

DFSORT



# アプリケーション・プログラミングの手引き

リリース 14



DFSORT



# アプリケーション・プログラミングの手引き

リリース 14

**ご注意**

本書の情報およびそれによりサポートされる製品を使用する前に、751 ページの『付録 F. 特記事項』に記載する一般情報をお読みください。

**注:** この PDF は同じ資料番号のドキュメントを日本語検索可能にしたものです。以前のものと内容に違いはありません。

本書は、前の版、SC88-7061-03 に代わるものです。本書において導入された技術上の変更については「変更の要約」にまとめられています。また、変更箇所は左の欄外に、縦棒が示されています。技術上あまり重要でない編集上の変更については、特に示していません。

本書は、DFSORT リリース 14 (プログラム番号 5740-SM1)、および新版や TNL で特に断りがない限り、それ以降のすべてのリリースに適用されます。製品のレベルに合った正しい版をご使用ください。

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。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。

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)

原 典 : SC33-4035-21  
DFSORT  
Application Programming Guide  
Release 14

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 2002.6

この文書では、平成明朝体™W3、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™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 1973, 2002. All rights reserved.

© Copyright IBM Japan 2002

# 目次

まえがき	xiii
本書について	xiii
本書の追加情報の入手先	xiv
DFSORT 資料	xv
DFSORT ライブラリー・ソフトコピー情報	xv
表記の規則	xv
<b>変更の要約</b>	xvii
第 22 版 (2002.6)	xvii
リリース 14 の新しいプログラミング・サポート (2002 年 6 月 PTF)	xvii
第 21 版 (2000.7)	xxi
リリース 14 の新しいプログラミング・サポート (2000 年 7 月 PTF)	xxi
第 20 版 (1999.3)	xxiv
リリース 14 の新しいプログラミング・サポート (1999 年 3 月 PTF)	xxiv
第 19 版 (1998.9)	xxiv
新しいプログラミング・サポート (リリース 14)	xxiv
<b>第 1 章 DFSORT の紹介</b>	1
DFSORT の概要	1
Web 上の DFSORT	4
DFSORT の FTP サイト	4
DFSORT 呼び出し	4
DFSORT の機能のしくみ	5
オペレーティング・システム	5
制御フィールドおよび照合順序	6
文化的環境についての考慮事項	7
DFSORT 処理	8
入力データ・セット - SORTIN および SORTINnn	12
出力データ・セット - SORTOUT および OUTFIL	12
データ・セットに関する考慮事項	13
レコードの分類またはコピー	13
レコードを組み合わせる	13
データ・セットの注意事項および制約事項	14
SmartBatch パイプに関する考慮事項	18
HFS ファイルの考慮事項	18
インストール・デフォルト	18
他の分類用プロダクトから DFSORT への移行	25
DFSORT メッセージおよび戻りコード	26
可能な限りブロック・セットを使用する	27
<b>第 2 章 ジョブ制御言語による DFSORT の呼び出し</b>	29
JCL の使用	29
JOB ステートメントの使用	31
EXEC ステートメントの使用	32
EXEC ステートメントにカタログ式プロシージャを指定する	32
EXEC/DFSPARM PARM オプションの指定	34
PARM オプションの別名	66
DD ステートメントの使用	67
DD 名の重複	70

磁気テープ装置の共有	70
システム DD ステートメント	70
プログラム DD ステートメント	73
<b>第 3 章 DFSORT プログラム制御ステートメントの使用</b>	<b>111</b>
プログラム制御ステートメントの使用	114
制御ステートメントの概要	114
基本タスクの説明	114
レコードの組み込みまたは除外	115
レコードの再フォーマット設定と編集	115
複数の出力および報告書作成とレコード変換	115
追加機能およびオプションの呼び出し	115
シンボルの使用	116
一般的なコーディングの規則	116
継続行	118
注釈ステートメントの挿入	119
コーディング上の制約事項	119
ALTSEQ 制御ステートメント	120
EBCDIC 照合順序の更新 - 例	121
DEBUG 制御ステートメント	123
診断オプションの指定 - 例	128
END 制御ステートメント	129
制御ステートメント読み取りの中断 - 例	129
INCLUDE 制御ステートメント	130
関係条件	132
比較	133
出力データ・セットへのレコードの組み込み - 比較の例	139
サブストリング比較テスト	141
出力データ・セットへのレコードの組み込み - サブストリングの比較の例	142
ビット論理テスト	143
方法 1: ビット演算子のテスト	143
埋め込みと切り捨て	145
出力データ・セットにレコードを組み込む - ビット演算子テストの例	145
方法 2: ビット比較テスト	146
出力データ・セットにレコードを組み込む - ビット比較テストの例	148
日付比較	149
出力データ・セットへのレコードの組み込み - 日付比較	151
INCLUDE/OMIT ステートメントに関する注意事項	152
INREC 制御ステートメント	153
INREC ステートメントに関する注意事項	163
処理前のレコードの再フォーマット設定 - 例	165
MERGE 制御ステートメント	169
MERGE または COPY の指定 - 例	172
MODS 制御ステートメント	173
ユーザー出口ルーチンの指定 - 例	176
OMIT 制御ステートメント	177
出力データ・セットからレコードを除外する - 例	179
OPTION 制御ステートメント	181
OPTION ステートメント・オプションの別名	233
DFSORT オプションまたは COPY の指定 - 例	233
OUTFIL 制御ステートメント	238
OUTFIL ステートメントに関する注意事項	307

OUTFIL 機能 - 例 . . . . .	311
OUTREC 制御ステートメント . . . . .	327
OUTREC ステートメントに関する注意事項 . . . . .	335
出力レコードの再フォーマット設定 - 例 . . . . .	337
RECORD 制御ステートメント . . . . .	340
レコード形式および長さの記述 - 例 . . . . .	344
SORT 制御ステートメント . . . . .	345
SORT/MERGE ステートメントに関する注意事項 . . . . .	353
SORT または COPY の指定 - 例 . . . . .	353
SUM 制御ステートメント . . . . .	355
SUM ステートメントに関する注意事項 . . . . .	357
合計フィールドの加算 - 例 . . . . .	359
<b>第 4 章 ユーザー独自のユーザー出口ルーチンの使用 . . . . .</b>	<b>361</b>
ユーザー出口ルーチンの概要 . . . . .	362
DFSORT プログラム・フェーズ . . . . .	363
ユーザー出口ルーチンの機能 . . . . .	366
DFSORT の入力 / ユーザー出口 / 出力論理の例 . . . . .	366
データ・セットのオープンおよび初期設定 . . . . .	368
制御フィールドの変更 . . . . .	368
レコードの挿入、削除、および変更 . . . . .	368
レコードの合計 . . . . .	368
特殊入出力の処理 . . . . .	368
VSAM ユーザー出口機能 . . . . .	369
中間記憶域不足時の処置の決定 . . . . .	369
データ・セットのクローズ . . . . .	370
DFSORT の終了 . . . . .	370
ユーザー出口のアドレッシング・モードおよび常駐モード . . . . .	370
ユーザー出口ルーチンが DFSORT のパフォーマンスに与える影響 . . . . .	371
ユーザー出口ルーチンに関する規則 . . . . .	371
ユーザー出口ルーチンのロード方法 . . . . .	372
ユーザー出口関係規則 . . . . .	373
ユーザー出口ルーチンのダイナミック・リンク編集の方法 . . . . .	374
アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口) . . . . .	375
E11 ユーザー出口: データ・セットのオープン / ルーチンの初期設定 . . . . .	375
E15 ユーザー出口: レコードの引き渡しまたは変更 (分類アプリケーション およびコピー・アプリケーションの場合) . . . . .	375
E16 ユーザー出口: 中間記憶装置の計算違いの処理 . . . . .	378
E17 ユーザー出口: データ・セットのクローズ . . . . .	379
E18 ユーザー出口: 入力データ・セットの処理 . . . . .	379
E19 ユーザー出口: 作業データ・セットへの出力の処理 . . . . .	383
E61 ユーザー出口: 制御フィールドの変更 . . . . .	384
アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口) . . . . .	386
E31 ユーザー出口: データ・セットのオープン / ルーチンの初期設定 . . . . .	386
E32 ユーザー出口: 組み合わせのみの入力の処理 . . . . .	386
E35 ユーザー出口: レコードの変更 . . . . .	388
E37 E37 ユーザー出口: データ・セットのクローズ . . . . .	392
E38 ユーザー出口: 入力データ・セットの処理 . . . . .	392
E39 ユーザー出口: 出力データ・セットの処理 . . . . .	392
アセンブラーで作成されたサンプル・ルーチン . . . . .	393
E15 ユーザー出口: レコード長の変更 . . . . .	393
E16 ユーザー出口: NMAX 超過時の現行レコードの分類 . . . . .	394

E35 ユーザー出口: レコード長の変更 . . . . .	395
E61 ユーザー出口: 制御フィールドの変更 . . . . .	396
COBOL ユーザー出口ルーチン . . . . .	396
COBOL ユーザー出口に関する要件 . . . . .	397
COBOL ユーザー出口ルーチン (入力フェーズのユーザー出口) . . . . .	400
COBOL E15 ユーザー出口: レコードの引き渡しまたは変更 . . . . .	400
COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口) . . . . .	407
COBOL E35 ユーザー出口: レコードの変更 . . . . .	407
COBOL で作成されたサンプル・ルーチン . . . . .	414
COBOL E15 ユーザー出口: レコードの変更 . . . . .	414
COBOL E35 ユーザー出口: レコードの挿入 . . . . .	415
E15/E35 戻りコードおよび EXITCK . . . . .	417
<b>第 5 章 プログラムからの DFSORT の呼び出し . . . . .</b>	<b>421</b>
DFSORT の動的呼び出し . . . . .	421
システム・マクロ命令とは . . . . .	421
システム・マクロ命令の使用 . . . . .	422
JCL DD ステートメントの使用 . . . . .	422
DFSORT 制御ステートメントをプログラムから指定変更する . . . . .	422
24 ビット・パラメーター・リストを用いた DFSORT の呼び出し . . . . .	423
プログラム制御ステートメントの準備 . . . . .	423
拡張パラメーター・リストによる DFSORT の呼び出し . . . . .	431
プログラム制御ステートメントの準備 . . . . .	431
マクロ命令の書き方 . . . . .	434
パラメーター・リストの例 . . . . .	435
動的呼び出しに関する制約事項 . . . . .	439
組み合わせに関する制約事項 . . . . .	439
コピーに関する制約事項 . . . . .	439
<b>第 6 章 ICETOOL の使用 . . . . .</b>	<b>441</b>
概要 . . . . .	443
ICETOOL/DFSORT の関係 . . . . .	443
ICETOOL JCL の要約 . . . . .	444
ICETOOL 演算子の要約 . . . . .	445
完全な ICETOOL の例 . . . . .	446
シンボルの使用 . . . . .	447
ICETOOL の呼び出し . . . . .	447
ICETOOL の使用 . . . . .	448
ICETOOL のジョブ制御言語 . . . . .	451
JCL に関する制約事項 . . . . .	453
ICETOOL ステートメント . . . . .	454
一般的なコーディングの規則 . . . . .	454
COPY 演算子 . . . . .	455
オペランドの説明 . . . . .	455
COPY の例 . . . . .	458
COUNT 演算子 . . . . .	459
オペランドの説明 . . . . .	460
COUNT の例 . . . . .	462
DEFAULTS 演算子 . . . . .	462
オペランドの説明 . . . . .	464
DEFAULTS の例 . . . . .	465
DISPLAY 演算子 . . . . .	467



簡単な報告書	468
調整済み報告書	469
セクションごとにまとめた報告書	470
オペランドの説明	471
DISPLAY の例	492
MODE 演算子	508
オペランドの説明	509
MODE の例	509
OCCUR 演算子	510
簡単な報告書	511
調整済み報告書	512
オペランドの説明	514
OCCUR の例	521
RANGE 演算子	524
オペランドの説明	525
RANGE の例	526
SELECT 演算子	527
オペランドの説明	529
SELECT の例	531
SORT 演算子	534
オペランドの説明	535
SORT の例	536
STATS 演算子	539
オペランドの説明	539
STATS の例	540
UNIQUE 演算子	541
オペランドの説明	542
UNIQUE の例	542
VERIFY 演算子	543
オペランドの説明	544
VERIFY の例	544
プログラムからの ICETOOL の呼び出し	545
TOOLIN インターフェース	545
パラメーター・リスト・インターフェース	545
ICETOOL に関する注意事項および制約事項	551
ICETOOL の戻りコード	552
<b>第 7 章 フィールドと定数のシンボルの使用</b>	<b>555</b>
フィールドおよび定数シンボルの概説	555
DFSORT の例	556
SYMNAMES DD ステートメント	558
SYMNOUT DD ステートメント	559
SYMNAMES ステートメント	559
注釈ステートメントとブランク・ステートメント	559
シンボル・ステートメント	559
キーワード・ステートメント	567
SYMNAMES ステートメントのチェックのための SYMNOUT の使用	569
DFSORT ステートメントでのシンボルの使用	570
SORT と MERGE	571
SUM	571
INCLUDE と OMIT	572
INREC と OUTREC	572

OUTFIL . . . . .	573
ICETOOL 演算子でのシンボルの使用 . . . . .	575
COUNT . . . . .	575
DISPLAY . . . . .	576
OCCUR . . . . .	576
RANGE . . . . .	576
SELECT . . . . .	576
STATS、UNIQUE および VERIFY . . . . .	576
ICETOOL の例 . . . . .	577
シンボルに関する注意事項 . . . . .	578
<b>第 8 章 拡張機能サポートの使用 . . . . .</b>	<b>581</b>
EFS の使用 . . . . .	582
EFS プログラムのアドレッシング・モードと常駐モード . . . . .	583
EFS の働き . . . . .	583
DFSORT プログラム・フェーズ . . . . .	584
DFSORT による EFS プログラムの呼び出し . . . . .	584
EFS を使用して行えること . . . . .	589
データ・セットのオープンおよび初期設定 . . . . .	590
制御ステートメントの検査、変更、または無視 . . . . .	590
EFS プログラム・ユーザー出口ルーチンによるユーザー定義データ・タイプ の処理 . . . . .	593
メッセージ・データ・セットに印刷するためのメッセージの提供 . . . . .	593
DFSORT の終了 . . . . .	593
データ・セットのクローズおよびハウスキーピング . . . . .	593
EFS インターフェース・パラメーター・リストの構造 . . . . .	593
処置コード . . . . .	595
制御ステートメント要求リスト . . . . .	596
EFS プログラムへ送られる制御ステートメント・ストリング . . . . .	596
EFS プログラムにより戻される制御ステートメントのストリング . . . . .	598
SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形 式 . . . . .	600
FIELDS オペランドの D1 形式 . . . . .	600
COND オペランドの D2 形式 . . . . .	601
元の制御ステートメントの長さ . . . . .	602
変更後の制御ステートメントの長さ . . . . .	603
EFS プログラム・コンテキスト域 . . . . .	603
抽出バッファー・オフセット・リスト . . . . .	603
レコード長リスト . . . . .	603
情報フラグ . . . . .	604
メッセージ・リスト . . . . .	605
EFS プログラム出口ルーチン . . . . .	606
EFS01 および EFS02 機能の説明 . . . . .	606
EFS01 ユーザー出口ルーチン . . . . .	607
EFS02 ユーザー出口ルーチン . . . . .	608
EFS プログラムのユーザー出口ルーチンのアドレッシング・モードおよび常 駐モード . . . . .	611
ユーザーが指定しなければならない EFS プログラム戻りコード . . . . .	612
レコード処理の順序 . . . . .	612
SNAP ダンプの要求の方法 . . . . .	616
EFS プログラムの例 . . . . .	616
DFSORT 初期設定フェーズ . . . . .	616

DFSORT 終了フェーズ . . . . .	619
<b>第 9 章 効率の改善</b> . . . . .	<b>621</b>
パフォーマンスの向上 . . . . .	622
パフォーマンスを最大にするためのユーザー・アプリケーションの設計 . . . . .	623
DFSORT 処理の直接呼び出し . . . . .	623
新しいアプリケーション設計の事前計画 . . . . .	623
効率的な分類 / 組み合わせ手法の指定 . . . . .	624
入出力データ・セット特性の正確な指定 . . . . .	625
順次式ストライピングの使用 . . . . .	626
圧縮の使用 . . . . .	626
DFSMSrmm 管理テープ、または ICETPEX の使用 . . . . .	626
SmartBatch パイプの使用 . . . . .	626
拡張記憶域での VIO の使用 . . . . .	626
経過時間を改善する装置の指定 . . . . .	626
パフォーマンスを向上させるオプションの使用 . . . . .	627
DFSORT の高速で、効率的な生産性機能の使用 . . . . .	629
パフォーマンスを低下させるオプションの回避 . . . . .	630
主記憶域の効率的な使用 . . . . .	631
一時ワークスペースの効率的な割り振り . . . . .	636
ハイパー分類の使用 . . . . .	637
データ空間を使用する分類 . . . . .	638
IEBGENER に代わる、ICEGENER の使用 . . . . .	639
ICEGENER の戻りコード . . . . .	642
SAS システム用の DFSORT のパフォーマンス・ブースターの使用 . . . . .	643
DFSORT の BLDINDEX サポートの使用 . . . . .	643
<b>第 10 章 DFSORT ジョブ・ストリームの例</b> . . . . .	<b>645</b>
例の要約 . . . . .	645
記憶管理責任者の例 . . . . .	646
REXX の例 . . . . .	647
CLIST の例 . . . . .	647
分類処理の例 . . . . .	648
例 1. ALTSEQ を使用した分類処理 . . . . .	648
例 2. OMIT、SUM、OUTREC、DYNALLOC および ZDPRINT を使用した 分類処理 . . . . .	650
例 3. ISCI/ASCII 磁気テープ装置を使用した分類処理 . . . . .	652
例 4. E15、E35、FILSZ、AVGRLN、および DYNALLOC を使用した分類 処理 . . . . .	654
例 5. SORTCNTL、CHALT、DYNALLOC、および FILSZ を使用した呼び出 し済み分類処理 . . . . .	656
例 6. VSAM 入力 / 出力、DFSPARM、およびオプション指定変更を使用し た分類処理 . . . . .	658
例 7. COBOL E15、EXEC PARM、COBEXIT、および MSGDDN を使用し た分類処理 . . . . .	660
例 8. 出口のダイナミック・リンク編集を使用した分類処理 . . . . .	662
例 9. 拡張パラメーター・リスト・インターフェースを使用した分類処理 . . . . .	664
例 10. OUTFIL を使用した分類処理 . . . . .	667
例 11. SmartBatch パイプおよび OUTFIL SPLIT を使用した分類処理 . . . . .	669
例 12. INCLUDE および LOCALE を使用した分類処理 . . . . .	670
例 13. HFS ファイルでの分類 . . . . .	671
組み合わせの例 . . . . .	673

例 1. EQUALS を使用した組み合わせ . . . . .	673
例 2. LOCALE および OUTFIL を使用した組み合わせ処理 . . . . .	674
コピー処理の例 . . . . .	676
例 1. EXEC PARMs、SKIPREC、MSGPRT および ABEND を使用したコピー処理 . . . . .	677
例 2. INCLUDE および VLSHRT を使用したコピー処理 . . . . .	679
ICEGENER の例 . . . . .	681
ICETOOL の例 . . . . .	683
<b>付録 A. ワークスペースの使用 . . . . .</b>	<b>685</b>
はじめに . . . . .	685
ハイパー空間 . . . . .	686
作業データ・セット装置 . . . . .	686
DASD および磁気テープ装置 . . . . .	686
装置の数 . . . . .	687
非同期記憶サブシステム . . . . .	687
作業データ・セットの割り振り . . . . .	688
作業データ・セットの動的割り振り . . . . .	689
ワークスペースの動的過剰割り振り . . . . .	691
作業データ・セットの JCL 割り振り . . . . .	691
DASD 容量に関する考慮事項 . . . . .	693
DASD ワークスペース容量の超過 . . . . .	693
磁気テープ容量に関する考慮事項 . . . . .	694
磁気テープ・ワークスペース容量の超過 . . . . .	694
<b>付録 B. DFSORT オプションの指定 / 指定変更 . . . . .</b>	<b>697</b>
DFSORT オプションのソースの主な機能 . . . . .	698
DFSPARM データ・セット . . . . .	698
EXEC ステートメントの PARM オプション . . . . .	698
SORTCNTL データ・セット . . . . .	698
SYSIN データ・セット . . . . .	699
パラメーター・リスト . . . . .	699
指定変更テーブル . . . . .	699
直接呼び出し DFSORT . . . . .	700
直接呼び出し DFSORT テーブルの注 . . . . .	711
拡張パラメーター・リストを用いたプログラム呼び出し DFSORT . . . . .	711
拡張パラメーター・リスト・テーブルの注 . . . . .	721
24 ビット・パラメーター・リストを用いたプログラム呼び出し DFSORT . . . . .	721
24 ビット・リスト・テーブルの注 . . . . .	731
<b>付録 C. データ形式記述 . . . . .</b>	<b>733</b>
<b>付録 D. EBCDIC および ISCI/ASCII の照合順序 . . . . .</b>	<b>741</b>
EBCDIC . . . . .	741
ISCI/ASCII . . . . .	743
<b>付録 E. DFSORT の異常終了処理 . . . . .</b>	<b>747</b>
チェックポイント / 再始動 . . . . .	747
DFSORT の異常終了の種類 . . . . .	748
予期しない異常終了の場合の異常終了リカバリー処理 . . . . .	748
A タイプ・メッセージを伴うエラー異常終了の処理 . . . . .	749
CTR <sub>x</sub> 異常終了処理 . . . . .	750

付録 F. 特記事項 . . . . .	751
プログラミング・インターフェースの情報 . . . . .	752
商標 . . . . .	752
変更の要約 . . . . .	753
リリース 13 の新しいプログラミング・サポート (1996 年 4 月 以降の PTF) . . . . .	753
新規 2000 年対応機能 . . . . .	753
OS/390 の登録 . . . . .	753
リリース 13 の新しいプログラミング・サポート (1996 年 4 月 PTF) . . . . .	753
2000 年対応機能 . . . . .	753
FLR および VLR ブロック・セット分類におけるパフォーマンスの向上 . . . . .	753
SUM の浮動小数点 . . . . .	754
セキュリティーの向上 . . . . .	755
EXCPVR 処理の除去 . . . . .	755
リリース 13 で新たに加わった装置サポート (PTF) . . . . .	755
新しいプログラミング・サポート (リリース 13) . . . . .	755
SAS** システム用の DFSORT のパフォーマンス・ブースター . . . . .	755
動的ハイパー分類 . . . . .	755
パフォーマンス . . . . .	755
OUTFIL の処理 . . . . .	756
各国語サポート . . . . .	756
ICETOOL の機能強化 . . . . .	757
INCLUDE/OMIT サブストリングの検索 . . . . .	757
SMF タイプ 16 レコードの機能強化 . . . . .	758
その他の機能強化 . . . . .	758
リリース 12 で新たに加わったプログラミング・サポート (PTF) . . . . .	759
リリース 12 で新たに加わった装置サポート (PTF) . . . . .	760
索引 . . . . .	761



---

## まえがき

本書は DFSORT を使用してデータ・セットの分類、組み合わせ、およびコピーを行う場合に役立つように作成されています。本書は DFSORT の使用法を学習するためのものではなく、すでに DFSORT の基本を理解していて、その機能やオプションについて、作業 (タスク) 本位の手引きを必要とするプログラマーのためのものです。初めてお使いになる場合、まず *DFSORT 使用者の手引き R14* をお読みください。*DFSORT 使用者の手引き R14* は、DFSORT を使い始める前に知っておく必要のある事項を簡単に説明した自習書で、例と図表がステップごとに示されています。

---

## 本書について

本書の各節は、ユーザーが行う作業別にまとめられた、関連情報について述べています。最初の 4 つの章では、DFSORT の基本的なレコード処理機能呼び出して使用する上で理解しておくべき事項について説明しています。残りの章では、特殊機能について説明しています。付録では、各種のトピックについての特定情報を提供しています。

- 1 ページの『第 1 章 DFSORT の紹介』は、DFSORT の概要を示し、DFSORT を使用してできること、そして DFSORT 処理を呼び出す方法について説明します。また、DFSORT の機能、データ・セットの形式と制約、およびユーザーのシステムでインストール時に変更できるデフォルトについて説明します。
- 29 ページの『第 2 章 ジョブ制御言語による DFSORT の呼び出し』は、DFSORT ジョブを実行するためのジョブ制御言語 (JCL) の使用方法を説明しています。JOB、EXEC、および DD ステートメントのコーディングの方法を説明し、またカタログ式プロシージャおよび EXEC PARM オプションを使用してユーザーの JCL を簡略化し、インストール時に設定した DFSORT デフォルトを指定変更する方法について説明しています。
- 111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』では、データの分類、組み合わせ、およびコピーを行うための DFSORT 制御ステートメントについて説明しています。必要なデータだけを使用できるようにするためにデータにフィルターをかける方法、レコードを再フォーマット設定または総計して編集する方法、複数の出力データ・セットと報告書を作成する方法などを説明しています。また、処理中にユーザーのルーチンを使用するよう DFSORT に指示するステートメントの書き方についても説明しています。
- 361 ページの『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』では、DFSORT のプログラム出口を使用して、プログラム処理中にユーザーのルーチンを呼び出す方法を説明しています。レコードを削除、挿入、変更、および合計するためのルーチンを作成したり、またユーザー独自のエラー・リカバリー・ルーチンを組み込むこともできます。
- 421 ページの『第 5 章 プログラムからの DFSORT の呼び出し』では、システム・マクロ命令を使用して、ユーザーのアセンブラー・プログラムから DFSORT 処理を開始する方法について説明しています。また、PL/I および COBOL から DFSORT を呼び出す際の制約事項についても記載しています。

- 441 ページの『第 6 章 ICETOOL の使用』では、多目的 DFSORT ユーティリティー ICETOOL の使用方法を説明しています。ICETOOL を使用して各種の作業を実行するために使用できる JCL および演算子について説明しています。
- 555 ページの『第 7 章 フィールドと定数のシンボルの使用』は、シンボルを定義する方法と、その定義したシンボルを DFSORT 制御ステートメントと ICETOOL 演算子で使用方法を説明しています。
- 581 ページの『第 8 章 拡張機能サポートの使用』では、拡張機能サポート (EFS) インターフェースを使用して、制御ステートメントを調整したり、ユーザー定義のデータ・タイプや照合順序を扱ったり、またカスタマイズした通知メッセージを、処理の過程で DFSORT に発行させる方法について説明しています。
- 621 ページの『第 9 章 効率の改善』では、DFSORT 処理効率を最大にするためにお勧めする方法について説明しています。この章では、お客さまのシステムで効率的に処理を行うために、個々のアプリケーションを設計することから、ハイパー分類、データ空間分類、ICEGENER などの DFSORT 機能を使用することまで、ユーザーが改善できる事項を広範囲に渡って述べています。
- 645 ページの『第 10 章 DFSORT ジョブ・ストリームの例』では、レコードを分類、組み合わせ、およびコピーするジョブ・ストリームの例を注釈を付けて示しています。
- 685 ページの『付録 A. ワークスペースの使用』では、主記憶域に関する考慮事項と、ユーザーがデータを分類するときに必要となる中間記憶域の大きさを見積もる方法について説明しています。
- 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』には、一連のテーブルが示されており、このテーブルを使用すると、異なるソースで指定された同様のオプションに対するオーバーライド指定順序を見つけることができます。
- 733 ページの『付録 C. データ形式記述』では、アセンブル済みデータ形式の例を示しています。
- 741 ページの『付録 D. EBCDIC および ISCI/ASCII の照合順序』では、EBCDIC および ISCI/ASCII 文字について、低位から高位への照合順序をリストしています。
- 747 ページの『付録 E. DFSORT の異常終了処理』では、異常終了 (ABEND) を処理するための ESTAE リカバリー・ルーチン、およびチェックポイント / 再始動機能について説明しています。
- 751 ページの『付録 F. 特記事項』には、特記事項、プログラミング・インターフェース情報、および商標リストが記載されています。

---

## 本書の追加情報の入手先

z/OS 資料の最新情報については、z/OS 情報ロードマップを参照してください。また、本書のソフトコピー版は z/OS インターネット・ライブラリー <http://www.ibm.com/servers/eservers/zseries/zos/bkserv/> より入手できます。

---

## DFSORT 資料

DFSORT アプリケーション・プログラミングの手引き リリース 14 は、より広範な情報を提供する DFSORT ライブラリーの一部です。ライブラリー内のその他の資料は、以下のとおりです。



作業	資料名	資料番号
DFSORT の計画およびカスタマイズ	DFSORT 導入とカスタマイズ リリース 14	SC88-7163
DFSORT パネル使用の学習	DFSORT パネルの手引き	GC88-7478
DFSORT 使用の学習	DFSORT 使用者の手引き R14	SC88-7160
早見表	DFSORT 参照要約 R14	N: SX33-8001
障害の診断およびメッセージの解釈	DFSORT メッセージ、コード、および診断の手引き リリース 14	SC88-7479
DFSORT の調整	DFSORT Tuning Guide R14	SC26-3111

DFSORT 資料の全セットは SBOF-1243 のオーダー番号で注文できますが、*DFSORT Licensed Program Specifications R14*、GC33-4032 は除きます。

## DFSORT ライブラリー・ソフトコピー情報

DFSORT ライブラリーのソフトコピーが次の CD-ROM でご利用いただけます。この CD-ROM には、それぞれ リリース 13 と 14 の DFSORT の資料がすべて含まれておりますが、*DFSORT 参照要約* は、入っていません。

資料番号	名称
SK3T-4269	z/OS Collection

## 表記の規則

本書に示されている構文図は、DFSORT プログラム制御ステートメントを簡潔、かつ正確にコーディングできるようになっています。線と矢印は、完成したステートメントの中に現れる順序と構文で、演算子、パラメーター、および区切り文字を結ぶ経路または流れ図を示しています。必要なすべてのパラメーターが含まれた適切なダイアグラムの経路に従ってステートメントを構成し、ダイアグラムの指示どおりの順序でパラメーターをコーディングしてください。以下の規則に従うと、ダイアグラム内のどの経路をとっても、正しいステートメントをコーディングできます。

- ダイアグラムは左から右へ、上から下へ読みます。
- ステートメントのコーディングは、2 つの矢印で表示されているところから開始します。



- 線の終わりに矢印が 1 つあるときは、ダイアグラムが次の行または示された箇所へ続くことを示しています。



継続行は 1 つの矢印で始まります。

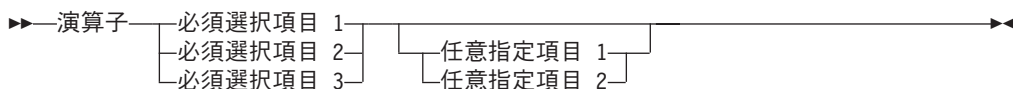


## 表記の規則

- 大文字のストリングと句読 (括弧、アポストロフィなど) は、表示どおりに正確にコーディングします。
  - プログラム制御ステートメントと EXEC PARM ストリングでは、セミコロンとコンマのどちらかを使用できます。煩雑さを避けるために、本書ではコンマのみ使用します。
- 小文字だけのストリングは、ユーザーが提供する情報を表します。
- 必須パラメーターは、演算子と同じ水平線 (主経路) 上に表示し、任意指定パラメーターは、主経路の下の分岐経路に表示しています。

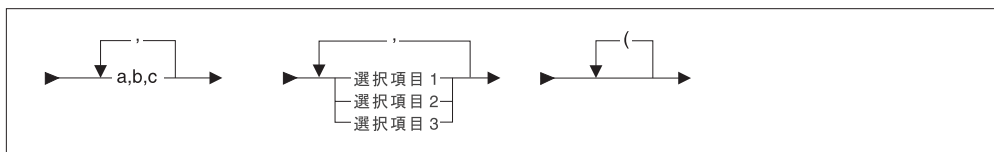


- 2 つ以上のパラメーターの中から 1 つを選択できる場合は、代替項目を縦に並べて (スタックして) 表示しています。



主経路に示されているスタックの項目の 1 つを選択する場合 (上記の例の左を参照) は、代替項目の 1 つを 指定します。スタックが主経路の下側に示されている場合 (上記の例の右を参照)、選択は任意で、そのうち 1 つを選択するか、あるいはすべて選択しないこともできます。

- 反復記号は、あるパラメーターを 2 回以上指定したい場合 (下記の左の例を参照)、同一スタックから一度に 2 つ以上の項目を選択したい場合 (下記の中央の例を参照)、あるいは括弧をネストしたい場合 (下記の例の右を参照) に、構文図内の前の位置に戻ることが可能であることを表しています。



反復記号を、非互換のパラメーターを指定できるという意味に解釈しないでください。たとえば、同一 EXEC ステートメントに **ABEND** と **NOABEND** の両方を指定したり、括弧を不正にネストしたりしないでください。

反復記号の中に表示されているすべての句読または区切り文字は、反復項目を区切るために使用されます。

- 線の終わりに 2 つの矢印があるのは、構文図がそこで終わることを示しています。



---

## 変更の要約

---

### 第 22 版 (2002.6)

#### リリース 14 の新しいプログラミング・サポート (2002 年 6 月 PTF)

##### **z900**

DFSORT は、ストレージ・スペースとデータ空間を 2 ギガバイトを超える実記憶装置に戻し、ハイパー分類用拡張記憶域の代わりに中央記憶装置を使用することで、z900 の 64 ビットの実アーキテクチャーを活用できるようになりました。

##### **複数のハイパー空間**

DFSORT は、十分なシステム・リソースを使用可能な場合にラージ分類アプリケーション用のハイパー分類を使用する能力が向上し、外部ストレージ要件に対応する複数のハイパー空間を使用できるようになりました。

##### **管理テープ**

DFSORT は、テープ出口ルーチン (ICETPEX) を提供するテープ管理システムが管理するテープの、入力ファイルのサイズの正確な情報を、自動的に取得できるようになりました。これにより、分類のパフォーマンスが向上し、動的ワークスペースの割り振りがさらに正確になります。

DFSORT は、テープ出口ルーチン (ICETPEX) を提供するテープ管理システムが管理するテープの RECFM、LRECL、および BLKSIZE などの、入出力属性を自動的に取得できるようになりました。

##### **ICETOOL の機能強化**

ICETOOL の COUNT 演算子の新規オプション、EMPTY、NOTEMPTY、HIGHER(n)、LOWER(n)、EQUAL(n)、および NOTEQUAL(n) により、データ・セットのレコード数に基づいて RC=12 または RC=0 を設定できます。

ICETOOL の SELECT 演算子の新規 FIRSTDUP オプションにより、重複するレコードの各グループそれぞれについて、最初のレコードのみ保持できます。

ICETOOL の SELECT 演算子の新規 LASTDUP オプションにより、重複するレコードの各グループそれぞれについて、最後のレコードのみ保持できます。

ICETOOL の SELECT 演算子の DISCARD(savedd) オプションは、TO(outdd) オペランドを使用せずに設定できるようになりました。

ICETOOL で使用する CH フィールドの最大長は、1500 まで拡張されています。

ICETOOL で使用する HEX フィールドの最大長は、1000 まで拡張されています。

ICETOOL の DISPLAY および OCCUR 演算子の新規 DT1、DT2 および DT3 形式により、SMF 日付値を分かりやすく表記できます。

ICETOOL の DISPLAY および OCCUR 演算子の新規 TM1、TM2、TM3、および TM4 形式により、SMF 時間値を分かりやすく表記できます。

ICETOOL の DISPLAY および OCCUR 演算子の新規 INDENT(n) オプションで、インデントのブランクの数を指定できます。

ICETOOL の DISPLAY および OCCUR 演算子の新規 BETWEEN(n) オプションで、列間のブランクの数を指定できます。

ICETOOL の DISPLAY 演算子の新規 STATLEFT オプションで、データの最初の列の左側に、統計のヘッダーを印刷できます。

フォーマット項目の使用は、ICETOOL の DISPLAY 演算子のレコード長、レコード番号、段落フィールド、および ICETOOL の OCCUR 演算子のデータ・フィールド、レコード長およびレコード値の数まで拡張されています。

また、以下のフォーマット項目が追加されました。E'pattern' (電話番号、社会保障番号、日付など、指定されたパターンの数値を使用)、/D (10 単位で区切る)、/C (100 単位で区切る)、/DK (10000 単位で区切る)、/CK (100000 単位で区切る)、LZ (マスクの先行ゼロ)、NOST (統計なし)、Ndd (dd 桁)

ICETOOL の DISPLAY および OCCUR 演算子の新規 DATENS(abc) オプションにより、区切り文字を使用せずに ICETOOL 実行の日付をタイトルに挿入できます。区切り文字は既存のオプション、DATE および DATE(abcd) に含まれます。

ICETOOL の新規 DISPLAY および OCCUR 演算子の新規 TIMENS(ab) オプションにより、区切り文字を使用せずに ICETOOL 実行の時間をタイトルに挿入できます。区切り文字は既存のオプション、TIME および TIME(abc) に含まれます。

## **OUTFIL の機能強化**

OUTFIL の新規 FTOV オプションで、固定長入力レコード (たとえば FB) を可変長出力レコード (たとえば VB) に変換できます。OUTFIL の新規 VTOF オプション (CONVERT のエイリアス) で、可変長入力レコード (たとえば VB) を固定長出力レコード (たとえば FB) に変換できます。

OUTFIL の新規 VLTRIM=byte オプションで、ブランク、2 進ゼロ またはアスタリスクなどの後書きバイトを可変長出力レコードから削除できます。

OUTFIL の新規 REMOVECC オプションで、報告書から ANSI 制御文字を削除できます。

OUTFIL OUTREC の新規 DATE<sub>n</sub>、DATE<sub>n</sub>(c) および DATE<sub>n</sub>P オプションで、DFSORT 実行の日付を様々な形式でレコードに挿入できます。

OUTFIL OUTREC の新規 TIME<sub>n</sub>、TIME<sub>n</sub>(c) および TIME<sub>n</sub>P オプションで、DFSORT 実行の時間を様々な形式でレコードに挿入できます。

OUTFIL OUTREC の新規 DT1、DT2 および DT3 形式により、SMF 日付値を分かりやすく表記できます。

OUTFIL OUTREC の新規 TM1、TM2、TM3 および TM4 形式により、SMF 時間値を分かりやすく表記できます。

OUTFIL OUTREC の新規 TRAN=LTOU オプションで、フィールド内の位置を問わず、EBCDIC 文字を小文字から大文字に変更できます。

OUTFIL OUTREC の新規 TRAN=UTOL オプションで、フィールド内の位置を問わず、EBCDIC 文字を大文字から小文字に変更できます。

OUTFIL OUTREC の新規 TRAN=ALTSEQ オプションで、フィールド内の位置を問わず、有効な ALTSEQ 変換テーブルに従って、文字を変更できます。

OUTFIL OUTREC の新規 M26 編集マスクで、ST...T (S は符号、T は有効数字) のパターンに従って、数値を編集できます。

OUTFIL INCLUDE および OMIT で使用するフィールドの終端位置は、最大 32752 まで拡張されました。

OUTFIL INCLUDE および OMIT の新規 DATE<sub>n</sub>、DATE<sub>n</sub>(c) および DATE<sub>n</sub>P オプションで、様々な形式のフィールドを DFSORT 実行の日付と比較できます。

正の 10 進数 (n または +n) は、OUTFIL INCLUDE および OMIT の 2 進 (BI) フィールドと比較できるようになりました。

OUTFIL HEADER<sub>x</sub> および TRAILER<sub>x</sub> の新規 DATENS(abc) オプションにより、区切り文字を使用せずに DFSORT 実行の日付をヘッダーとトレーラーに挿入できます。区切り文字は既存のオプション、DATE および DATE=(abcd) に含まれません。

OUTFIL HEADER<sub>x</sub> および TRAILER<sub>x</sub> の新規 TIMENS=(ab) オプションにより、区切り文字を使用せずに DFSORT 実行の時間をヘッダーとトレーラーに挿入できます。区切り文字は既存のオプション、TIME および TIME=(abc) に含まれます。

OUTFIL SECTIONS の新規 SKIP=0L オプションにより、同じページ中のセクション間のブランク行を抑止できます。

## **INREC および OUTREC の機能強化**

INREC および OUTREC の新規 DATE<sub>n</sub>、DATE<sub>n</sub>(c) および DATE<sub>n</sub>P オプションで、DFSORT 実行の日付を様々な形式でレコードに挿入できます。

INREC および OUTREC の新規 TIME<sub>n</sub>、TIME<sub>n</sub>(c) および TIME<sub>n</sub>P オプションで、DFSORT 実行の時間を様々な形式でレコードに挿入できます。

INREC および OUTREC の新規 DT1、DT2 および DT3 形式により、SMF 日付値を分かりやすく表記できます。

INREC および OUTREC の新規 TM1、TM2、TM3 および TM4 形式により、SMF 時間値を分かりやすく表記できます。

INREC および OUTREC の新規 TRAN=LTOU オプションで、フィールド内の位置を問わず、EBCDIC 文字を小文字から大文字に変更できます。

INREC および OUTREC の新規 TRAN=UTOL オプションで、フィールド内の位置を問わず、EBCDIC 文字を大文字から小文字に変更できます。

INREC および OUTREC の新規 TRAN=ALTSEQ オプションで、フィールド内の位置を問わず、有効な ALTSEQ 変換テーブルに従って、文字を変更できます。

INREC および OUTREC の新規 M26 編集マスクで、ST...T (S は符号、T は有効数字) のパターンに従って、数値を編集できます。

### **SUM の機能強化**

SUM ステートメントが指定されている場合、VLSHRT オプションで、短い SORT および MERGE 制御フィールドと、INCLUDE および OMIT 比較フィールドが処理できるようになりました。短い SUM 要約フィールドのレコードは合計されないままとなります。

### **SORT および MERGE の機能強化**

SORT および MERGE で使用するフィールドの終端位置は、最大 32752 まで拡張されました。

SORT および MERGE で使用する AQ または AC フィールドの最大長は、4092 まで拡張されました。

### **INCLUDE および OMIT の機能強化**

INCLUDE および OMIT の新規 DATEn、DATEn(c) および DATEnP オプションで、様々な形式のフィールドを DFSORT 実行の日付と比較できます。

INCLUDE および OMIT で使用するフィールドの終端位置は、最大 32752 まで拡張されました。

正の 10 進数 (n または +n) は、INCLUDE および OMIT の 2 進 (BI) フィールドと比較できるようになりました。

### **その他の機能強化**

ICEMAC オプションの DYNSPC に対する IBM 提供のデフォルトが、32 メガバイトから 256 メガバイトに変更されました。これにより、DFSORT はデフォルトで、入力ファイルのサイズが不明な場合、動的に割り振られたワークスペースを使用してより多くのデータを分類できます。

DYNSPC は実行時オプションとして指定できるようになりました。これにより、入力ファイルのサイズが不明な個別の分類アプリケーションに対して動的に割り振られたワークスペースを調整できます。

DFSORT は、入力ファイルのサイズが分類アプリケーションで不明な場合、メッセージ ICE118I を出して、SMF タイプ 16 のレコード・フィールド、ICEFSZFL の 3 ビット目を使用します。これにより、FILSZ=En または DYNSPC=n の値が有用となる分類アプリケーションを識別します。

DFSORT コピーは、DUMMY で代替 SYSIN DD 名を使用するプログラムから ICEGENER が呼び出された場合に使用できるようになりました。これは、ICEGENER が IEBGENER の置き換えとしてインストールされた場合、RACF の IRRUT200 ユーティリティのパフォーマンスを向上させます。

可変長レコードへのすべての入力が入力 E15 または E32 出口から提供され、また RECFM が出力データ・セットに指定されていないとき、DFSORT は、出力のシステム決定最適ブロック・サイズを使用できる場合に、出力 RECFM を非ブロック化ではなくブロック化されたものとして設定できるようになりました。

ハイフン (-) が DFSORT シンボル名の 2 番目以降の文字に使用できるようになりました。

QNAME オプションが出力 DD ステートメント上で指定される場合、DFSORT はメッセージ ICE042A を出し、終了します。

---

## 第 21 版 (2000.7)

### リリース 14 の新しいプログラミング・サポート (2000 年 7 月 PTF)

#### OS/390 R10 でのテープ・ブロック・サイズの増大

DFSORT は、入出力に 32760 バイトより大きいブロック・サイズをもつテープ・データ・セットを使用できるようになり、パフォーマンスおよびテープ使用効率が改善されました。

DFSORT は、有効な BLKSZLIM 値により許可されている場合、32760 バイトより大きいシステム決定最適ブロック・サイズを使用できるようになりました。インストール・オプションおよび実行時オプションの SDB=INPUT (新規の IBM 提供デフォルト)、SDB=LARGE (新規)、SDB=YES (またはその別名 SDB=SMALL) および SDB=NO を使用すると、テープ出力データ・セットの 32760 バイトより大きいブロック・サイズも含め、システム決定最適ブロック・サイズの DFSORT による使用を制御できます。

DFSORT の ICEGENER は、IEBGENER と同様、ユーザーが提供した SDB=value パラメーターを使用します。

#### 新規の装置サポート

DFSORT は FICON チャンネルをサポートするようになりました。

DFSORT は、IBM 2105 制御装置を認識するようになり、作業データ・セット用のキャッシュ高速書き込みなどの、キャッシュ制御装置に適した最適化を提供します。

DFSORT は、パフォーマンスに関して望ましい選択として、エミュレートされた 3390 モデル 9 装置での作業データ・セットを扱うようになりました。DFSORT は、引き続き、実 3390 モデル 9 装置での作業データ・セットをパフォーマンスに関して望ましくない選択として扱います。

#### DFSMSrmm 管理のテープ

DFSORT は、DFSMSrmm 管理テープの正確な入力ファイルのサイズ情報を自動的に取得できるようになりました。これにより、分類のパフォーマンスが向上し、動的ワークスペースの割り振りがさらに正確になります。

DFSORT は、DFSMSrmm 管理テープの RECFM、LRECL、および BLKSIZE などの、入出力属性を自動的に取得できるようになりました。

## VSAM 処理

DFSORT は、拡張フォーマット VSAM データ・セットに対する VSAM 拡張アドレッシング機能をサポートするようになり、これにより 4 ギガバイトより大きな VSAM データ・セットの可能性が提供されます。

DFSORT では、空の VSAM 入力データ・セットを受け入れ、ゼロのレコードをもつとして処理できるようになりました。新規のインストール・オプション VSAMEMT=YES/NO および実行時オプション VSAMEMT および NVSAMEMT を使用すると、空の VSAM 入力データ・セットの処理を制御できます。

DFSORT では、REUSE で定義された VSAM 出力データ・セットを、NEW または MOD データ・セットとして処理することができるようになりました。新規のインストール・オプション RESET=YES/NO および実行時オプション RESET および NORESET を使用すると、VSAM 出力データ・セットの処理を制御できます。

DFSORT は、REUSE で定義された VSAM データ・セットをインプレース分類できるようになりました。新規のインストール・オプション VSAMIO=YES/NO および実行時オプション VSAMIO および NOVAMIO を使用すると、入力および出力に対する同じ VSAM データ・セットの使用を制御できます。

## HFS ファイル

DFSORT は、階層ファイル・システム (HFS) ファイルを入力および出力用としてサポートするようになりました。

## 生産性および他の分類用プロダクトからの移行

デフォルトで、DFSORT は、SORTOUT LRECL が使用できないときの SORTOUT LRECL として、L3、OUTREC または INREC レコード長を使用するようになりました。新規のインストール・オプション SOLRF=YES/NO および実行時オプション SOLRF および NOSOLRF を使用すると、SORTOUT LRECL の再フォーマット設定されたレコード長の使用を制御できます。

DFSORT は、以下の機能を含む、OUTFIL の拡張機能を提供するようになりました。

- 可変長レコードから固定長レコードに変換するときに、デフォルトで、短いレコードをブランクで埋め込む。
- トレーラーに編集済みカウントおよびサブカウントを作成する。

DFSORT は、以前は OUTFIL OUTREC でだけ使用可能であった、INREC および OUTREC 用の以下の拡張機能を提供するようになりました。

- 16 進表示。
- 日付の変換。
- 数値フィールドの編集。
- 参照と変更。

DFSORT は以下の機能を含む、INREC、OUTREC、および OUTFIL OUTREC 用の拡張機能を提供するようになりました。

- 数値フィールドから他の数値形式への変換。
- 変換された日付の編集。



- 変換された日付から数値形式への変換。
- シーケンス番号の作成。

DFSORT は、可変長レコードを超える INCLUDE/OMIT フィールド (つまり、短フィールド) を、2 進ゼロで埋め込まれるのと同様に比較できるようになりました。これにより、短フィールドと非短フィールドを比較できます。新規のインストール・オプション VLSCMP=YES/NO および実行時オプション VLSCMP および NOVLSCMP を使用すると、短制御フィールドに対する 2 進ゼロによる埋め込みの使用を制御できます。

DFSORT は、-0 および +0 を、DFSORT ステートメント INCLUDE、INREC、MERGE、OMIT、OUTFIL、OUTREC、および SORT による処理の場合に、符号なし (つまり同じ) として扱えるようになりました。新規のインストール・オプション SZERO=YES/NO および実行時オプション SZERO および NOSZERO を使用すると、これらのステートメントに対する符号付き、または符号なしゼロ処理を選択できます。

DFSORT は、-0 および +0 を、ICETOOL 演算子 DISPLAY、OCCUR、SELECT、および UNIQUE による処理の場合に、符号なし (つまり同じ) として扱えるようになりました。新規の導入時オプション UZERO を使用すると、これらの操作に符号なしゼロ処理を選択できます。

DFSORT は常に、入力が VSAM であるときや、E15 または E32 出口がすべての入力レコードを提供するとき、レコード・タイプ (F または V) を判別または割り当てるようになりました。

DFSORT は、SYSIN、SORTCNTL、および DFSPARM で 80 より大きい LRECL を許可するようになりました。この場合、各制御ステートメントの最初の 80 バイトだけ使用されます。

DFSORT は、出力データ・セットの LRECL より長い可変長出力レコードを、LRECL まで切り捨てるできるようになりました。新規のインストール・オプション VLLONG=YES/NO および実行時オプション VLLONG および NOVLLONG を使用すると、可変長出力レコードの切り捨てを制御できます。

DFSORT は、EXEC/DFSPARM PARM オプションとして次のオプションを受け入れ、処理するようになりました。COBEXIT=COB1/COB2、NOVERIFY、NOVLSHRT、NZDPRINT、OVFLO=RC0/RC4/RC16、PAD=RC0/RC4/RC16、SPANINC=RC0/RC4/RC16、TRUNC=RC0/RC4/RC16、VERIFY、VLSHRT、および ZDPRINT。

DFSORT は、OPTION ステートメントのオプションとして、EXITCK=STRONG/WEAK、WRKREL、および WRKSEC を受け入れ、処理するようになりました。

DFSORT は、SORT および MERGE ステートメントのオプションとして、Y2PAST=s/f を受け入れ、処理するようになりました。

DFSORT は必要に応じて、DFSORT オプションの多数の別名を、受け入れて処理するか、無視するようになりました。

DFSORT は、24 ビットのパラメーター・リスト制御ステートメントの入力コードとして、X'10'、X'16'、X'18'、および X'20' ~ X'29' を受け入れ、処理するようになりました。

---

## 第 20 版 (1999.3)

### リリース 14 の新しいプログラミング・サポート (1999 年 3 月 PTF)

#### 2000 年対応機能

新規の Y2T、Y2U、Y2V、Y2W、Y2X、および Y2Y 形式は、CH、ZD、および PD の完全日付 (yymmdd、yyddd、yymm、yyq、mmdyy、dddyy、mmyy、および qyy) とそれらの特殊標識 (ゼロと 9 など) を、指定された固定またはスライドの「世紀」ウィンドウに従って処理します。

これらの新規形式を SORT、MERGE、INCLUDE、OMIT、および OUTFIL ステートメントで使用して、完全日付を分類、組み合わせ、比較、および変換できます。

---

## 第 19 版 (1998.9)

### 新しいプログラミング・サポート (リリース 14)

#### フィールドと定数のためのシンボル

DFSORT では、DFSORT と ICETOOL ステートメントにシンボルを使用する簡単で柔軟な方法を提供します。DFSORT 制御ステートメントまたは ICETOOL 演算子で認識される、どんなフィールドまたは定数にも、シンボルを定義して使用できます。このシンボルを使用すると 頻繁に使用するデータ などに、シンボルの集合を作成して、繰り返し使用 (マッピング) できます。

また、他の製品 (たとえば、RACF、DFSMSrmm、DCOLLECT など) に関連するデータ専用、あるいは顧客先で独自に作成された、DFSORT シンボルを、集合で入手または使用できます。

DFSORT または ICETOOL を使って特定レコードを処理する場合、DFSORT シンボルを使用すると、関連するフィールドの位置、長さ、形式や、リテラル、数字、およびビット・フラグなどの定数が自動的に指定され、生産性をあげることができます。

#### パフォーマンス、容量および記憶域使用上の改善

ブロック・セットのコピー・アプリケーションと組み合わせアプリケーションに、16 メガバイトより上の仮想記憶域を使用できるようになり、パフォーマンスが改善され、仮想記憶域の制約が緩和されました。

ブロック・セットのコピー・モジュールと組み合わせモジュールが、16 メガバイトより上の仮想記憶域に入るようになり、仮想記憶域の制約が軽減されます。

DFSORT は、従来より大量の INCLUDE および OMIT 条件を扱えるようになりました。

DFSORT は、従来より大量の SUM フィールドを扱えるようになりました。

DFSORT のブロック・セット手法で指定して使用できる、JCL の数と動的に割り振られる作業データ・セット数の上限が、100 から 255 に増えました。使用する作業データ・セットを増やすと、1 回の分類アプリケーションで DFSORT が処理できるデータの最大量を増やすことができます。SORTWKdd または SORTWKd などの形式の有効なすべての DD 名が、DASD 作業データ・セットに使用できるようになりました (たとえば、SORTWK01、SORTWK3、SORTWK2、SORTWK#5、SORTWKA、SORTWKXY など)。

ブロック・セット組み合わせアプリケーションで指定して使用できる入力データ・セットの数の上限が、16 から 100 に増えました。使用する組み合わせ入力データ・セットを増やすと、1 つの組み合わせ処理で DFSORT が扱うデータの最大量を増やすことができます。

### 時刻 (TOD) オプション制御

新しい時刻インストール・モジュール (ICETD1-4) では、DFSORT アプリケーションを実行する日時に基づいて、インストール・システムに複数の異なるデフォルトを設定できるようになりました。各環境でのインストール・モジュール (ICEAM1-4) ごとに、1 つ以上の時刻インストール・モジュールが使用できるようになりました。この結果、インストール先のデフォルトに対して新しいレベルのコントロールができるようになりました。たとえば、週の勤務時間外や週末にバッチ・プログラムが起動する DFSORT アプリケーションに対してのみ、記憶域、ハイパー分類、データ分類の限界をより大きくすることができます。

### パッケージの組み直し

このプロダクトは、再パッケージされ、インストールとカスタマイズが簡単になりました。

- DFSORT、DFSMSdfp、MVS/DFP のチームが、IEBGENER の ICEGENER への置き換えプロセスを単純化しました。これにより、DFSMS または DFP PTF を適用するだけで、IEBGENER に "IEBGENR" の別名が指定され、システムのプログラム検索順序には、"IEBGENER" の別名をもつ ICEGENER が、IEBGENER の前に置かれるようになりました。
- FMID の数が、10 から 3 に減りました。
- DFSORT のインストールに必要なライブラリーの数が、40 から 26 に減りました。
- DFSORT リリース 14 は、常駐および非常駐のどちらの機能も、プロダクトの 1 つのインストールでサポートできるようになりました。この機能のため、インストールの方式にかかわらず、DFSORT の使用方法を指定できるようになり、その結果、インストール時に決定すべき事項が削減されました。
- DFSORT リリース 14 では、英語と日本語のメッセージとパネルの両方の FMID を含む FMID をすべて同時にインストールできます。

### OUTFIL 処理機能の強化

OUTFIL は、入力レコードのフィールドを使用して、複数の出力レコードを作成できるようになりました。この機能を使用すると、1 つのレコードを細かく分解したり、1 つのフィールドを複数のレコードに組み込んだり、異なる複数のフィールドをさまざまなレコードに組み込むことなどができます。

OUTFIL は、OUTFIL OUTREC 指定フィールドのすべてを含むには短すぎる、可変長入力レコードが処理できるようになりました。OUTFIL の新しいオペランド VLFILL=byte を使用すると、OUTFIL OUTREC フィールドの失われたバイトが指定した充てんバイトで置き換えられ、充てんフィールドの処理ができるようになります。

## ICETOOL 機能の強化

ICETOOL の SELECT 演算子の、新しい DISCARD(savedd) オペランドを使用すると、選択しなくても savedd データ・セットにレコードを保管することができるようになりました。したがって、1 回の入力だけで、指定した基準に合うレコードが入った outdd データ・セットと、それ以外の基準に合わないレコードが入った savedd データ・セットを作成できます。DISCARD(savedd) を使用すると、ALLDUPS、NODUPS、HIGHER(x)、LOWER(y)、EQUAL(v)、FIRST または LAST が廃棄するレコードを、保管しておくことができます。

## インストール・オプションと実行時オプションの機能強化

インストール・オプションの EXPRES、EXPOLD、EXPMAX に新しく p% 値が使用できるようになり、インストールおよび実行時オプションの HIPRMAX が使用できるようになりました。p% を使用すると、実行時のシステム上の構成済み拡張記憶域のパーセントとして、DFSORT が対応するオプションのために計算する限界を修正できます。システム上の構成済み拡張記憶域を変更すると、p% が動作して、対応するオプションのために計算される実行時限界も、それに応じて変更されます。シスプレックスのように、システム間で DFSORT インストール・オプションを共用する場合、p% を使用すると、アプリケーションを実行するシステムに対して、DFSORT が対応するオプションのために計算する限界を調整できます。

新しい SPANINC インストールおよび実行時オプションでは、不完全なスパン・レコードを検出した時、DFSORT が処理する方法を指定しておくことができます。この機能により、不完全なスパン・レコードに対する、処置 (不完全なスパン・レコードを削除して、レコードを有効なものにリカバリーして処理を継続するか、または終了するか)、メッセージのタイプ (通知のみ、またはエラー・メッセージ)、戻りコード (0、4 または 16) を制御できます。

新しい OVFLO インストール・オプションおよび実行時オプションを使用すると、BI、FI、PD または ZD の合計フィールドがオーバーフローを起こしたとき、DFSORT の処理方法を指定しておくことができます。この機能により、合計オーバーフローに対する、処置 (継続または終了)、メッセージのタイプ (通知のみ、またはエラー・メッセージ)、戻りコード (0、4 または 16) を制御できます。

新しい PAD インストール指定と実行時オプションを使用すると、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きい場合に、DFSORT が行う処理を指定しておくことができます。このオプションにより、LRECL 埋め込みに対する、処置 (継続または終了)、メッセージのタイプ (通知のみ、またはエラー・メッセージ)、戻りコード (0、4 または 16) を制御できます。

新しい TRUNC インストール・オプションおよび実行時オプションを使用すると、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さい場合に、DFSORT が行う処理を指定しておくことができます。このオプションにより、LRECL 切り捨て

に対する、処置 (継続または終了)、メッセージのタイプ (通知のみ、またはエラー・メッセージ)、戻りコード (0、4 または 16) を制御できます。

ICEMAC オプションの DSA に対する IBM 提供のデフォルトが、16 メガバイトから 32 メガバイトに変更されました。

ICEMAC オプションの GENER に対する IBM 提供のデフォルトが、IEBGENER から IEBGENR に変更されました。

ICEMAC オプションの OVERRGN の最大値が、64 キロバイトから 16128 キロバイトに変更されました。

### **その他の機能強化**

新しいメッセージ ICE178I と ICE179A が、VIO 作業データ・セットの再割り当てに関する情報を提供します。

有効オプションのメッセージ (ICE127I-ICE133I) が、ブロック・セットのコピー・アプリケーションおよび組み合わせアプリケーションで、印刷されるようになりました。

ユーザー出口アドレス定数が、ブロック・セット組み合わせアプリケーションで、E32 ユーザー出口に渡されるようになりました。

可変スパン入力レコードのヌル・セグメントは、DFSORT が処理するようになり、ヌル・セグメントが原因で終了することはなくなりました。ヌル・セグメントとは、ブロックに 1 つもセグメントがないことを意味します。

### **OS/390 および MVS/ESA のみの適用**

DFSORT リリース 14 は、OS/390 と MVS/ESA の環境だけをサポートします。MVS/XA および MVS/XA の VIRTDSP 処理は、サポートしていません。



---

## 第 1 章 DFSORT の紹介

DFSORT の概要 . . . . .	1
Web 上の DFSORT . . . . .	4
DFSORT の FTP サイト . . . . .	4
DFSORT 呼び出し . . . . .	4
DFSORT の機能のしくみ . . . . .	5
オペレーティング・システム . . . . .	5
制御フィールドおよび照合順序 . . . . .	6
文化的環境についての考慮事項 . . . . .	7
DFSORT 処理 . . . . .	8
入力データ・セット - SORTIN および SORTINnn . . . . .	12
出力データ・セット - SORTOUT および OUTFIL . . . . .	12
データ・セットに関する考慮事項 . . . . .	13
レコードの分類またはコピー . . . . .	13
レコードを組み合わせる . . . . .	13
データ・セットの注意事項および制約事項 . . . . .	14
一般的な考慮事項 . . . . .	14
埋め込みと切り捨て . . . . .	15
QSAM に関する考慮事項 . . . . .	16
VSAM に関する考慮事項 . . . . .	16
SmartBatch パイプに関する考慮事項 . . . . .	18
HFS ファイルの考慮事項 . . . . .	18
インストール・デフォルト . . . . .	18
他の分類用プロダクトから DFSORT への移行 . . . . .	25
DFSORT メッセージおよび戻りコード . . . . .	26
可能な限りブロック・セットを使用する . . . . .	27

---

## DFSORT の概要

本章では、IBM DFSORT ライセンス・プログラム、5740-SM1 について紹介します。

DFSORT は、問題プログラム状態でユーザー・キー (キー 8 またはそれ以上のキー) で実行することを目的としています。

DFSORT は、情報の分類、組み合わせ、およびコピーを行うために使用するプログラムです。

- レコードを 分類 すると、元の順序よりもユーザーにより使いやすい、選択した順序に、レコードを並び変えることができます。
- レコードを 組み合わせる ときは、あらかじめ分類されているデータ・セットの 2 つまたはそれ以上の内容を結合して 1 つにします。
- レコードを コピーする ときは、データ・セットの中のレコードをそれぞれ正確に複製します。

レコードを組み合わせるときは、入力データ・セットを、組み合わせるために使用するそれぞれの情報を同じ方法で分類し、出力として必要な同じ順序に並べておく必要があります。1 度に 100 までのデータ・セットを組み合わせることができます。

## DFSORT の概要

上記の 3 つの基本機能のほかに、以下のような処理も同時に実行できます。

アプリケーションで INCLUDE および OMIT ステートメントを使用すると、DFSORT の最終出力データ・セットにどのレコードを保存するかを制御できます。これらのステートメントはフィルターの役目を果たし、レコードをユーザーが提供する基準と照合してテストし、出力データ・セットに保存するレコードだけを残します。たとえば、事務所の所在地の予約フィールドに “Kuala Lumpur” の値をもつレコードだけを処理するように指定できます。あるいは、1987 年以降の日付をもち、かつ従業員が 21 人以上の値をもつすべてのレコードを取り除くように指定することもできます。

INREC および OUTREC ステートメントを使用すると、他の処理を行う前または後で、レコードを編集および再フォーマット設定することができます。INREC および OUTREC ステートメントは、以下のように広範囲にわたるタスクをサポートします。

- 再フォーマット設定されたレコードの入力フィールドの前、入力フィールド間、および入力フィールドの後に、ブランク、ゼロ、ストリング、現在日付、現在時刻およびシーケンス番号を挿入する。
- 16 進数表示、EBCDIC 文字の小文字から大文字へ、または大文字から小文字への変換、ALTSEQ 変換テーブルを使用した文字変換、ある形式から別の形式への数値変換などの、高度な変換機能。
- 長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、数値フィールドの表示方法を制御するなどの、高度な編集機能。  
一般に使用されている数値編集パターンとして 27 個の事前定義編集マスクが用意されており、世界中で使用されている数値表記の大部分を網羅している。さらに、ユーザー定義の編集マスクにより、實際上無制限の数の数値編集パターンが利用できる。
- SMF 日付値および時間値の、より有効なフォームへの変換。
- 指定された固定またはスライドの「世紀」ウィンドウを使用した、2 桁年の日付の各種形式から 4 桁年の日付の各種形式への変換。
- 入力としての文字、16 進数、またはビット・ストリングに基づいて、参照テーブルから文字または 16 進数ストリングを選択する (つまり、参照と変更)。

SUM ステートメントを使用すると、多数のレコードから数値情報を合計して、1 つのレコードに入れることができます。たとえば、1 年間の給与合計額を知りたい場合、全従業員のレコードから給与が入っているフィールドの値のみ合計できます。

OUTFIL 制御ステートメントを使用すると、1 つ以上の入力データ・セットに対して 1 回の入力だけで、分類、コピー、組み合わせアプリケーション用に 1 つ以上の出力データ・セットを作成できます。複数の OUTFIL ステートメントを使用して、その各ステートメントで 1 つ以上の出力データ・セットに対して OUTFIL 処理を実行するように指定できます。他のすべての処理が終了した後 (つまり、出口、オプション、およびその他の制御ステートメントを処理した後)、OUTFIL 処理が開始されます。OUTFIL ステートメントは、次のような様々な出力データ・セットのタスクをサポートします。



- 1 つ以上の入力データ・セットに対する単一のパスを使用することにより、未編集のレコードまたは編集済みのレコードを含む複数の出力データ・セットを作成する。
- 1 つ以上の入力データ・セットに対する単一のパスを使用して、異なる範囲またはサブセットのレコードを含む複数の出力データ・セットを作成する。さらに、どのサブセットとしても選択されないレコードは、別の出力データ・セットに保管できる。
- 可変長レコード・データ・セットから固定長レコード・データ・セットへ変換する。
- 固定長レコード・データ・セットから可変長レコード・データ・セットへ変換する。
- 再フォーマット設定されたレコードの入力フィールドの前、入力フィールド間、および入力フィールドの後に、ブランク、ゼロ、ストリング、現在日付、現在時刻およびシーケンス番号を挿入する。
- 16 進数表示、EBCDIC 文字の小文字から大文字へ、または大文字から小文字への変換、ALTSEQ 変換テーブルを使用した文字変換、ある形式から別の形式への数値変換などの、高度な変換機能。
- 長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、数値フィールドの表示方法を制御するなどの、高度な編集機能。  
 一般に使用されている数値編集パターンとして 27 個の事前定義編集マスクが用意されており、世界中で使用されている数値表記の大部分を網羅している。さらに、ユーザー定義の編集マスクにより、実際上無制限の数の数値編集パターンが利用できる。
- SMF 日付値および時間値の、より有効なフォームへの変換。
- 指定された固定またはスライドの「世紀」ウィンドウを使用した、2 桁年の日付の各種形式から 4 桁年の日付の各種形式への変換。
- 入力としての文字、16 進数、またはビット・ストリングに基づく、参照テーブルからの出力用の文字または 16 進数ストリングの選択 (つまり、参照と変更)。
- 詳細な 3 つのレベル (報告書、ページ、およびセクション) の報告書を提供する。報告書には、ユーザーが指定できる多くの報告書要素 (たとえば、現在の日付、現在の時間、ページ番号、文字ストリング、およびブランク行) や、入力レコードから取り出すことができる多くの報告書要素 (たとえば、文字フィールド、編集された数値入力フィールド、レコード数、および数値入力フィールドの編集済みの合計、最大、最小、および平均) が含まれている。
- 各入力レコードから、ブランクの出力レコードを間に挿入するか、あるいは挿入せずに、複数の出力レコードを作成する。

他の制御ステートメントを用いて、代替照合順序の指定、ユーザー出口ルーチンの呼び出し、インストール・デフォルトの指定変更、などを行うことにより、**DFSORT 機能を制御できます。**

ユーザーは、設計および作成したルーチンに対し、**DFSORT** により、**実行時に制御を渡すことができます。** たとえば、ユーザー出口ルーチンを作成して、処理中にレコードを集計、挿入、削除、短縮、もしくは他の形に変更できます。ただし、INCLUDE、OMIT、INREC、OUTREC、SUM、OUTFIL ステートメントが提供する

## DFSORT の概要

広範な編集機能により、ユーザー出力ルーチンを作成する必要がなくなったことに留意してください。ユーザーは、自分のルーチンを作成して、DFSORT が処理できない入出力エラーを訂正したり、DFSORT が終了する前に必要となるタスクの異常終了操作を実行できます。

**EFS (拡張機能サポート) プログラムを作成すると、DFSORT が使用する前に、DFSORT 制御ステートメントと PARM オプションを代行受信して、変更したり、ユーザー定義のデータに代替順序サポートを提供できます。**

DFSORT 制御ステートメントまたは ICETOOL 演算子で認識される、どんなフィールドまたは定数にも、**シンボルを定義して使用できます**。シンボルを使用すると、さまざまなレコード・レイアウトに関連した情報を表すシンボルの集合を作成して、繰り返し使用 (つまり、マッピング) できます。555 ページの『第 7 章 フィールドと定数のシンボルの使用』を参照してください。

---

## Web 上の DFSORT

記事、オンライン・ブック、ニュース、ヒント、手法、使用例、その他については、下記の URL の DFSORT/MVS ホーム・ページをご利用ください (英語版のみ)。

<http://www.ibm.com/storage/dfsort/>

---

## DFSORT の FTP サイト

以下の アドレスに anonymous FTP で入ると、DFSORT の記事や使用例を入手できます (英語版のみ)。

<ftp.software.ibm.com/storage/dfsort/mvs/>

---

## DFSORT 呼び出し

DFSORT 処理は、以下の方法で呼び出すことができます。

- プログラム (たとえば、PGM=ICEMAN または PGM=SORT) の名前またはカタログ式プロシージャ (たとえば、SORTD) の名前を用いて、入力ストリームで EXEC ジョブ制御ステートメントを使用する方法。29 ページの『第 2 章 ジョブ制御言語による DFSORT の呼び出し』を参照。

TSO ユーザーは、必要な DD 名 (たとえば、SYSOUT、SORTIN、SORTOUT、および SYSIN) を割り振ることができ、また PGM=ICEMAN に相当する呼び出し方式を使用して DFSORT を呼び出すことができる。例：

```
call *(iceman)
```

TSO ユーザーは、以下を使用する DSORT を呼び出せないことに注意が必要。

```
iceman
```

DFSORT を REXX および CLIST から呼び出す例については、645 ページの『第 10 章 DFSORT ジョブ・ストリームの例』を参照。

- システム・マクロ命令を用いて、基本アセンブラー言語で作成されたプログラムを使用する方法。421 ページの『第 5 章 プログラムからの DFSORT の呼び出し』を参照。

- COBOL または PL/I のいずれかで作成されたプログラムをその言語の特殊機能により使用する方法。導入先で使用可能なコンパイラ・バージョンについて説明しているプログラマーの手引きを参照。
- ICETOOL ユーティリティーを使用する方法。441 ページの『第 6 章 ICETOOL の使用』を参照。
- ISPF および ISMF でサポートされている対話式パネルを使用する方法。詳細については、DFSORT パネルの手引きを参照。

**注:** DFSORT のパネルは対話式になっており、DFSORT で使用可能なサブセット機能をパネルでサポートしています。DFSORT の 2000 年対応機能や OUTFIL ステートメントの機能のための対話式パネルは含まれていませんが、Free Form Entry (フリー・フォームの入力) パネルを使用すると、これらの機能の DFSORT 制御ステートメントを指定できます。

本書では、直接呼び出された という用語は、DFSORT が別のプログラムから開始されたのではないことを意味します。プログラム呼び出しの という用語は、DFSORT が別のプログラムから開始されたことを意味します。

---

## DFSORT の機能のしくみ

この節では、DFSORT でサポートされるオペレーティング・システムをリストし、データ・セット・レコードの分類、組み合わせ、およびコピー処理を行なうために、DFSORT が制御フィールドと照合順序をどのように使用するかについて説明します。

ブロック・セット手法は、データ・セットの分類、組み合わせ、およびコピーを行うための、最も効率的な DFSORT の手法です。DFSORT は、可能な限りブロック・セット手法を使用して、その高度な最適化内部アルゴリズムを利用し、IBM ハードウェアを効率的に活用します。ブロック・セットが使用できない場合は、DFSORT は別の手法、すなわちピアレッジ / ベール手法または従来の手法を使用します。

## オペレーティング・システム

DFSORT はオペレーティング・システムの制御下で実行されるため、適切な規則に従って開始する必要があります。このリリースでサポートされるオペレーティング・システムは、次のとおりです。

- z/OS
- OS/390

さらに、DFSORT は z/VM™ または VM/ESA でゲストとして稼動している場合、z/OS または OS/390 で稼動します。

DFSORT は、z/OS または OS/390 がサポートするすべての IBM プロセッサと互換性があります。さらに、プログラムに含まれており、これらのオペレーティング・システム (OS) がサポートする装置すべてに加え、DFSORT は、QSAM または VSAM が入力または出力で使用するすべての装置でも作動します。

### 制御フィールドおよび照合順序

DFSORT で分類または組み合わせを行う情報は、*制御フィールド* を定義して、識別します。データ・セットの内容について考慮する場合は、名前、日付、顧客番号、その他類似した有用な情報を使用します。たとえば、データ・セットを分類するときに、レコードを氏名別、英字順に並べ替えるとします。この場合、各レコードの、氏名の入っている部分の *バイト位置* と *長さ* (バイト単位) を使用して、その部分を DFSORT で処理する制御フィールドとして定義できます。

DFSORT は、処理中に、ユーザーが定義した制御フィールドをキーとして使用します。キー とは、家族名のような 1 つの概念であり、特定のアプリケーションについてレコードの処理方針を設計するときに想定するものです。一方、制御フィールドとは、DFSORT がレコードを識別して分類または組み合わせを行うための形式に対応するテキストやシンボルを含んでいる、レコードの個別部分を指します。事実上、キーは、DFSORT が処理中に使用する制御フィールドと同じものと考えることができます。

レコードを特定の順序に配置する場合は、レコードの 1 つ以上の制御フィールドをキーとして識別します。制御フィールドをリストする順序は、DFSORT がレコードの配置に使用する優先順位の順序になります。指定する最初の制御フィールドを、*大制御フィールド* と呼びます。それ以降の制御フィールドは、*小制御フィールド* と呼び、順に第 1 小制御フィールド、第 2 小制御フィールド、第 3 小制御フィールドと呼びます。

2 つ以上のレコードが最初の制御フィールドで同じ値をもっている場合は、2 番目のフィールドの値に従って配列されます。第 1 と第 2 のフィールドで同じ値をもつレコードは、3 番目のフィールドの値に従って配列するというようにして、フィールド間に相違が見つかるか、あるいは制御フィールドがなくなるまで、その値に従って配列を続けます。

指定されたすべての制御フィールドで同じ値をもっているレコードは、2 つのオプション EQUALS または NOEQUALS のどちらが有効になっているかにより、元の入力順序のまま維持されるか、あるいはランダムに配列されます。EQUALS を指定すると、すべての制御フィールドで同じ値をもつレコードを元の入力順序で維持するように、DFSORT に指示できます。

制御フィールドは互いに重なり合うことも、(たとえば、10 桁の電話番号の中の 3 桁の地域コードのように) 他の制御フィールド内に含まれることもできます。制御フィールドは連続している必要はありませんが、レコードの最初の 32752 バイト内に存在する必要があります (7 ページの図 1 を参照)。

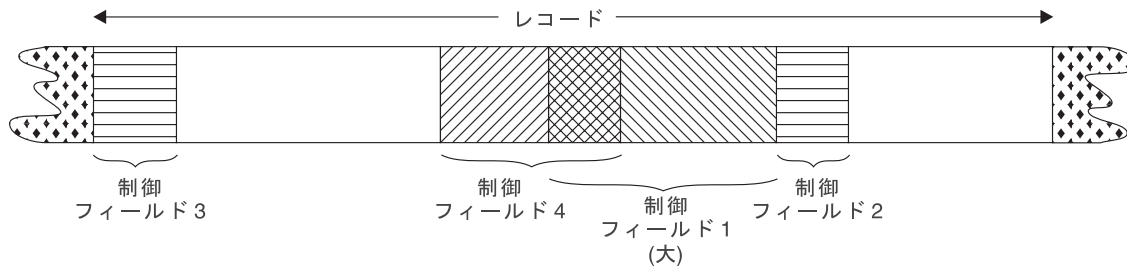


図1. 制御フィールド：制御フィールドは互いに重なり合うことも、他の制御フィールド内に含まれることもできます

DFSORT には、いくつかの標準照合順序があります。レコードを配置する場合は、これらの標準照合順序か、あるいは活動ロケールで定義された照合順序のいずれかを選択できます。概念的には、照合順序とは、異なる 2 つのレコードの同じ制御フィールドにある 2 つの値のうち、どちらを先に配置するか決定するために使用する文字優先順位の特定の配置を指します。DFSORT は、レコードを分類または組み合わせる場合、EBCDIC (IBM の標準照合順序) または ISCI/ASCII 照合順序を使用します。ロケール処理が有効になっている場合は、DFSORT は、活動ロケールで定義されている照合順序を使用します。

文字データと 2 進データの照合順序には絶対値が使用され、文字フィールドと 2 進フィールドは符号がないものと解釈されます。パック 10 進数、ゾーン 10 進数、固定小数点、正規化浮動小数点、および符号付き数値データ形式の場合は、照合には代数値が使用され、それぞれの数量は代数符号をもっているものと解釈されます。

たとえば、国別文字のアルファベット照合順序を使用する場合は、標準の EBCDIC 順序を別の照合順序に変更することもできます。代替照合順序は、インストール時に ICEMAC ALTSEQ オプションを用いて定義でき、また実行時に ALTSEQ プログラム制御ステートメントを用いてユーザーが定義することもできます。さらに、E61 ユーザー出口または EFS プログラムを使用し、照合順序を変更して、指定することもできます。

LOCALE インストール・オプションまたは実行時オプションを指定して、活動ロケールの照合規則を使用できます。

## 文化的環境についての考慮事項

DFSORT の照合の動作を、使用する文化環境に合わせて修正できます。ユーザーの文化環境は、X/Open\*\* ロケール・モデルを使用して DFSORT に定義されます。ロケールとは、ユーザーの文化環境に関する情報を記述する、カテゴリごとにグループ化されたデータの集まりです。

ロケールの照合カテゴリは、照合要素 (単一文字照合要素と複数文字照合要素) 間の相対順位を決定する順序宣言の集まりです。順序宣言は、照合規則を定義します。

文化環境は、活動ロケールを選択することにより設定されます。活動ロケールは、ロケールにより変化する機能の実行に影響を与えます。特に、活動ロケールの照合規則は、DFSORT の分類、組み合わせ、組み込み、および除外処理に、次のような影響を与えます。

## 文化的環境についての考慮事項

- 分類および組み合わせ

DFSORT は、活動ロケールで定義された照合規則に従って、分類または組み合わせレコードを出力用に作成します。これにより、言語の文化特性と地域特性を保持する定義済みの照合規則に基づいた、単一バイトまたは複数バイトの文字データの分類と組み合わせが行われます。

- 組み込み / 除外

DFSORT は、活動ロケールで定義された照合規則に従って、出力用レコードを組み込み、または除外します。これにより、言語の文化特性と地域特性を保持する定義済みの照合規則に基づいた、単一バイトまたは複数バイトの文字データの組み合わせと除外が行われます。

DFSORT オプションである LOCALE は、ロケール処理の使用の有無を指定し、さらに使用する場合は活動ロケールを設定します。一度に 1 つのロケールしか活動状態にできません。

## DFSORT 処理

DFSORT パネルを使用してジョブの準備と実行要求を行う場合 (*DFSORT* パネルの手引き を参照) 以外は、ユーザーは、ジョブ制御言語 (JCL) と DFSORT プログラム制御ステートメントを使用して、DFSORT 処理を呼び出す必要があります。JCL ステートメント (421 ページの『第 5 章 プログラムからの DFSORT の呼び出し』を参照) は、オペレーティング・システムにより処理されます。JCL ステートメントは、ユーザーのデータ・セットをオペレーティング・システムに記述し、DFSORT 処理を開始します。DFSORT プログラム制御ステートメント (111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』を参照) は、DFSORT プログラムで処理されます。このステートメントは、ユーザーが実行する機能を記述し、要求する処理を呼び出します。

分類処理には、通常、プログラム実行中のワークスペースとして中間記憶域が必要です。この記憶域は以下のいずれかになります。

1. ハイパー空間。DFSORT のハイパー分類機能を使用します。
2. 作業データ・セット。このデータ・セットは DFSORT の DYNALLOC 機能で割り当てるか、あるいはユーザーが JCL DD ステートメントを使って指定します。ユーザーが指定する場合は、中間記憶装置とワークスペースの大きさを指定します。割り当てたワークスペースの大きさを計算する方法は、685 ページの『付録 A. ワークスペースの使用』で説明します。
3. ハイパー空間と作業データ・セットの組み合わせ。

組み合わせアプリケーションとコピー・アプリケーションには、中間記憶装置は必要ありません。

9 ページの図 2 は、レコード処理、出口、ステートメント、およびオプションの処理順序を示しています。このダイアグラムとあわせて説明を参照し、DFSORT がユーザーのジョブを実行する順序を理解してください。

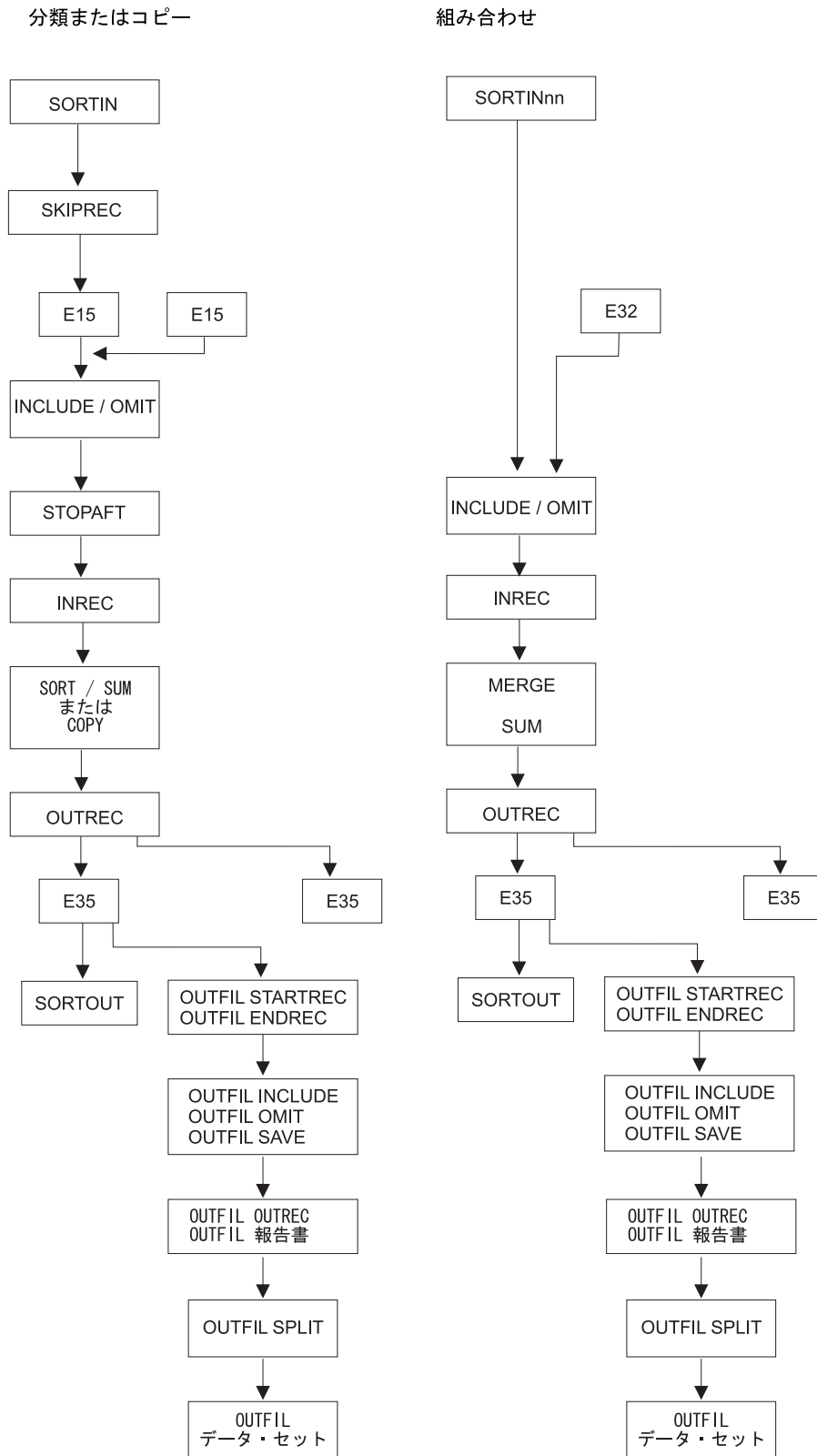


図 2. レコード処理の順序

図 2 に示されているように、DFSORT は次の順序で処理します。

## DFSORT 処理

1. DFSORT は、最初に SORTIN データ・セット (SORT および COPY ジョブの場合) または SORTINnn データ・セット (MERGE の場合) のいずれかが提供されたかを確認します。提供されている場合、DFSORT はそこから入力レコードを読み込みます。
  - SORT または COPY ジョブの SORTIN データ・セットがない場合、E15 ユーザー出口を使用して、すべてのレコードを挿入する必要があります。(これは、パラメーター・リストに E15 ユーザー出口のアドレスをもつプログラムから DFSORT を呼び出す場合も同様ですが、これは SORTIN が無視されるためです。) E15 ユーザー出口を MODS ステートメントで指定すると、DFSORT は COBOL E15 ルーチンを使用できます。
  - 組み合わせジョブで SORTINnn データ・セットがない場合は、E32 ユーザー出口を使用してすべてのレコードを挿入する必要があります。
2. SORT または COPY ジョブの入力レコードが SORTIN データ・セットから読み込まれた場合、DFSORT は SKIPREC オプションで指定された処理を実行します。DFSORT は、SKIPREC カウントが満足されるまでレコードを削除します。SORT または COPY を行う前にレコードを削除するので、パフォーマンスが向上します。
3. SORT または COPY ジョブの入力レコードが SORTIN データ・セットから読み込まれた場合、DFSORT は E15 ユーザー出口が指定されているかどうかを確認します。指定されている場合、DFSORT はユーザー出口ルーチンに制御を渡します。E15 ユーザー出口を MODS ステートメントで指定すると、ユーザーは COBOL E15 ルーチンを使用できます。E15 ルーチンは、レコードの挿入、削除、再フォーマット設定を行なうことができます。
4. DFSORT は、INCLUDE または OMIT ステートメントで指定された処理を行ないます。E15 ユーザー出口ルーチンを使用してレコード形式を修正した場合、ユーザーが指定する INCLUDE/OMIT 制御フィールド定義は、もとの形式ではなく、現行の形式に適用される必要があります。INCLUDE または OMIT ステートメントを使用して不要なレコードを削除してから SORT、MERGE、または COPY 処理を行なうと、ジョブはさらに効率良く実行されます。
5. SORT または COPY ジョブの場合、DFSORT は STOPAFT オプションで指定された処理を実行します。ユーザーが指定した最大数 (n) のレコードを受け取ると、レコードの入力は停止します。DFSORT は、以下の条件に合うレコードを受け入れて、処理します。
  - SORTIN から読み込まれたか、E15 が挿入したレコード
  - SKIPREC が削除しないレコード
  - E15 が削除しないレコード
  - INCLUDE または OMIT ステートメントが削除しないレコード
6. DFSORT は、INREC ステートメントで指定された処理を行ないます。このステップの前にレコード形式を変更する場合、ユーザーが指定する INREC 制御フィールド定義および区切りフィールド定義は、もとの形式ではなく現行の形式を適用する必要があります。
7. DFSORT は、SORT、MERGE、または OPTION COPY ステートメントで指定された処理を行ないます。
  - SORT の場合、すべての入力レコードが処理された後で、出力レコードが処理されます。



- COPY または MERGE の場合、入力レコードが 1 つ処理された後で、出力レコードが 1 つ処理されます。
  - SORT または MERGE の場合で SUM ステートメントがあると、DFSORT は それを SORT または MERGE 処理の間に処理します。DFSORT はレコードを集計して、重複したレコードを削除します。このステップの前にレコード形式になんらかの変更を行った場合、ユーザーが指定する SORT または MERGE フィールド定義および SUM フィールド定義は、もとの形式ではなく現行の形式に適用する必要があります。
8. DFSORT は、OUTREC ステートメントで指定された処理を行いません。このステップの前にレコード形式を変更する場合、ユーザーが指定する OUTREC 制御フィールド定義および区切りフィールド定義は、もとの形式ではなく現行の形式を適用する必要があります。
  9. E35 ユーザー出口ルーチンがある場合、すべてのステートメント処理が完了すると、DFSORT はユーザー出口ルーチンに制御を渡します。レコード形式を変更した場合、E35 ユーザー出口ではもとの形式ではなく現行の形式でレコードが受け取られます。E35 ユーザー出口を MODS ステートメントで指定すると、COBOL E35 ルーチンを使用できます。E35 出口ルーチンを使用して、レコードの追加、削除、または再フォーマット設定を行うことができます。  
SORTOUT および OUTFIL データ・セットがない場合、DFSORT がこれらのレコードを削除されたものとして扱うため、E35 出口ルーチンがすべてのレコードを処理する必要があります。(これは、OUTFIL データ・セットを指定せずに、パラメーター・リストに E35 ユーザー出口のアドレスで DFSORT を呼び出す場合も同様です。これは SORTOUT が無視されるためです。)
  10. DFSORT は、ユーザーのレコードを SORTOUT データ・セットに書き込みます (データ・セットがある場合)。
  11. DFSORT は、1 つ以上の OUTFIL ステートメントがあった場合、そのステートメントが指定する処理を実行します。
    - DFSORT は、STARTREC および ENDREC またはそのどちらかのステートメントで指定された処理を行いません。OUTFIL データ・セットのレコード入力は、STARTREC で指示されたレコードから始まり、ENDREC で指示されたレコードで終わります。
    - DFSORT は、INCLUDE、OMIT、または SAVE オプションで指定された処理を行いません。レコードは、指定された基準にしたがって、OUTFIL データ・セットから組み込みまたは削除されます。
    - DFSORT は、OUTREC、LINES、HEADER1、TRAILER1、HEADER2、TRAILER2、SECTIONS、および NODETAIL オプションで指定された処理を実行します。データ・レコードは、再フォーマット設定され、OUTFIL データ・セットの報告書レコードが生成されます。
    - DFSORT は、SPLIT 処理を実施します。レコードは、OUTFIL データ・セット間でできるだけ均等に配布されます。
    - DFSORT は、ユーザーの OUTFIL レコードを、適切な OUTFIL データ・セットに書き込みます。

## 入力データ・セット - SORTIN および SORTINnn

DFSORT は、SORTIN データ・セット (または単に SORTIN) と SORTINnn データ・セット (または単に SORTINnn) と呼ばれる、2 種類の入力データ・セットを処理します。

SORTIN DD ステートメントは、分類アプリケーションまたはコピー・アプリケーションのための入力データ・セット (または連結入力データ・セット) を指定します。SORTIN DD ステートメントがある場合、パラメーター・リストの E15 ユーザー出口のアドレスをもつプログラムから DFSORT が呼び出されない限り、このステートメントが分類またはコピー処理のデフォルトとして使用されます。

SORTINnn DD ステートメント (ここで nn は 00 ~ 99) は、組み合わせ処理のデータ・セットを指定します。SORTINnn DD ステートメントがある場合、パラメーター・リストの E32 ユーザー出口のアドレスをもつプログラムから DFSORT が呼び出されない限り、このステートメントが組み合わせアプリケーションのデフォルトとして使用されます。

13 ページの『データ・セットに関する考慮事項』では、入力データ・セットに関する一般的な情報について説明しています。SORTIN データ・セットに関する特定情報については、75 ページの『SORTIN DD ステートメント』を参照してください。SORTINnn データ・セットに関する特定情報については、77 ページの『SORTINnn DD ステートメント』を参照してください。

---

## 出力データ・セット - SORTOUT および OUTFIL

DFSORT は、SORTOUT データ・セット (または単に SORTOUT) と OUTFIL データ・セットと呼ばれる、2 種類の実出力データ・セットを処理します。

SORTOUT DD ステートメントは、分類、コピー、または組み合わせアプリケーションのための OUTFIL 以外の単一出力データ・セットを指定します。OUTFIL 処理は、SORTOUT には適用されません。SORTOUT DD ステートメントがある場合、パラメーター・リストの E35 ユーザー出口のアドレスをもつプログラムから DFSORT が呼び出されない限り、このステートメントが分類、コピー、または組み合わせアプリケーションのデフォルトとして使用されます。

1 つ以上の OUTFIL ステートメントの FNAMES または FILES (またはその両方) のパラメーターは、分類、コピー、または組み合わせアプリケーションの OUTFIL データ・セットの DD 名を指定します。各 OUTFIL ステートメントごとに指定されたパラメーターは、そのステートメントに関連する OUTFIL データ・セットについて OUTFIL 処理が実施されるように定義します。指定された各 DD 名には、対応する DD ステートメントが必ず存在する必要があります。

SORTOUT DD 名は、実際は OUTFIL データ・セットに使用することもできますが、『SORTOUT』という用語は単一の OUTFIL 出力以外のデータ・セットを示すのに使用します。

13 ページの『データ・セットに関する考慮事項』では、入力データ・セットに関する一般的な情報について説明しています。SORTOUT データ・セットに関する特定情報については、82 ページの『SORTOUT および OUTFIL DD ステートメント』

を参照してください。OUTFIL データ・セットに関する特定情報については、82 ページの『SORTOUT および OUTFIL DD ステートメント』と 238 ページの『OUTFIL 制御ステートメント』を参照してください。

---

### データ・セットに関する考慮事項

ユーザーは、DFSORT に提供するすべてのデータ・セットを、使用するオペレーティング・システムが必要とする規定に従って定義する必要があります。オペレーティング・システムのラベル検査機能は、DFSORT 処理中に使用できます。詳しくは、*アプリケーション開発の手引き* を参照してください。

DFSORT パネルを使ってジョブを作成または実行要求を行わない限り、すべてのデータ・セット (DYNALLOC パラメーターで割り当てられたものを除く) を DD ステートメントに記述する必要があります。DD ステートメントは、DFSORT 処理の割り当てを行うジョブ・ステップとともに、オペレーティング・システムの入カストリームに含める必要があります。

DFSORT パネルは、前景および背景の 2 つのモードで作動します。前景モードでは、JCL のかわりに CLIST 処理を使用します。そのため、この手法を選択した場合、JCL はまったく必要ありません。背景モードでは、DFSORT パネルのユーザー・プロファイルですでにコーディング済みのジョブ制御言語 (DD ステートメントを含む) が含まれている DFSORT ジョブを作成します。この JCL は、ユーザー自身がコーディングしたのと同じものです。詳細については、*DFSORT パネルの手引き* を参照してください。

### レコードの分類またはコピー

分類またはコピー処理の入力では、固定長または可変長のレコードを含む、ブロック化された、またはブロック化されていない QSAM または VSAM データ・セットを使用できます。QSAM 入力データ・セットは、異なる装置の場合でも連結できます。適用する場合の制約については、75 ページの『SORTIN DD ステートメント』を参照してください。

分類またはコピー処理の出力は、入力が QSAM であるか VSAM であるかにかかわらず、ブロック化または非ブロック化した QSAM または VSAM データ・セットにすることができます。OUTFIL を使用して可変入力を固定出力に、または固定入力を可変出力に変換しない限り、出力データ・セットは入力データ・セットと同じタイプ (固定または可変) である必要があります。

分類アプリケーションとコピー・アプリケーションの入出力として、階層ファイル・システム (HFS) ファイルがサポートされています。

分類アプリケーションとコピー・アプリケーションの入出力として、SmartBatch パイプがサポートされています。

### レコードを組み合わせる

組み合わせアプリケーションへの入力には、固定長または可変長レコードを含む最大 100 個のブロック化または非ブロック化 QSAM または VSAM データ・セットを使用できます。入力データ・セットは、QSAM または VSAM のいずれかになり

## データ・セットに関する考慮事項

ますが、両方を使用できません。入力データ・セットのレコードはすべて、出力に必要な順序と同じ順序であらかじめ分類しておく必要があります。

組み合わせアプリケーションの出力は、入力が QSAM または VSAM のどちらであるかにかかわらず、ブロック化または非ブロック化 QSAM データ・セットまたは VSAM データ・セットを使用できます。OUTFIL を使用して可変入力を固定出力に、または固定出力を可変出力に変換しない限り、出力データ・セットは入力データ・セットと同じタイプ (固定または可変) である必要があります。

組み合わせアプリケーションの入出力として、階層ファイル・システム (HFS) ファイルがサポートされています。

組み合わせアプリケーションの入出力として、SmartBatch パイプがサポートされています。

## データ・セットの注意事項および制約事項

データ・セットについて知っておく必要のある注意事項と制約事項がいくつかあります。以下の項目で、これらについて説明します。

特定の DFSORT データ・セットの詳細については、67 ページの『DD ステートメントの使用』を参照してください。

### 一般的な考慮事項

可変長レコードは、1 ~ 4 桁目のレコード記述子ワード (RDW) から処理されるため、データは 5 桁目から始まります。固定長レコードは RDW なしで処理されるため、データは 1 桁目から始まります。必要に応じて、制御ステートメント位置を指定する必要があります。

ユーザーのレコードは、EBCDIC、ISCI/ASCII、日本語、およびユーザー自身が定義するデータ・タイプにすることができます。DFSORT で日本語のデータ・タイプを処理する場合は、IBM 2 バイト文字セット配列支援プログラム (DBCS 配列)、ライセンス・プログラム 5665-360、リリース 2.0 を使用するか、適切なロケールのロケール処理を使用します。

入力および出力データ・セットは、QSAM または VSAM で使用できる装置上に必ず存在する必要があります。

標準システム・データ管理規則は、すべてのデータ・セット処理に適用されます。特に、入力データ・セットに固定標準レコード形式を使用するときは、最初の短ブロックはボリュームの終わりのように取り扱われることに注意してください。詳細については、*z/OS DFSMS: Using Data Sets*を参照してください。

DFSORT が処理できる最大レコード長には、以下の制約があります。

- レコード長は、ユーザーが指定した最大レコード長を超えてはなりません。
- 可変長レコードは 32756 バイトまでに制限されます。
- VSAM 可変長レコードは 32752 バイトまでに制限されます。
- 可変長レコードは 32760 バイトまでに制限されます。
- 可変ブロック・スパン・レコードは 32767 バイトまでに制限されます。

- テープ作業データ・セットの分類での最大レコード長は、NOEQUALS が有効の場合は 32752 バイトまで、EQUALS が有効の場合は 32748 バイトまでに制限されます。

**注:** AQ 形式を指定している場合、または CH 形式を指定して、かつ CHALT オプションが有効な場合、可変長レコードの最大レコード長は 32767 バイトから制御フィールドの長さを引いたものになります。

指定したサイズの記憶域を使用して分類できるレコードの数は、以下の事項により削減されます。

- 異なる形式の制御フィールドを処理する場合
- 制御フィールドの数が多い場合
- 中間データ・セットの数が多い場合

EFS01 ルーチンを指定した拡張機能サポート・プログラムを使用すると、可変長レコードの処理時に使用できるレコード長が制限されることがあります。

テープ作業データ・セットの最小ブロック長は 18 バイト、最小レコード長は 14 バイトです。

### 埋め込みと切り捨て

SORTOUT LRECL が、TRUNC オプションを指定した SORTIN/SORTINnn LRECL より小さい場合、DFSORT が行う処置を制御できます。これについては、181 ページの『OPTION 制御ステートメント』で説明しています。

SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいと、以下の条件に合う場合、DFSORT は固定長レコードの右側を切り捨てます。

- アプリケーションが、従来の組み合わせまたは、テープ作業データ・セットではない。
- TRUNC=RC16 が活動状態ではない。

181 ページの『OPTION 制御ステートメント』で説明する VLLONG または NOVLLONG オプションを使用して、可変長出力レコードが、書き込まれる SORTOUT または OUTFIL データ・セットの LRECL よりも長い場合に DFSORT に行わせる処置を制御できます。

SORTOUT LRECL が、SORTIN/SORTINnn LRECL より大きい場合に DFSORT が行う処置を、PAD オプションにより制御できます。これについては、181 ページの『OPTION 制御ステートメント』で説明しています。

以下の条件を満たして、SORTOUT LRECL が SORTIN LRECL より大きい場合、DFSORT は固定長レコードの右側を 2 進ゼロで埋め込みます。

- ブロック・セット手法が選択されている
- アプリケーションが分類またはコピーである
- PAD=RC16 が活動状態ではない

DFSORT は、E15 または E35 ユーザー出口から戻ったレコードについては、埋め込みも切り捨ても行いません。これは、ユーザー出口がレコードを適切に埋め込みまたは切り捨てするものと想定しているからです。

## データ・セットに関する考慮事項

INREC、OUTREC、および OUTFIL を使用して、レコードの埋め込み、切り捨て、および再フォーマット設定を行うことができます。詳細は、153 ページの『INREC 制御ステートメント』と 327 ページの『OUTREC 制御ステートメント』を参照してください。

ICEGENER による埋め込みおよび切り捨てについては、639 ページの『IEBGENER に代わる、ICEGENER の使用』を参照してください。

ブロック・セットおよびその他の DFSORT 手法の詳細については、624 ページの『効率的な分類 / 組み合わせ手法の指定』を参照してください。

### QSAM に関する考慮事項

- 入力データ・セット用の DD ステートメントに DSN=NULLFILE を使用した場合、システムの制約により DFSORT は EXCP アクセス方式を使用できません。
- 空の入力データ・セットを使用できます。
- 入力データ・セットが標準ラベルをもたないテープ上にある場合は、その入力データ・セットの DD ステートメントに DCB パラメーターを指定する必要があります。
- ISO/ANSI バージョン 1 のテープ・ファイルは、入力としてだけ使用できます。出力としては使用できません。
- DFSORT は、入力および出力データ・セットに適切な BUFNO 値を設定します。これらのデータ・セットに DD ステートメントで BUFNO を指定しても適用されません。

その他の考慮事項については、75 ページの『SORTIN DD ステートメント』を参照してください。

### VSAM に関する考慮事項

- ユーザーは DFSORT に VSAM レコードを、固定長 (F) または可変長 (V) として処理させることができます。VSAM 入力を使用するとき、DFSORT は、ユーザーが RECORD TYPE=F を指定すれば固定長処理を選択し、RECORD TYPE=V を指定すれば可変長処理を選択します。RECORD TYPE=x が指定されない場合は、DFSORT は、340 ページの『RECORD 制御ステートメント』の TYPE オペランドの解説で説明した「規則」に従って、使用するレコード・タイプを選択します。選択されるレコード・タイプは、レコードの処理法や、制御ステートメントの位置の指定方法に影響します。以下のとおりです。
  - **可変長処理:** RRDS、KSDS、ESDS、または VRRDS は常に可変長として処理できます。VSAM 入力データの場合、DFSORT は、各レコードを読み取り、そのレコード記述子ワード (RDW) を作成します。VSAM 出力の場合は、DFSORT は各レコードを書き込む前に RDW を除去します。DFSORT は 1 ~ 4 桁目の RDW を使用して可変長レコードを処理するので、データは 5 桁目から開始されます。必要に応じて、制御ステートメント位置を指定する必要があります。
  - **固定長処理:** RRDS は常に固定長として処理されます。入力に使用される KSDS、ESDS、または VRRDS は、そのレコードのすべての長さが、クラスタに定義された最大レコード・サイズに等しい場合のみ、固定長として処理する必要があります。そうでなければ、最大レコード・サイズより短い入力レコ

## データ・セットに関する考慮事項

ードは、ゼロのバイト (つまり、「ガーベッジ」バイト) か、ゼロでないバイトで埋め込まれます。DFSORT は、固定長レコードを処理するのに RDW を使用しないため、データは 1 桁目から開始されます。必要に応じて、制御ステートメント位置を指定する必要があります。

- データ・セットがパスワードで保護されている場合、パスワードをコンソールから、または (いくつかの制約はありますが) ユーザー出口 E18、E38 および E39 のルーチンから入力できます。

**注:** OUTFIL データ・セットの場合は、この方法でパスワードを処理できません。

- VSAMIO および RESET が有効であれば、REUSE で定義したデータ・セットを、分類の入出力として使用できます。すなわち、そのデータ・セットをインプレース分類することができます。
- 入力または出力に使用されるデータ・セットは、あらかじめ定義しておく必要があります。
- VSAMEMT が有効な場合、空の入力データ・セットは、ゼロ・レコードをもつとして処理されます。
- VSAM データ・セットを連結できません (システム制限)。
- VSAM 入力データ・セットと 非 VSAM 入力データ・セットを、両方一緒に分類、組み合わせ、コピーなどのアプリケーションに指定できません。
- 出力が VSAM キー順データ・セット (KSDS) の場合、そのキーは最初の制御フィールド (またはキー・フィールドが最初の制御フィールドと同じ順序) である必要があります。VSAM では、重複する 1 次キーをもつレコードを保管できません。
- 入力データ・セットで利用できる VSAM 出口機能は、EODAD を除いてすべて使用できます。VSAM の E18 使用については、361 ページの『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』を参照してください。
- VSAM 出口リストは、VSAM 出口機能を処理するルーチンのアドレスを指定して、VSAM EXLST マクロ命令を使って作成する必要があります。
- VSAM 入力および非 VSAM 出力で可変長レコードを処理する場合、出力 LRECL は、クラスターに定義されている最大レコード・サイズより少なくとも 4 バイト以上大きい必要があります。非 VSAM 可変長レコードには、各レコードの先頭に 4 バイトの長さのレコード記述子ワード (RDW) がありますが、VSAM レコードにはこれがありません。したがって、VSAM クラスターで定義されたレコード・サイズは、非 VSAM LRECL より 4 バイト小さくなります。
- REUSE なしで定義された出力データ・セットは、MOD として処理されます。
- RESET が有効な場合、REUSE で定義された出力データ・セットは NEW として処理されます。NORESET が有効な場合、REUSE で定義された出力データ・セットは MOD として処理されます。
- DFSORT は RLS モードで VSAM データ・セットにアクセスできません。つまり、VSAM の入力データ・セットと出力データ・セットについては、RLS=CR と RLS=NRI はサポートされません。

### SmartBatch パイプに関する考慮事項

SmartBatch パイプ・データ・セットは、入力および出力に使用できますが、ブロック・セット手法でしか使用できません。SmartBatch パイプ・データ・セットを使用する DFSORT アプリケーション用にブロック・セットを選択しなかった場合、DFSORT は、エラー・メッセージを出して、終了します。

DFSORT が、SmartBatch パイプ・データ・セットが入力または出力用に使用されていることを判別すると、DFSORT は、以下を実行します。

- 自動的に ABEND オプションをオンにして、エラーが検出された場合に、ABEND が生成されるようにする。
- E15、E32 または E35 ユーザー出口プログラムが終了を要求する場合は、戻りコード 16 ではなく、ユーザー ABEND ゼロで終了する。

ICETOOL が、SmartBatch パイプ・データ・セットが入力または出力用に使用されていることを判別すると、ICETOOL は、戻りコード 12 ではなく、ユーザー ABEND 2222 で自動的に終了します。

上記のような状態から ABEND が生成されるため、システムはエラーを、DFSORT または ICETOOL と同じ SmartBatch パイプ・データ・セットにアクセスしている可能性のある他のアプリケーションにも、適切に伝えることができます。

DFSORT または ICETOOL がエラーを検出するのが、SmartBatch パイプ・データ・セットが使用されていることを判別する前、または SmartBatch パイプ・データ・セットをオープンする前である場合、適切なエラー伝搬が生じません。これにより、別のアプリケーションを SmartBatch パイプ・データ・セットの永続待機に付加させることがあります。

---

### HFS ファイルの考慮事項

階層ファイル・システム (HFS) ファイルは入力および出力に使用できますが、ブロック・セット手法でしかサポートされません。HFS ファイルを使用する DFSORT アプリケーション用にブロック・セットを選択しなかった場合、DFSORT は、エラー・メッセージを出して、終了します。

HFS ファイルを使用する場合は、*z/OS UNIX システム・サービス ユーザーズ・ガイド* に記載されているそれらのファイルに関する情報に精通しておくことが必要です。DFSORT は、BSAM を使用して HFS ファイルにアクセスするので、それに伴うすべての機能と制約事項 (*z/OS DFSMS: Using Data Sets* に記載されています) に従います。

---

### インストール・デフォルト

DFSORT をインストールするときにシステム・プログラマーは、次の 8 つのインストール・モジュール用にデフォルトで使用する、インストール・パラメーターのセット (ICEMAC) を個別に選択してしています。

#### ICEAM1 (JCL)

バッチ直接呼び出し環境インストール・モジュール。バッチ・ジョブが DFSORT を直接 (プログラムからではなく) 呼び出した場合、実行時に、こ



のデフォルトのセットを使用する。使用する時刻 (TOD) インストール・モジュール (ICETDx) は、活動化されていないものと仮定する。

### ICEAM2 (INV)

バッチ・プログラム呼び出し環境インストール・モジュール。バッチ・プログラムが DFSORT を呼び出した場合、実行時に、このデフォルトのセットを使用する。使用する時刻 (TOD) インストール・モジュール (ICETDx) は、活動化されていないものと仮定する。

### ICEAM3 (TSO)

TSO 直接呼び出し環境モジュール。前景の TSO ユーザーが、DFSORT を直接 (プログラムからではなく) 呼び出した場合、実行時に、このデフォルトのセットを使用する。使用する時刻 (TOD) インストール・モジュール (ICETDx) は、活動化されていないものと仮定する。

### ICEAM4 (TSOINV)

TSO プログラム呼び出し環境インストール・モジュール。前景の TSO ユーザーが、DFSORT をプログラムから呼び出した場合、実行時に、このデフォルトのセットを使用する。使用する時刻 (TOD) インストール・モジュール (ICETDx) は、活動化されていないものと仮定する。

### ICETD1 (TD1)

最初の時刻 (TOD) インストール・モジュール。実行用時刻 (TOD) として活動化されると、このデフォルトのセットが実行時に使用される。有効な環境インストール・モジュール (ICEAMx) が、これを使用可能にしているものとする。

### ICETD2 (TD2)

第二の時刻 (TOD) インストール・モジュール。実行用時刻 (TOD) として活動化されると、このデフォルトのセットが実行時に使用される。有効な環境インストール・モジュール (ICEAMx) が、これを使用可能にしているものとする。

### ICETD3 (TD3)

第三の時刻 (TOD) インストール・モジュール。実行用時刻 (TOD) として活動化されると、このデフォルトのセットが実行時に使用される。有効な環境インストール・モジュール (ICEAMx) が、これを使用可能にしているものとする。

### ICETD4 (TD4)

第四の時刻 (TOD) インストール・モジュール。実行用時刻 (TOD) として活動化されると、このデフォルトのセットが実行時に使用される。有効な環境インストール・モジュール (ICEAMx) が、これを使用可能にしているものとする。

選択したデフォルトは、アプリケーションの実行方法に影響することがあり、多くの場合、実行時に適切なパラメーターを指定して変更できます (指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください)。本書では、DFSORT は製品引き渡し時のデフォルトのままユーザーのシステムにインストールされたものと仮定しています。

次のような ICETOOL ジョブを使用すると、8 つのインストール・モジュールについてユーザー・システムで実際に使用するインストール時のデフォルトと、必要に

## インストール・デフォルト

応じて指定変更される IBM 提供のデフォルトをリストできます。

表 1. ICETOOL を使用して、インストール時のデフォルトをリストする方法

```
//DFRUN JOB A402,PROGRAMMER
//LISTDEF EXEC PGM=ICETOOL,REGION=1024K
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//SHOWDEF DD SYSOUT=A
//TOOLIN DD *
DEFAULTS LIST(SHOWDEF)
/*
```

ICETOOL および DEFAULTS 演算子の使用の詳細については、441 ページの『第 6 章 ICETOOL の使用』と 462 ページの『DEFAULTS 演算子』を参照してください。

使用できる ICEMAC パラメーターの機能を、以下に要約します。DFSORT 導入とカスタマイズ リリース 14 には、使用可能な ICEMAC パラメーターの詳細な説明と、DFSORT インストールの計画に関する考慮事項と一般情報が記載されています。DFSORT プログラム資料説明書には、ステップごとのインストール手順をリストしています。

パラメーター	機能
<b>INVIJCLITSOITSOINVITD1ITD2ITD3ITD4</b>	デフォルトのセットの ICEMAC を使用する、環境インストール・モジュール (ICEAMx)、または 時刻 (TOD) インストール・モジュール (ICETDx) を指定します。
<b>ENABLE</b>	ICEAMx 環境モジュールが活動化された場合に、ICETDx インストール・モジュールを使用するかどうかを指定します。
<b>day</b>	この ICETDx インストール・モジュールを活動化する場合、週の毎日の時間帯を指定します。
<b>ABCODE</b>	クリティカル・エラーが原因で DFSORT が異常終了した場合に使用する ABEND コードを指定します。
<b>ALTSEQ</b>	ALTSEQ 変換テーブルへの変更を指定します。
<b>ARESALL</b>	仮想記憶域上部に予約するシステム用のバイト数を指定します。
<b>ARESINV</b>	DFSORT がプログラムで呼び出される場合に、その呼び出しプログラム用に予約する、16 メガバイト境界より上の記憶域のバイト数を指定します。
<b>CFW</b>	作業データ・セットの処理時に、DFSORT がキャッシュ高速書き込みを使用するかどうかを指定します。
<b>CHALT</b>	形式 AQ だけでなく形式 CH も変換するか、または形式 AQ のみを変換します。

<b>CHECK</b>	出力データ・セットのない E35 ユーザー出口ルーチンを使用するアプリケーションに対して、レコード数を検査するか、しないかを指定します。
<b>CINV</b>	DFSORT が VSAM データ・セットに対して制御インターバル・アクセスを使用できるようにするかどうかを指定します。
<b>COBEXIT</b>	COBOL E15 ルーチンと E35 ルーチンにライブラリーを指定します。
<b>DIAGSIM</b>	SORTDIAG DD ステートメントを DFSORT アプリケーション用にシミュレートするかどうかを指定します。
<b>DSA</b>	ブロック・セット分類アプリケーションの動的ストレージ調整のために、DFSORT が使用する記憶域の最大量を指定します。
<b>DSPSIZE</b>	データ空間分類に使用するデータ空間の最大量を指定します。
<b>DYNALOC</b>	装置名と作業データ・セットの数に対して、動的に割り当てられるデフォルトを指定します。これらのデフォルトは、ICEMAC オプションの DYNAUTO と実行時オプション DYNALOC に使用されます。
<b>DYNAUTO</b>	作業データ・セットの動的割り振りを自動的に行うかどうかを指定します。
<b>DYNSPC</b>	ファイルのサイズが不明な場合に動的に割り振られる作業データ・セットへの、デフォルトの 1 次スペース割り振り全体を指定します。
<b>EFS</b>	DFSORT が呼び出すユーザー作成の拡張機能サポート・プログラムの名前を指定します。
<b>EQUALS</b>	照合して一致するレコード順序を、入力から出力まで保持するかどうかを指定します。
<b>ERET</b>	クリティカル・エラーを検出したときに DFSORT が行う処置を指定します。
<b>ESTAE</b>	DFSORT が ESTAE リカバリー・ルーチンを実行の早期段階で削除するか、実行の全期間を通して使用するかを指定します。
<b>EXITCK</b>	E15 または E35 ユーザー出口ルーチンから特定の無効な戻りコードを受信した場合、DFSORT を終了するか、続行するかを指定します。
<b>EXPMAX</b>	すべての MVS/ESA システムのハイパー分類アプリケーションが利用可能な記憶域の最大合計数量を任意の一時点で使用することを指定します。

## インストール・デフォルト

	<b>EXPOLD</b>	すべてのハイパー分類アプリケーションが古い記憶域の最大合計量を任意の一時点で使用することを指定します。
	<b>EXPRES</b>	使用できる記憶域のうち最低の大きさの記憶域を非ハイパー分類アプリケーション用に予約しておくよう指定します。
	<b>FSZEST</b>	実行時オプション、FILSZ=n および SIZE=n を、DFSORT が正確なファイル・サイズまたは見積ファイル・サイズのいずれとして扱うかを指定します。
	<b>GENER</b>	ICEGENER が、制御を IEBGENER システム・ユーティリティに移動する場合に使用する名前を指定します。(ICEGENER は DFSORT の IEBGENER ジョブ用の機能。)
	<b>GNPAD</b>	LRECL の埋め込みのために ICEGENER が行う処置を指定します。
	<b>GNTRUNC</b>	LRECL 切り捨てのために ICEGENER が行う処置を指定します。
	<b>HIPRMAX</b>	ハイパー分類に使用するハイパー空間の最大量を指定します。
	<b>IDRCPCT</b>	圧縮をサポートする IBM 磁気テープ装置などの、改良データ記録機能 (IDRC) を使用して行う、データ短縮の率をパーセントで指定します。
	<b>IEEXIT</b>	DFSORT がユーザー・システムの ICEIEXIT ルーチンに制御を渡すかどうか指定します。
	<b>IGNCKPT</b>	チェックポイント / 再始動機能が実行時に要求され、かつブロック・セット手法 (チェックポイント / 再始動機能はサポートしない) が使用できる場合に、チェックポイント / 再始動機能を見捨てるかどうかを指定します。
	<b>IOMAXBF</b>	SORTIN、SORTINnn、SORTOUT などのデータ・セットに使用する、バッファ空間の上限の量を指定します。
	<b>LIST</b>	DFSORT が制御ステートメントを印刷するかどうかを指定します。
	<b>LISTX</b>	拡張機能サポート・プログラムが戻した制御ステートメントを、DFSORT が印刷するかどうかを指定します。
	<b>LOCALE</b>	ロケール処理を使用するかどうか、また使用する場合、活動ロケールを指定するかどうかを設定します。
	<b>MAXLIM</b>	16 メガバイト境界より下の、DFSORT が使用する主記憶域の上限の量を指定します。

## インストール・デフォルト

<b>MINLIM</b>	DFSORT が使用する主記憶域の下限の量を指定します。
<b>MSGCON</b>	マスター・コンソールに書き出すプログラム・メッセージのクラスを指定します。
<b>MSGDDN</b>	メッセージ・データ・セットの代替名を指定します。
<b>MSGPRT</b>	メッセージ・データ・セットに書き出すプログラム・メッセージのクラスを指定します。
<b>NOMSGDD</b>	メッセージ・データ・セットが要求され、使用できない場合、DFSORT を終了するか続行するかを指定します。
<b>ODMAXBF</b>	各 OUTFIL データ・セットごとに使用する、バッファ空間の上限を指定します。
<b>OUTREL</b>	未使用の一時出力データ・セット・スペースを解放するかどうかを指定します。
<b>OUTSEC</b>	一時的な出力データ・セットまたは新しい出力データ・セットに、DFSORT が自動的に 2 次割り振りを使用するようにするかどうかを指定します。
<b>OVERRGN</b>	REGION の値より上の、ブロック・セットが使用する主記憶域の量を指定します。
<b>OVFLO</b>	BI、FI、PD、または ZD の合計フィールドがオーバーフローした場合、DFSORT が行う処置を指定します。
<b>PAD</b>	LRECL の埋め込みのために DFSORT が行う処置を指定します。
<b>PARMDDN</b>	DFSORT DFSPARM データ・セットの代替 DD 名を指定します。
<b>RESALL</b>	SIZE/MAINSIZE=MAX が有効な場合に、システムとアプリケーションが使用する記憶域を予約します。
<b>RESET</b>	DFSORT が、REUSE で定義された VSAM 出力データ・セットを NEW または MOD データ・セットとして処理するかどうかを指定します。
<b>RESINV</b>	SIZE/MAINSIZE=MAX が有効な場合に、DFSORT を呼び出すプログラム用の記憶域を予約します。
<b>SDB</b>	ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用するかどうかを指定します。
<b>SDBMSG</b>	ブロック・サイズがゼロの場合に、DFSORT と ICETOOL がメッセージとリスト・データ・セットにシステム決定最適ブロック・サイズを使用するかどうかを指定します。

## インストール・デフォルト

<b>SIZE</b>	DFSORT が使用する主記憶域の量を指定します。
<b>SMF</b>	DFSORT が SMF タイプ 16 のレコードを生成するかどうかを指定します。
<b>SOLRF</b>	DFSORT が SORTOUT LRECL に再フォーマット設定されたレコード長を使用するかどうかを指定します。
<b>SORTLIB</b>	テープ作業データ・セット分類または従来の組み合わせで使用するモジュールを、DFSORT がシステムまたは専用ライブラリーで探索するかどうかを指定します。
<b>SPANINC</b>	不完全なスパン・レコードを検出した場合、DFSORT が行う処置を指定します。
<b>STIMER</b>	DFSORT が STIMER マクロを使用するかどうかを指定します。 DFSORT が STIMER マクロを使用しない場合、プロセッサ時間データは、SMF レコードまたは ICETEXIT 統計には表示されません。
<b>SVC</b>	DFSORT にユーザー SVC 番号を指定し、インストール・システムが同時に 2 つの異なるリリースの DFSORT を使用できるようにします。
<b>SZERO</b>	DFSORT が、数字の -0 および +0 値を、符号付きとして (つまり、異なる値として) 処理するか、符号なしとして (つまり、同じ値として) 処理するかを指定します。
<b>TEXTIT</b>	DFSORT がユーザー・システムの ICETEXIT ルーチンに制御を渡すかどうかを指定します。
<b>TMAXLIM</b>	SIZE/MAINSIZE=MAX が有効な場合に、DFSORT が使用する 16 メガバイト境界の上下を合計した、主記憶域量の上限を指定します。
<b>TRUNC</b>	LRECL の切り捨てのために DFSORT が行う処置を指定します。
<b>VERIFY</b>	出力レコードの順序を検査するかどうか指定します。
<b>VIO</b>	作業データ・セットの仮想割り振りを受け入れるかどうかを指定します。
<b>VLLONG</b>	DFSORT が長い可変長出力レコードを切り捨てるかどうかを指定します。
<b>VLSCMP</b>	DFSORT が短い可変長比較フィールドを埋め込むかどうかを指定します。
<b>VLSHRT</b>	短い可変長の制御フィールド、比較フィールド、または要約フィールドが検出された場合に、DFSORT が処理を続行するかどうかを指定します。

<b>VSAMBSP</b>	DFSORT が使用する VSAM バッファの数を指定します。
<b>VSAMEMT</b>	DFSORT が空の VSAM 入力データ・セットを受け入れるかどうかを指定します。
<b>VSAMIO</b>	DFSORT が、REUSE で定義された VSAM データ・セットのインプレースの分類を許可するかどうかを指定します。
<b>WRKREL</b>	使用しなくなった一時作業データ・セットのスペースを解放するかどうかを指定します。
<b>WRKSEC</b>	一時作業データ・セットに対して、DFSORT が自動的に 2 次割り振りを使用するかどうかを指定します。
<b>Y2PAST</b>	「世紀」のスライド・ウィンドウまたは固定ウィンドウを指定します。
<b>ZDPRINT</b>	集計結果を示す正の ZD フィールドから、DFSORT が印刷可能な数を生成するかどうかを指定します。

使用できるソースとオプションの指定変更の順序を示した表が、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』に記載されています。

## 他の分類用プロダクトから DFSORT への移行

DFSORT へ移行する場合は、IBM 提供の ICEMAC デフォルトを検討し、それらの値を、これまでご使用の分類用プロダクトの同等の設定値に対応するように必要に応じて変更する必要があります。特に、以下の表に示すオプションは、DFSORT に、他の分類用プロダクトによく似た動作をさせるので、移行が容易になります。*DFSORT 導入とカスタマイズ リリース 14* に記載されている ICEMAC オプションを使用すると、DFSORT の動作の方法をデフォルトでグローバルに変更できます。29 ページの『第 2 章 ジョブ制御言語による DFSORT の呼び出し』、および 111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』に記載されている実行時オプションを使用すると、特定のジョブについて ICEMAC オプションを指定変更できます。

表 2. 移行を容易にするオプション

ICEMAC	実行時
ABCODE=MSG/n	
DYNALOC=(d,n)	DYNALOC=(d,n)
DYNAUTO=YES/IGNWKDD/NO	DYNALOC=(d,n)
DYNSPC=n	DYNSPC=n
EQUALS=YES/NO/VBLKSET	EQUALS/NOEQUALS
EXITCK=STRONG/WEAK	EXITCK=STRONG/WEAK
FSZEST=YES/NO	FILSZ=n/En/Un
NOMSGDD=QUIT/ALL/CRITICAL/NONE	
PARMDDN=ddname	
RESET=YES/NO	RESET/NORESET

## インストール・デフォルト

表 2. 移行を容易にするオプション (続き)

ICEMAC	実行時
SORTLIB=SYSTEM/PRIVATE	
SZERO=YES/NO	SZERO/NOSZERO
VLLONG=YES/NO	VLLONG/NOVLLONG
VLSCMP=YES/NO	VLSCMP/NOVLSCMP
VSAMEMT=YES/NO	VSAMEMT/NVSAMEMT
VSAMIO=YES/NO	VSAMIO/NOVSAMIO
ZDPRINT=YES/NO	ZDPRINT/NZDPRINT

## DFSORT メッセージおよび戻りコード

DFSORT がメッセージをメッセージ・データ・セットまたはマスター・コンソールのどちらかへ書き込むか、あるいはその両方に書き込むかをインストールおよび実行時に決定できます。また、拡張機能サポート・プログラムに対して、メッセージをメッセージ・データ・セットに書き込むように指示することもできます。

メッセージ・データ・セットに書き込むメッセージは、インストール時または実行時に、重大エラー・メッセージ、通知エラー・メッセージ、または診断メッセージのいずれかに決定できます。

マスター・コンソールに書き込むメッセージは、インストール時に重大エラー・メッセージまたは通知エラー・メッセージのいずれかに決定できます。

DFSORT メッセージの詳細については、*DFSORT* メッセージ、コード、および診断の手引き リリース 14 を参照してください。

正常終了の場合、DFSORT はオペレーティング・システムまたは呼び出しプログラムに、戻りコード 0 または 4 を戻します。

オペレーティング・システムがサポートされていないために正常に終了しない場合、DFSORT はオペレーティング・システムまたは呼び出しプログラムに戻りコード 24 を戻します。

正常に終了せず、NOABEND が有効である場合、DFSORT は、オペレーティング・システムまたは呼び出しプログラムに戻りコード 16 または 20 を戻します。

正常に終了せず、ABEND が有効である場合、DFSORT は、ICEMAC オプション ABCODE により指定されるような該当するコード (エラー・メッセージ番号または 1 から 99 の間の番号) をもつユーザー異常終了を出します。

DFSORT が (レジスター 15 に) 戻す戻りコードの意味は次のとおりです。

- 0** 正常終了。DFSORT は正常に終了しました。
- 4** 正常終了。DFSORT は正常に終了し、かつ以下のいずれか。
  - OVFLO=RC4 が有効であって、合計フィールドがオーバーフローした。または、



- PAD=RC4 が有効であり、SORTOUT LRECL が SORTIN/SORTINnn より大きい (LRECL 埋め込み)。または、
- TRUNC=RC4 が有効であり、SORTOUT LRECL が SORTIN/SORTINnn より小さい (LRECL 切り捨て)。または、
- SPANINC=RC4 が有効であり、1 つ以上の不完全スパン・レコードが検出された。

16 異常終了。DFSORT が正常に終了するのを妨げるエラーを検出しました。

20 メッセージ・データ・セットの欠落。ICEMAC オプション NOMSGDD=QUIT が有効であり、メッセージ・データ・セット DD ステートメントも SYSOUT DD ステートメントも指定されませんでした。

24 サポートされていないオペレーティング・システム。このリリースの DFSORT は使用中のオペレーティング・システムをサポートしません。

---

## 可能な限りブロック・セットを使用する

ブロック・セットは、DFSORT のなかでも特に効果的な手法です。ブロック・セットは、効果の点では少し劣る DFSORT のピアレッジ / ベール手法や、従来の手法ではサポートされなかった、多くの機能をサポートします (これらの機能のリストについては、DFSORT メッセージ、コード、および診断の手引き リリース 14 の ICE189A を参照してください)。コピー・アプリケーションの場合は、DFSORT は、必ずブロック・セットを選択します。何か障害がある場合を除いて、DFSORT は、分類アプリケーションと組み合わせアプリケーションにも、必ずブロック・セットを選択します (ブロック・セットを使用できない理由のリストについては、DFSORT メッセージ、コード、および診断の手引き リリース 14 の ICE800I を参照してください)。

注: ブロック・セットは、BDAM データ・セットの処理には使用できません。

メッセージの ICE143I に、ブロック・セット、あるいはより効果的でない手法のいずれが、特定の実行に選択されたかを示します。分類あるいは組み合わせアプリケーションにブロック・セットが選択されなかった場合、メッセージ、ICE800I の理由コードをチェックしてください。ブロック・セットを使用できなかった理由がわかります。ICE800I を受け取らなかった場合は、次の DD ステートメントをアプリケーションに加えて、再度実行してください。

```
//SORTDIAG DD DUMMY
//SYSOUT DD SYSOUT=*
```

可能で適切であれば、ブロック・セットが選択されない原因となっている障害を取り除いてください。



---

## 第 2 章 ジョブ制御言語による DFSORT の呼び出し

JCL の使用	29
JOB ステートメントの使用	31
EXEC ステートメントの使用	32
EXEC ステートメントにカタログ式プロシーチャーを指定する	32
SORT カatalog式プロシーチャー	32
SORTD カatalog式プロシーチャー	34
EXEC/DFSPARM PARM オプションの指定	34
PARM オプションの別名	66
DD ステートメントの使用	67
DD 名の重複	70
磁気テープ装置の共有	70
システム DD ステートメント	70
プログラム DD ステートメント	73
SORTLIB DD ステートメント	74
SYMNAMES DD ステートメントと SYMNOU DD ステートメント	75
SORTIN DD ステートメント	75
SORTINnn DD ステートメント	77
SORTWKdd DD ステートメント	79
SORTOUT および OUTFIL DD ステートメント	82
SORTCKPT DD ステートメント	84
SORTCNTL DD ステートメント	84
DFSPARM DD ステートメント	85
SORTDKdd DD ステートメント	88
SORTDIAG DD ステートメント	88
SORTSNAP DD ステートメント	88
DD ステートメントの使用	88

---

### JCL の使用

オペレーティング・システムは、ユーザーが DFSORT プログラム制御ステートメントで指定するジョブ制御言語 (JCL) を使用して、以下の操作を行います。

- ユーザーを許可されたユーザーとして識別する。
- ユーザーのジョブを実行するために必要なリソースを割り振る。
- ユーザーのジョブを実行する。
- 結果に関する情報をユーザーに戻す。
- ユーザーのジョブを終了する。

対話式 DFSORT パネル機能を用いてジョブを作成する (*DFSORT* パネルの手引きを参照) 場合を除き、実行依頼する DFSORT のジョブにはすべて、JCL ステートメントが必要です。

JCL には、1 つの JOB ステートメント、1 つの EXEC ステートメント、およびいくつかの DD ステートメントを組み込む必要があります。必要なステートメントとその正確な形式は、次の条件により決まります。

- DFSORT を呼び出すために、入力ジョブ・ストリーム内の EXEC ステートメントを使用するか、別のプログラム内でシステム・マクロ命令のいずれを使用するか。

## JCL の使用

- DFSORT を呼び出すために、EXEC ステートメントのカタログ式プロシージャーを使用するかどうか。
- 各種の DFSORT 制御ステートメントまたは PARM オプションを指定するかどうか。
- プログラム出口を使用してユーザーのルーチンを活動化するかどうか。
- ダイナミック・リンク編集を使用するかどうか。
- 診断メッセージを見たいかどうか。

DFSORT パネルには、JCL を直接コーディングできる代替手段が備わっています。パネルを使用して、実行するジョブまたはデータ・セットに保管するジョブを作成する場合、必須 JCL の多くは DFSORT ユーザー・プロファイルの内容から自動的に供給されます。TSO の下の前景で送り込むためにユーザーが作成する DFSORT ジョブは、JCL ではなく CLIST を処理します。DFSORT パネルの使用方法については、*DFSORT* パネルの手引きを参照してください。

JCL ステートメントおよびそれらの機能について、以下に説明します。その後で、個々のステートメントのコーディングについて詳しく説明します。

JCL ステートメント	説明
<b>//JOB LIB DD</b>	ユーザーのプログラム・リンク・ライブラリーがまだシステムに知らされていない場合、そのライブラリーを定義します。
<b>//STEPLIB DD</b>	//JOB LIB DD と同じ。
<b>//SORTLIB DD</b>	特殊ロード・モジュールを含むデータ・セットを定義します (システムにまだ指定していない場合)。
<b>//SYSOUT DD<sup>1</sup></b>	メッセージ・データ・セットを定義します。
<b>//SYMNAMES DD</b>	シンボルの処理に使用するステートメントを入れる、SYMNAMES データ・セットを定義します。
<b>//SYMNOUT DD</b>	SYMNAMES ステートメントとシンボルのテーブルをリストするデータ・セットを定義します。
<b>//SORTIN DD<sup>1</sup></b>	分類またはコピーのための入力データ・セットを定義します。
<b>//SORTINnn DD<sup>1</sup></b>	組み合わせのための入力データ・セットを定義します。
<b>//SORTOUT DD<sup>1</sup></b>	分類、組み合わせ、コピー用の SORTOUT 出力データ・セットを定義します。
<b>//outfil DD</b>	分類、組み合わせ、コピー用の OUTFIL 出力データ・セットを定義します。
<b>//SORTWKdd DD<sup>1</sup></b>	分類のための中間記憶域データ・セットを定義します。
<b>//DFSPARM DD<sup>1</sup></b>	DFSORT PARM オプションおよびプログラム制御ステートメントを含んでいます。
<b>//SYSIN DD</b>	DFSORT プログラム制御ステートメントを含んでいます。

<b>//SORTCNTL DD<sup>1</sup></b>	//SYSIN DD と同じ。
<b>//SORTDIAG DD</b>	すべてのメッセージおよびプログラム制御ステートメントを印刷するように指定します。
<b>//SORTCKPT DD</b>	チェックポイント・レコード用のデータ・セットを定義します。
<b>//SYSUDUMP DD</b>	システム ABEND ダンプ・ルーチンからの出力用のデータ・セットを定義します。
<b>//SYSMDUMP DD</b>	//SYSUDUMP DD と同じ。
<b>//SYSABEND DD</b>	//SYSUDUMP DD と同じ。
<b>//SORTSNAP DD</b>	DFSORT が動的に割り振るスナップ・ダンプ・データ・セットを定義します。
<b>//ddname</b>	出口ルーチン (MODS プログラム制御ステートメントで指定した) を含んでいるデータ・セットを定義します。

以下の DD ステートメントは、出口ルーチンのダイナミック・リンク編集の場合だけ必要なものです。

**//SYSPRINT DD**  
リンケージ・エディター用のメッセージ・データ・セットを定義します。

**//SYSUT1 DD**  
リンケージ・エディター用の中間記憶域データ・セットを定義します。

**//SYSLIN DD**  
リンケージ・エディター用の制御情報データ・セットを定義します。

**//SYSLMOD DD**  
リンケージ・エディターの出力データ・セットを定義します。

**//SORTMODS DD**  
SYSIN からのユーザー出口ルーチン用の一時区分データ・セットを定義します。

<sup>1</sup> これらは、DFSORT が引き渡されたときのデフォルト DD 名です。SYSOUT および DFSPARM は、DFSORT インストール時に変更されている場合があります。示された DD 名は、すべて実行時にユーザー自身が変更できます。情報を上書きする場合は、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

---

## JOB ステートメントの使用

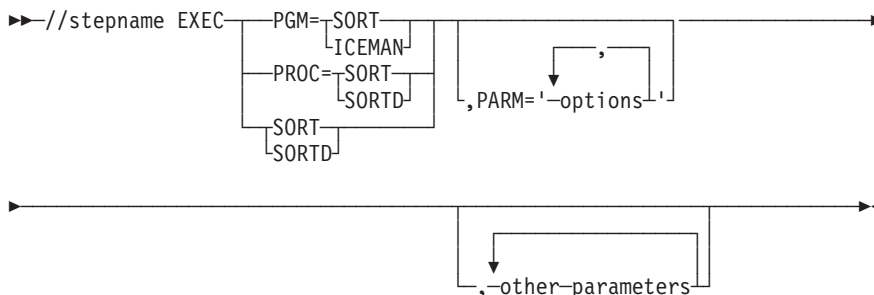
JOB ステートメントは、ユーザーのジョブの最初の JCL ステートメントです。このステートメントの名前フィールドには有効なジョブ名を、そして命令フィールドには JOB という語を指定します。オペランド・フィールドのパラメーターはすべて任意指定ですが、ユーザー・システム側でアカウント番号やプログラマーの名前などの情報を必須にすることもできます。

//jobname JOB accounting information, programmer's name, etc.

## EXEC ステートメントの使用

EXEC ステートメントは、各ジョブ・ステップのまたはカタログ式プロシージャー内の各プロシージャー・ステップの最初の JCL ステートメントです。これは、オペレーティング・システムに対して DFSORT を識別します。また、EXEC ステートメントに DFSORT オプションを指定することもできます。

EXEC ステートメントの形式は次のとおりです。



カタログ式プロシージャー (以下で詳述) を使用する場合は、PROC= SORT または PROC= SORTD を指定します。PROC= を省略して、単に SORT または SORTD を指定することもできます。しかし、PROC= を指定すれば、カタログ式プロシージャーが使用されていることを常に確認できます。

カタログ式プロシージャーを使用しない場合は、PGM= を使用して、分類モジュールの実際の名前 (ICEMAN) または別名の SORT、IERRCO00、IGHRCO00 のいずれかを指定します。ユーザー・システムでこの別名が変更されていないことを確認してください。

## EXEC ステートメントにカタログ式プロシージャーを指定する

カタログ式プロシージャーは、DD ステートメントを含む 1 組の JCL ステートメントで、名前が割り当てられ、プロシージャー・ライブラリーと呼ばれる区画データ・セットに入れられます。SORT および SORTD という 2 つのカタログ式プロシージャーが提供されます。これらのカタログ式プロシージャーは、PROC= SORT、PROC= SORTD、あるいは単に SORT または SORTD により、EXEC ステートメントの最初のパラメーターに指定します。

### SORT カタログ式プロシージャー

リンク・エディットの必要なユーザー・ルーチンを組み込む場合は、提供されている SORT カタログ式プロシージャーを使用できます。ユーザー・ルーチンを組み込むかどうかにかかわらず、SORT カタログ式プロシージャーはリンケージ・エディター・データ・セットを割り振るため、リンク・エディットされるユーザー・ルーチンがない場合にこのプロシージャーを使用することは非効率的です。

EXEC PROC= SORT または EXEC SORT を指定すると、次の JCL ステートメントが生成されます。

```

//SORT      EXEC  PGM=ICEMAN                      00
//STEPLIB   DD    DSNAME=yyy,DISP=SHR             10
//SORTLIB   DD    DSNAME=xxx,DISP=SHR             20
//SYSOUT    DD    SYSOUT=A                       30
//SYSPRINT  DD    DUMMY                          40
//SYSLMOD   DD    DSNAME=&GOSET,UNIT=SYSDA,SPACE=(3600,(20,20,1)) 50
//SYSLIN    DD    DSNAME=&LOADSET,UNIT=SYSDA,SPACE=(80,(10,10)) 60
//SYSUT1    DD    DSNAME=&SYSUT1,SPACE=(1024,(60,20)), 70
//          UNIT=(SYSDA,SEP=(SORTLIB,SYSLMOD,SYSLIN)) 80

```

## 行 説明

- 00** プロシーチャーのステップ名は SORT です。この EXEC ステートメントは、ICEMAN という名前のプログラムを開始します。
- 10** STEPLIB DD ステートメントは、DFSORT プログラム・モジュールを入れるデータ・セットを定義します。DFSORT が通常のリンク・ライブラリーの一部としてインストールされている場合は、STEPLIB DD ステートメントは不要です。このステートメントが必要なのは、DFSORT が「リンク・リスト」の一部ではない別のリンク・ライブラリーにある場合だけです。(この情報については、システム・プログラマーに問い合わせてください。)示されている STEPLIB DD ステートメントでは、yyy で表されているデータ・セット名はカタログされていると想定しています。
- 20** SORTLIB DD ステートメントは、テープ作業ファイルを使用する分類または従来の手法を使用する組み合わせに必要なモジュールを入れる、専用データ・セットを定義します。データ・セットはカタログされ、xxx で表されるデータ・セット名はインストール時に指定されています。xxx を SYS1.SORTLIB にすることができます。
- モジュールがシステム・ライブラリーにインストール済みで、ICEMAC SORTLIB=SYSTEM が使用される場合は、SORTLIB DD ステートメントは不要になり、ユーザー出口のダイナミック・リンクが使用されない限り無視されます。
- 30** システム使用のための出力データ・セット (メッセージ) を定義します。これはシステム出力クラス A に出力されます。
- 40** リンケージ・エディターの診断出力が不要であるため、SYSPRINT をダミー・データ・セットとして定義します。
- 50** リンケージ・エディター出力用のデータ・セットを定義します。どのようなシステム直接アクセス装置でも出力に使用できます。平均 3600 バイトの長さのレコード 20 個分のスペースが必要です。これは 1 次割り振りです。1 次スペース割り振りが十分でない場合は、さらに 20 レコード分のスペースが要求されます。これは 2 次割り振りです。1 次スペース割り振りがなくなるたびに要求されます。最後の値はディレクトリー用のスペースで、SYSLMOD が新しい区分データ・セットであるために、必要になります。
- 60** SYSLIN データ・セットは、リンケージ・エディター制御ステートメント用として、プログラムで使用されます。これは任意のシステム直接アクセス装置上に作成され、平均 80 バイトの長さのレコード 10 個分のスペースをもちます。1 次スペース割り振りを使い切ると、10 個分のレコードを入れる大きさの追加スペースがブロック単位で要求されます。ディレクトリー用のスペースは不要です。

## EXEC ステートメントの使用

**70/80** SYSUT1 DD ステートメントは、リンケージ・エディター用の作業データ・セットを定義します。

### **SORTD カタログ式プロシージャ**

ユーザー・ルーチンを組み込まない場合、あるいはリンク・エディットの必要がないユーザー・ルーチンを組み込む場合は、IBM 提供の SORTD カタログ式プロシージャを使用できます。

EXEC PROC=SORTD または EXEC SORTD を指定すると、次の JCL ステートメントが生成されます。

```
//SORT EXEC PGM=ICEMAN                                00
//STEPLIB DD DSNAME=yyy,DISP=SHR                      10
//SORTLIB DD DSNAME=xxx,DISP=SHR                      20
//SYSOUT DD SYSOUT=A                                  30
```

#### 行 説明

**00** SORTD プロシージャのステップ名は SORT です。

**10** STEPLIB DD ステートメントは、DFSORT プログラム・モジュールを入れるデータ・セットを定義します。DFSORT が通常のリンク・ライブラリーの一部としてインストールされている場合は、STEPLIB DD ステートメントは不要です。このステートメントが必要なのは、DFSORT が「リンク・リスト」の一部ではない別のリンク・ライブラリーにある場合だけです。(この情報については、システム・プログラマーに問い合わせてください。) 示されている STEPLIB DD ステートメントでは、yyy で表されているデータ・セット名はカタログされていると想定しています。

**20** SORTLIB DD ステートメントは、テープ作業ファイルを使用する分類または従来の手法を使用する組み合わせに必要なモジュールを入れる、専用データ・セットを定義します。xxx で表されるプログラム・サブルーチン・ライブラリーのデータ・セット名は、インストール時に指定されます。xxx を SYS1.SORTLIB にすることができます。

モジュールがシステム・ライブラリーにインストール済みで、ICEMAC SORTLIB=SYSTEM が使用される場合は、SORTLIB DD ステートメントは不要となり、ユーザー出口のダイナミック・リンクが使用されない限り無視されます。

**30** システム出カクラス A にメッセージを出力させます。

## EXEC/DFSPARM PARM オプションの指定

JCL で DFSORT を呼び出す場合、次ページの図のように、EXEC ステートメントの PARM パラメーターでいくつかの DFSORT オプションを指定できます。これらのオプションには、EFS、LIST、NOLIST、LISTX、NOLISTX、MSGPRT、および MSGDDN が含まれます。これらのオプションは、SYSIN の OPTION ステートメントで指定されている場合は無視されます。指定変更および適用の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。



## EXEC ステートメントの使用

代わりに DFSPARM DD ステートメントを使用すると、他のすべてのソースを変更する単一のソース・データ・セットに、EXEC PARM オプションと DFSORT 制御ステートメントの両方を指定できます。85 ページの『DFSPARM DD ステートメント』を参照してください。

PARM オプションの別名の詳細については、個々のオプションの説明に記載されています。使用可能な別名は、66 ページの『PARM オプションの別名』に要約されています。

DFSORT は、以下の EXEC/DFSPARM PARM オプションを受け入れますが、処理は行いません。受け入られるオプションは、BALANCE、BALN、BIAS=value、BMSG、CASCADE、CMP=value、CPU、CRCX、DEBUG、DIAG、ELAP、EXCPVR=value、IO、INCOR=value、INCORE=value、LRGSORT、L6=value、L7=value、NOCOMMAREA、NOINC、NOIOERR、OPT=value、OSCL、PEER、POLY、および PRINT121 です。

注: DEBUG が DFSPARM ステートメントで最初の値として指定される場合、DFSPARM PARM オプションとしてではなく、DEBUG 制御ステートメントとして解釈されます。

## EXEC ステートメントの使用

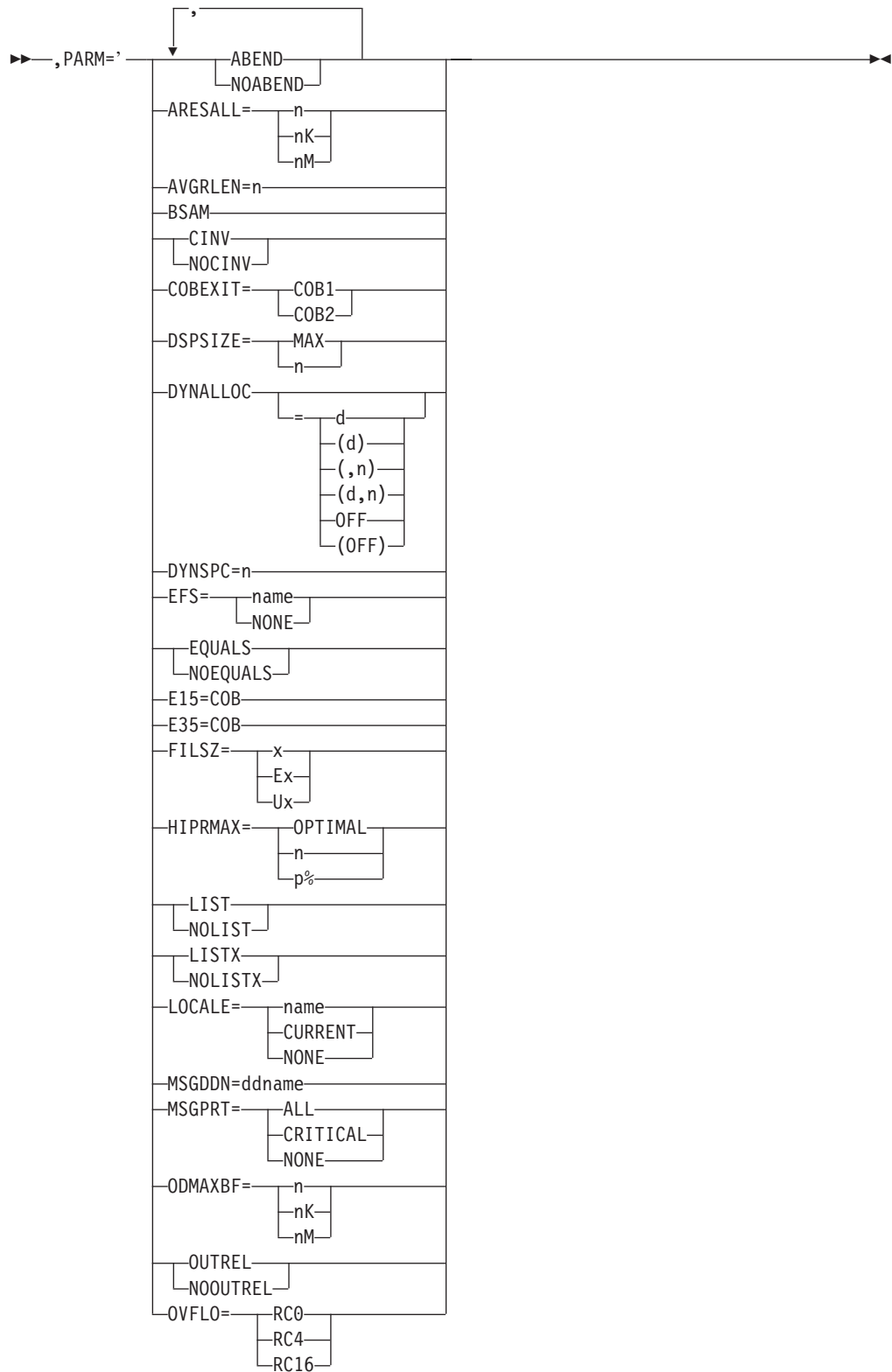


図 3. EXEC PARM の構文図 (1/3)

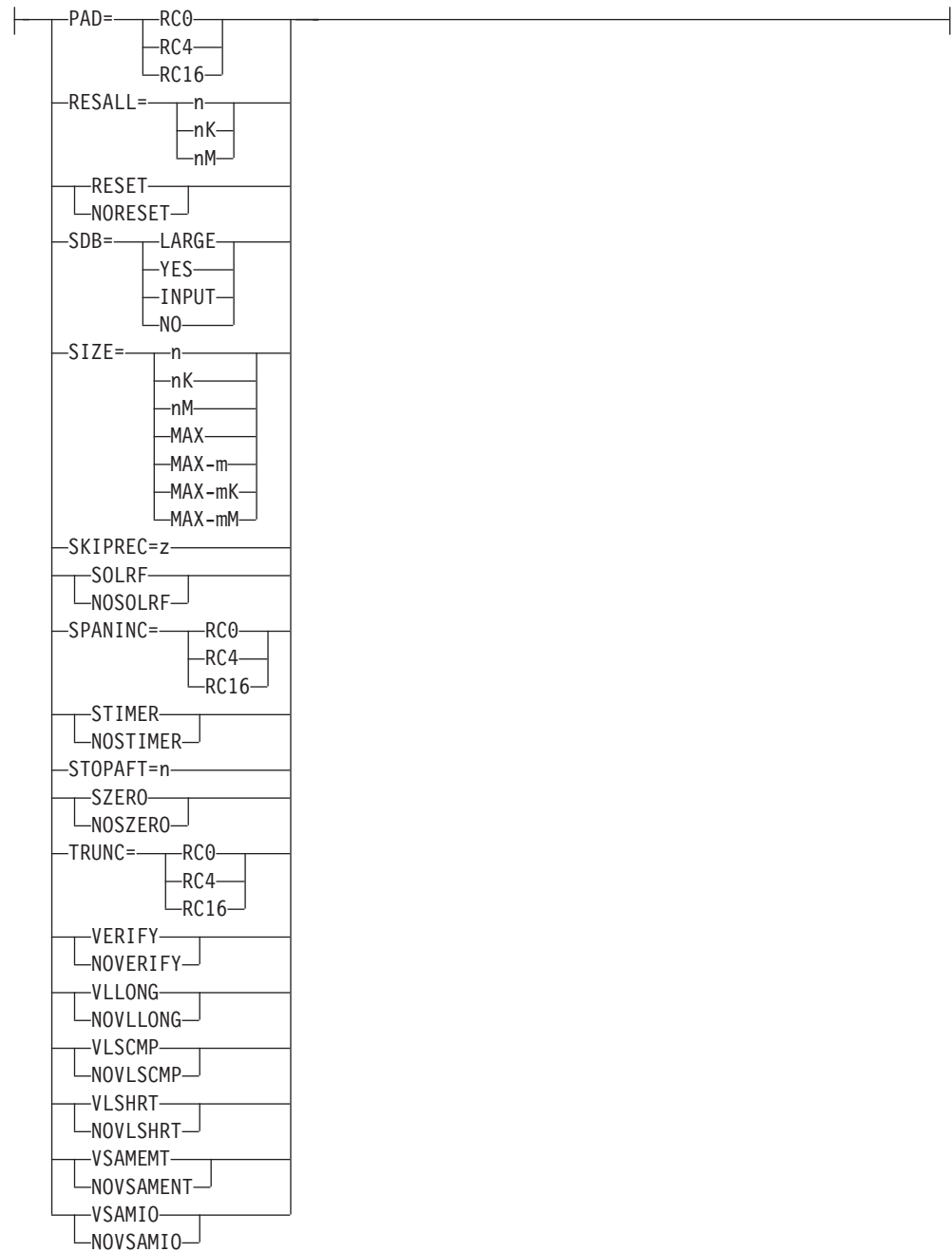


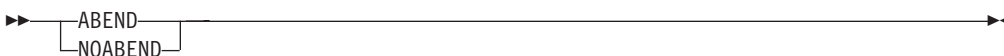
図3. EXEC PARM の構文図 (2/3)

## EXEC ステートメントの使用



図3. EXEC PARM の構文図 (3/3)

### ABEND または NOABEND



ERET インストール・オプションを一時的に指定変更して、分類、コピー、または組み合わせが成功しなかった場合に、DFSORT を異常終了または戻りコード 16 で終了させるかどうかを指定します。

#### ABEND

分類、コピー、または組み合わせが成功しなかった場合に、該当するメッセージ番号と同じユーザー完了コードか、もしくはインストール時に ICEMAC オプションの ABCODE=n でセットされた 1 ~ 99 の間のユーザー定義の番号で、DFSORT が異常終了することを指定します。

DEBUG ABEND が有効な場合、テープ作業データ・セットの分類または従来の組み合わせが成功しなかった場合に、ユーザー ABEND コード 0 を出すことができます。

#### NOABEND

成功しなかった分類、コピー、または組み合わせが、戻りコード 16 で終了することを指定します。

#### 注:

1. ABEND および NOABEND の代わりに、それぞれ RC16=ABE および NORC16 を使用できます。
2. DFSORT が、SmartBatch パイプ・データ・セットが使用中であると判断すると、ABEND オプションを自動的にオンにし、エラーが検出された場合は異常終了させます。これにより、同じ SmartBatch パイプ・データ・セットにアクセスしている可能性のある他のアプリケーションに対して、システムが適切なエラー通知を行います。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### ARESALL



ARESALL インストール・オプションを一時的に指定変更し、仮想記憶域の 16 メガバイト境界より上にシステム用に予約するバイト数を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の ARESALL オプションの説明を参照してください。

**n** n バイトの記憶域を予約することを指定します。

限界: 8 桁

**nK**

1024 バイトの n 倍の記憶域を予約することを指定します。

限界: 5 桁

**nM**

1048576 バイトの n 倍の記憶域を予約することを指定します。

限界: 2 桁

**注:** ARESALL=value の代わりに RESERVEX=value を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## AVGRLLEN



可変長レコードの分類アプリケーションの入力レコード長の平均をバイト単位で指定します。詳細については、181 ページの『OPTION 制御ステートメント』の AVGRLLEN オプションの説明を参照してください。

**n** 入力レコード長の平均を指定します。n の値は 4 から 32767 の範囲で指定します。また、4 バイトのレコード記述子ワード (RDW) を含む必要があります。

**注:** AVGRLLEN=n の代わりに L5=n を使用できます。

**デフォルト:** AVGRLLEN=n が指定されていない場合は、DFSORT は最大レコード長の 2 分の 1 を平均レコード長として使用します。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## EXEC ステートメントの使用

### BSAM

▶▶ BSAM ◀◀

入出力データ・セットに通常使用する EXCP アクセス方式を、一時的にバイパスします。BSAM は、VSAM 入出力データ・セットの場合は無視されます。ブロック・セットを選択せず、連結 SORTIN 入力で BSAM 処理を使用し、さらにヌルのデータ・セットとヌル以外のデータ・セットの両方を指定している場合は、非ヌル・データ・セット前にヌル・データ・セットが配置される必要があります。それ以外の場合、予期せぬ結果を招くことがあります。

**注:** このオプションはパフォーマンスを低下させる可能性があります。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### CINV または NOCINV

▶▶ CINV  
NOCINV ◀◀

CINV インストール・オプションを一時的に指定変更して、DFSORT が VSAM データ・セットに制御インターバル・アクセスを使用できるかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の CINV オプションの説明を参照してください。

#### CINV

VSAM データ・セットに対して、可能な場合は制御インターバル・アクセスを使用するように DFSORT に指示します。

#### NOCINV

制御インターバル・アクセスを使用しないように DFSORT に指示します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### COBEXIT

▶▶ COBEXIT= COB1  
COB2 ◀◀

COBEXIT インストール・オプションを一時的に指定変更して、COBOL E15 および E35 ルーチン用のライブラリーを指定します。

**COB1**

COBOL E15 および E35 ルーチンを、OS/VS COBOL 実行時ライブラリーを用いて、あるいは場合によりは COBOL 実行時ライブラリーを用いずに実行することを指定します。

**COB2**

COBOL E15 および E35 ルーチンを、VS COBOL II 実行時ライブラリーまたは言語環境プログラム実行時ライブラリーのどちらかを用いて実行することを指定します。

**注:** COBEXIT=COB2 の使用については、397 ページの『COBOL ユーザー出口に関する要件』を参照してください。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**DSPSIZE**

▶▶ DSPSIZE= MAX  
n ▶▶

**DSPSIZE** インストール・オプションを一時的に指定変更して、データ空間分類に使用するデータ空間の最大値を指定します。データ空間分類の詳細については、181 ページの『OPTION 制御ステートメント』の DSPSIZE オプションの説明を参照してください。

**MAX**

データ空間分類に使用するデータ空間の最大量を、DFSORT が動的に決定するように指定します。この場合、DFSORT は分類するファイルのサイズ、およびシステムのページング活動に基づいて、そのデータの空間の使用を決定します。

**n** データ空間分類に使用するデータ空間の最大量をメガバイト単位で指定します。n の値は、0 から 9999 の範囲で指定します。使用するデータ空間の実際の量は、n 未満になります。これは分類の対象となるファイルのサイズやシステムのページング活動に基づいて決まります。

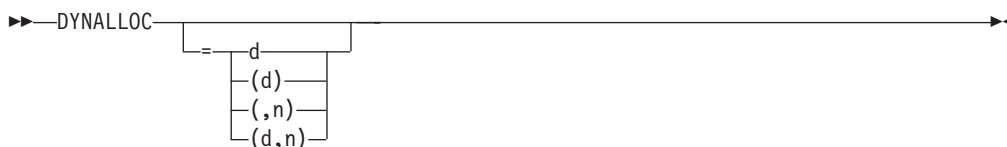
n がゼロの場合は、データ空間分類は使用されません。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**DYNALLOC**

## EXEC ステートメントの使用



DFSORT が必要なワークスペースを動的に割り振るように指定します。プログラムで必要なワークスペースの大きさを計算して、JCL で指定する必要はありません。

詳細については、181 ページの『OPTION 制御ステートメント』および 685 ページの『付録 A. ワークスペースの使用』の DYNALLOC オプションの説明を参照してください。

**d** 装置名を指定します。ユーザーのシステムでサポートされている、IBM 直接アクセス記憶装置または磁気テープ装置であれば、JCL UNIT パラメータに指定するのと同じ方法で指定できます。また、DISK や SYSDA などのグループ名を指定することもできます。

最高のパフォーマンスを得るため、エミュレートされた 3390-9 装置 (RAMAC など) または別の高速 IBM DASD 装置を指定し、磁気テープ、仮想 (VIO) または実 3390-9 装置の指定は避けてください。

**n** 要求する作業データ・セットの最大数を指定します。255 よりも大きな値を指定すると、255 が使用されます。1 を指定し、ブロック・セット手法が選択されると、最大 2 つのデータ・セットが使用されます。32 よりも大きな値を指定し、ブロック・セット手法が選択されない場合は、最大 32 のデータ・セットが使用されます。

**注:** 仮想記憶域などのリソースを最適に割り振るために、作業データ・セットの数は必要以上に多く指定しないでください。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### DYNALLOC=OFF



DFSORT が中間ワークスペースを動的に割り振らないように指示します。ICEMAC インストール・オプションの DYNAUTO=YES または実行時に指定された DYNALLOC パラメータ (OFF の指定なし) を指定変更します。詳細については、181 ページの『OPTION 制御ステートメント』の DYNALLOC オプションの説明を参照してください。

### OFF

DFSORT が中間ワークスペースを動的に割り振らないように指示します。



デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## DYNSPC

▶▶—DYNSPC=n————▶▶

入力ファイルのサイズが不明な場合に動的に割り振られるすべての作業データ・セットへの、デフォルトの 1 次スペース割り振り全体を指定する DYNSPC インストール・オプションを、一時的に指定変更します。すなわち、DFSORT が分類アプリケーションへの入力ファイルのサイズを決定できず、複数のレコードが FILSZ または SIZE の値で提供されないケースです。詳細については、181 ページの『OPTION 制御ステートメント』の DYNSPC オプションの説明を参照してください。

**n** すべての 動的割り振り作業データ・セットに割り振られる、デフォルトの 1 次スペース 全体 をメガバイトで指定します (n は、各データ・セットの 1 次スペース ではありません)。n の値は 1 から 65535 の範囲で指定します。

使用可能な DASD スペースを超える値を指定しないでください。その値を使用する分類アプリケーションに対し、動的割り振りで障害が発生します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## EFS

▶▶—EFS=name————▶▶  
           └─NONE─┘

EFS インストール・オプションを一時的に指定変更して、DFSORT が EFS プログラムへ制御を渡すかどうかを指定します。EFS の詳細については、581 ページの『第 8 章 拡張機能サポートの使用』を参照してください。

### name

DFSORT とのインターフェースのために呼び出される EFS プログラムの名前を指定します。

### NONE

EFS プログラムに対する呼び出しが行なわれないことを意味します。

**注:** SORT、MERGE、INCLUDE、または OMIT フィールドのロケール処理を使用する場合は、EFS プログラムは使用しないでください。DFSORT のロケ

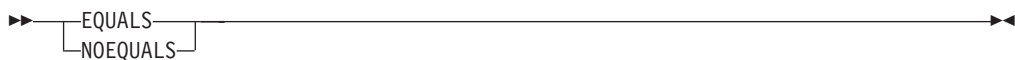
## EXEC ステートメントの使用

ール処理を使用すれば、EFS プログラムを使用する必要がない場合があります。ロケール処理の詳細については、181 ページの『OPTION 制御ステートメント』を参照してください。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EQUALS または NOEQUALS



EQUALS インストール・オプションを一時的に指定変更し、分類または組み合わせの場合に照合結果が同じになったレコードの順序を、入力から出力まで元の順序のまま維持するかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の EQUALS および NOEQUALS オプションの説明を参照してください。

#### EQUALS

元の順序を維持しなければならないことを指定します。

#### NOEQUALS

元の順序を維持する必要がないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### E15=COB



E15 ユーザー・ルーチンが COBOL で作成されることを指定し、E15 についての MODS ステートメントを一時的に指定変更します。E15=COB を指定しても、MODS ステートメントで E15 モジュールを指定しなければ、E15=COB は無視されます。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### E35=COB

▶▶—E35=COB—▶▶

E35 ユーザー・ルーチンが COBOL で作成されることを指定し、E35 についての MODS ステートメントを一時的に指定変更します。E35=COB を指定しても、MODS ステートメントで E35 モジュールを指定しなければ、E35=COB は無視されます。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## FILSZ

▶▶—FILSZ=—x—▶▶  
 —Ex—  
 —Ux—

分類または組み合わせに使用されるレコードの正確な数、または分類に使用されるレコードの見積数を指定します。このレコード数は、次の 2 つの目的のために DFSORT が使用します。

1. 分類または組み合わされたレコードの実際の数、が、予定したレコードの正確な数であるかどうかを検査するため。FILSZ=x はこの検査を実行し、数が等しくない場合はメッセージ ICE047A を出して処理を終了させます。
2. 分類アプリケーションに使用する入力ファイルのサイズを決定するため。DFSORT は、ユーザーが指定したレコード数および他のパラメーター (AVGRLLEN など) に基づいて計算を行い、分類するバイト数の合計を見積もります。この値は、動的作業データ・セットの割り振りの場合と同様に、いくつかの内部最適化のために使用されるので重要です (OPTION DYNALLOC を参照)。分類アプリケーションに入力レコード数が指定されていない (または見積もりのみ指定されている) 場合は、DFSORT は最適化および割り振りに使用するために、自動的にファイル・サイズを計算しようとします。

指定された FILSZ 値のタイプ (x、Ex、Ux、またはなし) は、DFSORT が上記の 2 つの機能を実行する方法を制御し、パフォーマンスと作業データ・セット割り振りに重要な効力を持ちます。ファイル・サイズの考慮事項については、625 ページの『入出力データ・セット特性の正確な指定』と 688 ページの『作業データ・セットの割り振り』を参照してください。

- x 分類または組み合わせが行われるレコードの正確な数を指定します。レコード検査およびファイル・サイズ計算の両方で、この値が必ず使用されます。FILSZ=x を使用することにより、x に基づくファイル・サイズの計算を DFSORT に強制的に実行させ、また x が正確でない場合は、DFSORT に分類または組み合わせアプリケーションを終了させることができます。

FSZEST=NO インストール・オプションが有効で FILSZ=x が指定されている場合、終了の前にメッセージ ICE047A (またはメッセージ ICE054I) の IN フィールドに入れられたレコードの実際数が指定の値 (x) と異なっていると、DFSORT は終了します。しかし、FSZEST=YES インストール・オ

## EXEC ステートメントの使用

プシオンが有効な場合は、DFSORT が FILSZ=x を FILSZ=Ex と同様に扱い、実際の数字が x と同じではなくても終了しません。

指定値 (x) には、入力データ・セット内のレコード、出口 E15 または出口 E32 が挿入または削除するレコード、INCLUDE/OMIT ステートメント、SKIPREC、STOPAFT が削除するレコードの数を考慮に入れる必要があります。分類または組み合わせを行なうレコードの数がなんらかの方法で大幅に変更される場合は、必ず x を変更する必要があります。

FILSZ=0 で、ハイパー分類、データ空間分類、ワークスペースの動的割り振りを使用しないようにした場合、この指定は、分類または組み合わせを行なうレコードの数が 0 でない限り、メッセージ ICE047A を出して終了します。

限界: 28 桁 (有効数字 15 桁)

**Ex** 分類されるレコードの見積もり数を指定します。この値はレコード検査には使用されません。DFSORT が自動的にファイル・サイズを計算しない場合のみ、この値はファイル・サイズ計算に使用されます。それ以外の場合は、DFSORT はこの値を無視します。DFSORT が FILSZ=Ex を使用したり、無視したりするタイミングについては、689 ページの『作業データ・セットの動的割り振り』を参照してください。

指定する値 (x) には、入力データ・セットのレコード、出口 E15 が挿入または削除するレコード、INCLUDE/OMIT ステートメント、SKIPREC、STOPAFT が削除するレコードの数などを考慮に入れる必要があります。分類するレコードの数が大幅に変更される場合は、x を必ず変更します。

FILSZ=E0 は常に無視されます。

限界: 28 桁 (有効数字 15 桁)

### Ux

分類するレコードの数を指定します。この値はレコード検査には使用されませんが、ファイル・サイズ計算には必ず使用されます。FILSZ=Ux を使用すると、x が正確でない場合でも終了させないで、x に基づくファイル・サイズ計算を DFSORT に強制的に行わせることができます。

FSZEST インストール・オプションは、FILSZ=Ux 処理には無効です。

指定する値 (x) には、入力データ・セットのレコード、出口 E15 が挿入または削除するレコード、INCLUDE/OMIT ステートメント、SKIPREC、STOPAFT が削除するレコードの数などを考慮に入れる必要があります。分類するレコードの数が大幅に変更される場合は、x を必ず変更します。

FILSZ=U0 の指定により、ハイパー分類、データ空間分類およびワークスペースの動的割り振りが使用されず、分類されるレコードの実際のが 1 以上で、相当大きな場合は、パフォーマンスの低下を引き起こすか、またはメッセージ ICE046A を出して終了します。

限界: 28 桁 (有効数字 15 桁)

表 3 は、3 種類の FILSZ の相違点を要約したものです。

表 3. 各種 FILSZ の要約：

インストール・オプション FSZEST=YES が指定されている場合、FILSZ=n は FILSZ=En と同等になります。

	FILSZ=n	FILSZ=Un	FILSZ=En
レコード数	正確	見積もり	見積もり
アプリケーション	分類、組み合わせ	分類	分類
誤りがあれば終了するか	Y	N	N
ファイル・サイズ計算に使用するか	Y	Y	DFSORT がファイル・サイズを計算できない場合
n には次のレコードが含まれる			
入力データ・セット	Y	Y	Y
E15 により挿入 / 削除されるレコード	Y	Y	Y
E32 により挿入されるレコード	Y	N	N
INCLUDE/OMIT により削除されたレコードステートメント	Y	Y	Y
SKIPREC により削除されるレコード	Y	Y	Y
STOPAFT により削除されるレコード	Y	Y	Y
レコード数が変更されたときの n の更新:	任意に	大幅に	大幅に
n=0 の効果	ハイパー分類および DYNALLOC が使用されない	ハイパー分類および DYNALLOC が使用されない	なし

**注:** FILSZ パラメーターを使用して不正確な情報を DFSORT に与えると、DFSORT のパフォーマンスに悪影響を与える場合があります。また、ワークスペースが動的に割り振られると、DASD スペースを無駄にしたり、メッセージ ICE083A または ICE046A を出して終了してしまうことがあります。したがって、分類が行われるレコードの数が大幅に変更される場合は、必ずレコード数の値を更新することが重要です。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## HIPRMAX

## EXEC ステートメントの使用



**HIPRMAX** インストール・オプションを一時的に指定変更して、ハイパー分類用に確保してあるハイパー空間の最大値を指定します。詳細については、181ページの『OPTION 制御ステートメント』の **HIPRMAX** オプションの説明を参照してください。

### OPTIMAL

ハイパー分類用に使用されるハイパー空間の最大値を、DFSORT が動的に決定することを指定します。

**n** ハイパー分類用に使用されるハイパー空間の最大値 (n メガバイトの限界値) を、DFSORT が動的に決定するように指定します。n の値は 0 から 32767 の範囲で指定します。n が 0 の場合、ハイパー分類は使用されません。

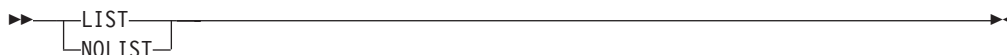
### p%

ハイパー分類用に使用されるハイパー空間の最大値を、構成される拡張記憶域の p% を限界として、DFSORT が動的に決定するように指定します。64 ビットの実モードでは、`HIPRMAX=p%` は、中央記憶装置の適用可能な部分の割合を指定します。p の値は 0 から 100 の範囲で指定します。p が 0 の場合、ハイパー分類は使用されません。p% の計算値は、32767 メガバイトを上限とし、メガバイト未満の端数は切り捨てられます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## LIST または NOLIST



**LIST** インストール・オプションを一時的に指定変更して、DFSORT プログラム制御ステートメントをメッセージ・データ・セットに書き込むかどうかを指定します。メッセージ・データ・セットの使用方法については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

### LIST

すべての DFSORT 制御ステートメントをメッセージ・データ・セットに印刷することを指定します。

### NOLIST

DFSORT 制御ステートメントを印刷しないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## LISTX または NOLISTX

```
▶▶—LISTX—▶▶
   |
   |—NOLISTX—|
```

LISTX インストール・オプションを一時的に指定変更して、EFS プログラムにより戻されるプログラム制御ステートメントを、DFSORT がメッセージ・データ・セットに書き込むかどうかを指定します。メッセージ・データ・セットの使用方法については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

### LISTX

EFS プログラムにより戻される制御ステートメントをメッセージ・データ・セットに印刷することを指定します。

### NOLISTX

EFS プログラムにより戻される制御ステートメントをメッセージ・データ・セットに印刷しないことを指定します。

### 注:

1. 最終指定変更規則が適用された後で EFS=NONE が有効な場合は、NOLISTX は有効になります。
2. LISTX と NOLISTX は LIST と NOLIST とは関係なく使えます。
3. EFS 制御ステートメントの印刷についての詳細は、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## LOCALE

```
▶▶—LOCALE=—▶▶
   |
   |—name—|
   |—CURRENT—|
   |—NONE—|
```

LOCALE インストール・オプションを一時的に指定変更して、ロケール処理を使用するかどうかを指定します。ロケール処理を使用する場合は、活動状態のロケールを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の LOCALE オプションの説明を参照してください。

**name** ロケール処理を使用することを指定し、DFSORT 処理の間に活動状態にするロケールの名前を指定します。

ロケールは記述名を使用して指定されます。たとえば、フランス語とカナダの文化慣行を表す活動ロケールを設定するには、LOCALE=FR\_CA

## EXEC ステートメントの使用

と指定します。記述ロケール名には、最高 32 文字まで指定できます。ロケール名は大文字 / 小文字の区別をしません。ロケールの命名規則の詳細については、[ロケールの使用](#) を参照してください。

IBM 提供のロケールとユーザー定義のロケールを使用できます。

DFSORT に入る以前の活動ロケールの状態は、DFSORT の完了時に復元されます。

### CURRENT

ロケール処理が使用されることを指定し、DFSORT に入る際の現行活動ロケールが、DFSORT 処理の間も活動ロケールのままになります。

**NONE** ロケール処理が使用されないことを指定します。DFSORT は、ユーザー・データを照合し比較するために定義されたコード・ページの 2 進数のエンコードを使用します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### MSGDDN

▶▶—MSGDDN=ddname—◀◀

MSGDDN インストール・オプションを一時的に指定変更して、メッセージ・データ・セットの代替 DD 名を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の MSGDDN オプションの説明を参照してください。

DD 名は、1 文字から 8 文字までの任意の名前を使用できますが、ジョブ・ステップにおいて固有である必要があります。DFSORT が使用する名前 (たとえば SORTIN) は使用しないでください。指定された DD 名が実行時に使用できない場合は、代わりに SYSOUT が使用されます。メッセージ・データ・セットの使用の詳細については、[DFSORT メッセージ、コード、および診断の手引きリリース 14](#) を参照してください。

注: MSGDDN=ddname の代わりに MSGDD=ddname を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### MSGPRT





MSGPRT インストール・オプションを一時的に指定変更して、メッセージ・データ・セットに書き込まれるメッセージのクラスを指定します。メッセージ・データ・セットの使用方法については、*DFSORT* メッセージ、コード、および診断の手引き リリース 14 を参照してください。

### ALL

診断メッセージ ICE800I から ICE999I 以外のすべてのメッセージをメッセージ・データ・セットに印刷するように指定します。制御ステートメントは、LIST が有効な場合のみ印刷されます。

### CRITICAL

重要なメッセージだけがメッセージ・データ・セットに印刷されるように指定します。制御ステートメントは、LIST が有効な場合のみ印刷されます。

### NONE

メッセージも制御ステートメントも印刷しないことを指定します。

注: FLAG(I)|FLAG(U)|NOFLAG、および

MSG={NO|NOFI|ABI|API|AC|CB|CC|CP|PC|SC|SP} の形式も受け入れられます。次の表は、代替の形式についての MSGPRT/MSGCON 指定の対応関係を示しています。

オプション	MSGPRT	MSGCON
NO	NONE	NONE
NOF	NONE	NONE
AB	ALL	ALL
AP	ALL	CRITICAL
AC	NONE	ALL
CB	CRITICAL	CRITICAL
CC	NONE	CRITICAL
CP	CRITICAL	CRITICAL
PC	ALL	ALL
SC	ALL	CRITICAL
SP	CRITICAL	ALL
NOFLAG	NONE	CRITICAL
FLAG(I)	ALL	CRITICAL
FLAG(U)	CRITICAL	CRITICAL

図 4. MSGPRT/MSGCON オプションの別名

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## EXEC ステートメントの使用

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### ODMAXBF



ODMAXBF インストール・オプションを一時的に指定変更して、各 OUTFIL データ・セット用に DFSORT が使用できる最大バッファ・スペースを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の ODMAXBF オプションの説明を参照してください。

**n** 各 OUTFIL データ・セットに、最大  $n$  バイトのバッファ・スペースを使用することを指定します。262144 よりも小さい値を指定すると、262144 が使用されます。16777216 よりも大きな値を指定すると、16777216 が使用されます。

限界: 8 桁

#### nK

各 OUTFIL データ・セットに、1024 バイトの  $n$  倍の最大バッファ・スペースを使用することを指定します。256 キロバイトよりも小さな値を指定すると、256 キロバイトが使用されます。16384 キロバイトよりも大きな値を指定すると、16384 キロバイトが使用されます。

限界: 5 桁

#### nM

各 OUTFIL データ・セットに、最大 1048576 バイトの  $n$  倍のバッファ・スペースを使用することを指定します。0 メガバイトを指定すると、256 キロバイトが使用されます。16 メガバイトよりも大きな値を指定すると、16 メガバイトが使用されます。

限界: 2 桁

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### OUTREL または NOOUTREL



OUTREL インストール・オプションを一時的に指定変更して、未使用の一時出力データ・セット・スペースを解放するかどうかを指定します。

#### OUTREL

未使用の一時出力データ・セット・スペースを解放するように指定します。

**NOOUTREL**

使用していない一時出力データ・セットのスペースを解放しないことを指定します。

**注:** OUTREL および NOOUTREL の代わりに、それぞれ RLSOUT および NORLSOUT を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**OVFLO**

OVFLO インストール・オプションを一時的に指定変更して、BI、FI、PD または ZD 合計フィールドがオーバーフローしたときの、DFSORT の処置を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の OVFLO オプションの説明を参照してください。

**RC0**

DFSORT に対して、合計フィールドがオーバーフローしたら、メッセージ ICE152I を (一度) 発行し、戻りコードを 0 に設定して処理を続行するよう指定します。

**RC4**

DFSORT に対して、合計フィールドがオーバーフローしたら、メッセージ ICE152I を (一度) 発行し、戻りコードを 4 に設定して、処理を続行するよう指定します。

**RC16**

DFSORT に対して、合計フィールドがオーバーフローしたら、メッセージ ICE195A を発行して終了し、戻りコード 16 を戻すよう指定します。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**PAD**

PAD インストール・オプションを一時的に指定変更して、DFSORT が LRECL 埋め込みを許可する場合に、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときの DFSORT の処置を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の PAD オプションの説明を参照してください。

## EXEC ステートメントの使用

### RC0

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE171I を (一度) 発行し、戻りコードを 0 に設定して処理を続行するよう指定します。

### RC4

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE171I を発行し、戻りコードを 4 に設定して処理を続行するよう指定します。

### RC16

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE196A を発行して終了し、戻りコード 16 を戻すよう指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## RESALL



RESALL インストール・オプションを一時的に指定変更して、SIZE/MAINSIZE=MAX が有効な場合にシステム使用のために REGION に予約するバイト数を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の RESALL オプションの説明を参照してください。

**n** n バイトの記憶域を予約することを指定します。4096 よりも小さい値を指定すると、4096 が使用されます。

限界: 8 桁

### nK

1024 バイトの n 倍の記憶域を予約することを指定します。4 キロバイトよりも小さい値を指定すると、4 キロバイトが使用されます。

限界: 5 桁

### nM

1048576 バイトの n 倍の記憶域を予約することを指定します。0 メガバイトを指定すると、4 キロバイトが使用されます。

限界: 2 桁

**注:** RESALL=value の代わりに RESERVE=value を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## RESET または NORESET



RESET インストール・オプションを一時的に指定変更して、DFSORT が、REUSE で定義された VSAM 出力データ・セットを NEW または MOD データ・セットとして処理するかどうかを指定します。

### RESET

DFSORT が、REUSE で定義された VSAM 出力データ・セットを NEW データ・セットとして処理することを指定します。頻繁に使用される RBA はゼロにリセットされ、出力データ・セットは実際は、当初空のクラスターとして扱われます。

### NORESET

DFSORT が、REUSE で定義された VSAM 出力データ・セットを MOD データ・セットとして処理することを指定します。頻繁に使用される RBA はリセットされず、出力データ・セットは実際は、当初空でないクラスターとして扱われます。

注: REUSE なしで定義された VSAM 出力データ・セットは、MOD データ・セットとして処理されます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## SDB



SDB インストール・オプションを一時的に指定変更して、ブロック・サイズがゼロと指定されたか、またはデフォルトのゼロが使用された場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用するかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の SDB オプションの説明を参照してください。

### LARGE

ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用することを指定します。OS/390 リリース 10 以上では、SDB=LARGE を指定すると、DFSORT は該当する場合、テープ出力データ・セットに 32760 バイトより大きいブロック・サイズを選択できます。

### YES

ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシス

## EXEC ステートメントの使用

テム決定最適ブロック・サイズを使用するが、選択されるブロック・サイズを 32760 バイトの最大値に限定することを指定します。

### INPUT

ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシステム決定の最適ブロック・サイズを使用するが、入力ブロック・サイズが 32760 以下であれば、選択されるブロック・サイズを 32760 バイトの最大値に限定することを指定します。

### NO

DFSORT が出力データ・セットにシステム決定の最適ブロック・サイズを使用しないことを指定します。

**注:** SDB=YES の代わりに SDB、SDB=ON、および SDB=SMALL を使用できます。SDB=NO の代わりに NOSDB および SDB=OFF を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SIZE



**SIZE** インストール・オプションを一時的に指定変更して、DFSORT が使用できる主記憶域の大きさを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の MAINSIZE オプションの説明を参照してください。

**n** n バイトの記憶域を割り振ることを指定します。2097152000 よりも大きな値を指定すると、2097152000 が使用されます。

制限: 10 桁

#### nK

1024 バイトの n 倍の記憶装置を割り振ることを指定します。2048000 キロバイトよりも大きな値を指定すると、2048000 キロバイトが使用されます。

制限: 7 桁

#### nM

割り振られる記憶域が 1048576 バイトの n 倍であることを指定します。2000 メガバイトよりも大きな値を指定すると、2000 メガバイトが使用されます。

限界: 4 桁

**MAX**

使用できる主記憶域の大きさを計算しアプリケーションに適切な最大サイズ (最大で TMAXLIM または MAXLIM のインストール時値までの) を割り振ることを DFSORT に指示します。

4 キロバイトよりも小さい値を指定すると、4 キロバイトが使用されます。

**MAX-m**

RESALL 値 (m) をバイト単位で指定します。MAX-m は DFSORT に、使用できる記憶域の大きさを計算し、割り振るよう指示します。この大きさは、システムとアプリケーションが使用するために予約された記憶域 (RESALL) を MAX 値から 差し引いた 値が上限となります。

4096 よりも小さな値を m に指定すると、4096 が使用されます。

m の制限: 8 桁

**MAX-mK**

RESALL 値 (1024 の m 倍) をキロバイト単位で指定します。MAX-mK は DFSORT に、使用する記憶域の大きさを計算し、割り振るよう指示します。割り振る大きさは、システムとアプリケーションが使用するために予約されている記憶域 (RESALL) を MAX 値から 差し引いた 値が上限となります。

4 キロバイトよりも小さい値を m に指定すると、4 キロバイトが使用されます。

m の限界: 5 桁

**MAX-mM**

RESALL の値 (1048576 の倍数 m) をメガバイト単位で指定します。MAX-mM は、使用できる記憶域の大きさを計算し、割り振ることをプログラムに指示します。割り振る大きさは、システムとアプリケーションが使用するために予約した記憶域 (RESALL) を MAX 値から 差し引いた 値が上限となります。

m に 0 メガバイトを指定すると 4 キロバイトが使用されます。

m の限界: 2 桁

**注:** SIZE=value の代わりに SIZE(value)、CORE=value、および CORE(value) の形式を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**SKIPREC**

▶—SKIPREC=z—▶

入力データ・セットの分類またはコピーを開始する前にスキップ (削除) したいレコード数 (z) を指定します。通常 SKIPREC は、前の DFSORT ジョブでの

## EXEC ステートメントの使用

| 未処理のレコードをバイパスするために使用されます。詳細については、181 ページの『OPTION 制御ステートメント』の SKIPREC オプションの説明を参照してください。

**z** スキップするレコードの数を指定します。

限界: 28 桁 (有効数字 15 桁)

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SOLRF または NOSOLRF



SOLRF インストール・オプションを一時的に指定変更して、DFSORT が、SORTOUT LRECL が不明のときに、SORTOUT LRECL を再フォーマット設定されたレコード長に設定するかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の SOLRF および NOSOLRF オプションの説明を参照してください。

#### SOLRF

DFSORT が、SORTOUT LRECL が指定されていないか使用可能でないときに、SORTOUT LRECL の再フォーマット設定されたレコード長を使用することを指定します。

#### NOSOLRF

DFSORT が、SORTOUT LRECL の再フォーマット設定されたレコード長を使用しないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SPANINC



SPANINC インストール・オプションを一時的に指定変更して、可変スパン入力データ・セット内に不完全なスパン・レコードが 1 つ以上検出されたときの、DFSORT の処置を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の SPANINC オプションの説明を参照してください。

#### RC0

DFSORT に対して、メッセージ ICE197I を (一度) 発行し、戻りコードを 0 に設定して、検出した不完全なスパン・レコードをすべて除去するように指定します。



**RC4**

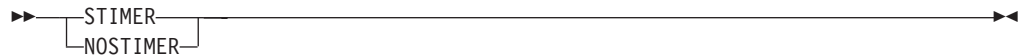
DFSORT に対して、メッセージ ICE197I を (一度) 発行し、戻りコードを 4 に設定して、検出した不完全なスパン・レコードをすべて除去するよう指定します。

**RC16**

DFSORT に対して、不完全なスパン・レコードを検出したら、メッセージ ICE204A を発行して終了し、戻りコード 16 を戻すよう指定します。

デフォルト: 通常は、インストール時のデフォルトです。697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**STIMER または NOSTIMER**

STIMER インストール・オプションを一時的に変更して、DFSORT が STIMER マクロを使用できるかどうかを指定します。

**STIMER**

STIMER が使用できることを指定します。プロセッサ時間データが SMF レコードおよび ICETEXIT 統計に表示されます。

**NOSTIMER**

STIMER が使用できないことを指定します。プロセッサ時間データは、SMF レコードまたは ICETEXIT 統計には表示されません。

注: ユーザー出口がチェックポイントをとる場合は、STIMER を発行できません。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**STOPAFT**

分類またはコピーに受け入れるレコードの最大数 (すなわち、SORTIN から読み取るか、または E15 が挿入するレコードで、しかも SKIPREC、E15、または INCLUDE/OMIT ステートメントが削除しないレコードの数) を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の STOPAFT オプションの説明を参照してください。

**n** 受け入れたいレコードの最大数を指定します。

## EXEC ステートメントの使用

限界: 28 桁 (有効数字 15 桁)

注: (1) EXEC PARM に FILSZ=x を指定するか、あるいは (2) OPTION または SORT ステートメントに SIZE=x または FILSZ=x を指定し、かつ処理するために受け入れられるレコードの数が x に等しくない場合、インストール時に FSZEST=YES が指定されていない場合は、DFSORT はエラー・メッセージを出して終了します。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SZERO または NOSZERO



SZERO インストール・オプションを一時的に指定変更して、DFSORT が照合、比較、編集、変換、最小値および最大値で、数字の -0 および +0 値を、符号付きとして (つまり、異なる値として) 処理するか、符号なしとして (つまり、同じ値として) 処理するかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の SZERO および NOSZERO オプションの説明を参照してください。

#### SZERO

DFSORT が数値ゼロを符号付きとして扱うことを指定します。

#### NOSZERO

DFSORT が数値ゼロを符号なしとして扱うことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### TRUNC



TRUNC インストール・オプションを一時的に指定変更して、DFSORT が LRECL 短縮を許可する場合に、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいときの、DFSORT の処置を指定します。詳細については、181 ページの『OPTION 制御ステートメント』の TRUNC オプションの説明を参照してください。

#### RC0

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいときは、メッセージ ICE171I を発行し、戻りコードを 0 に設定して処理を続行するよう指定します。

**RC4**

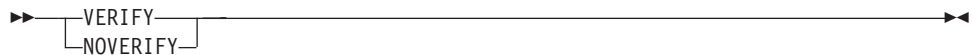
DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいときは、メッセージ ICE171I を発行し、戻りコードを 4 に設定して処理を続行するよう指定します。

**RC16**

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE196A を発行して終了し、戻りコード 16 を戻すよう指定します。

デフォルト: 通常は、インストール時のデフォルトです。697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**VERIFY または NOVERIFY**

VERIFY インストール・オプションを一時的に変更して、最終出力レコードのシーケンス検査を行なう必要があるかどうかを指定します。

**VERIFY**

シーケンス検査が行なわれることを指定します。

**NOVERIFY**

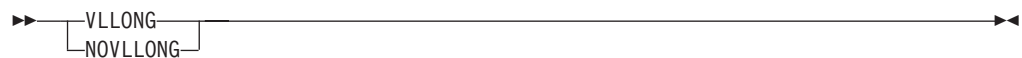
シーケンス検査が行なわれないことを指定します。

**注:**

1. VERIFY を使用すると、パフォーマンスを低下させることがあります。
2. VERIFY の代わりに SEQ=YES を使用できます。NOVERIFY の代わりに SEQ=NO を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**VLLONG または NOVLLONG**

VLLONG インストール・オプションを一時的に指定変更して、DFSORT が「長い」可変長出力レコードを切り捨てるかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の VLLONG および NOVLLONG オプションの説明を参照してください。

**VLLONG**

DFSORT が長い可変長出力レコードを、SORTOUT または OUTFIL データ・セットの LRECL に切り捨てることを指定します。

## EXEC ステートメントの使用

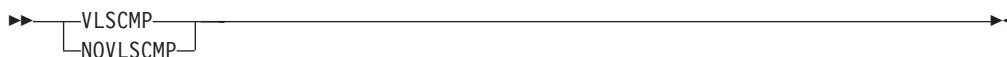
### NOVLLONG

長い可変長出力レコードが見つかった場合に、DFSORT が終了することを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VLSCMP または NOVLSCMP



VLSCMP インストール・オプションを一時的に指定変更して、DFSORT が「短い」可変長 INCLUDE/OMIT 比較フィールドを 2 進ゼロで埋め込むかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の VLSCMP および NOVLSCMP オプションの説明を参照してください。

#### VLSCMP

短い可変長比較フィールドが 2 進ゼロで埋め込まれることを指定します。

#### NOVLSCMP

短い可変長比較フィールドが埋め込まれないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VLSHRT または NOVLSHRT



VLSHRT インストール・オプションを一時的に指定変更して、「短い」可変長 SORT/MERGE 制御フィールド、INCLUDE/OMIT 比較フィールド、または SUM 要約フィールドが検出された場合に、DFSORT が処理を続行するかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の VLSHRT および NOVLSHRT オプションの説明を参照してください。

#### VLSHRT

短い制御フィールドまたは比較フィールドが検出された場合に、DFSORT が処理を続行することを指定します。

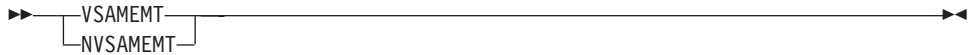
#### NOVLSHRT

短い制御フィールドまたは比較フィールドが検出された場合に、DFSORT が終了することを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VSAMEMT または NVSAMEMT



VSAMEMT インストール・オプションを一時的に指定変更して、DFSORT が空の VSAM 入力データ・セットを受け入れるかどうかを指定します。

#### VSAMEMT

DFSORT が空の VSAM 入力データ・セットを受け入れ、ゼロのレコードをもつとして処理することを指定します。

#### NVSAMEMT

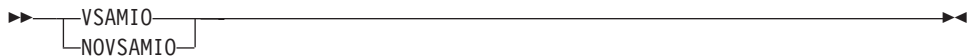
空の VSAM 入力データ・セットが検出された場合に、DFSORT が終了することを指定します。

注: VSAMEMT の代わりに VSAMEMT=YES を使用できます。NVSAMEMT に代わりに VSAMEMT=NO を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VSAMIO または NOVSAMIO



VSAMIO インストール・オプションを一時的に指定変更して、DFSORT が、REUSE で定義された VSAM データ・セットをインプレース分類で使用できるかどうかを指定します。

#### VSAMIO

以下のすべての条件に合致したとき、DFSORT が入力と出力に同じ VSAM データ・セットを使用できることを指定します。

- アプリケーションが分類である。
- RESET が有効。
- VSAM データ・セットが REUSE で定義された。

これらの条件により、VSAM データ・セットは確実に、出力には NEW として処理され、分類済みの入力レコードを含むこととなります。つまり、インプレースで分類されます。

DFSORT は、同じ VSAM データ・セットが入力と出力に指定されていて、上記の条件に合致しない場合、終了します。

## EXEC ステートメントの使用

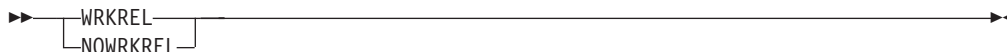
### NOVSAMIO

同じ VSAM データ・セットが入力と出力に指定されているとき、DFSORT が終了することを指定します。

デフォルト: 通常は、インストール時のデフォルトです。697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### WRKREL または NOWRKREL



WRKREL インストール・オプションを一時的に指定変更して、未使用の一時 SORTWKdd データ・セット空間を解放するかどうかを指定します。

#### WRKREL

使用されなくなったスペースを解放することを指定します。

#### NOWRKREL

使用しないスペースを解放しないことを指定します。

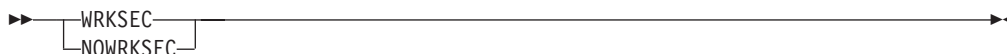
#### 注:

1. すでにいくつかのボリュームが SORTWKdd データ・セット専用になっていて、使用していない一時スペースを解放したくない場合は、NOWRKREL を指定する必要があります。
2. WRKREL が有効な場合、DFSORT は終了の直前に SORTWKdd データ・セット用のスペースを解放します。分類アプリケーションに使用した SORTWKdd データ・セットのスペースだけが解放されます。
3. NOWRKREL に代わりに RELEASE=OFF および RLS=0 を使用できます。WRKREL の代わりに RELEASE=ON および RLS=n (n は、0 より大) を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### WRKSEC または NOWRKSEC



WRKSEC インストール・オプションを一時的に指定変更して、DFSORT が一時 JCL SORTWKdd データ・セット用に自動 2 次割り振りを使用するかどうかを指定します。

**WRKSEC**

一時 JCL SORTWKdd データ・セットに自動 2 次割り振りを使用し、また 1 次割り振りの 25 % を 2 次割り振りとして使用するよう指定します。

**NOWRKSEC**

一時 JCL SORTWKdd データ・セットに自動 2 次割り振りを使用しないことを指定します。

**注:** NOWRKSEC の代わりに SECOND=OFF および SEC=0 を使用できます。  
WRKSEC の代わりに SECOND=ON および SEC=n (n は、0 より大) を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**Y2PAST**

Y2PAST インストール・オプションを一時的に指定変更して、スライド (s) または固定 (f) の「世紀」ウィンドウを指定します。「世紀」ウィンドウを、DFSORT の Y2 形式とともに使用すると、2 桁年のデータ値が 4 桁年のデータ値として、正しく解釈されます。

- s** スライド「世紀」ウィンドウの開始年を設定するために、DFSORT が現在の年から減算する年数を指定します。Y2PAST の値を現在の年から減算するため、現在の年が変わると「世紀」ウィンドウもスライドします。たとえば、Y2PAST=81 とすると、「世紀」ウィンドウは、1996 年には 1915-2014、1997 年には 1916-2015 となります。s の値は 0 から 100 の範囲で指定します。
- f** 固定「世紀」ウィンドウの開始年を指定します。たとえば、Y2PAST=1962 とすると、「世紀」ウィンドウは 1962-2061 になります。f の値は 1000 から 3000 の範囲で指定します。

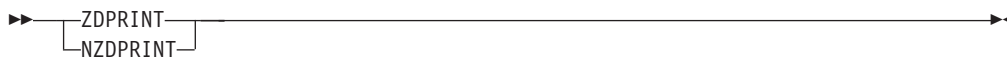
**注:** Y2PAST=value の代わりに CENTWIN=value を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**ZDPRINT または NZDPRINT**

## EXEC ステートメントの使用



ZDPRINT インストール・オプションを一時的に指定変更して、集計の結果生じた正のゾーン 10 進数 (ZD) フィールドを、印刷可能な数字に変換しなければならないかどうかを指定します。詳細については、181 ページの『OPTION 制御ステートメント』の ZDPRINT および NZDPRINT オプションの説明を参照してください。

### ZDPRINT

正の ZD 集計結果を印刷可能な数字に変換することを意味します。

### NZDPRINT

正の ZD 集計結果を印刷可能な数字に変換しないことを意味します。

注: ZDPRINT の代わりに ZDPRINT=YES を使用できます。NZDPRINT の代わりに ZDPRINT=NO を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## PARM オプションの別名

互換性の点から、下記のリストした別名を使用して、以下の EXEC/DFSPARM PARM オプションを指定できます。詳細については、記載された PARM オプションを参照してください。

表 4. PARM オプションの別名

別名	PARM オプション
CENTWIN=value	Y2PAST=value
CORE=value	SIZE=value
FLAG(I)	MSGPRT=ALL
FLAG(U)	MSGPRT=CRITICAL
L5=value	AVGRLLEN=value
MSG=value	MSGPRT=value
MSGDD=value	MSGDDN=value
NOFLAG	MSGPRT=NONE
NORC16	NOABEND
NORLSOUT	NOOUTREL
NOSDB	SDB=NO
RC16=ABE	ABEND
RELEASE=ON	WRKREL
RELEASE=OFF	NOWRKREL
RESERVE=value	RESALL=value



表 4. PARM オプションの別名 (続き)

別名	PARM オプション
RESERVEX=value	ARESALL=value
RLS=n	WRKREL
RLS=0	NOWRKREL
RLSOUT	OUTREL
SDB	SDB=YES
SDB=ON	SDB=YES
SDB=OFF	SDB=NO
SDB=SMALL	SDB=YES
SEC=n	WRKSEC
SEC=0	NOWRKSEC
SECOND=ON	WRKSEC
SECOND=OFF	NOWRKSEC
SEQ=YES	VERIFY
SEQ=NO	NOVERIFY
VSAMEMT=YES	VSAMEMT
VSAMEMT=NO	NVSAMEMT
ZDPRINT=YES	ZDPRINT
ZDPRINT=NO	NZDPRINT

## DD ステートメントの使用

DFSORT ジョブでは、EXEC ステートメントの後に必ず DD ステートメントが必要です。DD ステートメントは、次の 2 つのカテゴリに分類されます。

- システム DD ステートメント (70 ページの『システム DD ステートメント』に詳述)
- プログラム DD ステートメント (73 ページの『プログラム DD ステートメント』に詳述)

システム DD ステートメント、および一部のプログラム DD ステートメントは、通常、カタログ式プロシージャを使用すれば自動的に与えられます。その他の DD ステートメントはユーザー自身が指定します。

DD ステートメント・パラメーター、それが必要になる条件、およびデフォルトについては、68 ページの表 5 に要約しています。DCB パラメーターのサブパラメーター (DD ステートメント・パラメーター) については、同様に 69 ページの表 6 に説明しています。

### 注:

1. 可変長レコードに DCB の LRECL サブパラメーターを正確に指定すると、パフォーマンスが向上します。特定の構成について指定する入力レコードの最大長については、14 ページの『データ・セットの注意事項および制約事項』で説明しています。

## DD ステートメントの使用

2. DFSORT アプリケーションを使用する場合は、DD ステートメントで FREE=CLOSE を使用できません。ただし DFSPARM は除きます。

表 5. DFSORT で使用される DD ステートメント・パラメーター

パラメーター	必要な場合	パラメーター値	デフォルト
{AMPI BUFSP}	パスワードで保護された VSAM データ・セットを使用し、そのパスワードが E18、E38、または E39 で指定される場合。	データ・セットの作成時に指定される最小バッファ・プール値。	なし。
DCB	7トラック・テープを使用する場合 (標準ラベルをもたないテープを入力するため)、およびデフォルトが適用できない場合。	データ・セットに関連するデータ制御ブロック (DCB) を満たす情報を指定する。	(69 ページの表 6 の個々のサブパラメーターを参照)。
DISP	デフォルトが適用できない場合。	データ・セットの状況と処置を示す。	システムは (NEW, DELETE) と見なす。
DSNAME または DSN	DD ステートメントがラベル付き入力データ・セットを定義する場合 (たとえば、SORTIN)、または作成中のデータ・セットを保存またはカタログ (たとえば、SORTOUT)、または別のステップに渡す場合。	データ・セットの完全修飾名または一時名を指定する。	システムが固有な名前を指定する。
LABEL	デフォルトが適用できない場合。	データ・セットのラベル付けおよび保存に関する情報を指定する。	システムは標準ラベル付けと見なす。
SPACE	DD ステートメントが直接アクセス装置上に新しいデータ・セットを定義する場合。	データ・セットを入れるために必要なスペースの大きさを指定する。	なし。
UNIT	入力データ・セットがカタログもされず引き渡しも行われない場合、またはデータ・セットが作成される場合。	データ・セットが必要とする入出力装置のタイプと数を (記号で、または具体的に) 指定する。	なし。
VOLUME または VOL	入力データ・セットがカタログもされず引き渡しも行われない場合、入力が複数リールの場合、または出力データ・セットが直接アクセス装置上にあり、しかも保存またはカタログされる場合。	データ・セットが占有するボリューム (1 つ以上) の識別に使用する情報を指定する。	なし。

表 6. DFSORT で使用される DCB サブパラメーター

サブパラメーター	必要時の条件	サブパラメーター値	デフォルト
BUFOFF	ISCI/ASCII 形式でデータを処理する場合。	バッファ・オフセットの長さを指定、またはバッファ・オフセットがブロック長標識であることを指定する。	
DEN	データ・セットが 7 トラック・テープ装置上にある場合。	テープが記録された密度を指定する。	800 bpi (ビット / インチ)。
OPTCD	ISCI/ASCII 形式でデータを処理する場合。	処理されるテープが ISCI/ASCII 形式であることを指定する。	
TRTCH	データ・セットが IDRC 付きの磁気テープ装置にあり、かつシステム IDRC を使用しない場合。	データ・セットを短縮するかどうかを指定する。	システム・デフォルト・オプション。
BLKSIZE <sup>1, 2</sup>	DCB パラメーターが必要であり、デフォルト値が、SORTWKdd ステートメント以外で適用できない場合。	データ・セット内の物理レコードの最大長 (バイト単位) を指定する。	<ul style="list-style-type: none"> <li>古いデータ・セットの場合、データ・セット・ラベル内の値。</li> <li>新しい出力データ・セットの場合、入力データ・セットと RECORD ステートメント値に基づく適切な値。</li> </ul> <p>SDB=NO が有効でない限り、出力データ・セット・ブロック・サイズがゼロであれば、ブロック・セットは、システム決定の最適ブロック・サイズを使用します。</p> <p>アプリケーションが特定の出力データ・セット・ブロック・サイズを必要とする場合は、そのブロック・サイズを変更して <b>EXPLICITLY</b> に指定しなければならない。</p> <ul style="list-style-type: none"> <li>ラベルなしテープまたは BLP または NSL 上の入力が指定された場合は、デフォルトなし。</li> </ul>
LRECL <sup>2, 3</sup>		データ・セット内の論理レコードの最大長 (バイト単位) を指定する。	
RECFM		データ・セット内のレコードの形式を指定する。	

1. 75 ページの『SORTIN DD ステートメント』および 77 ページの『SORTINnn DD ステートメント』を参照してください。

2. DD \* データ・セットには、このサブパラメーターだけしか使用できません。

3. 固定長レコードの埋め込みまたは切り捨てについては、14 ページの『データ・セットの注意事項および制約事項』を参照してください。

## DD ステートメントの使用

### DD 名の重複

同一ステップ内で特定の DD 名 (たとえば、SORTIN) を 2 回以上指定した場合、DFSORT は最初の DD 名を使用し、その後の重複 DD 名は無視します。処理は正常に続けられます。

さらに、SORTIN00、SORTIN01...SORTIN09 の代わりに、それぞれ SORTIN0、SORTIN1...SORTIN9 を指定できます。SORTIN<sub>n</sub> と SORTIN0<sub>n</sub> を両方同じジョブ・ステップで指定した場合、DFSORT はこれらを重複 DD 名として取り扱い、使用している最初のもの以外はすべて無視します。たとえば、SORTIN2 および SORTIN02 は重複として取り扱われ、SORTIN2 だけが使用されます。

**注:** 従来の組み合わせの場合、SORTIN01、SORTIN02...SORTIN16 しか使用できないという制約があるために、SORTIN<sub>n</sub> は認識されません。これらの受け入れられる DD 名が重複して使用されている場合は無視されます。

307 ページの『OUTFIL ステートメントに関する注意事項』に説明されているとおり、OUTFIL ステートメント・レベルでは OUTFIL DD 名は無視されます。

### 磁気テープ装置の共有

以下の DFSORT データ・セットのペアを、1 つの磁気テープ装置に割り当てることができます。

- SORTIN データ・セットと SORTWK01 データ・セット (テープ作業データ・セットは分類のみ)
- SORTIN データ・セットと SORTOUT データ・セットまたは OUTFIL データ・セット (分類アプリケーションのみ)

SORTIN データ・セットと SORTWK01 を関連付けたい場合は、SORTWK01 についての DD ステートメントにパラメーター UNIT=AFF=SORTIN を組み込みます。AFF サブパラメーターを指定すると、システムはそのデータ・セットを、サブパラメーターに続く DD 名 (この場合は、SORTIN) をもつデータ・セットと同じ装置上に置きます。

同様に、SORTOUT または OUTFIL DD ステートメントで UNIT=AFF=SORTIN を指定すると、SORTIN データ・セットを SORTOUT データ・セットまたは OUTFIL データ・セットと関連付けることができます。

SORTIN<sub>nn</sub> テープ・データ・セットは同時に読み取られるので、それらはすべて別々の磁気テープ装置に入れておく必要があります。SORTOUT と OUTFIL テープ・データ・セットは同時に書き込まれるので、すべて別々の磁気テープ装置に入れておく必要があります。

### システム DD ステートメント

DFSORT の呼び出しに SORT または SORTD カタログ式プロシージャーを使用しない場合は、入力ジョブ・ストリームにシステム DD ステートメントを指定しなければならないことがあります (DFSORT 専用の DD ステートメント、たとえば SORTIN については以下の項を参照してください)。カタログ式プロシージャーに含まれる (またはユーザーが指定する) DD ステートメントは、次のとおりです。

**//JOB LIB DD**

ユーザーのプログラム・リンク・ライブラリーがまだシステムに知らされていない場合、そのライブラリーを定義します。

**//STEPLIB DD**

//JOB LIB DD と同じ。

**//SYSIN DD**

DFSORT が別のプログラムにより呼び出されるのではなく、JCL により呼び出される場合、DFSORT 制御ステートメント、注釈ステートメント、ブランク・ステートメント、および注記を含みます。また、DFSORT がリンク・エディットするオブジェクト・デッキ形式のユーザー出口ルーチンを含めることもできます。

- DFSPARM を使用する場合、ユーザーのジョブがリンク・エディットを必要としない限り SYSIN は不要です。
- SYSIN 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、順次データ・セットまたは区分データ・セットのメンバーとして定義することもできます。
- データ・セットは、F または FB の RECFM で定義します。LRECL は、80、またはそれ以上 (有効なとき) の値を指定できます。LRECL が 80 より大きい場合、DFSORT は各レコードの最初の 80 バイトを使用します。  
ユーザー出口ルーチンが SYSIN にある場合は、LRECL は 80 である必要があります。
- DFSORT は、システムが BSAM の「類似した」連結データ・セットをサポートする程度に SYSIN データ・セットをサポートします。「類似した」連結データ・セットについては、*z/OS DFSMS: Using Data Sets* を参照してください。

**注:** OPTION ステートメントのキーワードの EFS、LIST、NOLIST、LISTX、NOLISTX、LOCALE、MSGPRT、MSGDDN、SMF、SORTDD、SORTIN、および SORTOUT は、拡張パラメーター・リスト、または DFSPARM データ・セット内に入れて渡される場合のみ使用します。それらが、SYSIN または SORTCNTL データ・セットから読み込まれた OPTION ステートメントに指定されている場合、キーワードは認識されますが、パラメーターは無視されます。

代わりに DFSPARM DD ステートメントを使用すると、他のすべてのソースを変更する単一のソース・データ・セットに、EXEC PARM オプションと DFSORT 制御ステートメントの両方を指定できます。85 ページの『DFSPARM DD ステートメント』を参照してください。

ユーザー出口ルーチンが SYSIN にある場合は、次のことを確認してください。

- SYSIN の LRECL が 80 であること。
- END ステートメントが最後の 制御 ステートメントであること。
- ユーザー出口ルーチンが数字順に並んでいること (たとえば、E11 は E15 の前)。

## DD ステートメントの使用

- ユーザー出口ルーチンが END 制御ステートメントの直後に指定されていること。
- SYSIN の最後のオブジェクト・デックの後には何もないこと。
- SORTMODS DD ステートメントが含まれていること。

DFSORT がプログラムで呼び出され、24 ビットまたは拡張パラメーター・リストの SORTCNTL または DFSPARM から DFSORT 制御ステートメントを指定した場合でも、SYSIN は引き続き、システム入力ストリームに入れられたユーザー出口ルーチンのソースのままです。

### //SYSOUT DD

DFSORT メッセージ・データ・セットを示します。デフォルト DD 名は SYSOUT ですが、MSGDDN インストール時または実行時オプションを使用して、メッセージ・データ・セットに別の DD 名を指定することもできます。カタログ・プロシージャを使用しない場合は常に、メッセージ・データ・セットに DD ステートメントを指定してください。(COBOL プログラムから DFSORT を呼び出し、メッセージ・データ・セットに SYSOUT の DD 名を使用する場合、COBOL プログラムで EXHIBIT または DISPLAY を使用すると、不確かな印刷結果になることがあります。)

DFSORT は、RECFM=FBA、LRECL=121、およびメッセージ・データ・セットに指定された BLKSIZE を使用します。指定した BLKSIZE が 121 の倍数でない場合は、DFSORT は BLKSIZE=121 を使用します。BLKSIZE を指定しないと、DFSORT は SDBMSG インストール・オプションで指定されているブロック・サイズを選択します (*DFSORT 導入とカスタマイズ リリース 14* を参照してください)。

一時または永久メッセージ・データ・セットを使用するときは、MOD の後処置を指定して、メッセージ・データ・セット内のすべてのメッセージおよび制御ステートメントを確実に見ることができるようにするのが、最もよい方法です。

### //SYSUDUMP DD

DD システム ABEND ダンプ・ルーチンからの出力用のデータ・セットを定義します。

### //SYSMDUMP DD

//SYSUDUMP DD と同じ。

### //SYSABEND DD

//SYSUDUMP DD と同じ。

IBM 提供の SORT カタログ式プロシージャを使用する場合は、下記のステートメントが自動的に提供されます。SORT カタログ式プロシージャを使用せずにリンケージ・エディターを使用している場合は、次の DD ステートメントを提供します。

### //SYSPRINT DD

リンケージ・エディターからのメッセージが含まれます。

### //SYSUT1 DD

リンケージ・エディター用の中間記憶域データ・セットを定義します。

**//SYSLIN DD**

リンケージ・エディター用の制御情報データ・セットを定義します。

**//SYSLMOD DD**

リンケージ・エディターからの出力用データ・セットを定義します。

**注:** ユーザー・ルーチンを組み込まない場合、またはリンク・エディットを必要としないユーザー・ルーチンを組み込む場合は、IBM 提供の SORTD カタログ式プロシージャを使用できます。リンク・エディットを必要とするユーザー・ルーチンを組み込む場合は、SORT カタログ式プロシージャを使用できます。

## プログラム DD ステートメント

SORT または SORTD カタログ式プロシージャを使用して DFSORT を呼び出す場合でも、専用 DD ステートメントを追加指定しなければならないことがあります。次に、これらのステートメントを要約したリストを示し、その後で個々のステートメントについて、さらに詳しく説明します。

**//SORTLIB DD**

DFSORT の特殊なロード・モジュールを入れるデータ・セットを定義します。通常は省略できます。

**//SYMNAMES DD**

シンボルの処理に使用するステートメントを入れる、SYMNAMES データ・セットを定義します。シンボルが処理される場合のみ、この定義が必要です。

**//SYMNOUT DD**

SYMNAMES ステートメントとシンボルのテーブルをリストするデータ・セットを定義します。SYMNAMES DD が指定されている場合は、オプションです。それ以外は、無視されます。

**//SORTIN DD**

分類アプリケーションまたはコピー・アプリケーションの入力データ・セットを定義します。組み合わせアプリケーションには使用されません。

**//SORTINnn DD**

組み合わせアプリケーションの入力データ・セットを定義します。分類アプリケーションまたはコピー・アプリケーションには使用されません。

**//SORTWKdd DD**

中間記憶データ・セットを定義します。動的割り振りが要求されない限り、通常、分類アプリケーションに必要です。コピーまたは組み合わせアプリケーションには使用されません。

**//SORTOUT DD**

分類アプリケーション、組み合わせアプリケーション、またはコピー・アプリケーションの出力データ・セット、SORTOUT を定義します。

**//outfil DD**

分類アプリケーション、組み合わせアプリケーション、またはコピー・アプリケーションの出力データ・セット、OUTFIL を定義します。

## DD ステートメントの使用

### //SORTCKPT DD

最後のチェックポイントから分類を再始動するためにシステムが必要とする情報を保管するために使用するデータ・セットを定義します。チェックポイント機能を使用する場合のみ必要です。

### //SORTCNTL DD

DFSORT がプログラムで呼び出される場合に、追加または変更済み DFSORT 制御ステートメントを読み取るデータ・セットを定義します。

### //DFSPARM DD

DFSORT が直接呼び出される場合、またはプログラムで呼び出される場合は、追加または変更済み DFSORT プログラム制御ステートメントと EXEC ステートメントの PARM オプションの両方を読み取るデータ・セットを定義します。

### //SORTDKdd DD

動的に再割り振りされる場合、DFSORT が VIO SORTWKdd 割り振りに使用するデータ・セットを定義します。SORTDKdd をジョブ・ストリームで指定できません。

### //SORTDIAG DD

すべてのメッセージおよび制御ステートメントを印刷することを指定します。診断およびデバッグの場合に使用されます。

### //SORTSNAP DD

DFSORT が動的に割り振るスナップ・ダンプ・データ・セットを定義します。SORTSNAP をジョブ・ストリーム内で指定できません。

### //SORTMODS DD

一時区分データ・セットを定義します。この一時データ・セットは、アプリケーション用の SYSIN 内にあるすべてのユーザー・ルーチンを入れるために十分な大きさである必要があります。SYSIN 内にユーザー・ルーチンがない場合は、このステートメントは不要です。ユーザー・ルーチンがライブラリー内にある場合、そのライブラリーを定義する DD ステートメントを指定する必要があります。

DFSORT は、SYSIN 内のユーザー・ルーチンをリンク・エディットまたは処理する前に、この DD ステートメントで定義されたデータ・セットへ一時的に転送します。

## SORTLIB DD ステートメント

通常は、SORTLIB DD ステートメントは省略できます。このステートメントは特殊な DFSORT ロード・モジュールを入れるデータ・セットを記述するものです。

**必要となる場合:** ICEMAC オプションの SORTLIB=PRIVATE が有効であるか、またはユーザー出口のダイナミック・リンク編集が指定されていて、次の条件に合う場合、このステートメントが必要となります。

- テープ作業データ・セットを使用する分類アプリケーションの場合
- ブロック・セットが使用できない組み合わせアプリケーションの場合 (メッセージ ICE800I を参照)



ICEMAC SORTLIB オプションは、テープ作業データ・セット分類または従来の組み合わせに必要なロード・モジュールを、DFSORT がシステム・ライブラリーまたは専用ライブラリーで探すかどうかを決めます。

### 例 1 SORTLIB DD ステートメント:

```
//SORTLIB DD DSN=USORTLIB,DISP=SHR
```

この例は、すでにカタログされている入力データ・セットを定義する DD ステートメント・パラメーターを示しています。

#### DSNAME

これにより、システムは USORTLIB という名前のデータ・セットをカタログの中から探します。データ・セットが見つかったら、SORTLIB という DD 名と関連付けられます。制御プログラムはカタログから装置割り当てとボリューム通し番号を入手し、そのボリュームがまだ取り付けられていなければ、取り付け指示メッセージをオペレーターに書き出します。

#### DISP

データ・セットがこのジョブ・ステップの前に存在していたこと、このジョブ・ステップの後でも保存しておく必要があること、そして複数ジョブで同時に使用できること (SHR) を示します。どのジョブもこのデータ・セットを変更できません。

SORTLIB DD ステートメントで使用するパラメーター、それらが必要になる条件、およびパラメーターを指定しなかったときのデフォルトについては、68 ページの表 5 を参照してください。DCB パラメーターのサブパラメーターについては、同様に 69 ページの表 6 を参照してください。詳細については、*z/OS MVS JCL 解説書*および *z/OS MVS JCL ユーザーズ・ガイド* を参照してください。

### SYMNAMES DD ステートメントと SYMNOUT DD ステートメント

詳細については、555 ページの『第 7 章 フィールドと定数のシンボルの使用』を参照してください。

### SORTIN DD ステートメント

SORTIN DD ステートメントは、分類またはコピーしたいレコードが含まれるデータ・セットの特性を記述し、またその位置を示します。

**必要となる場合:** SORTIN DD ステートメントは、すべての分類アプリケーションまたはコピー・アプリケーションに必要です。ただし、DFSORT へすべての入力を与える E15 ユーザー出口を設定し、かつプログラム制御ステートメントに RECORD を指定した場合を除きます。ユーザー・プログラムが DFSORT を呼び出し、E15 ユーザー出口のアドレスをパラメーター・リストで渡す場合は、SORTIN DD ステートメントは無視されます。

**データ・セットの特性:** 分類処理またはコピー処理の場合、DFSORT は空またはヌルの非 VSAM データ・セットを受け入れます (DCB パラメーターを必ず指定してください)。VSAMEMT が有効なとき、分類またはコピー処理について、DFSORT は空の VSAM データ・セットも受け入れます。非 VSAM データ・セットの場合、DFSORT は形式 1 DSCB の DSILSTAR フィールドを調べて、データ・セットが空またはヌルであるかどうかを判別します。DSILSTAR がゼロの場合は、DFSORT

## DD ステートメントの使用

はそのデータ・セットを空またはヌルとして取り扱います。データ・セットがヌルのマルチボリューム・データ・セットであり、そのマルチボリューム・データ・セットの最初のボリュームの形式 1 DSCB の DS1IND80 フラグがオフの場合、DFSORT は出力のためにそのデータ・セットをオープンし、そのデータ・セットを入力に使用する前に強制的にファイルの終わり (EOF) マークを付けます。

ヌルのデータ・セットとは、新しく作成されたにもかかわらず、正しくクローズされなかったデータ・セットのことです。テープ作業データ・セットの分類では、ヌルのデータ・セットは正しく処理できません。DASD ボリューム目録 (VTOC 内の DSCB) のデータ・セット・ラベルの「システム・コード」は、その中に DOS または VSE の文字が入っている場合は、VSE オペレーティング・システムが作成するデータ・セットを示します。そのようなデータ・セットはヌルとして扱われることはありませんが、空である場合があります。VSE DASD データ・セットがシステム・コード・フィールド内に DOS または VSE をもたない場合、DFSORT はそのデータ・セットを処理することができません。

考慮事項について詳しくは、13 ページの『データ・セットに関する考慮事項』を参照してください。

連結データ・セットには、次の規則が適用されます。

- 連結されているデータ・セットについて、RECFM はすべて固定長であるか、あるいはすべて可変長のいずれかである必要があります。
- BLKSIZE は可変です。ただし、テープ・データ・セットが最大のブロック・サイズをもち、連結内の最初のデータ・セットでない場合、以下の 2 つの状態では、BLKSIZE を DD ステートメントに明示的に指定する必要があります。
  - ブロック・セットが選択されていて、テープ・データ・セットのブロック・サイズは 32760 バイトより大きい、DFSMSrmm または ICETPEX で使用できない。
  - ブロック・セットが選択されていない。
- 固定長レコードの場合は、LRECL はすべてのデータ・セットについて同一である必要があります。可変長レコードの場合は、LRECL は可変ですが、最初のデータ・セットは最大 LRECL である必要があります。
- データ・セットが非類似装置上にある場合は、ユーザー出口 18 で EXLST パラメーターを使用できません。
- ブロック・セットが選択されずに BSAM が使用されている場合は、すべてのヌル・データ・セットは、ヌル以外のデータ・セットの前に置く必要があります。そうしないと、予期しない結果が生じることがあります。
- DFSORT は、ヌル・データ・セットの形式 1 の DSCB DS1IND80 フラグがオフの場合、BSAM を使用してそのヌル・データ・セットを処理する前に、データ・セットすべてに強制的に EOF マークを付けます。
- DUMMY パラメーターを使用してデータ・セットを定義する場合、そのデータ・セットに他のデータ・セットを連結できません。システムは、DUMMY に連結されたデータ・セットを無視します。
- VSAM データ・セットを連結できません (システム制限)。
- 入力に VSAM および 非 VSAM データ・セットの両方を含めることはできません。

**一般的なコーディングの規則:**

- コピー・アプリケーションにおいては、SORTIN データ・セットは SORTOUT データ・セットまたは OUTFIL データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。
- 分類アプリケーションでは、SORTIN データ・セットは、SORTWKdd データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。SORTIN データ・セットは、SORTOUT データ・セットまたは OUTFIL データ・セットと同じでもよい場合があります。しかし、この場合、分類アプリケーションが正常に終了しない場合はデータ・セットが失われる結果を招くことがあります。
- FREE=CLOSE を指定できません。ユーザー・ラベルはコピーされません。

**例 2 SORTIN DD ステートメント:**

```
//SORTIN DD DSN=INPUT,DISP=SHR
```

この例は、すでにカタログされている入力データ・セットを定義する DD ステートメント・パラメーターを示しています。

**DSNAME**

これにより、システムは INPUT という名前のデータ・セットをカタログの中から探します。データ・セットが見つかったら、SORTIN という DD 名と関連付けられます。制御プログラムはカタログから装置割り当てとボリューム通し番号を入手し、そのボリュームがまだ取り付けられていなければ、取り付け指示メッセージをオペレーターに書き出します。

**DISP**

データ・セットがこのジョブ・ステップの前に存在していたこと、このジョブ・ステップの後でも保存しておく必要があること、そして複数ジョブで同時に使用できること (SHR) を示します。どのジョブもこのデータ・セットを変更できません。

**例 3 SORTIN DD のボリューム・パラメーター:**

```
//SORTIN DD DSN=SORTIN,DISP=(OLD,KEEP),UNIT=3490,  
// VOL=SER=(75836,79661,72945)
```

入力データ・セットが複数リールの磁気テープに含まれる場合、SORTIN DD ステートメントに VOLUME パラメーターを指定し、テープ・リールの通し番号を示す必要があります。この例では、入力データ・セットは通し番号が 75836、79661、および 72945 の 3 つのリール上にあります。

データ・セットがディスクまたは標準ラベル付きテープ上にない場合は、その DD ステートメントに DCB パラメーターを指定します。

**SORTINnn DD ステートメント**

SORTINnn DD ステートメントは、組み合わせを行なうレコードが入っているデータ・セットの特性を記述し、これらのデータ・セットの位置を示します。

## DD ステートメントの使用

**必要となる場合:** 組み合わせには SORTINnn DD ステートメントが必須です。ただし、組み合わせプログラムが別のプログラムから呼び出され、すべての入力ユーザー出口 E32 のルーチンから与えられる場合を除きます。

**データ・セットの特性:** 入力データ・セットには、非 VSAM または VSAM のどちらか一方のみ使用できます。組み合わせ処理の場合、DFSORT は空またはヌルの非 VSAM データ・セットを受け入れます (DCB パラメーターを必ず指定してください)。VSAMMEMT が有効なとき、組み合わせ処理について、DFSORT は空の VSAM データ・セットも受け入れます。非 VSAM データ・セットの場合、DFSORT は形式 1 DSCB の DSILSTAR フィールドを調べて、データ・セットがヌルまたは空であるかどうかを判別します。DSILSTAR がゼロの場合は、DFSORT はそのデータ・セットをヌルまたは空として取り扱います。ヌル・データ・セットとは、新しく作成されたにもかかわらず正しくクローズされなかったデータ・セットを意味します。従来の組み合わせ手法では、ヌル・データ・セットを正しく処理できません。

すべての入力データ・セットについて RECFM は同一である必要があります。

BLKSIZE は同じである必要はありませんが、従来の組み合わせ処理では、SORTIN01 には最大ブロック・サイズを指定します。

固定長レコードの場合は、LRECL はすべてのデータ・セットについて、同じである必要があります。可変長レコードの場合、LRECL は同じである必要はありません。

データ・セットはマルチボリュームにできますが、連結できません。SORTINnn データ・セットがマルチボリュームで、かつヌルの場合、DFSORT は使用の前にそのデータ・セットに EOF マークを強制的に付けます。

