 PDSE から PDS へ
 PDS から PDSE へ

グループをアンロードする場合は、

PDSE から PS へ

グループをロードする場合は、

PS から PDSE へ

どちらのデータ・セットも PDSE ではない場合、要求は COPYGRP の構文要件に応じて COPY 操作として処理されます。

COPYGRP ステートメントを使用するときは、以下のようになります。

- グループ内のすべての別名はメンバーと共にコピーされるか、またはグループ内の別名もメンバーもコピーされません。
- コピー操作ごとに使用する INDD は、1 つだけです。
- SELECT ステートメントを使用して、選択的にメンバーをコピーできます。メンバー名または別名のどちらかを指定して、メンバーおよびそのすべての別名をコピーすることができます。
- SELECT ステートメントには置換 (R) を含めないでください。
- EXCLUDE ステートメントはサポートされていません。

プログラム・オブジェクトを置換する

置換 (R) オプションを INDD パラメーターに指定した場合は、以下のようになります。

- 出力データ・セット・メンバーおよびその別名は、入力データ・セットのメンバーおよび別名と同じメンバー名および別名の場合に置換されます。

例:

入力データ・セットにメンバー A とその別名 B があり、出力データ・セットにメンバー A とその別名 B があります。入力データ・セットのメンバーおよび別名は、出力データ・セットのメンバーおよび別名を置換します。

- 出力データ内のメンバーの別名が入力データ・セット内の別の名前前のメンバーの別名と同じである場合、コピーは失敗します。

例:

入力データ・セットにメンバー A とその別名 B があり、出力データ・セットにメンバー C とその別名 B があります。別名 B が出力データ・セット上の他の名前前のメンバーを指しているため、コピーは失敗します。

- 出力データ・セットのメンバーおよび別名が入力データ・セットのメンバーおよび別名と一致しない場合、入力データ・セットのメンバーおよび別名がすべてコピーされます。

例:

入力データ・セットにメンバー A とその別名 B があり、出力データ・セットにメンバー C とその別名 D があります。コピーの後、出力データ・セットには A およびその別名 B、および C とその別名 D が含まれます。

EXCLUDE ステートメントはサポートされていません。

区分データ・セットを圧縮する

区分データ・セットには、削除済みのメンバーまたは更新済みメンバーの古いバージョンがかつて存在していた未使用域 (ガスと呼ばれることもある) が含まれています。この未使用域がレクラメーション処理されるのは、区分データ・セットが新規のデータ・セットにコピーされる時、または同じ場所での圧縮操作が正常に完了した後だけです。それは PDSE にとっては意味がなく、要求されても無視されません。

同所圧縮を要求する最も簡単な方法は、SELECT / EXCLUDE ステートメントを無指定で COPY ステートメントの OUTDD および INDD パラメーターに同じ DD 名を指定することです。

しかし、実際に圧縮が実行されるのは、入力および出力がどちらも同じボリューム上の同じデータ・セットであるときです。たとえば、以下のジョブ・ステップはデータ・セット Pacanowska を圧縮します。

```
//COMPRESS EXEC PGM=IEBCOPY
//A DD DSNAME='Pacanowska',DISP=OLD
//B DD DSNAME='Pacanowska',DISP=OLD
//SYSIN DD *
      COPY OUTDD=B,INDD=A
```

複数の入力が INDD ステートメント上で行われる場合、それら入力 DD 名のどれかが OUTDD 名と一致する場合に同所圧縮が実行されます。圧縮操作は、INDD リスト内の DD 名と同じ相対順序で実行されます。

たとえば、以下の COPY ステートメントを考えてください。

```
COPY OUTDD=B,INDD=(A,B,C,B)
```

- DD 名 A のデータ・セットが DD 名 B にコピーされます

- データ・セット B が圧縮されます
- DD 名 C が DD 名 B にコピーされます
- データ・セット B が再び圧縮されます

同所圧縮を実際に行う前に、圧縮予定のデータ・セットのコピーを作成することをお勧めします。IEBCOPY の 1 回の実行でその両方を行うことができ、以下のジョブ・ステップによって IEBCOPY ジョブ・ステップが戻りコード 0 で終了した場合にバックアップ・コピーを削除できます。

重要: 電源障害、異常終了、TSO アテンション、入出力エラーなどにより IEBCOPY が処理の途中で中断した場合、区分データ・セットが破壊されることがあります。同所圧縮が正常に終了するまで、区分データ・セットのバックアップを保持してください。

重要: 現在複数のユーザーが使用している区分データ・セットは圧縮しないでください。それを圧縮した場合、データ・セットが損傷していても、他のユーザーには損傷したように見えてしまいます。他のユーザーが圧縮後に区分データ・セットを更新すると、その更新によって実際に区分データ・セットが損傷します。

圧縮処理の考慮事項

- PDSE に対して同所圧縮を実行しようとする、IEBCOPY はその要求を無視して次の制御ステートメントから処理を継続します。
- 同所圧縮は、データ・セットに割り当てられた範囲を解除しません。
- 圧縮操作中、選択したメンバーの組み込み、除外、または名前変更は行えません。
- データ・セットの圧縮中に、ブロック・サイズなどのデータ・セット DCB パラメーターを変更できません。

ロード・モジュールを変更する

ALTERMOD は、古いシステムからのロード・モジュールを 1 回の操作で更新するために設計されました。IEBCOPY を使用して、MVS/370 以前のリンケージ・エディターによって記述された区分データ・セット・ロード・モジュールを更新して、より高速にロードするようにできます。MVS/370、および MVS と DFSMS の後続のバージョンにあるリンケージ・エディターで処理されたロード・モジュールは、変更の必要がありません。PDSE 内のプログラム・オブジェクトも変更の必要がありません。

ALTERMOD は、正確な再配置辞書 (RLD) のカウントおよびセグメント・ブロックのカウントを、モジュール内の制御レコードに入れます。ALTERMOD は、ロード・モジュールの新規のコピーを作成することなくこの更新を実行します。それを使用して、誤った RLD カウントを持つモジュールを変更できます。誤った RLD カウントを持つモジュールの例としては、リンケージ・エディター以外のプログラムによって作成されたモジュール、または IEBCOPY 以外のプログラムによってコピーされたモジュールなどがあります。

ALTERMOD ステートメントは、以下の場合には機能しません。

- ロード・モジュールが分散ロード形式であるか、または編集不能 (NE) 属性によってリンク・エディットされている場合。

- データ・セットが PDSE である場合。(それは無視されます。)

SELECT ステートメントを使用して、ALTERMOD によって処理されるメンバーを識別するとき、それらのメンバーを名前変更することはできません。

ロード・モジュールのコピーおよび再ブロック

COPYMOD ステートメントを使用すると、ロード・モジュールをコピーおよび再ブロックして、データ・セットのコピー先の装置に適したブロック・サイズにすることができます。

テキスト・レコード、再配置辞書 (RLD)/ 制御レコード、およびオーバーレイ・ロード・モジュールの注釈リスト・レコードは、COPYMOD を使用するときには再作成されます。SYM および CESD レコードなどの他のレコードは、変更されことなくコピーされます。COPYMOD によって処理されるロード・モジュールは、再びリンク・エディットすることができます。

- ページ位置合わせフォーマットのロード・モジュールは、操作が COPYMOD ではなく COPY であるかのように、再ブロックなしでコピーされます。さらに、ALTERMOD の機能が、作成されたコピーに対して実行されます。
- 分散ロード / フォーマットのロード・モジュール、および編集不能 (NE) 属性によってリンク・エディットされたモジュールは、コピーされますが、再ブロックまたは変更されません。
- ロード・モジュールとして認識されないメンバーは、コピーされますが、再ブロックまたは変更されません。
- 下位互換性 (DC) のあるリンケージ・エディター属性を持つロード・モジュールは、MINBLK または MAXBLK パラメーターで指定された値に関係なく、最大ブロック・サイズの 1024 (1K) に再ブロックされます。

出力データ・セット・ラベルのブロック・サイズは、MAXBLK 値に一致するために必要なだけ、COPYMOD によって増加させられます。

COPYMOD は、出力データ・セットのブロック・サイズよりも長いレコードを書きません。しかし、COPYMOD がメンバーを処理できないため、代わりに COPY が使用される場合、COPY は出力データ・セットのレコード・サイズよりも長いレコードを含むすべてのレコードをコピーします。

COPYMOD の再ブロック機能により、以下のブロック・サイズを指定できます。

- 他のシステムまたはプログラムと互換性を取るための最大ブロック・サイズ
- トラックの最後に書き込む最小のブロックを指定するための最小ブロック・サイズ

IEBCOPY は、次に書き込まれるブロックのサイズを割り当てる前に、トラックに残っているスペースの量を判別します。この量が出力ブロック・サイズよりも小さい場合、IEBCOPY はトラックの残りのスペースを使用するため、より小さいブロックを書き込めるかどうかを判別しようとします。COPYMOD 機能によって生成される最大ブロック・サイズは 32760 バイトです。

変更された COPYMOD 操作

MVS/DFP バージョン 3、リリース 2 より前は、COPYMOD のデフォルトの MAXBLK サイズは 32760 でした。その目的は、可能な限りテキスト・ブロックを長くして、モジュールを取り出すために読み取るテキスト・ブロックの数を減らすことでした。その結果、しばしばデータ・セット・ラベル内の BLKSIZE よりも長い物理レコードがデータ・セット内に作成されました。(これらのブロックは、肥大ブロックと呼ばれます。) この状態は、入出力エラーまたはデータ保全性違反を生じさせます。

バージョン 3、リリース 2 以降は、デフォルトの MAXBLK 値は 32760 ではなくデータ・セットのブロック・サイズとなります。さらに、MAXBLK 値がデータ・セット・ラベルのブロック・サイズよりも大きな値に指定された場合、データ・セット・ラベル内の値は増加して、指定の MAXBLK 値となります。

IEBCOPY がテーブルおよびバッファのために仮想記憶域を使用する方法

IEBCOPY ESCON[®] パフォーマンス・アップデート以降では、作業域サイズの最小値は約 208K です。ジョブの REGION サイズが小さい場合、それを増加させなければならぬことがあります。

IEBCOPY ジョブのための推奨最小 REGION サイズは、区分データ・セットだけがコピーされる場合は 1M、PDSE がコピーされる場合は 2M です。

JCL EXEC PARM フィールドの WORK=nnn パラメーターは、IEBCOPY に必要な作業域のスペースを制御します。作業パラメーターの値の増加が効果を得るためには、REGION サイズは領域サイズとプログラム・サイズを加えた値よりも大きくなければなりません。

この作業域から、区分データ・セット・ディレクトリー用のテーブル、バッファ、およびストレージを取り出せます。作業域が十分にない場合、区分データ・セット・ディレクトリーは SYSUT3 および SYSUT4 にスピルします。さらにストレージが必要な場合、IEBCOPY はメッセージを出して停止します。WORK および REGION 値が大きいほど、SYSUT3 および SYSUT4 に割り当てられたスピル・データ・セットをオープンすることなく、より大きなディレクトリーの処理が可能になります。

WORK および REGION サイズが大きい場合、小さなデータ・セットをコピーするときには SIZE パラメーターを指定してください。これにより、バッファが過大になってパフォーマンスの低下や実ストレージの不足が生じることを防ぐことができます。

IEBCOPY がテーブルおよびバッファを割り振る方法

1. IEBCOPY は WORK パラメーターの値を上限として可能な限り大きな作業域を取得します。
2. 初期の最小サイズのバッファおよびチャネル・プログラム構造領域は、作業域から割り当てられます。

この領域のサイズは、最大の入力または出力装置トラックのサイズの約 4 倍に約 3% を加えた値となります。

IEBCOPY が最小サイズの作業域を割り当てられない場合、メッセージを出して停止します。

- IEBCOPY は制御ステートメントの処理、区分データ・セット・ディレクトリーの読み取り、およびコピーするメンバーのテーブルの作成を行います。作業域がテーブルおよびディレクトリーを保持できない場合、ディレクトリー項目は SYSUT3 および SYSUT4 にスピルして、テーブルが拡張します。残っているスペースがなく、すべてのディレクトリーがスピルした場合、IEBCOPY はメッセージを出して停止します。
- 作業域に未使用のストレージが十分に残っている場合、それは 2 次バッファースとして使用されます。

SIZE= パラメーターが指定された場合、両方のバッファースの合計サイズが指定値を超えないように、2 次バッファースのサイズは制限されます。SIZE= の値が最小サイズの 2 次バッファースを許可しない場合、2 次バッファースは作成されません。

ヒント: SIZE=999999 (または WORK= 値よりも数千少ない任意の数値) をコーディングした場合、IEBCOPY はバッファースを作業域に割り当てることができても、テーブルのためのスペースが残っていないことがあります。その場合、REGION= および WORK= 値を増加させるか、または SIZE= パラメーターを除去してください。

大きなディレクトリーのあるデータ・セットをコピーする場合、ディレクトリー情報を保存するために最大量の仮想記憶を使用可能にします。WORK=8M (または他の大きな値)、およびそれに対応する大きな REGION を指定します。

小さなディレクトリーのデータ・セットが大きな領域サイズを伴ってコピーされる時、2 次入出力バッファースが非常に大きく (数メガバイト) なって、実ストレージの不足が生じることがあります。その結果、バッファースの大部分が入出力中に固定されている実フレームによって占められるため、システム・ページングおよびシステム不調が増加することがあります。この場合、SIZE=1M またはより小さい値を指定してバッファースに使用するストレージの量を制限し、ディレクトリー情報に多くのストレージを使用できるようにしてください。

制御ステートメントを指定しなければならない状態を回避する

SYSIN DD ステートメントが DD DUMMY であって、空のファイルを指している場合、または省略されている場合、IEBCOPY は COPY ステートメントを生成します。これを使用すれば、SYSIN 用の制御ステートメント・データ・セットを指定しなくても IEBCOPY を実行できます。

生成されるステートメントは、JCL EXEC PARM フィールドに指定されたパラメーターに応じて、数種類のフォームをとります。

- COMPRESS および REPLACE を指定しない場合、生成されるステートメントは COPY OUTDD=SYSUT2,INDD=SYSUT1 となります。これは、SYSUT1 DD ステートメントで指定したデータ・セットから SYSUT2 DD ステートメントで指定したデータ・セットへの置換なしのコピーを行います。
- COMPRESS を指定して REPLACE を指定しない場合、生成されるステートメントは COPY OUTDD=SYSUT2,INDD=(SYSUT1,R) となります。これは、SYSUT1

DD ステートメントで指定したデータ・セットから SYSUT2 DD ステートメントで指定したデータ・セットに置換を伴うコピーを行います。

3. COMPRESS を指定した場合、生成されるステートメントは COPY OUTDD=SYSUT2,INDD=SYSUT2 となります。これは、SYSUT2 DD ステートメントで指定したデータ・セットを同じ場所で圧縮します。

パラメーター COMPRESS、COPY、COPYMOD、または REPLACE のいずれかを JCL EXEC PARM フィールドに指定すると、SYSIN はオープンされずに、上記のような制御ステートメントが自動的に生成されます。

入出力

IEBCOPY は以下の入力を使用します。

- コピー、マージ、変更、再ブロック、ロード、またはアンロードするメンバーを含む、区分データ・セット、PDSE、またはアンロード・データ・セット。
- ユーティリティー制御ステートメントを含むオプションの制御データ・セット。制御データ・セットは以下のいずれかの場合に必要です。
 - 処理する入力または出力データ・セットが複数ある場合。
 - 指定したメンバーを選択または除外する場合。
 - ロード・モジュール・ライブラリーを変更または再ブロックする場合 (ALTERMOD または COPYMOD が必要)。

IEBCOPY は以下の出力を生成します。

- コピー、マージ、変更、再ブロック、またはアンロードされたメンバーを含む出力データ・セット。出力データ・セットは新規のデータ・セット (コピー、再ブロック、ロード、またはアンロードによる)、または古いデータ・セット (マージ、同所圧縮、コピー、変更、またはロードによる) です。
- 制御ステートメント、IEBCOPY アクティビティー、およびエラー・メッセージのうち、該当するものをリスト表示するメッセージ・データ・セット。

IEBCOPY では、以下のものが必要となることがあります。

- 入力または出力区分データ・セット・ディレクトリーのための仮想記憶域が不十分の場合にスペースを提供するために使用される一時データ・セットである、オプションのスピル・データ・セット。これらのデータ・セットは、必要とされるときにだけオープンされます。

IEBCOPY をアプリケーション・プログラムから起動した場合、IEBCOPY を呼び出す前に SVC 99 を発行することによりデータ・セットを動的に割り振ることができます。

戻りコード

IEBCOPY の戻りコードについては、付録 A を参照してください。

制約事項

IEBCOPY の使用については、以下の特性、規則、および制約事項を参照してください。

- IEBCOPY は、VIO (仮想入出力) データ・セットをサポートします。
- IEBCOPY は、EXCP アクセス方式および特別な入出力付加機能を使用します。したがって、
 - IEBCOPY は APF 許可プログラムであり、そのため許可ライブラリーから実行しなければなりません。つまり、他のプログラムがそれを呼び出すとき、そのプログラムも APF 許可でなければなりません。システム保全性を保護するため、呼び出し側プログラムは、*z/OS MVS プログラミング：アsembler・サービス ガイド*、GC88-6583 に示されているシステム保全性要件に従ってなければなりません。
 - BUFNO など一部の共通 DCB パラメーターは無視されます。
 - IEBCOPY が (チャンネル) プログラム制御割り込み (PCI) を十分速く受信してサービスを提供できない場合、DASD および CPU の一部の組み合わせのパフォーマンスが影響を受けることがあります。
- IEBCOPY を監視プログラム状態または記憶保護キー・ゼロによってロードすることはできません。これは、監視プログラム状態または記憶保護キー・ゼロで実行するシステム・カーネルに想定される特別なシステム・インターフェースを使用しないアプリケーション・プログラムです。
- 可変長スパン形式レコード (VS または VBS) は、区分データ・セットではサポートされません。
- 共有または使用中のデータ・セットは、対象データ・セットが共有不可 となるまでは同じ場所での圧縮または更新を行わないでください。7 ページの『データ・セットの共用』を参照してください。
- PDSE が関与していて、PDSE 処理ルーチン上で使用可能な仮想記憶域の量が小さいとき、記憶域についてのメッセージはコンソールにだけ表示されて SYSPRINT データ・セットには入れられないことがあります。
- 下位互換性 (DC) のあるリンケージ・エディター属性を持つロード・モジュールが COPYMOD 処理中に検出された場合、MINBLK および MAXBLK パラメーターで指定された数値に関係なく、最大ブロック・サイズの 1024 (1K) に再ブロックされます。
- 入力または出力データ・セットのどちらかが以下の場合に、再ブロックは実行できません。
 - 不定形式レコード
 - キー付きレコード
 - トラック・オーバーフロー・レコード
 - 注釈リストまたはユーザー TTRN

同じ場所での圧縮が指定されている場合にも、再ブロックは実行できません。(不定形式レコードおよび注釈リストを持ったロード・モジュールは、COPYMOD ステートメントの際に再ブロックされることがあります。)
- 同所圧縮の機能は以下のものに対しては実行できません。
 - アンロード・データ・セット
 - トラック・オーバーフロー・レコードのあるデータ・セット

- キー付きレコードのあるデータ・セット
- 移動不能データ・セット
- PDSE (要求は無視されます)
- PDSE には、注釈リスト、キー、またはトラック・オーバーフローのあるメンバーを含めることはできません。同じ PDSE にロード・モジュールおよび非ロード・モジュールを混在させることはできません。
- DASD に対して OPTCD=W を使用すると、COPY 操作の速度がかなり低下することがあります。OPTCD はデータ・セット・ラベル内にあるので、OPTCD が DD ステートメント上にコーディングされていないときにアクティブになります。
- OPTCD=W が役立つのは、JCL でコーディングされたときだけです。データ・セット・ラベル内に OPTCD=W があると、そのラベルから削除されます。

注: IEBCOPY が出力区分データ・セットのブロック・サイズよりも物理的に長いレコードをコピーする場合、メッセージ IEB175I (戻りコード 4) が出力されて、データ・セットに FAT ブロック、つまりデータ・セット内に作成されたデータ・セット・ラベル内の BLKSIZE よりも長い物理レコードが含まれていることが通知されます。

- COPYMOD サイズ制限
 - 入力として使用するロード・モジュールでは、1 つのテキスト・ブロック内に 60 を超える CSECT があってはなりません。
 - オーバーレイ・ロード・モジュールには、255 を超えるセグメントがあってはなりません。
- IEBCOPY ユーザー TTR 制限
 - ディレクトリー内のユーザー TTRN フィールドは、最大 3 つまでです。
 - そのうちのいずれか 1 つだけが n>0 の値を持てます。
 - n>0 になっているユーザー TTRN で識別される注釈リスト・レコードの最大長は、ブロックおよびレコード記述子ワードを含めて 1291 バイトです。
 - 注釈リスト・レコード内の TTRN フィールドは、n>0 の値を持てません。
- アンロード・データ・セットからのロード・モジュールは、プログラム・オブジェクトとして PDSE に再ロードすることはできません。ロード・モジュールを区分データ・セットに再ロードしてから、その区分データ・セットを PDSE にコピーして、アンロードされたロード・モジュールをプログラム・オブジェクトに変換してください。
- IEBCOPY サイズ制限
 - 1 回のコピー操作で許容されるすべての SELECT または EXCLUDE カード上での名前変更数は、 $(2 * \text{max_trk}) / 16$ です。ここで、『max_trk』はコピー操作で使用されるより大きな装置のシリンダー長です。3380 では、この制限は約 5925 となります。3390 では約 7050、9345 では約 5800 が限界値です。
 - PDSU ブロック・サイズ + 20 よりも小さな PDSU ブロック・サイズを使用しないでください。
 - SYSUT4 スペースは、単一の連続したスペースでなければなりません。

制御

IEBCOPY は、JCL およびユーティリティー制御ステートメントによって制御されます。

ジョブ制御ステートメント

表 9 は、IEBCOPY で使用するジョブ制御ステートメントをまとめたものです。

表 9. IEBCOPY で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	IEBCOPY を開始します。
SYSPRINT DD	制御ステートメントおよびメッセージのリストに使用する順次データ・セットを定義します。
SYSUT1 または anyname1 DD	入力用の区分データ・セットまたはアンロード・データ・セットを定義します。区分データ・セットは DASD 上に存在するか、または VIO データ・セットでなければなりません。アンロード・データ・セットは、アンロード操作の結果として作成された順次データ・セットで、DASD、磁気テープ、または QSAM アクセス方式によってサポートされる他の任意の装置上に存在することができます。このアンロード・データ・セットは、基本フォーマット、ラージ・フォーマット、拡張フォーマット、圧縮フォーマット、またはテープであることができます。
SYSUT2 または anyname2 DD	出力用の区分データ・セットまたはアンロード・データ・セットを定義します。区分データ・セットは DASD 上に存在するか、または VIO データ・セットでなければなりません。アンロード・データ・セットは、アンロード操作の結果として作成された順次データ・セットで、DASD、磁気テープ、または QSAM アクセス方式によってサポートされる他の任意の装置上に存在することができます。
SYSUT3 DD	DASD または VIO 装置上にスピル・データ・セットを定義します。SYSUT3 は、現行の入力データ・セット・ディレクトリー項目の一部またはすべてを格納するスペースが仮想記憶域にない場合に使用します。
SYSUT4 DD	DASD または VIO 装置上にスピル・データ・セットを定義します。SYSUT4 は、出力データ・セット・ディレクトリーのためのスペースが仮想記憶域にない場合に使用します。
SYSIN DD	オプションの制御データ・セットを定義します。

EXEC ステートメント

EXEC ステートメントの構文は次のとおりです。

//[stepname]	EXEC	[,PGM=IEBCOPY. [,REGION={nlnKlnM}] [,PARM=<parms>]
--------------	------	----------------------------------------------------------

各項目の意味は以下のとおりです。

PGM=IEBCOPY

IEBCOPY プログラムを実行することを指定します。

REGION={nlmKlnM}

オペレーティング・システムによって IEBCOPY が使用可能になるストレージの大きさを指定します。区分データ・セットだけを使用する場合、1M を指定します。PDSE を使用する場合、2M を指定します。

注:

1. 数値 n は任意の桁の数字で、10 進数で指定します。K は数値を 1024 バイト (1 キロバイト) で乗算し、M は数値を 1024K つまり 1048576 で乗算します。
2. より大きな REGION サイズを指定しても、WORK= 値を同時に増加させない限り効果は期待できません。
3. REGION サイズを JOB ステートメントで指定すると、EXEC ステートメント上で指定された REGION を指定変更します。

PARM=

以下に示す任意のパラメーターを任意の順序で IEBCOPY に指定できます。複数のパラメーターは、互いの間にコンマを入れて区分します。

CMWA=nK

より大きなロード・モジュールを処理する場合、このパラメーターを指定すると COPYMOD 作業域サイズが 120K から増加します。その場合、メッセージ IEB1133E が出されるので、知ることができます。

COMPRESS

IEBCOPY を使用して制御ステートメントを生成する場合、このパラメーターを指定して制御ステートメントが COPY 操作ではなく同所圧縮操作を実行するようにします。37 ページの『制御ステートメントを指定しなければならない状態を回避する』を参照してください。

COPY

IEBCOPY を使用して制御ステートメントを生成したい場合は、このパラメーターを指定して制御ステートメントが COPY 操作を実行するようにします。37 ページの『制御ステートメントを指定しなければならない状態を回避する』を参照してください。

COPYGRP

IEBCOPY を使用して制御ステートメントを生成したい場合は、このパラメーターを指定して制御ステートメントが COPYGRP 操作を実行するようにします。37 ページの『制御ステートメントを指定しなければならない状態を回避する』を参照してください。

COPYMOD

IEBCOPY を使用して制御ステートメントを生成したい場合は、このパラメーターを指定して制御ステートメントが COPYMOD 操作を実行するようにします。37 ページの『制御ステートメントを指定しなければならない状態を回避する』を参照してください。

LC=n**LPP=n****LINECOUNT=n**

n は SYSPRINT 出力リストの各ページを印刷するときの見出しを含めた行数です。デフォルトは 60 です。

LIST=NO**LIST=YES**

LIST= オペランドが COPY、COPYMOD、または ALTERMOD ステートメントで省略された場合のデフォルト値を設定します。デフォルトは **LIST=YES** です。

RC4NOREP

このパラメーターを使用すると、REPLACE が指定されていないためにソース・データ・セットからターゲット・データ・セットにモジュールがコピーされないときに、IEBCOPY が X'04' の戻りコードを設定します。
'RC4NOREP' を指定すると、REPLACE オプションが指定されていないためにコピーされないモジュールごとにメッセージ IEB1067W が出されます。

注: メッセージ IEB1067W は要求されている LIST オプションに関係なく出されます。

REPLACE

IEBCOPY を使用して制御ステートメントを生成したい場合は、このパラメーターを指定して制御ステートメントが置換を伴うコピー操作を実行するようにします。37 ページの『制御ステートメントを指定しなければならない状態を回避する』を参照してください。

SIZE={nlmKlnM}

IEBCOPY がバッファとして使用できる仮想記憶域の最大数を指定します。

このパラメーターを使用しないで、IEBCOPY にバッファ・サイズを選択させるのが最善です。

バッファ・サイズの最小値は、使用する装置の最大トラック・サイズの約 4 倍に約 3% を加えた値となります。この値に最大値はありませんが、IEBCOPY は作業域で使用可能な量よりも多くを使用することはできません。

数値 *n* は任意の桁の数字で、10 進数で指定します。K は数値を 1024 バイト (1 キロバイト) で乗算し、M は数値を 1024K つまり 1048576 で乗算します。

36 ページの『IEBCOPY がテーブルおよびバッファを割り振る方法』を参照してください。

SPCLCMOD

メンバーが COPYMOD 処理に不適格であり (たとえば、モジュールがページ位置合わせされている)、モジュールが通常の COPY としてスケジュールされている場合に、レコードがターゲット・データ・セットのブロック・サイズよりも大きいことが検出されると、操作全体が終了するように指定します (FAT ブロックは作成されません)。

COPYMOD が不適格であるために COPY 操作が成功すると、操作の戻りコードは、4 ではなくて、0 にセットされます (LIST=YES が有効である場合には、通常、戻りコードは 4 にセットされて、影響を受けるモジュールに対して、COPYMOD ではなく、COPY が実行されたことを示します)。

WORK={nlnKlnMl1M}

作業域がディレクトリー項目、内部テーブル、および入出力バッファを保持するために要求する仮想記憶域のバイト数を指定します。デフォルトの要求は 1M です。実際に取得する量は、REGION で使用可能なスペースを超えません。

数値 *n* は任意の桁の数字で、10 進数で指定します。K は数値を 1024 バイト (1 キロバイト) で乗算し、M は数値を 1024K つまり 1048576 で乗算します。

36 ページの『IEBCOPY がテーブルおよびバッファのために仮想記憶域を使用する方法』を参照してください。

SYSPRINT DD ステートメント

IEBCOPY は、制御ステートメントおよびそのアクションのログを SYSPRINT DD ステートメントに書き込みます。SYSPRINT が使用できない場合、IEBCOPY は処理を続けて要約メッセージをコンソールに出します。

SYSPRINT を SYSOUT または任意の QSAM データ・セットに割り当てることができます。指定できる有効な DCB パラメーターは以下のとおりです。

DSORG=PS

出力は常に順次編成となります。

RECFM=[F|FB|V|VB|FA|FBA|VA|VBA]

レコード・フォーマットは、固定長、固定長ブロック、および可変長または可変長ブロックとなります。これらのレコード・フォーマットのどれも、ISO/ANSI 制御文字 (VB の代わりに VBA など) を含めるように指定できます。

LRECL=

指定可能な論理レコード長の最小値は、固定長では 60、可変長では 64 です。最大値は 250 です。

BLKSIZE=

固定長ブロック・レコードを使用している場合、ブロック・サイズは論理レコード長の任意の倍数とすることができます。可変長または可変長ブロック・レコードを使用している場合、ブロック・サイズの最小値は論理レコード長に 4 を加えた値でなければなりません。ブロック・サイズは 32760 バイトを超えてはなりません。

DCB パラメーターに何も指定しない場合、データ・セット・ラベルに DCB パラメーターがなければ、IEBCOPY は RECFM=FBA、LRECL=121、およびブロック・サイズを選択します。出力装置が DASD または標準ラベル・テープである場合、システムは使用するブロック・サイズを判別します。出力装置が DASD または標準ラベル・テープではない場合、IEBCOPY は特定の装置のための最大値を RECFM の必要に応じて調整して使用します。

IEBCOPY が選択するものと異なるパラメーターを指定した場合、IEBCOPY はそれを受け入れます。RECFM を省略すると可変長フォーマット・レコードが選択されます。RECFM を指定すると、その値および指定した LRECL または BLKSIZE と調和する LRECL および BLKSIZE が選択されます。

ヒント:LRECL に TSO 端末の幅よりも 1 バイト少ない値を指定すると、右または左にスクロールすることなくレコード全体を表示できます。

SYSUT1 (anyname1) および SYSUT2 (anyname2) DD ステートメント

DD ステートメントは入力および出力データ・セットに必要です。ジョブ・ステップで使用される固有のデータ・セットごとに、1 つの DD ステートメントが必要です。アンロードする入力データ・セットごとに、固有の出力 DD ステートメント (アンロード・データ・セット用) を指定しなければなりません。複数のデータ・セットを同じアンロード・データ・セットにアンロードすることはできません。

1 つのコピー操作で入力データ・セットとして使用されるデータ・セットは、他のコピー操作で出力データ・セットとして使用できます。

複数の入力データ・セットを同じ DD ステートメント上で連結することはできません。

固定長または可変長レコードは再ブロックできますが、固定長フォーマット・レコードを可変長フォーマット・レコードに、または可変長フォーマット・レコードを固定長フォーマット・レコードに変換することはできません。入力データ・セットのブロック・サイズが出力データ・セットのブロック・サイズと等しくないとき、再ブロックまたはブロック解除は自動的に実行されます。

可変長形式データ・セットの場合、出力 LRECL は入力 LRECL より大きいか、または同じでなければなりません。固定長形式データ・セットの場合、入力および出力 LRECL は一致しなければなりません。不定形式データ・セットの場合、出力 BLKSIZE は入力 BLKSIZE より大きいか、または同じでなければならず、LRECL は無視されます。

PDSE は、区分データ・セットが同じデータを保持するために必要なスペースよりもいくらか多くのスペースを必要とします。区分データ・セットから PDSE にコピーするとき、IEBCOPY はジョブを停止するのではなく、出力 PDSE に対する 2 次スペース割り振りのゼロを指定変更します。1 次割り振り (トラック、シリンダー、またはブロック) が 1 単位増分するごとに、コピーに必要な追加のスペースが取得されます。データ・セット・ラベル内の 2 次スペース量は変更されません。

IEBCOPY が DCB パラメーターを出力データ・セット用に指定しなければならない場合、入力データ・セットからの対応する値が使用されます。特に、システム判別のブロック・サイズは出力データ・セットのブロック・サイズとしては使用されず、システム判別のブロック・サイズまたは再ブロック可能フラグが出力データ・セット・ラベルに設定されることはありません。

IEBCOPY がアンロード・データ・セット用の DCB パラメーターを選択するようにしてください。ブロック・サイズを指定する必要がある場合、入力データ・セットのブロック・サイズよりも 20 バイト以上大きな値を指定します。(これにより、アンロード・データ・セットにスパン・レコードが作成されることを防ぎます。) 詳細については、27 ページの『IEBCOPY アンロード・データ・セットの DCB パラメーター』を参照してください。

OPTCD=W は、書き込まれたデータの書き込み検査を行って、DASD キャッシュ制御装置 (3880-21、3990-3 など) が『ライトスルー』操作を行うよう要求するために使用できます。

ESCON サポートから始まり、OPTCD=W はチャネル・プログラムの異なるセットを使用して ECKD™ 対応の DASD への書き込み、およびすべての DASD からの読み取りを行います。これらのプログラムの実行速度はデフォルトよりもかなり遅いことがあります。そのため、OPTCD=W を使用すると大きなパフォーマンス上のコストが生じます。

広範囲のエラー・リカバリーがハードウェア、および MVS 入出力監視プログラムによって使用されるエラー・リカバリー手順 (ERP) によって生じるため、OPTCD=W によって要求される書き込み検査は通常は必要ありません。OPTCD=W は装置から読み取られるデータと仮想記憶域内のデータとを比較しないので、仮想記憶域から装置制御装置への転送中に誤りの混在したデータを検出することができません。OPTCD=W は、制御装置が装置からデータ (書き込まれた後) を読み取って、データと共に保管されたエラー検査コードを検査するようにします。

SYSUT3 および SYSUT4 DD ステートメント

IEBCOPY を使用するほとんどの場合に、スピル・データ・セット (SYSUT3 および SYSUT4) にスペースを準備する必要はありません。IEBCOPY が 2M 以上の領域サイズで実行されている場合、出力データ・セットのディレクトリー・ブロック数が 1600 よりも少ないときはどちらのスピル・データ・セットも必要ありません。

DASD 上にスペースを保持するため、これらのデータ・セットに VIO を使用できます。しかし、領域サイズ (および WORK= パラメーター値) を増加させて、SYSUT3 および SYSUT4 データ・セットを全く使用しないことの方が効果的です。これらのデータ・セットにマルチボリューム・データ・セットを使用することはできません。

SYSUT3 に必要なスペースは、コピーまたはロードされるメンバーの数によって異なります。SYSUT4 に割り振る必要のあるスペースは、IEBCOPY ジョブ・ステップ内で最大の出力区分データ・セット・ディレクトリーに割り振られるブロック数と同じか、またはそれ以上でなければなりません。ブロック・サイズを 80 としてスペースの必要量を計算してください。

必要なスペースは、出力データ・セットに書き込むディレクトリー・ブロック数によって異なります。SYSUT4 が使用されるのは、指定の INDD リスト上に複数のデータ・セットが指定されたときだけです。データ・セットには、データ・ブロックごとに 1 つのディレクトリー・ブロックが含まれます。ブロック・サイズを 256、キー長を 8 としてスペースの必要量を計算してください。

要件:SYSUT4 に出力区分データ・セット・ディレクトリーのコピーが含まれるため、SYSUT4 は単一の連続した範囲内に存在しなければなりません。データ・マネージメントの設計により、区分データ・セット・ディレクトリーが単一の範囲内に存在することが必要です。

IEBCOPY は、SYSUT3 または SYSUT4 に対して指定されたすべての DCB 情報を無視します。

SYSIN DD ステートメント

SYSIN DD ステートメントはオプションです。それを省略した場合、またはそれが DUMMY データ・セットか空のデータ・セットである場合、IEBCOPY は PARM フィールドのオプションを使用して制御ステートメントを生成します。

COPY、COPYMOD、REPLACE、または COMPRESS が PARM フィールドにある場合、SYSIN はオープンされません。

単一のデータ・セットをコピー、ロード、またはアンロードする場合、またはデータ・セットを同じ場所で圧縮する場合、SYSUT1 を入力データ・セット、SYSUT2 を出力データ・セットに使用することにより、ユーティリティー制御ステートメントを使用しないで済みます。詳細については、37 ページの『制御ステートメントを指定しなければならない状態を回避する』を参照してください。

固定長、可変長、または不定形式レコード・フォーマットを使用できます。さらに、レコード・フォーマットおよび互いに調和する BLKSIZE と LRECL も使用できます。レコード・フォーマットが紙送り制御を指示する場合、各レコード内の紙送り制御文字は無視されます。

シーケンス番号はオプションです。IEBCOPY はシーケンス番号 (8 桁) を可変長フォーマット・レコードの前および固定長フォーマット・レコードの後に探します。IEBCOPY がそれらを最初の制御ステートメントで見つけた場合、それらの列をすべての制御ステートメントで無視します。

ユーティリティー制御ステートメント

IEBCOPY は、表 10 のユーティリティー制御ステートメントによって制御されます。

表 10. IEBCOPY で使用するユーティリティー制御ステートメント：マイナー・ステートメント (SELECT または EXCLUDE) は各メジャー・ステートメントの後に指定して、メジャー・ステートメントの範囲を制限することができます。

ステートメント	用途
メジャー・ステートメント	
ALTERMOD	ロード・モジュールの同所圧縮操作の開始を指示します。
COPY	COPY 操作の開始を指示します。
COPYGRP	COPYGRP 操作の開始を指示します。
COPYMOD	コピーおよびロード・モジュール再ブロック操作の開始を指示します。
INDD=	他のコピー・ステップの開始を指示します。
マイナー・ステートメント	
EXCLUDE	入力データ・セット内のメンバーをコピー・ステップから除外するように指定します。
SELECT	入力データ・セット内のコピーするメンバーを指定します。

省略語

COPY は、MINBLK を除くすべてのキーワードの最初の文字をその省略形として受け入れます。COPYMOD の省略形は CM です。COPYGRP の省略形は CG です。

継続

コピー制御ステートメントを継続させるには、コンマで止めます。72 桁目に任意の非空白文字を入れて、次のレコードの 16 桁目から始めます。

注釈

注釈ステートメントを作成するには、ラベル・フィールドのある左列にアスタリスク (*) を入れます。そのレコードは印刷されて、その後無視されます。さらに、オペランドのある任意の制御ステートメントにも注釈を入れることができます。オペランドの後に 1 つ以上のスペースを置いて、注釈を開始します。

実行する IEBCOPY 操作を決定する

- *COPY* 操作を要求するには、入力および出力として区分データ・セットを指定します。
- *UNLOAD* 操作を要求するには、区分入力データ・セットおよび順次出力データ・セットを指定します。
- *LOAD* 操作を要求するには、順次入力データ・セットおよび区分出力データ・セットを指定します。

操作の範囲

IEBCOPY は、コピー操作 / コピー・ステップの概念を使用します。コピー操作は、COPY、COPYGRP、COPYMOD、または ALTERMOD ステートメントから開始して、他の COPY、COPYGRP、COPYMOD、または ALTERMOD ステートメントが見つかるまで、または制御データ・セットの終端が見つかるまで継続します。各コピー操作の中に、1 つまたは複数のコピー・ステップが存在します。SELECT または EXCLUDE ステートメントはコピー操作内で次のコピー・ステップの開始を示し、前のコピー・ステップの終了を示します。コピー操作内にそのような INDD ステートメントが見つからない場合、コピー操作は 1 つだけのコピー・ステップから構成されています。

表 11 は、コピー操作 / コピー・ステップの概念を示したものです。2 つのコピー操作が図に示されています。最初の操作は COPOPER1 というステートメントで始まり、2 番目の操作は COPOPER2 というステートメントで始まります。

表 11. 1 つのジョブ・ステップ内の複数のコピー操作

最初のコピー操作			
ステップ 1	COPOPER1	COPY	OUTDD=AA INDD=ZZ INDD=(BB,CC) INDD=DD INDD=EE
		SELECT	MEMBER=(MEMA,MEMB)
		SELECT	MEMBER=(MEMC)
ステップ 2			INDD=GG INDD=HH
		EXCLUDE	MEMBER=(MEMD,MEMH)
2 番目のコピー操作			

表 11. 1 つのジョブ・ステップ内の複数のコピー操作 (続き)

ステップ 1	COOPER2	COPY	OUTDD=YY INDD=(MM,PP) LIST=NO MEMBER=MEMB
		SELECT	
ステップ 2			INDD=KK INDD=(LL,NN)

表 11 に示す最初のコピー操作は、メンバーの 2 つのグループのコピーです。最初のもは COPY ステートメントから開始して、2 つの SELECT ステートメントまで続きます。2 番目のものは 2 つの SELECT ステートメントに続く最初の INDD ステートメントから開始して、2 番目の COPY ステートメントに先立つ EXCLUDE ステートメントまで続きます。

2 番目のコピー操作には 2 つのステップがあります。最初のステップは COPY ステートメントから開始して、SELECT ステートメントまで続きます。2 番目のステップは SELECT ステートメントの直後にある INDD ステートメントから開始します。

ALTERMOD ステートメント

ALTERMOD ステートメントは、ロード・モジュールを同じ場所を変更するために必要です。ALTERMOD は PDSE ではなく区分データ・セットでのみ機能します。

ALTERMOD ステートメントの構文は次のとおりです。

[label]	ALTERMOD	OUTDD=DDname [,LIST={YES NO}]
---------	----------	----------------------------------

各項目の意味は以下のとおりです。

OUTDD=DDname

変更する区分データ・セットの DD 名を指定します。

LIST={YES|NO}

変更するメンバー名を SYSPRINT データ・セットにリスト表示するかどうかを指定します。このパラメーターを省略すると、EXEC PARM フィールドのデフォルトが適用されます。

COPY ステートメント

1 つまたは複数のコピー、アンロード、またはロード操作を開始するには、COPY ステートメントを使用します。1 つの COPY ステートメントの後には任意の数の操作を続けることができます。1 つのジョブ・ステップ内には任意の数の COPY ステートメントを入れることができます。

COPY ステートメントの構文は次のとおりです。

[label]	COPY	OUTDD=DDname ,INDD=[(DDname(DDname,R))[,...]] [LIST={YESINO}]
---------	------	-------------------------------------------------------------------

各項目の意味は以下のとおりです。

OUTDD=DDname

出力データ・セットを位置指定する DD ステートメントの名前を指定します。

INDD=[(DDname(DDname,R))[,...]]

入力データ・セットを位置指定する DD ステートメントの名前を指定します。

INDD= が単独でレコード内にある (つまり COPY キーワードを伴わない) 場合、それは制御ステートメントとして機能して、現行の COPY 操作内に新規のステップを開始します。

コーディングできる値は以下のとおりです。

DDname

入力データ・セットの DD ステートメントの DD 名。アンロード操作では、COPY 操作ごとに 1 つの DD 名だけを指定してください。コピーまたはロード操作について複数の DD 名を指定した場合、入力データ・セットが処理されるシーケンスは DD 名が指定されたシーケンスと同じになります。

R この入力データ・セットからコピーまたはロードされるすべてのメンバーが、出力区分データ・セット上の同名のメンバーを置換するように指定します。(さらに、出力データ・セット上に名前のないメンバーは、通常どおりコピーまたはロードされます。) このオプションを指定した場合、DD 名および R パラメーターを 1 組の括弧で囲まなければなりません。INDD 内の最初の DD 名を使用して指定した場合、INDD パラメーターを除くフィールド全体を 2 つ目の括弧の組で囲みます。

LIST={YESINO}

コピーするメンバー名を各入力データ・セットの最後に SYSPRINT データ・セットにリスト表示するかどうかを指定します。このパラメーターを省略すると、EXEC PARM フィールドのデフォルトが適用されます。

COPY の使用上の注意: 制御ステートメントの操作およびキーワード・パラメーターは最初の文字を省略形とすることができます。たとえば、COPY の省略形は C、OUTDD の省略形は O です。

COPY レコードに OUTDD 以外のキーワードが存在しない場合、注釈はレコード上に置かれないことがあります。

COPY を使用して、固定長ブロックまたは可変長ブロック・レコードのあるデータ・セットをコピーおよび再ブロックできます。COPY を使用して不定形式レコードまたはロード・モジュール・ライブラリーを再ブロックすることはできません。ロード・ライブラリーをコピーおよび再ブロックするには、代わりに COPYMOD を使用してください。

1 つのレコード上に置けるのは、INDD キーワードが 1 つ、OUTDD キーワードが 1 つだけです。OUTDD は COPY ステートメント上に必要です。INDD が別のレ

コード上にあるとき、そのレコード上には他のオペランドを指定することができず、INDD の前にはコンマが付きません。

コピー、アンロード、またはロード・ステップでメンバーが選択または除外される
とき、COPY ステートメントの前に SELECT または EXCLUDE ステートメント
が必要です。さらに、入力 DD 名を別の INDD ステートメント上に指定する場
合、それは COPY ステートメントの直後、そして対応する SELECT または
EXCLUDE ステートメントの前になければなりません。

COPYGRP ステートメント

グループのコピー、アンロード、またはロードを開始するには、COPYGRP ステ
ートメントを使用します。1 つのグループは、1 つのメンバーおよびそのすべての別
名から構成されます。COPYGRP はグループを単一のエンティティとして扱いま
す。

COPYGRP ステートメントの構文は次のとおりです。

[label]	COPYGRP	OUTDD=DDname ,INDD={DDname((DDname,R))} [,LIST={YES NO}]
---------	----------------	---------------------------------------------------------------------------

各項目の意味は以下のとおりです。

OUTDD=DDname

出力データ・セットを位置指定する DD ステートメントの名前を指定します。

INDD={DDname ((DDname,R))}

入力データ・セットを位置指定する DD ステートメントの名前を指定します。

制限:複数の INDD ステートメントは COPYGRP では使用できません。

コーディングできる値は以下のとおりです。

DDname

DD ステートメント上で指定されている、入力データ・セットの DD 名を
指定します。

R この入力データ・セットからコピーまたはロードされるグループが、出力デ
ータ・セット上のグループを置換するように指定します。このオプションを
指定した場合、DD 名および **R** パラメーターを 1 組の括弧で囲まなければ
なりません。そして、INDD パラメーターを除くフィールド全体を 2 つ目
の括弧の組で囲みます。

LIST={YES|NO}

コピーするメンバー名を各入力データ・セットの最後に SYSPRINT データ・セ
ットにリスト表示するかどうかを指定します。このパラメーターを省略する
と、EXEC PARM フィールドのデフォルトが適用されます。

COPYGRP の使用上の注意: この制御ステートメントの省略形は CG です。
COPYGRP 制御ステートメントごとに、使用できる入力 DD 名は 1 つだけです。

INDD および OUTDD キーワードは、COPYGRP ステートメント上に必要です。

コピー、アンロード、またはロード・ステップでメンバーが選択される時、COPYGRP ステートメントの前に SELECT ステートメントが必要です。COPYGRP の直後の SELECT ステートメントには、**R** (置換) パラメーターを含めることができません。

EXCLUDE ステートメントは、COPYGRP ステートメントの後に置くことができません。

COPYMOD ステートメント

COPYMOD ステートメントを使用して、再ブロックしたいロード・モジュール・ライブラリーをコピーできます。出力データ・セットは区分編成でなければなりません。入力データ・セットには、アンロード操作で作成された区分または順次データ・セットを使用できます。COPYMOD への入力が有効なロード・モジュールではない場合、COPYMOD は COPY として機能します。

COPYMOD ステートメントの構文は次のとおりです。

[label]	COPYMOD	OUTDD=DDname ,INDD=([{DDname}(DDname,R)][...][D] [,MAXBLK={nnnnnlnnK}] [,MINBLK={nnnnnlnnK}] [,LIST={YES NO}]
---------	----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

OUTDD=DDname

出力データ・セットを位置指定する DD ステートメントの名前を指定します。

INDD=([{DDname}(DDname,R)][...][D]

入力データ・セットを位置指定する DD ステートメントの名前を指定します。

INDD= が単独でレコード内にある (つまり COPY キーワードを伴わない) 場合は、制御ステートメントとして機能して、現行のコピー操作内に新規のステップを開始します。

コーディングできる値は以下のとおりです。

DDname

DD ステートメント上で指定されている、ロード・モジュール・ライブラリーである入力データ・セットの DD 名を指定します。

R この入力データ・セットからコピーされるすべてのメンバーが、出力ロード・モジュール・ライブラリー上の同名のメンバーを置換するように指定します。(さらに、出力データ・セット上に名前のないメンバーは、通常どおりコピーされます。) このオプションを指定した場合、DD 名および **R** パラメーターを 1 組の括弧で囲まなければなりません。INDD 内の最初の DD 名を使用して指定した場合、INDD パラメーターを除くフィールド全体を 2 つ目の括弧の組で囲みます。

MAXBLK={nnnnnlnnK}

出力区分データ・セット内にあるレコードの最大ブロック・サイズを指定しま

す。MAXBLK は通常、データ・セットのレコード長に異なる装置またはプログラムとの互換性を持たせるため、デフォルトよりも小さいブロック・サイズの指定に使用します。

nnnnn 値は 10 進数として指定されます。K は *nn* 値を 1024 バイトで乗算するように指示します。

MAXBLK と共に MINBLK を指定することも、指定しないこともできます。指定した MAXBLK が、ターゲット・データ・セットのブロック・サイズよりも大きい場合は、データ・セットのブロック・サイズは、MAXBLK 値に変更されます。

デフォルト: COPYMOD MAXBLK パラメーターのデフォルトは、出力データ・セットのブロック・サイズです。出力データ・セットのブロック・サイズがゼロの場合、デフォルトは入力データ・セットのブロック・サイズです。

MINBLK={nnnnnlnnK}

出力区分データ・セット内にあるレコードの最小ブロック・サイズを指定します。MINBLK は、トラックの最後に書き込む最小のブロックを指定します。

MINBLK キーワードは、もっと大きく、トラック・サイズよりも小さい MINBLK 値によってモジュールのプログラム取り出しパフォーマンスを強化できる、より初期の MVS リリースとの互換性のために提供されています。通常の状態下では、MINBLK を指定しないでください。

nnnnn 値は 10 進数として指定されます。K は *nn* 値を 1024 バイトで乗算するように指示します。

MINBLK と共に MAXBLK を指定することも、指定しないこともできます。

デフォルト: 1K (1024)。MAXBLK よりも大きな値を指定した場合、MINBLK は実際に使用される MAXBLK 値 (指定した値またはデフォルト値) に設定されます。1K より小さな値を指定した場合、MINBLK は 1K に設定されます。

LIST={YESINO}

コピーするメンバー名を各入力データ・セットの最後に SYSPRINT データ・セットにリスト表示するかどうかを指定します。このパラメーターを省略すると、EXEC PARM フィールドのデフォルトが適用されます。

INDD= ステートメント

さらに、COPY ステートメントのパラメーター INDD が COPY ステートメント以外のレコード上に存在する場合、それは INDD ステートメントとして参照されません。また、このコンテキストでは、制御ステートメントとして機能します。

1 つまたは複数の INDD ステートメントの直後にファイル終わり (EOF)、または他の COPY、COPYMOD、または ALTERMOD ステートメントがある場合、フル・コピー、アンロード、またはロード操作は最後に指定された出力データ・セットを使用して完了します。

EXCLUDE ステートメント

EXCLUDE ステートメントは、コピー、アンロード、またはロード・ステップからメンバーを除外するように指定します。各 EXCLUDE ステートメントで指定されたメンバーを除く入力データ・セット内のすべてのメンバーが操作の対象となりま

す。複数の EXCLUDE ステートメントを連続して使用できます。その場合、2 番目およびそれに続くステートメントは 1 番目からの続きとして処理されます。

EXCLUDE ステートメントの直前には、COPY ステートメント、ALTERMOD、COPYMOD ステートメント、または 1 つ以上の INDD= ステートメントが必要です。同一のコピー、アンロード、またはロード・ステップで、EXCLUDE ステートメントと SELECT ステートメントを同時に指定することはできません。

EXCLUDE ステートメントは同所圧縮または COPYGRP では使用できません。

SELECT も EXCLUDE も指定されない場合、データ・セットの全体がコピーされます (『フル・コピー』)。

EXCLUDE ステートメントの構文は次のとおりです。

[label]	EXCLUDE	MEMBER=[(]name1[,name2][,...]D]
---------	---------	---------------------------------

各項目の意味は以下のとおりです。

MEMBER=[(]name1[, name2][,...]D]

入力データ・セット内のメンバーで、出力データ・セットにコピー、アンロード、またはロードされないものを指定します。それらのメンバーは入力データ・セットから削除されません。

制御ステートメント操作およびキーワード・パラメーターは最初の文字を省略形とすることができます。EXCLUDE の省略形は E、MEMBER の省略形は M です。

EXCLUDE ステートメント上の各メンバー名は固有のものでなければなりません。

SELECT ステートメント

SELECT ステートメントは、入力データ・セットから選択して出力データ・セットに変更、コピー、ロード、またはアンロードするコピー・メンバーを指定します。このステートメントはさらに、出力データ・セット上の選択されたメンバーを名前変更または置換するためにも使用します。複数の SELECT ステートメントを連続して使用できます。その場合、2 番目およびそれに続くステートメントは 1 番目からの続きとして処理されます。

SELECT ステートメントの直前には、COPY ステートメント、COPYGRP ステートメント、COPYMOD ステートメント、または 1 つまたは複数の INDD ステートメントが必要です。同一のコピー、アンロード、またはロード・ステップで、SELECT ステートメントと EXCLUDE ステートメントを同時に指定することはできません。さらに、SELECT ステートメントを同所圧縮機能と共に使用することはできません。INDD と OUTDD ステートメントの両方が同一データ・セットを指している場合、SELECT ステートメント内のメンバーは同一データ・セット内で新規の名前に置換またはコピーされます。これは、置換または新規名オプションに基づいて行われます。

選択されたメンバーが入力データ・セットで見つかり、そのコピー・ステップではそれは再び検索されません。

SELECT ステートメント上の各メンバー名は固有のものでなければなりません。

置換オプションがデータ・セットまたはメンバー・レベルのいずれかに指定されていなければ、選択されたメンバーが出力データ・セット内の同名のメンバーを置換することはありません。(メンバーの置換に関する説明は、30 ページの『データ・セット内のメンバーを置換する』を参照してください。) さらに、置換オプションが指定されていなければ、名前変更されたメンバーが出力データ・セット内の名前変更されたメンバーと同名のメンバーを置換することはありません。

SELECT ステートメントの構文は次のとおりです。

[label]	SELECT	MEMBER={({name (name,newname[,R]) (name,,R)} [, {name2 (name2,newname2[,R]) (name2,,R)}][,...])
---------	--------	--------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

MEMBER={({name|(name, newname[,R])|(name,,R)} [,...])

入力データ・セットから選択するメンバーを指定します。

メンバーを名前変更するには、メンバーの古い名前と、その後に新しい名前、そしてオプションとして **R** (置換) パラメーターを指定します。その後、このグループを括弧で囲まなければなりません。

メンバーを置換するには、メンバーの名前および **R** パラメーターを指定します。**R** とメンバー名とは 2 つのコンマで区切り、そのグループを括弧で囲まなければなりません。

括弧内のオプションが **MEMBER** フィールドのどこかで指定されている場合、**MEMBER** キーワードを除くフィールド全体を 2 つ目の括弧の組で囲まなければなりません。

コーディングできる値は以下のとおりです。

name

処理する特定のメンバーを指定します。1 つのコピー・ステップで指定するすべての名前および新規の名前は固有のものでなければなりません。どのような状況でも、古い名前または新規の名前、もしくはその両方を重複させることはできません。A を B に、そして B を C に名前変更することは、B が 2 回出現するので、行えません。長さが 8 文字を超える名前を指定することはできません。

newname

選択した名前に対する新規の名前を指定します。メンバー名は、A～Z、0～9、または \$ # @ _、または { で構成することができ、長さが 8 文字を超えることはできません。メンバーはその新規の名前を使用して出力データ・セットにコピー、アンロード、またはロードすることができます。その名前がすでに出力区分データ・セットまたは PDSE にある場合、置換が指定されていなければメンバーはコピーされません。*newname* を **ALTERMOD** と共に指定することはできません。

R 入力メンバーが出力データ・セットに存在する同名のメンバーを置換するよ

うに指定します。入力メンバーの名前が出力メンバーの別名と同じである場合、その名前は古いメンバーではなく新しいメンバーを参照します。

R を ALTERMOD または COPYGRP と共にコーディングすることはできません。

制御ステートメント操作およびキーワード・パラメーターは最初の文字を省略形とすることができます。たとえば、SELECT の省略形は S、MEMBER の省略形は M です。

IEBCOPY の例

以下の各例では、IEBCOPY の使用方法の一部を示します。表 12 は IEBCOPY の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 12. IEBCOPY の例の一覧表

操作	装置	注釈	例
同じ場所での変更	ディスク	選択したメンバーを同じ場所に変更します。	10
PDSE への変換	ディスク	区分データ・セットを PDSE に変換します。	13
COPY	ディスク	1 つのディスク・ボリュームから別のディスク・ボリュームにデータ・セット全体をコピーします。	1
COPY	ディスク	3 つの入力データ・セットを既存の出力データ・セットに 2 コピーします。	2
COPY	ディスク	PDSE を PDSE にコピーします。	14
COPY	ディスク	2 つの入力データ・セットからメンバーを選択して、それらを既存の出力データ・セットにコピーします。1 つのメンバーは、出力データ・セットにすでに存在する同名のメンバーを置換します。	3
COPY	ディスク	複数の入力データ・セットから 1 つの出力データ・セットにメンバーを選択、除外、およびコピーします。複数のコピー操作を例示します。	6
COPY	ディスク	複数の入力データ・セットから複数の異なる出力データ・セットにメンバーを選択、除外、およびコピーします。複数のコピー操作を例示します。	7
COPY	ディスク	別名を指定して選択したグループをコピーします。	16
COPY	ディスク	置換 (R) オプションを指定して、PDSE の全体を PDSE にコピーします。	15
コピーおよび同所圧縮	ディスク	2 つの入力データ・セットを既存の出力データ・セットに 5 コピーして、それらを同じ場所で圧縮します。データ・セットのメンバーすべてをコピーおよび置換します。出力データ・セットのメンバーは、置換されたものと同じ名前になります。	5
コピーおよび再ブロック	ディスク (3380)	ロード・ライブラリーを最適なブロック・サイズの異なる複数の装置にコピーします。再ブロックは、メンバーをロード・ライブラリーに追加できるようになる前に必要です。	11

表 12. IEBCOPY の例の一覧表 (続き)

操作	装置	注釈	例
コピーおよび再ブロック	ディスク (3380)	ロード・ライブラリーをブロック・サイズの異なる複数の装置にコピーします。ライブラリーをそのライブラリーがコピーされる各装置と互換性のあるサイズに再ブロックしてから、それらの装置をコピーします。	12
ロード	磁気テープおよびディスク	順次データ・セットをディスクにロードします。	8
アンロードおよび同所圧縮	ディスクおよび磁気テープ	区分データ・セットをテープ・ボリュームにアンロードして、圧縮されたバックアップ・コピーを作成します。	4
アンロード、ロード、およびコピー	ディスクおよび磁気テープ	選択したメンバーを除外、アンロード、ロード、およびコピーします。	9

各例で、実際の装置番号の代わりに **disk** または **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: データ・セット全体をコピーする

この例では、区分データ・セット (DATASET5) を 1 つのディスク・ボリュームから他のディスク・ボリュームにコピーします。58 ページの図 3 は、処理の前後の入力および出力データ・セットを示しています。

```
//COPY      JOB      ...
//JOBSTEP   EXEC     PGM=IEBCOPY
//SYSPRINT  DD      SYSOUT=A
//SYSUT1    DD      DSN=DATASET5,UNIT=disk,VOL=SER=111113,
//           DISP=SHR
//SYSUT2    DD      DSN=DATASET4,UNIT=disk,VOL=SER=111112,
//           DISP=(NEW,KEEP),SPACE=(TRK,(5,1,2))
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、2 つのメンバー (A および C) を含む区分データ・セット、**DATASET5** を定義しています。
- **SYSUT2 DD** では、コピー操作後に保管する新規の区分データ・セット **DATASET4** を定義しています。5 つのトラックがデータ・セットに割り振られます。2 つのブロックがディレクトリー項目に割り振られます。
- 区分データ・セットには 2 つのメンバーしかないので、**SYSUT3** および **SYSUT4 DD** は必要ありません。
- 入力および出力データ・セットは **SYSUT1** および **SYSUT2** として識別されるので、**SYSIN** データ・セットは必要ありません。**SYSUT1** データ・セットは全体が **SYSUT2** データ・セットにコピーされます。コピー操作が終了した後、**DATASET4** には **DATASET5** と同じメンバーが含まれています。しかし、**DATASET4** には組み込まれた未使用のスペースがありません。**PDSE** をコピーする場合、**PDSE** には組み込まれた未使用のスペースがないことを除いて、処理は同じです。

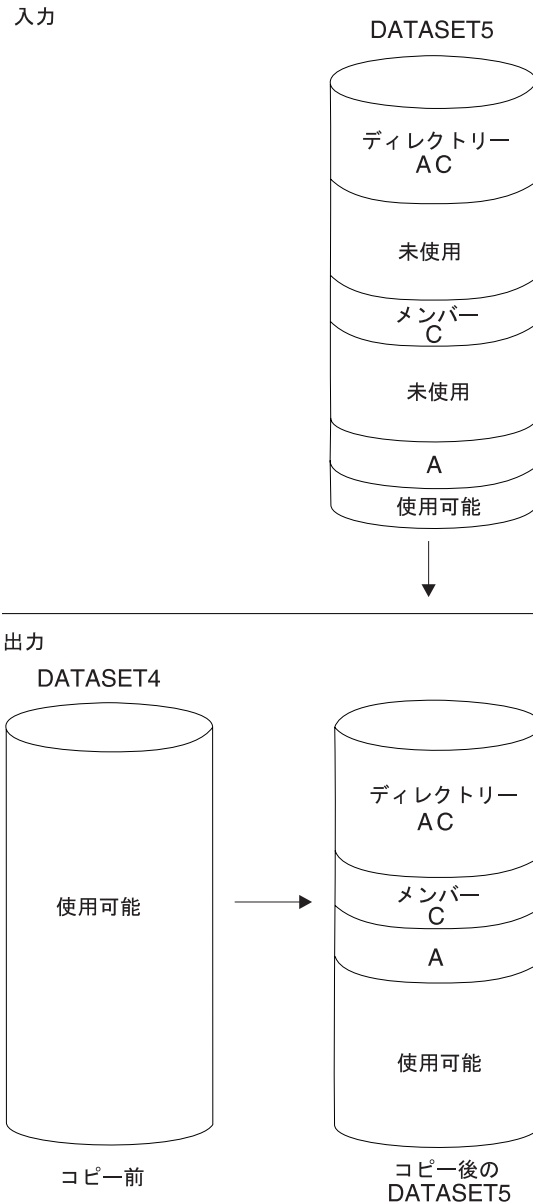
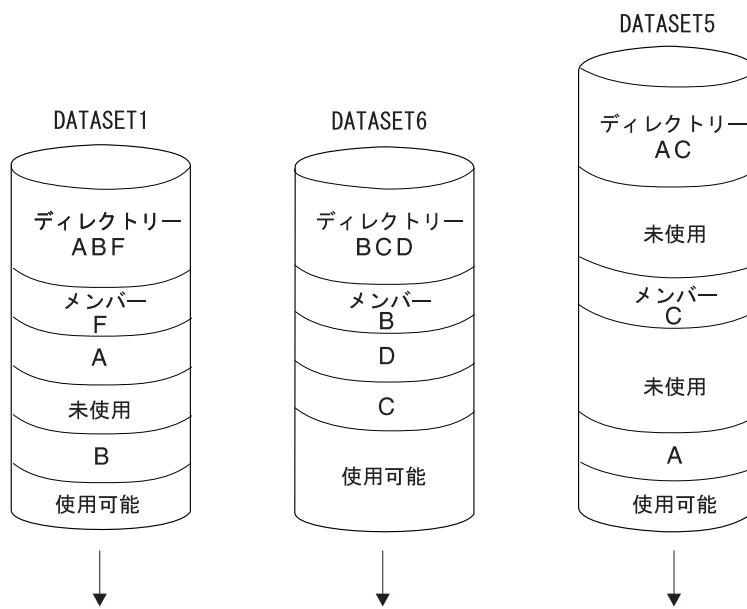


図3. 区分データ・セットをコピーする — フル・コピー

例 2: 4 つのデータ・セットをマージする

この例では、メンバーを 3 つの入力区分データ・セット (DATASET1、DATASET5、および DATASET6) から既存の出力区分データ・セット (DATASET2) にコピーします。制御ステートメントの指定順によって、区分データ・セットが処理される方法およびシーケンスが制御されます。59 ページの図 4 は、処理の前後の入力および出力データ・セットを示しています。

入力



出力

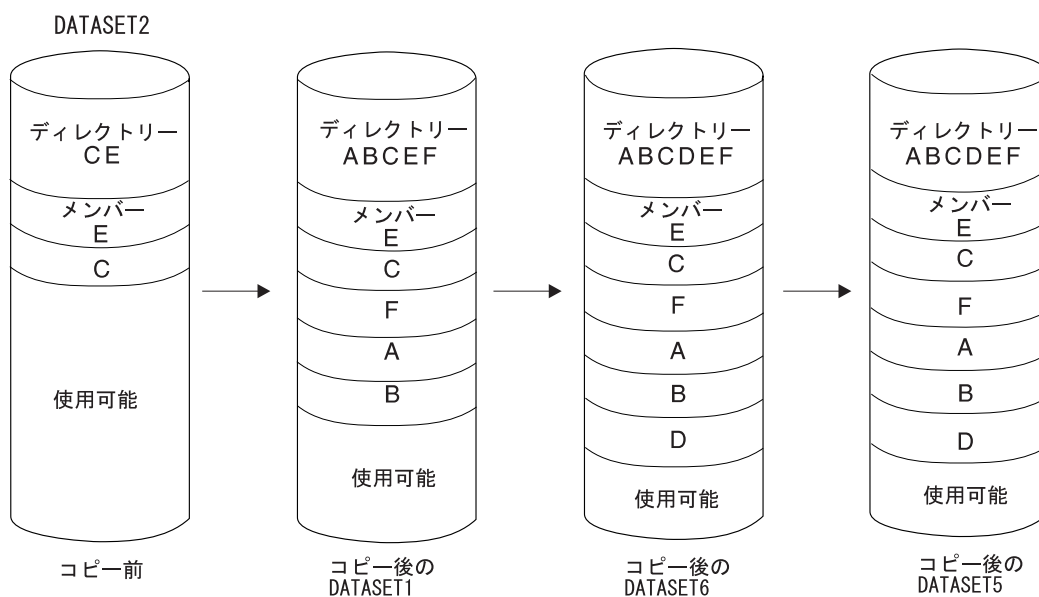


図 4. 3 つの入力区分データ・セットからコピーする

以下に例を示します。

```
//COPY    JOB    ...
//JOBSTEP EXEC  PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//IN1     DD   DSNAME=DATASET1,UNIT=disk,VOL=SER=111112,
//         DISP=SHR
//IN5     DD   DSNAME=DATASET5,UNIT=disk,VOL=SER=111114,
//         DISP=OLD
//OUT2    DD   DSNAME=DATASET2,UNIT=disk,VOL=SER=111115,
//         DISP=(OLD,KEEP)
//IN6     DD   DSNAME=DATASET6,UNIT=disk,VOL=SER=111117,
//         DISP=(OLD,DELETE)
//SYSUT3  DD   UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN   DD   *
COPYOPER  COPY  OUTDD=OUT2
```

```
INDD=IN1
INDD=IN6
INDD=IN5
```

```
/*
```

各制御ステートメントについての説明は以下のとおりです。

- IN1 DD では、区分データ・セット (DATASET1) を定義しています。このデータ・セットには、論理レコード長が 80 バイトでブロック・サイズが 80 バイトの固定長フォーマットのメンバー 3 つ (A、B、および F) が含まれます。
- IN5 DD では、区分データ・セット (DATASET5) を定義しています。このデータ・セットには、論理レコード長が 80 バイトでブロック・サイズが 160 バイトの固定長ブロック・フォーマットのメンバー 2 つ (A および C) が含まれます。
- OUT2 DD では、区分データ・セット (DATASET2) を定義しています。このデータ・セットには、固定長ブロック・フォーマットのメンバー 2 つ (C および E) が含まれます。メンバーの論理レコード長は 80 バイト、ブロック・サイズは 240 バイトです。
- IN6 DD では、区分データ・セット (DATASET6) を定義しています。このデータ・セットには、論理レコード長が 80 バイトでブロック・サイズが 400 バイトの固定長ブロック・フォーマットのメンバー 3 つ (B、C、および D) が含まれます。このデータ・セットは、処理が完了したときに削除されます。
- SYSUT3 では、一時スピル・データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、COPY ステートメントおよび 3 つの INDD ステートメントが含まれます。
- COPY はコピー操作の開始を指示します。OUTDD パラメーターは、DATASET2 を出力データ・セットとして指定します。
- 最初の INDD ステートメントは、DATASET1 を最初に処理する入力データ・セットとして指定します。すべてのメンバー (A、B、および F) は DATASET2 にコピーされます。
- 2 番目の INDD ステートメントは、DATASET6 を 2 番目に処理する入力データ・セットとして指定します。処理は以下のように実行されます。
 1. 置換が指定されていないので、DATASET2 に既に存在するメンバー B および C は、DATASET2 にコピーされません。
 2. メンバー D が DATASET2 にコピーされます。
 3. データ・セットが削除されると、DATASET6 内のすべてのメンバーは失われます。
- 3 番目の INDD ステートメントは、DATASET5 を 3 番目に処理する入力データ・セットとして指定します。すべてのメンバーは DATASET2 に存在するため、DATASET2 にコピーされるメンバーはありません。

例 3: データ・セットから選択したメンバーをコピーおよび置換する

この例では、2 つのメンバー (A および B) を 2 つの入力区分データ・セット (DATASET5 および DATASET6) から選択して、既存の出力区分データ・セット (DATASET1) にコピーします。メンバー B は、出力データ・セットにすでに存在

する同名のメンバーを置換します。図5は、処理の前後の入力および出力データ・セットを示しています。

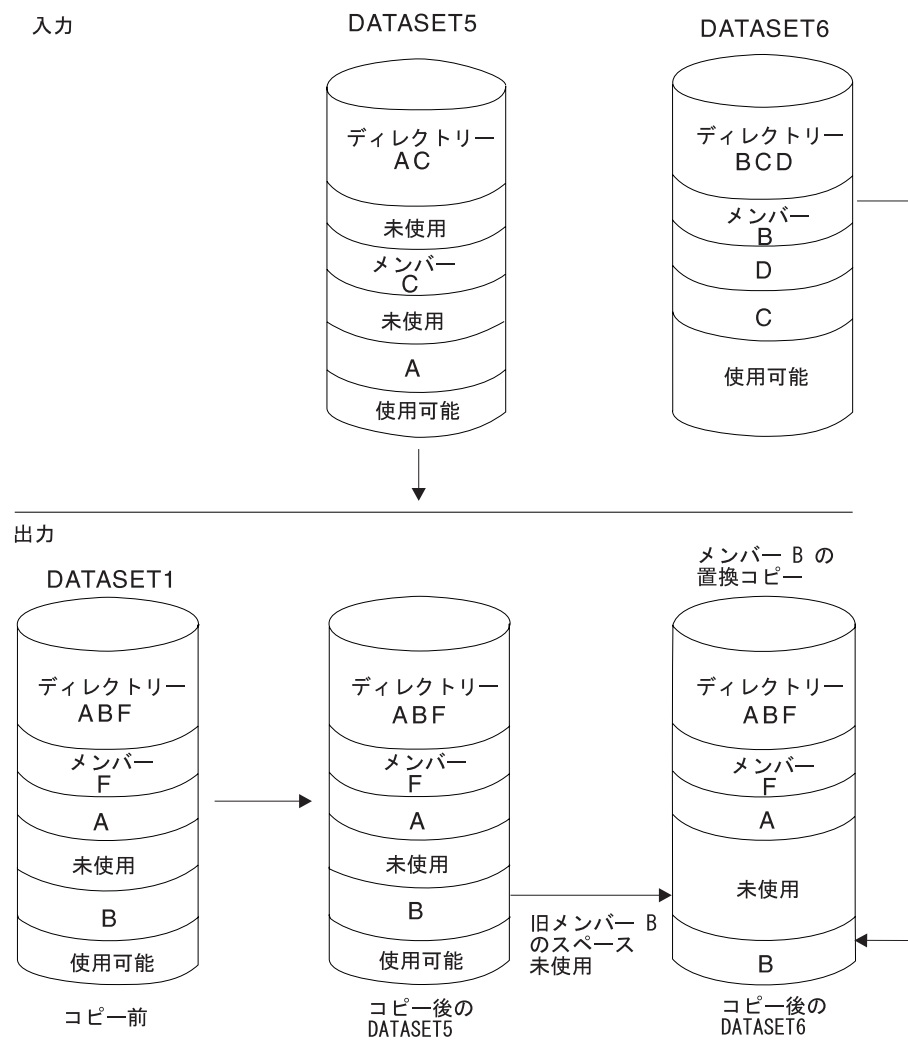


図5. メンバー・レベルで置換を指定した選択的なコピー

```
//COPY JOB ...
//JOBSTEP EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//OUT1 DD DSN=DATASET1,UNIT=disk,VOL=SER=111112,
// DISP=(OLD,KEEP)
//IN6 DD DSN=DATASET6,UNIT=disk,VOL=SER=111115,
// DISP=OLD
//IN5 DD DSN=DATASET5,UNIT=disk,VOL=SER=111116,
// DISP=(OLD,KEEP)
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN DD *
COPYOPER COPY OUTDD=OUT1
INDD=IN5,IN6
SELECT MEMBER=((B,,),A)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- OUT1 DD では、3 つのメンバー (A、B、および F) を含む区分データ・セット (DATASET1) を定義しています。
- IN6 DD では、3 つのメンバー (B、C、および D) を含む区分データ・セット (DATASET6) を定義しています。
- IN5 DD では、2 つのメンバー (A および C) を含む区分データ・セット (DATASET5) を定義しています。
- SYSUT3 および SYSUT4 DD では、一時スピル・データ・セットを定義しています。ディスク・ボリューム上のそれぞれについて、1 つのトラックが割り振られます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、COPY ステートメント、INDD ステートメント、および SELECT ステートメントが含まれます。
- COPY はコピー操作の開始を指示します。SELECT ステートメントを使用すると、選択的なコピーが行われます。OUTDD パラメーターは、DATASET1 を出力データ・セットとして指定します。
- INDD は、DATASET5 を最初に処理する入力データ・セットとして指定し、DATASET6 を 2 番目に処理する入力データ・セットとして指定します。処理は以下のように実行されます。
 1. 選択したメンバーが DATASET5 で検索されます。
 2. メンバー A が見つかりますが、DATASET1 には既に『A』という名前のメンバーが存在していて、置換オプションがメンバー A に指定されていないため、DATASET1 にコピーされません。
 3. 選択したメンバーで DATASET5 で見つからないものは、DATASET6 で検索されます。
 4. 置換オプションがメンバー・レベルでメンバー B に指定されているため、DATASET1 のメンバー『B』が DATASET1 に既に存在していても、メンバー B が見つかって DATASET1 にコピーされます。DATASET1 のディレクトリーのポインターは新規の (コピーされた) メンバー B を指すように変更されます。したがって、古いメンバー B によって占められていたスペースは未使用となります。
- SELECT は、入力データ・セット (DATASET5 および DATASET6) から選択して出力データ・セット (DATASET1) にコピーするメンバーを指定します。

例 4: データ・セットをアンロードおよび圧縮する

この例では、区分データ・セットをテープ・ボリュームにアンロードして、データ・セットのバックアップ・コピーを作成します。このステップが正常に終了すると、その区分データ・セットは同じ場所で圧縮されます。

```
//SAVE      JOB    ...
//STEP1     EXEC   PGM=IEBCOPY
//SYSPRINT  DD     SYSOUT=A
//SYSUT1    DD     DSN=PARTPDS,UNIT=disk,VOL=SER=PCP001,
//           DISP=OLD
//SYSUT2    DD     DSN=SAVDATA,UNIT=tape,VOL=SER=TAPE03,
//           DISP=(NEW,KEEP),LABEL=(,SL)
//SYSUT3    DD     DSN=TEMP1,UNIT=disk,VOL=SER=111111,
//           DISP=(NEW,DELETE),SPACE=(80,(60,45))
//SYSIN     DD     DUMMY
//STEP2     EXEC   PGM=IEBCOPY,COND=(0,NE),PARM='SIZE=500K'
```



```
//SYSRINT DD SYSOUT=A
//COMPDS DD DSNAME=PARTPDS,UNIT=disk,DISP=OLD,
//          VOL=SER=PCP001
//SYSUT3 DD DSNAME=TEMPA,UNIT=disk,VOL=SER=111111,
//          DISP=(NEW,DELETE),SPACE=(80,(60,45))
//SYSIN DD *
COPY OUTDD=COMPDS,INDD=COMPDS
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、ディスク・ボリューム上に存在して 700 のメンバーを持つと想定される区分データ・セット (PARTPDS) を定義しています。メンバーの数は、SYSUT3 のスペース割り振りを計算するために使用されます。
- **SYSUT2 DD** では、PARTPDS をアンロード・フォームで保持するための順次データ・セットを定義しています。オプションとして、ブロック・サイズ情報を追加できます。このデータ・セットは新規のものでなければなりません。
- **SYSUT3 DD** では、一時スピル・データ・セットを定義しています。SYSUT4 データ・セットがアンロード操作に使用されることはありません。
- **SYSIN DD** では、制御データ・セットを定義しています。SYSIN がダミーとされていて、SYSUT2 が順次データ・セットを定義するので、SYSUT1 データ・セットのすべてのメンバーは SYSUT2 データ・セットにアンロードされます。
- 2 番目の EXEC ステートメントは、同所圧縮操作の開始を示します。SIZE パラメーターは、バッファーを可能な限り大きくするように指示しています。COND パラメーターは、アンロード操作が正常に終了した場合にだけ同所圧縮を実行するように指示しています。
- **COMPDS DD** では、700 のメンバーを含み、ディスク・ボリューム上に存在する区分データ・セット (PARTPDS) を定義しています。
- **SYSUT3 DD** では、入力データ・セットのディレクトリー項目用に十分なスペースが主記憶域にない場合に使用する、一時スピル・データ・セットを定義しています。TEMPA は、メンバーごとに 80 文字のレコードを 1 つ含みます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、COPY ステートメントが含まれます。
- **COPY** はコピー操作の開始を示します。同じ DD ステートメントが INDD および OUTDD オペランドの両方に指定されているので、データ・セットは同じ場所で圧縮されます。PDSE が使用されている場合、このステップは処理されません。

IEBCOPY の 1 回の使用で複数のデータ・セットをアンロードしたい場合、アンロード操作ごとに別個の COPY ステートメントを使用しなければなりません。1 つのアンロード操作で指定できる入力データ・セットは 1 つだけです。

例 5: データ・セットをマージして、マージしたデータ・セットを圧縮する

この例では、2 つの入力区分データ・セット (DATASET5 および DATASET6) を既存の出力区分データ・セット (DATASET1) にコピーします。さらに、DATASET6 上のすべてのメンバーがコピーされます。出力データ・セット上のメンバーでコピーされたメンバーと同名の場合は、置換されます。DATASET6 が処理された後、

出力データ・セット (DATASET1) は同じ場所で圧縮されます。 図 6 は、処理の前後の入力および出力データ・セットを示しています。

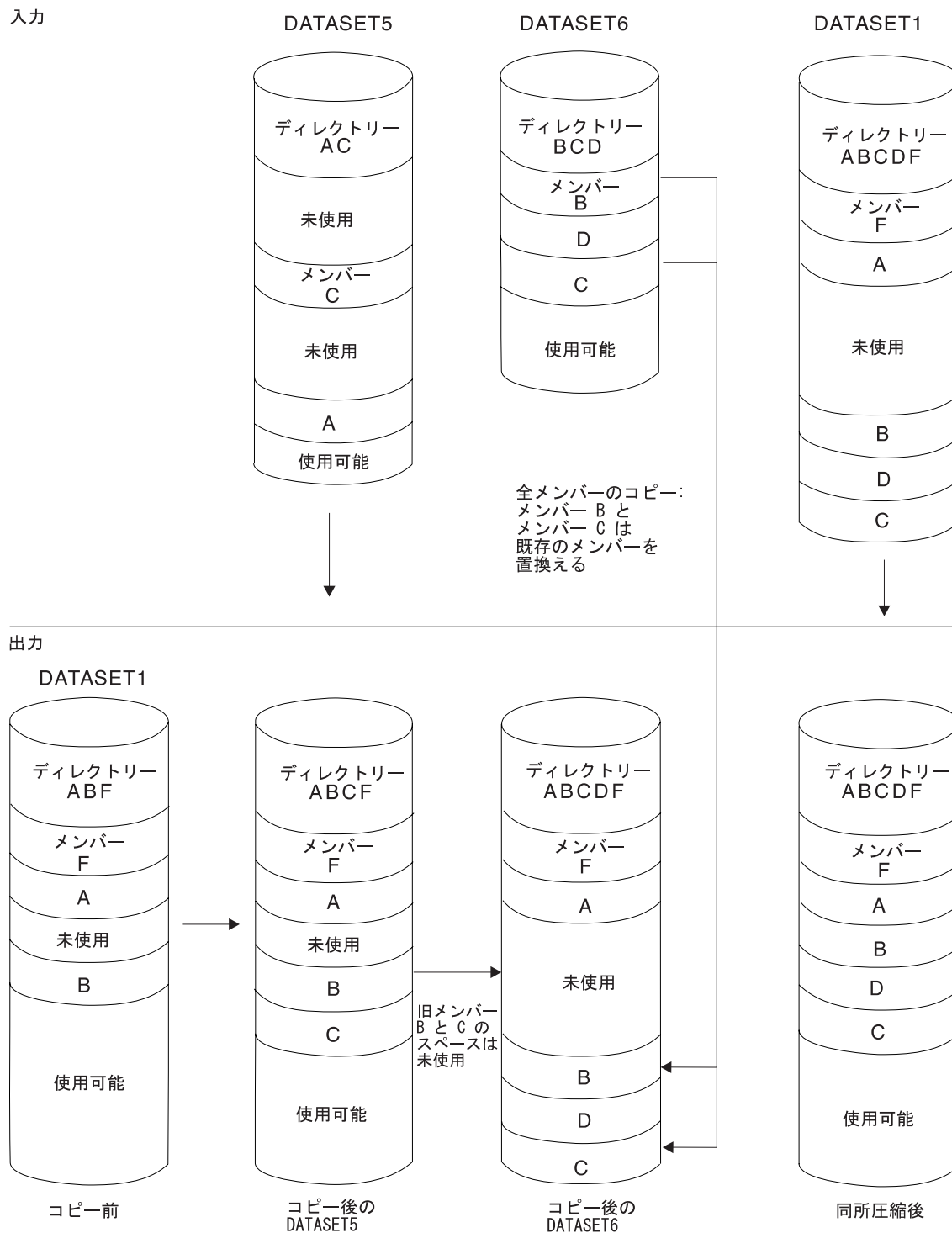


図 6. “置換” を指定したフル・コピーの後、同所圧縮を行う

```
//COPY      JOB ...
//JOBSTEP   EXEC PGM=IEBCOPY
//SYSPRINT  DD  SYSOUT=A
//INOUT1    DD  DSN=DATASET1,UNIT=disk,VOL=SER=111112,
//          DISP=(OLD,KEEP)
```

```
//IN5      DD  DSN=DATASET5,UNIT=disk,VOL=SER=111114,
//          DISP=OLD
//IN6      DD  DSN=DATASET6,UNIT=disk,VOL=SER=111115,
//          DISP=(OLD,KEEP)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(TRK,(1))
//SYSUT4   DD  UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN    DD  *
COPYOPER  COPY  OUTDD=INOUT1,INDD=(IN5,(IN6,R),INOUT1)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- INOUT1 DD では、3 つのメンバー (A、B、および F) を含む区分データ・セット (DATASET1) を定義しています。
- IN5 DD では、2 つのメンバー (A および C) を含む区分データ・セット (DATASET5) を定義しています。
- IN6 DD では、3 つのメンバー (B、C、および D) を含む区分データ・セット (DATASET6) を定義しています。
- SYSUT3 および SYSUT4 DD では、一時スピル・データ・セットを定義しています。ディスク・ボリューム上のそれぞれについて、1 つのトラックが割り振られます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、COPY ステートメントが含まれます。
- COPY はコピー操作の開始を指示します。OUTDD オペランドは、DATASET1 を出力データ・セットとして指定します。

INDD オペランドは、DATASET5 を最初に処理する入力データ・セットとして指定します。その後、DATASET6 を 2 番目に処理する入力データ・セットとして指定します。さらに、置換オプションが DATASET6 からコピーされるすべてのメンバーに対して指定されます。最後に、DATASET1 を最後に処理する入力データ・セットとして指定します。DATASET1 は出力データ・セットでもあるので、DATASET1 は同じ場所で圧縮されます。しかし、DATASET1 が PDSE である場合、同所圧縮操作は処理されません。

処理は以下のように実行されます。

1. メンバー A は既に DATASET1 に存在していて、置換オプションが DATASET5 に設定されていないので、DATASET5 から DATASET1 にコピーされません。
2. メンバー C は DATASET5 から DATASET1 にコピーされて、使用可能な最初のスペースを占めます。
3. すべてのメンバーは DATASET6 から DATASET1 に、直前のメンバーのすぐ後に続いてコピーされます。メンバー B および C は、出力データ・セットが既に同名のメンバーを含んでいても、置換オプションがデータ・セット・レベルで指定されているのでコピーされます。

DATASET1 のディレクトリーのポインターは新規の (コピーされた) メンバー B および C を指すように変更されます。したがって、古いメンバー B および C によって占められていたスペースは未使用となります。現在 DATASET1 にあるメンバーは同じ場所で圧縮されて、組み込まれた未使用スペースを少なくします。

例 6: 1 つの出力データ・セットへの複数のコピー操作

この例では、入力区分データ・セットから出力区分データ・セットにメンバーを選択、除外、およびコピーします。この例は、複数のコピー操作を例示するためのものです。

以下に例を示します。68 ページの図 7 は、処理の前後の入力および出力データ・セットを示しています。

```
//COPY      JOB    ...
//JOBSTEP   EXEC   PGM=IEBCOPY
//SYSPRINT  DD     SYSOUT=A
//INOUTA    DD     DSN=DATASETA,UNIT=disk,VOL=SER=111113,
//           DISP=OLD
//INB       DD     DSN=DATASETB,UNIT=disk,VOL=SER=111115,
//           DISP=(OLD,KEEP)
//INC       DD     DSN=DATASETC,UNIT=disk,VOL=SER=111114,
//           DISP=(OLD,KEEP)
//IND       DD     DSN=DATASET D,UNIT=disk,VOL=SER=111116,
//           DISP=OLD
//INE       DD     DSN=DATASETE,UNIT=disk,VOL=SER=111117,
//           DISP=OLD
//OUTX      DD     DSN=DATASET X,UNIT=disk,VOL=SER=111112,
//           DISP=(NEW,KEEP),SPACE=(TRK,(3,1,2))
//SYSUT3    DD     UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN     DD     *
COPERST1   COPY   O=OUTX,I=INOUTA
            COPY   OUTDD=INOUTA,INDD=INOUTA
            INDD=INB
            COPY   OUTDD=INOUTA
            INDD=IND
            EXCLUDE MEMBER=MM
            INDD=INC
            SELECT MEMBER=((ML,MD,R))
            INDD=INE
/*
```

各制御ステートメントについての説明は以下のとおりです。

- INOUTA DD では、7 つのメンバー (MA、MB、MC、MD、ME、MF、および MG) を含む区分データ DATASETA を定義しています。
- INB DD では、2 つのメンバー (MA および MJ) を含む区分データ・セット DATASET B を定義しています。
- INC DD では、4 つのメンバー (MF、ML、MM、および MN) を含む区分データ・セット DATASETC を定義しています。
- IND DD では、2 つのメンバー (MM および MP) を含む区分データ・セット DATASET D を定義しています。
- INE DD では、4 つのメンバー (MD、ME、MF、および MT) を含む区分データ・セット DATASETE を定義しています。
- OUTX DD では、区分データ・セット (DATASET X) を定義しています。このデータ・セットは新規のものであり、コピー操作の後も保持されます。ディスク・ボリューム上のデータ・セットには、3 つのトラックが割り振られます。2 つのブロックがディレクトリー項目に割り振られます。
- SYSUT3 では、一時スピル・データ・セットを定義しています。

- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには 2 つの COPY ステートメント、いくつかの INDD ステートメント、SELECT ステートメント、および EXCLUDE ステートメントが含まれます。
- 最初の COPY ステートメントは、最初のコピー操作の開始を指示します。このコピー操作は、DATASETA のバックアップ・コピーを作成するために使用します。
- 2 番目の COPY ステートメントは、別のコピー操作の開始を指示します。DATASETA が INDD および OUTDD パラメーターの両方に指定されているため、DATASETA は同じ場所で圧縮されます。

出力データ・セットには組み込まれた未使用スペースが存在することが分かっているため、同じ場所での圧縮は第一にスペースの節約を目的としています。

以下の INDD ステートメントは、次にコピーする入力データ・セットとして DATASET B を指定します。DATASETA には既に『MA』という名前のメンバーが含まれているので、メンバー MJ だけがコピーされます。

- 3 番目の COPY ステートメントは、3 番目のコピー操作の開始を指示します。OUTDD パラメーターは、DATASETA を出力データ・セットとして指定します。このコピー操作には、複数のコピー・ステップが含まれます。

最初の INDD ステートメントは、DATASET D を最初に処理する入力データ・セットとして指定します。EXCLUDE ステートメントがメンバー MM をこのコピー操作内の最初のコピー・ステップから除外するように指定しているため、メンバー MP だけが DATASETA にコピーされます。

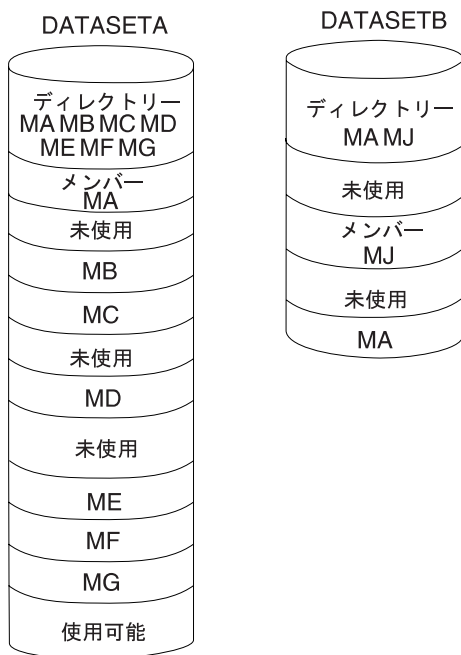
2 番目の INDD ステートメントは、このコピー操作での 2 番目のコピー・ステップの開始を示して、DATASET C を 2 番目に処理する入力データ・セットとして指定します。SELECT ステートメントは DATASET C のメンバー ML が『MD』に名前変更されるように、および新規のメンバーが DATASETA 内の『MD』という名前のメンバーを置換するように指定します。

3 番目の INDD ステートメントは、このコピー操作での 3 番目のコピー・ステップの開始を示して、DATASET E を 3 番目にコピーする入力データ・セットとして指定します。DATASETA には既に他のメンバーが含まれているので、メンバー MT だけがコピーされます。INDD ステートメントの後に EXCLUDE または SELECT ステートメントがないので、フル・コピーが実行されます。

IEBCOPY

同所圧縮

入力



出力

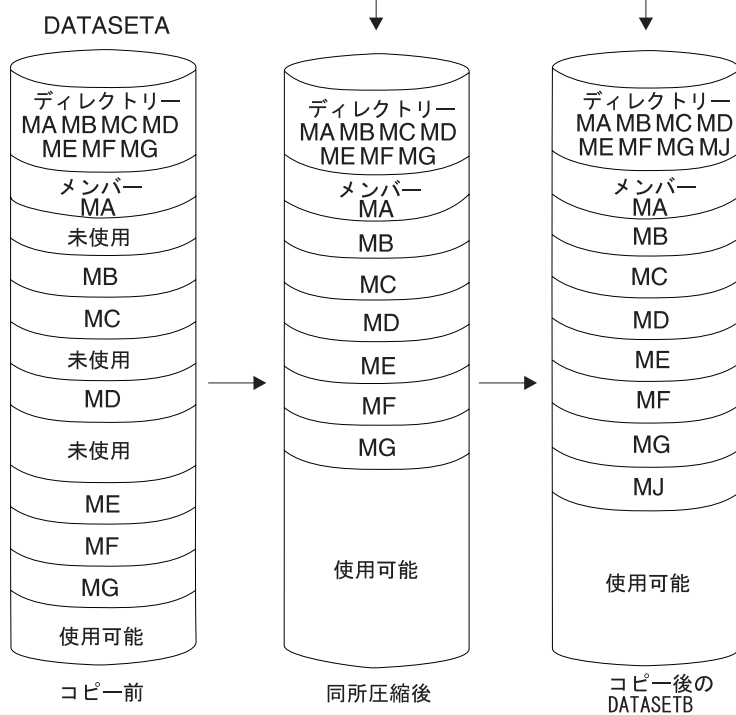
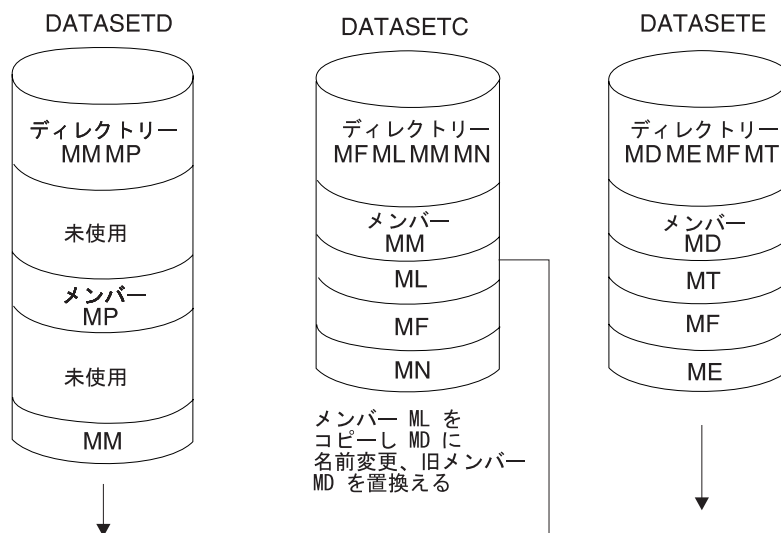


図7. 複数のコピー操作 / コピー・ステップ (1/2)

複数コピー

入力



出力

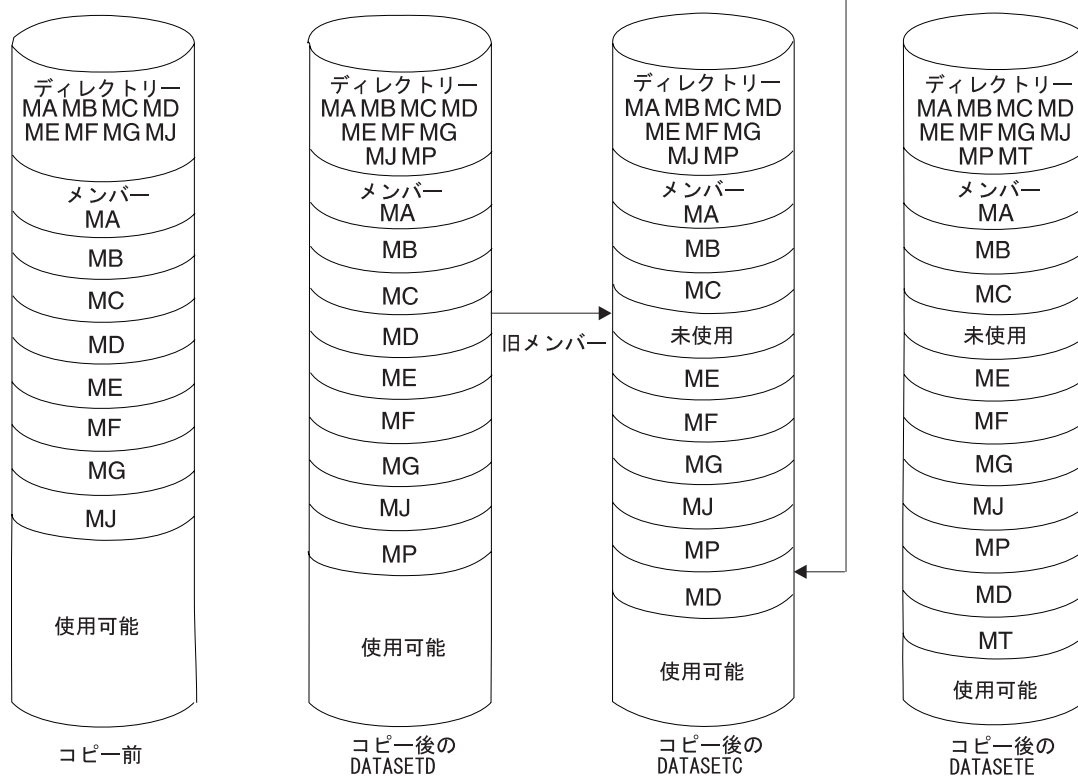


図 7. 複数のコピー操作 / コピー・ステップ (2/2)

例 7: 複数の異なる出力データ・セットへの複数のコピー操作

この例では、入力区分データ・セットから出力区分データ・セットにメンバーを選択、除外、およびコピーします。この例は、複数のコピー操作を例示するためのものです。72 ページの図 8 は、処理の前後の入力および出力データ・セットを示しています。

以下に例を示します。

```
//COPY      JOB    ...
//JOBSTEP   EXEC   PGM=IEBCOPY
//SYSPRINT  DD    SYSOUT=A
//OUTA      DD    DSN=DATASETA,UNIT=disk,VOL=SER=111113,
//           DISP=OLD
//INOUTB    DD    DSN=DATASETB,VOL=SER=111115,UNIT=disk,
//           DISP=(OLD,KEEP)
//INOUTC    DD    DSN=DATASETC,VOL=SER=111114,UNIT=disk,
//           DISP=(OLD,KEEP)
//INOUTD    DD    DSN=DATASET D,VOL=SER=111116,DISP=OLD,
//           UNIT=disk
//INE       DD    DSN=DATASETE,VOL=SER=111117,DISP=OLD,
//           UNIT=disk
//SYSUT3    DD    UNIT=SYSDA,SPACE=(TRK,(1))
//SYSUT4    DD    UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN     DD    *
           COPY  OUTDD=OUTA
           INDD=INE
           SELECT MEMBER=(MA,MJ)
           INDD=INOUTC
           EXCLUDE MEMBER=(MM,MN)
           COPY  OUTDD=INOUTB,INDD=INOUTD
           INDD=((INOUTC,R),INOUTB)
           COPY  OUTDD=INOUTD,INDD=((INOUTB,R))
           SELECT MEMBER=MM
/*
```

各制御ステートメントについての説明は以下のとおりです。

- OUTA DD では、3 つのメンバー (MA、MB、および MD) を含む区分データ・セット DATASETA を定義しています。
- INOUTB DD では、2 つのメンバー (MA および MJ) を含む区分データ・セット DATASET B を定義しています。
- INOUTC DD では、4 つのメンバー (MF、ML、MM、および MN) を含む区分データ・セット DATASETC を定義しています。
- INOUTD DD では、2 つのメンバー (MM および MP) を含む区分データ・セット DATASET D を定義しています。
- INE DD では、3 つのメンバー (MA、MJ、および MK) を含む区分データ・セット DATASETE を定義しています。
- SYSUT3 および SYSUT4 DD では、一時スピル・データ・セットを定義しています。ディスク・ボリューム上のそれぞれについて、1 つのトラックが割り振られます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには 3 つの COPY ステートメント、2 つの SELECT ステートメント、1 つの EXCLUDE ステートメント、およびいくつかの INDD ステートメントが含まれます。
- 最初の COPY ステートメントは、コピー操作の開始を指示します。OUTDD オペランドは、DATASETA を出力データ・セットとして指定します。

最初の INDD ステートメントは、DATASETE を最初に処理する入力データ・セットとして指定します。SELECT ステートメントは、メンバー MA および MJ を DATASETE から DATASETA にコピーするように指定します。処理は以下のように実行されます。

1. メンバー MA が検索されて見つかりますが、置換オプションが指定されていないためコピーされません。
2. メンバー MJ が検索されて見つかり、DATASET A にコピーされます。

2 番目の INDD ステートメントは最初のコピー・ステップの終了、および最初のコピー操作内の 2 番目のコピー・ステップの開始を示します。また、DATASET C を 2 番目に処理する入力データ・セットとして指定します。EXCLUDE ステートメントで指定されていないメンバー MF および ML は、そのどちらも DATASET A に含まれていないためコピーされます。EXCLUDE は、メンバー MM および MN を DATASET C から DATASET A にコピーしないように指定します。

- 2 番目の COPY ステートメントは、別のコピー操作の開始を指示します。OUTDD パラメーターは、DATASET B を出力データ・セットとして指定します。INDD パラメーターは、DATASET D を最初に処理する入力データ・セットとして指定します。メンバー MP および MM は、DATASET B にコピーされます。

次の INDD ステートメントは、DATASET C を 2 番目に、そして DATASET B を 3 番目に処理する入力データ・セットとして指定します。メンバー MF、ML、MM、および MN は、DATASET C からコピーされます。DATASET B には既にメンバー MM が含まれていますが、置換オプションが指定されているのでメンバー MM はコピーされます。

DATASET B のディレクトリーのポインターは新規の (コピーされた) メンバー MM を指すように変更されます。したがって、置換されたメンバー MM によって占められていたスペースは、組み込まれた未使用のスペースとなります。その後、DATASET B は同じ場所で圧縮されて、組み込まれた未使用スペースを除去します。(DATASET B は、入力および出力データ・セットの両方として指定されています。)

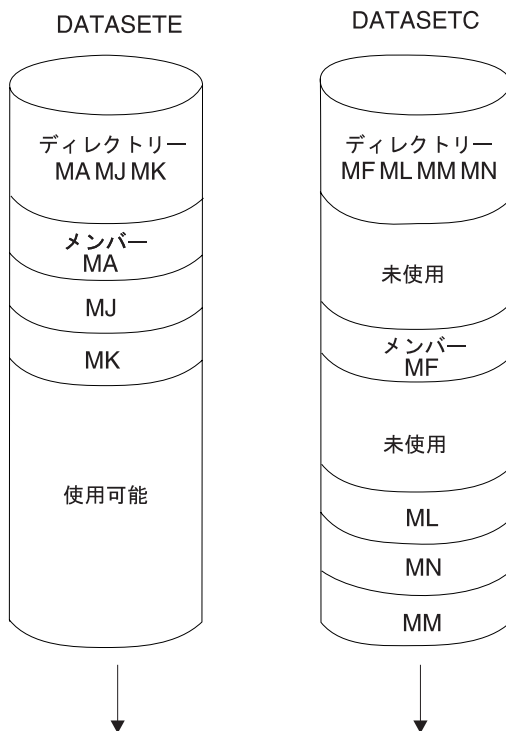
- 3 番目の COPY ステートメントは、最後のコピー操作の開始を指示します。OUTDD パラメーターは、DATASET D を出力データ・セットとして指定します。INDD パラメーターは、DATASET B を入力データ・セットとして指定します。

SELECT は、メンバー MM を DATASET B から DATASET D にコピーするように指定します。置換オプションがデータ・セット・レベルで指定されているので、メンバー MM はコピーされて、DATASET D のメンバー MM を置換します。

IEBCOPY

1 回目のコピー

入力



出力

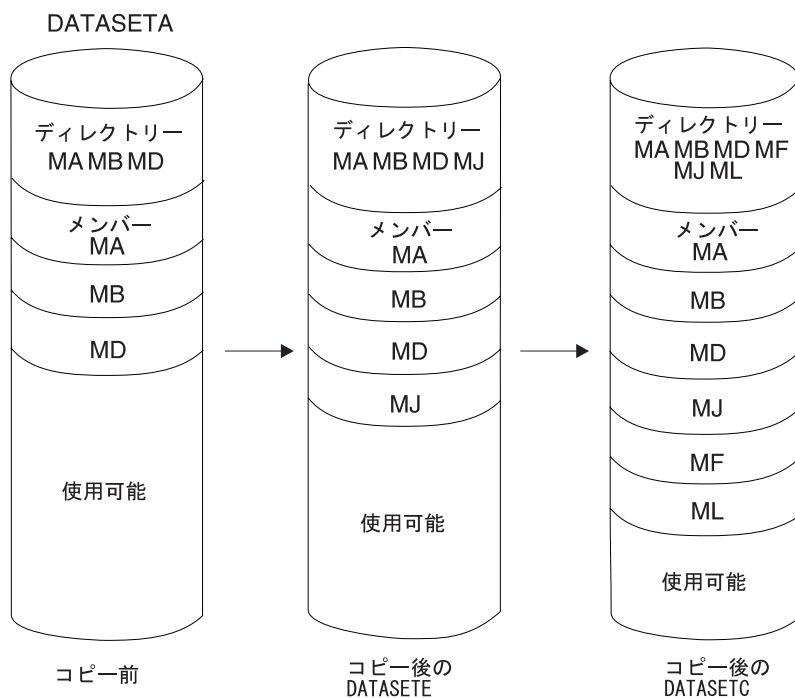
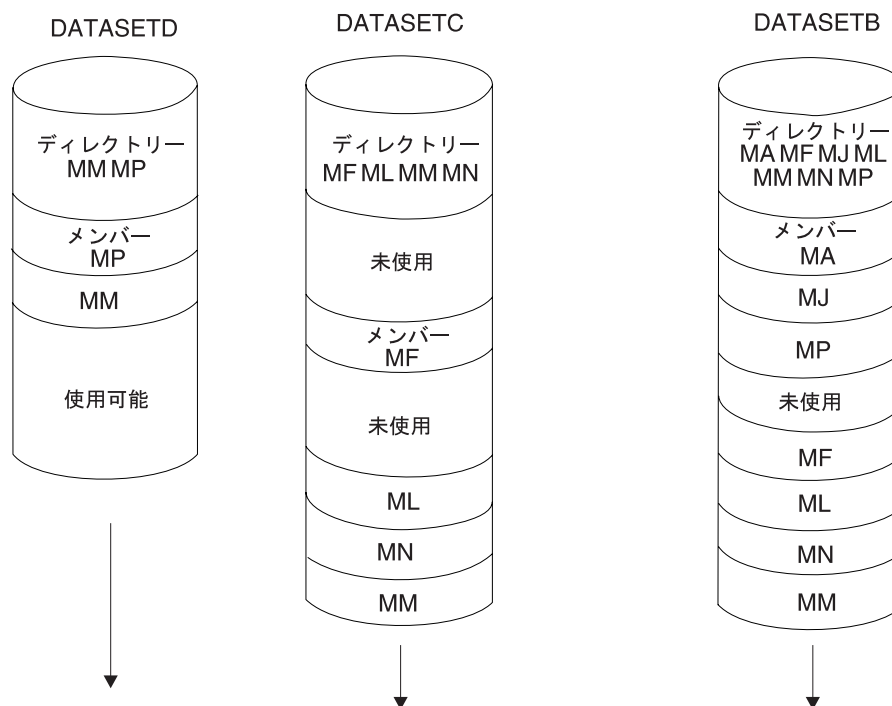


図 8. 複数のコピー操作 / ジョブ・ステップ内のコピー・ステップ (1/3)

2 回目のコピー

入力



出力

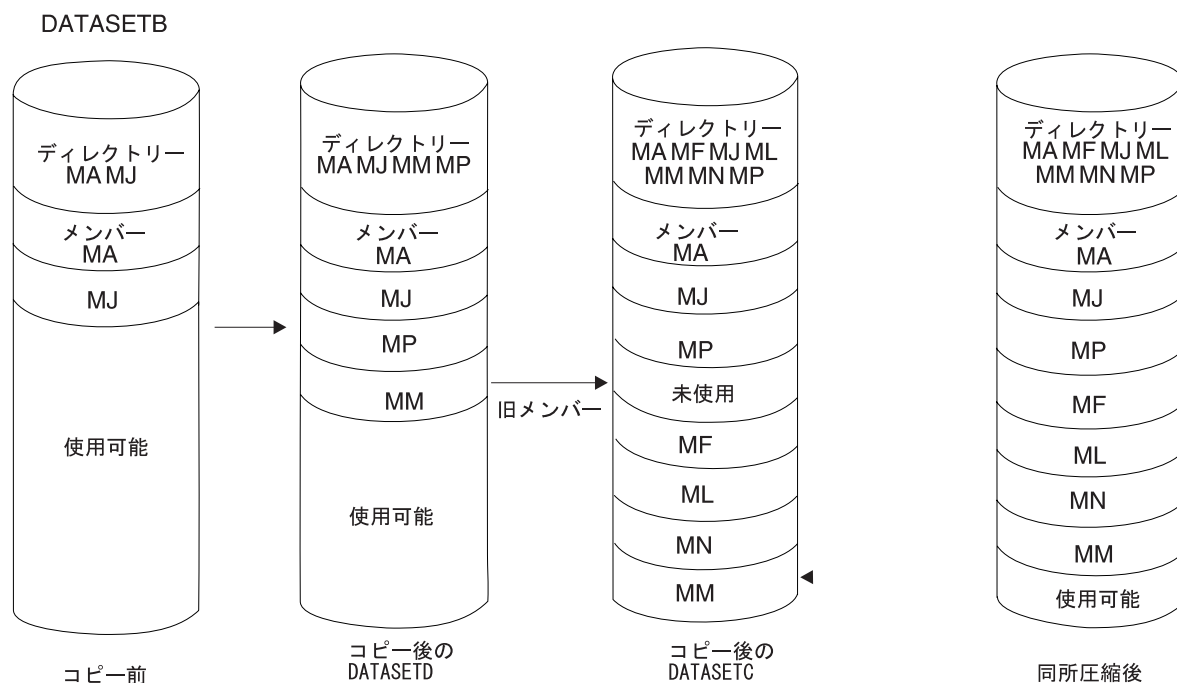
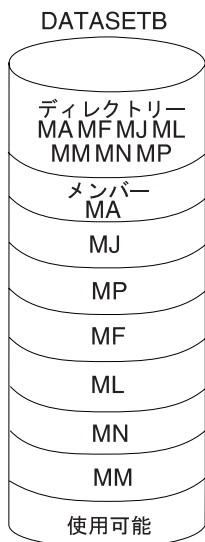


図 8. 複数のコピー操作 / ジョブ・ステップ内のコピー・ステップ (2/3)

IEBCOPY

3 回目のコピー

入力



出力

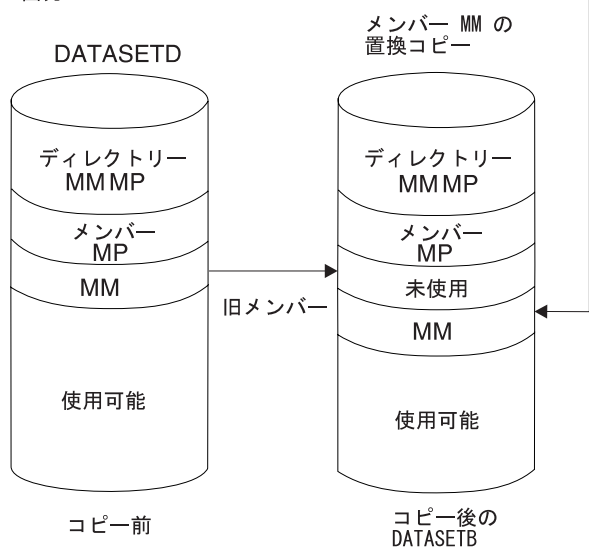


図 8. 複数のコピー操作 / ジョブ・ステップ内のコピー・ステップ (3/3)

例 8: データ・セットをロードする

この例では、IEBCOPY アンロード操作によって作成された順次データ・セットをロードします。

```
//LOAD    JOB    ...
//STEPA   EXEC  PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//SYSUT1  DD   DSN=UNLOADSET,UNIT=tape,LABEL=(,SL),
//         VOL=SER=TAPE01,DISP=OLD
//SYSUT2  DD   DSN=DATASET4,UNIT=disk,VOL=SER=2222222,
//         DISP=(NEW,KEEP),SPACE=(CYL,(10,5,10))
//SYSUT3  DD   DSN=TEMP1,UNIT=disk,VOL=SER=111111,
//         DISP=(NEW,DELETE),SPACE=(80,(15,1))
//SYSIN   DD   DUMMY
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、IEBCOPY アンロード操作によって以前に作成された順次データ・セットを定義しています。データ・セットには、28 のメンバーが順次編成で含まれています。
- **SYSUT2 DD** では、ディスク・ボリューム上に新規の区分データ・セットを定義しています。このデータ・セットはロード操作の後も保持されます。10 個のシリンダーがデータ・セットに割り振られます。10 個のブロックがディレクトリ一項目に割り振られます。
- **SYSUT3 DD** では、ディスク・ボリューム上に一時スピル・データ・セットを定義しています。
- **SYSIN DD** では、制御データ・セットを定義しています。SYSIN がダミーとされており、SYSUT1 が順次データ・セットを定義し、SYSUT2 が区分データ・セットを定義するため、SYSUT1 データ・セットの全体が SYSUT2 データ・セットにロードされます。

例 9: 選択したメンバーをアンロード、ロード、コピー、およびマージする

この例では、メンバーを選択、除外、アンロード、ロード、およびコピーします。処理は以下のように実行されます。

1. メンバーを除外して、アンロードする
2. メンバーを選択して、アンロードする
3. メンバーをマージするためにロードおよびコピーする

```
//COPY      JOB    ...
//STEP      EXEC  PGM=IEBCOPY
//SYSPRINT  DD    SYSOUT=A
//PDS1      DD    DSN=ACCOUNTA,UNIT=disk,VOL=SER=333333,
//              DISP=OLD
//PDS2      DD    DSN=ACCOUNTB,UNIT=disk,VOL=SER=333333,
//              DISP=OLD
//SEQ1      DD    DSN=SAVAC,UNIT=disk,VOL=SER=333333,
//              DISP=(NEW,KEEP),SPACE=(CYL,(5,2))
//SEQ2      DD    DSN=SAVACB,UNIT=tape,VOL=SER=T01911,
//              DISP=(NEW,KEEP),LABEL=(,SL)
//NEWUP     DD    DSN=NEWACC,UNIT=tape,VOL=SER=T01219,
//              DISP=OLD,LABEL=(,SL)
//MERGE     DD    DSN=ACCPDAT,UNIT=disk,VOL=SER=222222,
//              DISP=OLD
//SYSUT3    DD    DSN=TEMP1,VOL=SER=666666,UNIT=disk,
//              DISP=(NEW,DELETE),SPACE=(80,(1,1))
//SYSUT4    DD    DSN=TEMP2,VOL=SER=666666,UNIT=disk,
//              DISP=(NEW,DELETE),SPACE=(256,(1,1)),DCB=(KEYLEN=8)
//SYSIN     DD    *
              COPY  OUTDD=SEQ1,INDD=PDS1
              EXCLUDE MEMBER=(D,C)
              COPY  OUTDD=SEQ2,INDD=PDS2
              SELECT MEMBER=(A,K)
              COPY  OUTDD=MERGE,INDD=((NEWUP,R),PDS1,PDS2)
              EXCLUDE MEMBER=A
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **PDS1 DD** では、6 つのメンバー (A、B、C、D、E、および F) を含む ACCOUNTA と呼ばれる区分データ・セットを定義しています。

- PDS2 DD では、3 つのメンバー (A、K、および L) を含む ACCOUNTB と呼ばれる区分データ・セットを定義しています。
- SEQ1 DD では、SAVAC と呼ばれる新規の順次データ・セットを定義しています。
- SEQ2 DD では、SAVACB と呼ばれる新規の順次データ・セットをテープ・ボリューム上に定義しています。テープには IBM 標準ラベルがあります。
- NEWUP DD では、8 つのメンバー (A、B、C、D、M、N、O、および P) を含む区分データ・セットのアンロードされたフォームである、NEWACC と呼ばれる古い順次データ・セットを定義しています。これはテープ・ボリューム上に存在します。
- MERGE DD では、6 つのメンバー (A、B、C、D、Q、および R) を含む ACCUPDAT と呼ばれる区分データ・セットを定義しています。
- SYSUT3 および SYSUT4 DD では、一時スピル・データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- 最初の COPY ステートメントは、最初のコピー操作の開始を指示します。OUTDD パラメーターは SAVAC が出力データ・セットであることを定義し、INDD パラメーターは ACCOUNTA が入力データ・セットであることを定義します。SAVAC が順次データ・セットであるため、ACCOUNTA はこのコピー操作でアンロードされます。

EXCLUDE ステートメントは、メンバー D および C を ACCOUNTA のその他のメンバーと共に SAVAC にアンロードしないように指定します。

- 2 番目の COPY ステートメントは、2 番目のコピー操作の開始を指示します。OUTDD パラメーターは SAVACB が出力データ・セットであることを定義し、INDD パラメーターは ACCOUNTB が入力データ・セットであることを定義します。SAVACB が順次データ・セットであるため、ACCOUNTB はこのコピー操作でアンロードされます。

SELECT ステートメントは、ACCOUNTB のメンバーのうちメンバー A および K だけを SAVACB にアンロードするように指定します。

- 3 番目の COPY ステートメントは、最後のコピー操作の開始を指示します。OUTDD パラメーターは、ACCUPDAT を出力データ・セットとして指定します。EXCLUDE ステートメントは、メンバー A がこのコピー操作から除外されるように指定します。INDD パラメーターに指定された 3 つのデータ・セットは、以下のように処理されます。
 1. データ・セット NEWACC は順次データ・セットなので、ACCUPDAT にロードされます。置換オプションが指定されているので、NEWACC 内のメンバー B、C、および D は ACCUPDAT 内の同名のメンバーを置換します。NEWACC の残りのメンバーも ACCUPDAT にコピーされますが、A はコピー操作から除外されます。
 2. データ・セット ACCOUNTA は区分データ・セットなので、そのメンバーは ACCUPDAT にコピーされます。置換が指定されていないので、メンバー E および F だけがコピーされます。
 3. データ・セット ACCOUNTB は区分データ・セットなので、そのメンバーは ACCUPDAT にコピーされます。メンバー K および L だけがコピーされます。

