

z/OS



DFSORT: ご使用の手引き

z/OS



DFSORT: ご使用の手引き

お願い

本書および本書で紹介する製品をご使用になる前に、197 ページの『特記事項』に記載されている情報をお読みください。

本書は、z/OS (5694-A01) のバージョン 1 リリース 10 および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

本書は SD88-6335-01 の改訂版です。

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

原典： SC26-7527-03

z/OS

DFSORT: Getting Started

発行： 日本アイ・ビー・エム株式会社

担当： ナショナル・ランゲージ・サポート

第4版第1刷 2008.12

© Copyright International Business Machines Corporation 1983, 2008. All rights reserved.

目次

図	vii
表	ix
本書について	xi
本書の使い方	xi
必要な製品知識	xii
インターネット上の z/OS DFSORT 資料へのアクセス	xiii
LookAt を使用してメッセージ説明を検索する方法	xiii

第 1 部 入門 1

第 1 章 DFSORT とは	3
ワールド・ワイド・ウェブ上の DFSORT	3
DFSORT FTP サイト	3
データ・セット、レコード、およびフィールド	3
データ・セットのソート	4
データ・セットのマージ	5
データ・セットのコピー	6
DFSORT でほかにできること	6
DFSORT ジョブの作成および実行	7
ジョブの作成	7
DFSORT 制御ステートメントの要約	7
ジョブの実行	8
サンプル・データ・セットの作成および使用	8

第 2 部 JCL および DFSORT 制御 ステートメントの書き方を学ぶ. . . . 13

第 2 章 データ・セットのソート、マージ、およびコピー	15
データ・セットのソート	15
複数のフィールドによるソート	17
ステートメントの継続	19
コメント・ステートメント	19
データ・セットを直接ソートするための JCL	19
データ・セットのマージ	21
MERGE 制御ステートメントの作成	22
データ・セットを直接マージするための JCL	23
VB データ・セットについての考慮事項	24
開始位置	25
短制御フィールド	25
データ・セットのコピー	26
SORT、MERGE、または OPTION ステートメントへの COPY の指定	26
データ・セットを直接コピーするための JCL	26

第 3 章 レコードの組み込みまたは省略	29
INCLUDE ステートメントの作成	30

OMIT ステートメントの作成	32
INCLUDE および OMIT で許容される比較	33
定数の作成	35
文字ストリング	35
16 進数ストリング	35
10 進数	35
INCLUDE および OMIT の数値テスト	36
INCLUDE および OMIT の場合のサブストリング検索	37
VB データ・セットについての考慮事項	37
開始位置	37
短制御フィールド	38

第 4 章 レコードの合計.	39
SUM ステートメントの作成	39
重複制御フィールドを持つレコードを抑止する	41
オーバーフローの処理	42
VB データ・セットについての考慮事項	42
開始位置	42
短要約フィールド	43

第 5 章 固定フィールドを持つレコードの再フォーマット	45
BUILD または FIELDS を使用したソート後のレコードの再フォーマット	46
フィールドを予約スペースに再配列する	48
2 進ゼロの挿入	48
ブランクの挿入	49
ストリングの挿入	51
文字ストリング	51
16 進数ストリング	52
基本報告書のセットアップ	52
大文字から小文字への変更	53
数値フィールドから異なるフォーマットへの変換	54
数値フィールドの編集	55
データを 16 進数で表示する	58
数値フィールドおよび定数に算術演算を実行する	59
ルックアップおよび変更	60
データの左寄せと右寄せ	61
データの左/右詰め	64
OVERLAY を使用したレコードの再フォーマット	68
OVERLAY を使用したレコードの拡張	70
IFTHEN を使用したレコードの再フォーマット	71
ソート前にレコードを再フォーマットする	74
INREC と一緒に他のステートメントを使用する	74
値を合計するときのオーバーフローを防止する	75
シーケンス番号の挿入	76
VB データ・セットについての考慮事項	79
RDW	80
開始位置および桁	80
可変データ	81

第 6 章 可変フィールドを持つレコードの再フォーマット 83

BUILD および OVERLAY での %nn 解析済みフィールドの使用	83
IFTHEN での %nn 解析済みフィールドの使用	85
BUILD および OVERLAY で %nn フィールドを使用できる項目	87
PARSE パラメーター	88

第 7 章 複数の出力データ・セットおよび報告書の作成 89

複数の同一コピーの作成	90
相対レコード番号による選択および抽出	91
廃棄レコードの組み込み、省略、および保管	93
再フォーマット	95
繰り返し	97
分割	98
報告書の作成: OUTFIL 対 ICETOOL	100
OUTFIL を使用した報告書の作成	101
データ	101
ヘッダー	102
トレーラーおよび統計	105
データまたは紙送り制御文字がない場合	107
ヘッダーとトレーラーに関する VB データ・セットの考慮事項	107
セクション	108
FB から VB への変換	111
VB から FB への変換	113

第 8 章 プログラムからの DFSORT の呼び出し 115

制御ステートメントの受け渡し	115
COBOL プログラムからの DFSORT の呼び出し	116
レコードのソート	116
レコードのマージ	119
COBOL FASTSORT によるソート	121
PL/I プログラムからの DFSORT の呼び出し	121

第 9 章 インストール・デフォルトの指定変更 123

JCL EXEC ステートメントへの PARM パラメーターの指定	124
DFSPARM への OPTION 制御ステートメントの指定	124

第 10 章 DFSORT の効率的な使用 127

主記憶域を十分に提供する	127
メモリー・オブジェクト・ソート、ハイパー・ソート、およびデータ・スペース・ソートを許可する	128
作業データ・セットに高速ディスクを使用する	128
INREC を使用して不要なフィールドを除去する	128
INCLUDE または OMIT を使用して不要なレコードを除去する	129
STOPAFT および SKIPREC を使用して不要なレコードを除去する	129

SUM を使用してレコードを統合する	129
OUTFIL を使用して複数の出力データ・セットを作成する	129
DFSORT 制御ステートメントを使用してプログラム・ロジックを置き換える	130
COBOL の場合には FASTSORT を使用する	130
パフォーマンスを低下させる可能性のあるオプションを避ける	130

第 3 部 ICETOOL の使い方を学ぶ 131

第 11 章 ICETOOL ユーティリティの使用 133

ICETOOL オペレーター	133
サンプル入力データ・セット	134
必要な JCL ステートメントの作成	135
ICETOOL コメント・ステートメントおよびブランク・ステートメント	136
数値フィールドの統計を印刷する	136
オペレーター・ステートメントの継続	138
VB データ・セット・レコード長の統計	139
同一のソート済みデータ・セットの作成	139
ソート済みデータ・セットのいろいろなサブセットを作成する	142
複数の未ソート・データ・セットを作成する	144
ある範囲内の値をカウントする	145
簡潔報告書を印刷する	147
調整済み報告書を印刷する	148
フォーマット項目を使用する	150
編集マスク	150
先行ゼロ	152
編集パターン	152
統計なし	153
除法	153
先行、浮動、および末尾の文字	154
分割した報告書を印刷する	154
フィールドの出現回数を印刷する	157
フィールド出現回数によってレコードを選択する	159
いろいろなデータ・セットからのフィールドを結合する	162
いろいろなデータ・セットからのレコードを突き合わせる	165
完全な ICETOOL ジョブと TOOLMSG 出力	169

第 4 部 シンボルの使い方を学ぶ 175

第 12 章 シンボルの定義および使用 177

SYMNAMES データ・セットの作成	177
フィールドを表すシンボルの定義	178
DFSORT ステートメントでフィールドにシンボルを使用する方法	179
ICETOOL オペレーターでフィールドにシンボルを使用する方法	180
定数を表すシンボルの定義と使用	181

第 5 部 付録	185
付録 A. サンプル・データ・セットの作成	187
付録 B. サンプル・データ・セットの記述	189
付録 C. 制御ステートメントの処理順序	193
付録 D. アクセシビリティ	195

支援機能の使用	195
ユーザー・インターフェースのキーボード・ナビゲーション	195
z/OS 情報	195
特記事項	197
プログラミング・インターフェース情報	198
商標	198
索引	199



1. 比較演算子	31	4. SORT コマンドを持つサンプル PL/I プログラ	
2. SORT コマンドを持つサンプル COBOL プロ		ム	122
グラム	117	5. 完全な ICETOOL ジョブ	169
3. MERGE コマンドを持つサンプル COBOL プ		6. 完全な TOOLMSG データ・セット	171
ログラム	120	7. 制御ステートメントの処理順序	193

表

1. 関連資料	xii	33. 編集マスク・パターン	57
2. DFSORT は情報を昇順および降順に配置する	5	34. 学部コースと価格を変更した本	69
3. 一般的に使用されるデータ・フォーマット	5	35. 新価格と旧価格	70
4. DFSORT は 2 つのデータ・セットをマージして 1 つのデータ・セットにする	5	36. 提案された本の割引	72
5. 学部コース (Course Department) の昇順でソートされたサンプル・ブックストア・データ・セット	10	37. INREC を使用した場合の出版社 (Publisher)、在庫数 (Number in Stock)、および販売数 (Number Sold) だけの書き込み	75
6. 学部 (Department) によってソートする SORT ステートメントの作成ステップ	15	38. 要約フィールドの埋め込み	76
7. 学部コース (Course Department) の昇順でソートされたサンプル・ブックストア・データ・セット	16	39. 要約フィールドの埋め込み	76
8. 価格の降順でソートされたサンプル・ブックストア・データ・セット	17	40. コース別の本価格の合計	77
9. 複数のフィールドでソートされたサンプル・ブックストア・データ・セット	18	41. コース別の本の価格の合計 (シーケンス番号付き)	77
10. 学部コースおよびブック・タイトルでソートされたサンプル・ブックストア・データ・セット	22	42. 複数の出力データ・セット・ジョブを得るための OUTFIL ステートメントの作成	90
11. 学部コースおよびブック・タイトルでソートされた 5 つの新規レコード	22	43. コピーの場合の相対レコード番号	91
12. 5 つの新規レコードでマージされたサンプル・ブックストア・データ・セット	23	44. ソートの場合の相対レコード番号	92
13. オーダーする必要がある本についての INCLUDE ステートメントの作成ステップ	30	45. ENGLOUT データ・セット内の出力レコード	94
14. 販売数 (Number Sold) が在庫数 (Number In Stock) よりも大きくなっている本	31	46. HISTOUT データ・セット内の出力レコード	94
15. 販売数 (Number Sold) が在庫数 (Number In Stock) よりも大きくなっている COR 本	32	47. PSYCHOUT データ・セット内の出力レコード	94
16. 教育コースに必要でない本を省いたソート済みデータ・セット	33	48. RESTOUT データ・セット内の出力レコード	95
17. 許容されるフィールドとフィールドの比較	33	49. INREC の場合の制御ステートメントおよび対応するレコード	128
18. 許容されるフィールドと定数の比較	34	50. ブランク・ステートメントとコメント・ステートメントを作成するステップ	136
19. 価格を求める SUM ステートメントの作成ステップ	39	51. STATS オペレーターの作成ステップ	137
20. English 学部の価格の合計	40	52. SORT オペレーターの作成ステップ	140
21. English 学部の価格の合計	40	53. SORT オペレーターのための JCL ステートメントを作成するステップ	141
22. それぞれの出版社ごとの在庫数と販売数の合計	41	54. 出版社 VALD および WETH の本	141
23. 重複を削除した出版社のリスト	42	55. SORT オペレーターの作成ステップ	142
24. レコードを再フォーマットする OUTREC ステートメントの作成ステップ	46	56. SORT オペレーターのための JCL ステートメントを作成するステップ	143
25. 出版社 (Publisher)、在庫数 (Number In Stock)、および販売数 (Number Sold) フィールドだけの書き込み	47	57. 市区町村によってソートされたカリフォルニアに関するレコード	144
26. フィールドの再配列	48	58. 市区町村によってソートされたコロラドに関するレコード	144
27. 2 進ゼロの挿入	48	59. 編集マスク・パターン	150
28. ブランクを挿入した後の出力	50	60. 5 冊以上の本が使用されている出版社からの本	160
29. 報告書の出力	53	61. 結合用の REGION.IN1 データ・セット	163
30. BI から ZD への変換	55	62. 結合用の REGION.IN2 データ・セット	163
31. BI から FS への変換	55	63. 結合用の REGION.OUT データ・セット	163
32. PD 値を持つ入力レコード	56	64. REGION.IN1 からの T1 データ・セット・フィールド	164
		65. REGION.IN2 からの T1 データ・セット・フィールド	164
		66. SORT.SAMPIN からの T1 データ・セット・フィールド	166
		67. SORT.SAMPADD からの T1 データ・セット・フィールド	166
		68. COURSE.MATCH 出力	167

69.	COURSE.INONLY 出力	167	73.	SORT.SAMPIN および SORT.SAMPADD のフ	
70.	COURSE.ADDONLY 出力	168		ィールド記述	189
71.	フィールドを表すシンボルの定義ステップ	178	74.	SORT.BRANCH のフィールド記述	189
72.	定数を表すシンボルの定義ステップ	182			

本書について

「z/OS DFSORT: ご使用の手引き」は、DFSORT™ (データ機能分類プログラム) のための使用者の手引きであるとともにチュートリアルとなっています。 z/OS V1R5 を初めてご使用になるユーザーで、DFSORT および DFSORT の ICETOOL ユーティリティを使用してデータ・セットを処理するさまざまな方法を学習する必要がある場合に、本書をご利用ください。 DFSORT の新規ユーザーも経験のあるユーザーも、本書を、DFSORT および ICETOOL で使用可能な多くの機能の全般的な手引きとして利用できます。本書では、多数の例と共に、JCL、制御ステートメント、および DFSORT と ICETOOL の諸機能を紹介しています。「z/OS DFSORT: ご使用の手引き」をお読みいただくと、本書で紹介する全トピックの詳細が記載されている「z/OS DFSORT アプリケーション・プログラミング・ガイド」の大部分をお読みいただく必要がなくなるかもしれません。

本書の各章では、ユーザーがジョブ制御言語 (JCL) を使用したことがあり、データ・セットの処理方法を理解しているものと想定しています。また、ユーザーは、ご使用のサイトでどんなデータ・セットが使用可能であるかを知っておいてください。

本書の使い方

本書では、DFSORT ジョブを作成し実行依頼するのに必要なすべての情報および命令を紹介します。JCL および DFSORT プログラム制御ステートメントを書くことによって、DFSORT を使用できます。

新規ユーザーは「z/OS DFSORT: ご使用の手引き」を始めから終わりまでお読みください。本書で説明している各タスクは、前のタスクで得られた知識を基にして構築しています。目次には、各章に含まれているメインタスクと要約がリストしてあります。これらのタスクを行った経験がある場合は、15 ページの『第 2 章 データ・セットのソート、マージ、およびコピー』から始まるチュートリアルに直接進むことができます。

『第 1 章 DFSORT とは』では、ソート、マージ、およびコピーの基本原則の概要を紹介し、本書の例に必要なサンプル・データ・セットの作成方法と使用方法を説明しています。

15 ページの『第 2 章 データ・セットのソート、マージ、およびコピー』から 127 ページの『第 10 章 DFSORT の効率的な使用』では、JCL および DFSORT 制御ステートメント (例えば、SORT、MERGE、OPTION、INCLUDE、OMIT、INREC、OUTREC、SUM、および OUTFIL) を書くことによって DFSORT ジョブを作成し処理する方法を示します。これによって、サブセット化、再フォーマット、合計、および報告書作成などの各種の操作をレコードに行いながら、ソート済み、マージ済み、またはコピー済みの出力データ・セットを作成します。

133 ページの『第 11 章 ICETOOL ユーティリティの使用』では、JCL および ICETOOL ステートメントを書くことによって ICETOOL ジョブを作成し処理する方法を示します。ICETOOL は多目的の DFSORT ユーティリティであり、

DFSORT の諸機能を使用して、単一ステップで 1 つ以上のデータ・セットに複数の操作を実行します。ICETOOL の 13 個のオペレーター (COPY、COUNT、DEFAULTS、DISPLAY、MODE、OCCUR、RANGE、SELECT、SORT、SPLICE、STATS、UNIQUE、および VERIFY) は、DFSORT の機能を相当拡張します。

177 ページの『第 12 章 シンボルの定義および使用』では、フィールドおよび定数のシンボルを定義する方法、DFSORT 制御ステートメントおよび ICETOOL オペレーターでそれらのシンボルを使用する方法を示します。

その後、いくつかの付録と索引があります。

必要な製品知識

本書を効率よく使用するには、以下の情報を熟知している必要があります。

- ジョブ制御言語 (JCL)
- データ管理

さらに、以下の関連資料で提示されている情報も理解している必要があります。

表 1. 関連資料

資料タイトル	資料番号
<i>z/OS DFSORT</i> アプリケーション・プログラミング・ガイド	SD88-6331
<i>z/OS MVS JCL</i> 解説書	SA88-8569
<i>z/OS MVS JCL</i> ユーザーズ・ガイド	SA88-8570
<i>z/OS DFSMS</i> データ・セットの使用法	SC88-9114

「*z/OS DFSORT: ご使用の手引き*」は、広範囲にわたる DFSORT ライブラリーの一部です。以下の資料は、DFSORT を効率よく処理するのに役立ちます。

タスク	資料タイトル	資料番号
アプリケーション・プログラミング	<i>z/OS DFSORT</i> アプリケーション・プログラミング・ガイド	SD88-6331
DFSORT の計画およびカスタマイズ	<i>z/OS DFSORT</i> インストールおよびカスタマイズ	SD88-6332
障害の診断とメッセージの解釈	<i>z/OS DFSORT</i> メッセージ、コードおよび診断の手引き	SD88-6333
DFSORT の調整	<i>z/OS DFSORT</i> チューニング・ガイド	SD88-6334

インターネット上の z/OS DFSORT 資料へのアクセス

IBM では、CD-ROM で入手できるソフトコピー資料のほかに、インターネット上の非ライセンス z/OS ソフトコピー資料へのアクセスも提供します。インターネット上で z/OS 資料を見つけるには、まず z/OS のホーム・ページ (<http://www.ibm.com/servers/eserver/zseries/zos>) にアクセスしてください。

この Web サイトから Library アイコンを選択することによって、z/OS ソフトコピー資料に直接リンクすることができます。IBM Publications Center にリンクして、印刷資料を注文することもできます。

LookAt を使用してメッセージ説明を検索する方法

LookAt というオンライン機能により、多くの IBM® メッセージや、いくつかのシステム異常終了およびコードに関する説明を検索できます。LookAt では、通常、該当メッセージの説明がただちに表示されるため、従来の方よりも短時間で、必要な情報を検索することができます。

以下の場所から LookAt を使用して、z/OS® エlementおよびフィーチャー、z/VM®, および VSE に関する IBM メッセージを検索できます。

- インターネット。LookAt Web サイトの <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/> から直接 IBM メッセージ説明にアクセスできます。
- ご使用の z/OS TSO/E ホスト・システム。TSO/E コマンド行 (たとえば、TSO/E プロンプト、ISPF、または OMVS を実行中の z/OS UNIX® システム・サービス) から LookAt を使用して、ご使用の z/OS または z/OS.e システムに、IBM メッセージ説明にアクセスするためのコードをインストールすることができます。
- ご使用の Windows® ワークステーション。Windows DOS コマンド行から LookAt を使用して、z/OS Collection (SK3T-4269) にある IBM メッセージ説明にアクセスするためのコードをインストールすることができます。
- ワイヤレス・ハンドヘルド・デバイス。ワイヤレス・アクセスおよびインターネット・ブラウザを備えたハンドヘルド・デバイスによって LookAt モバイル・エディションを使用できます (たとえば、ポケット PC 用の Internet Explorer、Blazer、Palm OS 用の Eudora、または Linux ハンドヘルド・デバイス用の Opera など)。LookAt Web サイトから LookAt モバイル・エディションにリンクしてください。

LookAt をホスト・システムまたは Windows ワークステーションにインストールするためのコードは、z/OS Collection (SK3T-4269) のディスクから、または LookAt Web サイトから (**Download** をクリックし、要件に合ったプラットフォーム、リリース、コレクション、およびロケーションを選択します) 入手することができます。ダウンロード・プロセス時に入手できる LOOKAT.ME ファイルに詳細情報があります。

第 1 部 入門

第 1 章 DFSORT とは	3
ワールド・ワイド・ウェブ上の DFSORT	3
DFSORT FTP サイト	3
データ・セット、レコード、およびフィールド	3
データ・セットのソート	4
データ・セットのマージ	5
データ・セットのコピー	6
DFSORT でほかにできること	6
DFSORT ジョブの作成および実行	7
ジョブの作成	7
DFSORT 制御ステートメントの要約	7
ジョブの実行	8
サンプル・データ・セットの作成および使用	8

第 1 章 DFSORT とは

DFSORT は、z/OS のための IBM の高性能のソート、マージ、コピー、分析、および報告書作成プロダクトです。

DFSORT を使用することにより、データ・セットをソート、マージ、およびコピーできます。ユーザーは DFSORT を使用して、名前のリストのアルファベット順配列のような単純な作業を行うこともできますし、目録の作成や請求書作成発行システムの実行のような複雑な作業に役立てることもできます。DFSORT は、レコード・レベル、フィールド・レベル、およびビット・レベルで、多方面のデータ処理機能を提供します。

ワールド・ワイド・ウェブ上の DFSORT

DFSORT の商品、オンライン資料、ニュース、ヒント、技法、例、などについては、次の URL にある DFSORT ホーム・ページにアクセスしてください。

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

DFSORT FTP サイト

匿名 FTP で DFSORT の商品および例を入手できます。アドレスは次のとおりです。

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

データ・セット、レコード、およびフィールド

ユーザーが DFSORT で取り扱う情報は、データ・セットに入っています。データ・セット という用語は、1 つ以上のレコードが含まれているファイルを指します。レコードの名前付きグループはいずれもデータ・セット と呼ばれます。データ・セット およびファイル という用語は同義であり、本書で使用されている用語は互いに交換可能です。

ソート、コピー、またはマージしたい情報は、データ・セットに入れておきます。DFSORT によって行われるほとんどの処理では、データ・セット全体が影響を受けます。しかし、一部の DFSORT 処理形式では、データ・セット内の特定のレコードだけしか関与しません。