考慮事項について詳しくは、14 ページの『データ・セットの注意事項および制約事項』を参照してください。

### 一般的なコーディングの規則:

- SORTINnn データ・セットは、SORTOUT データ・セットまたは OUTFIL データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータになったり、予期せぬ結果を招くことがあります。
- ブロック・セットによる組み合わせなら最大 100 まで、従来の組み合わせなら最大 16 までのデータ・セットを組み合わせることができません。従来の組み合わせが選択される場合、ブロック・セットを使用できなかった理由をメッセージの ICE800I で調べ、可能であれば、示されている状況を訂正してください。
  - ブロック・セット組み合わせを使用する場合、nn には 00 (最初の 0 は任意指定) ~ 99 の整数を任意の順序で指定できます。ブロック・セット組み合わせは、SORTINn と SORTIN0n の形式の DD 名を重複する DD 名として取り扱い、最初の DD 名以外は何回現れても無視します。たとえば、次のように指定すると仮定します。

```
//SORTIN4 DD . . .  
//SORTIN04 DD . . .
```

この場合 SORTIN04 DD は無視されます。

- 従来の組み合わせを使用する場合、nn は 01 ~ 16 の範囲で指定できます。ただし、最初の番号には 01 を使用しなければならず、残りの番号は数字の順序どおりにならなければならず、数字をスキップすることもできません。従来の組み合わせで SORTIN0-SORTIN9、SORTIN00 または SORTIN17-SORTIN99 のような形式の DD 名を使用できません。

- FREE=CLOSE を指定できません。ユーザー・ラベルはコピーされません。

#### 例 4 SORTIN01 ~ 03 DD ステートメント (組み合わせ):

```
//SORTIN01 DD  DSNAME=MERGE1,VOLUME=SER=000111,DISP=OLD,
//              LABEL=(,NL),UNIT=3590,
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//SORTIN02 DD  DSNAME=MERGE2,VOLUME=SER=000121,DISP=OLD,
//              LABEL=(,NL),UNIT=3590,
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//SORTIN03 DD  DSNAME=MERGE3,VOLUME=SER=000131,DISP=OLD,
//              LABEL=(,NL),UNIT=3590,
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
```

#### 例 5 SORTIN01 ~ 02 DD ステートメント (組み合わせ):

```
//SORTIN01 DD  DSNAME=INPUT1,VOLUME=SER=000101, *
//              UNIT=3390,DISP=OLD                *DCB PARAMETERS
//SORTIN02 DD  DSNAME=INPUT2,VOLUME=SER=000201, *SUPPLIED FROM
//              UNIT=3390,DISP=OLD                *LABELS
```

### SORTWKdd DD ステートメント

SORTWKdd DD ステートメントは、分類するレコードの中間記憶域に使用するデータ・セットの特性を記述します。また、この特性はデータ・セットの位置も示します。

最大 255 個までの SORTWKdd DD ステートメントを指定できます。ただし、32 個以上を指定しても、ブロック・セット手法を選択しないと、最初の 32 個のみが使用されます。

**必要となる場合:** 分類アプリケーションごとに、1 つ以上の SORTWKdd ステートメントが必要です (組み合わせまたはコピーの場合は不要)。ただし、次の場合を除きます。

- 主記憶域に入力を収容できる場合
- 動的ワークスペース割り振りが要求された場合 (DYNALLOC)
- ハイパー分類または データ空間分類が使用される場合

作業データ・セットの使用方法については、685 ページの『付録 A. ワークスペースの使用』を参照してください。

診断メッセージ ICE803I は、中間記憶域の割り振りと使用に関する情報を提供します。

**装置:** SORTWKdd データ・セットは DASD (直接アクセス記憶装置) またはテープのどちらか一方にのみ、存在できます。DASD は各種のタイプを混用できます。

## DD ステートメントの使用

入力が 7 トラック・テープでない限り、テープは 9 トラックである必要があります。7 トラックの場合は作業テープを 7 トラック にすることができますが、必須ではありません。

### 一般的なコーディングの規則:

- 入力ファイルが非常に大きい場合を除き、通常は 1 つまたは 2 つの SORTWKdd データ・セットだけで十分です。小さなデータ・セットをいくつか使用するより、大きな SORTWKdd データ・セットを 1 つか 2 つを使用することをお勧めします。SORTWKdd データ・セットは、それぞれ別の装置に配置するほうがパフォーマンスを向上させることができます。  
仮想記憶域などのリソースを最適に割り振るために、作業データ・セットの数は必要以上に多く指定しないでください。
- SORTWKdd データ・セットは、SORTIN データ・セット、SORTOUT データ・セット、OUTFIL データ・セット、または SORTWKdd データ・セットのいずれとも異なる必要があります。同じである場合、データが失われるか、データが誤ったものになるか、あるいは予期せぬ結果となる可能性があります。
- 性能上の理由から、シリンダー割り振りが望まれます。一時的にトラックまたはブロックで割り振られた SORTWKdd データ・セット (ROUND なし) は、DFSORT がシリンダーに再調整します。
- DASD 作業データ・セットには、SORTWKdd または SORTWKd の形式の有効な DD 名が使用できます (たとえば、SORTWK01、SORTWKC3、SORTWK2、SORTWK#5、SORTWKA、SORTWKXY など)。DD 名は任意の順序で指定できます。SORTWKd と SORTWK0d は、重複 DD 名 (たとえば、SORTWK5 と SORTWK05 を指定すると、両方使用でき、また SORTWKQ と SORTWK0Q も同様) としては扱われません。ただし、255 以上の DD 名を指定して、ブロック・セット手法を選択すると、最初の 255 だけが使用されます。32 以上の DD 名を指定して、ブロック・セット手法を選択しないと、最初の 32 の DD 名だけが使用されます。
- テープ作業データ・セットの場合、最低 3 つの SORTWKdd データ・セットを指定する必要があります。その最初の 3 つの DD 名は、SORTWK01、SORTWK02、SORTWK03 である必要があります。この 3 つの DD 名に続く DD 名は、指定する場合、SORTWK04 から SORTWK32 の順になっている必要があります。数字を飛ばすことはできません。
- FREE=CLOSE を指定できません。
- スプール、ダミー、パイプ、および HFS のデータ・セット、および HFS ファイルは、作業データ・セットとして指定してはいけません。
- ISCI/ASCII データに関連するパラメーターを指定できません。

### DASD 作業データ・セットのコーディング上の注意:

- データ・セットは物理的に連続している必要があります。分割または拡張された形式では使用できません。
- SPLIT シリンダー・パラメーターを指定できません。
- 一時 SORTWKdd データ・セットに 2 次割り振りが要求されない場合、NOWRKSEC が有効な場合を除き、自動的に 2 次割り振りが使用されます。(ピアレッジ / ベール分類手法の場合のみ、2 次割り振りは 12 個の作業データ・セットに制限されます。)

## DD ステートメントの使用

- データ・セットを VIO に割り振ると、自動 2 次割り振りは行なわれません。
- 作業データ・セットに 2 次割り振りを要求できます。多くの作業データ・セットが定義されている場合、それらは 1 次割り振りだけが使用されます。(ピアレッジ / ベール分類手法の場合のみ、2 次割り振りは 12 個の作業データ・セットに制限されます。)
- DFSORT は、マルチボリューム・データ・セット用に指定された最初のボリュームのスペースのみを使用します。2 番目以降のボリュームのスペースは使用されません。このため、マルチボリュームの SORTWKdd データ・セットは単一ボリュームの SORTWKdd データ・セットとして取り扱われます。
- 1 次スペースが断片化されると、最初のもの以外はすべて 2 次スペースとして扱われます。

**仮想入出力:** 仮想記憶装置で SORTWKdd データ・セットを指定すると、次のようになります。

- VIO=NO を指定している場合、DFSORT は、その仮想装置と同じタイプの実装置の ddname、SORTDKdd を使って動的再割り振りを行ないません。仮想装置に対応する実装置が利用できない場合は、DFSORT はメッセージ ICE083A を出して終了します。このエラーの詳細については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。VIO SORTWKdd データ・セットが存在している場合は、VIO SORTWKdd データ・セット以外のデータ・セットも再割り振りされます。
- VIO=YES が指定されている場合、仮想記憶装置が使用され、パフォーマンスが低下します。

次の例は、DASD 作業データ・セットを使用した SORTWKdd DD ステートメントを示したものです。

### 例 6 SORTWK01 DD ステートメント、DASD 作業データ・セット:

```
//SORTWK01 DD SPACE=(CYL,(15,5)),UNIT=3390
```

チェックポイント / 再始動機能を使用していて、据え置き再始動を行なう必要がある場合は、分類作業データ・セットが失われないように、上記のステートメントに次のパラメーターを付け加えておく必要があります。

```
DSNAME=name1,DISP=(NEW,DELETE,CATLG)
```

したがって、据え置き再始動の場合は、この SORTWKdd DD ステートメントは次のようになります。

```
//SORTWK01 DD DSNAME=name1,UNIT=3390,SPACE=(CYL,(15,5)),  
// DISP=(NEW,DELETE,CATLG)
```

次の例は、3 つのテープ装置を使用する SORTWKdd DD ステートメントを示したものです。

## DD ステートメントの使用

### 例 7 SORTWK01 ~ 03 DD ステートメント、テープ中間記憶装置:

```
//SORTWK01 DD UNIT=3480,LABEL=(,NL)
//SORTWK02 DD UNIT=3480,LABEL=(,NL)
//SORTWK03 DD UNIT=3480,LABEL=(,NL)
```

DFSORT が正しく終了しなかった場合に上記の DD ステートメントが指定されていると、ステップが正しく再実行されるか、あるいは何らかの方法でデータ・セットが削除されるまで、中間記憶データ・セットはシステム内に残ったままになります。

これらのパラメーターは、3 つの 3480 テープ装置上の、ラベルなしデータ・セットを指定します。DSNAME パラメーターが省略されているため、システムが固有な名前を割り当てます。

### SORTOUT および OUTFIL DD ステートメント

SORTOUT および OUTFIL DD ステートメントは、処理済みレコードを入れるデータ・セットの特性を記述し、その位置を示します。

SORTOUT DD ステートメントは、分類、コピー、または組み合わせアプリケーションのための、非 OUTFIL 単一出力データ・セットを指定します。OUTFIL 処理は、SORTOUT には適用されません。

1 つ以上の OUTFIL ステートメントの FNAMES または FILES (またはその両方の) パラメーターは、分類、コピー、または組み合わせアプリケーションの OUTFIL データ・セットの DD 名を指定します。各 OUTFIL ステートメントごとに指定されたパラメーターは、そのステートメントに関連する OUTFIL データ・セットについて OUTFIL 処理が実施されるように定義します。OUTFIL 処理に関する特定の情報については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

SORTOUT DD 名は、実際は OUTFIL データ・セットに使用することもできますが、『SORTOUT』という用語は単一の OUTFIL 出力以外のデータ・セットを示すのに使用します。

**必要となる場合:** OUTFIL ステートメントに指定された各 DD 名は、その OUTFIL データ・セットと対応する DD ステートメントを必要とします。

OUTFIL ステートメントを指定しない場合、すべての出力を処理する E35 ユーザー出口を使用する場合を除き、SORTOUT DD ステートメントは必須です。ユーザー・プログラムが DFSORT を呼び出し、パラメーター・リストで E35 ユーザー出口のアドレスを渡す場合は、SORTOUT DD ステートメントは無視されます。

OUTFIL ステートメントを指定する場合は、SORTOUT DD ステートメントまたは E35 ユーザー出口を指定する必要はありませんが、その両方またはいずれか一方を使用することはできます。

**データ・セットの特性:** 考慮事項について詳しくは、13 ページの『データ・セットに関する考慮事項』を参照してください。

**ブロック・サイズ:** SDB=NO が有効でない限り、出力データ・セット・ブロック・サイズがゼロであるとき、ブロック・セットは、多くの場合、システム決定の



## SORTOUT および OUTFIL DD ステートメント

最適ブロック・サイズを使用します。DFSORT でのシステム決定ブロック・サイズの使用に関する詳細は、181 ページの『OPTION 制御ステートメント』の SDB オプションの説明を参照してください。

一部のジョブの場合、より大きな出力データ・セット・ブロック・サイズを選択すると、DFSORT 処理を正常終了させるために必要な記憶域の量を大きくしなければならぬことがあります。

特定の出力データ・セット・ブロック・サイズが必要なアプリケーションは、そのブロック・サイズを明示的に指定するように変更する必要があります。

SDB=NO が有効な場合は、出力データ・セット、入力データ・セット、および RECORD ステートメントの使用可能な属性に基づいて、DFSORT は出力データ・セットの適切な (必ずしも最適ではない) ブロック・サイズを選択します。出力データ・セットのブロック・サイズは、入力ブロック・サイズと必ずしも同じではありません。

**再ブロック標識:** 次の場合、DFSORT は出力データ・セット・ラベルに再ブロック標識をセットします。

ブロック・セットが選択され、かつ次のいずれかの場合

- DFSORT が出力データ・セット用にシステム決定最適ブロック・サイズをセットする場合 (82 ページの『ブロック・サイズ』を参照) または、
- DFSORT に制御が渡る前に、割り振りが出力データ・セット用にシステム決定最適ブロック・サイズをセットする場合

### 一般的なコーディングの規則:

- コピー・アプリケーションにおいては、SORTOUT データ・セットと OUTFIL データ・セットは、SORTIN データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。
- 組み合わせアプリケーションにおいては、SORTOUT データ・セットと OUTFIL データ・セットは、SORTINnn データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。
- 分類アプリケーションの場合、SORTIN データ・セットと同じ SORTOUT データ・セットまたは OUTFIL データ・セットを使用できますが、分類アプリケーションが正常に終了しなかった場合、データ・セットが消失することがあります。
- OUTFIL データ・セットは、SORTOUT データ・セットまたは OUTFIL データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。
- 全機能付きの IBM 3480 テープ装置に OPTCD=W を指定できません。指定すると、その要求は無効となります。また、このパラメーターを 3420 互換モードで作動する (3400-9 と指定された) 3480 に指定すると、OPTCD=W 要求は無効にはなりませんが、パフォーマンスが低下することがあります。
- 一時または新規出力データ・セットに 2 次割り振りが要求されない場合、NOOUTSEC が有効な場合を除き、自動的に 2 次割り振りが使用されます。

## SORTOUT および OUTFIL DD ステートメント

- テープ・ラベルの RECFM、LRECL、BLKSIZE は、テープ出力データ・セットのみに使用され、その出力データ・セットの指定が、DISP=MOD、DD の通し番号のあるもの、AL、SL、または NSL ラベルをもつものに限られます。
- FREE=CLOSE を指定できません。
- SORTOUT LRECL に関連する詳細については、181 ページの『OPTION 制御ステートメント』の SOLRF および NOSOLRF オプションの説明を参照してください。

### 例 8 SORTOUT DD ステートメント:

```
//SORTOUT DD DSN=C905460.OUTPT,UNIT=3390,SPACE=(CYL,5),  
//          DISP=(NEW,CATLG)
```

#### DISP

データ・セットはオペレーティング・システムに認識されておらず (NEW)、C905460.OUTPT という名前でもカタログ (CATLG) されることを指定します。

#### DSNAME

データ・セットは C905460.OUTPT と呼ばれることを指定します。

#### SPACE

このデータ・セットには 5 シリンダの記憶域が必要です。

#### UNIT

データ・セットが 3390 装置上にあることを示します。

## SORTCKPT DD ステートメント

SORTCKPT データ・セットは、基本順次アクセス方式 (BSAM) で作動する任意の装置に割り振ることができます。最後にとられたチェックポイントからしか、処理を再始動できません。

### 例 9 SORTCKPT DD ステートメント:

```
//SORTCKPT DD DSNAME=CHECK,VOLUME=SER=000123,  
//          DSP=(NEW,KEEP),UNIT=3480
```

SORTCKPT データ・セットを割り振る場合は、作業データ・セットが少なくとも 1 つ必要です。

CKPT オペランドを OPTION または SORT 制御ステートメントで指定すると、さらに多くの中間記憶域が必要になることがあります。

チェックポイント / 再始動機能を使用する場合は、747 ページの『チェックポイント / 再始動』を参照してください。

## SORTCNTL DD ステートメント

SORTCNTL データ・セットは、DFSORT が別のプログラム (たとえば、COBOL または PL/I で作成されたプログラム) から呼び出された場合に、DFSORT 制御ステートメント、注釈ステートメント、ブランク・ステートメント、および注記を作成するために使用できます。

## SORTCNTL DD ステートメント

- SORTCNTL データ・セットは、通常は入力ストリームに入れられますが、順次データ・セットまたは区分データ・セットのメンバーとして定義することもできます。
- データ・セットは、F または FB の RECFM で定義します。LRECL は、80、またはそれ以上 (有効なとき) の値を指定できます。LRECL が 80 より大きい場合、DFSORT は各レコードの最初の 80 バイトを使用します。
- DFSORT は、システムが BSAM の「類似した」連結データ・セットをサポートする程度に SORTCNTL データ・セットをサポートします。「類似した」連結データ・セットについては、*z/OS DFSMS: Using Data Sets* を参照してください。
- DFSORT を PL/I プログラムから呼び出す場合、SORTCNTL または DFSPARM データ・セットを使用して新しい RECORD 制御ステートメントを作成できません。

### 例 10 SORTCNTL DD ステートメント:

```
//SORTCNTL DD *  
OPTION MAINSIZE=8M
```

#### 注:

1. OPTION ステートメントのキーワードの EFS、LIST、NOLIST、LISTX、NOLISTX、LOCALE、MSGPRT、MSGDDN、SMF、SORTDD、SORTIN、および SORTOUT は、拡張パラメーター・リスト、または DFSPARM データ・セット内に入れて渡される場合のみ使用します。それらが、SYSIN または SORTCNTL データ・セットから読み込まれた OPTION ステートメントに指定されている場合、キーワードは認識されますが、パラメーターは無視されます。  
ユーザー・プログラムが 2 回以上 DFSORT を呼び出す場合、各呼び出しごとに異なるバージョンの SORTCNTL データ・セットを読み取るように DFSORT に指示できます。181 ページの『OPTION 制御ステートメント』の SORTDD パラメーターの説明を参照してください。
2. SORTCNTL DD ステートメントの代わりに DFSPARM DD ステートメントを使用すると、他のすべてのソースを変更する単一ソースのデータ・セットに、EXEC PARM オプションと DFSORT 制御ステートメントの両方を指定できます。『DFSPARM DD ステートメント』を参照してください。指定変更の規則については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## DFSPARM DD ステートメント

DFSPARM DD ステートメントを使用して、単一の DD ソースから DFSORT プログラム制御ステートメントおよび EXEC ステートメントの PARM オプションを作成できます。DFSORT がプログラムに呼び出されたか直接呼び出されたかにかかわらず、DFSPARM データ・セット内のステートメントは読み込まれるので、DFSORT を他のプログラムから呼び出すときに (SORTCNTL とは異なり)、EXEC PARM オプションを指定できます。DFSPARM は、すべての DFSORT プログラム制御ステートメント、すべての EXEC ステートメント PARM オプション (SYSIN および SORTCNTL により無視されるものを含む)、および DFSORT OPTION ステートメントに指定されたすべての同等のオプションを受け入れます。

## DFSPARM DD ステートメント

また、DFSPARM は注釈ステートメント、ブランク・ステートメント、および注記も受け入れます。

プログラムから DFSORT を呼び出す場合の DFSPARM の使用の例は、422 ページの『DFSORT 制御ステートメントをプログラムから指定変更する』を参照してください。

指定変更および適用度の詳細については、下記および 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』に記載してあります。

- DFSPARM を使用する場合、ユーザーのジョブがリンク・エディットを必要としない限り SYSIN は不要です。
- DFSPARM データ・セットは、通常は入力ストリームに入れますが、順次データ・セットまたは区分データ・セットのメンバーとして定義することもできます。
- データ・セットは、F または FB の RECFM で定義します。LRECL は、80、またはそれ以上 (有効なとき) の値を指定できます。LRECL が 80 より大きい場合、DFSORT は各レコードの最初の 80 バイトを使用します。
- システムが BSAM の「類似した」連結データ・セットをサポートするのと同じように DFSORT は、DFSPARM データ・セットをサポートします。「類似した」連結データ・セットについては、*z/OS DFSMS: Using Data Sets* を参照してください。
- DFSORT を PL/I プログラムから呼び出す場合、SORTCNTL または DFSPARM データ・セットを使用して新しい RECORD 制御ステートメントを作成できません。

**注:** 本書では、EXEC PARM オプションおよび DFSORT プログラム制御ステートメント用のこのデータ・セット・ソースに DFSPARM という名前の DD 名を使用しています。DFSORT のインストール時に、システム・プログラマーが ICEMAC インストール・マクロの PARMDDN オプションを用いて、この名前をユーザーのシステムに適したものに変更している場合があります。DFSPARM で使用できる機能を使う前に、DD 名が正しいことを確認してください。

**一般的なコーディングの規則:** DFSPARM DD ステートメントのパラメーターのコーディング規則は、JCL EXEC ステートメントの PARM オプションおよび SYSIN または SORTCNTL で指定されるプログラム制御ステートメントに適用される規則と同じです。ただし、次の例外があります。

- ラベルを使用できません。
- PARM オプションとプログラム制御ステートメントを同じ行に置くことはできませんが、行を変えればどのような順序でも指定できます。
- PARM オプションは、PARM= キーワードおよび引用符を使用しないで指定します。
- コンマ (またはセミコロン) は受け入れられますが、PARM オプションが別の行に続く場合は必ずしも必要ではありません。
- PARM オプションに先行ブランクは必須ではありませんが、プログラム制御ステートメントには少なくとも 1 つの先行ブランクが必要です。

## DFSPARM DD ステートメント

FREE=CLOSE は、適当な DFSPARM データ・セットに使用できます (たとえば、一時的データ・セットと永続順次データ・セット、ただし DD \* データ・セットの指定のないもの)。

他のプログラムから DFSORT が呼び出される時、FREE=CLOSE の指定があると、呼び出し先のプログラムに DFSORT の制御が戻ったとき、DFSPARM データ・セットは解放されます。このようにすると、引き続いての呼び出しで、もう 1 度 DFSPARM データ・セットを使用できます。

たとえば、COBOL プログラムに SORT 動詞が 3 つ入っているとします。最初の SORT 動詞には DP1 の制御ステートメントが、2 番目の SORT 動詞には DP2 の制御ステートメントが、3 番目の SORT 動詞には DP3 の制御ステートメントが使用されることになります。

```
//DFSPARM DD DSN=DP1,DISP=SHR,FREE=CLOSE
//DFSPARM DD DSN=DP2,DISP=SHR,FREE=CLOSE
//DFSPARM DD DSN=DP3,DISP=SHR,FREE=CLOSE
```

FREE=CLOSE の指定がない場合は、DP1 が 3 つの SORT 動詞すべてに使われます。

### 例 11 DFSPARM DD ステートメント:

```
//DFSPARM DD *
  SORT FIELDS=(1,2,CH,A),STOPAFT=300
  ABEND
  OPTION SORTIN=DATAIN
  STOPAFT=500
```

この例では、DFSPARM DD データ・セットは DFSORT SORT ステートメント、ABEND パラメーターと STOPAFT パラメーター (JCL EXEC ステートメントの PARM='ABEND,STOPAFT=500' の指定に相当)、および DFSORT OPTION ステートメントを渡します。

#### 注:

1. SORT および OPTION は、制御ステートメントです。ABEND および STOPAFT=500 は PARM オプションです。
2. PARM オプションの STOPAFT=500 は、SORT 制御ステートメント・オプションの STOPAFT=300 を指定変更します。

### 例 12 DFSPARM DD ステートメント:

```
//DFSPARM DD *
  SORT FIELDS=(5,2,CH,D),SKIPREC=10
  STOPAFT=100,BSAM,SKIPREC=5
  OPTION SORTIN=DATAIN,SKIPREC=20
```

この例では、DFSPARM DD データ・セットは、SORT プログラム制御ステートメント、1 行に置かれた 3 つの PARM オプション、および OPTION プログラム制御ステートメントを含んでいます。

## DFSPARM DD ステートメント

注: PARM オプションはプログラム制御ステートメントを指定変更するので、DFSORT は SKIPREC=5 を使用して、その他の SKIPREC 指定を無視します。

DFSPARM DD ステートメントで使用するパラメーター、それらが必要になる条件、およびパラメーターを指定しなかった場合のデフォルトについては、34 ページの『EXEC/DFSPARM PARM オプションの指定』および 111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』を参照してください。

## SORTDKdd DD ステートメント

SORTWKdd データ・セットは VIO に割り振ることができます。ICEMAC パラメーター VIO が指定されているか、またはデフォルト NO が使用されている場合、SORTDKdd という DD 名を用いて、DFSORT は SORTWKdd データ・セットの割り振り解除と再割り振りを行いません。DD 名、SORTDKdd は、DFSORT が使用するために予約されています。

## SORTDIAG DD ステートメント

SORTDIAG DD ステートメントは、診断メッセージ (ICE800I ~ ICE999I) を含むすべてのメッセージと制御ステートメントをメッセージ・データ・セットに書き出すことを指定します。このステートメントは DFSORT のすべての手法に使用することができ、EXCP カウントや中間記憶域の割り振りと使用などの情報を提供します。SORTDIAG DD ステートメントはコンソール・メッセージには影響を与えません。診断のツールとして使用することを目的としています。

SORTDIAG を使用する場合は、SYSOUT DD ステートメントまたは ddname DD ステートメントを指定します (DD 名とはインストール時または実行時に指定された代替メッセージ・データ・セットの DD 名です)。ICEMAC オプションの NOMSGDD=QUIT が有効で、かつ代替メッセージ・データ・セットの DD 名のステートメントも SYSOUT の DD 名のステートメントも指定されていない場合、DFSORT は戻りコード 20 で終了します。

### 例 13 SORTDIAG DD ステートメント:

```
//SORTDIAG DD DUMMY
```

## SORTSNAP DD ステートメント

SORTSNAP DD ステートメントは、ESTAE リカバリー・ルーチンが要求したスナップ・ダンプ、または EFS プログラムに対する呼び出しの前または後に要求されたスナップ・ダンプを印刷するデータ・セットを定義します。DFSORT は、必要な場合はいつでも SORTSNAP を動的に割り振ります。SORTSNAP という DD 名は、DFSORT 用に予約されています。

## DD ステートメントの使用

DFSORT ジョブでは、EXEC ステートメントの後に必ず DD ステートメントが必要です。DD ステートメントは、次の 2 つのカテゴリーに分類されます。

- システム DD ステートメント (92 ページの『システム DD ステートメント』に詳述)
- プログラム DD ステートメント (94 ページの『プログラム DD ステートメント』に詳述)

## DD ステートメントの使用

システム DD ステートメント、および一部のプログラム DD ステートメントは、通常、カタログ式プロシージャを使用すれば自動的に与えられます。その他の DD ステートメントはユーザー自身が指定します。

DD ステートメント・パラメーター、それが必要になる条件、およびデフォルトについては、表 7 に要約しています。DCB パラメーターのサブパラメーター (DD ステートメント・パラメーター) については、同様に 90 ページの表 8 に説明しています。

### 注:

1. 可変長レコードに DCB の LRECL サブパラメーターを正確に指定すると、パフォーマンスが向上します。特定の構成について指定する入力レコードの最大長については、14 ページの『データ・セットの注意事項および制約事項』で説明しています。
2. DFSORT アプリケーションを使用する場合は、DD ステートメントで FREE=CLOSE を使用できません。

表 7. DFSORT で使用される DD ステートメント・パラメーター

パラメーター	必要な場合	パラメーター値	デフォルト
{AMPI BUFSP}	パスワードで保護された VSAM データ・セットを使用し、そのパスワードが E18、E38、または E39 で指定される場合。	データ・セットの作成時に指定される最小バッファ・プール値。	なし。
DCB	7トラック・テープを使用する場合 (標準ラベルをもたないテープを入力するため)、およびデフォルトが適用できない場合。	データ・セットに関連するデータ制御ブロック (DCB) を満たす情報を指定する。	(90 ページの表 8 の個々のサブパラメーターを参照)。
DISP	デフォルトが適用できない場合。	データ・セットの状況と処置を示す。	システムは (NEW, DELETE) と見なす。
DSNAME または DSN	DD ステートメントがラベル付き入力データ・セットを定義する場合 (たとえば、SORTIN)、または作成中のデータ・セットを保存またはカタログ (たとえば、SORTOUT)、または別のステップに渡す場合。	データ・セットの完全修飾名または一時名を指定する。	システムが固有な名前を指定する。
LABEL	デフォルトが適用できない場合。	データ・セットのラベル付けおよび保存に関する情報を指定する。	システムは標準ラベル付けと見なす。
SPACE	DD ステートメントが直接アクセス装置上に新しいデータ・セットを定義する場合。	データ・セットを入れるために必要なスペースの大きさを指定する。	なし。

## DD ステートメントの使用

表 7. DFSORT で使用される DD ステートメント・パラメーター (続き)

パラメーター	必要な場合	パラメーター値	デフォルト
UNIT	入力データ・セットがカタログもされず引き渡しも行われない場合、またはデータ・セットが作成される場合。	データ・セットが必要とする入出力装置のタイプと数を (記号で、または具体的に) 指定する。	なし。
VOLUME または VOL	入力データ・セットがカタログもされず引き渡しも行われない場合、入力が複数リールの場合、または出力データ・セットが直接アクセス装置上にあり、しかも保存またはカタログされる場合。	データ・セットが占有するボリューム (1 つ以上) の識別に使用する情報を指定する。	なし。

表 8. DFSORT で使用される DCB サブパラメーター

サブパラメーター	必要時の条件	サブパラメーター値	デフォルト
BUFOFF	ISCI/ASCII 形式でデータを処理する場合。	バッファ・オフセットの長さを指定、またはバッファ・オフセットがブロック長標識であることを指定する。	
DEN	データ・セットが 7 トラック・テープ装置上にある場合。	テープが記録された密度を指定する。	800 bpi (ビット / インチ)。
OPTCD	ISCI/ASCII 形式でデータを処理する場合。	処理されるテープが ISCI/ASCII 形式であることを指定する。	
TRTCH	データ・セットが 7 トラック・テープ装置上にある場合。  データ・セットが IDRC 付きの 3490 または 3480 にあり、かつシステム IDRC を使用しない場合。	7 トラック・テープ上で 8 ビット・バイトを記録する手法を指定する。  データ・セットを短縮するかどうかを指定する。	コンバーターと変換プログラムは使用されない。奇数パリティ。  システム・デフォルト・オプション。



表 8. DFSORT で使用される DCB サブパラメーター (続き)

サブパラメーター	必要時の条件	サブパラメーター値	デフォルト
BLKSIZE <sup>4</sup> <sup>5</sup>	DCB パラメーターが必要であり、デフォルト値が、SORTWKdd ステートメント以外で適用できない場合。	データ・セット内の物理レコードの最大長 (バイト単位) を指定する。	<ul style="list-style-type: none"> <li>古いデータ・セットの場合、データ・セット・ラベル内の値。</li> <li>新しい出力データ・セットの場合、入力データ・セットと RECORD ステートメント値に基づく適切な値。</li> </ul> SDB=YES が有効でない限り、出力データ・セット・ブロック・サイズがゼロであるとき、ブロック・セットは、システム決定の最適ブロック・サイズを使用する。 特定の出力データ・セット・ブロック・サイズが必要なアプリケーションは、そのブロック・サイズを明示的に指定するように変更する必要があります。
LRECL <sup>5</sup> <sup>3</sup>		データ・セット内の論理レコードの最大長 (バイト単位) を指定する。	

**DD 名の重複:** 同一ステップ内で特定の DD 名 (たとえば、SORTIN) を 2 回以上指定した場合、DFSORT は最初の DD 名を使用し、その後の重複 DD 名は無視します。処理は正常に続けられます。

さらに、次の規則が適用されます。

- SORTIN00、SORTIN01...SORTIN09 の代わりに、それぞれ SORTIN0、SORTIN1...SORTIN9 を指定できます。および SORTIN0x を両方同じジョブ・ステップで指定した場合、DFSORT はこれらを重複 DD 名として取り扱い、使用している最初のもの以外はすべて無視します。たとえば、SORTIN2 および SORTIN02 は重複として取り扱われ、SORTIN2 だけが使用されます。

**注:** 従来の組み合わせの場合、SORTIN01,SORTIN02...SORTIN16 しか使用できないという制約があるために、認識されません。これらの受け入れられる DD 名が重複して使用されている場合は無視されます。

4. 96 ページの『SORTIN DD ステートメント』 および 99 ページの『SORTINnn DD ステートメント』を参照してください。

5. DD \* データ・セットには、このサブパラメーターだけしか使用できません。

6. 固定長レコードの埋め込みおよび切り捨てについては、14 ページの『データ・セットの注意事項および制約事項』を参照してください。

## DD ステートメントの使用

307 ページの『OUTFIL ステートメントに関する注意事項』に説明されているとおり、OUTFIL ステートメント・レベルでは OUTFIL DD 名は無視されます。

**磁気テープ装置の共有:** 以下の DFSORT データ・セットのペアを、1 つの磁気テープ装置に割り当てることができます。

- SORTIN データ・セットと SORTWK01 データ・セット (テープ作業データ・セットは分類のみ)
- SORTIN データ・セットと SORTOUT データ・セットまたは OUTFIL データ・セット (分類アプリケーションのみ)

SORTIN データ・セットと SORTWK01 を関連付けたい場合は、SORTWK01 についての DD ステートメントにパラメーター UNIT=AFF=SORTIN を組み込みます。AFF サブパラメーターを指定すると、システムはそのデータ・セットを、サブパラメーターに続く DD 名 (この場合は、SORTIN) をもつデータ・セットと同じ装置上に置きます。

同様に、SORTOUT または OUTFIL DD ステートメントで UNIT=AFF=SORTIN を指定すると、SORTIN データ・セットを SORTOUT データ・セットまたは OUTFIL データ・セットと関連付けることができます。

SORTINnn テープ・データ・セットは同時に読み取られるので、それらはすべて別々の磁気テープ装置に入れておく必要があります。SORTOUT と OUTFIL テープ・データ・セットは同時に書き込まれるので、すべて別々の磁気テープ装置に入れておく必要があります。

**システム DD ステートメント:** DFSORT の呼び出しに SORT または SORTD カタログ式プロシージャーを使用しない場合は、入力ジョブ・ストリームにシステム DD ステートメントを指定しなければならないことがあります (DFSORT 専用の DD ステートメント、たとえば SORTIN については以下の項を参照してください)。カタログ式プロシージャーに含まれる (またはユーザーが指定する) DD ステートメントは、次のとおりです。

### //JOB LIB DD

ユーザーのプログラム・リンク・ライブラリーがまだシステムに知らされていない場合、そのライブラリーを定義します。

### //STEPLIB DD

//JOB LIB DD と同じ。

### //SYSIN DD

DFSORT が別のプログラムにより呼び出されるのではなく、JCL により呼び出される場合、DFSORT 制御ステートメント、注釈ステートメント、ブランク・ステートメント、および注記を含みます。また、DFSORT がリンク・エディットするオブジェクト・デッキ形式のユーザー出口ルーチンを含めることもできます。

- DFSPARM を使用する場合、ユーザーのジョブがリンク・エディットを必要としない限り SYSIN は不要です。
- SYSIN 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、順次データ・セットまたは区分データ・セットのメンバーとして定義することもできます。

## DD ステートメントの使用

- データ・セットは、RECFM=F または FB と、LRECL=80 で定義します。
- DFSORT は、システムが BSAM の「類似した」連結データ・セットをサポートする程度に SYSIN データ・セットをサポートします。「類似した」連結データ・セットについては、*z/OS DFSMS: Using Data Sets* を参照してください。

**注:** OPTION ステートメントのキーワードの EFS、LIST、NOLIST、LISTX、NOLISTX、LOCALE、MSGPRT、MSGDDN、SMF、SORTDD、SORTIN、および SORTOUT は、拡張パラメーター・リスト、または DFSPARM データ・セット内に入れて渡される場合のみ使用します。それらが、SYSIN または SORTCNTL データ・セットから読み込まれた OPTION ステートメントに指定されている場合、キーワードは認識されますが、パラメーターは無視されます。

代わりに DFSPARM DD ステートメントを使用すると、他のすべてのソースを変更する単一のソース・データ・セットに、EXEC PARM オプションと DFSORT 制御ステートメントの両方を指定できます。107 ページの『DFSPARM DD ステートメント』を参照してください。

ユーザー出力ルーチンが SYSIN にある場合は、次のことを確認してください。

- END ステートメントが最後の 制御 ステートメントであること。
- ユーザー出力ルーチンが数字順に並んでいること (たとえば、E11 は E15 の前)。
- ユーザー出力ルーチンが END 制御ステートメントの直後に指定されていること。
- SYSIN の最後のオブジェクト・デッキの後には何も無いこと。
- SORTMODS DD ステートメントが含まれていること。

DFSORT がプログラムで呼び出され、24 ビットまたは拡張パラメーター・リストの SORTCNTL または DFSPARM から DFSORT 制御ステートメントを指定した場合でも、SYSIN は引き続き、システム入力ストリームに入れられたユーザー出力ルーチンのソースのままです。

### //SYSOUT DD

メッセージのシステム出力データ・セットを示します。カタログ式プロシージャを使用しない場合は、常にこのステートメントを使用します。DFSORT を他のプログラムから呼び出す場合、メッセージ・データ・セットに代替 DD 名を指定できます。(COBOL プログラムから DFSORT を呼び出し、SYSOUT 以外の DD 名を何も使用しない場合、COBOL プログラムで EXHIBIT または DISPLAY を使用すると、不確かな印刷結果になることがあります。) DFSORT メッセージを印刷する前に、改ページが実行されます。DFSORT は、データ・セット属性に RECFM=FBA、LRECL=121、指定された BLKSIZE を使用します。指定した BLKSIZE が 121 の倍数でない場合は、DFSORT は BLKSIZE=121 を使用します。BLKSIZE を指定しないと、DFSORT は SDBMSG インストール・オプションで指定されているブロック・サイズを選択します (*DFSORT 導入とカスタマイズ リリース 14* を参照してください)。

## DD ステートメントの使用

### **//SYSUDUMP DD**

DD システム ABEND ダンプ・ルーチンからの出力用のデータ・セットを定義します。

### **//SYSMDUMP DD**

//SYSUDUMP DD と同じ。

### **//SYSABEND DD**

//SYSUDUMP DD と同じ。

IBM 提供の SORT カタログ式プロシージャを使用する場合は、下記のステートメントが自動的に提供されます。SORT カタログ式プロシージャを使用せずにリンケージ・エディターを使用している場合は、次の DD ステートメントを提供します。

### **//SYSPRINT DD**

リンケージ・エディターからのメッセージが含まれます。

### **//SYSUT1 DD**

リンケージ・エディター用の中間記憶域データ・セットを定義します。

### **//SYSLIN DD**

リンケージ・エディター用の制御情報データ・セットを定義します。

### **//SYSLMOD DD**

リンケージ・エディターからの出力用データ・セットを定義します。

**注:** ユーザー・ルーチンを組み込まない場合、またはリンク・エディットを必要としないユーザー・ルーチンを組み込む場合は、IBM 提供の SORTD カタログ式プロシージャを使用できます。リンク・エディットを必要とするユーザー・ルーチンを組み込む場合は、SORT カタログ式プロシージャを使用できます。

**プログラム DD ステートメント:** SORT または SORTD カタログ式プロシージャを使用して DFSORT を呼び出す場合でも、専用 DD ステートメントを追加指定しなければならないことがあります。次に、これらのステートメントを要約したリストを示し、その後で個々のステートメントについて、さらに詳しく説明します。

### **//SORTLIB DD**

DFSORT の特殊なロード・モジュールを入れるデータ・セットを定義します。通常は省略できます。

### **//SORTIN DD**

分類アプリケーションまたはコピー・アプリケーションの入力データ・セットを定義します。組み合わせアプリケーションには使用されません。

### **//SORTINnn DD**

組み合わせアプリケーションの入力データ・セットを定義します。分類アプリケーションまたはコピー・アプリケーションには使用されません。

### **//SORTWKdd DD**

中間記憶データ・セットを定義します。動的割り振りが要求されない限り、通常、分類アプリケーションに必要です。コピーまたは組み合わせアプリケーションには使用されません。

**//SORTOUT DD**

分類アプリケーション、組み合わせアプリケーション、またはコピー・アプリケーションの出力データ・セット、SORTOUT を定義します。

**//outfil DD**

分類アプリケーション、組み合わせアプリケーション、またはコピー・アプリケーションの出力データ・セット、OUTFIL を定義します。

**//SORTCKPT DD**

最後のチェックポイントから分類を再始動するためにシステムが必要とする情報を保管するために使用するデータ・セットを定義します。チェックポイント機能を使用する場合のみ必要です。

**//SORTCNTL DD**

DFSORT がプログラムで呼び出される場合に、追加または変更済み DFSORT 制御ステートメントを読み取るデータ・セットを定義します。有効なデータ・セット属性については、SYSIN DD ステートメントを参照してください。

**//DFSPARM DD**

DFSORT が直接呼び出される場合、またはプログラムで呼び出される場合は、追加または変更済み DFSORT プログラム制御ステートメントと EXEC ステートメントの PARM オプションの両方を読み取るデータ・セットを定義します。有効なデータ・セット属性については、SYSIN DD ステートメントを参照してください。

**//SORTDKdd DD**

動的に再割り振りされる場合、DFSORT が VIO SORTWKdd 割り振りに使用するデータ・セットを定義します。SORTDKdd をジョブ・ストリームで指定できません。

**//SORTDIAG DD**

すべてのメッセージおよび制御ステートメントを印刷することを指定します。診断およびデバッグの場合に使用されます。

**//SORTSNAP DD**

DFSORT が動的に割り振るスナップ・ダンプ・データ・セットを定義します。SORTSNAP をジョブ・ストリーム内で指定できません。

**//SORTMODS DD**

一時区分データ・セットを定義します。この一時データ・セットは、アプリケーション用の SYSIN 内にあるすべてのユーザー・ルーチンを入れるために十分な大きさである必要があります。SYSIN 内にユーザー・ルーチンがない場合は、このステートメントは不要です。ユーザー・ルーチンがライブラリー内にある場合、そのライブラリーを定義する DD ステートメントを指定する必要があります。

DFSORT は、SYSIN 内のユーザー・ルーチンをリンク・エディットまたは処理する前に、この DD ステートメントで定義されたデータ・セットへ一時的に転送します。

**SORTLIB DD ステートメント:** 通常は、SORTLIB DD ステートメントは省略できます。このステートメントは特殊な DFSORT ロード・モジュールを入れるデータ・セットを記述するものです。

## DD ステートメントの使用

**SORTLIB DD ステートメントが必要なケース：**ICEMAC オプションの SORTLIB=PRIVATE が有効であるか、またはユーザー出口のダイナミック・リンク編集が指定されていて、次の条件に合う場合、このステートメントが必要となります。

- テープ作業データ・セットを使用する分類アプリケーションの場合
- ブロック・セットが使用できない組み合わせアプリケーションの場合 (メッセージ ICE800I を参照)

ICEMAC SORTLIB オプションは、テープ作業データ・セット分類または従来の組み合わせに必要なロード・モジュールを、DFSORT がシステム・ライブラリーまたは専用ライブラリーで探すかどうかを決めます。

### 例 1 SORTLIB DD ステートメント：

```
//SORTLIB DD DSN=USORTLIB,DISP=SHR
```

この例は、すでにカタログされている入力データ・セットを定義する DD ステートメント・パラメーターを示しています。

#### DSNAME

これにより、システムは USORTLIB という名前のデータ・セットをカタログの中から探します。データ・セットが見つかったら、SORTLIB という DD 名と関連付けられます。制御プログラムはカタログから装置割り当てとボリューム通し番号を入手し、そのボリュームがまだ取り付けられていなければ、取り付け指示メッセージをオペレーターに書き出します。

#### DISP

データ・セットがこのジョブ・ステップの前に存在していたこと、このジョブ・ステップの後でも保存しておく必要があること、そして複数ジョブで同時に使用できること (SHR) を示します。どのジョブもこのデータ・セットを変更できません。

SORTLIB DD ステートメントで使用するパラメーター、それらが必要になる条件、およびパラメーターを指定しなかったときのデフォルトについては、89 ページの表 7 を参照してください。DCB パラメーターのサブパラメーターについては、同様に 90 ページの表 8 を参照してください。詳細については、*z/OS MVS JCL 解説書*および *z/OS MVS JCL ユーザーズ・ガイド* を参照してください。

**SORTIN DD ステートメント：** SORTIN DD ステートメントは、分類またはコピーしたいレコードが含まれるデータ・セットの特性を記述し、またその位置を示します。

SORTIN DD ステートメントは、すべての分類アプリケーションまたはコピー・アプリケーションに必要です。ただし、DFSORT へすべての入力を与える E15 ユーザー出口を設定し、かつプログラム制御ステートメントに RECORD を指定した場合を除きます。ユーザー・プログラムが DFSORT を呼び出し、E15 ユーザー出口のアドレスをパラメーター・リストで渡す場合は、SORTIN DD ステートメントは無視されます。

データ・セット特性 DFSORT は、空またはヌルの非 VSAM データ・セットを受け入れます (DCB パラメーターを必ず指定してください)。DFSORT は、分類または

コピー処理について、空の永続 VSAM データ・セットも受け入れます。非 VSAM データ・セットの場合、DFSORT は形式 1 DSCB の DSILSTAR フィールドを調べて、データ・セットが空またはヌルであるかどうかを判別します。DSILSTAR がゼロの場合は、DFSORT はそのデータ・セットを空またはヌルとして取り扱います。データ・セットがヌルのマルチボリューム・データ・セットであり、そのマルチボリューム・データ・セットの最初のボリュームの形式 1 DSCB の DSIND80 フラグがオフの場合、DFSORT は出力のためにそのデータ・セットをオープンし、そのデータ・セットを入力に使用する前に強制的にファイルの終わり (EOF) マークを付けます。

ヌルのデータ・セットとは、新しく作成されたにもかかわらず、正しくクローズされなかったデータ・セットのことです。テープ作業データ・セットの分類では、ヌルのデータ・セットは正しく処理できません。DASD ボリューム目録 (VTOC 内の DSCB) のデータ・セット・ラベルの「システム・コード」は、その中に DOS または VSE の文字が入っている場合は、VSE オペレーティング・システムが作成するデータ・セットを示します。そのようなデータ・セットはヌルとして扱われることはありませんが、空である場合があります。VSE DASD データ・セットがシステム・コード・フィールド内に DOS または VSE をもたない場合、DFSORT はそのデータ・セットを処理することができません。

考慮事項について詳しくは、13 ページの『データ・セットに関する考慮事項』を参照してください。

連結データ・セットには、次の規則が適用されます。

- 連結されているデータ・セットについて、RECFM はすべて固定長であるか、あるいはすべて可変長のいずれかである必要があります。
- BLKSIZE は可変ですが、以下 2 つは例外です。
  - 以下 3 つの条件すべてが適用される場合。
    - (1) ブロック・セット手法が選択されていない。
    - (2) 連結するデータ・セットの最大ブロック・サイズが、テープ・データ・セットに使用されている。
    - (3) テープ・データ・セットは、連結の 1 次データ・セットではない。
 上記の場合、BLKSIZE パラメーターを使用して、テープ・データ・セットのブロック・サイズを明示的に指定する必要がある。
  - テープ作業データ・セットについて、連結の 1 次データ・セットは最大ブロック・サイズでなければならない。
- 固定長レコードの場合は、LRECL はすべてのデータ・セットについて同一である必要があります。可変長レコードの場合、LRECL は可変ですが、連結の 1 次データ・セットは最大レコード長である必要があります。
- データ・セットが非類似装置上にある場合は、ユーザー出口 18 で EXLST パラメーターを使用できません。
- ブロック・セットが選択されないで BSAM が使用されている場合は、すべてのヌル・データ・セットは、ヌル以外のデータ・セットの前に置く必要があります。そうしないと、予期しない結果が生じることがあります。
- DFSORT は、ヌル・データ・セットの形式 1 の DSCB DSIND80 フラグがオフの場合、BSAM を使用してそのヌル・データ・セットを処理する前に、データ・セットすべてに強制的に EOF マークを付けます。

## DD ステートメントの使用

- DUMMY パラメーターを使用してデータ・セットを定義する場合、そのデータ・セットに他のデータ・セットを連結できません。システムは、DUMMY に連結されたデータ・セットを無視します。
- VSAM データ・セットが連結している場合、システムは 1 次データ・セット だけ処理します。
- 入力に VSAM および 非 VSAM データ・セットの両方を含めることはできません。

### 一般的なコーディングの規則

- コピー・アプリケーションにおいては、SORTIN データ・セットは SORTOUT データ・セットまたは OUTFIL データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。
- 分類アプリケーションでは、SORTIN データ・セットは、SORTWKdd データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。SORTIN データ・セットは、SORTOUT データ・セットまたは OUTFIL データ・セットと同じでもよい場合があります。しかし、この場合、分類アプリケーションが正常に終了しない場合はデータ・セットが失われる結果を招くことがあります。
- FREE=CLOSE を指定できません。ユーザー・ラベルはコピーされません。

### 例 2 SORTIN DD ステートメント

```
//SORTIN DD DSNAME=INPUT,DISP=SHR
```

この例は、すでにカタログされている入力データ・セットを定義する DD ステートメント・パラメーターを示しています。

### DSNAME

これにより、システムは INPUT という名前のデータ・セットをカタログの中から探します。データ・セットが見つかり、SORTIN という DD 名と関連付けられます。制御プログラムはカタログから装置割り当てとボリューム通し番号を入手し、そのボリュームがまだ取り付けられていなければ、取り付け指示メッセージをオペレーターに書き出します。

### DISP

データ・セットがこのジョブ・ステップの前に存在していたこと、このジョブ・ステップの後でも保存しておく必要があること、そして複数ジョブで同時に使用できること (SHR) を示します。どのジョブもこのデータ・セットを変更できません。

### 例 3 SORTIN DD のボリューム・パラメーター

```
//SORTIN DD DSN=SORTIN,DISP=(OLD,KEEP),UNIT=3490,  
//          VOL=SER=(75836,79661,72945)
```

入力データ・セットが複数リールの磁気テープに含まれる場合、SORTIN DD ステートメントに VOLUME パラメーターを指定し、テープ・リールの通し番号を示す必要があります。この例では、入力データ・セットは通し番号が 75836、79661、および 72945 の 3 つのリール上にあります。



データ・セットがディスクまたは標準ラベル付きテープ上にない場合は、その DD ステートメントに DCB パラメーターを指定します。

**SORTINnn DD ステートメント:** SORTINnn DD ステートメントは、組み合わせを行なうレコードが入っているデータ・セットの特性を記述し、これらのデータ・セットの位置を示します。

組み合わせには SORTINnn DD ステートメントが必須です。ただし、組み合わせプログラムが別のプログラムから呼び出され、すべての入力ユーザー出口 E32 のルーチンから与えられる場合を除きます。

**データ・セット特性**入力データ・セットは、非 VSAM または VSAM のどちらか一方のみ使用できます。空またはヌルの非 VSAM データ・セットが受け入れられます。空の VSAM データ・セットにより VSAM オープン・エラー (コード 160) が発生し、DFSORT は終了します。非 VSAM データ・セットの場合、DFSORT は形式 1 DSCB の DSILSTAR フィールドを調べて、データ・セットがヌルまたは空であるかどうかを判別します。DSILSTAR がゼロの場合は、DFSORT はそのデータ・セットをヌルまたは空として取り扱います。ヌル・データ・セットとは、新しく作成されたにもかかわらず正しくクローズされなかったデータ・セットを意味します。従来の組み合わせ手法では、ヌル・データ・セットを正しく処理できません。

すべての入力データ・セットについて RECFM は同一である必要があります。

BLKSIZE は同じである必要はありませんが、従来の組み合わせ処理では、SORTIN01 には最大ブロック・サイズを指定します。

固定長レコードの場合は、LRECL はすべてのデータ・セットについて、同じである必要があります。可変長レコードの場合、LRECL は同じである必要はありません。

データ・セットはマルチボリュームにできますが、連結できません。SORTINnn データ・セットがマルチボリュームで、かつヌルの場合、DFSORT は使用の前にそのデータ・セットに EOF マークを強制的に付けます。

考慮事項について詳しくは、14 ページの『データ・セットの注意事項および制約事項』を参照してください。

一般的なコーディングの規則

- SORTINnn データ・セットは、SORTOUT データ・セットまたは OUTFIL データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータになったり、予期せぬ結果を招くことがあります。
- 最大 16 のデータ・セットを組み合わせることができます (使用可能な記憶装置によりは、それ以上のブロック・セット組み合わせが実現します)。
- ブロック・セット組み合わせを使用する場合、nn には 00 (最初の 0 は任意指定) ~ 99 の整数を任意の順序で指定できます。ブロック・セット組み合わせは、SORTIN0x の形式の DD 名を重複する DD 名として取り扱い、最初の DD 名以外は何回現れても無視します。たとえば、DFSORT が DD ステートメントを SORTIN4 DD... と読み取り、後続の DD ステートメントを SORTIN04 DD... と読み取った場合、後の DD statement は無視されます。

## DD ステートメントの使用

- 従来の組み合わせを使用する場合、nn は 01 ~ 16 の範囲で指定できます。ただし、最初の番号には 01 を使用し、残りの番号は数字の順序どおりに指定します。数字をスキップできません。従来の組み合わせは、“ ” に指定された DD 名を無視することに注意してください。
- FREE=CLOSE を指定できません。ユーザー・ラベルはコピーされません。

### 例 4 SORTIN01 ~ 03 DD ステートメント (組み合わせ)

```
//SORTIN01 DD  DSNAME=MERGE1,VOLUME=SER=000111,DISP=OLD,  
//              LABEL=(,NL),UNIT=3400-3,  
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=240)  
//SORTIN02 DD  DSNAME=MERGE2,VOLUME=SER=000121,DISP=OLD,  
//              LABEL=(,NL),UNIT=3400-3,  
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=240)  
//SORTIN03 DD  DSNAME=MERGE3,VOLUME=SER=000131,DISP=OLD,  
//              LABEL=(,NL),UNIT=3400-3,  
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=240)
```

### 例 5 SORTIN01 ~ 02 DD ステートメント (組み合わせ)

```
//SORTIN01 DD  DSNAME=INPUT1,VOLUME=SER=000101, *  
//              UNIT=3390,DISP=OLD                *DCB PARAMETERS  
//SORTIN02 DD  DSNAME=INPUT2,VOLUME=SER=000201, *SUPPLIED FROM  
//              UNIT=3390,DISP=OLD                *LABELS
```

**SORTWKdd DD ステートメント:** SORTWKdd DD ステートメントは、分類するレコードの中間記憶域に使用するデータ・セットの特性を記述します。また、この特性はデータ・セットの位置も示します。

最大 255 個までの SORTWKdd DD ステートメントを指定できます。ただし、32 個以上を指定しても、ブロック・セット手法を選択しないと、最初の 32 個のみが使用されます。

分類アプリケーションごとに、1 つ以上の SORTWKdd ステートメントが必要です (組み合わせまたはコピーの場合は不要)。ただし、次の場合を除きます。

- 主記憶域に入力を収容できる場合
- 動的ワークスペース割り振りが要求された場合 (DYNALLOC)
- ハイパー分類または データ空間分類が使用される場合

作業データ・セットの使用方法については、685 ページの『付録 A. ワークスペースの使用』を参照してください。

診断メッセージ ICE803I は、中間記憶域の割り振りと使用に関する情報を提供します。

SORTWKdd データ・セットは DASD (直接アクセス記憶装置) またはテープのどちらか一方のみで使用できます。DASD は各種のタイプを混用できます。

入力が 7 トラック・テープでない限り、テープは 9 トラックである必要があります。7 トラックの場合は作業テープを 7 トラック にすることができますが、必須ではありません。

一般的なコーディングの規則

- 入力ファイルが非常に大きい場合を除き、通常は 1 つまたは 2 つの SORTWKdd データ・セットだけで十分です。小さなデータ・セットをいくつか使用するより、大きな SORTWKdd データ・セットを 1 つか 2 つを使用することをお勧めします。SORTWKdd データ・セットは、それぞれ別の装置に配置するほうがパフォーマンスを向上させることができます。  
仮想記憶域などのリソースを最適に割り振るために、作業データ・セットの数は必要以上に多く指定しないでください。
- SORTWKdd データ・セットは、SORTIN データ・セット、SORTOUT データ・セット、OUTFIL データ・セット、または SORTWKdd データ・セットのいずれとも異なる必要があります。同じである場合、データが失われるか、データが誤ったものになるか、あるいは予期せぬ結果となる可能性があります。
- 性能上の理由から、シリンダー割り振りが望まれます。一時的にトラックまたはブロックで割り振られた SORTWKdd データ・セット (ROUND なし) は、DFSORT がシリンダーに再調整します。
- DASD 作業データ・セットには、SORTWKdd または SORTWKd の形式の有効な DD 名が使用できます (たとえば、SORTWK01、SORTWK3、SORTWK2、SORTWK#5、SORTWKA、SORTWKXY など)。DD 名は任意の順序で指定できます。SORTWld と SORTWK0d は、重複 DD 名 (たとえば、SORTWK5 と SORTWK05 を指定すると、両方使用でき、また SORTWKQ と SORTWK0Q も同様) としては扱われません。ただし、255 以上の DD 名を指定して、ブロック・セット手法を選択すると、最初の 255 だけが使用されます。32 以上の DD 名を指定して、ブロック・セット手法を選択しないと、最初の 32 の DD 名だけが使用されます。
- テープ作業データ・セットの場合、最低 3 つの SORTWKdd データ・セットを指定する必要があります。その最初の 3 つの DD 名は、SORTWK01、SORTWK02、SORTWK03 である必要があります。この 3 つの DD 名に続く DD 名は、指定する場合、SORTWK04 から SORTWK32 の順になっている必要があります、数字を飛ばすことはできません。
- FREE=CLOSE を指定できません。
- DD DUMMY は使用できません。
- ISCI/ASCII データに関連するパラメーターを指定できません。

#### DASD 作業データ・セットのコーディング上の注意

- データ・セットは区分ではなく、順次である必要があります。
- SPLIT シリンダー・パラメーターを指定できません。
- 一時 SORTWKdd データ・セットに 2 次割り振りが要求されない場合、NOWRKSEC が有効な場合を除き、自動的に 2 次割り振りが使用されます。(ピアレッジ / ベール分類手法の場合のみ、2 次割り振りは 12 個の作業データ・セットに制限されます。)
- データ・セットを VIO に割り振ると、自動 2 次割り振りは行なわれません。
- 作業データ・セットに 2 次割り振りを要求できます。多くの作業データ・セットが定義されている場合、それらは 1 次割り振りだけが使用されます。(ピアレッジ / ベール分類手法の場合のみ、2 次割り振りは 12 個の作業データ・セットに制限されます。)

## DD ステートメントの使用

- DFSORT は、マルチボリューム・データ・セット用に指定された最初のボリュームのスペースのみを使用します。2 番目以降のボリュームのスペースは使用されません。このため、マルチボリュームの SORTWKdd データ・セットは単一ボリュームの SORTWKdd データ・セットとして取り扱われます。
- 1 次スペースが断片化されると、最初のもの以外はすべて 2 次スペースとして扱われます。

仮想記憶装置で SORTWKdd データ・セットを指定すると、仮想入出力は次のようになります。

- VIO=NO を指定している場合、DFSORT は、その仮想装置と同じタイプの実装置の DD 名、SORTDKdd を使って動的再割り振りを行いません。仮想装置に対応する実装置が利用できない場合は、DFSORT はメッセージ ICE083A を出して終了します。このエラーの詳細については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。VIO SORTWKdd データ・セットが存在している場合は、VIO SORTWKdd データ・セット以外のデータ・セットも再割り振りされます。
- VIO=YES が指定されている場合、仮想記憶装置が使用され、パフォーマンスが低下します。

次の例は、DASD 作業データ・セットを使用した SORTWKdd DD ステートメントを示したものです。

### 例 6 SORTWK01 DD ステートメント、DASD 作業データ・セット

```
//SORTWK01 DD SPACE=(CYL,(15,5)),UNIT=3390
```

チェックポイント / 再始動機能を使用していて、据え置き再始動を行なう必要がある場合は、分類作業データ・セットが失われないように、上記のステートメントに次のパラメーターを付け加えておく必要があります。

```
DSNAME=name1,DISP=(NEW,DELETE,CATLG)
```

したがって、据え置き再始動の場合は、この SORTWKdd DD ステートメントは次のようになります。

```
//SORTWK01 DD DSNAME=name1,UNIT=3390,SPACE=(CYL,(15,5)),  
// DISP=(NEW,DELETE,CATLG)
```

次の例は、3 つのテープ装置を使用する SORTWKdd DD ステートメントを示したものです。

### 例 7 SORTWK01 ~ 03 DD ステートメント、テープ中間記憶装置

```
//SORTWK01 DD UNIT=3400-3,LABEL=(,NL)  
//SORTWK02 DD UNIT=3400-3,LABEL=(,NL)  
//SORTWK03 DD UNIT=3400-3,LABEL=(,NL)
```

DFSORT が正しく終了しなかった場合に上記の DD ステートメントが指定されていると、ステップが正しく再実行されるか、あるいは何らかの方法でデータ・セットが削除されるまで、中間記憶データ・セットはシステム内に残ったままになります。

## DD ステートメントの使用

これらのパラメーターは、3 つの 3400 シリーズ・テープ装置上の、ラベルなしデータ・セットを指定します。DSNAME パラメーターが省略されているため、システムが固有な名前を割り当てます。

***SORTOUT および OUTFIL DD ステートメント:*** SORTOUT および OUTFIL DD ステートメントは、処理済みレコードを入れるデータ・セットの特性を記述し、その位置を示します。

SORTOUT DD ステートメントは、分類、コピー、または組み合わせアプリケーションのための、非 OUTFIL 単一出力データ・セットを指定します。OUTFIL 処理は、SORTOUT には適用されません。

1 つ以上の OUTFIL ステートメントの FNAMES または FILES (またはその両方の) パラメーターは、分類、コピー、または組み合わせアプリケーションの OUTFIL データ・セットの DD 名を指定します。各 OUTFIL ステートメントごとに指定されたパラメーターは、そのステートメントに関連する OUTFIL データ・セットについて OUTFIL 処理が実施されるように定義します。OUTFIL 処理に関する特定の情報については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

SORTOUT DD 名は、実際は OUTFIL データ・セットに使用することもできますが、『SORTOUT』という用語は単一の OUTFIL 出力以外のデータ・セットを示すのに使用します。

OUTFIL ステートメントに指定された各 DD 名は、その OUTFIL データ・セットと対応する DD ステートメントを必要とします。

OUTFIL ステートメントを指定しない場合、すべての出力を処理する E35 ユーザー出口を使用する場合を除き、SORTOUT DD ステートメントは必須です。ユーザー・プログラムが DFSORT を呼び出し、パラメーター・リストで E35 ユーザー出口のアドレスを渡す場合は、SORTOUT DD ステートメントは無視されます。

OUTFIL ステートメントを指定する場合は、SORTOUT DD ステートメントまたは E35 ユーザー出口を指定する必要はありませんが、その両方またはいずれか一方を使用することはできます。

データ・セット特性の考慮事項について、詳しくは、13 ページの『データ・セットに関する考慮事項』を参照してください。

SDB=YES が有効な限り、出力データ・セット・ブロック・サイズがゼロであるとき、ブロック・セットは、多くの場合、システム決定の最適ブロック・サイズを使用します (SDB の使用に関する制約事項のすべてのリストについて、DFSORT 導入とカスタマイズ リリース 14 を参照してください)。システム決定ブロック・サイズは、SMS 管理データ・セットと非 SMS 管理データ・セットの両方に適用され、出力データ・セットが入れられる記憶装置のスペースを最も有効に使用します。

- DASD 出力データ・セットの場合、使用する出力装置の最適ブロック・サイズは、出力データ・セットの RECFM および LRECL 属性に基づいて選択されます。JFCB または形式 1 DSCB からこれらの出力データ・セット属性を使用できない場合、DFSORT は通常どおり、SORTIN 属性または RECORD ステートメント

## SORTOUT および OUTFIL DD ステートメント

ントから最適ブロック・サイズを決定し、この値をシステム決定出力データ・セット・ブロック・サイズの基本とします。

- テープ出力データ・セットの場合、システム決定ブロック・サイズは AL 以外のラベル・タイプのデータ・セットだけに使用されます。最適ブロック・サイズは、出力データ・セットの RECFM および LRECL 属性に基づいて選択されま  
す。表9 に示されているとおりです。JFCB またはテープ・ラベル (該当する  
とき、AL、SL または NSL ラベルで DISP=MOD の場合のみ) からこれらの出力  
データ・セット属性を使用できない場合、DFSORT は通常どおり、SORTIN 属性  
または RECORD ステートメントから最適ブロック・サイズを決定し、この値を  
システム決定出力データ・セット・ブロック・サイズの基本とします。

表9. テープ出力データ・セットのシステム決定ブロック・サイズ

RECFM	BLKSIZE の設定値
F または FS	LRECL
FB または FBS	32760 より小か等しい、LRECL の可能な倍 数の最大値。
V、D、VS または DS	LRECL + 4
VB、DB、VB または DBS	32760

一部のジョブの場合、システム決定ブロック・サイズを使用するときに、より大き  
な出力データ・セット・ブロック・サイズを選択すると、DFSORT 処理を正常終了  
させるために必要な、記憶域の容量を大きくしなければならないことがあります。

特定の出力データ・セット・ブロック・サイズが必要なアプリケーションは、その  
ブロック・サイズを明示的に指定するように変更する必要があります。あるいは、  
システム決定ブロック・サイズをすべての DFSORT アプリケーションで使用する  
のを避けるため、ICEMAC オプション SDB=NO を選択できます。

SDB=YES が有効でない場合は、出力データ・セット、SORTIN、および RECORD  
ステートメントの使用可能な属性に基づいて、DFSORT は出力データ・セットの適  
切な (必ずしも最適ではない) ブロック・サイズを選択します。出力データ・セッ  
ト・ブロック・サイズは、SORTIN ブロック・サイズと必ずしも同じではありません。

次の場合、DFSORT は出力データ・セット・ラベルに再ブロック標識をセットしま  
す。

ブロック・セットが選択され、かつ次のいずれかの場合

- DFSORT が出力データ・セット用にシステム決定最適ブロック・サイズをセット  
する場合または、
- DFSORT に制御が渡る前に、割り振りが出力データ・セット用にシステム決定最  
適ブロック・サイズをセットする場合

一般的なコーディングの規則

- コピー・アプリケーションにおいては、SORTOUT データ・セットと OUTFIL デ  
ータ・セットは、SORTIN データ・セットと異なる必要があります。同じである  
場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くこ  
とがあります。

## SORTOUT および OUTFIL DD ステートメント

- 組み合わせアプリケーションにおいては、SORTOUT データ・セットと OUTFIL データ・セットは、SORTINnn データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータとなったり、予期せぬ結果を招くことがあります。
- 分類アプリケーションの場合、SORTIN データ・セットと同じ SORTOUT データ・セットまたは OUTFIL データ・セットを使用できますが、分類アプリケーションが正常に終了しなかった場合、データ・セットが消失することがあります。
- OUTFIL データ・セットは、SORTOUT データ・セットまたは OUTFIL データ・セットと異なる必要があります。同じである場合は、データが失われたり、誤ったデータになったり、予期せぬ結果を招くことがあります。
- RETPD または LABEL=RETPD を標準ラベル・テープに SORTOUT または OUTFIL DD ステートメントで指定する場合、DCB パラメーターも指定してください。DCB パラメーターが指定されていない場合、テープ・データ・セットが 2 度開かれることがあります。
- 全機能付きの IBM 3480 テープ装置に OPTCD=W を指定できません。指定すると、その要求は無効となります。また、このパラメーターを 3420 互換モードで作動する (3400-9 と指定された) 3480 に指定すると、OPTCD=W 要求は無効にはなりませんが、パフォーマンスが低下することがあります。
- 一時または新規出力データ・セットに 2 次割り振りが要求されない場合、NOOUTSEC が有効な場合を除き、自動的に 2 次割り振りが使用されます。
- テープ・ラベルの RECFM、LRECL、BLKSIZE は、テープ出力データ・セットのみに使用され、その出力データ・セットの指定が、DISP=MOD、DD の通し番号のあるもの、AL、SL、または NSL ラベルをもつものに限られます。
- FREE=CLOSE を指定できません。

### 例 8 SORTOUT DD ステートメント

```
//SORTOUT DD DSN=C905460.OUTPT,UNIT=3390,SPACE=(CYL,5),  
//          DISP=(NEW,CATLG)
```

#### DISP

データ・セットはオペレーティング・システムに認識されておらず (NEW)、C905460.OUTPT という名前でカタログ (CATLG) されることを指定します。

#### DSNAME

データ・セットは C905460.OUTPT と呼ばれることを指定します。

#### SPACE

このデータ・セットには 5 シリンダーの記憶域が必要です。

#### UNIT

データ・セットが 3390 装置上にあることを示します。

**SORTCKPT DD ステートメント:** SORTCKPT データ・セットは、基本順次アクセス方式 (BSAM) で作動する任意の装置に割り振ることができます。最後にとられたチェックポイントからしか、処理を再始動できません。

### 例 9 SORTCKPT DD ステートメント

## SORTCKPT DD ステートメント

```
//SORTCKPT DD  DSNAME=CHECK,VOLUME=SER=000123,  
//              DSP=(NEW,KEEP),UNIT=3400-3
```

SORTCKPT データ・セットを割り振る場合は、作業データ・セットが少なくとも 1 つ必要です。

CKPT オペランドを OPTION または SORT 制御ステートメントで指定すると、さらに多くの中間記憶域が必要になることがあります。

チェックポイント / 再始動機能を使用する場合は、747 ページの『チェックポイント / 再始動』を参照してください。

**SORTCNTL DD ステートメント:** SORTCNTL データ・セットは、DFSORT が別のプログラム (たとえば、COBOL または PL/I で作成されたプログラム) から呼び出された場合に、DFSORT 制御ステートメント、注釈ステートメント、ブランク・ステートメント、および注記を作成するために使用できます。

- SORTCNTL データ・セットは、通常は入力ストリームに入れられますが、順次データ・セットまたは区分データ・セットのメンバーとして定義することもできます。
- データ・セットは、RECFM=F または FB と、LRECL=80 で定義します。
- DFSORT は、システムが BSAM の「類似した」連結データ・セットをサポートする程度に SORTCNTL データ・セットをサポートします。「類似した」連結データ・セットについては、*z/OS DFSMS: Using Data Sets* を参照してください。
- DFSORT を PL/I プログラムから呼び出す場合、SORTCNTL または DFSPARM データ・セットを使用して新しい RECORD 制御ステートメントを作成できません。

### 例 10 SORTCNTL DD ステートメント

```
//SORTCNTL DD *
```

#### 注:

1. OPTION ステートメントのキーワードの EFS、LIST、NOLIST、LISTX、NOLISTX、LOCALE、MSGPRT、MSGDDN、SMF、SORTDD、SORTIN、および SORTOUT は、拡張パラメーター・リスト、または DFSPARM データ・セット内に入れて渡される場合のみ使用します。それらが、SYSIN または SORTCNTL データ・セットから読み込まれた OPTION ステートメントに指定されている場合、キーワードは認識されますが、パラメーターは無視されます。  
ユーザー・プログラムが 2 回以上 DFSORT を呼び出す場合、各呼び出しごとに異なるバージョンの SORTCNTL データ・セットを読み取るように DFSORT に指示できます。181 ページの『OPTION 制御ステートメント』の SORTDD パラメーターの説明を参照してください。
2. SORTCNTL DD ステートメントの代わりに DFSPARM DD ステートメントを使用すると、他のすべてのソースを変更する単一ソースのデータ・セットに、EXEC PARM オプションと DFSORT 制御ステートメントの両方を指定できます。107 ページの『DFSPARM DD ステートメント』を参照してください。指定変更の規則については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。



**DFSPARM DD ステートメント:** DFSPARM DD ステートメントを使用して、単一の DD ソースから DFSORT プログラム制御ステートメントおよび EXEC ステートメントの PARM オプションを作成できます。DFSORT がプログラムに呼び出されたか直接呼び出されたかにかかわらず、DFSPARM データ・セット内のステートメントは読み込まれるので、DFSORT を他のプログラムから呼び出すときに (SORTCNTL とは異なり)、EXEC PARM オプションを指定できます。DFSPARM は、すべての DFSORT プログラム制御ステートメント、すべての EXEC ステートメント PARM オプション (SYSIN および SORTCNTL により無視されるものを含む)、および DFSORT OPTION ステートメントに指定されたすべての同等のオプションを受け入れます。

また、DFSPARM は注釈ステートメント、ブランク・ステートメント、および注記も受け入れます。

指定変更および適用度の詳細については、下記および 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』に記載してあります。

- DFSPARM を使用する場合、ユーザーのジョブがリンク・エディットを必要としない限り SYSIN は不要です。
- DFSPARM データ・セットは、通常は入力ストリームに入れますが、順次データ・セットまたは区分データ・セットのメンバーとして定義することもできます。
- データ・セットは、RECFM=F または FB と、LRECL=80 で定義します。
- システムが BSAM の「類似した」連結データ・セットをサポートするのと同じように DFSORT は、DFSPARM データ・セットをサポートします。「類似した」連結データ・セットについては、*z/OS DFSMS: Using Data Sets* を参照してください。
- DFSORT を PL/I プログラムから呼び出す場合、SORTCNTL または DFSPARM データ・セットを使用して新しい RECORD 制御ステートメントを作成できません。

**注:** 本書では、EXEC PARM オプションおよび DFSORT プログラム制御ステートメント用のこのデータ・セット・ソースに DFSPARM という名前の DD 名を使用しています。DFSORT のインストール時に、システム・プログラマーが ICEMAC インストール・マクロの PARMDDN オプションを用いて、この名前をユーザーのシステムに適したものに変更している場合があります。DFSPARM で使用できる機能を使う前に、DD 名が正しいことを確認してください。

DFSPARM DD ステートメントのパラメーターのコーディング規則は、JCL EXEC ステートメントの PARM オプションおよび SYSIN または SORTCNTL で指定されるプログラム制御ステートメントに適用される規則と同じです。ただし、次の例外があります。

- ラベルを使用できません。
- PARM オプションとプログラム制御ステートメントを同じ行に置くことはできませんが、行を変えればどのような順序でも指定できます。
- PARM オプションは、PARM= キーワードおよび引用符を使用しないで指定します。

## DFSPARM DD ステートメント

- コンマ (またはセミコロン) は受け入れられますが、PARM オプションが別の行に続く場合は必ずしも必要ではありません。
- PARM オプションに先行ブランクは必須ではありませんが、プログラム制御ステートメントには少なくとも 1 つの先行ブランクが必要です。

FREE=CLOSE は、適当な DFSPARM データ・セットに使用できます (たとえば、一時的データ・セットと永続順次データ・セット、ただし DD \* データ・セットの指定のないもの)。

他のプログラムから DFSORT が呼び出される時、FREE=CLOSE の指定があると、呼び出し先のプログラムに DFSORT の制御が戻ったとき、DFSPARM データ・セットは解放されます。このようにすると、引き続いての呼び出しで、もう 1 度 DFSPARM データ・セットを使用できます。

たとえば、COBOL プログラムに SORT 動詞が 3 つ入っているとします。最初の SORT 動詞には DP1 の制御ステートメントが、2 番目の SORT 動詞には DP2 の制御ステートメントが、3 番目の SORT 動詞には DP3 の制御ステートメントが使用されることになります。

```
//DFSPARM DD DSN=DP1,DISP=SHR,FREE=CLOSE
//DFSPARM DD DSN=DP2,DISP=SHR,FREE=CLOSE
//DFSPARM DD DSN=DP3,DISP=SHR,FREE=CLOSE
```

FREE=CLOSE の指定がない場合は、DP1 が 3 つの SORT 動詞すべてに使われます。

### 例 11 DFSPARM DD ステートメント:

```
//DFSPARM DD *
  SORT FIELDS=(1,2,CH,A),STOPAFT=300
ABEND
  OPTION SORTIN=DATAIN
  STOPAFT=500
```

この例では、DFSPARM DD データ・セットは DFSORT SORT ステートメント、ABEND パラメーターと STOPAFT パラメーター (JCL EXEC ステートメントの PARM='ABEND,STOPAFT=500' の指定に相当)、および DFSORT OPTION ステートメントを渡します。

#### 注:

1. SORT および OPTION は、制御ステートメントです。ABEND および STOPAFT=500 は PARM オプションです。
2. PARM オプションの STOPAFT=500 は、SORT 制御ステートメント・オプションの STOPAFT=300 を指定変更します。

### 例 12 DFSPARM DD ステートメント

```
//DFSPARM DD *
  SORT FIELDS=(5,2,CH,D),SKIPREC=10
  STOPAFT=100,BSAM,SKIPREC=5
  OPTION SORTIN=DATAIN,SKIPREC=20
```

## DFSPARM DD ステートメント

この例では、DFSPARM DD データ・セットは、SORT プログラム制御ステートメント、1 行に置かれた 3 つの PARM オプション、および OPTION プログラム制御ステートメントを含んでいます。

**注:** PARM オプションはプログラム制御ステートメントを指定変更するので、DFSORT は SKIPREC=5 を使用して、その他の SKIPREC 指定を無視します。

DFSPARM DD ステートメントで使用するパラメーター、それらが必要になる条件、およびパラメーターを指定しなかった場合のデフォルトについては、34 ページの『EXEC/DFSPARM PARM オプションの指定』および 111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』を参照してください。

**SORTDKdd DD ステートメント:** SORTWKdd データ・セットは VIO に割り振ることができます。ICEMAC パラメーター VIO が指定されているか、またはデフォルト NOVIO が使用されている場合、SORTDKdd という DD 名を用いて、DFSORT は SORTWKdd データ・セットの割り振り解除と再割り振りを行いません。DD 名、SORTDKdd は、DFSORT が使用するために予約されています。

**SORTDIAG DD ステートメント:** SORTDIAG DD ステートメントは、診断メッセージ (ICE800I ~ ICE999I) を含むすべてのメッセージと制御ステートメントをメッセージ・データ・セットに書き出すことを指定します。このステートメントは DFSORT のすべての手法に使用することができ、EXCP カウントや中間記憶域の割り振りと使用などの情報を提供します。SORTDIAG DD ステートメントはコンソール・メッセージには影響を与えません。診断のツールとして使用することを目的としています。

SORTDIAG を使用する場合は、SYSOUT DD ステートメントまたは ddname DD ステートメントが指定される必要があります (DD 名とはインストール時または実行時に指定された代替メッセージ・データ・セットの DD 名です)。ICEMAC オプションの NOMSGDD=QUIT が有効で、かつ代替メッセージ・データ・セットの DD 名のステートメントも SYSOUT の DD 名のステートメントも指定されていない場合、DFSORT は戻りコード 20 で終了します。

例 13 SORTDIAG DD ステートメント

```
//SORTDIAG DD DUMMY
```

**SORTSNAP DD ステートメント:** SORTSNAP DD ステートメントは、ESTAE リカバリー・ルーチンが要求したスナップ・ダンプ、または EFS プログラムに対する呼び出しの前または後に要求されたスナップ・ダンプを印刷するデータ・セットを定義します。DFSORT は、必要な場合はいつでも SORTSNAP を動的に割り振ります。SORTSNAP という DD 名は、DFSORT 用に予約されています。



## 第 3 章 DFSORT プログラム制御ステートメントの使用

プログラム制御ステートメントの使用	114
制御ステートメントの概要	114
基本タスクの説明	114
レコードの組み込みまたは除外	115
レコードの再フォーマット設定と編集	115
複数の出力および報告書作成とレコード変換	115
追加機能およびオプションの呼び出し	115
シンボルの使用	116
一般的なコーディングの規則	116
継続行	118
注釈ステートメントの挿入	119
コーディング上の制約事項	119
EFS プログラムが有効な場合の EFS の制約事項	119
他の IBM プログラムの制御ステートメントの使用	120
ALTSEQ 制御ステートメント	120
EBCDIC 照合順序の更新 - 例	121
例 1	121
例 2	121
例 3	122
例 4	122
例 5	122
DEBUG 制御ステートメント	123
診断オプションの指定 - 例	128
例 1	128
例 2	129
END 制御ステートメント	129
制御ステートメント読み取りの中断 - 例	129
例 1	129
例 2	129
INCLUDE 制御ステートメント	130
関係条件	132
比較	133
関係条件の形式	133
埋め込みと切り捨て	138
文化的環境についての考慮事項	139
出力データ・セットへのレコードの組み込み - 比較の例	139
例 1	139
例 2	139
例 3	140
例 4	140
例 5	141
サブストリング比較テスト	141
関係条件の形式	141
出力データ・セットへのレコードの組み込み - サブストリングの比較の例	142
例	142
ビット論理テスト	143
方法 1: ビット演算子のテスト	143
関係条件の形式	143

## DFSORT プログラム制御ステートメントの使用

フィールド . . . . .	144
マスク . . . . .	145
埋め込みと切り捨て . . . . .	145
出力データ・セットにレコードを組み込む - ビット演算子テストの例	145
例 1 . . . . .	145
例 2 . . . . .	145
例 3 . . . . .	146
方法 2: ビット比較テスト . . . . .	146
関係条件の形式 . . . . .	146
フィールド . . . . .	147
ビット定数 . . . . .	147
埋め込みと切り捨て . . . . .	147
出力データ・セットにレコードを組み込む - ビット比較テストの例	148
例 1 . . . . .	148
例 2 . . . . .	148
例 3 . . . . .	148
日付比較 . . . . .	149
関係条件の形式 . . . . .	149
出力データ・セットへのレコードの組み込み - 日付比較	151
例 1 . . . . .	151
例 2 . . . . .	152
INCLUDE/OMIT ステートメントに関する注意事項 . . . . .	152
INREC 制御ステートメント . . . . .	153
INREC ステートメントに関する注意事項 . . . . .	163
処理前のレコードの再フォーマット設定 - 例 . . . . .	165
例 1 . . . . .	165
例 2 . . . . .	166
例 3 . . . . .	167
例 4 . . . . .	168
例 5 . . . . .	168
MERGE 制御ステートメント . . . . .	169
MERGE または COPY の指定 - 例 . . . . .	172
例 1 . . . . .	172
例 2 . . . . .	172
例 3 . . . . .	172
例 4 . . . . .	173
MODS 制御ステートメント . . . . .	173
ユーザー出口ルーチンの指定 - 例 . . . . .	176
例 1 . . . . .	176
例 2 . . . . .	176
OMIT 制御ステートメント . . . . .	177
出力データ・セットからレコードを除外する - 例 . . . . .	179
例 . . . . .	179
OPTION 制御ステートメント . . . . .	181
OPTION ステートメント・オプションの別名 . . . . .	233
DFSORT オプションまたは COPY の指定 - 例 . . . . .	233
例 1 . . . . .	233
例 2 . . . . .	234
例 3 . . . . .	234
例 4 . . . . .	235
例 5 . . . . .	235
例 6 . . . . .	235

## DFSORT プログラム制御ステートメントの使用

例 7 . . . . .	236
例 8 . . . . .	236
例 9 . . . . .	237
OUTFIL 制御ステートメント . . . . .	238
OUTFIL ステートメントに関する注意事項 . . . . .	307
OUTFIL 機能 - 例 . . . . .	311
例 1 . . . . .	311
例 2 . . . . .	311
例 3 . . . . .	312
例 4 . . . . .	315
例 5 . . . . .	318
例 6 . . . . .	319
例 7 . . . . .	319
例 8 . . . . .	320
例 9 . . . . .	320
例 10 . . . . .	321
例 11 . . . . .	322
例 12 . . . . .	322
例 13 . . . . .	323
例 14 . . . . .	324
例 15 . . . . .	324
例 16 . . . . .	325
例 17 . . . . .	325
OUTREC 制御ステートメント . . . . .	327
OUTREC ステートメントに関する注意事項 . . . . .	335
出力レコードの再フォーマット設定 - 例 . . . . .	337
例 1 . . . . .	337
例 2 . . . . .	337
例 3 . . . . .	337
例 4 . . . . .	337
例 5 . . . . .	338
例 6 . . . . .	338
例 7 . . . . .	339
RECORD 制御ステートメント . . . . .	340
レコード形式および長さの記述 - 例 . . . . .	344
例 1 . . . . .	344
例 2 . . . . .	344
SORT 制御ステートメント . . . . .	345
SORT/MERGE ステートメントに関する注意事項 . . . . .	353
SORT または COPY の指定 - 例 . . . . .	353
例 1 . . . . .	353
例 2 . . . . .	354
例 3 . . . . .	354
例 4 . . . . .	355
例 5 . . . . .	355
例 6 . . . . .	355
SUM 制御ステートメント . . . . .	355
SUM ステートメントに関する注意事項 . . . . .	357
合計フィールドの加算 - 例 . . . . .	359
例 1 . . . . .	359
例 2 . . . . .	359
例 3 . . . . .	359

---

## プログラム制御ステートメントの使用

プログラム制御ステートメントは、レコードを処理するときに DFSORT に指示を出します。プログラム制御ステートメントには、必須のものとオプションのものがあります。次の場合に制御ステートメントを使用します。

- 分類、組み合わせ、またはコピーを実行するかどうか指示する。
- 使用する制御フィールドを記述する。
- ユーザー独自のルーチンに制御を渡すためのプログラム出口を指示する。
- 呼び出したい DFSORT 機能を記述する。
- 入力ファイルおよび出力ファイルを記述する。
- 処理中に使用する各種オプションを指示する。

ユーザーは、以下のものからプログラム制御ステートメントを DFSORT に与えることができます。

- SYSIN データ・セット
- SORTCNTL データ・セット
- DFSPARM データ・セット
- 24 ビット・パラメーター・リスト
- 拡張パラメーター・リスト

各ソースをどのようなときに使用するかについての説明は、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

DFSORT パネルは、プログラム制御ステートメントを直接コーディングする代替手段を提供します。パネルを使用してジョブを作成したりあるいはデータ・セットに保管する場合、情報やコマンドをオンラインで入力することにより、必要なステートメントを正しい構文で作成できます。詳細については、*DFSORT* パネルの手引きを参照してください。

この章では、DFSORT プログラム制御ステートメントの要約とコーディング規則について説明します。各ステートメントの詳細説明については、その後に記述します。

---

## 制御ステートメントの概要

### 基本タスクの説明

DFSORT アプリケーションで必要とされるプログラム制御ステートメントは、ユーザーがレコードの分類、組み合わせ、またはコピーのいずれの処理を行うかを指定する SORT、MERGE、または OPTION ステートメントだけです。(コピー処理は、この 3 つのステートメントのいずれでも指定できます。)

**SORT** 分類処理をコーディングしている場合に制御フィールドを記述したり、コピー処理を指定します。分類を昇順または降順のどちらで行うかを指定します。



**MERGE**

組み合わせ処理をコーディングしている場合に制御フィールドを記述したり、コピー処理を指定します。組み合わせを昇順または降順のどちらで行うかを指定します。

**OPTION**

インストール時のデフォルト (EQUALS、CHALT、CHECK など) を指定変更して、任意の情報 (DYNALLOC、SKIPREC など) を与えます。コピー・アプリケーションを指定することもできます。

**レコードの組み込みまたは除外**

特定のレコードを出力データ・セット内に組み込むか、または出力データ・セットから除外するかを指定できます。

**INCLUDE**

特定の基準に合うフィールドをもつレコードだけを組み込むように指定します。

**OMIT** 特定の基準に合うフィールドをもつレコードだけを削除することを指定します。

**OUTFIL**

レコードを複数の出力データ・セットに組み込むかあるいは除外するかを指定します。

**レコードの再フォーマット設定と編集**

フィールドの削除、フィールドの並べ替え、およびブランク、ゼロ、または定数の挿入を行うことにより、個々のレコードを変更できます。

**INREC**

分類、コピー、または組み合わせを行う前に、どのようにレコードを再フォーマット設定するかを指定します。

**OUTREC**

分類、コピー、または組み合わせを行なった後で、どのようにレコードを再フォーマット設定するかを指定します。

**OUTFIL**

複数の出力データ・セットで、どのようにレコードを再フォーマット設定するかを指定します。

**複数の出力および報告書作成とレコード変換**

複数の出力データ・セットや報告書を作成して、可変長レコードを固定長レコードに、また固定長レコードを可変長レコードに変換できます。

**OUTFIL**

出力データ・セットを指定して、どのレコードを各データ・セットに表示させるかを指定します。可変長から固定長または固定長から可変長にレコードを変換する方法を指定します。

**追加機能およびオプションの呼び出し**

残りの制御ステートメントは、各種のタスクを行うために使用できます。

## 制御ステートメントの概要

### ALTSEQ

ALTSEQ 変換テーブルへの変更を指定して、形式 AQ の場合は SORT、MERGE、INCLUDE または OMIT フィールドで使用し、TRAN=ALTSEQ の場合は INREC、OUTREC および OUTFIL OUTREC フィールドで使用します。

### DEBUG

各種の診断オプションを指定します。

**END** DFSORT に SYSIN、SORTCNTL、または DFSPARM の読み取りを中止させます。

**MODS** DFSORT ジョブで 1 つ以上のユーザー出口ルーチンを使うことを指定します。ユーザー出口ルーチンについては、361 ページの『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』を参照してください。

### RECORD

長さタイプを提供するために使用できます。

**SUM** 等しい制御フィールドをもつレコードの数値合計フィールドを 1 つのレコードで合計し、その他のレコードは削除するように指定します。

## シンボルの使用

INCLUDE、INREC、MERGE、OMIT、OUTFIL、OUTREC、SORT、および SUM の各 DFSORT 制御ステートメントで、どのフィールド、または定数にも、シンボルを定義して使用できます。シンボルを使用すると、さまざまなレコード・レイアウトに関連した情報を表すシンボルの集合を作成して、繰り返し使用 (つまり、マッピング) できます。詳しくは、555 ページの『第 7 章 フィールドと定数のシンボルの使用』を参照してください。

---

## 一般的なコーディングの規則

注釈ステートメント、ブランク・ステートメント、および注記の使用方法については、119 ページの『注釈ステートメントの挿入』を参照してください。DFSORT プログラム制御ステートメントおよび EXEC PARM オプションは、ユーザー定義の DD データ・セットでまとめて指定することもできます。この DD ソースに適用される特別なコーディング規則については、85 ページの『DFSPARM DD ステートメント』を参照してください。

その他の DFSORT 制御ステートメントはすべて、117 ページの図 5 に示されるように同じ一般形式のステートメントです。ただし、この図に示されている形式は、パラメーター・リストに指定する制御ステートメントには適用されません。適用される特別な規則については、421 ページの『第 5 章 プログラムからの DFSORT の呼び出し』を参照してください。

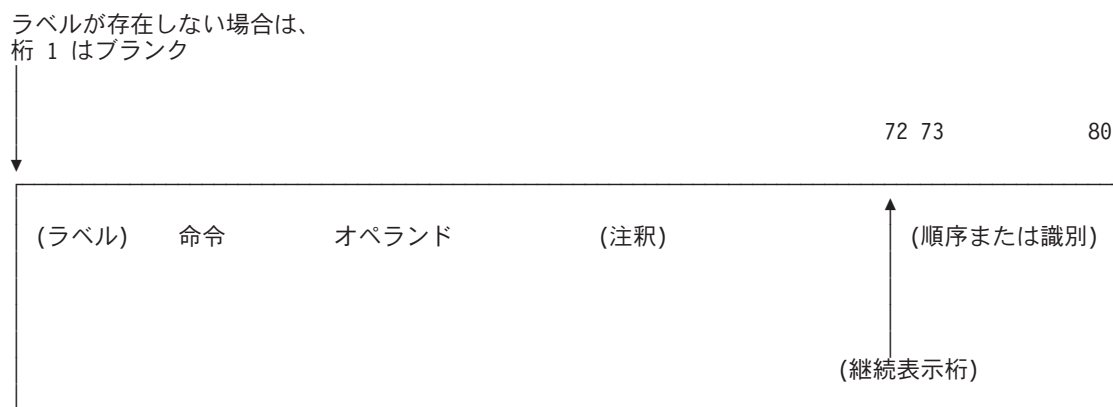


図 5. 制御ステートメントの形式

制御ステートメントはフリー・フォームです。つまり、命令定義子、オペランド、および注釈欄は、それらが正しい順序で記述され、1 つ以上の空白文字で区切られていれば、ステートメントのどこにでも入れることができます。各制御ステートメントの 1 桁目は、最初のフィールドがラベルの場合を除いて、必ず空白にします。

• **ラベル・フィールド**

ラベルがある場合、ラベルは 1 桁目から始まり、ステートメント・ラベルに関するオペレーティング・システム要件に従っている必要があります。

• **命令フィールド**

このフィールドは、最初の行の 2 ~ 71 桁であればこの位置にでも配置できます。このフィールドには、プログラムにステートメントのタイプを識別させる語 (たとえば、SORT または MERGE) が含まれます。下記の例では、命令定義子 SORT がサンプル制御ステートメントの命令フィールド内にあります。

• **オペランド・フィールド**

オペランド・フィールドは、コンマまたはセミコロンで区切られた 1 つ以上のオペランドで構成されます。このフィールドは命令フィールドの後に続き、フィールドとフィールドの間は少なくとも 1 つの空白で区切られている必要があります。パラメーター内には空白を入れることはできませんが、すべてのパラメーターの終わりには空白が 1 つ必要です。ステートメントが複数行にまたがる場合は、最初の行でオペランドが開始される必要があります。各オペランドには、オペランド定義子またはパラメーター (オペランドのタイプを DFSORT に知らせる文字グループ) があります。1 つのパラメーターには、1 つ以上の値が関係していることがあります。オペランド形式には、次の 3 つがあります。

- パラメーター
- パラメーター=値
- パラメーター=(value1,value2...,valuen)

次の例は、これらの形式のそれぞれを示すものです。

`SORT EQUALS,FORMAT=CH,FIELDS=(10,30,A)`

• **注釈フィールド**

## 一般的なコーディングの規則

このフィールドには、どのような情報でも入れることができます。このフィールドは必須ではありませんが、使用する場合は、少なくとも 1 つの空白で最後のオペランド・フィールドと区切る必要があります。

### • 継続表示桁 (72)

この桁に空白以外の文字が入っていると、いまあるステートメントが次の行に継続されることを意味します。ただし、行のオペランド・フィールドの最後の文字が、コンマまたはセミコロンでその後に空白が続いている限り、プログラムは次の行を継続行と見なします。72 桁目に空白以外の文字が必要になるのは、注釈フィールドが継続される場合、またはオペランドが 71 桁目で中断される場合だけです。

### • 73 ~ 80 桁

このフィールドはどのような目的にも使用できます。

## 継続行

DFSORT 継続行の形式は、図 6 に示すとおりです。

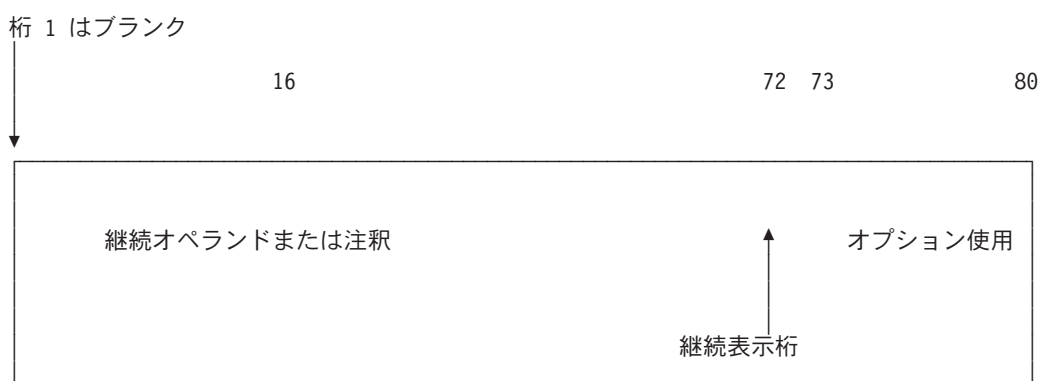


図 6. 継続行の形式

継続行の継続表示桁および 73 ~ 80 桁目までは、制御ステートメントの最初の行と同じ目的に使用します。1 桁目は、必ず空白である必要があります。

継続行は、その前の行を論理的に拡張したものと見なされます。オペランドまたは注釈フィールドは、複数の行にまたがる場合があります。例示のとおり、次の規則が適用されます。

- オペランド・フィールドが途切れた場合、または新規行から開始される場合、継続表示桁 (72 桁目) は空白以外の文字を指定し、継続行は 2 ~ 71 桁目のどこからでも開始できます。
- オペランド・フィールドがコンマまたはセミコロンの後で途切れた場合、継続表示桁 (72 桁目) は空白のままにしておくことができ、継続行は 2 ~ 71 桁目のどこからでも開始できます。71 桁目にコンマまたはセミコロンがあり、72 桁目に非空白文字がある場合は、16 桁目から継続を開始します。
- オペランド・フィールドがコンマまたはセミコロンの後で途切れていない場合、オペランド・フィールドは 71 桁目で切る必要があります。72 桁目には非空白文字が含まれます。継続は、16 桁目から開始します。

```

1          16          72
↓          ↓          ↓
SORT FIELDS=(5,8,A,20,2,D),
  FORMAT=CH
OPTION SKIPREC=2,LIST,  SKIP 2 RECORDS—LIST CONTROL STATEMENTS—
  DYNALLOC              USE DYNAMIC ALLOCATION
INCLUDE COND=(1,10,CH,EQ,C'STOCKHOLM',AND 21,8,ZD,GT,+500,OR,31,4,CH,N*
  E,C'HERR')

```

図 7. 正しい継続行の例:

## 注釈ステートメントの挿入

- 注釈ステートメントは、1 桁目にアスタリスク (\*) をコーディングして指定します。注釈ステートメントは、他の DFSORT プログラム制御ステートメントと一緒に印刷されますが、その他の処理は行われません。
- 1 ~ 71 桁がブランクのステートメントは、注釈ステートメントとして扱われません。
- 注釈ステートメントは、DFSPARM、SYSIN、および SORTCNTL データ・セットのみに使用できます。

## コーディング上の制約事項

制御ステートメントを作成する場合は、次の規則が適用されます。

- ラベル、命令定義子、およびオペランドは、大文字の EBCDIC で指定します。
- 各制御ステートメントの 1 桁目は、ラベルまたは注釈ステートメントの場合 (1 桁目がアスタリスクで始まる) のみ使用できます。
- ラベルは 1 桁目から始まり、ステートメント・ラベルに関するオペレーティング・システム要件に従う必要があります。
- 命令定義子はすべて、制御ステートメントの 1 行目に含まれます。
- 第 1 オペランドは、制御ステートメントの最初の行から開始します。ステートメントの最後のオペランドには、少なくとも 1 つのブランクが続く必要があります。
- オペランド内にブランクを使用できません。ブランクに続くものはすべて、注釈フィールドの一部と見なされます。
- 通常、値には 8 文字までの英数字を入れることができます。レコード数を指定する値 (たとえば、SKIPREC、STOPAFT、および FILSZ などの値) には、最後の 15 桁が (ゼロ以外の) 有効数字である、最大 28 桁までを含めることができます。LOCALE には、最大 32 桁の英数字の値を指定できます。
- コンマ、セミコロン、およびブランクは区切り文字としてだけ使用できます。値の中で使用できるのは、その値が定数の場合だけです。
- プログラム制御ステートメントの各タイプは、単一のソース (たとえば、SYSIN データ・セット) の中では 1 回しか指定できません。

### EFS プログラムが有効な場合の EFS の制約事項

前述の事項のほか、EFS プログラムの制御ステートメントを作成する場合は、次の制約事項が適用されます。

- DFSORT 以外の命令定義子は、最大 8 バイトの長さにすることができます。

## 一般的なコーディングの規則

- オペランドのない命令定義子を使用できるのは次の場合だけです。
  - その命令定義子が SYSIN、SORTCNTL、または DFSPARM を介して指定された場合
  - その命令定義子が行上の唯一のものであった場合。72 桁目にはブランクが入ります。

## 他の IBM プログラムの制御ステートメントの使用

他の IBM 分類プログラムが使用する INPFIL 制御ステートメントは、受け入れられますが処理されません。ただし、INPFIL ステートメントの継続により、制御ステートメント・エラーが起こる場合があります。他の IBM プログラムの INPFIL ステートメントに含まれている情報は、DD ステートメントで DFSORT に与えられません。

DFSORT が OPTION 制御ステートメントを使用するため、他の IBM 分類プログラムのジョブ・ストリームに OPTION 制御ステートメントが含まれている場合、他のプログラムのパラメーターが DFSORT OPTION 制御ステートメント・パラメーターの規則に従っていなければ、DFSORT は終了します。

---

## ALTSEQ 制御ステートメント

▶▶ ALTSEQ CODE=(fftt)▶▶

ALTSEQ 制御ステートメントは、代替変換テーブル (ALTSEQ テーブル) の変更に使われます。指定した変更はすべて、標準 EBCDIC 変換テーブルに適用されます。変更された ALTSEQ テーブルは、インストール時のデフォルトの ALTSEQ テーブル (出荷時のデフォルトは EBCDIC 変換テーブル) を指定変更します。

ALTSEQ テーブルには、以下の 2 つの用途があります。

- 形式 AQ (または CHALT が有効な形式 CH) の SORT、MERGE、INCLUDE または OMIT フィールドに、代替照合シーケンスを適用します。この場合、ALTSEQ テーブルは、データ自体を変更するのではなく、照合する順序だけを変更するのに使用されます。ALTSEQ 制御ステートメントを指定せずに形式 AQ (または CHALT が有効な CH) を指定する場合、DFSORT はインストール時のデフォルトの ALTSEQ テーブルを使用します。

たとえば、\$ (X'5B') を大文字の Z (X'E9') の後の X'EA' の位置で照合するように指定する場合は、以下を指定します。

```
ALTSEQ CODE=(5BEA)
```

- TRAN=ALTSEQ の INREC、OUTREC、または OUTFIL OUTREC フィールドの文字を変換します。この場合、ALTSEQ テーブルは、実際のデータを変更するのに使用されます。ALTSEQ 制御ステートメントを指定せずに TRAN=ALTSEQ を指定する場合、DFSORT はインストール時のデフォルトの ALTSEQ テーブルを使用します。

たとえば、\$ (X'5B') を \* (X'5C') に変更する場合は、以下を指定します。

```
ALTSEQ CODE=(5B5C)
```

CODE



元の EBCDIC 照合位置および変更後の EBCDIC 照合位置を指定します。

**ff** ALTSEQ テーブルで位置が変更される文字を 16 進数で指定します。

**tt** ALTSEQ テーブルで文字の新しい位置を 16 進数で指定します。

パラメーターが指定される順序は重要ではありません。

注:

1. CHALT が有効な場合、形式 AQ の制御フィールドのほかに、形式 CH の制御フィールドが ALTSEQ テーブルを用いて照合されます。
2. SORT、MERGE、INCLUDE、または OMIT フィールドでロケール処理を使用する場合、CHALT は使用できません。特定のフィールドで代替順序処理が必要な場合は、形式 AQ を使用します。
3. ALTSEQ を使用すると、パフォーマンスが低下することがあります。

デフォルト: 通常は、インストール・オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## EBCDIC 照合順序の更新 — 例

### 例 1

```
SORT FIELDS=(18,20,AQ,A)
ALTSEQ CODE=(5BEA)
```

文字 \$ (X'5B') の照合を X'EA' の位置、すなわち大文字 Z (X'E9') の後に行います。

### 例 2

```
MERGE FIELDS=(25,7,A,1,10,D),FORMAT=CH
OPTION CHALT
ALTSEQ CODE=(F0B0,F1B1,F2B2,F3B3,F4B4,F5B5,F6B6,
F7B7,F8B8,F9B9)
```

数字 0 ~ 9 は大文字の前 (ただし、小文字の後) で照合されます。

## ALTSEQ 制御ステートメント

### 例 3

```
SORT FIELDS=(55,8,AQ,A)
ALTSEQ CODE=(C1F1,C2F2)
```

大文字 A (X'C1') は、数字 1 (X'F1') と同じ位置で照合され、大文字 B (X'C2') は、数字 2 (X'F2') と同じ位置で照合されます。

この ALTSEQ ステートメントは、A を 1 の前または 1 の後、あるいは B を 2 の前または 2 の後に照合するのではないことに注意してください。

### 例 4

```
SORT FIELDS=(55,8,AQ,A)
ALTSEQ CODE=(81C1,82C2,83C3,84C4,85C5,86C6,87C7,
88C8,89C9,91D1,92D2,93D3,94D4,95D5,96D6,
97D7,98D8,99D9,A2E2,A3E3,A4E4,A5E5,A6E6,
A7E7,A8E8,A9E9)
```

小文字はそれぞれ対応する大文字と同じ位置で照合されます。たとえば、小文字 (X'81') は、大文字 A (X'C1') と同じ位置で照合されます。このため、大文字と小文字の区別が照合に影響します。

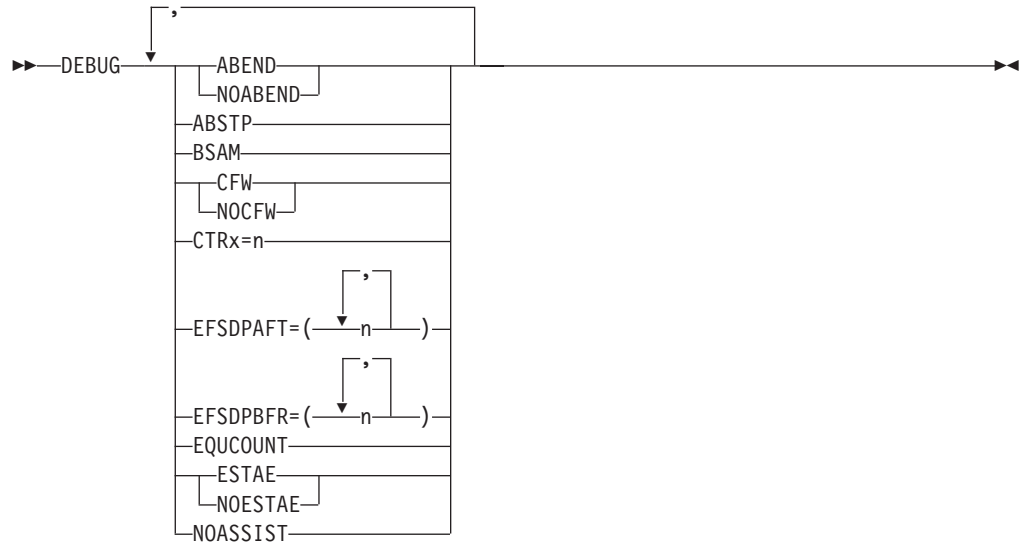
### 例 5

```
OPTION COPY
ALTSEQ CODE=(0040)
OUTREC FIELDS=(1,80,TRAN=ALTSEQ)
```

2 進ゼロ (X'00') はそれぞれスペース (X'40') に変更されます。

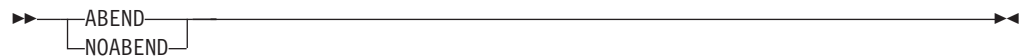


## DEBUG 制御ステートメント



DEBUG 制御ステートメントは、常に使用されるものではありません。一般的に使用されるのは、ABEND、NOABEND、BSAM のみです。テープ作業の分類または従来の組み合わせの場合は、DEBUG ステートメントの ABEND または NOABEND 以外のパラメーターは使用できません。問題診断の詳細については、*DFSORT* メッセージ、コード、および診断の手引き リリース 14を参照してください。

### ABEND または NOABEND



ERET インストール・オプションを一時的に指定変更して、分類、コピー、または組み合わせが成功しなかった場合に、*DFSORT* を異常終了または戻りコード 16 で終了させるかどうかを指定します。

#### ABEND

分類、コピー、または組み合わせが成功しなかった場合に、該当するメッセージ番号と同じユーザー完了コードか、もしくはインストール時に ICEMAC オプションの ABCODE=n でセットされた 1 ~ 99 の間のユーザー定義の番号で、*DFSORT* が異常終了することを指定します。

DEBUG ABEND が有効な場合、テープ作業データ・セットの分類または従来の組み合わせが成功しなかったときに、ユーザー異常終了コード 0 を出すことができます。

#### NOABEND

成功しなかった分類、コピー、または組み合わせが、戻りコード 16 で終了することを指定します。

**注:** *DFSORT* が、SmartBatch パイプ・データ・セットが使用中であると判別すると、ABEND オプションを自動的に強制実行し、エラーが検出された場合

## DEBUG 制御ステートメント

は異常終了させます。これにより、同じ SmartBatch パイプ・データ・セットにアクセスしている可能性のある他のアプリケーションに対して、システムが適切なエラー通知を行います。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### ABSTP

ブロック・セット終了時にダンプ内の必要な情報が失われないようにします。このオプションは、ERET、ABEND、および NOABEND に優先します。DFSORT アプリケーションが正常に終了しなかった場合、該当するメッセージ番号と等価の完了コード、あるいはインストール時に ICEMAC オプションの ABCODE=MSG または ABCODE=n でセットされたユーザー ABEND コードで強制的に異常終了します。NOESTAE が有効でない場合は、メッセージは書き込まれません。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### BSAM

▶▶—BSAM—▶▶

入出力データ・セットに使用される EXCP アクセス方式を、一時的にバイパスします。BSAM は、VSAM 入出力データ・セットの場合は無視されます。ブロック・セットが選択されておらず、BSAM 処理が連結 SORTIN 入力を用いて使用され、さらにヌルのデータ・セットとヌル以外のデータ・セットの両方が指定されている場合は、非ヌル・データ・セットより前のデータ・セットはすべてヌル・データ・セットである必要があります。それ以外の場合は、予期せぬ結果を招くことがあります。

注: このオプションはパフォーマンスを低下させる可能性があります。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### CFW または NOCFW

▶▶—CFW  
      └─NOCFW—▶▶

## DEBUG 制御ステートメント

CFW インストール・オプションを一時的に変更して、キャッシュされた 3990 制御装置に接続された装置上に常駐する SORTWKdd データ・セットを処理する場合、DFSORT がキャッシュ高速書き込みを使用するかどうかを指定します。

### CFW

SORTWKdd データ・セットの処理時に、DFSORT がキャッシュ高速書き込みを使用することを指定します。

### NOCFW

DFSORT がキャッシュ高速書き込みを使用できないことを指定します。

**注:** NOCFW オプションはパフォーマンスを低下させる可能性があります。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### CTRx

▶▶—CTRx=n—▶▶

入力または出力レコード数を保持し、その数が n に達するとコード 0C1 を出して異常終了します。x に割り当てることができる数値は以下のとおりです。

- 2 入力バッファから移送される入力レコード数 (コピーの場合は使用されない)
- 3 出力バッファから移送される出力レコード数 (コピーまたは組み合わせの場合は使用されない)
- 4 E15 により挿入される入力レコード数 (ブロック・セットの場合は使用されない)
- 5 E35 により削除される出力レコード数 (ブロック・セットの場合は使用されない)

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EFSDP AFT

▶▶—EFSDP AFT=()—▶▶

EFS プログラムに対するメジャー・コールの後に、SNAP ダンプを開始します。数字を任意に組み合わせて指定できます。

## DEBUG 制御ステートメント

数字の意味は次のとおりです。

- 2 EFS プログラムへのメジャー・コール 2 の後で SNAP ダンプを取得する
- 3 EFS プログラムへのメジャー・コール 3 の後で SNAP ダンプを取得する
- 4 EFS プログラムへのメジャー・コール 4 の後で SNAP ダンプを取得する
- 5 EFS プログラムへのメジャー・コール 5 の後で SNAP ダンプを取得する

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EFSDPBFR



EFS プログラムに対するメジャー・コールの前に、SNAP ダンプを開始します。数字を任意に組み合わせて指定できます。

数字の意味は次のとおりです。

- 2 EFS プログラムへのメジャー・コール 2 の前に SNAP ダンプを取得する
- 3 EFS プログラムへのメジャー・コール 3 の前に SNAP ダンプを取得する
- 4 EFS プログラムへのメジャー・コール 4 の前に SNAP ダンプを取得する
- 5 EFS プログラムへのメジャー・コール 5 の前に SNAP ダンプを取得する

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EQUCOUNT



ブロック・セット手法により分類された (メッセージ ICE184I に印刷された) 等しいキー (つまり、重複キー) をもつレコード数を判別します。可変長レコードの場合、EQUCOUNT はハイパー空間 (ハイパー分類の使用時) または作業データ・セットでしか使用できません。

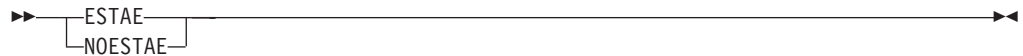
**注:**

1. EQUCOUNT オプションは、パフォーマンスを低下させる可能性があります。
2. ICETOOL の UNIQUE および OCCUR 演算子には、固有キーまたは非固有キーの報告機能があり、アプリケーションにおいて EQUCOUNT より便利な場合があります。
3. VLSHRT が有効な場合、EQUCOUNT は使用されません。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**ESTAE または NOESTAE**



ESTAE インストール・オプションを一時的に変更して、DFSORT が実行の早い時点で ESTAE リカバリー・ルーチンを削除するか、実行の全期間を通して使用するかどうかを決めます。

DFSORT は通常、実行の開始時に ESTAE リカバリー・ルーチンを設定します。異常終了が起きたときに ESTAE オプションが有効であれば、システムはこのリカバリー・ルーチンへ制御を渡します。ルーチンは次の操作を試みた後、実行を打ち切ります。

- 追加の異常終了情報の印刷
- SORTOUT 出力の正常終了後、分類、組み合わせ、またはコピー・アプリケーションを継続
- 終結処置およびハウスキーピングのためのメジャー・コール 4 と 5 での EFS プログラムに対する呼び出し
- SMF レコードの書き出し
- ICETEXIT 終了出口に対する呼び出し

異常終了が起きたときに ESTAE オプションが有効でなければ、これらの機能は実行されないこともあります。

**ESTAE**

DFSORT が実行の全期間を通して ESTAE リカバリー・ルーチンを使用できることを指定します。

**NOESTAE**

DFSORT が処理の早い時点で ESTAE リカバリー・ルーチンを削除するこ

## DEBUG 制御ステートメント

とを指定します。DFSORT がこの時点に到達する前に終了または異常終了した場合は、ESTAE リカバリー・ルーチンは削除されません。すなわち、NOESTAE は有効になりません。

**注:** DFSORT ESTAE リカバリー・ルーチンの詳細については、747 ページの『付録 E. DFSORT の異常終了処理』を参照してください。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## NOASSIST

▶▶—NOASSIST—◀◀

使用可能な場合、DFSORT はシステム /370-XA 分類命令を使用します。これらの命令を使用したくない場合は、このパラメーターを指定することにより一時的にバイパスできます。

**注:** このオプションはパフォーマンスを低下させる可能性があります。

**デフォルト:** なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## 診断オプションの指定 — 例

### 例 1

```
SORT FIELDS=(1,4,CH,A)
DEBUG EQUCCOUNT
```

入力レコードに以下のキーが含まれる場合、

KEYA、KEYA、KEYB、KEYB、KEYC、KEYD、KEYD、KEYE

次のメッセージが発行されます。

```
ICE184I THE NUMBER OF RECORDS SORTED WITH EQUAL KEYS IS 3
```

この 3 つの等しいキーは、KEYA、KEYB、および KEYD です。

**注:** ICETOOL の UNIQUE および OCCUR 演算子には完全な重複キーの報告機能がありますので、EQUCCOUNT の代わりに使用してください。

## 例 2

```
SORT FIELDS=(12,2,BI,D)
DEBUG BSAM,ABEND
```

SORTIN および SORTOUT データ・セットに BSAM アクセス方式を使用し、分類処理が成功しなかった場合に異常終了するように、DFSORT に指示します。

---

**END 制御ステートメント**

▶▶—END—◀◀

END 制御ステートメントにより、DFSORT はファイルの終わり (EOF) の前に、SYSIN、DFSPPARM、または SORTCNTL の読み取りを中断できます。

ユーザー出口ルーチンを動的にリンク・エディットする場合、END ステートメントは DFSORT 制御ステートメントの終わりと SYSIN の出口ルーチンのオブジェクト・デックの始まりにマークを付けます。

**制御ステートメント読み取りの中断 — 例**

## 例 1

```
//SYSIN DD *
SORT FIELDS=(1,6,A,28,5,D),FORMAT=CH
RECORD TYPE=V,LENGTH=(200,,80)
END
OPTION DYNALLOD
```

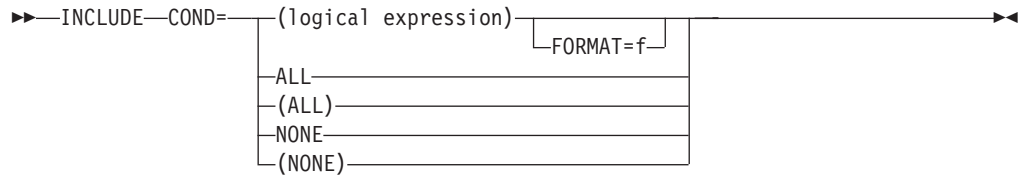
OPTION ステートメントは END ステートメントの後にあるため、読み取られません。

## 例 2

```
//SYSIN DD *
SORT FIELDS=(5,8,CH,A)
MODS E15=(E15,1024,SYSIN,T)
END
◀object deck for E15 user exit here▶
```

END ステートメントは、SYSIN 内で E15 ユーザー出口ルーチンのオブジェクト・デックより前にあります。

## INCLUDE 制御ステートメント

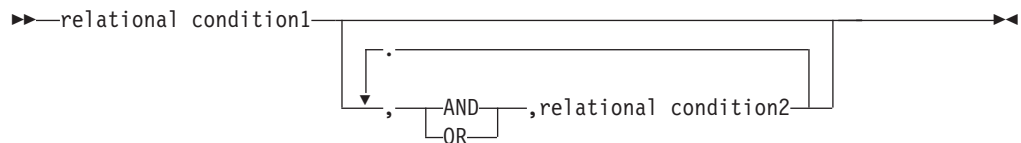


特定のレコードだけを出力データ・セットに入れたい場合に、INCLUDE ステートメントを使用します。INCLUDE ステートメントでは、組み込みたいレコードを選択します。

同じ DFSORT 実行において、INCLUDE ステートメントまたは OMIT ステートメントのどちらか一方のみを指定できます。

DFSORT が短い INCLUDE/OMIT 比較フィールドを処理する方法は、VLSCMP/NOVLSCMP および VLSHRT/NOVLSHRT の設定により異なります。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張されるようなフィールドを指します。短いレコードの組み込みまたは省略の詳細については、181 ページの『OPTION 制御ステートメント』の VLSCMP および NOVLSCMP オプションの説明を参照してください。

論理式は、入力レコードの各フィールドに基づいて論理的に組み合わせられた 1 つ以上の関係条件であり、以下の高度レベルで表すことができます。



あるレコードについて論理式が真ならば、そのレコードは出力データ・セットに組み込まれます。

次の 4 種類の関係条件を使用できます。

### 1. 比較:

2 つの比較フィールド間の比較、あるいは 1 つの比較フィールドと 10 進数、16 進数、文字または現在日付の定数の間の比較を行うことができます。たとえば、各レコードの最初の 6 バイトと最後の 6 バイトを比較して、これらのフィールドが同じレコードだけを組み込むことができます。あるいは、フィールドと現在の日付を比較して、今後のイベントのレコードだけを組み込むこともできます。

比較の詳細については、133 ページの『比較』を参照してください。

### 2. サブstring比較テスト:

フィールド値内の定数または定数内のフィールド値を検索します。



## INCLUDE 制御ステートメント

たとえば、6 バイト・フィールド内の値を検索して、文字定数 'C'OK' を見つけ出し、フィールドのどこかに 'C'OK' のあるレコードだけを組み込むことができます。あるいは、3 バイト・フィールドの値で文字定数 'C'J69,L92,J82' を検索して、フィールド内に 'C'J69'、'C'L92'、または 'C'J82' が見つかるレコードだけを組み込むことができます。

サブストリング比較テストについては、141 ページの『サブストリング比較テスト』を参照してください。

### 3. ビット論理テスト:

ビット・マスクまたは 16 進数マスク、あるいはビット定数を使用して、2 進数フィールドの選択したビットの状態 (オンまたはオフ) をテストします。

たとえば、1 バイト・フィールドのビット 0 およびビット 2 がオンのレコードだけを組み込むことができます。あるいは、2 バイト・フィールドのビット 3 と 12 がオンで、ビット 6 と 8 がオフのレコードだけを組み込むことができます。

ビット論理テストについては、143 ページの『ビット論理テスト』を参照してください。

### 4. 日付比較:

有効な「世紀」ウィンドウを使用して、2 桁の年日付フィールドを 2 桁の年日付定数、現在またはその他の 2 桁の年日付フィールドと比較します。

たとえば、Z'yymm' 日付フィールドが、1996 年 1 月と 2005 年 3 月の間にあるレコードだけを組み込むことができます。あるいは、P'dddy' フィールドが、別の P'dddy' フィールドより小さいレコードだけを組み込むことができます。

日付比較の詳細については、149 ページの『日付比較』を参照してください。

括弧内に関係条件をネストすることにより、さらに複雑な論理式を作成できます。

わかりやすくするために、比較、サブストリング比較テスト、ビット論理テスト、および日付比較については以下で別々に説明していますが、これらを組み合わせて論理式を作成できます。

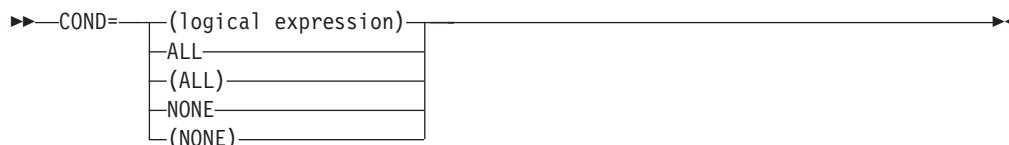
INCLUDE ステートメントと OUTFIL ステートメントの INCLUDE パラメーターは、次の点で異なります。

- INCLUDE ステートメントはすべての入力レコードに適用されますが、INCLUDE パラメーターの場合は OUTFIL グループの OUTFIL 入力レコードのみに適用されます。
- FORMAT=f は、INCLUDE ステートメントでは指定できますが、INCLUDE パラメーターでは指定できません。
- D2 形式は、INCLUDE ステートメントでは指定できますが、INCLUDE パラメーターでは指定できません。

OUTFIL INCLUDE パラメーターの詳細については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

## COND

## INCLUDE 制御ステートメント



### logical expression

入力レコードのフィールドに基づいて、論理的に組み合わせられた 1 つ以上の関係条件を指定します。あるレコードについて論理式が真ならば、そのレコードは出力データ・セットに組み込まれます。

### ALL または (ALL)

すべての入力レコードが出力データ・セットに組み込まれるように指定します。

### NONE または (NONE)

入力レコードが出力データ・セットに組み込まれないことを指定します。

デフォルト: ALL。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## FORMAT



FORMAT=f が使用できるのは、論理式全体の入力フィールドすべてが同じ形式の場合のみです。比較に指定可能なフィールド形式は、134 ページの表 10 に示されています。SS (サブストリング) は、サブストリング比較テストで唯一指定できるフィールド形式です。BI (無符号 2 進数) は、ビット論理テスト用に認められている唯一のフィールド形式です。Y2x 形式は、日付比較の場合に使用できる唯一のフィールド形式です。

デフォルト: なし。COND=(logical expression) パラメーターに組み込まれていない場合は、必ず指定する必要があります。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**注:** FORMAT と COND の両方に形式値を指定すると、DFSORT は通知メッセージを出し、COND の形式値を使用し (f は各制御フィールドごとに指定します)、FORMAT で指定した形式値は使用しません。

## 関係条件

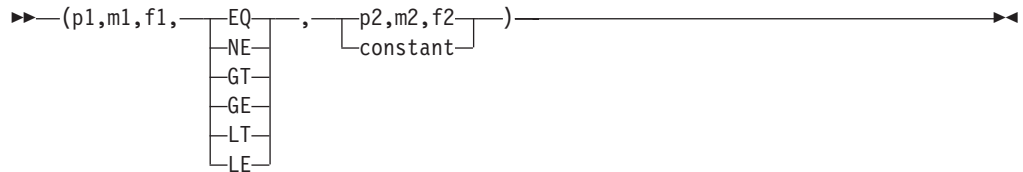
関係条件により、比較またはビット論理テストの実施を指定します。関係条件は、AND または OR を用いて論理的に組み合わせ、論理式を作成できます。組み合わせる場合、以下の規則が適用されます。

- AND ステートメントは、括弧を使用して評価の順序を変更しない限り、OR ステートメントの前に評価されます。括弧内の式は、常に最初に評価されます。(括弧のネストを制限するのは、使用可能記憶域の大きさのみです。)
- AND と OR の語の代わりに、それぞれ記号の & と | を使用できます。

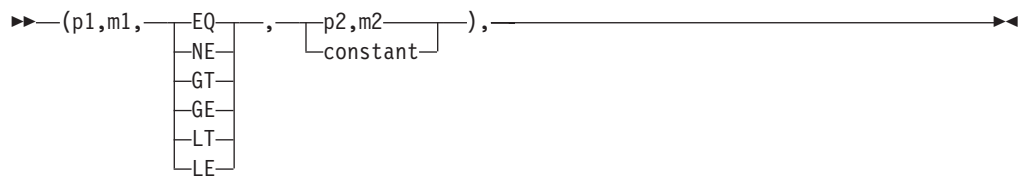
## 比較

### 関係条件の形式

関係条件には、次の 2 つの形式を使用できます。



または、FORMAT=f オペランドを使用する場合、



比較演算子には次のものがあります。

- EQ** 等しい
- NE** 等しくない
- GT** より大きい
- GE** 以上
- LT** より小さい
- LE** 以下

### フィールド:

**p1,m1,f1:** これらの変数は、入力レコード内の別のフィールドまたは定数のどちらかと比較する、入力レコードのフィールドを指定します。

- p1 は、比較フィールドの最初のバイトが、入力レコードの先頭から相対的にどの位置にあるかを指定します。<sup>7</sup> 固定長レコード (FLR) の最初のデータ・バイトは、相対位置が 1 になります。可変長レコード (VLR) の最初のデータ・バイトは、相対位置が 5 (最初の 4 バイトにはレコード記述子が入るため) になります。すべての比較フィールドは、バイト境界から開始します。また、どの比較フィールドも 32752 バイトを超えることはできません。
- m1 は、比較フィールドの長さを指定します。それぞれの形式に指定できる長さは、134 ページの表 10 のとおりです。

7. E15 ユーザー出力ルーチンがレコードを形式化する場合、p1 は、その出口により再フォーマット設定されたときのレコードを指定するものである必要があります。

## INCLUDE 制御ステートメント

- f1 は、比較フィールドのデータ形式を指定します。指定できる形式は、表 10 のとおりです。

すべての比較フィールドに同じタイプのデータが含まれている場合は、この値を省略できますが、その場合は FORMAT=f オペランドを使用します。

表 10. 比較フィールドの形式と長さ

形式コード	長さ	説明
CH	1 ~ 256 バイト	文字 <sup>8</sup>
AQ	1 ~ 256 バイト	代替照合順序をもつ文字
ZD	1 ~ 256 バイト	ゾーン 10 進数、符号あり
PD	1 ~ 255 バイト	パック 10 進数、符号あり
FI	1 ~ 256 バイト	固定小数点、符号あり
BI	1 ~ 256 バイト	2 進数、符号なし
AC	1 ~ 256 バイト	ISCI/ASCII 文字
CSF または FS	1 ~ 16 バイト	任意指定の浮動符号が左端にある符号付き数字
CSL または LS	2 ~ 256 バイト	区切り符号が左端にある符号付き数字
CST または TS	2 ~ 256 バイト	符号桁が右端にある数字
CLO または OL	1 ~ 256 バイト	符号が左端にオーバーパンチされている数字
CTO または OT	1 ~ 256 バイト	符号が右端にオーバーパンチされている数字
ASL	2 ~ 256 バイト	符号桁が左端にある ISCI/ASCII 数字
AST	2 ~ 256 バイト	符号桁が右端にある ISCI/ASCII 数字
D2	1 ~ 256 バイト	ユーザー定義のデータ・タイプ (EFS プログラムが必要)

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

**p2, m2, f2:** これらの変数には、p1,m1,f1 フィールドが比較される入力レコード内の別のフィールドを指定します。形式の異なる比較フィールド間で可能な比較は、表 11 に示すとおりです。

AC、ASL、および AST 形式は、ISCI/ASCII 照会順序を使用して EBCDIC データの順序付けを行います。

表 11. INCLUDE/OMIT に指定可能なフィールド間の比較

フィールド形式	BI	CH	ZD	PD	FI	AC	ASL	AST	CSF または FS	CSL または LS	CST または TS	CLO または OL	CTO または OT	AQ	D2
BI	X	X													

8. CHALT が有効な場合、CH は AQ として扱われます。

表 11. INCLUDE/OMIT に指定可能なフィールド間の比較 (続き)

フィールド形式	BI	CH	ZD	PD	FI	AC	ASL	AST	CSF または FS	CSL または LS	CST または TS	CLO または OL	CTO または OT	AQ	D2
CH	X	X													
ZD			X	X											
PD			X	X											
FI					X										
AC						X									
ASL							X	X							
AST							X	X							
CSF または FS									X	X	X				
CSL または LS									X	X	X				
CST または TS									X	X	X				
CLO または OL												X	X		
CTO または OT												X	X		
AQ														X	
D2															X

注: D2 フィールドの形式はユーザーにより定義されます。

**定数:** 定数には、10 進数 (n, +n, -n)、文字ストリング、(C'xx...x')、または 16 進数ストリング (X'yy...yy') が使用可能です。現在日付もまた、10 進数 (DATE1P、DATE2P、DATE3P) または文字ストリング (DATE1、DATE1(c)、DATE2、DATE2(c)、DATE3、DATE3(c)) として使用できます。各種の定数の詳細が、次に示してあります。形式の異なる比較フィールド間で可能な比較は、表 12 に示すとおりです。

表 12. INCLUDE/OMIT に指定可能なフィールドと定数の比較

フィールド形式	自己定義項		
	10 進数	文字ストリング	16 進数ストリング
BI	X	X	X
CH		X	X
ZD	X		
PD	X		
FI	X		
AC		X	X
ASL	X		
AST	X		

## INCLUDE 制御ステートメント

表 12. INCLUDE/OMIT に指定可能なフィールドと定数の比較 (続き)

フィールド形式	自己定義項		
	10 進数	文字ストリング	16 進数ストリング
CSF または FS	X		
CSL または LS	X		
CST または TS	X		
CLO または OL	X		
CTO または OT	X		
AQ		X	X
D2	X	X	X

注: D2 フィールドの形式はユーザーにより定義されます。

**10 進数の形式:** 10 進定数のコーディング形式は次のとおりです。

$[\pm]n$

FI フィールドを10 進定数と比較する場合、n または +n は +2147483647 より小さい値、-n は -2147483648 より大きい値である必要があります。

BI フィールドを 10 進定数と比較する場合、n または +n は +4294967295 より小さく、+0 より大きい値である必要があります。BI フィールドを負数 (-n) と比較することはできません。NOSZERO が有効な場合でも、BI フィールドを -0 と比較できません。

有効な 10 進数定数および無効な 10 進数定数の例は、次のとおりです。

有効	無効	説明
15	++15	符号文字が多すぎる
+15	15+	符号の位置が正しくない
-15	1.5	無効な文字を含んでいる
18000000	1,500	無効な文字を含んでいる

図 8. 有効な 10 進数定数と無効な 10 進数定数

**10 進数の現在日付:** DATE1P、DATE2P、または DATE3P は実行日付を表す 10 進数の生成に使用されます。表 13 に、現在日付オペランドごとにそれぞれ生成された 10 進数の例を示します。yyyy は年、mm は月 (01 ~ 12)、dd は日付 (01 ~ 31)、ddd は年間通日 (001 ~ 366) を表します。

表 13. 10 進数の現在日付オペランド

オペランド	定数	2001 年 4 月 19 日
DATE1P	+yyyymmdd	+20010419
DATE2P	+yyyymm	+200104
DATE3P	+yyyddd	+2001109

**文字ストリングの形式:** 文字ストリング定数のコーディング形式は次のとおりです。

**C'xx...x'**

値 x は任意の EBCDIC 文字の場合があります (EBCDIC 文字ストリングは、AC または AQ フィールドとの比較の際に適切に変換されます)。最大 256 文字を指定できます。

文字ストリング内に単一のアポストロフィを組み込む場合は、2 つの単一アポストロフィで指定します。したがって、次のようになります。

Required: 0'NEILL Specify: C'0'NEILL'

文字ストリング定数の有効な例と無効な例は、次のとおりです。

有効	無効	説明
C'JDCO'	C''''	アポストロフィが対になっていない
C'\$@#'	'ABCDEF'	C 識別子がない
C'+0.193'	C'ABCDEF	アポストロフィがない
C'Frank's'	C'Frank's'	1 つのアポストロフィに対して 2 つの単一アポストロフィが必要

図 9. 有効な文字ストリング定数と無効な文字ストリング定数

INCLUDE/OMIT 比較の場合、文字ストリングで 2 バイト・データが使用される場合があります。2 バイト・データは、シフト・アウト (SO) 制御文字 (X'0E') で始め、シフト・イン (SI) 制御文字 (X'0F') で終わらせることで区切ります。SO および SI 制御文字は文字ストリングの一部であり、ゼロとまたは偶数の中継バイトと組み合わせます。ネストされたシフト・コードは使用できません。SO と SI の間にあるすべての文字は、有効な 2 バイト文字で指定します。2 バイト・データからは、単一バイトの意味を引き出すことはできません。

2 バイト文字を含む文字ストリング定数の有効な例と無効な例を以下の記号を用いて次のように示します。

- < は SO を示します
- > は SI を示します
- Dn は 2 バイト文字を示します

有効	無効	説明
C'Q<D1D2>T'	C'Q<R>S'	SO/SI 内の単一バイト・データ
C'<D1D2D3>'	C'D1D2D3'	SO/SI がない。単一バイト・データとして扱われる
C'Q<D1>R<D2>'	C'Q<D1<D2>>'	ネストされた SO/SI

図 10. 2 バイト・データをもつ有効なストリングと無効なストリング

**文字ストリングの現在日付:** DATE1、DATE1(c)、DATE2、DATE2(c)、DATE3 および DATE3(c) は、実行日付を表す文字ストリングの生成に使用されます。 138 ページの

## INCLUDE 制御ステートメント

ーの表 14 に、現在日付オペランドごとにそれぞれ生成された文字ストリングの例を、関連箇所 (c) に対し (l) を使用して、示します。yyyy は年、mm は月 (01 ~ 12)、dd は日付 (01 ~ 31)、ddd は年間通日 (001 ~ 366) を表します。また、c はブランク 以外 のあらゆる文字として使用できます。

表 14. 文字ストリングの現在日付オペランド

オペランド	定数	2001 年 4 月 19 日
DATE1	C'yyyymmdd'	C'20010419'
DATE1(c)	C'yyyycmmdd'	C'2001/04/19'
DATE2	C'yyyymm'	C'200104'
DATE2(c)	C'yyyycmm'	C'2001/04'
DATE3	C'yyyddd'	C'2001109'
DATE3(c)	C'yyyycddd'	C'2001/109'

**16 進数ストリングの形式:** 16 進数ストリング定数のコーディング形式は、次のとおりです。

**X'yy...yy'**

値 yy は、16 進数の任意の組み合わせを示します。最大 256 組の 16 進数が指定できます。

16 進数定数の有効な例および無効な例は、次のとおりです。

有効	無効	説明
X'ABCD'	X'ABGD'	無効な 16 進数
X'BF3C'	X'BF3'	数字が対になっていない。
X'AF050505'	'AF050505'	X 識別子がない。
X'BF3C'	'BF3C'X	X 識別子の位置が正しくない

図 11. 有効な 16 進数定数と無効な 16 進数定数

### 埋め込みと切り捨て

フィールド同士の比較では、短いほうのフィールドが適切に埋め込まれます。フィールドと定数の比較では、定数が比較フィールドの長さまで埋め込まれるか切り捨てられます。

文字ストリングと 16 進数ストリングの切り捨てと埋め込みは、右端で行なわれます。

埋め込み文字は次のとおりです。

- 文字ストリングの場合 X'40'
- 16 進数ストリングの場合 X'00'

10 進定数の埋め込みと切り捨ては、左端で行われます。埋め込みは正しい形式のゼロを用いて行われます。



## 文化的環境についての考慮事項

DFSORT の照合の動作を、使用する文化環境に合わせて修正できます。文化環境は、活動ロケールを選択することにより設定されます。活動状態のロケールの照合規則は、DFSORT の INCLUDE と OMIT の処理に対して、次のような影響を与えます。

- DFSORT は、活動ロケールで定義された照合規則に従って、出力用レコードを組み込み、または除外します。これにより、言語の文化特性およびロケール特性を維持した照合規則の定義に基づいて、単一または複数バイト文字データの組み合わせや除外を行うことができます。

ロケール処理を使用する場合は、文字 (CH) 比較フィールド、および文字 (CH) 比較フィールドに対して比較される、文字定数と 16 進定数を処理する場合のみ、活動ロケールが使用されます。

ロケールの処理の詳細については、7 ページの『文化的環境についての考慮事項』または 181 ページの『OPTION 制御ステートメント』の LOCALE の項を参照してください。

## 出力データ・セットへのレコードの組み込み - 比較の例

### 例 1

```
INCLUDE COND=(5,8,GT,13,8,|,105,4,LE,1000),FORMAT=CSF
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 5 ~ 12 バイトにある浮動符号付き数が、13 ~ 20 バイトにある浮動符号付き数より大きい場合

OR

- 105 ~ 108 バイトにある浮動符号付き数が 1000 以下の場合

3 つの比較フィールドがすべて同じ形式であることに注意してください。

### 例 2

```
INCLUDE COND=(1,10,CH,EQ,C'STOCKHOLM',
AND,21,8,ZD,GT,+50000,
OR,31,4,CH,NE,C'HERR')
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 最初の 10 バイトの値が STOCKHOLM (この 9 桁のストリングの右端にはブランクが埋め込まれている) で、かつ (AND)、21 ~ 28 バイトのゾーン 10 進数が 50 000 より大きい場合

OR

- 31 ~ 34 バイトの値が HERR ではない場合

AND (かつ) は OR (または) の前に評価されることに注意してください。(179 ページの『出力データ・セットからレコードを除外する - 例』は、括弧を用いて値を

## INCLUDE 制御ステートメント

求める順序を変更する方法を図示したものです。) また、コンマまたはセミコロンの後に空白を続けて行を終わらせると、パラメーターは 2 ~ 71 桁目の任意の位置から始まる次の行へ継続することを示すことになるので、注意してください。

### 例 3

```
INCLUDE COND=((5,1,CH,EQ,8,1,CH),&,
              ((20,1,CH,EQ,C'A',&,30,1,FI,GT,10),|,
              (20,1,CH,EQ,C'B',&,30,1,FI,LT,100),|,
              (20,1,CH,NE,C'A',&,20,1,CH,NE,C'B')))
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 5 バイトの値が 8 バイトに等しい場合  
AND
- 以下のいずれかが真の場合
  - 20 バイトが 'A' に等しく、かつ 30 バイトの値が 10 より大きい場合
  - 20 バイトが 'B' に等しく、かつ 30 バイトの値が 100 より小さい場合
  - 20 バイトが 'A' にも 'B' にも等しくない場合

### 例 4

```
INCLUDE COND=(7,2,CH,EQ,C'T1',OR,
              (1,2,BI,GE,X'001A',AND,20,2,CH,EQ,25,2,CH))
```

この例は、短いレコードが存在している場合の、INCLUDE 処理時の VLSCMP/NOVLSCMP および VLSHRT/NOVLSHRT の影響を示しています。

141 ページの図 12 に示すレコードについて考えてみます。

- VLSCMP が有効な場合、20 ~ 21 と 25 ~ 26 バイトの比較で短いフィールドが含まれる場合でも、7 ~ 8 バイトは C'T1' に等しいため、最初のレコードが組み込まれます。2 番目のレコードは、20 ~ 21 と 25 ~ 26 バイトの比較に基づいて、組み込まれたり省略されたりします。
- NOVLSCMP および VLSHRT が有効な場合、20 ~ 21 と 25 ~ 26 バイトの比較に短いフィールドが含まれるので、最初のレコードは省略されます。2 番目のレコードは、20 ~ 21 と 25 ~ 26 バイトの比較に基づいて、組み込まれたり省略されたりします。
- NOVLSCMP および NOVLSHRT が有効な場合、20 ~ 21 と 25 ~ 26 バイトの比較に短いフィールドが含まれるので、最初のレコードによりメッセージ ICE015A または ICE218A が出力されます。

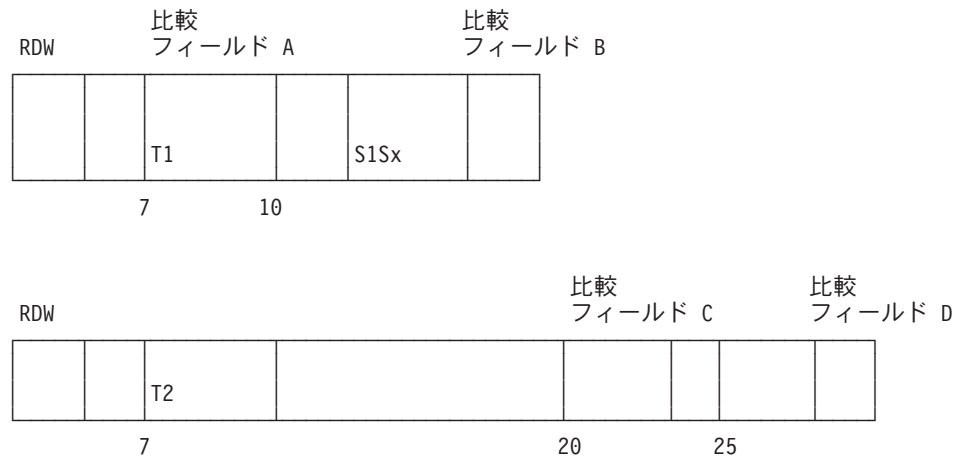


図 12. サンプルのレコード

### 例 5

```
INCLUDE COND=(21,8,ZD,GT,DATE1P)
```

この例は、21 ~ 28 バイトの形式 Z'yyyymmdd' のゾーン 10 進数日付が現在の日付より大きいレコードを、組み込む方法について示したものです。DATE1P は、現在日付 10 進数を形式 +yyyymmdd で生成します。

## サブstring比較テスト

サブstring比較テストには、次の 2 種類があります。

1. フィールド値内の定数を検出します。たとえば、6 バイト・フィールドでの値を検索して、文字定数 C'OK' を見つけます。たとえば、フィールド値が C'\*\*\*OK\*\*' または C'\*\*\*\*OK' の場合の関係条件は真で、フィールド値が '\*\*ERR\*\*' の場合は関係条件は偽です。
2. 定数内のフィールド値を検出します。たとえば、文字定数 C'J69,L92,J82' を検索して、3 バイト・フィールドの値を見つけてみます。フィールド値が C'J69'、C'L92'、または C'J82' の場合の関係条件は真で、フィールド値が C'X24' の場合の関係条件は偽です。定数内ではコンマを使って有効な 3 文字の値を分離していることに注意してください。フィールド値にない文字であれば、どの文字でも定数内の区切り文字として使うことができます。

### 関係条件の形式

関係条件には、次の 2 つの形式を使用できます。

▶▶ (p1,m1,SS,  $\begin{matrix} \text{EQ} \\ \text{NE} \end{matrix}$ , —constant—) ▶▶

または、FORMAT=SS オペランドを使用する場合、

## INCLUDE 制御ステートメント

▶▶(p1,m1, 

EQ
NE

, —constant—)▶▶

注: FORMAT=SS は COND の前に指定することはできませんが、その後に指定できません。

サブstring比較演算子には、次のものがあります。

**EQ** 等しい  
**NE** 等しくない

### フィールド:

**p1,m1:** これらの変数は、サブstring・テストのために入力レコードの文字フィールドを指定するものです。

- p1 は、サブstring・テストの文字入力フィールドの最初のバイトが、入力レコードの先頭から相対的にどの位置にあるかを指定します。<sup>9</sup> 固定長レコード (FLR) の最初のデータ・バイトは、相対位置が 1 になります。可変長レコード (VLR) の最初のデータ・バイトは、相対位置が 5 になります (最初の 4 バイトにレコード記述ワードが含まれるため)。テストされるすべてのフィールドは、必ずバイト境界から始まり、32752 バイトを超えてはなりません。
- m1 は、テストされるフィールドの長さを指定します。長さは 1 ~ 256 バイトです。

**定数:** 定数は、文字stringまたは 16 進数stringになります。詳細は、137 ページの『文字stringの形式』と 138 ページの『16 進数stringの形式』を参照してください。

m1 が定数の長さより大きい場合、フィールド値が定数を求めて検索され、EQ 比較演算子が指定されているときには、一致するものが見つかった場合に条件が真になり、NE 比較演算子が指定されているときには、一致するものが見つからなかった場合に条件が真になります。

m1 が定数の長さより小さい場合、定数がフィールド値を求めて検索され、EQ 比較演算子が指定されているときには一致するものが見つかった場合に条件が真になり、NE 比較演算子が指定されているときには一致するものが見つからなかった場合に条件が真になります。

## 出力データ・セットへのレコードの組み込み — サブstringの比較の例

### 例

```
INCLUDE FORMAT=SS,COND=(11,6,EQ,C'OK',OR,21,3,EQ,C'J69,L92,J82')
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 11 ~ 16 バイトのどこかに OK が見つかる場合

9. E15 ユーザー出口ルーチンがレコードを形式化する場合、p1 はその出口により再フォーマット設定されたときのレコードを指定するものである必要があります。

または

- 21 ~ 23 バイトに J69、L92 または J82 がある場合

## ビット論理テスト

ビット論理テストには次の 2 つの方法があります。

- 16 進数またはビット・マスクを用いるビット演算子
- ビット比較テスト

2 つの方法のいずれを使用してもビット論理テストを指定できますが、それぞれには他方にはない固有な利点があります。

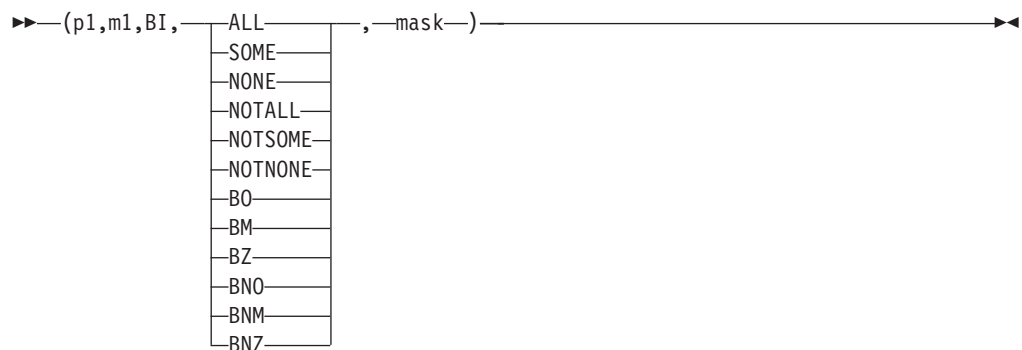
2 つの方法のどちらでも、フィールド内の特定のビットを指定することができ、必要な結果を得るために指定しなければならない INCLUDE 条件の数を大幅に減らすことができます。これは指定されなかったビットを数に入れる必要がなくなるためです。

### 方法 1: ビット演算子のテスト

この方法のビット論理テストでは、2 進数フィールドの選択ビットがすべてオンになっているか、すべてオフになっているか、オン / オフ混合状態か、またはこれらの状態の組み合わせが選択された状態かをテストできます。この方法により、Test Under Mask (TM) 機械語命令に類似した単一の命令で、多くの異なるビットの組み合わせをテストできます。ただし、あるフィールドがオンとオフのビットの特別な組み合わせであるかどうかを厳密に判別するためには、次に説明する方法 2 のほうが適しています。

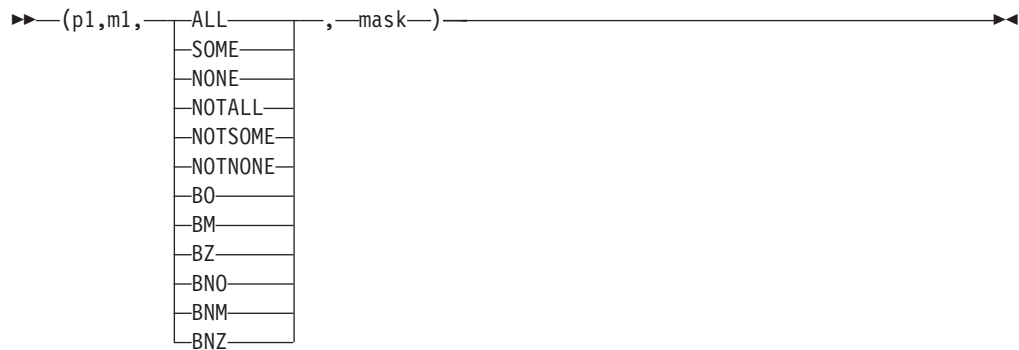
#### 関係条件の形式

関係条件には、次の 2 つの形式を使用できます。



または、FORMAT=BI オペランドを使用する場合、

## INCLUDE 制御ステートメント



ビット演算子は、入力フィールドとマスクの関係を次のようにテストするように記述します。

### ALL または BO

すべてのマスク・ビットが入力フィールドでオンです。

### SOME または BM

すべてではないが、いくつかのマスク・ビットが入力フィールドでオンです。

### NONE または BZ

入力フィールドですべてのマスク・ビットがオンではありません。

### NOTALL または BNO

入力フィールドのいくつかのマスク・ビットがオンであるか、またはすべてのマスク・ビットがオンではありません。

### NOTSOME または BNM

入力フィールドのすべてのマスク・ビットがオンであるか、またはすべてのマスク・ビットがオフです。

### NOTNONE または BNZ

入力フィールドのすべてのマスク・ビットまたはいくつかのマスク・ビットがオンです。

組の最初の演算子 (ALL、SOME、など) は、意味をもった簡略記号を使用する人のためのものです。組の 2 番目の演算子 (BO、BM、など) は、Test Under Mask (TM) 命令に関連する条件に慣れた人のためのものです。

## フィールド

**p1,m1:** これらの変数は、マスクに対してテストされる、入力レコードの 2 進数フィールドを指定するものです。

- p1 は、マスクに対してテストされ、入力レコードの先頭と相対的な、2 進入力フィールドの最初のバイトを指定します。<sup>10</sup> 固定長レコード (FLR) の最初のデータ・バイトは、相対位置が 1 になります。可変長レコード (VLR) の最初のデータ・バイトは、相対位置が 5 (最初の 4 バイトにはレコード記述子が入るため) になります。テストされるすべてのフィールドは、必ずバイト境界から始まり、32752 バイトを超えてはなりません。

10. E15 ユーザー出力ルーチンがレコードを形式化する場合、p1 はその出口により再フォーマット設定されたときのレコードを指定するものである必要があります。

- m1 は、テストされるフィールドの長さを指定します。長さは 1 ~ 256 バイトです。

### マスク

テストのために選択されたフィールド内のビットを表す、16 進数ストリングまたはビット・ストリングです。マスク・ビットがオン (1) の場合は、フィールド内の対応するビットがテストされます。マスク・ビットがオフ (0) の場合は、フィールド内の対応するビットは無視されます。

**16 進数ストリングの形式:** 16 進数ストリングのコーディング形式は次のとおりです。

**X'**yy...yy'

値 yy は、バイトを構成する 16 進数の任意の対 (8 ビット) を表します。各ビットは、1 (テスト・ビット) または 0 (無視ビット) で指定します。最大 256 組の 16 進数が指定できます。

**ビット・ストリングの形式:** ビット・ストリングのマスクのコーディング形式は次のとおりです。

**B'**bbbbbbbb...bbbbbbbb'

値 bbbbbbbb は、1 バイトを構成する 8 ビットを表します。各ビットは、1 (テスト・ビット) または 0 (無視ビット) で指定します。最大 256 までの 8 ビットのグループを指定できます。マスク内のビット数の合計は、8 の倍数で指定します。ビット・マスク・ストリングは、ビット演算子を用いる場合のみ使用できます。

## 埋め込みと切り捨て

16 進またはビット・マスクは、2 進数フィールドのバイト長になるように右端で切り捨てまたは埋め込みが行われます。埋め込み文字は、X'00' です (すべてのビットがオフであり、このためテストされません)。

## 出力データ・セットにレコードを組み込む - ビット演算子テストの例

### 例 1

```
INCLUDE COND=(27,1,CH,EQ,C'D',AND,18,1,BI,ALL,B'10000000')
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 27 バイト目が D の場合  
AND
- 18 バイト目のビット 0 がオンの場合

### 例 2

```
INCLUDE COND=(11,1,BI,BM,X'85')
```

## INCLUDE 制御ステートメント

この例では、11 バイト目のビット 0、5、および 7 のいくつか（ただし、全部ではなく）オンになっているレコードだけを組み込む方法を示しています。選択されたフィールド値に対する結果は、下記のとおりです。

表 15. ビット比較の例 2: 選択されたフィールド値の結果

11,1,BI 値	11,1,BI 結果	処置
X'85'	偽	レコードを除外
X'C1'	真	レコードを組み込む
X'84'	真	レコードを組み込む
X'00'	偽	レコードを除外

### 例 3

```
INCLUDE COND=(11,2,ALL,B'0001001000110100',
              OR,21,1,NONE,B'01001100'),FORMAT=BI
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 11 ~ 12 バイト目のビット 3、6、10、11、および 13 がすべてオンの場合  
または
- 21 バイト目のビット 1、4、および 5 がすべてオンではない場合

選択されたフィールド値に対する結果は、下記のとおりです。

表 16. ビット比較の例 3: 選択されたフィールド値の結果

11,2,BI 値	11,2,BI 結果	21,1,BI 値	21,1,BI 結果	処置
X'1234'	真	X'4C'	偽	レコードを組み込む
X'02C4'	偽	X'81'	真	レコードを組み込む
X'0204'	偽	X'40'	偽	レコードを除外
X'F334'	真	X'00'	真	レコードを組み込む
X'1238'	偽	X'4F'	偽	レコードを除外

## 方法 2: ビット比較テスト

この方法のビット論理テストでは、2 進数フィールドの選択ビットがオンとオフのパターンどおりであるか、あるいはパターンどおりでないかを厳密にテストできます。前述の方法 1 とは異なり、『等しい』比較と『等しくない』比較だけしか行うことができません。しかし、この方法には、ビットのオンとオフの組み合わせを厳密にテストできるという利点があります。

### 関係条件の形式

関係条件には、次の 2 つの形式を使用できます。



```
▶▶(p1,m1,BI,  $\begin{array}{l} \text{EQ} \\ \text{NE} \end{array}$ ,—constant—)▶▶
```

または、FORMAT=BI オペランドを使用する場合、

```
▶▶(p1,m1,  $\begin{array}{l} \text{EQ} \\ \text{NE} \end{array}$ ,—constant—)▶▶
```

ビット比較演算子は次のとおりです。

**EQ** 等しい  
**NE** 等しくない

## フィールド

**p1,m1:** これらの変数は、ビット定数に比較される入力レコードの 2 進数フィールドを指定します。

- p1 は、ビット定数と比較され、入力レコードの先頭と相対的な、2 進入力フィールドの最初のバイトを指定します。<sup>11</sup> 固定長レコード (FLR) の最初のデータ・バイトは、相対位置が 1 になります。可変長レコード (VLR) の最初のデータ・バイトは、相対位置が 5 (最初の 4 バイトにはレコード記述子が入るため) になります。テストされるすべてのフィールドは、必ずバイト境界から始まり、32752 バイトを超えてはなりません。
- m1 は、テストされるフィールドの長さを指定します。長さは 1 ~ 256 バイトです。

## ビット定数

ビット・ストリング定数は、2 進数フィールドと比較されるパターンを指定します。定数内のビットが 1 または 0 の場合、フィールド内の対応するビットは、それぞれ 1 または 0 と比較されます。定数内のビットが . (ピリオド) の場合、フィールド内の対応するビットは無視されます。

**ビット・ストリングの形式:** ビット・ストリング定数のコーディング形式は、次のとおりです。

```
B'bbbbbbbb...bbbbbbbb'
```

値 bbbbbbbb は、1 バイトを構成する 8 ビットを表します。各ビットは、1 (1 のテスト・ビット)、0 (0 のテスト・ビット)、または . (無視ビット) で指定します。最大 256 までの 8 ビットのグループを指定できます。マスク内のビット数の合計は、8 の倍数で指定します。ビット定数を使用できるのは、ビット比較テスト (BI 形式で EQ または NE 演算子) の場合のみです。

## 埋め込みと切り捨て

ビット定数は、2 進数フィールドのバイト長になるように右端で切り捨てられるか、または埋め込まれます。埋め込み文字は B'00000000' (すべてのビットが 0) で

11. E15 ユーザー出口ルーチンがレコードを形式化する場合、p1 はその出口により再フォーマット設定されたときのレコードを指定するものである必要があります。

## INCLUDE 制御ステートメント

す。埋め込まれたバイトが 2 進数フィールドの余分のバイトと比較されることに注意してください。フィールド長を短くして埋め込み文字を除去するか、またはビット定数を長くして正確なテスト・パターンを指定することにより、好ましくない結果が起こらないようにしてください。

## 出力データ・セットにレコードを組み込む - ビット比較テストの例

### 例 1

```
INCLUDE COND=(27,1,CH,EQ,C'D',AND,18,1,BI,EQ,B'1.....')
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 27 バイトが D の場合  
AND
- 18 バイトが指定パターン (ビット 0 がオン) に等しい場合

### 例 2

```
INCLUDE COND=(11,1,BI,NE,B'10...1.1')
```

この例は、11 バイトが指定のパターン (ビット 0 がオン、ビット 1 がオフ、ビット 5 がオン、かつビット 7 がオン) と等しくないレコードだけを組み込む方法を示したものです。選択されたフィールド値に対する結果は、下記のとおりです。

表 17. ビット比較の例 2: 選択されたフィールド値の結果

11,1,BI 値	11,1,BI 結果	処置
X'85'	偽	レコードを除外
X'C1'	真	レコードを組み込む
X'84'	真	レコードを組み込む
X'97'	偽	レコードを除外

### 例 3

```
INCLUDE COND=(11,2,EQ,B'..01....0.....1',  
OR,21,1,1,EQ,B'01.....'),FORMAT=BI
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 11 ~ 12 バイトが指定パターン (ビット 2 がオフ、ビット 3 がオン、ビット 8 がオフ、かつビット 15 がオン) に等しい場合  
または
- 21 バイトが指定パターン (ビット 0 がオフで、かつビット 1 がオン) に等しい場合

選択されたフィールド値に対する結果は、下記のとおりです。

表 18. ビット比較の例 3: 選択されたフィールド値の結果

11,2,BI 値	11,2,BI 結果	21,1,BI 値	21,1,BI 結果	処置
X'1221'	真	X'C0'	偽	レコードを組み込む
X'02C4'	偽	X'41'	真	レコードを組み込む
X'1234'	偽	X'00'	偽	レコードを除外
X'5F7F'	真	X'7F'	真	レコードを組み込む
X'FFFF'	偽	X'2F'	偽	レコードを除外

## 日付比較

DFSORT の Y2 形式を以下のように、有効な「世紀」ウィンドウと共に使用できます。

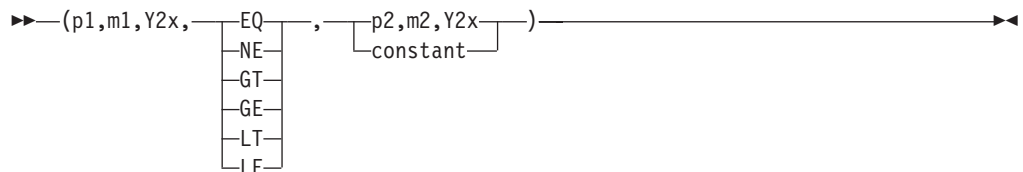
- 完全な日付形式 (Y2T、Y2U、Y2V、Y2W、Y2X、および Y2Y) を使用して、2 桁年の日付フィールドを 2 桁年の日付定数 (Y 定数) または別の 2 桁年の日付フィールドと比較する。
- 年形式 (Y2C、Y2Z、Y2S、Y2P、Y2D、および Y2B) を使用して、2 桁年フィールドを 2 桁年の定数 (Y 定数) または別の 2 桁年フィールドと比較する。

たとえば、Z'yymm' 日付フィールドが、1996 年 1 月と 2005 年 3 月の間にあるレコードだけを組み込むことができます。あるいは、P'dddy' フィールドが、別の P'dddy' フィールドより小さいレコードだけを組み込みことができます。

Y2 フィールドおよび Y 定数で比較のために使用される日付および特殊標識の順序付けは、Y2 フィールドの分類および組み合わせの昇順と同じです (詳細は、345 ページの『SORT 制御ステートメント』を参照してください)。

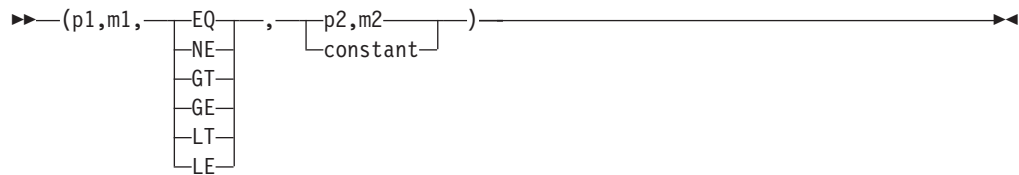
### 関係条件の形式

関係条件には、次の 2 つの形式を使用できます。



または、FORMAT=Y2x オペランドを使用する場合、

## INCLUDE 制御ステートメント



比較演算子には次のものがあります。

<b>EQ</b>	等しい
<b>NE</b>	等しくない
<b>GT</b>	より大きい
<b>GE</b>	以上
<b>LT</b>	より小さい
<b>LE</b>	以下

### フィールド:

**p1,m1,Y2x:** これらの変数は、入力レコード内の別の 2 桁年の日付フィールドまたは 2 桁年の日付定数のどちらかと比較する、入力レコードの 2 桁年の日付フィールドを指定します。

- p1 は、日付フィールドの最初のバイトが、入力レコードの先頭から相対的にどの位置にあるかを指定します。<sup>12</sup> 固定長レコード (FLR) の最初のデータ・バイトは、相対位置が 1 になります。可変長 (VLR) レコードの最初のデータ・バイトは、相対位置が 5 になります (最初の 4 バイトに、レコード記述子ワードが含まれるため)。すべての日付フィールドは、バイト境界から開始します。また、どの日付フィールドも 32752 バイトを超えることはできません。
- m1 は、日付フィールドの長さを指定します。733 ページの『付録 C. データ形式記述』は、日付フィールドのそれぞれのタイプごとに長さ形式を説明しています。
- Y2x は Y2 形式を指定します。733 ページの『付録 C. データ形式記述』は、それぞれの日付フィールドごとに長さ (m) と形式 (Y2x) を説明しています。  
すべての日付フィールドに同じ Y2 形式を使用する場合、Y2x を省略できます。そのときは、FORMAT=Y2x オペランドを使用する必要があります。

**p2,m2,Y2x:** これらの変数には、p1,m1,Y2x フィールドが比較される入力レコード内の別の 2 桁年の日付フィールドを指定します。

**定数:** p1,m1,Y2x フィールドが比較される Y'string' 形式の 2 桁年の日付定数。

**比較:** 日付フィールドは、年以外の桁 (x) が同じ数の日付定数または別の日付フィールドと比較できます。151 ページの表 19 は、使用できるフィールドとフィールドの比較およびフィールドと定数の比較のタイプを示しています。日付のタイプ (たとえば、yyx および xyy) に表示されているフィールドを、その日付タイプに表示されているその他のフィールドや、その日付のタイプに示されている Y 定数と比較できます。

12. E15 ユーザー出力ルーチンによりレコードが形式設定される場合、p1 で、その出口で再フォーマット設定されたレコードを参照する必要があります。

表 19. 日付に使用できる比較

日付のタイプ	フィールド (m,f)		Y 定数
yyx および xyy	3,Y2T 3,Y2W	2,Y2U 2,Y2X	Y'yyx'
yyxx および xxyy	4,Y2T 4,Y2W	3,Y2V 3,Y2Y	Y'yyxx' Y'DATE2'
yyxxx および xxxyy	5,Y2T 5,Y2W	3,Y2U 3,Y2X	Y'yyxxx' Y'DATE3'
yyxxxx および xxxxyy	6,Y2T 6,Y2W	4,Y2V 4,Y2Y	Y'yyxxxx' Y'DATE1'
yy	2,Y2C 2,Y2S 1,Y2D	2,Y2Z 2,Y2P 1,Y2B	Y'yy'

Y'DATE1' は、現在日付の Y 定数を形式 Y'yymmdd' で生成します。 Y'DATE2' は、現在日付の Y 定数を形式 Y'yymm' で生成します。 Y'DATE3' は、現在日付の Y 定数を形式 Y'yyddd' で生成します。

Y 定数では、日付のタイプと同じ桁数を使用します。先行ゼロを指定する必要があります (たとえば、Y'yymm' の場合、2000 年 1 月に Y'0001' を使い、2001 年 1 月に Y'0101' を使用します)。

また、特殊標識に Y 定数を使用することもできます。

- Y'0...0' (CH/ZD/PD ゼロ) および Y'9...9' (CH/ZD/PD の 9) は、Y2T、Y2U、Y2V、Y2W、Y2X、および Y2Y 日付で使用できます。日付のタイプと同じ桁数を使用する必要があります (たとえば、yyq または qyy に Y'000' を使用、yymm または mmyy に Y'0000' を使用する、などです)。
- Y'LOW' (BI ゼロ)、Y'BLANKS' (ブランク)、および Y'HIGH' (BI の 1) を、Y2T、Y2W、および Y2S 日付で使用できます。

## 出力データ・セットへのレコードの組み込み - 日付比較

### 例 1

```
INCLUDE FORMAT=Y2T,
COND=(3,4,GE,Y'9901',AND,
3,4,LE,Y'0312',OR,
3,4,LE,Y'0000')
```

この例は、次の条件を満たすレコードだけを組み込む方法を示したものです。

- 3 ~ 6 バイトの C'yymm' 日付フィールドは、1999 年 1 月から 2003 年 12 月の間です。  
または
- 3 ~ 6 バイトには、CH ゼロ (C'0000')、ZD ゼロ (Z'0000')、または BI ゼロ (X'00000000') が含まれます。

## INCLUDE 制御ステートメント

C'yymm' 日付フィールドの実際の日付と同様、Y'9901' および Y'0312' 日付定数を解釈するためにも、有効な「世紀」ウィンドウが使用されることに注意してください。ただし、C'yymm' 日付フィールドの Y'0000' 特殊標識定数または特殊標識の解釈には、「世紀」ウィンドウは使用されません。

### 例 2

```
INCLUDE COND=(2,3,Y2X,LT,36,5,Y2T)
```

この例は、2 ~ 4 バイトの P'dddy' 日付フィールドが、36 ~ 40 バイトの Z'yyddd' 日付フィールドより小さいレコードだけを組み込む方法を示しています。

P'dddy' および Z'yyddd' 日付フィールドの実際の日付を解釈するために、有効な「世紀」ウィンドウが使用されることに注意してください。ただし、P'dddy' および Z'yyddd' 日付フィールドの特殊標識の解釈には、「世紀」ウィンドウは使用されません。

## INCLUDE/OMIT ステートメントに関する注意事項

- INCLUDE または OMIT ステートメントで、浮動小数点の比較フィールドを参照できません。
- INCLUDE または OMIT ステートメントを用いると、任意の選択を行うことができます。INCLUDE と OMIT は、同時には使用できないステートメントです。
- いくつかの関係条件が AND および OR 論理演算子の組み合わせで結合されている場合は、AND ステートメントが最初に評価されます。COND 式の中で括弧を使用すれば、評価の順序を変更できます。
- E15 または E32 によりレコード形式が変更される場合は、INCLUDE または OMIT ステートメントは最新の形式を適用します。
- INCLUDE または OMIT ステートメントが、テープ作業データ・セット分類または従来の組み合わせ処理に指定されると、DFSORT はメッセージを出して終了します。
- SZERO が有効な場合、数値フィールドと定数が使用される時、-0 は、+0 より小として比較されます。NOSZERO が有効な場合は、数値フィールドと定数が使用される時、-0 は、+0 と等しいとして比較されます。

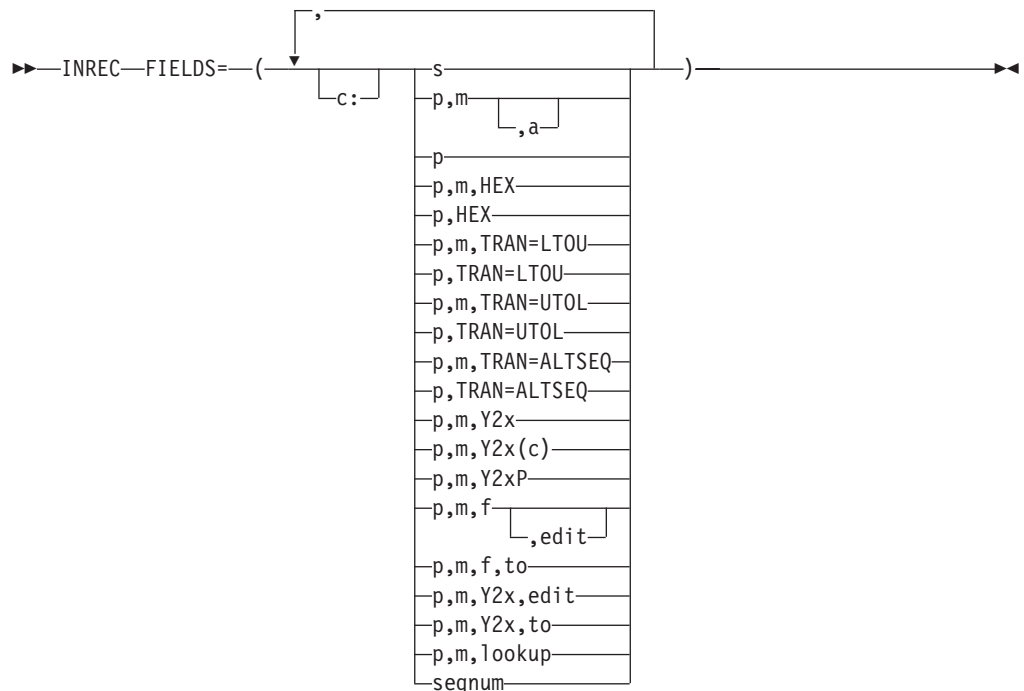
153 ページの表 20 は、ステートメントが INCLUDE または OMIT のどちらであるか、そして関係条件の次に AND と OR 論理演算子のどちらが続くかに応じて、DFSORT が関係条件比較の結果にどのように対処するかを示したものです。

複合ステートメントを書く場合、期待どおりの結果が得られるように表 20 を利用してください。

表 20. INCLUDE/OMIT の論理テーブル

ステートメント	関係条件	次の論理演算子が下記の場合のプログラム処置	
	比較	AND	または
OMIT	真	次の比較を検査、またはレコードを除去 (最後の比較の場合)	レコードを除去
OMIT	偽	レコードを組み込む	次の比較を検査、またはレコードを組み込む (最後の比較の場合)
INCLUDE	真	次の比較を検査、またはレコードを組み込む (最後の比較の場合)	レコードを組み込む
INCLUDE	偽	レコードを除去	比較を検査、またはレコードを除外 (最後の比較の場合)

## INREC 制御ステートメント



INREC 制御ステートメントを使用すると、入力レコードを処理する前にそれらを再フォーマット設定することができます。つまり、入力レコードのどの部分を、再フォーマット設定した入力レコードに組み込むか、それらをどのような順序で組み込むか、さらにどのように境界合わせするかを定義できます。

## INREC 制御ステートメント

再フォーマット設定は、入力レコードから 1 つ以上のフィールドを定義することにより行います。再フォーマット設定された入力レコードは、これらのフィールドだけで構成され、ユーザーが指定した順序で配列され、指定した境界または桁に合わせられています。

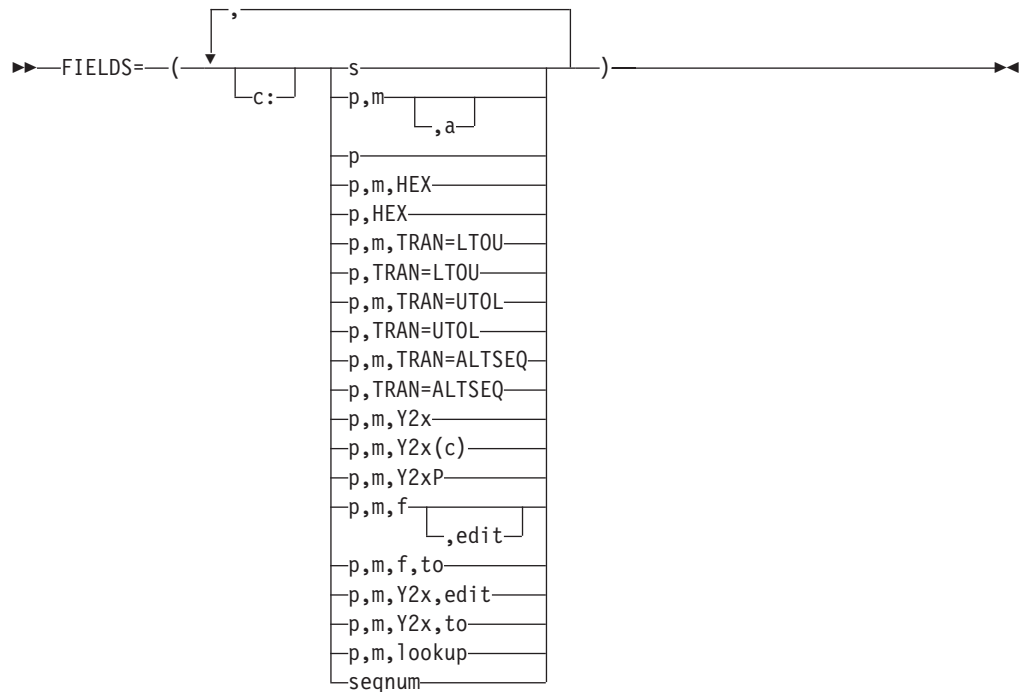
INREC 制御ステートメントでは以下の機能も提供されます。

- 再フォーマット設定されたレコードの入力フィールドの前、入力フィールド間、および入力フィールドの後に、ブランク、ゼロ、ストリング、現在日付、現在時刻およびシーケンス番号を挿入する。
- 16 進数表示、EBCDIC 文字の小文字から大文字へ、または大文字から小文字への変換、ALTSEQ 変換テーブルを使用した文字変換、ある形式から別の形式への数値変換などの、高度な変換機能。
- 長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、数値フィールドの表示方法を制御するなどの、高度な編集機能。  
一般に使用されている数値編集パターンとして 27 個の事前定義編集マスクが用意されており、世界中で使用されている数値表記の大部分を網羅しています。さらに、ユーザー定義の編集マスクにより、実際上無制限の数の数値編集パターンが利用できます。
- SMF 日付値および時間値の、より有効なフォームへの変換。
- 指定された固定またはスライドの「世紀」ウィンドウを使用した、2 桁年の日付の各種形式から 4 桁年の日付の各種形式への変換。
- 入力としての文字、16 進数、またはビット・ストリングに基づいて、参照テーブルから文字または 16 進数ストリングを選択する (つまり、参照と変更)。

INREC および OUTREC の相互関係については、163 ページの『INREC ステートメントに関する注意事項』を参照してください。

## FIELDS





再フォーマット設定された後の入力レコード内での、区切りフィールド (ブランク、ゼロ、ストリング、現在日付および現在時刻)、未編集および編集済みの入力フィールドおよびシーケンス番号の順序と境界合わせを指定します。

**c:** 区切りフィールド、入力フィールドまたはシーケンス番号の位置 (桁) を、再フォーマット設定された入力レコードの先頭から相対的に指定します。指定された列の前にある未使用のスペースには、EBCDIC のブランクが埋め込まれます。次の規則が適用されます。

- c の値は 1 から 32752 の範囲で指定します。
- c: の後には、入力フィールドまたは区切りフィールドが続きます。
- 再フォーマット設定された後の入力レコード内で、c は前の入力フィールドまたは区切りフィールドと重なり合うことはできません。
- 可変長レコードの場合、最初の入力フィールド (レコード記述子ワード) の前にも入力レコードの可変部の後にも、c: を指定できません。
- 次の行への継続する場合、コロン (;) はコンマ (,) またはセミコロン (;) と同様に取り扱われます。

表 21 は、有効な例と無効な例を示しています。

表 21. 桁合わせの有効な例と無効な例

有効性	指定	結果
有効	33:C'State '	1 ~ 32 桁 - ブランク 33 ~ 38 桁 - 'State '
有効	20:5,4,30:10,8	1 ~ 19 桁 - ブランク 20 ~ 23 桁 - 入力フィールド (5,4) 24 ~ 29 桁 - ブランク 30 ~ 37 桁 - 入力フィールド (10,8)

## INREC 制御ステートメント

表 21. 桁合わせの有効な例と無効な例 (続き)

有効性	指定	結果
無効	0:5,4	桁の値にゼロは指定できない。
無効	:25Z	桁の値を指定しなければならない。
無効	32753:21,8	無効 - 桁の値は 32753 より小さくしなければならない。
無効	5:10:2,5	桁の値は隣接して指定できない。
無効	20,10,6:C'AB'	桁の値が前のフィールドに重なっている。

**s** 区切りフィールド (ブランク、ゼロ、文字ストリング、16 進数ストリング、現在日付または現在時刻) を再フォーマット設定された入力レコードに挿入するように指定します。任意の入力フィールドの前と後に指定できます。連続する区切りフィールドを指定できます。可変長レコードの場合、最初の入力フィールド (レコード記述子ワード) の前にも、入力レコードの可変部の後にも、区切りフィールドを指定できません。暗黙的値は、  
**nX**, **nZ**, **nC'xx...x'**, **nX'yy...yy'**, **DATE1**, **DATE1(c)**, **DATE1P**, **DATE2**, **DATE2(c)**, **DATE2P**, **DATE3**, **DATE3(c)**, **DATE3P**, **TIME1**, **TIME1(c)**, **TIME1P**, **TIME2**, **TIME2(c)**, **TIME2P**, **TIME3**, および **TIME3P** です。

**nX** ブランクによる区切り。n バイトの EBCDIC ブランク (X'40') を再フォーマット設定された入力レコードに表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

表 22 は、ブランクによる区切りの有効な例と無効な例を示したものです。

表 22. ブランクによる区切りの有効な例と無効な例

有効性	指定	結果
有効	X または 1X	1 つのブランク
有効	4095X	4095 のブランク
無効	5000X	反復が多すぎる。代わりに 2 つの隣接する区切りフィールドを使用します (たとえば、2500X,2500X)。
無効	0X	0 は指定できない。

**nZ** 2 進ゼロによる区切り。n バイトの 2 進ゼロ (X'00') を再フォーマット設定された入力レコードに表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

表 23 は、ブランクによる区切りの有効な例と無効な例を示したものです。

表 23. 2 進ゼロによる区切りの有効な例および無効な例

有効性	指定	結果
有効	Z または 1Z	1 つの 2 進ゼロ
有効	4095Z	4095 の 2 進ゼロ
無効	4450Z	反復が多すぎる。代わりに 2 つの隣接する区切りフィールドを使用します (たとえば、4000Z,450Z)。

表 23. 2 進ゼロによる区切りの有効な例および無効な例 (続き)

有効性	指定	結果
無効	0Z	0 は指定できない。

**nC'xx...x'**

文字ストリングによる区切り。文字ストリング定数 (C'xx...x') を再フォーマット設定した入力レコードに n 回挿入します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。x には任意の EBCDIC 文字を指定できます。1 ~ 256 文字を指定できます。

文字ストリングに単一のアポストロフィを組み込む場合は、それを 2 つの単一アポストロフィで指定します。

Required: 0'NEILL      Specify: C'0''NEILL'

表 24 は、文字ストリングによる区切りの、有効な例と無効な例を示したものです。

表 24. 文字ストリングによる区切りの、有効な例と無効な例

有効性	指定	結果	長さ
有効	C'John Doe'	John Doe	8
有効	C'JOHN DOE'	JOHN DOE	8
有効	C'\$@#'	\$@#	3
有効	C'+0.193'	+0.193	6
有効	4000C' '	8000 のブランク	8000
有効	20C'**FILLER**'	**FILLER** が 20 回繰り返されます。	200
有効	C'Frank's'	Frank's	7
無効	C''''	アポストロフィが対になっていない	n/a
無効	'ABCDEF'	C 識別子がない	n/a
無効	C'ABCDE	アポストロフィがない	n/a
無効	4450C'1'	反復が多すぎる。代わりに 2 つの隣接する区切りフィールドを使用する (たとえば、4000C'1',450C'1')。	n/a
無効	0C'ABC'	0 は指定できない。	n/a
無効	C''	文字が指定されていない。	n/a
無効	C'Frank's'	1 つのアポストロフィに対して 2 つの単一アポストロフィが必要	n/a

**nX'yy...yy'**

16 進数ストリングの区切り。16 進数ストリング定数 (X'yy...yy') を再フォーマット設定した入力レコードに n 回挿入します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

値 yy は、16 進数の任意の組み合わせを示します。1 組から 256 組までの 16 進数字を指定できます。158 ページの表 25 は、16 進数ストリングによる区切りの、有効な例と無効な例を示したもので

す。

表 25. 16 進数ストリングによる区切りの、有効な例および無効な例

有効性	指定	結果	長さ
有効	X'FF'	FF	1
有効	X'BF3C'	BF3C	2
有効	3X'00000F'	00000F00000F00000F	9
有効	4000X'FFFF'	FF が 8000 回繰り返される。	8000
無効	X'ABGD'	G は 16 進数ではない。	n/a
無効	X'F1F'	数字が対になっていない。	n/a
無効	'BF3C'	X 識別子がない。	n/a
無効	'F2F1'X	X の位置が正しくない。	n/a
無効	8000X'01'	反復が多すぎる。代わりに 2 つの隣接する区切りフィールドを使用する (たとえば、4000X'01',4000X'01')。	n/a
無効	0X'23AB'	0 は指定できない。	n/a
無効	X''	16 進数が指定されていない。	n/a

#### DATEn, DATEN(c), DATEnP

現在日付の定数。実行日付を再形成化された入力レコードに表示します。表 26 に、指定可能な区切りフィールドごとにそれぞれ生成された定数を、ストリング長および例とあわせて、関連箇所 (c) に対し (l) を使用して、示します。yyyy は年、mm は月 (01 ~ 12)、dd は日付 (01 ~ 31)、ddd は年間通日 (001 ~ 366) を表します。また、c はブランク 以外のあらゆる文字として使用できます。

表 26. 現在日付の定数

区切りフィールド	定数	長さ (バイト)	2001 年 4 月 19 日
DATE1	C'yyyymmdd'	8	C'20010419'
DATE1(c)	C'yyyymmdd'	10	C'2001/04/19'
DATE1P	P'yyyymmdd'	5	P'20010419'
DATE2	C'yyyymm'	6	C'200104'
DATE2(c)	C'yyyymm'	7	C'2001/04'
DATE2P	P'yyyymm'	4	P'200104'
DATE3	C'yyyddd'	7	C'2001109'
DATE3(c)	C'yyyddd'	8	C'2001/109'
DATE3P	P'yyyddd'	4	P'2001109'

**TIME<sub>n</sub>, TIME<sub>n</sub>(c), TIME<sub>n</sub>P**

現在時刻の定数。実行時間を再形成化された入力レコードに表示します。  
表 27 に、指定可能な区切りフィールドごとにそれぞれ生成された定数を、  
ストリング長および例とあわせて、関連箇所 (c) に対し (:) を使用して、  
示します。hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~  
59) を表します。また、c はブランク 以外 のあらゆる文字として使用でき  
ます。

表 27. 現在時刻の定数

区切り フィールド	定数	長さ (バイト)	01:55:43 PM
TIME1	C'hmmss'	6	C'135543'
TIME1(c)	C'hcmmcss'	8	C'13:55:43'
TIME1P	P'hmmss'	4	P'135543'
TIME2	C'hmm'	4	C'1355'
TIME2(c)	C'hcmmm'	5	C'13:55'
TIME2P	P'hmm'	3	P'1355'
TIME3	C'hh'	2	C'13'
TIME3P	P'hh'	2	P'13'

**p,m,a**

未編集の入力フィールドを、再フォーマット設定された入力レコードに出力する  
ことを指定します。

**p** 入力フィールドの最初のバイトが入力レコードの先頭から相対的にどの  
位置にあるかを指定します。<sup>13</sup> 固定長レコードの最初のデータ・バイト  
は、相対位置が 1 になります。可変長レコードの最初のデータ・バイ  
トは、相対位置が 5 になります (最初の 4 バイトに RDW が含まれる  
ため)。すべてのフィールドはバイト境界から開始します。また、フィー  
ルドは 32752 バイトを超えることはできません。可変長レコードに関  
する特殊な規則については、163 ページの『INREC ステートメントに  
関する注意事項』を参照してください。

**m** 入力フィールドの長さを指定します。データに符号がある場合は、その  
符号を含んでいなければならない、かつ、必ず整数のバイトで指定しま  
す。詳細については、163 ページの『INREC ステートメントに関する  
注意事項』を参照してください。

**a** 再フォーマット設定された後の入力レコード内の入力フィールドの境界  
合わせ (変位) を、再フォーマット設定された入力レコードの先頭から  
相対的に指定します。

次の値を **a** に指定できます。

**H** ハーフワード位置合わせ。再フォーマット設定された入力レコ  
ードの先頭からのフィールドの変位 (p-1) が、バイト単位で 2  
の倍数 (すなわち、1、3、5 桁目など) であることを示します。

13. E15 ユーザー出口でレコードを再フォーマット設定する場合、p はその出口により再フォーマット設定されたレコードを指定する必  
要があります。

## INREC 制御ステートメント

- F** フルワード位置合わせ。変位は、4 の倍数 (すなわち、1、5、9 桁目など) です。
- D** ダブル・ワード位置合わせ。変位は、8 の倍数 (すなわち、1、9、17 など) です。

たとえば、COMPUTATIONAL 項目が SYNCHRONIZED 文節により境界合わせされている COBOL アプリケーション・プログラムでデータが使用される場合は、境界合わせが必要となります。境界合わせされたフィールドの前にある未使用のスペースは、常に 2 進ゼロで埋め込まれます。

- p** 入力レコードの未編集の部分 (最小レコード長より大きい部分) が、再フォーマット設定された入力レコードに最後のフィールドとして表示されるように指定します。再フォーマット設定された入力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

p に指定される値は、最小レコード長 (RECORD ステートメント L4 値) に 1 バイトを加算した値以下である必要があります。

### **p,m,HEX**

入力フィールドの 16 進表示が再フォーマット設定された入力レコードに表示されるように指定します。

詳細については、OUTFIL OUTREC の p,m,HEX の説明を参照してください。

### **p,HEX**

入力レコードの可変部の 16 進表示 (最小レコード長を超える部分) が、再フォーマット設定された入力レコードに最後のフィールドとして表示されるように指定します。再フォーマット設定された入力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,HEX の説明を参照してください。

### **p,m,TRAN=LTOU**

入力フィールドの小文字の EBCDIC 文字 (すなわち、a ~ z) が、大文字の EBCDIC 文字 (すなわち、A ~ Z) として再フォーマット設定された入力レコードに表示されるように指定します。

詳細については、OUTFIL OUTREC の p,m,TRAN=LTOU の説明を参照してください。

### **p,TRAN=LTOU**

入力レコードの可変部 (最小レコード長を超える部分) の小文字の EBCDIC 文字 (すなわち、a ~ z) が、大文字の EBCDIC 文字 (すなわち、A ~ Z) として再フォーマット設定された入力レコードに最終フィールドとして表示されるように指定します。再フォーマット設定された入力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,TRAN=LTOU の説明を参照してください。

**p,m,TRAN=UTOL**

入力フィールドの大文字の EBCDIC 文字 (すなわち、A ~ Z) が、小文字の EBCDIC 文字 (すなわち、a ~ z) として、再フォーマット設定された入力レコードに表示されるように指定します。

詳細については、OUTFIL OUTREC の p,m,TRAN=UTOL の説明を参照してください。

**p,TRAN=UTOL**

入力レコードの可変部 (最小レコード長を超える部分) の大文字の EBCDIC 文字 (すなわち、A ~ Z) が、小文字の EBCDIC 文字 (すなわち、a ~ z) として、再フォーマット設定された入力レコードに最終フィールドとして表示されるように指定します。再フォーマット設定された入力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,TRAN=UTOL の説明を参照してください。

**p,m,TRAN=ALTSEQ**

入力フィールドの文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された入力レコードで変更されるように指定します。

詳細については、OUTFIL OUTREC の p,m,TRAN=ALTSEQ の説明を参照してください。

**p,TRAN=ALTSEQ**

入力レコードの可変部 (最小レコード長を超える部分) の文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された入力レコードで、最終フィールドとして変更されるように指定します。再フォーマット設定された入力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,TRAN=ALTSEQ の説明を参照してください。

## サンプル構文:

Fixed input records

```
INREC FIELDS=(1:5,10,15:8C'0',25:20,15,TRAN=LTOU,80:X)
```

Variable input records

```
INREC FIELDS=(1,4,C' RDW=',1,4,HEX,C' FIXED=',
5,20,HEX,C' VARIABLE=',21,HEX)
```

**p,m,Y2x**

2 桁年の入力日付フィールドの 4 桁年の CH 日付表記を、再フォーマット設定された入力レコードに出力するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2T は、C'000000' から C'00000000' に変換されます)。

詳細については、OUTFIL OUTREC の p,m,Y2x の説明を参照してください。

## サンプル構文:

```
INREC FIELDS=(21,3,Y2V,X,12,5,Y2W)
```

## INREC 制御ステートメント

### p,m,Y2x(c)

2 桁年の入力日付フィールドの区切り文字付きの 4 桁の CH 日付表記を、再フォーマット設定された入力レコードに出力するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2T(/) は、C'000000' から C'0000/00/00' に変換されます)。

詳細については、OUTFIL OUTREC の p,m,Y2x(c) の説明を参照してください。

サンプル構文:

```
INREC FIELDS=(25,6,Y2T(-),X,14,2,Y2U(/))
```

### p,m,Y2xP

2 桁年の入力日付フィールドの 4 桁年の PD 日付表記を、再フォーマット設定された入力レコードに出力するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2TP は、C'000000' から P'00000000' に変換されます)。

詳細については、OUTFIL OUTREC の p,m,Y2xP の説明を参照してください。

サンプル構文:

```
INREC FIELDS=(11,3,Y2XP,X,21,4,Y2WP)
```

### p,m,f,edit

編集済みの数値入力フィールドを再フォーマット設定された入力レコードに表示するように指定します。事前定義の編集マスク (M0 ~ M26) またはユーザーが定義する特定の編集パターンのいずれかを使用して、BI、FI、PD、PD0、ZD、CSF/FS、DT1、DT2、DT3、TM1、TM2、TM3 または TM4 フィールドを編集できます。長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、編集済みフィールドの表示方法を制御できます。

詳細については、OUTFIL OUTREC の項の p,m,f,edit の説明を参照してください。

サンプル構文:

```
INREC FIELDS=(5:21,8,ZD,M19,X,46,5,ZD,M13,  
31:35,6,FS,SIGNS=(.,+,-),LENGTH=10,  
51:8,4,PD,EDIT=(**II,IIT.TTXS),SIGNS=(.,+,-))
```

### p,m,f,to

変換済みの数値入力フィールドを再フォーマット設定された入力レコードに出力するように指定します。BI、FI、PD、PD0、ZD、CSF/FS、DT1、DT2、DT3、TM1、TM2、TM3 および TM4 フィールドを、BI、FI、PD、ZD、または CSF/FS フィールドに変換できます。

詳細については、OUTFIL OUTREC の p,m,f の説明を参照してください。

サンプル構文:

```
INREC FIELDS=(21,5,ZD,T0=PD,X,8,4,ZD,FI,LENGTH=2)
```



**p,m,Y2x,edit**

2桁年の入力日付フィールドの編集済み 4桁年の CH 日付表記を、再フォーマット設定された入力レコードに出力するように指定します。

詳細については、OUTFIL OUTREC の p,m,Y2x,edit の説明を参照してください。

**p,m,Y2x,to**

2桁年の入力日付フィールドの変換済み 4桁年の日付表記を、再フォーマット設定された入力レコードに出力するように指定します。

詳細については、OUTFIL OUTREC の p,m,Y2x の説明を参照してください。

**p,m,lookup**

参照テーブルの文字または 16進数ストリングが、再フォーマット設定された入力レコードに出力されるように指定します。p,m,lookup を使用して、文字、16進数、またはビット定数に対する入力値の突き合わせに基づく、指定の文字または 16進数ストリングを選択できます。

詳細については、OUTFIL OUTREC の p,m,lookup の説明を参照してください。

サンプル構文:

```
INREC FIELDS=(11,1,
              CHANGE=(6,
                      C'R',C'READ',
                      C'U',C'UPDATE',
                      X'FF',C'EMPTY',
                      C'A',C'ALTER'),
              NOMATCH=(11,6),
              4X,
              21,1,
              CHANGE=(10,
                      B'.1.....',C'VSAM',
                      B'.0.....',C'NON-VSAM'))
```

**seqnum**

シーケンス番号を再フォーマット設定された入力レコードに出力するように指定します。シーケンス番号は、INREC 処理でレコードが受け取られる順序で割り当てられます。BI、PD、ZD、または CSF/FS シーケンス番号を作成し、それらの長さ、開始値および増分値を制御できます。

詳細については、OUTFIL OUTREC の seqnum の説明を参照してください。

サンプル構文:

```
INREC FIELDS=(SEQNUM,6,ZD,START=1000,INCR=50,1,60)
```

デフォルト: なし。必須のパラメーターです。697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**INREC ステートメントに関する注意事項**

- INREC が指定されていると、DFSORT はユーザー出口 E15 または INCLUDE/OMIT ステートメント処理が完了した後で、入力レコードを再フォーマット設定します。したがって、E15 ユーザー出口と INCLUDE/OMIT ステート

## INREC 制御ステートメント

メントによるフィールドの参照には影響しませんが、SORT、OUTREC、および SUM ステートメントでは、再フォーマット設定した入力レコードのフィールドを参照します。E35 ユーザー出口では、再フォーマット設定された出力レコードのフィールドを参照します。

- 通常、SORT および SUM ステートメントが元の入力レコードのフィールドを参照できるように、INREC ではなく OUTREC を使用する必要があります。
- SORT、MERGE、INCLUDE、または OMIT フィールドでロケール処理を使用する場合、INREC は使用できません。INREC ではなく、OUTREC ステートメントまたは OUTFIL ステートメントの OUTREC オペランドを使用します。
- INREC を指定する場合は、再フォーマット設定された後の入力レコードのサイズおよびレイアウトの変化を把握しておく必要があります。
- INREC を用いてレコードの長さを大幅に短くすると、パフォーマンスが向上します。INREC および OUTREC は、レコードを再フォーマット設定するために実際に必要とされる場合以外は使用しないでください。
- 可変長レコードの場合、FIELDS パラメーターの最初の入力項目には、4 バイトのレコード記述ワード (RDW) を指定するか、もしくは組み込みます。DFSORT は、再フォーマット設定されたレコードの長さを RDW で設定します。

入力レコードのデータ部分にある最初のフィールドが、RDW の直前にある再フォーマット設定された入力レコードに挿入される場合、FIELDS パラメーター内の入力は、RDW とデータ・フィールドの両方を一度に指定できます。それ以外の場合は、RDW を再フォーマット設定した入力レコードに組み込む必要があります。

- SORTOUT LRECL が指定されているか使用可能な場合、再フォーマット設定された INREC レコード長に SORTOUT LRECL の値が一致しなくとも、DFSORT はこの値を使用します。これにより、再フォーマット設定された INREC レコードの埋め込みまたは切り捨て、または終了が発生します。SORTOUT LRECL が指定されていないか使用可能でない場合、該当すれば、DFSORT が自動的に、SORTOUT LRECL として再フォーマット設定された INREC レコード長を使用できます。181 ページの『OPTION 制御ステートメント』の SOLRF および NOSOLRF オプションの説明を参照してください。

VSAM データ・セットの場合、クラスター内で定義される最大レコード・サイズは、固定長レコードを処理する場合の LRECL に等しく、また可変長レコードを処理する場合の LRECL より 4 バイト小さくなります。詳細については、16 ページの『VSAM に関する考慮事項』を参照してください。

- 入力レコードの可変部 (最小レコード長を超える部分) は、再フォーマット設定された入力レコードに組み込むことができます。これを組み込む場合は、それを最後の部分にします。この場合、*pn* には最小レコード長 (RECORD 制御ステートメントの L4 を参照) に 1 バイトを加えた値以下の値を指定し、*mn* および *an* は省略します。

INREC と OUTREC の両方を指定すると、その最後の部分については、両方とも位置だけを指定するか、あるいは両方とも位置だけを指定しないかのどちらかである必要があります。

再フォーマット設定された入力に RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになります。

- INREC で指定されると、入力レコードは処理の前に再フォーマット設定されません。出力レコードは、OUTREC が INREC とともに指定されていない限り、INREC で指定された形式になります。
- INREC ステートメントで参照される各フィールドは、そのフィールド同士で重なり合うことも、制御フィールドと重なり合うこともできます。
- 入力が可変レコードであれば、出力も可変になります。つまり、DFSORT が出力前に各レコードに正しい RDW を与えます。
- 合計中にオーバーフローが起こる可能性がある場合、INREC を使用すれば、再フォーマット設定された入力レコード内にさらに大きな SUM フィールドを作成できます (おそらく、分類または組み合わせではレコードはより大きくなります)。したがって、オーバーフローは起こらなくなります。
- テープ作業データ・セットの分類、または従来の組み合わせ処理の場合に、INREC ステートメントが指定されると、DFSORT はメッセージを出して終了します。
- SZERO が有効な場合、編集済みまたは変換済みの入力フィールドについて、-0 は負として扱われ、+0 は正として扱われます。NOSZERO が有効な場合は、編集済みまたは変換済みの入力フィールドについて、-0 および +0 は正として扱われます。

## 処理前のレコードの再フォーマット設定 - 例

### 例 1

#### INREC 方式:

```
INCLUDE COND=(5,1,GE,C'M'),FORMAT=CH
INREC FIELDS=(10,3,20,8,33,11,5,1)
SORT FIELDS=(4,8,CH,A,1,3,FI,A)
SUM FIELDS=(17,4,BI)
```

#### OUTREC 方式:

```
INCLUDE COND=(5,1,GE,C'M'),FORMAT=CH
OUTREC FIELDS=(10,3,20,8,33,11,5,1)
SORT FIELDS=(20,8,CH,A,10,3,FI,A)
SUM FIELDS=(38,4,BI)
```

上記の例は、固定長入力データ・セットを分類し、出力用に再フォーマット設定する方法を示しています。INREC または OUTREC を使用して、不要なフィールドは出力レコードから削除されます。SORTIN LRECL は 80 です。

また、INCLUDE ステートメントを使用してレコードの組み込みまたは排除を行い、SUM ステートメントを使用してレコードを合計します。

再フォーマット設定された入力レコードは、レコード・サイズが 23 バイトの固定長になります。SOLRF (IBM 提供のデフォルト) が有効なので、SORTOUT LRECL が指定されているか使用可能である場合を除き、23 バイトの再フォーマッ

## INREC 制御ステートメント

ト設定されたレコード長に自動的に設定されます。再フォーマット設定されたレコードは、INREC または OUTREC 処理の後、次のようになります。

位置	内容
<b>1-3</b>	入力位置 10 ~ 12
<b>4-11</b>	入力位置 20 ~ 27
<b>12-22</b>	入力位置 33 ~ 43
<b>23</b>	入力位置 5

INREC と OUTREC のどちらを使用しても、同じ結果が得られます。ただし、OUTREC を使用したほうが、SORT ステートメントと SUM ステートメントのコーディングが簡単になります。どちらの場合でも、INCLUDE COND パラメーターは元の入力レコードのフィールドを参照します。ただし、INREC を使用した場合、SUM および SORT FIELDS パラメーターは、再フォーマット設定された入力レコードのフィールドを指定する必要がありますが、OUTREC の場合は、これらのパラメーターは、元の入力レコードのフィールドを指定します。

### 例 2

```
INREC FIELDS=(1,35,2Z,36,45)
MERGE FIELDS=(20,4,CH,D,10,3,CH,D),FILES=3
SUM FIELDS=(36,4,BI,40,8,PD)
RECORD TYPE=F,LENGTH=(80,,82)
```

この例は、3 つの固定長データ・セットを組み合わせて出力用に再フォーマット設定する場合に、どのようにすれば合計フィールドのオーバーフローを防ぐことができるかを示したものです。入力レコード・サイズは 80 バイトです。RECORD ステートメントの使用例を示すために、SORTIN および SORTOUT は存在しない (すなわち、すべての入出力はユーザー出口により処理される) ものと想定します。

再フォーマット設定された入力レコードは、レコード・サイズが 82 バイト (元の 80 バイトからわずかに増加) の固定長レコードです。このレコードは、次のようになります。

位置	内容
<b>1-35</b>	入力位置 1 ~ 35
<b>36-37</b>	2 進数ゼロ (オーバーフローを防ぐため)
<b>38-82</b>	入力位置 36 ~ 80

MERGE および SUM ステートメントは、再フォーマット設定された後の入力レコードのフィールドを参照します。

再フォーマット設定された後の出力レコードは、再フォーマット設定された入力レコードと同じになります。

つまり、元の入力レコードの 36 と 37 桁目にある 2 バイトの合計フィールドは、組み合わせ処理の 前に、再フォーマット設定された入出力レコードの 36 ~ 39 桁目にある、4 バイトの合計フィールドに拡張されます。これにより、この合計フィールドのオーバーフローを防ぐことができます。INREC の代わりに OUTREC を

使用すると、レコードは組み合わせ処理の後に再フォーマット設定されるため、2 バイトの合計フィールドはオーバーフローを起こす可能性があることに注意してください。

**注:** オーバーフローを防ぐこの方法は、1 ではなくゼロによる埋め込みが符号を変えてしまうため、負の FI 合計フィールドには使用できません。

### 例 3

```
INREC FIELDS=(20,4,12,3)
SORT FIELDS=(1,4,D,5,3,D),FORMAT=CH
OUTREC FIELDS=(5X,1,4,H,19:1,2,5,3,DATE1(-),80X'FF')
```

この例は、固定長入力データ・セットを分類し、出力用に再フォーマット設定する方法を示したものです。分類前に INREC を使用して、できるだけ入力レコードを減らし、分類後に OUTREC を使用して、埋め込み、現在日付 および繰り返されたフィールドを挿入することにより、さらに効率的な分類が行われます。SORTIN LRECL は 80 バイトです。

再フォーマット設定された入力レコードは、レコード・サイズが 7 バイト (元の 80 バイトから大幅に短縮) の固定長レコードです。このレコードは、次のようになります。

位置	内容
1-4	入力位置 20 ~ 23
5-7	入力位置 12 ~ 14

SORT および OUTREC ステートメントには、再フォーマット設定された入力レコードのフィールドを指定します。

再フォーマット設定された出力レコードは固定長で、レコード・サイズは 113 バイトです。SOLRF (IBM 提供のデフォルト) が有効なので、SORTOUT LRECL が指定されているか使用可能である場合を除き、113 の再フォーマット設定されたレコード長に自動的に設定されます。再フォーマット設定された出力レコードは、次のようになります。

位置	内容
1-5	EBCDIC ブランク
6	2 進数ゼロ (H 境界合わせのための)
7-10	入力位置 20 ~ 23
11-18	EBCDIC ブランク
19-20	入力位置 20 ~ 21
21-23	入力位置 12 ~ 14
24-33	形式 C'yyyy-mm-dd' で示された現在日付
34-113	16 進数 FF

したがって、INREC および OUTREC を使用することにより、出力レコードが 113 バイトであっても、80 バイト・レコードではなく 7 バイト・レコードの分類を行うことができます。

## INREC 制御ステートメント

### 例 4

```
INREC FIELDS=(8100,10,1,8099,8110,891)
SUM FIELDS=(1,10,ZD)
SORT FIELDS=(5011,6,CH,A)
OUTREC FIELDS=(11,8099,1,10,8110,891)
```

この例では、INREC および OUTREC を使用して、DFSORT の通常の限界 (4092 バイト) を超えるフィールドで合計を行なう方法を示しています。

「分類」フィールドは、入力位置 5001 ~ 5006 です。「合計」フィールドは、入力位置 8100 ~ 8109 です。INREC ステートメントを使用して入力レコードが再フォーマット設定されるため、「合計」フィールドは最初の 4092 バイト内になります。再フォーマット設定された入力レコードは、次のようになります。

位置	内容
1-10	入力位置 8100 ~ 8109。ここで合計フィールドは、8100 桁目ではなく、1 桁目から開始します。
11-8109	入力位置 1 から 8099、すなわち、合計フィールドの前のレコードの一部です。ここで分類フィールドは、5001 桁目ではなく、5011 桁目から開始します。
8110-9000	入力位置 8110 から 9000、すなわち、合計フィールドの後のレコードの一部です。

INREC ステートメントにより、SUM ステートメントは、1 桁目の再フォーマット設定されたレコードの合計フィールドを参照できます。SORT ステートメントは、5011 桁目で再フォーマット設定されたレコードの分類フィールドを参照する必要があります。レコードを元の形式に復元するには、OUTREC ステートメントが使用されます。ただし、1 から 10 桁目までの、更新済みの合計フィールドが必要です。

### 例 5

```
OPTION COPY,Y2PAST=1985
INREC FIELDS=(SEQNUM,4,BI,
              8,5,ZD,TO=PD,
              31,2,PD,TO=FI,LENGTH=2,
              15,6,Y2TP,
              25,3,CHANGE=(1,C'L92',X'01',C'M72',X'02',C'J42',X'03'),
              NOMATCH=(X'FF'))
```

この例は、シーケンス番号を生成する方法、ある数値形式または日付形式の値を別の形式に変換する方法、および参照テーブルの使用方法を表しています。

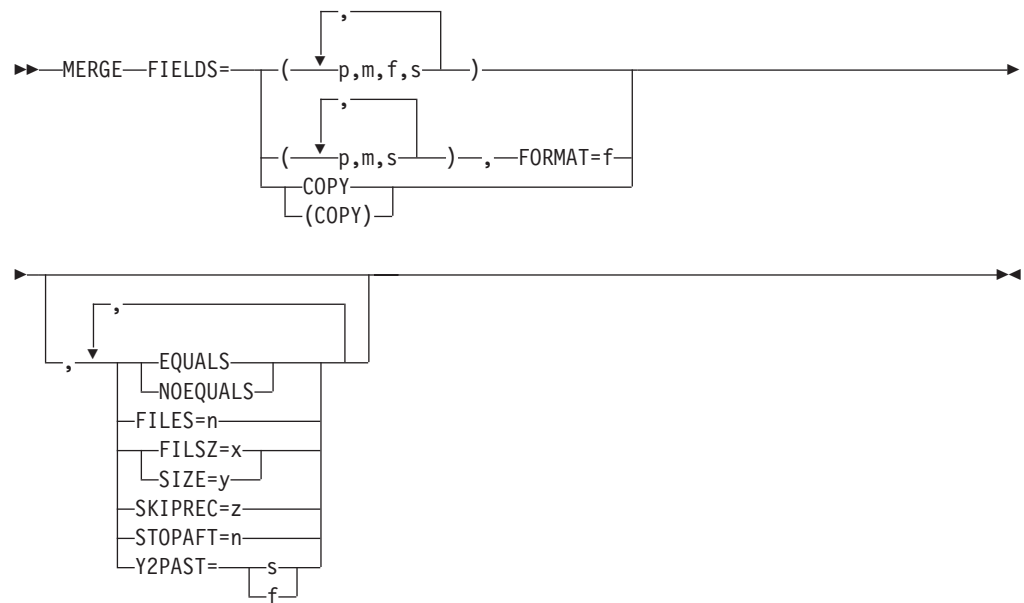
再フォーマット設定された入力レコードは、次のようになります。

位置	内容
1-4	1 から始まり、1 増分される 2 進数のシーケンス番号。
5-7	入力位置 8 ~ 12 の変換済み ZD フィールドを含む PD フィールド。
8-9	入力位置 31 ~ 32 の変換済み PD フィールドを含む FI フィールド。

- 10-14** 1985 ~ 2084 の指定された「世紀」ウィンドウに従って変換された、入力位置 15 ~ 20 からの C'yymmdd' 日付フィールドを含む P'yyyymmdd' 日付フィールド。
- 15** 25 ~ 27 の入力フィールドに参照テーブルを使用して判別した、X'01'、X'02'、X'03'、または X'FF' を含む BI フィールド。

この結果、SORT ステートメントは再フォーマット設定された入力レコードの「分類」フィールドを参照できます。レコードを元の形式に復元するには、OUTREC ステートメントが使用されます。

## MERGE 制御ステートメント



組み合わせ操作を実行する場合は、MERGE 制御ステートメントを使用します。このステートメントは、入力データ・セットがあらかじめ分類されている入力レコード内の制御フィールドを記述します。

また MERGE ステートメントを使用して、コピー・アプリケーションを指定することもできます。ユーザー・ラベルは出力データ・セットにコピーされません。

ブロック・セットによる組み合わせなら最大 100 まで、従来の組み合わせなら最大 16 までのデータ・セットを組み合わせることができます。ブロック・セット組み合わせ手法を選択しない場合、SORTDIAG DD ステートメントを使用すると、ブロック・セットが使用できない理由を示すコードを示すメッセージ ICE800I を発行させることができます。

DFSORT が短い MERGE 制御フィールドを処理する方法は、VLSHRT/NOVLSHRT の設定により異なります。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張され

## MERGE 制御ステートメント

るようなフィールドを指します。短いレコードの組み合わせの詳細については、181 ページの『OPTION 制御ステートメント』の VLSHRT および NOVLSHRT オプションの説明を参照してください。

MERGE ステートメントで使用可能なオプションは、他のソースでも同様に指定できます。これらのオプションのすべての可能なソースおよび指定変更の順序を示す表が、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』に記載されています。MERGE または OPTION ステートメントのどちららでも指定できるオプションは、OPTION ステートメントで指定する方がよいでしょう。

DFSORT は、WORK=value および ORDER=value の MERGE オペランドは受け入れませんが、処理は行いません。

DFSORT の照合の動作を、使用する文化環境に合わせて修正できます。文化環境は、活動ロケールを選択することにより設定されます。活動ロケールの照合規則は、MERGE 処理に次のような影響を与えます。

- DFSORT は、活動ロケールに定義されている照合規則に従って、出力用の組み合わせレコードを作成します。これは、単一の文字データまたは複数バイトの文字データの組み合わせを行います。この組み合わせは、言語の文化特性およびローカル特性を保存する定義済みの照合規則に基づいています。

ロケール処理を使用する場合は、活動ロケールは文字 (CH) 制御フィールドの処理のみに使用されます。

ロケールの処理の詳細については、7 ページの『文化的環境についての考慮事項』または 181 ページの『OPTION 制御ステートメント』の LOCALE の項を参照してください。

**注:** 組み合わせアプリケーションの場合、E35 出口ルーチンの間に削除されたレコードは、シーケンス検査されません。出力データ・セットなしで E35 ユーザー出口ルーチンを使用する場合は、レコードが E35 ユーザー出口に渡される時点ではシーケンス検査は行われません。したがって、入力レコードが正しい順序になっていることを確認する必要があります。

### FIELDS

→ FIELDS=(p,m,f,s) →

組み合わせの場合も分類の場合も全く同じ書き方をします。p、m、f、および s の意味は、SORT ステートメントの項で説明しています。このデフォルトおよび次のパラメーターについても、その項で説明しています。345 ページの『SORT 制御ステートメント』を参照してください。

### FIELDS=COPY または FIELDS=(COPY)

→ FIELDS= COPY →

→ FIELDS=(COPY) →



## MERGE 制御ステートメント

181 ページの『OPTION 制御ステートメント』の OPTION ステートメントの COPY オプションを参照してください。

### FORMAT=f

▶▶—FORMAT=f—▶▶

345 ページの『SORT 制御ステートメント』の FORMAT オペランドの説明を参照してください。組み合わせの場合も分類の場合と同じように使用されます。

### EQUALS または NOEQUALS

▶▶—EQUALS—  
      └─NOEQUALS—▶▶

181 ページの『OPTION 制御ステートメント』の OPTION ステートメントのこれらのオプションを参照してください。

### FILES=n

▶▶—FILES=n—▶▶

入力が E32 出口ルーチンにより与えられる場合に、組み合わせ用の入力ファイルの数を指定します。

デフォルト: なし。E32 出口ルーチンを使用する場合は、指定します。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### FILSZ または SIZE

▶▶—FILSZ=x—  
      └─SIZE=y—▶▶

181 ページの『OPTION 制御ステートメント』の OPTION ステートメントのこれらのオプションを参照してください。

### SKIPREC

▶▶—SKIPREC=z—▶▶

181 ページの『OPTION 制御ステートメント』の OPTION ステートメントのこれらのオプションを参照してください。

### STOPAFT

▶▶—STOPAFT=n—▶▶

181 ページの『OPTION 制御ステートメント』の OPTION ステートメントのこれらのオプションを参照してください。

## MERGE 制御ステートメント

### Y2PAST



181 ページの『OPTION 制御ステートメント』の OPTION ステートメントのこれらのオプションを参照してください。

注: Y2PAST=value の代わりに CENTURY=value および CENTWIN=value を使用できます。

## MERGE または COPY の指定 — 例

### 例 1

```
MERGE FIELDS=(2,5,CH,A),FILSZ=29483
```

#### FIELDS

制御フィールドは、入力データ・セットの各レコードの 2 バイト目から始まります。このフィールドの長さは 5 バイトで、昇順で事前に分類された文字 (EBCDIC) データを含みます。

#### FILSZ

入力データ・セットには、正確に 29483 件のレコードが含まれています。

### 例 2

```
MERGE FIELDS=(3,8,ZD,E,40,6,CH,D)
```

#### FIELDS

大制御フィールドは各レコードの 3 バイト目から始まり、8 バイトの長さで、ゾーン 10 進数データを含みます。このデータは組み合わせ処理で検査する前に、ユーザー・ルーチンにより変更されます。

2 番目の制御フィールドは 40 バイト目から始まり、長さは 6 バイトで、文字データを含み、降順に並んでいます。

### 例 3

```
MERGE FIELDS=(25,4,A,48,8,A),FORMAT=ZD
```

#### FIELDS

大制御フィールドは各レコードの 25 バイト目から始まり、4 バイトの長さで、ゾーン 10 進数データを含み、昇順に並んでいます。

2 番目の制御フィールドは 48 バイト目から始まり、8 バイトの長さです。これもゾーン 10 進形式で、昇順に並んでいます。どちらの制御フィールドも同じデータ形式であるため、FORMAT パラメーターを使用できます。

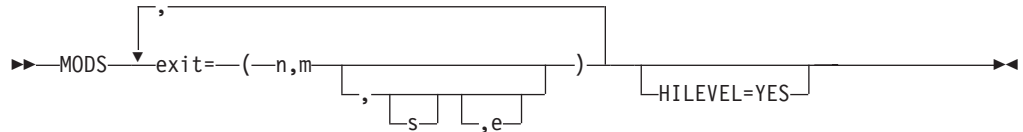
## 例 4

```
MERGE FIELDS=COPY
```

## FIELDS

入力データ・セットが出力にコピーされます。組み合わせは行われません。

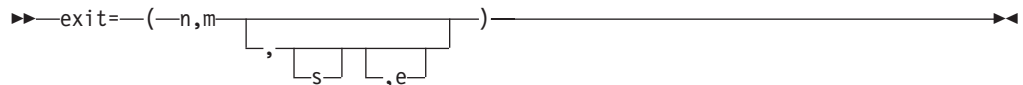
## MODS 制御ステートメント



DFSORT がユーザー出口でユーザー・ルーチンに制御を渡す場合のみ、MODS ステートメントが必要になります。MODS ステートメントは、ユーザー・ルーチンと特定の DFSORT 出口を関連付け、これらのルーチンの記述を DFSORT に提供します。DFSORT ユーザー出口とユーザー・ルーチンの使用の詳細については、361 ページの『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』を参照してください。

ユーザー出口の 1 つを使用するには、上記の MODS ステートメントの *exit* を 3 文字の名前 (たとえば E31) に置き換えます。E32 以外の任意の有効なユーザー出口を指定できます。(E32 は、プログラムから呼び出される組み合わせ操作でだけ使用されます。したがって、そのアドレスはパラメーター・リストで渡される必要があります。)

## exit



*exit* パラメーターに続く値は、ユーザー・ルーチンを記述します。これらの値は次の通りです。

- n** ユーザー・ルーチンの名前 (ユーザー・ルーチンがライブラリーにある場合はメンバー名) を指定します。ユーザー・ルーチンには、任意の有効なオペレーティング・システム名を使用できます。これにより、異なる名前をもつ複数の代替ルーチンを、同一ライブラリーに入れておくことができます。
- m** ユーザー・ルーチンが使用する主記憶域のバイト数を指定します。これには、ユーザー・ルーチンから (あるいは、たとえば OPEN により)、GETMAIN により獲得される記憶域、および COBOL ライブラリー・サブルーチンをロードするために必要な記憶域を含めます。
- s** ユーザー・ルーチンが入っているライブラリーを定義する DFSORT ジョブ・ス

## MODS 制御ステートメント

トップ内の DD ステートメント名を指定するか、またはユーザー・ルーチンが入力ストリームにある場合は SYSIN を指定します。SYSIN は、コピー処理に対しては無効です。

s の値が指定されていない場合は、DFSORT は次の順序で探索を行ない、ユーザー・ルーチンが入っているライブラリーを見つけます。

1. STEPLIB DD ステートメントで指定されたライブラリー
2. JOBLIB DD ステートメントで指定されたライブラリー (STEPLIB DD ステートメントがない場合)
3. リンク・ライブラリー

**e** ユーザー・ルーチンのリンケージ・エディター要件を指定するか、またはユーザー・ルーチンが COBOL で作成されていることを示します。次の値をとることができます。

**N** ユーザー・ルーチンがすでにリンク・エディットされており、これ以上リンク・エディットせずに DFSORT 処理で使用できることを指定します。これが e のデフォルトです。N (指定されたか、またはデフォルトとして使用された) は、EXEC PARM パラメーターの 'E15=COB' および 'E35=COB'、または HILEVEL=YES パラメーターで指定変更できます。

**C** E15 または E35 ルーチンが COBOL で作成されていることを指定します。他の出口に C を指定すると、その指定は無視されて N と見なされます。COBOL で作成されたルーチンは、あらかじめリンク・エディットされている必要があります。OPTION ステートメントの COBEXIT オプションが出口ルーチンのライブラリーを指定します。

**T** ユーザー・ルーチンが、DFSORT の同じフェーズで使用される他のルーチン (たとえば E1n ルーチン) とともにリンク・エディットされていなければならないことを指定します。詳しくは、374 ページの『ユーザー出口ルーチンのダイナミック・リンク編集の方法』を参照してください。この値は、コピー処理に対しては無効です。

**S** ユーザー・ルーチンのリンク・エディットが必要で、DFSORT の特定のフェーズで使用される他のルーチン (たとえば E3n ルーチン) とは別個にリンク・エディットしなければならないことを指定します。別個にリンク・エディットできるルーチンは、E11 および E31 出口ルーチンのみです。詳しくは、374 ページの『ユーザー出口ルーチンのダイナミック・リンク編集の方法』を参照してください。この値は、コピー処理に対しては無効です。

e に値を指定しない場合は、N と見なされます。

### HILEVEL=YES

▶▶—HILEVEL=YES—▶▶

次のことを指定します。

- E15 ルーチンが MODS ステートメントで識別されている場合は、これは COBOL で作成されています。
- E35 ルーチンが MODS ステートメント上に指定されている場合は、これは COBOL で作成されています。

E15 ルーチンおよび E35 ルーチンを MODS ステートメントで識別する場合は、これらのルーチンがともに COBOL で作成されている場合に限り、HILEVEL=YES を指定してください。E15 または E35 ルーチンを MODS ステートメントに指定しない場合は、HILEVEL=YES は無視されます。

注: HILEVEL=YES の代わりに COBOL=YES を使用できます。

注:

1. e パラメーターに N または C が指定されている各ルーチンに対しては、s パラメーターは同一のものであるか、あるいは省略する必要があります (ライブラリー連結は許されます)。これらのルーチンを SYSIN に入れることはできません。このようなルーチンは、それぞれがロード・モジュールである必要があります。
2. e パラメーターに T または S が指定されている各ルーチンは、任意のライブラリーまたは SYSIN に入れることができます。それらのルーチンがすべて同じライブラリーまたは SYSIN にある必要はありません (入れることはできません)。一部のルーチンをそれぞれ異なるライブラリーに (あるいは同じライブラリーに) 入れ、残りのルーチンを SYSIN に入れることもできます。そのような各ルーチンは、ライブラリーにある場合、オブジェクト・デックまたはロード・モジュールです。SYSIN にある場合は、オブジェクト・デックです。
3. 同じルーチンを DFSORT プログラムの入力フェーズ (すなわち、E1n ルーチン) と出力フェーズ (すなわち、E3n ルーチン) の両方で使用する場合は、それぞれの出口に対して、ルーチンの別々のコピーを用意します。
4. HILEVEL=YES を 4 番目のパラメーターとして C の代わりに使用し、E15 または E35 ルーチンが COBOL で作成されていることを示すことができます。この場合、E15 または E35 の 4 番目のパラメーターとして T が指定されると、DFSORT は終了します。E15 ルーチンおよび E35 ルーチンを MODS ステートメントで識別する場合は、これらのルーチンがともに COBOL で作成されている場合に限り、HILEVEL=YES を指定してください。
5. EXEC PARM パラメーターの E15=COB を 4 番目のパラメーターとして C の代わりに使用し、E15 が COBOL で作成されていることを示すことができます。この場合、E15 の 4 番目のパラメーターとして T が指定されると、DFSORT は終了します。
6. EXEC PARM パラメーターの E35=COB を 4 番目のパラメーターとして C の代わりに使用し、E35 が COBOL で作成されていることを示すことができます。この場合、E35 の 4 番目のパラメーターとして T が指定されると、DFSORT は終了します。
7. HILEVEL=YES の場合は、E15=COB または E35=COB を 4 番目のパラメーターとして C の代わりに使用して、出口が COBOL で作成されていることを示すことができます。出口の 4 番目のパラメーターは、N として指定しなければならないか、あるいは指定できません。
8. COBOL E15 または E35 を、従来の組み合わせまたはテープ作業データ・セット分類に指定すると、DFSORT は終了します。
9. s パラメーターと e パラメーターの両方を省略するために、exit=(n,m) を使用できます。
10. exit=(n,m,e) を使用すると、s パラメーターを省略できますが、パラメーターは省略できません。

## MODS 制御ステートメント

- 従来の組み合わせまたはテープ作業データ・セットの分類の場合、または e パラメーターに S または T が指定されている場合は、s パラメーターを指定します。

デフォルト: なし。出口ルーチンを使用する場合は、指定します。N は 4 番目のパラメーターのデフォルトです。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

SYSIN のユーザー出口ルーチンについては、70 ページの『システム DD ステートメント』を参照してください。

ユーザー・ルーチンの設計方法については、371 ページの『ユーザー出口ルーチンに関する規則』を参照してください。

MODS ステートメントを使用する場合、DFSORT 自体が使用する主記憶域を適切に割り振ることができるように、DFSORT はユーザー・ルーチンが必要とする主記憶域の大きさを知っておく必要があることに注意してください。ユーザー・プログラムが必要とする正確なバイト数 (システム・サービスの所要量を含む) が分からない場合は、少し多めに見積もってください。MODS ステートメントの m の値は、正確な値の場合も見積もりの値の場合も同じ書き方をします。見積もりの値の前に E を付けてはなりません。

## ユーザー出口ルーチンの指定 - 例

### 例 1

```
MODS E15=(ADDREC,552,MODLIB),E35=(ALTREC,11032,MODLIB)
```

#### E15

出口 E15 で、DFSORT はユーザー・ルーチンへ制御を移します。ユーザー・ルーチンは、DD 名が MODLIB のジョブ制御ステートメントにより定義されたライブラリーにあります。このメンバー名は ADDREC で、552 バイトを使用します。

#### E35

出口 E35 で、DFSORT はユーザー・ルーチンへ制御を移します。ユーザー・ルーチンは、DD 名が MODLIB のジョブ制御ステートメントにより定義されたライブラリーにあります。このメンバー名は ALTREC で、11032 バイトを使用します。

### 例 2

```
MODS E15=(COBOLE15,7000,,C),  
E35=(COBOLE35,7000,EXITC,C)
```

#### E15

出口 E15 で、DFSORT はユーザー・ルーチンへ制御を移します。ユーザー・ル

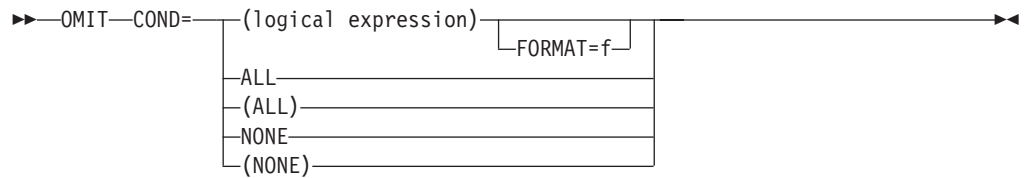
ーチンは COBOL で作成され、STEPLIB/JOBLIB またはリンク・ライブラリー  
にあります。このメンバー名は COBOLE15 で、7000 バイトを使用します。