例 10: ロード・モジュールを同じ場所を変更する

この例では、データ・セット MODLIBJ のすべてのメンバー、データ・セット MODLIBK のメンバー MODX、MODY、および MODZ、および MYMACRO と MYJCL を除くデータ・セット MODLIBL のすべてのメンバーを同じ場所を変更します。

```
//ALTERONE JOB ...
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
//LIBJ DD DSNAME=MODLIBJ,DISP=(OLD,KEEP)
//LIBK DD DSNAME=MODLIBK,DISP=(OLD,KEEP)
//LIBL DD DSNAME=MODLIBL,DISP=(OLD,KEEP)
//SYSIN DD *
        ALTERMOD OUTDD=LIBJ
        ALTERMOD OUTDD=LIBK,LIST=NO
        SELECT MEMBER=(MODX,MODY,MODZ)
        ALTERMOD OUTDD=LIBL
        EXCLUDE MEMBER=(MYMACRO,MYJCL)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- LIBJ DD では、事前に作成およびカタログされた区分データ・セット MODLIBJ を定義しています。
- LIBK DD では、事前に作成およびカタログされた区分データ・セット MODLIBK を定義しています。
- LIBL DD では、事前に作成およびカタログされた区分データ・セット MODLIBL を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- 最初の ALTERMOD ステートメントは、LIBJ で定義されたデータ・セット全体を同じ場所を変更するように指定します。
- 2 番目の ALTERMOD ステートメントおよびそれに続く SELECT ステートメントは、メンバー MODX、MODY、および MODZ を同じ場所での変更するように指示します。MODLIBK の残りは変更されません。
- 3 番目の ALTERMOD ステートメントおよびそれに続く EXCLUDE ステートメントは、MYMACRO および MYJCL と呼ばれるメンバーを除く MODLIBL のすべてを同じ場所を変更するように指示します。これらのメンバーは変更されないままとなります。

例 11: COPYMOD を使用してロード・モジュールを置換する

この例では、既存のロード・ライブラリーにあるロード・モジュールを他のモジュールによって置換します。新規のモジュールは最初は 3390 DASD 装置に存在し、それがコピーされるロード・ライブラリーは 3380 に存在します。モジュールのブロック・サイズは、出力データ・セットに割り当てられたブロック・サイズよりも大きいので、ロード・ライブラリーに追加する前にそのモジュールを再ブロックしなければなりません。

この例は、サイズの異なる複数の装置間でロード・モジュールを転送する方法を例示しています。この場合、更新されたモジュールは 3390 上で作成されて、一般的な使用のためにロード・ライブラリーに追加される前に検査されます。

```

//STEP1 EXEC PGM=IEBCOPY
//REPLACE JOB ...
//SYSPRINT DD SYSOUT=A
//TESTLIB DD DSN=JOHNDOE.COBOL.TESTLOAD,DISP=SHR,UNIT=3390,
//          VOL=SER=TEST01,DCB=(BLKSIZE=23470)
//PRODLIB DD DSN=PAYROLL.MASTER.LOADLIB,DISP=(OLD,KEEP)
//          UNIT=3380,VOL=SER=PROD01,DCB=(BLKSIZE=19069)
//SYSIN DD *
          COPYMOD OUTDD=PRODLIB,INDD=TESTLIB
          SELECT MEMBER=((WAGETAX,,R))
/*

```

各制御ステートメントについての説明は以下のとおりです。

- TESTLIB DD では、3390 直接アクセス装置上のロード・ライブラリーを定義しています。そのブロック・サイズは 23470 です。
- PRODLIB DD では、3380 直接アクセス装置上のロード・ライブラリーを定義しています。そのブロック・サイズは 19069 です。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- COPYMOD ステートメントは、PAYROLL.MASTER.LOADLIB を出力データ・セットとして、JOHNDOE.COBOL.TESTLOAD を入力データ・セットとして定義します。SELECT ステートメントは、ロード・モジュール WAGETAX を入力データ・セットからコピーして、出力データ・セットにある同名のメンバーを置換するように指示します。メンバーも 19069 に再ブロックされます。

この場合、メンバー WAGETAX を PAYROLL.MASTER.LOADLIB にコピーするために、COPYMOD を使用しなければならないことに注意してください。

WAGETAX の元のブロック・サイズが、出力データ・セットに常駐できるブロック・サイズの最大値よりも大きいため、COPY ステートメントを指定してこの操作を実行しようとしても成功しません。ブロック・サイズが不正であるとして、その問題には DCB 違反エラーの属性が付けられます。

例 12: ロード・ライブラリーを再ブロックして、複数の異なる装置タイプに配布する

この例では、元のロード・ライブラリーが存在した装置とは最大ブロック・サイズが異なる装置にロード・ライブラリーを (コピーによって) 配布します。ライブラリーはまず、そのライブラリーがコピーされる各装置と互換性のある最大ブロック・サイズに再ブロックされます。その後、そのライブラリーはそれらの装置にコピーされます。

この例は、ロード・ライブラリーを 1 つのタイプの直接アクセス装置上で作成してから、別のタイプの直接アクセス装置に配布する方法を示しています。

```

//RBLKCOPY JOB ...
//REBLOCK EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//TESTED DD DSN=TESTED.MASTER.LOADLIB,DISP=SHR
//STDSIZE DD DSN=PROGRAM.MASTER.LOADLIB,DISP=(OLD,KEEP),
//          UNIT=3390,VOL=SER=PROG01,DCB=(BLKSIZE=23470)
//SYSIN DD *
          COPYMOD OUTDD=STDSIZE,INDD=TESTED,MAXBLK=13030
/*
//DISTRIB EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//STDSIZE DD DSN=PROGRAM.MASTER.LOADLIB,DISP=SHR
//LIB3350 DD DSN=PROGRAM.LIB3380.LOADLIB,DISP=(OLD,KEEP),

```

```
//          UNIT=3380,VOL=SER=PACK01,DCB=(BLKSIZE=19069)
//LIB3330 DD  DSN=PROGRAM.LIB3380.LOADLIB,DISP=(OLD,KEEP),
//          UNIT=3380,VOL=SER=PACK02,DCB=(BLKSIZE=13030)
//SYSIN    DD  *
           COPY OUTDD=LIB3380,INDD=STDSIZE
           COPY OUTDD=LIB3380,INDD=STDSIZE
/*
```

各制御ステートメントについての説明は以下のとおりです。

- REBLOCK EXEC ステートメントは再ブロックのステップを開始します。
- TESTED DD では、カタログ化されたロード・ライブラリー TESTED.MASTER.LOADLIB を定義しています。
- STDSIZE DD では、3390 直接アクセス装置に存在してブロック・サイズが 23470 である既存のデータ・セット、PROGRAM.MASTER.LOADLIB を定義しています。
- SYSIN データ・セット内の COPYMOD ステートメントは、TESTED.MASTER.LOADLIB を PROGRAM.MASTER.LOADLIB にマージするように指定します。さらに、PROGRAM.MASTER.LOADLIB を最大ブロック・サイズ 13030 で再ブロックするようにも指定します。このブロック・サイズが選択されたのは、3380 および 3390 直接アクセス装置の両方に入れることのできるサイズだからです。
- DISTRIB EXEC ステートメントは配布ステップを開始します。このステップで再ブロックされたデータ・セットは、最大ブロック・サイズの異なる複数の装置にコピーされます。
- STDSIZE DD では、直前のステップで再ブロックされたものと同じデータ・セットを定義しています。
- LIB3380 DD では、3380 直接アクセス装置に存在するデータ・セット PROGRAM.LIB3380.LOADLIB を定義しています。
- SYSIN データ・セット内の COPY ステートメントは、データ・セット PROGRAM.MASTER.LOADLIB を再ブロックしないで出力データ・セットにコピーするように指定します。PROGRAM.MASTER.LOADLIB がより小さなブロック・サイズに再ブロックしていない場合、このステップは成功しないで終了します。

例 13: 区分データ・セットを PDSE に変換する

この例では、区分データ・セットを PDSE に変換します。

```
//CONVERT  JOB  ...
//STEP1   EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//SYSUT1  DD  DSNAME=PDSSET,DISP=SHR,DSNTYPE=PDS
//SYSUT2  DD  DSNAME=PDSESET,LIKE=PDSSET,DSNTYPE=LIBRARY,
//          DISP=(NEW,CATLG),STORCLAS=SCLASX,DATACLAS=DCLASY
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、区分データ・セットである入力データ・セット PDS を定義しています。これは既存のデータ・セットなので、DSNTYPE キーワードには影響されません。

- SYSUT2 DD では、拡張区分データ・セットである出力データ・セット PDSE を定義しています。この新規データ・セットには、ストレージ・クラスがあるので、SMS によって管理されます。

LIKE パラメーターは、PDSESET の DCB および SPACE 属性を PDSESET からコピーするように指定します。DSNTYPE パラメーターは、新規のデータ・セットを区分データ・セットとしてではなく PDSE として定義します。

DATACLAS=DCLASY は、PPDSE を不定形式の論理レコード長を持つプログラム・オブジェクト PDSE として示します。

ストレージ管理サブシステムは、SCLASX の定義に応じて、割り振りに使用する適切なボリュームを選択します。

- DD 名『SYSUT1』および『SYSUT2』が入力および出力データ・セットの定義に使用されるので、SYSIN データ・セットは必要ありません。

例 14: PDSE から PDSE にグループをコピーする

この例では、メンバーおよび別名 (グループ) を PDSE から PDSE にコピーします (データ・セットのフル・コピー)。グループをコピーすることについての情報は、32 ページの『プログラム・オブジェクトをコピーする (COPYGRP ステートメント)』を参照してください。

```
//CPYGRP   JOB   ...
//STEP1    EXEC PGM=IEBCOPY
//SYSPRINT DD   SYSOUT=A
//DDIN     DD   DSNAME=PDSESETA,DISP=SHR
//DDOUT    DD   DSNAME=PDSESETB,LIKE=PDSESETA,DSNTYPE=LIBRARY,
//          DISP=(NEW,CATLG)
//SYSUT3   DD   UNIT=SYSDA,SPACE=(TRK,(1,1))
//SYSIN    DD   *
GROUPPCPY COPYGRP INDD=DDIN,OUTDD=DDOUT
/*
```

各制御ステートメントについての説明は以下のとおりです。

- DDIN DD では、拡張区分データ・セットである入力データ・セット PDSESETA を定義しています。

DDOUT DD では、拡張区分データ・セットである出力データ・セット PDSESETA を定義しています。

LIKE サブパラメーターは、PDSESETB の DCB および SPACE 属性を PDSESETB からコピーするように指定します。DSNTYPE サブパラメーターは、新規のデータ・セットを PDSE として定義します。

ストレージ管理サブシステムは、割り振りに使用する適切なボリュームを選択します。

- SYSUT3 DD では、一時スピル・データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、COPYGRP ステートメント、INDD ステートメント、および OUTDD ステートメントが含まれます。
- COPYGRP はコピー操作の開始を指示します。

INDD パラメーターは、PDSESETA を入力データ・セットとして示します。

OUTDD パラメーターは、PDSESETB を出力データ・セットとして指定します。

例 15: グループを PDSE から PDSE に置換付きでコピーする

この例では、メンバーおよび別名をグループごとに PDSE から PDSE に置換 (R) オプション付きでコピーします。COPYGRP を使用してグループを置換することについての情報は、32 ページの『プログラム・オブジェクトを置換する』を参照してください。

```
//CPYGRP   JOB   ...
//STEP1    EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=A
//DDIN     DD  DSNAME=PDSESETA,DISP=SHR
//DDOUT    DD  DSNAME=PDSESETB,LIKE=PDSESETA,DSNTYPE=LIBRARY,
//         DD  DISP=(NEW,CATLG)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(TRK,(1,1))
//SYSIN    DD  *
GROUPCPY   COPYGRP INDD=((DDIN,R)),OUTDD=DDOUT
/*
```

各制御ステートメントについての説明は以下のとおりです。

- DDIN DD では、拡張区分データ・セットである入力データ・セット PDSE を定義しています。

DDOUT DD では、拡張区分データ・セットである出力データ・セット PDSE を定義しています。

LIKE パラメーターは、PDSESETB の DCB および SPACE 属性を PDSESETB からコピーするように指定します。

DSNTYPE パラメーターは、新規のデータ・セットを PDSE として定義します。

ストレージ管理サブシステムは、割り振りに使用する適切なボリュームを選択します。

- SYSUT3 DD では、一時スピル・データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、COPYGRP ステートメント、INDD ステートメント、および OUTDD ステートメントが含まれます。
- COPYGRP はコピー操作の開始を指示します。

((INDD,R)) パラメーターは、PDSESETA を、PDSESETB 内の同名のメンバーを置換するメンバーを含む入力データ・セットとして示します。

OUTDD パラメーターは、PDSESETB を出力データ・セットとして指定します。

例 16: 選択したグループを PDSE から PDSE にコピーする

この例では、選択したメンバーおよび別名を PDSE から PDSE にコピーします。メンバーの名前または最大 8 つの文字を SELECT ステートメントに指定できます。COPYGRP を使用してグループを選択することについての情報は、32 ページの『プログラム・オブジェクトをコピーする (COPYGRP ステートメント)』を参照してください。

IEBCOPY

```
//CPYGRP   JOB   ...
//STEP1    EXEC  PGM=IEBCOPY
//SYSPRINT DD   SYSOUT=A
//DDIN     DD   DSNAME=PDSESETA,DISP=SHR
//DDOUT    DD   DSNAME=PDSESETB,LIKE=PDSESETA,DSNTYPE=LIBRARY,
//         DD   DISP=(NEW,CATLG)
//SYSUT3   DD   UNIT=SYSDA,SPACE=(TRK,(1,1))
//SYSIN    DD   *
GROUPCPY   COPYGRP INDD=DDIN,OUTDD=DDOUT
           SELECT  MEMBER=(ALIAS001)

/*
```

各制御ステートメントについての説明は以下のとおりです。

- **DDIN DD** では、拡張区分データ・セットである入力データ・セット **PDSE** を定義しています。

DDOUT DD では、拡張区分データ・セットである出力データ・セット **PDSE** を定義しています。

すべての **PDSE** は、ストレージ管理サブシステムによって管理されなければなりません。

LIKE パラメーターは、**PDSESETB** の **DCB** および **SPACE** 属性を **PDSESETB** からコピーするように指定します。

DSNTYPE パラメーターは、新規のデータ・セットを **PDSE** として定義します。

ストレージ管理サブシステムは、割り振りに使用する適切なボリュームを選択します。

- **SYSUT3 DD** では、一時スピル・データ・セットを定義しています。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。データ・セットには、**COPYGRP** ステートメント、**INDD** ステートメント、および **OUTDD** ステートメントが含まれます。
- **COPYGRP** はコピー操作の開始を指示します。

INDD パラメーターは、**PDSESETA** を入力データ・セットとして示します。

OUTDD パラメーターは、**PDSESETB** を出力データ・セットとして指定します。

- **SELECT** ステートメントは、別名 **ALIAS001** を持つグループを入力データ・セット (**PDSESETA**) から選択して、出力データ・セット (**PDSESETB**) にコピーするように指示します。

第 4 章 IEBDG (テスト・データ・ジェネレーター) プログラム

IEBDG は、プログラミング・デバッグの補助として使用するテスト・データのパターンを提供するために使用する、データ・セット・ユーティリティーです。その後、このデータのパターンはすぐに分析されて、予想可能な結果が示されます。

入力データなしでデータ・セットを作成することも、入力データを取るデータ・セットを指定することもできます。作成するデータ・セットには、任意の形式のレコードを使用できます。順次データ・セット、または区分データ・セットまたは PDSE のメンバーを、入力または出力として使用できます。

さらに IEBDG には、各出力レコードを書き出す前に表示するための独自の出口ルーチンを指定するオプションも備わっています。

フィールドの内容を定義するとき、以下の事柄を決めなければなりません。

- どのパターンを定義済みフィールドに最初に入れるか。独自のパターンを使用することも、または IBM が提供するパターンを使用することもできます。
- フィールドの内容が出力レコードごとに選択された後、それを変更するためにどのようなアクションが実行されるか。

IEBDG をアプリケーション・プログラムから起動した場合、IEBDG を呼び出す前に SVC 99 を出すことによりデータ・セットを動的に割り振ることができます。

パターンの選択

ここに記載した IBM 提供のパターンのうちの 1 つを使用することも、ユーザーの所有するパターンを指定することもできます。

IBM 提供のパターン

IBM は以下の 7 つのパターンを提供します。

- 英数字
- 英字
- ゾーン 10 進数
- パック 10 進数
- 2 進数
- 照合シーケンス
- 乱数

フィールドの内容を定義するとき、1 つのパターンを選択することができます。パック 10 進数、2 進数、ゾーン 10 進数、およびランダム数のパターンを除くすべてのパターンは、定義されたフィールド長に繰り返しを許可するのに十分な長さがあれば、所定のフィールド内で繰り返されます。たとえば、英字パターンは以下のとおりです。

```
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF...
```


表 13 は、IBM 提供のパターンを示しています。

表 13. IBM 提供のテスト・データ・パターン

タイプ	16 進での表記	印刷可能文字での表記
英数字	C1C2...E9、F0...F9	AB...Z、0...9
英字	C1C2...E9	AB...Z
ゾーン 10 進	F0F0...F9F9	00...99
パック 10 進	0000...001C (正のパターン) 0000...001D (負のパターン)	該当なし
2 進数	00000001 (正のパターン) FFFFFFFF (負のパターン)	該当なし
照合 シーケンス	40...F9	b@.<(+#!\$*);-/,%_>?:'="A...Z、0...9
乱数	ランダム 16 進数字	該当なし

4 バイトのパック 10 進数または 2 進数、ゾーン 10 進数、またはランダム数は定義されたフィールドで右寄せされます。このフィールドの残りの部分には、ランダム数には充てん文字、パック 10 進数と 2 進数にはゼロ (X'00')、10 進数パターンには、ゾーン付きゼロ (X'F0') が入ります。

英数字、英字、または照合シーケンス・フィールドを定義するとき、開始文字を指定できます。たとえば、開始文字として『H』を指定した 10 バイトの英字フィールドは以下のようになります。

HIJKLMNOPQ

同じ 10 バイトの英字フィールドで開始文字を指定しないと、以下のようになります。

ABCDEFGHIJ

パック 10 進または 2 進フィールドを指定するとき、数学符号を指定できます。符号が指定されていない場合、そのフィールドは正であると想定されます。

ユーザー指定のパターン

IBM 提供のパターンを選択する代わりに、定義済みフィールドに入れる独自のパターンを指定したい場合があります。以下のものを指定できます。

- 文字ストリング
- IEBDG によってパック 10 進に変換される 10 進数
- IEBDG によって 2 進に変換される 10 進数

パターンを指定するとき、指定されたフィールド長と同じかより短いパターン長を指定しなければなりません。文字パターンは定義済みフィールドで左寄せされます。パック 10 進または 2 進に変換された 10 進数は定義済みフィールドで右寄せされます。

定義済みフィールドには最初に、文字または 16 進数字を入れることができます。たとえば、10 バイトのパターン『BADCFEHGJI』は 15 バイトのフィールドに入れます。文字『2』がフィールドの埋め込みに使用されます。結果は BADCFEHGJI22222 となります。(充てん文字が提供されない場合、残りのバイトには 2 進ゼロが入ります。) 充てん文字を指定した場合、それは定義済みフィールド内のパターンより前の部分にある各バイトに書き込まれます。

レコード内のフィールドを変更する

指定された方法により、IEBDG を使用してフィールドの内容を変更できます。フィールドを該当する各出力レコードに含めた後に変更するため、以下のアクションの 1 つを選択できます。

- リップル
- 左シフト
- 右シフト
- 左切り捨て
- 右切り捨て
- 固定
- ロール
- ウェーブ

86 ページの図 9 は、各アクションの効果を 6 バイトの英字フィールドで示しています。ロールおよびウェーブのアクションが適用されるのは、ユーザー作成のパターンを使用した場合だけであることに注意してください。さらに、リップル・アクションの結果はどちらのタイプのパターン (IBM 提供またはユーザー作成) を使用するかによって異なります。

IEBDG

ユーザー提供 リップル・パターン	IBM 提供 リップル・パターン	左シフト	右シフト
ABCDEF	ABCDEF	ABCDEF	ABCDEF
BCDEFA	BCDEFG	BCDEF	ABCDE
CDEFAB	CDEFGH	CDEF	ABCD
DEFABC	DEFGHI	DEF	ABC
EFGBCD	EFGHIJ	EF	AB
FABCDE	FGHIJK	F	A
ABCDEF	GHIJKL	ABCDEF	ABCDEF
BCDEFA	HIJKLM	BCDEF	ABCDE

左切り捨て	右切り捨て	固定	ユーザー提供 ロール・パターン	ユーザー提供 ウェーブ・パターン
ABCDEF	ABCDEF	ABCDEF	AAA	AAA
BCDEF	ABCDE	ABCDEF	AAA	AAA
CDEF	ABCD	ADCDEF	AAA	AAA
DEF	ABC	ABCDEF	AAA	AAA
EF	AB	ABCDEF	AAA	AAA
F	A	ABCDEF	AAA	AAA
ABCDEF	ABCDEF	ABCDEF	AAA	AAA
BCDEF	ABCDE	ABCDEF	AAA	AAA

図9. IEBDG のアクション

アクションを選択しない場合、または指定したアクションがフォーマットと互換性を持たない場合、IEBDG は**固定アクション**を想定します。

入出力

IEBDG は以下の入力を使用します。

- 出力データ・セット、または区分データ・セットか PDSE のメンバーの作成に使用されるレコードを含む入力データ・セット。入力データ・セットはオプションです。つまり、出力データ・セットはユーティリティー制御ステートメントによってすべて作成することが可能です。
- 任意のセット数のユーティリティー制御ステートメントを含む制御データ・セット。

IEBDG は以下の出力を生成します。

- IEBDG 操作の結果である出力データ・セット。出力データ・セットは、ジョブ・ステップに含まれるユーティリティー制御ステートメントの各ステップによって作成されます。
- 通知メッセージ、ユーティリティー制御ステートメントの内容、およびエラー・メッセージ (存在する場合) を含むメッセージ・データ・セット。

入力および出力データ・セットは、順次、または区分データ・セットか PDSE のメンバーとすることができます。BDAM および VSAM はサポートされていません。

戻りコードについては、付録 A を参照してください。

制御

IEBDG は、ジョブおよびユーティリティー制御ステートメントによって制御されます。ジョブ制御ステートメントは IEBDG を処理して、IEBDG によって使用および生成されるデータ・セットを処理します。ユーティリティー制御ステートメントは、プログラムの機能の制御、および出力レコードの内容の定義に使用されます。

ジョブ制御ステートメント

表 14 は、IEBDG で使用するジョブ制御ステートメントをまとめたものです。

入力および出力データ・セットはどちらも、固定長、可変長、および不定形式レコードを含むことができます。

表 14. IEBDG で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEBDG)。このジョブ制御ステートメントがプロシージャ・ライブラリー内にある場合はプロシージャ名を指定します。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。データ・セットは、システム出力装置、テープ・ボリューム、または DASD ボリューム上に書き込むことができます。
anyname1 DD	オプションの入力データ・セット、またはデータ・セットのメンバーを定義します。任意の数のこれらのステートメント (それぞれにジョブ・ステップ内の他のすべての DD 名と異なる DD 名がある) を、ジョブ・ステップ内に含めることができます。
anyname2 DD	出力データ・セットまたはデータ・セットのメンバーを定義します。任意の数のこれらの DD ステートメントをジョブ・ステップごとに含めることができます。しかし、ユーティリティー制御ステートメントの各セットについて適用可能なステートメントは 1 つだけです。
SYSIN DD	ユーティリティー制御ステートメント、およびオプションとして入力レコードを含む制御データ・セットを定義します。データ・セットは通常、入力ストリーム内に存在します。しかし、それを順次データ・セット、または区分データ・セットや PDSE のメンバーとして定義することも可能です。

EXEC ステートメント

EXEC ステートメントには、メッセージ・データ・セット内の見出しと見出しとの間に印刷する行数を指定する PARM パラメーターを、オプションで含めることができます。

表 15 は、EXEC ステートメントの構文規則を示しています。

表 15. EXEC ステートメントの構文

//[stepname]	EXEC	PGM=IEBDG[,PARM='LINECT=nnnn']
--------------	------	--------------------------------

各項目の意味は以下のとおりです。

PGM=IEBDG

IEBDG プログラムを実行することを指定します。

PARM='LINECT=nnnn'

出力リスト上のページごとに印刷する行数を指定します。その数値は、0000~9999 までの 4 桁の 10 進数です。

PARM を省略した場合、見出しと見出しとの間に 58 行が印刷されます。(例外として、チャンネル 12 穿孔が紙送り制御テープで検出された場合、チャンネル 1 へのスキップが実行されて見出しが印刷されます。)

SYSPRINT DD ステートメント

SYSPRINT DD ステートメントを省略した場合、メッセージは書き込まれません。SYSPRINT データ・セットのブロック・サイズは、121 の倍数でなければなりません。任意のブロック化因数を指定できます。

anyname1 DD ステートメント

任意の数のデータ・セットまたはメンバーを IEBDG への入力として使用できます。データ・セットまたはメンバーごとに、別個の DD ステートメントをコーディングしなければなりません。

たとえば、データ・セットに 1 つの DD ステートメントをコーディングするだけで区分データ・セットの 2 つのメンバーを使用することはできません。入力データ・セットが区分データ・セットまたは PDSE のメンバーである場合、そのメンバーを参照するように DD ステートメントをコーディングしなければなりません。これを行うには、データ・セット名パラメーターを `DSNAME=datasetname(membername)` としてコーディングします。

さらに、入力データ・セットとして順次データ・セットを指定することができます。このデータ・セットが順次の場合は、基本フォーマット、ラージ・フォーマット、拡張フォーマット、圧縮フォーマット、UNIX ファイル、TSO 端末またはテープ上のファイルであることができます。データ・セットの最大ブロック・サイズは 32760 バイトを超えてはなりません。

anyname2 DD ステートメント

IEBDG を 1 回使用するだけで、任意の数のデータ・セットを作成できます。作成しているデータ・セットまたはメンバーごとに、別個の DD ステートメントをコーディングしなければなりません。データ・セットまたはメンバーは新規のものでなければなりません。

区分データ・セットまたは PDSE の新しいメンバーを作成している場合、DD ステートメントはその特定のメンバーを参照する必要があります。これを行うには、データ・セット名パラメーターを `DSNAME=datasetname(membername)` としてコーディングします。

さらに、順次データ・セットを作成できます。このデータ・セットが順次の場合には、基本フォーマット、ラージ・フォーマット、拡張フォーマット、圧縮フォーマット、UNIX ファイル、TSO 端末またはテープ上のファイルであることができます。1 回の操作で入力と出力に同じ順次データ・セットを指定する場合、入力データの一部が読み取られる前にユーティリティーによって破壊される可能性があります。これにより、入出力エラーが生じることがあります。ユーティリティーはメンバーをデータ・セットの別の領域に書き込むため、それらは区分データ・セットの同じメンバーであることがあります。データ・セットの最大ブロック・サイズは 32760 バイトを超えてはなりません。

スペース割り振りの見積もりについては、*z/OS DFSMS* データ・セットの使用法を参照してください。

SYSIN DD ステートメント

SYSIN データ・セットのブロック・サイズは、80 の倍数でなければなりません。任意のブロック化因数を指定できますが、ブロック・サイズは 32760 バイトを超えてはなりません。

ユーティリティー制御ステートメント

IEBDG は表 16 のユーティリティー制御ステートメントによって制御されます。

表 16. IEBDG で使用するユーティリティー制御ステートメント

ステートメント	用途
DSD	入力および出力データ・セットの DD 名を指定します。ユーティリティー制御ステートメントの各セットに対して 1 つの DSD ステートメントを含めなければなりません。
FD	出力レコードの作成に使用するフィールドの内容および長さを定義します。
REPEAT	CREATE ステートメントまたは CREATE ステートメントのグループが出力レコードの生成に使用される回数を指定します。
CREATE	出力レコードの内容を定義します。
END	IEBDG ユーティリティー制御ステートメントのセットの終了を示します。

1 つのジョブ中に任意の数の制御ステートメントのセットを入れることができます。各セットは 1 つのデータ・セットを定義します。制御ステートメントのセットには、1 つの DSD ステートメント、任意の数の FD、CREATE、および REPEAT ステートメント、および FD ステートメントで INPUT が省略されているときは 1 つの END ステートメントが含まれます。

ユーティリティー制御ステートメントの継続のための一般要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

DSD ステートメント

DSD ステートメントはユーティリティー制御ステートメントのセットの開始を示し、IEBDG が入力として使用するデータ・セットを指定します。DSD ステートメントを使用して、IEBDG の適用ごとに 1 つの出力データ・セット、および任意の数の入力データ・セットを指定できます。

DSD ステートメントの構文は次のとおりです。

[label/]	DSD	OUTPUT=(ddname) [,INPUT=(ddname1[,ddname2][,...])]
----------	-----	-------------------------------------------------------

各項目の意味は以下のとおりです。

OUTPUT=(ddname)

出力データ・セットを定義している DD ステートメントの DD 名を指定します。

INPUT=(ddname1[,ddname2] [...])

プログラムへの入力として使用するデータ・セットを定義している DD ステートメントの DD 名を指定します。任意の数のデータ・セットを入力として含まれます。つまり、対応する DD ステートメントを参照する任意の数の DD 名をコーディングできます。DD 名を継続レコードに含める場合には常に、それが 4 桁目から開始するようにしなければなりません。

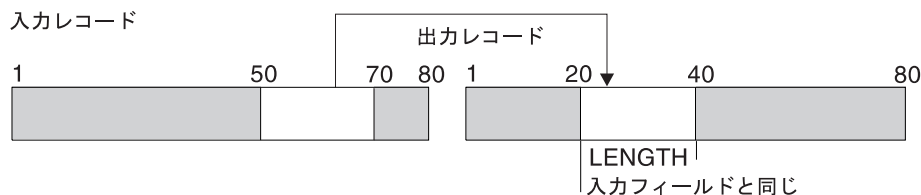
DD 名の SYSIN を DSD 制御ステートメント上の INPUT パラメーター内にコーディングしてはなりません。制御ステートメントごとに含めることのできる DD 名は 1 つまでです。

DSD ステートメントごとに含めることのできるパラメーターは 1 つまでです。

FD ステートメント

FD ステートメントは、出力レコードを形成するため CREATE ステートメント (1 つまたは複数) によって継続的に使用されるフィールドの内容および長さを定義します。入力論理レコード内に定義されたフィールドは、後続の CREATE ステートメントによって名前により参照されている場合、出力フィールドに使用するために選択できます。

図 10 は、FD ステートメントを使用して入力レコード内のフィールドを出力レコードに使用するように指定する方法を示しています。図の左側は、50 バイト目から開始する入力レコード内のフィールドを出力レコードに使用するように選択されたことを示しています。図の右側は、そのフィールドを出力レコードの 20 バイト目に配置することを示しています。



FD NAME=FIELD1,LENGTH=20,STARTLOC=20,FROMLOC=50,INPUT=INSET

図 10. 入力レコードからフィールドを選択して出力レコードで使用する

入力レコードからフィールドを選択する場合 (FD INPUT=ddname)、そのフィールドはユーティリティ制御ステートメントの各セット内で FD ステートメントにより定義されなければなりません。その場合、フィールド選択のために定義されたフィ

ールドはユーティリティー制御ステートメントのセットを超えて使用することはできません。そのような FD レコードを複製して、ジョブ・ステップ内の複数のユーティリティー制御ステートメント内で使用することは可能です。

さらに、数値フィールドが (1 つまたは複数の) CREATE ステートメントによって n 回参照された後に変更されるように、つまり n サイクルの後に変更が行われるように指示することができます。変更は、フィールドに指定した数値を追加します。

FD ステートメントの構文は次のとおりです。

[label]	FD	NAME=name ,LENGTH=length [STARTLOC=starting-location] [FILL={'character'IX'nn'}] [{FORMAT=pattern[CHARACTER=character] PICTURE=length,{character-string}'P'n'B}}] [SIGN=sign] [ACTION={FX RO RP SL SR TL TR WV}] [,INDEX=n[,CYCLEn[,RANGE=n]]] [,INPUT=ddname] [,FROMLOC=number]
---------	-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

NAME=name

この FD ステートメントで定義されるフィールドの名前を指定します。

LENGTH=length

定義されるフィールドの長さをバイト数で指定します。可変長レコードでは、フィールドの合計は最大レコード長よりも 4 バイト少なくしなければなりません。これは、レコード記述子ワードが 4 バイトであるためです。

INPUT パラメーターを指定すると、LENGTH パラメーターは各入力論理レコードの長さを指定変更します。FROMLOC および LENGTH の値の組み合わせが入力レコードよりも長い場合、結果は予測できません。

ACTION=RP または WV の場合、長さは 16383 バイトに制限されます。

ACTION=RO の場合、長さは 10922 バイトに制限されます。

STARTLOC=starting-location

フィールドを開始する開始位置 (このフィールドを使用するすべての出力レコード内での) を指定します。たとえば、出力レコードの最初のバイトを開始位置として選択する場合、このキーワードを STARTLOC=1 とコーディングします。10 番目のバイトを選択する場合、STARTLOC=10 とコーディングします。

デフォルト: フィールドは、出力レコード内の使用可能な最初のバイト (適用可能な CREATE ステートメントに指定したフィールド名の順序によって決まる) で開始します。可変長レコードでは、開始位置は長さの記述子より後の最初のバイトです。

FILL={'character'IX 'nn'}

フィールドを構成する他の操作より前に、出力レコード内のフィールドの各バイトに入れるべき値を指定します。コーディングできる値は以下のとおりです。

1

'character'

EBCDIC に変換して出力フィールドの各バイトに入れる文字を指定します。

X'*nn*'

出力フィールドの各バイトに入れる 2 桁の 16 進数字を指定します (たとえば、FILL=X'40'、または FILL=X'FF' など)。

デフォルト: 2 進ゼロが出力フィールドに入れられます。

FORMAT=pattern[,CHARACTER=character]

定義されたフィールドに入れる IBM 提供のパターンを指定します。 PICTURE を使用しているときは、FORMAT を使用することはできません。コーディングできる値は以下のとおりです。

パターン

以下の IBM 提供のパターンを指定します。

AL

英字パターンを指定します。

AN

英数字パターンを指定します。

BI

2 進パターンを指定します。

CO

照合シーケンス・パターンを指定します。

PD

パック 10 進パターンを指定します。

RA

ランダム 2 進パターンを指定します。

ZD

ゾーン 10 進パターンを指定します。

CHARACTER=character

フィールドの開始文字を指定します。開始文字の詳細については、83 ページの『IBM 提供のパターン』を参照してください。

PICTURE=length,{ 'character-string'*lP'n*'*lB'n*'}

ユーザー指定のパターンの長さおよび内容を指定します。FORMAT を使用しているときは、PICTURE を使用することはできません。PICTURE および NAME の両方を省略した場合、CREATE ステートメントで指定した充てん文字が適用される出力レコードの各バイトに入ります。コーディングできる値は以下のとおりです。

length

パターンが占めるバイト数を指定します。*length* は、FD ステートメントの LENGTH パラメーター値以下でなければなりません。

'character-string'

EBCDIC に変換して適当なレコードに入れる文字ストリングを指定します。文字ストリングは定義された開始位置で左寄せされます。72 桁目に非ブランクを入れて次のステートメントの 4 桁目からストリングを継続することにより、文字ストリングを 71 桁目で分割できます。引用符で囲まれた文字数は、*length* サブパラメーターで指定した数と同じでなければなりません。

P'n'

パック 10 進に変換して出力レコードまたは定義済みフィールド内で右寄せ (定義された長さおよび開始バイトの範囲で) される 10 進数を指定します。引用符で囲まれた文字数は、*length* サブパラメーターで指定した数と同じでなければなりません。

B'n'

2 進に変換して出力レコードまたは定義済みフィールド内で右寄せ (定義された長さおよび開始バイトの範囲で) される 10 進数を指定します。引用符で囲まれた文字数は、*length* サブパラメーターで指定した数と同じでなければなりません。

SIGN=sign

パック 10 進または 2 進フィールドを指定するときに使用する数学符号 (+ または -) を指定します。

デフォルト: 正 (+)。

ACTION={FX|RO|RP|SL|SR|TL|TR|WV}

フィールドを出力レコードに含めた後に、定義済みフィールドの内容を変更する場合の方法を指定します。

FORMAT および PICTURE 値と互換性のあるシステム処置については、95 ページの表 17 を参照してください。IEBDG ACTION の例については、86 ページの図 9 を参照してください。コーディングできる値は以下のとおりです。

FX

フィールドを出力レコードに含めた後に、定義済みフィールドの内容が固定されるように指定します。

FX がデフォルトです。

RO

フィールドを出力レコードに含めた後に、定義済みフィールドの内容がロールされるように指定します。パターン (『ピクチャー』) は、ピクチャーの最初の非ブランク文字がフィールドの最初のバイトに来るまで、出力レコードごとに 1 バイトずつ左に移動します。その後、フィールド内の元の位置に戻るまで、ピクチャーは出力レコードごとに 1 バイトずつ右に移動します。

RO を使用できるのは、ユーザー定義のパターンだけです。RO を有効にするには、ピクチャーの長さをフィールドの長さよりも短くしなければなりません。

RP

フィールドを出力レコードに含めた後に、定義済みフィールドの内容がリップルされるように指定します。

SL フィールドを出力レコードに含めた後に、定義済みフィールドの内容が左にシフトされるように指定します。

SR

フィールドを出力レコードに含めた後に、定義済みフィールドの内容が右にシフトされるように指定します。

TL

フィールドを出力レコードに含めた後に、定義済みフィールドの内容が左に切り捨てられるように指定します。

TR

フィールドを出力レコードに含めた後に、定義済みフィールドの内容が右に切り捨てられるように指定します。

WV

フィールドを出力レコードに含めた後に、定義済みフィールドの内容がウェーブされるように指定します。パターン (『ピクチャー』) は、ピクチャーの最初の非ブランク文字がフィールドの最初のバイトに来るまで、出力レコードごとに 1 バイトずつ左に移動します。その後、文字ストリングは元の位置にリセットされます。

WV を使用できるのは、ユーザー定義のパターンだけです。WV を有効にするには、ピクチャーの長さをフィールドの長さよりも短くしなければなりません。

INDEX=*n*[,CYCLE=*n*] [,RANGE=*n*]

指定された数のレコードが書き込まれたときにこのフィールドに追加される 10 進数を指定します。INDEX が有効なのは、FORMAT パターンが ZD、PD、および BI であるか、または PICTURE パターンが P'*n*' および *n* のときだけです。コーディングできるその他の値は以下のとおりです。

CYCLE=*n*

INDEX キーワードによってグループとして処理される出力レコード (出力として書き込むか、または出口ルーチンから使用可能にするもの) の数を指定します。このフィールドを使用してレコード数を指定する場合、それは INDEX パラメーターでの指定値に変更されます。たとえば、CYCLE=3 をコーディングした場合、出力レコードは 111 222 333 444 (以下同様) となります。このパラメーターをコーディングできるのは、INDEX をコーディングした場合だけです。

RANGE=*n*

このフィールドの内容が超過できない絶対値を指定します。インデックス操作が指定された絶対値を超過しようとした場合、直前のインデックス操作でのフィールドの内容が使用されます。

デフォルト: 索引付けは実行されません。CYCLE を省略して INDEX をコーディングする場合、CYCLE 値には 1 が想定されます。つまり、フィールドが潜在的な出力レコードに含められた後で、それぞれの索引付けが行われます。

INPUT=*ddname*

フィールド選択の入力として使用するデータ・セットを定義する DD ステートメントの DD 名を指定します。FD ステートメントに記述されたレコードの部分だけが出力レコードに入れられます。出力データ・セットのレコード・フォーマットが可変長レコードを示している場合、出力レコード内での位置は STARTLOC が指定されていないければ、出力レコードに対して最後に挿入された位置に依存します。

DD 名の SYSIN を FD 制御ステートメント上の INPUT パラメーター内にコーディングしてはなりません。制御ステートメントごとに含めることのできる DD 名は 1 つまでです。

入力レコードを読み取るために、対応する DD 名も関連した CREATE ステートメントに指定しなければなりません。

FROMLOC=*number*

入力論理レコード内の選択フィールドの位置を指定します。数値は入力レコード内での位置を表します。たとえば、FROMLOC=10 とコーディングした場合、指定フィールドは 10 番目のバイトから開始します。FROMLOC=1 とコーディングした場合、指定フィールドは最初のバイトから開始します。(可変長レコードでは、4 バイトの長さの記述子の後に有効なデータが開始します。)

固定長または固定長ブロック・レコード・フォーマットで RKP>0 のデータ・セットを検索するとき、レコードはキーおよびキーの組み込まれたデータから構成されます。レコード全体をコピーするには、出力論理レコード長を入力論理レコード長とキー長との合計にします。データ (組み込みキーを含む) だけをコピーする場合、FROMLOC をキー長と同じ値に設定します。

デフォルト: 入力レコードの開始。

FD の使用上の注意: 一部の FD キーワードは特定のパターンまたはピクチャーを選択したときに適用されません。たとえば、INDEX、CYCLE、RANGE、および SIGN パラメーターを使用できるのは数値フィールドだけです。表 17 は、IEBDG キーワードとパターンとの使用可能な組み合わせを示したものです。FD ステートメントごとに含めることのできるキーワードは 1 つまでです。

表 17. 互換性のある IEBDG 操作

FORMAT/PICTURE 値	互換性のあるパラメーター
FORMAT=AL (英字)	ACTION=SL (左にシフト)
FORMAT=AN (英数字)	ACTION=SR (右にシフト)
FORMAT=CO (照合シーケンス)	ACTION=TL (左で切り捨て)
	ACTION=TR (右で切り捨て)
	ACTION=FX (固定)
	ACTION=RP (リップル)
FORMAT=ZD (ゾーン 10 進)	INDEX= <i>n</i>
FORMAT=PD (パック 10 進)	CYCLE= <i>n</i>
FORMAT=BI (2 進)	RANGE= <i>n</i>
	SIGN= <i>n</i> ¹
	SIGN= <i>n</i> (1)
PICTURE=P' <i>n</i> ' (パック 10 進)	INDEX= <i>n</i>
PICTURE= <i>n</i>	CYCLE= <i>n</i>
(2 進)	RANGE= <i>n</i>
	SIGN= <i>n</i>

表 17. 互換性のある IEBDG 操作 (続き)

FORMAT/PICTURE 値	互換性のあるパラメーター
PICTURE='string' (EBCDIC)	ACTION=SL (左にシフト) ACTION=SR (右にシフト) ACTION=TL (左で切り捨て) ACTION=TR (右で切り捨て) ACTION=FX (固定) ACTION=RP (リップル) ACTION=WV (ウェーブ) ACTION=RO (ロール)

注: 1 ゾーン 10 進数 (ZD) には符号は含まれません。 (1) ゾーン 10 進数 (ZD) には符号は含まれません。

REPEAT ステートメント

REPEAT ステートメントは、CREATE ステートメントまたは CREATE ステートメントのグループが出力レコードの生成に繰り返し使用される回数を指定します。REPEAT ステートメントは、それが適用される CREATE ステートメントの**前に置かれます**。REPEAT ステートメントの構文は次のとおりです。

[label]	REPEAT	QUANTITY=number[,CREATE=number]
---------	--------	---------------------------------

各項目の意味は以下のとおりです。

QUANTITY=number

CREATE ステートメントの定義済みグループが繰り返し使用される回数を指定します。この数値は 65,535 を超えることができません。

CREATE=number

グループに含める以下の CREATE ステートメントの数を指定します。

デフォルト: 繰り返されるのは、最初の CREATE ステートメントだけです。

CREATE ステートメント

CREATE ステートメントは、出力レコードとして直接記述するレコード、または指定する出口ルーチンから使用可能にするレコードの、内容を定義します。出力レコードは、以前に定義されたフィールドを名前参照することにより、またはレコードに入れるパターン (『ピクチャー』) を指定することにより作成されます。単一の CREATE ステートメントを使用して複数のレコードを生成できます。

出力レコードは以下の順序で作成されます。

1. 指定したまたはデフォルト (2 進ゼロ) の充てん文字は、出力レコードの各バイトに最初にロードされます。
2. INPUT オペランドを FD ステートメント上ではなく CREATE ステートメント上に指定した場合、入力レコードは対応する出力レコードで左寄せされます。
3. INPUT オペランドが任意の FD ステートメントに *ddname* を指定している場合、FD ステートメントに記述されているフィールドだけが出力レコードに入れます。

4. FD フィールドが存在する場合、それらは CREATE ステートメントに名前が指定された順番で出力レコード内に置かれます。出力レコード内のフィールドの位置は、そのフィールドに指定された開始位置があるかどうかによって依存します (STARTLOC)。

たとえば、どのフィールドの開始位置をも指定しない場合、フィールドは出力レコード内に出力レコードの最初の位置から開始して順番に置かれます。図 11 は、フィールド X を 2 つの異なるレコードに追加する様子を示したものです。レコード 1 では、フィールド X は CREATE ステートメントによって参照される最初のフィールドなので、フィールド X は出力レコードの最初のバイトで開始します。レコード 2 では、フィールド A およびフィールド B の 2 つのフィールドは CREATE ステートメントによって既に参照されています。次に参照されるフィールド X は、フィールド B の直後から開始します。フィールド X は、この例では特別の開始位置はありません。

レコード 1: **CREATE NAME=X**

1	21	80
フィールド X		

レコード 2: **CREATE NAME=(A,B,X)**

1	21	41	61	80
フィールド A	フィールド B	フィールド X		

図 11. IEBDG を使用して出力レコード内にフィールドを配置するときのデフォルト

FD フィールドに明示的に指定された開始位置がある場合、各フィールドは出力レコード内の指定された開始位置に置かれます。これらのフィールドは、CREATE ステートメントに指定された順序で出力レコードに書き込まれます。したがって、あるフィールドに他のフィールドと同じ開始位置または他のフィールドと重なる開始位置がある場合、CREATE ステートメント上で後に指定されているフィールドがその位置を占めることになります。図 12 は、フィールドが重なる結果になるような開始位置の指定された 2 つのフィールドの例を示したものです。

FD NAME=FIELD1,LENGTH=30,STARTLOC=20,...
FD NAME=FIELD2,LENGTH=40,STARTLOC=40,...
CREATE NAME=(FIELD1,FIELD2)

1	20	40	50	80
	FIELD1	FIELD2		

図 12. 出力位置の指定されたフィールドの配置

一部のフィールドに開始位置が指定されていて、他のフィールドには指定されていない場合、フィールドの位置はそれらを CREATE ステートメントで指定した順序に依存します。開始位置を指定していないフィールドは、最後に書き込まれたレコードの直後から開始します。

たとえば、図 13 は開始位置の指定されたものと指定されていないものの 2 つのフィールドが出力レコードに書き込まれる方法を示しています。レコード 1 は、開始位置の指定された FIELD1 が最初に書き込まれることを示しています。FIELD1 は位置 20 から開始して、20 バイトを占めます。FIELD2 が次に書き込まれるので、位置 40 から開始します。レコード 2 は、開始位置の指定されていない FIELD2 が最初に書き込まれることを示しています。FIELD2 は出力レコード内の最初の位置から開始するように置かれます。その後 FIELD1 が、長さ 30 の FIELD2 と重なりますが位置 20 に置かれます。

FD NAME=FIELD1,LENGTH=20,STARTLOC=20,...
FD NAME=FIELD2,LENGTH=30,...

レコード 1: **CREATE NAME=(FIELD1,FIELD2)**

1	20	40	70	80
	FIELD1	FIELD2		

レコード 2: **CREATE NAME=(FIELD2,FIELD1)**

1	20	30	40	70	80
FIELD2	FIELD1				

図 13. 一部の出力位置だけが指定されたフィールドの配置

5. CREATE ステートメント・ピクチャーが存在する場合、それは出力レコード内に置かれます。

99 ページの図 14 は、出力レコードをユーティリティー制御ステートメントから作成する 3 つの方法を示しています。

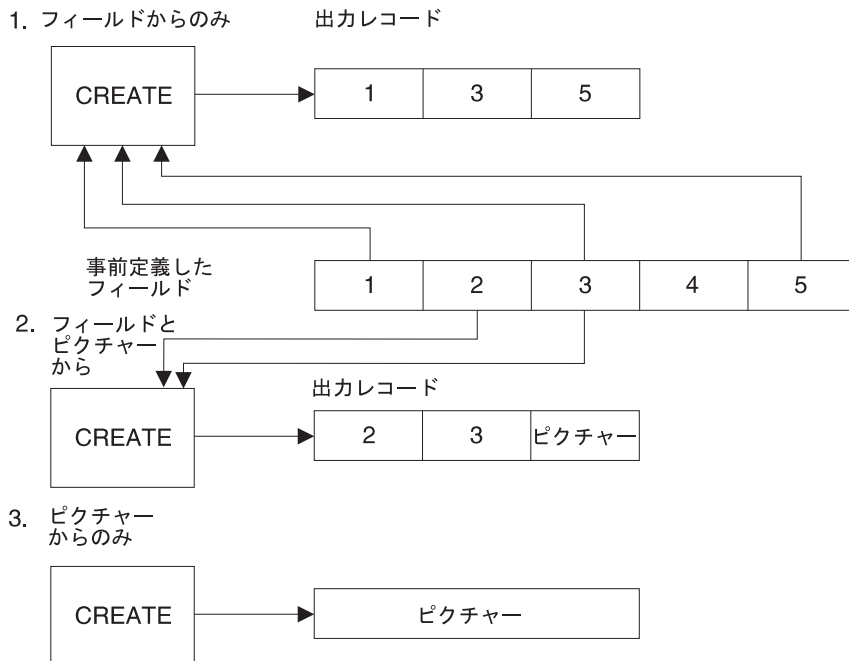


図 14. ユーティリティー制御ステートメントを使用して出力レコードを作成する

CREATE ステートメントでピクチャーを定義するとき、出力レコードでのその長さおよび開始位置を指定しなければなりません。指定する長さは、指定する英字または数字の数と同じでなければなりません。(指定する 10 進数がパック 10 進または 2 進に変換される場合、それは自動的に右寄せされます。)

出力レコードを作成するための入力レコードのソースとして別のデータ・セットを使用することも、または入力レコードを入力ストリームまたは SYSIN データ・セットを含めることもできます。個々の CREATE ステートメントで使用できる入力データ・セットは 1 つだけです。

CREATE ステートメントの構文は次のとおりです。

[label]	CREATE	[QUANTITY= <i>n</i>] [.FILL={ <i>character</i> ' X'nn'}] [.INPUT={ <i>ddname</i> SYSIN[({ <i>cccc</i> \$\$\$E})]}] [.PICTURE= <i>length,startloc</i> , { <i>character-string</i> ' P'n'IB'n'}] [.NAME={({ <i>namelist</i>) (<i>namelist-or-(copygroup)</i>)}] [.EXIT= <i>routinename</i>]
---------	--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

注: CREATE ステートメントごとに含めることのできるキーワードは 1 つまでです。

QUANTITY=*n*

この CREATE ステートメントが生成するレコードの数を指定します。各レコードの内容は、他のパラメーターによって指定されます。QUANTITY および

INPUT の両方がコーディングされた場合、そして指定された数量が入力データ・セット内のレコード数よりも大きい場合、作成されるレコード数は処理される入力レコード数に生成されるデータを加えた数 (指定値を上限とする) となります。

デフォルト: QUANTITY を省略して INPUT を指定しない場合、作成される出力レコードは 1 つだけです。QUANTITY を省略して INPUT を指定する場合、作成されるレコード数は入力データ・セット内のレコード数と同じです。

QUANTITY および INPUT の両方がコーディングされた場合、そして QUANTITY が入力データ・セット内のレコード数よりも小さい場合、QUANTITY で指定した数のレコードだけが出力データ・セットに書き込まれます。

FILL={character}'IX'nn}

出力レコード内で、レコードを構成する他の操作より前の各バイトに入れる値を指定します。コーディングできる値は以下のとおりです。

'character'

EBCDIC に変換して出力レコードの各バイトに入れる文字を指定します。

X'nn'

出力レコードの各バイトに入れる 2 桁の 16 進数字を指定します (たとえば、FILL=X'40'、または FILL=X'FF' など)。

デフォルト: 2 進ゼロ (X'00') が出力レコードに入れられます。

INPUT={ddname}SYSIN [({cccc}\$\$\$E)]]

出力レコードの作成にレコードが使用される入力データ・セットを定義します。INPUT をコーディングした場合、QUANTITY もコーディングしてください。そうしないと、残りの入力レコードがすべてこの CREATE ステートメントによって処理されます。INPUT がこの CREATE ステートメントによって参照される FD ステートメントで指定されている場合、入力レコードを読み取るために、対応する DD 名を CREATE ステートメントに指定しなければなりません。コーディングできる値は以下のとおりです。

ddname

入力データ・セットを定義する DD ステートメントの DD 名を指定します。

SYSIN[({cccc}\$\$\$E)]]

SYSIN データ・セットに、出力レコードの作成に使用されるレコード (ユーティリティ制御ステートメント以外の) が含まれるように指定します。SYSIN をコーディングした場合、入力レコードはこの CREATE ステートメントの後に続きます (例外として、CREATE ステートメントが REPEAT グループに含まれる場合は、入力レコードはグループの最後の CREATE ステートメントの後に続きます)。『cccc』値は、1~4 文字の任意の組み合わせです。cccc をコーディングした場合、入力レコードの最後は 1 桁目に含まれる文字を含むレコードによって指示されます。

cccc のデフォルト値は \$\$\$E です。INPUT=SYSIN を使用するとき cccc の値をコーディングしない場合、SYSIN 内の入力レコードの最後を示すために \$\$\$E を使用しなければなりません。先頭の 3 文字は、ドル記号の X'5B' です。

PICTURE=*length,startloc*, {'*character-string*'| **P**'*n*'|**B**'*n*'}

ユーザー指定のパターンの長さ、開始位置、および内容を指定します。

PICTURE および **NAME** の両方を省略した場合、**CREATE** ステートメントで指定した充てん文字が適用される出力レコードの各バイトに入ります。コーディングできる値は以下のとおりです。

length

パターン (『ピクチャー』) が占めるバイト数を指定します。 *length* は、**FD** ステートメントの **LENGTH** パラメーター値以下でなければなりません。

startloc

ピクチャーが開始する開始位置を (適当な出力レコード内に) 指定します。

'*character-string*'

適当なレコードに入れる文字ストリングを指定します。文字ストリングは定義された開始位置で左寄せされます。72 桁目に非ブランクを入れて次のステートメントの 4 桁目からストリングを継続することにより、文字ストリングを 71 桁目で分割できます。

P'*n*'

パック 10 進に変換して出力レコードまたは定義済みフィールド内で右寄せ (定義された長さおよび開始位置の範囲で) される 10 進数を指定します。

B'*n*'

2 進に変換して出力レコードまたは定義済みフィールド内で右寄せ (定義された長さおよび開始位置の範囲で) される 10 進数を指定します。

NAME={(*namelist*)| (*namelist-or-(copygroup)*)}

適当な出力レコード内に含める事前定義されたフィールドの名前 (複数も可) を指定します。 **NAME** および **PICTURE** の両方を省略した場合、**CREATE** ステートメントで指定した充てん文字が適用される出力レコードの各バイトに入ります。コーディングできる値は以下のとおりです。

(*namelist*)

適当な出力レコード内に含めるフィールド (複数も可) の名前 (複数も可) を指定します。複数の名前を指定するときは、コンマで区切ります。各フィールド (指名した **FD** ステートメントで事前に定義されている) は、その名前を **CREATE** ステートメントに指定した順序で出力レコードに含められます。1 つの名前だけをコーディングする場合、括弧はオプションです。

(*namelist-or-(copygroup)*)

フィールドの一部または全部を出力レコードにコピーするように指定します。つまり、選択したフィールドは出力レコード内に複数回出現することになります。コピーされるフィールドは、以下のように指定します。

(**COPY**=*n,name1*[,*name2*] [...])

ここで、*n* は示されたフィールドをグループとして扱い、この **CREATE** ステートメントで生成する各出力レコード内で *n* 回コピーするように指定します。任意の数の *copygroups* を **NAME** と共に含められます。*copygroup* に含められるフィールド名は、最大 20 までです。

コピーしないフィールド名は、*copygroups* の前、後、または間に、**NAME** パラメーターの *copygroups* によって指定できます。

たとえば、以下のようになります。

```
NAME=(NAME1,(COPY=2,NAME2),NAME3,(COPY=4,NAME4))
```

EXIT=routinename

各出力レコードを書き込む前に IEBDG から制御を受け取るルーチンの名前を指定します。IEBDG によって出力ルーチンを指定することについての情報は、377 ページの『付録 C. ユーティリティー・プログラムのユーザー出口を指定する』を参照してください。

END ステートメント

END ステートメントを使用して、ユーティリティー制御ステートメントのセットの終了を示します。制御ステートメントの各セットは任意の数の入力データ・セットに対応させることができますが、出力データ・セットは 1 つだけです。

END ステートメントの構文は次のとおりです。

[label]	END	
---------	-----	--

IEBDG の例

以下の各例では、IEBDG の使用方法の一部を示します。表 18 は IEBDG の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 18. IEBDG の例の一覧表

操作	データ・セット編成	装置	注釈	例
ユーティリティー制御ステートメントから出力レコードを作成	順次	ディスク	出力はブロック化されています。	3
ユーティリティー制御ステートメントから区分メンバーを作成	区分	ディスク	出力はブロック化されています。メンバーごとに 1 セットのユーティリティー制御ステートメント。	5
区分メンバーおよび入力ストリームからのレコードを変更	区分、順次	ディスク	再ブロックが実行されます。出力レコードの各ブロックには、10 の変更された区分入力レコード、および 2 つの入力ストリーム・レコードが含まれます。	4
選択したフィールドに 2 進ゼロを配置	順次	磁気テープ	入力および出力はブロック化されています。	1
リップル英字パターン	順次	磁気テープ、ディスク	入力および出力はブロック化されています。	2
ユーザー作成のロールおよびウェーブ・パターン	順次	ディスク	出力レコードはユーティリティー制御ステートメントから作成されます。	6

各例で、実際の装置番号の代わりに **disk** または **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: 順次データ・セットからコピーされたレコードに 2 進ゼロを入れる

この例では、順次データ・セットからコピーされた 100 レコードの 2 つのフィールドに 2 進ゼロを入れます。操作の後、コピーされたデータ・セット (OUTSET) 内の各レコードには位置 20 から 29、および 50 から 59 に 2 進ゼロが含まれます。

```
//CLEAROUT JOB ...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//SEQIN DD DSNAME=INSET,UNIT=tape,DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
// LABEL=(,NL),VOLUME=SER=222222
//SEQOUT DD DSNAME=OUTSET,UNIT=tape,DISP=(,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
// VOLUME=SER=222333,LABEL=(,NL)
//SYSIN DD *
DSD OUTPUT=(SEQOUT),INPUT=(SEQIN)
FD NAME=FIELD1,LENGTH=10,STARTLOC=20
FD NAME=FIELD2,LENGTH=10,STARTLOC=50
CREATE QUANTITY=100,INPUT=SEQIN,NAME=(FIELD1,FIELD2)
END
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SEQIN DD では、順次入力データ・セット (INSET) を定義しています。データ・セットは最初、ラベルなしテープ・ボリュームに書き込まれています。
- SEQOUT DD では、テスト・データ・セット (OUTSET) を定義しています。出力レコードは入力レコードと同じですが、例外として位置 20 から 29、および 50 から 59 には操作の終了時に 2 進ゼロが含まれます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- DSD はユーティリティー制御ステートメントのセットの開始を示し、入力および出力データ・セットを定義する DD ステートメントを参照します。
- 最初および 2 番目の FD ステートメントは、2 つの 10 バイト・フィールド (FIELD1 および FIELD2) を作成します。これらのフィールドにパターンは指定されないため、各フィールドにはデフォルトの 2 進ゼロが充てんされます。フィールドは、各出力レコードの 20 および 50 番目のバイトから開始します。
- CREATE は、事前に定義されたフィールド (FIELD1、FIELD2) の内容が各出力レコードの対応する開始位置に入れられている、100 の出力レコードを作成します。データ・セット INSET からの入力レコードは、出力レコードの基礎として使用されます。
- END は、ユーティリティー制御ステートメントの終了を示します。

例 2: 10 バイトの英字パターンのリップル

この例では、10 バイトの英字パターンをリップルします。ジョブ・ステップが終了すると、最初の出力レコードには『ABCDEFGHIJ』が入り、それに続いて位置 11

から 80 には入力レコードからのデータが入ります。2 番目のレコードには『BCDEFGHIJK』が入り、それに続いて位置 11 から 80 には入力レコードからのデータが入り、以下同様となります。

```
//RIPPLE JOB ...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//SEQIN DD DSN=INSET,UNIT=tape,VOL=SER=222222,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(OLD,KEEP)
//SEQOUT DD DSN=OUTSET,UNIT=disk,VOLUME=SER=111111,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
// DISP=(,KEEP),SPACE=(TRK,(10,10))
//SYSIN DD *
DSD OUTPUT=(SEQOUT),INPUT=(SEQIN)
FD NAME=FIELD1,LENGTH=10,FORMAT=AL,ACTION=RP,STARTLOC=1
CREATE QUANTITY=100,INPUT=SEQIN,NAME=FIELD1
END
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SEQIN DD では、入力順次データ・セット (INSET) を定義しています。データ・セットは最初、標準ラベル・テープ・ボリュームに書き込まれています。
- SEQOUT DD では、テスト出力データ・セット (OUTSET) を定義しています。10トラックの1次スペースおよび10トラックの2次スペースがディスク・ボリューム上の順次データ・セットに割り振られます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- DSD はユーティリティー制御ステートメントのセットの開始を示し、入力および出力データ・セットを定義する DD ステートメントを参照します。
- FD ステートメントは、パターン ABCDEFGHIJ が最初に入れている 10 バイトのフィールドを作成します。データは各出力レコードが書き込まれた後にアップルされます。
- CREATE は、事前に定義されたフィールド (FIELD1) の内容が含まれる 100 の出力レコードを作成します。CREATE ステートメントは、データ・セット INSET からの入力レコードを出力レコードの基礎として使用します。
- END は、ユーティリティー制御ステートメントの終了を示します。

例 3: ユーティリティー制御ステートメントから出力レコードを作成する

この例では、出力レコードの全体をユーティリティー制御ステートメントから作成します。出力レコードを作成するために、3つのフィールドが作成および使用されます。それらのフィールドの2つでは、英字データが切り捨てられます。他の1つのフィールドは数値フィールドで、出力レコードが書き込まれるたびに1ずつ増分(索引付け)します。105ページの図15は、ジョブ・ステップが終了したときの出力レコードの内容を示しています。

フィールド 1	フィールド 2	充てん	フィールド 3 (パック 10 進数)		
1	31	61	71	75	80
ABCDEFGHIJKLMNQRSTUUVWXYZABCD	ABCDEFGHIJKLMNQRSTUUVWXYZABCD	FF ... FF	00 ... 00	0123456789	0C
BCDEFGHIJKLMNQRSTUUVWXYZABCD	ABCDEFGHIJKLMNQRSTUUVWXYZABC	FF ... FF	00 ... 00	0123456789	1C
CDEFGHIJKLMNQRSTUUVWXYZABCD	ABCDEFGHIJKLMNQRSTUUVWXYZAB	FF ... FF	00 ... 00	0123456789	2C
DEFGHIJKLMNQRSTUUVWXYZABCD	ABCDEFGHIJKLMNQRSTUUVWXYZA	FF ... FF	00 ... 00	0123456789	3C
EFGHIJKLMNQRSTUUVWXYZABCD	ABCDEFGHIJKLMNQRSTUUVWXYZ	FF ... FF	00 ... 00	0123456789	4C

図 15. ジョブ・ステップが終了したときの出力レコード

72

```

//UTLYONLY JOB ...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//SEQOUT DD DSNAME=OUTSET,UNIT=disk,DISP=(,KEEP),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//          SPACE=(TRK,(10,10)),VOLUME=SER=111111
//SYSIN DD DATA
DSD OUTPUT=(SEQOUT)
FD NAME=FIELD1,LENGTH=30,STARTLOC=1,FORMAT=AL,ACTION=TL
FD NAME=FIELD2,LENGTH=30,STARTLOC=31,FORMAT=AL,ACTION=TR
FD NAME=FIELD3,LENGTH=10,STARTLOC=71,PICTURE=10,
          P'1234567890',INDEX=1
CREATE QUANTITY=100,NAME=(FIELD1,FIELD2,FIELD3),FILL=X'FF'
END
/*

```

各制御ステートメントについての説明は以下のとおりです。

- SEQOUT DD では、テスト出力データ・セットを定義しています。10トラックの1次スペースおよび10トラックの2次スペースがディスク・ボリューム上の順次データ・セットに割り振られます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- DSD はユーティリティー制御ステートメントのセットの開始を示し、出力データ・セットを定義する DD ステートメントを参照します。
- FD は、出力レコードの作成に使用される3つのフィールドの内容を定義します。最初のフィールドには、各出力レコードが書き込まれた後で左を切り捨てられる30バイトの英字データが含まれます。2番目のフィールドには、各出力レコードが書き込まれた後で右を切り捨てられる30バイトの英字データが含まれます。3番目のフィールドは、各レコードが書き込まれた後で増加するパック10進数(1234567890)を含む10バイトのフィールドです。
- CREATE は、事前に定義されたフィールド(FIELD1、FIELD2、およびFIELD3)の内容が含まれる100の出力レコードを作成します。各レコードが書き込まれた後で、FIELD1 および FIELD2 の幅が完全に復元されることに注目してください。
- END は、ユーティリティー制御ステートメントの終了を示します。

例 4: 出力メンバーの基礎としてメンバーおよび入力レコードを使用する

この例では、入力ストリームからの2つのメンバーおよび入力レコードを区分出力メンバーの基礎として使用します。12の出力レコードの各ブロックには、入力区分メンバーからの10の変更済みレコード、および入力ストリームからの2つのレ

コードが含まれます。図 16 は、ジョブ・ステップが終了したときの出力区分メンバーの内容を示しています。

入力	出力レコード
部門 21 (INSET1(MEMBA) レコード 1 の右端 67 バイト)	1 12 のうち最初のブロック
●	●
●	●
●	●
●	●
部門 21 (INSET1(MEMBA) レコード 10 の右端 67 バイト)	10
入力レコード 1 入力ストリームから	11
入力レコード 2 入力ストリームから	12
部門 21 (INSET1(MEMBA) レコード 11 の右端 67 バイト)	1 12 のうち 2 番目のブロック
●	●
●	●
●	●
●	●
部門 21 (INSET1(MEMBA) レコード 20 の右端 67 バイト)	10
入力レコード 3 入力ストリームから	11
入力レコード 4 入力ストリームから	12
●	
●	
●	
●	
部門 21 (INSET1(MEMBA) レコード 91 の右端 67 バイト)	1 12 のうち 10 番目のブロック
●	●
●	●
●	●
●	●
部門 21 (INSET1(MEMBA) レコード 100 の右端 67 バイト)	10
入力レコード 19 入力ストリームから	11
入力レコード 20 入力ストリームから	12
部門 21 (INSET2(MEMBA) レコード 1 の右端 67 バイト)	1 12 のうち 11 番目のブロック
●	●
●	●
●	●
●	●
部門 21 (INSET2(MEMBA) レコード 10 の右端 67 バイト)	10
入力レコード 21 入力ストリームから	11
入力レコード 21 入力ストリームから	12
●	
●	
●	

図 16. ジョブ・ステップが終了したときの出力区分メンバー

```
//MIX      JOB    ...
//STEP1    EXEC  PGM=IEBDG
//SYSPRINT DD  SYSOUT=A
//PARIN1   DD   DSN=INSET1(MEMBA),UNIT=disk,DISP=OLD,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DSORG=PS),
//           VOLUME=SER=111111
//PARIN2   DD   DSN=INSET2(MEMBA),UNIT=disk,DISP=OLD,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=960,DSORG=PS),
//           VOLUME=SER=222222
//PAROUT   DD   DSN=PARSET(MEMBA),UNIT=disk,DISP=(,KEEP),
//           VOLUME=SER=333333,SPACE=(TRK,(10,10,5)),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=960,DSORG=PO)
//SYSIN    DD   DATA
           DSD   OUTPUT=(PAROUT),INPUT=(PARIN1,PARIN2)
           FD    NAME=FIELD1,LENGTH=13,PICTURE=13,'DEPARTMENT 21'
           REPEAT QUANTITY=10,CREATE=2
           CREATE QUANTITY=10,INPUT=PARIN1,NAME=FIELD1
           CREATE QUANTITY=2,INPUT=SYSIN
```

(入力レコード 1 ~ 20)

```
$$$E
  REPEAT  QUANTITY=10,CREATE=2
  CREATE  QUANTITY=10,INPUT=PARIN2,NAME=FIELD1
  CREATE  QUANTITY=2,INPUT=SYSIN
```

(入力レコード 21 ~ 40)

```
$$$E
  END
/*
```

各制御ステートメントについての説明は以下のとおりです。

- PARIN1 DD では、入力区分メンバーの 1 つを定義しています。
- PARIN 2 DD では、入力区分メンバーの 2 つ目を定義しています。(メンバーが異なる区分データ・セットから取られていることに注目してください。)
- PAROUT DD では、出力区分メンバーを定義しています。この例は、区分データ・セットがジョブ・ステップの前に存在しなかったと想定しています。つまり、この DD ステートメントは区分データ・セットのスペースを割り振ります。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- DSD はユーティリティー制御ステートメントのセットの開始を示し、入力および出力データ・セットを定義する DD ステートメントを参照します。
- FD は、ピクチャー『DEPARTMENT 21』が入れられる 13 バイトのフィールドを作成します。
- 最初の REPEAT ステートメントは、2 つの CREATE ステートメントの以下のグループを 10 回繰り返すように指示します。
- 最初の CREATE ステートメントは、10 の出力レコードを作成します。各出力レコードは、入力レコード (区分データ・セット INSET1 からの) および事前に定義された FIELD1 から作成されます。
- 2 番目の CREATE ステートメントは、2 つのレコードを入力ストリームで次に含まれている入力レコードから作成するように指示します。
- \$\$\$E レコードは、入力レコードを REPEAT ステートメントから分離します。次の REPEAT ステートメント・グループは直前のグループと同じですが、例外として異なる区分メンバーからのレコードを入力として使用します。
- END は、ユーティリティー制御ステートメントの終了を示します。

例 5: 3 つの出力データ・セットにレコードを作成して、それらを 3 つの区分データ・セット・メンバーに書き込む

制約事項 データ・セットがシステム管理または SMS データ・セットである場合、この例は使用できません。

この例では、出力レコードを 3 つのユーティリティー制御ステートメントのセットから作成して、1 つの区分データ・セットの 3 つのメンバーに書き込みます。出力レコードを作成するために、4 つのフィールドが作成および使用されます。フィールドの 2 つ (FIELD1 および FIELD3) では、英字データはシフトされます。FIELD2 は固定ゾーン 10 進数で、FIELD4 は固定英数字です。108 ページの図

17 は、ジョブ・ステップの最後での区分データ・セット・メンバーを示しています。

MEMBA			
フィールド 1	フィールド 3	フィールド 2	バイナリー・ゼロ
1	31	51	71 80
ABCDEF GHIJ KLMN OPQRST UVWZY ABCD	ABCDEF GHIJ KLMN OPQRST	00000000000000000000000000000001	充てん
BCDEF GHIJ KLMN OPQRST UVWZY ABCD	ABCDEF GHIJ KLMN OPQRS	00000000000000000000000000000001	充てん
CDEF GHIJ KLMN OPQRST UVWZY ABCD	ABCDEF GHIJ KLMN OPQR	00000000000000000000000000000001	充てん
DEFG HIJ KLMN OPQRST UVWZY ABCD	ABCDEF GHIJ KLMN OPQ	00000000000000000000000000000001	充てん

MEMBB			
フィールド 2	フィールド 3	フィールド 3	フィールド 3
1	21	41	61 80
00000000000000000000000000000001	ABCDEF GHIJ KLMN OP	ABCDEF GHIJ KLMN OP	ABCDEF GHIJ KLMN OP
00000000000000000000000000000001	ABCDEF GHIJ KLMNO	ABCDEF GHIJ KLMNO	ABCDEF GHIJ KLMNO
00000000000000000000000000000001	ABCDEF GHIJ KLMN	ABCDEF GHIJ KLMN	ABCDEF GHIJ KLMN
00000000000000000000000000000001	ABCDEF GHIJ KLM	ABCDEF GHIJ KLM	ABCDEF GHIJ KLM

MEMBC		
フィールド 4	フィールド 1	バイナリー・ゼロ
1	31	61 80
ABCDEF GHIJ KLMN OPQRST UVWXY Z0123	EFGHIJ KLMN OPQRST UVWZY ABCD	充てん
ABCDEF GHIJ KLMN OPQRST UVWXY Z0123	FGHIJ KLMN OPQRST UVWZY ABCD	充てん
ABCDEF GHIJ KLMN OPQRST UVWXY Z0123	GHIJ KLMN OPQRST UVWZY ABCD	充てん
ABCDEF GHIJ KLMN OPQRST UVWXY Z0123	HIJ KLMN OPQRST UVWZY ABCD	充てん

図 17. ジョブ・ステップが完了したときの区分データ・セット・メンバー

```
//UTSTS JOB ...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//PAROUT1 DD DSN=PARSET(MEMBA),UNIT=disk,DISP=(,KEEP),
//          VOLUME=SER=11111,SPACE=(TRK,(10,10,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//PAROUT2 DD DSN=PARSET(MEMBB),UNIT=AFF=PAROUT1,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//          DISP=OLD,VOLUME=SER=11111
//PAROUT3 DD DSN=PARSET(MEMBC),UNIT=AFF=PAROUT1,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//          DISP=OLD,VOLUME=SER=11111
//SYSIN DD DATA
DSD OUTPUT=(PAROUT1)
FD NAME=FIELD1,LENGTH=30,FORMAT=AL,ACTION=SL
FD NAME=FIELD2,LENGTH=20,FORMAT=ZD
FD NAME=FIELD3,LENGTH=20,FORMAT=AL,ACTION=SR
FD NAME=FIELD4,LENGTH=30,FORMAT=AN
CREATE QUANTITY=4,NAME=(FIELD1,FIELD3,FIELD2)
END
DSD OUTPUT=(PAROUT2)
CREATE QUANTITY=4,NAME=(FIELD2,(COPY=3,FIELD3))
END
DSD OUTPUT=(PAROUT3)
CREATE QUANTITY=4,NAME=(FIELD4,FIELD1)
END
/*
```

各制御ステートメントについての説明は以下のとおりです。

- PAROUT1 DD では、区分出力データ・セットの最初のメンバー (MEMBA) を定義しています。この例は、区分データ・セットがこのジョブ・ステップの前に存在しなかったと想定しています。つまり、この DD ステートメントはデータ・セットのスペースを割り振ります。
- PAROUT2 および PAROUT3 DD では、出力区分データ・セットのそれぞれ 2 番目および 3 番目のメンバーを定義しています。各 DD ステートメントが DISP=OLD および UNIT=AFF=PAROUT1 を指定することに注目してください。
- SYSIN DD では、入力ストリームに続く制御データ・セットを定義しています。
- DSD はユーティリティー制御ステートメントのセットの開始を示し、ユーティリティー制御ステートメントに適用できるメンバーを定義する DD ステートメントを参照します。
- FD は、後続の出力レコードの作成に使用されるフィールドの内容を定義します。
- CREATE は、事前に定義されたフィールドの組み合わせから 4 つのレコードを作成します。
- END は、ユーティリティー制御ステートメントの終了を示します。

例 6: 独自のパターンを使用してレコードを作成する

この例では、ユーザー作成の文字パターンを含む 10 のフィールドを使用して出力レコードを作成します。レコードが書き込まれた後、適用される FD ステートメントでの指定に従って、各フィールドはロールまたはウェーブされます。図 18 は、ジョブ・ステップが終了したときの出力レコードの内容を示しています。

FIELD1	FIELD2	FIELD3	FIELD4	FIELD5	FIELD6	FIELD7	FIELD8	FIELD9	FIELD10
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC
AAAAA	BBBBB	A AA	BB B	AAA	CCCCC	DDDD	C CC	DD D	CCC

図 18. ジョブ・ステップが終了したときの出力レコードの内容

```
//ROLLWAVE JOB ...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//OUTSET DD DSNAME=SEQSET,UNIT=disk,DISP=(,KEEP),
//          VOLUME=SER=SAMP,SPACE=(TRK,(10,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSIN DD *
```

72

```

DSD   OUTPUT=(OUTSET)
FD    NAME=FIELD1,LENGTH=8,PICTURE=8,'AAAAA',ACTION=RO
FD    NAME=FIELD2,LENGTH=8,PICTURE=8,'BBBBB ',ACTION=RO
FD    NAME=FIELD3,LENGTH=8,PICTURE=8,'A  AA ',ACTION=RO
FD    NAME=FIELD4,LENGTH=8,PICTURE=8,' BB  B',ACTION=RO
FD    NAME=FIELD5,LENGTH=8,PICTURE=8,' AAA ',ACTION=RO
FD    NAME=FIELD6,LENGTH=8,PICTURE=8,'  CCCC',ACTION=WV
FD    NAME=FIELD7,LENGTH=8,PICTURE=8,' DDDD ',ACTION=WV
FD    NAME=FIELD8,LENGTH=8,PICTURE=8,' C  CC ',ACTION=WV
FD    NAME=FIELD9,LENGTH=8,PICTURE=8,' DD  D',ACTION=WV
FD    NAME=FIELD10,LENGTH=8,PICTURE=8,' CCC ',ACTION=WV
CREATE QUANTITY=300,NAME=(FIELD1,FIELD2,FIELD3,
                           FIELD4,FIELD5,FIELD6,FIELD7,FIELD8,FIELD9,FIELD10)
END
/*

```

各制御ステートメントについての説明は以下のとおりです。

- **OUTSET DD** では、ディスク・ボリューム上の出力順次データ・セットを定義しています。10トラックの1次スペースおよび10トラックの2次スペースがデータ・セットに割り振られます。
- **SYSIN DD** では、入力ストリームに続く制御データ・セットを定義しています。
- **DSD** はユーティリティー制御ステートメントのセットの開始を示し、出力データ・セットを定義する **DD** ステートメントを参照します。
- **FD** は、後続の出力レコードの作成に使用されるフィールドを定義します。最初のロールまたはウェーブの方向および頻度は、フィールド内でのデータの位置によって決まります。
- **CREATE** は、事前定義されたフィールドの内容から300のレコードを作成します。
- **END** は、ユーティリティー制御ステートメントの終了を示します。

第 5 章 IEBEDIT (ジョブ・ストリームの編集) プログラム

IEBEDIT を使用して、選択したジョブまたはジョブ・ステップを含むデータ・セットを作成することができます。これらのジョブまたはジョブ・ステップは、後で処理するためにジョブ・ストリームに入れられます。

入力ジョブ・ストリームを編集して出力データ・セットに選択的にコピーするには、IEBEDIT を使用します。このプログラムでは、次のものをコピーできます。

- ジョブ (複数も可) の全体。これには、JOB ステートメントおよび関連している JOBLIB または JOBCAT ステートメント、そして JES2 または JES3 制御ステートメントが含まれます。
- 選択したジョブ・ステップ。これには、JOB ステートメント、JOB ステートメントに続く JES2 または JES3 制御ステートメント、そして関連している JOBLIB または JOBCAT ステートメントが含まれます。

選択したすべての JOB ステートメント、JES2 または JES3 制御ステートメント、JOBLIB または JOBCAT ステートメント、ジョブまたはジョブ・ステップは、入力データ・セットで表示されているのと同じ順序で出力データ・セットにも配置されます。JES2 または JES3 制御ステートメント、または JOBLIB または JOBCAT ステートメントは、それが選択した JOB ステートメントの後に続いている場合のみコピーされます。

IEBEDIT が、1~3 桁目に『.*』(ピリオド、ピリオド、アスタリスク) の文字がある入力レコードを含む、選択したジョブ・ステップに遭遇すると、プログラムは自動的にそのレコードを終了ステートメント (* ステートメント) に変換し、出力データ・セットに配置します。

『/*nonblank』は、JES2 または JES3 制御ステートメントを示します。

入出力

IEBEDIT は以下の入力を使用します。

- 入力データ・セット。これは、ジョブ・ストリームによって構成されている順次データ・セットです。入力データ・セットは、出力順次データ・セットを作成するためのソース・データとして使用されます。
- 制御データ・セット。これには出力データ・セットにジョブおよびジョブ・ステップの編成を指定するために使用する、ユーティリティー制御ステートメントが含まれています。

IEBEDIT は以下の出力を生成します。

- 出力データ・セット。これは、結果のジョブ・ストリームによって構成されている順次データ・セットです。
- メッセージ・データ・セット。これは、適当な制御ステートメント、エラー・メッセージ、および適用可能な場合はオプションとして出力データ・セットを含む順次データ・セットです。

IEBEDIT の戻りコードの詳細については、付録 Aを参照してください。

制御

IEBEDIT はジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。ジョブ制御ステートメントは、プログラムを実行またはロードして、プログラムによって使用され、生成されるデータ・セットを定義するのに必要です。ユーティリティー制御ステートメントは、プログラムの機能を制御するために使用されます。

ジョブ制御ステートメント

表 19 は、IEBEDIT で使用するジョブ制御ステートメントをまとめたものです。

表 19. IEBEDIT で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEBEDIT)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。データ・セットはシステム出力装置、磁気テープ・ボリュウム、直接アクセス・ボリュウムなどに出力できます。
SYSUT1 DD	カード読取装置、磁気テープ・ボリュウム、または直接アクセス装置の順次入力データ・セットを定義します。
SYSUT2 DD	カード穿孔装置、プリンター、磁気テープ・ボリュウム、または直接アクセス装置の順次出力データ・セットを定義します。
SYSIN DD	制御データ・セットを定義します。データ・セットは通常、入力ストリームにあります。プロシージャー・ライブラリーのメンバーとして、または入力ストリーム以外のどこかに存在している順次データ・セットとして定義することもできます。

SYSPRINT DD ステートメント

SYSPRINT データ・セットのブロック・サイズは、121 の倍数でなければなりません。このサイズでない場合は、ジョブ・ステップは終了して、戻りコード 8 が戻されます。

SYSUT1、SYSUT2、および SYSIN DD ステートメント

SYSIN、SYSUT1、および SYSUT2 SYSIN データ・セットのブロック・サイズは、80 の倍数でなければなりません。これらのレコード・サイズには、任意のブロック化因数を指定できます。

SYSUT1 データ・セットで選択した JOB ステートメントに続く JES2 または JES3 制御ステートメント、または JOBLIB DD ステートメントは、自動的に出力データ・セットにコピーされます。JOB ステートメントより前にある JES2 または JES3 制御ステートメントは、直前のジョブに属していると見なされます。最初の JOB ステートメントより前にある JES2 または JES3 制御ステートメントは、全体のコピーを要求する場合にのみ含まれます。

しかし、SYSUT1 データ・セットが入力ストリーム (SYSUT1 DD DATA) に含まれている場合は、『/*』以外の区切り文字が SYSUT1 DD DATA ステートメント内でコーディングされている場合にのみ、JES2 または JES3 制御ステートメントが含まれます。別の区切り文字をコーディングする方法については、*z/OS MVS JCL ユーザーズ・ガイド* を参照してください。別の区切り文字がコーディングされていない場合は、JES2 または JES3 制御ステートメントの最初の 2 文字が区切り文字として機能して、SYSUT1 データ・セットを終了させます。

ユーティリティー制御ステートメント

IEBEDIT は、ユーティリティー制御ステートメント EDIT だけを使用します。このステートメントの継続要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

EDIT ステートメントを使用して、入力データ・セット内のジョブのどのステップ (複数も可) を出力データ・セットに含めたいのかを示すことができます。EDIT ステートメントは、操作の中にいくつでも含めることができます。したがって、1 つの操作でいくつものジョブ・ステップを指定してデータ・セットを作成できます。

EDIT ステートメントは、対象の入力ジョブと同じ順序で指定する必要があります。入力データ・セットの全体をコピーするには、EDIT ステートメントを省略します。

EDIT ステートメントの構文は次のとおりです。

[label]	EDIT	[START=jobname] [,TYPE={POSITION INCLUDE EXCLUDE}] [,STEPNAME=(namelist)] [,NOPRINT]
---------	------	-----------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

START=jobname

EDIT ステートメントが適用される入力ジョブの名前を指定します。(それぞれの EDIT ステートメントは、別々のジョブに適用されなければなりません。) START が TYPE および STEPNAME なしで指定されている場合、JOB ステートメントと、指定したジョブのすべてのジョブ・ステップが出力に含まれます。

デフォルト: START が省略されていて 1 つの EDIT ステートメントのみが提供されている場合、入力データ・セットで遭遇する最初のジョブが処理されます。START が、最初の EDIT ステートメント以外の EDIT ステートメントから省略されている場合、入力データ・セット内で次に見つかる JOB ステートメントまで処理が継続されます。

TYPE={POSITION|INCLUDE|EXCLUDE}

出力データ・セットの内容を指定します。コーディングできる値は以下のとおりです。

POSITION

出力に JOB ステートメント、STEPNAME パラメーターで指定されているジョブ・ステップ、およびそのジョブ・ステップに続くすべてのステップが

含まれるように指定します。指定したステップよりも前にあるジョブ・ステップはすべて、操作から省略されます。デフォルトは POSITION です。

INCLUDE

出力データ・セットに JOB ステートメントおよび STEPNAME パラメータで指定されているすべてのジョブ・ステップが含まれるように指定します。

EXCLUDE

出力データ・セットに、JOB ステートメントおよび STEPNAME パラメータで指定されているステップ以外のジョブに属するすべてのジョブ・ステップが含まれるように指定します。

STEPNAME=(namelist)

処理したいジョブ・ステップの名前を指定します。

namelist は、単一のジョブ・ステップ名、コンマによって区切られているステップ名のリスト、またはハイフンによって区切られているステップの順次的な範囲(たとえば STEPA-STEPE) にすることができます。これらのどんな組み合わせも、1 つの *namelist* で使用することができます。複数のステップ名が指定されている場合は、*namelist* 全体を括弧で囲まなければなりません。

TYPE=POSITION によってコーディングされている場合、最初のジョブ・ステップは、STEPNAME によって出力データ・セットに配置されるように指定されます。このステップよりも前にあるジョブ・ステップは、出力データ・セットにはコピーされません。

TYPE=INCLUDE または TYPE=EXCLUDE によってコーディングされる場合は、STEPNAME によって操作に含めるか、または操作から除外することになるジョブ・ステップの名前を指定します。たとえば、

STEPNAME=(STEPS,STEPF-STEPL,STEPZ) は、ジョブ・ステップ STEPA、STEPF から STEPL、および STEPZ を操作に含めるか、操作から除外するかを指定します。

STEPNAME が省略される場合、EDIT ステートメントで名前が指定されている入力ジョブ全体がコピーされることとなります。ジョブ名が指定されていない場合は、最初に遭遇するジョブが処理されます。

NOPRINT

メッセージ・データ・セットに出力データ・セットのリストが含まれないように指定します。

デフォルト: 結果の出力が、メッセージ・データ・セットにリスト表示されません。

IEBEDIT の例

以下の各例では、IEBEDIT の使用方法の一部を示します。115 ページの表 20 は、これらの使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 20. IEBEDIT の例の一覧表

操作	装置	注釈	例
EDIT	ディスク	JOBA の JOB ステートメント、ジョブ・ステップ STEPF、および後続のすべてのステップをコピーします。	4
EDIT	ディスクおよび磁気テープ	1 つのジョブからのジョブ・ステップを含め、別のジョブからのジョブ・ステップを除外します。	3
EDIT	磁気テープ	1 つのジョブを出力データ・セットにコピーします。	1
EDIT	磁気テープ	3 つのジョブのそれぞれから選択してジョブ・ステップをコピーします。	2
EDIT	磁気テープ	入力データ・セット全体をコピーします。『.*』レコードは、出力データ・セットでは『/*』ステートメントに変換されます。	5
EDIT	磁気テープ	JES2 制御ステートメントを含む、入力データ・セット全体をコピーします。	6

各例で、実際の装置番号の代わりに **disk** または **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: 1 つのジョブをコピーする

この例では、1 つのジョブ (JOBA) をすべてのジョブ・ステップと一緒に出力データ・セットにコピーします。入力データ・セットには、JOBA、JOB B、JOB C という 3 つのジョブが含まれます。

```
//EDIT1 JOB ...
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=INJOBS,UNIT=tape,
// DISP=(OLD,KEEP),VOL=SER=001234
//SYSUT2 DD DSN=OUTTAPE,UNIT=tape,DISP=(NEW,KEEP),
// VOL=SER=001235,DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),
//SYSIN DD *
EDIT START=JOBA
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、入力データ・セット INJOBS を定義しています。データ・セットは、標準ラベル・テープ・ボリューム (001234) にあります。
- SYSUT2 DD では、OUTTAPE という出力データ・セットを定義しています。データ・セットは、1 次データ・セットとして標準ラベル・テープ・ボリューム (001235) に入ります。システムは、最適なブロック・サイズを選択します。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- EDIT は、JOBA 全体がコピーされることを示します。

例 2: 3 つのジョブからのステップをコピーする

この例では、3 つのジョブのそれぞれからジョブ・ステップをコピーします。入力データ・セットには、3 つのジョブが含まれています。1 つ目は、STEP A、STEP B、STEP C、および STEP D を含む JOBA、2 つ目は、STEP E、STEP F、および STEP G を含む JOB B、3 つ目は、STEP H および STEP J を含む JOB C です。

```
//EDIT2 JOB ...
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=INJOBS,DISP=(OLD,KEEP),VOLUME=SER=001234,
// UNIT=tape
//SYSUT2 DD DSN=OUTSTRM,UNIT=tape,DISP=(NEW,KEEP),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80),LABEL=(2,SL)
//SYSIN DD *
EDIT START=JOBA,TYPE=INCLUDE,STEPNAME=(STEPD,STEPJ)
EDIT START=JOB,TYPE=INCLUDE,STEPNAME=STEPE
EDIT START=JOB,TYPE=INCLUDE,STEPNAME=STEPJ
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、入力データ・セット INJOBS を定義しています。データ・セットは、標準ラベル・テープ・ボリューム (001234) にあります。
- SYSUT2 DD では、出力データ・セット OUTSTRM を定義しています。データ・セットは、第 2 データ・セットとして標準ラベル・テープ・ボリューム (001235) に入ります。ブロック・サイズが短いと、効率が非常に悪くなります。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- EDIT ステートメントは、以下のように JOB ステートメントとジョブ・ステップをコピーします。
 1. JOBA の JOB ステートメントおよびステップ STEPC と STEPD。
 2. JOBB の JOB ステートメントおよび STEPE。
 3. JOBC の JOB ステートメントおよび STEPJ。

例 3: 1 つのジョブからのステップを含め、別のジョブからのステップを除外する

この例では、1 つのジョブからのジョブ・ステップを含め、別のジョブからのジョブ・ステップを除外します。入力データ・セットには、3 つのジョブが含まれています。1 つ目は、STEPA、STEPB、STEPD、および STEPD を含む JOBA、2 つ目は、STEPE、STEPF、および STEPG を含む JOBB、3 つ目は、STEPH および STEPJ を含む JOBC です。

```
//EDIT3 JOB ...
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=INSET,UNIT=disk,DISP=(OLD,KEEP),
// VOLUME=SER=111111
//SYSUT2 DD DSNAME=OUTTAPE,UNIT=tape,LABEL=(,NL),
// DCB=(DEN=2,RECFM=FB,LRECL=80,BLKSIZE=8160),
// DISP=(,KEEP)
//SYSIN DD *
EDIT START=JOB,TYPE=INCLUDE,STEPNAME=(STEPF-STEPG)
EDIT START=JOB,TYPE=EXCLUDE,STEPNAME=STEPJ
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、入力データ・セット INSET を定義しています。データ・セットは、ディスク・ボリューム (111111) にあります。
- SYSUT2 DD では、出力データ・セット OUTTAPE を定義しています。データ・セットは、1 次または唯一のデータ・セットとしてラベルなし (800 ビット / インチ) テープ・ボリュームに入ります。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。

- EDIT ステートメントは、以下のように、JOB ステートメントとジョブ・ステップをコピーします。
 1. JOBB の JOB ステートメントおよびステップ STEPF と STEPG。
 2. JOBC の JOB ステートメントおよび STEPH (STEPJ は除外)。

例 4: JOBA および JOB STEPF のステートメントをコピーする

この例では、JOBA の JOB ステートメント、ジョブ・ステップ STEPF、および後続のすべてのステップをコピーします。入力データ・セットには 1 つのジョブ (JOBA) が含まれます。これには、STEPSA から STEPL ままで含まれます。ジョブ・ステップ STEPSA から STEPE は出力データ・セットに含まれません。

```
//EDIT4 JOB ...
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=INSTREAM,UNIT=disk,
//          DISP=(OLD,KEEP),VOLUME=SER=111111
//SYSUT2 DD DSNAME=OUTSTREAM,UNIT=disk,
//          DISP=(,KEEP),VOLUME=SER=222222,SPACE=(TRK,2)
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=24000),
//SYSIN DD *
          EDIT START=JOBA,TYPE=POSITION,STEPNAME=STEPF
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、入力データ・セット INSTREAM を定義しています。データ・セットは、ディスク・ボリューム (111111) にあります。
- SYSUT2 DD では、出力データ・セット OUTSTREAM を定義しています。データ・セットは、ディスク・ボリューム (222222) に入ります。2 つのトラックが出力データ・セットに割り振られています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- EDIT は、JOBA の JOB ステートメントおよびジョブ・ステップ STEPF から STEPL をコピーします。

例 5: 入力データ・セット全体をコピーする

この例では、入力データ・セット全体をコピーします。列 1 から 3 に文字『.*』を含むレコードは、出力データ・セットでは『/*』ステートメントに変換されます。

```
//EDIT5 JOB ...
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSNAME=OUTTAPE,UNIT=tape,
//          VOLUME=SER=001234,DISP=(NEW,KEEP),
//          DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN DD DUMMY
//SYSUT1 DD DATA
//BLDGDGIX JOB ...
//          EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//SYSIN DD *
          BLDG INDEX=A.B.C,ENTRIES=10,EMPTY
.*
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT2 DD では、出力データ・セット OUTTAPE を定義しています。データ・セットは、テープ・ボリューム (001234) の 1 次データ・セットになります。
- SYSIN DD では、ダミー制御データ・セットを定義しています。
- SYSUT1 DD では、入力ストリームの中のそれ以降の入力データ・セットを定義しています。ジョブは、ステートメント『/*』に遭遇するときに停止します。(したがって、SYSUT1 には BLDGDGIX JOB ステートメント、EXEC ステートメント、SYSPRINT、DD1、および SYSIN DD ステートメントが含まれます。)

例 6: 新しい区切り文字を含めるためにデータ・セット全体をコピーする

この例では、新しい区切り文字 (JP) がコーディングされたため、JES2 制御ステートメントを含む入力データ・セット全体をコピーします。これを行わないと、JES2 制御ステートメントにある『/*』が入力を停止します。

```
//EDIT6 JOB ...
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=TAPEOUT,UNIT=tape,
//          VOL=SER=001234,LABEL=(,SL),DISP=(NEW,KEEP)
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//SYSIN DD DUMMY
//SYSUT1 DD DATA,DLM=JP
//LISTVTOC JOB ...
/*MESSAGE JOB NEEDS VOLUME 338000
//FSTEP EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD2 DD UNIT=disk,VOL=SER=338000,DISP=OLD
//SYSIN DD *
LISTVTOC FORMAT,VOL=disk=338000
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT2 DD では、出力データ・セット TAPEOUT を定義しています。データ・セットは、標準ラベル・テープ・ボリューム (001234) の 1 次データ・セットになります。
- SYSIN DD では、ダミー制御データ・セットを定義しています。
- SYSUT1 DD では、入力ストリームの中のそれ以降の入力データ・セットを定義しています。DLM パラメーターは、文字 JP を、入力データの区切り文字として機能するように定義します。
- IEBEDIT は、『/*』ステートメントを介して JOB ステートメントをコピーします (LISTVTOC および MESSAGE ジョブ・ステートメント、FSTEP EXEC ステートメント、SYSPRINT、DD2 および SYSIN DD ステートメントが含まれます)。

第 6 章 IEBGENER (データ・セットの順次コピー / 順次生成) プログラム

IEBGENER を使用して、以下のタスクを実行できます。

- 順次データ・セット、区分データ・セット または PDSE のメンバー、または z/OS UNIX システム・サービス (z/OS UNIX) ファイル (HFS ファイルなど) のバックアップ・コピーを作成すること。
- 順次データ・セットまたは z/OS UNIX ファイルから、区分データ・セット、PDSE、 および区分データ・セットまたは PDSE のメンバーを生成すること。
- 区分メンバーを作成してそれらを既存のデータ・セットにマージすることにより、既存の区分データ・セットまたは PDSE を拡張すること。
- 編集した順次または区分データ・セットまたは PDSE を生成すること。
- 2 バイト文字セット・データを含むデータ・セットを操作すること。
- 順次データ・セット、区分データ・セットまたは PDSE のメンバー、および z/OS UNIX ファイルを印刷すること。
- データ・セットの論理レコード長を再ブロックしたり、変更したりすること。
- 順次出力データ・セットのユーザー・ラベルをコピーすること。(382 ページの『ユーザー・ラベルの処理』を参照。)
- ラベルを処理するルーチンの編集機能と出口を提供し、入力データを操作し、キーを作成し、永続入力/出力エラーを処理すること。ユーザー出口ルーチンのリンケージ規則の説明については、377 ページの『付録 C. ユーティリティ・プログラムのユーザー出口を指定する』を参照してください。

推奨: DFSORT™ 製品をインストールしている場合は、データ・セットまたはメンバーの未編集コピーを作成するときに、IEBGENER の代わりに ICEGENER を使用しなければなりません。この製品は、IEBGENER という名前ですでにインストールされていることがあります。これを使用すると通常はパフォーマンスが良くなります。

バックアップ・コピーの作成

順次データ・セット、区分データ・セットのメンバーまたは PDSE のバックアップ・コピーを生成するには、データ・セットまたはメンバーを IBM サポートの出力装置にコピーします。たとえば、テープ間のコピー、DASD からテープへのコピーなどが行えます。

直接アクセス・ボリュームに入っているデータ・セットは、名前を変更することによって、そのボリュームにもコピーできます。

制約事項: IEBGENER を使用して、区分データ・セットを順次データ・セットとして処理するときは、IEBGENER はディレクトリー項目処理 (ロード・モジュール・ライブラリーのメンバーの属性のコピーなど) を行いません。IEBCOPY は、ディレクトリー項目処理を実行します。

順次入力からの区分データ・セットまたは PDSE の生成

ユーティリティー制御ステートメントを使用することにより、論理的に順次データ・セットをレコード・グループに分割して、そのレコード・グループにメンバー名を割り当てることができます。IEBGENER は、新しく作成されたメンバーを出力区分データ・セットまたは PDSE に配置します。

入力または出力データ・セットにスパン・レコードが含まれる場合は、区分データ・セットまたは PDSE を生成することはできません。

図 19 は、区分データ・セットまたは PDSE が、入力として使用される順次データ・セットから生成される様子を示したものです。図の左側は、順次データ・セットを示しています。ユーティリティー制御ステートメントは、順次データ・セットをレコード・グループに分割し、それぞれのレコード・グループにメンバー名を提供するときに使用します。図の右側は、順次入力から生成された区分データ・セットまたは PDSE を示しています。

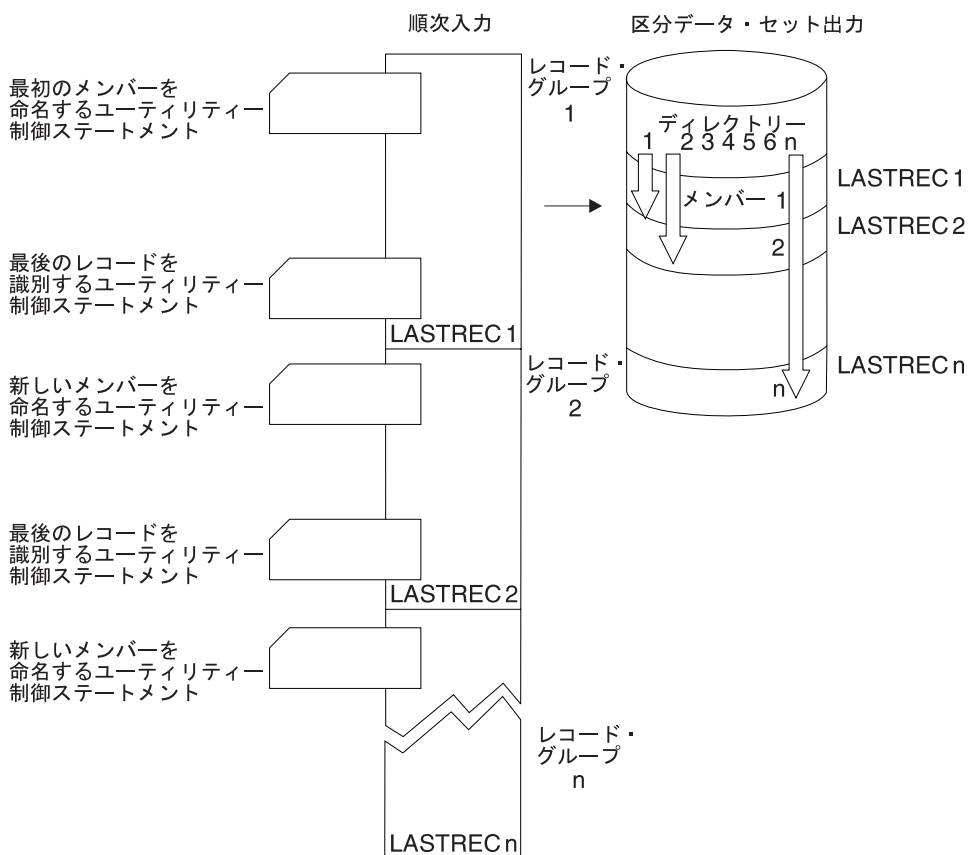


図 19. IEBGENER による、順次入力からの区分データ・セットまたは PDSE の作成

区分データ・セットまたは PDSE にメンバーを追加する

IEBGENER を使用して、区分データ・セットまたは PDSE にメンバーを追加できます。IEBGENER は順次入力からメンバーを作成し、それらをデータ・セットに追加します。マージ操作 (区分ディレクトリーの並べ替え) は、プログラムによって自動的に実行されます。

図 20 は、順次入力が、既存の区分データ・セットまたは PDSE にマージされるメンバーに変換される様子を示したものです。図の左側は、図の真ん中にある区分データ・セットまたは PDSE とマージされる順次入力を示しています。ユーティリティー制御ステートメントは、順次データ・セットをレコード・グループに分割し、それぞれのレコード・グループにメンバー名を提供するときに使用します。図の右側は、拡張された区分データ・セットまたは PDSE を示しています。使用可能なスペースにあった、順次データ・セットからのメンバー B、D、および F が、区分ディレクトリーに順次配列されていることに注目してください。

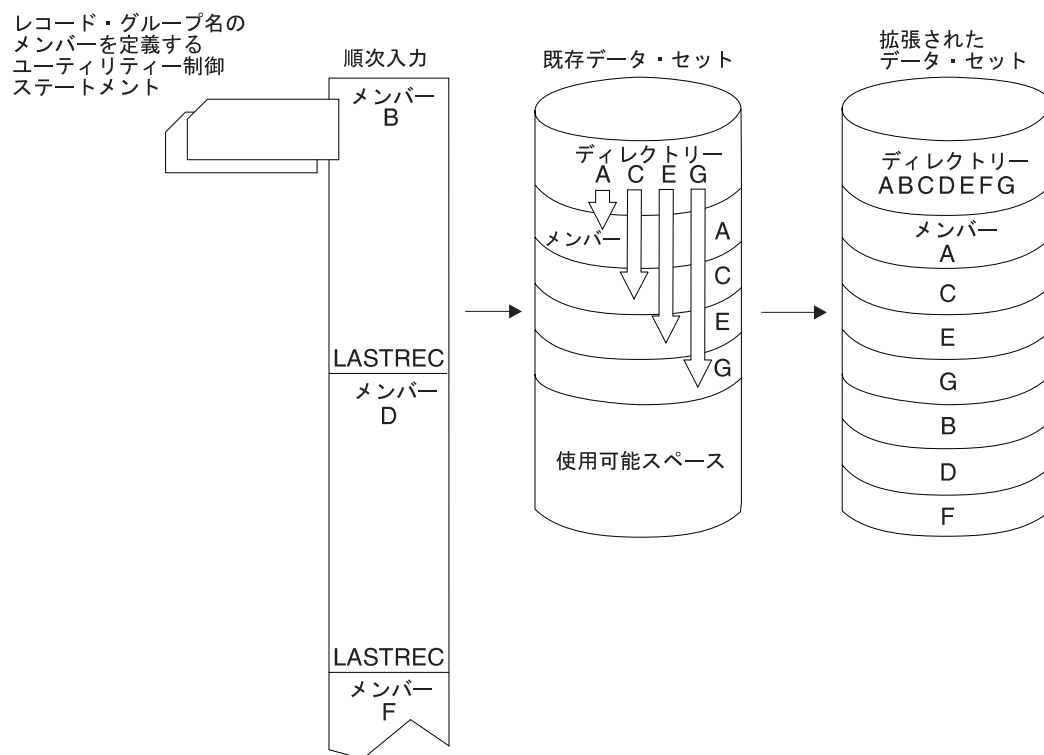


図 20. IEBGENER による、区分データ・セットまたは PDSE へのメンバーの追加

編集したデータ・セットの生成

IEBGENER を使用して、編集した順次または区分データ・セット、または PDSE を生成できます。ユーティリティー制御ステートメントを使用することにより、1つのレコード、レコードのグループ、選択したレコードのグループ、またはデータ・セット全体に適用される編集情報を指定できます。

編集したデータ・セットは、次のようにして生成できます。

- レコードに定義されているデータ・フィールドを再配列または省略する。
- リテラル情報を置換データとして提供する。
- データをパック 10 進からアンパック 10 進モードに変換するか、アンパック 10 進をパック 10 進モードに変換するか、または BCDIC (ここでは 2 進化 10 進交換コードの標準 H 文字セット) を EBCDIC モードに変換する。BCDIC から EBCDIC への変換の詳細については、*z/OS DFSMS データ・セットの使用法* を参照してください。
- データ・セットに 2 バイト文字セットが含まれているときに、シフトアウト / シフトイン文字 X'0E' および X'0F' を追加するか、または削除する。

図 21 は、編集した順次データ・セットの一部を示したものです。図の左側は、編集を実行する前のデータ・セットを示しています。ユーティリティー制御ステートメントは、編集するレコード・グループと編集情報を提供するレコード・グループを識別するために使用します。この図では、リテラル置換情報が、定義済みフィールドにある情報のために提供されています。(データは同じ方法で再配列されるか、省略されるか、または変換されます。) レコード・グループにあるそれぞれのレコードの BBBB フィールドは、CCCC に置換されます。図の右側には、編集後のデータ・セットが示されています。

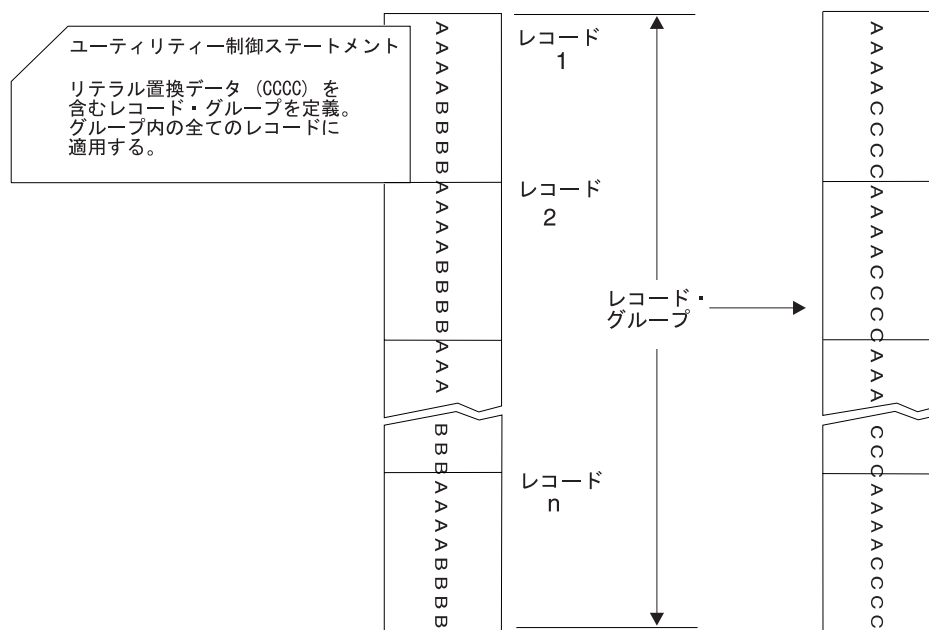


図 21. IEBGENER による、順次データ・セットの編集

入出力データ・セットが可変長スパン (VS) または可変長ブロック・スパン (VBS) によって構成されており、同じブロック・サイズと論理レコード長がある場合は、IEBGENER は、データ・セットを編集するためには使用できません。このような場合、編集を指定しているユーティリティー制御ステートメントは無視されます。つまり、入力データ・セットから読み取るそれぞれの物理レコードのために、ユーティリティーは、出力データ・セット上に編集されていない物理レコードを作成します。

論理レコード長の変更

IEBGENER を使用して、入力論理レコード長とは異なる論理レコード長がある固定長または可変長レコードを含む、再ブロックされた出力データ・セットを生成します。

入力データ・セットの論理レコード長とは異なる論理レコード長があるデータ・セットを作成するときは、出力レコードにあるそれぞれのバイトがどこから来るか (リテラル、入力レコードのいずれか) を指定しなければなりません。未指定のフィールドは、予測不能データを含むことになります。入出力データ・セットの両方に可変長または可変長ブロック化レコード・フォーマットがある場合は、論理レコード長を変更することはできません。

2 バイト文字セット・データでの IEBGENER の使用

IEBGENER を使用して、2 バイト文字セット (DBCS) データを含むデータ・セットをコピー、編集、再ブロック、または印刷できます。さらに、DBCS データを含む順次データ・セットを、区分データ・セットに変換することもできます。2 バイト文字セットは、1 バイト文字セットによって表すには複雑過ぎる言語を表すために使用します。たとえば、日本語を表示するには 2 バイト文字セットが必要です。

DBCS データは通常、シフトアウト / シフトイン X'0E' および X'0F' の 1 バイト文字に囲まれ、これによってそのデータが単一バイトではなく、対になっているバイトとして処理されなければならないことを示します。シフトアウト は、データが 1 バイト文字セットからシフトアウトして、2 バイト文字セット・ストリングにシフトイン することを示し、シフトイン は、2 バイト文字セット・ストリングが 1 バイト文字セット・ストリングにシフトインすることを示します。シフトイン / シフトアウト文字を追加または削除するには、IEBGENER を使用します。しかし、シフトイン / シフトアウト文字をデータ・セットに追加する場合は、出力データ・セットの論理レコード長を判別するときに、必要となる付加的なバイトを考慮しなければなりません。さらに、DBCS データの妥当性検査にも IEBGENER を使用できます。すべての文字の各バイトの値が X'41' および X'FE' 間の値 (両端の文字を含める) であるか、または DBCS 文字が DBCS スペース (X'4040') でない限り、DBCS データは『有効』とは見なされません。この検査によって、それぞれの DBCS 文字を印刷可能にすることができます。

DBCS データを含むデータ・セットを印刷するためには、DBCS ストリングがシフトアウト / シフトイン文字によって囲まれていなければなりません。

入出力

IEBGENER は以下の入力を使用します。

- コピーするか、編集するか、区分データ・セットまたは PDSE に変換するか、または既存のデータ・セットにマージされるメンバーに変換するデータが入っている入力データ・セット。入力は、順次データ・セット、区分データ・セット、PDSE のいずれかです。
- ユーティリティー制御ステートメントが含まれている、制御データ・セット。編集を実行するか、または出力データ・セットが区分データ・セットまたは PDSE になる場合は、制御データ・セットが必要です。

- JCL EXEC ステートメントの PARM パラメーターまたは動的起動と同等のもの (125 ページの『EXEC ステートメント』を参照してください)。

IEBGENER は以下の出力を生成します。

- 出力データ・セット。これは、順次 (ラージ・フォーマット順次を含む) または区分のどちらかにすることができます。出力データ・セットは、新規データ・セット (現在のジョブ・ステップ中に作成されたもの) か、拡張された既存の区分データ・セットまたは PDSE のどちらかにすることができます。1 区分データ・セットまたは PDSE が作成される場合、これは新規ディレクトリー項目の新規メンバーになります。直前のディレクトリー項目から情報がコピーされることはありません。
- 情報メッセージ (たとえば、ユーティリティー制御ステートメントの内容) およびエラー・メッセージを含むメッセージ・データ・セット。

データまたはメンバーを、有効期限が切れていない既存のデータ・セットに追加するときに、メッセージ IEC507D が 2 回出されます。このことは、入出力データ・セットが 2 回オープンされるために生じます。

IEBGENER をアプリケーション・プログラムから起動した場合、動的にデータ・セットを割り振るには、IEBGENER を呼び出す前に SVC 99 を出します。

このユーティリティーによって出されるメッセージについては *z/OS MVS システム・メッセージ 第 7 巻 (IEB-IEE)* に文書化されています。

IEBGENER の戻りコードについては、付録 A を参照してください。

制御

IEBGENER は、ジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。ジョブ制御ステートメントは IEBGENER を実行し、プログラムによって使用および生成されるデータ・セットを定義します。ユーティリティー制御ステートメントは、IEBGENER の機能を制御します。

ジョブ制御ステートメント

表 21 は、IEBGENER で使用するジョブ制御ステートメントをまとめたものです。

表 21. IEBGENER で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEBGENER)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。データ・セットは、システム出力装置、テープ・ボリューム、または DASD ボリュームに書き込むことができます。

表 21. IEBGENER で使用するジョブ制御ステートメント (続き)

ステートメント	用途
SYSUT1 DD	入力データ・セットを定義します。順次データ・セット、区分データ・セットのメンバー、PDSE、z/OS UNIX ファイルのバックアップ・コピーを定義できます。順次データ・セットは、基本フォーマット、ラージ・フォーマット、拡張フォーマット、圧縮フォーマット、スプールされた入力、テープ、カード・リーダー、TSO 端末またはダミーであることができます。
SYSUT2 DD	出力データ・セットを定義します。このデータ・セットは、順次データ・セット、区分データ・セットあるいは PDSE のメンバー、区分データ・セットあるいは PDSE あるいは z/OS UNIX ファイルを定義できます。順次データ・セットは、基本フォーマット、ラージ・フォーマット、拡張フォーマット、圧縮フォーマット、スプールされた出力、テープ、カード・パンチ、プリンター、TSO 端末またはダミーであることができます。
SYSIN DD	制御データ・セットを定義するか、または出力が順次であり、編集が指定されていない場合は、DUMMY を指定します。制御データ・セットは通常、入力ストリームにあります。区分データ・セットや PDSE の中のメンバーとして定義することもできます。

EXEC ステートメント

EXEC ステートメントは IEBGENER を使用するたびに必要となります。

EXEC ステートメント上に PARM='SDB=xxxxx' のような値をコーディングすることができます。これは、出力ブロック・サイズがどのようなソースからも指定されていない場合に限り、有効です。システムは、「システム決定ブロック・サイズ」と呼ばれる機能を通じて、最適な値を計算します。この PARM 値は、ブロック・サイズが 32 760 バイトを超えることができるかどうかを制御します。SDB は EXEC ステートメントの唯一の有効なパラメーターです。xxxxx にコーディングできる値は、以下のとおりです。

INPUT または YES

入力ブロック・サイズが 32 760 バイトより大きい場合は、デフォルトの出力ブロック・サイズを 32 760 バイトより大きくすることができます。そうでない場合は、出力ブロック・サイズを 32 760 より大きくすることはできません。

SMALL

デフォルトの出力ブロック・サイズは 32 760 バイト以下です。

LARGE

デフォルトの出力ブロック・サイズを 32 760 バイトより大にすることができます。古いプログラムでは、これほど大きなブロックを持つデータ・セットを読むことはできません。

NO デフォルトの出力ブロック・サイズは 32 760 バイト以下ですが、SMALL を指定したときよりもブロック・サイズは非効率となります。

システム・プログラマーは、SDB のデフォルト値をセットするために、SYS1.PARMLIB の DEVSUP_{xx} メンバーに COPYSDB をセットします。COPYSDB に対する IBM 提供のデフォルトは INPUT です。OS/390 バージョン 2 リリース

10 の導入される以前には、IEBGENER は PARM 値を無視して、SDB=NO とコーディングされたかのように操作していました。現時点では、32 760 バイトを超すブロック・サイズに対応できるのは、磁気テープ装置とダミー・データ・セットだけです。この制約事項を、固定的なものと考えする必要はありません。将来のオペレーティング・システムのレベルでは、より多くの種類のデータ・セットで、より大きなブロックを処理できるようになります。

以下の条件は、出力データ・セットが 32 760 バイトより大きいブロック・サイズを持つ原因となります。

- BLKSIZE キーワードを DD ステートメントにコーディングし、装置がその値を扱えるようにする。
- 出力データ・セットが存在し大きな BLKSIZE を持つ場合は、DISP=MOD を SYSUT2 DD ステートメントにコーディングする。
- 出力デバイス・タイプが大きなブロック・サイズをサポートしているが、それをコーディングしていなくて、SDB=LARGE が有効になっている。
- 出力デバイス・タイプが大きなブロック・サイズをサポートしているが、それをコーディングしなかった。入力データ・セットが大きなブロック・サイズを持っていて SDB=INPUT が有効になっている。

注: 最後の 2 つのケースでは、システムがブロック・サイズをブロック・サイズの上限に制限します。IEBGENER は、装置に対して大きすぎるブロック・サイズ値は算出しません。固定ブロック・レコードについては、ブロック・サイズの上限は LRECL の整数倍である必要はありません。ブロック・サイズの上限は、以下の中で使用可能な最初の 1 つです。

- 出力 DD ステートメント上の BLKSZLIM キーワード。
- データ・クラスのブロック・サイズの上限。これはデータ・セットが SMS 管理でない場合も同様です。DATACLAS キーワードを DD ステートメントにコーディングするか、システムの ACS ルーチンがデータ・クラスを割り当てる。
- SYS1.PARMLIB の DEVSUP_{xx} メンバー中の TAPEBLKSZLM キーワード。システム・プログラマーが、これをシステムのデフォルト値として設定する。
- 32 760 バイト。

IEBGENER に複数のバッファを使用すると、プログラムを使用するのに必要な仮想記憶域の量が増えます。80A 異常終了を避けるには、付加的なストレージの REGION パラメーターを変更するか、または追加する必要があります。

バッファ数のデフォルトは 5 です。これを指定変更するには、SYSUT1 または SYSUT2 DD ステートメントで BUFNO または NCP を指定します。

IEBGENER を実行する前に、プログラムを実行するのに必要な (仮想記憶域の) 領域サイズを計算してから、REGION パラメーターでこの値を指定する必要がある場合があります。

領域サイズを計算するには、次の計算式を使用します。

$$\text{領域サイズ} = 50K + (2 + \text{SYSUT1 BUFNO}) * (\text{SYSUT1 BLKSIZE}) + (2 + \text{SYSUT2 BUFNO}) * (\text{SYSUT2 BLKSIZE})$$

BUFNO を JCL では使用しない場合、デフォルト値 5 を使用してください。

141 ページの『例 5: 非ブロック化ディスク・ファイルから磁気テープにブロック化コピーを生成する』から取った次の情報は、必要な領域サイズを計算する方法を示しています。

```
SYSUT1 BUFNO = 20
SYSUT1 BLKSIZE = 2K
SYSUT2 BUFNO = 無指定の場合はデフォルトが使用されます
SYSUT2 BLKSIZE = 32K
```

領域サイズ = $50K + (2 + 20) * (2K) + (2 + 5) * (32K)$

したがって、領域サイズ = 318K (すなわち REGION=318K)

SYSPRINT DD ステートメント

SYSPRINT DD ステートメントは IEBGENER を使用するときに必要となります。SYSPRINT データ・セットのブロック・サイズは、121 の倍数でなければなりません。このレコード・サイズには、任意のブロック化因数を指定できますが、ブロック・サイズの最大値は、32670 バイトです。

SYSUT1 DD ステートメント

IEBGENER の入力データ・セットには、SYSUT1 で指定されているように、固定長、可変長、不定形式、または可変長スパン・レコードを入れることができます。*unlike* 属性を指定して連結データ・セットを使用することはできません。ただし、IEBGENER への入力としてのブロック・サイズまたは装置タイプは例外です。SYSUT1 のデータ・セットの連結規則は、DASD およびテープにあるデータ・セットを連結するのではない限り、BSAM の連結データ・セットの連結規則と同じです。DASD およびテープにあるデータ・セットを連結する場合、連結規則は QSAM の連結規則と同じです。BSAM および QSAM にデータ・セットを連結する方法については、*z/OS DFSMS* データ・セットの使用法 を参照してください。

SYSUT1 データ・セットがシステム入力ストリームにあり、それに JCL ステートメントが含まれている場合、SYSUT1 DD * ではなく SYSUT1 DD DATA をコーディングします。

入力データ・セットのブロック・サイズは、データ・セット・ラベル (DSCB)、磁気テープ・ラベル、または DD ステートメントで使用可能である必要があります。磁気テープ・データ・セットの場合は、ブロック・サイズは 32 760 バイトを超えることができます。

入力データ・セットのデフォルトのレコード・フォーマットは、不定形式 (U) です。データ・セットが新しいか、不定形式であるか、またはダミー・データ・セットである場合は、レコード・フォーマットを指定しなければなりません。

入力論理レコード長は、レコード・フォーマットが固定長ブロック、可変長スパン、または可変長ブロック・スパンであるときに指定するか、またはデータ・セットが新しいときかまたはダミー・データ・セットのときに指定しなければなりません。入力と出力の論理レコード長が使用不可であり、レコード・フォーマットが V または VB の場合、IEBGENER は、最大の 32760 バイトを使用して、ブロック・サイズの -4 バイトに最大レコード長を設定します。他の場合はすべて、デフォルトの論理レコード長 80 が使用されます。

入力データ・セットにスパン・レコードが含まれている場合は、区分データ・セットまたは PDSE は生成できません。

SYSUT1 と SYSUT2 DD ステートメントの両方が標準的なユーザー・ラベル (SUL) を指定する場合、IEBGENER は SYSUT1 から SYSUT2 にユーザー・ラベルをコピーします。ユーザー・ラベル処理に使用できるオプションについては、382 ページの『ユーザー・ラベルの処理』を参照してください。

SYSUT2 DD ステートメント

IEBGENER の出力データ・セットには、SYSUT2 で指定されているように、固定長、可変長、不定形式、または可変長スパン・レコードを入れることができます。(可変長スパン・レコードを含めることができない区分データ・セットまたは PDSE は例外です)。SYSUT2 DD ステートメントで新たに最大ブロック長を指定した場合、IEBGENER はレコードをブロック化し直す可能性があります。固定長または可変長レコードを再ブロックしている場合、キーを保存できるのはユーザーがキーを保存するために出口ルーチンを提供している場合だけです。可変長スパン・レコードを再ブロックするときは、キーを保存することはできません。

出力データ・セットがカード穿孔装置上にあるか、またはプリンター (スプール・データ・セットではない) 上にある場合、DCB 情報を SYSUT2 DD ステートメントに指定しなければなりません。

データ・セットが DASD または磁気テープにあり、レコード・フォーマットと、出力データ・セットのために論理レコード長が JCL で指定されておらず、データ・クラスがそれらを提供しない場合、それぞれの値は入力データ・セットからコピーされます。RECFM が未定義の場合、論理レコード長はコピーされません。

注: IEBGENER に SMF ダンプ・データ・セットを指定して使用している場合、IEBGENER は、データを切り捨てることができます。SMF ダンプ・プログラム IFASMFDP は、IEBGENER が許可するよりも大きい論理レコード長を持つデータ・セットを作成します。したがって、IEBGENER はデータ・セットを論理レコード長 32 760 に切り捨て、結果として 7 バイトを失うこととなります。これらの 7 バイトに実際のデータがない場合は、その切り捨てによって失うものはなにもありません。しかし、IEBGENER に SMF ダンプ・プログラムを指定して使用する場合には、注意する必要があります。

論理レコード長およびレコード・フォーマットが指定されているか、または入力データ・セットに使用可能な場合は、出力ブロック・サイズを指定する必要はありません。EXEC ステートメント (SDB= の説明は、125 ページの『EXEC ステートメント』を参照)、入力データ・セット、および出力装置タイプが、出力ブロック・サイズ (無指定の場合) を左右します。出力装置タイプが磁気テープまたはダミーの場合は、ブロック・サイズは 32 760 バイトを超えることができます。BLKSIZE を指定すると、EXEC ステートメントの PARM 値の SDB は、影響を受けません。SYSUT2 DD ステートメントで RECFM、LRECL、および BLKSIZE が使用可能である場合のその影響の詳細は、129 ページの表 22 を参照してください。

編集が実行されることになっており、さらにレコード・フォーマットが固定長ブロック、可変長スパン、または可変長ブロック・スパンであるときに、出力論理レコード長を指定しなければなりません。

入出力データ・セットにスパン・レコードが含まれている場合は、区分データ・セットまたは PDSE は生成できません。

表 22 は、SYSUT2 DD ステートメント の RECFM、LRECL、および BLKSIZE の可用性の影響を示しています。

表 22. 出力 DD ステートメントの影響：「有効」は、その項目をコーディングしているか、またはその値がデータ・セット・ラベルまたはデータ・クラスに存在することを意味します。” (SDB の説明については 125 ページの『EXEC ステートメント』を参照してください。)

RECFM	LRECL	BLKSIZE	結果
有効	有効	有効	すべてが使用されています。
有効	有効	省略	ブロック・サイズは、システムによって決定されます。(SDB= の説明については 125 ページの『EXEC ステートメント』を参照してください。)
有効	省略	有効	RECFM が未定義の場合を除き、LRECL は入力からコピーされます。指定された RECFM と BLKSIZE は使用されます。
有効	省略	省略	RECFM が未定義の場合を除き、LRECL は入力からコピーされます。BLKSIZE がその装置に対して長すぎる場合 (この場合は、システムが最適な BLKSIZE を決定) を除き、入力から BLKSIZE がコピーされます。
省略	有効	有効	RECFM は入力からコピーされます。指定された LRECL と BLKSIZE は使用されます。
省略	有効	省略	RECFM は入力からコピーされます。指定された LRECL は使用されます。BLKSIZE は、システムによって決定されます。
省略	省略	有効	RECFM と LRECL はその入力からコピーされますが、RECFM が未定義の場合は LRECL はコピーされません。
省略	省略	省略	BLKSIZE が装置に対して大き過ぎない場合には、3 項目はすべてコピーされます。この場合には、システムは最適な BLKSIZE を決定します。

SYSIN DD ステートメント

SYSIN DD ステートメントは IEBGENER を使用するたびに必要となります。SYSIN データ・セットのブロック・サイズは、80 の倍数でなければなりません。このブロック・サイズに、任意のブロック化因数を指定できます。SYSIN がダミーである場合は、IEBGENER は入力データ・セットを連続してコピーします。

ユーティリティー制御ステートメント

IEBGENER は、ユーティリティー制御ステートメントによって制御されます。130 ページの表 23 は、各ステートメントとそれぞれの指定順序をまとめたものです。

制御ステートメントは必要に応じて制御データ・セット (CDS) に組み込みます。ユーティリティー制御ステートメントが制御データ・セットに組み込まれていない場合は、入力データ・セット全体が順番にコピーされます。

出力が順次であり、編集が実行されるときに、1つの GENERATE ステートメント、および必要なだけの RECORD ステートメントが使用されます。出口ルーチンを使用する場合は、EXITS ステートメントが必要です。

出力を区分化するときに、要求されたとおりに、1つの GENERATE ステートメント、出力メンバーごとに1つの MEMBER ステートメント、および RECORD ステートメントが使用されます。出口ルーチンを使用する場合は、EXITS ステートメントが必要です。

継続行は、4～16 桁目で開始されなければなりません。72 桁目の非ブランク継続行はオプションです。

表 23. IEBGENER で使用するユーティリティー制御ステートメント

ステートメント	用途
GENERATE	制御データ・セットにあるメンバー名および別名の数、レコード ID、リテラル、および編集情報を示します。
EXITS	ユーザー・ルーチンが提供されていることを示します。
LABELS	ユーザー・ラベル処理を指定します。
MEMBER	作成する区分データ・セットまたは PDSE のメンバーのメンバー名および別名を指定します。
RECORD	処理するレコード・グループを定義し、編集情報を提供します。

GENERATE ステートメント

GENERATE ステートメントは、出力を区分するとき、編集を実行するとき、ユーザー・ルーチンが提供されているとき、ラベル処理が指定されているときに必要です。GENERATE ステートメントは、他のどの IEBGENER ユティリティー・ステートメントよりも前に指定しなければなりません。これにエラーが含まれているか、または他のステートメントと矛盾している場合は、IEBGENER が終了します。

GENERATE の構文は次のとおりです。

[label]	GENERATE	[,MAXNAME= <i>n</i>] [,MAXFLDS= <i>n</i>] [,MAXGPS= <i>n</i>] [,MAXLITS= <i>n</i>] [,DBCS={YES NO}]
---------	----------	---------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

MAXNAME=*n*

数 1～3276 を指定します。これは後の MEMBER ステートメントに表示されているメンバー名と別名の合計より大きいか等しい数です。MAXNAME は、1 つまたは複数の MEMBER ステートメントがある場合に必要です。

MAXFLDS=*n*

数 1～4095 を指定します。これは後の RECORD ステートメントに表示されている FIELD パラメーターの合計より大きいか等しい数です。MAXFLDS は、後の RECORD ステートメントに FIELD パラメーターがある場合に必要です。

MAXGPS=*n*

数 1～2520 を指定します。これは後の RECORD ステートメントに表示されている IDENT または IDENTG の合計より大きいか等しい数です。MAXGPS は、後の RECORD ステートメントに IDENT または IDENTG パラメーターがある場合に必要です。

MAXLITS=*n*

数 1～2730 を指定します。これは後の RECORD ステートメントの FIELD リテラルに表示されている文字の合計より大きいか等しい数です。FIELD パラメーターにリテラルとして使用されている DBCS 文字は、それぞれ 2 文字と数えられます。

MAXLITS は、後の RECORD ステートメントの FIELD パラメーターにリテラルがある場合に必要です。MAXLITS は、IDENT または IDENTG パラメーターで使用されているリテラルには提供されません。

DBCS={YES|NO}

入力データ・セットに 2 バイト文字セット・データが含まれている場合に指定します。

EXITS ステートメント

EXITS ステートメントは、IEBGENER で使用する出口ルーチンを識別するときに使用します。出口ルーチンの詳細については、377 ページの『付録 C. ユーティリティー・プログラムのユーザー出口を指定する』を参照してください。

ユーザー・ラベルをデータ・セット記述子として処理する方法の詳細、およびユーザー・ラベルの合計の詳細については、382 ページの『ユーザー・ラベルの処理』を参照してください。

EXITS ステートメントの構文は次のとおりです。

[<i>label</i>]	EXITS	[INHDR = <i>routinename</i>] [OUTHDR = <i>routinename</i>] [INTLR = <i>routinename</i>] [OUTTLR = <i>routinename</i>] [KEY = <i>routinename</i>] [DATA = <i>routinename</i>] [IOERROR = <i>routinename</i>] [TOTAL =(<i>routinename,size</i>)]
------------------	--------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

INHDR=*routinename*

ユーザー入力ヘッダー・ラベルを処理するルーチンの名前を指定します。

OUTHDR=*routinename*

ユーザー出力ヘッダー・ラベルを作成するルーチンの名前を指定します。出力データ・セットが区分である場合は、OUTHDR は無視されます。

INTLR=routinename

ユーザー入力トレーラー・ラベルを処理するルーチンの名前を指定します。

OUTTLR=routinename

ユーザー出力トレーラー・ラベルを処理するルーチンの名前を指定します。出力データ・セットが区分である場合は、OUTTLR は無視されます。

KEY=routinename

出力レコード・キーを作成するルーチンの名前を指定します。このルーチンは、可変長スパン (VS) または可変長ブロック・スパン (VBS) タイプ・レコードによって構成されているデータ・セットが処理されるときには、制御は受け取りません。これは、このようなタイプのデータにはキーが許可されていないために生じます。

DATA=routinename

IEBGENER によって処理される前に、物理レコード (可変長ブロック・タイプ・レコードの場合は論理レコード) を修正するルーチンの名前を指定します。

IOERROR=routinename

永続入出力エラー条件を処理するルーチンの名前を指定します。

TOTAL=(routinename,size)

各レコードを作成する前に、ユーザー出力ルーチンが提供されるように指定します。キーワード OPTCD=T を必ず SYSUT2 DD ステートメントに指定しなければなりません。順次データ・セットを処理するのに IEBGENER が使用されている場合にのみ TOTAL は有効です。コーディングしなければならない値は以下のとおりです。

routinename

ユーザーの合計処理用ルーチンの名前を指定します。

size

合計、カウンター、ポインターなどを含む必要のあるバイトの数を指定します。size は、10 進数の整数としてコーディングしなければなりません。

LABELS ステートメント

LABELS ステートメントは、ユーザー・ラベルが IEBGENER によってデータとして扱われるかどうかを指定します。このオプションの詳細については、382 ページの『ユーザー・ラベルの処理』を参照してください。

LABELS ステートメントは、ユーザー・ラベルを出力データ・セットにコピーしないことを指定するとき、ユーザー・ラベルを、SYSIN データ・セットのデータ部分のレコードから出力データ・セットにコピーするとき、またはユーザーのラベル処理ルーチンによってユーザー・ラベルを処理した後で、ユーザー・ラベルを出力データ・セットにコピーするときに使用します。有効な LABELS ステートメントが 2 つ以上組み込まれている場合は、最後の LABELS ステートメント以外はすべて無視されます。

LABELS ステートメントの構文は次のとおりです。

[label]	LABELS	[DATA={YES NO ALL ONLY INPUT}]
---------	--------	--------------------------------

各項目の意味は以下のとおりです。

DATA={YES|NO|ALL|ONLY|INPUT}

ユーザー・ラベルが IEBGENER によってデータとして扱われるかどうかを指定します。コーディングできる値は以下のとおりです。

YES

EXIT ステートメントで指定したラベル処理ルーチンによってリジェクトされないユーザー・ラベルが、すべてデータとして扱われるように指定します。ラベルのデータとしての処理は、標準的な戻りコードにしたがって終了します。デフォルトは YES です。

NO

ユーザー・ラベルをデータとして扱わないことを指定します。標準的なユーザー・ラベル (SUL) 出口を非アクティブにするには、標準外ラベル (NSL) のある入出力データ・セットを処理するときに NO を指定しなければなりません。

ALL

現在処理されているグループにあるすべてのユーザー・ラベルをデータとして扱うように指定します。戻りコード 16 は、IEBGENER がユーザー・ラベルのグループの残りの処理を完了させ、ジョブ・ステップを停止するようにさせます。

ONLY

ユーザー・ヘッダー・ラベルだけをデータとして扱うことを指定します。ユーザー・ヘッダー・ラベルは戻りコードに関係なく、データとして処理されます。ジョブは OPEN ルーチンから戻った時点で終了します。

INPUT

出力データ・セットのユーザー・ラベルに、SYSIN のデータ部分にある 80 バイトの入力レコードが提供されるように指定します。ユーザー・ラベルとして扱われる必要のある入力レコードの数は、RECORD ステートメントによって識別されなければなりません。

LABELS DATA=NO は、標準外ラベル (NLS) のある入出力データ・セットが処理されるときに、標準的なユーザー・ラベル (SUL) が非アクティブで存在するように指定します。

MEMBER ステートメント

MEMBER ステートメントは、出力データ・セットを区分編成にするときに使用します。1 つの MEMBER ステートメントを、IEBGENER によって作成されるそれぞれのメンバーに含めなければなりません。MEMBER ステートメントは、新規メンバーの名前と別名を提供します。

MEMBER ステートメントに続くすべての RECORD ステートメントは、その MEMBER ステートメントで指定されている項目を参照します。MEMBER ステートメントがない場合は、出力データ・セットは順次編成になります。

MEMBER ステートメントの構文は次のとおりです。

[label]	MEMBER	NAME=(name[,alias 1][,alias 2][,...])
---------	--------	---------------------------------------

各項目の意味は以下のとおりです。

NAME=(*name*[,*alias*][,...])