データ・セットをカタログ するようにすると、データ・セットの保管場所を指定しなくても、データ・セットを名前参照できるようになります。カタログ式データ・セットとカタログ式プロシージャ とを混同しないでください。カタログ式プロシージャは、データ・セットに保管された JCL の名前付き集合のことであり、カタログ式データ・セットは、その名前がシステムによって記録されているデータ・セットのことです。

本書を通して、レコード という用語は、1 つの単位として使用される関連情報の集合 (データベース内の 1 つの項目、ある学部 の 1 メンバーに関する個人データ、

など) を指します。フィールド という用語は、特定のカテゴリーのデータに使用される、レコードの特定の部分 (従業員の名前や学部など) を指します。

DFSORT は、固定長または可変長のレコードをソート、コピー、またはマージできます。データ・セットのタイプと長さは、そのレコード・フォーマット (RECFM) と論理レコード長 (LRECL) によって定義されます。固定長データ・セットの RECFM は、F、FB、FBS、というようになります。可変長データ・セットの RECFM は、V、VB、VBS、というようになります。本書では簡単のように、固定長データ・セットおよび固定長レコードを短縮して、それぞれ FB データ・セットおよび FB レコードとして使用し、可変長レコード・データ・セットおよび可変長レコードを短縮して、それぞれ VB データ・セットおよび VB レコードとして使用しています。

RECFM=FB および LRECL=25 のデータ・セットとは、レコード長が 25 バイトの固定長 (FB) データ・セットです (B はブロック化を意味します)。FB データ・セットの場合、LRECL はデータ・セット内の各レコードの長さを知らせます。すなわち、レコードはすべて同じ長さです。FB レコードの最初のデータ・バイトは、位置 1 にあります。LRECL=25 の FB データ・セット内のレコードは、次のようになっています。

```
Positions 1-3: Country Code = 'USA'
Positions 4-5: State Code = 'CA'
Positions 6-25: City = 'San Jose' padded with 12 blanks on the right
```

RECFM=VB および LRECL=25 のデータ・セットとは、最大レコード長が 25 バイトの可変長 (VB) データ・セットです。VB データ・セットの場合、それぞれのレコードは異なる長さを持つことができます。各レコードの最初の 4 バイトには、RDW (レコード記述子ワード) が入り、RDW の最初の 2 バイトには、そのレコードの長さ (2 進表記) が入ります。VB レコードの最初のデータ・バイトは、位置 1-4 の 4 バイトの RDW の後の、位置 5 にあります。LRECL=25 の VB データ・セット内のレコードは、次のようになっています。

```
Positions 1-2: Length in RDW = hex 000E = decimal 14
Positions 3-4: Zeros in RDW = hex 0000 = decimal 0
Positions 5-7: Country Code = 'USA'
Positions 8-9: State Code = 'CA'
Positions 10-17: City = 'San Jose'
```

特に注記のない限り、本書の例では、作業および説明のしやすい FB データ・セットを処理します。ただし、本書の該当する場所では、VB データ・セットの処理に関する特別な考慮事項について説明します。

データ・セットのソート

DFSORT を使用して、データ・セットのレコードを再配置することができます。ソート とは、レコードを昇順または降順でファイルに再配置することです。5 ページの表 2 は、最初は昇順、次に降順にソートされた、名前からなるサンプル・データ・セットを示しています。

表 2. DFSORT は情報を昇順および降順に配置する

未ソート・ データ・セット	昇順 ソート	降順 ソート
Andy	Andy	Edward
Edward	Betty	Dan
Carol	Carol	Carol
Dan	Dan	Betty
Betty	Edward	Andy

データはいろいろなフォーマットでソートすることができます。表 3 は、最も一般的に使用される DFSORT データ・フォーマットとそれらを指定するのに使用するフォーマット ID を示しています。

表 3. 一般的に使用されるデータ・フォーマット

データ・フォーマット	フォーマット ID
EBCDIC (文字)	CH
2 進数 (符号なし数値)	BI
固定小数点 (符号付き数値)	FI
ゾーン 10 進数 (符号付き数値)	ZD
パック 10 進数 (符号付き数値)	PD
浮動符号 (符号付き数値)	FS
フリー・フォーム (符号なし数値)	UFF
フリー・フォーム (符号付き数値)	SFF

使用可能なフォーマットについて詳しくは、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

データ・セットのマージ

DFSORT を使用してデータ・セットをマージすることもできます。DFSORT は、ソート済みレコードの複数のファイルを組み合わせることによってデータ・セットをマージし、ソート済みレコードからなる 1 つのデータ・セットを形成します。

表 4. DFSORT は 2 つのデータ・セットをマージして 1 つのデータ・セットにする

データ・セット 1	データ・セット 2	マージされたデータ・セット
Andy	Amy	Amy
Betty	Chris	Andy
Carol	Sue	Betty
Dan		Carol
Edward		Chris
		Dan
		Edward
		Sue

マージするデータ・セットは、あらかじめ同じ順序 (昇順または降順) にソートしておく必要があります。

データ・セットのコピー

また、DFSORT では、ソートまたはマージを行わなくてもデータ・セットをコピーできます。データ・セットをソートまたはマージするのとはほとんど同じ方法で、データ・セットをコピーします。

DFSORT でほかにできること

データ・セットのソート、マージ、またはコピーを行っている間に、次のような作業を行うこともできます。

- 入力データ・セットからレコードのサブセットを選択する。指定した基準を満たすレコードを組み込んだり省略したりすることができます。例えば、多数のいろいろな学部のコース資料のレコードが含まれている入力データ・セットをソートする場合に、1 つの学部に関する資料だけをソートすることができます。
- さまざまな方法でレコードを再フォーマットする。一度に 1 項目ずつレコードを作成したり、特定の列のみをオーバーレイしたり、レコードごとに異なる方法で再フォーマットしたりすることができます。フィールドの編集、変更、追加、または削除を行えます。固定位置/長さフィールドを直接処理したり、以降の処理のために可変位置/長さフィールド（コンマ区切り値など）を固定解析フィールドに変換したりすることができます。また、ブランク、ゼロ、ストリング、現在日付、将来の日付、過去の日付、現在時刻、シーケンス番号、10 進定数、および算術命令の結果を、入力フィールドの前、間、および後に入れることもできます。例えば、入力データ・セットから文字ストリングと特定の編集フィールドだけを収めた、異なって配置された出力データ・セットを作成することができます。
- ソートまたはマージ中に、選択したレコードの値を合計する（コピー中にはできません）。コース資料のレコードを収めたデータ・セットの例では、DFSORT を使用して 1 つの学部に関する本の金額を合計することができます。
- 入力データ・セットを 1 回受け渡すだけで、複数の出力データ・セットおよび簡潔または複雑な報告書を作成する。例えば、それぞれの学部のレコードごとに異なる出力データ・セットを作成することができます。
- VB データ・セットを FB データ・セットへ、または FB データ・セットを VB データ・セットへ変換する。
- レコードを抽出または繰り返す。
- 選択したロケールで定義されている照合規則に従って、レコードのソート、マージ、組み込み、または省略を行う。
- レコードをソートまたはマージするときに照合シーケンスを変更する（コピー中にはできません）。例えば、大文字の後に小文字を照合させることができます。
- DFSORT でレコードを処理するのに IBM 2 バイト文字セット発注サポート (DBCS 発注) (5665-360 ライセンス・プログラムのリリース 2.0 または等価のプログラム) を使用している場合には、日本語データをソート、マージ、またはコピーする。

DFSORT ジョブの作成および実行

DFSORT によるデータ・セットの処理には、次の 2 つのステップが必要です。

1. DFSORT ジョブの作成
2. DFSORT ジョブの実行

DFSORT ジョブは、次のようないろいろな方法で処理を呼び出すことによって実行できます。

- プログラムの名前またはカタログ式プロシージャーを使用して、JCL EXEC ステートメントによって。
- COBOL、PL/I、または基本アセンブラー言語で書かれたプログラム内から。

本書では、「直接」または「JCL 呼び出し」という言い方は、DFSORT プログラムが、PGM=SORT または PGM=ICEMAN を指定した JCL EXEC ステートメントによって開始されることを意味します。「プログラムによって呼び出される」または「動的呼び出し」という言い方は、DFSORT プログラムが別のプログラムから開始されることを意味します。

ジョブの作成

ご使用のサイトで DFSORT がどのようにインストールされているかにかかわらず、JCL および DFSORT 制御ステートメントを書くことによって DFSORT を使用することができます。第 1 部に、JCL および DFSORT プログラム制御ステートメントを書くのに必要な説明を収めています。

DFSORT 処理を呼び出すには、JCL ステートメントおよび DFSORT プログラム制御ステートメントを作成する必要があります。JCL ステートメントは、オペレーティング・システムによって処理されます。JCL ステートメントによって、オペレーティング・システムに対してユーザーのデータ・セットを記述し、DFSORT 処理を開始します。DFSORT プログラム制御ステートメントは DFSORT によって処理されます。DFSORT プログラム制御ステートメントによって、ユーザーが行いたい処理を記述し開始します。

DFSORT 制御ステートメントの要約

最も重要な DFSORT 制御ステートメントの機能を、以下に要約しています。

SORT	ソート・アプリケーションのフィールドを記述するか、またはコピー・アプリケーションを要求します。
MERGE	マージ・アプリケーションのフィールドを記述するか、またはコピー・アプリケーションを要求します。
OPTION	インストール・デフォルトを指定変更するか、またはオプション機能もしくはコピー・アプリケーションを要求します。
INCLUDE	ソート、コピー、またはマージの前にレコードを組み込むのに使用する基準を記述します。
OMIT	ソート、コピー、またはマージの前にレコードを省略するのに使用する基準を記述します。
INREC	ソート、コピー、またはマージの前にレコードをどのように再フォーマットするかを記述します。

OUTREC	ソート、コピー、またはマージの後にレコードをどのように再フォーマットするかを記述します。
SUM	ソートまたはマージの後にフィールドをどのように合計するかを記述します。
OUTFIL	ソート、コピー、またはマージの後に 1 つ以上の出力データ・セットに対して実行するさまざまなタイプの処理を記述します。

それほど重要でない DFSORT 制御ステートメントの機能を、以下に要約しています。

ALTSEQ	通常の変換テーブルに対する変更を記述します。
MODS	ユーザー出口ルーチンを記述します。
RECORD	必要になったときにデータ・セットのレコード・タイプとレコード長情報を提供します。
DEBUG	診断機能を要求します。
END	制御ステートメントの終わりを示します。

ジョブの実行

DFSORT ジョブは、PGM=SORT または PGM=ICEMAN を使用する JCL EXEC ステートメントによって直接実行することができます。あるいは、COBOL、アセンブラー、PL/I、またはその他のタイプのプログラムから動的に DFSORT を呼び出すこともできます。

サンプル・データ・セットの作成および使用

本書の多くの例では、サンプル・データ・セット SORT.SAMPIN、SORT.SAMPADD、SORT.BRANCH、および SORT.SAMPOUT を参照しています。これらのデータ・セットを使用している本書の例を試したい場合は、付録 A『サンプル・データ・セットの作成』を参照してください。そこに、DFSORT に付属の ICESAMP と呼ばれるプログラムを使用して、これらのデータ・セットのユーザー独自のコピーを作成する方法が示されています。

注: 一部の例では、SORT.SAMPIN、SORT.SAMPOUT、SORT.SAMPADD、および SORT.BRANCH 以外のデータ・セットが使用されています。テキストで使用されているデータ・セットと一致させるには最初からデータ・セットを作成することもできますし、すでにお持ちのデータ・セットに同様の処理を行うこともできます。

始める前に、189 ページの『付録 B. サンプル・データ・セットの記述』をお読みください。本書の多くの例では、入力データ・セットとしてサンプル・ブックストア・データ・セットを参照しているので、これらのデータ・セットの内容を知っておいてください。入力データ・セットは、ユーザーが配置またはソートしたいデータを含んでいます。実行するどの DFSORT ジョブでも入力データ・セットを指定する必要があります。サンプル・ブックストア・データ・セットは **SORT.SAMPIN** という名前で、追加のブックストア・データ・セットは **SORT.SAMPADD** という名前です。

ブックストア・データ・セット内の各レコードは、12 個のフィールドを持っています (Book Title (ブック・タイトル)、Author's Last Name (著者のラストネーム)、など)。1 つのレコードは、ページの横の 1 行によって表すことができます。1 のフィールドは、ページの縦の 1 列によって表すことができます。

データ・セットをソートするには、レコードを順序付ける (昇順または降順に配置する) のに使用する 1 つ以上のフィールドを選択します。これらのフィールドは、制御フィールド (または COBOL ではキー) と呼ばれます。

以下のページの練習問題を進めていくときは、制御フィールドだけではなく、それぞれのレコード全体がソートされることを覚えておいてください。ただし、簡単にするために、テキストの図では、論議している制御フィールドだけしか示していません。ソート済みレコードは実際にはすべてのフィールドを含んでいますが、1 ページでは狭すぎて全部を表示できないからです。『付録 B. サンプル・データ・セットの記述』に、各レコードのすべてのフィールドを示してあります。また、各フィールドのバイト位置を示す見出しと数値によっても配置されます。数値フィールドは 2 進数形式 (5 ページの表 3 を参照) であり、したがって、ほとんどの表示では本書で示されているようには表示されません。データを配置および表示するために使用できる方法については、後続の DFSORT 機能に関する章で説明します。

10 ページの表 5 は、ソート済みフィールドの例を示しています。ソート済みフィールドの上に数値の行があります。これらの数値は、それらのフィールドのバイト位置を表しています。ユーザーはバイト位置を使用して、DFSORT に対してフィールドを識別することになります。例では、DFSORT の使用法を学習するのに役立つようにバイト位置を示しています。バイト位置は、処理されたデータ・セットの中では実際には現れることはありません。

10 ページの表 5 では、学部コース (Course Department) フィールドに何も表示されていない最初の 2 つのレコードは、特定のコースには必要でない一般図書です。この例の場合、制御フィールドは Course Department フィールドです。

DFSORT とは

表 5. 学部コース (Course Department) の昇順でソートされたサンプル・ブックストア・データ・セット

Book Title		Course Department		Price
1	75	110	114	170 173
LIVING WELL ON A SMALL BUDGET				9900
PICK'S POCKET DICTIONARY				295
INTRODUCTION TO BIOLOGY		BIOL		2350
SUPPLYING THE DEMAND		BUSIN		1925
STRATEGIC MARKETING		BUSIN		2350
COMPUTER LANGUAGES		COMP		2600
VIDEO GAME DESIGN		COMP		2199
COMPUTERS: AN INTRODUCTION		COMP		1899
NUMBERING SYSTEMS		COMP		360
SYSTEM PROGRAMMING		COMP		3195
INKLINGS: AN ANTHOLOGY OF YOUNG POETS		ENGL		595
EDITING SOFTWARE MANUALS		ENGL		1450
MODERN ANTHOLOGY OF WOMEN POETS		ENGL		450
THE COMPLETE PROOFREADER		ENGL		625
SHORT STORIES AND TALL TALES		ENGL		1520
THE INDUSTRIAL REVOLUTION		HIST		795
EIGHTEENTH CENTURY EUROPE		HIST		1790
CRISIS OF THE MIDDLE AGES		HIST		1200
INTRODUCTION TO PSYCHOLOGY		PSYCH		2200
ADVANCED TOPICS IN PSYCHOANALYSIS		PSYCH		2600

表 5 のレコードのうち、等価の照合制御フィールド (この場合は、同じ学部) を持つレコードは、元の順序で示されていることに注意してください。例えば、COMP (Computer Science - 情報科学) 学部の中では、ブック・タイトル *Video Game Design* は元の順序通りに *Computers: An Introduction* より前に表示されています。

等価の照合制御フィールドを持つレコードを元の順序で表示するかどうか、または DFSORT がそれらをランダムに順序付けするかどうかを制御できます。システム・プログラマーはインストール時にデフォルトを設定しますが、ユーザーは特定の DFSORT オプションを使用して実行時に変更することができます。本書の例では、デフォルトは、等価の照合制御フィールドを持つレコードを元の順序で表示することです。

要約

本書ではこれまでに、以下の概念を説明しました。

- DFSORT を使用して、データ・セットをソート、コピー、またはマージできます。
- JCL および DFSORT プログラム制御ステートメントを書くことによって、DFSORT ジョブを作成し処理することができます。
- DFSORT ジョブを直接実行することもできますし、DFSORT をプログラムから呼び出すこともできます。

さらに、本章では、DFSORT で提供されるサンプル・データ・セットの使い方と読み取り方を説明しました。次に、DFSORT 制御ステートメントの作成方法を示すチュートリアルに進みます。

第 2 部 JCL および DFSORT 制御ステートメントの書き方を学ぶ

第 2 章 データ・セットのソート、マージ、および

コピー	15
データ・セットのソート	15
複数のフィールドによるソート	17
ステートメントの継続	19
コメント・ステートメント	19
データ・セットを直接ソートするための JCL	19
データ・セットのマージ	21
MERGE 制御ステートメントの作成	22
データ・セットを直接マージするための JCL	23
VB データ・セットについての考慮事項	24
開始位置	25
短制御フィールド	25
データ・セットのコピー	26
SORT、MERGE、または OPTION ステートメントへの COPY の指定	26
データ・セットを直接コピーするための JCL	26

第 3 章 レコードの組み込みまたは省略

INCLUDE ステートメントの作成	30
OMIT ステートメントの作成	32
INCLUDE および OMIT で許容される比較	33
定数の作成	35
文字ストリング	35
16 進数ストリング	35
10 進数	35
INCLUDE および OMIT の数値テスト	36
INCLUDE および OMIT の場合のサブストリング検索	37
VB データ・セットについての考慮事項	37
開始位置	37
短制御フィールド	38

第 4 章 レコードの合計

SUM ステートメントの作成	39
重複制御フィールドを持つレコードを抑止する	41
オーバーフローの処理	42
VB データ・セットについての考慮事項	42
開始位置	42
短要約フィールド	43

第 5 章 固定フィールドを持つレコードの再フォーマット

BUILD または FIELDS を使用したソート後のレコードの再フォーマット	45
フィールドを予約スペースに再配列する	48
2 進ゼロの挿入	48
ブランクの挿入	49
ストリングの挿入	51
文字ストリング	51

16 進数ストリング	52
基本報告書のセットアップ	52
大文字から小文字への変更	53
数値フィールドから異なるフォーマットへの変換	54
数値フィールドの編集	55
データを 16 進数で表示する	58
数値フィールドおよび定数に算術演算を実行する	59
ルックアップおよび変更	60
データの左寄せと右寄せ	61
データの左/右詰め	64
OVERLAY を使用したレコードの再フォーマット	68
OVERLAY を使用したレコードの拡張	70
IFTHEN を使用したレコードの再フォーマット	71
ソート前にレコードを再フォーマットする	74
INREC と一緒に他のステートメントを使用する	74
値を合計するときのオーバーフローを防止する	75
シーケンス番号の挿入	76
VB データ・セットについての考慮事項	79
RDW	80
開始位置および桁	80
可変データ	81

第 6 章 可変フィールドを持つレコードの再フォーマット

BUILD および OVERLAY での %nn 解析済みフィールドの使用	83
IFTHEN での %nn 解析済みフィールドの使用	85
BUILD および OVERLAY で %nn フィールドを使用できる項目	87
PARSE パラメーター	88

第 7 章 複数の出力データ・セットおよび報告書の作成

複数の同一コピーの作成	89
相対レコード番号による選択および抽出	91
廃棄レコードの組み込み、省略、および保管	93
再フォーマット	95
繰り返し	97
分割	98
報告書の作成: OUTFIL 対 ICETOOL	100
OUTFIL を使用した報告書の作成	101
データ	101
ヘッダー	102
トレーラーおよび統計	105
データまたは紙送り制御文字がない場合	107
ヘッダーとトレーラーに関する VB データ・セットの考慮事項	107
セクション	108
FB から VB への変換	111

VB から FB への変換	113
第 8 章 プログラムからの DFSORT の呼び出し	115
制御ステートメントの受け渡し	115
COBOL プログラムからの DFSORT の呼び出し	116
レコードのソート	116
レコードのマージ	119
COBOL FASTSRT によるソート	121
PL/I プログラムからの DFSORT の呼び出し	121
第 9 章 インストール・デフォルトの指定変更	123
JCL EXEC ステートメントへの PARM パラメータ	
の指定	124
DFSPARM への OPTION 制御ステートメントの指	
定	124
第 10 章 DFSORT の効率的使用	127
主記憶域を十分に提供する	127
メモリー・オブジェクト・ソート、ハイパー・ソー	
ト、およびデータ・スペース・ソートを許可する	128
作業データ・セットに高速ディスクを使用する	128
INREC を使用して不要なフィールドを除去する	128
INCLUDE または OMIT を使用して不要なレコー	
ドを除去する	129
STOPAFT および SKIPREC を使用して不要なレコ	
ードを除去する	129
SUM を使用してレコードを統合する	129
OUTFIL を使用して複数の出力データ・セットを作	
成する	129
DFSORT 制御ステートメントを使用してプログラ	
ム・ロジックを置き換える	130
COBOL の場合には FASTSRT を使用する	130
パフォーマンスを低下させる可能性のあるオプショ	
ンを避ける	130

第 2 章 データ・セットのソート、マージ、およびコピー

このチュートリアルでは、JCL を使用して処理される DFSORT プログラム制御ステートメントを作成することによって、データ・セットをどのようにソート、マージ、およびコピーするかを紹介します。

DFSORT プログラム制御ステートメントは JCL の入力であり、DFSORT を実行するために使用されます。指示を単純にするため、最初にプログラム制御ステートメントを説明し、その後に関連 JCL ステートメントを説明しています。大部分のチュートリアルでは、JCL で呼び出す DFSORT、すなわち JCL を使用した DFSORT の実行を中心に説明します。プログラムから DFSORT を呼び出す方法 (動的呼び出し) については、115 ページの『第 8 章 プログラムからの DFSORT の呼び出し』で説明します。

データ・セットのソート

DFSORT を直接使用するには (JCL 呼び出し)、制御フィールドと各フィールドのソート順序を記述する SORT 制御ステートメントを作成します。作成した制御ステートメントは、JCL 内の SYSIN データ・セットの一部となります。SYSIN データ・セットは、通常は、//SYSIN DD * の後に「インライン」制御ステートメントが続いたものとして指定されます (本書の例を参照)。ただし、レコードとして制御ステートメントを持つ、順次データ・セットまたは区分データ・セットのメンバーを、SYSIN データ・セットとして使用することもできます。