**E35**

出口 E35 で、DFSORT はユーザー・ルーチンへ制御を移します。ユーザー・ルーチンは COBOL で作成されており、DD 名が EXITC のジョブ制御ステートメントにより定義されたライブラリーにあります。このメンバー名は COBOLE35 で、7000 バイトを使用します。

---

## OMIT 制御ステートメント

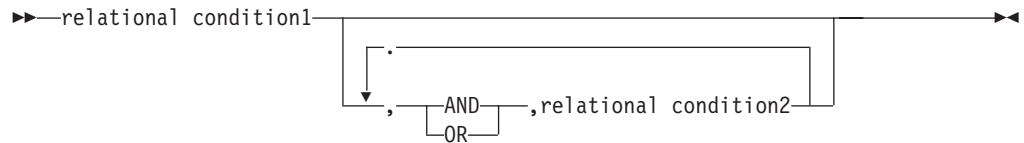


入力レコードのすべてを出力データ・セットに出力したくない場合は、OMIT ステートメントを使用してください。OMIT ステートメントは、組み込まないレコードを選択します。

同じ DFSORT 実行において、INCLUDE ステートメントまたは OMIT ステートメントのどちらかを指定できますが、その両方を指定できません。

DFSORT が短い INCLUDE/OMIT 比較フィールドを処理する方法は、VLSCMP/NOVLSCMP および VLSHRT/NOVLSHRT の設定により異なります。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張されるようなフィールドを指します。短いレコードの組み込みまたは省略の詳細については、181 ページの『OPTION 制御ステートメント』の VLSCMP および NOVLSCMP オプションの説明を参照してください。

論理式は、入力レコードの各フィールドに基づいて論理的に組み合わせられた 1 つ以上の関係条件であり、以下の高度レベルで表すことができます。



論理式があるレコードに関して真である場合は、そのレコードは出力データ・セットから省略されます。

次の 4 種類の関係条件を使用できます。

1. 比較:

2 つの比較フィールド間の比較、あるいは 1 つの比較フィールドと 10 進数、16 進数、文字または現在日付の定数の間の比較を行うことができます。

## OMIT 制御ステートメント

たとえば、各レコードの最初の 6 バイトと最後の 6 バイトを比較し、このフィールドが一致するレコードを除外できます。あるいは、フィールドと現在の日付を比較して、過去のイベントのレコードを省略することもできます。

### 2. サブストリング比較テスト:

フィールド値内の定数または定数内のフィールド値を検索します。

たとえば、6 バイト・フィールド内の値を検索して、文字定数 'C'OK' を見つけ出し、フィールドのどこかに 'C'OK' のあるレコードのみを省くことができます。あるいは、3 バイト・フィールドの値で文字定数 'C'J69,L92,J82' を検索して、フィールド内に 'C'J69'、'C'L92'、または 'C'J82' が見つかるレコードを除去できます。

### 3. ビット論理テスト:

ビット・マスクまたは 16 進数マスク、あるいはビット定数を使用して、2 進数フィールドの選択したビットの状態 (オンまたはオフ) をテストします。

たとえば、ある 1 バイト・フィールド内のビット 0 および 2 がオンのレコードを除外できます。あるいは、ある 2 バイト・フィールド内のビット 3 および 12 がオンで、ビット 6 および 8 がオフのレコードを除去できます。

### 4. 日付比較:

有効な「世紀」ウィンドウを使用して、2 桁の年日付フィールドを 2 桁の年日付定数、現在またはその他の 2 桁の年日付フィールドと比較します。

たとえば、Z'yyymm' 日付フィールドが、1996 年 1 月と 2005 年 3 月の間にあるレコードのみを省略できます。あるいは、P'dddy' フィールドが、別の P'dddy' フィールドより小さいレコードのみを省略できます。

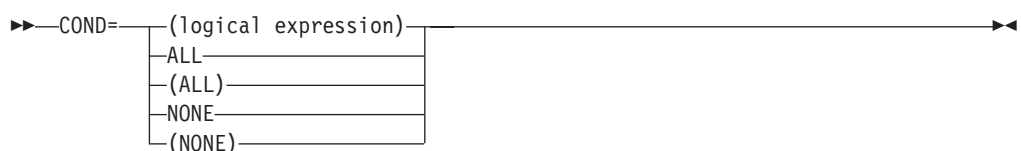
OMIT 制御ステートメントのパラメーターについては、130 ページの『INCLUDE 制御ステートメント』を参照してください。

OMIT ステートメントと OUTFIL ステートメントの OMIT パラメーターは、次の点で異なります。

- OMIT ステートメントは、すべての入力レコードに適用されます。これに対し OMIT パラメーターは、その OUTFIL グループの OUTFIL 入力レコードのみに適用されます。
- FORMAT=f は、OMIT ステートメントを用いて指定できますが、OMIT パラメーターを用いては指定できません。
- D2 形式は、OMIT ステートメントを用いて指定できますが、OMIT パラメーターを用いては指定できません。

OUTFIL OMIT パラメーターの詳細については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

## COND





**logical expression**

入力レコードのフィールドに基づいて、論理的に組み合わせられた 1 つ以上の関係条件を指定します。論理式があるレコードに関して真である場合は、そのレコードは出力データ・セットから省略されます。

**ALL または (ALL)**

すべての入力レコードが出力データ・セットから除外されることを指定します。

**NONE または (NONE)**

どの入力レコードも出力データ・セットから除外されないことを指定します。

デフォルト: なし。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**FORMAT**

▶▶—FORMAT=f————▶▶

FORMAT=f が使用できるのは、論理式全体の入力フィールドすべてが同じ形式の場合のみです。比較に指定可能なフィールド形式は、134 ページの表 10 に示されています。SS (サブストリング) は、サブストリング比較テストで唯一指定できるフィールド形式です。BI (無符号 2 進数) は、ビット論理テスト用に認められている唯一のフィールド形式です。Y2x 形式は、日付比較の場合に使用できる唯一のフィールド形式です。

デフォルト: なし。COND=(logical expression) パラメーターに組み込まれていない場合は、必ず指定する必要があります。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

注: FORMAT と COND の両方に形式値を指定すると、DFSORT は通知メッセージを出し、COND の形式値を使用し (f は各制御フィールドごとに指定します)、FORMAT で指定した形式値は使用しません。

**出力データ・セットからレコードを除外する — 例****例**

```
OMIT COND=(27,1,CH,EQ,C'D',&,
           (22,2,BI,SOME,X'C008',|,
           28,1,BI,EQ,B'.1....01'))
```

このステートメントは、次の条件を満たすレコードを除外します。

- 27 バイト目が D の場合

## OMIT 制御ステートメント

### AND

- 22 ～ 23 バイト目のビット 0、1、および 12 のいずれかがオン (ただし、必ずしもすべてのビットがオンではない) か、または (OR)、28 バイト目が指定パターン (ビット 1 がオン、ビット 6 がオフ、ビット 7 がオン) と等しい場合

AND および OR 演算子は、AND および OR 記号を用いて書くことができ、また括弧を使用すると AND および OR の評価の順序を変更できます。

論理式の詳細については、130 ページの『INCLUDE 制御ステートメント』を参照してください。

OPTION 制御ステートメント

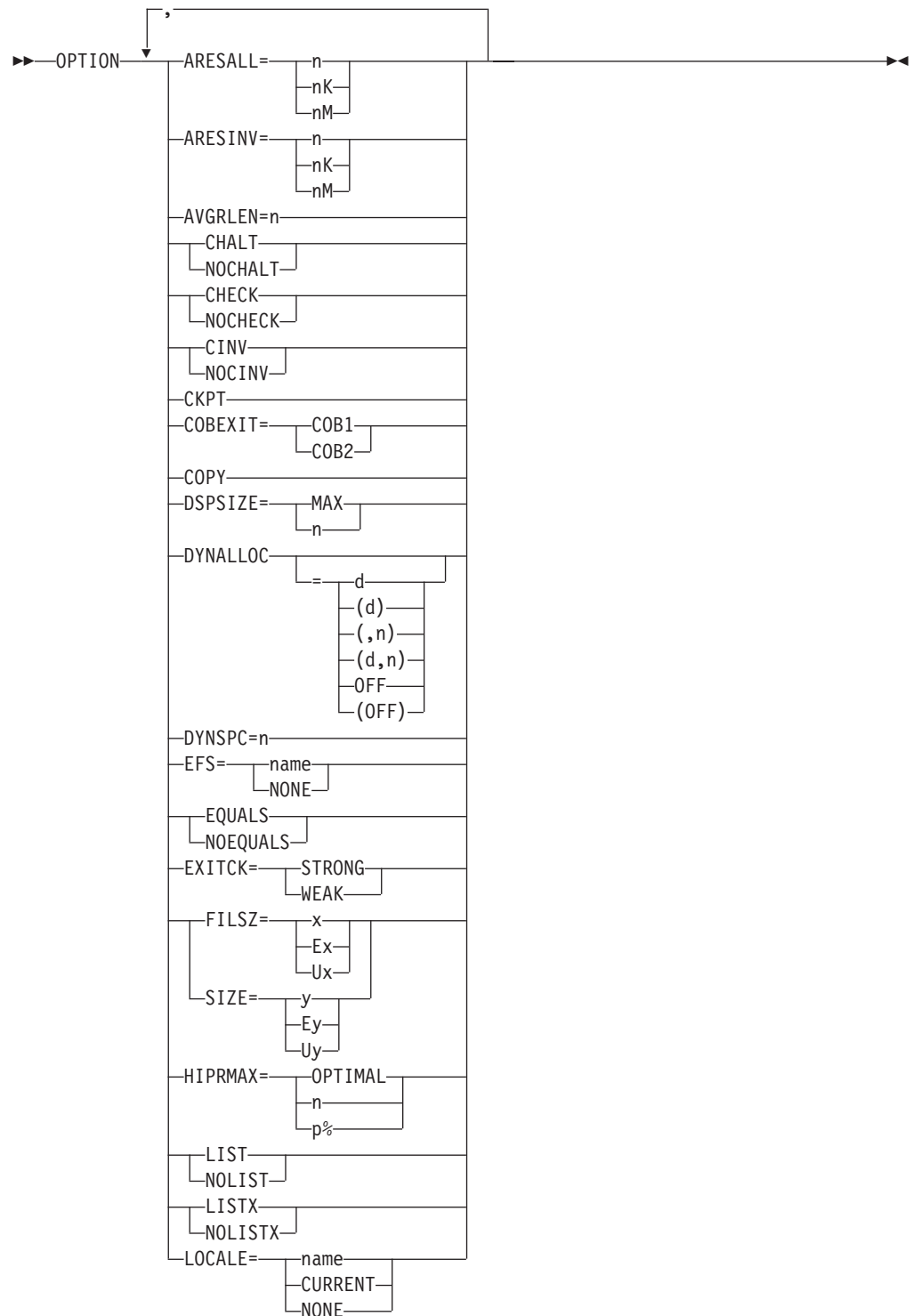


図 13. OPTION 制御ステートメントの構文図 (1/3)

## OPTION 制御ステートメント

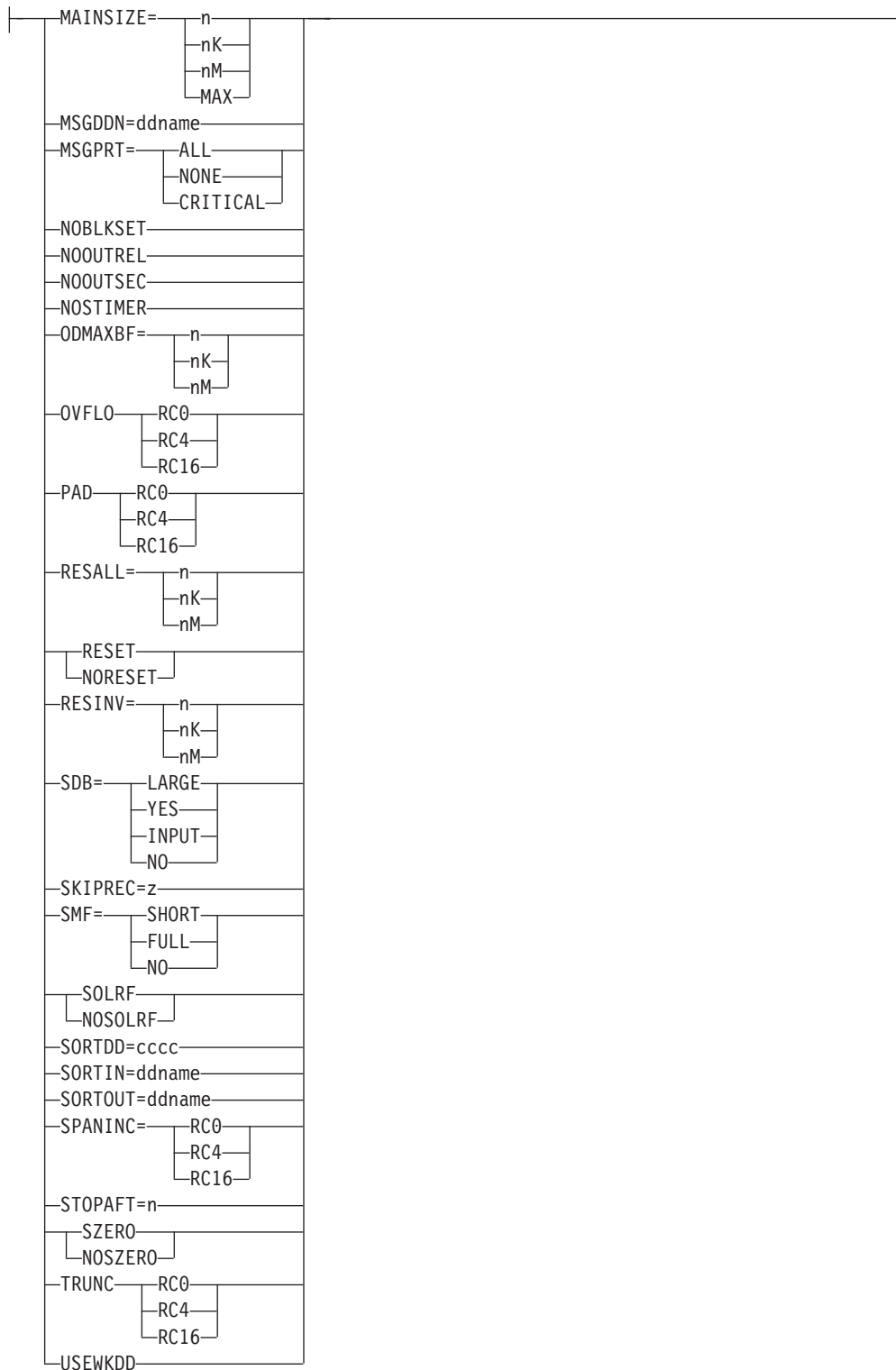


図 13. OPTION 制御ステートメントの構文図 (2/3)

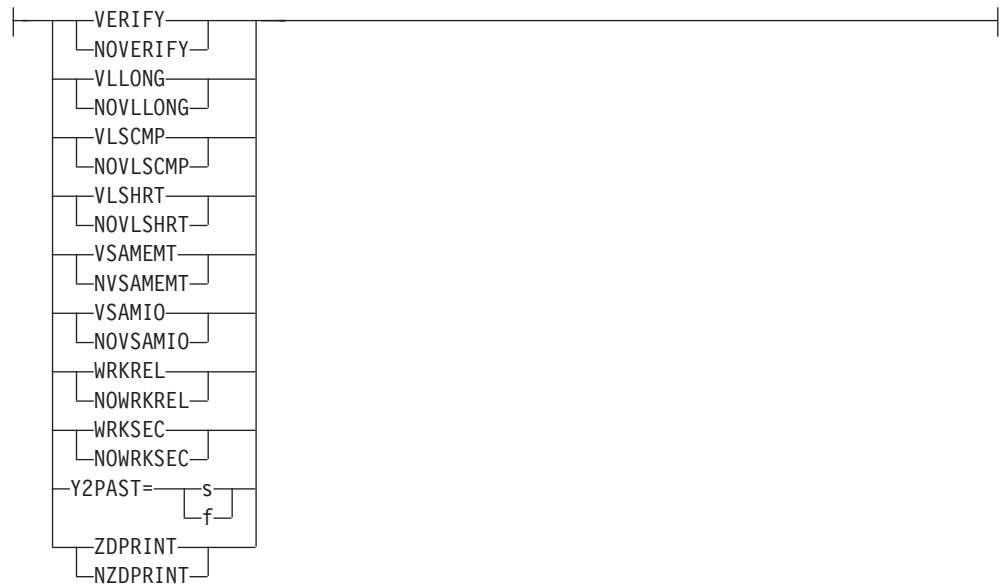


図 13. OPTION 制御ステートメントの構文図 (3/3)

**構文図に関する注意事項:** キーワード EFS、LIST、NOLIST、LISTX、NOLISTX、MSGDDN、MSGPRT、SMF、SORTDD、SORTIN、SORTOUT、および USEWKDD が使用されるのは、これらが拡張パラメーター・リストにより渡される OPTION 制御ステートメント上に指定されている場合、または DFSPARM データ・セットに指定されている場合だけです。それらが、SYSIN または SORTCNTL データ・セットから読み込まれた OPTION ステートメントに指定されている場合、キーワードは認識されますが、パラメーターは無視されます。

OPTION 制御ステートメントを使用すれば、インストール時に使用できるオプションの一部 (たとえば、EQUALS および CHECK) を指定変更したり、その他の任意指定の情報 (たとえば、DYNALLOC、COPY、および SKIPREC) を指定できます。

OPTION ステートメントで使用できるオプションの中には、SORT または MERGE ステートメントでも使用できるものがあります (たとえば FILSZ や SIZE など)。これらのオプションは OPTION ステートメントで指定するほうが適切です。指定変更の規則については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

OPTION ステートメント・オプションの別名の詳細については、個々のオプションの説明に記載されています。使用可能な別名は、233 ページの表 31 に要約されています。

DFSORT は次に挙げるオペランドを受け入れますが、これらの処理は行いません。このようなオペランドは、ALGQ、APP、APPEND、BIAS=value、BLKSET、CASCADE、DIAG、ERASE、EXCPVR=value、MAXPFI=value、NEW、NEWFILE、NODIAG、NOERASE、NOINC、NOSWAP、OPT=value、REP、REPLACE、WRKADR=value、WRKDEV=value、および WRKSIZ=value です。

## OPTION 制御ステートメント

### ARESALL



ARESALL インストール・オプションを一時的に指定変更し、仮想記憶域の 16 メガバイト境界より上にシステム用に予約するバイト数を指定します。

ARESALL は、仮想境界より上の主記憶域の大きさだけに適用されます。16 メガバイト境界より上の使用可能記憶域は大きいため、このオプションは通常は必要ありません。(ARESALL のデフォルトは 0 バイト)。RESALL オプションは、16 メガバイト仮想境界より下の主記憶域の大きさに適用されます。

**n** n バイトの記憶域を予約することを指定します。

限界: 8 桁

**nK** 1024 バイトの n 倍の記憶域を予約することを指定します。

限界: 5 桁

**nM** 1048576 バイトの n 倍の記憶域を予約することを指定します。

限界: 2 桁

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### ARESINV



ARESINV インストール・オプションを一時的に指定変更して、仮想記憶域の 16 メガバイト境界より上に常駐するか、そのスペースを使用する呼び出しプログラムのユーザー出口に予約するバイト数を指定します。予約されるスペースは、呼び出しプログラム自体の実行可能コード用に使用されるものではありません。ARESINV は、DFSORT が動的に呼び出される場合のみ使用されます。

ARESINV は、16 メガバイト境界より上の主記憶域の大きさだけに適用されます。RESINV オプションは、16 メガバイト境界より下の主記憶域の大きさに適用されます。

**n** n バイトの記憶域を予約することを指定します。

限界: 8 桁

**nK**

1024 バイトの n 倍の記憶域を予約することを指定します。

限界: 5 桁

**nM**

1048576 バイトの n 倍の記憶域を予約することを指定します。

限界: 2 桁

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**AVGRLLEN**

▶▶—AVGRLLEN=n—◀◀

可変長レコードの分類アプリケーションの入力レコード長の平均をバイト単位で指定します。この値は、入力ファイル・サイズを決定する必要がある場合に使用されます。分類アプリケーションの場合、この値が重要になります。動的作業データ・セット割り振りと同様に、いくつかの内部最適化のためにこの値が使用されるためです (OPTION DYNALLOC を参照)。ファイル・サイズの考慮事項については、625 ページの『入出力データ・セット特性の正確な指定』と 688 ページの『作業データ・セットの割り振り』を参照してください。

**n** 入力レコード長の平均を指定します。n の値は 4 から 32767 の範囲で指定します。また、4 バイトのレコード記述子ワード (RDW) を含んでいる必要があります。

**注:**

1. AVGRLLEN=n および OPTION ステートメントは、両方とも指定されていれば、RECORD ステートメントの L5 値 (LENGTH オペランド) を指定変更します。RECORD ステートメントの L5 値は、ブロック・セットの場合は無視されます。
2. AVGRLLEN=n の代わりに L5=n を使用できます。

デフォルト: AVGRLLEN=n が指定されていない場合は、DFSORT は最大レコード長の 2 分の 1 を平均レコード長として使用します。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**CHALT または NOCHALT**

▶▶—CHALT  
—NOCHALT—◀◀

## OPTION 制御ステートメント

CHALT インストール・オプションを一時的に変更して、形式 CH のフィールドを形式 AQ と同様に代替照合順序により変換するか、あるいは形式 AQ のみを変換するかを指定します。

### CHALT

DFSORT が代替照合順序を使用して、形式 CH および AQ の文字制御フィールドを変換することを指定します。

### NOCHALT

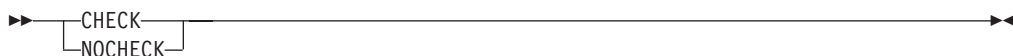
形式 CH フィールドを変換しないことを指定します。

注: SORT、MERGE、INCLUDE、または OMIT フィールドでロケール処理を使用する場合、CHALT は使用できません。特定のフィールドで代替順序処理が必要な場合は、形式 AQ を使用します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## CHECK または NOCHECK



CHECK インストール・オプションを一時的に変更して、E35 ユーザー出力ルーチンを使用するアプリケーションのために、出力データ・セットを用いないでレコード数を検査するかどうかを指定します。

### CHECK

レコード数を検査することを指定します。

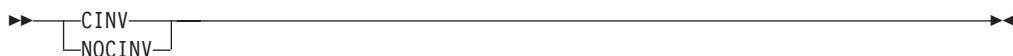
### NOCHECK

レコード数の検査を行わないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## CINV または NOCINV



CINV インストール・オプションを一時的に指定変更して、DFSORT が VSAM データ・セットに制御インターバル・アクセスを使用できるかどうかを指定します。ブロック・セット手法は、可能な場合、VSAM 入力データ・セットに対する制御インターバル・アクセスを使用して、パフォーマンスを改善します。



**CINV**

可能な場合は、VSAM データ・セットに対する制御インターバル・アクセスを DFSORT が使用するよう指定します。

**NOCINV**

DFSORT が制御インターバル・アクセスを使用しないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**CKPT**

▶▶—CKPT—▶▶

ピアレッジ手法またはベール手法を使用する分類の場合、チェックポイント / 再始動機能を活動化します。

CKPT をサポートするのはピアレッジおよびベール手法だけであるため、チェックポイント / 再始動機能を使用するためには、ブロック・セット手法をバイパスする必要があります。インストール・オプション IGNCCKPT=NO は、実行時に CKPT が指定されると、ブロック・セットをバイパスします。NOBLKSET オプションを使用して、実行時にブロック・セットをバイパスすることも可能です。

チェックポイント / 再始動機能を使用する場合、SORTCKPT DD ステートメントをコーディングします (SORTCKPT DD ステートメントを参照)。

**注:**

1. CKPT の代わりに CHKPT を使用できます。
2. OUTFIL 処理のような機能 (これらはブロック・セット手法によるのみサポートされる) は、チェックポイント / 再始動機能の使用中には使用できません。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**COBEXIT**

▶▶—COBEXIT—  
           ┌──COB1──┐  
           └──COB2──┘

COBEXIT インストール・オプションを一時的に指定変更して、COBOL E15 および E35 ルーチン用のライブラリーを指定します。

## OPTION 制御ステートメント

### COB1

COBOL E15 および E35 ルーチンを、OS/VS COBOL 実行時ライブラリーを用いて、あるいは場合によりは COBOL 実行時ライブラリーを用いずに実行することを指定します。

### COB2

COBOL E15 および E35 ルーチンを、VS COBOL II 実行時ライブラリーまたは言語環境プログラム実行時ライブラリーのどちらかを用いて実行することを指定します。

**注:** COBEXIT=COB2 の使用については、397 ページの『COBOL ユーザー出口に関する要件』を参照してください。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## COPY

▶▶ COPY ◀◀

このオプションを指定すると、E35 出口ルーチンによりすべてのレコードが処理されている場合をのぞき、DFSORT は、SORTIN データ・セットまたは挿入されたレコードを出力データ・セットにコピーします。レコードは E15 および E35 出口ルーチン、INCLUDE/OMIT、INREC、OUTREC、OUTFIL ステートメント、SKIPREC および STOPAFT パラメーターにより編集できます。SORTIN または E15 レコードがコピーされるたびに、その後で E35 に入ります。

コピー・アプリケーションでは、次のものを使用することができません。

- BDAM データ・セット
- ダイナミック・リンク編集

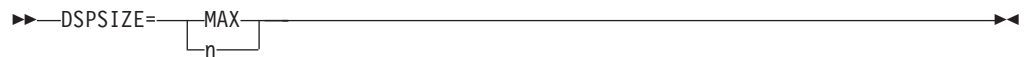
コピー・アプリケーションに適用されるその他の制約事項については、DFSORT メッセージ、コード、および診断の手引き リリース 14 のメッセージ ICE160A を参照してください。

**注:** ユーザー・ラベルは出力データ・セットにコピーされません。

**デフォルト:** なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## DSPSIZE



**DSPSIZE** インストール・オプションを一時的に指定変更して、データ空間分類で使用されるデータ空間の最大量を指定します。データ空間とは、実記憶装置、拡張記憶装置、および補助記憶装置により提供される連続した仮想記憶の区域で、このうちのどの記憶装置が必要になるかは、システムが決定します。

**DFSORT** は、データ空間を使用して大量のデータを分類することができるので、CPU 時間と経過時間を減少させます。

**DFSORT** が使用するデータ空間の大きさは、インストール時指定またはユーザー指定の **DSPSIZE** 値までに制限され、またユーザー・システムの **IEFUSI** 出口により制限されます。**DSPSIZE=MAX** (IBM 提供のデフォルト) は、**DFSORT** が入力ファイルのサイズ、およびシステムのページング活動に基づいて、使用するデータ空間の最大量を選択することを意味します。最大値をメガバイト単位で指定することにより、**DFSORT** が使用するデータ空間の大きさをさらに制限できます。

**DFSORT** が使用することに決定したデータ空間の大きさが十分であれば、**DFSORT** はユーザー・データを主記憶域内で分類し、追加の一時ワークスペースを必要としません。データ空間の大きさが不十分な場合は、**DFSORT** は **DASD** を一時ワークスペースとして使用します。データ空間分類が可能な場合は必ず、**DYNAUTO=NO** は **DYNAUTO=YES** に変更されます。データ空間分類が使用される場合は、ハイパー空間は使用されません。

### MAX

データ空間分類に使用するデータ空間の最大量を、**DFSORT** が動的に決定するように指定します。この場合、**DFSORT** は分類するファイルのサイズ、およびシステムのページング活動に基づいて、そのデータの空間の使用を決定します。

- n** データ空間分類に使用されるデータ空間の大きさをメガバイト単位で指定します。**n** の値は、0 から 9999 の範囲で指定します。使用されるデータ空間の実際の大きさが **n** を超えることはありませんが、分類されるファイルのサイズ、およびシステムのページング活動により、**n** より小さくなることはあります。

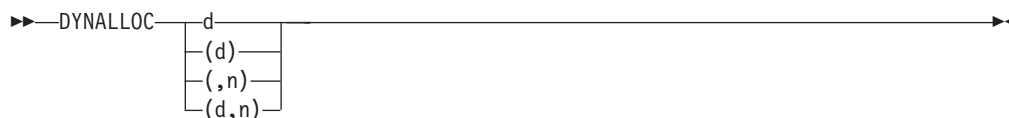
**n** がゼロの場合は、データ空間分類は使用されません。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. **DFSORT** オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. **DFSORT** オプションの指定 / 指定変更』を参照してください。

### DYNALLOC

## OPTION 制御ステートメント



必要なワークスペースを動的に割り振るタスクを DFSORT に割り当てます。プログラムで必要なワークスペースの大きさを計算して、JCL で指定する必要はありません。DFSORT はオペレーティング・システムの動的割り振り機能を使用して、ユーザー用にワークスペースを割り振ります。

DYNALLOC 使用の指針については 685 ページの『付録 A. ワークスペースの使用』を参照してください。

**d** 装置名を指定します。ユーザーのシステムでサポートされている、IBM 直接アクセス記憶装置または磁気テープ装置であれば、JCL UNIT パラメーターに指定するのと同じ方法で指定できます。また、DISK や SYSDA などのグループ名を指定することもできます。

最高のパフォーマンスを得るため、エミュレートされた 3390-9 装置 (RAMAC など) または別の高速 IBM DASD 装置を指定し、磁気テープ、仮想 (VIO) または実 3390-9 装置の指定は避けてください。

**n** 要求する作業データ・セットの最大数を指定します。255 よりも大きな値を指定すると、255 が使用されます。1 を指定し、ブロック・セット手法が選択されると、最大 2 つのデータ・セットが使用されます。32 よりも大きな値を指定し、ブロック・セット手法が選択されない場合は、最大 32 のデータ・セットが使用されます。

**注:** 仮想記憶域などのリソースを最適に割り振るために、作業データ・セットの数は必要以上に多く指定しないでください。

テープ作業データ・セットの場合は、(明示的にまたはデフォルトにより) 指定されるボリューム数がプログラムに対して割り振られます。プログラムは、標準ラベル・テープを要求します。

ICEMAC DYNAUTO=IGNWKDD が指定されていて、しかも OPTION USEWKDD が有効ではない場合を除き、SORTWKdd DD ステートメントを指定した場合、DYNALLOC は使用されません。

VIO=NO が有効な場合

- ワークスペースを、非一時データ・セット (DSNAME パラメーターで指定) に割り振ることができます。
- ユーザーが指定する装置 (d) が仮想記憶装置で、実記憶装置への再割り振りが失敗した場合は、DFSORT は VIO=NO を無視して仮想記憶装置を使用します。

**注:** メッセージ ICE165I は、作業データ・セットの割り振り / 使用に関する情報を提供します。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION 制御ステートメント

- ICEMAC DYNAUTO オプションを使用すれば、DYNALLOC を自動的に活性化できます。
- DYNALLOC で d を指定しない場合、d のデフォルトは、インストール時に ICEMAC DYNALLOC オプションで指定された (またはデフォルトが使用された) 値になります。
- DYNALLOC で n を指定しない場合、n のデフォルトは、インストール時に ICEMAC DYNALLOC オプションで指定した (またはデフォルトが使用された) 値になります。

DYNALLOC は、n を指定しないで、あるいは d を指定しないで、またはその両方とも指定しないで、指定できます。DYNALLOC が n の指定なしで指定され、DYNALLOC インストール・オプションの n の値の IBM 提供のデフォルトが選択された場合は、次のようになります。

- ブロック・セット手法の 1 つが選択されると、4 つの作業データ・セットが要求されます。
- ブロック・セット以外の手法が選択されると、3 つの作業データ・セットが要求されます。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### DYNALLOC=OFF

▶▶—DYNALLOC=—(OFF)—▶▶  
└—OFF—┘

DFSORT がワークスペースを動的に 割り振らない ように指示し、ICEMAC インストール・オプション DYNAUTO=YES、DYNAUTO=IGNWKDD、または実行時オプション DYNALLOC (OFF を指定しない) の機能を指定変更します。メモリー内の分類を実行可能なことをユーザーが承知しており、かつワークスペースの動的割り振りを抑制したい場合に、このオプションを使用してください。

### OFF

DFSORT がワークスペースを動的に割り振らないように指示します。

**注:** ハイパー分類またはデータ空間分類が有効な場合は、DYNALLOC=OFF が指定されている場合でも、必要であれば、DFSORT は動的割り振りを使用します。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### DYNSPC

▶▶—DYNSPC=n—▶▶

## OPTION 制御ステートメント

DYNSPC=ntemporarily は、入力ファイルのサイズが不明な場合に動的に割り振られるすべての作業データ・セットへの、デフォルトの 1 次スペース割り振り全体を指定する DYNSPC インストール・オプションを、一時的に指定変更します。すなわち、DFSORT が分類アプリケーションへの入力ファイルのサイズを決定できず、複数のレコードが FILSZ または SIZE の値で提供されないケースです。

通常、DFSORT は自動的に入力ファイル・サイズを決定できます。ただし、E15 が入力レコードのすべてを供給する場合、テープ・データ・セットに関する情報をテープ管理システムから利用できない場合、またはブロック・セットが選択されていない場合など、状況により DFSORT が正確なファイル・サイズを決定できないことがあります。このような場合において、FILSZ または SIZE 実行時オプションから複数のレコードが提供されず、また作業データ・セットの動的割り振りが使用されるとき、DFSORT は有効な DYNSPC 値を 1 次スペースの概算値として使用します。DFSORT は、20% の 1 次スペースを 2 次スペースとして使用します。1 次スペースがすでに割り振られていても、2 次スペース (最大 15 エクステンツ) は必要な場合のみ、割り振られます。

DYNSPC を使用することで、DFSORT 作業データ・セットで使用可能な DASD スペースの容量、およびこのアプリケーションで分類されるデータの容量に応じて、より大きい値または小さい値で、インストール時デフォルトを指定変更できます。ガイドラインとして、表 28 に、ブロック・セットが不特定数の 6000 バイトのレコードを分類するとき、3390 に割り振られた、シリンダーの概算 1 次スペースを示します。

表 28. DYNSPC 1 次スペースの例

DYNSPC 値 (メガバイト)	1 次スペース (シリンダー)
32	48
64	93
128	183
256	366
512	732

ファイル・サイズが不明な場合、DYNSPC 値が大きいほど、DFSORT はより多くのデータを分類できます。たとえば、表 28 に示す 1 次スペースを持つ動的割り振りワークスペース (ハイパー空間またはデータ空間なし) と、それに対応するすべての 2 次スペースだけを使用するテストにおいて、ブロック・セットは、DYNSPC=32 の場合およそ 150 メガバイト、DYNSPC=256 の場合およそ 1200 メガバイトのデータを分類できます。ハイパー空間またはデータ空間が、動的割り振りワークスペースと共に使用できる場合、DFSORT は、使用可能なハイパー空間または データ空間の容量に応じ、より大きい容量のデータを分類できます。

**n** すべての 動的割り振り作業データ・セットに割り振られる、デフォルトの 1 次スペース 全体 をメガバイトで指定します (n は、各データ・セットの 1 次スペースではありません)。n の値は 1 から 65535 の範囲で指定します。

## OPTION 制御ステートメント

使用可能な DASD スペースを超える値を指定しないでください。その値を使用する分類アプリケーションに対し、動的割り振りで障害が発生します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EFS

▶—EFS=——┐ name  
                 └— NONE

EFS インストール・オプションを一時的に指定変更して、DFSORT が拡張機能サポート (EFS) プログラムへ制御を渡すかどうかを指定します。詳細については、581 ページの『第 8 章 拡張機能サポートの使用』を参照してください。

#### name

DFSORT とのインターフェースのために呼び出される EFS プログラムの名前を指定します。

#### NONE

EFS プログラムに対する呼び出しが行なわれないことを指定します。

#### 注:

1. EFS が処理されるのは、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合だけです。
2. SORT、MERGE、INCLUDE、または OMIT フィールドのロケール処理を使用する場合は、EFS プログラムは使用しないでください。DFSORT のロケール処理を使用すれば、EFS プログラムを使用する必要がない場合があります。ロケール処理に関連する詳細については、本節の後半の LOCALE オプションの項を参照してください。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EQUALS または NOEQUALS

▶—EQUALS——┐  
                 └— NOEQUALS

EQUALS インストール・オプションを一時的に指定変更し、分類または組み合わせの場合に照合結果が同じになったレコードの順序を、入力から出力まで元の順序のまま維持するかどうかを指定します。

## OPTION 制御ステートメント

### EQUALS

元の順序を維持しなければならないことを指定します。

### NOEQUALS

元の順序を維持する必要がないことを指定します。

分類アプリケーションの場合、出力レコードの順序は次のようになります。

- SORTIN ファイルのレコードの順。
- E15 ユーザー出力ルーチンにより挿入されたレコードの順。
- SORTIN からの入力内に挿入された E15 レコードの順。

組み合わせアプリケーションの場合、出力レコードの順序は次のようになります。

- SORTINnn ファイルからのレコードの順。照合が一致するレコードは、それぞれのファイルの増分番号順に出力されます。たとえば、SORTIN01 からのレコードは、照合が一致する SORTIN02 からのレコードよりも先に出力されます。
- 同一ファイル増分番号に対する E32 ユーザー出力ルーチンからのレコードの順。照合が一致する E32 からのレコードは、それぞれのファイルの増分番号順に出力されます。たとえば、増分 0 のファイルからのレコードは、照合が一致する増分 4 からのいずれのレコードよりも先に出力されます。

### 注:

1. EQUALS が有効な場合は、すべての制御フィールドが占める合計のバイト数は 4088 を超えてはなりません。
2. EQUALS を使用すると、パフォーマンスが低下することがあります。
3. EQUALS が SUM で有効な場合、合計されたレコードの最初のレコードが保持されます。 NOEQUALS が SUM で有効な場合、保持されるレコードは予測できません。  
ブロック・セット以外の手法が選択されると、SUM が指定されている場合に NOEQUALS が強制されます。
4. 可変長レコードがテープ作業ファイルを使用して分類され、RDW が制御フィールドの一部である場合は、EQUALS は指定しないでください。
5. 分類するレコードの数は、4294967295 (4 ギガレコード - 1) を超えてはなりません。レコードの数がこの数を超過している場合は、メッセージ ICE121A が出力され、DFSORT は終了します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### EXITCK





EXITCK インストール・オプションを一時的に指定変更して、E15 または E35 ユーザー出口ルーチンから特定の無効な戻りコードを受信した場合、DFSORT を終了するか、続行するかを指定します。このパラメーターによる影響を受ける戻りコードの完全詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

**STRONG**

DFSORT が、E15 または E35 ユーザー出口ルーチンから無効な戻りコードを受信したときに、エラー・メッセージを発行し、終了することを指定します。

**WEAK**

DFSORT が、E15 および E35 ユーザー出口ルーチンからの特定の無効戻りコードを有効と解釈し、処理を続行することを指定します。

EXITCK=WEAK を使用した場合、E15 および E35 ユーザー出口ルーチンの論理内のエラーを検出することが難しくなることがあります。

注: EXITCK=WEAK は、次の場合、EXITCK=STRONG のように扱われます。

- 分類アプリケーションにテープ作業データ・セットが指定されている。
- 組み合わせアプリケーションにブロック・セット手法が選択されていない。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**FILSZ または SIZE**



FILSZ パラメーターは、分類または組み合わせに使用されるレコードの正確な数、または分類に使用されるレコードの見積数を指定します。SIZE パラメーターは、入力データ・セット内のレコードの正確な数、または入力データ・セット内のレコードの見積数を指定します。指定されたレコード数は、次の 2 つの目的のために DFSORT が使用します。

1. 分類または組み合わせされたレコードの実際の数、または入力データ・セット内のレコード数が、予定したレコードの正確な数と等しいかどうかを検査するため。FILSZ=x または SIZE=y はこの検査を実行し、数が等しくない場合はメッセージ ICE047A を出して処理を終了させます。
2. 分類アプリケーションに使用する入力ファイルのサイズを決定するため。DFSORT は、ユーザーが指定したレコード数および他のパラメーター (AVGRLLEN など) に基づいて計算を行い、分類するバイト数の合計を見積もります。分類実行の場合、この値が重要になります。動的作業データ・セッ

## OPTION 制御ステートメント

ト割り振りと同様に、いくつかの内部最適化のためにこの値が使用されるためです (OPTION DYNALLOC を参照)。分類実行に入力レコード数が指定されていない (または見積もりだけ指定されている) 場合は、DFSORT は、最適化および割り振りに使用するファイル・サイズを自動的に計算しようとします。

指定された FILSZ または SIZE の値のタイプ (x/y、Ux/Uy、Ex/Ey、または指定なし) は、DFSORT が上記の 2 つの機能を実行する方法を制御し、パフォーマンスと作業データ・セット割り振りに大きな影響を与えます。ファイル・サイズの考慮事項については、621 ページの『第 9 章 効率の改善』と 690 ページの『ファイル・サイズおよび動的割り振り』を参照してください。

### x または y

分類または組み合わせられるレコードの正確な数 (x) または入力データ・セット内のレコードの正確な数 (y) を指定します。この値は、常に、レコード検査およびファイル・サイズ計算の両方で使用されます。FILSZ=x または SIZE=y を使用することにより、x または y に基づくファイル・サイズの計算を DFSORT に強制的に実行させ、また x または y が正確でない場合は、DFSORT に分類または組み合わせアプリケーションを終了させることができます。

FSZEST=NO インストール・オプションが有効で、FILSZ=x または SIZE=y のどちらかが指定されている場合は、実際のレコード数が指定された正確な値 (x または y) と異なっていると、DFSORT は終了します。この場合、終了前に実際のレコード数が、メッセージ ICE047A (場合によりはメッセージ ICE054I) の IN フィールドに入れられます。ただし、FSZEST=YES インストール・オプションが有効な場合は、DFSORT は、FILSZ=x または SIZE=y を、それぞれ FILSZ=Ex または SIZE=Ey と同様に扱います。また、実際の数字が x または y と同じではなくても終了しません。

FILSZ=0 の場合は、ハイパー分類、データ空間分類、およびワークスペースの動的割り振りは使用されず、分類または組み合わせが行なわれたレコードの数が 0 でない限り、メッセージ ICE047A を出して終了します。E15 ユーザー出口が存在しない場合は、SIZE=0 はハイパー分類およびワークスペースの動的割り振りに関して同じ効果をもち、入力データ・セット内のレコードの数が 0 でない場合は、メッセージ ICE047A を出して終了します。

- x** 分類または組み合わせられるレコードの正確な数を指定します。この場合、入力データ・セット内のレコードの数、E15 または E32 により挿入または削除されるレコードの数、および INCLUDE/OMIT ステートメント、SKIPREC、および STOPAFT により削除されるレコードの数を考慮に入れる必要があります。分類または組み合わせを行なうレコードの数がなんらかの方法で大幅に変更される場合は、必ず x を変更する必要があります。
- y** 入力データ・セット内のレコードの正確な数を指定します。この場合、STOPAFT により削除されるレコードの数を考慮に入れる必要があります。入力データ・セット内のレコードの数がいくらかでも変更された場合は、y を必ず変更します。

限界: 28 桁 (有効数字 15 桁)

**Ex または Ey**

分類されるレコードの見積数 (x)、または入力データ・セット内のレコードの見積数 (y) を指定します。この値はレコード検査には使用されませんが、この値はファイル・サイズの計算に使用されますが、使用されるのは、DFSORT がそれ自体の入力ファイル・サイズを適切に見積もることができなかった場合だけです。それ以外の場合は、DFSORT はこの値を無視します。どのような場合に見積レコード数が使用され、またどのような場合に DFSORT がこれを見積レコード数を見積もるかについての詳細は、689 ページの『作業データ・セットの動的割り振り』を参照してください。

FILSZ=E0 または SIZE=E0 は常に無視されます。

- x** 分類するレコードの数の見積もりを指定します。この場合、入力データ・セット内のレコードの数、E15 により挿入または削除されるレコードの数、および INCLUDE/OMIT ステートメント、SKIPREC、および STOPAFT により削除されるレコードの数を考慮に入れる必要があります。分類するレコードの数が大幅に変更される場合は、x を必ず変更します。
- y** 入力データ・セット内のレコードの見積数を指定します。この場合、STOPAFT により削除されるレコードの数を考慮に入れる必要があります。入力データ・セット内のレコードの数が大幅に変更される場合は、y を必ず変更します。

限界: 28 桁 (有効数字 15 桁)

**Ux または Uy**

分類されるレコードの数 (x)、または入力データ・セット内のレコードの数 (y) を指定します。この値はレコード検査には使用されませんが、ファイル・サイズ計算には必ず使用されます。FILSZ=Ux または SIZE=Uy を使用すると、x または y が正確でなくても、DFSORT を終了させないで、x または y に基づいてファイル・サイズの計算を、DFSORT に強制的に行わせることができます。

FSZEST インストール・オプションは、FILSZ=Ux または SIZE=Uy 処理では無効です。

FILSZ=U0 により、ハイパー分類、データ空間分類、およびワークスペースの動的割り振りは使用されず、分類が行われたレコードの実際数が 0 よりも大幅に大きい場合は、パフォーマンスの低下を引き起こすか、またはメッセージ ICE046A を出して終了してしまふことがあります。E15 ユーザー出口が存在しない場合は、SIZE=U0 は、ハイパー分類、データ空間分類、およびワークスペースの動的割り振りに関して同じ効果をもち、入力データ・セット内のレコードの実際数が 0 よりも大幅に大きい場合は、パフォーマンスの低下を引き起こすか、メッセージ ICE046A を出して終了してしまふことがあります。

- x** 分類するレコードの数を指定します。この場合、入力データ・セット内のレコードの数、E15 により挿入または削除されるレコードの数、および INCLUDE/OMIT ステートメント、SKIPREC、および STOPAFT により削除されるレコードの数を考慮に入れる必要があります。分類するレコードの数が大幅に変更される場合は、x を必ず変更します。
- y** 入力データ・セット内のレコードの数を指定します。この場合、

## OPTION 制御ステートメント

STOPAFT により削除されるレコードの数を考慮に入れる必要があります。入力データ・セット内のレコードの数が大幅に変更される場合は、y を必ず変更します。

限界: 28 桁 (有効数字 15 桁)

表 29 は、3 種類の FILSZ の相違点を要約したものです。

表 29. 各種 FILSZ の要約: インストール・オプション FSZEST=YES が指定されている場合、FILSZ=n は FILSZ=En と同等になります。

条件	FILSZ=n	FILSZ=Un	FILSZ=En
レコード数	正確	見積もり	見積もり
アプリケーション	分類、組み合わせ	分類	分類
n に誤りがあれば終了するか	Y	N	N
ファイル・サイズ計算に使用するか	Y	Y	DFSORT がファイル・サイズを計算できない場合
n には次のレコードが含まれる			
入力データ・セット	Y	Y	Y
E15 により挿入 / 削除されるレコード	Y	Y	Y
E32 により挿入されるレコード	Y	N	N
INCLUDE/OMIT により削除されたレコード	Y	Y	Y
SKIPREC により削除されるレコード	Y	Y	Y
STOPAFT により削除されるレコード	Y	Y	Y
レコード数が変更されたときの n の更新:	任意に	大幅に	大幅に
n=0 の効果	ハイパー分類および DYNALLOC が使用されない	ハイパー分類および DYNALLOC が使用されない	なし

表 30 は、3 種類の SIZE の相違点を要約したものです。

表 30. 各種 SIZE の要約: インストール・オプション FSZEST=YES が指定されている場合、SIZE=n は SIZE=En と同等になります。

条件	SIZE=n	SIZE=Un	SIZE=En
レコード数	正確	見積もり	見積もり
アプリケーション	分類、組み合わせ	分類	分類
n に誤りがあれば終了するか	Y	N	N
ファイル・サイズ計算に使用するか	Y	Y	DFSORT がファイル・サイズを計算できない場合
n には次のレコードが含まれる			

表 30. 各種 SIZE の要約 (続き): インストール・オプション FSZEST=YES が指定されている場合、SIZE=n は SIZE=En と同等になります。

条件	SIZE=n	SIZE=Un	SIZE=En
入力データ・セット	Y	Y	Y
E15 により挿入 / 削除されるレコード	N	N	N
E32 により挿入されるレコード	N	N	N
INCLUDE/OMIT により削除されたレコード	N	N	N
SKIPREC により削除されるレコード	N	N	N
STOPAFT により削除されるレコード	Y	Y	Y
レコード数が変更されたときの n の更新:	任意に	大幅に	大幅に
n=0 の効果	ハイパー分類および DYNALLOC が使用されない	ハイパー分類および DYNALLOC が使用されない	なし

**注:** SIZE または FILSZ パラメーターを使用して DFSORT に不正確な情報を与えると、DFSORT のパフォーマンスが悪い影響を受けることがあります。また、ワークスペースが動的に割り振られると、DASD スペースを無駄に使用したり、メッセージ ICE083A または ICE046A を出して終了してしまうことがあります。したがって、分類が行われるレコードの数が大幅に変更される場合は、必ずレコード数の値を更新することが重要です。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### HIPRMAX



HIPRMAX インストール・オプションを一時的に指定変更して、ハイパー分類用に使用されるハイパー空間の最大値を指定します。ハイパー空間とは、拡張記憶域または 64 ビット実モードの中央記憶装置に常駐する高パフォーマンスのデータ空間であり、補助記憶装置により提供されます (必要な場合)。ハイパー分類を使用すると入出力処理が減少するので、経過時間、EXCP カウント、およびチャネル使用も減少します。

以下のいくつかの要因により、アプリケーションが使用するハイパー空間が制限されることがあります。

## OPTION 制御ステートメント

- IEFUSI 出口が、ハイパー空間およびアプリケーションに使用可能なデータ空間の合計の大きさを制限することがあります。
- HIPRMAX が、下記で説明するように、アプリケーションで使用可能なハイパー空間の大きさを制限することがあります。
- DFSORT のハイパー空間を提供するために、十分な大きさの記憶域が使用可能である必要があります。使用可能な記憶域とは、新しいハイパー空間データを格納するために使用される記憶域のことであり、次の 2 つのタイプから構成されます。
  1. 空き記憶域。これはアプリケーションにより使用されていない記憶域のことです。
  2. 古い記憶域。これは、別のアプリケーションで使用される記憶域であり、そのデータがかなり長期間参照されていないため、システムは、この記憶域を補助記憶装置に移行して、新しいハイパー空間データのためのスペースを設けます。

使用可能な記憶域の大きさは、現行システムの活動に応じて、常に変動します。したがって、DFSORT は、ハイパー分類アプリケーションを通して記憶域のレベルをチェックし、使用可能な記憶域のレベルが低すぎる場合は、ハイパー空間から作業データ・セットへスイッチします。

- その他の並行ハイパー分類アプリケーションが、使用可能な記憶域の大きさをさらに制限することがあります。ハイパー分類アプリケーションは、システム上の他すべてのハイパー分類アプリケーションが必要としている記憶域を認識しているので、他のハイパー分類アプリケーションが必要とする記憶域を用いてハイパー空間のデータを提供することはありません。これにより、同一システム上で複数の並行ハイパー分類アプリケーションが同時期に開始しても、記憶域のオーバー・コミットメントを防ぐことができます。
- インストール・オプションの EXPMAX、EXPOLD、および EXPRES を使用しても、ハイパー分類アプリケーションで使用可能な記憶域の大きさをさらに制限できます。EXPMAX は、DFSORT ハイパー空間を提供するために任意の一時点で使用できる使用可能な記憶域の大きさを制限します。EXPOLD は、DFSORT ハイパー空間を提供するために任意の一時点で使用できる古い記憶域の大きさを制限します。EXPRES は、非ハイパー分類のアプリケーションが使用できる記憶域用に指定された大きさを別に確保しておきます。

これらの制限の中には、ハイパー分類アプリケーションの実行時間中は、システムや他のハイパー分類の活動により決定されるものもあります。したがって、ハイパー分類アプリケーションが使用するハイパー空間の大きさは、実行の度に变化するという事です。

HIPRMAX=n は、HIPRMAX の固定値を指定します。HIPRMAX=p% は、実行時にシステム上に構成された拡張記憶域に対して、ある割合で変化する値として HIPRMAX の値を指定します。64 ビットの実モードでは、HIPRMAX=p% は、中央記憶装置の適用可能な部分の割合を指定します。システム上に記憶域が変わると、HIPRMAX=p% の場合は、DFSORT が選択する HIPRMAX 値もそれに対応して変わりますが、HIPRMAX=n の場合は変わりません。シスプレックスの場合のように、システム間で DFSORT インストール・オプションを共用するときは、HIPRMAX=p% を使用して、アプリケーションにより選択されたシステ

## OPTION 制御ステートメント

ムに合わせて HIPRMAX 値を調整することで、HIPRMAX=n の場合よりも動的に HIPRMAX 値を変えることができます。

ハイパー分類に使用可能なハイパー空間の大きさが、レコードの一時記憶用として不十分な場合は、中間 DASD 記憶域がハイパー空間とともに使用されます。ハイパー空間の大きさが小さすぎて、パフォーマンスを改善できない場合は、ハイパー分類は使用されません。DYNAUTO=NO は、ハイパー分類用に DYNAUTO=YES に変更されます。

ハイパー分類は、多少の CPU 時間の低下をもたらすことがあります。CPU の最適化を考慮する場合は、HIPRMAX=0 を使用してハイパー分類を抑制できます。

**注:** HIPRMAX=OPTIMAL の代わりに HIPRLIM=OPTIMAL を使用できます。HIPRMAX=n の代わりに HIPRLIM=m を使用できます。HIPRLIM=m は、ハイパー空間の限度として、最も近いメガバイトに切り上げられた、4096 バイトの m 倍の限界値を指定します。m の値は、0 から 2559744 の範囲で指定します。m が 0 の場合、ハイパー分類は使用されません。

### OPTIMAL

ハイパー分類用に使用されるハイパー空間の最大値を、DFSORT が動的に決定することを指定します。

**n** ハイパー分類用に使用されるハイパー空間の最大値 (n メガバイトの限界値) を、DFSORT が動的に決定するように指定します。n の値は 0 から 32767 の範囲で指定します。n が 0 の場合、ハイパー分類は使用されません。

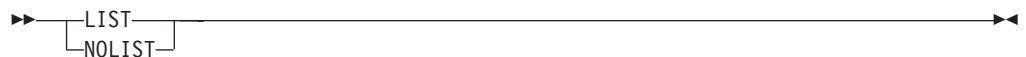
### p%

ハイパー分類用に使用されるハイパー空間の最大値を、構成される拡張記憶域の p% を限界として、DFSORT が動的に決定するように指定します。64 ビットの実モードでは、HIPRMAX=p% は、中央記憶装置の適用可能な部分の割合を指定します。p の値は 0 から 100 の範囲で指定します。p が 0 の場合、ハイパー分類は使用されません。p% の計算値は、32767 メガバイトを上限とし、メガバイト未満の端数は切り捨てられます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### LIST または NOLIST



LIST インストール・オプションを一時的に指定変更して、DFSORT プログラム制御ステートメントをメッセージ・データ・セットに書き込むかどうかを指定します。メッセージ・データ・セットの使用方法については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

## OPTION 制御ステートメント

### LIST

DFSORT 制御ステートメントをメッセージ・データ・セットに印刷することを指定します。

### NOLIST

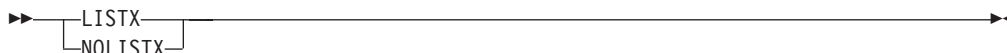
DFSORT 制御ステートメントをメッセージ・データ・セットに印刷しないことを指定します。

**注:** LIST または NOLIST は、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合のみ処理されます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### LISTX または NOLISTX



LISTX インストール・オプションを一時的に指定変更して、EFS プログラムにより戻されるプログラム制御ステートメントを、DFSORT がメッセージ・データ・セットに書き込むかどうかを指定します。メッセージ・データ・セットの使用方法については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

### LISTX

EFS プログラムにより戻される制御ステートメントをメッセージ・データ・セットに印刷することを指定します。

### NOLISTX

EFS プログラムにより戻される制御ステートメントをメッセージ・データ・セットに印刷しないことを指定します。

**注:**

1. LISTX または NOLISTX は、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合のみ処理されます。
2. 最後の指定変更規則が適用された後で EFS=NONE が有効な場合、NOLISTX が有効になります。
3. LISTX と NOLISTX は LIST と NOLIST とは関係なく使えます。
4. EFS 制御ステートメントの印刷についての詳細は、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。



## LOCALE



LOCALE インストール・オプションを一時的に指定変更して、ロケール処理を使用するかどうかを指定します。ロケール処理を使用する場合は、活動状態のロケールを指定します。

DFSORT の照合の動作を、使用する文化環境に合わせて修正できます。ユーザーの文化環境は、X/Open ロケール・モデルを使用して、DFSORT に定義されません。ロケールとは、ユーザーの文化環境に関する情報を記述する、カテゴリごとにグループ化されたデータの集まりです。

ロケールの照合カテゴリは、照合要素 (単一文字照合要素と複数文字照合要素) 間の相対順位を決定する順序宣言の集まりです。順序宣言は、照合規則を定義します。

ロケール処理が使用されると、活動ロケールは DFSORT の SORT、MERGE、INCLUDE、および OMIT 機能の動作に影響します。SORT および MERGE の場合は、活動ロケールは文字 (CH) 制御フィールドの処理のみに使用されます。INCLUDE および OMIT の場合は、活動ロケールは文字 (CH) 比較フィールドの処理のみに使用され、文字および 16 進数は文字 (CH) 比較フィールドと比較されます。

**name** ロケール処理を使用することを指定し、DFSORT 処理の間に活動状態にするロケールの名前を指定します。

ロケールは記述名を使用して指定されます。たとえば、フランス語とカナダの文化慣行を表す活動ロケールを設定するには、LOCALE=FR\_CA と指定します。記述ロケール名には、最高 32 文字まで指定できます。ロケール名は大文字 / 小文字の区別をしません。ロケールの命名規則の詳細については、[ロケールの使用](#) を参照してください。

IBM 提供のロケールとユーザー定義のロケールを使用できます。

DFSORT に入る以前の活動ロケールの状態は、DFSORT の完了時に復元されます。

**CURRENT**

ロケール処理が使用されることを指定し、DFSORT に入る際の現行活動ロケールが、DFSORT 処理の間も活動ロケールのままになります。

**NONE** ロケール処理が使用されないことを指定します。DFSORT は、ユーザー・データを照合し比較するために定義されたコード・ページの 2 進数のエンコードを使用します。

**注:**

1. LOCALE が処理されるのは、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合のみです。
2. IBM 提供のロケールを使用するには、DFSORT が、言語環境プログラム実行時ライブラリーにアクセスできる必要があります。たとえば、このライブ

## OPTION 制御ステートメント

ラリは SYS1.SCEERUN と呼ばれる場合があります。このライブラリーの名前が分からない場合は、システム管理者にお問い合わせください。ユーザー定義のロケールを使用するには、DFSORT は、上記のロケール・ロード・モジュールが入っているデータ・セットが入っているロード・ライブラリーにアクセスします。

- ロケール処理を SORT、MERGE、INCLUDE、または OMIT フィールドに使用する場合、
  - VLSHRT は SORT または MERGE には使用されません。
  - CHALT、INREC、EFS プログラム、または E61 ユーザー出口は使用しないでください。
- DFSORT の SORT、MERGE、INCLUDE、および OMIT 機能でのロケール処理では、必要な照合結果を作成するためのデータの事前処理または事後処理 (あるいはその両方) を行うアプリケーションに関するパフォーマンスを改善できます。ただし、ロケール処理は、必要な場合のみ使用します。文字エンコード値を使用した照合に関して、パフォーマンスの低下を示すことがあるためです。
- DFSORT ロケール処理により記憶域が余計に必要なことがあります。必要になる記憶域の量は、ロケールをサポートする環境とロケール自身により変わります。ロケール処理を使用する DFSORT アプリケーションに数メガバイト以上の REGION を指定することが必要になる場合があります。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## MAINSIZE



SIZE インストール・オプションを一時的に指定変更して、DFSORT が使用できる主記憶域の大きさを指定します。指定する値は、DFSORT のインストール時に設定した MINLIM 値よりも大きくする必要があります。

MAINSIZE は、16 メガバイト境界の上下の主記憶域の合計の大きさに適用されます。DFSORT が、16 メガバイトより上と下の記憶域をどれだけ割り振るかを決定しますが、その記憶域の合計が MAINSIZE を超えることはできません。

OUTFIL 処理で使用する記憶域は、次のいくつかの要因により、自動的に調整されます。

- 使用可能記憶域の合計
- 非 OUTFIL 処理の必要記憶域
- OUTFIL データ・セットの数およびその属性 (たとえば、ブロック・サイズ)

## OPTION 制御ステートメント

OUTFIL 処理は、ODMAXBF 限界およびユーザー・システムの記憶域限界 (たとえば、IEFUSI) の制約を受けますが、DFSORT 記憶域限界 (つまり、SIZE/MAINSIZE、MAXLIM、および TMAXLIM) の制約は受けません。DFSORT は、可能な限り、16 メガバイト境界より上の記憶域を、OUTFIL 処理に使用しようとしています。

主記憶域の割り振りの詳細については、632 ページの『主記憶域の調整』を参照してください。

**n** n バイトの記憶域を割り振ることを指定します。2097152000 よりも大きな値を指定すると、2097152000 が使用されます。

限界: 10 桁

### nK

1024 バイトの n 倍の記憶装置を割り振ることを指定します。2048000 キロバイトよりも大きな値を指定すると、2048000 キロバイトが使用されます。

限界: 7 桁

### nM

割り振られる記憶域が 1048576 バイトの n 倍であることを指定します。2000 メガバイトよりも大きな値を指定すると、2000 メガバイトが使用されます。

限界: 4 桁

### MAX

使用できる仮想記憶域の大きさを計算し、ブロック・セットが選択されたときには TMAXLIM または DSA インストール値まで、ブロック・セットが選択されていないときには、MAXLIM インストール値までの大きさの記憶域を割り振ることを、DFSORT に指示します。

**注:** MAINSIZE=value の代わりに CORE=value を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## MSGDDN

▶▶—MSGDDN=ddname—◀◀

MSGDDN インストール・オプションを一時的に指定変更して、メッセージ・データ・セットの代替 DD 名を指定します。次の場合は、MSGDDN が有効である必要があります。

- DFSORT を呼び出すプログラムが SYSOUT を使用し (たとえば、COBOL が SYSOUT を使用)、DFSORT メッセージとプログラム・メッセージを混在させたくない場合
- E15 および E35 ルーチンが COBOL で作成され、DFSORT メッセージとプログラム・メッセージを混在させたくない場合

## OPTION 制御ステートメント

- プログラムが DFSORT を複数回呼び出し、DFSORT の各呼び出しごとにメッセージを分けたい場合

DD 名には任意の 1 ~ 8 文字の名前を使用できますが、ジョブ・ステップ内では固有である必要があります。DFSORT が使用する名前 (たとえば SORTIN) は使用しないでください。指定された DD 名が実行時に使用できない場合は、代わりに SYSOUT が使用されます。メッセージ・データ・セットの使用の詳細については、*DFSORT* メッセージ、コード、および診断の手引き リリース 14 を参照してください。

**注:** MSGDDN が処理されるのは、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合だけです。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### MSGPRT



MSGPRT インストール・オプションを一時的に指定変更して、メッセージ・データ・セットに書き込まれるメッセージのクラスを指定します。メッセージ・データ・セットの使用の詳細については、*DFSORT* メッセージ、コード、および診断の手引き リリース 14 を参照してください。

#### ALL

診断メッセージ (ICE800I から ICE999I) 以外のすべてのメッセージが印刷されることを指定します。制御ステートメントは、LIST が有効な場合のみ印刷されます。

#### CRITICAL

重要なメッセージだけを印刷することを指定します。制御ステートメントは、LIST が有効な場合のみ印刷されます。

#### NONE

メッセージおよび制御ステートメントを印刷しないことを指定します。

#### 注:

1. MSGPRT が処理されるのは、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合のみです。
2. MSGPRT=value の代わりに PRINT=value を使用できます。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### NOBLKSET

▶▶—NOBLKSET—▶▶

分類または組み合わせアプリケーションに通常使用されるブロック・セット手法を、DFSORT にバイパスさせます。このオプションを使用すると、一般的にパフォーマンスが低下します。

注: OUTFIL 処理のような機能 (これらはブロック・セット手法によりだけサポートされる) の場合は、NOBLKSET オプションが無視されます。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### NOOUTREL

▶▶—NOOUTREL—▶▶

OUTREL インストール・オプションを一時的に指定変更して、未使用の一時出力データ・セット・スペースを解放するかどうかを指定します。NOOUTREL は、未使用の一時出力データ・セット・スペースを解放しないことを意味します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### NOOUTSEC

▶▶—NOOUTSEC—▶▶

OUTSEC インストール・オプションを一時的に指定変更して、一時または新規の出力データ・セットに対して自動 2 次割り振りを使用するかどうかを指定します。NOOUTSEC は、出力データ・セットに対して自動 2 次割り振りを使用しないことを意味します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION 制御ステートメント

### NOSTIMER

▶▶—NOSTIMER—▶▶

STIMER インストール・オプションを一時的に変更して、DFSORT が STIMER マクロを使用できるかどうかを指定します。NOSTIMER は、DFSORT が STIMER マクロを使用しないことを意味します。したがって、プロセッサ時間データは、SMF レコードまたは ICETEXIT 終了インストール出口に提供される統計に表示されません。

ユーザー出口がチェックポイントをとり、STIMER=YES がインストール時のデフォルトである場合、このパラメーターを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### ODMAXBF

▶▶—ODMAXBF= $\left. \begin{array}{l} n \\ -nK \\ -nM \end{array} \right\}$ —▶▶

ODMAXBF インストール・オプションを一時的に指定変更して、各 OUTFIL データ・セット用に DFSORT が使用できる最大バッファ・スペースを指定します。特定の OUTFIL データ・セットに使用されるバッファ・スペースの実際の大きさは、ODMAXBF 限界を超えないが、限界値よりも小さい値です。OUTFIL 処理は、分類アプリケーション、コピー・アプリケーション、および組み合わせアプリケーションに関してブロック・セット手法でサポートされません。

OUTFIL 処理の使用記憶域は、使用可能な記憶域、OUTFIL でない処理に必要な記憶域、および OUTFIL データ・セットとその属性 (たとえばブロック・サイズ) の合計に応じて自動的に調整されます。OUTFIL 処理は有効な ODMAXBF 限界、およびシステム記憶域限界 (たとえば IEFUSI) に制約されますが、DFSORT 記憶域限界 (つまり SIZE、MAXLIM、および TMAXLIM) には制約されません。DFSORT は、可能な限り、16 メガバイト境界より上の記憶域を、OUTFIL 処理に使用しようとします。

ODMAXBF を 2 メガバイトより下に入れると、アプリケーションの性能低下を引き起こすことがあります。OUTFIL 処理で使用する記憶域の大きさに問題があると考えられる場合は、それが必要になることもあります。ODMAXBF を引き上げると、アプリケーションの EXCP を改善できますが、必要な記憶域も増加します。

**n** 各 OUTFIL データ・セットに、最大 n バイトのバッファ・スペースを使

## OPTION 制御ステートメント

用することを指定します。262144 よりも小さい値を指定すると、262144 が使用されます。16777216 よりも大きな値を指定すると、16777216 が使用されます。

限界: 8 桁

### nK

各 OUTFIL データ・セットに、1024 バイトの n 倍の最大バッファ・スペースを使用することを指定します。256 キロバイトよりも小さな値を指定すると、256 キロバイトが使用されます。16384 キロバイトよりも大きな値を指定すると、16384 キロバイトが使用されます。

限界: 5 桁

### nM

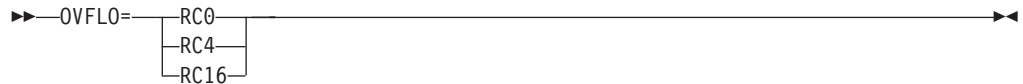
各 OUTFIL データ・セットに、最大 1048576 バイトの n 倍のバッファ・スペースを使用することを指定します。0 メガバイトを指定すると、256 キロバイトが使用されます。16 メガバイトよりも大きな値を指定すると、16 メガバイトが使用されます。

限界: 2 桁

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OVFLO



OVFLO インストール・オプションを一時的に指定変更して、BI、FI、PD または ZD 合計フィールドがオーバーフローしたときの、DFSORT の処置を指定します。

### RC0

DFSORT に対して、合計フィールドがオーバーフローしたら、メッセージ ICE152I を (一度) 発行し、戻りコードを 0 に設定して処理を続行するよう指定します。合計がオーバーフローしたレコードのペアは、合計されないまま残り、どちらも削除されません。合計のオーバーフローにより、その後の集計処理が中止されることはありません。

### RC4

DFSORT に対して、合計フィールドがオーバーフローしたら、メッセージ ICE152I を (一度) 発行し、戻りコードを 4 に設定して、処理を続行するよう指定します。合計がオーバーフローしたレコードのペアは、合計されないまま残り、どちらも削除されません。合計のオーバーフローにより、その後の集計処理が中止されることはありません。

## OPTION 制御ステートメント

### RC16

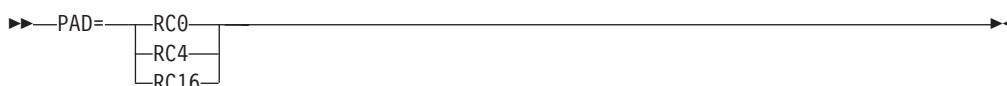
DFSORT に対して、合計フィールドがオーバーフローしたら、メッセージ ICE195A を発行して終了し、戻りコード 16 を戻すよう指定します。

**注:** 合計のオーバーフローに対して設定される戻りコード 0 または 4 は、他の理由によるもっと大きい値の戻りコードで上書きされることがあります。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### PAD



PAD インストール・オプションを一時的に指定変更して、DFSORT が LRECL 埋め込みを許可する場合に、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときの DFSORT の処置を指定します。

### RC0

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE17II を発行し、戻りコードを 0 に設定して処理を続行するよう指定します。

### RC4

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE17II を発行し、戻りコードを 4 に設定して処理を続行するよう指定します。

### RC16

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE196A を発行して終了し、戻りコード 16 を戻すよう指定します。

### 注:

1. LRECL 埋め込みに対して設定される戻りコード 0 または 4 は、他の理由によるもっと大きい値の戻りコードで上書きされることがあります。
2. ICEGENER アプリケーションの場合は、GNPAD 値が使用され、PAD 値は無視されます。
3. ある条件下の LRECL 埋め込み (テープ作業データ・セットの分類など) では、DFSORT は ICE043A を発行して、戻りコード 16 で終了します。このような場合は、PAD 値は効力がありません。
4. DFSORT は次の場合は LRECL 埋め込みを検査しません。
  - a. SORTIN DD (分類 / コピー)、SORTINnn DD (組み合わせ)、または SORTOUT DD が指定されていないとき。

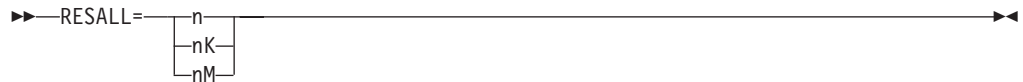


- b. SORTIN DD (分類 / コピー)、SORTINnn DD (組み合わせ)、または SORTOUT DD に VSAM データ・セットが指定されているとき。
- 5. DFSORT は、LRECL 埋め込みに関して OUTFIL データ・セットを検査しません。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**RESALL**



RESALL インストール・オプションを一時的に指定変更して、システム使用のために REGION に予約するバイト数を指定します。通常、システム使用のために領域で使用可能でなければならない主記憶域は、4 キロバイト (標準のデフォルト) です。しかし、これでは不十分な場合があります。たとえば、ユーザーのインストール・システムに BSAM/QSAM モジュールが常駐していない場合、データ・セットをオープンするユーザー出口がある場合、または COBOL 出口がある場合などです。RESALL は MAINSIZE/SIZE=MAX が有効な場合のみ使用されます。

RESALL は、16 メガバイト境界より下の主記憶域の大きさだけに適用されます。ARESALL オプションは、16 メガバイト境界より上の主記憶域の大きさに適用されます。

**n** n バイトの記憶域を予約することを指定します。4096 よりも小さい値を指定すると、4096 が使用されます。

限界: 8 桁

**nK**

1024 バイトの n 倍の記憶域を予約することを指定します。4 キロバイトよりも小さい値を指定すると、4 キロバイトが使用されます。

限界: 5 桁

**nM**

1048576 バイトの n 倍の記憶域を予約することを指定します。0 メガバイトを指定すると、4 キロバイトが使用されます。

限界: 2 桁

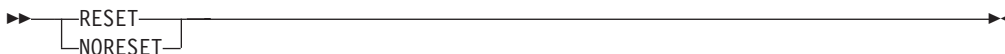
**注:** MODS ステートメントにより活動化されるユーザー出口に必要な記憶域を予約する良い方法は、MODS ステートメントの m パラメーターを使用することです。

## OPTION 制御ステートメント

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### RESET または NORESET



RESET インストール・オプションを一時的に指定変更して、DFSORT が、REUSE で定義された VSAM 出力データ・セットを NEW または MOD データ・セットとして処理するかどうかを指定します。

#### RESET

DFSORT が、REUSE で定義された VSAM 出力データ・セットを NEW データ・セットとして処理することを指定します。頻繁に使用される RBA はゼロにリセットされ、出力データ・セットは実際は、当初空のクラスタとして扱われます。

#### NORESET

DFSORT が、REUSE で定義された VSAM 出力データ・セットを MOD データ・セットとして処理することを指定します。頻繁に使用される RBA はリセットされず、出力データ・セットは実際は、当初空でないクラスタとして扱われます。

注: REUSE なしで定義された VSAM 出力データ・セットは、MOD データ・セットとして処理されます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### RESINV



RESINV インストール・オプションを一時的に指定変更し、呼び出しプログラム使用のための REGION 内に予約されるバイト数を指定します。RESINV は、DFSORT が動的に呼び出され、MAINSIZE/SIZE=MAX が有効な場合のみ使用されます。

RESINV は、16 メガバイト境界より下の主記憶域の大きさだけに適用されます。ARESINV オプションは、16 メガバイト境界より上の主記憶域の大きさに適用されます。

## OPTION 制御ステートメント

この余分のスペースは、通常、DFSORT が実行中に呼び出しプログラムまたはユーザー出口によりデータを処理するために必要です (PL/I や COBOL により呼び出された分類アプリケーションの場合など)。したがって、ユーザーの呼び出しプログラムのユーザー出口がデータ・セット処理を行わない場合は、このパラメーターを指定する必要はありません。予約されるスペースは、呼び出しプログラム自体の実行可能コード用に使用されるものではありません。

必要なスペースの大きさは、どのようなルーチンが用意されているか、どのようにデータが保管されているか、またどのようなアクセス方式が使用されるかにより決まります。

**n** n バイトの記憶域を予約することを指定します。

限界: 8 桁

**nK**

1024 バイトの n 倍の記憶域を予約することを指定します。

限界: 5 桁

**nM**

1048576 バイトの n 倍の主記憶域を予約することを指定します。

限界: 2 桁

**注:** ユーザー出口に必要な記憶域を予約する良い方法は、MODS ステートメントの *m* パラメーターを使用することです。

**デフォルト:** 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**適用できる機能:** 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## SDB



SDB インストール・オプションを一時的に指定変更して、ブロック・サイズがゼロと指定されたか、またはデフォルトのゼロが使用された場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用するかどうかを指定します。システム決定ブロック・サイズは、SMS 管理データ・セットと非 SMS 管理データ・セットの両方に適用され、出力データ・セットが入れられる記憶装置のスペースを最も有効に使用します。

OS/390 リリース 10 以上では、DFSORT は、テープ出力データ・セットに、32760 バイトより大きいシステム決定の最適ブロック・サイズを選択できます。

DASD およびテープ出力データ・セットの場合にシステム決定最適ブロック・サイズを DFSORT に使用させたい場合は、次のいずれかの値を指定します。

## OPTION 制御ステートメント

- DFSORT が、32760 バイトより大きいテープ出力ブロック・サイズを選択できるようにしたい場合、SDB=LARGE。
- DFSORT が、32760 バイトより小か等しい出力ブロック・サイズを選択できるようにしたい場合、SDB=YES (またはその別名 SDB=SMALL)。
- 32760 バイトより大きいブロック・サイズをもつテープ入力データ・セットが使用されるときだけ、DFSORT が、32760 バイトより大きいテープ出力ブロック・サイズを選択できるようにしたい場合、SDB=INPUT。

DFSORT は、有効な BLKSZLIM より大きいテープ出力ブロック・サイズは選択しません。特に、BLKSZLIM のデフォルト 32760 が有効な場合は、DFSORT は 32760 バイトより大きい出力ブロック・サイズを選択しません。そのため、特定のジョブについて DFSORT が、32760 バイトより大きいテープ出力ブロック・サイズを選択できるようにするには、JCL またはデータ・クラスで、それらのジョブ用として十分大きい BLKSZLIM 値 (たとえば、1 ギガバイト) が指定されていることを確認することが必要になります。

DFSORT に、システム決定最適ブロック・サイズを使用させない場合は、SDB=NO を指定します (インストール・オプションとしては推奨されていません)。

### LARGE

ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用することを指定します。OS/390 リリース 10 以上では、SDB=LARGE を指定すると、DFSORT は該当する場合、テープ出力データ・セットに 32760 バイトより大きいブロック・サイズを選択できます。大きいテープ・ブロック・サイズを使用すると、経過時間とテープ使用効率を向上させることができますが、結果として得られるテープ・データ・セットを使用するアプリケーションが、大きいブロック・サイズを処理できることを確認する必要があります。

DFSORT は以下のように、システム決定最適ブロック・サイズを選択します。

- DASD 出力データ・セットの場合、使用する装置の最適ブロック・サイズは、出力データ・セットの取得または導き出された RECFM および LRECL 属性に基づいて選択されます。DASD 出力データ・セットの最大ブロック・サイズは 32760 バイトです。
- テープ出力データ・セットの場合、最適ブロック・サイズは、出力データ・セットの取得または派生の RECFM および LRECL 属性に基づいて選択されます。215 ページの図 14 に示されているとおりです。

RECFM	BLKSIZE の設定値
F または FS	LRECL
FB または FBS	有効な BLKSZLIM に応じて、装置の最適ブロック・サイズより小か等しい、LRECL の可能な倍数の最大値。
V、D、VS または DS	LRECL + 4
VB、DB、VB または DBS	有効な BLKSZLIM に応じて、装置の最適ブロック・サイズ。

図 14. テープ出力データ・セットの *SDB=LARGE* ブロック・サイズ

DFSORT は、ブロック・サイズがゼロのとき、ほとんどの場合、出力データ・セット用にシステム決定最適ブロック・サイズを使用します。ただし、次のような条件では DFSORT は、システム決定ブロック・サイズを使用できません。

- 出力データ・セット・ブロック・サイズが、JFCB (DASD またはテープ) または形式 1 DSCB (DASD) またはテープ・ラベル (該当するとき、AL、SL、または NSL ラベルで DISP=MOD の場合のみ) で使用可能 (つまり、非ゼロ) である。
- MVS/DFP 3.1.0 以上でない。
- 出力が、スプール、ダミー、VSAM、または移動不能データ・セット、または HFS ファイルである。
- 出力データ・セットが、AL のラベル・タイプをもつテープ上にある。
- DFSORT のブロック・セット手法が選択されていない。

上記の場合、DFSORT は、指定されているブロック・サイズを使用するか、出力データ・セットの適切な (ただし、必ずしも最適ではない) ブロック・サイズを判別します。選択されるブロック・サイズは、32760 バイトに限定されます。

### YES

ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用するが、選択されるブロック・サイズを 32760 バイトの最大値に限定することを指定します。詳細は、*SDB=LARGE* の説明を参照してください。*SDB=LARGE* と *SDB=YES* の唯一の相違は、*SDB=LARGE* では、テープ出力データ・セットに 32760 バイトより大きいブロック・サイズを許可するのに対し、*SDB=YES* では許可されないという点です。

### INPUT

ブロック・サイズがゼロの場合に、DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用するが、入力ブロック・サイズが 32760 以下であれば、選択されるブロック・サイズを 32760 バイトの最大値に限定することを指定します。そのため、入力ブロック・サイズが 32760 バイトより大きい (テープ入力データ・セットの場合のみ可能) 場合は、*SDB=INPUT* は *SDB=LARGE* のように動作し、入力ブロック・サイズが 32760 バイトより小か等しい場合は、*SDB=YES* のように動作します。詳細は、*SDB=LARGE* および *SDB=YES* の説明を参照してください。

## OPTION 制御ステートメント

### NO

DFSORT が出力データ・セットにシステム決定最適ブロック・サイズを使用しないことを指定します。出力データ・セットのブロック・サイズがゼロである場合、DFSORT は、取得または派生した出力属性または入力属性に基づいて、出力データ・セットの適切な (必ずしも最適ではない) ブロック・サイズを選択します。SDB=NO は、選択されるブロック・サイズを 32760 バイトの最大値に制限します。

入力ブロック・サイズが 32760 バイトより大きい (テープ入力データ・セットの場合のみ可能) 場合は、SDB=NOT は SDB=YES のように動作します。詳細については、SDB=YES の説明を参照してください。

### 注:

1. OS/390 R9 およびそれ以前のリリースでは、SDB=LARGE と SDB=INPUT は SDB=YES のように動作します。
2. SDB=NO を指定すると、割り振り時や、DFSORT に制御が渡る前に出力データ・セットのブロック・サイズがセットされるその他の場合に、出力データ・セットのシステム決定最適ブロック・サイズが使用されません。
3. DFSORT がシステム決定最適ブロック・サイズを使用するとき、選択される出力データ・セット・ブロック・サイズは、前に選択されたブロック・サイズとは異なることがあります。特定の出力データ・セット・ブロック・サイズが必要なアプリケーションは、そのブロック・サイズを明示的に指定するように変更する必要があります。
4. SDB=YES の代わりに SDB および SDB=SMALL を使用できます。SDB=NO の代わりに NOSDB を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SIZE

FILSZ を参照してください。

### SKIPREC

▶▶—SKIPREC=z—◀◀

入力データ・セットの分類またはコピーを開始する前にスキップ (削除) したいレコード数 *z* を指定します。SKIPREC は、通常、前の DFSORT 実行で、その入力データ・セットの一部のみを処理した場合に使用します。

中間記憶装置の許容量を超える入力データ・セットをもつアプリケーションは、通常、異常終了します。ただし、テープ作業データ・セット分類の場合は、E16 でルーチンを使用し (361 ページの『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』で説明)、すでに読み込まれたレコードだけを分類するようにプログラムに指示できます。次に、分類されたレコードの数を示すメッセージを印刷します。後続の分類処理の実行で SKIPREC を使用して、分類済みのレコード

## OPTION 制御ステートメント

をバイパスし、残りのレコードだけを分類し、その後で別の実行からの出力を組み合わせて、アプリケーションを完了できます。

**z** スキップするレコードの数を指定します。

限界: 28 桁 (有効数字 15 桁)

### 注:

1. SKIPREC は、SORTIN から読み込まれたレコードだけに適用されます (E15 ルーチンから読み込まれたレコードには適用されません)。(9 ページの図 2 を参照してください。)
2. SKIPREC=0 が有効な場合は、SKIPREC は使用されません。
3. SKIPREC 使用の代替方法として、OUTFIL ステートメントの STARTREC パラメーターを使用できます。

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## SMF



SMF インストール・オプションを一時的に指定変更して、DFSORT SMF レコードを、DFSORT 導入とカスタマイズ リリース 14 に記されているように作成するかどうかを指定します。

### SHORT

DFSORT が正常実行用に短縮 SMF タイプ 16 を作成することを指定します。短縮 SMF レコードには、レコード長分配統計やデータ・セットのセクションは含まれていません。

### FULL

DFSORT が正常実行用に完全な SMF タイプ 16 を作成することを指定します。完全な SMF レコードには、短縮レコードと同じ情報が含まれおり、さらにレコード長分配およびデータ・セットのセクションも適切な形で含まれています。

### NO

DFSORT が、この実行用に SMF タイプ 16 レコードを作成しないことを指定します。

### 注:

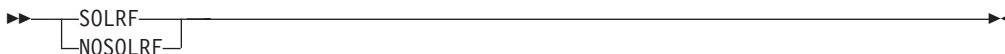
1. SMF は、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントに渡された場合のみ処理されます。
2. SMF=FULL は、可変長レコードのアプリケーションのパフォーマンスを低下させることがあります。

## OPTION 制御ステートメント

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SOLRF または NOSOLRF



SOLRF インストール・オプションを一時的に指定変更して、DFSORT が、SORTOUT LRECL が不明のときに、SORTOUT LRECL を再フォーマット設定されたレコード長に設定するかどうかを指定します。

#### SOLRF

DFSORT が、SORTOUT LRECL が指定されていないか使用可能でないときに、SORTOUT LRECL の再フォーマット設定されたレコード長を使用することを指定します。DFSORT は、SORTOUT LRECL に以下のいずれかを (リストされている順序で) 使用します。

1. JFCB、形式 1 DSCB、DFSMSrmm、ICETPEX、または磁気テープ・ラベルから使用可能な場合、SORTOUT LRECL。
2. RECORD ステートメントに指定されている場合、L3 の長さ。
3. OUTREC ステートメントが指定されている場合、OUTREC の長さ。
4. INREC ステートメントが指定されている場合、INREC の長さ。
5. RECORD ステートメントに指定されており、E15 ユーザー出口が存在する場合、L2 の長さ。
6. JFCB、形式 1 DSCB、DFSMSrmm、ICETPEX、または磁気テープ・ラベルから使用可能な場合、SORTIN または SORTINnn LRECL。
7. RECORD ステートメントの L1 の長さ。

#### NOSOLRF

DFSORT が、SORTOUT LRECL の再フォーマット設定されたレコード長を使用しないことを指定します。DFSORT は、SORTOUT LRECL に以下のいずれかを (リストされている順序で) 使用します。

1. JFCB、形式 1 DSCB、DFSMSrmm、ICETPEX、または磁気テープ・ラベルから使用可能な場合、SORTOUT LRECL。
2. E35 出口を指定する RECORD ステートメントに指定されており、OUTREC ステートメントまたは INREC ステートメントが存在する場合、L3 の長さ。
3. RECORD ステートメントに指定されており、E15 ユーザー出口が存在する場合、L2 の長さ。
4. JFCB、形式 1 DSCB、DFSMSrmm、ICETPEX、または磁気テープ・ラベルから使用可能な場合、SORTIN または SORTINnn LRECL。
5. RECORD ステートメントの L1 の長さ。



**注:**

1. 有効な SOLRF (IBM 提供のデフォルト) では、DFSORT は、適切なときに SORTOUT LRECL を INREC または OUTREC レコード長にセットします。これが通常は、INREC または OUTREC を使用するときに行いたい処置です。INREC または OUTREC が存在するときでも、SORTOUT LRECL の入力長さを DFSORT に使用させる場合は NOSOLRF を使用できます。ただし、これにより、再フォーマット設定されたレコードの埋め込みや切り捨て、または終了が起こることに注意してください。
2. SOLRF の代わりに CAOUTREC を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**SORTDD**

▶—SORTDD=cccc—▶

1 つのプログラム・ステップで複数回 DFSORT を動的に呼び出す場合、使用される DD 名に 4 文字の接頭部を指定します。次の DD 名の中の『SORT』が、この 4 文字で置き換えられます。つまり、SORTIN、SORTOUT、SORTINn、SORTINnn、SORTOFd、SORTOFdd、SORTWKd、SORTWKdd、および SORTCNTL です。これにより、DFSORT の各呼び出しに対して、異なる DD 名のセットを使用できます。

**cccc**

4 文字の接頭部を指定します。この 4 文字はすべて英数字または国別文字 (\$、#、または @) である必要があります。先頭の文字は英字で指定します。最初の 3 文字に SYS を指定できません。

たとえば、置換文字として ABC# を使用すると、DFSORT は、SORTIN、SORTCNTL、SORTWKdd、および SORTOUT の代わりに、ABC#IN、ABC#CNTL、ABC#WKdd、および ABC#OUT の DD ステートメントを使用します。

**注:**

1. SORTDD は、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡された場合のみ処理されます。
2. SORTIN=ddname と SORTDD=cccc の両方を指定すると、DFSORT 入力には ddname が使用されます。
3. SORTOUT=ddname と SORTDD=cccc の両方を指定すると、DFSORT 出力には ddname が使用されます。

デフォルト: SORT。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION 制御ステートメント

### SORTIN

▶▶—SORTIN=ddname—◀◀

SORTIN データ・セットに関連付けられる DD 名を指定します。これにより、1 つのプログラム・ステップで複数回 DFSORT を動的に呼び出し、それぞれの入力データ・セットごとに、異なる DD 名を渡すことができます。

DD 名には 1 文字から 8 文字までの名前を使用できますが、ジョブ・ステップ内で固有である必要があります。DFSORT 使用に予約されている DD 名 (たとえば SYSIN など) は 使用しないでください。

#### 注:

1. SORTIN は、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡される場合のみ処理されます。
2. SORTIN=ddname と SORTDD=cccc の両方を指定すると、入力ファイルには ddname が使用されます。SORTIN と SORTOUT に同じ DD 名を指定できません。
3. テープ作業データ・セット分類の場合に SORTIN を使用すると、DFSORT は終了します。

デフォルト: SORTIN。ただし、SORTDD=cccc が指定され、ccccIN がデフォルトである場合を除きます。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SORTOUT

▶▶—SORTOUT=ddname—◀◀

SORTOUT データ・セットに関連付けられる DD 名を指定します。これにより、1 つのプログラム・ステップで複数回 DFSORT を動的に呼び出し、それぞれの出力データ・セットに異なる DD 名を渡すことができます。

DD 名には 1 文字から 8 文字までの名前を使用できますが、ジョブ・ステップ内で固有である必要があります。DFSORT 使用に予約されている DD 名 (たとえば SYSIN など) は 使用しないでください。

#### 注:

1. SORTOUT は、拡張パラメーター・リストまたは DFSPARM の OPTION 制御ステートメントで渡される場合のみ処理されます。
2. SORTOUT=ddname と SORTDD=cccc の両方を指定すると、出力ファイルには ddname が使用されます。SORTIN と SORTOUT に同じ DD 名を指定できません。
3. 従来の組み合わせまたはテープ作業データ・セット分類の場合に SORTOUT を指定すると、DFSORT は終了します。

## OPTION 制御ステートメント

デフォルト: SORTOUT。ただし、SORTDD=cccc が指定され、ccccOUT がデフォルトである場合を除きます。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SPANINC



SPANINC インストール・オプションを一時的に指定変更して、可変スパン入力データ・セット内に不完全なスパン・レコードが 1 つ以上検出されたときの、DFSORT の処置を指定します。

#### RC0

DFSORT に対して、メッセージ ICE197I を (一度) 発行し、戻りコードを 0 に設定して、検出した不完全なスパン・レコードをすべて除去するよう指定します。有効なレコードはリカバリーされます。

#### RC4

DFSORT に対して、メッセージ ICE197I を (一度) 発行し、戻りコードを 4 に設定して、検出した不完全なスパン・レコードをすべて除去するよう指定します。有効なレコードはリカバリーされます。

#### RC16

DFSORT に対して、不完全なスパン・レコードを検出したら、メッセージ ICE204A を発行して終了し、戻りコード 16 を戻すよう指定します。

#### 注:

1. 不完全なスパン・レコードに対して設定される戻りコード 0 または 4 は、他の理由によるもっと大きい値の戻りコードで上書きされることがあります。
2. スパン・レコードを正しくアセンブルできない場合 (セグメント長が 4 バイト未満の場合など)、DFSORT は ICE141A を発行して、戻りコード 16 で終了します。このような場合は、SPANINC 値は効力がありません。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### STOPAFT



分類またはコピーのために受け入れたいレコードの最大数 (n) (つまり、SORTIN から読み取られるか、E15 により挿入され、しかも SKIPREC、E15、

## OPTION 制御ステートメント

または INCLUDE/OMIT ステートメントにより削除されないレコードの数) を指定します。n 個のレコードが受け入れられると、それ以上は SORTIN から読み込まれません。E15 は、あたかも EOF になった場合のように、戻りコード 8 が戻されるまで、入力が続けますが、それ以上のレコードは挿入されません。n 個のレコードが受け入れられる前にファイルの終わりになると、それまでに受け入れられたレコードだけが分類またはコピーされます。

**n** 受け入れたいレコードの最大数を指定します。

限界: 28 桁 (有効数字 15 桁)

デフォルト: なし。オプションです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### SZERO または NOSZERO



SZERO インストール・オプションを一時的に指定変更して、DFSORT が照合、比較、編集および変換、最小値および最大値で、数字の -0 および +0 値を、符号付きとして (つまり、異なる値として) 処理するか、符号なしとして (つまり、同じ値として) 処理するかを指定します。このオプションで影響を受けるのは、INCLUDE、INREC、MERGE、OMIT、OUTFIL、OUTREC、および SORT の各 DFSORT 制御ステートメントです。

#### SZERO

DFSORT が数値ゼロを符号付きとして扱うことを指定します。-0 および +0 は、異なる値として扱われます。つまり、-0 は負の値として、+0 は正の値として扱われます。SZERO は次のように、数値の DFSORT 処理に影響します。

- SORT フィールドと MERGE フィールドの照合の場合、-0 は、昇順で +0 の前を照合し、降順で +0 の後を照合します。
- INCLUDE、OMIT、OUTFIL INCLUDE および OMIT フィールド、および定数の比較の場合、-0 は、+0 より小として比較されます。
- INREC、OUTREC、および OUTFIL OUTREC フィールドの編集および変換の場合、-0 は負として、+0 は正として扱われます。
- OUTFIL TRAILERx フィールドの最小値および最大値の場合、-0 は負として、+0 は正として扱われます。

#### NOSZERO

DFSORT が数値ゼロを符号なしとして扱うことを指定します。-0 および +0 は、同じ値として扱われます。つまり、-0 と +0 は両方とも正の値として扱われます。NOSZERO は次のように、数値の DFSORT 処理に影響します。

- SORT フィールドと MERGE フィールドの照合の場合、-0 は +0 と等しく照合されます。

## OPTION 制御ステートメント

- INCLUDE、OMIT、OUTFIL INCLUDE および OMIT フィールド、および定数の比較の場合、-0 は、+0 と等しく比較されます。
- INREC、OUTREC、および OUTFIL OUTREC フィールドの編集および変換の場合、-0 と +0 は正として扱われます。
- OUTFIL TRAILERx フィールドの最小値および最大値の場合、-0 と +0 は正として扱われます。

**注:** OPTION SZERO または OPTION NOSZERO は、OUTFIL INCLUDE=(...) または OUTFIL OMIT=(...) について無視されます (OPTION ステートメントが、OUTFIL ステートメントの後に「検出」された場合)。これを回避するには、SZERO または NOSZERO を、EXEC/DFSPARM PARM オプションとして指定するか、同じソースの OUTFIL ステートメントの前の OPTION ステートメントで指定します。

```
//SYSIN DD *  
OPTION NOSZERO,COPY  
OUTFIL INCLUDE=(...)  
/*
```

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## TRUNC



TRUNC インストール・オプションを一時的に指定変更して、DFSORT が LRECL 短縮を許可する場合に、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいときの、DFSORT の処置を指定します。

### RC0

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいときは、メッセージ ICE171I を発行し、戻りコードを 0 に設定して処理を続行するよう指定します。

### RC4

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より小さいときは、メッセージ ICE171I を発行し、戻りコードを 4 に設定して処理を続行するよう指定します。

### RC16

DFSORT に対して、SORTOUT LRECL が SORTIN/SORTINnn LRECL より大きいときは、メッセージ ICE196A を発行して終了し、戻りコード 16 を戻すよう指定します。

## OPTION 制御ステートメント

注:

1. LRECL 短縮に対して設定される戻りコード 0 または 4 は、他の理由によるもっと大きい値の戻りコードで上書きされることがあります。
2. ICEGENER アプリケーションの場合は、GNTRUNC 値が使用され、TRUNC 値は無視されます。
3. ある条件下の LRECL 短縮 (テープ作業データ・セットの分類など) では、DFSORT は ICE043A を発行して、戻りコード 16 で終了します。このような場合は、TRUNC 値は効力がありません。
4. DFSORT は、次の場合は LRECL 短縮を検査しません。
  - a. SORTIN DD (分類 / コピー)、SORTINnn DD (組み合わせ)、または SORTOUT DD が指定されていないとき。
  - b. SORTIN DD (分類 / コピー)、SORTINnn DD (組み合わせ)、または SORTOUT DD に VSAM データ・セットが指定されているとき。
5. DFSORT は、LRECL 短縮に関して OUTFIL データ・セットを検査しません。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### USEWKDD

▶—USEWKDD—▶

DYNAUTO=IGNWKDD オプションを一時的に指定変更して、SORTWKdd DD ステートメントが存在している場合でも、動的作業データ・セットを使用することを指定します。このオプションにより、JCL SORTWKdd を割り振り解除するのではなく、使用できます。

注: USEWKDD は、拡張パラメーター・リストまたは DFSPARM の、OPTION 制御ステートメントで渡される場合のみ処理されます。

デフォルト: なし。任意指定です。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VERIFY または NOVERIFY

▶—VERIFY—▶  
└—NOVERIFY—┘

VERIFY インストール・オプションを一時的に変更して、最終出力レコードのシーケンス検査を行なう必要があるかどうかを指定します。

**VERIFY**

シーケンス検査が行なわれることを指定します。

**NOVERIFY**

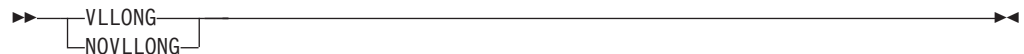
シーケンス検査が行なわれないことを指定します。

**注:**

1. VERIFY を使用すると、パフォーマンスを低下させることがあります。
2. VERIFY の代わりに SEQ=YES を使用し、NOVERIFY の代わりに SEQ=NO を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**VLLONG または NOVLLONG**

VLLONG インストール・オプションを一時的に指定変更して、DFSORT が「長い」可変長出力レコードを切り捨てるかどうかを指定します。長い出力レコードとは、書き込み先の SORTOUT または OUTFIL データ・セットの LRECL より大きい長さをもつレコードのことです。

VLLONG は、固定長出力レコード処理については意味がありません。

**VLLONG**

DFSORT が長い可変長出力レコードを、SORTOUT または OUTFIL データ・セットの LRECL に切り捨てることを指定します。

**NOVLLONG**

長い可変長出力レコードが見つかった場合に、DFSORT が終了することを指定します。

**注:**

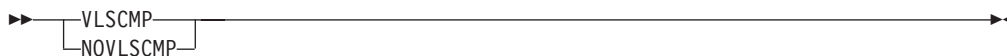
1. VLLONG の使用は、長い可変長出力レコードの終わりにあるデータを、DFSORT アプリケーションのために切り捨てたい場合だけに限定してください。VLLONG を不適切に使用すると、データの望ましくない消失が発生することがあります。
2. VLLONG を使用すると、長い OUTFIL データ・レコードを切り捨てることができますが、長い OUTFIL ヘッダー・レコードやトレーラー・レコードには効果がありません。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION 制御ステートメント

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VLSCMP または NOVLSCMP



VLSCMP インストール・オプションを一時的に指定変更して、DFSORT が「短い」可変長 INCLUDE/OMIT 比較フィールドを 2 進ゼロで埋め込むかどうかを指定します。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張されるようなフィールドを指します。VLSCMP および NOVLSCMP は、INCLUDE および OMIT ステートメントと、OUTFIL ステートメントの INCLUDE および OMIT パラメーターに適用されます。

比較フィールドは、テストのために一時的に埋め込まれるだけであり、出力のために実際に変更されるわけではありません。

VLSCMP は、固定長レコード処理には意味がありません。

VLSCMP/NOVLSCMP および VLSHRT/NOVLSHRT の設定は、以下の階層で、短い INCLUDE/OMIT フィールドの 3 つのレベルの処理を提供します。

1. VLSCMP は、いくつかのフィールドが短い場合でも、すべての INCLUDE/OMIT 比較を実行させることができます。短いフィールドは 2 進ゼロで埋め込まれるので、短いフィールドを含む比較は偽です (下記のように、2 進ゼロに対するテストが関係する場合は除く)。短くないフィールドを含む比較は、真のことも、偽のこともあります。
2. NOVLSCMP および VLSHRT は、短いフィールドがある場合は、INCLUDE/OMIT 論理式全体を偽として扱います。そのため、比較に短いフィールドが含まれている場合、短くないフィールドを含む比較は無視されません。
3. いずれかのフィールドが短い場合、NOVLSCMP および NOVLSHRT の結果は終了となります。

これがどのように動作するかを理解できるように、次の INCLUDE ステートメントを使用するものとします。

```
INCLUDE COND=(6,1,CH,EQ,C'1',OR,70,2,CH,EQ,C'T1')
```

可変長入力レコード長が 71 バイトより短い場合、70 ~ 71 バイト目のフィールドは短く、以下のことが起こります。

- VLSCMP では、レコードは、入力レコードの 6 バイト目が C'1' であれば、組み込まれ、6 バイト目が C'1' でなければ、切り捨てられます。C'T1' に等しい 70 ~ 71 バイト目の比較は偽です。これは、70 ~ 71 バイト目に、X'hh00' (70 バイトのレコード長の場合)、または X'0000' (レコード長が 70 バイトより短い場合) が含まれているためです。短くないフィールドを含む比較は、短いフィールドが存在しても、実行されます。



## OPTION 制御ステートメント

- NOVLSCMP および VLSHRT では、短いフィールドにより論理式全体が偽になるため、レコードは切り捨てられます。短くないフィールドを含む比較は、短いフィールドが存在するため実行されません。
- NOVLSCMP および NOVLSHRT では、DFSORT は、短いフィールドにより終了が起こるため、終了します。

一般に、短いフィールドを含む比較は、VLSCMP では偽です。ただし、2 進ゼロ値が比較に関係がある場合、埋め込みのための 2 進ゼロの使用により、比較が真になる可能性があります。たとえば、次の INCLUDE ステートメントを使用するものとします。

```
INCLUDE COND=(21,2,CH,EQ,C'JX',OR,
              (55,2,CH,EQ,58,2,CH,AND,
              70,1,BI,LT,X'08'))
```

可変長入力レコード長が 71 バイトより短い場合、70 バイト目のフィールドは短く、X'00' で埋め込まれます。これにより、X'08' より小さい 70 バイト目の比較は、70 バイト目が短いフィールドであるため、おそらく関係がないときでも、真になります。

同様に、可変長入力レコード長が 55 バイトより短い場合、55 ~ 56 および 58 ~ 59 バイト目のフィールドは短く、それぞれが X'0000' に埋め込まれ、70 バイト目のフィールドは短く、X'00' に埋め込まれます。これにより、58 ~ 59 に等しい 55 ~ 56 バイト目の比較は真になり、X'08' よりより小さい 70 バイトの比較は、3 つのフィールドすべてが短く、しかもおそらく関係がないときでも、真になります。

短いフィールドの 2 進ゼロによる埋め込みにより、不要な真の比較が発生する場合は、以下のような、INCLUDE/OMIT 論理式にレコード長の適切なチェックを追加することで、必要な結果を得ることができます。

```
INCLUDE COND=(21,2,CH,EQ,C'JX',OR,
              (1,2,BI,GE,X'0046',AND,
              55,2,CH,EQ,58,2,CH,AND,
              70,1,BI,LT,X'08'))
```

これで、55 ~ 56、58 ~ 59、および 70 バイト目を含む比較は、70 バイト (X'0046') またはそれより長いレコードについてだけ真になります。そのため、短いフィールドを含む関係のない比較は除去されます。

短い比較フィールドは VLSCMP が有効なときにゼロで埋め込まれることに注意し、それを可能にするため、あるいは利用するためにも、INCLUDE/OMIT 論理式をコーディングしてください。

### VLSCMP

短い可変長比較フィールドが 2 進ゼロで埋め込まれることを指定します。

### NOVLSCMP

短い可変長比較フィールドが埋め込まれないことを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION 制御ステートメント

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VLSHRT または NOVLSHRT



VLSHRT インストール・オプションを一時的に指定変更して、「短い」可変長 SORT/MERGE 制御フィールド、INCLUDE/OMIT 比較フィールド、または SUM 要約フィールドが検出された場合に、DFSORT が処理を続行するかどうかを指定します。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張されるようなフィールドを指します。VLSHRT は、SORT、MERGE、INCLUDE、OMIT および SUM ステートメントと、OUTFIL ステートメントの INCLUDE および OMIT パラメーターに適用されます。

VLSHRT は、固定長レコード処理には意味がありません。

DFSORT が短い INCLUDE/OMIT 比較フィールドを処理する方法は、VLSCMP/NOVLSCMP および VLSHRT/NOVLSHRT の設定により異なります。詳細については、VLSCMP および NOVLSCMP オプションの説明を参照してください。

#### VLSHRT

短い制御フィールド、比較フィールド、または要約フィールドが検出された場合に、DFSORT が処理を続行することを指定します。

#### NOVLSHRT

短い制御フィールド、比較フィールド、または要約フィールドが検出された場合に、DFSORT が終了することを指定します。

#### 注:

1. INREC、または OUTREC ステートメントが指定されている場合、もしくは EFS01 または EFS02 ルーチンが用意されている場合、あるいはロケール処理が SORT または MERGE フィールド用に使用されている場合は、VLSHRT は使用されません。これらの状態のいずれでも、VLSCMP の使用は妨げられないことに注意してください。
2. OUTREC ステートメントとは異なり、OUTFIL ステートメントの OUTREC パラメーターは、NOVLSHRT を強制しません。したがって、OUTFIL とともに VLSHRT を使用して、INCLUDE または OMIT パラメーターを用いてレコードを除去し、残りのレコードを OUTREC パラメーターを用いて再フォーマット設定することができます。短い OUTFIL OUTREC フィールドが検出された場合は、OUTFIL の VLFILL=byte パラメーターが指定されていない限り、(VLSHRT が有効であっても) DFSORT は終了します。
3. VLSHRT が有効で、ブロック・セットが選択された場合は、次のようになります。
  - DFSORT は SORT または MERGE 制御フィールドを、2 進ゼロで埋め込み、互いに長さは異なっても内容の等しい制御フィールドをもつレコードの順序を予測可能にします。制御フィールドは、照合のために一時的

に埋め込まれるだけであり、出力のために実際に変更されるわけではありません。埋め込みは、ワークスペースの必要量を増加させることがあります。

- 短い SUM 要約フィールドを持つレコードは、合計の対象にはなりません。つまり、合計しようとする 2 つのレコードのうち 1 つに短い SUM フィールドがある場合、それらのレコードは合計されず、またどちらも削除されません。

4. VLSHRT が有効で、ブロック・セットが選択されなかった場合は、次のようになります。

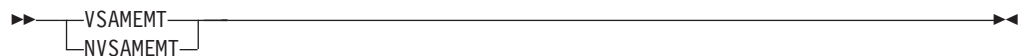
- 最初の (大) SORT または MERGE 制御フィールドの 1 桁目のバイトが、レコード内に含まれていない場合、DFSORT は終了します。
- DFSORT は、短い SORT または MERGE 制御フィールドを埋め込みません。したがって、長さが異なる、内容の等しい制御フィールドをもつレコードがどのように順序付けられるか予測できません。
- SORT または MERGE 制御フィールドの数と位置によりは、VLSHRT が使用されない場合もあります。
- EQUALS は使用されません。

注: SORTDIAG DD ステートメントを使用すると、メッセージ ICE800I を出力させて、ブロック・セットが使用できない理由を示すコードを表示できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VSAMEMT または NVSAMEMT



VSAMEMT インストール・オプションを一時的に指定変更して、DFSORT が空の VSAM 入力データ・セットを受け入れるかどうかを指定します。

#### VSAMEMT

DFSORT が空の VSAM 入力データ・セットを受け入れ、ゼロのレコードをもつとして処理することを指定します。

#### NVSAMEMT

空の VSAM 入力データ・セットが検出された場合に、DFSORT が終了することを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION 制御ステートメント

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### VSAMIO または NOVSAMIO



VSAMIO インストール・オプションを一時的に指定変更して、DFSORT が、REUSE で定義された VSAM データ・セットをインプレースで分類するかどうかを指定します。

#### VSAMIO

以下のすべての条件に合致したとき、DFSORT が入力と出力に同じ VSAM データ・セットを使用できることを指定します。

- アプリケーションが分類である。
- RESET が有効。
- VSAM データ・セットが REUSE で定義された。

これらの条件により、VSAM データ・セットは確実に、出力では NEW として処理され、分類済みの入力レコードを含むことになります。つまり、インプレースで分類されます。

DFSORT は、同じ VSAM データ・セットが入力と出力に指定されていて、上記の条件に合致しない場合、終了します。

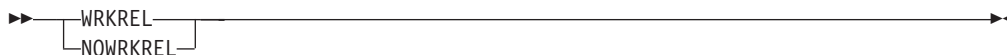
#### NOVSAMIO

同じ VSAM データ・セットが入力と出力に使用される場合、DFSORT が終了することを指定します。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### WRKREL または NOWRKREL



WRKREL インストール・オプションを一時的に指定変更して、未使用の一時 SORTWKdd データ・セット空間を解放するかどうかを指定します。

#### WRKREL

使用されなくなったスペースを解放することを指定します。

#### NOWRKREL

使用しないスペースを解放しないことを指定します。

**注:**

1. すでにいくつかのボリュームが SORTWKdd データ・セット専用になっていて、使用していない一時スペースを解放したくない場合は、NOWRKREL を指定する必要があります。
2. WRKREL が有効な場合、DFSORT は終了の直前に SORTWKdd データ・セット用のスペースを解放します。分類アプリケーションに使用した SORTWKdd データ・セットのスペースだけが解放されます。
3. NOWRKREL の代わりに RLS=0 を使用できます。WRKREL の代わりに RLS=n (n は、0 より大) を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**WRKSEC または NOWRKSEC**



WRKSEC インストール・オプションを一時的に指定変更して、DFSORT が一時 JCL SORTWKdd データ・セット用に自動 2 次割り振りを使用するかどうかを指定します。

**WRKSEC**

一時 JCL SORTWKdd データ・セットに自動 2 次割り振りを使用し、また 1 次割り振りの 25 % を 2 次割り振りとして使用するよう指定します。

**NOWRKSEC**

一時 JCL SORTWKdd データ・セットに自動 2 次割り振りを使用しないことを指定します。

**注:** NOWRKSEC の代わりに SEC=0 を使用できます。WRKSEC の代わりに SEC=n (n は、0 より大) を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**Y2PAST**



## OPTION 制御ステートメント

Y2PAST インストール・オプションを一時的に指定変更して、スライド (s) または固定 (f) の「世紀」ウィンドウを指定します。「世紀」ウィンドウを、DFSORT の Y2 形式とともに使用すると、2 桁年のデータ値が 4 桁年のデータ値として、正しく解釈されます。

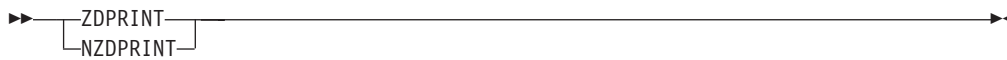
**s** スライド「世紀」ウィンドウの開始年を設定するために、DFSORT が現在の年から減算する年数を指定します。Y2PAST の値を現在の年から減算するため、現在の年が変わると「世紀」ウィンドウもスライドします。たとえば、Y2PAST=81 とすると、「世紀」ウィンドウは、1996 年には 1915-2014、1997 年には 1916-2015 となります。s の値は 0 から 100 の範囲で指定します。

**f** 固定「世紀」ウィンドウの開始年を指定します。たとえば、Y2PAST=1 とすると、1962-2061 の「世紀」ウィンドウがセットされます。f の値は 1000 から 3000 の範囲で指定します。

注: Y2PAST=value の代わりに CENTURY=value および CENTWIN=value を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## ZDPRINT または NZDPRINT



ZDPRINT インストール・オプションを一時的に指定変更して、集計の結果生じた正のゾーン 10 進数 (ZD) フィールドを、印刷可能な数字に変換しなければならないかどうかを指定します (つまり、最後の桁のゾーンを 16 進数の C から 16 進数の F へ変更するかどうかを指定します)。ZDPRINT および NZDPRINT の使用についての詳細は、355 ページの『SUM 制御ステートメント』を参照してください。

### ZDPRINT

正の ZD 集計結果を印刷可能な数字に変換することを意味します。たとえば、16 進数 F3F2C5 (32E と印刷) を F3F2F5 (325 と印刷) に変更します。

### NZDPRINT

正の ZD 集計結果を印刷可能な数字に変換しないことを意味します。

注: ZDPRINT の代わりに ZDPRINT=YES を使用できます。NZDPRINT の代わりに ZDPRINT=NO を使用できます。

デフォルト: 通常は、インストール時のデフォルトです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## OPTION ステートメント・オプションの別名

互換性の点から、下記のリストした別名を使用して、以下の OPTION ステートメントのオプションを指定できます。完全な詳細については、指示された OPTION ステートメント・オプションを参照してください。

表 31. OPTION ステートメント・オプションの別名

別名	OPTION ステートメント・オプション
CAOUTREC	SOLRF
CENTURY=value	Y2PAST=value
CENTWIN=value	Y2PAST=value
CHKPT	CKPT
CORE=value	MAINSIZE=value
HIPRLIM=value	HIPRMAX=value
L5=value	AVGRLLEN=value
NOSDB	SDB=NO
PRINT=value	MSGPRT=value
RLS=n	WRKREL
RLS=0	NOWRKREL
SDB	SDB=YES
SDB=SMALL	SDB=YES
SEC=n	WRKSEC
SEC=0	NOWRKSEC
SEQ=YES	VERIFY
SEQ=NO	NOVERIFY
ZDPRINT=YES	ZDPRINT
ZDPRINT=NO	NZDPRINT

## DFSORT オプションまたは COPY の指定 - 例

### 例 1

```
SORT FIELDS=(1,20,CH,A)
OPTION SIZE=50000,SKIPREC=5,EQUALS,DYNALOC
```

#### FIELDS

制御フィールドは、入力データ・セット内の各レコードの 1 バイト目から始まり、20 バイトの長さで文字データを含み、昇順に分類されます。

#### SIZE

分類するデータ・セットには、50000 レコードが含まれています。

#### SKIPREC

入力データ・セットの処理を開始する前に、5 つのレコードをスキップ (削除) します。

#### EQUALS

照合が一致するレコード順序が、入力から出力まで保持されます。

## OPTION 制御ステートメント

### DYNALLOC

2 つのデータ・セット (デフォルト) は、SYSDA (デフォルト) 上に割り振られます。データ・セット上のスペースは、有効な SIZE 値を使用して計算されません。

### 例 2

```
SORT  FIELDS=(1,2,CH,A),CKPT
OPTION EQUALS,NOCHALT,NOVERIFY,CHECK
```

### FIELDS

制御フィールドは、入力データ・セット内の各レコードの 2 バイト目から始まり、2 バイトの長さで文字データを含み、昇順に分類されます。

### CKPT

DFSORT は、実行中にチェックポイントを取ります。

**注:** ブロック・セット手法が使用されると、CKPT は無視されます。チェックポイントが必要な場合は、NOBLKSET オプションを指定するか、ICEMAC インストール・マクロで IGNCKPT=NO を指定するかして、ブロック・セット手法をバイパスする必要があります。ただし、OUTFIL のような機能 (これらはブロック・セット手法によりだけサポートされる) は、チェックポイント / 再始動機能の使用中には使用できません。

### EQUALS

照合が一致するレコード順序が、入力から出力まで保持されます。

### NOCHALT

AQ フィールドだけが、ALTSEQ 変換テーブルで変換されます。インストール時に CHALT=YES が指定されていた場合、NOCHALT は一時的にこれを変更します。

### NOVERIFY

最終出力レコードには、シーケンス検査が行なわれません。

### CHECK

プログラム処理の終了時にレコード数が検査されます。

### 例 3

```
OPTION FILSZ=50,SKIPREC=5,DYNALLOC=3390
SORT  FIELDS=(1,2,CH,A),SKIPREC=1,SIZE=200,DYNALLOC=(3380,5)
```

この例では、OPTION 制御ステートメントで指定したパラメーターが、SORT 制御ステートメントで指定したパラメーターをどのように変更するかを (2 つのステートメントの順序にかかわらず) 示しています。

### FILSZ

DFSORT は、入力データ・セットに 50 のレコードが入っているものと予測します。(FILSZ と SIZE では意味が異なること、また OPTION に FILSZ を指定すると SIZE の代わりに FILSZ が使用されることに注意してください。)



**SKIPREC**

DFSORT は入力ファイルの始めから 5 つのレコードをスキップします。(SORT ステートメントの SKIPREC=1 は無視されます。)

**DYNALLOC**

DFSORT は、2 つの作業データ・セット (デフォルト) を IBM 3390 上に割り振ります。

**FIELDS**

制御フィールドは、入力データ・セット内の各レコードの 2 バイト目から始まり、2 バイトの長さで文字データを含み、昇順に分類されます。

**例 4**

```
OPTION NOBLKSET
```

**NOBLKSET**

DFSORT は、分類または組み合わせにブロック・セット手法を使用しません。

**例 5**

```
OPTION STOPAFT=100,COBEXIT=COB2
```

**STOPAFT**

DFSORT は、分類またはコピー処理の前に 100 レコードを受け入れます。

**COBEXIT**

COBOL E15 および E35 ルーチンを、VS COBOL II ライブラリーまたは言語環境プログラム・ライブラリーを用いて実行します。

**例 6**

```
OPTION RESINV=32000,MSGPRT=NONE,
MSGDDN=SORTMSG, SORTDD=ABCD, SORTIN=MYINPUT,
SORTOUT=MYOUTPUT, NOLIST
```

この例では、パラメーター RESINV、MSGPRT、MSGDDN、SORTDD、SORTIN、SORTOUT、および NOLIST と、SYSIN データ・セットまたは SORTCNTL データ・セットから読み取られた OPTION ステートメントにこれらのパラメーターが指定された場合に取りられる処置を示しています。パラメーターは認識されますが、使用されません。

**RESINV**

32000 バイトの記憶域がユーザー用に予約されます。

**MSGPRT=NONE**

このキーワードは無視され、インストール時に指定されたデフォルトに従ってメッセージが印刷されます。

**MSGDDN=SORTMSG**

このキーワードは無視され、すべてのメッセージが SYSOUT データ・セットに書き出されます。

## OPTION 制御ステートメント

### **SORTDD=ABCD**

このキーワードは無視され、標準接頭部の SORT が使用されます。

### **SORTIN=MYINPUT**

このキーワードは無視され、入力データ・セットの参照には DD 名 SORTIN が使用されます。

### **SORTOUT=MYOUTPUT**

このキーワードは無視され、出力データ・セットの参照には DD 名 SORTOUT が使用されます。

### **NOLIST**

このキーワードは無視され、インストール時に指定したデフォルトに従って制御ステートメントが印刷されます。

## 例 7

```
OPTION RESINV=32000,MSGPRT=CRITICAL
MSGDDN=SORTMSGs,SORTDD=ABCD,SORTIN=MYINPUT,
SORTOUT=MYOUTPUT,NOLIST
```

この例では、キーワード RESINV、MSGPRT、MSGDDN、SORTDD、SORTIN、SORTOUT、および NOLIST と、DFSPARM で渡された OPTION ステートメントにこれらのキーワードが指定された場合に取られる処置を示しています。これらのオプションは、拡張パラメーター・リストで渡すこともできますが、継続行のない連続する 1 つのステートメントとしてコーディングする必要があります。

### **RESINV**

32000 バイトの記憶域がユーザー用に予約されます。

### **MSGPRT=CRITICAL**

重大メッセージだけがメッセージ・データ・セットに印刷されます。

### **MSGDDN=SORTMSGs**

メッセージは SORTMSGs データ・セットに書き出されます。

### **SORTDD=ABCD**

SORT はすべての分類名の接頭部として ABCD を使用します。

### **SORTIN=MYINPUT**

入力データ・セットの参照には、DD 名 MYINPUT が使用されます。

### **SORTOUT=MYOUTPUT**

出力データ・セットの参照には、DD 名 MYOUTPUT が使用されます。

### **NOLIST**

制御ステートメントは印刷されません。

## 例 8

```
SORT FIELDS=(3,4,CH,A)
OPTION COPY,SKIPREC=10,CKPT
MODS E15=(E15,1024,MODLIB),E35=(E35,1024,MODLIB)
```

**SORT**

COPY オプションが指定されているため、SORT ステートメントは無視されま  
す。

**COPY**

コピー処理は必ずレコード単位で行われます。したがって、レコードが  
SORTIN から読み込まれるたびに、E15 出口に渡され、E35 出口に渡され、そ  
して SORTOUT に書き込まれます。(この処理方法と分類処理方法を対比して  
みてください。分類処理ではすべてのレコードが SORTIN から読み込まれて  
E15 出口に渡されてから、任意のレコードが E35 出口に渡されて SORTOUT  
に書き込まれます。)

**SKIPREC**

コピー処理の開始前に 10 レコードをスキップします。

**CKPT**

コピー・アプリケーションにはチェックポイント・オプションは使用されませ  
ん。

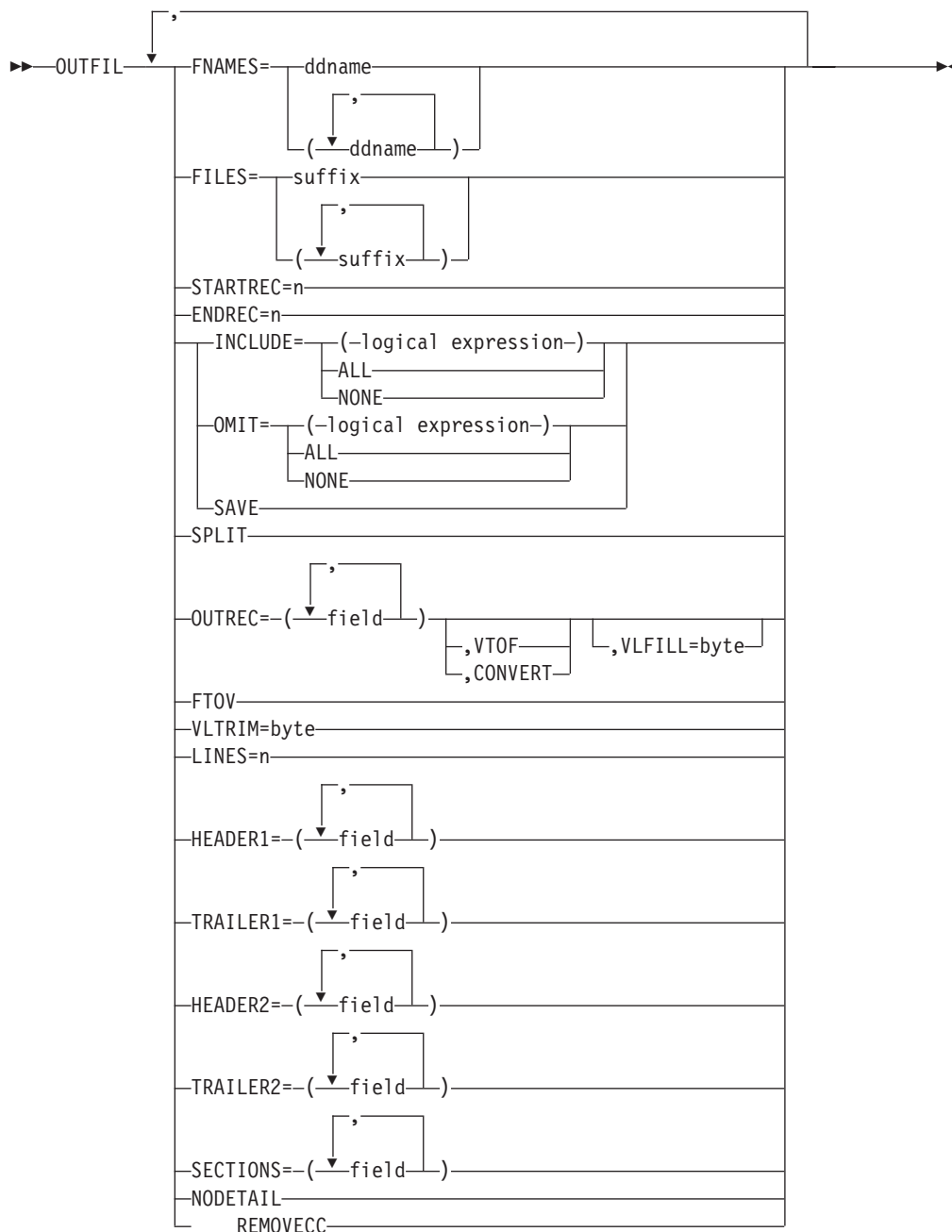
**例 9**

```
SORT  FIELDS=(5,4,CH,A)
SUM  FIELDS=(12,5,ZD,25,6,ZD)
OPTION ZDPRINT
```

**ZDPRINT**

DFSORT は最後の桁に F 符号を使用するため、合計された正の ZD 値は印刷  
可能です。

## OUTFIL 制御ステートメント



OUTFIL 制御ステートメントを使用することにより、1 つ以上の入力データ・セットに対する単一パスから、分類、コピー、または組み合わせ処理のために 1 つ以上の出力データ・セットを作成できます。複数の OUTFIL ステートメントを使用して、その各ステートメントで 1 つ以上の出力データ・セットに対して OUTFIL 処理を実行するように指定できます。他のすべての処理が終了した後（つまり、出口、オプション、およびその他の制御ステートメントを処理した後）、OUTFIL 処理が開始されます。OUTFIL ステートメントは、次のような様々な出力データ・セットのタスクをサポートします。

## OUTFIL 制御ステートメント

- 1 つ以上の入力データ・セットに対する単一のパスを使用することにより、未編集のレコードまたは編集済みのレコードを含む複数の出力データ・セットを作成する。
- 1 つ以上の入力データ・セットに対する単一のパスを使用して、異なる範囲またはサブセットのレコードを含む複数の出力データ・セットを作成する。さらに、どのサブセットとしても選択されないレコードは、別の出力データ・セットに保管できる。
- 可変長レコード・データ・セットから固定長レコード・データ・セットへ変換する。
- 固定長レコード・データ・セットから可変長レコード・データ・セットへ変換する。
- 再フォーマット設定されたレコードの入力フィールドの前、入力フィールド間、および入力フィールドの後に、ブランク、ゼロ、ストリング、現在日付、現在時刻およびシーケンス番号を挿入する。
- 16 進数表示、EBCDIC 文字の小文字から大文字へ、または大文字から小文字への変換、ALTSEQ 変換テーブルを使用した文字変換、ある形式から別の形式への数値変換などの、高度な変換機能。
- 長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、数値フィールドの表示方法を制御するなどの、高度な編集機能。  
一般に使用されている数値編集パターンとして 27 個の事前定義編集マスクが用意されており、世界中で使用されている数値表記の大部分を網羅しています。さらに、ユーザー定義の編集マスクにより、実際上無制限の数の数値編集パターンが利用できます。
- SMF 日付値および時間値の、より有効なフォームへの変換。
- 指定された固定またはスライドの「世紀」ウィンドウを使用した、2 桁年の日付の各種形式から 4 桁年の日付の各種形式への変換。
- 入力としての文字、16 進数、またはビット・ストリングに基づく、参照テーブルからの出力用の文字または 16 進数ストリングの選択 (つまり、参照と変更)。
- 詳細な 3 レベル (報告書、ページ、およびセクション) の報告書を提供する。この報告書には、ユーザーが指定できる多くの報告書要素 (たとえば、現在の日付、現在の時間、ページ番号、文字ストリング、およびブランク行) や、入力レコードから取り出すことができる多くの報告書要素 (たとえば、文字フィールド、編集された数値入力フィールド、レコード数、および数値入力フィールドの編集済み合計、最大、最小、および平均) が含まれています。
- 各入力レコードから、ブランクの出力レコードを間に挿入するか、あるいは挿入せずに、複数の出力レコードを作成する。

OUTFIL のパラメーターは、主要目的により次のようにグループ分けされます。

- **FNAMES** および **FILES** は、作成する **OUTFIL データ・セット** の DD 名を指定します。作成する各 **OUTFIL データ・セット** は、**OUTFIL ステートメント** の **FNAMES** または **FILES** を使用して具体的に識別する必要があります。一方、**SORTOUT データ・セット** は、デフォルトの DD ステートメントが存在する場合は、そのデフォルトにより作成されます。「**SORTOUT データ・セット**」とい

## OUTFIL 制御ステートメント

用語は、単一の非 OUTFIL 出力データ・セットのことを意味しますが、実際には、SORTOUT DD 名は、明示指定またはデフォルトにより、OUTFIL データ・セットで使用できます。

SORTOUT が明示指定により (たとえば、FILES=OUT を用いて)、あるいはデフォルト (FILES または FNAMES のない OUTFIL) により OUTFIL DD 名として識別される場合は、SORTOUT DD 名に関連するデータ・セットは、SORTOUT データ・セットとしてではなく OUTFIL データ・セットとして処理されます。

OUTFIL データ・セットの特性と要件は SORTOUT データ・セットの場合と似ていますが、処理の方法が異なります。主な違いとしては、OUTFIL データ・セットでは E39 出口ルーチンは実行されず、また OUTFIL 処理では固定形式 OUTFIL レコードを埋め込むために LRECL 値を使用することができません。(DFSORT は、RECFM、LRECL、および BLKSIZE が指定されていないかまたは使用できないような OUTFIL データ・セットについて、それぞれ適切な属性を自動的に判別して設定します。)

単一の DFSORT アプリケーションの場合、OUTFIL データ・セットは、VSAM と非 VSAM、テープと DASD、などを混在して使用できます。特定の OUTFIL ステートメントで指定されたデータ・セットは、すべて同様の方法で処理されるため、**OUTFIL グループ**と見なされます。(つまり、1 つの OUTFIL ステートメントで同じオペランドを指定することにより、それらのオペランドを使用する OUTFIL データ・セットがグループ分けされます。) たとえば、最初の OUTFIL ステートメントに INCLUDE オペランドがあって、DASD 上に 1 つの非 VSAM データ・セットをもち、さらにテープに別のデータ・セットをもつ OUTFIL グループにそのオペランドが適用される場合があります。また、2 番目の OUTFIL ステートメントに OMIT および OUTREC オペランドがあって、1 つの非 VSAM データ・セットを DASD 上にもち、さらに 2 つの VSAM データ・セットをもつ OUTFIL グループにこれらのオペランドが適用される場合があります。

OUTFIL の場合、レコードは、他のすべての DFSORT 処理が完了した後、SORTOUT の場合と同様に処理します。概念上は、**OUTFIL 入力レコード**は、E35 出口から渡されてから SORTOUT に書き込まれるまでの間に代行受信されるものと考えることができます。ただし、E35 出口も SORTOUT も、実際は、OUTFIL 処理で指定する必要はありません。そのことを念頭においた上で、OUTFIL 入力レコードを処理する前に行われる処理について、9 ページの図 2 を参照してください。特に次のことに注意してください。

- E15 または E35 出口、INCLUDE、OMIT、または SUM ステートメント、あるいは SKIPREC または STOPAFT パラメーターにより削除されたレコードは、OUTFIL 処理では使用できません。
- E15 出口、INREC または OUTREC ステートメント、あるいは E35 出口によりレコードが再フォーマット設定された場合、その結果として再フォーマット設定されたレコードは、OUTFIL フィールドが参照する必要がある OUTFIL 入力レコードです。
- **STARTREC** は、特定の OUTFIL 入力レコードから OUTFIL グループの処理を開始します。**ENDREC** は、特定の OUTFIL 入力レコードで OUTFIL グループの処理を終了します。STARTREC および ENDREC は、別々にあるいは一緒に、後続の OUTFIL 処理が適用される一定の範囲のレコードを選択します。
- **INCLUDE**、**OMIT**、および **SAVE** は、OUTFIL グループのデータ・セットに組み込むレコードを選択します。INCLUDE および OMIT は、各 OUTFIL 入力レコ

## OUTFIL 制御ステートメント

ードの指定フィールドに作用して、OUTFIL グループの出力レコードを選択します (すべてのレコードはデフォルトにより選択されます)。SAVE は、他のどの OUTFIL グループ用にも選択されないレコードを選択します。

INCLUDE および OMIT ステートメントはすべての入力レコードに適用されますが、INCLUDE および OMIT パラメーターはそれらの OUTFIL グループの OUTFIL 入力レコードのみに適用されます。INCLUDE および OMIT パラメーターには、INCLUDE および OMIT ステートメントのもつすべての論理式機能が備わっています。

- **SPLIT** は、出力レコードを OUTFIL グループのデータ・セットの間で順番に分割します。最初の出力レコードはグループの最初の OUTFIL データ・セットに書き込まれ、2 番目の出力レコードは 2 番目のデータ・セットに書き込まれる、という風になります。各 OUTFIL データ・セットに 1 つずつレコードが入ると、最初の OUTFIL データ・セットから再びローテーションが始まります。
- **OUTREC** は、OUTFIL グループの出力レコードを再フォーマット設定します。OUTREC を使用すると、OUTFIL 入力レコードのフィールドの再編成、編集、および変更を行うことができ、またブランク、ゼロ、ストリング、現在日付、現在時刻およびシーケンス番号およびシーケンス番号を挿入できます。  
OUTREC を使用すると、各入力レコードから、ブランクの出力レコードを間に挿入するか、あるいは挿入せずに、複数の再フォーマット設定された出力レコードを作成することもできます。  
VTOF または CONVERT を OUTREC と一緒に使用して、可変長入力レコードを固定長出力レコードに変更できます。  
VLFILL を使用すると、OUTREC 指定フィールドのすべてを含むには短すぎる、可変長入力レコードが処理できます。  
OUTREC ステートメントはすべての入力レコードに適用されますが、OUTREC パラメーターは、その OUTFIL グループの OUTFIL 入力レコードにしか適用されません。さらに、OUTREC パラメーターは、OUTREC ステートメントではサポートされない、ブランク・レコードおよび新規レコードの作成のための順方向のスラッシュ (/) 区切り文字がサポートされています。
- **FTOV** を使用して、固定長入力レコードを可変長出力レコードに変更できます。FTOV は、OUTREC と一緒でも、一緒でなくても使用できます。
- **VLTRIM** を使用して、ブランク、2 進ゼロまたはアスタリスクなどの、指定値の後書きバイトを可変長レコードから削除できます。VLTRIM は、FTOV と一緒でも、一緒でなくても使用できます。
- **LINES、HEADER1、TRAILER1、HEADER2、TRAILER2、SECTIONS、** および **NODETAIL** は、OUTFIL グループの報告書が作成されることを示し、その報告書用に作成する報告書レコードの詳細を指定します。報告書には、報告書ヘッダー (最初のページ)、報告書トレーラー (最後のページ)、ページ・ヘッダーとページ・トレーラー (それぞれ各ページの一番上と一番下)、およびセクション・ヘッダーとセクション・トレーラー (それぞれ各セクションの前と後) についての報告書レコードを入れることができます。

報告書のためのデータ・レコードは、OUTFIL 入力レコードを組み込むことにより得られます。再フォーマット設定したデータ・レコードを OUTFIL 入力レコードから作成するために、OUTREC パラメーターのすべての機能が利用可能です。

## OUTFIL 制御ステートメント

OUTFIL 順次入力レコードの各セットが指定フィールドについて同じ 2 進値をもっている場合は、対応するデータ・レコードのセットが報告書の中の 1 つのセクションとして扱われます。

データ・レコードの長さは、報告書の最大レコード長と同じかそれ以上である必要があります。報告書に使用される OUTFIL データ・セットは、ANSI 制御文字形式 (たとえば、RECFM=FBA または RECFM=VBA の 'A') を指定します。また、DFSORT が各報告書およびデータ・レコードに付加する紙送り制御文字のための LRECL に、余分のバイトが使用できるようにします。DFSORT はこれらの制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。DFSORT は、可能な場合、適切な紙送り制御文字 (たとえば、3 スペースに C'-) をヘッダーおよびトレーラー・レコードに使用して、書き出す報告書レコードの数を減らします。DFSORT は、常にシングル・スペース紙送り制御文字 (C' ) をデータ・レコードに使用します。OUTFIL データ・セットを表示させたときに、(表示方法により) これらの紙送り制御文字が表れない場合がありますが、報告書を印刷する場合はこれらが使用されます。

- **REMOVECC** を使用して、報告書から ANSI 制御文字を削除できます。このとき、'A' は RECFM に対して追加または要求されず、新たなバイトは LRECL に付加または要求されません。
- 243 ページの図 15 は、OUTFIL レコードとパラメーターが処理される順序を示しています。



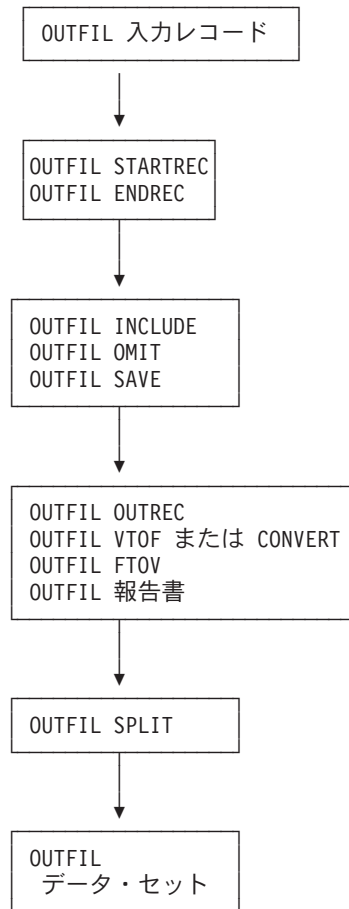


図 15. OUTFIL 処理の順序

**注:**

1. DFSORT は、以下の OUTFIL オペランドを受け付けますが、処理はしません。  
つまり、BLKSIZE=value、BUFLIM=value、BUFOFF=value、CARDS=value、  
CLOSE=value、DISK、ESDS、EXIT、FREEOUT、KSDS、LRECL=value、  
NOTPMK、OPEN=value、OUTPUT、PAGES=value、PRINT、PUNCH、  
REUSE、RRDS、SPAN、SYSLST、TAPE、および TOL。
2. この項では、サンプル構文が示されます。完全な OUTFIL ステートメント例  
は、311 ページの『OUTFIL 機能 - 例』に示され、その説明があります。

**FNAMES**



この OUTFIL ステートメントの OUTFIL データ・セットと関連のある DD 名を指定します。FNAMES および FILES パラメーターを使って指定された DD 名は、この OUTFIL ステートメントで他のすべてのパラメーターが適用されるこの OUTFIL グループの出力データ・セットの構成要素となります。

## OUTFIL 制御ステートメント

FNAMES が SORTOUT データ・セットに有効な DD 名 (つまり、SORTOUT、SORTOUT=name の name、または SORTDD=cccc の ccccOUT のうち、有効なもの) を指定している場合、DFSORT はその DD 名と関連するデータ・セットを、SORTOUT データ・セットではなく、OUTFIL データ・セットとして扱います。

### ddname

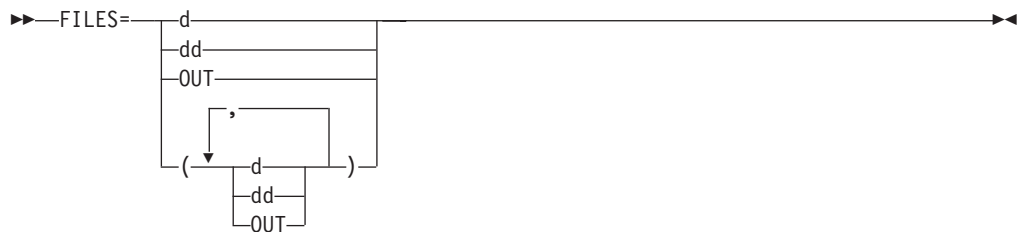
1 ~ 8 文字の DD 名を指定します。この DD 名には、DD ステートメントを使用します。

サンプル構文:

```
OUTFIL FNAMES=(OUT1,OUT2,PRINTER,TAPE)
OUTFIL FNAMES=BACKUP
```

FNAMES のデフォルト: OUTFIL ステートメントで FNAMES または FILES のどちらも指定されていない場合、SORTDD=cccc が有効であると、デフォルトの DD 名は SORTOUT または ccccOUT になります。

## FILES



DD 名のサフィックスが、この OUTFIL ステートメントの OUTFIL データ・セットに関連付けられるように指定します。FNAMES および FILES パラメータを使って指定された DD 名は、この OUTFIL ステートメントで他のすべてのパラメータが適用されるこの OUTFIL グループの出力データ・セットの構成要素となります。

FILES が SORTOUT データ・セットに有効な DD 名 (つまり、SORTOUT、SORTOUT=name の name、または SORTDD=cccc の ccccOUT のうち有効なもの) を指定している場合は、DFSORT は、その DD 名と関連するデータ・セットを、SORTOUT データ・セットではなく OUTFIL データ・セットとして扱います。

- d** 1 文字のサフィックスを使用して、SORTDD=cccc が有効な場合に DD 名 SORTOFd または ccccOFd を生成するように指定します。この DD 名には、DD ステートメントを使用します。
- dd** 2 文字のサフィックスを使用して、SORTDD=cccc が有効な場合に DD 名 SORTOFdd または ccccOFdd を生成するように指定します。この DD 名には、DD ステートメントを使用します。

### OUT

サフィックス OUT を使用して、SORTDD=cccc が有効な場合に DD 名 SORTOUT または ccccOUT を生成するように指定します。この DD 名には、DD ステートメントを使用します。

サンプル構文:

```
OUTFIL FILES=(1,2,PR,TP)
OUTFIL FILES=OUT
```

*FILES* のデフォルト: OUTFIL ステートメントで *FNAMES* または *FILES* のどちらも指定されていない場合、*SORTDD=cccc* が有効であると、デフォルトの *DD* 名は *SORTOUT* または *ccccOUT* になります。

### STARTREC

▶▶—STARTREC=*n*—————▶▶

OUTFIL グループで OUTFIL 処理を開始しようとしている OUTFIL 入力レコードを指定します。この開始レコードより前の OUTFIL 入力レコードは、この OUTFIL グループのデータ・セットには組み込まれません。

**n** 相対レコード番号を指定します。n に入る値は、1 (最初のレコード) から、28 桁 (有効な 15 の数字) までです。

サンプル構文:

```
OUTFIL FNAMES=SKIP20,STARTREC=21
```

*STARTREC* のデフォルト: 1。

### ENDREC

▶▶—ENDREC=*n*—————▶▶

OUTFIL グループで OUTFIL 処理を終了しようとしている OUTFIL 入力レコードを指定します。この終了レコードより後の OUTFIL 入力レコードは、この OUTFIL グループのデータ・セットには組み込まれません。

ENDREC 値と STARTREC 値の両方が同じ OUTFIL ステートメントで指定されている場合、ENDREC 値は STARTREC 値と同じかそれ以上である必要があります。

**n** 相対レコード番号を指定します。n に入る値は、1 (最初のレコード) から、28 桁 (有効な 15 の数字) までです。

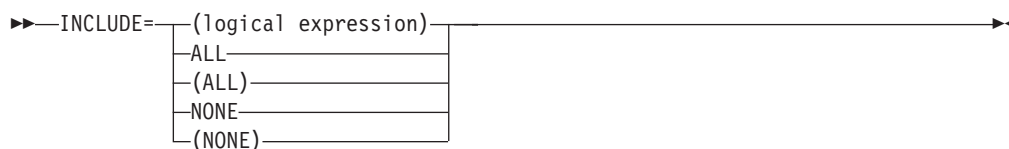
サンプル構文:

```
OUTFIL FNAMES=TOP10,ENDREC=10
OUTFIL FNAMES=FRONT,ENDREC=500
OUTFIL FNAMES=MIDDLE,STARTREC=501,ENDREC=2205
OUTFIL FNAMES=BACK,STARTREC=2206
```

*ENDREC* のデフォルト: 最後の OUTFIL 入力レコード。

### INCLUDE

## OUTFIL 制御ステートメント



この OUTFIL グループのデータ・セットに組み込むレコードを選択します。

INCLUDE パラメーターは、以下の点を除き、INCLUDE ステートメントと同じ動作をします。

- INCLUDE ステートメントはすべての入力レコードに適用されますが、INCLUDE パラメーターの場合はその OUTFIL グループの OUTFIL 入力レコードのみに適用されます。
- FORMAT=f は、INCLUDE ステートメントでは指定できますが、INCLUDE パラメーターでは指定できません。
- D2 形式は、INCLUDE ステートメントでは指定できますが、INCLUDE パラメーターでは指定できません。

詳細については、130 ページの『INCLUDE 制御ステートメント』を参照してください。

### logical expression

OUTFIL 入力レコードのフィールドに基づいて、論理的に組み合わせられた 1 つ以上の関係条件を指定します。あるレコードについて論理式が真であれば、そのレコードはこの OUTFIL グループのデータ・セットに組み込まれます。

#### ALL または (ALL)

すべての OUTFIL 入力レコードが、この OUTFIL グループのデータ・セットに組み込まれるように指定します。

#### NONE または (NONE)

この OUTFIL グループのデータ・セットに OUTFIL 入力レコードが 1 つも組み込まれないように指定します。

サンプル構文:

```
OUTFIL FNAMES=J69,INCLUDE=(5,3,CH,EQ,C'J69')
OUTFIL FNAMES=J82,INCLUDE=(5,3,CH,EQ,C'J82')
```

INCLUDE のデフォルト: ALL。

## OMIT



この OUTFIL グループのデータ・セットから除外するレコードを選択します。

## OUTFIL 制御ステートメント

OMIT パラメーターは、以下の点を除き、OMIT ステートメントと同じ動作をします。

- OMIT ステートメントは、すべての入力レコードに適用されます。これに対し OMIT パラメーターは、その OUTFIL グループの OUTFIL 入力レコードのみに適用されます。
- FORMAT=f は、OMIT ステートメントでは指定できますが、OMIT パラメーターでは指定できません。
- D2 形式は、OMIT ステートメントでは指定できますが、OMIT パラメーターでは指定できません。

完全な詳細については、177 ページの『OMIT 制御ステートメント』および 130 ページの『INCLUDE 制御ステートメント』を参照してください。

### logical expression

OUTFIL 入力レコードのフィールドに基づいて、論理的に組み合わせられた 1 つ以上の関係条件を指定します。あるレコードについて論理式が真であれば、そのレコードはこの OUTFIL グループのデータ・セットから除外されます。

### ALL または (ALL)

すべての OUTFIL 入力レコードが、この OUTFIL グループのデータ・セットから除外されるように指定します。

### NONE または (NONE)

この OUTFIL グループのデータ・セットから OUTFIL 入力レコードが 1 つも除外されないように指定します。

サンプル構文:

```
OUTFIL FILES=01,OMIT=NONE
OUTFIL OMIT=(5,1,BI,EQ,B'110.....')
OUTFIL FNAMES=(OUT1,OUT2),
      OMIT=(7,2,CH,EQ,C'32',OR,18,3,CH,EQ,C'XYZ')
```

OMIT のデフォルト: NONE。

## SAVE

▶▶—SAVE—◀◀

INCLUDE または OMIT により他のいずれの OUTFIL グループにも組み込まれていない OUTFIL 入力レコードを、この OUTFIL グループのデータ・セットに組み込むように指定します。SAVE は、SAVE が指定されていない他のすべての OUTFIL ステートメントについて、グローバル方式で作動し、これにより、そうしなければ保持できないようなすべての OUTFIL 入力レコードを保持できます。SAVE は、指定されたそれぞれのグループについて同じレコードを保管します。

サンプル構文:

```
OUTFIL INCLUDE=(8,6,CH,EQ,C'ACCTNG'),FNAMES=GP1
OUTFIL INCLUDE=(8,6,CH,EQ,C'DVPMNT'),FNAMES=GP2
OUTFIL SAVE,FNAMES=NOT10R2
```

SAVE のデフォルト: なし。必ず指定する必要があります。

## OUTFIL 制御ステートメント

### SPLIT

▶—SPLIT—▶

SPLIT は、出力レコードをこの OUTFIL グループのデータ・セットの間で順番に分割します。したがって、 $n$  個のデータ・セットをもつ OUTFIL グループの場合、グループの最初の OUTFIL データ・セットが受け取るレコードは 1、 $1+n$ 、 $1+2n$ 、...、2 番目のデータ・セットが受け取るレコードは 2、 $2+n$ 、 $2+2n$ 、...、というようにグループ内の各データ・セットごとにレコードを受け取っていき、ついには、すべての出力レコードが書き込まれます。その結果、グループ内のすべてのデータ・セット間にできる限り均等にレコードが分割されます。

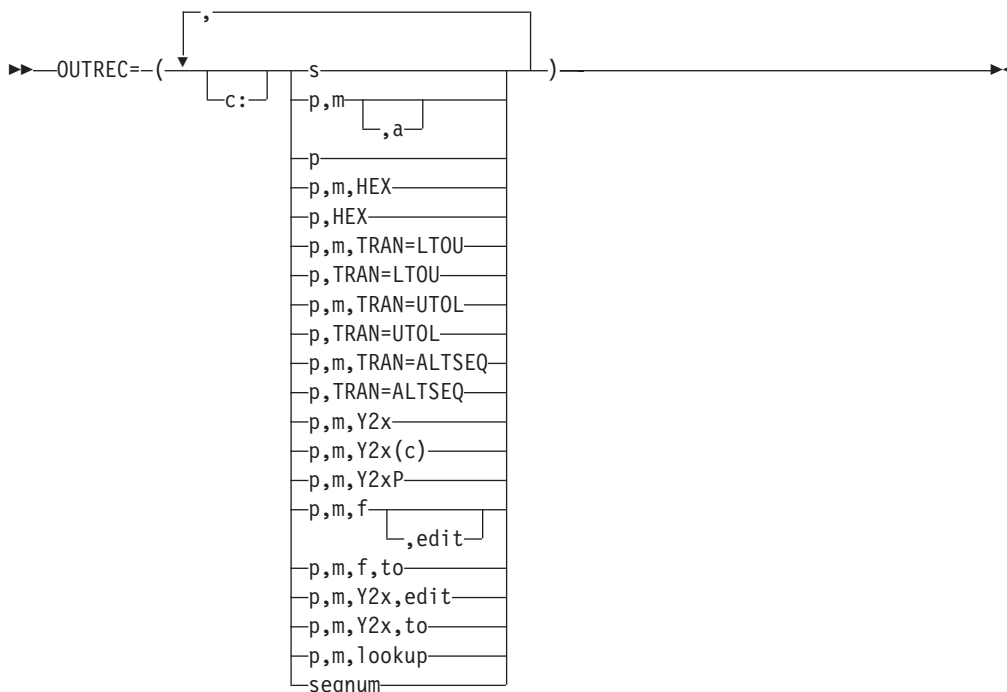
SPLIT パラメーターは、どの報告書パラメーター (LINES、HEADER1、TRAILER1、HEADER2、TRAILER2、SECTIONS、および NODetail) でも使用することができません。これは、報告書のレコードを分割することに意味がないからです。

サンプル構文:

```
OUTFIL FNAMES=(PIPE1,PIPE2,PIPE3,PIPE4),SPLIT
OUTFIL FNAMES=(TAPE1,TAPE2),SPLIT,
INCLUDE=(8,2,ZD,EQ,27),OUTREC=(5X,1,75)
```

SPLIT のデフォルト: なし。必ず指定する必要があります。

### OUTREC



| この OUTFIL グループのデータ・セットのレコードをどのように再フォーマット設定するかを指定します。 OUTREC は、OUTFIL 入力レコードのどの部分

## OUTFIL 制御ステートメント

を再フォーマット設定された OUTFIL 出力レコードに組み込むか、その部分をどのような順序で表示するか、どのように位置合わせをするか、さらにどのように編集または変更するかを指定できます。また、入力フィールドの前、間、および後に区切フィールド (ブランク、ゼロ、ストリング、現在日付、および現在時刻) とシーケンス番号を挿入したり、ブランクの出力レコードを間に挿入するか、あるいは挿入せずに、各入力レコードに複数の再フォーマット設定された出力レコードを作成することもできます。

OUTREC パラメーターを VTOF または CONVERT パラメーターと組み合わせて使用すると、可変長レコード・データ・セットを固定長レコード・データ・セットに変換できます。

OUTREC パラメーターを FTOV パラメーターと組み合わせて使用し、固定長レコード・データ・セットを可変長レコード・データ・セットに変換できます。

VLFILL を使用すると、OUTREC 指定フィールドのすべてを含むには短すぎる、可変長入力レコードが処理できます。

OUTREC パラメーターを報告書パラメーター (LINES、HEADER1、TRAILER1、HEADER2、TRAILER2、SECTIONS、および NODetail) のいずれかまたは全部を使用して、報告書を作成できます。報告書パラメーターでは、作成する報告書レコードが指定され、OUTREC パラメーターでは、作成する再フォーマット設定されたデータ・レコードが指定されます。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。REMOVECC パラメーターを使用して、ANSI 紙送り制御文字を削除できます。

OUTFIL 報告書を作成する場合、データ・レコードの最大長または唯一のデータ・レコードの長さは、報告書の最大レコード長と同じかそれ以上である必要があります。OUTREC パラメーターを使用して、どの報告書レコードよりも長いデータ・レコード長を指定できます。その場合は、DFSORT に LRECL を計算して設定させるか、あるいは算出された LRECL が、既存のまたは指定の LRECL 値と等しいことを確認できます。ANSI 紙送り制御文字の場合、LRECL に余分のバイトを用意することを忘れないでください。

たとえば、データ・レコードが 40 バイトであるのに最長報告書レコードが 60 バイトである場合、次のような OUTREC パラメーターを使用できます。

```
OUTREC=(1,40,80:X)
```

すると、DFSORT は LRECL を 81 (ANSI 紙送り制御文字の 1 バイトとデータ・レコード長の 80 バイト) に設定して、右側のデータ・レコードをブランクで埋め込みます。

ANSI 紙送り制御文字を出力データ・セットで使用しない場合は、REMOVECC パラメーターで削除します。たとえば、次のように指定すると仮定します。

```
OUTREC=(1,40,80:X),REMOVECC
```

DFSORT は LRECL を 81 の代わりに 80 に設定し、ANSI 紙送り制御文字が書き込まれる前に各レコードから削除します。

## OUTFIL 制御ステートメント

**注:** OUTREC ステートメントはすべての入力レコードに適用されますが、OUTFIL ステートメントの OUTREC パラメーターは、その OUTFIL グループの OUTFIL 入力レコードにしか適用されません。OUTFIL ステートメントの OUTREC パラメーターは、OUTREC ステートメントではサポートされない、ブランク・レコードおよび新規レコードの作成のための順方向のスラッシュ (/) 区切り文字がサポートされています。

再フォーマット設定した OUTFIL 出力レコードで、以下のいずれかまたは全部を組み込むよう選択できます。

- ブランク、2 進ゼロ、文字ストリング、16 進数ストリング
- 各種形式の現在日付および現在時刻
- バイト境界、ハーフワード境界、フルワード境界、ダブルワード境界で位置合わせされた未編集入力フィールド
- 2 進入力フィールドの 16 進表示
- 別の数値形式、あるいは文字形式 (符号、千単位の区切り、10 進小数点、先行ゼロまたは非先行ゼロなどを含むように編集された) に変換された各種形式の数値入力フィールド
- 別の数値形式、あるいは、区切り文字、先行ゼロまたは非先行ゼロなどを含むように編集された文字形式に変換された、SMF 日付または時間フィールド
- 別の数値形式の 4 桁年の日付か、あるいは、区切り文字、先行ゼロまたは非先行ゼロなどを含むように編集された文字形式に変換された、各種形式の 2 桁年の入力日付
- 各種形式のシーケンス番号
- 参照テーブルの文字ストリング または 16 進数ストリング

再フォーマット設定された OUTFIL 出力レコードは、ユーザーが選択した区切りフィールド、編集済みおよび未編集の入力フィールドで構成されます。これらのフィールドは、ユーザーが指定した順序で、境界またはユーザーが指示する桁で位置合わせされます。

- c:** 区切りフィールド、入力フィールドまたはシーケンス番号の位置 (桁) を、再フォーマット設定された OUTFIL レコードの先頭から相対的に指定します。c: を指定した場合、RDW (可変長出力レコードのみ) をカウントしますが、紙送り制御文字 (報告書の場合) はカウントしません。つまり、1: は固定長出力レコードのデータの最初のバイトを示し、5: は可変長出力レコードのデータの最初のバイトを示します。

指定された列の前にある未使用のスペースには、EBCDIC のブランクが埋め込まれます。次の規則が適用されます。

- c の値は 1 から 32752 の範囲で指定します。
- c: の後には、入力フィールドまたは区切りフィールドが続きます。
- 再フォーマット設定された OUTFIL 出力レコードでは、c は前の入力フィールドまたは区切りフィールドと重なり合うことはできません。
- 可変長レコードの場合、最初の入力フィールドの前 (レコード記述子ワード) にも OUTFIL 入力レコードの可変部の後にも、c: を指定できません。
- 次の行への継続する場合、コロン (;) はコンマ (,) またはセミコロン (;) と同様に取り扱われます。



桁合わせの有効な例と無効な例については、155 ページの表 21 を参照してください。

- s** 区切りフィールド (ブランク、ゼロ、文字ストリング、16 進数ストリング、現在日付、または現在時刻) が、再フォーマット設定された OUTFIL 出力レコードに表示されるように指定するか、またはブランクの出力レコードを間に挿入するか、あるいは挿入せずに、新しい出力レコードを開始させるように指定します。これらの区切り要素 (区切りフィールド、新規レコード標識、およびブランク・レコード標識) は、任意の入力フィールドの前または後に指定できます。連続した区切り要素を指定できます。可変長レコードの場合、最初の入力フィールドの前 (レコード記述子ワード)、または OUTFIL 入力レコードの可変部の後に、区切り要素を指定できません。暗黙的値は、nX、nZ、nC'xx...x'、nX'yy...yy'、DATE1、DATE1(c)、DATE1P、DATE2、DATE2(c)、DATE2P、DATE3、DATE3(c)、DATE3P、TIME1、TIME1(c)、TIME1P、TIME2、TIME2(c)、TIME2P、TIME3、TIME3P、/.../、および n/ です。

**nX** ブランクによる区切り。n バイトの EBCDIC ブランク (X'40') を、再フォーマット設定された OUTFIL 出力レコードに表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

ブランクによる区切りの有効な例および無効な例については、156 ページの表 22 を参照してください。

**nZ** 2 進ゼロによる区切り。n バイトの 2 進ゼロ (X'00') を、再フォーマット設定された OUTFIL 出力レコードに表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

2 進ゼロによる区切りの有効な例および無効な例については、156 ページの表 23 を参照してください。

**nC'xx...x'**

文字ストリングによる区切り。文字ストリング定数 (C'xx...x') を、再フォーマット設定した OUTFIL 出力レコードに n 回挿入します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。x には任意の EBCDIC 文字を指定できます。1 ~ 256 文字を指定できます。

文字ストリングに単一のアポストロフィを組み込む場合は、それを 2 つの単一アポストロフィで指定します。

Required: 0'NEILL Specify: C'0'NEILL'

文字ストリングによる区切りの有効な例と無効な例については、157 ページの表 24 を参照してください。

**nX'yy...yy'**

16 進数ストリングの区切り。16 進数ストリング定数 (X'yy...yy') を、再フォーマット設定した OUTFIL 出力レコードに n 回挿入します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

値 yy は、16 進数の任意の組み合わせを示します。1 組から 256 組までの 16 進数字を指定できます。

## OUTFIL 制御ステートメント

16 進数ストリングによる区切りの有効な例と無効な例については、158 ページの表 25 を参照してください。

### DATE $n$ , DATE $n$ ( $c$ ), DATE $n$ P

現在日付の定数。実行日付を再形成化された OUTFIL 出力レコードに表示します。表 32 に、指定可能な区切りフィールドごとにそれぞれ生成された定数を、ストリング長および例とあわせて、関連箇所 (c) に対し (/) を使用して、示します。yyyy は年、mm は月 (01 ~ 12)、dd は日付 (01 ~ 31)、ddd は年間通日 (001 ~ 366) を表します。また、c はブランク 以外のあらゆる文字として使用できます。

表 32. 現在日付の定数

区切り フィールド	定数	長さ (バイト)	2001 年 4 月 19 日
DATE1	C'yyyymmdd'	8	C'20010419'
DATE1(c)	C'yyyymmdd'	10	C'2001/04/19'
DATE1P	P'yyyymmdd'	5	P'20010419'
DATE2	C'yyyymm'	6	C'200104'
DATE2(c)	C'yyyymm'	7	C'2001/04 '
DATE2P	P'yyyymm'	4	P'200104'
DATE3	C'yyyddd'	7	C'2001109'
DATE3(c)	C'yyyddd'	8	C'2001/109'
DATE3P	P'yyyddd'	4	P'2001109'

### TIME $n$ , TIME $n$ ( $c$ ), TIME $n$ P

現在時刻の定数。実行時間を再形成化された OUTFIL 出力レコードに表示します。表 33 に、指定可能な区切りフィールドごとにそれぞれ生成された定数を、ストリング長および例とあわせて、関連箇所 (c) に対し (:) を使用して、示します。hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を表します。また、c はブランク 以外のあらゆる文字として使用できます。

表 33. 現在時刻の定数

区切り フィールド	定数	長さ (バイト)	01:55:43 PM
TIME1	C'hmmss'	6	C'135543'
TIME1(c)	C'hmmss'	8	C'13:55:43'
TIME1P	P'hmmss'	4	P'135543'
TIME2	C'hmm'	4	C'1355'
TIME2(c)	C'hmm'	5	C'13:55'
TIME2P	P'hmm'	4	P'1355'
TIME3	C'hh'	2	C'13'
TIME3P	P'hh'	2	P'13'

### /.../ または n/

ブランク・レコードまたは新規レコード。ブランクの出力レコードを間に挿入するか、あるいは挿入せずに、新規出力レコードを開始しま

## OUTFIL 制御ステートメント

す。/.../ または n/ が OUTREC の先頭か最後に指定されていると、ブランクの出力レコードが n レコード作成されます。/.../ または n/ が OUTREC の途中で指定されていると、ブランクの出力レコードが n-1 レコード作成されます (したがって、/ または / / とすると、新規出力レコードの間にブランクの出力レコードは挿入されません)。

/.../ または n/ を使用するときは、少なくとも 1 つの入力フィールドまたは区切りフィールドを指定する必要があります。たとえば、OUTREC=(//) は指定できませんが、OUTREC=(//X) は指定できます。

n/ (たとえば、5/) または複数の / (たとえば、////) のどちらかを使用できます。n は 1 ~ 255 の範囲の値になります。n が省略されていると、1 が使用されます。

たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(2/,C'Field 2 contains ',4,3,/,
                C'Field 1 contains ',1,3)
```

入力レコードの内容が次のとおりなら、

```
111222
```

次の 4 つの出力レコードが作成されます。

```
Blanks
Blanks
Field 2 contains 222
Field 1 contains 111
```

**1 つの OUTFIL 入力レコードごとに、4 つの OUTFIL 出力レコードが作成されることに注意してください。**

### p,m,a

未編集の入力フィールドを再フォーマット設定された OUTFIL 出力レコードに表示することを指定します。

**p** 入力フィールドの最初のバイトが、OUTFIL 入力レコードの先頭を基準にして、どの位置から始まるかを指定します。可変長レコードの最初のデータ・バイトは、相対位置が 5 になります。これは、最初の 4 バイトを RDW が占有するためです。すべてのフィールドはバイト境界から開始します。また、フィールドは 32752 バイトを超えることはできません。可変長レコードに関する特別な規則については、307 ページの『OUTFIL ステートメントに関する注意事項』を参照してください。

**m** 入力フィールドの長さをバイト単位で指定します。

**a** 再フォーマット設定された OUTFIL 出力レコード内の入力フィールドの位置合わせ (変位) を、再フォーマット設定された OUTFIL 出力レコードの先頭を基準にして指定します。

次の値を **a** に指定できます。

**H** ハーフワード位置合わせ。再フォーマット設定された OUTFIL 入力レコードの先頭からのフィールドの変位 (p-1) が、バイト単位で 2 の倍数 (すなわち、1、3、5 桁目など) であることを示します。

**F** フルワード位置合わせ。変位は、4 の倍数 (すなわち、1、5、9 桁目など) です。

## OUTFIL 制御ステートメント

- D** ダブル・ワード位置合わせ。変位は、8 の倍数 (すなわち、1、9、17 桁目など) です。

たとえば、COMPUTATIONAL 項目が SYNCHRONIZED 文節により位置合わせされている COBOL アプリケーション・プログラムでデータが使用される場合は、位置合わせが必要となります。位置合わせされたフィールドの前にある未使用のスペースは、常に 2 進ゼロが埋め込まれます。

- p** OUTFIL 入力レコードの未編集の可変部 (最小レコード長を超える部分) が、再フォーマット設定された OUTFIL 出力レコードに最後のフィールドとして表示されるよう指定します。再フォーマット設定された OUTFIL レコードに RDW と OUTFIL 入力レコードの可変部分だけを組み込むと、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

p に指定する値は、OUTFIL 入力レコードの最小レコード長に 1 バイトを加えた値と同じかそれ以下である必要があります。

### p,m,HEX

入力フィールドの 16 進表示が、再フォーマット設定された OUTFIL 出力レコードに表示されるように指定します。

- p** p,m,a の項の p を参照してください。

- m** 入力フィールドの長さをバイト単位で指定します。m の値は、1 から 16376 の範囲で指定します。

### HEX

入力フィールドの 16 進表示を要求します。入力フィールドの各バイトは、それに相当する 2 バイト値で置き換えられます。たとえば、文字 AB は C1C2 に置き換えられます。

### p,HEX

OUTFIL 入力レコードの可変部 (最小レコード長を超える部分) の 16 進表示が、再フォーマット設定された OUTFIL 出力レコードに最後のフィールドとして表示されるように指定します。再フォーマット設定された OUTFIL レコードに RDW と OUTFIL 入力レコードの可変部分だけを組み込むと、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

- p** p に指定する値は、最小レコード長に 1 バイトを加えた値と同じかそれ以下である必要があります。

### HEX

OUTFIL 入力レコードの可変部の 16 進表示を要求します。入力フィールドの各バイトは、それに相当する 2 バイト値で置き換えられます。たとえば、文字 AB は C1C2 に置き換えられます。

### p,m,TRAN=LTOU

入力フィールドの小文字の EBCDIC 文字 (すなわち、a ~ z) が、大文字の EBCDIC 文字 (すなわち、A ~ Z) として再フォーマット設定された OUTFIL 出力フィールドに表示されるように指定します。

- p** p,m,a の項の p を参照してください。

- m** p,m,a の項の m を参照してください。

### TRAN=LTOU

小文字から大文字への変換を要求します。入力フィールドの文字 a ~ z は

それぞれ、それに相当する文字 A ~ Z で置き換えられます。その他の文字は変更されません。たとえば、文字 'Vicky-123,x' は 'VICKY-123,X' に置き換えられます。

**p,TRAN=LTOU**

OUTFIL 入力レコードの可変部 (最小レコード長を超える部分) の小文字の EBCDIC 文字 (すなわち、a ~ z) が、大文字の EBCDIC 文字 (すなわち、A ~ Z) として再フォーマット設定された OUTFIL 出力レコードに最終フィールドとして表示されるように指定します。再フォーマット設定された OUTFIL レコードに RDW と OUTFIL 入力レコードの可変部分だけを組み合わせると、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

**p** p に指定する値は、最小レコード長に 1 バイトを加えた値と同じかそれ以下である必要があります。

**TRAN=LTOU**

EBCDIC 文字の小文字から大文字への変換を要求します。OUTFIL 入力レコードの可変部の文字 a ~ z はそれぞれ、それに相当する文字 A ~ Z で置き換えられます。その他の文字は変更されません。たとえば、文字 'Vicky-123,x' は 'VICKY-123,X' に置き換えられます。

**p,m,TRAN=UTOL**

入力フィールドの大文字の EBCDIC 文字 (すなわち、A ~ Z) が、小文字の EBCDIC 文字 (すなわち、a ~ z) として、再フォーマット設定された OUTFIL 出力レコードに表示されるように指定します。

**p** p,m,a の項の p を参照してください。

**m** p,m,a の項の m を参照してください。

**TRAN=UTOL**

EBCDIC 文字の大文字から小文字への変換を要求します。OUTFIL 入力レコードの可変部の文字 A ~ Z はそれぞれ、それに相当する文字 a ~ z で置き換えられます。その他の文字は変更されません。たとえば、文字 'CARRIE-005, CA' は 'carrie-005, ca' に置き換えられます。

**注:** TRAN-UTOL が数値データに使用される場合、意図せぬ結果を招くことがあります。たとえば、ZD フィールドの 32J (X'F3F2D1 = Z'-321') は、32j (X'F3F28A' = ZD の無効な数字) と表示されます。

**p,TRAN=UTOL**

OUTFIL 入力レコードの可変部 (最小レコード長を超える部分) の大文字の EBCDIC 文字 (すなわち、A ~ Z) が、小文字の EBCDIC 文字 (すなわち、a ~ z) として、再フォーマット設定された OUTFIL 出力レコードに最終フィールドとして表示されるように指定します。再フォーマット設定された OUTFIL レコードに RDW と OUTFIL 入力レコードの可変部分だけを組み合わせると、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

**p** p に指定する値は、最小レコード長に 1 バイトを加えた値と同じかそれ以下である必要があります。

**TRAN=UTOL**

大文字から小文字への変換を要求します。OUTFIL 入力レコードの可変部の文字 A ~ Z はそれぞれ、それに相当する文字 a ~ z で置き換えられま

## OUTFIL 制御ステートメント

す。その他の文字は変更されません。たとえば、文字 'CARRIE-005, CA' は 'carrie-005, ca' に置き換えられます。

**注:** TRAN-UTOL が数値データに使用される場合、意図せぬ結果を招くことがあります。たとえば、ZD フィールドの 32J (X'F3F2D1 = Z'-321') は、32j (X'F3F291'= 無効な 9 を含む ZD の数字) と表示されます。

### p,m,TRAN=ALTSEQ

入力フィールドの文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された OUTFIL 出力レコードで変更されるように指定します。

**p** p,m,a の項の p を参照してください。

**m** p,m,a の項の m を参照してください。

### TRAN=ALTSEQ

OUTFIL 入力レコードの可変部 (最小レコード長を超える部分) の文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された OUTFIL 出力レコードで、最終フィールドとして変更されるように指定します。再フォーマット設定された OUTFIL レコードに RDW と OUTFIL 入力レコードの可変部分だけを組み込むと、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

**注:** TRAN=ALTSEQ が数値データに使用される場合、意図せぬ結果を招くことがあります。たとえば、TRAN=ALTSEQ および ALTSEQ CODE=(0040) を使用する PD フィールドの X'00003C' (P'3') は、X'40403C' (P'40403') と表示されます。

### p,TRAN=ALTSEQ

OUTFIL 入力レコードの可変部 (最小レコード長を超える部分) の文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された OUTFIL 出力レコードで、最終フィールドとして変更されるように指定します。再フォーマット設定された OUTFIL レコードに RDW と OUTFIL 入力レコードの可変部分だけを組み込むと、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

**p** p に指定する値は、最小レコード長に 1 バイトを加えた値と同じかそれ以下である必要があります。

### TRAN=ALTSEQ

有効な ALTSEQ 変換テーブルに応じて、文字の変換を要求します。

OUTFIL 入力レコードの可変部の文字は、ALTSEQ テーブルで別の文字が割り当てられている場合、その文字で置き換えられます。その他の文字は変更されません。たとえば、次の ALTSEQ ステートメントを指定すると仮定します。

```
ALTSEQ CODE=(5C61,C1F1)
```

文字 '/\*XA\*/' は '//X1//' に置き換えられます。ALTSEQ テーブルの置き換え文字の割り当て方について、詳細は、120 ページの『ALTSEQ 制御ステートメント』を参照してください。

**注:** TRAN=ALTSEQ が数値データに使用される場合、意図せぬ結果を招くことがあります。たとえば、TRAN=ALTSEQ および ALTSEQ

## OUTFIL 制御ステートメント

CODE=(0040) を使用する PD フィールドの X'00003C' (P'3') は、X'40403C' (P'40403') と表示されます。

サンプル構文:

```
Fixed input records:
  OUTFIL FNames=(OUT1,OUT2),
           OUTREC=(1:5,10,15:8C'0',
                   25:20,15,TRAN=UTOL,80:X)

Variable input records:
  OUTFIL OUTREC=(1,4,C' RDW=',1,3,HEX,C' FIXED=',
                 5,20,HEX,C' VARIABLE=',21,HEX)
```

### p,m,Y2x

2 桁年の入力日付フィールドの 4 桁年の CH 日付表記を、再フォーマット設定された OUTFIL 出力レコードに表示するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2T は、C'000000' を C'00000000' に変換します)。

**p** p,m,a の項の p を参照してください。

**m** 2 桁年の日付フィールドの長さをバイト単位で指定します。

### Y2x

Y2 形式を指定します。形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

サンプル構文:

```
OUTFIL OUTREC=(21,3,Y2V,X,12,5,Y2W)
```

表 34 は、それぞれの日付タイプごとに作成される出力を示しています。

表 34. p,m,Y2x 出力

日付のタイプ	フィールド (m,f)		p,m,Y2x の出力
yyx	3,Y2T	2,Y2U	C'yyyyx'
yyxx	4,Y2T	3,Y2V	C'yyyyxx'
yyxxx	5,Y2T	3,Y2U	C'yyyyxxx'
yyxxxx	6,Y2T	4,Y2V	C'yyyyxxxx'
xyy	3,Y2W	2,Y2X	C'xyyyy'
xyy	4,Y2W	3,Y2Y	C'xyyyy'
xxyy	5,Y2W	3,Y2X	C'xxxyyyy'
xxxyy	6,Y2W	4,Y2Y	C'xxxyyyy'
yy	2,Y2C	2,Y2Z	C'yyyy'
yy	2,Y2S	2,Y2P	C'yyyy'
yy	1,Y2D	1,Y2B	C'yyyy'

### p,m,Y2x(c)

2 桁年の入力日付フィールドの、区切り文字付きの 4 桁年の CH 日付表記を、

## OUTFIL 制御ステートメント

再フォーマット設定された OUTFIL 出力レコードに表示するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2T(/) は、C'000000' を C'0000/00/00' に変換します)。

**p** p,m,a の項の p を参照してください。

**m** 2 桁年の日付フィールドの長さをバイト単位で指定します。

### Y2x

Y2 形式を指定します。形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

**c** 区切り文字を指定します。c には、ブランクを除く任意の文字を指定できます。

サンプル構文:

```
OUTFIL OUTREC=(25,6,Y2T(-),X,14,2,Y2U(/))
```

表 35 は、c に / が使用されるときの、Y2x(c) 日付フィールドのそれぞれのタイプで作成される出力を示しています。

表 35. p,m,Y2x(c) 出力

日付のタイプ	フィールド (m,f)		p,m,Y2x(/) の出力
yyx	3,Y2T	2,Y2U	C'yyyy/x'
yyxx	4,Y2T	3,Y2V	C'yyyy/xx'
yyxxx	5,Y2T	3,Y2U	C'yyyy/xxx'
yyxxxx	6,Y2T	4,Y2V	C'yyyy/xx/xx'
xyy	3,Y2W	2,Y2X	C'x/yyyy'
xyyy	4,Y2W	3,Y2Y	C'xx/yyyy'
xyyyy	5,Y2W	3,Y2X	C'xxx/yyyy'
xyyyyy	6,Y2W	4,Y2Y	C'xx/xx/yyyy'

### p,m,Y2xP

2 桁年の入力日付フィールドの 4 桁年の PD 日付表記を、再フォーマット設定された OUTFIL 出力レコードに出力するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2TP は、C'000000' を P'00000000' に変換します)。

**p** p,m,a の項の p を参照してください。

**m** 2 桁年の日付フィールドの長さをバイト単位で指定します。

### Y2xP

Y2 形式を指定します。形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

サンプル構文:

```
OUTFIL OUTREC=(11,3,Y2XP,X,21,4,Y2WP)
```



表 36 は、それぞれの日付タイプごとに作成される出力を示しています。

表 36. *p,m,Y2xP* 出力

日付のタイプ	フィールド (m,f)		<i>p,m,Y2xP</i> の出力
yyx	3,Y2TP	2,Y2UP	P'yyyyx'
yyxx	4,Y2TP	3,Y2VP	P'yyyyxx'
yyxxx	5,Y2TP	3,Y2UP	P'yyyyxxx'
yyxxxx	6,Y2TP	4,Y2VP	P'yyyyxxxx'
xyy	3,Y2WP	2,Y2XP	P'xyyyy'
xyyy	4,Y2WP	3,Y2YP	P'xyyyyy'
xxyy	5,Y2WP	3,Y2XP	P'xxyyyy'
xxxxyy	6,Y2WP	4,Y2YP	P'xxxxyyyy'
yy	2,Y2PP		P'yyyy'
yy	1,Y2DP		X'yyyy'

**p,m,f,edit**

編集済みの数値入力フィールドを再フォーマット設定された OUTFIL 出力レコードに表示するように指定します。事前定義の編集マスク (M0 ~ M26) またはユーザーが定義する特定の編集パターンのいずれかを使用して、BI、FI、PD、PD0、ZD、CSF/FS、DT1、DT2、DT3、TM1、TM2、TM3 または TM4 フィールドを編集できます。長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、編集済みフィールドの表示方法を制御できます。

**p** *p,m,a* の項の *p* を参照してください。

**m** 数値フィールドの長さをバイト単位で指定します。データが符号付きの場合、符号も長さに含まれます。長さに指定できる値については、260 ページの表 37 を参照してください。

**f** 数値フィールドの形式を指定します。

## OUTFIL 制御ステートメント

表 37. 編集フィールドの形式と長さ

形式コード	長さ	説明
BI	1 ~ 4 バイト	2 進数、符号なし
FI	1 ~ 4 バイト	固定小数点、符号あり
PD	1 ~ 8 バイト	パック 10 進数、符号あり
PD0	2 ~ 8 バイト	パック 10 進数、符号あり、1 桁目は無視
ZD	1 ~ 15 バイト	ゾーン 10 進数、符号あり
CSF または FS	1 ~ 16 バイト (15 桁まで)	任意指定の浮動符号が左端にある符号付き数字
DT1	4 バイト	Z'yyyymmdd' と解釈された SMF 日付
DT2	4 バイト	Z'yyyymm' と解釈された SMF 日付
DT3	4 バイト	Z'yyyddd' と解釈された SMF 日付
TM1	4 バイト	Z'hmmss' と解釈された SMF 時間
TM2	4 バイト	Z'hmmm' と解釈された SMF 時間
TM3	4 バイト	Z'hh' と解釈された SMF 時間
TM4	4 バイト	Z'hmmssxx' と解釈された SMF 時間

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

CSF または FS 形式フィールドの場合、

- 最大 15 桁が使用できます。16 桁の CSF/FS 値が見つかった場合、一番左の桁は正の符号を示すものとして扱われます。

ZD 形式または PD 形式フィールドの場合、

- 無効な数字があると、データ例外 (0C7 ABEND)、または誤った数値出力が起きます。A ~ F は無効な数字です。ICETOOL の VERIFY または DISPLAY 演算子を使用して、無効桁の 10 進値を識別できます。
- 符号が F、E、C、A、8、6、4、2、または 0 の場合、値は正の値として扱われます。
- 符号が D、B、9、7、5、3、または 1 の場合、値は負の値として扱われます。

PD0 形式フィールドの場合、

- 1 桁目は無視されます。
- 1 桁目以外に無効な数字があると、データ例外 (0C7 ABEND)、または誤った数値出力が起きます。A ~ F は無効な数字です。
- 符号は無視され、値は正の数値として扱われます。

DT1、DT2 または DT3 形式フィールドの場合、

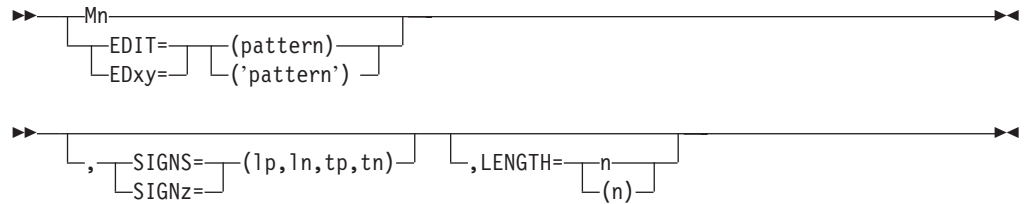
- 無効な SMF 日付があると、データ例外 (0C7 ABEND)、または誤った ZD 日付が発生します。
- SMF 日付値は常に正の数値として扱われます。

TM1、TM2、TM3、または TM4 形式フィールドの場合、

- 無効な SMF 時間があると、誤った ZD 時間が発生します。

- SMF 時間値は常に正の数値として扱われます。

**edit**



数値フィールドをどのようにして出力用に編集するかを指定します。

Mn、EDIT、または EDxy パラメーターが指定されていない場合、

- DT1、DT2、DT3、TM1、TM2、TM3、または TM4 フィールドは、M11 編集マスクを使用して編集されます。
- BI、FI、PD、PD0、ZD、または CSF/FS フィールドは、M0 編集マスクを使用して編集されます。

**Mn**

数値データを表すために 27 個の事前定義編集マスク (M0 ~ M26) の中の 1 つを指定します。これらの事前定義編集マスクが数値データを表すのに適切でない場合は、EDIT パラメーターにより自由にユーザー独自の編集パターンを定義できます。

27 個の事前定義編集マスクは次のように表すことができます。

表 38. 編集マスクのパターン

マスク	パターン	例	
		値	結果
M0	IIIIIIIIIIITS	+01234	1234
		-00001	1-
M1	TTTTTTTTTTTTTTS	-00123	00123-
		+00123	00123
M2	I,III,III,III,IIT.TTS	+123450	1,234.50
		-000020	0.20-
M3	I,III,III,III,IIT.TTCR	-001234	12.34CR
		+123456	1,234.56
M4	SI,III,III,III,IIT.TT	+0123456	+1,234.56
		-1234567	-12,345.67
M5	SI,III,III,III,IIT.TTS	-001234	(12.34)
		+123450	1,234.50
M6	III-TTT-TTTT	00123456	012-3456
		12345678	1-234-5678
M7	TTT-TT-TTTT	00123456	000-12-3456
		12345678	012-34-5678
M8	IT:TT:TT	030553	3:05:53
		121736	12:17:36

## OUTFIL 制御ステートメント

表 38. 編集マスクのパターン (続き)

マスク	パターン	例	
		値	結果
M9	IT/TT/TT	123094	12/30/94
		083194	8/31/94
M10	IIIIIIIIIIIT	01234	1234
		00000	0
M11	TTTTTTTTTTTTTTTT	00010	00010
		01234	01234
M12	SII,III,III,III,IT	+1234567	1,234,567
		-0012345	-12,345
M13	SIII.III.III.III.IT	+1234567	1.234.567
		-0012345	-12.345
M14	SIII III III III IITS	+1234567	1 234 567
		-0012345	(12 345)
M15	III III III III IITS	+1234567	1 234 567
		-0012345	12 345-
M16	SIII III III III IIT	+1234567	1 234 567
		-0012345	-12 345
M17	SIII'III'III'III'IIT	+1234567	1'234'567
		-0012345	-12'345
M18	SI,III,III,III,IT.TT	+0123456	1,234.56
		-1234567	-12,345.67
M19	SI.III.III.III.IT.TT	+0123456	1.234.56
		-1234567	-12.345.67
M20	SI III III III IIT,TTS	+0123456	1 234,56
		-1234567	(12 345,67)
M21	I III III III IIT,TTS	+0123456	1 234,567
		-1234567	12 345,67-
M22	SI III III III IIT,TT	+0123456	1 234,56
		-1234567	-12 345,67
M23	SI'III'III'III'IIT.TT	+0123456	1'234.56
		-1234567	-12'345.67
M24	SI'III'III'III'IIT,TT	+0123456	1'234,56
		-1234567	-12'345,67
M25	SIIIIIIIIIIIT	+01234	1234
		-00001	-1
I M26	STTTTTTTTTTTTTTTTT	1234	+01234
		-1	-00001

261 ページの表 38 に示した編集マスクの表示に用いられる要素は、次のとおりです。

- I - 先行する意味のない数字。ゼロの場合は、この桁は表示されません。

- **T** - 有効数字。ゼロの場合、この桁は表示されます。
- **CR** (M3 の場合) - 値が負の場合は数字の右に印刷され、それ以外の場合は数字の右に 2 つのブランクが印刷されます。
- **S** - 符号。そのパターンの先頭文字として表れる場合は、先行の符号です。パターンの最終文字に表れる場合は、後書き符号です。S がパターン先頭文字と最終文字の両方に表れる場合 (例: M5)、先頭文字は先行の符号で最終文字は後書き符号です。使用される符号の値は、先行の正符号 (lp)、先行の負符号 (ln)、後書きの正符号 (tp)、および後書きの負符号 (tn) の 4 つです。Mn 編集マスクに適切な値は次のとおりです。

表 39. 編集マスク符号

マスク	lp	ln	tp	tn
M0	なし	なし	ブランク	-
M1	なし	なし	ブランク	-
M2	なし	なし	ブランク	-
M3	なし	なし	なし	なし
M4	+	-	なし	なし
M5	ブランク	(	ブランク	)
M6	なし	なし	なし	なし
M7	なし	なし	なし	なし
M8	なし	なし	なし	なし
M9	なし	なし	なし	なし
M10	なし	なし	なし	なし
M11	なし	なし	なし	なし
M12	ブランク	-	なし	なし
M13	ブランク	-	なし	なし
M14	ブランク	(	ブランク	)
M15	なし	なし	ブランク	-
M16	ブランク	-	なし	なし
M17	ブランク	-	なし	なし
M18	ブランク	-	なし	なし
M19	ブランク	-	なし	なし
M20	ブランク	(	ブランク	)
M21	なし	なし	ブランク	-
M22	ブランク	-	なし	なし
M23	ブランク	-	なし	なし
M24	ブランク	-	なし	なし
M25	ブランク	-	なし	なし
M26	+	-	なし	なし

- 他のすべての文字 (たとえば、/) は、後述の特定の規則に従って、表示通りに印刷されます。

## OUTFIL 制御ステートメント

編集済みの出力フィールドの暗黙の長さは、使用する特定の編集マスクのパターンに必要な数字と文字の数により異なります。LENGTH パラメーターを使用して、編集済み出力フィールドの暗黙の長さを変更できます。

必要な数字の長さは、数値フィールドの形式および長さにより、次のように異なります。

表 40. 数値フィールドに必要な桁数

形式	入力の長さ	必要な桁数
ZD	m	m
PD	m	2m-1
PD0	m	2m-2
BI, FI	1	3
BI, FI	2	5
BI, FI	3	8
BI, FI	4	10
CSF または FS	16	15
CSF または FS	m (16 未満)	m

出力フィールドの長さは、各パターンごとに表 40 のように表されます (ここで、d は必要な桁数)。結果は最も近い整数に丸められます。

表 41. 編集マスク出力フィールドの長さ

マスク	出力フィールドの長さ	例	
		入力 (f,m)	出力の長さ
M0	$d + 1$	ZD,3	4
M1	$d + 1$	PD,5	10
M2	$d + 1 + d/3$	BI,4	14
M3	$d + 2 + d/3$	ZD,6	10
M4	$d + 1 + d/3$	PD,8	21
M5	$d + 2 + d/3$	FI,3	12
M6	12	ZD,10	12
M7	11	PD,5	11
M8	8	ZD,6	8
M9	8	PD,4	8
M10	d	BI,1	3
M11	d	PD,5	9
M12	$d + 1 + (d - 1)/3$	PD,3	7
M13	$d + 1 + (d - 1)/3$	FS,5	7
M14	$d + 2 + (d - 1)/3$	ZD,5	8
M15	$d + 1 + (d - 1)/3$	FI,3	11
M16	$d + 1 + (d - 1)/3$	ZD,6	8
M17	$d + 1 + (d - 1)/3$	FI,4	14
M18	$d + 1 + d/3$	BI,4	14

表 41. 編集マスク出力フィールドの長さ (続き)

マスク	出力フィールドの長さ	例	
		入力 (f,m)	出力の長さ
M19	d + 1 + d/3	PD,8	21
M20	d + 2 + d/3	FI,3	12
M21	d + 1 + d/3	ZD,3	5
M22	d + 1 + d/3	BI,2	7
M23	d + 1 + d/3	PD,6	15
M24	d + 1 + d/3	ZD,9	13
M25	d + 1	CSF,16	16
M26	d + 1	ZD,4	5

DFSORT がどのようにして数値から編集済みの出力を作成するかについて、以下の例で概念的に説明します。

```
OUTFIL OUTREC=(5,7,ZD,M5)
with ZD values of C'0123456'(+0123456)
and C'000302J'(-0003021)
```

上の表に示したように、次のことが決められています。

- M5 の通常のパターンは `SI,III,...,IIT.TTS` です。
- 使用する符号は、先行の正符号のときは空白、先行の負符号のときは `C('`、後書きの正符号のときは空白、後書きの負符号のときは `C')` です。
- 必要な桁数は 7 です。
- 出力フィールドの長さは 11 ( $7 + 2 + 7/3$ ) です。
- 出力フィールドの指定パターンは、次のようになります。`C'SII,IIT.TTS'`

`C'0123456'` の数字は、パターンにマップされて、`C'S01,234.56S'` になります。値が正のため、先行する符号と後書き符号は空白で置き換えられ、`C' 01,234.56 '` となります。最終的には、先頭の子ゼロ数字 (この場合は 1) より前にある数字はすべて空白に置き換えられ、最終出力結果は `C' 1,234.56 '` になります。

`C'000302J'` の数字は、パターンにマップされて、`C'S00,030.21S'` になります。値が負のため、先行する符号は `C('` で置き換えられ、後書き符号も `C')` で置き換えられて、その結果 `C'(00,030.21)` となります。先頭の子ゼロ数字 (この場合は 3) より前にある数字はすべて空白に置き換えられて、`C'( 30.21)` になります。最後に、先行する符号は先頭の子ゼロ数字の隣まで右に「浮動」され、最終出力結果は `C' (30.21)` になります。

より一般的な用語でこの規則を説明するために、DFSORT が概念的に数値から編集済み出力を作成するステップを以下に示します。

- 特定のパターンとその長さを、前記の表を用いて判別する。
- 数値の桁をパターンにマップする。

## OUTFIL 制御ステートメント

- 値が正の場合、先行する符号および後書きの符号があれば、263 ページの表 39 に示される正の値の文字で置き換える。値が正でない場合、先行する符号および後書きの符号があれば、同じ表に示される負の値の文字で置き換える。
- 先頭の子ゼロ数字 (I) または有効数字 (T) より前にあるすべての桁を空白で置き換える。
- 先行する符号があれば、先頭の子ゼロ数字 (I) または有効数字 (T) の隣まで右に移動させる。

以下の追加の規則は、編集マスクに適用されます。

- 一般的なパターンから特定のパターンを判別するには、符号、および入力の形式と長さから判別される必要な右端の数字、これらの右端の数字の間にあるすべての文字を組み込むことにより行います。これにより、有効数字 (T) を間違えて切り捨ててしまう場合があります。例として、5,1,ZD,M4 を指定すると、出力フィールドの長さは  $2(1 + 1 + 1/3)$  になります。M4 の一般的なパターンは、SI,III,...,IIT.TT ですが、特定のパターンは ST (先行する符号および右端の数字) になります。

### EDIT

数値データを表すための編集パターンを指定します。事前定義編集マスク (M0 ~ M26) が数値データを表すのに適切でない場合は、EDIT パラメーターにより自由にユーザー独自の編集パターンを定義できます。パターンを指定するために使用する要素は、編集マスクに使用するものと同じです。つまり、I、T、S、および印刷可能文字です。ただし、SIGNS パラメーターが同時に指定されていなければ、S は符号標識として認識されません。

#### pattern

使用する編集パターンを指定します。パターンを単一のアポストロフィで囲んでいないと、パターンに次の文字を指定することができません。つまり、空白、アポストロフィ、右括弧または左括弧、16 進数の 20、21、および 22 です。たとえば、EDIT=((IIT.TT)) は有効ですが、EDIT=(C)IIT.TT)、EDIT=(I / T)、および EDIT=(S'IIT) は有効ではありません。

ユーザーがパターン内で指定できる桁 (I および T) の最大数は、15 を超えてはなりません。パターンの最大長は、22 文字を超えてはなりません。

#### 'pattern'

使用する編集パターンを指定します。パターンを単一のアポストロフィで囲むと、16 進数の 20、21、または 22 を除くすべての文字をパターンに指定できます。パターン内に単一アポストロフィを組み込みたい場合は、2 つの単一アポストロフィで指定する必要がありますが、これらはパターン内では単一文字としてカウントされます。たとえば、EDIT=('C)IIT.TT')、EDIT=('I / T')、および EDIT=('S'IIT') はすべて有効です。

ユーザーがパターン内で指定できる桁 (I および T) の最大数は、15 を超えてはなりません。パターンの最大長は、22 文字を超えてはなりません。

編集済み出力フィールドの暗黙の長さは、パターンの長さと同じです。

LENGTH パラメーターを使用して、編集済み出力フィールドの暗黙の長さを変更できます。



DFSORT がどのようにして数値から編集済みの出力を作成するかについて、以下の例で概念的に説明します。

```
OUTFIL OUTREC=(1,5,ZD,EDIT=(**I/ITTCR))
```

```
with ZD values of C'01230'(+1230)
and C'0004J' (-41)
```

C'01230' の数字はパターンにマップされて、C'\*\*0/1230CR' になります。値が正のため、文字 (C'CR') のうち最後の数字の右側にある文字は空白に置き換えられ、C'\*\*0/1230 ' になります。先頭の非ゼロ数字 (この場合は 1) より前にある数字はすべて空白に置き換えられて、C'\*\* /1230 ' になります。最終的に、パターンの先頭の数字より前にある文字はすべて、先頭の非ゼロ数字の隣まで右に浮動して、C' \*\*1230 ' になります。

C'0004J' の数字はパターンにマップされて、C'\*\*0/0041CR' になります。値が負のため、文字 (C'CR') のうち最後の数字の右側にある文字は保持されます。先頭の T 数字より前にある数字はすべて空白で置き換えられ、C'\*\* / 041CR' になります。最終的に、パターンの先頭の数字より前にある文字はすべて、先頭の非ゼロ数字の隣まで右に浮動して、C' \*\*041CR' になります。

一般的には、DFSORT が概念的に数値から編集済み出力を作成するステップは、次のとおりです。

- 数値の桁をパターンにマップして、必要があればその左にゼロを埋め込む。
- 値が正の場合、先行の符号および後書き符号があれば、SIGNS パラメーターで指定された正の値に対応する文字で置き換えて、最後の数字と後書き符号 (もしあれば) の間にあるすべての文字を空白で置き換える。値が正でない場合、先行する符号および後書き符号があれば、SIGNS パラメーターで指定された負の値に対応する文字で置き換えて、最後の数字と後書き符号 (もしあれば) の間にあるすべての文字を保持する。
- 先頭の非ゼロ数字 (I) または有効数字 (T) より前にあるすべての桁を空白で置き換える。
- パターン内の先頭の数字より前に文字があれば、すべての文字を先頭の非ゼロ数字 (I) または有効数字 (T) まで右に移動させる。

以下の追加の規則は、編集マスクに適用されます。

- 有効数字 (T) より後の非有効数字 (I) は、有効数字として扱われます。
- SIGNS が指定されていると、パターン内の最初の文字または最後の文字にある S は符号として扱われ、パターン内のそれ以外の場所にある S は文字の S として扱われます。SIGNS が指定されていない場合は、S はパターン内のどこにあっても文字の S として扱われます。
- パターンに含まれる数字が値よりも少ない場合は、意図的にまたは意図せずに値の左端の数字が失われます。たとえば、5,5,ZD,EDIT=(IIT) を値 C'12345' に指定すると、結果は C'345' になります。また、たとえば、1,6,ZD,EDIT=(SIIT.T) を値 C'100345' に指定すると、結果は C' \$34.5' になります。

### EDxy

数値データを表すための編集パターンを指定します。EDxy は EDIT の特別な種類で、パターン内のその他の文字を I および T で置き換えることができま

## OUTFIL 制御ステートメント

す。たとえば、EDIT の代わりに EDAB を使用すると、パターン内で I と T の数字を表すために I の代わりに A を、T の代わりに B を使用します。x と y を同じ文字で指定できません。SIGNS が指定されている場合、x と y を S にすることはできません。SIGNz が指定されている場合、x と y を z と同じ文字で指定できません。x と y は、A ~ Z、#、\$、@、および 0 ~ 9の中から選択できます。

### SIGNS

編集マスク (Mn) あるいはパターン (EDIT または EDxy) に従って数値を編集するときに使用する符号値を指定します。4 つの符号値のうちのいずれかまたはすべてを指定できます。指定されていない値はすべてコンマで表す必要があります。ユーザーが指定していない符号の値すべてに、ブランクが使用されます。たとえば、SIGNS=(+,-) では、lp に +、ln に -、tp にブランク、tn にブランクが指定され、SIGNS=(,+,+) では、lp にブランク、ln にブランク、tp に +、tn に - が指定されます。

**lp** 先行する正の符号の値を指定します。編集マスクまたはパターンの最初の文字として S が指定され、しかもその値が正の場合、先行する符号として lp の値が使用されます。

**ln** 先行する負の符号の値を指定します。編集マスクまたはパターンの最初の文字として S が指定され、しかもその値が負の場合、先行する符号として ln の値が使用されます。

**tp** 後書きの正の符号の値を指定します。編集マスクまたはパターンの最後の文字として S が指定され、しかもその値が正の場合、後書きの符号として tp の値が使用されます。

**tn** 後書きの負の符号の値を指定します。編集マスクまたはパターンの最後の文字として S が指定され、しかもその値が負の場合、後書きの符号として tn の値が使用されます。

次のいずれかの文字を符号値として使用する場合、それらの文字を単一アポストロフィで囲む必要があります。コンマ、ブランク、あるいは左括弧または右括弧。単一アポストロフィは、4 つのアポストロフィ (つまり、単一アポストロフィで囲まれた 2 つの単一アポストロフィ) として指定する必要があります。

符号文字間の区切り文字としてのコンマをセミコロンで置き換えることはできません。

### SIGNz

パターン (EDIT または EDxy) に従って数値を編集するときに使用する符号の値を指定します。SIGNz は SIGNS の特別な種類で、パターン内の別の文字を S に置き換えることができます。たとえば、SIGNS の代わりに SIGNX を使用すると、符号を識別するためにパターン内で S の代わりに X を使用する必要があります。EDIT が指定されている場合、z を I または T にすることはできません。EDxy が指定されている場合、z を x または y のどちらかと同じ文字で指定できません。z は、A ~ Z、#、\$、@、および 0 ~ 9の中から選択できます。

### LENGTH

編集された出力フィールドの長さを指定します。編集マスクまたは編集パターン

## OUTFIL 制御ステートメント

を使用して作成された編集済み出力フィールドの暗黙の長さが数値データを表すのに適切でない場合、LENGTH を使って、編集済み出力フィールドを長くしたり短くしたりすることができます。

**n** 編集された出力フィールドの長さを指定します。n の値は 1 から 22 の範囲で指定します。

LENGTH では使用するパターンは変更されず、結果として作成される編集済み出力フィールドの長さだけを変更できます。たとえば、前に説明した Mn で次のように指定すると仮定します。

```
OUTFIL OUTREC=(5,1,ZD,M4)
```

この場合、数字の長さが 1 のため、パターンは C'ST.TT' ではなく C'ST' になります。次のように指定すると仮定します。

```
OUTFIL OUTREC=(5,1,ZD,M4,LENGTH=5)
```

これにより、パターンは C'ST.TT' ではなく、C' ST' になります。

暗黙の長さより短い値を n に指定すると、編集後に左側の切り捨てが行われます。たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(1,5,ZD,EDIT=($IIT.TT),LENGTH=5)
```

C'12345' の値が指定の \$IIT.TT パターンに従って編集されると C'\$123.45' が作成されますが、長さ 5 が指定されているためにこの値が C'23.45' に切り捨てられます。

暗黙の長さより長い値を n に指定すると、編集後にブランクでその左側を埋め込みます。たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(1,5,ZD,EDIT=($IIT.TT),LENGTH=10)
```

C'12345' の値が指定の \$IIT.TT パターンに従って編集されると C'\$123.45' が作成されますが、長さ 10 が指定されているために、この値が C' \$123.45' となります。

サンプル構文:

```
OUTFIL FNames=OUT1,OUTREC=(5:21,8,ZD,M19,25:46,5,ZD,M13)
OUTFIL FILES=1,OUTREC=(5,2,BI,C' * ',18,2,BI,80:X),
      ENDREC=2000,OMIT=(5,2,BI,EQ,18,2,BI)
OUTFIL FILES=(2,3),
      OUTREC=(11:35,6,FS,SIGNS=(,+,),LENGTH=10,
      31:8,4,PD,EDIT=(**II,IIT.TT),SIGNS=(,+,,-))
```

### p,m,f,to

変換済みの数値入力フィールドを再フォーマット設定された OUTFIL 出力レコードに表示するように指定します。BI、FI、PD、PD0、ZD、CSF/FS、DT1、DT2、DT3、TM1、TM2、TM3、または TM4 フィールドを、BI、FI、PD、ZD、または CSF/FS フィールドに変換できます。

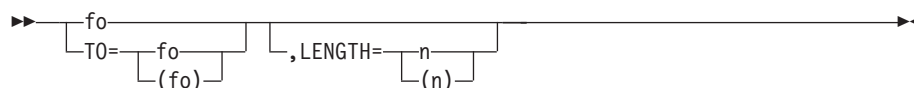
**p** p,m,f,edit の項の p を参照してください。

**m** p,m,f,edit の項の m を参照してください。

**f** p,m,f,edit の項の f を参照してください。

**to**

## OUTFIL 制御ステートメント



数値フィールドをどのようにして出力用に変換するかを指定します。

- fo** 出力フィールドの形式を指定します。BI、FI、PD、ZD、および CSF/FS の形式を指定できます。これらの出力フィールド形式 (fo) はいずれも、どの入力フィールド形式 (f) とも使用できます。

LENGTH パラメーターを指定しないと、DFSORT は、入力フィールドの長さ (m) と形式 (f) および出力フィールドの形式 (fo) から、出力フィールドの暗黙の長さを判別します。出力フィールドの暗黙の長さは、各出力形式で次のように表されます。ここで、d は、264 ページの表 40 に示されているように入力フィールドに必要な桁数であり、結果は最も近い整数に丸められます。

表 42. To 出力フィールドの長さ

出力形式	出力の長さ	入力例 (f,m)	出力長さの例
BI	4	FS,9	4
FI	4	ZD,7	4
PD	d/2 + 1	BI,4	6
ZD	d	PD,4	7
CSF または FS	d + 1	FI,3	9

ZD 出力の場合、F が正符号として使用され、D が負符号として使用されません。

PD 出力の場合、C が正符号として使用され、D が負符号として使用されません。

CSF/FS 出力の場合、空白が正符号として使用され、- が負符号として使用され、先行ゼロは抑制されます。

ZD、PD、および CSF/FS 出力の場合、最大値は 99999999999999 (15 桁)、最小値は -99999999999999 (15 桁) です。それぞれ、最大入力値と最小入力値に対応します。

BI 出力の場合、

- 4294967295 (X'FFFFFFF') より大きい入力値は、4294967295 (X'FFFFFFF') の出力値を作成します。
- ゼロより小さい入力値は、絶対出力値を作成します。たとえば、P'-5000' の入力値は、5000 (X'1388') の BI 出力値を作成します。

FI 出力の場合、2147483647 (X'7FFFFFFF') より大きい入力値は、2147483647 (X'7FFFFFFF') の出力値を作成し、-2147483648 (X'80000000') より小さい入力値は -2147483648 (X'80000000') の出力値を作成します。

fo、TO=fo、および TO=(fo) は、以下の点を除き、交換可能です。

- fo が LENGTH パラメーターの前に指定する必要があるのに対し、TO は LENGTH パラメーターの前後どちらにも指定できます。

## OUTFIL 制御ステートメント

- TO=fo または TO=(fo) は、fo が f と誤って解釈されないように、fo ではなく、シンボルの後に使用する必要があります。詳細については、第 7 章の OUTFIL OUTREC の説明を参照してください。

### LENGTH

変換された出力フィールドの長さを指定します。出力フィールドの暗黙の長さが適切でない場合、LENGTH を使って、出力フィールドを長くしたり短くしたりすることができます。

- n 変換された出力フィールドの長さを指定します。n の値は 1 から 22 の範囲で指定します。

暗黙の長さより短い値を n に指定すると、変換後に左側の切り捨てが行われます。たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(1,8,ZD,TO=PD,LENGTH=3)
```

値を ZL8'-12345678' (X'F1F2F3F4F5F6F7D8') および ZL8'58' (X'F0F0F0F0F0F0F5F8') とすると、暗黙の長さ (5) による変換で、PL5'-12345678' (X'012345678D') および PL5'58' (X'0000000058C') が作成されます。指定された長さ 3 により、PL3'-45678' (X'45678D') および PL3'58' (X'00058C') と切り捨てが行われます。

暗黙の長さより長い値を n に指定すると、変換後にその左側がブランクで埋め込まれます。

- CSF/FS 出力値はブランク
- ZD 出力値は文字ゼロ
- PD および BI 出力値は 2 進ゼロ
- 正の FI 出力値は 2 進ゼロ
- 負の FI 出力値は 2 進数の 1

たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(1,4,ZD,TO=FI,LENGTH=6)
```

値 ZL4'-1234' (X'F1F2F3D4') および ZL4'58' (X'F0F0F5F8') とすると、暗黙の長さ (4) による変換で、FL4'-1234' (X'FFFFFFB2E') および FL4'58' (X'000004D2') が作成されます。指定された長さ 6 により、FL6'-1234' (X'FFFFFFF2E') および FL6'58' (X'0000000003A') と埋め込みが行われます。

サンプル構文:

```
OUTFIL OUTREC=(21,5,ZD,TO=PD,X,8,4,ZD,FI,LENGTH=2)
```

### p,m,Y2x,edit

2 桁年の入力日付フィールドの編集済みの 4 桁年の CH 日付表記を、再フォーマット設定された OUTFIL 出力レコードに表示するように指定します。ユーザーが指定した 2 桁年の日付フィールドは、257 ページの表 34 に示されているような 4 桁年の ZD 日付フィールドに変換されてから、指定された編集パラメーターに従って編集されます。

たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(28,3,Y2V,EDIT=(TTTT-T-T))
```



れます。 find 定数は、文字ストリング定数および 16 進数ストリング定数か、ビット定数のどちらかを指定できます。

- 文字ストリング定数 (C'xx...x') および 16 進数ストリング定数 (X'yy...yy') は、1 ~ m バイトを指定でき、混在が可能ですが、ビット定数と一緒に使うことはできません。文字ストリング定数および 16 進数ストリング定数のコーディングの詳細については、130 ページの『INCLUDE 制御ステートメント』を参照してください。  
ストリングが m バイト未満の場合、文字ストリング定数の代わりに空白 (X'40')、または 16 進数ストリング定数の代わりにゼロ (X'00') を使用して、右側に m バイトの長さまで埋め込まれます。
- ビット定数 (B'bbbbbb') は 1 バイトの長さでなければならず、また文字またはストリング定数と混在させることはできません。ビット定数のコーディングの詳細については、130 ページの『INCLUDE 制御ステートメント』を参照してください。

### set

対応する find 定数が入力フィールドの値と一致する場合に出力フィールドとして使用される set 定数を指定します。 set 定数には、1 ~ v バイトの文字ストリング定数 (C'xx...x') または 16 進数ストリング定数 (X'yy...yy') が使用でき、これらを混在させることが可能です。文字ストリング定数および 16 進数ストリング定数のコーディングの詳細については、130 ページの『INCLUDE 制御ステートメント』を参照してください。

ストリングが v バイト未満の場合、ストリング定数の代わりに空白 (X'40')、あるいは 16 進数ストリング定数の代わりにゼロ (X'00') を使用して、右側に v バイトの長さまで埋め込まれます。

ビット定数の場合、ビットの指定が無視されるため、複数の find 定数が入力フィールドの値と一致することがありますが、最初に一致が見つかった set 定数が出力フィールドに使用されます。たとえば、次のように指定すると仮定します。

```
OUTFIL OUTREC=(5,1,
  CHANGE=(2,B'11.....',C'A',B'1.....',C'B'))
```

この場合、入力フィールドの値 X'C0' (B'11000000') は両方のビット定数と一致しますが、C'A' のほうが find 定数で最初に一致するため、set 定数に使用されます。

### NOMATCH

入力フィールドの値がいずれの find 定数とも一致しない場合の処置を指定します。NOMATCH を指定しておらず、しかもいずれの入力値でも一致するものが見つからない場合、DFSORT は処理を終了します。

NOMATCH を指定する場合は、CHANGE の後に指定します。

### set

一致するものが見つからない場合に出力フィールドに使用する set 定数を指定します。詳細については、CHANGE の項の set を参照してください。

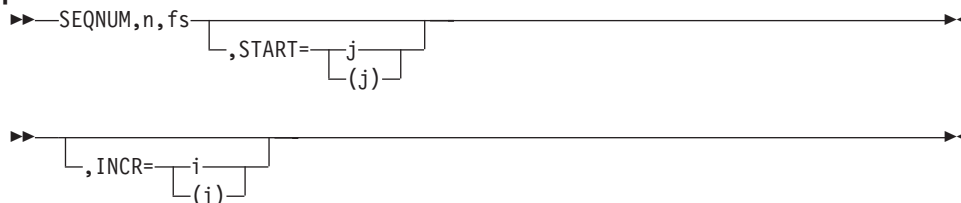
## OUTFIL 制御ステートメント

- q** 一致するものが見つからない場合に出カフィールドとして使用する入力フィールドの位置を指定します。詳細については、p,m,a の項の p を参照してください。
- n** 一致するものが見つからない場合に出カフィールドに使用する入力フィールドの長さを指定します。n の値は 1 から v の範囲で指定します。n が v より小さい場合は、入力フィールドはブランク (X'40') を使って右側に v バイトの長さまで埋め込まれます。

サンプル構文:

```
OUTFIL FILES=1,
OUTREC=(11,1,
CHANGE=(6,
C'R',C'READ',
C'U',C'UPDATE',
X'FF',C'EMPTY',
C'A',C'ALTER'),
NOMATCH=(11,6),
4X,
21,1,
CHANGE=(10,
B'.1.....',C'VSAM',
B'.0.....',C'NON-VSAM'))
```

### seqnum



シーケンス番号を再フォーマット設定された OUTFIL 出力レコードに表示するように指定します。シーケンス番号は、OUTFIL OUTREC 処理でレコードが受け取られる順序で割り当てられます。BI、PD、ZD、または CSF/FS シーケンス番号を作成し、それらの長さ、開始値および増分値を制御できます。

- n** シーケンス番号の長さを指定します。n の値は 1 から 16 の範囲で指定します。
- fs** シーケンス番号の形式を指定します。BI、PD、ZD、および CSF/FS の形式を指定できます。

ZD 形式のシーケンス番号の場合、F が符号として使用されます。

PD 形式のシーケンス番号の場合、C が符号として使用されます。

CSF/FS 形式のシーケンス番号の場合、ブランクが符号として使用され、先行ゼロは抑制されます。

PD、ZD、または CSF/FS 形式のシーケンス番号の場合、DFSORT が作成できる最大値は、15 桁の 10 進数または出力フィールド長 (n) の小さい方の値に制限されます。シーケンス番号がこの限度をオーバーフローする場合は、15 桁の 10 進数または出力フィールド長の小さい方に切り捨てられ、後で通常通り増分されます。

BI 形式のシーケンス番号の場合、DFSORT が作成できる最大値は、8 バイトの 1 (X'FFFFFFFFFFFFFF') か、指定された出力フィールド (n) に収ま



## OUTFIL 制御ステートメント

る 1 の数値の小さい方に制限されます。シーケンス番号がこの限度をオーバーフローする場合は、8 バイトまたは出力フィールド長の小さい方に切り捨てられ、後で通常通り増分されます。

### START

シーケンス番号の開始値を指定します。

**j** 開始値を指定します。j の値は、0 から 100000000000 の範囲で指定します。j のデフォルトは 1 です。

### INCR

シーケンス番号の増分値を指定します。

**i** 増分値を指定します。i の値は、1 から 10000000 の範囲で指定します。i のデフォルトは 1 です。

サンプル構文:

```
OUTFIL FNames=01,OUTREC=(SEQNUM,6,ZD,START=1000,INCR=50,  
X,22,8,X,13,5)  
OUTFIL FNames=02,OUTREC=(1,12,SEQNUM,4,BI)
```

*OUTREC* のデフォルト: なし。必ず指定する必要があります。

### VTOF または CONVERT



この OUTFIL グループの可変長 OUTFIL 入力レコードを固定長 OUTFIL 出力レコードに変換することを指定します。

OUTREC パラメーターを指定してください。指定したフィールドと列は、RDW を含まない再フォーマット設定された固定長 OUTFIL 出力レコードを作成します (データは 1 桁目から開始します)。指定した OUTREC フィールドはすべて、可変長 OUTFIL 入力レコードに適用します (データは 4 バイトの RDW の後の 5 桁目から開始します)。ただし、OUTFIL 入力レコードの可変部にパラメーターは指定できません (たとえば、p または p,HEX)。指定した OUTREC 列はすべて、再フォーマット設定された固定長 OUTFIL 出力レコードに適用します。

デフォルトでは、VTOF または CONVERT は自動的に VLFILL=X'40' (ブランク充てんバイト) を使用して、短すぎるために、指定されたすべての OUTREC フィールドを含むことができない可変長入力レコードを処理できるようにします。VLFILL=byte を指定して、充てんバイトを変更できます。

OUTFIL データ・セットに RECFM を指定しない場合、レコード形式 FB が設定されます。

OUTFIL データ・セットに RECFM を設定する場合、固定長レコード形式 (たとえば FB) である必要があります。

固定長入力レコードに VTOF または CONVERT が指定されているときは使用されません。

## OUTFIL 制御ステートメント

| VTOF または CONVERT が FTOV と一緒に指定される場合、DFSORT は終了  
| します。

サンプル構文:

```
OUTFIL FNAMES=FIXOUT,VTOF,  
OUTREC=(1:5,14,35:32,8,50:22,6,7c'*)
```

VTOF または CONVERT のデフォルト: なし。必ず指定する必要があります。

### VLFILL

▶▶—VLFILL=byte—◀◀

可変長 OUTFIL 入力レコードが短すぎて、指定されたすべての OUTFIL OUTREC フィールドをこの OUTFIL グループに含めることができないとわかった場合でも、DFSORT が処理を続行できるようにします。VLFILL=byte を指定しない場合は、短いレコードがあると、DFSORT はメッセージ ICE218A を発行して、終了します。VLFILL=byte を指定した場合は、OUTFIL OUTREC フィールドの失われたバイトが、充てんバイトで置き換えられ、充てんされたフィールドが処理できるようになります。

固定長入力レコードに VLFILL=byte が指定されているときは、この指定は使用されません。

| VLFILL=byte が FTOV と一緒に指定される場合、DFSORT は終了します。

#### byte

充てんバイトを指定します。指定できる値は C'x' と X'yy' です。

#### C'x'

文字バイト: 値 x は EBCDIC 1 文字で指定します。アポストロフィを充てんバイトとして使用するときは、C''' と指定します。

#### X'yy'

16 進バイト: 値 yy は 16 進数字 (00 ~ FF) のペアで指定します。

サンプル構文:

```
| OUTFIL FNAMES=FIXOUT,VTOF,OUTREC=(5,20,2X,35,10),VLFILL=C' *'  
| OUTFIL FNAMES=OUT1,VLFILL=X'FF',OUTREC=(1,4,15,5,52)
```

| VLFILL のデフォルト: VTOF または CONVERT が指定されている場合、  
| VLFILL=X'40' (ブランク充てんバイト)。指定されていない場合、なし。必  
| ず指定する必要があります。

### FTOV

▶▶—FTOV—◀◀

| この OUTFIL グループの固定長 OUTFIL 入力レコードを可変長 OUTFIL 出力  
| レコードに変換することを指定します。

## OUTFIL 制御ステートメント

OUTREC パラメーターを指定しない場合、指定した OUTREC フィールドおよび列は、可変長 OUTFIL 出力レコードに変換され、再フォーマット設定された固定長 OUTFIL 入力レコードを作成します。4 バイトの RDW は、書き込まれる前に固定長レコードとなります。

OUTREC パラメーターを指定する場合、指定した OUTREC フィールドおよび列は、可変長 OUTFIL 出力レコードに変換され、再フォーマット設定された固定長 OUTFIL レコードを作成します。指定した OUTREC フィールドはすべて、固定長 OUTFIL 入力レコードに適用します (データは 1 桁目から開始します)。4 バイトの RDW は、書き込まれる前に再フォーマット設定された固定長レコードとなります。

OUTFIL データ・セットに RECFM を指定しない場合、レコード形式 VB が設定されます。

OUTFIL データ・セットに RECFM を設定する場合、可変長レコード形式 (たとえば VB または VBS) である必要があります。

OUTFIL データ・セットに LRECL を指定しない場合、作成される最大の可変長出力レコードを含む LRECL が設定されます。非スパン・レコード形式 (たとえば VB) の場合は最大 32756、スパン・レコード形式 (たとえば VBS) の場合は最大 32767 が含まれます。

OUTFIL データ・セットに LRECL を指定する場合、作成する最大の可変長出力レコードを含むのに十分な容量である必要があります。

最大の可変長出力レコードが 32757 から 32767 バイトである場合、出力データ・セットにスパン・レコード形式 (たとえば VBS) を指定してください。

可変長入力レコードに FTOV が指定されているときは使用されません。

FTOV が VTOF、CONVERT または VLFILL=byte と一緒に指定される場合、DFSORT は終了します。

サンプル構文:

```
OUTFIL FNAMES=VAROUT,FTOV  
OUTFIL FNAMES=V1,FTOV,OUTREC=(1,20,26:21,10,6C'*)'
```

FTOV のデフォルト: なし。必ず指定する必要があります。

### VLTRIM=byte

▶▶—VLTRIM=byte—◀◀

VLTRIM=byte は、レコードが書き込まれる前に、この OUTFIL グループの可変長 OUTFIL 出力レコードの終端から後書きバイトが削除されるように指定します。

トリム・バイトには、ブランク、2 進ゼロまたはアスタリスクなど、あらゆる値を使用できます。DFSORT が 1 つ以上のトリム・バイトを可変長 OUTFIL データ・レコードまたは報告書レコードの終端に検出した場合、後書きのトリム・バイトを削除して、レコード長を相応かつ効果的に短縮します。ただし、

## OUTFIL 制御ステートメント

VLTRIM=byte は、RDW、(作成されている場合) ANSI 紙送り制御文字、または最初のデータ・バイトは削除しません。

たとえば、可変長データ・レコードに変換する、次の 17 バイトの固定長データ・レコードを使用すると仮定します。

```
123456*****
0003*****
ABCDEFGHIJ*****22
*****
```

次を使用する場合

```
OUTFIL FTOV
```

以下の可変長出力レコードが書き込まれます (データに続く 4 バイト RDW)。

Length	Data
21	123456*****
21	0003*****
21	ABCDEFGHIJ*****22
21	*****

次を使用する場合

```
OUTFIL FTOV,VLTRIM=C'*
```

以下の可変長出力レコードが書き込まれます (データに続く 4 バイト RDW)。

Length	Data
10	123456
8	0003
21	ABCDEFGHIJ*****22
5	*

VLTRIM=C'\*' は、1 番目と 2 番目のレコードから後書きのアスタリスクを削除しました。3 番目のレコードには、削除するアスタリスクがありませんでした。4 番目のレコードはすべてアスタリスクであったため、1 つのアスタリスクのみ保持されました。

固定長出力レコードに VLTRIM=byte が指定されているときは、この指定は使用されません。

**byte** トリム・バイトを指定します。指定できる値は C'x' と X'yy' です。

**C'x'** 文字バイト: 値 x は EBCDIC 1 文字で指定します。アポストロフィをトリム・バイトとして使用するとき、C'''' と指定します。

**X'yy'** 16 進バイト: 値 yy は 16 進数字 (00 ~ FF) のペアで指定します。

サンプル構文:

```
Fixed input:
OUTFIL FNames=TRIM1,FTOV,VLTRIM=C' '

Variable input:
OUTFIL FNames=TRIM2,VLTRIM=X'00'
OUTFIL FNames=TRIM3,VLTRIM=C'*',
OUTREC=(1,15,5X,16,8,5X,28)
```

VLTRIM のデフォルト: なし。必ず指定する必要があります。

**LINES**

▶—LINES=n—▶

この OUTFIL グループで作成された報告書に使用するページあたり行数を指定します。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

**n** ページあたり行数を指定します。n の値は 1 ~ 255 の値で指定します。ただし、n (または LINES が指定されていない場合の n のデフォルト) は、以下の各行に必要な行数と同じかそれ以上である必要があります。

- HEADER1 行
- TRAILER1 行
- HEADER2、TRAILER2、HEADER3 (複数)、TRAILER3 (複数)、および 1 つの入力レコードから作成されたデータ行 (複数) とブランク行 (複数) のすべての行の合計

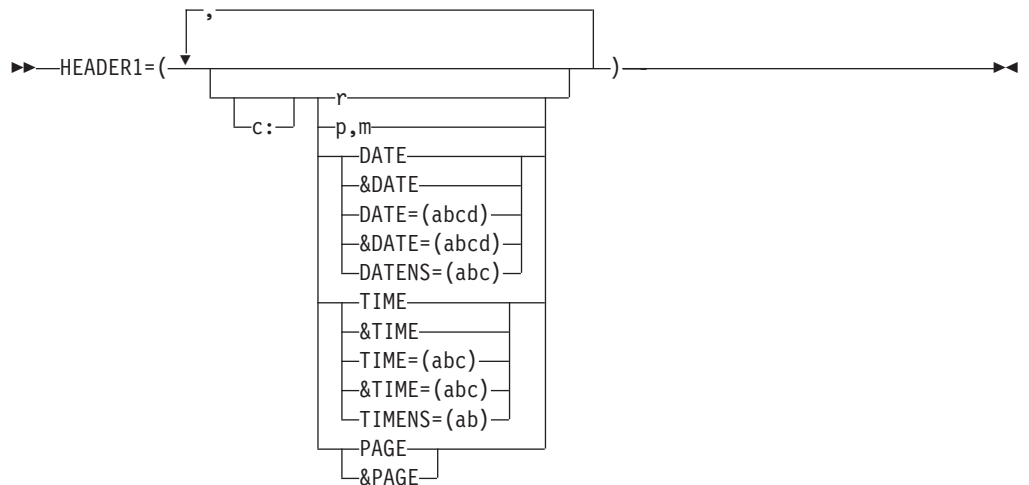
サンプル構文:

```
OUTFIL FNAMES=RPT1,LINES=50
```

LINES のデフォルト: なし。

HEADER1、TRAILER1、HEADER2、TRAILER2、SECTIONS、または NODetail が指定されていないければ、必ず指定する必要があります。これらが指定されている場合は、LINES のデフォルトは 60 です。

**HEADER1**



この OUTFIL グループで作成される報告書で使用する報告書ヘッダーを指定します。報告書ヘッダーは単独で報告書の最初のページに表示されます。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

報告書ヘッダーには、以下の報告書要素のいずれかまたは全部を組み込むように選択できます。

## OUTFIL 制御ステートメント

- ブランクおよび文字ストリング
- 最初の OUTFIL 入力レコードの未編集入力フィールド
- 現在の日付
- 現在の時間
- ページ番号

報告書ヘッダーは、ユーザーが選択した要素を指定した順番で、指定した桁または行に構成されます。

- c:** 報告書レコードのデータの先頭を基準にして、関連する報告書要素の最初の位置が表示される列を指定します。c: を指定すると、RDW (可変長報告書レコードのみ) および紙送り制御文字は無視されます。すなわち、1: は、固定長報告書レコードおよび可変長報告書レコードの両方の報告書レコードにあるデータの最初のバイトを示します。

指定された列の前にある未使用のスペースには、EBCDIC のブランクが埋め込まれます。次の規則が適用されます。

- c の値は 1 から 32752 の範囲で指定します。
  - c: の後に報告書要素を続ける必要がありますが、/ または n/ の前に置いてはなりません。
  - c は、報告書レコードの内前の報告書要素と重複してはいけません。
  - 次の行への継続する場合、コロン (;) はコンマ (,) またはセミコロン (;) と同様に取り扱われます。
- r** ブランクまたは文字ストリングが報告書レコードに表示されるように指定するか、あるいは、ブランク行を間に挿入するかどうかにかかわらず、新しい報告書レコードをヘッダーに開始させるように指定します。これらの報告書要素は、その他の報告書要素の前にも後にも指定できます。連続する文字ストリングまたはブランク行を指定できます。値には、nX、n'xx...x'、nC'xx...x'、/.../ および n/ が指定できます。

**nX** ブランク。n バイトの EBCDIC ブランク (X'40') が報告書レコードに表示されます。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

### n'xx...x'

文字ストリング。文字ストリング定数 ('xx...x') が報告書レコードに n 回現れます。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。x には任意の EBCDIC 文字を指定できます。1 ~ 256 文字を指定できます。

n'xx...x' の代わりに、nC'xx...x' を使用できます。

文字ストリングに単一のアポストロフィを組み込む場合は、それを 2 つの単一アポストロフィで指定します。

Required: 0'NEILL      Specify: '0''NEILL' or C'0''NEILL'

### /.../ または n/

ブランク行または改行。ブランク行を間に挿入するかどうかにかかわらず、新規の報告書レコードがヘッダーに開始されます。/.../ または n/ がヘッダーの始めまたは終わりで指定されていると、n ブランク行がヘッダーに現れます。/.../ または n/ がヘッダーの途中で