メンバー名を指定します。その後別名のリストが続きます。ステートメントで別名が指定されていない場合は、メンバー名を括弧で囲む必要はありません。

RECORD ステートメント

RECORD ステートメントは、レコード・グループを定義し、編集情報を提供するために使用します。レコード・グループには、同様に処理されることになるレコードが含まれています。

RECORD ステートメントは、出力が区分編成のとき、編集を実行するとき、出力データ・セットのユーザー・ラベルを、SYSIN データ・セットのデータ部分にあるレコードから作成するときを使用します。RECORD ステートメントは、グループの最後のレコードをリテラル名によって識別することによって、レコード・グループを定義します。

RECORD ステートメントが使用されていない場合は、入力データ・セット全体またはメンバーは、編集なしで処理されます。複数の RECORD ステートメントが IEBGENER 制御ステートメント・ストリームに現れることがあります。

RECORD ステートメント内では、1 つの IDENT または IDENTG パラメーターを使用してレコード・グループを定義できます。1 つまたは複数の FIELD パラメーターを使用して、レコード・グループに適用できる編集情報を提供できます。さらに 1 つの LABELS パラメーターを使用してこのステートメントのすぐ後に出力ラベル・レコードがあることを示すことができます。

出力ヘッダー・ラベルと出力トレーラー・ラベルの両方が SYSIN データ・セットに含まれる場合、1 つの RECORD ステートメント (LABELS パラメーターを含む) を含めて、ユーザー・ヘッダー・ラベルとして扱われる入力レコードの数を示し、さらに別の RECORD ステートメントを (これも LABELS パラメーターを含む) ユーザー・トレーラー・ラベルのために含めなければなりません。最初のそのような RECORD ステートメントは、ユーザー・ヘッダー・ラベルの数を示し、次のステートメントは、ユーザー・トレーラー・ラベルの数を示します。出力トレーラー・ラベルのみが SYSIN データ・セットに含まれている場合、SYSIN データ・セットに出力ヘッダー・ラベルがないことを示す RECORD ステートメントが含まれていなければなりません (LABELS=0)。このステートメントは、トレーラー・ラベル入力レコードの開始を示す RECORD LABELS=*n* ステートメントの前になければなりません。

LABELS オプションの詳細については、382 ページの『ユーザー・ラベルの処理』を参照してください。

RECORD ステートメントの構文は次のとおりです。

[<i>label</i>]	RECORD	[{ IDENT IDENTG }=(<i>length</i> ,' <i>name</i> ', <i>input-location</i>)] [, FIELD =([<i>length</i>],[<i>input-location</i> ' <i>literal</i> '], [<i>conversion</i>],[<i>output-location</i>])][, FIELD =...] [, LABELS = <i>n</i>]
------------------	---------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

{IDENT|IDENTG}= (length,'name',input-location)

入力データ・セットにあるレコード集合の、最後のレコードを識別します。このパラメーターを使用して、同じ RECORD ステートメント上の FIELD パラメーターに基づいて編集する最後のレコードを識別します。区分データ・セットまたは PDSE を作成している場合、このパラメーターは、直前の MEMBER ステートメントで指名された区分データ・セットまたは PDSE メンバーに含める最後のレコードを識別します。RECORD ステートメントの後に、追加の RECORD または MEMBER ステートメントが続くことはなく、IDENT または IDENTG も処理する最後のレコードを識別します。

IDENT は、標準的な 1 バイト文字ストリングを識別するために使用します。

IDENTG は、2 バイト文字ストリングを識別するために使用します。

IDENT または IDENTG でコーディングできる値は以下のとおりです。

length

識別する名前の長さ (バイト単位) を指定します。ID の長さは、8 より大きくなってはなりません。

IDENTG に関しては、長さは偶数でなければなりません。

'name'

レコードのグループの最後の入力レコードを識別するリテラルを指定します。'name' は、一重のアポストロフィで囲まなければなりません。

IDENTG を使用している場合、'name' は 2 バイト文字ストリングでなければなりません。DBCS ストリングは、シフトアウト / シフトイン (SO/SI) 文字で囲まなければなりません。SO/SI 文字は、'name' によって指定されたリテラルの一部としては見なされず、length として数える必要はありません。IEBGENER は 'name' との一致を探すときに、シフトイン / シフトアウト文字を考慮しません。

'name' は 16 進で指定できます。これを行うには、'name' を name とコーディングします。したがって、特定の文字を生成するキーボードがない場合は、それらをその 16 進バージョンで指定することができます。シフトイン / シフトアウト文字の値は、順に X'0E'、X'0F' です。

'name' と一致するものがない場合は、入力データの残りが 1 つのレコード・グループと見なされ、続く RECORD および MEMBER ステートメントは無視されます。

input-location

入力レコードに識別名があるフィールドの開始位置を指定します。

input-location は、10 進の整数としてコーディングしなければなりません。

IDENT または IDENTG を指定しない場合、すべての入力データは、1 つのレコード・グループと見なされます。最初の RECORD および MEMBER ステートメントのみが IEBGENER によって使用されます。

FIELD=([length], [{input-location| 'literal'}], [conversion],[output-location])

フィールド処理と編集の情報を指定します。入力レコードにある指定したフィールドの内容のみが出力レコードにコピーされます。つまり、出力レコードにあるフィールドで、指定されていないものはすべて意味のないデータを持っているということです。

FIELD パラメーターの変数は定位置のものであることに注意してください。したがって、オプションのどれかをコーディングしない場合は、その変数に先行している関連のあるコンマをコーディングしなければなりません。

コーディングできる値は以下のとおりです。

length

入力フィールドまたは処理するリテラルの長さ (バイト単位) を指定します。 *length* が指定されていない場合は、長さが 80 と想定されます。リテラルを処理することになっている場合、長さに 40 またはそれ以下が指定されます。

input-location

処理するフィールドの開始位置を指定します。 *input-location* は、10 進の整数としてコーディングしなければなりません。 *input-location* が指定されていない場合は、デフォルト 1 になります。

'literal'

指定した出力位置に入れるリテラルを指定します (最大長は 40 バイト)。リテラルにアポストロフィがある場合は、それぞれのアポストロフィを二重のアポストロフィにしなければなりません。

リテラルを 16 進数で指定するには、 *X'literal'* をコーディングします。さらに 2 バイト文字セット・ストリングをリテラルとして指定することもできます。

conversion

このフィールドで実行する変換のタイプを示すコードを指定します。変換のタイプが指定されていない場合は、フィールドは変更されずに出力域に移されます。コーディングできる値は以下のとおりです。

CG

シフトアウト / シフトイン文字が除去されますが、 DBCS データの妥当性検査は行われないうように指定します。 DBCS=YES を GENERATE ステートメントで指定しなければなりません。

CV

DBCS データの妥当性検査が行われ、入力レコードに 2 バイト文字セットと同じように 1 バイト文字セットが含まれるように指定します。 DBCS=YES を GENERATE ステートメントで指定しなければなりません。

GC

DBCS データを囲むようにシフトアウト / シフトイン文字を挿入するように指定します。 DBCS=YES を GENERATE ステートメントで指定しなければなりません。

GV

DBCS データの妥当性検査が行われ、 DBCS データがシフトアウト / シフトイン文字によって囲まれないように指定します。 DBCS=YES を GENERATE ステートメントで指定しなければなりません。

HE

H セット BCDIC データが EBCDIC に変換されるように指定します。

PZ

パック 10 進データをアンパック 10 進データに変換することを指定します。下位数字および符号のアンパックの結果として、英字になることがあります。入力パック 10 進数フィールドのこの最大長は 16380 バイトです。

VC

DBCS データの妥当性検査が行われ、DBCS データを囲むためにシフトアウト / シフトイン文字を挿入するように指定します。DBCS=YES を GENERATE ステートメントで指定しなければなりません。

VG

DBCS データの妥当性検査が行われ、レコードからシフトアウト / シフトイン文字を除去するように指定します。DBCS=YES を GENERATE ステートメントで指定しなければなりません。

ZP

アンパック 10 進データがパック 10 進データに変換されるように指定します。

ZP パラメーターが指定されている場合、同じ場所での変換が実行されます。元のアンパック・フィールドは、新しいパック 10 進フィールドによって置換されます。したがって、ZP パラメーターは続くフィールドの参照から省略されなければなりません。このフィールドが元のアンパック・フォームに必要である場合は、ZP パラメーターを使用する前に参照しなければなりません。

conversion が FIELD パラメーターで指定されている場合は、それぞれの交換指定のために出力レコードの長さを計算できます。L が入力レコードの長さと同じであるときの計算は次のようになります。

- PZ (パック 10 進からアンパック 10 進) 指定の場合は、 $2L-1$ 。
- ZP (アンパック 10 進からパック 10 進) 指定の場合は、 $(L/2) + C$ 。L が奇数である場合は、C は $1/2$ であり、L が偶数である場合は、C は 1。
- HE (H セット BCDIC から EBCDIC) 指定の場合は L。
- DBCS 変換コードでは、シフトアウト / シフトイン文字は 1 バイトずつ計算します。それらを追加または削除する場合は、追加のバイトを計算しなければなりません。

output-location

出力レコードにあるこのフィールドの開始位置を指定します。*output-location* は、10 進の整数としてコーディングしなければなりません。

output-location が指定されていない場合は、位置がデフォルト 1 になります。

LABELS=*n*

オプション・パラメーターであり、ユーザー・ラベルとして扱われる SYSIN データ・セットにあるレコードの数を示します。数字 *n* (0 から 8 までの数字) は、RECORD ステートメントに続くラベル・レコードの正確な数を指定しな

ければなりません。このパラメーターが含まれている場合、DATA=INPUT は入力ストリームにあるこのパラメーターの前に LABELS ステートメント上でコーディングしなければなりません。

IEBGENER の例

以下の各例では、IEBGENER の使用方法の一部を示します。表 24 は IEBGENER の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 24. IEBGENER の例の一覧表

操作	データ・セット編成	装置	注釈	例
印刷	順次	ディスクおよびプリンター	データ・セットがプリンターにリスト表示されます。	1
変換	順次から区分	磁気テープおよびディスク	出力はブロック化されています。3 つのメンバーが作成されます。	2
マージ	順次から区分	ディスク	出力はブロック化されています。2 つのメンバーが既存のデータ・セットへマージされます。	3
COPY	順次	インストリームおよび磁気テープ	出力はブロック化されています。	4
コピーおよび再ブ ロック	順次	ディスクおよび磁気テープ	ブロック磁気テープ・コピーをディスクから作成します。明示的なバッファ要求。	5
コピー - 編集あ り	順次	磁気テープ	出力はブロック化されています。データ・セットが、1 つのレコード・グループとして編集されます。	6
コピー - 編集あ り	順次	z/OS UNIX ファイルからディスク	出力はブロック化されています。出力データ・セットに新しいレコード長が指定されます。2 つのレコード・グループが指定されます。	7
コピー - DBCS 妥当性検査あり	順次	ディスク	DBCS データをコピーする前に、その妥当性検査と編集が行われます。	8

各例で、実際の装置番号の代わりに **disk** または **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: 順次データ・セットを印刷する

この例では、順次データ・セットを印刷します。印刷出力は左寄せであり、印刷出力の各行には、80 バイト・レコードが 1 つずつ表示されます。

```
//PRINT    JOB    ...
//STEP1    EXEC  PGM=IEBGENER
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  DUMMY
//SYSUT1   DD  DSNAME=D80.DATA,DISP=SHR
//SYSUT2   DD  SYSOUT=A
```


各ジョブ制御ステートメントについての説明は以下のとおりです。

- **SYSIN DD** では、ダミー・データ・セットを定義しています。編集が行われないため、ユーティリティー制御ステートメントは必要ありません。
- **SYSUT1 DD** では、入力順次データ・セットを定義しています。
- **SYSUT2 DD** では、出力がシステム出力装置 (プリンター) に書き込まれることを指定しています。IEBGENER は LRECL と RECFM を SYSUT1 データ・セットからコピーし、システムは BLKSIZE を判別します。

例 2: 順次入力から区分データ・セットを作成する

この例では、区分データ・セット (3 つのメンバーから構成されている) を順次入力から作成します。

```
//TAPEDISK JOB ...
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=INSET,UNIT=tape,LABEL=(,SL),
//          DISP=(OLD,KEEP),VOLUME=SER=001234
//SYSUT2 DD DSN=NEWSET,UNIT=disk,DISP=(,KEEP),
//          VOLUME=SER=111112,SPACE=(TRK,(10,5,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000)
//SYSIN DD *
          GENERATE MAXNAME=3,MAXGPS=2
          MEMBER NAME=MEMBER1
GROUP1 RECORD IDENT=(8,'FIRSTMEM',1)
          MEMBER NAME=MEMBER2
GROUP2 RECORD IDENT=(8,'SECNDMEM',1)
          MEMBER NAME=MEMBER3
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、入力データ・セット (INSET) を定義しています。このデータ・セットは、テープ・ボリュームの 1 次データ・セットです。
- **SYSUT2 DD** では、出力区分データ・セット (NEWSET) を定義しています。このデータ・セットは、ディスク・ボリュームに配置されます。今後のデータ・セットの拡張のために、1 次スペースの 10 トラック、2 次スペースの 5 ブロック (それぞれ 256 バイト) が割り当てられています。データ・セットが必要とするスペースを削減するために、出力レコードはブロックされます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。ユーティリティー制御ステートメントは、順次入力データからメンバーを作成するために使用します。ステートメントは、編集を指定することはありません。
- **GENERATE** は、続く **MEMBER** ステートメントに 3 つのメンバー名が含まれることを示し、さらに **IDENT** パラメーターが、続く **RECORD** ステートメントに 2 度現れることを示します。
- 最初の **MEMBER** ステートメントは、メンバー名 (MEMBER1) を最初のメンバーに割り当てます。
- 最初の **RECORD** ステートメント (GROUP1) は、最初のメンバーに配置する最後のレコードを識別します。このレコード (FIRSTMEM) の名前は、入力レコードの最初の 8 つの位置に表示されます。
- 残りの **MEMBER** および **RECORD** ステートメントは、2 番目と 3 番目のメンバーを定義します。3 番目の **MEMBER** ステートメントには **RECORD** ステ

トメントが関連付けられていないため、入力ファイルの残りの部分が 3 番目のメンバーとしてロードされることに注意してください。

例 3: 順次入力を区分メンバーに変換する

この例では、順次入力を 2 つの区分メンバーに変換します。新しく作成されるメンバーは、既存の区分データ・セットにマージされます。入力データ・セットのユーザー・ラベルは、ユーザー・ラベルに渡されます。

```
//DISKTODK JOB ...
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=INSET,UNIT=disk,DISP=(OLD,KEEP),
// VOLUME=SER=111112,LABEL=(,SUL)
//SYSUT2 DD DSN=EXISTSET,UNIT=disk,DISP=(MOD,KEEP),
// VOLUME=SER=111113
// GENERATE MAXNAME=3,MAXGPS=1
// EXITS INHDR=ROUT1,INTLR=ROUT2
// MEMBER NAME=(MEMX,ALIASX)
GROUP1 RECORD IDENT=(8,'FIRSTMEM',1)
MEMBER NAME=MEMY
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、入力データ・セット (INSET) を定義しています。ディスク・ボリュームに入っている入力データ・セットには、標準的なユーザー・ラベルがあります。
- **SYSUT2 DD** では、出力区分データ・セット (EXISTSET) を定義しています。このジョブ・ステップで作成されるメンバーは、区分データ・セットにマージされます。
- **SYSIN DD** ステートメントは省略されています。GENERATE 行は // で始まっていないため、システムはその行の前に //SYSIN DD * 行があると想定します。**SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義しています。ユーティリティ制御ステートメントは、順次入力データからメンバーを作成するために使用します。ステートメントは、編集を指定することはありません。JCL ステートメントまたはジョブ・ストリームの終了は、入力ストリーム・データ・セットの終了を示すため、DD * データ・セットの最後の /* は無くても構いません。
- **GENERATE** は、最大数である 3 つの名前および別名が、続く **MEMBER** ステートメントにあり、1 つの **IDENT** パラメーターが、続く **RECORD** ステートメントに現れることを示します。
- **EXITS** では、ユーザー・ラベルを処理するユーザー・ルーチンを定義しています。
- 最初の **MEMBER** ステートメントは、メンバー名 (MEMX) および別名 (ALIASX) を最初のメンバーに割り当てます。
- **RECORD** ステートメントは、最初のメンバーに配置する最後のレコードを識別します。このレコード (FIRSTMEM) の名前は、入力レコードの最初の 8 つの位置に表示されます。
- 2 番目の **MEMBER** ステートメントは、メンバー名 (MEMY) を 2 番目のメンバーに割り当てます。入力データ・セットの残りの部分は、このメンバーに組み込まれます。

例 4: インストリーム入力、順次データ・セットをテープ・ボリュームにコピーする

この例では、インストリーム入力、順次データ・セットをテープ・ボリュームにコピーします。

```
//CDTOTAPE JOB ...
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD DSNAME=OUTSET,UNIT=tape,LABEL=(,SL),
// DISP=(,KEEP),VOLUME=SER=001234,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000)
//SYSUT1 DD *
(インストリーム・データ)
/*
```

各ジョブ制御ステートメントについての説明は以下のとおりです。

- **SYSIN DD** では、ダミー・データ・セットを定義しています。編集が行われないため、ユーティリティー制御ステートメントは必要ありません。
- **SYSUT2 DD** では、出力データ・セット **OUTSET** を定義しています。このデータ・セットは、IBM 標準ラベルを使用してテープ・ボリュームに書き込まれます。データ・セットは、最初の (または唯一の) データ・セットとしてボリュームに入ります。
- **SYSUT1 DD** では、実際には **JES SYSIN** データ・セットであるインストリーム・データを定義しています。データ・セットにステートメントは含まれていません。

例 5: 非ブロック化ディスク・ファイルから磁気テープにブロック化コピーを生成する

この例では、非ブロック化順次ディスク・ファイルから磁気テープにブロック化コピーを生成します。ディスク・データ・セットのブロック・サイズが比較的小さいため、明示的に要求されるバッファの数がデフォルト 5 よりも大きくなります。さらに、**SYSUT1** データ・セットの読み取りを **SYSUT2** データ・セットの書き込みにオーバーラップさせることを許可することにより、パフォーマンスが向上します。

```
//COPYJOB JOB
//STEP1 EXEC PGM=IEBGENER,REGION=318K
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSNAME=INPUT,UNIT=disk,
// DISP=OLD,VOL=SER=X13380,
// DCB=(BUFNO=20,RECFM=F,LRECL=2000,BLKSIZE=2000)
//SYSUT2 DD DSNAME=OUTPUT,UNIT=tape,DISP=(NEW,KEEP),
// DCB=(RECFM=FB,LRECL=2000,BLKSIZE=32000)
```

各ジョブ制御ステートメントについての説明は以下のとおりです。

- **EXEC** ステートメントは、**IEBGENER** プログラムを指名し、必須の仮想記憶領域サイズを指定します。(領域サイズの計算については、88 ページの表 15 に説明があります。)
- **SYSIN DD** ステートメントはダミーであるため、編集は実行されません。
- **SYSUT1 DD** ステートメントでは、入力ディスク・ファイルを定義しています。通常は、**DCB RECFM**、**LRECL**、および **BLKSIZE** 情報が **VTOC** のデータ・セ

ット・ラベルに入っているため、これらの情報は、既存のディスク・ファイルの DD ステートメントに指定すべきではありません。ただし、この例では、出力データ・セットとの対比を示すために、そのような情報を指定しています。データ・セットがカタログである場合は、装置およびボリューム通し番号情報は省略できます。DCB 情報は BUFNO=20 を指定して、ディスクのローテーションごとに 20 ブロックまでを読み取ることを許可します。このときに、ディスク・トラックが、それだけのブロックを保持していると想定されます。

- SYSUT2 DD ステートメントでは、出力磁気テープ・データ・セットを識別し、32,000 バイトのブロック・サイズを指定します。入力のペースを保持するのに十分な、5 つのバッファのデフォルトは 8 です。

例 6: ラベルのある順次入力データ・セットの編集およびコピー

この例では、順次入力データ・セットを編集してコピーします。

```
//TAPETAPE JOB ...
//STEP1 EXEC PGM=IEBGNER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=OLDSET,UNIT=tape,DISP=(OLD,KEEP),
//          VOLUME=SER=001234,LABEL=(3,SUL)
//SYSUT2 DD DSN=NEWSET,UNIT=tape,DISP=(NEW,PASS),
//          DCB=(RECFM=FB,LRECL=80),
//          VOLUME=SER=001235,LABEL=(,SUL)
//SYSIN DD *
          GENERATE MAXFLDS=3,MAXLITS=11
          RECORD FIELD=(10,'*****',,1),
          FIELD=(5,1,HE,11),FIELD=(1,'=',,16)
          EXITS INHDR=ROUT1,OUTTLR=ROUT2
          LABELS DATA=INPUT
          RECORD LABELS=2
```

(最初のヘッダー・ラベル・レコード)
(2 番目のヘッダー・ラベル・レコード)

```
RECORD LABELS=2
```

(最初のトレーラー・ラベル・レコード)
(2 番目のトレーラー・ラベル・レコード)

```
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、順次入力データ・セット (OLDSET) を定義しています。このデータ・セットは、元は 3 番目のデータ・セットとしてテープ・ボリュームに書き込まれていました。
- SYSUT2 DD では、順次出力データ・セット (NEWSET) を定義しています。このデータ・セットは、テープ・ボリュームの 1 次データ・セットとして書き込まれます。データ・セットが必要とするスペースを削減したり、データ・セットが順次に参照されているときに必要なアクセス時間を減らしたりするために、出力レコードはブロックされます。BLKSIZE パラメーターが省略されているので、システム・プログラマーが、別途、デフォルトをセットしない限り、システムは、32 760 バイト以下の最適値を計算します。125 ページの『EXEC ステートメント』を参照してください。データ・セットは、後続のジョブ・ステップに渡されます。LABEL=(,SUL) は、作成されたユーザー・ラベルのために必須です。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。

- GENERATE は、最大数である 3 つの FIELD フィールドが、続く RECORD ステートメントに含まれており、最大数である 11 リテラル文字が、続く FIELD パラメーターに含まれていることを示します。
- 最初の RECORD ステートメントでは、アスタリスクが位置 1~10 に配置され、入力レコードの 1~5 が H セット BCDIC から EBCDIC モードに変換されて位置 11~15 に移され、等号が位置 16 に配置される、という具合に編集を制御しています。
- EXITS では、SYSUT1 がオープンされ、SYSUT2 がクローズしたときに、指定したユーザー・ルーチンに制御が必要なことを示しています。
- LABELS では、ラベルが入力ストリームに組み込まれていることを示しています。
- 2 番目の RECORD ステートメントでは、SYSIN の次の 2 つのレコードが、SYSUT2 のユーザー・ヘッダー・ラベルとして書き出される必要があることを示しています。
- 3 番目の RECORD ステートメントでは、SYSIN の次の 2 つのレコードが、SYSUT2 のユーザー・トレーラー・ラベルとして書き込まれる必要があることを示しています。

この例では、RECORD LABELS ステートメント、LABELS ステートメント、および EXITS ステートメントの関係を示しています。IEBGENER は、SYSUT2 のクローズ時に、制御をシステムに戻す前に、最初および 2 番目のラベル・トレーラーをユーザー・ラベルとして書き込もうとします。ユーザー・ルーチン ROUT2 は、これらのレコードを検討して、必要に応じて変更できます。

例 7: 順次 z/OS UNIX ファイルを順次データ・セットに編集およびコピーする

この例では、z/OS UNIX システム・サービス (z/OS UNIX) ファイルを編集してコピーします。出力データ・セットの論理レコード長が、入力データ・セットの論理レコード長よりも短くなっています。

```
//DISKDISK JOB ...
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD PATH='/dist3/stor44/sales.mon',FILEDATA=TEXT,PATHOPTS=ORDONLY,
//          LRECL=100,BLKSIZE=1000,RECFM=FB
//SYSUT2 DD DSNAME=NEWSET,UNIT=disk,DISP=(NEW,KEEP),
//          VOLUME=SER=111113,DCB=(RECFM=FB,LRECL=80,
//          BLKSIZE=640),SPACE=(TRK,(20,10))
//SYSIN DD *
          GENERATE MAXFLDS=4,MAXGPS=1
          EXITS IOERROR=ERRORRT
GRP1 RECORD IDENT=(8,'FIRSTGRP',1),FIELD=(21,80,,60),FIELD=(59,1,,1)
GRP2 RECORD FIELD=(11,90,,70),FIELD=(69,1,,1)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、入力ファイルを定義しています。この名前は /dist3/stor44/sales.mon です。これには、100 バイト・レコードのテキストが含まれています。レコード区切り文字については、ここでは記述していません。ファイルは、ネットワーク・ファイル・システム (NFS) を介して利用可能な非 System/390 システムにある場合があります。

- SYSUT2 DD では、出力データ・セット (NEWSET) を定義しています。1 次ストレージの 20 トラック、および 2 次ストレージの 10 トラックは、ディスク・ボリュームのデータ・セットに割り振られます。出力レコードの論理レコード長は 80 バイトであり、出力はブロックされます。
- SYSDIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- GENERATE は、最大数である 4 つの名 FIELD パラメーターが、続く RECORD ステートメントにあり、1 つの IDENT パラメーターが、続く RECORD ステートメントに現れることを示します。
- EXITS は、入出力ハンドルを扱うユーザー・ルーチンを識別します。

図 22 は、順次入力データ・セットを編集してコピーする方法を示したものです。

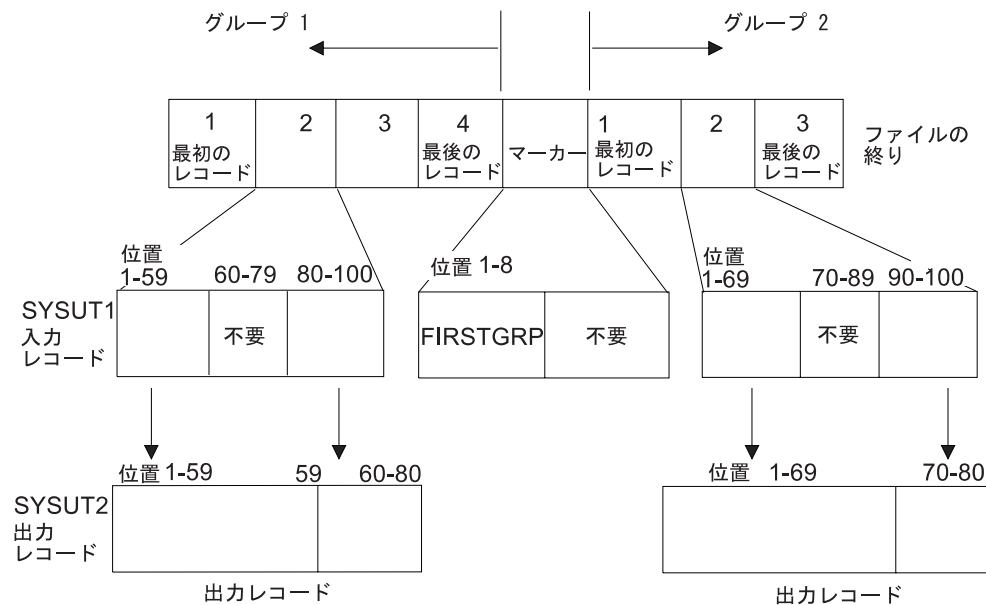


図 22. 順次データ・セットを編集してコピーする方法

- 最初の RECORD ステートメント (GRP1) は、最初のレコード・グループの編集を制御します。入力レコードの最初の位置 8 に現れる FIRSTGRP は、レコードの最初のグループにある最後のレコードとして定義されます。それぞれの入力レコードの位置 80~100 にあるデータは、それぞれの対応する出力レコードの位置 60~80 に移されます。(この例では、最初のレコード・グループにある入力レコードの位置 60~79 がなくなるので、論理レコード長は 20 バイト短くなります。)それぞれの入力レコードの残りの位置にあるデータは、2 番目の FIELD パラメーターで指定したとおり、直接出力レコードに転送されます。
- 2 番目の RECORD ステートメント (GRP2) では、入力レコードの残りが、2 番目のレコード・グループとして処理されることを示しています。それぞれの入力レコードの位置 90~100 にあるデータは、出力レコードの位置 70~80 に移されます。(この例では、グループ 2 の入力レコードの位置 70~89 がなくなるので、論理レコード長は 20 バイト短くなります。)それぞれの入力レコードの残りの位置にあるデータは、2 番目の FIELD パラメーターで指定したとおり、直接出力レコードに転送されます。

例 8: 2 バイト文字セット・データを編集する

この例では、2 バイト文字セット・データを含む、編集されたデータ・セットを作成します。シフトアウト / シフトイン文字 (< および >) が、DBCS スtringを囲むように挿入されます。

```
//DBLBYTE JOB ...
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=INPUT,DISP=(OLD,KEEP),UNIT=disk
//SYSUT2 DD DSN=OUTPUT,UNIT=disk,DISP=(NEW,CATLG),
//          DCB=(LRECL=80,BLKSIZE=3200,RECFM=FB),SPACE=(TRK,(1,1))
//SYSIN DD *
          GENERATE MAXFLDS=4,MAXLITS=9,DBCS=YES
          RECORD FIELD=(20,1,,1),FIELD=(16,33,VC,21),
                   FIELD=(30,50,VC,39),FIELD=(9,'*****',,72)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、ディスク・ボリューム上にある入力データ・セット **INPUT** を定義しています。
- **SYSUT2 DD** では、ディスク・ボリューム上にある出力データ・セット **OUTPUT** を定義しています。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **GENERATE** は、最大数である 4 つの **FIELD** パラメーターおよび 9 つのリテラル文字が、続く **RECORD** ステートメントに表示され、入力データ・セットに **DBCS** データが含まれていることを示します。
- **RECORD** では、入力レコードが出力データ・セットに配置される前に、どのようにその入力レコードを編集するかを指定しています。最初の **FIELD** パラメーターは、入力レコードの最初の 20 位置 (バイト) が出力レコードの最初の 20 位置に配置されることを示しています。
- 2 番目の **FIELD** パラメーターは、位置 33 から 48 にあるデータが必ず有効な **DBCS** データであることが検査され、さらにシフトアウト / シフトイン文字がこのフィールドの回りに挿入されることを示しています。 **DBCS** データを有効にするには、2 バイト文字のそれぞれのバイトに **X'41'** および **X'FE'** の間の 16 進値があるか、または 2 バイト文字が **DBCS** スペース (**X'4040'**) でなければなりません。検査および挿入が完了したら、このフィールドは、位置 21 で始まる出力レコードにコピーされます。
- 3 番目の **FIELD** パラメーターは、入力レコードの位置 50 から始まる 30 バイト・フィールドで操作を行います。このフィールドの **DBCS** データが有効であることが検査され、シフトアウト / シフトイン文字がフィールドの回りに挿入されます。結果フィールドは、位置 39 から始まる出力レコードにコピーされます。

FIELD パラメーターで出力位置を指定するときに、シフトイン / シフトアウト文字が使用する追加位置を考慮しなければならないことに注意してください。たとえば、位置 21 から始まる 8 文字 (16 バイト) **DBCS** スtringは、位置 36 では終了せず、38 で終了します。シフトイン/シフトアウト文字は 1 バイト文字であり、このペアで 2 つの位置を使用します。

最後の **FIELD** パラメーターは、アスタリスクのある出力レコードの最後の位置を消去します。

第 7 章 IEBIMAGE (プリンター・イメージの作成) プログラム

IEBIMAGE は、IBM 3800 印刷サブシステムおよび IBM 4248 印刷装置の以下のようなタイプのモジュールを作成および維持して、さらにそれらをライブラリーに保管するためのデータ・セット・ユーティリティーです。

- 各行ごとに垂直行送りおよび 12 チャンネル・コードのどれか 1 つを指定する、3800 および 4248 の用紙制御バッファ・モジュール。
- 出力データ・セットの指定されているコピーのすべてのページに印刷されるデータを指定する、3800 のコピー変更モジュール。
- 入力データを印刷可能文字に変換し、関連している文字セットと図形文字変更モジュールを識別する、3800 の文字配列テーブル・モジュール。
- ユーザーが設計した文字の走査パターン、または IBM 提供のモジュールからの文字を含む、3800 の図形文字変更モジュール。
- ユーザーが定義した文字セットの走査パターンまたは IBM 提供の文字セットを含む、3800 のライブラリー文字セット・モジュール。

IEBIMAGE プログラムは、3800 モデル 1 およびモデル 3 印刷装置で使用するために必要なすべてのモジュールを作成し、維持します。プログラム・デフォルトは、これらのモジュールを 3800 モデル 1 フォーマットで構築するためのものとなっています。しかし、IEBIMAGE ユーティリティー制御ステートメントを使用して 3800 モデル 3 互換性を指定することができます。

IEBIMAGE を使用して 4248 印刷装置の FCB モジュールを作成し、維持することもできます。これらのモジュールは、3262 モデル 5 印刷装置と互換性がありますが、3262 モデル 5 は、プリンター可変速度または 4248 の水平コピー機能はサポートしません。特に指摘がない限りは、この章で 4248 印刷装置を参照している部分に、3262 モデル 5 を置き換えることができます。

他のタイプのプリンター・イメージを作成するための情報については、*z/OS DFSMSdfp* 拡張サービスを参照してください。

SYS1.IMAGELIB データ・セットのストレージ要件

SYS1.IMAGELIB のトラックのための補助記憶装置の要件を次に示します。

$$\text{Number of tracks} = (A + B) / T$$

各項目の意味は以下のとおりです。

- A** 1403 UCS イメージ、3211 UCS イメージ、3211 FCB イメージ、3525 データ保護イメージ、3890™ SCI プログラム、3800 FCB モジュール、4248 FCB イメージ、3262 モデル 5 FCB イメージ、3800 文字配列テーブルおよび 3800 ライブラリー文字セットの数です (IBM によって提供されたイメージまたはモジュールを含む)。

IBM は、12 個の 1403 UCS イメージ、5 個の 3211 UCS イメージ、4 個の 3211 FCB イメージ、1 個の 3800 FCB イメージ、1 個の 4245

UCS イメージ・テーブル、1 個の 4248 UCS イメージ・テーブル、または 84 個の 3800 文字配列テーブル、20 個の 3800 モデル 1 ライブラリー文字データ・セット、20 個の 3800 モデル 3、6、および 8 ライブラリー文字セット、および図形文字変更モジュールを提供します。

制約:IBM は、4245 または 4248 UCS イメージを SYS1.IMAGELIB に提供することはありません。4245 および 4248 印刷装置は電源を入れるときに、それ自体の UCS イメージを UCS バッファーに入れます。IBM は、4245 および 4248 FCB イメージを提供しており、これらを使用できます。プリンター提供の UCS または FCB イメージの詳細については、*z/OS DFSMSdfp 拡張サービス* を参照してください。

- B** 各 3800 図形文字変更モジュールおよびライブラリー文字セット・モジュール、各 3800 コピー変更モジュール、4245 UCS イメージ・テーブル、4248 UCS イメージ・テーブル、および約 600 バイトより多い各 3890 SCI プログラムに対して次のようになります。

(V + 600) / 1500

V 各モジュールの仮想記憶域要件 (バイト数) です。

World Trade National Use Graphics を含む、IBM 提供の 3800 図形文字変更モジュールの仮想記憶域は、モデル 1 の場合は 32420、モデル 3、6、8 の場合は 55952 です。モデル 1 のために設定されている IBM 提供の 3800 ライブラリー文字セットの仮想記憶域は 4680 バイトであり、モデル 3、6、8 の場合は 8064 バイトです。

- T** トラックごとのメンバーの概数であり、ボリュームのタイプによって異なります。ロード・モジュールのオーバーヘッド・バイトによって、80 バイト・モジュールと 400 バイト・モジュールのスペース要件の差は小さくなります。

これらの定数は、平均メンバーが 8 ブロック (ファイル・マークを含む) で、データ長の合計が 800 バイトであることを前提としています。たとえば、ブロック・オーバーヘッドが 523 バイトある 3380 では、想定される平均は 4984 バイトになります。各メンバーの平均メンバー・データ長およびブロックの平均数が異なる可能性がある場合、これらの定数は、トラックごとの実際の数を反映しなければなりません。

トラックごとのメンバーの数を判別するには、ブロック・オーバーヘッドを含むメンバー長の平均を、装置のトラック容量に分割します (表 25 を参照してください)。(DASD のトラック容量については、*z/OS DFSMS Macro Instructions for Data Sets* に説明があります。)

表 25. 各装置のトラック (T) ごとのメンバー

T	装置タイプ
17	3380、すべてのモデル
20	3390、すべてのモデル
16	9345、すべてのモデル

結果である **(A + B) / T** がトラック要件になります。

SYS1.IMAGELIB のディレクトリー・ブロックの数は、次の計算で出されます。

Number of directory blocks = (A + C + D) / 6

各項目の意味は以下のとおりです。

- A** トラック要件計算の **A** と同じ値です。
- C** トラック要件を計算するときに **B** を計算するために使用するモジュールの数です。
- D** 別名の数です。IBM 提供の 1403 UCS イメージには 4 つの別名があり、IBM 提供の 3211 UCS イメージには 6 つの別名があります。これらの別名を使用しない場合は、システムがインストールされた後に消去することができます。

SYS1.IMAGELIB データ・セットの維持

通常は、SYS1.IMAGELIB を維持するために、いくつかのプログラムを IEBIMAGE と併用します。たとえば、モジュールを名前変更したり削除したり、データ・セット全体の内容を圧縮したりリスト表示したりすることが必要になるかもしれません。そのような場合は PDF、DFSMSdss、IEBCOPY、IEBPTCH、IEHLIST、IEHPROGM などのプログラムを使用して、SYS1.IMAGELIB を維持しなければなりません。AMASPZAP プログラムも、診断のために使用できます。このことについては、*z/OS MVS 診断: ツールと保守援助プログラム* に説明があります。

維持のために IEBIMAGE 以外のプログラムを使用する場合、モジュール名全体を指定しなければなりません。モジュールの完全名は、4 文字の接頭部と、その後続く、ユーザーが割り当てた 1~4 文字の名前によって構成されています。したがって、5~8 文字の次のようなフォームのメンバー名になります。

FCB2xxxx	3203、3211、3262 モデル 5、4248、または 4245 印刷装置で使用できる FCB モジュール。4248 は、同様に 3203、3211、3262 モデル 5、または 4245 印刷装置で機能する FCB を受け入れます。さらに、FCB2 モジュールは IEBIMAGE を使用して作成することはできませんが、IEBIMAGE は FCB4 モジュールを作成するための入力として FCB2 モジュールを使用できます。FCB2 モジュールの維持および作成のための情報については、 <i>z/OS DFSMSdfp 拡張サービス</i> を参照してください。
FCB3xxxx	3800 FCB モジュール
FCB4xxxx	4248 または 3262 モデル 5 印刷装置で使用できる FCB モジュール。
MOD1xxxx	3800 コピー変更モジュール
XTB1xxxx	3800 文字配列テーブル・モジュール
GRAFxxxx	3800 モデル 1 の図形文字変更モジュール
GRF2xxxx	3800 モデル 3 の図形文字変更モジュール
LCS1nn	3800 モデル 1 のライブラリー文字セット・モジュール
LCS2nn	3800 モデル 3 のライブラリー文字セット・モジュール

各項目の意味は以下のとおりです。

xxxx

モジュールの 1~4 文字のユーザー割り当て名です。

nn モジュールの 2 文字のユーザー割り当て ID です。

別名は IEBIMAGE によってサポートされていないため、別名を使用するときは注意が必要です。たとえば、別名を指定してモジュールを変更する場合、別名が新しいモジュールのメイン名となり、古いモジュールは、別名を介してアクセスできなくなります。元のメイン名を使用するとアクセスできます。

モジュールの一般構造

それぞれのモジュールには、データの前に 8 バイトの見出し情報があります。図 23 は、3800 印刷サブシステムの一般的なモジュール・ヘッダーを示したものです。

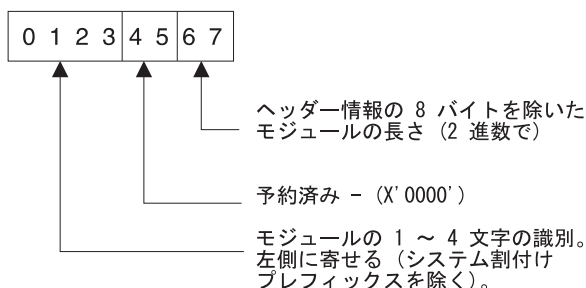


図 23. 3800 の一般的なモジュール・ヘッダー

153 ページの図 25 は、4248 印刷装置の FCB モジュールのヘッダー情報とモジュール・フォーマットを示したものです。

SETPRT マクロ命令は、名前を使用して次のことを行えます。

- イメージ・ライブラリーのモジュールを識別すること。
- 今後の要求の最適化のために名前を保管すること。

SETPRT マクロ命令は、長さを使用して次のことを行えます。

- モジュールのために十分なストレージを得ること。
- データをプリンターにロードするためのチャンネル・プログラムを作成すること。

モジュールの命名規則

IEBIMAGE ユーティリティによってライブラリーに配置された各モジュールには、システムによって割り当てられた 4 文字の接頭部が、その名前の先頭に付きます。これらの接頭部については、149 に説明があります。

作成したモジュールに 1~4 文字の ID (名前) を割り当てるには、モジュールを作成するために使用する操作グループで NAME 制御ステートメントを使用します。モジュールがライブラリー文字セットである場合、それに割り当てられている ID は、正確に 2 文字でなければなりません。これらの文字のそれぞれは、0~9 および A~F の範囲内のものでなければなりません。2 番目の文字は、奇数の 16 進数字を表していなければなりません。しかし、X'7F' と X'FF' の組み合わせは使用で

きません。ライブラリー文字セット・モジュール以外の場合、この ID は、ロードするモジュールを識別するために、JCL、SETPRT マクロ指示、または文字配列テーブルで使用されます。

IEBIMAGE は 1~4 文字の名前か、または接頭部に追加される 2 文字の ID (接尾部) のみを参照するものの、他のユーティリティーを使用するときには完全名を使用しなければなりません (IEBPTPCH または IEHPROGM など)。

IEBIMAGE の使用

用紙制御バッファ・モジュールの作成

用紙制御バッファ (FCB) モジュールは可変長であり、そこには垂直行送り情報 (3800 モデル 1 の 1 インチごとに 6、8、または 12 行、4248 の 1 インチごとに 6 または 8 行、3800 モデル 3 の 1 インチごとに 6、8、10、または 12 行) が含まれています。FCB モジュールは、行ごとの 12 の紙送り制御チャンネル・コードの 1 つを識別することもできます。4248 印刷装置では、モジュールに、水平コピー機能およびプリンター速度の情報も含まれています。

FCB モジュールは、FCB および NAME ユーティリティー制御ステートメントを使用して、イメージ・ライブラリーに作成され、保管されます。4248 FCB モジュールでは、INCLUDE および OPTION ステートメントをコーディング化して、既存の FCB モジュール (接頭部 FCB2 または FCB4) をモデルとして使用するよう指定できます。

3800 では、IBM はデフォルトの FCB イメージを SYS1.IMAGELIB に提供します。これは、FCB3STD1 と呼ばれます。4248 では、最後にロードされた FCB イメージは、電源をオンにするときにプリンターによって再びロードされますが、IBM は、4248 以外の印刷装置によっても使用可能な 2 つの FCB イメージを提供します。3262 モデル 5 でも、デフォルトの FCB イメージが提供されます。

3800 FCB モジュールの構造

ヘッダー情報に続く FCB データは、用紙の物理行ごとの、一連の 1 バイト行の制御コードです。これらのバイトは用紙の長さによって 18~144 になります。

それぞれのバイトは、ビット・パターンであり、これらは、用紙の位置決めのための 12 チャンネル・コードの 1 つ、および垂直行送りの 4 行 / インチ・コードの 1 つを説明しています。152 ページの図 24 は、3800 FCB モジュールの構造を示したものです。

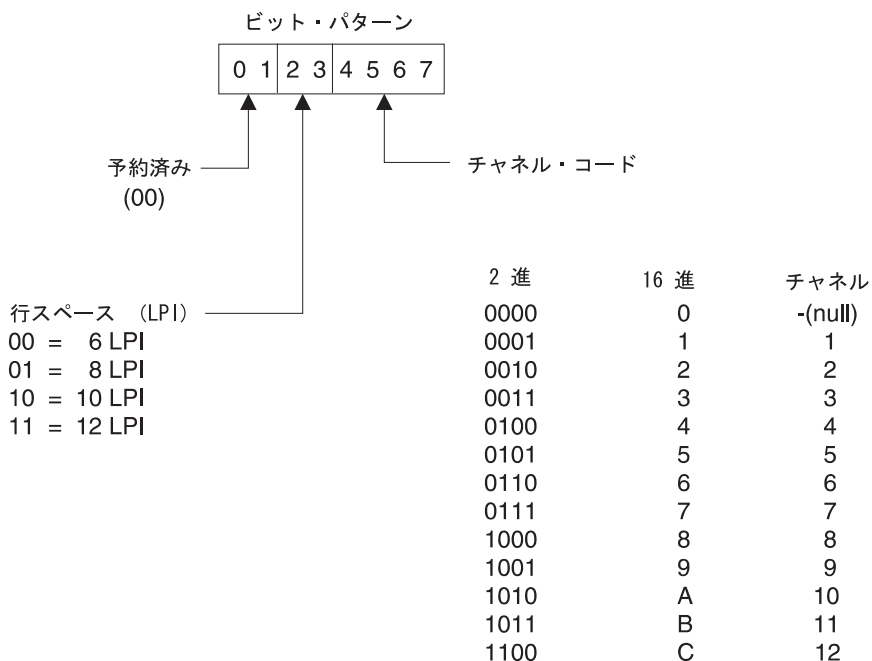


図 24. 3800 FCB モジュールの構造

- 各ページの上部と下部の 1/2 インチは印刷不能であり、これらの位置に対応するバイトにチャンネル・コードが入ってはいけません。上部および下部 1/2 インチのために、3 バイトの 2 進ゼロが IEBIMAGE ユーティリティによって提供されます。
- モジュールで定義されている行の合計数が、用紙の長さと同様であればなりません。定義されている印刷可能行は、上部から 1/2 インチ下のところで始まり、用紙の下部から 1/2 インチのところで停止させなければなりません。

4248 FCB モジュールの構造

ヘッダー情報に続く FCB データは、少なくとも 5 バイトによって構成されています。つまり、フラグ・バイト (X'7E')、制御バイト (水平コピー機能およびプリンター速度に関する情報を含む)、オフセット・バイト、1 つまたは複数の FCB データ・バイト (用紙の物理行ごとの 3800 データ・バイトと同様)、用紙終わりバイト (X'FE') です。153 ページの図 25 は、4248 FCB モジュールの構造を示したものです。

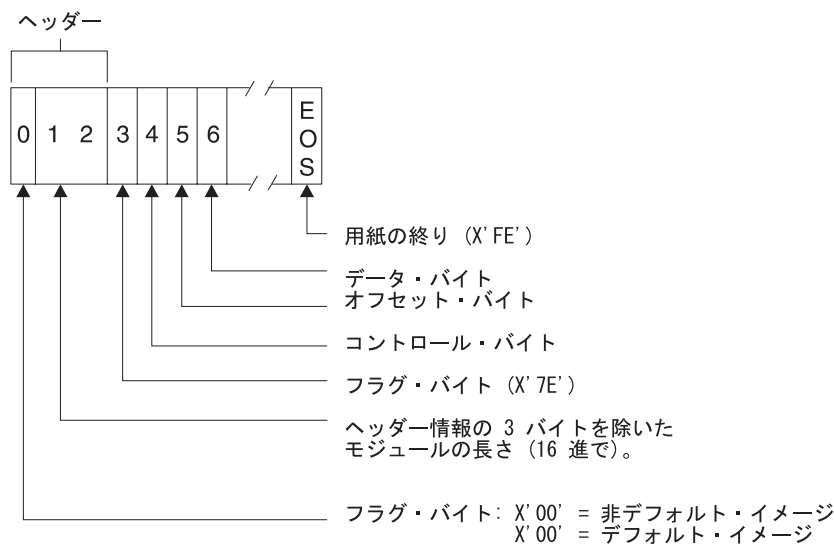


図 25. 4248 FCB モジュールの構造

制御バイトは、水平コピー機能がアクティブかどうかを説明し、FCB がバッファにロードされるときに、どの印刷速度が使用されるかを記述するビット・パターンです。図 26 は、制御バイトの構造を示したものです。

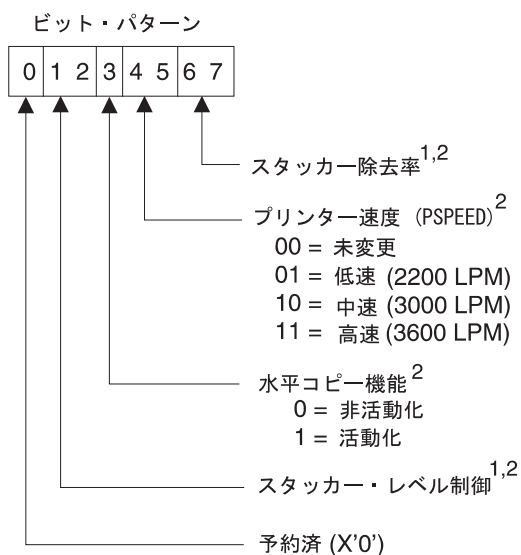


図 26. 4248 FCB モジュールの制御バイト

図 26 の注

- 1 IEBIMAGE は、これらのビットをゼロに設定します。スタッカー除去比率ビットおよびスタッカー・レベル制御ビットの詳細については、それぞれの印刷装置のハードウェア・マニュアルを参照してください。
- 2 モジュールが 3262 モデル 5 印刷装置によって使用されている場合、これらのビットは無視されます。

オフセット・バイトは制御バイトに続いており、これはゼロまたは水平コピー (2~168) の印刷位置のどちらかに設定されます。

データ・バイトは、3800 印刷サブシステムのために生成されたビット・パターンと同様のビット・パターンです。それぞれのデータ・バイトは、垂直用紙の位置決めのための 12 チャンネル・コードの 1 つ、および垂直行送りの許可されている行 / インチ・コードの 1 つを記述しています。図 27 は、データ・バイトの構造を示したものです。

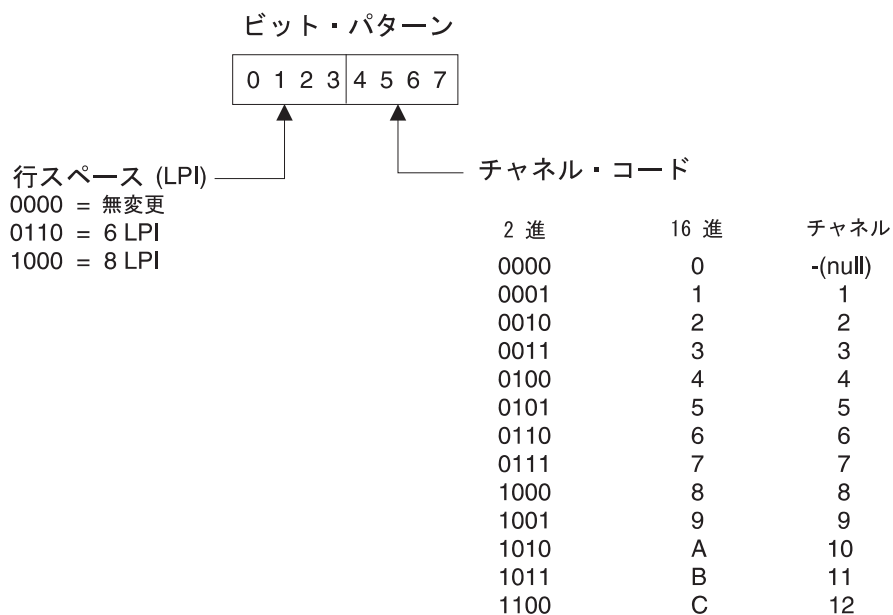


図 27. 4248 FCB モジュールのデータ・バイト

モジュールで定義されている行の合計数が、用紙の長さと同しくなければなりません。

FCB モジュールのリスト表示

155 ページの図 28 は、IEBIMAGE による 3800 FCB モジュールのリスト表示を示したものです。図の後の注では、図の中の四角で囲まれた番号の部分について説明しています。

4248 FCB モジュールでは、IEBIMAGE のリスト表示にも、水平コピー機能、プリンター速度設定、およびデフォルトの設定が含まれます。

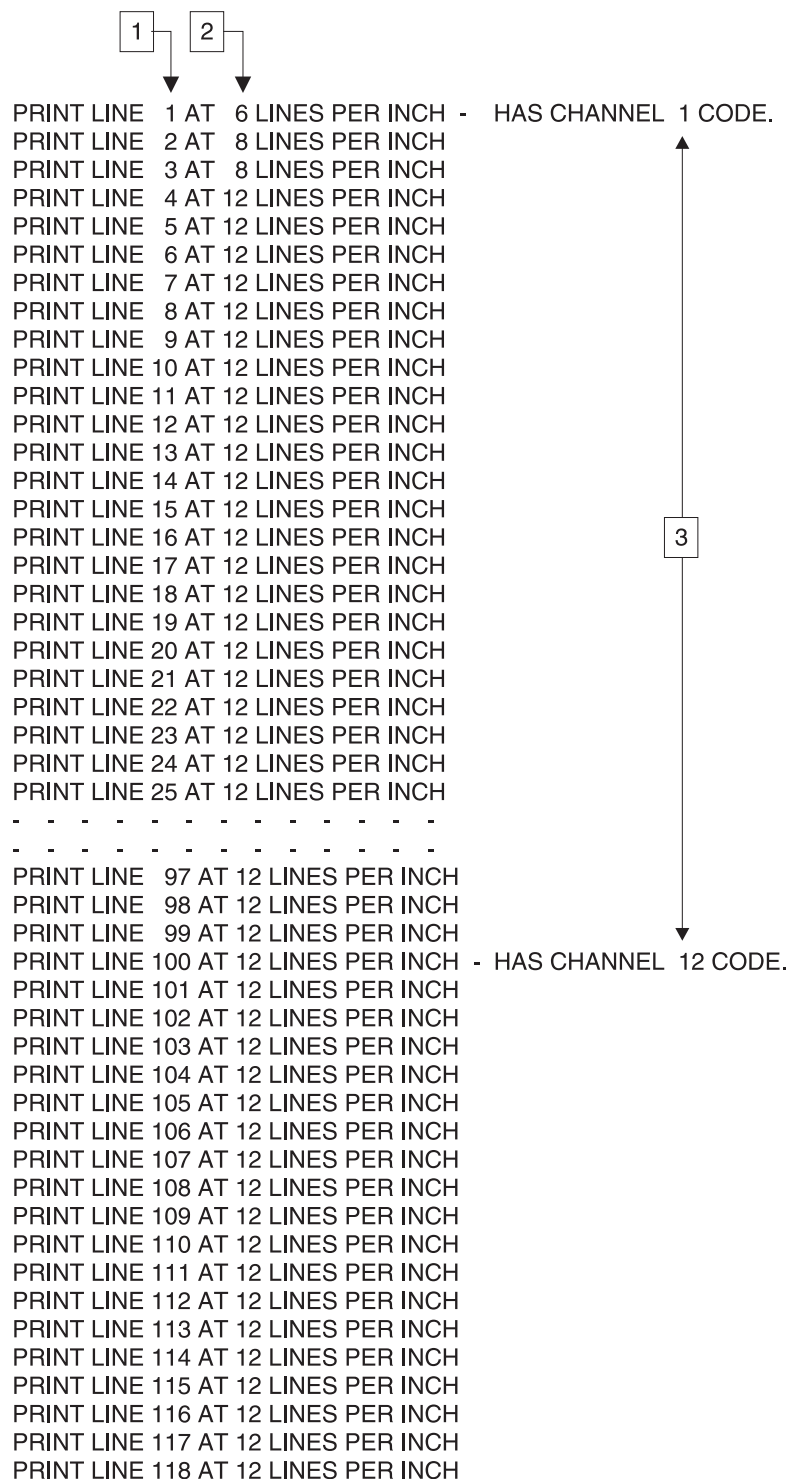


図 28. IEBIMAGE による用紙制御バッファ・モジュールのリスト表示

図 28 の注

1. 行番号です。用紙の各行が、このようにリスト表示されます。
2. 垂直行送り (行 / インチ) です。
3. チャンネル・コードであり、チャンネル・コードを含む行ごとに印刷されます。

コピー変更モジュールの作成

3800 コピー変更モジュールには、出力データ・セットの一部またはすべてのコピーを修正するための、事前定義データが含まれています。モジュールのセグメントには、事前定義のテキスト、出力データ・セットの各ページにおけるそのテキストの位置、およびテキストが適用されるコピー（複数も可）が含まれています。

INCLUDE、OPTION、COPYMOD、および NAME ユーティリティー制御ステートメントを使用することにより、コピー変更モジュールがイメージ・ライブラリーに作成され、保管されます。

INCLUDE ステートメントは、新しく作成されるモジュールの基礎としてコピーされ、使用されるモジュールを識別します。OPTION ステートメントに OVERRUN を指定すると、ジョブに適用できない垂直行送りの行オーバーラン条件メッセージが印刷されなくなります。DEVICE パラメーターが指定されている OPTION ステートメントは、3800 モデル 3 互換モード処理を指定しています。COPYMOD ステートメントは、新しいモジュールのセグメントの 1 つの内容を記述するために使用されます。NAME ステートメントは、新しいモジュールを識別し、さらにこれが新しいものか、または既存のモジュールを同じ名前置き換えるのかを示します。

COPYMOD モジュールの構造

ヘッダー情報に続くコピー変更データは、一連のセグメントになっています。各セグメントは可変長です。図 29 は、各コンポーネントを示したものです。

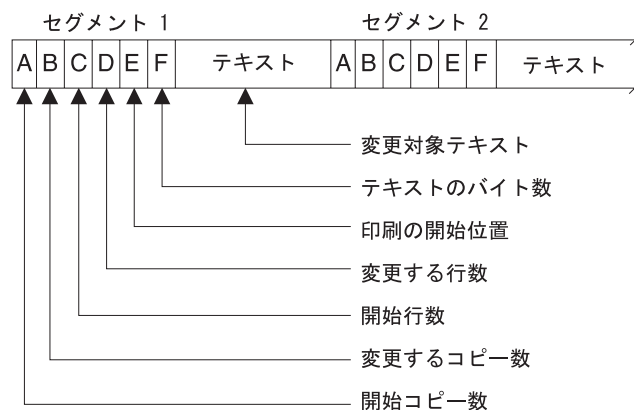


図 29. コピー変更モジュールの構造

A、B、C、D、E、F は、それぞれ 1 バイト・フィールドです。

- モジュールに複数のセグメントが含まれている場合、開始コピー番号は、直前のセグメントの開始コピー番号以上の値になっている必要があります。
- テキストにある同じ文字のストリングはすべて、3 バイトに圧縮されます。最初のそのようなバイトは X'FF'、2 番目のバイトは圧縮文字の数、3 番目のバイトは文字のデータ・コードです。
- モジュールのサイズは、データが 8192 バイト、ヘッダー情報が 8 バイトに制限されています。

COPYMOD モジュールのリスト表示

図 30 は、コピー変更モジュールの 3 つのセグメントのリスト表示を示したものです。このリストでは、変更するテキストの位置のみを示しています。テキスト自体を印刷するには、IEBTPCH ユーティリティ・プログラムまたは AMASPZAP 保守援助機能を使用できます。図の後の注では、四角で囲まれた番号の項目について説明しています。

セグメント	初期のコピー番号	コピー数	初期の行番号	行数	初期の印刷位置	文字数
1	1	4	58	1	35	18
2	2	1	1	1	50	23
3	2	1	34	3	75	10

図 30 の注に示すように、各セグメントの初期行番号と行数は、初期印刷位置と文字数と関連しています。また、図中の四角で囲まれた番号 1, 2, 3, 4, 5 は、それぞれ特定の行番号やコピー番号を指しています。

図 30. IEBIMAGE によるコピー変更モジュールの 3 つのセグメントのリスト表示

図 30 の注

この例の各注は、モジュールの 3 番目のセグメントに関するものです。

1. SYS1.IMAGELIB データ・セットのディレクトリーに存在する、コピー変更モジュールの名前 (システムが割り当てた 4 バイト接頭部も含む)。
2. 変更セグメントのセグメント番号。
3. このセグメントは、出力データ・セットの 2 番目のコピーにのみ適用されます。
4. セグメントのテキストは、34、35、36 行目に配置されています。
5. それぞれの行のテキストは、75 番目の文字から始まっており、10 文字のスペースを使用します。

文字配列テーブル・モジュールの作成

3800 文字配列テーブル・モジュールは固定長であり、3 つのセクションによって構成されています。

- システム制御情報。モジュールの名前と長さを含んでいます。
- 変換テーブル。8 ビットのデータ・コード (X'00'~X'FF') に対応する、256 の 1 バイト変換テーブルを含んでいます。変換テーブル項目は、4 つの書込可能文字生成モジュール (WCGM) のどれか 1 つに、64 文字位置のいずれかを認識することができます。ただし、4 番目に位置している WCGM (WCGM 3) は例外です。これは、X'FF' によってアドレッシングされます。コード X'FF' は、印刷不能文字を示すために予約されています。使用不能 8 ビット・データ・コードを変換しようとした結果として、X'FF' の項目が検出されるときに、プリンターはブランクを印刷し、データ・チェック標識をオンに設定します。ただし、ブロック・データ・チェック・オプションが有効になっている場合は例外です。

- ID。文字配列テーブルと関連している、文字セットと図形文字変更モジュールを識別します。

3800 モデル 1 または モデル 3 では、文字セット ID が偶数である場合は、文字セットはプリンターのフレキシブル・ディスクからアクセスされます。ID が奇数である場合は、文字セットはイメージ・ライブラリーから検索されます。

文字配列テーブルは、INCLUDE、TABLE、および NAME ユーティリティー制御ステートメントを使用して作成されます。INCLUDE ステートメントは、新しいモジュールの基礎としてコピーされ、使用される既存の文字配列テーブルを識別します。TABLE ステートメントは、新規または変更済みモジュールの内容について記述します。NAME ステートメントは、文字配列テーブルを識別し、さらに、それが新規であるか、または既存のモジュールを同じ名前で置き換えるのかを示します。

DEVICE=3800M3 パラメーターが指定されている OPTION ステートメントは、3800 モデル 3 の、既存の文字配列テーブルを印刷するときに指定してください。これは、システムから、文字配列テーブルと関連している図形変更モジュール名に正しい接頭部を割り当てるようにするためです。

IBM 提供の文字配列テーブルおよび文字セットについては、「*IBM 3800 Printing Subsystem Programmer's Guide*」を参照してください。

注：選択した文字配列テーブルは、文字セットにある文字をすべて含んでいない場合があります。文字配列テーブルは、印刷トレーン (1 つまたは複数の完全な完全な文字セットのサブセットの場合もある) に対応します。文字セットがロードされるたびに、セット内のすべての文字 (64 文字まで) は、プリンターの WCGM にロードされます。変換テーブルによって参照される文字だけを印刷することができます。

TABLE モジュールの構造

ヘッダー情報に続く文字配列テーブルデータは、以下のコンポーネントによって構成されています。

- 256 バイトの変換テーブル
- コード識別文字セットの 4 つの 2 バイト・フィールドおよび WCGM シーケンス番号
- 図形文字変更モジュール名の 4 つの 4 バイト・フィールド

変換テーブルは 256 個の 1 バイト項目によって構成されており、それぞれは、4 つの WCGM のいずれかにある 64 個の位置の 1 つを指しています。

- それぞれの変換テーブル・バイトのビット 0 および 1 は、4 つの WCGM のいずれかを参照しており、ビット 2~7 は、WCGM にある 64 アドレス (0~63) を指しています。変換テーブルのコピーを使用して、SETPRT が文字セットを、呼び出された WCGM (SETPRT) 以外の WCGM にロードする場合、変換テーブルのそれぞれの非 X'FF' バイトのビット 0 および 1 を、ロードされている WCGM に対応するように変更してください。159 ページの図 31 は、文字配列テーブル・モジュールの構造を示したものです。

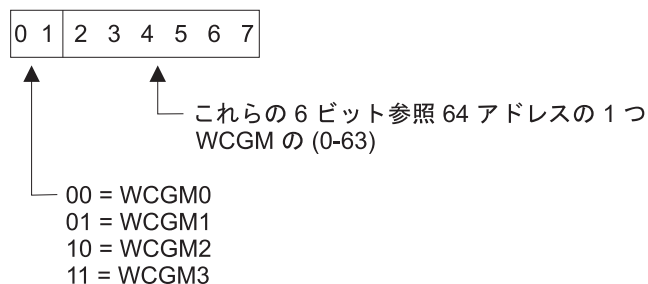


図 31. 文字配列テーブル・モジュールの構造

- 'X'FF' のバイト値は、使用不能文字を示し、空白として印刷し、さらにデータ・チェックを行います。データ・チェックのブロック・オプションを選択すると、データ・チェックは行われません。
- 変換テーブルは複数の WCGM をアドレッシングでき、さらに複数の変換テーブルは 1 つの WCGM をアドレッシングできます。IBM によって提供されている変換テーブルは、1 つまたは 2 つの WCGM のどちらかをアドレッシングします。

次の 2 つのコンポーネントは、文字セットと図形文字変更モジュールとのリンクを提供します。これらのコンポーネントは、4 つの 2 バイト・フィールドによって構成されており、文字セット ID と、それに対応する WCGM シーケンス番号が含まれています。さらに、その後に図形文字変更モジュールの 4 つの 4 文字名が続きます。フォーマットは次のとおりです。

- それぞれの CGMID は、2 つの 16 進数字を含む、1 バイト文字セット ID であり、ライブラリー文字セットを参照しています (「*IBM 3800 Printing Subsystem Programmer's Guide*」にリストアップされています)。それぞれの WCGMNO は、対応する WCGM シーケンス (X'00'~X'03') を参照しています。それぞれの名前は、図形文字変更モジュールの 4 文字名です。図 32 は、図形文字変更モジュールのフォーマットを示しています。

CGMID0	WCGMN00	CGMID1	WCGMN01
CGMID2	WCGMN02	CGMID3	WCGMN03
名前 1			
名前 2			
名前 3			
名前 4			

図 32. 図形文字変更モジュール

- ほとんどの標準文字配列テーブルには、図形文字変更は必要ありません。参照しているモジュールがない場合は、名前は空白 (X'40') になります。
- 最初の文字セットの後に、参照されている文字セットがない場合は、CGMID_x および WCGMNO_x は両方とも X'00' になります。

TABLE モジュールのリスト表示

図 33 は、文字配列テーブル・モジュールのリスト表示を示したものです。図の後の注では、四角で囲まれた番号の項目について説明しています。

	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	XA	XB	XC	XD	XE	XF
0X	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
1X	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
2X	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
3X	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
4X	0 00	*	*	*	*	*	*	*	*	*	0 0A	0 0B	0 0C	0 0D	0 0E	0 0F
5X	0 10	*	*	*	*	*	*	*	*	*	0 1A	0 1B	0 1C	0 1D	0 1E	0 1F
6X	0 20	0 21	*	*	*	*	*	*	*	*	0 2B	0 2C	0 2D	0 2E	0 2F	
7X	*	*	*	*	*	*	*	*	*	*	0 3A	0 3B	0 3C	0 3D	0 3E	0 3F
8X	*	1 01	1 02	1 03	1 04	1 05	1 06	1 07	1 08	1 09	*	1 0D	1 0E	1 0F	1 1A	1 1B
9X	*	1 11	1 12	1 13	1 14	1 15	1 16	1 17	1 18	1 19	*	1 1D	1 1E	1 1F	1 2A	1 2B
AX	1 3A	1 10	1 22	1 23	1 24	1 25	1 26	1 27	1 28	1 29	*	1 2A	1 2C	1 0A	1 2E	1 0B
BX	1 30	1 31	1 32	1 33	1 34	1 35	1 36	1 37	1 38	1 39	*	1 2D	1 2B	1 1B	1 21	1 1C
CX	*	0 01	0 02	0 03	0 04	0 05	0 06	0 07	0 08	0 09	*	*	*	*	*	*
DX	*	0 11	0 12	0 13	0 14	0 15	0 16	0 17	0 18	0 19	*	*	*	*	*	*
EX	*	*	0 22	0 23	0 24	0 25	0 26	0 27	0 28	0 29	*	*	*	*	*	*
FX	0 30	0 31	0 32	0 33	0 34	0 35	0 36	0 37	0 38	0 39	*	*	*	*	*	*

CGM 識別順序	0	1	2	3
----------	---	---	---	---

CGM 識別	BF	11	*	*
--------	----	----	---	---

GRAPHIC 変更レコード	GRF2
----------------	------

図 33. IEBIMAGE による文字配列テーブル・モジュールのリスト表示

図 33 の注

1. イメージ・ライブラリーのディレクトリーに存在している、文字配列テーブル・モジュールの名前 (システムが割り当てた 4 バイト接頭部も含む)。
2. IBM 提供の文字セットの 1 バイトの ID (この例では、テキスト 1 およびテキスト 2 文字セットです。これらの ID は X'8F' および X'11' です)。

SYS1.IMAGELIB にあるすべての文字またはユーザー指定のイメージ・ライブラリーは、奇数番号の ID によって表されています。3800 モデル 3 の場合、文字セット ID が偶数番号で指定されていれば、印刷時に 1 つ増やされて、その ID のある文字セットがロードされます。

3. WCGM の下に示されている文字セットを含むことになっている WCGM のシーケンス番号です (この例では、2 番目の WCGM であり、この ID は 1)。
4. この変換テーブル項目を配置する、8 ビット・データ・コードの走査パターンを含む WCGM のシーケンス番号です。
5. 3800 モデル 3 アドレスに転送された 8 ビットのデータ・コード X'B9' です。値が X'39' である、この変換テーブルの B9 位置は、使用する走査パターンのある WCGM に索引として入れられます (この例では、テキスト 2 スーパースクリプト 9)。
6. X'FF' に含まれている変換テーブル項目のそれぞれのリスト表示に、アスタリスクが示されています。これは、このロケーションをアドレッシングしている 8 ビット・データ・コードに図形が定義されておらず、したがって、印刷不可であることを示しています。
7. 文字セット ID のリストにあるアスタリスクは、対応する WCGM を使用するために文字セットがどれも指定されていないことを示しています。7F または FF を文字セット ID として指定する場合 (WCGM をロードせずに WCGM へのアクセスを許可する場合)、7F または FF はここで印刷します。
8. 図形文字変更モジュールの名前です。ライブラリーのディレクトリーにある名前と同じです (システムが割り当てた接頭部も含む)。

図形文字変更モジュールを文字配列テーブルと関連付けるときは、OPTION ステートメントに DEVICE パラメーター (3800 モデル 3 の場合) を指定して、システムから正しい接頭部 (GRF2) を図形文字変更モジュール名に割り当てるようにしなければなりません。

図形文字変更モジュールの作成

3800 図形文字変更モジュールは可変長であり、64 個までのセグメントがあります。それぞれのセグメントには 1 バイト (3800 モデル 1 の場合) または 6 バイト (3800 モデル 3 の場合) の記述情報があり、72 バイト (3800 モデル 1 の場合) または 120 バイト (3800 モデル 3 の場合) の図形文字の走査パターンがあります。

図形文字変更モジュールを作成するには、INCLUDE、GRAPHIC、OPTION および NAME ユーティリティ制御ステートメントを使用します。

INCLUDE ステートメントは、新しいモジュールの基礎としてコピーされ、使用される既存の図形文字変更モジュールを識別します。

図形文字変更モジュールを 3800 モデル 3 互換モード・モジュールに作成するには、OPTION ステートメントに DEVICE パラメーターを指定しなければなりません。

GRAPHIC ステートメントの後に 1 つまたは複数のデータ・ステートメントがある場合、ユーザー作成文字が定義されます。GRAPHIC ステートメントでは、別の図形文字変更モジュールから文字セグメントを選択することもできます。各 GRAPHIC ステートメントは、新規モジュールに含めるためのセグメントを作成します。

NAME ステートメントは新しいモジュールを作成し、モジュールがライブラリーに追加されるか、または同じ名前の既存のモジュールと置き換えるかを示します。複数の GRAPHIC ステートメントを INCLUDE および NAME ステートメント間にコーディングできます。このような GRAPHIC ステートメントはすべて同じ図形文字修正モジュールに適用されます。

GRAPHIC モジュールの構造

ヘッダー情報に続く図形文字変更データは、3800 モデル 1 の場合は 73 バイト・セグメントであり、3800 モデル 3 の場合は 126 バイト・セグメントです。モジュールでは最高 64 個までのセグメントが許可されています。図 34 は、このモジュールの構造を示したものです。

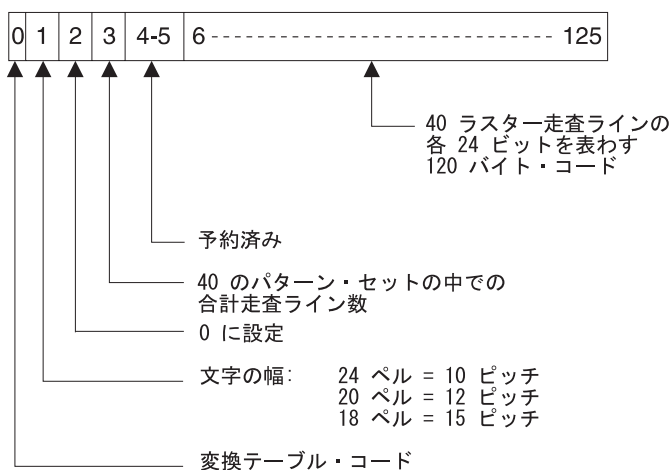


図 34. 1 文字のための 3800 図形文字変更モジュールの構造

図形文字を修正するとき、3800 は変換テーブル・コードを使用して、変換テーブルに索引化します。その位置 (1 バイトの WCGM コード) で見つかった内容によって、走査パターンと文字データが配置される WCGM 位置が決定します。

3800 モデル 1 印刷サブシステムの場合:

1 つの文字のための走査パターンを構成する 72 バイトの図形定義は、24 の 3 バイト・グループに分割されます。それぞれの 3 バイト・グループは、18 個の 1 ビット・エレメントの水平行 (パリティ情報を加えたもの) を表します。

3800 モデル 3 の場合: 1 つの文字のための走査パターンを構成する 120 バイトの図形定義は、40 の 3 バイト・グループに分割されます。それぞれの 3 バイトのグループは、24 個の 1 ビット・エレメントの水平行を表します。

GRAPHIC モジュールのリスト表示

163 ページの図 35 は、図形文字変更モジュールのリスト表示の抜粋を示したものです。この抜粋には、モジュールの 2 つのセグメントのリスト表示が含まれています。図の後の注では、図の中の四角で囲まれた番号の項目について説明しています。

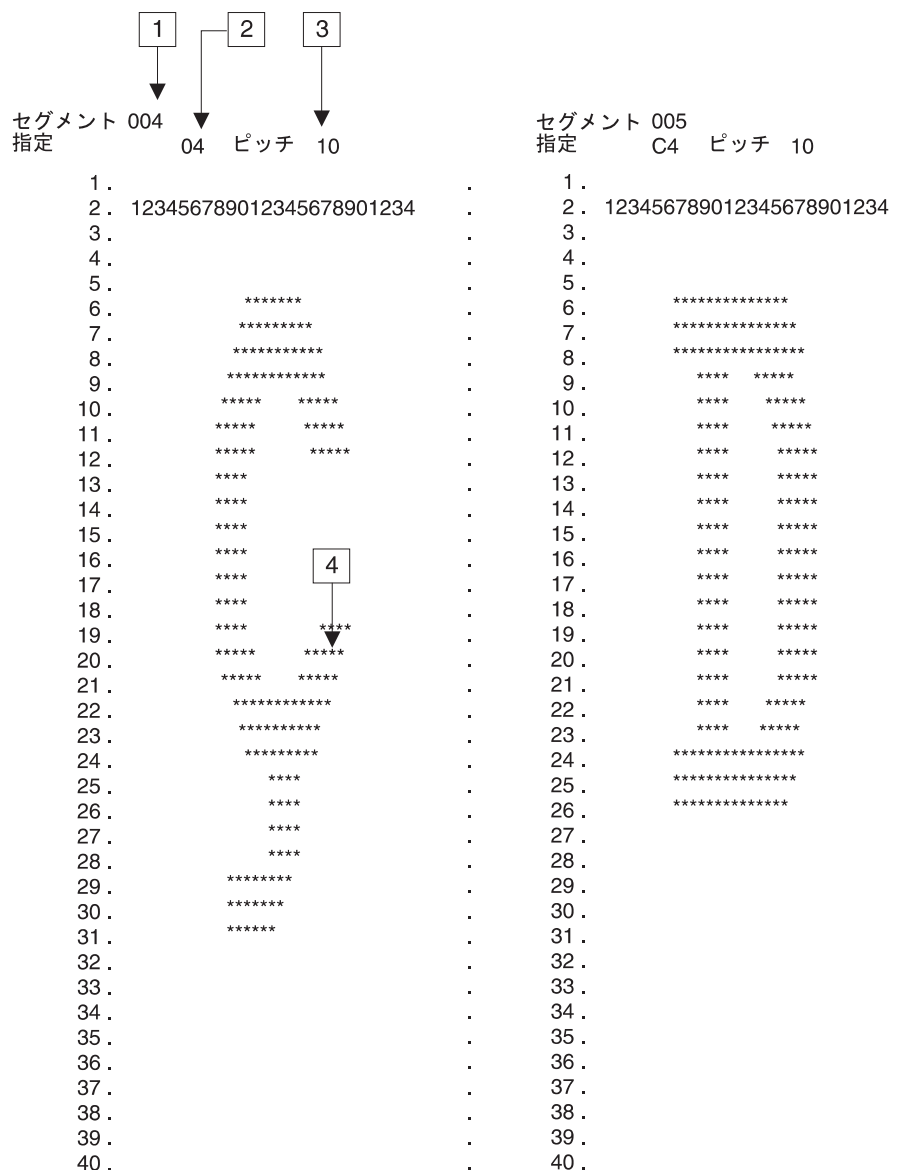


図 35. IEBIMAGE による図形文字変更モジュールの 2 つのセグメントのリスト表示

図 35 の注

1. モジュール内の文字セグメントのセグメント番号です。
2. 文字の 8 ビット・データ・コードです。
3. 文字のピッチです。
4. 文字の走査パターンです。指定されているビットがピッチの範囲外にある場合は、アスタリスクの代わりにドル記号 (\$) が印刷されます。

ライブラリー文字セット・モジュールの作成

3800 ライブラリー文字セット・モジュールは固定長モジュールであり、64 セグメントによって構成されています。それぞれのセグメントには、73 バイト (3800 モデル 1 の場合) または 126 バイト (3800 モデル 3 の場合) の、図形文字の走査パターンを含む情報、および WCGM 位置を識別するコード (00~3F) (ここに走査パターンがロードされる) が含まれます。

ライブラリー文字セット・モジュールを作成するには、INCLUDE、CHARSET、NAME 制御ステートメントを使用します。

INCLUDE ステートメントは、既存のモジュールを識別します。

3800 モデル 3 互換モード・モジュール・フォーマットでライブラリー文字セット・モジュールを作成するためには、DEVICE パラメーターが指定されている OPTION ステートメントが必要です。

CHARSET ステートメントの後に 1 つまたは複数のデータ・ステートメントがある場合、ユーザー作成文字が定義されます。CHARSET ステートメントでは、別のライブラリー文字セットまたは図形文字変更モジュールから文字セグメントを選択することもできます。

NAME ステートメントは、作成された文字セットの ID を指定し、さらにそれが、既存のモジュールを置き換わるかどうかを示します。複数の CHARSET ステートメントを INCLUDE および NAME ステートメント間にコーディングできます。このような CHARSET ステートメントはすべて同じライブラリー文字セット・モジュールに適用されます。

CHARSET モジュールの構造

ヘッダー情報に続くライブラリー文字セット・データは、3800 モデル 1 の場合は 73 バイト・セグメントであり、3800 モデル 3 の場合は 126 バイト・セグメントです。それぞれのモジュールには 64 セグメントがあります。未定義のままライブラリー文字セット・モジュールに残されているそれぞれのセグメントごとに、IEBIMAGE は、未定義文字のグラフィック・シンボルを挿入します。図 36 は、ライブラリー文字セット・モジュールの構造を示したものです。

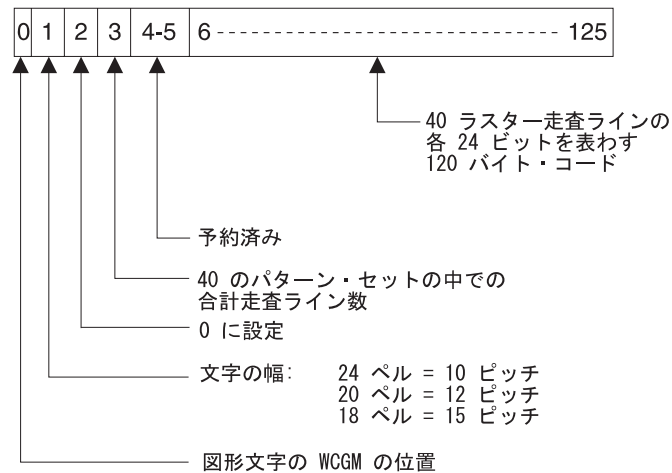


図 36. 1 文字のための 3800 モデル 3 ライブラリー文字セット・モジュールの構造

ライブラリー文字セットが、直接 WCGM にロードされます。SETPRT は、それぞれの 73 バイト・セグメントの最初のバイト (3800 モデル 1 の場合) または 126 バイト・セグメントの最初のバイト (3800 モデル 3 の場合) にある 6 ビット・コードを、WCGM 位置のアドレスとして使用します。ここに、残りの 72 バイト (3800 モデル 1 の場合) または 125 バイト (3800 モデル 3 の場合) がロードされます。

3800 モデル 1 の場合: 1 つの文字のための走査パターンを構成する 73 バイトの図形定義は、24 の 3 バイト・グループに分割されます。それぞれの 3 バイトのグループは、18 個の 1 ビット・エレメントの水平行を表します。

3800 モデル 3 の場合: 1 つの文字のための走査パターンを構成する 126 バイトの図形定義は、40 の 3 バイト・グループに分割されます。それぞれの 3 バイトのグループは、24 個の 1 ビット・エレメントの水平行を表します。

CHARSET モジュールのリスト表示

図 37 は、ライブラリー文字セット・モジュールのリスト表示の抜粋を示したものです。この抜粋には、ライブラリー文字セットの 2 つのセグメントのリスト表示が含まれています。図の後の注では、四角で囲まれた番号の項目について説明しています。

セグメント 指定	06	ピッチ	10	LCS299	セグメント 指定	008	ピッチ	10	
1.				.	1.			.	
2.	123456789012345678901234			.	2.	123456789012345678901234		.	
3.				.	3.			.	
4.				.	4.			.	
5.				.	5.			.	
6.				.	6.		***	.	
7.		*****		.	7.		*****	.	
8.		*****		.	8.	*****		.	
9.		*****		.	9.	*****		.	
10.		*****		.	10.	*****		.	
11.		*****		.	11.	*****	*****	.	
12.		*****		.	12.	*****	*****	.	
13.		*****		.	13.	*****	*****	.	
14.		***	***	.	14.	***	*****	.	
15.		***	***	.	15.	***	*****	.	
16.		***	***	.	16.	***	*****	.	
17.		***	*****	.	17.	***	*****	.	
18.		***	*****	.	18.	***	*****	***	.
19.		***	*****	.	19.	***	*****	***	.
20.		***	***	.	20.	***	*****	***	5
21.		***	***	.	21.	*****	***	.	
22.		*****		.	22.	*****	***	.	
23.		*****		.	23.	*****	***	.	
24.		*****		.	24.	*****	***	.	
25.		***	***	.	25.	*****	***	.	
26.		***	***	.	26.	*****	***	.	
27.		***	*****	.	27.	*****	*****	.	
28.		***	*****	.	28.	*****	*****	.	
29.				.	29.	*****	*****	.	
30.				.	30.	*****	*****	.	
31.				.	31.	*****		.	
32.				.	32.	***		.	
33.				.	33.			.	
34.				.	34.			.	
35.				.	35.			.	
36.				.	36.			.	
37.				.	37.			.	
38.				.	38.			.	
39.				.	39.			.	
40.				.	40.			.	

図 37. IEBIMAGE によるライブラリー文字セットの 2 つのセグメントのリスト表示

図 37 の注:

1. システムが割り当てた 4 バイト接頭部を含む、ライブラリー文字セット・モジュールの名前です。
2. モジュール内の文字セグメントのセグメント番号です。
3. WCGM 位置の 6 ビット・コードです。
4. 文字のピッチです。
5. 文字の走査パターンです。指定されているビットがピッチの範囲外にある場合は、アスタリスクの代わりにドル記号 (\$) が印刷されます。

入出力

IEBIMAGE は以下の入力を使用します。

- ユーティリティー制御ステートメントを含む制御データ・セット。
- 文字変換援助機能によって生成されるソース・ステートメント。

IEBIMAGE は以下の出力を生成します。

- 3800 モデル 1 およびモデル 3 印刷装置、3262 モデル 5 印刷装置、または 4248 印刷装置で使用するための新しいモジュール (複数も可) をイメージ・ライブラリーに格納します。モジュールのタイプは、次のいずれかになります。
 - 用紙制御バッファ・モジュール (3800 または 4248)
 - コピー変更モジュール (3800 のみ)
 - 文字配列テーブル・モジュール (3800 のみ)
 - 図形文字変更モジュール (3800 のみ)
 - ライブラリー文字セット・モジュール (3800 のみ)

4248 FCB モジュールの作成では、4248 (接頭部 FCB4) または 3211 (接頭部 FCB2) フォーマット FCB が入力として使用されることがあります。IEBIMAGE は、最初に名前に FCB4 接頭部を使用します。次に、その名前でモジュールが存在しない場合は、接頭部は FCB2 に変更されます。しかし、IEBIMAGE を使用して FCB2 モジュールを出力として作成することはできません。

- それぞれの新しいモジュールごとの出力データ・セットのリスト表示には、次のものが含まれます。
 - モジュール ID
 - ジョブで使用されるユーティリティー制御ステートメント
 - モジュールの内容
 - メッセージおよび戻りコード

IEBIMAGE の戻りコードについては、付録 Aを参照してください。

制御

IEBIMAGE は、ジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。

ジョブ制御ステートメント

167 ページの表 26 は、IEBIMAGE で使用するジョブ制御ステートメントをまとめたものです。

表 26. IEBIMAGE で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEBIMAGE)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。 PARM パラメーターは、指定できません。
SYSPRINT DD	ステートメントおよびメッセージをシステム出力装置にリスト表示するとき使用する順次メッセージ・データ・セットを定義します。
SYSUT1 DD	ライブラリー・データ・セットを定義します (SYS1.IMAGELIB またはユーザー定義ライブラリー)。
SYSIN DD	制御データ・セットを定義します。これは、通常は入力ストリーム内にあります。

SYSPRINT DD ステートメント

SYSPRINT データ・セットのブロック・サイズは、121 または 121 の倍数でなければなりません。任意のブロック化因数を指定できます。それぞれの 121 バイト出力レコードの最初の文字は、ISO/ANSI 制御文字です。

SYSUT1 DD ステートメント

IEBIMAGE ジョブの実行中にライブラリー・データ・セットが他のジョブによって更新されないようにするには、DISP=OLD を SYSUT1 DD ステートメントで指定しなければなりません。

SYSUT1 DD では、ユーザー・イメージ・ライブラリーを指定できます。このライブラリーは、SYS1.IMAGELIB と同じ特性がなければならぬため、SETPRT マクロ命令でシステムに対して識別されるか、『SYS1.IMAGELIB』に名前変更されることとなります。JES を使用している場合、SETPRT でイメージ・ライブラリーを指定することはできません。3800 印刷サブシステムでユーザー独自のイメージ・ライブラリーを使用する方法については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。SETPRT については、*z/OS DFSMS Macro Instructions for Data Sets* を参照してください。

SYSIN DD ステートメント

SYSIN データ・セットのブロック・サイズは、80 または 80 の倍数でなければなりません。任意のブロック化因数を指定できます。SYSIN DD ステートメントの DCB は、JCL では省略します。

ユーティリティー制御ステートメント

IEBIMAGE は、168 ページの表 27 のユーティリティー制御ステートメントによって制御されます。

ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

表 27. IEBIMAGE で使用するユーティリティ制御ステートメント

ステートメント	用途
FCB	3800 または 4248 用紙制御バッファ・モジュールを作成し、それをイメージ・ライブラリーに保管します。
COPYMOD	3800 コピー変更モジュールを作成し、それをイメージ・ライブラリーに保管します。
TABLE	3800 文字配列テーブル・モジュールを作成し、それをイメージ・ライブラリーに保管します。
GRAPHIC	3800 図形文字変更モジュールを作成し、それをイメージ・ライブラリーに保管します。
CHARSET	3800 ライブラリー文字セット・モジュールを作成し、それをイメージ・ライブラリーに保管します。
INCLUDE	新しいモジュールの基礎としてコピーされ、使用されるイメージ・ライブラリー・モジュールを識別します。
NAME	新規または既存のライブラリー・モジュールの名前を指定します。
OPTION	IEBIMAGE ジョブの 3800 モデル 3 または 4248 印刷装置互換性、または COPYMOD オーバーラン行 / インチを指定します (任意)。

操作グループ

IEBIMAGE ユーティリティ制御ステートメントは、ライブラリー・モジュールを作成するか、または印刷するために、グループとしてまとめられています。ステートメントのグループはそれぞれ、**操作グループ**と呼ばれています。ジョブの入カストリームには、多くの操作グループを入れることができます。記述できる操作グループは以下のとおりです (オペランドなしでコーディング)。

- FCB モジュールを作成または印刷する場合:

```
[OPTION]
[INCLUDE]
FCB
NAME
```

注: 有効な操作を FCB ステートメントにコーディングしなければ、4248 FCB モジュールを印刷することはできません。

- コピー変更モジュールを作成または印刷する場合:

```
[INCLUDE]
[OPTION]
COPYMOD
[additional COPYMOD statements]
NAME
```

- 文字配列テーブル・モジュールを作成または印刷する場合:

```
[INCLUDE]
[OPTION]
TABLE
NAME
```

- 図形文字変更モジュールを作成または印刷する場合:

```
[INCLUDE]
[OPTION]
{GRAPHIC[GRAPHIC, followed immediately by data statements]}
[additional GRAPHIC statements]
NAME
```

- ・ ライブラリー文字セット・モジュールを作成または印刷する場合:

```
[INCLUDE]
[OPTION]
{CHARSET|CHARSET, followed immediately by data statements}
[additional CHARSET statements]
NAME
```

モジュールを印刷するためにユーザーが行う必要があるのは、オペランドを指定せずに機能ステートメント (FCB、COPYMOD、TABLE、GRAPHIC、CHARSET のいずれか) を指定し、その後にモジュールを命名するための NAME ステートメントを続けることだけです。しかし、有効なオペランドを FCB ステートメントにコーディングしなければ、4248 FCB モジュールを印刷することはできません。

FCB ステートメント

FCB ステートメントは、3800、3262 モデル 5、または 4248 印刷装置の用紙制御バッファ (FCB) モジュールの内容を指定します。つまり、行送りコード (行 / インチ)、チャンネル・コード (シミュレート紙送り制御穿孔)、および用紙のサイズを指定します。4248 印刷装置の場合、FCB ステートメントでは、水平コピー機能の印刷位置、プリンター速度、FCB イメージがデフォルトとして使用されるかどうかも指定します。

FCB ステートメントの後には、必ず NAME ステートメントを指定しなければなりません。また、OPTION ステートメントで DEVICE=4248 を指定する場合に限り、FCB ステートメントの前に INCLUDE ステートメントを指定できます。

オペランドが指定されていない FCB ステートメントの後に、イメージ・ライブラリーにある 3800 FCB モジュールを識別する NAME ステートメントが続く場合は、モジュールがフォーマットされて印刷されます。3262 モデル 5 と 4248 FCB モジュールは、有効なオペランドが実行されない限り、FCB ステートメントによって印刷することはできません。FCB モジュールを作成するには、少なくとも 1 つのオペランドのある FCB ステートメントをコーディングしてください。154 ページの『FCB モジュールのリスト表示』は、印刷した FCB モジュールのフォーマットを示したものです。

FCB ステートメントの構文は次のとおりです。

[label]	FCB	[LPI=(([l],[n])[,([l],[n])[,...]])] [,CHx=(line[,line[,...]])[,CHx=(line[,line[,...]])[,...]] [,SIZE=length] [,LINES=lines] [,COPYP=position] [,PSPEED={LIM HIN}] [,DEFAULT={YES NO}]
		注: COPYP、PSPEED、DEFAULT は、4248 FCB モジュールの場合にのみ指定できます。

各項目の意味は以下のとおりです。

LPI=(([l],[n]) [,([l],[n])[,...]])

インチごとの行数、および行送りで印刷する行数を指定します。

- l* インチごとの行数を指定します。6、8、12 (3800 モデル 1 の場合)、6、8 (3262 モデル 5 または 4248 の場合)、6、8、10、12 (3800 モデル 3 の場合) を指定できます。
- n* 行送り *l* で行数を指定します。プリンターで共通に使用される用紙サイズを使用する場合、*l* が 6 のときは *n* は 10 進値 1 から 60 になり、*l* が 8 のときは 1 から 80 になり、*l* が 10 のときは 1 から 100 になり、*l* が 12 のときは 1 から 120 になります。

プリンターで ISO の用紙サイズを使用する場合、*l* が 6 のときは *n* は 10 進値 1 から 66 になり、*l* が 8 のときは 1 から 88 になり、*l* が 10 のときは 1 から 110 になり、*l* が 12 のときは 1 から 132 になります。用紙サイズについては、「*IBM 3800 Printing Subsystem Programmer's Guide*」を参照してください。

指定した行数の合計数が、必ず 1/2 インチの倍数になるようにすることは、ユーザーの責任です。なお、1/2 インチは、1.27cm に当たります。

行数の合計は、その用紙の使用可能な長さを超えるものであってはなりません。3800 の場合、用紙の上下 1/2 インチのコーディングを指定しないでください。その指定は、IEBIMAGE が行います。

SIZE、LINES、LPI パラメーターが FCB ステートメントで指定されているときに、それぞれのパラメーター値がその他のパラメーター値に対して確認され、矛盾したページの長さが指定されないようにします。たとえば、SIZE=35 は 3-1/2 インチの長さを指定しますが、3800 の有効な LPI 値 では、この長さとして印刷可能な 2-1/2 インチ以上を定義することはできません。

(*l*,*n*) の組み合わせを複数指定するときは、*l* をそれぞれの組み合わせに対して指定し、*n* を最後の組み合わせ以外の各組み合わせに指定しなければなりません。

それぞれのインチごとに 12 行を指定するときは、圧縮文字セットのいずれかを使用します。他の文字セットが 12 行 / インチで印刷される場合、その文字の上部または下部は印刷されないことがあります。

l のみが指定されているとき、または *l* が LPI リストの最後のパラメーターであるときは、そのページにある残りの行はすべて、1 行 / インチになります。

LPI が指定されていないときに、そのページの行はすべて、6 行 / インチになります。

指定されている行の合計数が、指定可能な最大数よりも少ない場合、残りの行のデフォルトは、6 行 / インチになります。

INCLUDE が指定されている場合、LPI の値は、組み込まれている FCB モジュールから取られます。INCLUDE ステートメントの *module name* パラメーターの説明を参照してください。

CHx=(line[,line [...]])

このコードが指定されているときに、スキップ先のチャンネル・コード (複数も可) と行番号 (複数も可) を指定します。

CHx

チャンネル・コードを指定します。x は、1~12 の 10 進整数です。

line

スキップ先の印刷行の行番号を指定します。これは、10 進整数で表されます。このページの最初の印刷可能行は、行番号 1 です。

line の値は、用紙の最後の印刷可能行の行番号より大きくすることはできません。

印刷行に指定できるチャンネル・コードは 1 つだけです。ただし、複数の印刷行に同じチャンネル・コードを含めることができます。

規則:

- チャンネル 1 は、用紙の最初の印刷可能行を識別するために使用します。BSAM または QSAM を使用して 3800 に直接割り振りを行うためのジョブ入力サブシステムおよび CLOSE ルーチンには、印刷されるデータにチャンネル 1 へのスキップがない場合でも、チャンネル 1 コードが必要です。
- チャンネル 9 は、特別な行を識別するときに使用します。チャンネル 9 を使用した場合の入出力割り込みを避けるには、行数を計算して、行の位置を判別してください。
- チャンネル 12 は、使用する用紙の最後の印刷行を識別するために使用します。チャンネル 12 を使用した場合の入出力割り込みを避けるには、行数を計算して、ページ・サイズを判別してください。
- スキップ操作を停止するチャンネル・コードがない FCB を使用すると、対応するスキップが行われるときにプリンターでデータ・チェックが実行されます。データ・チェックはブロックできません。

INCLUDE が指定されている場合、CHx の値は、組み込まれている FCB モジュールから取られます。INCLUDE ステートメントの *module name* パラメータの説明を参照してください。

SIZE=length

用紙の垂直の長さを 1/10 インチ単位で指定します。3800 で有効な長さについては、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。用紙の全体の長さを指定します (たとえば、3800 では、11 インチ用紙で SIZE=110 と指定する) が、印刷に利用可能な実際のスペースの量としては、印刷が行われない上下 1/2 インチを減らさなければなりません。

SIZE、LINE、LPI キーワードが FCB ステートメントで指定されているときに、それぞれのパラメータ値がその他のパラメータ値に対して確認され、矛盾したページの長さが指定されないようにします。たとえば、SIZE=35 は 3-1/2 インチの長さを指定しますが、3800 の有効な LPI 値 では、この長さとして印刷可能な 2-1/2 インチ以上を定義することはできません。

SIZE が指定されていないときは、用紙の長さのデフォルトは、LINES で指定されている値になります。LINES が指定されていない場合は、SIZE は 11 インチ (110) と想定されます。

INCLUDE が指定されている場合、SIZE の値は、組み込まれている FCB モジュールから取られます。INCLUDE ステートメントの *module name* パラメータの説明を参照してください。

LINES=lines

FCB に含める行数の合計を指定します。

lines

1~256 の 10 進数であり、ページ上の行数を示します。

LINES、SIZE、LPI パラメーターが FCB ステートメントで指定されているときに、それぞれのパラメーター値がその他のパラメーター値に対して確認され、矛盾したページの長さが指定されないようにします。

LINES が指定されていないときは、用紙の長さのデフォルトは、LPI の値に SIZE 値を掛けた値になります。単位はインチです。SIZE パラメーターが指定されていない場合は、LINES のデフォルトである LPI 値の 11 倍になります。

INCLUDE が指定されている場合、LINES の値は、組み込まれている FCB モジュールから取られます。INCLUDE ステートメントの *module name* パラメーターの説明を参照してください。

COPYP=position

水平コピーの印刷開始位置 (左余白からの文字スペースの数) を指定します。

position

2~168 の 10 進数であり、水平コピーの印刷開始場所を示します。4248 印刷装置に 132 個の印刷位置しかない場合、指定可能な最大数は 132 です。

COPYP=0 がコーディングされている場合、組み込まれている FCB モジュールに設定されていた COPYP 値はすべて指定変更され、水平コピー機能はオフにされます。COPYP=1 は指定できません。

INCLUDE が指定されており、組み込まれている FCB モジュールが 4248 印刷装置用のみフォーマットされている場合、デフォルトは、組み込まれている FCB モジュールの COPYP 値になります。COPYP 値が指定されていない場合は、デフォルト値は 0 になります。

COPYP は、3800 FCB モジュールの場合は無効です。また、3262 モデル 5 FCB モジュールでは無視されます。

指定されている COPYP 値は、プリンターに送信されるデータの最大数に影響します。活動化されている水平コピー機能で実行されているチャンネル・プログラムは、必ず長さの誤り抑制 (SIL) ビットを設定しなければなりません。また、データの長さが印刷位置の数の 1.5 倍のサイズか、または 2 つのコピー位置の小さい方のサイズを超えないものにする必要があります。

PSPEED={L|MIH|N}

4248 印刷装置の印刷速度を指定します。プリンター速度が印刷の品質に影響することに注意してください。低速の方が品質は良くなります。

L または LOW

プリンター速度を 2200 行 / 分 (LPM) に設定します。

M または MEDIUM

プリンター速度を 3000 LPM に設定します。

H または HIGH

プリンター速度を 3600 LPM に設定します。

N または NOCHANGE

現在のプリンター速度を変化させないことを示します。

INCLUDE が指定されており、組み込まれているモジュールが 4248 印刷装置用のみフォーマットされている場合、デフォルトは、組み込まれている FCB モジュールの PSPEED 値になります。それ以外の場合は、デフォルトは NOCHANGE (または N) になります。

PSPEED は、3800 FCB モジュールの場合は無効です。また、3262 モデル 5 FCB モジュールでは無視されます。

DEFAULT={YES|NO}

この 4248 FCB イメージが、OPEN 処理によってデフォルト・イメージとして扱われるかどうかを指定します。デフォルト・イメージは、特定のイメージを要求していないジョブの場合に、システムが使用するイメージです。

ジョブが特定の FCB イメージを要求しておらず、現在のイメージがデフォルトではない場合、OPEN 時にオペレーターに FCB イメージの指定が要求されます。

INCLUDE が、デフォルト・イメージとして指定されていた 4248 FCB モジュールをコピーするために使用される場合、DEFAULT=NO が指定されていない限り、新しいモジュールもデフォルト・イメージと見なされます。

DEFAULT は、3800 FCB イメージの場合は無効です。

COPYMOD ステートメント

コピー変更モジュールは、ヘッダー情報の後に、1 つまたは複数の変更セグメントが続きます。ヘッダー情報には、モジュール名と長さが含まれます。それぞれの変更セグメントには印刷するテキストが含まれており、テキストが適用されるコピー (複数も可) を識別し、コピーの各ページのテキスト位置を指定します。

COPYMOD ステートメントは、コピー変更モジュールの変更セグメントの 1 つの内容を指定します。操作グループ内に複数の COPYMOD ステートメントをコーディングできます。そのようにコーディングされたすべての COPYMOD ステートメントは、同じコピー変更モジュールに適用されます。

IEBIMAGE は、コピー変更モジュールに指定されている変更セグメントを分析し、モジュールがプリンターで使用されているときに生じる行オーバーラン条件を予測します。行オーバーラン条件は、行を印刷する時間内に行の修正が完了しないときに生じます。コピー変更で使用可能な時間は、プリンターの垂直行送り (行 / インチ) によって異なります。

IEBIMAGE がユーザー仕様でコピー変更モジュールを作成する場合、プログラムは、計画した印刷を行う時に、修正にかかる時間を見積もります。12 LPI (行 / インチ) で行を印刷するのに利用可能な時間内に修正を行うことができる場合は、その修正はさらに、6、8 LPI (モデル 1 の場合)、または 6、8、10 LPI (モデル 3 の場合) でも行うことができます。(6、8、10、12 LPI は、3800 モデル 3 印刷装置でのみ利用可能な印刷密度であることに注意してください。) しかし、コピー変更

モジュールが複雑なため、行を 6 LPI で印刷するために利用できる時間内でコピー変更モジュールを作成できない場合には、8、10 (モデル 3 の場合のみ) または 12 LPI で作成することもできません。(10、12 LPI では、行を印刷するために利用できる時間が 6 LPI よりもずっと少ないことに注意してください。)

IEBIMAGE が、指定した行数/インチで印刷するときに、コピー変更モジュールを使用する場合に、そのモジュールがオーバーランの原因になりそうということが判別されると、プログラムは、その影響に関する警告メッセージを生成します。警告が 6 LPI に適用される場合、オーバーラン条件は 8、10 (モデル 3 の場合のみ)、および 12 LPI にも適用されます。警告が 8 LPI に適用される場合、条件は 10 (モデル 3 の場合のみ) および 12 LPI にも適用されます。警告が 10 LPI に適用される場合、条件は 12 LPI にも適用されます。

6 LPI で印刷する場合だけに特定のコピー変更モジュールを使用することを計画している場合、必要のない 8、10 (モデル 3 の場合のみ)、および 12 LPI の警告メッセージを抑止するには、OPTION ステートメントの **OVERRUN** パラメーターに 6 を指定します。8 LPI でのみ印刷することを計画している場合、OPTION ステートメントに **OVERRUN=8** を指定して、10 (モデル 3 の場合のみ) および 12 LPI の不要な警告メッセージの抑止を要求できます。**OVERRUN** のコーディングの詳細については、186 ページの『**OVERRUN** の使用』を参照してください。コピー変更モジュールの使用方法については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。同じ文字サイズまたはスタイルを使用するか、または出力データ・セットのデータを印刷するために使用したサイズまたはスタイルから 1 だけ異なるものを使用して、コピー変更テキストを印刷できます。

COPYMOD ステートメントの後には、必ず **NAME** ステートメントまたは別の **COPYMOD** ステートメントを指定しなければならず、その前には **INCLUDE** ステートメントを指定することができます。複数の **COPYMOD** ステートメントをコーディングするときは、**IEBIMAGE** は、コピー番号内の行番号に基づいてステートメントをソートします。オペランドが指定されていない **COPYMOD** ステートメントで、その後、コピー変更モジュールを識別する **NAME** ステートメントが続く場合は、モジュールがフォーマットされて印刷されます。157 ページの『**COPYMOD** モジュールのリスト表示』は、印刷したモジュールの構文を示したものです。

コピー変更モジュールのセグメントを作成するときに使用する、**COPYMOD** ステートメントの構文は次のとおりです。

[label]	COPYMOD	COPIES =(starting-copy[,copies]) ,LINES =(starting-line[,lines]) ,POS =position ,TEXT =(([d]t,'text') [(,[d]t,'text')][,...])
---------	----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

COPIES=(starting-copy[,copies])

開始コピー番号および修正コピーの合計数を指定します。

starting-copy

開始コピー番号を指定します。これは 1~255 の 10 進整数で表されます。開始コピーの値は必須です。

copies

修正テキストを含むことになるコピーの数を指定します。これは 1~255 の 10 進整数で表されます。コピー数が指定されていない場合、デフォルトは 1 つのコピーになります。

starting-copy と *copies* の合計が 256 (JES3 の場合は 255) を超えてはなりません。

LINES=(*starting-line* [, *lines*])

開始行番号および修正行の合計数を指定します。

starting-line

開始行番号を指定します。これは 1~132 の 10 進整数で表されます。*starting-line* の値は必須です。

lines

修正セグメントのテキストを含むことになる行の数を指定します。これは 1~132 の 10 進整数で表されます。*lines* が指定されていない場合、デフォルトは、1 行になります。

starting-line と *lines* の合計が 133 を超えてはなりません。この合計が用紙サイズとして (FCB ステートメントで) 指定されている行数を超える場合、用紙の末尾を超えた行の変更テキストは印刷されません。

POS=*position*

修正テキストの印刷開始位置 (左余白からの文字位置数) を指定します。

position

印刷開始位置を指定します。これは、1~204 の整数で表されます。下記の TEXT パラメーターの制約事項を参照してください。

印刷行に入れることができる文字の最大数は、各文字のピッチおよび用紙の幅によって異なります。

各用紙幅で印刷行に入れることのできる文字の最大数については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。

TEXT=((*d*]*t*,*text*') [*d*]*t*,*text*')][,...])

変更テキストを指定します。テキストの用紙上の位置は、LINES および POS パラメーターに基づいて決まります。これらの位置で出力データ・セットのテキストが置き換えられます。

d 複写因数 (テキストが繰り返される回数) を指定します。*d* は、1~204 の 10 進整数によって表されます。*d* が指定されていない場合は、デフォルトは 1 になります。

t テキストが入力される用紙を指定します。*C* は文字、*X* は 16 進数です。*t* は必須です。

text

テキストを指定します。これは、単一引用符で囲まれます。

テキスト・タイプが *C* である場合、任意の有効文字を指定できます。ブランクも有効文字です。単一引用符は、2 つの単一引用符としてコーディングします。結果が *X'FF'* となる文字を指定することはできません。テキスト・タイプが *X* である場合、テキストは *X'00'~X'FE'* 間の値を指定する 2 文字の増分値でコーディングします。 *X'FF'* を指定することはできません。

印刷開始位置の合計 (POS パラメーターを参照) およびテキスト文字の合計数が、205 を超えてはなりません。用紙幅がテキストに必要なスペースの量よりも少ない場合 (文字ピッチ、開始位置、文字数によって決まる)、文字が用紙の右余白を超えて印刷されることはありません。

テキスト文字が、変換テーブル項目に *X'FF'* が入っている文字を指定している場合、コピー変更モジュールがロードされる時に、プリンターはデータ・チェック・エラー標識を設定します。このエラー標識はブロックできます。

TABLE ステートメント

TABLE ステートメントは、文字配列テーブル・モジュールを作成するために使用します。文字配列テーブルが IEBIMAGE によって作成され、INCLUDE ステートメントが指定されているときは、コピーした文字配列テーブルの内容が、新しい文字配列テーブルの基礎として使用されます。INCLUDE ステートメントが指定されていない場合は、新しい文字配列テーブル・モジュールの各変換テーブル項目が *X'FF'* に初期化され、図形文字変更モジュール名フィールドがブランクで設定され (*X'40'*)、最初の文字セット ID が *X'83'* (ゴシック 10 ピッチ・セット) に設定されます。残りの ID は *X'00'* に設定されます。

文字配列テーブルを初期化した後に、IEBIMAGE は、TABLE ステートメントで指定したデータで表を変更します。文字セット ID、図形文字変更モジュール、および指定した変換テーブル項目のデータを変更します。文字配列テーブルを作成するときは、その表に、少なくとも 1 つの印刷可能文字への参照が含まれていなければなりません。それぞれの操作グループに指定できる TABLE ステートメントは 1 つです。TABLE ステートメントの前に INCLUDE ステートメントおよび OPTION ステートメントを置くことができます。TABLE ステートメントの後には必ず NAME ステートメントを指定しなければなりません。

オペランドが指定されていない TABLE ステートメントの後に、ライブラリーにある文字配列モジュールを識別する NAME ステートメントが続く場合は、モジュールがフォーマットされて印刷されます。TABLE ステートメントの前に OPTION ステートメントを指定しなければなりません。3800 モデル 3 の場合は、そのステートメントに *DEVICE=3800M3* パラメーターを指定しなければなりません。160 ページの『TABLE モジュールのリスト表示』は、印刷された文字配列テーブル・モジュールのフォーマットを示したものです。

TABLE ステートメントの構文は次のとおりです。

[label]	TABLE	[CGMID=(set0[, set1][,...]) [GCMLIST={(gcm1[, gcm2][,...])DELETE} [LOC=((xloc[, {cloc[,setm] FF})[,...])]
---------	-------	-----------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

CGMID=(set0[,set1][,...])

文字配列テーブルで使用される文字セットを識別します。(IBM 提供の文字セットおよびその ID については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。) CGMID が指定されているときは、すべての文字セット ID が変更されます。1 つの文字セットのみが指定されている場合、他の 3 つの ID は 'X'00' に設定されます。

setx

文字セットの 1 バイトの ID です。4 つの文字セット ID までを指定できます。set0 は最初の書込可能文字生成モジュール (WCGM) にロードされる文字セット、set1 は 2 番目の WCGM にロードされる文字セット、という具合に指定します。文字セット ID は、変換テーブル項目に合わせて正しいシーケンスで指定する必要があります。

GCMLIST={(gcm1[, gcm2][,...])|DELETE}

文字配列テーブルと関連付けられる図形文字変更モジュールを 4 つまで指名します。GCMLIST が指定されているときに、すべての図形文字変更モジュール名フィールドは変更されます (1 つのモジュール名のみが指定されている場合、他の 3 つの名前フィールドはブランクに設定されます)。

gcmx

図形文字変更モジュールの 1~4 文字の名前です。4 つまでのモジュール名を指定できます。図形文字変更モジュールが現在その名前で存在するかどうかにかかわらず、名前は文字配列テーブルに入れられます。ただし、モジュールが存在しない場合、IEBIMAGE は警告メッセージを出します。文字配列テーブルが参照しているすべての図形文字変更モジュールが、イメージ・ライブラリーに保管されていない限り、その文字配列テーブルは使用しないでください。

DELETE

すべての図形文字変更モジュール名フィールドがブランクに設定されることを指定します。

LOC=((xloc[, {cloc[,setno] | FF}][,...])

256 個の変換テーブル項目の一部または全部の値を指定します。それぞれの変換テーブル項目は、WCGM のいずれかにある、64 文字の 1 つの位置を識別します。

xloc

変換テーブルに入れる索引であり、'X'00~'X'FF の 16 進値として指定されます。xloc は、項目の内容としてではなく、変換テーブル項目として識別されます。

cloc

WCGM にある、64 文字の 1 つの位置を識別し、これが 'X'00~'X'3F 間の 16 進値として指定されます。cloc が指定されていないときに、デフォルトは 'X'FF' になりますが、これは無効な文字です。

setno

WCGM の 1 つを識別し、0~3 の 10 進整数として指定されます。setno が指定されていないときは、デフォルトは 0 です。setno は、cloc も指定されている時以外は指定できません。

cloc および *setno* は、*xloc* によって配置された変換テーブルの内容を指定します。同じ *cloc* および *setno* の値を、複数の *xloc* に指定できます。

GRAPHIC ステートメント

GRAPHIC ステートメントは、図形文字変更モジュールの 1 つまたは複数の文字セグメントの内容を指定します。図形文字変更モジュールは、ヘッダー情報とそれに続く 1~64 の文字セグメントによって構成されています。それぞれの文字セグメントには、次のものが含まれています。

- 文字の 8 ビット・データ・コード、走査パターン、ピッチ (3800 モデル 1 の場合)
- 6 バイトの説明情報、120 バイトの走査パターン (3800 モデル 3 の場合)

INCLUDE ステートメントを使用することにより、モジュール全体から、DELSEG キーワードで削除したセグメントを抜いたものをコピーできます。さらに、GRAPHIC ステートメントにある GCM キーワードを使用して指名したモジュールから文字セグメントを選択できます。GRAPHIC ステートメントはさらに、新しい文字の走査パターンと特性を指定することもできます。

GRAPHIC ステートメントは常に、NAME ステートメント、別の GRAPHIC ステートメント、または 1 つまたは複数のデータ・ステートメントの後に指定しなければなりません。図形文字変更モジュールを 3800 モデル 3 互換モード・モジュール・フォーマットで作成するには、DEVICE パラメーターが指定されている OPTION ステートメントを GRAPHIC ステートメントの前に指定する必要があります。GRAPHIC ステートメントの前に INCLUDE ステートメントを置くこともできます。操作グループ内に複数の GRAPHIC ステートメントをコーディングできます。操作グループに、既存のモジュールから文字を選択する GRAPHIC ステートメント、および新しい文字を作成する GRAPHIC ステートメントを含めることができます。INCLUDE ステートメントの前にある GRAPHIC ステートメントは、1 つまたは複数のセグメントを、既存のモジュールのコピーから削除して、新しいモジュールを作成するときに使用できます。

オペランドが指定されていない GRAPHIC ステートメントで、その後に、図形文字変更モジュールを識別する NAME ステートメントが続く場合は、モジュールがフォーマットされて印刷されます。3800 モデル 3 の場合に図形文字変更モジュールを印刷するように指定するときは、OPTION ステートメントに DEVICE パラメーターを指定して、システムから正しい接頭部 (GRF2) を図形文字変更モジュール名に割り当てるようにしなければなりません。

別の図形文字変更モジュールから文字セグメントを選択するときに使用される GRAPHIC ステートメントの構文は次のとおりです。

[label]	GRAPHIC	[REF=((segno[,xloc])[,(segno[,xloc])][,...]) [,GCM=name]]
---------	---------	--------------------------------------------------------------

各項目の意味は以下のとおりです。

REF=((segno[,xloc]) [(segno[,xloc])] [...])

既存の図形文字変更モジュールにある、1 つまたは複数の文字セグメントを識別します。それぞれの文字セグメントには、文字の走査パターン、および 6 バ

イトの説明情報 (変換テーブル項目を見つけるときに使用する) が含まれていません。6 バイトの説明情報は、*xloc* サブパラメーターを使用して、再指定できます。REF パラメーターは、文字のピッチまたは走査パターンを変更するために使用することはできません。

segno

セグメント番号であり、1~999 の 10 進整数です。文字セグメントを IBM 提供の World Trade National Use Graphics 図形文字変更モジュールからコピーする場合は、*segno* を 64 より大きくすることができます。

IEBIMAGE プログラムを使用して、文字セグメントを図形文字変更モジュールからコピーする場合は、*segno* が 1~64 までの数になります。

xloc

文字の 8 ビット・データ・コードを指定します。X'00'~X'FF' の間の任意の値を使用できます。*xloc* は、必ず WCGM の文字位置を指す変換テーブル項目を識別するようにしてください (つまり、変換テーブル項目に X'FF' は含まれません)。*xloc* が指定されていない場合、セグメントがコピーされるときに、文字の 8 ビット・データ・コードは未変更のままになります。

REF パラメーターは、ASSIGN パラメーターを含む GRAPHIC ステートメントでコーディングできます。

GCM=name

REF パラメーターがコーディングされているときに指定できます。REF パラメーターによって参照されている文字セグメントを含む、図形文字変更モジュールを識別します。

name

図形文字変更モジュールの 1~4 文字のユーザー指定名を指定します。

GCM をコーディングした場合は、REF もコーディングしなければなりません。

GCM がコーディングされないときは、セグメントは IBM 提供の World Trade National Use Graphics 図形文字変更モジュールからコピーされます。

走査パターンと新しく作成した文字の特性を指定するときの GRAPHIC ステートメントの構文は次のとおりです。

[label]	GRAPHIC	ASSIGN=(<i>xloc</i>[,<i>pitch</i>]) <i>data statements</i> SEQ=<i>m</i>
---------	----------------	------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

ASSIGN=(*xloc*[,*pitch*])

新しく作成した文字およびその特性を識別します。ASSIGN パラメーターは、新しい文字の 8 ビット・データ・コードおよびそのピッチを指定します。

IEBIMAGE が ASSIGN パラメーターを検出すると、続くすべてのステートメント (列 25~28 に文字 **SEQ=** がないステートメントまで) が、文字の走査パターンを指定するデータ・ステートメントであると想定されます。

xloc

文字の 8 ビット・データ・コードを指定します。X'00'~X'FF' の間の任意

の値を使用できます。 *xloc* は、必ず WCGM の文字位置を指す変換テーブル項目を識別するようにしてください (つまり、変換テーブル項目に X'FF' は含まれません)。 *xloc* は、ASSIGN をコーディングするときが必要です。

pitch

文字の水平サイズを指定します。10 進整数の 10、12、15 のいずれかになります。 *pitch* が指定されていない場合は、デフォルトは 10 になります。

少なくとも 1 つのデータ・ステートメントが、ASSIGN パラメーターを含む GRAPHIC ステートメントの後に続いていなければなりません。

data statements

文字デザイン用紙で表されている文字のデザインを記述します。文字のデザイン方法の詳細、および文字デザイン用紙の使用方法については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。

各データ・ステートメントは、デザイン用紙の 1 行を表しています。デザイン用紙の各非ブランク行は、データ・ステートメントによって表されていなければなりません。さらにブランク行もデータ・ステートメントによって表されていなければなりません。最高で 24 個 (3800 モデル 1 の場合) または 40 個 (3800 モデル 3 の場合) のデータ・ステートメントをコーディングして、新しい文字のパターンを表すことができます。

各ステートメントでは、文字が 10 ピッチのときに、1~18 桁目 (モデル 1 の場合) または 1~24 桁目 (モデル 3 の場合) に非ブランク・グリッド位置を含めることができます。非ブランク文字はいずれも、非ブランク・グリッド位置を示す各列に穿孔することができます。文字が 12 ピッチのときに、1~15 桁目 (モデル 1 の場合) または 1~20 桁目 (モデル 3 の場合) に非ブランク・グリッド位置を含めることができます。文字が 15 ピッチのときに、1~15 桁目 (モデル 1 の場合) または 1~16 桁目 (モデル 3 の場合) に非ブランク・グリッド位置を含めることができます。

SEQ=nn

データ・ステートメントの列 25~30 に入れなければならないシーケンス番号を指定し、その行をデータ・ステートメントとして識別します。 *nn* は、行番号を指定します (文字デザイン用紙の行と対応する)。その行番号は、01~40 の 2 桁の 10 進数になります。

CHARSET ステートメント

CHARSET ステートメントは、ライブラリー文字セット・モジュールの 1 つまたは複数の文字セグメントの内容を指定します。ライブラリー文字セット・モジュールは、ヘッダー情報とそれに続く 64 の文字セグメントによって構成されています。それぞれの文字セグメントには、WCGM 位置のための文字の 6 ビット・コード、走査パターン、およびそのピッチが含まれています。INCLUDE ステートメントを使用することにより、モジュール全体から、DELSEG キーワードで削除したセグメントを抜いたものをコピーできます。さらに、CHARSET ステートメントを使用して、ライブラリー文字セット ID または GCM キーワードを使用して指名したモジュールから文字セグメントを選択できます。CHARSET ステートメントはさらに、新しい文字の走査パターンと特性を指定することもできます。

CHARSET ステートメントは常に、NAME ステートメント、別の CHARSET ステートメント、または 1 つまたは複数のデータ・ステートメントの後に指定しなければなりません。3800 モデル 3 互換モード・モジュール・フォーマットでライブラリー文字セット・モジュールを作成したい場合は、CHARSET ステートメントの前に、DEVICE パラメーターが指定されている OPTION ステートメントが必要です。CHARSET ステートメントの前に INCLUDE ステートメントを置くこともできます。操作グループ内に複数の CHARSET ステートメントをコーディングできます。操作グループに、既存のモジュールから文字を選択する CHARSET ステートメント、および新しい文字を作成する CHARSET ステートメントを含めることができます。INCLUDE ステートメントの前にある CHARSET ステートメントは、1 つまたは複数のセグメントを、既存のモジュールのコピーから削除して、新しいモジュールを作成するときに使用できます。

オペランドが指定されていない CHARSET ステートメントで、その後に、ライブラリー文字セット・モジュールを識別する NAME ステートメントが続く場合は、モジュールがフォーマットされて印刷されます。

別のモジュールから文字セグメントを選択するときの CHARSET ステートメントの構文は次のとおりです。

[label]	CHARSET	[REF=((<i>segno,cloc</i>) [<i>,(segno,cloc)</i>][,...]) [,{GCM= <i>name</i> ID= <i>xx</i> }]]
---------	---------	------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

REF=((*segno,cloc*) [*,(segno,cloc)*][,...])

既存の図形文字変更モジュールまたはライブラリー文字セット・モジュールにある、1 つまたは複数の文字セグメントを識別します。GCM を参照している場合、参照されている文字の走査パターンが使用され、6 ビットの WCGM 位置コードが割り当てられます。ライブラリー文字セットにある文字を参照している場合、6 ビットの WCGM 位置コードを含むセグメント全体が使用されます。ただし、セグメントに *cloc* サブパラメーターが指定されている場合は例外です。REF パラメーターは、文字のピッチまたは走査パターンを変更するために使用することはできません。

segno

セグメント番号であり、1~999 の 10 進整数です。文字セグメントを IBM 提供の World Trade National Use Graphics 図形文字変更モジュールからコピーする場合は、*segno* を 64 より大きくすることができます。

IEBIMAGE プログラムを使用して、文字セグメントを図形文字変更モジュール、またはライブラリー文字セット・モジュールからコピーする場合は、*segno* が 1~64 までの数になります。

cloc

WCGM 位置を指す 6 ビット・データ・コードを指定します。X'00'~X'3F' の間の任意の値を使用できます。ライブラリー文字セット・セグメントが参照されているときに、*cloc* が指定されていない場合、セグメントがコピーされるときに、文字の 6 ビット・データ・コードは未変更のままになります。図形文字変更セグメントが参照されている場合は、*cloc* を指定する必要があります。

REF パラメーターは、ASSIGN パラメーターを含む CHARSET ステートメントでコーディングできます。

GCM=name

REF パラメーターがコーディングされているときに指定できます。REF パラメーターによって参照されている文字セグメントを含む、図形文字変更モジュールを識別します。

name

図形文字変更モジュールの 1~4 文字のユーザー指定名を指定します。

GCM をコーディングした場合は、REF もコーディングしなければなりません。GCM は、ID を使用してコーディングしないでください。

GCM または ID のどちらもコーディングしていない場合、セグメントは IBM 提供の World Trade National Use Graphics 図形文字変更モジュールからコピーされます。

ID=xx

REF パラメーターがコーディングされているときに指定できます。REF パラメーターによって参照されている文字セグメントを含む、ライブラリー文字セットを識別します。

xx ライブラリー文字セット・モジュールの、16 進数の 2 桁の ID を指定します。2 番目の数字は奇数でなければならず、'7F' および 'FF' は使用できません。

GCM では、ID をコーディングしないでください。

ID または GCM のどちらもコーディングしないときは、セグメントは IBM 提供の World Trade National Use Graphics 図形文字変更モジュールからコピーされます。

走査パターンと新しく作成した文字の特性を指定するときの CHARSET ステートメントの構文は次のとおりです。

[label]	CHARSET	ASSIGN=(cloc[,pitch]) <i>data statements SEQ=nn</i>
---------	----------------	---------------------------------------------------------------

各項目の意味は以下のとおりです。

ASSIGN=(cloc[,pitch])

新しく作成した文字およびその特性を識別します。ASSIGN パラメーターは、新しい文字の 6 ビット・コードおよびそのピッチを指定します。IEBIMAGE が ASSIGN パラメーターを検出すると、続くすべてのステートメント (列 25~28 に文字 **SEQ=** が不在ステートメントまで) が、文字の走査パターンを指定するデータ・ステートメントであると想定されます。

cloc

WCGM 位置を指す、文字の 6 ビット・データ・コードを指定します。X'00'~X'3F' の間の任意の値を使用できます。cloc は、ASSIGN をコーディングするときに必要です。

pitch

文字の水平サイズを指定します。10 進数 10、12、15 のいずれかになります。 *pitch* が指定されていない場合は、デフォルトは 10 になります。

少なくとも 1 つのデータ・ステートメントが、ASSIGN パラメーターを含む CHARSET ステートメントの後に続いていなければなりません。

data statements

文字デザイン用紙で表されている文字のデザインを記述します。文字のデザイン方法の詳細、および文字デザイン用紙の使用方法については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。

各データ・ステートメントは、デザイン用紙の 1 行を表しています。デザイン用紙の各非ブランク行は、データ・ステートメントによって表されていなければなりません。さらにブランク行もデータ・ステートメントによって表されていなければなりません。最高で 24 個 (3800 モデル 1 の場合) または 40 個 (3800 モデル 3 の場合) のデータ・ステートメントをコーディングして、新しい文字のパターンを表すことができます。

各ステートメントでは、文字が 10 ピッチのときに、1~18 桁目 (モデル 1 の場合) または 1~24 桁目 (モデル 3 の場合) に非ブランク・グリッド位置を含めることができます。非ブランク文字はいずれも、非ブラック・グリッド位置を示す各列に穿孔することができます。文字が 12 ピッチのときに、1~15 桁目 (モデル 1 の場合) または 1~20 桁目 (モデル 3 の場合) に非ブランク・グリッド位置を含めることができます。文字が 15 ピッチのときに、1~15 桁目 (モデル 1 の場合) または 1~16 桁目 (モデル 3 の場合) に非ブランク・グリッド位置を含めることができます。

SEQ=nn

データ・ステートメントの列 25~30 に入れなければならないシーケンス番号を指定し、その行をデータ・ステートメントとして識別します。nn は、行番号を指定します (文字デザイン用紙の行と対応する)。その行番号は、01~40 の 2 桁の 10 進数になります。

INCLUDE ステートメント

IEBIMAGE オペレーション・グループを使用して新しいモジュールを作成するときに、INCLUDE ステートメントは、新しいモジュールの基礎としてコピーされ、使用されるイメージ・ライブラリー・モジュールを識別します。イメージ・ライブラリー・モジュールを更新するために操作グループを使用するときに、INCLUDE ステートメントは、参照されるモジュールを識別するので、必ず指定しなければなりません。

- INCLUDE ステートメントを操作グループ内にコーディングする場合は、FCB、COPYMOD、TABLE、GRAPHIC、CHARSET ステートメントのいずれかよりも前に指定しなければなりません。
- 各操作グループにコーディングできる INCLUDE ステートメントは 1 つだけです。複数のステートメントをコーディングする場合は、最後のものだけが使用され、それ以外のものは無視されます。
- DEVICE=4248 パラメーターが OPTION ステートメントで指定されているときのみ、FCB モジュールのために INCLUDE ステートメントをコーディングできます。3211 フォーマットまたは 4248 フォーマットの FCB のいずれかを含め

ことができます。IEBIMAGE は、4248 フォーマットの FCB を最初に見つけようとしています。これが見つからない場合は、3211 フォーマットを探します。

- INCLUDE を使用して 3800 FCB モジュールをコピーすることはできません。

INCLUDE ステートメントの構文は次のとおりです。

[label]	INCLUDE	module name [,DELSEG=(segno[, segno][,...])]
---------	---------	-------------------------------------------------

各項目の意味は以下のとおりです。

module name

ライブラリー・モジュールを命名するか、または識別します。モジュール名は、1~4 個の英数字および各国語文字 (\$,#,@) を任意の順番で並べた名前になります。ライブラリー文字セット・モジュールの場合は、2 桁の 16 進数字 (0~9,A~F) を表す 2 文字 ID になります。2 番目の数字は奇数にしなければなりません。7F および FF は使用できないので注意してください。

3800 INCLUDE 操作では、指名されたモジュールのタイプは、作成されるモジュールのタイプと同じでなければなりません。

しかし、4248 印刷装置の場合、指名された FCB モジュールが接頭部 FCB4 のグループに存在していないことが判明すると、同じモジュール名の、既存の 3211 FCB モジュール (接頭部 FCB2) が使用されます。この場合、FCB ステートメントの LINES、SIZE、CHx、LPI パラメーターに指定された値は、新しい値が 3211 印刷装置と互換性がない場合、組み込まれているモジュールで以前に指定したデフォルトになります。3211 モジュールがデフォルト・イメージである場合、DEFAULT パラメーターが NO に指定されていない限り、4248 モジュールもデフォルト・イメージになります。

DELSEG=(segno[,segno][,...])

モジュールがコピーされるときに削除される、コピー先モジュールのセグメントを指定します。セグメント番号はどの順序で指定しても構いません。このパラメーターでは、モジュールの最初のセグメントを参照するのに、セグメント 1 を使用します。DELSEG パラメーターをコーディングするときに、現在のモジュールの内容のリストを使用して、必ず不必要なセグメントを正しく識別するようにしなければなりません。

DELSEG パラメーターをコーディングできるのは、指名されたモジュールが、コピー変更モジュール、図形文字変更モジュール、ライブラリー文字セット・モジュールのいずれかであるときに限られます。

NAME ステートメント

NAME ステートメントは、IEBIMAGE プログラムによって作成される新しいライブラリー・モジュールを命名します。NAME ステートメントでは、既存のライブラリー・モジュールの名前も指定できます。NAME ステートメントは必須であり、各操作グループの最後のステートメントでなければなりません。

NAME ステートメントの構文は次のとおりです。

[label]	NAME	module name[(R)]
---------	------	------------------

各項目の意味は以下のとおりです。

module name

ライブラリー・モジュールを命名するか、または識別します。モジュール名は、1~4 個の英数字および各国語文字 (\$,#,@) を任意の順番で並べた名前になります。ライブラリー文字セット・モジュールの場合は、2 桁の 16 進数字 (0~9,A~F) を表す 2 文字 ID になります。2 番目の数字は奇数にしなければなりません。7F および FF は使用できないので注意してください。

4248 FCB モジュールを作成する場合、3211 FCB モジュール (接頭部 FCB2) を INCLUDE ステートメントの入力として使用する場合でも、指定する名前の接頭部は FCB4 になります。IEBIMAGE を使用して、FCB2 モジュールを作成したり置換したりすることはできません。

(R)

このモジュールが、同じ名前の新しいモジュール (存在する場合) に置換されることを示します。R は、括弧で囲みます。

OPTION ステートメント

ライブラリー文字セット・モジュールと図形文字変更モジュールを、3800 モデル 3 で使用可能なフォーマットで作成するには、OPTION ステートメントに DEVICE=3800M3 パラメーターを指定する必要があります。DEVICE=3800M3 パラメーターが指定されている OPTION ステートメントは、コピー変更モジュールおよび文字配列テーブル・モジュールを作成する場合はオプションです。

3262 モデル 5 または 4248 印刷装置用の用紙制御バッファ・モジュールを作成するには、OPTION ステートメントに DEVICE=4248 パラメーターを指定する必要があります。DEVICE=4248 は、FCB 以外のモジュールを作成するときには使用できません。

OVERRUN パラメーターが指定されている OPTION ステートメントは、COPYMOD 操作グループでのみ使用され、そのグループの INCLUDE ステートメントの前にでも後にでも指定できます。OVERRUN パラメーターの値は、オーバーラン警告メッセージ IEBA33I を印刷する対象となっている行の最大密度を指定します。オーバーラン条件およびオーバーラン警告メッセージの抑止については、186 ページの『OVERRUN の使用』を参照してください。

OVERRUN パラメーターを指定した OPTION ステートメントを効果的に使用するには、まずコピー変更モジュールが使用される印刷行の一番大きい密度 (6,8,10,12) を判別し、それよりも大きい行密度に対する警告メッセージを除去するために、OVERRUN パラメーターで密度を指定します。

OPTION ステートメントは、後続の操作グループにのみ適用されます。OPTION ステートメントは、ジョブ入力ストリームの各操作グループごとに指定しなければなりません。

OPTION ステートメントの構文は次のとおりです。

[label]	OPTION	[DEVICE={3800M3 4248}] [,OVERRUN={0 6 8 10 12}]
---------	--------	----------------------------------------------------

各項目の意味は以下のとおりです。

DEVICE={3800M3|4248}

プリンター互換モードのモジュール・フォーマットおよび処理上の考慮事項を指定します。

3800M3

3800 モデル 3 互換性を指定します。

4248

FCB ステートメントによって作成または変更したモジュールを 3262 モデル 5 または 4248 印刷装置用にフォーマットするように指定します。4248 FCB モジュールの構文については、153 ページの図 26 を参照してください。

DEVICE パラメーターが省略されている場合、3800 モデル 1 用のモジュールが作成されます。

OVERRUN={0|6|8|10|12}

COPYMOD 操作のメッセージ IEBA33I が印刷される最大行数 / インチを指定します。たとえば、OVERRUN=8 は、6 および 8 行 / インチのメッセージは許可しますが、10 および 12 行 / インチのメッセージは抑止します。

OVERRUN=0 を指定すると、どのような場合でもメッセージ IEBA33I は抑止されます。OVERRUN=12 を指定すると、メッセージ抑止は行われません。

OVERRUN=10 が有効なのは、3800 モデル 3 の場合に限られます。

OPTION ステートメントが省略されている場合、OVERRUN パラメーターのデフォルト値は 12 であり、メッセージは抑止されません。OVERRUN パラメーターが省略される場合も、デフォルト値は 12 です。

パラメーターの指定が間違っている (たとえば、OVERRUN=16 が指定されている) 場合、操作グループ全体が正常には完了しません。

COPYMOD で OVERRUN パラメーターを使用するための詳細については、『OVERRUN の使用』を参照してください。

OVERRUN の使用

187 ページの表 28 は、コピー変更モジュールのセグメントのオーバーラン警告付きのリスト表示を示したものです。OPTION ステートメントで OVERRUN=0 を指定しており、オーバーラン警告メッセージが印刷されていない場合でも、オーバーランの可能性のあるセグメント記述の左に注釈が印刷されます。

行オーバーラン条件を決定するために使用されるエレメントは次のとおりです。

- 行ごとの変更数
- モジュールごとのセグメント数

COPYMOD セグメントを結合させると、行オーバーラン条件の可能性が少なくなります。

コピー変更モジュールによって行オーバーラン条件が発生するかどうかを見定めるための計算式については、*Reference Manual for IBM 3800 Printing Subsystem Model 1* を参照してください。

表 28. IEBIMAGE によるコピー変更モジュールのオーバーラン注釈付きのリスト表示

注釈	セグメント	コピー開始		行開始番号	行数	印刷開始	
		番号	コピー数			位置	文字数
注釈 (0) ¹	1	1	200	10	96	10	180
注釈 (1) ²	2	2	200	10	96	11	180
注釈 (1) ²	3	3	200	10	96	12	180
注釈 (2) ³	4	4	200	10	96	10	180
注釈 (2) ³	5	5	200	10	96	11	180
注釈 (3) ⁴	6	6	200	10	96	12	180
注釈 (3) ⁴	7	7	200	10	96	10	180
注釈 (3) ⁴	8	8	200	10	96	11	180
注釈 (3) ⁴	9	9	200	10	96	12	180

注:

1. 12 LPI で印刷する場合に、コピー変更オーバーランの可能性を示します。
2. 8 LPI で印刷する場合に、コピー変更オーバーランの可能性を示します。
3. 8 または 12 LPI で印刷する場合に、コピー変更オーバーランの可能性を示します。
4. 6、8 または 12 LPI で印刷する場合に、コピー変更オーバーランの可能性を示します。つまり、どの LPI でもオーバーランの可能性があるとということになります。

IEBIMAGE の例

以下の各例では、IEBIMAGE の使用方法の一部を示します。表 29 は 使用例の早見表として使うことができます。

通常は、IBM 3800 モデル 3 の例から OPTION DEVICE=3800M3 ステートメントを削除して、OVERRUN パラメーターを 10 以外の数に指定すると、IBM 3800 モデル 1 の例に変更できます。

表 29. IEBIMAGE の例の一覧表

作成するモジュール	プリンター	注釈	例
CHARSET	3800 モデル 1	走査パターンのあるライブラリー文字セット全体が印刷されま す。	19
CHARSET	3800 モデル 3	セグメントが IBM 提供の GRAPHIC モジュールからコピーされ ます。	20
CHARSET	3800 モデル 3	新しいモジュールにユーザー・デザイン文字が含まれます。既存 の文字配列 (TABLE) は、新しい文字を含めるために修正されま す。	21
CHARSET	3800 モデル 1	セグメントが既存のモジュールからコピーされます。ユーザー・ デザイン文字が作成されます。	22
COPYMOD	3800 モデル 1	4 つの変更セグメント。	8
COPYMOD	3800 モデル 3	既存のモジュールが、新しいモジュールの基礎として使用されま す。OVERRUN が指定されます。	9
FCB	3800 モデル 1	11 インチの用紙。	1

IEBIMAGE

表 29. IEBIMAGE の例の一覧表 (続き)

作成する モジュール	プリンター	注釈	例
FCB	3800 モデル 1	5-1/2 インチの用紙。既存の SYS1.IMAGELIB メンバーを置換します。複数のチャンネル・コードが指定されます。	2
FCB	3800 モデル 1	3-1/2 インチの用紙。既存の SYS1.IMAGELIB メンバーを置換します。いろいろな値の垂直行送り。	3
FCB	3800 モデル 1	7 インチの用紙。いろいろな値の垂直行送り。	4
FCB	3800 モデル 1	12 インチの ISO 用紙。IBM 提供のモジュールを置換します。	5
FCB	3800 モデル 3	7-1/2 インチの ISO 用紙。いろいろな値の垂直行送り。	6
FCB	4248	11 インチの用紙。既存のモジュールが基礎になります。新しい印刷速度およびコピー位置が指定されます。	7
GRAPHIC	3800 モデル 1	IBM 提供のモジュール全体が印刷されます。	14
GRAPHIC	3800 モデル 3	セグメントが IBM 提供のモジュールからコピーされます。	15
GRAPHIC	3800 モデル 3	新しいモジュールにユーザー・デザイン文字が含まれます。既存の文字配列 (TABLE) は、新しい文字を含めるために修正されます。	16
GRAPHIC	3800 モデル 1	セグメントが既存のモジュールからコピーされます。ユーザー・デザイン文字が作成されます。	17
GRAPHIC	3800 モデル 3	新しい GRAPHIC モジュールにユーザー・デザイン文字が含まれます。既存の文字配列 (TABLE) は、新しい文字を含めるために修正されます。COPYMOD が、新しい文字を印刷するために作成されます。結果がテストされます。	18
TABLE	3800 モデル 3	IBM 提供のモジュールが、別の文字を組み込むために変更されます。	10
TABLE	3800 モデル 3	既存のモジュールが、新しいモジュールの基礎として使用されます。ピッチが変更されます。	11
TABLE	3800 モデル 1	既存のモジュールが、新しいモジュールの基礎として使用されます。GRAPHIC モジュールのユーザー・デザイン文字を含めます。	12
TABLE	3800 モデル 3	既存のモジュールが、新しいモジュールの基礎として使用されます。新しいモジュールは、すべての GRAPHIC 参照を削除し、変換テーブル項目をリセットします。	13

例 1: 新しい 3800 用紙制御バッファ・モジュールを作成する

3800 モデル 1

この例では、11 インチの用紙の垂直行送りおよびチャンネル・コードを指定し、SYS1.IMAGELIB データ・セットにモジュールを新しいメンバーとして追加します。

```
//FCBMOD1 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
FCB CH1=1,CH12=80,LPI=8
NAME IJ
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- CH1=1 は、行 1 にチャンネル 1 コードを指定しているため、行 1 で位置決めが行われます。
- CH12=80 は、行 80 にチャンネル 12 コードを指定しているため、行 80 で位置決めが行われ、行 80 (そのページの最後の印刷可能行) で装置例外指示が表示されます。
- LPI=8 は、用紙全体を 8 行 / インチの垂直行送りに指定しています。SIZE パラメーターが省略されているため、用紙の長さは、デフォルトの 11 インチになります。11 インチの用紙には、印刷可能スペースが 10 インチあるため、8 行 / インチで 80 行が印刷されます。
- 新しい FCB モジュールの名前は IJ です。これは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 2: 3800 用紙制御バッファ・モジュールを置換する

3800 モデル 1

この例では、5-1/2 インチの用紙のサイズおよびチャンネル・コードを指定し、SYS1.IMAGELIB データ・セットにモジュールを既存のメンバーの置換として追加します。新しいモジュールはデータ・セットの最後に追加されます。データ・セットのディレクトリーにある名前は、新しいモジュールを指すように更新されます。古いモジュールには、データ・セットのディレクトリーを介してアクセスすることはできなくなります。

```
//FCBMOD2 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
FCB CH1=(1,7,13,20),CH12=26,SIZE=55
NAME S55(R)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。

- CH1=(1,7,13,20) は、印刷可能行 1、7、13、20 のチャンネル 1 コードを指定します。
- CH12=26 は、印刷可能行 26 のチャンネル 12 コードを指定します。
- SIZE=55 は、用紙の長さを 1 インチの 10 分の 55、つまり 5-1/2 インチに指定します。
- LPI パラメーターが省略されているため、垂直行送りは、デフォルトの 6 行 / インチになります。5-1/2 インチの用紙には、印刷可能行が 4-1/2 インチあるため、この用紙では 27 の印刷行があります。
- FCB モジュールの名前は S55 であり、それは同じ名前を持つ既存の FCB モジュールを置換します。新しい FCB モジュールは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 3: 3800 用紙制御バッファ・モジュールを置換する

3800 モデル 1

この例では、用紙の垂直行送りおよびチャンネル・コードを指定し、SYS1.IMAGELIB データ・セットにモジュールを既存のメンバーの置換として追加します。新しいモジュールはデータ・セットの最後に追加されます。データ・セットのディレクトリーにある名前は、新しいモジュールを指すように更新されます。古いモジュールには、データ・セットのディレクトリーを介してアクセスすることはできなくなります。

```
//FCBMOD3 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
FCB CH1=1,CH2=4,CH5=11,SIZE=35,LPI=((6,2),(8,3),(6,4),(8,9))
NAME HL(R)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- CH1=1 は、印刷可能行 1 のチャンネル 1 コードを指定します。
- CH2=4 は、行 4 のチャンネル 2 コードを指定します。
- CH5=11 は、行 11 のチャンネル 5 コードを指定します。
- LPI=((6,2),(8,3),(6,4),(8,9)) は、用紙の最初の 18 印刷可能行の垂直行送りを指定します。
 - 行 1、2 を指定する (6,2) は、6 行 / インチの垂直行送りとなり、2/6 インチを使用します。
 - 行 3~5 を指定する (8,3) は、8 行 / インチの垂直行送りとなり、3/8 インチを使用します。
 - 行 6~9 を指定する (6,4) は、6 行 / インチの垂直行送りとなり、4/6 インチを使用します。
 - 行 10~18 を指定する (8,9) は、8 行 / インチの垂直行送りとなり、1-1/8 インチを使用します。

- SIZE=35 は、用紙の長さを 1 インチの 10 分の 35、つまり 3-1/2 インチに指定します。3-1/2 インチの用紙には、2-1/2 インチの印刷可能スペースがあり、LPI パラメーターは 2-1/2 インチの垂直行送りを指定するため、用紙のすべての行の垂直行送りが可能です。
- FCB モジュールの名前は HL であり、それは同じ名前を持つ既存のモジュールを置換します。新しい FCB モジュールは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 4: 新しい 3800 用紙制御バッファ・モジュールを作成する

3800 モデル 1

この例では、垂直行送り、チャンネル・コード、用紙の長さを指定し、SYS1.IMAGELIB データ・セットにモジュールを新しいメンバーとして追加します。

```
//FCBMOD4 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
FCB CH1=1,CH6=33,SIZE=70,LPI=((8,32),(12,2))
NAME TGT
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- CH1=1 は、印刷可能行 1 のチャンネル 1 コードを指定します。
- CH6=33 は、行 33 のチャンネル 6 コードを指定します。
- LPI=((8,32),(12,2)) は、用紙の最初の 32 印刷可能行を 8 行 / インチの垂直行送り、続く 2 つの印刷可能行を 12 行 / インチの垂直行送りに指定しています。
- SIZE=70 は、用紙の長さを 1 インチの 10 分の 70、つまり 7 インチに指定します。7 インチの用紙に、6 インチの印刷可能行があり、LPI パラメーターが 8 行 / インチ (つまり 4 インチで 32 行) と、12 行 / インチ (つまり 1/6 インチで 2 行) を指定するため、残りの 1-5/6 インチの垂直行送りは、デフォルトの 6 行 / インチになります。

したがって、用紙の 1~32 行は 8 行 / インチ、33、34 行は 12 行 / インチ、35~45 行は 6 行 / インチとなり、行 1 と行 33 にチャンネル・コードが入ります。

- 新しい FCB モジュールの名前は TGT です。これは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 5: 3800 用紙制御バッファ・モジュール STD3 を置換する

3800 モデル 1

この例では、ISO 用紙サイズを使用する FCB モジュールを定義し、STD3 という名前の IBM 提供のモジュールと置換します。このことは、ダンプ・フォーマット・ルーチンの前に行う必要があります。その場合、印刷高密度ダンプがそのプリンターでルーチンを 8 行 / インチで印刷できます。

```
//FCBMOD5 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
FCB CH1=1,CH12=88,LPI=(8,88),SIZE=120
NAME STD3(R)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- CH1=1 は、印刷可能行 1 の チャンネル 1 コードを指定します。CH12=88 は、行 88 のチャンネル 12 コードを指定します。
- LPI=(8,88) は、用紙のすべての 88 印刷可能行を 8 行 / インチの垂直行送りに指定しています。
- SIZE=120 は、用紙の長さを 1 インチの 10 分の 120、つまり 12 インチ (最大の ISO 用紙サイズ) に指定します。
- 新しい FCB モジュールの名前は STD3 です。これは、SYS1.IMAGELIB にある同じ名前のモジュールと置換されます。

例 6: その他の ISO 用紙サイズ用に新しい 3800 用紙制御バッファ・モジュールを作成する

3800 モデル 3

この例では、ISO 用紙サイズ追加機能がインストールされている FCB モジュールで、ISO 用紙サイズを使用するように定義します。

```
//FCBMOD JOB ... 72
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
FCB CH1=1,CH12=75,SIZE=85,
LPI=((10,35),(12,4),(10,35),(6,1)) X
NAME ARU
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- CH1=1 は、行 1 にチャンネル 1 コードを指定しているため、行 1 で位置決めが行われます。
- CH12=75 は、行 75 にチャンネル 12 コードを指定しているため、行 75 で位置決めが行われ、行 75 (そのページの最後の印刷可能行) で装置例外指示が表示されます。
- LPI=((10,35),(12,4),(10,35),(6,1)) は、用紙上の印刷可能域の垂直行送りを指定しています。用紙の最後の印刷可能行には、6 行 / インチの垂直行送りがなければなりません。割り振られている行の合計は、1/2 の倍数でなければなりません。

例

$$(10,35)=3 \frac{1}{2} \quad (12,4)=\frac{2}{6} \quad (6,1)=\frac{1}{6}$$

$$3 \frac{1}{2} + \frac{2}{6} + 3 \frac{1}{2} + \frac{1}{6} = 7 \frac{1}{2} \text{ (1/2 の倍数)}$$

- SIZE=85 は、用紙の長さを 1 インチの 10 分の 85 つまり 8-1/2 インチに指定します。なお印刷可能域は 7-1/2 インチです。
- 新しい FCB モジュールの名前は ARU です。これは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 7: 4248 用紙制御バッファ・モジュールを作成する

この例では、既存の FCB モジュールをモデルとして、新しい 4248 デフォルト FCB モジュールを作成します。新しいモジュール NEW1 は、新規メンバーとして SYS1.IMAGELIB に追加されます。既存のモジュール OLD1 は未変更のまま残ります。OLD1 は、4248 FCB の呼び出し先 FCB4OLD1、または 3211 FCB の呼び出し先 FCB2OLD1 のいずれかになります。(両方のモジュールが存在する場合、FCB4OLD1 が使用されます。)

```
//FCBMOD7 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        OPTION DEVICE=4248
        INCLUDE OLD1
        FCB COPYP=67,PSPEED=M,DEFAULT=YES
        NAME NEW1
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- OPTION ステートメントの DEVICE=4248 は、4248 印刷装置のためにこのモジュールを作成することを指定します。
- INCLUDE ステートメントは、既存のモジュール OLD1 のコピーが新しいモジュール NEW1 の基礎として使用されることを指定します。
- COPYP=67 は、水平コピー機能が活動化され、さらにその水平コピーが左余白の 67 番目の印刷位置から印刷を開始しなければならないことを示しています。この設定は、モジュール OLD1 ですでに設定されている COPYP 値をすべて指定変更します。これはモジュール NEW1 に適用されますが、OLD1 で設定されている値は変更されません。

値 67 は、132 ハンマー印刷装置を、2 つの同じサイズの水平コピー用の 2 つの同等コピー域に分割します。COPYP=67 を指定すると、最大 66 バイトまでをプリンターに送信できます。

- PSPEED=M は、プリンター速度を中 (3000 LPM) に設定することを示しています。この設定は、モジュール OLD1 ですでに設定されている PSPEED 値をすべて指定変更します。これはモジュール NEW1 に適用されますが、OLD1 で設定されている値は変更されません。
- DEFAULT=YES は、このモジュール NEW1 がこのシステムのデフォルトの FCB モジュールになることを示しています。

- これらのパラメーターが指定されていないため、LINES、SIZE、LPI、CHx は、デフォルトとして、すでにモジュール OLD1 に存在している値になります。
- NAME ステートメントは、このモジュールの名前を NEW1 と指定しています。

例 8: 新しいコピー変更モジュールを作成する

3800 モデル 1

この例では、4 つの変更セグメントがあるコピー変更モジュールを作成します。このモジュールは、新規メンバーとして SYS1.IMAGELIB データ・セットに追加されます。

```

//COPMOD1 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
COPY1 COPYMOD COPIES=(1,1),LINES=(1,1),POS=50, X
                TEXT=(C,'CONTROLLER'S COPY')
COPY2A COPYMOD COPIES=(2,1),LINES=(1,1),POS=50, X
                TEXT=(C,'SHIPPING MANAGER'S COPY')
COPY2B COPYMOD COPIES=(2,1),LINES=(34,3),POS=75, X
                TEXT=(10C,' ')
COPYALL COPYMOD COPIES=(1,4),LINES=(58,1),POS=35, X
                TEXT=((C,'***'),(C,'CONFIDENTIAL'),(3X,'5C'))
                NAME RT01
/*

```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **COPY1 COPYMOD** ステートメントは、出力データ・セットの最初のコピーの各ページに適用されるテキストを指定しています。

LINES=(1,1) および **POS=50** は、テキストが各ページの最初の印刷可能行に入れられることを示しています。始まりは、左から 50 番目の印刷位置になります。

TEXT パラメーターは、コピーのそれぞれのページを『Controller's Copy』として識別しています。

- **COPY2A COPYMOD** ステートメントは、出力データ・セットの 2 番目のコピーの各ページに適用されるテキストを指定しています。テキストは、各ページの最初の行の左から 50 番目の印刷位置に入れられます。コピーの各ページは『Shipping Manager's Copy』になります。
- **COPY2B COPYMOD** ステートメントは、2 番目のコピーの出力データ・セット・テキストの一部をブランクにして、最初、3 番目、および後続のコピーに、2 番目のコピーにはない情報を含めるように指定します。ブランク域は、行 34、35、36 の左から 75 番目の印刷位置から始まります。印刷位置 75~84 の間の行 34、35、36 にあるテキスト（つまり、**TEXT** パラメーターの単一引用符で囲まれた文字）がブランクになります。
- **COPYALL COPYMOD** ステートメントは、出力データ・セットの最初の 4 つのコピーに適用されるテキストを指定しています。この例では、出力データ・セットを生成するジョブを処理するときに、5 つ以上のコピーが印刷されないことを前提としています。テキストは、各ページの 58 行目の左から 35 番目の印刷位

置に行に入れられます。『***CONFIDENTIAL***』が、コピーの各ページに入れられます。テキストは、文字および 16 進フォーマットの両方でコーディングできることに注意してください。

- コピー変更モジュールの名前は RTO1 です。これは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 9: 既存のコピー変更モジュールから新しいコピー変更モジュールを作成する

3800 モデル 3

この例では、既存のコピー変更モジュールのコピー RTO1 を、新しいコピー変更モジュールの基礎として使用します。新しいモジュールは、新規メンバーとして SYS1.IMAGELIB データ・セットに追加されます。既存のモジュール RTO1 は、未変更のまま残り、そのまま使用できます。

72

```
//COPMOD2 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE RTO1,DELSEG=1
OPTION OVERRUN=8,DEVICE=3800M3
COPYMOD COPIES=(2,3),LINES=(52,6),POS=100, X
        TEXT=(X,'40404040404040405C5C')
NAME AP
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **INCLUDE** ステートメントは、**RTO1** という名前のコピー変更モジュールのコピーが新しいモジュールの基礎として使用され、**RTO1** の最初の変更セグメントがコピーから削除されることを指定しています。
- **OPTION** ステートメントの **OVERRUN=8** は、6 および 8 行 / インチで印刷しているときに、コピー変更によって行オーバーラン条件が発生した場合に、IEBIMAGE プログラムが警告メッセージを印刷するように指定しています。このプログラムでは、10 および 12 行 / インチでの印刷に適用される警告メッセージを抑制しています。**OPTION** ステートメントの **DEVICE=3800M3** は、3800 モデル 3 互換モード処理を指定しています。
- **COPYMOD** ステートメントは、出力データ・セットの 2 番目、3 番目、4 番目のコピーの各ページに適用されるテキストを指定しています。

LINES=(52,6) および **POS=100** は、テキストが各ページの 52 行目に入れられ、53 行目から 57 行目で繰り返されるように指定します。始まりは、左から 100 番目の印刷位置になります。

TEXT ステートメントは、テキストを 16 進フォームで指定します。8 つのブランクの後に 2 つのアスタリスクが続きます (この例では、**X'40'** がブランクとして印刷され、**X'5C'** がアスタリスクとして印刷されることが前提となっていま

す。実際に実行するときは、コピー変更モジュールで使用される文字配列テーブルによって X'40' と X'5C' が他の印刷可能文字に変更されることがあります)。

- 新しいコピー変更モジュールの名前は AP です。これは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 10: 文字配列テーブル・モジュールに新しい文字を追加する

3800 モデル 3

この例では、IBM 提供の文字配列テーブル・モジュールを変更して別の文字を組み込んでから、それを SYS1.IMAGELIB データ・セットに追加して IBM 提供のモジュールと置き換えます。

```
//CHARMOD1 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE GF10
OPTION DEVICE=3800M3
TABLE LOC=((2A,2A),(6A,2A),(AA,2A),(EA,2A))
NAME GF10(R)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- INCLUDE ステートメントは、GF10 という名前の文字配列テーブルのコピーが新しいモジュールの基礎として使用されることを指定しています。
- DEVICE パラメーターが指定されている OPTION ステートメントは、3800 モデル 3 互換モード処理を指定しています。
- TABLE ステートメントは、4 つの変換テーブル項目 X'2A'、X'6A'、X'AA'、X'EA' の更新後の情報を指定しています。(これらの 4 つの位置は、IBM 提供の GF10 テーブルでは未使用になっています。) 4 つの変換テーブル項目のそれぞれは、最初の WCGM の '2A' (43 文字目) を指します (この WCGM には、ひし形の走査パターンが含まれています)。
- 文字配列テーブルの名前は GF10 であり、これは新しいモジュールとして SYS1.IMAGELIB データ・セットに保管されます。データ・セットのディレクトリーは、新しいモジュールを指すように更新されます。古い GF10 モジュールには、データ・セットのディレクトリーを介してアクセスすることはできなくなります。

例 11: 既存の文字配列テーブル・モジュールから新しい文字配列テーブル・モジュールを作成する

3800 モデル 3

この例では、既存の文字配列テーブル・モジュールをコピーして、新しいモジュールの基礎として使用します。新しい文字配列テーブルは、ゴシック 10 ピッチ文字セットの代わりに、ゴシック 15 ピッチ文字セットを使用している以外は、古いものと同じです。

```
//CHARMOD2 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE A11
OPTION DEVICE=3800M3
TABLE CGMID=87
NAME A115
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **INCLUDE** ステートメントは、**A11** という名前の文字配列テーブルのコピーが新しいモジュールの基礎として使用されることを指定しています。**A11** 文字配列テーブルは、8 ビット・データ・コードをゴシック 10 ピッチ文字セットの印刷可能文字に変換します。
- **DEVICE** パラメーターが指定されている **OPTION** ステートメントは、3800 モデル 3 互換モード処理を指定しています。
- **TABLE** ステートメントは、新しい文字セット **ID X'87'** を指定しています。これが、ゴシック 15 ピッチ文字セットの **ID** になります。文字配列テーブルでの他の変更は行われません。新しいテーブルでは、ゴシック 15 ピッチ文字セットの文字が必要になります。
- 新しい文字配列テーブルの名前は **A115** です。これは、**SYS1.IMAGELIB** データ・セットのメンバーとして保管されます。

例 12: 文字配列テーブル・モジュールの図形文字を作成する

3800 モデル 1

この例では、既存の文字配列テーブル・モジュールをコピーして、図形文字変更モジュールのユーザー・デザイン文字を含む新しいモジュールの基礎として使用します。その後、新しいモジュールは **SYS1.IMAGELIB** データ・セットに追加されます。

```
//CHARMOD3 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE ONB
TABLE GCMLIST=ONB1,LOC=((6F,2F,1),(7C,3C,1),(6A,2A,0))
NAME ONBZ
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **INCLUDE** ステートメントは、**ONB** という名前の文字配列テーブルのコピーが新しいモジュールの基礎として使用されることを指定しています。**ONB** は 2 つの **WCGM** を参照します。

- TABLE ステートメントは、図形文字変更モジュールを識別し、その各セグメントの変換テーブル項目を指定しています。

GCMLIST=ONB1 は、ONB1 という名前の図形文字変更モジュールを識別しています。LOC パラメーターは、変換テーブル項目位置、文字位置、およびそのモジュールの各セグメントの WCGM 番号を指定しています。

最初のセグメントは、8 ビット・データ・コード X'6F' に対応します。セグメントの走査パターンは、2 番目の WCGM の文字位置 X'2F' (つまり、48 文字目の位置) にロードされます。

2 番目のセグメントは、8 ビット・データ・コード X'7C' に対応します。セグメントの走査パターンは、2 番目の WCGM の文字位置 X'3C' (つまり、48 文字目の位置) にロードされます。

3 番目のセグメントは、8 ビット・データ・コード X'6A' に対応しています。セグメントの走査パターンは、最初の WCGM の文字位置 X'2A' (つまり、43 文字目の位置) にロードされます。

新しい文字配列テーブルの名前は ONBZ です。これは、SYS1.IMAGELIB データ・セットの新しいモジュールとして保管されます。

例 13: 文字配列テーブル・モジュールから図形参照を削除する

3800 モデル 3

この例では、既存の文字配列テーブル・モジュールをコピーして、新しいモジュールの基礎として使用します。新しい文字配列テーブルは、すべての図形文字変更モジュールの参照を削除し、図形文字変更モジュールのセグメントの文字位置を指すために使用された変換テーブル項目をリセットします。

```
//CHARMOD4 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE ZYL
OPTION DEVICE=3800M3
TABLE GCMLIST=DELETE,LOC=((6A),(6B))
NAME ZYLA
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- INCLUDE ステートメントは、ZYL という名前の文字配列テーブルのコピーが新しいモジュールの基礎として使用されることを指定しています。
- DEVICE パラメーターが指定されている OPTION ステートメントは、3800 モデル 3 互換モード処理を指定しています。
- TABLE ステートメントは、図形文字変更モジュールへの参照を削除し、2 つの変換テーブル項目をリセットします。

GCMLIST=DELETE は、ZYL 文字配列がコピーされたときにモジュールを使用して組み込まれた、すべての図形文字変更モジュールの名前がブランクにリセットされるように指定します (X'40')。

LOC パラメーターは、変換テーブルで、X'FF' (文字位置がないか、WCGM 値が指定されているときのデフォルト値) に設定されることになる 2 つの位置 X'6A' および X'6B' を識別します。

- 新しい文字配列テーブルの名前は ZYLA です。これは、SYS1.IMAGELIB データ・セットのメンバーとして保管されます。

例 14: World Trade National Use Graphics 図形文字変更モジュールをリスト表示する

3800 モデル 1

この例では、World Trade National Use Graphics が含まれている、IBM 提供の図形文字変更モジュールの各セグメントを印刷します。各セグメントは固有のもので、ただし、各セグメントの走査パターンは、8 ビット・データ・コードのみが異なるだけで、それ以外は同一の場合もあります。

```
//GRAFMOD1 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
GRAPHIC
NAME *
/*
```

各制御ステートメントについての説明は以下のとおりです。

- ライブラリーが更新されていないため、DISP=SHR をコーディングしています。
- World Trade National Use Graphics 図形文字変更モジュールは、偽名『*』によって識別されています。モジュール内の各文字の走査パターンが印刷されます。

例 15: 文字変更モジュール World Trade GRAFMOD から図形文字変更モジュールを作成する

3800 モデル 3

この例では、図形文字変更モジュールを作成します。World Trade National Use Graphics 図形文字変更モジュールからコピーされたセグメントが、そのモジュールの文字になります。(文字の EBCDIC 割り当てについては、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。)新しいモジュールは、SYS1.IMAGELIB システム・データ・セットに保管されます。

```
//GRAFMOD2 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
OPTION DEVICE=3800M3
GRAPHIC REF=((24),(25),(26),(27),(28),(31),(33),(35),(38),(40))
NAME CSTW
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。

- OPTION ステートメントにある DEVICE=3800M3 は、3800 モデル 3 互換モード・モジュール・フォーマットを指定しています。
- GCM キーワードを指定していないので、GRAPHIC ステートメントは、World Trade National Use Graphics 図形文字変更モジュールを指定します。10 個のセグメントがコピーされ、新しいモジュールで使用されます。
- 図形文字変更モジュールの名前は CSTW です。これは、SYS1.IMAGELIB データ・セットの新しいモジュールとして保管されます。

例 16: 新しい図形文字変更モジュールを作成して、文字配列テーブルを変更する

3800 モデル 3

この例では、図形文字変更モジュールを作成します。このモジュールには、ユーザー・デザイン文字である反転 'E' が含まれます。この文字の 8 ビット・データ・コードは X'E0' と指定されており、この文字のピッチは 10 です。既存の文字配列テーブルは、反転 E を含むように変更されます。

```
//GRAFMOD3 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        OPTION DEVICE=3800M3
        GRAPHIC ASSIGN=(E0,10)
        XXXXXXXXXXXXXXXX SEQ=10
        XXXXXXXXXXXXXXXX SEQ=11
        XXXXXXXXXXXXXXXX SEQ=12
        XXXX SEQ=13
        XXXX SEQ=14
        XXXX SEQ=15
        XXXX SEQ=16
        XXXX SEQ=17
        XXXX SEQ=18
        XXXX SEQ=19
        XXXXXXXXXXXXXXXX SEQ=20
        XXXXXXXXXXXXXXXX SEQ=21
        XXXXXXXXXXXXXXXX SEQ=22
        XXXX SEQ=23
        XXXX SEQ=24
        XXXX SEQ=25
        XXXX SEQ=26
        XXXX SEQ=27
        XXXX SEQ=28
        XXXX SEQ=29
        XXXXXXXXXXXXXXXX SEQ=30
        XXXXXXXXXXXXXXXX SEQ=31
        XXXXXXXXXXXXXXXX SEQ=32
        NAME BODE
        INCLUDE GS10
        OPTION DEVICE=3800M3
        TABLE CGMID=(83,FF),GCMLIST=BODE,LOC=(E0,03,1)
        NAME RE10
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。

- OPTION ステートメントにある DEVICE=3800M3 は、3800 モデル 3 互換モード処理を指定しています。
- GRAPHIC ステートメントの ASSIGN パラメーターは、ユーザー・デザイン文字用に、8 ビット・データ・コードとして X'E0'、幅として 10 ピッチを指定しています。GRAPHIC ステートメントに続く、データ・ステートメントは、文字の走査パターンを記述しています。
- 図形文字変更モジュールの名前は BODE です。これは、SYS1.IMAGELIB データ・セットの新しいモジュールとして保管されます。
- INCLUDE ステートメントは、GS10 文字配列テーブルのコピーが新しいテーブルの基礎として使用されることを指定しています。
- TABLE ステートメントは、GS10 テーブルのコピーに反転 E を追加することを指定しています。

CGMID=(83,FF) は、ゴシック 10 セット (このセットはすでに GS10 テーブルによって使用されている) の文字セット ID X'83' を指定し、さらに、X'FF' を文字セット ID として指定して、2 番目の WCGM をロードすることなく、そのアクセスを可能にしています。

GCMLIST=BODE は、テーブルに含めるための反転 E がある図形文字変更モジュールを識別しています。

LOC=(E0,03,1) は、8 ビット・データ・コード X'E0' が割り当てられている反転 E が、2 番目の WCGM の位置 X'03' にロードされることを指定しています。これ以外の場合に、この 2 番目の WCGM は使用されることがないため、反転 E は、この中のどの位置でも使用することができます。

- 新しい文字配列テーブルの名前は RE10 です。これは、SYS1.IMAGELIB データ・セットのモジュールとして保管されます。

例 17: 複数のソースから図形文字変更モジュールを作成する

3800 モデル 1

この例では、図形文字変更モジュールを作成します。この内容は、3 つの異なるソースからのものです。まず、INCLUDE ステートメントで、9 つのセグメントを既存のモジュールからコピーします。次に、GRAPHIC ステートメントで、コピーする別のセグメントを選択します。さらに、GRAPHIC で、ユーザー・デザイン文字の特性も設定します。新しい図形文字変更モジュールは、作成後に SYS1.IMAGELIB に追加されます。

```
//GRAFMOD4 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE CSTW,DELSEG=3
GRAPHIC REF=(1,6A),GCM=BODE,ASSIGN=9A
***** SEQ=06
***** SEQ=07
**** ** SEQ=08
*** ** SEQ=09
*** **** SEQ=10
*** ***** SEQ=11
*** ***** SEQ=12
```



```

***      ****      SEQ=13
***      ****      SEQ=14
***      ***       SEQ=15
***      ***       SEQ=16
*** ****  ****     SEQ=17
*** *****      SEQ=18
*** *****      SEQ=19
NAME  JPCK
/*

```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **INCLUDE** ステートメントは、**CSTW** という名前の図形文字変更モジュールのコピーを、新しいモジュールに含めることを指定しています。3番目のセグメント以外の **CSTW** のすべてのセグメント (**DELSEG=3** の結果) は、新しいモジュールにコピーされ、モジュールの最初から9番目までの変更セグメントになります。
- **GRAPHIC** ステートメントは、モジュールの10番目と11番目のセグメントを指定しています。

REF=(1,6A) および **GCM=BODE** は、新しいモジュールの10番目のセグメントを、**BODE** という名前の図形文字変更モジュールから最初のセグメントをコピーすることにより入手するように指定しています。さらに、セグメントの8ビット・データ・コードが変更され、その文字がコード **X'6A'** によって識別されるようになります。

ASSIGN=9A は、新しいモジュールの11番目のセグメントがユーザー・デザイン文字であることを指定しています。この8ビット・データ・コードは **X'9A'** であり、その幅は10ピッチです (これは、ピッチ値が指定されていないときのデフォルトです)。**GRAPHIC** ステートメントの後に、文字の走査パターンを指定するデータ・ステートメントが続きます。

- 図形文字変更モジュールの名前は **JPCK** です。これは、**SYS1.IMAGELIB** データ・セットの新しいモジュールとして保管されます。

例 18: 図形文字変更モジュールで文字を定義して使用する

3800 モデル 3

この例では、ユーザー・デザイン文字が含まれる図形文字変更モジュールを作成します。次に、フォーマット文字配列テーブルを変更して、その新しい文字を含めるようにします。その後、コピー変更モジュールを作成し、形式文字のボックスで囲まれている新しい文字を印刷します。最後に、出力と入力を比較するために結果をテストします。

```

//CHAR      JOB    ...
//BUILD     EXEC  PGM=IEBIMAGE
//SYSUT1    DD    DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
              OPTION  DEVICE=3800M3
STEP1      GRAPHIC ASSIGN=5C
XXX                XXX  SEQ=01
XXX                XXX  SEQ=02
XXX                XXX  SEQ=03

```



```

XXX                XXX SEQ=04
XXXXXXXXXXXXXXXXXXXX SEQ=05
XXXXXXXXXXXXXXXXXXXX SEQ=06
XXXXXXXXXXXXXXXXXXXX SEQ=07
XXX                XXX SEQ=08
XXX                XXX SEQ=09
XXX                XXX SEQ=10
XXX                XXX SEQ=11
                   SEQ=12
                   SEQ=13
                   SEQ=14
XXXXXXXXXXXXXXXXXXXX SEQ=15
XXXXXXXXXXXXXXXXXXXX SEQ=16
XXXXXXXXXXXXXXXXXXXX SEQ=17
XXX                XXX    XXX SEQ=18
XXX                XXX    XXX SEQ=19
XXX                XXX    XXX SEQ=20
XXX                XXX    XXX SEQ=21
XXXX               XXXX   XXXX SEQ=22
  XXXX             XXXXXX  XXXX SEQ=23
    XXXXXXXXXXXXXXXX SEQ=24
      XXXXX        XXXXXX  SEQ=25
                               SEQ=26
                               SEQ=27
                               SEQ=28
XXXXXXXXXXXXXXXXXXXX SEQ=29
XXXXXXXXXXXXXXXXXXXX SEQ=30
XXXXXXXXXXXXXXXXXXXX SEQ=31
                   XXXXXXX SEQ=32
                   XXXXXXX SEQ=33
XXXXXXXXXXXXXXXXXXXX SEQ=34
XXXXXXXXXXXXXXXXXXXX SEQ=35
                   XXXXXXX SEQ=36
                   XXXXXXX SEQ=37
XXXXXXXXXXXXXXXXXXXX SEQ=38
XXXXXXXXXXXXXXXXXXXX SEQ=39
XXXXXXXXXXXXXXXXXXXX SEQ=40
      NAME AIBM

STEP2  OPTION  DEVICE=3800M3
        INCLUDE FM10
        TABLE GCMLIST=AIBM,LOC=(5C,2C)
        NAME    BIBM

STEP3  OPTION  DEVICE=3800M3
        COPYMOD COPIES=1,LINES=58,POS=5,TEXT=(C,'W6X')
        COPYMOD COPIES=1,LINES=59,POS=5,TEXT=(C,'7*7')
        COPYMOD COPIES=1,LINES=60,POS=5,TEXT=(X,'E9F6E8')
        NAME    CIBM

/*
//TEST  EXEC  PGM=IEBIMAGE
//SYSUT1 DD  DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A,CHARS=(GF10,BIBM),
//        MODIFY=(CIBM,1)
//SYSIN  DD  *
          OPTION  DEVICE=3800M3
          GRAPHIC
          NAME    AIBM

/*

```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **GRAPHIC** ステートメントの **ASSIGN** パラメーターは、ユーザー・デザイン文字の 8 ビット・データ・コードが **X'5C'** であり、幅が 10 ピッチ (ピッチが指定されていないときのデフォルト) であることを指定しています。 **GRAPHIC** ステータス

トメントの後に、6行 / インチの垂直行送りのための、文字の走査パターンを指定するデータ・ステートメントが続きます。

- 図形文字変更モジュールの名前は AIBM です。これは、SYS1.IMAGELIB の新しいモジュールとして保管されます。
- STEP2 の INCLUDE ステートメントは、FM10 文字配列テーブルのコピーが新しいモジュールの基礎として使用されることを指定しています。
- TABLE ステートメントは、直前のステップで作成された AIBM という名前の図形文字変更モジュールを識別しています。TABLE ステートメントの LOC パラメーターは、変換テーブル項目位置 (文字の 8 ビット・データ・コード) X'5C' と、文字が WCGM にロードされるときの位置 (X'2C') を指定しています。
- 新しい文字配列テーブルの名前 (SYS1.IMAGELIB に追加される) は BIBM です。
- STEP3 の 3 つの COPYMOD ステートメントは、出力データ・セットの最初のコピーの行 58、59、60 に配置されるテキストを指定しています。始まりは、各行の印刷位置 5 になります。BIBM 文字配列テーブルで使用するとき、文字 W、6、X が左上部、水平線セグメント、右上部として、すべて行負荷 3 で印刷されます。文字 7、*、7 は、STEP1 で作成したユーザー・デザイン文字の両側に、負荷 3 の水平線セグメントとして印刷されます (アスタリスクの EBCDIC 割り当ては 5C です。これはその文字をアドレッシングしています)。16 進 E9、F6、E8 は、文字の回りの行負荷 3 フォーマットのボックスを完了させます。
- コピー変更モジュールの名前は CIBM です。これは、SYS1.IMAGELIB の新しいモジュールとして保管されます。
- TEST の EXEC ステートメントは、作成したばかりのモジュールをテストするために、IEBIMAGE プログラムをもう 1 度実行することを要求します。SYSPRINT DD ステートメントでは、BIBM 文字配列テーブルが、2 つ指定されたうちの 2 番目となり、CIBM コピー変更モジュールにテーブル参照文字 1 が指定されて、その BIBM テーブルが使用できるようになります。
- オペランドが指定されていない GRAPHIC ステートメントは、モジュール AIBM (その後に NAME ステートメントが指定されている) の印刷を要求します。この IEBIMAGE 実行の出力リスト表示の各ページの左下隅には、修正部分の小さなイメージが印刷されます。
- STEP1、STEP2、STEP3 の、DEVICE パラメーターが指定されている OPTION ステートメントは、3800 モデル 3 互換モード・モジュール形式と、処理上の考慮事項を指定しています。

例 19: ライブラリー文字セット・モジュールをリスト表示する

3800 モデル 1

この例では、ライブラリー文字セットの各セグメントを印刷します。モジュール内の各文字の走査パターンが印刷されます。

```
//LIBMOD1 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSNAME=SYS1.IMAGELIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
```

```
//SYSIN DD *
CHARSET
NAME 83
/*
```

各制御ステートメントについての説明は以下のとおりです。

- NAME は、ライブラリー文字セット (83) の名前を指定しています。

例 20: ライブラリー文字セット・モジュールを作成する

3800 モデル 3

この例では、ライブラリー文字セット・モジュールを作成します。World Trade National Use Graphics 図形文字変更モジュールからコピーされたセグメントが、そのモジュールの文字になります。このモジュールのすべてのセグメントのリスト表示については、*IBM 3800 Printing Subsystem Programmer's Guide* を参照してください。文字の EBCDIC 割り当ては、WCGM 位置コードによって置き換えられます。新しいモジュールは、SYS1.IMAGELIB システム・データ・セットに保管されます。

```
//LIBMOD2 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
OPTION DEVICE=3800M3
CHARSET REF=((24,01),(25,02),(26,03),(27,04),(28,05),
(31,06),(33,07),(35,08),(38,09),(40,0A)) X
NAME 73
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- OPTION ステートメントにある DEVICE=3800M3 は、3800 モデル 3 互換モード・モジュール・フォーマットを指定しています。
- GCM キーワードまたはライブラリー文字セット ID を設定していないので、CHARSET ステートメントは、World Trade National Use Graphics 図形文字変更モジュールを識別します。10 個のセグメントがコピーされ、新しいモジュールで使用されます。たとえば、24 番目のセグメントがコピーされ、WCGM 位置 01 が割り当てられます。REF パラメーター (24,01) を参照してください。
- ライブラリー文字セット・モジュールの名前は 73 です。これは、SYS1.IMAGELIB データ・セットの新しいモジュールとして保管されます。

例 21: ライブラリー文字セット・モジュールを作成して、文字配列テーブルを変更する

3800 モデル 3

この例では、ライブラリー文字セット・モジュールを作成します。このモジュールには、ユーザー・デザイン文字である反転 'E' が含まれます。この文字の 6 ビット

WCGM 位置コードは X'03' と指定されており、この文字のピッチは 10 です。既存の文字配列テーブルは、反転 E を含むように変更されます。

```
//LIBMOD3 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
OPTION DEVICE=3800M3
CHARSET ASSIGN=(03,10)
XXXXXXXXXXXXXXXX SEQ=10
XXXXXXXXXXXXXXXX SEQ=11
XXXXXXXXXXXXXXXX SEQ=12
XXXXXXXXXXXXXXXX SEQ=13
XXXXXXXXXXXXXXXX SEQ=14
XXXXXXXXXXXXXXXX SEQ=15
XXXXXXXXXXXXXXXX SEQ=16
XXXXXXXXXXXXXXXX SEQ=17
XXXXXXXXXXXXXXXX SEQ=18
XXXXXXXXXXXXXXXX SEQ=19
XXXXXXXXXXXXXXXX SEQ=20
XXXXXXXXXXXXXXXX SEQ=21
XXXXXXXXXXXXXXXX SEQ=22
XXXXXXXXXXXXXXXX SEQ=23
XXXXXXXXXXXXXXXX SEQ=24
XXXXXXXXXXXXXXXX SEQ=25
XXXXXXXXXXXXXXXX SEQ=26
XXXXXXXXXXXXXXXX SEQ=27
XXXXXXXXXXXXXXXX SEQ=28
XXXXXXXXXXXXXXXX SEQ=29
XXXXXXXXXXXXXXXX SEQ=30
XXXXXXXXXXXXXXXX SEQ=31
XXXXXXXXXXXXXXXX SEQ=32
NAME 73
INCLUDE GS10
OPTION DEVICE=3800M3
TABLE CGMID=(83,73),LOC=(E0,03,1)
NAME RE10
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** ステートメントには **DISP=OLD** が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- **OPTION** ステートメントにある **DEVICE=3800M3** は、3800 モデル 3 互換モード・モジュール・フォーマットと、処理上の考慮事項を指定しています。
- **CHARSET** ステートメントの **ASSIGN** パラメーターは、ユーザー・デザイン文字用に、6 ビット・データ・コードとして X'03'、幅として 10 ピッチを指定しています。CHARSET ステートメントに続くデータ・ステートメントは、文字の走査パターンを記述しています。
- ライブラリー文字セット・モジュールの名前は 73 です。これは、SYS1.IMAGELIB データ・セットの新しいモジュールとして保管されます。
- **INCLUDE** ステートメントは、GS10 文字配列テーブルのコピーが新しいテーブルの基礎として使用されることを指定しています。
- **TABLE** ステートメントは、GS10 テーブルのコピーに反転 E を含むライブラリー文字セットを追加することを指定しています。

CGMID=(83,73) は、ゴシック 10 セット (このセットはすでに GS10 テーブルによって使用されている) の文字セット ID X'83' を指定し、さらに、X'73' を文字セット ID として指定して、ライブラリー文字セット 73 の 2 番目の WCGM のロードを可能にしています。

LOC=(E0,03,1) は、2 番目の WCGM に WCGM 位置 03 を割り当てられている反転 E が、EBCDIC コード X'E0' によって参照されることを指定しています。

- 新しい文字配列テーブルの名前は RE10 です。これは、SYS1.IMAGELIB データ・セットのモジュールとして保管されます。

例 22 複数のソースからライブラリー文字セット・モジュールを作成する

3800 モデル 1

この例では、ライブラリー文字セット・モジュールを作成します。この内容は、3 つの異なるソースからのものです。まず、INCLUDE ステートメントで、62 個のセグメントを既存のモジュールからコピーされます。次に、CHARSET ステートメントで、コピーする別のセグメントを選択します。さらに、CHARSET で、ユーザー・デザイン文字の特性も設定します。新しいライブラリー文字セット・モジュールは、作成後に SYS1.IMAGELIB に追加されます。

```
//LIBMOD4 JOB ...
//STEP1 EXEC PGM=IEBIMAGE
//SYSUT1 DD DSN=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INCLUDE 33,DELSEG=(3,4)
CHARSET REF=(1,02),GCM=BODE,ASSIGN=03
***** SEQ=06
***** SEQ=07
**** SEQ=08
*** SEQ=09
*** SEQ=10
*** SEQ=11
*** SEQ=12
*** SEQ=13
*** SEQ=14
*** SEQ=15
*** SEQ=16
*** SEQ=17
*** SEQ=18
*** SEQ=19
NAME 53
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD ステートメントには DISP=OLD が含まれており、このジョブの実行中に、別のジョブがデータ・セットを修正できないようになっています。
- INCLUDE ステートメントは、33 という名前のライブラリー文字セット・モジュールのコピーを、新しいモジュールに含めることを指定しています。DELSEG=(3,4) の結果として、3 番目と 4 番目のセグメント以外の 33 のすべてのセグメントは、新しいモジュールにコピーされ、新しいモジュールの基本になります。

- CHARSET ステートメントは、モジュールの 3 番目および 4 番目のセグメントを指定しています。

REF=(1,02) および GCM=BODE は、新しいモジュールの 3 番目のセグメントを、BODE という名前の図形文字変更モジュールから最初のセグメントをコピーすることにより入手するように指定しています。セグメントの 6 ビット WCGM 位置コードが設定されるため、その文字はコード X'02' によって識別されます。

ASSIGN=03 は、新しいモジュールの 4 番目のセグメントがユーザー・デザイン文字であることを指定しています。この 6 ビット WCGM 位置コードは X'03' であり、その幅は 10 ピッチです (これは、ピッチ値が指定されていないときのデフォルトです)。CHARSET ステートメントの後に、文字の走査パターンを指定するデータ・ステートメントが続きます。

- ライブラリー文字セット・モジュールの名前は 53 です。これは、SYS1.IMAGELIB データ・セットの新しいモジュールとして保管されます。

第 8 章 IEBISAM プログラム

| IEBISAM プログラムは現在提供していません。 z/OS V1R7 から始まりますが、最
| 早、ISAM データ・セットは処理 (作成、オープン、コピー、またはダンプ) できな
| くなります。お客様でまだ使用状態にある ISAM データ・セットは、VSAM キー
| 順データ・セットに変換する必要があります。

| z/OS V1R7 よりも前では、アクセス方式サービスを使用して、 VSAM キー順デー
| タ・セットを割り振り、そこに ISAM データ・セットをコピーすることができまし
| た。アクセス方式サービスは、 VSAM キー順データ・セットを操作するためにも
| 使用できます。 ISAM データ・セットを VSAM キー順データ・セットに変換する
| ための情報については、 *z/OS DFSMS データ・セットの使用法* および *z/OS*
| *DFSMS カタログのためのアクセス方式サービス・プログラム* の REPRO コマンド
| を参照してください。

第 9 章 IEBTPCH (印刷 / 穿孔) プログラム

IEBTPCH は、順次データ・セット、区分データ・セット、PDSE の選択部分またはその全部を印刷または穿孔するのに使用します。さらに、データをディスクやテープに『印刷』したり、『穿孔』したりすることも可能です。

IEBTPCH を使用すれば、以下のものを印刷または穿孔できます。

- 順次データ・セット、区分データ・セット、PDSE の全体
- 区分データ・セットや PDSE の選択メンバー
- 順次データ・セット、区分データ・セット、PDSE の選択レコード
- 区分データ・セットや PDSE のディレクトリー
- 編集した順次データ・セット、区分データ・セット、PDSE
- 2 バイト文字セット (DBCS) データが含まれているデータ・セット

印刷または穿孔するレコード・フォーマットは、自分で指定することも、また IEBTPCH のデフォルト・フォーマットを使うこともできます。デフォルト・フォーマットは以下のとおりです。

- 各論理出力レコードは、新しい印刷行または穿孔カードで始まります。
- 各印刷行はそれぞれが 8 文字からなるいくつかのグループで構成されており、グループ間は 2 個のブランクで区切られています。1 つの印刷行に入れることのできるデータの文字数は最大 96 です。各穿孔カードには最大 80 バイトの情報を連続で入れることができます。
- 印刷不能文字はブランクとして出力されます。
- 入力データがブロック化されている場合、各論理出力レコードは『*』で、また各ブロックは『**』で区切られます。
- 1 ページ当たりの印刷行数は 60 です。

RECORD ユーティリティー制御ステートメントを使って独自のフォーマットを指定する場合は、そのレコード長が出力装置の印刷可能範囲を超えないようにしてください。

IEBTPCH にはオプションの編集機能も備っており、ラベルの処理をしたり入力レコードや出力レコードを操作したりするときに使用するルーチンへ制御を渡します。

データ・セット全体または選択メンバーの印刷と穿孔

IEBTPCH を使用すれば、順次データ・セット、区分データ・セット、PDSE の全体を印刷または穿孔することも、区分データ・セットや PDSE の選択メンバーのみを印刷または穿孔することも可能です。メンバーの選択は、MEMBER ユーティリティー制御ステートメントで行うことができます。

データ・セットは必要に応じて 16 進値で印刷できます。印刷しているデータ・セットにパック 10 進値が含まれている場合は、そのパック 10 進値をアンパック 10

進モードか 16 進モードに変換し、すべての文字が確実に印刷できるようにしなければなりません。データの変換は、RECORD ユーティリティー制御ステートメントを使って指定できます。

編集したデータ・セットの印刷と穿孔

IEBTPCH を使用すれば、編集した順次データ・セット、区分データ・セット、PDSE を印刷または穿孔することができます。ユーティリティー制御ステートメントを使うことにより、指定した編集情報をレコード、レコードのグループの 1 つ、レコードの複数の選択グループ、メンバーやデータ・セットの全体などに適用できます。1 行に印刷できる文字数は最大 144 文字です。

編集したデータ・セットは以下のような場合に生成されます。

- レコード内の定義済みデータ・フィールドの配列を変更するか、省略する場合
- データをパック 10 進値からアンパック 10 進値に、あるいは英数字表示から 16 進表示に変換する場合

2 バイト文字セット (DBCS) データの印刷と穿孔

IEBTPCH を使用すれば、2 バイト文字セット (DBCS) データを含んでいるデータ・セットの印刷や穿孔を行うことができます。2 バイト文字セットは、標準の 1 バイト文字セット (SBCS) では表現しきれない言語を表示するのに使われます。たとえば、日本語を表示するには 2 バイト文字セットが必要です。DBCS データを処理する必要があることを示すには、PRINT または PUNCH ステートメントに DBCS=YES パラメーターをコーディングします。

2 バイト文字のストリングは、シフトアウト (<) 文字とシフトイン (>) 文字で囲むことによって識別されます。IEBTPCH はシフトアウト文字を検出した時点でそのデータが 1 バイト文字セットのストリングから『シフトアウト』したものと理解し、シフトイン文字を検出した時点で 1 バイト文字セットに『シフトイン』したものと理解します。

2 バイト文字の各バイトは X'41'~X'FE' の範囲内になければなりません。あるいは、その DBCS 文字が DBCS スペース (X'4040') でなければなりません。IEBTPCH を使えば、データを印刷または穿孔する前にそのデータがこの条件を満たしているかどうかを検査できます。さらに、必要な場合にはシフトアウト / シフトイン文字を IEBTPCH が挿入します。この検査は、RECORD ステートメントの FIELD パラメーターで変換変数を使うことにより指定できます。

DBCS データが含まれるデータ・セットのデフォルトの印刷フォーマットは、1 バイト文字セットのデータのデフォルトとは異なります。PRINT DBCS=YES をコーディングすると、各出力行ごとに最大数の DBCS 文字が印刷されます。IEBTPCH は DBCS ストリングの途中で出力レコードが終わらないようにします。DBCS ストリング全体が 1 つの印刷行に収まらないと、IEBTPCH は各行をシフトアウト / シフトイン文字で囲みます。これらの制御文字は印刷されませんが、プリンターがそのデータを確実に DBCS ストリングとして認識するようにします。

選択レコードの印刷と穿孔

IEBTPCH では、印刷または穿孔するデータ・セットから特定のレコードのみを選択することが可能です。ユーティリティー制御ステートメントを使って以下のものを指定できます。

- 指定した数のレコードを印刷または穿孔した後の、印刷または穿孔操作の終了
- n 個おきのレコードの印刷または穿孔
- 指定した数のレコードの後の、印刷または穿孔操作の開始

区分ディレクトリーの印刷と穿孔

IEBTPCH を使って、区分ディレクトリーの内容を印刷または穿孔することができます。ディレクトリーを 16 進表現で印刷した場合、印刷される各ディレクトリー・ブロックの先頭 4 文字はそのブロックで使われている合計バイト数を示します。ディレクトリーのフォーマットの詳細については、*z/OS DFSMS* データ・セットの使用法 を参照してください。ディレクトリーを穿孔すると、ディレクトリー・ブロックからのデータは、そのブロックを表している穿孔カードの列に連続して穿孔されます。

PDSE ディレクトリーはブロックを含んでいませんが、区分データ・セットのディレクトリーと同じフォーマットで印刷または穿孔されます。

ディスクまたは磁気テープへの印刷と穿孔

IEBTPCH を使用すれば、データ・セットをディスクや磁気テープに『印刷』または『穿孔』することができます。これを実行した場合、各レコードの先頭文字は ASA 紙送り制御文字になります。この文字は、印刷または穿孔操作を制御するのに使われるものです。データ・セットの印刷や穿孔は後ほど、PRINT または PUNCH ステートメントの PREFORM=A パラメーターを使って行うことが可能です。

データ・セットをディスクや磁気テープに穿孔するときに、位置 73~80 にシーケンス番号を含めることができます (任意)。

データ・セットに 2 バイト文字セットのデータが含まれている場合、該当する DBCS スtringが出力データ・セットの論理レコード長を超えると、シフトアウト / シフトイン文字 (< と >) がレコードの先頭と末尾に挿入されます。

入出力

IEBTPCH で使用する入力はい下のとおりです。

- 印刷または穿孔しているデータが入っている、入力データ・セット。この入力データ・セットは順次、区分のいずれでも構いません。
- ユーティリティー制御ステートメントが含まれている、制御データ・セット。制御データ・セットは IEBTPCH を使用するたびに必要となります。

IEBTPCH は以下のような出力を生成します。

- 印刷または穿孔しているデータ・セットである、出力データ・セット。
- 通知メッセージ (たとえば、制御ステートメントの内容) や何らかのエラー・メッセージが入っている、メッセージ・データ・セット。

IEBTPCH をアプリケーション・プログラムから呼び出す場合、IEBTPCH を呼び出す前に SVC 99 を出すことにより、データ・セットを動的に割り振ることができます。

IEBTPCH の戻りコードについては、付録 Aを参照してください。

制御

IEBTPCH はジョブ制御ステートメントやユーティリティー制御ステートメントで制御します。ジョブ制御ステートメントは、IEBTPCH プログラムの処理やロード、およびそのプログラムが使用したり生成したりするデータ・セットの定義を行うときに必要になります。ユーティリティー制御ステートメントは、IEBTPCH の機能を制御するのに使用します。

ジョブ制御ステートメント

表 30 は、IEBTPCH で使用するジョブ制御ステートメントをまとめたものです。

表 30. IEBTPCH で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブ・ステップを開始します。
EXEC	プログラム名を指定します (PGM=IEBTPCH)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。データ・セットはシステム出力装置、磁気テープ・ボリューム、直接アクセス装置などに出力できます。
SYSUT1 DD	入力用の順次データ・セット、区分データ・セット、PDSE を定義します。
SYSUT2 DD	出力 (印刷または穿孔) データ・セットを定義します。
SYSIN DD	制御データ・セットを定義します。制御データ・セットは通常、入力ストリームにありますが、区分データ・セットや PDSE の中のメンバーとして定義することもできます。

SYSPRINT DD ステートメント

SYSPRINT DD ステートメントは IEBTPCH を使用するときに必要となります。レコード・フォーマットは常に FBA であり、論理レコード長は常に 121 です。出力はブロック・サイズを指定することでブロック化できます。ブロック・サイズは SYSPRINT DD ステートメントに 121 の倍数で指定します。デフォルト・ブロック・サイズは 121 です。

SYSUT1 DD ステートメント

SYSUT1 DD ステートメントは IEBTPCH を使用するときに必要となります。レコード・フォーマット (不定形式レコードの場合を除く)、ブロック・サイズ、および論理レコード長 (不定形式レコードや固定長非ブロック化レコードの場合を除く) は、DSCB 内の DD ステートメントに、または磁気テープ・ラベルになければなりません。

入力データ・セットには、固定長、可変長、不定形式、可変長スパンといったレコードを入れることができます。可変長スパン・レコードを使用できるのは、入力データが順次の場合に限られます。ブロック・サイズは、32760 バイトを超えてはなりません。

区分データ・セットや PDSE のディレクトリーは順次データ・セットと見なされます。PRINT または PUNCH ステートメントには TYPORG=PS を指定してください。SYSUT1 DD ステートメントには RECFM=U、BLKSIZE=256、LRECL=256 を指定する必要があります。

SYSUT2 DD ステートメント

SYSUT2 DD ステートメントは、IEBTPCH を使用するときには必ず必要になります。レコード・フォーマットは常に FBA または FBM になります。LRECL パラメーター、または BLKSIZE パラメーター（論理レコード長が指定されていない場合）には、印刷行当たりの、または穿孔カード当たりの出力文字数を指定します（このカウントには制御文字も含める必要があります）。指定する文字数は 2~145 の範囲内でなければなりません。編集した出力行のデフォルト値は、印刷行当たりで 121 文字、穿孔カード当たりで 81 文字です。

SYSUT2 データ・セットは LRECL パラメーターと BLKSIZE パラメーターの両方を指定することによってブロック化できます。この場合、ブロック・サイズは論理レコード長の倍数でなければなりません。ブロック・サイズは 32760 バイトを超えてはなりません。

出力データ・セットとメッセージ・データ・セットは、システム出力装置がプリンターであれば出力可能です。

入力レコードの論理レコード長のために、出力データが出力レコード長を超えてしまうと、標準印刷出力や標準穿孔出力の場合、あるいは PREFORM パラメーターが指定されている場合は、IEBTPCH はそのレコードを複数の行またはカードに分割します。標準以外の出力の場合、あるいは PREFORM パラメーターが指定されていない場合は、入力レコードの一部だけが印刷または穿孔されます（最大文字数をご使用の出力装置の特性によって決まります）。

SYSIN DD ステートメント

SYSIN DD ステートメントは IEBTPCH を使用するたびに必要となります。レコード・フォーマットは常に FB であり、論理レコード長は常に 80 です。80 の倍数であり、最大 32760 までであれば、任意のブロック化因数をブロック・サイズに指定することができます。デフォルト・ブロック・サイズは 80 です。

ユーティリティー制御ステートメント

IEBTPCH はユーティリティー制御ステートメントで制御します。216 ページの表 31 に示す制御ステートメントは、そこに示されている順序どおりに指定しなければなりません。

制御ステートメントは必要に応じて制御データ・セットに組み込みます。MEMBER および RECORD ステートメントは 1 つのジョブ・ステップにいくつでも組み込むことができます。

IEBTPCH の継続ステートメントでは、72 桁目に非ブランク文字を任意で指定できます。ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

表 31. IEBTPCH のユーティリティー制御ステートメント

ステートメント	用途
PRINT	データを印刷することを指定します。
PUNCH	データを穿孔することを指定します。
TITLE	表題を印刷データまたは穿孔データの前に出力することを指定します。
EXITS	出口ルーチンをユーザーが提供することを指定します。
MEMBER	区分データ・セットや PDSE のどのメンバーを印刷または穿孔するかを指定します。
RECORD	データを印刷または穿孔するフォーマットを指定します。
LABELS	ラベルをデータとして扱うかどうかを指定します。

PRINT および PUNCH ステートメント

PRINT および PUNCH ステートメントは、データ・セットを印刷または穿孔するかどうかを指定するのに使用します。制御データ・セットには、その最初のステートメントとしてこのいずれかのステートメントを組み込む必要があります。両方のステートメントを同時に使用することはできず、同じステートメントを 2 回以上使うこともできません。

PRINT および PUNCH ステートメントの構文は次のとおりです。

[label]	{PRINT PUNCH}	[PREFORM={A M}] [,TYPORG={PSIPO}] [,TOTCONV={XE PZ}] [,CNTRL={n1}] [,STRTAFT=n] [,STOPAFT=n] [,SKIP=n] [,MAXNAME=n] [,MAXFLDS=n] [,MAXGPS=n] [,MAXLITS=n] [,DBCS={YES NO}] [,INITPG=n] [,MAXLINE=n] [,CDSEQ=n] [,CDINCR=n]
注:		
<ul style="list-style-type: none"> INITPG と MAXLINE は、PRINT の場合にのみ指定可能です。 CDSEQ と CDINCR は、PUNCH の場合にのみ指定可能です。 		

各項目の意味は以下のとおりです。

PREFORM={A|M}

印刷または穿孔する各レコードの先頭文字を制御文字にすることを指定します。

制御文字は、行送り、ページ当たりの行数、ページ排出、スタッカーの選択を制御するのに使用します。つまり、出力データはあらかじめフォーマット設定され、そのフォーマットに従って印刷または穿孔が行われることとなります。エラーが発生すると印刷 / 穿孔操作は停止します。 **PREFORM** がコーディングされている場合は、**PRINT** または **PUNCH** のすべてのオペランド (**TYPORG** を除く)、および他のすべての制御ステートメント (**LABELS** を除く) は無視されます。しかしながら、無視されたステートメントやオペランドに対しても構文が正しいかどうかの検査は行われます。 **PREFORM** は、32K バイトを超える **VS** または **VBS** レコードを持つデータ・セットの印刷や穿孔には使わないでください。上記の値は以下のようにコーディングします。

- A** 印刷または穿孔する各レコードの先頭文字を **ASA** 制御文字にすることを指定します。入力レコード長が出力レコード長を上回る場合には、ユーティリティーはこの **ASA** 文字を使って最初の行を印刷し、レコードのそれ以降のすべての行では単一のスペース文字を使います (**PRINT** の場合)。あるいは、レコードのそれぞれの出力カードにこの **ASA** 文字を毎回出力します (**PUNCH** の場合)。

印刷または穿孔するデータ・セットが **IEBTPCH** を使ってフォーマット設定したものであれば、**PREFORM=A** をコーディングしなければなりません。

- M** 印刷または穿孔する各レコードの先頭文字をマシン・コード制御文字にすることを指定します。入力レコード長が出力レコード長を上回る場合には、ユーティリティーは *print-skip-one-line* 文字を使ってレコードのすべての行を印刷し、最後の行になるまで印刷を続けます。これには、入力データとして提供される実際の文字が含まれます (**PRINT** の場合)。あるいは、レコードのそれぞれの出力カードにこのマシン・コード文字を毎回出力します (**PUNCH** の場合)。

TYPORG={PSIPO}

入力データ・セットの編成を指定します。上記の値は以下のようにコーディングします。

- PS** 入力データ・セットが順序どおりに編成されることを指定します。これがデフォルトです。

PO

入力データ・セットが区分化されることを指定します。

TOTCONV={XE|PZ}

印刷または穿孔するデータの表示方法を指定します。 **TOTCONV** は、同じデータに属する任意のユーザー指定 (**RECORD** ステートメント) で指定変更することができます。上記の値は以下のようにコーディングします。

XE

1 バイトを 2 文字で表す 16 進表現でデータを穿孔することを指定します (例、C3 40 F4 F6)。 **XE** を指定していないと、データは 1 バイトを 1 文字で表す英数字表現で穿孔されます。この場合、前述の例は C 46 となります。

入力レコード (長さが L) の変換部分は、出力データの 2L 字分を占めます。

PZ

データ (パック 10 進モード) をアンパック 10 進モードに変換することを指定します。IEBPTPCH はパック 10 進モードでの検査を行いません。

入力レコード (長さが L) の変換部分は、穿孔時に出力データの 2L - 1 字分を占めます。

デフォルト: TOTCONV を省略すると、データは変換されません。

CNTRL={n1}

PRINT の場合、CNTRL には行送りを示す出力装置用の ASA 制御文字を指定します。1 は通常の行送り (デフォルト)、2 は 2 倍の行送り、3 は 3 倍の行送りをそれぞれ意味します。

PUNCH の場合、CNTRL にはスタッカーを選択するのに使う出力装置用の ASA 制御文字を指定します。1 は最初のスタッカー (デフォルト)、2 は 2 番目のスタッカー、3 は 3 番目のスタッカーを (もしあれば) それぞれ意味します。

STRTAFT=n

順次データ・セットの場合、印刷または穿孔を開始する前にスキップする論理レコード (レコードが可変長スパン (VS) または可変長ブロック・スパン (VBS) であり、32K バイトより長いときには物理レコード) の数を指定します。区分データ・セットまたは PDSE の場合は、印刷または穿孔を開始する前に各メンバーでスキップする論理レコードの数を指定します。n の値は 32767 を超えてはなりません。STRTAFT が指定されており、RECORD がすでにある場合は、メンバーの最初の RECORD ステートメントに、印刷または穿孔する最初の論理レコードのフォーマットを記述します。

STOPAFT=n

順次データ・セットの場合、印刷または穿孔する論理レコード (レコードが VS または VBS であり、32K バイトより長いときには物理レコード) の数を指定します。区分データ・セットまたは PDSE の場合は、処理する各メンバーに印刷または穿孔する論理レコード (レコードが VS または VBS であり、32K バイトより長いときには物理レコード) の数を指定します。n の値は 32767 を超えてはなりません。STOPAFT が指定されており、RECORD ステートメントの IDENT パラメーターも指定されている場合は、STOPAFT カウントに達するか、最初のレコード・グループの終わりに達するとその時点で操作が停止します。

SKIP=n

レコード (レコードが VS または VBS であり、32K バイトより長いときは物理ブロック) を n 個おきに印刷または穿孔することを指定します。

デフォルト: 論理レコードを連続して印刷または穿孔します。

MAXNAME=n

後続の MEMBER ステートメントに示されるメンバー名および別名の合計数より大きい数を指定します。この値は 32767 を超えてはなりません。

MEMBER ステートメントがある場合に MAXNAME を省略すると、印刷または穿孔の要求は停止します。

MAXFLDS=*n*

後続の RECORD ステートメントに示される FIELD パラメーターの合計数より大きい数を指定します。この値は 32767 を超えてはなりません。

FIELD ステートメントがある場合に MAXFLDS を省略すると、印刷または穿孔の要求は停止します。

MAXGPS=*n*

後続の RECORD ステートメントに示される IDENT パラメーターの合計数より大きい数を指定します。この値は 32767 を超えてはなりません。

IDENT ステートメントがある場合に MAXGPS を省略すると、印刷または穿孔の要求は停止します。

MAXLITS=*n*

後続の RECORD ステートメントの IDENT リテラルに含まれている文字の合計数より大きい数を指定します。この値は 32767 を超えてはなりません。

リテラルがある場合に MAXLITS を省略すると、印刷または穿孔の要求は終了します。

DBCS={YES|NO}

印刷または穿孔するデータ・セットに 2 バイト文字セットのデータが含まれるかどうかを指定します。デフォルトは NO です。

INITPG=*n*

最初のページ番号を指定します。各ページには、この番号から始まる番号が順序に従って付けられます。INITPG の値は 9999 を超えてはなりません。デフォルトは 1 です。

INITPG をコーディングできるのは PRINT ステートメントの場合のみです。

MAXLINE=*n*

印刷ページに対する最大行数を指定します。この数にはスペース、表題、副題も含まれます。デフォルトは 1 ページ当たり 60 行です。

MAXLINE をコーディングできるのは PRINT ステートメントの場合のみです。

CDSEQ=*n*

穿孔カードのデッキの最初のシーケンス番号を指定します。この値は 73~80 桁目に入れなければなりません。シーケンス番号の指定は、区分データ・セットのメンバーごとに初期化されます。デフォルトではカードには番号が付けられません。

CDSEQ をコーディングできるのは PUNCH ステートメントの場合のみです。

CDINCR=*n*

シーケンス番号の生成に使う増分値を指定します。デフォルトの増分値は、CDSEQ がコーディングされていなければ 10 になります。CDSEQ がコーディングされている場合、レコードに番号は付けられません。

CDINCR をコーディングできるのは PUNCH ステートメントの場合のみです。

TITLE ステートメント

TITLE ステートメントは、データ・セットで印刷または穿孔したい任意の表題や副題を指定するのに使います。TITLE ステートメントは IEBTPCH を使用するたびに組み込むことができます。最初の TITLE ステートメントには表題を、2 番目には

副題を定義します。TITLE ステートメントは、組み込まれている場合には、制御データ・セットの中の PRINT または PUNCH ステートメントの後にあります。

データ・セットを印刷する場合、指定してある表題はすべてのページに印刷されません。

TITLE ステートメントの構文は次のとおりです。

[label]	TITLE	ITEM=('title'[output-location])[,ITEM=...]
---------	-------	------------------------------------------------

各項目の意味は以下のとおりです。

ITEM=('title'[output-location])

表題または副題の情報を指定します。コーディングできる値は以下のとおりです。

'title'

表題または副題のリテラル (最大長 40 バイト) をアポストロフィで囲んで指定します。リテラルにアポストロフィが含まれている場合は、それぞれのアポストロフィを 2 個の連続したアポストロフィとしてコーディングしなければなりません。'title' にコーディングするリテラルは、PRINT または PUNCH ステートメントの TOTCONV パラメーターには影響されません。表題には、DBCS (2 バイト文字セット) スtringを指定できます。この場合は、その DBCS Stringをシフトアウト / シフトイン文字 (< と >) で囲んでください。

さらに、表題は 16 進値で指定することもできます。この場合は、表題を *title* として符号化します。これは、必要な文字がすべて収められたキーボードを持っていない場合に特に役立ちます。シフトアウト / シフトイン文字はそれぞれ X'0E' と X'0F' です。

output-location

該当する項目のリテラルを入れる出力レコード内の開始位置を指定します。*output-location* を併用するときは、指定した表題の長さに *output-location* を加えた長さが、出力データの論理レコード長 - 1 を超えないようにします。

デフォルト: 先頭の位置 (バイト) が想定されます。

TITLE ステートメントには ITEM を 2 回以上指定できます。この方法により、表題の長さを 40 文字以上にしたり、表題の形式を必要に合わせて設定できます。

EXITS ステートメント

EXITS ステートメントは、IEBTPCH がラベルやレコードを処理するときを使用する出口ルーチンを指示するのに使います。入力データ・セットが区分化されている場合は、ラベル処理ルーチンの出口は無視されます。ユーザー・ルーチンとの間のリンケージについては、377 ページの『付録 C. ユーティリティ・プログラムのユーザー出口を指定する』で説明しています。

EXITS ステートメントが使われている場合は、TITLE ステートメントの直後にあるか、PRINT または PUNCH ステートメントの後になければなりません。

EXITS ステートメントの構文は次のとおりです。

[label]	EXITS	[INHDR=routinename] [,INTLR=routinename] [,INREC=routinename] [,OUTREC=routinename]
---------	-------	----------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

INHDR=routinename

ユーザー入力ヘッダー・ラベルを処理するルーチンの名前を指定します。

INTLR=routinename

ユーザー入力トレーラー・ラベルを処理するルーチンの名前を指定します。

INREC=routinename

各論理レコード (レコードが VS または VBS であり、32K バイトより長いときは物理ブロック) を処理する前にそれら进行操作するルーチンの名前を指定します。

OUTREC=routinename

各論理レコード (レコードが VS または VBS であり、32K バイトより長いときは物理ブロック) を印刷または穿孔する前にそれら进行操作するルーチンの名前を指定します。

MEMBER ステートメント

MEMBER ステートメントは、印刷または穿孔したい区分データ・セットや PDSE のメンバーを示すのに使います。MEMBER ステートメントの後のすべての RECORD ステートメントは、その MEMBER ステートメントで示されているメンバーだけに適用されます。RECORD ステートメントと MEMBER ステートメントを使う場合は、最初の RECORD ステートメントの前に少なくとも 1 つの MEMBER ステートメントが必要です。RECORD ステートメントを使わないと、そのメンバーはデフォルト・フォーマットで処理されます。

MEMBER ステートメントがない状態で区分データ・セットまたは PDSE を処理すると、そのデータ・セットのすべてのメンバーが印刷または穿孔されます。

MEMBER ステートメントは 1 つのジョブ・ステップにいくつでも組み込むことができます。

MEMBER ステートメントが入カストリーム中にある場合は、MAXNAME を PRINT または PUNCH ステートメントに指定する必要があります。

MEMBER ステートメントの構文は次のとおりです。

[label]	MEMBER	NAME={membername aliasname}
---------	--------	-----------------------------

各項目の意味は以下のとおりです。

NAME={*membername*|*aliasname*}

印刷または穿孔する区分データ・セットや PDSE のメンバーを指定します。コーディングできる値は以下のとおりです。

membername

メンバーをそのメンバー名で指定します。

aliasname

メンバーをその別名で指定します。

MEMBER ステートメントが入カストリーム中にある場合は、MAXNAME を PRINT または PUNCH ステートメントに指定する必要があります。

RECORD ステートメント

RECORD ステートメントは、指定内容に従って印刷または穿孔するレコードのグループ (レコード・グループ という) を定義するのに使います。レコード・グループは、同一の編集を行う任意の数のレコードで構成されます。

RECORD ステートメントがないと、データ・セット全体、または指定のメンバー全体がデフォルト・フォーマットで印刷または穿孔されます。デフォルト・フォーマットには、以下のような特性があります。

- 各印刷行には、16 進値による情報のグループ (それぞれが 8 文字) が含まれています。
- 各入力レコードの先頭では、印刷出力の改行が行われます。
- 入力レコードのサイズとキャリッジ幅は、入力レコードごとに必要とされる印刷出力の行数によって決まります。

RECORD ステートメントが使われている場合は、そのステートメントで定義しているレコード・グループの後にくるすべてのデータ (区分メンバー内の、あるいは順次データ・セット内の) が、他の RECORD ステートメントで定義されている必要があります。RECORD ステートメントは 1 つのジョブ・ステップにいくつでも組み込むことができます。

RECORD ステートメントが参照している区分データ・セットまたは PDSE にメンバーが指定されていない場合、そのステートメントに含める必要があるのは FIELD パラメーターだけです。これらはデータ・セットのすべてのメンバー内のレコードに適用されます。

FIELD パラメーターが RECORD ステートメントに組み込まれている場合は、MAXFLDS を PRINT または PUNCH ステートメントに指定しなければなりません。

IDENT パラメーターが RECORD ステートメントに組み込まれている場合は、MAXGPS と MAXLITS を PRINT または PUNCH ステートメントに指定しなければなりません。

RECORD ステートメントの構文は次のとおりです。

[label]	RECORD	[IDENT=(length,'name', input-location)] [,FIELD=(length,[input-location],[conversion], [output-location])][,FIELD=...]
---------	--------	---------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

IDENT=(length,'name',input-location)

FIELD パラメーターが適用されるレコード・グループの最後のレコードを示します。コーディングできる値は以下のとおりです。

length

入力レコード内の識別名が入っているフィールドの長さ (バイト単位) を指定します。この長さは 8 バイトを超えてはなりません。

'name'

レコード・グループの最後のレコードを示す正確なリテラルを、アポストロフィで囲んで指定します。リテラルにアポストロフィが含まれている場合は、それぞれを 2 個の連続したアポストロフィとして記述しなければなりません。

'name' は、*name* の部分を符号化することにより、16 進値で指定することが可能です。さらに、'name' には DBCS スtring も指定することも可能で、DBCS 文字をそのまま指定することも、DBCS 文字を 16 進表現にして指定することもできます。DBCS スtring に対して 16 進値を使用する場合、シフトアウト / シフトイン文字の 16 進値はそれぞれ X'0E' と X'0F' になります。

input-location

入力レコード内の識別名が入っているフィールドの開始位置を指定します。

長さを入力位置を合計した数字は、入力データの論理レコード長 + 1 以下でなければなりません。

デフォルト: IDENT が省略されており、STOPAFT が PRINT または PUNCH ステートメントに組み込まれていないと、データ・セット内の最後のレコードの後でレコードの処理が停止します。IDENT は省略されているが、STOPAFT が PRINT または PUNCH ステートメントに組み込まれている場合は、STOPAFT カウントに達するか、データ・セット内の最後のレコードを処理した後でレコードの処理が停止します。

IDENT パラメーターが RECORD ステートメントに組み込まれている場合は、MAXGPS と MAXLITS を PRINT または PUNCH ステートメントに指定しなければなりません。

FIELD=(length,[input-location] [,conversion],[output-location])

フィールド処理と編集の情報を指定します。

FIELD パラメーターの変数は定位置である点に注意してください。つまり、いずれかのオプションをコーディングしないときは、その変数の前には必ずの関連付けられたコンマをコーディングしなければなりません。

コーディングできる値は以下のとおりです。

length

処理する入力フィールドの長さ (バイト単位) を指定します。指定する長さは、入力データの最初の論理レコード長以下でなければなりません。

input-location

処理する入力フィールドの開始位置を指定します。長さと入力位置を合計した数字は、入力データの論理レコード長 + 1 以下でなければなりません。

デフォルト: 先頭の位置 (バイト) が想定されます。

conversion

該当するフィールドを印刷または穿孔する前にそのフィールドに対して実行する変換のタイプを指定します。コーディングできる値は以下のとおりです。

CV

DBCS (2 バイト文字セット) 文字を SBCS (1 バイト文字セット) 文字に結合し、その DBCS 文字を検査して印刷可能であることを確認することを指定します。DBCS スtringを囲むためのシフトアウト / シフトイン文字は挿入されません。

DBCS=YES を PRINT または PUNCH ステートメントに指定する必要があります。

PZ

パック 10 進データをアンパック 10 進データに変換することを指定します。入力レコード (長さが L) の変換部分は、穿孔時には出力データの 2L - 1 字分、印刷時には出力データの 2L 字分を占めます。

VC

DBCS (2 バイト文字セット) 文字を検査して印刷可能であるかを確認し、DBCS スtringを囲むためのシフトアウト / シフトイン文字 (< と >) を挿入することを指定します。

DBCS=YES を PRINT または PUNCH ステートメントに指定する必要があります。

XE

英数字データを 16 進データに変換することを指定します。入力レコード (長さが L) の変換部分は、出力データの 2L 字分を占めます。

デフォルト: フィールドを変更せずに出力域に移動します。

output-location

出力レコード内の該当するフィールドの開始位置を指定します。出力レコード内の未指定フィールドは、印刷または穿孔する出力データでは空白で出力されます。SYSUT2 プリンター (穿孔装置) のサイズを上回るデータは印刷 (穿孔) されません。1 つまたは複数の FIELD を指定する場合、変換に必要とされるすべての長さとの追加文字の合計が、出力データの LRECL - 1 以下でなければなりません。

デフォルト: 先頭の位置 (バイト) が想定されます。

FIELD パラメーターが RECORD ステートメントに組み込まれている場合は、MAXFLDS を PRINT または PUNCH ステートメントに指定しなければなりません。

LABELS ステートメント

LABELS ステートメントは、データ・セットのラベルをデータとして扱うかどうかを指定するのに使います。このオプションの詳細については、382 ページの『ユーザー・ラベルの処理』を参照してください。

入力データ・セットを標準外ラベル (NSL) で処理している場合、標準ユーザー・ラベル (SUL) の出口を非アクティブにするには、LABELS DATA=NO を指定しなければなりません。

有効な LABELS ステートメントが 2 つ以上組み込まれている場合は、最後の LABELS ステートメント以外はすべて無視されます。

LABELS ステートメントの構文は次のとおりです。

[label]	LABELS	[CONV={PZ XE}] [,DATA={YES NO ALL ONLY}]
---------	--------	---------------------------------------------

各項目の意味は以下のとおりです。

CONV={PZ|XE}

該当するフィールドを印刷または穿孔する前にそのフィールドに対して実行する変換のタイプを指定します。コーディングできる値は以下のとおりです。

PZ

データ (パック 10 進) をアンパック 10 進データに変換することを指定します。入力レコード (長さが L) の変換部分は、穿孔時には出力データの 2L - 1 字分、印刷時には出力データの 2L 字分を占めます。

XE

英数字データを 16 進データに変換することを指定します。入力レコード (長さが L) の変換部分は、出力データの 2L 字分を占めます。

デフォルト: フィールドを変更せずに出力域に移動します。

DATA={YES|NO|ALL|ONLY}

ユーザー・ラベルをデータとして扱うかどうかを指定します。コーディングできる値は以下のとおりです。

YES

任意のユーザー・ラベルをデータとして扱うことを指定します。ただし、EXITS ステートメントに指定したラベル処理ルーチンがそれらのラベルを受け入れない場合はデータとして扱われません。ラベルをデータとして処理する場合、その処理は標準の戻りコードに従って停止します。デフォルトは YES です。

NO

ユーザー・ラベルをデータとして扱わないことを指定します。入力 / 出力

IEBPTPCH

データ・セットを標準外ラベル (NSL) で処理する場合は、標準ユーザー・ラベル (SUL) の出口を非アクティブにするために NO を指定しなければなりません。

ALL

すべてのユーザー・ラベルをデータとして扱うことを指定します。戻りコードが 16 の場合、ユーティリティはそのユーザー・ラベルのグループの残りのラベルをすべて処理した後、そのジョブ・ステップを停止します。

ONLY

ユーザー・ヘッダー・ラベルだけをデータとして扱うことを指定します。ユーザー・ヘッダー・ラベルは戻りコードに関係なく、データとして処理されます。ジョブは OPEN ルーチンから戻った時点で終了します。

IEBPTPCH の例

以下の各例では、IEBPTPCH の使用方法の一部を示します。表 32 は IEBPTPCH の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 32. IEBPTPCH の例の一覧表

操作	データ・セット編成	装置	注釈	例
印刷	区分	ディスクおよびシステム・プリンター	デフォルト・フォーマット。16 進値への変換。各メンバーから 10 個のレコードが印刷されます。	8
印刷	区分	ディスクおよびシステム・プリンター	デフォルト・フォーマット。16 進値への変換。2 つのメンバーが印刷されます。	9
印刷	区分	ディスクおよびシステム・プリンター	デフォルト・フォーマット。DBCS データを検査して印刷します。	10
印刷	順次	システム・プリンター	16 進値への変換。	3
印刷	順次	テープおよびシステム・プリンター	デフォルト・フォーマット。	4
印刷	順次	ディスクおよびシステム・プリンター	ユーザー指定のフォーマット。ユーザー・ルーチンが用意されています。3 番目のレコード・グループを印刷した後、あるいは STOPAFT に達した時点で処理が終了します。	5
印刷	順次	システム・プリンター	ユーザー出口ルーチンを使って印刷します。	6
印刷	順次、区分	ディスクおよびシステム・プリンター	SYSOUT フォーマット。16 進値への変換。	7
PUNCH	順次	ディスクおよびカード穿孔装置	ユーザー指定のフォーマット。シーケンス番号が割り当てられて穿孔されます。	1

表 32. IEBTPCH の例の一覧表 (続き)

操作	データ・セット編成	装置	注釈	例
PUNCH	順次	カード読取装置 およびカード穿孔装置	ユーザー指定のフォーマットのセットのコピーが作成されます。	2

各例で、実際の装置番号の代わりに **disk** または **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: 区分データ・セットを印刷する

この例では、区分データ・セットのメンバーをユーザー指定に従って印刷します。

```
PRINT TYPORG=PO,MAXNAME=1,MAXFLDS=1
MEMBER NAME=UTILUPD8
RECORD FIELD=(80)
```