SORT は、その他のすべての DFSORT 制御ステートメントと一緒に使用できます。

学部コース (Course Department) フィールド (16 ページの表 7 に示されている) によってブックストア・レコードをソートする SORT ステートメントは、次のようになります。

```
1 2                                71    80
      SORT FIELDS=(110, 5, CH, A)
```

- 110 → Department (学部) フィールドの開始
- 5 → Department (学部) フィールドの長さ
- CH → 文字データ
- A → 昇順

ステートメントは、桁 2 から 71 の間にコーディングしてください。

以下に、この SORT ステートメントを作成するためのステップを示します。

表 6. 学部 (Department) によってソートする SORT ステートメントの作成ステップ

ステップ	アクション
1	少なくとも 1 つのブランクを入れてから、 SORT を入力する

表 6. 学部 (Department) によってソートする *SORT* ステートメントの作成ステップ (続き)

ステップ	アクション
2	少なくとも 1 つのブランクを入れてから、 FIELDS= を入力する
3	以下のものをコンマで区切り、括弧で囲んで入力する <ol style="list-style-type: none"> ブックストア・データ・セット内のレコードの先頭 (先頭位置はバイト 1) からの相対位置で表した、学部コース (Course Department) フィールドが始まる場所。学部コース・フィールドは、バイト 110 から始まります。 学部フィールドの長さ (バイト数)。学部フィールドは 5 バイトの長さです。 データ・フォーマットのフォーマット ID。学部フィールドには文字データが含まれており、文字データは CH として指定されます。(5 ページの表 3 に、一般に使用されるデータ・フォーマットのコードを示しています。) 昇順を表す文字 A。

表 7 は一部のフィールドしか示していませんが、表示されるフィールドは出力データ・セット内にあるものとは限らないことに注意してください。出力データ・セットは、より厳密には、サンプル・ブックストア・データ・セットの折り込みに似ています。

表 7. 学部コース (Course Department) の昇順でソートされたサンプル・ブックストア・データ・セット

Book Title	Course Department
1 75	110 114
LIVING WELL ON A SMALL BUDGET	
PICK'S POCKET DICTIONARY	
INTRODUCTION TO BIOLOGY	BIOL
SUPPLYING THE DEMAND	BUSIN
STRATEGIC MARKETING	BUSIN
COMPUTER LANGUAGES	COMP
VIDEO GAME DESIGN	COMP
COMPUTERS: AN INTRODUCTION	COMP
NUMBERING SYSTEMS	COMP
SYSTEM PROGRAMMING	COMP
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	ENGL
EDITING SOFTWARE MANUALS	ENGL
MODERN ANTHOLOGY OF WOMEN POETS	ENGL
THE COMPLETE PROOFREADER	ENGL
SHORT STORIES AND TALL TALES	ENGL
THE INDUSTRIAL REVOLUTION	HIST
EIGHTEENTH CENTURY EUROPE	HIST
CRISES OF THE MIDDLE AGES	HIST
INTRODUCTION TO PSYCHOLOGY	PSYCH
ADVANCED TOPICS IN PSYCHOANALYSIS	PSYCH

レコードを降順でソートするには、A の代わりに D を指定します。例えば、各本を価格の降順でソートするには、次のように入力します。

SORT FIELDS=(170, 4, BI, D)

ソート順序は、バイト 170 から 173 まです。降順の 2 進数データとして指定されます。表 8 は、降順のソートの結果を示しています。

表 8. 価格の降順でソートされたサンプル・ブックストア・データ・セット

Book Title		Price	
1	75	170	173
LIVING WELL ON A SMALL BUDGET		9900	
SYSTEM PROGRAMMING		3195	
COMPUTER LANGUAGES		2600	
ADVANCED TOPICS IN PSYCHOANALYSIS		2600	
STRATEGIC MARKETING		2350	
INTRODUCTION TO BIOLOGY		2350	
INTRODUCTION TO PSYCHOLOGY		2200	
VIDEO GAME DESIGN		2199	
SUPPLYING THE DEMAND		1925	
COMPUTERS: AN INTRODUCTION		1899	
EIGHTEENTH CENTURY EUROPE		1790	
SHORT STORIES AND TALL TALES		1520	
EDITING SOFTWARE MANUALS		1450	
CRISES OF THE MIDDLE AGES		1200	
THE INDUSTRIAL REVOLUTION		795	
THE COMPLETE PROOFREADER		625	
INKLINGS: AN ANTHOLOGY OF YOUNG POETS		595	
MODERN ANTHOLOGY OF WOMEN POETS		450	
NUMBERING SYSTEMS		360	
PICK'S POCKET DICTIONARY		295	

複数のフィールドによるソート

複数の制御フィールドを指定することにより、ブックストア・データ・セット内のレコードをさらにソートすることができます。複数の制御フィールドを指定するときは、優先順位の高いものから順にフィールドを指定します。制御フィールドは、他の制御フィールドとオーバーラップしたり、他の制御フィールド内に含まれることがあることに注意してください。

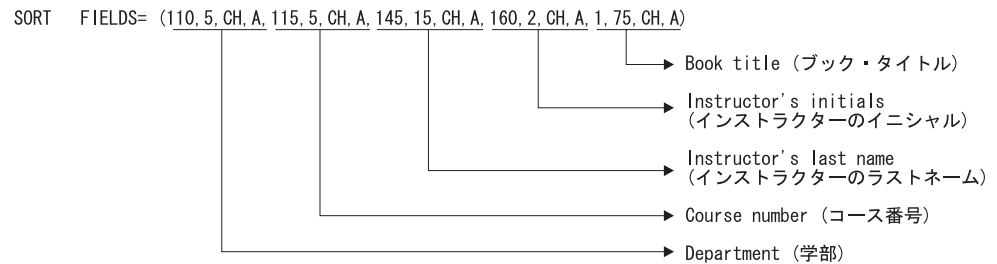
18 ページの表 9 は、以下の制御フィールドを以下の順序で指定した場合にレコードがどのようにソートされるかを示しています。

1. 学部コース (Course Department)
2. コース番号 (Course Number)
3. インストラクターのラストネーム (Instructor's Last Name)
4. インストラクターのイニシャル (Instructor's initials)

5. ブック・タイトル (Book Title)

したがって、2 つのレコードが同じ学部である場合、それらのレコードはコース番号でソートされます。コース番号も同じであると、それらのレコードはインストラクターのラストネームでソートされます。ラストネームも同じであると、それらはイニシャルでソートされます。最後は、イニシャルも同じであると、それらはタイトルでソートされます。

それぞれの制御フィールドの位置、長さ、データ・フォーマット、および順序を、次のように指定します。



レコードは、表 9 に示されたようにソートされます。

表 9. 複数のフィールドでソートされたサンプル・ブックストア・データ・セット

Book	Title	Course Department		Course Number		Instructor's Last Name		Instructor's Initials	
1	75	110	114	115	119	145	159	160	161
LIVING WELL ON A SMALL BUDGET									
PICK'S POCKET DICTIONARY									
INTRODUCTION TO BIOLOGY		BIOL		80521		GREENBERG		HC	
STRATEGIC MARKETING		BUSIN		70124		LORCH		HH	
SUPPLYING THE DEMAND		BUSIN		70251		MAXWELL		RF	
NUMBERING SYSTEMS		COMP		00032		CHATTERJEE		AN	
COMPUTER LANGUAGES		COMP		00032		CHATTERJEE		CL	
COMPUTERS: AN INTRODUCTION		COMP		00032		CHATTERJEE		CL	
SYSTEM PROGRAMMING		COMP		00103		SMITH		DC	
VIDEO GAME DESIGN		COMP		00205		NEUMANN		LB	
SHORT STORIES AND TALL TALES		ENGL		10054		BUCK		GR	
EDITING SOFTWARE MANUALS		ENGL		10347		MADRID		MM	
THE COMPLETE PROOFREADER		ENGL		10347		MADRID		MM	
INKLINGS: AN ANTHOLOGY OF YOUNG POETS		ENGL		10856		FRIEDMAN		KR	
MODERN ANTHOLOGY OF WOMEN POETS		ENGL		10856		FRIEDMAN		KR	
THE INDUSTRIAL REVOLUTION		HIST		50420		GOODGOLD		ST	
CRISES OF THE MIDDLE AGES		HIST		50521		WILLERTON		DW	
EIGHTEENTH CENTURY EUROPE		HIST		50632		BISCARDI		HR	
INTRODUCTION TO PSYCHOLOGY		PSYCH		30016		ZABOSKI		RL	
ADVANCED TOPICS IN PSYCHOANALYSIS		PSYCH		30975		NAKATSU		FL	

制御ステートメントの長さは、しばしば短縮できます。レコード内で互いに隣にあり、データ・フォーマットおよび順序が同じであるフィールドは、一緒に指定でき

ます。学部とコース番号を 1 つのフィールドとして一緒に指定し、インストラクターのラストネームとイニシャルを 1 つのフィールドとして一緒に指定することによって、この最後のステートメントを短縮することができます。

```
SORT  FIELDS=(110,10,CH,A,145,17,CH,A,1,75,CH,A)
```

→ Title (タイトル)
→ Instructor's last name および initials
(インストラクターのラストネームおよびイニシャル)
→ Department および Course number (学部およびコース番号)

さらに、すべての制御フィールドが同じデータ・フォーマットである場合は、**FORMAT=f** パラメーターを使用することにより、データ・フォーマットを 1 回だけ指定すれば済みます。以下に例を示します。

```
SORT  FORMAT=CH,FIELDS=(110,10,A,145,17,A,1,75,A)
```

一部の制御フィールドは同じデータ・フォーマットであるが、その他は同じでない場合は、個々のデータ・フォーマットと一緒に **FORMAT=f** パラメーターを指定できます。以下に例を示します。

```
SORT  FORMAT=CH,FIELDS=(110,10,A,170,4,BI,D,145,17,A,1,75,A)
```

上記の指定は、以下の指定と等価です。

```
SORT  FIELDS=(110,10,CH,A,170,4,BI,D,145,17,CH,A,1,75,CH,A)
```

ステートメントの継続

SORT ステートメント (またはその他の **DFSORT** 制御ステートメント) を桁 2 から 71 の間に収めることができない場合は、次の行に継続することができます。コンマの後にブランクを付けて行を終了すると、**DFSORT** は次の行を継続行として扱います。継続行は、桁 2 から 71 の間のどこで始まっても構いません。

以下に例を示します。

```
SORT  FORMAT=CH,FIELDS=(110,10,A,145,17,A,  
1,75,A)
```

コメント・ステートメント

桁 1 のアスタリスク (*) で開始することにより、コメント・ステートメントと制御ステートメントを混在させることができます。コメント・ステートメントは、**DFSORT** では印刷されますが、それ以外では無視されます。

以下に例を示します。

```
* Sort by department and course number  
SORT  FIELDS=(110,10,CH,A)
```

データ・セットを直接ソートするための JCL

ソートを行うのに必要となるジョブ制御言語 (JCL) は、**DFSORT** を直接実行するのか、**DFSORT** をプログラムから呼び出すのかによって異なります。ここでは、

DFSORT を直接実行する場合を中心に説明します。プログラムから DFSORT を呼び出す方法は、115 ページの『第 8 章 プログラムからの DFSORT の呼び出し』で説明します。

オペレーティング・システムは、DFSORT プログラム制御ステートメントと一緒に提供された JCL を使用して、以下のことを行います。

- ユーザーを許可ユーザーとして識別する
- ジョブを実行するのに必要なリソースを割り振る
- ジョブを実行する
- 結果に関する情報をユーザーに戻す
- ジョブを終了する

実行依頼するすべての DFSORT ジョブに JCL を提供する必要があります。

必須 JCL には、JOB ステートメント、EXEC ステートメント、およびいくつかの DD ステートメントが含まれています。必要なステートメントとそれらの正確な形式は、以下によって異なります。

- DFSORT を、入力ジョブ・ストリーム内の EXEC ステートメントによって呼び出すのか、別のプログラム内のシステム・マクロ命令によって呼び出すのか。
- DFSORT を呼び出すのに EXEC ステートメント・カタログ式プロシーチャーを使用するかどうか。
- EXEC ステートメントで PARM オプションを指定するかどうか。
- DFSPARM データ・セットに PARM オプションまたは制御ステートメントを指定するかどうか。
- SYSIN データ・セットに制御ステートメントを指定するかどうか。
- ユーザー独自のルーチンを活動化するためにプログラム出口を使用するかどうか。

上記の各オプションを選択する際の情報については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」に説明があります。

ほとんどのジョブで必要となる JCL ステートメントを、以下に説明しています。

//jobname JOB ジョブの開始を合図します。ご使用のサイトで、ユーザー名やアカウント番号などの情報を JOB ステートメントに指定することが必要になる場合があります。

//stepname EXEC ジョブ・ステップの開始を合図し、どんなプログラムを実行するかをオペレーティング・システムに知らせます。DFSORT を実行するには、次のような EXEC ステートメントを作成します。

//stepname EXEC PGM=SORT

//STEPLIB DD DFSORT プログラムは、システムに認識されているライブラリーに含まれていることが普通なので、**//STEPLIB DD** ステートメントは必要ありません。ただし、DFSORT がシステムに認識されているライ

ブラリー内にない場合は、//STEPLIB DD ステートメントで、DFSORT プログラムが入っているライブラリーを定義します。

//SYSOUT DD

DFSORT メッセージおよび制御ステートメントがリストされたデータ・セットを定義します。

//SORTIN DD

入力データ・セットまたは連結入力データ・セットを定義します。

//SORTWKdd DD

ソート用の作業データ・セットを定義します。
DFSORT はソート用の作業データ・セットを動的に割り振ることができるので、通常は、このステートメントは必要ありません。

//SORTOUT DD

出力データ・セットを定義します。

//SYSIN DD

DFSORT プログラム制御ステートメントの前に置かれるか、DFSORT プログラム制御ステートメントを含んでいます。

以下に、DFSORT を実行する JCL の代表的な例を示しています。

```
//EXAMP JOB A492,PROGRAMMER
//SORT EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//SORTOUT DD DSN=A123456.SORT.SAMPOUT,DISP=OLD
//SYSIN DD *
        SORT FORMAT=CH,
        FIELDS=(110,10,A,145,17,A,1,75,A)
/*
```

DFSORT の直接実行に関する追加情報については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

これまで

本章ではこれまで、SORT プログラム制御ステートメントを作成する方法および JCL ステートメントを使用してそのソートを実行する方法を紹介しました。次のチュートリアルでは、MERGE プログラム制御ステートメントを使用して 2 つのデータ・セットをマージする方法を説明します。

データ・セットのマージ

一般には、データ・セットをマージする理由は、すでにソート済みのデータ・セットにさらにレコードを追加するためです。

例えば、ブックストア・データ・セットが (22 ページの表 10 に示されたように) 学部コース (Course Department) およびブック・タイトル (Book Title) ですすでにソートされており、これを、やはり学部コースとブック・タイトルですすでにソート済みの 5 つの新規レコードを含むデータ・セットとマージすることによって更新したいとします。

表 10. 学部コースおよびブック・タイトルでソートされたサンプル・ブックストア・データ・セット

Book Title	Course	Department
1	75	110 114
LIVING WELL ON A SMALL BUDGET		
PICK'S POCKET DICTIONARY		
INTRODUCTION TO BIOLOGY	BIOL	
STRATEGIC MARKETING	BUSIN	
SUPPLYING THE DEMAND	BUSIN	
COMPUTER LANGUAGES	COMP	
COMPUTERS: AN INTRODUCTION	COMP	
NUMBERING SYSTEMS	COMP	
SYSTEM PROGRAMMING	COMP	
VIDEO GAME DESIGN	COMP	
EDITING SOFTWARE MANUALS	ENGL	
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	ENGL	
MODERN ANTHOLOGY OF WOMEN POETS	ENGL	
SHORT STORIES AND TALL TALES	ENGL	
THE COMPLETE PROOFREADER	ENGL	
CRISES OF THE MIDDLE AGES	HIST	
EIGHTEENTH CENTURY EUROPE	HIST	
THE INDUSTRIAL REVOLUTION	HIST	
ADVANCED TOPICS IN PSYCHOANALYSIS	PSYCH	
INTRODUCTION TO PSYCHOLOGY	PSYCH	

この例の場合は、表 11 に示されたような新規データ・セットを使用してください。

表 11. 学部コースおよびブック・タイトルでソートされた 5 つの新規レコード

Book Title	Course	Department
1	75	110 114
INTERNATIONAL COOKBOOK		
WORLD JOURNEYS BY TRAIN		
ARTS AND CRAFTS OF ASIA	ART	
BIOCHEMISTRY	BIOL	
BEHAVIORAL ANALYSIS	PSYCH	

データ・セットをマージするには、MERGE 制御ステートメントといくつかの JCL ステートメントを作成します。データ・セットをマージするときは常に、それぞれのレコードが同じフォーマットであること、およびそれらが同じ制御フィールドによってあらかじめソート済みであることを確認してください。一度に最大 100 個のデータ・セットをマージできます。

MERGE は、その他のすべての DFSORT 制御ステートメントと一緒に使用できます。

MERGE 制御ステートメントの作成

MERGE ステートメントのフォーマットは、SORT ステートメントのフォーマットと同じです。ブックストア・マスター・データ・セットと 5 つの新規レコードを含むデータ・セットとをマージするには、次のステートメントを作成します。

MERGE FORMAT=CH, FIELDS=(110, 5, A, 1, 75, A)

▶ Title (タイトル)
▶ Department (学部)

表 12 は、マージされた出力を示しています。

表 12. 5 つの新規レコードでマージされたサンプル・ブックストア・データ・セット

Book Title	Course Department
1	75
110	114
INTERNATIONAL COOKBOOK	
LIVING WELL ON A SMALL BUDGET	
PICK'S POCKET DICTIONARY	
WORLD JOURNEYS BY TRAIN	
ARTS AND CRAFTS OF ASIA	ART
BIOCHEMISTRY	BIOL
INTRODUCTION TO BIOLOGY	BIOL
STRATEGIC MARKETING	BUSIN
SUPPLYING THE DEMAND	BUSIN
COMPUTER LANGUAGES	COMP
COMPUTERS: AN INTRODUCTION	COMP
NUMBERING SYSTEMS	COMP
SYSTEM PROGRAMMING	COMP
VIDEO GAME DESIGN	COMP
EDITING SOFTWARE MANUALS	ENGL
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	ENGL
MODERN ANTHOLOGY OF WOMEN POETS	ENGL
SHORT STORIES AND TALL TALES	ENGL
THE COMPLETE PROOFREADER	ENGL
CRISES OF THE MIDDLE AGES	HIST
EIGHTEENTH CENTURY EUROPE	HIST
THE INDUSTRIAL REVOLUTION	HIST
ADVANCED TOPICS IN PSYCHOANALYSIS	PSYCH
BEHAVIORAL ANALYSIS	PSYCH
INTRODUCTION TO PSYCHOLOGY	PSYCH

データ・セットを直接マージするための JCL

ソートの場合と同様、必要となる JCL は、DFSORT を直接実行するか、プログラムから呼び出すかによって異なります。本章では、DFSORT の直接実行についてのみ説明します。

マージに必要な JCL は、ソートの場合と同じですが、ただし以下の例外があります。

- 作業データ・セットの動的割り振りまたは SORTWKdd DD ステートメントは必要ありません。

- 入力データ・セットを定義するには、SORTIN DD ステートメントではなく SORTINnn DD ステートメントを使用します。 SORTINnn DD ステートメントによって、入力データ・セットを指名し、マージするデータ・セットの数を知らせます。マージするそれぞれのデータ・セットごとに 1 つの SORTINnn DD ステートメントが必要になります。 SORTINnn の nn は、00 から 99 の数値です。したがって、5 つのデータ・セットをマージしたい場合、通常は、SORTIN01、SORTIN02、SORTIN03、SORTIN04、および SORTIN05 の DD ステートメントを使用します。

事前ソート・ブックストア・データ・セットと新規レコードを含むデータ・セットとをマージするには、この例では以下の JCL ステートメントをコーディングします。新規データ・セットは A123456.NEW という名前であり、ブックストア・データ・セットのソート済みバージョンは A123456.MASTER という名前です。この例の場合、入力データ・セットはカタログされており、出力データ・セットもカタログされると想定されています。

```
//EXAMP      JOB   A492,PROGRAMMER
//MERGE      EXEC  PGM=SORT
//SYSOUT    DD   SYSOUT=A
//SORTIN01 DD   DSN=A123456.MASTER,DISP=SHR
//SORTIN02 DD   DSN=A123456.NEW,DISP=SHR
//SORTOUT   DD   DSN=A123456.SORT.SAMPOUT,DISP=OLD
//SYSIN     DD   *
MERGE      FIELDS=(110,5,CH,A,1,75,CH,A)
/*
```

115 ページの『第 8 章 プログラムからの DFSORT の呼び出し』では、プログラムから DFSORT を呼び出すときのデータ・セットのマージ方法について学習します。

これまで

本章ではこれまで、SORT および MERGE プログラム制御ステートメントの作成方法、および JCL ステートメントを使用してこれらの制御ステートメントを処理する方法を紹介しました。次は、COPY に関するチュートリアルです。

VB データ・セットについての考慮事項

VB データ・セットのレコードは、次のようになっています。

RDW	固定データ	可変データ
-----	-------	-------

RDW (レコード記述子ワード) は 4 バイトの 2 進数フィールドであり、最初の 2 バイトにはレコードの長さが入ります。固定データは、すべてのレコードに存在するデータ・バイトから構成されます。可変データは、レコードに存在する場合と存在しない場合がある 1 つ以上のデータ・バイトから構成され、したがって、それぞれのレコードはデータ・セットの最大論理レコード長 (LRECL) までの異なる長さを持つことができます。

開始位置

FB データ・セットの場合、最初のデータ・バイトは位置 1 から始まります。しかし、VB データ・セットの場合は、位置 1-4 は RDW であるので、最初のデータ・バイトは位置 5 から始まります。したがって、VB データ・セットをソートまたはマージするための制御フィールドをコーディングするときは、4 バイトの RDW のための 4 を開始位置に追加するのを忘れないでください。例えば、次の SORT ステートメントは、VB レコードの第 3 から第 5 データ・バイトに CH 制御フィールドを指定します。

```
SORT FIELDS=(7,3,CH,A)
```