## OUTFIL 制御ステートメント

指定されていると、n-1 ブランク行がヘッダーに現れます (この場合、/ または // はブランク行が間にない改行を意味します)。

n/ (たとえば、5/) または複数の / (たとえば、/////) のどちらかを使用できます。n は 1 ~ 255 の範囲の値になります。n が省略されていると、1 が使用されます。

たとえば、次のように指定すると仮定します。

```
OUTFIL HEADER1=(2/,'First line of text',/,  
                'Second line of text',2/,  
                'Third line of text',2/)
```

報告書ヘッダーは、印刷されると次のように表示されます。

```
blank line  
blank line  
First line of text  
Second line of text  
blank line  
Third line of text  
blank line  
blank line
```

### p,m

データ・レコードが報告書に現れる最初の OUTFIL 入力レコードの未編集入力レコードが報告書レコードに表示されるように指定します。

**p** 入力フィールドの最初のバイトが、OUTFIL 入力レコードの先頭を基準にして、どの位置から始まるかを指定します。可変長レコードの最初のデータ・バイトは、相対位置が 5 になります。これは、最初の 4 バイトを RDW が占有するためです。すべてのフィールドはバイト境界から開始します。また、フィールドは 32752 バイトを超えることはできません。可変長レコードに関する特別な規則については、307 ページの『OUTFIL ステートメントに関する注意事項』を参照してください。

**m** 入力フィールドの長さをバイト単位で指定します。m の値は、1 から 256 の範囲で指定します。

### DATE

報告書レコードに現在の日付が 'mm/dd/yy' の形式で表示されるように指定します。ここで、mm は月 (01 ~ 12)、dd は日 (01 ~ 31)、yy は年の最後の 2 桁 (たとえば、95) を表します。

### &DATE

&DATE を DATE の代わりに使用できます。

### DATE=(abcd)

報告書レコードに現在の日付が 'adbdc' の形式で表示されるように指定します。ここで、a、b、および c は月、日、年が表示される順序と年が 2 桁または 4 桁のどちらで表示されるかを示し、d は月、日、年を区切るために使用されます。

a、b、および c には、月 (01 ~ 12) を示す M、日 (01 ~ 31) を示す D、年度の最後の 2 桁 (たとえば、95) を示す Y、または 4 桁の年 (たとえば、1995) を示す 4 を使用します。M、D、および Y または 4 は、それぞれ一度しか指定できません。例: DATE=(DMY.) の場合、'dd.mm.yy' 形式の日付が作成され、March 29, 1995 (1995 年 3 月 29 日) は '29.03.95' の

## OUTFIL 制御ステートメント

ように表示されます。DATE=(4MD-) の場合、'yyyy-mm-dd' 形式の日付が作成され、March 29, 1995 (1995 年 3 月 29 日) は '1995-03-29' のように表示されます。

a、b、c、および d は必ず指定する必要があります。

### **&DATE=(abcd)**

DATE=(abcd) の代わりに、&DATE=(abcd) を使用できます。

### **DATENS=(abc)**

報告書レコードに現在の日付が 'abc' の形式で表示されるように指定します。ここで、a、b、および c は月、日、年が表示される順序と年が 2 桁または 4 桁のどちらで表示されるかを示します。

a、b および c には、月 (01 ~ 12) を示す M、日 (01 ~ 31) を示す D、年度の最後の 2 桁 (たとえば、02) を示す Y、または 4 桁の年 (たとえば、2002) を示す 4 を使用します。M、D、および Y または 4 は、それぞれ一度しか指定できません。例: DATENS=(DMY) の場合、'ddmmyy' 形式の日付が作成され、March 29, 2002 (2002 年 3 月 29 日) は '290302' のように表示されます。DATENS=(4MD) の場合、'yyyymmdd' 形式の日付が作成され、March 29, 2002 (2002 年 3 月 29 日) は '20020329' のように表示されます。

a、b および c は必ず指定する必要があります。

### **TIME**

報告書レコードに現在の時間が 'hh:mm:ss' の形式で表示されるように指定します。ここで、hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を示します。

### **&TIME**

&TIME を TIME の代わりに使用できます。

### **TIME=(abc)**

報告書レコードに現在の時間が 'hhcmmcss' (24 時間表示) 形式または 'hhcmmcss xx' (12 時間表示) の形式で表示されるように指定します。

ab が 24 になっていると、時間は 'hhcmmcss' (24 時間表示) 形式で表示されます。ここで、hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を示し、c は時間、分、秒を区切るために使用される文字です。例: TIME=(24.) の場合、'hh.mm.ss' 形式の時間が作成され、08:25:13 pm (午後 8 時 25 分 13 秒) は '20.25.13' のように表示されます。

ab が 12 になっていると、時間は 'hhcmmcss xx' (12 時間表示) 形式で表示されます。ここで、hh は時間 (01 ~ 12)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を示し、xx は午前または午後、c は時間、分、秒を区切るために使用される文字です。例: TIME=(12.) の場合、'hh.mm.ss xx' 形式の時間が作成され、08:25:13 pm (午後 8 時 25 分 13 秒) は '08.25.13 pm' のように表示されます。

ab および c は必ず指定する必要があります。

### **&TIME=(abc)**

TIME=(abc) の代わりに、&TIME=(abc) を使用できます。



**TIMENS=(ab)**

報告書レコードに現在の時間が 'hhmmss' (24 時間表示) 形式または 'hhmmss xx' (12 時間表示) の形式で表示されるように指定します。

ab が 24 になっていると、時間は 'hhmmss' (24 時間表示) 形式で表示されます。ここで、hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を示します。例: TIMENS=(12) の場合、'hhmmss' 形式の時間が作成され、08:25:13 pm (午後 8 時 25 分 13 秒) は '202513' のように表示されます。

ab が 12 になっていると、時間は 'hhmmss xx' (12 時間表示) 形式で表示されます。ここで、hh は時間 (01 ~ 12)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を示します。例: TIMENS=(12) の場合、'hhmmss' 形式の時間が作成され、08:25:13 pm (午後 8 時 25 分 13 秒) は '082513' のように表示されます。

ab は必ず指定する必要があります。

**PAGE**

報告書レコードにページ番号が表示されるように指定します。報告書ヘッダーのページ番号は、' 1' のように表示されます。

HEADER1 が PAGE ありと指定されていると、報告書ヘッダー (最初のページ) の PAGE は ' 1' となり、次のページ (2 番目のページ) の PAGE は ' 2' となります。HEADER1 が PAGE なしと指定されていると、報告書ヘッダー以降のページ (2 番目のページ) の PAGE は ' 1' (カバー・シート付きの報告書の典型) になります。

**&PAGE**

&PAGE を PAGE の代わりに使用できます。

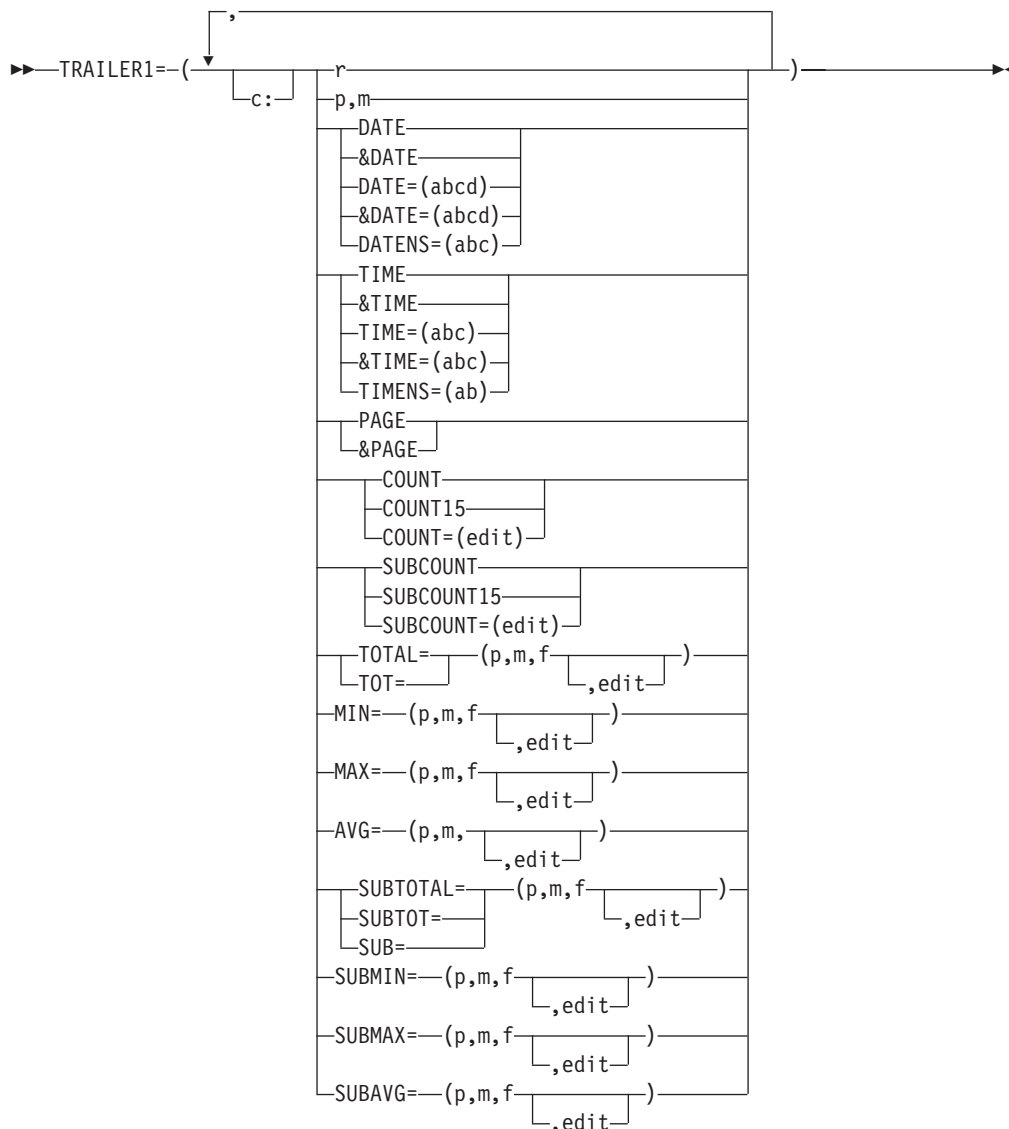
サンプル構文:

```
OUTFIL FNames=(RPT1,RPT2),
        HEADER1=(30:'January Report',4/,
                 28:'Prepared on ',DATE,//,
                 32:'at ',TIME,//,
                 28:'using DFSORT'S OUTFIL',5/,
                 10:'Department: ',12,8,50:'Page:',PAGE)
```

HEADER1 のデフォルト: なし。必ず指定する必要があります。

**TRAILER1**

## OUTFIL 制御ステートメント



この OUTFIL グループで作成された報告書に使用する報告書トレーラーを指定します。報告書トレーラーは、単独で報告書の最後のページに現れます。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

報告書トレーラーには、以下の報告書要素のいずれかまたは全部を組み込むように選択できます。

- ブランクおよび文字ストリング
- 最後の OUTFIL 入力レコードの未編集入力フィールド
- 現在の日付
- 現在の時間
- ページ番号
- 以下の統計のいずれかまたはすべて
  - 報告書にあるデータ・レコード数

## OUTFIL 制御ステートメント

- 符号、10 進小数点、先行ゼロ、または先行ゼロなし、などを含むように編集されたレコードのデータ・レコードの ZD、PD、BI、FI、または CSF/FS と指定された各数値入力フィールドの合計、最小、最大、または平均。

報告書トレーラーは、ユーザーが選択した要素を指定した順番で、指定した列または行に構成されます。

**c:** HEADER1 の項の c: を参照してください。

**r** ブランクまたは文字ストリングが報告書レコードに表示されるように指定したり、あるいはブランク行を間に挿入するかどうかにかかわらず、新しい報告書レコードをトレーラーに開始させるように指定します。これらの報告書要素は、その他の報告書要素の前にも後にも指定できます。連続する文字ストリングまたはブランク行を指定できます。値には、  
nX、n'xx...x'、nC'xx...x'、/.../、および n/ が指定できます。

**nX** ブランク。HEADER1 の r の項の nX を参照してください。

**n'xx...x'**

文字ストリング。HEADER1 の r の項の n'xx...x' を参照してください。n'xx...x' の代わりに、nC'xx...x' を使用できます。

**/.../ または n/**

ブランク行または改行。ブランク行を間に挿入するかどうかにかかわらず、新規の報告書レコードがトレーラーに開始されます。/.../ または n/ がトレーラーの始めまたは終わりで指定されていると、n ブランク行がトレーラーに現れます。/.../ または n/ がヘッダーの途中で指定されていると、n-1 ブランク行がトレーラーに現れます (この場合、/ または / はブランク行が間のない改行を意味します)。

n/ (たとえば、5/) または複数の / (たとえば、///// ) のどちらかを使用できます。n は 1 ~ 255 の範囲の値になります。n が省略されていると、1 が使用されます。

**p,m**

データ・レコードが報告書に表示される最後の OUTFIL 入力レコードから未編集入力レコードが報告書レコードに表示されるように指定します。

**p** HEADER1 の項の p を参照してください。

**m** HEADER1 の項の m を参照してください。

**DATE**

HEADER1 の項の DATE を参照してください。

**&DATE**

&DATE を DATE の代わりに使用できます。HEADER1 の項の &DATE を参照してください。

**DATE=(abcd)**

HEADER1 の項の DATE=(abcd) を参照してください。

**&DATE=(abcd)**

DATE=(abcd) の代わりに、&DATE=(abcd) を使用できます。HEADER1 の項の &DATE=(abcd) を参照してください。

## OUTFIL 制御ステートメント

|                   **DATENS=(abc)**  
|                   HEADER1 の項の DATENS=(abc) を参照してください。

**TIME**  
HEADER1 の項の TIME を参照してください。

**&TIME**  
&TIME を TIME の代わりに使用できます。HEADER1 の項の &TIME を参照してください。

**TIME=(abc)**  
HEADER1 の項の TIME=(abc) を参照してください。

**&TIME=(abc)**  
TIME=(abc) の代わりに、&TIME=(abc) を使用できます。HEADER1 の項の &TIME=(abc) を参照してください。

|                   **TIMENS=(ab)**  
|                   HEADER1 の項の TIMENS=(ab) を参照してください。

**PAGE**  
現行のページ番号が報告書レコードに表示されるように指定します。トレーラーのページ番号は、6 桁、右寄せ、先行ゼロ抑制で表示されます。たとえば、ページ番号が 12 のときは、'     12' のように表示されます。

**&PAGE**  
&PAGE を PAGE の代わりに使用できます。

**COUNT**  
報告書のデータ・レコード数が、8 桁、右寄せ、先行ゼロ抑制で報告書レコードに表示されるように指定します。たとえば、報告書に 6810 の入力レコードがある場合、レコード数は '     6810' のように表示されます。  
  
COUNT は入力レコードの数であり、データ・レコードの数ではありません。ただし、OUTREC に複数のレコードを作成するスラッシュ (/) が指定されていない限り、このレコード数はデータ・レコードの数も表します。

**COUNT15**  
カウントが 15 桁で表示されること以外は、COUNT と同じです。

**COUNT=(edit)**  
指定通りの編集で 15 桁のカウントが表示されること以外は、COUNT と同じです。使用できる編集フィールドの詳細については、OUTREC の項の p,m,f,edit を参照してください。

**SUBCOUNT**  
報告書のこの時点までの入力レコード数が、8 桁、右寄せ、先行ゼロ抑制で報告書レコードに表示されるように指定します。  
  
TRAILER1 では、実行レコード数はレコード数と同じなので、SUBCOUNT は COUNT と同じ値になります。  
  
SUBCOUNT は入力レコードの数であり、データ・レコードの数ではありません。ただし、OUTREC に複数のレコードを作成するスラッシュ (/) が指定されていない限り、この実行レコード数はデータ・レコードの数も表します。

**SUBCOUNT15**

実行中カウントが 15 桁で表示されること以外は、SUBCOUNT と同じです。

**SUBCOUNT=(edit)**

指定通りの編集で 15 桁の実行カウントが表示されること以外は、SUBCOUNT と同じです。使用できる編集フィールドの詳細については、OUTREC の項の p,m,f,edit を参照してください。

**TOTAL**

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された合計が報告書レコードに表示されるように指定します。

TOT を TOTAL の代わりに使用できます。

**p,m,f,edit**

合計を出したい数値入力フィールドを指定し、出力フィールド (つまり、合計) をどのように編集するかを指定します。

詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は TOTAL には使用できないこと、および TOTAL に関しては、Mn 編集マスクで必要とされる桁数が、次のように実際のフィールド長ではなく、その形式タイプの最大値であることに注意してください。

表 43. TOTAL フィールドに必要な桁数

形式	必要な桁数
ZD	15
PD	15
BI	10
FI	10
CSF または FS	15

**MIN**

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された最小値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

最小値を出したい数値入力フィールドを指定し、出力フィールド (つまり、最小値) をどのように編集するかを指定します。

詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は MIN には使用できないことに注意してください。

**MAX**

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された最大値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

最大値を出したい数値入力フィールドを指定し、出力フィールド (つまり、最大値) をどのように編集するかを指定します。

## OUTFIL 制御ステートメント

詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は MAX には使用できないことに注意してください。

### AVG

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された平均値が報告書レコードに表示されるように指定します。平均を算出するには、レコード数で合計を割り、その結果を最も近い整数に丸めます。たとえば、下記のようにします。

+2305 / 152 = +15  
-2305 / 152 = -15

#### p,m,f,edit

平均値を出したい数値入力フィールドを指定し、出力フィールド (つまり、平均) をどのように編集するかを指定します。

詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は AVG には使用できないことに注意してください。

### SUBTOTAL

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された実行レコード数の合計が報告書レコードに表示されるように指定します。

SUBTOT または SUB を SUBTOTAL の代わりに使用できます。

TRAILER1 では、実行合計は合計と同じなので、SUBTOTAL は TOTAL と同じ値になります。

#### p,m,f,edit

実行中の合計を出したい数値入力フィールドを指定し、出力フィールド (つまり、実行合計) をどのように編集するかを指定します。

詳細については、TOTAL の項の p,m,f,edit を参照してください。

### SUBMIN

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された実行最小値が報告書レコードに表示されるように指定します。

TRAILER1 では、実行最小値は最小値と同じなので、SUBMIN は MIN と同じ値になります。

#### p,m,f,edit

実行最小値を出したい数値入力フィールドを指定し、出力フィールド (つまり、実行最小値) をどのように編集するかを指定します。

詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は SUBMIN には使用できないことに注意してください。

### SUBMAX

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された実行最大値が報告書レコードに表示されるように指定します。

TRAILER1 では、実行最大値は最大値と同じなので、SUBMAX は MAX と同じ値になります。

**p,m,f,edit**

実行最大値を出したい数値入力フィールドを指定し、出力フィールド (つまり、実行最大値) をどのように編集するかを指定します。

詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は SUBMAX には使用できないことに注意してください。

**SUBAVG**

報告書のすべてのデータ・レコードの数値入力フィールドの値について、編集された実行平均値が報告書レコードに表示されるように指定します。

TRAILER1 では、実行平均値は平均値と同じなので、SUBAVG は AVG と同じ値になります。

**p,m,f,edit**

実行の平均値を出したい数値入力フィールドを指定し、出力フィールド (つまり、実行の平均値) をどのように編集するかを指定します。

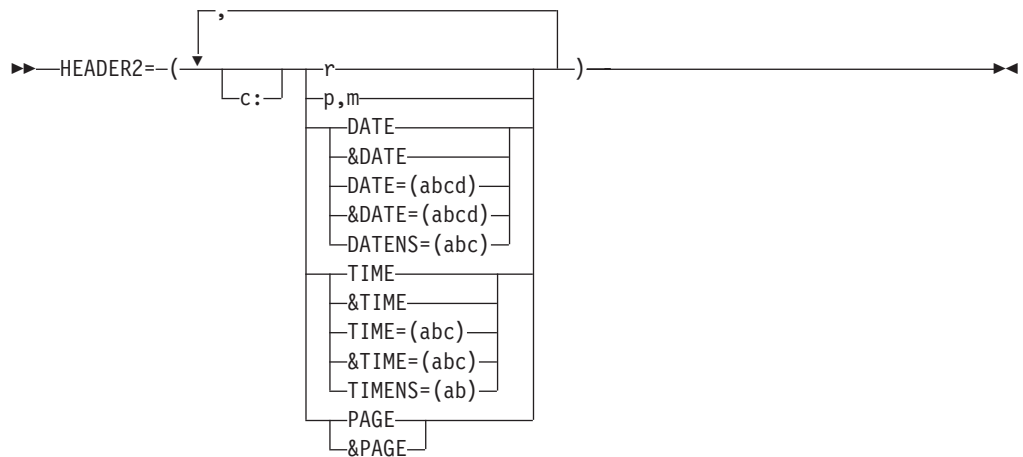
詳細については、OUTREC の項の p,m,f,edit を参照してください。ただし、PD0、DT1、DT2、DT3、TM1、TM2、TM3、および TM4 は SUBAVG には使用できないことに注意してください。

サンプル構文:

```
OUTFIL FNAMES=RPT,
TRAILER1=(5/,
10:'Summary of Report for Division Revenues',3/,
10:'Number of divisions reporting: ',COUNT,2/,
10:'Total revenue: ',TOTAL=(25,5,PD,M5),2/,
10:'Lowest revenue: ',MIN=(25,5,PD,M5),2/,
10:'Highest revenue: ',MAX=(25,5,PD,M5),2/,
10:'Average revenue: ',AVG=(25,5,PD,M5))
```

TRAILER1 のデフォルト: なし。必ず指定する必要があります。

**HEADER2**



この OUTFIL グループで作成される報告書に使用するページ・ヘッダーを指定します。ページ・ヘッダーは、報告書のヘッダー・ページ (もしあれば) と報告書のトレーラー・ページ (もしあれば) 以外の、報告書の各ページの一番上に表

## OUTFIL 制御ステートメント

示されます。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

ページ・ヘッダーには、以下の報告書要素のいずれかまたは全部を組み込むように選択できます。

- ブランクおよび文字ストリング
- データ・レコードがこのページに現れる最初の OUTFIL 入力レコードの未編集入力フィールド
- 現在の日付
- 現在の時間
- ページ番号

ページ・ヘッダーは、ユーザーが指定した順番で、指定した列または行に、ユーザーが選択した要素で構成されます。

**c:** HEADER1 の項の c: を参照してください。

**r** HEADER1 の項の r を参照してください。

### **p,m**

データ・レコードがページに現れる最初の OUTFIL 入力レコードの未編集入力レコードが報告書レコードに表示されるように指定します。詳細については、HEADER1 の項の p,m を参照してください。

### **DATE**

HEADER1 の項の DATE を参照してください。

### **&DATE**

&DATE を DATE の代わりに使用できます。HEADER1 の項の &DATE を参照してください。

### **DATE=(abcd)**

HEADER1 の項の DATE=(abcd) を参照してください。

### **&DATE=(abcd)**

DATE=(abcd) の代わりに、&DATE=(abcd) を使用できます。HEADER1 の項の &DATE=(abcd) を参照してください。

### **DATENS=(abc)**

HEADER1 の項の DATENS=(abc) を参照してください。

### **TIME**

HEADER1 の項の TIME を参照してください。

### **&TIME**

&TIME を TIME の代わりに使用できます。HEADER1 の項の &TIME を参照してください。

### **TIME=(abc)**

HEADER1 の項の TIME=(abc) を参照してください。

### **&TIME=(abc)**

TIME=(abc) の代わりに、&TIME=(abc) を使用できます。HEADER1 の項の &TIME=(abc) を参照してください。



**TIMENS=(ab)**

HEADER1 の項の TIMENS=(ab) を参照してください。

**PAGE**

現行のページ番号が OUTFIL 報告書レコードに表示されるように指定します。ヘッダーのページ番号は、6 桁、右寄せ、先行ゼロ抑制で表示されます。たとえば、ページ番号が 3 のときは、' 3' のように表示されます。

**&PAGE**

&PAGE を PAGE の代わりに使用できます。

HEADER1 が PAGE ありと指定され、かつ HEADER2 が PAGE ありと指定されている場合、最初のページ・ヘッダーのページ番号は ' 2' となります。HEADER1 が指定されていないかまたは PAGE なしと指定されており、かつ HEADER2 が PAGE ありと指定されている場合、最初のページ・ヘッダーのページ番号は ' 1' となります。

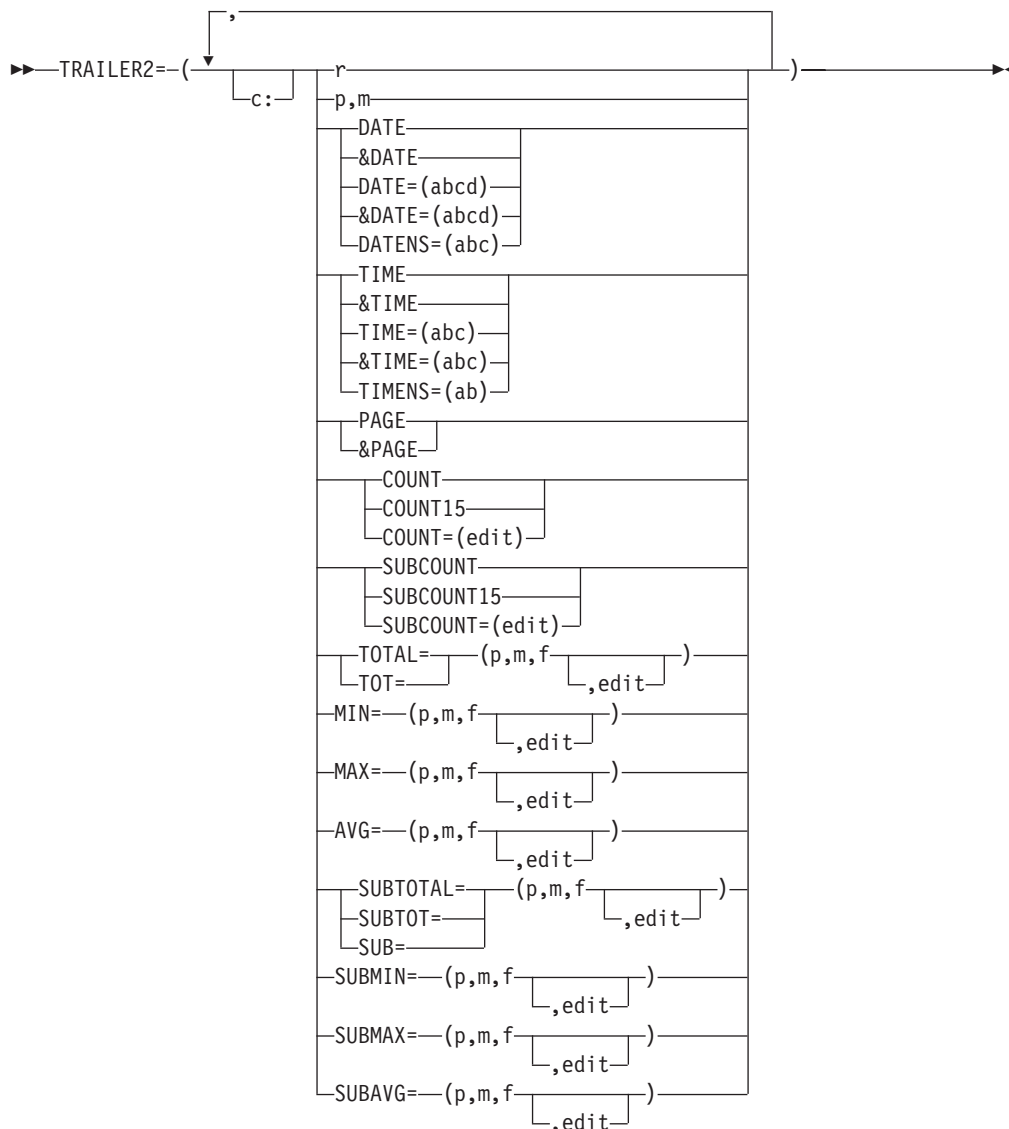
サンプル構文:

```
OUTFIL FNames=STATUS,
      HEADER2=(5:'Page ',PAGE,' of Status Report for ',DATE=(MD4/),
               ' at ',TIME=(12:),2/,
               10:'Item ',20:'Status ',35:'Count',/,
               10:'-----',20:'-----',35:'-----'),
      OUTREC=(10:6,5,
              20:14,1,CHANGE=(12,
                              C'S',C'Ship',
                              C'H',C'Hold',
                              C'T',C'Transfer'),
              NOMATCH=(C'*Check Code*'),
              36:39,4,ZD,M10,
              132:X)
```

HEADER2 のデフォルト: なし。必ず指定する必要があります。

**TRAILER2**

## OUTFIL 制御ステートメント



この OUTFIL グループのために作成される報告書に使用するページ・トレーラーを指定します。ページ・トレーラーは、報告書のヘッダー・ページ (もしあれば) と報告書のトレーラー・ページ (もしあれば) 以外の、報告書の各ページの一番下 (指定されたもの、あるいは LINES 値のデフォルトによるもの) に表示されます。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

ページ・トレーラーには、以下の報告書要素のいずれかまたは全部を組み込むよう選択できます。

- ブランクおよび文字ストリング
- データ・レコードがこのページに現れる最後の OUTFIL 入力レコードの未編集入力フィールド
- 現在の日付
- 現在の時間
- ページ番号

- 以下の統計のいずれかまたはすべて
  - ページ上のデータ・レコード数
  - 符号、10 進小数点、先行ゼロ、非先行ゼロなどを含むように編集されたページ上のデータ・レコードの ZD、PD、BI、FI、または CSF/FS と指定された各数値入力フィールドの合計、最小、最大、または平均。
  - 符号、10 進小数点、先行ゼロ、非先行ゼロなどを含むように編集された、この時点までのデータ・レコードの ZD、PD、BI、FI、または CSF/FS と指定された各数値入力フィールドについての実行合計、最小、最大、または平均。

ページ・トレーラーは、ユーザーが選択した要素を指定した順番で、指定した列または行に構成されます。

**c:** HEADER1 の項の c: を参照してください。

**r** TRAILER1 の項の r を参照してください。

**p,m**

データ・レコードがページに現れる最後の OUTFIL 入力レコードの未編集入力レコードが報告書レコードに表示されるように指定します。詳細については、TRAILER1 の項の p,m を参照してください。

**DATE**

HEADER1 の項の DATE を参照してください。

**&DATE**

&DATE を DATE の代わりに使用できます。HEADER1 の項の &DATE を参照してください。

**DATE=(abcd)**

HEADER1 の項の DATE=(abcd) を参照してください。

**&DATE=(abcd)**

DATE=(abcd) の代わりに、&DATE=(abcd) を使用できます。HEADER1 の項の &DATE=(abcd) を参照してください。

|  
|

**DATENS=(abc)**

HEADER1 の項の DATENS=(abc) を参照してください。

**TIME**

HEADER1 の項の TIME を参照してください。

**&TIME**

&TIME を TIME の代わりに使用できます。HEADER1 の項の &TIME を参照してください。

**TIME=(abc)**

HEADER1 の項の TIME=(abc) を参照してください。

**&TIME=(abc)**

TIME=(abc) の代わりに、&TIME=(abc) を使用できます。HEADER1 の項の &TIME=(abc) を参照してください。

|  
|

**TIMENS=(ab)**

HEADER1 の項の TIMENS=(ab) を参照してください。

## OUTFIL 制御ステートメント

### PAGE

TRAILER1 の項の PAGE を参照してください。

### &PAGE

&PAGE を PAGE の代わりに使用できます。TRAILER1 の項の &PAGE を参照してください。

### COUNT

ページ上のデータ・レコード数が、8 桁、右寄せ、先行ゼロ抑制で報告書レコードに表示されるように指定します。たとえば、入力レコードが、1 ページ目に 40、2 ページ目に 40、3 ページ目に 26 ある場合、COUNT は 1 ページ目が ' 40'、2 ページ目が ' 40'、3 ページ目が ' 26' となります。

COUNT は入力レコードの数であり、データ・レコードの数ではありません。ただし、複数のレコードを作成するスラッシュ (/) が指定されていない限り、このレコード数はデータ・レコードの数も表します。

### COUNT15

カウントが 15 桁で表示されること以外は、COUNT と同じです。

### COUNT=(edit)

指定通りの編集で 15 桁のカウントが表示されること以外は、COUNT と同じです。使用できる編集フィールドの詳細については、OUTREC の項の p,m,f,edit を参照してください。

### SUBCOUNT

報告書のこの時点までの入力レコード数が、8 桁、右寄せ、先行ゼロ抑制で報告書レコードに表示されるように指定します。実行レコード数は、現行のページを含むすべてのページのレコード数の累積になります。たとえば、入力レコードが、1 ページ目に 40、2 ページ目に 40、3 ページ目に 26 ある場合、SUBCOUNT は 1 ページ目が ' 40'、2 ページ目が ' 80'、3 ページ目が ' 106' となります。

SUBCOUNT は入力レコードの数であり、データ・レコードの数ではありません。ただし、複数のレコードを作成するスラッシュ (/) が指定されていない限り、この実行中レコード数はデータ・レコードの数も表します。

### SUBCOUNT15

実行中カウントが 15 桁で表示されること以外は、SUBCOUNT と同じです。

### SUBCOUNT=(edit)

指定通りの編集で 15 桁の実行カウントが表示されること以外は、SUBCOUNT と同じです。使用できる編集フィールドの詳細については、OUTREC の項の p,m,f,edit を参照してください。

### TOTAL

ページ上のデータ・レコードの数値入力フィールドの値について、編集された合計が報告書レコードに表示されるように指定します。

TOT を TOTAL の代わりに使用できます。

### p,m,f,edit

TRAILER1 - TOTAL の項の p,m,f,edit を参照してください。

**MIN**

ページ上のデータ・レコードの数値入力フィールドの値について、編集された最小値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

TRAILER1 - MIN の項の p,m,f,edit を参照してください。

**MAX**

ページ上のデータ・レコードの数値入力フィールドの値について、編集された最大値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

TRAILER1 - MAX の項の p,m,f,edit を参照してください。

**AVG**

ページ上のデータ・レコードの数値入力フィールドの値について、編集された平均値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

TRAILER1 - AVG の項の p,m,f,edit を参照してください。

**SUBTOTAL**

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行合計が報告書レコードに表示されるように指定します。実行合計は、現行のページを含むすべてのページのレコード数の累積になります。たとえば、選択された数値フィールドの合計が、1 ページ目で +200、2 ページ目で -250、3 ページ目で +90 の場合、SUBTOTAL は 1 ページ目で +200、2 ページ目で -50、3 ページ目で +40 となります。

SUBTOT または SUB を SUBTOTAL の代わりに使用できます。

**p,m,f,edit**

TRAILER1 - SUBTOTAL の項の p,m,f,edit を参照してください。

**SUBMIN**

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行最小値が報告書レコードに表示されるように指定します。実行最小値には、現行のページを含むすべてのページから最小値が選択されます。たとえば、選択された数値フィールドの最小値が、1 ページ目で +200、2 ページ目で -250、3 ページ目で +90 の場合、SUBMIN は 1 ページ目で +200、2 ページ目で -250、3 ページ目で -250 となります。

**p,m,f,edit**

TRAILER1 - SUBMIN の項の p,m,f,edit を参照してください。

**SUBMAX**

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行最大値が報告書レコードに表示されるように指定します。実行最大値には、現行のページを含むすべてのページから最大値が選択されます。たとえば、選択された数値フィールドの最大値が、1 ページ目で -100、2 ページ目で +250、3 ページ目で +90 の場合、SUBMAX は 1 ページ目で -100、2 ページ目で +250、3 ページ目で +250 となります。

**p,m,f,edit**

TRAILER1 - SUBMAX の項の p,m,f,edit を参照してください。

## OUTFIL 制御ステートメント

### SUBAVG

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行平均値が報告書レコードに表示されるように指定します。実行平均値には、現行のページを含むすべてのページから平均値が算出されます。たとえば、データ・レコード数と選択された数値フィールドの合計値がそれぞれ、1 ページ目で 60 レコードと +2205、2 ページ目で 60 レコードと -6252、3 ページ目で 23 レコードと -320 の場合、SUBAVG は 1 ページ目で +36、2 ページ目で -33、3 ページ目で -30 となります。

### p,m,f,edit

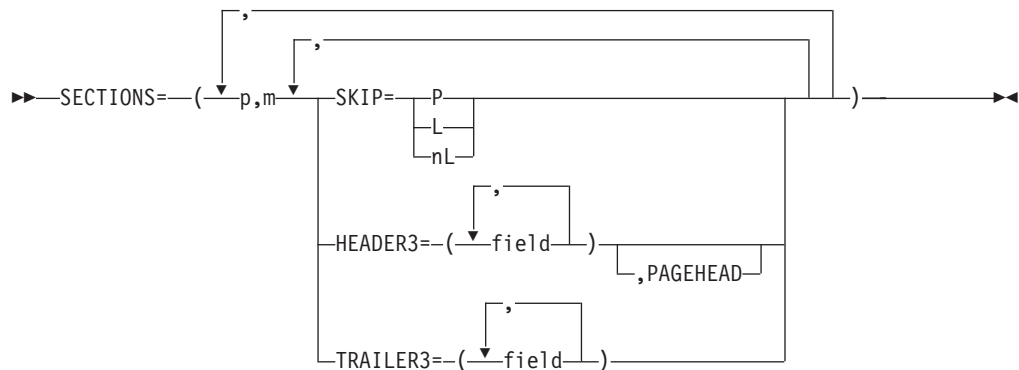
TRAILER1 - SUBAVG の項の p,m,f,edit を参照してください。

サンプル構文:

```
OUTFIL FAMES=STATS,
STARTREC=3,
OUTREC=(20:23,3,PD,M16,
        30:40,3,PD,M16,
        80:X),
TRAILER2=(/,2:'Average on page:',
          20:AVG=(23,3,PD,M16),
          30:AVG=(40,3,PD,M16),/,
          2:'Average so far:',
          20:SUBAVG=(23,3,PD,M16),
          30:SUBAVG=(40,3,PD,M16))
```

TRAILER2 のデフォルト: なし。必ず指定する必要があります。

### SECTIONS



この OUTFIL グループで作成される報告書で使用するセクション段落処理を指定します。セクション段落フィールドにより報告書は、そのフィールドに対して同じ 2 進値をもった連続する OUTFIL 入力レコードの集合に分割されます。これにより、対応するデータ・レコードの集合（つまり、セクション）が報告書内にできます。2 進値が変更されると、段落替えが生じると言われています。もちろん、段落替えはどのレコードでも可能なので、セクションのデータ・レコードは報告書のページにまたがって分割できます。

ユーザーが指定する各セクション段落フィールドごとに、次のいずれかまたはすべてを選択して組み込むことができます。

- セクション間でのページ替え。
- 同一ページのセクション間に表示する 0 行または 1 行以上のブランク行。

## OUTFIL 制御ステートメント

- 各セクションの最初のデータ・レコードの前、あるいは任意の選択により各ページの一番上に表示するセクション・ヘッダー。ページ・ヘッダーおよびセクション・ヘッダーの両方をページの一番上に表示させる場合、セクション・ヘッダーはページ・ヘッダーの後になります。
- 各セクションの最後のデータ・レコードの後に表示するセクション・トレーラー。ページ・トレーラーおよびセクション・トレーラーの両方をページの一番下に表示させる場合、ページ・トレーラーはセクション・トレーラーの後になります。

DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

複数のセクション段落フィールドを使用する場合、これらのフィールドを分類するのと同じように、最初のフィールドから最後のフィールドへ順番に処理されます。実際、入力データ・セットは、セクション段落フィールドで分類されるのが通常であり、同じセクション段落値をもつレコードを報告書用にまとめてグループ分けされます。この分類は、報告書を作成する同じアプリケーションにより、または直前のアプリケーションにより行われます。

セクション段落フィールド 1 において段落替えがあると、セクション段落フィールド 2 ~ n の間で段落替えが生じます。セクション段落 2 に段落替えがあると、セクション段落フィールド 3 ~ n で段落替えが生じる (以下同様) という具合になります。セクション・ヘッダーは最初のものから昇順に各セクションの前に表示されますが、セクション・トレーラーは最後のものから最初のものへ降順に表示されます。たとえば、ヘッダー H3A とトレーラー T3A をもつ B1、ヘッダー H3B とトレーラー T3B をもつ B2、ヘッダー H3C とトレーラー T3C をもつ B3 により表されるセクション段落フィールドが順番に指定されると、次のように表示させることができます。

```
H3A (header for B1=1 section)
  H3B (header for B2=1 section)
    H3C (header for B3=1 section)
      data records for B1=1, B2=1, B3=1 (new B1, B2, and B3 section)
    T3C (trailer for B3=1 section)
  H3C (header for B3=2 section)
    data records for B1=1, B2=1, B3=2 (new B3 section)
  T3C (trailer for B3=2 section)
T3B (trailer for B2=1 section)
H3B (header for B2=2 section)
  H3C (header for B3=1 section)
    data records for B1=1, B2=2, B3=1 (new B2 and B3 section)
  T3C (trailer for B3=1 section)
T3B (trailer for B2=2 section)
T3A (trailer for B1=1 section)
H3A (header for B1=2 section)
  H3B (header for B2=2 section)
    H3C (header for B3=0 section)
      data records for B1=2, B2=2, B3=0 (new B1, B2, and B3 section)
    T3C (trailer for B3=0 section)
  H3C (header for B3=1 section)
    data records for B1=2, B2=2, B3=1 (new B3 section)
  T3C (trailer for B3=1 section)
T3B (trailer for B2=2 section)
T3A (trailer for B1=2 section)
```

### p,m

報告書をセクションに分割するために使用する OUTFIL 入力レコードのセクション段落フィールドを指定します。連続する OUTFIL 入力レコードの各組が、

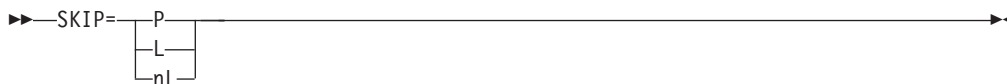
## OUTFIL 制御ステートメント

セクション段落フィールドについて同じ 2 進値をもっていると、それら是对應する組のデータ・レコードになります。このような各組のデータ・レコードはそれぞれ、報告書のセクションとして扱われます。2 進値が変更されると、段落替えが生じると言われています。

**p** HEADER1 の項の p を参照してください。

**m** HEADER1 の項の m を参照してください。

### SKIP



この OUTFIL グループで作成される報告書について、次のいずれかを指定します。

- 関連するセクション段落フィールドの各セクションを新規ページに表示させる。または、
- 後ろに同一ページの別のセクションが続く場合、このセクション段落フィールドに関連する各セクションの後に 0 行または 1 行以上の空白行を表示させる。

このように、SKIP を使用してどのように各セクションを互いに分離するかを指定できます。

**P** 各セクションが新しいページに表示されるように指定します。

**L** 同一ページ上のセクションの間に空白行を 1 行表示させるように指定します。L と 1L は同じです。

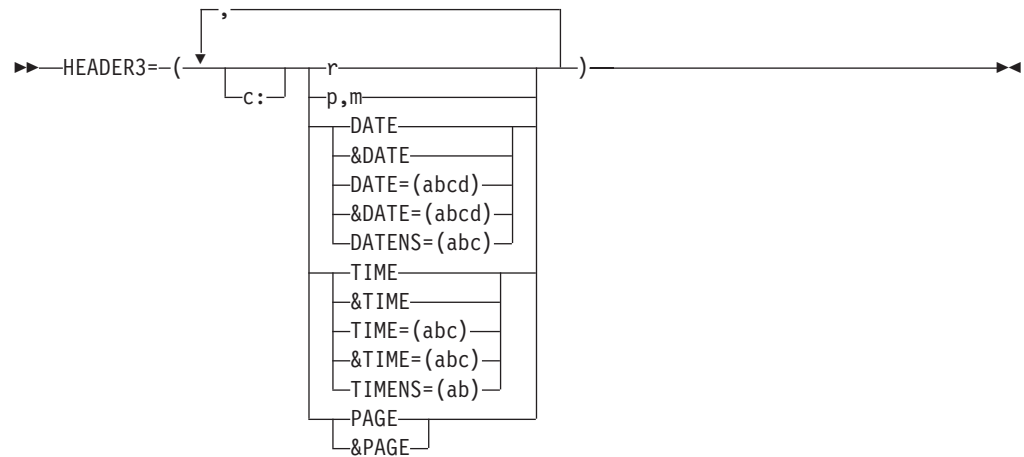
**nL** 同一ページ上のセクションの間に空白行を n 行表示させるように指定します。n に指定できる値は 0 ~ 255 の範囲です。

サンプル構文:

```
OUTFIL FNames=(PRINT,TAPE),  
SECTION=(10,20,SKIP=P,  
42,10,SKIP=3L)
```

### HEADER3





この OUTFIL グループについて作成される報告書の関連セクション段落フィールドで、このセクション・ヘッダーを使用するように指定します。セクション・ヘッダーは、各セクションの最初のデータ・レコードの前に表示されます。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

セクション・ヘッダーには、以下の報告書要素のいずれかまたは全部を組み込むよう選択できます。

- ブランクおよび文字ストリング
- データ・レコードがこのセクションに現れる最初の OUTFIL 入力レコードの未編集入力フィールド
- 現在の日付
- 現在の時間
- ページ番号

セクション・ヘッダーは、ユーザーが選択した要素を指定した順番で、指定した列または行に構成されます。

**c:** HEADER1 の項の c: を参照してください。

**r** HEADER1 の項の r を参照してください。

#### **p,m**

データ・レコードがこのセクションに現れる最初の OUTFIL 入力レコードの未編集入力レコードが報告書レコードに表示されるように指定します。詳細については、HEADER1 の項の p,m を参照してください。

#### **DATE**

HEADER1 の項の DATE を参照してください。

#### **&DATE**

&DATE を DATE の代わりに使用できます。HEADER1 の項の &DATE を参照してください。

#### **DATE=(abcd)**

HEADER1 の項の DATE=(abcd) を参照してください。

## OUTFIL 制御ステートメント

### &DATE=(abcd)

DATE=(abcd) の代わりに、&DATE=(abcd) を使用できます。HEADER1 の項の &DATE=(abcd) を参照してください。

### DATENS=(abc)

HEADER1 の項の DATENS=(abc) を参照してください。

### TIME

HEADER1 の項の TIME を参照してください。

### &TIME

&TIME を TIME の代わりに使用できます。HEADER1 の項の &TIME を参照してください。

### TIME=(abc)

HEADER1 の項の TIME=(abc) を参照してください。

### &TIME=(abc)

TIME=(abc) の代わりに、&TIME=(abc) を使用できます。HEADER1 の項の &TIME=(abc) を参照してください。

### TIMENS=(ab)

HEADER1 の項の TIMENS=(ab) を参照してください。

### PAGE

現行のページ番号が OUTFIL 報告書レコードに表示されるように指定します。ヘッダーのページ番号は、6 桁、右寄せ、先行ゼロ抑制で表示されます。たとえば、ページ番号が 3 のときは、' 3' のように表示されます。

### &PAGE

&PAGE を PAGE の代わりに使用できます。

サンプル構文:

```
OUTFIL FNAMES=STATUS1,
      HEADER2=(10:'Status Report for all departments',5X,
              '- ',&PAGE,' -'),
      SECTIONS=(10,8,
                HEADER3=(2/,10:'Report for department ',10,8,' on ',&DATE,2/,
                          10:'  Number',25:'Average Time',/,
                          10:'Completed',25:' (in days)',/,
                          10:'-----',25:'-----')),
                OUTREC=(10:21,5,ZD,M10,LENGTH=9,
                        25:38,4,ZD,EDIT=(III.T),LENGTH=12,
                        132:X)
```

### PAGEHEAD

▶▶—PAGEHEAD—◀◀

関連するセクション段落フィールドで使用されるセクション・ヘッダーを、報告書のヘッダー・ページ (もしあれば) と報告書のトレーラー・ページ (もしあれば) 以外で、各セクションの前に表示すると同時に、報告書の各ページの一番上に表示するように指定します。PAGEHEAD が指定されていないときは、セクション・ヘッダーは各セクションの前のみに表示されます。したがって、セクションが複数ページに分かれる場合は、セクション・ヘッダーはページの途中のみ

## OUTFIL 制御ステートメント

に表示されます。HEADER3 をセクション・ヘッダーと同時にページ・ヘッダーとしても使用する場合、PAGEHEAD を使用できます。

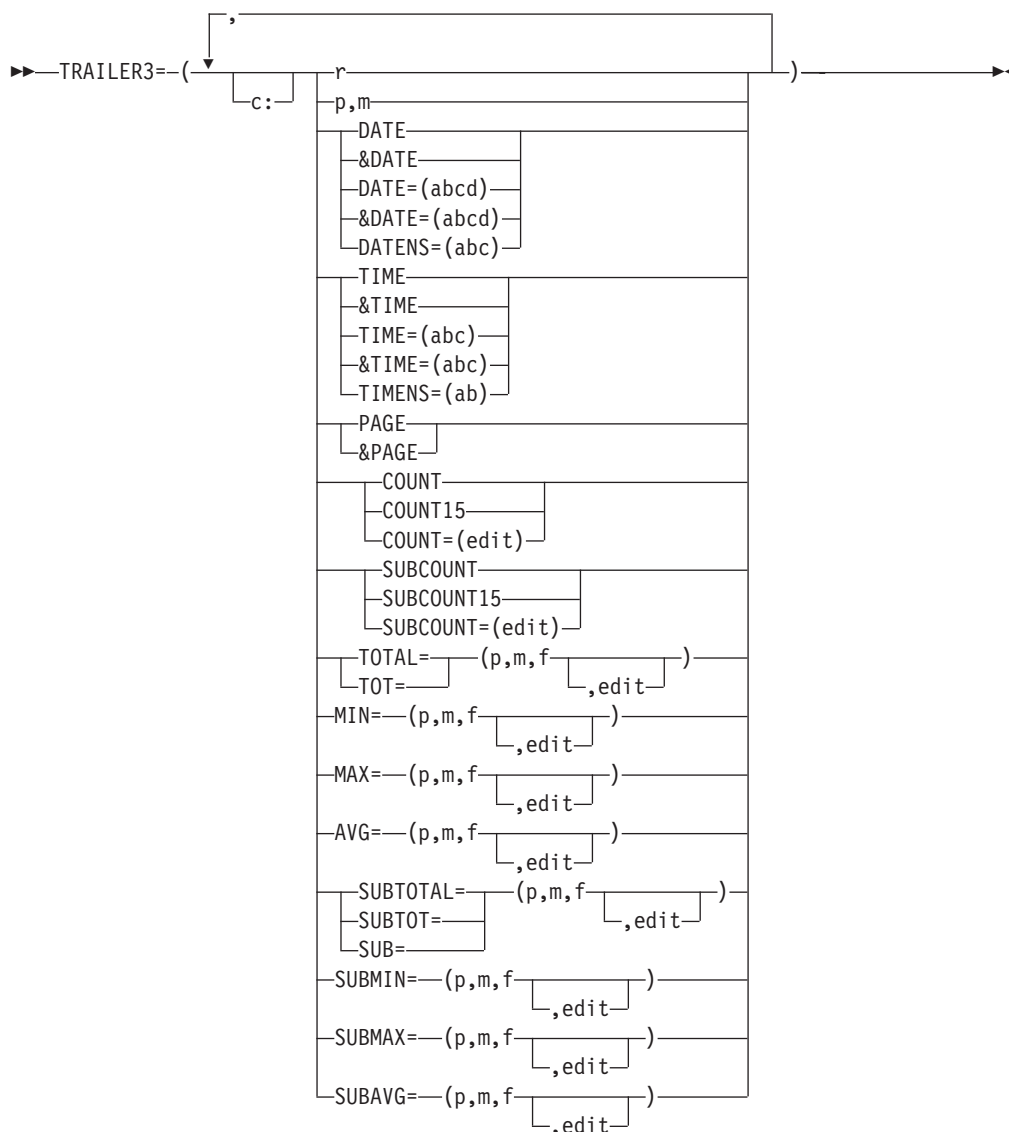
セクション段落フィールドに PAGEHEAD が指定されていても HEADER3 が指定されていない場合、PAGEHEAD は使用されません。

サンプル構文:

```
OUTFIL FNames=STATUS2,
  HEADER2=(10:'Status Report for all departments',5X,
           '- ',&PAGE,' -'),
  SECTIONS=(10,8,
    HEADER3=(2/,10:'Report for department ',10,8,' on ',&DATE,2/,
              10:'  Number',25:'Average Time',/,
              10:'Completed',25:'  (in days)',/,
              10:'-----',25:'-----'),
    PAGEHEAD,SKIP=P),
  OUTREC=(10:21,5,ZD,M10,LENGTH=9,
           25:38,4,ZD,EDIT=(III.T),LENGTH=12,
           132:X)
```

## OUTFIL 制御ステートメント

### TRAILER3



この OUTFIL グループについて作成される報告書の関連するセクション段落フィールドで、このセクション・トレーラーを使用するように指定します。セクション・トレーラーは、各セクションの最後のデータ・レコードの後に表示されません。DFSORT は ANSI 紙送り制御文字を使用して、報告書のページ替えや行配置をユーザーの指定に従って制御します。

セクション・トレーラーには、以下の報告書要素のいずれかまたは全部を組み込むよう選択できます。

- ブランクおよび文字ストリング
- データ・レコードがこのセクションに現れる最後の OUTFIL 入力レコードの未編集入力フィールド
- 現在の日付
- 現在の時間

- ページ番号
- 以下の統計のいずれかまたはすべて
  - このセクションにあるデータ・レコード数
  - 符号、10 進小数点、先行ゼロ、非先行ゼロなどを含むように編集されたセクションのデータ・レコードで指定された ZD、PD、BI、FI、または CSF/FS の各数値入力フィールドの合計、最小、最大、または平均。
  - 符号、10 進小数点、先行ゼロ、非先行ゼロなどを含むように編集された、この時点までのデータ・レコードの ZD、PD、BI、FI、または CSF/FS と指定された各数値入力フィールドについての実行合計、最小、最大、または平均。

セクション・トレーラーは、ユーザーが選択した要素を指定した順番で、指定した列または行に構成されます。

**c:** HEADER1 の項の c: を参照してください。

**r** TRAILER1 の項の r を参照してください。

**p,m**

データ・レコードがこのセクションに現れる最後の OUTFIL 入力レコードの未編集入力レコードが報告書レコードに表示されるように指定します。詳細については、TRAILER1 の項の p,m を参照してください。

**DATE**

HEADER1 の項の DATE を参照してください。

**&DATE**

&DATE を DATE の代わりに使用できます。HEADER1 の項の &DATE を参照してください。

**DATE=(abcd)**

HEADER1 の項の DATE=(abcd) を参照してください。

**&DATE=(abcd)**

DATE=(abcd) の代わりに、&DATE=(abcd) を使用できます。HEADER1 の項の &DATE=(abcd) を参照してください。

**DATENS=(abc)**

HEADER1 の項の DATENS=(abc) を参照してください。

**TIME**

HEADER1 の項の TIME を参照してください。

**&TIME**

&TIME を TIME の代わりに使用できます。HEADER1 の項の &TIME を参照してください。

**TIME=(abc)**

HEADER1 の項の TIME=(abc) を参照してください。

**&TIME=(abc)**

TIME=(abc) の代わりに、&TIME=(abc) を使用できます。HEADER1 の項の &TIME=(abc) を参照してください。

**TIMENS=(ab)**

HEADER1 の項の TIMENS=(ab) を参照してください。

## OUTFIL 制御ステートメント

### PAGE

TRAILER1 の項の PAGE を参照してください。

### &PAGE

&PAGE を PAGE の代わりに使用できます。TRAILER1 の項の &PAGE を参照してください。

### COUNT

セクションのデータ・レコード数が、8 桁、右寄せ、先行ゼロ抑制で報告書レコードに表示されるように指定します。たとえば、入力レコードが、最初のセクションに 40、2 番目のセクションに 40、3 番目のセクションに 26 ある場合、COUNT は、最初のセクションが ' 40'、2 番目のセクションが ' 40'、3 番目のセクションが ' 26' となります。

COUNT は入力レコードの数であり、データ・レコードの数ではありません。ただし、複数のレコードを作成するスラッシュ (/) が指定されていない限り、このレコード数はデータ・レコードの数も表します。

### COUNT15

カウントが 15 桁で表示されること以外は、COUNT と同じです。

### COUNT=(edit)

指定通りの編集で 15 桁のカウントが表示されること以外は、COUNT と同じです。使用できる編集フィールドの詳細については、OUTREC の項の p,m,f,edit を参照してください。

### SUBCOUNT

報告書のこの時点までの実行中入力レコード数が、8 桁、右寄せ、先行ゼロ抑制で報告書レコードに表示されるように指定します。実行数は、現行のセクションまでのすべてのセクションのレコード数の累積になります。たとえば、入力レコードが、最初のセクションに 40、2 番目のセクションに 40、3 番目のセクションに 26 ある場合、SUBCOUNT は、最初のセクションが ' 40'、2 番目のセクションが ' 80'、3 番目のセクションが ' 106' となります。

SUBCOUNT は入力レコードの数であり、データ・レコードの数ではありません。ただし、複数のレコードを作成するスラッシュ (/) が指定されていない限り、このレコード数はデータ・レコードの数も表します。

### SUBCOUNT15

実行中カウントが 15 桁で表示されること以外は、SUBCOUNT と同じです。

### SUBCOUNT=(edit)

指定通りの編集で 15 桁の実行中カウントが表示されること以外は、SUBCOUNT と同じです。使用できる編集フィールドの詳細については、OUTREC の項の p,m,f,edit を参照してください。

### TOTAL

セクション内のデータ・レコードの数値入力フィールドの値について、編集された合計が報告書レコードに表示されるように指定します。

TOT を TOTAL の代わりに使用できます。

### p,m,f,edit

TRAILER1 - TOTAL の項の p,m,f,edit を参照してください。

**MIN**

セクション内のデータ・レコードの数値入力フィールドの値について、編集された最小値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

TRAILER1 - MIN の項の p,m,f,edit を参照してください。

**MAX**

セクション内のデータ・レコードの数値入力フィールドの値について、編集された最大値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

TRAILER1 - MAX の項の p,m,f,edit を参照してください。

**AVG**

セクション内のデータ・レコードの数値入力フィールドの値について、編集された平均値が報告書レコードに表示されるように指定します。

**p,m,f,edit**

TRAILER1 - AVG の項の p,m,f,edit を参照してください。

**SUBTOTAL**

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行合計が報告書レコードに表示されるように指定します。実行合計は、現行のセクションまでのすべてのセクションのレコード数の累積になります。たとえば、選択された数値フィールドでの合計が、1 番目のセクションで +200、2 番目のセクションで -250、3 番目のセクションで +90 の場合、SUBTOTAL は 1 番目のセクションで +200、2 番目のセクションで -50、3 番目のセクションで +40 となります。

SUBTOT または SUB を SUBTOTAL の代わりに使用できます。

**p,m,f,edit**

TRAILER1 - SUBTOTAL の項の p,m,f,edit を参照してください。

**SUBMIN**

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行最小値が報告書レコードに表示されるように指定します。実行最小値には、現行のセクションまでのすべてのセクションから最小値が選択されます。たとえば、選択された数値フィールドでの最小値が、1 番目のセクションで +200、2 番目のセクションで -250、3 番目のセクションで +90 の場合、SUBMIN は 1 番目のセクションで +200、2 番目のセクションで -250、3 番目のセクションで -250 となります。

**p,m,f,edit**

TRAILER1 - SUBMIN の項の p,m,f,edit を参照してください。

**SUBMAX**

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行最大値が報告書レコードに表示されるように指定します。実行最大値には、現行のセクションまでのすべてのセクションから最大値が選択されます。たとえば、選択された数値フィールドでの最大値が、1 番目のセクションで -100、2 番目のセクションで +250、3 番目のセクションで +90 の場合、SUBMAX は 1 番目のセクションで -100、2 番目のセクションで +250、3 番目のセクションで +250 となります。

## OUTFIL 制御ステートメント

### p,m,f,edit

TRAILER1 - SUBMAX の項の p,m,f,edit を参照してください。

### SUBAVG

報告書のこの時点までのデータ・レコードの数値入力フィールドの値について、編集された実行平均値が報告書レコードに表示されるように指定します。実行平均値には、現行のセクションまでのすべてのセクションから平均値が算出されます。たとえば、データ・レコード数と選択された数値フィールドの合計値がそれぞれ、1 番目のセクションで 60 レコードと +2205、2 番目のセクションで 60 レコードと -6252、3 番目のセクションで 23 レコードと -320 の場合、SUBAVG は 1 番目のセクションで +36、2 番目のセクションで -33、3 番目のセクションで -30 となります。

### p,m,f,edit

TRAILER1 - SUBAVG の項の p,m,f,edit を参照してください。

サンプル構文:

```
OUTFIL FNames=SECRPT,
INCLUDE=(11,4,CH,EQ,C'SSD'),
SECTIONS=(3,5,SKIP=P,
HEADER3=(2:'Department: ',3,5,4X,'Date: ',&DATE,2/),
TRAILER3=(2/,2:'The average for ',3,5,' is ',
AVG=(40,3,PD,M12),/,
2:'The overall average so far is ',
SUBAVG=(40,3,PD,M12)),
45,8,SKIP=3L,
HEADER3=(4:'Week Ending ',45,8,2/,
4:'Item Number',20:'Completed',/,
4:'-',20:'-'),
TRAILER3=(4:'The item count for week ending ',45,8,
' is ',COUNT=(EDIT=(II,IIT))),
OUTREC=(11:16,4,22:40,3,PD,M12,120:X)
```

SECTIONS のデフォルト: なし。必ず指定する必要があります。

### NODETAIL

▶▶—NODETAIL—◀◀

この OUTFIL グループについて作成された報告書にデータ・レコードが出力されないように指定します。NODETAIL を用いた場合、データ・レコードは、入力フィールド、統計、レコード数、セクション段落などに関しては完全に処理されますが、OUTFIL データ・セットには書き込まれず、ページの終わりを判別するための行数にも含まれません。NODETAIL を使用すると、データ・レコードを実際に表示させることなく要約できます。

サンプル構文:

```
OUTFIL FNames=SUMMARY,NODETAIL,
HEADER2=(' Date: ',DATENS=(DMY.),4X,'Page: ',PAGE,2/,
10:'Division',25:' Total Revenue',/,
10:'-----',25:'-----'),
SECTIONS=(3,5,
TRAILER3=(10:3,5,
25:TOTAL=(25,4,FI,M19,
LENGTH=17))),
TRAILER1=(5/,10:'Summary of Revenue ',4/,
12:'Number of divisions reporting is ',
```



```
COUNT,/,
12:'Total revenue is ',
TOTAL=(25,4,FI,M19)
```

NODETAIL のデフォルト: なし。必ず指定する必要があります。

OUTFIL ステートメントのデフォルト: なし。必ず指定する必要があります。同一ソースおよび異なるソースに複数の OUTFIL ステートメントを指定できます。指定変更は DD 名のレベルで行われます。

OUTFIL ステートメントで適用可能な機能: 分類、組み合わせ、コピー。

## REMOVECC

▶—REMOVECC—◀

ANSI 紙送り制御文字が、レコードが書き込まれる前に、この OUTFIL グループの OUTFIL 出力レコードから削除されるように指定します。さらに、ページ・トレーラー (TRAILER2) をページの下部に位置付けるとき、ブランク行は使用されません。

REMOVECC が報告書パラメーターなしで指定されているときは、使用されません。報告書パラメーターは、LINES、HEADER1、TRAILER1、HEADER2、TRAILER2、SECTIONS、および NODETAIL です。

サンプル構文:

```
OUTFIL FNames=RPTWOCC,
TRAILER1=(3/,'Number of records is ',
COUNT=(M11,LENGTH=6)),
REMOVECC
```

REMOVECC のデフォルト: なし。必ず指定する必要があります。

## OUTFIL ステートメントに関する注意事項

- OUTFIL 処理は、分類アプリケーション、組み合わせアプリケーション、コピー・アプリケーションでサポートされますが、ブロック・セット手法によるのみサポートされています。
- 有効な ODMAXBF 値により、各 OUTFIL データ・セットに使用する最大バッファ空間が指定されます。ODMAXBF 値は、インストール・パラメーターまたは実行時パラメーターとして指定するか、あるいは ICEEXIT ルーチンで指定できます。有効な ODMAXBF オプションでは、デフォルトの 2 メガバイトが推奨されます。ODMAXBF 値を小さくすると、アプリケーションにおいてパフォーマンスの低下が起こることがありますが、OUTFIL 処理のために使用する記憶域量が問題と考えられる場合は、それが必要になります。ODMAXBF 値を大きくすると、アプリケーションの EXCP を向上させることができますが、必要記憶域量も増加させる可能性があります。
- OUTFIL 処理のために使用する記憶域は、使用可能な記憶域の合計、非 OUTFIL 処理に必要な記憶域、および OUTFIL データ・セットの数およびそれらの属性 (たとえば、ブロック・サイズ) にしたがって自動的に調整されます。OUTFIL 処理は、有効になっている ODMAXBF の限界およびシステム記憶域の制限 (たとえば IEFUSI) の影響を受けますが、DFSORT 記憶域限界 (つまり、SIZE、

## OUTFIL 制御ステートメント

MAXLIM、および TMAXLIM) の影響は受けません。DFSORT は、可能な限り、16 メガバイト境界より上の記憶域を、OUTFIL 処理に使用しようとしています。

- VSAM BSP オプションは、SORTOUT データ・セットには適用されますが、OUTFIL データ・セットには適用されません。OUTFIL データ・セットを処理する場合は、NOBLKSET オプションは無視されます。SORTOUT データ・セットの場合は E39 出口ルーチンに入ることができませんが、OUTFIL データ・セットの場合は入ることはできません。
- 固定形式 OUTFIL データ・セットの場合、DFSORT はグループの OUTREC レコード長または OUTFIL 入力レコード長 (OUTREC がこのグループで指定されていない場合) に基づいて、各 OUTFIL データ・セット LRECL を決定します。VSAM データ・セットの場合、クラスターで定義されている最大レコード・サイズは LRECL と同じです。

OUTFIL データ・セット LRECL が指定されていないかまたは使用できない場合は、DFSORT はそれを決定された LRECL に設定します。OUTFIL データ・セット LRECL が指定されているかまたは使用できる場合は、決定された LRECL より小さかったり、あるいは、OUTREC パラメーターが指定されている場合は、決定された LRECL より大きくてはなりません。つまり、LRECL 値は出力レコードの埋め込み、または OUTREC パラメーター処理により作成されるレコードの切り捨てに使用できません。

通常、OUTREC 処理はレコードの埋め込みや切り捨てに使用し、また、LRECL を指定したり、再フォーマット設定レコードの長さに設定したりしないでください。

- 可変形式 OUTFIL データ・セットの場合、DFSORT は、グループの OUTREC レコード長または OUTFIL 入力レコード長 (OUTREC がグループで指定されていない場合) に基づいて、各 OUTFIL データ・セットの LRECL 最大値を決定します。OUTFIL データ・セットの最大 LRECL が指定されていないかまたは使用できない場合は、DFSORT はそれを決定された LRECL 最大値に設定します。VSAM データ・セットの場合、クラスターで定義されている最大レコード・サイズは、LRECL 最大値より 4 バイト大きくなります。
- OUTFIL 報告書を作成する場合、データ・レコードの最大長または唯一のデータ・レコードの長さは、報告書の最大レコード長と同じかそれ以上である必要があります。OUTREC パラメーターを使用して、どの報告書レコードよりも長いデータ・レコード長を指定できます。その場合は、DFSORT に LRECL を計算して設定させるか、あるいは算出された LRECL が、既存のまたは指定の LRECL 値と等しいことを確認できます。ANSI 紙送り制御文字の場合、LRECL に余分のバイトを用意することを忘れないでください。

たとえば、データ・レコードが 40 バイトであるのに最長報告書レコードが 60 バイトである場合、次のような OUTREC パラメーターを使用できます。

```
OUTREC=(1,40,80:X)
```

すると、DFSORT は LRECL を 81 (ANSI 紙送り制御文字の 1 バイトとデータ・レコード長の 80 バイト) に設定して、右側のデータ・レコードをブランクで埋め込みます。

ANSI 紙送り制御文字を出力データ・セットで使用しない場合は、REMOVECC パラメーターで削除します。たとえば、次のように指定すると仮定します。

```
OUTREC=(1,40,80:X),REMOVECC
```

DFSORT は LRECL を 81 の代わりに 80 に設定し、ANSI 紙送り制御文字が書き込まれる前に各レコードから削除します。

使用しているプリンターで処理できる長さよりも長いレコードが入った OUTFIL 報告書を印刷すると、システム・エラーを起こす可能性があります。

- DFSORT は、可能な場合、適切な ANSI 紙送り制御文字 (たとえば、3 スペースに C'-) をヘッダー・レコードおよびトレーラー・レコードに使用して、書き出す報告書レコードの数を減らします。DFSORT は、常にシングル・スペース紙送り制御文字 (C' ) をデータ・レコードに使用します。これらの紙送り制御文字は、OUTFIL データ・セットを表示させたときに、(表示方法によりは) 見えない場合がありますが、報告書を印刷する場合はこれらが使用されます。表示のために報告書を作成し、ブランク行をヘッダーとトレーラーに表示したい場合は、n/ を使用する代わりに、ブランク行を指定してください。たとえば、次のように指定した場合、

```
OUTFIL FNAMES=RPT,
  HEADER2=(2/, 'start of header', 2/, 'next line')
```

プリンターではブランク行が挿入されますが、表示はされません。このため、以下のように指定します。

```
OUTFIL FNAMES=RPT,
  HEADER2=(X,/, X,/, 'start of header',/, X,/, 'next line')
```

ANSI 紙送り制御文字を出力データ・セットで使用しない場合は、REMOVECC パラメーターで削除します。

- OUTREC を用いて可変長 OUTFIL 出力レコードまたはデータ・レコードを定義する場合は、各レコードの始めに 4 バイトの RDW を明示指定する必要があります。OUTREC に / を使用するとき、各新規出力またはデータ・レコードの先頭に、4 バイトの RDW を明示指定します。  
可変長 OUTFIL ヘッダー・レコードまたはトレーラー・レコードを定義する場合は、レコードの始めに 4 バイトの RDW を指定できません。
- 可変長 OUTFIL レコードの場合、レコードの可変部を、INREC または OUTREC ステートメントの最後のフィールドとして指定するときは、すべての OUTFIL OUTREC レコードの最後のフィールドとして指定します。レコードの可変部を、INREC または OUTREC ステートメントの最後のフィールドとして指定しないときは、どの OUTFIL OUTREC レコードにも指定してはいけません。INREC および OUTREC ステートメントを指定しない場合は、レコードの可変部を、OUTFIL OUTREC レコードごとに指定したり、しなかったりできます。
- OUTFIL グループの OUTFIL 入力レコードがない場合は、データ・レコードなしにヘッダーおよびトレーラーが表示されます。指定された未編集入力フィールドにはブランクが使用され、指定されたすべての統計フィールドにはゼロの値が使用されます。
- 可変長 OUTFIL 入力レコードが短すぎて、報告書のヘッダーまたはトレーラーのために指定された、未編集入力フィールドを含めることができない場合は、欠落しているバイト部分にブランクが使用されます。可変長 OUTFIL 入力レコードが短すぎて、指定されたセクション段落フィールド、または統計フィールドを含めることができない場合は、欠落しているバイト部分に意図的に、または意図せずにゼロが使用されます。

## OUTFIL 制御ステートメント

- 可変長 OUTFIL 入力レコードが短すぎて、OUTFIL INCLUDE または OMIT 比較フィールドを含むことができない場合に、DFSORT が行う処置は VLSCMP/NOVLSCMP および VLSHRT/NOVLSHRT の設定により異なります。詳細については、181 ページの『OPTION 制御ステートメント』の VLSCMP および NOVLSCMP オプションの説明を参照してください。
- 可変長 OUTFIL 入力レコードが短すぎて、OUTFIL OUTREC フィールドを含むことができない場合、VLFILL=byte パラメーターが指定されていない限り、DFSORT は終了します。
- 可変長 OUTFIL 出力データ・レコードが長すぎて、その OUTFIL データ・セットの LRECL より長い場合に、DFSORT が行う処置は VLLONG/NOVLLONG の設定により異なります。詳細については、181 ページの『OPTION 制御ステートメント』の VLLONG および NOVLLONG オプションの説明を参照してください。VLLONG を使用すると、長い OUTFIL データ・レコードを切り捨てることができますが、長い OUTFIL ヘッダー・レコードやトレーラー・レコードには効果がありません。
- ページ番号が 6 桁 (PAGE、&PAGE) をオーバーフローするか、レコード数または実行レコード数が 15 桁 (COUNT、COUNT15、SUBCOUNT、SUBCOUNT15、AVG、SUBAVG) をオーバーフローするか、あるいは合計または実行合計が 15 桁 (TOTAL、SUBTOTAL、AVG、SUBAVG) をオーバーフローした場合、オーバーフローの値は、意図的にまたは意図せずに許容桁数まで切り捨てられます。
- 同一ソースおよび異なるソースに複数の OUTFIL ステートメントを指定できません。同一ソースに DD 名が複数回でてきた場合は、その DD 名はそれが最初に現れた OUTFIL と関連付けられます。たとえば、SYSIN に次のように指定すると、

```
OUTFIL FNAMES=(OUT1,OUT2),INCLUDE=(1,1,CH,EQ,C'A')
OUTFIL FNAMES=(OUT3,OUT1),SAVE
```

OUT1 および OUT2 は最初の OUTFIL グループの一部として (つまり、INCLUDE を用いて) 処理されます。OUT3 は、2 番目の OUTFIL グループの一部として (つまり、SAVE を用いて) 処理されますが、OUT1 は重複する DD 名であるため処理されません。

複数のソースに DD 名がある場合は、その DD 名はそれが現れた一番高いソース OUTFIL グループと関連付けられます。たとえば、DFSPARM に次のように指定すると、

```
OUTFIL FNAMES=(OUT1,OUT2),INCLUDE=(1,1,CH,EQ,C'A')
```

さらに SYSIN に次のように指定すると、

```
OUTFIL FNAMES=(OUT3,OUT1),SAVE
```

OUT1 および OUT2 は、DFSPARM OUTFIL グループの一部として (つまり、INCLUDE を用いて) 処理されます。OUT3 は、SYSIN OUTFIL グループの一部として (つまり、SAVE を用いて) 処理されますが、OUT1 は重複する DD 名であるため処理されません。

- OUTFIL ステートメントを EFS プログラムに渡したり、そのプログラムから戻すことはできません。D2 形式は、OUTFIL ステートメントの INCLUDE または OMIT パラメーターで指定できません。

## OUTFIL 制御ステートメント

- SZERO が有効な場合、数値フィールドと定数が使用されるとき、-0 は、+0 より小として比較されます。NOSZERO が有効な場合は、数値フィールドと定数が使用されるとき、-0 は、+0 と等しいとして比較されます。

注: OPTION SZERO または OPTION NOSZERO は、OUTFIL INCLUDE=(...) または OUTFIL OMIT=(...) について無視されます。OPTION ステートメントが、同じソースの OUTFIL ステートメントの前に「検出」された場合。たとえば、下記のようにします。

```
//SYSIN DD *  
OPTION NOSZERO,COPY  
OUTFIL INCLUDE=(...)  
/*
```

## OUTFIL 機能 — 例

### 例 1

```
OPTION COPY  
OUTFIL INCLUDE=(15,6,CH,EQ,C'MSG005'),FNAMES=M005  
OUTFIL INCLUDE=(15,6,CH,EQ,C'MSG022'),FNAMES=M022  
OUTFIL INCLUDE=(15,6,CH,EQ,C'MSG028'),FNAMES=M028  
OUTFIL INCLUDE=(15,6,CH,EQ,C'MSG115'),FNAMES=M115  
OUTFIL SAVE,FNAMES=UNKNOWN
```

この例では、ユーザーが指定する基準に基づいて、どのようにしてレコードを異なる OUTFIL データ・セットに書き込むかを示しています。

- 15 ~ 20 バイト目にある MSG005 をもつ入力レコードは、DD 名 M005 と関連付けられた OUTFIL データ・セットに書き込まれます。
- 15 ~ 20 バイト目にある MSG022 をもつ入力レコードは、DD 名 M022 と関連付けられた OUTFIL データ・セットに書き込まれます。
- 15 ~ 20 バイト目にある MSG028 をもつ入力レコードは、DD 名 M028 と関連付けられた OUTFIL データ・セットに書き込まれます。
- 15 ~ 20 バイト目にある MSG115 をもつ入力レコードは、DD 名 M115 と関連付けられた OUTFIL データ・セットに書き込まれます。
- 15 ~ 20 バイト目に上記以外をもつ入力レコードは、DD 名 UNKNOWN と関連付けられた OUTFIL データ・セットに書き込まれます。

### 例 2

```
SORT FIELDS=(18,5,ZD,D)  
OUTFIL FNAMES=(V,VBU1,VBU2)  
OUTFIL FNAMES=(F,FBU1),  
CONVERT,OUTREC=(11,3,X,18,5,X,X'0000000F')  
OUTFIL FNAMES=VINP,OUTREC=(1,4,C'*',5,20,C'*',25)
```

この例では、分類された複数の出力データ・セットが作成される方法と、可変長レコード・データ・セットが固定長レコード・データ・セットに変換される方法について示しています。

## OUTFIL 制御ステートメント

- 最初の OUTFIL ステートメントは、可変長入力レコードを、DD 名 V、VBU1、および VBU2 と関連付けられた可変長 OUTFIL データ・セットに書き込みます。
- 2 番目の OUTFIL ステートメントは、可変長入力レコードを固定長出力レコードに再フォーマット設定して、それを DD 名 F および FBU1 と関連付けられた固定長 OUTFIL データ・セットに書き込みます。 CONVERT は、可変長データ・セットを固定長データ・セットに変換することを示すために使用され、 OUTREC は、可変長入力レコードをどのようにして固定長出力レコードとして再フォーマット設定するかを示すために使用されます。
- 3 番目の OUTFIL ステートメントは、可変長入力レコードを再フォーマット設定して、それを DD 名 VINP と関連付けられた可変長 OUTFIL データ・セットに書き込みます。 OUTREC を使用して、フィールド間にアスタリスクを挿入します。 1,4 は RDW を表します。 25 は入力レコードの終端の可変部を表します。

### 例 3

```
SORT FIELDS=(15,6,ZD,A)
OUTFIL FNAMES=USA,
  HEADER2=(5:'Parts Completion Report for USA',2/,
    5:'Printed on ',DATE,
    ' at ',TIME=(12:),3/,
    5:'Part ',20:'Completed',35:' Value ($)',/,
    5:'-----',20:'-----',35:'-----'),
  OUTREC=(5:15,6,ZD,M11,
    20:3,4,ZD,M12,LENGTH=9,
    35:38,8,ZD,M18,LENGTH=12,
    132:X)
OUTFIL FNAMES=FRANCE,
  HEADER2=(5:'Parts Completion Report for France',2/,
    5:'Printed on ',DATE=(DM4/),
    ' at ',TIME=3/,
    5:'Part ',20:'Completed',35:' Value (F)',/,
    5:'-----',20:'-----',35:'-----'),
  OUTREC=(5:15,6,ZD,M11,
    20:3,4,ZD,M16,LENGTH=9,
    35:38,8,ZD,M22,LENGTH=12,
    132:X)
OUTFIL FNAMES=DENMARK,
  HEADER2=(5:'Parts Completion Report for Denmark',2/,
    5:'Printed on ',DATE=(DMY-),
    ' at ',TIME=(24.),3/,
    5:'Part ',20:'Completed',35:' Value (kr)',/,
    5:'-----',20:'-----',35:'-----'),
  OUTREC=(5:15,6,ZD,M11,
    20:3,4,ZD,M13,LENGTH=9,
    35:38,8,ZD,M19,LENGTH=12,
    132:X)
```

この例では、3 つの異なる国のための報告書が、分類された固定長入力レコードからどのようにして作成されるかを示しています。これらの報告書は、日付、時間、および数値の形式の指定方法だけが異なります。

- 最初の OUTFIL ステートメントでは、米国で一般的に使用されている日付、時間、および数値の形式をもつ報告書が作成されます。
- 2 番目の OUTFIL ステートメントでは、フランスで一般的に使用されている日付、時間、および数値の形式をもつ報告書が作成されます。

3. 3 番目の OUTFIL ステートメントでは、デンマークで一般的に使用されている日付、時間、および数値の形式をもつ報告書が作成されます。

もちろん、これらの 3 つの報告書はどれでも、3 つの OUTFIL ステートメントの代わりに単一の OUTFIL ステートメントを用いて、単独で作成できます。(この方法は、その国の指定ロケールに従った文字データを分類する場合に必要となる場合があります。)

**FNAMES** パラメーターでは、その報告書の固定長データ・セットに関連する DD 名 (USA、FRANCE、DENMARK) を指定します。

**HEADER2** パラメーターは、その報告書の各ページの一番上に現れるページ・ヘッダーを指定するもので、次のものから構成されています。

- 報告書を識別するテキスト行。報告書のすべての英文テキストは、その国の言語のテキストに置き換えることができます。
- ブランク行 (2/)。
- 日付と時間を示すテキスト行。DATE、DATE=(abcd)、TIME、および TIME=(abc) オペランドの変形オペランドを使用して、日付と時間が、その国で一般に使用されている形式で指定されます。
- 2 行 のブランク行 (3/)。
- データの列のヘッダーを示すテキスト行 2 行。適切な通貨記号をテキストに組み込むことができます。

**OUTREC** パラメーターは、3 つの列のデータが各入力レコードごとに次のように表示されるように指定します。

- M11 により指定されたパターンに従って、15 ~ 20 バイト目にある ZD 値を変形することにより作成された 6 バイトの編集済み数値。M11 は、先行ゼロをもつ整数を表示するためのパターンです。
- その国で一般に使用されている千単位区切り文字をもつ整数値のパターンに従って、3 ~ 6 バイト目にある ZD 値を変形することにより作成された 9 バイト (LENGTH=9) の編集済み数値。M12 では、千単位区切り文字としてコンマを使用します。M16 では、千単位区切り文字としてブランクを使用します。M13 では、千単位区切り文字としてピリオドを使用します。
- その国で一般に使用されている千単位区切り文字および小数点をもつ 10 進値のパターンに従って、38 ~ 45 バイト目にある ZD 値を変形することにより作成された 12 バイト (LENGTH=12) の編集済み数値。M18 では、千単位区切り文字にコンマ、小数点にピリオドを使用します。M22 では、千単位区切り文字にブランク、小数点にコンマを使用します。M19 では、千単位区切り文字にピリオド、小数点にコンマを使用します。

261 ページの表 38 に、選択可能な 27 個の事前定義編集マスク (M0 ~ M26) を示します。

132:X を OUTREC パラメーターの終わりに使用して、データ・レコードが報告書レコードよりも長くなるようにします。この結果、固定長 OUTFIL データ・セットでは、LRECL が 132 (ANSI 制御文字の 1 バイトとデータのための 131 バイト) になります。

## OUTFIL 制御ステートメント

3 つの報告書は次のようになります。

### Parts Completion Report for USA

Printed on 03/25/95 at 01:56:20 pm

Part	Completed	Value (\$)
000310	562	8,317.53
001184	1,234	23,456.78
029633	35	642.10
192199	3,150	121,934.65
821356	233	2,212.34

### Parts Completion Report for France

Printed on 25/03/1995 at 13:56:20

Part	Completed	Value (F)
000310	562	8 317,53
001184	1 234	23 456,78
029633	35	642,10
192199	3 150	121 934,65
821356	233	2 212,34

### Parts Completion Report for Denmark

Printed on 25-03-95 at 13.56.20

Part	Completed	Value (kr)
000310	562	8.317,53
001184	1.234	23.456,78
029633	35	642,10
192199	3.150	121.934,65
821356	233	2.212,34



## 例 4

```

SORT FIELDS=(3,10,A,16,13,A),FORMAT=CH
OUTFIL FNames=WEST,
INCLUDE=(42,6,CH,EQ,C'West'),
HEADER1=(5/,18:'    Western Region',3/,
          18:'Profit and Loss Report',3/,
          18:'    for ',&DATE,3/,
          18:'    Page',&PAGE),
OUTREC=(6:16,13,24:31,10,ZD,M5,LENGTH=20,75:X),
SECTIONS=(3,10,SKIP=P,
HEADER3=(2:'Division: ',3,10,5X,'Page:',&PAGE,2/,
          6:'Branch Office',24:'    Profit/(Loss)',/,
          6:'-----',24:'-----'),
TRAILER3=(6:'=====',24:'=====')/,
          6:'Total',24:TOTAL=(31,10,ZD,M5,LENGTH=20),/,
          6:'Lowest',24:MIN=(31,10,ZD,M5,LENGTH=20),/,
          6:'Highest',24:MAX=(31,10,ZD,M5,LENGTH=20),/,
          6:'Average',24:AVG=(31,10,ZD,M5,LENGTH=20),/,
          3/,2:'Average for all Branch Offices so far:',
          X,SUBAVG=(31,10,ZD,M5)),
TRAILER1=(8:'Page ',&PAGE,5X,'Date: ',&DATE,5/,
          8:'Total Number of Branch Offices Reporting: ',
          COUNT,2/,
          8:'Summary of Profit/(Loss) for all',
          'Western Division Branch Offices',2/,
          12:'Total:',
          22:TOTAL=(31,10,ZD,M5,LENGTH=20),/,
          12:'Lowest:',
          22:MIN=(31,10,ZD,M5,LENGTH=20),/,
          12:'Highest:',
          22:MAX=(31,10,ZD,M5,LENGTH=20),/,
          12:'Average:',
          22:AVG=(31,10,ZD,M5,LENGTH=20))

```

上記の例では、固定長入力レコードの分類済みサブセットから、ヘッダー・ページとトレーラー・ページ、およびデータ列のセクションをもつ報告書がどのように作成されるかを示しています。

**FNames** パラメーターでは、その報告書の固定長データ・セットに関連する DD 名 (WEST) を指定します。

**INCLUDE** パラメーターでは、報告書のために選択したレコードを指定します。

**HEADER1** パラメーターでは、報告書の 1 ページ目に現れる報告書ヘッダーを指定します。このヘッダーは、5 行のブランク行 (5/) と、その後の 4 行のテキスト (それぞれ、2 行のブランク行 (3/) により区切られている) から構成されています。テキストの最後の 2 行は、それぞれ日付 (&DATE) およびページ番号 (&PAGE) を示します。

**OUTREC** パラメーターでは、2 つのデータ列が、選択された各入力レコードごとに次のように表示されるように指定します。

- 入力レコードの 16 ~ 28 バイト目の文字ストリング。
- M5 により指定されたパターンに従って、31 ~ 40 バイト目にある ZD 値を変形することにより作成された 20 バイト (LENGTH=20) の編集済み数値。

## OUTFIL 制御ステートメント

SECTIONS パラメーターでは、セクション段落フィールド (3,10)、セクションの間のページ替え (SKIP=P)、各セクションの前に表示されるヘッダー (HEADER3)、さらに各セクションの後に表示されるトレーラー (TRAILER3) を指定します。セクション・ヘッダーは、ページ番号を示すテキスト行、ブランク行 (2)、およびデータ列のヘッダーを示す 2 行のテキストで構成されます。セクション・トレーラーは、データとトレーラーを区切るテキスト行、セクションのデータの合計 (TOTAL)、最小 (MIN)、最大 (MAX)、および平均 (AVG) を編集済み数値として示すテキスト行、2 行のブランク行、およびこの時点までの報告書の全データ・レコードについての実行平均 (SUBAVG) を示すテキスト行で構成されます。

TRAILER1 パラメーターは、報告書の最後のページに現れる報告書トレーラーを指定するもので、ページと日付を示すテキスト行、4 行のブランク行 (5)、報告書内のデータ・レコードの数を示すテキスト行、1 行のブランク行、1 行のテキスト行、1 行のブランク行、さらに報告書の全データの合計、最小、最大、および平均を編集済み数値として示したテキスト行で構成されます。

75:X を OUTREC パラメーターの終わりに使用して、データ・レコードが報告書レコードよりも長くなるようにします。この結果、固定長 OUTFIL データ・セットでは LRECL が 76 (ANSI 制御文字の 1 バイトとデータのための 75 バイト) になります。

報告書は、次のようになります。

```
Western Region

Profit and Loss Report

for 05/11/95

Page 1
```

```
Division: Chips      Page: 2

Branch Office      Profit/(Loss)
-----
Gilroy             554,843.42
Los Angeles        (22,340.14)
Morgan Hill        987,322.32
Oakland            234,124.32
San Francisco      (32,434.31)
San Jose           1,232,133.35
San Martin         889,022.03
=====
Total              3,842,670.99
Lowest             (32,434.31)
Highest            1,232,133.35
Average            548,952.99

Average for all Branch Offices so far:      548,952.99
```

Division: Ice Cream      Page:      3

Branch Office	Profit/(Loss)
-----	-----
Marin	542,341.23
Napa	857,342.83
San Francisco	922,312.45
San Jose	(234.55)
San Martin	1,003,467.30
=====	=====
Total	3,325,229.26
Lowest	(234.55)
Highest	1,003,467.30
Average	665,045.85

Average for all Branch Offices so far:      597,325.02

Division: Pretzels      Page:      4

Branch Office	Profit/(Loss)
-----	-----
Marin	5,343,323.44
Morgan Hill	843,843.40
Napa	5,312,348.56
San Francisco	5,412,300.05
San Jose	1,234,885.34
San Martin	(2,343.82)
=====	=====
Total	18,144,356.97
Lowest	(2,343.82)
Highest	5,412,300.05
Average	3,024,059.49

Average for all Branch Offices so far:      1,406,236.51

Page      5      Date: 05/11/95

Total Number of Branch Offices Reporting:      18

Summary of Profit/(Loss) for all Western Division Branch Offices

Total:	25,312,257.22
Lowest:	(32,434.31)
Highest:	5,412,300.05
Average:	1,406,236.51

## OUTFIL 制御ステートメント

### 例 5

```
SORT FIELDS=(6,5,CH,A)
OUTFIL FNames=STATUS,
HEADER2=(1:C'PAGE ',&PAGE,C' OF STATUS REPORT FOR ',&DATE,2/,
        6:C'ITEM ',16:C'STATUS      ',31:C'PARTS',/,
        6:C'-----',16:C'-----',31:C'-----'),
OUTREC=(1,4,
        10:6,5,
        20:14,1,CHANGE=(12,
                        C'1',C'SHIP',
                        C'2',C'HOLD',
                        C'3',C'TRANSFER'),
        NOMATCH=(C'*CHECK CODE*'),
        37:39,1,BI,M10,
        120:X)
```

この例では、参照テーブルを使用して、ページ・ヘッダーとデータ列をもつ報告書が、どのようにして分類済みの可変長入力レコードから作成されるかを示しています。

**FNames** パラメーターでは、その報告書の可変長データ・セットに関連する DD 名 (STATUS) を指定します。

**HEADER2** パラメーターは、ページ・ヘッダーが各ページの一番上に表示されるように指定するもので、ページ番号 (&PAGE) と日付 (&DATE) を示すテキスト行、ブランク行 (2)、およびデータ列のヘッダーを示す 2 行のテキスト行で構成されます。

**OUTREC** パラメーターでは、RDW および 3 つのデータ列が各入力レコードごとに次のように表示されるように指定します (可変長レコードでは、5 バイト目がデータの 1 バイト目となることに注意してください)。

- 入力レコードの 6 ~ 10 バイト目の文字ストリング。
- **CHANGE** で定義されたテーブルで入力レコードの 14 バイト目と一致するものを検索する (参照および変更) ことにより作成される文字ストリング。 **NOMATCH** は、14 バイト目が **CHANGE** テーブルのどの項目とも一致しない場合に使用される文字ストリングを示します。
- **M10** により指定されたパターンに従って、39 バイト目の **BI** 値を変換することにより作成された編集済み数値。

可変長入力レコードの場合に、**OUTREC** に **c:** 値を指定しているときは、**RDW** を説明しますが、ヘッダーやトレーラーについては説明する必要はありません。

**HEADER2** の 1 行目に **1:** を使用すると、それが最初のデータ・バイトから開始されます (これに対して、可変長レコードの場合は、**5:** を使用して、最初の **OUTREC** データ・バイトを指定する必要があります)。また、**ITEM** ヘッダーには **6:** が使われているので、ヘッダーとデータを列に配置するために、**10:** を **ITEM** データに使用する必要があります。

**120:X** を **OUTREC** パラメーターの終わりに使用して、データ・レコードが報告書レコードよりも長くなるようにします。この結果、可変長 **OUTFIL** データ・セットの **LRECL** 最大値は 121 (ANSI 制御文字の 1 バイトとデータのための最大 120 バイト) になります。

印刷される報告書の 1 ページ目は、次のようになります。

```

PAGE          1 OF STATUS REPORT FOR 05/12/95

ITEM          STATUS          PARTS
-----
00082        HOLD            36
00123        SHIP            106
00300        *CHECK CODE*          95
10321        TRANSFER         18
12140        SHIP            120
    
```

### 例 6

```

OPTION COPY
OUTFIL FNames=(PIPE1,PIPE2,PIPE3,PIPE4,PIPE5),SPLIT
    
```

この例では、一組の SmartBatch パイプの中で、どのようにして出力レコードをできるだけ均等に分割できるかを示しています。1 番目のレコードは PIPE1 に関連した書き込み機能に書き込まれ、2 番目のレコードは PIPE2 に、3 番目のレコードは PIPE3 に、4 番目のレコードは PIPE4 に、5 番目のレコードは PIPE5 に、6 番目のレコードは PIPE1 に、というように、すべてのレコードが書き込まれるまで続きます。

もちろん、レコードは、パイプと同様にデータ・セットにも書き込むことができます。

### 例 7

```

OPTION COPY
OUTFIL FNames=RANGE1,ENDREC=1000000
OUTFIL FNames=RANGE2,STARTREC=1000001,ENDREC=2000000
OUTFIL FNames=RANGE3,STARTREC=2000001,ENDREC=3000000
OUTFIL FNames=RANGE4,STARTREC=3000001,ENDREC=4000000
OUTFIL FNames=(RANGE5,EXTRA),STARTREC=4000001
    
```

この例では、特定範囲の出力レコードを、どのようにして異なる出力データ・セットに書き込むかを示しています。典型的なアプリケーションとしては、データベース区分化があります。

最初の 100 万レコードは、RANGE1 に関連するデータ・セットに書き込まれ、次の 100 万レコードは RANGE2 に、3 番目の 100 万レコードは RANGE3 に、4 番目の 100 万レコードは RANGE4 に書き込まれます。残りのレコードは、RANGE5 に関連するデータ・セットと EXTRA に関連するデータ・セットの両方に書き込まれます (この場合、SAVE と STARTREC=4000001 は同じ目的を果たします)。

OUTFIL の INCLUDE、OMIT、および SAVE パラメーターを使用して、ユーザーが指定する基準に基づいて別の出力データ・セットに書き込むレコードを選択することもできます。

## OUTFIL 制御ステートメント

### 例 8

```
OPTION COPY,Y2PAST
OUTFIL FNames=Y4,
      OUTREC=(1,19,
              21,2,PD0,M11,C'/', transform mm
              22,2,PD0,M11,C'/', transform dd
              20,2,Y2P,      transform yy to yyyy
              24,57)
```

この例では、P'yymmdd' (X'0yymmddC') 形式のパック 10 進日付フィールドを 20 ~ 23 バイト目にもつ既存のデータ・セットを、C'mm/dd/yyyy' 形式の文字日付フィールドを 20 ~ 29 バイト目にもつ新規データ・セットに変換する方法を示します。yy は 2 桁年を、yyyy は 4 桁年を、mm は月を、dd は日を、C は正の符号をそれぞれ表します。

入力データ・セットの LRECL は 80 であり、Y4 データ・セットの LRECL は 86 になります。

Y2PAST=26 オプションにより、2 桁年を 4 桁年に変換するために使用する、「世紀」ウィンドウが設定されます。現在の年が 1996 であれば、「世紀」ウィンドウは、1970~2069 に設定されます。この「世紀」ウィンドウを使用すると、入力フィールドと出力フィールドは、次のようになります。

Input Field (HEX)	Output Field (CH)
20	20
0020505F	05/05/2002
0950823C	08/23/1995
0980316C	03/16/1998
0000316F	03/16/2000

### 例 9

```
OPTION COPY,Y2PAST=1996
OUTFIL FNames=SPCL,
      OUTREC=(1,14, copy positions 1-14
              15,6,Y2T, transform yy to yyyy - allow blanks
              21,20) copy positions 21 - 40
```

この例では、C'yymmdd' 形式の文字日付フィールドとブランク特殊標識を 15 ~ 20 バイト目にもつ既存のデータ・セットを、C'yyymmdd' 形式の文字日付フィールドとブランク特殊標識を 15 ~ 22 バイト目にもつ新規データ・セットに変換する方法を示します。

入力データ・セットの LRECL は 40 であり、SPCL データ・セットの LRECL は 42 になります。

Y2PAST=1996 オプションにより、「世紀」ウィンドウは 1996-2095 に設定されます。「世紀」ウィンドウは、2 桁年の 4 桁年への変換には使用されますが、特殊ブランク標識には使用されません。

入力レコードが次のとおりなら、

```
MORGAN HILL          CA
SAN JOSE             960512 CA
BOCA RATON           000628 FL
DENVER               951115 CO
```

出力レコードは次のようになります。

```
MORGAN HILL          CA
SAN JOSE             19960512 CA
BOCA RATON           20000628 FL
DENVER               20951115 CO
```

## 例 10

```
OPTION COPY
OUTFIL FNames=ALL,OUTREC=(C'US ',1,10,C' is in ',11,15,/,
                          C'WW ',1,10,C' is in ',26,20,2/)
OUTFIL FNames=(US,WW),SPLIT,
              OUTREC=(1,10,C' is in ',11,15,/,
                      1,10,C' is in ',26,20)
```

この例では、各 OUTFIL 入力レコードから、複数の OUTFIL 出力レコードとブランク・レコードを作成する方法を示します。入力データ・セットは、LRECL が 50 で、次の 3 レコードを含みます。

```
Finance San Francisco Buenos Aires
Research New York Amsterdam
Marketing Los Angeles Mexico City
```

最初の OUTFIL ステートメントは、DD 名 ALL に関連したデータ・セットを作成します。このデータ・セットの LRECL は 40 になります (26,20 入力フィールドを含む、最長の出力レコードの長さ)。次に示すように、各入力レコードから、2 つのデータ・レコードと 2 つのブランク・レコードが出力されます。

### ALL データ・セット

```
US Finance is in San Francisco
WW Finance is in Buenos Aires
```

```
US Research is in New York
WW Research is in Amsterdam
```

```
US Marketing is in Los Angeles
WW Marketing is in Mexico City
```

2 番目の OUTFIL ステートメントは、DD 名 US および WW に関連した 2 つのデータ・セットを作成します。これらのデータ・セットの LRECL は 37 になります (26,20 入力フィールドを含む、最長の出力レコードの長さ)。各入力レコードから 2 つのデータ・レコードが出力されます。SPLIT を指定すると、最初のデータ・レコードが US データ・セットに書き込まれ、2 番目のデータ・レコードが WW データ・セットに書き込まれます。したがって、次に示すように、1 入力レコードから各 OUTFIL データ・セットに 1 レコードが作成されます。

### US データ・セット

## OUTFIL 制御ステートメント

```
Finance    is in San Francisco
Research   is in New York
Marketing  is in Los Angeles
```

### WW データ・セット

```
Finance    is in Buenos Aires
Research   is in Amsterdam
Marketing  is in Mexico City
```

### 例 11

```
Sort Fields=(6,3,CH,D)
OUTFIL FNames=SET60,OUTREC=(1,60),VLFILL=C' '
OUTFIL FNames=VARFIX,VTOF,OUTREC=(5,20,5X,28,20),VLFILL=C'*'
```

この例では、短すぎてすべての OUTFIL OUTREC フィールドを含むことができない可変長レコードを、正常に処理する方法を示します。

入力データ・セットは、RECFM=VB で LRECL=80 です。このデータ・セットのレコードは、15 ~ 75 バイトの可変長です。

最初の OUTFIL ステートメントは、DD 名 SET60 に関連したデータ・セットを作成します。このデータ・セットは、RECFM=VB で LRECL=60 です。このデータ・セットのレコードの長さは、すべて 60 です。1,60 フィールドにより、60 バイトより長いレコードは 60 バイトに切り捨てられます。VLFILL=C' ' が指定されているため、1,60 フィールドにより、60 バイトより短いレコードは、ブランク (C' ') を充てんバイトとして、60 バイトまで埋め込まれます。

**注:** VLFILL=byte が指定されていないと、この OUTFIL ステートメントはメッセージ ICE218A を発行して終了します。一部の入力レコードが短すぎて、OUTREC フィールドを含むことができないためです。

2 番目の OUTFIL ステートメントは、DD 名 VARFIX に関連したデータ・セットを作成します。このデータ・セットは、RECFM=FB で LRECL=45 になります。VTOF により、可変長入力レコードが、OUTREC が指定するフィールドに従って、固定長出力レコードに変換されます。VLFILL=C'\*' により、短い入力レコードが処理できます。OUTFIL OUTREC フィールドの欠落した各バイトは、アスタリスク (C'\*) 充てんバイトで置き換えられます。

#### 注:

1. VTOF の代わりに CONVERT を使用できます。
2. VLFILL=C'\*' は、VTOF または CONVERT の VLFILL=X'40' のデフォルトを指定変更します。

### 例 12

```
OPTION COPY
OUTFIL OUTREC=(SEQNUM,4,BI,Z,8,5,ZD,TO=PD,Z,
               31,2,PD,TO=FI,LENGTH=2)
```



## OUTFIL 制御ステートメント

この例は、シーケンス番号を生成する方法、ある数値形式の値を別の数値形式に変換する方法を表しています。

入力データ・セットの LRECL は 50 であり、SORTOUT データ・セットの LRECL は 11 になります。

OUTFIL ステートメントは、以下のフィールドをもつ出力レコードを作成します。

- 1 ~ 4 バイト目に、1 から始まり、1 ずつ増分される 2 進数のシーケンス番号。
- 5 バイト目に X'00'。
- 6 ~ 8 バイト目に、入力バイト 8 ~ 12 からの変換済み ZD フィールドを含む PD フィールド。
- 9 桁目に X'00'。
- 10 ~ 11 バイト目に、入力バイト 31 ~ 32 からの変換済み PD フィールドを含む FI フィールド。

### 例 13

```
SORT FIELDS=COPY
OUTFIL FNAMES=VAROUT1,FTOV
OUTFIL FNAMES=VAROUT2,FTOV,
      OUTREC=(20,8,35,10)
OUTFIL FNAMES=VAROUT3,FTOV,VLTRIM=X'40'
```

この例は、OUTFIL の FTOV パラメーターを使用して、固定長レコード・データ・セットを可変長レコード・データ・セットに変換するいくつかの方法を表しています。

入力データ・セットは、RECFM=FB で LRECL=60 です。

- 最初の OUTFIL ステートメントは、固定長入力データ・セットを DD 名 VAROUT1 に関連する可変長 OUTFIL データ・セットに変換します。VAROUT1 には、RECFM=VB および LRECL=64 が含まれます。すべてのレコードの長さは、64 バイト (4 バイトの RDW と 60 バイトの入力レコード) になります。
- 2 番目の OUTFIL ステートメントは、固定長入力データ・セットを DD 名 VAROUT2 に関連する可変長 OUTFIL データ・セットに変換します。OUTREC を使用して、出力レコードに対し 20 ~ 27 バイト目と 35 ~ 44 バイト目の 2 つの入力フィールドを選択します。VAROUT2 には、RECFM=VB および LRECL=22 が含まれます。すべてのレコードの長さは、22 バイト (4 バイトの RDW、8 バイトの入力フィールド、10 バイトの入力フィールド) になります。
- 3 番目の OUTFIL ステートメントは、固定長入力データ・セットを DD 名 VAROUT3 に関連する可変長 OUTFIL データ・セットに変換します。VAROUT3 には、RECFM=VB および LRECL=64 が含まれます。VLTRIM=X'40' を使用して、可変長出力レコードから後書きブランクを削除します。レコードの長さは、各レコードの後書きブランクの数に応じて、5 バイトから 64 バイトに変わります。

## OUTFIL 制御ステートメント

### 例 14

```
OPTION COPY
OUTFIL FNames=OUT1,OUTREC=(DATE1(/),X,TIME1(:),X,1,80)
OUTFIL FNames=OUT2,OUTREC=(DATE2P,TIME3P,1,80)
OUTFIL FNames=OUT3,OUTREC=(DATE3(.),X,TIME2,X,1,80)
```

この例は、レコードにタイム・スタンプを挿入するいくつかの異なる方法を表しています。

入力データ・セットは、RECFM=FB で LRECL=80 です。

最初の OUTFIL ステートメントは、DD 名 OUT1 に関連したデータ・セットを作成します。このデータ・セットは、LRECL=100 になります。出力レコードにはそれぞれ、元の入力レコード (80 バイト) の後に、形式 C'yyyy/mm/dd hh:mm:ss ' (20 バイト) の実行日時で構成される、タイム・スタンプが指定されます。

2 番目の OUTFIL ステートメントは、DD 名 OUT2 に関連したデータ・セットを作成します。このデータ・セットは、LRECL=86 になります。出力レコードにはそれぞれ、元の入力レコード (80 バイト) の後に、形式 P'yyyymm' (4 バイト) の実行日付、および形式 P'hh' (2 バイト) の実行時間で構成される、タイム・スタンプが指定されます。

3 番目の OUTFIL ステートメントは、DD 名 OUT3 に関連したデータ・セットを作成します。このデータ・セットは、LRECL=94 になります。出力レコードにはそれぞれ、元の入力レコード (80 バイト) の後に、形式 C'yyyy.ddd hhmm ' (14 バイト) の実行日時で構成される、タイム・スタンプが指定されます。

### 例 15

```
OPTION COPY
OUTREC FIELDS=(1,4,11,4,DT1,7,4,TM1,60:X)
OUTFIL NODETAIL,
TRAILER1=(//,
3:'Earliest SMF timestamp is ',
MIN=(5,14,ZD,EDIT=('TTTT/TT/TT TT:TT:TT')),/,
3:'Latest SMF timestamp is ',
MAX=(5,14,ZD,EDIT=('TTTT/TT/TT TT:TT:TT')))
```

この例は、SMF レコード群からの最も古い、または最新のタイム・スタンプの表示方法を表しています。

OUTREC ステートメントは、DT1 形式を使用して、入力バイト 11 ~ 15 の SMF 日付を 5 ~ 12 バイト目の Z'yyyymmdd' 値に変換し、TM1 形式を使用して、入力バイト 7 ~ 10 の SMF 時間を 13 ~ 18 バイト目の Z'hhmmss' 値に変換します。

OUTFIL ステートメントは、OUTREC により作成された 5 ~ 18 バイト目の Z'yyyymmddhhmmss' 値を使用して、最小 (最も古い) および最大 (最新) のタイム・スタンプを決定し、このタイム・スタンプをトレーラー・レコードに形式 C'yyyy/mm/dd hh:mm:ss' で表示します。

報告書は、次のようになります。

```
Earliest SMF timestamp is 2001/01/09 10:27:04
Latest SMF timestamp is 2001/04/24 06:13:22
```

### 例 16

```
SORT FIELDS=(1,20,BI,A)
OUTFIL FNAMES=FUPPER,OUTREC=(1,80,TRAN=LTOU)
OUTFIL FNAMES=FHEX,OUTREC=(1,80,HEX)
OUTFIL FNAMES=FTR,OUTREC=(1,80,TRAN=ALTSEQ)
ALTSEQ CODE=(005C)
```

この例は、固定長レコードの変換を表しています。小文字から大文字への変換、16進変換、ALTSEQ テーブルを使用した変換の 3 つです。

入力データには、RECFM = FB および LRECL = が含まれます。

最初の OUTFIL ステートメントは、DD 名 FUPPER に関連したデータ・セットを作成します。このデータ・セットには、RECFM = FB および LRECL = 80 が含まれます。1 バイト目から 80 バイト目の小文字の EBCDIC 文字 (a ~ z) すべてが、大文字の EBCDIC 文字 (A ~ Z) に変換されます。その他の文字は変更されません。たとえば、文字列 'san jose, ca 95193' は、'SAN JOSE, CA 95193' に変換されます。

2 番目の OUTFIL ステートメントは、DD 名 FHEX に関連したデータ・セットを作成します。このデータ・セットには、RECFM = FB および LRECL = 160 (2 \* 80 データ・バイト) が含まれます。1 バイト目から 80 バイト目の各バイトはそれぞれ、その 16 進数値を表す 2 バイトの値に変換されます。たとえば、'A12' の 3 文字は 'C1F1F2' の 6 文字に変換されます。

3 番目の OUTFIL ステートメントは、DD 名 FTR に関連したデータ・セットを作成します。このデータ・セットには、RECFM = FB および LRECL = 80 が含まれます。1 バイト目から 80 バイト目までの各 2 進ゼロ (X'00') はそれぞれ、'\*' (X'5C') に変換されます。その他の文字は変更されません。

**注:** ALTSEQ テーブルは、形式が AQ ではなく BI であるため、分類フィールドには使用されません。

### 例 17

```
OPTION COPY
OUTFIL FNAMES=VUPPER,OUTREC=(1,4,5,TRAN=UTOL)
OUTFIL FNAMES=VHEX,OUTREC=(1,4,5,HEX)
OUTFIL FNAMES=VTR,OUTREC=(1,4,5,TRAN=ALTSEQ)
ALTSEQ CODE=(F040,5C40)
```

この例は、可変長レコードの変換を表しています。大文字から小文字への変換、16進変換、ALTSEQ テーブルを使用した変換の 3 つです。

入力データには、RECFM = VB および LRECL = 5000 が含まれます。

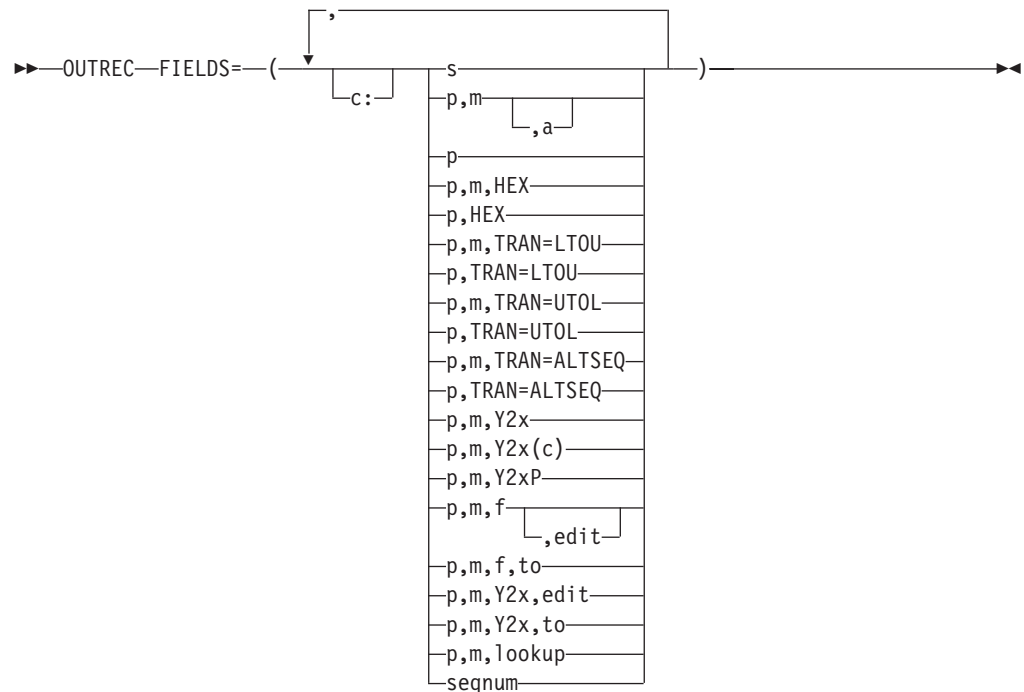
## OUTFIL 制御ステートメント

最初の OUTFIL ステートメントは、DD 名 VUPPER に関連したデータ・セットを作成します。このデータ・セットには、RECFM = VB および LRECL = 5000 が含まれます。5 バイト目 (RDW の後) から各レコードの終端までの、大文字の EBCDIC 文字 (A ~ Z) すべてが、小文字の EBCDIC 文字 (a ~ z) に変換されます。その他の文字は変更されません。たとえば、文字列 'SAN JOSE, CA 95193' は、'san jose, ca 95193' に変換されます。

2 番目の OUTFIL ステートメントは、DD 名 VHEX に関連したデータ・セットを作成します。このデータ・セットには、RECFM = VB および LRECL = 9996 (RDW の 4 バイト目と  $2 * 4996$  のデータ・バイト) が含まれます。5 バイト目 (RDW の後) から各レコードの終端までのバイトは、その 16 進数値を表す 2 バイトの値に変換されます。たとえば、'A12' の 3 文字は 'C1F1F2' の 6 文字に変換されます。

3 番目の OUTFIL ステートメントは、DD 名 VTR に関連したデータ・セットを作成します。このデータ・セットには、RECFM = VB および LRECL = 5000 が含まれます。5 バイト目 (RDW の後) から各レコードの終端までの '0' (X'F0') および '\*' (X'5C') 文字はそれぞれ、スペース (X'40') に変換されます。その他の文字は変更されません。

## OUTREC 制御ステートメント



OUTREC 制御ステートメントを使用すると、入力レコードが出力される前に、これを再フォーマット設定することができます。つまり、入力レコードのどの部分を再フォーマット設定済みの出力レコードに組み込むか、それらをどのような順序で表示するか、さらにそれらをどのように境界合わせするかを定義できます。

再フォーマット設定は、入力レコードから 1 つ以上のフィールドを定義することにより行います。再フォーマット設定された出力レコードは、これらのフィールドだけで構成されています。これらのフィールドはユーザーが指定した順序で配列され、ユーザーが示した境界または列に位置合わせされます。

OUTREC 制御ステートメントでは以下の機能も提供されます。

- 再フォーマット設定されたレコードの入力フィールドの前、入力フィールド間、および入力フィールドの後に、ブランク、ゼロ、ストリング、現在日付、現在時刻およびシーケンス番号を挿入する。
  - 16 進数表示、EBCDIC 文字の小文字から大文字へ、または大文字から小文字への変換、ALTSEQ 変換テーブルを使用した文字変換、ある形式から別の形式への数値変換などの、高度な変換機能。
  - 長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、数値フィールドの表示方法を制御するなどの、高度な編集機能。
- 一般に使用されている数値編集パターンとして 27 個の事前定義編集マスクが用意されており、世界中で使用されている数値表記の大部分を網羅しています。さらに、ユーザー定義の編集マスクにより、実際上無制限の数の数値編集パターンが利用できます。

## OUTREC 制御ステートメント

- SMF 日付値および時間値の、より有効なフォームへの変換。
- 指定された固定またはスライドの「世紀」ウィンドウを使用した、2 桁年の日付の各種形式から 4 桁年の日付の各種形式への変換。
- 入力としての文字、16 進数、またはビット・ストリングに基づいて、参照テーブルから文字または 16 進数ストリングを選択する (つまり、参照と変更)。

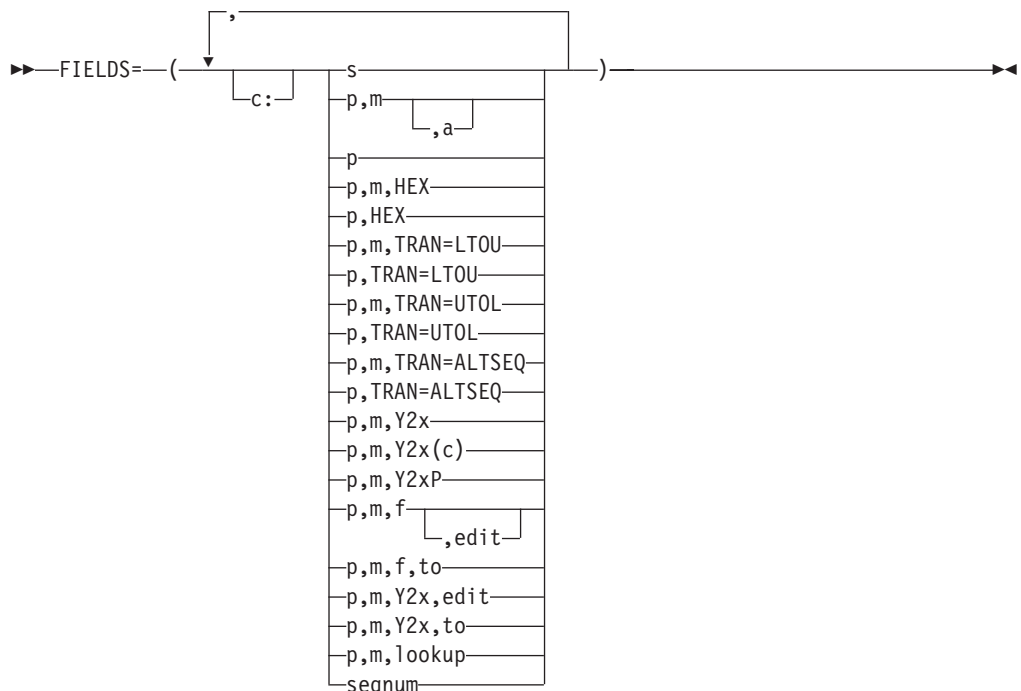
INREC および OUTREC の相互関係については、163 ページの『INREC ステートメントに関する注意事項』および 335 ページの『OUTREC ステートメントに関する注意事項』を参照してください。

OUTREC ステートメントと OUTFIL ステートメントの OUTREC パラメーターは、次の点で異なります。

- OUTREC ステートメントは、すべての入力レコードに適用されます。これに対し、OUTREC パラメーターは、その OUTFIL グループの OUTFIL 入力レコードにのみ適用されます。
- OUTREC パラメーターは、OUTREC ステートメントではサポートされない、ブランク・レコードおよび新規レコードの作成のための順方向のスラッシュ (/) 区切り文字がサポートされています。

OUTFIL OUTREC パラメーターの詳細については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

### FIELDS



- 再フォーマット設定された後の出力レコード内での、区切りフィールド (ブランク、ゼロ、ストリング、現在日付および現在時刻)、未編集および編集済みの入力フィールドおよびシーケンス番号の順序と境界合わせを指定します。

**c:**

区切りフィールド、入力フィールドまたはシーケンス番号の位置 (列) を、再フォーマット設定された出力レコードの先頭から相対的に指定します。指定された列の前にある未使用のスペースには、EBCDIC のブランクが埋め込まれます。次の規則が適用されます。

- c の値は 1 から 32752 の範囲で指定します。
- c: の後には、入力フィールドまたは区切りフィールドが続きます。
- c は、再フォーマット設定された出力レコードの、以前の入力フィールドまたは区切りフィールドと重なってはなりません。
- 可変長レコードの場合、最初の入力フィールド (レコード記述子ワード) の前にも、入力レコードの可変部の後にも、c: を指定できません。
- 次の行への継続する場合、コロン (;) はコンマ (,) またはセミコロン (;) と同様に取り扱われます。

桁合わせの有効な例と無効な例については、155 ページの表 21 を参照してください。

**s**

区切りフィールド (ブランク、ゼロ、文字ストリング、16 進数ストリング、現在日付、または現在時刻) が、再フォーマット設定された出力レコードに表示されるように指定します。任意の入力フィールドの前と後に指定できます。連続した区切りフィールドを指定できます。可変長レコードの場合、最初の入力フィールド (レコード記述子ワード) の前にも、入力レコードの可変部の後にも、区切りフィールドを指定できません。暗黙的値は、nX、nZ、nC'xx...x'、nX'yy...yy'、DATE1、DATE1(c)、DATE1P、DATE2、DATE2(c)、DATE2P、DATE3、DATE3(c)、DATE3P、TIME1、TIME1(c)、TIME1P、TIME2、TIME2(c)、TIME2P、TIME3、および TIME3P です。

**nX**

ブランクによる区切り。n バイトの EBCDIC ブランク (X'40') を再フォーマット設定された出力レコードに表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

ブランクによる区切りの有効な例および無効な例については、156 ページの表 22 を参照してください。

**nZ**

2 進ゼロによる区切り。n バイトの 2 進ゼロ (X'00') を再フォーマット設定された出力レコードに表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

2 進ゼロによる区切りの有効な例および無効な例については、156 ページの表 23 を参照してください。

**nC'xx...x'**

文字ストリングによる区切り。文字ストリング定数 (C'xx...x') の n 回の繰り返しを再フォーマット設定された後の出力レコード内に表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。x には任意の EBCDIC 文字を指定できます。1 ~ 256 文字を指定できます。

文字ストリングに単一のアポストロフィを組み込む場合は、それを 2 つの単一アポストロフィで指定します。

Required: O'NEILL      Specify: C'O'NEILL'

文字ストリングによる区切りの有効な例と無効な例については、157 ページの表 24 を参照してください。

**nX'yy...yy'**

16 進数ストリングによる区切り。 16 進数ストリング定数 (X'yy...yy') の n 回の繰り返しが再フォーマット設定された後の出力レコード内に表示します。n は 1 ~ 4095 の範囲の値になります。n が省略されていると、1 が使用されます。

値 yy は、16 進数の任意の組み合わせを示します。1 組から 256 組までの 16 進数字を指定できます。

16 進数ストリングによる区切りの有効な例と無効な例については、158 ページの表 25 を参照してください。

**DATE<sub>n</sub>, DATE<sub>n</sub>(c), DATE<sub>n</sub>P**

現在日付の定数。実行日付を再形成化された出力レコードに表示します。表 44 に、指定可能な区切りフィールドごとにそれぞれ生成された定数を、ストリング長および例とあわせて、関連箇所 (c) に対し (l) を使用して、示します。yyyy は年、mm は月 (01 ~ 12)、dd は日付 (01 ~ 31)、ddd は年間通日 (001 ~ 366) を表します。また、c は空白 以外 のあらゆる文字として使用できます。

表 44. 現在日付の定数

区切り フィールド	定数	長さ (バイト)	2001 年 4 月 19 日
DATE1	C'yyyymmdd'	8	C'20010419'
DATE1(c)	C'yyyymmdd'	10	C'2001/04/19'
DATE1P	P'yyyymmdd'	5	P'20010419'
DATE2	C'yyyymm'	6	C'200104'
DATE2(c)	C'yyyymm'	7	C'2001/04 '
DATE2P	P'yyyymm'	4	P'200104'
DATE3	C'yyyddd'	7	C'2001109'
DATE3(c)	C'yyyddd'	8	C'2001/109'
DATE3P	P'yyyddd'	4	P'2001109'

**TIMEn, TIMEn(c), TIMEnP**

現在時刻の定数。実行時間を再形成化された出力レコードに表示します。331 ページの表 45 に、指定可能な区切りフィールドごとにそれぞれ生成された定数を、ストリング長および例とあわせて、関連箇所 (c) に対し (:) を使用して、示します。hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59) を表します。また、c は空白 以外 のあらゆる文字として使用できます。



表 45. 現在時刻の定数

区切り フィールド	定数	長さ (バイト)	01:55:43 PM
TIME1	C'hmmss'	6	C'135543'
TIME1(c)	C'hcmmcss'	8	C'13:55:43'
TIME1P	P'hmmss'	4	P'135543'
TIME2	C'hmm'	4	C'1355'
TIME2(c)	C'hcmm '	5	C'13:55'
TIME2P	P'hmm'	3	P'1355'
TIME3	C'hh'	2	C'13'
TIME3P	P'hh'	2	P'13'

**p,m,a**

未編集の入力フィールドを再フォーマット設定された出力レコードに表示することを指定します。

**p** 入力フィールドの 1 バイト目が、入力レコードの先頭を基準にして、どの位置から始まるかを指定します。<sup>14</sup> 可変長レコードの最初のデータ・バイトは、相対位置が 5 になります。これは、最初の 4 バイトを RDW が占有するためです。すべてのフィールドはバイト境界から開始します。また、フィールドは 32752 バイトを超えることはできません。可変長レコードに関する特別な規則については、335 ページの『OUTREC ステートメントに関する注意事項』を参照してください。

**m**

入力フィールドの長さを指定します。データに符号がある場合は、その符号を含めます。また、全バイト数で指定します。詳細については、335 ページの『OUTREC ステートメントに関する注意事項』を参照してください。

**a** 再フォーマット設定された出力レコードの先頭を基準にした、再フォーマット設定された入力フィールドの境界合わせ (変位) を指定します。

次の値を **a** に指定できます。

**H** ハーフワード位置合わせ。これは再フォーマット設定された入力レコードの先頭からのフィールドの変位 (p-1) が、バイト単位で、2 の倍数 (つまり、1、3、5 桁目など) であることを意味します。

**F** フルワード位置合わせ。変位は、4 の倍数 (すなわち、1、5、9 桁目など) です。

**D** ダブル・ワード位置合わせ。変位は、8 の倍数 (すなわち、1、9、17 桁目など) です。

たとえば、COMPUTATIONAL 項目が SYNCHRONIZED 文節により境界合わせされている COBOL アプリケーション・プログラムでデータが使用さ

14. INREC が指定されている場合、p は、INREC により再フォーマット設定されたレコードを指す必要があります。E15 ユーザー出口でレコードを再フォーマット設定し、しかも INREC が指定されていない場合は、p は E15 ユーザー出口により再フォーマット設定されたレコードを参照します。

## OUTREC 制御ステートメント

れる場合は、境界合わせが必要となります。位置合わせされたフィールドの前にある未使用のスペースは、常に 2 進ゼロが埋め込まれます。

- p** 入力レコードの未編集の部分 (最小レコード長より大きい部分) が、再フォーマット設定された出力レコードに最後のフィールドとして表示されるように指定します。再フォーマット設定された出力レコードに RDW と入力レコードの変数部分だけを組み合わせると、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

p に指定される値は、最小レコード長 (RECORD ステートメント L4 値) に 1 バイトを加算した値以下である必要があります。

### **p,m,HEX**

入力フィールドの 16 進表示が、再フォーマット設定された出力レコードに表示されるように指定します。

詳細については、OUTFIL OUTREC の p,m,HEX の説明を参照してください。

### **p,HEX**

入力レコードの変数部の 16 進表示 (最小レコード長を超える部分) が、再フォーマット設定された出力レコードに最後のフィールドとして表示されるように指定します。再フォーマット設定された出力レコードに RDW と入力レコードの変数部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,HEX の説明を参照してください。

### **p,m,TRAN=LTOU**

入力フィールドの小文字の EBCDIC 文字 (すなわち、a ~ z) が、大文字の EBCDIC 文字 (すなわち、A ~ Z) として再フォーマット設定された出力レコードに表示されるように指定します。

詳細については、OUTFIL OUTREC の p,m,TRAN=LTOU の説明を参照してください。

### **p,TRAN=LTOU**

入力レコードの変数部 (最小レコード長を超える部分) の小文字の EBCDIC 文字 (すなわち、a ~ z) が、大文字の EBCDIC 文字 (すなわち、A ~ Z) として再フォーマット設定された出力レコードに最終フィールドとして表示されるように指定します。再フォーマット設定された出力レコードに RDW と入力レコードの変数部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,TRAN=LTOU の説明を参照してください。

### **p,m,TRAN=UTOL**

入力フィールドの大文字の EBCDIC 文字 (すなわち、A ~ Z) が、小文字の EBCDIC 文字 (すなわち、a ~ z) として、再フォーマット設定された出力レコードに表示されるように指定します。

詳細については、OUTFIL OUTREC の p,m,TRAN=UTOL の説明を参照してください。

### **p,TRAN=UTOL**

入力レコードの変数部 (最小レコード長を超える部分) の大文字の EBCDIC 文

## OUTREC 制御ステートメント

字 (すなわち、A ~ Z) が、小文字の EBCDIC 文字 (すなわち、a ~ z) として、再フォーマット設定された出力レコードに最終フィールドとして表示されるように指定します。再フォーマット設定された出力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,TRAN=UTOL の説明を参照してください。

### p,m,TRAN=ALTSEQ

入力フィールドの文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された出力レコードで変更されるように指定します。

詳細については、OUTFIL OUTREC の p,m,TRAN=ALTSEQ の説明を参照してください。

### p,TRAN=ALTSEQ

入力レコードの可変部 (最小レコード長を超える部分) の文字が、有効な ALTSEQ 変換テーブルに応じて、再フォーマット設定された出力レコードで、最終フィールドとして変更されるように指定します。再フォーマット設定された入力レコードに RDW と入力レコードの可変部分だけを組み合わせると、RDW だけが含まれる「ヌル」レコードになることに注意してください。

詳細については、OUTFIL OUTREC の p,TRAN=ALTSEQ の説明を参照してください。

### サンプル構文:

Fixed input records

```
OUTREC FIELDS=(1:5,10,15:8C'0',  
                25:20,15,TRAN=LTOU,80:X)
```

Variable input records

```
OUTREC FIELDS=(1,4,C' RDW=',1,4,HEX,C' FIXED=',  
                5,20,HEX,C' VARIABLE=',21,HEX)
```

### p,m,Y2x

2 桁年の入力日付フィールドの 4 桁年の CH 日付表記を、再フォーマット設定された出力レコードに表示するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2T は、C'000000' から C'00000000' に変換されます)。

詳細については、OUTFIL OUTREC の p,m,Y2x の説明を参照してください。

### サンプル構文:

```
OUTREC FIELDS=(21,3,Y2V,X,12,5,Y2W)
```

### p,m,Y2x(c)

2 桁年の入力日付フィールドの区切り文字付きの 4 桁の CH 日付表記を、再フォーマット設定された出力レコードに表示するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2T(/) は、C'000000' から C'0000/00/00' に変換されます)。

## OUTREC 制御ステートメント

詳細については、OUTFIL OUTREC の p,m,Y2x(c) の説明を参照してください。

サンプル構文:

```
OUTREC FIELDS=(25,6,Y2T(-),X,14,2,Y2U(/))
```

### p,m,Y2xP

2 桁年の入力日付フィールドの 4 桁年の PD 日付表記を、再フォーマット設定された出力レコードに表示するように指定します。有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して、実際の日付が変換されます。特殊標識には「世紀」ウィンドウは使用されません。特殊標識は適宜、展開されます (たとえば、p,6,Y2TP は、C'000000' から P'00000000' に変換されます)。

詳細については、OUTFIL OUTREC の p,m,Y2xP の説明を参照してください。

サンプル構文:

```
OUTREC FIELDS=(11,3,Y2XP,X,21,4,Y2WP)
```

### p,m,f,edit

編集済みの数値入力フィールドを再フォーマット設定された出力レコードに表示するように指定します。事前定義の編集マスク (M0 ~ M26) またはユーザーが定義する特定の編集パターンのいずれかを使用して、BI、FI、PD、PD0、ZD、CSF/FS、DT1、DT2、DT3、TM1、TM2、TM3 または TM4 フィールドを編集できます。長さ、先行ゼロまたは抑制ゼロ、千単位区切り文字、10 進小数点、先行および後書きの正負符号などについて、編集済みフィールドの表示方法を制御できます。

詳細については、OUTFIL OUTREC の項の p,m,f,edit の説明を参照してください。

サンプル構文:

```
OUTREC FIELDS=(5:21,8,ZD,M19,X,46,5,ZD,M13,  
31:35,6,FS,SIGNS=(,+,),LENGTH=10,  
51:8,4,PD,EDIT=(**II,IIT.TTXS),SIGNS=(,+,))
```

### p,m,f,to

変換済みの数値入力フィールドを再フォーマット設定された出力レコードに表示するように指定します。BI、FI、PD、PD0、ZD、CSF/FS、DT1、DT2、DT3、TM1、TM2、TM3、または TM4 フィールドを、BI、FI、PD、ZD、または CSF/FS フィールドに変換できます。

詳細については、OUTFIL OUTREC の p,m,f の説明を参照してください。

サンプル構文:

```
OUTREC FIELDS=(21,5,ZD,TO=PD,X,8,4,ZD,FI,LENGTH=2)
```

### p,m,Y2x,edit

2 桁年の入力日付フィールドの編集済みの 4 桁年の CH 日付表記を、再フォーマット設定された出力レコードに表示するように指定します。

詳細については、OUTFIL OUTREC の p,m,Y2x,edit の説明を参照してください。

**p,m,Y2x,to**

2桁年の入力日付フィールドの変換済み 4桁年の日付表記を、再フォーマット設定された出力レコードに表示するように指定します。

詳細については、OUTFIL OUTREC の p,m,Y2x の説明を参照してください。

**p,m,lookup**

参照テーブルの文字または 16進数ストリングが再フォーマット設定された出力レコードに表示されるように指定します。p,m,lookup を使用して、文字、16進数、またはビット定数に対する入力値の突き合わせに基づく、指定の文字または 16進数ストリングを選択できます。

詳細については、OUTFIL OUTREC の p,m,lookup の説明を参照してください。

サンプル構文:

```
OUTREC FIELDS=(11,1,
                CHANGE=(6,
                        C'R',C'READ',
                        C'U',C'UPDATE',
                        X'FF',C'EMPTY',
                        C'A',C'ALTER'),
                NOMATCH=(11,6),
                4X,
                21,1,
                CHANGE=(10,
                        B'.1.....',C'VSAM',
                        B'.0.....',C'NON-VSAM'))
```

**seqnum**

シーケンス番号を再フォーマット設定された出力レコードに表示するように指定します。シーケンス番号は、OUTREC 処理でレコードが受け取られる順序で割り当てられます。BI、PD、ZD、または CSF/FS シーケンス番号を作成し、それらの長さ、開始値および増分値を制御できます。

詳細については、OUTFIL OUTREC の seqnum の説明を参照してください。

サンプル構文:

```
OUTREC FIELDS=(SEQNUM,6,ZD,START=1000,INCR=50,1,60)
```

デフォルト: なし。必須のパラメーターです。指定変更の詳細については、697ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

**OUTREC ステートメントに関する注意事項**

- 入力レコードが INREC または E15 により再フォーマット設定される場合は、OUTREC は、該当する再フォーマット設定されたレコードのフィールドを参照します (163ページの『INREC ステートメントに関する注意事項』を参照)。
- OUTREC を指定するときは、再フォーマット設定された出力レコードのレコード・サイズとレイアウトの変化を把握しておく必要があります。
- SORTOUT LRECL が指定されているか使用可能な場合、再フォーマット設定された OUTREC レコード長にその値が一致しなくとも、DFSORT は SORTOUT

## OUTREC 制御ステートメント

LRECL を使用します。これにより、再フォーマット設定された OUTREC レコードの埋め込みまたは切り捨て、または終了が発生します。SORTOUT LRECL が指定されていないか使用可能でない場合、該当すれば、DFSORT が自動的に、SORTOUT LRECL として再フォーマット設定された OUTREC レコード長を使用します。詳細については、181 ページの『OPTION 制御ステートメント』の SOLRF および NOSOLRF オプションの説明を参照してください。

VSAM データ・セットの場合は、クラスター内に定義されている最大のレコード・サイズは、固定長レコードを処理するときの LRECL に等しく、可変長レコードを処理するときの LRECL よりも 4 バイト長くなります。詳細については、16 ページの『VSAM に関する考慮事項』を参照してください。

- 可変長レコードの場合は、FIELDS パラメーターの最初の項目には 4 バイトの RDW を指定するか、または組み込む必要があります。DFSORT は、再フォーマット設定されたレコードの長さを RDW で設定します。

入力レコードのデータ部分の最初のフィールドが、再フォーマット設定された後の出力レコード内で RDW の直後にある場合、FIELDS パラメーターの項目には、RDW とデータ・フィールドをまとめて指定できます。それ以外の場合は、RDW を再フォーマット設定された出力レコード内に必ず組み込む必要があります。

- 入力レコードの可変部 (最小レコード長を超える部分) を再フォーマット設定された出力レコードの最後の部分として組み込むことができます。この場合、*pn* には最小レコード長 (RECORD ステートメントの L4 値) に 1 バイトを加えた値以下の値を指定し、*mn* および *an* は省略します。INREC と OUTREC の両方を指定する場合は、その最後の部分について両方とも位置だけを指定するか、あるいは両方とも位置だけを指定しないかのどちらかである必要があります。

再フォーマット設定された入力に RDW と入力レコードの可変部分だけを組み合わせると、RDW だけを含む「ヌル」レコードになる場合があることに注意してください。

- 再フォーマット設定された出力レコードは、INREC が指定されているかどうかにかかわらず、OUTREC で指定した形式になります。
- OUTREC ステートメントで参照される各フィールドは、そのフィールド同士や制御フィールドと重なり合うことができます。
- 入力が可変レコードであれば、出力も可変になります。つまり、DFSORT が出力前に各レコードに正しい RDW を与えます。
- OUTREC が指定されている場合は、E35 ユーザー出口ルーチンは、再フォーマット設定された出力レコードのフィールドを参照します。
- テープ作業データ・セットの分類または従来の組み合わせアプリケーションの場合、OUTREC ステートメントが指定されると、DFSORT はメッセージを出して処理を終了します。
- OUTREC を指定した場合、VLSHRT は使用されません。VLSHRT を指定しても無視されます。
- SZERO が有効な場合、編集済みまたは変換済みの入力フィールドについて、-0 は負として扱われ、+0 は正として扱われます。NOSZERO が有効な場合は、編集済みまたは変換済みの入力フィールドについて、-0 および +0 は正として扱われます。

## 出力レコードの再フォーマット設定 - 例

165 ページの『処理前のレコードの再フォーマット設定 - 例』を参照してください。例 1、例 3、および例 4 は、INREC および OUTREC ステートメントがともに同じアプリケーションで使用されるアプリケーションを示しています。

### 例 1

```
OUTREC FIELDS=(11,32)
```

このステートメントは、入力レコードの 11 バイト目から始まる 32 バイト分を出力レコードに含めることを指定します。このステートメントは、最初の 4 バイトを含んでいないため、固定長入力レコードの場合しか使用できません。

### 例 2

```
OUTREC FIELDS=(1,4,11,32,D,101)
```

このステートメントは、最小の長さが 100 バイトの可変長レコードに関するもので、RDW と入力レコードの 11 バイト目から始まる 32 バイト分 (レコードの先頭を基準にして、ダブルワード境界合わせされる) に入力レコードの可変部分全体を加えたものを、出力レコードに含めることを指定します。

最初の境界合わせパラメーターが省略されたことを示す余分なコンマがコーディングされていないことに注意してください。余分なコンマを組み込むと、DFSORT はメッセージを出して処理を終了させてしまいます。

### 例 3

```
OUTREC FIELDS=(1,42,D,101)
```

このステートメントは最小の長さが 100 バイトの可変長レコードに関するもので、RDW と入力レコードの最初の 38 データ・バイトと入力レコードの全体の変数部分を加えたものを、出力レコードに含むことを指定します。

最初のフィールドは必ず出力レコードの先頭になるため、'D' パラメーターは有効ではありません。

### 例 4

```
SORT FIELDS=(20,4,CH,D,10,3,CH,D)
OUTREC FIELDS=(7:20,4,C' FUTURE ',20,2,10,3,1Z,1,9,13,7,
24,57,TRAN=LTOU,6X'FF')
```

この例は、固定長入力データ・セットを分類し、出力用に再フォーマット設定する方法を示したものです。SORTIN LRECL は 80 バイトです。

再フォーマット設定された出力レコードは固定長で、レコード・サイズは 103 バイトです。SOLRF (IBM 提供のデフォルト) が有効なので、SORTOUT LRECL が指

## OUTREC 制御ステートメント

定されているか使用可能である場合を除き、103 の再フォーマット設定されたレコード長に自動的に設定されます。再フォーマット設定されたレコードは、次のようになります。

位置	内容
1-6	EBCDIC ブランク (桁合わせのための)
7-10	入力位置 20 ~ 23
11-18	文字ストリング: 'C' FUTURE '
19-20	入力位置 20 ~ 21
21-23	入力位置 10 ~ 12
24	2 進ゼロ
25-33	入力位置 1 ~ 9
34-40	入力位置 13 ~ 19
41-97	EBCDIC 文字が小文字から大文字に変換された入力位置 24 ~ 80
98-103	16 進数ストリング: 'XXXXXXXXXXXX'

### 例 5

```
SORT FIELDS=(12,4,PD,D)
RECORD TYPE=V,LENGTH=(,,100)
OUTREC FIELDS=(1,7,5Z,5X,28,8,6X,101)
```

この例は、可変長入力データ・セットを分類し、出力用に再フォーマット設定する方法を示したものです。入力レコードの可変部は出力レコードに組み込まれます。最小入力レコード・サイズは 100 バイトで、最大入力レコード・サイズ (SORTIN LRECL または VSAM では最大レコード・サイズ) は 200 バイトです。

再フォーマット設定された出力レコードは、最大レコード・サイズが 131 バイトの可変長です。再フォーマット設定されたレコードは、次のようになります。

位置	内容
1-4	RDW (入力位置 1 ~ 4)
5-7	入力位置 5 ~ 7
8-12	2 進ゼロ
13-17	EBCDIC ブランク
18-25	入力位置 28 ~ 35
26-31	EBCDIC ブランク
32-n	入力位置 101 から n (入力レコードの可変部分)

### 例 6

```
MERGE FIELDS=(28,4,BI,A)
OUTREC FIELDS=(1,4,5Z,5X,5,3,28,8,6Z,DATE3,TIME1)
```

この例は、入力ファイルを組み合わせ、再フォーマット設定して、現在日付と現在時刻を含めて出力する方法を示しています。入力レコードの可変部は出力レコードに組み込まれません。SORTINnn LRECL は 50 バイトです。

再フォーマット設定された出力レコードは、最大レコード・サイズが 44 バイトの可変長です。再フォーマット設定されたレコードは、次のようになります。



位置	内容
1-4	RDW (入力位置 1 ~ 4)
5-9	2 進ゼロ
10-14	EBCDIC ブランク
15-17	入力位置 5 ~ 7
18-25	入力位置 28 ~ 35
26-31	2 進ゼロ
32-38	形式 'C'yyyyddd' で示された現在日付
39-44	形式 'C'hmmss' で示された現在時刻

### 例 7

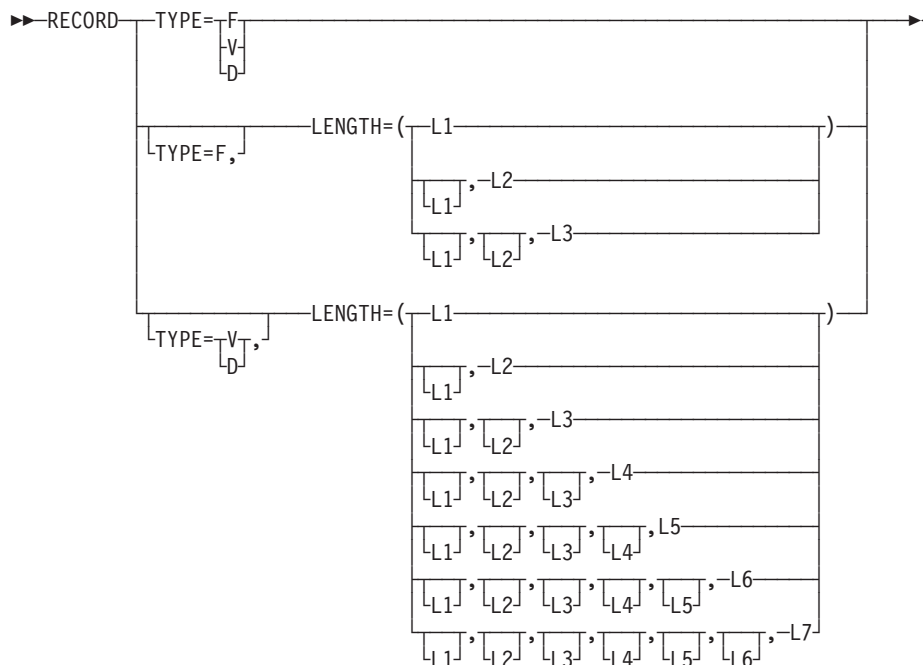
```
OPTION COPY,Y2PAST=1985
OUTREC FIELDS=(SEQNUM,8,ZD,START=1000,INCR=100,
               11:8,4,PD,M12,
               31:15,4,Y2V(/),
               51:2,1,CHANGE=(3,
               X'01',C'L92',X'02',C'M72',X'03',C'J42'),
               NOMATCH=(C'???'))
```

この例は、シーケンス番号を生成する方法、数値形式および日付形式の値を編集する方法、および参照テーブルの使用方法を表しています。

再フォーマット設定された出力レコードは、次のようになります。

位置	内容
1-8	1000 から始まり、100 ずつ増分されるゾーン 10 進数のシーケンス番号。
11-20	M12 編集マスクに従って編集された入力位置 8 ~ 11 からの PD フィールドを含む CH フィールド。
31-40	1985 ~ 2084 の指定された「世紀」ウィンドウに従って変換された、入力位置 15 ~ 18 からの P'yymmdd' 日付フィールドを含む C'yyyy/mm/dd' 日付フィールド。
51-53	C'L92'、C'M72'、C'J42'、または C'???' を含む CH フィールド (2 桁目の入力フィールドの参照テーブルを使用して判別したもの)。

## RECORD 制御ステートメント



RECORD 制御ステートメントを使用して、処理するレコードのタイプと長さ、および可変長分類の最小レコード長と平均レコード長を指定できます。

RECORD 制御ステートメントは、次のような場合に必要です。

- ユーザー出口でレコード長が変更される。
- ユーザー出口ですべての入力レコードが提供される。
- 従来の組み合わせやテープ作業データ・セットの分類で、VSAM 入力を使用される。

### TYPE

TYPE=x

入力が VSAM であるときや、E15 または E32 出口がすべての入力レコードを提供するとき、レコード・タイプを指定できます。レコード・タイプは次の値にできます。

- **固定長 (F)**。固定長レコードは RDW なしで処理されるため、データは 1 桁目から始まります。必要に応じて、制御ステートメント位置を指定する必要があります。

RRDS は常に、固定長として処理できます。入力に使用される KSDS、ESDS、または VRRDS は、そのレコードのすべての長さが、クラスターに定義された最大レコード・サイズに等しい場合のみ、固定長として処理する必要があります。そうでなければ、最大レコード・サイズより短い入力レコードは、ゼロのバイト (つまり、「ガーベッジ」バイト) か、ゼロでないバイトで埋め込まれます。

## RECORD 制御ステートメント

- **可変長 (V)**。可変長レコードは、1 ~ 4 桁目の RDW から処理されるため、データは 5 桁目から始まります。必要に応じて、制御ステートメント位置を指定する必要があります。

RRDS、KSDS、ESDS、または VRRDS は常に可変長として処理できます。

VSAM 入力データの場合、DFSORT は、各レコードを読み取り、その RDW を作成します。VSAM 出力の場合、DFSORT は各レコードを書き込む前に RDW を除去します。

TYPE が必要なのは、VSAM 入力を使用するか、すべての入力レコードを提供する E15 または E32 出口を使用する、従来の組み合わせまたはテープ作業データ・セット分類の場合のみです。

入力が非 VSAM である場合は、DFSORT は、入力データ・セットの RECFM からレコード・タイプを判別し、TYPE を無視します。

入力が VSAM であるときや、E15 または E32 出口がすべての入力レコードを提供するとき、DFSORT は、リストされている順序で情報を使用して、次のように、レコード・タイプを判別または割り当てることができます。

1. 指定されている場合、RECORD TYPE から F または V。
2. 使用可能な場合、SORTOUT RECFM から F または V。
3. OUTFIL VTOF、CONVERT または VLFILL が指定されている場合、V。または、OUTFIL FTOV が指定されている場合、F。
4. 使用可能な場合、OUTFIL RECFM から F または V。
5. SORTIN が VSAM で、SORTOUT が VSAM である場合、V。そうでなければ、F。

### 注:

- a. 選択されたレコード・タイプが、DFSORT に使用させたいものでない場合は、適宜、RECORD TYPE=F または RECORD TYPE=V を指定してください。
- b. 従来の組み合わせ、またはテープ作業データ・セット分類の場合、適宜、RECORD TYPE=F または RECORD TYPE=V を指定する必要があります。

x には次のいずれかの値を指定できます。

### F 固定長レコード処理。

注: F の代わりに、FB を使用できます。

### V 可変長レコード処理。

注: V の代わりに VB を使用できます。

### D ISCI/ASCII 可変長レコード処理。

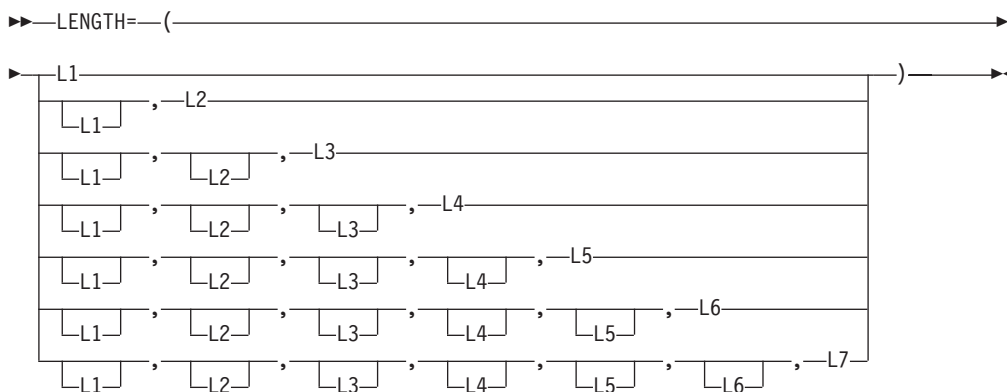
注: D の代わりに、DB を使用できます。

デフォルト: 上記のとおり、F または V。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## RECORD 制御ステートメント

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### LENGTH



様々なレコード長を指定するために使用できます。L1 ~ L3 は、固定長レコード処理と可変長レコード処理に適用されます。L4 と L5 は、可変長レコード処理に適用されます。L6 と L7 は受け入れられますが、使用されません。

LENGTH は、以下の場合のみ必要です。

- ユーザー出口でレコード長が変更される。
- ユーザー出口ですべての入力レコードが提供される。

#### L1

入力レコード長。可変長レコードの場合は、最大入力レコード長です。

##### 注:

1. SORTIN からの入力レコード長が使用できる場合、L1 は無視されます。
2. L2 が指定されていない場合に、SORTIN または SORTINnn データ・セットがなければ、L1 が必要です。

デフォルト: SORTIN または SORTINnn レコード長。VSAM データ・セットの場合は、最大レコード・サイズ (RECSZ 値) です。

#### L2

E15 以後のレコード長。可変長レコードの場合は、E15 以後の最大レコード長です。

##### 注:

1. E15 が使用されない場合、L2 は無視されます。
2. E15 でレコード長を変更する場合、L2 について正確な値を指定します。
3. テープ作業データ・セットを使用する場合には、L2 に少なくとも 18 バイトを指定します。
4. L1 が指定されていない場合に、SORTIN または SORTINnn データ・セットがなければ、L2 は無視されます。

デフォルト: L1。

**L3**

出力レコード長。可変長レコードの場合、最大出力レコード長。

**注:** レコード長 (LRECL または VSAM RECSZ) が SORTOUT から使用できる場合、あるいは、NOSOLRF が有効で、E35、INREC、OUTREC、および OUTFIL が使用されない場合、L3 は無視されます。

デフォルト: 以下のいずれか (リストされている順序)。

1. SORTOUT レコード長 (使用できる場合)
2. OUTREC レコード長 (SOLRF が有効な場合)
3. INREC レコード長 (SOLRF が有効な場合)
4. L2 (E15 が使用され、指定されている場合)
5. SORTIN または SORTINnn レコード長 (使用できる場合)
6. L1

**L4**

最小レコード長。

**注:**

1. ブロック・セット手法が選択された場合、L4 は使用されません。
2. L4 は、可変長レコード分類のアプリケーションの場合のみ使用されません。
3. L4 を指定すると、パフォーマンスを改善できますが、L4 が大きすぎると、DFSORT は失敗し、メッセージ ICE015A が出されます。

デフォルト: すべての制御フィールドを含むために必要な最小の長さ。最大入力レコード長が 18 バイトよりも大きい場合は、この数値は少なくとも 18 バイト必要です。それ以外の場合は、DFSORT は L4 を 18 バイトにセットします。

**L5**

平均レコード長。

**注:**

1. ブロック・セット手法が選択された場合は、L5 は使用されません。
2. L5 と AVGRLEN パラメーターの両方が指定された場合は、L5 は AVGRLEN パラメーターにより上書きされます。
3. L5 は可変長分類のみに使用されます。

デフォルト: なし。オプションです。

**L6, L7**

このレコード長は受け入れられますが、将来使用するために予約されています。

**注:**

1. 値を右から省略できます。たとえば、LENGTH=(80,70,70,70) とすることができます。
2. コンマまたはセミコロンによる値の省略を示せば、中間または左から値を省略できます。たとえば、LENGTH=(,,,30,80) とすることができます。

## RECORD 制御ステートメント

- L1 だけが指定されている場合は、括弧はオプションです。L2 ~ L7 のいずれかを指定する場合は、L1 の有無にかかわらず、括弧が必要です。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## レコード形式および長さの記述 — 例

### 例 1

```
MODS E15=(INEX,1000,EXIT),E35=(OUTEX,2000,EXIT)
RECORD LENGTH=(,175,180)
```

この例は、RECORD ステートメントを使用して、E15 および E35 出口がレコード長を変更することを指示する方法を表しています。レコード・タイプ (F) および入力レコード長 (200) は自動的に、入力データ・セットの RECFM および LRECL からそれぞれ取得されます。

#### LENGTH

L2 は、E15 出口が 175 バイト・レコードを渡すことを指定します。L3 は、E35 出口が 180 バイト・レコードを渡すことを指定します。

### 例 2

```
MODS E15=(E15ONLY,1000,EXIT)
RECORD TYPE=V,LENGTH=60
```

この例は、RECORD ステートメントを使用して、E15 出口が入力データのすべてを可変長レコードとして提供するときに、レコード・タイプおよび最大入力レコード長を設定する方法を表しています。

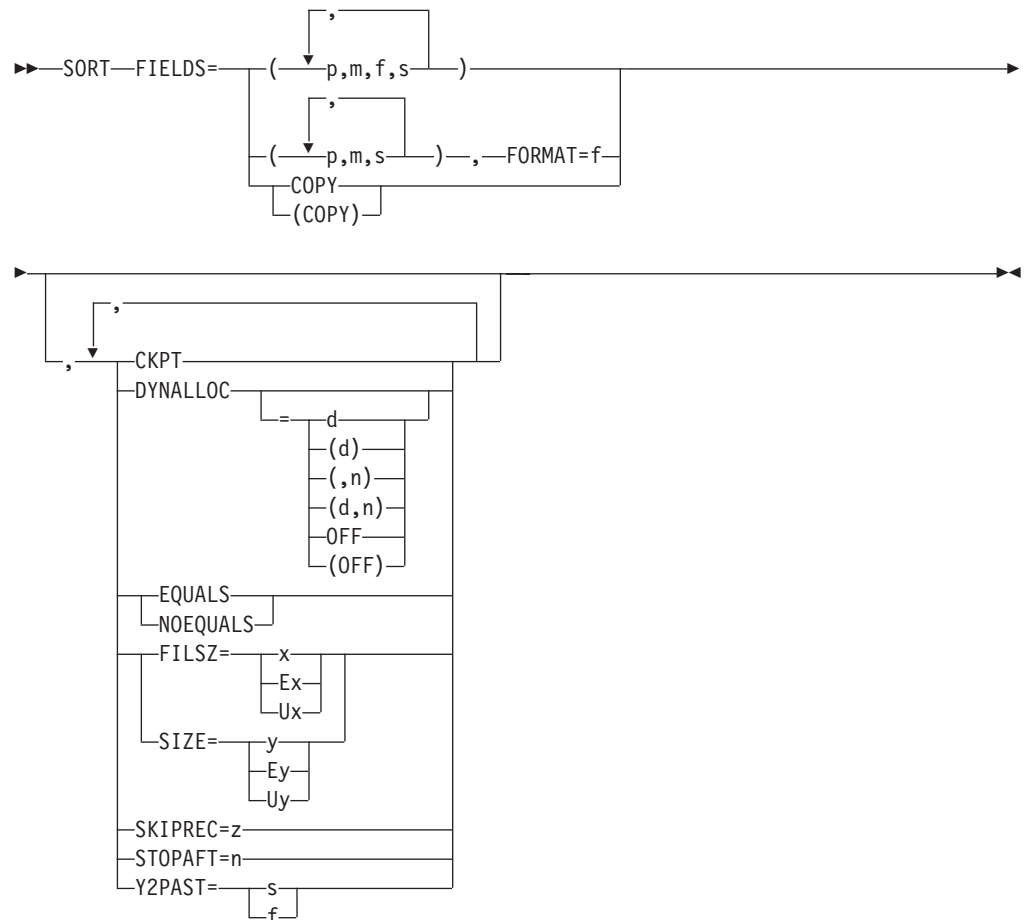
#### TYPE

V は、E15 出口が可変長レコードを挿入すること、つまり、挿入されたレコードの 1 ~ 4 桁目に RDW を含み、データが 5 桁目から開始されることを指定します。

#### LENGTH

L1 は、E15 出口が、最大長 60 バイトのレコードを挿入することを指定します。

## SORT 制御ステートメント



分類アプリケーションを実行するときは、SORT 制御ステートメントを使用します。このステートメントは、プログラムが分類する入力レコードの制御フィールドを記述します。また、SORT ステートメントを使用して、コピー・アプリケーションを指定することもできます。ユーザー・ラベルは出力データ・セットにコピーされません。

DFSORT が短い SORT 制御フィールドを処理する方法は、VLSHRT/NOVLSHRT の設定により異なります。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張されるようなフィールドを指します。短いレコードの分類の詳細については、181 ページの『OPTION 制御ステートメント』の VLSHRT および NOVLSHRT オプションの説明を参照してください。

SORT ステートメントで使用可能なオプションは、他のソースでも指定できます。これらのオプションで使用できるすべてのソースおよび指定変更の順序を示す表が、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』に記載されています。

## SORT 制御ステートメント

SORT ステートメントまたは OPTION ステートメントのいずれかで指定できるオプションは、OPTION ステートメントで指定するのが適当です。

DFSORT は、WORK=value および ORDER=value の SORT オペランドは受け入れませんが、処理は行いません。

DFSORT の照合の動作を、使用する文化環境に合わせて修正できます。文化環境は、活動ロケールを選択することにより設定されます。活動ロケールの照合規則は、SORT 処理に次のような影響を与えます。

- DFSORT は、活動ロケールに定義された照合規則に従って、出力用の分類済みレコードを作成します。これにより、言語の文化特性とローカル特性を保持する定義済み照合規則に基づいて、単一または複数バイトの文字データの分類が行われます。

ロケール処理を使用する場合は、活動ロケールは文字 (CH) 制御フィールドの処理のみに使用されます。

ロケールの処理の詳細については、7 ページの『文化的環境についての考慮事項』または 181 ページの『OPTION 制御ステートメント』の LOCALE の項を参照してください。

### FIELDS



入力レコードの各制御フィールドに関する 4 つの項目、すなわち、レコード内のフィールドの位置、フィールドの長さ、フィールド内のデータの形式、およびフィールドが分類される順序、が必要です。これらの項目は、p、m、f、および s で表され、FIELD オペランドの値により DFSORT に伝えられます。

すべての制御フィールドは、レコードの最初の 32752 バイト内にある必要があります。

VLSHRT が有効でない限り、制御フィールドは分類される最も短いレコードより長くはなりません。集められた制御フィールド (制御ワードを含む) は、4092 バイト (または EQUALS が有効な場合は 4088 バイト) を超えてはなりません。FIELDS オペランドは 2 通りの方法で書くことができます。

最初の FIELDS オペランド形式を使用して、異なるデータ形式を含む制御フィールドを記述します。2 番目の形式 (これについては本節の後半 FORMAT パラメーターの項で説明しています) を使用して、同じ形式のデータを含む SORT フィールドを記述します。2 番目の形式はオプションです。したがって、希望すれば、常に最初の形式を使用できます。

プログラムは最初は大制御フィールドを調べるため、大制御フィールドを最初に指定する必要があります。小制御フィールドは大制御フィールドの次に指定します。p、m、f、および s は制御フィールドを記述します。以下にその指定の詳細を説明します。



- p** 入力レコードの先頭を基準にしたときの、制御フィールドの最初のバイトを指定します。<sup>15</sup>

固定長レコードの最初のデータ・バイトは、相対位置は 1 になります。可変長レコードの最初のデータ・バイトは、相対位置が 5 になります。最初の 4 バイトにはレコード記述子ワードが含まれます。2 進数以外のすべての制御フィールドは、バイト境界から開始します。浮動小数点フィールドの最初のバイトは、符号付き指数と解釈されます。フィールドの残りの部分は小数部と解釈されます。

可変長レコードの始めには、実際のレコードの前にある 4 バイトの RDW が含まれます。これは VSAM 入力レコードの場合にも言えます。VSAM 入力レコードでは、入力時に、DFSORT が必要な RDW をプログラムに提供し、(VSAM データ・セットに出力する場合は) 出力時に再びそれを除去します。したがって、可変長レコードのバイト位置には必ず 4 バイトを加える必要があります。

2 進数値を含むフィールドは、次のような「バイト・ビット」表記法で記述されます。

1. まず、レコードの先頭を基準にしたバイト位置を指定し、その後にピリオドを付けます。
2. 次に、そのバイトの始めを基準にしたビット位置を指定します。バイトの最初の (高位) ビットがビット 0 になる (ビット 1 ではない) ことに注意してください。残りのビット番号は 1 ~ 7 になります。

したがって、1.0 はレコードの始まりを表します。レコードの 3 番目のバイトの 3 番目のビットから始まる 2 進数フィールドは、3.2 と表されます。2 進数フィールドの始めがバイト境界であれば (たとえば、4 番目のバイトであれば)、次のような 3 通りのうちのいずれかで書くことができます。

```
4.0
4.
4
```

この表記法の他の例が、348 ページの図 16 に示されています。

---

15. INREC が指定されている場合は、p は INREC により再フォーマット設定されたレコードを参照します。E15 ユーザー出口でレコードを再フォーマット設定し、しかも INREC が指定されていない場合は、p は E15 ユーザー出口により再フォーマット設定されたレコードを参照します。

## SORT 制御ステートメント

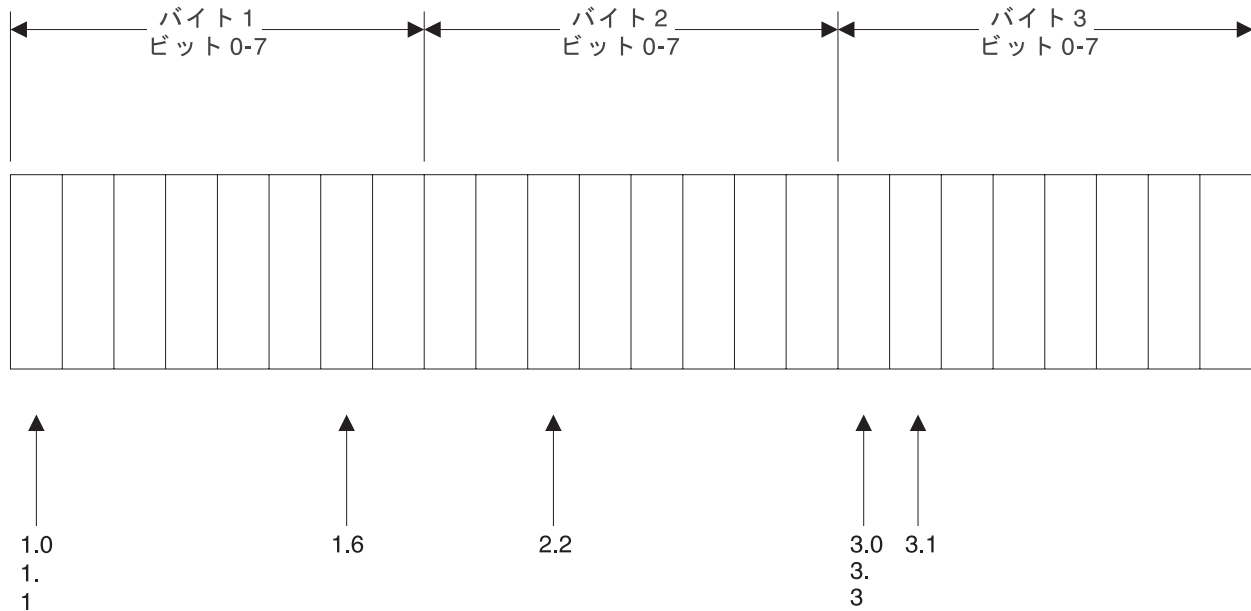


図 16. 2 進数フィールドの表記法の例

### m

制御フィールドの長さを指定します。2 進数フィールド以外のすべての制御フィールドの値は、整数のバイト数で表す必要があります。2 進数フィールドは、「バイト・ビット」表記で表すことができます。長さが整数値 (d) の 2 進数制御フィールドは、次の 3 通りのうちのいずれかで表すことができます。

d.0  
d.  
d

指定するビット数は、7 を超えてはなりません。長さが 2 ビットの制御フィールドは、0.2 と表します。

すべての制御フィールドが占めるバイトの合計数は、4092 (または、EQUALS オプションが使用されている場合は 4088 バイト) を超えてはなりません。合計を決定する場合、2 進数フィールドがバイトの一部しか占めていない場合でも、そのフィールドが 1 バイト全体を占めているものとして計算してください。たとえば、2.6 バイト目から始まる 3 ビットの長さの 2 進数フィールドは 2 バイトを占めます。すべてのフィールドは、レコードの最初の 32752 バイトにすべて含まれる必要があります。

**f** 制御フィールドのデータの形式を指定します。受け入れられる制御フィールドの長さ (バイト単位) および使用できる形式については、349 ページの表 46 を参照してください。

表 46. 制御フィールドの形式と長さ

形式	長さ	説明
CH	1 ~ 4092 バイト	文字 <sup>16</sup>
AQ	1 ~ 4092 バイト	代替照合順序をもつ文字
ZD	1 ~ 32 バイト	ゾーン 10 進数、符号あり
PD	1 ~ 32 バイト	パック 10 進数、符号あり
PD0	2 ~ 8 バイト	パック 10 進数、符号あり、1 桁目は無視
FI	1 ~ 256 バイト	固定小数点、符号あり
BI	1 ~ 4092 バイト	2 進数、符号なし
FL	1 ~ 256 バイト	浮動小数点、符号付き
AC	1 ~ 4092 バイト	ISCI/ASCII 文字
CSF または FS	1 ~ 16 バイト	任意指定の浮動符号が左端にある符号付き数字
CSL または LS	2 ~ 256 バイト	区切り符号が左端にある符号付き数字
CST または TS	2 ~ 256 バイト	符号桁が右端にある数字
CLO または OL	1 ~ 256 バイト	符号が左端にオーバーパンチされている数字
CTO または OT	1 ~ 256 バイト	符号が右端にオーバーパンチされている数字
ASL	2 ~ 256 バイト	符号桁が左端にある ISCI/ASCII 数字
AST	2 ~ 256 バイト	符号桁が右端にある ISCI/ASCII 数字
D1	1 ~ 4092 バイト	ユーザー定義のデータ・タイプ (EFS プログラムが必要)
Y2T	3 ~ 6 バイト	特殊標識付きの文字またはゾーン yyx...x 完全日付形式
Y2U	2 または 3 バイト	特殊標識付きのパック 10 進数の yyx および yyxxx の完全日付形式
Y2V	3 または 4 バイト	特殊標識付きのパック 10 進数の yyxx および yyxxxx の完全日付形式
Y2W	3 ~ 6 バイト	特殊標識付きの文字またはゾーン x...xyy 完全日付形式
Y2X	2 または 3 バイト	特殊標識付きのパック 10 進数の xyy および xxxyy の完全日付形式
Y2Y	3 または 4 バイト	特殊標識付きのパック 10 進数の xxyy および xxxxyy の完全日付形式

16. CHALT が有効な場合、CH は AQ として扱われます。

## SORT 制御ステートメント

表 46. 制御フィールドの形式と長さ (続き)

形式	長さ	説明
Y2C または Y2Z	2 バイト	2 桁文字またはゾーン 10 進数年データ
Y2P	2 バイト	2 桁パック 10 進数年データ
Y2D	1 バイト	2 桁 10 進数年データ
Y2S	2 バイト	特殊標識付きの 2 桁文字またはゾーン 10 進数年データ
Y2B	1 バイト	2 桁バイナリー年データ

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

CSF/FS、Y2、および PD0 形式フィールドは、ブロック・セットが選択されている場合のみ使用されます。

Y2 形式のフィールドの場合、実際の日付は、有効な Y2PAST オプションにより設定された「世紀」ウィンドウを使用して照合されますが、特殊標識には「世紀」ウィンドウは使用されません。そのため、Y2 形式では次のように、実際の日付と特殊標識を照合します。

• Y2T および Y2W:

**昇順:** BI ゼロ、ブランク、CH/ZD ゼロ、世紀の下位日付 (たとえば、19yy)、世紀の上位日付 (たとえば、20yy)、CH/ZD の 9、BI の 1。

**降順:** BI の 1、CH/ZD の 9、世紀の上位日付 (たとえば、20yy)、世紀の下位日付 (たとえば、19yy)、CH/ZD ゼロ、ブランク、BI ゼロ。

• Y2U、Y2V、Y2X、および Y2Y:

**昇順:** PD ゼロ、世紀の下位日付 (たとえば、19yy)、世紀の上位日付 (たとえば、20yy)、PD の 9。

**降順:** PD の 9、世紀の上位日付 (たとえば、20yy)、世紀の下位日付 (たとえば、19yy)、PD ゼロ。

• Y2C、Y2Z、Y2P、Y2D、および Y2B:

**昇順:** 世紀の下位日付 (たとえば、19yy)、世紀の上位日付 (たとえば、20yy)。

**降順:** 世紀の上位日付 (たとえば、20yy)、世紀の下位日付 (たとえば、19yy)。

• Y2S:

**昇順:** BI ゼロ、ブランク、世紀の下位日付 (たとえば、19yy)、世紀の上位日付 (たとえば、20yy)、BI の 1。

**降順:** BI の 1、世紀の上位日付 (たとえば、20yy)、世紀の下位日付 (たとえば、19yy)、ブランク、BI ゼロ。

AC 形式は、ISCH/ASCII 照合順序を使用して EBCDIC データの順序付けを行います。

## SORT 制御ステートメント

2 つ以上の制御フィールドを指定し、それらの制御フィールドのすべてに同じタイプのデータが含まれる場合は、f パラメーターを省略し、下記の任意指定の FORMAT オペランドを使用できます。

浮動小数点データをすべて正規化しておかないと、プログラムがそれを正しく照合できません。E15 または E61 ユーザー出口を使用すれば、処理時に正規化を行うことができます。E61 を使用する場合は、このユーザー出口で変更する各制御フィールドごとに、FIELDS オペランドの s の値に E オプションを指定します。

**s** 制御フィールドをどのように配列するかを指定します。有効なコードは次のとおりです。

- A** 昇順
- D** 降順
- E** 変更予定の制御フィールド

E61 ユーザー出口を組み込んで、プログラムが分類を行う前に制御フィールドを変更する場合は、E を指定します。E61 ユーザー出口が制御フィールドを変更すると、DFSORT は指定された形式を使用して、レコードを昇順に照合します。<sup>17</sup>

ユーザー出口を組み込む方法については、361 ページの『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』を参照してください。

デフォルト: なし。必須のパラメーターです。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## FORMAT

▶▶—FORMAT=f—▶▶

FORMAT=f は、FIELDS 式全体の制御フィールドが同じ形式の場合にのみ使用できます。使用できるフィールド形式については、フィールドの f の項を参照してください。

COPY オペランドを指定した場合は、FORMAT=f を指定できません。

デフォルト: なし。FIELDS パラメーターに組み込まれていない場合は、このパラメーターを指定する必要があります。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

17. 従来の組み合わせ、またはテープ作業データ・セット分類では、E が指定された制御フィールドは、それが実際にどのような形式で指定されていても、2 進数バイトの形式として処理されます。

## SORT 制御ステートメント

注: FORMAT と FIELDS の両方に形式値が指定されると、DFSORT は、通知メッセージを出し、FIELDS の形式値を使用し (f は各制御フィールドごとに指定します)、FORMAT で指定した形式値は使用しません。

### FIELDS=COPY または FIELDS=(COPY)



181 ページの『OPTION 制御ステートメント』の COPY オプションの説明を参照してください。

### CKPT



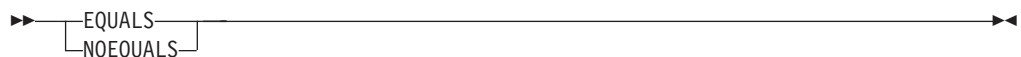
181 ページの『OPTION 制御ステートメント』のこのオプションの説明を参照してください。

### DYNALLOC



181 ページの『OPTION 制御ステートメント』のこのオプションの説明を参照してください。

### EQUALS または NOEQUALS



181 ページの『OPTION 制御ステートメント』のこれらのオプションの説明を参照してください。

### FILSZ または SIZE



181 ページの『OPTION 制御ステートメント』のこれらのオプションの説明を参照してください。

**SKIPREC**

▶▶—SKIPREC=z—▶▶

181 ページの『OPTION 制御ステートメント』のこのオプションの説明を参照してください。

**STOPAFT**

▶▶—STOPAFT=n—▶▶

181 ページの『OPTION 制御ステートメント』のこのオプションの説明を参照してください。

**Y2PAST**▶▶—Y2PAST= 

s
f

 —▶▶

181 ページの『OPTION 制御ステートメント』のこのオプションの説明を参照してください。

注: Y2PAST=value の代わりに CENTURY=value および CENTWIN=value を使用できます。

**SORT/MERGE ステートメントに関する注意事項**

- レコードが INREC (SORT および MERGE) または E15 (SORT) により再フォーマット設定される場合、FIELDS は、該当する再フォーマット設定されたレコードのフィールドを参照します。
- SZERO が有効な場合、数値フィールドが分類または組み合わせられる時に、-0 は、昇順で +0 の前に照合し、降順で +0 の後に照合します。NOSZERO が有効な場合は、数値フィールドが分類または組み合わせられるとき、-0 は、+0 と等しいとして比較されます。ただし、従来の組み合わせ、またはテープ作業データ・セット分類のアプリケーションの場合、SZERO が常に使用されます。

**SORT または COPY の指定 — 例****例 1**

```
SORT FIELDS=(2,5,FS,A),FILSZ=29483
```

**FIELDS**

制御フィールドは入力データ・セットの各レコードの 2 バイト目から始まり、5 バイトの長さで浮動符号データを含みます。昇順に分類されます。

**FILSZ**

分類するデータ・セットには、正確に 29483 レコードが含まれます。

## SORT 制御ステートメント

### 例 2

```
SORT  FIELDS=(7,3,CH,D,1,5,FI,A,398.4,7.6,BI,D,99.0,230.2,  
          BI,A,452,8,FL,A),DYNALLOC=(3390,4)
```

#### FIELDS

最初の 4 つの値は、大制御フィールドを記述します。このフィールドは各レコードの 7 バイト目から始まり、3 バイトの長さで、文字 (EBCDIC) データを含みます。降順に分類されます。

次の 4 つの値は、2 番目の制御フィールドを記述します。このフィールドは 1 バイト目から始まり、5 バイトの長さで、固定小数点データを含み、昇順で分類されます。

3 番目の制御フィールドは 398 バイト目の 5 番目のビットから始まります (ビットには 0 から 7 の番号が付けられています)。このフィールドは 7 バイトと 6 ビットの長さ (9 バイトを占める) で、2 進データが降順で入れられます。

4 番目の制御フィールドは 99 バイト目から始まり、230 バイトと 2 ビットの長さで、2 進データを含みます。昇順に分類されます。

5 番目の制御フィールドは 452 バイト目から始まり、8 バイトの長さで、正規化された浮動小数点データを含みます。昇順に分類されます。このフィールドのデータが正規化されていない場合は、A の代わりに E を指定し、ユーザー独自の E61 ユーザー出力ルーチンを組み込んで、プログラムが調べる前にこのフィールドを正規化できます。

#### DYNALLOC

4 つの作業データ・セットが 3390 に割り振られます。各データ・セット上のスペースは、FILSZ 値を使用して計算されます。

### 例 3

```
SORT  FIELDS=(3,8,ZD,E,40,6,CH,D)
```

#### FIELDS

最初の 4 つの値は、大制御フィールドを記述します。このフィールドは各レコードの 3 バイト目から始まり、8 バイトの長さで、ゾーン 10 進数を含み、分類プログラムがこのフィールドを調べる前にユーザー・ルーチンにより変更されます。

2 番目のフィールドは 40 バイト目から始まり、6 バイトの長さで、文字 (EBCDIC) データを含み、降順に分類されます。

### 例 4

```
SORT  FIELDS=(7025,4,A,5048,8,A),FORMAT=ZD,EQUALS
```

#### FIELDS

大制御フィールドは各レコードの 7025 バイト目から始まり、4 バイトの長さで、ゾーン 10 進数データ (FORMAT=ZD) を含み、昇順に分類されます。



## SORT 制御ステートメント

2 番目の制御フィールドは 5048 バイト目から始まり、8 バイトの長さで、最初のフィールドのデータ形式と同じです。このフィールドも昇順に分類されます。

### FORMAT

両方の制御フィールドのデータ形式が同じなので、**FORMAT=f** オプションを使用できます。また、この **SORT** ステートメントを次のように書くこともできます。

```
SORT  FIELDS=(7025,4,ZD,A,5048,8,ZD,A),EQUALS
```

### EQUALS

照合結果が等しいレコードの順序を、入力から出力まで維持することを指定します。

### 例 5

```
SORT  FIELDS=COPY
```

### FIELDS

入力データ・セットを、分類処理や組み合わせ処理を行わずに、出力データ・セットにコピーします。

### 例 6

```
OPTION Y2PAST=1950  
SORT  FIELDS=(21,6,Y2T,A,13,3,Y2X,D)
```

### Y2PAST

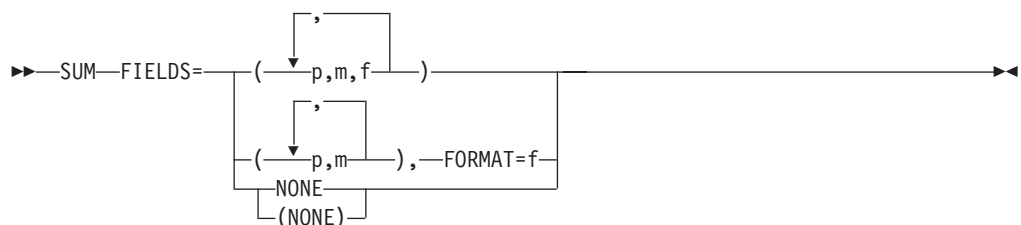
1950 ~ 2049 の「世紀」ウィンドウをセットします。

### FIELDS

21 ~ 26 桁目の C'yymmdd' (または Z'yymmdd') 日付を昇順で分類し、13 ~ 15 桁目の P'dddy' 日付を降順で分類します。「実際の」日付は、1950 ~ 2049 の「世紀」ウィンドウを使用して分類されます。特殊標識は、「実際の」日付を基準に正しく分類されます。

---

## SUM 制御ステートメント



SUM 制御ステートメントは、同じ分類または組み合わせ制御フィールドをもつ 2 つのレコードが見つかる都度、その合計フィールドの内容を加算し、その合計をレコードのうちの 1 つに入れ、他方のレコードを削除することを指定します。

## SUM 制御ステートメント

EQUALS オプションが有効な場合、合計された最初のレコードが保持されます。  
 NOEQUALS オプションが有効な場合、保持されるレコードは予測できません。詳細については、357 ページの『SUM ステートメントに関する注意事項』を参照してください。

ZDPRINT オプションが有効な場合、ZD の正の合計値を印刷できます。NZDPRINT オプションが有効な場合、ZD の正の合計値は印刷できません。詳細については、357 ページの『SUM ステートメントに関する注意事項』を参照してください。

DFSORT が短い SUM 要約フィールドを処理する方法は、VLSHRT または NOVLSHRT オプションが有効かどうかにより異なります。短いフィールドとは、可変長レコードが短すぎてフィールド全体を含むことができない、つまり、フィールドがレコードを超えて拡張されるようなフィールドを指します。短いレコードの分類、組み合わせおよび要約の詳細については、181 ページの『OPTION 制御ステートメント』の VLSHRT および NOVLSHRT オプションの説明を参照してください。

### FIELDS



入力レコードの数値フィールドを合計フィールドとして指定します。

**p** 入力レコードの先頭を基準にしての、フィールドの最初のバイトを指定します。<sup>18</sup> 固定長レコードの最初のデータ・バイトは相対位置が 1 になります。最初の 4 バイトを RDW が占めているため、可変長レコードの最初のデータ・バイトの相対位置は 5 になります。すべてのフィールドは、バイト境界から開始します。また、レコードの最初の 4092 バイト内に含まれる必要があります。ただし、168 ページの『例 4』に示すように、INREC および OUTREC を使用してレコードを再配列し、最初の 4092 バイトを超えるフィールドを合計できるようにすることが可能です。

**m** 加算される合計フィールドの長さをバイト単位で指定します。長さに指定できる値については、下記を参照してください。

**f** 合計フィールドのデータの形式を指定します。

表 47. 合計フィールドの形式と長さ

形式コード	長さ	説明
BI	2、4、または 8 バイト	2 進数、符号なし
FI	2、4、または 8 バイト	固定小数点、符号あり
FL	4、8、または 16 バイト	浮動小数点、符号付き
PD	1 ~ 16 バイト	パック 10 進数、符号あり
ZD	1 ~ 18 バイト	ゾーン 10 進数、符号あり

18. INREC が指定されている場合は、p は INREC により再フォーマット設定されたレコードを参照します。E15 ユーザー出口でレコードを再フォーマット設定し、しかも INREC が指定されていない場合は、p は E15 ユーザー出口により再フォーマット設定されたレコードを参照します。

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

#### NONE または (NONE)

複写キーをもつレコードを除去します。各キーにつき 1 つのレコードだけが保持され、合計は行われません。

注: ICETOOL の SELECT 演算子の FIRST オペランドを使用して、OPTION EQUALS 付きの SUM FIELDS=NONE と同じ機能を実行できます。さらに、SELECT の ALLDUPS、NODUPS、HIGHER(x)、LOWER(y)、EQUAL(v)、および LAST オペランドを使用して、複写キーおよび非複写キーに関する別の基準に基づいてレコードを選択できます。SELECT の DISCARD(savedd) オペランドを使用すると、FIRST、ALLDUPS、NODUPS、HIGHER(x)、LOWER(y)、EQUAL(v)、または LAST により廃棄されたレコードを保管できます。SELECT 演算子の詳細については、527 ページの『SELECT 演算子』を参照してください。

デフォルト: なし。必須のパラメーターです。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

#### FORMAT

▶▶—FORMAT=f—▶▶

FORMAT=f は、FIELDS 式全体の合計フィールドが同じ形式の場合のみ使用できます。使用できるフィールド形式については、FIELDS の 'f' の項を参照してください。

デフォルト: なし。FIELDS パラメーターに組み込まれていない場合は、このパラメーターを指定する必要があります。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

適用できる機能: 697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

注: FORMAT と FIELDS の両方に形式値が指定されると、DFSORT は、通知メッセージを出し、FIELDS の形式値を使用します (f は各合計フィールドごとに指定します)。FORMAT で指定した形式値は使用しません。

## SUM ステートメントに関する注意事項

- 無効な PD または ZD 符号、あるいは無効な数字は、データ例外 (0C7 ABEND) を引き起こします。0 ~ 9 は無効な符号、A ~ F は無効な数字です。たとえば、3.5 (X'F34BF5') という ZD 値は、"." (X'4B') が無効な数字として扱われるため、0C7 になります。ICETOOL の DISPLAY または VERIFY 演算子を使用して、無効な数字をもつ 10 進数値を識別できます。ICETOOL の VERIFY 演算子を使用して、無効な符号をもつ 10 進数値を識別できます。

## SUM 制御ステートメント

- 正の合計 ZD の結果が印刷可能な数字かどうかは、NZDPRINT または ZDPRINT のどちらが有効になっているかにより決まります。(これらは ICEMAC の ZDPRINT オプション、および OPTION ステートメントの NZDPRINT および ZDPRINT パラメーターにより設定されます。)
  - NZDPRINT が有効な場合は、元の値が印刷可能な数字から構成されているかどうかにかかわらず、正の合計 ZD の結果は、印刷可能な数字にはなりません。たとえば、X'F2F3F1' ('231' と印刷) および X'F3F0F6' ('306' と印刷) が合計されると、NZDPRINT が有効な場合の結果は X'F5F3C7' ('53G' と印刷) になります。
  - ZDPRINT が有効な場合は、元の値が印刷可能な数字から構成されているかどうかにかかわらず、正の合計 ZD の結果は、印刷可能な数字になります。たとえば、X'F2F3C1' ('23A' と印刷) および X'F3F0F6' ('306' と印刷) が合計されると、ZDPRINT が有効な場合の結果は X'F5F3F7' ('537' と印刷) になります。

したがって、正の合計 ZD の結果が必ず印刷可能であるようにするためには、ZDPRINT が有効である必要があります。

合計されない正の ZD 値は、NZDPRINT または ZDPRINT のどちらが有効であるにかかわらず、元の符号を保持します。たとえば、X'F2F8C5' が合計されなければ、X'F2F8C5' ('28E' と印刷) のままです。OUTFIL の OUTREC パラメーターを使用して、下記の例 4 で図示しているように、合計 ZD 値または非合計 ZD 値のすべてを常に印刷可能にすることができます。

- 入力レコードが INREC または E15 により再フォーマット設定されている場合は、SUM は再フォーマット設定された該当レコードのフィールドを参照します (前述の *p* の項を参照)。
- 合計フィールドは制御フィールドとして使用できません。合計フィールドと制御フィールド、または合計フィールド同士が重なり合うことはできず、また合計フィールドと RDW も重なり合うことはできません。
- 合計する FL 値は、正規化も非正規化もできます。ただし、結果として得られる FL 値は常に正規化されます。ハードウェアによる正規化処理では、FL 値が異なる順序で合計され、合計値が異なる値になる場合があります。
- FL 値の合計が指数オーバーフローを起こすと、指数オーバーフロー例外 (0CC ABEND) になります。
- FL 値の合計が指数下位オーバーフローを起こすと、結果は真のゼロになります。
- レコードを合計する場合、どのレコードが合計を受け取り (そして保持され)、どのレコードが削除されるかを予測できるのは、EQUALS が有効で、オーバーフローが発生せず、ブロック・セット手法が使用される場合のみです。この場合は、最初のレコード (181 ページの『OPTION 制御ステートメント』の EQUALS または NOEQUALS パラメーターの項で記述されている順序に基づく) が選択されて合計が入れられます。
- 合計フィールド以外のフィールドは変更されず、合計を受け取るレコードからフィールドがとられます。
- BI、FI、PD または ZD 値がオーバーフローしたときの DFSORT の処置は、181 ページの『OPTION 制御ステートメント』で説明するように、OVFLO パラメーターにより制御できます。

## SUM 制御ステートメント

- テープ作業データ・セット分類または従来の組み合わせの場合に SUM ステートメントを指定すると、DFSORT はメッセージを出して処理を終了します。
- DFSORT は、SUM 処理により削除されたレコードを SORTXSUM DD データ・セットに書き込むために、競合分類製品が提供する XSUM パラメーターをサポートしません。ただし、ICETOOL の SELECT 演算子は、FIELDS=NONE を含む XSUM と同じ機能を実行できます。たとえば、以下の ICETOOL ジョブを実行すると仮定します。

```
//S1EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//SORTIN DD DSN=...
//SORTOUT DD DSN=...
//SORTXSUM DD DSN=...
//TOOLIN DD *
SELECT FROM(SORTIN) TO(SORTOUT)-
  ON(5,4,CH) FIRST DISCARD(SORTXSUM)
/*
```

上記は、以下の XSUM ジョブと同等です。

```
//S1 EXEC PGM=ICEMAN
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=...
//SORTOUT DD DSN=...
//SORTXSUM DD DSN=...
//SYSIN DD *
  SORT FIELDS=(5,4,CH,A)
  SUM FIELDS=NONE,XSUM
/*
```

ICETOOL の SELECT 演算子で、XSUM では使用できない新たな機能を実行できます。詳細については、441 ページの『第 6 章 ICETOOL の使用』を参照してください。

## 合計フィールドの加算 - 例

### 例 1

```
SUM FIELDS=(21,8,PD,11,4,FI)
```

このステートメントは、21 バイト目の 8 バイトのパック 10 進フィールドと、11 バイト目の 4 バイトの固定小数点整数フィールドを合計フィールドとして指定します。

### 例 2

```
SUM FIELDS=NONE
```

このステートメントは、重複レコードの除去を示しています。

### 例 3

```
SUM FIELDS=(41,8,49,4),FORMAT=ZD
OPTION ZDPRINT
```

## SUM 制御ステートメント

これらのステートメントは、FORMAT オペランドおよび ZDPRINT オプションの使用を示しています。SUM ステートメントは、2 つのゾーン 10 進フィールドを指定します。つまり、1 つは 41 バイト目から始まる 8 バイトの長さフィールドで、もう 1 つは 49 バイト目から始まる 4 バイトの長さのフィールドです。ZDPRINT オプションの結果、正の合計 ZD 値は印刷可能になります。ただし、オーバーフローや固有のキーのために合計されない ZD 値に対しては、ZDPRINT オプションは効力を持たないことに注意してください。次の例では、OUTFIL を使用してすべての合計フィールドを印刷可能にする方法について示しています。

### 例 4

```
SUM FIELDS=(41,8,49,4),FORMAT=ZD
OUTFIL OUTREC=(1,40,41,8,ZD,M11,49,4,ZD,M11,53,28)
```

これらのステートメントは、OUTFIL ステートメントを使用して、出力データ・セット内のすべての正の ZD 値を印刷可能にする方法を示しています。ZDPRINT オプションが効力をもつのは、正の合計 ZD フィールドだけであるため、OUTFIL を使用して、値が合計されているかどうかにかかわらず、正または負の BI、FI、PD、あるいは ZD 値を編集できます。また、OUTFIL を使用して複数の出力データ・セット、報告書などを作成することもできます。OUTFIL 処理の詳細については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

注: 図で示すために、この例では入力レコードを 80 バイトと想定しています。

## 第 4 章 ユーザー独自のユーザー出口ルーチンの使用

ユーザー出口ルーチンの概要	362
DFSORT プログラム・フェーズ	363
ユーザー出口ルーチンの機能	366
DFSORT の入力 / ユーザー出口 / 出力論理の例	366
データ・セットのオープンおよび初期設定	368
制御フィールドの変更	368
レコードの挿入、削除、および変更	368
レコードの合計	368
特殊入出力の処理	368
読み取りエラー・ルーチン	369
書き込みエラー・ルーチン	369
VSAM ユーザー出口機能	369
中間記憶域不足時の処置の決定	369
データ・セットのクローズ	370
DFSORT の終了	370
ユーザー出口のアドレッシング・モードおよび常駐モード	370
ユーザー出口ルーチンが DFSORT のパフォーマンスに与える影響	371
ユーザー出口ルーチンに関する規則	371
ユーザー出口ルーチンのロード方法	372
ユーザー出口関係規則	373
関係の例	373
ユーザー出口ルーチンのダイナミック・リンク編集の方法	374
アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)	375
E11 ユーザー出口: データ・セットのオープン / ルーチンの初期設定	375
E15 ユーザー出口: レコードの引き渡しまたは変更 (分類アプリケーション およびコピー・アプリケーションの場合)	375
E15 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報	376
E15 の戻りコード	377
E15 ユーザー出口のための記憶域の使用法	378
E16 ユーザー出口: 中間記憶装置の計算違いの処理	378
E16 の戻りコード	379
E17 ユーザー出口: データ・セットのクローズ	379
E18 ユーザー出口: 入力データ・セットの処理	379
QSAM/BSAM による E18 ユーザー出口の使用	380
VSAM による E18 ユーザー出口の使用	381
E19 ユーザー出口: 作業データ・セットへの出力の処理	383
QSAM/BSAM による E19 ユーザー出口の使用	383
E61 ユーザー出口: 制御フィールドの変更	384
E61 ユーザー出口のいくつかの使用法	385
E61 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報	385
アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)	386
E31 ユーザー出口: データ・セットのオープン / ルーチンの初期設定	386
E32 ユーザー出口: 組み合わせのみの入力の処理	386
E32 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報	387
E32 の戻りコード	387
E35 ユーザー出口: レコードの変更	388
E35 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報	389
E35 の戻りコード	390

## ユーザー独自のユーザー出口ルーチンの使用

E35 ユーザー出口のための記憶域の使用法 . . . . .	391
E37 E37 ユーザー出口: データ・セットのクローズ . . . . .	392
E38 ユーザー出口: 入力データ・セットの処理 . . . . .	392
VSAM による E38 ユーザー出口の使用 . . . . .	392
E39 ユーザー出口: 出力データ・セットの処理 . . . . .	392
QSAM/BSAM による E39 ユーザー出口の使用 . . . . .	393
VSAM による E39 ユーザー出口の使用 . . . . .	393
アセンブラで作成されたサンプル・ルーチン . . . . .	393
E15 ユーザー出口: レコード長の変更 . . . . .	393
E16 ユーザー出口: NMAX 超過時の現行レコードの分類 . . . . .	394
E35 ユーザー出口: レコード長の変更 . . . . .	395
E61 ユーザー出口: 制御フィールドの変更 . . . . .	396
COBOL ユーザー出口ルーチン . . . . .	396
COBOL ユーザー出口に関する要件 . . . . .	397
COBOL に関する要件 (コピー処理の場合) . . . . .	398
COBOL 記憶域の所要量 . . . . .	399
COBOL ユーザー出口ルーチン (入力フェーズのユーザー出口) . . . . .	400
COBOL E15 ユーザー出口: レコードの引き渡しまたは変更 . . . . .	400
COBOL との E15 インターフェース . . . . .	401
E15 LINKAGE SECTION フィールド (固定長および可変長レコードの場合) . . . . .	403
E15 の戻りコード . . . . .	405
E15 PROCEDURE DIVISION に関する要件 . . . . .	406
COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口) . . . . .	407
COBOL E35 ユーザー出口: レコードの変更 . . . . .	407
COBOL との E35 インターフェース . . . . .	407
E35 LINKAGE SECTION フィールド (固定長および可変長レコードの場合) . . . . .	410
E35 の戻りコード . . . . .	412
E35 PROCEDURE DIVISION に関する要件 . . . . .	413
COBOL で作成されたサンプル・ルーチン . . . . .	414
COBOL E15 ユーザー出口: レコードの変更 . . . . .	414
COBOL E35 ユーザー出口: レコードの挿入 . . . . .	415
E15/E35 戻りコードおよび EXITCK . . . . .	417

---

## ユーザー出口ルーチンの概要

DFSORT は、実行可能コードの中の、ユーザー出口 と呼ばれるいくつかの箇所  
で、プログラム制御をユーザー独自のルーチンに渡すことができます。ユーザー出  
口ルーチンでは、レコードの削除、挿入、変更、合計など、各種の機能を行うこと  
ができます。

これらの作業を実行する必要がある場合は、DFSORT がすでに、各種の DFSORT  
プログラム制御ステートメントに指定されたユーザー・データを処理するために、  
広範な機能を提供していることを知っておく必要があります。INCLUDE、  
OMIT、INREC、OUTFIL、OUTREC、および SUM プログラム制御ステートメント  
については、111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』  
を参照してください。ユーザーの要求を満たすには、プログラム制御ステート  
メントを使用してレコードを処理するほうが、より適切であると判断されることが  
あります。



本章では、アセンブラーまたは COBOL で作成されたルーチンについてのみ記述していますが、この他にも次の機能をもつ任意の言語で出口ルーチンを作成できます。

- 下記のアドレスを汎用レジスター 1 で渡したり受け取ったりする機能
  - レコード
  - フルワードのゼロ
  - パラメーター・リスト
- レジスター 15 に戻りコードを渡す機能

MODS プログラム制御ステートメントを使用すれば、実行時にユーザー出口ルーチンを容易に活動化できます (173 ページの『MODS 制御ステートメント』を参照)。また、ある特定の条件下では、呼び出しパラメーター・リストにユーザー出口ルーチンのアドレスを渡すことにより、ユーザー出口ルーチンを活動化することもできます。詳細については、421 ページの『第 5 章 プログラムからの DFSORT の呼び出し』を参照してください。

ユーザー出口ルーチンの処理方法に影響を与えるパラメーターは次のとおりです。

- MODS ステートメント (173 ページの『MODS 制御ステートメント』を参照)
- ICEMAC インストール・マクロの COBEXIT オプション (18 ページの『インストール・デフォルト』を参照)
- EXEC ステートメントの E15=COB および E35=COB PARM オプション (34 ページの『EXEC/DFSPARM PARM オプションの指定』を参照)
- OPTION ステートメントの COBEXIT オプション (181 ページの『OPTION 制御ステートメント』を参照)
- ICEMAC インストール・オプションの EXITCK (18 ページの『インストール・デフォルト』を参照)
- OPTION ステートメントの EXITCK オプション (181 ページの『OPTION 制御ステートメント』を参照)

**注:** 本章では、あいまいにならないようにするために、ユーザー・システムで IBM デフォルトの EXITCK=STRONG が選択されているものと仮定します。

ある種のユーザー出口ルーチンは、特殊インターフェースを使用して、COBOL で作成できます。ユーザー出口ルーチンを PL/I で作成する場合は、PL/I サブルーチン機能を使用します。

ユーザー出口で使用するためのスペースを予約しておく必要がある場合もあります。記憶域の詳細については、631 ページの『主記憶域の効率的な使用』を参照してください。

---

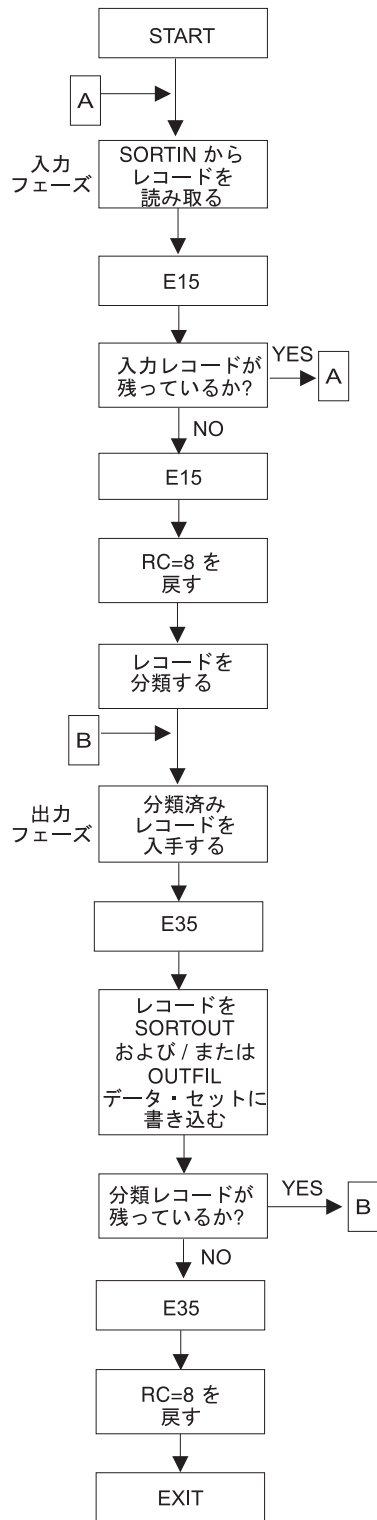
## DFSORT プログラム・フェーズ

DFSORT プログラム・フェーズとは、出力ファイルの作成などのような特定のタスクを実行するために設計された、大きな DFSORT 構成要素です。各種のユーザー入出力が入出力フェーズに含まれており、DFSORT 処理中の特定の時点で活動化されます。入力フェーズは、分類またはコピーにのみ使用されます。出力フェーズが終

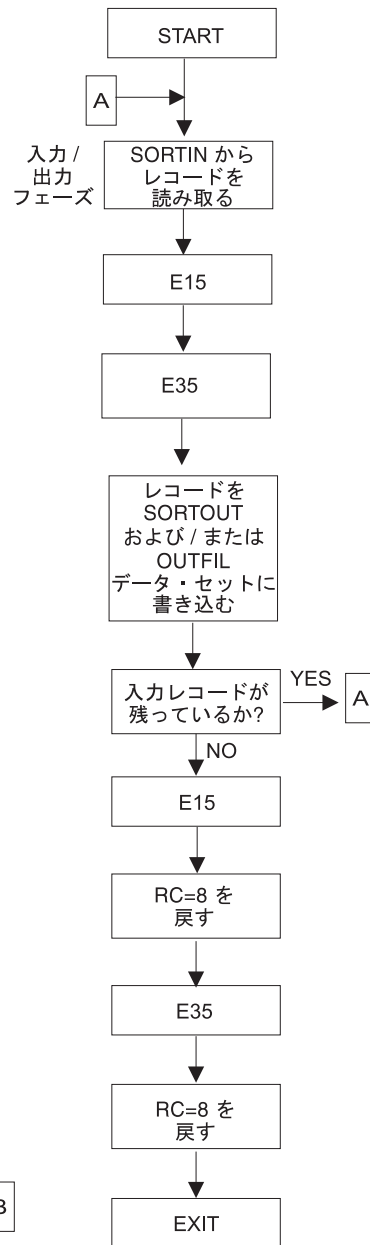
## DFSORT プログラム・フェーズ

わると、DFSORT はオペレーティング・システムまたは呼び出しプログラムに制御を戻します。365 ページの図 17 に、DFSORT の入出力論理を示します。

入力 / 出力 / 出力論理  
(分類アプリケーションの場合)



入力 / 出力 / 出力論理  
(コピー・アプリケーションの場合)



入力 / 出力 / 出力論理  
(組み合わせアプリケーションの場合)

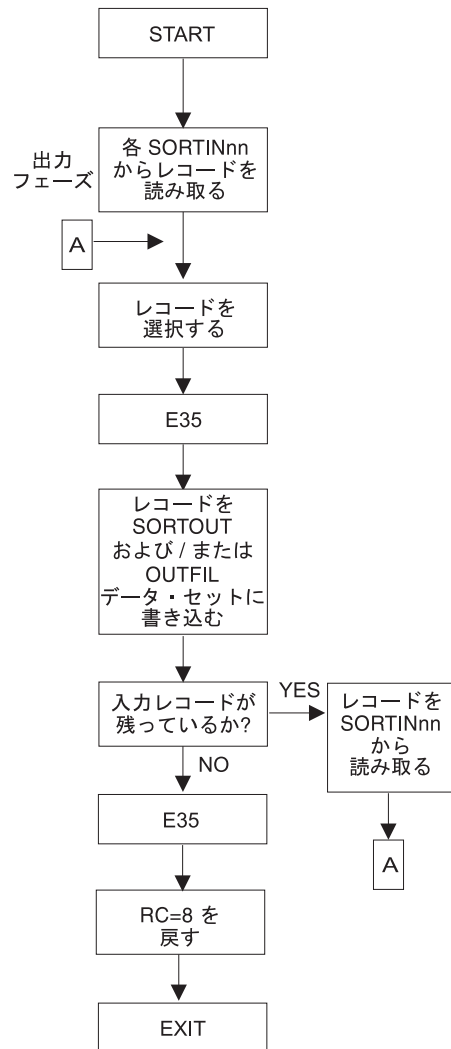


図 17. DFSORT の入力 / ユーザー出口 / 出力論理の例

## ユーザー出口ルーチンの機能

出口ルーチンを使用して、次のような各種のタスクを行うことができます。

- データ・セットのオープンおよび初期設定
- 制御フィールドの変更
- レコードの挿入、削除、または変更
- レコードの合計
- 特殊入出力条件の処理
- 中間記憶域不足時の処置の決定
- データ・セットのクローズ
- DFSORT の終了

365 ページの図 17、367 ページの表 48、および 367 ページの表 49 は、ユーザー出口ルーチンの機能およびそれらのルーチンが関連する出口とフェーズについて要約したものです。

## DFSORT の入力 / ユーザー出口 / 出力論理の例

365 ページの図 17 は、分類、コピー、および組み合わせアプリケーションが SORTINnn、E15 または E35 ユーザー出口、および SORTOUT に関連している場合の論理の流れの例を示しています。この図は、E15 および E35 ユーザー出口が、アプリケーションの論理にどのように適合されるかを示すためのものです。可能なすべての経路を網羅しているわけではありません。分かりやすくするために、適用可能なデータ・セットと出口がすべて存在し、レコードの挿入や削除が行われないことを前提としています。(組み合わせの場合、SORTINnn データ・セットからではなく、E32 ユーザー出口によりレコードが与えられる場合は、同様の論理が使用されます。)

365 ページの図 17 で示されている論理は、次のとおりです。

- E15 および E35 ユーザー出口が戻りコード 8 を返すまで、繰り返し、E15 および E35 ユーザー出口 に入ります。処理する入力レコードがまだ残っているときに、ユーザー出口が戻りコード 8 を DFSORT に渡すと、それらのレコードはユーザー出口に 渡されずに、DFSORT により処理されます。
- 分類処理中、各レコードは SORTIN から読み込まれ、E15 ユーザー出口に渡されます。このようにして すべての レコードが処理されると、レコードは DFSORT により分類され、分類された各レコードは E35 に渡されて、出力データ・セットに書き込まれます。
- コピー処理中、各レコードは SORTIN から読み込まれ、E15 および E35 ユーザー出口に渡され、出力データ・セットに書き出されます。
- 組み合わせ処理中、最初に各 SORTINnn データ・セットからレコードが 1 つずつ読み込まれます。出力するレコードが選択され、E35 に渡され、出力データ・セットに書き込まれます。次に、選択されたレコードは、同じ SORTINnn データ・セットからレコードを読み取るにより置き換えられ、処理が続けられます。

**注:** 組み合わせアプリケーションの場合、E35 ユーザー出口ルーチンの中で削除されたレコードのシーケンス検査は行われません。出力データ・セットのない E35 ユーザー出口ルーチンを使用する場合は、レコードが E35 ユーザー

## ユーザー出口ルーチンの機能

出口に渡される時点ではシーケンス検査は行われません。したがって、入力レコードが正しい順序になっていることを確認する必要があります。

表 48. プログラム・ユーザー出口のルーチンの機能 (分類の場合)

機能	分類入力フェーズ	分類出力フェーズ
オープン / 初期設定	E11、E15 ユーザー出口	E31 ユーザー出口
制御フィールドの変更	E61 ユーザー出口	該当せず
挿入、削除 / 変更	E15 ユーザー出口	E35 ユーザー出口
レコードの合計		E35 ユーザー出口 <sup>1</sup>
特殊入出力条件の処理: QSAM/BSAM および VSAM SORTIN QSAM/BSAM SORTOUT VSAM SORTOUT	E18 ユーザー出口 E19 ユーザー出口 <sup>2</sup> 該当せず	E38 ユーザー出口 <sup>2</sup> E39 ユーザー出口 <sup>3</sup> E39 ユーザー出口 <sup>3</sup>
中間記憶域不足時の処置の決定	E16 ユーザー出口 <sup>4</sup>	該当せず
クローズ / 操作援助	E15、E17 ユーザー出口	E35、E37 ユーザー出口
DFSORT を終了する	E15 ユーザー出口	E35 ユーザー出口

**注:**

1. ユーザー独自のルーチンの代わりに SUM 制御ステートメントを使用して、レコードの合計をとることができます。
2. テープ作業データ・セット分類だけに適用されます。
3. E39 は SORTOUT には使用できますが、OUTFIL データ・セットには使用できません。
4. テープ作業データ・セット分類、または作業データ・セットを使用しないピアレッジ / ベール分類だけに適用されます。

表 49. プログラム・ユーザー出口のルーチンの機能 (コピーおよび組み合わせの場合)

機能	コピー	組み合わせ
オープン / 初期設定	E15、E31 ユーザー出口	E31 ユーザー出口
制御フィールドの変更	該当せず	E61 ユーザー出口
挿入	E15、E35 ユーザー出口	E32、E35 ユーザー出口
削除 / 変更	E15、E35 ユーザー出口	E35 ユーザー出口
レコードの合計	E35 ユーザー出口	E35 ユーザー出口 <sup>1</sup>
特殊入出力条件の処理: QSAM/BSAM および VSAM SORTIN(nm) QSAM/BSAM および VSAM SORTOUT	E38 ユーザー出口 E39 ユーザー出口	E38 ユーザー出口 E39 ユーザー出口
クローズ / 操作援助	E35、E37 ユーザー出口	E35、E37 ユーザー出口
DFSORT を終了する	E15、E35 ユーザー出口	E32、E35 ユーザー出口

**注:**

1. ユーザー独自のルーチンの代わりに SUM 制御ステートメントを使用して、レコードの合計をとることができます。

## データ・セットのオープンおよび初期設定

ユーザーは、独自のルーチンを作成して、データ・セットをオープンし、各種の初期設定を行なうことができます。その場合、それらのルーチンを E11、E15、E31、および E35 ユーザー出口と関連付けます。

入力ファイルのラベルの検査には、E18 および E38 ユーザー出口を使用します。

## 制御フィールドの変更

DFSORT が制御フィールドを比較する前にそれらを変更するルーチンを作成できます。これにより、たとえば浮動小数点の制御フィールドを正規化できます。また、レコードが最終的に分類または組み合わせされる順序を変更することもできます。ただし、この機能については、通常は DFSORT の ALTSEQ プログラム制御ステートメントを使用します。このルーチンを E61 ユーザー出口と関連付ける必要があります。

E61 ユーザー出口を使用するときは、その後の比較では常に、変更された制御フィールドが昇順に配列されます。

注: 比較の前に制御フィールドを変更しても、元のレコードは変更されません。

## レコードの挿入、削除、および変更

ユーザーは独自のユーザー・ルーチンを作成して、レコードを挿入、削除、および変更できます。これらのルーチンを E15、E32、および E35 ユーザー出口に関連付ける必要があります。

注: DFSORT には、ユーザーのフィールド基準に基づいて、レコードの組み込みや削除を自動的に行う INCLUDE ステートメントと OMIT ステートメント、および OUTFIL INCLUDE パラメーターと OMIT パラメーターも備わっています。これらの制御ステートメントの詳細については、111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』を参照してください。

## レコードの合計

E35 ユーザー出口を使用して、出力用のレコードを合計できます。ただし、DFSORT の SUM プログラム制御ステートメントを使用すると、ユーザー出口を用いなくても同じことが行えます。355 ページの『SUM 制御ステートメント』を参照してください。

## 特殊入出力の処理

DFSORT には、特殊入出力条件を処理する 4 つの出口があります。つまり、SORTIN および SORTINnn 用には E18 および E38 ユーザー出口、SORTOUT 用には E19 および E39 ユーザー出口です (ただし、OUTFIL データ・セット用はありません)。これらは、特にテープ作業データ・セット分類に有効です。すべての DASD 作業データ・セット分類処理では、E19 および E38 ユーザー出口は無視されます。

これらの出口を使用して、ユーザー独自の、または導入先の入出力エラー・リカバリー・ルーチンを DFSORT に組み込むことができます。ユーザーの読み書きエラー・ルーチンは、区分データ・セット (ライブラリー) に存在する必要があります。

ユーザーのライブラリー・ルーチンは、関連フェーズとともに主記憶域内に呼び込まれます。訂正不能な入出力エラーが起こると、DFSORT は QSAM/BSAM または VSAM が渡すパラメーターと同じものを渡します。ユーザー・ルーチンがない場合に訂正不能な読み書きエラーが起こると、DFSORT はエラー・メッセージを出して終了します。

QSAM/BSAM の場合、次の情報がユーザーの同期エラー・ルーチンに渡されます。

- 汎用レジスター 0 と 1 は変更されません。これらのレジスターには QSAM/BSAM から渡された情報が含まれています (データ管理の資料を参照)。
- 汎用レジスター 14 には、DFSORT の戻りアドレスが含まれています。
- 汎用レジスター 15 には、ユーザーのエラー・ルーチンのアドレスが含まれています。

VSAM は、ユーザーが適宜、E18、E38、または E39 ユーザー出口を介して DFSORT に渡した EXLST マクロに指定されている任意のルーチンへ直接入ります。ユーザー・ルーチンは、レジスター 14 を介して VSAM へ戻る必要があります。詳細については、*z/OS DFSMS Macro Instructions for Data Sets* または *z/OS DFSMS: Using Data Sets* を参照してください。

### 読み取りエラー・ルーチン

これらのルーチンを、E18 および E38 ユーザー出口に関連付ける必要があります。これらのルーチンは、DFSORT に一定の制御ブロック情報を戻して、レコードをそのまま受け取るか、ブロックをスキップするか、または終了を要求するかを指示します。エラーの訂正を試みることもできます。

### 書き込みエラー・ルーチン

これらのルーチンを、E19 および E39 ユーザー出口に関連付ける必要があります。これらのルーチンは、DFSORT が終了する前に、SORTOUT に必要なすべてのタスク異常終了処理を実行できます。

## VSAM ユーザー出口機能

VSAM SORTIN、SORTINnn、および SORTOUT データ・セットで使用できる (ただし OUTFIL データ・セットでは使用できない) ユーザー出口が 3 つあり、これらの出口は、パスワードまたはユーザー出口を提供して、VSAM データ・セットのジャーナルを行います。これらのユーザー出口は、EODAD を除くその他の VSAM 出口機能を実行できます。ユーザー出口は、SORTIN 用には E18、SORTINnn またはコピー SORTIN 用には E38、SORTOUT 用には E39 です。

## 中間記憶域不足時の処置の決定

入力データ・セットの処理に使用可能な中間記憶域が不十分であると DFSORT が判別した場合に、DFSORT プログラムが行う処置を示すルーチンをユーザーが作成できます。テープ作業データ・セットを使用する分類処理の場合、このルーチンを E16 ユーザー出口に関連付ける必要があります。テープ・データ・セットを使用する分類処理の場合、ユーザーは現行レコードだけを分類するか、最後まで分類を行うか、または DFSORT を終了させるかの中から選択できます。詳細については、694 ページの『磁気テープ・ワークスペース容量の超過』を参照してください。

### データ・セットのクローズ

ユーザーは独自のユーザー・ルーチンを作成して、データ・セットをクローズし、必要なハウスキーピングを実行できます。これらのルーチンは E15、E17、E35、および E37 ユーザー出口に関連付ける必要があります。SORTOUT ラベルを作成するには、E19 および E39 ユーザー出口を使用します。SORTIN に使用するファイルの終了ルーチンがある場合は、それを E18 ユーザー出口に組み込みます。

### DFSORT の終了

出口ルーチンを作成して、すべてのレコードが処理される前に DFSORT を終了させることができます。これらのルーチンを E15、E16、E32、および E35 ユーザー出口に関連付ける必要があります。

**注:** ユーザー出口が、SmartBatch パイプ・データ・セットを使用しているアプリケーションの終了を要求する場合は、DFSORT はユーザー・アベンドのゼロを出して終了します。これにより、同じ SmartBatch パイプ・データ・セットにアクセスしている可能性のある他のアプリケーションに対して、システムが適切なエラー通知を行います。

---

### ユーザー出口のアドレッシング・モードおよび常駐モード

ブロック・セットまたはピアレッジ / ベールにより呼び出されたユーザー出口を 16 メガバイト境界より上または下に置き、24 ビットまたは 31 ビット・アドレッシングを使用できるようにし、さらにユーザー出口アドレス定数を使用できるようにするために、DFSORT には次のような機能が備わっています。

- DFSORT が確実に正しいアドレッシング・モードでユーザー出口に入るようにする場合、以下の規則に従ってください。
  - ユーザー出口名が MODS 制御ステートメントに指定されている場合は、そのルーチンのリンケージ・エディター属性が示すアドレッシング・モード (たとえば AMODE 31 が指定されている場合は、31 ビット・アドレッシングが有効) で、そのユーザー出口に入ります。
  - 出口のアドレスが 24 ビット・リストを介して DFSORT に渡される場合 (ロード済みの出口) は、有効な 24 ビット・アドレッシングでそのユーザー出口に入ります。
  - ユーザー出口のアドレスが、拡張パラメーター・リストを介して DFSORT に渡される場合 (ロード済みの出口) は、24 ビット・アドレッシング (リスト内のユーザー出口のビット 0 が 0 の場合) を有効にして、または 31 ビット・アドレッシング (ビット 0 が 1 の場合) を有効にして、そのユーザー出口に入ります。
- ユーザー出口は、24 ビットまたは 31 ビット・アドレッシングを有効にして、DFSORT に戻ることができます。DFSORT がレジスター 14 に入れた戻りアドレスを、使用する必要があります。
- ユーザー出口アドレス定数 (アセンブラーで作成された E15、E32、または E35 ユーザー出口のいずれかへ、変更されないまま渡される) を除き、DFSORT はユーザー出口パラメーター・リスト・アドレス (すなわち、パラメーター・リストへのポインターとパラメーター・リスト内のアドレス) を次のように処理します。



## ユーザー出口のアドレッシング・モードおよび常駐モード

- 24 ビット・アドレッシングを有効にしてユーザー出口に入る場合、DFSORT は、クリーンな (最初の 8 ビットがすべてゼロ) 24 ビット・アドレスをユーザー出口に渡します。このようなユーザー出口は、24 ビット・アドレスを DFSORT へ戻す必要があります。ユーザー出口が、31 ビット・アドレッシングを有効にして DFSORT に戻る場合は、24 ビット・アドレスは、クリーンである必要があります。
- 31 ビット・アドレッシングを有効にしてユーザー出口に入る場合、DFSORT は、クリーンな 24 ビット・アドレスをユーザー出口に渡します。このようなユーザー出口は、31 ビット・アドレスまたは 24 ビット・アドレスを DFSORT へ戻す必要があります。唯一の例外は、オプションのアドレス (例えば E18 SYNAD アドレス) が渡されることを示すために、高位バイトが使用される場合です。この場合、DFSORT が、24 ビット・アドレスをクリーンにします。

**注:** 従来の組み合わせ、またはテープ作業データ・セット分類のアプリケーションの場合、ユーザー出口は次の条件を満たしている必要があります。

- 16 メガバイト境界よりも下に常駐する。
- 24 ビット・アドレッシング・モードを使用する。
- ユーザー出口アドレス定数を使用しない。

---

## ユーザー出口ルーチンが DFSORT のパフォーマンスに与える影響

ユーザー出口ルーチンを作成する前に、次の点について考慮してください。

- ユーザー・ルーチンは、DFSORT が使用できる主記憶域を占有します。この主記憶域には制限があるため、DFSORT はデータを分類するために余分なパスを実行しなければならない場合があります。そのため、分類時間が増加します。
- ユーザー出口ルーチンは全体の実行時間を増加させます。『戻り不要』の戻りコードを DFSORT に渡すまで、多くのユーザー出口では、各レコードごとに 1 回ずつ制御をユーザー・ルーチンに渡すことに注意してください。ユーザー・ルーチンを設計する際には、これを念頭に置く必要があります。
- ユーザー出口ルーチンの代わりに INCLUDE、OMIT、INREC、OUTFIL、OUTREC、および SUM を使用すると、DFSORT はさらに効率的に機能することができます。

---

## ユーザー出口ルーチンに関する規則

ユーザー・ルーチンを作成する際には、次の点に留意してください。

- ユーザー作成のルーチンは標準の連係規則に従う必要があります。また必要なインターフェースを使用する必要があります。COBOL E15 および E35 ユーザー出口は、提供されている特別のインターフェースを使用します。
- E32 ユーザー出口を使用する場合、ユーザーの呼び出しプログラムはパラメーター・リストで、DFSORT へそのアドレスを渡す必要があります。
- 他のいずれかのユーザー出口を使用するには、MODS 制御ステートメントを使用して、ユーザー・ルーチンを該当するユーザー出口に関連付ける必要があります。173 ページの『MODS 制御ステートメント』を参照してください。

## ユーザー出口ルーチンに関する規則

- 代わりに、ユーザーの呼び出しプログラムがパラメーター・リストで、DFSORT へ E15、E18、E35、および E39 ユーザー出口のアドレスを渡すこともできます。
- ブロック・セットまたはピアレッジ / ベールが使用され、しかもユーザー出口が再入可能な場合は、DFSORT プログラム全体が再入可能になります。
- ISCI/ASCII 入力を使用している場合は、ユーザー出口でそのユーザー出口に表示されるデータは EBCDIC 形式になっています。特殊な英字の ISCI/ASCII 照合の解決に E61 ユーザー出口を使用している場合、置換される文字は EBCDIC で指定しますが、照合順序の結果は置換される文字の ISCI/ASCII 変換のバイト値により決まります。

## ユーザー出口ルーチンのロード方法

各ユーザー出口は、別個のプログラムとしてアセンブルまたはコンパイルする必要があります。ユーザー出口が独立に作動する場合は、MODS ステートメントで使用するメンバー名を用いて、その出口を区画データ・セット (ライブラリー) に別個にリンク・エディットしてください。ユーザー出口が同一フェーズ内で他のユーザー出口とともに作動する (たとえば E11、E15、および E17 ユーザー出口のすべてが同じ DCB を使用する) 場合は、それらをまとめて動的にリンク・エディットするように DFSORT に要求できます (MODS ステートメントを参照)。あるいは、次の規則に従って、区分データ・セットにそれらをまとめてリンク・エディットすることもできます。

1. RENT をエディター・パラメーターとして指定する。
2. ルーチンの外部入り口名 (たとえば、CSECT 名) を使用して、各ユーザー出口について、それぞれ ALIAS ステートメントを組み込む。
3. MODS ステートメントで、各ユーザー出口ごとに適切な ALIAS 名を指定する。

DFSORT は、各フェーズ中に実行するモジュールのリストに、ユーザー出口の名前および位置を組み込みます。どのユーザー出口も 1 つのプログラム・フェーズで 2 回以上ロードされることはありませんが、同じユーザー出口を異なるフェーズにロードすることはできます。たとえば、同じ読み取りエラー・ユーザー出口は両方のフェーズで使用できますが、1 つのフェーズで 2 回使用できません。

ユーザー出口に指定する長さには、ユーザー出口自体が使用する記憶域のほかに、ロード・モジュール外のユーザー出口 (入出力バッファー、COBOL ライブラリー・サブルーチンなど) が使用する記憶域も含めておく必要があります。MODS ステートメントにユーザー出口の DD 名を指定する場合、そのユーザー出口が入っているライブラリーを定義する DD ステートメントと同じ名前を指定します。たとえば、下記のようにします。

```
//MYLIB DD DSN=MYRTN, etc.  
.  
.  
.  
MODS E15=(MODNAME,500,MYLIB,N)
```

## ユーザー出口関係規則

ユーザー出口に入るには、DFSORT はレジスター 14 に入っている DFSORT 戻り点のアドレスと、レジスター 15 に入っているユーザー出口ルーチンのアドレスをロードします。その後で、レジスター 15 内のアドレスへの分岐が実行されます。

パラメーターの連係および連絡のために DFSORT により使用される汎用レジスターは、オペレーティング・システムの規則に従っている必要があります。ユーザー・ルーチンが制御を得たときは、汎用レジスターの内容は以下のようになります。

レジスター	内容
1	DFSORT は、パラメーター・リストのアドレスをこのレジスターに入れます。
13	DFSORT は、標準の保管域のアドレスをこのレジスターに入れます。この区域を使用して、ユーザー出口で使用するレジスターの内容を保管できます。この区域の最初のワードの下位 3 バイトには、文字 SMI が含まれています。
14	DFSORT 戻り点のアドレスが含まれています。
15	ユーザー出口のアドレスが含まれています。このレジスターはユーザー出口の基底レジスターとして使用できます。また、ユーザー出口はこのレジスターを使用して、戻りコードを DFSORT へ渡すこともできます。

レジスター 14 内の DFSORT 戻り点アドレスへの分岐を実行するか、あるいは RETURN マクロ命令を使用して、DFSORT へ制御を戻すことができます。また、ユーザー出口で複数の処置が使用できる場合、RETURN 命令を使用して戻りコードをセットすることもできます。

ユーザー出口は、使用するすべての汎用レジスターを保管します。SAVE マクロ命令を使用して、保管を行うことができます。レジスターを保管する場合、そのレジスターを復元する必要もあります。レジスターの復元には、RETURN マクロ命令を使用します。

### 連係の例

ユーザー出口を呼び出す場合、DFSORT は戻りアドレスを汎用レジスター 14 に、ユーザー・ルーチンの入り口点アドレスを汎用レジスター 15 に入れます。DFSORT は、レジスターの保管域アドレスをすでに汎用レジスター 13 に入れています。DFSORT は次に、ユーザー・ルーチンへの分岐を行います。

E15 ユーザー出口のユーザー・ルーチンには、次のようなアセンブラー命令を組み込むことができます。

```

ENTRY E15
.
.
E15 SAVE (5,9)
.
.
RETURN (5,9)

```

このコーディングは、汎用レジスター 5 から 9 の内容を保管し復元します。マクロ命令は、次のようなアセンブラー言語コードに拡張されます。

## ユーザー出口ルーチンに関する規則

```
ENTRY E15
.
.
E15 STM 5,9,40(13)
.
.
LM 5,9,40(13)
BR 14
```

ユーザー出口で複数の処置を実行できる場合、ユーザー・ルーチンは汎用レジスター 15 に戻りコードをセットして、取るべき処置を DFSORT に知らせます。次のマクロ命令を使用すると、戻りコード 13 をレジスター 15 に入れて、DFSORT へ戻ることができます。

```
RETURN RC=12
```