メンバー・カード項目が使われていないと、データ・セル全体が印刷されます。

例 2: 順次データ・セットを穿孔する

この例では、順次データ・セットをユーザー指定に従って穿孔します。

```
//PHSEQNO JOB ...
//STEP1 EXEC PGM=IEBTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=MASTER.SEQSET,LABEL=(,SUL),DISP=SHR
//SYSUT2 DD SYSOUT=B
//SYSIN DD *
PUNCH MAXFLDS=1,CDSEQ=0,CDINCR=20
RECORD FIELD=(72)
LABELS DATA=YES
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、ディスクまたはテープ・ボリューム上にある入力データ・セット **MASTER SEQSET** を定義しています。このデータ・セットには 80 バイトの固定長ブロック・レコードが含まれています。
- **SYSUT2 DD** では、システム出力クラス (穿孔が想定されています) を定義しています。FIELD パラメーターで定義している入力データの各レコードのその部分は、1 つの穿孔カードで表されます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **PUNCH** では、穿孔操作を開始し、1 つの FIELD パラメーターを後続の RECORD ステートメントに組み込むことを示し (MAXFLDS=1)、最初の穿孔カードのシーケンス番号 (00000000) とそれ以降のシーケンス番号の増分値 (20) を割り当てています。シーケンス番号の位置は出力レコードの 73~80 桁目です。
- **RECORD** では、入力レコードの位置 1~72 を穿孔することを示しています。入力レコードのバイト 73~80 は、出力カード・デッキ内の新しいシーケンス番号に置き換えられます。
- **LABELS** では、ユーザー・ヘッダー・ラベルとユーザー・トレーラー・ラベルを穿孔することを指定しています。

ラベルは編集することができず、それらは常に出力バッファの先頭 80 バイトに移動させられます。ユーザー・ヘッダーおよびユーザー・トレーラーのレコードが含まれているカードにはシーケンス番号が出力されません。

例 3: カード・デックを複写する

この例では、有効な穿孔カードのコードまたは BCD が入っているカード・デックが複写されます。

```
//PUNCH    JOB    ...
//STEP1    EXEC  PGM=IEBPTPCH
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  DSNAME=PDSLIB(PNCHSTMT),DISP=(OLD,KEEP)
//SYSUT2   DD  SYSOUT=B
//SYSUT1   DD  DATA
```

```
(// カードを含む入力カード・データ・セット)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSIN DD では、メンバー PNCHSTMT が入っている制御データ・セット PDSLIB を定義しています。(このデータ・セットはカタログ化されています。) レコード・フォーマットは FB、論理レコード長は 80 でなければなりません。
- SYSUT2 DD では、システム出力クラス (穿孔が想定されています) を定義しています。
- SYSUT1 DD では、入力ストリーム内の後続の入力カード・データ・セットを定義しています。

例 4: 順次データ・セットをデフォルト・フォーマットで印刷する

この例では、順次データ・セットをデフォルト・フォーマットに従って印刷します。印刷される出力データは 16 進値に変換されます。

```
//PRINT    JOB    ...
//STEP1    EXEC  PGM=IEBPTPCH
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=INSET,UNIT=tape,
//           LABEL=(,NL),VOLUME=SER=001234,
//           DISP=(OLD,KEEP),DCB=(RECFM=U,BLKSIZE=2000)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  *
           PRINT  TOTCONV=XE
           TITLE  ITEM=('PRINT SEQ DATA SET WITH CONV TO HEX',10)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、磁気テープ・ボリューム上にある入力データ・セットを定義しています。このデータ・セットには不定形式レコードが含まれていますが、どのレコードも 2,000 バイト以下です。
- SYSUT2 DD では、出力データ・セットを定義しています。このデータ・セットはシステム出力装置 (プリンターが想定されています) に出力されます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT では、印刷操作を開始し、英数字表現から 16 進表現への変換を指定しています。

- TITLE では、表題を印刷出力の列 10 から始まる位置に出力することを指定しています。この表題は 16 進値には変換されません。

例 5: 順次データ・セットをユーザー指定に従って印刷する

この例では、順次データ・セットをユーザー指定に従って印刷します。

```
//PTNONSTD JOB ...
//STEP1 EXEC PGM=IEBTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SEQSET,UNIT=tape,LABEL=(2,SUL),
//          DISP=(OLD,KEEP),VOLUME=SER=001234
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
          PRINT MAXFLDS=1
          EXITS INHDR=HDRIN,INTLR=TRLIN
          RECORD FIELD=(80)
          LABELS DATA=YES
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、テープ・ボリューム上の 2 番目のデータ・セットである入力データ・セット SEQSET を定義しています。
- SYSUT2 DD では、システム出力装置 (プリンターが想定されています) 上の出力データ・セットを定義しています。各印刷行には、連続した 80 文字 (1 レコード) の情報が入っています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT では、印刷操作を開始し、後続の RECORD ステートメントに 1 つの FIELD パラメーターが組み込まれることを示しています (MAXFLDS=1)。
- EXITS では、ユーザー・ヘッダー・ラベルやトレーラー・ラベルが SYSUT1 データ・セットで検出されると、それぞれの処理ルーチンが出口となることを示しています。
- RECORD では、各入力レコードの全体 (80 バイト) が処理されることを示しています。各入力レコードは、プリンターの 1~80 桁目に出力されます。
- LABELS では、ユーザー・ヘッダー・ラベルとユーザー・トレーラー・ラベルをユーザー出口が出す戻りコードに従って印刷することを指定しています。

例 6: 3 つのレコード・グループを印刷する

この例では、3 つのレコード・グループが印刷されます。出力レコードを印刷する前にそれら进行操作するためのユーザー・ルーチンを提供しています。

```
//PRINT JOB ...
//STEP1 EXEC PGM=IEBTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SEQDS,UNIT=disk,DISP=(OLD,KEEP),
//          LABEL=(,SUL),VOLUME=SER=111112
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
          PRINT MAXFLDS=9,MAXGPS=9,MAXLITS=23,STOPAFT=32767
          TITLE ITEM=('TIMECONV-DEPT D06'),
                ITEM=('JAN10-17',80)
          EXITS OUTREC=NEWTIME,INHDR=HDRS,INTLR=TLRS
          RECORD IDENT=(6,'498414',1),
                FIELD=(8,1,,10),FIELD=(30,9,XE,20)
          RECORD IDENT=(2,'**',39),
                FIELD=(8,1,,10),FIELD=(30,9,XE,20)
/*
```



```

RECORD  IDENT=(6,'498414',1),
        FIELD=(8,1,,10),FIELD=(30,9,XE,20)
LABELS  CONV=XE,DATA=ALL
/*

```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、SEQDS という名前の入力データ・セットを定義しています。このデータ・セットはディスク・ボリューム上にあります。
- **SYSUT2 DD** では、システム出力装置 (プリンターが想定されています) 上の出力データ・セットを定義しています。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **PRINT** ステートメントでは以下の内容を実行しています。
 1. 印刷操作を初期化する。
 2. 後続の **RECORD** ステートメントに 9 個以下の **FIELD** パラメーターが組み込まれることを示す (**MAXFLDS=9**)。
 3. 後続の **RECORD** ステートメントに 9 個以下の **IDENT** パラメーターが組み込まれることを示す (**MAXGPS=9**)。
 4. 後続の **IDENT** パラメーターに 23 字以下のリテラルが組み込まれることを示す (**MAXLITS=23**)。
 5. 32767 個のレコードを処理した後、あるいは 3 番目のレコード・グループを処理した後で処理を終了することを示す。 **MAXLINE** が省略されているため、各ページには 60 行が印刷されます。
- **TITLE** では 2 つの表題が指定されており、それらは 1 つの行に印刷されます。これらの表題は 16 進値には変換されません。
- **EXITS** では、ユーザー・ルーチンの名前 (**NEWTIME**) を指定しています。この名前は、出力レコードを印刷する前にそれらを実行するときに使われます。
- 最初の **RECORD** ステートメントでは、最初に処理するレコード・グループを定義し、入力レコードからの情報を出力レコードのどの部分に出力するかを示しています。入力レコードの位置 1~8 は印刷出力の位置 10~17 に出力されます。また、位置 9~38 は 16 進表現で位置 20~79 に出力されます。
- 2 番目の **RECORD** ステートメントでは、2 番目に処理するグループを定義しています。 **IDENT** オペランド内のパラメーターでは、入力レコードの最後のレコードがこの **RECORD** ステートメント内の **FIELD** オペランドに従って編集されることを指定します。この **FIELD** オペランドでは、入力レコードの位置 1~8 が印刷出力の位置 10~17 に出力され、位置 9~38 が 16 進表現で位置 20~79 に出力されることを指定しています。
- 3 番目で最後の **RECORD** ステートメントは、最初の **RECORD** ステートメントと同じものです。 **IDENT** オペランドのパラメーターに一致する入力レコードがあると、それが **PRINT** ステートメントの **STOPAFT** パラメーターにまだ一致していない場合には、処理が終了します。
- **LABELS** では、ユーザーの出口ルーチンが出した戻りコードに関係なく (16 は例外)、すべてのユーザー・ヘッダー・ラベルとユーザー・トレーラー・ラベルを印刷することを指定しています。さらに、ラベルを英数字表現から 16 進表現に変換することを示しています (**CONV=XE**)。

例 7: 以前にフォーマット設定したデータ・セットを印刷する

この例では、以前に標準ラベル・テープの 2 番目のデータ・セットとして出力した SYSOUT (順次) データ・セットが入力データになっています。これは SYSOUT フォーマットで印刷されています。

```
//PTSYSOUT JOB ...
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=tape,LABEL=(2,SL),DSNAME=LISTING,
//          DISP=(OLD,KEEP),VOL=SER=001234
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
          PRINT PREFORM=A
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、以前に標準ラベル・テープの 2 番目のデータ・セットとして出力した入力データを定義しています。このデータ・セットには LISTING という名前がすでに割り当てられています。
- SYSUT2 DD では、システム出力装置 (プリンターが想定されています) 上の出力データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT ステートメントでは、印刷操作を開始し、印刷する各レコードの先頭文字を ASA 制御文字にすることを示しています (PREFORM=A)。

例 8: 区分データ・セットのディレクトリーを印刷する

この例では、区分データ・セットのディレクトリーが、デフォルト・フォーマットで印刷されます。印刷される出力データは 16 進値に変換されます。

```
//PRINTDIR JOB ...
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=MAIN.PDS,
//          DISP=(OLD,KEEP),DCB=(RECFM=U,BLKSIZE=256)
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
          PRINT TYPORG=PS,TOTCONV=XE
          TITLE ITEM=('PRINT PARTITIONED DIRECTORY OF PDS',10)
          TITLE ITEM=('FIRST TWO BYTES SHOW NUM OF USED BYTES',10)
          LABELS DATA=NO
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ディスク・ボリューム上にある入力データ・セット (区分ディレクトリー) を定義しています。DCB キーワードで記述されているのは、メンバーの内容ではなくて、ディレクトリーです。
- SYSUT2 DD では、システム出力装置 (プリンターが想定されています) 上の出力データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT では、印刷操作を開始し、区分ディレクトリーの編成が順次であることを示し、英数字表現から 16 進表現への変換を指定しています。
- 最初の TITLE ステートメントでは表題を、また 2 番目の TITLE ステートメントでは副題を指定しています。どちらの表題も 16 進値には変換されません。

- LABELS では、ユーザー・ラベルを印刷しないことを指定しています。

注: ディレクトリー・ブロック内のすべてのバイトに、この区分データ・セットに属しているデータが入っている必要はありません。未使用のバイトはオペレーティング・システムが一時作業域として使うことがあります。16 進表現への変換を行うと、256 バイトからなるブロックのうち区分データ・セットに属しているバイト数が、印刷出力の最初の 4 文字に示されます。未使用バイトは使用バイトの中に散在しているわけではなく、すべてディレクトリー・ブロックの後ろの部分にまとまっています。

例 9: 区分データ・セットの選択レコードを印刷する

この例では、区分データ・セット (各メンバーから 10 個のレコード) をデフォルト・フォーマットで印刷します。印刷される出力データは 16 進値に変換されます。

```
//PRINTPDS JOB ...
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=WAYNE.TEST.DATA,DISP=SHR
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
PRINT TOTCONV=XE,TYPORG=PO,STOPAFT=10
TITLE ITEM=('PRINT PDS - 10 RECS EACH MEM',20)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ディスク・ボリューム上にある、WAYNE.TEST.DATA という名前の入力データ・セットを定義しています。
- SYSUT2 DD では、システム出力装置 (プリンターが想定されています) 上の出力データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT では、印刷操作を開始し、英数字表現から 16 進表現への変換を指定し、入力データ・セットが区分化されていることを示し、さらに各メンバーから 10 個のレコードを印刷することを指定しています。
- TITLE では、表題を印刷出力の列 20 から始まる位置に出力することを指定しています。この表題は 16 進値には変換されません。

例 10: 区分データを 16 進値に変換してから印刷する

この例では、2 つの区分メンバーをデフォルト・フォーマットで印刷します。印刷される出力データは 16 進値に変換されます。

```
//PRNTMEMS JOB ...
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=BROWN.MAIN.LIB,DISP=SHR
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
PRINT TYPORG=PO,TOTCONV=XE,MAXNAME=2
TITLE ITEM=('PRINT TWO MEMBS WITH CONV TO HEX',10)
MEMBER NAME=MEMBER1
MEMBER NAME=MEMBER2
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ディスク・ボリューム上にある、BROWN.MAIN.LIB という名前の入力データ・セットを定義しています。
- SYSUT2 DD では、システム出力装置 (プリンターが想定されています) 上の出力データ・セットを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT では、印刷操作を開始し、入力データ・セットが区分であることを示し、英数字表現から 16 進表現への変換を指定し、そして 2 つの MEMBER ステートメントがこの制御データ・セットにあることを示します (MAXNAME=2)。
- TITLE では、表題を印刷出力の列 10 から始まる位置に出力することを指定しています。この表題は 16 進値には変換されません。
- MEMBER では、印刷するメンバーのメンバー名を指定しています (MEMBER1 および MEMBER2)。

例 11: DBCS データが入っているメンバーを印刷する

この例では、DBCS データを検査してそのすべての DBCS 文字が印刷可能であることを確認した後、その DBCS データが入っている区分データ・セットのメンバーを印刷します。

```
//DBCS      JOB    ...
//STEP1    EXEC  PGM=IEBTPCH
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSNAME=PDS,UNIT=disk,DISP=(OLD,KEEP)
//SYSUT2   DD   SYSOUT=A
//SYSIN    DD   *
            PRINT TYPORG=PO,DBCS=YES,MAXFLDS=1,MAXNAME=1
            MEMBER NAME=MEM1
            RECORD FIELD=(,CV)

/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ディスク・ボリューム上にある入力データ・セット PDS を定義しています。
- SYSUT2 DD では、システム・プリンターを出力データ・セットとして定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- PRINT では、印刷操作を開始し、入力データ・セットが区分であることを示し、DBCS (2 バイト文字セット) データを印刷することを示しています。さらに、この制御データ・セットに 1 つの MEMBER ステートメントがあり、また後続の RECORD ステートメントに 1 つの FIELD パラメーターがあることも示しています。
- MEMBER では、印刷するメンバー、MEM1 を指定しています。
- RECORD では、DBCS データを検査してそれが印刷可能であることを確認することを指定しています。

第 10 章 IEBUPDTE (データ・セットの更新) プログラム

IEBUPDTE は、順次データ・セット、区分データ・セット、PDSE の作成と変更に使えます。ただし、このプログラムを使用できるのは、対象となるデータ・セットに含まれている固定長レコードが 80 バイトを超えない場合に限られます。(このプログラムは主にプロシージャ、ソース、およびマクロのライブラリーの更新に使用します。JCL が入っているライブラリーはその一例です。)

IEBUPDTE を使用して、以下のタスクを実行できます。

- 区分データ・セット (PDS) ユーザーの IBM、またはユーザーによってソース言語に加えられた変更を順次データ・セット、区分データ・セット、PDSE に組み込む。
- データ・セットのライブラリーを作成および更新する。
- 既存の順次データ・セット、区分データ・セット、PDSE の既存メンバーを変更する。
- データ・セットの編成を、順次から区分または PDSE へ、あるいはその逆に変更する。

さらに、自分で作成した出口ルーチンを使ってヘッダー・ラベルやトレーラー・ラベルを処理することも可能です。

データ・セットのライブラリーの作成と更新

IEBUPDTE を使用すれば、区分メンバーの論理レコード長が 80 以下である場合に、それらのメンバーのライブラリーを作成できます。さらに、元から割り振られていたスペースにメンバーを新たに組み込むだけの余地がある場合には、メンバーを既存のライブラリーに直接追加することも可能です。この方法を使えば、カタログ式プロシージャをプロシージャ・ライブラリーの中に入れてたり、一連のジョブ制御ステートメントやユーティリティー制御ステートメントを 1 つのメンバーとして区分ライブラリーの中に入れることができます。対象となるライブラリーは、区分データ・セットと PDSE のいずれのライブラリーでも構いません。

既存データ・セットの変更

IEBUPDTE は、既存の区分データ・セット、順次データ・セット、PDSE を変更するときに使うことができます。論理レコードの置換、削除、番号変更、メンバーやデータ・セットへの追加などが可能です。

磁気テープ・ボリューム上にある順次データ・セットは、データ・セットの新規マスター (つまり、変更したコピー) を作成するのに使うことができます。直接アクセス装置上にある順次データ・セットを変更するには、新規マスターを作成するか、そのデータ・セットが置かれているボリューム上でそれを直接変更します。

区分データ・セットや PDSE を変更するには、新規マスターを作成するか、そのデータ・セットが置かれているボリューム上でそれを直接変更します。

データ・セットの編成の変更

IEBUPDTE を使用することにより、データ・セットの編成を順次から区分または PDSE に変更したり、区分データ・セットや PDSE の 1 つのメンバーを順次データ・セットに変更したりすることが可能です。メンバーを 1 つだけ変更した場合は、元のデータ・セットの残りの部分に変更されません。さらに、論理レコードの置換、削除、番号変更、メンバーやデータ・セットへの追加なども可能です。

入出力

IEBUPDTE で使用する入力は以下のとおりです。

- 変更された、あるいは新規マスターのソース・データとして使われた入力データ・セット (古いマスター・データ・セットともいいます)。この入力データ・セットは、順次データ・セット、または順次データ・セットや PDSE のメンバーのいずれかです。
- ユーティリティー制御ステートメント、および適用される場合には入力データが含まれている、制御データ・セット。このデータ・セットは IEBUPDTE を使用するたびに必要となります。

IEBUPDTE は以下のような出力を生成します。

- IEBUPDTE 操作の結果である、出力データ・セット。このデータ・セットは順次、区分、PDSE のいずれかです。また、これは新規のデータ・セット (つまり、現在のジョブ・ステップの間に作成されたもの)、あるいは現在のジョブ・ステップの間に変更された既存のデータ・セットのいずれかになります。
- ユーティリティー・プログラムの識別、ジョブ・ステップで使われている制御ステートメント、入力データ・セットに対する変更、および診断メッセージ (もしあれば) が入っている、メッセージ・データ・セット。メッセージ・データ・セットは順次です。

IEBUPDTE の戻りコードについては、付録 A を参照してください。

制御

IEBUPDTE はジョブ制御ステートメントやユーティリティー制御ステートメントで制御します。ジョブ制御ステートメントは、IEBUPDTE の処理やロード、およびそのプログラムが使用したり生成したりするデータ・セットの定義を行うときに必要になります。ユーティリティー制御ステートメントは、IEBUPDTE の機能を制御したり、場合によっては新規データや置換データを入れたりするときにも使用します。

ジョブ制御ステートメント

表 33 は、IEBUPDTE で使用するジョブ制御ステートメントをまとめたものです。

表 33. IEBUPDTE で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。

表 33. IEBUPDTE で使用するジョブ制御ステートメント (続き)

ステートメント	用途
EXEC	プログラム名を指定します (PGM=IEBUPDTE)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。それ以外の情報は、EXEC ステートメントの PARM パラメーターに指定できます。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。データ・セットはシステム出力装置、磁気テープ・ボリュウム、直接アクセス・ボリュウムなどに出力できます。
SYSUT1 DD	入力 (前のマスター) データ・セットを定義します。カード読取装置、磁気テープ・ボリュウム、直接アクセス・ボリュウム上の順次データ・セットを定義できます。また、直接アクセス・ボリュウム上の区分データ・セットや PDSE も定義できます。
SYSUT2 DD	出力データ・セットを定義します。カード穿孔装置、プリンター、磁気テープ・ボリュウム、直接アクセス装置上の順次データ・セットを定義できます。また、直接アクセス・ボリュウム上の区分データ・セットや PDSE も定義できます。
SYSIN DD	制御データ・セットを定義します。制御データ・セットは通常、入力ストリームにあります。区分データ・セットや PDSE のメンバーとして定義することもできます。

EXEC ステートメント

補足的な情報は、EXEC ステートメントの PARM パラメーターに以下のように記述できます。

//[stepname]	EXEC	PGM=IEBUPDTE [,PARM='{NEW MOD}[,inhdr][,intlr]']
--------------	------	-----------------------------------------------------

各項目の意味は以下のとおりです。

PGM=IEBUPDTE

実行するプログラムが IEBUPDTE であることを指定します。

PARM='{NEW|MOD} [,inhdr][,intlr]'

オプションの情報を以下のように指定します。

NEW

入力データを制御データ・セットのみで構成することを指定します。NEW を指定した場合は、SYSUT1 DD ステートメントを記述しないでください。

MOD

入力データを制御データ・セットと入力データ・セットの両方で構成することを指定します。デフォルトは MOD です。

inhdr

制御データ・セット用のユーザー・ヘッダー・ラベルを処理するルーチンの名前を指定します。

intlr

制御データ・セット用のユーザー・トレーラー・ラベルを処理するルーチンの名前を指定します。

PARM にコーディングする値が 1 つだけであれば、単一引用符 (または括弧) は付けなくても構いません。

SYSPRINT DD ステートメント

メッセージ・データ・セットの論理レコード長は 121 バイトであり、固定長レコード、ブロック・レコード、または非ブロック化レコードで構成されています。そして各レコードの第 1 バイトは ANSI (米国規格協会) 制御文字になっています。

SYSUT1 および SYSUT2 DD ステートメント

SYSUT1 と SYSUT2 の両方の DD ステートメントで同じ順次データ・セットを定義している場合、実行できる操作は既存のデータ・セットの末尾にデータを追加する操作に限られます。このような場合は以下の点に注意してください。

- EXEC ステートメントの PARM パラメーターが MOD を指定 (または暗黙指定) している必要があります。(237 ページの『EXEC ステートメント』を参照。)
- SYSUT1 DD ステートメントの DISP パラメーターには OLD を指定しなければなりません。

SYSUT1 と SYSUT2 が同じ区分データ・セットか PDSE を定義している場合、SYSUT2 に対する更新によって生じた新しいエクステン트는 SYSUT1 では検索できません。

入力データ・セットや出力データ・セットに入っているレコードはブロック論理レコードまたは非ブロック化論理レコードであり、レコード長は最大 80 バイトです。入力データ・セットと出力データ・セットのブロック・サイズが異なっていることもあります。あくまでも論理レコード長の倍数になっています。ただし、再ブロック化したレコードに割り当てたスペースが不十分だと、その更新要求は終了します。

EXEC ステートメントに ADD 操作を PARM=NEW で指定した場合は、SYSUT1 DD ステートメントをコーディングする必要はありません。

SYSUT1 DD ステートメントがテープ上の順次データ・セットを定義している場合は、そのデータ・セットのファイル・シーケンス番号を LABEL キーワードに組み込む必要があります (該当するデータ・セットがそのボリュームの最初または唯一のデータ・セットである場合を除きます)。

出力データ・セット (SYSUT2) がまだ存在しておらず、直接アクセス装置に置かれる予定になっている場合は、スペースを割り振る必要があります。SYSUT2 DD ステートメントには絶対に DUMMY データ・セットを指定しないでください。

ADD 機能ステートメントを使って既存の区分データ・セットまたは PDSE にメンバーを追加するときには、SYSUT1 および SYSUT2 DD ステートメント (指定された唯一のステートメントが SYSUT2 DD ステートメントである場合は、その SYSUT2 DD ステートメント) に指定する DCB パラメーターを、そのデータ・セットに対してすでに使われている既存の DCB パラメーターと同じにしなければなりません。

UPDATE=INPLACE 操作を指定した場合は、SYSUT2 DD ステートメントを記述しないでください。

SYSUT1 および SYSUT2 DD ステートメントの両方で標準ユーザー・ラベル (SUL) を指定している場合、IEBUPDTE はユーザー・ラベルを SYSUT1 から SYSUT2 にコピーします。

SYSUT1 および SYSUT2 DD ステートメントで同じ区分データ・セットまたは PDSE を定義している場合、新しいマスター・データ・セットを作成することなく古いマスター・データ・セットを更新できます。この場合、更新された 1 つまたは複数のメンバーのコピーが、古いマスター・データ・セットに割り振られていたスペースのエクステント内に書き込まれます。それ以後にこの更新メンバーを参照すると、新しく書き込まれたメンバーが参照先になります。これらのメンバー名そのものは DD ステートメントに記述せずに、IEBUPDTE 制御ステートメントを介して参照するようにしてください。なお、各メンバーの古いディレクトリーはコピーされません。

SYSIN DD ステートメント

SYSIN DD ステートメントは IEBUPDTE を使用するたびに必要となります。この制御データ・セットには 80 バイトのブロック・レコードまたは非ブロック化レコードが含まれています。

ユーティリティー制御ステートメント

表 34 には、IEBUPDTE を制御するのに使われるユーティリティー制御ステートメントが示されています。

表 34. IEBUPDTE のユーティリティー制御ステートメント

ステートメント	用途
機能	IEBUPDTE の操作を開始します (ADD、CHANGE、REPL、REPRO)。
詳細	特殊なアプリケーションに対する機能ステートメントで使います。
データ	出力データ・セットの中の新規レコードや置換レコードとして使われるデータの論理レコード。
LABEL	後続のデータ・ステートメントをユーザー・ラベルとして扱うことを示します。
ALIAS	別名を割り当てます。
ENDUP	IEBUPDTE を停止します。

要件:他のユーティリティー制御ステートメントとは異なる点として、IEBUPDTE のすべてのユーティリティー制御ステートメント (データ・ステートメントを除く) には、先頭に必ず『./』 (ピリオド、スラッシュ) を 1 と 2 桁目の位置にコーディングします。

ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

機能ステートメント

機能ステートメント (ADD、CHANGE、REPL、REPRO) は、IEBUPDTE の操作を開始するのに使います。処理するメンバーまたはデータ・セットごとに、少なくとも 1 つの機能ステートメントがなければなりません。

古いマスターに割り振られていたスペースに別のメンバーまたはデータ・セットを新たに組み込めるだけの余地があるときは、そのメンバーまたはデータ・セットを古いマスター・データ・セットに直接追加することが可能です。ADD は、メンバーまたはデータ・セットを古いマスター・データ・セットに追加することを指定します。メンバーを追加したものの、そのメンバー名が古いマスター・データ・セットにすでに存在していると処理が停止します。ただし、EXEC ステートメントに PARM=NEW が指定されていると、このメンバーは置き換えられます。順次出力マスター・データ・セットの場合には、EXEC ステートメントに PARM=NEW を常に指定しなければなりません。

あるメンバーが古いマスター・データ・セット上にある同一の名前が付いたメンバーを置き換えた場合、あるいは古いマスター上であるメンバーが変更されて再び書き込まれた場合でも、元のメンバーの別名（もしあれば）はその元のメンバーを参照しています。ただし、新しく書き込まれたメンバーに同一の別名が指定されている場合には、ディレクトリー内の元の別名項目は新しく書き込まれたメンバーを参照するように変更されます。

REPL では、データ・セットのメンバーの全体を入力して順次データ・セット、または古いマスター・データ・セットのメンバーを置き換えることを指定します。そのメンバー名は、古いマスター・データ・セットにすでに存在しているものでなければなりません。CHANGE では、既存のメンバーまたはデータ・セットに対して変更を行うことを指定します。CHANGE 機能ステートメントを使うときに NUMBER または DELETE 詳細ステートメント、あるいはデータ・ステートメントを指定しないと、エラー条件が発生します。REPRO では、メンバーまたはデータ・セットの全体を新しいマスター・データ・セットにコピーすることを指定します。

メンバーをライブラリーのコピーから論理的に削除するには、同じジョブ・ステップ内にある一連の REPRO 機能ステートメントからそれらのメンバーを削除します。

特定の 1 つのジョブ・ステップにコピーできる順次データ・セットは 1 つです。順次データ・セットを新しいボリュームから論理的に削除するには、一連のジョブ・ステップからそれらのデータ・セットを削除します。これにより、新しいボリュームには必要なデータ・セットだけがコピーされます。EXEC ステートメントに NEW サブパラメーターがコーディングされている場合、指定できるのは ADD 機能ステートメントに限定されます。

機能ステートメントの構文は次のとおりです。

/label]	{ADD CHANGE REPL REPRO}	[LIST=ALL] [,SEQFLD={ddl(ddl,ddl)}] [,NEW={PO PS}] [,MEMBER=membername] [,COLUMN={nn1}] [,UPDATE=INPLACE] [,INHDR=routinename] [,INTLR=routinename] [,OUTHDR=routinename] [,OUTTLR=routinename] [,TOTAL=(routinename,size)] [,NAME=membername] [,LEVEL=hh] [,SOURCE=x] [,SSI=hhhhhhhh]
注: COLUMN と UPDATE=INPLACE を使用できるのは、CHANGE の場合に限られます。		

各項目の意味は以下のとおりです。

LIST=ALL

SYSPRINT データ・セットに更新メンバーまたはデータ・セットの全体と、その作成時に使われた制御ステートメントとを入れることを指定します。

デフォルト: 古いデータ・セットの場合、LIST が省略されていると SYSPRINT データ・セットに入れられるのは変更内容と制御ステートメントだけになります。UPDATE が指定されたときは、番号変更が行われた場合に限り、更新メンバーの全体が表示されます。新しいデータ・セットの場合、メンバーまたはデータ・セットの全体と、その作成時に使われた制御ステートメントとは、常に SYSPRINT データ・セットに書き込まれます。

SEQFLD={ddl(ddl,ddl)}

ddl には、既存の論理レコードと後続のデータ・ステートメントの中にあるシーケンス番号の開始桁 (80 桁目が上限) および長さ (8 以下) を 10 進値で指定します。注意すべき点として、開始桁の指定値 (dd) に長さ (l) を加えた数は論理レコード長 (LRECL) + 1 を超えてはなりません。受け取るデータ・ステートメントと既存の論理レコードのシーケンス番号は、そのシーケンス・フィールドの長さがいっぱいになるまで左側から順にゼロを埋め込んでおく必要があります。

(ddl,ddl)

英数字によるシーケンス番号を生成する必要があるときに使うことができます。最初の ddl には前述の方法でシーケンス番号の桁を指定します。2 番目の ddl には、後続の NUMBER ステートメントにあるシーケンス番号の数字部分の開始桁 (80 桁目が上限) および長さ (8 以下) を 10 進値で指定します。この情報は、NEW1 パラメーターで指定したシーケンス番号において、値を増やしていく部分と、レコードを挿入または番号変更すると生成される新しいシーケンス番号にコピーすべき部分とを区別するのに使われます。

数字部分の桁は、最初の *ddl* で指定された (あるいはデフォルトになっている) シーケンス番号の桁の範囲内になければなりません。

デフォルト: 738 がデフォルトと見なされます。つまり、73 桁から始まる 8 バイトのシーケンス番号ということになります。そのため、既存の論理レコードと後続のデータ・ステートメントのシーケンス番号が 73~80 桁目に収まっていればこのキーワードをコーディングする必要はありません。

NEW={POIPS}

古いマスター・データ・セットの編成と、更新した出力データの編成を指定します。新しいマスター・データ・セットの編成が古いマスターの編成と異なっているのでない限り、NEW は指定しないでください。NEW をコーディングできるのは最初の制御レコードだけです。NAME や MEMBER と組み合わせた NEW の使用法については、245 ページの表 35 を参照してください。コーディングできる値は以下のとおりです。

PO

古いマスター・データ・セットが順次データ・セットであり、更新された出力データが区分データ・セットまたは PDSE のメンバーになることを指定します。

PS 古いマスター・データ・セットが区分データ・セットまたは PDSE であり、そのデータ・セットのメンバーが順次データ・セットに変換されることを指定します。

MEMBER=membername

SYSUT2 DD ステートメントで定義された区分データ・セットまたは PDSE の中にあるメンバーに、名前を割り当てることを指定します。MEMBER が使われるのは、SYSUT1 で定義されているのが順次データ・セットであり、SYSUT2 で定義されているのが区分データ・セットまたは PDSE であり、さらに NEW=PO が指定されている場合に限られます。NEW と組み合わせた MEMBER の使用法については、245 ページの表 35 を参照してください。

COLUMN={nm1}

論理レコード内のデータ・フィールドの開始桁を 10 進値で指定します。このフィールドは論理レコードの末尾まで拡張されます。定義したフィールド内のデータは、既存の論理レコードの中で、後続のデータ・ステートメントからのデータに置き換えられます。COLUMN の制約事項については、244 ページの『機能の制約事項』を参照してください。

COLUMN は、CHANGE の場合にのみコーディングできます。

UPDATE=INPLACE

古いマスター・データ・セットをそれが実際に占めているスペース内で更新することを指定します。古いマスター・データ・セットは直接アクセス装置上になければなりません。UPDATE=INPLACE のコーディングは、CHANGE の場合にのみ有効です。それ以外の機能ステートメント (ADD、REPL、REPRO) は同じジョブ・ステップ中に指定できません。UPDATE=INPLACE の制約事項については、244 ページの『機能の制約事項』を参照してください。

INHDR=routinename

任意のユーザー入力 (SYSUT1) ヘッダー・ラベルを処理するユーザー・ルーチンの名前を指定します。このパラメーターは、順次データ・セットを処理してい

る場合に限り有効です。ユーザー入力ヘッダー・ラベルの更新については、15 ページの表 7 にある情報を参照してください。

INTLR=routinename

任意のユーザー入力 (SYSUT1) トレーラー・ラベルを処理するユーザー・ルーチンの名前を指定します。INTLR は、処理対象が順次データ・セットであり、UPDATE=INPLACE がコーディングされていない場合に限り有効です。

OUTHDR=routinename

任意のユーザー出力 (SYSUT2) ヘッダー・ラベルを処理するユーザー・ルーチンの名前を指定します。OUTHDR は、処理対象が順次データ・セットであり、UPDATE=INPLACE がコーディングされていない場合に限り有効です。

OUTTLR=routinename

任意のユーザー出力 (SYSUT2) トレーラー・ラベルを処理するユーザー・ルーチンの名前を指定します。OUTTLR は、処理対象が順次データ・セットであり、UPDATE=INPLACE がコーディングされていない場合に限り有効です。

TOTAL=(routinename,size)

各レコードを書き込む前にあらかじめユーザー・ルーチンの出口を提供することを指定します。このパラメーターは、順次データ・セットを処理している場合に限り有効です。コーディングする値は以下のとおりです。

routinename

ユーザーの合計処理用ルーチンの名前を指定します。

size

データを入れるのに必要なバイト数を指定します。このサイズは 32K バイトを超えることはできず、かつ 2 バイト以上でなければなりません。さらに、SYSUT2 (出力) DD ステートメントにキーワード OPTCD=T を指定しておく必要があります。

ユーザー・ルーチンのリンケージ規則については、377 ページの『付録 C. ユーティリティー・プログラムのユーザー出口を指定する』の説明を参照してください。

NAME=membername

区分データ・セットまたは PDSE の中に入れるメンバーの名前を示します。DD ステートメント自体にメンバー名を指定する必要はありません。各入力メンバーを識別するために、NAME は必ず指定してください。入力が順次データ・セットの場合は、NAME を指定しないでください。NAME の使用法については、245 ページの表 35 を参照してください。このパラメーターは、区分データ・セットや PDSE のメンバーを処理している場合に限り有効です。

LEVEL=hh

変更 (更新) のレベルを 16 進値 (00~FF) で指定します。このレベル番号は、出力メンバーのディレクトリー項目に記録されます。このレベル番号は SSI (システム状況情報) の先頭バイトにあります。そのメンバー・エントリーに ISPF 情報が入っている場合、このレベル番号は ISPF 情報の一部をオーバーレイします。このパラメーターは、区分データ・セットや PDSE のメンバーを処理している場合に限り有効です。SSI が指定されている場合、LEVEL は無効です。

SOURCE=*x*

x の値が 0 のときにはユーザーによる変更を、*x* の値が 1 のときには IBM による変更を指定します。そのソースは出力メンバーのディレクトリー項目に記録されます。この値は SSI (システム状況情報) の 2 番目のバイトにあります。そのメンバー・エンタリーに ISPF 情報が入っている場合、この値は ISPF 情報の 1 ビットをオーバーレイします。このパラメーターは、区分データ・セットや PDSE のメンバーを処理している場合に限り有効です。SSI が指定されている場合、SOURCE は無効です。

SSI=hhhhhhhh

システム状況情報 (SSI) を示す 8 文字からなる 16 進文字を指定します。これは新しいマスター・データ・セットのディレクトリーの中に、パック 10 進値による 4 バイトのユーザー・データとして入れられます。そのメンバー・エンタリーに ISPF 情報が入っている場合、SSI 値は ISPF 情報の一部をオーバーレイします。PTF (IBM からのソフトウェア更新情報) は、一般にこの SSI を設定します。このパラメーターは、区分データ・セットや PDSE のメンバーを処理している場合に限り有効です。SSI は、同じ機能ステートメントに LEVEL または SOURCE パラメーターが指定されていればそれを無効にします。

機能の制約事項

UPDATE=INPLACE を指定すると以下のような制約があります。

- SYSUT2 DD ステートメントはコーディングしません。
- EXEC ステートメントの PARM パラメーターが MOD を指定 (または暗黙指定) している必要があります。
- NUMBER 詳細ステートメントを使って番号変更操作を指定できます。
- データ・ステートメントを使えるのは、置換情報を指定する場合に限られます。
- 1 つのジョブ・ステップで使用できる CHANGE 機能ステートメントと UPDATE=INPLACE パラメーターはそれぞれ 1 つです。
- 置換、番号変更、(LABEL ステートメントを利用した) ヘッダー・ラベルの変更以外の機能は指定できません。
- データ・セット全体の番号を変更した場合を除き、置換したレコードだけが一覧に表示されます。
- システム状況情報は変更できません。

REPRO を指定すると、SYSUT1 と SYSUT2 の両方が区分データ・セットまたは PDSE である場合に限り、ADD ステートメントを同じジョブ・ステップの中で使用できます。これ以外の場合は、予測できない結果が生じます。

COLUMN パラメーターで定義したフィールド内のデータは、既存の論理レコードの中で、後続のデータ・ステートメントからのデータに以下のようにして置き換えられます。

1. IEBUPDTE はデータ・ステートメントのシーケンス番号と、既存の論理レコードのシーケンス番号とを突き合わせます。このようにして COLUMN の指定内容が特定の論理レコードに適用されます。
2. データ・ステートメントの中にあるフィールドの情報は、既存の論理レコードの中にあるフィールドの情報を置き換えます。たとえば COLUMN=40 は、後続のデータ・ステートメントの 40~80 桁目 (論理レコード長を 80 バイトとした場

合) が、それと一致したシーケンス番号の示す論理レコードの 40~80 桁目にあるデータを置き換えることを示します。(既存の論理レコードまたはデータ・ステートメント内のシーケンス番号は、定義されたフィールドの中にある必要はありません。)

COLUMN の指定内容は機能全体に適用されますが、例外として以下のものには適用されません。

- 後続の DELETE 詳細ステートメントで削除される論理レコード。
- 既存の論理レコードと一致しているシーケンス番号がない、後続のデータ・ステートメント。
- 削除した 1 つまたは複数の論理レコードのあった場所に挿入すべき情報が入ったデータ・ステートメント。

表 35 では、入力および出力データ・セットの編成が異なる場合に NEW、MEMBER、NAME の各パラメーターを使用する方法を示しています。

表 35. 機能ステートメントの NEW、MEMBER、および NAME パラメーター

入力データ・セットの編成	出力データ・セットの編成	パラメーターの組み合わせ
なし	区分または PDSE (新規)	それぞれの ADD 機能ステートメントで、データ・セットに入れる各メンバーの名前を NAME を使って割り当てます。
区分または PDSE	区分または PDSE	ADD 機能ステートメントでは、SYSUT2 DD ステートメントで定義したデータ・セットに入れるメンバーの名前を NAME を使って指定します。さらに別の名前が必要な場合は、ALIAS ステートメントも併用できます。 CHANGE、REPL、または REPRO 機能ステートメントでは、SYSUT1 DD ステートメントで定義したデータ・セットの中にあるメンバーの名前を NAME を使って指定します。SYSUT2 DD ステートメントで定義したデータ・セット内のメンバーに別の名前、あるいは補足的な名前を付ける必要がある場合は、ALIAS ステートメントも併用します。
区分または PDSE	順次	どの機能ステートメントでも、SYSUT1 DD ステートメントで定義したデータ・セット内のメンバーの名前は NAME を使って指定します。編成を区分から順次に変更する場合は、NEW=PS を使ってそのことを指定します。(出力マスター・データ・セットに割り当てられた名前とファイル・シーケンス番号 (もしあれば) は、SYSUT2 DD ステートメントに指定します。)
順次	区分または PDSE	適用可能な機能ステートメントでは、SYSUT2 DD ステートメントで定義したデータ・セットに入れるメンバーに名前を割り当てるときに MEMBER を使用します。編成を順次から区分に変更する場合は、NEW=PO を使ってそのことを指定します。

詳細ステートメント

詳細ステートメントは、特定のアプリケーション、たとえば選択した論理レコードを削除したりその番号を変更したりするアプリケーションに対する機能ステートメントと併用します。CHANGE 機能ステートメントと同時にコーディングした場合、NUMBER 詳細ステートメントでは、1 つまたは複数の論理レコードのシーケンス番号を変更することを指定します。ADD または REPL 機能ステートメントと同時にコーディングした場合は、新しい、あるいは置き換えるメンバーやデータ・セットの中にあるレコードに割り当てるシーケンス番号を指定します。ADD また

は REPL 機能ステートメントと同時に使用するときは、それぞれの ADD または REPL 機能ステートメントに NUMBER 詳細ステートメントを 2 つ以上指定することはできません。

CHANGE 機能ステートメントと同時にコーディングした場合、DELETE 詳細ステートメントでは、メンバーまたはデータ・セットから削除する 1 つまたは複数の論理レコードを指定します。

論理レコードは部分的に削除することはできません。それで、機能ステートメント内の COLUMN パラメーターの指定内容は、削除するレコードには適用されません。特定のシーケンス番号はそれぞれ 1 つの操作で 1 回だけ処理されます。

DETAIL ステートメントの構文は次のとおりです。

./[label]	{NUMBER DELETE}	[SEQ1={ccccccc ALL}] [,SEQ2=ccccccc] [,NEW1=ccccccc] [,INCR=ccccccc] [,INSERT=YES]
注: NEW1、INCR、および INSERT を使えるのは NUMBER の場合に限られます。		

各項目の意味は以下のとおりです。

SEQ1={ccccccc|ALL}

番号変更、削除、またはシーケンス番号の割り当てを行うレコードを指定します。コーディングできる値は以下のとおりです。

ccccccc

番号変更または削除を行う最初の論理レコードのシーケンス番号を指定します。この値は、ADD または REPL 機能ステートメントと同時に使用する NUMBER 詳細ステートメントにはコーディングしません。この値を挿入操作で使用すると、挿入を実行した後の既存の論理レコードが指定されます。これは、置換または追加したばかりのステートメントの数と同じではありません。詳細については、INSERT パラメーターの項を参照してください。

ALL

メンバーまたはデータ・セットの全体に対する番号変更操作を指定します。ALL を使用するのには、CHANGE 機能ステートメントと NUMBER 詳細ステートメントが使われている場合に限られます。シーケンス番号がブランクになっている既存の論理レコードにシーケンス番号を割り当てる場合には ALL を必ずコーディングしてください。ALL をコーディングしないと、シーケンス番号がブランクになっているすべての既存の論理レコードは出力マスター・データ・セットに直接コピーされます。ALL をコーディングすれば SEQ2 をコーディングする必要はなく、1 つの機能ステートメントにつき指定できる NUMBER 詳細ステートメントは 1 つになります。詳細については、INSERT パラメーターの項を参照してください。

SEQ2=ccccccc

番号変更または削除を行う最後の論理レコードのシーケンス番号を指定します。DELETE 詳細ステートメントにはすべて SEQ2 を指定する必要があります。削

除するレコードが 1 つだけの場合は、SEQ1 と SEQ2 の指定内容が同一でなければなりません。SEQ2 は、ADD または REPL 機能ステートメントと同時に使用する NUMBER 詳細ステートメントにはコーディングしません。

NEW1=ccccccc

新しい、または置き換えるデータに割り当てる最初のシーケンス番号を指定します。あるいは、番号変更操作で割り当てる最初のシーケンス番号を指定します。NEW1 に指定する値は SEQ1 に指定する値より大きくなければなりません (SEQ1=ALL が指定されている場合は例外で、この規則は適用されません)。

INCR=ccccccc

新しい、または置き換える論理レコードに連続したシーケンス番号を割り当てる時に使う増分値を指定します。あるいは、既存の論理レコードの番号を変更する時に使う増分値を指定します。

INSERT=YES

論理レコードのブロックを挿入することを指定します。レコード (シーケンス番号がブランクになっているデータ・ステートメント) に番号が割り振られて、出力マスター・データ・セットに挿入されます。INSERT は、CHANGE 機能ステートメントと NUMBER 詳細ステートメントの両方が同時にコーディングされている場合にのみ有効です。最初の NUMBER 詳細ステートメントには SEQ1、NEW1、および INCR を必ず指定しなければなりません。

INSERT=YES をコーディングすると以下の点が適用されます。

- SEQ1 パラメーターでは、挿入を実行した後の既存の論理レコードを指定します。SEQ1=ALL をコーディングすることはできません。
- SEQ2 パラメーターをコーディングする必要はありません。
- NEW1 パラメーターでは、挿入する最初の論理レコードにシーケンス番号を割り当てます。このパラメーターが英数字の場合は、機能ステートメントに SEQFLD=(ddl,ddl) パラメーターをコーディングしなければなりません。
- INCR パラメーターは、最初の挿入を行った時点からのメンバーまたはデータ・セットの番号を必要な数だけ変更するのに使われます。メンバーまたはデータ・セットの番号変更は、割り当てる次のシーケンス番号と同じかそれよりも大きいシーケンス番号を持つ既存の論理レコードが検出されるまで行われます。そのような論理レコードが検出されないときは、メンバーまたはデータ・セット全体の番号が変更されます。
- さらに別の NUMBER 詳細ステートメントがある場合には、それらのステートメントに INSERT=YES を指定する必要があります。以前の番号割り当ての操作により、後続の NUMBER 詳細ステートメントの SEQ1 パラメーターに指定した論理レコードの番号が変更されたときには、後の方の NUMBER 詳細ステートメントにある任意の NEW1 または INCR パラメーターは指定変更されます。以前の増分値は、連続した次のシーケンス番号を割り当てるのに使われます。以前の番号割り当ての操作では後続の NUMBER 詳細ステートメントの SEQ1 パラメーターに指定した論理レコードの番号が変更されなかった場合は、後の方のステートメントに NEW1 または INCR を必ず指定しなければなりません。
- 挿入するデータ・ステートメントのブロックのシーケンス番号はブランクにしておく必要があります。

- 機能ステートメント、詳細ステートメント、ファイル終わり (EOF) 標識、シーケンス番号が入っているデータ・ステートメントのいずれかが検出されると、挿入操作が停止します。

詳細な制約事項

SEQ1、SEQ2、NEW1 の各パラメーター (SEQ1=ALL は例外) には、英数字で 8 文字 (最大) を指定します。INCR パラメーターには、数字で 8 文字 (最大) を指定します。コーディングする必要があるのは数字のシーケンス番号の有効部分だけです。たとえば、SEQ1=00000010 であれば SEQ1=10 のように短く指定できます。ただし、この番号が英数字であれば、英字の部分は必ず指定してください。たとえば、SEQ1=00ABC010 を短く指定するときは SEQ1=ABC010 のようにします。

データ・ステートメント

データ・ステートメントは機能ステートメントと、あるいは機能ステートメントと詳細ステートメントの両方と併用できます。データ・ステートメントには、既存の論理レコードの置換データとして使う論理レコード、あるいは出力マスター・データ・セットに組み込む新しいデータを入れます。

データ・ステートメントにはそれぞれ 1 つの論理レコードが入っており、その論理レコードはデータ・ステートメントの先頭桁から始まっています。この論理レコードの長さは、出力マスター・データ・セットに指定した論理レコード長 (LRECL) と同じです。各論理レコードには、そのデータが出力マスター・データ・セットのどこにあるかを判別するためのシーケンス番号が入っています (NUMBER ステートメントで INSERT=YES が指定されている場合を除きます)。

CHANGE 機能ステートメントと同時に使用する場合、データ・ステートメントには以下のようにして新しい、あるいは置き換えるデータを入れます。

- データ・ステートメント内のシーケンス番号が、既存の論理レコード内のシーケンス番号と同一である場合、そのデータ・ステートメントは出力マスター・データ・セット内の該当する既存の論理レコードを置き換えます。
- 対応するシーケンス番号が既存のレコードの中で見つからない場合、そのデータ・ステートメントは出力マスター・データ・セットの中で適切な照合シーケンスに従って挿入されます。(この機能を正しく実行するには、古いマスター・データ・セットの中のすべてのレコードにシーケンス番号がなければなりません。)
- シーケンス番号を持つデータ・ステートメントが使われており、INSERT=YES が指定された場合は、挿入操作が停止します。このシーケンス番号が次の古いマスター・レコード (参照されているシーケンス・レコードの次にくるレコード) と少なくとも等しいときには、IEBUPDTE は処理を継続します。

ADD または REPL 機能ステートメントと同時に使用する場合、データ・ステートメントには出力マスター・データ・セットに入れる新しいデータを入れておきます。

古いマスター・データ・セットの中のシーケンス番号は昇順になっていると見なされます。シーケンス番号の妥当性検査は、データ・ステートメントや既存のレコードに対しては実行されません。

データ・ステートメント内のシーケンス番号は、既存の論理レコード内のシーケンス番号と同じ相対位置になければなりません。(シーケンス番号には先行ゼロが含

まれており、73~80 桁目にあると見なされます。番号がそれ以外の桁にあるときは、その前にある機能ステートメントの中の SEQFLD パラメーターで長さや相対位置を指定することが必要です。)

LABEL ステートメント

LABEL ステートメントは、その後にくるデータ・ステートメント (ラベル・データ・ステートメントという) をユーザー・ラベルとして扱うことを示します。これらの新しいユーザー・ラベルは出力データ・セットに置かれます。その次の制御ステートメントでは、グループの最後のラベル・データ・ステートメントがすでに読み取られていることを IEBUPDTE に示します。

LABEL ステートメントの構文は次のとおりです。

./[name]	LABEL	
----------	-------	--

LABEL ステートメントを IEBUPDTE の 1 回の実行につき 2 回まで指定できます。また、どの LABEL ステートメントの後にも 8 個を超えるラベル・データ・ステートメントを指定することはできません。80 バイトからなる各ラベル・データ・ステートメントの先頭 4 バイトはユーザー・ラベルに関する IBM 規則に従い、『UHL*n*』または『UTL*n*』(*n* は 1~8 で、それぞれ入力ヘッダー・ラベルと入力トレーラー・ラベルを表します) になっていなければなりません。そのような場合、データ管理によって該当するデータが適切な 4 文字でオーバーレイされます。

IEBUPDTE は LABEL ステートメントを検出するとデータ・ステートメントを 8 個まで読み取り、ユーザー出力ラベル・ルーチンがそれら进行处理できるように保管します。これに該当するルーチンがない場合は、保管されたレコードは OPEN または CLOSE によって出力データ・セットにユーザー・ラベルとして書き込まれます。ユーザー出力ラベル処理ルーチンがある場合は、IEBUPDTE はパラメーター・リストをその出力ラベル・ルーチンに渡します。(377 ページの『付録 C. ユーティリティー・プログラムのユーザー出口を指定する』を参照。) ラベル・バッファに入っているラベル・データ・レコードは、ラベルとして出力する前にユーザー・ルーチンで処理することが可能です。ユーザー・ルーチンで (IEBUPDTE に対する戻りコードによって) 指定した項目の数がこのバッファにあるラベル・データ・レコードの数よりも多いと、ユーザー・ルーチンで指定した残りの項目に対する無意味な情報がラベル・バッファに入れられます。

SYSIN データ・セットにおける LABEL ステートメントの (他の機能ステートメントに対する相対的な) 位置は、その LABEL ステートメントの後にくるユーザー・ラベルの種類を以下のように示します。

- 出力ヘッダー・ラベルを作成するには、LABEL ステートメントとそれに関連しているラベル・データ・ステートメントを、同じ入力ストリームの中の他のすべての機能ステートメントの前に置きます。LABEL を除く機能ステートメントは、そのグループの最後のラベル・データ・ステートメントの後に置かなければなりません。
- 出力トレーラー・ラベルを作成するには、LABEL ステートメントとそれに関連しているラベル・データ・ステートメントを、同じ入力ストリームの中の他のすべての機能ステートメントの後、かつ ENDUP ステートメントの前に置きます。

この場合、ENDUP ステートメントはオプションではありません。IEBUPDTE が出力トレーラー・ラベルを作成する場合は、グループの最後のラベル・データ・ステートメントの後にこのステートメントを指定する必要があります。

CHANGE ステートメントで UPDATE=INPLACE を指定すればユーザー入力ヘッダー・ラベルをユーザー・ルーチンで更新できますが、入力トレーラー・ラベルと出力ラベルはユーザー・ルーチンでは更新できません。ユーザー・ラベルは追加や削除が不可能です。ユーザー入力ヘッダー・ラベルをユーザー・ルーチンが処理できるようにするには、パラメーター・リストにラベル・バッファのアドレスを指定しておきます。(ユーティリティ・プログラムとユーザー・ラベル処理ルーチンとの間のリンケージについて詳しくは、382 ページの『ユーザー・ラベルの処理』を参照してください。) CHANGE UPDATE=INPLACE が使われているときの戻りコードは、標準の戻りコードとは多少異なっています。該当する戻りコードについては、377 ページの『付録 C. ユーティリティ・プログラムのユーザー出口を指定する』を参照してください。

古いマスター・データ・セットから置き換えたラベルを調べるには、次のことを行う必要があります。

1. UPDATE=INPLACE パラメーターを機能ステートメントにコーディングすることにより、古いマスターの更新を指定する。
2. ヘッダー・ラベルまたはトレーラー・ラベルのいずれかの入力データ・セットに LABEL ステートメントを組み込む。
3. 対応するユーザー・ラベル・ルーチンを指定する。

上記の条件を満たしている場合には、標準のパラメーター・リストに 4 番目と 5 番目のパラメーター・ワードが追加されます。4 番目のパラメーター・ワードは現時点では使われませんが、5 番目には古いマスターから置き換えたラベルへのポインターが入っています。この場合、SYSIN データ・セットに指定したラベルの数は、古いマスター・データ・セット上のラベルの数を超えてはなりません。ユーザーのヘッダー・ラベル・ルーチンに指定した項目の数が入力ストリームの中にあるラベルの数を上回っていると、最初のパラメーターが指し示すヘッダー・ラベルは、上回った分の項目の古いマスター・データ・セットにある現在のヘッダー・ラベルになります。この場合、5 番目のパラメーターは無意味になります。

ALIAS ステートメント

ALIAS ステートメントは、別名を (区分または PDSE による) 出力ディレクトリーで作成したり保存したりするのに使います。ALIAS ステートメントはどの機能ステートメントにおいても併用できます。1 つのメンバーにつき最大 16 までの複数の別名を割り当てることが可能です。

データ・セットにすでに存在する名前を ALIAS ステートメントで指定すると、そのディレクトリー項目の元の TTR (トラック・レコード) が破棄されます。

ALIAS ステートメントを使用する場合、入力ストリームの中にデータ・ステートメントがあれば必ずその後ろに指定しなければなりません。

ALIAS ステートメントの構文は次のとおりです。

./[label]	ALIAS	NAME=aliasname
-----------	-------	----------------

各項目の意味は以下のとおりです。

NAME=aliasname

1～8 文字の別名を指定します。

ENDUP ステートメント

ENDUP ステートメントはオプションです。これは SYSIN の入力データの終わりをそのジョブ・ステップに示すのに使います。それより前に区切りステートメントがない場合は、このステートメントはデータ終わり (EOD) 標識としての役目を果たします。ENDUP ステートメントは SYSIN 制御ステートメントの最後のグループの後に指定します。

ENDUP ステートメントを使用する場合、IEBUPDTE を使って出力トレーラー・ラベルを作成しているときにはそれを最後のラベル・データ・ステートメントの後に指定しなければなりません。

ENDUP ステートメントの構文は次のとおりです。

/[label]	ENDUP	
----------	--------------	--

IEBUPDTE の例

以下の各例では、IEBUPDTE の使用方法の一部を示します。表 36 は IEBUPDTE の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 36. IEBUPDTE の例の一覧表

操作	データ・セット編成	装置	注釈	例
ADD および REPL	区分	ディスク	JCL プロシージャがプロシージャ・ライブラリー (PROCLIB) の新しいメンバーとして保管されます。別の JCL プロシージャは PROCLIB 内の既存のメンバーを置き換えます。	1
COPY	順次	ディスク	順次データ・セットが 1 つの直接アクセス・ボリュームから別の直接アクセス・ボリュームにコピーされます。ユーザー・ラベルは出口ルーチンで処理することができます。	10
CREATE	区分	ディスク	新しい世代を作成します。	11
CREATE	順次	カード読取装置 およびディスク	ユーザー・ラベルが指定された順次データ・セットをカード入力から作成します。	9
CREATE 区分データ・セット	区分	ディスク	制御データ・セットからの入力データ、および既存の区分データ・セットからの入力データ。出力区分データ・セットには 3 つのメンバーが入ります。	3
CREATE 区分ライブラリー	区分	ディスク	入力データは制御データ・セットにあります。出力区分データ・セットには 3 つのメンバーが入ります。	2

表 36. IEBUPDTE の例の一覧表 (続き)

操作	データ・セット編成	装置	注釈	例
CREATE および DELETE	区分、順次	ディスクおよび 磁気テープ	区分ディスク入力から順次マスターが作成 されます。選択レコードが削除されます。 出力はブロック化されています。	5
CREATE、DELETE、 および UPDATE	順次、区分	磁気テープおよ びディスク	順次入力から区分データ・セットが作成さ れます。レコードの削除と更新が行われま す。シーケンス番号は 73~80 桁目以外の 桁に入られます。出力データ・セットに は 1 つのメンバーが入られます。	6
INSERT	区分	ディスク	論理レコードのブロックが既存のメンバー に挿入されます。	7
INSERT	区分	ディスク	論理レコードのブロックが 2 つ、既存の メンバーに挿入されます。シーケンス番号 は英数字です。	8
UPDATE INPLACE および番号変更	区分	ディスク	入力データ・セットは出力データ・セット とも見なされます。そのため、SYSUT2 DD ステートメントは不要です。	4

各例で、実際の装置番号の代わりに **disk** または **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: 2 つのプロシージャーを SYS1.PROCLIB に入れる

この例では、カタログ式プロシージャー・ライブラリー、SYS1.PROCLIB に 2 つのプロシージャーを入れています。ここでは、対象となる 2 つのプロシージャーが、プロシージャー・ライブラリーに元から割り当てられているスペースに収まることを想定しています。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE, PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PROCLIB, DISP=OLD
//SYSUT2 DD DSN=SYS1.PROCLIB, DISP=OLD
//SYSIN DD DATA
./ ADD LIST=ALL, NAME=ERASE, LEVEL=01, SOURCE=0
./ NUMBER NEW1=10, INCR=10
//ERASE EXEC PGM=IEBUPDTE
//DD1 DD UNIT=disk, DISP=(OLD, KEEP), VOLUME=SER=111111
//SYSPRINT DD SYSOUT=A
./ REPL LIST=ALL, NAME=LISTPROC
./ NUMBER NEW1=10, INCR=10
//LIST EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=SHR, DSN=SYS1.PROCLIB(&MEMBER)
//SYSUT2 DD SYSOUT=A, DCB=(RECFM=F, BLKSIZE=80)
//SYSIN DD DATA
./ ENDUP
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 および SYSUT2 DD では、SYS1.PROCLIB データ・セット (カタログされていることを想定) を定義しています。

- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。このデータ・セットの中にはユーティリティ制御ステートメントと、プロシージャ・ライブラリーに入れるデータとがあります。
- ADD 機能ステートメントでは、制御データ・セット内のレコード (データ・ステートメント) を出力データに入れることを示しています。新しく作成したプロシージャ、ERASE はメッセージ・データ・セットに出力されます。

SYSUT2 が参照する新しいマスター・データ・セットに ERASE という名前のメンバーがすでに存在していると、この ADD 機能は実行されません。

- 最初の NUMBER 詳細ステートメントでは、新しい、あるいは置き換えるプロシージャにシーケンス番号を割り当てることを示しています。各プロシージャの最初のレコードには 10、次のレコードには 20、という具合にシーケンス番号が割り当てられていきます。
- ERASE EXEC ステートメントは、最初の新しいプロシージャをそこから開始することを示しています。
- REPL 機能ステートメントでは、制御データ・セット内のレコード (データ・ステートメント) が既存メンバーを置き換えることを示しています。このメンバーは、SYSUT2 で参照している新しいマスター・データ・セットに保管されています。この REPL 機能が実行されるのは、SYSUT1 が参照する古いマスター・データ・セットに LISTPROC という名前のメンバーがすでに存在している場合に限られます。
- 2 番目の NUMBER 詳細ステートメントは、最初のものと同じです。
- LIST EXEC ステートメントは、2 番目の新しいプロシージャをそこから開始することを示しています。
- ENDUP ステートメントは、SYSIN DD 入力データがそこで終わることを示しています。

例 2: 3 つのメンバーからなるライブラリーを作成する

この例では、3 つのメンバーからなる区分ライブラリーを作成します。入力データは制御データ・セットにしか入っていません。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSNAME=OUTLIB,UNIT=disk,DISP=(NEW,CATLG),
//          VOLUME=SER=111112,SPACE=(TRK,(50,,10)),
//          DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN DD DATA
./ ADD NAME=MEMB1,LEVEL=00,SOURCE=0,LIST=ALL

(データ・ステートメント、73 から 80 桁目にシーケンス番号)

./ ADD NAME=MEMB2,LEVEL=00,SOURCE=0,LIST=ALL

(データ・ステートメント、73 から 80 桁目にシーケンス番号)

./ ADD NAME=MEMB3,LEVEL=00,SOURCE=0,LIST=ALL

(データ・ステートメント、73 から 80 桁目にシーケンス番号)

./ ENDUP
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT2 DD** では、新しい区分マスターである **OUTLIB** を定義しています。それ以後の変更を、新しいマスター・データ・セットを作成することなく実行できるだけのスペースが割り振られます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。このデータ・セットの中にはユーティリティ制御ステートメントと、出力区分データ・セットに 3 つのメンバーとして入れるデータとがあります。
- **ADD** 機能ステートメントでは、後続のデータ・ステートメントをメンバーとして出力区分データ・セットに入れることを示しています。それぞれの **ADD** 機能ステートメントは後続のデータのメンバー名を指定しており、そのメンバーと制御ステートメントをメッセージ・データ・セットに出力することを示しています。
- データ・ステートメントの中には、出力区分データ・セットの各メンバーに入れるデータがあります。
- **ENDUP** は、制御データ・セットの入力データがそこで終わっていることを示しています。

シーケンス番号はデータ・ステートメントに含まれているため (番号がブランクの場合を除く)、この例では **NUMBER** 詳細ステートメントが含まれていません。

例 3: **SYS1.MACLIB** をソースにして新しいライブラリーを作成する

この例では、3 つのメンバーからなる区分データ・セット (**NEWMCLIB**) を作成します。このデータ・セットには、既存の区分データ・セット (**SYS1.MACLIB**) からコピーした **ATTACH** と **DETACH** という 2 つのメンバー、および制御データ・セットの中にある **EXIT** という新しいメンバーが入ることになります。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT2 DD DSN=DEV.DRIVER3.NEWMCLIB,UNIT=disk,
//          DISP=(NEW,CATLG),SPACE=(TRK,(100,,10)),
//          DCB=(RECFM=F,LRECL=80,BLKSIZE=4000)
//SYSIN DD DATA
./ REPRO NAME=ATTACH,LEVEL=00,SOURCE=1,LIST=ALL
./ REPRO NAME=DETACH,LEVEL=00,SOURCE=1,LIST=ALL
./ ADD NAME=EXIT,LEVEL=00,SOURCE=1,LIST=ALL
./ NUMBER NEW1=10,INCR=100
```

(EXIT メンバーのデータ・レコード)

```
./ ENDUP
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、入力区分データ・セット **SYS1.MACLIB** (カタログされていることを想定) を定義しています。
- **SYSUT2 DD** では、出力区分データ・セット **DEV.DRIVER3,NEWMCLIB** を定義しています。それ以後の変更を、新しいマスター・データ・セットを作成することなく実行できるだけのスペースが割り振られます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。

- REPRO 機能ステートメントでは、出力データ・セットにコピーする既存の入力メンバー (ATTACH と DETACH) を識別しています。これらのメンバーは、メッセージ・データ・セットにも出力されます (LIST=ALL が指定されているため)。
- ADD 機能ステートメントでは、レコード (後続のデータ・ステートメント) をメンバーとして EXIT という出力区分データ・セットに入れることを示しています。このデータ・ステートメントはメッセージ・データ・セットに出力されません。
- NUMBER 詳細ステートメントでは、データ・ステートメントにシーケンス番号を割り当てています。(このデータ・ステートメントの中の 73~80 桁目は、シーケンス番号がブランクになっています。) 出力メンバーの最初のレコードにはシーケンス番号として 10 が割り当てられ、それ以後のレコード番号は 100 ずつ大きくなっていきます。
- ENDUP は、SYSIN データがそこで終わっていることを示しています。

なお、指定されている 3 つの入力メンバー (ATTACH、DETACH、EXIT) は、古いマスターでの照合シーケンスどおりに指定する必要はありません。

例 4: ライブラリー・メンバーを更新する

この例では、メンバー (MODMEMB) をそれが実際に占めているスペースの範囲内で更新します。既存の 2 つの論理レコードが置き換えられ、メンバー全体の番号が変更されます。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=PDS,UNIT=disk,DISP=(OLD,KEEP),
//          VOLUME=SER=111112
//SYSIN DD *
./ CHANGE NAME=MODMEMB,LIST=ALL,UPDATE=INPLACE
./ NUMBER SEQ1=ALL,NEW1=10,INCR=5
```

(データ・ステートメント 1, シーケンス番号 00000020)
(データ・ステートメント 2, シーケンス番号 00000035)

/*

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、更新を同じ場所で行う区分データ・セットを定義します。(メンバー名を DD ステートメントに指定する必要がないことに注意してください。)
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- CHANGE 機能ステートメントでは、更新するメンバーの名前を示し (MODMEMB)、UPDATE=INPLACE 操作を指定しています。このメンバーの全体はメッセージ・データ・セットに出力されます。注意すべき点として、ここでは UPDATE=INPLACE を指定しているため、LIST=ALL パラメーターを指定していなかったとしても、番号変更の操作が完了した時点でこの出力が行われます。詳細については LIST パラメーターの項を参照してください。
- NUMBER 詳細ステートメントでは、メンバーの全体で番号を変更することを示し、割り当てる最初のシーケンス番号とそれ以降のシーケンス番号のための増分値 (5) を指定しています。

- データ・ステートメントは、シーケンス番号が 20 と 35 になっている既存の論理レコードを置き換えています。

例 5: 新しいマスター・データ・セットの作成と選択レコードの削除

この例では、新しいマスター順次データ・セットを区分入力データから作成し、選択した論理レコードを削除しています。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=DCB.PARTDS,DISP=(OLD,KEEP)
//          VOLUME=SER=111112
//SYSUT2 DD DSN=SEQDS,UNIT=tape,LABEL=(2,SL),
//          DISP=(,KEEP),VOLUME=SER=001234,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000)
//SYSIN DD *
./ CHANGE NEW=PS,NAME=OLDMEMB1
```

(データ・ステートメント 1, シーケンス番号 00000123)

```
./ DELETE SEQ1=223,SEQ2=246
```

(データ・ステートメント 2, シーケンス番号 00000224)

/*

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ディスク・ボリューム上に置かれている入力区分データ・セット DCB.PARTDS を定義しています。
- SYSUT2 DD では、出力順次データ・セット SEQDS を定義しています。このデータ・セットは、磁気テープ・ボリューム上の 2 番目のデータ・セットとして書き込まれます。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- CHANGE では、入力メンバー (OLDMEMB1) を識別し、出力データが順次データ・セットであることを示しています (NEW=PS)。
- 最初のデータ・ステートメントでは、シーケンス番号がこのデータ・ステートメント内のシーケンス番号 (00000123) と同一である論理レコードを置き換えています。該当する論理レコードが存在しなければ、このデータ・ステートメントは出力データ・セットの中に適切なシーケンスで組み込まれます。
- DELETE 詳細ステートメントでは、シーケンス番号が 223~246 の範囲にある論理レコードを削除しています。
- 2 番目のデータ・ステートメントは、シーケンス番号が 224 になっている論理レコードが存在しないため (該当するレコードは直前のステートメントで削除されています)、出力データ・セットの中に適切なシーケンスで組み込まれます。

編成を順次に変換するときに入力データとして使用できるメンバーは 1 つだけである点に注意してください。

例 6: ライブラリー・メンバーの作成と更新

この例では、区分データ・セットのメンバーを順次入力データから作成し、既存の論理レコードを更新します。

```

//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=BROWN.OLDSEQDS,UNIT=tape,
// DISP=(OLD,KEEP),VOLUME=SER=001234
//SYSUT2 DD DSN=BROWN.NEWPART,UNIT=disk,DISP=(,CATLG),
// VOLUME=SER=111112,SPACE=(TRK,(10,5,5)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=4080)
//SYSIN DD *
./ CHANGE NEW=PO,MEMBER=PARMEM1,LEVEL=01,
./ SEQFLD=605,COLUMN=40,SOURCE=0

(データ・ステートメント 1, シーケンス番号 00020)

./ DELETE SEQ1=220,SEQ2=250

(データ・ステートメント 2, シーケンス番号 00230)
(データ・ステートメント 3, シーケンス番号 00260)

./ ALIAS NAME=MEMB1
/*

```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、入力順次データ・セット (**BROWN.OLDSEQDS**) を定義しています。このデータ・セットはテープ・ボリューム上にあります。
- **SYSUT2 DD** では、出力区分データ・セット (**BROWN.NEWPART**) を定義しています。将来追加するかもしれないメンバーを入れるのに十分なスペースが割り振られます。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **CHANGE** ステートメントでは、出力メンバー (**PARMEM1**) を識別し、順次入力データから区分出力データへの変換を行うことを示しています。 **SEQFLD** パラメーターでは、5 バイトからなるシーケンス番号が各データ・ステートメントの 60~64 桁目に位置していることを示しています。 **COLUMN=40** パラメーターでは、置換情報の取得元となる (後続のデータ・ステートメント内の) フィールドの開始桁を指定しています。 **SOURCE=0** では、置換情報をユーザーが提供することを示しています。
- 最初のデータ・ステートメントは置換データとして使われています。このステートメントの 40~80 桁目は、対応する論理レコードの 40~80 桁目を置き換えます。該当する論理レコードが存在しなければ、カード全体のイメージが出力データ・セットのメンバーに挿入されます。
- **DELETE** 詳細ステートメントでは、シーケンス番号が 220~250 である論理レコードをすべて削除します。
- 2 番目のデータ・ステートメントは、上記の **DELETE** 詳細ステートメントに指定した範囲内にそのシーケンス番号があるため、その全体が出力データ・セットのメンバーに組み込まれます。
- 3 番目のデータ・ステートメントは、**DELETE** 詳細ステートメントの範囲外にあるため、最初のデータ・ステートメントと同じ方法で扱われます。
- **ALIAS** では、出力データ・セットのメンバー **PARMEM1** に別名 **MEMB1** を割り当てています。

例 7: ライブラリー・メンバーにレコードを挿入する

この例では、3つの論理レコードのブロックが既存のメンバーに挿入され、更新したメンバーが既存の区分データ・セットに入れます。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=DSNAME=PDS,UNIT=disk,DISP=(OLD,KEEP),
//          VOLUME=SER=111112
//SYSUT2 DD DSN=DSNAME=PDS,UNIT=disk,DISP=(OLD,KEEP),
//          VOLUME=SER=111112
//SYSIN DD *
./ CHANGE NAME=RENUM,LIST=ALL,LEVEL=01,SOURCE=0
./ NUMBER SEQ1=15,NEW1=20,INCR=5,INSERT=YES
```

(データ・ステートメント 1)
(データ・ステートメント 2)
(データ・ステートメント 3)

/*

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 および SYSUT2 DD では、区分データ・セット (PDS) を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- CHANGE 機能ステートメントでは、入力メンバー RENUM を識別しています。このメンバーの全体はメッセージ・データ・セットに出力されます。
- NUMBER 詳細ステートメントでは、挿入操作を指定し、後述する方法で番号変更操作を制御しています。
- データ・ステートメントは、挿入する論理レコードになっています。(シーケンス番号はデータ・ステートメントの挿入時に割り当てられます。)

この例では、既存の論理レコードのシーケンス番号はそれぞれ 10、15、20、25、30 となっています。シーケンス番号は NUMBER 詳細ステートメントによって以下のように割り当てられています。

1. データ・ステートメント 1 にはシーケンス番号 20 が割り当てられ (NEW1=20)、既存の論理レコード 15 の後ろに挿入されます (SEQ1=15)。
2. データ・ステートメント 2 および 3 にはシーケンス番号 25 および 30 が割り当てられ (INCR=5)、データ・ステートメント 1 の後ろに挿入されます。
3. 既存の論理レコード 20、25、30 には、シーケンス番号としてそれぞれ 35、40、45 が割り当てられます。

表 37 には、既存のシーケンス番号、挿入されるデータ・ステートメント、そして結果的に割り当てられる新しいシーケンス番号が示されています。

表 37. 番号変更されるシーケンス番号の例

シーケンス番号と挿入される	
データ・ステートメント	新しいシーケンス番号
10	10
15	15
データ・ステートメント 1	20

表 37. 番号変更されるシーケンス番号の例 (続き)

シーケンス番号と挿入される データ・ステートメント	新しいシーケンス番号
データ・ステートメント 2	25
データ・ステートメント 3	30
20	35
25	40
30	45

例 8: レコードの番号を変更してライブラリー・メンバーに挿入する

この例では、既存のメンバーに 2 つのブロック (1 つのブロックにつき 3 つの論理レコード) が挿入され、そのメンバーは区分データ・セットに入れます。出力メンバーの一部分も番号が変更されます。

```
//UPDATE JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=PDSD,UNIT=disk,DISP=(OLD,KEEP),
//          VOLUME=SER=111112
//SYSUT2 DD DSN=PDSD,UNIT=disk,DISP=(OLD,KEEP),
//          VOLUME=SER=111112
//SYSIN DD *
./ CHANGE NAME=RENUM,LIST=ALL,LEVEL=01,SOURCE=0,SEQFLD=(765,783)
./ NUMBER SEQ1=AA015,NEW1=AA020,INCR=5,INSERT=YES

(データ・ステートメント 1)
(データ・ステートメント 2)
(データ・ステートメント 3)

./ NUMBER SEQ1=AA030,INSERT=YES

(データ・ステートメント 4)
(データ・ステートメント 5)
(データ・ステートメント 6)
(データ・ステートメント 7, シーケンス番号 AA035)

/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 および SYSUT2 DD では、区分データ・セット (PDS) を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- CHANGE 機能ステートメントでは、入力メンバー RENUM を識別しています。このメンバーの全体はメッセージ・データ・セットに出力されます。
- NUMBER 詳細ステートメントでは、挿入操作を指定し (INSERT=YES)、番号変更操作を以下のように制御しています。
- データ・ステートメント 1、2、3 とデータ・ステートメント 4、5、6 は、それぞれが挿入する論理レコードのブロックです。それらのステートメントではシーケンス番号が空白であるため、それらのステートメントの挿入時にシーケンス番号が割り当てられます。

- データ・ステートメント 7 にはシーケンス番号が入っているため、挿入操作はそこで停止します。このシーケンス番号は、古いマスター・データ・セット内の次のレコードの番号と同じになっています。そのため、データ・ステートメント 7 は同じ番号を持つ古いマスター・レコードを出力データ・セット内で置き換えます。

この例では、既存の論理レコードのそれぞれのシーケンス番号が AA010、AA015、AA020、AA025、AA030、AA035、AA040、AA045、AA050、BB010、BB015 となっています。挿入と番号変更の操作は以下のように実行されます。

- データ・ステートメント 1 にはシーケンス番号 AA020 が割り当てられ (NEW1=AA020)、既存の論理レコード AA015 の後ろに挿入されます (SEQ1=AA015)。
- データ・ステートメント 2 および 3 にはシーケンス番号 AA025 および AA030 が割り当てられ (INCR=5)、データ・ステートメント 1 の後ろに挿入されます。
- 既存の論理レコード AA020、AA025、AA030 にはそれぞれシーケンス番号 AA035、AA040、AA045 が割り当てられます。
- データ・ステートメント 4 はシーケンス番号 AA050 が割り当てられてから挿入されます。(2 番目の NUMBER ステートメントに指定した SEQ1=AA030 によって、このデータ・ステートメントは既存の論理レコード AA030 (この時点では論理レコード AA045 になっている) の後ろに入れられます。)
- データ・ステートメント 5 および 6 にはそれぞれシーケンス番号 AA055 および AA060 が割り当てられ、データ・ステートメント 4 の後ろに挿入されます。
- 既存の論理レコード AA035 はデータ・ステートメント 7 で置き換えられ、シーケンス番号 AA065 がそれに割り当てられます。
- メンバー内の残りの論理レコードは、論理レコード BB010 が検出されるまで番号が変更されます。このレコードのシーケンス番号は次に割り当てられるはずの番号よりも大きいいため、番号変更操作は終了します。

表 38 には、既存のシーケンス番号、挿入されるデータ・ステートメント、そして新しいシーケンス番号が示されています。

表 38. シーケンスが変更されたシーケンス番号

シーケンス番号と挿入される	
データ・ステートメント	新しいシーケンス番号
AA010	AA010
AA015	AA015
データ・ステートメント 1	AA020
データ・ステートメント 2	AA025
データ・ステートメント 3	AA030
AA020	AA035
AA025	AA040
AA030	AA045
データ・ステートメント 4	AA050
データ・ステートメント 5	AA055

表 38. シーケンスが変更されたシーケンス番号 (続き)

シーケンス番号と挿入される データ・ステートメント	新しいシーケンス番号
データ・ステートメント 6	AA060
AA035 (データ・ステートメント 7)	AA065
AA040	AA070
AA045	AA070
AA050	AA075
BB010	BB010
BB015	BB015

例 9: カード入力から順次データ・セットを作成する

この例では、IEBUPDTE を使ってカード入力から順次データ・セットを作成します。同じ入カストリームからのユーザー・ヘッダー・ラベルとユーザー・トレーラー・ラベルもこの順次データ・セットに入れられます。

```
//LABEL JOB ...
//CREATION EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=LABEL,VOLUME=SER=123456,UNIT=disk,
// DISP=(NEW,KEEP),LABEL=(,SUL),SPACE=(TRK,(15,3))
//SYSIN DD *
./ LABEL
```

(ヘッダー・ラベル)

```
./ ADD LIST=ALL,OUTHDR=ROUTINE1,OUTTLR=ROUTINE2
```

(データ・レコード)

```
./ LABEL
```

(トレーラー・ラベル)

```
./ ENDUP
```

```
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT2 DD では、ディスク・ボリューム上にある LABEL という名前の出力順次データ・セットを定義し、スペースを割り振っています。
- SYSIN DD では、入カストリーム内の後続の制御データ・セットを定義します。(この制御データ・セットには、カード上にある順次入力データ・セットとユーザー・ラベルが含まれています。)
- 最初の LABEL ステートメントでは、ユーザー・ヘッダー・ラベルとなる入カストリーム内の、80 バイトからなるカード・イメージを識別しています。(これらは ADD 機能ステートメントに指定したユーザーのヘッダー・ラベル処理ルーチンで変更できます。)
- ADD 機能ステートメントでは、後続のデータ・ステートメントを出力データ・セットに入れることを示しています。新しく作成したデータ・セットはメッセージ・データ・セットに出力されません。ユーザー出力ヘッダー・ルーチンおよびユ

ユーザー出力・トレーラー・ルーチンには、ヘッダー・ラベルおよびトレーラー・ラベルを書き込む前に制御が渡されます。

- 2 番目の LABEL ステートメントでは、ユーザー・トレーラー・ラベルとなる入力ストリーム内の、80 バイトからなるカード・イメージを識別しています。(これらは ADD 機能ステートメントに指定したユーザーのトレーラー・ラベル処理ルーチンで変更できます。)
- ENDUP は、制御データ・セットがそこで終わっていることを示しています。

例 10: 順次データ・セットをボリューム間でコピーする

この例では、IEBUPDTE を使って 1 つの DASD ボリュームから別の DASD ボリュームに順次データ・セットをコピーします。ユーザー・ラベルはユーザー出力ルーチンで処理されます。

```
//LABELS JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=(MOD,,INTLRN)
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=OLDMAST,DISP=OLD,LABEL=(,SUL),
//          VOLUME=SER=111111,UNIT=disk
//SYSUT2 DD DSN=NEWMAS,DISP=(NEW,KEEP),LABEL=(,SUL),
//          UNIT=disk,VOLUME=SER=XB182,
//          SPACE=(TRK,(5,10))
//SYSIN DD DSN=INPUT,DISP=OLD,LABEL=(,SUL),
//          VOLUME=SER=222222,UNIT=disk
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、ディスク・ボリューム上に置かれている入力順次データ・セット OLDMAST を定義しています。
- SYSUT2 DD では、ディスク・ボリューム上に置かれる出力順次データ・セット NEWMAS を定義しています。
- SYSIN DD では、制御データ・セットを定義しています。この例でディスク上に置かれているこのデータ・セットの内容は次のとおりです。

```
./ REPRO LIST=ALL,INHDR=INHRTN,INTLR=INTRN, C
./ OUTHDR=OUTHRTN,OUTTLR=OUTTRN
./ ENDUP
```

- REPRO 機能ステートメントでは、既存の入力順次データ・セットを出力データ・セットにコピーすることを示しています。この出力データ・セットはメッセージ・データ・セットに出力されます。入力データ・セットまたは出力データ・セットのいずれかでヘッダー・ラベルかトレーラー・ラベルが検出されると、ユーザーのラベル処理ルーチンに制御が渡されます。
- ENDUP は、制御データ・セットがそこで終わっていることを示しています。

例 11: 新しい区分データ・セットを作成する

この例では、3 つのメンバーで構成されている区分世代別データ・セットが、新しい区分データ・セットを作成するときのソースとして使われます。IEBUPDTE はまた、その 3 つのソース・メンバーに 4 番目のメンバーを追加し、この新しいメンバーに番号を割り当てるのにも使われています。その結果として作成されるデータ・セットは、新しい区分データ・セットとしてカタログされます。

```
//NEWGDS JOB ...
//STEP1 EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=A
```

```
//SYSUT1 DD DSNAME=A.B.C,DISP=OLD
//SYSUT2 DD DSNAME=A.B.C,DISP=(,CATLG),UNIT=disk,
//          VOLUME=SER=111111,SPACE=(TRK,(100,10,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSIN   DD DATA
./  REPRO  NAME=MEM1,LEVEL=00,SOURCE=0,LIST=ALL
./  REPRO  NAME=MEM2,LEVEL=00,SOURCE=0,LIST=ALL
./  REPRO  NAME=MEM3,LEVEL=00,SOURCE=0,LIST=ALL
./  ADD    NAME=MEM4,LEVEL=00,SOURCE=0,LIST=ALL
./  NUMBER NEW1=10,INCR=5
```

(MEM4 を含むデータ・レコード)

```
./  ENDUP
/*
```

各制御ステートメントについての説明は以下のとおりです。

- **SYSUT1 DD** では、ソース・データとして使われる区分データ・セットを定義しています。これは、PDSE の場合もあります。
- **SYSUT2 DD** では、ソースの区分データ・セットから作成した、および入力データやデータとして含まれている追加のメンバーから作成した、新しい区分データ・セットを定義しています。これは、PDSE の場合もあります。
- **SYSIN DD** では、入力ストリーム内の後続の制御データ・セットを定義します。
- **REPRO** 機能ステートメントでは、名前付きソース・メンバーを出力区分データ・セット内に複製しています。
- **ADD** 機能ステートメントでは、MEM4 として含まれる入力ストリームの後ろにあるデータ・レコードを指定しています。
- **NUMBER** 詳細ステートメントでは、新しいメンバーに割り当てられるシーケンス番号が列 73~80 にあることを示しています。最初のレコードにはシーケンス番号 10 が割り当てられます。それ以後の各レコードのシーケンス番号は 5 ずつ大きくなります。
- **ENDUP** は、入力カード・データがそこで終わっていることを示しています。

第 11 章 IEHINITT (テープの初期化) プログラム

IEHINITT は、1 つまたは複数の磁気テープ装置に装てんされた任意の数のテープに、標準ボリューム・ラベル・セットを配置するのに使うシステム・ユーティリティです。ASCII (情報交換用米国標準コード) で作成された ISO/ANSI バージョン 3 または ISO/ANSI バージョン 4 のボリューム・ラベル・セット、あるいは EBCDIC で作成された IBM 標準ラベルのいずれでも使用できます。

本書で「バージョン 3」という用語を使用する場合は、ANSI X3.27-1978、ISO 1001-1979 および FIPS 79 の各標準を指しています。

「バージョン 4」という用語を使用する場合は、ANSI X3.27-1987 レベル 4 および ISO 1001-1986(E) レベル 4 の両標準を指しています。

米国政府はかつて FIPS (連邦情報処理標準) 79 (1980 年 10 月 17 日に規格化) に従っており、連邦標準として ISO/ANSI バージョン 3 を採用していました。米国政府は後日 FIPS 79 を撤廃し、それに代わる標準を採用しませんでした。

IEHINITT は APF 許可のプログラムです。つまり、他のプログラムがそれを呼び出すとき、そのプログラムも APF 許可でなければなりません。システムの健全性を保護するために、呼び出し側プログラムは *z/OS MVS Programming: Authorized Assembler Services Guide* に記載されているシステム健全性の要件に従ってなければなりません。

IEHINITT は、有効期限やセキュリティ保護を考慮せずに、前もってラベル付けされたテープに上書きできるため、IBM では、セキュリティ管理者が次のコマンド・シーケンスにより、プログラム保護を行うことをお勧めします。

- RDEFINE PROGRAM IEHINITT ADDMEM('SYS1.LINKLIB'//NOPADCHK) UACC(NONE)
- PERMIT IEHINITT CLASS(PROGRAM) ID(アクセス権の必要なユーザーまたはグループ) ACCESS(READ)
- SETROPTS WHEN(PROGRAM) REFRESH [これまでこのオプションをアクティブにしていなかった場合、REFRESH は省略します]

不正な上書き操作からテープをより確実に保護するため、非ライブラリー環境においては、IEHINITT は毎回、テープの装てんを検査するようオペレーターに要求します。システム管理テープ環境における IEHINITT の使用は、セキュリティ管理が運用データの消滅を防止することをインプリメントされていることを前提としています。SAF/RACF は、ライブラリー環境において、許可された処理として呼び出されます。初期化に先行して必要とされる許可のレベルは、CLASS=TAPEVOL の場合 UPDATE です。

さらに、動的出口サービスの CSVDPYNEX 経由で管理されたインストール・システム出口を使用できます。インストール・システム出口の使用は完全に任意選択ですが、これがインプリメントされていれば、インストールによってすべての初期化要

求を見直し、任意のボリュームが初期化されるべきかどうかを明確にすることができます。詳細は、*z/OS DFSMS Installation Exits* を参照してください。

IEHINITT には、SAF/RACF を使用したボリューム・レベルのセキュリティー検査が備えられていますが、データ・セット・レベルの検査は、呼び出されません。SAF/RACF TAPEVOL プロファイルを使用しない場合は、IEHINITT をシステム管理のライブラリーで使用する際に、特別な予防措置が必要になります。

注: IEHINITT の代替手段として、268 ページの『DFSMSrmm の使用』に記載されている EDGINERS ユーティリティーの使用を検討してください。EDGINERS はセキュリティーとボリューム所有権を検査し、監査を行います。IEHINITT は行いません。

このプログラムが作成するボリューム・ラベル・セットにはそれぞれ以下のものが含まれています。

- ユーザー指定の通し番号、所有者識別、そしてブランクのセキュリティー・バイトを持つ、標準ボリューム・ラベル。

ISO/ANSI バージョン 3 および バージョン 4 ラベルには、ACCESS キーワードを使用することによって、ASCII スペース以外のアクセス・コードが含まれます。ISO/ANSI バージョン 3 のラベルが要求されている場合は、バージョン 3 に準拠したラベルが作成されます。また、バージョン 4 のラベルが要求されている場合は、バージョン 4 に準拠したラベルが構成されます。

IBM 標準ボリューム・ラベル、およびバージョン 3 とバージョン 4 のボリューム・ラベルについては、*z/OS DFSMS Using Magnetic Tapes* で詳しく説明しています。

- 80 バイトからなるダミーのヘッダー・ラベル。IBM 標準ラベルの場合、このレコードは文字『HDR1』とその後に続く複数個のゼロで構成されます。バージョン 3 またはバージョン 4 のラベルの場合、このレコードは文字『HDR1』、とその後ろの残りの部分を占める複数個のゼロとで構成されます。ただし、以下のような例外があります。
 - 位置 54。ここには ASCII スペースが入ります。
 - ファイル・セクション番号、ファイル・シーケンス番号、および世代番号の各フィールドにある『1』。
 - 作成日フィールドと有効期限フィールドの先行スペース。
 - システム・コードである『IBMZLA』。ラベルを作成しているオペレーティング・システムを識別します。その後ろには 14 個のスペースがあります。
- テープ・マーク。

ラベル付きテープがその後で受け取り側のボリュームとして使われると、OPEN 機能または、EOV 機能は、以下ようになります。

1. ボリューム・ラベルを書き換えます。
2. IEHINITT が作成したダミーの HDR1 レコードを、オペレーティング・システムのデータ、装置依存の情報、およびデータ・セット情報で書き換えます。
3. データ・セット特性を収めた HDR2 レコードを書き込みます。

4. ユーザー・プログラムでユーザー・ラベル・ルーチンへの出口ルーチンを使用する場合は、ユーザー・ヘッダー・ラベルを書き込みます。
5. テープ・マークを書き込みます。
6. ユーザー・プログラム用のボリュームを配置して、データを書き込みます。

バージョン 3 およびバージョン 4 のテープ・ラベルについての注意点は以下のとおりです。

バージョン 3 の場合、ACCESS キーワードにある大文字の A~Z を検査する以外は、IEHINITT の処理中にアクセス可能度コードの検査は行われません。そのため、作成したテープのボリューム・アクセス・コードが、受け取り側のオペレーティング・システムで認識されないものになる可能性があります。このような状況では、受け入れ可能なアクセス・コードを含めるためにテープを再初期化する必要が生じることがあります。

バージョン 3 における有効な文字は以下のとおりです。

```
大文字 A ~ Z、数字 0 ~ 9、および特殊文字
| " % & ' ( ) * + , - . / : ; < = > ? スペース
```

バージョン 4 における有効な文字は以下のとおりです。

```
大文字 A ~ Z、数字 0 ~ 9、および特殊文字
| " % & ' ( ) * + , - . / : ; < = > ? _ スペース
```

上記の 2 つの特殊文字のリストで違っているのは _ (下線) だけです。

バージョン 3 またはバージョン 4 のボリュームを IEHINITT だけで初期化した場合、生成されるラベルでは交換に必要な空の (ヌル) データは構成されません。ISO/ANSI 標準で規定されているラベルの対称性を実現するには、少なくとも最小限の回数のオープン/クローズ操作を行う必要があります。たとえば、以前に IEHINITT で初期化したボリュームを交換のためにシステムから外す前にデータ・セット・ユーティリティ IEBGENER を使用すると、そのボリュームでラベルの対称性が実現します。以下の例を参照してください。

```
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY
//SYSUT1 DD DUMMY,DCB=(RECFM=F,BLKSIZE=80,LRECL=80)
//SYSUT2 DD DSN=DUMMY,UNIT=(tape),LABEL=(,AL),DISP=OLD,
// DCB=(RECFM=F,BLKSIZE=80,LRECL=80),VOL=SER=volser
//SYSIN DD DUMMY
```

ボリューム・ラベルについては、*z/OS DFSMS Using Magnetic Tapes* の説明を参照してください。

標準ラベル・セットを磁気テープ上に配置する

IEHINITT を利用すれば、BCDIC ラベルを 7 トラックのテープ・ボリュームに、EBCDIC または ASCII (ISO/ANSI フォーマット) ラベルを 9、18、36、128、256 トラックのテープ・ボリュームに書き込むことができます。IEHINITT の 1 回の実行で、任意の数のテープ・カートリッジまたはテープ・ボリュームにラベルを付けることが可能です。

IEHINITT は IBM 3494 および 3495 自動化テープ・ライブラリー・データ・サーバーのすべての環境、さらに 3495-M10 環境でサポートされています。IEHINITT

を 3494 または 3495 自動化テープ・ライブラリー・データ・サーバーで使用する際には特別な考慮事項が適用されます。詳細については、271 ページの『磁気テープ・ライブラリー・データ・サーバーの考慮事項』を参照してください。

単一の INITT コマンドによって初期化された複数のテープ・ボリュームに、通し番号順にラベルを付けるには、最初のテープ・ボリュームに書き込む通し番号を指定します。この通し番号は 6 桁の数字で指定する必要があり、その数値はテープ・ボリュームが増えるごとに 1 つずつ大きくなっていきます。ラベルを付けるテープ・ボリュームが 1 つしかない場合は、指定する通し番号は数字と英数字のいずれでも構いません。

以下のような補足的な情報を追加することも可能です。

- 所有者名
- 巻き戻しまたはアンロードの指定
- フォーマット
- アクセス・コード

ラベルを付けるテープをすべて用意し、それぞれのテープをどこに装てんするかに関するオペレーターへの明確な指示を各ジョブ要求に含めておく必要があります。テープ・ライブラリー・データ・サーバーの中にあるテープについては、271 ページの『磁気テープ・ライブラリー・データ・サーバーの考慮事項』を参照してください。

IEHINITT は、7 トラック・テープのラベルに偶数パリティ (変換プログラムをオン、コンバーターをオフ) で書き込みます。

あらかじめラベルの付けられたテープは、有効期限およびセキュリティー保護を考慮することなく、新規ラベルで上書きすることができます。SAF/RACF が起動され、テープ・ライブラリー・データ・サーバー内のドライブについて、テープ・ボリュームへの適正なアクセス・レベルを決定します。

テープにラベルを付ける段階でエラーが検出されると、そのテープにはラベルが付けられません。IEHINITT は処理すべきテープがその他に残っていれば、それらのテープにラベルを付けようとします。

標準ラベルや標準外ラベルを書き込むルーチンの作成については、*z/OS DFSMS Using Magnetic Tapes* を参照してください。

DFSMSrmm の使用

IBM TotalStorage Enterprise 自動化テープ・ライブラリー (3495) の内部と外部の両方にあるラベル付きテープの場合、IEHINITT に代わって DFSMSrmm の EDGINERS のユーティリティーを推奨します。その理由は以下のとおりです。

1. DFSMSrmm を用いてボリューム・セットにラベルを付けることができます。
2. DFSMSrmm は、ボリューム上のデータ・セットの有効期限が切れたことを確認します。
3. DFSMSrmm は、ボリューム・ラベルを作成するまえに、正しいボリュームがマウントされたかを確認します。

4. DFSMSrmm は、ボリュームにラベルを付ける必要があることを追跡でき、その制御データ・セットにある情報を使用して、テープにラベルを付けるのを自動化することができます。
5. DFSMSrmm は、テープの有効期限が切れたときに、そのテープ上のデータを消去するための機能も提供しています。
6. EDGINERS ユーティリティーについて、詳しくは *z/OS DFSMSrmm インプリメンテーションとカスタマイズのガイド* を参照してください。

入出力

IEHINITT は入力データとして、ユーティリティー制御ステートメントが入っている制御データ・セットを使用します。IEHINITT が生成する出力データ・セットには以下のものが入っています。

- ユーティリティー・プログラムの識別
- 正常にラベルを付けられたテープ・ボリュームごとの、初期ボリューム・ラベル情報
- ユーティリティー制御ステートメントの内容
- すべてのエラー・メッセージ

IEHINITT は、キャプチャー解除された UCB アドレスの使用をサポートしています。キャプチャーとキャプチャー解除は、16 MB 境界を超える 31 ビット・アドレスをそのラインより下の 24 ビット・アドレスに変換するためにオペレーティング・システムが使用するプロセスです。装置が JCL を介して割り振られている場合（これは IEHINITT を呼び出す通常の方法です）は、ユーザーが注意すべき特別な考慮事項はありません。ただし、ユーザー作成のアプリケーションは、装置をキャプチャーせずに動的に割り振り、IEHINITT に渡すことができます。これは、357 ページの『付録 A. アプリケーション・プログラムからユーティリティー・プログラムを起動する』で説明しています。

IEHINITT は戻りコードをレジスター 15 で設定します。IEHINITT の戻りコードについては、357 ページの『付録 A. アプリケーション・プログラムからユーティリティー・プログラムを起動する』を参照してください。

制御

IEHINITT はジョブ制御ステートメントやユーティリティー制御ステートメントで制御します。ジョブ制御ステートメントは IEHINITT の処理やロード、および IEHINITT が使用したり生成したりするデータ・セットの定義などに使われます。ユーティリティー制御ステートメントは、適用可能なラベル情報を指定するのに使われます。

ジョブ制御ステートメント

表 39 は、IEHINITT で使用するジョブ制御ステートメントをまとめたものです。

表 39. IEHINITT で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。

表 39. IEHINITT で使用するジョブ制御ステートメント (続き)

ステートメント	用途
EXEC	プログラム名を指定します (PGM=IEHINITT)。このジョブ制御ステートメントがプロシージャー・ライブラリー内にある場合は、プロシージャー名を指定します。EXEC ステートメントには補足的なパラメーター情報を組み込むこともできます。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。
anyname (任意名) DD	ラベル操作で使われる磁気テープ装置を定義します。複数の磁気テープ装置を識別することができます。
SYSIN DD	制御データ・セットを定義します。制御データ・セットは通常、入力ストリームにあります。区分データ・セットや PDSE のメンバーとして定義したり、入力ストリームの外部にある順次データ・セットとして定義したりすることもできます。

EXEC ステートメント

EXEC ステートメントには、メッセージ・データ・セットの見出しと見出しの間に印刷すべき行数を指定する PARM 情報を組み込むことができます。EXEC ステートメントは以下のようにコーディングします。

//[stepname]	EXEC	PGM=IEHINITT [,PARM='LINECNT=nn']
--------------	------	--------------------------------------

各項目の意味は以下のとおりです。

PGM=IEHINITT

実行するプログラムが IEHINITT であることを指定します。

PARM='LINECNT=nn'

SYSPRINT データ・セットに出力する 1 ページ当たりの行数を指定します。PARM が省略されている場合は、見出しと見出しの間に 60 行が出力されません。

IEHINITT が別のプログラムから呼び出された場合は、LINK または ATTACH マクロ命令の *optionaddr* サブパラメーターが参照しているパラメーター・リストに、行カウントのオプションを渡すことができます。さらに、LINK または ATTACH マクロ命令の *hdingaddr* サブパラメーターが参照している 6 バイトのパラメーター・リストに、開始ページ番号を渡すこともできます。リンケージの規則については、357 ページの『付録 A. アプリケーション・プログラムからユーティリティー・プログラムを起動する』の説明を参照してください。

SYSPRINT DD ステートメント

SYSPRINT データ・セットの論理レコード長は 121 でなければなりません。このデータ・セットは、各レコードの先頭バイトが ISO/ANSI 制御文字になっている固定長のレコードで構成されている必要があります。任意のブロック化因数を指定できます。

anyname (任意名) DD ステートメント

「anyname」DD ステートメントは以下のように入力します。

```
//anyname DD DCB=DEN=x,UNIT=(xxxx,n,DEFER)
```

DEN パラメーターでは、ラベルを書き込む密度を指定しています。UNIT パラメーターでは、装置タイプ、ラベル処理で使用する装置の数、および据え置き装てんを指定しています。DEFER は、IEHINITT が処理を行える状態になるまでテープの装てんを遅らせるときに必ず指定します。DEFER を使用しない場合は、JCL で指定されたボリュームは割り振りによって装てんされます。IEHINITT が制御を受け取ると、この装てんされたボリュームを検出して、無条件に取り外します。これを指定することで、装てんされているテープが正しいかどうかを確認する時間をオペレーターに与えることができます。

DEN および UNIT パラメーターの詳細については、*z/OS MVS JCL 解説書* を参照してください。

名前「anyname」をユーティリティー制御ステートメントに指定した名前と同一にすることにより、ここで指定した装置をそのユーティリティー制御ステートメントに関連付ける必要があります。また、DISP=(,PASS) をコーディングすれば、テープをドライブに装てんさせたままにしておき、後続のジョブ・ステップで使用することもできます。278 ページの『例 8: 取り外しや再装てんを行わないラベル付けに従って、磁気テープに書き込む』を参照してください。

磁気テープ・ライブラリー・データ・サーバーの考慮事項

IEHINITT をライブラリー・データ・サーバー環境で実行する場合は、以下のような JCL を使用することになります。

```
//anyname DD UNIT=(,DEFER),
//          VOLUME=SER=volser,
//          DISP=(OLD,KEEP)
```

「anyname」は、INITT ユーティリティー制御ステートメントにある名前です。

「volser」は、以下のすべての要件を満たしているボリュームのボリューム通し番号です。

- 初期化するボリュームが入っている磁気テープ・ライブラリーの中に置かれているボリュームである。
- PRIVATE 使用属性を有するボリュームである。
- テープにラベルを付けるときに使用される、18、36、128、または 256 トラック用のテープ・レコード・テクノロジーで記録されるボリュームである。
- 読み取り互換特別属性を有していないボリュームである。
- ラベルを付けることになっているボリュームの 1 つではない。

注: 「volser」がライブラリーに置かれているボリュームであれば、装置名を指定する必要はありません。

IEHINITT を自動化テープ・ライブラリー・データ・サーバーで実行するときの、さらに別の考慮事項は以下のとおりです。

- IEHINITT は、ライブラリー・カテゴリーを認識しないため、テープが SCRATCH と PRIVATE のどちらのカテゴリーに属しているかに関係なく、磁気テープ・ライブラリー・データ・サーバー内の磁気テープをマウントして初期化します。
- メッセージ IEC701D、テープ / オペレーターのコンソールに出される通常の WTOR 許可メッセージは、自動化テープ・データ・サーバーのドライブ上で処理を行っているときにはコンソールに表示されません。

SYSIN DD ステートメント

SYSIN データ・セットの論理レコード長は 80 でなければなりません。ブロック化因数に指定できる最大ブロック・サイズは 32720 です。

ユーティリティー制御ステートメント

IEHINITT では、ラベル処理操作のための制御情報を提供するのにユーティリティー制御ステートメント INITT が使われます。ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

このプログラムの 1 回の実行に、INITT ユーティリティー制御ステートメントをいくつでも組み込むことができます。各 INITT ステートメントには、入力ストリーム内の DD ステートメントを識別する DD 名を使ってラベルを付ける必要があります。

図 38 には、INITT ステートメントと初期ボリューム・ラベル情報が含まれているメッセージ・データ・セットの出力結果が示されています。この例では、通し番号 001122 と 001123 を 2 つの標準ラベル・テープ・ボリュームに付けるのに使われている INITT ステートメントは 1 つです。VOL1001122 と VOL1001123 は以下のように解釈します。

- VOL1 は、初期ボリューム・ラベルが正常にテープ・ボリュームに書き込まれたことを示しています。
- 001122 および 001123 は、それぞれのボリュームに書き込まれた通し番号です。
- 通し番号の後ろにあるブランクはボリューム・セキュリティー・フィールドを表しており、標準ラベル・テープで OPEN/CLOSE/EOV 処理を行っている間は使われません。

処理中に発生したエラーはありません。

```

SYSTEM SUPPORT UTILITIES  IEHINITT

ALL  INITT  SER=001122,NUMBTAPE=2,OWNER='P.T.BROWN',DISP=REWIND

VOL1001122                P.T.BROWN
VOL1001123                P.T.BROWN

```

図 38. INITT ステートメントの指定内容と初期ボリューム・ラベル情報の出力結果

INITT ステートメントの構文は次のとおりです。

<i>ddname</i>	INITT	SER=serial number
---------------	--------------	--------------------------

		[,DISP={REWIND UNLOAD}] [,OWNER='name'] [,NUMBTAPE={n1}] [,LABTYPE=AL] [,VERSION={3 4 }] [,ACCESS=c]
--	--	---------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

ddname

磁気テープ装置 (1 つまたは複数) を定義している DD ステートメントの名前フィールドにある DD 名 と同一の名前を指定します。この名前は、INITT ステートメントが入っているレコードの 1 桁目で始まっていなければなりません。

SER=serial number

ラベルを付ける、最初のテープまたは 1 本だけのテープのボリューム通し番号を指定します。最大 6 文字まで指定します。IBM 標準ラベル (SL) が付いているテープの場合、その通し番号には空白、コンマ、アポストロフィ、等号を含めることができません。特殊文字については、ピリオド、ハイフン、ドル記号、ポンド記号、アットマーク (@) 以外のは含めることができません。ISO/ANSI ラベル (AL) が付いているテープであれば、OWNER キーワードの項で説明している有効な ISO/ANSI 'a' タイプの文字ならどれでも含めることができます。ただし、非英数字文字 (ピリオドとハイフンを含む) がある場合は、区切り用のアポストロフィを含める必要があります。

空白はボリューム通し番号の先頭文字として使うことができません。

指定した通し番号は、ラベルを付けるテープが増えるごとに 1 ずつ大きくなります。(通し番号 999999 の次は 000000 になります。) 複数のテープを処理している場合は、ボリューム通し番号がすべて数字でなければなりません。

DISP={REWIND|UNLOAD}

テープを巻き戻す場合、あるいは巻き戻してからアンロードする場合に指定します。磁気テープ・ライブラリー・データ・サーバーの中にあるテープはこのパラメーターの指定内容に関係なく、無条件にアンロードされます。コーディングできる値は以下のとおりです。

REWIND

ラベルの書き込み後にテープを巻き戻す (ただしアンロードしない) ことを指定します。

UNLOAD

ラベルの書き込み後にテープを巻き戻してからアンロードすることを指定します。これがデフォルトです。

OWNER='name'

所有者の名前、またはそれと同様の識別を指定します。この情報は文字定数として指定し、EBCDIC および BCD ボリューム・ラベルの場合には最大 10 バイト、ASCII によるボリューム・ラベルの場合には最大 14 バイトまで指定できます。空白、コンマ、アポストロフィ、等号、または特殊文字 (ピリオドとハイフンを除く) が含まれているときは、区切り用のアポストロフィがなければなりません。ASCII テープで使われる有効な ISO/ANSI 'a' タイプの文字は、

英大文字 A~Z、数字 0~9、および特殊文字 !*"%'()*+,-./:;<=>? です。有効な EBCDIC 文字は、英大文字 A~Z、数字 0~9、および特殊文字 < > (+ !(X'6A') l(X'4F') ' & ! \$ *) ; ^ - \ / , % _ ? ' : # @ ' = " ~ { } \ です。

OWNER の name フィールドにアポストロフィを含める場合は、連続した 2 個のアポストロフィでそれを記述しなければなりません。バージョン 3 またはバージョン 4 のテープには OWNER キーワードを指定できます。バージョン 4 を指定している場合は、任意の ISO/ANSI 'a' タイプの文字を使用できます。

NUMBTAPE={n1}

この制御ステートメントでの指定内容に従ってラベルを付けるテープの数を指定します。値 *n* は 1~255 の数を表します。複数のテープを指定している場合は、最初のテープのボリューム通し番号が数字でなければなりません。

LABTYPE=AL

LABTYPE=AL を INITT ステートメントに指定すると、IEHINITT は VERSION キーワードの指定に従ってバージョン 3 またはバージョン 4 の標準に準拠するようテープを初期化します。VERSION キーワードのフォーマットは以下のとおりです。

デフォルト: このテープは、テープ・カートリッジや 9 トラックのボリュームの場合は IBM 標準フォーマットの EBCDIC で、また 7 トラックのテープ・ボリュームの場合は BCDIC で書き込まれます。

VERSION={3|4}

LABTYPE=AL を指定したときに、フォーマットをバージョン 3 とバージョン 4 のいずれにするかをこの VERSION キーワードで決定します。

- 3 テープをバージョン 3 に初期化します。
- 4 テープをバージョン 4 に初期化します。

ボリューム・ラベル (VOL 1) はバージョン 3 でもバージョン 4 でも同じように書き込まれますが、例外として、フィールド『Label Standard Version』は、バージョン 3 の場合には 3 に、またバージョン 4 の場合には 4 になります。ヘッダー・ラベルの初期化に違いはありません。

デフォルト: このキーワードを指定するとその値が使われます。指定されていない場合は、DEVSUPxx PARMLIB メンバーにあるインストール・バージョン・レベルが使われます。これも指定されていないとバージョン 3 がデフォルトになります。

ACCESS=c

バージョン 3 およびバージョン 4 ボリュームのアクセス可能度コードを指定します。バージョン 3 のときの *c* の有効な値は、英大文字 A~Z のみです。デフォルト値は空白であり、これはボリュームに対するアクセスが制限されないことを示します。バージョン 3 の場合は、アクセス・コードに空白を指定することはできません。これはデフォルトのときだけ可能です。バージョン 4 で有効な値は、次の任意の ISO/ANSI 'a' タイプの文字です。つまり、英大文字 A~Z、数字 0~9、および特殊文字 !*"%'()*+,-./:;<=>? です。 .

ACCESS を指定した場合は、ボリューム・アクセスのインストール・システム 出口ルーチンを変更して、ボリュームをそれ以後も使用できるようにしなければ

なりません。ボリュームのアクセス可能度と ISO/ANSI インストール・システム出口の詳細については、*z/OS DFSMS Using Magnetic Tapes* を参照してください。

ACCESS を指定する場合は、LABTYPE=AL と VERSION も指定しなければなりません。

ACCESS キーワードは、バージョン 3 とバージョン 4 のどちらの場合でも指定できます。バージョン 4 を指定している場合は、キーワードに任意の ISO/ANSI 'a' タイプの文字を指定できます。

IEHINITT の例

以下の各例では、IEHINITT の使用方法の一部を示します。表 40 は IEHINITT の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 40. IEHINITT の例の一覧表

操作	注釈	例
LABEL	3 つの 9 トラック・テープにラベルを付けます。	1
LABEL	1 つの 9 トラック・テープに ISO/ANSI でラベルを付けます。	2
LABEL	2 つのグループの 9 トラック・テープ・ボリュームにラベルを付けます。	3
LABEL	9 トラック・テープ・ボリュームにラベルを付けます。シーケンス番号は、10 ずつに区切られます。	4
LABEL	3 つの 9 トラック・テープ・ボリュームにラベルを付けます。1 つのテープ・ボリュームには英数字のラベルを付け、残りの 2 つのテープ・ボリュームには数字のラベルを付けます。	5
LABEL	2 つの 9 トラック・テープ・ボリュームにラベルを付けます。1 つ目のボリュームには 6250 ビット / インチの密度で、2 つ目には 1600 バイト / インチの密度でラベルが付けられます。	6
LABEL	1 つの 9 トラック・テープ・ボリュームは、非ブランク・アクセス・コードを用いて、ISO/ANSI フォーマットでラベルが付けられます。	7

各例で、実際の装置番号の代わりに **tape** となっている箇所は、使用する前に変更しておく必要があります。実際の装置番号は、インストール時にその装置がシステムにどのように定義されたかによって異なります。

例 1: EBCDIC ラベルを 3 つのテープに書き込む

この例では、通し番号 001234、001235、001236 を 3 つのテープ・ボリュームにそれぞれ付けます。これらのラベルは EBCDIC の 800 bpi (ビット / インチ) で書き込まれます。ラベルを付ける各ボリュームは、必要であれば単一の 9 トラック磁気テープ装置に装てんされます。

```
//LABEL1 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//LABEL DD DCB=DEN=2,UNIT=(tape,1,DEFER)
//SYSIN DD *
LABEL INITT SER=001234,NUMTAPE=3
/*
```


各制御ステートメントについての説明は以下のとおりです。

- LABEL DD では、ラベル処理操作で使われる磁気テープ装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- LABEL INITT では、ラベルを付けるテープの数 (3) を指定しています。数え始める最初のテープは 001234 です。

例 2: ISO/ANSI ラベルをテープに書き込む

この例では、通し番号 001001 が 1 つの ISO/ANSI テープ・ボリュームに付けられます。(そのラベルは 800 ビット / インチで書かれます。)ラベルを付けるボリュームは、必要であれば 1 つの 9 トラック磁気テープ装置に装てんされます。

```
//LABEL2 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//ASCIILAB DD DCB=DEN=2,UNIT=(tape,1,DEFER)
//SYSIN DD *
ASCIILAB INITT SER=001001,OWNER='SAM A. BROWN',LABTYPE=AL
/*
```

各制御ステートメントについての説明は以下のとおりです。

- ASCIILAB DD では、ラベル処理操作で使われるテープ・ボリュームを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- ASCIILAB INITT では、ボリュームの通し番号、所有者 ID、およびラベルのタイプを指定しています。VERSION キーワードを指定していないため、この ISO/ANSI テープは DEVSUPxx PARMLIB メンバーの指定内容にもとづいて、あるいはデフォルトではバージョン 3 として作成されます。

例 3: 通し番号の 2 つのグループを 6 つのテープ・ボリュームに付ける

この例では、通し番号の 2 つのグループ (001234、001235、001236 のグループと、001334、001335、001336 のグループ) を 6 つのテープ・ボリュームに付けます。これらのラベルは EBCDIC の 800 bpi (ビット / インチ) で書き込まれます。ラベルを付ける各ボリュームは、必要であれば単一の 9 トラック磁気テープ装置に装てんされます。

```
//LABEL3 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//LABEL DD DCB=DEN=2,UNIT=(tape,1,DEFER)
//SYSIN DD *
LABEL INITT SER=001234,NUMBTAPE=3
LABEL INITT SER=001334,NUMBTAPE=3
/*
```

各制御ステートメントについての説明は以下のとおりです。

- LABEL DD では、ラベル処理操作で使われる磁気テープ装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- LABEL INITT では、6 つのテープ・ボリュームに付ける通し番号の 2 つのグループを定義しています。

例 4: 通し番号を 8 つのテープ・ボリュームに付ける

この例では、001234、001244、001254、001264、001274、001284、001294、001304 などの通し番号を 8 つのテープ・ボリュームに付けます。これらのラベルは EBCDIC の 800 bpi (ビット / インチ) で書き込まれます。ラベルを付ける各ボリュームは、必要であれば 4 つの 9 トラック磁気テープ装置に装てんされます。

```
//LABEL4 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//LABEL DD DCB=DEN=2,UNIT=(tape,4,DEFER)
//SYSIN DD *
LABEL INITT SER=001234
LABEL INITT SER=001244
LABEL INITT SER=001254
LABEL INITT SER=001264
LABEL INITT SER=001274
LABEL INITT SER=001284
LABEL INITT SER=001294
LABEL INITT SER=001304
/*
```

各制御ステートメントについての説明は以下のとおりです。

- LABEL DD では、ラベル処理操作で使われる磁気テープ装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- LABEL INITT では、ボリューム通し番号でラベルを付けるテープを定義しています。

例 5: EBCDIC ラベルを異なる密度で書き込む

この例では、通し番号 TAPE1 を 1 つのテープ・ボリュームに、通し番号 001234 と 001235 を 2 つのテープ・ボリュームに付けます。これらのラベルは、それぞれ EBCDIC の 800 および 1600 ビット / インチで書き込まれます。

```
//LABEL5 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//LABEL1 DD DCB=DEN=2,UNIT=(tape,1,DEFER)
//LABEL2 DD DCB=DEN=3,UNIT=(tape,1,DEFER)
//SYSIN DD *
LABEL1 INITT SER=TAPE1
LABEL2 INITT SER=001234,NUMBTAPE=2
/*
```

各制御ステートメントについての説明は以下のとおりです。

- LABEL1 DD と LABEL2 DD では、ラベル処理操作で使われる 2 つのテープ・ボリュームを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- LABEL1 INITT では、LABEL1 DD で定義したテープ・ボリュームに通し番号 TAPE1 を付けています。 LABEL2 INITT では、LABEL2 DD で定義したテープ・ボリュームに通し番号 001234 および 001235 を付けています。

例 6: 通し番号を 2 つの密度でテープ・ボリュームに書き込む

この例では、通し番号 006250 は 6250 bpi (バイト / インチ) の密度で、テープ・ボリュームに EBCDIC を用いて書き込まれます。さらに、通し番号 001600 は 1600 bpi の密度で、2 つ目のボリュームに EBCDIC を用いて書き込まれます。

```
//LABEL6 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//DDFIRST DD DCB=DEN=4,UNIT=(tape,1,DEFER)
//DDSECOND DD DCB=DEN=3,UNIT=(tape,1,DEFER)
//SYSIN DD *
DDFIRST INITT SER=006250
DDSECOND INITT SER=001600
/*
```

各制御ステートメントについての説明は以下のとおりです。

- DDFIRST DD では、使用する最初のテープ・ボリュームを定義しています。
- DDSECOND DD では、使用する 2 番目のテープ・ボリュームを定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- DDFIRST INITT では、DDFIRST DD で定義したボリュームに通し番号 006250 を書き込んでいます。DDSECOND INITT では、DDSECOND DD で定義したボリュームに通し番号 001600 を書き込んでいます。

例 7: ISO/ANSI ラベルにアクセス・コードを指定して書き込む

この例では、バージョン 4 ISO/ANSI (AL) でラベルを付けたテープをブランク以外のアクセス・コードで作成します。ボリューム通し番号は TAPE01 です。

```
//LABEL7 JOB ...
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//LABEL DD UNIT=(tape,1,DEFER),DCB=DEN=4
//SYSIN DD *
LABEL INITT SER=TAPE01,OWNER=TAPOWNER,LABTYPE=AL,ACCESS=A,VERSION 4
/*
```

各制御ステートメントについての説明は以下のとおりです。

- LABEL DD では、テープを装てんする装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- INITT ステートメントでは、ボリューム通し番号が TAPE01 で所有者が TAPOWNER であるテープに付ける バージョン 4 ISO/ANSI ラベルを作成します。ACCESS のコードは『A』として指定されているため、このボリュームを受け入れるには、これを受け取るオペレーティング・システムがこの『A』を認識できなければなりません。

例 8: 取り外しや再装てんを行わないラベル付けに従って、磁気テープに書き込む

この例では、ジョブの 1 つのステップで磁気テープにラベル付けをし、次に、ステップの間にシステムがその磁気テープを取り外して、再装てんすることなく、同じジョブの次のステップで磁気テープに書き込むことができます。必要な JCL は次のとおりです。