短制御フィールド

VB レコードは固定部分と可変部分を持つので、制御フィールドの一部が欠落することがあります。次の SORT ステートメントを例にとります。

```
SORT FIELDS=(21,12,CH,A)
```

制御フィールドは位置 21-32 にあります。VB レコードが 25 固定データ・バイトで LRECL=45 の場合、レコードの長さは、29 バイト (4 バイトの RDW、25 バイトの固定データ、および 0 バイトの可変データ) から 45 バイト (4 バイトの RDW、25 バイトの固定データ、および 16 バイトの可変データ) まで変化します。32 バイトまたはそれより長いレコードは、制御フィールド全体を含んでいます。しかし、32 バイトより短いレコードでは「短」制御フィールドとなります。すなわち、制御フィールドの終わりの数バイトが欠落することになります。欠落バイトは値を持たないので、欠落バイトを使用して制御フィールドを有効にソートまたはマージすることはできません。

VB レコードが短制御フィールドを持つと分かっている場合は、必要であれば VLSHRT オプションを指定することにより、DFSORT が終了するのを阻止することができます。以下に例を示します。

```
OPTION VLSHRT  
SORT FIELDS=(21,12,CH,A)
```

VLSHRT は、欠落している制御フィールド・バイトを一時的に 2 進ゼロで置き換えるよう DFSORT に指示するもの (出力レコードではゼロは保持されません) で、これにより DFSORT は短制御フィールドを有効にソートまたはマージできるようになります。

重要: NOVLSHRT が有効である場合、DFSORT は、VB レコード内で短制御フィールドを検出すると終了します。

DFSORT の VLSHRT オプションの詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

データ・セットのコピー

DFSORT では、ソートまたはマージを実行せずにデータ・セットを直接コピーすることができます。

COPY は、SUM 以外の他のすべての DFSORT 制御ステートメントと一緒に使用できます。DFSORT は、後続の章で説明されている制御ステートメントを使用することにより、ユーザーがコピーしたい特定のデータ・セットを選択し再フォーマットすることができます。

コピー・ステートメントは、SORT、MERGE、または OPTION ステートメントに COPY を指定することによって作成します。

SORT、MERGE、または OPTION ステートメントへの COPY の指定

SORT および MERGE ステートメントで COPY を指定する場合、変更はほとんどありません。通常は括弧で囲んで指定する情報を、次のように単に COPY というワードで置き換えるだけです。

```
SORT FIELDS=COPY
MERGE FIELDS=COPY
```

次のように、OPTION ステートメントでも COPY を指定できます。

```
OPTION COPY
```

これらの 3 つのステートメントはすべて同一の結果をもたらします。

データ・セットを直接コピーするための JCL

コピー・アプリケーションに必要な JCL は、ソートの場合と同じです。ただし、作業データ・セットの動的割り振りまたは SORTWKdd DD ステートメントは必要ありません。

このサンプル JCL は、OPTION COPY ステートメントを使用して、SORT.SAMPIN データ・セットを一時出力データ・セットにコピーします。

```
//EXAMP JOB A492,PROGRAMMER
//COPY EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//SORTOUT DD DSN=*&TEMP,DISP=(,PASS),SPACE=(CYL,(5,5)),UNIT=SYSDA
//SYSIN DD *
        OPTION COPY
/*
```

OPTION COPY の代わりに SORT FIELDS=COPY または MERGE FIELDS=COPY を使用しても、同じ結果が得られます。

要約

本章では、以下の概念を扱いました。

- **SORT**、**COPY**、または **MERGE** プログラム制御ステートメントの作成
- ソート、コピー、またはマージを処理するための **JCL** ステートメントの使用
方法

後続のチュートリアルでは、出力データ・セットの用途に合わせて入力データ・セットのサブセットを入手する 2 つの方法について学習します。 29 ページの『第 3 章 レコードの組み込みまたは省略』では、**INCLUDE** および **OMIT** で許容される比較演算子、各種の定数、サブストリング検索、および埋め込みと切り捨ての規則を学習します。

第 3 章 レコードの組み込みまたは省略

データ・セット内のレコードのサブセットだけがアプリケーションに必要となることがよくあります。本章では、ソート、コピー、またはマージして出力データ・セットに入れるために、入力データ・セットから特定のレコードだけを組み込むまたは省略する方法について説明します。

ソート、コピー、またはマージを行う前に INCLUDE または OMIT ステートメントによって不要なレコードを除去することにより、ソート、コピー、またはマージの速度を高めることができます。レコード数が少なければ、それだけ処理時間も少なくなります。

入力データ・セットからレコードのサブセットを選択するには、次のような方法があります。

- INCLUDE 制御ステートメントを使用して、必要なレコードを集める。
- OMIT 制御ステートメントを使用して、不要なレコードを除外する。
- OUTFIL ステートメントの INCLUDE または OMIT パラメーターを使用して、それぞれ、必要なレコードを集めるか、不要なレコードを除外する。異なる OUTFIL ステートメントでは、異なる INCLUDE および OMIT パラメーターを使用できます。

INCLUDE または OMIT のいずれを選択するかは、特定のアプリケーションに合わせてステートメントを書くのにいずれがより容易でかつ効率的であるかによって決まります。両方のステートメントを一緒に使用することはできません。

INCLUDE および OMIT ステートメントに関して本章で提示する情報は、以下の点を除いて、OUTFIL ステートメントの INCLUDE および OMIT パラメーターにも適用されます。

- OUTFIL は、ソート、コピー、またはマージの後に処理されます。
- OUTFIL では FORMAT=f パラメーターは使用できません。

OUTFIL については、89 ページの『第 7 章 複数の出力データ・セットおよび報告書の作成』で説明します。

以下のいずれかによって、組み込むまたは省略するレコードを選択します。

1. フィールドの内容を次のどちらかと比較する。

別のフィールド

例えば、著者のラストネームがインストラクターのラストネームと同じであるレコードを選択できます。

定数

文字ストリング、10 進数、16 進数ストリング、現在日付、将来の日付、または過去の日付を、定数として使用することができます。例えば、学部 (Department) フィールドに 『HIST』 の文字ストリングを持つレコードを選択できます。

レコードの組み込みまたは省略

- フィールドに「数値」があるか、「非数値」があるかをテストする。例えば、従業員 (Employees) フィールドまたは売り上げ (Revenue) フィールドに非数値があるレコードを選択できます。

論理 AND および OR を使用して複数の条件を組み合わせることもできます。例えば、学部フィールドに『HIST』または『PSYCH』のいずれかを持つレコードを選択できます。

INCLUDE と OMIT は共に、強力なサブストリング検索機能を提供します。

さらに、INCLUDE および OMIT では、ビット論理テストおよび 2 桁の年の比較の結果に基づいてレコードを選択することもできます。この 2 つの機能については本書では触れておらず、「z/OS DFSORT アプリケーション・プログラミング・ガイド」に詳細情報があります。

INCLUDE ステートメントの作成

年末に、来年追加オーダーする必要のある本だけをタイトルによってソートしたいとします。特定の本の今年販売された冊数が在庫数より大きい場合は、さらに何冊かをオーダーする必要があると見なすことができます。

オーダーする必要のある本だけを選択する INCLUDE ステートメントは、次のようになります。

```
INCLUDE COND=(166, 4, GT, 162, 4), FORMAT=BI
```

Number in stock (在庫数)

Number sold (販売数)

以下は、この INCLUDE ステートメントを作成するためのステップです。

表 13. オーダーする必要のある本についての INCLUDE ステートメントの作成ステップ

ステップ	アクション
1	少なくとも 1 つのブランクを入れてから、 INCLUDE を入力する
2	少なくとも 1 つのブランクを入れてから、 COND= を入力する
3	以下のものをコンマで区切り、括弧で囲んで入力する <ol style="list-style-type: none">販売数 (Number Sold) フィールドの位置、長さ、およびデータ・フォーマットより大を表す比較演算子 GT (比較演算子は 31 ページの図 1 に示されています)在庫数 (Number In Stock) フィールドの位置、長さ、およびデータ・フォーマット

以下の比較演算子から選択できます。

比較演算子	意味
EQ	等しい
NE	等しくない
GT	より大
GE	より大または等しい
LT	より小
LE	より小または等しい

図 1. 比較演算子

SORT ステートメントは、**INCLUDE** ステートメントの前または後のいずれかに置くことができます。制御ステートメントはどのような順序で指定しても構いません。しかし、処理される順序で制御ステートメントをコーディングする習慣をつけておくと有用です。すべての制御ステートメントの処理順序を示すフローチャートについては、193 ページの『付録 C. 制御ステートメントの処理順序』を参照してください。

```
INCLUDE COND=(166,4,BI,GT,162,4,BI)
SORT FIELDS=(1,75,CH,A)
```

これは、タイトルの昇順で、入力レコードの選択サブセットをソートします。表 14 は、ソート済みデータ・セットを示しています。

表 14. 販売数 (Number Sold) が在庫数 (Number In Stock) よりも大きくなっている本

Book Title	Number In Stock	Number Sold
1 75	162 165	166 169
ADVANCED TOPICS IN PSYCHOANALYSIS	1	12
COMPUTER LANGUAGES	5	29
COMPUTERS: AN INTRODUCTION	20	26
CRISES OF THE MIDDLE AGES	14	17
EDITING SOFTWARE MANUALS	13	32
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	2	32
INTRODUCTION TO BIOLOGY	6	11
MODERN ANTHOLOGY OF WOMEN POETS	1	26
NUMBERING SYSTEMS	6	27
STRATEGIC MARKETING	3	35
SUPPLYING THE DEMAND	0	32
SYSTEM PROGRAMMING	4	23
THE COMPLETE PROOFREADER	7	19

入力レコードのサブセットをさらに削減して、COR 出版社からオーダーする必要のある本だけをソートしたいとします。この場合は、次の 2 つの条件が真でなければなりません。

- 販売数が在庫数より大きい。
- COR から出版された本である。

レコードの組み込みまたは省略

2 番目の条件を追加するには、論理 AND を追加することによって INCLUDE ステートメントを拡張し、出版社 (Publisher) フィールドの内容を文字ストリング『COR』と比較します (定数の指定方法については、35 ページの『定数の作成』を参照してください)。出版社 (Publisher) フィールドは 4 バイトの長さなので、『COR』の右側に 1 つのブランクが埋め込まれます。

```
INCLUDE COND=(166,4,BI,GT,162,4,BI,AND,106,4,CH,EQ,C'COR')
SORT FIELDS=(1,75,CH,A)
```

表 15 は、結果を示しています。

表 15. 販売数 (Number Sold) が在庫数 (Number In Stock) より大きくなっている COR 本

Book Title	Publisher	Number In Stock	Number Sold
1 75	106 109	162 165	166 169
CRISES OF THE MIDDLE AGES	COR	14	17
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	COR	2	32
MODERN ANTHOLOGY OF WOMEN POETS	COR	1	26
SUPPLYING THE DEMAND	COR	0	32

もう 1 つの例として、次のような INCLUDE および SORT ステートメントを作成すると、コース 00032 および 10347 に関する本だけをソートすることができます。

```
INCLUDE COND=(115,5,CH,EQ,C'00032',OR,115,5,CH,EQ,C'10347')
SORT FIELDS=(115,5,CH,A)
```

注: この例で、C'00032' の代わりに C'32' を使用することはできません。というのは、文字定数は右側にブランクを埋め込まれるからです。DFSORT は、埋め込みおよび切り捨てを行う際に以下の規則を使用します。

- | | |
|------|----------------------------------|
| 埋め込み | データに充てん文字 (通常はゼロまたはブランク) を追加します。 |
| 切り捨て | ストリングの先行部分または末尾部分を削除または省略します。 |

比較では、以下の規則が適用されます。

- フィールドとフィールドの比較では、短い方のフィールドに適宜 (ブランクまたはゼロが) 埋め込まれます。
- フィールドと定数の比較では、フィールドの長さに合わせて定数が埋め込まれるか切り捨てられます。10 進定数は、左側で埋め込みまたは切り捨てが行われます。文字定数と 16 進定数は、右側で埋め込みまたは切り捨てが行われます。

OMIT ステートメントの作成

一般の読書ではなく教育コースに使用されるすべての本をタイトルによってソートしたいとします。この場合は、学部コース (Course Department) フィールドがブランクのレコードを除外する OMIT ステートメントを使用することができます。

OMIT ステートメントのフォーマットは、INCLUDE ステートメントのフォーマットと同じです。一般の読書用の本を除外するには、次のステートメントを作成します。

```
OMIT COND=(110,5,CH,EQ,C' ')
SORT FIELDS=(1,75,CH,A)
```

表 16 は、ソート済みデータ・セットを示しています。

表 16. 教育コースに必要なでない本を省いたソート済みデータ・セット

Book Title	Course Department
1 75	110 114
ADVANCED TOPICS IN topCHOANALYSIS	PSYCH
COMPUTER LANGUAGES	COMP
COMPUTERS: AN INTRODUCTION	COMP
CRISES OF THE MIDDLE AGES	HIST
EDITING SOFTWARE MANUALS	ENGL
EIGHTEENTH CENTURY EUROPE	HIST
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	ENGL
INTRODUCTION TO BIOLOGY	BIOL
INTRODUCTION TO PSYCHOLOGY	PSYCH
MODERN ANTHOLOGY OF WOMEN POETS	ENGL
NUMBERING SYSTEMS	COMP
SHORT STORIES AND TALL TALES	ENGL
STRATEGIC MARKETING	BUSIN
SUPPLYING THE DEMAND	BUSIN
SYSTEM PROGRAMMING	COMP
THE COMPLETE PROOFREADER	ENGL
THE INDUSTRIAL REVOLUTION	HIST
VIDEO GAME DESIGN	COMP

INCLUDE および OMIT で許容される比較

表 17 と 34 ページの表 18 は、INCLUDE および OMIT で一般に使用されるデータ・フォーマットに関して許容され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を示しています。INCLUDE および OMIT で使用できるすべてのデータ・フォーマットの詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

表 17. 許容されるフィールドとフィールドの比較

フィールド・フォーマット	BI	FI	CH	ZD	PD	FS	UFF	SFF
BI	✓		✓					
FI		✓						
CH	✓		✓					
ZD				✓	✓			
PD				✓	✓			
FS						✓	✓	✓

表 17. 許容されるフィールドとフィールドの比較 (続き)

フィールド・フォーマット	BI	FI	CH	ZD	PD	FS	UFF	SFF
UFF						✓	✓	✓
SFF						✓	✓	✓

表 18. 許容されるフィールドと定数の比較

フィールド・フォーマット	文字ストリング	16 進数ストリング	10 進数
BI	✓	✓	✓
FI			✓
CH	✓	✓	
ZD			✓
PD			✓
FS			✓
UFF			✓
SFF			✓

例えば、著者の名前によってソートし、著者のラストネームが “M” で始まる本だけを組み込みたい場合は、文字フォーマットであるバイト 76 (著者のラストネームの最初のバイト) の内容を、文字ストリングまたは 16 進数ストリングと比較することができます。

```
INCLUDE COND=(76,1,CH,EQ,C'M')
SORT FIELDS=(76,15,CH,A)
```

または

```
INCLUDE COND=(76,1,CH,EQ,X'D4')
SORT FIELDS=(76,15,CH,A)
```

さらに、在庫数が 10 未満の本だけを在庫数によってソートしたい場合には、2 進数形式である在庫数 (Number In Stock) フィールドの内容を 10 進定数または 16 進数ストリングと比較することができます。

```
INCLUDE COND=(162,4,BI,LT,10)
SORT FIELDS=(162,4,BI,A)
```

または

```
INCLUDE COND=(162,4,BI,LT,X'0000000A')
SORT FIELDS=(162,4,BI,A)
```

16 進定数の場合は、埋め込みと切り捨ての規則を忘れないでください。X'0A' を指定した場合、ストリングは、左側ではなく右側に埋め込まれます。10 進定数の場合は、10 または +10 を使用でき、埋め込みや切り捨てを心配する必要はありません。

定数の作成

文字ストリング、16 進数ストリング、および 10 進数を書く際のフォーマットを、以下に示します。

文字ストリング

文字ストリングを書く際のフォーマットは、次のとおりです。

```
C'x...x'
```

ここで、*x* は EBCDIC 文字です。例えば、C'FERN'。

ストリング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として指定する必要があります。例えば、O'NEILL は C'O'NEILL' として指定してください。

特殊なキーワードを使用して、実行の現在日付の文字ストリングをさまざまな形式で指定することができます。これについては、「z/OS DFSORT アプリケーション・プログラミング・ガイド」に説明があります。例えば、位置 42 から始まる C'yyyy/mm/dd' 形式の 10 文字の日付が今日の日付と等しいレコードを選択したい場合は、次のステートメントを作成します。

```
INCLUDE COND=(42,10,CH,EQ,DATE1(/))
```

また、特殊なキーワードを使用して、(実行の現在日付を基準にした) 将来または過去の日付の文字ストリングをさまざまな形式で指定することもできます。これについては、「z/OS DFSORT アプリケーション・プログラミング・ガイド」に説明があります。例えば、位置 42 から始まる C'yyyy/mm/dd' 形式の 10 文字の日付が、30 日前から 30 日後までの間にあるレコードを選択したい場合は、次のステートメントを作成します。

```
INCLUDE COND=(42,10,CH,GE,DATE1(/)-30,AND,42,10,CH,LE,DATE1(/)+30)
```

16 進数ストリング

16 進数ストリングを書く際のフォーマットは、次のとおりです。

```
X'yy...yy'
```

ここで、*yy* は一対の 16 進数字です。例えば、X'7FB0'。

10 進数

10 進数を書く際のフォーマットは、次のとおりです。

```
n...n     or     ±n...n
```

ここで、*n* は 10 進数字です。例えば、24、+24、-24、など。

10 進数にはコンマまたは小数点を含めてはなりません。

特殊なキーワードを使用して、実行の現在日付を表す 10 進数をさまざまな形式で指定することができます。これについては、「z/OS DFSORT アプリケーション・プ

「プログラミング・ガイド」に説明があります。例えば、位置 28 から始まる P'yyyyddd' (16 進数 yyyydddC) という 4 バイトのパック 10 進数日付が今日の日付と等しいレコードを選択したい場合は、次のステートメントを作成します。

```
INCLUDE COND=(28,4,PD,EQ,DATE3P)
```

また、特殊なキーワードを使用して、(実行の現在日付を基準にした) 将来または過去の日付を表す 10 進数をさまざまな形式で指定することもできます。これについては、「z/OS DFSORT アプリケーション・プログラミング・ガイド」に説明があります。例えば、位置 28 から始まる P'yyyyddd' (16 進数 yyyydddC) という 4 バイトのパック 10 進数日付が、30 日前から 30 日後までの間にあるレコードを選択したい場合は、次のステートメントを作成します。

```
INCLUDE COND=(28,4,PD,GE,DATE3P-30,AND,28,4,PD,LE,DATE3P+30)
```

INCLUDE および OMIT の数値テスト

従業員 (Employees) フィールドの値の一部に無効な数値データが入っている可能性があります。それらの値を持つレコード (ある場合) を選択したいものと想定します。4 バイトの従業員フィールドの各バイトには、'0' から '9' が含まれている必要があります。これ以外の文字 (例えば、'A' や '.') は無効であるとしします。'1234' は、すべてが数字であるので、有効な数値です。'12.3' は、非数値が含まれているので無効な数値です。INCLUDE ステートメントの数値テスト機能の 1 つを使用すると、次のように必要なレコードを収集できます。

```
INCLUDE COND=(18,4,FS,NE,NUM)
```

フィールド内の値 (18,4,FS) が数値 (NUM) と等しくない (NE) 場合、レコードが組み込まれます。出力データ・セット内のレコードは、フィールドに非数値がある ('0' から '9' 以外の文字がフィールド内のどこかにある) レコードです。

数値または非数値のテストを指定するには、NUM を使用します。

数値があるかどうかをテストするには EQ を使用し、非数値があるかどうかをテストするには NE を使用します。

文字数値 (すべてのバイト内の '0' から '9') があるかどうかをテストしたい場合は、フィールドの FS フォーマットを使用します。

ゾーン 10 進数 (すべての符号なしバイト内の '0' から '9'、符号付きバイト内の X'F0' から X'F9'、X'D0' から X'D9' または X'C0' から X'C9') があるかどうかをテストしたい場合は、フィールドの ZD フォーマットを使用します。

パック 10 進数 (すべての桁の 0 から 9、符号の F、D または C) があるかどうかをテストしたい場合は、フィールドの PD フォーマットを使用します。

売り上げ (Revenue) フィールドと利益 (Profit) フィールドにパック 10 進数がある (つまり、これらのフィールドに無効なパック 10 進数値がない) レコードのみを組み込む INCLUDE ステートメントは、次のとおりです。

```
INCLUDE COND=(22,6,PD,EQ,NUM,AND,28,6,PD,EQ,NUM)
```

INCLUDE および OMIT の場合のサブストリング検索

Biology (生物学)、History (歴史学)、Business (ビジネス)、および Psychology (心理学) 学部の本だけを選択したいとします。以前に学習した内容を基にすると、次の INCLUDE ステートメントを使用することにより、これらの本を選択できます。

```
INCLUDE COND=(106,5,CH,EQ,C'BIOL',OR,
               106,5,CH,EQ,C'HIST',OR,
               106,5,CH,EQ,C'BUSIN',OR,
               106,5,CH,EQ,C'PSYCH')
```

ただし、もっと多くの学部を組み込みたい場合は、さらに多くの入力を行う必要があります。その代わりに、INCLUDE および OMIT のサブストリング検索機能の 1 つを使用すると、次のようなより単純な形式でステートメントを作成することができます。

```
INCLUDE COND=(106,5,SS,EQ,C'BIOL ,HIST ,BUSIN,PSYCH')
```

サブストリング検索 (SS フォーマット) では、フィールドを 1 回書き、文字定数を書くだけで、検索したいすべてのストリングが組み込まれます。フィールドの値がいずれかのストリング (例えば、"BUSIN") と一致すれば、そのレコードは組み込まれます。フィールドの値がどのストリングとも一致しない場合、そのレコードは省略されます。