本章の連係規則およびマクロ命令の詳細については、*アプリケーション開発マクロ解説書* を参照してください。

## ユーザー出口ルーチンのダイナミック・リンク編集の方法

任意の言語で作成された任意の出口ルーチンを動的にリンク・エディットして、レコードやパラメーターの位置またはアドレスを汎用レジスター 1 に入れて渡し、戻りコードをレジスター 15 に入れて渡すことができます。COBOL で作成された E15 および E35 ユーザー出口は、これには含まれません。

ダイナミック・リンク編集は、リンク・エディット・オプション T の AMODE 31 や RMODE 31 のサポートは行いません。DFSORT により まとめてリンク・エディットされたユーザー出口は、16 メガバイト境界より上にはロードされず、また、31 ビット・アドレッシング・モードに入ることもできません。S オプションでリンク・エディットされたユーザー出口は、オブジェクト・モジュールの AMODE および RMODE 属性を保持するので、ロード・モジュールの RMODE に応じて、16 メガバイトの上または下にロードされます。これらはユーザー出口のアドレッシング・モードに入ります。

### 注:

1. ダイナミック・リンク編集の場合、ブロック・セット手法は使用されません。
2. ダイナミック・リンク編集は、コピー処理には使用できません。

リンク・エディット・オプション T がユーザー出口ルーチンに指定されている場合、そのルーチンには関連するプログラム・ユーザー出口の名前をもつ入り口点を含める 必要があります。これは、DFSORT の特殊なダイナミック・リンク編集要件を満たすためです。たとえば、リンク・エディット・オプション T が E35 の MODS ステートメントに指定されている場合、E35 ユーザー出口に関連したユーザー出口ルーチンには、次のようなアセンブラー命令が組み込まれている必要があります。

```
ENTRY E35
E35 .
.
.
または
E35 CSECT
.
.
```

他のすべての場合、ユーザー出口は、関連するプログラム・ユーザー出口の名前をもつ入り口点を必要としません。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

下記のプログラム・ユーザー出口を、DFSORT 入力フェーズで使用できます。

E11  
E15  
E16  
E17  
E18  
E19  
E61

これらのユーザー出口については、順次説明します。アプリケーションにどの特定ユーザー出口を使用できるかを判別するには、367 ページの表 48 および 367 ページの表 49 を参照してください。

### E11 ユーザー出口: データ・セットのオープン / ルーチンの初期設定

このユーザー出口のルーチンを使用して、入力フェーズの他のルーチンが必要とするデータ・セットをオープンできます。また、他のルーチンを初期設定するために使用することもできます。しかし、戻りコードは使用されません。

**注:** リンケージ・エディターの特殊要件を避けるため (371 ページの『ユーザー出口ルーチンに関する規則』を参照)、これらの機能を、独立した E11 ルーチンではなく、E15 ユーザー出口に組み込むことができます。

### E15 ユーザー出口: レコードの引き渡しまたは変更 (分類アプリケーションおよびコピー・アプリケーションの場合)

COBOL で E15 ユーザー出口を作成する場合は、396 ページの『COBOL ユーザー出口ルーチン』および 400 ページの『COBOL E15 ユーザー出口: レコードの引き渡しまたは変更』を参照してください。

EXITCK オプションは、DFSORT がユーザー出口 E15 からの特定の戻りコードを解釈する方法に影響を与えます。ここでは、あいまいにならないように、ユーザーのシステムで IBM のデフォルトである EXITCK=STRONG が選択されているものと仮定します。EXITCK=STRONG および EXITCK=WEAK を使用する様々な状況での E15 戻りコードの詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

DFSORT は、新しいレコードが入力フェーズに呼び込まれるたびに、E15 ユーザー出口ルーチンに入ります。DFSORT は、ユーザー出口が戻りコード 8 を使用して、戻らないように DFSORT に指示するまで、E15 に入ります (入力レコードがない場合でも)。

論理の流れの詳細については、365 ページの図 17 を参照してください。

E15 ユーザー出口の用途は次のとおりです。

- レコードを入力データ・セットへ追加する。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

- 入力データ・セット全体を DFSORT へ引き渡す。
- レコードを入力データ・セットから削除する。
- 入力データ・セット内のレコードを変更する。

### 注:

1. E15 ユーザー出口が可変長レコードを処理する場合は、変更または挿入する各レコードの先頭に 4 バイトの RDW を入れてから、レコードを DFSORT に戻してください。RDW の形式については、*z/OS DFSMS: Using Data Sets* または *System Programming Reference* を参照してください。(あるいは、レコードを最大長まで埋め込み処理して、固定長として処理することもできます。)
2. DFSORT は、指定された値かそのデフォルトを RECORD ステートメントの L2 に使用して、E15 ユーザー出口が DFSORT に戻すレコードの長さを判別します。固定長レコードの場合、E15 ユーザー出口が変更または挿入する各レコードの長さが、必ず L2 の指定値またはデフォルトと一致するようにしてください。可変長レコードの場合、E15 ユーザー出口が変更または挿入する各レコードの RDW が、必ず L2 の指定値またはデフォルト以下になるようにしてください。E15 ユーザー出口が間違ったレコード長を DFSORT に渡すと、望ましくない短縮またはアベンドが起きることがあります。  
L2 値の詳細については、340 ページの『RECORD 制御ステートメント』を参照してください。
3. E15 ユーザー出口を使用して、すべてのレコードを DFSORT に渡す場合は、SORTIN DD ステートメントを省略できます。この場合は、プログラム制御ステートメントに RECORD ステートメントを組み込んでください。
4. アセンブラー・プログラムから DFSORT を呼び出し、E15 ユーザー出口のアドレスをパラメーター・リストで渡す場合、DFSORT は SORTIN データ・セットを無視し、MODS ステートメントで E15 を指定している場合は終了します。
5. SORTIN DD ステートメントが省略されているか、あるいはそれが無視される場合は、すべての入力レコードは、ユーザー出口 E15 のユーザー・ルーチンを介して DFSORT へ渡されます。各入力レコードのアドレスは順番に汎用レジスター 1 に入れられ、戻りコード 12 で DFSORT へ戻ります。最後のレコードが渡された後に DFSORT が E15 ユーザー出口へ戻ると、レジスター 15 には「戻り不可」を示す戻りコード 8 を入れて、DFSORT へ戻ります。
6. DFSORT は、戻りコード 8 を受け取るまでは、E15 ユーザー出口に再入を続けます。ただし、STOPAFT が有効な場合は、STOPAFT カウントが満たされると (戻りコード 12 が返されても) それ以上のレコードは DFSORT には挿入されません。
7. 可変長 VSAM レコードについては、RDW を作成します (*z/OS DFSMS: Using Data Sets* を参照)。

### E15 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報

新しいレコードが入力フェーズに呼び込まれるたびに、E15 ユーザー出口ルーチンが呼び出されます。これが呼び出されるたびに、DFSORT は次の 2 つのワードをユーザー・ルーチンに渡します。

- **新しいレコードのアドレス。** E15 ユーザー出口に渡すレコードがなくなると、入力の終わりになります。DFSORT は、E15 ユーザー出口に入る前に、このアドレスをゼロにセットして入力の終わりを示します。入力データ・セットにレコード

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

がない場合 (または入力データ・セットがない場合) には、E15 に最初に入った時点でこのアドレスはゼロになっています。

入力の終わりになっても、ユーザーが戻りコード 8 を返すまで、DFSORT はユーザー出口ルーチンへ入り続けます。

E15 ユーザー出口では、新しいレコードのアドレスを変更できません。

- **ユーザー出口アドレス定数。** パラメーター・リストのアドレス定数を用いて DFSORT を呼び出すと、E15 ユーザー出口に最初に入ったときに、アドレス定数が E15 に渡されます。E15 ユーザー出口に入ったときはいつでも、このアドレス定数を変更できます。アドレス定数は、その後 E15 ユーザー出口へ入るたびに渡されるほか、E35 ユーザー出口に最初に入ったときにも渡されます。たとえば、ユーザーは、動的ストレージを獲得し、これを E15 ユーザー出口で使用し、そのアドレスを E35 ユーザー出口に渡すことができます。

**注:** ユーザー出口アドレス定数を、テープ作業データ・セット分類アプリケーションに使用できません。

DFSORT は、レコード・アドレスおよびユーザー・アドレス定数が入ったパラメーター・リストのアドレスを、汎用レジスター 1 に入れます。このリストは、2 フルワードの長さで、フルワード境界から始まります。パラメーター・リストの形式は次のとおりです。

表 50. E15 ユーザー出口パラメーター・リスト

1 ~ 4 バイト	新しいレコードのアドレス
5 ~ 8 バイト	ユーザー出口アドレス定数

## E15 の戻りコード

E15 ユーザー・ルーチンは、戻りコードを DFSORT へ渡す必要があります。E15 ユーザー出口の戻りコードは次のとおりです。

戻りコード	説明
00 (X'00')	処置をとらない / レコードを変更する
04 (X'04')	レコードを削除する
08 (X'08')	戻らない
12 (X'0C')	レコードを挿入する
16 (X'10')	DFSORT を終了する

### 0: 処置をとらない

レコードを変更せずに DFSORT にそのまま保持させたい場合は、そのレコードのアドレスを汎用レジスター 1 に入れ、戻りコード 0 (ゼロ) で DFSORT に戻ります。

### 0: レコードを変更する

レコードを DFSORT に戻す前に変更したい場合は、ユーザー・ルーチンはそのレコードを作業域へ移して、変更を行い、変更したレコードのアドレスを汎用レジスター 1 に入れ、戻りコード 0 (ゼロ) で戻ります。

### 4: レコードを削除する

DFSORT に入力データ・セットからのレコードを削除させたい場合は、戻りコード 4 で DFSORT へ戻ります。レコードのアドレスを汎用レジスター 1 に入れる必要はありません。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

### 8: 戻らない

DFSORT は戻りコード 8 を受け取るまで、制御をユーザー・ルーチンに戻し続けます。その後は、DFSORT アプリケーションの処理中に、そのユーザー出口が再び使用されることはありません。戻りコード 8 で戻る場合は、汎用レジスター 1 にアドレスを入れる必要はありません。データ・セットの終わりの後でレコードを挿入しない限り、ユーザーは、プログラムがデータ・セットの終わりを示した際に、戻りコード 8 を渡す必要があります。これを行うには、パラメーター・リストにゼロのアドレスを入れて、ユーザー・ルーチンに渡します。

処理する入力レコードがまだ残っている段階でユーザー出口ルーチンが戻りコード 8 を DFSORT に渡すと、残りのレコードは DFSORT により処理されますが、ユーザー出口には 渡されません。

### 12: レコードを挿入する

ユーザー・ルーチンにアドレスが渡されたばかりのレコードの前にレコードを追加するには、追加するレコードのアドレスを汎用レジスター 1 に入れて、戻りコード 12 で DFSORT に戻ります。DFSORT は前と同じレコード・アドレスでユーザー・ルーチンへ戻り続けるので、ユーザー・ルーチンはその場所にさらにレコードを挿入したり、現行レコードを変更できます。入力データ・セットの最後のレコードの後 (DFSORT がパラメーター・リストにゼロのアドレスを入れた後) で、追加を行うことができます。DFSORT は、戻りコード 8 を渡されるまで、ユーザー・ルーチンへ戻り続けます。

### 16: DFSORT を終了する

DFSORT を終了したい場合は、コード 16 で戻ります。すると DFSORT は、戻りコード 16 で呼び出しプログラムまたはシステムへ戻ります。

各種の状況での戻りコードの意味の詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

## E15 ユーザー出口のための記憶域の使用法

DFSORT は、E15 ユーザー出口ルーチンに渡すパラメーター・リストとレコードのために、(GETMAIN または STORAGE OBTAIN を使用して) 記憶域を確保します。DFSORT が確保した記憶域を変更または解放しないでください。

更新したレコードを DFSORT に渡すときなどのために、E15 ユーザー出口ルーチンが使用する記憶域を確保する必要がある場合は、次の方法が使用できます。

1. 出口ルーチンが初めて呼び出されたときに、必要な記憶域を確保します。
2. 出口ルーチンが呼び出されるたびに、確保した記憶域を使用します。
3. 戻りコード 8 を DFSORT に返す前に、記憶域を開放します。

**注:** 記憶域を確保したら、そのアドレスをユーザー出口アドレス定数に保管しておき、それ以降ユーザー出口が呼び出されるたびに復元できます。

## E16 ユーザー出口: 中間記憶装置の計算違いの処理

テープ作業データ・セット分類の場合、または作業データ・セットを使用しないピアレッジ / ベール分類の場合、ユーザーは、このユーザー出口のルーチンを使用して、主記憶域および中間記憶域の一定の大きさで処理できるレコード数の計算見積もりを超える分類処理の場合に、どのような処置を取ればよいかを決定します。作業データ・セットの分類の場合、このユーザー出口は無視されます。DFSORT が



## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

WRKSEC オプションを使用して、2 次割り振りが可能かどうかを判別するためです。79 ページの『SORTWKdd DD ステートメント』を参照してください。また、694 ページの『磁気テープ・ワークスペース容量の超過』を参照してください。

**注:** 磁気テープを使用する際には、システムが 2400 フィートの長さのテープを想定していることを忘れないでください。それ以外の長さのテープを使用すると、Nmax 数値が正しくなりません。短いテープの場合は、『NMAX EXCEEDED』が表示される前に容量を超えていることがあります。

### E16 の戻りコード

E16 ユーザー・ルーチンは、戻りコードを DFSORT へ渡す必要があります。E16 ユーザー出口の戻りコードは次のとおりです。

戻りコード	説明
00 (X'00')	現行レコードだけを分類する
04 (X'04')	追加のレコードの分類を試みる
08 (X'08')	DFSORT を終了する

#### 0: 現行レコードだけを分類する

DFSORT に、処理可能と見積もった入力データ・セットの一部だけの処理を継続させたい場合は、戻りコード 0 (ゼロ) で戻ります。メッセージ ICE054I には、分類が継続されているレコードの数が含まれています。SKIPREC を使用してすでに分類済みのレコードをスキップし、データ・セットの残りの部分を、1 回または複数回にわたって分類できます。次に、分類出力を組み合わせて、操作を完了できます。

#### 4: 引き続きレコードの分類を試みる

DFSORT にすべての入力データ・セットの処理を継続させたい場合は、戻りコード 4 で戻ります。テープを使用する場合は、DFSORT が処理を完了するのに十分なスペースが使用できる必要があります。十分なスペースが使用できない場合は、DFSORT はメッセージを生成して終了します。694 ページの『磁気テープ・ワークスペース容量の超過』を参照してください。

#### 8: DFSORT を終了する

DFSORT を終了させたい場合は、戻りコード 8 で戻ります。すると DFSORT は、戻りコード 16 で呼び出しプログラムまたはシステムへ戻ります。

## E17 ユーザー出口: データ・セットのクローズ

E17 ユーザー出口ルーチンへは、入力フェーズの終わりで 1 回だけ入ります。このルーチンを使用して、そのフェーズの別のルーチンが使用したデータ・セットをクローズしたり、ユーザー・ルーチンのハウスキーピング機能を実行できます。

**注:** リンケージ・エディターの特長要件を避けるため (371 ページの『ユーザー出口ルーチンに関する規則』を参照)、これらの機能を、独立した E17 ユーザー出口ルーチンではなく、E15 ユーザー出口に組み込むことができます。

## E18 ユーザー出口: 入力データ・セットの処理

このユーザー出口を使用すると、QSAM/BSAM および VSAM 入力データ・セットの特殊な入出力条件を処理できます。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

### QSAM/BSAM による E18 ユーザー出口の使用

このユーザー出口ルーチンは、3 つのデータ制御ブロック (DCB) フィールド (SYNAD、EXLST、および EROPT) の仕様を含むパラメーター・リストを、DFSORT に渡すことができます。E18 ユーザー出口ルーチンは、4 番目の DCB フィールド (EODAD) を DFSORT へ渡すこともできます。

**注:** ディスク分類手法を使用している場合は、EROPT オプションは無視されます。

DFSORT がパラメーター・リストを獲得できるように、各フェーズの最初にユーザー・ルーチンに入ります。フェーズの処理中、パラメーター・リストに示された箇所ですべてルーチンに入ります。たとえば、EXLST オプションを選択すると、DFSORT は分類 (入力) フェーズの早い段階で E18 ユーザー出口ルーチンに入ります。DFSORT は、EXLST アドレスを含むパラメーター・リストを取り出します。フェーズの後半で、データ・セットがオープンされると、DFSORT は再び EXLST アドレスでユーザー・ルーチンに入ります。

**E18 ユーザー出口でユーザー・ルーチンが DFSORT へ渡す情報:** 制御を DFSORT に戻す前に、ユーザー・ルーチンはパラメーター・リストのアドレスを汎用レジスター 1 に入れて、パラメーター・リストの DCB フィールドを渡します。パラメーター・リストはフルワード境界から始まり、フルワードの整数倍の長さで指定します。各ワードの高位バイトには、パラメーターを識別する文字コードが含まれます。1 つ以上のワードを省略できます。すべてゼロのワードは、リストの終わりを示します。

VSAM パラメーターを指定すると、そのパラメーターは受け入れられますが無視されます。

リストの形式は次のとおりです。

1 バイト	2 バイト	3 バイト	4 バイト
01	SYNAD フィールド		
02	EXLST フィールド		
03	00	00	EROPT コード
04	EODAD フィールド		
00	00	00	00

### SYNAD

ユーザーの読み取り同期エラー・ルーチンの位置が含まれています。オペレーティング・システムがエラーの訂正を試みて、失敗に終わった後のみ、このルーチンに入ります。このルーチンは、E18 ユーザー出口ルーチンの一部としてアセンブルされる必要があります。ルーチンが制御を受け取る場合、レジスター 13 が指し示す保管域にレジスターを 保管できません。

### EXLST

ラベルの検査およびデータ管理プログラムが処理しない他のタスクを実行するために使用するルーチンを指し示すポインターのリストの位置が含まれています。リストおよびリストが指し示すルーチンは、ユーザーの読み取りエラー・ルーチンに組み込みます。このパラメーターは、最初の SORTIN データ・セットのオープンに関連した EXLST ルーチンに対してだけ使用されます。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

### EROPT

訂正不能な読み取りエラーが起こったときに、DFSORT がどのような処置をとるべきかを示します。次の 3 つの処置およびそれらに対応するコードが可能です。

**X'80'** レコード (ブロック) をそのまま受け入れる

**X'40'** レコード (ブロック) をスキップする

**X'20'** プログラムを終了する

このパラメーターを DCB フィールド・リストに組み込む場合は、上のコードのどれか 1 つを、ワードの 4 バイトに入れます。ワードの 2 と 3 バイトにはゼロが入っている必要があります。

EROPT オプションを使用する際には、SYNAD フィールドおよび EODAD フィールドの 2 ~ 4 バイトに適切なアドレスを入れる必要があります。あるいは、使用できるルーチンがない場合は、2 と 3 バイトにはゼロを、4 バイトには X'01' を入れます。アセンブラー命令の DC AL3(1) を使用して、2 ~ 4 バイトをセットできます。

### EODAD

ユーザーのファイルの終わりルーチンのアドレスが含まれています。EODAD を指定する場合は、ファイルの終わりルーチンをユーザー・ルーチンに組み込む必要があります。

DCB フィールドの詳細が、*z/OS DFSMS Macro Instructions for Data Sets* に記載されています。

### VSAM による E18 ユーザー出口の使用

DFSORT への入力が VSAM データ・セットの場合は、E18 ユーザー出口を使用して、各種の VSAM ユーザー出口機能を実行したり、VSAM 入力 ACB にパスワードを挿入できます。

**E18 ユーザー出口の制約事項:** パスワードがユーザー出口から入力され、ブロック・セットが選択されない場合は、データ・セットを初期設定フェーズの間にオープンできません。つまり、プログラムでは必要な計算が行えないために、MAINSIZE|SIZE=MAX を使用できません。

**E18 ユーザー出口でユーザー・ルーチンが DFSORT へ渡す情報:** DFSORT へ戻るときに、パラメーター・リストのアドレスを汎用レジスター 1 に入れます。

1 バイト	2 ~ 4 バイト
05	VSAM ユーザー出口リストのアドレス
06	パスワード・リストのアドレス
00	000000

代わりに QSAM パラメーターを渡した場合、パラメーターは受け入れられますが無視されます。

どちらのアドレス項目も省略できます。両方の項目を含める場合は、順序は任意です。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

**E18 パスワード・リスト:** ユーザー・ルーチンに組み込まれるパスワード・リストは、次の形式である必要があります。

ハーフワード境界の 2 バイト:

---

リスト内の項目数

---

それに続く 16 バイトの項目:

---

8 バイト: DD 名

---

8 バイト: パスワード

---

DD 名フィールドの最後のバイトは、DFSORT により破棄されます。プログラムの実行中は、どの時点でも、このリストを変更できません。この機能を使用している場合は、MAINSIZE/SIZE=MAX を使用できません。

**E18 ユーザー出口リスト:** VSAM EXLST マクロ命令を使用して VSAM ユーザー出口リストを作成し、VSAM ユーザー出口機能を実行するユーザー・ルーチンのアドレスを指定する必要があります。VSAM はユーザー・ルーチンに直接分岐します。このルーチンは、レジスター 14 を介して VSAM に戻ります。

入力データ・セットで使用できる VSAM ユーザー出口機能は、EODAD を除いてすべて使用できます。EODAD 処理を実行する必要がある場合は、LERAD ユーザー出口を作成し、RPL の FDBK フィールドに X'04' が入っているかどうかを検査してください。この値は入力 EOD を示します。DFSORT もこのフィールドを必要とするため、VSAM へ戻るときにこのフィールドを変更できません。

詳細については、*z/OS DFSMS Macro Instructions for Data Sets*を参照してください。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

図 18 は、DFSORT へ制御を戻すためにユーザー・プログラムが使用できるコーディングの例です。

```
ENTRY    E18
        .
        .
E18      LA      1,PARMLST
        RETURN
        CNOP    0,4
PARMLST  DC      X'01'
        DC      AL3(SER)
        DC      X'02'
        DC      AL3(LST)
        DC      X'03'
        DC      X'000080'          EROPT CODE
        DC      A(0)
        DC      X'04'
        DC      AL3(QSAMEOD)
        DC      X'05'
        DC      AL3(VSAMEXL)
        DC      X'06'
        DC      AL3(PWDLST)
        DC      A(0)
        .
        .
VSAMEXL  EXLST   SYNAD=USYNAD,LERAD=ULERAD
PWDLST   DC      H'1'
        DC      CL8'SORTIN'        SORTIN DDNAME
        DC      CL8'INPASS'        SORTIN PASSWORD
USYNAD   ...
ULERAD   ...
SER      ...
LST      DC      X'85',AL3(RTN)    EXLST ADDRESS LIST1
RTN      ...
QSAMEOD  ...                      EXLST ROUTINE
                                         QSAM END OF FILE ROUTINE
```

図 18. E18 ユーザー出口の例

<sup>1</sup> X'85'= X'80' + X'05' ただし、

X'80' は、この項目がリストの LAST ENTRY であることを意味しています。

X'05' は、このユーザー出口がデータ制御ブロック・ユーザー出口であることを意味しています。

詳細については、*z/OS DFSMS: Using Data Sets*を参照してください。

## E19 ユーザー出口: 作業データ・セットへの出力の処理

DFSORT が作業データ・セットへの書き込みエラーを訂正できない場合に、このユーザー出口を使用して、入力フェーズの書き込みエラー状態を処理します。これが使用されるのは、テープ作業データ・セット分類に対してだけです。

### QSAM/BSAM による E19 ユーザー出口の使用

このユーザー出口ルーチンは、2 つの DCB フィールド (SYNAD および EXLST) の仕様を含むパラメーター・リストを DFSORT に渡すことができます。DFSORT がパラメーター・リストを獲得できるようにするために、入力フェーズの初期に、まずユーザー・ルーチンに入ります。フェーズの後半、パラメーター・リストのオプションに示された箇所ですでにルーチンに入ります。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

**E19 ユーザー出口でユーザー・ルーチンが DFSORT へ渡す情報:** 制御を DFSORT に戻す前に、ユーザー・ルーチンはパラメーター・リストのアドレスを汎用レジスター 1 に入れて、パラメーター・リストの DCB フィールドを渡します。リストはフルワード境界から始まり、フルワードの整数倍の長さで指定します。各ワードの 1 バイト目には、パラメーターを識別する文字コードが含まれます。どちらのワードも省略できます。すべてゼロのワードは、リストの終わりを示します。

VSAM パラメーターが渡された場合、そのパラメーターは受け入れられますが無視されます。

リストの形式は次のとおりです。

1 バイト	2 バイト	3 バイト	4 バイト
01	SYNAD フィールド		
02	EXLST フィールド		
00	00	00	00

### SYNAD

このフィールドには、ユーザーの書き込み同期エラー・ルーチンの位置が含まれています。オペレーティング・システムがエラーの訂正を試みて、失敗に終わった後のみ、このルーチンに入ります。このルーチンは、ユーザー・ルーチンの一部としてアセンブルする必要があります。

### EXLST

EXLST フィールドには、ポインターのリストの位置が含まれています。これらのポインターは、ラベルを処理したり、データ管理プログラムが処理しない他のタスクを実行したりするために使用されるルーチンを指し示します。このリストおよびリストが指し示すルーチンは、ユーザー・ルーチンの一部として組み込む必要があります。

これらの DCB フィールドの詳細については、*z/OS DFSMS Macro Instructions for Data Sets* に記載されています。

## E61 ユーザー出口: 制御フィールドの変更

このユーザー出口ルーチンを使用して、レコード内の任意の制御フィールドを長くしたり、短くしたり、あるいは変更できます。このルーチンにより変更される制御フィールドについては、SORT または MERGE 制御ステートメントの s パラメーターに E オプションを指定します。169 ページの『MERGE 制御ステートメント』および 345 ページの『SORT 制御ステートメント』を参照してください。ユーザー・ルーチンが制御フィールドを変更すると、その後 DFSORT は、指定された形式を使用して、レコードを昇順に照合します。<sup>19</sup>

### 注:

1. E61 ルーチンは、D1 形式の EFS フィールドでは使用されません。
2. 比較の前にユーザー E61 ルーチンは制御フィールドを変更しますが、ユーザーの元のレコードは変更されません。

19. 従来の組み合わせまたはテープ作業データ・セット分類の場合、E が指定された制御フィールドは、実際に指定された形式がどのようなものであっても、2 進数バイトの形式として処理されます。

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

3. ロケール処理を SORT または MERGE フィールドに使用する場合は、E61 ユーザー出口は使用しないでください。DFSORT のロケール処理は、E61 ユーザー出口に関する要求を削除する場合があります。ロケール処理の詳細については、181 ページの『OPTION 制御ステートメント』を参照してください。

### E61 ユーザー出口のいくつかの使用法

ユーザー・ルーチンは、浮動小数点制御フィールドを正規化したり、その他のタイプの制御フィールドを任意の方法で変更できます。オペレーティング・システムで使用する標準データ形式をよく理解してから、制御フィールドの変更を行ってください。

たとえば、EBCDIC データの照合順序を変更して、国別文字のアルファベット照合順序を使用できるようにする場合、ALTSEQ 制御ステートメントを使用することで、E61 ユーザー出口ルーチンを使用する必要はなくなります (120 ページの『ALTSEQ 制御ステートメント』を参照してください)。

### E61 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報

DFSORT は、パラメーター・リストのアドレスを汎用レジスター 1 に入れます。このリストはフルワード境界で始まり、3 フルワードの長さです。E61 ユーザー出口のパラメーター・リストは次のとおりです。

1 バイト	2 バイト	3 バイト	4 バイト
00	00	00	制御フィールド番号
00	制御フィールド・イメージのアドレス		
未使用		制御フィールドの長さ	

制御フィールドの長さは、より汎用的な変更ルーチンを作成することを可能にします。

制御フィールドを変更するためには、示されているアドレスで制御フィールド・イメージを変更します (アドレス自体を変更しても、効果はありません)。

制御フィールド番号は、SORT または MERGE ステートメントのすべてのフィールドに関連しています。たとえば、次のように指定すると仮定します。

```
SORT FIELDS=(4,2,CH,A,8,10,CH,E,25,2,BI,E)
```

フィールド番号 2 および 3 が、ユーザー出口 E61 に渡されます。

2 進数フィールド以外のすべてのフィールドでは、DFSORT がユーザー・ルーチンへ渡すバイト数の合計は、SORT または MERGE ステートメントの *m* パラメーターで指定した長さと等しくなります。

ユーザー・ルーチンに渡された 2 進数フィールドは、すべて整数バイトです。制御フィールドのすべてのビットが入っているすべてのバイトが渡されます。制御フィールドが 256 バイトよりも長い場合は、DFSORT は制御フィールドを 256 バイトずつのフィールドに分け、1 回に 1 つずつユーザー・ルーチンに渡します。

ユーザー・ルーチンは、制御フィールドの長さを物理的に変更できません。照合するために長さを増やす必要がある場合は、前もってその値を、SORT または MERGE ステートメントの *m* パラメーターに指定しておく必要があります。制御フ

## アセンブラー・ユーザー出口ルーチン (入力フェーズのユーザー出口)

フィールドを短くする必要がある場合は、指定された長さまで埋め込みを行ってから、DFSORT へそのフィールドを戻す必要があります。ユーザー・ルーチンは、ルーチンに入ったときと同じ数のバイトが含まれているフィールドを DFSORT に戻す必要があります。

E61 ユーザー出口が使用される場合は、レコードは常に昇順で配列されます。それ以外の順序が必要であれば、フィールドをさらに変更できます。たとえば、2 進数制御フィールドに対して予定の変更を行った後で、かつ DFSORT へ順序を戻す前に、全ビットを反転すれば、そのフィールドは、396 ページの図 24 の E61 の例に示されているとおり、実質的に降順に照合されます。

E61 を使用して特殊な英字の ISCI/ASCII 照合を解決する場合、置換された文字は EBCDIC で指定しますが、順序付けは置換された文字の ISCI/ASCII 変換のバイト値により異なることに注意してください。

---

## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

下記のプログラム・ユーザー出口を、DFSORT 出力フェーズで使用できます。

E31  
E32  
E35  
E37  
E38  
E39

これらのユーザー出口の機能については、順次説明します。

### E31 ユーザー出口: データ・セットのオープン / ルーチンの初期設定

このユーザー出口ルーチンを使用して、出力フェーズ内の別のユーザー・ルーチンで必要とするデータ・セットをオープンしたり、別のユーザー・ルーチンを初期設定できます。戻りコードは使用されません。

**注:** リンケージ・エディターの特殊要件を避けるため (371 ページの『ユーザー出口ルーチンに関する規則』を参照)、これらの機能を、独立した E31 ルーチンではなく、E35 ユーザー出口に組み込むことができます。

### E32 ユーザー出口: 組み合わせのみの入力の処理

このユーザー出口は、プログラムから呼び出された組み合わせ操作でだけ使用でき、MODS ステートメントには指定できません。E32 ユーザー出口が活動化されると、この出口は、組み合わせに対してすべての入力を提供します。E32 ユーザー出口が使用されると、DFSORT は SORTINnn データ・セットを無視します。

(1) MERGE 制御ステートメントの FILES=n オプション、または (2) 24 ビット・パラメーター・リストの X'4' 項目のいずれかを使用して、組み合わせたい入力ファイルの数を指定します。E32 ユーザー出口ルーチンは、DFSORT の要求に応じて、これらのファイルにレコードを挿入します。

入力が可変長レコードの場合は、組み合わせを行う前に、各レコードの先頭には 4 バイトの RDW が含まれます。RDW の形式については、*z/OS DFSMS Macro*



## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

*Instructions for Data Sets* を参照してください。(あるいは、レコードを固定長として宣言し、そのレコードに最大長になるように埋め込みを行うこともできます。)

論理の流れの詳細については、365 ページの図 17 を参照してください。

### E32 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報

組み合わせプログラムが新しい入力レコードを必要とするたびに、E32 ユーザー出口ルーチンに入ります。DFSORT はユーザー・ルーチンに次の 3 つのワードを渡します。

- **入力に使用される次のファイルの増分。** ファイル増分は 0,4,8,...,N-4、(ただし、N は入力ファイル番号の 4 倍) です。したがって、増分 0 (ゼロ) は最初の入力ファイルを表し、4 は 2 番目のファイル、8 は 3 番目のファイル (以下同様) を表します。
- **次の入力レコードの。** ユーザー・ルーチンは、使用する各入力ファイルごとに別個の入力バッファを提供します。各ファイルから最初のレコードを DFSORT に渡すまでは、ファイルの最初のレコードが含まれている入力バッファを変更できません。
- **ユーザー出口アドレス定数。** パラメーター・リストのアドレス定数を用いて DFSORT を呼び出すと、E32 ユーザー出口に最初に入ったときに、アドレス定数が E32 に渡されます。E32 ユーザー出口に入ったときはいつでも、このアドレス定数を変更できます。アドレス定数は、その後 E32 ユーザー出口および E35 ユーザー出口に渡されます。たとえば、ユーザーは、動的ストレージを獲得し、これを E32 ユーザー出口で使用し、そのアドレスを E35 ユーザー出口に渡すことができます。

**注:** ユーザー出口アドレス定数を、従来の組み合わせアプリケーションに使用できません。

DFSORT は、ファイル増分、レコード・アドレス、およびユーザー・アドレス定数が入ったパラメーター・リストのアドレスを、汎用レジスター 1 に入れます。このリストは 3 フルワードの長さで、フルワード境界から始まります。パラメーター・リストの形式は次のとおりです。

表 51. E32 ユーザー出口パラメーター・リスト

1 ~ 4 バイト	入力に使用される次のファイルの増分
5 ~ 8 バイト	次の入力レコードのアドレス
9 ~ 12 バイト	ユーザー出口アドレス定数

DFSORT へ制御を戻す前に、次のことを行う必要があります。

- 要求された入力ファイルの次の入力レコードのアドレスを、パラメーター・リストの 2 番目のワードへ入れる
- 戻りコードをレジスター 15 へ入れる

### E32 の戻りコード

E32 ルーチンは、戻りコードを DFSORT へ渡す必要があります。E32 ユーザー出口の戻りコードは次のとおりです。

戻りコード      説明

## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

- 08 (X'08') 要求されたファイルの入力の終わり
- 12 (X'0C') レコードを挿入する
- 16 (X'10') DFSORT を終了する

### 8: 要求されたファイルの入力の終わり

DFSORT は入力ファイルごとに戻りコード 8 を受け取るまで、制御をユーザー・ルーチンに戻し続けます。その後は、DFSORT アプリケーションの処理中にそのユーザー出口が再び使用されることはありません。戻りコード 8 で戻る場合は、パラメーター・リストの 2 番目のワードにアドレスを入れる必要はありません。

### 12: レコードを挿入する

要求された入力ファイルからレコードを追加するには、追加するレコードのアドレスをパラメーター・リストの 2 番目のワードに入れて、戻りコード 12 で DFSORT に戻ります。DFSORT は、すべての入力ファイルについて戻りコード 8 が渡されるまで、ユーザー・ルーチンへ戻り続けます。

### 16: DFSORT を終了する

DFSORT を終了したい場合は、コード 16 で戻ります。次に、DFSORT は、戻りコード 16 で呼び出しプログラムへ戻ります。

## E35 ユーザー出口: レコードの変更

COBOL で E35 ユーザー出口を作成する場合は、396 ページの『COBOL ユーザー出口ルーチン』および 407 ページの『COBOL E35 ユーザー出口: レコードの変更』を参照してください。

EXITCK オプションは、DFSORT がユーザー出口 E35 からの特定の戻りコードを解釈する方法に影響を与えます。ここでは、あいまいにならないように、ユーザーのシステムで IBM のデフォルトである EXITCK=STRONG が選択されているものと仮定します。EXITCK=STRONG および EXITCK=WEAK を使用する様々な状況での E35 戻りコードの意味の詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

DFSORT は、出力域にレコードを入れる準備が整うたびに、E35 ユーザー出口ルーチンに入ります。

論理の流れの詳細については、365 ページの図 17 を参照してください。

E35 ユーザー出口の用途は次のとおりです。

- 出力データ・セットのレコードを追加する
- 出力データ・セットのレコードを省略する
- 出力データ・セットのレコードを変更する

注:

1. E35 ユーザー出口が可変長レコードを処理する場合は、変更または挿入する各レコードの先頭に 4 バイトの RDW を入れてから、レコードを DFSORT に戻してください。RDW の形式については、*z/OS DFSMS: Using Data Sets* または *System Programming Reference* を参照してください。(あるいは、レコードを最大長まで埋め込み処理して、固定長として処理することもできます。)
2. DFSORT は、指定された値かそのデフォルトを RECORD ステートメントの L3 に使用して、E35 ユーザー出口が DFSORT に戻すレコードの長さを判別しま

## アセンブラー・ユーザー出力ルーチン (出力フェーズのユーザー出口)

す。固定長レコードの場合、E35 ユーザー出口が変更または挿入する各レコードの長さが、必ず L3 の指定値またはデフォルトと一致するようにしてください。可変長レコードの場合、E35 ユーザー出口が変更または挿入する各レコードの RDW が、必ず L3 の指定値またはデフォルト以下になるようにしてください。E35 ユーザー出口が間違っただけのレコード長を DFSORT に渡すと、望ましくない短縮またはアベンドが起きることがあります。

L3 値の詳細については、340 ページの『RECORD 制御ステートメント』を参照してください。

- E35 ユーザー出口を使用してすべての出力レコードを処理する場合は、SORTOUT DD ステートメントを省略できます。
- プログラムから DFSORT を呼び出し、E35 ユーザー出口のアドレスをパラメーター・リストに渡す場合、
  - DFSORT は SORTOUT データ・セットを無視します (ただし、OUTFIL データ・セットを除きます)。
  - MODS ステートメントに E35 を指定しておくこと、DFSORT は終了します。
- SORTOUT DD ステートメントを省略した場合、または SORTOUT DD ステートメントが無視された場合に、OUTFIL データ・セットを指定しないとき、E35 ユーザー出力ルーチンは、各出力レコードを処理して、戻りコード 4 で DFSORT に戻ります。ユーザーが最後のレコードを処理した後で DFSORT がユーザー・ルーチンに戻った場合は、戻りコード 8 で『戻り不可』を示して、DFSORT へ戻ります。
- 入力レコードが VSAM データ・セットからの可変長レコードであれば、4 バイトの RDW が先頭に付けられていることに注意してください。
- レコードを出力域へ入れてしまった後では、その長さを増やすことはできません。
- 組み合わせアプリケーションの場合、E35 ユーザー出力ルーチンの中で削除されたレコードのシーケンス検査は行われません。出力データ・セットがなく E35 ユーザー出力ルーチンを使用する場合は、シーケンス検査は行われません。この場合、レコードが正しい順序になっていることを確認する必要があります。

### E35 ユーザー出口で DFSORT がユーザー・ルーチンへ渡す情報

DFSORT がレコード (最初のレコードを含む) を出力域に入れる準備が整うたびに、E35 ユーザー出力ルーチンに入ります。DFSORT はユーザー・ルーチンに次の 3 つのワードを渡します。

- DFSORT を離れるレコードのアドレス。**これは、通常、出力域のレコードに続きます。E35 ユーザー出口に渡すレコードがなくなると、入力の終わりになります。DFSORT はこのアドレスをゼロにセットして入力の終わりを示してから、E35 ユーザー出口に入ります。

入力の終わりになった後でも、戻りコード 8 が戻されるまでは、DFSORT はユーザー出力ルーチンへ入り続けます。

E35 ユーザー出口は、DFSORT を離れるレコードのアドレスを変更できません。

- 出力域のレコードのアドレス。**この時点では、出力域に何もレコードが入っていないため、最初にユーザー・ルーチンに入る時はゼロです。ユーザーが DFSORT へ戻りコード 4 (レコードの削除) を渡すと、このアドレスはゼロのままです。

## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

**注:** 指し示されたレコードが可変長の場合は、VSAM データ・セットへの出力であっても、この場所に RDW をもっています。

- **ユーザー出口アドレス定数。**これは、E15 または E32 ユーザー出口、あるいは呼び出しプログラムのパラメーター・リストでセットされたとおりに、ユーザー出口へ渡されます。

**注:** ユーザー出口アドレス定数を、従来の組み合わせアプリケーションまたはテープ作業データ・セット分類のアプリケーションに使用できません。

DFSORT は、2 つのレコード・アドレスとユーザー・アドレス定数を含むパラメーター・リストのアドレスを、汎用レジスター 1 に入れます。このリストは 3 フルワードの長さで、フルワード境界から始まります。パラメーター・リストの形式は次のとおりです。

表 52. E35 ユーザー出口パラメーター・リスト

1 ~ 4 バイト	DFSORT を離れるレコードのアドレス
5 ~ 8 バイト	出力域のレコードのアドレス
9 ~ 12 バイト	ユーザー出口アドレス定数

### E35 の戻りコード

E35 ユーザー・ルーチンは、戻りコードを DFSORT へ渡す必要があります。E35 ユーザー出口の戻りコードは次のとおりです。

戻りコード	説明
00 (X'00')	処置をとらない / レコードを変更する
04 (X'04')	レコードを削除する
08 (X'08')	戻らない
12 (X'0C')	レコードを挿入する
16 (X'10')	DFSORT を終了する

#### 0: 処置をとらない

DFSORT に未変更のレコードをそのまま保持させる場合は、DFSORT を離れるレコードのアドレスを汎用レジスター 1 にロードし、戻りコード 0 (ゼロ) で DFSORT へ戻ります。

#### 0: レコードを変更する

レコードが出力データ・セットに入れられる前に変更したい場合は、そのレコードを作業域に移して変更を加え、変更されたレコードのアドレスを汎用レジスター 1 にロードし、戻りコード 0 (ゼロ) で DFSORT へ戻ります。

#### 4: レコードを削除する

ユーザー・ルーチンは、戻りコード 4 で DFSORT に戻るにより、DFSORT を離れるレコードを、削除できます。汎用レジスター 1 にアドレスを入れる必要はありません。

#### 8: 戻らない

DFSORT は、戻りコード 8 を渡されるまで、ユーザー・ルーチンへ戻り続けます。その後は、DFSORT アプリケーションの処理中に、そのユーザー出口が再び使用されることはありません。戻りコード 8 で戻る場合は、汎用レジスター 1 にアドレスを入れる必要はありません。データ・セットの終わりの後でレコードを挿入しない限り、DFSORT がデータ・セットの終わりを示した際に、戻り

## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

コード 8 を渡す必要があります。これは、DFSORT を離れるレコードのアドレスとして、ゼロを渡すことにより行われます。

出力データ・セットがなく、EOF の前に戻りコード 8 で戻るのが通常である場合は、OPTION 制御ステートメントに NOCHECK を指定することで (インストール時にあらかじめ CHECK=NO を指定していない場合)、ICE025A メッセージが出されるのを防ぎます。

処理する入力レコードがまだ残っている段階でユーザー出口ルーチンが戻りコード 8 を DFSORT に渡すと、残りのレコードは DFSORT により処理されますが、ユーザー出口には 渡されません。

### 12: レコードを挿入する

DFSORT を離れるレコードの前に出力レコードを追加するには、新しいレコードのアドレスを汎用レジスター 1 に入れ、戻りコード 12 で DFSORT に戻ります。DFSORT は、DFSORT を離れるレコードのために行った前回のユーザー出口の呼び出しで渡したものと同一アドレスでユーザー・ルーチンへ戻ります。DFSORT は、挿入されたレコードのアドレスを出力域に入れます。その箇所にさらに挿入を行ったり、あるいは DFSORT を離れるレコードを削除できます。

DASD 作業データ・セット分類の場合、DFSORT はシーケンス検査を行いません。テープ作業データ・セット分類の場合、DFSORT を離れるレコードを削除し、それに代わるレコードを挿入しない限り、DFSORT は挿入するレコードのシーケンス検査を行いません。DFSORT は、戻りコード 8 を渡されるまで、ユーザー・ルーチンへ戻り続けます。

### 16: DFSORT を終了する

DFSORT を終了したい場合は、コード 16 で戻ります。すると DFSORT は、戻りコード 16 で呼び出しプログラムまたはシステムへ戻ります。

各種の状況での戻りコードの意味の詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

**E35 ユーザー出口でのレコードの合計:** SUM 制御ステートメントを使用して、レコードを合計できます。しかし、出力域のレコードを変更し、さらに必要であれば、DFSORT を離れるレコードを削除することにより、出力のレコードを合計できます。DFSORT は、DFSORT を離れる新しいレコードのアドレスでユーザー・ルーチンへ戻ります。同じレコードは出力域に残るので、集計を続けることができます。DFSORT を離れるレコードを削除しないと、そのレコードは出力域に追加され、そのアドレスが、出力域内の前のレコードのアドレスに置き換わります。DFSORT は、DFSORT を離れる新しいレコードのアドレスで戻ります。

### E35 ユーザー出口のための記憶域の使用法

DFSORT は、E35 ユーザー出口ルーチンに渡すパラメーター・リストとレコードのために、(GETMAIN または STORAGE OBTAIN を使用して) 記憶域を確保します。DFSORT が確保した記憶域を変更または解放しないでください。

更新したレコードを DFSORT に渡すときなどのために、E35 ユーザー出口ルーチンが使用する記憶域を確保する必要がある場合は、次の方法が使用できます。

1. 出口ルーチンが初めて呼び出されたときに、必要な記憶域を確保します。
2. 出口ルーチンが呼び出されるたびに、確保した記憶域を使用します。
3. 戻りコード 8 を DFSORT に返す前に、記憶域を開放します。

## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

注: 記憶域を確保したら、そのアドレスをユーザー出口アドレス定数に保管しておく、それ以降ユーザー出口が呼び出されるたびに復元できます。

## E37 E37 ユーザー出口: データ・セットのクローズ

出力フェーズの終わりで、E37 ユーザー出口ルーチンに 1 回だけ入ります。このルーチンを使用して、そのフェーズの別のルーチンが使用したデータ・セットをクローズしたり、ユーザー・ルーチンのハウスキーピング機能を実行できます。

注: リンケージ・エディターの特長要件を避けるため (371 ページの『ユーザー出口ルーチンに関する規則』を参照)、これらの機能を、独立した E37 ルーチンではなく、E35 ユーザー出口に組み込むことができます。

## E38 ユーザー出口: 入力データ・セットの処理

このルーチンは、E18 と同じです。ブロック・セットまたはピアレッジ / ベール手法が選択されている場合は、入出力エラー条件を E38 ユーザー出口で処理できません。

### VSAM による E38 ユーザー出口の使用

このユーザー出口を組み合わせたまたはコピー処理の間に使用して、VSAM パスワードを VSAM 入力 ACB に挿入し、各種の VSAM ユーザー出口機能を実行できます。次の例は、DFSORT に制御を返すためにユーザー・プログラムが使用できるコーディングを示しています。

```
ENTRY   E38
      .
      .
E38     LA      1,PARMLST
      RETURN
      CNOP    0,4
PARMLST DS      0H
      DC     X'05'
      DC     AL3(VSAMEXL)
      DC     X'06'
      DC     AL3(PWDLST)
      DC     A(0)
      .
      .
VSAMEXL EXLST  SYNAD=USYNAD,LERAD=ULERAD
PWDLST  DC     H'2'
      DC     CL8'SORTIN01'   SORTIN01 DDNAME
      DC     CL8'INPASS1'   SORTIN01 PASSWORD
      DC     CL8'SORTIN02'   SORTIN02 DDNAME
      DC     CL8'INPASS2'   SORTIN02 PASSWORD
USYNAD  ...
ULERAD  ...
```

図 19. E38 ユーザー出口の例

## E39 ユーザー出口: 出力データ・セットの処理

E39 ユーザー出口ルーチンへは、SORTOUT データ・セットの場合に入りますが、OUTFIL データ・セットの場合には入りません。

## アセンブラー・ユーザー出口ルーチン (出力フェーズのユーザー出口)

### QSAM/BSAM による E39 ユーザー出口の使用

この手法は QSAM/BSAM の E19 の場合と同じです。詳細については、383 ページの『E19 ユーザー出口: 作業データ・セットへの出力の処理』を参照してください。

### VSAM による E39 ユーザー出口の使用

VSAM の場合は、このユーザー出口を使用して VSAM パスワードを VSAM SORTOUT ACB に挿入し、各種の VSAM ユーザー出口機能を実行できます。次の例は、DFSORT へ制御を戻すためにユーザー・プログラムが使用できるコーディングを示しています。

```
          ENTRY  E39
          .
          .
E39      LA      1,PARMLST
          RETURN
          CNOP   0,4
PARMLST DS      0H
          DC     X'05'
          DC     AL3(VSAMEXL)
          DC     X'06'
          DC     AL3(PWDLST)
          DC     A(0)
          .
          .
VSAMEXL EXLST  SYNAD=USYNAD, LERAD=ULERAD
PWDLST  DC     H'1'
          DC     CL8'SORTOUT'      SORTOUT DDNAME
          DC     CL8'OUTPASS'      SORTOUT PASSWORD
USYNAD  ...
ULERAD  ...                          VSAM SYNCH ERROR RTN
                                          VSAM LOGIC ERROR RTN
```

図 20. E39 ユーザー出口の例

---

## アセンブラーで作成されたサンプル・ルーチン

ここでは、アセンブラーで作成されたプログラム・ユーザー出口の例をいくつか示します。

### E15 ユーザー出口: レコード長の変更

このルーチンは可変長入力レコードを変更して、すべてのレコードを同じ長さにします。

## アセンブラーで作成されたサンプル・ルーチン

```

E15      CSECT
* IF A RECORD IS GREATER THAN 204 BYTES, TRUNCATE IT TO 204 BYTES.
* IF A RECORD IS LESS THAN 204 BYTES, PAD IT OUT TO 204 BYTES.
* ALL OF THE RESULTING RECORDS WILL BE 204 BYTES LONG
* (4 BYTES FOR THE RDW AND 200 BYTES OF DATA).
        USING E15,12          SHOW BASE REG
        STM 14,12,12(13)      SAVE ALL REGS EXCEPT 13
        LA 12,0(0,15)         SET BASE REG
        ST 13,SAVE15+4        SAVE BACKWARD POINTER
        LA 14,SAVE15          SET FORWARD POINTER
        ST 14,8(13)           IN SAVE AREA
        LR 13,14              SET OUR SAVE AREA
        LR 2,1                SAVE PARM LIST POINTER
        L 3,0(,2)             LOAD ADDR OF RECORD
        LTR 3,3               EOF
        BZ EOF                YES - DO NOT RETURN
        LH 4,0(,3)            GET RDW
        CH 4,CON204           IS RDW EQ 204
        BE ACCEPT             YES-ACCEPT IT
        BL PAD                LESS THAN 204-PAD
        LH 4,CON204           LIMIT LENGTH TO 204
        B TRUNC               MORE THAN 204-TRUNCATE
PAD      DS 0H                PAD OR TRUNCATE
        MVI DATA,X'00'       ZERO OUT THE BUFFER
        MVC DATA+1(199),DATA
        BCTR 4,0              DECREASE RDW FOR EXECUTE
TRUNC    DS 0H                PAD OR TRUNCATE
        EX 4,MVPAD            MOVE RECORD INTO PAD/TRUNCATE BUFFER
        MVC NEWRDW(2),CON204 SET NEW RDW TO 204
        LA 3,BUFFER           POINT TO PADDED/TRUNCATED RECORD
ACCEPT   DS 0H
        SR 15,15              SET RC=0
        LR 1,3                SET RECORD POINTER
        B GOBACK
EOF      LA 15,8              EOF - SET RC=8
GOBACK   L 13,4(,13)
        L 14,12(,13)
        LM 2,12,28(13)        RESTORE REGS
        BR 14                 RETURN
MVPAD    MVC BUFFER(*-*),0(3) FOR EXECUTE
SAVE15   DS 18F
CON204   DC H'204'
BUFFER   DS 0H
NEWRDW   DS H                NEW RDW OF 204
        DC H'0'
DATA     DC XL200'00'         BUFFER FOR PADDING/TRUNCATING
        END

```

図 21. E15 ユーザー出口の例

## E16 ユーザー出口: NMAX 超過時の現行レコードの分類

メッセージ “NMAX EXCEEDED” が出力されると、このルーチンは、すでに読み込まれたレコードだけを分類するように DFSORT に指示します。



```

E16  CSECT
      LA    15,0    SET RETURN CODE
      BR    14
      END
    
```

図 22. E16 ユーザー出口の例

## E35 ユーザー出口: レコード長の変更

このルーチンは、可変長出力レコードを変更して、すべて同じ長さにします。

```

E35  CSECT
* IF A RECORD IS GREATER THAN 204 BYTES, TRUNCATE IT TO 204 BYTES.
* IF A RECORD IS LESS THAN 204 BYTES, PAD IT OUT TO 204 BYTES.
* ALL OF THE RESULTING RECORDS WILL BE 204 BYTES LONG
* (4 BYTES FOR THE RDW AND 200 BYTES OF DATA).
      USING E35,12          SHOW BASE REG
      STM  14,12,12(13)    SAVE ALL REGS EXCEPT 13
      LA   12,0(0,15)      SET BASE REG
      ST   13,SAVE15+4     SAVE BACKWARD POINTER
      LA   14,SAVE15       SET FORWARD POINTER
      ST   14,8(13)       IN SAVE AREA
      LR   13,14          SET OUR SAVE AREA
      LR   2,1            SAVE PARM LIST POINTER
      L    3,0(,2)        LOAD ADDR OF RECORD
      LTR  3,3            EOF
      BZ   EOF            YES - DO NOT RETURN
      LH   4,0(,3)        GET RDW
      CH   4,CON204       IS RDW EQ 204
      BE   ACCEPT         YES-ACCEPT IT
      BL   PAD            LESS THAN 204-PAD
      LH   4,CON204       LIMIT LENGTH TO 204
      B    TRUNC          MORE THAN 204-TRUNCATE
PAD    DS   0H            PAD OR TRUNCATE
      MVI  DATA,X'00'    ZERO OUT THE BUFFER
      MVC  DATA+1(199),DATA
      BCTR 4,0            DECREASE RDW FOR EXECUTE
TRUNC  DS   0H            PAD OR TRUNCATE
      EX  4,MVPAD         MOVE RECORD INTO PAD/TRUNCATE BUFFER
      MVC NEWRDW(2),CON204 SET NEW RDW TO 204
      LA  3,BUFFER        POINT TO PADDED/TRUNCATED RECORD
ACCEPT DS   0H
      SR  15,15          SET RC=0
      LR  1,3            SET RECORD POINTER
      B   GOBACK
EOF    LA  15,8          EOF - SET RC=8
GOBACK L   13,4(,13)
      L   14,12(,13)
      LM  2,12,28(13)    RESTORE REGS
      BR  14            RETURN
MVPAD  MVC  BUFFER(*-*),0(3) FOR EXECUTE
SAVE15 DS   18F
CON204 DC   H'204'
BUFFER DS   0H
NEWRDW DS   H          NEW RDW OF 204
      DC   H'0'
DATA   DC   XL200'00'    BUFFER FOR PADDING/TRUNCATING
      END
    
```

図 23. E35 ユーザー出口の例

## E61 ユーザー出口: 制御フィールドの変更

このルーチンを使用すれば、このルーチンに渡される 2 進数制御フィールド (つまり、'E' が指定されている) の順序を、昇順から降順へ変更できます。

```
* E61 PARAMETER LIST DSECT
PARML  DSECT
        DS    3C
PARMNUM DS    C  CONTROL FIELD NUMBER
PARMPTR DS    A  ADDRESS OF CONTROL FIELD
        DS    2C
PARMLEN DS    H  CONTROL FIELD LENGTH
*
E61REV  CSECT
* CHANGE THE ORDER OF EACH CONTROL FIELD PASSED TO THIS ROUTINE
* FROM ASCENDING TO DESCENDING BY REVERSING ALL OF THE BITS.
* ASSUMES THAT ONLY BI CONTROL FIELDS ARE PASSED.
        USING E61REV,12          SHOW BASE REG
STM     14,12,12(13)          SAVE ALL REGS EXCEPT R13
LA      12,0(0,15)            SET BASE REG
ST      13,SAVE61+4          SAVE BACKWARD POINTER
LA      14,SAVE61            SET FORWARD POINTER
ST      14,8(13)             IN SAVE AREA
LR      13,14                SET OUR SAVE AREA
LR      3,1                  SET PARM LIST POINTER
        USING PARML,3
L        4,PARMPTR            GET POINTER TO CONTROL FIELD IMAGE
LH      5,PARMLEN            GET LENGTH OF CONTROL FIELD
BCTR   5,0                    SUBTRACT 1 FOR EXECUTE
EX      5,REVCF              CHANGE FROM ASCENDING TO DESCENDING
GOBACK  L        13,4(,13)
        LM      14,12,12(13)    RESTORE REGS
        BR      14              RETURN
REVCF  XC      0(*-*,4),REVFF  REVERSE CONTROL FIELD BITS
SAVE61 DS      18F
REVFF  DC      256X'FF'
        LTORG
        END
```

図 24. E61 ユーザー出口の例

---

## COBOL ユーザー出口ルーチン

COBOL で作成された E15 および E35 ユーザー出口ルーチンを用いて、アセンブラーで作成された E15 および E35 ユーザー出口ルーチンと同じタスクを実行できます。ただし、COBOL ルーチンとアセンブラー・ルーチンでは、それぞれのルーチンと DFSORT の間で情報を渡す方法が異なります。

- COBOL ルーチンは、DATA DIVISION の LINKAGE SECTION に記述したフィールドを介して情報を渡す必要があります。アセンブラーは汎用レジスター 1 および、パラメーター・リスト内のポインターを使用します。
- COBOL は、COBOL の特殊レジスターである RETURN-CODE を使用します。アセンブラーは戻りコード用にレジスター 15 を使用します。
- COBOL は、戻りコード 20 を使用して、レコードの変更または置換を行います。アセンブラーは戻りコード 0 を使用します。

- COBOL ルーチンは、E15/E35 連絡にユーザー出口域を使用できます。アセンブラーはユーザー・アドレス定数を使用します。

## COBOL ユーザー出口に関する要件

COBOL ユーザー出口には次の規則が適用されます。これらの COBOL ユーザー出口規則に従っていないと、終了してしまったり、予期しない結果をもたらしたりします。

**注:** 「VS COBOL II またはそれ以降」とは、VS COBOL II、COBOL for MVS & VM、COBOL for OS/390 & VM、COBOL for z/OS and OS/390、および言語環境プログラムを意味します。

- COBOL で作成されるユーザー出口は、STOP RUN ステートメントを使用できません。DFSORT へ戻るには、GOBACK ステートメントを使用してください。
- VS COBOL II ユーザー出口は、RES/RENT コンパイラー・オプションを指定してコンパイルします。
- OS/VS COBOL ユーザー出口のコンパイルに RES コンパイラー・オプションを指定すると、VS COBOL II またはそれ以降への移行を容易にします。ただし、NORES でコンパイルされたユーザー出口は DFSORT の下で実行されます。
- READY TRACE、EXHIBIT、または DISPLAY ステートメントを含んでいるユーザー出口の場合、通常 SYSOUT へ書き込まれる DFSORT メッセージは、MSGDDN パラメーターを使用して別のデータ・セットへ送ります。READY TRACE、EXHIBIT、および DISPLAY ステートメントの場合は、COBOL は SYSOUT へも書き込みます。したがって、SYSOUT へのメッセージが、出力のインターリーピングのために、消失することがあります。

あるいは、SYSx コンパイラー・オプション (OS/VS COBOL の場合)、または OUTDD コンパイラー・オプション (VS COBOL II、COBOL for MVS & VM、COBOL for OS/390 & VM、COBOL for z/OS and OS/390 の場合) を使用して、COBOL 出力を別のデータ・セットへ書き込む方法もあります。

**注:** READY TRACE および EXHIBIT は OS/VS COBOL でだけサポートされません。

- COBOL ユーザー出口には、SORT または MERGE 動詞が含まれていてはなりません。
- MODS 制御ステートメントをコーディングして、COBOL ユーザー出口を記述する場合は、4 番目のパラメーターに C を使用してください。この C は、正しいパラメーター・リストを作成するように DFSORT に指示します。
- DFSORT を VS COBOL II またはそれ以降のプログラムから呼び出す場合は、VS COBOL II またはそれ以降の FASTSRT オプションが、入力に対して有効であれば COBOL E15 を、また、FASTSRT が出力に対して有効であれば COBOL E35 を使用できます。COBOL ユーザー出口は、VS COBOL II またはそれ以降でコンパイルしてください。
- VS COBOL II でコンパイルしたユーザー出口を実行する場合は、VS COBOL II ライブラリーか言語環境プログラム・ライブラリーを使用する必要があります。COBEXIT=COB2 がユーザーのインストール・システムのデフォルトでない場合は、必ず COB2 パラメーターを OPTION 制御ステートメントに指定してください。指定しないと、パフォーマンスが低下します。

## COBOL ユーザー出口ルーチン

- COBOL for MVS & VM、COBOL for OS/390 & VM または COBOL for z/OS and OS/390 でコンパイルしたユーザー出口を実行する場合は、言語環境プログラム・ライブラリーを使用する必要があります。COBEXIT=COB2 がユーザーのインストール・システムのデフォルトでない場合は、必ず COB2 パラメーターを OPTION 制御ステートメントに指定してください。指定しないと、パフォーマンスが低下します。
- この実行で COBEXIT=COB2 が有効な場合は、たとえ COBOL ユーザー出口が OS/VS COBOL コンパイラーでコンパイルされたものであっても、VS COBOL II ライブラリーまたは言語環境プログラム・ライブラリーを使用してください。
- OS/VS COBOL コンパイラーまたは VS COBOL II コンパイラーのいずれかでコンパイルされたユーザー出口を実行し、かつ RES オプションを指定した場合は、実行時に COBOL ライブラリー・ルーチンが使用可能である必要があります。COBOL ライブラリーは、OS/VS COBOL の NORES オプションを用いてコンパイルされたユーザー出口で必要になる場合があります。COBOL ライブラリーを必要とするオプションについての詳細は、OS/VS COBOL 解説書を参照してください。
- OS/VS COBOL でコンパイルされたユーザー出口は、OS/VS COBOL または VS COBOL II ライブラリーのいずれかを用いて実行したり、場合によりはライブラリーを使用しないでも実行できます。
- COBOL for MVS & VM、COBOL for OS/390 & VM、および COBOL for z/OS and OS/390 は、実行時に、言語環境プログラムの COBOL ライブラリー・ルーチンを必要とします。
- OS/VS COBOL でコンパイルされ、VS COBOL II ライブラリーを用いて実行するユーザー出口は、DFSORT NOESTAE オプションが無効な場合は STAE を発行できません。(STAE を発行する OS/VS COBOL コンパイラー・オプションは、STATE、FLOW、SYMDMP、COUNT、および TRACE です。)

### COBOL に関する要件 (コピー処理の場合)

コピー処理の場合、以下の制約を除くすべての分類処理要件が適用されます。

- DFSORT を直接呼び出し、かつ COBEXIT=COB2 が有効な場合は、別個にコンパイルされた COBOL E15 ユーザー出口、または別個にコンパイルされた COBOL E35 ユーザー出口の いずれか を使用できます。
- VS COBOL II またはそれ以降のプログラムから DFSORT を呼び出す場合は、FASTSRT が何に対して有効であるかにより、次のような制限があります。
  - 入力に対してのみ有効な場合: 別個にコンパイルされた E15 ユーザー出口が使用できますが、別個にコンパイルされた E35 ユーザー出口は使用できません。
  - 出力に対してのみ有効な場合: 別個にコンパイルされた E35 ユーザー出口が使用できますが、別個にコンパイルされた E15 ユーザー出口は使用できません。
  - 入力および出力に対して有効な場合: 別個にコンパイルされた E15 または別個にコンパイルされた E35 の いずれか が使用できますが、両方は使用できません (COBEXIT=COB2 の場合)。

別個にコンパイルされた E15 および E35 ユーザー出口が同時に検出された場合は、DFSORT コピー処理は終了します。メッセージ ICE161A が出力されます。

## COBOL 記憶域の所要量

RES コンパイラー・オプションを指定してコンパイルされた COBOL ユーザー出口を実行する場合は、COBOL ライブラリー・サブルーチンに使用できる十分な記憶域が確保されていることを確認してください。(ただし、ライブラリーが常駐ライブラリーとしてインストールされている場合は、これは当てはまりません。)

DFSORT の主記憶域の最小必要量のほかに、OS/VS COBOL ライブラリー・サブルーチン用にユーザーの REGION 内にさらに 40 キロバイトの記憶域、および VS COBOL II ライブラリー・サブルーチン用に 150 キロバイト記憶域が必要になります。ほとんどの VS COBOL II ライブラリー・サブルーチンは、16 メガバイト仮想記憶域より上に常駐させることができます。ただし、実際に VS COBOL II ライブラリー・サブルーチンを 16 メガバイト境界よりも上にロードできるかどうかは、それらのサブルーチンがどのようにインストールされたかにより決まります。MVS & VM の言語環境プログラムを実行するには、1200 キロバイトが必要です。COBPACKS を 16 メガバイト境界より上にロードすることにより、16 メガバイト境界より下の MVS & VM の言語環境プログラムの記憶域必要量を最小化できます。詳細については、言語環境プログラム MVS および VM 版 導入およびカスタマイズ、SC88-7014、または z/OS 言語環境プログラム カスタマイズ を参照してください。

特定の条件下では、16 メガバイト境界より下のユーザー REGION 内の全記憶域を DFSORT が使用できるため、ユーザー出口の実行中に必要な COBOL ライブラリー・サブルーチンをロードするスペースが残されていないことがあります。

TMAXLIM または SIZE/MAINSIZE オプションで極端に高い値 (たとえば、16 メガバイト境界より上の主記憶域に関するユーザー・システム限界値) を指定していなければ、16 メガバイトより上の主記憶域を使用できます。この場合、ARESALL または ARESINV オプションを使用して記憶域を解放できます。

処理中に、COBOL ライブラリー・サブルーチンに必要な記憶域の実際の大きさは、COBOL ユーザー出口で実行される機能により異なります。OS/VS COBOL ライブラリー・サブルーチンを用いて実行する場合は、最低 40 キロバイトを、また VS COBOL II ライブラリーまたは MVS & VM 言語環境プログラムを用いて実行する場合は、20 キロバイトをユーザー出口のサイズに追加してください。ユーザー出口が入出力を行う場合は、入出力バッファー用に追加記憶域を予約する必要があります。バッファー用の追加記憶域は、MODS ステートメントの *m* パラメーターで指定します。VS COBOL II またはそれ以降のユーザー出口は、類似の OS/VS COBOL ユーザー出口ほど記憶域を必要としません。ユーザー出口が呼び出される前に、DFSORT が一部の COBOL ライブラリー・サブルーチンのための記憶域を自動的に解放するためです。

SIZE/MAINSIZE=MAX が有効な場合は、記憶域を解放する代替方法として、RESALL または RESINV オプションを使用できます。

**注:** 次の場合、さらに 70 キロバイトの記憶域を解放することが必要になることがあります。

- E15 と E35 の両方のユーザー出口を呼び出す場合
- 非常駐の VS COBOL II ライブラリー・サブルーチンを使用して実行する場合
- 16 メガバイト境界より上にある DFSORT を使用して分類を実行する場合

## COBOL ユーザー出力ルーチン

これは、次のいずれかに 70 キロバイトを加えることにより行うことができます。

- E35 ユーザー出力の場合は、MODS ステートメントの m パラメーター (m = E35 ユーザー出力サイズ + 20 キロバイト + 70 キロバイト)
- SIZE/MAINSIZE=MAX が有効な場合は、RESALL オプション

---

## COBOL ユーザー出力ルーチン (入力フェーズのユーザー出力)

### COBOL E15 ユーザー出力: レコードの引き渡しまたは変更

EXITCK オプションは、DFSORT がユーザー出力 E15 からの特定の戻りコードを解釈する方法に影響を与えます。ここでは、あいまいにならないように、ユーザーのシステムで IBM のデフォルトである EXITCK=STRONG が選択されているものと仮定します。EXITCK=STRONG および EXITCK=WEAK を使用する様々な状況での E15 戻りコードの詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

DFSORT は、新しいレコードが入力フェーズに呼び込まれるたびに、E15 ユーザー出力ルーチンに入ります。DFSORT は、ユーザー出力が戻りコード 8 を使用して (戻り不要) を DFSORT に知らせるまで、E15 の呼び出しを続けます (入力レコードがない場合でも)。

論理の流れの詳細については、365 ページの図 17 を参照してください。

E15 ユーザー出力の用途は次のとおりです。

- レコードを入力データ・セットへ追加する。
- 入力データ・セット全体を DFSORT へ引き渡す。
- レコードを入力データ・セットから削除する。
- 入力データ・セット内のレコードを変更する。

注:

1. E15 および E35 ユーザー出力の両方を使用する場合、どちらも同じバージョンの COBOL を使用している必要があります。
2. E15 ユーザー出力を使用して、すべてのレコードを DFSORT に渡す場合は、SORTIN DD ステートメントを省略できます。この場合は、プログラム制御ステートメントに RECORD ステートメントを組み込んでください。
3. SORTIN DD ステートメントを省略すると、すべての入力レコードが COBOL E15 ユーザー出力を介して DFSORT に渡されます。戻りコード 12 で DFSORT へ戻ります。最後のレコードが渡された後に DFSORT が E15 ユーザー出力へ戻ると、レジスター 15 には「戻り不可」を示す戻りコード 8 を入れて、DFSORT へ戻ります。
4. DFSORT は、戻りコード 8 を受け取るまでは、E15 ユーザー出力に再入を続けます。ただし、STOPAFT が有効な場合は、STOPAFT カウントが満たされると (戻りコード 12 が返されても) それ以上のレコードは DFSORT には挿入されません。
5. COBOL E15 ユーザー出力では、ダイナミック・リンク編集を使用できません。

## COBOL との E15 インターフェース

E15 ユーザー出口が呼び出されるたびに、DFSORT は次のフィールドを渡します。

- レコード・フラグ
- 新しいレコード
- 新しいレコードの長さ (可変長レコードの場合)
- ユーザー出口域の長さ
- ユーザー出口域

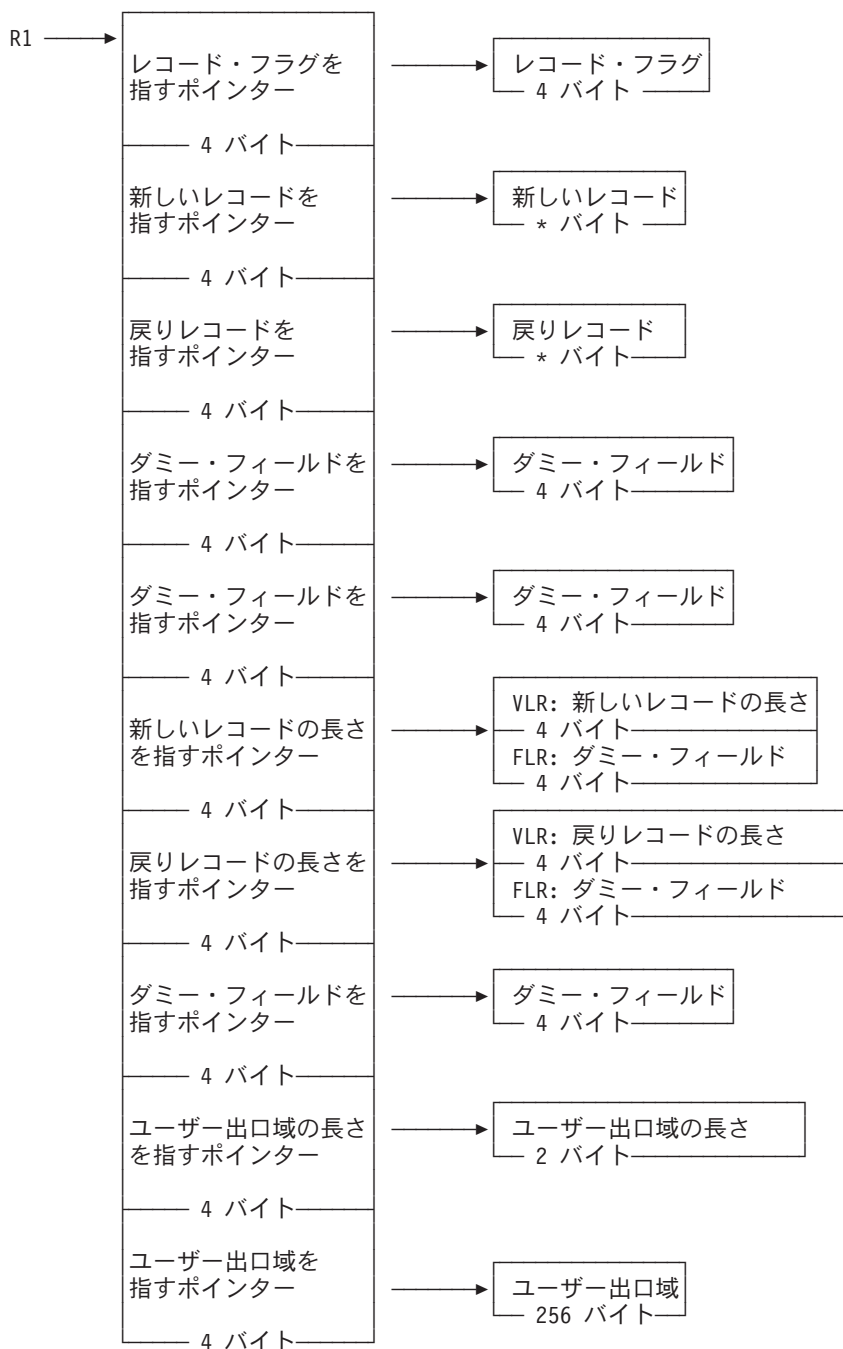
E15 が DFSORT へ戻る際には、E15 ユーザー出口は下記に示したフィールドの一部またはすべてを DFSORT へ渡します。最初のフィールドは必須ですが、その他のフィールドは必要に応じて変更できます。

- RETURN-CODE (COBOL の特殊レジスター RETURN-CODE をセットすることにより、ユーザー出口が割り当てる)
- 戻りレコード
- 戻りレコードの長さ (VLR の場合)
- ユーザー出口域の長さ
- ユーザー出口域

これらのフィールドを COBOL E15 ユーザー出口で使用方法の詳細については、403 ページの『E15 LINKAGE SECTION フィールド (固定長および可変長レコードの場合)』を参照してください。

402 ページの図 25 は、E15 ユーザー出口についての COBOL へのインターフェースを詳しく示したものです。

## COBOL ユーザー出口ルーチン (入力フェーズのユーザー出口)



バイト数

- \* - VLR: バイト数は、対応する長さフィールドによって与えられます。
- FLR: バイト数は LRECL と同じです。

図 25. COBOL との E15 DFSORT インターフェース

**E15 LINKAGE SECTION の例:** 403 ページの図 26 は、論理レコード長 (LRECL) が 100 である固定長レコード (FLR) データ・セットの LINKAGE SECTION のコーディング例です。この例は、ユーザーの COBOL ルーチンへ渡されるフィールドのレイアウトを示しています。



## COBOL ユーザー出口ルーチン (入力フェーズのユーザー出口)

```
LINKAGE SECTION.  
01 RECORD-FLAGS      PIC 9(8) COMPUTATIONAL.  
   88 FIRST-REC      VALUE 00.  
   88 MIDDLE-REC     VALUE 04.  
   88 END-REC        VALUE 08.  
01 NEW-REC           PIC X(100).  
01 RETURN-REC       PIC X(100).  
01 UNUSED1          PIC 9(8) COMPUTATIONAL.  
01 UNUSED2          PIC 9(8) COMPUTATIONAL.  
01 UNUSED3          PIC 9(8) COMPUTATIONAL.  
01 UNUSED4          PIC 9(8) COMPUTATIONAL.  
01 UNUSED5          PIC 9(8) COMPUTATIONAL.  
01 EXITAREA-LEN     PIC 9(4) COMPUTATIONAL.  
01 EXITAREA.  
   05 EAREA OCCURS 1 TO 256 TIMES  
      DEPENDING ON EXITAREA-LEN PIC X.
```

図 26. E15 (固定長レコード) の LINKAGE SECTION のコーディング例

図 27 は、最大 LRECL が 200 である可変長レコード (VLR) データ・セットの LINKAGE SECTION のコーディング例です。この例は、ユーザーの COBOL ルーチンへ渡されるフィールドのレイアウトを示しています。

```
LINKAGE SECTION.  
01 RECORD-FLAGS      PIC 9(8) COMPUTATIONAL.  
   88 FIRST-REC      VALUE 00.  
   88 MIDDLE-REC     VALUE 04.  
   88 END-REC        VALUE 08.  
01 NEW-REC.  
   05 NREC OCCURS 1 TO 200 TIMES  
      DEPENDING ON NEW-REC-LEN PIC X.  
01 RETURN-REC.  
   05 RREC OCCURS 1 TO 200 TIMES  
      DEPENDING ON RETURN-REC-LEN PIC X.  
01 UNUSED1          PIC 9(8) COMPUTATIONAL.  
01 UNUSED2          PIC 9(8) COMPUTATIONAL.  
01 NEW-REC-LEN      PIC 9(8) COMPUTATIONAL.  
01 RETURN-REC-LEN  PIC 9(8) COMPUTATIONAL.  
01 UNUSED3          PIC 9(8) COMPUTATIONAL.  
01 EXITAREA-LEN     PIC 9(4) COMPUTATIONAL.  
01 EXITAREA.  
   05 EAREA OCCURS 1 TO 256 TIMES  
      DEPENDING ON EXITAREA-LEN PIC X.
```

図 27. E15 (可変長レコード) の LINKAGE SECTION のコーディング例

### E15 LINKAGE SECTION フィールド (固定長および可変長レコードの場合)

LINKAGE SECTION のフィールドは、下記に示すように DFSORT およびユーザー・ルーチンが使用します。分かりやすくするために、図 27 のフィールド名を使用しています。

- 渡されたレコードの状況をユーザーの COBOL ルーチンへ知らせるために、DFSORT は、以下に示すように、レコード・フラグ・フィールド (RECORD-FLAGS) を使用します。

#### 0 (FIRST-REC)

新しいレコードは最初に渡されたレコードです。

## COBOL ユーザー出口ルーチン (入力フェーズのユーザー出口)

### 4 (MIDDLE-REC)

新しいレコードは最初に渡されたレコードではありません。

### 8 (END-REC)

すべてのレコードがユーザー・ルーチンへ渡されたか、あるいは渡すレコードがありません。

- DFSORT は、新しいレコード・フィールド (NEW-REC) へ次の入力レコードを入れます。VLR には RDW は含まれていませんが、DFSORT はこの VLR の長さを新しいレコード長のフィールド (NEW-REC-LEN) に入れます。  
NEW-REC-LEN フィールド内の値は、レコードだけの長さで、RDW 用の 4 バイトは含みません。
- ユーザー・ルーチンが挿入 / 置換レコードを戻りレコード・フィールド (RETURN-REC) に入れる場合は、VLR に RDW を含めてはなりません。ユーザー・ルーチンはこのレコードの長さを戻りレコード長さフィールド (RETURN-REC-LEN) に入れます。RETURN-REC-LEN フィールドの値はレコードだけの長さで、RDW 用の 4 バイトを含んではなりません。
- DFSORT が COBOL E15 または COBOL E35 ユーザー出口を呼び出すたびに、DFSORT は 256 バイトのユーザー出口域フィールド (EXITAREA) をユーザー出口へ渡します。ユーザー出口域フィールドが最初に COBOL E15 ユーザー出口に渡されるときには、このフィールドには 256 個のブランクが入っており、ユーザー出口域の長さフィールド (EXITAREA-LEN) には 256 が入っています。  
ユーザー出口域フィールドまたはユーザー出口長さフィールドに対して行なった変更はすべて、ユーザーの COBOL E15 ユーザー出口と COBOL E35 ユーザー出口の両方へ渡されます。

#### 注:

1. ユーザー出口域の長さフィールドを、256 バイトを超える値にセットできません。
2. 入力に使用されるデータが COBOL の実行で作成されなかった場合、ユーザーのデータ・セット用に定義された LRECL について知っておく必要があります。VLR の場合、ユーザーの COBOL ユーザー出口に定義されるレコードの最大長は、LRECL 値よりも 4 バイト短くなります。COBOL がレコードの一部として RDW を含めないためです。(各 VLR は、4 バイトの RDW フィールドで始まります。RDW は、COBOL ユーザー出口に渡されたレコードには含まれていません。)
3. ユーザーがコーディングする必要があるのは、ユーザー・ルーチンが実際に使用する最後のフィールドまで (たとえば、ユーザー出口域を使用しない場合は、RETURN-REC まで) です。
4. DFSORT は、指定された値かそのデフォルトを RECORD ステートメントの L2 に使用して、E15 ユーザー出口が DFSORT に戻すレコードの長さを判別します。固定長レコードの場合、E15 ユーザー出口が変更または挿入する各レコードの長さが、必ず L2 の指定値またはデフォルトと一致するようにしてください。可変長レコードの場合、E15 ユーザー出口が変更または挿入する各レコードの長さが、必ず L2 の指定値またはデフォルト以下になるようにしてください。E15 ユーザー出口が間違ったレコード長を DFSORT に渡すと、望ましくない短縮またはアベンドが起きることがあります。

L2 値の詳細については、340 ページの『RECORD 制御ステートメント』を参照してください。

## E15 の戻りコード

COBOL E15 ユーザー・ルーチンは、COBOL の特殊レジスターである RETURN-CODE フィールドに戻りコードを入れて DFSORT へ渡す必要があります。E15 ユーザー出口の戻りコードは次のとおりです。

戻りコード	説明
00 (X'00')	処置をとらない
04 (X'04')	レコードを削除する
08 (X'08')	戻らない
12 (X'0C')	レコードを挿入する
16 (X'10')	DFSORT を終了する
20 (X'14')	レコードを変更または置き換える

### 0: 処置をとらない

DFSORT にレコードをそのまま保持させたい場合は、RETURN-CODE を 0 にセットして戻ります。

### 4: レコードを削除する

DFSORT にレコードを削除させたい場合は、RETURN-CODE を 4 にセットして戻ります。

### 8: 戻らない

DFSORT は RETURN-CODE を 8 にセットして戻るまで、ユーザー・ルーチンに入り続けます。その後は、DFSORT アプリケーションの処理中にそのユーザー出口が再び使用されることはありません。データ・セットの終わりの後でレコードを挿入しない限り、ユーザーは、DFSORT がデータ・セットの終わりを示すときに、RETURN-CODE に 8 をセットします。DFSORT はレコード・フラグ・フィールドに 8 をセットしてユーザー・ルーチンへ入ることにより、データ・セットの終わりを示します。

処理する入力レコードがまだ残っている段階でユーザー出口ルーチンが戻りコード 8 を DFSORT に渡すと、残りのレコードは DFSORT により処理されますが、ユーザー出口には 渡されません。

### 12: レコードを挿入する

入力データ・セットの新しいレコードの前に、DFSORT にレコードを追加させる場合、次のことを行ないます。

- 挿入レコードを戻りレコード・フィールドへ移動する。
- VLR の場合は、その長さを戻りレコード長さフィールドへ移動する (この長さに 4 バイトの RDW を含めないでください)。
- RETURN-CODE を 12 にセットして戻ります。

DFSORT は、新しいレコード・フィールドに前と同じレコードを入れたままユーザー・ルーチンへ戻るため、ユーザー・ルーチンはさらにレコードを挿入したり、新しいレコードを処理できます。

データ・セットの終わりの後でレコードを挿入することもできます。DFSORT は、ユーザーが RETURN-CODE 12 を渡している限り、ユーザー・ルーチンへ戻り続けます。これは、RETURN-CODE に 8 をセットして戻るまで続けます。

## COBOL ユーザー出口ルーチン (入力フェーズのユーザー出口)

### 16: DFSORT を終了する

DFSORT を終了させたい場合は、RETURN-CODE に 16 をセットして戻ります。すると DFSORT は、戻りコード 16 で呼び出しプログラムまたはシステムへ戻ります。

### 20: レコードを変更する

新しいレコードを変更する場合は、次の処理を行います。

- 新しいレコードを戻りレコード・フィールドへ移動する。
- 戻りレコード・フィールドのレコードを変更する。
- VLR レコードの場合は、その長さを戻りレコード長さフィールドへ移動する。
- RETURN-CODE を 20 にセットして戻る。

注: ユーザー・ルーチンがレコード・サイズを変更する場合は、新しいサイズを RECORD ステートメントに示す必要があります。

### 20: レコードを置き換える

新しいレコードを置き換える場合は、次の処理を行います。

- 置き換えレコードを戻りレコード・フィールドへ移動する。
- VLR レコードの場合は、その長さを戻りレコード長さフィールドへ移動する。(この長さに 4 バイトの RDW を含めないでください。)
- RETURN-CODE を 20 にセットして戻る。

各種の状況での戻りコードの意味の詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

## E15 PROCEDURE DIVISION に関する要件

PROCEDURE DIVISION を コーディングする場合、次の要件を満たす必要があります。

- DFSORT へ制御を戻す場合は、GOBACK ステートメントを使用します。
- PROCEDURE DIVISION のヘッダーの USING オプションで、LINKAGE SECTION に 各 01 レベル名を指定します。ヘッダーの前の 01 レベル名を必ずしもすべて使用するわけではない場合でも、使用を計画している最後の名前まで順に指定する必要があります。

例:

403 ページの図 26 の FLR の例については、次のようにコーディングします。

```
PROCEDURE DIVISION USING RECORD-FLAGS, NEW-REC,  
RETURN-REC, UNUSED1, UNUSED2, UNUSED3,  
UNUSED4, UNUSED5, EXITAREA-LEN, EXITAREA.
```

403 ページの図 27 の VLR の例については、次のようにコーディングします。

```
PROCEDURE DIVISION USING RECORD-FLAGS, NEW-REC,  
RETURN-REC, UNUSED1, UNUSED2,  
NEW-REC-LEN, RETURN-REC-LEN,  
UNUSED3, EXITAREA-LEN, EXITAREA.
```

## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

### COBOL E35 ユーザー出口: レコードの変更

EXITCK オプションは、DFSORT がユーザー出口 E35 からの特定の戻りコードを解釈する方法に影響を与えます。ここでは、あいまいにならないように、ユーザーのシステムで IBM のデフォルトである EXITCK=STRONG が選択されているものと仮定します。EXITCK=STRONG および EXITCK=WEAK を使用する様々な状況での E35 戻りコードの詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

DFSORT は、出力域にレコードを入れる準備が整うたびに、E35 ユーザー出口ルーチンに入ります。

論理の流れの詳細については、365 ページの図 17 を参照してください。

E35 ユーザー出口の用途は次のとおりです。

- 出力データ・セットのレコードを追加する
- 出力データ・セットのレコードを省略する
- 出力データ・セットのレコードを変更する

DFSORT がデータ・セットの終わりを示した (レコード・フラグ・フィールドに 8 がセットされた) ときは、RETURN-CODE に 8 をセットします (データ・セットの終わりの後にレコードを挿入する場合を除きます)。セットしないと、DFSORT は E35 に入り続けます。

注:

1. E15 および E35 ユーザー出口の両方を使用する場合、どちらも同じバージョンの COBOL を使用している必要があります。
2. E35 ユーザー出口を使用してすべての出力レコードを処理する場合は、SORTOUT DD ステートメントを省略できます。
3. SORTOUT DD ステートメントを省略した場合、および OUTFIL データ・セットを指定しない場合、E35 ユーザー出口ルーチンは、各出力レコードを処理して、戻りコード 4 で DFSORT に戻ります。ユーザーが最後のレコードを処理した後で DFSORT がユーザー・ルーチンに戻った場合は、戻りコード 8 で『戻り不可』を示して、DFSORT へ戻ります。
4. COBOL E35 ユーザー出口では、ダイナミック・リンク編集を使用できません。

### COBOL との E35 インターフェース

E35 ユーザー出口が呼び出されるたびに、DFSORT は次のフィールドを渡します。

- レコード・フラグ
- DFSORT を離れるレコード
- DFSORT を離れるレコードの長さ (可変長レコードの場合)
- ユーザー出口域の長さ
- ユーザー出口域

E35 が DFSORT へ戻る際には、E35 ユーザー出口は下記に示されているフィールドの一部またはすべてを DFSORT へ渡します。最初のフィールドは必須ですが、その他のフィールドは必要に応じて変更できます。

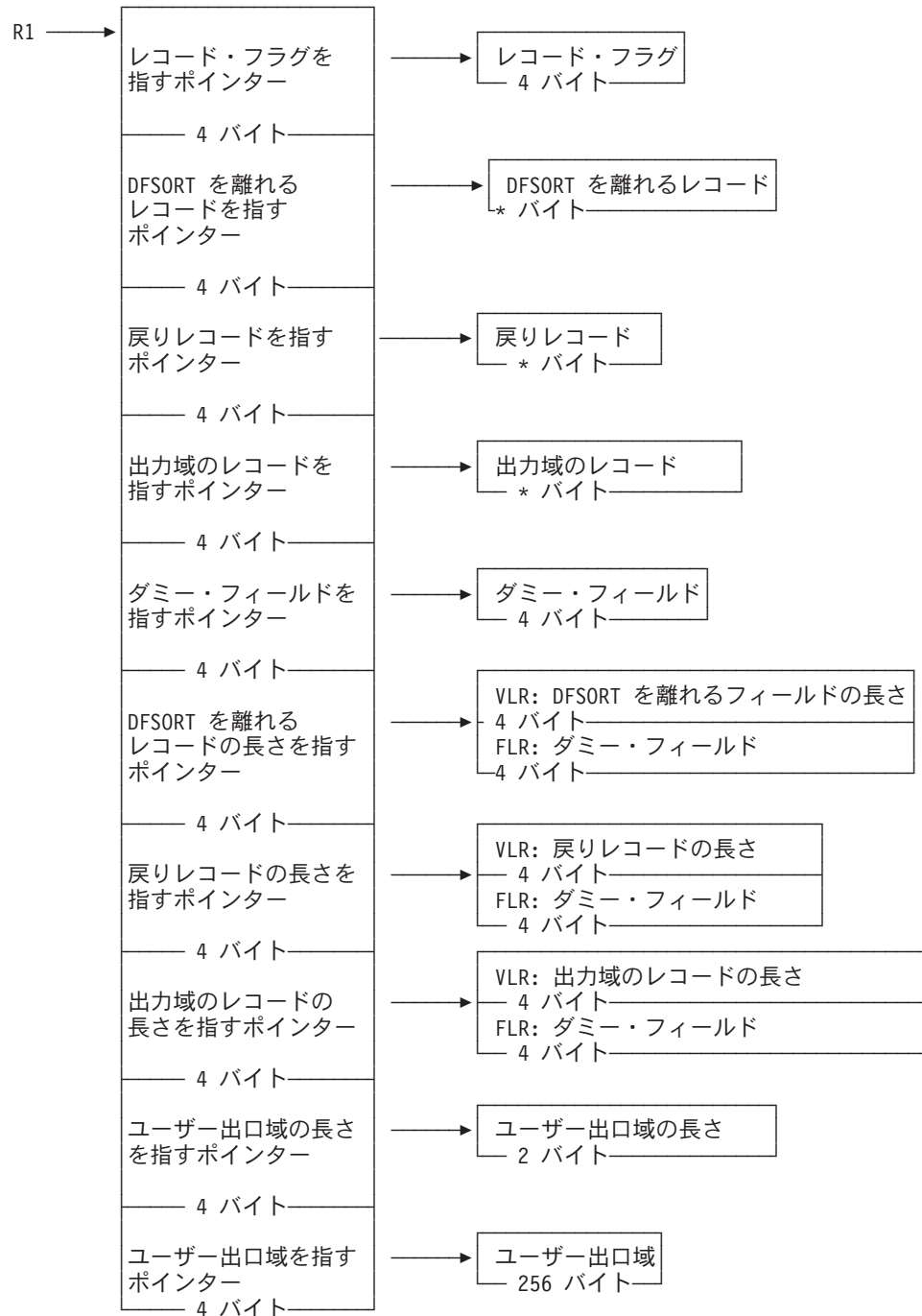
## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

- RETURN-CODE (COBOL の特殊レジスター RETURN-CODE をセットすることにより、ユーザー出口が割り当てる)
- 戻りレコード
- 戻りレコードの長さ (可変長レコードの場合)
- ユーザー出口域の長さ
- ユーザー出口域

これらのフィールドを COBOL E35 ユーザー出口で使用方法の詳細については、410 ページの『E35 LINKAGE SECTION フィールド (固定長および可変長レコードの場合)』を参照してください。

409 ページの図 28 は、E35 ユーザー出口についての COBOL へのインターフェースを詳しく示したものです。

## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)



バイト数

- \* - VLR: バイト数は対応する長さフィールドで与えられます。
- FLR: バイト数は LRECL と同じです。

図 28. COBOL との E35 インターフェース

**E35 LINKAGE SECTION の例:** 図 29 は、論理レコード長 (LRECL) が 100 である固定長レコード (FLR) データ・セットの LINKAGE SECTION のコーディング例です。この例は、ユーザーの COBOL ルーチンへ渡されるフィールドのレイアウトを示しています。

## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

```
LINKAGE SECTION.  
01 RECORD-FLAGS      PIC 9(8) COMPUTATIONAL.  
   88 FIRST-REC      VALUE 00.  
   88 MIDDLE-REC     VALUE 04.  
   88 END-REC        VALUE 08.  
01 LEAVING-REC       PIC X(100).  
01 RETURN-REC        PIC X(100).  
01 OUTPUT-REC        PIC X(100).  
01 UNUSED1           PIC 9(8) COMPUTATIONAL.  
01 UNUSED2           PIC 9(8) COMPUTATIONAL.  
01 UNUSED3           PIC 9(8) COMPUTATIONAL.  
01 UNUSED4           PIC 9(8) COMPUTATIONAL.  
01 EXITAREA-LEN      PIC 9(4) COMPUTATIONAL.  
01 EXITAREA.  
   05 EAREA OCCURS 1 TO 256 TIMES  
      DEPENDING ON EXITAREA-LEN PIC X.
```

図 29. E35 (固定長レコード) の LINKAGE SECTION のコーディング例

図 30 は、最大 LRECL が 200 である可変長レコード (VLR) データ・セットの LINKAGE SECTION のコーディング例です。この例は、ユーザーの COBOL ルーチンへ渡されるフィールドのレイアウトを示しています。

```
LINKAGE SECTION.  
01 RECORD-FLAGS      PIC 9(8) COMPUTATIONAL.  
   88 FIRST-REC      VALUE 00.  
   88 MIDDLE-REC     VALUE 04.  
   88 END-REC        VALUE 08.  
01 LEAVING-REC.  
   05 LREC OCCURS 1 TO 200 TIMES  
      DEPENDING ON LEAVING-REC-LEN PIC X.  
01 RETURN-REC.  
   05 RREC OCCURS 1 TO 200 TIMES  
      DEPENDING ON RETURN-REC-LEN PIC X.  
01 OUTPUT-REC.  
   05 OREC OCCURS 1 TO 200 TIMES  
      DEPENDING ON OUTPUT-REC-LEN PIC X.  
01 UNUSED1           PIC 9(8) COMPUTATIONAL.  
01 LEAVING-REC-LEN   PIC 9(8) COMPUTATIONAL.  
01 RETURN-REC-LEN    PIC 9(8) COMPUTATIONAL.  
01 OUTPUT-REC-LEN    PIC 9(8) COMPUTATIONAL.  
01 EXITAREA-LEN      PIC 9(4) COMPUTATIONAL.  
01 EXITAREA.  
   05 EAREA OCCURS 1 TO 256 TIMES  
      DEPENDING ON EXITAREA-LEN PIC X.
```

図 30. E35 (可変長レコード) の LINKAGE SECTION のコーディング例

### E35 LINKAGE SECTION フィールド (固定長および可変長レコードの場合)

LINKAGE SECTION のフィールドは、下記に示すように DFSORT およびユーザー・ルーチンが使用します。分かりやすくするために、図 30 のフィールド名を使用しています。

- 渡されたレコードの状況をユーザーの COBOL ルーチンへ知らせるために、DFSORT は、以下に示すように、レコード・フラグ・フィールド (RECORD-FLAGS) を使用します。



## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

### 0 (FIRST-REC)

DFSORT を離れるレコードは最初に渡されたレコードです。

### 4 (MIDDLE-REC)

DFSORT を離れるレコードは最初に渡されたレコードではありません。

### 8 (END-REC)

DFSORT を離れるレコードで渡すレコードはありません。すべてのレコードがユーザー・ルーチンへ渡されたか、あるいは渡すレコードがありません。

- DFSORT は、離脱レコード・フィールド (LEAVING-REC) に、次の出力レコード (通常は出力域のレコードに続いている) を入れます。VLR には RDW は含まれませんが、DFSORT はこの VLR の長さを離脱レコード長のフィールド (LEAVING-REC-LEN) に入れます。LEAVING-REC-LEN フィールド内の値は、レコードだけの長さで、RDW 用の 4 バイトは含みません。
- ユーザー・ルーチンが挿入または置換レコードを戻りレコード・フィールド (RETURN-REC) に入れる場合は、VLR に RDW が含まれていてはなりません。ユーザー・ルーチンはこのレコードの長さを戻りレコード長さフィールド (RETURN-REC-LEN) に入れます。RETURN-REC-LEN フィールド内の値は、レコードだけの長さで、RDW 用の 4 バイトは含みません。
- DFSORT は、すでに出力域にあるレコードを出力域フィールドのレコード (OUTPUT-REC) に入れます。VLR は RDW を含みません。DFSORT は、出力域長さフィールドのレコード (OUTPUT-REC-LEN) にこの VLR の長さ (RDW 用の 4 バイトを含まない) を入れます。
- DFSORT は COBOL E35 ユーザー・ルーチンに 256 バイトのユーザー出口域フィールド (EXITAREA) を渡します。このフィールドには、COBOL E15 ユーザー・ルーチンにより渡された情報を含めることができます。最初に EXITAREA フィールドが COBOL E15 ユーザー・ルーチンに渡されたときに、この COBOL E15 ルーチンがこのフィールドに渡す情報がない場合、EXITAREA には 256 個のブランクが入り、ユーザー出口域長さフィールド (EXITAREA-LEN) には 256 が入ります。

ユーザー出口域フィールドまたはユーザー出口域長さフィールドに対して行った変更はすべて、DFSORT が呼び出すたびにユーザーの COBOL E35 ルーチンに渡されます。

#### 注:

1. ユーザー出口域の長さフィールドを、256 バイトを超える値にセットできません。
2. VLR レコードは、各レコードの先頭に 4 バイトの RDW フィールドをもっています。最大レコード長に RDW を加えた長さが、ユーザーの出力データ・セットの LRECL 属性に定義される長さです。COBOL プログラムは RDW を使用しないため、COBOL ユーザー出口に定義される最大長は、LRECL 値よりも 4 バイト短くなります。
3. ユーザーがコーディングする必要があるのは、ユーザー・ルーチンが実際に使用する最後のフィールドまで (たとえば、ユーザー出口域を使用しない場合は、OUTPUT-REC-LEN まで) です。
4. DFSORT は、指定された値かそのデフォルトを RECORD ステートメントの L3 に使用して、E35 ユーザー出口が DFSORT に戻すレコードの長さを判別します。固定長レコードの場合、E35 ユーザー出口が変更または挿入する各レ

## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

コードの長さが、必ず L3 の指定値またはデフォルトと一致するようにしてください。可変長レコードの場合、E35 ユーザー出口が変更または挿入する各レコードの長さが、必ず L3 の指定値またはデフォルト以下になるようにしてください。E35 ユーザー出口が間違ったレコード長を DFSORT に渡すと、望ましくない短縮またはアベンドが起きることがあります。

L3 値の詳細については、340 ページの『RECORD 制御ステートメント』を参照してください。

### E35 の戻りコード

COBOL E35 ユーザー・ルーチンは、COBOL の特殊レジスターである RETURN-CODE フィールドに戻りコードを入れて DFSORT へ渡す必要があります。E35 出口の戻りコードは次のとおりです。

戻りコード	説明
00 (X'00')	処置をとらない
04 (X'04')	レコードを削除する
08 (X'08')	戻らない
12 (X'0C')	レコードを挿入する
16 (X'10')	DFSORT を終了する
20 (X'14')	レコードを変更または置き換える

#### 0: 処置をとらない

DFSORT を離れるレコードを変更しないで DFSORT に保持させたい場合は、RETURN-CODE を 0 にセットして戻ります。

#### 4: レコードを削除する

DFSORT を離れるレコードを DFSORT に削除させる場合は、RETURN-CODE を 4 にセットして戻ります。

#### 8: 戻らない

DFSORT は、ユーザーが RETURN-CODE に 8 をセットして渡すまで、ユーザー・ルーチンへ戻り続けます。その後は、DFSORT アプリケーションの処理中にそのユーザー出口が再び使用されることはありません。データ・セットの終わりの後でレコードを挿入しない限り、DFSORT がデータ・セットの終わりを示したときに、RETURN-CODE に 8 をセットして渡す必要があります。これは、レコード・フラグ・フィールドに 8 をセットしてユーザー・ルーチンに入ることにより行います。

処理する入力レコードがまだ残っている段階でユーザー出口ルーチンが戻りコード 8 を DFSORT に渡すと、残りのレコードは DFSORT により処理されますが、ユーザー出口には 渡されません。

出力データ・セットがなく、EOF の前に戻りコード 8 で戻るのが通常である場合は、OPTION 制御ステートメントに NOCHECK を指定することで (インストール時にあらかじめ CHECK=NO を指定していない場合)、ICE025A メッセージが出されるのを防ぎます。

#### 12: レコードを挿入する

DFSORT を離れるレコードの前に、DFSORT に出力レコードを追加させたい場合は、次の処理を行います。

- 挿入レコードを戻りレコード・フィールドへ移動する。

## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

- VLR レコードの場合は、その長さを戻りレコード長さフィールドへ移動する。
- RETURN-CODE を 12 にセットして戻ります。

DFSORT は、挿入されたレコードをレコード出力域フィールドに入れ、DFSORT 離脱レコード・フィールドに以前と同じレコードを入れたままユーザー・ルーチンに戻ります。このようにして、ユーザー・ルーチンはさらにレコードを挿入したり、DFSORT を離れるレコードを処理できます。

データ・セットの終わりの後でレコードを挿入することもできます。DFSORT は、ユーザーが RETURN-CODE 12 を渡している限り、ユーザー・ルーチンへ戻り続けます。これは、RETURN-CODE に 8 をセットして戻るまで続けます。

DASD 作業データ・セット分類の場合、DFSORT はシーケンス検査を行いません。テープ作業データ・セット分類の場合、ユーザーが DFSORT を離れるレコードを削除して、それを置き換えない限り、DFSORT は挿入されたレコードのシーケンス検査を実行しません。

### 16: DFSORT を終了する

DFSORT を終了させたい場合は、RETURN-CODE に 16 をセットして戻ります。すると DFSORT は、戻りコード 16 で呼び出しプログラムまたはシステムへ戻ります。

### 20: レコードを変更する

DFSORT を離れるレコードを変更する場合、次の処理を行います。

- DFSORT を離れるレコードを戻りレコード・フィールドへ移動する。
- 戻りレコード・フィールドのレコードを変更する。
- VLR レコードの場合は、その長さを戻りレコード長さフィールドへ移動する。
- RETURN-CODE を 20 にセットして戻る。

**注:** ユーザー・ルーチンがレコード・サイズを変更する場合は、新しいサイズを RECORD ステートメントに示す必要があります。

### 20: レコードを置き換える

DFSORT を離れるレコードを置き換える場合、次の処理を行います。

- 置き換えレコードを戻りレコード・フィールドへ移動する。
- VLR レコードの場合は、その長さを戻りレコード長さフィールドへ移動する。
- RETURN-CODE を 20 にセットして戻る。

各種の状況での戻りコードの意味の詳細については、417 ページの『E15/E35 戻りコードおよび EXITCK』を参照してください。

## E35 PROCEDURE DIVISION に関する要件

PROCEDURE DIVISION を コーディングする場合、次の要件を満たす必要があります。

- DFSORT へ制御を戻す場合は、GOBACK ステートメントを使用します。

## COBOL ユーザー出口ルーチン (出力フェーズのユーザー出口)

- PROCEDURE DIVISION のヘッダーの USING オプションで、LINKAGE SECTION に 各 01 レベル名を指定します。ヘッダーの前の 01 レベル名を必ずしもすべて使用するわけではない場合でも、使用を計画している最後の名前まで順に指定する必要があります。

例:

410 ページの図 29 の FLR の例については、次のようにコーディングします。

```
PROCEDURE DIVISION USING RECORD-FLAGS, LEAVING-REC,  
RETURN-REC, OUTPUT-REC, UNUSED1, UNUSED2,  
UNUSED3, UNUSED4, EXITAREA-LEN, EXITAREA.
```

410 ページの図 30 の VLR の例については、次のようにコーディングします。

```
PROCEDURE DIVISION USING RECORD-FLAGS, LEAVING-REC,  
RETURN-REC, OUTPUT-REC, UNUSED1,  
LEAVING-REC-LEN, RETURN-REC-LEN,  
OUTPUT-REC-LEN, EXITAREA-LEN, EXITAREA.
```

---

## COBOL で作成されたサンプル・ルーチン

ここでは、COBOL で作成されたプログラム・ユーザー出口の例をいくつか示します。

### COBOL E15 ユーザー出口: レコードの変更

図 31 は、100 バイトの固定長レコードのデータ・セットの場合の COBOL E15 ルーチンの例を示しています。このルーチンは、渡されたレコードの部門フィールドを調べ、次の処置をとります。

- 部門が D29 の場合は、それを J99 に変更します。
- 部門が D29 でなければ、レコードをそのまま受け入れます。

```

IDENTIFICATION DIVISION.
PROGRAM-ID.
    CE15.
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
01 RECORD-FLAGS          PIC 9(8) COMPUTATIONAL.
   88 FIRST-REC          VALUE 00.
   88 MIDDLE-REC         VALUE 04.
   88 END-REC            VALUE 08.
01 NEW-REC.
   05 NFILL1             PIC X(10).
   05 NEW-DEPT           PIC X(3).
   05 NFILL2             PIC X(87).
01 RETURN-REC.
   05 RFILL1             PIC X(10).
   05 RETURN-DEPT        PIC X(3).
   05 RFILL2             PIC X(87).

PROCEDURE DIVISION USING RECORD-FLAGS, NEW-REC, RETURN-REC.

    IF END-REC
        MOVE 8 TO RETURN-CODE
        GO TO BACK-TO-SORT.

    IF NEW-DEPT EQUAL TO "D29"
        MOVE NEW-REC TO RETURN-REC
        MOVE "J99" TO RETURN-DEPT
        MOVE 20 TO RETURN-CODE

    ELSE
        MOVE 0 TO RETURN-CODE.

BACK-TO-SORT.
GOBACK.

```

図 31. COBOL E15 ルーチンの例 (FLR)

## COBOL E35 ユーザー出口: レコードの挿入

図 32 は、200 バイトの可変長レコードのデータ・セットの場合の E35 ルーチンの例を示しています。このルーチンは、渡された各レコード (レコードは部門フィールドで分類されるものとします) の部門フィールドを調べ、次の処置をとります。

- 部門が K22 のレコードを適切な順序で挿入します。
- 渡されたすべてのレコードをそのまま受け入れます。

## COBOL で作成されたサンプル・ルーチン

```
IDENTIFICATION DIVISION.
PROGRAM-ID.
    CE35.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 INSERT-DONE PIC 9(1) VALUE 0.
01 K22-REC.
    05 K22-MANAGER PIC X(20) VALUE "J. DOE".
    05 K22-DEPT    PIC X(3)  VALUE "K22".
    05 K22-FUNC   PIC X(20) VALUE "ACCOUNTING".
    05 K22-LATER  PIC X(30) VALUE SPACES.
01 LEAVING-VAR-LEN PIC 9(8) COMPUTATIONAL.
LINKAGE SECTION.
01 RECORD-FLAGS      PIC 9(8) COMPUTATIONAL.
    88 FIRST-REC      VALUE 00.
    88 MIDDLE-REC     VALUE 04.
    88 END-REC        VALUE 08.
01 LEAVING-REC.
    05 LREC-MANAGER PIC X(20).
    05 LREC-DEPT   PIC X(3).
    05 LREC-FUNC   PIC X(20).
    05 LREC-LATER  OCCURS 1 TO 157 TIMES
                    DEPENDING ON LEAVING-VAR-LEN PIC X.
01 RETURN-REC.
    05 RREC        OCCURS 1 TO 200 TIMES
                    DEPENDING ON RETURN-REC-LEN  PIC X.
01 OUTPUT-REC.
    05 OREC        OCCURS 1 TO 200 TIMES
                    DEPENDING ON OUTPUT-REC-LEN  PIC X.
01 UNUSED1        PIC 9(8) COMPUTATIONAL.
01 LEAVING-REC-LEN PIC 9(8) COMPUTATIONAL.
01 RETURN-REC-LEN PIC 9(8) COMPUTATIONAL.
01 OUTPUT-REC-LEN PIC 9(8) COMPUTATIONAL.

PROCEDURE DIVISION USING RECORD-FLAGS, LEAVING-REC,
    RETURN-REC, OUTPUT-REC, UNUSED1, LEAVING-REC-LEN,
    RETURN-REC-LEN, OUTPUT-REC-LEN.

    IF END-REC
        MOVE 8 TO RETURN-CODE
        GO TO BACK-TO-SORT.
    IF INSERT-DONE EQUAL TO 1
        MOVE 0 TO RETURN-CODE
        GO TO BACK-TO-SORT.
    SUBTRACT 43 FROM LEAVING-REC-LEN
        GIVING LEAVING-VAR-LEN.
    IF LREC-DEPT GREATER THAN K22-DEPT
        MOVE 1 TO INSERT-DONE
        MOVE 43 TO RETURN-REC-LEN
        MOVE K22-REC TO RETURN-REC
        MOVE 12 TO RETURN-CODE
    ELSE
        MOVE 0 TO RETURN-CODE.
BACK-TO-SORT.
GOBACK.
```

図 32. COBOL E35 ルーチンの例 (VLR)

## E15/E35 戻りコードおよび EXITCK

DFSORT の E15 および E35 戻りコードの解釈は、EXITCK=STRONG または EXITCK=WEAK が有効かどうかにより異なります。詳細については、181 ページの『OPTION 制御ステートメント』の EXITCK オプションの説明を参照してください。以下の表は、EXITCK=STRONG および EXITCK=WEAK を使用した場合に考えられるすべての状況における、E15 および E35 戻りコードの正確な意味を示したものです。

注:

- EXITCK=STRONG と EXITCK=WEAK のどちらが有効であるかは、メッセージ ICE132I から判断できます。
- EXITCK=WEAK を使用した場合、E15 および E35 ユーザー出口ルーチンの論理内のエラーを検出することが難しくなることがあります。
- 分類アプリケーションにテープ作業データ・セットが指定されている場合や、組み合わせアプリケーションにブロック・セット手法を選択しない場合は、EXITCK=WEAK は EXITCK=STRONG と同様に処理されます。

表 53. SORTIN データ・セットを用いない E15

E15 戻りコード	EXITCK=STRONG での意味	EXITCK=WEAK での意味
0	無効	戻らない
4	無効	戻らない
8	戻らない	戻らない
12	レコードを挿入する	レコードを挿入する
16	DFSORT を終了する	DFSORT を終了する
20 (COBOL のみ)	無効	戻らない
その他すべて	無効	無効

表 54. 入力の終わりの前に SORTIN データ・セットを用いる E15

E15 戻りコード	EXITCK=STRONG または EXITCK=WEAK での意味
0	処置をとらない / レコードは変更されない
4	レコードを削除する
8	戻らない
12	レコードを挿入する
16	DFSORT を終了する
20 (COBOL のみ)	レコードを変更 / 置換する
その他すべて	無効

## E15/E35 戻りコードおよび EXITCK

表 55. 入力の終わりの後で SORTIN データ・セットを用いる E15

E15 戻りコード	EXITCK=STRONG での意味	EXITCK=WEAK での意味
0	無効	戻らない
4	無効	戻らない
8	戻らない	戻らない
12	レコードを挿入する	レコードを挿入する
16	DFSORT を終了する	DFSORT を終了する
20 (COBOL のみ)	無効	戻らない
その他すべて	無効	無効

表 56. 入力の終わりの前に SORTOUT または OUTFIL データ・セットを用いる E35

E35 戻りコード	EXITCK=STRONG または EXITCK=WEAK での意味
0	処置をとらない / レコードは変更されない
4	レコードを削除する
8	戻らない
12	レコードを挿入する
16	DFSORT を終了する
20 (COBOL のみ)	レコードを変更 / 置換する
その他すべて	無効

表 57. 入力の終わりの前に SORTOUT または OUTFIL データ・セットを用いない E35

E35 戻りコード	EXITCK=STRONG での意味	EXITCK=WEAK での意味
0	無効	レコードを削除する
4	レコードを削除する	レコードを削除する
8	戻らない	戻らない
12	無効	レコードを削除する
16	DFSORT を終了する	DFSORT を終了する
20 (COBOL のみ)	無効	レコードを削除する
その他すべて	無効	無効

表 58. 入力の終わりの後で SORTOUT または OUTFIL データ・セットを用いる E35

E35 戻りコード	EXITCK=STRONG での意味	EXITCK=WEAK での意味
0	無効	戻らない
4	無効	戻らない
8	戻らない	戻らない
12	レコードを挿入する	レコードを挿入する
16	DFSORT を終了する	DFSORT を終了する
20 (COBOL のみ)	無効	戻らない
その他すべて	無効	無効



## E15/E35 戻りコードおよび EXITCK

表 59. 入力の終わりの後で SORTOUT または OUTFIL データ・セットを用いない E35

E35 戻りコード	EXITCK=STRONG での意味	EXITCK=WEAK での意味
0	無効	戻らない
4	無効	戻らない
8	戻らない	戻らない
12	無効	戻らない
16	DFSORT を終了する	DFSORT を終了する
20 (COBOL のみ)	無効	戻らない
その他すべて	無効	無効

E15/E35 戻りコードおよび EXITCK

---

## 第 5 章 プログラムからの DFSORT の呼び出し

DFSORT の動的呼び出し . . . . .	421
システム・マクロ命令とは . . . . .	421
システム・マクロ命令の使用 . . . . .	422
JCL DD ステートメントの使用 . . . . .	422
DFSORT 制御ステートメントをプログラムから指定変更する . . . . .	422
24 ビット・パラメーター・リストを用いた DFSORT の呼び出し . . . . .	423
プログラム制御ステートメントの準備 . . . . .	423
制御ステートメント・イメージの例 . . . . .	424
24 ビット・パラメーター・リストの形式 . . . . .	424
拡張パラメーター・リストによる DFSORT の呼び出し . . . . .	431
プログラム制御ステートメントの準備 . . . . .	431
拡張パラメーター・リストの形式 . . . . .	431
マクロ命令の書き方 . . . . .	434
パラメーター・リストの例 . . . . .	435
動的呼び出しに関する制約事項 . . . . .	439
組み合わせに関する制約事項 . . . . .	439
コピーに関する制約事項 . . . . .	439

---

### DFSORT の動的呼び出し

DFSORT は、COBOL または PL/I で作成されたプログラムから動的に呼び出すことができます。動的呼び出しに関する特定情報については、該当する COBOL および PL/I のプログラミングの手引きを参照してください。JCL に関する要件は、アセンブラーの場合と同じです。

本節では、入力ストリーム内の EXEC ジョブ制御ステートメントの代わりにシステム・マクロ命令を使用して、ユーザーのアセンブラー・プログラム内から DFSORT を開始するために知っておく必要のある事項について説明します。PL/I および COBOL から DFSORT を呼び出す場合の特定の制約事項については、439 ページの『動的呼び出しに関する制約事項』に記載してあります。

---

### システム・マクロ命令とは

システム・マクロ命令とは、サービス要求を制御プログラムに知らせるための、IBM 提供のマクロです。これらの命令は、アセンブラー言語でプログラミングしている場合のみ使用できます。システム・マクロ命令は、IBM 提供のマクロ定義を使用してアセンブラー・プログラムが処理するもので、ユーザーの制御プログラムをインストールするときにマクロ・ライブラリーに入れられます。

LINK、ATTACH、または XCTL のシステム・マクロ命令のうちのいずれかを指定することにより、プログラムへ制御を渡すことができます。

これらの命令のうちの 1 つを出すと、DFSORT の最初のロード・モジュールが主記憶域へ呼び込まれます。ユーザー・プログラムと DFSORT の間の関係は、どの命令を使用したかにより異なります。マクロ命令の詳細およびそれらの使用方法については、アプリケーション開発の手引き およびアプリケーション開発マクロ解説書を参照してください。

---

## システム・マクロ命令の使用

システム・マクロ命令を使用して DFSORT 処理を開始するためには、次のことを行う必要があります。

- 必要なジョブ制御言語 (JCL) の DD ステートメントを作成します。
- アセンブラー DC 命令のオペランドとして DFSORT 制御ステートメントを作成します。(DFSPARM または SORTCNTL データ・セットを使用して、プログラム制御ステートメントを指定するほうが便利な場合があります。詳細については、111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』を参照。)
- DFSORT に渡す情報を含むパラメーター・リストおよびそのパラメーター・リストのアドレスを含むポインターを作成します。DFSORT は、24 ビット・パラメーター・リストと拡張パラメーター・リストの 2 つのタイプのパラメーター・リストを受け入れます。ユーザーはどちらのパラメーター・リストも選択できますが、拡張パラメーター・リストは 24 ビット・パラメーター・リストのスーパーセット機能を実行することができるため、新しい DFSORT アプリケーション用に使用する必要があります。

**注:** システムの EXEC PARM パラメーター・リストの規則 (たとえば、パラメーター・リストは 16 メガバイト境界より下に入っていないなければならない) を守れば、それを使用して、DFSORT を呼び出すこともできます。DFSORT は、EXEC PARM パラメーター・リストを使用する呼び出しを、プログラム呼び出しではなく、直接呼び出しと解釈します。

- ICEMAN、SORT、IERRC000、または IGHRC000 のうちの 1 つを入り口点名として指定するマクロ命令を作成します。

**注:** DFSORT に渡される保管域は、フルワード境界から開始される必要があります。

さらに、次の規則が適用されます。

- DFSORT を繰り返し呼び出す場合 (たとえば、E15 または E35 ユーザー出口から)、必ず最後に呼び出された分類が終了するのを待ってからでないと、以前呼び出された分類のユーザー出口へ制御を戻すことはできません。

---

## JCL DD ステートメントの使用

DFSORT を別のプログラムから呼び出す際には、通常、JCL DD ステートメントが必要です。ステートメントおよびそれらに必要なパラメーターについては、詳細に説明しています。

---

## DFSORT 制御ステートメントをプログラムから指定変更する

プログラム (たとえば、COBOL SORT 動詞 または PLISRTx ルーチンなど) により生成され、あるいは渡された制御ステートメントを、DFSORT の DFSPARM データ・セットを使用して指定変更できます。

## DFSORT 制御ステートメントをプログラムから指定変更する

たとえば、COBOL SORT 動詞が生成した SORT ステートメントを、DFSORT の 2000 年機能を使用するために次の方法で指定変更できます。

```
//DFSPARM DD *  
OPTION Y2PAST=1956      * set fixed CW of 1956-2055  
SORT FIELDS=(11,6,Y2T,A, * sort C'yymmdd' using CW  
                31,10,CH,A) * sort other key  
/*
```

プログラムからの DFSORT の複数呼び出しに対して、DFSPARM を使用して、異なる DFSORT 制御ステートメントを指定することもできます。ただし、制御ステートメントを一時または永続データ・セットに配置し、FREE=CLOSE を使用することが必要です。3 つの SORT 動詞をもつ COBOL プログラムの制御ステートメントを、DFSPARM を使用して指定変更する例を示します。

```
//DFSPARM DD DSN=DP1,DISP=SHR,FREE=CLOSE  
//DFSPARM DD DSN=DP2,DISP=SHR,FREE=CLOSE  
//DFSPARM DD DSN=DP3,DISP=SHR,FREE=CLOSE
```

DP1、DP2、および DP3 には、ユーザーが自由に DFSORT 制御ステートメントを設定できます。DP1 のステートメントは DFSORT の最初の呼び出しで使用され、DP2 のステートメントは 2 番目、DP3 のステートメントは 3 番目の呼び出しで使用されます。

DFSPARM の詳細については、85 ページの『DFSPARM DD ステートメント』を参照してください。

---

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

### プログラム制御ステートメントの準備

24 ビットパラメーター・リストを使用する際には、実行時に使用される各制御ステートメントの有効なイメージの開始アドレスおよび終了アドレスを指定する必要があります。アセンブラー DC 命令を使用して、EBCDIC 形式の文字ストリングとしてのイメージを指定する必要があります。プログラム制御ステートメントを作成する場合の規則は、次のとおりです。

- 少なくとも 2 つの制御ステートメントを指定します。一般的には、SORT または MERGE、および RECORD です。制御ステートメントを 16 以上指定すると、最初の 15 制御ステートメントだけが受け入れられ、残りはすべて無視されます。制御ステートメントを SORTCNTL または DFSPARM で指定することもできます。
- MODS ステートメントは、E15、E32、および E35 以外のユーザー出口が使用される場合に必要です。このステートメントは、E15 または E35 ルーチンのアドレスがパラメーター・リストで渡されない場合にも必要になります。
- 次の制御ステートメントは、24 ビット・パラメーター・リストを使用して渡すことができます。すなわち、SORT または MERGE、RECORD、ALTSEQ、DEBUG、MODS、SUM、INREC、OUTREC、INCLUDE または OMIT、および OUTFIL の各ステートメントです。

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

- 命令定義子 (たとえば SORT) の後には、少なくとも 1 つの空白が必要です。制御ステートメントは、1 つ以上の空白で始めたり、終わらせたりできますが、それ以外の空白は使用できません。
- ステートメントの内容と形式については、次の事項を除き、111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』で説明されています。
  - ラベルは使用できません。ただし、先行空白は任意指定です。
  - 各制御ステートメント・イメージは、1 つ以上のアセンブラー DC 命令で連続的に定義しなければならないため、ステートメントの明示的および暗黙的な連結は必要もなければ、また許可もされません。
- 注釈ステートメント、空白・ステートメント、注釈フィールドはいずれも使用できません。
- ATTACH を使用してプログラムを開始する場合は、チェックポイント / 再始動機能を使用できません。したがって、SORT ステートメント・イメージに CKPT を指定できません。  
指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### 制御ステートメント・イメージの例

```
SORTBEG DC C' SORT FIELDS=(10,15,CH,A)'  
SORTEND DC C' '
```

このように後書き空白を別個に定義する形式を使用すると、SORTEND という名前でステートメントの再度のバイト (SORT ステートメントの終わりのアドレス) を参照できます。

```
INCLBEG DC C' INCLUDE COND=(5,3,CH,NE,C'J82')'  
INCLEND DC C' '
```

**注:** 1 つの単一アポストロフィを表す場合、アセンブラーは 2 つの単一アポストロフィを必要とします。

### 24 ビット・パラメーター・リストの形式

425 ページの図 33 は、24 ビット・パラメーター・リストの形式および DFSORT へ渡す必要のあるアドレスを含むポインターを示しています。図の後に、パラメーター・リストの各項目について詳述しています。

指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

オフセット		バイト 1	バイト 2	バイト 3 およびバイト 4	注
(16 進)	(10 進)	未使用	未使用	パラメーター・リストの長さ (バイト単位)	
-2	-2				
2	2	X'00'	SORT または MERGE ステートメント・イメージの開始アドレス		1,3
6	6	X'00'	SORT または MERGE ステートメント・イメージの終了アドレス		1,5
A	10	X'00'	RECORD ステートメント・イメージの開始アドレス		1,3
E	14	X'00'	RECORD ステートメント・イメージの終了アドレス		1,5
12	18	X'00'	E15 または E32 ルーチンのアドレス (ない場合、ゼロ)		1
16	22	X'00'	E35 ルーチンのアドレス (ない場合、ゼロ)		1
1A	26	X'02'	MODS ステートメント・イメージの開始アドレス		2,3
1E	30	X'00'	MODS ステートメント・イメージの終了アドレス		2,5
22	34	X'00'	主記憶域値		2
26	38	X'01'	予約記憶域値		2
2A	42	X'03'	8 文字メッセージ DD 名のアドレス		2
2E	46	X'04'	入力ファイルの数 (E32 による MERGE)		2,4
32	50	X'05'	DEBUG ステートメント・イメージの開始アドレス		2,3
36	54	X'00'	DEBUG ステートメント・イメージの終了アドレス		2,5
3A	58	X'06'	ALTSEQ ステートメント・イメージの開始アドレス		2,3
3E	62	X'00'	ALTSEQ ステートメント・イメージの終了アドレス		2,5
42	66	X'F6'	256 バイト ALTSEQ 変換テーブルのアドレス		2
46	70	X'F7'	ユーザー出口のアドレス定数		2
4A	74	X'FD'	X'FD' より後の 3 バイトは無視される		2
4E	78	X'FE'	104 バイト ESTAE 作業域へのポインターのアドレス (またはゼロ)		2
52	82	X'FF'	メッセージ・オプション		2
56	86	"SORT" DD ステートメント名の 4 文字接頭部			2
5A	90	X'07'	SUM ステートメント・イメージの開始アドレス		2,3
5E	94	X'00'	SUM ステートメント・イメージの終了アドレス		2,5
62	98	X'08'	INCLUDE または OMIT ステートメント・イメージの開始アドレス		2,3
66	102	X'00'	INCLUDE または OMIT ステートメント・イメージの終了アドレス		2,5
6A	106	X'09'	OUTREC ステートメント・イメージの開始アドレス		2,3
6E	110	X'00'	OUTREC ステートメント・イメージの終了アドレス		2,5
72	114	X'0A'	INREC ステートメント・イメージの開始アドレス		2,3
76	118	X'00'	INREC ステートメント・イメージの終了アドレス		2,5
7A	122	X'0B'	OUTFIL ステートメント・イメージの開始アドレス		2,3
7E	126	X'00'	OUTFIL ステートメント・イメージの終了アドレス		2,5

図 33. 24 ビット・パラメーター・リスト

### 図 33 の注:

1. 必須項目。図で示す相対位置に表示される必要があります。示されているオフセットは、この項目の実際のオフセットです。
2. オプション項目。必須項目の後の任意の位置に指定できます。ここで表示さ

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

れているオフセットは、識別のためだけのものです。この項目の実際のオフセットは変動します。オプション項目は連続している必要がありますが、指定の順序は任意です。

3. 特定の制御ステートメント。図示するだけのために示したものです。SORT または MERGE、RECORD、ALTSEQ、DEBUG、MODS、SUM、INREC、OUTREC、INCLUDE または OMIT、および OUTFIL は、X'00' (注 1 を参照)、X'02'、X'05' ~ X'0B'、X'10'、X'16'、X'18'、および X'20' ~ X'29' のいずれかの 16 進入力コードを使用して渡すことができます。
4. MERGE ステートメントが使用されていて、入力が E32 ユーザー出口から行われる場合、必須項目です。MERGE ステートメントの FILES オプションが指定されている場合は、この項目は必須ではありません。
5. 必須項目。制御ステートメントの終了アドレスを含み、同じ制御ステートメントの開始アドレスを含む項目の直後である必要があります。

パラメーター・リストの各項目の指定は、次のとおりです。

### バイト 説明

#### -2 ~ -1

使用しません。

#### 0 ~ +1

バイト・カウント。この 2 バイト・フィールドには、パラメーター・リストの長さがバイト単位で含まれます。リストにより占められたバイトの数をカウントする際には、この 2 バイト・フィールドは含まれません。

必須項目の全長は 24 (X'0018') です。すべてのオプション項目は 4 バイトの長さですが、制御ステートメント・イメージを参照する項目は例外で、これらはそれぞれ 8 バイトの長さです。

#### 2 ~ 5

SORT または MERGE ステートメント・イメージの開始アドレス。このフルワードの最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

#### 6 ~ 9

SORT または MERGE ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

#### 10 ~ 13

RECORD ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

#### 14 ~ 17

RECORD ステートメントの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

#### 18 ~ 21

ユーザー・プログラムが主記憶域に入れた E15 または E32 ルーチンのアドレス (それがあある場合)。それがあない場合は、すべてゼロです。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。



## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

### 22 ~ 25

ユーザー・プログラムが主記憶域に入れた E35 ルーチンのアドレス (それがあある場合)。それがあない場合は、すべてゼロです。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 26 ~ 29

MODS ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'02' が含まれます。

### 30 ~ 33

MODS ステートメントの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 34 ~ 37

主記憶域の値。最初のバイトには X'00' が含まれます。次の 3 バイトには、文字 MAX または 16 進値が入ります。このオプションを使用して、SIZE インストール・オプションを一時的に指定変更できます。指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。この値の説明については、181 ページの『OPTION 制御ステートメント』の MAINSIZE パラメーターの項を参照してください。

### 38 ~ 41

予約済みの主記憶域の値。最初のバイトには X'01' が含まれます。次の 3 バイトには、予約するバイト数を指定する 16 進値が入ります。この値の最小値は 4 キロバイトです。この値の説明については、181 ページの『OPTION 制御ステートメント』の RESINV パラメーターの項を参照してください。

このオプションを使用して、RESINV インストール・オプションを一時的に指定変更できます。指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### 42 ~ 45

メッセージ DD 名。最初のバイトには X'03' が含まれます。次の 3 バイトには、メッセージ・データ・セット用の 8 バイトの DD ステートメントのアドレスが入り、必要であれば、右側にはブランクが埋め込まれます。名前は任意の有効な DD ステートメントを使用できますが、固有なものである必要があります。

このオプションを使用して、MSGDDN インストール・オプションを一時的に指定変更できます。指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。メッセージ・データ・セットの使用の詳細については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

### 46 ~ 49

組み合わせを行う入力ファイルの数。この項目が必要になるのは、FILE オプションの指定なしで MERGE ステートメントが使用され、しかも組み合わせのための入力 E32 ユーザー出口を介して提供される場合だけです。最初のバイトには X'04' が含まれます。次の 3 バイトには、ファイルの数

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

が 16 進数で入ります。指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### 50 ~ 53

DEBUG ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'05' が含まれます。

### 54 ~ 57

DEBUG ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 58 ~ 61

ALTSEQ ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'06' が含まれます。

### 62 ~ 65

ALTSEQ ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 66 ~ 69

ALTSEQ ステートメントの代わりに提供される 256 バイトの変換テーブルのアドレス。最初のバイトには X'F6' が含まれます。このパラメーターがあると、X'06' パラメーターは無視されます。指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### 70~73

ユーザー出口アドレス定数。DFSORT が X'F7' を X'00' で置き換えた後で、これらの 4 バイトは、E15 (E15 パラメーター・リストのオフセット 4)、E32 (E32 パラメーター・リストのオフセット 8)、または E35 (E35 パラメーター・リストのオフセット 8) に渡されます。

**注:** ユーザー出口アドレス定数を、従来の組み合わせアプリケーションまたはテープ作業データ・セット分類のアプリケーションに使用できません。

### 74 ~ 77

最初のバイトの X'FD' (VLSHRT オプション) は、可変長入力レコードが短すぎて、指定された制御フィールド、比較フィールド、または要約フィールドのすべてを入れることができないと分かった場合に、DFSORT が処理を継続することを指定します。このオプションの詳細については、181 ページの『OPTION 制御ステートメント』の VLSHRT パラメーターの説明を参照してください。このオプションを使用して、VLSHRT インストール・オプションを一時的に指定変更できます。指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### 78 ~ 81

最初のバイトに X'FE' が入っている場合、次の 3 バイトを使用して、ESTAE 情報が保管される 104 バイトの保管域フィールドのアドレスを渡すことができます。ESTAE 情報を保管しない場合は、これらのバイトはゼロである必要があります。

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

システムまたはユーザー出口の異常終了が生じた場合は、DFSORT ESTAE リカバリー・ルーチンは、より高位レベルのいずれかの ESTAE リカバリー・ルーチンへ戻る前に、SDWA の最初の 104 バイトをこの保管域にコピーします。

DFSORT ESTAE リカバリー・ルーチンの詳細については、747 ページの『付録 E. DFSORT の異常終了処理』を参照してください。

### 82 ~ 85

メッセージ・オプション。最初のバイトには X'FF' が含まれます。次の 3 バイトには、文字 NOF、(I)、または (U) が入ります。このオプションを使用して、MSGPRT インストール・オプションを一時的に指定変更できます。

**NOF** メッセージ・ステートメントと制御ステートメントは印刷されません。重要メッセージはマスター・コンソールに書き込まれます。

**(I)** 診断メッセージ (ICE800I から ICE999I) 以外のすべてのメッセージが印刷されます。重要メッセージはマスター・コンソールにも書き込まれます。制御ステートメントは、LIST が有効な場合のみ印刷されます。

**(U)** 重大メッセージだけが印刷されます。重要メッセージはマスター・コンソールにも書き込まれます。制御ステートメントは印刷されません (NOLIST が強制されます)。

すべてのメッセージはメッセージ・データ・セットに書き込まれます。メッセージ・データ・セットの使用の詳細については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。指定変更および適用度の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

互換性のため、形式 (NO、(AB、(AP、(AC、(CB、(CC、(CP、(PC、(SC、および (SP も受け入れられます。

次のリストは、これらの別名と等価の指定を示しています。

表 60. メッセージ・オプションの別名

オプション	MSGPRT	MSGCON
(NO	NONE	NONE
(AB	ALL	ALL
(AP	ALL	CRITICAL
(AC	NONE	ALL
(CB	CRITICAL	CRITICAL
(CC	NONE	CRITICAL
(CP	CRITICAL	CRITICAL
(PC	ALL	ALL
(SC	ALL	CRITICAL
(SP	CRITICAL	ALL

### 86 ~ 89

次の DD 名の中の『SORT』を置き換える 4 文字。すなわち、SORTIN、

## 24 ビット・パラメーター・リストを用いた DFSORT の呼び出し

SORTOUT、SORTINn、SORTINnn、SORTOFd、SORTOFdd、SORTWKd、SORTWKdd、および SORTCNTL です。プログラム・ステップで複数回 DFSORT を動的に呼び出す場合は、このオプションを使用してください。

この 4 文字はすべて英数字または国別文字 (\$、#、または @) である必要があります。先頭文字は英字を指定してください。また、予約済みの名前 DIAG、BALN、OSCL、POLY、CRCX、PEER、LIST、および SYSc (c は任意の英数字) は使用できません。これらの名前を使用すると、その 4 文字は無視されます。

たとえば、置換文字として ABC# を使用すると、DFSORT は、SORTIN、SORTCNTL、SORTWKdd、および SORTOUT の代わりに、それぞれ ABC#IN、ABC#CNTL、ABC#WKdd、および ABC#OUT のステートメントを使用します。

注: このパラメーターは、SORTDD=cccc 実行時オプションと同じです。

### 90 ~ 93

SUM ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'07' が含まれます。

### 94 ~ 97

SUM ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 98 ~ 101

INCLUDE または OMIT ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'08' が含まれます。

### 102 ~ 105

INCLUDE または OMIT ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 106 ~ 109

OUTREC ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'09' が含まれます。

### 110 ~ 112

OUTREC ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 114 ~ 116

INREC ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'0A' が含まれます。

### 118 ~ 121

INREC ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

### 122 ~ 125

OUTFIL ステートメント・イメージの開始アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'0B' が含まれます。

OUTFIL ステートメント・イメージの終了アドレス。最後の 3 バイト内に存在する必要があります。最初のバイトには X'00' が含まれます。

---

## 拡張パラメーター・リストによる DFSORT の呼び出し

### プログラム制御ステートメントの準備

拡張パラメーター・リストを使用する場合は、制御ステートメントはパラメーター・リストが指し示す単一区域に作成されます。制御ステートメント域は次のものから構成されます。

- 2 バイト・フィールド。後に続く文字ストリングの長さ (2 進表示) を入れます。
- EBCDIC 形式の文字ストリングで、アセンブラー DC 命令を使用し、実行時に使用する制御ステートメントの有効なイメージを含んでいます。

プログラム制御ステートメントを作成する場合の規則は、次のとおりです。

- 制御ステートメントは 1 つ以上のブランクで区切る必要があります。最初のステートメントの前のブランクはオプションですが、後書きブランクは必須です。ラベル、注釈ステートメント、および注釈フィールドは使用できません。各制御ステートメント・イメージは、1 つ以上のアセンブラー DC 命令で連続的に定義しなければならないため、ステートメントの明示的および暗黙的な連続は必要なく、また許可もされません。
- E15、E18、E32、E35、および E39 以外のユーザー出口が使用される場合、あるいはパラメーター・リストにより E15、E18、E32、または E39 ルーチン・アドレスが渡されない場合は、MODS ステートメントが必要です。
- 111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』で説明されている制御ステートメントはすべて指定できます。どれも必須ではありませんが、パラメーター・リスト、SORTCNTL、または DFSPARM には、SORT、MERGE、または OPTION COPY を指定する必要があります。
- ATTACH を使用してプログラムを開始する場合は、チェックポイント / 再始動機能を使用できません。したがって、SORT または OPTION ステートメントに CKPT を指定できません。

指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

### 拡張パラメーター・リストの形式

432 ページの図 34 は、拡張パラメーター・リストの形式およびリストのアドレスの入っているポインター (ユーザーが DFSORT に渡さなければならない) を示したものです。

最初のパラメーターは指定する必要があります。X'FFFFFFF' を含む 4 バイトのフィールドを使用して、パラメーター・リストの終わりを示す必要があります。このフィールドは最初のパラメーターの後であればどこにでもコーディングできます。

パラメーターを指定する場合は、パラメーターは指示された位置に入れる必要があります。また、31 ビット・アドレスあるいはクリーン (最初の 8 ビットがゼロ) 24 ビット・アドレスを含んでいる必要があります。パラメーターを指定しない場合

## 拡張パラメーター・リストによる DFSORT の呼び出し

は、ゼロとして指定したものと見なされます。指定変更および適用できる機能の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

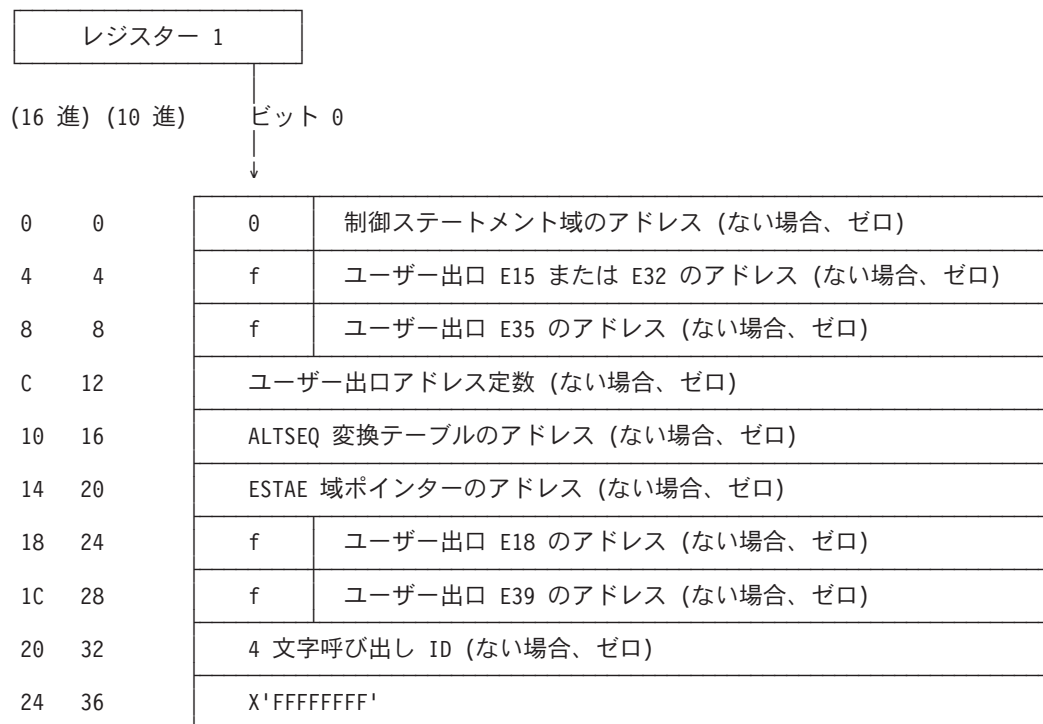


図 34. 拡張パラメーター・リスト

図の後に、パラメーター・リストの各項目について詳述しています。

### バイト 説明

#### 0 ~ 3

必須。DFSORT 制御ステートメントがある場合、それを含む区域のアドレス。ない場合は、すべてゼロです。これが拡張パラメーター・リストであると示すためには、最高位ビットが 0 である必要があります。

制御ステートメント区域の形式については、前の節を参照してください。この区域は 2 バイトの長さフィールドで始まらなければならないことに注意してください。

このパラメーターをゼロと指定する場合は、必要な制御ステートメントをすべて DFSPARM または SORTCNTL で指定する必要があります。

#### 4 ~ 7

オプション。ユーザー・プログラムが主記憶域内に入れた (たとえば、LOAD により) E15 または E32 ユーザー出口ルーチンがあれば、そのアドレス。ない場合は、すべてゼロになります。

f (ビット 0) は、次の意味をもっています。

0 = 24 ビット・アドレッシングを有効にしてユーザー出口を呼び出さず (AMODE 24)。

## 拡張パラメーター・リストによる DFSORT の呼び出し

1 = 31 ビット・アドレッシングを有効にしてユーザー出口を呼び出し  
ます (AMODE 31)。

**注:** ブロック・セットまたはピアレッジ / ベール手法が選択されていない場  
合は、必ず 24 ビット・アドレッシング (AMODE 24) を有効にしてユ  
ーザー出口に入ります。

### 8 ~ 11

オプション。ユーザー・プログラムが主記憶域内に入れた (たとえば、  
LOAD により) E35 ユーザー出口ルーチンがあれば、そのアドレス。ない  
場合は、すべてゼロになります。

f (ビット 0) は、次の意味をもっています。

0 = 24 ビット・アドレッシングを有効にしてユーザー出口を呼び出し  
ます (AMODE 24)。

1 = 31 ビット・アドレッシングを有効にしてユーザー出口を呼び出し  
ます (AMODE 31)。

**注:** ブロック・セットまたはピアレッジ / ベール手法が選択されていない場  
合は、必ず 24 ビット・アドレッシング (AMODE 24) を有効にしてユ  
ーザー出口に入ります。

### 12 ~ 15

オプション。このフィールドは E15、E32、または E35 ユーザー出口ルー  
チンへ渡されます。

**注:** ユーザー出口アドレス定数を、従来の組み合わせアプリケーションまた  
はテープ作業データ・セット分類のアプリケーションに使用できませ  
ん。

### 16 ~ 19

オプション。ALTSEQ ステートメントの代わりに 256 バイトの ALTSEQ  
変換テーブルがある場合は、そのアドレス。ない場合は、すべてゼロです。  
このオプションを使用して、インストール時に指定された ALTSEQ 変換テ  
ーブルを指定変更できます。指定変更および適用できる機能の詳細につい  
ては、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参  
照してください。

### 20 ~ 23

オプション。ESTAE 情報が保管される 112 バイトの作業域のアドレスを含  
む 4 バイトのフィールドのアドレス。保管する ESTAE 情報がない場合  
は、すべてゼロです。

システムまたはユーザー出口アベンドが生じた場合、DFSORT リカバリ  
ー・ルーチンはユーザーの ESTAE リカバリー・ルーチンへ戻る前に、ソフ  
トウェア診断作業域 (SDWA) の最初の 112 バイトをこの保管域にコピーし  
ます。

### 24 ~ 27

オプション。ユーザー・プログラムが主記憶域内に入れた (たとえば、  
LOAD により) E18 ユーザー出口ルーチンがあれば、そのアドレス。ない  
場合は、すべてゼロになります。

## 拡張パラメーター・リストによる DFSORT の呼び出し

**注:** 組み合わせアプリケーションおよびテープ作業データ・セットの分類アプリケーションの場合、このパラメーターは無視されます。

f (ビット 0) は、次の意味をもっています。

0 = 24 ビット・アドレッシングを有効にしてユーザー出口を呼び出します (AMODE 24)。

1 = 31 ビット・アドレッシングを有効にしてユーザー出口を呼び出します (AMODE 31)。

**注:** ブロック・セットまたはピアレッジ / ベール手法が選択されていない場合は、必ず 24 ビット・アドレッシング (AMODE 24) を有効にしてユーザー出口に入ります。

### 28 ~ 31

オプション。ユーザー・プログラムが主記憶域内に入れた (たとえば、LOAD により) E39 ユーザー出口ルーチンがあれば、そのアドレス。ない場合は、すべてゼロになります。

**注:** 従来の組み合わせアプリケーションおよびテープ作業データ・セットの分類アプリケーションの場合、このパラメーターは無視されます。

f (ビット 0) は、次の意味をもっています。

0 = 24 ビット・アドレッシングを有効にしてユーザー出口を呼び出します (AMODE 24)。

1 = 31 ビット・アドレッシングを有効にしてユーザー出口を呼び出します (AMODE 31)。

**注:** ブロック・セットまたはピアレッジ / ベール手法を選択しない場合は、必ず 24 ビット・アドレッシング (AMODE 24) を有効にしてユーザー出口に入ります。

### 32 ~ 35

オプション。DFSORT に対するこの呼び出しの識別子として使用される 4 文字。このフィールドを使用して、DFSORT を複数回呼び出すプログラムからの DFSORT 呼び出しを、それぞれ固有に識別できます。DFSORT は、ユーザーが指定した通りにフィールド識別子を示すメッセージ ICE200I を印刷します。このフィールドでは、有効な文字についての検査は行われません。

フィールド識別子を指定する場合は、指示された位置に含める必要があります。識別子フィールドがゼロ (X'00000000') の場合、または X'FFFFFFFF' を使用して、フィールド識別子の前あるいはフィールド識別子がある場所でパラメーター・リストを終了させる場合、DFSORT はメッセージ ICE200I を印刷しません。

**注:** リストはどのパラメーターの後でも終了できます。リストの最後のパラメーターの後には、X'FFFFFFFF' を続ける 必要があります。

---

## マクロ命令の書き方

LINK、ATTACH、または XCTL マクロ命令を書く場合、次のことが必要です。



- 命令の EP パラメーターに SORT (入り口点) を指定します。(これは分類、組み合わせ、およびコピー・ジョブに適用されます。)
- パラメーター・リストを指し示すポインタのアドレスを、レジスター 1 にロードします (あるいは、それを命令の MF パラメーターに渡します)。

注: ATTACH を使用する場合は、ECB パラメーターも必要になる場合があります。

パラメーター・リストに E15 ユーザー出口ルーチンを指定すると、DFSORT は SORTIN データ・セットを無視します。したがって、E15 ユーザー・ルーチンはすべての入力レコードを DFSORT へ渡す必要があります。E32 ルーチン・アドレスを指定すると、同様のことが組み合わせにも当てはまります。つまり、ユーザー・ルーチンは、入力データ・セットが完了するまで戻りコード 12 (挿入レコード) を出し、完了後、戻りコード 8 (『戻り不可』) を出す必要があります。

パラメーター・リストに E35 ルーチン・アドレスを指定すると、DFSORT は SORTOUT データ・セットを無視します。OUTFIL 処理を使用しない限り、ユーザー・ルーチンはすべての出力レコードの処理を行う必要があります。ユーザー・ルーチンは、出力データ・セットの各レコードに対して戻りコード 4 (レコードの削除) を出す必要があります。プログラムがレコードをすべて削除すると、ユーザー・ルーチンは戻りコード 8 (『戻り不可』) を出します。

DFSORT は、実行の完了後、制御を呼び出しルーチンへ渡します。

単一のタスクが 2 つ以上のプログラム・アプリケーションを接続している場合、標準 DD 名を変更して、固有なものにする必要があります。その方法および指定変更の規則については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

同一プログラムから 2 つ以上の DFSORT アプリケーションを ATTACH する場合、DFSORT とユーザー出口が再入可能としてインストールされていない限り、前の各アプリケーションが完了してから次のアプリケーションをタスク生成する必要があります。

XCTL を介して DFSORT を開始するときは、パラメーター・リスト、アドレス・リスト、任意指定パラメーター、および変更ルーチン (それがあある場合) が保管されている区域について、特別な考慮が必要です。モジュールは DFSORT によりオーバーレイされるため、この情報をそのモジュール内に入れてはなりません。

この問題を解決するための 2 つの方法があります。1 つは、XCTL を出すモジュールを接続しているタスクに制御情報を入れる方法です。もう 1 つは、XCTL を出すモジュールがまず最初に GETMAIN マクロ命令を出し、獲得した主記憶域に制御情報を入れる方法です。この区域は、XCTL が出される際にオーバーレイされません。この区域内の制御情報のアドレスは、汎用レジスター 1 で DFSORT に渡す必要があります。

## パラメーター・リストの例

### 24 ビット・パラメーター・リストの例 1

## マクロ命令の書き方

図 35 は、分類アプリケーションの場合に、主記憶域オプションを指定するために使用する 24 ビット・パラメーター・リストの形式を示したものです。

(16 進)(10 進)	バイト 1	バイト 2	バイト 3 および 4
-2 -2	未使用		X'001C'
2 2	X'00'	SORT ステートメントの開始アドレス	
6 6	X'00'	SORT ステートメントの終了アドレス	
A 10	X'00'	RECORD ステートメントの開始アドレス	
E 14	X'00'	RECORD ステートメントの終了アドレス	
12 18	X'00'	ゼロ (E15 ルーチンが提供されていない)	
16 22	X'00'	ゼロ (E35 ルーチンが提供されていない)	
1A 26	X'00'	主記憶域値 (16 進数)	

図 35. 主記憶域オプションの指定 (24 ビット・パラメーター・リスト)

### 24 ビット・パラメーター・リストの例 2

図 36 は、E32 ルーチンから入力を行い、プログラムの障害時に ESTAE ルーチンに制御を渡したい場合に、組み合わせアプリケーションに使用する 24 ビット・パラメーター・リストの形式を示したものです。

(16 進)(10 進)	バイト 1	バイト 2	バイト 3 および 4
-2 -2	未使用		X'001C'
2 2	X'00'	MERGE ステートメントの開始アドレス	
6 6	X'00'	MERGE ステートメントの終了アドレス	
A 10	X'00'	RECORD ステートメントの開始アドレス	
E 14	X'00'	RECORD ステートメントの終了アドレス	
12 18	X'00'	E32 ルーチンのアドレス	
16 22	X'00'	ゼロ (E35 ルーチンが提供されていない)	
1A 26	X'04'	入力ファイルの数	
1E 30	X'FE'	ゼロ (作業域アドレスが提供されていない)	

図 36. E32 および ESTAE ルーチンの指定 (24 ビット・パラメーター・リスト)

### 24 ビット・パラメーター・リストの例 3

437 ページの図 37 は、主記憶域内の 24 ビット・パラメーター・リストの状態を示しています。汎用レジスター 1 には、位置 1000 にあるパラメーター・リストのアドレスを指し示すポインターが含まれています。このアドレスは、位置 1006 から始まるパラメーター・リストを指し示します。パラメーター・リストの最初の 2 バ

イトには、リストのバイト数 (10 進数の 36) を 16 進数で右寄せした値が入っています。

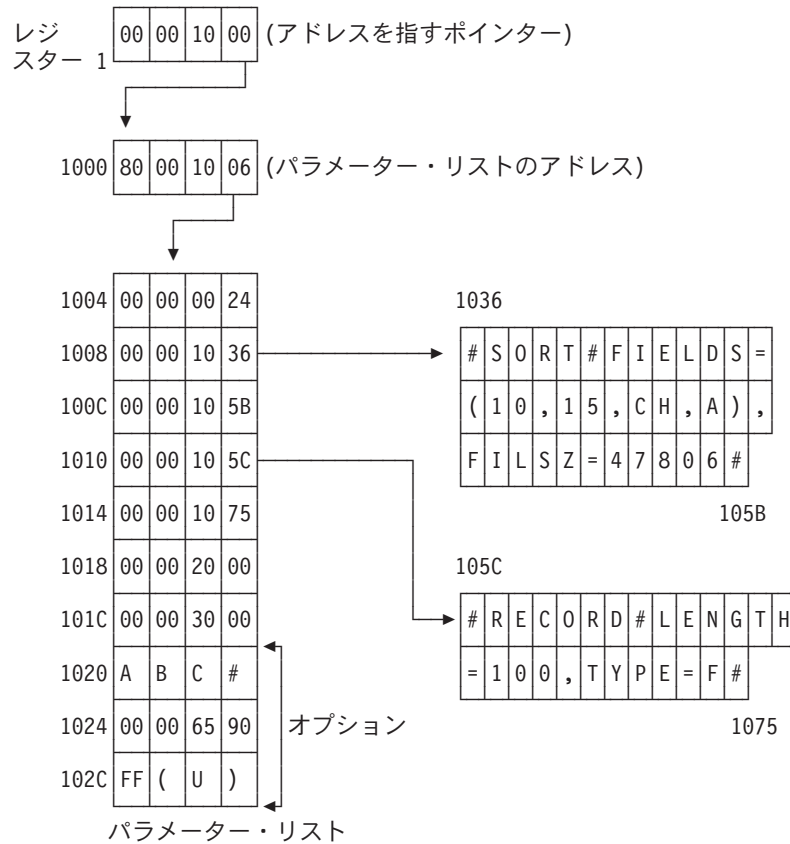


図 37. 主記憶域内の 24 ビット・パラメータ・リスト

パラメータ・リストの最初の 2 つのフルワードは、SORT 制御ステートメントの始め (位置 1036) と終わり (位置 105B) を指し示します。次の 2 つのフルワードは、RECORD ステートメントの始め (位置 105C) と終わり (位置 1075) を指し示します。

リストの 5 番目と 6 番目のフルワードには、E15 ユーザー出口 (記憶位置 2000) と E35 ユーザー出口 (記憶場所 3000) の入り口点アドレスが入っています。

リストのその次のフルワードには、標準の DD ステートメントの DD 名の 'SORT' を置き換えるための 4 文字が入っています。

リストのその次の 2 つのフルワードは、このアプリケーションとメッセージ・オプションの主記憶域値を指定します。

#### 24 ビット・パラメータ・リストの例 4

438 ページの図 38 の例は、図 37 の 24 ビット・パラメータ・リストに必要なパラメータとステートメント・イメージのコーディング方法をアセンブラ言語で示したものです。DFSORT へ制御を渡す方法も示しています。

## マクロ命令の書き方

```

        LA 1,PARLST          LOAD ADDR OF PARAM POINTER IN R1
        ATTACH EP=SORT      INVOKE SORT
        .
        .
PARLST DC X'80',AL3(ADLST)  POINTER FLAG/ADDRESS OF PARAM LIST
        .
        .
        CNOP 2,4           ALIGN TO CORRECT BOUNDARY
ADLST  DC AL2(LISTEND-LISTBEG)  PARAM LIST LENGTH
LISTBEG DC A(SORTA)          BEGINNING ADDRESS OF SORT STMT
        DC A(SORTZ)          END ADDRESS OF SORT STMT
        DC A(RECA)          BEGINNING ADDR OF RECORD STMT
        DC A(RECZ)          END ADDR OF RECORD STMT
        DC A(MOD1)          ADDR OF E15 RTN
        DC A(MOD2)          ADDR OF E35 RTN
        DC C'ABC#'          DDNAME CHARACTERS
        DC F'720000'        OPTIONAL MAIN STORAGE VALUE
        DC X'FF'            MESSAGE OPTION FLAG BYTE
        DC C'(U)'          MESSAGE OPTION
LISTEND EQU *
SORTA  DC C' SORT FIELDS=(10,15,CH,A),'  SORT CONTROL STMT
        DC C'FILSZ=E4780'      (CONTINUED)
SORTZ  DC C' '                DELIMITER
RECA   DC C' RECORD LENGTH=100,TYPE=F'  RECORD CONTROL STMT
RECZ   DC C' '                DELIMITER
        DS 0H
        USING *,15
MOD1   (routine for E15 user exit)
        .
        .
        USING *,15
MOD2   (routine for E35 user exit)

```

図 38. 24 ビット・パラメーター・リストのコーディング

### 拡張パラメーター・リストの例 1

439 ページの図 39 の例は、拡張パラメーター・リストを使用してパラメーターとステートメント・イメージのコーディングする方法、および DFSORT へ制御を渡す方法をアセンブラー言語で示したものです。

```

      .
      .
      .
*     LA   R1,PL1           SET ADDRESS OF PARAMETER LIST
                                TO BE PASSED TO SORT/MERGE
*     ST   R2,PL4           SET ADDRESS OF GETMAINED AREA
                                TO BE PASSED TO E15
*     LINK EP=SORT         INVOKE SORT/MERGE
      .
      .
      .
PL1   DC   A(CTLST)         ADDRESS OF CONTROL STATEMENTS
PL2   DC   A(E15)           ADDRESS OF E15 ROUTINE
PL3   DC   A(0)             NO E35 ROUTINE
PL4   DS   A                 USER EXIT ADDRESS CONSTANT
PL5   DC   F'-1'           INDICATE END OF LIST
CTLST DS   0H               CONTROL STATEMENTS AREA
      DC   AL2(CTL2-CTL1)   LENGTH OF CHARACTER STRING
CTL1  DC   C' SORT FIELDS=(4,5,CH,A) '
      DC   C' OPTION '
      DC   C' RESINV=2048,FILSZ=E25000,MSGDDN=MSGOUT '
      DC   C' OMIT COND=(5,8,EQ,13,8),FORMAT=FI '
      DC   C' RECORD TYPE=F,LENGTH=80 '
CTL2  EQU   *
OUT   DCB  DDNAME=SYSOUT,... MYSORT USES SYSOUT
E15   DS   0H               E15 ROUTINE
      .
      .
      .
      BR   R14               RETURN TO SORT/MERGE
* MAPPING OF PARAMETER LIST PASSED TO E15 FROM SORT/MERGE
SRTLST DS  A                 ADDRESS OF RECORD
GMA    DS  A                 ADDRESS OF AREA GETMAINED BY
*                                MYSORT
      .
      .
      .

```

図 39. 拡張パラメーター・リストのコーディング

---

## 動的呼び出しに関する制約事項

### 組み合わせに関する制約事項

DFSORT が PL/I のプログラムから呼び出される場合は、組み合わせアプリケーションは実行できません。

### コピーに関する制約事項

DFSORT が PL/I のプログラムから呼び出される場合は、コピー・アプリケーションは実行できません。

DFSORT を COBOL プログラムから呼び出す場合は、次のような制約があります。

- OS/VS COBOL を使用する場合、コピー・アプリケーションを実行できません。
- VS COBOL II またはそれ以降を使用する場合、OPTION COPY ステートメントを、COBOL II IGZSRTCD データ・セット、または DFSORT SORTCNTL あるいは DFSPARM データ・セットのいずれかに入れておくことができます。

## 動的呼び出しに関する制約事項

- COBOL SORT ステートメントの一部またはすべてに、COBOL II またはそれ以降の FASTSORT コンパイラー・オプションを使用すると、コピー・アプリケーションを実行できます。
- COBOL MERGE ステートメントを使用する場合は、コピー・アプリケーションを実行できません。

ユーザー出口に関する要件については、398 ページの『COBOL に関する要件 (コピー処理の場合)』を参照してください。

## 第 6 章 ICETOOL の使用

概要	443
ICETOOL/DFSORT の関係	443
ICETOOL JCL の要約	444
ICETOOL 演算子の要約	445
完全な ICETOOL の例	446
シンボルの使用	447
ICETOOL の呼び出し	447
ICETOOL の使用	448
各種統計の入手	448
データ・セットの複数のバージョン / 組み合わせの作成	449
ICETOOL のジョブ制御言語	451
JCL に関する制約事項	453
ICETOOL ステートメント	454
一般的なコーディングの規則	454
COPY 演算子	455
オペランドの説明	455
COPY の例	458
例 1	458
例 2	458
例 3	459
COUNT 演算子	459
オペランドの説明	460
COUNT の例	462
例 1	462
例 2	462
例 3	462
DEFAULTS 演算子	462
オペランドの説明	464
DEFAULTS の例	465
DISPLAY 演算子	467
簡単な報告書	468
調整済み報告書	469
セクションごとにまとめた報告書	470
オペランドの説明	471
DISPLAY の例	492
例 1	492
例 2	493
例 3	494
例 4	494
例 5	496
例 6	496
例 7	497
例 8	500
例 9	502
例 10	504
例 11	505
例 12	507
MODE 演算子	508

## ICETOOL の使用

オペランドの説明	509
MODE の例	509
OCCUR 演算子	510
簡単な報告書	511
調整済み報告書	512
オペランドの説明	514
OCCUR の例	521
例 1	521
例 2	521
例 3	522
例 4	523
例 5	523
RANGE 演算子	524
オペランドの説明	525
RANGE の例	526
SELECT 演算子	527
オペランドの説明	529
SELECT の例	531
例 1	531
例 2	531
例 3	532
例 4	532
例 5	533
例 6	533
SORT 演算子	534
オペランドの説明	535
SORT の例	536
例 1	537
例 2	537
例 3	538
STATS 演算子	539
オペランドの説明	539
STATS の例	540
UNIQUE 演算子	541
オペランドの説明	542
UNIQUE の例	542
VERIFY 演算子	543
オペランドの説明	544
VERIFY の例	544
プログラムからの ICETOOL の呼び出し	545
TOOLIN インターフェース	545
パラメーター・リスト・インターフェース	545
フィールドの説明	546
パラメーター・リスト・インターフェースの例	548
ICETOOL に関する注意事項および制約事項	551



## 概要

本章では、多目的 DFSORT ユーティリティーの ICETOOL について説明します。ICETOOL は DFSORT の機能を使用して、単一ジョブ・ステップで 1 つ以上のデータ・セットに複数の操作を実行します。これには次のようなものがあります。

- 分類済み、編集済み、または未編集の入力データ・セットの複数のコピーを作成する。
- 文字フィールドおよび数値フィールドの値に関する各種の基準、あるいは固有な値が発生する回数に基づいて、入力データ・セットのサブセットが入っている出力データ・セットを作成する。
- 異なるフィールド配列の入力データ・セットが入っている出力データ・セットを作成する。
- 文字フィールドおよび数値フィールドを、各種の簡単な、調整済みでセクション別にまとめた報告書形式で表すリスト・データ・セットを作成する。タイトル、日付、時刻、ページ番号、ヘッダー、ページ当たり行数、フィールド形式、および数値データ欄の合計、最大、最小、および平均の値を制御できる。
- 特定の数値フィールドに関する統計情報 (最小、最大、平均、合計、値のカウント、固有な値のカウントなど) を示すメッセージを印刷する。
- 無効な 10 進数値を識別するメッセージを印刷する。
- レコード数を示すメッセージを印刷する。
- レコード数に基づき RC=12 または RC=0 を設定する。
- DFSORT インストール・デフォルトを示すリスト・データ・セットを作成する。
- 特定の文字フィールドおよび数値フィールドの固有な値、およびそれぞれが発生する回数を示すリスト・データ・セットを、簡単な調整済み報告書の各種の形式で作成する。
- 重複する値、重複しない値、n 回発生する値、n 回より少なく発生する値、あるいは n 回より多く発生する値をもつレコードについて、リスト・データ・セットおよび出力データ・セットを作成する。
- 3 つの異なるモード (停止、続行、および走査) を使用して、演算子のグループに関するエラー検査およびエラー検出後の処置を制御する。

## ICETOOL/DFSORT の関係

ICETOOL は、DFSORT の機能を使用してユーザーが要求する操作を実行する、バッチ・フロントエンドのユーティリティーです。

ICETOOL は分類、コピー、統計、および報告の操作を実行する 12 の演算子から構成されます。ICETOOL により実行される操作の大部分は、JCL および演算子ステートメントだけを必要とします。一部の ICETOOL 操作では、DFSORT の機能を十分に活用するために、完全な DFSORT 制御ステートメント (たとえば、SORT、INCLUDE、OUTFIL など) を指定することを、ユーザーに要求あるいは許可します。

## 概要

ICETOOL は、各操作に必要な特定の DFSORT 制御ステートメントとオプション (たとえば、分類の場合は DYNALLOC) を用いて、DFSORT を自動的に呼び出します。

また、ICETOOL は、各操作の結果と検出されたエラーを記述するメッセージと戻りコードを作成します。通常、ユーザーは、ICETOOL 実行の結果作成された DFSORT メッセージを見る必要はありませんが、必要な場合は、これらのメッセージを個別のデータ・セットで利用できます。

ICETOOL は直接呼び出すことも、プログラムから呼び出すこともできます。ICETOOL を使用すると、データ・セットあるいは呼び出しプログラムにより渡されるパラメーター・リスト内に、演算子ステートメント (および注釈) を作成できます。パラメーター・リストに提供された各演算子について、ICETOOL は、その命令操作に関連するパラメーター・リストに情報を入れます。したがって、呼び出しプログラムは ICETOOL から提供された情報を使用できます。

## ICETOOL JCL の要約

ICETOOL で使用される JCL ステートメントを要約すると、下記のようになります。詳細については、451 ページの『ICETOOL のジョブ制御言語』を参照してください。また、453 ページの『JCL に関する制約事項』および 551 ページの『ICETOOL に関する注意事項および制約事項』も参照してください。

### **//JOB LIB DD**

ユーザーのプログラム・リンク・ライブラリーがまだシステムに知らされていない場合、そのライブラリーを定義します。

### **//STEPLIB DD**

//JOB LIB DD と同じ。

### **//TOOLMSG DD**

すべての操作作用の ICETOOL メッセージ・データ・セットを定義します。

### **//DFSMSG DD**

すべての操作作用の DFSORT メッセージ・データ・セットを定義します。

### **//SYMNAMES DD**

シンボルの処理に使用するステートメントを入れる、SYMNAMES データ・セットを定義します。

### **//SYMNOUT DD**

SYMNAMES ステートメントとシンボルのテーブルをリストするデータ・セットを定義します。

### **//TOOLIN DD**

ICETOOL 制御ステートメントを含んでいます。

### **//indd DD**

COPY、COUNT、DISPLAY、OCCUR、RANGE、SELECT、SORT、STATS、UNIQUE、または VERIFY 操作作用に入力データ・セットを定義します。

### **//outdd DD**

COPY、SELECT、または SORT 操作作用に出力データ・セットを定義します。

**//savedd DD**

SELECT 操作用に出力データ・セットを定義します。

**//listdd DD**

DEFAULTS、DISPLAY、または OCCUR 操作用にリスト・データ・セットを定義します。

**//xxxxCNTL DD**

COPY、COUNT、または SORT 操作用の DFSORT 制御ステートメントを含んでいます。

## ICETOOL 演算子の要約

ICETOOL には、各種の機能を実行するために使用される 12 の演算子があります。これらの演算子の機能を要約すると、下記ようになります。詳細については、454 ページの『ICETOOL ステートメント』を参照してください。さらにここでは、パラメーター・リストを使用して ICETOOL ステートメントを提供する呼び出しプログラムについて、各演算子に関する情報を示してあります。詳細については、545 ページの『パラメーター・リスト・インターフェース』を参照してください。

**COPY**

データ・セットを 1 つ以上の出力データ・セットにコピーします。

**COUNT**

データ・セット内のレコードの数を示すメッセージを印刷します。COUNT はまた、データ・セットの複数のレコードで一致する基準に基づき、RC=12 または RC=0 を設定します。

**DEFAULTS**

別個のリスト・データ・セットに DFSORT インストール・デフォルトを印刷します。

**DISPLAY**

別個のリスト・データ・セット内の指定した数値フィールドまたは文字フィールドの値または文字を印刷します。簡単な、調整済み、またはセクション別にまとめられた報告書を作成できます。

**MODE**

演算子のグループに対してセットまたはリセットできる 3 つのモードが使用できます。

- STOP (停止) モード (デフォルト) は、エラーが見つかった場合に後続の操作を停止します。
- CONTINUE (続行) モードは、エラーが見つかった場合でも後続の操作を続行します。
- SCAN (走査) モードは、実際に操作を行わずに ICETOOL ステートメントの検査を行うことができます。

**OCCUR**

指定された数値フィールドまたは文字フィールドの各固有値と、それが発生した回数を別個のデータ・セット内に印刷します。簡単な報告書または調整済み報告書を作成できます。印刷される値を、その値のカウントが指定基準を満たした値だけに限定できます (たとえば、重複した値のみ、あるいは重複しない値のみ)。

### **RANGE**

データ・セット内の指定数値フィールドについて、指定した範囲内の値のカウントを含むメッセージを印刷します。

### **SELECT**

指定した数値フィールドまたは文字フィールドの値が発生した回数についての、指定基準（たとえば、重複する値のみ、あるいは重複しない値のみ）に基づいて、データ・セットから出力データ・セットに組み込まれるレコードを選択します。選択されないレコードは、別の出力データ・セットに保管できます。

### **SORT**

データ・セットを 1 つ以上の出力データ・セットへ分類します。

### **STATS**

データ・セット内の指定数値フィールドについて、最小、最大、平均、および合計の値を含むメッセージを印刷します。

### **UNIQUE**

指定の数値フィールドまたは文字フィールドについて、固有な値の数を含むメッセージを印刷します。

### **VERIFY**

データ・セット内の指定の 10 進数フィールドを検査し、各フィールドごとに見つかった無効な値を示すメッセージを印刷します。

## 完全な ICETOOL の例

682 ページの『ICETOOL の例』には、すべての必須の JCL および制御ステートメントを用いた完全な ICETOOL のジョブの例が示されています。下記の例は、簡単な ICETOOL ジョブの JCL および制御ステートメントを示しています。

表 61. 簡単な ICETOOL ジョブ

```

//EXAMP JOB A402,PROGRAMMER
//RUNIT EXEC PGM=ICETOOL,REGION=1024K
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//TOOLIN DD *
* Show installation (ICEMAC) defaults
DEFAULTS LIST(SHOWDEF)
* Create three copies of a data set
COPY FROM(IN1) TO(OUT1,OUT2,OUT3)
* Print a report
DISPLAY FROM(IN2) LIST(REPORT) DATE TITLE('Monthly Report') PAGE -
  HEADER('Location') ON(1,25,CH) -
  HEADER('Revenue') ON(23,10,FS) -
  HEADER('Profit') ON(45,10,FS) -
  TOTAL('Totals') AVERAGE('Averages') BLANK
* Select all records with duplicate (non-unique) keys
SELECT FROM(IN2) TO(DUPKEYS) ON(1,25,CH) ALLDUPS -
* Save all records with non-duplicate (unique) keys
DISCARD (UNQKEYS)
/*
//SHOWDEF DD SYSOUT=A
//IN1 DD DSN=FLY.INPUT1,DISP=SHR
//IN2 DD DSN=FLY.INPUT2,DISP=SHR
//OUT1 DD DSN=FLY.NEW,DISP=OLD
//OUT2 DD DSN=FLY.BU1,DISP=OLD
//OUT3 DD DSN=FLY.BU2,DISP=OLD
//DUPKEYS DD DSN=FLY.DUPS,DISP=OLD
//UNQKEYS DD DSN=FLY.UNQS,DISP=OLD
//REPORT DD SYSOUT=A

```

## シンボルの使用

COUNT、DISPLAY、OCCUR、RANGE、SELECT、STATS、UNIQUE および VERIFY の各 ICETOOL 演算子では、任意のフィールドまたは定数にシンボルを定義して使用できます。ICETOOL の実行用に指定する DFSORT 制御ステートメントに、シンボルを使用することもできます。シンボルを使用すると、さまざまなレコード・レイアウトに関連した情報を表すシンボルの集合を作成して、繰り返し使用（つまり、マッピング）できます。詳しくは、555 ページの『第 7 章 フィールドと定数のシンボルの使用』を参照してください。

## ICETOOL の呼び出し

以下の 3 つの方法で、ICETOOL を呼び出すことができます。

- TOOLIN インターフェースを使用して直接呼び出す（つまり、プログラムから呼び出すのではない）
- TOOLIN インターフェースを使用してプログラムから呼び出す
- パラメーター・リスト・インターフェースを使用してプログラムから呼び出す

TOOLIN インターフェースを使用して、ユーザーは、TOOLIN DD ステートメントで定義されたデータ・セット内に ICETOOL ステートメントを提供します。ICETOOL は、TOOLMSG DD ステートメントで定義されたデータ・セット内にメッセージを印刷します。

ユーザー・プログラムは、パラメーター・リスト・インターフェースを使用して、パラメーター・リストに ICETOOL ステートメントを提供します。ICETOOL は、

## 概要

TOOLMSG DD ステートメントで定義されたデータ・セット内にメッセージを印刷し、また、ユーザー・プログラムで使用するために情報をパラメーター・リスト内に入れます。

## ICETOOL の使用

12 の ICETOOL 演算子をさまざまに組み合わせることにより、多くの複雑なタスクを実行するアプリケーションを簡単に作成できます。以下の 2 つの小さな例は、ICETOOL を用いて実行できるいくつかの操作を示したものです。

### 各種統計の入手

表 62. 各種統計の入手

```
MODE STOP
VERIFY FROM(DATA1) ON(22,7,PD)
DISPLAY FROM(DATA1) LIST(SALARIES) -
  TITLE('Employee Salaries') DATE TIME -
  HEADER('Employee Name') HEADER('Salary') -
  ON(1,20,CH) ON(22,7,PD) BLANK -
  AVERAGE('Average Salary')
STATS FROM(DATA1) ON(22,7,PD)
RANGE FROM(DATA1) ON(22,7,PD) LOWER(20000)
RANGE FROM(DATA1) ON(22,7,PD) HIGHER(19999) LOWER(40000)
RANGE FROM(DATA1) ON(22,7,PD) HIGHER(40000)
OCCUR FROM(DATA1) LIST(SALARIES) -
  TITLE('Employees Receiving Each Salary') DATE TIME -
  HEADER('Salary') HEADER('Employee Count') -
  ON(22,7,PD) ON(VALCNT) BLANK
```

示されているデータ・セットに対して、次の DD 名をもつ DD ステートメントを指定するものとします。

#### DATA1

各従業員の名前、給与、部門、勤務地などを含むデータ・セット。名前フィールドは文字形式で 1 桁目～ 20 桁目にあり、給与フィールドはパック 10 進数形式で 22 桁目～ 28 桁目にあります。

#### SALARIES

SYSOUT データ・セット

表 62 の ICETOOL 演算子を使用して、次のことを行うことができます。

#### MODE STOP

演算子の 1 つを処理しているときにエラーが見つかった場合、後続の演算子は処理されません (すなわち、各演算子が処理されるかどうかは、その前の演算子が成功したかどうかによります)。

#### VERIFY

パック 10 進数フィールドに無効な値があると、それを示すエラー・メッセージを TOOLMSG データ・セットに印刷します。

#### DISPLAY

各従業員の名前と給与、および全従業員の平均給与の報告書を SALARIES リスト・データ・セットに印刷します。

**STATS**

個々の給与の最小、最大、平均、および合計を示すメッセージを、  
TOOLMSG データ・セットに印刷します。

**RANGE**

3 つの RANGE 演算子は、\$19,999 以下、\$20,000 ～ \$39,999、および  
\$40,000 以上の範囲にある給与の数を示すメッセージを TOOLMSG デー  
タ・セットに印刷します。

**OCCUR**

固有な各給与、およびそれを受け取る従業員の数の報告書を SALARIES リ  
スト・データ・セットに印刷します。

**データ・セットの複数のバージョン / 組み合わせの作成**

表 63. データ・セットの複数のバージョン / 組み合わせの作成

```
* GROUP 1
MODE CONTINUE
COPY FROM(DATA1) TO(DATA2)
COPY FROM(MSTR1) TO(MSTR2)
SELECT FROM(DATA1) TO(SMALLDPT) ON(30,4,CH) LOWER(10)
UNIQUE FROM(MSTR1) ON(30,4,CH)
* GROUP 2
MODE STOP
COPY FROM(DATA1) TO(TEMP1) USING(NEW1)
COPY FROM(DATA1) TO(TEMP2) USING(NEW2)
COPY FROM(DATA1) TO(TEMP3) USING(NEW3)
SORT FROM(CONCAT) TO(FINALD,FINALP) USING(FINL)
```

示されているデータ・セットに対して、次の DD 名をもつ DD ステートメントを  
指定するものとします。

**DATA1**

各従業員の名前、給与、部門、勤務地などを含むデータ・セット。部門フィール  
ドは文字形式で、30 桁目～ 33 桁目にあります。

**MSTR1**

各従業員の名前と部門だけを含むマスター・データ・セット。部門フィールドは  
文字形式で、30 桁目～ 33 桁目にあります。

**DATA2、MSTR2、および SMALLDPT**

永続データ・セット。

**NEW1CNTL**

DFSORT 制御ステートメントを含むデータ・セット。この制御ステートメント  
は部門 X100 の従業員を組み込み (INCLUDE)、MSTR1 の形式と一致するよう  
にレコードを変更します。

**NEW2CNTL**

NEW1CNTL と同じですが、部門は X200 です。

**NEW3CNTL**

NEW1CNTL と同じですが、部門は X300 です。

**TEMP1、TEMP2、および TEMP3**

一時データ・セット。

**FINLCNTL**

DFSORT 制御ステートメントを含むデータ・セット。この制御ステートメントはデータ・セットを部門および従業員名により分類します。

**CONCAT**

TEMP1、TEMP2、TEMP3、および MSTR1 データ・セットを連結したものの。

**FINALD**

永続データ・セット。

**FINALP**

SYSOUT データ・セット

表 63 の ICETOOL 演算子を使用して、次のことを行うことができます。

**MODE CONTINUE**

グループ 1 の演算子のいずれかを処理しているときにエラーが見つかった場合でも、グループ 1 の後続の演算子は引き続き処理されます。つまり、その前のグループ 1 の演算子が成功したかどうかにかかわらず、グループ 1 の次の演算子は処理されます。

**COPY** 2 つの COPY 演算子は、DATA1 および MSTR1 のバックアップ・コピーを作成します。

**SELECT**

従業員が 9 人以下の部門の各従業員の名前、給与、部門、勤務地などを含む永続出力データ・セットを作成します。

**UNIQUE**

固有の部門の数を示すメッセージを TOOLMSG データ・セットに印刷します。

**MODE STOP**

グループ 2 の演算子のいずれかを処理しているときにエラーが見つかった場合、グループ 2 の後続の演算子は処理されません。つまり、グループ内の前の演算子が成功したかどうかにより、グループ 2 の次の演算子は処理されるかどうかが決まります。

**COPY** 3 つの COPY 演算子は、各部門の従業員について名前および部門だけを含む出力データ・セットを作成します。USING(yyyy) オペランドで要求された DD 名が yyyyCNTL であることに注意してください。たとえば、USING(NEW1) は DD 名 NEW1CNTL を要求します。

**SORT** COPY 演算子により作成された 3 つの出力データ・セットを、名前 / 部門のマスター・データ・セットとともに分類し、そこで分類されたレコードを含む永続データ・セットおよび SYSOUT データ・セットを作成します。

これらの例を結合して、単一の ICETOOL ジョブ・ステップにすることができます。



## ICETOOL のジョブ制御言語

以下に、ICETOOL のジョブ制御言語 (JCL) ステートメントの概要を示した後、各 ICETOOL DD ステートメント、および予約された DD ステートメントと DD 名の使用について説明します。

表 64. ICETOOL の JCL ステートメント

```
//EXAMPL JOB ...
/* ICETOOL CAN BE CALLED DIRECTLY OR FROM A PROGRAM
//STEP EXEC PGM=ICETOOL (or PGM=program_name)
/* THE FOLLOWING DD STATEMENTS ARE ALWAYS REQUIRED
//TOOLMSG DD SYSOUT=A (or DSN=...)
//DFSMSG DD SYSOUT=A
/* THE FOLLOWING DD STATEMENTS ARE USED FOR SYMBOL PROCESSING
/* SYMNames DD ...
/* SYMNOUT DD SYSOUT=A (OR DSN=...)
/* THE TOOLIN DD STATEMENT IS ONLY REQUIRED IF THE TOOLIN INTERFACE
/* IS USED.
//TOOLIN DD *
    ICETOOL statements
/*
/* THE FOLLOWING DD STATEMENTS ARE ONLY REQUIRED IF SPECIFIED IN
/* ICETOOL STATEMENTS.
//indd DD ...
.
.
.
//outdd DD ...
.
.
.
//listdd DD SYSOUT=A (or DSN=...)
.
.
.
//xxxxCNTL DD *
    DFSORT control statements
/*
.
.
.
```

### TOOLMSG DD ステートメント

すべての操作の ICETOOL メッセージ・データ・セットを定義します。ICETOOL メッセージおよびステートメントはこのデータ・セット内にあります。TOOLMSG データ・セットについて、ICETOOL は RECFM=FBA、LRECL=121、および指定の BLKSIZE を使用します。指定した BLKSIZE が 121 の倍数でない場合は、ICETOOL は BLKSIZE=121 を使用します。BLKSIZE を指定しないと、ICETOOL は SDBMSG インストール・オプションで指定されているブロック・サイズを選択します (*DFSORT* 導入とカスタマイズ リリース 14 を参照してください)。

## ICETOOL のジョブ制御言語

TOOLMSG DD ステートメントは、必ず 使用してください。

### DFSMSG DD ステートメント

すべての操作用の DFSORT メッセージ・データ・セットを定義します。

DFSORT に対するすべての ICETOOL 呼び出しからの DFSORT メッセージおよび制御ステートメントが、このデータ・セット内にあります。詳細については、70 ページの『システム DD ステートメント』の SYSOUT の項を参照してください。

DFSMSG DD ステートメントは、必ず 使用してください。

注: DFSMSG には SYSOUT データ・セットを使用します。DFSMSG を一時データ・セットまたは永続データ・セットとして定義した場合、MOD の後処置を使用して新しいデータ・セットを割り振らない限り、DFSORT に対する最後の呼び出しからの DFSORT メッセージだけが示されます。

### //SYMNAMES DD

シンボルの処理に使用するステートメントを入れる、SYMNAMES データ・セットを定義します。詳しくは、555 ページの『第 7 章 フィールドと定数のシンボルの使用』を参照してください。

### //SYMNOUT DD

SYMNAMES ステートメントとシンボルのテーブルをリストするデータ・セットを定義します。詳しくは、555 ページの『第 7 章 フィールドと定数のシンボルの使用』を参照してください。

### TOOLIN DD ステートメント

ICETOOL ステートメント・データ・セットを定義します。このデータ・セットは RECFM=F または RECFM=FB で、かつ LRECL=80 の属性をもっている必要があります。

TOOLIN インターフェイスが使用される場合は、TOOLIN DD ステートメントが必要です。パラメーター・リスト・インターフェイスが使用される場合は、TOOLIN DD ステートメントは必要がなく、あっても無視されます。

### indd DD ステートメント

操作の入力データ・セットを定義します。詳細については、75 ページの『SORTIN DD ステートメント』を参照してください。さらに ICETOOL では、このデータの LRECL は 4 以上の長さでなければならないという制約があります。

各 FROM オペランドに指定される固有な各 indd に対して、1 つの indd DD ステートメントが使用されます。

### outdd DD ステートメント

COPY、SELECT、または SORT 操作用に出力データ・セットを定義します。詳細については、82 ページの『SORTOUT および OUTFIL DD ステートメント』を参照してください。

各 TO オペランドに指定される固有な outdd 名のそれぞれに対して、1 つの outdd DD ステートメントが使用されます。

### savedd DD ステートメント

SELECT 操作用に出力データ・セットを定義します。詳細については、82 ページの『SORTOUT および OUTFIL DD ステートメント』を参照してください。

各 DISCARD オペランドに指定される固有な savedd 名のそれぞれに対して、1 つの savedd DD ステートメントが使用されます。

#### listdd DD ステートメント

DEFAULTS、DISPLAY、または OCCUR 操作のためのリスト・データ・セットを定義します。各 listdd データ・セットについて、ICETOOL は、RECFM=FBA、LRECL=121 (DEFAULTS としての) または WIDTH オペランドで指定された LRECL を使用し、または WIDTH が指定されていない場合は、必要に応じて、計算された (DISPLAY および OCCUR) LRECL を使用します。また、指定されたブロック・サイズを使用します。指定した BLKSIZE が LRECL の倍数でない場合は、ICETOOL は BLKSIZE=LRECL を使用します。BLKSIZE を指定しないと、ICETOOL は SDBMSG インストール・オプションで指定されているブロック・サイズを選択します (DFSORT 導入とカスタマイズ リリース 14 を参照してください)。

各 LIST オペランドに指定される固有な各 listdd 名ごとに、1 つの listdd DD ステートメントが使用されます。

#### xxxxCNTL DD ステートメント

SORT、COPY、または COUNT 操作のための DFSORT 制御ステートメント・データ・セットを定義します。詳細については、84 ページの『SORTCNTL DD ステートメント』を参照してください。

各 USING オペランドに指定される固有な各 xxxx ごとに、1 つの xxxxCNTL DD ステートメントが使用されます。

## JCL に関する制約事項

ICETOOL および ICETOOL オペランドの DFSORT 用に予約されている DD 名 (FROM、TO、LIST、DISCARD) を使用できません。通常、ICETOOL を使用するときには、DFSORT 用に予約されている DD 名をもつ DD ステートメントも指定できません。指定すると、結果は予測できません。特に、次のことに注意してください。

- SORTCNTL を ICETOOL 演算子で DD 名として使用したり、あるいは DD 名として提供できません。xxxxCNTL DD ステートメントは、USING(xxxx) オペランドを指定する場合のみ使用されます。xxxx は、形式 xxxxCNTL を構成したときに有効な DD 名となるような、4 文字である必要があります。xxxx に SYSx は指定できません。
- SYSin、SORTCNTL、SORTIN、SORTOUT、SORTINnn、および xxxxINnn (xxxx は USING オペランドに指定される) は、ICETOOL 演算子で DD 名として使用したり、あるいは DD ステートメントとして使用できません。
- TOOLMSG、DFSMSG、SYMNAMES、SYMNOUT、TOOLIN、SYSUDUMP、および SYSABEND は、ICETOOL 演算子で DD 名として使用できません。
- 通常、xxxxWKdd DD 名は、ICETOOL 演算子の DD 名として使用するべきではありません。ただし、OCCUR、SELECT および UNIQUE といった ICETOOL 演算子の作業データ・セットの、動的割り振りを指定変更する場合、SORTWKdd DD ステートメントを ICETOOL 演算子の DD 名として使用できます。ICETOOL 演算子である SORT の作業データ・セットの、動的割り振りを指定変更する場合、xxxxWKdd DD ステートメント (xxxx は、USING オペランドで指定される) を ICETOOL 演算子の DD 名として使用できます。

## JCL に関する制約事項

- DFSPARM (または ICEMAC オプションの PARMDDN に指定された DD 名) は、ICETOOL 演算子で DD 名として使用できません。DFSPARM は、すべての演算子の DFSORT オプションを指定変更する場合のみ、DD ステートメントとして使用すべきものです。詳細については、85 ページの『DFSPARM DD ステートメント』を参照してください。
- xxxxCNTL データ・セットの OUTFIL ステートメントが FILES=dd を指定するとき、DD 名として xxxxOFdd (ここで、xxxx は USING オペランドに指定される) が必要です。この要件を避けるためには、FILES=dd オペランドではなく、FNAMES=ddname オペランドを OUTFIL ステートメントに使用します。また、指定された DD 名として DD ステートメントを組み込みます。FNAMES オペランドの詳細については、238 ページの『OUTFIL 制御ステートメント』を参照してください。

---

## ICETOOL ステートメント

演算子ステートメントを使用して、各操作を ICETOOL に対して記述する必要があります。さらに、ICETOOL では、注釈ステートメントおよびブランク・ステートメントを使用することもできます。下記の ICETOOL ステートメントのコーディングに関する一般的な規則に続けて、各演算子について記述します。

### 一般的なコーディングの規則

すべての ICETOOL 演算子ステートメントの一般的な形式は次のとおりです。

```
OPERATOR オペランド ... オペランド
```

ここで、各オペランドは、KEYWORD (パラメーター、パラメーター...) または単に KEYWORD で構成されます。任意の数の演算子を任意の順序で指定できます。

演算子ステートメントについては、次の規則が適用されます。

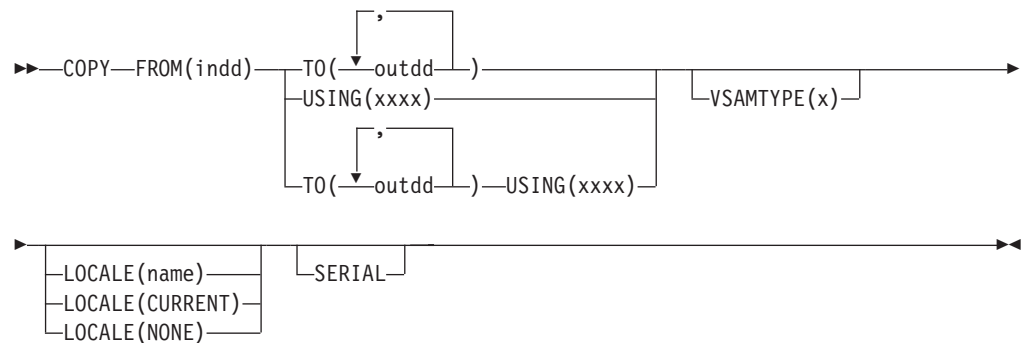
- 演算子とオペランドは大文字の EBCDIC である必要があります。
- 演算子は最初に指定します。
- 演算子と最初のオペランドの間には 1 つのブランクが必要です。
- オペランドとオペランドの間には 1 つのブランクが必要です。
- 演算子または任意のオペランドの前後にはいくつでもブランクを指定できますが、引用符付き文字ストリング以外の場所ではブランクを指定できません。
- 括弧が示されている個所では括弧を使用します。コンマが示されている個所では、コンマまたはセミコロンを使用します。
- オペランドは任意の順序で指定できます。
- 1 ~ 72 桁目は走査されますが、73 ~ 80 桁目は無視されます。
- 演算子の後または任意のオペランドの後のハイフン (-) は、継続を示します。次のオペランドは次の行に指定します。たとえば、下記のようにします。

```
SORT FROM(INDD) -  
      USING(ABCD) -  
      TO(OUTPUT1,OUTPUT2,OUTPUT3)
```

ハイフンの後に指定された文字はすべて無視されます。各オペランドは、1 つの行の上ですべて指定します。

1 桁目がアスタリスク (\*) のステートメントは、注釈ステートメントとして扱われます。注釈ステートメントは他の ICETOOL ステートメントとともに印刷されますが、印刷以外の処理は行われません。1 ~ 72 桁目がブランクのステートメントは、ブランク・ステートメントとして扱われます。ICETOOL は適切な個所でブランク行を印刷しますので、ブランク・ステートメントは無視されます。

## COPY 演算子



入力データ・セットを 1 つ以上の出力データ・セットにコピーします。

indd データ・セットを最初の outdd データ・セットにコピーするために、DFSORT が呼び出されます。USING(yyyy) が指定されている場合は、yyyyCNTL 中の DFSORT 制御ステートメントが使用されます。yyyyCNTL データ・セット内の DFSORT 制御ステートメントおよびオプションを使用して、入力レコードのサブセットのコピー (INCLUDE または OMIT ステートメント、SKIPREC および STOPAFT オプション、OUTFIL INCLUDE、OMIT、STARTREC、ENDREC、および SPLIT オペランド、ユーザー出口ルーチン)、出力用のレコードを再フォーマット設定 (INREC および OUTREC ステートメント、OUTFIL OUTREC オペランド、ユーザー出口ルーチン) などを行うことができます。

yyyyCNTL データ・セットの中で INCLUDE または OMIT ステートメントあるいは OUTFIL INCLUDE または OMIT オペランドが指定されている場合、活動ロケールの照合規則が、INCLUDE および OMIT 処理に影響を与えます (130 ページの『INCLUDE 制御ステートメント』における『文化環境に関する考慮事項』の説明を参照)。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

この操作で DFSORT により読み込まれる入力データ・セットの DD 名を指定します。indd DD ステートメントが必要で、入力データ・セットを定義します。このデータ・セットは DFSORT の SORTIN データ・セットに関する規則に従ったものである必要があります。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### TO(outdd,...)

この操作で DFSORT により書き出される出力データ・セットの DD 名を指定します。1 ~ 10 個の outdd 名を指定できます。指定する各 outdd 名ごとに 1 つの outdd DD ステートメントが必要です。1 つの outdd データ・セットが指定された場合は、SORTOUT 処理を使用して、outdd データ・セットに indd データ・セットをコピーするために、DFSORT が一度呼び出されます。outdd データ・セットは、DFSORT の SORTOUT データ・セットに関する規則に従ったものである必要があります。複数の outdd データ・セットが指定され、SERIAL が指定されない場合は、OUTFIL 処理を使用して、outdd データ・セットに indd データ・セットをコピーするために、DFSORT が一度呼び出されます。outdd データ・セットは、DFSORT の OUTFIL データ・セットに関する規則に従ったものである必要があります。

TO および USING の両方を指定できます。USING が指定されない場合は、TO を指定する必要があります。TO が指定されない場合は、USING を指定する必要があります。

FROM オペランドで指定された DD 名を、TO オペランドで指定できません。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### USING(yyyy)

この操作で DFSORT により使用される制御ステートメント・データ・セットの DD 名の最初の 4 文字を指定します。yyyy は、形式 yyyyCNTL を構成したときに有効な DD 名となるような、4 文字である必要があります。yyyy に SYSx は指定できません。

USING を指定する場合、1 つの yyyyCNTL DD ステートメントが必要で、その中の制御ステートメントは DFSORT の SORTCNTL データ・セットに関する規則に従ったものである必要があります。

TO および USING の両方を指定できます。USING が指定されない場合は、TO を指定する必要があります。TO が指定されない場合は、USING を指定する必要があります。また、yyyyCNTL データ・セットには、1 つ以上の OUTFIL ステートメントか、あるいはすべてのレコードを後処理する E35 ルーチンのための MODS ステートメントのいずれかが含まれます。他のステートメントはオプションです。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### VSAMTYPE(x)

VSAM 入力データ・セットのレコード・タイプを指定します。x は、固定長レコード処理の場合は F、可変長レコード処理の場合は V を指定します。

VSAMTYPE(x) を指定すると、ICETOOL は RECORD TYPE=x 制御ステートメントを DFSORT に渡します。(yyyyCNTL データ・セットで RECORD TYPE=x ステートメントを指定すると、それが、ICETOOL により渡されたものを指定変更します。)

VSAM 入力のレコード・タイプ処理に関する完全情報は、340 ページの『RECORD 制御ステートメント』を参照してください。

#### LOCALE(name)

ロケール処理を使用することを指定し、DFSORT 処理中、活動状態にしておくロケールの名前を指定します。LOCALE(name) は、LOCALE インストール・オプションを指定変更するのに使用できます。LOCALE(name) の詳細については、181 ページの『OPTION 制御ステートメント』の LOCALE オペランドについての説明を参照してください。

#### LOCALE(CURRENT)

ロケール処理が使用されることを指定します。DFSORT に入る際の現行活動ロケールは、DFSORT 処理の間は活動ロケールのままです。LOCALE(CURRENT) は、LOCALE インストール・オプションを指定変更するのに使用できます。LOCALE(CURRENT) の詳細については、181 ページの『OPTION 制御ステートメント』の LOCALE オペランドについての説明を参照してください。

#### LOCALE(NONE)

ロケール処理が使用されないことを指定します。DFSORT は、ユーザー・データを照合し比較するために定義されたコード・ページの 2 進数のエンコードを使用します。LOCALE(NONE) は、LOCALE インストール・オプションを指定変更するのに使用できます。LOCALE(NONE) の詳細については、181 ページの『OPTION 制御ステートメント』の LOCALE オペランドについての説明を参照してください。

#### SERIAL

複数の outdd データ・セットが指定された場合に、OUTFIL 処理を使用しないことを指定します。DFSORT は複数呼び出され、SORTOUT 処理を使用しません。outdd データ・セットは DFSORT の SORTOUT データ・セットに関する規則に従っている必要があります。SERIAL の使用はお勧めしません。OUTFIL 処理の代わりに逐次処理 (つまり、DFSORT の複数呼び出し) を使用すると、パフォーマンスが低下し、下記のような制約を課す場合があります。単一の outdd データ・セットが指定された場合、SERIAL は無視されます。

USING(yyyy) が指定されている場合は、DFSORT を呼び出し、yyyyCNTL 内にある DFSORT 制御ステートメントを使用して、indd データ・セットを最初の出力データ・セットにコピーします。最初のコピーが成功すると、DFSORT を必要な回数呼び出して、最初の outdd データ・セットを 2 番目以降の outdd データ・セットにコピーします。したがって、最大の効率を得るために、DASD およびテープの両方が含まれる outdd データ・セットのリストでは DASD データ・セットを最初に使用してください。複数の outdd データ・セットが指定されている場合、DFSORT は、outdd データ・セットが書き込まれた後、最初の outdd データ・セットを読み取って、それを他の outdd データ・セットにコピーできる必要があります。次の理由から、SYSOUT または DUMMY データ・セットを outdd データ・セット・リストの最初のデータ・セットとして使用しないでください。

- 最初のデータ・セットが SYSOUT の場合、DFSORT は、その SYSOUT データ・セットを 2 番目の outdd データ・セットにコピーしようとする異常終了します。

## COPY 演算子

- 最初のデータ・セットが DUMMY の場合、DFSORT は、空の DUMMY データ・セットを他の outdd データ・セットにコピーします。その結果、すべての outdd データ・セットは空になります。

## COPY の例

下記の例にある COPY 演算子は、すべて単一の ICETOOL ジョブ・ステップに組み込むことができますが、わかりやすくするために、別々に示して説明します。

### 例 1

```
* Method 1
COPY FROM(MASTER) TO(PRINT,TAPE,DASD)
```

```
* Method 2
COPY FROM(MASTER) TO(DASD,TAPE,PRINT) SERIAL
```

この例では、複数の出力データ・セットを作成する 2 つの異なる方式を示しています。

方式 1 は、DFSORT の呼び出しを 1 回、入力データ・セットに対して 1 回のパスを必要とし、出力データ・セットは任意の順序で指定できます。COPY 演算子は、OUTFIL 処理を使用して、MASTER データ・セットからのすべてのレコードを PRINT (SYSOUT)、TAPE、および DASD データ・セットにコピーします。

方式 2 は、DFSORT の呼び出しを 3 回、入力データ・セットに対して 3 回のパスを必要とし、SYSOUT データ・セットは最初の TO データ・セットであってはならないという制限が課されます。COPY 演算子は、MASTER データ・セットからのすべてのレコードを DASD データ・セットにコピーし、次に、その DASD データ・セットを TAPE および PRINT (SYSOUT) データ・セットにコピーします。最初の TO データ・セットは 3 回 (書き込み、読み取り、読み取り) 処理されるので、DASD データ・セットを最初に入れたほうが TAPE データ・セットを最初に入れるよりもっと効率的です。PRINT を、TO リストの中で最初にセットできません。SYSOUT データ・セットは読み取れないためです。

### 例 2

```
* Method 1
COPY FROM(IN) TO(DEPT1) USING(DPT1)
COPY FROM(IN) TO(DEPT2) USING(DPT2)
COPY FROM(IN) TO(DEPT3) USING(DPT3)
```

```
* Method 2
COPY FROM(IN) USING(ALL3)
```

この例は、入力データ・セットのサブセットを作成するための 2 つの異なる方式を示しています。次のように仮定します。

- DPTICNTL データ・セットには、次のものが含まれています。  
INCLUDE COND=(5,3,CH,EQ,C'D01')



- DPT2CNTL データ・セットには、次のものが含まれています。  
INCLUDE COND=(5,3,CH,EQ,C'D02')
- DPT3CNTL データ・セットには、次のものが含まれています。  
INCLUDE COND=(5,3,CH,EQ,C'D03')
- ALL3CNTL データ・セットには、次のものが含まれています。  
OUTFIL FNAMES=DEPT1,INCLUDE=(5,3,CH,EQ,C'D01')  
OUTFIL FNAMES=DEPT2,INCLUDE=(5,3,CH,EQ,C'D02')  
OUTFIL FNAMES=DEPT3,INCLUDE=(5,3,CH,EQ,C'D03')

方式 1 は、DFSORT の 3 回の呼び出しと入力データ・セットの 3 回のパスを必要とします。

- 最初の COPY 演算子は、5 ～ 7 桁目に D01 を含む IN データ・セットのレコードを DEPT1 データ・セットにコピーします。
- 2 番目の演算子は、5 ～ 7 桁目に D02 を含む IN データ・セットのレコードを DEPT2 データ・セットにコピーします。
- 3 番目の演算子は、5 ～ 7 桁目に D03 を含む IN データ・セットのレコードを DEPT3 データ・セットにコピーします。

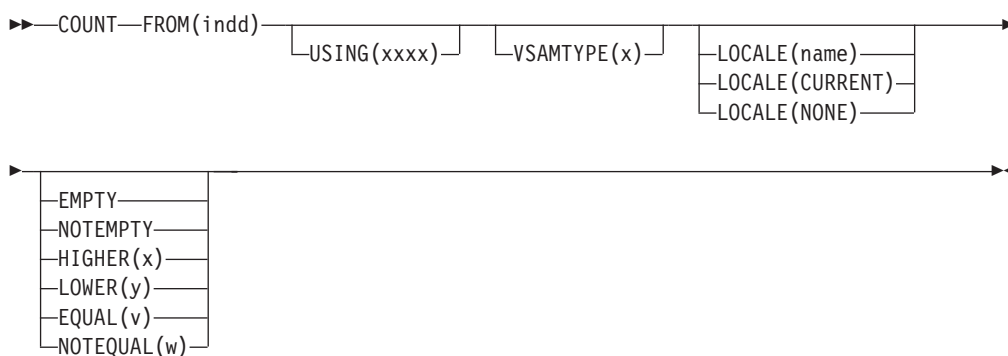
方式 2 は、方式 1 と同じ結果を作り出しますが、TO オペランドの代わりに OUTFIL ステートメントを使用するため、DFSORT の 1 回の呼び出し、入力データ・セットの 1 回のパスで済みます。

### 例 3

```
COPY FROM(VSAMIN) TO(VSAMOUT) VSAMTYPE(V)
```

COPY 演算子は、VSAMIN データ・セットからのすべてのレコードを VSAMOUT データ・セットにコピーします。VSAM レコードは可変長として扱われます。

## COUNT 演算子



データ・セット内のレコードの数を示すメッセージを印刷します。また、データ・セットの複数のレコードで一致する基準に基づき、RC=12 または RC=0 を設定します。

## COUNT 演算子

DFSORT を呼び出して、indd データ・セットを ICETOOL の E35 ユーザー出口にコピーします。USING(yyyy) が指定されている場合は、yyyyCNTL 中の DFSORT 制御ステートメントが使用されます。yyyyCNTL 内の DFSORT INCLUDE または OMIT ステートメントを使用して、入力レコードのサブセットをカウントできます。

INCLUDE または OMIT ステートメントが yyyyCNTL データ・セットで指定されている場合、130 ページの『INCLUDE 制御ステートメント』の『文化環境に関する考慮事項』の項で説明されているように、活動ロケールの照合規則が INCLUDE および OMIT 処理に影響を与えます。

EMPTY、NOTEMPTY、HIGHER(x)、LOWER(y)、EQUAL(v)、または NOTEQUAL(w) が指定されない場合、ICETOOL は、その E35 ユーザー出口が決定したレコード数を含むメッセージを印刷します。

EMPTY、NOTEMPTY、HIGHER(x)、LOWER(y)、EQUAL(v)、または NOTEQUAL(w) が指定される場合、ICETOOL は、その E35 ユーザー出口が決定したレコード数が、指定された基準と一致するか確認します。基準を満たす場合 (たとえば、HIGHER(20) が指定され、レコード数が 21 以上)、ICETOOL は COUNT 演算子に RC=12 を設定します。基準を満たさない場合 (たとえば、HIGHER(20) が指定され、レコード数が 20 以下)、ICETOOL は COUNT 演算子に RC=0 を設定します。ICETOOL は、DFSORT の STOPAFT オプションを使用して、基準と一致するかどうか判断する上で最小限必要なレコード数を処理します。

**注:** 基準を含む COUNT 演算子を最初にセットアップするときは、TOOLMSG のメッセージをチェックして、RC=12 が構文エラーにより発行されていないか確認してください。

ユーザー独自の DFSORT MODS ステートメントを指定できません。このステートメントが、この演算子に対して ICETOOL により渡された MODS ステートメントを指定変更してしまうためです。

**注:** DISPLAY、OCCUR、RANGE、SELECT、STATS、UNIQUE、および VERIFY 演算子の場合も、レコード数が印刷されます。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### USING(yyyy)

この操作で DFSORT により使用される制御ステートメント・データ・セットの DD 名の最初の 4 文字を指定します。yyyy は、形式 yyyyCNTL を構成したときに有効な DD 名となるような、4 文字である必要があります。yyyy に SYSx は指定できません。

USING を指定する場合、yyyyCNTL DD ステートメントが必要であり、またその中の制御ステートメントには次のことが必要です。

1. DFSORT の SORTCNTL データ・セットに関する規則に従っている必要があります。
2. 通常は、INCLUDE または OMIT ステートメント、または注釈ステートメントのみに使用します。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

**VSAMTYPE(x)**

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

**LOCALE(name)**

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

**LOCALE(CURRENT)**

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

**LOCALE(NONE)**

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

**EMPTY**

入力データ・セットまたはサブセットが空の場合、この COUNT 演算子に RC=12 を設定します。入力データ・セットまたはサブセットが空でない場合は、この COUNT 演算子に RC=0 を設定します。

EMPTY は EQUAL(0) と同じです。

**NOTEEMPTY**

入力データ・セットまたはサブセットが空でない場合、この COUNT 演算子に RC=12 を設定します。入力データ・セットまたはサブセットが空の場合は、この COUNT 演算子に RC=0 を設定します。

EMPTY は NOTEQUAL(0) と同じです。

**HIGHER(x)**

レコード数が x より多い場合、この COUNT 演算子に RC=12 を設定します。レコード数が x と同じか少ない場合は、この COUNT 演算子に RC=0 を設定します。

x は n または +n として指定します (ここで、n は 0 ~ 562949953421310)。

**LOWER(y)**

レコード数が x より少ない場合、この COUNT 演算子に RC=12 を設定します。レコード数が y と同じか多い場合は、この COUNT 演算子に RC=0 を設定します。

y は n または +n として指定します (ここで、n は 0 ~ 562949953421310)。

**EQUAL(v)**

レコード数が v と同じである場合、この COUNT 演算子に RC=12 を設定します。レコード数が v と同じでない場合は、この COUNT 演算子に RC=0 を設定します。

v は n または +n として指定します (ここで、n は 0 ~ 562949953421310)。

## COUNT 演算子

### NOTEQUAL(w)

レコード数が w と同じでない場合、この COUNT 演算子に RC=12 を設定します。レコード数が w と同じである場合は、この COUNT 演算子に RC=0 を設定します。

w は n または +n として指定します (ここで、n は 0 ~ 562949953421310)。

## COUNT の例

### 例 1

この例では、CTL1CNTL データ・セットに DFSORT INCLUDE ステートメントが含まれているものとします。

```
COUNT FROM(IN1)
COUNT FROM(IN2) USING(CTL1)
```

最初の COUNT 演算子は、IN1 データ・セット内のレコードの数を示すメッセージを印刷します。

2 番目の COUNT 演算子は、IN2 データ・セットから組み込まれたレコードの数を示すメッセージを印刷します。

### 例 2

```
COUNT FROM(INPUT1) EMPTY
```

INPUT1 が空の場合 (すなわち、INPUT1 にレコードがない)、RC=12 を設定します。INPUT1 が空でない場合 (すなわち、INPUT1 に少なくとも 1 つのレコードがある)、RC=0 を設定します。

### 例 3

この例では、CTL2CNTL データ・セットに DFSORT INCLUDE ステートメントが含まれているものとします。

```
COUNT FROM(INPUT2) HIGHER(50000) USING(CTL2)
```

50000 を超えるレコードが INPUT2 から含まれる場合、RC=12 を設定します。50000 以下のレコードが INPUT2 から含まれる場合、RC=0 を設定します。

---

## DEFAULTS 演算子

▶▶—DEFAULTS—LIST(listdd)—▶▶

別個のリスト・データ・セットに DFSORT インストール・デフォルトを印刷します。

DFSORT により、次のような 8 つのインストール・モジュールを使用して、8 つの異なるインストール・デフォルトのセットを維持できます。

- 環境インストール・モジュール
  - JCL (ICEAM1 モジュール) - バッチ JCL が直接呼び出すインストール・モジュール
  - INV (ICEAM2 モジュール) - バッチ・プログラムが呼び出すインストール・モジュール
  - TSO (ICEAM3 モジュール) - TSO が直接呼び出すインストール・モジュール
  - TSOINV (ICEAM4 モジュール) - TSO プログラムが呼び出すインストール・モジュール
- 時刻インストール・モジュール
  - TD1 (ICETD1 モジュール) - 1 番目の時刻インストール・モジュール
  - TD2 (ICETD2 モジュール) - 2 番目の時刻インストール・モジュール
  - TD3 (ICETD3 モジュール) - 3 番目の時刻インストール・モジュール
  - TD4 (ICETD4 モジュール) - 4 番目の時刻インストール・モジュール

各インストール・デフォルトには、2 つまたはそれ以上の値があります。DFSORT は、IBM 提供の一組のデフォルトを付けて出荷されますが、そのデフォルトは ICEMAC マクロを使用して変更できます。DEFAULTS 演算子により、DFSORT のインストール時に各インストール・モジュールに選択されたインストール・デフォルトを、簡単に判別できます。ICEMAC、8 つのインストール・モジュール、およびインストール・デフォルトとその値の詳細については、*DFSORT 導入とカスタマイズ リリース 14* を参照してください。

DEFAULTS は、ICEAM1-4 のインストール・デフォルトを示し、続いて ICETD1-4 のインストール・デフォルトを示す報告書を作成します。DEFAULTS が作成する報告書の形式は、選択されたデフォルトにより変わりますが、大体は次のようになります。

## DEFAULTS 演算子

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS - p -

\* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	TSO (ICEAM3)	TSOINV (ICEAM4)
-----	-----	-----	-----	-----
item	value	value	value	value
.				
.				
item	value	value	value	value
item	value	value * IBM_value	value	value
.				
.				

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS - p -

\* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	TD1 (ICETD1)	TD2 (ICETD2)	TD3 (ICETD3)	TD4 (ICETD4)
-----	-----	-----	-----	-----
item	value	value	value	value
.				
.				
item	value * IBM_value	value * IBM_value	value	value
item	value	value	value	value
.				
.				

各項目の値 (8 つのインストール環境のそれぞれについて) は、STEPLIB、JOBLIB、またはリンク・ライブラリーからロードされた、ICEAM1-4 および ICETD1-4 インストール・モジュールにセットされたとおりに表示されます。IBM 提供の値と異なる値があれば、その下に IBM 提供の値が示されます。

各項目の値 (8 つのインストール環境のそれぞれについて) は、STEPLIB、JOBLIB、またはリンク・ライブラリーからロードされた、ICEAM1-4 および ICETD1-4 インストール・モジュールにセットされたとおりに表示されます。IBM 提供の値と異なる値があれば、その下に IBM 提供の値が示されます。

各レコードの最初のバイトは制御文字です。タイトルおよびヘッダーは常に印刷されます。p はページ番号です。項目名の列は 10 バイトを占め、項目値の列のそれぞれは 20 バイトを占め、列と列の間には 5 つのブランクがあります。

## オペランドの説明

### LIST(listdd)

この操作で ICETOOL によりリスト・データ・セットの DD 名を作成することを指定します。listdd DD ステートメントは必須です。ICETOOL は、リスト・データ・セットについて指定された RECFM=FBA、LRECL=121、および BLKSIZE を使用します。指定した BLKSIZE が 121 の倍数でない場合は、ICETOOL は BLKSIZE=121 を使用します。BLKSIZE を指定しないと、ICETOOL は、ICEAM2 または ICEAM4 から SDBMSG インストール・オプションで指定されているブロック・サイズを選択します (DFSORT 導入とカスタマイズ リリース 14 を参照)。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

## DEFAULTS の例

```
DEFAULTS LIST(OPTIONS)
```

OPTIONS データ・セットで、DFSORT インストール・デフォルトを印刷します。  
OPTIONS 出力は新しいページで始まり、次のように表示されます (ICEAM1-4 報告書と ICETD1-4 報告書では、最初のいくつかの項目が、説明の都合上の値で示されています)。

## DEFAULTS 演算子

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS - 1 -

\* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	TSO (ICEAM3)	TSOINV (ICEAM4)
RELEASE	14.0	14.0	14.0	14.0
MODULE	ICEAM1	ICEAM2	ICEAM3	ICEAM4
APAR LEVEL	PQ56402	PQ56402	PQ56402	PQ56402
COMPILED	03/06/02	03/06/02	03/06/02	03/06/02
ENABLE	NONE	TD1	NONE	NONE
ABCODE	MSG	99 * MSG	MSG	99 * MSG
ALTSEQ	SEE BELOW	SEE BELOW	SEE BELOW	SEE BELOW
ARESALL	0	0	0	0
ARESINV	NOT APPLICABLE	0	NOT APPLICABLE	0
CFW	YES	YES	YES	YES
CHALT	YES * NO	YES * NO	NO	NO
CHECK	YES	YES	YES	YES
CINV	YES	YES	YES	YES
COBEXIT	COB2 * COB1	COB2 * COB1	COB2 * COB1	COB2 * COB1
DIAGSIM	NO	NO	NO	NO
DSA	32	32	32	32
.				
.				
.				

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS - 4 -

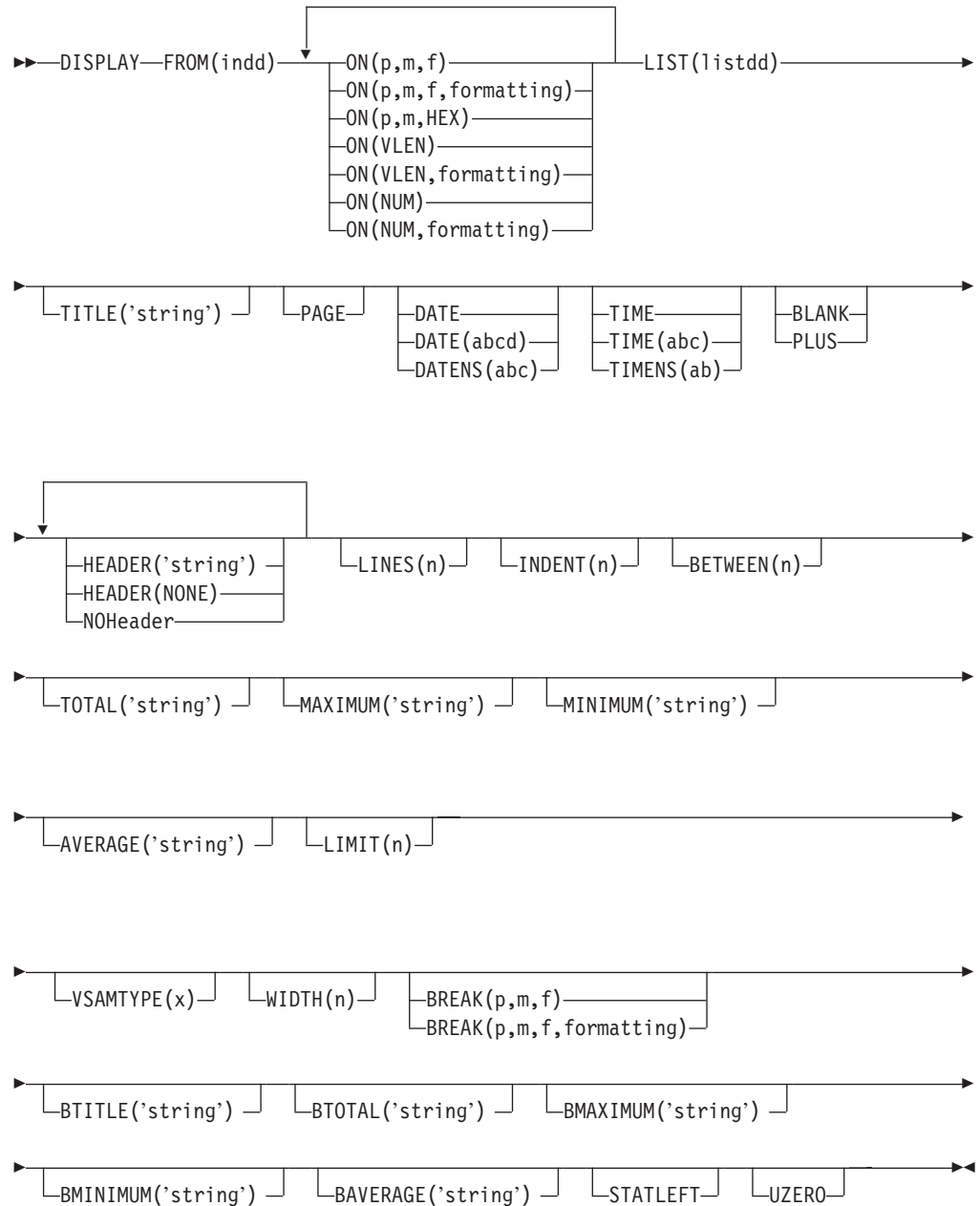
\* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	TD1 (ICETD1)	TD2 (ICETD2)	TD3 (ICETD3)	TD4 (ICETD4)
RELEASE	14.0	14.0	14.0	14.0
MODULE	ICETD1	ICETD2	ICETD3	ICETD3
APAR LEVEL	PQ56402	PQ56402	PQ56402	PQ56402
COMPILED	03/06/02	03/06/02	03/06/02	03/06/02
SUN	0600-2000 * NONE	NONE	NONE	NONE
MON	NONE	NONE	NONE	NONE
TUE	NONE	NONE	NONE	NONE
WED	NONE	NONE	NONE	NONE
THU	NONE	NONE	NONE	NONE
FRI	NONE	NONE	NONE	NONE
SAT	0600-2000 * NONE	NONE	NONE	NONE
ABCODE	99 * MSG	MSG	MSG	MSG
ALTSEQ	SEE BELOW	SEE BELOW	SEE BELOW	SEE BELOW
ARESALL	0	0	0	0
ARESINV	0	0	0	0
CFW	YES	YES	YES	YES
CHALT	YES * NO	NO	NO	NO
CHECK	YES	YES	YES	YES
CINV	YES	YES	YES	YES
COBEXIT	COB2 * COB1	COB1	COB1	COB1
DIAGSIM	NO	NO	NO	NO
DSA	48 * 32	32	32	32
.				
.				
.				

タイトルと適当なヘッダー行が、各ページの一番上に示されています。指定された IBM 提供の ALTSEQ テーブルが、他の項目の後に別々に印刷されます。



DISPLAY 演算子



| 別個のリスト・データ・セット内の指定した数値フィールド (SMF 日付または時間  
 | を含む) または文字フィールドの値または文字を印刷します。簡単で、調整済み  
 | の、セクションごとにまとめられた報告書を作成できます。1 ~ 20 個のフィール  
 | ドを指定できますが、その結果出力されるリスト・データ・セットの行の長さは、  
 | **WIDTH** オペランドで指定された限度を超えてはなりません。また、**WIDTH** が指定  
 | されていない場合は、2048 バイトを超えてはなりません。レコード番号は、特殊フ  
 | ールドとして印刷できます。

## DISPLAY 演算子

DFSORT を呼び出して、indd データ・セットを ICETOOL の E35 ユーザー出口にコピーします。ICETOOL はその E35 ユーザー出口を使用して、リスト・データ・セット内の適切なタイトル、ヘッダー、およびデータを印刷します。

ユーザー独自の DFSORT MODS、INREC、または OUTREC ステートメントを指定できません。これらのステートメントが、この演算子で ICETOOL により渡される DFSORT ステートメントを指定変更してしまうためです。

出力データ列をそれより小さいフィールドに『圧縮』できる形式設定項目または **PLUS** または **BLANK** オペランドを指定すると、プリンターまたはディスプレイがサポートする文字幅により行の長さが制限されている場合に、ユーザー報告書に最大 20 個までのより多くのフィールドを含めることができます。

## 簡単な報告書

必要なオペランドを指定して、簡単な報告書を作成できます。たとえば、FROM および LIST オペランドを指定し、また 10 バイトの文字フィールドと 7 バイトのゾーン 10 進数フィールドに ON オペランドを指定した場合は、リスト・データ・セットの出力を次のように表すことができます。

(p,m,f) characters	(p,m,f) sdddddddddddddd
.	.
.	.
.	.