```
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=*
(1) //LABEL DD UNIT=(tape,1,DEFER),DISP=(,PASS)
//SYSIN DD *
(2) LABEL INITT SER=serial,DISP=REWIND
//*
//STEP2 EXEC PGM=user_program
```

```
//INPUT DD DSN=input_dsn,DISP=SHR
//OUTPUT DD DSN=dsname,DISP=(NEW,CATLG),
//          DCB=(dcbinfo),
(3) //          UNIT=tape,VOL=(,RETAIN,SER=serial)
```

注:

- (1) DISP=(NEW,PASS) または VOL=(,RETAIN) を指定してください。
- (2) DISP=REWIND を INITT ステートメントで指定してください。
- (3) VOL=SER=serial を指定してください。
VOL=REF=*.STEP1.LABEL は機能しません。

第 12 章 IEHLIST (システム・データのリスト表示) プログラム

IEHLIST は、システム・ユーティリティーの 1 つであり、これを使用すれば、1 つまたは複数の区分データ・セットまたは PDSE のディレクトリー内の項目、または索引付きおよび索引なしボリューム目録内の項目をリスト表示することができます。このプログラムの 1 回の実行で、いくつでもリストを要求することができます。IEHLIST によって作成される VTOC リストの例、およびそのリストのフィールドに関する詳細説明は、387 ページの『付録 D. IEHLIST の VTOC リスト表示』を参照してください。

区分データ・セットまたは PDSE のディレクトリーのリスト表示

IEHLIST は、一度に 10 個までの区分データ・セットまたは PDSE のディレクトリーをリスト表示できます。

区分データ・セットのディレクトリーは、256 バイト・ブロックにブロック化された可変長レコードで構成されます。各ディレクトリー・ブロックには、メンバー名または別名、および区分メンバーのそのほかの属性を示す、1 つまたは複数の項目が含まれます。IEHLIST はこれらのブロックを、編集フォーマットおよび非編集フォーマットでリスト表示することができます。

PDSE のディレクトリーがリスト表示される際には、分割データ・セットのディレクトリーと同じフォーマットになります。

編集フォーマット

区分データ・セット・ディレクトリーの編集フォーマットは、モジュール・ライブラリーとともに使用するためのものです。編集されたリスト表示で得られる情報のほとんどは、ロード・モジュールにのみ利用価値があります。

メンバーがロード・モジュールではない区分データ・セットまたは PDSE の編集されたリスト表示を要求すると、編集されたリスト表示が得られます。その場合、リストの中には、データ・セットについての不正確な情報が含まれることがあります。このため、メンバーがロード・モジュールであるデータ・セットのみに関して、編集されたリスト表示を要求してください。

編集フォーマットのモジュール・ライブラリーを要求すると、IEHLIST は以下のような情報を提供します。

- メンバー名
- 入り口点
- メンバー開始の相対アドレス
- テキスト開始の相対アドレス
- 連続する仮想記憶域の所要量
- 最初のテキスト・ブロックの長さ
- 最初のテキスト・ブロックの起点
- システム状況の標識
- リンケージ・エディターの属性

IEHLIST

- 必要な APF 許可
- そのほかの情報

図 39 は、区分化されたメンバー LOADMOD の編集された項目を示しています。項目は、IEHLIST プログラムがリスト表示したままの状態です。ただし、この図は、区分データ・セットまたは PDSE ディレクトリーのリスト表示の一例に過ぎません。IEHLIST プログラムが実際に作成する編集されたリスト表示は、これとは異なる可能性があります。

```
OTHER INFORMATION INDEX
SCATTER FORMAT SCRT=SCATTER/TRANSLATION TABLE TTR IN HEX, LEN OF SCTR LIST IN DEC, LEN OF TRANS TABLE IN DEC,
                ESDID OF FIRST TEXT RCD IN DEC, ESDID OF CSECT CONTAINING ENTRY POINT IN DEC

OVERLAY FORMAT ONLY=NOTE LIST RCD TTR IN HEX, NUMBER OF ENTRIES IN NOTE LIST RCD IN DEC

ALIAS NAMES    ALIAS MEMBER NAMES WILL BE FOLLOWED BY AN ASTERISK IN THE PDS FORMAT LISTING

ATTRIBUTE INDEX

BIT  ON   OFF      BIT  ON   OFF      BIT  ON   OFF      BIT  ON   OFF
 0  RENT  NOT RENT  4  OL    NOT OL   8  NOT DC  DC       12  NOT EDIT  EDIT
 1  REUS  NOT REUS  5  SCTR  BLOCK    9  ZERO ORG  NOT ZERO 13  SYMS     NO SYMS
 2  ONLY  NOT ONLY  6  EXEC  NOT EXEC 10  EP ZERO  NOT ZERO 14  F LEVEL  E LEVEL
 3  TEST  TEST      7  1 TXT  MULTI RCD 11  NO RLD   RLD       15  REFR     NOT REFR

MEMBER  ENTRY  ATTR  REL  ADDR-HEX  CONTIG  LEN 1ST  ORG 1ST  SST  VS  AUTH  OTHER
NAME   PT-HEX  HEX   BEGIN 1ST TXT  STOR-DEC  TXT-DEC  TXT-HEX  INFO  ATTR  REQ  REQ INFORMATION
LOADMOD 000000 06E2  000004 00020F  000166248 0927  ABSENT 880000 NO  SCTR=000000
                                00484,01084,32,32

OF THE 00002 DIRECTORY BLOCKS ALLOCATED TO THIS PDS, 00001 ARE(IS) COMPLETELY UNUSED
```

図 39. 区分ディレクトリーの編集された項目のサンプル

メンバー名、属性 (フィールド 3 と 10)、およびその他の情報 (フィールド 12) の後にアスタリスク (*) が付いている場合は、ディレクトリーの項目の最初のページを印刷する前に、それについて説明した索引が印刷されます。OTHER INFORMATION INDEX の下には、分散フォーマットおよびオーバーレイ・フォーマットのデータが、リスト内と同じ定位置で記述されます。ATTRIBUTE INDEX の下には、それぞれの属性ビットの意味が説明されます。VS ATTR フィールドの索引はありません。このフィールドに表示されるデータは、PDS ディレクトリー内の PDS2FTBO フィールドからのデータです。

各ディレクトリー項目はそれぞれ 1 行で印刷されます。ただし、メンバー名が別名であり、しかも主メンバー名および関連した入り口点がユーザー・データ・フィールド内に表示されている場合は例外です。このような場合は 2 行にわたって印刷され、それぞれの別名の後にはアスタリスクが付きます。主メンバーの名前が変更されている場合、元のメンバー名が依然として別名ディレクトリー項目の中に表示されるために、それが第 2 行目に印刷されます。

非編集 (ダンプ) フォーマット

IEHLIST による非編集フォーマットのリスト表示を使用すると、すべてのタイプの区分データ・セットまたは PDSE のディレクトリーをリスト表示できます。リンケージ・エディターによってメンバーが作成されなかった区分データ・セットまたは PDSE のディレクトリーは、非編集フォーマットを使用すると最も効果的にリスト表示できます。編集されていないリスト表示では、各メンバーが別々に、16 進でリスト表示されます。

図 40 は、3 つのメンバーからなる区分データ・セットまたは PDSE ディレクトリーの、非編集リスト・フォーマットを示したものです。ただし、この図が示しているのは、IEHLIST プログラムが作成する非編集フォーマットのリストの一例に過ぎません。実際に IEHLIST プログラムが作成する編集されていないリスト表示は、これとは異なる可能性があります。

注: 図 40 のような整理された区分データ・セットまたは PDSE ディレクトリーのリストは、ほかに IEBPTPCH を使用しても作成できます (211 ページの『第 9 章 IEBPTPCH (印刷 / 穿孔) プログラム』を参照)。

```
MEMBERS      TTRC      VARIABLE USER DATA ---(USER DATA AND TTRC ARE IN HEX)
MEMBER1      0009150F  0100000000 89135F0089 135F101300 5C005C0000 D1C1D9C5C4
MEMBER2      000E010F  0100001800 92217F0092 217F163900 1300130000 C9C2D4E4E2
MEMBER3      000D0B0F  0100000000 91194F0091 194F125100 1100110000 D1C1D9C5C4
```

図 40. 区分データ・セットまたは PDSE ディレクトリーの非編集リスト・フォーマット: 印刷される 16 進文字の各グループは 5 バイトです。

ユーザー・データ情報を正しく解釈するには、区分化された項目のフォーマットを理解している必要があります。ディレクトリー項目のフォーマットについては、*z/OS DFSMS* データ・セットの使用法 で説明されています。

ボリューム目録のリスト表示

IEHLIST を使用して、指定されたボリューム目録 (VTOC) 内の項目を、部分的または全体的にリスト表示することができます (索引付きまたは索引なしで)。このプログラムは、選択したデータ・セット制御ブロック (DSCB) の内容を、編集フォーマットまたは非編集フォーマットでリスト表示します。

索引付き VTOC の詳細については、*z/OS DFSMSdfp* 拡張サービスを参照してください。

編集フォーマット

使用できる編集フォーマットは 2 つあります。

1 次編集フォーマット

最初の編集フォーマットは、VTOC 内の DSCB の包括的なリスト表示です。これはボリュームの状況と属性、およびボリューム上にあるデータ・セットの詳細情報を提供します。このリストは以下のものを含みます。

- 論理レコード長とブロック・サイズ
- 初期割り振りおよび 2 次割り振り
- エクステンツの上限と下限
- 代替トラックについての情報
- 使用可能スペースについての情報 (未割り振りのシリンダーやトラックについて、および未割り振り (フォーマット 0) の DSCB について)
- オプション・コード (2 桁の 16 進数字として印刷される)
- レコード・フォーマット
- SMS 標識

リストに現れる最初の DSCB は、常に VTOC (フォーマット 4) DSCB です。これは、VTOC 自体のスコープを定義します。つまり、VTOC に参照されるデータ・セットではなく、VTOC およびボリュームについての情報を含んでいます。

索引付き VTOC: 索引付き VTOC の場合、リストの編集フォーマットには 2 つのタイプがあります。これらのタイプは INDEXDSN パラメーターを使って指定されます。

INDEXDSN を省略すると、作成されるリストには次のものが含まれます。

- 索引内のレベルの数を示す文 (索引が使用可能な場合)。
- フォーマット済みのフォーマット 4 の DSCB。
- 英数字順のフォーマット済みのデータ・セット項目 (索引が使用不可の場合は、物理順のフォーマット 1 DSCB)。
- フォーマット済みのフリー・スペース情報。
- 未割り振りのシリンダーやトラック、未割り振り (フォーマット 0) の DSCB、および未割り振りの VIR のそれぞれの合計。

INDEXDSN=name が指定されていれば、上の項目に加えて、次のような項目がリストに含まれます。

- スペースおよび割り振りに関するフォーマット済みの情報。
- レベル順およびレベル内のキー・シーケンスにリスト表示された、割り振り済み VTOC 索引項目のフォーマット済みの情報 (索引が使用不可の場合は、物理順)。
- VTOC 索引が使用不可の場合は、その旨を示す文が含まれます。

注: リスト内に各 DSCB が表示される様子を示した、1 次編集フォーマットのサンプルについては、387 ページの『付録 D. IEHLIST の VTOC リスト表示』を参照してください。

第 2 編集フォーマット

第 2 編集フォーマットは、短縮された形でデータ・セットを記述します。特定のフォーマットを要求しない場合は、デフォルトとしてこのフォーマットが使用されます。これは、次のような情報を提供します。

- データ・セット名
- 作成日 (yyyy.ddd)
- 有効期限 (yyyy.ddd)
- パスワード表示
- データ・セットの編成
- エクステンント
- ボリューム通し番号
- SMS 標識

リストの最終行は、VTOC 内に残っているスペースの量を示します。

索引なし VTOC の場合、データ・セットの項目は物理順にリスト表示されます。また、未割り振りのシリンダーやトラック、および未割り振り (フォーマット 0) の DSCB のそれぞれの合計もリスト表示されます。

索引付き VTOC の場合、このリストは次のものを含みます。

- 索引内のレベルの数を示す文。
- 英数字順にリストされた、データ・セット項目。
- 未割り振りのシリンダーやトラック、未割り振り (フォーマット 0) の DSCB、および未割り振りの VIR のそれぞれの合計。
- SMS 標識。

非編集 (ダンプ) フォーマット

このオプションは、VTOC 内の DSCB の完全なリストを 16 進数で作成します。このリストは、非編集 (ダンプ) フォーマットであり、これを扱うには、適用できる DSCB のさまざまなフォーマットを理解する必要があります。VTOC オーバーレイを使用してダンプ・フォーマットによる VTOC の IEHLIST リストを作成すると、DSCB のフィールドを識別するうえで役立ちます。

索引なし VTOC の場合、このリストは次のものを含まれます。

- 物理順の、ダンプされた DSCB。
- 未割り振りのシリンダーやトラック、および未割り振り (フォーマット 0) の DSCB のそれぞれの合計。

索引付き VTOC の場合、リストのダンプ・フォーマットには 2 つのタイプがあります。これらのタイプは INDEXDSN パラメーターを使って指定されます。

INDEXDSN を省略すると、作成されるリストには次のものが含まれます。

- 物理順の、ダンプされた DSCB。
 - 装置のトラックが 64K 以下であれば、単一トークンのフォーマット 5 DSCB と識別されます。
 - 装置のトラックが 64K を超えていれば、トークンのフォーマット 5 DSCB およびトークンのフォーマット 7 DSCB と識別されます。
- 未フォーマットのフリー・スペースに関する情報。
- 未割り振りのシリンダーやトラック、未割り振り (フォーマット 0) の DSCB、および未割り振りの VIR のそれぞれの合計。

INDEXDSN=*name* が指定されていれば、上の項目に加えて、次のような項目がリストに含まれます。

- ダンプ・フォーマットの、スペースおよび割り振りに関する情報。

VTOC 索引が使用不可の場合は、割り振り済みレコードと未割り振りレコードの両方が、物理順でダンプされます。

- VTOC 索引が使用不可の場合は、その旨を示す文が含まれます。

注: 大規模ボリュームの場合、使用可能な DSCB 数には、索引された VTOC 数に対して最大値の 65499 が入ります。一方、そのサマリー・セクションは正しいブランクの DSCB 数を示します。

データ・セット制御ブロックに想定できるさまざまなフォーマットについての説明は、*z/OS DFSMSdfp* 拡張サービスを参照してください。

入出力

IEHLIST は以下の入力を使用します。

- リスト表示されるデータが入っている、1 つまたは複数のソース・データ・セット。入力データ・セットには次のものを使用できます。
 - VTOC
 - 区分データ・セットまたは PDSE
- 制御データ・セット。これに含まれるユーティリティー制御ステートメントを使って、IEHLIST の機能を制御します。

IEHLIST は、出力として、IEHLIST 操作の結果を収めているメッセージ・データ・セットを生成します。このメッセージ・データ・セットには、データのリストとエラー・メッセージが含まれています。

アプリケーション・プログラムから IEHLIST を起動する場合は、IEHLIST を呼び出す前に SVC 99 を出すことによって、装置やデータ・セットを動的に割り振ることができます。

IEHLIST の戻りコードについては、付録 A を参照してください。

制御

IEHLIST は、ジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。ジョブ制御ステートメントは、IEHLIST を処理またはロードするため、あるいは IEHLIST が使用するデータ・セットや作成するデータ・セットを定義するために使います。

ユーティリティー制御ステートメントは、プログラムの機能を制御したり、変更されるデータ・セットおよびボリュームを定義したりするのに使います。

ジョブ制御ステートメント

表 41 は、IEHLIST で使用するジョブ制御ステートメントをまとめたものです。

表 41. IEHLIST で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEHLIST)。このジョブ制御ステートメントがプロシージャ・ライブラリー内にある場合は、プロシージャ名を指定します。また、PARM に関する追加情報を指定することによって、各ページごとに印刷される行数を制御できます。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。
anyname (任意名) DD	永続的に取り付けられたボリュームまたは永続的に取り付け可能なボリュームを定義します。割り振る装置の最大数は、ジョブ・ステップ当たり 256 を超えてはなりません。
SYSIN DD	制御データ・セットを定義します。制御データ・セットは通常、入力ストリーム内でジョブ制御言語の後に配置します。しかし、これを非ブロック化順次データ・セットまたはプロシージャ・ライブラリーのメンバーとして定義することもできます。

IEHLIST では、リスト表示しようとしているオブジェクトの逐次化はできません。IEHLIST がデータ・セットを読み取ろうとする直前に、別のプログラムがデータ・セットのブロックを更新した場合は、メッセージ (IEH105I または IEH114I) が出力されることがあり、IEHLIST の作成した出力は正しくない可能性があります。このような場合は、ジョブを再実行してください。

EXEC ステートメント

EXEC ステートメントの PARM パラメーターを使用して、IEHLIST が各ページごとに出力を印刷する行数を制御できます。EXEC ステートメントは以下のように記述します。

//[stepname]	EXEC	PGM=IEHLIST[,PARM='LINECNT=xx']
--------------	------	---------------------------------

各項目の意味は以下のとおりです。

PGM=IEHLIST

実行するプログラムが IEHLIST であることを指定します。

PARM='LINECNT=xx'

各ページごとに xx 行を印刷することを指定します。xx に指定する数は、01~99 です。LINECNT を指定しない場合は、各ページごとに 58 行が印刷されます。

LINECNT を指定しない場合、PARM には、埋め込まれたブランク、ゼロ、およびほかの PARM キーワードを含めることはできません。

SYSPRINT DD ステートメント

SYSPRINT のブロック・サイズは、121 の倍数でなければなりません。このブロック・サイズに、任意のブロック化因数を指定できます。

anyname (任意名) DD ステートメント

永続的に取り付けられているボリュームまたは永続的に取り付け可能なボリュームのうち、ジョブ・ステップの中で参照されているものについては、それぞれ 1 つずつ DD ステートメントを含める必要があります。これらの DD ステートメントは、装置の割り振りに使われます (実際にはこれらはデータ定義ステートメントではありません)。連結した DD ステートメントは使用できません。

IEHLIST は、装置割り振り DD ステートメントによって作成された内部制御ブロックを変更します。このため、これらの DD ステートメントに DSNAME パラメーターを含めてはなりません。(すべてのデータ・セットは、明示的にまたは暗黙のうちにユーティリティ制御ステートメントによって定義されます。)

DD ステートメントは、次のように入力できます。

```
//anyname DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

UNIT および VOLUME=SER パラメーターは、装置タイプおよびボリューム通し番号を定義しますが、その際に実際のデータ・セットをボリューム上に割り振る必要はありません。

SYSIN DD ステートメント

SYSIN のブロック・サイズは、80 の倍数でなければなりません。このブロック・サイズに、任意のブロック化因数を指定できます。SYSIN の場合は DD ステートメントを連結することができます。

ユーティリティー制御ステートメント

表 42 は、IEHLIST で使用するユーティリティー制御ステートメントをまとめたものです。

表 42. IEHLIST で使用するユーティリティー制御ステートメント

ステートメント	用途
LISTPDS	1 つまたは複数の区分データ・セットまたは PDSE のディレクトリー・リスト表示を要求します。
LISTVTOC	ボリューム目録の全体または一部のリスト表示を要求します。

ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

LISTPDS ステートメント

LISTPDS ステートメントを使用して、同じボリュームに収められている 1 つまたは複数の区分データ・セットまたは PDSE のディレクトリー・リスト表示を要求します。

LISTPDS ステートメントの FORMAT オプションは、リンケージ・エディターでメンバーが作成された区分データ・セット上でのみ使用できます。モジュール・ライブラリーの編集されたリスト表示については、281 ページの『区分データ・セットまたは PDSE のディレクトリーのリスト表示』を参照してください。メンバーがロード・モジュールではない PDSE や区分データ・セットの場合に FORMAT を使用すると、作成されるリストには信頼できない情報が含まれます。

LISTPDS ステートメントの構文は次のとおりです。

[label]	LISTPDS	DSNAME=(name[,name[,...]]) [VOL=device=serial] [DUMP FORMAT]
---------	---------	--------------------------------------------------------------------

各項目の意味は以下のとおりです。

DSNAME=(name[,name[,...]])

ディレクトリーをリスト表示する区分データ・セットまたは PDSE の完全修飾名を指定します。最大で 10 個の名前を指定できます。名前を 1 つしか指定しない場合は、括弧を省略することができます。

VOL=device=serial

区分データ・セットまたは PDSE ディレクトリーが収められているボリュームの、装置タイプおよびボリューム通し番号を指定します。区分データ・セットまたは PDSE がシステム常駐ボリューム上にない場合、VOL パラメーターの指定は必須です。

DUMP

非編集の 16 進フォーマットでリスト表示することを指定します。デフォルトは DUMP です。

FORMAT

各ディレクトリー項目ごとにリストを編集することを指定します。

FORMAT オプションは、リンケージ・エディターでメンバーが作成された区分データ・セット上でのみ使用できます。リンケージ・エディターによって作成されなかったメンバーのディレクトリー項目は、非編集 (DUMP) フォーマットでリスト表示されます。

LISTVTOC ステートメント

LISTVTOC ステートメントを使用して、指定したボリューム目録内の項目の一部または全部をリスト表示するよう要求します。

VTOC および索引付き VTOC の索引付きデータ・セットの 両方をリスト表示するために IEHLIST を使用している場合は、283 ページの『ボリューム目録のリスト表示』を参照してください。

LISTVTOC ステートメントの構文は次のとおりです。

[label]	LISTVTOC	[{DUMPIFORMAT[,PDSESPACE]}] [,INDEXDSN=SYS1.VTOCIX.xxxx] [,DATE={ddyy\ddyyyy}] [,VOL=device=serial] [,DSNAME=(name[,name[,...]])]
---------	----------	-----------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

DUMP

非編集の 16 進フォーマットでリスト表示することを指定します。このダンプ・オプションによって、SMS 標識とフィールドが、それぞれ VTOC (フォーマット 1 およびフォーマット 4 の DSCB) および VTOC 索引 (16 進フォーマット) に表示されます。

FORMAT [,PDSESPACE]

包括的な編集されたリスト表示を生成することを指定します。

PDSESPACE を指定した場合は、割り振られたスペース、および使用中のスペースが表示されます。これらは PDSE データ・セットに関してのみ表示されず (K バイト単位)。

FORMAT と DUMP をどちらも省略した場合は、短縮された編集フォーマットで生成されます。

INDEXDSN=SYS1.VTOCIX.xxxx

VTOC に加えて、索引情報もリスト表示することを指定します。xxxx の値は、任意の第 3 レベルの修飾子です。INDEXDSN を指定した場合は、DUMP または FORMAT のいずれかを指定しなければなりません。索引付き VTOC の詳細については、283 ページの『ボリューム目録のリスト表示』を参照してください。

DATE={dddyl|dddyyyy}

この日付よりも前に有効期限になる各項目について、リスト内の項目名の後にアスタリスク (*) のフラグを付けることを指定します。このパラメーターが適用できるのは、短縮された編集フォーマットだけです。dddylという指定は、ddd19yyと同じです。

VTOC の「期限切れなし」という有効期限 (1999.365 または 1999.366) に対しては、特別な処理が行われます。つまり、DATE= パラメーターが 1999 年より後の日付を指定しても、このような有効期限は、アスタリスク (*) でフラグを立てられません。「期限切れなし」という有効期限は、どのような DATE= を指定をしても、アスタリスクでフラグを立てられることはありません。

DATE= パラメーターは、通常の日付を指定します。DATE=36599 のような日付が指定された場合でも、「期限切れなし」という日付を意味しません。実際には、単に 1999/12/31 の日付を意味するだけです。したがって、VTOC 有効期限が 1999 年以降の日付の場合は、アスタリスクでフラグが立てられることはありません。

1999 年より後の日付をコーディングすると、36599、36699、3651999、3661999、99999、または 9991999 の特別な有効期限を持つデータ・セットは、アスタリスクでフラグを立てられません。これらのデータ・セットは、有効期限が切れることはありません。

ユーザーがコーディングする日付は、「期限切れなし」の日付としては扱われません。たとえば、36599 または 3651999 とコーディングすると、1999 年 12 月 30 日に有効期限が切れるデータ・セットにはフラグが立てられ、有効期限が 1999 年 12 月 31 日および 2000 年 1 月 1 日であるデータ・セットには、フラグが立てられません。該当する年の日数を超える ddd 値をコーディングすると、IEHLIST は、余った日数を翌年に追加します。たとえば、36699 は、2000 年 1 月 1 日を意味し、3671999 は、2000 年 1 月 2 日、9999 は 2001 年 9 月 25 日を意味します。

dddyl

ddd はその年の日を指定し、年の最後 2 桁である yl は 2000 より前の年をこの形式で指定します。

dddyyyy

yyyy は 1900 年から 2155 年までの範囲の年を、ddd はその年の日をそれぞれ指定します。

デフォルト: リストにはアスタリスクが表示されません。

VOL=device=serial

VTOC が収められているボリュームの、装置タイプおよびボリューム通し番号を指定します。

DSNAME=(name[,name[,...]])

項目をリスト表示するデータ・セットの完全修飾名を指定します。最大で 10 個の名前を指定できます。名前を 1 つしか指定しない場合は、括弧を省略することができます。

注: 指定した DSNAME が VSAM クラスタ名である場合、戻される情報は、物理装置 (ボリューム) 上にデータ・セットが存在することを表します。た

だし、VSAM クラスタ用にはフォーマット 1 DSCB がないため、VSAM クラスタ名に対する出力は、通常データ・セットに対する出力と同じではありません。

IEHLIST の例

以下の各例では、IEHLIST の使用方法の一部を示します。表 43 は IEHLIST の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

表 43. IEHLIST の例の一覧表

操作	装置	注釈	例
LISTPDS	ディスクおよびシステム出力装置	1 つの PDSE ディレクトリーおよび 2 つの区分データ・セット・ディレクトリーがリスト表示されます。	1
LISTVTOC	ディスクおよびシステム出力装置	ボリューム目録が編集フォーマットでリスト表示されます。選択したデータ・セット制御ブロックが非編集フォーマットでリスト表示されます。	2

実際の装置名または装置名の指定する場所に **disk** を使用している例は、使用する前に変更する必要があります。実際の装置番号または装置名は、システムに対する装置定義の方法によって異なります。

例 1 DUMP および FORMAT を使用して、区分化されたディレクトリーをリスト表示する

この例では、PDSE のディレクトリーをリスト表示します。また、システム常駐ボリュームにある 2 つの区分データ・セットのディレクトリーもリスト表示します。

```
//LISTPDIR JOB ...
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=diskB,VOLUME=REF=SYS1.NUCLEUS,DISP=OLD
//DD2 DD UNIT=diskA,VOLUME=SER=222222,DISP=OLD
//SYSIN DD *
LISTPDS DSNAME=D42.PDSE1,VOL=diskA=222222
LISTPDS DSNAME=(D55.PART1,D55.PART2),FORMAT
/*
```

各制御ステートメントについての説明は以下のとおりです。

- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、ディスク・ボリューム (222222) が取り付けられている装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- 最初の LISTPDS ステートメントでは、データ・セット D42.PDSE1 に属する PDSE のディレクトリーをリスト表示ことを示しています。非編集 (ダンプ) フォーマットでリストが作成されます。データ・セットはボリューム 222222 にあります。
- 2 番目の LISTPDS ステートメントでは、データ・セット D55.PART1 および D55.PART2 に属する区分データ・セットのディレクトリーをリスト表示することを示しています。リストは、編集フォーマットで作成されます。これらのデータ・セットは、システム常駐ボリュームにあります。

283 ページの図 40 は、区分化されたメンバーのための非編集フォーマットの項目を示しています。282 ページの図 39 は編集された項目を示しています。

例 2 索引なしボリューム目録をリスト表示する

この例では、索引なしボリューム目録を 1 次編集フォーマットでリスト表示します。編集されたリストの補足として、選択したデータ・セット制御ブロックを非編集フォーマットでリスト表示します。

```
//VTOCLIST JOB ...
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD2 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//SYSIN DD *
LISTVTOC FORMAT,VOL=disk=111111
LISTVTOC DUMP,VOL=disk=111111,DSNAME=(SET1,SET2,SET3)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- DD2 DD では、指定したボリューム目録のある装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- 最初の LISTVTOC ステートメントでは、指定したディスク・ボリューム上のボリューム目録を編集フォーマットでリスト表示することを示しています。
- 2 番目の LISTVTOC ステートメントでは、それぞれデータ・セット SET1、SET2、および SET3 を表すデータ・セット制御ブロックが、非編集フォーマットでリスト表示されることを示しています。

第 13 章 IEHMOVE (システム・データの移動) プログラム

IEHMOVE は、オペレーティング・システム・データの論理集合を移動またはコピーするために使用するシステム・ユーティリティです。

IEHMOVE に関して以下で説明する事柄は、互換性についての情報のみです。ストレージ管理サブシステムによって管理されているボリュームにデータを移動またはコピーするには、IEHMOVE ではなく DFSMSdss を使用する必要があります。PDSE を処理するには、DFSMSdss または IEBCOPY を使用する必要があります。PDSE、ISAM あるいは VSAM データ・セット、またはラージ・フォーマットの順次データ・セットに対して IEHMOVE を使用できません。

SMS 管理ボリュームに、IEHMOVE を使ってデータ・セットを移動またはコピーする場合には、ターゲット・データ・セットをすべて事前に割り当てる必要があります。コピーまたは移動するデータ・セットがカタログ化されていて、かつ SMS 管理ボリュームにこれを移動またはコピーする場合、そのデータ・セットを名前変更しなければなりません。

SMS 管理データ・セットを割り振る方法の詳細については、*z/OS DFSMS データ・セットの使用法* を参照してください。また、ALLOCATE コマンドの詳細については、*z/OS DFSMS カタログのためのアクセス方式サービス・プログラム* を参照してください。

IEHMOVE を使用して、以下のものを移動またはコピーすることができます。

- 1~5 個のボリュームに収められている、順次データ・セット、区分データ・セット、または BDAM データ・セット。
- 統合カタログ機能カタログにカタログ化された非 VSAM データ・セットのグループ。
- 複数のデータ・セットからなる 1 つのボリューム。
- 可変長スパン・レコードを含んでいる BDAM データ・セット。

移動操作では、データ・セットが直接アクセス・ボリュームに収められていて有効期限を過ぎた場合、ソース・データがスクラッチされます。一方、コピー操作では、ソース・データはそのまま残ります。さらに、カタログ式データ・セットの場合、移動操作では (特にほかの指定がない限り)、移動先のデータ・セットを参照するようにカタログを更新しますが、コピー操作ではカタログを更新しません。

以下のようにして、基本的な移動操作またはコピー操作の範囲を拡大することができます。

- 移動操作またはコピー操作にデータ・セットを含める、または操作からデータ・セットを除く
- 複数の区分データ・セットからメンバーをマージする
- 選択したメンバーを含める、または除く
- 移動またはコピーしたメンバーを名前変更する
- 選択したメンバーを置き換える

パスワードで保護されたデータ・セットを含んでいるデータ・セット・グループやボリュームを移動またはコピーする場合、データ・セットをオープンまたはスクラッチするたびに、パスワードを入力する必要があります。

IEHMOVE は、入力データ・セットに関連付けられたすべてのユーザー・ラベルを、常に移動またはコピーします。ユーザー自身のラベルを使用して、IEHMOVE のルーチンを処理することはできません。

移動操作またはコピー操作では、データ・セットが移動またはコピーされるか、何も処理されないか、ソース・データ・セットがアンロードされるかのいずれかの結果になります。

注: 指定したデータを正常に移動またはコピーできない場合、IEHMOVE はそのデータの再編成、および指定した出力装置へ入れようと試みます。再編成されたデータ (アンロードされたデータ・セットという) とは、80 バイトのブロック化レコードからなる順次データ・セットであり、これにはソース・データを最初と同じ状態に再構築するのに必要なソース・データおよび制御情報が含まれています。上記のどの結果になるかは、以下の点に関連した、ソース・ボリュームと受け取り側ボリュームとの互換性によって決まります。

- ボリュームのサイズ
- 受け取り側ボリュームにおけるスペース割り振り
- データ・セットの編成 (順次、区分、または BDAM)
- ソース・データ・セットの移動可能性

ボリューム・サイズの互換性についての考慮事項

区分データ・セット編成または順次データ・セット編成を使用している場合、ソース・レコード・サイズが受け取り側ボリュームのトラック・サイズを超えていない限り、2 つのボリュームにはサイズに関して互換性があります。

BDAM データ・セット編成を使用している場合、ソース・ボリュームのトラック容量が受け取り側ボリュームのトラック容量を超えていない限り、2 つのボリュームにはサイズに関して互換性があります。より小さい装置タイプやテープに移動またはコピーされた BDAM データ・セットは、アンロードされます。アンロードされたデータ・セットをロードする場合には、アンロードされる前と同じ装置タイプにロードしなければなりません。

表 44 は、受け取り側ボリュームが DASD ボリュームであり、サイズに関してソース・ボリュームと互換性のある場合の、移動操作およびコピー操作の結果を示したものです。ソース・データ・セットの編成、および受け取り側ボリュームの特性が示されています。

表 44. 移動操作およびコピー操作 - サイズに関してソース・ボリュームと互換性のある DASD 受け取り側ボリューム

受け取り側ボリューム の特性	順次データ・セット	区分データ・セット	BDAM データ・ セット
IEHMOVE によって 割り振られたスペース (移動可能なデータ)	移動またはコピーされ る	移動またはコピーされ る	移動またはコピーされ る

表 44. 移動操作およびコピー操作 - サイズに関してソース・ボリュームと互換性のある DASD 受け取り側ボリューム (続き)

受け取り側ボリューム の特性	順次データ・セット	区分データ・セット	BDAM データ・ セット
IEHMOVE によって 割り振られたスペース (移動不能なデータ)	移動またはコピーされ る	移動またはコピーされ る	処理されない
以前に割り振られた、 未使用のスペース	移動またはコピーされ る	移動またはコピーされ る	処理されない
以前に割り振られ、一 部が使用されているス ペース	処理されない	移動またはコピーされ る (マージされる)	処理されない

表 45 は、受け取り側ボリュームが DASD ボリュームであり、サイズに関してソース・ボリュームと互換性のない場合の、移動操作およびコピー操作の結果を示したものです。ソース・データ・セットの編成、および受け取り側ボリュームの特性が示されています。

表 45. 移動操作およびコピー操作 - サイズに関してソース・ボリュームと互換性のない DASD 受け取り側ボリューム

受け取り側ボリューム の特性	順次データ・セット	区分データ・セット	BDAM データ・ セット
IEHMOVE によって 割り振られたスペース	アンロードされる	アンロードされる	アンロードされる
以前に割り振られた、 未使用のスペース	アンロードされる	アンロードされる	処理されない
以前に割り振られ、一 部が使用されているス ペース	処理されない	処理されない	処理されない

表 46 は、受け取り側ボリュームが DASD ボリュームではない場合の、移動操作およびコピー操作の結果を示したものです。ソース・データ・セットの編成、および受け取り側ボリュームの特性が示されています。

表 46. 移動操作およびコピー操作 - 非 DASD の受け取り側ボリューム

受け取り側ボリューム の特性	順次データ・セット	区分データ・セット	BDAM データ・ セット
移動可能なデータ	移動またはコピーされ る	アンロードされる	アンロードされる
移動不能なデータ	アンロードされる	アンロードされる	処理されない

移動またはコピーされるデータ・セットのためのスペース割り振り

ユーザーが DD ステートメントを使用することによって、または IEHMOVE ジョブ・ステップの中の IEHMOVE によって、受け取り側ボリューム上にデータ・セット用のスペースを割り振ることができます。

ソース・データが移動不能である場合 (つまり、位置に依存するコードが含まれている場合)、受け取り側ボリュームのスペース割り振りに絶対トラック割り振りを使用することによって、受け取り側ボリュームに収められるデータ・セットの相対位置が、必ずソース・ボリュームと同じになるようにしなければなりません。

IEHMOVE によってスペースが割り振られる場合、移動不能なデータを移動またはコピーすることはできますが、ソース・ボリュームにあったときと同じ位置に、データが受け取り側ボリュームでも置かれるとは限りません。

類似していない DASD 装置の間でデータ・セットを移動またはコピーする場合、2 次割り振りを行って、受け取り側ボリュームに十分なスペースが使用できるようにする必要があります。

BDAM データ・セットを移動またはコピーする前に (アンロードされていない時)、ユーザーが新規データ・セット用にスペースを割り振ってはなりません。これは、新規データ・セットが空かどうかを IEHMOVE が判別できなくなるからです。

IEHMOVE によって新規データ・セットのスペース割り振りが行われる場合、古いデータ・セットがあれば、そのスペース要件情報が適用されます。このスペース要件情報は、DASD ボリューム上にある場合はソース・データ・セットの DSCB から、アンロードされたデータ・セットの場合は制御情報から得られます。

スペース要件情報が使用可能な場合、IEHMOVE はこの情報を使って、トラック容量やオーバーヘッド因数などの装置特性の違いを考慮に入れながら、受け取り側ボリュームのスペースの割り振りを行います。ただし、可変長または不定形式のレコード・フォーマットを含むデータ・セットを、類似していない DASD 装置の間で移動またはコピーする時には、受け取り側デバイス上の各レコードが必要とするスペースに関しては、何も想定されません。

通常、可変長または不定形式のレコード・フォーマットを移動またはコピーするとき、IEHMOVE は十分なスペースの割り振りを試みます。この場合、以下のような状況では、必要以上にスペースが割り振られる可能性があります。

- ブロック・オーバーヘッドが比較的大きい装置から、ブロック・オーバーヘッドのより小さい装置に、ブロック・サイズの小さなブロックを移動またはコピーするとき。
- ブロック・オーバーヘッドが比較的小さい装置から、ブロック・オーバーヘッドのより大きい装置に、ブロック・サイズの大きなブロックを移動またはコピーするとき。

可変長または不定形式のレコード・フォーマットを持つ BDA M データ・セットには、常に同じサイズのスペースが IEHMOVE によって割り振られます。これによって、データ・セット内に存在するすべての相対トラック・アドレッシング・システムが維持されます。

非 DASD ボリューム上にある、アンロードされたデータ・セットではない順次データ・セットを DASD ボリュームに移動またはコピーする場合、以前の割り振りからスペース属性を利用できなければ、IEHMOVE はデフォルトのスペース割り振りを行います。デフォルトの割り振りの構成は、72,500 バイトの DASD ストレージ (データおよびギャップ) の 1 次割り振りが 1 つ、および各 36,250 バイトの 2 次割り振りが最大で 15 個までです。

割り振りをする DD ステートメントの SPACE パラメーターで順次編成を暗黙指定しないかぎり、アンロードされる区分データ・セット用にスペースを前もって割り振ることはできません。BDAM データ・セットを前もって割り振らないください。それが空かどうかを IEHMOVE が判別できなくなるためです。

移動操作またはコピー操作が失敗した場合、ソース・データはそのまま残ります。

移動操作またはコピー操作に失敗した場合、スペースが IEHMOVE によって割り振られていれば、その操作に関連したすべてのデータは受け取り側 DASD ボリュームからスクラッチされます。受け取り側ボリュームがテープの場合は、データ・セットの一部がこれに含まれます。

スペースを前もって割り振っていても、移動操作またはコピー操作に失敗した場合、受け取り側ボリュームからデータはスクラッチされません。たとえば、105 個のメンバーからなる区分データ・セットのうち 104 個を IEHMOVE が移動し終えて、105 番目のメンバーの移動中に入出力エラーが発生した場合、次のようになります。

- IEHMOVE によってスペースが割り振られた場合、区分データ・セット全体が受け取り側ボリュームからスクラッチされます。
- スペースを事前に割り振った場合には、受け取り側ボリュームからデータはスクラッチされません。この場合、エラーの性質を判別した後、105 番目のメンバーだけを受け取り側の区分データ・セットに移動します。

ユーザー・トレーラー・ラベルだけで構成されるデータ・セットをテープ・ボリュームから DASD ボリュームに移動する場合、事前にスペースを DASD ボリュームに割り振ることによって、ユーザー・ラベルを受け取るためのトラックを必ず予約する必要があります。

データ・セットの再ブロック

希望のブロック・サイズを事前に受け取り側ボリュームに割り振ることによって、固定長レコードまたは可変長レコードを含むデータ・セットを異なるブロック・サイズに再ブロックすることができます。ロード中またはアンロード中には、再ブロックを実行できません。また、可変長スパン・レコードまたは可変長ブロック・スパン・レコードを含むデータ・セットでは、再ブロックを実行できません。

不定形式のレコード・フォーマットを含むデータ・セットを移動またはコピーして、より小さいブロック・サイズで再ブロックする場合（つまり、元の装置のトラック容量よりも小さいトラック容量の装置にレコードを転送する場合）、受け取り側ボリュームのブロック・サイズを、移動元またはコピー元データ・セットの最も大きいレコードのサイズと同じか、またはより大きくする必要があります。

不定形式のレコード・フォーマットを含むデータ・セットを、より大きいトラック容量の装置にコピーする場合、IEHMOVE は出力データ・セットをより大きなブロック・サイズに再ブロックしません。IEHMOVE は、単にソース・データ・セットをターゲット・データ・セットにコピーするだけです。

しかし、ターゲット・データ・セットをソース・データ・セットよりも大きなブロック・サイズで事前に割り振った場合、コピー中にはソース・ブロック・サイズが使われるため、データ・セットは使用不可になります。

ユーザー・データ TTRN やキーを含まないブロック化フォーマット・データ・セットの場合、(そのデータ・セットへの事前スペース割り振りに使用する) DD ステートメントの DCB オペランドに、適切なキーワード・サブパラメーターを組み込むことによって、再ブロック化または非ブロック化できます。新規ブロック化因数は、最初にデータ・セットに割り当てられた論理レコード長の倍数でなければなりません。ユーザー・データ TTRN については、z/OS DFSMS データ・セットの使用法を参照してください。

RACF[®] における IEHMOVE の使用

OS/390 のセキュリティー・サーバーのコンポーネントであるリソース・アクセス管理機能 (RACF) がアクティブの場合、以下の事項が考慮されます。

- RACF で定義されたデータ・セットに IEHMOVE を使用してアクセスするには、有効な RACF 許可が必要です。MOVE 機能の実行のためにソース・データ・セットにアクセスするには、ソース・データがスクラッチされるので、ALTER 許可が必要です。複数のデータ・セットからなる 1 つのボリューム全体、または 1 つのグループ全体を移動する場合には、ボリューム上またはグループ内の RACF によって保護されたすべてのデータ・セットに対する、適切なアクセス許可が必要です。
- RACF ADSP 属性があれば、IEHMOVE によって受け取り側データ・セットのスペースを割り振る時、そのデータ・セットは自動的に RACF に定義されます。データ・セットの 1 次レベル修飾子がユーザー ID ではない場合、以下の条件のうち少なくとも 1 つが満たされなければなりません。
 1. RENAME とともに MOVE または COPY を指定することによって、1 次レベル修飾子を正しいユーザー ID にする。
 2. 移動元またはコピー元のデータ・セットがグループ・データ・セットであり、CREATE 許可でそのグループに接続している。
 3. OPERATION 属性がある。
- COPYAUTH を指定した場合、ユーザーが ADSP 属性を持っているかどうかにかかわらず、入力データ・セットが RACF 保護されていて、しかも出力データ・セットが事前割り振りされていなければ、MOVE または COPY 操作の受け取り側データ・セットには、入力データ・セットの RACF 保護アクセス・リストのコピーが割り振り中に渡されます。このリストは同じ制限で管理され、上記のように、ADSP 属性を持つユーザーに対するデータ・セット定義に使われます。COPYAUTH を使って MOVE または COPY のいずれかを行うには、入力データ・セットに対する ALTER アクセス許可が必要です。
- 一時作業ファイルは、標準外の名前 (**SYSUT1.T<time>、**SYSUT2.T<time>、および **SYSUT3.T<time>) を使って割り振られます。このオプションを使用する場合には、これらの名前を、有効なデータ・セット名として RACF 命名規則表に組み込む必要があります。

データ・セットの移動またはコピー

IEHMOVE を使用すると、以下のように、順次データ・セット、区分データ・セット、および BDAM データ・セットを移動またはコピーできます。

- 順次データ・セットの場合

1. 1 つの DASD ボリュームまたは非 DASD ボリュームから別のボリュームに (またはそれが DASD ボリュームであれば同じボリュームに) 移動できます。
 2. 1 つのボリュームから別のボリュームにコピーできます (または受け取り側ボリュームが DASD ボリュームであれば、データ・セット名を変更して同じボリュームにコピーできます)。
- 区分データ・セットの場合
 1. 1 つの DASD ボリュームから別の DASD ボリューム (または同じボリューム) に移動できます。
 2. 1 つの DASD ボリュームから別の DASD ボリュームに (またはデータ・セット名を変更して同じボリュームに) コピーできます。
 - 受け取り側の装置タイプが同じタイプまたはそれより大きく、レコード・サイズが 32 キロバイトを超えない限り、BDAM データ・セットをある DASD ボリュームから別の DASD ボリュームに移動またはコピーできます。

順次データ・セット

表 47 は、順次データ・セットの基本およびオプションの移動操作とコピー操作を示しています。

表 47. 順次データ・セットの移動およびコピー

操作	基本処理	オプションの処理
順次の移動	データ・セットを移動します。DASD の場合、ソース・データをスクラッチします。非カタログ式データ・セットの場合は、移動先データ・セットを参照するために、適切なカタログを更新します。	移動先のデータ・セットを自動的にカタログ化しません。移動先のデータ・セットを名前変更します。
順次のコピー	データ・セットをコピーします。ソース・データ・セットはスクラッチされません。コピー先データ・セットを参照するための、カタログ更新は行われません。	ソース・データ・セットのカタログ項目を削除します。受け取り側ボリュームにコピーされたデータ・セットをカタログ化します。コピー先のデータ・セットを名前変更します。

直接アクセス装置にある順次データ・セットを移動またはコピーする場合、入出力に複数の BSAM バッファを使うことにより、IEHMOVE 実行時間を減らせます。IEHMOVE によって使われるバッファの数を指定する方法については、308 ページの『EXEC ステートメント』を参照してください。

区分データ・セット

300 ページの表 48 は、区分データ・セットの基本およびオプションの移動操作とコピー操作を示しています。

表 48. 区分データ・セットの移動およびコピー

操作	基本処理	オプションの処理
区分の移動	データ・セットを移動します。ソース・データをスクラッチします。カタログ式データ・セットの場合は、移動先データ・セットを参照するために、適切なカタログを更新します。	移動先のデータ・セットを自動的にカタログ化しません。移動先のデータ・セットを名前変更します。移動中に IEHMOVE によってディレクトリー・スペースが割り振られた場合、ディレクトリー・スペースを再割り振りします。複数のデータ・セットをマージします。選択したメンバーだけを移動します。メンバーを置換します。データ・セットをアンロードします。
区分のコピー	データ・セットをコピーします。ソース・データはスクラッチされません。コピー先データ・セットを参照するための、カタログ更新は行われません。	ソース・データ・セットのカタログ項目を削除します。コピー先のデータ・セットをカタログします。コピー先のデータ・セットを名前変更します。事前割り振りしたスペースの一部が使われていなければ、ディレクトリー・スペースを再割り振りします。複数のデータ・セットをマージします。選択したメンバーだけをコピーします。メンバーを置換します。データ・セットをアンロードします。

301 ページの図 41 は、コピー先の区分データ・セットを示しています。

IEHMOVE は、区分メンバーが区分ディレクトリーに現れる順番に、区分メンバーを移動またはコピーします。つまり、移動またはコピーしたメンバーは、照合シーケンスで受け取り側ボリュームに収められます。IEBCOPY ユーティリティ・プログラム (23 ページの『第 3 章 IEBCOPY (ライブラリーのコピー) プログラム』を参照) を使用すると、メンバーが照合されないデータ・セットをコピーできます。

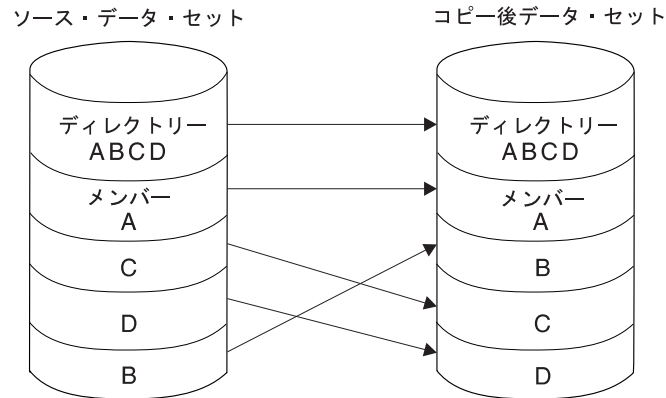


図 41. IEHMOVE コピー操作の前後の区分データ・セット

既存のデータ・セットにマージされたメンバーは、既存のデータ・セットの最後のメンバーの後に、照合順に置かれます。移動されるデータ・セットのメンバーと同じ名前のメンバーがターゲット・データ・セットに含まれる場合、REPLACE ステートメントをコーディングしない限り、そのメンバーは移動またはコピーされません。

図 42 は、データ・セットのメンバーを既存のデータ・セットにマージする様子を示しています。メンバー B および F は照合シーケンスでコピーされます。

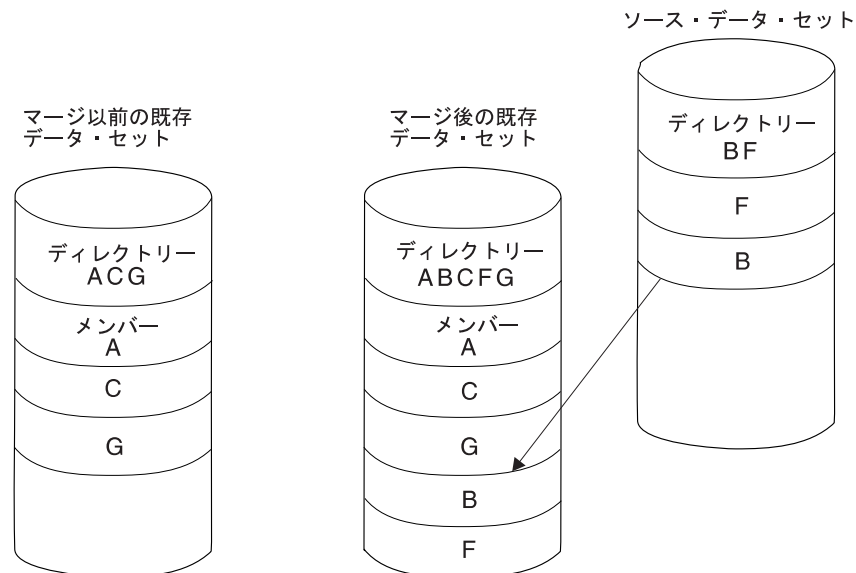
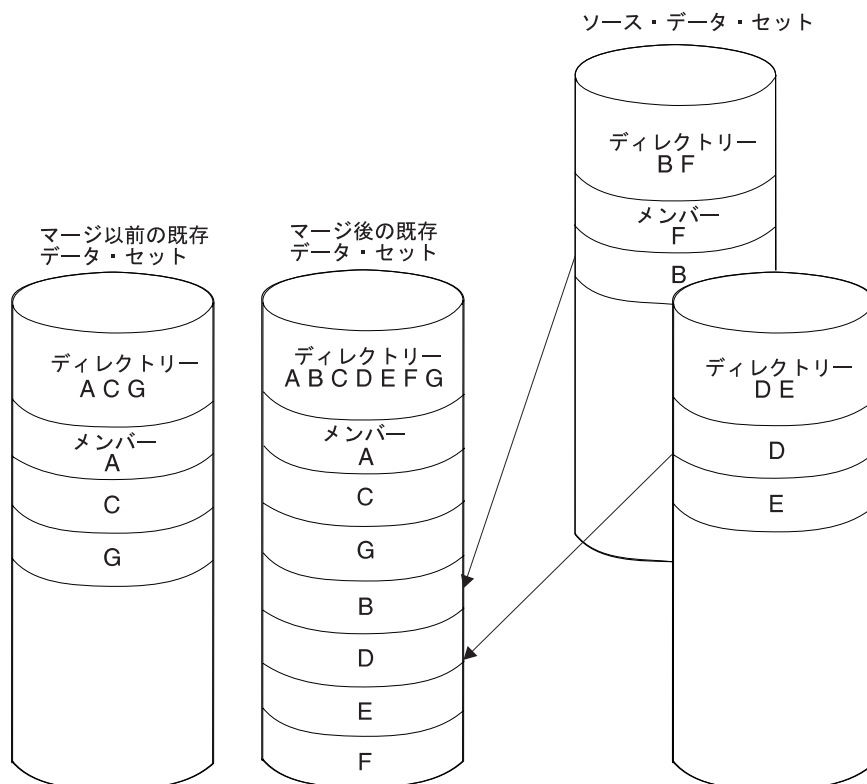


図 42. IEHMOVE を使った 2 つのデータ・セットのマージ

302 ページの図 43 は、2 つのデータ・セットのメンバーが既存のデータ・セットにマージされる様子を示しています。このほかのデータ・セットのメンバーも、これと同様にマージできます。ソース・データ・セットのメンバー F、B、D および E は、照合シーケンスでコピーされます。



注：新しいメンバーは、既存メンバーの後に照合シーケンスで置かれる。

図 43. IEHMOVE を使った 3 つのデータ・セットのマージ

BDAM データ・セット

1 つの装置から同じタイプの別の装置に BDAM データ・セットを移動またはコピーする場合、相対トラックおよび相対ブロックの整合性が保たれます。

より大きい装置に BDAM データ・セットを移動またはコピーする場合、可変長のレコード・フォーマットおよび不定形式のレコード・フォーマットを含むデータ・セットについては、相対トラックの整合性が保たれます。また、固定長のレコード・フォーマットを含むデータ・セットについては、相対ブロックの整合性が保たれます。

より小さい装置またはテープに BDAM データ・セットを移動またはコピーする場合、データ・セットはアンロードされます。アンロードされたデータ・セットは、アンロードされる前と同じ装置タイプに移動またはコピーされる場合に限ってロードできます。

レコード・サイズが 32 キロバイトを超えない限り、1 つの DASD ボリュームから互換性のある別の DASD ボリュームに BDAM データ・セットをコピーできます。(ボリュームの互換性については、294 ページの『ボリューム・サイズの互換性についての考慮事項』を参照してください。)

BDAM データ・セットは 1～5 個のボリュームに分けて常駐することができるため (移動またはコピー操作中には、そのすべてのボリュームがマウントされなければならない)、データ・セットがボリュームをスパンする可能性があります。ただし、単一の可変長スパン・レコードは 1 つのボリュームに収められます。

スパン・レコードの移動操作およびコピー操作では、相対トラックの整合性が保たれます。移動先またはコピー先の BDRAM データ・セットは、ソース装置で占有していたのと同じ相対番号のトラックを占有します。

BDAM データ・セットがアンロードされた場合 (より小さい装置やテープに移動またはコピーされた場合)、アンロードされる前と同じ装置タイプに再びロードしなければなりません。

より大きい装置に可変長スパン・レコードを移動またはコピーする場合、レコード・セグメントは結合され、必要であれば再スパンされます。残っているトラック・スペースは新しいレコードに使用できるため、可変長スパン・レコードは、より小さい装置に移動またはコピーされる前にアンロードされます。

データ管理 BDMACRO を使わずに BDMAM データ・セットを作成する場合には、すべてのデータ管理仕様に従う必要があります。R0 トラック容量レコードの内容、セグメント記述子の入力内容、および BFTEK=R パラメーターに関するデータ管理仕様については、特に注意が必要です。データ管理仕様の使用方法については、*z/OS DFSMS データ・セットの使用法* を参照してください。

マルチボリューム・データ・セットを移動またはコピーする場合、BDAM データ・セットのための 2 次割り振りには少なくとも 2 つのトラックを割り振る必要があります。(z/OS DFSMS Macro Instructions for Data Sets の『WRITE』マクロを参照。)

マルチボリューム・データ・セット

IEHMOVE を使用して、マルチボリューム・データ・セットを移動またはコピーできます。マルチボリューム・データ・セットを移動またはコピーするには、DD ステートメントの VOL=SER パラメーターで完全なボリューム・リストを指定します。ボリュームは最大 5 つまで指定できます。入力データ・セットに既にあるボリュームと同数のボリュームを、出力データ・セットにも指定する必要があります。より少ないボリュームにデータ・セットを収めるためにマルチボリューム・データ・セットを統合するには、データ・セットを移動またはコピーする前に、ユーザーが出力データ・セットをターゲット・ボリュームに割り振らなければなりません。複数のテープ・ボリュームに収められているデータ・セットを移動またはコピーするには、ユーティリティー制御ステートメントで、すべてのテープ・ボリュームのボリューム通し番号、およびテープ・ボリューム上のデータ・セットのシーケンス番号を指定します。(移動またはコピーするデータ・セットがそのボリュームの唯一のデータ・セットであっても、シーケンス番号を指定できます。) 複数のテープ・ボリュームにデータ・セットを移動またはコピーするには、ユーティリティー制御ステートメントで、すべての受け取り側ボリュームのボリューム通し番号を指定します。

アンロードされたデータ・セット

指定したデータを正常に移動またはコピーできない場合、IEHMOVE はそのデータの再編成、および指定した出力装置へ入れようと試みます。再編成されたデータ (アンロードされたデータ・セット という) とは、80 バイトのブロック化レコードからなる順次データ・セットであり、これにはソース・データを最初と同じ状態に再構築するのに必要なソース・データおよび制御情報が含まれています。

アンロードされたデータ・セットが、そのデータの実際のフォームをサポートする装置に (IEHMOVE を介して) 移動またはコピーされる場合、データは自動的に再構築されます。たとえば、区分データ・セットをテープ・ボリュームに移動しようとする、データはそのボリュームにアンロードされます。アンロードされたデータ・セットを DASD ボリュームに移動するだけで、データ・セットを再作成できます。

移動不能データ・セット

移動不能属性を持つデータ・セットは、受け取り側ボリュームにスペースが事前に割り振られていれば、1 つの DASD ボリュームから別のボリュームに (または同じボリュームに) 移動またはコピーできます。同じボリュームに移動またはコピーする場合には、データ・セットの名前を変更してください。

カタログ式データ・セットのグループの移動またはコピー

IEHMOVE を使用すれば、統合カタログ機能でカタログ化され、1 つまたは複数の同一名で名前が修飾されている区分データ・セット、順次データ・セット、または BDAM データ・セットのグループ (『DSGROUP』) を移動またはコピーできます。たとえば、A.B という名前で修飾されているデータ・セットのグループに、データ・セット名 A.B.D および A.B.E を含めることはできますが、データ・セット名 A.C.D や A.D.F を含めることはできません。

ストレージ管理サブシステムによって管理されているボリュームへは、IEHMOVE を使用して DSGROUP を移動またはコピーすることができません。

データ・セットのグループを磁気テープに移動またはコピーする場合には、データ・セット名およびファイル・シーケンス番号によって、または (ラベルの付いていないテープや標準外ラベルの付いたテープの場合は) ファイル・シーケンス番号によって、データ・セットを 1 つずつ検索する必要があります。

アクセス方式サービスを使って、統合カタログ機能カタログの構造を判別することができます。詳細については、*z/OS DFSMS カatalogのためのアクセス方式サービス・プログラム* を参照してください。

305 ページの表 49 は、区分カタログ式データ・セット、順次カタログ式データ・セット、または BDAM カatalog式データ・セットの基本およびオプションの移動操作とコピー操作を示したものです。

表 49. カタログ式データ・セットのグループの移動およびコピー

操作	基本処理	オプションの処理
カタログ式データ・セットのグループの移動	データ・セット・グループ (パスワードで保護されたデータ・セットを除く) を指定したボリュームに移動します。ソース・データ・セットをスクラッチします (BDAM の場合のみ)。マージは行われません。	所定のカタログの更新が行われないようにします。パスワードで保護されたデータ・セットを操作に含めます。データ・セットをアンロードします。データ・セット・グループがカタログ化されている場合、操作中にほかのデータ・セットを含めたり、除外したりすることが可能です。
カタログ式データ・セットのグループのコピー	データ・セット・グループ (パスワードで保護されたデータ・セットを除く) をコピーします。ソース・データ・セットはスクラッチされません。マージは行われません。	パスワードで保護されたデータ・セットを操作に含めます。ソース・データ・セットのカタログ項目を削除します。受け取り側ボリュームにコピーされたデータ・セットをカタログします。1 つまたは複数のデータ・セットをアンロードします。データ・セット・グループがカタログ化されている場合、操作中にほかのデータ・セットを含めたり、除外したりすることが可能です。

データ・セットのボリュームの移動またはコピー

IEHMOVE を使用して、DASD ボリューム全体のデータ・セットを別のボリュームに移動またはコピーできます。移動操作ではソース・データ・セットがスクラッチされるのに対して、コピー操作ではソース・データ・セットはスクラッチされません。どちらの操作の場合も、ソース・データ・セットに関連したすべてのカタログ化項目は変更されません。

ソース・ボリュームに SYSCTLG データ・セットが含まれる場合、それは受け取り側ボリュームに移動またはコピーされる最後のデータ・セットです。

データ・セットのボリュームをテープに移動またはコピーする場合、順次データ・セットは移動されます。区分データ・セットおよび BDAM データ・セットはアンロードされます。データ・セット名およびファイル・シーケンス番号によって、または (ラベルの付いていないテープや標準外ラベルの付いたテープの場合は) ファイル・シーケンス番号によって、データ・セットを 1 つずつ検索する必要があります。

データ・セットのボリュームをコピーする場合、オプションで、受け取り側ボリュームの SYSCTLG データ・セット内にあるすべてのソース・データ・セットをカタログ化できます。ただし、SYSCTLG データ・セットがソース・ボリューム上にあ

る場合、ソース SYSCTLG 項目が受け取り側ボリュームの SYSCTLG データ・セットにマージされる時に、矛盾する索引構造が存在している旨を示すエラー・メッセージが生成されます。

ボリューム移動機能によって区分データ・セットはマージされません。移動元のボリューム上の任意のデータ・セット名と同じデータ・セット名が受け取り側ボリュームにある場合、そのデータ・セットは受け取り側ボリュームに移動またはマージされません。

ボリューム・コピー機能によって、区分データ・セットはマージされます。コピーされるボリューム上の任意のデータ・セット名が受け取り側ボリュームのデータ・セット名と同じである場合、そのデータ・セットは受け取り側ボリュームにコピーされてマージされます。

表 50 は、データ・セットのボリュームに対する、基本およびオプションの移動操作とコピー操作を示したものです。

表 50. データ・セットのボリュームの移動およびコピー

操作	基本処理	オプションの処理
データ・セットのボリュームの移動	パスワードで保護されていないすべてのデータ・セットを、指定した DASD ボリュームに移動します。DASD ボリュームのソース・データ・セットをスクラッチします。	パスワードで保護されたデータ・セットを操作に含めません。データ・セットをアンロードします。
データ・セットのボリュームのコピー	パスワードで保護されていないすべてのデータ・セットを、指定した DASD ボリュームにコピーします。ソース・データ・セットはスクラッチされません。	パスワードで保護されたデータ・セットを操作に含めません。すべてのデータ・セットをカタログ化します。データ・セットをアンロードしません。

入出力

IEHMOVE は以下の入力を使用します。

- 1 つまたは複数の区分データ・セット、順次データ・セット、または BDAM データ・セット。これらは出力データ・セットに移動、コピーまたはマージされるデータを含んでいます。
- 制御データ・セット。これに含まれるユーティリティー制御ステートメントを使って、プログラムの機能を制御します。
- 作業データ・セット。これは IEHMOVE によって使われる作業域です。

IEHMOVE は、VIO (仮想入出力) データ・セットをサポートしません。

IEHMOVE は以下の出力を生成します。

- 出力データ・セット。これは移動、コピーまたはマージ操作の結果です。

- メッセージ・データ・セット。これには、通知メッセージ (たとえば、移動またはコピーされたデータ・セット名) およびエラー・メッセージ (該当する場合) が含まれます。

アプリケーション・プログラムから IEHMOVE を起動する場合は、IEHMOVE を呼び出す前に SVC 99 を出すことによって、装置やデータ・セットを動的に割り振ることができます。

IEHMOVE の戻りコードについては、付録 Aを参照してください。

制御

IEHMOVE は、ジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。ジョブ制御ステートメントを使用して、プログラムを処理またはロードしたり、IEHMOVE が使用および生成する装置やボリュームを定義したり、データ・セットの誤った削除を防止したりします。

IEHMOVE は APF 許可プログラムです。つまり、他のプログラムがそれを呼び出すとき、そのプログラムも APF 許可でなければなりません。システムの保全性を保護するために、呼び出し側プログラムは *z/OS MVS プログラミング: アセンブラー・サービス ガイド* に記載されているシステム保全性の要件に従っていなければなりません。

ユーティリティー制御ステートメントは、プログラムの機能を制御したり、使用されるデータ・セットおよびボリュームを定義したりするのに使います。

ジョブ制御ステートメント

表 51 は、IEHMOVE で使用するジョブ制御ステートメントをまとめたものです。

表 51. IEHMOVE で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IEHMOVE)。このジョブ制御ステートメントがプロシージャ・ライブラリー内にある場合は、プロシージャ名を指定します。このステートメントには、オプションのパラメーター情報を含めることもできます。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。データ・セットは、システム出力装置、磁気テープ・ボリューム、または直接アクセス・ボリュームに書き込むことができます。
SYSUT1 DD	IEHMOVE に必要な 3 つの作業データ・セットが割り振られるボリュームを定義します。
anyname (任意名) DD	永続的に取り付けられた DASD ボリュームまたは永続的に取り付け可能な DASD ボリュームを定義します。永続的に取り付けられたボリュームが少なくとも 1 つ、識別されなければなりません。
tape DD	マウント可能な磁気テープ装置を定義します。

表 51. IEHMOVE で使用するジョブ制御ステートメント (続き)

ステートメント	用途
SYSIN DD	制御データ・セットを定義します。ユーティリティー制御ステートメントを含んでいるこのデータ・セットは通常、入力ストリーム内で一連のジョブ制御ステートメントの後に配置します。しかし、これを順次データ・セットまたはプロシージャ・ライブラリーのメンバーのいずれかとして定義することもできます。

SYSUT2 および SYSUT3 は IEHMOVE 用に予約済みです。これらを IEHMOVE ジョブ・ステップで使用しないようにしてください。

EXEC ステートメント

IEHMOVE の EXEC ステートメントには、追加のワークスペースの割り振りや出力リストの行間幅の制御のためにプログラムが使用するパラメーター情報を含めることができます。また、REGION サブパラメーターをコーディングすれば、順次データ・セットの移動またはコピー時に IEHMOVE が操作を行う領域のサイズを制御できます。

EXEC ステートメントの構文は次のとおりです。

//[stepname]	EXEC	PGM=IEHMOVE [.PARAM='[POWER= <i>n</i>] [,LINECNT= <i>xx</i> '] [.REGION={ <i>n</i> K <i>n</i> M}]
--------------	------	----------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

PGM=IEHMOVE

IEHMOVE を実行することを指定します。

PARAM='[POWER=*n*] [,LINECNT=*xx*']

IEHMOVE に渡されるオプション・パラメーター情報を指定します。

POWER=*n*

IEHMOVE の使用する作業域に割り振られるスペースを *n* 倍に増やすことを指定します。 *n* の範囲は 1~999 です。大きな区分データ・セットを移動またはコピーする際には、このパラメーターを使用してください。

区分データ・セットのメンバーが 750 個を上回る場合、POWER をコーディングする必要があります。 *n* の値は、それぞれ以下のように指定します。

- POWER=2 (移動またはコピーされるメンバーが 750~1500 個の場合)。
- POWER=3 (移動またはコピーされるメンバーが 1501~2250 個の場合)。
- POWER=4 (移動またはコピーされるメンバーが 2251~3000 個の場合)。

たとえば、POWER=2 の場合、それぞれ連続した 26 個、26 個、および 52 個のトラックからなる 3 つの領域を 3380 上で使用可能にしなければなりません。

LINECNT=*xx*

SYSPRINT データ・セットのリストに印刷されるページ当たりの行数を指定します。 *xx* には 04~99 の 2 桁の数を指定できます。

PARM キーワード値のコーディングの詳細については、*z/OS MVS JCL 解説書*を参照してください。

REGION={nK|nM}

順次 データ・セットを移動またはコピーする際に IEHMOVE が稼働する領域のサイズを指定します。このパラメーターを使うと IEHMOVE のパフォーマンスを改善できますが、これは順次データ・セットを移動またはコピーする際の必須パラメーターではありません。

IEHMOVE コピー操作のパフォーマンス向上のために必要な最小限のバッファ一数は 4 つです (入力用に 2 つ、出力用に 2 つ)。それぞれの入力バッファのサイズは、以下のように計算されます。

$$(\text{INPUT BLOCKSIZE} + \text{KEY LENGTH}) + 24$$

それぞれの出力バッファのサイズは、以下のように計算されます。

$$(\text{OUTPUT BLOCKSIZE} + \text{KEY LENGTH}) + 40$$

IEHMOVE によって使われる入力バッファの最大数は、入力トラック・サイズに入るバッファの数の 2 倍です。IEHMOVE によって使われる出力バッファの最大数は、出力トラック・サイズに入るバッファの数の 2 倍です。

最小限の 4 つのバッファのためのスペースが使用できない場合、バッファが 1 つだけ使用され、メッセージ IEH476I が出されます。

なお、このパラメーターは、EXEC ステートメントではなく JOB ステートメント内でコーディングすることもできます。REGION パラメーターをコーディングするための詳細については、*z/OS MVS JCL 解説書*を参照してください。

複数の BSAM バッファが使用されるたびに、バッファの数およびサイズを記述したメッセージ IEH477I が出されます。バッファの最大数に達するだけの十分な領域サイズが指定されていない場合、このメッセージの最後の行には、最大数のバッファを得るために増やさなければならない REGION パラメーターの値が示されます。

IEHMOVE の移動操作またはコピー操作の実行時間は、使用可能なバッファの数、データ・セットのサイズ、およびブロック・サイズによって変わります。

SYSPRINT DD ステートメント

SYSPRINT データ・セットのブロック・サイズは、121 の倍数でなければなりません。任意のブロック化因数を指定できます。

SYSUT1 DD ステートメント

SYSUT1 DD ステートメントは、IEHMOVE が作業域として使用する DASD ボリュームを定義します。SYSUT1 DD ステートメントは、以下のようにコーディングする必要があります。

```
//SYSUT1 DD UNIT=xxxx,VOL=SER=xxxxxx,DISP=OLD
```

UNIT パラメーターと VOLUME パラメーターによって、装置タイプとボリューム通し番号を定義します。DISP=OLD の指定により、データ・セットを誤って削除することがないようにしています。SYSUT1 DD ステートメントでは、SMS 管理ボリュームは定義できません。

SYSUT1 DD ステートメントで定義されたボリューム上では、ワークスペースとして、それぞれ 13 個、13 個、および 26 個の連続したトラックからなる少なくとも 3 つのユーティリティ作業域が使用可能でなければなりません。(この数字は、作業ボリュームが 3380 であると想定した場合です。3380 以外の直接アクセス装置をご使用の場合、それに相当するスペースを使用可能にしなければなりません。)

IEHMOVE は、作業域に必要なスペースの量を自動的に計算して割り振ります。このため、SYSUT1 DD ステートメントでは SPACE パラメーターをコーディングしてはなりません。その代わりに、EXEC ステートメントの PARM パラメーターの POWER 値をコードすることによって、作業域を大きくすることができます。

IEHMOVE を実行する前に、IEHPROGM ユーティリティ・プログラムのダミーの実行を指定することによって、事前にスペースを割り振ることができます。

注: IEHMOVE は標準外のデータ・セット名を使って、作業データ・セットを割り振ります。名前は 1 つ以上のアスタリスクで始まります。これらの作業データ・セットは、要求された機能の完了時に削除されます。

ただし、IEHMOVE が正常に終了しない場合 (異常終了、システム誤動作など)、これらの作業データ・セットは DASD ボリューム上に残り、いずれの IBM ユーティリティを使ってでも削除できません。それらを削除するには、DISP=(OLD=DELETE) を指定したそれらのデータ・セット名を単一引用符で囲んで、IEFBR14 ジョブを実行すれば可能です。

anyname (任意名) DD ステートメント

永続的に取り付けられているボリュームまたは永続的に取り付け可能なボリュームのうち、ジョブ・ステップの中で参照されているものについては、それぞれ 1 つずつ DD ステートメントを含める必要があります。これらの DD ステートメントは、装置の割り振りに使われます。

DD ステートメントは、以下のようにコーディングする必要があります。

```
//anyname1 DD UNIT=xxxx,VOL=SER=xxxxxx,DISP=OLD
```

UNIT パラメーターと VOLUME パラメーターによって、装置タイプとボリューム通し番号を定義します。DISP=OLD を指定することによって、データ・セットが誤って作成されないようにしています。

また、名前を付けたデータ・セットがそのボリューム上にあれば、DSN パラメーターをコーディングしてボリュームを識別することができます。1 つの DASD ボリュームから別のボリュームにデータ・セットをアンロードする際、データ・セットがアンロードされるためには、このパラメーターが必須です。DASD ボリューム上のアンロードされたデータ・セットは、アンロードされる前と同じ装置タイプのみロードできます。

取り付け可能なボリュームについては、処理されるボリュームの数が DD ステートメントで定義された装置の数よりも多い場合には、複数のボリュームを処理する旨の指示を、適切な DD ステートメントに含める必要があります。この指示は、以下のような据え置き取り付けのフォームにすることができます。

```
//anyname2 DD UNIT=(xxxx,,DEFER),VOL=(PRIVATE,...),
// DISP=(...,KEEP)
```

ここで、VOL パラメーターの PRIVATE 指示はオプションです。取り付け可能装置を定義する DD ステートメントで、装置類似性を使用することはできません。

tape DD ステートメント

tape DD ステートメントは、以下のようにコーディングできます。

```
//tape DD DSN=xxxxxxx,UNIT=xxxx,VOLUME=SER=xxxxxx,
//        DISP=(...,KEEP),LABEL=(...,...),DCB=(TRTCH=C,DEN=x)
```

1 つの DASD ボリュームから別のボリュームにデータ・セットをアンロードする際、データ・セットがアンロードされるためには、DD ステートメントのデータ・セット名 (DSN=) のコードは必須です。DASD ボリューム上のアンロードされたデータ・セットは、アンロードされる前と同じ装置タイプのみロードできます。

ユーティリティー制御ステートメントのパラメーターは、ラベル情報およびモード情報を得るために tape DD ステートメントを参照します。

TODD パラメーターがユーティリティー制御ステートメントで指定されていれば、データ・セットが磁気テープ・ボリュームに移動またはコピーされた日付は、標準テープ・ラベルの HDR1 レコードに自動的に記録されます。有効期限を指定するには、TODD パラメーターが参照する DD ステートメント内で、LABEL キーワードの EXPDT サブパラメーターまたは RETPD サブパラメーターを含めます。

テープ・ボリューム上にあるデータ・セットのシーケンス番号、または特定の装置番号 (たとえば、装置アドレス 190) は、DD ステートメントへの参照ではなく、ユーティリティー制御ステートメントで指定する必要があります。複数のデータ・セットを含むテープ・ボリュームから (あるいは、複数のデータ・セットを含むテープ・ボリュームへ) データ・セットを移動またはコピーするには、ユーティリティー制御ステートメントの中でデータ・セットのシーケンス番号を指定してください。特定の装置から、または特定の装置へデータ・セットを移動またはコピーするには、ユーティリティー制御ステートメントの中で、グループ名や装置タイプではなく、装置アドレスを指定します。ユニット・レコードまたはラベルなしテープ・ボリュームにコピーするには、ユーティリティー制御ステートメントでいずれかの標準の名前または番号を指定します。

tape DD ステートメントを使用して、標準のラベルが付いていないテープ・ボリュームに入っているデータ・セットの DCB 属性を IEHMOVE に伝えることができます。DCB 属性が指定されない場合、不定形式レコードおよびブロック・サイズ 2560 が想定されます。ただし、ラベルなしテープ・ボリューム上のアンロードされたデータ・セットを認識するには、DCB 属性を以下のように指定する必要があります。

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=800).
```

いかなる場合でも、ブロック・サイズは 32760 バイトを超えてはなりません。

SYSIN DD ステートメント

SYSIN データ・セットのブロック・サイズは、80 の倍数でなければなりません。ブロック化因数に指定できる最大ブロック・サイズは 32760 です。

ユーティリティー制御ステートメント

IEHMOVE は表 52 のユーティリティー制御ステートメントによって制御されます。

表 52. IEHMOVE で使用するユーティリティー制御ステートメント

ステートメント	用途
MOVE DSNAME	データ・セットを移動します。
COPY DSNAME	データ・セットをコピーします。
MOVE DSGROUP	区分カタログ式データ・セット、順次カタログ式データ・セット、または BDAM カタログ式データ・セットのグループを移動します。
COPY DSGROUP	区分カタログ式データ・セット、順次カタログ式データ・セット、または BDAM カタログ式データ・セットのグループをコピーします。
MOVE PDS	区分データ・セットを移動します。
COPY PDS	区分データ・セットをコピーします。
MOVE VOLUME	データ・セットのボリュームを移動します。
COPY VOLUME	データ・セットのボリュームをコピーします。

これに加えて、MOVE DSGROUP、COPY DSGROUP、MOVE PDS、COPY PDS、MOVE CATALOG、または COPY CATALOG 操作の効果を変更するために使用できる副次的な制御ステートメントが 4 つあります。副次的な制御ステートメントは、以下のとおりです。

- **INCLUDE** ステートメント。変更するステートメントには、明示的に指定されていないメンバーまたはデータ・セットを含めることによって、MOVE PDS、または COPY PDS ステートメントのスコープを拡張します。
- **EXCLUDE** ステートメント。MOVE PDS、COPY PDS ステートメントとともに使用すれば、移動またはコピー操作からメンバーが除外されます。
- **REPLACE** ステートメント。これは、移動またはコピー操作からメンバーを除外して、別の区分データ・セットからのメンバーと置き換えるために、MOVE PDS または COPY PDS ステートメントとともに使用します。
- **SELECT** ステートメント。これは、移動またはコピーするメンバーを選択したり、指定されたメンバーの名前を変更したりするために、MOVE PDS または COPY PDS ステートメントとともに使用します。

ユーティリティー制御ステートメントの継続のための要件については、9 ページの『ユーティリティー制御ステートメントの継続』を参照してください。

MOVE DSNAME および COPY DSNAME ステートメント

MOVE DSNAME ステートメントを使用して、データ・セットを移動します。ソース・データ・セットはスクラッチされます。データ・セットがカタログ化されている場合、UNCATLG と FROM が指定されていなければ、カタログは自動的に更新されます。

COPY DSNAME ステートメントを使用して、データ・セットをコピーします。ソース・データ・セットがカタログ化されている場合、RENAME パラメーターと FROM パラメーターを使わずに UNCATLG または CATLG を指定した場合を除く

て、ソース・データ・セットはカタログ化されたままになります。コピーでは、ソース・データ・セットはスクラッチされません。

MOVE DSNAME および COPY DSNAME ステートメントの構文は次のとおりです。

[label]	{MOVE COPY}	DSNAME=name ,TO=device={serial(list)} [,FROM=device={serial (list)}] [,UNCATLG] [,CATLG] [,RENAME=name] [,FROMDD=ddname] [,TODD=ddname] [,UNLOAD] [,COPYAUTH]
注: CATLG は COPY DSNAME でのみコーディングできます。		

各項目の意味は以下のとおりです。

DSNAME=name

移動元またはコピー元のデータ・セットの完全修飾名を指定します。

TO=device={serial (list)}

指定されたデータ・セットの移動先またはコピー先ボリュームの装置タイプとボリューム通し番号を指定します。データ・セットが複数のボリュームに収められている場合、括弧内にボリューム通し番号のリストをコーディングします (コンマで各番号を区切る)。

FROM=device={serial (list)}

データ・セットが収められているボリュームの装置番号または装置タイプと、ボリューム通し番号を指定します。データ・セットが複数のボリュームに収められている場合、通し番号のリストを括弧で囲みます (コンマで各番号を区切る)。

データ・セットがカタログ化されている場合は、『FROM』をコーディングしないでください。

装置タイプではなく特定の装置を指定する場合は、*device* サブパラメーターにその装置番号をコーディングしてください。

FROM とともに MOVE DSNAME を使用する場合、カタログは更新されません。

データ・セットが磁気テープ装置に収められている場合、『(通し番号、シーケンス番号)』という形式で、通し番号とデータ・セット・シーケンス番号を括弧で囲んで指定する必要があります。

FROM を省略した場合は、データ・セットが統合カタログ機能カタログ /JOB CAT/STEP CAT カタログ内でカタログ化されていると想定します。

UNCATLG

ソース・データ・セットに関連したカタログ項目を削除するよう指定します。このパラメーターは、ソース・データ・セットがカタログ化されている場合にのみ使用してください。ボリュームが FROM によって識別されていれば、UNCATLG は無視されます。ソース・データ・セットのカタログ内の別名項目

は失われます。データ・セットを後でカタログ化する場合には、別名項目は、アクセス方式サービスを使って置き換えることができます。詳細については、*z/OS DFSMS カタログのためのアクセス方式サービス・プログラム* の DELETE コマンドについての説明を参照してください。MOVE 操作の場合、UNCATLG は出力データ・セットがカタログ化されないようにします。

CATLG

コピー先のデータ・セットを以下のようにカタログ化するよう指定します。

1. 統合カタログ機能または /JOB/CAT/STEP/CAT カタログ内でカタログ化が行なわれます。
2. RENAME および FROM パラメーターの両方を省略した場合、ソース・データ・セットの項目を適切なカタログから削除することによって、コピー先データ・セットの再カタログ化が可能になります。

適切な結果を出すためには、DD ステートメントの『DISP』パラメーターで『CATLG』オプションを指定する代わりに、この制御ステートメント・オプションを使用する必要があります。

CATLG は COPY DSNAMES でのみコーディングできます。

RENAME=name

データ・セットを名前変更するよう指定して、新しい名前を指示します。

FROMDD=ddname

テープ・ボリュームに収められている入力データ・セットのための、DCB 情報や (データ・セット・シーケンス番号を除く) LABEL 情報を入手できる DD ステートメント名を指定します。データ・セットに標準ラベルが付いていて、9トラックからなる1つのテープ・ボリュームに収められている場合、この FROMDD オペランドを省略できます。

TODD=ddname

テープ・ボリュームに収められる出力データ・セットのための、DCB 情報 (RECFM、BLKSIZE および LRECL を除く)、および LABEL 情報 (データ・セット・シーケンス番号を除く) を入手できる DD ステートメント名を指定します。

この DD ステートメントは、テープ・ボリューム上に出力データ・セットを作成するときに使用するモード情報およびラベル情報を記述しています。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報がコーディングされていれば、それらは無視されます。

UNLOAD が指定されている場合、この DD ステートメントは、データ・セットをアンロードするときに使用するモード情報およびラベル情報を記述します。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報をコードする場合、(RECFM=FB, BLKSIZE=800, LRECL=80) と指定しなければなりません。

有効期限データ (EXPDT) や保存期間 (RETPD) を作成または変更するときは、制御ステートメント内で TODD を指定する必要があります。

標準ラベルの付いた、指定した装置タイプのデフォルト密度になっている9トラックのテープの場合は、TODD パラメーターを省略することができます。

UNLOAD

データ・セットを受け取り側ボリュームにアンロードするよう指定します。

COPYAUTH

入力データ・セットが RACF で保護されており、しかも出力データ・セットが事前に割り振られていない場合、入力データ・セットと同じアクセス・リストを受け取り側データ・セットに提供するように指定します。

MOVE DSGROUP および COPY DSGROUP ステートメント

MOVE DSGROUP を使用して、1 つまたは複数の同一名によって名前が部分修飾されているデータ・セットのグループを移動します。これらのデータ・セットは、複数のカタログ上でカタログ化される場合があります。ソース・データ・セットはスクラッチされます。移動するデータ・セット・グループは、DASD ボリュームに納められていなければなりません。MOVE DSGROUP によって移動できるのは、MOVE DSNAMES または MOVE PDS によって移動できるデータ・セットだけです。データ・セットのカタログ内の別名項目は失われ、別名項目は、アクセス方式サービスを使って置き換えることができます。詳細については、*z/OS DFSMS カタログのためのアクセス方式サービス・プログラム* を参照してください。

COPY DSGROUP を使用して、名前が 1 つまたは複数の同一名によって部分修飾されているデータ・セットのグループをコピーします。これらのデータ・セットは、複数のカタログ上でカタログ化される場合があります。COPY DSGROUP を使用してコピーできるのは、COPY DSNAMES または COPY PDS によってコピーできるデータ・セットだけです。コピー元のデータ・セット・グループは、DASD ボリュームに納められていなければなりません。

UNCATLG が指定されていない限り、MOVE DSGROUP 操作によってカタログは自動的に更新されます。RENAME パラメーターと FROM パラメーターを使わずに UNCATLG または CATLG を指定した場合を除いて、COPY DSGROUP 操作はソース・データ・セットのカタログ化された状態を保ちます。

COPY DSGROUP ステートメントの構文は次のとおりです。

[label]	{MOVE} COPY}	DSGROUP=[name] ,TO=device={serial(list)} [,PASSWORD] [,UNCATLG] [,CATLG] [,TODD=ddname] [,UNLOAD] [,COPYAUTH]
注: CATLG は COPY DSGROUP でのみコーディングできます。		

各項目の意味は以下のとおりです。

DSGROUP=[name]

移動元またはコピー元のカタログ式データ・セットを指定します。name に完全修飾データ・セット名を指定すると、そのデータ・セットだけを移動またはコピーします。name が 1 つまたは複数の修飾子であるものの、完全修飾名ではない場合、name によって修飾される名前をもつすべてのデータ・セットを移動またはコピーします。name を省略した場合、カタログ内を検索して見つかったすべての名前のデータ・セットを移動またはコピーします。

TO=device={serial(list)}

指定されたデータ・セット・グループの、移動先またはコピー先ボリュームの装置タイプとボリューム通し番号を指定します。データ・セット・グループが複数のボリュームに収められている場合、括弧内に通し番号のリストをコーディングします (コンマで各番号を区切る)。

PASSWORD

パスワードで保護されたデータ・セットを操作に含めるよう指定します。これは VSAM パスワード保護ではなく、データ・セット・パスワード体系です。詳細については、*z/OS DFSMSdfp 拡張サービス* に記載されています。PASSWORD を省略した場合、保護されていないデータ・セットのみがコピーまたは移動されます。

UNCATLG

ソース・データ・セットに関連したカタログ項目を削除するよう指定します。このパラメーターは、ソース・データ・セットがカタログ化されている場合にのみ使用してください。ボリュームが FROM によって識別されていれば、UNCATLG は無視されます。ソース・データ・セットの統合カタログ機能内の別名項目は失われます。これらのデータ・セットを後でカタログ化する場合には、別名項目は、アクセス方式サービスを使って置き換えることができます。詳細については、*z/OS DFSMS カタログのためのアクセス方式サービス・プログラム* を参照してください。MOVE 操作の場合、UNCATLG は出力データ・セットがカタログ化されないようにします。

CATLG

カタログ化が、統合カタログ機能または /JOB/CAT/STEP/CAT カタログ内で行なわれるように指定します。

適切な結果を出すためには、DD ステートメントの『DISP』パラメーターで『CATLG』オプションを指定する代わりに、この制御ステートメント・オプションを使用する必要があります。

CATLG は COPY DSGROUP でのみコーディングできます。

TODD=ddname

テープ・ボリュームに収められる一連の出力データ・セットのための、DCB 情報 (RECFM、BLKSIZE および LRECL を除く)、および LABEL 情報 (データ・セット・シーケンス番号を除く) を入手できる DD ステートメント名を指定します。

この DD ステートメントは、テープ・ボリューム上に出力データ・セットを作成するとき使用するモードおよびラベルの情報を記述しています。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報がコーディングされていれば、それらは無視されます。

UNLOAD が指定されている場合、この DD ステートメントは、アンロードされたバージョンのデータ・セットを磁気テープ・ボリュームに作成するとき使用するモード情報およびラベル情報を記述します。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報をコードする場合、(RECFM=FB, BLKSIZE=800, LRECL=80) と指定しなければなりません。

有効期限データ (EXPDT) や保存期間 (RETPD) を作成または変更するときは、制御ステートメント内で TODD を指定する必要があります。

標準ラベルの付いた、指定した装置タイプのデフォルト密度になっている 9 トラックのテープの場合は、 TODD パラメーターを省略することができます。

UNLOAD

データ・セットを受け取り側ボリュームにアンロードするよう指定します。

COPYAUTH

入力データ・セットが RACF で保護されており、しかも出力データ・セットが事前に割り振られていない場合、入力データ・セットと同じアクセス・リストを受け取り側データ・セットに提供するよう指定します。

MOVE PDS および COPY PDS ステートメント

MOVE PDS ステートメントを使用して、区分データ・セットを移動します。MOVE PDS ステートメントを INCLUDE、EXCLUDE、REPLACE、または SELECT ステートメントとともに使用すると、いくつかの区分データ・セットの選択したメンバーをマージしたり、削除したりすることができます。ソース・データ・セットはスクラッチされます。

COPY PDS ステートメントを使用して、区分データ・セットをコピーします。COPY PDS ステートメントを INCLUDE、EXCLUDE、REPLACE、または SELECT ステートメントとともに使用すると、いくつかの区分データ・セットの選択したメンバーをマージしたり、削除したりすることができます。

IEHMOVE を使って出力区分データ・セットのためのスペースを割り振る場合、MOVE PDS または COPY PDS ステートメントを使用すると、区分ディレクトリを拡張できます。

受け取り側ボリュームに同じ名前の区分データ・セットがある場合、その 2 つのデータ・セットはマージされます。

UNCATLG および FROM が指定されていなければ、MOVE PDS によって適切なカタログが自動的に更新されます。RENAME パラメーターと FROM パラメーターを使わずに UNCATLG または CATLG を指定した場合を除いて、COPY PDS はソース・データ・セットのカタログ化された状態を保ちます。

MOVE PDS および COPY PDS ステートメントの構文は次のとおりです。

[label]	{MOVE} COPY}	PDS=name ,TO=device={seriallist} [,{FROM=device=serial}] [,{EXPAND=nn}] [,{UNCATLG}] [,{CATLG}] [,{RENAME=name}] [,{FROMDD=ddname}] [,{TODD=ddname}] [,{UNLOAD}] [,{COPYAUTH}]
注: CATLG は COPY PDS でのみコーディングできます。		

各項目の意味は以下のとおりです。

PDS=name

移動元またはコピー元の区分データ・セットの完全修飾名 (つまり修飾子があれば、それら修飾子をすべて含めた名前) を指定します。

TO=device={seriallist}

区分データ・セットの移動先またはコピー先ボリュームの装置タイプとボリューム通し番号を指定します。区分データ・セットを複数のテープ・ボリュームにアロードしている場合、括弧内に通し番号のリストをコーディングします (コマで各番号を区切る)。

FROM=device=serial

データ・セットが収められているボリュームの装置タイプと通し番号を指定します。

データ・セットがカタログ化されている場合は、『FROM』をコーディングしないでください。

装置タイプではなく特定の装置を指定する場合は、*device* サブパラメーターにその装置番号をコーディングしてください。

FROM とともに MOVE PDS を使用する場合、カタログは更新されません。

FROM を省略した場合は、データ・セットが統合カタログ機能カタログまたは /JOB CAT/STEP CAT カタログ内でカタログ化されていると想定します。

EXPAND=nn

指定された区分データ・セットのディレクトリーに付け加える 256 バイト長レコードの数 (10 進数、99 まで) を指定します。COPY の場合は、スペースが前もって割り振られていれば、EXPAND を指定することはできません。

MOVE の場合、スペースが前もって割り振られていれば、EXPAND は無視されます。

UNCATLG

ソースの区分データ・セットに関連したカタログ項目を削除するよう指定します。このパラメーターは、ソース・データ・セットがカタログ化されている場合にのみ使用してください。ボリュームが FROM によって識別されていれば、UNCATLG は無視されます。ソース・データ・セットのカタログ内の別名項目は失われます。データ・セットを後でカタログ化する場合には、別名項目は、アクセス方式サービスを使って置き換えることができます。詳細については、*z/OS DFSMS* カタログのためのアクセス方式サービス・プログラム を参照してください。MOVE 操作の場合、UNCATLG は出力データ・セットがカタログ化されないようにします。

CATLG

コピー先のデータ・セットを以下のようにカタログ化するよう指定します。

1. 統合カタログ機能または /JOB CAT/STEP CAT カタログ内でカタログ化が行なわれます。
2. RENAME および FROM パラメーターの両方を省略した場合、ソース・データ・セットの項目を適切なカタログから削除することによって、コピー先データ・セットの再カタログ化が可能になります。

適切な結果を出すためには、DD ステートメントの『DISP』パラメーターで『CATLG』オプションを指定する代わりに、この制御ステートメント・オプションを使用する必要があります。

CATLG は COPY PDS でのみコーディングできます。

RENAME=name

データ・セットを名前変更するよう指定して、新しい名前を指示します。

FROMDD=ddname

テープ・ボリュームに収められている入力データ・セットのための、DCB 情報や (データ・セット・シーケンス番号を除く) LABEL 情報を入手できる DD ステートメント名を指定します。このテープ・データ・セットは、アンロードされた区分データ・セットでなければなりません。データ・セットに標準ラベルが付いていて、9トラックからなる1つのテープ・ボリュームに収められている場合、この FROMDD オペランドを省略できます。

TODD=ddname

データ・セットをテープにアンロードする際、出力データ・セットのための DCB 情報 (RECFM、BLKSIZE および LRECL を除く)、および LABEL 情報 (データ・セット・シーケンス番号を除く) を入手できる DD ステートメント名を指定します。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報をコードする場合、(RECFM=FB, BLKSIZE=800, LRECL=80) と指定しなければなりません。

有効期限データ (EXPDT) や保存期間 (RETPD) を作成または変更するときは、制御ステートメント内で TODD を指定する必要があります。

標準ラベルの付いた、指定した装置タイプのデフォルト密度になっている9トラックのテープの場合は、TODD パラメーターを省略することができます。

UNLOAD

データ・セットを受け取り側ボリュームにアンロードするよう指定します。

COPYAUTH

入力データ・セットが RACF で保護されており、しかも出力データ・セットが事前に割り振られていない場合、入力データ・セットと同じアクセス・リストを受け取り側データ・セットに提供するよう指定します。

MOVE VOLUME および COPY VOLUME ステートメント

MOVE VOLUME ステートメントを使用して、指定したボリュームに収められているすべてのデータ・セットを移動します。COPY VOLUME ステートメントを使用して、指定したボリュームに収められているすべてのデータ・セットをコピーします。

データ・セットに関連したカタログ項目は、いずれも変更されません。移動元またはコピー元のデータ・セットは、DASD ボリュームに収められていなければなりません。

MOVE VOLUME および COPY VOLUME ステートメントの構文は次のとおりです。

[label]	{MOVE} COPY}	VOLUME=device=serial ,TO=device=list [PASSWORD] [CATLG] [TODD=ddname]
---------	-----------------	-----------------------------------------------------------------------------------

		[UNLOAD] [COPYAUTH]
注: CATLG は COPY VOLUME でのみコーディングできます。		

各項目の意味は以下のとおりです。

VOLUME=*device=serial*

ソース・ボリュームの装置タイプおよびボリューム通し番号を指定します。

TO=*device=serial*

データ・セットのボリュームの移動先またはコピー先ボリュームの装置タイプとボリューム通し番号を指定します。

PASSWORD

パスワードで保護されたデータ・セットを操作に含めるよう指定します。これは VSAM パスワード保護ではなく、OS パスワード体系です。PASSWORD を省略した場合、保護されていないデータ・セットのみがコピーまたは移動されます。

CATLG

コピーされたデータ・セットをカタログ化するよう指定します。

適切な結果を出すためには、DD ステートメントの『DISP』パラメーターで『CATLG』オプションを指定する代わりに、この制御ステートメント・オプションを使用する必要があります。

CATLG は COPY VOLUME でのみコーディングできます。

TODD=*ddname*

テープ・ボリュームに収められる一連の出力データ・セットのための、DCB 情報 (RECFM、BLKSIZE および LRECL を除く)、および LABEL 情報 (データ・セット・シーケンス番号を除く) を入手できる DD ステートメント名を指定します。

この DD ステートメントは、テープ・ボリューム上に出力データ・セットを作成するときに使用するモードおよびラベルの情報を記述しています。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報がコーディングされていれば、それらは無視されます。

UNLOAD が指定されている場合、この DD ステートメントは、アンロードされたバージョンのデータ・セットを磁気テープ・ボリュームに作成するときに使用するモード情報およびラベル情報を記述します。レコード・フォーマット、ブロック・サイズ、および論理レコード長に関する情報をコードする場合、(RECFM=FB, BLKSIZE=800, LRECL=80) と指定しなければなりません。

有効期限データ (EXPDT) や保存期間 (RETPD) を作成または変更するときは、制御ステートメント内で TODD を指定する必要があります。

標準ラベルの付いた、指定した装置タイプのデフォルト密度になっている 9 トラックのテープの場合は、TODD パラメーターを省略することができます。

UNLOAD

データ・セットを受け取り側ボリュームにアンロードするよう指定します。

COPYAUTH

入力データ・セットが RACF で保護されており、しかも出力データ・セットが

事前に割り振られていない場合、入力データ・セットと同じアクセス・リストを受け取り側データ・セットに提供するよう指定します。

INCLUDE ステートメント

INCLUDE ステートメントを使用して、MOVE DSGROUP、COPY DSGROUP、MOVE PDS、または COPY PDS のスコープを拡張します。ここでは、これらのステートメントで明示的に定義されていないメンバーまたはデータ・セットを含めます。INCLUDE ステートメントは、MOVE または COPY ステートメントの後に続けて指定します。ここに含まれる区分データ・セットのレコード特性は、移動またはコピーされるほかの区分データ・セットと互換性がなければなりません。

INCLUDE ステートメントをいくつ使用しても、MOVE または COPY ステートメントを変更できます。区分データ・セットの場合、データをアンロードする時、あるいはアンロードされたデータを移動またはコピーする時には、INCLUDE ステートメントが無効になります。DSGROUP 操作の場合、INCLUDE は無効です。

INCLUDE ステートメントの構文は次のとおりです。

[<i>label</i>]	INCLUDE	DSNAME= <i>name</i> [,MEMBER= <i>membername</i>] [{FROM= <i>device</i> = { <i>serial</i> (<i>list</i>)}]
------------------	----------------	-------------------------------------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

DSNAME=*name*

データ・セットの完全修飾名を指定します。MOVE または COPY DSGROUP とともに使用すると、ここで指定したデータ・セットがグループ内に含まれます。MOVE または COPY PDS とともに使用すると、ここで指定した区分データ・セット、または (MEMBER パラメーターを指定した場合) そのメンバーのいずれかが操作に含まれます。

MEMBER=*membername*

DSNAME パラメーターで指定した区分データ・セット内の、1つのメンバーの名前を指定します。ここで指定したメンバーは、移動またはコピーされる区分データ・セットとマージされます。ここで指定したメンバーを含んでいる区分データ・セット、およびこのメンバーそのものは、どちらもスクラッチされません。

FROM=*device***=**{*serial* (*list*)}

操作に含めるデータ・セットが収められているボリュームの、装置タイプと通し番号を指定します。データ・セットが複数のボリュームに収められている場合、通し番号のリストを括弧で囲みます (コンマで各番号を区切る)。

データ・セットがカタログ化されている場合は、『FROM』をコーディングしないでください。

装置タイプではなく特定の装置を指定する場合は、*device* サブパラメーターにその装置番号をコーディングしてください。

データ・セットが磁気テープ装置に収められている場合、『(通し番号、シーケンス番号)』という形式で、通し番号とデータ・セット・シーケンス番号を括弧で囲んで指定する必要があります。

EXCLUDE ステートメント

EXCLUDE ステートメントを使用して、MOVE DSGROUP、COPY DSGROUP、MOVE PDS、COPY PDS、MOVE CATALOG、または COPY CATALOG 操作の効果を制限します。このステートメントでは、それらのステートメント内で定義されているデータの特定部分を除外します。

区分データ・セットのメンバーのうち、MOVE PDS 操作から除外されたものを回復することはできません (ソース・データ・セットはスクラッチされる)。EXCLUDE ステートメントをいくつ使用しても、MOVE PDS または COPY PDS ステートメントを変更できません。

MOVE DSGROUP または MOVE CATALOG 操作から除外されたソース・データ・セットは、引き続き使用できます。MOVE DSGROUP、COPY DSGROUP の変更で使用できる EXCLUDE ステートメントは 1 つだけです。データをアンロードする時、あるいはアンロードされたデータを移動またはコピーする時には、EXCLUDE ステートメントは無効になります。

EXCLUDE ステートメントの構文は次のとおりです。

[label]	EXCLUDE	{DSGROUP= <i>name1</i> MEMBER= <i>membername</i> }
---------	---------	----------------------------------------------------

各項目の意味は以下のとおりです。

DSGROUP=*name*

データ・セット・グループまたはカタログを移動する際に、除外するカタログ式データ・セットまたはカタログ項目を指定します。MOVE DSGROUP または COPY DSGROUP とともに使用すると、*name* が修飾する名前をもつすべてのカタログ式データ・セットを操作から除外します。MOVE CATALOG または COPY CATALOG とともに使用すると、*name* が修飾する名前をもつすべてのカタログ項目を操作から除外します。

MEMBER=*membername*

MOVE または COPY PDS 操作から除外するメンバーの名前を指定します。

SELECT ステートメント

SELECT ステートメントを MOVE PDS または COPY PDS ステートメントとともに使用して、移動またはコピーするメンバーを選択したり、これらのメンバーの名前を変更したりします。同じ MOVE PDS または COPY PDS を変更するために、SELECT ステートメントを EXCLUDE ステートメントや REPLACE ステートメントとともに使用することはできません。データをアンロードする時、あるいはアンロードされたデータを移動またはコピーする時には、SELECT ステートメントは無効になります。ソース・データ・セットがスクラッチされるため、MOVE PDS 操作で選択されていないメンバーを回復することはできません。

SELECT ステートメントの構文は次のとおりです。

[label]	SELECT	MEMBER={(<i>name1</i> [, <i>name2</i>][,...]) ((<i>name1,newname1</i>) [,(<i>name2,newname2</i>)][,...])}
---------	--------	---------------------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

MEMBER={(*name1* [*,name2*][*,...*])| (*(name1,newname1)* [*(name2,newname2)*][*,...*])}
 MOVE または COPY PDS 操作で移動またはコピーするメンバー名を指定し、
 オプションで、それらのメンバーに割り当てる新しい名前を指定します。

REPLACE ステートメント

REPLACE ステートメントを MOVE PDS または COPY PDS ステートメントとともに使用して、あるメンバーを操作から除外し、別の区分データ・セットからのメンバーと置換します。新しいメンバーは古いメンバーと同じ名前を持ち、レコード特性に互換性がなければなりません。REPLACE ステートメントをいくつか使用しても、MOVE PDS または COPY PDS ステートメントを変更できます。データをアンロードする時、あるいはアンロードされたデータを移動またはコピーする時には、REPLACE ステートメントは無効になります。

REPLACE ステートメントの構文は次のとおりです。

[<i>label</i>]	REPLACE	DSNAME = <i>name</i> ,MEMBER = <i>name</i> [{FROM = <i>device=serial</i>
------------------	----------------	------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

DSNAME=*name*

置換するメンバーを含んでいる区分データ・セットの完全修飾名を指定します。

MEMBER=*membername*

DSNAME パラメーターで指定した区分データ・セット内の、1つのメンバーの名前を指定します。このメンバーは、移動またはコピーされる区分データ・セット内の同じ名前を持つメンバーと置き換わります。ここで指定したメンバーを含んでいる区分データ・セット、およびこのメンバーそのものは、どちらもスクラッチされません。

FROM=*device=serial*

置換するメンバーを含むデータ・セットが収められているボリュームの、装置タイプおよびボリューム通し番号を指定します。

データ・セットがカタログ化されている場合は、『FROM』をコーディングしないでください。

装置タイプではなく特定の装置を指定する場合は、*device* サブパラメーターにその装置番号をコーディングしてください。

IEHMOVE の例

以下の各例では、IEHMOVE の使用方法の一部を示します。324 ページの表 53 は IEHMOVE の使用例の早見表として使うことができます。『例』の欄に示されている番号は、この表の後にある各例に対応しています。

IEHMOVE

表 53. IEHMOVE の例の一覧表

操作	データ・セット編成	装置	注釈	例
MOVE	データ・セット・グループ	ディスク	データ・セット・グループを移動します。	8
MOVE	区分	ディスク	区分データ・セットを移動します。別の区分データ・セットからの 1 つのメンバーをこれにマージします。	2
MOVE	区分	ディスク	事前にスペースが割り振られたボリュームに、データ・セットを移動します。	4
MOVE	区分	ディスク	事前にスペースが割り振られた 1 つのボリュームに、3 つのデータ・セットを移動してアンロードします。	5
MOVE	順次	ディスク	ジョブの完了後に、ソース・ボリュームをデマウントします。マウント可能なディスクは 2 つあります。	1
MOVE	順次	ディスクおよび磁気テープ	順次データ・セットを、9トラックか らなるラベルなしテープ・ボリューム にアンロードします。	6
MOVE	順次	ディスクおよび磁気テープ	アンロードされたデータ・セットを、1 つのボリュームからロードします。	7
MOVE	ボリューム	ディスク	データ・セットのボリュームをディス ク・ボリュームに移動します。	3

実際の装置名または装置番号を指定するべきところで **disk** または **tape** を使用している例は、使用する前に変更する必要があります。実際の装置名または装置番号は、システムに対する装置定義の方法によって異なります。

例 1: 複数の順次データ・セットをディスク・ボリュームから別々のボリュームに移動する

この例では、3 つの順次データ・セット (SEQSET1、SEQSET2、および SEQSET3) を 1 つのディスク・ボリュームから 3 つの別々のディスク・ボリュームに移動します。3 つの受け取り側ボリュームは、それぞれ IEHMOVE から要求された時点で取り付けられます。ソース・データ・セットはカタログ化されていません。スペースの割り振りは IEHMOVE が行います。

```
//MOVEDS JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=333333,DISP=OLD
//DD1 DD UNIT=(disk,,DEFER),DISP=OLD,
// VOLUME=(PRIVATE,,SER=(222222))
//DD2 DD UNIT=(disk,,DEFER),DISP=OLD,
// VOLUME=(PRIVATE,,SER=(222333))
//DD3 DD UNIT=(disk,,DEFER),DISP=OLD,
// VOLUME=(PRIVATE,,SER=(222444))
//DD4 DD VOLUME=(PRIVATE,RETAIN,SER=(444444)),
// UNIT=disk,DISP=OLD
//SYSIN DD *
```

```

MOVE   DSNAME=SEQSET1,T0=disk=222222,FROM=disk=444444
MOVE   DSNAME=SEQSET2,T0=disk=222333,FROM=disk=444444
MOVE   DSNAME=SEQSET3,T0=disk=222444,FROM=disk=444444
/*

```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入るディスク装置を定義しています。
- DD1、DD2、および DD3 DD では、受け取り側のボリュームを定義しています。
- DD4 DD では、ソース・ボリュームが取り付けられている装置を定義しています。RETAIN サブパラメーターが含まれているため、ジョブが完了するまでボリュームは取り付けられたままになります。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE は、ソース・データ・セットをそれぞれボリューム 222222、222333、および 222444 に移動します。ソース・データ・セットはスクラッチされます。

例 2: 区分データ・セットをディスク・ボリュームに移動してマージする

この例では、区分データ・セット (PARTSET1) を 1 つのディスク・ボリュームへ移動します。さらに、別の区分データ・セット (PARTSET2) からのメンバー (PARMEM3) を受け取り側ボリューム上のソース・メンバーとマージします。ソース区分データ・セット (PARTSET1) はスクラッチされます。スペースの割り振りは IEHMOVE が行います。

```

//MOVEPDS JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=333000,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,VOLUME=SER=222111,DISP=OLD
//DD3 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD4 DD UNIT=disk,VOLUME=SER=222333,DISP=OLD
//SYSIN DD *
MOVE PDS=PARTSET1,T0=disk=222333,FROM=disk=222111
INCLUDE DSNAME=PARTSET2,MEMBER=PARMEM3,FROM=disk=222222
/*

```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入るディスク・ボリュームを定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2、DD3、および DD4 ステートメントでは、それぞれ 2 つのソース・ボリュームを収めている装置、および受け取り側ボリュームを収める装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE では、ソース区分データ・セット、それを含むボリューム、およびそれを受け取るボリュームを定義しています。
- INCLUDE では、2 番目の区分データ・セットからの 1 つのメンバーを操作に含めています。

例 3: データ・セットのボリュームをディスク・ボリュームに移動する

この例では、データ・セットのボリュームをディスク・ボリュームに移動します。正常に移動されたすべてのデータ・セットは、ソース・ボリュームからスクラッチされます。ただし、それらのデータ・セットに関連したカタログ項目は、いずれも変更されません。スペースの割り振りは IEHMOVE が行います。

```
//MOVEVOL JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD3 DD UNIT=disk,VOLUME=SER=333333,DISP=OLD
//SYSIN DD *
MOVE VOLUME=disk=333333,T0=disk=222222,PASSWORD
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入る装置を定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、受け取り側ボリュームが取り付けられている装置を定義しています。
- DD3 DD では、ソース・ボリュームが取り付けられている装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE では、データ・セットのボリュームの移動操作を指定し、ソース・ボリュームおよび受け取り側ボリュームを定義しています。さらに、このステートメントは、パスワードで保護されたデータ・セットを操作に含めるよう指定しています。

例 4: 区分データ・セットを割り振りスペースに移動する

この例では、区分データ・セット用に事前にスペースが割り振られたディスク・ボリュームへ、区分データ・セットを移動します。ソース・データ・セットはスクラッチされます。

```
//MOVEPDS JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD3 DD UNIT=disk,VOLUME=SER=333333,DISP=OLD
//SYSIN DD *
MOVE PDS=PDSSET1,T0=disk=222222,FROM=disk=333333
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入る装置を定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、受け取り側ボリュームが取り付けられる装置を定義しています。

- DD3 DD では、ソース・ボリュームが取り付けられている装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE では、区分データ・セット PDSSET1 の移動操作を指定し、ソース・ボリュームおよび受け取り側ボリュームを定義しています。

例 5: 区分データ・セット・ボリュームを移動してアンロードする

この例では、3 つの区分データ・セットを 3 つの別々のソース・ボリュームから 1 つのディスク・ボリュームに移動します。ソース・データ・セット PDSSET3 はアンロードされます。(レコード・サイズが受け取り側ボリュームのトラック容量を上回ります。)

```
//MOVEPDS JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=(disk,,DEFER),DISP=OLD,
// VOLUME=(PRIVATE,,SER=(333333))
//DD3 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//SYSIN DD *
MOVE PDS=PDSSET1,T0=disk=222222,FROM=disk=333333
MOVE PDS=PDSSET2,T0=disk=222222,FROM=disk=222222
MOVE PDS=PDSSET3,T0=disk=222222,FROM=disk=444444,UNLOAD
/*
```

PDSSET1、PDSSET2、および PDSSET3 は、あらかじめ受け取り側ボリューム上に割り振られています。PDSSET3 は順次データ・セットとして、PDSSET1 および PDSSET2 は区分データ・セットとして割り振られています。PDSSET3 は順次データ・セットに移動されるため、アンロードされます。

スペース割り振りの見積もりに関する説明は、*z/OS DFSMS* データ・セットの使用法を参照してください。

PDSSET3 の DCB 属性は次のとおりです。

```
DCB=(RECFM=U,BLKSIZE=5000)
```

アンロードされた属性は次のとおりです。

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入る装置を定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、必要な時点でソース・ボリュームを取り付ける装置を定義しています。
- DD3 DD では、受け取り側ボリュームが取り付けられている装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE では、区分データ・セットの移動操作を指定し、各データ・セットのソース・ボリュームおよび受け取り側ボリュームを定義しています。

例 6: 順次データ・セットをラベルなしテープ・ボリュームにアンロードする

この例では、1つの順次データ・セットを、9トラックのラベルなしテープ・ボリューム (800 ビット / インチ) にアンロードします。

```
//UNLOAD JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//TAPEOUT DD UNIT=tape,VOLUME=SER=SCRCH2,DISP=OLD,
// DCB=(DEN=2,RECFM=FB,LRECL=80,BLKSIZE=800),
// LABEL=(,NL)
//SYSIN DD *
MOVE DSNAME=SEQSET1,TO=tape=SCRCH2,FROM=disk=222222,TODD=TAPEOUT
/*
```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入る装置を定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、ソース・ボリュームが取り付けられている装置を定義しています。
- TAPEOUT DD では、受け取り側テープ・ボリュームが取り付けられる装置を定義しています。また、このステートメントでは、ラベルおよびモードに関する情報も指定しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE は、順次データ・セット SEQSET1 をディスク・ボリュームから受け取り側テープ・ボリュームに移動します。このデータ・セットはアンロードされます。このステートメントの TODD パラメーターは、ラベル情報およびモード情報を得るために TAPEOUT DD ステートメントを参照します。

例 7: アンロードされた順次データ・セットをラベル付きテープからロードする

この例では、アンロードされた3つの順次データ・セットを、ラベル付きの7トラック・テープ・ボリューム (556 ビット / インチ) からディスク・ボリュームにロードします。スペースの割り振りは IEHMOVE が行います。この例では、元のフォームのデータ・セットをディスク・ボリュームがサポートできることを想定しています。

```
//LOAD JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//TAPESETS DD UNIT=tape,VOLUME=SER=001234,DISP=OLD,
// LABEL=(1,SL),DCB=(DEN=1,TRTCH=C)
//SYSIN DD *
MOVE DSNAME=UNLDSET1,TO=disk=222222, X
FROM=tape=(001234,1),FROMDD=TAPESETS
MOVE DSNAME=UNLDSET2,TO=disk=222222, X
```

72

```

FROM=tape=(001234,2),FROMDD=TAPESETS
MOVE  DSNAME=UNLDSET3,T0=disk=222222,      X
      FROM=tape=(001234,3),FROMDD=TAPESETS
/*

```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入る装置を定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、受け取り側ボリュームが取り付けられる装置を定義しています。
- TAPESETS DD では、ソース・テープ・ボリュームが取り付けられている装置を定義しています。このステートメントでは、DCB 情報も提供しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE は、アンロードされたデータ・セットを受け取り側ボリュームに移動します。

複数のデータ・セットを収めているテープ・ボリュームから 1 つのデータ・セットを移動するためには、ユーティリティ制御ステートメントの FROM パラメータの list フィールドで、そのデータ・セットのシーケンス番号を指定する必要があります。

例 8: カタログ式データ・セットのグループを移動する

この例では、データ・セット A.B.C.X、A.B.C.Y、および A.B.C.Z で構成されるカタログ式データ・セット・グループ A.B.C を、2 つのディスク・ボリュームから 3 番目のボリュームに移動します。スペースの割り振りは IEHMOVE が行います。受け取り側ボリュームを参照するために、カタログが更新されます。ソース・データ・セットはスクラッチされます。

```

//MOVEDSG JOB ...
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//DD3 DD UNIT=disk,VOLUME=SER=333333,DISP=OLD
//DD4 DD UNIT=disk,VOLUME=SER=444444,DISP=OLD
//SYSIN DD *
MOVE DSGROUP=A.B.C,T0=disk=222222
/*

```

各制御ステートメントについての説明は以下のとおりです。

- SYSUT1 DD では、作業データ・セットが入る装置を定義しています。
- DD1 DD では、システム常駐装置を定義しています。
- DD2 DD では、受け取り側ボリュームが取り付けられる装置を定義しています。
- DD3 DD では、ソース・ボリュームのうち 1 つが取り付けられている装置を定義しています。
- DD4 DD では、ソース・ボリュームのうち 1 つが取り付けられている装置を定義しています。
- SYSIN DD では、入力ストリーム内の後続の制御データ・セットを定義します。
- MOVE は、指定されたデータ・セットをボリューム 222222 に移動します。

この例では、DD4 ステートメントを使わずに 1 つ少ない取り付け可能ディスク装置を使用しても、同様の結果が得られます。DD3 と DD4 を使用する場合は、両

IEHMOVE

方のソース・ボリュームがジョブの始めに取り付けられます。DD3 だけを使用する場合は、ボリューム 333333 がジョブの始めに取り付けられます。ボリューム 333333 が処理されると、ユーティリティーはボリューム 444444 を取り付けられるようオペレーターに要求します。この場合、DD3 ステートメントを次のようにコーディングします。

```
//DD3 DD UNIT=(disk,,DEFER),DISP=OLD,  
//      VOLUME=(PRIVATE,,SER=(333333))
```

第 14 章 IEHPROGM (プログラム維持) プログラム

IEHPROGM は、組織レベルでシステム制御データを変更したりデータ・セットを維持したりするためのシステム・ユーティリティです。IEHPROGM は、その使用をローカルに許可されているプログラマーだけが使用するようになっています。

パスワードによって保護されているデータ・セットの処理中の動的 UCB は、IEHPROGM ではサポートされていません。

IEHPROGM を使用して、以下のタスクを実行できます。

- データ・セットまたは区分データ・セットのメンバーをスクラッチ (削除) する。
- データ・セットまたは区分データ・セットのメンバーの名前を変更する。
- データ・セット・パスワードを維持する。

IEHPROGM を使用するには、RACF 権限が必要です。ストレージ管理サブシステム (SMS) のための RACF 要件については、*z/OS DFSMSdfp* ストレージ管理リファレンスを参照してください。

SMS 管理データ・セットの場合は、IDCAMS を使用することをお勧めします。IDCAMS については、*z/OS DFSMS* カタログのためのアクセス方式サービス・プログラムを参照してください。

アセンブラー・プログラムを書いて任意の IEHPROGM 機能を実行することができます。*z/OS DFSMSdfp* 拡張サービス および *z/OS DFSMS* データ・セットの使用法を参照してください。

データ・セットまたはメンバーのスクラッチまたは名前変更

IEHPROGM は、1 つ以上の DASD ボリュームから下記のデータ・セットをスクラッチする場合に使用できます。

- 順次、BDAM、区分データ・セット、または PDSE。それらは、オペレーティング・システムによって名前を指定できます。
- 区分データ・セットのメンバー。
- 一時 VSAM データ・セット。

データ・セットは、それが属するボリュームのボリューム目録 (VTOC) から、対応するデータ・セット制御ブロックが削除された時点でスクラッチされたと見なされます。再割り振りによって、そのスペースは使用可能になります。

メンバーは、それが属する区分データ・セットのディレクトリーからその名前が削除された時点でスクラッチされたと見なされます。

PDSE 以外の区分データ・セットの場合、スクラッチされたメンバーの占めていたスペースを再割り振りのために使用することは、その区分データ・セットがスクラッチまたは圧縮されるまでできません。(区分データ・セットのメンバーをスクラッチした場合、そのメンバーの別名もディレクトリーからすべて除去する必要があります。)

SCRATCH 要求において PURGE または NOPURGE キーワードが指定されていても、SMS 管理データ・セットについては無視されます。SMS 管理以外のデータ・セットの場合、PURGE および NOPURGE キーワードが使用されていれば、それはそのまま変更なしで使用されます。