各ストリングの長さは、フィールドの長さとは一致していなければなりません。学部 (Department) フィールドは 5 文字なので、4 文字の "BIOL" と "HIST" の場合は終わりに 1 つのブランクを追加する必要がありますが、5 文字の "BUSIN" と "PSYCH" の場合は必要ありません。

サブストリング検索を使用するもう 1 つの方法として、フィールド内の定数を検索する方法があります。例えば、Title に "INTRODUCTION" を含む本だけを選択したい場合は、次の INCLUDE ステートメントを使用できます。

```
INCLUDE COND=(1,75,SS,EQ,C'INTRODUCTION')
```

出力として選択される本は、次のとおりです。

```
COMPUTERS: AN INTRODUCTION
INTRODUCTION TO PSYCHOLOGY
INTRODUCTION TO BIOLOGY
```

VB データ・セットについての考慮事項

SORT および MERGE ステートメントのところで学習した同じ VB データ・セットに関する考慮事項が、INCLUDE および OMIT ステートメントにも適用されます。

開始位置

VB レコードの組み込みまたは省略のために比較フィールドをコーディングするとき、4 バイトの RDW のための 4 を開始位置に追加するのを忘れないでください。例えば、次の INCLUDE ステートメントは、VB レコードの第 3 から第 5 データ・バイトの PD フィールドを、VB レコードの第 6 から第 8 バイトの PD フィールドと比較します。

```
INCLUDE COND=(7,3,PD,EQ,10,3,PD)
```

短制御フィールド

VB レコードが短比較フィールドを持つと分かっている場合は、必要であれば VLSCMP オプションを指定することにより、DFSORT が終了するのを阻止することができます。以下に例を示します。

```
OPTION COPY,VLSCMP
INCLUDE COND=(21,8,CH,EQ,C'Type 200')
```

VLSCMP は、欠落している比較フィールド・バイトを一時的に 2 進ゼロで置き換えるよう DFSORT に指示するもので、したがって短フィールドが有効に比較されるようになります (出力レコードではゼロは保持されません)。上記の例では、28 バイトより短いレコードは組み込まれません。これは、フィールドの欠落バイトに追加された 2 進ゼロにより、28 バイトより短いレコードが 'Type 200' と等しくなることはないからです。

VB レコードが短比較フィールドを持つ場合に DFSORT が終了するのを阻止するもう 1 つの方法として、VLSHRT オプションを指定する方法 (該当する場合) があります。以下に例を示します。

```
OPTION COPY,VLSHRT
INCLUDE COND=(21,8,CH,EQ,C'Type 200')
```

VLSHRT は、短フィールドに関与する比較を偽として扱うよう DFSORT に指示します。上記の例では、28 バイトより短いレコードは組み込まれません。

重要: NOVLSCMP および NOVLSHRT が有効である場合、DFSORT は、VB レコード内で短比較フィールドを検出すると終了します。

DFSORT の VLSCMP および VLSHRT オプションの詳細については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

要約

本章では、より効率的に処理を行うために、入力レコードのサブセットだけを選択する 3 つの方法について説明しました。INCLUDE および OMIT ステートメントを作成し、許容される比較演算子、各種の定数、数値テスト、およびサブstring検索を学習しました。

第 4 章 レコードの合計

English (英文学) 学部でそのすべてのコースに関する本の合計価格を知りたいとします。INCLUDE ステートメントを使用することにより、English 学部のレコードだけを組み込み、SORT および SUM ステートメントを使用することにより、本の価格を一緒に加算することができます。

SUM 制御ステートメントでは、等価の照合制御フィールドを持つレコードが見つかるたびに合計される、1 つ以上の数値フィールドを指定します (制御フィールドは SORT ステートメントで指定します)。SUM ステートメントで指定できるデータ・フォーマットは、2 進数 (BI)、固定小数点 (FI)、パック 10 進数 (PD)、ゾーン 10 進数 (ZD)、および浮動小数点 (FL) です。

English 学部のレコードだけの価格を合計するには、SUM ステートメントに価格 (Price) フィールドを指定し、SORT ステートメントに学部 (Department) フィールドを指定します。INCLUDE ステートメントは、SUM および SORT の処理前に English 学部のレコードだけを選択するので、組み込まれるすべてのレコードで学部フィールドが等しくなり、価格が合計されるようになります。(INCLUDE、SUM、および SORT ステートメントの処理順序を示すフローチャートについては、193 ページの『付録 C. 制御ステートメントの処理順序』を参照してください。)

レコードの合計には、次の 2 種類のフィールドが関与しています。

制御フィールド

SORT ステートメントで指定

要約フィールド

SUM ステートメントで指定

要約フィールドの内容は、同じ制御フィールドを持つレコード (「重複」レコードとも呼ばれる) のグループに関して合計されます。

SUM ステートメントの作成

価格を合計する SUM ステートメントは、次のようになります。

```
SUM FIELDS=(170, 4, BI)
└──┬──> Price (価格)
```

以下に、この SUM ステートメントを作成するためのステップを示します。

表 19. 価格を求める SUM ステートメントの作成ステップ

ステップ	アクション
1	少なくとも 1 つのブランクを入れてから、 SUM を入力する
2	少なくとも 1 つのブランクを入れてから、 FIELDS= を入力する

レコードの合計

表 19. 価格を求める SUM ステートメントの作成ステップ (続き)

ステップ	アクション
3	価格フィールドの位置、長さ、およびデータ・フォーマットをコンマで区切り、括弧で囲んで入力する。

INCLUDE、SORT、および SUM ステートメントを、以下に示します。

```
INCLUDE COND=(110,5,CH,EQ,C'ENGL')
SORT FIELDS=(110,5,CH,A)
SUM FIELDS=(170,4,BI)
```

価格が合計されると、最終合計が 1 つのレコードの価格フィールドに現れ、他のレコードは削除されます。したがって、結果 (表 20) は、合計が収められた 1 つのレコードだけになります。レコードが元の順序を保持するよう指定する場合は、どのレコードを出力させるかを制御できます。例の場合、デフォルトでは、等価の照合制御フィールドを持つレコードは元の順序で出力されます (EQUALS が有効です)。元の順序を保持しながらレコードを合計する場合、DFSORT は、合計を収めるレコードとして最初のレコードを選択します。

表 20. English 学部の価格の合計

Book Title	Course Department	Price
1 75	110 114	170 173
INKLINGS: AN ANTHOLOGY OF YOUNG POETS ¹	ENGL ²	4640 ³

注:

¹ 要約レコードのフィールドの中には、表 20 のブック・タイトル (Book Title) のように意味のないフィールドもあります。次の章で、意味のないフィールドを省く 2 通りの方法について学習します。

² 制御フィールドとして指定されます。

³ 要約フィールドとして指定されます。

English 学部でそのそれぞれの コース別の本の合計価格を知りたいとします。この場合は、やはり INCLUDE を使用して English 学部の本だけを選択し、SUM ステートメントに価格フィールドを指定しますが、SORT ステートメントにはコース番号 (Course Number) を指定します。

```
INCLUDE COND=(110,5,CH,EQ,C'ENGL')
SORT FIELDS=(115,5,CH,A)
SUM FIELDS=(170,4,BI)
```

► Price (価格)

表 21 は、1 コースにつき 1 レコードの結果を示しています。

表 21. English 学部の価格の合計

Book Title	Course Number	Price
1 75	115 119	170 173

表 21. English 学部の価格の合計 (続き)

Book Title	Course Number	Price
SHORT STORIES AND TALL TALES	10054	1520
EDITING SOFTWARE MANUALS	10347	2075
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	10856	1045

2 つの要約フィールドを使用する例の場合に、在庫調べの目的で、それぞれの出版社ごとに在庫数と販売数を別々に合計したいとします。

このアプリケーションの場合は、**SORT** ステートメントに制御フィールドとして出版社を指定し、**SUM** ステートメントに要約フィールドとして在庫数と販売数を指定します。今回は入力データ・セット内のすべてのレコードを使用したいので、**INCLUDE** または **OMIT** ステートメントをコーディングする必要はありません。(SORT および SUM は、INCLUDE または OMIT ステートメントと一緒に使用することもできますし、なしでも使用できます。)

```

SORT FIELDS=(106, 4, CH, A)
SUM FIELDS=(162, 4, 166, 4), FORMAT=BI

```

表 22 は、1 出版社につき 1 レコードの結果を示しています。

表 22. それぞれの出版社ごとの在庫数と販売数の合計

Book Title	Publisher	Number In Stock	Number Sold
1 75	106 109	162 165	166 169
LIVING WELL ON A SMALL BUDGET	COR	103	161
COMPUTER LANGUAGES	FERN	19	87
VIDEO GAME DESIGN	VALD	42	97
COMPUTERS: AN INTRODUCTION	WETH	62	79

重複制御フィールドを持つレコードを抑止する

値を合計することは別として、**SUM** を使用して、重複制御フィールドを持つレコード (「重複レコード」とも呼ばれる) を削除することもできます。

例えば、出版社を昇順にリストする場合に、それぞれの出版社は 1 度だけしか出力する必要がない場合があります。 **SORT** ステートメントだけを使用した場合、COR は 7 回現れ (ファイル内の 7 冊の本が COR から出版されているので)、FERN は 4 回、VALD は 5 回、WETH は 4 回現れます。

下記のように **SUM** ステートメントに **FIELDS=NONE** を指定すると、**DFSORT** は 1 出版社につき 1 レコードだけしか書き込みません。

レコードの合計

```
SORT FIELDS=(106,4,CH,A)
SUM FIELDS=NONE
```

表 23 は、結果を示しています。

表 23. 重複を削除した出版社のリスト

Book Title	Publisher
1 75	106 109
LIVING WELL ON A SMALL BUDGET	COR
COMPUTER LANGUAGES	FERN
VIDEO GAME DESIGN	VALD
COMPUTERS: AN INTRODUCTION	WETH

133 ページの『第 11 章 ICETOOL ユーティリティの使用』で、ICETOOL の強力な SELECT、OCCUR、および SPLICE オペレーターの使用法を学習します。これらのオペレーターを使用すると、重複および非重複レコードに関連した数多くの機能を実行できます。

オーバーフローの処理

合計がそれに使用可能なスペースよりも大きくなると、オーバーフロー が起こります。例えば、2 バイトの 2 進数フィールド (符号なし) に 'X'FFFF' が含まれているときに、'X'0001' を加算すると、合計には 2 バイトより多くのバイトが必要となるので、オーバーフローが起こります。

```
FFFF
0001
10000
```

オーバーフローが起こった場合、関与する 2 つのレコードの要約フィールドは加算されません。すなわち、レコードはそのまま保持され、どちらも削除されません。

場合によっては、INREC 制御ステートメントを使用して要約フィールドにゼロを埋め込むことによって、オーバーフローを訂正することができます。75 ページの『値を合計するときのオーバーフローを防止する』に、これを行う方法を示しています。

VB データ・セットについての考慮事項

SORT、MERGE、INCLUDE、および OMIT ステートメントのところで学習した同じ VB データ・セットに関する考慮事項が、SUM ステートメントにも適用されます。

開始位置

VB レコードの要約フィールドをコーディングするときは、4 バイトの RDW のための 4 を開始位置に追加するのを忘れないでください。例えば、次の SUM ステートメントは、VB レコードの第 3 から第 5 データ・バイトに PD 要約フィールドを指定します。

```
SUM FIELDS=(7,3,PD)
```

短要約フィールド

VB レコードが短要約フィールドを持つと分かっている場合は、必要であれば VLSHRT オプションを指定することにより、DFSORT が終了するのを阻止することができます。以下に例を示します。

```
OPTION VLSHRT  
SORT FIELDS=(6,2,CH,A)  
SUM FIELDS=(21,8,ZD)
```

VLSHRT は、短要約フィールドを持つレコードを未合計のままにするよう DFSORT に指示します。すなわち、要約操作に関与する 2 つのレコードの一方が短要約フィールドを持つときは、レコードはそのまま保持され、どちらのレコードも削除されません。

重要: NOVLSHRT が有効である場合、DFSORT は、VB レコード内で短要約フィールドを検出すると終了します。

DFSORT の VLSHRT オプションの詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

要約

本章では、データ・セットのレコードの合計方法を紹介しました。SUM ステートメントを使用して等価の制御フィールドを持つレコードを合計する方法、および重複制御フィールドを持つレコードを抑止する方法を説明しました。次に、データ・セットを再フォーマットするための OUTREC と INREC の使用法を学習します。

レコードの合計

第 5 章 固定フィールドを持つレコードの再フォーマット

INREC、OUTREC、および OUTFIL 制御ステートメントを使用して、データ・セット内のレコードを再フォーマットすることができます。これらのステートメントは、別々に使用することも一緒に使用することもできます。

この章では、**固定フィールド**を持つレコードを再フォーマットする方法について説明します。固定フィールドとは、どのレコードでも同じ位置で開始し、同じ長さを持つフィールドです。この次の章では、**可変フィールド**を持つレコードを同じように再フォーマットする方法を説明します。可変フィールドとは、レコードごとに開始位置と長さが異なるフィールド (コンマ区切り値 (CSV) など) です。

INREC、OUTREC、および OUTFIL を使用すると、ソート、コピー、またはマージ中に、以下を含む、広範囲にわたるタスクを実行できます。

- フィールドを削除する。
- フィールドを再配列する。
- 分離文字 (ブランク、ゼロ、ストリング、現在日付、将来の日付、過去の日付、および現在時刻) を挿入する。
- データをその 16 進表記に変換する。
- 小文字を大文字に変換するか、大文字を小文字に変換する。
- 数値を各種の印刷可能フォーマットに変換する (3 桁ごとの区切り付きまたはなし、小数点付きまたはなし、先行ゼロまたはゼロ抑止、および符号付きまたはなし)。
- 数値をあるフォーマットから別のフォーマットに変換する。
- 数値と 10 進定数を使用して算術演算を実行する。
- ルックアップ・テーブルを使用して値を変更する。
- シーケンス番号を挿入する。
- データの左寄せ、データの右寄せ、データの左詰め、およびデータの右詰め (ブランクなどを削除) を行う。

レコードの再フォーマットは、次の 3 とおりの方法のいずれかで行うことができます。

- **BUILD:** 各レコードのすべての項目を 1 つずつ指定して、レコードを再フォーマットします。BUILD では、再フォーマットされるレコード内の必要な項目、およびそれらの項目が表示される順序を完全に制御できます。フィールドおよび定数の削除、再編成、および挿入が可能です。

注: INREC および OUTREC には、BUILD パラメーターまたは FIELDS パラメーターのどちらかを使用できます。OUTFIL には、BUILD パラメーターまたは OUTREC パラメーターのどちらかを使用できます。

- **OVERLAY:** 特定の桁をオーバーレイする項目だけを指定して、各レコードを再フォーマットします。OVERLAY を使用すると、レコード全体に影響を与えることなく、特定の既存桁を変更できます。
- **IFTHEN 文節:** BUILD または OVERLAY 項目が、所定の基準を満たすレコードにどのように適用されるかを指定して、レコードごとに異なる方法で再フォーマット

レコードの再フォーマット

ットします。 **IFTHEN** 文節を使用すると、高度な条件論理を使用して、さまざまなレコード・タイプの再フォーマット方法を選択できます。

INREC、**OUTREC**、および **OUTFIL** ステートメントはすべて、同じ機能を実行できます。ただし、次のような違いがあります。

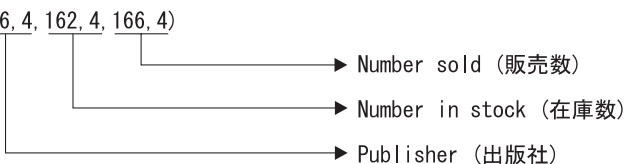
- **INREC** ステートメントは、ソート、コピー、またはマージの**前**にレコードを再フォーマットします。 **INREC** が他の制御ステートメントにどのような影響を与えるかについては、74 ページの『**INREC** と一緒に他のステートメントを使用する』を参照してください。
- **OUTREC** ステートメントは、ソート、コピー、またはマージの**後**でレコードを再フォーマットします。
- **OUTFIL** ステートメントは、ソート、コピー、またはマージの**後**で **OUTFIL** 出力データ・セットのレコードを再フォーマットできます。異なる **OUTFIL** データ・セットに対しては、異なる再フォーマット・パラメーターを使用できます。本章で提示している **INREC** および **OUTREC** ステートメントに関する情報は、**OUTFIL** ステートメントにも適用されます。**OUTFIL** については、89 ページの『第 7 章 複数の出力データ・セットおよび報告書の作成』で説明します。

BUILD または FIELDS を使用したソート後のレコードの再フォーマット

前の章では、**SUM** ステートメントを使用して、それぞれの出版社ごとに在庫の本および販売された本の価格を合計しました。ここで、**OUTREC** ステートメントの **FIELDS** または **BUILD** パラメーターを使用することにより、アプリケーションに不要なすべてのフィールド（すなわち、その内容が合計レコードでは意味のないフィールド）を削除できます。出版社 (Publisher)、在庫数 (Number In Stock)、および販売数 (Number Sold) フィールドだけしか書き込まれないので、出力レコード長は 12 バイトに減少します。

OUTREC ステートメントは、次のようになります。

```
OUTREC FIELDS=(106, 4, 162, 4, 166, 4)
```



→ Number sold (販売数)
→ Number in stock (在庫数)
→ Publisher (出版社)

以下に、この **OUTREC** ステートメントを作成するためのステップを示します。

表 24. レコードを再フォーマットする **OUTREC** ステートメントの作成ステップ

ステップ	アクション
1	少なくとも 1 つのブランクを入れてから、 OUTREC を入力する
2	少なくとも 1 つのブランクを入れてから、 FIELDS= (または BUILD=) を入力する

表 24. レコードを再フォーマットする OUTREC ステートメントの作成ステップ (続き)

ステップ	アクション
3	以下のものをコンマで区切り、括弧で囲んで入力する <ol style="list-style-type: none"> 1. 出版社 (Publisher) フィールドの位置と長さ 2. 在庫数 (Number In Stock) フィールドの位置と長さ 3. 販売数 (Number Sold) フィールドの位置と長さ

SORT、SUM、および OUTREC ステートメントを、以下に示します。

```
SORT FIELDS=(106,4,CH,A)
SUM FIELDS=(162,4,BI,166,4,BI)
OUTREC FIELDS=(106,4,162,4,166,4)
```

表 25 は、出力を示しています。

表 25. 出版社 (Publisher)、在庫数 (Number In Stock)、および販売数 (Number Sold) フィールドだけの書き込み

Publisher	Number In Stock	Number Sold
1 4	5 8	9 12
COR	103	161
FERN	19	87
VALD	42	97
WETH	62	79

表 25 で示された出力レコードを印刷または表示した場合、在庫数 (Number In Stock) および販売数 (Number Sold) フィールドは、実際には判読不能な 2 進数値です。2 進数およびその他の判読不能な数値を読みやすい形式に変換する方法については、本章の後半で学習します。

出力データ・セットに対して次の SORTOUT DD ステートメントを使用するとします。

```
//SORTOUT DD DSN=MY.OUTPUT,DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(5,5))
```

DFSORT は、新規出力データ・セット (MY.OUTPUT) の LRECL を再フォーマット済み出力レコード長に自動的に設定するので、この場合は LRECL=12 が設定されます。DFSORT がユーザーに代わって適切に LRECL を設定するので、SORTOUT DD ステートメントに LRECL を指定する必要はありません。

SORTOUT データ・セットがすでに LRECL 値を持っている場合 (OLD データ・セット) や、SORTOUT DD ステートメントで LRECL 値を指定することにした場合、その値は、DFSORT が LRECL に設定する値を指定変更します。この結果として、もし再フォーマットされたレコードの計算された長さがユーザー指定の LRECL と一致しない場合には、故意でも故意でなくても、出力レコードの埋め込みや切り捨てが行われる可能性があります。

一般には、NEW SORTOUT データ・セットの RECFM、LRECL、および BLKSIZE は DFSORT に設定させるようにするのが最善です (ただし、これらの値を指定変更

レコードの再フォーマット

する特別な理由がある場合は別です)。 OLD SORTOUT データ・セットの場合は、既存の RECFM、LRECL、および BLKSIZE 値がご使用の制御ステートメントに適切であることを確認してください。

フィールドを予約スペースに再配列する

各フィールドは常に、ユーザーが指定した順序で現れます。したがって、表 26 に示されたように、販売数 (Number Sold) を在庫数 (Number In Stock) より前に出力させたい場合には、OUTREC ステートメントにおけるそれらの順序を逆にします。

```
SORT FIELDS=(106,4,CH,A)
SUM FIELDS=(162,4,BI,166,4,BI)
OUTREC FIELDS=(106,4,166,4,162,4)
```

表 26. フィールドの再配列

Publisher	Number Sold	Number In Stock
1 4	5 8	9 12
COR	161	103
FERN	87	19
VALD	97	42
WETH	79	62

2 進ゼロの挿入

最後の例を作成する際、レコードを再フォーマットして、在庫数の後に新規の 4 バイトの 2 進数フィールド (バイト 13 から始まる) を組み込みたいとします。この場合、新規フィールドのプレースホルダーとして (後でデータを埋め込めるように) 2 進ゼロを挿入することができます。単一の 2 進ゼロを指定するには、Z または 1Z を使用できます。 n 個の 2 進ゼロを指定するには、nZ を使用できます。

4 個の 2 進ゼロを挿入するには、最後のフィールドの後に 4Z を書き込んでください。

```
SORT FIELDS=(106,4,CH,A)
SUM FIELDS=(162,4,BI,166,4,BI)
OUTREC FIELDS=(106,4,166,4,162,4,4Z)
```

表 27 は、結果を示しています。

表 27. 2 進ゼロの挿入

Publisher	Number Sold	Number In Stock	X'0...0'
1 4	5 8	9 12	13 16
COR	161	103	0...0
FERN	87	19	0...0
VALD	97	42	0...0
WETH	79	62	0...0

ブランクの挿入

出力データ・セットが文字データだけを含む場合は、次の SORTOUT DD ステートメントを書くことによって印刷できます。

```
//SORTOUT DD SYSOUT=A
```

OUTREC ステートメントを使用して、フィールドをブランクで区切り、余白を作成するようにすると、印刷出力をもっと読みやすくすることができます。ブランクは、フィールドの前、間、または後に挿入できます。単一ブランクを指定するには、X または 1X を使用できます。n 個のブランクを指定するには、nX を使用できます。