制御バイトは、各リスト・データ・セットの最初のバイトを占有します。左寄せされた標準ヘッダーが各ページの一番上に印刷され、各列の内容を示します。その後には各レコードごとにそのレコードのフィールド内の文字と数値を示す 1 行が続きます。

フィールドは、DISPLAY ステートメントで指定された順序と同じ順序で列内に印刷されます。すべてのフィールドは左寄せされます。数値フィールドの場合、先行ゼロが印刷されます。負符号には -、正符号には + が使用されます。

デフォルトでは、データの最初の列は制御文字の直後から開始され、列の間には 3 つのブランクが入ります。INDENT オペランドを使用して、データの最初の列の前に挿入されるブランクの数を変更します。BETWEEN オペランドを使用して、列の間のブランクの数を変更します。

標準の列の幅は次のとおりです。

- 文字データ: 文字フィールドの長さ、または 20 バイト (フィールド長が 20 バイト以下の場合)
- 数値データ: 16 バイト
- レコード番号: 15 バイト

HEADER オペランドを使用して、ヘッダーを変更したり抑制できます。形式設定項目、または PLUS または BLANK オペランドを使用して、報告書内の数値フィールドの形式を変更できます。PLUS、BLANK、および HEADER オペランドを使用して、数値フィールドと文字フィールドの列の幅や、ヘッダーとフィールドの位置調整を変更できます。

NOHEADER オペランドを使用して、データ・レコードだけを含むリスト・データ・セットを作成できます。このようにして作成されたデータ・セットはさらに、文字値の場合は CH 形式、また数値の場合は CSF/FS 形式を使用する他の演算子（たとえば、STATS または UNIQUE）により処理できます。

TOTAL、MAXIMUM、MINIMUM、および AVERAGE を使用して、データ列の後に数値フィールドの統計を印刷できます。形式設定項目を使用して、選択した数値フィールドの統計を抑制できます。

## 調整済み報告書

タイトル、日付、時間、ページ番号、ヘッダー、ページ当たり行数、フィールド形式、および数値データの列については、合計、最大、最小、および平均の値を制御する各種のオペランドを使用して、リスト・データ・セット内の出力を調整できます。任意指定のオペランドをさまざまな組み合わせで使用して、各種の報告書形式を作成できます。たとえば、FROM、LIST、BLANK、TITLE、PAGE、DATE、TIME、HEADER および AVERAGE オペランド、および 10 バイト文字フィールドと 7 バイトのゾーン 10 進数フィールドの場合は ON オペランドを指定すれば、リスト・データ・セット内の出力を次のように表示できます。

title	- p -	mm/dd/yy	hh:mm:ss
header	header		
-----	-----		
characters	sd		
.	.		
.	.		
.	.		
average	sd		

制御バイトは、各リスト・データ・セットの最初のバイトを占有します。タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素（タイトル・ストリング、ページ番号、日付および時間）が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

ユーザーが指定したヘッダー（下線が引かれている）が、各ページのタイトル行の次に印刷され、各列の内容を示します。このヘッダーの次には 1 レコードごとに 1 行が続き、そのレコードのフィールド内に文字と数字が表示されます。文字フィールドのヘッダーは左寄せされ、数値フィールドのヘッダーは右寄せされます。

ユーザーが指定した統計行（合計、最大、最小、平均、およびそれらに関連するストリング）は、データ列の次に、選択した数値フィールドごとに印刷されます。関連するストリングは最初の列またはその左の列に印刷されます。

フィールドは、DISPLAY ステートメントで指定された順序と同じ順序で列内に印刷されます。文字フィールドは左寄せされ、数値フィールドは右寄せされます。数値フィールドの場合、先行ゼロは抑制され、負符号には - が、正符号には空白が使用されます（正符号に + を使用する場合は、BLANK ではなく PLUS を指定できます）。

## DISPLAY 演算子

形式設定項目を使用して、報告書の各数値フィールドの形を、区切り文字、桁数、小数点、小数部、符号、および先行ゼロについて変更できます。また、10、100、1000、10000、100000、1000000、1000000000、1024、1048576 (1024\*1024)、または 1073741824 (1024\*1024\*1024) による分離のほか、先行ストリング、浮動小数点ストリング、および後書きストリングについても変更できます。形式設定項目はまた、先行または後書きストリングを文字フィールドに挿入するのに使用できます。

列の幅は、ヘッダーの長さ、文字データまたは数値データに必要な最大バイト数に応じて、動的に調整されます。

## セクションごとにまとめた報告書

BREAK オペランドを含めて、報告書をいくつかのセクションに分けるために使用する段落フィールドを示すことにより、セクションごとにまとめられた (簡単な、または調整済みの) 報告書を作成できます。各一組の順次入力レコード (必要に応じて、事前に段落フィールドと他のフィールドで分類されている) が、指定された段落フィールドについて同じ値をもっている場合は、報告書の中の 1 つのセクションとして扱われる一組のデータ行になります。

段落フィールドは、各セクションの先頭に印刷されます。形式設定項目を使用して、数値段落フィールドの表示を変更できます。さらに、文字または数値段落フィールドの前後にストリングを挿入することもできます。

オプションの段落オペランドを使用して、各セクションの段落のタイトルを修正できます (段落値は常に段落のタイトルの一部として印刷されます)。さらに、各セクションで選択した数値フィールドの統計を印刷することもできます。たとえば、BTITLE、BREAK、BMAXIMUM、および BMINIMUM を上述の調整済み報告書のオペランドに追加すると、リスト・データ・セットの中の出力の各セクションは新しいページで始まり、次のように表示できます。

```
title      - p -      mm/dd/yy      hh:mm:ss
btitle bvalue
header      header
-----
characters      sd
.
.
.
bmaximum      sd
bminimum      sd
```

全体の統計を示す最後のページは新しいページで始まり、次のように表示できます。

```
title      - p -      mm/dd/yy      hh:mm:ss
header      header
-----
average      sd
```

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

この操作で DFSORT により読み込まれる入力データ・セットの DD 名を指定します。indd DD ステートメントが必要で、入力データ・セットを定義します。このデータ・セットは DFSORT の SORTIN データ・セットに関する規則に従ったものである必要があります。さらに、データ・セットの LRECL は 4 以上である必要があります。

### ON(p,m,f)

この操作で使用される数値フィールドまたは文字フィールドの位置、長さ、および形式を指定します。'(p,m,f)' は標準列見出しに使用されます (代替のヘッダー・オプションについては、HEADER('string')、HEADER(NONE) および NOHEADER を参照してください)。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。p は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です (RRRR は、4 バイト・レコード記述子ワードを表します)。

Fixed-length record	Variable-length record
D   A   T   A   ...	R   R   R   R   D   A   T   A   ...
p= 1 2 3 4	p= 1 2 3 4 5 6 7 8

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを超えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
BI	1 ~ 4 バイト	2 進数、符号なし
FI	1 ~ 4 バイト	固定小数点、符号あり
PD	1 ~ 8 バイト	パック 10 進数、符号あり
ZD	1 ~ 15 バイト	ゾーン 10 進数、符号あり
CH	1 ~ 1500 バイト	文字
CSF または FS	1 ~ 16 バイト (15 桁まで)	任意指定の浮動符号が左端にある符号付き数字
DT1	4 バイト	Z'yyyymmdd' と解釈された SMF 日付
DT2	4 バイト	Z'yyyymm' と解釈された SMF 日付
DT3	4 バイト	Z'yyyyddd' と解釈された SMF 日付
TM1	4 バイト	Z'hmmss' と解釈された SMF 時間
TM2	4 バイト	Z'hmm' と解釈された SMF 時間

## DISPLAY 演算子

形式コード	長さ	説明
TM3	4 バイト	Z'hh' と解釈された SMF 時間
TM4	4 バイト	Z'hmmssxx' と解釈された SMF 時間

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

CSF 形式または FS 形式フィールドの場合、

- 最大 15 桁が使用できます。16 桁の CSF/FS 値が見つかったら、ICETOOL はエラー・メッセージを出して操作を終了します。

ZD 形式または PD 形式フィールドの場合、

- 10 進数値が無効な数字 (A ~ F) を含んでいる場合、ICETOOL はその誤った値をメッセージに識別し、リスト・データ・セット内のその値、および合計、最大、最小、および平均 (指定されている場合) に対してアスタリスクを印刷します。間違った値の数が無効な 10 進値の LIMIT に達すると、ICETOOL は操作を終了します。LIMIT オペランドが指定されていない場合は、無効な 10 進値の限界としてデフォルトの 200 が使用されます。
- 符号が F、E、C、A、8、6、4、2、または 0 の場合、値は正の値として扱われます。
- 符号が D、B、9、7、5、3、または 1 の場合、値は負の値として扱われます。

DT1、DT2 または DT3 形式フィールドの場合、

- 無効な SMF 日付があると、データ例外 (0C7 ABEND)、または誤った ZD 日付が発生します。
- SMF 日付値は常に正の数値として扱われます。

TM1、TM2、TM3 または TM4 形式フィールドの場合、

- 無効な SMF 時間があると、誤った ZD 時間が発生します。
- SMF 時間値は常に正の数値として扱われます。

### **ON(p,m,f,formatting)**

この操作で使用される数値フィールドまたは文字フィールドの位置、長さ、および形式を指定し、また、このフィールドのデータの印刷に関してどのように形式設定するかを指定します。BLANK オペランドは、自動的に有効になります。

詳細については、ON(p,m,f) を参照してください。

### **formatting**



このフィールドのデータが印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。数値フィールドにはどんな形式設定項目も指定できますが、L'string' および T'string' に限り、文字フィールドに指定します。

列の幅は、すべての形式設定項目が指定された結果挿入されるバイトの最大数が収容できるように動的に調整されます。

**mask**

編集マスクをこのフィールドの数値データに適用するように指定します。32 の事前定義の編集マスクが使用でき、これらの編集マスクは、世界中で使われている区切り文字、小数点、小数部、符号といった数値表記の多くを包含しています。ICETOOL は、選択されたマスクに従ってデータを編集します。他の形式設定項目が指定されていて、マスクが指定されていない場合は、A0 のデフォルト・マスクがデータに適用されます。

E'pattern' は、マスクと一緒に指定できません。

各マスク・グループの属性は下記の通りです。

表 65. 編集マスクの属性

マスク	区切り文字	10 進数位置	正符号	負符号
A0	N	0	ブランク	-
A1-A5	Y	0	ブランク	-
B1-B6	Y	1	ブランク	-
C1-C6	Y	2	ブランク	-
D1-D6	Y	3	ブランク	-
E1-E4	Y	0	ブランク	( )
F1-F5	Y	2	ブランク	( )

下表は使用可能なマスクを記述していて、値 12345678 および -1234567 が各マスクにたいしてどのように印刷されるかを示しています。パターンでは、

- **d** は、10 進数 (0 ~ 9) を表すのに使用されます。
- **w** は、先頭の符号を表すのに使用され、正の値ではブランク、負の値では - になります。
- **x** は、後書きの符号を表すのに使用され、正の値ではブランク、負の値では - になります。

## DISPLAY 演算子

- **y** は、先頭の符号を表すのに使用され、正の値ではブランク、負の値では ( になります。
- **z** は、後書きの符号を表すのに使用され、正の値ではブランク、負の値では) になります。

表 66. 編集マスクのパターン

マスク	パターン	12345678	-1234567
A0	wdddddddddddd	12345678	-1234567
A1	wddd,ddd,ddd,ddd,ddd	12,345,678	-1,234,567
A2	wddd.ddd.ddd.ddd.ddd	12.345.678	-1.234.567
A3	wddd ddd ddd ddd ddd	12 345 678	-1 234 567
A4	wddd'ddd'ddd'ddd'ddd	12'345'678	-1'234'567
A5	ddd ddd ddd ddd dddx	12 345 678	1 234 567-
B1	wdd,ddd,ddd,ddd,ddd.d	1,234,567.8	-123,456.7
B2	wdd.ddd.ddd.ddd.ddd,d	1.234.567,8	-123.456,7
B3	wdd ddd ddd ddd ddd,d	1 234 567,8	-123 456,7
B4	wdd'ddd'ddd'ddd'ddd.d	1'234'567.8	-123'456.7
B5	wdd'ddd'ddd'ddd'ddd,d	1'234'567,8	-123'456,7
B6	dd ddd ddd ddd ddd,dx	1 234 567,8	123 456,7-
C1	wd,ddd,ddd,ddd,ddd.dd	123,456.78	-12,345.67
C2	wd.ddd.ddd.ddd.ddd,dd	123.456,78	-12.345,67
C3	wd ddd ddd ddd ddd,dd	123 456,78	-12 345,67
C4	wd'ddd'ddd'ddd'ddd.dd	123'456.78	-12'345.67
C5	wd'ddd'ddd'ddd'ddd,dd	123'456,78	-12'345,67
C6	d ddd ddd ddd ddd,dxx	123 456,78	12 345,67-
D1	wddd,ddd,ddd,ddd.ddd	12,345.678	-1,234.567
D2	wddd.ddd.ddd.ddd.ddd	12.345,678	-1.234,567
D3	wddd ddd ddd ddd,ddd	12 345,678	-1 234,567
D4	wddd'ddd'ddd'ddd.ddd	12'345.678	-1'234.567
D5	wddd'ddd'ddd'ddd,ddd	12'345,678	-1'234,567
D6	ddd ddd ddd ddd,dxx	12 345,678	1 234,567-
E1	yddd,ddd,ddd,ddd,dddz	12,345,678	(1,234,567)
E2	yddd.ddd.ddd.ddd,dddz	12.345.678	(1.234.567)
E3	yddd ddd ddd ddd,dddz	12 345 678	(1 234 567)
E4	yddd'ddd'ddd'ddd,dddz	12'345'678	(1'234'567)
F1	yd,ddd,ddd,ddd,ddd.ddz	123,456.78	(12,345.67)
F2	yd.ddd.ddd.ddd,ddd,ddz	123.456,78	(12.345,67)
F3	yd ddd ddd ddd,ddd,ddz	123 456,78	(12 345,67)
F4	yd'ddd'ddd'ddd,ddd,ddz	123'456.78	(12'345.67)
F5	yd'ddd'ddd'ddd,ddd,ddz	123'456,78	(12'345,67)

LZ が指定されている場合、先行ゼロが印刷されます。たとえば、+1 は、ON(21,6,FS,C1,LZ) の指定で 0,000.01 と表示されます。



LZ が指定されていない場合、不適切でない限り先行ゼロは抑制されます。たとえば、+1 は、ON(21,6,FS,A1) の指定で 1 と表示されます。また、ON(21,6,FS,C1) の指定では 0.01 と表示されます。

先頭の符号 (正の数には空白、負の数には -) は形式設定値の抑制されていない最初の桁に表示されます。たとえば、-1 は、ON(21,6,FS,A2) の指定で -1 と表示されます。また、ON(21,6,FS,A2,LZ) の指定では -000.001、ON(21,6,FS,C2) の指定では -0.01 と表示されます。

### E'pattern'

編集パターンをこのフィールドの数値データに適用するように指定します。

E'pattern' は、電話番号、日付、時刻、社会保障番号など、**符号のない**数値データの形式設定に役立ちます。たとえば、0123456789 は、ON(21,10,ZD,E'(999)-999-9999) の指定で (012)-345-6789 と表示されます。

パターン (1 ~ 24 文字) は、単一アポストロフィで囲まれます。パターン (15 文字まで) 中の **9** は、相当する桁の数値と置き換えられます。パターンの 9 以外の文字は、指定どおりに表示されます。単一アポストロフィ (') をパターンの中にもめる場合は、2 個の単一アポストロフィ (") を指定します。

F'string' またはマスクは、E'pattern' と一緒に指定できません。

フィールドに E'pattern' が指定される場合、以下のようになります。

- 値は符号なしで表示されます。たとえば、+120622 および -120622 は両方とも、ON(12,7,FS,E'99:99:99) の指定で 12:06:22 と表示されます。
- 値の中の有効数字の数がパターン中の 9 の数より少ない場合、左側はすべて 0 で表示されます。たとえば、1234 は、ON(12,6,FS,E'9999-99) の指定で 0012-34 と表示されます。
- 値の中の有効数字の数がパターン中の 9 の数より多い場合、左側から桁が切り捨てられます。たとえば、1234567 は、ON(9,4,PD,E'99:99\*') の指定で \*45:67\* と表示されます。

### L'string'

先頭ストリングがこのフィールドの文字または数値データ列の最初に表示されるように指定します。たとえば、'DFSORT' は、ON(1,8,CH,L'\*\*\*') の指定で '\*\*\*DFSORT' と表示されます。

ストリング (1 ~ 10 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中にもめる場合は、2 個の単一アポストロフィ (") を指定します。

### F'string'

浮動ストリングが、このフィールドの形式設定済み数値データの非空白文字の左に表示されるように指定します。たとえば、0001234 は、ON(9,7,ZD,C1,F'\$') の指定で \$12.34 と表示されます。

ストリング (1 ~ 10 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中にもめる場合は、2 個の単一アポストロフィ (") を指定します。

E'pattern' は、F'string' と一緒に指定できません。

### T'string'

後書きのストリングがこのフィールドの文字または数値データの最後に表示され

## DISPLAY 演算子

るように指定します。たとえば、'DFSORT' は、ON(1,8,CH,L'\*\*\*',T'\*\*\*\*') の指定で、'\*\*\*DFSORT\*\*\*' と表示されます。

ストリング (1 ~ 10 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中にも含む場合は、2 個の単一アポストロフィ (') を指定します。

**LZ** 指定した編集マスクがフィールドの数値データに適用する場合、抑制される先行ゼロのデフォルトを指定変更して、先行ゼロが印刷されるように指定します。たとえば、+123 は、ON(21,6,FS,A0) の指定で 123 と表示されますが、ON(21,6,FS,A0,LZ) の指定では 000123 と表示されます。

LZ は、アカウント番号など、先行ゼロの印刷が必須の数値データを形式設定するのに役立ちます。

先行ゼロは、LZ が指定されているかいないかにかかわらず、E'pattern' で印刷されます。

### NOST

必要な統計 (TOTAL、MAXIMUM、MINIMUM、AVERAGE、BTOTAL、BMAXIMUM、BMINIMUM、BAVERAGE) が、この数値フィールドに印刷されないように指定します。

### Ndd

列幅を決定するときに、数値フィールドに使用する桁数を指定します。dd に 2 桁の数字 01 から 15 で桁数を指定します。

TOTAL または BTOTAL を指定する場合、デフォルトの桁数は 15 です。TOTAL または BTOTAL に 15 桁未満の値を指定する必要がある場合、Ndd を使用してそれを指示できます。これにより、桁数で決定されている列幅を縮小できます。

dd がデフォルトの桁数以上の場合は、dd が使用されます。dd がデフォルトの桁数より少ない場合、デフォルトの値が使用されます。たとえば、下記のようにします。

TOTAL または BTOTAL を使用する場合、

- ON(1,5,ZD) の指定で、列幅を決定する桁数は 15 桁 (TOTAL のデフォルト) となります。
- ON(1,5,ZD,N10) の指定で、10 桁 (N10 の指定による) となります。
- ON(1,5,ZD,N03) の指定で、5 桁 (5,ZD のデフォルト) となります。

TOTAL および BTOTAL を使用しない場合、

- ON(1,5,ZD) の指定で、列幅を決定する桁数は 5 桁 (5,ZD のデフォルト) となります。
- ON(1,5,ZD,N10) の指定で、10 桁 (N10 の指定による) となります。
- ON(1,5,ZD,N03) の指定で、5 桁 (5,ZD のデフォルト) となります。

Ndd および桁数が使用するオーバーフローの合計を使用する場合、ICETOOL はその合計に対しアスタリスクを印刷し、処理を終了します。

桁数はパターンにより決定されるため、E'pattern' を指定した場合 Ndd は無視されます。

**/x** 形式設定する前にこのフィールドの数値データを単位で区切ることを指定します。x は、下記に述べるとおり、使用する分割の係数を表します。結果の値は、最も近い整数になるように端数が切り捨てられます。統計 (TOTAL、MAXIMUM、MINIMUM、AVERAGE、BTOTAL、BMAXIMUM、BMINIMUM、BAVERAGE) および列幅は、分割した数値を反映します。

**/D** 形式設定の前に、10 の単位で区切ることを指定します。たとえば、-1234 は、ON(11,2,FI,/D) の指定で -123 と表示されます。

**/C** 形式設定の前に、100 の単位で区切ることを指定します。たとえば、12345 は、ON(11,2,BI,/C,B1) の指定で 12.3 と表示されます。

**/K** 形式設定の前に、1000 の単位で区切ることを指定します。たとえば、-1234567890 は ON(1,11,FS,/K,A3) の指定で (1 234 567) と表示されます。

**/DK** 形式設定の前に、10000 (10\*1000) の単位で区切ることを指定します。たとえば、6213849653 は、ON(31,10,FS,/DK,E'9-9999-99') の指定で 0-6213-84 と表示されます。

**/CK** 形式設定の前に、100000 (100\*1000) の単位で区切ることを指定します。たとえば、98765432101 は、ON(21,11,ZD,C1,/CK) の指定で 9,876.54 と表示されます。

**/M** 形式設定の前に、1000000 (1000\*1000) の単位で区切ることを指定します。たとえば、-123456789 は ON(31,10,FS,/M,C4) の指定で -1.23 と表示されます。

**/G** 形式設定の前に、1000000000 (1000\*1000\*1000) の単位で区切ることを指定します。たとえば、1234567898765 は ON(15,13,ZD,A4,/G) の指定で 1'234 と表示されます。

**/KB** 形式設定の前に、1024 の単位で区切ることを指定します。たとえば、1234567890 は ON(45,10,ZD,/KB,A3) の指定で 1 205 632 と表示されます。

**/MB** 形式設定する前に 1048 576 (1024\*1024) の単位で区切ることを指定します。たとえば、123456789 は ON(60,9,FS,/MB) の指定で 117 と表示されます。

**/GB** 形式設定する前に 1073 741 824 (1024\*1024\*1024) の単位で区切ることを指定します。たとえば、1234567898765 は ON(15,13,ZD,/GB,A1) の指定で 1,149 と表示されます。

### ON(p,m,HEX)

文字フィールドの位置と長さが、この操作で使用され、16 進形式 (各バイトごとに 00 ~ FF) で印刷されるように指定します。(p,m,HEX)' は標準列見出しに使用されます。代替のヘッダー・オプションについては、HEADER('string')、HEADER(NONE)、および NOHEADER を参照してください。

**p** については、ON(p,m,f) を参照してください。

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを超えるように拡張できません。フィールドは 1 ~ 1000 バイトにできます。

## DISPLAY 演算子

### ON(VLEN)

ON(1,2,BI) の指定と同じです。すなわち、1 桁目から始まる 2 バイト 2 進数フィールドです。可変長レコードの場合、ON(VLEN) は各レコードのレコード長を表します。RECORD LENGTH' は標準列見出しに使用されます。代替のヘッダー・オプションについては、HEADER('string')、HEADER(NONE)、および NOHEADER を参照してください。

### ON(VLEN,formatting)

ON(1,2,BI,formatting) n の指定と同じです。すなわち、1 桁目から始まる 2 バイト 2 進数フィールドです。また、このフィールドのデータの印刷に関してどのように形式設定するかを指定します。BLANK オペランドは、自動的に有効になります。

詳細については、ON(VLEN) を参照してください。

### formatting



このフィールドのデータが印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。

列の幅は、すべての形式設定項目が指定された結果挿入されるバイトの最大数が収容できるように動的に調整されます。

**formatting**については、ON(p,m,f,formatting) を参照してください。

### ON(NUM)

レコード番号を印刷することを指定します。レコード番号は 1 から始まり、レコードがリスト・データ・セットに印刷されるたびに 1 つずつ増加します。'RECORD NUMBER' は標準列見出しに使用されます。代替のヘッダー・オプションについては、HEADER('string')、HEADER(NONE)、および NOHEADER を参照してください。

### ON(NUM,formatting)

レコード番号を印刷することを指定します。また、レコード番号が印刷のためにどのように形式設定されるかを指定します。BLANK オペランドは、自動的に有効になります。

詳細については、ON(NUM) を参照してください。

### formatting



レコード番号が印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。

列の幅は、すべての形式設定項目が指定された結果挿入されるバイトの最大数が収容できるように動的に調整されます。

**mask** **mask**については、ON(p,m,f,formatting) を参照してください。

#### **E'pattern'**

**E'pattern'** については、ON(p,m,f,formatting) を参照してください。

#### **L'string'**

**L'string'** については、ON(p,m,f,formatting) を参照してください。

#### **F'string'**

**F'string'** については、ON(p,m,f,formatting) を参照してください。

#### **T'string'**

**T'string'** については、ON(p,m,f,formatting) を参照してください。

#### **LZ**

**LZ** については、ON(p,m,f,formatting) を参照してください。

#### **Ndd**

列幅を決定するときに、レコード番号に使用する桁数を指定します。dd に **2** 桁の数字 01 から 15 で桁数を指定します。

レコード番号のデフォルトの桁数は、15 です。レコード番号に 15 桁未満の値を指定する必要がある場合、Ndd を使用してそれを指示できます。これにより、桁数で決定されている列幅を縮小できます。たとえば、ON(NUM,N09) の指定で、15 (レコード番号のデフォルトの桁数) の代わりに 9 桁 (N09 の指定による) が使用されます。

Ndd および桁数が使用するレコード数のオーバーフローを使用する場合、ICETOOL は処理を終了します。

桁数はパターンにより決定されるため、E'pattern' を指定した場合 Ndd は無視されます。

#### **LIST(listdd)**

この操作で ICETOOL によりリスト・データ・セットの DD 名を作成することを指定します。listdd DD ステートメントは必須です。

ICETOOL はリスト・データ・セットの属性を下記のようにセットします。

- RECFM を FBA にセットする。
- LRECL を下記のいずれかにセットする。

- WIDTH(n) が指定された場合、LRECL を n にセットする。LRECL を特定の値にセットしなければならない場合は、WIDTH(n) を使用してください (たとえば、DISP=MOD を使用して、いくつかの報告書を同一のデータ・セットに入れる場合)。
- WIDTH(n) が指定されない場合は、LRECL を 121 にセットするか、または 121 文字より大きい場合は、計算された必要行長にセットする。LRECL を特定の値にセットする必要がない場合は、ICETOOL に決定を任せ、WIDTH(n) を指定することなく適切な LRECL 値にセットできます。
- BLKSIZE を下記のいずれかにセットする。
  - BLKSIZE が使用 LRECL の倍数である場合は、DD ステートメント、DSCB、または ラベルから取った BLKSIZE。
  - DD ステートメント、DSCB、またはラベルから取った BLKSIZE が使用 LRECL の倍数でない場合は、LRECL。
  - BLKSIZE が DD ステートメント、DSCB、またはラベルで使用できない場合は、SDBMSG インストール・オプションにより指示されたブロック・サイズ (DFSORT 導入とカスタマイズ リリース 14 を参照してください)。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### TITLE('string')

タイトル・ストリングをタイトル行に印刷することを指定します。タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個のブランクがあります。タイトル行の後にはブランク行が 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中にもめる場合は、2 個の単一アポストロフィ (') を指定します。ストリングの最初にブランクがあると、テキストは右に移動します。ストリングの終わりにブランクがあると、ストリングと次のタイトル要素の間のスペースが増えます。

**PAGE** ページ番号をタイトル行に印刷することを指定します。ページ番号は -p - 形式 (ここで、p は先行ゼロのない 10 進数) で印刷されます。ページ番号は第 1 ページが 1 で、その後は 1 ページごとに 1 つずつ増加していきます。

タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個のブランクがあります。タイトル行の後にはブランク行が 1 行印刷されます。

**DATE** タイトル行に日付を印刷することを指定します。日付は mm/dd/yy 形式 (ここで、mm は月、dd は日、yy は年) で印刷されます。DATE は DATE(MDY) の指定と同じです。

タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

**DATE(abcd)**

タイトル行に日付を印刷することを指定します。日付は abc および d に指定された値に応じて、'adbdc' の形式で印刷されます。たとえば、2002 年 3 月 29 日の場合、DATE(4MD-) は '2002-03-29' を表示し、DATE(MDY.) は '03.29.02' を表示します。

abc は、M、D、および Y または 4 (それぞれ一度指定される) の任意の組み合わせで指定できます。ここで、M は月 (01 ~ 12) を表し、D は日 (01 ~ 31) を表し、Y は年の最後の 2 桁 (たとえば、02) を表し、そして 4 は年の 4 桁 (たとえば、2002) を表します。

d は任意の文字で、月、日、年を区切るために使用します。

タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

**DATENS(abc)**

タイトル行に日付を印刷することを指定します。日付は abc に指定された値に応じて、'abc' の形式で印刷されます。たとえば、2002 年 3 月 29 日の場合、DATENS(4MD) は '20020329' と表示し、DATENS(MDY) は '032902' と表示します。

abc は、M、D、および Y または 4 (それぞれ一度指定される) の任意の組み合わせで指定できます。ここで、M は月 (01 ~ 12) を表し、D は日 (01 ~ 31) を表し、Y は年の最後の 2 桁 (たとえば、02) を表し、そして 4 は年の 4 桁 (たとえば、2002) を表します。

タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

**TIME** タイトル行に時刻を印刷することを指定します。時刻は hh:mm:ss (ここで、hh は時間、mm は分、ss は秒) の形式で印刷されます。TIME は TIME(24;) の指定と同じです。

タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

**TIME(abc)**

タイトル行に時刻を印刷することを指定します。時刻は ab および c に

## DISPLAY 演算子

指定された値に応じて、'hhcmmss xx' の形式で印刷されます。たとえば、08:25:13 pm の場合、TIME=(24:) は '20:25:13' と表示し、TIME=(12.) は '08.25.13 pm' と表示します。

ab には次の値を指定できます。

- 12 時間制を表すためには 12 を指定します。hh (時間) は 01~12、mm (分) は 00~59、ss (秒) は 00~59、xx は am または pm となります。
- 24 時間制を表すためには 24 を指定します。hh (時間) は 00~23、mm (分) は 00~59、ss (秒) は 00~59 となります。xx は組み込まれません。

c は任意の文字を指定することができ、時間、分、および秒を区切るために使用されます。タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

### TIMENS(ab)

タイトル行に時刻を印刷することを指定します。時刻は ab に指定された値に応じて、'hhmmss xx' の形式で印刷されます。たとえば、08:25:13 pm の場合、TIMENS=(24) は '202513' と表示し、TIMENS=(12) は '082513 pm' と表示します。

ab には次の値を指定できます。

- 12 時間制を表すためには 12 を指定します。hh (時間) は 01~12、mm (分) は 00~59、ss (秒) は 00~59、xx は am または pm となります。
- 24 時間制を表すためには 24 を指定します。hh (時間) は 00~23、mm (分) は 00~59、ss (秒) は 00~59 となります。xx は組み込まれません。

タイトル行は、リスト・データ・セットの各ページの一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

### BLANK

文字および数字の印刷について、次のような代替形式を指定します。

- 形式設定が指定されていない数値は、空白 (正符号の場合)、- (負符号で先行ゼロなしの場合) で印刷されます (正符号で先行ゼロのある場合は、+ のデフォルトを指定変更)。

したがって、数値は次のように表示されます。

- d...d (正符号の場合、数字のすぐ左側には空白符号が付き、先行ゼロなし)
- -d...d (負符号の場合、数字のすぐ左側には - 符号が付き、先行ゼロなし)



- 列の幅は、ヘッダーの長さ、および文字または数値データに必要なバイトの最大数に応じて動的に調整されます。
- 数値フィールドのヘッダーおよびデータは右寄せされます (数値フィールドのヘッダーおよびデータの左寄せのデフォルトを変更)。

**PLUS** 文字および数字の印刷について、次のような代替形式を指定します。

- 形式設定が指定されていない数値は、+ (正符号)、- (負符号で先行ゼロなし) を付けて印刷されます (先行ゼロのデフォルトを指定変更)。  
したがって、数値は次のように表示されます。
  - +d...d (正符号の場合、数字のすぐ左側には + 符号が付き、先行ゼロなし)
  - -d...d (負符号の場合、数字のすぐ左側には - 符号が付き、先行ゼロなし)
- 列の幅は、ヘッダーの長さ、および文字または数値データに必要なバイトの最大数に応じて動的に調整されます。
- 数値フィールドのヘッダーおよびデータは右寄せされます (数値フィールドのヘッダーおよびデータの左寄せのデフォルトを変更)。

ON(NUM) の場合、PLUS は BLANK として扱われます。

#### HEADER('string')

対応する ON フィールドについて、印刷されるヘッダーを指定します。対応する ON フィールドについて、標準列見出しの代わりに指定されたストリングが使用されます。(ON フィールドおよび HEADER オペランドは、指定された順序に従って 1 対 1 で対応します。すなわち、最初の HEADER オペランドは最初の ON フィールドに、2 番目の HEADER オペランドは 2 番目の ON フィールドにというように対応します。)

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (') を指定します。ストリングの長さが、対応する ON フィールドの列の幅より大きい場合は、その列の幅はストリングの長さに増やされます。

ヘッダーは左寄せ (文字フィールドの場合) または右寄せ (数値フィールドの場合) され、列の幅全体にハイフンで下線が引かれます (デフォルトの右寄せ、下線なしのヘッダーを指定変更)。文字の値は左寄せされ、数値は右寄せされます (左寄せフィールド値のデフォルトを指定変更)。

ヘッダー・ストリングの先頭または終わりにあるブランクは、ヘッダーまたは列の幅を指定変更する可能性があります。

いずれかの ON フィールドに HEADER('string') を使用する場合は、各 ON フィールドに HEADER('string') または HEADER(NONE) を使用します。

#### HEADER(NONE)

対応する ON フィールドのヘッダーを印刷しないことを指定します。対応する ON フィールドの標準列見出しが抑制されます。

## DISPLAY 演算子

いずれかの ON フィールドに HEADER('string') を使用する場合は、各 ON フィールドに HEADER('string') または HEADER(NONE) を使用します。ON フィールドごとに HEADER(NONE) を指定することは、NOHEADER を指定することと同じです。

### NOHEADER

ON フィールドのヘッダーを印刷しないことを指定します (ON フィールドの標準ヘッダーを印刷するデフォルトを指定変更)。

NOHEADER を使用する場合は、1 回だけ指定します。また、HEADER('string') または HEADER(NONE) を使用できません。

TITLE、DATE、TIME、または PAGE オペランドを指定しないで NOHEADER を指定すると、リスト・データ・セットにはデータ・レコードだけしか含まれないこととなります。このようにして作成されたデータ・セットはさらに、文字値の場合は CH 形式を、数値の場合は CSF/FS 形式を使用する他の演算子 (たとえば、STATS または UNIQUE) により処理できます。

### LINES(n)

リスト・データ・セットに対して、ページ当たり行数を指定します (デフォルトの 58 を指定変更)。n は 9 より大きく、1000 より小さくします。

### INDENT(n)

報告書のインデントに使用されるブランクの数を指定します (デフォルトの 0 を指定変更)。n は 0 ~ 50 の範囲で指定できます。

INDENT(n) が指定されていない場合、報告書は 2 列目 (制御文字の後) から開始されます。一方、INDENT(10) が指定されている場合、報告書は 12 列目 (制御文字と 10 個のブランクの後) から開始されます。

### BETWEEN(n)

データの列と列の間で使用されるブランクの数を指定します (デフォルトの 3 を指定変更)。n は 0 ~ 50 の範囲で指定できます。たとえば、BETWEEN(n) が指定されない場合、3 個のブランクが列間に使用されます。一方、BETWEEN(7) が指定される場合、7 個のブランクが列間に使用されます。

### TOTAL('string')

報告書のデータ行の後に全体の TOTAL 行を印刷することを指定します。指定されたストリングを全体の TOTAL 行の インデント列から印刷し、その後に、各数値データ列の全体の合計が続きます。STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。合計は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。合計は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。全体の TOTAL 行の前にブランクが 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中にもめる場合は、2 個の単一アポストロフィ (``) を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、TOTAL('') を指定します。

各数値 ON フィールドの全体の合計は、ユーザーが指定した形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの合計は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。合計は ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

デフォルトでは、列幅は最大で 1 つの符号と 15 桁を合計として印刷できるように調整されます。ON フィールドの全体の合計が 15 桁を超えると、ICETOOL はそのフィールドの全体の合計にアスタリスクを入れ、処理を終了します。

Ndd 形式設定項目を使用して、合計に使用する桁数を減らすことができます。ON フィールドの全体の合計が使用されている桁数を超えると、ICETOOL はそのフィールドの全体の合計にアスタリスクを入れ、処理を終了します。

TOTAL、MAXIMUM、MINIMUM、および AVERAGE 行は、ユーザーがそれらを指定した順序で印刷されます。

### MAXIMUM('string')

報告書のデータ行の後に全体の MAXIMUM 行を印刷することを指定します。指定されたストリングを全体の MAXIMUM 行の インデント列から印刷し、その後、各数値データ列の全体の最大値が続きます。STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。最大値は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。最大値は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。全体の MAXIMUM 行の前にブランクが 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (``) を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、MAXIMUM('') を指定します。

各数値 ON フィールドの全体の最大値は、ユーザーが指定した形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの最大値は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。最大値は、ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

TOTAL、MAXIMUM、MINIMUM、および AVERAGE 行は、ユーザーがそれらを指定した順序で印刷されます。

### MINIMUM('string')

全体の MINIMUM 行を報告書のデータ行の後に印刷することを指定します。指定されたストリングを全体の MINIMUM 行の インデント列から印刷し、その後、各数値データ列の全体の最小値が続きます。STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。最小値は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷さ

## DISPLAY 演算子

れます。最小値は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。全体の MINIMUM 行の前に空白が 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (``) を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、MINIMUM('') を指定します。

各数値 ON フィールドの全体の最小値は、ユーザーが指定した形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの最小値は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。最小値は、ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

TOTAL、MAXIMUM、MINIMUM、および AVERAGE 行は、ユーザーがそれらを指定した順序で印刷されます。

### AVERAGE('string')

全体の AVERAGE 行を報告書のデータ行の後に印刷することを指定します。指定されたストリングを全体の AVERAGE 行の インデント列から印刷し、その後、各数値データ列の全体の平均が続きます。

STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。平均は、ストリングとして同じ行に印刷されます。

STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。平均は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。空白行を全体の AVERAGE 行の前に 1 行印刷します。

全体の平均 (つまり、算術平均) は、全体の合計を報告書内の値の数で割り、端数を切り捨てて最も近い整数にします (たとえば、 $23 / 5 = 4$ 、 $-23 / 5 = -4$ )。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (``) を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、AVERAGE('') を指定します。

各数値 ON フィールドの全体の平均値は、ユーザーが指定した形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの平均は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。平均値は、ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

ON フィールドの全体の合計が 15 桁を超えると、ICETOOL はそのフィールドの全体の平均にアスタリスクを入れ、処理を終了します。

TOTAL、MAXIMUM、MINIMUM、および AVERAGE 行は、ユーザーがそれらを指定した順序で印刷されます。

### LIMIT(n)

無効な 10 進数値の数について限界を指定します (デフォルトの 200 を

指定変更)。n 個の無効な 10 進数値が見つかったら、ICETOOL は操作を終了します。n には 1 ~ 15 桁の 10 進数を指定できますが、0 より大きくなくてはなりません。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### WIDTH(n)

ユーザーのリスト・データ・セット用に ICETOOL に使用させたい行の長さとして LRECL を指定します。n は 121 ~ 2048 の範囲で指定できます。

ICETOOL は常に、すべてのタイトル、ヘッダー、データ、および統計を印刷するために必要な行の長さを計算し、それを次のように使用します。

- WIDTH(n) が指定され、計算された行の長さが n より小さいかまたは等しい場合、ICETOOL は 行の長さとして LRECL を n にセットします。
- WIDTH(n) が指定され、計算された行の長さが n より大きい場合、ICETOOL はエラー・メッセージを出して操作を終了します。
- WIDTH(n) が指定されず、計算された行の長さが 121 より小さいかまたは等しい場合、ICETOOL は 行の長さとして LRECL を 121 にセットします。
- WIDTH(n) が指定されず、計算された行の長さが 122 から 2048 の場合、ICETOOL は 行の長さとして LRECL を計算された行の長さにします。
- WIDTH(n) が指定されず、計算された行の長さが 2048 より大きい場合、ICETOOL はエラー・メッセージを出して操作を終了します。

LRECL を特定の値 (たとえば、同じデータ・セットの中に複数の報告書を組み込むために DISP=MOD を使用するような場合) にセットしなければならない場合、あるいは、報告書の行の長さが特定の最大値 (たとえば、133 バイト) を超えないようにしたい場合は、WIDTH(n) を使用してください。WIDTH(n) を使用したくない場合は、ICETOOL に計算を任せ、適切な行の長さとして LRECL をセットさせることができます。

### BREAK(p,m,f)

報告書をいくつかのセクションに分けるために、数字または文字段落フィールドを使用することを指定します。各一組の順次入力レコードが、指定された段落フィールドについて同じ値をもっている場合は、対応する組のデータ行は報告書の中でセクションとして扱われることとなります。SORT 演算子 (または別のアプリケーション) は、DISPLAY 演算子の前に指定する必要があります。SORT 演算子は段落フィールドと他の該当するフィールドを、報告書の中で希望の順序に分類します。

各セクションは新しいページから始まります。セクションの各ページには、セクションの段落値を示す段落タイトル行が含まれます。数字段落値は、正符号のときはブランク、負符号のときは -、および先行ゼロな

## DISPLAY 演算子

して印刷されます。BTITLE は、段落タイトル行に表示するストリングを指定するのに使用できます。段落値と段落タイトルのストリングは、ユーザーが BREAK と BTITLE をそれらを指定した順序で表示されます。段落タイトル要素間には、2 つの空白が表示されます。段落タイトル行の後に空白が 1 行印刷されます。

BTOTAL、BMAXIMUM、BMINIMUM、および BAVERAGE を使用して、各数値 ON フィールドについての段落統計を作成できます。たとえば、ON(5,3,ZD) についてのセクションの最大値および ON(22,2,BI) についてのセクションの最大値。セクションの段落統計は、それぞれのセクションの終わりに印刷されます (段落タイトルを含む 1 または複数のページ上に)。TOTAL、MAXIMUM、MINIMUM、および AVERAGE を使用して、各数値 ON フィールドについての全体の統計を作成できます。たとえば、ON(5,3,ZD) についての報告書の最大値および ON(22,2,BI) についての報告書の最大値。各セクションについての全体の統計は、報告書の最後に印刷されます (段落タイトルを含まない別のページに)。

**p** および **m** については、ON(p,m,f) を参照してください。

**f** は次のようにフィールドの形式を指定します。

CSF または FS 形式段落フィールドの場合は、以下の通りです。

- 最大 15 桁が使用できます。16 桁の値が見つかったら、ICETOOL はエラー・メッセージを出して操作を終了します。

ZD または PD 形式段落フィールドの場合は、以下の通りです。

- 無効な数字 (A~F) をもつ 10 進数値が見つかったら、ICETOOL はメッセージを出して操作を終了します。
- 符号が F、E、C、A、8、6、4、2、または 0 の場合、値は正の値として扱われます。
- 符号が D、B、9、7、5、3、または 1 の場合、値は負の値として扱われます。

DT1、DT2 または DT3 形式フィールドの場合、

- 無効な SMF 日付があると、データ例外 (0C7 ABEND)、または誤った ZD 日付が発生します。
- SMF 日付値は常に正の数値として扱われます。

TM1、TM2、TM3 または TM4 形式フィールドの場合、

- 無効な SMF 時間があると、誤った ZD 時間が発生します。
- SMF 時間値は常に正の数値として扱われます。

### **BREAK(p,m,f,formatting)**

報告書をセクションに分割するのに使用される数値フィールドまたは文字段落フィールドを指定し、また、このフィールドのデータの印刷に関してどのように形式設定するかを指定します。

詳細については、BREAK(p,m,f) を参照してください。

### **formatting**



レコード番号が印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。数値段落フィールドにはどんな形式設定項目も指定できますが、L'string' および T'string' に限り、文字段落フィールドに指定します。

**mask** mask については、ON(p,m,f,formatting) を参照してください。

#### **E'pattern'**

**E'pattern'** については、ON(p,m,f,formatting) を参照してください。

#### **L'string'**

**L'string'** については、ON(p,m,f,formatting) を参照してください。

#### **F'string'**

**F'string'** については、ON(p,m,f,formatting) を参照してください。

#### **T'string'**

**T'string'** については、ON(p,m,f,formatting) を参照してください。

**LZ** LZ については、ON(p,m,f,formatting) を参照してください。

#### **BTITLE('string')**

STRING を、セクションの各ページごとに印刷される段落タイトル行に表示するよう指示します。BTITLE は、BREAK が指定されている場合のみ指定できます。段落値と段落タイトルの STRING は、ユーザーが BREAK と BTITLE をそれらを指定した順序で表示されます。段落タイトル要素間には、2 つの空白が表示されます。段落タイトル行の後に空白が 1 行印刷されます。

STRING (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') を STRING の中に含める場合は、2 個の単一アポストロフィ (') を指定します。STRING の最初に空白があると、テキストは右に移動します。BTITLE が BREAK の前に指定されている場合に、STRING の終わりに空白があれば、STRING と段落値との間のスペースが増加します。

#### **BTOTAL('string')**

各セクションのデータ行の後に段落 TOTAL 行を印刷することを指定します。BTOTAL は、BREAK が指定されている場合のみ指定できます。指定された STRING は、段落の TOTAL 行のインデント列から開始され、その後、各数値データ列の段落合計が続きます。STATLEFT が指

## DISPLAY 演算子

定される場合、ストリングは最初のデータ列の左に印刷されます。合計は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。合計は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。段落 TOTAL 行の前に空白が 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (``) を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、BTOTAL('') を指定します。

各数値 ON フィールドの段落合計は、ユーザーが指定する形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの合計は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。合計は ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

デフォルトでは、列幅は最大で 1 つの符号と 15 桁を合計として印刷できるように調整されます。ON フィールドの段落の合計が 15 桁を超えると、ICETOOL はそのフィールドの段落の合計にアスタリスクを入れ、処理を終了します。

Ndd 形式設定項目を使用して、合計に使用する桁数を減らすことができます。ON フィールドの段落の合計が使用されている桁数を超えると、ICETOOL はそのフィールドの段落の合計にアスタリスクを入れ、処理を終了します。

BTOTAL、BMAXIMUM、BMINIMUM、および BAVERAGE 行はユーザーがそれらを指定した順序で印刷されます。

### **BMAXIMUM('string')**

各セクションのデータ行の後に段落 MAXIMUM 行を印刷することを指定します。BMAXIMUM は、BREAK が指定されている場合のみ指定できます。指定されたストリングを段落の MAXIMUM 行のインデント列から印刷し、その後、各数値データ列の段落の最大値が続きます。STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。最大値は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。最大値は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。段落 MAXIMUM 行の前に空白が 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (``) を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、BMAXIMUM('') を指定します。

各数値 ON フィールドの区分最大値は、ユーザーが指定する形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの最大値は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。最大値は、ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。



BTOTAL、BMAXIMUM、BMINIMUM、および BAVERAGE 行はユーザーがそれらを指定した順序で印刷されます。

#### **BMINIMUM('string')**

各セクションのデータ行の後に段落 MINIMUM 行を印刷することを指定します。BMINIMUM は、BREAK が指定されている場合のみ指定できます。指定されたストリングを段落の MINIMUM 行の インデント列から印刷し、その後に、各数値データ列の段落の最小値が続きます。STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。最小値は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。最小値は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。区分 MINIMUM 行の前に空白が 1 行印刷されます。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (') を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、BMINIMUM('') を指定します。

各数値 ON フィールドの段落最小値は、ユーザーが指定する形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの最小値は、NOST 形式設定項目がそのフィールドに指定される場合、抑制されます。最小値は、ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

BTOTAL、BMAXIMUM、BMINIMUM、および BAVERAGE 行はユーザーがそれらを指定した順序で印刷されます。

#### **BAVERAGE('string')**

各セクションのデータ行の後に段落 AVERAGE 行を印刷することを指定します。BAVERAGE は、BREAK が指定されている場合のみ指定できます。指定されたストリングを段落の AVERAGE 行の インデント列から印刷し、その後に、各数値データ列の段落の平均が続きます。STATLEFT が指定される場合、ストリングは最初のデータ列の左に印刷されます。平均は、ストリングとして同じ行に印刷されます。STATLEFT が指定されない場合、ストリングは最初のデータ列に印刷されます。平均は、ストリングとして同じ行に、またはアポストロフィとして次の行に印刷されます。段落 AVERAGE 行の前に空白が 1 行印刷されます。

段落平均 (つまり、算術平均) は、段落合計をセクション内の値の数で割り、最も近い整数になるように端数を切り捨てます (たとえば、 $23 / 5 = 4$ 、 $-23 / 5 = -4$ )。

ストリング (1 ~ 50 文字) は、単一アポストロフィで囲まれます。単一アポストロフィ (') をストリングの中を含める場合は、2 個の単一アポストロフィ (') を指定します。ストリングの印刷を抑制するには、2 つのアポストロフィを使用して、BAVERAGE('') を指定します。

各数値 ON フィールドの段落平均値は、ユーザーが指定する形式 (形式設定、PLUS、BLANK、または標準) で印刷されます。特定の数値フィールドの平均は、NOST 形式設定項目がそのフィールドに指定される場

## DISPLAY 演算子

合、抑制されます。平均値は、ON(VLEN) フィールドについては印刷されますが、ON(NUM) フィールドについては印刷されません。

ON フィールドの段落の合計が 15 桁を超えると、ICETOOL はそのフィールドの段落の平均にアスタリスクを入れ、処理を終了します。

BTOTAL、BMAXIMUM、BMINIMUM、および BAVERAGE 行はユーザーがそれらを指定した順序で印刷されます。

### STATLEFT

統計のストリング (TOTAL、MAXIMUM、MINIMUM、AVERAGE、BTOTAL、BMAXIMUM、BMINIMUM、BAVERAGE) が、最初のデータ列の左に表示されるように指定します (最初の列のストリングのデフォルトの表示を指定変更)。STATLEFT は常に、統計行をデータ列から目立たせて、各統計をそれぞれのストリングと同じ行に表示します。

### UZERO

-0 および +0 が符号なしゼロ値として扱われる、つまり、同じ値として処理されることを指定します。UZERO では、-0 および +0 は、ON、MINIMUM、MAXIMUM、BREAK、BMINIMUM、および BMAXIMUM 処理において正として扱われます。

UZERO は、-0 および +0 を符号付きのゼロ値として扱う、つまり、異なる値として処理するというデフォルトを指定変更します。UZERO を指定しない場合、ON、MINIMUM、MAXIMUM、BREAK、BMINIMUM、および BMAXIMUM 処理において、-0 は負として、+0 は正として扱われます。

## DISPLAY の例

下記の例の DISPLAY 演算子は、すべて単一の ICETOOL ジョブ・ステップに組み込むことができますが、わかりやすくするために、別々に示して説明しています。報告書形式を調整するための追加の例に関しては 510 ページの『OCCUR 演算子』を参照してください。

### 例 1

```
DISPLAY FROM(SOURCE) LIST(FIELDS) ON(NUM) ON(40,12,CH) -  
ON(20,8,PD)
```

FIELDS データ・セットに次のものを印刷します。

- 標準見出しを含むヘッダ行
- 次のものを含む標準形式のデータ行
  - 標準形式のレコード番号
  - SOURCE データ・セットの 40 ~ 51 桁目の文字
  - 標準形式の SOURCE データ・セットの 20 ~ 27 桁目のパック 10 進数値

FIELDS 出力は新しいページから始まり、以下のように表示されます (最初の 2 つのレコードは、図示のための値で示されています)。

```

RECORD NUMBER      (40,12,CH)          (20,8,PD)
0000000000000001  SAN JOSE              000000000003745
0000000000000002  MORGAN HILL          000000000016502
      .
      .
      .
    
```

ヘッダー行は各ページの一番上にあります。

## 例 2

```

DISPLAY FROM(IN) LIST(LIST1) -
  TITLE('National Accounting Report') -
  PAGE DATE TIME -
  HEADER('Division') HEADER('Revenue') HEADER('Profit/Loss') -
  ON(1,25,CH)          ON(45,10,ZD)          ON(35,10,ZD) -
  BLANK -
  TOTAL('Company Totals') -
  AVERAGE('Company Averages')
    
```

LIST1 データ・セットに次のものを印刷します。

- 指定したタイトル、ページ番号、日付および時間が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが入っている BLANK 形式のデータ行
  - IN データ・セットの 1 ~ 25 桁目の文字
  - IN データ・セットの 45 ~ 54 桁目のゾーン 10 進数値
  - IN データ・セットの 35 ~ 44 桁目のゾーン 10 進数値
- 指定したSTRING、および 2 つのゾーン 10 進数フィールドのそれぞれの合計が BLANK 形式で含まれている TOTAL 行
- 指定したSTRING、および 2 つのゾーン 10 進数フィールドのそれぞれの平均が BLANK 形式で含まれている AVERAGE 行

LIST1 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードは、図示のための値で示されています)。

Division	Revenue	Profit/Loss
National Accounting Report	- 1 -	10/21/92 18:52:44
-----	-----	-----
Research and Development	54323456	-823325
Manufacturing	159257631	1372610
.	.	.
.	.	.
.	.	.
Company Totals	612867321	5277836
Company Averages	76608415	659729

合計行および下線付きヘッダー行は各ページの一番上にあります。

例 3

```
DISPLAY FROM(DATA) LIST(JUSTDATA) -
NOHEADER -
ON(17,5,PD) ON(1,2,FI)
```

JUSTDATA データ・セットに次のものを印刷します。

- 次のものを含む標準形式のデータ行
  - 標準形式の DATA データ・セットの 17 ~ 21 桁目のパック 10 進数値
  - 標準形式の DATA データ・セットの 1 ~ 2 桁目の固定小数点値

JUSTDATA 出力にはページ替えまたはヘッダー行含まれず、次のようになります (最初の 2 つのレコードの値が図示されています)。

```
-0000000000273216 +0000000000000027
+0000000000993112 +0000000000000321
      .              .
      .              .
      .              .
```

例 4

```
COPY FROM(INPUT) TO(TEMP) USING(TREG)
DISPLAY FROM(TEMP) LIST(REGULAR) -
TITLE('Report on Regular Tools      ') PAGE -
HEADER(NONE) ON(1,18,CH) -
HEADER('Item') ON(35,5,CH) -
HEADER('Percent Change') ON(28,4,FS,B1) -
LINES(66)
COPY FROM(INPUT) TO(TEMP) USING(TPOW)
DISPLAY FROM(TEMP) LIST(POWER) -
TITLE('Report on Power Tools        ') PAGE -
HEADER(NONE) ON(1,18,CH) -
HEADER('Item') ON(35,5,CH) -
HEADER('Percent Change') ON(28,4,FS,B1) -
LINES(66)
```

この例は、異なるデータ・サブセットの報告書がどのように作成されるかを示しています。次のように仮定します。

- TREGCNTL データ・セットには、次のものが含まれています。
 

```
INCLUDE COND=(44,8,CH,EQ,C'Regular')
```
- TPOWCNTL データ・セットには、次のものが含まれています。
 

```
INCLUDE COND=(44,8,CH,EQ,C'Power')
```

最初の COPY 演算子は、44 ~ 51 桁目に 'Regular' が入っている INPUT データ・セットのレコードを TEMP (一時) データ・セットにコピーします。

最初の DISPLAY 演算子は、TEMP データ・セットのレコードの最初のサブセットを使用して、以下の REGULAR データ・セットに印刷します。

- 指定したタイトルおよびページ番号が含まれているタイトル行。TITLE スtring 後尾の余分なブランク、およびタイトル・String とページ番号の間の 8 個のブランクにより、ページ番号は右へ移動されます。

- 指定された下線付きヘッダーが含まれているヘッダー行 (最初の ON フィールドにはヘッダーなし)。
- 以下のものを含んでいるレコードの最初のサブセットのデータ行。
  - 1 ~ 18 桁目からの文字
  - 35 ~ 39 桁目からの文字
  - 1 桁の小数部および小数点としてのピリオドをもつように形式設定された 28 ~ 31 桁目からの浮動小数点符号値

2 番目の COPY 演算子は、44 ~ 51 桁目に 'Power' が入っている INPUT データ・セットのレコードを TEMP (一時) データ・セットにコピーします。

2 番目の DISPLAY 演算子は、TEMP データ・セットの中の 2 番目のレコードのサブセットを使用して、以下のような POWER データ・セットに印刷します。

- 指定したタイトルおよびページ番号が含まれているタイトル行。TITLE スtring 後尾の余分なブランク、およびタイトル・String とページ番号の間の 8 個のブランクにより、ページ番号は右へ移動されます。
- 指定された下線付きヘッダーが含まれているヘッダー行 (最初の ON フィールドにはヘッダーなし)。
- 次のものが含まれているレコードの 2 番目のサブセット
  - 1 ~ 18 桁目からの文字
  - 35 ~ 39 桁目からの文字
  - 1 桁の小数部および小数点としてのピリオドをもつように形式設定された 28 ~ 31 桁目からの浮動小数点符号値

REGULAR 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードが、図示のための値で示されています)。

Report on Regular Tools		- 1 -
	Item	Percent Change
	-----	-----
Hammers	10325	-7.3
Wrenches	00273	15.8
.	.	.
.	.	.
.	.	.

合計行および下線付きヘッダー行は各ページの一番上にあります。ページ当たり行数は 66 で、デフォルトの 58 を指定変更します。

POWER 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードが、図示のための値で示されています)。

Report on Power Tools		- 1 -
	Item	Percent Change
	-----	-----
Saws	31730	9.8
Drills	68321	123.0
.	.	.
.	.	.
.	.	.

## DISPLAY 演算子

合計行および下線付きヘッダー行は各ページの一番上にあります。ページ当たり行数は 66 で、デフォルトの 58 を指定変更します。

### 例 5

```
DISPLAY FROM(INV) LIST(RDWLIST1) -  
  TITLE('No Frills RDW Report') -  
  ON(NUM) -  
  ON(VLEN) -  
  ON(1,4,HEX) -  
  MINIMUM('Smallest') -  
  MAXIMUM('Largest')
```

RDWLIST1 データ・セットに次のものを印刷します。

- 指定したタイトルを含むタイトル行
- 標準見出しを含むヘッダー行
- 次のものを含む標準形式のデータ行
  - レコード番号
  - レコード長
  - 16 進数のレコード記述子ワード (RDW)
- 指定した文字列および標準形式の最小レコード長を含む MINIMUM 行
- 指定した文字列および標準形式の最大レコード長を含む MAXIMUM 行

RDWLIST1 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードが、図示のための値で示されています)。

```
No Frills RDW Report  
  
RECORD NUMBER      RECORD LENGTH      (1,4,HEX)  
0000000000000001  +000000000000075  004E0000  
0000000000000002  +000000000000071  00470000  
      .              . .  
      .              . .  
      .              . .  
  
Smallest            +000000000000058  
Largest             +000000000000078
```

タイトル行とヘッダー行は各ページの一番上にあります。

### 例 6

```
DISPLAY FROM(INV) LIST(RDWLIST2) -  
  DATE(DMY.) -  
  TITLE(' Fancy RDW Report ') -  
  TIME(12:) -  
  HEADER('Relative Record') ON(NUM) -  
  HEADER(' RDW (length)') ON(VLEN) -  
  HEADER('RDW (Hex)') ON(1,4,HEX) -  
  BLANK -  
  MINIMUM('Smallest Record in Variable Data Set:') -  
  MAXIMUM('Largest Record in Variable Data Set:')
```

RDWLIST2 データ・セットに次のものを印刷します。

- 日付、指定したタイトル、および時間が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが入っている BLANK 形式のデータ行
  - レコード番号
  - レコード長
  - 16 進数のレコード記述子ワード (RDW)
- 指定したストリングおよび BLANK 形式の最小レコード長を含む MINIMUM 行
- 指定したストリングおよび BLANK 形式の最大レコード長を含む MAXIMUM 行

RDWLIST2 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードは、図示のための値で示されています)。

Relative Record	RDW (length)	RDW (Hex)
1	75	004B0000
2	71	00470000
.	.	.
.	.	.
.	.	.
Smallest Record in Variable Data Set:		
	58	
Largest Record in Variable Data Set:		
	78	

合計行および下線付きヘッダー行は各ページの一番上にあります。

### 例 7

```

SORT FROM(PARTS) TO(TEMP) USING(SRT1)
DISPLAY FROM(TEMP) LIST(USA) -
  TITLE('Parts Completion Report for USA') DATE -
  HEADER('Part') HEADER('Completed') HEADER('Value ($)') -
  ON(15,6,CH) ON(3,4,ZD,A1) ON(38,8,ZD,C1) -
  TOTAL('Total:')
DISPLAY FROM(TEMP) LIST(FRANCE) -
  TITLE('Parts Completion Report for France') DATE(DM4/) -
  HEADER('Part') HEADER('Completed') HEADER('Value (F)') -
  ON(15,6,CH) ON(3,4,ZD,A3) ON(38,8,ZD,C3) -
  TOTAL('Total:')
DISPLAY FROM(TEMP) LIST(DENMARK) -
  TITLE('Parts Completion Report for Denmark') DATE(DMY-) -
  HEADER('Part') HEADER('Completed') HEADER('Value (kr)') -
  ON(15,6,CH) ON(3,4,ZD,A2) ON(38,8,ZD,C2) -
  TOTAL('Total:')
    
```

この例は、異なる 3 つの国の報告書がどのように作成されるかを示しています。各報告書の違いは、日付と数値の表示方法のみです。

SRTICNTL データ・セットに次のものが含まれていると仮定します。

```
SORT FIELDS=(15,6,CH,A)
```

## DISPLAY 演算子

SORT 演算子は、SRTICNTL の SORT ステートメントを使用して、PARTS データ・セットを TEMP データ・セットに分類します。

最初の DISPLAY 演算子は、TEMP データ・セットの中の分類済みレコードを使用して、USA データ・セットに次のものを印刷します。

- 指定したタイトルと、米国で普通に使用される形式の日付が含まれているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが含まれているデータ行
  - 15 ~ 20 桁目からの文字
  - 米国で普通に使用される区切り文字で形式設定された 3 ~ 6 桁目からのゾーン 10 進数
  - 2 桁の小数部および、米国で普通に使用される区切り文字と小数点で形式設定された 38 ~ 45 桁目からのゾーン 10 進数値
- 指定したストリングおよび、データ値と同じように形式設定された 2 つのゾーン 10 進数フィールドのそれぞれの合計が入っている TOTAL 行

2 番目の DISPLAY 演算子は、TEMP データ・セットの中の分類済みレコードを使用して、次のものを FRANCE データ・セットに印刷します。

- 指定したタイトルおよび、フランスで普通に使用される形式の日付が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが含まれているデータ行
  - 15 ~ 20 桁目からの文字
  - フランスで普通に使用される区切り文字で形式設定された 3 ~ 6 桁目からのゾーン 10 進数値
  - 2 桁の小数部および、フランスで普通に使用される区切り文字と小数点で形式設定された 38 ~ 45 桁目からのゾーン 10 進数値
- 指定したストリングおよび、データ値と同じように形式設定された 2 つのゾーン 10 進数フィールドのそれぞれの合計が入っている TOTAL 行

3 番目の DISPLAY 演算子は、TEMP データ・セットの中の分類済みレコードを使用して、次のものを DENMARK データ・セットに印刷します。

- 指定したタイトルおよび、デンマークで普通に使用される形式の日付が含まれているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが含まれているデータ行
  - 15 ~ 20 桁目からの文字
  - デンマークで普通に使用される区切り文字で形式設定された 3 ~ 6 桁目からのゾーン 10 進数値
  - 2 桁の小数部および、デンマークで普通に使用される区切り文字と小数点で形式設定された 38 ~ 45 桁目からのゾーン 10 進数値
- 指定したストリングおよび、データ値と同じように形式設定された 2 つのゾーン 10 進数フィールドのそれぞれの合計が入っている TOTAL 行



USA 出力は新しいページから始まり、次のようになります (いくつかのレコードが、図示のための値で示されています)。

Parts Completion Report for USA		01/14/95
Part	Completed	Value (\$)
000310	562	8,317.53
001184	1,234	23,456.78
029633	35	642.10
192199	3,150	121,934.65
821356	233	2,212.34
Total:	5,214	156,563.40

合計行および下線付きヘッダー行は各ページの一番上にあります。

FRANCE 出力は新しいページから始まり、次のようになります (いくつかのレコードが、図示のための値で示されています)。

Parts Completion Report for France		14/01/1995
Part	Completed	Value (F)
000310	562	8 317,53
001184	1 234	23 456,78
029633	35	642,10
192199	3 150	121 934,65
821356	233	2 212,34
Total:	5 214	156 563,40

合計行および下線付きヘッダー行は各ページの一番上にあります。

DENMARK 出力は新しいページから始まり、次のようになります (いくつかのレコードが、図示のための値で示されています)。

Parts Completion Report for Denmark		14-01-95
Part	Completed	Value (kr)
000310	562	8.317,53
001184	1.234	23.456,78
029633	35	642,10
192199	3.150	121.934,65
821356	233	2.212,34
Total:	5.214	156.563,40

合計行および下線付きヘッダー行は各ページの一番上にあります。

例 8

```

SORT FROM(DATA) TO(TEMP) USING(SRTX)
DISPLAY FROM(TEMP) LIST(WEST) -
  DATE TITLE('Western Region Profit/Loss Report') PAGE -
  BTITLE('Division:') BREAK(3,10,CH) -
  HEADER('Branch Office') ON(16,13,CH) -
  HEADER('Profit/Loss (K)') ON(41,4,PD,/K,E1) -
  BMINIMUM('Lowest Profit/Loss in this Division:') -
  BMAXIMUM('Highest Profit/Loss in this Division:') -
  BAVERAGE('Average Profit/Loss for this Division:') -
  MINIMUM('Lowest Profit/Loss for all Divisions:') -
  MAXIMUM('Highest Profit/Loss for all Divisions:') -
  AVERAGE('Average Profit/Loss for all Divisions:')

```

この例は、いくつかのセクションをもつ報告書がどのように作成されるかを示しています。

SRTXCNTL データ・セットに次のものが含まれていると仮定します。

```
SORT FIELDS=(3,10,A,16,13,A),FORMAT=CH
```

SORT 演算子は、SRTXCNTL の SORT ステートメントを使用して、DATA データ・セットを TEMP データ・セットに分類します。

DISPLAY 演算子は、TEMP データ・セットの分類済みレコードを使用して、各セクションを以下のものと一緒に WEST データ・セットに印刷します。

- 日付、指定したタイトル・ストリング、およびページ番号が入っているタイトル行
- 指定した段落タイトル・ストリング、および 3 ~ 12 桁目からの段落フィールド文字が入っている段落タイトル
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが含まれているデータ行
  - 16 ~ 28 桁目からの文字
  - 指定した通りに区切り文字と符号で形式設定された 41 ~ 44 桁目からのパック 10 進数値を 1000 で割った値
- データ値と同じように形式設定された、指定ストリングとこのセクションのパック 10 進数フィールド値の統計が含まれている段落 MINIMUM、MAXIMUM、および AVERAGE 行

報告書の最後のページには、次のものが含まれています。

- 日付、指定したタイトル・ストリング、およびページ番号が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- データ値と同じように形式設定された、指定ストリングとこのセクションのパック 10 進数フィールド値の統計が含まれている全体の MINIMUM、MAXIMUM、および AVERAGE 行

WEST 出力の最初のセクションは新しいページから始まり、次のようになります(いくつかのレコードが、図示のための値で示されています)。

01/14/95 Western Region Profit/Loss Report - 1 -

Division: Chips

Branch Office	Profit/Loss (K)
-----	-----
Gilroy	3,293
Los Angeles	(141)
Morgan Hill	213
Oakland	1,067
San Francisco	(31)
San Jose	92
San Martin	1,535

Lowest Profit/Loss in this Division:  
(141)

Highest Profit/Loss in this Division:  
3,293

Average Profit/Loss for this Division:  
861

タイトル行、段落タイトル行、および下線付きヘッダー行が、セクションの各ページの最上部に表示されます。

WEST 出力の 2 番目のセクションは新しいページから始まり、次のようになります(いくつかのレコードが、図示のための値で示されています)。

01/14/95 Western Region Profit/Loss Report - 2 -

Division: Ice Cream

Branch Office	Profit/Loss (K)
-----	-----
Marin	673
Napa	95
San Francisco	(321)
San Jose	2,318
San Martin	21

Lowest Profit/Loss in this Division:  
(321)

Highest Profit/Loss in this Division:  
2,318

Average Profit/Loss for this Division:  
557

タイトル行、段落タイトル行、および下線付きヘッダー行が、セクションの各ページの最上部に表示されます。

WEST 出力の最後のページは、新しいページで始まり次のように表示されます。

```
01/15/95      Western Region Profit/Loss Report      - 3 -
Branch Office  Profit/Loss (K)
-----
Lowest Profit/Loss for all Divisions:
                    (321)
Highest Profit/Loss for all Divisions:
                    3,293
Average Profit/Loss for all Divisions:
                    734
```

## 例 9

```
MODE CONTINUE
VERIFY FROM(CHECK) ON(2,3,PD) LIMIT(500)
DISPLAY FROM(CHECK) LIST(PDREPORT) BLANK LIMIT(500) -
  HEADER('Relative Record') ON(NUM) -
  HEADER('Numeric') ON(2,3,PD) -
  HEADER('Hexadecimal') ON(2,3,HEX) -
  HEADER('Associated Field') ON(21,20,CH)
```

この例は、無効な 10 進数値を含む各レコードをどのようにして識別できるのか、つまり、その相対レコード番号で識別できるのか、あるいはレコード内の関連するフィールドで識別できるのかを示しています。

MODE 演算子は、VERIFY 演算子が無効な 10 進数値を識別した場合に、DISPLAY 演算子が処理されるようにします。

VERIFY 演算子は、CHECK データ・セットの 2 ～ 4 桁目のパック 10 進数値の中に無効な文字 (A ～ F) と無効な数字 (0 ～ 9) がないかどうか調べます。無効な数字または符号を含む値 (もしあれば) には、TOOLMSG データ・セットにメッセージ ICE618A が印刷されます。500 個の無効値が検出された場合は、操作は終了します。

DISPLAY 演算子は、CHECK データ・セットの 2 ～ 4 桁目のパック 10 進数の中に無効な文字 (A ～ F) 含まれていないかどうかを調べます。無効な桁を含む値 (もしあれば) に対しては、TOOLMSG データ・セットにメッセージ ICE618A が印刷されます。500 個の無効値が検出された場合は、操作は終了します。無効な符号について調べる必要がある場合は、VERIFY 演算子を使用する必要があります。DISPLAY 演算子は、無効な数字のみ調べるためです。符号を調べる必要がない場合は、VERIFY 演算子は必要ありません。

DISPLAY 演算子は、次のものも PDREPORT データ・セットに印刷します。

- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが入っている BLANK 形式のデータ行
  - 相対レコード番号。この番号と、ICE618A メッセージに印刷されている RECORD 番号とを突き合わせて、無効な符号をもつレコードを検出できません。
  - 2 ～ 4 桁目のパック 10 進数の数値表現。識別しやすいように、無効な桁にはアスタリスクが示されます。無効な符号をもつ数値には、アスタリスクは示

されません。無効な符号をもつ数値は、相対レコード番号と ICE618A のレコード番号とを突き合わせるにより識別する必要があります。

- 2 ~ 4 桁目のバック 10 進数値の 16 進表示 (ICE618A にも示されています)。これにより、無効な特定の 16 進数や符号を検出するのが容易になります。
- 21 ~ 40 桁目からの文字。このような関連するフィールドを使用して、無効な値をもつレコードをより容易に識別できます。

VERIFY 演算子に対する TOOLMSG の ICE618A メッセージは次のとおりです。

```
ICE618A 0 INVALID (2,3,PD)      VALUE - RECORD: 0000000000000003,
      HEX VALUE 53A54C
ICE618A 0 INVALID (2,3,PD)      VALUE - RECORD: 0000000000000012,
      HEX VALUE 621540
ICE618A 0 INVALID (2,3,PD)      VALUE - RECORD: 0000000000000019,
      HEX VALUE 400F3C
```

DISPLAY 演算子に対する TOOLMSG の ICE618A メッセージは次のとおりです。

```
ICE618A 0 INVALID (2,3,PD)      VALUE - RECORD: 0000000000000003,
      HEX VALUE 53A54C
ICE618A 0 INVALID (2,3,PD)      VALUE - RECORD: 0000000000000019,
      HEX VALUE 400F3C
```

PDREPORT 出力は次のように表示されます。

Relative Record	Numeric	Hexadecimal	Associated Field
1	18600	18600C	Wagar
2	-93	00093B	Gellai
3	*****	53A54C	Giulianelli
4	86399	86399C	Mehta
5	24215	24215F	Johnson
6	8351	08351C	Packer
7	19003	19003C	Childers
8	-31285	31285D	Burg
9	88316	88316C	Monkman
10	1860	01860C	Veizinaw
11	-29285	29285D	Mead
12	62154	621540	Wu
13	-328	00328D	Madrid
14	-11010	11010D	Warren
15	1363	01363F	Burt
16	92132	92132C	Mao
17	-48500	48500D	Shen
18	-55	00055D	Yamamoto-Smith
19	*****	400F3C	Yaeger
20	33218	33218C	Leung
21	96031	96031C	Kaspar

PDREPORT を ICE618A メッセージと一緒に使用して、以下のことを識別できます。

- レコード 3 が、無効な数字 A および関連するフィールド『Giulianelli』をもっている
- レコード 12 が、無効な符号 0 および関連するフィールド『Wu』をもっている
- レコード 19 が、無効な数字 F および関連するフィールド『Yaeger』をもっている

例 10

```
COPY FROM(IN) USING(OUTF)
DISPLAY FROM(TEMP) LIST(EMPCT) BLANK -
    TITLE('Employees by Function') -
    DATE -
    HEADER('Function') HEADER('Employees') -
    ON(1,25,CH)          ON(30,4,ZD)
```

この例は、ICETOOL 報告書の暗号値を意味のある語句に置き換えるのに、OUTFIL 表索引機能をどのように使用するかを示しています。次のように仮定します。

- OUTFCNTL データ・セットには、次のものが含まれています。

```
OUTFIL FNAMES=TEMP,
    OUTREC=(1:9,2,CHANGE=(25,
        C'MN',C'Manufacturing',
        C'RD',C'Research and Development',
        C'FN',C'Finance',
        C'MR',C'Marketing',
        C'IS',C'Information Systems'),
    30:4,4)
```

COPY 演算子は、OUTFCNTL の OUTFIL ステートメントを使用して、IN データ・セット・レコードを TEMP (一時) データ・セットに再フォーマット設定します。DISPLAY 演算子で使用するために、次の 2 つのフィールドが抽出されます。

- 表索引機能を使用して、9 ~ 10 桁目の 2 文字の部門コードを、1 ~ 25 桁目の 25 文字の名前に変更します。
- 4 ~ 7 桁目のゾーン 10 進数値を 30 ~ 33 桁目に移動します。

DISPLAY 演算子は、TEMP データ・セットの再フォーマット設定されたフィールドを使用して、次のものを EMPCT データ・セットに印刷します。

- 指定したタイトルと日付が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが入っている BLANK 形式のデータ行
  - 部門コードで置き換えられた 1 ~ 25 桁目からの名前
  - 30 ~ 33 桁目からのゾーン 10 進数値

EMPCT 出力は、新しいページで始まり次のように表示されます。

Function	Employees
-----	-----
Manufacturing	486
Marketing	21
Research and Development	55
Information Systems	123
Finance	33

例 11

```

DISPLAY FROM(ACCTS) LIST(PLAIN) -
  TITLE('Accounts Report for First Quarter') -
  DATE(MD4/) BLANK -
  HEADER('Amount') ON(12,6,ZD) -
  HEADER(Id') ON(NUM) -
  HEADER('Acct#') ON(31,3,PD) -
  HEADER('Date') ON(1,4,ZD) -
  TOTAL('Total for Q1') -
  AVERAGE('Average for Q1')

DISPLAY FROM(ACCTS) LIST(FANCY) -
  TITLE('Accounts Report for First Quarter') -
  DATE(MD4/) BLANK -
  HEADER('Amount') ON(12,6,ZD,C1,N08) -
  HEADER(Id') ON(NUM,N02) -
  HEADER('Acct#') ON(31,3,PD,N0ST,LZ) -
  HEADER('Date') ON(1,4,ZD,E'99/99',N0ST) -
  INDENT(2) BETWEEN(5) -
  STATLEFT -
  TOTAL('Total for Q1') -
  AVERAGE('Average for Q1')
    
```

この例は、DISPLAY 報告書の表示を改善するために使用可能な、いくつかのオプションを示しています。最初の DISPLAY 演算子は、「簡易 (PLAIN)」報告書を作成し、2 番目の DISPLAY 演算子は、太字で表示されるオプションを使用し、「詳細 (FANCY)」報告書を作成します。

PLAIN 出力は、新しいページで始まり次のように表示されます。

Accounts Report for First Quarter		05/04/2001	
Amount	Id	Acct#	Date
93271	1	15932	106
137622	2	187	128
83147	3	15932	212
183261	4	2158	217
76389	5	187	305
920013	6	15932	319
Total for Q1			
1493703		50328	1287
Average for Q1			
248950		8388	214

FANCY 出力は、新しいページで始まり次のように表示されます。

Accounts Report for First Quarter		05/04/2001		
Amount	Id	Acct#	Date	
-----	---	-----	-----	
932.71	1	15932	01/06	
1,376.22	2	00187	01/28	
831.47	3	15932	02/12	
1,832.61	4	02158	02/17	
763.89	5	00187	03/05	
9,200.13	6	15932	03/19	
Total for Q1	14,937.03			
Average for Q1	2,489.50			

“FANCY” 報告書で使用するその他のオプションについて、以下に説明します。

- 最初の ON フィールド: PLAIN 報告書では、BLANK は ICETOOL を作成し、6 バイトの ZD 値を、先行ゼロが抑制された編集されていない桁として印刷します。ただしこの例では、ドルおよびセントが実際に表示されることが確認できます。そのため FANCY 報告書では、C1 形式設定項目 (使用可能な 33 のマスクの 1 つ) を使用し、3 桁ごとの区切りとしてコンマ (,) を、小数点として ピリオド (.) を含んだ値を印刷します。

PLAIN 報告書では、値の表示に何桁必要か不明なため、TOTAL は ICETOOL を作成し、値を 15 桁で表示可能にします。ただしこの例では、合計値が 8 桁を超えないことが確認できます。そのため FANCY 報告書では、N08 形式設定項目を使用し、桁数を 8 に設定します。これにより、フィールドの列幅を縮小できます。

- 2 番目の ON フィールド: PLAIN 報告書では、レコード番号の表示に何桁必要か不明なため、NUM は ICETOOL を作成し、レコード番号を 15 桁で表示可能にします。ただしこの例では、レコード数が 99 を超えないことが確認できます。そのため FANCY 報告書では、N02 形式設定項目を使用し、桁数を 2 に設定します。これにより、レコード番号の列幅を縮小できます。
- 3 番目の ON field: PLAIN 報告書では、TOTAL および AVERAGE は ICETOOL を作成し、この 3 バイトの PD フィールドの合計および平均を印刷します。ただしこの例では、フィールドがアカウント番号であるため、統計を必要としません。そのため FANCY 報告書では、NOST 形式設定項目を使用し、このフィールドの統計を抑制します。

PLAIN 報告書では、A0 のデフォルトのマスクは ICETOOL を作成し、この 3 バイトの PD フィールドの先行ゼロを抑制します。ただしこの例では、このフィールドがアカウント番号であるため、先行ゼロを表示させる必要があります。そのため FANCY 報告書では、LZ 形式設定項目を使用し、このフィールドの先行ゼロを印刷します。

- 4 番目の ON フィールド: PLAIN 報告書では、BLANK は ICETOOL を作成し、4 バイトの ZD 値を、先行ゼロが抑制された編集されていない桁として印刷します。ただしこの例では、その桁が日付 (月および日) を表示することが確認できます。そのため FANCY 報告書では、E'99/99' 形式設定項目を使用し、先行ゼロおよび月と日の間にスラッシュ (/) を含んだ値を印刷します。

PLAIN 報告書では、TOTAL および AVERAGE は、ICETOOL を作成し、この 4 バイトの ZD フィールドの合計および平均を印刷します。ただしこの例では、



フィールドが日付であるため、合計または平均を必要としません。そのため FANCY 報告書では、**NOST** 形式設定項目を使用し、このフィールドの統計を抑制します。

**注:** アプリケーションにより、E'pattern' で表示される日付の最小値および最大値が必要となります。その場合は、日付フィールドに **NOST** は指定しません。

- **INDENT: PLAIN** 報告書では、ICETOOL はデフォルトで、2 列目 (制御文字の後) から報告書を開始します。ただしこの例では、報告書を少し字下げする必要があります。そのため FANCY 報告書では、**INDENT(2)** オペランドを使用し、報告書をブランク 2 個分字下げします。したがって、報告書は 4 列目から開始されます。
- **BETWEEN: PLAIN** 報告書では、ICETOOL はデフォルトで、データ列とデータ列の間に 3 個のブランクを使用します。ただしこの例では、列と列の間により多くのスペースが必要です。そのため FANCY 報告書では、**BETWEEN(5)** オペランドを使用し、ブランク 2 個を列と列の間に挿入します。
- **STATLEFT: PLAIN** 報告書では、ICETOOL はデフォルトで、TOTAL および AVERAGE のストリングを最初のデータ列の下に印刷します。また、それぞれの統計ごとに 2 つの行を使用し、ストリングが値と重なり合うのを回避します。ただしこの例では、TOTAL および AVERAGE ストリングを報告書上で目立たせ、さらに各ストリングをそれぞれの値と同じ行に表示する必要があります。そのため FANCY 報告書では、**STATLEFT** オペランドを使用し、最初のデータ列の左側に TOTAL および AVERAGE ストリングを印刷します。

## 例 12

```

SORT FROM(RAWSMF) TO(SMF14) USING(SMFI)
DISPLAY FROM(SMF14) LIST(SMF14RPT) -
    TITLE('SMF Type-14 Records') DATE(4MD/) -
    HEADER('Date') ON(11,4,DT1,E'9999/99/99') -
    HEADER('Time') ON(7,4,TM1,E'99:99:99') -
    HEADER('Sys') ON(15,4,CH) -
    HEADER('Jobname') ON(19,8,CH) -
    HEADER('Datasetname') ON(69,44,CH)
    
```

この例は、SMF 日付値および時間値を、SMF 14 タイプのレコードの報告書に分かりやすく表示する方法について示しています。

**SORT** 演算子は、RAWSMF データ・セットからタイプ 14 のレコードを選択し、SMF14 データ・セットに日付および時間で分類します。SMFICNTL で以下の制御ステートメントを使用します。

```

INCLUDE COND=(6,1,BI,EQ,14)
SORT FIELDS=(11,4,PD,A,7,4,BI,A)
    
```

**DISPLAY** 演算子は、SMF14 のタイプ 14 のレコードを SMF14RPT データ・セットで使用して印刷します。

- 指定したタイトルと日付が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものが含まれているデータ行
  - C'yyyy/mm/dd' で表示される、11 ~ 14 桁目の SMF 日付値

## DISPLAY 演算子

- C'hh:mm:ss' で表示される、7 ~ 10 桁目の SMF 時間値
- 15 ~ 18 桁目の文字値
- 19 ~ 26 桁目の文字値
- 69 ~ 112 桁目の文字値

SMF14RPT 出力は、新しいページで始まり次のように表示されます。

SMF Type-14 Records		2001/04/24		
Date	Time	Sys	Jobname	Datasetname
2001/04/20	06:03:15	ID03	JOB00003	SYS1.QRS
2001/04/20	10:03:22	ID02	JOB00002	SYS1.XYZ
2001/04/21	14:05:37	ID03	JOB00004	SYS1.MNO
2001/04/21	22:11:00	ID03	JOB00005	SYS1.MNO
2001/04/24	00:00:08	ID03	JOB00006	SYS1.MNO

**注:** SMF 日付形式 (DTn) または SMF 時間形式 (TMn) を使用する場合、SMF 値は数値として扱われます。これにより、マスクおよびパターンなど、数値の形式設定項目を使用して、SMF 値を編集できます。DTn および TMn ヘッダーはデフォルトで、上記の SMF14RPT 出力の例に示すように、他の数値のヘッダーと同様、右寄せで表示されます。数値のヘッダーを中央そろえまたは左寄せにする場合、HEADER('string') の末尾に適切な数のブランクを追加できます。

たとえば、SMF 日付のヘッダーを左寄せにする場合、以下のように 6 個のブランクをヘッダーのストリングの末尾に使用できます。

```
HEADER('Date      ')
```

上記により、以下のヘッダーを作成できます。

```
Date
-----
```

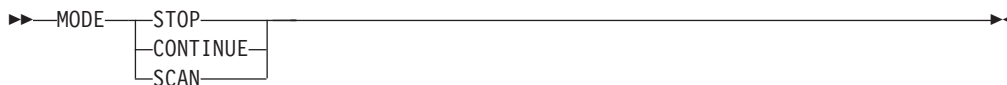
たとえば、SMF 日付のヘッダーを中央そろえにする場合、以下のように 3 個のブランクをヘッダーのストリングの末尾に使用できます。

```
HEADER('Date   ')
```

上記により、以下のヘッダーを作成できます。

```
Date
-----
```

## MODE 演算子



エラー検査とエラー検出後の処置を制御するための 3 つのモードのうちの 1 つを指定します。MODE 演算子は、後続の演算子から次の MODE 演算子 (ある場合) までの『処理』(つまり、ICETOOL ステートメントのエラー検査および DFSORT 呼び出し) に影響を与えます。MODE 演算子を使用すれば、演算子のグループまたはすべての演算子に対して次のことを行うことができます。

- 12 または 16 の戻りコードの後、演算子の処理を停止または続行する。12 または 16 の戻りコードは、ICETOOL または DFSORT によりステートメントまたは実行時エラーが検出された結果、セットされる場合があります。
- ICETOOL ステートメントにエラーがないかどうかを調べるが、DFSORT を呼び出さない。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### STOP

12 または 16 の戻りコードがセットされた場合に、後続の操作を停止します。ある演算子についてエラーが検出されると、SCAN が自動的に有効としてセットされます。DFSORT は後続の演算子では呼び出されませんが、エラーに対する ICETOOL ステートメントの検査は続行されます。

STOP モードを使用して、従属する演算子をグループ化できます (つまり、ある演算子が失敗した場合、残りの演算子を処理しません)。

STOP モードは、ICETOOL の開始時に自動的に有効としてセットされます。

### CONTINUE

12 または 16 の戻りコードがセットされたかどうかにかかわらず、後続の操作が続行されます。ある演算子がエラーになっても、その後の演算子の処理は続行されます。

CONTINUE モードを使用して、独立の演算子をグループ化できます (つまり、他の演算子が成功するか失敗するかにかかわらず、各演算子処理します)。

### SCAN

ICETOOL ステートメントはエラー検査されますが、DFSORT は呼び出されません。

SCAN モードを使用して、ICETOOL ステートメントのエラー検査を行うことができます。

注: STOP モードのときにエラーが検出されると、SCAN モードが自動的にセットされます。

## MODE の例

```
MODE SCAN
  RANGE ...
  UNIQUE ...
MODE STOP
  VERIFY ...
  DISPLAY ...
MODE CONTINUE
  COPY ...
  SORT ...
  STATS ...
```

SCAN モード: RANGE および UNIQUE はステートメント・エラーが検査されますが、DFSORT は呼び出されません。

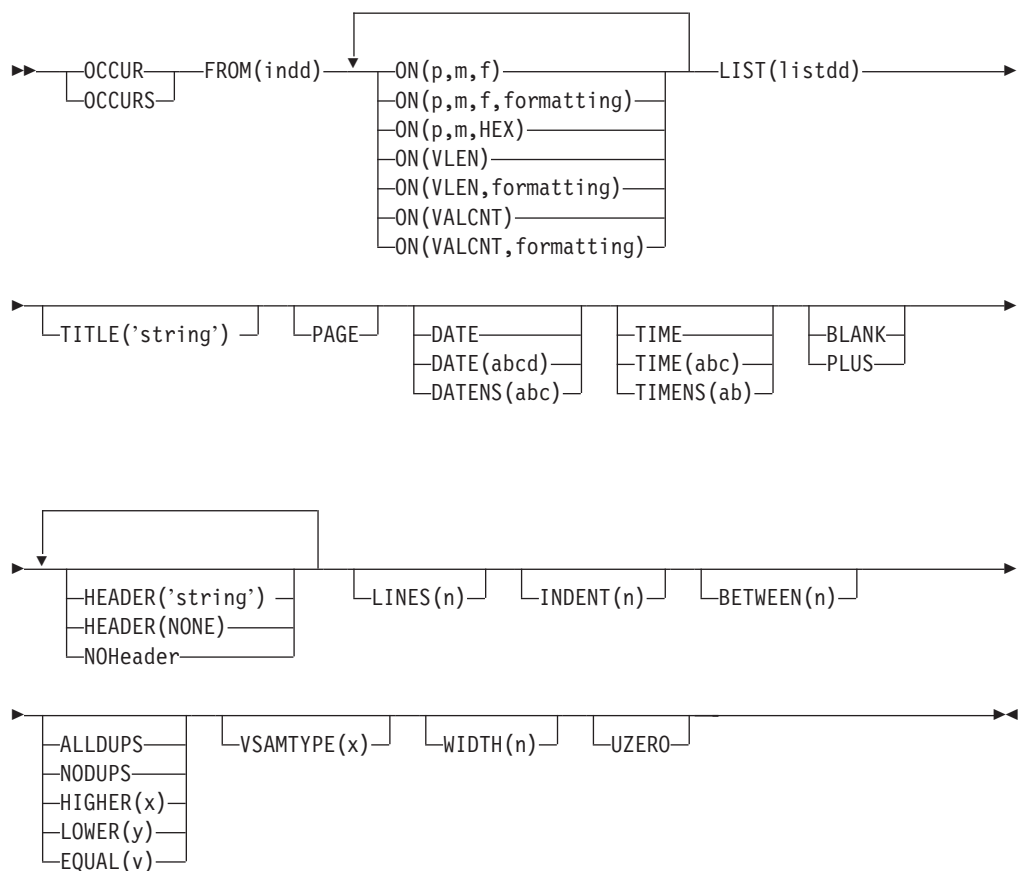
## MODE 演算子

STOP モード: DISPLAY は VERIFY に依存します。VERIFY の戻りコードが 12 または 16 であれば、SCAN モードになります。この場合、DISPLAY のステートメント・エラーが検査されますが、DFSORT は呼び出されません。

CONTINUE モード: COPY、SORT、および STATS は互いに独立しています。COPY の戻りコードが 12 または 16 であっても、SORT は処理されます。COPY または SORT の戻りコードが 12 でも 16 であっても、STATS は処理されます。

一方の演算子グループの戻りコードは、他方の演算子グループに影響を与えないことに注意してください。

## OCCUR 演算子



| 指定された数値フィールド (SMF 日付または時間を含む) または文字フィールドの  
 | 各固有値と、それが発生した回数を別個のデータ・セット内に印刷します。簡単な  
 | 報告書または調整済み報告書を作成できます。印刷する値を、値のカウントが指定  
 | 基準を満たすものだけに限定できます。

1 ~ 10 個のフィールドを指定できますが、その結果のリスト・データ・セット行の長さは、WIDTH オペランドで指定された限度、あるいは WIDTH が指定されていない場合 2048 バイトを超えてはなりません。少なくとも 1 つの ON(VLEN) ま

たは ON(p,m,f) フィールドを指定します。このような指定済みのすべての ON フィールドは、レコードが固有な値を含んでいるかどうかを判別するために使用されます。固有な値のそれぞれについて、単一のリスト・データ・セット・レコードが印刷されます。ON(VALCNT) が指定されている場合は、“値カウント” (すなわち、ON 値が発生した回数) が他の ON 値とともにリスト・データ・セットに印刷されません。

出力データ列をそれより小さいフィールドに“圧縮”できる形式設定項目または **PLUS** または **BLANK** オペランドを指定すると、プリンターまたはディスプレイがサポートする文字幅により行の長さが制限されている場合に、ユーザー報告書に最大 10 個までのより多くのフィールドを含めることができます。

ALLDUPS、NODUPS、HIGHER(x)、LOWER(y)、または EQUAL(v) を指定して、印刷する ON 値を、値カウントが指定基準を満たすものだけに限定できます (たとえば、重複値だけの ALLDUPS)。デフォルトの基準は HIGHER(0) で、各固有値ごとに ON 値が印刷されることになります。

DFSORT が呼び出され、indd データ・セットを ICETOOL の E35 ユーザー出口に分類します。ICETOOL はその E35 出口を使用して、適切なタイトル、ヘッダー、およびデータをリスト・データ・セットに印刷します。

ユーザー独自の DFSORT MODS、INREC、OUTREC、SUM、または RECORD ステートメントを指定できません。なぜなら、これらのステートメントは、ICETOOL がこの演算子のために渡した DFSORT ステートメントを指定変更するからです。

ワークスペースを分類処理で必ず使用できるようにするために、DYNALLOC オプションが DFSORT に渡されます。動的割り振りのためのインストール・デフォルトが OCCUR 演算子に適切でない場合は、次のいずれかの処置をとることができます。

1. 次のような OPTION 制御ステートメントを使用して、DYNALLOC オプションを指定変更する。

```
OPTION DYNALLOC=(3390,5)
```

これは、DFSPARM データ・セット内にあります (すべての OCCUR、SELECT、SORT、および UNIQUE 演算子に適用されます)。

2. SORTWKdd DD ステートメントを使用して、動的割り振りを指定変更する (すべての OCCUR、SELECT、および UNIQUE 演算子に適用されます)。詳細については、79 ページの『SORTWKdd DD ステートメント』を参照してください。

ICETOOL でテープ作業データ・セットを使用することはできません。

## 簡単な報告書

必要なオペランドを指定して、簡単な報告書を作成できます。たとえば、FROM および LIST オペランド指定し (また 10 バイト文字フィールドと 7 バイトにゾーン 10 進数フィールドに対しては ON オペランド)、および値カウントを指定すれば、リスト・データ・セット内の出力を次のように表示できます。

## OCCUR 演算子

(p,m,f) characters	(p,m,f) sdddddddddddddd	VALUE COUNT dddddddddddddd
.	.	.
.	.	.
.	.	.

制御バイトは、各リスト・データ・セットの最初のバイトを占有します。左寄せされた標準ヘッダーが各ページの一番上に印刷され、各列の内容を示します。またヘッダーの後には、そのレコードのフィールド内の文字と数値、および指定した値の発生数 (値カウント) を表示した行が、各レコードごとに 1 行ずつ印刷されます。

フィールドは、OCCUR ステートメントで指定された順序と同じ順序で列に印刷されます。すべてのフィールドは左寄せされます。数値フィールドの場合、先行ゼロが印刷されます。負符号には -、正符号には + が使用されます。値カウントでは、先行ゼロが印刷されます。

デフォルトでは、データの最初の列は制御文字の直後から開始され、列の間には 3 つの空白が入ります。INDENT オペランドを使用して、データの最初の列の前に挿入される空白の数を変更します。BETWEEN オペランドを使用して、列の間の空白の数を変更します。

標準の列の幅は次のとおりです。

- 文字データ: 文字フィールドの長さ、または 20 バイト (フィールド長が 20 バイト以下の場合)
- 数値データ: 16 バイト
- 値カウント: 15 バイト

HEADER オペランドを使用して、ヘッダーを変更したり抑制できます。PLUS または BLANK オペランドを使用して、数値フィールドの形式を変更できます。

PLUS、BLANK、および HEADER オペランドを使用して、数値フィールドおよび文字フィールドの列の幅や、ヘッダーおよびフィールドの位置調整を変更できます。

NOHEADER オペランドを使用して、データ・レコードだけを含むリスト・データ・セットを作成できます。このようにして作成されたデータ・セットはさらに、CH (文字値の場合) または CS/FS (数値の場合。値カウントを含む) を使用する他の演算子 (たとえば、STATS または UNIQUE) により処理できます。

## 調整済み報告書

タイトル、日付、時刻、ページ番号、ヘッダー、ページ当たり行数、およびフィールド形式を制御する各種のオペランドを使用して、リスト・データ・セット内の出力を調整できます。任意指定のオペランドをさまざまな組み合わせで使用して、各種の報告書形式を作成できます。たとえば、FROM、LIST、BLANK、TITLE、PAGE、DATE、TIME、および HEADER オペランドを指定し、(また 10 バイトの文字フィールドと 7 バイトのゾーン 10 進数フィールドに対しては ON オペランド)、および値カウントを指定すれば、リスト・データ・セット内の出力を次のように表示できます。

title	- p -	mm/dd/yy	hh:mm:ss
header	header	header	
-----	-----	-----	
characters	sd		d
.	.		.
.	.		.
.	.		.

制御バイトは、各リスト・データ・セットの最初のバイトを占有します。タイトル行は、リスト・データ・セットの各ページ一番上に印刷されます。これには、ユーザーが指定した要素 (タイトル・ストリング、ページ番号、日付および時間) が指定した順序で含まれています。タイトル要素とタイトル要素の間には 8 個の空白があります。タイトル行の後には空白行が 1 行印刷されます。

ユーザーが指定したヘッダー (下線が引かれている) が、各ページのタイトル行の次に印刷され、各列の内容を示します。このヘッダーの次には 1 レコードごとに 1 行が続き、そのレコードのフィールド内に文字と数字が示されます。文字フィールドのヘッダーは左寄せされ、数値フィールドのヘッダーは右寄せされます。

フィールドは、OCCUR ステートメントで指定された順序と同じ順序で列に印刷されます。文字フィールドは左寄せされ、数値フィールドは右寄せされます。数値フィールドの場合、先行ゼロは抑制され、負符号には - が、正符号には空白が使用されます (正符号に + を使用する場合は、BLANK ではなく PLUS を指定できます)。値カウントでは、先行ゼロは抑制されます。

形式設定項目を使用して、報告書の各数値フィールドの表示を、区切り文字、桁数、小数点、小数部、符号、先行ゼロ、先行ストリング、浮動小数点ストリング、および後書きストリングについて変更できます。形式設定項目はまた、先行または後書きストリングを文字フィールドに挿入するのに使用できます。

列の幅は、ヘッダーの長さ、および文字データまたは数値データに必要な最大バイト数に応じて、動的に調整されます。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### ON(p,m,f)

この操作で使用される数値フィールドの位置、長さ、および形式を指定します。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。p は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です (RRRR は、4 バイト・レコード記述子ワードを表します)。

	Fixed-length record					Variable-length record									
	D	A	T	A	...		R	R	R	R	D	A	T	A	...
p=	1	2	3	4		p=	1	2	3	4	5	6	7	8	

## OCCUR 演算子

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを越えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
BI	1 ~ 4 バイト	2 進数、符号なし
FI	1 ~ 4 バイト	固定小数点、符号あり
PD	1 ~ 8 バイト	パック 10 進数、符号あり
ZD	1 ~ 15 バイト	ゾーン 10 進数、符号あり
CH	1 ~ 1500 バイト	文字
CSF または FS	1 ~ 16 バイト (15 桁まで)	任意指定の浮動符号が左端にある符号付き数字
DT1	4 バイト	Z'yyyymmdd' と解釈された SMF 日付
DT2	4 バイト	Z'yyyymm' と解釈された SMF 日付
DT3	4 バイト	Z'yyyddd' と解釈された SMF 日付
TM1	4 バイト	Z'hmmss' と解釈された SMF 時間
TM2	4 バイト	Z'hhmm' と解釈された SMF 時間
TM3	4 バイト	Z'hh' と解釈された SMF 時間
TM4	4 バイト	Z'hmmssxx' と解釈された SMF 時間

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

CSF 形式または FS 形式フィールドの場合、

- 最大 15 桁が使用できます。16 桁の CSF/FS 値が見つかったら、ICETOOL はエラー・メッセージを出して操作を終了します。

ZD 形式または PD 形式フィールドの場合、

- 10 進数値に無効な数字 (A~F) が含まれている場合、ICETOOL はメッセージにその誤りのある値を示し、操作を終了します。
- F、E、C、A、8、6、4、2、および 0 は、等価の正符号として扱われます。したがって、ゾーン 10 進数値 F2F3C1、F2F3F1、および 020301 は 1 つの固有な値としてだけカウントされます。
- D、B、9、7、5、3、および 1 は、等価の負符号として扱われます。したがって、ゾーン 10 進数値 F2F3B0、F2F3D0、および 020310 は 1 つの固有な値としてだけカウントされます。

指定された基準を満たさないレコードのフィールドは、無効な数字 (PD と ZD) または超過数字 (CSF と FS) についての検査は行われません。



DT1、DT2 または DT3 形式フィールドの場合、

- 無効な SMF 日付があると、データ例外 (0C7 ABEND)、または誤った ZD 日付が発生します。
- SMF 日付値は常に正の数値として扱われます。

TM1、TM2、TM3 または TM4 形式フィールドの場合、

- 無効な SMF 時間があると、誤った ZD 時間が発生します。
- SMF 時間値は常に正の数値として扱われます。

### ON(p,m,f,formatting)

この操作で使用される数値フィールドまたは文字フィールドの位置、長さ、および形式を指定し、また、このフィールドのデータの印刷に関してどのように形式設定するかを指定します。BLANK オペランドは、自動的に有効になります。

詳細については、ON(p,m,f) を参照してください。

### formatting



このフィールドのデータが印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。数値フィールドにはどんな形式設定項目も指定できますが、L'string' および T'string' に限り、文字フィールドに指定します。

列の幅は、すべての形式設定項目が指定された結果挿入されるバイトの最大数が収容できるように動的に調整されます。

**mask** 467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**mask** に関する説明を参照してください。

#### E'pattern'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**E'pattern'** に関する説明を参照してください。

#### L'string'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**L'string'** に関する説明を参照してください。

#### F'string'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**F'string'** に関する説明を参照してください。

## OCCUR 演算子

### T'string'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、T'string' に関する説明を参照してください。

### LZ

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、LZ に関する説明を参照してください。

### Ndd

列幅を決定するときに、数値フィールドに使用する桁数を指定します。dd に 2 桁の数字 01 から 15 で桁数を指定します。

dd がデフォルトの桁数以上の場合は、dd が使用されます。dd がデフォルトの桁数より少ない場合、デフォルトの値が使用されます。

桁数はパターンにより決定されるため、E'pattern' を指定した場合 Ndd は無視されます。

### ON(p,m,HEX)

文字フィールドの位置と長さが、この操作で使用され、16 進形式 (各バイトごとに 00 ~ FF) で印刷されるように指定します。標準列見出しには '(p,m,HEX)' が使用されます (代替のヘッダー・オプションについては、HEADER('string')、HEADER(NONE)、および NOHEADER を参照してください)。

p については、ON(p,m,f) を参照してください。

m は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを超えるように拡張できません。フィールドは 1 ~ 1000 バイトにできます。

### ON(VLEN)

ON(1,2,BI) の指定と同じです。すなわち、1 桁目から始まる 2 バイト 2 進数フィールドです。可変長レコードの場合、ON(VLEN) は各レコードのレコード長を表します。RECORD LENGTH' は標準列見出しに使用されます。代替のヘッダー・オプションについては、467 ページの『DISPLAY 演算子』の HEADER('string')、HEADER(NONE)、および NOHEADER を参照してください。

### ON(VLEN,formatting)

ON(1,2,BI,formatting) n の指定と同じです。すなわち、1 桁目から始まる 2 バイト 2 進数フィールドです。また、このフィールドのデータの印刷に関してどのように形式設定するかを指定します。BLANK オペランドは、自動的に有効になります。

詳細については、ON(VLEN) を参照してください。

### formatting



このフィールドのデータが印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。

列の幅は、すべての形式設定項目が指定された結果挿入されるバイトの最大数が収容できるように動的に調整されます。

**mask** 467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**mask** に関する説明を参照してください。

#### **E'pattern'**

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**E'pattern'** に関する説明を参照してください。

#### **L'string'**

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**L'string'** に関する説明を参照してください。

#### **F'string'**

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**F'string'** に関する説明を参照してください。

#### **T'string'**

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**T'string'** に関する説明を参照してください。

#### **LZ**

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**LZ** に関する説明を参照してください。

#### **Ndd**

列幅を決定するときに、数値フィールドに使用する桁数を指定します。dd に **2 桁** の数字 01 から 15 で桁数を指定します。

dd がデフォルトの桁数以上の場合は、dd が使用されます。dd がデフォルトの桁数より少ない場合、デフォルトの値が使用されます。

桁数はパターンにより決定されるため、E'pattern' を指定した場合 Ndd は無視されます。

### **ON(VALCNT)**

固有な値のそれぞれについて、その発生回数を印刷することを指定します。

'VALUE COUNT' は標準列見出しに使用されます (代替のヘッダー・オプションについては、HEADER('string')、HEADER(NONE) および NOHEADER を参照してください)。

## OCCUR 演算子

### ON(VALCNT,formatting)

それぞれの固有値について、その発生回数を印刷することを指定します。また、値の数が印刷のためにどのように形式設定されるかを指定します。BLANK オペランドは、自動的に有効になります。

詳細については、ON(VALCNT) を参照してください。

### formatting



値の数が印刷のためにどのように形式設定されるかを示す形式設定項目を指定します。形式設定項目は、任意の順序で指定できますが、各項目はそれぞれ 1 回ずつしか指定できません。

列の幅は、すべての形式設定項目が指定された結果挿入されるバイトの最大数が収容できるように動的に調整されます。

**mask** 467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**mask** に関する説明を参照してください。

#### E'pattern'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**E'pattern'** に関する説明を参照してください。

#### L'string'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**L'string'** に関する説明を参照してください。

#### F'string'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**F'string'** に関する説明を参照してください。

#### T'string'

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**T'string'** に関する説明を参照してください。

#### LZ

467 ページの『DISPLAY 演算子』の ON(p,m,f,formatting) で、**LZ** に関する説明を参照してください。

#### Ndd

列幅を決定するときに、値の数に使用する桁数を指定します。dd に 2 桁の数字 01 から 15 で桁数を指定します。

値の数のデフォルトの桁数は、15 です。値の数に 15 桁未満の値を指定する必要がある場合、Ndd を使用してそれを指示できます。これにより、桁数で決定されている列幅を縮小できます。たとえば、ON(VALCNT,N06) の指定で、15 (値の数のデフォルトの桁数) の代わりに 6 桁 (N06 の指定による) が使用されます。

Ndd および桁数が使用する値の数のオーバーフローを使用する場合、  
ICETOOL は処理を終了します。

桁数はパターンにより決定されるため、E'pattern' を指定した場合 Ndd  
は無視されます。

**LIST(listdd)**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**TITLE('string')**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**PAGE**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**DATE**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**DATE(abcd)**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**DATENS(abc)**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**TIME**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**TIME(abc)**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**TIMENS(ab)**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**BLANK**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**PLUS**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

ON(VALCNT) の場合、PLUS は BLANK として扱われます。

**HEADER('string')**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

## OCCUR 演算子

### HEADER(NONE)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### NOHEADER

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### LINES(n)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### INDENT(n)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### BETWEEN(n)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### ALLDUPS

印刷される ON 値を、複数回発生する値 (つまり、重複するフィールド値をもつもの) に限定します。値カウント > 1 の場合に ON 値が印刷されます。

ALLDUPS は HIGHER(1) と同じです。

### NODUPS

印刷される ON 値を、1 回だけ発生する値 (つまり、重複するフィールド値をもたないもの) に限定します。値カウント = 1 の場合に ON 値が印刷されます。

NODUPS は EQUAL(1) または LOWER(2) と同じです。

### HIGHER(x)

印刷される ON 値を、x 回発生する値に限定します。値カウント > x の場合に ON 値が印刷されます。

x は n または +n として指定します (ここで、n は 1 ~ 15 桁の 10 進数)。

### LOWER(y)

印刷される ON 値を、y 回より少なく発生する値に限定します。値カウント < y の場合に ON 値が印刷されます。

y は n または +n として指定します (ここで、n は 1 ~ 15 桁の 10 進数)。

### EQUAL(v)

印刷される ON 値を、v 回発生する値に限定します。値カウント = v の場合に ON 値が印刷されます。

v は n または +n として指定します (ここで、n は 1 ~ 15 桁の 10 進数)。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### WIDTH(n)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**UZERO**

-0 および +0 が符号なしゼロ値として扱われる、つまり、同じ値として処理されることを指定します。UZERO では、-0 および +0 は、ON 処理について正として扱われます。

UZERO は、-0 および +0 を符号付きのゼロ値として扱う、つまり、異なる値として処理するというデフォルトを指定変更します。UZERO を指定しない場合、ON 処理について、-0 は負として、+0 は正として扱われます。

**OCCUR の例**

下記の例に示されている OCCUR 演算子は、すべて単一の ICETOOL ジョブ・ステップに組み込むことができますが、わかりやすくするために、別々に示して説明しています。報告書形式を調整するための追加の例に関しては 467 ページの『DISPLAY 演算子』を参照してください。

**例 1**

```
OCCUR FROM(SOURCE) LIST(VOLSERS) ON(40,6,CH) ON(VALCNT)
```

VOLSERS データ・セットに次のものを印刷します。

- 標準見出しを含むヘッダ行
- 次のものを含む標準形式の固有な ON(40,6,CH) 値のそれぞれについて 1 つのデータ行
  - 固有な値の SOURCE データ・セットの 40~45 桁目の文字
  - 固有な値の SOURCE データ・セット内の発生回数

VOLSERS 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードが、図示のための値で示されています)。

(40,6,CH)	VALUE COUNT
ABC001	000000000000025
ABC002	000000000000011
.	.
.	.
.	.

ヘッダ行は各ページの一番上にあります。

**例 2**

```
OCCUR FROM(IN) LIST(LIST1) -
  TITLE(' 3090 Distribution ') -
  PAGE -
  HEADER('Data Centers') ON(VALCNT) -
  HEADER('State') ON(1,16,CH) -
  HEADER('3090s') ON(25,3,PD) -
  BLANK
```

LIST1 データ・セットに次のものを印刷します。

- 指定したタイトルとページ番号が入っているタイトル行

## OCCUR 演算子

- 指定した下線付きヘッダーが入っているヘッダー行
- 固有な ON(1,16,CH) および ON(25,3,PD) 値のそれぞれについて、次のものを含む BLANK 形式のデータ行
  - 固有な値の IN データ・セット内での発生回数
  - 固有な値の IN データ・セットの 1 ~ 16 桁目の文字
  - 固有な値の IN データ・セットの 25 ~ 27 桁目のパック 10 進数値

LIST1 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードは、図示のための値で示されています)。

3090 Distribution - 1 -		
Data Centers	State	3090s
12	Alabama	1
6	Alabama	2
.	.	.
.	.	.
.	.	.

合計行および下線付きヘッダー行は各ページの一番上にあります。

### 例 3

```
OCCURS FROM(FAILURES) LIST(CHECKIT) -  
  DATE TITLE('Possible System Intruders') PAGE -  
  HEADER(' Userid ') HEADER(' Logon Failures ') -  
  ON(23,8,CH)      ON(VALCNT) -  
  HIGHER(4) -  
  BLANK
```

CHECKIT データ・セットに次のものを印刷します。

- 日付、指定したタイトル、およびページ番号が入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 5 回以上発生する固有な ON(23,8,CH) 値のそれぞれについて、次のものを含む BLANK 形式のデータ行
  - FAILURES データ・セットの 23 ~ 30 桁目の文字
  - FAILURES データ・セットの 23 ~ 30 桁目の文字の発生回数

CHECKIT 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードは、図示のための値で示されています)。

10/21/92 Possible System Intruders - 1 -		
Userid	Logon Failures	
B7234510	5	
D9853267	11	
.	.	
.	.	
.	.	

合計行および下線付きヘッダー行は各ページの一番上にあります。



## 例 4

```
OCCUR FROM(VARIN) LIST(ONCE) -
      TITLE('Record lengths that occur only once') -
      TIME(12:) DATE(DMY.) -
      ON(VLEN) NODUPS BLANK
```

ONCE データ・セットに次のものが印刷されます。

- 指定したタイトル、時間、および日付が入っているタイトル行
- 標準ヘッダーが入っているヘッダー行
- 1 回だけ発生する各レコード長ごとに、そのレコード長を含む BLANK 形式のデータ行

ONCE 出力は新しいページから始まり、次のようになります (最初の 2 つのレコードは、図示のための値で示されています)。

```
Record lengths that occur only once      09:52:17 am      21.10.92
RECORD LENGTH
      57
      61
      .
      .
      .
```

タイトル行とヘッダー行は各ページの一番上にあります。

## 例 5

```
OCCUR FROM(BRANCH) LIST(CALLRPT)-
      DATENS(4MD)-
      TITLE('Yearly Branch Phone Call Counts')-
      HEADER('Phone Number') ON(7,10,ZD,E'(999)-999-9999')-
      HEADER('Calls') ON(VALCNT,A1,N05)-
      INDENT(5) BETWEEN(10)
```

CALLRPT データ・セットに次のものを印刷します。

- 日付 (区切り文字は含まない) および指定したタイトルが入っているタイトル行
- 指定した下線付きヘッダーが入っているヘッダー行
- 次のものを含む固有な ON(7,10,ZD) 値のそれぞれについて 1 つのデータ行
  - E'pattern' 形式設定項目に従って (ddd)-ddd-dddd と印刷される、BRANCH データ・セットの 7 ~ 16 桁目からの、ゾーン 10 進数。
  - A1 および N05 形式設定項目に従って dd,ddd と印刷されるこの値の発生回数

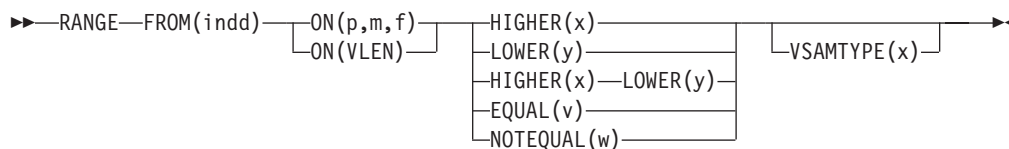
INDENT(5) オペランドの指定により、報告書は 5 個のブランクで字下げされます。また、BETWEEN(10) オペランドの指定により、列と列の間に 10 個のブランクが挿入されます。

CALLRPT 出力は、新しいページで始まり次のように表示されます。

## OCCUR 演算子

Phone Number	Calls
(037)-325-1807	3,125
(216)-721-5530	2,087
(218)-062-7214	872

## RANGE 演算子



特定の数値フィールドについて、指定した範囲内の値カウントを含むメッセージを印刷します。

DFSORT を呼び出して、`indd` データ・セットを ICETOOL の E35 ユーザー出口にコピーします。ICETOOL は、その E35 ユーザー出口が判別した範囲カウントを含むメッセージを印刷します。

この範囲は、 $x$  より高く、あるいは  $y$  より低く、 $x$  より高く  $y$  より低く、 $v$  と等しく、 $w$  と等しくなく（ここで、 $x$ 、 $y$ 、 $v$ 、および  $w$  は符号ありまたは符号なしの 10 進数値）指定できます。範囲を  $x$  より高く  $y$  より低く指定した場合、この範囲は有効である必要があります（たとえば、5 より高く 6 より低くと指定した場合、そのような基準を満たす整数値がないため有効ではありません）。

ユーザー独自の DFSORT MODS、INREC、または OUTREC ステートメントを指定できません。これらのステートメントが、この演算子で ICETOOL により渡される DFSORT ステートメントを指定変更するためです。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### ON(p,m,f)

この操作で使用される数値フィールドの位置、長さ、および形式を指定します。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。**p** は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です（RRRR は、4 バイト・レコード記述子ワードを表します）。

Fixed-length record				Variable-length record																									
	D		A		T		A		...		R		R		R		R		D		A		T		A		...		
p=	1	2	3	4						p=	1	2	3	4	5	6	7	8											

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを超えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
BI	1 ~ 4 バイト	2 進数、符号なし
FI	1 ~ 4 バイト	固定小数点、符号あり
PD	1 ~ 8 バイト	パック 10 進数、符号あり
ZD	1 ~ 15 バイト	ゾーン 10 進数、符号あり
CSF または FS	1 ~ 16 バイト (15 桁まで)	任意指定の浮動符号が左端にある符号付き数字

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

CSF 形式または FS 形式フィールドの場合、

- 最大 15 桁が使用できます。16 桁の CSF/FS 値が見つかったら、ICETOOL はエラー・メッセージを出して操作を終了します。

ZD 形式または PD 形式フィールドの場合、

- 10 進数値に無効な数字 (A~F) が含まれている場合、ICETOOL はメッセージにその誤りのある値を示し、操作を終了します。
- 符号が F、E、C、A、8、6、4、2、または 0 の場合、値は正の値として扱われます。
- 符号が D、B、9、7、5、3、または 1 の場合、値は負の値として扱われます。

ZD、PD、または CSF/FS 形式フィールドの場合、負のゼロ値は正のゼロ値として扱われます。

### ON(VLEN)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### HIGHER(x)

x より高い値が、範囲に含まれているものとしてカウントされます。

HIGHER(x) だけが指定されている場合は、x < 値のときに、範囲カウントが増分されます。LOWER(y) も指定されている場合は、x < 値 < y のときに、範囲カウントが増分されます。

x は、n、+n、または -n として指定します。ここで、n は 1 ~ 15 桁の数字です。

### LOWER(y)

y より低い値が、範囲に含まれるものとしてカウントされます。LOWER(y) だ

## RANGE 演算子

けが指定されている場合は、値  $< y$  ときに、範囲カウントが増分されます。  
HIGHER(x) も指定されている場合は、 $x < \text{値} < y$  のときに、範囲カウントが増分されます。

y は、n、+n、または -n として指定します。ここで、n は 1 ~ 15 桁の数字です。

### EQUAL(v)

v に等しい値が、範囲に含まれるものとしてカウントされます。v = 値のときに、範囲カウントが増分されます。

v は、n、+n、または -n として指定します。ここで、n は 1 ~ 15 桁の 10 進数です。

### NOTEQUAL(w)

w に等しくない値が、範囲に含まれるものとしてカウントされます。w  $\neq$  値のときに、範囲カウントが増分されます。

w は、n、+n、または -n として指定します。ここで、n は 1 ~ 15 桁の 10 進数です。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

## RANGE の例

```
RANGE FROM(DATA1) ON(VLEN) HIGHER(10)
RANGE FROM(DATA2) ON(11,6,ZD) LOWER(+3000)
RANGE FROM(DATA3) ON(29001,4,FI) -
    HIGHER(-10000) LOWER(27)
RANGE FROM(DATA2) ON(25,3,PD) EQUAL(-999)
RANGE FROM(DATA3) ON(40,1,BI) NOTEQUAL(199)
```

最初の RANGE 演算子は、DATA1 データ・セットの 1 ~ 2 桁目の 2 進数値で、10 より高い値のカウントが含まれているメッセージを印刷します。

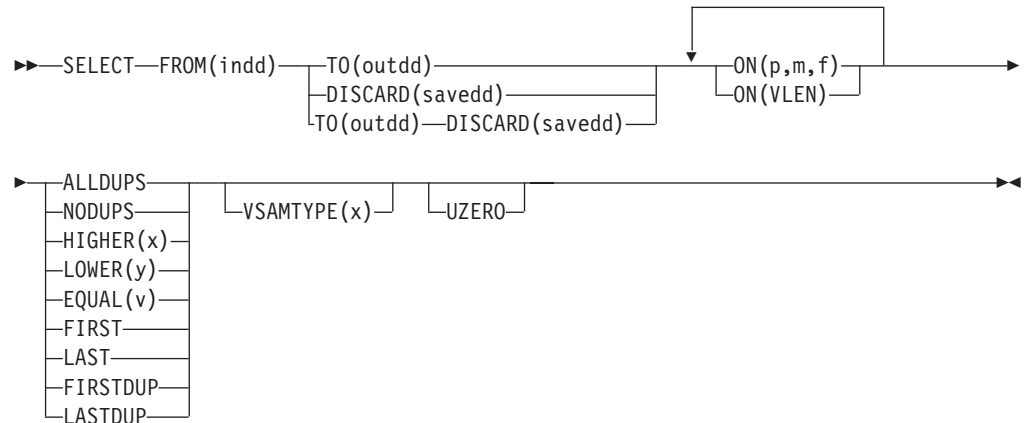
2 番目の RANGE 演算子は、DATA2 データ・セットの 11 ~ 16 桁目のゾーン 10 進数値で、3000 より低い値のカウントが含まれているメッセージを印刷します。

3 番目の RANGE 演算子は、DATA3 データ・セットの 29 桁目 001 ~ 29 桁目 004 の固定小数点値で、-10 000 より高く 27 より低い値のカウントが含まれているメッセージを印刷します。

4 番目の RANGE 演算子は、DATA2 データ・セットの 25 ~ 27 桁目のパック 10 進数値で、-999 に等しい値のカウントが含まれているメッセージを印刷します。

5 番目の RANGE 演算子は、DATA3 データ・セットの 40 桁目の 2 進数値で、199 と等しくない値のカウントが含まれているメッセージを印刷します。この RANGE 演算子を使用して、40 桁目が 'G' ではないレコードの数をカウントできます。199 (X'C7') が 'G' の EBCDIC コードであるためです。あるいは、COUNT 演算子を OMIT COND=(40,1,CH,EQ,C'G') とともに使用することもできます。

## SELECT 演算子



指定した数値フィールド値または文字フィールド値が出現した回数を満たす基準に基づいて、入力データ・セットからレコードを選択します。これにより、重複フィールド値をもつレコードのみの保存、重複しないフィールド値をもつレコードのみの保存、または  $n$  回より多く、または少なく、もしくは正確に  $n$  回だけ出現するフィールド値をもつレコードのみの保存、あるいはそれぞれの固有または重複フィールド値をもつ最初または最後のレコードのみの保存を行うことができます。1 ~ 10 個のフィールドを指定できます。少なくとも 1 つの ON(VLEN) または ON(p,m,f) フィールドを指定します。指定された ON フィールドは、すべて "値カウンタ" (つまり、ON 値が出現した回数) が基準に適合しているかどうかを判別するために使用されます。

DISCARD(savedd) を使用して、基準を満たさないレコード (すなわち、廃棄されるレコード) を、savedd データ・セットに保管できます。DISCARD(savedd) は、TO(outdd) を指定してもしなくても使用できます。

DFSORT が呼び出され、indd データ・セットを分類します。ICETOOL はその E35 出口を使用して、どのレコードを outdd データ・セットまたは savedd データ・セットに組み込むかを決定します。

ユーザーのレコードがユーザー指定の基準を満たしているかどうかを判別できるようになるまでそのレコードを保管しておくために、ICETOOL は、ICETOOL および DFSORT が通常必要とする記憶域より上に、SELECT 処理のための余分な記憶域を必要とします。多くの場合、少しの記憶域しか必要でないので、入手も可能です (16 メガバイト境界より上)。ただし、レコード長が大きい FROM データ・セットで、しかも多くの保管レコードを必要とする基準の場合は、多くの記憶域が必要となります。たとえば、32 756 のレコード長で HIGHER(99) の場合、3 メガバイトを超える記憶域が必要です。ICETOOL が必要な記憶域を入手することができない場合は、メッセージを出して SELECT 操作を終了します。メッセージに示された大きさを使用して REGION を増加すると、ICETOOL は正常に実行できるようになります。

## SELECT 演算子

ワークスペースを分類処理で必ず使用できるようにするために、DYNALLOC オプションが DFSORT に渡されます。動的割り振りのためのインストール・デフォルトが SELECT 演算子について適切でない場合、次のいずれかの処置をとることができます。

1. 次のような OPTION 制御ステートメントを使用して、DYNALLOC オプションを指定変更する。

```
OPTION DYNALLOC=(3390,5)
```

これは、DFSPARM データ・セット内にあります (すべての OCCUR、SELECT、SORT、および UNIQUE 演算子に適用されます)。

2. SORTWKdd DD ステートメントを使用して、動的割り振りを指定変更する (すべての OCCUR、SELECT、および UNIQUE 演算子に適用されます)。詳細については、79 ページの『SORTWKdd DD ステートメント』を参照してください。

ICETOOL でテープ作業データ・セットを使用することはできません。

ユーザー独自の DFSORT MODS、INREC、OUTREC、または OUTFIL ステートメントを指定できません。これらのステートメントが、この演算子 (SELECT) のために ICETOOL により渡される、DFSORT ステートメントを指定変更してしまうからです。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### TO(outdd)

DFSORT がその操作で選択するレコード (すなわち、指定された基準を満たすレコード) を書き込む、出力データ・セットの DD 名を指定します。したがって、outdd データ・セットは ALLDUPS、NODUPS、HIGHER(x)、LOWER(y)、EQUAL(v)、FIRST、LAST、FIRSTDUP または LASTDUP が選択したレコードを含みます。

outdd DD ステートメントは必須であり、DFSORT の SORTOUT データ・セット (DISCARD オペランドが指定されていない場合)、または OUTFIL データ・セット (DISCARD オペランドが指定されている場合) の規則に従う、出力データ・セットを定義します。

TO および DISCARD の両方を指定できます。DISCARD が指定されない場合は、TO を指定する必要があります。TO が指定されない場合は、DISCARD を指定する必要があります。

TO オペランドに指定する DD 名に、FROM または DISCARD オペランドに指定した DD 名と同じものは指定できません。

詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### DISCARD(savedd)

DFSORT がこの操作で選択しないレコード (すなわち、指定された基準を満た

さないレコード) を書き込む、出力データ・セットの DD 名を指定します。したがって、savedd データ・セットは ALLDUPS、NODUPS、HIGHER(x)、LOWER(y)、EQUAL(v)、FIRST、LAST、FIRSTDUP または LASTDUP が廃棄したレコードを含みます。

savedd DD ステートメントは必須であり、DFSORT の OUTFIL データ・セットの規則に従う、出力データ・セットを定義します。

TO および DISCARD の両方を指定できます。DISCARD が指定されない場合は、TO を指定する必要があります。TO が指定されない場合は、DISCARD を指定する必要があります。

DISCARD オペランドに指定する DD 名に、FROM または TO オペランドに指定した DD 名と同じものは指定できません。

詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### ON(p,m,f)

この操作で使用される数値フィールドまたは文字フィールドの位置、長さ、および形式を指定します。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。**p** は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です (RRRR は、4 バイト・レコード記述子ワードを表します)。

Fixed-length record				Variable-length record																							
	D		A		T		A		...		R		R		R		R		D		A		T		A		...
p=	1		2		3		4			p=	1		2		3		4		5		6		7		8		

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを超えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
BI	1 ~ 1500 バイト	2 進数、符号なし
FI	1 ~ 256 バイト	固定小数点、符号あり
PD	1 ~ 8 バイト	パック 10 進数、符号あり
ZD	1 ~ 15 バイト	ゾーン 10 進数、符号あり
CH	1 ~ 1500 バイト	文字
CSF または FS	1 ~ 16 バイト	任意指定の浮動符号が左端にある符号付き数字

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

ZD 形式または PD 形式フィールドの場合、

- F、E、C、A、8、6、4、2、および 0 は、等価の正符号として扱われます。したがって、ゾーン 10 進数値 F2F3C1、F2F3F1、および 020301 は 1 つの固有な値としてだけカウントされます。

## SELECT 演算子

- D、B、9、7、5、3、および 1 は、等価の負符号として扱われます。したがって、ゾーン 10 進数値 F2F3B0、F2F3D0、および 020310 は 1 つの固有な値としてだけカウントされます。
- 数字は妥当性について検査されません。

### ON(VLEN)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### ALLDUPS

選択するレコードを、2 回以上発生する ON 値をもつレコード (値カウント > 1) に限定します。このオペランドを使用して、重複フィールド値をもつレコードだけを保存できます。

ALLDUPS は HIGHER(1) と同じです。

### NODUPS

選択するレコードを、1 回だけ発生する ON 値をもつレコード (値カウント = 1) に限定します。このオペランドを使用して、重複フィールド値をもたないレコードだけを保存できます。

NODUPS は EQUAL(1) または LOWER(2) と同じです。

### HIGHER(x)

選択するレコードを、x 回以上発生する ON 値をもつレコード (値カウント > x) に限定します。このオペランドを使用して、x 回より多く発生するフィールド値をもつレコードだけを保存できます。

x は n または +n として指定します (ここで、n は 0 ~ 99)。

### LOWER(y)

選択するレコードを、y 回より少なく発生する ON 値をもつレコード (値カウント < y) に限定します。このオペランドを使用して、y 回より少なく発生するフィールド値をもつレコードだけを保存できます。

y は n または +n として指定します (ここで、n は 0 ~ 99)。

### EQUAL(v)

選択するレコードを、v 回以上発生する ON 値をもつレコード (値カウント = v) に限定します。このオペランドを使用して、v 回発生するフィールド値をもつレコードだけを保存できます。

v は n または +n として指定します (ここで、n は 0 ~ 99)。

### FIRST

選択するレコードを、1 回だけ発生する ON 値をもつレコード (値カウント = 1) および 1 回以上発生する ON 値をもつレコードの最初のレコードに限定します。このオペランドを使用して、固有な各フィールド値ごとに、最初のレコードだけを保存できます。

### LAST

選択するレコードを、1 回だけ発生する ON 値をもつレコード (値カウント = 1) および 2 回以上発生する ON 値をもつレコード (値カウント = 1) の最後のレコードに限定します。このオペランドを使用して、固有な各フィールド値ごとに、最後のレコードだけを保存できます。



**FIRSTDUP**

選択するレコードを、1 回以上発生する ON 値をもつレコード (値カウント > 1) の最初のレコードに限定します。このオペランドを使用して、重複フィールド値をもつレコードの最初のレコードだけを保存できます。

**LASTDUP**

選択するレコードを、1 回以上発生する ON 値をもつレコード (値カウント > 1) の最後のレコードに限定します。このオペランドを使用して、重複フィールド値をもつレコードの最後のレコードだけを保存できます。

**VSAMTYPE(x)**

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

**UZERO**

510 ページの『OCCUR 演算子』で説明されている OCCUR ステートメントのこのオペランドの項を参照してください。

**SELECT の例**

下記の例の SELECT 演算子は、すべて単一の ICETOOL ジョブ・ステップに含めることができますが、わかりやすくするために、別々に示して説明しています。

**例 1**

```
SELECT FROM(INPUT) TO(DUPS) ON(11,8,CH) ON(30,44,CH) ALLDUPS
```

INPUT データ・セットを DUPS データ・セットに分類します。11 ~ 18 桁目の文字および 30 ~ 73 桁目の文字が 2 回以上発生するレコードのみ (つまり、重複 ON フィールド値をもつレコードのみ) を INPUT から選択します。

DUPS データ・セットは次のようになります (いくつかのレコードが、図示のために記載されています)。

USR002	EISSLER	12	DOC.EXAMPLES
DFSRT2	EISSLER	5	DOC.EXAMPLES
DFSRT5	MADRID	20	MYDATA
DFSRT1	MADRID	20	MYDATA
SYS003	MADRID	20	MYDATA
DFSRT2	MADRID	20	SORTST1.TEST
USR003	MADRID	20	SORTST1.TEST
.	.	.	.
.	.	.	.
.	.	.	.

**例 2**

```
SELECT FROM(INPUT) TO(ONLYONE) ON(23,3,FS) NODUPS
```

INPUT データ・セットを ONLYONE データ・セットに分類します。23 ~ 25 桁目の浮動符号値が 1 回だけ発生するレコード (つまり、重複 ON フィールドをもたないレコード) のみを INPUT から選択します。

## SELECT 演算子

ONLYONE データ・セットは次のようになります (いくつかのレコードが、図示のために記載されています)。

DFSRT2	EISSLER	5	DOC.EXAMPLES
DFSRT1	PACKER	8	ICETOOL.SMF.RUNS
USR002	EISSLER	12	DOC.EXAMPLES
SYS003	YAEGER	32	ICETOOL.TEST.CASES
DFSRT2	MCNEILL	108	FS.TEST.CASES
.	.	.	.
.	.	.	.
.	.	.	.

### 例 3

```
SELECT FROM(FAILURES) TO(CHECKOUT) ON(28,8,CH) ON(1,5,CH) -  
HIGHER(3)
```

FAILURES データ・セットを CHECKOUT データ・セットに分類します。28 ~ 35 桁目の文字と 1 ~ 5 桁目の文字が 4 回以上発生するレコード (つまり、4 つ以上の ON フィールド値をもつレコード) のみを FAILURES から選択します。

CHECKOUT データ・セットは次のようになります (いくつかのレコードが、図示のために記載されています)。

03/12/91	08:36:59	A3275647
03/12/91	09:27:32	A3275647
03/12/91	09:03:18	A3275647
03/12/91	08:56:13	A3275647
03/06/91	15:12:01	C3275647
03/06/91	14:57:00	C3275647
03/06/91	15:43:19	C3275647
03/06/91	16:06:39	C3275647
03/06/91	15:22:08	C3275647
.	.	.
.	.	.
.	.	.

### 例 4

```
SELECT FROM(BOOKS) TO(PUBLISHR) ON(29,10,CH) FIRST
```

BOOKS データ・セットを PUBLISHR データ・セットに分類します。29 ~ 38 桁目の文字が 1 回だけ発生するレコード、および 29 ~ 38 桁目の文字が 2 回以上発生するレコードの最初のレコード (つまり、固有な各 ON フィールド値ごとに 1 レコードずつ) のみを BOOKS から選択します。

PUBLISHR データ・セットは次のようになります (いくつかのレコードが、図示のために記載されています)。

Banana Slugs I Have Known	Brent	Animals
Toads on Parade	Cooper	Animals
Pets Around the World	Davis	Animals
.	.	.
.	.	.
.	.	.

## 例 5

```
SELECT FROM(BOOKS) TO(PUBLISHR) ON(29,10,CH) FIRST -
DISCARD(SAVEREST)
```

この例では、例 4 と同じ PUBLISHR データ・セットを作成します。さらに、PUBLISHER データ・セットに書き込まれないすべてのレコードを含む、SAVEREST データ・セットを作成します。SAVEREST データ・セットは次のようになります (いくつかのレコードが、図示のために記載されています)。

```
How to Talk to Your Amoeba Brent Animals
What Buzzards Want Davis Animals
Birds of Costa Rica Davis Animals
.
.
.
```

## 例 6

```
SELECT FROM(MASTPULL) TO(MATCH) ON(5,8,CH) FIRSTDUP
```

この例では、アカウント番号のリストを "pull" データ・セットで使用し、"master" データ・セットからのアカウント番号をもつレコードのみを選択する方法を示します。MASTPULL DD は、"master" データ・セットおよび "pull" データ・セットを (この順序で) 連結させます。

SELECT 演算子は、連結データ・セットを分類し、5 ~ 12 桁目の文字が 1 回以上発生するレコード (つまり、重複する各 ON フィールド値ごとに 1 レコードずつ) のみを選択します。"master" データ・セットが連結の最初になるため、選択したレコードは "master" データ・セットからのものとなります。

"master" データ・セットが以下のようなものである場合、

```
A52 RB172832 2001/03/15
N92 MX328126 2001/01/27
B12 LB018725 2000/12/28
J73 AB007231 2001/02/13
Q28 SP973004 2000/11/19
```

また、"pull" データ・セットが以下のようなものである場合、

```
AB007231
RS859276
QN005001
MX328126
```

MATCH データ・セットは以下ようになります。

```
J73 AB007231 2001/02/13
N92 MX328126 2001/01/27
```

**注:** この例は、重複アカウント番号が "master" または "pull" のどちらにも存在していないことを前提としています。これに該当しない場合、SELECT を FIRST または LAST と一緒に適切なデータ・セットに使用して、この前提を満たすことができます。たとえば、"master" データ・セットに重複アカウント番号が存在



これは xxxxCNTL データ・セット内 (この演算子だけに適用) または DFSPARM データ・セット (すべての OCCUR、SELECT、SORT、および UNIQUE 演算子に適用) にあります。

2. xxxxWKdd DD ステートメント (xxxx は USING オペランドで指定される) を使用して、動的割り振りの使用を指定変更する (この命令コードだけに適用)。詳細については、79 ページの『SORTWKdd DD ステートメント』を参照してください。

ICETOOL でテープ作業データ・セットを使用することはできません。

## オペランドの説明

下記オペランドは、任意の順序で指定できます。

### FROM(indd)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### USING(yyyy)

この操作で DFSORT により使用される制御ステートメント・データ・セットの DD 名の最初の 4 文字を指定します。yyyy は、形式 xxxxCNTL を構成したときに有効な DD 名となるような、4 文字である必要があります。yyyy に SYSx は指定できません。

xxxxCNTL DD ステートメントは必須であり、その中の制御ステートメントは、DFSORT の SORTCNTL データ・セットの規則に従っている必要があります。

xxxxCNTL データ・セットには、SORT ステートメントが含まれます。TO が指定されない場合は、xxxxCNTL データ・セットにも、1 つ以上の OUTFIL ステートメントか、またはすべてのレコードの後処置を行う E35 ルーチンのための MODS ステートメントを含める必要があります。他のステートメントはオプションです。

この処理の作業データ・セットの、動的割り振りを指定変更する場合、xxxxWKdd DD ステートメントを使用できます。

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### TO(outdd,...)

この操作で DFSORT により書き出される出力データ・セットの DD 名を指定します。1 ~ 10 個の outdd 名を指定できます。指定する各 outdd 名ごとに 1 つの outdd DD ステートメントが必要です。1 つの outdd データ・セットが指定された場合、SORTOUT 処理を使用して、indd データ・セットを outdd データ・セットに分類するために、DFSORT が一度呼び出されます。outdd データ・セットは、DFSORT の SORTOUT データ・セットに関する規則に従ったものである必要があります。複数の outdd データ・セットが指定され、SERIAL が指定されない場合は、OUTFIL 処理を使用して、indd データ・セットを outdd データ・セットに分類するために、DFSORT が一度呼び出されます。outdd データ・セットは DFSORT の OUTFIL データ・セットに関する規則に従ったものである必要があります。

FROM オペランドで指定された DD 名を、TO オペランドで指定できません。

## SORT 演算子

DD 名の選択の詳細については、453 ページの『JCL に関する制約事項』を参照してください。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### LOCALE(name)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### LOCALE(CURRENT)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### LOCALE(NONE)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### SERIAL

複数の outdd データ・セットが指定された場合に、OUTFIL 処理を使用しないことを指定します。DFSORT は複数回呼び出され、SORTOUT 処理を使用します。outdd データ・セットは DFSORT の SORTOUT データ・セットに関する規則に従っている必要があります。SERIAL の使用はお勧めしません。OUTFIL 処理の代わりに逐次処理 (つまり、DFSORT の複数回呼び出し) を使用すると、パフォーマンスが低下し、下記のような制約を課す場合がありますためです。単一の outdd データ・セットが指定された場合、SERIAL は無視されます。

xxxxCNTL データ・セット内の DFSORT 制御ステートメントを使用して、indd データ・セットを最初の outdd データ・セットへ分類するために、DFSORT が呼び出されます。分類操作が成功すれば、最初の outdd データ・セットを 2 番目の outdd データ・セットと後続の outdd データ・セットにコピーするために、DFSORT が必要な回数だけ呼び出されます。したがって、最大の効率を得るために、DASD およびテープの両方が含まれる outdd データ・セットのリストでは DASD データ・セットを最初に使用してください。複数の outdd データ・セットが指定されている場合、DFSORT は、outdd データ・セットが書き込まれた後、最初の outdd データ・セットを読み取って、それを他の outdd データ・セットにコピーできる必要があります。次の理由から、SYSOUT または DUMMY データ・セットを outdd データ・セット・リストの最初のデータ・セットとして使用しないでください。

- 最初のデータ・セットが SYSOUT の場合、DFSORT は、その SYSOUT データ・セットを 2 番目の outdd データ・セットにコピーしようとするとき異常終了します。
- 最初のデータ・セットが DUMMY の場合、DFSORT は空の DUMMY データ・セットを他の outdd データ・セットにコピーします (つまり、その結果出力された outdd データ・セットはすべて空になります)。

## SORT の例

下記の例の SORT 演算子は、すべて単一の ICETOOL ジョブ・ステップに組み込むことができますが、わかりやすくするために、別々に示して説明しています。

## 例 1

```
* Method 1
SORT FROM(MASTER) TO(PRINT,TAPE,DASD) USING(ABCD)
```

```
* Method 2
SORT FROM(MASTER) TO(DASD,TAPE,PRINT) USING(ABCD) SERIAL
```

この例は、複数の分類済み出力データ・セットを作成する 2 つの異なる方式を示しています。ABCDCNTL データ・セットに次のものが含まれていると仮定します。

```
SORT FIELDS=(15,20,CH,A,1,5,PD,D)
```

方式 1 は、DFSORT の呼び出しを 1 回、入力データ・セットに対して 1 回のパスを必要とし、出力データ・セットは任意の順序で指定できます。SORT 演算子は、ABCDCNTL データ・セットの中の SORT ステートメントおよび OUTFIL 処理を使用して、MASTER データ・セットからのすべてのレコードを PRINT (SYSOUT)、TAPE、および DASD データ・セットに分類します。

方式 2 は、DFSORT の呼び出しを 3 回、入力データ・セットに対して 3 回のパスを必要とし、SYSOUT データ・セットは最初の TO データ・セットであってはならないという制限が課されます。SORT 演算子は、ABCDCNTL データ・セットの中の SORT ステートメントを使用して、MASTER データ・セットからのすべてのレコードを DASD データ・セットに分類します。次に、その結果の DASD データ・セットを TAPE および PRINT (SYSOUT) データ・セットにコピーします。最初の TO データ・セットは 3 回 (書き込み、読み取り、読み取り) 処理されるので、DASD データ・セットを最初に入れたほうが TAPE データ・セットを最初に入れるよりもっと効率的です。PRINT を、TO リストの中で最初にセットできません。SYSOUT データ・セットは読み取れないためです。

## 例 2

```
* Method 1
SORT FROM(IN) TO(DEPT1) USING(DPT1)
SORT FROM(IN) TO(DEPT2) USING(DPT2)
SORT FROM(IN) TO(DEPT3) USING(DPT3)
```

```
* Method 2
SORT FROM(IN) USING(ALL3)
```

この例は、入力データ・セットの分類済みサブセットを作成するための 2 つの異なる方式を示しています。次のように仮定します。

- DPT1CNTL データ・セットには、次のものが含まれています。

```
SORT FIELDS=(51,2,BI,A,18,5,CH,A,58,4,BI,A)
INCLUDE COND=(5,3,CH,EQ,C'D01')
```

- DPT2CNTL データ・セットには、次のものが含まれています。

```
SORT FIELDS=(51,2,BI,A,18,5,CH,A,58,4,BI,A)
INCLUDE COND=(5,3,CH,EQ,C'D02')
```

- DPT3CNTL データ・セットには、次のものが含まれています。

## SORT 演算子

```
SORT FIELDS=(51,2,BI,A,18,5,CH,A,58,4,BI,A)
INCLUDE COND=(5,3,CH,EQ,C'D03')
```

- ALL3CNTL データ・セットには、次のものが含まれています。

```
SORT FIELDS=(51,2,BI,A,18,5,CH,A,58,4,BI,A)
OUTFIL FNAMES=DEPT1,INCLUDE=(5,3,CH,EQ,C'D01')
OUTFIL FNAMES=DEPT2,INCLUDE=(5,3,CH,EQ,C'D02')
OUTFIL FNAMES=DEPT3,INCLUDE=(5,3,CH,EQ,C'D03')
```

方式 1 は、DFSORT の 3 回の呼び出しと入力データ・セットの 3 回のパスを必要とします。

- 最初の SORT 演算子は、5 ~ 7 桁目に D01 を含む IN データ・セットからのレコードを DEPT1 データ・セットに分類する
- 2 番目の COPY 演算子は、5 ~ 7 桁目に D02 を含む IN データ・セットからのレコードを DEPT2 データ・セットに分類する
- 3 番目の COPY 演算子は、5 ~ 7 桁目に D03 を含む IN データ・セットからのレコードを DEPT3 データ・セットに分類する

方式 2 は、方式 1 と同じ結果を作り出しますが、TO オペランドの代わりにOUTFIL ステートメントを使用するため、DFSORT については 1 回の呼び出し、入力データ・セットについては 1 回のパスで済みます。

### 例 3

```
SORT FROM(IN1) TO(FRANCE) USING(SRT1) LOCALE(FR_FR)
SORT FROM(IN1) TO(CANADA) USING(SRT1) LOCALE(FR_CA)
SORT FROM(IN1) TO(BELGIUM) USING(SRT1) LOCALE(FR_BE)
```

この例は、異なる 3 つの国のために分類されたデータがどのように作成されるかを示しています。SRT1CNTL データ・セットに次のものが含まれていると仮定します。

```
SORT FIELDS=(5,20,CH,A,31,15,CH,A,1,4,FI,D,63,10,CH,D)
```

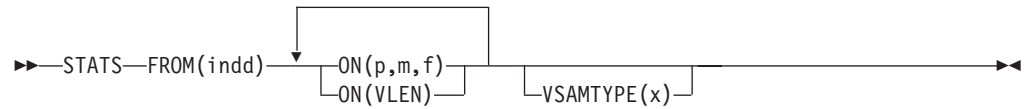
最初の SORT 演算子は、SRT1CNTL データ・セットの中の SORT ステートメントを使用して、IN1 データ・セットからのすべてのレコードを FRANCE データ・セットに分類します。文字 (CH) 制御フィールドは、ロケール FR\_FR (フランスでのフランス語) で定義されている照合規則に従って分類されます。

2 番目の SORT 演算子は、SRT1CNTL データ・セットの中の SORT ステートメントを使用して、IN1 データ・セットからのすべてのレコードを CANADA データ・セットに分類します。文字 (CH) 制御フィールドは、ロケール FR\_CA (カナダでのフランス語) で定義されている照合規則に従って分類されます。

3 番目の SORT 演算子は、SRT1CNTL データ・セットの中の SORT ステートメントを使用して、IN1 データ・セットからのすべてのレコードを BELGIUM データ・セットに分類します。文字 (CH) 制御フィールドは、ロケール FR\_BE (ベルギーでのフランス語) で定義されている照合規則に従って分類されます。



## STATS 演算子



指定した数値フィールドの最小、最大、平均、および合計の値を含むメッセージを印刷します。1 ~ 10 個のフィールドを指定できます。

DFSORT を呼び出して、indd データ・セットを ICETOOL の E35 ユーザー出口にコピーします。ICETOOL は、その E35 出口により決定された各フィールドの最小、最大、平均、および合計を含むメッセージを印刷します。

平均 (つまり、算術平均) は、合計をレコード数で割り、その結果が最も近い整数になるように端数を切り捨てて計算されます (たとえば、 $23 / 5 = 4$ 、 $-23 / 5 = -4$ )。

ユーザー独自の DFSORT MODS、INREC、または OUTREC ステートメントを指定できません。これらのステートメントが、この演算子で ICETOOL により渡される DFSORT ステートメントを指定変更するためです。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

**FROM(indd)**

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

**ON(p,m,f)**

この操作で使用される数値フィールドの位置、長さ、および形式を指定します。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。p は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です (RRRR は、4 バイト・レコード記述子ワードを表します)。

Fixed-length record	Variable-length record
D   A   T   A   ...	R   R   R   R   D   A   T   A   ...
p= 1 2 3 4	p= 1 2 3 4 5 6 7 8

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを超えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
BI	1 ~ 4 バイト	2 進数、符号なし
FI	1 ~ 4 バイト	固定小数点、符号あり
PD	1 ~ 8 バイト	パック 10 進数、符号あり

## STATS 演算子

形式コード	長さ	説明
ZD	1 ~ 15 バイト	ゾーン 10 進数、符号あり
CSF または FS	1 ~ 16 バイト	任意指定の浮動符号が左端にある符号付き数字

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

フィールドの合計がオーバーフローした場合、ICETOOL は処理を続行しますが、その FIELD の平均と合計にアスタリスクを印刷します。

CSF 形式または FS 形式フィールドの場合、

- 最大 15 桁が使用できます。16 桁の CSF/FS 値が見つかると、ICETOOL はエラー・メッセージを出して操作を終了します。

ZD 形式または PD 形式フィールドの場合、

- 10 進数値が無効な数字 (A~F) を含んでいると、ICETOOL はその間違っただけの値をメッセージに示し、そのフィールドの最小、最大、平均、および合計にアスタリスクを印刷します。
- 符号が F、E、C、A、8、6、4、2、または 0 の場合、値は正の値として扱われます。
- 符号が D、B、9、7、5、3、または 1 の場合、値は負の値として扱われます。

ZD、PD、または CSF/FS 形式フィールドの場合、負のゼロ値は正のゼロ値として扱われます。

### ON(VLEN)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

## STATS の例

```
STATS FROM(DATA1) ON(VLEN) ON(15,4,ZD)
```

DATA1 データ・セットの 1 ~ 2 桁目の 2 進数値の最小、最大、平均、および合計を含むメッセージを印刷します。可変長レコードの場合、これはレコードの長さについての統計になります。DATA1 データ・セットの 15 ~ 18 桁目のゾーン 10 進数値の最小、最大、平均、および合計を含むメッセージを印刷します。

## UNIQUE 演算子



指定の数値フィールドまたは文字フィールドについて、固有な値の数を含むメッセージを印刷します。

DFSORT が呼び出され、indd データ・セットを ICETOOL の E35 ユーザー出口に分類します。ICETOOL は、その E35 ユーザー出口により決定された固有なカウントを含むメッセージを印刷します。

ワークスペースを分類処理で必ず使用できるようにするために、DYNALLOC オプションが DFSORT に渡されます。動的割り振りのためのインストール・デフォルトが UNIQUE 演算子について適切でない場合、次のいずれかの処置をとることができます。

1. 次のような OPTION 制御ステートメントを使用して、DYNALLOC オプションを指定変更する。

```
OPTION DYNALLOC=(3390,5)
```

これは、DFSPARM データ・セット内にあります (すべての OCCUR、SELECT、SORT、および UNIQUE 演算子に適用されます)。

2. SORTWKdd DD ステートメントを使用して、動的割り振りを指定変更する (すべての OCCUR、SELECT、および UNIQUE 演算子に適用されます)。詳細については、79 ページの『SORTWKdd DD ステートメント』を参照してください。

ICETOOL でテープ作業データ・セットを使用することはできません。

ユーザー独自の DFSORT MODS、INREC、OUTREC、SUM、または RECORD ステートメントを指定できません。これらのステートメントが、この演算子で ICETOOL により渡される DFSORT ステートメントを指定変更するためです。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

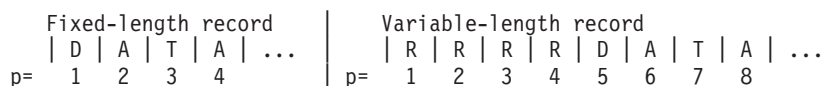
467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### ON(p,m,f)

この操作で使用される数値フィールドまたは文字フィールドの位置、長さ、および形式を指定します。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。**p** は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です (RRRR は、4 バイト・レコード記述子ワードを表します)。

## UNIQUE 演算子



**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを越えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
BI	1 ~ 256 バイト	2 進数、符号なし
FI	1 ~ 256 バイト	固定小数点、符号あり
PD	1 ~ 32 バイト	パック 10 進数、符号あり
ZD	1 ~ 32 バイト	ゾーン 10 進数、符号あり
CH	1 ~ 1500 バイト	文字
CSF または FS	1 ~ 16 バイト	任意指定の浮動符号が左端にある符号付き数字

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

ZD 形式または PD 形式フィールドの場合、

- F、E、C、A、8、6、4、2、および 0 は、等価の正符号として扱われます。したがって、ゾーン 10 進数値の F2F3C1、F2F3F1、および 020301 は 1 つの固有な値としてだけカウントされます。
- D、B、9、7、5、3、および 1 は、等価の負符号として扱われます。したがって、ゾーン 10 進数値 F2F3B0、F2F3D0、および 020310 は 1 つの固有な値としてだけカウントされます。
- 数字は妥当性について検査されません。

### ON(VLEN)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

### UZERO

510 ページの『OCCUR 演算子』で説明されている OCCUR ステートメントのこのオペランドの項を参照してください。

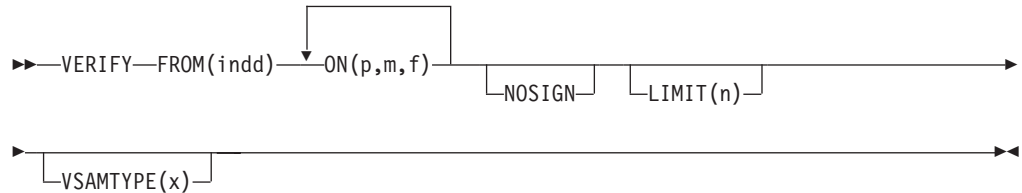
## UNIQUE の例

```
UNIQUE FROM(DATAIN) ON(20,40,CH)
UNIQUE FROM(DATAIN) ON(5,3,ZD)
```

最初の UNIQUE 演算子は、DATAIN データ・セットの 20 ~ 59 桁目にある固有な文字データのカウンタを含むメッセージを印刷します。

2 番目の UNIQUE 演算子は、DATAIN データ・セットの 5 ～ 7 桁目にある固有なゾーン 10 進数値のカウントを印刷します。

## VERIFY 演算子



データ・セット内の特定の 10 進数フィールドを検査し、各フィールドについて見つかったそれぞれの無効値を示すメッセージを印刷します。1 ～ 10 個のフィールドを指定できます。

DFSORT を呼び出して、indd データ・セットを ICETOOL の E35 ユーザー出口にコピーします。ICETOOL はその E35 ユーザー出口を使用して、それぞれの値の数字と符号の妥当性を検査し、見つかった無効値のそれぞれについて、レコード番号とフィールド値 (16 進数) を含むエラー・メッセージを印刷します。

ユーザー独自の DFSORT MODS、INREC、または OUTREC ステートメントを指定できません。これらのステートメントが、この演算子で ICETOOL により渡される DFSORT ステートメントを指定変更するためです。

### 注:

1. 無効な 10 進数値は、DISPLAY、OCCUR、RANGE、および STATS 演算子の場合にも識別されます。
2. 467 ページの『DISPLAY 演算子』の例 9 に示されているように、DISPLAY 演算子を使用して、各無効 (および有効) 10 進数値ごとに、相対レコード番号、16 進数値、および関連フィールドを識別した報告書を印刷できます。

## オペランドの説明

下記のオペランドは、任意の順序で指定できます。

### FROM(indd)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### ON(p,m,f)

この操作に使用される 10 進数フィールドの位置、長さ、および形式を指定します。

**p** は、入力レコードの先頭を基準にした、このフィールドの最初のバイトを指定します。**p** は、下表に示されているように、固定長レコードの最初のデータ・バイトの場合は 1、可変長レコードの最初のデータ・バイトの場合は 5 です (RRRR は、4 バイト・レコード記述子ワードを表します)。

## VERIFY 演算子

Fixed-length record				Variable-length record																							
	D		A		T		A		...		R		R		R		R		D		A		T		A		...
p=	1		2		3		4		...	p=	1		2		3		4		5		6		7		8		...

**m** は、フィールドの長さをバイト単位で指定します。フィールドの位置を 32752 またはレコードの終わりを越えるように拡張できません。フィールドの最大長はその形式により異なります。

**f** は次のようにフィールドの形式を指定します。

形式コード	長さ	説明
PD	1 ~ 16 バイト	バック 10 進数、符号あり
ZD	1 ~ 18 バイト	ゾーン 10 進数、符号あり

注: 形式の詳細については、733 ページの『付録 C. データ形式記述』を参照してください。

次のいずれかの状況下では、値は無効であると見なされます。

- 数字としての A ~ F を含む場合 (たとえば、00AF の PD フィールド、または F2FB の ZD フィールド)
- 符号としての 0 ~ 9 を含み、NOSIGN オペランドが指定されていない場合 (たとえば、3218 の PD フィールド、または F235 の ZD フィールド)

間違った値の数が無効な 10 進値の LIMIT に達すると、ICETOOL は操作を終了します。LIMIT オペランドが指定されていない場合は、無効な 10 進値の限界としてデフォルトの 200 が使用されます。

### NOSIGN

10 進数値の符号が妥当性検査されないことを指定します (0 ~ 9 についての無効符号検査のデフォルトを指定変更)。

### LIMIT(n)

467 ページの『DISPLAY 演算子』の DISPLAY ステートメントのこのオペランドの項を参照してください。

### VSAMTYPE(x)

455 ページの『COPY 演算子』で説明されている COPY ステートメントのこのオペランドの項を参照してください。

## VERIFY の例

```
VERIFY FROM(NEW) ON(22,16,PD) ON(7,9,ZD)
VERIFY FROM(NEW) ON(22,16,PD) ON(7,9,ZD) NOSIGN LIMIT(10)
```

最初の VERIFY 演算子は、無効数字 (A ~ F) および無効符号 (0 ~ 9) について、NEW データ・セットの 22 ~ 37 桁目のバック 10 進数値と 7 ~ 15 桁目のゾーン 10 進数値を検査します。無効な数字または符号を含む値 (もしあれば) を識別するメッセージが印刷されます。200 個の無効値が検出された場合は、操作は終了します。

2 番目の VERIFY 演算子は、無効な数字 (A ~ F) について、NEW データ・セットの 22 ~ 37 桁目のバック 10 進数値および 7 ~ 15 桁目のゾーン 10 進数値を

検査します。無効な数字を含む値 (もしあれば) を識別するメッセージが印刷されません。10 個の無効値が検出された場合は、操作は終了します。

**注:** 467 ページの『DISPLAY 演算子』の例 9 に示されているように、DISPLAY 演算子を使用して、各無効 (および有効) 10 進数値ごとに、相対レコード番号、16 進数値、および関連フィールドを識別した報告書を印刷できます。

## プログラムからの ICETOOL の呼び出し

LINK、ATTACH、または XCTL システム・マクロを使用して、ICETOOL をアセンブラー・プログラムから呼び出すことができます。標準関係規則を使用します。ICETOOL は、処理を終了すると、これまでに検出した最高位操作戻りコードをレジスター 15 にセットして、呼び出しプログラムに戻ります。ICETOOL 戻りコードの説明については、552 ページの『ICETOOL の戻りコード』を参照してください。

プログラムから ICETOOL を呼び出す場合、2 つの異なるインターフェース、TOOLIN インターフェースとパラメーター・リスト・インターフェースのいずれかを選択します。

## TOOLIN インターフェース

TOOLIN インターフェースでは、TOOLIN データ・セット内に ICETOOL ステートメントを指定します。ICETOOL は、TOOLMSG データ・セットにメッセージを印刷しますが、ユーザー・プログラムに情報を直接は戻しません。

TOOLIN インターフェースを使用するには、下記のように ICETOOL を呼び出す前に、0 の値をレジスター 1 にセットするか、あるいは X'80000000' を含む 4 バイト・フィールドのアドレスをレジスター 1 に入れます。

```

...
SLR   R1,R1           TOOLIN INTERFACE - METHOD 1
LINK  EP=ICETOOL     CALL ICETOOL
...
LA    R1,NOPLIST     TOOLIN INTERFACE - METHOD 2
LINK  EP=ICETOOL     CALL ICETOOL
...
NOPLIST DC X'80',AL3(0)  TOOLIN INTERFACE INDICATOR
...

```

## パラメーター・リスト・インターフェース

パラメーター・リスト・インターフェースを使用すれば、ICETOOL が引き出した情報をユーザー・プログラム内で使用できます。このインターフェースでは、パラメーター・リスト内に ICETOOL ステートメントを指定します。ICETOOL は、メッセージを TOOLMSG データ・セットに印刷し、ユーザーの呼び出しプログラムが使用できるようにするために、パラメーター・リスト内に操作メッセージおよび『操作特定値』を入れます。

## プログラムからの ICETOOL の呼び出し

図 40 は、パラメーター・リスト・インターフェースで使用されるパラメーター・リストの形式を示したものです。548 ページの表 67 は、呼び出しプログラムに戻される操作特定値を示したものです。

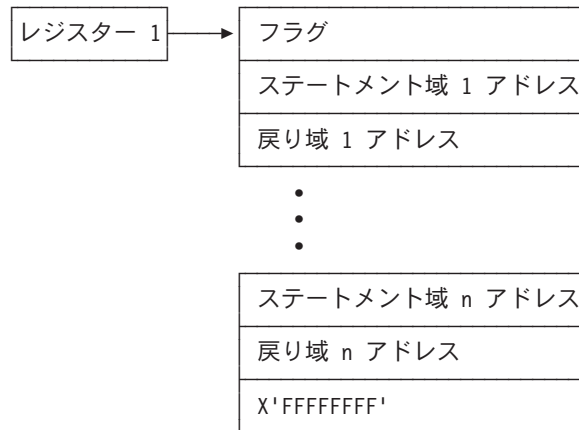


図 40. パラメーター・リスト・インターフェースの場合のパラメーター・リスト

フラグ・フィールドの指定は必須です。パラメーター・リストの終わりを示すために、X'FFFFFFFF' を含む 4 バイト・フィールドを使用します。これは、ステートメント / 戻りアドレスの任意のペアの後でコーディングできます。

パラメーター・リスト内のアドレスはすべて、31 ビット・アドレスまたは消去された 24 ビット・アドレス (最初の 8 ビットにゼロを含む) となります。

## フィールドの説明

### フラグ

#### ビット 0 = 0:

パラメーター・リスト・インターフェースを使用する。このパラメーター・リストからの ICETOOL ステートメントを処理し、情報をこのパラメーター・リストに戻す。TOOLIN を無視する。

#### ビット 0 = 1:

TOOLIN インターフェースを使用する。TOOLIN からの ICETOOL ステートメントを処理する。このパラメーター・リストを無視する。

#### ビット 1 ~ 31:

予約済み。将来拡張できるように、ゼロにセットしなければならない。

### ステートメント域アドレスとステートメント域

各ステートメント域アドレスは、実行される ICETOOL 操作を記述するステートメント域の位置を示します。ステートメント域アドレスが 0 の場合は、ICETOOL はこのステートメント域 / 戻り域のペアを無視します。0 でない場合は、ステートメント域アドレスは、次の形式でステートメント域を指し示す必要があります。

- この操作のステートメント域の長さを含む 2 バイト長のフィールド。このフィールドが 0 の場合は、ICETOOL はこのステートメント域 / 戻り域のペアを無視します。



## プログラムからの ICETOOL の呼び出し

- 454 ページの『ICETOOL ステートメント』に記述されている形式の、1 つ以上の 80 文字 ICETOOL ステートメント・イメージ。各ステートメント域は、1 つの演算子ステートメントをもっていなければならない。演算子ステートメントの前の注釈ステートメントと BLANK ステートメントが処理されます。最初の演算子ステートメントの終わりの後では、注釈、BLANK、および追加の演算子ステートメントは無視されます。

### 戻り域アドレスと戻り域

各戻り域アドレスは、対応するステートメント域に記述された操作について、ICETOOL が操作特定情報を戻す戻り域の位置を示します。戻り域アドレスが 0 の場合は、ICETOOL はこの操作について情報を戻しません。0 でない場合は、戻り域アドレスは次の一般形式で戻り域を指し示している必要があります。

- この操作の戻り域の長さを含む 2 バイト長のフィールド。このフィールドが 0 の場合は、ICETOOL はこの操作についての情報を戻しません。
- ICETOOL によりセットされる、次のような 1 バイトの操作状況表示
  - 0 =** この操作は実行され、戻りコード 0 または 4 で戻されました。操作特有の値 (下記を参照) が戻されます。
  - 4 =** この操作は実行されなかった (たとえば、スキャン・モードが有効になっていた) か、あるいは戻りコード 0 または 4 で完了しませんでした。操作特有の値 (下記を参照) は戻されませんでした。
- 操作特有の値。ICETOOL により戻される各値は、C (正符号の場合) または D (負符号の場合) をもつ 8 バイトのパック 10 進数値です。ICETOOL が操作状況を 4 にセットした場合は、ICETOOL はこの操作で何の値も戻していません。

**注:** ICETOOL を呼び出す LPALIB 内のプログラムは、ICETOOL がその中に保管しておくことのできる戻り域を備えている必要があります。

必要な戻り域の長さ、および各演算子ごとに戻される操作特有の値は、548 ページの表 67 に示されています。戻り域の長さが必要な長さより小さい場合は、ICETOOL はメッセージを出してその操作を終了します。

## プログラムからの ICETOOL の呼び出し

表 67. 戻り域の長さ / 操作特有の値

演算子	戻り域の長さ (バイト)	戻される操作特有の値
COPY	1	なし
COUNT	9	処理されたレコードの数、または基準が指定されている場合 0
DEFAULTS	1	なし
DISPLAY	9	処理されたレコードの数
MODE	1	なし
OCCUR	17	処理されたレコードの数、基準から作成されたレコードの数
RANGE	17	処理されたレコードの数、範囲内の値の数
SELECT	17	処理されたレコードの数、基準から作成されたレコードの数
SORT	1	なし
STATS	32*n+9	処理されたレコードの数、ON フィールド 1 の最小、最大、平均、合計から、ON フィールド n の最小、最大、の平均、合計まで
UNIQUE	17	処理されたレコードの数、固有な値の数
VERIFY	9	処理されたレコードの数

### パラメーター・リスト・インターフェースの例

549 ページの図 41 の例は、パラメーター・リスト・インターフェースを使用するアセンブラー言語プログラムの一部を示したものです。551 ページの表 68 は、549 ページの図 41 でプログラムを実行するために使用できる JCL を示しています。

## プログラムからの ICETOOL の呼び出し

```

DEPTVIEW CSECT
...
* SET UP PARAMETER LIST AND CALL ICETOOL
  LA   R1,PARLST          LOAD ADDRESS OF PARAMETER LIST
  LINK EP=ICETOOL        CALL ICETOOL
  LTR  R15,R15           IF ANY OPERATIONS WERE NOT SUCCESSFUL,
  BNZ  CKSTAT1           DETERMINE WHICH ONE FAILED
* ALL OPERATIONS WERE SUCCESSFUL
* CHECK EMPLOYEES PER DEPARTMENT AGAINST ACCEPTABLE LEVEL
  CP   RT2AVG1,EMAVGCK   IF AVERAGE IS ACCEPTABLE,
  BNH  CKQUAL           NO MESSAGE IS NEEDED
* ISSUE A MESSAGE SHOWING AVERAGE, MINIMUM, MAXIMUM, AND
* TOTAL NUMBER OF EMPLOYEES PER DEPARTMENT.
...
* CHECK EXPENSES PER DEPARTMENT AGAINST ACCEPTABLE LEVEL
CKQUAL CP   RT2AVG2,TLAVGCK IF AVERAGE IS ACCEPTABLE,
  BNH  PCTCALC         NO MESSAGE IS NEEDED
* ISSUE A MESSAGE SHOWING AVERAGE, MINIMUM, MAXIMUM, AND
* TOTAL EXPENSES PER DEPARTMENT.
...
* CALCULATE THE PERCENTAGE OF DEPARTMENTS OVER/UNDER EMPLOYEE LIMIT
PCTCALC MVC  WORK+2(4),RT3RCDS+4 COPY NUMBER OF DEPARTMENTS
  SP   WORK+2(4),RT3RNG+4(4) SUBTRACT 'NUMBER WITHIN LIMITS' TO
*                               GET 'NUMBER OVER/UNDER LIMIT'
  CP   WORK+2(4),P0       IF NONE OVER/UNDER LIMIT,
  BE   PCTPRT            PERCENTAGE IS ZERO
  MP   WORK+2(4),P100    MULTIPLY NUMBER OVER/UNDER BY 100
  DP   WORK(6),RT3RCDS+4(4) DIVIDE BY NUMBER OF DEPARTMENTS
* ISSUE A MESSAGE SHOWING THE PERCENTAGE OF DEPARTMENTS THAT ARE
* OVER/UNDER EMPLOYEE LIMIT
PCTPRT UNPK PCTVAL,WORK(2)   CONVERT AVERAGE TO PRINTABLE EBCDIC
  OI   PCTVAL+2,X'F0'     ENSURE LAST DIGIT IS PRINTABLE
...
* ONE OR MORE OPERATIONS FAILED
CKSTAT1 CLI  RT1STAT,0     IF OPERATION 1 WORKED,
  BNE  CKSTAT2           CHECK OPERATION 2
* ISSUE MESSAGE: OPERATION 1 FAILED - CHECK TOOLMSG
...
* PARAMETER LIST
PARLST DC  A(0)           USE PARAMETER LIST INTERFACE
  DC  A(ST1A)           STATEMENT AREA 1 ADDRESS
  DC  A(RT1A)           RETURN AREA 1 ADDRESS
  DC  A(ST2A)           STATEMENT AREA 2 ADDRESS
  DC  A(RT2A)           RETURN AREA 2 ADDRESS
  DC  A(ST3A)           STATEMENT AREA 3 ADDRESS
  DC  A(RT3A)           RETURN AREA 3 ADDRESS
  DC  F'*. *-1'        END OF PARAMETER LIST* OPERATOR STATEMENT AREAS

```

図 41. ICETOOL パラメーター・リスト・インターフェースの例 (1/2)

## プログラムからの ICETOOL の呼び出し

```

* COPY OPERATION
ST1A  DC  AL2(ST1E-ST1)      LENGTH OF STATEMENT AREA 1
ST1   DC  CL80'* CREATE TWO COPIES OF THE DENVER SITE'
      DC  CL80'* DEPARTMENT RECORDS'
      DC  CL80'COPY FROM(IN) TO(OUT1,OUT2) USING(CTL1)'
ST1E  EQU  *
* STATS OPERATION
ST2A  DC  AL2(ST2E-ST2)      LENGTH OF STATEMENT AREA 2
ST2   DC  CL80'* GET STATISTICS FOR NUMBER OF EMPLOYEES'
      DC  CL80'* AND TRAVEL EXPENSES PER DEPARTMENT'
      DC  CL80'STATS FROM(OUT1) ON(15,2,PD) ON(28,8,ZD)'
ST2E  EQU  *
* RANGE OPERATION
ST3A  DC  AL2(ST3E-ST3)      LENGTH OF STATEMENT AREA 3
ST3   DC  CL80'* DETERMINE THE NUMBER OF DEPARTMENTS THAT ARE'
      DC  CL80'* WITHIN THE LIMIT FOR NUMBER OF EMPLOYEES'
      DC  CL80'RANGE FROM(OUT1) ON(15,2,PD) -'
      DC  CL80'HIGHER(10) LOWER(21) '
ST3E  EQU  *
* RETURN AREAS
  COPY OPERATION
RT1A  DC  AL2(RT1E-RT1STAT)  LENGTH OF RETURN AREA 1
RT1STAT DS  C                OPERATION STATUS
RT1E  EQU  *
* STATS OPERATION
RT2A  DC  AL2(RT2E-RT2STAT)  LENGTH OF RETURN AREA 2
RT2STAT DS  C                OPERATION STATUS
RT2RCDS DS  PL8              COUNT OF RECORDS PROCESSED
RT2MIN1 DS  PL8              FIELD 1 - MINIMUM VALUE
RT2MAX1 DS  PL8              FIELD 1 - MAXIMUM VALUE
RT2AVG1 DS  PL8              FIELD 1 - AVERAGE VALUE
RT2TOT1 DS  PL8              FIELD 1 - TOTAL VALUE
RT2MIN2 DS  PL8              FIELD 2 - MINIMUM VALUE
RT2MAX2 DS  PL8              FIELD 2 - MAXIMUM VALUE
RT2AVG2 DS  PL8              FIELD 2 - AVERAGE VALUE
RT2TOT2 DS  PL8              FIELD 2 - TOTAL VALUE
RT2E  EQU  *
* RANGE OPERATION
RT3A  DC  AL2(RT3E-RT3STAT)  LENGTH OF RETURN AREA 3
RT3STAT DS  C                OPERATION STATUS
RT3RCDS DS  PL8              COUNT OF RECORDS PROCESSED
RT3RNG DS  PL8              COUNT OF VALUES IN RANGE
RT3E  EQU  *
* VARIABLES/CONSTANTS
WORK  DS  PL6                WORKING VARIABLE
P100  DC  P'100'             CONSTANT 100
P0    DC  P'0'               CONSTANT 0
EMAVGCK DC  P'17'           ACCEPTABLE AVERAGE EMPLOYEE COUNT
TLAVGCK DC  P'5000'         ACCEPTABLE AVERAGE TRAVEL EXPENSES
PCTVAL DS  PL3              PERCENTAGE OF DEPARTMENTS THAT ARE
*                               OVER/UNDER EMPLOYEE LIMIT
*                               ...

```

図 41. ICETOOL パラメーター・リスト・インターフェースの例 (2/2)

表 68. パラメーター・リスト・インターフェース・プログラムの JCL の例

```
//EXAMP JOB A402,PROGRAMMER
//INVOKE EXEC PGM=DEPTVIEW,REGION=1024K
//STEPLIB DD DSN=... Link library containing DEPTVIEW
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//IN DD DSN=ALL.DEPTS,DISP=SHR
//OUT1 DD DSN=ALL.DEPTS.BACKUP1,DISP=OLD
//OUT2 DD DSN=ALL.DEPTS.BACKUP2,DISP=OLD
//CTLICNTL DD *
* SELECT ONLY THE DENVER SITE DEPARTMENT RECORDS
  INCLUDE COND=(1,12,CH,EQ,C'DENVER')
/*
```

## ICETOOL に関する注意事項および制約事項

- REGION 値が小さいと、ICETOOL が DFSORT を呼び出すときに記憶域問題を引き起こす可能性があります。REGION 値が大きいと、DFSORT は、最高のパフォーマンスを得るために必要な記憶域を自由に使用できるようになります。少なくとも 1024 キロバイトの REGION 値を ICETOOL に使用することをお勧めします。
- 各 ICETOOL 操作の結果、一組の ICETOOL メッセージが TOOLMSG データ・セット内に、また対応する一組の DFSORT メッセージが DFSMSG データ・セット内に入れられます。DFSORT に対する特定の呼び出しでは、その呼び出しの固有な識別子を使用して、TOOLMSG データ・セットと DFSMSG データ・セット内のメッセージのセットを関連付けることができます。ICETOOL メッセージ ICE606I または ICE627I に印刷された識別子を、DFSORT メッセージ ICE200I に印刷された同一の識別子に一致させます。これは、DFSORT によりエラーが検出された (戻りコード 16) ために ICETOOL 操作が失敗した場合に、特に重要です。
- ICETOOL は DFSORT を呼び出すので、DFSORT に使用されるインストール・オプションは、該当するプログラムにより呼び出された環境に有効なもの、つまり、ICEAM2 または ICEAM4 環境用に起動された、ICEAM2 または ICEAM4 または ICETDx モジュール に有効なものになります。DFSORT インストール・オプションは、DFSORT のみに適用され、ICETOOL には適用されません。たとえば、ICEMAC オプション MSGCON=ALL は、DFSORT (ICETOOL ではなく) が、マスター・コンソールへメッセージを書き出すようにします。ICEMAC のオプション SDBMSG は例外です。ICEAM2 または ICEAM4 からの有効な値は、ICETOOL の TOOLMSG データ・セットに使用されます。
- ICETOOL は、DFSORT を呼び出すと、実行中の特定の操作に特有の制御ステートメントとオプションを渡します。ICETOOL により渡される DFSORT 制御ステートメントまたはオプションを指定変更できません。ただし、指定変更した結果について分かっている場合は、この限りではありません。  
たとえば、ICETOOL は NOABEND オプションを DFSORT に渡して、DFSORT がエラー・メッセージを出した場合に必ず ICETOOL が再び制御を得られるようにします。次のように指定したとします。

```
//DFSPARM DD *
  DEBUG ABEND
```

## ICETOOL に関する注意事項および制約事項

この場合、DFSORT がエラー・メッセージを出すと、DFSORT を異常終了させます。こうして、ICETOOL が後続の演算子を実行しないようにします。

- ICETOOL でテープ作業データ・セットを使用することはできません。
- ON フィールドは、可変長レコードの固定部分の長さを超えるバイト数を含んではなりません。指定されたフィールド全体がすべての入力レコードに入っている必要があります。入っていない場合は、DFSORT はメッセージ ICE015A、ICE218A、または ICE027A を出して終了します。
- SmartBatch パイプ・データ・セットが FROM、TO、LIST、または DISCARD に使用されていて、ICETOOL または DFSORT によりエラーが検出された場合、ABEND が生成されます。これにより、同じ SmartBatch パイプ・データ・セットにアクセスしているかもしれない他のアプリケーションに対して、システムから適切なエラーを伝えることができるようになります。SmartBatch パイプ・データ・セットに関連した、特殊なユーザー異常終了処理については、18 ページの『SmartBatch パイプに関する考慮事項』を参照してください。

DFSORT は、エラーを検出すると、ABCODE インストール・オプションで指示されている、適切な ABEND を出します (DFSORT 導入とカスタマイズ リリース 14 を参照してください)。

ICETOOL がエラーを検出した場合、ABEND 2222 を出します。

---

## ICETOOL の戻りコード

ICETOOL は、STOP または CONTINUE モードで実行する各操作について戻りコードをセットし、検出した最も高い戻りコードをオペレーティング・システムまたは呼び出したプログラムに戻します。

すべての操作が正常終了した場合は、ICETOOL は、戻りコード 0 または 4 をオペレーティング・システムまたは呼び出したプログラムに戻します。

オペレーティング・システムがサポートされていないために正常終了できなかった場合は、ICETOOL は、戻りコード 24 をオペレーティング・システムまたは呼び出したプログラムに戻します。

1 つ以上の操作が正常に終了しなかった場合は、ICETOOL は、戻りコード 12、16、または 20 をオペレーティング・システムまたは呼び出したプログラムに戻します。

ICETOOL が戻す戻りコード (レジスター 15 で) の意味は以下のとおりです。

- 0** 正常終了。すべての操作は正しく完了しました。
- 4** 正常終了。すべての操作は正しく完了しました。DFSORT は 1 つ以上の操作に対して、戻りコード 4 を返しました。詳細については、26 ページの『DFSORT メッセージおよび戻りコード』を参照してください。
- 12** 異常終了。ICETOOL が、正常に終了することを妨げた 1 つ以上のエラーを検出しました。これらのエラー・メッセージは、TOOLMSG データ・セットに印刷されます。
- 16** 異常終了。DFSORT が、ICETOOL の成功終了を妨げた 1 つ以上のエラーを検出しました。これらのエラー・メッセージは、DFSMSG データ・セットに印刷されます。

- 20   メッセージ・データ・セット・エラー。TOOLMSG DD ステートメントが存在しないか、または TOOLMSG データ・セットがオープンされませんでした。
- 24   サポートされていないオペレーティング・システム。このオペレーティング・システムは、このリリースの DFSORT によりサポートされていません。

|  
|  
|

## ICETOOL の戻りコード



---

## 第 7 章 フィールドと定数のシンボルの使用

フィールドおよび定数シンボルの概説 . . . . .	555
DFSORT の例 . . . . .	556
SYMNAMES DD ステートメント . . . . .	558
SYMNOUT DD ステートメント . . . . .	559
SYMNAMES ステートメント . . . . .	559
注釈ステートメントとブランク・ステートメント . . . . .	559
シンボル・ステートメント . . . . .	559
キーワード・ステートメント . . . . .	567
SYMNAMES ステートメントのチェックのための SYMNOUT の使用 . . . . .	569
DFSORT ステートメントでのシンボルの使用 . . . . .	570
SORT と MERGE . . . . .	571
SUM . . . . .	571
INCLUDE と OMIT . . . . .	572
INREC と OUTREC . . . . .	572
OUTFIL . . . . .	573
ICETOOL 演算子でのシンボルの使用 . . . . .	575
COUNT . . . . .	575
DISPLAY . . . . .	576
OCCUR . . . . .	576
RANGE . . . . .	576
SELECT . . . . .	576
STATS、UNIQUE および VERIFY . . . . .	576
ICETOOL の例 . . . . .	577
シンボルに関する注意事項 . . . . .	578

---

### フィールドおよび定数シンボルの概説

この章では、DFSORT および ICETOOL ステートメントでシンボルを使用する、DFSORT の簡単で柔軟な方法を説明します。DFSORT 制御ステートメントまたは ICETOOL 演算子で認識される、どんなフィールドまたは定数にも、シンボルを定義して使用できます。シンボルを使用すると、さまざまなレコード・レイアウトに関連した情報を表すシンボルの集合を作成して、繰り返し使用 (つまり、マッピング) できます。

また、他の製品 (たとえば、RACF、DFSMSrmm、DCOLLECT など)、またはユーザーのサイトで作成されたレコードに対して専用の DFSORT シンボルを作成して、そのシンボルを集合で入手または使用できます。次に示す URL の DFSORT ホーム・ページにアクセスすると、他の製品により作成されたレコードのための DFSORT シンボル・マッピングのダウンロードに関する情報と、シンボルの使用例を入手できます。

<http://www.ibm.com/storage/dfsorrt/>

DFSORT または ICETOOL を使って特定レコードを処理する場合、シンボルを使用すると、そのレコードに関連するフィールドの位置、長さ、形式や、定数の値が自動的に指定され、生産性を上げることができます。

DFSORT または ICETOOL でシンボルを使用するには、次の手順に従います。

## フィールドと定数のシンボルの使用

1. 必要な DFSORT シンボル・データ・セットを、作成または取得します。シンボル・データ・セットには、レコード内のフィールドをマップするシンボルと、比較に使用する定数、表題、ヘッダーなどが入れられます。シンボルは DFSORT の簡単で、しかも柔軟な SYMNAMES ステートメントの形式で指定されます。シンボルは、ISPF EDIT などのエディターを使用して、簡単に追加または変更できます。
2. SYMNAMES DD ステートメントをジョブに組み込みます。SYMNAMES には、DFSORT または ICETOOL アプリケーションが使用する、1 つ以上のシンボル・データ・セット (順次、区分メンバー、DD \*) を指定します。SYMNAMES を使用して、シンボル・データ・セットをいくつでも連結できます。
3. SYMNAMES のシンボルを、DFSORT 制御ステートメントまたは ICETOOL 演算子の適切な個所で使用します。シンボル (たとえば、Last\_Name) を、通常のフィールド (たとえば、p、m、f) および定数 (たとえば、C'string') と混合して使用できます。

DFSORT または ICETOOL は SYMNAMES を読み取り、その中に含まれるシンボルを使用してシンボル置換を実行することにより、ステートメントを「変換」します。DFSORT または ICETOOL はその後、変換されたステートメントを、ユーザーが直接指定するのと同様に使用します。

レコード・レイアウトが変わったときは、それと同じ変更を DFSORT シンボル・データ・セットに加えるだけです。DFSORT は、位置が変わっていても、新しいマッピングを使用してシンボルを正しく「変換」します。したがって、ステートメントを変更する必要はありません。シンボル定義がレコード・レイアウトに一致していることを確認した後、それらを使用してください。

## DFSORT の例

下記の例は、シンボルを使用する簡単な DFSORT ジョブの、JCL および制御ステートメントを示しています。

次の SYMNAMES ステートメントを含む、MY.SYMBOLS という名前のシンボル・データ・セットを作成したとします。

```
* Fields
First_Name,6,20,CH
Last_Name,*,=,=
Account_Number,53,3,PD
SKIP,2
Balance,*,6,ZD
Type,*,8,CH

* Constants
Loan,'LOAN'
Check,'CHECKING'
Level1,50000
Level2,-100
```

この例のための JCL および制御ステートメントを示します。

```
//EXAMP JOB A402,PROGRAMMER
//RUNIT EXEC PGM=ICEMAN
//SYMNAMES DD DSN=MY.SYMBOLS,DISP=SHR
//SYMNOUT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SORTIN DD ...
```

```
//SORTOUT DD ...
//SYSIN DD *
INCLUDE COND=((Type,EQ,Loan,AND,Balance,GT,Level1),OR,
              (Type,EQ,Check,AND,Balance,LE,Level2))
SORT FIELDS=(Last_Name,A,First_Name,A,
             Type,A,Account_Number,D)
/*
```

この例は、シンボルの使い方の概要を簡単に示すことだけを目的としています。詳細についてはすべてこの章の他の項で説明されていますが、注意が必要な重要事項のいくつかをここで示します。

- SYMNames DD は、DFSORT または ICETOOL にシンボル処理を行うことを指示します。SYMNames データ・セットに、フィールドと定数のシンボルがあります。
- SYMNOUT を指定すると、DFSORT または ICETOOL は、元のシンボルと、元のシンボルから構成したシンボル・テーブルを、SYMNOUT データ・セットに印刷します。一連のシンボルのデバッグ時に SYMNOUT を使用して、後で削除できます。あるいは、SYMNOUT を永続的に保持して、常に元のシンボルとシンボル・テーブルを見ることができるようにもできます。
- 簡単で、しかも柔軟な SYMNames ステートメントの形式を次に示します。

```
symbol,value remark
```

ここで、value はフィールド (p,m,f または p,m または p) または定数 (C'string'、c'string'、'string'、X'string'、x'string'、B'string'、b'string'、n、+n または -n) を表すことができます。シンボルの前に先行ブランクを置き、字下げを使用できます。たとえば、次の SYMNames ステートメントを指定できます。

```
Div1_Department,8,1,BI      Division 1 Department
Research,B'0001....'      Research Departments
Marketing,B'0010....'      Marketing Departments
Development,B'0100....'    Development Departments
```

- シンボルは大文字小文字を区別します。Frank、FRANK、frank は、3つの異なるシンボルです。
- 次の例のように、アスタリスク (\*) を使用して 次の位置 を p に割り当てることができます。

```
Symbola,6,20,CH
Symbolb,*,5,BI
Symbolc,*,12,ZD
```

これは、次の指定と同じです。

```
Symbola,6,20,CH
Symbolb,26,5,BI
Symbolc,31,12,ZD
```

p に \* を使用すると、レコード内の連続したフィールドを、実際の位置を計算することなくマップできます。

- SKIP,n を使用して、 次の位置 を n バイト進めて、\* を使用できるようにできます。たとえば、下記のようにします。

```
Symbola,6,20,CH
SKIP,2
Symbolb,*,5,BI
```

これは、次の指定と同じです。

## フィールドと定数のシンボルの使用

```
Symbola,6,20,CH  
Symbolb,28,5,BI
```

SKIP,n を使用すると、未使用のバイトを簡単にスキップできます。他のマッピング補助機能により、次の位置をリセットしたり (POSITION,q または POSITION,symbol)、次の位置をハーフワード (ALIGN,H)、フルワード (ALIGN,F)、またはダブルワード (ALIGN,D) に位置合わせできます。

- p、m または f に等号 (=) を使用して、p、m または f に、直前の位置、長さ、または形式を割り当てることができます。たとえば、下記のようにします。

```
Symbola,6,20,CH  
Symbola1,=,8,=  
Symbola2,*,12,=  
Symbold,*,=,ZD
```

これは、次の指定と同じです。

```
Symbola,6,20,CH  
Symbola1,6,8,CH  
Symbola2,14,12,CH  
Symbold,26,12,ZD
```

= と \* を使用することにより、フィールドを他のフィールドに簡単にマップできます。

- フィールドおよび定数のシンボルは、どのような順番でも指定できます。ただし、\* と = を使用すると、フィールドのシンボルは順番に依存するようになります。
- SYMNames に、注釈ステートメントとブランク・ステートメントも指定できます。

---

## SYMNames DD ステートメント

SYMNames DD ステートメントにより、DFSORT または ICETOOL にシンボル処理を行うことを指示します。SYMNames DD ステートメントは、SYMNames データ・セット (省略形は SYMNames) を指定します。SYMNames データ・セットは、1 つの DFSORT シンボル・データ・セット、または連結した多数のシンボル・データ・セットにより構成できます。

シンボル・データ・セットには、順次データ・セット、区分メンバー、または DD \* データ・セットのいずれも使用できます。この 3 タイプをすべて連結して、SYMNames DD に指定できます。各シンボル・データ・セットは、DFSORT または ICETOOL アプリケーションが使用する、フィールドおよび定数のシンボルを記述した、SYMNames ステートメントを含んでいる必要があります。各シンボル・データ・セットの属性は、RECFM=F または RECFM=FB で LRECL=80 である必要があります。

最高のパフォーマンスを得るために、すべての DFSORT シンボル・データ・セットについて、システム決定の最適ブロック・サイズのような、大きいブロック・サイズを使用するようにしてください。

SYMNames DD ステートメントを指定しない場合、あるいは SYMNames が空の場合は、シンボル処理は実行されません。

## SYMNOUT DD ステートメント

SYMNOUT DD ステートメントは、DFSORT または ICETOOL が、元の SYMNames ステートメント、またはそれから構成されたシンボル・テーブルを印刷するデータ・セットを指定します。DFSORT または ICETOOL は、SYMNOUT データ・セット (省略形は SYMNOUT) について、RECFM=FBA、LRECL=121、および指定の BLKSIZE を使用します。

指定した BLKSIZE が 121 の倍数でない場合、あるいは BLKSIZE を指定しない場合は次のようになります。

- システム決定の最適ブロック・サイズが使用されます (サポートされている場合)。
- それ以外の場合は、BLKSIZE=121 が使用されます。

最高のパフォーマンスを得るために、SYMNOUT データ・セットについて、システム決定の最適ブロック・サイズのような、大きいブロック・サイズを使用するようにしてください。

## SYMNames ステートメント

SYMNames 内の各シンボルは、SYMNames ステートメントを使用して記述する必要があります。SYMNames ステートメントは、シンボル・ステートメント、キーワード・ステートメント、注釈ステートメント、またはブランク・ステートメントで指定できます。

## 注釈ステートメントとブランク・ステートメント

1 桁目がアスタリスク (\*) のステートメントは、注釈ステートメントとして扱われます。注釈ステートメントは SYMNOUT に印刷されますが (指定されている場合)、それ以外の処理はされません。1 ~ 80 桁がブランクのステートメントは、ブランク・ステートメントとして扱われます。注釈ステートメントは SYMNOUT に印刷されますが (指定されている場合)、それ以外の処理はされません。

## シンボル・ステートメント

シンボル・ステートメントの一般形式は、次のとおりです。

```
symbol,value remark
```

一般コーディング規則は、次のとおりです。

- 1 ~ 80 桁がスキャンされる。
- シンボルは、1 桁目あるいはそれ以降の任意の桁から開始できる。
- 注釈はオプションだが、指定する場合は少なくとも 1 つのブランクでその値と区切る必要がある。注釈は、SYMNOUT に印刷されるが (指定されている場合)、それ以外の処理はされない。
- シンボルと値の間には、コンマ (,) の代わりにセミコロン (;) も使用できる。
- 継続行は使用できない。各シンボルと値は、完全に 1 つの行に指定する必要がある。

## フィールドと定数のシンボルの使用

シンボル・ステートメントの特定構文を次に示します。

▶ `symbol`, `constant`  
          └── `field` ─┘

**シンボル:** シンボルは 1 から 50 までの EBCDIC 文字で、大文字 (A ~ Z)、小文字 (a ~ z)、数字 (0 ~ 9)、番号記号 (#)、ドル記号 (\$)、アットマーク (@)、下線 (\_)、およびハイフン (-) が使用できます。シンボルの先頭文字に数字またはハイフンは使用できません。シンボルは大文字小文字を区別します。  
Frank、FRANK、frank は、3 つの異なるシンボルです。

次の DFSORT/ICETOOL 予約語 (英大文字のみ) はシンボルとして使用できません。A、AC、ALL、AND、AQ、ASL、AST、BI、CH、CLO、COPY、COUNT、COUNT15、CSF、CSL、CST、CTO、D、DATE、DATE1、DATE1P、DATE2、DATE2P、DATE3、DATE3P、DT1、DT2、DT3、D1、D2、E、F、FI、FL、FS、H、HEX、LS、Mn、Mnn、NONE、NUM、OL、OR、OT、PAGE、PAGEHEAD、PD、PD0、SEQNUM、SS、SUBCOUNT、SUBCOUNT15、TIME、TIME1、TIME1P、TIME2、TIME2P、TIME3、TIME3P、TM1、TM2、TM3、TM4、TS、VALCNT、VLEN、X、Y2x、Y2xx、Z および ZD。ただし、n は 0 ~ 9、x は任意の文字とします。これらのワードの小文字および大文字小文字が混合したフォーム (None や page など) は、シンボルとして使用できます。

POSITION、SKIP および ALIGN (英大文字のみ) は、567 ページの『キーワード・ステートメント』で説明するようにキーワードとして扱われるため、シンボルとして認識されません。ただし、Position や skip など、これらのワードの小文字および大文字小文字が混合したフォームはシンボルとして使用できます。

Account\_Number、CON12、PHONE#、count、SORT-KEY および \_Invalid は、有効なシンボルの例です。

123\_Account (先頭文字が数字)、COUNT (予約語)、および -Invalid (1 文字目はハイフン) は、無効なシンボルの例です。

**定数:** 定数は、文字ストリング、16 進数ストリング、ビット・ストリング、10 進数、または 2 桁年の日付ストリングで指定できます。

定数値のシンボルは、その定数が使用でき、期待される結果が得られる場合のみ使用してください。それ以外の場合にシンボルを定数に置換すると、エラー・メッセージが出力されたり、意図しない処理が行われることがあります。たとえば、次の SYMNames ステートメントを指定する場合があります。

```
SYMB,B'10110001'
```

、SYMB は、次の INCLUDE ステートメントに使用できます。

```
INCLUDE COND=(12,1,BI,EQ,SYMB)
```

ビット・ストリングと 2 進数フィールドは、比較できるからです。しかし、SYMB を次の INCLUDE ステートメントに使用するとエラー・メッセージが出されます。

```
INCLUDE COND=(12,1,CH,EQ,SYMB)
```

ビット・ストリングと文字フィールドは、比較できないためです。

シンボルに置換される定数が、適切であることを確認してください。疑わしいときは、関係のあるオペランドの説明に示されている、定数の規則をチェックしてください。

シンボルには、次に示すタイプの定数を指定できます。

- 形式 'xx...x'、C'xx...x' または c'xx...x' 形式の文字ストリング。

値 x は任意の EBCDIC 文字です。文字ストリングには、最高 64 文字まで指定できます。c'xx...x' は C'xx...x' と同じ扱いになります。

文字ストリング内に単一のアポストロフィを組み込む場合は、2 つの単一アポストロフィとして指定します (アポストロフィの各ペアは、文字ストリングの 64 文字制限に対して、2 文字としてカウントされます)。したがって、次のようになります。

```
Required: 0'Neill          Specify: C'0'Neill'
```

2 バイト・データを文字ストリング内で使用できます (シフトイン / シフトアウト文字の各ペアと各 2 バイト文字は、文字ストリングの 64 文字制限に対して 2 文字としてカウントされます)。2 バイト・データの詳細については、130 ページの『INCLUDE 制御ステートメント』を参照してください。

'+0.193'、c'Title'、C'O'Neil'、C'J62,J82,M72'、および '' は、有効な文字ストリングの例です。

C'AB'' (アポストロフィがペアになっていない)、c'title (終了のアポストロフィがない) および C'O'NEIL' (O の後に 2 つ必要なアポストロフィが 1 つしかない) は、無効な文字ストリングの例です。

C'xx...x' と 'xx...x' は交換可能です。SYMNAMES に C'xx...x' が指定されていても、シンボルは適切に 'xx...x' に置換されます。同じように、SYMNAMES に 'xx...x' が指定されていても、シンボルは適切に C'xx...x' に置換されます。たとえば、次の SYMNAMES ステートメントが指定されているものとします。

```
My_Title,c'My Report'
My_Heading,C'January'
DEPT1,'J82'
DEPT2,c'M72'
```

次の ICETOOL 演算子は、

```
DISPLAY TITLE(My_Title) HEADER(My_Heading) ...
```

次のように変換されます。

```
DISPLAY TITLE('My Report') HEADER('January') ...
```

次の INCLUDE ステートメントは、

```
INCLUDE COND=(5,3,EQ,DEPT1,OR,5,3,EQ,DEPT2),FORMAT=CH
```

次のように変換されます。

```
INCLUDE COND=(5,3,EQ,C'J82',OR,5,3,EQ,C'M72'),FORMAT=CH
```

シンボルとして使用される文字ストリングの規則は、一般に INCLUDE/OMIT 文字ストリングの規則に従いますが、同じ規則がすべての DFSORT および ICETOOL オペランドに適用できるわけではないことに注意し、適切な文字スト

## フィールドと定数のシンボルの使用

リングを表すシンボルを使用するようにしてください。たとえば、ICETOOL では TITLE 文字ストリングに 50 文字までしか使用できないため、TITLE(MYCON) は、MYCON が 64 桁の文字ストリングである場合はエラーになります。その場合に MYCON を INCLUDE ステートメントに使用してもエラーにはなりません。もう 1 つ例をあげると、シンボルから置換された文字ストリング内の 2 バイト文字は、INCLUDE ステートメント内では認識されますが、OUTREC ステートメント内では認識されません。

- 形式 X'yy...yy' または x'yy...yy' の 16 進数ストリング。

値 yy は、16 進数の任意の組み合わせを示します。各 16 進数字は、0 ~ 9、A ~ F または a ~ f である必要があります。最大 32 組の 16 進数が指定できます。x'yy...yy' は X'yy...yy' と同じ扱いになります。a ~ f は A ~ F と同じ扱いになります。

X'F2C3'、x'2fa71e'、および X'07' は有効な 16 進数ストリングの例です。

X'F2G301' (G は有効な 16 進数字ではない)、x'bf3' (16 進数字が対になっていない)、および X' (16 進数字が指定されていない) は、無効な 16 進数ストリングの例です。

- 形式 B'bbbbbbbb...bbbbbbbb' または b'bbbbbbbb...bbbbbbbb' のビット・ストリング。

値 bbbbbbbb は、1 バイトを構成する 8 ビットを表します。各ビットは、1、0、または . (ピリオド) である必要があります。最大 8 までの 8 ビットのグループを指定できます。b'bbbbbbbb...bbbbbbbb' は B'bbbbbbbb...bbbbbbbb' と同じ扱いになります。

B'01100100'、b'11..00..000..111'、および B'11.....' は有効なビット・ストリングの例です。

B'0101' (8 ビットの倍数になっていない)、b'00..11....' (8 ビットの倍数になっていない)、b'00000002' (2 は有効なビットではない)、および B'' (ビットが指定されていない) は、無効なビット・ストリングの例です。

- 形式 n、+n または -n の 10 進数。1 から 15 桁までの有効数字を指定できます。

+270、270、000036、+0、および -2000000 は有効な 10 進数の例です。

++15 (正符号が多すぎる)、280- (符号の位置が違う)、および 2.8 (ピリオドは使用できない) は、無効な 10 進数の例です。

- 形式 Y'string' または y'string' の 2 桁年の日付ストリング。

ストリングには次の値を指定できます。

– yy、yyx,yyxx、yyxxx または yyxxxx。y は 16 進数の年を表す桁 (0 ~ 9)、x は 16 進数の年以外の桁 (0 ~ 9)。

– 大文字、小文字、または大文字小文字混合で指定された

DATE1、DATE2、DATE3、LOW、BLANKS または HIGH の形式。

Y'99'、y'00123'、y'date2'、および Y'Blanks' は、有効な 2 桁年日付ストリングの例です。

Y'9'、y'AB123' および Y'blank' は、無効な 2 桁年日付ストリングの例です。



**フィールド:** フィールドは、p,m,f (位置、長さ、および形式)、p,m (位置および長さ)、または p (位置のみ) として指定できます。

フィールド値のシンボルは、そのフィールドが使用でき、期待される結果が得られる場合のみ使用してください。それ以外の場合にシンボルをフィールドに置換すると、エラー・メッセージが出力されたり、意図しない処理が行われることがあります。たとえば、次の SYMNames ステートメントを指定する場合を考えます。

```
Field1,15,2,CH
```

Field1 を、次のように SORT ステートメントで使用できます。

```
SORT FIELDS=(Field1,A)
```

文字フィールドは、SORT ステートメント内で使用できるからです。しかし、Field1 を次の SUM ステートメントに使用するとエラー・メッセージが出されます。

```
SUM FIELDS=(Field1)
```

文字フィールドは、SUM ステートメント内で使用できないからです。

シンボルに置換されるフィールドが、適切であることを確認してください。疑わしいときは、関係のあるオペランドの説明に示されている、p, m および f の規則をチェックしてください。

p,m,f を指定したフィールド・シンボルを、p,m が必要とされているところに使用できます。DFSORT または ICETOOL が、p,m だけを適切に置換するからです。たとえば、SYMNames に次のように指定するものとします。

```
First_Field,12,2,BI
Second_Field,18,6,CH
Third_Field,28,5,PD
Fourth_Field,36,3
Fifth_Field,52,4,PD
Max,200000
```

次の DFSORT 制御ステートメントは、

```
OMIT COND=(Fifth_Field,GT,Max)
SORT FIELDS=(First_Field,A,Fourth_Field,A),FORMAT=CH
SUM FIELDS=(Second_Field,ZD)
OUTFIL OUTREC=(First_Field,2X,Third_Field,M11,Fourth_Field)
```

次のように変換されます。

```
OMIT COND=(52,4,PD,GT,200000)
SORT FIELDS=(12,2,A,36,3,A),FORMAT=CH
SUM FIELDS=(18,6,ZD)
OUTFIL OUTREC=(12,2,2X,28,5,PD,M11,36,3)
```

DFSORT が次の置換を行ったことに注意してください。

- OMIT ステートメント: Fifth\_Field の p,m,f を、COND の要求に従って FORMAT なしで置換。
- SORT ステートメント: First\_Field と Fourth\_Field の p,m を、FIELDS の要求に従って FORMAT なしで置換。
- SUM ステートメント: Second\_Field の p,m を、symbol,f (つまり、Second\_Field,ZD) の要求に従って置換。

## フィールドと定数のシンボルの使用

- **OUTFIL** ステートメント: **First\_Field** の **p,m** は、**OUTREC** オペランドの無編集フィールド (つまり、**First\_Field**) の要求に従って置換。しかし、**Third\_Field** の **p,m,f** は、**OUTREC** オペランドの編集フィールド (つまり、**Third\_Field,M11**) の要求に従って置換。

**p**、**m** および **f** をシンボル・ステートメントで使用する際の一般規則は、次のとおりです。

- **p** には、数字、アスタリスク (\*) または等号 (=) を使用できます。**p,m** または **p,m,f** には、1 から 32752 までの数字を指定できます。**p** (位置のみ) は、定数 **n** と区別できないため、1 から 15 桁までの有効数字を指定できます。

アスタリスク (\*) を使用して、次の位置を **p** に割り当てることができます。

**p,m,f** または **p,m** が指定されたシンボルが読み取られるたびに、次の位置が **p+m** に設定されます。また、次の位置は、キーワード・ステートメントで変更できます (567 ページの『キーワード・ステートメント』参照)。**p** に \* を指定すると、次の位置が **p** に割り当てられます。次の位置が設定されていないときに、**p** に \* を指定すると (たとえば、最初のシンボルに \* を使用したときなど)、**p** は 1 に設定されます。

**SYMNOUT** データ・セット (指定されている場合) に印刷されたシンボル・テーブルを見ると、**p** に \* を指定した場合に割り当てられた、実際の位置がわかります。

\* の使用例として、次の **SYMNAMES** ステートメントを指定した場合を考えます。

```
Sym1,*,5,ZD
Con1,27
Sym2,*,2,BI
Field1,8,13,CH
Field2,*,5,PD
Field3,*,2,FI
```

**SYMNOUT** には、次のシンボル・テーブルが示されます。

```
Sym1,1,5,ZD
Con1,27
Sym2,6,2,BI
Field1,8,13,CH
Field2,21,5,PD
Field3,26,2,FI
```

**p** に \* を使用すると、レコード内の連続したフィールドを、実際の位置を計算することなくマップできます。また、他のフィールドの間に追加されたフィールドを、元のフィールドまたは挿入されたフィールドの **p** の値を変更することなくマップすることもできます。\* は、連結されたシンボル・データ・セットを使用して、連続するフィールドのマッピングを作成するときにも便利です。簡単な例として、次のように指定した場合を考えます。

```
//SYMNAMES DD DSN=MY.SYMPDS(RDW),DISP=SHR
//          DD DSN=MY.SYMPDS(SECTION1),DISP=SHR
//          DD DSN=MY.SYMPDS(SECTION2),DISP=SHR
```

**RDW** メンバーの内容は、次のとおりです。

```
RDW,1,4,BI
```

**SECTION1** メンバーの内容は、次のとおりです。

```
Flag_Byte,*,1,BI
  Error1,X'80'
  Error2,X'40'
Count_of_Parts,*,5,ZD
```

SECTION2 メンバーの内容は、次のとおりです。

```
New_Parts,*,5,ZD
Old_Parts,*,5,ZD
Variable_Fields,*
```

SYMNOUT には、次のシンボル・テーブルが示されます。

```
RDW,1,4,BI
Flag_Byte,5,1,BI
Error1,X'80'
Error2,X'40'
Count_of_Parts,6,5,ZD
New_Parts,11,5,ZD
Old_Parts,16,5,ZD
Variable_Fields,21
```

これらのシンボルは、たとえば次のステートメントで使用できます。

```
OPTION COPY
OUTFIL FNAMES=ERR1,INCLUDE=(Flag_Byte,EQ,Error1),
      OUTREC=(RDW,Count_of_Parts,Variable_Fields)
OUTFIL FNAMES=ERR2,INCLUDE=(Flag_Byte,EQ,Error2),
      OUTREC=(RDW,New_Parts,Old_Parts,Variable_Fields)
```

等号 (=) を使用して、直前の位置 を p に割り当てることができます。p,m,f または p,m が指定されたシンボルが読み取られるたびに、直前の位置 が p に設定されます。また、直前の位置 は、POSITION キーワード・ステートメントにより変更できます (下記参照)。p に = を指定すると、直前の位置 が p に割り当てられます。直前の位置 が設定されていないときに、p に = を指定すると、エラー・メッセージが出されます。

SYMNOUT データ・セット (指定されている場合) に印刷されたシンボル・テーブルを見ると、p に = を指定した場合に割り当てられた、実際の位置がわかりません。

p に = を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
Sym1,5,4,CH
  Sym2,=,2,CH
  Sym3,*,2,CH
```

SYMNOUT には、次のシンボル・テーブルが示されます。

```
Sym1,5,4,CH
Sym2,5,2,CH
Sym3,7,2,CH
```

p に = と \* を使用することにより、フィールドを他のフィールドに簡単にマップできます。

p に = を使用するときは、直前の位置 が、希望する位置に設定されていることを必ず確認してください。特に、p に = を使用しているシンボルの前に現在の直前の位置と異なる直前の位置をもつ新しいフィールド・シンボルを挿入するときは、= を希望する実際の位置への変更が必要になります。

## フィールドと定数のシンボルの使用

- **m** には、等号 (=) または 1 から 32752 までの数字を使用できます。等号 (=) を使用して、直前の長さを **m** に割り当てることができます。p,m,f または p,m が指定されたシンボルが読み取られるたびに、直前の長さが **m** に設定されます。**m** に = を指定すると、直前の長さが **m** に割り当てられます。直前の長さが設定されていないときに **m** に = を指定すると、エラー・メッセージが出されます。

SYMNOUT データ・セット (指定されている場合) に印刷されたシンボル・テーブルを見ると、**m** に = を指定した場合に割り当てられた、実際の長さがわかります。

**m** に = を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
Flags1,5,1,BI
Error1,X'08'
Flags2,15,=,BI
Error2,X'04'
Flags3,22,=,BI
Error3,X'23'
```

SYMNOUT には、次のシンボル・テーブルが示されます。

```
Flags1,5,1,BI
Error1,X'08'
Flags2,15,1,BI
Error2,X'04'
Flags3,22,1,BI
Error3,X'23'
```

**m** に = を使用するときは、直前の長さが希望する長さに設定されていることを必ず確認してください。特に、**m** に = を使用しているシンボルの前に、現在の直前の長さとは異なる長さをもつ新しいフィールド・シンボルを挿入するときは、= を希望する実際の長さへの変更が必要になります。

- **f** には、等号 (=) か、次の形式のいずれかを指定できます。AC、AQ、ASL、AST、BI、CH、CLO、CSF、CSL、CST、CTO、DT1、DT2、DT3、D1、D2、FI、FL、FS、LS、OL、OT、PD、PD0、SS、TS、Y2B、Y2C、Y2D、Y2DP、Y2P、Y2PP、Y2S、Y2T、Y2TP、Y2U、Y2UP、Y2V、Y2VP、Y2W、Y2WP、Y2X、Y2XP、Y2Y、Y2YP、Y2Z、または ZD。

**f** は、大文字 (たとえば、CH)、小文字 (たとえば、ch)、または大文字小文字の混合 (たとえば、Ch) で指定できます。**f** は、どの文字で指定しても英大文字として扱われます。

等号 (=) を使用して、直前の形式を **f** に割り当てることができます。p,m,f が指定されたシンボルが読み取られるたびに、直前の形式が **f** に設定されます。**f** に = を指定すると、直前の形式が **f** に割り当てられます。直前の形式が設定されていないときに **f** に = を指定すると、エラー・メッセージが出されます。

SYMNOUT データ・セット (指定されている場合) に印刷されたシンボル・テーブルを見ると、**f** に = を指定した場合に割り当てられた実際の形式がわかります。

**f** に = を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。



## フィールドと定数のシンボルの使用

POSITION,q を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
POSITION,27
Account_Balance,*,5,PD
Account_Id,*,8,CH
POSITION,84
New_Balance,=,20
```

SYMOUT には、次のシンボル・テーブルが示されます。

```
Account_Balance,27,5,PD
Account_Id,32,8,CH
New_Balance,84,20
```

- **POSITION,symbol** を使用すると、次の位置 と 直前の位置 を、指示されたシンボルに対して設定されている位置に設定できます。上記の p の項で説明したように、シンボル・ステートメントの p に、アスタリスク (\*) を指定すると 次の位置 が使用され、等号 (=) を指定すると 直前の位置 が使用されます。POSITION,symbol を使用するときは、次のシンボルの p に \* または = のどちらかを使用できます。

**symbol** には、すでに定義済みの任意のフィールド・シンボルを指定できます。そのため、POSITION,symbol をアセンブラの ORG 命令のように使用して、別々のフィールドを同じエリアにマップできます。

POSITION,symbol を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
Workarea,21,100      Use workarea for volsers
  volser1,=,6,CH
  volser2,*,6,CH
POSITION,Workarea   Reuse workarea for status and dsname
  status,=,1,BI
  dsname,*,44,CH
```

SYMOUT には、次のシンボル・テーブルが示されます。

```
Workarea,21,100
volser1,21,6,CH
volser2,27,6,CH
status,21,1,BI
dsname,22,44,CH
```

- **SKIP,n** を使用すると、次の位置 に n バイトを加算できます。上記の p の項で説明したように、シンボル・ステートメントの p にアスタリスク (\*) を指定すると、次の位置 が使用されます。n には 1 から 32752 までの数字を指定できません。

SKIP,n を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
Field#1,15,6,FS
  SKIP,4   Unused bytes
Field#2,*,5,=
  SKIP,2   Unused bytes
Field#3,*,8,CH
```

SYMOUT には、次のシンボル・テーブルが示されます。

```
Field#1,15,6,FS
Field#2,25,5,FS
Field#3,32,8,CH
```

- **ALIGN,H** を使用すると、次の位置 をハーフワード境界上、つまり 1、3、5 などに位置合わせできます。上記の p の項で説明したように、シンボル・ステートメントの p にアスタリスク (\*) を指定すると、次の位置 が使用されます。  
ALIGN,h は ALIGN,H と同じ扱いになります。

ALIGN,H を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
A1,7,3,CH
ALIGN,H
A2,*,2,BI
```

SYMOUT には、次のシンボル・テーブルが示されます。

```
A1,7,3,CH
A2,11,2,BI
```

- **ALIGN,F** を使用すると、次の位置 をフルワード境界上、つまり 1、5、9 などに位置合わせできます。上記の p の項で説明したように、シンボル・ステートメントの p にアスタリスク (\*) を指定すると、次の位置 が使用されます。ALIGN,f は ALIGN,F と同じ扱いになります。

ALIGN,F を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
B1,7,3,CH
ALIGN,f
B2,*,4,BI
```

SYMOUT には、次のシンボル・テーブルが示されます。

```
B1,7,3,CH
B2,13,4,BI
```

- **ALIGN,D** を使用すると、次の位置 をダブルワード境界上、つまり 1、9、17 などに位置合わせできます。上記の p の項で説明したように、シンボル・ステートメントの p にアスタリスク (\*) を指定すると、次の位置 が使用されます。  
ALIGN,d は ALIGN,D と同じ扱いになります。

ALIGN,D を使用する方法の例として、次の SYMNames ステートメントを指定した場合を考えます。

```
C1,7,3,CH
ALIGN,D
C2,*,8,BI
```

SYMOUT には、次のシンボル・テーブルが示されます。

```
C1,7,3,CH
C2,17,8,BI
```

## SYMNames ステートメントのチェックのための SYMOUT の使用

不測の事態を避けるために、SYMNames ステートメントを DFSORT または ICETOOL ステートメントで使用する前に、位置、長さ、および形式に誤りがないかチェックするのはよい考えです。

次の簡単なジョブを実行すると、DFSORT が SYMNames ステートメントで検出したエラーに関するメッセージを SYSOUT に出力することができ、処理を続行する前にエラーを訂正できます。SYMNames ステートメントにエラーがなければ、

## フィールドと定数のシンボルの使用

このジョブにより、DFSORT はシンボル・テーブルを SYMNOUT に出力するので、(たとえば、\*、=、SKIP などの誤った使用による) シンボルの不正な位置、長さ、または形式を訂正できます。

```
//CHECK JOB A402,PROGRAMMER
//DOIT EXEC PGM=ICEMAN
//SYMNAMES DD ...          SYMNAMES statements to be checked
//SYMNOUT DD SYSOUT=*
//SORTIN DD *
//SORTOUT DD DUMMY
//SYSIN DD *
      OPTION COPY
/*
```

SYMNAMES ステートメントの「デバッグ」が完了したら、DFSORT および ICETOOL ステートメントで使用できます。

---

## DFSORT ステートメントでのシンボルの使用

INCLUDE、INREC、MERGE、OMIT、OUTFIL、OUTREC、SORT、および SUM の各 DFSORT 制御ステートメントでシンボルを使用できます。一般に、定数 ('string'、C'string'、X'string'、B'string'、n、+n、または -n)、およびフィールド (p,m,f または p,m または p) を使用できる DFSORT ステートメントに、シンボルを使用できます。構文の詳細については、111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』の各制御ステートメントの説明を参照してください。

あらゆるソース (つまり、DFSPARM、SYSIN、SORTCNTL、およびパラメーター・リスト) 内の上記のような制御ステートメントで、シンボルを使用できます。

DFSORT は、こうした制御ステートメントを変換するときに、ラベルと注釈を取り除き、72 桁目にアスタリスクを置いて、次の行を 16 桁目から開始することによりステートメントを継続させます。DFSORT は、元の制御ステートメントをソースの指定 (ラベル、注釈、注釈ステートメント、およびブランク・ステートメント付き) に従って、変換後のステートメントとともにリストします。

シンボルの使用方法の詳細な説明と例を、適用可能な各 DFSORT 制御ステートメントについて、以下に示します。この例は、シンボルのさまざまな使用方法および変換方法を説明するためのものです。したがって、この例のシンボルの使い方は、実際のアプリケーションにおけるシンボルの使い方とは異なる場合があります。

この例では、次の SYMNAMES ステートメントを使用します。

```
C_Field1,6,5,CH
Any_Format,12,3
Z_Field1,22,8,ZD
P_Field1,30,4,PD
C_Field2,4,2,ch
SubString,16,3,SS
LIMIT,+12500
Depts,'J82,L92,M72'
Code_1,c'86A4Z'
QCON,C'Carrie''s Constant'
Stopper,X'FFFFFF'
Flags,35,1,BI
      Error,B'11010000'
      Empty,B'.....01'
      Full,X'FF'
```



```

Lookup,52,1,BI
  Entry1,X'05'
  Value1,'Read'
  Entry2,X'20'
  Value2,'Update'
RDW,1,4          Record Descriptor Word
Variable_Fields,451 Variable fields at end of variable-length record
* Constants for report
Div_Title,'Division: '
  BO_Title,'Branch Office'
BO_Hyphens,'-----'
  BO_Equals,'===== '
  PL_Title,'          Profit/(Loss) '
PL_Hyphens,'-----'
  PL_Equals,'===== '
Total,'Total'
Lowest,'Lowest'
* Fields for report
Division,3,10,CH
Branch_Office,16,13,CH
Profit_or_Loss,31,10,ZD

```

## SORT と MERGE

**FIELDS** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。FORMAT=f または symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

### 例 1

```
SORT FIELDS=(C_Field1,A,Z_Field1,D,
             C_Field2,ZD,A),EQUALS
```

次のように変換されます。

```
SORT FIELDS=(6,5,CH,A,22,8,ZD,D,4,2,ZD,A),EQUALS
```

### 例 2

```
MERGE FIELDS=(Any_Format,A,C_Field1,A),FORMAT=CH
```

次のように変換されます。

```
MERGE FIELDS=(12,3,A,6,5,A),FORMAT=CH
```

## SUM

**FIELDS** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。FORMAT=f または symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

### 例 1

```
SUM FIELDS=(Z_Field1,C_Field1,ZD)
```

次のように変換されます。

```
SUM FIELDS=(22,8,ZD,6,5,ZD)
```

### 例 2

```
SUM FORMAT=ZD,FIELDS=(C_Field1,Any_Format)
```

次のように変換されます。

## フィールドと定数のシンボルの使用

```
SUM FORMAT=ZD,FIELDS=(6,5,12,3)
```

## INCLUDE と OMIT

**COND** オペランド: フィールド (p1,m1,f1 および p1,m1 および p2,m2,f2 および p2,m2)、および定数 (n, +n, -n, C'xx...x', X'yy...yy', Y'yyx...x' および B'bbbbbbb...bbbbbbb') を使用できるところなら、シンボルを使用できます。FORMAT=f または symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。'string' を指定したシンボルは、常に C'string' に置き換えられます。

### 例 1

```
INCLUDE COND=((Z_Field1,GT,LIMIT,AND,Any_Format,CH,EQ,C_Field2),OR,
              (SubString,NE,Depts),OR,
              (Flags,ALL,Error,AND,Flags,NE,Empty))
```

次のように変換されます。

```
INCLUDE COND=((22,8,ZD,GT,+12500,AND,12,3,CH,EQ,4,2,CH),OR,(16,3,SS,NE*,
              ,C'J82,L92,M72'),OR,(35,1,BI,ALL,B'11010000',AND,35,1,BI*,
              ,NE,B'.....01'))
```

### 例 2

```
OMIT FORMAT=BI,COND=(C_Field1,EQ,Code_1,OR,
                    Any_Format,EQ,Stopper,OR,
                    Flags,EQ,Full)
```

次のように変換されます。

```
OMIT FORMAT=BI,COND=(6,5,EQ,C'86A4Z',OR,12,3,EQ,X'FFFFFF',OR,35,1,EQ,X*
                    'FF')
```

## INREC と OUTREC

**FIELDS** オペランド: フィールド (p,m,f および p,m and p)、および非反復定数 (C'xx...x' および X'yy...yy'、ただし、nC'xx...x' または nX'yy...yy' を除く) を使用できるところなら、シンボルを使用できます。シンボルは、編集パターン ('pattern') には使用できません。

**CHANGE** および **NOMATCH** サブオペランド内は、フィールド (q,n) および定数 (C'xx...x', X'yy...yy', および B'bbbbbbb') を使用できるところなら、シンボルを使用できます。

p,m,f または p,m,Y2x を指定したシンボルでは、symbol,TO=fo が指定された場合は、それぞれ、p,m,f または p,m,Y2x に置き換えられますが、symbol,fo が指定された場合は、fo を f と区別できないので、p,m が置き換えられます。たとえば、Sym1 が 5,4,ZD として定義されると、SYM1,TO=PD は 5,4,ZD,TO=PD に変換されます。ここで、SYM1,PD は 5,4,PD に変換されます。そのため、常に、symbol,fo ではなく、symbol,TO=fo を使用する必要があります。

```
| symbol,f, symbol,HEX, symbol,TRAN=LTOU, symbol,TRAN=UTOL、または
| symbol,TRAN=ALTSEQ が指定されていない場合、p,m,Y2x を指定したシンボル
| は、p,m,Y2x に置き換えられます。symbol,f, symbol,HEX、
| symbol,TRAN=LTOU, symbol,TRAN=UTOL、または symbol,TRAN=ALTSEQ が指定
| されていない場合、p,m,Y2x を指定したシンボルは、p,m,Y2x に置き換えられま
```

す。symbol,edit または symbol,to が指定されていない場合、p,m,f が指定されたシンボル (ただし、f は Y2x または Y2xP 以外) は、p,m に置き換えられます。

'string' を指定したシンボルは、常に C'string' に置き換えられます。

### 例 1

```
INREC FIELDS=(11:C_Field2,2X,C_Field1,F,Stopper,5C'*,
              Z_Field1,55:Depts,X,P_Field1,T0=FS,X,Z_Field1,M10)
```

次のように変換されます。

```
INREC FIELDS=(11:4,2,2X,6,5,F,X'FFFFFF',5C'*,22,8,55:C'J82,L92,M72',X*,
              ,30,4,PD,T0=FS,X,22,8,ZD,M10)
```

### 例 2

```
OUTREC FIELDS=(RDW, ** Record Descriptor Word **
                Z_Field1,2Z,
                3C'Symbol cannot be used for a repeated constant',
                Code_1,Flags,
                Variable_Fields) ** Variable part of input record
```

次のように変換されます。

```
OUTREC FIELDS=(1,4,22,8,2Z,3C'Symbol cannot be used for a repeated constant',C'86A4Z',35,1,451)
```

## OUTFIL

**INCLUDE および OMIT オペランド:** フィールド (p1,m1,f1 および p1,m1 および p2,m2,f2 および p2,m2)、および定数 (n、+n、-n、C'xx...x'、X'yy...yy'、Y'yyx...x'、および B'bbbbbb...bbbbbb') を使用できる場所なら、シンボルを使用できます。symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。'string' を指定したシンボルは、常に C'string' に置き換えられます。

**OUTREC オペランド:** フィールド (p,m,f および p,m および p)、および非反復定数 (C'xx...x' および X'yy...yy'、ただし nC'xx...x' および nX'yy...yy' は除く) を使用できる場所なら、シンボルを使用できます。シンボルは、編集パターン ('pattern') には使用できません。

**CHANGE および NOMATCH サブオペランド内は、**フィールド (q,n) および定数 (C'xx...x'、X'yy...yy'、および B'bbbbbb') を使用できる場所なら、シンボルを使用できます。

p,m,f または p,m,Y2x を指定したシンボルでは、symbol,TO=fo が指定された場合は、それぞれ、p,m,f または p,m,Y2x に置き換えられますが、symbol,fo が指定された場合は、fo を f と区別できないので、p,m が置き換えられます。たとえば、Sym1 が 5,4,ZD として定義されると、SYM1,TO=PD は 5,4,ZD,TO=PD に変換されます。ここで、SYM1,PD は 5,4,PD に変換されます。そのため、常に、symbol,fo ではなく、symbol,TO=fo を使用する必要があります。

symbol,f、symbol,HEX、symbol,TRAN=LTOU、symbol,TRAN=UTOL、または symbol,TRAN=ALTSEQ が指定されていない場合、p,m,Y2x を指定したシンボルは、p,m,Y2x に置き換えられます。symbol,f、symbol,HEX、symbol,TRAN=LTOU、symbol,TRAN=UTOL、または symbol,TRAN=ALTSEQ が指定されていない場合、

## フィールドと定数のシンボルの使用

p,m,Y2x を指定したシンボルは、p,m,Y2x に置き換えられます。symbol,edit または symbol,to が指定されていない場合、p,m,f が指定されたシンボル (ただし、f は Y2x または Y2xP 以外) は、p,m に置き換えられます。

'string' を指定したシンボルは、常に C'string' に置き換えられます。

**VLFILL オペランド:** 定数 (C'x' および X'yy') を使用できるところなら、シンボルを使用できます。'string' を指定したシンボルは、常に C'string' に置き換えられます。

**VLTRIM オペランド:** 定数 (C'x' および X'yy') を使用できるところなら、シンボルを使用できます。'string' を指定したシンボルは、常に C'string' に置き換えられます。

**HEADER1 および HEADER2 オペランド:** フィールド (p,m) および非反復定数 ('xx...x' および C'xx...x'、ただし n'xx...x' および nC'xx...x' は除く) を使用できるところなら、シンボルを使用できます。p,m,f を指定したシンボルは、常に p,m のみの指定に置き換えられます。'string' を指定したシンボルは、常に C'string' に置き換えられます。

**TRAILER1 および TRAILER2 オペランド:** フィールド (p,m) および非反復定数 ('xx...x' および C'xx...x'、ただし n'xx...x' および nC'xx...x' は除く) を使用できるところなら、シンボルを使用できます。symbol,f が指定されている場合、あるいはサブオペランド TOTAL、TOT、MIN、MAX、AVG、SUBTOTAL、SUBTOT、SUB、SUBMIN、SUBMAX、および SUBAVG の外側でシンボルが指定されている場合は、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。'string' を指定したシンボルは、常に C'string' に置き換えられます。

**SECTIONS セクション:** 上記の "HEADER1 および HEADER2 オペランド" の説明が、SECTIONS の HEADER3 サブオペランドにも適用されます。上記の "TRAILER1 および TRAILER2 オペランド" の説明が、SECTIONS の TRAILER3 サブオペランドにも適用されます。

HEADER3 および TRAILER3 サブオペランドの外側では、フィールド (p,m) を使用できるところなら、シンボルを使用できます。p,m,f を指定したシンボルは、常に p,m のみの指定に置き換えられます。

### 例 1

```
OUTFIL FNAMES=OUT1,
      INCLUDE=(Z_Field1,GT,LIMIT,AND,Any_Format,CH,EQ,C_Field2),
      OUTREC=(12:P_Field1,M0,2X,Any_Format,BI,LENGTH=3,2X,QCON,2X,
              C_Field2,HEX,2X,Z_Field1,EDIT=('I III IIT.T'),2X,
* Lookup Table
      Lookup,CHANGE=(6,Entry1,Value1,Entry2,Value2),
      NOMATCH=(Lookup))
```

次のように変換されます。

```
OUTFIL FNAMES=OUT1,INCLUDE=(22,8,ZD,GT,+12500,AND,12,3,CH,EQ,4,2,CH),0*
      UTREC=(12:30,4,PD,M0,2X,12,3,BI,LENGTH=3,2X,C'Carrie's *
      Constant',2X,4,2,HEX,2X,22,8,ZD,EDIT=('I III IIT.T'),2X,*
      52,1,CHANGE=(6,X'05',C'Read',X'20',C'Update'),NOMATCH=(5*
      2,1))
```

## 例 2

```

OUTFIL FNAMES=REPORT,
      OUTREC=(6:Branch_Office,24:Profit_or_Loss,M5,LENGTH=20,75:X),
      SECTIONS=(Division,SKIP=P,
        HEADER3=(2:Div_Title,Division,5X,'Page:',&PAGE;,2/,
          6:BO_Title,24:PL_Title,/,
          6:BO_Hyphens,24:PL_Hyphens),
        TRAILER3=(6:BO_Equals,24:PL_Equals,/,
          6:Total,24:TOTAL=(Profit_or_Loss,M5,LENGTH=20),/,
          6:Lowest,24:MIN=(Profit_or_Loss,M5,LENGTH=20)))

```

次のように変換されます。

```

OUTFIL FNAMES=REPORT,OUTREC=(6:16,13,24:31,10,ZD,M5,LENGTH=20,75:X),SE*
CTIONS=(3,10,SKIP=P,HEADER3=(2:C'Division: ',3,10,5X,'P*
age:',&PAGE;,2/,6:C'Branch Office',24:C' Profit/(Lo*
ss)',/,6:C'-----',24:C'-----'),TR*
AILER3=(6:C'=====',24:C'=====',/,*
6:C'Total',24:TOTAL=(31,10,ZD,M5,LENGTH=20),/,6:C'Lowest*
',24:MIN=(31,10,ZD,M5,LENGTH=20)))

```

---

## ICETOOL 演算子でのシンボルの使用

COUNT、DISPLAY、OCCUR、RANGE、SELECT、STATS、UNIQUE および VERIFY の各 ICETOOL 演算子内で、シンボルを使用できます。一般に、上述の ICETOOL 演算子で、定数 ('string'、n、+n、または -n) およびフィールド (p,m,f または p,m) を使用できるところには、シンボルを使用できます。構文の詳細については、441 ページの『第 6 章 ICETOOL の使用』の各演算子の説明を参照してください。

ICETOOL は、SYMNAMES データ・セットを一度読み取り、実行中のすべての演算子および DFSORT 制御ステートメントに対して、その内容を使用します。演算子で、TOOLIN データ・セットまたはパラメーター・リストからのシンボルを使用できます。DFSORT 制御ステートメントでも、xxxxCNTL データ・セットまたは DFSPARM データ・セットのシンボルを使用できます (詳細については、570 ページの『DFSORT ステートメントでのシンボルの使用』参照)。

ICETOOL は、元の演算子ステートメントと変換後の演算子ステートメントをリストします。

シンボルの使用方法の詳細を、適用可能な各 ICETOOL 制御ステートメントについて以下に示し、次に ICETOOL の完全な例を示します。例は、シンボルのさまざまな使用方法および変換方法を説明するためのものです。したがって、この例のシンボルの使い方は、実際のアプリケーションにおけるシンボルの使い方とは異なる場合があります。

### COUNT

**HIGHER、LOWER、EQUAL および NOTEQUAL オペランド:** 定数 (x、y、v、および w) を使用できるところなら、シンボルを使用できます。

## フィールドと定数のシンボルの使用

### DISPLAY

**ON** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。symbol,f または symbol,HEX が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

**BREAK** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

**TITLE、HEADER、TOTAL、MAXIMUM、MINIMUM、AVERAGE、BTITLE、BTOTAL、BMAXIMUM、BMINIMUM、および BAVERAGE** オペランド: 定数 ('string') を使用できるところなら、シンボルを使用できます。C'string' を指定したシンボルは、常に 'string' に置き換えられます。

### OCCUR

**ON** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。symbol,f または symbol,HEX が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

**TITLE** および **HEADER** オペランド: 定数 ('string') を使用できるところなら、シンボルを使用できます。C'string' を指定したシンボルは、常に 'string' に置き換えられます。

**HIGHER、LOWER** および **EQUAL** オペランド: 定数 (x, y および v) を使用できるところなら、シンボルを使用できます。

### RANGE

**ON** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

**HIGHER、LOWER、EQUAL** および **NOTEQUAL** オペランド: 定数 (x, y, v および w) を使用できるところなら、シンボルを使用できます。

### SELECT

**ON** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

**HIGHER、LOWER** および **EQUAL** オペランド: 定数 (x, y および v) を使用できるところなら、シンボルを使用できます。

### STATS、UNIQUE および VERIFY

**ON** オペランド: フィールド (p,m,f と p,m) を使用できるところなら、シンボルを使用できます。symbol,f が指定されている場合、p,m,f を指定したシンボルは、p,m のみの指定に置き換えられます。

## ICETOOL の例