- SYS1.PARMLIB の IGDSMSxx メンバーで OVRD_EXPDT(NO) が指定されている場合、または OVRD_EXPDT キーワードが指定されていない場合、PURGE および NOPURGE キーワードが考慮されます。
- SYS1.PARMLIB の IGDSMSxx メンバーで OVRD_EXPDT(YES) が指定されている場合、PURGE および NOPURGE キーワードは考慮されません。有効期限が切れているかどうかに関係なく、データ・セットは常に削除されます。このことが当てはまるのは、データ・セットが DASD データ・セットで、かつ SMS 管理の場合だけです。

ストレージ管理サブシステム (SMS) で管理されるデータ・セットを スクラッチまたは名前変更する場合、VOL パラメーターの装置タイプとボリューム・リストは、そのデータ・セットに実際に割り振られているボリュームを反映するものでなければなりません。この制限は、SMS 管理データ・セットについても SMS 管理でないデータ・セットについても当てはまります。しかし、SMS 管理データ・セットの割り振り時にボリュームを指定した場合、SMS はそのボリュームにデータ・セットを自動的に割り振るわけではありません。

SMS 管理データ・セットをスクラッチする場合、IEHPROGM はそのデータ・セットをアンカタログします。

SMS 管理ボリューム上にあるアンカタログされたデータ・セットを削除する場合は、IDCAMS DELETE VR を使用してください。SMS 管理ボリューム上にあるアンカタログされたデータ・セットをスクラッチしようとした場合、そのデータ・セットのカタログ・バージョンがあれば、IEHPROGM はそのカタログ・バージョンのスクラッチ (アンカタログ) 「しか」実行しません。IEHPROGM で指定したボリュームが SMS 管理の場合、カタログ検索を使用してそのデータを含むボリュームが識別されます。カタログ検索は IEHPROGM で指定されたものとは異なるボリュームを戻し、間違ったデータ・セットがスクラッチされることになってしまいます。

データ・セットをスクラッチまたは名前変更する場合、VOL パラメーターの装置タイプとボリューム・リストは、そのデータ・セットに実際に割り振られているボリュームを反映するものでなければなりません。

IEHPROGM では、索引付き VTOC の索引を含むデータ・セットはスクラッチしません。

IEHPROGM を使用すると、DASD ボリューム上に存在するデータ・セットまたはメンバーの名前を変更することができます。さらに、このプログラムによって任意のメンバーの別名を変更することもできます。

SMS 管理データ・セットの名前を変更する場合、IEHPROGM はそのデータ・セットをアンカタログしてから、そのデータ・セットを適切なカタログに新しい名前でカタログします。別名が原因でアンカタログを実行できない場合、IEHPROGM はデータ・セットの名前を変更しません。

一時 VSAM データ・セットは、SCRATCH VTOC,SYS を使用してスクラッチすることができます。

RACF がアクティブである場合、RACF 定義データ・セットのスクラッチには ALTER 許可が必要であり、また区分データ・セットのメンバーのスクラッチまたは名前変更には UPDATE 許可が必要です。

注: IBM のセキュリティー・パッケージである RACF では、総称プロファイルによってのみカバーされているデータ・セットの名前を、総称プロファイルによってはカバーされない名前に変更することはできないことになっています。そのような操作は、そのデータ・セットの保護を解除することになってしまうためです。

データ・セット・パスワードの維持

IEHPROGM は、PASSWORD データ・セット中の非 VSAM パスワード項目を維持したり、データ・セット制御ブロック (DSCB) の中の DASD データ・セットの保護状況を変更する場合に使用することができます。データ・セット・パスワードと PASSWORD データ・セットについては、*z/OS DFSMSdfp 拡張サービス* を参照してください。その資料では、データ・セット・パスワードのセキュリティーが不十分である理由や、z/OS Security Server (RACF) をお勧めする理由について説明されています。

データ・セットのパスワード保護には 3 種類のうちのいずれかが使用されます。それは、DASD データ・セットの場合は DSCB の中で、また磁気テープ・データ・セットの場合は磁気テープ・ラベルの中で示されます。DSCB の内容については、*z/OS DFSMSdfp 拡張サービス* を参照してください。磁気テープ・ラベルについては、*z/OS DFSMS Using Magnetic Tapes* を参照してください。

データ・セット・パスワード保護の種類は次のとおりです。

- 保護なし。データ・セットの読み取りまたは書き込みにパスワードは不要。
- 読み取り/書き込み保護。データ・セットの読み取りまたは書き込みにパスワードが必要。
- 書き込み保護。データ・セットへの書き込みにのみパスワードが必要。データ・セットの読み取りにパスワードは不要。

システム・データ・セットがパスワードで保護されていて、データ・セットに関して問題が発生した場合には、そのデータ・セットにアクセスして問題を解決するために、保守要員がパスワードを提供する必要があります。

1 つのデータ・セットに、1 つまたは複数のパスワードを割り当てることができます。それぞれのパスワードに対して、PASSWORD データ・セットの中に 1 つずつ項目が作成されます。データ・セットに割り当てられるパスワードとしては、データ・セットの読み取り/書き込みアクセスを許可するものと、読み取り専用アクセスを許可するものが可能です。

334 ページの図 44 に、データ・セット ABC の保護状況と、そのデータ・セットに割り当てられたパスワードによって許可されるアクセスの種類との間の関係を示します。データ・セット ABC には、パスワード ABLE と BAKER が割り当てら

れています。DSCB または磁気テープ・ラベルの中でパスワード保護が設定されていないなら、データ・セット ABC はパスワードなしで読み取りと書き込みが可能です。DSCB または磁気テープ・ラベルの中で読み取り/書き込み保護が設定されている場合、データ・セット ABC はパスワード ABLE または BAKER によって読み取ることができ、またパスワード ABLE によって書き込みが可能です。DSCB または磁気テープ・ラベルの中で書き込み保護が設定されている場合、データ・セット ABC はパスワードなしで読み取ることができ、パスワード ABLE によって書き込みが可能です。パスワード BAKER が必要とされることはありません。

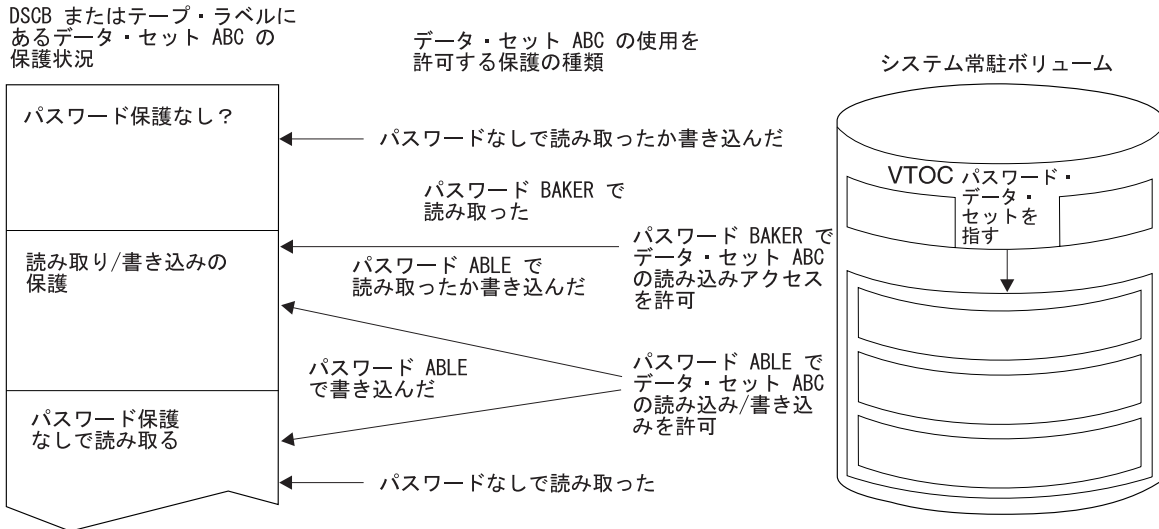


図 44. データ・セットの保護状況とそのパスワードとの関係

IEHPROGM を使用してデータ・セットのパスワードを維持するには、それ以前に PASSWORD データ・セットがシステム常驻ボリュームに存在していなければなりません。その場合に IEHPROGM を使用することによって、次のことを実行できます。

- PASSWORD データ・セットに項目を追加する。
- PASSWORD データ・セットの項目を置換する。
- PASSWORD データ・セットから項目を削除する。
- PASSWORD データ・セットの項目の情報のリストを作成する。

PASSWORD データ・セット内の 1 つの項目には、保護データの名前、パスワード、パスワードの保護モード、アクセス・カウンター、およびオプションとして 77 バイトのユーザー・データが含まれます。パスワードの保護モードは、そのパスワードによって可能なアクセスの種類と、そのパスワードが制御パスワードかそれとも 2 次パスワードかを定義するものです。特定のデータ・セットについて PASSWORD データ・セットに追加される最初のパスワードが、その項目の中でそのデータ・セットの制御パスワードとなります。それと同じデータ・セットに追加される 2 番目以降のパスワードは、2 次パスワードです。

DASD データ・セットの場合、IEHPROGM は制御パスワードの項目が追加、置換、または削除された時点で DSCB の中の保護状況を更新します。それによって、既存の DASD データ・セットの保護状況を、そのパスワードの追加、置換、または

削除と同時に設定およびリセットすることが可能になります。IEHPROGM が DSCB 内に記録されているデータ・セット保護状況を変更するには、下記の条件が成立している場合です。

- そのデータ・セットの制御パスワードが追加、置換、または削除されている。
- そのデータ・セットはオンラインである。
- そのデータ・セットの属するボリュームが、ユーティリティ制御ステートメントに指定されているか、またはそのデータ・セットがカタログされている。
- そのデータ・セットは、IEHPROGM ジョブの中に割り振られていない。

磁気テープ・データ・セットの場合、IEHPROGM はパスワードの項目が追加、置換、または削除された時点で磁気テープ・ラベルの中の保護状況を更新できません。磁気テープ・ラベルに記録されている保護状況は、JCL で設定する必要があります。

追加、置換、削除、またはリスト表示するパスワードは、ユーティリティ制御ステートメントで指定したり、コンソール・オペレーターが入力したりすることが可能です。ユーティリティ制御ステートメントのパスワードが欠落している場合や無効である場合、IEHPROGM はコンソール・オペレーターに対してメッセージを発行します。そのメッセージには、ジョブ名、ステップ名、およびユーティリティ制御ステートメントが含まれており、欠落しているか無効である特定のパスワードが示されています。各ユーティリティ制御ステートメントの 1 つのパスワード項目に対して、無効なパスワードが 2 回入力されると、要求は無視されます。1 つのジョブ・ステップの中のすべてのユーティリティ制御ステートメントを通じて、パスワード項目に対して無効なパスワードが合計で 5 回入力されると、そのステップはキャンセルされます。

データ・セット・パスワードの追加

データ・セットにパスワードを追加すると、PASSWORD データ・セットの中に 1 つの項目が作成されます。それには、指定されたデータ・セット名、パスワード名、そのパスワードの保護モード (読み取り/書き込みか読み取り専用か)、そしてオプションとして 77 バイトのユーザー・データが含まれます。その項目のアクセス・カウンターは 0 に設定されます。

2 次パスワードを追加、置換、または削除するには、常にデータ・セットの制御パスワードが指定されていない限りなりません。しかし、2 次パスワード項目の情報のリストを作成する場合、制御パスワードは指定しないでください。

データ・セットに 2 次パスワードを割り当てると、一部のユーザーをデータ・セットの読み取りだけに制限したり、特定のユーザーがデータ・セットにアクセスする回数を記録したりすることができます。各パスワード項目の中のアクセス・カウンターには、パスワードを使用してそのデータ・セットが正常にオープンされた回数記録されます。

オンライン DASD データ・セットの制御パスワードが追加された場合、そのデータ・セットの保護状況 (読み取り/書き込みかそれとも書き込みか) がその DSCB の中に設定されます。

データ・セット・パスワードの置換

パスワード項目の中で、パスワード、パスワードの保護モード (読み取り/書き込みか読み取り専用か)、および 77 バイトのユーザー・データの各情報は置換することができます。データ・セットの保護状況は、データ・セットの制御項目を置換することによって変更できます。

オンライン DASD データ・セットの制御項目が置換されると、DSCB もそのデータ・セットの保護状況の中で変更内容を示すように設定が変更されます。したがって、DASD データ・セットの保護情報を変更する場合には、ボリュームがオンラインであることが必要です。

データ・セット・パスワードの削除

PASSWORD データ・セットから制御パスワード項目を削除すると、そのデータ・セットのすべての 2 次パスワード項目も削除されます。しかし、2 次項目が削除されても、それ以外のパスワード項目は削除されません。

オンライン DASD データ・セットの制御パスワード項目が削除されると、DSCB の中に記録されているそのデータ・セットの保護状況も、保護がなくなったことを示すように変更されます。DASD データ・セットの制御パスワードを削除する場合は、ボリュームがオンラインであるようにする必要があります。ボリュームがオンラインでないなら、パスワード項目は削除されますが、DSCB の中ではデータ・セットが保護されていることが示されたままになっています。そのデータ・セットは、そのデータ・セットのための別のパスワードを追加しない限りアクセスできなくなります。

磁気テープ・データ・セットの制御パスワード項目が削除されると、そのテープ・ボリュームは、そのデータ・セットのための別のパスワードを追加しない限りアクセスできなくなります。

スクラッチされたデータ・セットのすべてのパスワード項目を削除して、新しい項目のためにスペースを使用できるようにするには、削除機能を使用してください。

パスワード項目のリスト表示

PASSWORD データ・セット中の項目の情報のリストは、その項目のパスワードを提供することによって SYSPRINT データ・セットの中で取得できます。リストには、パスワードを使用してデータ・セットが正常にオープンされた回数、そのパスワードの種類 (制御パスワードか 2 次パスワードか)、そのパスワードによって可能なアクセスの種類 (読み取り/書き込みか読み取り専用か)、およびその項目内のユーザー・データが含まれています。図 45 に、パスワード項目から出力される情報リストの例を示します。

```
DECIMAL ACCESS COUNT= 000025
PROTECT MODE BYTE= SECONDARY, READ ONLY
USER DATA FIELD= ASSIGNED TO J. BROWN
```

図 45. パスワード項目のリスト表示

入出力

IEHPROGM は、下記の入力を使用します。

- 変更するシステム制御データが含まれている 1 つまたは複数のデータ・セット。
- このプログラムの機能を制御するのに使用されるユーティリティー制御ステートメントを含む制御データ・セット。

IEHPROGM は、下記の出力を生成します。

- 変更後のオブジェクト・データ・セットまたはボリューム。
- エラー・メッセージと PASSWORD データ・セットからの情報を含むメッセージ・データ・セット。

アプリケーション・プログラムから IEHPROGM を呼び出す場合は、IEHPROGM を呼び出す前に SVC 99 を出すことによって装置とデータ・セットを動的に割り振ることができます。

IEHPROGM の戻りコードについては、付録 A を参照してください。

制御

IEHPROGM は、ジョブ制御ステートメントおよびユーティリティー制御ステートメントによって制御されます。

ジョブ制御ステートメントは、以下のタスクを実行するために使用されます。

- プログラムを処理またはロードする。
- 制御データ・セットを定義する。
- プログラム実行中に使用するボリュームまたは装置を定義する。
- データ・セットが誤って削除されないようにする。
- ボリュームがプログラムによって完全に処理される前に取り外しされないようにする。
- ユーティリティー制御ステートメントのリスト出力を抑止する。

ユーティリティー制御ステートメントは、プログラムの機能を制御したり、変更するデータ・セットまたはボリュームを定義したりするために使用します。

ジョブ制御ステートメント

表 54 は、IEHPROGM で使用するジョブ制御ステートメントをまとめたものです。

表 54. IEHPROGM で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。

表 54. IEHPROGM で使用するジョブ制御ステートメント (続き)

ステートメント	用途
EXEC	プログラム名 (PGM=IEHPROGM) を指定します。または、プロシージャ・ライブラリーにジョブ制御ステートメントが存在する場合はプロシージャ名を指定します。それ以外にも、PARM 情報を指定することによって、出力リストでの 1 ページの行数を設定したり、ユーティリティー制御ステートメントの出力を抑止したりすることができます。『EXEC ステートメント』を参照してください。
SYSPRINT DD	メッセージに使用する順次データ・セットを定義します。
anyname (任意名) DD	永続的に取り付けられたボリュームまたは永続的に取り付け可能なボリュームを定義します。
SYSIN DD	制御データ・セットを定義します。多くの場合、入力ストリーム中のジョブ制御ステートメントの後に、制御データ・セットを指定します。しかし、これはプロシージャ・ライブラリーのメンバーとして定義することも可能です。

EXEC ステートメント

EXEC ステートメントの構文は次のとおりです。

//[stepname]	EXEC	PGM=IEHPROGM [,PARM=[LINECNT=xx][,PRINT NOPRINT]]
--------------	------	------------------------------------------------------

各項目の意味は以下のとおりです。

PGM=IEHPROGM

IEHPROGM を実行することを指定します。

PARM=[LINECNT=xx] [,PRINT|NOPRINT]

SYSPRINT データ・セットの内容と、SYSPRINT の出力フォーマットを指定します。PARM に複数のサブパラメーターをコーディングする場合、それらのサブパラメーターは括弧または単一引用符で囲んでください。

LINECNT=xx

SYSPRINT データ・セットのリスト出力での 1 ページの行数を指定します。xx の値は 01~99 です。デフォルトは 45 です。

PRINT|NOPRINT

ユーティリティー制御ステートメントを SYSPRINT のリストに含めるかどうかを指定します。ユーティリティー制御ステートメントの出力を抑止すれば、データ・セットに割り当てられたパスワードの機密を保てます。しかし、出力を抑止すると、エラー・メッセージに関係のあるユーティリティー制御ステートメントがそのエラー・メッセージの前に出力されないため、エラー・メッセージの解釈が難しくなることがあります。

デフォルトには、すべての制御ステートメントが出力されます。

SYSPRINT DD ステートメント

SYSPRINT データ・セットのブロック・サイズは、121 の倍数でなければなりません。任意のブロック化因数を指定できます。

anyname (任意名) DD ステートメント

永続的に取り付けられているボリュームまたは永続的に取り付け可能なボリュームのうち、ジョブ・ステップの中で参照されているものについては、それぞれ 1 つずつ DD ステートメントを含める必要があります。それらの DD ステートメントは、真のデータ定義ステートメントとしてではなく、装置割り振りステートメントとして使用されます。

この DD ステートメントは、次のように入力できます。

```
//anyname DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

UNIT パラメーターと VOLUME パラメーターによって、装置タイプとボリューム通し番号を定義します。DISP=OLD を指定することによって、データ・セットが誤って作成されないようにしています。

IEHPROGM は装置割り振り DD ステートメントによって作成される内部制御ブロックを変更するため、DSNAME パラメーターを指定しても、IEHPROGM はそれを無視します。(すべてのデータ・セットは、明示的にまたは暗黙のうちにユーティリティー制御ステートメントによって定義されます。)

注: マルチタスキング環境において、IEHPROGM 実行中に、他のタスクによる装置の動的割り振り / 割り振り解除のために TIOT に対して変更がなされた場合、予測不能の結果になることがあります。

取り付け可能ボリュームの取り付けを据え置くことを指定するには、次のようにコーディングします。

```
//anyname DD VOLUME=(PRIVATE,SER=xxxxxx),
// UNIT=(xxxx,DEFER),DISP=OLD
```

取り付け可能装置を定義する DD ステートメントで、装置類似性を使用することはできません。

SYSIN DD ステートメント

SYSIN データ・セットのブロック・サイズは、80 の倍数でなければなりません。任意のブロック化因数を指定できます。

ユーティリティー制御ステートメント

表 55 は、IEHPROGM で使用するユーティリティー制御ステートメントをまとめたものです。

表 55. IEHPROGM で使用するユーティリティー制御ステートメント

ステートメント	用途
SCRATCH	DASD ボリュームのデータ・セットまたはメンバーをスクラッチします。
RENAME	DASD ボリューム上にあるデータ・セットまたはメンバーの名前または別名を変更します。
CATLG	カタログの索引の中に項目を生成します。
UNCATLG	項目をカタログから除去します。
ADD	PASSWORD データ・セットの項目を追加します。
REPLACE	パスワード項目の中の情報を置換します。

表 55. IEHPROGM で使用するユーティリティ制御ステートメント (続き)

ステートメント	用途
DELETEP	PASSWORD データ・セットの項目を削除します。
LIST	パスワード項目内の情報をフォーマットし、リスト表示します。

ユーティリティ制御ステートメントの継続のための要件については、9 ページの『ユーティリティ制御ステートメントの継続』を参照してください。

SCRATCH ステートメント

SCRATCH ステートメントは、DASD ボリュームからデータ・セットまたはメンバーをスクラッチする 場合に使用します。データ・セットまたはメンバーは、SCRATCH ステートメントの中で指定されているボリュームからしかスクラッチされません。この機能が、スクラッチされたデータ・セットのカatalog項目を削除することはありません。

並行して実行されているプログラムによってデータ・セットまたはボリュームが使用されている場合、SCRATCH 操作は処理されません。DD ステートメントの『DISP=OLD』は、単にデータ・セットが誤って削除されないようにするだけです。それは、ジョブ・ステップ実行中のデータ・セットの排他使用を保証するものではありません。区分データ・セットのメンバーをスクラッチする場合、そのデータ・セットが使用中でないことを自分で確認しておく必要があります。

マルチボリューム・データ・セットの場合、指定されたすべてのボリュームがオンラインでなければなりません。

ストレージ管理サブシステム (SMS) で管理されるデータ・セットをスクラッチする場合、VOL パラメーターの装置タイプとボリューム・リストは、そのデータ・セットに実際に割り振られているボリュームを反映するものでなければなりません。SMS 管理データ・セットの作成時にボリュームを指定しても、SMS はそのボリュームにデータ・セットを自動的に割り振るわけではありません。

SMS 管理データ・セットをスクラッチする場合、IEHPROGM はそのデータ・セットをアンカタログします。

SCRATCH ステートメントの構文は次のとおりです。

[label]	SCRATCH	{VTOCIDNAME=name} ,VOL=device=(list) [,PURGE] [,MEMBER=name] [,SYS]
---------	---------	---------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

VTOC

指定したボリューム上のすべてのデータ・セットをスクラッチすることを指定します。ただし、次のものは除きます。

- パスワードによって保護されているデータ・セット
- まだ有効期限が経過していないデータ・セット

- 索引付き VTOC の索引が含まれているデータ・セット

パスワード保護データ・セットは、正しいパスワードを指定した場合にスクラッチされます。

PURGE または SYS を指定した場合、VTOC の効果は変更されます。

DSNAME=name

スクラッチするデータ・セットまたはスクラッチするメンバーを含む区分データ・セットの完全修飾名を指定します。名前は、区切り文字を含めて 44 バイト以下です。

VOL=device=(list)

データ・セットを含むボリュームの装置タイプと通し番号を指定します (50 個以下)。1 個の通し番号だけを指定した場合、それを括弧で囲む必要はありません。複数の通し番号は、互いにコンマで区切る必要があります。

VTOC または MEMBER を指定した場合、VOL に複数のボリュームを指定することはできません。VOL にシステム常駐ボリュームを指定した場合、VTOC の指定には十分の注意が必要です。

PURGE/NOPURGE

DSNAME または VTOC によって指定されるデータ・セットをスクラッチするかどうかを指定します。

PURGE または NOPURGE キーワードが指定されていても、SMS 管理データ・セットについては無視されます。SYS1.PARMLIB の IGDSMSxx メンバーで OVRD_EXPDT が指定されている場合、PURGE および NOPURGE キーワードは考慮されません。有効期限が切れているかどうかに関係なく、データ・セットは常に削除されます。このことが当てはまるのは、データ・セットが DASD データ・セットで、かつ SMS 管理の場合だけです。SMS 管理以外のデータ・セットの場合、PURGE および NOPURGE キーワードが使用されていれば、それはそのまま変更なしで使用されます。

MEMBER=name

区分データ・セットのディレクトリーから削除する (名前付きデータ・セット内の) メンバーのメンバー名または別名を指定します。この名前が有効であるかどうかに関係なく、すべてのメンバーはアクセス可能でなければならないため、この名前の妥当性検査は実行されません。

デフォルト: name によって指定されるデータ・セットまたはボリュームの全体がスクラッチされます。

SYS

一時データ・セットだけを削除するように、SCRATCH VTOC のアクションを制限します。これは、JCL 上でアンパーサンドから始まる名前をコーディングされたデータ・セットを意味します。以下に例を示します。

```
DSN=&LIB
```

一時データ・セットの名前は、

『AAAAAAAA.AAAAAAAAA.AAAAAAAAA.AAAAAAAAA.』または『SYSnnnnn.T』で始まり、その位置 19 は『F』、『V』、または『A』です。それらは、オペレーティング・システムによってデータ・セットに割り当てられる名前です。

データ・セットの名前がこのフォームであれば、多くの場合、それは、通常のステップまたはジョブの終了時には削除されない一時データ・セットです。nnnnn は、そのデータ・セットが yyddd のフォーマットで作成された日付です。

SYS パラメーターは、VTOC を指定した場合にのみ有効です。

RENAME ステートメント

RENAME ステートメントは、DASD ボリュームにあるデータ・セットまたはメンバーの実名または別名を変更する場合に使用します。名前は、指定されたボリューム上のものだけが変更されます。名前変更操作を実行しても、カタログは更新されません。

並行して実行されているプログラムによってデータ・セットまたはボリュームが使用されている場合、RENAME 操作は処理されません。区分データ・セットのメンバーの名前を変更する場合、そのデータ・セットが使用中でないことを自分で確認しておく必要があります。

マルチボリューム・データ・セットの場合、指定されたすべてのボリュームがオンラインでなければなりません。

MEMBER パラメーターを指定しない場合、データ・セットの全体が名前変更されます。

ストレージ管理サブシステム (SMS) で管理されるデータ・セットを名前変更する場合、VOL パラメーターの装置タイプとボリューム・リストは、そのデータ・セットに実際に割り振られているボリュームを反映するものでなければなりません。SMS 管理データ・セットの割り振り時にボリュームを指定した場合、SMS はそのボリュームにデータ・セットを自動的に割り振るわけではありません。

SMS 管理データ・セットの名前を変更する場合、IEHPROGM はそのデータ・セットをアンカカタログしてから、そのデータ・セットを新しい名前で再カタログします。再カタログが必要なのに、別名のためにそれを実行できない場合、IEHPROGM はデータ・セットの名前を変更しません。

RENAME ステートメントの構文は次のとおりです。

[label]	RENAME	DSNAME= <i>name</i> ,VOL= <i>device=(list)</i> ,NEWNAME= <i>name</i> [,MEMBER= <i>name</i>]
---------	--------	-------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

DSNAME=*name*

名前変更するデータ・セットまたは名前変更するメンバーを含む区分データ・セットの完全修飾名を指定します。名前は、区切り文字を含めて 44 バイト以下です。

VOL=*device=(list)*

データ・セットを含むボリュームの装置タイプと通し番号を指定します (50 個

以下)。1 個の通し番号だけを指定した場合、それを括弧で囲む必要はありません。複数の通し番号は、互いにコンマで区切る必要があります。

MEMBER を指定した場合、VOL に複数のボリュームを指定することはできません。

NEWNAME=name

データ・セットまたは新しいメンバーの新しい完全修飾名または別名を指定します。

MEMBER=name

名前変更する (名前付きデータ・セット内の) メンバーのメンバー名または別名を指定します。この名前が有効であるかどうかに関係なく、すべてのメンバーはアクセス可能でなければならないため、この名前の妥当性検査は実行されません。

デフォルト: name によって指定されるデータ・セットまたはボリュームの全体が変更されます。

CATLG および UNCATLG ステートメント

CATLG ステートメントは、カタログ内に非 VSAM 項目を生成する場合に使用します。

UNCATLG ステートメントは、カタログから非 VSAM 項目を削除する場合に使用します。

SMS 管理データ・セットをカタログまたはアンカタログするのに、IEHPROGM を使用することはできません (名前変更とスクラッチを除く)。IEHPROGM を使用できるのは、SMS 管理以外の非 VSAM データ・セットをカタログまたはアンカタログする場合だけです。SMS 管理データ・セットをカタログまたはアンカタログするには、*z/OS DFSMS カatalogの管理* および *z/OS DFSMS カatalogのためのアクセス方式サービス・プログラム* を参照してください。

CATLG および UNCATLG ステートメントの構文は、次のとおりです。

[label]	{CATLG UNCATLG}	DSNAME=name ,VOL=device={(list) (serial,seqno[,...])}
注: VOL を指定できるのは、CATLG を指定した場合だけです。		

各項目の意味は以下のとおりです。

DSNAME=name

カタログまたはアンカタログするデータ・セットの完全修飾名を指定します。名前は、区切り文字を含めて 44 バイト以下です。

VOL=device={(list)|(serial,seqno[,...])}

カタログにカタログするデータ・セットを含むボリュームの装置タイプ、通し番号、およびデータ・セット・シーケンス番号 (テープ・ボリュームの場合) を指定します (50 個以下)。

device には、常に総称装置名を使用してください (たとえば 3390)。

ボリューム通し番号は、データ・セット作成時に入カストリーム中の DD ステートメントで指定された順序と同じ順序で指定する必要があります。複数の通し番号は、互いにコンマで区切る必要があります。

seqno は、磁気テープ上のデータ・セットの場合にのみ有効です。この値が意味を持つのは、最初のボリューム通し番号に対してだけです。z/OS V1R5 またはそれ以降で指定可能な最大値は、65535 です。このカタログ・エントリーを z/OS V1R5 よりも前のレベルのシステム上で使用したい場合、9999 を超える値を指定しないでください。ファイル・シーケンス番号の詳細は、「z/OS DFSMS Using Magnetic Tapes」を参照してください。

VOL を指定できるのは、CATLG を指定した場合だけです。

ADD (パスワードの追加) および REPLACE (パスワードの置換) ステートメント

ADD ステートメントは、PASSWORD データ・セットの項目を追加する場合に使用します。オンライン DASD データ・セットの制御項目を追加した場合、そのデータ・セットに関して指示された保護状況が DSCB の中に設定されます。2 次項目を追加した場合、DSCB の中の保護状況は変更されません。

REPLACE ステートメントは、パスワード項目の中で、パスワード名、パスワードの保護モード (読み取り/書き込みかそれとも読み取り専用か)、およびユーザー・データのいずれかまたはすべての情報を置換する場合に使用します。オンライン DASD データ・セットの制御項目を置換した場合、必要に応じて DSCB に記録されているそのデータ・セットの保護状況が変更されます。2 次項目を置換した場合、DSCB の中の保護状況は変更されません。

ADD および REPLACE ステートメントの構文は、次のとおりです。

[label]	{ADD REPLACE}	DSNAME=name [,PASSWORD1=current-password] [,PASSWORD2=new-password] [,CPASSWORD=control-password] [,TYPE=code] [,VOL=device=(list)] [,DATA='user-data']
注: PASSWORD1 を指定できるのは、REPLACE を指定した場合だけです。		

各項目の意味は以下のとおりです。

DSNAME=name

パスワード項目を追加または変更するデータ・セットの完全修飾名を指定します。名前は、区切り文字を含めて 44 バイト以下です。

PASSWORD1=current-password

変更する項目のパスワードを指定します。PASSWORD1 を指定しない場合、現在のパスワードを入力するようオペレーターに対して指示が出されます。PASSWORD1 を指定できるのは、REPLACE を指定した場合だけです。

PASSWORD2=new-password

項目に追加または割り当てる新しいパスワードを指定します。パスワードを変更

しない場合であっても、新しいパスワードとして現在のパスワードを指定する必要があります。パスワードは 1~8 バイトの英数字です。PASSWORD2 を指定しない場合、新しいパスワードを入力するようオペレーターに対して指示が出されます。

CPASSWORD=control-password

データ・セットの制御パスワードを指定します。

ADD の場合、CPASSWORD の指定は必須です。ただし、これがそのデータ・セットに割り当てる最初のパスワードの場合は、PASSWORD2 で追加するパスワードを指定します。

REPLACE の場合、CPASSWORD の指定は必須です。ただし、制御項目を変更している場合は、PASSWORD1 で制御パスワードを指定します。

TYPE=code

パスワードの保護コードを指定します。また、BDAM オンライン・データ・セットについて制御パスワード項目を変更する場合または割り当てる場合は、データ・セットの保護状況を指定します。code に指定できる値は、次のとおりです。

- 1 このパスワードによって、データ・セットへの読み取り/書き込みアクセスが許可されることを指定します。制御パスワードを割り当てている場合または変更している場合は、DSCB の中に読み取り/書き込み保護が設定されます。
- 2 このパスワードによって、データ・セットへの読み取り専用アクセスが許可されることを指定します。制御パスワードを割り当てている場合または変更している場合は、DSCB の中に読み取り/書き込み保護が設定されます。
- 3 このパスワードによって、データ・セットへの読み取り/書き込みアクセスが許可されることを指定します。制御パスワードを割り当てている場合または変更している場合は、DSCB の中に書き込み保護が設定されます。

デフォルト: ADD の場合にこのパラメーターを省略すると、新しいパスワードには、そのデータ・セットの制御パスワードと同じ保護コードが割り当てられます。制御パスワードを『追加』している場合のデフォルトは TYPE=3 です。

REPLACE の場合、保護は変更されません。

VOL=device=(list)

データ・セットを含むボリュームの装置タイプと通し番号を指定します (50 個以下)。1 個の通し番号だけを指定した場合、それを括弧で囲む必要はありません。複数の通し番号は、互いにコンマで区切る必要があります。

これを省略した場合、データ・セットがカタログされていてオンラインになっているのでない限り、DSCB の中の保護状況は設定または変更されません。2 次パスワード項目の場合、または DSCB の中に希望する保護状況がすでに設定されている場合、あるいは変更しない場合、このパラメーターは不要です。

DATA='user-data'

パスワード項目の中に入れるユーザー・データを指定します。ユーザー・データは、長さが 77 バイト以下であり、アポストロフィで囲む必要があります。ユーザー・データ内に含まれる他のアポストロフィがある場合、それはそれぞれ単一アポストロフィを 2 回連続して入力する必要があります。

ADD 操作で DATA を省略した場合、77 バイトのブランクが使用されます。
REPLACE 操作で DATA を省略した場合、現在のユーザー・データは変更されません。

DELETEP (パスワードの削除) ステートメント

DELETEP ステートメントは、PASSWORD データ・セットの項目を削除するのに使います。制御項目を削除すると、そのデータ・セットの 2 次項目もすべて削除されます。2 次項目を削除する場合は、その項目だけが削除されます。オンライン DASD データ・セットの制御項目を削除する場合、そのデータ・セットが保護されなくなったことを示すように DSCB 中の保護状況が設定されます。

DELETEP ステートメントの構文は次のとおりです。

[label]	DELETEP	DSNAME=name [,PASSWORD1=current-password] [,CPASSWORD=control-password] [,VOL=device=(list)]
---------	---------	-------------------------------------------------------------------------------------------------------

各項目の意味は以下のとおりです。

DSNAME=name

パスワード項目を削除するデータ・セットの完全修飾名を指定します。名前は、区切り文字を含めて 44 バイト以下です。

PASSWORD1=current-password

削除する項目のパスワードを指定します。PASSWORD1 を指定しない場合、現在のパスワードを入力するようオペレーターに対して指示が出されます。

CPASSWORD=control-password

CPASSWORD の指定は必須です。ただし、制御項目を削除している場合は、PASSWORD1 で制御パスワードを指定します。

VOL=device=(list)

データ・セットを含むボリュームの装置タイプと通し番号を指定します (50 個以下)。1 個の通し番号だけを指定した場合、それを括弧で囲む必要はありません。複数の通し番号は、互いにコンマで区切る必要があります。

これを省略した場合、データ・セットがカタログされていてオンラインになっているのでない限り、DSCB 中の保護状況は変更されません。2 次パスワード項目の場合、または DSCB 中に希望する保護状況がすでに設定されている場合、このパラメーターは不要です。

LIST (パスワード情報のリスト表示) ステートメント

LIST ステートメントは、パスワード項目の情報をフォーマットして出力する場合に使用します。

LIST ステートメントの構文は次のとおりです。

[label]	LIST	DSNAME=name ,PASSWORD1=current-password
---------	------	--------------------------------------------

各項目の意味は以下のとおりです。

DSNAME=name

パスワード項目をリスト表示するデータ・セットの完全修飾名を指定します。名前は、区切り文字を含めて 44 バイト以下です。

PASSWORD1=current-password

リスト表示する項目のパスワードを指定します。PASSWORD1 を指定しない場合、現在のパスワードを入力するようオペレーターに対して指示が出されます。

IEHPROGM の例

以下に、IEHPROGM の使用例をいくつか示します。ここに示す IEHPROGM の例のための早見表として、表 56 を使用することができます。「例」欄の数字は、この後の例の番号です。

表 56. IEHPROGM の例の一覧

操作	マウント・ボリューム	注釈	例
BLDG	ディスク	JCL によって、新しい世代別データ・グループ索引を作成し、更新します。モデル DSCB を作成します。新しい世代を追加します。	10
BLDG、RENAME、CATLG	ディスク	世代別データ・グループ索引を作成します。3 つのデータ・セットの名前を変更し、索引に入れます。	9
LIST、REPLACE	ディスク	パスワード項目をリスト表示します。保護モードと状況を変更し、ユーザー・データを追加します。	6
RENAME	ディスク	区分データ・セットのメンバーの名前を変更します。	7
RENAME、DELETEP、ADD	ディスク	データ・セットの名前を変更します。変更前のパスワードを削除し、新しいパスワードを割り当てます。	5
RENAME、UNCATLG、CATLG	ディスク	2 つの取り付け可能装置上で 1 つのデータ・セットの名前を変更します。変更前のデータ・セット名を除去します。データ・セットをその新しい名前でカタログします。	3
SCRATCH	ディスク	データ・セットの DSCB をスクラッチします。	1
SCRATCH、UNCATLG	ディスク	2 つのデータ・セットをスクラッチし、カタログからそれらに対応する項目を削除します。	2
UNCATLG	ディスク	3 つの世代別データ・セットの索引構造をカタログから削除します。	4

実際の装置名または装置番号を指定するべきところで **disk** または **tape** を使用している例は、使用する前に変更する必要があります。実際の装置名または装置番号は、システムに対する装置定義の方法によって異なります。

例 1: 一時システム・データ・セットのスクラッチ

この例では、すべての一時システム・データ・セットをボリューム目録からスクラッチします。

```
//SCRVTOC JOB ...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD2 DD UNIT=disk,VOLUME=SER=222222,DISP=OLD
//SYSIN DD *
SCRATCH VTOC,VOL=disk=222222,SYS
/*
```


各制御ステートメントについての説明は以下のとおりです。

- DD2 ステートメントでは、ボリュームを定義しています。システム常駐ボリュームが参照されていないため、ボリュームを定義するための DD ステートメントが必要です。
- SYS を指定した SCRATCH ステートメントでは、指定したボリュームから、有効期限を過ぎている一時的システム・データ・セットをすべてスクラッチしています。

例 2: 2 つのデータ・セットのスクラッチとアンカタログ

この例では、2 つのデータ・セットをスクラッチします。SET1 と A.B.C.D.E をボリューム 222222 からスクラッチします。それら 2 つのデータ・セットをアンカタログします。

```
//SCRDSETS JOB ...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,DISP=OLD,VOLUME=SER=222222
//SYSIN DD *
SCRATCH DSNAME=SET1,VOL=disk=222222
UNCATLG DSNAME=SET1
SCRATCH DSNAME=A.B.C.D.E,VOL=disk=222222
UNCATLG DSNAME=A.B.C.D.E
/*
```

各ユーティリティー制御ステートメントについての説明は以下のとおりです。

- 最初の SCRATCH ステートメントでは、ボリューム 222222 にある SET1 をスクラッチすることを指定しています。
- 最初の UNCATLG ステートメントでは、SET1 をアンカタログすることを指定しています。
- 第 2 の SCRATCH ステートメントでは、ボリューム 222222 にある A.B.C.D.E をスクラッチすることを指定しています。
- 第 2 の UNCATLG ステートメントでは、A.B.C.D.E をアンカタログすることを指定しています。

例 3: マルチボリューム・データ・セット・カタログの名前変更

この例では、2 つの取り付け可能なボリュームにおいて、データ・セットの名前を変更します。変更前のデータ・セット名を削除し、データ・セットを新しいデータ・セット名でカタログします。

```
//RENAMEDS JOB ...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD VOLUME=SER=111111,UNIT=disk,DISP=OLD
//DD2 DD UNIT=(disk,,DEFER),DISP=OLD,
// VOLUME=(PRIVATE,SER=(222222,333333))
//SYSIN DD *
RENAME DSNAME=A.B.C,NEWNAME=NEWSET,VOL=disk=(222222,333333)
UNCATLG DSNAME=A.B.C
CATLG DSNAME=NEWSET,VOL=disk=(222222,333333)
/*
```

各制御ステートメントについての説明は以下のとおりです。

- RENAME では、ボリューム 222222 および 333333 にあるデータ・セット A.B.C の名前を NEWSET に変更するよう指定しています。
- UNCATLG では、データ・セット A.B.C をアンカタログすることを指定しています。
- CATLG では、ボリューム 222222 および 333333 にある NEWSET をカタログするよう指定しています。

例 4: 3 つのデータ・セットのアンカタログ

この例では、A.B.C.D.E.F.SET1、A.B.C.G.H.SET2、および A.B.I.J.K.SET3 という 3 つのデータ・セットをアンカタログします。

```
//DLTSTRUC JOB ...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//SYSIN DD *
        UNCATLG DSNAME=A.B.C.D.E.F.SET1
        UNCATLG DSNAME=A.B.C.G.H.SET2
        UNCATLG DSNAME=A.B.I.J.K.SET3
/*
```

各制御ステートメントについての説明は以下のとおりです。

- UNCATLG ステートメントでは、データ・セット A.B.C.D.E.F.SET1、A.B.C.G.H.SET2、および A.B.I.J.K.SET3 をアンカタログすることを指定しています。

例 5: データ・セットの名前変更と新しいパスワードの定義

この例では、データ・セットの名前を変更します。変更前のデータ・セット名に割り当てられていたデータ・セット・パスワードは削除します。その後、新しいデータ・セット名に 2 つのパスワードを割り当てます。そのデータ・セットがカタログされていない場合、LOCATE マクロ命令が異常終了したことを示すメッセージが出されます。

```
//ADDPASS JOB ...
//STEP1 EXEC PGM=IEHPROGM,PARM='NOPRINT'
//SYSPRINT DD SYSOUT=A
//DD1 DD VOLUME=SER=222222,DISP=OLD,
// UNIT=disk
//SYSIN DD *
        RENAME DSNAME=OLD,VOL=disk=222222,NEWNAME=NEW
        DELETEP DSNAME=OLD,PASSWORD1=KEY
        ADD DSNAME=NEW,PASSWORD2=KEY,TYPE=1, X
            DATA='SECONDARY IS READ'
        ADD DSNAME=NEW,PASSWORD2=READ,CPASSWORD=KEY,TYPE=2, X
            DATA='ASSIGNED TO J. DOE'
/*
```

各ユーティリティー制御ステートメントについての説明は以下のとおりです。

- RENAME では、OLD というデータ・セットの名前を NEW に変更することを指定しています。旧データ・セット名を変更するには、オペレーターがパスワードを提供する必要があります。
- DELETEP は、パスワード KEY の項目を削除することを指定しています。この例の KEY は制御パスワードなので、このデータ・セット名のすべてのパスワード項目が削除されます。DSCB の中で設定されているデータ・セットの保護状況

は変更しないため、VOL パラメーターは不要です。現在、DSCB の中で読み取り/書き込み保護が設定されており、新しいデータ・セット名でパスワードを再び割り当てる場合は、読み取り/書き込み保護が適しています。

- ADD ステートメントでは、パスワード KEY と READ のための項目を追加するよう指定しています。KEY が制御パスワードとなり、これはデータ・セットに対する読み取りアクセスと書き込みアクセスの両方を許可します。READ は 2 次パスワードとなり、これはデータ・セットに対する読み取りアクセスを許可します。DSCB の中でデータ・セットの保護状況はまだ設定されているため、VOL パラメーターは不要です。

例 6: パスワード情報のリスト表示と置換

この例では、パスワード項目の情報をリスト表示します。その後、パスワードの保護モード、データ・セットの保護状況、およびユーザー・データを変更します。

72

```
//REPLPASS JOB ...
//STEP1 EXEC PGM=IEHPROGM,PARM='NOPRINT'
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD VOLUME=(PRIVATE,SER=(222222,333333)),
// UNIT=(disk,,DEFER),DISP=OLD
//SYSIN DD *
LIST DSNAME=A.B.C,PASWORD1=ABLE
REPLACE DSNAME=A.B.C,PASWORD1=ABLE,PASWORD2=ABLE,TYPE=3, X
VOL=disk=(222222,333333), X
DATA='NO SECONDARIES; ASSIGNED TO DEPT 31'
/*
```

各ユーティリティー制御ステートメントについての説明は以下のとおりです。

- LIST では、パスワード ABLE のための項目のうち、アクセス・カウンター、保護モード、およびユーザー・データをリスト表示します。項目をリスト表示することによって、アクセス・カウンターが REPLACE によって 0 にリセットされる前に、アクセス・カウンターの内容を記録しておくことが可能になります。
- REPLACE では、パスワード ABLE の保護モードを、読み取り/書き込みアクセスを許可するように変更するよう指定し、さらにデータ・セットの保護状況を書き込み保護に変更しています。データ・セットの保護状況が変更され、この例の場合はデータ・セットがカタログされていないため、VOL パラメーターが必要です。これは制御パスワードなので、CPASSWORD パラメーターは不要です。

例 7: 区分データ・セットのメンバーの名前変更

この例では、区分データ・セットのメンバーの名前を変更します。

```
//REN JOB ...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD VOL=SER=222222,DISP=OLD,UNIT=disk
//SYSIN DD *
RENAME VOL=disk=222222,DSNAME=DATASET,NEWNAME=BC,MEMBER=ABC
/*
```

各制御ステートメントについての説明は以下のとおりです。

- DD1 DD では、永続的に取り付けられるボリュームを定義しています。
- SYSIN DD では、入力ストリームの中のそれ以降の入力データ・セットを定義しています。

- RENAME では、ディスク・ボリューム上にある区分データ・セット DATASET のメンバー ABC の名前を BC に変更することを指定しています。

第 15 章 IFHSTATR (ESV データのリスト表示) プログラム

IFHSTATR は、タイプ 21 SMF (システム管理機能) レコードの情報をフォーマットして印刷するためのシステム・ユーティリティです。この種のレコードは、ボリューム別エラー統計 (ESV) データを提供します。

図 46 は、タイプ 21 レコードのフォーマットを示しています。

オフセットおよび長さ (バイト):

0	1	2	3	4
0 (X'00')	合計レコード長 (62)		記述	
4 (X'04')	システム標識	レコード タイプ (21)	時間	
8 (X'08')	時間 (続き)		現在日付	
12 (X'0C')	現在日付 (続き)		システム識別番号	
16 (X'10')	システム識別番号 (続き)		このフィールドを含めた、 レコードの残りの長さ (44)	
20 (X'14')	ボリューム通し番号			
24 (X'18')	ボリューム通し番号 (続き)		チャンネル装置アドレス	
28 (X'1C')	UCB タイプ			
32 (X'20')	読み取り一時エラー	書き込み一時エラー	入出力開始	
36 (X'24')	読み取り永続エラー	書き込み永続エラー	ノイズ・ブロック	ギャップの消去
40 (X'28')	ギャップの消去 (続き)	クリーナー・アクション		テープ・フォーマット
44 (X'2C')	ブロック・サイズ		DCBOFLAG	テープ装置通し番号
48 (X'30')	テープ装置通し番号 (続き)		正方向読み取り一時エラー	
52 (X'34')	逆方向読み取り一時エラー		書き込み一時エラー	
56 (X'38')	4K バイト単位読み取り数			4K バイト単位書き込み数
60 (X'3C')	4K バイト単位書き込み数 (続き)			
62 (X'3E')				

図 46. SMF タイプ 21 (ESV) レコード・フォーマット: OS/390 リリース 10 からは、レコードの長さは 80 バイトになります。内容に関しては、Macro IGESMF21 および「システム管理機能」を参照してください。

ライブラリー内の磁気テープの品質評価

SMF によって ESV レコード内に収集された統計は、磁気テープ・ライブラリーの品質を評価するためにとっても役立ちます。IFHSTATR を使えば、ESV レコードを日付 / 時刻順に印刷できます。また、ESV レコードをボリューム通し番号順、装置アドレス順、またはエラー発生順にソートすれば、ライブラリーの状態を分析するのに役立ちます。

IFHSTATR 報告書は、劣化したメディア (磁気テープ) を識別するのに役立ちますが、時には、特定の磁気テープ・ドライブのパフォーマンス低下を識別できることもあります。磁気テープ装置通し番号を使って、その磁気テープに書き込んだテープ・ドライブを識別できます。

ESV レコードが SMF データ・セットに書き込まれるのは、以下の場合です。

1. ボリュームが取り外された場合
2. ボリュームが DDR を介して取り外された場合
3. 磁気テープ・ドライブがオフラインの場合
4. HALT EOD コマンドが実行された場合
5. EREP が実行された場合

ESV レコードは取り外し時以外にも書き込まれることがあるので、ボリュームの取り付け時に複数のレコードが書き込まれることもあります。したがって、1 つのボリュームのレコード数を根拠にして、ボリュームの取り付け数または使用数を判定することはできません。

入出力

IFHSTATR は、磁気テープ・ボリュームのエラーや使用状況に関する情報が入った ESV レコードを、入力として使用します。ESV レコードが見つからない場合は、出力データ・セットにメッセージが書き込まれます。ESV レコードの長さが 62 バイトまたは 80 バイトでない場合は、INVALID TYPE 21 RECORD というメッセージが印刷されます。

IFHSTATR を使用して ESV レコードからデータを取り出す前に、IFASMFDP を実行して、VSAM の SYS1.MANX または SYS1.MANY を物理順次編成に変換できます。IFASMFDP を実行するための詳細については、*z/OS MVS システム管理機能 (SMF)* を参照してください。

IFHSTATR では、ESV レコードからの情報を集めた出力データ・セットが作成されます。出力は、121 バイト非ブロック化レコードの形式になり、各レコードの最初のバイトが米国規格協会 (ANSI) の制御文字になります。

入力データ・セットのブロック・サイズは、32760 バイトを超えてはなりません。ただし、SMF レコードで表現される磁気テープの場合は、ブロック・サイズが 32760 バイトを超えても構いません。

タイプ 21 レコードのブロック・サイズ・フィールドまたは SIO カウント・フィールドの使用が 99,999 を超えた場合、IFHSTATR は、出力を調整します。たとえば、フィールドが 99,999 より大きく、1,000,000 より小さい場合は、1,000 の倍数に調整されて、“K” の文字が追加されます。フィールドが 999,999 より大きい場合は、1,000,000 の倍数に調整されて、“M” の文字が追加されます。この場合の “K”

と "M" の意味は、DD ステートメント上の BLKSIZE 値における "K" と "M" の意味とは異なることに注意してください。DD ステートメントでは、それぞれ、1024 と 1048576 の倍数を意味します。

図 47 は、IFHSTATR によって印刷される出力のサンプルです。印刷出力のフィールドの説明については、サンプルの後の凡例を参照してください。

MAGNETIC TAPE ERROR STATISTICS BY VOLUME 99172																		
VOLUME SERIAL	DATE	TIME OF DAY	DEV ADR	T/U SER	MODE	BLOCK SIZE	TAPE FORM	TEMP READ	TEMP READB	TEMP WRITE	PRM RD	PRM WRT	NOISE BLOCK	ERASE GAPS	CLEAN ACTS	USAGE S10	MBYTES READ	MBYTES WRITTEN
INVALID TYPE 21 RECORD																		
T34201	99172	08:04:22	0180	00000	OUT	N/A	N/A	1	N/A	2	3	4	5	6	7	8	N/A	N/A
T34202	99172	12:01:59	0281	56789	OUT	80	1600	1	N/A	2	3	4	5	6	7	8	N/A	N/A
T34200	99172	12:02:18	028C	67890	RB	32768	6250	255	N/A	255	255	255	255	65535	65535	65535	N/A	N/A
T34200	99172	12:03:21	0480	78901	RB	80	N/A	1	2	3	4	5	N/A	6	7	8	9	10
T34201	99172	12:04:21	0480	89012	RF	65535	N/A	65535	65535	65535	255	255	N/A	65535	65535	65535	65535	65535
T35901	99172	14:35:41	09A0	72310	RF	2147M	N/A	0	0	0	0	1	N/A	0	0	999K	0	0

* T342000 IS A 3420 WITH SMALL NUMBER OF ERRORS WITH BLOCKSIZE/DENSITY NOT AVAILABLE																		
* T342001 IS A 3420 WITH SMALL NUMBER OF ERRORS																		
* T342002 IS A 3420 WITH MAXIMUM NUMBER OF ERRORS																		
* T348000 IS A 3480 WITH SMALL NUMBER OF ERRORS																		
* T348001 IS A 3480 WITH MAXIMUM NUMBER OF ERRORS																		
* T359001 IS A 3590 WITH SJO COUNT GREATER THAN 99,999 BUT LESS THAN 1,000,000 AND A BLOCKSIZE GREATER THAN 999,999																		

図 47. IFHSTATR による出力のサンプル

凡例

TIME OF DAY	ESV レコードが書き込まれた時刻。
DEV ADR	磁気テープが取り付けられた磁気テープ・ドライブの装置アドレス。
T/U SER	磁気テープに書き込んだ磁気テープ・ドライブの通し番号。この番号は、入力用磁気テープがある場合は、その磁気テープ・ラベルから得られます。
MODE ¹	アクセスするデータ・セットの OPEN フラグ・ビット。 <ul style="list-style-type: none"> • OUT = OPENED (出力の場合) • RF = OPENED (順方向読み取り入力の場合) • RB = OPENED (逆方向読み取り入力の場合)
BLOCKSIZE ¹	最後にアクセスしたデータ・セットのブロック・サイズ。
TAPE FORMAT ¹	磁気テープの記録形式。
TEMP READ	正常に再試行できた読み取りデータ・チェックの数。
TEMP READB ²	逆方向読み取りコマンドで正常に再試行できた読み取りデータ・チェックの数。
TEMP WRITE	正常に再試行できた書き込みデータ・チェックの数。
PERM RD	正常に再試行できなかった読み取りデータ・チェックの数。
PERM WRT	正常に再試行できなかった書き込みデータ・チェックの数。
NOISE BLOCK	(NRZI のみ) 読み取りバイト数が 12 未満だった読み取りデータ・チェックの数。
ERASE GAPS	エラー・リカバリー中にギャップ消去コマンドが実行された回数。ギャップ消去コマンドは、書き込みデータ・チェックの再試行の前に実行されます。
CLEAN ACTS	読み取りデータ・チェック・リカバリー中に、磁気テープがクリーナー・ブレード上で移動した回数。磁気テープの移動は、元の読み取りコマンドの再試行が 4 回行われるごとに発生するのが普通です。

凡例

USAGE SIO	完了したチャンネル・プログラムの数 (ERP が開始したチャンネル・プログラムは含まれない)。チャンネル・プログラムの CCW の数は無制限なので、この値は読み取りまたは書き込みの数とは一致しないことがあります。
MBYTES READ ²	読み取られたメガバイト数。
MBYTES WRITTEN ²	書き込まれたメガバイト数。

注:

- 1 データは DCB または DCBE に由来するので、存在しない場合もあります。
- 2 バッファ磁気テープ装置の場合だけに限定されます。

制御

IFHSTATR は、ジョブ制御ステートメントによって制御されます。ユーティリティ制御ステートメントは使用しません。

表 57 は、IFHSTATR で使用するジョブ制御ステートメントをまとめたものです。

表 57. IFHSTATR で使用するジョブ制御ステートメント

ステートメント	用途
JOB	ジョブを開始します。
EXEC	プログラム名を指定します (PGM=IFHSTATR)。
SYSUT1 DD	入力データ・セット、および入力データ・セットを入れる装置を定義します。DSNAME および DISP パラメーターは必須です。IBM 標準ラベルがない磁気テープ装置の場合は、LABEL および DCB パラメーターが必要です。また、データ・セットがカタログ化されていない場合は、UNIT および VOLUME パラメーターが必要です。
SYSUT2 DD	出力を書き込む順次データ・セットを定義します。

IFHSTATR の例

この例では、IFHSTATR を使用してタイプ 21 SMF レコードを印刷します。

```
//REPORT JOB ...
//STEP1 EXEC PGM=IFHSTATR
//SYSUT1 DD UNIT=3480,DSNAME=SYS1.MAN,LABEL=(,SL),
//          VOL=SER=valid,DISP=OLD
//SYSUT2 DD SYSOUT=A
/*
```

出力データ・セットは、BSAM がサポートしている任意の出力装置に入れることができます。

注: 入力の LRECL および BLKSIZE パラメーターは、IFHSTATR では指定しません。この情報は、SYSUT1 DD ステートメントの DCB パラメーター、または磁気テープ・ラベルから得られます。

付録 A. アプリケーション・プログラムからユーティリティー・プログラムを起動する

この付録には、DFSMS が提供しているプログラミング・インターフェースとそれに関連する情報を記述しています。

この付録の目的は、アプリケーション・プログラムからユーティリティー・プログラムを起動する手助けをすることです。

アプリケーション・プログラムからユーティリティー・プログラムを起動するには、LINK マクロ命令を使用します。(ATTACH も使用できますが、パラメーターを追加しなければなりません。詳細については、*z/OS MVS プログラミング: アセンブラー・サービス 解説書 ABE-HSP* を参照してください。)

EXEC ステートメントの PARM パラメーターで通常指定する情報のほかに、ユーティリティーの実行対象データ・セットを定義する非標準 DD 名も指定する必要があります。

注: すべてのパラメーターは、16MB 境界の下に入れなければなりません (つまり、24 ビット・アドレスを持ちます)。LINK パラメーターの詳細については、*z/OS MVS プログラミング: アセンブラー・サービス 解説書 ABE-HSP* を参照してください。

アプリケーション・プログラムから IEBCOMPR、IEBCOPY、IEBDG、IEBGENER、IEBPTPCH、IEHLIST、IEHMOVE、IEHPROGM のいずれかを起動するか、TSO CALL コマンドを起動する場合は、ユーティリティーを呼び出す前に SVC 99 を出して装置を動的に割り振るか、JCL または TSO ALLOCATE の同等機能を使用する必要があります。

IEBCOPY、IEHINIT、IEHMOVE、および IEHPROGM は APF (許可プログラム機能) プログラムです。許可プログラムを実行する場合には、呼び出し側プログラムも許可が与えられていることが必要です。TSO を使用する場合は、非許可プログラムが TSO サービス・ルーチンである IKJEFTSR を使用して、IEBCOPY などの許可プログラムを起動できます。プログラム許可の詳細については、*z/OS MVS Programming: Authorized Assembler Services Guide* を参照してください。TSO の詳細については、*z/OS TSO/E Programming Services* を参照してください。

LINK マクロ命令の構文は次のとおりです。

[label]	LINK	EP=progrname ,PARAM=(optionaddr[,ddnameaddr [,hdingaddr]]) ,VL=1
---------	------	---------------------------------------------------------------------------

各項目の意味は以下のとおりです。

EP=progname

ユーティリティ・プログラムの名前を指定します。

PARAM=(optionaddr[,ddnameaddr [,hdingaddr]])

アプリケーション・プログラムからユーティリティ・プログラムに渡されるアドレス・パラメーターをサブリストとして指定します。すべてのパラメーターとパラメーター・リスト自体は、24 ビット・アドレス可能ストレージに入っている必要があります。これらのパラメーター・リストを作成するための詳細については、『パラメーター・リストの作成』を参照してください。コーディングできる値は以下のとおりです。

optionaddr

EXEC ステートメントの PARM パラメーターに通常指定するオプションが入ったオプション・リストのアドレスを指定します。このパラメーターは、すべてのユーティリティ・プログラムに対して指定する必要があります。

ddnameaddr

ユーティリティ・プログラムの処理中に使用されるデータ・セットの代替 DD 名のリストのアドレスを指定します。標準 DD 名を使用する場合、このパラメーターがリスト内の最後のパラメーターではない場合は、ハーフワードのゼロを指す必要があります。これが最後のパラメーターであれば、省略しても構いません。

hdingaddr

SYSPRINT データ・セットの EBCDIC 先頭ページ番号が入ったリストのアドレスを指定します。*hdingaddr* を省略する場合は、デフォルトのページ番号として 1 が使用されます。最後のパラメーターになる場合は、*hdingaddr* を省略しても構いません。

VL=1

アドレス・パラメーター・リストの最後のフルワードの符号ビットを 1 に設定することを指定します。

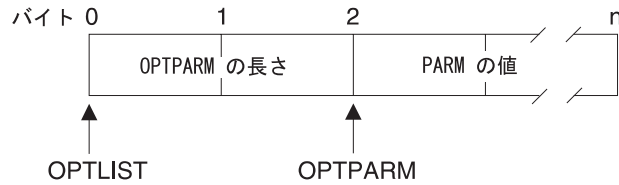
パラメーター・リストの作成

このセクションは、オプション・リスト、DD 名リスト、およびページ番号パラメーターについての情報を含んでいます。

オプション・リスト

オプション・リストは、EXEC ステートメントの PARM パラメーターに通常指定するオプションが入ったパラメーター・リストです。このリストは、ユーティリティ・プログラムに PARM オプションを渡さない場合も含め、どんな場合にも必要です。

PARM オプション・パラメーター・リスト (OPTLIST) の一般的な構文は、次のとおりです。



オプション・リストは、ハーフワード境界から始める必要があります。このリストの最初の 2 バイトには、オプション・リストの残りのバイト数の 2 進カウントを入れます。オプション・リストはフリー・フォームですが、それぞれのフィールドはコンマで区切ります。リスト内では、最初の 2 バイト (長さ標識) 以外の場所に、ブランクや 2 進のゼロを入れることはできません。

たとえば、IEBCOPY を開始して、いくつかの PARM パラメーターを渡す場合には、アセンブラーで次のようにコーディングできます。

```
LINK EP=IEBCOPY,PARAM=(OPTLIST),VL=1
.
.
.
OPTLIST DC AL2(L'OPTPARM)
OPTPARM DC C'SIZE=1000K,WORK=1M'
```

PARM パラメーターでは、小括弧や単一引用符を普通使いますが、上の例ではこれらをコーディングしていないことに注意してください。上の例の IEBCOPY の PARM パラメーターの場合は、PARAM=(SIZE=1000K,WORK=1M) または PARAM='SIZE=1000K,WORK=1M' とコーディングするのが普通ですが、アプリケーション・プログラムから IEBCOPY プログラムを開始するときには、小括弧や単一引用符で囲んだ値を渡すべきではありません。

ユーティリティー・プログラムに PARM パラメーター値を渡さない場合は、オプション・リストにハーフワードの 2 進ゼロをコーディングしてください。たとえば、PARM 値を使わずにデフォルトの DD 名を使用して、アプリケーション・プログラムから IEBGENER プログラムを開始するには、アセンブラーで次のようにコーディングします。

```
LINK EP=IEBGENER,PARAM=(OPTLIST),VL=1
.
.
.
OPTLIST DC H'0'
```

DD 名リスト

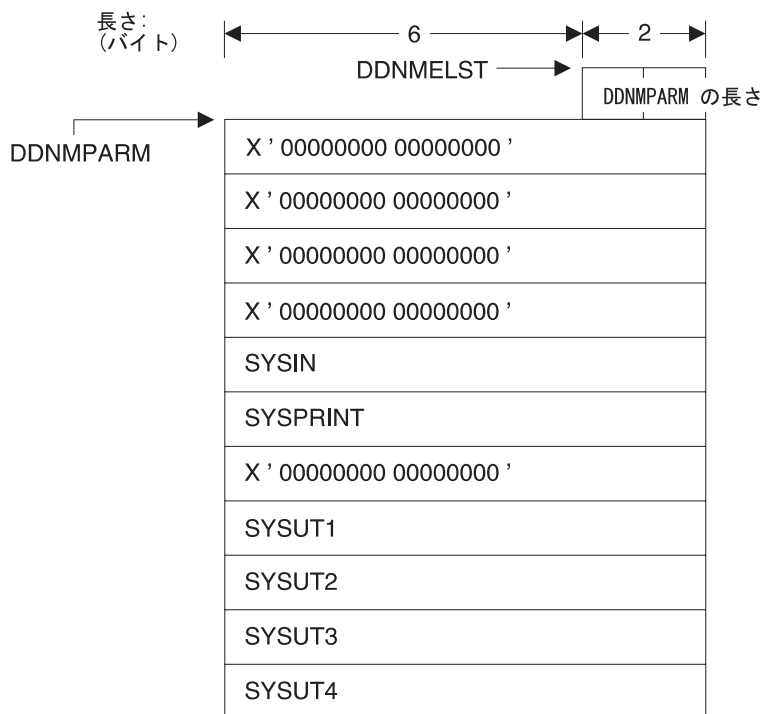
DD 名リストは、ユーティリティー・プログラムから使用するデータ・セットまたはボリュームの代替 DD 名が入ったパラメーター・リストです。データ・セットの標準 DD 名を使用する場合は、ページ・ヘッダー・パラメーターをコーディングする場合を除き、DD 名リストをコーディングする必要はありません。

代替 DD 名を使用する必要がある例として考えられるのは、ユーザーのプログラムが、異なるデータ・セットに対して同じ DD 名を使用する各種のプログラムを呼

び出す場合です。別の例としては、プログラムが動的にデータ・セットを割り振って、ユーティリティを呼び出す前にユーティリティ制御ステートメントを書き込む場合があげられます。

DD 名リストは、ハーフワード境界から始める必要があります。最初の 2 バイトには、リストの残りのバイト数のカウントを入れます。それぞれの DD 名は、8 バイトにしてください。DD 名が 8 バイトよりも短い場合は、左寄せにして、残りを空白で埋めるようにします。DD 名に 2 進ゼロをコーディングする場合や、DD 名リストを短くして DD 名を省略する場合は、標準 DD 名が使用されます。DD 名リストの途中から DD 名を省略する場合は、必ず 2 進ゼロを埋めてください。

DD 名パラメーター・リスト (DDNMELST) の一般的な構造は、次のとおりです。



たとえば、非標準 DD 名を使って IEBCOPY を開始する場合は、次のようにコーディングできます。

```

LINK EP=IEBCOPY,PARAM=(OPTLIST,DDNMELST),VL=1
.
.
.
OPTLIST DC H'0'
DDNMELST DC AL2(L'DDNMEND)
DDNMPARM DC 7XL8'0'
          DC CL8'INPDS  '
          DC CL8'OUTPDS  '
DDNMEND EQU DDNMPARM,*-DDNMPARM

```

この例では、IEBCOPY が入力データ・セットとして INPDS を使用し、出力データ・セットとして OUTPDS を使用することになっています。

IEBCOPY などのユーティリティーに複数の入力または出力データ・セットを設定するには、ユーティリティーに対して、SYSUT1 と SYSUT2 の代替 DD 名が入った DD 名リストを渡す必要があります。その他の必要な DD 名を識別するには、ユーティリティー制御ステートメントで十分です。

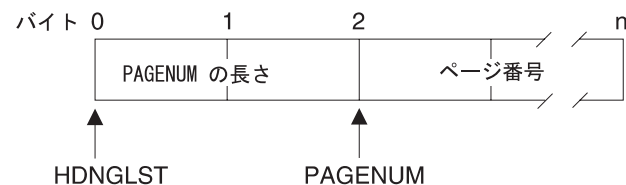
IEH システム・ユーティリティーを起動するときには、DD 名パラメーター・リストをコーディングする必要はありません。これらのユーティリティーの DD 名は、データ・セットではなく装置を定義するものであり、正しい装置を識別するには、これらのユーティリティーが使用するユーティリティー制御ステートメントで十分です。IEH ユーティリティーは、DD 名リストから SYSIN と SYSPRINT の項目だけを使用します。

ページ・ヘッダー・パラメーター

ユーティリティーにページ・ヘッダー・パラメーターを渡すことによって、印刷出力の先頭ページ番号を指定することができます。このパラメーターの最初の 2 バイトには、パラメーターの残りの長さを入れなければなりません。ページ番号は 4 バイトを超えてはならず、EBCDIC フォーマットにする必要があります。

一部のユーティリティーは、渡されたページ番号を更新し、最後に使用したページ番号に 1 を加えた値を使用します。これによって、連続的な起動が可能になります。

ページ・ヘッダー・パラメーター (HDNGLST) の一般的な構文は、次のとおりです。



たとえば、IEHLIST をロードして、最初のページがページ番号 10 から始まる印刷出力を得るには、次のようにコーディングします。

```
LINK EP=IEHLIST,PARAM=(OPTLIST,DDNMELST,HDNGLST),VL=1
.
.
.
OPTLIST DC H'0'
DDNMELST DC H'0'
HDNGLST DC AL2(L'PAGENUM)
PAGENUM DC C'10'
```

一部のユーティリティーは、1 つのページ番号に 4 バイト未満しか使用しません。ページ・ヘッダー・パラメーターに大き過ぎるページ番号を設定すると、予測不能な結果を招くことがあります。たとえば、ページ番号として 998 を設定して IEBIMAGE にリンクすると、ページ番号は以下ようになります。

998
999
(ブランク)
1
2
(以下同様)

この場合、999 よりも大きいページ番号を指定することはできません。

戻りコード

次のセクションでは、ユーティリティー・プログラムの戻りコードを定義します。

IEBCOMPR の戻りコード

IEBCOMPR は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。
08 (X'08')	比較は等しくない。処理は続行されます。
12 (X'0C')	回復不能エラーが存在します。ユーティリティーは終了します。
16 (X'10')	ユーザー・ルーチンが IEBCOMPR に戻りコード 16 を渡しました。ユーティリティーは終了します。

IEBCOPY の戻りコード

IEBCOPY は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。
04 (X'04')	1 つまたは複数のグループ・コピー操作が異常終了したか、最後まで完了しませんでした。リカバリーが可能な場合もあります。
08 (X'08')	回復不能エラーが存在します。ユーティリティーは終了します。

IEBCOPY のユーザー ABEND コード

診断処理中に、IEBCOPY がユーザー ABEND を出すことがあります。ただし、これは、*z/OS DFSMSdfp 診断解説書* に記されている手順に従った場合に限られます。ABEND の前には必ず IEBCOPY メッセージが表示されます。ABEND コードは、メッセージ番号と同じです。たとえば、ユーザー ABEND U1021 の前に表示されるメッセージは、IEB1021E になります。

IEBDG の戻りコード

IEBDG は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。

- 04 (X'04') ユーザー・ルーチンが IEBDGB にコード 16 を戻しました。ユーティリティは、ユーザーからの要求で終了します。
- 08 (X'08') ユーティリティ制御ステートメント・セットの処理中にエラーが発生しました。エラー後は、データが生成されません。その次のユーティリティ制御ステートメント・セットがあれば、それによって通常の処理が継続されます。
- 12 (X'0C') 入力または出力データ・セットの処理中にエラーが発生しました。ユーティリティは終了します。
- 16 (X'10') リカバリー不能なエラーが発生しました。ユーティリティは終了します。

IEBEDIT の戻りコード

IEBEDIT は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

- | コード | 意味 |
|------------|------------------------------------------------------------|
| 00 (X'00') | 正常終了。 |
| 04 (X'04') | エラーが発生しました。出力データ・セットをジョブ・ストリームとして使用できない可能性があります。処理は継続されます。 |
| 08 (X'08') | 入力、出力、制御データ・セットの処理を試みたときに回復不能エラーが発生しました。ユーティリティは終了します。 |

IEBGENER の戻りコード

IEBGENER は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

- | コード | 意味 |
|------------|----------------------------------------------------------|
| 00 (X'00') | 正常終了。 |
| 04 (X'04') | おそらく正常終了。警告メッセージが書き込まれます。 |
| 08 (X'08') | ユーザーがユーザー・ヘッダー・ラベルだけの処理を要求した後に処理が終了したか、DBCS エラーが検出されました。 |
| 12 (X'0C') | 回復不能エラーが存在するためにジョブ・ステップが停止したか、DBCS エラーが検出されました。 |
| 16 (X'10') | ユーザー・ルーチンが IEBGENER に戻りコード 16 を渡しました。ユーティリティは終了します。 |

IEBIMAGE の戻りコード

IEBIMAGE は、レジスター 15 に、プログラム実行中に検出された最も重大なエラーを示すコードを戻します。この戻りコードは、出力リストにも印刷されます。戻りコードとその意味は、次のとおりです。

- | コード | 意味 |
|------------|------------------------|
| 00 (X'00') | 正常終了。要求どおりに操作が実行されました。 |

- 04 (X'04') 操作は実行されましたが、例外状況を示すメッセージを調べてください。
- 08 (X'08') 操作は実行されませんでした。メッセージを調べてください。
- 12 (X'0C') 重大な例外。処理が終了する場合があります。
- 16 (X'10') 回復不能例外。ユーティリティは終了します。
- 20 (X'14') SYSPRINT データ・セットをオープンできませんでした。ユーティリティは終了します。
- 24 (X'18') ユーザー・パラメーター・リストが正しくありません。ユーティリティは終了します。

IEBPTPCH の戻りコード

IEBPTPCH は、プログラム実行の結果を示すレジスター 15 のコードを戻します。戻りコードとその意味は、次のとおりです。

- | コード | 意味 |
|------------|-----------------------------------------------------------------------------------|
| 00 (X'00') | 正常終了。 |
| 04 (X'04') | 物理順次データ・セットが空であるか、区分データ・セットにメンバーがありません。 |
| 08 (X'08') | 印刷または穿孔のために指定されたメンバーが入力データ・セット内に存在しないため、その次のメンバーで処理が続行されるか、あるいは DBCS エラーが検出されました。 |
| 12 (X'0C') | 回復不能エラーが発生したか、ユーザー・ルーチンは IEBPTPCH に戻りコード 12 を渡してユーティリティが終了するか、あるいは DBCS が検出されました。 |
| 16 (X'10') | ユーザー・ルーチンが IEBPTPCH に戻りコード 16 を渡しました。ユーティリティは終了します。 |

IEBUPDTE の戻りコード

IEBUPDTE は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

- | コード | 意味 |
|------------|-------------------------------------------------------------------------------------------------------------------|
| 00 (X'00') | 正常終了。 |
| 04 (X'04') | 制御ステートメントのコーディングが間違っているか、使い方に誤りがあります。入力、出力のいずれかが順次である場合は、ユーティリティが終了します。両方とも区分である場合は、その次の機能の実行によってプログラムの処理が続行されます。 |
| 12 (X'0C') | 回復不能エラーが存在します。ユーティリティは終了します。 |
| 16 (X'10') | ユーザーのラベル処理ルーチンからラベル処理コード 16 を受け取りました。ユーティリティは終了します。 |

IEHINITT の戻りコード

IEHINITT は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。メッセージ・データ・セットが作成されました。
04 (X'04')	正常終了。ユーザーによってメッセージ・データ・セットが定義されていません。
08 (X'08')	IEHINITT の操作が完了しましたが、処理中にエラー状態が検出されました。メッセージ・データ・セットが作成されました。
12 (X'0C')	IEHINITT の操作が完了しましたが、処理中にエラー状態が検出されました。ユーザーによってメッセージ・データ・セットが定義されていません。
16 (X'10')	制御データ・セットの読み取りを試みたときに検出されたエラー状態のために、IEHINITT の操作が終了しました。ユーザーによってメッセージ・データ・セットが定義されている場合は、メッセージ・データ・セットが作成されました。

IEHLIST の戻りコード

IEHLIST は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。
08 (X'08')	エラー状態のために指定の要求が無視されました。処理は続行されます。
12 (X'0C')	永続的な入出力エラーが発生しました。ジョブは終了します。
16 (X'10')	データ・セットの読み取り中に回復不能エラーが発生しました。ジョブは終了します。

IEHMOVE の戻りコード

IEHMOVE は、レジスター 15 に、プログラム実行の結果を示すコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。
04 (X'04')	指定の機能が正常に完了しませんでした。処理を続行します。
08 (X'08')	リカバリー可能な状態が存在します。処理を続行します。
12 (X'0C')	回復不能エラーが存在します。ユーティリティは終了します。
16 (X'10')	SYSIN または SYSPRINT データ・セットをオープンできません。ユーティリティは終了します。

IEHPROGM の戻りコード

IEHPROGM は、プログラム実行の結果を示すレジスター 15 のコードを戻します。戻りコードとその意味は、次のとおりです。

コード	意味
00 (X'00')	正常終了。
04 (X'04')	制御ステートメントの名前フィールド、または EXEC ステートメントの PARM フィールドで構文エラーが検出されました。処理は続行されます。
08 (X'08')	制御ステートメントが正しくないか、あるいはその他の理由で要求が無効であるために、特定の操作の要求が無視されました。この操作は実行されません。
12 (X'0C')	SYSPRINT、SYSIN、VTOC の読み取りまたは書き込みを試みたときに、入出力エラーが検出されました。ユーティリティは終了します。
16 (X'10')	回復不能エラーが存在します。ユーティリティは終了します。

付録 B. アンロード区分データ・セット・フォーマット

この付録は、プログラミング・インターフェースとそれに関連する情報について記述しています。

この付録の目的は、IEBCOPY アンロード操作で作成されたアンロード・データ・セットを使用する手助けをすることです。

はじめに

データ・セットを DASD 装置に保管するか、磁気テープに保管するにかかわらず、アンロード・データ・セットのフォーマットは、区分データ・セットや PDSE のフォーマットとは異なります。アンロード・データ・セットのレコードのほうが長いので、余計にスペースを必要とする場合があります。また、アンロード・データ・セットは順次編成になります。アンロード・データ・セットは、元の区分データ・セットや PDSE と同じように扱えない場合があります。

順次編成であるアンロード・データ・セットは、区分データ・セットや PDSE とよく混同されます。アンロード・データ・セット、またはコンテナ・データ・セットは、アンロード操作によって作成される順次データ・セットです。これに対して、アンロードされるデータ・セットとは、アンロード・データ・セットにアンロードされた元の区分データ・セットまたは PDSE のことをいいます。

アンロード・データ・セット内のレコード

ここで、アンロード・データ・セット内に存在するすべてのタイプのレコードをリストアップします。区分けや順番は、実際の並び方と同じです。

1. アンロード・ファイル・ヘッダー
 - COPYR1、1 次ヘッダー・レコードは必ず存在します。
 - COPYR2、第 2 ヘッダー・レコードは必ず存在します。
2. ディレクトリー情報
 - ディレクトリー・ブロック・レコード
 - 1 つまたは複数のレコードが必ず存在します。
 - 各レコードには、1 つまたは複数の区分データ・セット・ディレクトリー・ブロックが入ります。
 - さらに最後のレコードの場合は、12 バイトのゼロを並べた EOF ブロックで終了します。
 - PDSE の場合は、8 バイト以上の名前が入った長い名前のディレクトリーも含まれます。
 - データ・セット属性レコード
 - これはオプションです。PDSE フォーマットのアンロード・データ・セットにのみ存在します。
3. 個々のメンバーのデータ (このグループは各メンバーごとに繰り返されます。)
 - 注釈リスト・レコード

リンケージ・エディターや他のアプリケーションはこのレコードを使って、メンバー内のレコードの再配置可能アドレスを記録します。

- **メンバー・データ・レコード**
1 つまたは複数のレコードが存在します。
各レコードには、元のデータ・セットの 1 つまたは複数の物理ブロックが入ります。
さらに最後のレコードの場合は、EOF ブロック (12 バイトのゼロ) で終了します。
- **メンバー属性レコード**
これはオプションです。PDSE フォーマットのアンロード・データ・セットにのみ存在します。

各レコードの詳細な説明については、この後の段落を参照してください。

アンロード・データ・セットの各種フォーマット

アンロード・データ・セットには、3 種類のフォーマットがあります。それぞれの主な違いは、データ・セット内にどんなレコードが含まれるかということです。各フォーマットは以下のとおりです。

1. 無効なフォーマット

COPYR1 以降のレコードがあれば、それらはすべて不定形式になります。この状態は、エラーのためにアンロード操作が終了した場合に発生します。

COPYR1 は、エラー整理に『無効なフォーマット』を含めた形で、コンテナー・データ・セット内の最初のレコードとして書き直されます。

2. 古いフォーマット (PDSE 以前)

- 各メンバーの注釈リスト・レコードが存在する場合があります。
- 属性レコードは存在しません。
- 元のデータ・セットは、区分データ・セットであり、PDSE ではありません。
- 実際の装置のすべての DASD アドレスは有効なので、DEB と DEVTAB の情報は、元のデータ・セットが入っていた DASD 装置から得られます。
- データ・セット・ラベル情報の 2 番目のバッチ (COPYR1 レコードに入る最初の 46 バイト) は存在しません。

3. 新しいフォーマット

- リンケージ・エディターや他のアプリケーションは、注釈リスト・レコードを使って、メンバー内のレコードの再配置可能アドレスを記録するようになりました。
- 元のデータ・セットが PDSE の場合は、属性レコードが存在する場合があります。
- PDSE からのレコードには、256 個のトラックと 65536 個のシリンダーと 16M バイトのトラックを装備した見せかけの装置からの DASD アドレスが入ります。この便利な機能によって、BPAM で PDSE にアクセスするための制約 (*z/OS DFSMS データ・セットの使用法* を参照) に基づき、PDSE の RLT と MLT の最大数がマッピングされます。

これらの連続レコードのアドレスは厳密な昇順になりますが、一部のアドレスは使用されません。PDSE メンバーのレコード番号は、1 から始まって 3、5、7、9 ... と続く奇数になります。偶数のレコード番号は予約されています。

- データ・セット・ラベル情報の 2 番目のバッチ (COPYR1 レコードに入る最初の 46 バイト) は有効です。

詳細レコードの説明

一般的な規則と制約

1. アンロード・レコードの最大長は 32780 であり、アンロード中のデータ・セットのブロック・サイズが 32760 のときに、この最大値に達します。

この最大値は、物理ブロック長の最大許容値である 32760 バイトよりも大きい
ため、レコードが複数の物理ブロックにまたがることもあります。

また、この値は、データ・セット・ラベル内の LRECL の最大許容値も超えています。データ・セット・ラベル LRECL が 32760 の場合でも、アセンブル後の論理レコードは、実際にはもっと長くなる場合があります。

2. 1 つのレコードが複数の物理ブロックにまたがる場合、最後のブロック以外はすべて完全に埋め尽くされる必要があります。
3. アンロード・データ・セットの物理レコードの長さは、アンロード・データ・セットの論理レコードの長さ + 4 を加えた値を超えることができません。データ・セットのブロック・サイズからして、それ以上の長さの物理レコードが許容される場合でも、この制限が適用されます。

ヘッダー・レコード

IEBCOPY アンロード・データ・セットの最初の 2 つのレコードには、データ・セットをロードするために必要な情報が入ります。最初のレコード (COPYR1) には、状況フィールド、ID フィールド、DCB 情報フィールドがあります。状況フィールドと ID フィールドは、妥当性検査プロシージャのために使用されます。DCB フィールドは、ロード機能の実行中に出力データ・セットを初期化するために使用されます。2 番目のレコード (COPYR2) には、アンロードされた区分データ・セットの元のデータ・エクステンツ・ブロック (DEB) の一部が入ります。データ・セットをロードするときに、この情報に基づいて、ユーザー・リストと注釈リストのすべての TTR が更新されます。表 58 と 371 ページの表 59 は、COPYR1 レコードと COPYR2 レコードの各フィールドをまとめたものです。

表 58. COPYR1 記述子レコードの内容

レコードへの オフセット	フィールド・ サイズ (バイト)	データのタイプ	フィールドの内容
0	64	ストラクチャー	COPYR1 - 1 次のヘッダー・レコード
0	4	ストラクチャー	RECFM=VS データ・セットのブロック記述子ワード (BDW)
0	2	無符号 2 進	BDW を含めたブロックの長さ
2	2	予約済み	必ずゼロ
4	4	ストラクチャー	RECFM=VS データ・セットのセグメント記述子ワード (SDW)
4	2	無符号 2 進	SDW を含めたセグメントの長さ
6	2	ビット・フラグ	必ずゼロ (COPYR1 レコードは絶対にセグメント化されない)

表 58. COPYRI 記述子レコードの内容 (続き)

レコードへの オフセット	フィールド・ サイズ (バイト)	データのタイプ	フィールドの内容
8	1	ビット・フラグ	<p>アンロード・データ・セット情報。 MSB を "0" として、以下のようになります。</p> <p>0 & 1 - B'00' = 古いフォーマットの有効なアンロード・データ・セット。</p> <p>B'01' = PDSE フォーマットの有効なアンロード・データ・セット。</p> <p>B'10' = このアンロード・データ・セットが不完全であるか誤っていることが検出されているために、元のデータ・セットは再ロードできません。</p> <p>B'11' = 予約済みの形式。</p> <p>2 - 予約済みであり、必ずゼロ。</p> <p>4 - 設定された場合は、元のデータ・セットにプログラム・オブジェクトが入っていることが検出されました。設定されなかった場合は、内容がプログラムであるかどうかを検出されていません。</p> <p>4 - 予約済みであり、必ずゼロ。</p> <p>5 - 予約済みであり、必ずゼロ。</p> <p>6 - 予約済みであり、必ずゼロ。</p> <p>7 - 設定された場合は、元のデータ・セットが PDSE でした。</p>
9	3	2 進	<p>定数値 X'CA6D0F'。</p> <p>(以下のフィールドは元のデータ・セット・ラベルから得られる (フォーマット 1 DSCB)。)</p>
12	2	ビット・ストリング	データ・セット編成 (DS1DSORG)。 X'0200' は PDS。
14	2	無符号 2 進	ブロック・サイズ (DS1BLKL)
16	2	無符号 2 進	論理レコード長 (DS1LRECL)
18	1	ビット・フラグ	<p>レコード・フォーマット (DS1RECFM)。 MSB を "0" として、以下のようになります。</p> <p>0 & 1 - B'00' は不明のフォーマット</p> <p>B'01' は可変長フォーマット (RECFM=V)</p> <p>B'10' は固定長フォーマット (RECFM=F)</p> <p>B'11' は不定形式 (RECFM=U)</p> <p>2 - 設定された場合は、DASD トラック・オーバーフローを使用できます。</p> <p>3 - 設定された場合は、レコードをブロックできます。</p> <p>4 - 設定された場合は、可変長フォーマット・レコードをスパンできます。固定長フォーマットの場合の限り、最後のブロックは短くても構いません。</p> <p>5 & 6 - B'00' 最初のレコード・バイトがデータ・タイプ。</p> <p>B'01' 最初のバイトが ANSI/ISO 紙送り制御。</p> <p>B'10' 最初のバイトが IBM マシンの紙送り制御。</p> <p>B'11' 無効な組み合わせ。</p> <p>7 - 予約済みであり、ゼロか 1 のいずれか。</p>
19	1	無符号 2 進	レコード・キー・フィールドの長さ (DS1KEYL)
20	1	ビット・フラグ	データ・セットに関連したオプション・コード (DS1OPTCD)
21	1	ビット・フラグ	<p>SMS 標識 (DS1SMSFG)。 MSB を "0" として、以下のようになります。</p> <p>0 - 管理対象のデータ・セット</p> <p>1 - 予測不能</p> <p>2 - データ・セットは再ブロック可能</p> <p>3 - 予測不能</p>

表 58. COPYR1 記述子レコードの内容 (続き)

レコードへの オフセット	フィールド・ サイズ (バイト)	データのタイプ	フィールドの内容
			4 - データ・セットは PDSE 5 - 予測不能 6 - 予約済み 7 - 予約済みであり、必ずゼロ。(元のデータ・セット・ラベルからのフィールドの終わり)。
22	2	無符号 2 進	アンロード・データ・セットを含めたこのコンテナ・データ・セットのブロック・サイズ。
24	20	ストラクチャー	データ・セットのアンロード元の装置に関する情報。DEVTAB パラメーターを設定した DEVTYPE マクロによって得られます。 ¹
44	2	無符号 2 進	ヘッダー・レコードの数。ゼロは、2 に暗黙指定されます。
46	18	予約済み	"古い" フォーマットのアンロード・データ・セット内のゼロ (以下のフィールドは、元のデータ・セット・ラベルから得られる (フォーマット 1 DSCB)。)
46	1	予約済み	必ずゼロ
47	3	ストラクチャー	最後に yydddd として参照された日付 (DS1REFD)
50	3	ストラクチャー	2 次スペース拡張 (DS1SCEXT)
53	4	ストラクチャー	2 次割り振り (DS1SCALO)
57	3	ストラクチャー	TTR を使用した最後のトラック (DS1LSTAR)
60	2	無符号 2 進	最後のトラック・バランス (DS1TRBAL)
62	2	予約済み	必ずゼロ (元のデータ・セット・ラベルからのフィールドの終わり)。

注:

- これらのフィールドは、装置への依存度が高く、メンバー・データ・レコード内の絶対 DASD アドレス (MBBCCCHHR) を相対アドレス (TTR) に変換する必要があります。DEB 制御ブロックと DEVTYPE マクロについては、*z/OS DFSMSdfp 拡張サービス* に説明があります。

表 59. COPYR2 記述子レコードの内容

レコードへの オフセット	フィールド・ サイズ (バイト)	データのタイプ	フィールドの内容
0	284	ストラクチャー	COPYR2 - 1 次のヘッダー・レコード
0	4	ストラクチャー	RECFM=VS データ・セットのブロック記述子ワード (BDW)
0	2	無符号 2 進	BDW を含めたブロックの長さ
2	2	予約済み	必ずゼロ
4	4	ストラクチャー	RECFM=VS データ・セットのセグメント記述子ワード (SDW)
4	2	無符号 2 進	SDW を含めたセグメントの長さ
6	2	ビット・フラグ	必ずゼロ (COPYR2 レコードは絶対にセグメント化されない。)
8	16	ストラクチャー	元のデータ・セット用のデータ・エクステンツ・ブロック (DEB) の基本セクションの最後の 16 バイト。 ¹
24	256	ストラクチャー	元の DEB から得られた最初の 16 個のエクステンツ説明。 ¹
280	4	予約済み	必ずゼロ

注:

- これらのフィールドは、装置への依存度が高く、メンバー・データ・レコード内の絶対 DASD アドレス (MBBCCCHHR) を相対アドレス (TTR) に変換する必要があります。DEB 制御ブロックについては、*z/OS DFSMSdfp 拡張サービス* に説明があります。また、DEVTAB の情報については、DFP System Data Administration を参照してください。

ディレクトリー・ブロック・レコード

ディレクトリー・レコードは、ヘッダー・レコードのすぐ後に書き込まれます。このレコードには、すべてのメンバーをアンロードするための元のディレクトリー項目が入っているディレクトリー・ブロックの内容が含まれます。さらに、最後のディレクトリー・レコードには、ファイル終わりブロックも入ります。

各ディレクトリー・レコード (最後のレコード以外) の長さは、 $8 + n(276)$ になります。なお、 n はブロック化因数を表し、ゼロより大きい整数になります。最後のディレクトリー・レコードの長さは、 $8 + n(276)$ に 12 を加えた値になります。 n はブロック化因数ですが、ゼロの場合もあります。

図中のディレクトリー・ブロックには、カウント、キー、データの各フィールドが入っています。カウント・フィールドは、キー長 (X'08') とデータ長 (X'0100') 以外は、ゼロに設定されます。ディレクトリー・ブロックの詳細については、*z/OS DFSMS データ・セットの使用法* を参照してください。

図 48 は、アンロードされる区分データ・セットのディレクトリー・レコード・フォーマットをまとめたものです。ただし、以下のような前提があります。

- アンロード・データ・セットを入れるためのデータ・セットのブロック・サイズは 900 バイトです。
- すべてのアンロード・メンバー用の元のディレクトリー項目を入れるために、7 つの疑似ディレクトリー・ブロックが必要です。

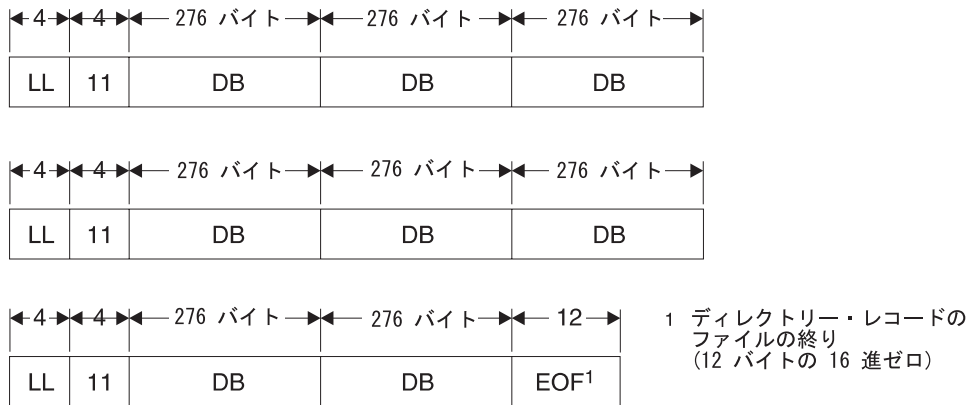


図 48. ディレクトリー・レコードのレイアウト

属性レコード

PDSE には属性があり、その属性は属性レコードに記録されます。データ・セット全体に関連した属性は、ディレクトリー・レコードの後に続き、各メンバーに関連した属性は、各メンバーのデータ・レコードの後に続きます。アンロード属性レコードのフォーマットは、373 ページの図 49 のとおりです。

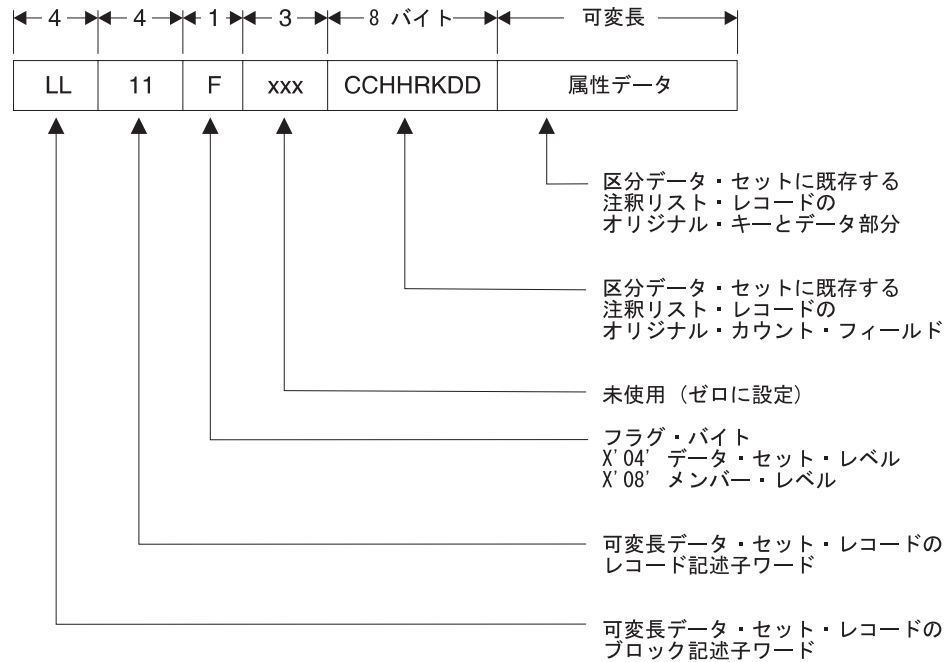


図 49. 属性レコードのレイアウト

注釈リスト・レコード

ここで説明する注釈リスト・レコードはすでに廃止されており、普通は、古いフォーマットのアンロード・データ・セットにも存在しません。注釈レコードは、新しいフォーマットのアンロード・データ・セット内ではメンバー・データとして処理されます。

アンロードするメンバーに注釈リストが含まれている場合、注釈リストはメンバー・データに先立ってアンロードされます。アンロード注釈リスト・レコードのフォーマットは、374 ページの図 50 のとおりです。

IEBCOPY ユーザー TTRN 限界:

- ディレクトリー内のユーザー TTRN フィールドは 3 つです。
- そのうちのいずれか 1 つだけが $n > 0$ の値を持てます。
- $n > 0$ になっているユーザー TTRN で識別される注釈リスト・レコードの最大長は、ブロックおよびレコード記述子ワードを含めて 1291 バイトです。
- 注釈リスト・レコード内の TTRN フィールドは、 $n > 0$ の値を持てません。
- 注釈リスト・レコード内または区分データ・セット・ディレクトリー内のユーザー TTRN フィールドは、左端のビット (つまり、TTRN 内の最初の『T』の最上位ビット) をオンにできません。

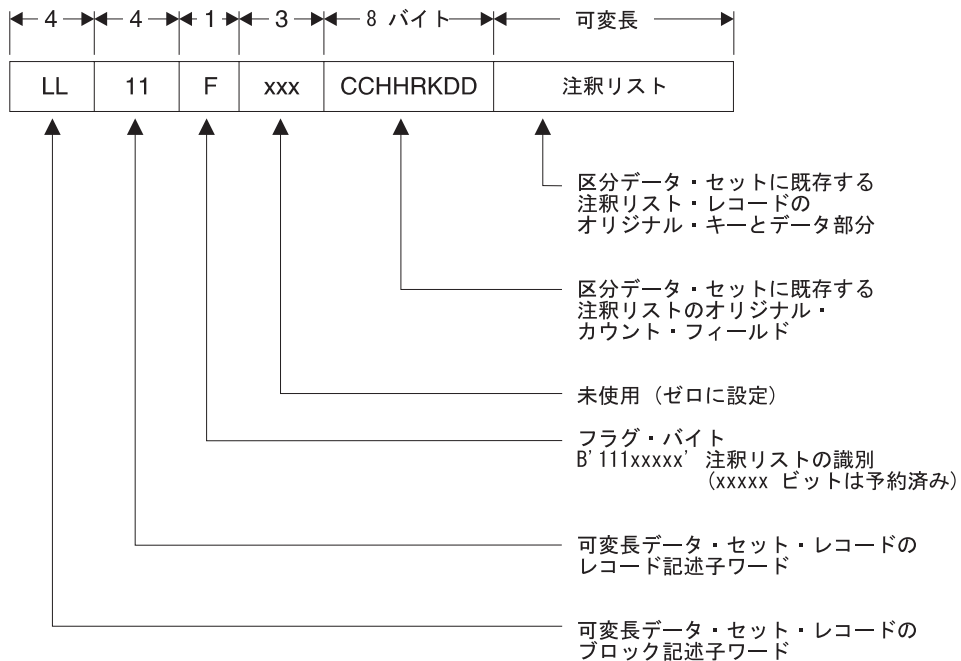


図 50. 注釈リスト・レコードのレイアウト

メンバー・データ・レコード

アンロードされるメンバー・データ・レコードには、アンロード・データ・セット・レコードに組み込まれるセット・メンバー・データ・ブロックの最大数が含まれます。各メンバー・データ・レコード内のブロックの数は、区分データ・セットまたは PDSE に、不定長ブロックまたは可変長ブロックが存在すると変わってきます。メンバー・データ・レコードには、1 つのメンバーだけのデータ・ブロックが入り、最後のメンバー・データ・ブロックの後にファイル終わりブロックが来ます。

375 ページの図 51 は、アンロードされるメンバー・データ・レコードのフォーマットの例です。ただし、以下のような前提があります。

- 区分データ・セットのブロック・サイズは 350 バイトです。
- 1 つのメンバーのメンバー・データ・ブロックは 6 個です (直接アクセス・ファイル終わりブロックも含む)。
- 区分データ・セットのレコード構文は固定されています。
- アンロード・データ・セットを入れるためのデータ・セットのブロック・サイズは 900 バイトです。



図 51. メンバー・データ・レコードのレイアウト

図 52 は、各メンバー・データ・ブロックの構成を示したものです。

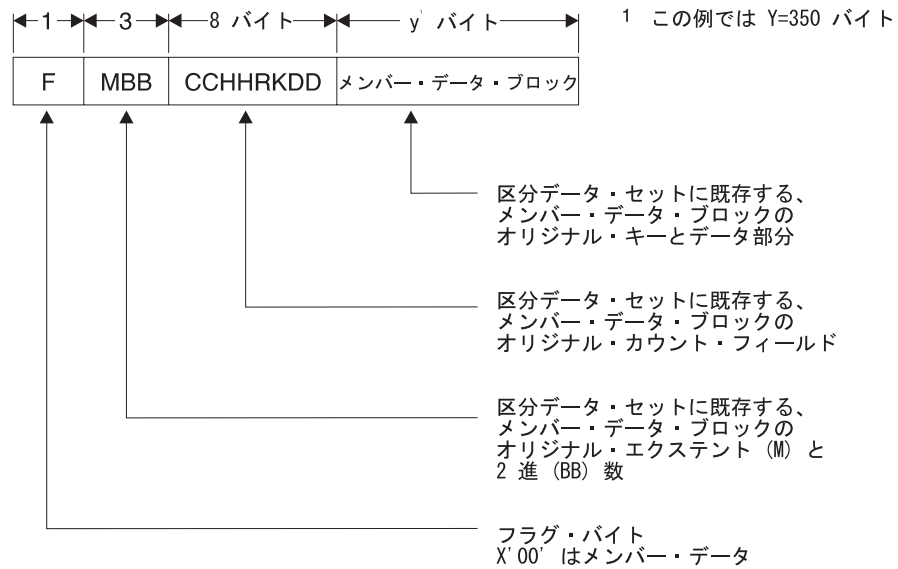


図 52. メンバー・データ・ブロックのレイアウト

376 ページの図 53 は、メンバーの最後のデータ・ブロックの後に来るファイル終わりブロックの構成を示したものです。

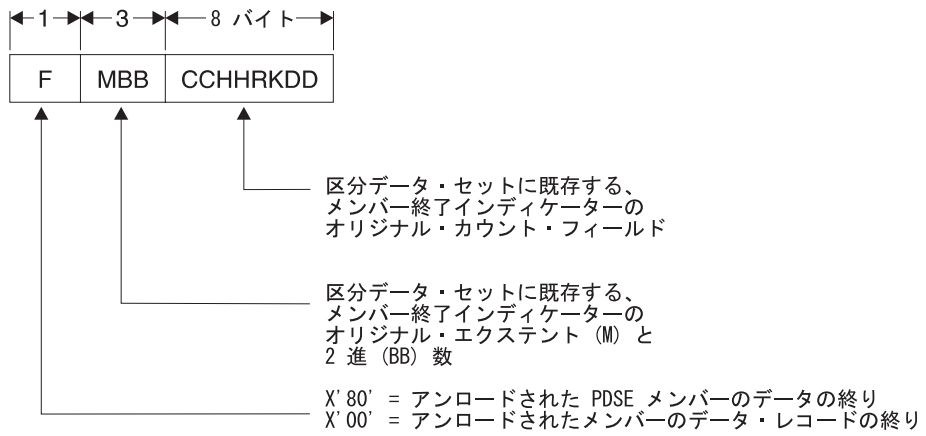


図 53. ファイル終わりブロックのレイアウト

付録 C. ユーティリティ・プログラムのユーザー出口を指定する

この付録は、プログラミング・インターフェースとそれに関連する情報を記述しています。

この付録の目的は、ユーティリティ・プログラムのユーザー出口を作成する手助けをすることです。

全体的な指針

各種ユーティリティに出口を指定すると、次のようなことができます。

- 物理レコードの変更
- 入出力エラーの処理
- ユーザー入出力のヘッダー・ラベルとトレーラー・ラベルの処理

出口は、各種ユーティリティ (IEBDG と IEBUPDTE を除く) の EXITS ステートメントのパラメーターで指定します。ユーティリティ・プログラムで使用できる出口を 表 60 にリストアップしました。

表 60. ユーティリティに指定できるユーザー出口ルーチン

出口ルーチン	使用できるとき	ユーティリティ	指定する場所
IEBGENER による処理前の物理レコードの変更	物理レコードを読み取ってから編集を実行するまでの間	IEBGENER	EXITS ステートメントの DATA パラメーター
入力のヘッダー・ラベルまたはトレーラー・ラベル	データ・セットが、入力用にオープンされたとき (ヘッダー) またはクローズされたとき (トレーラー)	IEBCOMPR、IEBGENER、IEBTPCH	EXITS ステートメントの INHDR/INTLR パラメーター
		IEBUPDTE	EXEC JCL ステートメントの PARM パラメーター、または ADD、CHANGE、REPL、REPRO ステートメントの INHDR、INTLR、OUTHDR、OUTTLR パラメーター。
出力のヘッダー・ラベルまたはトレーラー・ラベル	データ・セットが、出力用にオープンされたとき (ヘッダー) またはクローズされたとき (トレーラー)	IEBCOMPR、IEBGENER	EXITS ステートメントの OUTHDR/OUTLR パラメーター
合計	IEBGENER による各物理レコードの書き込みの前 (順次データ・セットのみ)	IEBGENER	EXITS ステートメントの TOTAL パラメーター
		IEBUPDTE	ADD、CHANGE、REPL、REPRO ステートメントの TOTAL パラメーター
入出力エラー	IEBGENER で永続エラーが発生したとき	IEBGENER	EXITS ステートメントの IOERROR パラメーター
IEBCOMPR によって検出されたエラー	同等でない比較の後	IEBCOMPR	EXITS ステートメントの ERROR パラメーター

表 60. ユーティリティに指定できるユーザー出口ルーチン (続き)

出口ルーチン	使用できるとき	ユーティリ ティー	指定する場所
出力レコード・キーの作成	IEBGENER によるレコードの書き込みの前	IEBGENER	EXITS ステートメントの KEY パラメーター
比較前の入力データ・セットの論理レコードの処理	入力レコードが IEBCOMPR によって処理される前	IEBCOMPR	EXITS ステートメントの PRECOMP パラメーター
IEBPTPCH 入出力レコードの処理	論理レコードの処理前 (INREC) または論理レコードの書き込み前 (OUTREC)	IEBPTPCH	EXITS ステートメントの INREC/OUTREC パラメーター
IEBDG 出力レコードの分析または変更	出力レコードが構成されてから出力データ・セット内に置かれるまでの間	IEBDG	CREATE ステートメントの EXIT パラメーター

ユーティリティ・プログラムから出口ルーチンへ入るときのレジスタの内容

レジスター	内容
1	パラメーター・リストのアドレス。
13	レジスター保管域のアドレス。この保管域は、ユーザー・ラベル処理ルーチンからは絶対に使用できません。
14	ユーティリティへの戻りアドレス。
15	出口ルーチンへの入り口アドレス。

プログラミングに関する注意点

出口ルーチンは、ジョブ・ライブラリー、ステップ・ライブラリー、リンク・ライブラリーのいずれかに入れる必要があります。

出口ルーチンからの戻り

出口ルーチンは、内部の RETURN マクロ命令によって、ユーティリティ・プログラムに制御を戻します。レジスター 1 から 14 までは、ユーティリティ・プログラムに制御が戻される前に復元されるはずですが、

RETURN マクロ命令のフォーマットは次のとおりです。

[label]	RETURN	[(r,r)] [,RC={n(15)}]
---------	--------	--------------------------

各項目の意味は以下のとおりです。

(r,r)

レジスター保管域のユーティリティ・プログラムによって再ロードされるレジスタの範囲 (0~15) を指定します。たとえば、(14,12) とすると、レジスター

13 以外のすべてのレジスターが復元されることとなります。このパラメーターを省略した場合は、レジスターが出口ルーチンによって正しく復元されたものと見なされます。

RC={nl(15)}

レジスター 15 の 10 進戻りコードを指定します。RC を省略した場合は、レジスター 15 が (r,r) で指定したとおりにロードされます。

n レジスター 15 に入れられる戻りコードを指定します。

(15)

汎用レジスター 15 にすでに有効な戻りコードが入っていることを指定します。

ユーザー・ラベル処理ルーチンは、表 61 のように、レジスター 15 にコードを戻すはずですが、以下の場合は例外です。

- ラベル処理ルーチンへの入り口の前で、バッファー・アドレスがゼロに設定された場合。この場合、システムは、戻りコードにかかわらず通常の処理を再開します。
- 回復不能な出力エラーの発生後に、ユーザー・ラベル処理ルーチンが実行された場合。この場合、システムは通常の処理を再開しようと試みます。

表 61 は、ユーザー出口ルーチンからユーティリティー・プログラムに対して出された戻りコードをまとめたものです。

表 61. ユーザー出口ルーチンが出す戻りコード

出口のタイプ	リターン・コード	処置
入力のヘッダー・ラベルまたはトレーラー・ラベル (UPDATE=INPLACE の場合の IEBUPDTE を除く)	0	システムは、通常の処理を再開します。ラベル・グループ内にさらにラベルがあっても、それらは無視されます。
	4	その次のユーザー・ラベルがラベル・バッファー域に読み込まれ、ユーザーのルーチンに制御が戻されます。もうラベルがない場合は、通常の処理が再開されます。
	16	ユーティリティー・プログラムが、ユーザー・ルーチンの要求によって終了します。

表 61. ユーザー出口ルーチンが出す戻りコード (続き)

出口のタイプ	リターン・コード	処置
入力のヘッダー・ラベルまたはトレーラー・ラベル (UPDATE=INPLACE の場合の IEBUPDTE)	0	システムは通常の処理を再開し、それ以上のユーザー・ラベルは無視されます。
	4	システムは、ラベルを書き込みません。その次のユーザー・ラベルがラベル・バッファ領域に読み込まれ、ユーザー出口ルーチンに制御が戻されます。もうユーザー・ラベルがない場合は、システムが通常の処理を再開します。
	8	システムは、ラベル・バッファ領域からのユーザー・ラベルを書き込み、通常の処理を再開します。
	12	システムは、ラベル・バッファ領域からのユーザー・ラベルを書き込み、その次の入力ラベルをラベル・バッファ領域に読み込んでから、ラベル処理ルーチンに制御を戻します。もうユーザー・ラベルがない場合は、システムが通常の処理を再開します。
出力のヘッダー・ラベルまたはトレーラー・ラベル	0	システムは、通常の処理を再開します。ラベル・バッファ領域からのラベルは書き込まれません。
	4	ユーザー・ラベルが、ラベル・バッファ領域から書き込まれます。それから、システムが通常の処理を再開します。
	8	ユーザー・ラベルが、ラベル・バッファ領域から書き込まれます。作成されたラベルが 8 個未満の場合は、ユーザーのルーチンが再び制御を得て、別のユーザー・ラベルを作成するようにします。8 個のラベルが作成された場合は、システムが通常の処理を再開します。
	16	ユーティリティ・プログラムが、ユーザー・ルーチンの要求によって終了します。
合計出口	0	処理は続行されますが、これ以降、ユーザー出口は取られません。
	4	通常のコマンドが続行されます。
	8	出力データ・セットの EOD 処理 (ユーザー・ラベル処理) 以外の処理は停止します。
	16	ユーティリティ・プログラムが停止します。

表 61. ユーザー出口ルーチンが出す戻りコード (続き)

出口のタイプ	リターン・コード	処置
ERROR	0	レコードは、エラー・データ・セットに入れられません。その次のレコードで処理が続行されます。
	4	レコードが、エラー・データ・セットに入れられます (SYSUT3)。
	8	レコードは、エラー・データ・セットに入れられませんが、有効なレコードとして処理されます (OUTREC と SYSUT2 を指定した場合、そこに送られる)。
	16	ユーティリティ・プログラムが終了します。
OUTREC (IEBPTPCH)	4	レコードは、通常の実出力データ・セットに入れられません。
	12 または 16	ユーティリティ・プログラムが終了します。
	その他の数値	レコードが、通常の実出力データ・セットに入れられます (SYSUT2)。
その他の出口	0-11 (その次の 4 の倍数に設定される)	戻りコードが、それまでの最高値の戻りコードと比較され、高いほうが保管され、低いほうが破棄されます。通常ジョブ終了では、最高値の戻りコードが呼び出しプロセッサに渡されます。
	12 または 16	ユーティリティ・プログラムが停止し、この戻りコードが呼び出しプロセッサに渡されます。

ラベル処理ルーチンに渡されるパラメーター

ユーザー・ラベル処理ルーチンに渡されるパラメーターは、80 バイトのラベル・バッファのアドレス、処理される DCB のアドレス、回復不能の入出力エラーが発生した場合の状況情報のアドレス、合計域のアドレスです。

80 バイトのラベル・バッファには、入力ラベルが処理されているときのユーザー・ラベルのイメージが入ります。出力ラベルの処理中は、ラベル処理ルーチンへの入り口で、バッファに重要な情報が入るわけではありません。ユーティリティ・プログラムに対してラベル生成要求が出された場合は、ラベル処理ルーチンがラベル・バッファ内にラベルを構成するはずですが、

標準ユーザー・ラベル (SUL) をデータ・セットの DD ステートメントに指定した場合は、データ・セットにユーザー・ラベルがなくても、システムは、該当するユーザー・ルーチンに対する指定の出口を取ります。そのような場合、ユーザー入力ラベル処理ルーチンは、バッファ・アドレス・パラメーターをゼロに設定して実行されます。

DCB の形式と内容については、*z/OS DFSMS Macro Instructions for Data Sets* で説明されています。

DCB アドレス・パラメーターのフラグ 1 のビット 0 は、以下の場合以外、0 の値に設定されます。

- ボリューム切り替え時に、ボリュームのトレーラー・ラベルまたはヘッダー・ラベルが処理されている場合。
- DISP=MOD データ・セットのトレーラー・ラベルが処理されている場合 (データ・セットのオープン時)。

ユーザー・ラベルの読み取りまたは書き込み中に回復不能の入出力エラーが発生した場合は、該当するラベル処理ルーチンが、状況情報アドレス・パラメーターのフラグ 2 のビット 0 をオンに設定して実行されます。このパラメーターの 3 つの低位バイトには、SYNAD ルーチンで指定した標準状況情報のアドレスが入ります。(SYNAD ルーチンは実行されません。)

非ラベル処理ルーチンに渡されるパラメーター

表 62 は、非ラベル処理ルーチンに対する出口の取り出し元プログラム、出口の名前、各出口ルーチンで使用できるパラメーターをまとめたものです。

表 62. 非ラベル処理出口ルーチンのパラメーター・リスト

プログラム	出口	パラメーター
IEBGENER	KEY	キーが置かれるアドレス (キーの後にレコードが続く)。DCB のアドレス。
	DATA	SYSUT1 レコードのアドレス。DCB のアドレス。
	IOERROR	DECB のアドレス。エラーの原因と DCB のアドレス。(3 つの低位バイトにアドレス、高位バイトにエラーの原因。)
IEBCOMPR	ERROR	SYSUT1 の DCB のアドレス。SYSUT2 の DCB のアドレス。
	PRECOMP	SYSUT1 レコードのアドレス。SYSUT1 レコードの長さ。SYSUT2 レコードのアドレス。SYSUT2 レコードの長さ。
IEBPTPCH	INREC	入力レコードのアドレス。入力レコードの長さ。
	OUTREC	出力レコードのアドレス。出力レコードの長さ。

ユーザー・ラベルの処理

ユーザー・ラベルは、IEBCOMPR、IEBGENER、IEBPTPCH、IEBUPDTE、IEHMOVE によって処理されます。ユーザー・ラベル処理が自動的に実行される場合もあれば、ユーザーが処理の実行を指定しなければならない場合もあります。一般に、ユーザーとしては次のことを行えます。

- ユーザー・ラベルをデータ・セット記述子として処理すること。
- ユーザー・ラベルをデータとして処理すること。

- 処理後のレコードを書き込む前に合計すること (IEBGENER と IEBUPDTE の場合のみ)。

最初の 2 つのオプションについては、ユーザー・ラベル処理の対象となる各データ・セットを定義する DD ステートメントに SUL を指定しなければなりません。合計ルーチンの場合は、DD ステートメントに OPTCD=T を指定する必要があります。

IEBUPDTE プログラムによってラベルを更新することはできません。この機能は、ユーザー・ラベル処理ルーチンによって実行する必要があります。ただし、IEBUPDTE では、入力ストリーム内に指定したデータからの出力データ・セットにラベルを作成できます。

IEHMOVE では、ユーザー・ルーチンへの出口を使用できません。また、ユーザー・ラベルをデータとして処理するためのオプションを認識できません。IEHMOVE は、ユーザー・ラベルを新しいデータ・セットに直接移動したりコピーしたりします。

マルチボリューム・データ・セットのボリューム切り替えラベルは、IEHMOVE、IEBGENER、IEBUPDTE によって処理することはできません。これらのユーティリティーが出力データ・セットを作成すると、ボリューム切り替えラベルは失われます。ボリューム切り替えラベルを失わないようにするには、マルチボリューム・データ・セットの場合でも 1 つずつボリュームを処理してください。

ユーザー・ラベルをデータ・セット記述子として処理する

ユーザー・ラベルをデータ・セット記述子として処理する場合、ラベル処理ルーチンのいずれか 1 つが指定のタイプの各ユーザー・ラベルの制御を受け取ります。そのルーチンでは、ユーザー・ラベルを含めたり、除外したり、変更したりできます。ユーザー・ラベルをデータ・セット記述子として処理する場合は、EXITS ステートメントで、使用するラベル処理ルーチンの名前を示したキーワード・パラメーターを指定し、そのような処理を設定します。

ユーザー出口ルーチンは、OPEN、EOV、CLOSE のいずれかのルーチンが指定のタイプのユーザー・ラベルを検出するたびに、制御を受け取ります。

384 ページの図 54 は、OPEN、EOV、CLOSE の実行時にシステムが行う処置をまとめたものです。SUL がデータ・セットの DD ステートメントに指定されていた場合、OPEN、EOV、CLOSE のいずれかがユーザー・ラベルを認識すると、ユーティリティー・プログラムに制御が渡されます。さらに、このタイプのラベルに出口が指定されていた場合は、ユーティリティー・プログラムがユーザー・ルーチンに制御を渡します。そのルーチンは、ラベルを処理してから、戻りコードを付けてユーティリティー・プログラムに制御を戻します。さらに、ユーティリティー・プログラムは、OPEN、EOV、CLOSE のいずれかに制御を戻します。

このサイクルは、グループ内のユーザー・ラベルの数とルーチンから出された戻りコードに基づいて、最高 8 回まで繰り返されます。

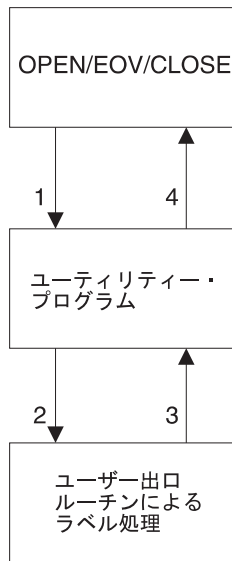


図 54. OPEN、EOV、CLOSE の実行時にシステムが行う処置

合計ルーチンへの出口

合計ルーチンへの出口が取られると、出力レコードが書き込まれる直前にルーチンに渡されます。パラメーターによって指示されている合計域の最初のハーフワードには、合計域の長さが入るので、その部分をルーチンが使用するべきではありません。ユーザー・ラベル出口を指定した場合は、該当するユーザー・ラベル・ルーチンに渡されるパラメーター・リストがこの合計域 (またはそのイメージ) を指すこととなります。

出力レコードは、IEBGENER を使用して、スパン・レコードが入ったデータ・セットを処理してフォーマットし直す場合を除き、物理レコード (ブロック) として定義されます。

合計ルーチンが戻すコードによって、図 55 に示すようなシステム応答が決定されます。

コード	意味
00 (X'00')	処理は続行されますが、これ以降、出口は取られません。
04 (X'04')	通常の処理が続行されます。
08 (X'08')	出力データ・セットの EOD 処理 (ユーザー・ラベル処理) 以外の処理は停止します。
16 (X'10')	処理は停止します。

図 55. ユーザー合計ルーチンの戻りコード

ユーザー・ラベルをデータとして処理する

ユーザー・ラベルをデータとして処理する場合、ユーザー・ラベルのグループのほかにデータ・セットも、ユーティリティー・プログラムによる通常の処理を受けます。IEBTPCH によってラベルを印刷または穿孔したり、IEBCOMPR によってラベルを比較したり、IEBGENER によってラベルをコピーしたりできます。

ユーザー・ラベルをデータとして処理することを指定するには、ユーザー・ラベルをデータとして処理するジョブ・ステップに LABELS ステートメントを組み込んでください。

LABELS ステートメントと EXITS ステートメントの間に直接の関係はありません。ユーティリティー・プログラムを実行するための制御ステートメント・ストリームに、いずれかを指定することも、両方を指定することも可能です。ただし、ユーザー・ラベル処理ルーチンがあると、その戻りコードによって、データとしてのラベル処理が影響を受けることもあります。さらに、ユーザー出力ラベル処理ルーチンは、各出力ラベルが書き込まれる前に制御を受け取るので、LABELS ステートメントの処置を指定変更することができます。その時点で、LABELS ステートメントの結果としてユーティリティーが作成したラベルがラベル・バッファーに入るので、その前にルーチンは、LABELS ステートメントを変更できます。

IEBDG での出力ルーチンの使用

IEBDG で提供されているユーザー出口によって、新しく構成したレコードを出力データ・セットに入れる前に、レコードを分析したり、さらに変更したりすることができます。この出力ルーチンは、CREATE ステートメントに指定します。

CREATE ステートメントでは、ユーザー・ルーチンから処理できるようにするレコードや、出力レコードとして直接書き込むレコードの内容を定義します。

ユーザー・ルーチンは、出力レコードになる各レコードを処理してから、出力レコードの処理方法を IEBDG に対して指定するための戻りコードをレジスター 15 に出すはずで、戻りコードを 386 ページの図 56 にリストアップします。

出力ルーチンがロードされ、ユーザーが IEBDG に制御を戻すと、レジスター 1 に、出力レコードの最初のバイトのアドレスが入ります。

コード	意味
00 (X'00')	このレコードが書き込まれます。
04 (X'04')	このレコードは書き込まれません。スキップされたレコードは、生成された出力レコードとしてはカウントされません。ただし、処理は、レコードが書き込まれたかのように続行されます。ユーザー出口によってスキップが要求され、入力レコードが指定された場合は、出力レコードの生成過程で、それぞれのスキップによって、余分の入力レコードが処理されることとなります。たとえば、CREATE ステートメントで、10 個の出力レコードを生成することを指定し、ユーザー出口で、2 個のレコードをスキップすることを指定した場合、12 個の入力レコードが処理されます。
12 (X'0C')	このユーティリティー制御ステートメント・セットの残りの処理が行われません。処理は、その次の DSD ステートメントで続行されます。
16 (X'10')	すべての処理が停止します。

図 56. IEBDG ユーザー出口の戻りコード

付録 D. IEHLIST の VTOC リスト表示

図 57 と 388 ページの図 58 は、IEHLIST の出力例です。最初の出力例である図 57 は、拡張フォーマット順次データ・セットのボリューム目録のリスト表示です。拡張フォーマット順次データ・セットの VTOC リスト表示は、順次データ・セット、区分データ・セット、PDSE の VTOC リスト表示とは異なります。388 ページの図 58 は、順次データ・セット、区分データ・セット、PDSE について IEHLIST から生成される典型的な VTOC リスト表示です。この 2 種類 (拡張フォーマット順次データ・セットとその他のデータ・セット) の VTOC リスト表示の主な違いは、出力の LST BLK フィールドにあります。さらに、IEHLIST によって拡張フォーマット順次データ・セットの VTOC リスト表示を生成する場合は、SMS.IND フィールドに属性 E が組み込まれます。拡張フォーマット順次データ・セットの特性の詳細については、*z/OS DFSMS* データ・セットの使用法を参照してください。

それぞれの図の後に、VTOC リスト表示の各フィールドの詳細な説明を載せています。LST BLK フィールドの説明の中に、2 種類の項目が入っていることにご注意ください。1 つは、拡張フォーマット順次データ・セット (TTTT-R) の説明であり、もう 1 つは、順次データ・セット、区分データ・セット、PDSE (T-R-L) の説明です。

なお、以下の図は、IEHLIST の VTOC リスト表示の例にすぎません。IEHLIST から生成される実際の VTOC リスト表示は異なります。