例えば、出版社 (Publisher) およびタイトル (Book Title) フィールドだけを印刷し、出版社フィールドを最初に出力することにします。出版社名のほとんどは 4 バイトの出版社フィールド全体を埋め尽くすので、2 つのフィールドをブランクで区切らない場合は、出版社の名前がタイトルに達してしまいます。さらに、マージンがないと、出版社の名前が用紙の端から始まってしまいます。

20 ブランクの左マージンを作成し、フィールドを 10 ブランクで区切るようにすると、印刷出力が読みやすくなります。

20 ブランクを挿入するには、最初のフィールドの前に 20X を書き込みます。10 ブランクを挿入するには、2 つのフィールドの間に 10X を書き込みます。この SORT ステートメントは、タイトルによって昇順にレコードをソートします (SORT、COPY、または MERGE が必ず必要であることを忘れないでください)。

```
SORT FIELDS=(1,75,CH,A)  
OUTREC BUILD=(20X,106,4,10X,1,75)
```

表 28 は、結果を示しています。

レコードの再フォーマット

表 28. ブランクを挿入した後の出力

Publisher				Book Title	
1	20	21 24	25 34	35	190
(20 ブランク)				(10 ブランク)	
		FERN		ADVANCED TOPICS IN PSYCHOANALYSIS	
		FERN		COMPUTER LANGUAGES	
		WETH		COMPUTERS: AN INTRODUCTION	
		COR		CRISES OF THE MIDDLE AGES	
		VALD		EDITING SOFTWARE MANUALS	
		WETH		EIGHTEENTH CENTURY EUROPE	
		COR		INKLINGS: AN ANTHOLOGY OF YOUNG POETS	
		VALD		INTRODUCTION TO BIOLOGY	
		COR		INTRODUCTION TO PSYCHOLOGY	
		COR		LIVING WELL ON A SMALL BUDGET	
		COR		MODERN ANTHOLOGY OF WOMEN POETS	
		FERN		NUMBERING SYSTEMS	
		COR		PICK'S POCKET DICTIONARY	
		VALD		SHORT STORIES AND TALL TALES	
		VALD		STRATEGIC MARKETING	
		COR		SUPPLYING THE DEMAND	
		WETH		SYSTEM PROGRAMMING	
		FERN		THE COMPLETE PROOFREADER	
		WETH		THE INDUSTRIAL REVOLUTION	
		VALD		VIDEO GAME DESIGN	

フィールドを開始したい桁が分かっている場合は、使用するブランク数を計算する代わりに、それらを直接指定することができます。例えば、上記のように、出版社フィールドを桁 21 から、タイトル・フィールドを桁 35 から開始したい場合は、次の OUTREC ステートメントを書くことができます。

```
OUTREC BUILD=(21:106,4,35:1,75)
```

c: を使用すると、DFSORT は、データのある最後のフィールド (または桁 1) から c の前の桁までブランクを埋め込みます。したがって、21: は桁 1-20 にブランクを入れ、35: は桁 25-34 にブランクを入れます。

最後のフィールドとして c:X を指定することが、出力レコードのレコード長を c バイトに増大させる簡単な方法です。例えば、入力レコードの最初の 30 バイトを含み、ブランクで埋め込まれた 80 バイトの出力レコードを作成したい場合は、次の OUTREC ステートメントを使用できます。

```
OUTREC BUILD=(1,30,80:X)
```

ユーザーが LRECL を指定変更しない限り、DFSORT は SORTOUT データ・セットに対して LRECL=80 を自動的に設定します。

ストリングの挿入

OUTREC を使用すると、印刷出力をより読みやすくするほか、ストリングの挿入によって基本的な報告書フォーマットをセットアップすることもできます。ICETOOL の DISPLAY オペレーターまたは OUTFIL 制御ステートメントを使用すると、複雑な報告書を作成できます (これについては、本書の後半で説明しています)。ストリングのフォーマットを、以下に示します。

文字ストリング

文字ストリングを繰り返す場合のフォーマットは、次のとおりです。

```
C'x...x'
```

ここで、*x* は EBCDIC 文字です。例えば、C'FERN'。

文字ストリングを繰り返す場合のフォーマットは、次のとおりです。

```
nC'x...x'
```

ここで、*n* は 1 から 4095 です。文字ストリング (C'x...x') の *n* 回の繰り返しが、再フォーマットされたレコードに挿入されます。*n* を省略すると、代わりに 1 が使用されます。

ストリング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として指定する必要があります。例えば、O'NEILL は C'O'NEILL' と指定してください。

特殊なキーワードを使用して、実行の現在の日付または時刻を表す文字ストリングまたはバック 10 進定数を、さまざまな形式で挿入することができます (これについては、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」で詳しく説明しています)。例えば、各 80 バイト・レコードの終わりに以下を挿入したいとします。

- ストリング 'Timestamp: '
- C'yyyy/mm/dd' 形式の実行の日付
- ブランク、および
- C'hh:mm' 形式の実行の時刻

次の OUTREC ステートメントを使用できます。

```
OUTREC FIELDS=(1,80,C'Timestamp: ',DATE1(/),X,TIME2(:))
```

また、特殊なキーワードを使用して、(実行の現在日付を基準にした) 将来または過去の日付を表す文字ストリングまたはバック 10 進定数をさまざまな形式で指定することもできます。これについては、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」に説明があります。例えば、各 20 バイト・レコードの始めに以下を挿入したいものとします。

- 現在日付 - 7 日 (C'yyyyddd' 形式)
- ブランク、および
- 現在日付 + 14 日 (C'yyyyddd' 形式)。

次の OUTREC ステートメントを使用できます。

```
OUTREC FIELDS=(DATE3-7,X,DATE3+14,1,20)
```

16 進数ストリング

16 進数ストリングを作成する場合のフォーマットは、次のとおりです。

```
X'yy...yy'
```

ここで、yy は一対の 16 進数字です。例えば、X'7FB0'。

16 進数ストリングを繰り返す場合のフォーマットは、次のとおりです。

```
X'yy...yy'
```

ここで、n は 1 から 4095 です。16 進数ストリング (X'yy...yy') の n 回の繰り返しが、再フォーマットされたレコードに挿入されます。n を省略すると、1 が使用されます。

基本報告書のセットアップ

ブックストア・データ・セットから出版社 (Publisher) の名前と著者 (Author) の名前からなる非常に基本的な報告書を作成するには、次の OUTREC ステートメントを使用して、『Publisher is』と『Author is』を文字分離文字として入れることができます。

```
OPTION COPY  
OUTREC FIELDS=(11:C'Publisher is ',106,4,  
31:C'Author is ',91,15,X,76,15)
```

結果を、53 ページの表 29 に示しています。

表 29. 報告書の出力

Publisher								Author's First Name		Author's Last Name	
1	10	11	22	24	27	31	39	41	55	57	71
(10 ブランク)	Publisher is			FERN		Author is		ROBERT		MURRAY	
	Publisher is			COR		Author is		FRANK		DEWAN	
	Publisher is			COR		Author is		TOM		MILLER	
	Publisher is			VALD		Author is		LORI		RASMUSSEN	
	Publisher is			COR		Author is		KAREN		WILDE	
	Publisher is			WETH		Author is		JOKHI		DINSHAW	
	Publisher is			COR		Author is		CAROL		GUSTLIN	
	Publisher is			VALD		Author is		VICTOR		OJALVO	
	Publisher is			FERN		Author is		WILLIAM		BAYLESS	
	Publisher is			VALD		Author is		MARK		YAEGER	
	Publisher is			WETH		Author is		DON		GROSS	
	Publisher is			COR		Author is		PETER		COWARD	
	Publisher is			COR		Author is		LINDA		DUZET	
	Publisher is			FERN		Author is		ANN		GREEN	
	Publisher is			WETH		Author is		RAUL		CAUDILLO	
	Publisher is			VALD		Author is		LILIANA		AVRIL	
	Publisher is			VALD		Author is		CHIEN		WU	
	Publisher is			FERN		Author is		DIANNE		OSTOICH	
	Publisher is			WETH		Author is		ALICE		MUNGER	
	Publisher is			COR		Author is		GREG		BENDER	

これまで

本章ではこれまで、OUTREC ステートメントの BUILD または FIELDS パラメーターによって、指定されたフィールドおよび定数のみを含むように、出力データ・セットのレコードを再フォーマットする方法について説明してきました。さらに、プレースホルダーとして 2 進ゼロを挿入する方法、および出力データ・セットの印刷出力を読みやすくするためにブランクやストリングを使用する方法を説明しました。後続の節では、出力する入力フィールドをさまざまな方法で変更するための、その他の再フォーマット機能の使い方について学習します。

大文字から小文字への変更

コンピューター・コースで使用される本の著者をすべて表示するには、以下のステートメントを使用できます。

```
INCLUDE COND=(110,5,CH,EQ,C'COMP')
SORT FIELDS=(76,15,CH,A,91,15,CH,A)
OUTREC FIELDS=(91,15,X,76,15)
```

これらのステートメントに対応して作成される結果は、次のようになります。

レコードの再フォーマット

WILLIAM	BAYLESS
RAUL	CAUDILLO
JOKHI	DINSHAW
ROBERT	MURRAY
LORI	RASMUSSEN

著者のファーストネームとラストネームは、ブックストア・データ・セットでは大文字で保管されているので、大文字で表示されます。これらの名前を最初の文字を大文字で、その後の文字を小文字で表示させたい場合は、次の OUTREC ステートメントを使用できます。

```
OUTREC FIELDS=(91,1,92,14,TRAN=UTOL,X,76,1,77,14,TRAN=UTOL)
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

William	Bayless
Raul	Caudillo
Jokhi	Dinshaw
Robert	Murray
Lori	Rasmussen

p,m,TRAN=UTOL は、フィールド内の大文字 (A-Z) のそれぞれを等価の小文字 (a-z) に変更します。上記の例では、p,m,TRAN=UTOL は、著者のファーストネームの 2 番目の文字と著者のラストネームの 2 番目の文字から始め、大文字を小文字に変更します。それぞれの名前の最初の文字は TRAN=UTOL フィールドに含まれません。これらの大文字はそのまま出力レコードに移されます。

p,m,TRAN=LTOU は、p,m,TRAN=UTOL に類似していますが、これはフィールド内の小文字 (a-z) のそれぞれを等価の大文字 (A-Z) に変更します。

p,m,TRAN=ALTSEQ を使用すると、フィールド内の文字を、ALTSEQ ステートメントで指定した他の文字に変更することができます。例えば、ヌル文字 (16 進数 00) をブランク文字 (16 進数 40) に変更できます。

すべての TRAN パラメーターの詳細については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

数値フィールドから異なるフォーマットへの変換

47 ページの表 25 の合計 2 進値を読みやすくして印刷または表示したいとします。次のステートメントを使用すると、BI 値を読みやすい ZD 値に変換できます。

```
OPTION COPY
OUTREC FIELDS=(1,4,X,
                5,4,BI,TO=ZD,LENGTH=6,X,
                9,4,BI,TO=ZD,LENGTH=6)
```

p,m,BI,TO=ZD は、BI 値を ZD 値に変換します。デフォルトでは、4 バイトの BI 値は 10 バイトの ZD 値を作成しますが、LENGTH=6 によってデフォルトの長さが指定変更され、6 バイトの ZD 値が作成されます。

表 30 は、読みやすい出力を示しています。

表 30. BI から ZD への変換

Publisher	Number In Stock	Number Sold
1 4	6 11	13 18
COR	000103	000161
FERN	000019	000087
VALD	000042	000097
WETH	000062	000079

あるいは、次の OUTREC ステートメントを使用すると、BI 値を読みやすい FS 値に変換できます。

```
OUTREC FIELDS=(1,4,X,
                5,4,BI,TO=FS,LENGTH=6,X,
                9,4,BI,TO=FS,LENGTH=6)
```

p,m,BI,TO=FS は、BI 値を FS 値に変換します。デフォルトでは、4 バイトの BI 値は 11 バイトの FS 値を作成しますが、LENGTH=6 によってデフォルトの長さが指定変更され、6 バイトの FS 値が作成されます。

表 31 は、読みやすい出力を示しています。

表 31. BI から FS への変換

Publisher	Number In Stock	Number Sold
1 4	6 11	13 18
COR	103	161
FERN	19	87
VALD	42	97
WETH	62	79

p,m,f,TO=f または p,m,f,f を使用すると、さまざまな数値フォーマットを他のさまざまな数値フォーマットに変換できます。LENGTH=n を使用すると、デフォルトの出力長を指定変更できます。

DFSORT の数値変換パラメーターの詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

数値フィールドの編集

PD フォーマットで保管された数値を持つ入力レコードを、解釈しやすいように表示または印刷したいとします。PD 値は、圧縮内部フォーマットで整数として保管されるので、値を印刷または表示した場合は実際には判読不能です。

ただし、以下のことが分かっています。

- 最初の PD フィールドは、小数点以下が 2 桁の数値を表します。
- 2 番目の PD フィールドは、小数点以下が 3 桁の数値を表します。
- 3 番目の PD フィールドは、mmddyyyy 形式の日付を表します。

レコードの再フォーマット

OUTREC ステートメントの編集機能を使用すると、PD 値の数字を読みやすい文字として表示できます。さらに、編集機能を使用して、符号、コンマ、小数点、およびハイフンを適切に挿入することにより、PD 値を解釈しやすくすることができます。

表 32 は、レコード内の関連入力フィールドを示しています。PD フィールドは、表 32 では読みやすい数値として表されていますが、実際には判読不能なフォーマットで保管されています。

表 32. PD 値を持つ入力レコード

CH フィールド	最初の PD フィールド	2 番目の PD フィールド	3 番目の PD フィールド
1 8	11 16	31 37	41 45
WEST	+1524900810	+0000000020000	+05122003
EAST	-0065781053	+0721500532006	+11292003
NORTH	+0000000000	-0000982630735	+02152004
SOUTH	-0000003562	-0003826254999	+12032003

次の OUTREC ステートメントを使用すると、PD 値を分かりやすくすることができます。

```
OUTREC FIELDS=(1,8,
  5X,
  11,6,PD,M4,
  5X,
  31,7,PD,EDIT=(SI,III,III,IIT.TTT),SIGNS=(,-),
  5X,
  41,5,PD,EDIT=(TT-TT-TTTT))
```

M4 は、DFSORT の 27 個の事前定義編集マスク (下記の 57 ページの表 33 を参照) の 1 つです。これは、パターン SI,III,III,III,IIT.TT に従って数値フィールドを編集します。EDIT=(SI,III,III,IIT.TTT) は、ユーザー定義の編集マスクです。これは、パターン SI,III,III,IIT.TTT に従って数値フィールドを編集します。EDIT=(TT-TT-TTTT) は、もう 1 つのユーザー定義編集マスクです。これは、パターン TT-TT-TTTT に従って数値フィールドを編集します。

パターンの中の各文字の意味は、次のとおりです。

- **I** は、1-9 として (または先行 0 の場合はブランクとして) 表示される、先行の非有効数字を示します。
- **T** は、0-9 として表示される有効数字を示します。
- 数字の前の **S** は、先行符号を示します。数字の後の **S** は、末尾符号を示します。M4 の場合、先行符号は、正の値であれば + として、負の値であれば - として表示されます。EDIT=(SI,III,III,IIT.TTT),SIGNS=(,-) の場合、先行符号は、正の値であればブランクとして、負の値であれば - として表示されます。
- その他の文字 (例えば、コンマ、小数点、またはハイフン) は適切に表示されます。

上記の OUTREC ステートメントに対応して作成される結果は、次のようになります。

WEST	+15,249,008.10	20.000	05-12-2003
EAST	-657,810.53	3,721,500,532.006	11-29-2003
NORTH	+0.00	-982,630.735	02-15-2004
SOUTH	-35.62	-3,826,254.999	12-03-2003

事前定義およびユーザー定義の編集マスクにより、出力する数値フィールドのフォーマット方法に関してかなりの柔軟性が与えられます。 p,m,f,Mnn または p,m,f,EDIT=(pattern) を使用すると、さまざまな出力パターンに従ってさまざまな数値フィールドを編集できます。 SIGNS=(a,b,c,d) を使用すると、出力符号 (先行正符号、先行負符号、末尾正符号、および末尾負符号) を指定変更または設定できます。 LENGTH=n を使用すると、出力長を指定変更または設定できます。

DFSORT の数値編集パラメーターの詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。参照しやすいように、表 33 に、27 個の事前定義編集マスク (M0-M26) のパターンと、それぞれのマスクの出力例を示しています。

表 33. 編集マスク・パターン

マスク	パターン	例	
		値	結果
M0	IIIIIIIIIIIIIIIIIIIIITS	+01234	1234
		-00001	1-
M1	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTS	-00123	00123-
		+00123	00123
M2	II,III,III,III,III,III,III,III,IIT.TTS	+123450	1,234.50
		-000020	0.20-
M3	II,III,III,III,III,III,III,III,IIT.TTCR	-001234	12.34CR
		+123456	1,234.56
M4	SII,III,III,III,III,III,III,III,IIT.TT	+0123456	+1,234.56
		-1234567	-12,345.67
M5	SII,III,III,III,III,III,III,III,IIT.TTS	-001234	(12.34)
		+123450	1,234.50
M6	III-TTT-TTTT	00123456	012-3456
		12345678	1-234-56788
M7	TTT-TT-TTTT	00123456	000-12-3456
		12345678	012-34-5678
M8	IT:TT:TT	030553	3:05:53
		121736	12:17:36
M9	IT/TT/TT	123004	12/30/04
		083104	8/31/04
M10	IIIIIIIIIIIIIIIIIIIIIT	01234	1234
		00000	0
M11	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	00010	00010
		01234	01234
M12	SI,III,III,III,III,III,III,III,IIT	+1234567	1,234,567
		-0012345	-12,345

レコードの再フォーマット

表 33. 編集マスク・パターン (続き)

マスク	パターン	例	
		値	結果
M13	SI.III.III.III.III.III.III.III.IIT	+1234567	1.234.567
		-0012345	-12.345
M14	SI III III III III III III III IITS	+1234567	1 234 567
		-0012345	(12 345)
M15	I III III III III III III III IITS	+1234567	1 234 567
		-0012345	12 345-
M16	SI III III III III III III III IIT	+1234567	1 234 567
		-0012345	-12 345
M17	SI'III'III'III'III'III'III'III'III'IIT	+1234567	1'234'567
		-0012345	-12'345
M18	SII,III,III,III,III,III,III,III,IIT.TT	+0123456	1,234.56
		-1234567	-12,345.67
M19	SII.III.III.III.III.III.III.III,IIT,TT	+0123456	1.234,56
		-1234567	-12.345,67
M20	SI III III III III III III III IIT,TTS	+0123456	1 234,56
		-1234567	(12 345,67)
M21	II III III III III III III III IIT,TTS	+0123456	1 234,567
		-1234567	12 345,67-
M22	SI III III III III III III III IIT,TT	+0123456	1 234,56
		-1234567	-12 345,67
M23	SII'III'III'III'III'III'III'III'IIT.TT	+0123456	1'234,56
		-1234567	-12'345,67
M24	SII'III'III'III'III'III'III'III'IIT,TT	+0123456	1'234,56
		-1234567	-12'345,67
M25	SIIIIIIIIIIIIIIIIIIIIIIIIIT	+01234	1234
		-00001	-1
M26	STTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	1234	+01234
		-1	-00001

データを 16 進数で表示する

56 ページの表 32 の CH 値と PD 値が実際にはどのように保管されているか知りたい場合は、次の OUTREC ステートメントを使用すると、これらのフィールドの 16 進表記を表示することができます。

```
OUTREC FIELDS=(1,8,HEX,
                2X,
                11,6,HEX,
                2X,
                31,7,HEX,
                2X,
                41,5,HEX)
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
E6C5E2E340404040 01524900810C 0000000020000C 005122003C
C5C1E2E340404040 00065781053D 3721500532006C 011292003C
D5D6D9E3C8404040 00000000000C 0000982630735D 002152004C
E2D6E4E3C8404040 000000003562D 0003826254999D 012032003C
```

数値フィールドおよび定数に算術演算を実行する

56 ページの表 32 の最初の PD フィールドから 56 ページの表 32 の 2 番目の PD フィールドを減算した結果を表示したいとします。前述したように、PD フィールドは整数として保管されていますが、以下のことが分かっています。

- 最初の PD フィールドは 6 バイトであり、小数点以下が 2 桁の、11 桁の符号付き数値を表します。
- 2 番目の PD フィールドは 7 バイトであり、小数点以下が 3 桁の、13 桁の符号付き数値を表します。

これらのフィールドに算術演算を実行するには、数値の整数と小数部分が正確に一致していることを確認する必要があります。例えば、最初のレコードでは、最初の PD 値は +15249008.10 を表す +1524900810 であり、2 番目の PD 値は +20.000 を表す +20000 です。最初の PD 値から 2 番目の PD 値を減算するためには、それらを次のような形式に置く必要があります。

```
+15249008.100
+000000020.000
```

次の OUTREC ステートメントを使用すると、減算を正しく処理できます。

```
OUTREC FIELDS=(1,8,
5X,
((11,6,PD,MUL,+10),SUB,31,7,PD),
EDIT=(SI,III,III,IIT.TTT),SIGNS=(,-))
```

11,6,PD,MUL,+10 は、小数点以下の桁を一致させるために最初の PD フィールドを 10 倍します。SUB,31,7,PD は、11,6,PD,MUL,+10 演算の結果から 2 番目の PD フィールドを減算します。

EDIT と SIGNS は、結果をさらに分かりやすくするために使用されます。

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
WEST          15,248,988.100
EAST          -3,722,158,342.536
NORTH         982,630.735
SOUTH         3,826,219.379
```

演算子 MIN (最小)、MAX (最大)、DIV (除算)、MUL (乗算)、MOD (剰余)、ADD (加算)、および SUB (減算) を使用して、数値フィールドおよび 10 進定数 (+n および -n) に算術演算を実行することができます。各演算子の計算の優先順位は次のとおりですが、括弧を使用すれば変更できます。

1. MIN および MAX
2. MUL、DIV、および MOD
3. ADD および SUB

レコードの再フォーマット

p,m,f、MIN、MAX、MUL、DIV、MOD、ADD、SUB、+n、-n、および括弧を使用することにより、さまざまな数値フォーマットに対して広範囲にわたる算術演算を実行できます。算術演算の結果は 15 桁の ZD 値であり、前述したように、この値を異なる数値フォーマットに変換したり、事前定義編集マスク (M0-M26) またはユーザー定義編集マスクを使用して編集したりすることができます。