```

//TOOLSVM JOB A402,PROGRAMMER
//DOIT EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//SYMNOUT DD SYSOUT=*
//SYMNAMES DD *
Rdw,1,4,BI
Account_Code,12,1,CH
Dept_Code,*,=,=
Customer_Name,*,20,CH
SKIP,2
Customer_Balance,*,10,ZD
Customer_Flags,*,1,BI
* Department Codes
Research,'R'
Marketing,'M'
* Balance Cutoffs
Cancel,+10000      100.00
Gift,+1000000      10,000.00
Stop_Check,-500    -5.00
* Headings and Titles
Title,'Customer Report for'
Head1,'Customer Name'
Head2,'Customer Balance'
Head3,'Customer Flags'
/*
//IN DD DSN=MY.CUSTOMER.INPUT,DISP=SHR
//OUT DD DSN=&0,UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE),
// DISP=(,PASS)
//LIST1 DD SYSOUT=*
//TOOLIN DD *
    RANGE FROM(IN) ON(Customer_Balance) LOWER(Stop_Check)
    SORT FROM(IN) TO(OUT) USING(CTL1)
    DISPLAY FROM(OUT) LIST(LIST1) BLANK WIDTH(133) -
        TITLE(Title) DATE(4MD/) PAGE -
        HEADER(Head1) ON(Customer_Name) -
        HEADER(Head2) ON(Customer_Balance,C1) -
        HEADER(Head3) ON(Customer_Flags,HEX)
/*
//CTL1CNTL DD *
    SORT FIELDS=(Customer_Balance,D,Customer_Name,A)
    INCLUDE COND=((Dept_Code,EQ,Research,OR,Dept_Code,EQ,Marketing),
        AND,Customer_Balance,GT,Gift)
/*

```

SYMNOUT には、次の内容が示されます。

```

----- ORIGINAL STATEMENTS FROM SYMNAMES -----
Rdw,1,4,BI
Account_Code,12,1,CH
Dept_Code,*,=,=
Customer_Name,*,20,CH
SKIP,2
Customer_Balance,*,10,ZD
Customer_Flags,*,1,BI
* Department Codes
Research,'R'
Marketing,'M'
* Balance Cutoffs
Cancel,+10000      100.00
Gift,+1000000      10,000.00
Stop_Check,-500    -5.00
* Headings and Titles
Title,'Customer Report for'
Head1,'Customer Name'

```

## フィールドと定数のシンボルの使用

```
Head2,'Customer Balance'  
Head3,'Customer Flags'
```

```
----- SYMBOL TABLE -----  
Rdw,1,4,BI  
Account_Code,12,1,CH  
Dept_Code,13,1,CH  
Customer_Name,14,20,CH  
Customer_Balance,36,10,ZD  
Customer_Flags,46,1,BI  
Research,C'R'  
Marketing,C'M'  
Cancel,+10000  
Gift,+1000000  
Stop_Check,-500  
Title,C'Customer Report for'  
Head1,C'Customer Name'  
Head2,C'Customer Balance'  
Head3,C'Customer Flags'
```

ICETOOL 演算子は、次のように変換されます。

```
RANGE FROM(IN) ON(36,10,ZD) LOWER(-500)
```

```
    SORT FROM(IN) TO(OUT) USING(CTL1)
```

```
DISPLAY FROM(OUT) LIST(LIST1) BLANK WIDTH(133)-  
TITLE('Customer Report for') DATE(4MD/) PAGE-  
HEADER('Customer Name') ON(14,20,CH)-  
HEADER('Customer Balance') ON(36,10,ZD,C1)-  
HEADER('Customer Flags') ON(46,1,HEX)
```

CTL1CNTL 内の DFSORT 制御ステートメントは、次のように変換されます。

```
SORT FIELDS=(36,10,ZD,D,14,20,CH,A)  
INCLUDE COND=((13,1,CH,EQ,C'R',OR,13,1,CH,EQ,C'M'),AND,36,10,ZD,GT,+10*  
000000)
```

---

## シンボルに関する注意事項

- EFS プログラムは、シンボル処理を行うときは使用できません。
- DFSORT または ICETOOL は、SYMNAMES ステートメントにエラーがないかスキャンし、検出した最初のエラーに対するエラー・メッセージを印刷します。マーカー (\$) が、SYMNAMES ステートメントのエラー個所のすぐ下に印刷されます (該当する場合)。  
最初のエラーを検出するとスキャンを停止し、次の SYMNAMES ステートメントのスキャンに移ります。ただし、一度エラーを検出すると、以降のステートメントで、p にアスタリスク (\*) を指定するか、または POSITION,symbol を使用することにより生成される位置については、エラーをチェックしません。いずれかのステートメントでエラーが検出された場合、DFSORT および ICETOOL は、すべての SYMNAMES ステートメントをスキャンした後、終了します。
- 置換フェーズの間 (つまり、シンボルを値に置き換える間) に制御ステートメントまたは演算子ステートメントにエラーを検出すると、DFSORT または ICETOOL は、次のいずれかの対応を取ります。
  - エラーとなった元のステートメントに続いて、\$ マーカー (該当する場合) とエラー・メッセージを印刷し、次のステートメントの置換処理を続行して、置換フェーズが完了したら終了します。または、



- エラーとなったステートメントの置換処理を停止して、次のステートメントに進み、エラーの対応は、置換フェーズの後の処理に任せます。この場合は、置換された値ではなく、シンボルを変換後のステートメントに出力できます。
- 置換フェーズが正常に完了すると、DFSORT および ICETOOL は、シンボルが使用できる場所なら、シンボルを値に置換します。特定のステートメントまたはオペランドで、置換された値が無効な場合は、置換フェーズの後に検出されず。こうすることにより、エラーの原因が判別しやすくなります。たとえば、SYMNAMES の内容が次のとおりの場合、

```
Sym1,5,4,ZD
Con1,'1234'
Con2,1234
```

次のステートメントは、

```
INCLUDE COND=(Sym1,EQ,Con1)
```

置換フェーズで次のように変換されます。

```
INCLUDE COND=(5,4,ZD,EQ,C'1234')
```

このステートメントについて、C'1234' の下に \$ マーカーが付いた ICE114A メッセージが発行されます。ZD フィールドは、文字定数と比較できないためです。この例では、ステートメントの Con1 を Con2 (10 進定数) に変えるか、または Con1 を 10 進定数として再定義することにより、エラーを修正できます。

- 一時または永久メッセージ・データ・セットを使用するときは、MOD の後処置を指定して、メッセージ・データ・セット内のすべてのメッセージおよび制御ステートメントを確実に見ることができるようにするのが、最もよい方法です。特に、シンボル処理を使用するときに、MOD を使用しないと、ブロック・セットが選択されていない限り、元の制御ステートメントを見ることはできません。
- レコードを (たとえば、E15、E35、INREC、OUTREC または OUTFIL を使用して) 再配置し、再配置したレコードに対してシンボルを使用するときは、必ずフィールドの新しい位置にマップするシンボルを使用してください。たとえば、次のステートメントを含む、SYMNAMES データ・セットを使用する場合を考えます。

```
Field1,1,5,ZD
Field2,*,6,ZD
Field3,*,3,ZD
Field4,*,4,ZD
```

次の INREC ステートメントを指定すると、

```
INREC FIELDS=(Field2,Field4)
```

結果として得られるレコードは、Field2 と Field4 だけを含みます。再配置されたレコード (たとえば、SORT ステートメントのレコード) に対してシンボルを使用するときは、次に示すように、再配置されたレコードにマップするシンボルをもつ、SYMNAMES データ・セットを使用することが必要になります。

```
New_Field2,1,6,ZD
New_Field4,*,4,ZD
```

## フィールドと定数のシンボルの使用

上記の例のように、再配置されたフィールドに対して一意的なシンボルを使用する場合、古いシンボル・データ・セットと新しいシンボル・データ・セットを連結して、次の例のように、古いシンボルと新しいシンボルを適切なところに使用できます。

```
INREC FIELDS=(Field2,Field4)
SORT FIELDS=(New_Field2,A,New_Field4,A)
```

## 第 8 章 拡張機能サポートの使用

EFS の使用 . . . . .	582
EFS プログラムのアドレッシング・モードと常駐モード . . . . .	583
EFS の働き . . . . .	583
DFSORT プログラム・フェーズ . . . . .	584
DFSORT による EFS プログラムの呼び出し . . . . .	584
初期設定フェーズ . . . . .	586
入力フェーズ . . . . .	588
終了フェーズ . . . . .	589
EFS を使用して行えること . . . . .	589
データ・セットのオープンおよび初期設定 . . . . .	590
制御ステートメントの検査、変更、または無視 . . . . .	590
EFS プログラム・ユーザー出口ルーチンによるユーザー定義データ・タイプ の処理 . . . . .	593
メッセージ・データ・セットに印刷するためのメッセージの提供 . . . . .	593
DFSORT の終了 . . . . .	593
データ・セットのクローズおよびハウスキーピング . . . . .	593
EFS インターフェース・パラメーター・リストの構造 . . . . .	593
処置コード . . . . .	595
制御ステートメント要求リスト . . . . .	596
EFS プログラムへ送られる制御ステートメント・ストリング . . . . .	596
OPTION および DEBUG 制御ステートメントの特殊処理 . . . . .	598
EFS プログラムにより戻される制御ステートメントのストリング . . . . .	598
解析に関する規則 . . . . .	599
SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形 式 . . . . .	600
FIELDS オペランドの D1 形式 . . . . .	600
COND オペランドの D2 形式 . . . . .	601
元の制御ステートメントの長さ . . . . .	602
変更後の制御ステートメントの長さ . . . . .	603
EFS プログラム・コンテキスト域 . . . . .	603
抽出バッファー・オフセット・リスト . . . . .	603
レコード長リスト . . . . .	603
情報フラグ . . . . .	604
メッセージ・リスト . . . . .	605
EFS プログラム出口ルーチン . . . . .	606
EFS01 および EFS02 機能の説明 . . . . .	606
EFS01 ユーザー出口ルーチン . . . . .	607
EFS01 パラメーター・リスト . . . . .	608
EFS02 ユーザー出口ルーチン . . . . .	608
EFS02 パラメーター・リスト . . . . .	611
EFS プログラムのユーザー出口ルーチンのアドレッシング・モードおよび常 駐モード . . . . .	611
ユーザーが指定しなければならない EFS プログラム戻りコード . . . . .	612
レコード処理の順序 . . . . .	612
SNAP ダンプの要求の方法 . . . . .	616
EFS プログラムの例 . . . . .	616
DFSORT 初期設定フェーズ . . . . .	616
メジャー・コール 1. . . . .	616

## 拡張機能サポートの使用

メジャー・コール 2. . . . .	617
メジャー・コール 3. . . . .	618
DFSORT 終了フェーズ. . . . .	619
メジャー・コール 4. . . . .	619
メジャー・コール 5. . . . .	619

---

## EFS の使用

『第 4 章 ユーザー独自のユーザー出口ルーチンの使用』のユーザー出口と同様、DFSORT 拡張機能サポート (EFS) インターフェースは、ユーザー自身が作成する EFS プログラムへ実行時の制御を渡すための 1 つの手段です。EFS プログラムは、DFSORT で 2 バイト文字セット (日本語文字など) を処理する場合は欠かせないものです。

DFSORT で日本語のデータ・タイプを処理する場合は、IBM 2 バイト文字セット配列支援プログラム (DBCS 配列)、ライセンス・プログラム 5665-360、リリース 2.0 を使用するか、適切なロケールのロケール処理を使用します。

EFS プログラムおよび EFS プログラム出口ルーチンを使用して、次の機能を行うことができます。

- ユーザー定義のデータ・タイプ (2 バイト文字セットなど) を、ユーザー定義の照合順序で分類または組み合わせる。
- ユーザー定義のデータ・タイプに基づいて、レコードの組み込みまたは除外を行う。
- メッセージ・データ・セットに印刷するために、ユーザーが作成したメッセージを DFSORT に提供する。
- DFSORT による処理の前に、制御ステートメントまたは EXEC PARM オプションを検査、変更、または無視する。

また、EFS プログラムは、データ・セットのオープンと初期設定、DFSORT の終了、データ・セットのクローズ、などの通常のタスクを行うこともできます。

標準レジスターおよびリンケージ規則を使用し、かつ以下の機能を備えている言語であれば、どのような言語でも、ユーザーの EFS プログラムを作成できます。

- パラメーター・リストとレコードをレジスター 1 に入れて渡す (EFS プログラムでユーザーが EFS01 および EFS02 出口ルーチンを提供する場合)
- 戻りコードを汎用レジスター 15 に入れて渡す

### 注:

1. 従来の組み合わせやテープ作業データ・セットの分類アプリケーションに対しては、DFSORT は EFS プログラムをサポートしません。
2. EFS 処理が有効で、かつ EFS プログラムが EFS01 または EFS02 出口ルーチンを提供している場合は、VLSHRT は使用できません。
3. SORT、MERGE、INCLUDE、または OMIT フィールドのロケール処理を使用する場合は、EFS プログラムは使用しないでください。DFSORT のロケール処理を使用すれば、EFS プログラムを使用する必要がない場合があります。ロケール処理の詳細については、181 ページの『OPTION 制御ステートメント』を参照してください。
4. シンボル処理を使用する場合は、EFS プログラムは使用しないでください。

DFSORT 目標ライブラリー SICEUSER には、ICEDEFS と呼ばれるマッピング・マクロが入っています。このマクロは、EFS パラメーター・リスト用に別個のアセンブラー DSECT を備えています。

---

## EFS プログラムのアドレッシング・モードと常駐モード

16 メガバイト境界より上または下に常駐させて実行するように EFS プログラムを設計できます。常駐モードおよびアドレッシング・モードは、24 ビット、31 ビット、および ANY の任意の有効な組み合わせが可能です。EFS プログラムが、16 メガバイト境界より下に常駐して実行するように設計されている場合、その EFS プログラムは正しい戻りモードを決定する必要があります。

---

## EFS の働き

EFS インターフェースは、DFSORT とプログラムの EFS プログラムの間の連絡に使用される可変長パラメーター・リストで構成されます。DFSORT は、ユーザーが実行時の特定の時点で作成した EFS プログラムを活動化し、EFS プログラムの実行中に、インターフェースを通じて情報のやり取りを行います。

ユーザーは、下記のような EFS=name オプション (name はユーザーの EFS プログラムの名前) を指定して、実行時に EFS プログラムを活動化できます。

- DFSORT インストール時に ICEMAC マクロを用いてセットする (18 ページの『インストール・デフォルト』を参照)
- ジョブ制御言語を用いて DFSORT を呼び出す場合、EXEC ステートメントの PARM パラメーターで指定する (34 ページの『EXEC/DFSPARM PARM オプションの指定』を参照)
- OPTION プログラム制御ステートメントで指定する (181 ページの『OPTION 制御ステートメント』を参照)

指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。584 ページの図 42 は、ユーザーが作成した EFS プログラムに DFSORT の処理機能をリンクする際に、EFS インターフェースが果たす役割を示しています。

## EFS の働き

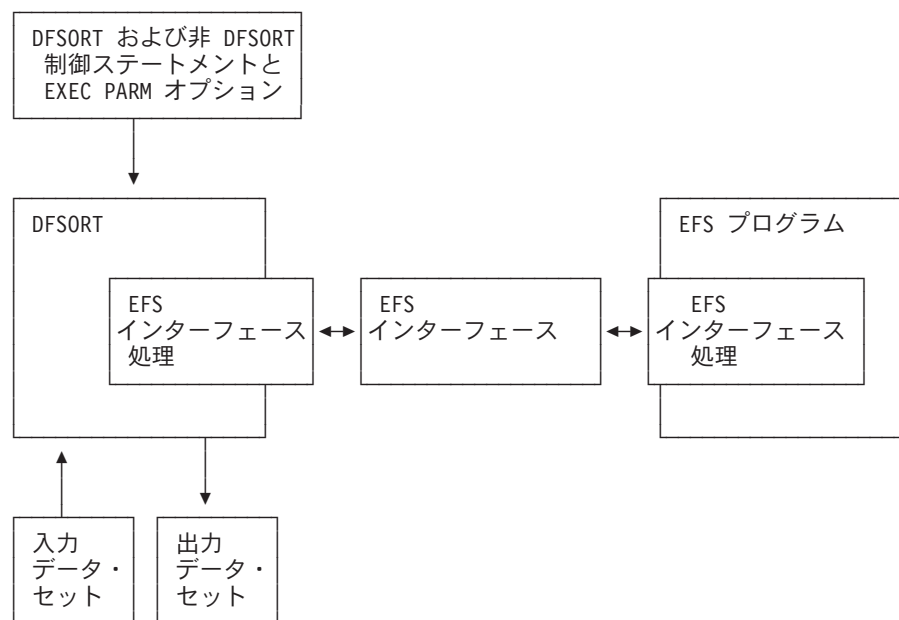


図42. DFSORT と EFS プログラムの関係

## DFSORT プログラム・フェーズ

DFSORT プログラム・フェーズとは、出力ファイルの作成などのような特定のタスクを実行するために設計された、大きな DFSORT 構成要素です。EFS プログラムは、DFSORT プログラム・フェーズ (EFS プログラムで可能な各種のタスクを実行する) の実行時のさまざまな時点で呼び出されます。終了フェーズが終わると、DFSORT はオペレーティング・システムまたは呼び出しプログラムに制御を戻します。

EFS 処理は、DFSORT の初期設定フェーズ、入力フェーズ、および終了フェーズのときに呼び出すことができます。DFSORT は、常に、初期設定フェーズ時に EFS プログラムを呼び出します。

入力フェーズでは、DFSORT は入力レコードを読み取り、そのレコードのすべての INCLUDE または OMIT ステートメント論理を実行します。EFS プログラムが出口ルーチン (EFS01 および EFS02) を生成すると、DFSORT は入力フェーズ時にこれらの出口ルーチンを呼び出します。

終了フェーズ時に、DFSORT はデータ・セットをクローズし、記憶域を解放し、呼び出しプログラムまたはシステムに制御を戻します。DFSORT は、常に、終了フェーズから EFS プログラムを呼び出します。

## DFSORT による EFS プログラムの呼び出し

DFSORT は、さまざまなフェーズで 5 つの呼び出し機能 (メジャー・コール 1 から 5) を行い、EFS インターフェースを介して、DFSORT とユーザーの EFS プログラムとの間の情報の受け渡しを行います。メジャー・コール 2 と 3 では、DFSORT は複数の呼び出しを行うことができます。プログラム・フェーズと実行時

の呼び出しの関係を図示した 図 43 と 586 ページの図 44 を参照の上、この項を読んでください。

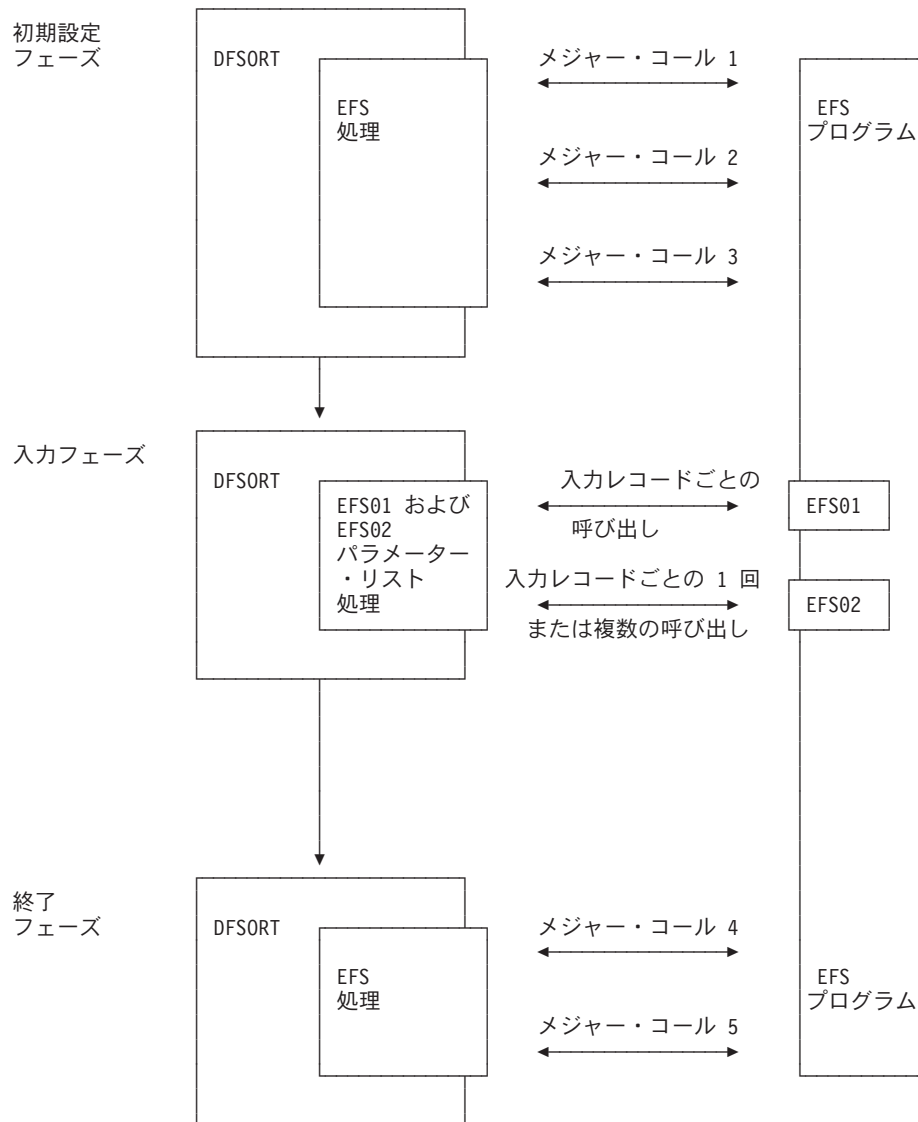


図 43. 分類処理のための EFS プログラム呼び出し：この図は、EFS プログラム EFS01 および EFS02 出力ルーチンの呼び出しも示しています。

## EFS の働き

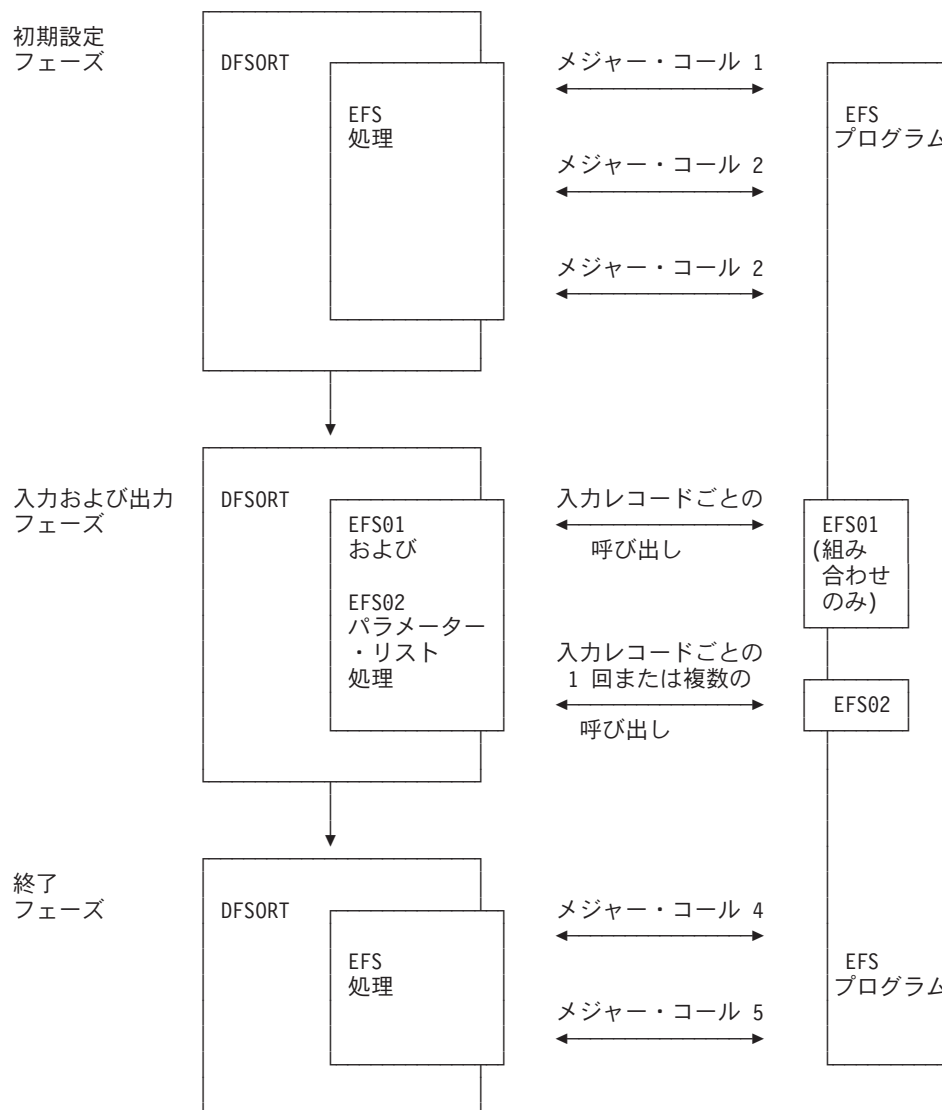


図 44. 組み合わせまたはコピー処理のための EFS プログラム呼び出し：この図は、EFS プログラム EFS01 および EFS02 出口ルーチンの呼び出しも示しています。

### 初期設定フェーズ

DFSORT は、初期設定フェーズでは、メジャー・コール 1 から 3 を実行します。

**メジャー・コール 1:** EFS プログラムは、データ・セットのオープンや記憶域の入手などの初期設定処理を行うことができます。

情報は、EFS インターフェースを介して、DFSORT と EFS プログラムの間で互いにやり取りされます。

メジャー・コール 1 では、DFSORT は EFS インターフェースを介して、次のようなフィールドを EFS プログラムに提供します。

- メジャー・コール 1 が有効であることを示す処置コード
- 現在の処理を記述する情報フラグ



制御を DFSORT に戻すときに、EFS プログラムは EFS インターフェースを介して、次のようなフィールドを提供できます。

- 制御ステートメント要求リスト (DFSORT および非 DFSORT 制御ステートメントの命令定義子のリストまたは EXEC PARM オプションを含む)

**注:** EFS プログラムで OUTFIL ステートメントを要求できません。

- EFS プログラム・コンテキスト域 (EFS プログラム用の専用連絡域)
- メッセージ・データ・セットに印刷するためのメッセージを含むリスト
- 汎用レジスター 15 内の戻りコード

**メジャー・コール 2:** この呼び出しでは、EFS プログラムは、DFSORT が制御ステートメントを処理する前に、制御ステートメントを検査、変更、または無視したり、ユーザー作成のメッセージをメッセージ・データ・セットに提供できます。DFSORT は、制御ステートメントまたは EXEC PARM が要求されるたびに、EFS プログラムを 1 回呼び出します。

メジャー・コール 2 では、DFSORT は EFS インターフェースを介して、次のようなフィールドを EFS プログラムに提供します。

- メジャー・コール 2 が有効であることを示す処置コード
- EFS プログラムが要求した元の制御ステートメントまたは EXEC PARM オプション
- 元の制御ステートメントまたは EXEC PARM オプションの長さ
- 現在の処理を記述する情報フラグ
- EFS プログラム・コンテキスト域 (EFS プログラム用の専用連絡域)

制御を DFSORT に戻すときに、EFS プログラムは EFS インターフェースを介して、次のようなフィールドを提供できます。

- DFSORT が EFS プログラムに送った制御ステートメントまたは EXEC PARM オプションの変更バージョン。ユーザー定義のデータ・タイプの分類または組み合わせを行いたい場合、あるいはユーザー定義のデータ・タイプの組み込みまたは省略を行いたい場合は、EFS プログラムで、SORT/MERGE あるいは INCLUDE/OMIT 制御ステートメントの新しい形式に戻す必要があります。これらの新しい形式 (D1 および D2) は、ユーザーが EFS プログラムに組み込んだ EFS01 および EFS02 出口ルーチン呼び出すように、DFSORT に通知を出します。

**注:** OUTFIL ステートメントを EFS プログラムへ渡すこともできないし、解析するために EFS プログラムから戻すこともできません。

- 変更済みの制御ステートメントまたは EXEC PARM オプションの長さ
- 制御ステートメントまたは EXEC PARM オプションを解析するか無視するかを DFSORT に通知する情報フラグ
- DFSORT がメッセージ・データ・セットに印刷するメッセージのリスト
- 汎用レジスター 15 内の戻りコード

**メジャー・コール 3:** メジャー・コール 3 では、EFS プログラムは、メッセージ・データ・セットに印刷するためのユーザー作成メッセージを DFSORT に提供できます。DFSORT は、ブロック・セット手法の場合は 1 回、ピアレッジ / ベー

## EFS の働き

ル手法の場合も 1 回、EFS プログラムを呼び出すことができます。DFSORT はこの呼び出しで、EFS01 および EFS02 出口ルーチンを処理するために必要な情報をさらに EFS プログラムから入手します。

メジャー・コール 3 では、DFSORT は EFS インターフェースを介して、次のようなフィールドを EFS プログラムに提供します。

- メジャー・コール 3 が有効であることを示す処置コード
- EFS01 出口ルーチンに必要な抽出バッファ・オフセット・リスト
- 入力レコードと出力レコードのレコード長リスト
- 現在の処理を記述する情報フラグ
- EFS プログラム・コンテキスト域 (EFS プログラム用の専用連絡域)

制御を DFSORT に戻すときに、EFS プログラムは EFS インターフェースを介して、次のようなフィールドを提供できます。

- EFS01 出口ルーチンのアドレス
- EFS02 出口ルーチンのアドレス
- メッセージ・データ・セットに印刷するためのメッセージのリスト
- 汎用レジスター 15 内の戻りコード

### 入力フェーズ

DFSORT は、入力フェーズ中に 2 つの出口ルーチン EFS01 と EFS02 を実行します。EFS01 ルーチンは、ユーザー定義の照合順序を用いてユーザー定義の分類または組み合わせをサポートし、各レコードごとに 1 回呼び出されます。EFS02 ルーチンは、ユーザー定義のデータ・タイプにレコードを組み込むまたはデータ・タイプからレコードを除去する論理を提供し、その論理に従って、各レコードごとに 1 回または複数回呼び出されます。

情報は、EFS01 および EFS02 パラメーター・リストを介して、DFSORT と出口ルーチンの間で互いにやり取りされます。

DFSORT は、パラメーター・リストを介して、次のようなフィールドを EFS01 に提供します。

- 抽出バッファ域。EFS01 は、すべての EFS 制御フィールドをこのバッファ域に移す必要があります。詳細については、607 ページの『EFS01 ユーザー出口ルーチン』を参照してください。
- 入力データ・レコード
- EFS プログラム・コンテキスト域 (EFS プログラム用の専用連絡域)

EFS01 ルーチンは、制御を DFSORT に戻すときに、戻りコードを汎用レジスター 15 に戻す必要があります。

DFSORT は、パラメーター・リストを介して、次のようなフィールドを EFS02 に提供します。

- 関連付け識別子。これは EFS フィールドを含む関係条件を識別します。詳細については、608 ページの『EFS02 ユーザー出口ルーチン』を参照してください。
- 入力データ・レコード

制御を DFSORT に戻すときに、EFS02 ルーチンは戻りコードを汎用レジスター 15 に戻す必要があります。

### 終了フェーズ

DFSORT は、終了フェーズでは、メジャー・コール 4 から 5 を実行します。各メジャー・コールごとに 1 回だけ呼び出しが行われます。

**注:** DFSORT の ESTAE リカバリー・ルーチンが有効になっている間にシステム異常終了が起きた場合に、メジャー・コール 4 と 5 がまだ実行されていないと、ESTAE ルーチンがそれらを実行します。メジャー・コール 1 の間に EFS の異常終了が起きた場合は、ESTAE ルーチンはメジャー・コール 4 と 5 の実行は行いません。ESTAE についての詳細は、747 ページの『付録 E. DFSORT の異常終了処理』を参照してください。

**メジャー・コール 4:** EFS プログラムは、メッセージ・データ・セットに印刷するためのすべての最終ユーザー作成メッセージを提供します。

メジャー・コール 4 では、DFSORT は EFS インターフェースを介して、次のようなフィールドを EFS プログラムに提供します。

- メジャー・コール 4 が有効であることを示す処置コード
- EFS プログラム・コンテキスト域 (EFS プログラム用の専用連絡域)

制御を DFSORT に戻すときに、EFS プログラムは EFS インターフェースを介して、次のようなフィールドを提供できます。

- メッセージ・データ・セットに印刷するためのメッセージを含むメッセージ・リスト
- 汎用レジスター 15 内の戻りコード

**メジャー・コール 5:** EFS プログラムは、すべての終了処理 (データ・セットのクローズや記憶域の解放など) を行います。

メジャー・コール 5 では、DFSORT は EFS インターフェースを介して、次のようなフィールドを EFS プログラムに提供します。

- メジャー・コール 5 が有効であることを示す処置コード
- EFS プログラム・コンテキスト域 (EFS プログラム用の専用連絡域)

EFS プログラムは、制御を DFSORT へ戻す際には、汎用レジスター 15 に戻りコードを入れます。

---

## EFS を使用して行えること

DFSORT の初期設定フェーズ、入力フェーズ、および終了フェーズで 7 つの基本タスクを実行するように、EFS プログラムを設計できます。それらの中のいくつかのタスクは、EFS プログラムが生成するユーザー出口ルーチン EFS01 および EFS02 を使用することが必要です。

## EFS を使用して行えること

表 69. 拡張機能サポート (EFS) プログラムの機能

EFS プログラム機能	初期設定フェーズ	入力フェーズ	終了フェーズ
オープンおよび初期設定	EFS プログラム		
DFSORT が処理する前の、DFSORT および非 DFSORT 制御ステートメントの検査、変更、または無視	EFS プログラム		
ユーザー定義の照合順序によるユーザー定義データ・タイプの分類または組み合わせ		EFS01	
ユーザー定義のデータ・タイプに基づくレコードの組み込みまたは除外を行うための論理の提供		EFS02	
メッセージ・データ・セットに印刷するためのメッセージの DFSORT への提供	EFS プログラム		EFS プログラム
DFSORT の終了	EFS プログラム	EFS01、EFS02	EFS プログラム
データ・セットのクローズおよびハウスキューピング			EFS プログラム

### データ・セットのオープンおよび初期設定

ユーザーの EFS プログラムは、データ・セットをオープンし、必要な記憶域を入手し、実行時に必要なその他の初期設定を行うことができます。

### 制御ステートメントの検査、変更、または無視

メジャー・コール 1 では、EFS プログラムは、制御ステートメント要求リストを送って、メジャー・コール 2 で DFSORT に EFS プログラムを送らせたい制御ステートメントまたは EXEC PARM オプションを示すことができます。EFS プログラムで OUTFIL ステートメントを要求できません。

メジャー・コール 2 では、EFS プログラムは、DFSORT が、EXEC ステートメント、SYSIN、SORTCNTL、DFSPARM、または呼び出しプログラムから渡されたパラメーター・リストから読み取った制御ステートメントと EXEC PARM オプションを検査、変更、または無視できます。OUTFIL ステートメントを EFS プログラムへ渡すこともできないし、解析するために EFS プログラムから戻すこともできません。

EFS プログラムが活動状態のときに使用される制御ステートメントの処理順序を図示している 592 ページの図 45 を参照してください。

## EFS を使用して行えること

元の制御ステートメントと制御パラメーターに適用される指定変更規則と同じ指定変更規則が、EFS プログラムから戻された制御ステートメントと制御パラメーターに適用されます。

たとえば、EFS プログラムにより SORT ステートメントに追加された STOPAFT パラメーターは、SORT ステートメントが最初から STOPAFT パラメーターを含んでいるのと同様に、OPTION ステートメントの STOPAFT パラメーターにより同じように指定変更されます。

指定変更の詳細については、697 ページの『付録 B. DFSORT オプションの指定 / 指定変更』を参照してください。

EFS を使用して行えること

DFSORT の直接呼び出し

DFSORT の製品呼び出し

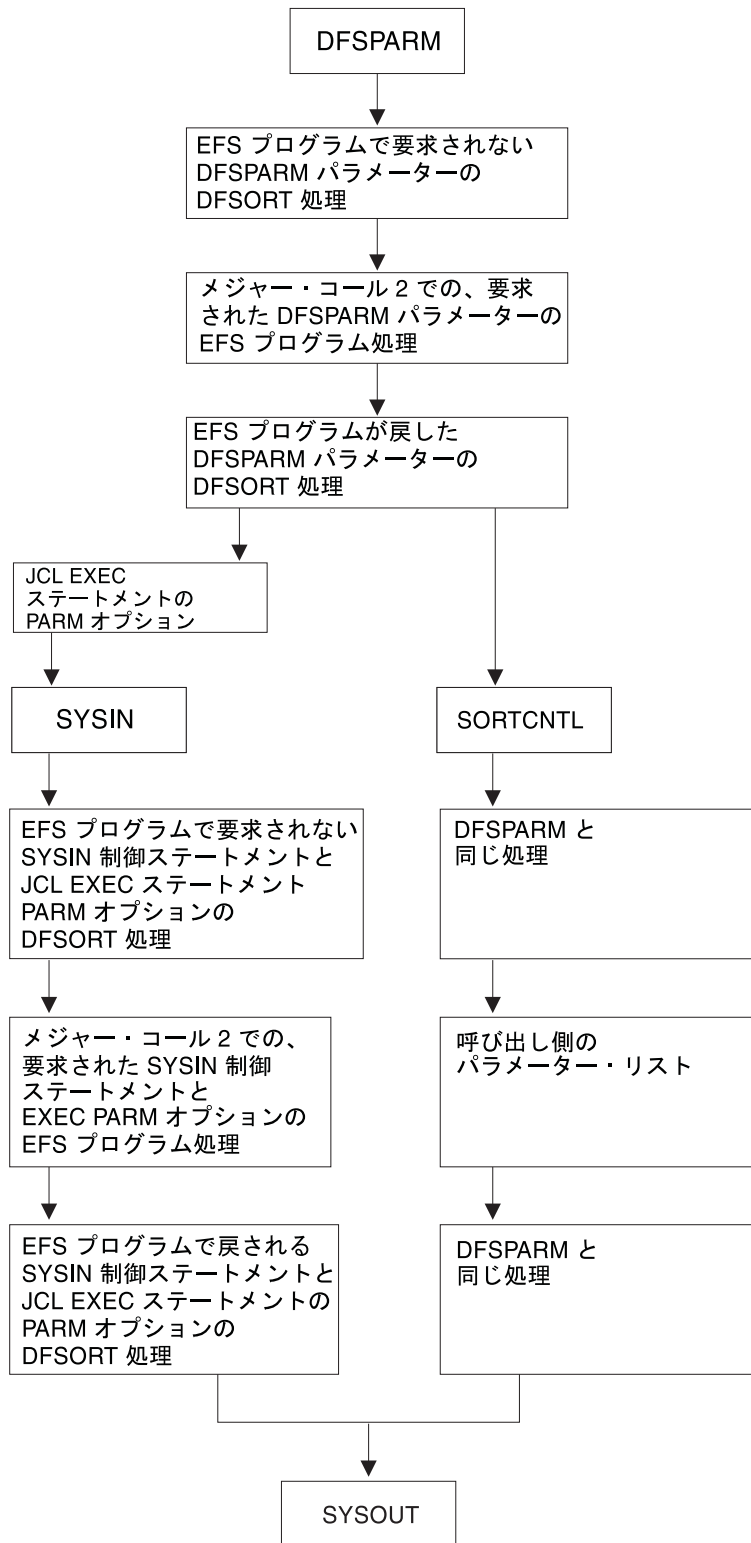


図 45. 制御ステートメント処理の順序

## EFS プログラム・ユーザー出口ルーチンによるユーザー定義データ・タイプの処理

実行中に各種のタスクを行うために、2 つのユーザー出口ルーチンを提供する EFS プログラムを作成できます。

EFS プログラムのユーザー出口ルーチンは、次の機能を行うことができます。

- ユーザー定義データ・タイプの処理。EFS プログラムは、入力レコードの任意の制御フィールドを変更するための EFS01 ルーチンを提供できます。
- ユーザー定義データ・タイプに基づく、レコードの組み込みまたは除外。EFS プログラムは、入力レコードを処理のために組み込むかどうかを判別するために、そのレコードのすべての入力フィールドを調べる出口ルーチンを提供できます。

## メッセージ・データ・セットに印刷するためのメッセージの提供

EFS プログラムを使用して、以下のいくつかの目的のためにメッセージを調整できます。

- 新しいタイプの操作を記述する
- 拡張フィールド・パラメーターを記述する
- メッセージ・データ・セットをユーザー・システム用にカスタマイズする
- 制御ステートメントまたは EXEC PARM オプションに関する統計情報を表示する

次の機能を使用して、EFS プログラムが戻した制御ステートメントをメッセージ・データ・セットに印刷するかどうかを制御できます。

- ICEMAC マクロの LISTX 演算子 (18 ページの『インストール・デフォルト』を参照)
- JCL EXEC ステートメントの PARM フィールド内の LISTX または NOLISTX 演算子 (34 ページの『EXEC/DFSPARM PARM オプションの指定』を参照)
- OPTION プログラム制御ステートメントの LIST または NOLIST 演算子

## DFSORT の終了

EFS プログラムは 5 つのメジャー・コールのいずれにおいても、DFSORT を終了できます。また、入力フェーズ中の 2 つの EFS プログラム出口のいずれからでも終了できます。

## データ・セットのクローズおよびハウスキューピング

メジャー・コール 5 で、EFS プログラムは、データ・セットをクローズし、記憶域を解放し、他の必要な任意のハウスキューピングを行うことができます。

---

## EFS インターフェース・パラメーター・リストの構造

EFS インターフェースは可変長のパラメーター・リストから構成され、DFSORT と EFS プログラムの間の連絡用に使用されます。DFSORT は、初期設定フェーズでパラメーター・リストをゼロに初期設定しますが、リストの終了標識だけは X'FFFFFFFF' にセットされます。

## EFS インターフェース・パラメーター・リストの構造

パラメーター・リストは 16 メガバイト境界より下に常駐し、EFS プログラムが活動状態になっている間はアクセス可能になっています。ただし、DFSORT が記憶域の使用を最適化するために、実行中にその記憶域を変更する場合があります。したがって、レジスター 1 (インターフェース・パラメーター・リストを渡すために使用) の実アドレスは、DFSORT の実行中に変更されることがあります。

図 46 は、EFS インターフェース・パラメーター・リストの構造を図示したものです。リストに示されている部分は、これから順次説明します。EXEC PARM は図には示されていませんが、処理には含まれています。

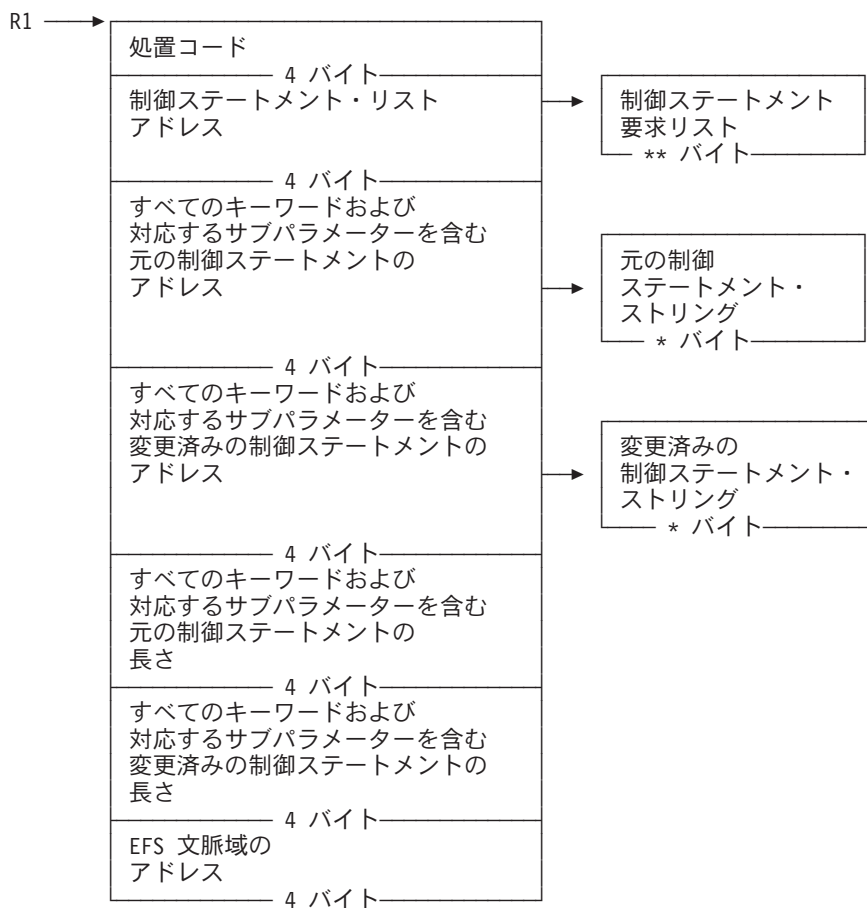
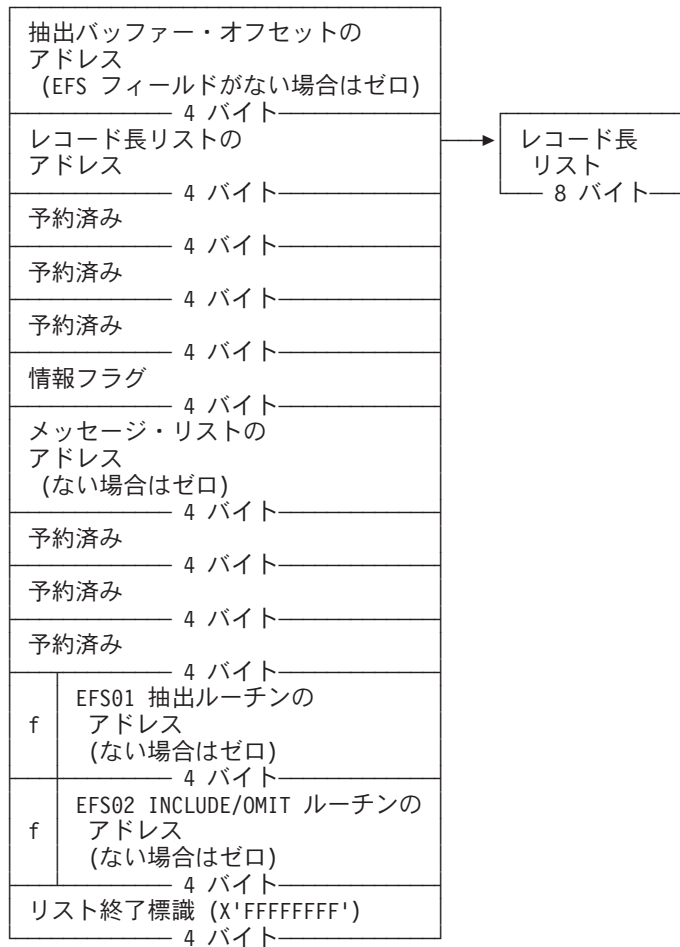


図 46. EFS インターフェース・パラメーター・リスト (1/2)



## EFS インターフェース・パラメーター・リストの構造



- \*\* - リストの中の長さフィールドによって決定された長さ
- \* - 対応する長さフィールドによって決定された長さ

図 46. EFS インターフェース・パラメーター・リスト (2/2)

## 処置コード

DFSORT は、EFS プログラムの呼び出しの前に、次の 5 つの処置コードのうちの 1 つをセットします。

- 0** EFS プログラムのメジャー・コール 1 を示します。DFSORT はこの処置コードを 1 回送ります。
- 4** EFS プログラムのメジャー・コール 2 を示します。DFSORT は、いくつの制御ステートメントが要求され、検出されたかに応じて、メジャー・コール 2 で、この処置コードを複数回送ることがあります。たとえば、SORT、MERGE、および INCLUDE 制御ステートメントのすべてが SYSIN で提供されて要求された場合、EFS プログラムは 2 回呼び出されます。つまり、1 回は SORT 制御ステートメントに対するもの (SORT と MERGE を同時に指定することはできないため、また SORT ステートメントが最初に指定されていると見なすため、SORT ステートメントだけが取られる)、1 回は INCLUDE 制御ステートメントに対するものです。

## EFS インターフェース・パラメーター・リストの構造

- 8 EFS プログラムのメジャー・コール 3 を示します。DFSORT はこの処置コードをブロック・セット手法に対して 1 回、ピアレッジ / ベール手法に対して 1 回送ることができます。
- 12 EFS プログラムのメジャー・コール 4 を示します。DFSORT はこの処置コードを 1 回送ります。
- 16 EFS プログラムのメジャー・コール 5 を示します。DFSORT はこの処置コードを 1 回送ります。

## 制御ステートメント要求リスト

制御ステートメント要求リストは、DFSORT が EFS プログラムへ送る制御ステートメントおよび PARM オプションを記述します。制御ステートメント要求リストは、制御ステートメント命令定義子と PARM オプション名から構成されます。命令定義子または PARM オプション名に指定できる最大長は 8 バイトです。命令定義子または PARM オプション名が長すぎる場合は、DFSORT は最初の 8 バイトだけを使用します。長さフィールドの値には、そのフィールド自身の長さを含めてはなりません。

EFS プログラムで OUTFIL ステートメントを要求できません。

非 DFSORT 命令定義子と PARM オプションは、EBCDIC 形式である必要があります。また、先頭文字に数字は指定できません。制御ステートメント要求リストの形式は次のとおりです。

次の命令定義子または EXEC PARM オプション名を指す連鎖ポインターまたはリストの終わりを示すゼロ	命令定義子または EXEC PARM オプション名の長さ	命令定義子または EXEC PARM オプション名 (可変長)
4 バイト	2 バイト	* バイト

アスタリスク (\*) は、長さに対応する長さフィールドにより決定されることを示します (最大 8 バイト)。

## EFS プログラムへ送られる制御ステートメント・ストリング

DFSORT は、SYSIN、SORTCNTL、DFSPARM、または呼び出しプログラムのパラメーター・リストから要求された制御ステートメントを走査して、連続する制御ステートメント・ストリングを作成します。すなわち、DFSORT は、SYSIN、SORTCNTL、または DFSPARM からの制御ステートメントに必要なすべての連続要求を処理します。DFSORT は要求された PARM オプションを走査して、隣接する PARM オプション・ストリングを作成します。

DFSORT は要求された制御ステートメントまたは PARM オプション・ストリングのコピーを、EFS プログラムの隣接する記憶域に入れます。制御ステートメントにはラベルは付けられません。ストリングのアドレスは、常に、該当する命令定義子または PARM オプションの最初のアドレスを指し示します。

## EFS インターフェース・パラメーター・リストの構造

DFSORT は、要求された制御ステートメントまたは PARM オプションを見つけたら、EFS プログラムに送ります。ただし、DFSORT は、制御ステートメントまたは PARM オプションの構文を制限付きで検査してから、それらを EFS プログラムへ送ります。

ユーザーは、116 ページの『一般的なコーディングの規則』の規則に従うほか、次のような非 DFSORT 制御ステートメントに関する規則も守ってください。

- DFSORT はオペランド指定のない制御ステートメントを認識します。ただし、この場合、命令定義子が、(1) SYSIN、SORTCNTL、または DFSPARM に指定されており、(2) その行で唯一の命令定義子であることが前提になります。
- 命令定義子が 8 バイトより長い場合は、それらが SYSIN、SORTCNTL、DFSPARM、または拡張パラメーター・リストを介して指定され、EFS プログラムにより要求されたものであっても、認知されません。

*z/OS MVS JCL ユーザーズ・ガイド* および *z/OS MVS JCL 解説書* に記載されている規則のほかに、次のような非 DFSORT PARM オプションに関する規則も守ってください。

- EFS プログラムにより要求された PARM オプションが 8 バイトより長い場合は、認識されません。

要求された DFSORT または非 DFSORT 制御ステートメントまたは PARM オプションが、DFSORT 指定変更規則が適用された後でも残っている場合は、DFSORT はそれらを送ります。

重複した DFSORT または非 DFSORT 制御ステートメント、あるいは PARM オプションが、同一のソース (たとえば、SYSIN) を介して指定された場合、DFSORT は最初に現れた制御ステートメントを送ります。2 番目に現れた DFSORT または非 DFSORT 制御ステートメントあるいは PARM オプションを、DFSORT は無視します。

重複した DFSORT または非 DFSORT 制御ステートメントが異なるソース (たとえば、拡張パラメーター・リスト、SORTCNTL、および DFSPARM) を介して指定された場合、DFSORT は、個々のソースの指定変更規則が適用された後も残っている制御ステートメントを送ります。ただし、DFSORT の OPTION および DEBUG 制御ステートメントの場合は例外です (598 ページの『OPTION および DEBUG 制御ステートメントの特殊処理』を参照)。

互いに排他的な DFSORT 制御ステートメント (たとえば、SORT/MERGE) が同一のソース (たとえば、SYSIN) を介して指定された場合、DFSORT は最初に現れた制御ステートメントを送ります。2 番目に現れた DFSORT 制御ステートメントを、DFSORT は無視します。

互いに排他的な DFSORT 制御ステートメント (たとえば、SORT/MERGE) が異なるソース (たとえば、拡張パラメーター・リスト、SORTCNTL、および DFSPARM) を介して指定された場合、DFSORT は、異なるソースの指定変更規則が適用された後で残っている制御ステートメントを送ります。送られなかった DFSORT 制御ステートメントを、DFSORT は無視します。

このように、EFS プログラムには、重複した DFSORT または非 DFSORT 制御ステートメント (ただし、598 ページの『OPTION および DEBUG 制御ステートメント』を参照) を送ります。

## EFS インターフェース・パラメーター・リストの構造

トの特殊処理』で説明している DFSORT の OPTION および DEBUG 制御ステートメントは除く)、または重複した PARM オプションは送られません。

EFS プログラムが DFSORT の OPTION 制御ステートメントで非 DFSORT オペランドを指定し、その OPTION 制御ステートメントを拡張パラメーター・リストの中に含める場合は、すべての DFSORT オペランドの後に、EFS プログラムで非 DFSORT オペランドを指定します。

制御ステートメントまたは PARM スtringのために取得した記憶域があれば、DFSORT はそれを解放します。

**注:** ブランクおよび引用符は、DFSORT が EFS プログラムに送る制御ステートメントを決定するうえで非常に重要です。対になっていない引用符を INCLUDE/OMIT 制御ステートメントに指定できません。DFSORT が、引用符に囲まれたデータを定数として扱い、引用符の外にあるブランクを主区切り文字として扱うためです。

### OPTION および DEBUG 制御ステートメントの特殊処理

DFSORT OPTION および DEBUG 制御ステートメントの両方の指定変更機能は、別々のソースを介して指定された場合、EFS 処理が有効で、EFS プログラムによりいずれかの制御ステートメントまたは両方の制御ステートメントが要求されるときに、特殊処理を必要とします。

たとえば、DFSORT は、OPTION および DEBUG 制御ステートメントについて、次のように指定変更処理を行いません。

- SORTCNTL で提供された OPTION 制御ステートメントは、拡張パラメーター・リスト内に指定された OPTION 制御ステートメント上の対応するオプションを選択的に指定変更します。
- SORTCNTL で提供された DEBUG 制御ステートメントは、24 ビット・パラメーター・リストまたは拡張パラメーター・リストに指定された DEBUG 制御ステートメント上の対応するオプションを選択的に指定変更します。

これらの指定変更機能のため、DFSORT は SORTCNTL で提供された OPTION 制御ステートメントを簡単には送ることができず、また、拡張パラメーター・リストで提供された OPTION 制御ステートメントも送ることができません。EFS プログラムが OPTION 制御ステートメントの可能なオペランドをすべて処理できるように、DFSORT は、SORTCNTL および拡張パラメーター・リストの両方に指定された OPTION 制御ステートメントを送る必要があります。DFSORT は、別々のソースを介して提供された OPTION および DEBUG 制御ステートメントを両方とも送ります。同じソースの中に重複した OPTION または DEBUG 制御ステートメントが指定され、また別のソースにもその OPTION または DEBUG 制御ステートメントが指定されている場合、DFSORT は、別々のソースを介して提供された OPTION および DEBUG 制御ステートメントの両方のうち最初に現れたものを送ります。

## EFS プログラムにより戻される制御ステートメントのストリング

EFS プログラムは、制御ステートメントまたは PARM オプションのストリングを変更して、それを元の連続する記憶域の中に置き換えることができます。その記憶域が小さすぎる場合は、ユーザー・プログラムは新しい連続区域を割り振る必要が

## EFS インターフェース・パラメーター・リストの構造

あります。ストリングが新しい記憶域に戻された場合、ユーザーの EFS プログラムは常にその取得した記憶域を解放します。

EFS プログラムは情報フラグをセットして、ストリング内の制御ステートメントまたは PARM オプションを DFSORT に解析させるか無視させるかを示す必要があります (詳細については、604 ページの『情報フラグ』を参照してください)。

OUTFIL ステートメントは、解析される EFS プログラムからは戻すことはできません。

### 解析に関する規則

解析される変更済みの制御ステートメントの内容と形式は、111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』で説明されている有効な DFSORT の値に対応している必要があります。ただし、SORT または MERGE を指定した FIELDS オペランド、あるいは INCLUDE または OMIT を指定した COND オペランドを使用する場合は例外です (600 ページの『SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形式』を参照)。

解析のために DFSORT に戻される制御ステートメントについて、ユーザーは次の規則に従ってください。

- 命令定義子およびそれに対応するオペランドは、大文字の EBCDIC 形式である必要があります。
- 命令定義子 (たとえば、SORT、MERGE、RECORD など) の後には、少なくとも 1 つの空白が必要です。制御ステートメントは、1 つ以上の空白で始まり、1 つ以上の空白で終わらせることができます。それ以外の空白は、定数の一部の空白でない限り認められません。
- ラベルは使用できません。制御ステートメント名の前の先行空白 (1 つ以上) はオプションです。
- 継続文字は使用できません。
- 注釈ステートメント、空白・ステートメント、および注記は使用できません。

解析される変更済みの EXEC PARM オプションの内容と形式は、34 ページの『EXEC/DFSPARM PARM オプションの指定』で説明されているように、有効な DFSORT 値に対応している必要があります。

次のオペランドは、EFS プログラムにより OPTION 制御ステートメントで戻された場合、DFSORT に無視されます。

EFS  
LIST  
NOLIST  
LISTX  
NOLISTX  
LOCALE  
MSGDDN  
MSGDD  
MSGPRT  
SMF  
SORTDD

## EFS インターフェース・パラメーター・リストの構造

SORTIN  
SORTOUT  
USEWKDD

次の EXEC PARM オプションは、EFS プログラムにより戻された場合は、DFSORT に無視されます。

EFS  
LIST  
NOLIST  
LISTX  
NOLISTX  
LOCALE  
MSGDDN  
MSGDD  
MSGPRT

## SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形式

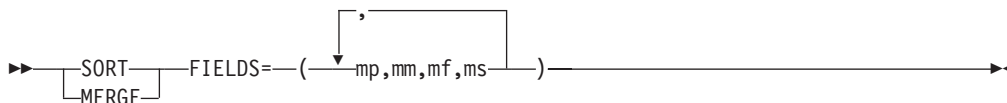
『プログラム制御ステートメント』の項で説明されている SORT、MERGE、INCLUDE、および OMIT 制御ステートメントを使用するほかに、FIELDS および COND パラメーターでさらに 2 つの形式を使用することもできます。それらの形式は D1 および D2 と呼ばれ、次のとおりです。

- D1 は、SORT または MERGE ステートメントの FIELDS パラメーターをもちます。
- D2 は、INCLUDE または OMIT ステートメントの COND パラメーターをもちます。

D1 および D2 を使用すると、それぞれ EFS プログラム出口ルーチン EFS01 と EFS02 による特殊処理を必要とするデータ・タイプを反映させることができます。OUTFIL ステートメントの INCLUDE または OMIT パラメーターをもつ D2 形式を指定できません。

## FIELDS オペランドの D1 形式

FIELDS オペランドで D1 形式を使用する SORT および MERGE ステートメントの構文は、次のとおりです。



### オペランド

#### 内容

- mp** 入力レコード内のフィールドの位置
- mm** フィールドの長さ
- mf** このフィールドを EFS 制御フィールドとして指定する D1 形式

## EFS インターフェース・パラメーター・リストの構造

**ms** 昇順 (A) または降順 (D) のいずれかを指定します。E61 出口による変更 (E) は指定できません。

表 70 は、EFS プログラムにより DFSORT に戻された SORT 制御ステートメントで D1 形式を使用する場合の例を示しています。

次のような D1 形式に関する要件に従ってください。

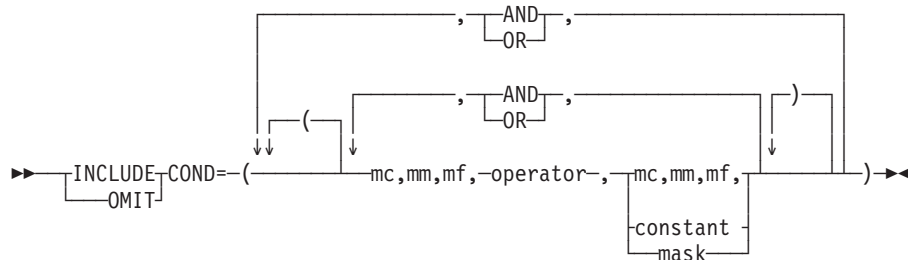
- 戻された mp、mm、および ms 値は、有効な SORT または MERGE 制御ステートメント値である必要があります。ただし、次のような例外があります。
  - mp と mm を組み合わせた値は、レコード長を超えることがあります。
  - CHALT は EFS フィールドには影響を与えないため、その長さを 256 に限定しません。
  - 値 E を ms に指定できません。EFS フィールドは E61 により変更されることはありません。
  - FORMAT=D1 は指定できません。

表 70. EFS プログラムにより戻される D1 形式

EFSPGM に送られる元の SORT 制御ステートメント
SORT FIELDS=(15,4,FF,A,20,4,CH,A,40,7,FF,D)
EFSPGM により戻される変更後の SORT 制御ステートメント
SORT FIELDS=(15,4,D1,A,20,4,CH,A,40,7,D1,D)
ここで、
FF はユーザー定義の形式であり、DFSORT に戻る前に EFS プログラムにより D1 に変更されます。

## COND オペランドの D2 形式

COND オペランドで D2 形式を使用する INCLUDE または OMIT ステートメントの構文は、次のとおりです。



### オペランド

#### 内容

- mc** 関連付け識別子。それぞれの関係条件を識別するために使用される数値
- mm** フィールドの長さ
- mf** 関係条件の範囲内にある EFS フィールドを指定する D2 形式

## EFS インターフェース・パラメーター・リストの構造

### operator

有効な DFSORT の比較演算子またはビット論理演算子

### constant

有効な DFSORT の 10 進数、文字、16 進数、またはビットによる定数

**mask** 有効な DFSORT の 16 進数ストリングまたはビット・ストリング

表 71 は、EFS プログラムにより DFSORT に戻された INCLUDE 制御ステートメントで、関連付け識別子と D2 形式を使用する例を示したものです。

**注:** EFS プログラムにより各関連条件に割り当てられる関連付け識別子の値は、任意の順序で選択できます。表 71 の例は、関連付け識別子の順序を示していません。

次のような D2 形式に関する要件に従ってください。

- 戻された mc、mm、または定数値は、有効な INCLUDE または OMIT 制御ステートメントの値である必要があります。ただし、次のような例外があります。
  - mc と mm を組み合わせた値は、レコード長を超えることがあります。
  - 有効な DFSORT の定数またはマスクであれば、どれでも使用できます。
  - COND=(mc1、mm1、mf1、演算子、mc2、mm2、mf2) を使用する場合、mf1 と mf2 はどちらも D2 である必要があります。
  - CHALT は、EFS フィールドには影響を与えません。
  - FORMAT=D2 は指定できません。

表 71. EFS プログラムにより戻される関連付け識別子および D2 形式

EFSPGM に送られる元の INCLUDE 制御ステートメント
<pre>INCLUDE COND=(15,4,FF,EQ,20,4,FF,AND,40,7,FF,NE,50,7,FF,OR,                 30,2,FF,NE,35,2,FF)</pre>
EFSPGM により戻される変更後の INCLUDE 制御ステートメント
<pre>INCLUDE COND=(1,4,D2,EQ,1,4,D2,AND,2,7,D2,NE,2,7,D2,OR,3,2,D2,NE,3,2,D2)</pre>
ここで、
<ul style="list-style-type: none"><li>• FF はユーザー定義の形式であり、DFSORT に戻る前に EFS プログラムにより変更されます。</li><li>• 指定された最初の関係条件 (1,4,D2,EQ,1,4,D2) は、関連付け識別子の値 1 を使用してこの関係条件を識別します。</li><li>• 指定された 2 番目の関係条件 (2,7,D2,NE,2,7,D2) は、関連付け識別子の値 2 を使用してこの関係条件を識別します。</li><li>• 指定された 3 番目の関係条件 (3,2,D2,NE,3,2,D2) は、関連付け識別子の値 3 を使用してこの関係条件を識別します。</li></ul>

## 元の制御ステートメントの長さ

制御ステートメントには、制御ステートメントの最初のバイトから制御ステートメントの最後のオペランドまでが含まれます。ただし、1 つの命令定義子だけしか定義されていない場合は、その命令定義子の長さが含まれます。DFSORT は、制御ステートメントで提供されているラベルは送りません。



## 変更後の制御ステートメントの長さ

この長さには、制御ステートメントの最初のバイトから制御ステートメントの最後のオペランドまでが含まれます。先行空白が提供される場合は、長さには 1 桁目の先行空白が含まれます。

## EFS プログラム・コンテキスト域

EFS プログラム・コンテキスト域は専用の連絡域で、EFS プログラムが必要に応じてセットアップして使用できるものです。DFSORT は、各メジャー・コールで、および EFS01 と EFS02 への呼び出しごとに、コンテキスト域のアドレスを EFS プログラムへ送ります。

EFS プログラムは、EFS プログラム・コンテキスト域に必要な記憶域を取得 (メジャー・コール 1 で) と、その解放 (メジャー・コール 5 で) を行う必要があります。

## 抽出バッファー・オフセット・リスト

抽出バッファーへのオフセットのリンク・リストは、ユーザーの EFS プログラムに渡されます。このオフセットは、SORT または MERGE の FIELDS オペランドで指定されている任意の EFS 制御フィールドのバッファー域への開始位置を示します。オフセットが送られるのは EFS 制御フィールドに関してだけで、FIELDS オペランドに指定されているのと同じ順序で送られます。EFS 制御フィールドがない場合は、オフセットへのアドレスはゼロになります。

DFSORT は、それが抽出バッファー・オフセット・リストのために獲得したすべての記憶域を解放します。抽出バッファー・オフセット・リストの形式は次のとおりです。

次のオフセットを指す連鎖ポインタまたはリストの終わりを示すゼロ	オフセット n
— 4 バイト —	— 4 バイト —

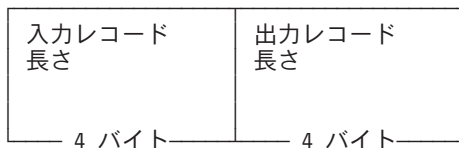
## レコード長リスト

レコード長リストは、入力レコード長および出力レコード長を含むリンク・リストです。実行時にレコード・サイズが変更されることがある (たとえば、E15 ユーザー出口を用いて) ことを知っておく必要があります。

入力レコード長と出力レコード長は、通知のためだけに EFS プログラムへ送られません。DFSORT は、EFS プログラムにより戻されたレコード長の値に対するすべての変更を無視します。

DFSORT は、レコード長リスト用に獲得したすべての記憶域を解放します。レコード長リストの形式は次のとおりです。

## EFS インターフェース・パラメーター・リストの構造



## 情報フラグ

情報フラグは次の図のように定義されます。

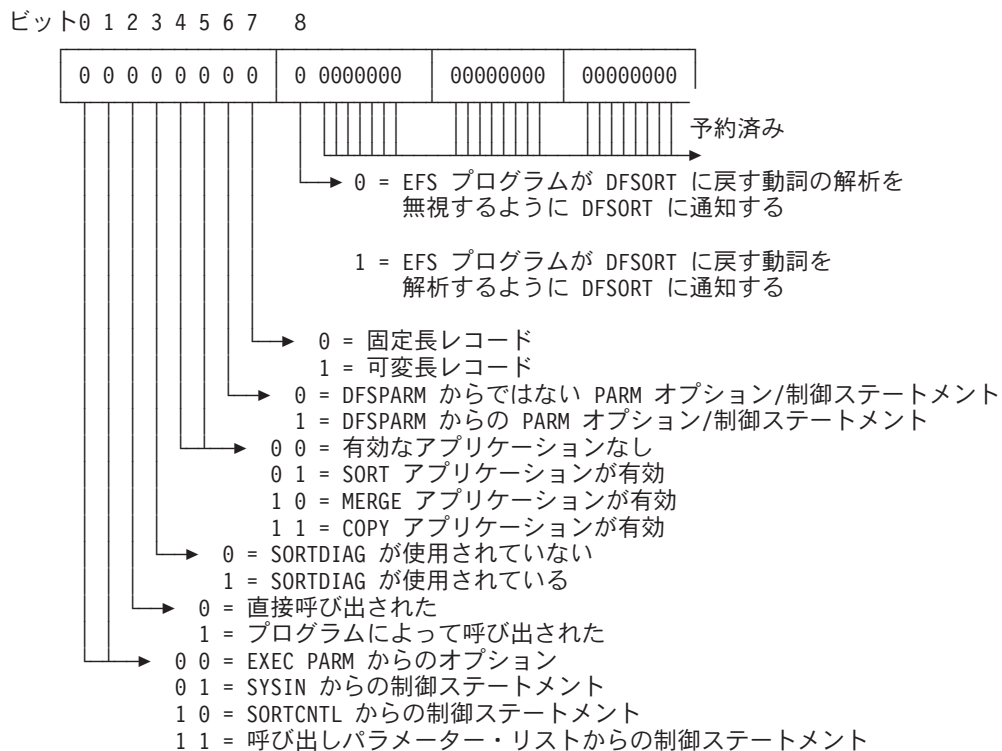


図 47. 情報フラグ

### ビット 説明

#### ビット 0 と 1

処理される制御ステートメントのソースを示します。情報フラグ 0 と 1 は、メジャー・コール 2 で EFS プログラムを呼び出す前に、DFSORT によりセットされます (メジャー・コール 2 では複数の呼び出しが可能です)。

#### ビット 2

DFSORT がどのように呼び出されたかを示します。情報フラグ 2 は、EFS プログラムのメジャー・コール 1 の前に、DFSORT によりセットされます。

#### ビット 3

診断メッセージを印刷するかどうかを示します。情報フラグ 3 は、EFS プログラムのメジャー・コール 1 の前に、DFSORT によりセットされます。

**ビット 4 と 5**

実行する DFSORT 機能を示します。情報フラグ 4 と 5 は、メジャー・コール 2 とメジャー・コール 3 で EFS プログラムの各呼び出しの前に、DFSORT によりセットされます (メジャー・コール 2 およびメジャー・コール 3 では、複数の呼び出しが可能です)。

**ビット 6**

DFSPPARM からの PARM オプションおよび制御ステートメントのソースを示します。情報フラグ 6 は、メジャー・コール 2 で EFS プログラムへの各呼び出しの前に、DFSORT によりセットされます (メジャー・コール 2 では、複数の呼び出しが可能です)。

**ビット 7**

固定長レコードまたは可変長レコードのどちらが処理されるかを示します。情報フラグ 7 は、メジャー・コール 3 で EFS プログラムへの各呼び出しの前に、DFSORT によりセットされます (メジャー・コール 3 では、複数の呼び出しが可能です)。

**ビット 8**

EFS プログラムにより戻される制御ステートメントを解析するか無視するかを DFSORT に通知するために、EFS プログラムによりセットされます。制御ステートメントの印刷は、LISTX/NOLISTX パラメーターにより管理されています (詳細については、181 ページの『OPTION 制御ステートメント』を参照してください)。情報フラグ 8 は、メジャー・コール 2 で各呼び出しから DFSORT へ戻る前に、EFS プログラムによりセットされます (メジャー・コール 2 では、複数の呼び出しが可能です)。

## メッセージ・リスト

EFS プログラムは、通知メッセージまたは重大メッセージを戻すことができます。汎用レジスター 15 の戻りコード 0 は通知メッセージを示し、戻りコード 16 は重大メッセージを示します。メジャー・コールの後で EFS プログラムで送るメッセージがない場合は、EFS プログラムは EFS インターフェース・パラメーター・リスト内のメッセージ・リスト・アドレスをゼロにする必要があります。

メジャー・コール 2 で EFS プログラムが制御ステートメント内に構文エラーを検出した場合、EFS プログラムはそのエラーの位置を示すために、ストリングの先頭からのオフセットを戻すことができます。DFSORT はまずエラーのある制御ステートメントを最初に印刷し、次にオフセットにより示された位置にドル記号 (\$) を含む別の行を印刷します。

DFSORT は相対オフセットと重大メッセージを関連付けるので、EFS プログラムは汎用レジスター 15 に戻りコード 16 を入れて戻す必要があります。相対オフセットが EXEC PARM に対して戻された場合、その相対オフセットは無視されます。EFS プログラムは、メッセージ用に獲得した記憶域を解放します。

長さフィールドの値には、そのフィールド自身の長さを含めてはなりません。

メッセージ・リストの形式は次のとおりです。

## EFS インターフェース・パラメーター・リストの構造

次のメッセージを指し示すポインターまたはリストの終わりを示すゼロ	(構文エラーへの) 相対オフセットまたはゼロ	メッセージ・テキストの長さ	メッセージ・テキスト (可変長)
4 バイト	2 バイト	2 バイト	* バイト

アスタリスク (\*) は、長さに対応する長さフィールドにより決定されることを示します。

DFSORT は、EFS プログラムにより戻されるメッセージ形式については、何の制限も課しません。ユーザーが希望すれば、DFSORT メッセージの形式を使用して、メッセージ・データ・セット内のメッセージの外観を一貫性のあるものにすることができます。DFSORT で使用するメッセージ形式については、DFSORT メッセージ、コード、および診断の手引き リリース 14 を参照してください。

---

## EFS プログラム出口ルーチン

EFS 制御フィールド (D1 形式) または EFS フィールド (D2 形式) を指定すると、DFSORT はそれぞれ EFS01 または EFS02 出口ルーチン呼び出して、それらのフィールドを処理します。これらのルーチンはユーザーの EFS プログラムにより生成され、メジャー・コール 3 でこれらのルーチンについて次のような情報を戻すことができます。

- 抽出ルーチン EFS01 のアドレス。このルーチンは、分類または組み合わせが実行される前に、入力レコードの制御フィールドをバッファ領域に抽出するために使用されます。EFS01 はコピー・アプリケーションには適用されません。
- INCLUDE または OMIT ルーチン EFS02 のアドレス。このルーチンは、レコードの組み込みまたは除外のための比較論理を処理するために使用されます。

これらのルーチンにより使用された記憶域があれば、EFS プログラムは終了フェーズでその記憶域を解放する必要があります。

## EFS01 および EFS02 機能の説明

各 DFSORT 制御ステートメントは、入力データに実行される操作のタイプを DFSORT に記述します。EFS インターフェースを介して、DFSORT は EFS プログラムがユーザー出口ルーチンを提供して DFSORT 制御ステートメントの能力を超えた機能を実行できるようにします。

EFS プログラムは、ユーザー出口ルーチン EFS01 を提供して、DFSORT の SORT/MERGE 制御ステートメントの機能を補足することができ、また、ユーザー出口ルーチン EFS02 を提供して、DFSORT の INCLUDE/OMIT 制御ステートメントの機能を実行できます。

EFS プログラム出口ルーチンを作成するときは、次のことを考慮してください。

- ルーチンは、標準のリンケージ規則に従っている必要があります。
- パラメーターのリンケージと連絡のために DFSORT により使用される汎用レジスターは、オペレーティング・システムの規則に従っている必要があります (607 ページの図 48 を参照)。

- ルーチンは、記述されたインターフェースを使用する必要があります（608 ページの『EFS01 パラメーター・リスト』および 610 ページの『EFS02 パラメーター・リスト』を参照）。

## レジスター

### 用途

- 1 DFSORT は、パラメーター・リストのアドレスをこのレジスターに入れます。
- 13 DFSORT は、標準の保管域のアドレスをこのレジスターに入れます。この区域を使用して、EFS プログラム出口ルーチンにより使用されるレジスターの内容を保管できます。この区域の最初のワードの下位 3 バイトには、文字 SM1 が含まれています。
- 14 DFSORT 戻り点のアドレスが含まれています。
- 15 EFS プログラム出口ルーチンのアドレスが含まれています。このレジスターを、EFS プログラム出口ルーチンの基底レジスターとして使用できます。DFSORT へ戻りコードを渡すために、EFS プログラム出口ルーチンもこのレジスターを使用できます。

図 48. レジスターに関する規則

## EFS01 ユーザー出口ルーチン

EFS01 出口ルーチンを用いてのユーザー定義データ・タイプの処理には、制御ステートメントを変更する機能を使用することが必要です。メジャー・コール 1 と 2 での、EFS プログラム要件は次のとおりです。

- メジャー・コール 1 では、EFS プログラムは SORT または MERGE 命令定義子を指定した制御ステートメント要求リストを提供する必要があります。詳細については、596 ページの『制御ステートメント要求リスト』を参照してください。
- メジャー・コール 2 では、EFS プログラムは新しい形式 D1 を SORT または MERGE 制御ステートメントで戻す必要があります。この形式は、DFSORT に EFS01 ルーチンを呼び出すように通知します (D1 形式で定義された制御フィールドは、EFS 制御フィールドとしても知られています)。詳細については、600 ページの『SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形式』を参照してください。EFS プログラムは、最終位置、長さ、および順序も戻す必要があります。DFSORT は最終位置と長さを使用して、オフセットのリストを作成します。

メジャー・コール 3 では、DFSORT は EFS プログラムにバッファーへのオフセットのリストを送ります。これらのオフセットは、EFS プログラムが EFS01 ルーチンに、EFS 制御フィールドにより示されたデータをバッファー内のどこに移動させる必要があるかを示します。詳細については、603 ページの『抽出バッファー・オフセット・リスト』を参照してください。メジャー・コール 3 では、EFS プログラムは EFS01 ルーチンのアドレスを DFSORT へ戻す必要があります。

## EFS プログラム出口ルーチン

入力フェーズのときに、DFSORT は各入力レコードごとに EFS01 ルーチン呼び出します。EFS01 出口ルーチンは、SORT または MERGE FIELDS オペランドで指定された EFS 制御フィールドにより示されているすべてのデータを、抽出バッファ・オフセット・リスト内のオフセットにより指定されているとおりに、入力レコードから抽出バッファ領域へ移す必要があります。各 EFS 制御フィールドについて、EFS01 によりバッファ領域に移される合計バイト数は、変更後の SORT または MERGE オペランドの *mm* パラメーターに指定されているバイト数と等しくなります。レコードは、変更後の *ms* パラメーターに応じて配列されます。

EFS01 ルーチンは、新しいレコードが入力フェーズにもち込まれるたびに、すべての EFS 制御フィールドを抽出バッファ領域に抽出するために呼び出されます。

DFSORT は抽出バッファ内のデータを使用して分類または組み合わせを行い、そのデータを 2 進データとして処理します。

### EFS01 パラメーター・リスト

DFSORT は、EFS01 ユーザー出口ルーチンが呼び出されるたびに、EFS01 ユーザー出口ルーチンに次のような 3 つのワードを送ります。

- 抽出バッファ領域のアドレス
- 入力レコードのアドレス
- EFS プログラムのコンテキスト域のアドレス

DFSORT は、パラメーター・リストのアドレスをレジスター 1 に入れます。このリストはフルワード境界で始まり、3 フルワードの長さです。パラメーター・リストの形式は次のとおりです。

<b>1 ~ 4 バイト目</b>
抽出バッファ領域のアドレス
入力レコードのアドレス
EFS プログラムのコンテキスト域のアドレス

EFS01 ルーチンは、汎用レジスター 15 に次のいずれかの戻りコードを戻す必要があります。

- 0** EFS 制御フィールドの抽出は正しく行なわれました。
- 16** EFS 制御フィールドの抽出は失敗しました。DFSORT を終了させます。

## EFS02 ユーザー出口ルーチン

EFS02 ユーザー出口ルーチンを用いて、ユーザー定義のデータ・タイプに基づくレコードの組み込みまたは省略を行う場合、制御ステートメントを変更する機能を使用する必要があります。メジャー・コール 1 と 2 での、EFS プログラム要件は次のとおりです。

- メジャー・コール 1 では、EFS プログラムは INCLUDE または OMIT 命令定義子を指定した制御ステートメント要求リストを提供する必要があります。詳細については、596 ページの『制御ステートメント要求リスト』を参照してください。
- メジャー・コール 2 では、EFS プログラムは新しい形式 D2 を INCLUDE または OMIT 制御ステートメントで戻す必要があります。この形式は、EFS02 を呼

## EFS プログラム出口ルーチン

び出すように DFSORT に通知します (D2 形式で定義されるフィールドは、EFS 比較フィールドとしても知られています)。詳細については、600 ページの『SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形式』を参照してください。また、EFS プログラムは、最終の長さを戻さなければならず、さらに位置の値の代わりに、特定の関係条件を示す識別子 (関連付け識別子) を送る必要もあります。EFS フィールドを含む各関連条件について、その特定の関係条件を識別するための固有な関連付け識別子が存在する必要があります。詳細については、600 ページの『SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの EFS 形式』を参照してください。

メジャー・コール 3 で、EFS プログラムは EFS02 ルーチンのアドレスを DFSORT に戻す必要があります。

EFS02 ルーチンは、EFS フィールドを含む各関係条件ごとに、INCLUDE または OMIT 比較論理を実行するために呼び出されます。入力フェーズで、DFSORT は AND、OR、または括弧により定義された評価に従って、各入力レコードごとに 1 回または複数回 EFS02 出口ルーチンを呼び出します。EFS02 出口ルーチンは関連付け識別子を使用して、実行されている現行関係条件を判別する必要があります。EFS02 は関連付け識別子により示された現行関係条件について比較論理を実行します。610 ページの図 49 は 602 ページの表 71 を再現したもので、DFSORT による EFS02 に対する呼び出し順序を示しています。

## EFS プログラム出口ルーチン

EFSPGM に送られる元の INCLUDE 制御ステートメント

```
INCLUDE COND=(15,4,FF,EQ,20,4,FF,AND,40,7,FF,NE,50,7,FF,OR,
              30,2,FF,NE,35,2,FF)
```

EFSPGM により戻される変更後の INCLUDE 制御ステートメント

```
INCLUDE COND=(1,4,D2,EQ,1,4,D2,AND,2,7,D2,NE,2,7,D2,OR,
              3,2,D2,NE,3,2,D2)
```

ここで、EFS02 への呼び出し順序を要約すると、次の表のようになります。

関連条件の場合	EFS02 は 戻りコード 0=真または 4=偽を戻す	次の論理演算子が下記の場合の DFSORT の処置
		AND
関連付け 識別子 1	真	関連付け ID 2 で EFS02 を呼び出す
	偽	関連付け ID 3 で EFS02 を呼び出す

関連条件の場合	EFS02 は 戻りコード 0=真または 4=偽を戻す	次の論理演算子が下記の場合の DFSORT の処置
		OR
関連付け 識別子 2	真	レコードを組み込む
	偽	関連付け ID 3 で EFS02 を呼び出す

関連条件の場合	EFS02 は 戻りコード 0=真または 4=偽を戻す	次の論理演算子が下記の場合の DFSORT の処置
		なし
関連付け 識別子 3	真	レコードを組み込む
	偽	レコードを省略する

図 49. DFSORT による EFS02 への呼び出し順序

### EFS02 パラメーター・リスト

DFSORT は、EFS02 出口ルーチンが呼び出されるたびに、EFS02 出口ルーチンに次のような 3 つのワードを送ります。

- 関連付け識別子のアドレス
- 入力レコードのアドレス
- EFS プログラムのコンテキスト域のアドレス

DFSORT は、パラメーター・リストのアドレスをレジスター 1 に入れます。このリストはフルワード境界で始まり、3 フルワードの長さです。パラメーター・リストの形式は次のとおりです。



1 バイト目	2 バイト目	3 バイト目	4 バイト目
00	00	00	関連付け識別子
入力レコードのアドレス			
EFS プログラムのコンテキスト域のアドレス			

EFS02 出口ルーチンは、汎用レジスター 15 の中に次のいずれかの戻りコードを戻す必要があります。

**0** 真

レコードは EFS フィールドの関連条件についての、INCLUDE または OMIT テストにパスしました。適用できる場合は、次の関連条件を用いて処理が継続されます。適用できない場合は、DFSORT は INCLUDE が指定されている場合はレコードを受け入れ、OMIT が指定されている場合はレコードを除外します。

**4** 偽

レコードは EFS フィールドの関連条件についての、INCLUDE または OMIT テストにパスしませんでした。適用できる場合は、次の関連条件を用いて処理が継続されます。適用できない場合は、DFSORT は INCLUDE が指定されている場合はレコードを除外し、OMIT が指定されている場合はレコードを組み込みます。

**16** 終了

INCLUDE または OMIT 論理の処理中にエラーが生じました。DFSORT を終了させます。

## EFS プログラムのユーザー出口ルーチンのアドレッシング・モードおよび常駐モード

DFSORT は次のような機能を提供して、16 メガバイト境界より上または下に常駐させ、24 ビットまたは 31 ビットのアドレッシングのいずれかを使用できるようにします。

f (EFS プログラム出口のルーチン・アドレスのビット 0)

- 0** 24 ビットのアドレッシングを有効にして、EFS プログラム出口を呼び出します。
- 1** 31 ビットのアドレッシングを有効にして、EFS プログラム出口を呼び出します。

EFS プログラムのユーザー出口ルーチンは、有効な 24 ビットまたは 31 ビット・アドレッシングで DFSORT に戻ることができます。DFSORT がレジスター 14 に入れた戻りアドレスを、使用する必要があります。

EFS プログラムのコンテキスト域アドレス (DFSORT はこのアドレスを変更せずに EFS プログラム出口に送る) の場合を除き、DFSORT は EFS プログラム出口ルーチンのパラメーター・リスト・アドレス (すなわち、EFS プログラム出口ルーチンのパラメーター・リストを指し示すポインターとパラメーター・リストの中のアドレス) を次のように処理します。

## アドレッシング・モードと常駐モード

- 24 ビットのアドレッシングを有効にして EFS プログラム出口ルーチンに入る場合は、DFSORT は消去された (最初の 8 ビットがすべてゼロの) 24 ビットのアドレスまたは 31 ビットのアドレスを EFS プログラム出口ルーチンに渡すことができます。EFS プログラム出口ルーチンが 31 ビット・アドレッシングを有効にして DFSORT へ戻る場合は、EFS プログラム出口ルーチンは、消去された 24 ビットのアドレスを戻す必要があります。
- 31 ビットのアドレッシングを有効にして EFS プログラム出口ルーチンに入る場合は、DFSORT は消去された 24 ビットのアドレスまたは 31 ビットのアドレスを EFS プログラム出口ルーチンに渡すことができます。EFS プログラム出口ルーチンは、31 ビットのアドレスまたは消去された 24 ビットのアドレスを戻す必要があります。

---

## ユーザーが指定しなければならない EFS プログラム戻りコード

ユーザーの EFS プログラムは、次の 2 つの戻りコードのうちの 1 つを DFSORT へ渡す必要があります。

### 0 処理を続行する

このメジャー・コールについて DFSORT に処理を続行させたい場合は、汎用レジスター 15 にゼロの戻りコードを指定して戻ります。

### 16 DFSORT を終了する

このメジャー・コールについて DFSORT に処理を終了させたい場合は、汎用レジスター 15 に 16 の戻りコードを指定して戻ります。

メジャー・コール 4 より前に EFS プログラムがメジャー・コールから 16 の戻りコードを戻した場合、または EFS プログラムが生成したユーザー出口ルーチンの 1 つが 16 の戻りコードを戻した場合、適用できる箇所で、中間のメジャー・コールを EFS プログラムまたはユーザー出口ルーチンまでスキップし、メジャー・コール 4 とメジャー・コール 5 で EFS プログラムを呼び出します。

メジャー・コール 2 とメジャー・コール 3 では、複数の呼び出しが可能です。メジャー・コール 2 で、複数の呼び出しの 1 つから、EFS プログラムが戻りコード 16 を戻した場合、該当すれば、そのメジャー・コール 2 での後続の呼び出しは完了します。メジャー・コール 3 で、複数の呼び出しの 1 つから、EFS プログラムが戻りコード 16 を戻した場合、該当すれば、そのメジャー・コール 3 での後続の呼び出しは完了しません。

EFS プログラムがメジャー・コール 4 で戻りコード 16 を戻した場合、DFSORT はそれ以降もメジャー・コール 5 で EFS プログラムを呼び出します。

---

## レコード処理の順序

EFS を使用する際のレコード処理の順序は、これを使用していないときの処理と同じです。614 ページの図 50 は、分類または組み合わせの場合のレコード処理の順序を示し、615 ページの図 51 は、EFS 処理が実施されている場合の順序を示しています。

この 2 つの図は、9 ページの図 2 と同じ時点を示していますが、次のような例外があります。

- レコード処理が **INCLUDE** または **OMIT** 制御ステートメントに関して行われる場合は、EFS02 ユーザー出力ルーチンが呼び出され、EFS フィールドの関連条件について比較論理が行われます。
- レコード処理が **SORT** または **MERGE** 制御ステートメントに関して行われる場合は、EFS01 ユーザー出力ルーチンが呼び出され、EFS 制御フィールドについて抽出処理が行われます。

# レコード処理の順序

分類アプリケーション

組み合わせアプリケーション

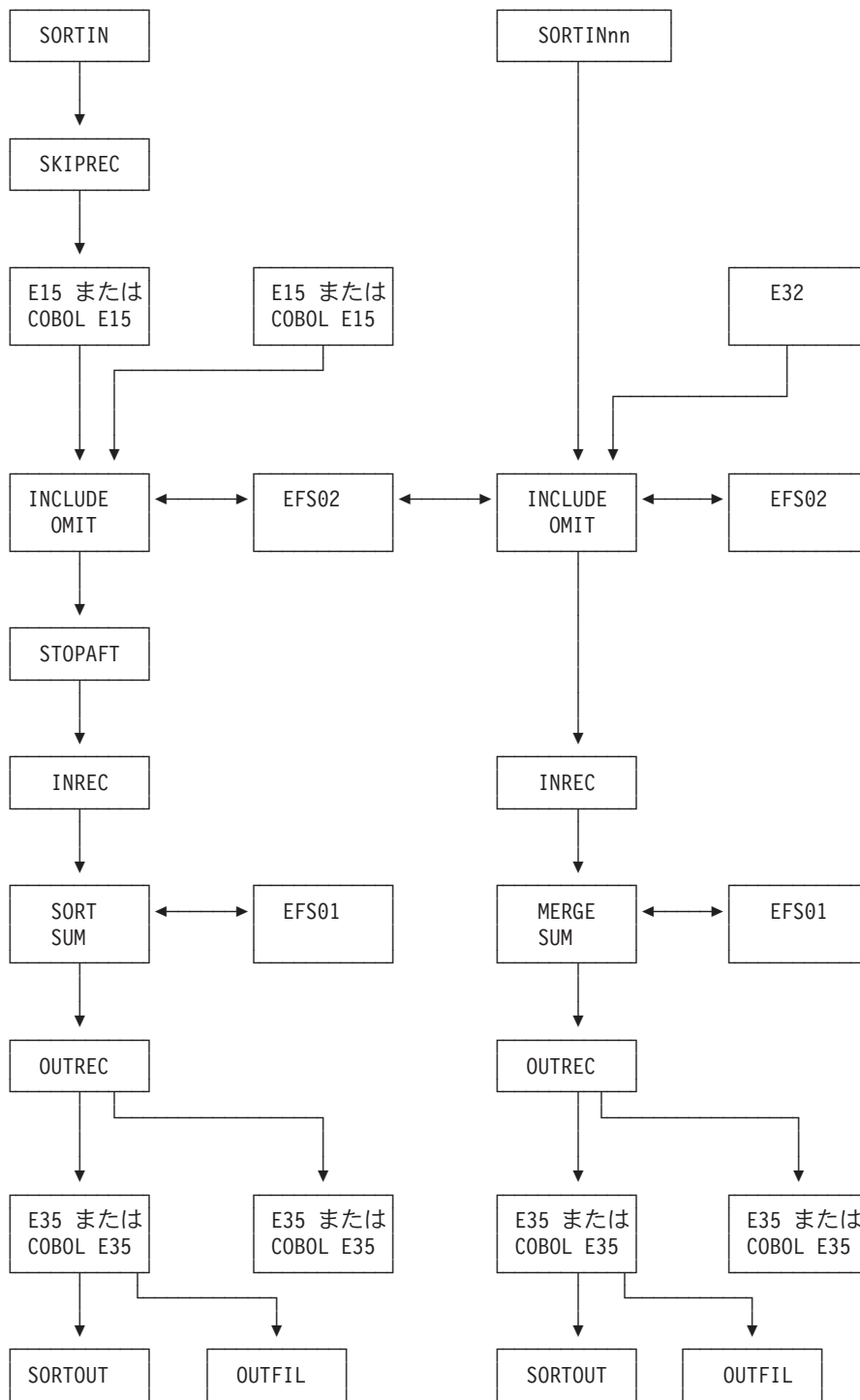


図 50. EFS レコード処理の順序 (分類または組み合わせの場合)

コピー・アプリケーション

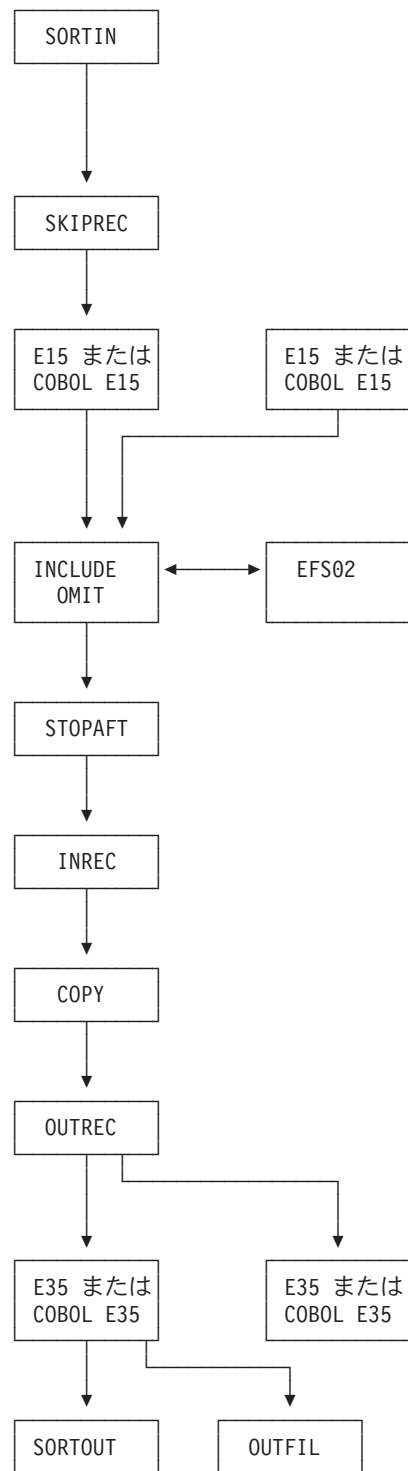


図 51. EFS レコード処理の順序 (コピーの場合)

### SNAP ダンプの要求の方法

メジャー・コール 1 以外の任意のメジャー・コールの前後で、診断目的のために SNAP ダンプを要求できます。DEBUG ステートメントで EFSDPBFR パラメーターまたは EFSDPAFT パラメーターを使用します。

正しい構文については、123 ページの『DEBUG 制御ステートメント』を参照してください。

### EFS プログラムの例

次の例は、EFS プログラムを使用して実行時に制御ステートメントを変更する方法を示しています。

この例の DFSORT 実行については、次のことを想定しています。

- EFS プログラム『EFSPGM』は、DFSORT のモジュールと同じライブラリー内に常駐しています。
- アプリケーション用の JCL ステートメントは次のとおりです。

```
//EXAMPLE1 JOB A12345,'J. SMITH'  
//S1 EXEC PGM=SORT,PARM='EFS=EFSPGM'  
//SYSOUT DD SYSOUT=A  
//SORTIN DD DSN=SMITH.INPUT,DISP=SHR,  
// UNIT=3380,SPACE=(TRK,(15,2)),VOL=SER=XYZ003,  
// DCB=(LRECL=80,BLKSIZE=80,RECFM=F)  
//SORTOUT DD DSN=SMITH.OUTPUT,DISP=(NEW,KEEP),  
// UNIT=3380,SPACE=(TRK,(15,2)),VOL=SER=XYZ003  
//SYSIN DD *  
SORT FIELDS=(5,20,CH,A,13,5,BI,D)  
OPTION STOPAFT=30,DYNALLOC=3390  
/*
```

### DFSORT 初期設定フェーズ

#### メジャー・コール 1

メジャー・コール 1 の前に、DFSORT は EFS インターフェースのパラメーター・リストの中に、次のようなフィールドをセットします。

- 処置コード=0  
メジャー・コール 1 が有効です。
- 情報ビット・フラグ 2=0  
DFSORT 実行は JCL により呼び出されます。
- 情報ビット・フラグ 3=0  
SORTDIAG は有効ではありません。

DFSORT はメジャー・コール 1 で EFS プログラム EFSPGM を呼び出し、EFSPGM は EFS インターフェース・パラメーター・リストに次のフィールドをセットします。

- 制御ステートメント要求リスト

OPTION 制御ステートメントが EFSPGM により要求されていることを、DFSORT に対して指示する OPTION 命令定義子を含みます。

- EFSPGM プログラム・コンテキスト域  
EFSPGM はこのコンテキスト域を使用することになります。
- メッセージ・リスト=0  
EFSPGM には、DFSORT がメッセージ・データ・セットへ印刷するメッセージがありません。汎用レジスター 15 はゼロにセットされます。

## メジャー・コール 2

メジャー・コール 2 の前に、DFSORT は EFS インターフェースのパラメーター・リストの中に、次のようなフィールドをセットします。

- 処置コード=4  
メジャー・コール 2 が有効です。
- 情報ビット・フラグ 4=0 および情報ビット・フラグ 5=0  
実施されているアプリケーションはありません。

EFSPGM は OPTION 制御ステートメントを要求しました。DFSORT は、要求された各制御ステートメントに対して、EFS プログラム EFSPGM を呼び出します。この場合は 1 つです。DFSORT は、EFS インターフェース・パラメーター・リストに次のフィールドもセットします。

- 情報ビット・フラグ 0=0 および情報ビット・フラグ 1=1  
要求された制御ステートメントは SYSIN からのものです。
- すべてのオペランドとそれに対応するサブパラメーターを含む、元の OPTION 制御ステートメント。  
OPTION STOPAFT=30,DYNALLOC=3390
- すべてのオペランドとそれに対応するサブパラメーターを含む、元の OPTION 制御ステートメントの長さ。  
元の制御ステートメント・ストリングの長さは 31 バイトです。

DFSORT はメジャー・コール 2 で EFS プログラム EFSPGM を呼び出し、EFSPGM は EFS インターフェース・パラメーター・リストに次のフィールドをセットします。

- 情報ビット・フラグ 8=1  
DFSORT は、EFSPGM により戻される制御ステートメントを解析する必要があります。
- すべてのオペランドとサブパラメーターを含む、変更後の OPTION 制御ステートメント。  
OPTION STOPAFT=30,DYNALLOC=3380,EQUALS
- すべてのオペランドとサブパラメーターを含む、変更後の OPTION 制御ステートメントの長さ。  
変更後の制御ステートメント・ストリングの長さは 38 バイトです。
- メッセージ・リスト=0  
EFSPGM には、DFSORT がメッセージ・データ・セットへ印刷するメッセージがありません。汎用レジスター 15 はゼロにセットされます。

## EFS プログラムの例

表 72 は、EFS プログラム EFSPGM に送られる元の制御ステートメントと、EFS プログラム EFSPGM により戻される変更後の制御ステートメントを示したものです。

表 72. 元の制御ステートメントと変更後の制御ステートメント

EFSPGM に送られる <b>元</b> の OPTION 制御ステートメント
OPTION STOPAFT=30,DYNALLOC=3390
EFSPGM に送られる <b>変更後</b> の OPTION 制御ステートメント
OPTION STOPAFT=30,DYNALLOC=3380,EQUALS
ここで、
STOPAFT=30 は元オペランドと値です。
DYNALLOC=3380 は新しい値を指定された元オペランドです。
EQUALS オプションが追加されています。

### メジャー・コール 3

メジャー・コール 3 の前に、DFSORT は EFS インターフェースのパラメーター・リストの中に、次のようなフィールドをセットします。

- 処置コード=8  
メジャー・コール 3 が有効です。
- 情報ビット・フラグ 4=0 および情報ビット・フラグ 5=1  
分類アプリケーションが実施されています。
- 情報ビット・フラグ 7=0  
固定長レコードが処理されています。
- レコード長リストの値=80  
入出力データ・セットの LRECL は 80 です。SORTOUT LRECL が指定されなかったため、DFSORT は SORTOUT LRECL にデフォルト SORTIN LRECL を使用しました。
- 抽出バッファー・オフセット・リスト=0  
SORT 制御ステートメントに EFS 制御フィールドが指定されていませんでした。

DFSORT はメジャー・コール 3 で EFS プログラム EFSPGM を呼び出し、EFSPGM は EFS インターフェース・パラメーター・リストに次のフィールドをセットします。

- EFS01 アドレス=0  
SORT 制御ステートメントに EFS 制御フィールドの指定がないため、EFS01 ユーザー出力ルーチンは使用されません。  
INCLUDE 制御ステートメントが (EFS フィールドで) 指定されなかったため、EFS02 ユーザー出力ルーチンは使用されません。
- メッセージ・リスト=0



EFSPGM には、DFSORT がメッセージ・データ・セットへ印刷するメッセージがありません。汎用レジスター 15 はゼロにセットされます。

## DFSORT 終了フェーズ

### メジャー・コール 4

メジャー・コール 4 の前に、DFSORT は EFS インターフェースのパラメーター・リストの中に、次のようなフィールドをセットします。

- 処置コード=12

メジャー・コール 4 は有効です。

DFSORT はメジャー・コール 4 で EFS プログラム EFSPGM を呼び出し、EFSPGM は EFS インターフェース・パラメーター・リストに次のフィールドをセットします。

- メッセージ・リスト=0

EFSPGM には、DFSORT がメッセージ・データ・セットへ印刷するメッセージがありません。

さらに、汎用レジスター 15 はゼロにセットされます。

### メジャー・コール 5

メジャー・コール 5 の前に、DFSORT は EFS インターフェースのパラメーター・リストの中に、次のようなフィールドをセットします。

- 処置コード=16

メジャー・コール 5 は有効です。

DFSORT はメジャー・コール 5 で EFS プログラム EFSPGM を呼び出し、EFSPGM は EFS インターフェース・パラメーター・リスト内のフィールドのセットは行いませんが、汎用レジスター 15 はゼロにセットします。



## 第 9 章 効率の改善

パフォーマンスの向上 . . . . .	622
パフォーマンスを最大にするためのユーザー・アプリケーションの設計 . . . . .	623
DFSORT 処理の直接呼び出し . . . . .	623
新しいアプリケーション設計の事前計画 . . . . .	623
効率のよいブロック化 . . . . .	623
効率的な分類 / 組み合わせ手法の指定 . . . . .	624
分類手法 . . . . .	624
組み合わせ手法 . . . . .	624
入出力データ・セット特性の正確な指定 . . . . .	625
入力ファイル・サイズ . . . . .	625
可変長レコード . . . . .	625
直接アクセス記憶装置 . . . . .	625
順次式ストライピングの使用 . . . . .	626
圧縮の使用 . . . . .	626
DFSMSrmm 管理テープ、または ICETPEX の使用 . . . . .	626
SmartBatch パイプの使用 . . . . .	626
拡張記憶域での VIO の使用 . . . . .	626
経過時間を改善する装置の指定 . . . . .	626
パフォーマンスを向上させるオプションの使用 . . . . .	627
CFW . . . . .	627
COBEXIT. . . . .	627
DSA . . . . .	627
DPSIZE . . . . .	627
FASTSRT. . . . .	628
SDB. . . . .	629
HIPRMAX . . . . .	629
DFSORT の高速で、効率的な生産性機能の使用 . . . . .	629
INCLUDE または OMIT、STOPAFT、および SKIPREC . . . . .	629
OUTFIL . . . . .	629
LOCALE . . . . .	630
SUM . . . . .	630
ICETOOL. . . . .	630
パフォーマンスを低下させるオプションの回避 . . . . .	630
CKPT . . . . .	630
EQUALS . . . . .	631
EQUCOUNT. . . . .	631
LOCALE . . . . .	631
NOCINV . . . . .	631
NOBLKSET . . . . .	631
VERIFY . . . . .	631
テープ作業データ・セット . . . . .	631
ユーザー出口ルーチン . . . . .	631
ダイナミック・リンク編集 . . . . .	631
EFS プログラム . . . . .	631
主記憶域の効率的な使用 . . . . .	631
主記憶域の調整 . . . . .	632
主記憶域の解放 . . . . .	634
一時ワークスペースの効率的な割り振り . . . . .	636

## 効率の改善

直接アクセス作業記憶装置 . . . . .	636
作業データ・セットの仮想入出力 . . . . .	636
テープ作業記憶装置 . . . . .	637
ハイパー分類の使用 . . . . .	637
データ空間を使用する分類 . . . . .	638
IEBGENER に代わる、ICEGENER の使用 . . . . .	639
ICEGENER の戻りコード . . . . .	642
SAS システム用の DFSORT のパフォーマンス・ブースターの使用 . . . . .	643
DFSORT の BLDINDEX サポートの使用 . . . . .	643

---

## パフォーマンスの向上

DFSORT は、パフォーマンスを自動的に最適化するように設計されています。DFSORT は最適化変数 (たとえば、バッファ・サイズ) をセットして、いくつかの分類手法または組み合わせ手法の中から最も効率的なものを選択します。

次のようないくつかの方法で、DFSORT のパフォーマンスを向上させることができます。

- パフォーマンスを最大にするように、ユーザーのアプリケーションを設計する。
  - DFSORT 処理を直接呼び出す。
  - 新しいアプリケーションを設計する場合、事前に計画を立てる。
  - 効率的な分類 / 組み合わせ手法を指定する。
  - 入力 / 出力データ・セットの特性を正確に指定する。
  - 順次式ストライピングを使用する。
  - 圧縮を使用する。
  - DFSMSrmm 管理テープ、または ICETPEX を使用する。
  - SmartBatch パイプを使用する。
  - 拡張記憶域での VIO の使用
  - 経過時間を改善する装置を指定する。
  - パフォーマンスを向上させるオプションを使用する。
  - DFSORT の高速で効率的な生産性機能を使用する。
  - パフォーマンスを低下させるオプションを使用しない。
- 主記憶域を効率的に使用する。
- 一時ワークスペースを効率的に割り振る。
- ハイパー分類を使用する。
- データ空間を使用して分類する。
- IEBGENER の代わりに ICEGENER を使用する。
- DFSORT のパフォーマンス・ブースターを SAS システムに使用する。
- DFSORT の BLDINDEX サポートを使用する。

本章で説明している事項の多くについては、DFSORT の *DFSORT Tuning Guide R14* に説明されています。

## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

ユーザー・アプリケーションを実行するときに、DFSORT は自動的にパフォーマンスを最適化しますが、使用可能なリソースを DFSORT が最良の方法で利用できるような指定やオプションを使用することにより、さらに効率を高めることができます。

### DFSORT 処理の直接呼び出し

DFSORT を COBOL または PL/I プログラムから呼び出す代わりに、JCL を使用して呼び出すことにより、パフォーマンスを向上させることができます。一般に、COBOL または PL/I は便利であるために使用されます。しかし、便利さと引きかえに、パフォーマンスが低下することがあります。最適なパフォーマンスを得るために DFSORT のインストール・デフォルトと実行時オプションを調整し、特に INCLUDE/OMIT、INREC/OUTREC、SUM、OUTFIL などの『一緒に作用する』制御ステートメントを使用することにより、効率を上げることができます。これらの機能を使用すると、別のプログラムからのルーチンを使用せずに、入力ファイルからレコードを削除したり、レコードを再フォーマット設定して不要なフィールドを削除したり、レコードを算術的に結合できます。

### 新しいアプリケーション設計の事前計画

新しいアプリケーションを設計する際には、いくつかの要因を考慮する必要があります。いくつかの要因を以下に示します。

可能な限り、次のようにしてください。

- EBCDIC 文字または 2 進数の制御フィールドのどちらかを使用する。
- 2 進数制御フィールドの始めと終わりを、バイト境界に合わせる。
- 代替照合順序による文字変換の使用を避ける。
- 固定小数点制御フィールドに常に正の値が含まれていると分かっている場合は、そのフィールドを 2 進数フィールドとして指定する。
- パック 10 進数またはゾーン 10 進数の制御フィールドに、常に同じ符号 (たとえば X'C') の正の値が含まれていると分かっている場合は、そのフィールドを 2 進数フィールドとして指定する。
- ゾーン 10 進数形式よりもパック 10 進数形式を使用する。
- 優先順位が正しい順序で並んでいる文字制御フィールドまたは 2 進数制御フィールドを、同じ順序 (昇順または降順) で分類または組み合わせる場合は、それらを 1 つの制御フィールドとして指定する。
- 制御フィールドの重複を避ける。
- SORT、MERGE、INCLUDE、または OMIT 文字フィールドが、2 進エンコードのデータを使用して処理できる場合は、ロケール処理の使用を避ける。

### 効率のよいブロック化

入出力レコードを効率的にブロック化すると、DFSORT のパフォーマンスを大幅に向上させることができます。可能な限り、データ・セットにシステム決定の最適ブロック・サイズを使用するようにしてください。

## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

出力データ・セットのシステム決定最適ブロック・サイズを DFSORT に選択させる方法の詳細は、181 ページの『OPTION 制御ステートメント』の SDB オプションの説明を参照してください。

## 効率的な分類 / 組み合わせ手法の指定

さまざまな条件に応じて、DFSORT は分類および組み合わせのための各種の手法を選択します。メッセージ ICE143I は、どの手法が選択されたかを知らせます。

コピー・アプリケーションの場合、ブロック・セット手法だけが使用されます。ユーザー・プログラムがブロック・セットを使用できない場合は、DFSORT はエラー・メッセージ ICE160A を出して処理を停止します。

### 分類手法

どの分類手法を DFSORT が選択するかに影響を与える条件の 1 つとして、中間記憶域に使用する装置のタイプがあります。テープ装置を使用すると、従来の手法が使用されますが、この手法はあまり効率が良くありません。中間記憶域にテープ装置を使用する方法の詳細については、637 ページの『テープ作業記憶装置』を参照してください。

ブロック・セットおよびピアレッジ / ベール手法は、DASD 作業データ・セットでだけ使用できます。これらの手法について、これから説明します。

**ブロック・セット分類手法:** DFSORT の最も効率的な手法である FLR- ブロック・セット (固定長レコードの場合) と VLR- ブロック・セット (可変長レコードの場合) は、大部分の分類アプリケーションに使用されます。

### 注:

- ブロック・セット手法は、ピアレッジ / ベール手法より多くの中間ワークスペースを必要とすることがあります。詳細については、636 ページの『一時ワークスペースの効率的な割り振り』を参照してください。
- ブロック・セット手法を選択しない場合は、SORTDIAG DD ステートメントを使用して、メッセージ ICE800I を発行させ、ブロック・セットが使用できない理由を示すコードを表示できます。

**ピアレッジ / ベール分類手法:** ブロック・セット分類手法を使用するための条件が満たされない場合、DFSORT はピアレッジ / ベール手法を使用します。

### 組み合わせ手法

組み合わせアプリケーションの場合は、DFSORT はブロック・セット手法と従来の手法を使用します。

**ブロック・セット組み合わせ手法:** DFSORT の最も効率的な手法である FLR- ブロック・セット (固定長レコードの場合) と VLR- ブロック・セット (可変長レコードの場合) は、大部分の組み合わせアプリケーションに使用されます。

**注:** ブロック・セット手法を選択しない場合は、SORTDIAG DD ステートメントを使用して、メッセージ ICE800I を発行させ、ブロック・セットが使用できない理由を示すコードを表示できます。

## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

**従来の組み合わせ手法:** ブロック・セット組み合わせ手法を使用するための条件が満たされない場合、DFSORT はピアレッジ / ベール手法を使用します。

### 入出力データ・セット特性の正確な指定

DFSORT は与えられた情報 (実行する処理に関する) を使用して、最高の効率を得るための最適化を行います。ユーザーが間違っただけの情報を指定したり、またはデータ・セット・サイズやレコード形式などの情報を指定しなかった場合は、プログラムが既定値を採用しますが、それが正しくなければ、効率が低下したり、プログラムが終了してしまうことがあります。

#### 入力ファイル・サイズ

DFSORT は、入力ファイルのサイズに関する正確な情報をもっている場合、主記憶域と中間作業用記憶域の両方とも最も効率的に使用できます。入力ファイル・サイズを指定する時期および方法については、690 ページの『ファイル・サイズおよび動的割り振り』を参照してください。

#### 可変長レコード

入力データ・セットが可変長レコードから構成され、中間データ・セットの動的割り振りが使用される場合は、OPTION ステートメントに `AVGRLLEN=n` を使用して、できるだけ正確に平均レコード長を指定してください。

#### 直接アクセス記憶装置

記憶域を、トラックやブロックではなく、シリンダー単位で指定すると、システム・パフォーマンスが向上します。可能な場合は、分類作業データ・セット上の記憶域が、シリンダー用に再調整されます。直接アクセス装置に関するシリンダー当たりのトラック数を表 73 に示します。

表 73. 直接アクセス装置の 1 シリンダー当たりのトラック数

装置	トラック数 (1 シリンダー当たり)
3380	15
3390	15
9345	15

WRKSEC が有効で、作業データ・セットが仮想入出力域に割り振られていない場合は、DFSORT は、JCL で要求されなくても、必要に応じて 2 次エクステンントを割り振ります。

作業データ・セットを割り振る場合、通常、入力データ・セットが使用するスペースの 2 倍を割り振るのが妥当です。条件によりは、さらに追加のスペースが必要になることがあります。その条件としては、次のものがあります。

- 制御ワードが長い場合 (151 バイト以上)
- 異なる装置タイプまたは作業データ・セットを使用する場合
- 代替照合順序を使用する場合
- 入力ファイル・サイズに対する使用可能記憶域の比率が低い場合

DCB の LRECL パラメーターと、データ・セットに含まれる実際の最大レコード長とが対応するように注意してください。

## 順次式ストライピングの使用

順次式ストライピングを使用すると、DFSORT がデータの読み取りと書き込みに費やす経過時間を大幅に減らすことができます。経過時間のパフォーマンスを向上させる方法として、DFSORT の入出力データ・セットに順次式ストライピングを使用することをお勧めします。

## 圧縮の使用

圧縮を使用すると、多くのタイプのデータに必要な DASD 記憶域および DFSORT がそのデータの読み取りや書き込みに費やす結果時間を大幅に減少させることができます。経過時間のパフォーマンスを向上させる方法として、DFSORT の入出力データ・セットに圧縮を使用することをお勧めします。

## DFSMSrmm 管理テープ、または ICETPEX の使用

DFSMSrmm により管理されるテープ、または ICETPEX を使用するテープ管理システムの使用により、DFSORT は、入力ファイル・サイズおよびデータ・セット特性についての正確な情報を取得できます。これにより、パフォーマンスが向上し、主記憶域と中間作業用記憶域の両方を最も効率よく使用できます。

## SmartBatch パイプの使用

SmartBatch の入出力パイプを使用すると、『書き込み』ジョブから並行『読み取り』ジョブへのデータのパイピングを行う際に生じる並行操作により、経過時間を大幅に短縮できます。たとえば、SORTOUT データ・セットをパイプ接続すると、DFSORT 出力処理を受取ジョブの入力処理とオーバーラップさせることができます。さらに、パイプは、DASD やテープ・データ・セットではなく、仮想記憶待ち行列であるため、データ転送時間と経過時間が大幅に短縮されます。

経過時間とデータ転送時間を向上させる方法として、適切に、DFSORT 入出力データ・セット用に SmartBatch パイプを使用することをお勧めします。

## 拡張記憶域での VIO の使用

一時非 VSAM 入出力データ・セットを、仮想入出力 (VIO) を使用する拡張記憶域に保留できます。DASD またはテープ・データ・セットではなく、拡張記憶域が使用されるので、データ転送時間と経過時間を大幅に短縮できます (ただし、データ・セット全体が使用可能な拡張記憶域に収まる場合)。たとえば、SORTOUT データ・セットが一時非 VIO データ・セットとして拡張記憶域に割り振られた場合、DFSORT の出力処理だけでなく、受取ジョブの入力処理もパフォーマンスの向上を示します。

経過時間とデータ転送時間を向上させる方法として、適切に、DFSORT 入出力データ・セット用に拡張記憶域で VIO を使用することをお勧めします。

VIO は、一般的には作業データ・セット用にはお勧めしません。636 ページの『作業データ・セットの仮想入出力』の説明を参照してください。

## 経過時間を改善する装置の指定

DFSORT で DASD を使用する際に最良の経過時間を得るには、SORTIN、SORTWKdd、SORTOUT、および OUTFIL データ・セット用に 3390 を使用する必



## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

必要があります。これらのデータ・セットに 3380 と 3390 を混用すると、3390 だけを使用した場合とは異なる経過時間改善の結果が生じることがあります。これは DFSORT の処理手法が間接的に影響するためですが、基本的には、3390 と比較して 3380 のパフォーマンス特性が低いことによります。

3390 使用時に見られる厳密な経過時間の改善は、DFSORT が特定の実行に使用する処理手法により異なるほか、どのデータ・セット (SORTIN、SORTWKdd、SORTOUT、OUTFIL) が 3390 上にあるかによりも異なります。すべての 3390 を使用することができない場合は、SORTWKdd データ・セットよりも SORTIN、SORTOUT、および OUTFIL データ・セットに 3390 を使用することをお勧めします。

## パフォーマンスを向上させるオプションの使用

最適なパフォーマンスを得るために、インストール時と実行時に指定したオプションを調整できます。パフォーマンスを向上させることができるオプションのいくつかを以下に示します。

### CFW

3390 キャッシュ記憶域管理を利用してブロック・セット分類のパフォーマンスを向上させる場合は、DEBUG 制御ステートメントに CFW を指定するか、あるいはインストール・デフォルトとして CFW=YES (CFW=YES が IBM 提供のデフォルト) を指定します。

### COBEXIT

DFSORT との COBOL II インターフェースを利用してパフォーマンスを向上させるには、VS COBOL II、COBOL for MVS & VM、COBOL for OS/390 & VM または COBOL for z/OS および OS/390 でコンパイルされたユーザー出口を実行するときに、OPTION 制御ステートメントに COBEXIT=COB2 を指定するか、またはそれをインストール・デフォルトとして定義します。

### DSA

ブロック・セット分類アプリケーションの場合は、動的ストレージ調整 (DSA) を使用することにより、パフォーマンスを向上させることができます。

DSA インストール・パラメーターは、SIZE/MAINSIZE=MAX が有効であるとき、ブロック・セット分類アプリケーションの動的ストレージ調整に対して、DFSORT に使用可能な記憶域の最大量をセットします。TMAXLIM 値より大きい DSA 値を指定すると、パフォーマンスの向上が可能な場合、DFSORT が TMAXLIM 値より大きい記憶域を使用することを認めます。DFSORT は、パフォーマンスを向上させるのに必要な記憶域を、最大 DSA 値まで、できるだけ多く獲得しようと試みません。

### DSPSIZE

ブロック・セット手法を使用する分類アプリケーションの場合は、データ空間分類を使用することにより、パフォーマンスを向上させることができます。

## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

DSPSIZE パラメータは、データ空間分類で使用するデータ空間の最大量をセットします。デフォルトの DSPSIZE=MAX を使用すれば、DFSORT は、分類するファイルのサイズおよびユーザー・システムのページング活動に基づいて、使用するデータ空間の最大量を選択できます。

### FASTSRT

VS COBOL II またはそれ以降の FASTSRT コンパイラ・オプションを指定すると、DFSORT プロセッサ時間、EXCP、および経過時間を大幅に短縮できます。FASTSRT を用いると、DFSORT の入出力操作はさらに効率良く行われます。COBOL ではなく DFSORT が入出力を行うためです (628 ページの図 52 を参照)。詳細については、VS COBOL II またはそれ以降の資料を参照してください。

SORT ステートメントで入出力プロシージャを使用する場合は、FASTSRT オプションは入出力に対して機能しません。入力または出力プロシージャで通常実行される機能の多くは、DFSORT の INREC、OUTFIL、OUTREC、INCLUDE、または OMIT、STOPAFT、SKIPREC、および SUM 機能で実行される機能と同じです。適切な DFSORT プログラム制御ステートメントをコーディングし、かつそれを DFSPARM (DFSORT)、SORTCNTL (DFSORT)、または IGZSRTCD (COBOL) データ・セットのいずれかに入れ、それによりユーザーの SORT ステートメントが FASTSRT を修飾できるようにすれば、ユーザーの入出力プロシージャを省略できます。

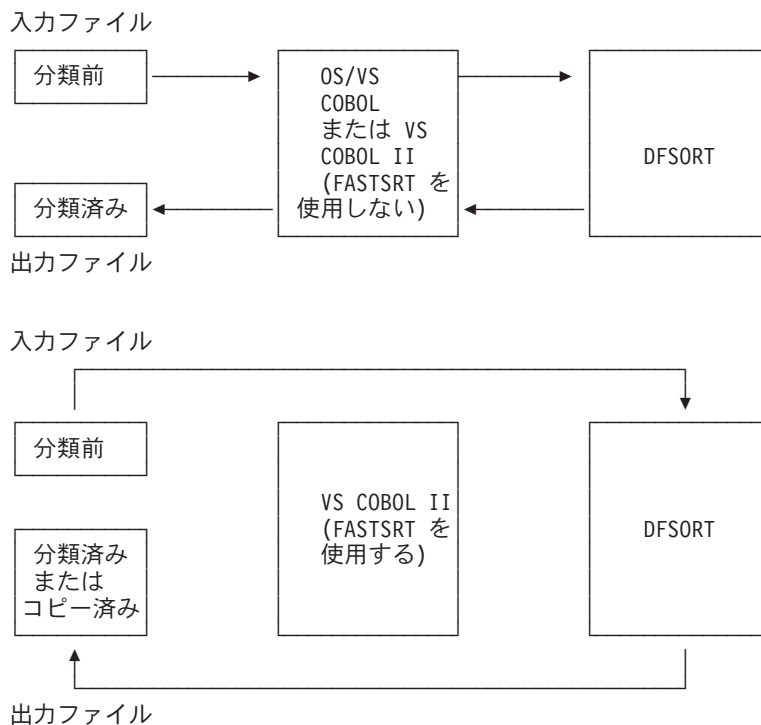


図 52. VS COBOL II による高速分類

## SDB

ブロック・セットの経過時間および DASD とテープの使用効率を向上させるには、ユーザーのサイトのインストール・デフォルトとして SDB=LARGE を指定してください (IBM 提供のデフォルトは、SDB=INPUT です)。SDB=LARGE を指定すると、DFSORT は適宜、ユーザーの DASD およびテープ出力データ・セットのためにシステム決定の最適ブロック・サイズを選択できます。

## HIPRMAX

一時記憶域用に DASD とともにハイパー空間を使用することにより、ブロック・セット分類のパフォーマンスを向上させることができます。

HIPRMAX パラメーターは、実行中にコミットするハイパー空間の最大値をセットします。HIPRMAX=OPTIMAL を指定すると、DFSORT は実行中の他のシステムおよび並行ハイパー分類の活動に応じて、実行時にコミットするハイパー空間の最大量を最適化できます。システム上でのハイパー分類の全活動を、DFSORT インストール・オプションの EXPMAX、EXPOLD、および EXPRES で、さらに制限できます。詳細について、181 ページの『OPTION 制御ステートメント』の HIPRMAX の説明を参照してください。

## DFSORT の高速で、効率的な生産性機能の使用

DFSORT には、高速で、効率的な一組の優れた生産性機能があります。これらの機能は、各種のタスクを実行するためのユーザー・コードの作成およびデバッグの余分な手間を省き、これらのタスクをさらに効率的に実行できます。各機能については、本書の該当個所で詳しく説明しています。本節では、これらの機能のパフォーマンスについて説明します。

### INCLUDE または OMIT、STOPAFT、および SKIPREC

INCLUDE または OMIT ステートメントのいずれかと、STOPAFT および SKIPREC オプションを使用して、処理するレコードの数を減らすことができます。これにより、プロセッサ時間とデータ転送時間を短縮できます。

- INCLUDE および OMIT ステートメントを使用すれば、フィールドを定数や他のフィールドと比較することにより、レコードを選択できます。
- STOPAFT オプションを使用すれば、分類またはコピー処理のために受け入れるレコードの最大数を指定できます。
- SKIPREC オプションを使用すれば、入力ファイルの最初のレコードをスキップし、残りのレコードのみを分類またはコピーできます。

### OUTFIL

同一の入力データ・セットから複数の出力データ・セットを作成する必要がある場合は、OUTFIL を使用して入力データ・セットを一度だけ読み込むことができます。これにより、パフォーマンスを向上させることができます。OUTFIL を分類アプリケーション、組み合わせアプリケーション、およびコピー・アプリケーション用に使用して、複雑なフィルター操作、編集、移行、検索と置換、および報告書機能を提供できます。

単一の出力データ・セットだけを作成し、OUTFIL の機能は必要ない場合は、OUTFIL ではなく、SORTOUT を使用して、最高のパフォーマンスを得ることができます。

## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

### LOCALE

LOCALE オプションを使用して、活動ロケールの照合規則に基づく文字データの分類および組み合わせ処理を行うことができます。これにより、以前は、ユーザー・データの事前処理または事後処理（あるいはその両方）を行ってからしか得られなかった結果を DFSORT で得ることができます。このように手間のかかる処理を省くことにより、時間と処理リソースを節約できます。

### SUM

SUM を使用してフィールドの内容を追加することにより、パフォーマンスを向上させることができます。SUM ステートメントは、指定された SUM フィールドの内容を同じ制御フィールドをもつレコードに追加します。結果が 1 つのレコードに入れられ、他のレコードは削除されます。これにより、DFSORT が出力するレコードの数が減ります。

ZDPRINT=YES インストール・オプションまたは ZDPRINT 実行時オプションを使用して、合計して得られる正のゾーン 10 進数フィールドを印刷可能に指定できます。つまり、ゾーンの最終桁を 16 進の C から 16 進の F へ変更するように、DFSORT に指示できます。

**重複レコードの排除:** SUM 制御ステートメントの使用時に、次のように指定することにより、重複キーの付いたレコードを排除できます。

```
SUM FIELDS=NONE
```

レコード処理ステートメント、ユーザー出口、およびオプションの処理順序については、9 ページの図 2 を参照してください。

### ICETOOL

ICETOOL は多目的ユーティリティです。これによりユーザーは、DFSORT の効率の高い入出力と処理を使用して、単一ジョブ・ステップ内の 1 つ以上のデータ・セットについて複数の操作を実行できます。ICETOOL の 12 の演算子を使用して、ユーザーは、分類、コピー、統計、および報告書の操作を迅速かつ効率的に実行できます。

## パフォーマンスを低下させるオプションの回避

オプションにはパフォーマンスに悪影響を与えるものがあり、このようなオプションは、必要な場合のみ使用します。たとえば、チェックポイント / 再始動を活動化する CKPT オプションを指定すれば、効率的なブロック・セット手法が使用できなくなります。

### CKPT

CKPT オプションは、より効率的なブロック・セット手法の使用を妨げる場合があります。

**注:** インストール・デフォルト IGNCKPT=YES が選択されていると、DFSORT はチェックポイント / 再始動要求を無視して、ブロック・セット手法を選択します。

## パフォーマンスを最大にするためのユーザー・アプリケーションの設計

### **EQUALS**

EQUALS オプションは、レコードの比較およびデータ転送に必要な時間を増加させます。

### **EQUCOUNT**

EQUCOUNT オプションを使用すると、等しいキーをもつレコードの数をカウントするために余分な時間がかかります。

### **LOCALE**

LOCALE オプションは、アプリケーションの実行に必要な時間を増加させる場合があります。

### **NOCINV**

NOCINV オプションは、より効率的な VSAM 処理のための制御インターバル・アクセスの使用を妨げます。

### **NOBLKSET**

NOBLKSET オプションは、より効率的なブロック・セット手法の使用を妨げます。

### **VERIFY**

VERIFY オプションは、余分な処理を含むため、パフォーマンスを低下させます。

### **テープ作業データ・セット**

中間記憶域用にテープ作業データを使用すると、さらに効率的なディスク手法の使用が妨げられます。

### **ユーザー出口ルーチン**

ユーザー出口ルーチンがアプリケーションに組み込まれると、通常、そのアプリケーションの実行に必要な時間が増加します。

大部分のユーザー出口ルーチンが必要とする実行時間は、通常、短いものですが、ユーザー出口 E15、E32、および E35 のルーチンは、データ・セットの各レコードごとに呼び出されます。大きな入力データ・セットの場合、これらのルーチンの合計実行時間は、かなり大きくなる可能性があります。

### **ダイナミック・リンク編集**

ユーザー出口ルーチンのダイナミック・リンク編集は、パフォーマンスを低下させます。

### **EFS プログラム**

EFS プログラムがアプリケーションに組み込まれると、そのアプリケーションの実行に必要な時間が増加する場合があります。

## **主記憶域の効率的な使用**

一般的に、DFSORT が使用できる主記憶域を大きくするほど、パフォーマンスも向上します。過剰なページングを避けるために、使用された主記憶域の量をバックアップできるのに十分な実記憶域があることを確かめてください。これは、特に 32 メガバイトより大きい主記憶域の場合は重大です。DFSORT が使用できる主記憶域のデフォルトの大きさは、DFSORT のインストール時に定義されます。

## 主記憶域の効率的な使用

DFSORT は最小限 88 キロバイトを必要としますが、より高いパフォーマンスを得るためには、さらに大きな記憶域を使用する必要があります。推奨できる記憶域の大きさは、約 4 メガバイトです。パフォーマンスの向上は、大きな入力ファイルを使用すると顕著に現れます。

**注:** ブロック・セットが選択されると、DFSORT は選択されたバッファを 16 メガバイト境界より上に置くことができます。これにより、REGION サイズを増やさずに、より大きな記憶域を DFSORT 用に解放できます。DFSORT が記憶域を効果的に使用できるようにするためには、少なくとも 440 キロバイトの REGION サイズが使用可能である必要があります。

### 主記憶域の調整

REGION 値または MAINSIZE/SIZE 値のいずれかにより、DFSORT が使用できる記憶域の大きさを決定できます。詳細については、*DFSORT 導入とカスタマイズ リリース 14*を参照してください。

一般的に、(仮想) 主記憶域を割り振る最も効率の良い方法は、明示的にまたはデフォルトで MAINSIZE/SIZE=MAX を使用することです。しかし、TMAXLIM または MAXLIM インストール・オプションの値が過度に高く (または低く) セットされると、問題が起こることがあります。これらの値の設定方法についての指針は、*DFSORT 導入とカスタマイズ リリース 14* を参照してください。

**注:** パスワードをユーザー出口ルーチンから入力する場合は、パスワードで保護されているデータ・セットに SIZE/MAINSIZE=MAX を使用できません。SIZE/MAINSIZE=MAX を使用すると、DFSORT が初期設定フェーズでそのデータ・セットをオープンして必要な計算を行うことができないためです。

MAINSIZE/SIZE=n を指定し、n に MINLIM インストール・オプションに指定された値よりも小さい値を指定すると、MINLIM が使用されます。

SIZE/MAINSIZE=MAX が有効であると、DFSORT は、パフォーマンスを向上するために、動的ストレージ (DSA) 機能を適宜使用します。DSA インストール・パラメーターの詳細については、*DFSORT 導入とカスタマイズ リリース 14*を参照してください。

MINLIM 値が、EXEC ステートメントの REGION に指定された値よりも大きい場合は、DFSORT は MINLIM に指定された値の使用を試みます。DFSORT は、MINLIM で指定された大きさを獲得できない場合でも、少なくとも 88 キロバイト (16 メガバイト境界より下) が使用できれば、引き続き実行を試みます。

DFSORT は最小 88 キロバイト (16 メガバイト境界より下) を必要としますが、必要な最小主記憶域の大きさはアプリケーションにより異なります。

最高のパフォーマンスを得るためには、最小限の大きさよりさらに大きな主記憶域を使用することを特にお勧めします。

一般に次のものを使用する場合は、より多くの主記憶域が必要になります。

- スパン・レコード
- COBOL ユーザー出口ルーチン
- CHALT または SMF オプション

- ALTSEQ、INCLUDE、OMIT、SUM、OUTREC、または INREC 制御ステートメント
- 非常に大きなブロックまたは論理レコード
- VSAM データ・セット
- 拡張機能サポート (EFS) プログラム
- ICETEXIT ルーチン
- 大きな ICEIEXIT ルーチン
- OUTFIL 制御ステートメント (特に OUTFIL データ・セットが指定されている場合、またはデータ・セットのブロック・サイズが大きい場合)
- ロケール処理
- 多数の JCL または動的に割り振られた作業データ・セット

OUTFIL 処理で使用する記憶域は、次のいくつかの要因により、自動的に調整されます。

- 使用可能記憶域の合計
- 非 OUTFIL 処理の必要記憶域
- OUTFIL データ・セットの数およびその属性 (たとえば、ブロック・サイズ)

OUTFIL 処理は、ODMAXBF 限界およびユーザー・システムの記憶域限界 (たとえば、IEFUSI) の制約を受けますが、DFSORT 記憶域限界 (つまり、SIZE/MAINSIZE、MAXLIM、および TMAXLIM) の制約は受けません。DFSORT は、可能な限り、16 メガバイト境界より上の記憶域を、OUTFIL 処理に使用しようとします。

**注:**

1. 場合によりは、このリリースの DFSORT は、以前のリリースの DFSORT よりも多くの記憶域を、同程度のアプリケーションに使用することがあります。これにより、一部のアプリケーションの操作が影響を受けることがあります。たとえば、以前のリリースでは、記憶域内分類として (SORTWKdd データ・セットを使用せずに) 実行されたアプリケーションが、このリリースでは記憶域内で実行されないことがあります。割り振られる記憶域の大きさは、通常、TMAXLIM により制御されます。DFSORT が満足いくパフォーマンスを得るためには、少なくとも 440 キロバイトの REGION サイズが使用可能である必要があります。領域値がそれより小さい場合、または DFSORT が 16 メガバイト境界より下にバッファを割り振る必要がある場合は、記憶域の割り振りに悪い影響を与えることがあります。
2. 極端に大きな分類処理 (たとえば、500 メガバイトまたはそれ以上のデータ) の場合は、ハイパー分類およびデータ空間分類を使用可能にするか、あるいは DFSORT が 16 メガバイト以上の主記憶域を使用できるようにする必要があります。

TMAXLIM、MAXLIM、MINLIM、および REGION の間の関係は、一連の抑制と均衡として説明できます。

システム・プログラマーは、ユーザー・システムの主要分類要件に従って、記憶域のデフォルトをセットしています。夜間ウィンドウまたはバッチ時間ウィンドウに従わなければならない場合、(REGION または SIZE/MAINSIZE=n を使用して) 記憶

## 主記憶域の効率的な使用

域を増加させることにより、時間的制約からいくらか解放されます。プロセッサ時間が気になる場合は、(REGION または SIZE/MAINSIZE=n を使用して) 記憶域を減らすことにより、小ファイルの分類に関連するプロセッサ時間を減らすことができます。

通常、DFSORT が使用できる記憶域の大きさを変化させると、次のようないくつかのことが起こります。

### 1. 記憶域の大きさを増加させた場合

- EXCP が減少します。
- 比較的大きなファイルの場合は、プロセッサ時間は、通常減少します。つまり、余分の記憶域の管理に要するオーバーヘッドは、DFSORT にとってデータ引き渡しの回数が少なくすむことにより相殺されることとなります。
- 負荷が非常に大きなシステムでは、DFSORT がより頻繁にスワップアウトされるため、経過時間が増加することがあります。
- 非常に小さな分類処理の場合は、余分の記憶域の管理に要するオーバーヘッドのため、プロセッサ時間はそのまま変わらないか、増大することがあります。比較的大きなファイルの場合は、プロセッサ時間は、通常減少します。余分な記憶域の管理に要するオーバーヘッドよりも、DFSORT によるデータ引き渡しの回数が少なくすむという利点のほうが大きいからです。

### 2. 記憶域の大きさを減少させた場合

- EXCP が増加します。
- 大部分の分類処理で経過時間が増加します。
- プロセッサ時間は、非常に小さいファイルでは減少しますが、比較的大きなファイルでは増加します。

主記憶域の割り振りを変更すると、システムの効率に影響を与えることがあります。割り振る主記憶域の大きさを減少させることにより、ユーザーは、DFSORT のパフォーマンスを低下させて、他のプログラムが同時に作動するために必要な記憶域を得ることができます。割り振りを増やすと、大きな DFSORT アプリケーションを効果的に実行できる代わりに、マルチプログラミングを共用する他のアプリケーションの効率が低下します。

## 主記憶域の解放

状況によりは、DFSORT はユーザーの REGION 内のすべての使用可能記憶域を使用することがあります。このような状況は、16 メガバイト境界より上の記憶域については、通常起こりません (起こった場合は、ARESINV または ARESALL オプションを使用するか、あるいは SIZE/MAINSIZE 値を低くします)。本節では、ユーザーの REGION 内の記憶域を解放する方法について説明します。

SIZE/MAINSIZE=n が有効で、n が REGION パラメーターまたはデフォルトの REGION 値より大きい場合、あるいは、SIZE/MAINSIZE=MAX および TMAXLIM がユーザーの REGION よりも大きい場合は、解放する必要のある記憶域を次の方法で決定します。

- ユーザー出口を使用するアプリケーションの場合
  - 直接呼び出す DFSORT の場合は、次のいずれかを選択できます。
  - MODS 制御ステートメントの m パラメーターを使用します。



- SIZE=MAX が有効になっている場合は、RESALL オプションを使用できません。
- REGION を変更して、REGION が SIZE/MAINSIZE より大きくなるようにします (その差が使用可能になります)。
- インストール・パラメーター OVERRGN がユーザー・システムの IEFUSI 値よりも小さい場合、その差が使用可能になります。(OVERRGN はインストール・オプションで、システム・プログラマーだけが変更できるものです。)
- プログラムで呼び出す DFSORT の場合、次のいずれかを選択できます。
  - ユーザー出口アドレスがパラメーター・リストで渡されない (すなわち、MODS ステートメントで指定する) 場合、MODS ステートメントに *m* を使用します。
  - ユーザー出口アドレスがパラメーター・リストで渡され、SIZE/MAINSIZE=MAX が有効になっている場合は、RESINV オプションを使用します。
  - ユーザー出口アドレスがパラメーター・リストで渡され、SIZE/MAINSIZE=n が有効になっている場合は、REGION を変更して、REGION を SIZE/MAINSIZE より大きくします (その差が使用可能になります)。
  - ユーザーの DFSORT アプリケーションの多くがパラメーター・リストでユーザー出口アドレスを渡し、SIZE/MAINSIZE=n が有効になっている場合は、システム・プログラマーは OVERRGN 値を変更して IEFUSI 値より小さくする必要があります。
- ユーザー出口を使用しないアプリケーションの場合
  - 直接呼び出す DFSORT の場合は、次のいずれかを選択できます。
    - SIZE/MAINSIZE=MAX が有効になっている場合は、RESALL オプションを使用します。
    - SIZE/MAINSIZE=n が有効になっている場合は、REGION を変更して、REGION を SIZE/MAINSIZE より大きくします (その差が使用可能になります)。
    - システム・プログラマーに、OVERRGN 値を変更して IEFUSI 値より小さくしてもらいます。
  - プログラムで呼び出す DFSORT の場合、次のいずれかを選択できます。
    - SIZE/MAINSIZE=MAX が有効になっている場合は、RESINV オプションを使用します。
    - SIZE/MAINSIZE=n が有効になっている場合は、REGION を変更して、REGION を SIZE/MAINSIZE より大きくします (その差が使用可能になります)。
    - システム・プログラマーに、OVERRGN 値を変更して IEFUSI 値より小さくしてもらいます。

SIZE/MAINSIZE が REGION より小さい場合、SIZE/MAINSIZE とユーザーの REGION 指定値 (またはデフォルト) との差が、システムまたはユーザー出口ルーチンが使用するために十分な記憶域になるようにしてください。

## 一時ワークスペースの効率的な割り振り

### 一時ワークスペースの効率的な割り振り

複数のチャンネルが使用できると、パフォーマンスは向上します。また、各装置とプログラムを実行している処理装置との間に 2 つのチャンネル・パスができるように装置を接続した場合も、パフォーマンスが向上します。

#### 直接アクセス作業記憶装置

装置、記憶域、およびチャンネルを効率的に使用することにより、プログラムのパフォーマンスは向上します。中間記憶域データ・セットを定義する DD ステートメントの UNIT パラメーターに特定の装置タイプ (たとえば UNIT=3390) を指定すると、DFSORT は区域を割り当て、ある種の最適化が自動的に行われます。次のようなときに直接アクセス中間記憶装置を使用すると、最高のパフォーマンスが得られます。

- 2 つ以上の作業データ・セットを使用する。
- 最も高速のデータ転送速度を備えた記憶装置を選択する。
- アクチュエーターごとに 1 つの作業データ・セットを割り当てる。
- 同じタイプの装置を使用する。
- 装置に 2 つのチャンネル経路を使用する。
- すべての作業データ・セットを同じ大きさ、またはできるだけ近い大きさにする。
- SORTWKdd データ・セットが装置またはチャンネルを、SORTIN、SORTOUT、または OUTFIL データ・セットと共用しないようにする。SmartBatch パイプを入出力データ・セットに適宜使用して、競合を避けます。
- 作業データ・セットに連続スペースを指定し、自動 2 次割り振りが不要になるように、十分な 1 次スペースを確保する。
- 687 ページの『非同期記憶サブシステム』で説明されているように、作業データ・セットが非同期装置に割り振られる場合は、十分な仮想記憶域を用意する。

DFSORT が SORTWKdd の書き出し中に入力を読み取り、SORTWKdd からの読み取り中に出力を書き込むことができる場合は、経過時間が短縮します。たとえば、2 つのチャンネルがある場合、最良の割り振り方法は、SORTIN、SORTOUT、および OUTFIL データ・セットに 1 つのチャンネルを、SORTWKdd データ・セットにもう 1 つのチャンネルを割り振ることです。

異なるディスク手法の記憶域所要量は、685 ページの『付録 A. ワークスペースの使用』のガイドラインを使用することにより見積もることができます。

#### 作業データ・セットの仮想入出力

VIO データ・セットは多くのアプリケーションのパフォーマンスを向上させることができますが、これらの作業データ・セットは、通常、DFSORT の DASD 作業データ・セットほどは効果的ではありません。

仮想装置 (VIO) に割り振られた DFSORT 一時作業データ・セットは、経過時間を減少させることができますが、逆に DFSORT アプリケーションの CPU 時間が増加します。一般に、これは良い方法ではなく、次のような場合を除き、VIO を DFSORT 作業データ・セットに使用できません。

- システムが拡張記憶域で VIO をサポートし、かつ
- 経過時間が最も重要である場合

## 一時ワークスペースの効率的な割り振り

作業データ・セットを VIO に割り振る場合、ICEMAC オプション VIO は、DFSORT に VIO をどのように取り扱うか指示します。

- VIO=YES を指定すれば、DFSORT は作業データ・セットでの VIO の使用を受け入れます。
- VIO=NO を指定すれば、DFSORT は作業データ・セットを仮想装置から実装置に再度割り振ります。再割り振りを成功させるには、仮想装置と同じ装置タイプの実装置が使用可能でなければならないことに注意してください。

再割り振りを行わない (VIO=YES) より、VIO データ・セットの再割り振りを行う (VIO=NO) ことをお勧めします。ただし、再割り振りは時間とリソースを浪費するため、まず VIO 作業データ・セットを使用しないようお勧めします。

## テープ作業記憶装置

テープ作業記憶装置を使用すると、より効果的なブロック・セット手法が使用できません。テープ中間記憶装置を使用して最高のパフォーマンスを得るためには、最高速タイプのテープ装置を 6 台以上使用することです。一般に、中間記憶装置には、使用可能なテープをできるだけ多く使用する必要があります。使用するテープの数を大きくすれば、それだけ 1 回のパスで組み合わせることができるストリングの数が増加し、これにより中間組み合わせフェーズで要求されるパスの数は減少します。このため経過時間が減少するほか、入出力操作の数もしばしば減少します。

しかし、作業単位の数を増やしても、中間記憶域に使用するブロック・サイズが小さくなります。バッファに使用できる主記憶域が比較的小さい場合は、これは重大な要因になることがあります。たとえば、DFSORT が 88 キロバイトしか操作に使用できない場合、5 つ以上のテープ作業単位を使用しても、おそらく何の改善も見られません (かえって、低下する可能性があります)。したがって、できるだけ多くのテープを使用するという一般規則は、DFSORT が 100 キロバイトより多くの記憶域を使用できる場合のみ適用してください。

各種のテープ手法を使用する場合の記憶域所要量の決定方法については、685 ページの『付録 A. ワークスペースの使用』を参照してください。

### 注:

1. DFSORT 作業ファイルの処理中にテープの方向転換が頻繁に起こると、IBM 3420 磁気テープ装置よりも、IBM 3480/3490/3590 磁気テープ・サブシステムのほうが、効率的なデータ転送速度に多くの影響を受けることになります。この特性があるため、中間記憶域におけるこれらのテープ装置間のパフォーマンス比較は、信頼性のあるものにはならず、また大幅に変動する可能性もあります。
2. 改良データ記録機能 (IDRC) を使用する装置は、逆方向読み取りコマンドでは正しく実行しないため、中間記憶装置としてはお勧めしません。

## ハイパー分類の使用

ハイパー分類はハイパー空間を使用します。ハイパー空間とは、拡張記憶域または 64 ビット実モードの中央記憶装置に常駐する高パフォーマンスのデータ空間であり、必要な場合に、補助記憶装置により提供されます。ハイパー分類を使用すると、ブロック・セット分類時に、レコードの一時記憶域として DASD の代替用としてハイパー空間が DASD とともに使用されます。ハイパー分類は入出力処理を減少

## ハイパー分類を使用する。

させるので、その結果、経過時間、EXCP、およびチャネル使用も減少します。入力または出力が圧縮順次データ・セットまたは VSAM データ・セットの場合は、ハイパー分類をお勧めします。

ハイパー分類アプリケーションのハイパー空間の最大の大きさは、HIPRMAX パラメーターを使用して制御できます。HIPRMAX は、実行開始時に使用可能な記憶域に基づいて、ハイパー空間の最大の大きさを動的に決定するように DFSORT に指示を出すことができます。CPU 時間の最適化を図ることが第一の目的の場合は、HIPRMAX を使用してハイパー分類を抑制することもできます。ハイパー分類は多少 CPU 時間を低下させることがあるためです。

ハイパー分類アプリケーションが使用するハイパー空間の実際の大きさは、いくつかの要因により変化します。詳細については、181 ページの『OPTION 制御ステートメント』の HIPRMAX の説明を参照してください。最も重要なことは、実行時に、DFSORT が使用可能な記憶域の大きさを決定するだけでなく、他の並行ハイパー分類アプリケーションに必要な記憶域の大きさも決定するということです。この情報に基づいて、DFSORT は、記憶域の不足が予想される場合、あるいはシステムでの全ハイパー分類活動が、DFSORT のインストール・オプション EXPMAX、EXPOLD、および EXPRES でセットされた限界に達した場合は、ハイパー空間の使用から DASD 作業データ・セットへの使用へ動的に切り替えます。これらのインストール・オプションの詳細については、DFSORT 導入とカスタマイズリリース 14 を参照してください。

## データ空間を使用する分類

データ空間分類 はデータ空間を使用して、DFSORT のブロック・セット手法を使用する分類アプリケーションのパフォーマンスを向上させます。

データ空間分類 を使用すると、DFSORT は大量のデータを一度に分類できます。これにより、CPU 時間および経過時間が減少します。

データ空間分類で使用するデータ空間の最大の大きさは、DSPSIZE オプションを使用して制御できます。DSPSIZE=MAX では、DFSORT は使用するデータ空間の最大の大きさを選択できます。この場合、使用される大きさは分類するファイルの大きさおよびユーザー・システムのページ活動に応じて決まります。DSPSIZE=0 は、DFSORT がデータ空間分類を使用しないということを意味します。

以下の機能およびデータ・セット・タイプは、データ空間分類用にはサポートされません。

- 入力としての、スプール、ダミー、またはパイプ・データ・セット、または HFS ファイル。
- ユーザー出口
- INREC、OUTFIL、OUTREC、および SUM
- EQUCOUNT

データ空間分類が 2 ～ 3 メガバイトのような非常に小さなデータ・セットに使用されることはほとんどありません。小さなデータはまとめて主記憶域に含め、分類処理を行うのが効率的であるためです。

データ空間分類を使用するには、十分な大きさの中央記憶装置、つまり、分類処理の開始時に SRM により報告された、未使用またはしばらく使用されていない中央記憶装置が使用できる必要があります。このような記憶域は、DFSORT が必要とする該当データ空間をサポートするために必要です。必要なデータ空間の大きさは変化します。一般的には、分類するデータの量が増えればデータ空間のこの大きさも増え、指定された主記憶域の大きさが増えるとデータ空間は小さくなります。

以下は、ユーザーが取ることができる処置で、データ空間分類の使用を増加させる可能性のあるものです。

- 十分な大きさの主記憶域を指定する。デフォルトは 4 メガバイトで、データ空間分類用に推奨する最小値です。指定された主記憶域の大きさを増やすと、より大きなデータ空間の分類が可能になります (特に、大量のデータ (数百メガバイト) を分類する場合)。12 メガバイトより大きな値を指定すると、データ空間の分類を使用するための DFSORT の決定に大きな影響を与えることはありませんが、大量の非データ空間の分類アプリケーションのパフォーマンスを向上させます。
- 作業データ・セット (特に 2 次エクステンツ) に十分な大きさのエクステンツ・サイズを指定する。データ空間分類は、DASD ワークスペースとともに頻繁に使用されますが、ハイパー空間やテープ作業データ・セットとは一緒に使用されません。
- DSPSIZE=MAX を指定する。
- 単一のアドレス空間が作成できるデータ空間のサイズに、IEFUSI が制限を加えないようにする。
- DFSORT が入力ファイルのサイズに関する正確な情報をもっていることを確認する。DFSORT は自動的に、DASD 入力データ・セットおよび DFSMSrmm または ICETPEX を使用するテープ管理システム管理のテープ・データ・セットのファイル・サイズを見積もることができます。DFSORT が自動的にファイル・サイズを見積もることができないケースとその対処方法について、690 ページの『ファイル・サイズおよび動的割り振り』を参照してください。

## IEBGENER に代わる、ICEGENER の使用

DFSORT の ICEGENER 機能を使用することにより、IEBGENER システム・ユーティリティーを使用するために設定されたアプリケーションをより効果的に処理できます。選択された IEBGENER ジョブは、それと同等の (しかし、同じではない) より効率的な DFSORT コピー機能により処理されます。なんらかの理由により、DFSORT のコピー機能が使用できない場合 (たとえば、IEBGENER 制御ステートメントが指定されている場合) は、制御は自動的に IEBGENER システム・ユーティリティーに移されます。

ICEGENER は、IEBGENER と同様、適宜、ユーザーが指定した SDB=value パラメーターを使用します。SDB=value パラメーターが指定されないと、ICEGENER は必要に応じて、SDB にユーザー・サイトの DFSORT インストール・デフォルトを使用します (IBM 提供のデフォルトは SDB=INPUT です)。ICEGENER が制御を IEBGENER へ移す場合、IEBGENER は、もしあれば、ユーザーが提供した SDB=value パラメーターを使用するか、SDB の通常のデフォルトを使用します。

ICEGENER は、DFSPARM または SORTCNTL ステートメント・エラー、あるいは DFSORT により検出された他のエラーのため、IEBGENER に制御を移動することが

## IEBGENER に代わる、ICEGENER の使用

あります。したがって、望ましくない IEBGENER 処理を避けるために、IEBGENER が使用できないアプリケーションには、ICEGENER を使用しないことをお勧めします。たとえば、DFSPARM の INCLUDE ステートメントで ICEGENER を使用した場合、IEBGENER は使用され、正常に完了しますが、INCLUDE ステートメントは無視されます。代わりに、IEBGENER が呼び出されないように、直接 DFSORT のコピーを使用する必要があります。

ただし、ICEGENER が DFSORT のコピーを使用すると確認できた場合、ICEGENER で DFSPARM データ・セットを使用して、DFSORT に制御ステートメントとパラメーターを送ることができます。たとえば、次のように指定すると仮定します。

```
//DFSPARM DD *  
    OPTION SPANINC=RC0  
/*
```

ここで ICEGENER が DFSORT のコピーを使用する場合、可変スパン入力データ・セットで DFSORT が検出する不完全なスパン・レコードは、除去されます。

ユーザー・システムに IEBGENER の名前呼び出される ICEGENER がインストールされている場合は、ICEGENER を使用するためにユーザーのアプリケーションに一切変更を加える必要はありません。ICEGENER の自動使用を選択していない場合は、ユーザーが選択する任意のアプリケーションで、EXEC ステートメント (DFSORT が直接呼び出される場合) または LINK マクロ (DFSORT がプログラムに呼び出される場合) で IEBGENER の代わりに ICEGENER という名前を使用することにより、ICEGENER を使用できます。プログラムにより呼び出されるアプリケーションは再コンパイルする必要があります。

次の例は、EXEC ステートメントで名前 IEBGENER の代わりに名前 ICEGENER を用いることにより、IEBGENER アプリケーションを変更して ICEGENER を使用できるようにする方法を示しています。

```
//GENER JOB...  
// EXEC PGM=ICEGENER  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=CONTROL.MASTER,DISP=OLD,UNIT=3380,VOL=SER=MASTER  
//SYSUT2 DD DSN=CONTROL.BACKUP,DISP=OLD,UNIT=3380,VOL=SER=BACKUP  
//SYSIN DD DUMMY
```

IEBGENER DD ステートメント SYSUT1 (入力用)、SYSUT2 (出力用)、および SYSPRINT (メッセージ用) が、DFSORT により、それぞれ SORTIN、SORTOUT、および SYSOUT に使用されます。これらの DD ステートメント名は、コピー機能呼び出すために、拡張パラメーター・リストを使用して変換されます。DFSORT を使用できない場合 (たとえば、IEBGENER 制御ステートメントが指定されているため) は、制御が IEBGENER へ移されます。

### 注:

1. SYSUT2 データ・セットは、SYSUT1 データ・セットと異なる必要があります。同じである場合はデータが失われたり、誤ったデータになったり、予期せぬ結果を招くことがあります。
2. ICEGENER がプログラムから呼び出されるかどうかにかかわらず、DFSORT は、拡張パラメーター・リストを使用して ICEGENER から呼び出されます。したがって、プログラム呼び出し環境 (つまり、ICEAM2 または ICEAM4、あるいは

## IEBGENER に代わる、ICEGENER の使用

は ICEAM2 または ICEAM4 環境に活動化された ICETDx モジュール) のインストール・オプションが適用され、SORTCNTL または DFSPARM を使用して、コピー・アプリケーションのための追加の制御ステートメント (たとえば、OPTION) を提供できるようになります。しかし、ICEGENER は、DFSPARM または SORTCNTL ステートメント・エラーあるいは DFSORT により検出された他のエラーのため、IEBGENER に制御を移動することがあります。したがって、INCLUDE、OUTREC、SUM などの DFSORT 処理ステートメントが必要な場合は、ICEGENER よりむしろ直接に DFSORT コピーを使用すべきです。

3. DFSORT コピーを使用できなくなるようなエラー条件が生じると、多くの場合、制御が IEBGENER システム・ユーティリティへ移されます。DFSORT メッセージは、SORTDIAG DD ステートメントが指定されていないと印刷されません。SORTDIAG DD ステートメントを使用すると、DFSORT コピーを使用できない理由を判別できます。
4. DFSORT コピーを使用する場合は、その操作とメッセージは、直接呼び出された DFSORT コピー・アプリケーションと同等のものになります。リカバリー不能のエラー (たとえば、入出力エラー) が検出された場合は、DFSORT の戻りコード 16 は ICEGENER により戻りコード 12 に変更され、障害のある IEBGENER アプリケーションからの戻りコードをエミュレートします。
5. DFSORT コピーは、IEBGENER では提供されていないいくつかの機能 (たとえば、特定の埋め込みや切り捨て操作など) を実行できます。ICEGENER 処理が、すべての場合において IEBGENER 処理と同じとは限りません。パフォーマンスを向上させるために、DFSORT コピーは、IEBGENER では使用されない方法 (たとえば、EXCP) を使用します。
6. SYSUT2 LRECL が SYSUT1 LRECL と違う場合、場合によりは、IEBGENER は終了します。ICEGENER は、ICEMAC オプション GNPAD (LRECL 埋め込み) または GNTRUNC (LRECL 切り捨て) のうちのどちらかの該当するオプションに従って、3 つの処置のうちの 1 つを実行します。

SYSUT2 LRECL が SYSUT1 LRECL より大きいときに、制御を ICEGENER から IEBGENER に移したい場合は、ICEMAC オプション GNPAD=IEB を使用します。ICEGENER で LRECL 埋め込みを処理したい場合は、GNPAD=RC0 (提供されているデフォルト) または GNPAD=RC4 を使用します。

SYSUT2 LRECL が SYSUT1 LRECL より小さいときに、制御を ICEGENER から IEBGENER に移したい場合は、ICEMAC オプション GNTRUNC=IEB を使用します。ICEGENER で LRECL の切り捨てを処理したい場合は、GNTRUNC=RC0 (提供されているデフォルト) または GNTRUNC=RC4 を使用します。

7. ICEGENER または ICEGENER のエイリアスとしての IEBGENER の呼び出しについて、レジスター 1 は、以下の 3 つのアドレスで構成される有効なパラメーター・リストを指定する必要があります。
  - アドレス 1: オプション・リストのアドレス。
  - アドレス 2: 代替 DD 名リストのアドレス。
  - アドレス 3: ページ番号リストのアドレス。

有効なパラメーター・リストを作成する ICEGENER を呼び出す方法で、ICEGENER は DFSORT のコピー機能を使用できます。一方、無効なパラメー

## IEBGENER に代わる、ICEGENER の使用

```
|
|           ター・リストを作成する ICEGENER を呼び出す方法では、ICEGENER は
|           IEBGENER への制御権移動を行います。たとえば、下記のようにします。
|
|           call *(icegener)
|
|           TSO コマンド行に上記を指定すると、有効なパラメーター・リストが作成され
|           ます。
|
|           icgener
|
|           TSO コマンド行に上記を指定すると、無効なパラメーター・リストが作成され
|           ます。
```

## ICEGENER の戻りコード

ICEGENER は、IEBGENER または DFSORT コピー機能のいずれかを使用できます。ただし、オペレーティング・システムがサポートされていないために正常に完了しない場合は、ICEGENER は、IEBGENER または DFSORT コピーを使用しないで、戻りコード 24 をオペレーティング・システムまたは呼び出しプログラムに返します。

ICEGENER が制御を IEBGENER へ移す場合、IEBGENER はその戻りコードをオペレーティング・システムまたは呼び出しプログラムに返します。

ICEGENER が DFSORT コピー機能を使用する場合は、次のようになります。

- 正常終了の場合、ICEGENER は、戻りコード 0 または 4 をオペレーティング・システムまたは呼び出しプログラムに戻します。
- NOABEND 指定が有効な異常終了の場合、ICEGENER は戻りコード 12 (16 から変更されたもの) をオペレーティング・システムまたは呼び出しプログラムに戻します。
- ABEND 指定が有効な異常終了の場合、DFSORT は、ICEMAC オプション ABCODE で指定されている適切なコード (エラー・メッセージ番号または 1 から 99 までの数値) を用いてユーザー・abendを出します。

ICEGENER が (レジスター 15 で) 戻す戻りコードの意味は、次のとおりです。

- 0** 正常終了。ICEGENER は正常に終了しました。
- 4** 正常終了。ICEGENER は正常に終了しました。さらに、
  - ICEMAC オプション GNPAD=RC4 が指定されており、かつ SYSUT2 LRECL が SYSUT1 LRECL (LRECL 埋め込み) よりも大きかったか、
  - ICEMAC オプション GNTRUNC=RC4 が指定されており、かつ SYSUT2 LRECL が SYSUT1 LRECL (LRECL 切り捨て) よりも小さかった。
  - SPANINC=RC4 が有効であり、1 つ以上の不完全スパン・レコードが検出された。
- 12** 異常終了。DFSORT は、ICEGENER の正常終了を妨げたエラーを検出しました。
- 24** サポートされていないオペレーティング・システム。このオペレーティング・システムは、このリリースの DFSORT によりサポートされていません。



## SAS システム用の DFSORT のパフォーマンス・ブースターの使用

DFSORT は、SAS アプリケーションに対して、大幅な CPU 時間の向上を提供します。この機能を利用するには、SAS Institute Inc. に連絡をとり、この機能を拡張するサポートについて詳細をお問い合わせください。

---

## DFSORT の BLDINDEX サポートの使用

DFSORT では、IDCAMS BLDINDEX が自動的に DFSORT を使用して、BLDINDEX 外部分類を必要とするほとんどの BLDINDEX ジョブのパフォーマンスを向上できるようなサポートを提供します。



## 第 10 章 DFSORT ジョブ・ストリームの例

例の要約 . . . . .	645
記憶管理責任者の例 . . . . .	646
REXX の例 . . . . .	647
CLIST の例 . . . . .	647
分類処理の例 . . . . .	648
例 1. ALTSEQ を使用した分類処理 . . . . .	648
例 2. OMIT、SUM、OUTREC、DYNALLOC および ZDPRINT を使用した 分類処理 . . . . .	650
例 3. ISCI/ASCII 磁気テープ装置を使用した分類処理 . . . . .	652
例 4. E15、E35、FILSZ、AVGRLLEN、および DYNALLOC を使用した分類 処理 . . . . .	654
例 5. SORTCNTL、CHALT、DYNALLOC、および FILSZ を使用した呼び出 し済み分類処理 . . . . .	656
例 6. VSAM 入力 / 出力、DFSPARM、およびオプション指定変更を使用し た分類処理 . . . . .	658
例 7. COBOL E15、EXEC PARM、COBEXIT、および MSGDDN を使用し た分類処理 . . . . .	660
例 8. 出口のダイナミック・リンク編集を使用した分類処理 . . . . .	662
例 9. 拡張パラメーター・リスト・インターフェースを使用した分類処理	664
例 10. OUTFIL を使用した分類処理 . . . . .	667
例 11. SmartBatch パイプおよび OUTFIL SPLIT を使用した分類処理 . . .	669
例 12. INCLUDE および LOCALE を使用した分類処理 . . . . .	670
例 13. HFS ファイルでの分類 . . . . .	671
組み合わせの例 . . . . .	673
例 1. EQUALS を使用した組み合わせ . . . . .	673
例 2. LOCALE および OUTFIL を使用した組み合わせ処理 . . . . .	674
コピー処理の例 . . . . .	676
例 1. EXEC PARMS、SKIPREC、MSGPRT および ABEND を使用したコピ ー処理 . . . . .	677
例 2. INCLUDE および VLSHRT を使用したコピー処理 . . . . .	679
ICEGENER の例 . . . . .	681
ICETOOL の例 . . . . .	683

### 例の要約

下記の表は、本章で示されている例を要約したものです。

アプリケーション	番号	入力	出力	機能 / オプション
分類	1	DASD	テープ	ALTSEQ
分類	2	DASD	DASD	OMIT、SUM、OUTREC、DYNALLOC、ZDPRINT
分類	3	テープ	テープ	ISCI/ASCII 磁気テープ装置
分類	4	テープ	DASD	E15、E35、FILSZ、AVGRLLEN、DYNALLOC
分類	5	DASD	DASD	プログラムにより呼び出される SORTCNTL、CHALT、DYNALLOC、FILSZ
分類	6	DASD	DASD	VSAM 入力 / 出力、DFSPARM、オプション指定 変更

## 例の要約

アプリケーション	番号	入力	出力	機能 / オプション
分類	7	DASD	DASD	COBOL E15、EXEC PARM、COBEXIT、MSGDDN
分類	8	DASD	DASD	出口のダイナミック・リンク編集
分類	9	E15	DASD	拡張パラメーター・リスト・インターフェース
分類	10	DASD	DASD および SYSOUT	OUTFIL
分類	11	パイプ	パイプ	SmartBatch パイプ、OUTFIL SPLIT、FILSZ、 DYNALLOC
分類	12	DASD	DASD	INCLUDE、LOCALE
分類	13	HFS ファイル	HFS ファイル	
組み合わせ	1	DASD	DASD	EQUALS
組み合わせ	2	DASD	DASD	LOCALE、OUTFIL
コピー	1	テープ	DASD	EXEC PARM、SKIPREC、MSGPRT、ABEND
コピー	2	DASD	DASD	INCLUDE、VLSHRT
ICEGENER	1	DASD	DASD	
ICETOOL	1	DASD	DASD	OCCUR、COPY、SORT、MODE、VERIFY、STATS、 DISPLAY

## 記憶管理責任者の例

DFSORT は、一組のサンプル・ジョブを提供していて、これらのサンプル・ジョブは、記憶管理責任者や、DFSMSHsm、DFSMSrmm、DCOLLECT、および SMF から集められたデータを分析する担当者が参考にするのできる技法を示します。これらのサンプル・ジョブは、SICESAMP ライブラリーの ICESTGEX メンバーにあります (詳細については、システム・プログラマーに連絡してください)。また、これらのサンプル・ジョブを DFSORT FTP サイトからダウンロードすることもできます。これらのサンプル・ジョブは、DFSORT 機能 (たとえば、ICETOOL および OUTFIL) を使用してデータを分析し、報告書を作成する多くの方法を示しています。

### DCOLEX1

DCOLLECT 例 1: VSAM 報告書

### DCOLEX2

DCOLLECT 例 2: 移行報告書

### DCOLEX3

DCOLLECT 例 3: 分析および報告書の容量計画

### DFHSMEX1

DFHSM 例 1: 活動記録ログの暗号解読

### DFHSMEX2

DFHSM 例 2: 壊れた索引をもつ DFHSM CDS のリカバリー

### RMMEEX1

DFSMSrmm 例 1: SMF 監査報告書

### RMMEEX2

DFSMSrmm 例 2: ADDVOLUME コマンドの作成

## REXX の例

DFSORT と ICETOOL を REXX から呼び出すことができます。キーは、必要なデータ・セットに ALLOCATE ステートメントを指定することです。それから、次のように ADDRESS ステートメントを使用します。

```
ADDRESS LINKMVS name
```

これは、標準システム検索リストを使用して、名前付きプログラムを取り出すことです。

下記は DFSORT を呼び出す REXX CLIST の例です。

```
/* Simple REXX CLIST to call DFSORT */
"FREE FI(SYSOUT SORTIN SORTOUT SYSIN)"
"ALLOC FI(SYSOUT) DA(*)"
"ALLOC FI(SORTIN) DA('Y897797.INS1') REUSE"
"ALLOC FI(SORTOUT) DA('Y897797.OUTS1') REUSE"
"ALLOC FI(SYSIN) DA('Y897797.SORT.STMTS') SHR REUSE"
ADDRESS LINKMVS ICEMAN
```

下記は Y897797.SORT.STMTS データ・セットに出力される、DFSORT 制御ステートメントです。

```
SORT FIELDS=(5,4,CH,A)
INCLUDE COND=(21,3,SS,EQ,C'L92,J82,M72')
```

下記は ICETOOL を呼び出す、REXX CLIST の例です。

```
/* Simple REXX CLIST to call ICETOOL */
"FREE FI(TOOLMSG DFSMSG VLR LENDIST TOOLIN)"
"ALLOC FI(TOOLMSG) DA(*)"
"ALLOC FI(DFSMSG) DUMMY"
"ALLOC FI(VLR) DA('Y897797.VARIN') REUSE"
"ALLOC FI(LENDIST) DA(*)"
"ALLOC FI(TOOLIN) DA('Y897797.TOOLIN.STMTS') SHR REUSE"
ADDRESS LINKMVS ICETOOL
```

下記は Y897797.TOOLIN.STMTS データ・セットに出力される、ICETOOL ステートメントです。

```
OCCURS FROM(VLR) LIST(LENDIST) -
TITLE('LENGTH DISTRIBUTION REPORT') BLANK -
HEADER('LENGTH') HEADER('NUMBER OF RECORDS') -
ON(VLEN) ON(VALCNT)
```

## CLIST の例

DFSORT と ICETOOL を CLIST から呼び出すことができます。キーは、必要なデータ・セットに ALLOCATE ステートメントを指定することです。それから、次のように CALL ステートメントを使用します。

```
CALL *(name)
```

下記は DFSORT を呼び出す CLIST の例です。

```
FREE FI(SYSOUT SORTIN SORTOUT SYSIN)
ALLOC FI(SYSOUT) DA(*)
ALLOC FI(SORTIN) DA('Y897797.INS1') REUSE
ALLOC FI(SORTOUT) DA('Y897797.OUTS1') REUSE
ALLOC FI(SYSIN) DA('Y897797.SORT.STMTS') SHR REUSE
CALL *(ICEMAN)
```

## 例の要約

下記は Y897797.SORT.STMTS データ・セットに出力される、DFSORT 制御ステートメントです。

```
SORT FIELDS=(5,4,CH,A)
INCLUDE COND=(21,3,SS,EQ,C'L92,J82,M72')
```

下記は ICETOOL を呼び出す、CLIST の例です。

```
FREE FI(TOOLMSG DFSMSG VLR LENDIST TOOLIN)
ALLOC FI(TOOLMSG) DA(*)
ALLOC FI(DFSMSG) DUMMY
ALLOC FI(VLR) DA('Y897797.VARIN') REUSE
ALLOC FI(LENDIST) DA(*)
ALLOC FI(TOOLIN) DA('Y897797.TOOLIN.STMTS') SHR REUSE
CALL *(ICETOOL)
```

下記は Y897797.TOOLIN.STMTS データ・セットに出力される、ICETOOL ステートメントです。

```
OCCURS FROM(VLR) LIST(LENDIST) -
  TITLE('LENGTH DISTRIBUTION REPORT') BLANK -
  HEADER('LENGTH') HEADER('NUMBER OF RECORDS') -
  ON(VLEN) ON(VALCNT)
```

---

## 分類処理の例

このセクションには、13 の分類処理の例を記載しています。

### 例 1. ALTSEQ を使用した分類処理

<b>INPUT</b>	DASD 上のブロック化可変長レコード	
<b>OUTPUT</b>	3490 上のブロック化可変長レコード	
<b>WORK DATA SETS</b>	2 つの 3390 データ・セット	
<b>USER EXITS</b>	なし	
<b>FUNCTIONS/OPTIONS</b>	ALTSEQ	
//EXAMP	JOB A400,PROGRAMMER	01
//S1	EXEC PGM=SORT	02
//SYSOUT	DD SYSOUT=A	03
//SORTIN	DD DSN=A123456.IN5,DISP=SHR	04
//SORTOUT	DD DSN=OUT1,UNIT=3490,DISP=(,KEEP),VOL=SER=VOL001	05
//SORTWK01	DD UNIT=3390,SPACE=(CYL,(10,10))	06
//SORTWK02	DD UNIT=3390,SPACE=(CYL,(10,10))	07
//SYSIN	DD *	08
*	COLLATE \$, # and @ AFTER Z	09
	SORT FIELDS=(7,5,AQ,A)	10
	ALTSEQ CODE=(5BEA,7BEB,7CEC)	11

行 説明

01 JOB ステートメント。このジョブをオペレーティング・システムに知らせます。

- 02 EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
- 03 SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス A に出力します。
- 04 SORTIN DD ステートメント。入力データ・セットは A123456.IN5 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから、RECFM が VB、最大の LRECL が 120、および BLKSIZE が 2200 であることを判別します。
- 05 SORTOUT DD ステートメント。出力データ・セットは OUT1 という名前をもち、3490 のボリューム VOL001 に割り振られ保存されます。DFSORT は、SORTIN からの RECFM および LRECL をセットし、この標準のラベル付きテープに適切な BLKSIZE を選択します。
- 06 SORTWK01 DD ステートメント。最初の作業データ・セットは 3390 に割り振られます。
- 07 SORTWK02 DD ステートメント。2 番目の作業データ・セットは 3390 に割り振られます。
- 08 SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 09 注釈ステートメント。印刷されますが、それ以外は無視されます。
- 10 SORT ステートメント。FIELDS は、7 桁目から始まる昇順の 5 バイトの文字制御フィールドを指定します (RDW が最初の 4 バイトを占めるため、これは 3 番目のデータ・バイトになります)。制御フィールドは、ALTSEQ ステートメントで記述された変更済みの順序に従って照合されます。
- 11 ALTSEQ ステートメント。CODE は、\$、#、および @ の 3 文字が、Z の後にこの順序で照合されることを指定します。

## 例 2. OMIT、SUM、OUTREC、DYNALLOC および ZDPRINT を使用した分類処理

<b>INPUT</b>	3380 および 3390 上のブロック化固定長レコード
<b>OUTPUT</b>	3390 上のブロック化固定長レコード
<b>WORK DATA SETS</b>	動的に割り振られる
<b>USER EXITS</b>	なし
<b>FUNCTIONS/OPTIONS</b>	OMIT、OUTREC、SUM、DYNALLOC、ZDPRINT
//EXAMP	JOB A400,PROGRAMMER 01
//STEP1	EXEC PGM=SORT 02
//SYSOUT	DD SYSOUT=H 03
//SORTIN	DD DSN=INP1,DISP=SHR,UNIT=3380,VOL=SER=SCR001 04
//	DD DSN=INP2,DISP=SHR,UNIT=3390,VOL=SER=SYS351 05
//SORTOUT	DD DSN=&&OUTPUT,DISP=(,PASS),UNIT=3390, 06
//	SPACE=(CYL,(5,1)),DCB=(LRECL=22) 07
//SYSIN	DD * 08
	OMIT COND=(5,1,CH,EQ,C'M') 09
	SORT FIELDS=(20,8,CH,A,10,3,FI,D) 10
	SUM FIELDS=(16,4,ZD) 11
	OPTION DYNALLOC,ZDPRINT 12
	OUTREC FIELDS=(10,3,20,8,16,4,2Z,5,1,C' SUM') 13

- |   |    |
|---|----|
| 行 | 説明 |
|---|----|
- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
- 03** SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス H に出力します。
- 04 ~ 05**  
 SORTIN DD ステートメント。2 つのデータ・セットの連結で構成されます。最初の入力データ・セットは INP1 という名前をもち、3380 ボリューム SCR001 に置かれます。2 番目の入力データ・セットは INP2 という名前をもち、3390 ボリューム SYS351 に置かれます。DFSORT は、データ・セット・ラベルから、レコード形式が FB、LRECL が 80、および最大 BLKSIZE が 27920 であることを判別します。
- 06 ~ 07**  
 SORTOUT DD ステートメント。出力データ・セットは一時的なもので、3390 に割り振られます。OUTREC ステートメントにより再フォーマット設定された出力レコード長は 22 バイトになるため、LRECL=22 を指定します。DFSORT は、SORTIN から RECFM をセットし、適切な BLKSIZE を選択します。
- 08** SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。



- 09** OMIT ステートメント。COND は、文字 M を 5 桁目にもつ入力レコードを出力データ・セットから除外することを指定します。
- 10** SORT ステートメント。FIELDS は、20 桁目から始まる昇順の 8 バイトの文字制御フィールド、および 10 桁目から始まる降順の 3 バイトの固定小数点制御フィールドを指定します。
- 11** SUM ステートメント。FIELDS は、16 桁目から始まる 4 バイトのゾーン 10 進数の合計フィールドを指定します。同一の制御フィールド (SORT ステートメントで指定) をもつ 2 つのレコードが見つかった場合は、必ずその合計フィールド (SUM ステートメントで指定) が合計され、1 つのレコードに入れられ、もう 1 つのレコードは削除されます。
- 12** OPTION ステートメント。DYNALLOC は、装置タイプおよび装置数のインストール・デフォルトを使用して、作業データ・セットが動的に割り振られることを指定します。ZDPRINT は、正の ZD SUM フィールドが印刷可能であることを指定します。
- 13** OUTREC ステートメント。FIELDS は、レコードを出力用に再フォーマット設定する方法を指定します。再フォーマット設定された後のレコードは 22 バイトの長さで、次のようになります。

位置	内容
<b>1 ~ 3</b>	入力位置 10 ~ 12
<b>4 ~ 11</b>	入力位置 20 ~ 27
<b>12 ~ 15</b>	入力位置 16 ~ 19
<b>16 ~ 17</b>	ゼロ
<b>18</b>	入力位置 5
<b>19 ~ 22</b>	文字ストリング 'SUM'

## 例 3. ISCI/ASCII 磁気テープ装置を使用した分類処理

```

INPUT  3590 上の可変長 ISCI/ASCII レコード
OUTPUT
      3590 上の可変長 ISCI/ASCII レコード
WORK DATA SETS
      1 つの SYSDA データ・セット
USER EXITS
      なし
FUNCTIONS/OPTIONS
      なし

//EXAMP   JOB   A400,PROGRAMMER                01
//RUNIT   EXEC SORTD                          02
//SORTIN  DD   DSN=SRTFIL,DISP=(OLD,DELETE),UNIT=3590,      03
//   DCB=(RECFM=D,LRECL=400,BLKSIZE=404,OPTCD=Q,BUFOFF=L),  04
//   VOL=SER=311500,LABEL=(1,AL)                05
//SORTOUT DD   DSN=OUTFIL,UNIT=3590,LABEL=(,AL),DISP=(,KEEP),  06
//   DCB=(BLKSIZE=404,OPTCD=Q,BUFOFF=L),VOL=SER=311501      07
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4))        08
//SYSIN   DD   *                                  09
          SORT FIELDS=(10,8,AC,D)                10
          RECORD TYPE=D,LENGTH=(,,20,80)         11

```

## 行 説明

- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。SORTD カタログ式プロシーチャーを使用して、DFSORT を直接呼び出します。
- 03 ~ 05**  
SORTIN DD ステートメント。入力データ・セットは SRTFIL という名前をもち、3590 ボリューム 311500 に置かれます。これは、このジョブ・ステップの後で削除されます。これは D の RECFM (可変長 ISCI/ASCII レコード)、400 の最大 LRECL、404 の BLKSIZE、および ISCI/ASCII ラベルをもっています。このジョブの場合、バッファ・オフセットはブロック長標識です。レコードは ISCI/ASCII から EBCDIC に変換されます。
- 06 ~ 07**  
SORTOUT DD ステートメント。出力データ・セットは OUTFIL という名前をもち、3590 のボリューム 311501 に割り振られ保存されます。これは ISCI/ASCII のラベルを付けて書き込まれます。DFSORT は、SORTIN からの RECFM および LRECL をセットし、DD ステートメントに指示されているように、BLKSIZE を 404 にセットします。このジョブの場合、バッファ・オフセットはブロック長標識です。このレコードは、EBCDIC から ISCI/ASCII に変換されます。
- 08** SORTWK01 DD ステートメント。作業データ・セットは SYSDA に割り振られます。
- 09** SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。

- 10 SORT ステートメント。FIELDS は、10 桁目から始まる降順の 8 バイトの ISCI/ASCII 制御フィールドを指定します。
- 11 RECORD ステートメント。TYPE は、ISCI/ASCII 可変長レコードを指定します。LENGTH は、最小レコード長が 20 で、平均レコード長が 80 であることを指定します。

## 例 4. E15、E35、FILSZ、AVGRLen、および DYNALLOC を使用した分類処理

<b>INPUT</b>	3490 の可変長レコード	
<b>OUTPUT</b>	SYSDA 上のブロック化可変長レコード	
<b>WORK DATA SETS</b>	動的に割り振られる	
<b>USER EXITS</b>	E15 および E35	
<b>FUNCTIONS/OPTIONS</b>	FILSZ、AVGRLen、DYNALLOC	
//EXAMP	JOB A400,PROGRAMMER	01
//STEP1	EXEC PGM=ICEMAN	02
//SYSOUT	DD SYSOUT=A	03
//SORTIN	DD DSN=INPUT,VOL=SER=FLY123,	04
//	UNIT=3490,DISP=OLD	05
//SORTOUT	DD DSN=&&OUT,DISP=(,PASS),SPACE=(CYL,(10,12)),	06
//	UNIT=SYSDA,DCB=(RECFM=VB)	07
//MODLIB	DD DSN=EXIT1.RTNS,DISP=SHR	08
//	DD DSN=EXIT2.RTNS,DISP=SHR	09
//SYSIN	DD *	10
	SORT FIELDS=(23,4,PD,A,10,6,FS,A)	11
	OPTION DYNALLOC=(3390,3),AVGRLen=75,FILSZ=E50000	12
	MODS E15=(MODREC,1024,MODLIB),E35=(ADDRAC,1200,MODLIB)	13

- |         |  |
|---------|--|
| 行       | 説明   |
| 01      | JOB ステートメント。このジョブをオペレーティング・システムに知らせます。   |
| 02      | EXEC ステートメント。DFSORT を直接呼び出します。   |
| 03      | SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス A に出力します。   |
| 04 ~ 05 | SORTIN DD ステートメント。入力データ・セットは INPUT という名前をもち、3490 ボリューム FLY123 に置かれます。DFSORT は、この標準のラベル付きテープのデータ・セット・ラベルから、RECFM が V、LRECL が 120、および BLKSIZE が 124 であることを判別します。 |
| 06 ~ 07 | SORTOUT DD ステートメント。出力データ・セットは一時的なもので、SYSDA に割り振られます。入力がブロック化されていなくて、出力がブロック化されるため、RECFM=VB を指定する必要があります。DFSORT は SORTIN から LRECL をセットし、適切な BLKSIZE を選択します。     |
| 08 ~ 09 | MODLIB DD ステートメント。出口ルーチンを含むロード・ライブラリー  |

を指定します。出口ルーチンが複数のライブラリーにある場合、それらのライブラリーは単一の DD ステートメントを用いて連結する必要があります。

- 10 SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 11 SORT ステートメント。FIELDS は、23 桁目から始まる昇順の 4 バイトのパック 10 進数の制御フィールドと、10 桁目から始まる昇順の 6 バイトの浮動符号制御フィールドを指定します。
- 12 OPTION ステートメント。DYNALOC=(3390,3) は、3 つの 3390 作業データ・セットが割り振られることを指定します。AVGRLLEN=75 は、75 の平均レコード長を指定します。AVGRLLEN は、DFSORT が可変長レコード入力用にワークスペースを最適化するのに役立ちます。FILSZ=E50000 は、50000 レコードの見積もりを指定します。3490 入力データ・セットは圧縮されているため、データ・セットが DFSMSrmm または ICETPEX を使用するテープ管理システムの管理である場合を除き、DFSORT はファイル・サイズを正確に判別できないことがあります。FILSZ の指定により、入力データ・セットが管理されていないときのワークスペースの最適化に著しい違いをもたらすことができます。
- 13 MODS ステートメント。E15 は、MODREC という名前のユーザー出口ルーチンを指定します。MODREC およびそれが実行するシステム・サービス(たとえば、GETMAIN と OPEN)には、約 1024 バイトが必要です。E35 は、ADDREC という名前のユーザー出口ルーチンを指定します。ADDREC およびそれが実行するシステム・サービスには、約 1200 バイトが必要です。MODREC および ADDREC は、MODLIB DD ステートメントで定義された定義に常駐します。

## 例 5. SORTCNTL、CHALT、DYNALLOC、および FILSZ を使用した呼び出し済み分類処理

<b>INPUT</b>	DASD 上のブロック化固定長レコード	
<b>OUTPUT</b>	DASD 上のブロック化固定長レコード	
<b>WORK DATA SETS</b>	動的に割り振られる	
<b>USER EXITS</b>	なし	
<b>FUNCTIONS/OPTIONS</b>	CHALT、DYNALLOC、FILSZ	
//EXAMP	JOB A400,PROGRAMMER	01
//RUNSORT	EXEC PGM=MYPGM	02
//STEPLIB	DD DSN=M999999.LOAD,DISP=SHR	03
//SYSOUT	DD SYSOUT=A	04
//SYSPRINT	DD SYSOUT=A	05
//SORTIN	DD DSN=M999999.INPUT(MASTER),DISP=OLD	06
//SORTOUT	DD DSN=M999999.OUTPUT.FILE,DISP=OLD	07
//SORTCNTL	DD *	08
OPTION	CHALT,DYNALLOC=(,3),FILSZ=U25000	09

行	説明
01	JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
02	EXEC ステートメント。MYPGM という名前のプログラムを呼び出します。これは次に DFSORT を呼び出します。
03	STEPLIB DD ステートメント。MYPGM を含むロード・ライブラリーを指定します。
04	SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス A に出力します。
05	SYSPRINT DD ステートメント。MYPGM 出力をシステム出力クラス A に出力します。
06	SORTIN DD ステートメント。入力データ・セットは、カタログされた区分データ・セット M999999.INPUT のメンバー MASTER です。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
07	SORTOUT DD ステートメント。出力データ・セットは M999999.OUTPUT.FILE という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
08	SORTCNTL DD ステートメント。DFSORT 制御ステートメントが続きます。

す。SORTCNTL 内のステートメントは、使用する DFSORT パラメーター・リスト内の MYPGM により渡されるステートメントを指定変更または補足します。

- 09** **OPTION** ステートメント。**CHALT** は、文字形式制御フィールド (MYPGM により渡される **SORT** ステートメントで指定) が、インストール・デフォルトの **ALTSEQ** テーブルを使用して分類されることを指定します。**DYNALLOC=(,3)** は、装置タイプのインストール・デフォルトを使用して 3 つの作業データ・セットが動的に割り振られることを指定します。**FILSZ=U25000** は、DFSORT が 25000 レコードのファイル・サイズを指定して、必要なワークスペースの大きさを判別することを指定します。入力データ・セットが PDS のメンバーであるため、**FILSZ** を指定すると、DFSORT が作業データ・セットのスペースを最適化するのに役立ちます。

## 例 6. VSAM 入力 / 出力、DFSPARM、およびオプション指定変更を使用した分類処理

<b>INPUT</b>	VSAM TYPE=V レコード	
<b>OUTPUT</b>	VSAM TYPE=V レコード	
<b>WORK DATA SETS</b>	動的に割り振られる	
<b>USER EXITS</b>	なし	
<b>FUNCTIONS/OPTIONS</b>	各種オプションの指定変更	
//EXAMP	JOB A400,PROGRAMMER	01
//S1	EXEC PGM=SORT	02
//SYSOUT	DD SYSOUT=A	03
//SORTIN	DD DSN=TEST.SORTIN.FILE,DISP=SHR	04
//SORTOUT	DD DSN=TEST.SORTOUT.FILE,DISP=SHR	05
//DFSPARM	DD *	06
	RECORD TYPE=V	07
	SORT FIELDS=(30,4,BI,A)	08
	OPTION HIPRMAX=10,DYNALLOC=3390,MAINSIZE=3M,	09
	MSGPRT=CRITICAL,NOLIST	10

この例では、図で示すために、DFSORT バッチ直接呼び出しの標準のインストール・デフォルトをインストール先で変更しないものと仮定します。

行	説明
01	JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
02	EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
03	SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス A に出力します。
04	SORTIN DD ステートメント。入力データ・セットは TEST.SORTIN.FILE です。DFSORT は、これが VSAM データ・セットであると判別し、その属性をカタログから入手します。
05	SORTOUT DD ステートメント。出力データ・セットは TEST.SORTOUT.FILE です。DFSORT は、これが VSAM データ・セットであると判別し、その属性をカタログから入手します。
06	DFSPARM DD ステートメント。DFSORT 制御ステートメントが続きます。DFSPARM は、DFSORT の直接呼び出しとプログラム呼び出しの両方に使用でき、他のすべてのソースからのオプションとステートメントを指定変更します。ある種オペランド (たとえば、MSGPRT や LIST/NOLIST) は、DFSPARM、EXEC PARM、または呼び出しパラメーター・リストで提供される場合は使用されますが、SYSIN や SORTCNTL で提供される場合は使用されません。



- 07** RECORD ステートメント。TYPE=V は、DFSORT が VSAM レコードを可変長として処理することを指定します。この場合、DFSORT は SORTIN および SORTOUT に VSAM データ・セットを使用するために、自動的に V のレコード・タイプをセットするので、RECORD ステートメントは省略できます。
- 08** SORT ステートメント。FIELDS は、30 桁目で始まる昇順の 4 バイトの 2 進数制御フィールドを指定します。この位置 (30 桁目) は、VSAM CLUSTER 用の KEYS(4 25) の指定に対応するものです (オフセット 25 の 4 バイトで、これは DFSORT が VSAM TYPE=V レコードの入力時に提供し、出力時に除去する RDW 用に付加された 4 バイトをもつ 26 桁目と同じです)。
- 09 ~ 10** OPTION ステートメント。HIPRMAX=10 は、最大 10 メガバイトまでのハイパー空間をハイパー分類用にコミットできるように、標準のインストール・デフォルト HIPRMAX=OPTIMAL を指定変更することを指定します。DYNALLOC=3390 は、標準のインストール・デフォルト SYSDA を指定変更し、作業データ・セットが 3390 に割り振られることを指定します。4 つの作業データ・セットの標準インストール・デフォルトは指定変更されません。MAINSIZE=3M は、最高 3 メガバイトの記憶域を使用できるように、標準のインストール・デフォルト MAINSIZE=MAX を指定変更することを指定します。MSGPRT=CRITICAL は、エラー・メッセージだけを印刷できるように、標準のインストール・デフォルト MSGPRT=ALL を指定変更することを指定します。NOLIST は、制御ステートメントを印刷しないように、標準のインストール・デフォルト LIST=YES を指定変更することを指定します。

## 例 7. COBOL E15、EXEC PARM、COBEXIT、および MSGDDN を使用した分類処理

<b>INPUT</b>	DASD 上の固定長レコード	
<b>OUTPUT</b>	SYSDA 上の固定長レコード	
<b>WORK DATA SETS</b>	なし	
<b>USER EXITS</b>	COBOL E15	
<b>FUNCTIONS/OPTIONS</b>	COBEXIT、MSGDDN	
//EXAMP	JOB A400,PROGRAMMER	01
//STEP1	EXEC PGM=SORT,PARM='MSGDDN=DFSOUT'	02
//STEPLIB	DD DSN=SYS1.COBLIB,DISP=SHR	03
//SYSOUT	DD SYSOUT=A	04
//DFSOUT	DD SYSOUT=A	05
//EXITC	DD DSN=COBEXITS.LOADLIB,DISP=SHR	06
//SORTIN	DD DSN=SORT1.IN,DISP=SHR	07
//SORTOUT	DD DSN=&&OUT,DISP=(,PASS),SPACE=(CYL,(5,5)),	08
//	UNIT=SYSDA,DCB=(LRECL=120)	09
//SYSIN	DD *	10
	SORT FIELDS=(5,4,A,22,2,A),FORMAT=BI	11
	MODS E15=(COBOLE15,37000,EXITC,C)	12
	RECORD LENGTH=(,120)	13
	OPTION COBEXIT=COB2	14

行	説明
01	JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
02	EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。MSGDDN=DFSOUT は、SYSOUT 内の COBOL メッセージが DFSORT メッセージおよび制御ステートメントの間にはさまれないようにするために、DFSORT メッセージおよび制御ステートメント用の代替メッセージ・データ・セットを指定します。
03	STEPLIB ステートメント。VS COBOL II ライブラリーまたは言語環境プログラムを適宜指定します。
04	SYSOUT ステートメント。COBOL メッセージをシステム出力クラス A に出力します。
05	DFSOUT ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス A に出力します (これは EXEC ステートメントの PARM フィールドで MSGDDN により指定された代替メッセージ・データ・セットです)。
06	EXITC ステートメント。出口ルーチンを含むロード・ライブラリーを指定します。

- 07** SORTIN DD ステートメント。入力データ・セットは SORT1.IN という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから、RECFM が F、LRECL が 100、および BLKSIZE が 100 であると判別します。
- 08 ~ 09** SORTOUT DD ステートメント。出力データ・セットは一時的なもので、SYSDA に割り振られます。E15 出口は、レコードの長さを 100 バイトから 120 バイトへ変更するので、LRECL=120 を指定する必要があります。DFSORT は、SORTIN からの RECFM をセットし、BLKSIZE を LRECL (非ブロック化レコード) にセットします。
- 10** SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 11** SORT ステートメント。FIELDS は、5 桁目から始まる昇順の 4 バイトの制御フィールドと、22 桁目から始まる昇順の 2 バイトの制御フィールドを指定します。FORMAT は、制御フィールドが 2 進数であることを指定します。
- 12** MODS ステートメント。E15 は、COBOL で作成された、COBOLE15 という名前のユーザー出口ルーチンを指定します。この出口とそれが実行するシステム・サービス (たとえば、GETMAIN および OPEN)、および COBOL ライブラリー・サブルーチン用に、約 37000 バイトが必要です。COBOLE15 は、EXITC DD ステートメントで定義させたライブラリーに常駐します。
- 13** RECORD ステートメント。LENGTH は、COBOL E15 ルーチンがレコードの長さを 120 バイトに変更することを指定します。
- 14** OPTION ステートメント。COBEXIT=COB2 は、COBOL E15 ルーチンが VS COBOL II ライブラリーまたは言語環境プログラムを使用して実行されることを適宜指定します。

## 例 8. 出口のダイナミック・リンク編集を使用した分類処理

```

INPUT  DASD 上のブロック化固定長レコード

OUTPUT
          3380 上のブロック化固定長レコード

WORK DATA SETS
          1 つの SYSDA データ・セット

USER EXITS
          E11、E15、E17、E18、E19、E31、E35、E38、E39

FUNCTIONS/OPTIONS
          なし

//EXAMP   JOB  A400,PROGRAMMER                01
//STEPA   EXEC SORT                          02
//SORTIN  DD  DSN=SMITH.INPUT,DISP=SHR        03
//SORTOUT DD  DSN=SMITH.OUTPUT,DISP=(NEW,CATLG), 04
// UNIT=3380,SPACE=(TRK,(10,2)),VOL=SER=XYZ003 05
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1,1))     06
//EXIT    DD  DSN=SMITH.EXIT.OBJ,DISP=SHR     07
//EXIT2   DD  DSN=SMITH.EXIT2.OBJ,DISP=SHR    08
//SORTMODS DD UNIT=SYSDA,SPACE=(TRK,(10,,3)) 09
//SYSIN   DD  *                               10
          SORT FIELDS=(1,8,CH,A,20,4,BI,D)    11
          MODS E11=(EXIT11,1024,EXIT,S),      12
              E15=(E15,1024,SYSIN,T),        13
              E17=(EXIT17,1024,EXIT2,T),     14
              E18=(EXIT18,1024,EXIT,T),      15
              E19=(E19,1024,SYSIN,T),        16
              E31=(PH3EXIT,1024,EXIT,T),     17
              E35=(PH3EXIT,1024,EXIT,T),     18
              E38=(PH3EXIT,1024,EXIT,T),     19
              E39=(E39,1024,SYSIN,T)        20
          END                                  21
<object deck for E15 exit here>              22
<object deck for E19 exit here>              23
<object deck for E39 exit here>              24

```

## 行 説明

- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。SORT カタログ式プロシージャーを使用して、DFSORT を直接呼び出し、リンケージ・エディターで必要な DD ステートメント（ここでは示されていません）を提供します。
- 03** SORTIN DD ステートメント。入力データ・セットは SMITH.INPUT という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 04 ~ 05** SORTOUT DD ステートメント。出力データ・セットは、SMITH.OUTPUT

という名前をもち、3380 ボリューム XYZ003 に割り振られカタログされています。DFSORT は、SORTIN から RECFM および LRECL をセットし、適切な BLKSIZE を選択します。

- 06 SORTWK01 DD ステートメント。作業データ・セットは SYSDA に割り振られます。
- 07 EXIT DD ステートメント。E11、E18、E31、E35、および E38 出口ルーチン用のオブジェクト・デックを含む区分データ・セットを指定します。
- 08 EXIT2 DD ステートメント。E17 出口ルーチン用のオブジェクト・デックを含む区分データ・セットを指定します。
- 09 SORTMODS DD ステートメント。リンケージ・エディターに対する入力用に SYSIN からの出口ルーチンのオブジェクト・デックを保持するための区分データ・セットを、SYSDA 上に割り振ります。
- 10 SYSIN DD ステートメント。DFSORT 制御ステートメント、およびリンケージ・エディターが使用するオブジェクト・デックが続きます。
- 11 SORT ステートメント。FIELDS は、1 桁目から始まる昇順の 8 バイトの文字制御フィールドと、20 桁目から始まる降順の 4 バイトの 2 進数の制御フィールドを指定します。

#### 12 ~ 20

MODS ステートメント。使用する出口ルーチン、各出口ごとに必要なバイト数の概数、および次のことを指定します。

- EXIT ライブラリー内の EXIT11 ルーチンは、他の入力フェーズ出口ルーチンとは別にリンク・エディットされ、ユーザー出口 E11 に関連付けられます。
- SYSIN 内の E15 および E19 ルーチン、EXIT2 内の EXIT17 ルーチン、および EXIT 内の EXIT18 ルーチンは一緒にリンク・エディットされ、それぞれユーザー出口 E15、E19、E17、および E18 に関連付けられます。
- PH3EXIT オブジェクト・デック内の E31、E35、および E38 ルーチン、および SYSIN 内の E39 ルーチンは一緒にリンク・エディットされ、それぞれユーザー出口 E31、E35、E38、および E39 に関連付けられます。

- 21 END ステートメント。DFSORT 制御ステートメントの終わりと、出口ルーチンのオブジェクト・デックの始まりを示します。

#### 22 ~ 24

オブジェクト・デック。E15、E19、および E39 出口ルーチンの 3 つのオブジェクト・デックが END ステートメントに続きます。

## 例 9. 拡張パラメーター・リスト・インターフェースを使用した分類処理

**INPUT** E15 からの固定長レコード

**OUTPUT**

SYSDA 上のブロック化固定長レコード

**WORK DATA SETS**

動的に割り振られる

**USER EXITS**

E15

**FUNCTIONS/OPTIONS**

OMIT、FILSZ、RESINV、DYNALLOC

```
//EXAMP JOB A400,PROGRAMMER 01
//STEP1 EXEC PGM=MYSORT 02
//SYSOUT DD SYSOUT=C 03
//MSGOUT DD SYSOUT=C 04
//STEPLIB DD DSN=A123456.LOAD,DISP=SHR 05
//SORTOUT DD DSN=&&OUTPUT,DISP=(,PASS),UNIT=SYSDA, 06
// SPACE=(CYL,(8,4)) 07
//SORTCNTL DD * 08
* Update file size estimate 09
OPTION FILSZ=E30000 10
```

```
-----
MYSORT CSECT 11
.
.
.
* LA R1,PL1 SET ADDRESS OF PARAMETER LIST 12
TO BE PASSED TO DFSORT 13
* ST R2,PL4 SET ADDRESS OF GETMAINED AREA 14
TO BE PASSED TO E15 15
LINK EP=SORT INVOKE DFSORT 16
.
.
.
PL1 DC A(CTLST) ADDRESS OF CONTROL STATEMENTS 17
PL2 DC A(E15) ADDRESS OF E15 ROUTINE 18
PL3 DC A(0) NO E35 ROUTINE 19
PL4 DS A USER EXIT ADDRESS CONSTANT 20
PL5 DC F'-1' INDICATE END OF LIST 21
CTLST DS 0H CONTROL STATEMENTS AREA 22
DC AL2(CTL2-CTL1) LENGTH OF CHARACTER STRING 23
CTL1 DC C' SORT FIELDS=(5,8,CSF,A) ' 24
DC C' RECORD TYPE=F,LENGTH=80 ' 25
DC C' OPTION FILSZ=E25000,DYNALLOC, ' 26
DC C'RESINV=8000 ' 27
DC C' OMIT FORMAT=CSF,COND=(5,8,EQ,13,8) ' 28
CTL2 EQU * 29
OUT DCB DDNAME=MSGOUT,... 30
E15 DS 0H E15 ROUTINE 31
.
.
.
L R2,4(,R1) GET ADDRESS OF GETMAINED AREA 32
.
.
.
BR R14 RETURN TO DFSORT 33
.
.
```

プログラム MYSORT を実行するための JCL、および拡張パラメーター・リストを使用して DFSORT を呼び出すために MYSORT が使用するコーディングの重要部分について、下記に示します。この例では、図で示すために、DFSORT のバッチ・プログラム呼び出しの標準インストール・デフォルトを変更しないものと仮定します。

**行**      **説明**

- 01**      JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02**      EXEC ステートメント。MYSORT という名前のプログラムを呼び出します。これは次に DFSORT を呼び出します。
- 03**      SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス C に出力します。
- 04**      MSGOUT DD ステートメント。MYSORT メッセージを SYSOUT クラス C に出力します。
- 05**      STEPLIB DD ステートメント。MYSORT を含むロード・ライブラリーを指定します。
- 06 ~ 07**  
         SORTOUT DD ステートメント。出力データ・セットは一時的なもので、SYSDA に割り振られます。SORTIN は使用されないので、DFSORT は、RECORD ステートメントからの RECFM と LRECL をセットし、BLKSIZE を LRECL (非ブロック化レコード) にセットします。
- 08**      SORTCNTL DD ステートメント。DFSORT 制御ステートメントが続きます。SORTCNTL 内のステートメントは、使用する拡張パラメーター・リスト内の MYSORT により渡されるステートメントを指定変更または補足します。
- 09**      注釈ステートメント。印刷されますが、それ以外は無視されます。
- 10**      OPTION ステートメント。FILSZ=E30000 は、30000 レコードの見積もりを指定し、拡張パラメーター・リストの OPTION ステートメントの FILSZ=E25000 を指定変更します。E15 ルーチンはすべての入力レコードを提供するため、DFSORT は、ファイル・サイズを正確に判別することができません。したがって、FILSZ を指定すると、E15 ルーチンがすべての入力レコードを提供するときに、ワークスペースの最適化に大幅な差が生じることがあります。入力レコードの数が大幅に増える場合は、必ず、FILSZ 値を変更することが重要です。
- 11**      これはプログラム MYSORT の始まりです。作業域を GETMAIN し、そのアドレスをレジスター 2 に保管し、E15 ルーチンが使用する情報で初期設定するものと想定してください。
- 12 ~ 13**  
         MYSORT は、DFSORT に渡される拡張パラメーター・リストのアドレスを、レジスター 1 に入れます。
- 14 ~ 15**  
         MYSORT は、GETMAINed 作業域のアドレスを、拡張パラメーター・リストのユーザー出口アドレス定数フィールドに入れます。DFSORT は、この

## 分類処理の例

アドレスを E15 ルーチンが呼び出されたときに、(E15 パラメーター・リストの 2 番目のワード内に) 渡します。

**16** MYSORT は、DFSORT をその別名 SORT で呼び出します。

### 17 ~ 21

拡張パラメーター・リストは、制御ステートメント区域のアドレス、E15 ルーチンのアドレス (E35 ルーチンは存在しません)、および GETMAINed 作業域のアドレスを指定します。F'-1' は拡張パラメーター・リストの終わりを示します。後続のパラメーター・リスト・フィールド (たとえば、ALTSEQ テーブルのアドレス) は、このアプリケーションでは使用されません。

E15 ルーチンのアドレスはパラメーター・リストに渡されるため、SORTIN を使用できません。SORTIN DD ステートメントがあっても無視されます。

### 22 ~ 23

これは制御ステートメント区域の始まりです。制御ステートメントの全長が指定されます。

**24** SORT ステートメント。FIELDS は、5 桁目から始まる昇順の 8 バイトの浮動符号制御フィールドを指定します。

**25** RECORD ステートメント。TYPE=F および LENGTH=80 は、E15 が 80 バイトの固定長レコードを挿入することを指定します。この場合、DFSORT は自動的に F のレコード・タイプをセットするので、TYPE=F ステートメントは省略できます。ただし、E15 がすべての入力レコードを提供するときは、LENGTH を指定する必要があります。

### 26 ~ 27

OPTION ステートメント。FILSZ=E25000 は、25000 レコードの見積もりを指定します。これは、SORTCNTL の OPTION ステートメントの FILSZ=E30000 で指定変更されます。DYNALLOC は、装置タイプおよび装置数のインストール・デフォルトを使用して、作業データ・セットが動的に割り振られることを指定します。RESINV=8000 は、MYSORT の E15 出口ルーチンが実行するシステム・サービス (たとえば、GETMAIN および OPEN) 用に、約 8000 バイトが必要であると指定します。

**28** OMIT ステートメント。FORMAT は、比較フィールドが浮動符号であることを指定します。COND は、5 桁目と 13 桁目から始まる 8 バイトの浮動符号比較フィールド (制御フィールドも) が等しい入力レコードを出力データ・セットから除外することを指定します。

**29** これは制御ステートメント域の終わりです。

**30** これは MYSORT の MSGOUT 出力用の DCB です。

### 31 ~ 33

これは MYSORT の E15 ルーチンです。E15 ルーチンは、GETMAINed 作業域のアドレスを E15 パラメーター・リストの 2 番目のワードからロードします。E15 ルーチンは、そのアドレスをレジスター 1 に入れ、またレジスター 15 に 12 (挿入) を入れることにより、各入力レコードを提供します。すべてのレコードが渡されると、E15 ルーチンは、レジスター 15 に 8 (『戻り不可』) を指定します。



## 例 10. OUTFIL を使用した分類処理

```

INPUT  固定長レコード・データ・セット
OUTPUT
          複数の固定長レコード・データ・セット
WORK DATA SETS
          動的に割り振られる (デフォルトによる)
USER EXITS
          なし
FUNCTIONS/OPTIONS
          OUTFIL

//EXAMP   JOB   A400,PROGRAMMER                01
//OUTFIL  EXEC  PGM=SORT                        02
//SYSOUT  DD   SYSOUT=A                        03
//SORTIN  DD   DSN=GRP.RECORDS,DISP=SHR        04
//ALLGPS  DD   DSN=GRP.ALLGRPS,DISP=OLD        05
//ALLBU   DD   DSN=GRP.BU,DISP=(NEW,CATLG,DELETE),
//         UNIT=3390,SPACE=(TRK,(10,10))      06
//G1STATS DD   SYSOUT=A                        08
//G2STATS DD   SYSOUT=A                        09
//SYSIN   DD   *                               10
          SORT FIELDS=(6,5,CH,A)              11

          OUTFIL FNAMES=(ALLGPS,ALLBU)        12

          OUTFIL FNAMES=G1STATS,              13
          INCLUDE=(1,3,CH,EQ,C'G01'),        14
          HEADER2=(1:'GROUP 1 STATUS REPORT FOR ',&DATE,
          ' - PAGE ',&PAGE,2/,
          6:'ITEM ',16:'STATUS ',31:'PARTS',/,
          6:'-----',16:'-----',31:'-----'),
          OUTREC=(6:6,5,                      19
          16:14,1,CHANGE=(12,                20
          C'1',C'SHIP',                      21
          C'2',C'HOLD',                      22
          C'3',C'TRANSFER'),                23
          NOMATCH=(C'*CHECK CODE*'),        24
          31:39,1,BI,M10,LENGTH=5,         25
          120:X)                             26

          OUTFIL FNAMES=G2STATS,              27
          INCLUDE=(1,3,CH,EQ,C'G02'),        28
          HEADER2=(1:'GROUP 2 STATUS REPORT FOR ',&DATE,
          ' - PAGE ',&PAGE,2/,              30
          6:'ITEM ',16:'STATUS ',31:'PARTS',/,
          6:'-----',16:'-----',31:'-----'),
          OUTREC=(6:6,5,                      33
          16:14,1,CHANGE=(12,                34
          C'1',C'SHIP',                      35
          C'2',C'HOLD',                      36
          C'3',C'TRANSFER'),                37
          NOMATCH=(C'*CHECK CODE*'),        38
          31:39,1,BI,M10,LENGTH=5,         39
          120:X)                             40

```

行 説明

## 分類処理の例

- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
- 03** SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。
- 04** SORTIN DD ステートメント。入力データ・セットは GRP.RECORDS という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから、RECFM が FB、LRECL が 80、および BLKSIZE が 23440 であると判別します。
- 05** ALLGPS DD ステートメント。最初の OUTFIL 出力データ・セットは GRP.ALLGRPS という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 06 ~ 07**  
ALLBU DD ステートメント。2 番目の OUTFIL 出力データ・セットは GRP.BU という名前をもち、3390 に割り振られ、カタログされています。DFSORT は、SORTIN から RECFM および LRECL をセットし、適切な BLKSIZE を選択します。
- 08** G1STATS DD ステートメント。3 番目の OUTFIL 出力データ・セットは、SYSOUT クラス A に出力されます。これは OUTFIL 報告書データ・セットであるため、DFSORT は、RECFM を FBA (SORTIN からの FB と ANSI 制御文字からの A) をセットし、LRECL を 121 (ANSI 制御文字用に 1 バイト、データ用に 120 バイト) をセットします。DFSORT は適切な BLKSIZE をセットします。
- 09** G2STATS DD ステートメント。4 番目の OUTFIL 出力データ・セットは、SYSOUT クラス A に出力されます。これは OUTFIL 報告書データ・セットであるため、DFSORT は、RECFM を FBA (SORTIN からの FB と ANSI 制御文字からの A) をセットし、LRECL を 121 (ANSI 制御文字用に 1 バイト、データ用に 120 バイト) をセットします。DFSORT は適切な BLKSIZE をセットします。
- 10** SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 11** SORT ステートメント。FIELDS は、6 桁目から始まる昇順の 5 バイト文字制御フィールドを指定します。
- 12** OUTFIL ステートメント。分類済みの入力レコードは、ALLGPS および ALLBU データ・セットへ書き込まれます。
- 13 ~ 26**  
OUTFIL ステートメント。1 桁目から 3 桁目にある、'G01' を含む分類済みの入力レコードのサブセットが使用され、G1STATS データ・セットへ書き込まれる報告書を作成します。
- 27 ~ 40**  
OUTFIL ステートメント。1 桁目から 3 桁目にある、'G02' を含む分類済みの入力レコードのサブセットが使用され、G2STATS データ・セットへ書き込まれる報告書を作成します。

## 例 11. SmartBatch パイプおよび OUTFIL SPLIT を使用した分類処理

```

INPUT SmartBatch パイプ
OUTPUT
    SmartBatch パイプ
WORK DATA SETS
    動的に割り振られる
USER EXITS
    なし
FUNCTIONS/OPTIONS
    FILSZ、OUTFIL、DYNALLOC

//EXAMP JOB A400,PROGRAMMER                                01
//RUNSORT EXEC PGM=ICESMAN                                02
//SYSOUT DD SYSOUT=H                                      03
//SORTIN DD DSN=INPUT.PIPE,SUBSYS=PIPE,                   04
//      DCB=(LRECL=60,RECFM=FB,BLKSIZE=32760)           05
//OUT1 DD DSN=OUTPUT.PIPE1,SUBSYS=PIPE,                  06
//      DCB=(LRECL=60,RECFM=FB,BLKSIZE=32760)           07
//OUT2 DD DSN=OUTPUT.PIPE2,SUBSYS=PIPE,                  08
//      DCB=(LRECL=60,RECFM=FB,BLKSIZE=32760)           09
//SYSIN DD *                                              10
OPTION DYNALLOC,FILSZ=U1000000                            11
SORT FIELDS=(1,20,CH,A,25,4,BI,A)                         12
OUTFIL FNAMES=(OUT1,OUT2),SPLIT                          13

```

## 行 説明

**01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。

**02** EXEC ステートメント。DFSORT を直接呼び出します。

**03** SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス H に出力します。

**04 ~ 05**

SORTIN DD ステートメント。SUBSYS=PIPE パラメーターは、INPUT.PIPE という名前のパイプのための、'PIPE' SmartBatch サブシステムへの割り振りを指示します。DCB ステートメントは、サブシステム PIPE にデータ・セット特性を記述します。

**06 ~ 07**

OUT1 DD ステートメント。SUBSYS=PIPE パラメーターは、OUTPUT.PIPE1 という名前のパイプのための、'PIPE' SmartBatch サブシステムへの割り振りを指示します。DCB ステートメントは、サブシステム PIPE にデータ・セット特性を記述します。

**08 ~ 09**

OUT2 DD ステートメント。SUBSYS=PIPE パラメーターは、OUTPUT.PIPE2 という名前のパイプのための、'PIPE' SmartBatch サブシス

## 分類処理の例

テムへの割り振りを指示します。DCB ステートメントは、サブシステム PIPE にデータ・セット特性を記述します。

- 10 SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 11 OPTION ステートメント。DYNALLOC は、装置タイプおよび装置の数のインストール・デフォルトを使用して、作業データ・セットが動的に割り振られることを指定します。FILSZ=U1000000 は、百万の入力レコードの見積もりを指定します。
- 12 SORT ステートメント。FIELDS は、1 桁目から始まる昇順の 20 バイトの文字制御フィールドと、25 桁目から始まる昇順の 4 バイトの 2 進数の制御フィールドを指定します。
- 13 OUTFIL ステートメント。SORTIN パイプからのレコードが分類され、OUT1 および OUT2 パイプに交互に書き込まれます (つまり、分類されたレコードは 2 つのパイプに等分されます)。

## 例 12. INCLUDE および LOCALE を使用した分類処理

<b>INPUT</b>	固定長レコード・データ・セット	
<b>OUTPUT</b>	固定長レコード・データ・セット	
<b>WORK DATA SETS</b>	動的に割り振られる (デフォルトによる)	
<b>USER EXITS</b>	なし	
<b>FUNCTIONS/OPTIONS</b>	INCLUDE, LOCALE	
//EXAMP	JOB A400,PROGRAMMER	01
//STEP1	EXEC PGM=SORT,PARM='LOCALE=FR_CA'	02
//STEPLIB	DD DSN=SYS1.SCEERUN,DISP=SHR	03
//SYSOUT	DD SYSOUT=A	04
//SORTIN	DD DSN=INPUT.FRENCH.CANADA,DISP=SHR	05
//SORTOUT	DD DSN=OUTPUT.FRENCH.CANADA,DISP=OLD	06
//SYSIN	DD *	07
	SORT FIELDS=(1,20,CH,A,25,1,BI,D,30,10,CH,A)	08
	INCLUDE COND=(40,6,CH,EQ,50,6,CH)	09

- | 行  | 説明   |
|----|--|
| 01 | JOB ステートメント。このジョブをオペレーティング・システムに知らせます。   |
| 02 | EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。EXEC PARM に指定された LOCALE は、LOCALE のインストール・デフォルトを指定変更します。フランス語およびカナダの文化規則のロケールが活動状態になります。 |
| 03 | STEPLIB DD ステートメント。動的にロード可能なロケールを含む言語環境プログラム実行時間ライブラリーを指定します。  |

- 04 SYSOUT ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。
- 05 SORTIN DD ステートメント。入力データ・セットは INPUT.FRENCH.CANADA という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 06 SORTOUT DD ステートメント。出力データ・セットは OUTPUT.FRENCH.CANADA という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 07 SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 08 SORT ステートメント。FIELDS は、1 桁目から始まる昇順の 20 バイトの文字制御フィールド、25 桁目から始まる降順の 1 バイトの 2 進数制御フィールド、および 30 桁目から始まる昇順の 10 バイトの文字制御フィールドを指定します。文字 (CH) 制御フィールドは、ロケール FR\_CA に定義された照合規則に従って分類されます。
- 09 INCLUDE ステートメント。COND は、40 桁目と 50 桁目で始まる 6 バイトの文字比較フィールドと等しい入力レコードだけを出力データ・セットに組み込むことを指定します。文字 (CH) 比較フィールドは、ロケール FR\_CA に定義された照合規則に従って比較されます。

## 例 13. HFS ファイルでの分類

```

INPUT  連結された HFS ファイル
OUTPUT
        HFS ファイル
WORK DATA SETS
        動的に割り振られる (デフォルトによる)
USER EXITS
        なし
FUNCTIONS/OPTIONS
        なし
//EXAMP  JOB  A400,PROGRAMMER                01
//S1     EXEC  PGM=SORT                      02
//SYSOUT DD  SYSOUT=A                        03
//SORTIN DD  PATH='/user/hfs.inp1.txt',PATHOPTS=ORDONLY,  04
//       LRECL=80,BLKSIZE=240,RECFM=FB,FILEDATA=TEXT    05
//       DD  PATH='/user/hfs.inp2.txt',PATHOPTS=ORDONLY,  06
//       LRECL=80,BLKSIZE=80,RECFM=F,FILEDATA=TEXT      07
//SORTOUT DD  PATH='/user/hfs.ut.txt',PATHOPTS=OWRONLY,  08
//       LRECL=80,BLKSIZE=80,RECFM=F,FILEDATA=TEXT      09
//SYSIN  DD  *                                     10
        SORT  FIELDS=(10,8,CH,A)                11

```

## 分類処理の例

行	説明
01	JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
02	EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
03	SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントをシステム出力クラス A に出力します。
04 ~ 05	SORTIN DD ステートメント。最初の入力ファイルは、/user/hfs.inp1.txt という名前の HFS ファイルです。読み取りアクセスだけ許可されます。ファイルは、テキスト・ファイルとして定義され、レコード・サイズが 80、ブロック・サイズが 240 の固定長レコードをもちます。
06 ~ 07	2 番目の入力ファイルは、/user/hfs.inp2.txt という名前の HFS ファイルです。読み取りアクセスだけ許可されます。ファイルは、テキスト・ファイルとして定義され、レコード・サイズが 80、ブロック・サイズが 80 の固定長レコードをもちます。
08 ~ 09	SORTOUT DD ステートメント。出力ファイルは、/user/hfs.ut.txt という名前の HFS ファイルです。書き込みアクセスだけ許可されます。ファイルは、テキスト・ファイルとして定義され、レコード・サイズが 80、ブロック・サイズが 80 の固定長レコードをもちます。
10	SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
11	SORT ステートメント。FIELDS は、10 桁目から始まる昇順の 8 バイトの文字制御フィールドを指定します。

## 組み合わせの例

このセクションでは、2 つの組み合わせの例を説明します。

### 例 1. EQUALS を使用した組み合わせ

```

INPUT  DASD 上のブロック化固定長レコード

OUTPUT
          3390 上のブロック化固定長レコード

WORK DATA SETS
          未適用

USER EXITS
          なし

FUNCTIONS/OPTIONS
          EQUALS

//EXAMP   JOB   A400,PROGRAMMER                01
//STEP1   EXEC  PGM=SORT                        02
//SYSOUT   DD   SYSOUT=A                       03
//SORTIN01 DD   DSN=M1234.INPUT1,DISP=SHR      04
//SORTIN02 DD   DSN=M1234.INPUT2,DISP=SHR      05
//SORTIN03 DD   DSN=M1234.INPUT3,DISP=SHR      06
//SORTOUT  DD   DSN=M1234.MERGOUT,DISP=(,KEEP), 07
//        SPACE=(CYL,(2,4)),UNIT=3390         08
//SYSIN    DD   *                               09
           MERGE  FIELDS=(1,8,CH,A,20,4,PD,A)   10
           OPTION EQUALS                       11

```

行	説明
01	JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
02	EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
03	SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。
04	SORTIN01 DD ステートメント。最初の入力データ・セットは M1234.INPUT1 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
05	SORTIN02 DD ステートメント。2 番目の入力データ・セットは M1234.INPUT2 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
06	SORTIN03 DD ステートメント。3 番目の入力データ・セットは M1234.INPUT3 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。

## 組み合わせの例

### 07 ~ 08

**SORTOUT DD** ステートメント。出力データ・セットは M1234.MERGOUT という名前をもち、3390 に割り振られて保持されます。DFSORT は、**SORTINnn** データ・セットから RECFM および LRECL をセットし、適切な BLKSIZE を選択します。

**09** **SYSIN DD** ステートメント。DFSORT 制御ステートメントが続きます。

**10** **MERGE** ステートメント。**FIELDS** は、1 桁目から始まる昇順の 8 バイトの文字制御フィールドと、20 桁目から始まる昇順の 4 バイトのパック 10 進数フィールドを指定します。各入力データ・セット内のレコードは、あらかじめ指定された順序になっている必要があります。

**25** **OPTION** ステートメント。**EQUALS** は、等しい制御フィールドをもつ出力レコードの順序が、入力データ・セットのファイル番号と各入力データ・セット内のレコードの元の順序に基づいて決まることを指定します。

## 例 2. LOCALE および OUTFIL を使用した組み合わせ処理

**INPUT** 固定長レコード・データ・セット

### OUTPUT

複数の固定長レコード・データ・セット

### WORK DATA SETS

未適用

### USER EXITS

なし

### FUNCTIONS/OPTIONS

LOCALE、OUTFIL

```
//EXAMP JOB A400,PROGRAMMER 01
//STEP1 EXEC PGM=SORT 02
//STEPLIB DD DSN=SYS1.SCEERUN,DISP=SHR 03
//SYSOUT DD SYSOUT=A 04
//SORTIN01 DD DSN=INPUT01.GERMAN.GERMANY,DISP=SHR 05
//SORTIN02 DD DSN=INPUT02.GERMAN.GERMANY,DISP=SHR 06
//SORTIN03 DD DSN=INPUT03.GERMAN.GERMANY,DISP=SHR 07
//GP1 DD DSN=OUTPUT.GERMAN.GP1,DISP=OLD 08
//GP2 DD DSN=OUTPUT.GERMAN.GP2,DISP=OLD 09
//GP3 DD DSN=OUTPUT.GERMAN.SAVE,DISP=OLD 10
//DFSPARM DD * 11
LOCALE=De_DE.IBM-1047 12
MERGE FIELDS=(25,5,CH,A,40,4,PD,D) 13
OUTFIL FNAMES=GP1, 14
INCLUDE=(23,1,CH,LE,C'Ö') 15
OUTFIL FNAMES=GP2, 16
INCLUDE=(23,1,CH,GT,C'Ö',AND, 17
23,1,CH,LT,C'Ü') 18
OUTFIL FNAMES=GP3,SAVE 19
```

### 行 説明

**01** **JOB** ステートメント。このジョブをオペレーティング・システムに知らせます。



- 02** EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
- 03** STEPLIB DD ステートメント。動的にロード可能なロケールを含む言語環境プログラム実行時間ライブラリーを指定します。
- 04** SYSOUT ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。
- 05** SORTIN01 DD ステートメント。最初の入力データ・セットは INPUT01.GERMAN.GERMANY という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 06** SORTIN02 DD ステートメント。2 番目の入力データ・セットは INPUT02.GERMAN.GERMANY という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 07** SORTIN03 DD ステートメント。3 番目の入力データ・セットは INPUT03.GERMAN.GERMANY という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 08** GP1 DD ステートメント。最初の OUTFIL 出力データ・セットは OUTPUT.GERMAN.GP1 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 09** GP2 DD ステートメント。2 番目の OUTFIL 出力データ・セットは OUTPUT.GERMAN.GP2 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 10** GP3 DD ステートメント。3 番目の OUTFIL 出力データ・セットは OUTPUT.GERMAN.GP3 という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。
- 11** DFSPARM DD ステートメント。DFSORT 制御ステートメントが続きます。
- 12** LOCALE パラメーター。LOCALE のインストール・デフォルトを指定変更します。ドイツ語と IBM-1047 エンコード・文字セットに基づくドイツの文化規則のロケールが活動状態になります。
- 13** MERGE ステートメント。FIELDS は、25 桁目から始まる昇順の 5 バイトの文字制御フィールドと、40 桁目から始まる降順の 4 バイトのパック 10 進数制御フィールドを指定します。文字 (CH) 制御フィールドは、ロケール De\_DE.IBM-1047 に定義された照合規則に従って組み合わせられます。各入力データ・セット内のレコードは、あらかじめ指定された順序になっている必要があります。
- 14 ~ 15** OUTFIL ステートメント。23 桁目に 'Ö' 以下の文字値をもつレコードのサ

## 組み合わせの例

ブセットが GP1 出力データ・セットに書き込まれます。文字 (CH) 比較フィールドおよび文字定数は、ロケール De\_DE.IBM-1047 に定義された照合規則に従って比較されます。

### 16 ~ 18

OUTFIL ステートメント。23 桁目に 'Ö' よりは大きい 'Ü' より小さい文字値をもつレコードのサブセットが、GP2 出力データ・セットに書き込まれます。文字 (CH) 比較フィールドおよび文字定数は、ロケール De\_DE.IBM-1047 に定義された照合規則に従って比較されます。

### 19

OUTFIL ステートメント。GP1 または GP2 出力データ・セットに書き込まれないレコードはすべて、GP3 出力データ・セットに書き込まれます。

---

## コピー処理の例

このセクションでは、2 つのコピー処理の例を説明します。

## 例 1. EXEC PARMs、SKIPREC、MSGPRT および ABEND を使用した コピー処理

**INPUT** 複数ボリューム 3490 上のブロック化固定長レコード

**OUTPUT**

DASD 上のブロック化固定長レコード

**WORK DATA SETS**

未適用

**USER EXITS**

なし

**FUNCTIONS/OPTIONS**

SKIPREC、MSGPRT、ABEND