```

SYSTEMS SUPPORT UTILITIES---IEHLIST                                PAGE 1
DATE: 1992.260  TIME: 17.13.16
CONTENTS OF VTOC ON VOL 1P0401 <THIS IS AN SMS MANAGED VOLUME>
-----DATA SET NAME-----SER NO  SEQNO  DATE.CRE  DATE.EXP  DATE.REF  EXT  DSORG  RECFM  OPTCD  BLKSIZE
EXAMPLE.OF.FORMAT          1P0401    1  1992.260   00.000   1992.260   20  PS   F    00    20480
SMS.IND  LRECL  KEYLEN  INITIAL  ALLOC  2ND  ALLOC  EXTEND  LAST  BLK(TTTT-R)  DIR.REM  F2  OR  F3(C-H-R)  DSCB(C-H-R)
S   E   20480          TRKS          1          19          2
          EXTENTS  NO  LOW(C-H)  HIGH(C-H)  NO  LOW(C-H)  HIGH(C-H)  NO  LOW(C-H)  HIGH(C-H)
          0      0  4      0  4      1  1  0      1  0      2  1  1      1  1
          3      1  5      1  5      4  1  6      1  6      5  1  7      1  7
          6      1  8      1  8      7  1  9      1  9      8  1 10      1 10
          9      1 11      1 11     10  1 12      1 12     11  1 13      1 13
         12      1 14      1 14     13  2  0      2  0     14  2  1      2  1
         15      2  2          2  2
         16      2  3          2  3     17  2  4      2  4     18  2  5      2  5
         19      2  6          2  6
          ----ON THE ABOVE DATA SET, THERE ARE  0  EMPTY TRACK(S).

```

図 57. IEHLIST の出力例 – VTOC (拡張フォーマット順次データ・セット)

DATE: 1987.284 TIME: 18.44.59

CONTENTS OF VTOC ON VOL EXAMPL <THIS IS AN SMS MANAGED VOLUME>

THERE IS A 1 LEVEL VTOC INDEX

```

FORMAT 4 DSCB NO AVAIL/MAX DSCB /MAX DIRECT NO AVAIL NEXT ALT   FORMAT 6 LAST FMT 1   VTOC EXTENT   THIS DSCB
      VI  DSCBS  PER TRK  BLK PER TRK  ALT TRK  TRK(C-H)  (C-H-R)  DSCB(C-H-R)/LOW(C-H) HIGH(C-H)  (C-H-R)
      81  1587   53    46    15   885  0          3  14  53  2  0  3  14  2  0  1

```

```

-----DATA SET NAME----- SER NO  SEQNO  DATE.CRE DATE.EXP DATE.REF  EXT DSORG RECFM OPTCD BLKSIZE
EXAMPLE.OF.FORMAT          EXAMPL  1  1987.284  00.000  00.000  1  PS  F    00   2048
SMS.IND  LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC  EXTEND  LAST BLK(T-R-L) DIR.REM F2 OR F3(C-H-R)  DSCB(C-H-R)
SR       2048                TRKS CONTIG    0          28  18  0          2  0  3
EXTENTS  NO  LOW(C-H)  HIGH(C-H)
          0  0  1      1  14
-----ON THE ABOVE DATA SET, THERE ARE    0 EMPTY TRACK(S).

```

```

VPSM          A = NUMBER OF TRKS IN ADDITION TO FULL CYLS IN THE EXTENT
TRK  FULL    TRK  FULL    TRK  FULL    TRK  FULL    TRK  FULL
ADDR CYLS  A  ADDR CYLS  A  ADDR CYLS  A  ADDR CYLS  A  ADDR CYLS  A
  6C   881   0

```

THERE ARE 881 EMPTY CYLINDERS PLUS 0 EMPTY TRACKS ON THIS VOLUME
THERE ARE 1587 BLANK DSCBS IN THE VTOC ON THIS VOLUME
THERE ARE 518 UNALLOCATED VIRS IN THE INDEX

図 58. IEHLIST の出力例 - VTOC (順次データ・セット、区分データ・セット、PDSE)

IEHLIST のフォーマット済み VTOC リスト表示の各フィールドの説明

フィールド	説明
BLKSIZE	ブロック・サイズ (バイト単位)。最大値は 32760 か、装置ごとの限界値になります。 <ul style="list-style-type: none"> 固定長レコードの場合は、ブロック・サイズが設定されます。 可変長または不定長レコードの場合は、最大ブロック・サイズが示されます。 フォーマット V の非ブロック化レコードのブロック・サイズは、LRECL 値に 4 を加えた値になります。
DATA SET NAME	最大長 44 バイト。
DATE.CRE	ユリウス暦の yyyy.ddd フォーマットで表した、データ・セットの作成日付。 ddd は日、 yyyy は年 (1900~2155) です。
DATE.EXP	ユリウス暦の yyyy.ddd フォーマットで表した、データ・セットの有効期限日付。 ddd は日、 yyyy は年 (1900~2155) です。
DATE.REF	ユリウス暦の yyyy.ddd フォーマットで表した、データ・セットの最終参照日付。 ddd は日、 yyyy は年 (1900~2155) です。
DIR.REM	最後のディレクトリー・ブロックが使用中になっている区分データ・セットの場合、この値は、その 256 バイト・ブロックが使っているバイト数になります。ここに値が示されない場合は、区分データ・セットが最後のディレクトリー・ブロックまで到達していません。
DSORG	データ・セット編成 (アクセス方式ごと): <ul style="list-style-type: none"> DA = 直接 (BDAM) PO = 区分 (BPAM) PS = 物理順次 (SAM, QSAM, BSAM) <p>上記の編成方式の後に、以下の条件が入ることもあります。</p> <ul style="list-style-type: none"> U = 移動不能 (ロケーションに依存)
EXT	このボリューム上にデータ・セットが保有しているエクステント (セクション) の数。

フィールド	説明
EXTEND	<p>スペース要求のタイプがバイト、キロバイト、メガバイトのいずれかの場合は、元の 2 次割り振りの量。スペース要求のタイプが平均ブロックの場合は、元の平均ブロック長。以下の 2 文字の ID の後に、実際の 2 次割り振り値が続きます。</p> <ul style="list-style-type: none"> • AV 平均ブロック長 • BY 元の 2 次割り振り量 (バイト) • KB 元の 2 次割り振り量 (キロバイト) • MB 元の 2 次割り振り量 (メガバイト)
EXTENT NO LOW (C-H) HIGH (C-H)	各エクステントのシリンダーとヘッド (トラック) のアドレス。
FMT 2 OR 3 (C-H-R)/DSCB (C-H-R)	<p>ここには 2 つのアドレスが入りますが、それぞれが VTOC 内のデータ・セット制御ブロック (DSCB) を指します。右側のシリンダー・ヘッド (トラック) レコード・アドレスは必ず示され、現在見えている内容を含んだ DSCB を指します (フォーマット 1 の DSCB)。</p> <p>それに関連したフォーマット 2 またはフォーマット 3 の DSCB も示されることがあります。フォーマット 3 のアドレスは、エクステントの数が 3 個を超えたデータ・セットの場合にのみ示されるので、追加エクステントの情報を入れるために、フォーマット 3 の DSCB を使用しなければなりません。</p>
INITIAL ALLOC	<p>データ・セットのすべてのエクステントを割り振るために使用したスペース属性を記述します。</p> <ul style="list-style-type: none"> • RECS = 平均ブロック・サイズ • TRKS = トラック • BLKS = ブロック • CYLS = シリンダー • ABSTR = 絶対トラック (絶対アドレス)
KEYLEN	このデータ・セット内のデータ・レコードのキーのバイト長 (1~255) 0 は、キーが存在しないことを意味します。
LAST BLK PTR (T-R-L)	順次データ・セット、区分データ・セット、PDSE のいずれかに書き込まれた最後のブロックを指します。最初の数値 (通常 4 桁、この例では 2 桁) は、データ・セットの先頭からの相対トラックです。2 番目の数値は、トラックの先頭からの相対ブロックです。最後の数値は、そのブロックの後のトラックの残りのバイト数です。データ・セットが PDSE であれば、フィールドがブランクになる場合もあります。
LAST BLK PTR (TTTT-R)	拡張フォーマット順次データ・セットに書き込まれた最後のブロックを指します。最初の数値 (通常 4 桁、この例では 2 桁) は、データ・セットの先頭からの相対トラックです。2 番目の数値は、トラックの先頭からの相対ブロックです。
LRECL	<p>論理レコードの長さ (バイト単位)。最高値は、非スパン・レコードで 32760、スパン・レコードで 32756 になります。</p> <ul style="list-style-type: none"> • 固定長レコードの場合は、LRECL が実際のレコード長になります。 • 可変長レコードの場合は、LRECL がその装置の最大許容長になります。 • 不定長レコードの場合は、LRECL がゼロになります。
OPTCD	オプション・コード (データ・セットの作成に使用した DCB で指定した値)。この 1 バイト・コードは、16 進文字で表されます。 <i>z/OS DFSMSdfp</i> 拡張サービス にある、DSCB1 データ・セットの DS1OPTCD フィールドを参照してください。

フィールド	説明
RECFM	レコード・フォーマット: <ul style="list-style-type: none"> • F = 固定長 • V = 可変長 • D = ASCII 可変長 (ディスク上では無効) • U = 不定長 以下のオプションも指定できます。 <ul style="list-style-type: none"> • B = ブロック化レコード • S = スパン・レコード • T = 許容されるトラック・オーバーフロー • A = ISO/ANSI 制御文字 • M = マシン制御文字
SEQNO	データ・セットが入っている最初のボリュームから見た、このボリュームの相対順位。(マルチボリューム・データ・セットの場合以外、SEQ NO は 1 に等しくなります。)
SER NO	データ・セットが入っているボリュームの通し番号。最大長は 6 バイトです。(データ・セットが書き込まれてからボリュームの名前が変更されている場合は、通し番号が変わることがありますが、このフィールドは、データ・セットのそれぞれのフォーマット 1 DSCB について共通の値でなければなりません。)
SMS.IND	システム管理ストレージ属性。 <ul style="list-style-type: none"> • A = データ・セットに対して拡張属性が存在する • B = DADSM 作成時に選択された最適ブロック・サイズ • C = 圧縮フォーマット • E = 拡張フォーマット • H = HFS タイプのデータ・セット • I = データ・セットは PDSE • L = ラージ・フォーマットのデータ・セット • R = データ・セットは再ブロック可能 • S = SMS 管理データ・セット • U = データ・セットの BCS 項目が存在しない • ? = 以下のいずれかです。 <ul style="list-style-type: none"> – PDSE と拡張フォーマット順次データ・セットは共存できません。両方のビットがオンになっている場合、1 つをオフにする必要があります。 – 圧縮拡張フォーマット・データ・セットのビットがオンの場合、拡張フォーマット順次データ・セットのビットはオンになっている必要があります。
2ND ALLOC	2 次割り振り量。ゼロの場合は、データ・セットが 1 次割り振りエクステントに限定されます。それ以外の場合は、必要に応じてさらに多くのエクステントに拡張できます。それぞれのエクステントは、ブロック、トラック、シリンダーのいずれかのサイズがこの数値になります。 各種の制限があります。

付録 E. アクセシビリティ

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。z/OS のアクセシビリティの主要機能により、ユーザーは以下のことができるようになります。

- 画面読み上げ機能および画面拡大機能などの支援機能の使用
- キーボードのみを使用して、特定の機能または画面を使用したのと同等の機能を操作
- 色、コントラスト、フォント・サイズなど表示属性のカスタマイズ

支援機能の使用

画面読み上げ機能などの支援機能は、z/OS のユーザー・インターフェースを使用して機能します。この支援機能を使用して z/OS インターフェースにアクセスする場合、その特定情報については支援機能の資料を参照してください。

ユーザー・インターフェースのキーボード・ナビゲーション

ユーザーは、TSO/E または ISPF を使用して z/OS ユーザー・インターフェースにアクセスできます。TSO/E および ISPF インターフェースへのアクセス方法については、*z/OS TSO/E 入門*、*z/OS TSO/E ユーザーズ・ガイド*、および *z/OS ISPF ユーザーズ・ガイド 第 1 巻* を参照してください。上記の資料には、キーボード・ショートカットまたはファンクション・キー (PF キー) の使用方法を含む TSO/E および ISPF の使用方法が記載されています。それぞれの資料では、PF キーのデフォルトの設定値とそれらの機能の変更方法についても説明しています。

z/OS 情報

z/OS の情報については、スクリーン・リーダーを使用して、以下のインターネット・ライブラリーにある z/OS ブックの BookServer/Library Server 版が利用できます。

www.ibm.com/servers/eserver/zseries/zos/bkserv/

1 つだけ例外があります。それはコマンド構文であり、線路形式で出版されます。すなわち、その構文情報が記載された z/OS 資料の画面読取可能コピーが、mhvrdfs@us.ibm.com への要求に従って ZIP 状態の HTML ファイル形式で個別に入手可能です。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

プログラミング・インターフェース情報

本書の目的は、DFSMS ユーティリティ・プログラムを使用する手助けをすることです。

本書は、DFSMS が提供する汎用プログラミング・インターフェースとそれに関する情報も記述しています。汎用プログラミング・インターフェースにより、お客様が DFSMS の機能を使用するプログラムを書くことができます。

汎用プログラミング・インターフェースおよびそれに関連する情報は、章または節の単位の場合はその冒頭で識別され、それ以外の場合は「汎用プログラミング・インターフェース情報とそれに関連する情報」という表記で識別されています。

本書には、特に注記しない限り、プログラミング・インターフェース情報は記載されていません。ただし、付録 A、B、および C は例外です。

商標

以下は、IBM Corporation の商標です。

3890	ESCON
DFSMS	IBM
DFSMS/MVS	MVS
DFSMSdfp	MVS/ESA
DFSMSdss	OS/390
DFSORT	RACF

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

用語集

この用語集は、DFSMS 資料で使用される技術用語と省略語を定義しています。探している用語が見つからない場合は、該当する DFSMS 資料の索引を参照するか、または下記のサイトにある *Glossary of Computing Terms* を参照してください。

<http://www.ibm.com/ibm/terminology/>

この用語集には、以下のものを出典とする用語と定義が含まれています。

- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990, 米国規格協会 (ANSI)。これは、American National Standards Institute, 11 West 42nd Street, New York, New York 10036 から購入することができます。これらの定義のあとには、シンボル (A) が付いています。
- *The Information Technology Vocabulary*。国際標準化機構および国際電気標準会議 (ISO/IEC JTC1/SC1) の第 1 合同技術委員会第 1 小委員会によって作成された資料。この用語集の中で出版されたものの定義の後にはシンボル (I) が付いています。国際標準案、委員会案、および ISO/IEC JTC1/SC1 で作成中の作業文書から取られた定義の後には、シンボル (T) が付いており、SC1 参加国間の最終合意がまだ得られていないことを示しています。
- *IBM Dictionary of Computing* (New York: McGraw-Hill, 1994)。

この用語集では、以下の相互参照を使用しています。

参照： これは、(a) 関連している用語、(b) 省略語または頭字語の拡張形式である用語、または (c) 同義語またはより適切な用語を示しています。

[ア行]

アクティブ制御データ・セット (Active Control Data Set (ACDS))。活動化された最新の SMS 構成とその後の更新情報が入っている VSAM 線形データ・セット。ACDS は、ストレージを管理するために同じ SMS 構成

を使用している各システムによって共有される。ソース制御データ・セット (*source control data set*) および通信データ・セット (*communications data set*) も参照。

アクティブ・データ (active data)。(1) 1 次ストレージ上のデータあるいはマイグレーション済みデータなど、ユーザーの特別なアクションがなくても利用できるデータ。アクティブ・データは、テープ・ボリュームにも保管できる。(2) 磁気テープ取り付け管理の場合、頻繁に参照され、サイズが小さく、磁気テープよりも DASD (直接アクセス記憶装置) で管理するほうが望ましいアプリケーション・データ。(3) 非アクティブ・データ (*inactive data*) と対比。

圧縮 (compress)。(1) システムが同一の語句をその語句に関連するもっと短いトークンで置き換えることによって、特定のデータ・セットに必要なストレージの量を少なくすること。(2) 未使用の全スペースをデータ・セットの最後に移動し、メンバーを削除または変更して、区分データ・セット内の未使用および使用不能なスペースをレクラメーション処理すること。

圧縮フォーマット (compressed format)。データ・クラス (COMPACTION) パラメーターで指定される拡張フォーマット・データ・セットの特定のタイプ。VSAM は、圧縮フォーマット・データ・セットの中の個々のレコードを圧縮することができる。SAM は、圧縮フォーマット・データ・セットの個々のブロックを圧縮することができる。「圧縮 (*compress*)」を参照。

アンロードされる区分データ・セット (PDSU) (partitioned data set unloaded (PDSU))。IEBCOPY アンロード・データ・セット。PDS を作成するために IEBCOPY が復元できる順次ファイル。

一時データ・セット (temporary data set)。& または && で始まる名前のアンカタログ・データ・セット。通常は、ジョブの実行中または対話式セッションにおいてのみに使用される。「永続データ・セット (*permanent data set*)」と対比。

移動不能データ・セット (unmovable data set)。アプリケーションが必要とする移動不能の DASD データ・セット。移動不能データ・セットは、再配置されないようにシステムから識別される必要がある。移動不能データ・セットを識別するには、そのデータ・セットを絶対トラックに割り振るか、移動不能属性を持ったデータ・セット編成でそのデータ・セットを割り振る。たとえ

ば、PSU、POU、DAU、ABSTR のいずれかとして割り振られたデータ・セットは、移動不能と見なされる。

永続データ・セット (permanent data set). 通常、ジョブの期間または対話式セッションの所要時間より長い間保存される、ユーザーによって名前が付けられたデータ・セット。「一時データ・セット (*temporary data set*)」と対比。

エラー・リカバリー (error recovery). インストール環境の業務で必須のデータを安全な場所にコピーおよび保管し、インストール環境にエラーが発生した場合にそのデータを回復するための手順。「重要レコード (*vital records*)」と対比。

オブジェクト (object). 特定の形式または記録方向のない名前付きバイト・ストリーム。

オブジェクト記憶階層 (object storage hierarchy). DB2 テーブル・スペースに保管されたオブジェクトから成る階層。これは、DASD、ライブラリーにある光ディスクまたは磁気テープ・ボリューム、シェルフ上の光ディスクまたはテープ・ボリューム上にある。「ストレージ階層 (*storage hierarchy*)」も参照。

オブジェクト記憶グループ (object storage group). DASD 上のオブジェクト、磁気テープ、または光ディスク・ボリュームが含まれるストレージ・グループのタイプ。「ストレージ・グループ (*storage group*)」も参照。

オブジェクト記憶テーブル (object storage tables). オブジェクトを含む DB2 テーブルの集合体。

オブジェクト・アクセス方式 (OAM) (Object Access Method (OAM)). オブジェクトに対してストレージ、検索、ストレージ階層の管理機能を提供し、システム管理ライブラリーに含まれているテープ・ボリュームに対してストレージ管理および検索管理を提供する。

オブジェクト・ディレクトリー・テーブル (object directory tables). オブジェクト記憶グループに保管されていたオブジェクトに関する情報を含む DB2 テーブルの集合体。

オブジェクト・バックアップ・ストレージ・グループ (object backup storage group). オブジェクトのバックアップ・コピーに使用される、光ディスクまたは磁気テープ・ボリュームを含むストレージ・グループのタイプ。「ストレージ・グループ (*storage group*)」も参照。

[力行]

カートリッジ・システム・テープ (Cartridge System Tape). 3480 または 3490 磁気テープ・サブシステムで使用される基本テープ・カートリッジ・メディア。「拡張容量カートリッジ・システム・テープ (*Enhanced Capacity Cartridge System Tape*)」と対比。

拡張区分データ・セット (PDSE) (partitioned data set extended (PDSE)). 区分データ・セットのディレクトリーとメンバーによく似た索引付きディレクトリーとメンバーが入っているシステム管理データ・セット。PDSE は区分データ・セットの代わりに使用することができる。

拡張フォーマット (extended format). EXTENDED のデータ・セット名タイプ (DSNTYPE) のデータ・セットのフォーマット。データ・セットの論理的な構成は拡張フォーマットでないデータ・セットと同じだが、物理的なフォーマットは異なる。拡張フォーマットのデータ・セットは、ストライプまたは圧縮可能である。拡張フォーマット VSAM KSDS のデータは、圧縮可能である。「ストライプ・データ・セット (*striped data set*)」、
「圧縮フォーマット (*compressed format*)」も参照。

拡張容量カートリッジ・システム・テープ (Enhanced Capacity Cartridge System Tape). 3490E 磁気テープ・サブシステムでのみ使用できる、容量を大きくしたカートリッジ・システム・テープ。カートリッジ・システム・テープ (*Cartridge System Tape*) と対比。

格納位置 (shelf location). (1) シェルフ上にある、取り外し可能メディアのストレージ用の単一スペース。(2) DFSMSrmm において、格納位置は、取り外し可能メディア・ライブラリー内ではラック番号によって指定され、保管場所ではビン番号によって指定される。「ラック番号 (*rack number*)」、
「ビン番号 (*bin number*)」も参照。

仮想入出力 (VIO) ストレージ・グループ (Virtual Input/Output (VIO) storage group). データ・セットをページング記憶装置に割り振るストレージ・グループのタイプ。これは、DASD ボリュームをシミュレートする。VIO ストレージ・グループには、実 DASD ボリュームは含まれていない。「ストレージ・グループ (*storage group*)」も参照。

稼働中の追加インストール (nondisruptive installation). サブシステムに追加の装置をインストールする場合、ユーザーが既存のストレージ・サブシステム内のデータにアクセスできるようにする機能。

可用性 (availability). 記憶サブシステムにおいて、ユーザー要求時に、データ・セットまたはオブジェクトにアクセスできる程度のこと。

管理クラス (management class). ストレージ管理者によって定義された管理属性の集合体。割り振られているが未使用のスペースの解放の制御、データ・セットの保存 / マイグレーション / バックアップの制御、集合グループの保存とバックアップの制御、オブジェクトの保存 / バックアップ / クラス遷移の制御に使用される。

疑似光ディスク・ライブラリー (pseudo optical library). オペレーター・アクセス可能な光ディスク・ドライブに関連する、格納光ディスク・ボリュームの集合。「実光ディスク・ライブラリー (real optical library)」も参照。

基本カタログ構造 (BCS) (basic catalog structure (BCS)). カタログ環境におけるカタログ構造名。

基本構成 (base configuration). SMS 構成のうち、デフォルト管理クラス、デフォルト装置、デフォルト装置配置など、一般的なストレージ管理属性を含む部分。SMS 構成が管理するシステムまたはシステム・グループも識別する。

- | **基本フォーマット.** BASIC のデータ・セット名タイプ
- | (DSNTYPE) のデータ・セットのフォーマット。基本フ
- | ォーマットのデータ・セットは順次データ・セットであ
- | り、ラージ・フォーマットまたは拡張フォーマットのい
- | ずれでもないデータ・セットとして指定される。基本フ
- | ォーマットのデータ・セットは、各ボリューム上で
- | 65535 トラックを超えることはできません。

キャッシュ高速書き込み (cache fast write). データを、不揮発性ストレージを使用せずに直接キャッシュに書き込むストレージ管理機能。キャッシュ高速書き込みは、一時データや DFSORT により作成されるソート作業ファイルなど、容易に再作成されるデータに対して便利。DASD 高速書き込み (DASD fast write) と対比。

区分データ・セット (partitioned data set (PDS)). それぞれにプログラム、プログラムの一部、またはデータを入れることができるメンバーと呼ばれる区画に分割された直接アクセス記憶装置のデータ・セット。

クラス遷移 (class transition). オブジェクトのサービス・レベル基準の変更を引き起こすイベント。OAM を使用して ACS ルーチンを起動し、新しいストレージ・クラスまたは管理クラスをオブジェクトに割り当てる。

グループ (group). (1) 区分データ・セットに関しては、PDS、PDSE、またはアンロードされた PDSE にあるメンバーおよびメンバーの別名。(2) 保護リソースに対するアクセス権限を共用できるユーザーの集合体。

グローバル・アクセス検査 (global access checking). RACF において、選択されたリソースの許可レベルを含むデフォルト値のストレージ内テーブルを設定する機能。RACF は、通常の RACHECK 処理を実行する前にこのテーブルを参照し、要求されたアクセス権限がグローバル値を超えていなければ、RACHECK を実行せずに要求を認可する。グローバル・アクセス検査は、リソースへのユーザー・アクセスを認可することができるが、アクセスを拒否することはできない。

グローバル・リソースの逐次化 (GRS) (Global Resource Serialization (GRS)). システム・リソースの使用の逐次化のため、また DASD ポリューム上でのハードウェア予約をデータ・セット待ち行列に変換するために使用される z/OS のコンポーネント。

[サ行]

サービス水準報告プログラム (SLR) (Service Level Reporter (SLR)). データ処理インストール・システムの管理のために、調整されたツールと手法、および一貫性のある情報をユーザーに提供する IBM ライセンス・プログラム。たとえば、SLR は SMF、IMS、CICS ログから情報を抽出し、選択した情報を表作成またはグラフィック・レポート用に形式設定し、データベース・テーブルの保守を援助する。

サービス・レベルの合意 (service-level agreement). (1) ユーザーが必要とするスペース、可用性、パフォーマンス、セキュリティを保証するために、ストレージ管理グループが提供するサービス・レベルを定義する、ストレージ管理グループとユーザー・グループの間の合意。(2) ストレージ管理グループが要求するストレージ管理ジョブの完了を保証するために、さまざまな操作で提供されるサービス・レベルの操作を定義する、ストレージ管理グループとさまざまな操作の間の合意。

シェルフ (shelf). 磁気テープおよび光ディスク・ボリュームなどの取り外し可能メディアの書き込みと読み取りが実行されていない場合に、それらを保管するための場所。

磁気テープ取り付け管理 (tape mount management). 磁気テープ・サブシステムの操作および使用を最適化するために使用される方法論。磁気テープ・データを効率的に管理するために使用されるハードウェアおよびソフトウェア機能で構成される。

磁気テープ・サブシステム (tape subsystem). 磁気テープ・サブシステムはコントローラーおよび装置から構成され、ユーザー・データをテープ・カートリッジに保管

することができる。磁気テープ・サブシステムの例として、IBM 3490 および 3490E 磁気テープ・サブシステムが含まれる。

磁気テープ・ストレージ・グループ (tape storage group). システム管理専用テープ・ボリュームを含むストレージ・グループのタイプ。磁気テープ・ストレージ・グループ定義は、テープ・ボリュームを入れることのできるシステム管理テープ・ライブラリーを指定する。「ストレージ・グループ (storage group)」も参照。

磁気テープ・ライブラリー (tape library). インストール・システムの磁気テープ環境をサポートする装置と設備のセット。これには、磁気テープ・ストレージ・ラック、磁気テープ・ドライブのセット、それらのドライブに取り付ける互いに関係のあるテープ・ボリュームのセットが含まれる。「システム管理テープ・ライブラリー (system-managed tape library)」、「自動化テープ・ライブラリー・データ・サーバー (automated tape library data server)」も参照。

磁気テープ・ライブラリー・データ・サーバー (Tape Library Dataserver). 磁気テープ・ドライブのセットに関連する磁気テープ・インベントリを保守するハードウェア装置。自動化テープ・ライブラリー・データ・サーバーは、磁気テープの取り付け、取り外し、ストレージも管理する。

磁気テープ・ライブラリアン (tape librarian). 磁気テープ・ライブラリーを管理する担当者。この担当者は、特殊なストレージ管理者である。

システム管理機能 (SMF) (system management facilities (SMF)). 入出力 (I/O) 統計情報を収集する z/OS のコンポーネント。データ・セットおよびストレージ・クラス・レベルで提供され、直接アクセス記憶装置サブシステムのパフォーマンスをモニターするのに役立つ。

システム管理ストレージ (system-managed storage). ストレージ管理サブシステムにより管理されるストレージ。SMS は、使用可能度、パフォーマンス、およびスペースの必要なサービスをアプリケーションに引き渡そうとする。

システム管理データ・セット (system-managed data set). ストレージ・クラスを割り当てられたデータ・セット。

システム管理テープ・ライブラリー (system-managed tape library). 磁気テープ構成データベースの中で定義される、磁気テープ・ボリュームおよび磁気テープ装置の集合体。システム管理テープ・ライブラリーには、自動または手動がある。「磁気テープ・ライブラリー (tape library)」も参照。

システム管理ボリューム (system-managed volume). ストレージ・グループに属する DASD、光ディスク、またはテープ・ボリューム。「DFSMSHsm 管理ボリューム (DFSMSHsm-managed volume)」、「DFSMSRmm 管理ボリューム (DFSMSRmm-managed volume)」と対比。

システム・データ (system data). z/OS またはそのサブシステムで初期化および制御のために必要なデータ・セット。

システム・プログラマー (system programmer). インストール・システム全体の生産性向上を目指して、オペレーティング・システムおよびアプリケーションの使用を計画、生成、保守、拡張、および制御するプログラマー。

実光ディスク・ライブラリー (real optical library). 光ディスク・ドライブおよび光ディスク・カートリッジを内蔵し、カートリッジ・ストレージと光ディスク・ドライブの間で光ディスクを移動させるためのメカニズムを含む物理ストレージ装置。「疑似光ディスク・ライブラリー (pseudo optical library)」と対比。

自動 1 次スペース管理 (automatic primary-space management). DFSMSHsm において、有効期限切れデータ・セットの削除、一時データ・セットの削除、未使用スペースの解放、および 1 次記憶ボリュームからのデータ・セットのマイグレーションを自動的に実行するプロセス。

自動 2 次スペース管理 (automatic secondary-space management). DFSMSHsm において、有効期限切れのマイグレーション済みデータ・セットの削除、マイグレーション制御データ・セットからの有効期限切れレコードの削除、マイグレーション・レベル 1 のボリュームからマイグレーション・レベル 2 のボリュームへのマイグレーション基準に達したデータ・セットのマイグレーションを自動的に実行するプロセス。

自動化テープ・ライブラリー・データ・サーバー (automated tape library data server). ロボット・コンポーネント、カートリッジ・ストレージ、磁気テープ・サブシステムから構成され、ハードウェアおよびソフトウェアを制御する装置。これには、ライブラリー内に存在し、ライブラリー磁気テープ装置に取り付けられるテープ・ボリュームのセットも含まれる。「手動テープ・ライブラリー (manual tape library)」と対比。「磁気テープ・ライブラリー (tape library)」も参照。

自動クラス選択ルーチン (Automatic Class Selection (ACS) routine). 手続き型 ACS (自動クラス選択) 言語ステートメントのセット。ACS 言語ステートメントは入力変数のセットに基づいて、データ・セットに関して

事前定義されている SMS クラスの名前、または事前定義されているストレージ・グループの名前のリストを生成する。

自動ダンプ (automatic dump). DFSMSShsm において、1 次ストレージ・ボリューム上に割り振られたすべてのスペースの全ボリュームを、指定されたテープ・ダンプ・ボリュームにダンプする処理を、DFSMSdss を使用して自動的に実行するプロセス。

自動データ・セット保護 (ADSP) (Automatic Data Set Protection (ADSP)). z/OS において、ユーザーによって作成されたすべての永続データ・セットが、個別 RACF (リソース・アクセス管理機能) プロファイルを使用して、RACF に対して自動的に定義されるようにするユーザー属性。

自動バックアップ (automatic backup). (1) DFSMSShsm において、1 次ストレージ・ボリュームまたはマイグレーション・ボリュームから、バックアップ・ボリュームにデータ・セットを自動的にコピーするプロセス。(2) OAM (オブジェクト・アクセス方式) において、DASD、光ディスク、または磁気テープ・ボリュームから、オブジェクト・バックアップ・ストレージ・グループに含まれているバックアップ・ボリュームに、自動的にオブジェクトの 1 次コピーをコピーするプロセス。

自動ボリューム・スペース管理 (automatic volume-space management). DFSMSShsm において、自動 1 次スペース管理および間隔マイグレーションを含むプロセス。

集合グループ (aggregate group). 定義されたバックアップまたはリカバリー・ストラテジーに適合するようにプールされていた関連のデータ・セットおよび制御情報の集合体。

集合バックアップ (aggregate backup). 集合グループのコピー処理、およびデータ・セットの集合体を後でグループとしてリカバリーできるようにするためのリカバリー命令。

重要レコード (vital records). 法的な要件など、外的に課せられた保存要件に合致させるために保守されるデータ・セットまたはボリューム。「災害時回復 (disaster recovery)」と対比。

重要レコード指定 (vital record specification). 災害時回復および重要レコードのためのデータ・セットおよびボリュームの保存と移動を管理するために定義したポリシー。

手動テープ・ライブラリー (manual tape library). 手動テープ・ライブラリーは、論理ユニットとして定義された磁気テープ・ドライブのインストール・システム定

義のセット、およびそれらのドライブに装着できるシステム管理ボリュームのセット。

順次データ・ストライプ (sequential data striping). パフォーマンスを上げるために複数のボリュームにデータ・セットを配布するディスク・アレイのソフトウェア・インプリメンテーション。

使用属性 (use attribute). (1) 新しいデータ・セットを割り振る際にボリュームが使用される場合、制御する DAD ボリュームに割り当てられる属性。使用属性には、*public*、*private*、*storage* がある。(2) システム管理テープ・ボリュームの場合、使用属性は *scratch* と *private*。

ストライプ (stripe). DFSMS において、ストライプ・データ・セットのうち、1 つのボリューム上にある部分 (たとえば、拡張形式データ・セット)。その部分のレコードは、必ずしも論理的に連続していない。パフォーマンスを上げるために複数のボリュームを同時に読み書きできるように、システムはいくつかのストライプにレコードを分散させる。データ・セットがストライピングされているかどうかは、アプリケーション・プログラムにはわからない。

ストライプ・データ・セット (striped data set). DFSMS において、複数のストライプから構成される拡張フォーマット・データ・セット。SMS は、そのストレージ・クラスの SUSTAINED DATA RATE の値に基づいて、使用するストライプの数を決定する。ストライプ・データ・セットには、順次データ・ストライプ・アクセス技法の利点がある。「ストライプ (*stripe*)」、
「ストライピング (*striping*)」を参照。

ストレージ階層 (storage hierarchy). 異なる速度および容量のさまざまなストレージ・デバイスの配置。ストレージ階層のレベルには、主記憶域 (メモリー、DASD キャッシュ)、1 次ストレージ (非圧縮データを含む DASD)、マイグレーション・レベル 1 (スペース節約フォーマットのデータを含む DASD)、マイグレーション・レベル 2 (スペース節約形式のデータを含むテープ・カートリッジ) がある。「1 次ストレージ (*primary storage*)」、「マイグレーション・レベル 1 (*migration level 1*)」、「マイグレーション・レベル 2 (*migration level 2*)」、「オブジェクト記憶階層 (*object storage hierarchy*)」も参照。

ストレージ管理 (storage control). プロセッサ・チャネルとストレージ・デバイスの間の対話を操作したり、チャネル・コマンドを実行したり、ストレージ・デバイスを制御したりするストレージ・サブシステム内のコンポーネント。

ストレージ管理 (storage management). データ・セットの割り振り、配置、モニター、マイグレーション、バックアップ、リコール、リカバリー、および削除のアクティビティ。これらは、手動あるいは自動化処理のどちらでも可能。それらの処理は、ストレージ管理サブシステムによって自動化され、ストレージ・リソースの最適化もなされる。「ストレージ管理サブシステム (Storage Management Subsystem)」も参照。

ストレージ管理グループ (storage administration group). インストール・システム内のストレージ・リソース管理を担当するデータ処理センター内の集中化されたグループ。

ストレージ管理サブシステム (SMS) (Storage Management Subsystem (SMS)). ストレージの管理を自動化し、集中化するために使用される DFSMS の機能。SMS を使用することによってストレージ管理者は、データ割り振り特性、パフォーマンスおよび可用性の目標、バックアップおよび保存要件、およびストレージ所要量をデータ・クラス、ストレージ・クラス、管理クラス、ストレージ・グループ、および ACS ルーチン定義を通じてシステムに記述する。

ストレージ管理者 (storage administrator). ストレージ管理ポリシーの定義、実施、および維持を担当するデータ処理センターの管理者。

ストレージ・クラス (storage class). パフォーマンスの目標と可用性要件を示すストレージ属性の集まり。ストレージ管理者により定義され、パフォーマンス目標と要件を満たすことができる装置を選ぶのに使用される。

ストレージ・グループ (storage group). ストレージ管理者が定義したストレージ・ボリュームおよび属性の集合体。この集合は、DASD ボリュームまたは磁気テープ・ボリュームのグループ、あるいは DASD、光ディスク、または磁気テープ・ボリュームのグループで、単一オブジェクト記憶階層として扱われる。「VIO ストレージ・グループ (VIO storage group)」、「プール・ストレージ・グループ (pool storage group)」、「磁気テープ・ストレージ・グループ (tape storage group)」、「オブジェクト記憶グループ (object storage group)」、「オブジェクト・バックアップ・ストレージ・グループ (object backup storage group)」、「ダミー・ストレージ・グループ (dummy storage group)」も参照。

ストレージ・グループ・カテゴリー (storage group category). 同じタイプのデータが入っている特定のストレージ・グループのグループ化。この概念は、非システム管理環境におけるストレージ・プールに類似している。

ストレージ・サブシステム (storage subsystem). ストレージ管理およびそれに付加されたストレージ・デバイス。「磁気テープ・サブシステム (tape subsystem)」も参照。

ストレージ・ディレクター (storage director). 3990 記憶制御装置において、同じストレージ・クラスター内の 1 つまたは複数の物理ストレージ・パスから構成される 1 つの論理エンティティ。3880 において、ストレージ・ディレクターはストレージ・パスに相当する。

接続性 (connectivity). (1) データ使用可能性の要求を満たす適切なデータ・パス (および代替データ・パス) を実現するために、ストレージ管理を DASD およびプロセッサに結合する方法に関する考慮事項。(2) システム管理ストレージ環境において、ボリュームおよびストレージ・グループのシステム状況。

ソース制御データ・セット (SCDS) (Source Control Data Set (SCDS)). SMS 構成を含む VSAM 線形データ・セット。SCDS 内の SMS 構成は、ISMF を使用して変更および妥当性検査を実行できる。「アクティブ制御データ・セット (active control data set)」、「通信データ・セット (communications data set)」も参照。

総称装置名 (generic unit name). ジオメトリ (形状) が同じである装置のクラスに割り当てられる名前 (3390 など)。「非公式装置名 (esoteric unit name)」と対比。

装置サポート機能 (ICKDSF) (Device Support Facilities (ICKDSF)). DASD ボリュームの初期化およびトラック・リカバリーに使用されるプログラム。

[夕行]

妥当性検査 (validate). 個々の ACS ルーチンまたは SMS 構成全体の完全性および整合性を検査する。

ダミー・ストレージ・グループ (dummy storage group). システムに接続されていない、ボリューム・シリアル番号を含むストレージ・グループのタイプ。ダミー・ストレージ・グループによって、既存の JCL を変更せずに機能させることができる。「ストレージ・グループ (storage group)」も参照。

ダンプ・クラス (dump class). ボリューム・ダンプが DFSMSShsm によって管理される方法を記述する特性の集合。

通信データ・セット (communications data set (COMMDS)). 単一の SMS 構成によって制御される、システム間通信の主要な手段。COMMDS は VSAM 線形データ・セットで、ACDS の名前および各システム管理ボリュームの現行の使用状況の統計が含まれている。

これは、SMS が稼働しているシステム相互間でスペースのバランスを調整するのに役立つ。「アクティブ制御データ・セット (active control data set)」 および「ソース制御データ・セット (source control data set)」 も参照。

データ機能分類プログラム (Data Facility Sort). 高速データ処理ユーティリティである IBM ライセンス・プログラム。DFSORT は、ソート、マージ、およびコピーの際に効率的で柔軟性のある操作性を提供し、またレコード、フィールド、およびビット・レベルのさまざまなデータ操作も提供する。

データ・クラス (data class). ストレージ管理者により定義され、データ・セットの作成に使用される、割り振りおよびスペース属性の集合体。

データ・セット (data set). DFSMS において、データ・ストレージおよび検索の主な単位。いくつかの指示された配置の 1 つになっているデータの集まりで構成され、システムがアクセスできる制御情報によって記述される。z/OS 非 UNIX 環境では、データ・セットとファイルという用語は一般に同じ意味であり、互いに交換可能な語として使用される場合がある。「ファイル (file)」も参照。z/OS UNIX 環境では、データ・セットとファイルという用語の意味はまったく違う。

テープ構成データベース (tape configuration database). システム管理テープ・ライブラリーおよびテープ・ボリュームのレコードの保守に使用される、1 つ以上のボリューム・カタログ。

テープ・ボリューム (tape volume). テープ・ボリュームは、単一のテープ・カートリッジまたはリール上の記録スペース。「ボリューム (volume)」も参照。

デフォルト管理クラス (default management class). SMS (ストレージ管理サブシステム) 基本構成の一部であり、管理クラスが割り当てられていないシステム管理データ・セットで使用される管理クラスを識別する。

デフォルト装置 (default unit). SMS (ストレージ管理サブシステム) 基本構成の一部であり、非公式 (SYSDA など) または汎用 (3390 など) 装置名を識別する。ユーザーが JCL 上の UNIT パラメーターまたは動的割り振りでそれに相当するものを省略した場合、データ・セットが MOD または NEW の後処置を実行し、またシステム管理データ・セットではない場合、SMS はデフォルト装置を適用する。

デフォルト装置配置 (default device geometry). SMS (ストレージ管理サブシステム) 基本構成の一部であり、装置名が指定されていない場合に、トラックまたはシリ

ンダー単位のスペース要求をバイト単位に変換するために、トラック当たりのバイト数とシリンドラー当たりのトラック数を識別する。

同所変換 (convert in place). 「同所変換 (in-place conversion)」を参照。

動的キャッシュ管理 (dynamic cache management). 3990 サブシステムの負荷、データ・セットの特性、およびストレージ管理者によって定義されたパフォーマンス要件に基づき、どのデータ・セットをキャッシュに入れるかを自動的に判別する機能。

ドライブ定義 (drive definition). 実光ディスク・ライブラリーまたは疑似光ディスク・ライブラリーのメンバーとして、光ディスク・ドライブを定義する際に使用される属性のセット。

取り外し可能メディア・ライブラリー (removable media library). 即時に使用可能なボリューム、およびボリュームを置けるシェルフ。

[ナ行]

二重化 (duplexing). データの第 2 のコピーを作成するために、同一レコードについて 2 つのセットを書き込むプロセス。

二重コピー (dual copy). IBM 3990 磁気ディスク制御装置の一部の型式において、不揮発性ストレージによって可能になった高可用性機能。二重コピーは、論理 3990 サブシステム内に、指示された DASD ボリュームの機能的に同一の 2 つのコピーを保持する。二重コピー論理ボリュームに対して書き込み操作が発行されるたびに、それら 2 つのコピーが自動的に更新される。

[ハ行]

ハードウェア構成定義 (Hardware Configuration Definition (HCD)). インストール・システムが単一の制御ポイントからハードウェア構成を定義できるようにするための MVS/ESA SP の対話式インターフェース。

バインダー (binder). 言語翻訳プログラムおよびコンパイラーの出力を処理して実行可能プログラム (ロード・モジュールまたはプログラム・オブジェクト) にする DFSMS プログラム。z/OS のリンケージ・エディターおよびバッチ・ローダーに相当する。

バックアップ (backup). 事故による消失の場合に使用されるデータ・セットまたはオブジェクトのコピーを作成するプロセス。

パフォーマンス (performance). (1) ある製品が、任意の量のリソースで生産できる作業量の測定。(2) システム管理ストレージ環境において、ストレージ管理者によって設定された目標を基準とした、実効データ処理速度の測定。パフォーマンスは、スループット、応答時間、システム使用可能性によってその大部分が決定される。

光ディスク・ドライブ (optical disk drive). 光ディスク上で、データのシーク、読み取り、書き込みに使用されるメカニズム。光ディスク・ドライブは、オペレーターからアクセス可能であるか、ライブラリーに置くことができる。

光ディスク・ボリューム (optical volume). ボリューム・ラベルにより識別される、光ディスク上のストレージ・スペース。「ボリューム (volume)」も参照。

光ディスク・ライブラリー (optical library). 光ディスク・ドライブおよび光ディスク・カートリッジを内蔵し、カートリッジ・ストレージと光ディスク・ドライブの間で光ディスクを移動させるためのメカニズムを含む記憶装置。

非公式装置名 (esoteric unit name). TAPE または SYSDA など、ハードウェア特性の類似した装置グループを定義する際に使用される名前。「総称装置名 (generic unit name)」と対比。

プール・ストレージ・グループ (pool storage group). システム管理 DASD ボリュームを含むストレージ・グループのタイプ。プール・ストレージ・グループによって、ボリュームのグループを単一のエンティティーとして管理することができる。「ストレージ・グループ (storage group)」も参照。

フィルター操作 (filtering). 指定した基準に基づくデータ・セットの選択処理。この基準は、完全修飾データ・セット名または部分修飾データ・セット名、またはいくつかのデータ・セット特性で構成される。

不揮発性ストレージ (NVS) (nonvolatile storage (NVS)). バックアップ電源給電部を備えた、追加のランダム・アクセス電子記憶装置。IBM キャッシュ記憶制御装置と共に使用でき、電源異常時にデータを保存するために使用される。不揮発性ストレージは、すべての記憶ディレクターからアクセスでき、DASD 高速書き込み、および二重コピー操作中にデータを保管する。

プログラム・オブジェクト (program object). 仮想記憶域にロードして実行するのに適した形式のコンピューター・プログラムのすべてまたは一部。プログラム・オブジェクトは PDSE プログラム・ライブラリーに保管され、ロード・モジュールよりも制約が少ない。プログラム・オブジェクトはバインダーによって作成される。

並行コピー (concurrent copy). 通常のアプリケーション・プログラム処理と並行しながら、一貫性のあるデータのバックアップやコピーを作成できるようにし、データのアクセス可能性を向上させる機能。

保管場所 (storage location). 災害時回復、バックアップ、および重要レコード管理のためにボリュームを保管する、取り外し可能メディア・ライブラリーから物理的に離れた場所。

ボリューム (volume). DASD、テープ、または光ディスク装置上のストレージ。このストレージ・スペースは、ボリューム・ラベルで示される。「DASD ボリューム (DASD volume)」、「光ディスク・ボリューム (optical volume)」、および「テープ・ボリューム (tape volume)」も参照。

ボリューム状況 (volume status). ストレージ管理サブシステムにおいて、システム管理についてボリュームが完全に使用可能であるかどうかを示す。

- 「初期 (initial)」は、ボリュームにシステム管理に不適格なデータ・セットが含まれているために、そのボリュームはシステム管理用に準備ができていないことを示す。
- 「変換 (converted)」は、ボリューム上のすべてのデータ・セットが、関連するストレージ・クラスがあり、カタログされていることを示す。
- 「非システム管理 (non-system-managed)」は、ボリュームにシステム管理データ・セットが含まれておらず、システム管理として初期化されていないことを示す。

ボリューム取り付けアナライザー (volume mount analyzer). 現行の磁気テープ環境の分析を支援するプログラム。磁気テープ取り付け管理により、SMS 機能を使用して管理用の DASD バッファーにリダイレクトできるデータ・セットを識別できる。

[マ行]

マイグレーション (migration). 高可用性データ用のスペースを確保するために、使用されていないデータを費用が安い記憶装置に移動するプロセスのこと。そのデータ・セットを使用したい場合は、リコールを実行する必要がある。「マイグレーション・レベル 1 (migration level 1)」、「マイグレーション・レベル 2 (migration level 2)」も参照。

マイグレーション・レベル 1 (migration level 1). 1 次ストレージ・ボリュームからマイグレーションしたデータ・セットを含む DFSMSHsm 所有の DASD ボリューム。データは圧縮可能。「ストレージ階層 (storage

hierarchy)」も参照。「1 次ストレージ (primary storage)」、「マイグレーション・レベル 2 (migration level 2)」と対比。

マイグレーション・レベル 2 (migration level 2). 1 次ストレージ・ボリュームまたはマイグレーション・レベル 1 のボリュームからマイグレーションしたデータ・セットを含む DFSMSHsm 所有のテープまたは DASD ボリューム。データは圧縮可能。「ストレージ階層 (storage hierarchy)」も参照。「1 次ストレージ (primary storage)」、「マイグレーション・レベル 1 (migration level 1)」と対比。

マイルストーンによるインプリメンテーション (implementation by milestone). ご使用のシステムのデータを DASD、テープ、または光ディスク装置上のシステム管理ストレージにステージごとに変換する変換方法。

[ヤ行]

ユーザー・グループ (user group). 組織内で単一の部門または機能を表す、インストール・システム内のユーザーのグループ。

ユーザー・グループ責任者 (user group representative). あるユーザー・グループに属する人のうち、ストレージ管理グループと交渉する際に、ユーザー・グループの関心を代表することを担当する人。

有効期限 (expiration). (1) 有効期限または保存期間が経過した時点でどのデータ・セットまたはオブジェクトを削除するかを識別するプロセス。DASD では、データ・セットおよびオブジェクトが削除される。テープの場合、すべてのデータ・セットが有効期限を過ぎると、テープ・ボリュームが再使用可能になる。(2) DFSMSRmm において、重要レコード指定ポリシーによるか、ボリュームへのデータ・セットの書き込み時にはユーザー指定の JCL によるか、またはインストール時のデフォルトによって、有効期限または保存期間がすべてのボリュームに設定されている。ボリュームがその有効期限または保存期間に達すると、解放可能になる。

容量計画 (capacity planning). 予期されるワークロードに適応するために、必要な物理コンピューティング・リソースの適切な量を予測し、計算するプロセス。

[ラ行]

ラージ・フォーマット (large format). LARGE のデータ・セット名タイプ (DSNTYPE) のデータ・セットのフォーマット。ラージ・フォーマットのデータ・セットには、順次 (非拡張フォーマット) データ・セットと同じ

特性がありますが、各ボリューム上でのそのサイズは 65535 トラックを超えても構いません。ラージ・フォーマットのデータ・セットには、最小サイズ要件がありません。

リカバリー (recovery). データが損傷を受けたり破棄されたりした後のデータ再作成のプロセスで、しばしばデータのバックアップ・コピーの使用、またはログに記録されたトランザクションの再適用によって行われる。

リソース測定機能 (RMF) (Resource Measurement Facility (RMF)). システム活動のうちの選択した分野を測定したり、収集されたデータを印刷レポート、システム管理機能 (SMF) レコード、あるいは表示レポートのフォーマットで表示したりするための、IBM ライセンス・プログラムまたは z/OS のオプションなエレメント。RMF を使用すると、システム・パフォーマンスを評価したり、パフォーマンス上の問題の原因を識別したりできる。

リソース・アクセス管理機能 (RACF) (Resource Access Control Facility (RACF)). z/OS セキュリティー・サーバーに組み込まれている IBM ライセンス・プログラムで、z/OS および VM 環境用の個別プログラムとしても使用できる。RACF は、システムに対してユーザーを識別および検査したり、保護リソースへのアクセスを許可したり、無許可でシステムに入る試みを検出してログに記録したり、保護リソースへのアクセスを検出してログ記録したりすることにより、アクセス制御を提供する。

[数字]

1 次ストレージ (primary storage). ユーザーがデータ割り振りに使用できる DASD ボリューム。1 次ストレージ内のボリュームは、1 次ボリュームと呼ばれる。

「ストレージ階層 (storage hierarchy)」も参照。「マイグレーション・レベル 1 (migration level 1)」、「マイグレーション・レベル 2 (migration level 2)」と対比。

1 次スペース割り振り (primary space allocation). データ・セット作成時にユーザーによって要求されるスペースの量。「2 次スペース割り振り (secondary space allocation)」と対比。

2 次スペース割り振り (secondary space allocation). 1 次スペースがいっぱいになった場合にデータ・セットに追加するスペースとして、ユーザーによって要求された量。「1 次スペース割り振り (primary space allocation)」と対比。

D

DASD 高速書き込み (DASD fast write). IBM 3990 磁気ディスク制御装置の一部の型式の拡張機能。データは並行してキャッシュおよび不揮発性ストレージに書き込まれ、DASD へのデステージングが自動的にスケジューラされる。2つのコピーはデータが完全に DASD に書き込まれるまで磁気ディスク制御装置に保存され、DASD への直接書き込みと同等のデータ保全性を提供する。システム管理データ・セットの DASD 高速書き込み機能の使用は、パフォーマンスの向上のため、ストレージ・クラス属性により制御される。「動的キャッシュ管理 (dynamic cache management)」も参照。「キャッシュ高速書き込み (cache fast write)」と対比。

DASD ボリューム (DASD volume). 共通ラベルにより示され、一連の関連アドレスによりアクセスされる DASD スペース。「ボリューム (volume)」、「1 次ストレージ (primary storage)」、「マイグレーション・レベル 1 (migration level 1)」、「マイグレーション・レベル 2 (migration level 2)」も参照。

DFSMS. データ機能記憶管理サブシステム (Data Facility Storage Management Subsystem)。

DFSMS 環境 (DFSMS environment). ストレージ管理の自動化と集中化のための環境。この環境は、ハードウェア、ソフトウェア、ポリシーの組み合わせにより実現される。MVS の DFSMS 環境では、この機能は、DFSMS、DFSORT、および RACF によって提供される。「システム管理ストレージ (system-managed storage)」も参照。

DFSMSdfp. ストレージ管理、データ管理、プログラム管理、装置管理、および分散データ・アクセスに必要な機能を提供する、z/OS の DFSMS 機能コンポーネントまたは基本エレメント。

DFSMSdss. データ・セットとボリュームのコピー、移動、ダンプ、復元に使用される、z/OS の DFSMS 機能コンポーネントまたは基本エレメント。

DFSMSShsm. データのバックアップとリカバリー、およびストレージ階層でのボリュームのスペース管理に使用される、z/OS の DFSMS の基本的コンポーネントまたは基本エレメント。

DFSMSShsm 管理ボリューム (DFSMSShsm-managed volume). (1) DFSMSShsm に対して定義されていて、ストレージ・グループに属していない 1 次ストレージ・ボリューム。(2) ストレージ・グループの中で、DFSMSShsm 自動ダンプ、マイグレーション、またはバックアップ・サービスを使用しているボリューム。「システム管理ボリューム (system-managed

volume)」、「DFSMSrmm 管理ボリューム (DFSMSrmm-managed volume)」と対比。

DFSMSShsm 制御データ・セット (DFSMSShsm control data set). DFSMSShsm において、DFSMSShsm 処理で使用されるレコードを含む 3 つの VSAM キー順データ・セットのうちの一つ。「バックアップ制御データ・セット (backup control data set)」、「マイグレーション制御データ・セット (migration control data set)」、「オフライン制御データ・セット (offline control data set)」も参照。

DFSMSrmm. 取り外し可能メディアを管理するための、z/OS の DFSMS 機能コンポーネントまたは基本エレメント。

DFSMSrmm 管理ボリューム (DFSMSrmm-managed volume). DFSMSrmm に対して定義されたテープ・ボリューム。「システム管理ボリューム (system-managed volume)」、「DFSMSShsm 管理ボリューム (DFSMSShsm-managed volume)」と対比。

M

MVS/ESA. 多重仮想記憶/エンタープライズ・システム体系 (Multiple Virtual Storage/Enterprise Systems Architecture)。ESA/390 をサポートする z/OS オペレーティング・システム環境。

MVS/ESA SP. z/OS オペレーティング・システムを制御するために使用される IBM ライセンス・プログラム。MVS/ESA SP は DFSMS と共に基本 MVS/ESA 操作環境を構成する。「z/OS」も参照。

O

OAM 管理ボリューム (OAM-managed volumes). オブジェクト・アクセス方式 (OAM) により制御される光ディスクまたは磁気テープ・ボリューム。

S

SMS 構成 (SMS configuration). 構成のベース、ストレージ管理サブシステム・クラス、グループ、ライブラリー、およびドライブの定義、またストレージを管理するためにストレージ管理サブシステムが使用する ACS ルーチン。「構成 (configuration)」、「基本構成 (base configuration)」、「ソース制御データ・セット (source control data set)」も参照。

V

VSAM ボリューム・データ・セット (VVDS) (VSAM volume data set (VVDS)). 指定された DASD ボリューム上にあるシステム管理データ・セットおよび VSAM の特性を記述するデータ・セット。カタログの一部。基本カタログ構造 (*basic catalog structure*) も参照。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アクセシビリティ 391
圧縮処理の考慮事項 34
アンカATALOG, データ・セットの例 348
アンロード
移動 304
区分データ・セット 25, 367
コピー 304
再編成 294
順次データ・セットの作成 23
選択したメンバーの例 75
ディスク・ボリュームの例 328
メンバー除外の例 75
メンバーの除外
例 62
ラベルなしテープ・ボリュームの例 328
PDSE 25
アンロード・データ・セット, IEBCOPY
プログラム
規則と制約 369
コピー 26
属性レコード 372
注釈リスト・レコード 373
ディレクトリー・レコード 372
フォーマット
新しい 368
区分データ・セット 367
転送 368
古い (PDSE 以前の) 368
無効な 368
メンバー・データ・レコード 374
レコード・フォーマット 367
ロード 26
DCB パラメーター 26
移動
オプション
順次データ・セット 299
カタログ式データ・セットの例 329
基本
順次データ・セット 299
失敗
割り振られたスペース 297
順次データ・セット 299

移動 (続き)
データ・セット
アンロード 304
移動不能 304
カタログ式データ・セット 304
区分データ・セット 299
再ブロック 297
マルチボリューム 303
BDAM 302
COPYAUTH (許可のコピー) 298
IEHMOVE プログラム 312
RACF 298
データ・セットのボリュームの例 326, 328
ディスク・ボリュームから別々のボリュームの例 324
ボリューム 305
マージの例 325
BDAM マクロ 303
IEHMOVE プログラムの例 324, 328
移動不能データ・セット
移動 304
コピー 304
イメージ
プリンター, IEBIMAGE プログラム 147
ライブラリー, IEBIMAGE プログラム 149, 183
印刷
区分データ・セット
データ・セット 211
ディレクトリー 213, 281, 288
メンバーの選択 211
例 231, 232, 235, 291
例, メンバーの印刷 227
磁気テープ 213
順次データ・セット
データ・セット 211
メンバーの選択 211
例 138, 228, 229, 231, 232
選択レコード
例 232
IEBPTPCH プログラム 213
タイプ 21 SMF レコードの例 356
データ・セット 211
ディスク 213
表題 219
編集したデータ・セット 212
メンバー, データ・セットの 211
レコード・グループ例 229
16 進数出力例 232

印刷 (続き)
DBCS
例 233
IEBGENER プログラム 123
IEBIMAGE プログラム 147
IFHSTATR プログラム
タイプ 21 レコード 353
例 356
PDSE
ディレクトリー 213, 281, 288
メンバーの選択 211
例, ディレクトリー 291
インデックス, コピー
ディレクトリー項目処理 119
IEBGENER プログラム 119
オペレーション・グループ, IEBIMAGE
プログラム 168
[カ行]
ガイド, ユーティリティー・プログラム機能の 1
仮想記憶域要件
図形文字変更モジュール 148
ライブラリー文字セット 148
カタログ式データ・セット, 移動
例 329
可変長スパン・レコード
BDAM データ・セット
移動 302
コピー 302
キー
作成 119
出力レコード 132
キーボード 391
機能ステートメント
制約事項 244
IEBUPDTE プログラム
オペレーションを始める 239
構文 241
例 265
基本直接アクセス方式 302
行オーバーラン条件 174, 185, 186
区分出力 133
区分データ・セット
圧縮
処理上の考慮事項 33
アンロード 25
アンロード・フォーマット,
IEBCOPY 367
移動 299

区分データ・セット (続き)

印刷の例
 選択メンバー 211, 227
 データ・セット 211
 ディレクトリー 231
 DBCS 235
 コピー 299
 別名を持つメンバー 29
 メンバーの除外 29
 メンバーの選択 28
 例 20, 57
 IEBCOPY プログラム 23, 25
 コピーされるときのブロック・サイズ
 52
 順次入力からの作成
 ユーティリティー制御ステートメン
 ト 120
 例 139, 141
 スクラッチ 331
 穿孔 211
 選択したメンバーのコピー
 例 62
 選択したメンバーのコピーの例 58,
 60
 選択したメンバーの置換の例 60
 選択レコード、印刷された例 232
 ソース言語の変更 235
 データ・セットのライブラリーの更新
 235
 データ・セットのライブラリーの作成
 235
 ディレクトリー
 印刷 213, 288
 情報のコピー 26
 穿孔 213
 データ・セットの比較 11
 非編集 (ダンプ) フォーマット
 282
 リスト表示 281
 例 231
 ディレクトリーの印刷
 例 291
 ディレクトリーのリスト表示 288
 例 291
 名前変更したメンバーの例 350
 バックアップ 25
 比較
 例 20, 21, 23
 比較例 20
 複数のオペレーションのコピーの例
 66
 複数のコピー操作
 例 74
 への変換 24
 変換
 順次に変換 236

区分データ・セット (続き)

変換 (続き)
 PDSE に 79
 変更 235
 マージ
 順次データ・セット 140
 例 63
 IEBCOPY プログラム 23
 マスターの作成 235
 未使用域 (ガス) 33
 メンバー
 アンロード 28
 置換 30
 追加 105, 121
 名前変更 31, 332
 ロード 28
 メンバーのリスト表示 288
 ロード
 アンロード・データ・セット 26
 例 74
 区分データ・セットまたは PDSE をアン
 ロードおよび圧縮する、例 62
 決定、IEBCOPY 操作の
 アンロード操作
 区分入力データ・セット 48
 順次出力データ・セット 48
 区分データ・セット
 出力 48
 入力 48
 ロード操作
 区分出力データ・セット 48
 順次入力データ・セット 48
 COPY 操作 48
 合計ルーチン、戻りコード 384
 構文 23
 ページ・ヘッダー・パラメーター、
 HDNGLST 361
 ユーティリティー・プログラム、
 LINK マクロ 357
 HDNGLST パラメーター 361
 IEBCOMPR プログラム
 COMPARE ステートメント 14
 EXITS ステートメント 15
 LABELS ステートメント 15
 IEBCOPY プログラム
 ALTERMOD ステートメント 49
 COPY ステートメント 50
 COPYGRP ステートメント 51
 COPYMOD ステートメント 52
 EXCLUDE ステートメント 54
 SELECT ステートメント 55
 IEBEDIT プログラム、EDIT ステート
 メント 113
 IEBGENER プログラム
 EXITS ステートメント 131
 GENERATE ステートメント 130

構文 (続き)

IEBGENER プログラム (続き)
 LABELS ステートメント 132
 MEMBER ステートメント 133
 RECORD ステートメント 134
 IEBIMAGE プログラム
 CHARSET ステートメント 181,
 182
 COPYMOD ステートメント 174
 FCB ステートメント 169
 GRAPHIC ステートメント 178,
 179
 INCLUDE ステートメント 184
 NAME ステートメント 184
 OPTION ステートメント 185
 TABLE ステートメント 176
 IEBPTPCH プログラム
 EXITS ステートメント 221
 LABELS ステートメント 225
 MEMBER ステートメント 221
 PRINT ステートメント 216
 PUNCH ステートメント 216
 RECORD ステートメント 223
 TITLE ステートメント 220
 IEBUPDTE プログラム
 機能ステートメント 241
 詳細ステートメント 246
 ALIAS ステートメント 250
 ENDUP ステートメント 251
 LABEL ステートメント 249
 IEHINIT プログラム 272
 IEHLIST プログラム
 LISTPDS ステートメント 288
 LISTVTOC ステートメント 289
 IEHMOVE プログラム
 COPY DSGROUP ステートメント
 315
 COPY DSNAME ステートメント
 313
 COPY PDS ステートメント 317
 COPY VOLUME ステートメント
 319
 INCLUDE ステートメント 321
 MOVE DSGROUP ステートメント
 315
 MOVE DSNAME ステートメント
 313
 MOVE PDS ステートメント 317
 MOVE VOLUME ステートメント
 319
 REPLACE ステートメント 323
 SELECT ステートメント 322
 IEHPROGM プログラム
 ADD ステートメント 344
 CATLG ステートメント 343
 DELETEP ステートメント 346

構文 (続き)

IEHPROGM プログラム (続き)
EXEC ステートメント 338
LIST ステートメント 346
RENAME ステートメント 342
REPLACE ステートメント 344
SCRATCH ステートメント 340
UNCATLG ステートメント 343

コピー

アンロード 304
アンロードされたデータ・セット 26
移動不能 304
オプション
 順次データ・セット 299
拡張区分データ・セット
 IEBCOPY プログラム 83
拡張区分データ・セット、IEBCOPY
 プログラム 81
カタログ式データ・セット 304
基本
 順次データ・セット 299
区切り文字があるデータ・セットの例
 118
区分データ・セット 299
 例 57, 62
 IEBCOPY プログラム 23, 80
区分データ・セット、IEBCOPY プロ
 グラム 25
区分データ・セットの例 20
再ブロック 297
順次データ・セット 299
 例 117, 119, 141, 142, 145
順次データ・セットの例 19
ジョブ・ステートメント 115, 116,
 119
ジョブ・ステップの例 115, 116, 119
スペースの割り振りの失敗 297
選択したメンバーの例 60
データ・セット 298, 305
ディレクトリー情報 26
複数のオペレーションの例 66, 74
複数のジョブの例 115
変更モジュール
 オーバーラン注釈 186
 作成 156, 173
 データの印刷、指定 147
 例、作成 194, 196
 IEBIMAGE によるリスト表示 186
編集ステートメントの例 117
編集ステップの例 117
ボリューム 305
マルチボリューム 303
メンバー
 除外 29
 別名を持つ 29
ユーザー・ラベル 119

コピー (続き)

例 115, 119
ロード・モジュール 34
BDAM データ・セット 302
BDAM マクロ 303
COPYAUTH ステートメント 298
DBCS
 例 145
DBCS データ
 IEBGENER プログラム 123
IEBCOPY プログラム 23
 アンロード・データ・セット 26
 データ・セット 25
 別名を持つメンバー 29
 ロード・モジュール 34
 COPY ステートメント 47, 49
 COPYGRP ステートメント 47, 51
 COPYMOD ステートメント 47,
 52
IEBUPDTE プログラムの例 262
IEHMOVE プログラム 312
JOB ステートメント
 例 115
PDSE
 例 57
 IEBCOPY プログラム 25
RACF 298
SMS 管理ボリューム 293

[サ行]

再ブロック

データ・セット 297
例 141
ロード・モジュール 34
DBCS 123

索引順次アクセス方式 (Indexed sequential
access method) 209

索引付き VTOC

リスト表示 284
参照用の補助、特別 10

磁気テープ

入力、データ・セットの比較 19
ライブラリーの状態 354

磁気テープ上のデータ・セット、比較、7-
トラックのテープの例 17

磁気テープ・ラベル

作成 278
例 275, 276
IEHINIT プログラム
ボリューム 265

システム決定ブロック・サイズ 125

システム・ユーティリティー・プログラム
5

出力、IEHPROGM プログラム 337

出力レコード、IEBDG による生成 104,
107

順次データ・セット

移動 299

印刷

例 138, 231, 232

IEBTPCH プログラム 211

区分データ・セットの作成、例 139

コピー 299

例 19, 141, 142, 145

再ブロック 141

スクラッチ 331

穿孔

例 227, 228

IEBTPCH プログラム 211

ソース言語の変更 235

名前変更 332

比較

密度の例 18

例 17, 19, 20

IEBCOMPR プログラム 11

フィールド 103

変換 236

変更 235

編集例 142, 145

マージ、区分データ・セット 140

マスターの作成 235

ユーザー指定、例 229

例

コピー 19

比較 19

IEHMOVE プログラム 299

ショートカット・キー 391

詳細ステートメント

制約事項 248

IEBUPDTE プログラム 245, 248

省略語

キーワード 47

例外 47

ジョブ制御ステートメント

ユーティリティー・プログラムの制御
6

IEBCOMPR プログラム 13, 14

IEBCOPY プログラム 41

IEBEDIT プログラム 112, 113

IEBGENER プログラム 124, 129

IEBIMAGE プログラム 166, 167

IEBTPCH プログラム 214, 215

IEBUPDTE プログラム 236, 239

IEHINIT プログラム 269, 272

IEHLIST プログラム 286, 288

IEHMOVE プログラム 307, 311

IEHPROGM プログラム 337, 339

IFHSTATR プログラム 356

ジョブ・ステップ

コピーの編集の例 117

ジョブ・ステップ (続き)
 コピーの例 118
 出力データ・セットへのコピーの例 115
 選択の例 116
 複数のジョブのコピーの例 115
 処理ルーチン
 非ラベル・パラメーター 382
 ラベル・パラメーター 381
 身体障害 391
 診断
 CVOL 281
 推奨
 DFSORT と ICEGENER 119
 PDSU ブロック・サイズ 28
 SuperC ユーティリティ 11
 垂直行送り 151
 スクラッチ、データ・セットの 331
 例 347, 348
 IEHPROGM プログラム 331
 図形文字変更モジュール 147
 構造 162
 作成 161, 178, 196, 204
 複数のソース 201
 文字配列テーブルの変更 200
 リスト表示 196, 204
 例、印刷 199
 例、作成 200, 201
 ワールド・トレード GRAFMOD 199
 IEBIMAGE プログラムによるリスト表示 162
 World Trade National Use Graphics 199
 ストレージ管理サブシステム 331
 ストレージ記述子ブロック (SDB) xx, 125, 128
 スペース割り振り
 IEHMOVE プログラム 295
 制御ステートメント 6
 制約事項
 アンロード・データ・セット、IEBCOPY 369
 出力レコード、IEBDG による生成 107
 詳細ステートメント、IEBUPDTE プログラム 248
 ユーティリティ・プログラム 9
 COPYGRP ステートメント
 複数の INDD ステートメント 51
 EXCLUDE ステートメント 29
 FUNCTION ステートメント、IEBUPDTE プログラム 244
 IEBCOPY プログラム 39
 IEBGENER、ディレクトリー項目処理 119
 IEBIMAGE、UCS イメージ 148

制約事項 (続き)
 INSERT パラメーター、IEBUPDTE プログラム 247
 UPDATE パラメーター、IEBUPDTE プログラム 244
 穿孔
 区分データ・セット 211
 区分データ・セット・ディレクトリー 213
 磁気テープ 213
 順次データ・セット
 例 227
 IEBPTPCH プログラム 211
 選択レコード 213
 データ・セット
 区分 211
 順次 211
 PDSE 211
 ディスク 213
 編集したデータ・セット 212
 DBCS 212
 PDSE 211
 PDSE ディレクトリー 213
 属性レコード、IEBCOPY アンロード・データ・セット 372

[夕行]

置換
 選択したメンバーの例 60
 データ・セット・メンバー 30
 置換レベル
 データ・セット 30
 メンバー 31
 注釈リスト・レコード、IEBCOPY アンロード・データ・セット 373
 データ・ステートメント
 ユーザー・デザイン文字 180, 183
 IEBUPDTE プログラム 248
 データ・セット
 新しい区分、作成例 262
 新しいマスターの作成例 256
 アンカタログ 348
 アンカタログの例 349
 アンロード
 コピー、IEBCOPY プログラム 26
 ロード、IEBCOPY プログラム 26
 DCB パラメーター、IEBCOPY プログラム 26
 維持、パスワードの 333
 移動
 アンロード 303, 304
 移動不能属性 304
 カタログ化されたグループ 304
 区分 298
 順次 299

データ・セット (続き)
 移動 (続き)
 説明 293
 ボリューム 305
 マルチボリューム 303
 BDAM 298, 302
 IEHMOVE プログラム 298, 312
 印刷 211
 コピー
 アンロード 303, 304
 移動不能属性 304
 カタログ化されたグループ 304
 区分 298
 順次 299
 説明 293
 ボリューム 305
 マルチボリューム 303
 BDAM 298, 302
 IEBCOPY プログラム 23
 IEBGENER プログラム 119
 IEHMOVE プログラム 298, 312
 再ブロック 297
 順次の例、カード入力 261
 スクラッチ 348
 スクラッチの例 347
 スペース割り振り
 IEHMOVE プログラム 295
 穿孔 211
 名前変更の例 348
 入出力 111
 比較 11
 編集 121
 編成の変更 236
 マージ 25
 メンバー
 置換 30
 名前変更 31
 別名 331
 ユーティリティ・プログラムの要約 5
 ライブラリー・メンバーの例、レコードの追加 259
 例
 アンカタログ 348
 スクラッチ 348
 論理レコード長 123
 IEBUPDTE プログラム・ライブラリー 235
 SYS1.IMAGELIB 147
 データ・セットのマージ、ディレクトリー・ブロックの割り振り 25
 ディレクトリー・ブロックの割り振り
 データ・セットのマージ 25
 ディレクトリー・リスト
 編集フォーマット 281

ディレクトリー・レコード、IEBCOPY ア
ンロード・データ・セット 372
 出力ルーチン
 合計 384
 識別 131
 非ラベル処理ルーチン 382
 戻りコード 379
 ユーティリティ・プログラム 377
 IEBCOMPR プログラム 14
 IEBDG プログラム 102
 IEBPTPCH プログラム 220
 IEBUPDTE プログラム 237, 242, 243
 RETURN マクロ 378
 テスト・データのパターン
 タイプの指定 92
 ユーザー指定
 フォーマット 84
 例 109
 IBM 提供 83
 デバッグ・ツール、IEBDG ユーティリ
 ティー 83
 統合カタログ機能 349
 同所圧縮操作
 処理上の考慮事項 33, 34
 データ・セット 33

[ナ行]

名前変更
 区分データ・セットの例 350
 データ・セット 331
 マルチボリューム・データ・セットの
 例 348
 入力、IEHPROGM プログラム 337
 入力ストリーム 19

[ハ行]

排他的なコピー、IEBCOPY プログラム
 31
 パスワード
 維持 333
 削除 336
 置換
 データ・セットに 336
 例 350
 追加 335
 リスト表示、項目の 336
 例
 定義 349
 リスト表示 350
 バックアップ
 データ・セット 119
 メンバー 119
 例 62

バックアップ (続き)
 IEBCOMPR プログラムによる検査
 11
 IEBCOPY プログラム 25
 バッファ
 作業域 36
 IEHMOVE プログラム 309
 パラメーター
 非ラベル処理 382
 ラベル処理 381
 パラメーター・リスト、作成 358
 比較、IEBCOMPR プログラム
 区分データ・セットのコピーの例 20
 区分データ・セットの例 20
 異なる密度の順次データ・セットの例
 18
 磁気テープ上のデータ・セットの例
 17
 順次データ・セットのコピーの例 19
 順次データ・セットの例 19
 データ・セット
 カード入力 19
 区分 11, 20, 21
 磁気テープ入力 19
 順次 11, 17, 20
 PDSE 11, 21, 23
 PDSE の例 21
 表記規則、ユーティリティ・プログラム
 xvi
 標準ラベル
 磁気テープ・ボリューム 267
 非ラベル処理ルーチン 382
 フォーマット、IFHSTATR プログラム、タ
 イプ 21 レコード 353
 複数のコピー操作の例 66, 74
 プリンター
 3800 FCB モジュール構造、モデル
 I 151
 4248 FCB モジュールの構造 152
 FCB 149
 プリンター・チャンネル・コード
 チャンネル 1、9、12 の規則 171
 FCB ステートメントでの指定 170
 FCB モジュール 151
 フル・コピー、IEBCOPY プログラム 31
 プログラム・オブジェクト
 別名 32
 メンバー 32
 ロード・モジュールへの変換 24
 ブロック・サイズ
 アンロード・データ・セット
 MAXBLK 27
 MAXBLK、MINBLK 46
 MINBLK 27
 COPYMOD ステートメント 52

ページの余白。3800 および 4248 印刷装
 置の場合に指定します。 171
 ページ・ヘッダー・パラメーター、
 HDNGLSTの構文 361
 別名
 作成、区分メンバーに対して 250
 変更、メンバー 331
 メンバーのコピー 29
 変換
 アンパック 10 進からバック 10 進へ
 137
 区分データ・セットを PDSE に 24,
 79
 順次データ・セットから区分データ・
 セットまたは PDSE へ 120
 パック 10 進からアンパック 10 進へ
 137
 プログラム・オブジェクトをロード・
 モジュールに 24
 ロード・モジュールをプログラム・オ
 ブジェクトに 24
 H セット BCDIC から EBCDIC へ
 136
 変換テーブル、モジュールの構造 158
 編集
 順次データ・セットの例 142, 145
 DBCS 123
 DBCS の例 145
 編集したデータ・セット
 印刷 212
 作成 121
 穿孔 212
 保護
 維持、パスワードの 333
 データ・セット
 削除、パスワードの 336
 置換、パスワードの 336
 追加、パスワードの 335, 349
 例 349
 パスワード、データ・セット項目のリ
 スト表示 336
 パスワードの置換の例 350
 パスワードのリスト表示の例 350
 RACF 298
 ボリューム
 目録 283
 ボリューム・サイズの互換性
 IEHMOVE プログラム 294
 ボリューム・データ・セットの移動 305
 ボリューム・データ・セットのコピー
 305
 ボリューム・ラベルの初期化 272
 ボリューム・ラベル・セット 265

[マ行]

マージ
区分データ・セット、IEBCOPY プログラム 23
区分データ・セットの例 58, 63
コピーの例 75
順次データ・セット、区分データ・セット 140
ロードの例 75
COPY ステートメント 317
MOVE ステートメント 317

マクロ
BDAM 303
CLOSE マクロ 383
LINK 270, 357
OPEN マクロ 383
RETURN 378
SETPRT 150, 167
マクロ・ライブラリー 235
マルチタスキング環境 339
マルチボリューム・データ・セット
移動 303
コピー 303
密度、順次データ・セットの比較例 18
命名
新しいイメージ・ライブラリー・モジュール 184
IEBIMAGE によって作成されたモジュール 150
メッセージ検索ツール、LookAt xv
メンバーの選択
アンロードのために 28
コピーする 28
ロードのために 28
メンバー・データ・レコード、IEBCOPY
アンロード・データ・セット 374
文字配列テーブル
構造 158
構築例 196, 198
作成 157, 176
図形参照の削除例 198
変更例 196, 198
リスト表示 160
3800 モジュール 147
モジュール
同じ場所での変更 34, 77
構造 150
コピー 34
再ブロック 34
別名 150
IEBIMAGE のモジュールの命名規則 150
戻りコード
合計ルーチン 384
ユーザー出口ルーチン 379

戻りコード (続き)
ユーティリティ・プログラム 362, 363, 366
IEBDG ユーザー出口ルーチン 385
IEBIMAGE プログラム 363
IEBPTPCH プログラム 364
IEBUPDTE プログラム 364
IEHINIT プログラム 365
IEHLIST プログラム 365
IEHMOVE プログラム 365
IEHPROGM プログラム 366

[ヤ行]

ユーザー ABEND コード 362
ユーザー指定
テスト・データのパターン 84, 92, 101, 109
例 109
ユーザー出口ルーチン、ユーティリティ・プログラムへの戻り 378
ユーザー・ラベル
システムの処置 383
処理
データ 385
データ・セット記述子 383
データとして処理 225
物理レコードの変更 133
IEBGENER での処理 132
ユーティリティ制御ステートメント
継続 9
コーディング 8, 10
説明 8
フィールド 8
フォーマット 8
ADD 239, 245, 344
CATLG 343
CHANGE 239, 245
COPY DSGROUP 315, 317
COPY DSNNAME 312, 315
COPY PDS ステートメント 317, 319
COPY VOLUME 319, 321
CREATE 96, 102
DELETE 245, 249
DELETEP 346
DSD 89
EDIT 113, 114
END 102
ENDUP 251
EXCLUDE 322
EXITS 14, 131, 220
FD 90, 96
GENERATE 130
IEBCOMPR プログラム 14, 16
IEBCOPY プログラム 47
IEBIMAGE プログラム 167

ユーティリティ制御ステートメント (続き)
IEBPTPCH プログラム 215, 226
IEBUPDTE プログラム 251
IEHINIT プログラム 269
IEHLIST プログラム 288, 291
IEHPROGM プログラム 339, 347
INCLUDE 321
INIT 272, 275
LABEL 249
LABELS 132, 225
LIST 346
LISTVTOC 289
MEMBER 133, 221
MOVE DSGROUP 315, 317
MOVE DSNNAME 312, 315
MOVE PDS ステートメント 317, 319
MOVE VOLUME 319, 321
NUMBER 245, 248, 249
PRINT 216, 219
PUNCH 216, 219
RECORD 134, 138, 222, 225
RENAME 342, 343
REPEAT 96
REPL 239, 245
REPLACE 323, 344
REPRO 239, 245
SCRATCH 340, 342
SELECT 322
TITLE 219
UNCATLG 343
ユーティリティ・プログラム
アプリケーション・プログラムからの起動 357
機能 1, 11
システム 1
制約事項 9
説明 1
データ・セット 5
データ・セットの共用 7
出口ルーチン
概説 377
合計 384
プログラミングに関する注意点 378
戻り 378
戻りコード 379, 384
レジスター内容 378
表記規則 xvi
RETURN マクロ
出口ルーチン 378

[ラ行]

ライブラリー
 アップデートの例 255
 維持 149
 磁気テープの品質 354
 プリンター・データ 147
 文字セット・モジュール
 構造 164
 作成 163, 180, 204, 209
 説明 147
 リスト表示 204, 209
 IEBIMAGE によるリスト表示 165
 IEBUPDTE プログラム
 区分メンバー 235
 例、区分メンバー 253
 例、作成 254
ライブラリー・メンバー
 データ・セット作成例 256
 データ・セットの更新例 256
ラベル処理 382
リスト表示
 区分データ・セット・ディレクトリー
 281
 項目 288
 例 291
 パスワード項目 336
 非編集フォーマット 282
 編集されたロード・モジュール 281
 変数 xvii
 ライブラリー文字セット・モジュール
 の例 204
 IEBIMAGE プログラム、COPYMOD
 モジュール 157
 PDSE ディレクトリー 281, 288
 例 291
 VTOC
 項目 289
 索引付き 284
 非編集 (ダンプ) フォーマット
 285
 編集フォーマット 283
 例 292
 IEHLIST プログラム 387
ルーチンへの入り口、ユーティリティー・
 プログラム 378
レコード
 グループ
 順次データ・セットの分割 120
 定義 134, 222
 フィールド 223
 フィールド
 内容のリプル 103
 IEBDG を使用した内容の定義 83,
 90, 96

レコード (続き)
 フィールド (続き)
 IEBDG を使用して内容を変更する
 85, 93
レコード・グループ、印刷例 229
レコード・フォーマット、IEBCOPY アン
 ロード・データ・セット 367
ロード
 区分データ・セットの再作成 23
 順次データ・セットの区分の例 74
 順次データ・セットの例 328
 マージの例 75
 ラベル付き 7 トラック・テープの例
 328
ロード処理、IEBCOPY プログラム 31
ロード・モジュール
 同じ場所での変更 34, 77
 コピー 34
 再ブロック
 例 77, 79
 IEBCOPY プログラム 34
 置換の例 77
 プログラム・オブジェクトへの変換
 24
 ブロック・サイズ 52
 リスト表示、編集された 281
 ALTERMOD ステートメント 49
ロード・ライブラリー
 再ブロックの例 78
 配布の例 78
論理レコード
 長さの変更 123
 IEBCOPY プログラム、長さ 27
 IEBUPDTE プログラム、データ・ステ
 ートメント 248

[ワ行]

割り振り / 割り振り解除 339

[数字]

16 進数出力印刷例 232
3800 印刷サブシステム モデル 1
 モジュール・タイプ 147
 FCB モジュールの構造 151
 GRAPHIC モジュールの構造 162
3800 印刷サブシステム・モデル 3
 図形文字セット・モジュールの作成
 185
 ライブラリー文字セット・モジュール
 の作成 185
 GRAPHIC モジュールの構造 162
4248 FCB モジュールの構造 152

4248 印刷装置、FCB モジュールの作成
 185
4248 印刷装置モジュール 147

A

ABEND コード、IEBCOPY プログラム
 362
ACCESS パラメーター
 INITT ステートメント 274
ADD ステートメント
 IEBUPDTE プログラム 239, 245
 IEHPROGM プログラム 344
ALIAS ステートメント、IEBUPDTE プロ
 グラム 250
ALTERMOD ステートメント
 制約事項 34
 IEBCOPY プログラム 34, 47
anyname (任意名) DD ステートメント
 IEHINITT プログラム 270, 271
 IEHLIST プログラム 286
 IEHMOVE プログラム 310
 IEHPROGM プログラム 338, 339
anyname1 DD ステートメント、IEBCOPY
 プログラム 41, 45
anyname2 DD ステートメント、IEBCOPY
 プログラム 41
APF (許可プログラム機能)、ユーティリテ
 ィ・プログラムの起動 357

B

BDAM
 マクロ 303
BDAM (基本直接アクセス方式)
 移動 302
 コピー 302
 スクラッチ 331
 名前変更 331

C

CATLG ステートメント、IEHPROGM プ
 ログラム 343
CHANGE ステートメント、IEBUPDTE プ
 ログラム 239, 245
CHARSET ステートメント
 ステートメント 180, 183
 モジュール 165
 IEBIMAGE プログラムの構文規則
 181, 182
CLOSE マクロ
 合計ルーチン 383
 ユーザー・ラベル 383

COMPARE ステートメント
 構文 14
 IEBCOMPR プログラム 14
 COMPRESS パラメーター、IEBCOPY プログラム 42
 COPY DSGROUP ステートメント、IEHMOVE プログラム 315, 317
 COPY DSNAME ステートメント、IEHMOVE プログラム 312, 313, 315
 COPY PDS ステートメント、IEHMOVE プログラム 312, 317, 319
 COPY VOLUME ステートメント
 IEHMOVE プログラム 321
 COPY VOLUME ステートメント、IEHMOVE プログラム 319
 COPY ステートメント、IEBCOPY プログラム 34, 47, 49
 COPYAUTH ステートメント
 移動およびコピー 298
 COPYGRP ステートメント
 構文 51
 IEBCOPY プログラム 47, 51
 COPYGRP メンバー、置換 32
 COPYMOD ステートメント
 オーバーラン注釈付きの IEBIMAGE リスト表示 175
 構文 52, 174
 説明 35
 モジュール 156
 例 194
 IEBCOPY プログラム 47, 52, 53
 IEBIMAGE プログラム 173, 176
 CREATE ステートメント 96
 CREATE パラメーター、ユーザー出口ルーチン 377

D

DATA パラメーター、ユーザー出口ルーチン 377
 DATATYPE パラメーター、SYSIN 8
 DATE フィールド、フォーマット済み VTOC リスト表示の 388
 DBCS (2 バイト文字セット)
 印刷 123, 212
 印刷の例 233
 コピー 123
 再ブロック 123
 シフトイン / シフトアウト
 削除 136
 挿入 136
 文字 123
 穿孔 212
 妥当性検査 123
 編集 123
 例、編集 145

DCB パラメーター
 IEBCOPY プログラム、データ・セットのアンロード 27, 46
 OPTCD=W 45
 DD JCL ステートメント
 IEBCOMPR プログラム 13
 IEBCOPY プログラム 44, 47
 IEBDG プログラム 88
 IEBEDIT プログラム 112
 IEBGENER プログラム 127
 IEBIMAGE プログラム 167
 IEBPTPCH プログラム 214
 IEBUPDTE プログラム 238, 239
 IEHINIT プログラム 270, 272
 IEHLIST プログラム 286, 288
 IEHMOVE プログラム 309, 311
 IEHPROGM プログラム 338, 339
 IFHSTATR プログラム 356
 DDM 属性、コピー 25
 ddname ステートメント
 INITT ステートメント 273
 DELETEP ステートメント、IEHPROGM プログラム
 構文 346
 パスワードの削除 346
 DETAIL ステートメント、IEBUPDTE プログラム 245, 248
 device 変数 xvii
 DFSMSdss (データ機能データ・セット・サービス) 307
 DFSMSrmm 269
 DFSORT (データ機能分類プログラム) 119
 DISP パラメーター
 INITT ステートメント 273
 DSD ステートメント、IEBDG プログラム 89

E

EDGINERS プログラム 269
 EDIT ステートメントの構文、IEBEDIT プログラム 113
 END ステートメント、IEBDG プログラム 102
 ENDUP ステートメント、IEBUPDTE プログラム 251
 EOVS (ボリュームの終わり)
 ユーザー・ラベル 383
 CLOSE マクロ 383
 ERROR パラメーター、ユーザー出口ルーチン 377
 ESV (ボリューム別エラー統計) データ、IFHSTATR プログラム
 印刷 353
 印刷出力 355

ESV (ボリューム別エラー統計) データ、IFHSTATR プログラム (続き)
 ソート 354
 例、印刷 356
 EXCLUDE ステートメント
 COPY ステートメント
 DSGROUP 312
 PDS 312
 IEBCOPY プログラム 47, 53, 56
 IEHMOVE プログラム 322
 MOVE ステートメント
 DSGROUP 312
 PDS 312
 EXEC JCL ステートメント
 IEBCOPY プログラム 41
 IEBEDIT プログラム 112
 EXEC ステートメント
 IEBGENER プログラム 124, 125, 127
 IEBIMAGE プログラム 167
 IEBPTPCH プログラム 214
 IEBUPDTE プログラム 237
 IEHINIT プログラム 270
 IEHLIST プログラム
 構文 287
 IEHMOVE プログラム 308
 構文 308
 IEHPROGM プログラム 338
 IFHSTATR プログラム 356
 EXITS ステートメント
 IEBCOMPR プログラムの構文規則 15
 IEBGENER プログラムの構文規則 131
 IEBPTPCH プログラム
 構文 221
 出口ルーチン 220

F

FCB (用紙制御バッファ)
 図形文字変更モジュール
 例、印刷 199, 202
 例、使用 202
 例、定義 202
 ステートメント、IEBIMAGE プログラム 169, 173
 プリンター 149
 文字配列テーブル・モジュール
 例、作成 196, 198
 例、図形参照の削除 198
 例、変更 196, 198
 モジュール
 構造 151, 152
 作成 169
 プリンター情報 151
 例、作成 189, 196

FCB (用紙制御バッファ) (続き)
 モジュール (続き)
 例、置換 189, 191
 3800 FCB モジュール 151, 169
 4248 FCB モジュール 152, 169
 IEBIMAGE によるリスト表示 154
 IEBIMAGE プログラム 169
 ライブラリー文字セット・モジュール
 例、作成 205
 例、複数のソースからの作成 207
 例、変更 205
 例、リスト表示 204
 3800 印刷装置モジュール 147
 4248 印刷装置モジュール 147
 IEBIMAGE プログラムの構文規則
 169
 FD ステートメント、IEBDG プログラム
 90, 96

G

GENERATE ステートメント、IEBGENER
 プログラム 130
 GRAPHIC ステートメントの構文、
 IEBIMAGE プログラム 178, 179, 180
 GRAPHIC モジュール、IEBIMAGE プロ
 グラム 162

H

HDNGLST ページ・ヘッダー・パラメー
 ターの構文 361

I

IBM TotalStorage Enterprise 自動化テー
 プ・ライブラリー (3495) 269
 ICEGENER プログラム 119
 ICF (統合カタログ機能) 349
 IEBCOMPR プログラム
 出力 12
 順次データ・セットの例 17
 ジョブ制御ステートメント 13, 14
 説明 11
 データ・セットの比較 11
 入力 12
 バックアップ・コピーの検査 11
 戻りコード 362
 ユーザー出口ルーチン 377
 ユーティリティー制御ステートメント
 14, 16
 例 16, 23
 COMPARE ステートメント 14
 EXEC ステートメント 13
 EXITS ステートメント 14

IEBCOMPR プログラム (続き)
 LABELS ステートメント 15
 SYSIN DD ステートメント 13
 SYSPRINT DD ステートメント 13
 SYSUT1 DD ステートメント 13
 SYSUT2 DD ステートメント 13
 IEBCOPY プログラム
 アプリケーション・プログラムからの
 起動
 例 359
 アプリケーション・プログラムからの
 起動の例 360
 アンロード・データ・セット
 規則と制約 369
 属性レコード 372
 注釈リスト・レコード 373
 ディレクトリー・レコード 372
 フォーマット 367
 メンバー・データ・レコード 374
 DCB パラメーター 27, 46
 アンロード・データ・セットのロード
 26
 仮想記憶域 36
 コピー
 アンロード・データ・セット 26
 データ・セット 25
 ディレクトリー情報 26
 プログラム・オブジェクト、
 COPYGRP 32
 別名を持つメンバー 29
 ロード・モジュール 34
 DDM 属性 25
 コピーするメンバーの選択 28
 作業域サイズ 36
 出力 38
 ジョブ制御ステートメント 41
 制約事項 39
 説明 23
 選択的なコピー 31
 データ・セット
 アンロード 25
 コピー 25
 バックアップ 25
 マージ 25
 データ・セットの圧縮、処理上の考慮
 事項 33
 データ・セットのアンロード 25
 データ・セットのマージ 25
 データ・セット・メンバーの置換 30
 テーブル・サイズ 36
 ディレクトリー情報
 コピー 26
 長い名前 32
 名前変更、選択したメンバーの 31
 入力 38
 バッファ・サイズ 36

IEBCOPY プログラム (続き)
 変換
 区分データ・セットを PDSE に
 24
 プログラム・オブジェクトをロー
 ド・モジュールに 24
 ロード・モジュールをプログラム・
 オブジェクトに 24
 メンバーの除外 29
 戻りコード 362
 ユーザー ABEND コード 362
 ユーティリティー制御ステートメント
 47
 例 56
 ロード処理 31
 ロード操作 23
 ロードまたはアンロードするメンバ
 ーの選択 28
 ロード・モジュールの同じ場所での変
 更 34
 ロード・モジュールの再ブロック 34
 論理レコード長 27
 ABEND コード 362
 ALTERMOD ステートメント 47, 49
 anyname1 DD ステートメント 45
 anyname2 DD ステートメント 45
 COPY ステートメント 47, 49
 COPYGRP ステートメント 47, 51
 COPYMOD ステートメント 47, 52
 COPYMOD の例 77
 EXCLUDE ステートメント 28, 47,
 53
 EXEC JCL ステートメント 41
 INDD ステートメント 47
 JOB ステートメント 41
 MEMBER パラメーター 31, 32
 RLD カウントの挿入 35
 SELECT ステートメント
 説明 54
 名前変更、メンバーの 31
 別名の置換 32
 メンバーの選択 28, 47
 SYSIN DD ステートメント 47
 SYSPRINT DD ステートメント 44
 SYSUT1 DD ステートメント 45
 SYSUT2 DD ステートメント 45
 SYSUT3 ステートメント 46
 SYSUT4 ステートメント 46
 IEBDG プログラム
 アクション 85, 93
 出力 86
 ジョブ制御ステートメント 87, 89
 説明 83
 出口 385
 入力 86
 戻りコード 362

IEBDG プログラム (続き)

ユーザー指定のパターン 84
ユーザー出口ルーチン 377
ユーティリティ制御ステートメント
89, 102
例 102, 111
レコード・フィールドの定義 83
レコード・フィールドの変更 85
anyname1 DD ステートメント 88
anyname2 DD ステートメント 88
CREATE ステートメント 385
DSD ステートメント 89
END ステートメント 102
EXEC ステートメント 87
FD ステートメント 90
IBM 提供のパターン 83
SYSPRINT DD ステートメント 88

IEBEDIT プログラム

出力 111
ジョブ制御ステートメント 112, 113
説明 111
入力 111
戻りコード 363
ユーティリティ制御ステートメント
113, 114
例 114, 119
EXEC JCL ステートメント 112
JOB ステートメント 112
SYSIN DD ステートメント 112
SYSPRINT DD ステートメント 112
SYSUT1 DD ステートメント 112
SYSUT2 DD ステートメント 112

IEBGENER プログラム

アプリケーション・プログラムからの
起動の例 359
区分データ・セット
メンバーの追加 121
例 139, 141
コピー、ディレクトリー項目処理なし
119
再ブロックの例 141
作成
区分データ・セットまたは
PDSE 120
編集したデータ・セット 121
シフトイン / シフトアウトの削除
136
シフトイン / シフトアウトの挿入
136
出力 123
順次データ・セットの例 141, 145
ジョブ制御ステートメント 124, 129
説明 119
入力 123
バッファ 127

IEBGENER プログラム (続き)

変換
アンパック 10 進からパック 10 進
へ 137
パック 10 進からアンパック 10 進
へ 137
H セット BCDIC から EBCDIC へ
136
戻りコード 363
ユーザー出口ルーチン 377
ユーティリティ制御ステートメント
129, 138
メンバー 133
EXITS 131
GENERATE ステートメント 130
LABELS ステートメント 132
RECORD ステートメント 134
領域サイズ計算 127
例 138, 145
論理レコード長の変更 123
DBCS
データ 123
例 145
DD JCL ステートメント 127
EXEC ステートメント 124
JOB ステートメント 124
PDSE 121
SYSIN DD ステートメント 124
SYSPRINT DD ステートメント 124,
127
SYSUT1 DD ステートメント 124
SYSUT2 DD ステートメント 124
IEBIMAGE プログラム
作成
コピー変更モジュール 156
図形文字変更モジュール 161
文字配列テーブル・モジュール
157
ライブラリー文字セット・モジュー
ル 163
FCB モジュール 151
サポートされているプリンター・モデ
ル 147
出力 166
ジョブ制御ステートメント 166, 167
説明 147
操作グループ 168
入力 166
文字配列テーブル・モジュール 157
モジュール
構造 150
命名規則 150
戻りコード 363
ユーティリティ制御ステートメント
167, 187
例 187, 209

IEBIMAGE プログラム (続き)

3800 FCB モジュール 151
4248 FCB モジュールの構造 152
CHARSET ステートメント 180, 183
CHARSET モジュールの構造
説明 164
3800 モデル 1, モデル 3 165
CHARSET モジュールのリスト表示
165
COPYMOD ステートメント 173, 176
COPYMOD モジュール
構造とリスト表示 157
COPYMOD モジュール構造とリスト表
示 156
EXEC ステートメント 167
FCB ステートメント 169, 173
FCB モジュールのリスト表示 154
GRAPHIC ステートメント 178, 180
GRAPHIC モジュール
構造 162
リスト表示 162
INCLUDE ステートメント 183
JOB ステートメント 167
NAME ステートメント 184
OPTION ステートメント 185
SYS1.IMAGELIB データ・セット 149
SYSIN DD ステートメント 167
SYSPRINT DD ステートメント 167
SYSUT1 DD ステートメント 167
TABLE ステートメント 176, 178
TABLE モジュール
構造 158
リスト表示 160
IEBISAM プログラム
説明 209
IEBPTPCH プログラム
印刷
区分データ・セット 235
区分ディレクトリー 213
磁気テープ 213
選択レコード 213
データ・セット 211
ディスク 213
メンバーの選択 211
DBCS 212
出力 213
ジョブ制御ステートメント 214, 215
説明 211
穿孔
区分データ・セット 235
区分ディレクトリー 213
磁気テープ 213
選択レコード 213
データ・セット 211
ディスク 213
メンバーの選択 211

IEBPTPCH プログラム (続き)

穿孔 (続き)

DBCS 212

入力 213

編集したデータ・セット 212

戻りコード 364

ユーザー出口ルーチン 377

ユーティリティ制御ステートメント
の使用 215

EXITS 220

LABELS 225, 226

MEMBER 221

RECORD 222

TITLE 219

例 226, 235

EXEC ステートメント 214

SYSIN DD ステートメント 214

SYSPRINT DD ステートメント 214

SYSUT DD ステートメント 214

SYSUT2 DD ステートメント 214

IEBUPDTE プログラム

機能ステートメント

構文 241

制約事項 244

説明 239

削除、レコードの

例 256

作成、カード入力による

例 261

作成、データ・セットの

例 256

例、区分データ・セット 256

出力 236

順次データ・セットの例、コピー 262

詳細ステートメント 245, 248

構文 246

制約事項 248

ジョブ制御ステートメント 236, 239

説明 235

データ・ステートメント 248

データ・セットの更新例、区分データ・
セット 256

入力 236

変更、既存データ・セットの 235

編成 236

マスターの作成 235

戻りコード 364

ユーティリティ制御ステートメント
の使用 239

ENDUP 251

LABEL 249

ライブラリー

区分 253

更新 235

作成 235

挿入 258

IEBUPDTE プログラム (続き)

ライブラリー (続き)

追加、レコードの 259

番号変更 259

例 252, 253, 258, 259, 265

SYS1.PROCLIB 252

例 251, 265

例、新しい区分 262

論理レコード 248

ALIAS ステートメントの構文 250

ENDUP ステートメント

構文 251

EXEC ステートメント 237

JOB ステートメント 236

LABEL ステートメントの構文 249

REPLACEステートメントの例 255

SYSPRINT DD ステートメント 238

SYSUT1 DD ステートメント 238

SYSUT2 DD ステートメント 238

IEHINITT プログラム

構文 270

出力 269

ジョブ制御ステートメント 269, 272

説明 265

入力 269

標準ラベル、磁気テープの 267

戻りコード 365

ユーティリティ制御ステートメント
272, 275

例 275, 278

anyname (任意名) DD ステートメント
270

EXEC ステートメント 270

JOB ステートメント 269

PARM パラメーター、EXEC ステート
メント 270

SYSIN DD ステートメント 270

SYSPRINT DD ステートメント 270

IEHLIST プログラム

アプリケーション・プログラムからの
起動の例 361

区分データ・セット 288

出力 286

ジョブ制御ステートメント 286, 288

説明 281

ディレクトリー

区分データ・セット 281

非編集 (ダンプ) フォーマット
282

PDSE 281

入力 286

戻りコード 365

ユーティリティ制御ステートメント
288, 291

LISTPDS ステートメント 288

LISTVTOC ステートメント 289

IEHLIST プログラム (続き)

リスト表示

区分データ・セット・ディレクトリ
ー 281, 282

索引付き VTOC 284

非編集 (ダンプ) フォーマット
282, 285

フォーマット済み VTOC 388

編集フォーマット 281, 283

PDSE ディレクトリー 281, 282

VTOC 283, 285

例 291, 293

EXEC ステートメント 286, 287

JOB ステートメント 286

SYSIN DD ステートメント 286

SYSPRINT DD ステートメント 286

VTOC リスト表示のサンプル 387

IEHMOVE プログラム

移動

アンロードされたデータ・セット
304

移動不能データ・セット 304

カタログ式データ・セットのグルー
プ 304

区分データ・セット 299

順次データ・セット 299

データ・セット 298

ボリューム 305

マルチボリューム・データ・セット
303

BDAM 302

区分データ・セット

例 326

区分データ・セットの例 325, 326

コピー

移動不能データ・セット 304

カタログ式データ・セットのグルー
プ 304

区分データ・セット 299

順次データ・セット 299

データ・セット 298

ボリューム 305

マルチボリューム・データ・セット
303

BDAM 302

出力 306

順次データ・セットの例 324, 328

ジョブ制御ステートメント 307, 311

スペース割り振り 295

説明 293, 311

データ・セットの再ブロック 297

入力 306

バッファ 309

ボリューム・サイズの互換性 294

戻りコード 365

- IEHMOVE プログラム (続き)
ユーティリティ制御ステートメントの使用 312
COPY VOLUME 319
EXCLUDE 322
INCLUDE 321
MOVE VOLUME 319
REPLACE 323
SELECT 322
ユーティリティ制御プログラム
COPY DSGROUP 315, 317
COPY PDS ステートメント 317
MOVE 317
MOVE DSGROUP 315
MOVE DSNAME ステートメント 312
MOVE PDS ステートメント 317
例 323, 324
anyname (任意名) DD ステートメント 309
EXCLUDE ステートメント 328
EXEC ステートメント 308
構文 308
JOB ステートメント 307
RACF 保護 298
SMS ボリューム、移動またはコピー 307
SYSIN DD ステートメント 309
SYSPRINT DD ステートメント 309
SYSUT1 DD ステートメント 309
tape DD ステートメント 311
IEHPROGM プログラム
出力 337
ジョブ制御ステートメント 337, 339
スクラッチ、データ・セットまたはメンバーの 331
説明 331
名前変更
データ・セット 332
メンバー 332
入力 337
パスワード
維持 333
削除 336
追加 335
パスワード項目、リスト表示 336
戻りコード 366
ユーティリティ制御ステートメントの使用 339
CATLG ステートメント 343
DELETEP 346
LIST 347
RENAME ステートメント 342
SCRATCH ステートメント 340
UNCATLG ステートメント 343
例 347
- IEHPROGM プログラム (続き)
anyname (任意名) DD ステートメント 338
EXEC ステートメント 338
JOB ステートメント 337
SYSIN DD ステートメント 338
SYSPRINT DD ステートメント 338
IEHPROGM を使用して索引を作成する 331
IFASMFDP 128
IFASMFDP 磁気テープ 354
IFHSTATR プログラム
印刷出力のサンプル 355
磁気テープの品質 354
ジョブ制御ステートメント 356
説明 353
入出力 354
例 356
EXEC ステートメント 356
JOB ステートメント 356
SYSUT1 DD ステートメント 356
SYSUT2 DD ステートメントの例 356
INCLUDE ステートメント
COPY ステートメント
DSGROUP 312
PDS 312
IEBIMAGE プログラムの構文規則 184
IEHMOVE プログラム 321
MOVE ステートメント
DSGROUP 312
PDS 312
INDD DD ステートメント、IEBCOPY プログラム 47, 53
INDD パラメーター
COPY ステートメント 50
COPYGRP ステートメント 51
COPYMOD ステートメント 52
INHDR/INTLR パラメーター、ユーザー出力ルーチン 377
INITT ステートメント、IEHINITT プログラム 272, 275
構文 272
例 272
ACCESS パラメーター 274
DISP パラメーター 273
LABTYPE パラメーター 274
NUMBTAPE パラメーター 274
OWNER パラメーター 273
SER パラメーター 273
INREC/OUTREC パラメーター、ユーザー出力ルーチン 377
INSERT パラメーターの制約事項 247
IOERROR パラメーター、ユーザー出力ルーチン 377
- ISAM データ・セット、VSAM への変換 209
ISO/ANSI
ボリュームのアクセス保護 274
ISO/ANSI ボリュームのアクセス保護 274
- J**
JCL (ジョブ制御言語)、IEBDG プログラム 87
- K**
KEY パラメーター、ユーザー出力ルーチン 377
- L**
LABELS ステートメント
処理ルーチン 381
IEBCOMPR プログラムの構文規則 15
IEBGENER プログラムの構文規則 132
IEBTPCH プログラムの構文規則 225
IEBUPDTE プログラム 249
LABTYPE パラメーター
INITT ステートメント 274
LC パラメーター、IEBCOPY プログラム 42
LINECNT パラメーター、IEBCOPY プログラム 42
LINK マクロ 270
パラメーター・リスト 362
ユーティリティ・プログラムの構文規則 357
LIST ステートメント、IEHPROGM プログラム 346
LIST パラメーター、IEBCOPY プログラム 43
LISTPDS ステートメント、IEHLIST プログラム
構文 288
LISTVTOC ステートメント、IEHLIST プログラム
構文 289
LookAt メッセージ検索ツール xv
LPP パラメーター、IEBCOPY プログラム 42

M

MAXBLK パラメーター、IEBCOPY プログラム 52

MEMBER ステートメント
IEBGENER プログラム
構文 133
説明 133

IEBPTPCH プログラム
構文 221
説明 221

MEMBER パラメーター、IEBCOPY プログラム
名前変更、メンバーの 31
EXCLUDE ステートメント 54
SELECT ステートメント 55

MINBLK パラメーター、IEBCOPY プログラム 53

MOVE DSGROUP ステートメント、IEHMOVE プログラム 315, 317

MOVE DSNAME ステートメント、IEHMOVE プログラム 312, 313, 315

MOVE PDS ステートメント、IEHMOVE プログラム 317, 319

MOVE VOLUME ステートメント、IEHMOVE プログラム 319, 321

N

NAME ステートメントの構文、IEBIMAGE プログラム 184

NUMBER ステートメント、IEBUPDTE プログラム 245, 249

NUMBTAPE パラメーター
INITT ステートメント 274

O

OPEN マクロ
ユーザー・ラベル 383
CLOSE マクロ 383

OPTCD=W 45

OPTION ステートメント、IEBIMAGE プログラム
構文 185
ライブラリー文字セット・モジュール 185

OUTHDR/OUTLR パラメーター、ユーザー出口ルーチン 377

P

PARM パラメーター (EXEC ステートメント)
IEBGENER プログラム 125

PARM パラメーター (EXEC ステートメント) (続き)
IEHINITT プログラム 270

PDSE (拡張区分データ・セット)
アンロード 25
アンロード・フォーマット、IEBCOPY 367
印刷 211
更新 235
コピー 25
作成
順次入力 120
ライブラリー、区分メンバーの 235
穿孔 211
ソース言語の変更 235
ディレクトリー
印刷 213
穿孔 213
非編集 (ダンプ) フォーマット 282
編集されたリスト 281
リスト表示 281, 282, 288
ディレクトリー情報
コピー 26
ディレクトリーの印刷
例 291
バックアップ 25
比較 11, 21
プログラム・オブジェクト
置換 81
PDSE に 80
変換
区分データ・セットの例から 79
区分データ・セットへの 24
順次に変換 236
変更 235
メンバー
コピー 28
コピーから除外 29
置換 30, 32
追加 121
名前変更 31
別名を使用するコピー 29
ロード 26

PRECOMP パラメーター、ユーザー出口ルーチン 377

PRINT ステートメント、IEBPTPCH プログラム 216, 219

PUNCH ステートメント、IEBPTPCH プログラム 216, 219

R

RACF (リソース・アクセス管理機能)
IEHMOVE プログラム 298

RECORD ステートメント
IEBGENER プログラム 134, 138
IEBPTPCH プログラム
構文 223, 225
レコード・グループの定義 222

RENAME ステートメント
IEHPROGM プログラム 343

RENAME ステートメント、IEHPROGM プログラム 342

RENAME パラメーター
SMS ボリュームへの
MOVE/COPY 293

REPEAT ステートメント、IEBDG プログラム 96

REPL ステートメント、IEBUPDTE プログラム 245

REPLACE オプション、使用 31

REPLACE ステートメント
副次的な制御ステートメント 312
IEBUPDTE プログラム 239
IEBUPDTE プログラム、例 255
IEHMOVE プログラム 323
IEHPROGM プログラム 344

REPLACE パラメーター、IEBCOPY プログラム 43

REPRO ステートメント、IEBUPDTE プログラム 239, 245

RETURN マクロ
フォーマット 378

RETURN マクロ、出口ルーチン 378

RLD (再配置辞書)、挿入カウント 35

S

SCRATCH ステートメント
IEHPROGM プログラム 340, 342

SCRATCH ステートメント、IEHPROGM プログラム 340

SDB (ストレージ記述子ブロック) xx, 125, 128

SELECT ステートメント
名前変更、メンバーの 31
メンバーの置換 32

COPY ステートメント、DSGROUP 312
IEBCOPY プログラム 47, 54, 56
IEHMOVE プログラム 322

MOVE ステートメント、DSGROUP 312

SER パラメーター
INITT ステートメント 273

SETPRT マクロ 150, 167

SIO 使用回数 356

SIZE パラメーター、IEBCOPY プログラム 43

SMF (システム管理機能)、タイプ 21 レコードのフォーマット 353

SMS (ストレージ管理サブシステム)
 移動 293
 索引付き VTOC リスト 285
 スクラッチ 331
 データ・セットの事前割り当て 293

SMS.IND フィールド、フォーマット済み VTOC リスト表示の 390

SPCLCMOD 43

SYS1.IMAGELIB データ・セット
 維持 149
 ストレージ要件 147

SYS1.MAN 磁気テープ 354

SYS1.MANX データ・セット 354

SYS1.MANY データ・セット 354

SYS1.PROCLIB の例 252

SYS1.VTOCIX データ・セット 289

SYSIN DD ステートメント
 IEBCOMPR プログラム 13
 IEBCOPY プログラム 41, 47
 IEBEDIT プログラム 112, 288
 IEBGENER プログラム 125, 129
 IEBIMAGE プログラム 167
 IEBPTPCH プログラム 214, 215
 IEBUPDTE プログラム 239
 IEHINIT プログラム 270, 272
 IEHLIST プログラム 286
 IEHMOVE プログラム 311
 IEHPROGM プログラム 338, 339

SYSOUT データ・セットの例、印刷 231

SYSPRINT DD ステートメント
 IEBCOMPR プログラム 13
 IEBCOPY プログラム 41, 44
 IEBEDIT プログラム 112
 IEBGENER プログラム 124, 127
 IEBIMAGE プログラム 167
 IEBPTPCH プログラム 214
 IEBUPDTE プログラム 238
 IEHINIT プログラム 270
 IEHLIST プログラム 286, 287
 IEHMOVE プログラム 309
 IEHPROGM プログラム 338

SYSUT DD プログラム、IEBPTPCH プログラム 214

SYSUT1 DD ステートメント
 IEBCOMPR プログラム 13
 IEBCOPY プログラム 41, 45
 IEBEDIT プログラム 112
 IEBGENER プログラム 125, 127
 IEBIMAGE プログラム 167
 IEBPTPCH プログラム 214
 IEBUPDTE プログラム 238
 IEHMOVE プログラム 309
 IFHSTATR プログラム 356

SYSUT2 DD ステートメント
 IEBCOMPR プログラム 13
 IEBCOPY プログラム 41
 IEBEDIT プログラム 112
 IEBGENER プログラム 125
 IEBPTPCH プログラム 214, 215
 IEBUPDTE プログラム 238
 IFHSTATR の例 356

SYSUT3 DD ステートメント、IEBCOPY プログラム 41, 46

SYSUT4 DD ステートメント、IEBCOPY プログラム 41, 46

T

TABLE
 ステートメント
 構文 176, 178
 説明 176
 モジュール構造、IEBIMAGE プログラム 158
 モジュールのリスト表示、IEBIMAGE プログラム 160

tape DD ステートメント
 IEHMOVE プログラム 311

TIOT 339

TITLE ステートメント、IEBPTPCH プログラム
 構文 220
 説明 219

TOTAL パラメーター、ユーザー出口ルーチン 377

U

UNCATLG ステートメント
 (IEHPROGM) 343

UNCATLG ステートメント、IEHPROGM プログラム 343

UPDATE パラメーターの制約事項 244

V

VERSION パラメーター
 INITT ステートメント 274

VTOC (ボリューム目録)
 リスト表示
 項目 289
 索引付き 284
 非編集 (ダンプ) フォーマット 285
 編集フォーマット 283
 例 292
 IEHLIST プログラム 283, 285

VTOC (ボリューム目録) (続き)
 IEHLIST のフォーマット済みリスト表示 388
 IEHLIST プログラムの出力 387

W

WORK パラメーター、IEBCOPY プログラム 44



プログラム番号: 5694-A01

Printed in Japan

SC88-8979-03



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12