算術演算の詳細については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

ルックアップおよび変更

以下のステートメントを使用すると、すべての学部コースと各学部で使用する本の出版社を表示できます。

```
SORT FIELDS=(110,5,CH,A,106,4,CH,A)
SUM FIELDS=NONE
OUTREC FIELDS=(110,5,X,106,4)
```

これらのステートメントに対応して作成される結果は、次のようになります。

```
      COR
BIOL  VALD
BUSIN  COR
BUSIN  VALD
COMP  FERN
COMP  VALD
COMP  WETH
ENGL  COR
ENGL  FERN
ENGL  VALD
HIST  COR
HIST  WETH
PSYCH  COR
PSYCH  FERN
```

このリストは、すべて短縮されているので、かなりあいまいです。次の OUTREC ステートメントを使用して、省略形を記述ワードで置き換えるようにすると、リストをもっと分かりやすくすることができます。

```
OUTREC FIELDS=(110,5,CHANGE=(16,
      C'HIST',C'History',
      C'BUSIN',C'Business',
      C'COMP',C'Computer Science',
      C'ENGL',C'English',
      C'BIOL',C'Biology',
      C'PSYCH',C'Psychology'),
      NOMATCH=(C'Unaffiliated'),
      C'|',
      106,4,CHANGE=(20,
      C'FERN',C'Fernal Brothers',
      C'CORN',C'Cornish Limited',
      C'VALD',C'Valdern and Co.',
      C'WETH',C'Wethman, Inc.'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

Unaffiliated	Cornish Limited
Biology	Valdern and Co.
Business	Cornish Limited
Business	Valdern and Co.
Computer Science	Fernal Brothers

Computer Science	Valdern and Co.
Computer Science	Wethman, Inc.
English	Cornish Limited
English	Fernald Brothers
English	Valdern and Co.
History	Cornish Limited
History	Wethman, Inc.
Psychology	Cornish Limited
Psychology	Fernald Brothers

テーブル・ルックアップおよび変更を使用すると、文字、16 進数、またはビット定数のテーブルから入力フィールド値を検出し、出力レコード内で対応する文字定数、16 進定数、または入力フィールドを設定できます。

`p, m, CHANGE=(v, find, set, ...)` は、入力フィールドの位置と長さ、出力フィールドの長さ、および検出定数と、設定定数または設定フィールドのペアからなるテーブルを与えます。例えば、上記の `OUTREC` ステートメントの最初のフィールドの場合は、次のようになっています。

- 入力フィールドは桁 110 から始まります。
- 入力フィールドおよび検出定数の長さは 5 です。
- 出力フィールドおよび設定定数の長さは 16 です。
- 最初の検出定数は 'HIST' です (5 文字になるまで終わりにブランクが埋め込まれます)。
- 最初の設定定数は 'History' です (16 文字になるまで終わりにブランクが埋め込まれます)。

学部コースに 'HIST' が含まれていると、出力フィールドはその都度 'History' に設定されます。学部コースに 'BUSIN' が含まれていると、出力フィールドはその都度 'Business' に設定され、というようになります。

`NOMATCH=(set)` を使用すると、入力フィールド値がどの検出定数とも一致しないときの出力値を設定できます。 `NOMATCH=(C'string')` は、文字ストリング定数を設定します。 `NOMATCH=(X'string')` は 16 進数ストリング定数を設定します。 `NOMATCH=(q,n)` は、入力フィールド値を設定します。例えば、上記の `OUTREC` ステートメントでは、入力フィールド値がブランクであると、出力フィールド値は 'Unaffiliated' となります (16 文字になるまで終わりにブランクが埋め込まれます)。入力フィールド値がどの検出定数とも一致しない場合に、`NOMATCH=(set)` が指定されていないと、`DFSORT` はエラー・メッセージを出して終了します。

ルックアップおよび変更の詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

データの左寄せと右寄せ

次のような入力レコードがあるとします。

```

      History
Psychology
                Business
      Biology
      Computer Science

```

この表示は乱雑なので、次のステートメントを使用してデータを左寄せすれば、表示の体裁を整えることができます。

レコードの再フォーマット

```
OPTION COPY
OUTREC FIELDS=(1,30,JFY=(SHIFT=LEFT))
```

SHIFT=LEFT は、左寄せすることを指定します。この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
History
Psychology
Business
Biology
Computer Science
```

あるいは、次のステートメントを使用すると、データを右寄せすることもできます。

```
OPTION COPY
OUTREC FIELDS=(1,30,JFY=(SHIFT=RIGHT))
```

SHIFT=RIGHT は、右寄せすることを指定します。この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
      History
    Psychology
      Business
        Biology
    Computer Science
```

また、DFSORT の位置調整機能では、先行ストリング、末尾のストリング、またはその両方をデータに追加することもできます。例えば、次のステートメントを使用すると、左寄せしたデータの前後に「< 」と「>」を付けることができます。

```
OPTION COPY
OUTREC FIELDS=(1,30,JFY=(SHIFT=LEFT,LEAD=C'<',TRAIL=C'>'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<History>
<Psychology>
<Business>
<Biology>
<Computer Science>
```

LEAD=string は、先行ストリングを文字定数または 16 進定数 (1 から 50 バイト) として指定します。TRAIL=string は、末尾のストリングを文字定数または 16 進定数 (1 から 50 バイト) として指定します。

左寄せを行うと、末尾のストリングは、データ内の最後の非ブランク文字のすぐ右側に置かれることに注意してください。

左寄せではなく、右寄せを行う場合は、次のステートメントを使用できます。

```
OPTION COPY
OUTREC FIELDS=(1,30,JFY=(SHIFT=RIGHT,LEAD=C'<',TRAIL=C'>'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```

        <History>
    <Psychology>
        <Business>
            <Biology>
    <Computer Science>

```

右寄せを行うと、先行ストリングは、データ内の最初の非ブランク文字のすぐ左側に置かれることに注意してください。

先行ストリングまたは末尾のストリングを付けたことにより、出力フィールドが入力フィールドより長くなる場合、文字が失われます。これを避けるには、LENGTH パラメーターを使用して出力フィールドの長さを増やすことができます。例えば、次のような入力レコードがあるとします。

```

    rats
    bats
    cats

```

さらに、次の制御ステートメントを使用して先行ストリングを追加したとします。

```

OPTION COPY
OUTREC FIELDS=(1,7,JFY=(SHIFT=LEFT,LEAD=C'*I love ',TRAIL=C'*'))

```

入力フィールドは 7 バイトであるので、デフォルトでは出力フィールドも 7 バイトです。したがって、出力フィールドは次のように切り捨てられます。

```

*I love
*I love
*I love

```

切り捨てを回避するために、LENGTH=16 を使用して、先行ストリングと末尾ストリング用に追加した 9 文字分、出力フィールドの長さを増やすことができます。

```

OPTION COPY
OUTREC FIELDS=(1,7,JFY=(SHIFT=LEFT,LEAD=C'*I love ',TRAIL=C'*',
    LENGTH=16))

```

出力フィールドが長くなると、先行ストリングと末尾ストリングを収容することができます。したがって、次のような出力が得られます。

```

*I love rats*
*I love bats*
*I love cats*

```

また、位置調整機能を使用して、ブランク以外の先行文字および末尾の文字をデータから除去することもできます。次のような入力レコードがあるとします。

```

        (History)
    (Psychology)
            (Business)
        (Biology)
    (Computer Science)

```

次のステートメントを使用すると、先行する「(」文字と末尾の「)」文字を除去した後、データを左寄せすることができます。

```

OPTION COPY
OUTREC FIELDS=(1,30,JFY=(SHIFT=LEFT,PREBLANK=C'()'))

```

PREBLANK=list は、DFSORT がデータの位置調整を開始する前にブランクで置き換える文字のリストを指定します。このリストは、文字定数または 16 進定数 (1 から 10 バイト) として指定できます。リスト内の各文字は、他の文字とは無関係

レコードの再フォーマット

であることに注意してください。例えば、`PREBLANK=C'*/'` は、位置調整処理が開始する前に、各先行または末尾の「*」文字および「/」文字をブランクで置き換えます (例えば、先行文字と末尾文字のシーケンス `/*、//*`、`*/、//`、および `*` はすべて、ブランクで置き換えられます。

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
History
Psychology
Business
Biology
Computer Science
```

次のステートメントを使用すると、データを右寄せし、先行の「(」文字と末尾の「)」文字を先行の「<」文字と末尾の「>」文字で置き換えることができます。

```
OPTION COPY
OUTREC FIELDS=(1,30,JFY=(SHIFT=RIGHT,PREBLANK=C'(',')',
    LEAD=C'<','>',TRAIL=C'>'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<History>
<Psychology>
<Business>
<Biology>
<Computer Science>
```

位置調整の詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

データの左/右詰め

次のような入力レコードがあるとします。

```
<tag> History </tag>
<tag> Psychology </tag>
<tag> Business </tag>
<tag>Biology</tag>
<tag> Science </tag>
```

空白 (ブランク) を除去したい場合は、次のステートメントを使用してデータを左詰めすることができます。

```
OPTION COPY
OUTREC FIELDS=(1,40,SQZ=(SHIFT=LEFT))
```

`SHIFT=LEFT` は、すべてのブランクを除去し、残りの文字を左にシフトさせ、必要に応じて右側にブランクを埋め込むことによって、左詰めすることを指定します。この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<tag>History</tag>
<tag>Psychology</tag>
<tag>Business</tag>
<tag>Biology</tag>
<tag>Science</tag>
```

あるいは、次のステートメントを使用してデータを右詰めすることもできます。


```
OPTION COPY
OUTREC FIELDS=(1,40,SQZ=(SHIFT=RIGHT))
```

SHIFT=RIGHT は、すべてのブランクを除去し、残りの文字を右にシフトさせ、必要に応じて左側にブランクを埋め込むことによって、右詰めすることを指定します。この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<tag>History</tag>
<tag>Psychology</tag>
<tag>Business</tag>
<tag>Biology</tag>
<tag>Science</tag>
```

また、DFSORT の squeeze 機能では、先行ストリング、末尾のストリング、またはその両方をデータに追加することもできます。例えば、次のステートメントを使用すると、左詰めされたデータの前後に「<tag1>」と「</tag1>」を付けることができます。

```
OPTION COPY
OUTREC FIELDS=(1,40,SQZ=(SHIFT=LEFT,LEAD=C'<tag1>',
TRAIL=C'</tag1>'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<tag1><tag>History</tag></tag1>
<tag1><tag>Psychology</tag></tag1>
<tag1><tag>Business</tag></tag1>
<tag1><tag>Biology</tag></tag1>
<tag1><tag>Science</tag></tag1>
```

LEAD=string は、先行ストリングを文字定数または 16 進定数 (1 から 50 バイト) として指定します。TRAIL=string は、末尾のストリングを文字定数または 16 進定数 (1 から 50 バイト) として指定します。

左詰めを行うと、末尾のストリングは、データ内の最後の非ブランク文字のすぐ右側に置かれることに注意してください。

左詰めではなく、右詰めを行う場合は、次のステートメントを使用できます。

```
OPTION COPY
OUTREC FIELDS=(1,40,SQZ=(SHIFT=RIGHT,LEAD=C'<tag1>',
TRAIL=C'</tag1>'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<tag1><tag>History</tag></tag1>
<tag1><tag>Psychology</tag></tag1>
<tag1><tag>Business</tag></tag1>
<tag1><tag>Biology</tag></tag1>
<tag1><tag>Science</tag></tag1>
```

右詰めを行うと、先行ストリングは、データ内の最初の非ブランク文字のすぐ左側に置かれることに注意してください。

先行ストリングまたは末尾のストリングを付けたことにより、出力フィールドが入力フィールドより長くなる場合、文字が失われます。これを避けるには、LENGTH

レコードの再フォーマット

パラメーターを使用して出力フィールドの長さを増やすことができます (上記の 61 ページの『データの左寄せと右寄せ』を参照)。

また、squeeze 機能を使用して、ブランク以外の文字をデータから除去することもできます。次のような入力レコードがあるとします。

```
<tag> (History) </tag>
  <tag> (Psychology) </tag>
    <tag> (Business) </tag>
<tag>(Biology)</tag>
  <tag> (Science) </tag>
```

次のステートメントを使用すると、「(」および「)」文字を除去した後、データを左詰めすることができます。

```
OPTION COPY
OUTREC FIELDS=(1,40,SQZ=(SHIFT=LEFT,PREBLANK=C'(')'))
```

PREBLANK=list は、DFSORT がデータの squeeze を開始する前にブランクで置き換える文字のリストを指定します。このリストは、文字定数または 16 進定数 (1 から 10 バイト) として指定できます。リスト内の各文字は、他の文字とは無関係であることに注意してください。例えば、PREBLANK=C'*/' は、squeeze 処理を開始する前に、各「*」文字および「/」文字をブランクで置き換えます (例えば、文字シーケンス /*、//*, */、//、および * はすべて、ブランクで置き換えられます)。

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
<tag>History</tag>
<tag>Psychology</tag>
<tag>Business</tag>
<tag>Biology</tag>
<tag>Science</tag>
```

次のステートメントを使用すると、データを右詰めし、「(」文字と「)」文字を除去することができます。

```
OPTION COPY
OUTREC FIELDS=(1,40,SQZ=(SHIFT=RIGHT,PREBLANK=C'(')'))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
      <tag>History</tag>
    <tag>Psychology</tag>
      <tag>Business</tag>
      <tag>Biology</tag>
      <tag>Science</tag>
```

squeeze 機能を使用すると、最初の非ブランクと最後の非ブランクとの間のブランクのグループを、他の文字で置き換えることができます。次のような入力レコードがあるとします。

Manufacturing	California	+100000
Research	Arizona	+50000
Marketing	Texas	+75000

次のステートメントを使用すると、コンマ区切り可変レコードを作成することができます。

```
OPTION COPY
OUTREC FIELDS=(1,80,SQZ=(SHIFT=LEFT,MID=C','))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
Manufacturing,California,+100000
Research,Arizona,+50000
Marketing,Texas,+75000
```

MID=string は、除去されたブランクまたは PREBLANK 文字を文字定数または 16 進定数 (1 から 10 バイト) として置き換えるためのストリングを指定します。

引用符ペアの間にあるブランクおよび PREBLANK 文字を DFSORT が「無視」するようにしたい場合は、SQZ と一緒に PAIR=QUOTE を使用できます。

次のような入力レコードがあるとします。

```
"Computer Science A+"      +123
"Ancient Civilization B-"  +521
"Sanskrit A-"              -263
```

次のステートメントを使用すると、引用符ペアの外部にあるブランク、+ 符号、および - 符号を除去するのに対して、引用符ペアの内部にあるブランク、+ 符号、および - 符号を「保護」し、詰められた 2 つのフィールド間に 1 つのブランクのみを残すことができます。

```
OPTION COPY
OUTREC FIELDS=(1,80,SQZ=(SHIFT=LEFT,PAIR=QUOTE,
PREBLANK=C'+-',MID=C' '))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

```
"Computer Science A+" 123
"Ancient Civilization B-" 521
"Sanskrit A-" 263
```

アポストロフィ・ペアの間にあるブランクおよび PREBLANK 文字を DFSORT が「無視」するようにしたい場合は、SQZ と一緒に PAIR=APOST を使用できます。

次のような入力レコードがあるとします。

```
'Computer Science A+'      +123
'Ancient Civilization B-'  +521
'Sanskrit A-'              -263
```

次のステートメントを使用すると、アポストロフィのペアの外部にあるブランク、+ 符号、および - 符号を除去するのに対して、アポストロフィのペアの内部にあるブランク、+ 符号、および - 符号を「保護」し、詰められた 2 つのフィールド間に 1 つのブランクのみを残すことができます。

```
OPTION COPY
OUTREC FIELDS=(1,80,SQZ=(SHIFT=LEFT,PAIR=APOST,
PREBLANK=C'+-',MID=C' '))
```

この OUTREC ステートメントに対応して作成される結果は、次のようになります。

レコードの再フォーマット

```
'Computer Science A+' 123
'Ancient Civilization B-' 521
'Sanskrit A-' 263
```

squeeze の詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

これまで

ここまで、入力レコードをさまざまな方法で再フォーマットして出力するために、OUTREC ステートメントの BUILD または FIELDS パラメーターで使用可能な諸機能の使用法を学習しました。これらの再フォーマット機能はすべて、OUTREC ステートメントだけでなく、INREC ステートメントの BUILD または FIELDS パラメーターでも、また OUTFIL ステートメントの BUILD または OUTREC パラメーターでも使用できることを覚えておいてください。事実上、該当する場合には、これらのステートメントをすべて使用できます。次に、1 つ以上のフィールドを変更し、かつレコードの残りの部分に影響を与えないようにするために、OUTREC ステートメントの OVERLAY パラメーターを使用する方法を学習します。

OVERLAY を使用したレコードの再フォーマット

OUTREC ステートメントの BUILD または FIELDS パラメーターでは、再フォーマットされた出力レコードを一度に 1 項目ずつ作成します。出力レコードに表示したい各項目（未編集、編集済み、または変換済みの入力フィールド、ブランク、ストリングなど）を、表示したい順に指定する必要があります。

しかし、1 個または数個の項目のみを変更し、レコードの残りの部分は元の形式のままにしたい場合は、より簡単な方法があります。OUTREC ステートメントの OVERLAY パラメーターを使用する方法です。BUILD または FIELDS パラメーターについて上記で記述したのと同じ項目を、OVERLAY パラメーターで使用できます。

学部コース (course department) を小文字に変更し、それぞれの本の価格を 10% 割引したいものとします。次のステートメントを使用すると、各出力レコードで、旧価格ではなく新価格を持つ出力データ・セットを作成できます。

```
OPTION COPY
OUTREC BUILD=(1,109,
               110:110,5,TRAN=UTOL,
               115:115,55,
               170:170,4,BI,SUB,(170,4,BI,DIV,+10),
               TO=BI,LENGTH=4)
```

OUTREC パラメーター内の演算式により、次のように新しい 4 バイト BI 価格が得られます。

$$\text{new price} = \text{old price} - (\text{old price} / 10)$$

BUILD パラメーターでは、レコード全体を作成する必要があります。したがって、次のものを指定する必要があります。

- 1,109。学部コースの前のバイトを保持します。

- 110,5,TRAN=UTOL。学部コースを大文字から小文字に変更します。
- 115,55。学部コースと価格との間のバイトを保持します。
- 170:170,4,BI,SUB,(170,4,BI,DIV,+10),TO=BI,LENGTH=4。price を price=price-(price/10) に置き換えます。

しかし、学部コースの前のバイト、または学部コースと価格との間のバイトは変更しないので、次のように OVERLAY パラメーターを使用して、変更したいバイトのみをオーバーレイする方が簡単です。

```
OPTION COPY
OUTREC OVERLAY=(110:110,5,TRAN=UTOL,
170:170,4,BI,SUB,(170,4,BI,DIV,+10),
TO=BI,LENGTH=4)
```

表 34 は、コピー済みデータ・セットを示しています。

表 34. 学部コースと価格を変更した本

Book	Title	Department	Price
1	75	110 114	170 173
	COMPUTER LANGUAGES	comp	2340
	LIVING WELL ON A SMALL BUDGET		8910
	SUPPLYING THE DEMAND	busin	1733
	VIDEO GAME DESIGN	comp	1980
	INKLINGS: AN ANTHOLOGY OF YOUNG POETS	engl	0536
	COMPUTERS: AN INTRODUCTION	comp	1710
	PICK'S POCKET DICTIONARY		0266
	EDITING SOFTWARE MANUALS	engl	1305
	NUMBERING SYSTEMS	comp	0324
	STRATEGIC MARKETING	busin	2115
	THE INDUSTRIAL REVOLUTION	hist	0716
	MODERN ANTHOLOGY OF WOMEN POETS	engl	0405
	INTRODUCTION TO PSYCHOLOGY	psych	1980
	THE COMPLETE PROOFREADER	engl	0563
	SYSTEM PROGRAMMING	comp	2876
	SHORT STORIES AND TALL TALES	engl	1368
	INTRODUCTION TO BIOLOGY	biol	2115
	ADVANCED TOPICS IN PSYCHOANALYSIS	psych	2340
	EIGHTEENTH CENTURY EUROPE	hist	1611
	CRISES OF THE MIDDLE AGES	hist	1080

OVERLAY パラメーターでは、1,109 または 115,55 を計算または指定する必要はありません。その代わりに、各出力の開始桁 (110: および 170:) とその置換項目だけを指定します。オーバーレイしない桁はすべてそのままです。また、BUILD 項目とは異なり、OVERLAY 項目は互いにオーバーラップし、任意の順序で指定できます。この例では、次の OUTREC ステートメントの結果は、直前の OUTREC ステートメントの結果と同じです。

```
OPTION COPY
OUTREC OVERLAY=(170:170,4,BI,SUB,(170,4,BI,DIV,+10),
TO=BI,LENGTH=4,
110:110,5,TRAN=UTOL)
```

OVERLAY を使用したレコードの拡張

また、OVERLAY パラメーターを使用して、出力レコードの残りの部分に影響を与えることなく、レコードを拡張することもできます。

旧価格を、10% 割引した新価格に置き換えたいものとします。ただし、元の価格を含む新フィールドでレコードを拡張するものとします。以下のステートメントを使用できます。

```
OPTION COPY
OUTREC OVERLAY=(174:170,4,
  170:170,4,BI,SUB,(170,4,BI,DIV,+10),
  TO=BI,LENGTH=4)
```

表 35 は、コピー済みデータ・セットを示しています。

表 35. 新価格と旧価格

新価格	旧価格
170 173	174 177
2340	2600
8910	9900
1733	1925
1980	2199
0536	0595
1710	1899
0266	0295
1305	1450
0324	0360
2115	2350
0716	0795
0405	0450
1980	2200
0563	0625
2876	3195
1368	1520
2115	2350
2340	2600
1611	1790
1080	1200