```
//EXAMP JOB A400,PROGRAMMER 01
//STEP1 EXEC PGM=SORT, 02
// PARM='SKIPREC=500,MSGPRT=CRITICAL,ABEND' 03
//SYSOUT DD SYSOUT=A 04
//SORTIN DD DSN=FLY.RECORDS,VOL=SER=(000333,000343), 05
// UNIT=(3490,2),DISP=OLD,LABEL=(,NL), 06
// DCB=(RECFM=FB,LRECL=12000,BLKSIZ=24000) 07
//SORTOUT DD DSN=FLY.RECORDS.COPY,DISP=OLD 08
//SYSIN DD * 09
SORT FIELDS=COPY 10
```

行 説明

**01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。

**02 ~ 03**

EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。SKIPREC=500 は、最初の 500 個の入力レコードを出力データ・セットに含めないことを指定します。MSGPRT=CRITICAL は、通知メッセージではなく、エラー・メッセージを印刷することを指定します。ABEND は、DFSORT がエラー・メッセージを出した場合、ユーザー ABEND で終了することを指定します。

**04** SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。

**05 ~ 07**

SORTIN DD ステートメント。入力データ・セットは FLY.RECORDS という名前をもち、3490 ボリューム 000333 と 000343 に置かれます。UNIT パラメーターは、データ・セットの各ボリュームにそれぞれ 1 つずつの、合計 2 つのタイプの装置を要求します。テープにはラベルが付いていないので、DCB パラメーターを提供して、RECFM が FB、LRECL が 12000、および BLKSIZ が 24000 ということを示す必要があります。

**08** SORTOUT DD ステートメント。出力データ・セットは

## コピー処理の例

FLY.RECORDS.COPY という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから RECFM、LRECL および BLKSIZE を判別します。

- 09 SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 10 SORT ステートメント。FIELDS=COPY はコピー・アプリケーションを指定します。

## 例 2. INCLUDE および VLSHRT を使用したコピー処理

```

INPUT  DASD 上のブロック化スパン・レコード
OUTPUT
          SYSDA 上のブロック化スパン・レコード
WORK DATA SETS
          未適用
USER EXITS
          なし
FUNCTIONS/OPTIONS
          INCLUDE、VLSHRT

//EXAMP   JOB   A400,PROGRAMMER                01
//COPY    EXEC  PGM=SORT                        02
//SYSOUT   DD   SYSOUT=A                        03
//SORTIN   DD   DSN=SMF.DATA,DISP=SHR           04
//SORTOUT  DD   DSN=SMF.VIOL,DISP=(,KEEP),SPACE=(CYL,(2,5)), 05
//        UNIT=SYSDA                            06
//SYSIN    DD   *                                07
          INCLUDE COND=(6,1,FI,EQ,80,AND,19,1,BI,EQ,B'1.....') 08
          OPTION COPY,VLSHRT                    09

```

## 行 説明

- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。DFSORT をその別名 SORT で直接呼び出します。
- 03** SYSOUT DD ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。
- 04** SORTIN DD ステートメント。入力データ・セットは SMF.DATA という名前をもち、カタログされています。DFSORT は、データ・セット・ラベルから、RECFM が VBS、LRECL が 32760、および BLKSIZE が 23476 であると判別します。
- 05 ~ 06** SORTOUT DD ステートメント。出力データ・セットは SMF.VIOL という名前をもち、SYSDA に割り振られて保持されます。DFSORT は、SORTIN から RECFM および LRECL をセットし、適切な BLKSIZE を選択します。
- 07** SYSIN DD ステートメント。DFSORT 制御ステートメントが続きます。
- 08** INCLUDE ステートメント。COND は、6 桁目の 1 バイトの固定小数点フィールドが 10 進数 80 の入力レコード、および 19 桁目の 1 バイトの 2 進数フィールドのビット 0 がオンの入力レコードだけを、出力データ・セットに含めるように指定します。
- 09** OPTION ステートメント。COPY はコピー・アプリケーションを指定しま

## コピー処理の例

す。VLSHRT は、INCLUDE 比較フィールドのすべてを含めるには短すぎるレコードを出力データ・セットに含めないように指定します。

## ICEGENER の例

このセクションでは、ICEGENER の例を説明します。

```

INPUT  IEBGENER ジョブの場合と同じ
OUTPUT
          IEBGENER ジョブの場合と同じ
WORK DATA SETS
          未適用
USER EXITS
          なし
FUNCTIONS/OPTIONS
          なし

//EXAMP  JOB  A400,PROGRAMMER                01
//GENR   EXEC  PGM=ICEGENER                  02
//SYSPRINT DD  SYSOUT=A                      03
//SYSUT1 DD  DSN=CTL.MASTER,DISP=SHR        04
//SYSUT2 DD  DSN=CTL.BACKUP,DISP=OLD        05
//SYSIN  DD  DUMMY                           06

```

この例は、ユーザーが IEBGENER の自動置換としての ICEGENER を未インストールの場合に、IEBGENER 用に ICEGENER 機能を使用する方法を示しています。ICEGENER 機能は、この IEBGENER ジョブ用にさらに効率的な DFSORT コピー機能を選択します。

#### 行 説明

- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。ICEGENER 機能を呼び出します。  
PGM=IEBGENER は PGM=ICEGENER により置き換えられました。
- 03 ~ 06**  
IEBGENER ジョブに対するその他の変更は必要ではありません。

## ICETOOL の例

このセクションでは、種々の演算子を使用する ICETOOL の例を説明します。

```

INPUT 複数の出力データ・セット

OUTPUT
    複数の出力データ・セット

WORK DATA SETS
    動的に割り振られた (自動)

USER EXITS
    ICETOOL の E35 (自動)

FUNCTIONS/OPTIONS
    OCCUR、COPY、SORT、MODE、VERIFY、STATS、DISPLAY

//EXAMP    JOB A400,PROGRAMMER                                01
//TOOLRUN  EXEC PGM=ICETOOL,REGION=1024K                      02
//TOOLMSG  DD SYSOUT=A                                        03
//DFSMSG   DD SYSOUT=A                                        04
//TOOLIN   DD *                                              05
* Print report showing departments with less than 5 employees 06
  OCCUR FROM(IN1) LIST(LT5) LOWER(5) BLANK -                  07
    TITLE('Small Departments') PAGE -                         08
    HEADER('Department') HEADER('Employees') -              09
    ON(45,3,CH) ON(VALCNT)                                    10
* Copy and reformat selected records                            11
  COPY USING(CJ69) FROM(IN1) TO(OUTJ69D)                      12
  COPY USING(CJ82) FROM(IN1) TO(OUTJ82D)                      13
* Sort/save/print the resulting combined data sets            14
  SORT FROM(CONCAT) TO(DEPTSD,DEPTSP) USING(ABCD)            15
* Do following operators even if a previous operator failed,  16
* but stop processing if a subsequent operator fails.         17
  MODE STOP                                                    18
* Verify decimal fields                                        19
  VERIFY FROM(IN2) ON(22,6,PD) ON(30,3,ZD)                    20
* Print statistics for record length and numeric fields       21
  STATS FROM(IN2) ON(VLEN) ON(22,6,PD) ON(30,3,ZD)           22
* Sort and produce total for each unique key                  23
  SORT FROM(IN2) TO(OUT4) USING(CTL1)                         24
* Print report containing:                                     25
* - key and total for each unique key                          26
* - lowest and highest of the totals                           27
  DISPLAY FROM(OUT4) LIST(LIST1) -                             28
    TITLE('Unique key totals report') DATE TIME -            29
    ON(5,10,CH) ON(22,6,PD) ON(30,3,ZD) -                     30
    MINIMUM('Lowest') MAXIMUM('Highest') PLUS                31
//LT5      DD SYSOUT=A                                        32
//CJ69CNTL DD *                                              33
* Select J69 employees, reformat fields, and insert text     34
  INCLUDE COND=(45,3,CH,EQ,C'J69')                            35
  OUTREC FIELDS=(21,10,X,1,15,C'is in department J69',34X)   36

```



```

//CJ82CNTL DD *                               37
* Select J82 employees, reformat fields, and insert text 38
  INCLUDE COND=(45,3,CH,EQ,C'J82')           39
  OUTREC FIELDS=(21,10,X,1,15,C'is in department J82',34X) 40
//IN1     DD DSN=FLY.INPUT1,DISP=SHR         41
//OUTJ69D DD DSN=&&OUTJ69D,DISP=(,PASS),SPACE=(TRK,(10,10)), 42
// UNIT=SYSDA                                43
//OUTJ82D DD DSN=&&OUTJ82D,DISP=(,PASS),SPACE=(TRK,(10,10)), 44
// UNIT=SYSDA                                45
//CONCAT  DD DSN=*.OUTJ69D,VOL=REF=*.OUTJ69D,DISP=(OLD,PASS) 46
//       DD DSN=*.OUTJ82D,VOL=REF=*.OUTJ82D,DISP=(OLD,PASS) 47
//ABCDCNTL DD *                               48
* Sort by last name, first name              49
  SORT FIELDS=(12,15,CH,A,1,10,CH,A)         50
//DEPTSD  DD DSN=FLY.OUTPUT1,DISP=SHR        51
//DEPTSP  DD SYSOUT=A                        52
//IN2     DD DSN=FLY.INPUT2,DISP=SHR        53
//OUT4    DD DSN=FLY.OUTPUT2,DISP=OLD       54
//CTL1CNTL DD *                               55
* Sort and produce totals in one record for each unique key 56
  SORT FIELDS=(5,10,CH,A)                    57
  SUM FIELDS=(22,6,PD,30,3,ZD)              58
//LIST1   DD SYSOUT=A                        59

```

この例は、単一ステップで複数の操作を実行するための ICETOOL の使用方法を示しています。

#### 行 説明

- 01** JOB ステートメント。このジョブをオペレーティング・システムに知らせます。
- 02** EXEC ステートメント。推奨 REGION の 1024 キロバイトを指定して、ICETOOL を呼び出します。
- 03** TOOLMSG DD ステートメント。ICETOOL メッセージおよびステートメントをシステム出力クラス A に出力します。
- 04** DFSMSG DD ステートメント。DFSORT メッセージおよび制御ステートメントを SYSOUT クラス A に出力します。
- 05** TOOLIN DD ステートメント。ICETOOL ステートメントが続きます。ICETOOL 実行のための MODE は最初に STOP にセットされています。ある演算子についてエラーが検出されると、SCAN モードに入ります。
- 06** 注釈ステートメント。印刷されますが、それ以外は無視されます。
- 07 ~ 10** OCCUR 演算子。LT5 データ・セットに、IN1 データ・セット内の指定フィールドの各値、およびその値が発生した回数を記述する報告書を印刷します。
- 11** 注釈ステートメント。
- 12** COPY 演算子。IN1 データ・セットからのレコードは、CJ69CNTL データ・セット内の DFSORT 制御ステートメントを使用して、OUTJ69D データ・セットにコピーされます。その結果、&&OUTJ69D には FLY.INPUT1 からの再フォーマット設定された後のレコードのサブセットが含まれます (これらのレコードは 45 から 47 桁目に 'J69' を含みます)。

## ICETOOL の例

- 13** COPY 演算子。IN1 データ・セットからのレコードは、CJ82CNTL データ・セット内の DFSORT 制御ステートメントを使用して、OUTJ82D データ・セットにコピーされます。その結果、&&OUTJ82D には FLY.INPUT1 からの再フォーマット設定された後のレコードのサブセットが含まれます (これらのレコードは 45 から 47 桁目に 'J82' を含みます)。
- 14** 注釈ステートメント。
- 15** SORT 演算子。CONCAT データ・セットからのレコードは、ABCDCNTL データ・セット内の DFSORT 制御ステートメントを使用して、DEPTSD および DEPTSP データ・セットに分類されます。その結果、FLY.OUTPUT1 および DEPTSP (SYSOUT) には、&&OUTJ69D および &&OUTJ82D からの、分類され、結合されたレコードが含まれます。
- 16 ~ 17** 注釈ステートメント。
- 18** MODE 演算子。MODE は STOP にリセットされます (前の演算子に関するエラーのために、SCAN モードに入っていた場合に必要)。後続の演算子についてエラーが検出されると、SCAN モードに入ります。これは、前の演算子と後続の演算子を 2 つの無関係なグループに分けます。
- 19** 注釈ステートメント。
- 20** VERIFY 演算子。IN2 データ・セットの指定した 10 進数フィールド内に無効な値がある場合、その無効値を識別します。FLY.INPUT2 内に無効値が見つかり、後続の操作を停止するために使用されます。
- 21** 注釈ステートメント。
- 22** STATS 演算子。IN2 データ・セットの指定したフィールドについて、最小、最大、平均、および合計を印刷します。  
ON(VLEN) は、FLY.INPUT2 内のレコードのレコード長に機能します。したがって、ON(VLEN) について印刷された値は、FLY.INPUT2 の最小レコード、最大レコード、レコード長の平均、および合計バイト数を表します。
- 23** 注釈ステートメント。
- 24** SORT 演算子。IN2 データ・セットからのレコードは、CTL1CNTL データ・セット内の DFSORT 制御ステートメントを使用して、OUT4 データ・セットに分類され集計されます。その結果、FLY.OUTPUT2 には、集計フィールドの合計をもつ各固有な分類フィールドのそれぞれについて、FLY.INPUT2 からのレコードが 1 つ含まれます。
- 25 ~ 27** 注釈ステートメント。
- 28 ~ 31** DISPLAY 演算子。LIST1 データ・セット内に、前の操作の結果得られた OUT4 データ・セットの各分類値および合計値、さらに各合計フィールドごとの最低値および最高値を記述する報告書を印刷します。
- 32 ~ 59** DD ステートメント。上記の ICETOOL 操作で使用されるデータ・セットおよび DFSORT 制御ステートメントを定義します。

---

## 付録 A. ワークスペースの使用

はじめに . . . . .	685
ハイパー空間 . . . . .	686
作業データ・セット装置 . . . . .	686
DASD および磁気テープ装置 . . . . .	686
装置の数 . . . . .	687
非同期記憶サブシステム . . . . .	687
作業データ・セットの割り振り . . . . .	688
作業データ・セットの動的割り振り . . . . .	689
自動的な動的割り振り . . . . .	689
装置のデフォルト . . . . .	690
ファイル・サイズおよび動的割り振り . . . . .	690
ワークスペースの動的過剰割り振り . . . . .	691
作業データ・セットの JCL 割り振り . . . . .	691
DASD 容量に関する考慮事項 . . . . .	693
DASD ワークスペース容量の超過 . . . . .	693
磁気テープ容量に関する考慮事項 . . . . .	694
磁気テープ・ワークスペース容量の超過 . . . . .	694

---

### はじめに

分類アプリケーションを仮想記憶域内だけで実行できない場合は、DFSORT はワークスペースを使用する必要があります。必要なワークスペースの大きさは、次のものにより異なります。

- 分類するデータの量
- DFSORT で使用できる仮想記憶域の大きさ
- DFSORT で使用できるハイパー空間の大きさ
- 使用する装置のタイプ
- DFSORT 機能およびユーザーが使用する機能 (たとえば、VLSHRT、ロケール処理、EFS、および ALTSEQ は、必要なワークスペースの大きさを増やすことができます)。

DFSORT アプリケーションにワークスペースを提供するためには、次の 3 つの方法があります。

- ハイパー空間
- 作業データ・セットの動的割り振り
- 作業データ・セットの JCL 割り振り

最高のパフォーマンスを得るためには、動的に割り振られた DASD 作業データ・セットと組み合わせて、最適の大きさのハイパー空間を使用することを特にお勧めします。HIPRMAX オプションの使用の詳細については、637 ページの『ハイパー分類の使用』を参照してください。DYNAUTO インストール・オプションまたは DYNALLOC 実行時オプションを使用して、作業データ・セットを動的に割り振ることができます。

---

## ハイパー空間

ハイパー空間は、DFSORT の中間記憶域の最も効率的な形式です。デフォルト ICEMAC オプション HIPRMAX=OPTIMAL を使用すると、DFSORT は、可能な限り、ハイパー分類用にハイパー空間を使用します。ユーザーは ICEMAC パラメーター EXPMAX、EXPOLD、および EXPRES を使用して、HIPRMAX=OPTIMAL の定義を調整できます。詳細については、*DFSORT 導入とカスタマイズ リリース 14*を参照してください。

DFSORT がハイパー空間を使用するかどうかは、拡張記憶域の使用可能度、64 ビットの実モードの中央記憶装置、アプリケーション実行中の他の並行ハイパー分類アプリケーションからの要求、および DFSORT インストール・オプション EXPMAX、EXPOLD、および EXPRES の設定値により決まります。したがって、同じアプリケーションでも、実行が異なると、さまざまな大きさのハイパー空間を使用することになります。十分な大きさのハイパー空間が使用できる場合は、DFSORT はハイパー空間を中間記憶域に専用します。ハイパー空間の大きさが不十分な場合は、DFSORT はハイパー空間と作業データ・セットを組み合わせて使用します。作業データ・セットだけを使用することもあります。

すべての DFSORT ハイパー空間データをサポートするのに十分な記憶域がある場合は、DFSORT はハイパー分類だけを使用します。ハイパー分類は非常に動的です。つまり、複数の並行ハイパー分類アプリケーションは、常に、互いに他の記憶域の要求を知っており、記憶域の同じ部分を用いて、それぞれのハイパー空間をサポートしようとはしません。さらに、DFSORT は、実行中、使用可能な記憶域を調べていて、記憶域の不足が予想される場合、あるいはシステムでの全ハイパー分類活動が DFSORT のインストール・オプション EXPMAX、EXPOLD、および EXPRES により設定された限界に達した場合は、ハイパー空間の使用から作業データ・セットの使用へ切り替わります。

ハイパー分類では、ハイパー空間だけでなく、作業データ・セットが使用できることが必要です。作業データ・セットが明示的に要求されていない場合、DFSORT は、ハイパー分類の動的割り振りを強制的に実行します。詳細については、181 ページの『OPTION 制御ステートメント』の HIPRMAX オプションを参照してください。

---

## 作業データ・セット装置

作業データ・セット用に選択した装置のタイプが、パフォーマンスに大きな影響を与えることがあります。作業データ・セット用に装置を選択する際には、次のことを考慮に入れてください。

## DASD および磁気テープ装置

最適なパフォーマンスを得るためには、作業データ・セットについて他の活動がほとんど行われない直接アクセス装置を使用します。Specify エミュレートされた 3390-9 装置 (RAMAC など) やその他の高速の IBM DASD 装置を使用し、テープ、仮想 (VIO) または 実 3390-9 装置の指定は極力避けてください。

DASD ではなく、磁気テープ装置を作業データ・セットに使用すると、次の理由から大幅なパフォーマンスの低下を招きます。

- テープ作業データ・セットを使用すると、DFSORT は、より効率的な分類手法であるブロック・セット手法およびピアレッジ / ベールを使用できなくなります。DASD 作業データ・セットを使用すれば、DFSORT は、これらの手法を使用できます。
- テープ作業データ・セットは、順次アクセスする必要があります。DASD データ・セットは、ランダムにアクセスできます。
- DASD 制御装置は、磁気テープ装置では使用できないキャッシュ高速書き込みなどの機能を実行できます。

## 装置の数

1 つの作業データ・セットで十分ですが、別々の装置で 2 つ以上の作業データ・セットを使用すると、通常、アプリケーションの経過時間を大幅に短縮できます。一般に、4 つ以上の作業データ・セットを使用しても、それ以上経過時間が短縮されることはありません。これは、作業データ・セットが小さいか、またはファイル・サイズが大きい場合のみ必要になります。

仮想記憶域などのリソースを最適に割り振るために、作業データ・セットの数は必要以上に多く指定しないでください。

255 個以上の作業データ・セットは指定できません。32 個より多い作業データ・セットを指定し、ブロック・セット手法を選択しない場合は、最大 32 個の作業データ・セットが使用されます。

## 非同期記憶サブシステム

非同期記憶サブシステムに接続された装置での作業データ・セットの割り振りが、いくつかの DFSORT アプリケーションのパフォーマンスに影響を与えることがあります。特定のアプリケーションが影響を受けるかどうかは、多くの要因により異なりますが、最も大きな要因は、入力ファイル・サイズと使用可能な記憶域との比率です。

一般に、パフォーマンスを最大化するには、DFSORT に次のものが必要になります。

- 分類するファイルのサイズを正確に知っていること。

多くの場合、DFSORT はファイル・サイズを正確に計算できます。ただし、DFSMSrmm または ICETPEX を使用するテープ管理システムにより管理されない、多くの入力テープ (特に圧縮された入力テープ) を分類するアプリケーション、またはレコードの追加や削除を行う E15 出口を使用するアプリケーションの場合は、FILSZ または SIZE パラメーターを使用してファイル・サイズを指定することをお勧めします。

- 分類するファイルのサイズに対して十分な記憶域を確保していること。

表 74 は、各種の入力ファイル・サイズに基づいて、DFSORT に提供する記憶域の推奨最少量を示したものです。

表 74. 各種のファイル・サイズに必要な最小記憶域

入力ファイル・サイズ	最小記憶域
200 メガバイトより小さい	4 メガバイト
200 メガバイト ~ 500 メガバイト	8 メガバイト

## ワークスペースの使用

表 74. 各種のファイル・サイズに必要な最小記憶域 (続き)

入力ファイル・サイズ	最小記憶域
500 メガバイト ~ 1 ギガバイト	16 ギガバイト
1 ギガバイトより大きい	16 ギガバイト ~ 32 ギガバイト

ある状況下では、DFSORT は、ESCON チャンネルを使用した場合、並列チャンネルを使用したときと同じようには効率よく実行しません。次の 2 つのタイプのアプリケーションは、顕著なパフォーマンスの低下を引き起こしやすいものです。

1. 分類するファイルのサイズを DFSORT が正確に判別できないアプリケーション。このようなアプリケーションには、しばしば、レコードを分類プロセスに挿入する DFSORT E15 ユーザー出口が関係します。
2. 入力ファイル・サイズに対する使用可能記憶域の比率が低い分類アプリケーション。

---

## 作業データ・セットの割り振り

JCL 割り振りとは比べた場合、動的割り振りには次のような利点があります。

- すべての分類アプリケーションで動的割り振りを自動的に活動化するように、ICEMAC を設定できます。

JCL 割り振りを使用するためには、適切な DD ステートメントを個々の各アプリケーションに指定する必要があります。

- アプリケーションの特性 (ファイル・サイズ、仮想記憶域など) が時間の経過とともに変わるのに伴い、DFSORT は、そのアプリケーションに動的に割り振られたワークスペースを自動的に最適化できます。これにより、DASD スペースの不要な割り振りがなくなります。

JCL 割り振りは固定されています。DFSORT はそれを調整できません。DASD スペースが無駄に使用される可能性があります。

- アプリケーションが使用できるハイパー空間の大きさは実行のたびに変動するため、DFSORT は、動的に割り振るスペースの大きさを自動的に調整して、ハイパー空間の大きさを補うことができます。これにより、DASD スペースの不要な割り振りがなくなります。

JCL 割り振りは固定されています。すべての分類処理をハイパー空間で行うことができる場合でも、DFSORT はそれを調整できません。DASD スペースが無駄に使用される可能性があります。

動的割り振りには 1 つの欠点があります。すなわち、690 ページの『ファイル・サイズおよび動的割り振り』で説明しているように、アプリケーションによりは、入力ファイル・サイズの妥当な見積もりを DFSORT に与えなければならない場合があるということです。後で、そのアプリケーションの入力ファイル・サイズが大幅に増加する場合は、それに応じて、見積もりのファイル・サイズを更新する必要があります。

ただし、すべてのアプリケーションに適用されるという点を除き、JCL 割り振りにと同じような欠点があります。最初から作業データ・セットを多めに割り振ってス

ペースを無駄にしていない限り、任意のアプリケーションで入力ファイル・サイズが大幅に増加するときには、JCL 割り振りを更新して、スペース不足の異常終了を回避する必要があります。

JCL で十分な作業データ・セットを割り振ることができ、アプリケーションが割り振られたスペースを超えることがないと保証できるような場合は、動的割り振りは必要ありません。ただし、DASD スペースの有効な使用は常に望ましいものであり、JCL 割り振りよりも動的割り振りをお勧めします。

動的割り振りと JCL 割り振りの両方について、次のことが言えます。

- 実際に使用されるワークスペースの大きさが、割り振られた大きさよりも小さいことがよくあります。DFSORT は、動的過剰割り振りを最小化しようと試みる一方、アプリケーションがスペース不足のために失敗することがないように処置します。JCL 割り振りを使用する場合、ユーザーは、割り振りスペースの大きさを手操作で最小化できますが、これを行うには、アプリケーションの特性が時間の経過に伴って変化するのに合わせて、JCL 割り振りを変更しなければなりません。
- DFSORT が使用できる仮想記憶域を制限することにより、必要なワークスペースの大きさを増加させることがあります。適切な大きさの記憶域 (たとえば 4 メガバイト) を使用すると、DFSORT は適切な大きさのワークスペースを使用して分類できます。記憶域が制限されると、さらに多くのワークスペースが必要になることがあります。記憶域が大幅に制限される (たとえば 200KB) と、さらに大量のワークスペースが必要になる可能性があります。

## 作業データ・セットの動的割り振り

ICEMAC オプションを使用して、作業データ・セットの自動的な動的割り振りを要求したり、装置タイプと装置数にデフォルトを提供できます。

ある特定のアプリケーションについては、動的割り振りを使用するときに入力ファイル・サイズの合理的な見積もりを指定することが非常に重要になります。

### 自動的な動的割り振り

システム・プログラマーは、動的割り振りを自動的に使用するか、あるいは DYNALLOC 実行時オプションにより要求されたときだけ使用するかを制御する DYNAUTO インストール・オプションをセットしています。

また、DYNAUTO は、JCL 作業データ・セットが指定されたときに、動的割り振り と JCL 割り振りのうちのどちらを優先するかを制御することもできます。

システム・プログラマーが DYNAUTO=IGNWKDD を選択した場合は、動的割り振りが JCL 割り振りに優先します (JCL 作業データ・セットは実際に割り振り解除されます)。選択されたアプリケーションについて優先順位を反対にしたい場合は、実行時オプション USEWKDD を使用します。

システム・プログラマーが DYNAUTO=YES を選択した場合は、JCL 割り振りが動的割り振りに優先します。反対の優先順位にしたい場合は、JCL 割り振りステートメントを除去する必要があります。

## 作業データ・セットの割り振り

システム・プログラマーが DYNAUTO=NO を選択した場合は、DYNALLOC 実行時オプションを指定しない限り、作業データ・セットの動的割り振りは使用されません。JCL 割り振りが動的割り振りに優先します。

### 装置のデフォルト

動的割り振りの場合の装置タイプ、あるいは装置数が明示的に指定されていない場合、DFSORT は、欠落した情報をシステム・プログラマー提供の DYNALLOC インストール・オプション情報から入手します。

### ファイル・サイズおよび動的割り振り

DFSORT は、分類するバイトの数、つまり入力ファイル・サイズに基づいて、動的に割り振るワークスペースの大きさを決めます。一般に、DFSORT は、入力レコードの数を決定することにより、ファイル・サイズの正確な決定を自動的に行うことができます。しかし、次のような場合は、DFSORT は必ずしも入力ファイル・サイズを正確に決定することができません。

- E15 ユーザー出口ルーチンがすべての入力レコードを提供する (入力データ・セットが存在しない)。 DFSORT は、挿入するレコードの数を自動的に決定することができません。
- E15 ユーザー出口ルーチンとともに、入力データ・セットが存在する。 DFSORT は入力データ・セット内のレコードの数を自動的に決定できますが、挿入または削除するレコードの数を自動的に決定できません。
- スプール (DD \*) またはパイプ・データ・セットが入力として使用される。
- 入力データが、DFSMSrmm または ICETPEX を使用するテープ管理システムにより管理されないテープ上の小さいデータ・セットで構成される。テープ・データ・セットが管理されない場合、DFSORT は使用するテープの数が分からないので、ドライブの通常最大密度でボリュームが満たされていると仮定してファイル・サイズを決定します。
- 改良データ記録機構 (IDRC) を入力装置として使用し、テープ・データ・セットが DFSMSrmm または ICETPEX を使用するテープ管理システムにより管理されない。
- 入力データ・セットが区分データ・セットのメンバーである。 DFSORT は区分データ・セット内のメンバーのサイズを決定することができません。したがって、入力データ・セットが区分化されている場合、DFSORT はデータ・セット全体のサイズを入力ファイル・サイズとして使用します。これは通常、過剰見積もりで、ワークスペースの過剰割り振りにつながります。

このような状況下で、FILSZ または SIZE オプションから複数のレコードが提供されない場合、メッセージ ICE118I が出されます。作業データ・セットの動的割り振りを使用する場合、DFSORT は、有効な DYNSPC 値に応じて 1 次スペースを割り振ります。このとき、過剰割り振りまたは過小割り振り割りや、場合によりスペースの無駄またはスペース不足が個々に発生する可能性があります。このような状況になった場合は、分類するレコード数を非常に正確に見積もって、FILSZ=En を指定する必要があります。 FILSZ=En を指定できない場合、DYNSPC=n を使用して、動的割り振り作業データ・セットの 1 次スペースを適切に調整してください。

注: FILSZ=E0 は無視されます。



可変長レコードの場合、DFSORT は最大レコード長 (LRECL) の 2 分の 1 をレコード数と共に使用して入力ファイル・サイズを決定します。ただし、AVGRLLEN=n を指定した場合を除きます。実際の平均レコード長が最大レコード長の 2 分の 1 と大幅にかけ離れている場合は、AVGRLLEN=n を使用すると DFSORT の動的ワークスペースの過剰割り振りや過小割り振りを防ぐことができます。

AVGRLLEN、DYNMPC、FILSZ、および SIZE オプションの詳細については、181 ページの『OPTION 制御ステートメント』を参照してください。

## ワークスペースの動的過剰割り振り

次のような状況下では、ユーザーがレコード数を指定した場合でも、DFSORT はワークスペースを動的に過剰割り振りすることがあります。

- 以下のものを使用して、多数のレコードを削除する場合
  - INCLUDE または OMIT ステートメント、あるいは SKIPREC オプション。これらのステートメントとオプションを使用しても、DFSORT に SIZE=En または FILSZ=En の指定を使用するように強制 できません。DFSORT は、入力ファイル・サイズを計算できない場合以外は、En の値を無視します。
  - 入力としての 1 つ以上の区分データ・セット・メンバー。DFSORT は、メンバーのサイズではなく、区分データ・セット全体のサイズを計算に使用します。DFSORT は、自分で入力ファイル・サイズを決定できない場合以外は、SIZE=En または FILSZ=En の値を無視します。

SIZE=Un または FILSZ=Un を指定することにより、このような場合の過剰割り振りを避けることができます。

- 可変長レコードの平均レコード長が、最大レコード長の 2 分の 1 よりも著しく短い場合。DFSORT がユーザー指定の正確なレコード数または見積レコード数を使用する場合は、最大レコード長の 2 分の 1 を使用してファイル・サイズを決定します。AVGRLLEN=n を指定することにより、このような場合の過剰割り振りを避けることができます。

ユーザーがレコード数を指定しない場合 (たとえば、テープ上の小さい入力データ・セットを使用する場合)、あるいはレコード数を指定した場合 (たとえば、多数のレコードを削除する場合) でも、ワークスペースの動的過剰割り振りが行われることがあります。このような場合は、作業データ・セットの JCL 割り振りを使用し、割り振りスペースの大きさを制御できます。ただし、これを行う場合は、前述したように、いくつかの欠点があります。DYNAUTO=IGNWKDD を使用する場合に、作業データ・セットの JCL 割り振りを使用するときには、実行時オプションの USEWKDD を忘れずに指定してください。

## 作業データ・セットの JCL 割り振り

必要なワークスペースの大きさは、使用する仮想記憶域や装置タイプなどの、多くの要因により異なりますが、特に入力データ・セットのファイル・サイズの影響を受けます。

多くの変数が関係するため、必要なワークスペースを計算するための正確な式が得られません。しかし通常は、以下のガイドラインが当てはまります。

## 作業データ・セットの割り振り

- 固定長レコード (FLR) 分類アプリケーションの場合は、通常、入力ファイル・サイズの 1.5 ~ 2 倍が適当です。
- 可変長レコード (VLR) 分類アプリケーションの場合は、通常、入力ファイル・サイズの 1.5 ~ 2.5 倍が適当です。

これらのガイドラインは、十分な大きさ (少なくとも 1M) を DFSORT が使用できるものと想定しています。使用可能な記憶域の大きさを制限すると、必要なワークスペースの大きさを増加させることがあります。

DFSORT は、しばしば、上記のガイドラインで示されたワークスペースより小さなワークスペースを用いて実行されます。

作業データ・セットの JCL 割り振りを使用して最良のパフォーマンスを得るためには、次のことを行なってください。

- 装置上での活動があまり多くない装置を使用する。
- 最高のパフォーマンスを得るためには、作業データ・セット用に、3390-9 装置 (RAMAC など) やその他の高速の IBM DASD 装置を使用し、作業データ・セットにテープ、仮想 (VIO)、または実 3390-9 装置を使用することは避けてください。
- スペースをシリンダー単位で割り振る。
- 各作業データ・セットごとに連続スペースを指定し、2 次スペースが必要ないように十分な 1 次スペースを確保する。
- 2 つ以上の作業データ・セットを割り振る。
- アクチュエーターごとに 1 つの作業データ・セットを割り当てる。
- 装置への複数チャネル経路を使用する。
- 作業データ・セットと入力 / 出力データ・セットに対して、異なるスピンドルと別個のチャネル経路を使用する。

次の表は、ハイパー分類とデータ空間分類を使用しない場合 (HIPRMAX=0 および DSPSIZE=0) の、各種の特性をもつアプリケーションに必要な 4 メガバイト記憶域をもつ作業データ・セット・スペースを示したものです。

表 75. 各種の入力特性のために必要なワークスペース

入力データ・セット特性				シリンダー (3390)	
ファイル・サイズ (MB)	FLR/VLR	最大 LRECL	BLKSIZE	入力データ・セット	作業データ・セット
4	FLR	80	27920	6	6
4	FLR	160	27840	6	6
20	FLR	80	27920	26	36
20	FLR	160	27840	26	36
20	FLR	1000	27000	26	36
40	FLR	80	27920	51	56
40	FLR	160	27840	51	56
40	FLR	1000	27000	52	56
150	FLR	160	27840	189	198
4	VLR	300	27998	6	9

表 75. 各種の入力特性のために必要なワークスペース (続き)

入力データ・セット特性				シリンダー (3390)	
ファイル・サイズ (MB)	FLR/VLR	最大 LRECL	BLKSIZE	入力データ・セット	作業データ・セット
40	VLR	300	27998	51	63
40	VLR	6000	27998	55	59
150	VLR	300	27998	188	200
150	VLR	6000	27998	203	200

## DASD 容量に関する考慮事項

与えられた分類アプリケーションに対して、ユーザーは、直接アクセス装置を混在させたものを指定できます。ユーザーのオペレーティング・システムによりサポートされているどの IBM DASD 装置でも作業データ・セットに使用できます。

最高のパフォーマンスを得るためには、作業データ・セット用に、エミュレートされた 3390-9 装置 (RAMAC など) やその他の高速の IBM DASD 装置を使用し、作業データ・セットに実 3390-9 装置を使用することは避けてください。

作業データ・セットを、トラックやブロックではなく、シリンダー単位で指定すると、システム・パフォーマンスが向上します。可能な場合は、一時作業データ・セット上の記憶域はシリンダー単位に再調整されます。直接アクセス装置に関するシリンダー当たりのトラック数を表 76 に示します。

表 76. 直接アクセス装置の 1 シリンダー当たりのトラック数

装置	トラック数 (1 シリンダー当たり)	使用される最大バイト 数 (1 トラック当たり)
3380	15	47476
3390	15	56664
9345	15	46456

WRKSEC が有効になっていて、作業データ・セットが VIO に割り振られていない場合は、JCL で要求されなくても、DFSORT は、必要に応じて 2 次エクステントを割り振ります。

## DASD ワークスペース容量の超過

分類処理中に、作業データ・セットのうちの 1 つで 2 次スペースの割り振りが失敗した場合は、システムは B37 通知メッセージを出します。DFSORT は、別の作業データ・セットで使用可能なものがあれば、それにスペースを割り振ることによりリカバリー処理を行うことができます。

DFSORT は、通常、JCL で要求されなくても、作業データ・セットの 2 次エクステントを割り振ります。これにより、ワークスペース容量を超える可能性が少なくなります。

DASD ワークスペースが分類処理を実行するのに不十分な場合は、DFSORT はメッセージを出して終了します。

## 磁気テープ容量に関する考慮事項

ユーザーのオペレーティング・システムによりサポートされている IBM 磁気テープ装置をすべてワークスペースに使用できます。ただし、DASD ではなく磁気テープ装置を作業データ・セットに使用すると、次の理由から大幅なパフォーマンスの低下を招きますので、その使用は避ける必要があります。

DFSORT には、3 つの異なるテープ作業データ・セット手法、すなわち、バランスド、ポリフェーズ、およびオッシレーティングが使用可能です。これらの所要量の計算方法については、表 77 を参照してください。

**注:** 『min』について入手する値は文字通り最小値です。たとえば、入力に DFSORT よりも効率の良いブロック化因数が使用されたり、入力がスパンされている場合は、さらに多くのワークスペースが必要になります。スペースの所要量についても、表 77 で要約しています。DFSORT はこれらの基準を使用して、最も適切なテープ手法を選択します。

表 77. 各種の磁気テープ手法のワークスペース必要量

磁気テープ手法	最大入力	必要なワークスペース域	作業域の最大数	コメント
バランスド・テープ (BALN)	15 ボリューム	テープ装置最小必要台数 = $2(V+1)^*$	32 ボリューム	4 台以上の作業用記憶域テープが提供されており、ファイル・サイズの指定がない場合に使用される。
ポリ・フェーズ・テープ (POLY)	1 ボリューム	磁気テープ装置最小必要台数 = 3	17 ボリューム	3 台の作業用記憶テープが使用できる場合に使用される。
オッシレーティング・テープ (Oscillating tape (OSCL))	15 ボリューム	Min = $V+2^*$ または 4 テープ装置のうち、いずれか大きいもの	17 ボリューム	ファイル・サイズを指定しなければならない。SORTIN を含むテープ装置を作業装置として使用することはできない。

**注:**

V = 入力ボリュームの数。ブロック化の入力ボリュームの数がワークスペースのブロック化と等しい。

## 磁気テープ・ワークスペース容量の超過

テープ作業データ・セットを使用する分類処理の開始時に、DFSORT は最大分類容量 (Nmax) を見積もり、メッセージ ICE038I を出します。詳細については、このメッセージの説明を参照してください。

メッセージ ICE038I に印刷される Nmax の値は、一番近い千の値に切り下げられた平均値です。この値はランダム入力を想定しています。逆順序ファイルとテープ・ワークスペースを使用する場合は、部分的に空の、ストリング終了ブロックが想定した数よりも多くなるため、分類容量が低い値で超過してしまうことがあります。

磁気テープの場合、Nmax の計算には、2400 フィートのテープ長が想定されます。それ以外の長さのテープの場合、その数字は正しくありません。記録密度が異なるテープが混在している場合は、最も低い記録密度が計算に使用されます。

実際のデータ・セット・サイズを指定し、そのサイズがプログラムで見積もった最大容量 (Nmax) よりも大きい場合は、プログラムは分類処理を始める前に終了します。見積データ・セット・サイズを指定するか、あるいは何も指定しなかった場合に、レコード数が最大値に達すると、E16 ユーザー出口ルーチンが作成され組み込まれていれば、プログラムはそのユーザー出口ルーチンに制御を与えます。このルーチンは、プログラムに次のいずれかの処置を取るよう指示できます。

- 使用可能なワークスペースを用いて、入力データ・セット全体の分類処理を続行すること。入力データ・セット・サイズの見積もりが高い場合は、アプリケーションを完了するために十分なワークスペースが残っている可能性があります。
- 入力データ・セットの一部だけを用いて分類処理を続行すること。残りを後で分類し、その 2 つの結果を組み合わせて、アプリケーションを完了します。
- それ以上の処理を行わずにプログラムを終了すること。

E16 ルーチンが組み込まれていない場合は、DFSORT はできる限りレコードの処理を続けます。入力データ・セットの全レコードを収容するだけの十分なワークスペースがある場合は、DFSORT は正常に完了します。ワークスペースが不十分な場合は、DFSORT はメッセージを出して終了します。

プログラムは、予想される 3 つのエラー条件に対して、それぞれ別個のメッセージを生成します。それらのメッセージは次のとおりです。

1. **ICE041A-N GT NMAX:** 正確なファイル・サイズが Nmax よりも大きいときに、分類処理が始まる前に生成されます。
2. **ICE046A-SORT CAPACITY EXCEEDED:** 分類処理が使用可能なすべてのワークスペースを使用してしまったときに生成されます。
3. **ICE048I-NMAX EXCEEDED:** 分類処理が Nmax の値を超過し、その処置のためにユーザー作成の E16 ルーチンへ制御を移したときに生成されます。

メッセージ ICE041A についてのテストを行う場合、できるだけ大きな計算値 (すなわち、DFSORT が必ず失敗するだけの値) を用いて行います。エラーが生じるかどうか疑わしい場合は、このメッセージは出されません。



## 付録 B. DFSORT オプションの指定 / 指定変更

18 ページの『インストール・デフォルト』では、DFSORT のインストール時 (ICEMAC) オプションおよび環境について説明し、ICETOOL の DEFAULTS 演算子を使用して、ユーザー・サイトで選択されたインストール・デフォルトをリストする方法を示しています。

以下に、IBM 提供のデフォルトを指定変更する各種のオプションを指定できる DFSORT 内の場所を示します。オプションのソースは、指定変更の順序でリストされています。つまり、このリストの上方に指定されているすべてのオプションは、それより下方に指定されているオプションを指定変更します。

### 直接呼び出し DFSORT

- DFSPARM データ・セット
  - PARM オプション
  - DEBUG および OPTION 制御ステートメント
  - その他の制御ステートメント
- EXEC ステートメントの PARM オプション
- SYSIN データ・セット
  - DEBUG および OPTION 制御ステートメント
  - その他の制御ステートメント
- インストール・マクロ (ICEMAC JCL または TSO)

### プログラム呼び出し DFSORT

- DFSPARM データ・セット
  - PARM オプション
  - DEBUG および OPTION 制御ステートメント
  - その他の制御ステートメント
- SORTCNTL データ・セット
  - DEBUG および OPTION 制御ステートメント
  - その他の制御ステートメント
- パラメーター・リスト
  - DEBUG および OPTION 制御ステートメント
  - その他の制御ステートメント
- インストール・マクロ (ICEMAC INV、TSOINV、または TDx)

### 注:

1. DEBUG および OPTION ステートメントの場合、指定変更はオプション・レベルで行われます。たとえば、

```
//DFSPARM DD *  
          OPTION EQUALS  
//SYSIN   DD *  
          OPTION NOEQUALS,SKIPREC=50
```

DFSPARM の EQUALS は、SYSIN の NOEQUALS を指定変更しますが、SYSIN の SKIPREC=50 は、DFSPARM の OPTION ステートメントによりは影響を受けません。したがって、EQUALS および SKIPREC=50 が使用されます。

## オプションの指定 / 指定変更

DEBUG および OPTION 以外の制御ステートメントに関しては、指定変更はステートメント・レベルで行われます。たとえば、

```
//DFSPARM DD *  
  MODS E15=(CHECK,4096,EXIT)  
//SYSIN DD *  
  MODS E35=(MOVE,2048,EXITX)
```

DFSPARM の MODS ステートメントは、SYSIN の MODS ステートメントを完全に指定変更します。したがって、E15 出口は使用されますが、E35 出口は使用されません。

2. EFS プログラムまたはインストール初期設定出口 (ICEIEXIT) ルーチンも、オプションを指定変更するために使用できます。ICEIEXIT の変更は、EFS プログラムにより行われるすべての対応する変更を指定変更します。
3. OUTFIL ステートメントの場合、指定変更は DD 名レベルで行われます。詳細については、307 ページの『OUTFIL ステートメントに関する注意事項』を参照してください。

---

## DFSORT オプションのソースの主な機能

IBM 提供の標準デフォルトを指定変更できるオプションのソースは 5 つあります。以下のリストを使用してそれぞれの主な機能を比較し、ユーザーにとって最も効率の良いソースを判断してください。

### DFSPARM データ・セット

- 直接呼び出しまたはプログラム呼び出しを使用します。
- 他のすべてのソースを指定変更します。
- SYSIN および SORTCNTL により無視される OPTION ステートメント・パラメーターも含め、すべての DFSORT プログラム制御ステートメントと、すべての EXEC PARM オプションを受け入れます。
- 注釈ステートメント、ブランク・ステートメント、および注記が使用できます。

### EXEC ステートメントの PARM オプション

- 直接呼び出しだけで使用します。
- SYSIN および SORTCNTL により無視される OPTION ステートメント・パラメーターと同等なものを含め、すべての EXEC PARM オプションを受け入れます。

### SORTCNTL データ・セット

- プログラム呼び出しだけで使用します。
- すべての DFSORT プログラム制御ステートメントを受け入れます。
- OPTION ステートメント・パラメーター EFS、LIST、NOLIST、LISTX、NOLISTX、LOCALE、MSGPRT、MSGDDN、SMF、SORTDD、SORTIN、SORTOUT、および USEWKDD を無視します。
- 注釈ステートメント、ブランク・ステートメント、および注記が使用できます。



- 複数のパラメーター・リストを使用して SORTCNTL データ・セットの名前変更を行って、DFSORT を複数回呼び出すプログラムに、異なる制御ステートメントを使用できます。

## SYSIN データ・セット

- 直接呼び出しだけで使用します。
- すべての DFSORT プログラム制御ステートメントを受け入れます。
- OPTION ステートメント・パラメーター EFS、LIST、NOLIST、LISTX、NOLISTX、LOCALE、MSGPRT、MSGDDN、SMF、SORTDD、SORTIN、SORTOUT、および USEWKDD を無視します。
- 注釈ステートメント、ブランク・ステートメント、および注記が使用できます。
- リンク・エディットのために、ユーザー出口ルーチンをオブジェクト・デッキ形式で含めることができます。

## パラメーター・リスト

- プログラム呼び出しだけで使用します。
- 拡張パラメーター・リストは、SYSIN および SORTCNTL により無視される OPTION ステートメント・パラメーターも含め、すべての DFSORT プログラム制御ステートメントを受け入れます。
- 24 ビット・パラメーター・リストは、DFSORT プログラム制御ステートメントのサブセットを受け入れます。
- 複数のパラメーター・リストを使用して SORTCNTL データ・セットの名前変更を行って、DFSORT を複数回呼び出すプログラムに、異なる制御ステートメントを使用できます。
- ユーザー・プログラムが主記憶域内に入れたユーザー出口のアドレスを受け渡すために使用できます。

注: 拡張パラメーター・リストは、24 ビット・パラメーター・リストの機能のスーパーセットを実行できます。

## 指定変更テーブル

以下に示すテーブルは、可能な指定のソースと個々のオプションの指定変更順序を示しています。

- 指定のソース間での指定変更順序は、左から右です。ある指定を行うと、それより右にある指定はすべて変更されます。
- ソース内での指定変更順序は、上から下です。ある指定を行うと、それより下にある指定はすべて変更されます。
- DFSPARM データ・セット内に指定できる EXEC PARM オプションは、テーブルの DFSPARM 欄で先頭に“PARM”という語を付けて表示し、制御ステートメントのオプションと区別しています。
- 機能欄は、各オプションを使用できる機能 (S=分類、M=組み合わせ、C=コピー)を示しています。
- 多くのオプションで別名を使用できますが、テーブルには示されていません。

## 直接呼び出し DFSORT

701 ページの表 78 は、DFSORT が直接呼び出される (すなわち、プログラムにより呼び出されない) 場合に、分類、組み合わせ、あるいはコピーのオプションをそれぞれどこに指定したらよいかを示しています。

**DFSPARM:** PARM オプションは、他のソースの対応するオプションを選択的に指定変更します。DEBUG および OPTION 制御ステートメント・オプションは、EXEC PARM および SYSIN の対応するオプションを選択的に指定変更します。DEBUG および OPTION 以外の制御ステートメントは、SYSIN の対応する制御ステートメントを完全に指定変更します。

**EXEC PARM** オプションは、SYSIN のオプションを選択的に指定変更します。

SORT および MERGE は、対応する制御ステートメントであると見なされます。

INCLUDE および OMIT は、対応する制御ステートメントと見なされます。

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更： オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSIN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
不可	不可	不可	ABCODE	ABEND コード	S,M,C
DEBUG ABSTP	不可	DEBUG ABSTP	不可	異常停止	S,M,C
ALTSEQ CODE	不可	ALTSEQ CODE	ALTSEQ	代替順序	S,M
PARAM ARESALL OPTION ARESALL	ARESALL	OPTION ARESALL	ARESALL	16 メガバイト境界より上のシステム記憶域	S,M,C
DEBUG NOASSIST	不可	DEBUG NOASSIST	不可	分類命令のバイパス	S
PARAM AVGRLEN OPTION AVGRLEN	AVGRLEN	OPTION AVGRLEN	不可	平均レコード長	S
PARAM BSAM DEBUG BSAM	BSAM	DEBUG BSAM	不可	BSAM の強制使用	S,M,C
DEBUG CFWINOCFW	不可	DEBUG CFWINOCFW	CFW	キャッシュ高速書き込み	S
OPTION CHALTINOCALT	不可	OPTION CHALTINOCALT	CHALT	CH フィールドの順序	S,M
OPTION CHECKINOCHECK	不可	OPTION CHECKINOCHECK	CHECK	レコード数の検査	S,M,C
PARAM CINVINOCINV OPTION CINVINOCINV	CINVINOCINV	OPTION CINVINOCINV	CINV	制御インターバル・アクセス	S,M,C
PARAM COBEXIT OPTION COBEXIT	COBEXIT	OPTION COBEXIT	COBEXIT	COBOL ライブラリー	S,M,C
INCLUDEIOMIT CONDIFORMAT	不可	INCLUDEIOMIT CONDIFORMAT	不可	組み込み / 除外フィールド	S,M,C
OPTION COPY SORTIMERGE FIELDS	不可	OPTION COPY SORTIMERGE FIELDS	不可	レコードのコピー	C

オプションの指定 / 指定変更

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSDN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
DEBUG CTRx	不可	DEBUG CTRx	不可	ABEND レコード・カウンタ	S,M
不可	不可	不可	活動化の時刻	SORTDIAG DD スタートメント のシミュレート	S,M,C
不可	不可	不可	DIAGSIM	SORTDIAG DD スタートメント のシミュレート	S,M,C
不可	不可	不可	DSA	動的ストレージ 調整の限界	S
PARM DSPSIZE OPTION DSPSIZE	DSPSIZE	OPTION DSPSIZE	DSPSIZE	データ空間分類	S
PARM DYNALLO OPTION DYNALLO SORT DYNALLO	DYNALLO	OPTION DYNALLO SORT DYNALLO	DYNALLO <sup>1</sup>	動的 SORTWK	S
PARM DYNALLO OPTION DYNALLOUSEWKDD SORT DYNALLO	DYNALLO	OPTION DYNALLO SORT DYNALLO	DYNAUTO	自動動的割り振り	S
PARM DYNNSPC OPTION DYNNSPC	DYNNSPC	OPTION DYNNSPC	DYNNSPC	動的割り振りの デフォルト・ス ペース	S
PARM EFS OPTION EFS	EFS	不可 <sup>2</sup>	EFS	EFS プログラム が指定済み	S,M,C
不可	不可	不可	ENABLE	使用可能な時刻 モジュール	S,M,C

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
PARM EQUALSINOEQUALS OPTION EQUALSINOEQUALS SORTMERGE EQUALSINOEQUALS	EQUALSINOEQUALS	OPTION EQUALSINOEQUALS SORTMERGE EQUALSINOEQUALS	EQUALS	照合結果が等しいレコードの順序	S,M
DEBUG EQUCOUNT	不可	DEBUG EQUCOUNT	不可	等しいキー・カウント・メッセージ	S
PARM ABENDINOABEND DEBUG ABENDINOABEND	ABENDINOABEND	DEBUG ABENDINOABEND	ERET	エラー処置	S,M,C
DEBUG ESTAEINOESTAE	不可	DEBUG ESTAEINOESTAE	ESTAE	ESTAE ルーチン	S,M,C
OPTION EXITCK	不可	OPTION EXITCK	EXITCK	E15/E35 戻りコード検査	S,M,C
不可	不可	不可	EXPMAX	すべての DFSORT ハイパースペースに使用可能な拡張記憶域の限界	S
不可	不可	不可	EXPOLD	すべてのハイパースペースの古い拡張記憶域の限界	S
不可	不可	不可	EXPRES	非ハイパー分類の使用に予約された使用可能な拡張記憶域	S

オプションの指定 / 指定変更

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSDN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
PARM E15=COB PARM E35=COB MODS ExxIHILEVEL=YES	E15=COB E35=COB	MODS ExxIHILEVEL=YES	不可	ユーザー出口 Exx (xx = 11, 15 ~ 19, 31, 35, 37 ~ 39, および 61)	S,M,C <sup>3</sup>
INREC FIELDS	不可	INREC FIELDS	不可	INREC フィールド	S,M,C
OUTREC FIELDS	不可	OUTREC FIELDS	不可	OUTREC フィールド	S,M,C
SORTMERGE FIELDSIFORMAT	不可	SORTMERGE FIELDSIFORMAT	不可	制御フィールド	S,M
SUM FIELDS/FORMAT	不可	SUM FIELDS/FORMAT	不可	合計フィールド	S,M
MERGE FILES	不可	MERGE FILES	不可	組み合わせ入力 ファイル	M
PARM FILSZ OPTION FILSZISIZE SORTMERGE FILSZISIZE	FILSZ	OPTION FILSZISIZE SORTMERGE FILSZISIZE	FSZEST	ファイル・サイズ	S,M
PARM HIPRMAX OPTION HIPRMAX	HIPRMAX	OPTION HIPRMAX	HIPRMAX	ハイパー分類	S
不可	不可	不可	IDRCPCT	IDRC 短縮	S
不可	不可	不可	IEEXIT	ICEEXIT	S,M,C
OPTION CKPT <sup>4</sup> SORT CKPT <sup>4</sup>	不可	OPTION CKPT <sup>4</sup> SORT CKPT <sup>4</sup>	IGNCKPT	チェックポイント	S

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSDN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
不可	不可	不可	IOMAXBF	最大 SORTIN/ SORTOUT データ・セット・バツファア・スペース	S,M,C
RECORD LENGTH	不可	RECORD LENGTH	不可	レコード長	S,M,C
PARM LISTINOLIST OPTION LISTINOLIST	LISTINOLIST	不可 <sup>2</sup>	LIST	DFSORT 制御ステートメントの印刷 <sup>5</sup>	S,M,C
PARM LISTXINOLIST OPTION LISTXINOLISTX	LISTXINOLISTX	不可 <sup>2</sup>	LISTX	EFS プログラムにより戻された制御ステートメントの印刷 <sup>5</sup>	S,M,C
PARM LOCALE OPTION LOCALE	LOCALE	不可 <sup>2</sup>	LOCALE	ロケール処理	S,M,C
不可	不可	不可	MAXLIM	16 メガバイト境界より下の最大記憶域 <sup>6</sup>	S,M,C
不可	不可	不可	MINLIM	最小記憶域	S,M,C
PARM MSGDDN OPTION MSGDDN	MSGDDN	不可 <sup>2</sup>	MSGDDN	代替メッセージ・データ・セット	S,M,C
不可	不可	不可	MSGCON	マスター・コンソールへのメッセージの書き出し	S,M,C

オプションの指定 / 指定変更

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSIN で指定される	ICEMAC JCL、TSO OR TDx で指定される	オプションの説明	機能
PARM MSGPRT OPTION MSGPRT	MSGPRT	不可 <sup>2</sup>	MSGPRT	メッセージの印刷	S,M,C
OPTION NOBLKSET	不可	OPTION NOBLKSET	不可	ブロック・セットのバイパス	S,M
不可	不可	不可	NOMSGDD	メッセージ・データ・セットが欠落している場合の処置	S,M,C
PARM ODMAXBF OPTION ODMAXBF	ODMAXBF	OPTION ODMAXBF	ODMAXBF	最大 OUTFIL データ・セット・バッファアークスペース	S,M,C
OUTFIL <sup>9</sup>	OUTFIL <sup>9</sup>	OUTFIL <sup>9</sup>	不可	OUTFIL 処理	S,M,C
PARM OUTRELNOOUTREL OPTION NOOUTREL	OUTRELNOOUTREL	OPTION NOOUTREL	OUTREL	出力データ・セット・スペースの解放	S,M,C
OPTION NOOUTSEC	不可	OPTION NOOUTSEC	OUTSEC	出力データ・セット 2 次割り振り	S,M,C
不可	不可	不可	OVERRGN	REGION を超える記憶域	S,M,C
PARM OVFO OPTION OVFO	OVFO	OPTION OVFO	OVFO	要約フィールドのオーバーフロー処置	S,M
PARM PAD OPTION PAD	PAD	OPTION PAD	PAD	DFSORT LRECL 埋め込み処置	S,M,C



表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSIN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
不可	不可	不可	PARMDDN	DFSPARM の代替 DD 名	S,M,C
PARM RESALL OPTION RESALL	RESALL	OPTION RESALL	RESALL	システム予約記憶域 <sup>6</sup>	S,M,C
PARM RESETNORESET OPTION RESETNORESET	RESETNORESET	OPTION RESETNORESET	RESET	NEW または MOD VSAM 出力	S,M,C
PARM SDB OPTION SDB	SDB	OPTION SDB	SDB	システム決定の出力データ・セット・ブロック・サイズ	S,M,C
不可	不可	不可	SDBMSG	メッセージおよびリスト・データ・セットのシステム決定ブロック・サイズ	S,M,C
PARM SIZE OPTION MAINSIZE	SIZE	OPTION MAINSIZE	SIZE	記憶域	S,M,C
PARM SKIPREC OPTION SKIPREC SORT SKIPREC	SKIPREC	OPTION SKIPREC SORT SKIPREC	不可	レコードのスキップ	S,C
OPTION SMF	不可	不可	SMF	SMF レコード	S,M,C
PARM SOLRFINOSLRF OPTION SOLRFINOSLRF	SOLRFINOSLRF	OPTION SOLRFINOSLRF	SOLRF	SORTOUT 長	S,M,C
OPTION SORTDD	不可	不可 <sup>2</sup>	不可	DD 名の接頭部	S,M,C

オプションの指定 / 指定変更

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSDN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
OPTION SORTIN <sup>7</sup>	不可	不可 <sup>2</sup>	不可	代替 SORTIN DD 名	S,C
不可	不可	不可	SORTLIB	従来のモジュール・ライブラリ	S,M
OPTION SORTOUT <sup>8</sup>	不可	不可 <sup>2</sup>	不可	代替 SORTOUT DD 名	S,M,C
PARM SPANINC OPTION SPANINC	SPANINC	OPTION SPANINC	SPANINC	不完全なスパン・レコードの処置	S,M,C
PARM STIMERINOSTIMER OPTION NOSTIMER	STIMERINOSTIMER	OPTION NOSTIMER	STIMER	STIMER の使用	S,M,C
PARM STOPAFT OPTION STOPAFT SORT STOPAFT	STOPAFT	OPTION STOPAFT SORT STOPAFT	不可	入力の制限	S,C
不可	不可	不可	SVC	DFSORT SVC 情報	S,M,C
PARM SZEROINOSZERO OPTION SZEROINOSZERO	SZEROINOSZERO	OPTION SZEROINOSZERO	SZERO	符号付きまたは符号なしのゼロ	S,M,C
不可	不可	不可	TEXTIT	ICETEXIT	S,M,C
不可	不可	不可	TMAXLIM	16 メガバイト境界より上および下の最大記憶域 <sup>6</sup> 。	S,M,C
PARM TRUNC OPTION TRUNC	TRUNC	OPTION TRUNC	TRUNC	DFSORT LRECL 切り捨て処置	S,M,C
RECORD TYPE	不可	RECORD TYPE	不可	レコード形式	S,M,C

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSDN で指定される	ICEMAC JCL, TSO OR TDx で指定される	オプションの説明	機能
PARM VERIFYNOVERIFY OPTION VERIFYNOVERIFY 不可	VERIFYNOVERIFY 不可	OPTION VERIFYNOVERIFY 不可	VERIFY VIO	シーケンス検査 SORTWK 仮想入出力	S,M S
PARM VLLONGINOVLLONG OPTION VLLONGINOVLLONG	VLLONGINOVLLONG	OPTION VLLONGINOVLLONG	VLLONG	長い出力レコードの切り捨て	S,M,C
PARM VLSCMPINOVLSOMP OPTION VLSCMPINOVLSOMP	VLSCMPINOVLSOMP	OPTION VLSCMPINOVLSOMP	VLSCMP	短い比較ファイルの埋め込み	S,M,C
PARM VLSHRTINOVLSHRT OPTION VLSHRTINOVLSHRT	VLSHRTINOVLSHRT	OPTION VLSHRTINOVLSHRT	VLSHRT	短い制御ファイルドまたは比較ファイルドの処理	S,M,C
不可	不可	不可	VSAMBSP	VSAM バッファオーバー・スペース	S
PARM VSAMEMTINVSAMEMT OPTION VSAMEMTINVSAMEMT	VSAMEMTINVSAMEMT	OPTION VSAMEMTINVSAMEMT	VSAMEMT	空の VSAM 入力データ	S,M,C
PARM VSAMIOINOVSAMIO OPTION VSAMIOINOVSAMIO	VSAMIOINOVSAMIO	OPTION VSAMIOINOVSAMIO	VSAMIO	同じ VSAM 入力および出力	S
PARM WRKRELINOWRKREL OPTION WRKRELINOWRKREL	WRKRELINOWRKREL	OPTION WRKRELINOWRKREL	WRKREL	SORTWK スペースの解放	S
PARM WRKSECINOWRKSEC OPTION WRKSECINOWRKSEC	WRKSECINOWRKSEC	OPTION WRKSECINOWRKSEC	WRKSEC	SORTWK の 2 次割り振り	S
PARM Y2PAST OPTION Y2PAST SORTIMERGE Y2PAST	Y2PAST	OPTION Y2PAST SORTIMERGE Y2PAST	Y2PAST	「世紀」ウィンドウの設定	S,M,C

## オプションの指定 / 指定変更

表 78. 直接呼び出し DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	EXEC PARM で指定される	SYSDN で指定される	ICEMAC JCL、TSO OR TDx で指定される	オプションの説明	機能
PARM ZDPRINTNZDPRINT OPTION ZDPRINTNZDPRINT	ZDPRINTNZDPRINT	OPTION ZDPRINTNZDPRINT	ZDPRINT	ZD SUM 結果	S,M

## 直接呼び出し DFSORT テーブルの注

- 1 動的割り振りを要求しません。デフォルトのみを提供します。
- 2 SYSIN では使用されません。
- 3 すべての機能がすべての出口に適用されるわけではありません。適用できる出口については、367 ページの表 48 および 367 ページの表 49 を参照してください。
- 4 ブロック・セットが選択され、かつ IGNCPT=YES が指定されている場合は使用されません。
- 5 MSGPRT=NONE が有効になっている場合は使用されません。その場合、制御ステートメントは印刷されません。
- 6 MAINSIZE=MAX が有効になっていない限り使用されません。
- 7 SORT 入力 DD 名の SORTDD を指定変更します。
- 8 SORT 出力 DD 名の SORTDD を指定変更します。
- 9 指定変更は DD 名レベルで行われます。

---

## 拡張パラメーター・リストを用いたプログラム呼び出し DFSORT

712 ページの表 79 は、DFSORT がプログラムにより呼び出され、拡張パラメーター・リストが渡される場合、分類、組み合わせ、またはコピーの各オプションをどこで指定すればよいかを示しています。

**DFSPARM:** PARM オプションは、他のソースの対応するオプションを選択的に指定変更します。DEBUG および OPTION 制御ステートメント・オプションは、SORTCNTL および パラメーター・リストの対応するオプションを選択的に指定変更します。DEBUG および OPTION 以外の制御ステートメントは、SORTCNTL およびパラメーター・リストの対応する制御ステートメントを完全に指定変更します。

**SORTCNTL:** DEBUG および OPTION 制御ステートメント・オプションは、パラメーター・リストの対応するオプションを選択的に指定変更します。DEBUG および OPTION 以外の制御ステートメントは、パラメーター・リストの対応する制御ステートメントを完全に指定変更します。

SORT および MERGE は、対応する制御ステートメントであると見なされます。

INCLUDE および OMIT は、対応する制御ステートメントと見なされます。

オプションの指定 / 指定変更

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更： オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
不可	不可	不可	ABCODE	ABEND コード	S,M,C
DEBUG ABSTP	DEBUG ABSTP	DEBUG ABSTP	不可	異常停止	S,M,C
ALTSEQ CODE	ALTSEQ CODE	オプション 16 の項目 ALTSEQ CODE	ALTSEQ	代替順序	S,M
PARM ARESALL OPTION ARESALL	OPTION ARESALL	OPTION ARESALL	ARESALL	16 メガバイト境界より上のシステム記憶域	S,M,C
OPTION ARESINV	OPTION ARESINV	OPTION ARESINV	ARESINV	呼び出しプログラムの 16 メガバイト境界より上の記憶域	S,M,C
DEBUG NOASSIST	DEBUG NOASSIST	DEBUG NOASSIST	不可	分類命令のバイパス	S
PARM AVGRLEN OPTION AVGRLEN	OPTION AVGRLEN	OPTION AVGRLEN	不可	平均レコード長	S
PARM BSAM DEBUG BSAM	DEBUG BSAM	DEBUG BSAM	不可	BSAM の強制使用	S,M,C
DEBUG CFWINOCFW	DEBUG CFWINOCFW	DEBUG CFWINOCFW	CFW	キャッシュ高速書き込み	S
OPTION CHALTINCHALT	OPTION CHALTINCHALT	OPTION CHALTINCHALT	CHALT	CH フィールドの順序	S,M
OPTION CHECKINOCHECK	OPTION CHECKINOCHECK	OPTION CHECKINOCHECK	CHECK	レコード数の検査	S,M,C
PARM CINVINOCINV OPTION CINVINOCINV	OPTION CINVINOCINV	OPTION CINVINOCINV	CINV	制御インターバル・アクセス	S,M,C
PARM COBEXIT OPTION COBEXIT	OPTION COBEXIT	OPTION COBEXIT	COBEXIT	COBOL ライブラリ	S,M,C
INCLUDEIOMIT CONDIFORMAT	INCLUDEIOMIT CONDIFORMAT	INCLUDEIOMIT CONDIFORMAT	不可	組み込み / 除外フィールド	S,M,C

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き): オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
OPTION COPY SORTMERGE FIELDS	OPTION COPY SORTMERGE FIELDS <sup>2</sup>	OPTION COPY SORTMERGE FIELDS	不可	レコードのコピー	C
DEBUG CTRx	DEBUG CTRx	DEBUG CTRx	不可	ABEND レコード・カウンタ	S,M
不可	不可	不可	日	活動化の時刻	S,M,C
不可	不可	不可	DIAGSIM	SORTDIAG DD ステートメントのシミュレート	S,M,C
不可	不可	不可	DSA	動的ストレージ調整の限界	S
PARM DSPSIZE OPTION DSPSIZE	OPTION DSPSIZE	OPTION DSPSIZE	DSPSIZE	データ空間分類	S
PARM DYNALLOC OPTION DYNALLOC SORT DYNALLOC	OPTION DYNALLOC SORT DYNALLOC <sup>2</sup>	OPTION DYNALLOC SORT DYNALLOC	DYNALLOC <sup>1</sup>	動的 SORTWK	S
PARM DYNALLOC OPTION DYNALLOC <sup>1</sup> USEWKDDSORT DYNALLOC	OPTION DYNALLOC SORT DYNALLOC	OPTION DYNALLOC <sup>1</sup> USEWKDDSORT DYNALLOC	DYNAUTO	自動的 DYNALLOC	S
PARM DYNNSPC OPTION DYNNSPC	OPTION DYNNSPC	OPTION DYNNSPC	DYNNSPC	動的割り振りのデフォルト・スペース	S
PARM EFS OPTION EFS	不可 <sup>3</sup>	OPTION EFS	EFS	EFS プログラムが指定済み	S,M,C
不可	不可	不可	ENABLE	使用可能な時刻モジュール	S,M,C

オプションの指定 / 指定変更

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM EQUALSINOEQUALS OPTION EQUALSINOEQUALS SORTMERGE EQUALSINOEQUALS	OPTION EQUALSINOEQUALS SORTMERGE EQUALSINOEQUALS <sup>2</sup>	OPTION EQUALSINOEQUALS SORTMERGE EQUALSINOEQUALS	EQUALS	照合結果が等しいレコードの順序	S,M
DEBUG EQUCOUNT	DEBUG EQUCOUNT	DEBUG EQUCOUNT	不可	等しいキー・カウン ト・メッセージ	S
PARM ABENDINOABEND DEBUG ABENDINOABEND	DEBUG ABENDINOABEND	DEBUG ABENDINOABEND	ERET	エラー処置	S,M,C
DEBUG ESTAEINOEESTAE OPTION EXITCK	DEBUG ESTAEINOEESTAE OPTION EXITCK	DEBUG ESTAEINOEESTAE OPTION EXITCK	ESTAE EXITCK	ESTAE ルーチン E15/E35 戻りコード 検査	S,M,C S,M,C
不可	不可	不可	EXPMAX	すべての DFSORT ハイパー空間に使用 可能な拡張記憶域 の限界	S
不可	不可	不可	EXPOLD	すべてのハイパー空 間用の古い拡張記憶 域の限界	S
不可	不可	不可	EXPRES	非ハイパー分類の使 用に予約された使用 可能な拡張記憶域	S
PARM E15=COB MODS E15 <sup>4</sup> HILEVEL=YES	MODS E15 <sup>4</sup> HILEVEL=YES	オフセット 4 の項目 <sup>4</sup> MODS E15 <sup>4</sup> HILEVEL=YES	不可	E15 出口	S,C
MODS E18 <sup>4</sup>	MODS E18 <sup>4</sup>	オフセット 24 の項目 <sup>4</sup> MODS E18 <sup>4</sup>	不可	E18 出口	S
不可	不可	オフセット 4 の項目	不可	E32 出口	M



表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き): オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM E35=COB MODS E35 <sup>4</sup> HILEVEL=YES	MODS E35 <sup>4</sup> HILEVEL=YES	オフセット 8 の項目 <sup>4</sup> MODS E35 <sup>4</sup> HILEVEL=YES	不可	E35 出口	S,M,C
MODS E39 <sup>4</sup>	MODS E39 <sup>4</sup>	オフセット 28 の項目 <sup>4</sup> MODS E39 <sup>4</sup>	不可	E39 出口	S,M,C
MODS Exx	MODS Exx	MODS Exx	不可	ユーザークーラー出口 Exx (xx = 11, 16, 17, 19, 31, 37, 38, および 61)	S,M,C <sup>5</sup>
INREC FIELDS	INREC FIELDS	INREC FIELDS	不可	INREC フィールド	S,M,C
OUTREC FIELDS	OUTREC FIELDS	OUTREC FIELDS	不可	OUTREC フィールド	S,M,C
SORTMERGE FIELDSFORMAT	SORTMERGE FIELDSFORMAT	SORTMERGE FIELDSFORMAT	不可	制御フィールド	S,M
SUM FIELDSFORMAT	SUM FIELDSFORMAT	SUM FIELDSFORMAT	不可	合計フィールド	S,M
MERGE FILES	MERGE FILES	MERGE FILES	不可	組み合わせ入力ファイル	M
PARM FILSZ OPTION FILSZSIZE SORTMERGE FILSZSIZE	OPTION FILSZSIZE SORTMERGE FILSZSIZE <sup>2</sup>	OPTION FILSZSIZE SORTMERGE FILSZSIZE	FSZEST	ファイル・サイズ	S,M
不可	不可	不可	GENER	IEBGENER 名	C
不可	不可	不可	GNPAD	ICEGENER LRECL 埋め込み処理	C
不可	不可	不可	GNTRUNC	ICEGENER LRECL 切り捨て処理	C
PARM HIPRMAX OPTION HIPRMAX	OPTION HIPRMAX	OPTION HIPRMAX	HIPRMAX	ハイパー分類	S
不可	不可	不可	IDRCPCT	IDRC 短縮	S
不可	不可	不可	IEEXIT	ICEEXIT	S,M,C

オプションの指定 / 指定変更

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
OPTION CKPT <sup>6</sup> SORTIMERGE CKPT <sup>6</sup> 不可	OPTION CKPT <sup>6</sup> SORTIMERGE CKPT <sup>2</sup> 、 <sup>6</sup> 不可	OPTION CKPT <sup>6</sup> SORTIMERGE CKPT <sup>6</sup> 不可	IGNCKPT  IOMAXBFB	チェックポイント  最大 SORTIN/SORTOUT データ・セット・パ ツファアー・スペース	S  S,M,C
RECORD LENGTH	RECORD LENGTH	RECORD LENGTH	不可	レコード長	S,M,C
PARM LISTINOLIST OPTION LISTINOLIST	不可 <sup>3</sup>	OPTION LISTINOLIST	LIST	DFSORT 制御ステ ートメントの印刷 <sup>7</sup>	S,M,C
PARM LISTXINOLISTX OPTION LISTXINOLISTX	不可 <sup>3</sup>	OPTION LISTXINOLISTX	LISTX	EFS プログラムによ り戻された制御ステ ートメントの印刷 <sup>7</sup>	S,M,C
PARM LOCALE OPTION LOCALE	不可 <sup>3</sup>	OPTION LOCALE	LOCALE	ロケール処理	S,M,C
不可	不可	不可	MAXLIM	16 メガバイト境界よ り下の最大記憶域 <sup>8</sup>	S,M,C
不可	不可	不可	MINLIM	最小記憶域	S,M,C
PARM MSGDDN OPTION MSGDDN	不可 <sup>3</sup>	OPTION MSGDDN	MSGDDN	代替メッセージ DD 名	S,M,C
不可	不可	不可	MSGCON	マスター・コンソー ルへのメッセージの 書き出し	S,M,C
PARM MSGPRT OPTION MSGPRT	不可 <sup>3</sup>	OPTION MSGPRT	MSGPRT	メッセージの印刷	S,M,C
OPTION NOBLKSET	OPTION NOBLKSET	OPTION NOBLKSET	不可	ブロック・セットの バイパス	S,M

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
不可	不可	不可	NOMSGDD	メッセージ・データ・セットが欠落している場合の処置	S,M,C
PARM ODMAXBFB OPTION ODMAXBFB	OPTION ODMAXBFB	OPTION ODMAXBFB	ODMAXBF	最大 OUTFIL データ・セット・バッファ・スペース	S,M,C
OUTFIL <sup>11</sup>	OUTFIL <sup>11</sup>	OUTFIL <sup>11</sup>	不可	OUTFIL 処理	S,M,C
PARM OUTRELNOUTREL OPTION NOOUTREL	OPTION NOOUTREL	OPTION NOOUTREL	OUTREL	出力データ・セット・スペースの解放	S,M,C
OPTION NOOUTSEC	OPTION NOOUTSEC	OPTION NOOUTSEC	OUTSEC	出力データ・セット 2 次割り振り	S,M,C
不可	不可	不可	OVERRGN	REGION を超える記憶域	S,M,C
PARM OVFL OPTION OVFL	OPTION OVFL	OPTION OVFL	OVFL	要約フィールドのオーバーフロー処置	S,M
PARM PAD OPTION PAD	OPTION PAD	OPTION PAD	PAD	DFSORT LRECL 処理 め込み処置	S,M,C
不可	不可	不可	PARMDDN	DFSPARM の代替 DD 名	S,M,C
PARM RESALL OPTION RESALL	OPTION RESALL	OPTION RESALL	RESALL	システム予約記憶域 <sup>8</sup>	S,M,C
PARM RESETNORESET OPTION RESETNORESET	OPTION RESETNORESET	OPTION RESETNORESET	RESET	NEW または MOD VSAM 出力	S,M,C
OPTION RESINV	OPTION RESINV	OPTION RESINV	RESINV	プログラム予約記憶域 <sup>8</sup>	S,M,C

オプションの指定 / 指定変更

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM SDB OPTION SDB	OPTION SDB	OPTION SDB	SDB	システム決定の出力データ・セット・ブロック・サイズ	S,M,C
不可	不可	不可	SDBMSG	メッセージおよびリスト・データ・セットのシステム決定ブロック・サイズ	S,M,C
PARM SIZE OPTION MAINSIZE	OPTION MAINSIZE	OPTION MAINSIZE	SIZE	記憶域	S,M,C
PARM SKIPREC OPTION SKIPREC SORTMERGE SKIPREC	OPTION SKIPREC SORTMERGE SKIPREC <sup>2</sup>	OPTION SKIPREC SORTMERGE SKIPREC	不可	レコードのスキップ	S,C
OPTION SMF	不可	OPTION SMF	SMF	SMF レコード	S,M,C
PARM SOLRFINOSOLRF OPTION SOLRFINOSOLRF	OPTION SOLRFINOSOLRF	OPTION SOLRFINOSOLRF	SOLRF	SORTOUT 長	S,M,C
OPTION SORTDD	不可 <sup>3</sup>	OPTION SORTDD	不可	DD 名の接頭部	S,M,C
OPTION SORTIN <sup>9</sup>	不可 <sup>3</sup>	OPTION SORTIN <sup>9</sup>	不可	代替 SORTIN DD 名	S,C
不可	不可	不可	SORTLIB	従来のモジュール・ライブラリー	S,M
OPTION SORTOUT <sup>10</sup>	不可 <sup>3</sup>	OPTION SORTOUT <sup>10</sup>	不可	代替 SORTOUT DD 名	S,M,C
PARM SPANINC OPTION SPANINC	OPTION SPANINC	OPTION SPANINC	SPANINC	不完全なスパン・レコードの処置	S,M,C
PARM STIMER/NOSTIMER OPTION NOSTIMER	OPTION NOSTIMER	OPTION NOSTIMER	STIMER	STIMER の使用	S,M,C

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM STOPAFT OPTION STOPAFT SORTMERGE STOPAFT 不可	OPTION STOPAFT SORTMERGE STOPAFT <sup>2</sup> 不可	OPTION STOPAFT SORTMERGE STOPAFT 不可	不可 SVC	入力の制限 DFSORT SVC 情報	S,C S,M,C
PARM SZEROINOSZERO OPTION SZEROINOSZERO 不可	OPTION SZEROINOSZERO 不可	OPTION SZEROINOSZERO 不可	SZERO TEXT	符号付きまたは符号なしのゼロ ICETEXIT	S,M,C S,M,C
不可	不可	不可	TMAXLIM	16 メガバイト境界より上および下の最大記憶域 <sup>8</sup>	S,M,C
PARM TRUNC OPTION TRUNC	OPTION TRUNC	OPTION TRUNC	TRUNC	DFSORT LRECL 切り捨て処置	S,M,C
RECORD TYPE PARM VERIFYNOVERIFY OPTION VERIFYNOVERIFY 不可	RECORD TYPE OPTION VERIFYNOVERIFY 不可	RECORD TYPE OPTION VERIFYNOVERIFY 不可	不可 VERIFY VIO	レコード形式 シーケンス検査 SORTWK 仮想入出力	S,M,C S,M S
PARM VLLONGINOVLLONG OPTION VLLONGINOVLLONG	OPTION VLLONGINOVLLONG	OPTION VLLONGINOVLLONG	VLLONG	長い出力レコードの切り捨て	S,M,C
PARM VLSCMPINOVLSCMP OPTION VLSCMPINOVLSCMP	OPTION VLSCMPINOVLSCMP	OPTION VLSCMPINOVLSCMP	VLSCMP	短い比較フィールドの埋め込み	S,M,C
PARM VLSHRTINOVLSHRT OPTION VLSHRTINOVLSHRT	OPTION VLSHRTINOVLSHRT	OPTION VLSHRTINOVLSHRT	VLSHRT	短い制御フィールドまたは比較フィールドの処置	S,M,C
不可	不可	不可	VSAMBSP	VSAM バックアワー・スペース	S

## オプションの指定 / 指定変更

表 79. 拡張パラメーター・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	拡張パラメーター・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM VSAMEMTINVSAMEMT OPTION VSAMEMTINVSAMEMT	OPTION VSAMEMTINVSAMEMT	OPTION VSAMEMTINVSAMEMT	VSAMEMT	空の VSAM 入力データ	S,M,C
PARM VSAMIOINOVSAMIO OPTION VSAMIOINOVSAMIO	OPTION VSAMIOINOVSAMIO	OPTION VSAMIOINOVSAMIO	VSAMIO	同じ VSAM 入力および出力	S
PARM WRKRELINOWRKREL OPTION WRKRELINOWRKREL	OPTION WRKRELINOWRKREL	OPTION WRKRELINOWRKREL	WRKREL	SORTWK スペースの解放	S
PARM WRKSECINOWRKSEC OPTION WRKSECINOWRKSEC	OPTION WRKSECINOWRKSEC	OPTION WRKSECINOWRKSEC	WRKSEC	SORTWK の 2 次割り振り	S
PARM Y2PAST OPTION Y2PAST SORTMERGE Y2PAST	OPTION Y2PAST SORTMERGE Y2PAST <sup>2</sup>	OPTION Y2PAST SORTMERGE Y2PAST	Y2PAST	「世紀」ウィンドウの設定	S,M,C
PARM ZDPRINTINZDPRINT OPTION ZDPRINTINZDPRINT	OPTION ZDPRINTINZDPRINT	OPTION ZDPRINTINZDPRINT	ZDPRINT	ZD SUM 結果	S,M

## 拡張パラメーター・リスト・テーブルの注

- 1 動的割り振りを要求しません。デフォルトのみを提供します。
- 2 拡張パラメーター・リストを介して指定された OPTION ステートメントの対応するオプションを指定変更しません。
- 3 SORTCNTL では使用されません。
- 4 出口がパラメーター・リスト項目を介して指定され、しかもその出口が MODS ステートメントで指定されている場合は、DFSORT は終了します。
- 5 すべての機能がすべての出口に適用されるわけではありません。適用できる出口については、367 ページの表 48 および 367 ページの表 49 を参照してください。
- 6 ブロック・セットが選択され、かつ IGNCKPT=YES が指定されている場合は使用されません。
- 7 MSGPRT=NONE が有効になっている場合は使用されません。その場合、制御ステートメントは印刷されません。
- 8 MAINSIZE=MAX が有効になっていない限り使用されません。
- 9 分類入力 DD 名の SORTDD を指定変更します。
- 10 分類出力 DD 名の SORTSDD を指定変更します。
- 11 指定変更は DD 名レベルで行われます。

---

## 24 ビット・パラメーター・リストを用いたプログラム呼び出し DFSORT

722 ページの表 80 は、DFSORT がプログラムにより呼び出され、24 ビット・パラメーター・リストが渡される場合、分類、組み合わせ、またはコピーのオプションをどこで指定すればよいかを示しています。

**DFSPARM:** PARM オプションは、他のソースの対応するオプションを選択的に指定変更します。DEBUG および OPTION 制御ステートメント・オプションは、SORTCNTL および パラメーター・リストの対応するオプションを選択的に指定変更します。DEBUG および OPTION 以外の制御ステートメントは、SORTCNTL およびパラメーター・リストの対応する制御ステートメントを完全に指定変更しません。

**SORTCNTL:** DEBUG 制御ステートメント・オプションは、パラメーター・リストの対応するオプションを選択的に指定変更します。DEBUG 以外の制御ステートメントは、パラメーター・リストの対応する制御ステートメントを完全に指定変更します。

SORT および MERGE は、対応する制御ステートメントであると見なされます。

INCLUDE および OMIT は、対応する制御ステートメントと見なされます。

オプションの指定 / 指定変更

表 80. 24 ビット・リスト DF SORT オプションの指定 / 指定変更： オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
不可	不可	不可	ABCODE	ABEND コード	S,M,C
DEBUG ABSTP	DEBUG ABSTP	DEBUG ABSTP	不可	異常停止	S,M,C
ALTSEQ CODE	ALTSEQ CODE	X'F6' の項目 ALTSEQ CODE	ALTSEQ	代替順序	S,M
PARM ARESALL OPTION ARESALL	OPTION ARESALL	不可	ARESALL	16 メガバイト境界より上のシステム記憶域	S,M,C
OPTION ARESINV	OPTION ARESINV	不可	ARESINV	呼び出しプログラム用の 16 メガバイト境界より上の記憶域	S,M,C
DEBUG NOASSIST	DEBUG NOASSIST	DEBUG NOASSIST	不可	分類命令のバイパス	S
PARM AVGRLEN OPTION AVGRLEN	OPTION AVGRLEN	不可	不可	平均レコード長	S
PARM BSAM DEBUG BSAM	DEBUG BSAM	DEBUG BSAM	不可	BSAM の強制使用	S,M,C
DEBUG CFWINOCFW	DEBUG CFWINOCFW	DEBUG CFWINOCFW	CFW	キャッシュ高速書き込み	S
OPTION CHALTNOCALT	OPTION CHALTNOCALT	不可	CHALT	CH フィールドの順序	S,M
OPTION CHECKINOCHECK	OPTION CHECKINOCHECK	不可	CHECK	レコード数の検査	S,M,C
PARM CINVINOCINV OPTION CINVINOCINV	OPTION CINVINOCINV	不可	CINV	制御インターバル・アクセス	S,M,C
PARM COBEXIT OPTION COBEXIT	OPTION COBEXIT	不可	COBEXIT	COBOL ライブラリー	S,M,C



表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
INCLUDEIOMIT CONDIFORMAT	INCLUDEIOMIT CONDIFORMAT	INCLUDEIOMIT CONDIFORMAT	不可	組み込み / 除外フイールド	S,M,C
OPTION COPY SORTMERGE FIELDS	OPTION COPY SORTMERGE FIELDS	SORTMERGE FIELDS	不可	レコードのコピー	C
DEBUG CTRx	DEBUG CTRx	DEBUG CTRx	不可	ABEND レコード・カウンタ	S,M
不可	不可	不可	日	活動化の時刻	S,M,C
不可	不可	不可	DIAGSIM	SORTDIAG DD ステートメントのシミュレート	S,M,C
不可	不可	不可	DSA	動的ストレージ調整の限界	S
PARM DSPSIZE OPTION DSPSIZE	OPTION DSPSIZE	不可	DSPSIZE	データ空間分類	S
PARM DYNALLO OPTION DYNALLO SORT DYNALLO	OPTION DYNALLO SORT DYNALLO	SORT DYNALLO	DYNALLO <sup>1</sup>	動的 SORTWK	S
PARM DYNALLO OPTION DYNALLOUSEWKDD SORT DYNALLO	OPTION DYNALLO SORT DYNALLO	SORT DYNALLO	DYNAUTO	自動的 DYNALLO	S
PARM DYNNSPC OPTION DYNNSPC	OPTION DYNNSPC	不可	DYNNSPC	動的割り振りのデフォルト・スペース	S
PARM EFS OPTION EFS	不可 <sup>2</sup>	不可	EFS	EFS プログラムが指定済み	S,M,C
不可	不可	不可	ENABLE	使用可能な時刻モジュール	S,M,C

オプションの指定 / 指定変更

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM EQUALSINOEQUALS OPTION EQUALSINOEQUALS SORTMERGE EQUALSI NOEQUALS	OPTION EQUALSINOEQUALS SORTMERGE EQUALSI NOEQUALS	SORTMERGE EQUALSINOEQUALS	EQUALS	照合結果が等しいレコードの順序	S,M
DEBUG EQUCOUNT	DEBUG EQUCOUNT	DEBUG EQUCOUNT	不可	等しいキー・カウント・メッセージ	S
PARM ABENDINOABEND DEBUG ABENDINOABEND	DEBUG ABENDINOABEND	DEBUG ABENDINOABEND	ERET	エラー処置	S,M,C
DEBUG ESTAEINOEESTAE OPTION EXITCK	DEBUG ESTAEINOEESTAE OPTION EXITCK	DEBUG ESTAEINOEESTAE 不可	ESTAE EXITCK	ESTAE ルーチン E15/E35 戻りコード検査	S,M,C S,M,C
不可	不可	不可	EXPMAX	すべての DFSORT ハイパー空間用に使用可能な拡張記憶の限界	S
不可	不可	不可	EXPOLD	すべてのハイパー空間用の古い拡張記憶の限界	S
不可	不可	不可	EXPRES	非ハイパー分類の使用に予約された使用可能な拡張記憶	S
PARM E15=COB MODS E15 <sup>3</sup> HILEVEL=YES	MODS E15 <sup>3</sup> HILEVEL=YES	オフセット 18 の項目 <sup>3</sup> MODS E15 <sup>3</sup> HILEVEL=YES	不可	E15 ユーザー出口	S,C
不可	不可	オフセット 18 の項目	不可	E32 ユーザー出口	M

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM E35=COB MODS E35 <sup>3</sup> HILEVEL=YES	MODS E35 <sup>3</sup> HILEVEL=YES	オフセット 22 の項目 <sup>3</sup> MODS E35 <sup>3</sup> HILEVEL=YES	不可	E35 ユーザー出口	S,M,C
MODS Exx	MODS Exx	MODS Exx	不可	ユーザー出口 Exx (xx = 11、16 ~ 19、31、37 ~ 39、および 61)	S,M,C <sup>4</sup>
INREC FIELDS	INREC FIELDS	INREC FIELDS	不可	INREC フィールド	S,M,C
OUTREC FIELDS	OUTREC FIELDS	OUTREC FIELDS	不可	OUTREC フィールド	S,M,C
SORTMERGE FIELDSFORMAT	SORTMERGE FIELDSFORMAT	SORTMERGE FIELDSFORMAT	不可	制御フィールド	S,M,C
SUM FIELDSFORMAT	SUM FIELDSFORMAT	SUM FIELDSFORMAT	不可	合計フィールド	S,M
MERGE FILES	MERGE FILES	X'04' の項目 MERGE FILES	不可	組み合わせ入力ファイル	M
PARM FILSZ OPTION FILSZISIZE SORTMERGE FILSZISIZE	OPTION FILSZISIZE SORTMERGE FILSZISIZE	SORTMERGE FILSZISIZE	FSZEST	ファイル・サイズ	S,M
不可	不可	不可	GENER	IEBGENER 名	C
不可	不可	不可	GNPAD	ICEGENER LRECL 埋め込み 処置	C
不可	不可	不可	GNTRUNC	ICEGENER LRECL 切り捨て 処置	C
PARM HIPRMAX OPTION HIPRMAX	OPTION HIPRMAX	不可	HIPRMAX	ハイパー分類	S
不可	不可	不可	IDRC PCT	IDRC 短縮	S

オプションの指定 / 指定変更

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
不可	不可	不可	IEXIT	ICEIEXIT	S,M,C
OPTION CKPT <sup>5</sup> SORTIMERGE CKPT <sup>5</sup>	OPTION CKPT <sup>5</sup> SORTIMERGE CKPT <sup>5</sup>	SORTIMERGE CKPT <sup>5</sup>	IGNCKPT	チェックポイント	S
不可	不可	不可	IOMAXBF	最大 SORTIN/SORTOUT データ・セット・ バッファアワー・スベ ース	S,M,C
RECORD LENGTH	RECORD LENGTH	RECORD LENGTH	不可	レコード長	S,M,C
PARM LISTINOLIST OPTION LISTINOLIST	不可 <sup>2</sup>	不可	LIST	DFSORT 制御ステ ートメントの印刷 。 <sup>6</sup>	S,M,C
PARM LISTXINOLISTX OPTION LISTXINOLISTX	不可 <sup>2</sup>	不可	LISTX	EFS プログラムに より戻された制御 ステートメントの 印刷。 <sup>6</sup>	S,M,C
PARM LOCALE OPTION LOCALE	不可 <sup>2</sup>	不可	LOCALE	ローカル処理	S,M,C
不可	不可	不可	MAXLIM	16 メガバイト境界 より下の最大記憶 域。 <sup>7</sup>	S,M,C
不可	不可	不可	MINLIM	最小記憶域	S,M,C
PARM MSGDDN OPTION MSGDDN	不可 <sup>2</sup>	X'03' の項目	MSGDDN	代替メッセージ DD 名	S,M,C
不可	不可	不可	MSGCON	マスター・コンソ ールへのメッセージ の書き出し	S,M,C

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM MSGPRT OPTION MSGPRT	不可 <sup>2</sup>	XFF の項目	MSGPRT	メッセージの印刷	S,M,C
OPTION NOBLKSET	OPTION NOBLKSET	不可	不可	ブロック・セットのバイパス	S,M
不可	不可	不可	NOMSGDD	メッセージ・データ・セットが欠落している場合の処置	S,M,C
PARM ODMAXBF OPTION ODMAXBF	OPTION ODMAXBF	不可	ODMAXBF	最大 OUTFIL データ・セット・バイパス	S,M,C
OUTFIL <sup>10</sup>	OUTFIL <sup>10</sup>	OUTFIL <sup>10</sup>	不可	OUTFIL 処理	S,M,C
PARM OUTRELNOOUTREL OPTION NOOUTREL	OPTION NOOUTREL	不可	OUTREL	出力データ・セット・スペースの解放	S,M,C
OPTION NOOUTSEC	OPTION NOOUTSEC	不可	OUTSEC	出力データ・セット 2 次割り振り	S,M,C
不可	不可	不可	OVERRGN	REGION を超える記憶域	S,M,C
PARM OVFO OPTION OVFO	OPTION OVFO	不可	OVFO	要約ファイルのオーバーフロー処置	S,M
PARM PAD OPTION PAD	OPTION PAD	不可	PAD	DFSORT LRECL 埋め込み処置	S,M,C
不可	不可	不可	PARMDDN	DFSPARM の代替 DD 名	S,M,C

オプションの指定 / 指定変更

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは、ICESMAC 欄にアルファベット順に並べられています。『不可』が ICESMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICESMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM RESALL OPTION RESALL	OPTION RESALL	不可	RESALL	システム予約記憶域 <sup>7</sup>	S,M,C
PARM RESETNORESET OPTION RESETNORESET	OPTION RESETNORESET	不可	RESET	NEW または MOD VSAM 出力	S,M,C
OPTION RESINV	OPTION RESINV	X'01' の項目	RESINV	プログラム予約記憶域 <sup>7</sup>	S,M,C
PARM SDB OPTION SDB	OPTION SDB	不可	SDB	システム決定の出力データ・セット・ブロック・サイズ	S,M,C
不可	不可	不可	SDBMSG	メッセージおよびリスト・データ・セットのシステム決定ブロック・サイズ	S,M,C
PARM SIZE OPTION MAINSIZE	OPTION MAINSIZE	X'00' の項目	SIZE	記憶域	S,M,C
PARM SKIPREC OPTION SKIPREC SORTMERGE SKIPREC	OPTION SKIPREC SORTMERGE SKIPREC	SORTMERGE SKIPREC	不可	レコードのスキップ	S,C
OPTION SMF	不可	不可	SMF	SMF レコード	S,M,C
PARM SOLRFINOSOLRF OPTION SOLRFINOSOLRF	OPTION SOLRFINOSOLRF	不可	SOLRF	SORTOUT 長	S,M,C
OPTION SORTDD	不可 <sup>2</sup>	接頭部項目	不可	DD 名の接頭部	S,M,C
OPTION SORTIN <sup>8</sup>	不可 <sup>2</sup>	不可	不可	代替 SORTIN DD 名	S,C

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
不可	不可	不可	SORTLIB	従来のモジュール・ライブラリー	S,M
OPTION SORTOUT <sup>9</sup>	不可 <sup>2</sup>	不可	不可	代替 SORTOUT DD 名	S,M,C
PARM SPANINC OPTION SPANINC	OPTION SPANINC	不可	SPANINC	不完全なスパン・レコードの処置	S,M,C
PARM STIMERINOSTIMER OPTION NOSTIMER	OPTION NOSTIMER	不可	STIMER	STIMER の使用	S,M,C
PARM STOPAFT OPTION SORT OPTION STOPAFT SORTMERGE STOPAFT	OPTION STOPAFT SORTMERGE STOPAFT	SORTMERGE STOPAFT	不可	入力の制限	S,C
不可	不可	不可	SVC	DFSORT SVC 情報	S,M,C
PARM SZERONOSZERO OPTION SZERONOSZERO	OPTION SZERONOSZERO	不可	SZERO	符号付きまたは符号なしのゼロ	S,M,C
不可	不可	不可	TEXTIT	ICETEXIT	S,M,C
不可	不可	不可	TMAXLIM	16 メガバイト境界より上および下の最大記憶域 <sup>7</sup>	S,M,C
PARM TRUNC OPTION TRUNC	OPTION TRUNC	不可	TRUNC	DFSORT LRECL 切り捨て処置	S,M,C
RECORD TYPE	RECORD TYPE	RECORD TYPE	不可	レコード形式	S,M,C
PARM VERIFYINVERIFY OPTION VERIFYINVERIFY	OPTION VERIFYINVERIFY	不可	VERIFY	シーケンス検査	S,M
不可	不可	不可	VIO	SORTWK 仮想入出力	S

オプションの指定 / 指定変更

表 80. 24 ビット・リスト DFSORT オプションの指定 / 指定変更 (続き) : オプションは ICEMAC 欄にアルファベット順に並べられています。『不可』が ICEMAC 欄に指定されている場合は、次の欄を左方へ見ていきます。

指定変更順序は、行内で左から右、そして上から下になります。

DFSPARM で指定される	SORTCNTL で指定される	24 ビット・リストで指定される	ICEMAC INV、TSOINV、または TDx で指定する	オプションの説明	機能
PARM VLLONGINOVLLONG OPTION VLLONGINOVLLONG	OPTION VLLONGINOVLLONG	不可	VLLONG	長い出力レコードの切り捨て	S,M,C
PARM VLSCMPINOVLSOMP OPTION VLSCMPINOVLSOMP	OPTION VLSCMPINOVLSOMP	不可	VLSCMP	短い比較ファイルの埋め込み	S,M,C
PARM VLSHRTINOVLSHRT OPTION VLSHRTINOVLSHRT	OPTION VLSHRTINOVLSHRT	不可	VLSHRT	短い制御ファイルまたは比較ファイルの処置	S,M,C
不可	不可	不可	VSAMBSP	VSAM バックアップ・スペース	S
PARM VSAMEMTINVSAMEMT OPTION VSAMEMTINVSAMEMT	OPTION VSAMEMTINVSAMEMT	不可	VSAMEMT	空の VSAM 入力データ	S,M,C
PARM VSAMIOINOVSAMIO OPTION VSAMIOINOVSAMIO	OPTION VSAMIOINOVSAMIO	不可	VSAMIO	同じ VSAM 入力および出力	S
PARM WRKRELINOWRKREL OPTION WRKRELINOWRKREL	OPTION WRKRELINOWRKREL	不可	WRKREL	SORTWK スペースの解放	S
PARM WRKSECINOWRKSEC OPTION WRKSECINOWRKSEC	OPTION WRKSECINOWRKSEC	不可	WRKSEC	SORTWK の 2 次割り振り	S
PARM Y2PAST OPTION Y2PAST SORTMERGE Y2PAST	OPTION Y2PAST SORTMERGE Y2PAST	SORTMERGE Y2PAST	Y2PAST	「世紀」ウィンドウの設定	S,M,C
PARM ZDPRINTINZDPRINT OPTION ZDPRINTINZDPRINT	OPTION ZDPRINTINZDPRINT	不可	ZDPRINT	ZD SUM 結果	S,M



## 24 ビット・リスト・テーブルの注

- 1 動的割り振りを要求しません。デフォルトのみを提供します。
- 2 SORTCNTL では使用されません。
- 3 出口がパラメーター・リスト項目を介して指定され、かつ MODS ステートメントにユーザー出口が指定されている場合は、DFSORT は終了します。
- 4 すべての機能がすべてのユーザー出口に適用されるわけではありません。適用できるユーザー出口については、367 ページの表 48 および 367 ページの表 49 を参照してください。
- 5 ブロック・セットが選択され、かつ IGNCKPT=YES が指定されている場合は使用されません。
- 6 MSGPRT=NONE または MSGPRT=CRITICAL が有効になっている場合は使用されません。その場合、制御ステートメントは印刷されません。
- 7 MAINSIZE=MAX が有効になっていない限り使用されません。
- 8 分類入力 DD 名の SORTDD を指定変更します。
- 9 SORTDD の分類出力 DD 名を変更します。
- 10 指定変更は DD 名レベルで行われます。

## オプションの指定 / 指定変更

## 付録 C. データ形式記述

形式記述とは、IBM システム /390 で使用される、アセンブル済みデータ形式を指します。たとえば、あるデータ変数が PL/I で FIXED DECIMAL と宣言された場合は、その変数は、SORT 制御ステートメントの 'F' フィールドで指定する必要のあるコンパイル済みの形式の変数であって、PL/I により宣言された形式の変数ではありません。この場合は、PL/I コンパイラーが固定小数点 10 進数をパック 10 進数形式に変換するため、'F' フィールドは PD (パック 10 進数) になります。

形式	説明																				
CH	<p>(EBCDIC 文字、符号なし)。各文字は、8 ビットの EBCDIC コードで表記されます。</p> <p>例: AB7 は、次のようになります。</p> <table><tr><td>C1</td><td>C2</td><td>F7</td><td>Hexadecimal</td></tr><tr><td>11000001</td><td>11000010</td><td>11110111</td><td>Binary</td></tr></table> <p>注:</p> <ol style="list-style-type: none"><li>1. CHALT が有効になっている場合は、形式 CH フィールドは有効な ALTSEQ (代替照合順序) テーブルに従って照合されます。AQ 形式を同じ目的に使用できます。</li><li>2. ロケール処理が有効になっている場合は、形式 CH フィールドは、活動ロケールの照合規則に従って照合されます。</li></ol>	C1	C2	F7	Hexadecimal	11000001	11000010	11110111	Binary												
C1	C2	F7	Hexadecimal																		
11000001	11000010	11110111	Binary																		
ZD	<p>(ゾーン 10 進数、符号あり)。10 進数の各桁は、8 ビットの EBCDIC 表示に変換されます。符号標識は、その数の下位バイトの最初の 4 ビットを置き換えます。</p> <p>例: -247 は、次のようになります。</p> <table><tr><td>2</td><td>4</td><td>-7</td><td>10 進数</td></tr><tr><td>F2</td><td>F4</td><td>D7</td><td>16 進数</td></tr><tr><td>11110010</td><td>11110100</td><td>11010111</td><td>2 進数</td></tr></table> <p>数 +247 は、次のようになります。</p> <table><tr><td>F2</td><td>F4</td><td>C7</td><td></td></tr><tr><td>11110010</td><td>11110100</td><td>11000111</td><td></td></tr></table> <p>注:</p> <ol style="list-style-type: none"><li>1. 次のものは正符号標識として扱われます: F,E,C,A,8,6,4,2,0</li><li>2. 次のものは負符号標識として扱われます: D,B,9,7,5,3,1</li><li>3. SUM 処理の場合、符号の 0 ~ 9 または数字の A ~ F は、データ例外 (0C7 ABEND) になります。たとえば、3.5 (X'F34BF5') という ZD 値は、0C7 になります。B は無効な数字として扱われるからです。ICETOOL の DISPLAY または VERIFY 演算子を使用して、無効な数字をもつ ZD 値を識別できます。ICETOOL の VERIFY 演算子を使用して、無効な符号をもつ ZD 値を識別できます。</li><li>4. 最後の数字の最初の 4 ビットは符号標識です。その他の各数字の最初の 4 ビットは無視されます。したがって、先行ブランク (X'40') と数字 0 (X'F0') は等しいため、EBCDIC スtring の '0025' と ' 25' は両方とも、25 として扱われます。</li></ol>	2	4	-7	10 進数	F2	F4	D7	16 進数	11110010	11110100	11010111	2 進数	F2	F4	C7		11110010	11110100	11000111	
2	4	-7	10 進数																		
F2	F4	D7	16 進数																		
11110010	11110100	11010111	2 進数																		
F2	F4	C7																			
11110010	11110100	11000111																			

## データ形式の例

形式	説明
PD	<p>(パック 10 進数、符号あり)。10 進数の各桁は、4 ビットの 2 進数に変換されます。符号標識は、その数の右端の 4 ビットに入られます。</p> <p>例: -247 は、次のようになります。</p> <pre> 2 4 7-      10 進数 2 4 7D      16 進数 00100100 01111101  2 進数 </pre> <p>数 +247 は、16 進数の 247C になります。</p> <p>注:</p> <ol style="list-style-type: none"> <li>1. 次のものは正符号標識として扱われます: F,E,C,A,8,6,4,2,0</li> <li>2. 次のものは負符号標識として扱われます: D,B,9,7,5,3,1</li> <li>3. SUM 処理の場合、符号の 0 ~ 9 または数字の A ~ F は、データ例外 (0C7 ABEND) になります。たとえば、X'0123BF' という PD 値は、0C7 になります。B が無効な数字として扱われるためです。ICETOOL の DISPLAY または VERIFY 演算子を使用して、無効な数字をもつ PD 値を識別できます。ICETOOL の VERIFY 演算子を使用して、無効な符号をもつ PD 値を識別できます。</li> </ol>
PD0	<p>(パック 10 進数、符号と最初の桁は無視される) PD0 形式は、次のように表記されます。</p> <pre> xddd...ds </pre> <p>x は 16 進数 0 ~ F で、無視されます。  d は 16 進数 0 ~ 9 で、10 進数字を表示します。  s は 16 進数 0 ~ F で、無視されます。</p> <p>PD0 は、PD フィールドをいくつかの部分に分けて使用されます。たとえば、PD フィールド P'mmddy' (16 進数 0mmddyC) は、PD0 では 0mmd (mm)、mddy (dd)、および ddyC (yy) のように分けて使用されます。</p>
FI	<p>(固定小数点、符号あり)。完全数は、対応する 2 進数で表示され、符号標識は最も有効なビット位置に入られます。</p> <p>+ の場合は 0、- の場合は 1 です。負の数は 2 の補数形式で表されます。</p> <p>例: +247 はハーフワード形式になります。</p> <pre> 00F7      16 進数 0000000011110111  2 進数 </pre> <p>数 -247 は、次のようになります。</p> <pre> FF09      16 進数 </pre>
BI	<p>(2 進数、符号なし)。任意のビット・パターン。</p>
FL	<p>(浮動小数点、符号あり)。指定された数は、指数部と小数部の 2 つの形式に分かれ、符号標識はビット位置 0 にあります。</p> <p>例: +247 は、次のようになります。</p> <pre> 0 1000010 111101110000000..... + 指数部  小数部 </pre> <p>-247 は、符号ビットが 1 に変わる以外は同じです。</p>
AC	<p>(文字 ASCII、符号なし)。この形式は CH に似ていますが、文字が ASCII コードで表記されます。</p> <p>例: AB7 は、次のようになります。</p> <pre> 41      42      37      16 進数 01000001 01000010 00110111  2 進数 (ASCII コード) </pre>

形式	説明																														
CSF または FS	<p>(任意指定の左端の浮動符号をもつ符号付き数字)</p> <p>浮動符号形式は、次のように表すことができます。</p> <pre>&lt;s&gt;d . . .d</pre> <p>s は任意指定の符号で、数字 d のすぐ左にあります。s が - の場合は、その数は負として扱われます。- でない場合は、正として扱われます。したがって、負符号には - を使用する必要がありますが、正符号にはその他の任意の文字 (たとえば、+ またはブランク) を使用できます。右から左にみて最初の非 10 進数字 (すなわち、0 ~ 9 以外) は符号として扱われ、符号より左にある桁はすべて無視されます。</p> <p>Examples: Value: Treated as:</p> <table border="0"> <tr><td>34</td><td>+34</td></tr> <tr><td>+34</td><td>+34</td></tr> <tr><td>00034</td><td>+34</td></tr> <tr><td>-003</td><td>-3</td></tr> <tr><td>-1234</td><td>-1234</td></tr> <tr><td>1234</td><td>+1234</td></tr> <tr><td>+01234</td><td>+1234</td></tr> <tr><td>0</td><td>+0</td></tr> </table> <p>CSF/FS 形式により処理されるデータのタイプは、下記に示すように (図示のために 4 の幅を使用)、いくつかの異なる FORTRAN、PL/I、および COBOL 形式により作成されるデータ・タイプも含んでいます。</p> <ul style="list-style-type: none"> <li>* FORTRAN: I4 ; G4.0 ; SP,I4 ; SP,I4.3 ; S,I4.3</li> <li>* PL/I: F(4) ; P'S999' ; P'SSS9' ; P'---9'</li> <li>* COBOL: PIC +++9 ; PIC +999 ; PIC ++++ ; PIC ---9 ; PIC ---- ; PIC ZZZZ</li> </ul> <p>CSF/FS 形式フィールドは他の形式よりも処理が効率的ではないため、他の形式が該当する場合は (たとえば、CSL)、CSF/FS を使用できません。</p>	34	+34	+34	+34	00034	+34	-003	-3	-1234	-1234	1234	+1234	+01234	+1234	0	+0														
34	+34																														
+34	+34																														
00034	+34																														
-003	-3																														
-1234	-1234																														
1234	+1234																														
+01234	+1234																														
0	+0																														
CSL または LS	<p>(符号付き数字、左端の別個の符号) この形式は、カードにパンチされ、その後、EBCDIC コードにアセンブルされた 10 進数データを指します。</p> <p>例: カードにパンチされた +247 は、次のようになります。</p> <table border="0"> <tr> <td>+</td> <td>2</td> <td>4</td> <td>7</td> <td>パンチされた数字データ</td> </tr> <tr> <td>4E</td> <td>F2</td> <td>F4</td> <td>F7</td> <td>16 進数</td> </tr> <tr> <td>01001110</td> <td>11110010</td> <td>11110100</td> <td>11110111</td> <td>2 進 EBCDIC コード</td> </tr> </table> <p>-247 は、次のようになります。</p> <table border="0"> <tr> <td>-</td> <td>2</td> <td>4</td> <td>7</td> <td>パンチされた数字データ</td> </tr> <tr> <td>60</td> <td>F2</td> <td>F4</td> <td>F7</td> <td>16 進数</td> </tr> <tr> <td>01100000</td> <td>11110010</td> <td>11110100</td> <td>11110111</td> <td>2 進 EBCDIC コード</td> </tr> </table>	+	2	4	7	パンチされた数字データ	4E	F2	F4	F7	16 進数	01001110	11110010	11110100	11110111	2 進 EBCDIC コード	-	2	4	7	パンチされた数字データ	60	F2	F4	F7	16 進数	01100000	11110010	11110100	11110111	2 進 EBCDIC コード
+	2	4	7	パンチされた数字データ																											
4E	F2	F4	F7	16 進数																											
01001110	11110010	11110100	11110111	2 進 EBCDIC コード																											
-	2	4	7	パンチされた数字データ																											
60	F2	F4	F7	16 進数																											
01100000	11110010	11110100	11110111	2 進 EBCDIC コード																											
CST または TS	<p>(符号付き数字、右端の別個の符号) これは、符号標識が数字の後にパンチされる以外は、CSL 形式と同じ表示です。</p> <p>例: カードにパンチされた 247+ は、次のようになります。</p> <table border="0"> <tr> <td>F2</td> <td>F4</td> <td>F7</td> <td>4E</td> <td>16 進数</td> </tr> </table>	F2	F4	F7	4E	16 進数																									
F2	F4	F7	4E	16 進数																											

## データ形式の例

形式	説明
CLO <sup>1</sup> または OL <sup>1</sup>	<p>(符号付き数字、左端のオーバーパンチ符号) この形式も、カードにパンチされ、その後、EBCDIC コードにアSEMBルされた 10 進数データを指します。ただし、符号標識はその数の最初の 10 進数にオーバーパンチされます。</p> <p>例: 2 にオーバーパンチされた +247 は、次のようになります。</p> <pre>+2      4      7      パンチされた数字データ C2      F4      F7      16 進数 11000010 11110100 11110111  2 進 EBCDIC コード</pre> <p>同様に、-247 は次のようになります。</p> <pre>D2      F4      F7</pre>
CTO または OT	<p>(符号付き数字、右端のオーバーパンチ符号) この形式は、符号標識がその数字の最後の 10 進数にオーバーパンチされている以外は、CLO 形式の場合と同じ表示になります。</p> <p>例: 7 にオーバーパンチされた +247 は、次のようになります。</p> <pre>F2 F4 C7  16 進数</pre>
ASL	<p>(符号付き ASCII 数字、左端の別個の符号) 10 進データが ASCII コードにアSEMBルされる以外は、CSL 形式と同じです。</p> <p>例: カードにパンチされた +247 は、次のようになります。</p> <pre>+      2      4      7      パンチされた数字データ 2B     32     34     37     16 進数 0101011 00110010 00110100 00110111  2 進 ASCII コード</pre> <p>同様に、-247 は次のようになります。</p> <pre>2D 32 34 37  16 進数</pre>
AST	<p>(符号付き ASCII 数字、右端の別個の符号) これは、符号が数字の後にパンチされる以外は、ASL 形式と同じビット表示になります。</p> <p>例: 247+ は、次のようになります。</p> <pre>32 34 37 2B  16 進数</pre>
Y2T	<p>(文字またはゾーン 10 進数の yyx、yyxx、yyxxx、および yyxxxx、特殊標識付きの完全な日付形式)</p> <p>日付フィールドは、次のように表記できます。</p> <pre>3,Y2T: C'yyx' または Z'yyx' 4,Y2T: C'yyxx' または Z'yyxx' 5,Y2T: C'yyxxx' または Z'yyxxx' 6,Y2T: C'yyxxxx' または Z'yyxxxx'</pre> <p>y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ 9 で、年以外の桁を表します。x...x は、正しい照合順序である必要があります。</p> <p>特殊標識は、X'00...00' (BI ゼロ)、X'40...40' (ブランク)、C'0...0' (CH ゼロ)、Z'0...0' (ZD ゼロ)、C'9...9' (CH の 9)、Z'9...9' (ZD の 9)、および X'FF...FF' (BI の 1) です。</p>
Y2U	<p>(パック 10 進数の yyx および yyxxx、特殊標識付きの完全な日付形式)</p> <p>日付フィールドは、次のように表記できます。</p> <pre>2,Y2U: P'yyx' (X'yyxs') 3,Y2U: P'yyxxx' (X'yyxxxs')</pre> <p>y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ 9 で、年以外の桁を表します。s は 16 進数 0 ~ F で、無視されます。xxx は、正しい照合順序である必要があります。</p> <p>特殊標識は、P'0...0' (PD ゼロ) および P'9...9' (PD の 9) です。</p>

形式	説明
Y2V	<p>(パック 10 進数の yyxx および yyxxxx、特殊標識付きの完全な日付形式)</p> <p>日付フィールドは、次のように表記できます。</p> <p>3,Y2V: P'yyxx' (X'0yyxxs')</p> <p>4,Y2V: P'yyxxxx' (X'0yyxxxxs')</p> <p>y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ 9 で、年以外の桁を表します。s は 16 進数 0 ~ F で、無視されます。xx または xxxx は正しい照合順序である必要があります。</p> <p>特殊標識は、P'0...0' (PD ゼロ) および P'9...9' (PD の 9) です。</p>
Y2W	<p>(文字またはゾーン 10 進数の xyy, xxyy, xxxyy, および xxxxyy、特殊標識付きの完全な日付形式)</p> <p>日付フィールドは、次のように表記できます。</p> <p>3,Y2W: C'xyy' または Z'xyy'</p> <p>4,Y2W: C'xxyy' または Z'xxyy'</p> <p>5,Y2W: C'xxxxyy' または Z'xxxxyy'</p> <p>6,Y2W: C'xxxxxyy' または Z'xxxxxyy'</p> <p>y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ 9 で、年以外の桁を表します。x...x は、正しい照合順序である必要があります。x...xyy は、日付フィールドを照合するとき、yyx...x として処理されます。</p> <p>特殊標識は、X'00...00' (BI ゼロ)、X'40...40' (ブランク)、C'0...0' (CH ゼロ)、Z'0...0' (ZD ゼロ)、C'9...9' (CH の 9)、Z'9...9' (ZD の 9)、および X'FF...FF' (BI の 1) です。</p>
Y2X	<p>(パック 10 進数の xyy および xxxxyy、特殊標識付きの完全な日付形式)</p> <p>日付フィールドは、次のように表記できます。</p> <p>2,Y2X: P'xyy' (X'xyys')</p> <p>3,Y2X: P'xxxxyy' (X'xxxxyys')</p> <p>y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ 9 で、年以外の桁を表します。s は 16 進数 0 ~ F で、無視されます。x...x は、正しい照合順序である必要があります。x...xyy は、日付フィールドを照合するとき、yyx...x として処理されます。</p> <p>特殊標識は、P'0...0' (PD ゼロ) および P'9...9' (PD の 9) です。</p>
Y2Y	<p>(パック 10 進数の xxyy および xxxxyy、特殊標識付きの完全な日付形式)</p> <p>日付フィールドは、次のように表記できます。</p> <p>3,Y2Y: P'xxyy' (X'0xxyys')</p> <p>4,Y2Y: P'xxxxyy' (X'0xxxxyys')</p> <p>y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ 9 で、年以外の桁を表します。s は 16 進数 0 ~ F で、無視されます。xx または xxxx は正しい照合順序である必要があります。x...xyy は、日付フィールドを照合するとき、yyx...x として処理されます。</p> <p>特殊標識は、P'0...0' (PD ゼロ) および P'9...9' (PD の 9) です。</p>

## データ形式の例

形式	説明												
Y2C または Y2Z	<p>(2 桁、2 バイト文字、またはゾーン 10 進数の年データ) 2 桁の年データは、次のように表記できます。</p> <p>xyxy y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ F で、無視されます。</p> <p>このようにして、96 は 16 進数 F9F6 (文字 96) として、あるいは 16 進数 F9C6 または 0906 (ゾーン 10 進数 96) として表記されます。</p>												
Y2P	<p>(2 桁、2 バイトのパック 10 進数の年データ) 2 桁の年データは、次のように表記できます。</p> <p>xyyx y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ F で、無視されます。</p> <p>このようにして、96 は 16 進数 096F または 896C (パック 10 進数 96) として表記されます。</p>												
Y2D	<p>(2 桁、1 バイトの 10 進数の年データ) 2 桁の年データは、次のように表記できます。</p> <p>yy y は 16 進数 0 ~ 9 で、年の桁を表示します。</p> <p>このようにして、96 は 16 進数 96 (10 進数 96) として表記されます。</p>												
Y2S	<p>(2 桁、2 バイト文字またはゾーン 10 進数の特殊標識の付いた年データ)</p> <p>2 桁の年データは、次のように表記できます。</p> <p>xyxy y は 16 進数 0 ~ 9 で、年の桁を表示します。x は 16 進数 0 ~ F で、無視されます。</p> <p>このようにして、96 は 16 進数 F9F6 (文字 96) として、あるいは 16 進数 F9C6 または 0906 (ゾーン 10 進数 96) として表現されます。</p> <p>特殊標識を、次のように表現できます。</p> <p>qxzx qx は 16 進数 00、40、または FF です。zx は 16 進数 00 ~ FF (通常は、00、40、および FF) です。</p> <p>このように、特殊標識は、16 進数 0000、0005、4040、FFFF、FF85 などです。</p>												
Y2B	<p>(2 桁、1 バイトの 2 進数の年データ) 2 進数の年データを次のように表現できます。</p> <p>hh hh は、次のように 10 進数 yy の値に等価な 16 進数です。</p> <table border="1" data-bbox="381 1417 1023 1522"> <thead> <tr> <th>Binary Values</th> <th>Decimal Values</th> <th>yy</th> </tr> </thead> <tbody> <tr> <td>X'00'-X'63'</td> <td>00-99</td> <td>00-99</td> </tr> <tr> <td>X'64'-X'C7'</td> <td>100-199</td> <td>00-99</td> </tr> <tr> <td>X'C8'-X'FF</td> <td>200-255</td> <td>00-55</td> </tr> </tbody> </table> <p>このようにして、96 は 16 進数 60 (10 進数 96) または C4 (10 進数 196) として表現されます。</p>	Binary Values	Decimal Values	yy	X'00'-X'63'	00-99	00-99	X'64'-X'C7'	100-199	00-99	X'C8'-X'FF	200-255	00-55
Binary Values	Decimal Values	yy											
X'00'-X'63'	00-99	00-99											
X'64'-X'C7'	100-199	00-99											
X'C8'-X'FF	200-255	00-55											
DT1	<p>(Z'yyyymmdd' と解釈された SMF 日付)。形式 P'cyyddd' (X'0cyydddF') で表される 4 バイトの SMF 日付値は、Z'yyyymmdd' の値に変換されます。c は世紀標識 (c=0 は 19、c=1 は 20、c&gt;1 は 21 にそれぞれ変換される)、yy は 2 桁年 (00 ~ 99)、ddd は年間通日 (000 ~ 366)、yyyy は 4 桁年、mm は月 (00 ~ 12)、および dd は日 (00 ~ 31) を示します。</p>												
DT2	<p>(Z'yyyymm' と解釈された SMF 日付)。形式 P'cyyddd' (X'0cyydddF') で表される 4 バイトの SMF 日付値は、Z'yyyymm' の値に変換されます。c は世紀標識 (c=0 は 19、c=1 は 20、c&gt;1 は 21 にそれぞれ変換される)、yy は 2 桁年 (00 ~ 99)、ddd は年間通日 (000 ~ 366)、yyyy は 4 桁年、および mm は月 (00 ~ 12) を示します。</p>												



形式	説明
DT3	(Z'yyyyddd' と解釈された SMF 日付)。形式 P'cyyddd' (X'0cyydddF') で表される 4 バイトの SMF 日付値は、Z'yyyyddd' の値に変換されます。c は世紀標識 (c=0 は 19、c=1 は 20、c>1 は 21 にそれぞれ変換される)、yy は 2 桁年 (00 ~ 99)、ddd は年間通日 (000 ~ 366)、および yyyy は 4 桁年を示します。
TM1	(Z'hmmss' と解釈された SMF 時間)。100 分の 1 秒単位で表される 4 バイトの SMF 時間値は、'hmmss' の値に変換されます。hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、および ss は秒 (00 ~ 59) を示します。
TM2	(Z'hmmm' と解釈された SMF 時間)。100 分の 1 秒単位で表される 4 バイトの SMF 時間値は、'hmmm' の値に変換されます。hh は時間 (00 ~ 23)、および mm は分 (00 ~ 59) を示します。
TM3	(Z'hh' と解釈された SMF 時間)。100 分の 1 秒単位で表される 4 バイトの SMF 時間値は、'hh' の値に変換されます。hh は時間 (00 ~ 23) を示します。
TM4	(Z'hmmssxx' と解釈された SMF 時間)。100 分の 1 秒単位で表される 4 バイトの SMF 時間値は、'hmmssxx' の値に変換されます。hh は時間 (00 ~ 23)、mm は分 (00 ~ 59)、ss は秒 (00 ~ 59)、および xx は 100 分の 1 秒 (00 ~ 99) を示します。

<sup>1</sup> オーバーパンチ符号ビットは、常に、正の場合は 'C' で、負の場合は 'D' です。

CH、ZD、PD、FI、BI、および FL データ形式の詳細については、アセンブラー解説書を参照してください。

次のテーブルは、各種のデータ形式のそれぞれで使用できるステートメント、オペランド、および演算子を示しています。

表 81. 頻繁に使用されるデータ・タイプで使用できるもの

ステートメント、オペランド、または演算子	CH	BI	FI	PD	ZD	FS または CSF	DTn または TMn
DFSORT ステートメント							
INCLUDE	X	X	X	X	X	X	
INREC		X	X	X	X	X	X
MERGE	X	X	X	X	X	X	
OMIT	X	X	X	X	X	X	
OUTREC		X	X	X	X	X	X
SORT	X	X	X	X	X	X	
SUM		X	X	X	X		
OUTFIL ステートメントのオペランド							
INCLUDE	X	X	X	X	X	X	
OMIT	X	X	X	X	X	X	
OUTREC		X	X	X	X	X	X
TRAILERx		X	X	X	X	X	
ICETOOL 演算子							
DISPLAY (ON, BREAK)	X	X	X	X	X	X	X
OCCUR (ON)	X	X	X	X	X	X	X
RANGE (ON)		X	X	X	X	X	
SELECT (ON)	X	X	X	X	X	X	
STATS (ON)		X	X	X	X	X	
UNIQUE (ON)		X	X	X	X	X	

## データ形式の例

表 81. 頻繁に使用されるデータ・タイプで利用できるもの (続き)

ステートメント、オペランド、または演算子	CH	BI	FI	PD	ZD	FS または CSF	DTn または TMn
VERIFY (ON)				X	X		

表 82. その他のデータ・タイプで利用できるもの

ステートメント またはオペランド	AQ	AC	FL	LS または CSL	TS または CST	OL または CLO	OT または CTO	ASL	AST	D1	D2	PD0	Y2x
DFSORT ステートメント													
INCLUDE	X	X		X	X	X	X	X	X		X		X
INREC												X	X
MERGE	X	X	X	X	X	X	X	X	X	X		X	X
OMIT	X	X		X	X	X	X	X	X		X		X
OUTREC												X	X
SORT	X	X	X	X	X	X	X	X	X	X		X	X
SUM			X										
OUTFIL ステートメントのオペランド													
INCLUDE	X	X		X	X	X	X	X	X				X
OMIT	X	X		X	X	X	X	X	X				X
OUTREC												X	X

## 付録 D. EBCDIC および ISCII/ASCII の照合順序

### EBCDIC

表 83 は、EBCDIC 文字および無符号 10 進数データの照合順序を示しています。照合順序は、低位のもの (00000000) から高位のもの (11111111) へと並べてあります。シンボルと対応しないビット構成 (0 ~ 73、81 ~ 89、など) は示されていません。これらの一部は、プリンターやその他の装置の制御コマンドに対応するものです。

ALTSEQ、CHALT、および LOCALE を使用して、文字データの代替照合順序を選択できます。

パック 10 進数、ゾーン 10 進数、固定小数点、および正規化浮動小数点の各データは代数的に照合されます。つまり、各数量は符号をもっているものとして解釈されます。

表 83. EBCDIC の照合順序

照合順序	ビット構成	シンボル	意味
0	00000000		
.			
.			
64	01100100	SP	スペース
.			
.			
74	01001010	θ	セント記号
75	01001011	.	ピリオド、10 進小数点
76	01001100	<	不等号 (より小)
77	01001101	(	左括弧
78	01001110	+	正符号
79	01001111	I	縦棒、論理 OR
80	01010000	&	アンパーサンド
.			
.			
90	01011010	!	感嘆符
91	01011011	\$	ドル記号
92	01011100	*	アスタリスク
93	01011101	)	右括弧
94	01011110	;	セミコロン
95	01011111		論理否定
96	01100000	—	負符号、ハイフン
97	01100001	/	スラッシュ
107	01101011	,	コンマ
108	01101100	%	% 記号
109	01101101	—	下線
110	01101110	>	不等号 (より大)
111	01101111	?	疑問符
.			
.			
122	01111010	:	コロン
123	01111011	#	番号記号
124	01111100	@	単価記号

表 83. EBCDIC の照合順序 (続き)

照合順序	ビット構成	シンボル	意味
125	01111101	'	アポストロフィ、プライム記号
126	01111110	=	等号
127	01111111	"	引用符
.			
.			
129	10000001	a	
130	10000010	b	
131	10000011	c	
132	10000100	d	
133	10000101	e	
134	10000110	f	
135	10000111	g	
136	10001000	h	
137	10001001	i	
.			
.			
145	10010001	j	
146	10010010	k	
147	10010011	l	
148	10010100	m	
149	10010101	n	
150	10010110	o	
151	10010111	p	
152	10011000	q	
153	10011001	r	
.			
.			
162	10100010	s	
163	10100011	t	
164	10100100	u	
165	10100101	v	
166	10100110	w	
167	10100111	x	
168	10101000	y	
169	10101001	z	
193	11000001	A	
194	11000010	B	
195	11000011	C	
196	11000100	D	
197	11000101	E	
198	11000110	F	
199	11000111	G	
200	11001000	H	
201	11001001	I	
.			
.			
209	11010001	J	
210	11010010	K	
211	11010011	L	
212	11010100	M	
213	11010101	N	
214	11010110	O	
215	11010111	P	

表 83. EBCDIC の照合順序 (続き)

照合順序	ビット構成	シンボル	意味
216	11011000	Q	
217	11011001	R	
.			
.			
226	11100010	S	
227	11100011	T	
228	11100100	U	
229	11100101	V	
230	11100010	W	
231	11100111	X	
232	11101000	Y	
233	11101001	Z	
.			
.			
240	11110000	0	
241	11110001	1	
242	11110010	2	
243	11110011	3	
244	11110100	4	
245	11110101	5	
246	11110110	6	
247	11110111	7	
248	11111000	8	
249	11111001	9	
.			
.			
255	11111111		

## ISCI/ASCII

表 84 は、ISCI/ASCII 文字および無符号 10 進数データの照合順序を示しています。照合順序は、低位のもの (00000000) から高位のもの (01111111) へと並べてあります。シンボルと対応しないビット構成は示されていません。

パック 10 進数、ゾーン 10 進数、固定小数点、正規化浮動小数点の各データ、および符号付きの数値データ形式は、代数的に照合されます。つまり、各数量は符号をもっているものとして解釈されます。

表 84. ISCI/ASCII の照合順序

照合順序	ビット構成	シンボル	意味
0	00000000	NULL	
.			
.			
32	00100000	SP	スペース
33	00100001	!	感嘆符
34	00100010	"	引用符
35	00100011	#	番号記号
36	00100100	\$	ドル記号
37	00100101	%	パーセント
38	00100110	&	アンパーサンド

表 84. ISCI/ASCII の照合順序 (続き)

照合順序	ビット構成	シンボル	意味
39	00100111	,	アポストロフィ、プライム記号
40	00101000	(	左括弧
41	00101001	)	右括弧
42	00101010	*	アスタリスク
43	00101011	+	正符号
44	00101100	,	コンマ
45	00101101	—	ハイフン、マイナス
46	00101110	.	ピリオド、10 進小数点
47	00101111	/	スラッシュ
48	00110000	0	
49	00110001	1	
50	00110010	2	
51	00110011	3	
52	00110100	4	
53	00110101	5	
54	001101100	6	
55	00110111	7	
56	00111000	8	
57	00111001	9	
58	00111010	:	コロン
59	00111011	;	セミコロン
60	00111100	<	不等号 (より小)
61	00111101	=	等号
62	00111110	>	不等号 (より大)
63	00111111	?	疑問符
64	01000000	@	単価記号
65	01000001	A	
66	01000010	B	
67	01000011	C	
68	01000100	D	
69	01000101	E	
70	01000110	F	
71	01000111	G	
72	01001000	H	
73	01001001	I	
74	01001010	J	
75	01001011	K	
76	01001100	L	
77	01001101	M	
78	01001110	N	
79	01001111	O	
80	01010000	P	
81	01010001	Q	
82	01010010	R	
83	01010011	S	
84	01010100	T	
85	01010101	U	
86	01010110	V	
87	01010111	W	
88	01011000	X	
89	01011001	Y	

表 84. ISCI/ASCII の照合順序 (続き)

照合順序	ビット構成	シンボル	意味
90	01011010	Z	
91	01011011	[	左大括弧
92	01011100	\	逆スラッシュ
93	01011101	]	右大括弧
94	01011110	^	曲折アクセント記号、論理否定
95	01011111	_	下線
96	01100000	`	抑音符号
97	01100001	a	
98	01100010	b	
99	01100011	c	
100	01100100	d	
101	01100101	e	
102	01100110	f	
103	01100111	g	
104	01101000	h	
105	01101001	i	
106	01101010	j	
107	01101011	k	
108	01101100	l	
109	01101101	m	
110	01101110	n	
111	01101111	o	
112	01110000	p	
113	01110001	q	
114	01110010	r	
115	01110011	s	
116	01110100	t	
117	01110101	u	
118	01110110	v	
119	01110111	w	
120	01111000	x	
121	01111001	y	
122	01111010	z	
123	01111011	{	左中括弧
124	01111100		縦線
125	01111101	}	右中括弧
126	01111110	~	波形記号





---

## 付録 E. DFSORT の異常終了処理

ここでは、DFSORT が異常終了をどのように処理するかについて説明します。この付録は、異常終了の原因となるエラーを診断するために必要なダンプを取得する際に役立ちます。

DFSORT により生成される異常終了ダンプはすべて、標準ダンプ分析プログラムで処理できるシステム異常終了ダンプです。SYSUDUMP、SYSABEND、または SYSMDUMP DD ステートメントをアプリケーションに組み込んでおけば、ダンプが生成されます。実際に出力されるシステム・ダンプは、ユーザーが SYS1.PARMLIB の IEADMP00、IEAABD00、または IEADMR00 メンバーに指定したシステム・パラメーターにより異なります。

ブロック・セットおよびピアレッジ / ベール・アプリケーションのシステムまたはユーザーの異常終了をトラップするために、DFSORT は、各実行の始めに ESTAE リカバリー・ルーチンを設定します。インストール時に ICEMAC ESTAE=NO を指定するか、あるいは DEBUG 制御ステートメントで NOESTAE を指定することにより、このルーチンを削除できます。常に ESTAE を有効にして実行することをお勧めします。そうすることにより、異常終了の際に次のような利点があります。

- 一般に、異常終了の発生時により近い時点でダンプを取得することができます。
- 異常終了の原因となっている問題を診断する上で役立つ追加情報を入手します。
- SMF、ICETEXIT ルーチン、または EFS プログラムを活動化すると、DFSORT は処理の継続を試みます。すなわち、アプリケーションが処理を終了する前に、SMF レコードが作成され、終了出口が呼び出され、あるいは、EFS プログラムへのメジャー・コール 4 および 5 が行われます。もちろん、上記のいずれかの機能が原因で異常終了が生じた場合は、その機能は正常に完了しません。

リカバリー・ルーチンの終わりに、DFSORT は必ず制御をシステムに戻して引き続き終了が行えるようにします。次に、システムは次の高位レベルの ESTAE リカバリー・ルーチンを呼び出します。

---

### チェックポイント / 再始動

チェックポイント / 再始動は、アプリケーションに関する情報を記録しておくことができるオペレーティング・システムです。これにより、異常終了の後、またはアプリケーションのいずれかの部分が完了した後で、同じアプリケーションの再始動が可能になります。再始動は直ちに行うこともできるし、あるいはそのアプリケーションが実行依頼されるまで延期することもできます。

ピアレッジ / ベール手法を使用する分類中に要求されると、DFSORT はチェックポイントを取ります。

DFSORT にチェックポイントを記録させるには、SORTCKPT DD ステートメントをコーディングし、ピアレッジ / ベール手法を必ず選択する必要があります。SORTCKPT および CKPT オプションの詳細については、それぞれ 84 ページの『SORTCKPT DD ステートメント』および 181 ページの『OPTION 制御ステートメント』を参照してください。

## 異常終了処理

一般に、次の条件がある場合は、チェックポイントは取られません。

- 作業データ・セットが指定されていない。
- アプリケーションがコピーまたは組み合わせである。
- ブロック・セットが選択されている。

注:

1. チェックポイント / 再始動処理中は、ANSI 標準ラベル・テープ・ファイルをオープンすることができません。
2. DFSORT DD ステートメントには `CHKPT=EOV` を指定できません。

チェックポイント / 再始動の詳細については、*z/OS DFSMS Checkpoint/Restart*を参照してください。

---

## DFSORT の異常終了の種類

DFSORT には 2 種類の異常終了があります。すなわち、予期しない異常終了と DFSORT により出されるユーザー異常終了です。

- 予期しない異常終了

予期しない異常終了とは、DFSORT により出されないシステム異常終了またはユーザー異常終了です。この場合の異常終了コードは、システム異常終了コードまたはユーザー異常終了コードです。普通のユーザー・エラーを検出したり、DFSORT プログラム障害を報告する操作に関する詳細は、*DFSORT* メッセージ、コード、および診断の手引き リリース 14 を参照してください。

- DFSORT により出されるユーザー異常終了

DFSORT は、次のような状況のもとで、ユーザー異常終了を出します。

- ABEND または ABSTP オプションが有効になっていて、DFSORT が、実行の完了を妨げるエラーを検出した
- DFSORT が内部論理内にエラーを検出した

DFSORT により出されるユーザー異常終了の詳細については、*DFSORT* メッセージ、コード、および診断の手引き リリース 14 を参照してください。

注: SmartBatch パイプ・データ・セットに伴う特殊なユーザー異常終了処理については、18 ページの『SmartBatch パイプに関する考慮事項』を参照してください。

---

## 予期しない異常終了の場合の異常終了リカバリー処理

DFSORT は通常、ブロック・セットおよびピアレッジ / ベール・アプリケーションのシステムまたはユーザー出口ルーチンの異常終了をトラップするために、ESTAE リカバリー・ルーチンを設定してあります。異常終了が生じると、システムは制御をこのルーチンに渡します。DFSORT ESTAE リカバリー・ルーチンの機能は次のとおりです。

- 異常終了ダンプ

リカバリー・ルーチンは、まず、システムに異常終了ダンプを発行させて、エラー発生時の環境を把握します。

- 終了機能

プログラムが正常に終了するか異常終了するかにかかわらず、DFSORT は次のようなタスクが指定されていれば、それを実行しようと試みます。

- EFS プログラムの呼び出し 4 および 5
- SMF レコードの作成
- ICETEXIT ルーチンの呼び出し

異常終了が起きた時点で、上記の機能のいずれかがまだ実行されていない場合は、DFSORT のリカバリー・ルーチンはそれを実行します。

- 異常終了通知メッセージ

予期しないシステム異常終了またはユーザー出口ルーチンの異常終了の場合、DFSORT リカバリー・ルーチンは、メッセージ ICE185A を出して、異常終了がいつ起こったかを知らせます。DFSORT メッセージ、コード、および診断の手引き リリース 14のこのメッセージの説明を参照してください。

- スナップ・ダンプ

DFSORT リカバリー・ルーチンは、システム診断作業域 (SDWA) のスナップ・ダンプを提供します。スナップ・ダンプは、SYSUDUMP (または SYSABEND または SYSDUMP) DD ステートメントがアプリケーションに含まれているかどうかにかかわらず、動的に割り振られたデータ・セットに書き込まれます。

- システム診断作業域のコピー

呼び出しプログラムが SDWA 区域のアドレスを 24 ビット・パラメーター・リストまたは拡張パラメーター・リストに渡す場合、DFSORT はシステム診断作業域の最初の 104 バイトまたは 112 バイトをユーザーの SDWA 区域にコピーします。詳細については、421 ページの『第 5 章 プログラムからの DFSORT の呼び出し』を参照してください。

- 正常 SORTOUT 出力後のアプリケーションの継続

分類、組み合わせ、またはコピー・アプリケーションが SORTOUT データ・セットを正常に書き込んだ後で予期しない異常終了が発生した場合は、DFSORT はメッセージ ICE186A を出し、通常の終結処置と終了機能を完了します。DFSORT により書き込まれた SORTOUT データ・セットはクローズされます。異常終了を引き起こした機能を除き、実行は正常に終了します。メッセージ ICE186A は、その実行が異常終了したにもかかわらず、SORTOUT データ・セットは使用可能であることを通知します。ユーザーはその SORTOUT データ・セットを使用するか、またはアプリケーションを再実行するかを決定できます。

- DFSORT は、異常終了のリカバリー処理の終わりに制御をシステムに戻し、リカバリー・ルーチンを呼び出せるようにします。

上記の DFSORT 異常終了リカバリー・ルーチン機能は、NOESTAE が有効になっている場合は、異常終了の後では実行できないことがあります。DFSORT ESTAE リカバリー・ルーチンは、常に、実行の始めに設定されます。NOESTAE が有効になっている場合は、このルーチンは、DFSORT 処理の早い段階で削除されます。

---

## A タイプ・メッセージを伴うエラー異常終了の処理

DFSORT は、重大なエラーを見つけると、A タイプ・メッセージを出して終了します。ユーザーは、DFSORT が、ABEND または ABSTP オプションを介して、異常終了でアプリケーションを終了させるように指定できます。

## 異常終了処理

異常終了による終了が有効になっていて、DFSORT が重大なエラーを見つけた場合は、DFSORT はまず、異常終了ダンプによりそのエラー時の環境を把握します。次に、A タイプ・メッセージを出します。また、異常終了で終了する前に、前述の終了機能を実行します。異常終了コードはエラー・メッセージの番号になるか、あるいはインストール時に ICEMAC ABCODE オプションで決定された 1 から 99 までの番号になります。

NOESTAE および ABEND が有効になっていれば、異常終了ダンプは A タイプ・メッセージの印刷後に作成され、他の終了機能が実行されます。その結果、作成されたダンプがエラー発生時の条件を反映していない場合があります。そのダンプには、エラーになったモジュールが含まれていない可能性があります。

NOESTAE および ABSTP が有効になっていれば、正しいモジュールはダンプされますが、A タイプ・メッセージは出されません。

---

## CTR<sub>x</sub> 異常終了処理

CTR<sub>x</sub> オプションを使用すると、レコードの入力処理中または出力処理中に、その DFSORT の異常終了を要求して、問題を診断できます。111 ページの『第 3 章 DFSORT プログラム制御ステートメントの使用』の DEBUG 制御ステートメントの項を参照してください。DFSORT は、CTR<sub>x</sub> カウントが満たされると 0C1 異常終了を引き起こします。DFSORT ESTAE リカバリー・ルーチンは、前述の予期しない異常終了の処理と同じ方法で、この異常終了を処理します。

DFSORT ESTAE リカバリー・ルーチンは制御をシステムに戻し、システムは、呼び出しプログラムにより設定された ESTAE ルーチンへ制御を渡します。

前述のように、呼び出しプログラムの SDWA 域のアドレスが DFSORT に渡された場合、DFSORT の ESTAE リカバリー・ルーチンは、その SDWA 域の中にシステム診断作業域の最初の 104 バイトまたは 112 バイトを保管します。

PL/I は通常、ESPIE を有効にしてプログラム・チェック (0C<sub>x</sub> 異常終了コード) を代行受信するため、NOSPIE を指定していない限り、DFSORT ESTAE リカバリー・ルーチンはこれらのエラーの後で呼び出されません。DFSORT の異常終了リカバリー処理は、その他すべてのタイプの異常終了に対して実行されます。

COBOL プログラムから呼び出した場合、または COBOL 出口を使用した場合は、複数の異常終了ダンプが作成されることがあります。

## 付録 F. 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できます。ただし、IBM 以外の製品、プログラムまたはサービスの操作性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、使用権等の許諾については、下記の宛先に書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31  
IBM World Trade Asia Corporation  
Intellectual Property Law & 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。国または地域によりは、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  
Information Enabling Requests  
Dept. DZWA  
5600 Cottle Road  
San Jose, CA 95193 U.S.A.

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・インターフェースの情報

本書は、お客さまがプログラムを書いて DFSORT のサービス入手するためのプログラミング・インターフェースとなるように作成されています。

---

## 商標

以下は、IBM Corporation の商標です。

AD/Cycle	Hiperspace	RAMAC
COBOL/370	IBM	System/390
DFSMS/MVS	Language Environment	VM/ESA
DFSMSrmm	MVS/ESA	z/OS
DFSORT	OS/390	z/VM
ES/9000	RACF	3090
ESCON		

2 つのアスタリスク (\*\*) が付けられている他社名、製品名、およびサービス名は、他社の商標またはサービス・マークです。

---

## 変更の要約

---

### リリース 13 の新しいプログラミング・サポート (1996 年 4 月 以降の PTF)

#### 新規 2000 年対応機能

新しい Y2S 形式は、年フィールドの 2 進ゼロ、ブランク、2 進 1 を特有の標識として扱い、2 桁文字またはゾーン 10 進数の年データを、「世紀」ウィンドウに応じて配列および変換します。

新しい Y2B 形式は、「世紀」ウィンドウに応じて、2 桁の 2 進年データを配列および変換します。

DFSPARM の FREE=CLOSE サポートは、同一 COBOL プログラム内の複数の COBOL SORT 動詞により作成された、SORT ステートメントの指定変更を実現します。

#### OS/390 の登録

OS/390 R2 以降では、DFSORT 製品が、OS/390 の機能としても別個のプログラム製品としても、使用ライセンス交付を受けているかどうか確認されます。

---

### リリース 13 の新しいプログラミング・サポート (1996 年 4 月 PTF)

#### 2000 年対応機能

新しい Y2C、Y2Z、Y2P および Y2D 形式を、新しい Y2PAST インストールおよび実行時オプションと共に使用して、以下のように 2 桁年データを処理できます。

- 使用するアプリケーションに、適切な「世紀」ウィンドウを設定 (たとえば、1915-2014 または 1950-2049)。
- Blockset SORT または MERGE を使用して、「世紀」ウィンドウに応じ、2 桁文字、ゾーン 10 進数、パック 10 進数、または 10 進数の年データを配列 (たとえば、1996 を示す 96 を 2000 を示す 00 の前に昇順で配列、または 00 を 96 の前に降順で配列)。
- OUTFIL OUTREC を使用して、「世紀」ウィンドウに応じ、2 桁文字、ゾーン 10 進数、パック 10 進数、または 10 進数の年データを 4 桁文字年データに変換 (たとえば、96 を 1996 に、00 を 2000 に変換)。

新しい PD0 形式で、SORT、MERGE および OUTFIL を使用して、パック 10 進数フィールドの一部を配列および変換できます (たとえば、日付フィールドの月日)。

#### FLR および VLR ブロック・セット分類におけるパフォーマンスの向上

FLR および VLR ブロック・セット分類のパフォーマンスは、以下のように向上しました。

- データ空間分類は、可変長レコード分類のアプリケーションの場合のみ使用されるようになりました。
- DFSORT データ処理方式が改善されました。
- 動的ストレージ調整は、DFSORT が、それによりパフォーマンスの向上が可能だと判断した場合、自動的にブロック・セット分類アプリケーションの TMAXLIM 値 より大きい記憶域を使用できる、新しい機能です。新しいインストール・オプション DSA=n が追加され、動的ストレージ調整の限度を指定できるようになりました。
- DFSORT が指定および使用できる、主記憶の容量の上限が、32 メガバイトから 2000 メガバイトに増えました。より容量の大きい主記憶域を指定することで、以下のメリットがあります。
  - DFSORT は、非常に大きなデータ・セットをより効率よく分類できるようになります。
  - 中間ワークスペースが不要になり、また分類アプリケーションの EXCP カウント を大幅に削減して、より多くの分類アプリケーションを主記憶域全体で使用できます。
  - 1 回の分類アプリケーションで DFSORT が処理できるデータの最大量を増やすことができます。
- 新しいインストール・オプション IOMAXBF=n が追加され、1 回の入出力操作で変換可能なデータの容量を順に制限する、SORTIN および SORTOUT データ・セット・バッファで使用する記憶域容量の上限を、指定できるようになりました。
- DFSORT のブロック・セット手法で指定して使用できる、JCL の数と動的に割り振られる作業データ・セット数の上限が、32 から 100 に増えました。使用する作業データ・セットを増やすと、1 回の分類アプリケーションで DFSORT が処理できるデータの最大量を増やすことができます。
- DFSORT SVC への変更により、キャッシュ選択機能強化が実現し、SORTIN および SORTOUT 装置については特に、ストレージ管理キャッシュ・パフォーマンスが向上しました。
- EQUALS=NO がインストール時に指定される場合、または NOEQUALS が実行時に指定される場合、DFSORT は、VLR ブロック・セット・アプリケーションの NOEQUALS を使用できるようになりました。NOEQUALS の使用により、パフォーマンスの向上が可能になりました。これは、同様に照合するレコードの配列を、入力から出力まで保存する必要がないアプリケーションに推奨されます。マイグレーションに関する考慮事項を最小限にとどめるため、IBM 提供の ICEMAC EQUALS オプションのデフォルトは、新しい値 VLBLKSET となっています。これは、VLR ブロック・セット・アプリケーションの EQUALS=YES、およびその他のアプリケーションの EQUALS=NO と同等の値です。

## SUM の浮動小数点

短い (4 バイト) または長い (8 バイト)、もしくは拡張された (16 バイト) 浮動小数点データについて、FL 形式を SUM 制御ステートメントで使用できるようになりました。



## セキュリティの向上

DFSORT SVC への変更により、DFSORT のセキュリティは、B1 セキュリティー標準レベルまで向上しました。

## EXCPVR 処理の除去

DFSORT のシステム保全性の保護を強化するため、EXCPVR 処理は今後使用されません。EXCPVR パラメーター値は、これまでどおり受け入れられますが、DFSORT の処理に対してなんら影響は与えません。通常、EXCPVR 処理によるパフォーマンスの向上は、より新しいテクノロジーと共に減少し、上記のパフォーマンスの向上により、オフセット以上のものとなります。本書の EXCPVR への参照は、すべて無視してください。それらの参照は、本書の更新時に削除されます。

---

## リリース 13 で新たに加わった装置サポート (PTF)

IBM 3590 磁気テープ・サブシステムは、入力、出力、および作業データ・セットでサポートされます。

---

## 新しいプログラミング・サポート (リリース 13)

### SAS\*\* システム用の DFSORT のパフォーマンス・ブースター

DFSORT リリース 13 は、SAS アプリケーションについて、CPU 時間の著しい改善を実現します。この新しい機能を利用するには、この拡張機能のサポートの詳細について、SAS Institute Inc. にお問い合わせください。

## 動的ハイパー分類

動的ハイパー分類は、新しい自動機能です。この機能は、意図しないシステム・ページング活動、および拡張記憶域とページング・データ・セットのスペース不足 (これらは、大量のハイパー分類活動、特に複数の並行ハイパー分類アプリケーションの場合に発生することがある) を取り除くものです。

動的ハイパー分類を使用すると、DFSORT とシステムのより最適なパフォーマンスを可能にし、また、インストール・オプションにより、ユーザー独自の基準に合うように HIPRMAX=OPTIMAL をカスタマイズできます。この機能の登場に伴い、HIPRMAX=OPTIMAL をユーザーのデフォルトとして使用することをお勧めします。

## パフォーマンス

ブロック・セット手法を使用する DFSORT アプリケーションのパフォーマンス強化には、次のものが含まれます。

- データ空間分類 (固定長レコードの分類アプリケーション用に R12 で導入されたもの)。可変長レコードの分類アプリケーションに使用できるようになりました (ただし MVS/ESA のみ)。
- 固定長レコードの分類アプリケーションのためのデータ処理方式の改善。
- 1 つ以上の入力データ・セットに対する単一のパスを使用した、複数の出力データ・セットを作成する OUTFIL 処理。

## OUTFIL の処理

OUTFIL は、新しい DFSORT 制御ステートメントです。これを使用すると、1 つ以上の入力データ・セットに対する単一のパスを使用して、分類、コピー、または組み合わせアプリケーションの 1 つ以上の出力データ・セットを作成できます。複数の OUTFIL ステートメントを使用して、その各ステートメントに 1 つ以上の出力データ・セットに対して OUTFIL 処理を実行するように指定できます。他のすべての処理が終了した後 (つまり、出口、オプション、およびその他の制御ステートメントを処理した後)、OUTFIL 処理が開始されます。OUTFIL ステートメントは、次のような様々な出力データ・セットのタスクをサポートします。

- 1 つ以上の入力データ・セットに対する単一のパスを使用することにより、未編集のレコードまたは編集済みのレコードを含む複数の出力データ・セットを作成する。
- 1 つ以上の入力データ・セットに対する単一のパスを使用して、異なる範囲またはサブセットのレコードを含む複数の出力データ・セットを作成する。さらに、どのサブセット用にも選択されないレコードを別の出力データ・セットに保管できる。
- 可変長レコード・データ・セットから固定長レコード・データ・セットへ変換する。
- 高度な編集機能。たとえば、長さ、先行ゼロまたは抑制ゼロ、記号 (たとえば、千単位区切り文字や 10 進小数点)、先行 / 後書きの正負符号などで数字フィールドを表記する方法を、16 進表示および制御する。一般に使用されている数値編集パターンとして 26 個の事前定義編集マスクが用意されており、世界中で使用されている数値表記の大部分を網羅している。さらに、ユーザー定義の編集マスクにより、実際上無制限の数の数値編集パターンが利用できる。
- 入力としての文字、16 進数、またはビット・ストリングに基づいて、参照テーブルからの出力用文字または 16 進数ストリングを選択する (つまり、参照と変更)。
- 詳細な 3 レベル (報告書、ページ、およびセクション) の報告書。報告書には、ユーザーが指定できる各種の報告書要素 (たとえば、現在の日付、現在の時間、ページ番号、文字ストリング、およびブランク行) や、入力レコードから入手できる各種の報告書要素 (たとえば、文字フィールド、編集済みの数値入力フィールド、レコード数、および数値入力フィールド用に編集された合計、最大、最小、および平均) が含まれている。

## 各国語サポート

### 文化的分類および組み合わせ

DFSORT では、インストール時または実行時に活動ロケールを選択することができ、活動ロケールに定義された照合規則に従って分類または組み合わせされたレコードを出力用に作成します。これにより、言語の文化的および地域的特性を保持する定義済みの照合規則に基づいて、単一または複数バイトの文字データの分類および組み合わせを行うことができます。

### 文化的組み込みおよび除去

DFSORT では、インストール時または実行時に活動ロケールを選択することができ、活動ロケールで定義された照合規則に従って出力用レコードの組み込みや除去

を行います。これにより、言語の文化のおよび地域的特性を保持する定義済みの照合規則に基づいて、単一または複数バイトの文字データの組み込みまたは除去を行うことができます。

## OUTFIL 報告書

OUTFIL を使用すると、報告書の日付、時間、および数値を世界中で使用されている表記の多くに形式設定できます。

## ICETOOL 報告書

ICETOOL の DISPLAY 演算子を使用すると、報告書の日付、時間、および数値を世界中で使用されている表記の多くに形式設定できます。

## ICETOOL の機能強化

ICETOOL は、既存の演算子の機能を拡張した結果、さらに多様性をもつようになりました。ICETOOL の改善点は次のとおりです。

- DISPLAY および OCCUR 演算子を使用して、さらに多くのデータを表示できます。DISPLAY では、最高 20 フィールド (今までは 10 フィールド) まで、また最高 2048 文字 (今までは 121 文字) までの行の長さを使用できます。OCCUR では、最高 2048 文字 (今までは 121 文字) までの行の長さを使用できます。
- DISPLAY 演算子を用いた、さらに広範囲にわたる数値フィールドの形式設定機能。形式設定項目を使用して、区切り文字、10 進小数点、小数部の位置、符号部、先行ストリング、浮動ストリング、および後書きストリングに関して、報告書内の個々の数値フィールドの表示を変更できます。通常使用される数値編集パターンには、事前定義の 33 種類の編集マスクが使用できるため、世界中で使用されている数値表記の多くが網羅されます。先行ストリングと後書きストリングを文字フィールドで使用することもできます。
- DISPLAY 演算子と OCCUR 演算子を用いて、4 桁または 2 桁で年を表示します。
- 文字または数値段落フィールドの値に基づいて、DISPLAY 演算子を用いて報告書を節に分けます。統計 (合計、最大、最小、または平均、あるいはそのすべて) を、報告書全体だけでなく、各節ごとにも表示できます。
- COPY および SORT 演算子を用いて TO DD 名のリストに OUTFIL 処理を自動的に使用します。これにより、FROM (入力) データ・セットに対する単一のパスから複数の TO (出力) データ・セットが作成されます。
- COPY および SORT 演算子を用いる TO オペランドに加え、またはその代わりに、USING データ・セットで OUTFIL ステートメントを指定できます。
- 活動ロケールに関するインストール時のデフォルトを変更するために、COPY、COUNT、および SORT 演算子として活動ロケールを指定できます。したがって、同一の ICETOOL ジョブ・ステップで、これらの演算子に関して複数の活動ロケールを使用できます。
- SELECT 演算子を用いて、各固有のフィールドに関して最後のレコードを保持できます。

## INCLUDE/OMIT サブストリングの検索

INCLUDE および OMIT 機能の拡張により、次のような場合にレコードの組み込みや除去を行うことができる、強力なサブストリング探索機能が提供されます。

- 指定の文字または 16 進定数は、指定の入力フィールド内のどこにでも入ります (つまり定数はフィールド内のサブストリングです)。または、
- 指定の入力値は、指定の文字または 16 進定数の中のどこにでも入ります (つまり、フィールドは定数内におけるサブストリングです)。

## SMF タイプ 16 レコードの機能強化

新しいフィールド (たとえば、各 DFSORT 実行の SORTIN、SORTINnn、SORTOUT、および OUTFIL データ・セットに関する情報)、制御ステートメント、レコード数、指定値 (E15、E35、HIPRMAX、DSPSIZE、FILSZ、LOCALE、および AVGRLEN についての) が DFSORT の SMF のタイプ 16 レコードに追加されました。

SMF=FULL、SMF=SHORT、および SMF=NO を DFSPARM の OPTION ステートメントまたは拡張パラメーター・リストに指定することにより、個々のアプリケーションに SMF のタイプ 16 レコードを作成したり、抑制したりすることができるようになりました。

**注:** フィールド ICESPGN、ICEUSER、および ICEGROUP のオフセットが、リリース 13 の SMF レコードで変更されました。これらのフィールドを参照するプログラムがある場合は、リリース 13 の SMF レコードを使用してそれらのプログラムを実行する前に、リリース 13 の ICESMF マクロを使用して、そのプログラムを再コンパイルしてください。

## その他の機能強化

以下に示すいくつかの ICEMAC インストール・オプションが、追加または変更されました。

- IBM 提供の EXCPVR のデフォルトが、ALL から NONE へ変更されました。
- IBM 提供の DYNAUTO のデフォルトが、NO から YES へ変更されました。
- SDBMSG を使用すれば、DFSORT が、システム定義の最適ブロック・サイズをメッセージ・データ・セット、および ICETOOL メッセージ / リスト・データ・セットに使用するかどうかを指定できます。
- LOCALE を使用すると、活動ロケールを選択できます。
- ODMAXBF を使用すると、DFSORT が各 OUTFIL データ・セット用に使用できる最大バッファ・スペースを指定できます。
- EXPMAX を使用すると、すべてのハイパー分類アプリケーション用に使用する記憶域の最大合計値を指定できます。
- EXPOLD を使用すると、すべてのハイパー分類アプリケーションが任意の一時点で使用する古い拡張記憶域の最大合計値を指定できます。
- EXPRES を使用すると、非ハイパー分類アプリケーションが使用するために DFSORT が予約する使用可能な拡張記憶域の最小値を指定できます。

いくつかの実行時オプションが追加または変更されました。

- LOCALE を使用すると、活動ロケールを選択できます。
- SMF を使用すると、DFSORT が SMF のタイプ 16 レコードを作成するかどうかを指定できます。

- ODMAXBF を使用すると、DFSORT が各 OUTFIL データ・セット用に使用できる最大バッファ・スペースを指定できます。
- NZDPRINT を使用すると、正の ZD 合計の結果が印刷可能な数字に変換されないように指示できます (ZDPRINT を指定変更します)。
- MODS ステートメントで HILEVEL=YES を使用すると、E15 および E35 ルーチンを COBOL 出口として扱うように指示できます。
- DEBUG オプション BUFFERS=ANY と BUFFERS=BELOW は認識されますが、使用されなくなりました。

DFSORT はアプリケーションに不要な DD ステートメントを無視するようになりました (たとえば、SORTIN DD ステートメントは、組み合わせアプリケーションでは無視されます)。

オペレーティング・システムがサポートされないために正常完了しない場合は、DFSORT、ICEGENER、および ICETOOL が戻りコード 24 をオペレーティング・システムまたは呼び出しプログラムに戻すようになりました。

インストール初期設定出口 ICEIEXIT を使用すると、DFSORT が各 OUTFIL データ・セットに使用できる最大バッファ・スペースを指定できます。

インストール・システム終了出口 ICETEXIT には、OUTFIL 処理が使用されたことを示すフラグなどの追加フィールドが含まれています。

INREC および OUTREC の場合は、次のとおりです。

- 列の上限とフィールドの末尾が、それぞれ 32000 から 32752 へ上がりました。
- 可変長レコードの RDW フィールドの前の 1: は受け入れられて、無視されません。

INCLUDE および OMIT の場合、COND=ALL、COND=(ALL)、COND=NONE、および COND=(NONE) を使用すると、すべてのレコードを組み込んだり、除去できます。

E15 または E32 ユーザー出口がすべての入力レコードを渡す際に L1 値が指定されていない場合は、RECORD ステートメントの L2 値が使用されます。

入力が VSAM データ・セットで、出力が RECFM の指定のない非 VSAM データ・セットの場合、DFSORT は、出力 RECFM を非ブロックとしてではなくブロックとしてセットします。こうすることにより、出力用のシステム定義の最適ブロック・サイズを使用することが可能になります。

---

## リリース 12 で新たに加わったプログラミング・サポート (PTF)

テープ出力データ・セットが DISP=MOD または DISP=OLD で指定されている場合に、RECFM、LRECL、または BLKSIZE を DD ステートメントに指定せずに、ICEGENER、コピー、およびブロック・セットの分類と組み合わせが使用できるようになりました。

入力および出力データ・セット用に、順次式ストライピングがサポートされます。

入力および出力データ・セット用に、圧縮がサポートされます。

BatchPipes/MVS の入力および出力パイプがサポートされます。

---

## リリース 12 で新たに加わった装置サポート (PTF)

4 桁の装置番号がサポートされます。

IBM 3390-9 DASD が入力、出力、および作業データ・セット用にサポートされます。ただし、パフォーマンス上の理由から、作業データ・セット用にはお勧めしません。

IBM RAMAC 配列 DASD および RAMAC 配列サブシステムが、入力、出力および作業データ・セット用にサポートされます。

IBM 3990-6 制御装置がサポートされます。

IBM キャッシュ 9343 制御装置モデルがサポートされます。

# 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]

アセンブラーで作成されたサンプル・ルーチン 393, 396  
アセンブラー・ユーザー出口ルーチン  
出力フェーズのユーザー出口 386, 393  
入力フェーズのユーザー出口 375, 386  
新しいアプリケーションの設計 623  
アドレッシング  
ユーザー出口 370  
EFS プログラム 583  
EFS プログラムのユーザー出口ルーチン 611  
移行 25  
異常終了  
重大な 749  
種類 748  
処理 747, 750  
チェックポイント / 再始動 747  
予期しない異常終了の場合の処理 748, 749  
リカバリー 748, 750  
ESTAE 747  
CTR<sub>x</sub> 処理 750  
異常終了後の再始動 747  
一時ワークスペースの効率的な割り振り 636, 637  
一般的なコーディングの規則 116, 120  
一般的な考慮事項 14, 15  
インストール・オプション 20, 25  
「ICEMAC」を参照 183  
インストール・オプション、パフォーマンスを向上させるために使用 627  
インストール・オプションによるパフォーマンスの向上 627  
インストール・デフォルト 18  
オプションの要約 20  
DEFAULTS 演算子 (ICETOOL) による表示 463  
ICETOOL でリストする 20  
インストール・デフォルトをリストするサンプル・ジョブ 20  
インターネット 4  
埋め込み  
切り捨て 138  
レコード 15, 138  
GNPAD 641  
INCLUDE/OMIT 138

エラー  
エラー・リカバリー・ルーチン 368  
重大な 749  
デバッグ・ジョブ 123  
EFS の診断 616  
エラー・メッセージ 26  
エラー・リカバリー・ルーチン  
ユーザー出口 368  
オーバーフロー 165, 358  
オカレンス  
OCCUR 演算子 (ICETOOL) 511  
SELECT 演算子 (ICETOOL) 527  
オプション、インストール 18  
オペランド・フィールド 117  
オペレーティング・システム、互換性のある 5

## [カ行]

解析に関する規則 599  
概要、DFSORT 1  
拡張機能サポート  
EFS を参照 582  
拡張パラメーター・リスト  
形式 431, 434  
例 438  
拡張パラメーター・リストの形式 431, 434  
カタログ式プロシージャ  
指定 32  
定義済み 32  
SORT 32  
SORT カタログ式プロシージャ 32  
SORTD 34  
SORTD カタログ式プロシージャ 34  
可変長レコード  
最長レコード長 17  
レコード記述子ワード 17  
関係条件  
形式 133, 138, 149  
説明 132, 133  
定義済み 132  
定数  
日付ストリング形式 150  
文字ストリングの形式 137  
10 進数形式 136  
16 進数ストリングの形式 138  
比較演算子 133, 150  
キー、定義 6  
キー順データ・セット (KSDS) 17

## 記憶域

- 一時 636
- 限界、OUTFIL 308
- 効率的な 625, 694
- 主
  - 解放 634
  - 調整 632
  - 要件に影響を与える要因 632
- 中間 636
- トラック対シリンダー 625, 693
- ユーザー出口ルーチン 369, 399
- ユーザー出口ルーチンを指定 173, 176
- 容量の超過 693, 694

## 記憶域の使用法

- E35 ユーザー出口のレコード 391

## 記憶域の割り振り

- 一時ワークスペース 636
- 主記憶域 631, 632
- 中間記憶装置 636

## 記憶管理責任者の例 646

- 規則、システム・データを管理する
  - システム・データ管理規則 14

## 規則、表記の xv

## キャッシュ高速書き込み

- パフォーマンス向上のための使用 627
- OPTION 制御ステートメントでの使用の指定 125

## 切り捨て

- レコード 15
- GNTRUNC 641
- INCLUDE/OMIT 138

## 区切りフィールド 156, 329

## 組み合わせ

- 概要 14
- 組み合わせるレコードの数の指定 45
- 組み合わせるレコードの数を指定する 46
- 組み合わせをするレコードの概数を指定する 46
- データ・セットの要件 14
- 定義済み 1
- ユーザー定義のデータ・タイプ 582, 588
- レコード 169

## 組み合わせ処理の例 673, 676

## 組み合わせに関する制約事項 439

## 組み合わせへの入力を処理

- E32 ユーザー出口 386

## 桁合わせフィールド 155, 329

## 経過時間

- 装置による改善 626

## 形式

- 後書きオーバーパンチ符号形式
  - 参照：CTO/OT (後書きオーバーパンチ符号) 形式

## 形式 (続き)

### 後書き符号形式

- 参照：CST/TS (後書き符号) 形式

### 固定小数点形式

- 参照：FI (固定小数点) 形式

### ゾーン 10 進数

- 参照：ZD (ゾーン 10 進数) 形式

### 代替文字形式

- 参照：AQ (代替文字) 形式

### パック 10 進数形式

- 参照：PD (パック 10 進数) 形式

### 浮動小数点形式

- 参照：FL (浮動小数点) 形式

### 浮動符号形式

- 参照：CSF/FS (浮動符号) 形式

### 前置きオーバーパンチ符号形式

- 参照：CLO/OL (前置きオーバーパンチ符号) 形式

### 前置き符号形式

- 参照：CSL/LS (前置き符号) 形式

### 文字形式

- 参照：CH (文字) 形式

### ユーザー定義形式 (D1)

- 参照：D1 形式

### ユーザー定義形式 (D2)

- 参照：D2 形式

### 2 進数形式

- 参照：BI (2 進数) 形式

### ISCI/ASCII 後書き符号形式

- 参照：AST (ISCI/ASCII 後書き符号) 形式

### ISCI/ASCII 前置き符号形式

- 参照：ASL (ISCI/ASCII 前置き符号) 形式

### ISCI/ASCII 文字形式

- 参照：AC (ISCI/ASCII 文字) 形式

## 形式設定

- OUTFIL 260

## 継続行 118, 119

## 継続表示桁 118

## コーディング上の制約 119, 120

## 合計

- レコード 355, 368

- E35 ユーザー出口のレコード 391

## 合計 フィールドの形式と長さの表 357

## 合計フィールド

- 形式 356

- 形式および長さの表 357

## 構文図

- 表記の規則 xv

- option 制御ステートメント 181

## 効率

- 主記憶域の使用 622



考慮事項  
キー順データ・セット (KSDS) 17  
データ・セット 13  
レコード記述子 (RDW) 17  
QSAM データ・セット 16  
VSAM データ・セット 17  
効率的な分類 / 組み合わせ手法の指定 624  
効率の改善 622, 643  
固定「世紀」ウィンドウ 231  
コピー  
概要 13  
データ・セットの要件 13  
定義済み 1  
コピー処理の例 676, 680  
コピーに関する制約事項 439, 440  
固有レコード  
OCCUR 演算子 (ICETOOL) 511  
SELECT 演算子 (ICETOOL) 527  
UNIQUE 演算子 (ICETOOL) 541

## [サ行]

最小ブロック長 15  
最小レコード長 15  
作業データ・セットへの出力の処理  
E19 ユーザー出口 383  
サブストリングの比較テスト 142  
関係条件の形式 141  
サブストリング比較演算子 142  
参照と変更 241, 272, 318  
時間定数 159, 252, 330  
磁気テープ  
効率 631, 637, 694  
作業記憶装置 637  
不十分な中間記憶装置 694  
容量に関する考慮事項 694, 695  
ワークスペース容量 694  
磁気テープ装置の共有 70, 92  
磁気テープ・ワークスペース容量の超過 694, 695  
システム DD ステートメント 70, 73, 92, 94  
システム決定ブロック・サイズ (SDB) 82, 103  
システム・データを管理する、規則  
システム・データ管理規則 14  
システム・マクロ命令  
書き方 434, 439  
使用 422, 434  
定義済み 421  
実行時フェーズ 584  
実行フェーズ 584  
指定変更  
インストール・デフォルト 183  
デフォルト 697

指定変更テーブル 699  
重大なエラー 749  
主記憶域  
解放 634  
効率的な使用 631, 635  
効率的な割り振り 632  
最小 631  
調整 632  
要件に影響を与える要因 632  
割り振り  
増加の結果 634  
主記憶域の解放 634  
主記憶域の調整 632  
出力データ・セット  
有効なタイプ 13  
要件 13  
出力データ・セットの処理  
E39 ユーザー出口 392  
処理および呼び出しプログラム 750  
処置コード 595  
処理順序、レコード 8  
初期設定フェーズ 586  
照合順序 120  
代替順序 7  
定義済み 7  
変更 7  
ユーザー出口による変更 368  
EBCDIC 7  
ISCI/ASCII 7  
照合順序の変更 120  
小制御フィールド  
参照：制御フィールド  
常駐モード  
ユーザー出口 370  
EFS プログラム 583  
EFS プログラムのユーザー出口ルーチン 611  
情報フラグ 604  
ジョブ制御言語 (JCL)  
「JCL」も参照 29  
ジョブ・ストリームの例 645  
シリンダー 625, 693  
診断  
EFS プログラム 616  
診断メッセージ 26  
シンボル  
概要 555  
キーワード・ステートメント 567  
シンボル・ステートメント 559  
注意事項 578  
注釈ステートメントとブランク・ステートメント  
559  
フィールドと定数 555

## シンボル (続き)

- 例 556
  - DFSORT ステートメント 570
  - ICETOOL 演算子
    - DISPLAY 576
    - ICETOOL の例 577
    - OCCUR 576
    - RANGE 575, 576
    - SELECT 576
    - STATS、UNIQUE および VERIFY 576
  - ICETOOL ステートメント 575
  - INCLUDE と OMIT 572
  - INREC と OUTREC 572
  - OUTFIL 573
  - SORT と MERGE 571
  - SUM 571
  - SYMNAMES DD ステートメント 558
  - SYMNAMES ステートメント 559
  - SYMNOUT DD ステートメント 559
- ## 数値の編集および形式設定
- DISPLAY 演算子 473
- ## スライド「世紀」ウィンドウ 231
- 「世紀」ウィンドウの設定 231, 319, 349, 709
- ## 制御ステートメント
- イメージの準備 423
  - オペランド・フィールド 117
  - 概要 114
  - 機能 114, 116
  - 形式 117
  - 継続表示桁 118
  - 検査、変更、または無視 590
  - コーディング 116
  - コーディング上の制約 119
  - 他の IBM プログラムを使用 120
  - 注釈ステートメント 119
  - 注釈フィールド 117
  - 命令フィールド 117
  - 要求リスト 596
  - 要約 114, 116
  - ラベル・フィールド 117
  - EFS インターフェース要求リスト 596
  - EFS コーディング規則 597, 599
  - EFS スtring 596
  - EFS プログラムにより戻されるString 598
  - EFS プログラムへ送られるString 596
  - EFS プログラムを使用して印刷 593
  - EXEC ステートメントの使用 34
- ## 制御ステートメントの継続 118
- ## 制御ステートメントのコーディング 116
- ## 制御ステートメントの指定変更 422
- ## 制御フィールド
- 概要 6, 7

## 制御フィールド (続き)

- 形式 349
  - 効率的な設計 623
  - 削除
    - INREC 制御ステートメントを使用 153
    - OUTREC 制御ステートメントの使用 327
  - 定義済み 6
  - 等号 6
  - 長さ 349
  - ユーザー出口による変更 368
  - リオーダー
    - INREC 制御ステートメントを使用 153
    - OUTREC 制御ステートメントの使用 327
  - E61 ユーザー出口による変更 384
  - MERGE 制御ステートメントの記述 170
  - SORT 制御ステートメントで記述 346
- ## 制御フィールドの形式と長さの表 349
- ## 制御フィールドの削除
- INREC を使用して 153
  - OUTREC 制御ステートメントの使用 329
- ## 制御フィールドの変更
- ユーザー出口による 368
  - E61 ユーザー出口 384
- ## 制御フィールドのリオーダー
- 「レコードの再フォーマット設定」を参照 153, 327
- ## 制御ワード 346
- ## 制約事項
- データ・セット 14
  - 長さ
    - 最小ブロック 15
    - 最小レコード 15
    - 最大レコード 14
  - レコード
    - 記憶域の制約 15
    - 最大長 14
- ## 装置、経過時間を改善する 626

## [夕行]

### 大制御フィールド

- 参照：制御フィールド
- ダイナミック・リンク編集
  - 「リンク・エディット」を参照 374
- 他の IBM プログラムの制御ステートメントの使用 120
- チェックポイント / 再始動 (CHKPT)
  - 使用 747
  - 制約事項 748
- 中間記憶域不足時の処置の決定 369
- 中間記憶装置 695
- 中間記憶装置の計算違いの処理
  - E16 ユーザー出口 378

- 注釈ステートメント 119
- 注釈ステートメントの挿入 119
- 注釈フィールド 117
- 抽出バッファ・オフセット・リスト 603
- 重複レコード
  - OCCUR 演算子 (ICETOOL) 511
  - SELECT 演算子 (ICETOOL) 527
  - SUM 制御ステートメント 355
- 直接アクセス記憶装置
  - 「DASD」を参照 625
- 直接アクセス作業記憶装置 636
- 直接呼び出し
  - 定義 5
  - DFSORT 処理 623
  - JCL の使用 700
- データ管理規則
  - システム・データ管理規則 14
- データ空間
  - 定義 189
  - EXEC PARM で指定する 41
  - OPTION 制御ステートメントによる指定 189
- データ空間分類
  - 考慮事項 638
  - 定義 638
  - 利点 638
- データ形式 477
  - 記述 733, 741
  - DFSORT ステートメント 739, 740
  - ICETOOL 演算子 739
  - OUTFIL ステートメントのオペランド 739, 740
- データ・セット 13
  - キー順、考慮事項 17
  - クローズ 370
  - システム・データ管理規則 14
  - 出力 12
    - 磁気テープ装置の共有 70, 92
  - 注意事項および制約事項 14, 18
  - 定義 13
  - 入力 12
    - 磁気テープ装置の共有 70, 92
  - メッセージ・データ・セット 26
  - ユーザー出口ルーチンによるオープン 368, 375, 386
  - ユーザー出口ルーチンによるクローズ 379, 392
  - ユーザー出口ルーチンによる処理 392
  - 有効なタイプ 13
  - 要件 13
  - page=end 考慮事項 18
  - QSAM に関する考慮事項 16
  - SmartBatch パイプ 18
  - VSAM に関する考慮事項 17
- データ・セットのオープン
  - ユーザー出口ルーチン 368
  - データ・セットのオープン (続き)
    - E11 ユーザー出口 375
    - E31 ユーザー出口 386
    - EFS 586
  - データ・セットのオープンおよび初期設定 368, 590
  - データ・セットのクローズ
    - ハウスキーピング 593
    - ユーザー出口による 370
    - E17 ユーザー出口 379
    - E37 ユーザー出口 392
    - EFS プログラムを使用 589, 593
  - データ・セットの結合
    - 「レコードの組み合わせ」を参照 169
  - データ・セットの初期設定 368, 586
  - データ・タイプ 14
  - 定義
    - インストール・オプション 20, 25
    - カタログ式プロシージャ 32
    - キー 6
    - 組み合わせ 1
    - コピー 1
    - 照合順序 7
    - 制御フィールド 6
    - 直接呼び出し 5
    - プログラム呼び出し 5
    - 分類 1
    - DD ステートメント 30
    - EXEC ステートメント 32
    - JOB ステートメント 32
- 定数
  - 日付ストリング 150
  - ビット・ストリング 147, 154
  - 文字ストリング 137
  - 10 進数 136
  - 16 進数ストリング 138
- 出口
  - 「ユーザー出口」も参照 362
  - MODS 制御ステートメント・オプション 173
- 出口ルーチン
  - EFS 606
- デバッグ・ジョブ 123
- デフォルト
  - インストール 18
  - ICETOOL でリストする 20
  - デフォルト、インストール 18
- 動的ストレージ調整 (DSA)
  - インストール・オプション 21
  - 限界 702
  - パフォーマンスの向上 627
- 動的呼び出し DFSORT
  - 拡張パラメーター・リストによる 711, 721

動的呼び出し DFSORT (続き)

24 ビット・パラメーター・リストを用いた 721,  
733

動的呼び出しに関する制約事項 439, 440

特殊入出力の処理 368

トラック 625, 693

## [ナ行]

長さ

可変長レコードの LRECL 17

最大レコード 14

変更後の制御ステートメント 603

元の制御ステートメント 602

レコード記述子 (RDW) 17

レコード長リスト 603

日本語文字 14, 582

入出力エラー 368

入出力データ・セット特性の正確な指定 625

入力データ・セット

有効なタイプ 13

要件 13

入力データ・セットの処理

E18 ユーザー出口 379

E38 ユーザー出口 392

入力ファイル

サイズおよび効率 625

入力フィールド 159

入力フェーズ 588

## [ハ行]

ハイパー空間

制限要因 199

定義済み 637

ハイパー分類

使用の利点 637

定義済み 637

パイプ

「SmartBatch パイプ」を参照 18

パフォーマンス

悪影響を与えるオプション 630

アプリケーション設計 623

一時ワークスペース 636

拡張記憶域での VIO の使用 626

組み合わせ手法 624

向上させるオプション 627

効率のよいブロック化 623

最大化 622

主記憶域 631

装置による経過時間の改善 626

データ空間分類 638

パフォーマンス (続き)

データ・セットの指定 625

ハイパー分類 637

分類手法 624

BLDINDEX サポートの使用 643

HIPRMAX 638

ICEGENER 639

JCL 623

ODMAXBF の効果 307

SAS システム用の DFSORT のパフォーマンス・ブ  
ースターの使用 643

SmartBatch パイプ 626

パフォーマンスの最大化 622

パフォーマンスを向上させるオプションの使用 627

パフォーマンスを最大にするためのアプリケーションの  
設計 623, 631

パラメーター・リスト

形式 424, 431

制御ステートメント 423, 431

説明 699

比較

OMIT 制御ステートメント 177

比較演算子 133, 150

比較フィールドの形式と長さのテーブル 134

日付形式

DISPLAY 738

INCLUDE と OMIT 178

OCCUR 514

OUTFIL 257

SORT と MERGE 349

日付定数 137, 150, 158, 252, 330

ビット演算子 143

ビット比較テスト 143

表記の規則 xv

フィールドおよび定数シンボル

概要 555

フィールド形式

制御 349

比較 134

要約 357

ICETOOL 演算子

DISPLAY 471

RANGE 514, 525

SELECT 529

STATS 539

UNIQUE 542

VERIFY 544

複数の出力データ・セット

OUTFIL で作成する 3, 239, 311

不十分な中間記憶装置 693

浮動小数点データ 152, 351, 368

浮動小数点フィールド 358

## 浮動符号形式

- ICETOOL の使用 472
- プログラム DD ステートメント 73, 88, 94, 109
- プログラム制御ステートメント
  - 拡張パラメーター・リスト 431, 434
  - EXEC ステートメントの使用 34
- プログラム呼び出し、定義 5
- プログラム・フェーズ
  - 終了 589
  - 初期設定 586
  - 定義済み 363
  - 入力 588
- プロシージャー、カタログ
  - 指定 32
  - 定義済み 32
- ブロック
  - 最小レコード長 15
- ブロック・セット
  - DFSORT 27
- 文化的環境
  - 「LOCALE」を参照 7
- 分類
  - 概要 13
  - データ空間の使用 628
  - データ・セットの要件 13
  - 定義済み 1
  - 分類する情報を識別する 6
  - 分類するレコードの数の指定 45
  - 分類するレコードの数を指定する 46
  - 分類するレコードの概数を指定する 46
  - ユーザー定義のデータ・タイプ 582, 588
  - レコード 345
- 分類処理の例 648, 671
- 別名
  - DFSORT 32
  - OPTION ステートメント・オプション 233
  - PARM オプション 66
- 変換文字
  - 大文字から小文字 154, 160, 255, 332
  - 小文字から大文字 154, 160, 254, 332
  - ALTSEQ 120, 161, 256, 333
- 変更 120
- 編集マスク
  - ICETOOL DISPLAY 演算子 473, 474
  - OUTFIL 261, 269
- ホーム・ページ (web) 4
- 報告書
  - トレーラー、OUTFIL 284
  - ヘッダー、OUTFIL 279
  - ANSI 紙送り制御文字 242, 249, 278, 279, 284, 290, 292, 297, 299, 302, 307, 308
  - ICETOOL DISPLAY 468, 505

## 報告書 (続き)

- ICETOOL OCCUR 511, 523, 524
- OUTFIL 要素 3, 239
- OUTFIL 用に作成 241, 249

## [マ行]

### マクロ命令

- 「システム・マクロ命令」を参照 421
- マスター・コンソール・メッセージ 26
- 命令フィールド 117
- メジャー・コール 1 616
- メジャー・コール 2 617
- メジャー・コール 3 618
- メジャー・コール 4 619
- メジャー・コール 5 619
- メッセージ
  - マスター・コンソール・メッセージ 26
  - メッセージ・データ・セット 26
  - 戻りコード 26
- メッセージ・データ・セット 26
- メッセージ・データ・セットに印刷するためのメッセージの提供 593
- メッセージ・リスト 605
- 文字定数 137
- 戻りコード
  - DFSORT 26
  - EFS 612
  - ICEGENER 642
  - ICETOOL 552

## [ヤ行]

### ユーザー出口

- アセンブラー・ルーチン
  - 出力フェーズ 386
  - 入力フェーズ 375
- アドレッシング・モードと常駐モード 370
- 概要 362
- 活動化 363
- 規則の要約 371, 375
- 機能 366
- 言語要件 363
- 効率 631
- ユーザー・ルーチンの使用 393, 421
- リンク・エディット 374
- ルーチンの規則 371
- ルーチンの使用 362, 393
- ルーチンのロード 372
- 関係規則 373
- COBOL ルーチン
  - 概要 396

## ユーザー出口 (続き)

### COBOL ルーチン (続き)

出力フェーズ 407

入力フェーズ 400

### DFSORT パフォーマンス 371

E11 375

E15 375, 400

E16 378

E17 379

E18 379

E19 383

E31 386

E32 386

E35 388, 407

E37 392

E38 392

E39 392

E61 384

MODS 制御ステートメントを使用して制御を渡す

173

RECORD 制御ステートメントの使用 340

ユーザー出口へ制御を渡す 173

ユーザー出口ルーチンが DFSORT のパフォーマンスに

与える影響 371

ユーザー出口ルーチンの機能 366, 370

ユーザー出口ルーチンのロード方法 372

ユーザー出口関係規則 373

予期しない異常終了 748

予期しない異常終了からのリカバリー 748

## 要件

### 主記憶域

影響を与える要因 632

出力データ・セット 13

入力データ・セット 13

JCL 29

## [ラ行]

ラベル・フィールド 117

リンク・エディット

パフォーマンス 631

ユーザー出口ルーチン 374

リンケージ・エディター 72, 94

ルーチンの初期設定

E11 ユーザー出口 375

E31 ユーザー出口 386

## レコード

埋め込み 138, 641

可変長

効率 625

記憶域の制約 15

記述子ワード (RDW) 17

## レコード (続き)

切り捨て 138, 641

組み合わせ 169

形式設定 153

合計 2, 355

ユーザー出口による 368

E35 ユーザー出口 391

コピー 170

固有な

OCCUR 演算子 (ICETOOL) 511

SELECT 演算子 (ICETOOL) 527

UNIQUE 演算子 (ICETOOL) 541

最小レコード長 15

最大長 14

再フォーマット設定 327

削除 130, 177

OMIT 制御ステートメントを使用 177

処理順序 8, 152, 164, 165, 335

EFS 612

挿入、削除、および変更 368

重複 355, 511, 527

データ・タイプ 14

ブロック化 623

分類 345

分類する数 46

分類する正確な数 45

分類する見積もり数 46

ユーザー定義のデータ・タイプ 582

ユーザー出口による変更 368

ユーザー出口ルーチンによる引き渡し 375

ユーザー出口ルーチンによる変更 388

EFS 制約 15

OUTFIL の処理 240

レコード処理の順序 612, 616

レコードの値を加算する 2

レコードの値を合計する 2

レコードの合計 355

レコードのコピー

MERGE 制御ステートメントによる制約事項 170

SORT 制御ステートメントのオプション 352

レコードの再配列

「レコードの分類」を参照 345

レコードの削除

E15 ユーザー出口 375

レコードの挿入 368

レコードの追加 368

E15 ユーザー出口 375

レコードの引き渡し

E15 ユーザー出口 375, 400

レコードのブロック化 623

レコードの分類 345

レコードの変更 368

レコードを切り捨てる 138  
レコードを組み込む 2, 130, 177  
    ユーザー定義のデータ・タイプ 582  
レコードを限定する 130, 177  
レコードを再フォーマット設定する 2, 368  
    INREC を使用して 153  
    OUTREC での使用 327  
レコードを削除する 368  
    E15 ユーザー出口 400  
    E35 ユーザー出口 407  
    INCLUDE 制御ステートメントで 130, 177  
    OMIT 制御ステートメントを使用 177  
レコードを省略する 2, 177  
    ユーザー定義のデータ・タイプ 582  
レコードを追加する  
    E15 ユーザー出口 400  
    E35 ユーザー出口 407  
レコードを変更する 2, 368  
    「レコードの再フォーマット設定」も参照 153  
    E15 ユーザー出口 375, 400  
    E35 388  
    E35 ユーザー出口 407  
レコードを編集する  
    参照：レコードを再フォーマット設定する  
レコードを保持する 2  
レコード・タイプ  
    指定 340  
関係規則 373  
関係の例 373  
ロケール  
    制約事項  
        CHALT 186  
        EFS 43, 193  
    定義済み 7  
    INCLUDE および OMIT 処理への影響 139  
    MERGE 処理への影響 170  
論理演算子 152

## [ワ行]

ワークスペース  
    使用 685, 695  
    DFSORT の要件 685

## [数字]

10 進数定数 136  
16 進数表示  
    DISPLAY 演算子 477  
    OCCUR 演算子 516  
16 進定数 138

2 桁年  
    移行 231, 319  
    日付変換 3, 239  
    分類 355  
2 バイト文字セット (DBCS) 順序付け  
    「DBCS 順序付け」を参照 14  
2 バイト文字セット配列支援プログラム  
    「DBCS 順序付け」を参照 582  
2000 年  
    「世紀」ウィンドウの設定 231  
    比較、日付の 151  
    日付の順序付け 349  
    日付変換 257  
24 ビット・パラメーター・リスト  
    形式 424, 431  
    例 435, 438  
24 ビット・パラメーター・リストの形式 424, 430  
4 桁年  
    日付変換 231

## A

A タイプ・メッセージを伴うエラー異常終了の処理  
    749, 750  
ABCODE  
    インストール・オプション 20  
    ABEND コード 38  
ABEND  
    DEBUG 制御ステートメント・オプション 123  
    EXEC PARM オプション 38  
ABSTP  
    処理 748  
    DEBUG 制御ステートメント・オプション 124  
AC (ISCI/ASCII 文字) 形式  
    説明 734  
    INCLUDE ステートメント 134  
    SORT ステートメント 349  
ALIAS ステートメント 372  
ALTSEQ  
    インストール・オプション 20  
ALTSEQ ステートメントの例 121, 122  
ALTSEQ 制御ステートメント 121  
    機能 116  
    使用 120, 122  
    例 121  
    TABLE オプション 120  
AMODE 370, 374  
AQ (代替文字) 形式  
    INCLUDE ステートメント 134  
    SORT ステートメント 349  
ARESALL  
    インストール・オプション 20

ARESALL (続き)  
主記憶域の解放 634  
ARESALL の代わりに RESERVEX を使用 39  
EXEC PARM オプション 38  
OPTION 制御ステートメント・オプション 184

ARESINV  
インストール・オプション 20  
主記憶域の解放 634  
OPTION 制御ステートメント・オプション 184

ASL (ISCI/ASCII 前置き符号) 形式  
説明 736  
INCLUDE ステートメント 134  
SORT ステートメント 349

AST (ISCI/ASCII 後書き符号) 形式  
説明 736  
INCLUDE ステートメント 134  
SORT ステートメント 349

ATTACH  
説明 421  
マクロ命令の書き方 434

AVGLEN  
EXEC PARM オプション 39  
OPTION 制御ステートメント・オプション 185

## B

BI (2 進数) 形式  
説明 734  
DISPLAY 演算子 471  
INCLUDE ステートメント 134  
INREC ステートメント 156  
OCCUR 演算子 514  
OUTFIL ステートメント 260  
RANGE 演算子 525  
SELECT 演算子 529  
SORT ステートメント 349  
STATS 演算子 539  
SUM ステートメント 357  
UNIQUE 演算子 542

BLDINDEX 643

BSAM  
DEBUG 制御ステートメント・オプション 124  
E18 ユーザー出口 380  
E19 ユーザー出口 383  
EXEC PARM オプション 40

## C

CFW  
インストール・オプション 20  
パフォーマンス向上のための使用 627  
OPTION 制御ステートメントで使用 125

CH (文字) 形式  
説明 733  
DISPLAY 演算子 471  
INCLUDE ステートメント 134  
OCCUR 演算子 514  
SELECT 演算子 529  
SORT ステートメント 349  
UNIQUE 演算子 542

CHALT  
インストール・オプション 20  
OPTION 制御ステートメント・オプション 185

CHECK  
インストール・オプション 21  
OPTION 制御ステートメント・オプション 186

CINV  
インストール・オプション 21  
EXEC PARM オプション 40  
OPTION 制御ステートメント・オプション 186

CKPT  
効率 630  
OPTION 制御ステートメント・オプション 187  
SORT 制御ステートメントのオプション 352

CLIST の例 647

CLO/OL (前置きオーバーパンチ符号) 形式  
説明 736  
INCLUDE ステートメント 134  
SORT ステートメント 349

COBEXIT  
インストール・オプション 21  
効率 627  
EXEC PARM オプション 40  
OPTION 制御ステートメント・オプション 187

COBOL  
概要 396  
記憶域の所要量 399  
コピー処理の要件 398  
出力フェーズのユーザー出口 407  
入力フェーズのユーザー出口 400  
ユーザー出口ルーチン 396, 400, 407  
ユーザー出口ルーチンに関する要件 397

COBOL E15 ユーザー出口  
分類のためのレコードの引き渡し 400  
分類のためのレコードの変更 400  
レコードを変更する 414

COBOL E35 ユーザー出口  
レコードの挿入 415  
レコードを変更する 407

COBOL で作成されたサンプル・ルーチン 414, 417

CODE  
ALTSEQ 制御ステートメント 121

COND  
INCLUDE 制御ステートメント・オプション 131



COND (続き)  
 OMIT 制御ステートメント・オプション 178  
 CONVERT パラメーター  
 OUTFIL 制御ステートメント・オプション 241, 275  
 COPY  
 OPTION 制御ステートメント・オプション 188  
 COPY 演算子 (ICETOOL) 455  
 COUNT 演算子 (ICETOOL) 459  
 CSF/FS (浮動符号) 形式  
 説明 735  
 DISPLAY 演算子 471  
 INCLUDE ステートメント 134  
 OCCUR 演算子 514  
 OUTFIL ステートメント 260  
 RANGE 演算子 525  
 SELECT 演算子 529  
 SORT ステートメント 349  
 STATS 演算子 539  
 UNIQUE 演算子 542  
 CSL/LS (符号が左端にある) 形式  
 説明 735  
 SORT ステートメント 349  
 CST/TS (後書き符号) 形式  
 説明 735  
 INCLUDE ステートメント 134  
 SORT ステートメント 349  
 CTO/OT (後置きオーバーパンチ符号) 形式  
 説明 736  
 INCLUDE ステートメント 134  
 SORT ステートメント 349  
 CTRx  
 異常終了処理 750  
 DEBUG 制御ステートメント・オプション 125

## D

D1 形式  
 FIELDS オペランド 600  
 SORT ステートメント 349  
 D2 形式  
 COND オペランド 601  
 INCLUDE ステートメント 134  
 DASD  
 効率 625, 636  
 容量に関する考慮事項 693  
 容量の超過 693  
 DBCS 配列 582  
 DD ステートメント  
 概要 67, 88  
 システム DD ステートメント 70, 92  
 使用 67, 88, 109  
 プログラム DD ステートメント 73, 94

DD ステートメント (続き)  
 要約 30  
 DD ステートメントの使用 67, 88, 109  
 DD 名  
 重複 70, 91  
 DD 名の重複 70, 91, 92  
 DEBUG ステートメントの例 128  
 DEBUG 制御ステートメント  
 機能 116  
 使用 123, 129  
 特殊処理 598  
 例 123, 128  
 DEFAULTS 演算子 (ICETOOL) 462  
 DFSORT 18  
 オプションの指定変更 697  
 概要 1  
 効率の改善 622  
 互換性のあるオペレーティング・システム 5  
 処理順序 8  
 ジョブ制御ステートメント 29, 88, 109  
 出口ルーチン 362  
 動的呼び出し 421  
 入力 / ユーザー出口 / 出力論理の例 366  
 プログラム制御ステートメント 114, 361  
 プログラム・フェーズ 363, 366, 584  
 メッセージ 26  
 ユーザー出口による終了 370  
 ユーザーの EFS プログラムの呼び出し 584  
 呼び出し 4  
 OUTFIL オペランドの処理 243  
 VM 下のゲストとして作動 5  
 DFSORT オプションの指定 / 指定変更 697, 733  
 DFSORT オプションのソースの主な機能 698, 699  
 DFSORT がユーザー・ルーチンへ渡す情報  
 E15 ユーザー出口 376  
 E32 ユーザー出口 387  
 E35 ユーザー出口 389  
 E61 ユーザー出口 385  
 DFSORT の開始  
 「DFSORT の呼び出し」を参照 422  
 DFSORT の終了  
 ユーザー出口による 370  
 E35 ユーザー出口 407  
 EFS プログラムを使用して 593  
 DFSORT の紹介 1, 27  
 DFSORT のホーム・ページ 4  
 DFSORT フェーズ  
 終了 589, 619  
 初期設定 586, 616  
 定義 584  
 入力 588

DFSORT プログラム制御ステートメントの使用 114, 361

DFSORT を呼び出す  
 拡張パラメーター・リスト 431, 434  
 動的 421  
 プログラムから 421, 440  
 方式 4  
 24 ビット・パラメーター・リスト 423, 431  
 JCL の使用 29

DFSPARM DD ステートメント  
 機能 74, 95  
 使用 85, 88, 107, 109  
 定義済み 30

DFSPARM ステートメント  
 PARM オプション 34  
 別名の PARM オプション 66

DFSPARM データ・セット 698

DIAGSIM  
 インストール・オプション 21

DISPLAY 演算子 (ICETOOL) 467

DSA (動的ストレージ調整)  
 インストール・オプション 21  
 限界 702  
 パフォーマンスの向上 627

DSPSIZE  
 インストール・オプション 21  
 パフォーマンスの向上 627  
 EXEC PARM オプション 41  
 OPTION 制御ステートメント・オプション 188

DTn (日付) 形式  
 説明 738  
 DISPLAY 演算子 471  
 OCCUR 演算子 514  
 OUTFIL ステートメント 260

DYNALLOC  
 EXEC PARM オプション 41  
 OPTION 制御ステートメント・オプション 189  
 SORT 制御ステートメントのオプション 352

DYNALLOC=OFF  
 EXEC PARM オプション 42  
 OPTION 制御ステートメント・オプション 191

DYNALOC  
 インストール・オプション 21

DYNAUTO  
 インストール・オプション 21

DYNSPC  
 インストール・オプション 21  
 EXEC PARM オプション 43  
 OPTION 制御ステートメント・オプション 191

## E

E11 ユーザー出口  
 データ・セットのオープン 375  
 ルーチンの初期設定 375

E15 ユーザー出口  
 可変長レコードの LINKAGE SECTION のコーディングの例 405  
 固定長レコードの LINKAGE SECTION のコーディング例 403  
 戻りコード 377, 405  
 レコード長の変更 393  
 レコードの引き渡し (分類アプリケーションおよびコピー・アプリケーションの場合) 375  
 レコードの変更 (分類アプリケーションおよびコピー・アプリケーションの場合) 375  
 COBOL とのインターフェース 401  
 EXEC PARM オプション 44  
 LINKAGE SECTION フィールド (可変長レコードの場合) 403  
 LINKAGE SECTION フィールド (固定長レコードの場合) 403  
 PROCEDURE DIVISION に関する要件 406

E15/E35 戻りコードおよび EXITCK 417, 421

E16 ユーザー出口  
 中間記憶装置の計算違いの処理 378  
 戻りコード 379  
 NMAX 超過時の現行レコードの分類 394

E17 ユーザー出口  
 データ・セットのクローズ 379

E18 ユーザー出口  
 入力データ・セットの処理 379  
 QSAM/BSAM による使用 380  
 VSAM による使用 381

E19 ユーザー出口  
 作業データ・セットへの出力の処理 383  
 QSAM/BSAM による使用 383

E31 ユーザー出口  
 データ・セットのオープン 386  
 ルーチンの初期設定 386

E32 ユーザー出口  
 組み合わせへの入力のみを処理 386  
 戻りコード 387  
 MERGE 制御ステートメントを使用する上での制約事項 173

E35 ユーザー出口  
 戻りコード 390  
 レコード長の変更 395  
 レコードの変更 388  
 COBOL とのインターフェース 407  
 EXEC PARM オプション 44

- E35 ユーザー出口 (続き)
  - LINKAGE SECTION フィールド (可変長レコードの場合) 410
  - LINKAGE SECTION フィールド (固定長レコードの場合) 410
  - PROCEDURE DIVISION に関する要件 413
- E37 ユーザー出口
  - データ・セットのクローズ 392
- E38 ユーザー出口
  - 入力データ・セットの処理 392
  - VSAM による使用 392
- E39 ユーザー出口
  - 出力データ・セットの処理 392
  - QSAM/BSAM による使用 393
  - VSAM による使用 393
- E61 ユーザー出口
  - 使用 385
  - 制御フィールドの変更 384, 396
  - DFSORT がユーザー・ルーチンへ渡す情報 385
- EFS 600
  - インストール・オプション 21
  - 効率 631
  - 終了フェーズ 589
  - 処理 584
  - 初期設定フェーズ 586
  - 使用 582, 619
  - 出口ルーチン 588
  - 入力フェーズ 588
  - フェーズ 584
  - EFS を使用して行えること 589, 593
  - EXEC PARM オプション 43
  - OPTION 制御ステートメント・オプション 193
- EFS インターフェース
  - 機能 583
  - 情報フラグ 604
  - 制御ステートメントの長さ 602
  - 制御ステートメント要求リスト 596
  - 制御ステートメント・ストリング 596, 598
  - 抽出バッファ・オフセット・リスト 603
  - 定義済み 593
  - プログラム・コンテキスト域 603
  - メッセージ・リスト 605
  - レコード長リスト 603
  - D1 形式 600
  - D2 形式 601
  - DFSORT 処置コード 595
- EFS の働き 583, 589
- EFS プログラム
  - アドレッシング・モードと常駐モード 583
  - インターフェース・パラメーター・リスト 593, 606
  - 活動化 583
  - 機能 582, 589
- EFS プログラム (続き)
  - コンテキスト域 603
  - 制御ステートメントの検査、変更、または無視 590
  - データ・セットのオープンおよび初期設定 590
  - データ・セットのクローズ 593
  - 出口ルーチン 593, 607, 608
    - 機能 606
  - プログラムが有効な場合の制約事項 119, 120
  - メッセージの提供 593
  - ユーザーが指定しなければならない戻りコード 612
  - ユーザー出口ルーチン
    - アドレッシング・モードと常駐モード 611
  - 例 616, 619
  - DFSORT の終了 593
- EFS プログラム・ユーザー出口ルーチンによるユーザー定義データ・タイプの処理 593
- EFS01
  - 機能の説明 606
  - パラメーター・リスト 608
  - ユーザー出口ルーチン 607
- EFS02
  - 機能の説明 606
  - パラメーター・リスト 610
  - ユーザー出口ルーチン 608
  - address=0 618
- EFSDPAFT 616
  - DEBUG 制御ステートメント・オプション 125
- EFSDPBFR 616
  - DEBUG 制御ステートメント・オプション 126
- END 制御ステートメント
  - 機能 116
  - 使用 129
  - 例 129
- ENDREC パラメーター
  - OUTFIL 制御ステートメント・オプション 240, 245
- EODAD 381
- EQUALS 6
  - インストール・オプション 21
  - 効率 631
  - EXEC PARM オプション 44
  - MERGE 制御ステートメント・オプション 171
  - OPTION 制御ステートメント・オプション 193
  - SORT 制御ステートメントのオプション 352
- EQUCOUNT
  - 効率 631
  - DEBUG 制御ステートメント・オプション 126
- ERET
  - インストール・オプション 21
- EROPT 381
- ESTAE
  - インストール・オプション 21
  - リカバリー・ルーチン 747

ESTAE (続き)  
  DEBUG 制御ステートメント・オプション 127  
EXEC ステートメント  
  オペランド 35, 65  
  カタログ式プロシージャール 32  
  SORT 32, 72, 94  
  SORTD 34, 73, 94  
  構文 35  
  使用 32, 67  
  制御ステートメントの使用 34  
  定義済み 32  
  PARM オプション 34, 698  
  別名の PARM オプション 66  
EXITCK  
  インストール・オプション 21  
  ユーザー出口の戻りコード 417  
  ICEMAC インストール・オプション 375, 417  
  OPTION 制御ステートメント・オプション 194  
EXLST 380, 384  
EXPMAX インストール・オプション 21, 629, 686  
EXPOLD インストール・オプション 22, 629, 686  
EXPRES インストール・オプション 22, 629, 686

## F

FASTSRT  
  効率 628  
FI (固定小数点) 形式  
  説明 734  
  DISPLAY 演算子 471  
  INCLUDE ステートメント 134  
  OCCUR 演算子 514  
  OUTFIL ステートメント 260  
  RANGE 演算子 525  
  SELECT 演算子 529  
  SORT ステートメント 349  
  STATS 演算子 539  
  SUM ステートメント 357  
  UNIQUE 演算子 542  
FIELDS  
  D1 形式 (EFS) 600  
  INREC 制御ステートメント 154  
  MERGE 制御ステートメント・オプション 170  
  OUTREC 制御ステートメントのオプション 328  
  SORT 制御ステートメントのオプション 346  
  SUM 制御ステートメントのオプション 356  
FIELDS=COPY  
  MERGE 制御ステートメント・オプション 170  
  SORT 制御ステートメントのオプション 352  
FIELDS=(COPY)  
  SORT 制御ステートメントのオプション 352

FILES パラメーター  
  MERGE 制御ステートメント・オプション 171  
  OUTFIL 制御ステートメント・オプション 239, 244  
FILSZ  
  EXEC PARM オプション 45  
  種類の要約 47  
  MERGE 制御ステートメント・オプション 171  
  OPTION 制御ステートメント・オプション 195  
  SORT 制御ステートメントのオプション 352  
FL (浮動小数点) 形式  
  説明 734  
  SORT ステートメント 349  
  SUM ステートメント 357  
FNAMES パラメーター  
  OUTFIL 制御ステートメント・オプション 239, 243  
FORMAT=f  
  INCLUDE 制御ステートメント・オプション 130, 132  
  MERGE 制御ステートメント・オプション 171  
  OMIT 制御ステートメント・オプション 177, 179  
  SORT 制御ステートメントのオプション 351  
  SUM 制御ステートメントのオプション 357  
FSZEST インストール・オプション 22  
FTOV パラメーター  
  OUTFIL 制御ステートメント・オプション 241, 276  
FTP サイト 4

## G

GENER インストール・オプション 22  
GNPAD インストール・オプション 22, 641  
GNTRUNC インストール・オプション 22, 641

## H

HEADER1 パラメーター  
  OUTFIL 制御ステートメント・オプション 241, 279, 283  
HEADER2 パラメーター  
  OUTFIL 制御ステートメント・オプション 241, 289  
HEADER3 パラメーター  
  OUTFIL 制御ステートメント・オプション 298  
HFS 18  
HILEVEL=YES  
  MODS 制御ステートメント・オプション 174  
HIPRMAX  
  インストール・オプション 22  
  効率 638  
  EXEC PARM オプション 47  
  OPTION 制御ステートメント・オプション 199

# I

## ICEGENER

効率 639

戻りコード 642

例 681

ICEGENER 機能 639, 643

ICEMAC インストール・オプション 20, 25

## ICETOOL 443

演算子 445

要約 445

COPY 445, 450, 455

COUNT 445, 459

DEFAULTS 462

DISPLAY 445, 448, 467

MODE 445, 448, 450, 508

OCCUR 445, 449, 510

RANGE 446, 449, 524

SELECT 446, 450, 527

SORT 446, 450, 534

STATS 446, 449, 539

UNIQUE 446, 450, 541

VERIFY 446, 448, 543

簡単なジョブの例 446

コーディング規則 454

サンプル・ジョブの終了 682

シンボルの使用 447

ステートメント 454

制約事項 551

説明 443

パラメーター・リスト・インターフェース 447, 452, 545

プログラムからの呼び出し 545

戻りコード 552

呼び出す 447

例 448, 449, 458, 462, 492, 509, 521, 526, 531, 536, 540, 542, 544

ICETOOL/DFSORT の関係 443

## JCL 444

ステートメント 451

制約事項 453

要約 444

DFSMSG DD ステートメント 444

JOBLIB DD ステートメント 444

STEPLIB DD ステートメント 444

SYMNAMES DD ステートメント 444

SYMNOUT DD ステートメント 444

TOOLIN DD ステートメント 444, 452

TOOLMSG DD ステートメント 444, 451

TOOLIN インターフェース 447, 452, 545

IDRCPCCT インストール・オプション 22

IEBGENER 639

IEFUSI 635

IEXIT インストール・オプション 22

IGNCKPT インストール・オプション 22

INCLUDE 制御ステートメント

関係条件 132

比較演算子 133, 150

機能 115

効率 629

サブストリング比較演算子 142

例 139, 149

論理演算子 152

INCLUDE パラメーター

OUTFIL 制御ステートメント・オプション 241, 245

INCLUDE/OMIT ステートメントに関する注意事項 152

INPFIL 制御ステートメント 120

INREC 制御ステートメント

機能 115

区切りフィールド

ブランクで区切る 156

文字ストリングによる区切り 157

16 進数ストリングの区切り 157

2 進ゼロの区切り 156

桁合わせ 155

使用 153, 167, 168, 169

注意事項 163, 165

入力フィールド 159

例 165, 168, 169

IOMAXBF インストール・オプション 22

# J

## JCL 29

概要 29

カタログ式プロシージャー 72, 73, 94

カタログ式プロシージャー、指定 32

必須 29

プロシージャー、カタログ化する 32

DD ステートメント要約 30

DFSORT 効率の向上 623

EFS コーディング規則 597

EXEC ステートメント 32

JOB ステートメント 32

JCL DD ステートメント 422, 433

JCL が呼び出した DFSORT 700, 711

JCL で DFSORT を実行する 67, 88, 109

JOB ステートメント

使用 31, 32

定義済み 32

JOBLIB DD ステートメント

使用 71, 92

定義済み 30

## L

- LINES パラメーター
  - OUTFIL 制御ステートメント・オプション 241, 278
- LINK 421
  - マクロ命令の書き方 434
- LIST
  - インストール・オプション 22
  - EFS プログラムを使用 593
  - EXEC PARM オプション 48
  - OPTION 制御ステートメント・オプション 201
- LISTX
  - インストール・オプション 22
  - EFS プログラムを使用 593
  - EXEC PARM オプション 49
  - OPTION 制御ステートメント・オプション 202
- LOCALE
  - インストール・オプション 22
  - 効率 630, 631
  - 使用 7
  - EXEC PARM オプション 49
  - OPTION 制御ステートメント・オプション 203

## M

- MAINSIZE
  - 記憶域の割り振り 632
  - 主記憶域の解放 634
  - OPTION 制御ステートメント・オプション 204
  - 参照: SIZE
- MAXLIM
  - インストール・オプション 22
  - 記憶域の割り振り 632
  - 主記憶域の解放 634
- MERGE 制御ステートメント
  - 機能 115
  - 使用 169, 173
  - 例 172, 173
- MINLIM
  - インストール・オプション 23
  - 記憶域の割り振り 632
- MODS 制御ステートメント
  - 機能 116
  - 使用 173, 177
  - 例 176, 177
- MSGCON インストール・オプション 23
- MSGDDN
  - インストール・オプション 23
  - EXEC PARM オプション 50
  - OPTION 制御ステートメント・オプション 205
- MSGPRT
  - インストール・オプション 23

- MSGPRT (続き)
  - 代替形式 51
  - EXEC PARM オプション 50
  - OPTION 制御ステートメント・オプション 206

## N

- NOABEND
  - DEBUG 制御ステートメント・オプション 123
  - EXEC PARM オプション 38
- NOASSIST
  - DEBUG 制御ステートメント・オプション 128
- NOBLKSET
  - 効率 631
  - OPTION 制御ステートメント・オプション 207
- NOCFW
  - OPTION 制御ステートメントで使用 125
- NOCHALT
  - OPTION 制御ステートメント・オプション 185
- NOCHECK
  - OPTION 制御ステートメント・オプション 186
- NOCINV
  - 効率 631
  - EXEC PARM オプション 40
  - OPTION 制御ステートメント・オプション 186
- NODETAIL パラメーター
  - OUTFIL 制御ステートメント・オプション 241, 306
- NOEQUALS
  - EXEC PARM オプション 44
  - MERGE 制御ステートメント・オプション 171
  - OPTION 制御ステートメント・オプション 193
- NOESTAE
  - DEBUG 制御ステートメント・オプション 127
- NOLIST
  - EFS プログラムを使用 593
  - EXEC PARM オプション 48
  - OPTION 制御ステートメント・オプション 201
- NOLISTX
  - EFS プログラムを使用 593
  - EXEC PARM オプション 49
  - OPTION 制御ステートメント・オプション 202
- NOMSGDD インストール・オプション 23
- NOOUTREL
  - EXEC PARM オプション 52
  - OPTION 制御ステートメント・オプション 207
- NOOUTSEC
  - OPTION 制御ステートメント・オプション 207
- NORESET
  - EXEC PARM オプション 55
- NOSOLRF
  - EXEC PARM オプション 58
  - OPTION 制御ステートメント・オプション 218

NOSTIMER  
 EXEC PARM オプション 59  
 OPTION 制御ステートメント・オプション 208

NOSZERO  
 EXEC PARM オプション 60  
 OPTION 制御ステートメント・オプション 222

NOVERIFY  
 EXEC PARM オプション 61  
 OPTION 制御ステートメント・オプション 224

NOVLLONG  
 EXEC PARM オプション 61  
 OPTION 制御ステートメント・オプション 225

NOVLSCMP  
 EXEC PARM オプション 62  
 OPTION 制御ステートメント・オプション 226

NOVLSHRT  
 EXEC PARM オプション 62  
 OPTION 制御ステートメント・オプション 228

NOVSAMIO  
 EXEC PARM オプション 63

NOWRKREL  
 EXEC PARM オプション 64  
 OPTION 制御ステートメント・オプション 230

NOWRKSEC  
 EXEC PARM オプション 64  
 OPTION 制御ステートメント・オプション 231

NVSAMEMT  
 EXEC PARM オプション 63

NZDPRINT  
 EXEC PARM オプション 65  
 OPTION 制御ステートメント・オプション 232

**O**

ODMAXBF  
 インストール・オプション 23  
 EXEC PARM オプション 52  
 OPTION 制御ステートメント・オプション 208  
 OUTFIL 制御ステートメント・オプション 307

OMIT ステートメントの例 179, 180

OMIT 制御ステートメント  
 機能 115  
 効率 629  
 使用 180  
 例 179, 180

OMIT パラメーター  
 OUTFIL 制御ステートメント・オプション 241, 246

OPTION および DEBUG 制御ステートメントの特殊処理 598

OPTION ステートメントの例 233, 238

OPTION 制御ステートメント  
 機能 115

OPTION 制御ステートメント (続き)  
 使用 181, 238  
 特殊処理 598  
 例 233, 238

OUTFIL  
 記憶域限界 204, 308, 633  
 効率 629  
 参照と変更 241, 272, 318  
 数値フィールドに必要な桁数 264  
 テーブル参照と変更 272, 318  
 編集フィールドの形式と長さ 260  
 編集マスク出力フィールドの長さ 264  
 編集マスク符号 263  
 編集マスク・パターン 261  
 報告書の作成 241, 249  
 DD ステートメント 82, 103

outfil DD ステートメント  
 機能 73, 95  
 定義済み 30

OUTFIL ステートメントに関する注意事項 307, 311

OUTFIL ステートメントの例 311, 320, 321, 322

OUTFIL 制御ステートメント  
 機能 115  
 使用 238, 320, 321, 322  
 例 311, 320, 321, 322

OUTREC ステートメントに関する注意事項 335

OUTREC ステートメントの例 337, 339

OUTREC 制御ステートメント  
 機能 115  
 区切りフィールド  
 16 進数ストリングの区切り 330  
 桁合わせ 329  
 使用 327, 339  
 入力フィールド 331  
 分離フィールド  
 現在日付の定数 330  
 ブランクで区切る 329  
 文字ストリングによる区切り 329  
 2 進ゼロの区切り 329  
 例 337, 339

OUTREC パラメーターとの相違点 328

OUTREC パラメーター  
 参照 241, 272, 318

OUTFIL 制御ステートメント・オプション 241, 248, 275

OUTREL  
 インストール・オプション 23  
 EXEC PARM オプション 52

OUTSEC インストール・オプション 23

OVERRGN 635  
 インストール・オプション 23  
 主記憶域の解放 634

OVFLO  
インストール・オプション 23  
EXEC PARM オプション 53  
OPTION 制御ステートメント・オプション 209

## P

PAD  
インストール・オプション 23  
EXEC PARM オプション 53  
OPTION 制御ステートメント・オプション 210  
PAGEHEAD パラメーター  
OUTFIL 制御ステートメント 300  
PARM オプション  
別名の PARM オプション 66  
PARMDDN インストール・オプション 23  
PD (パック 10 進数) 形式  
説明 734  
DISPLAY 演算子 471  
INCLUDE ステートメント 134  
OCCUR 演算子 514  
OUTFIL ステートメント 260  
RANGE 演算子 525  
SELECT 演算子 529  
SORT ステートメント 349  
STATS 演算子 539  
SUM ステートメント 357  
UNIQUE 演算子 542  
VERIFY 演算子 544  
PD0 (パック 10 進数の一部) 形式  
OUTFIL 制御ステートメント 260  
SORT ステートメント 349

## Q

QSAM  
データ・セット 13  
データ・セットに関する考慮事項 16  
E18 ユーザー出口 380  
E19 ユーザー出口 383

## R

RANGE 演算子 (ICETOOL) 524  
RECORD 制御ステートメント  
機能 116  
コーディングの注意 343  
使用 340  
例 344  
REGION  
記憶域の決定 632  
記憶域の割り振り 632

REGION (続き)  
サイズ 632  
主記憶域の解放 634  
REMOVECC パラメーター  
OUTFIL 制御ステートメント・オプション 307  
RENT 372  
RESALL  
インストール・オプション 23  
EXEC PARM オプション 54  
OPTION 制御ステートメント・オプション 211  
RESERVEVEX  
参照: ARESALL EXEC PARM option  
RESET  
インストール・オプション 23  
OPTION 制御ステートメント・オプション 212  
RESINV 635  
インストール・オプション 23  
OPTION 制御ステートメント・オプション 212  
REXX の例 647  
RMODE 374

## S

SAS  
SAS システム用の DFSORT のパフォーマンス・ブ  
ースター 643  
SAVE パラメーター  
OUTFIL 制御ステートメント・オプション 241, 247  
SDB  
インストール・オプション 23  
EXEC PARM オプション 55  
OPTION 制御ステートメント・オプション 213  
SDB (システム決定ブロック・サイズ) インストール・  
オプション 82, 103  
SDBMSG インストール・オプション 23  
SECTIONS パラメーター  
OUTFIL 制御ステートメント・オプション 241, 296  
SELECT 演算子 (ICETOOL) 527  
SIZE  
インストール・オプション 24  
記憶域の割り振り 632  
主記憶域の解放 634  
EXEC PARM オプション 56  
MERGE 制御ステートメント・オプション 171  
OPTION 制御ステートメント・オプション 195, 216  
SORT 制御ステートメントのオプション 352  
SKIP パラメーター  
OUTFIL 制御ステートメント 298  
SKIPREC  
効率 629  
EXEC PARM オプション 57  
MERGE 制御ステートメント・オプション 171



- SKIPREC (続き)
  - OPTION 制御ステートメント・オプション 216
  - SORT 制御ステートメントのオプション 353
- SmartBatch パイプ
  - 考慮事項 18
  - 分類処理例 669
  - ICETOOL 552
  - OUTFIL の例 319
- SMF
  - インストール・オプション 24
  - 日付 (DTn) および時間 (TMn) 261, 472, 514, 738
  - OPTION 制御ステートメント・オプション 217
- SNAP ダンプ 616
- SNAP ダンプの要求 616
- SOLRF
  - インストール・オプション 24
  - EXEC PARM オプション 58
  - OPTION 制御ステートメント・オプション 218
- SORT 演算子 (ICETOOL) 534
- SORT カタログ式プロシージャー 32, 34, 72, 94
- SORT ステートメントに関する注意事項 353
- SORT ステートメントの例 353, 355
- SORT ステートメント・イメージの例 424
- SORT 制御ステートメント
  - 機能 114
  - 使用 345, 355
  - フィールド形式 349
  - 例 353, 355
  - EQUALS の効果 346
- SORTCKPT DD ステートメント
  - 機能 74, 95
  - 使用 84, 105, 106
- SORTCNTL DD ステートメント
  - 機能 74, 95
  - 使用 84, 85, 106, 107
  - 定義済み 31
- SORTCNTL データ・セット 698, 699
- SORTD カタログ式プロシージャー 73, 94
- SORTDD
  - OPTION 制御ステートメント・オプション 219
- SORTDIAG DD ステートメント
  - 機能 74, 95
  - 使用 88, 109
  - 定義済み 31
- SORTDKdd DD ステートメント
  - 機能 74, 95
  - 使用 88, 109
- SORTIN
  - OPTION 制御ステートメント・オプション 220
- SORTIN DD ステートメント
  - 機能 73, 94
  - 使用 75, 77, 96, 99
- SORTIN DD ステートメント (続き)
  - 定義済み 30
- SORTINnn DD ステートメント
  - 機能 73, 94
  - 使用 77, 79, 99, 100
  - 重複 70, 91
  - 定義済み 30
- SORTLIB
  - ICEMAC インストール・オプション 75, 96
- SORTLIB DD ステートメント
  - 機能 73, 94
  - 使用 74, 75, 95, 96
  - 定義済み 30
- SORTLIB インストール・オプション 24
- SORTMODS DD ステートメント
  - 機能 74, 95
  - 定義済み 31
- SORTOUT
  - OPTION 制御ステートメント・オプション 220
  - OUTFIL DD 名 240
- SORTOUT DD ステートメント
  - 機能 73, 95
  - 使用 82, 84, 103, 105
  - 定義済み 30
- SORTSNAP DD ステートメント
  - 機能 74, 95
  - 使用 88, 109
  - 定義済み 31
- SORTWKdd DD ステートメント
  - 機能 73, 94
  - 使用 79, 100, 103
  - 重複 70, 91
  - 定義済み 30
- SORT、MERGE、INCLUDE、および OMIT 制御ステートメントの形式 600
- SPANINC
  - インストール・オプション 24
  - EXEC PARM オプション 58
  - OPTION 制御ステートメント 221
- SPLIT パラメーター
  - OUTFIL 制御ステートメント・オプション 241, 248
- STARTREC パラメーター
  - OUTFIL 制御ステートメント・オプション 240, 245
- STATS 演算子 (ICETOOL) 539
- STEPLIB DD ステートメント
  - 使用 71, 92
  - 定義済み 30
- STIMER
  - インストール・オプション 24
  - EXEC PARM オプション 59
- STOPAFT
  - 効率 629

STOPAFT (続き)  
EXEC PARM オプション 59  
MERGE 制御ステートメント・オプション 171  
OPTION 制御ステートメント・オプション 221  
SORT 制御ステートメントのオプション 353  
SUM ステートメントに関する注意事項 357, 359  
SUM ステートメントの例 359, 361  
SUM 制御ステートメント 361  
機能 116  
合計フィールド 356  
効率 630  
使用 361  
説明 355  
例 359, 361  
SVC インストール・オプション 24  
SYMNAMES DD ステートメント  
機能 73  
定義済み 30  
SYMNOUT DD ステートメント  
機能 73  
定義済み 30  
SYNAD 380, 384  
SYSABEND DD ステートメント  
使用 72, 94  
定義済み 31  
SYSIN DD ステートメント  
使用 71, 92  
定義済み 30  
SYSIN データ・セット 699  
SYSLIN DD ステートメント  
使用 73, 94  
定義済み 31  
SYSLMOD DD ステートメント  
使用 73, 94  
定義済み 31  
SYSMDUMP DD ステートメント  
使用 72, 94  
定義済み 31  
SYSOUT DD ステートメント  
使用 72, 93  
定義済み 30  
SYSPRINT DD ステートメント  
使用 72, 94  
定義済み 31  
SYSUDUMP DD ステートメント  
使用 72, 94  
定義済み 31  
SYSUT1 DD ステートメント  
使用 72, 94  
定義済み 31  
SZERO  
インストール・オプション 24

SZERO (続き)  
EXEC PARM オプション 60  
OPTION 制御ステートメント・オプション 222

## T

TEXIT インストール・オプション 24  
TMAXLIM  
インストール・オプション 24  
記憶域の割り振り 632  
主記憶域の解放 634  
TMn (時間) 形式  
説明 739  
DISPLAY 演算子 471  
OCCUR 演算子 514  
OUTFIL ステートメント 260  
TRAILER1 パラメーター  
OUTFIL 制御ステートメント・オプション 241,  
283, 289  
TRAILER2 パラメーター  
OUTFIL 制御ステートメント・オプション 241, 296  
TRAILER3 パラメーター  
OUTFIL 制御ステートメント・オプション 302  
TRUNC  
インストール・オプション 24  
EXEC PARM オプション 60  
OPTION 制御ステートメント・オプション 223  
TYPE  
RECORD 制御ステートメントの オプション 340

## U

UNIQUE 演算子 (ICETOOL) 541  
USEWKDD  
OPTION 制御ステートメント・オプション 224

## V

VERIFY  
インストール・オプション 24  
効率 631  
EXEC PARM オプション 61  
OPTION 制御ステートメント・オプション 224  
VERIFY 演算子 (ICETOOL) 543  
VIO  
インストール・オプション 24  
ICEMAC インストール・オプション 88, 109  
VLFILL パラメーター  
OUTFIL 制御ステートメント・オプション 276  
VLLONG  
インストール・オプション 24  
EXEC PARM オプション 61

VLLONG (続き)  
 OPTION 制御ステートメント・オプション 225

VLSCMP  
 インストール・オプション 24  
 EXEC PARM オプション 62  
 OPTION 制御ステートメント・オプション 226

VLSHRT  
 インストール・オプション 24  
 EXEC PARM オプション 62  
 OPTION 制御ステートメント・オプション 228

VLTRIM パラメーター  
 OUTFIL 制御ステートメント・オプション 277

VSAM  
 キー順データ・セット (KSDS) 17  
 最大レコード・サイズ  
 INREC 制御ステートメントを使用 164, 336  
 データ・セット 13  
 データ・セットに関する考慮事項 17  
 ユーザー出口機能 369  
 E18 ユーザー出口 381  
 E38 ユーザー出口 392  
 E39 ユーザー出口 393  
 RECORD 制御ステートメントの使用 340

VSAMBSP インストール・オプション 25

VSAMEMT  
 インストール・オプション 25  
 EXEC PARM オプション 63  
 OPTION 制御ステートメント・オプション 229

VSAMIO  
 インストール・オプション 25  
 EXEC PARM オプション 63  
 OPTION 制御ステートメント・オプション 230

VTOF パラメーター  
 OUTFIL 制御ステートメント・オプション 275

## W

Web 4  
 Web サイト 4

WRKREL  
 インストール・オプション 25  
 EXEC PARM オプション 64  
 OPTION 制御ステートメント・オプション 230

WRKSEC  
 インストール・オプション 25  
 EXEC PARM オプション 64  
 OPTION 制御ステートメント・オプション 231

## X

XCTL  
 使用 421

XCTL (続き)  
 マクロ命令の書き方 434

## Y

Y2 形式  
 説明 736  
 INCLUDE および OMIT における 149, 178  
 OUTFIL OUTREC での 257  
 SORT および MERGE における 349

Y2PAST  
 インストール・オプション 25  
 EXEC PARM オプション 65  
 MERGE 制御ステートメント・オプション 171  
 OPTION 制御ステートメント・オプション 231  
 SORT 制御ステートメントのオプション 353

## Z

ZD (ゾーン 10 進数) 形式  
 説明 733  
 DISPLAY 演算子 471  
 INCLUDE ステートメント 134  
 OCCUR 演算子 514  
 OUTFIL ステートメント 260  
 RANGE 演算子 525  
 SELECT 演算子 529  
 SORT ステートメント 349  
 STATS 演算子 539  
 SUM ステートメント 357  
 UNIQUE 演算子 542  
 VERIFY 演算子 544

ZDPRINT  
 インストール・オプション 25  
 EXEC PARM オプション 65  
 OPTION 制御ステートメント・オプション 232







プログラム番号: 5740-SM1

Printed in Japan

SC88-7061-04



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