位置 174-177 で新規フィールドを追加したので、DFSORT は自動的にレコードの長さを 173 バイトから 177 バイトに増やします。

この場合、両方の項目に旧価格を使用するので、2 つのオーバーレイ項目を指定する順序が重要です。したがって、旧価格を変更する前に旧価格のコピーを作成する必要があります。上記の OUTREC ステートメントでは、最初のオーバーレイ項目は旧価格を新しいフィールドにコピーし、2 番目のオーバーレイ項目は旧価格を新価格で置き換えます。次の OUTREC ステートメントを使用するとします。

```
OUTREC OVERLAY=(170:170,4,BI,SUB,(170,4,BI,DIV,+10),
  TO=BI,LENGTH=4,
  174:170,4)
```

この場合、不注意により旧価格を新価格で置き換えてから、新価格を新規フィールドにコピーしてしまう場合があります。したがって、一方のフィールドに新価格があり、もう一方のフィールドに旧価格があるのではなく、両方のフィールドに新価格があります。OVERLAY では項目のオーバーラップが可能であるので、後のオーバーレイ項目で使いたいフィールドを、前のオーバーレイ項目で破棄しないように注意する必要があります。

これまで

ここまで、OUTREC ステートメントの OVERLAY パラメーターを使用して、指定された桁を再フォーマット項目でオーバーレイし、かつ再フォーマット済みレコードの残りの部分に影響を与えないようにする方法を学習しました。これらの再フォーマット機能はすべて、OUTREC ステートメントだけでなく、INREC ステートメントと OUTFIL ステートメントの OVERLAY パラメーターでも使用できることを覚えておいてください。次に、OUTREC ステートメントで IFTHEN 文節を使用して、レコードごとに異なる方法で再フォーマットする方法を学習します。

IFTHEN を使用したレコードの再フォーマット

上記のセクションで説明した OVERLAY パラメーターと、BUILD または FIELDS パラメーターでは、すべての出力レコードに同じ再フォーマット項目を使用できます。OUTREC ステートメントの IFTHEN 文節では、出力レコードのサブセットを選択し、それらのサブセットに異なる BUILD または OVERLAY 項目を適用することができます。したがって、「type 1」レコードに 1 組の BUILD 項目を適用し、「type 2」レコードに 1 組の OVERLAY 項目を適用し、「type 3」レコードに別の組の BUILD 項目を適用し、それ以外のタイプのレコードに別の組の OVERLAY 項目を適用する（または項目を適用しない）ことが可能です。

次のように 4 つのタイプの IFTHEN 文節を使用できます。

- **WHEN=INIT:** 1 つ以上の WHEN=INIT 文節を使用して、すべての入力レコードに BUILD または OVERLAY 項目を適用します。WHEN=INIT 文節は、その他の IFTHEN 文節より前に処理されます。
- **WHEN=(logexp):** 1 つ以上の WHEN=(logexp) 文節を使用して、指定の論理式を満たすレコードのサブセットに BUILD または OVERLAY 項目を適用します。上記の 29 ページの『第 3 章 レコードの組み込みまたは省略』で説明した INCLUDE ステートメントの COND=(logexp) パラメーターに使用できる任意の論理式を WHEN=(logexp) に使用できます。WHEN=(logexp) 文節が満たされるのは、論理式が真と評価される場合です。
- **WHEN=ANY:** 複数の WHEN=(logexp) 文節の後で WHEN=ANY 文節を使用して、追加の BUILD または OVERLAY 項目をレコードに適用します（先行する WHEN=(logexp) 文節の基準を満たした場合）。
- **WHEN=NONE:** 1 つ以上の WHEN=NONE 文節を使用して、WHEN=(logexp) 文節の基準を満たしていないレコードに、BUILD または OVERLAY 項目を適用します。WHEN=NONE 文節は、他の IFTHEN 文節の後に処理されます。

レコードの再フォーマット

WHEN=NONE 文節を指定しない場合、WHEN=(logexp) 文節の基準を満たしていない入力レコードには、WHEN=INIT の変更 (ある場合) のみが適用されます。

以下の場合に、タイトル、出版社、編集済み価格、および割引 % を示すレポートを作成したいものとします。

- 出版社 COR の本のうち、\$20.00 を超える本の価格を 20% 割引する
- 出版社 COR の本のうち、\$20.00 以下の本の価格を 10% 割引する
- 出版社 VALD のすべての本の価格を 25% 割引する
- その他の出版社のすべての本の価格を 15% 割引する

以下のステートメントを使用できます。

```
OPTION COPY
OUTREC IFTHEN=(WHEN=INIT,
  BUILD=(1:1,40,50:106,4,60:170,4,BI,EDIT=(TT.TT),70:170,4)),
  IFTHEN=(WHEN=(50,4,CH,EQ,C'COR',AND,70,4,BI,GT,+2000),
    OVERLAY=(70:C'20%',X)),
  IFTHEN=(WHEN=(50,4,CH,EQ,C'COR',AND,70,4,BI,LE,+2000),
    OVERLAY=(70:C'10%',X)),
  IFTHEN=(WHEN=(50,4,CH,EQ,C'VALD'),OVERLAY=(70:C'25%',X)),
  IFTHEN=(WHEN=NONE,OVERLAY=(70:C'15%',X))
```

表 36 は、コピー済みデータ・セットを示しています。

表 36. 提案された本の割引

Book Title	Publisher	Price	Discount
1 40	50 53	60 64	70 73
COMPUTER LANGUAGES	FERN	26.00	15%
LIVING WELL ON A SMALL BUDGET	COR	99.00	20%
SUPPLYING THE DEMAND	COR	19.25	10%
VIDEO GAME DESIGN	VALD	21.99	25%
INKLINGS: AN ANTHOLOGY OF YOUNG POETS	COR	05.95	10%
COMPUTERS: AN INTRODUCTION	WETH	18.99	15%
PICK'S POCKET DICTIONARY	COR	02.95	10%
EDITING SOFTWARE MANUALS	VALD	14.50	25%
NUMBERING SYSTEMS	FERN	03.60	15%
STRATEGIC MARKETING	VALD	23.50	25%
THE INDUSTRIAL REVOLUTION	WETH	07.95	15%
MODERN ANTHOLOGY OF WOMEN POETS	COR	04.50	10%
INTRODUCTION TO PSYCHOLOGY	COR	22.00	20%
THE COMPLETE PROOFREADER	FERN	06.25	15%
SYSTEM PROGRAMMING	WETH	31.95	15%
SHORT STORIES AND TALL TALES	VALD	15.20	25%
INTRODUCTION TO BIOLOGY	VALD	23.50	25%
ADVANCED TOPICS IN PSYCHOANALYSIS	FERN	26.00	15%
EIGHTEENTH CENTURY EUROPE	WETH	17.90	15%
CRISES OF THE MIDDLE AGES	COR	12.00	10%

最初の IFTHEN 文節は、BUILD を使用してタイトル、出版社、編集済み価格、および割引のプレースホルダーですべてのレコードを初期設定する WHEN=INIT 文節です。出版社 (Publisher) フィールドは位置 50-53 にコピーされ、4 バイトの BI 価

格フィールドは位置 70-73 にコピーされることに注意してください。後続の IFTHEN 文節は、すべてのレコードに対して処理され、新しい位置にあるこれらのフィールドを参照する必要があります。

2 番目の IFTHEN 文節は、OVERLAY を使用する WHEN=(logexp) 文節です。出版社を「COR」、価格を 2000 より大とするレコードのサブセットに対して、割引フィールドを「20%」に設定します。この論理式を満たすレコードについては、後続の IFTHEN 文節は処理されません。この論理式を満たさないレコードに対して、後続の IFTHEN 文節が処理されます。

3 番目の IFTHEN 文節は、OVERLAY を使用する WHEN=(logexp) 文節です。出版社を「COR」、価格を 2000 以下とするレコードのサブセットに対して、割引フィールドを「10%」に設定します。この論理式を満たすレコードについては、後続の IFTHEN 文節は処理されません。この論理式を満たさないレコードに対して、後続の IFTHEN 文節が処理されます。

4 番目の IFTHEN 文節は、OVERLAY を使用する WHEN=(logexp) 文節です。出版社を「VALD」とするレコードのサブセットに対して、割引フィールドを「25%」に設定します。この論理式を満たすレコードについては、後続の IFTHEN 文節は処理されません。この論理式を満たさないレコードに対して、後続の IFTHEN 文節が処理されます。

5 番目の IFTHEN 文節は、OVERLAY を使用する WHEN=NONE 文節です。2 番目、3 番目、および 4 番目の IFTHEN 文節の論理式を満たしていないレコードのサブセットに対して、割引フィールドを「15%」に設定します。

DFSORT は、指定される IFTHEN 文節から適切な再フォーマット済み出力レコード長を判別します。ただし、IFOUTLEN=n パラメーターを使用すると、再フォーマット済みレコードに使用する長さを DFSORT に指示することができます。例えば、80 バイトの入力レコードがあり、次の DFSORT ステートメントを使用するとします。

```
OUTREC IFTHEN=(WHEN=INIT,OVERLAY=(81:SEQNUM,8,ZD)),
          IFTHEN=(WHEN=(81,8,ZD,EQ,+5),OVERLAY=(21:C'J82')),
          IFTHEN=(WHEN=(81,8,ZD,EQ,+10),OVERLAY=(21:C'M72'))
```

デフォルトで、DFSORT は、追加した 8 バイト ZD シーケンス番号を収容するために、再フォーマット済み出力レコード長を 88 にします。8 バイト・シーケンス番号を WHEN 比較に使用するだけで、再フォーマット済み出力レコードの一部としてシーケンス番号を必要としない場合、次のステートメントを使用すれば、再フォーマット済み出力レコード長に 88 ではなく、80 を使用するように DFSORT に指示することができます。

```
OUTREC IFTHEN=(WHEN=INIT,OVERLAY=(81:SEQNUM,8,ZD)),
          IFTHEN=(WHEN=(81,8,ZD,EQ,+5),OVERLAY=(21:C'J82')),
          IFTHEN=(WHEN=(81,8,ZD,EQ,+10),OVERLAY=(21:C'M72')),
          IFOUTLEN=80
```

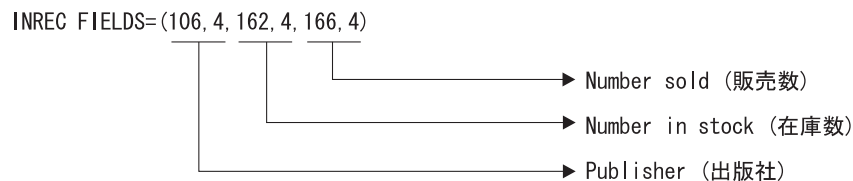
IFTHEN 文節と IFOUTLEN の詳しい使用法については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

これまで

ここまで、OUTREC ステートメントで IFTHEN 文節を使用して、レコードごとに異なる方法で再フォーマットする方法を学習しました。IFTHEN 文節は、OUTREC ステートメントで使えるだけでなく、INREC ステートメントおよび OUTFIL ステートメントでも使用できることを覚えておいてください。次に、INREC ステートメントを使用する上での特別な考慮事項について学習します。

ソート前にレコードを再フォーマットする

INREC ステートメントのフォーマットは、OUTREC ステートメントのフォーマットと同じです。したがって、46 ページの『BUILD または FIELDS を使用したソート後のレコードの再フォーマット』の最初の例で、出版社 (Publisher)、在庫数 (Number In Stock)、および販売数 (Number Sold) フィールドだけを書き込むために OUTREC を使用したところで、代わりに INREC を使用できます。



INREC と一緒に他のステートメントを使用する

INREC はソート前にレコードを再フォーマットするので、SORT および SUM ステートメントでは、出力データ・セットに現れるとおりに再フォーマット済みレコードを参照する必要があります。

したがって、INREC の後では、前の節の制御ステートメントの入力レコードは 12 バイトの長さになります (例については、47 ページの表 25 を参照)。

次のように、再フォーマット済みレコードのバイト位置を処理するように SORT および SUM ステートメントを作成します。

- * Move input positions 106-109 to reformatted record positions 1-4.
 - * Move input positions 162-169 to reformatted record positions 5-12.
- ```
INREC FIELDS=(106,4,162,8)
```
- \* Sort reformatted record positions 1-4.
- ```
SORT FIELDS=(1,4,CH,A)
```
- * Sum reformatted record positions 5-8 and 9-12.
- ```
SUM FIELDS=(5,4,BI,9,4,BI)
```

表 37 は、結果を示しています。

表 37. INREC を使用した場合の出版社 (Publisher)、在庫数 (Number in Stock)、および販売数 (Number Sold) だけの書き込み

| Publisher | Number<br>In Stock | Number<br>Sold |
|-----------|--------------------|----------------|
| 1 4       | 5 8                | 9 12           |
| COR       | 103                | 161            |
| FERN      | 19                 | 87             |
| VALD      | 42                 | 97             |
| WETH      | 62                 | 79             |

193 ページの『付録 C. 制御ステートメントの処理順序』のフローチャートで示されたように、DFSORT は、SORT、SUM、および OUTREC の前、ただし INCLUDE および OMIT の後に、INREC ステートメントを処理します。したがって、INREC ステートメントと一緒に使用された場合、SORT、SUM、および OUTREC では再フォーマット済み レコードを参照する必要があります、INCLUDE および OMIT では元の レコードを参照する必要があります。

DFSORT は、INREC および OUTREC ステートメントの後に、OUTFIL ステートメントを処理します。したがって、OUTFIL では、OUTREC が指定された場合は OUTREC で作成された再フォーマット済みレコード、または OUTREC が指定されずに INREC が指定された場合は INREC で作成された再フォーマット済みレコードを参照する必要があります。OUTFIL ステートメントについては、89 ページの『第 7 章 複数の出力データ・セットおよび報告書の作成』で学習します。

以下の制御ステートメントは、INREC および OUTREC がその他のステートメントにおける指定位置にどのような影響を与えるかを示しています。

```
INCLUDE COND=(110,5,CH,EQ,C'ENGL',OR,110,5,CH,EQ,C'PSYCH')
INREC FIELDS=(1:1,75,76:170,4,80:110,5)
SORT FIELDS=(76,4,BI,D)
OUTREC FIELDS=(1:1,75,85:76,4,BI,EDIT=($IT.TT),95:80,5)
OUTFIL FNAMES=ENGL,INCLUDE=(95,5,CH,EQ,C'ENGL')
OUTFIL FNAMES=PSYCH,INCLUDE=(95,5,CH,EQ,C'PSYCH')
```

INCLUDE および INREC ステートメントでは、入力レコードに現れるとおりにフィールドを参照します。SORT および OUTREC ステートメントでは、再フォーマット済み INREC レコードに現れるとおりにフィールドを参照します。OUTFIL ステートメントでは、再フォーマット済み OUTREC レコードに現れるとおりにフィールドを参照します。

## 値を合計するときのオーバーフローを防止する

場合によっては、INREC を使用して要約フィールドをゼロで埋めることにより、オーバーフローを防止することができます。ただし、この方法は、負の固定小数点 2 進数データには使用できません。1 ではなく 0 を埋め込むと符号が変わってしまうからです。

表 37 の要約フィールドがオーバーフローしていた場合は、表 38 に示されたように、各フィールドの左側に 4 バイトを埋め込むことができます (2 進数フィールドは 2、4、または 8 バイトの長さでなければなりません)。

## レコードの再フォーマット

表 38. 要約フィールドの埋め込み

| Publisher |   | X'0...0' |   | Number In Stock |    | X'0...0' |    | Number Sold |    |
|-----------|---|----------|---|-----------------|----|----------|----|-------------|----|
| 1         | 4 | 5        | 8 | 9               | 12 | 13       | 16 | 17          | 20 |
| COR       |   |          |   | 103             |    |          |    | 161         |    |
| FERN      |   |          |   | 19              |    |          |    | 87          |    |
| VALD      |   |          |   | 42              |    |          |    | 97          |    |
| WETH      |   |          |   | 62              |    |          |    | 79          |    |

```
INREC FIELDS=(106, 4, 4Z, 162, 4, 4Z, 166, 4)
SORT FIELDS=(1, 4, CH, A)
SUM FIELDS=(5, 8, BI, 13, 8, BI)
```

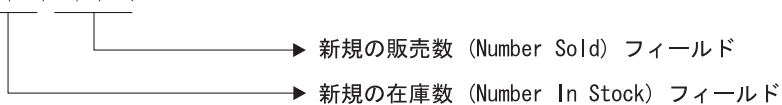


表 39 は、それぞれが 20 バイトの長さの出力レコードを示しています。

表 39. 要約フィールドの埋め込み

| Publisher |   | Number In Stock |    | Number Sold |    |
|-----------|---|-----------------|----|-------------|----|
| 1         | 4 | 5               | 12 | 13          | 20 |
| COR       |   | 103             |    | 161         |    |
| FERN      |   | 19              |    | 87          |    |
| VALD      |   | 42              |    | 97          |    |
| WETH      |   | 62              |    | 79          |    |

**重要:** OUTREC ステートメントは要約の後で処理されるので、OUTREC を使用してオーバーフローを防止することはできません。

## シーケンス番号の挿入

それぞれの学部コースに関連したすべての本の合計価格を表示したいが、学部コースでソートされたリストではなく、元の学部コース順のリストが必要であるとします。合計を得るには、以下のステートメントを使用して、学部 (Department) フィールドで SORT し、価格 (Price) フィールドを SUM する必要があります。

```
INREC FIELDS=(110,5,170,4)
SORT FIELDS=(1,5,CH,A)
SUM FIELDS=(6,4,BI)
```

77 ページの表 40 は、これらのステートメントの結果を示しています。

表 40. コース別の本価格の合計

| Department | Total Price |
|------------|-------------|
| 1 5        | 6 9         |
|            | 10195       |
| BIOL       | 2350        |
| BUSIN      | 4275        |
| COMP       | 10253       |
| ENGL       | 4640        |
| HIST       | 3785        |
| PSYCH      | 4800        |

SUM するために SORT する必要があったので (COPY では SUM できないことを覚えておいてください)、リストは、必要とする元の順序 (COMP、ブランク、BUSIN、ENGL、HIST、PSYCH、および BIOL という順序) ではなく、学部コースでソートされた順序になっています。元の順序に戻すには、レコードのソート前に各レコードにシーケンス番号を追加し、**2 番目のステップ**として、レコードをそのシーケンス番号によって元の順序にソートし戻すことができます。2 番目のステップの終わりにシーケンス番号を除去できます。

次のように、INREC ステートメントに若干変更を加えて上記のステートメントを使用すると、最初のステップとしてシーケンス番号を追加できます。

```
INREC FIELDS=(110,5,170,4,SEQNUM,8,ZD)
SORT FIELDS=(1,5,CH,A)
SUM FIELDS=(6,4,BI)
```

シーケンス番号は、最初のレコードの 00000001 から始まり、その後の各レコードごとに 00000001 ずつ増分されます。これにより、合計されたレコードをシーケンス番号によって元の学部コース順にソートし戻せるようになります。

表 41 は、この最初のステップの結果を示しています。出力レコードは一時データ・セットに保管されます。

表 41. コース別の本の価格の合計 (シーケンス番号付き)

| Department | Total Price | Sequence Number |
|------------|-------------|-----------------|
| 1 5        | 6 9         | 10 17           |
|            | 10195       | 00000002        |
| BIOL       | 2350        | 00000017        |
| BUSIN      | 4275        | 00000003        |
| COMP       | 10253       | 00000001        |
| ENGL       | 4640        | 00000005        |
| HIST       | 3785        | 00000011        |
| PSYCH      | 4800        | 00000013        |

2 番目のステップとして以下のステートメントを使用すると、シーケンス番号でソートし、分かりやすい方法でデータを表示し、シーケンス番号を除去することができます。この 2 番目のステップの入力は、最初のステップで作成した一時データ・セットです。

## レコードの再フォーマット

```
SORT FIELDS=(10,8,CH,A)
OUTREC FIELDS=(1,5,CHANGE=(16,
 C'HIST',C'History',
 C'BUSIN',C'Business',
 C'COMP',C'Computer Science',
 C'ENGL',C'English',
 C'BIOL',C'Biology',
 C'PSYCH',C'Psychology'),
 NOMATCH=(C'Unaffiliated'),
 C' | ',
 6,4,BI,EDIT=($III,IIT.TT))
```

この 2 番目のステップからの出力レコードは、次のようになります。

|                  |          |
|------------------|----------|
| Computer Science | \$102.53 |
| Unaffiliated     | \$101.95 |
| Business         | \$42.75  |
| English          | \$46.40  |
| History          | \$37.85  |
| Psychology       | \$48.00  |
| Biology          | \$23.50  |

SEQNUM,m,f を使用すると、さまざまな長さのシーケンス番号をさまざまなフォーマットで作成できます。 START=j を使用すれば、1 ではなく j からシーケンス番号が開始されます。 INCR=i を使用すれば、1 ではなく i ずつシーケンス番号が増分されます。 RESTART=(p,m) を使用すれば、特定フィールドの値が変更されるたびに、開始値からシーケンス番号を開始できます。

次の入力レコードがあるとしてします。

|            |              |
|------------|--------------|
| New York   | Albany       |
| California | Morgan Hill  |
| New York   | Buffalo      |
| Arizona    | Tuscon       |
| California | San Jose     |
| New York   | Poughkeepsie |
| Arizona    | Phoenix      |
| California | Davis        |
| New York   | Armonk       |

位置 1-15 の州 (State) フィールドおよび位置 16-30 の都市 (City) フィールドでレコードを分類し、1000 から始まり 10 ずつ増分するシーケンス番号で 3 番目のフィールドを追加する場合は、次のステートメントに表示されているように、START=1000 および INCR=10 を使用できます。

```
SORT FIELDS=(1,30,CH,A)
OUTREC OVERLAY=(32:SEQNUM,5,ZD,START=1000,INCR=10)
```

出力レコードは次のようになります。

|            |              |       |
|------------|--------------|-------|
| Arizona    | Phoenix      | 01000 |
| Arizona    | Tuscon       | 01010 |
| California | Davis        | 01020 |
| California | Morgan Hill  | 01030 |
| California | San Jose     | 01040 |
| New York   | Albany       | 01050 |
| New York   | Armonk       | 01060 |
| New York   | Buffalo      | 01070 |
| New York   | Poughkeepsie | 01080 |

各レコードには異なるシーケンス番号があることに注意してください。

州 (State) が変更されるたびに、再度 1000 からシーケンス番号を開始する場合は、次のステートメントに表示されているように、RESTART=(1,15) を追加できます。

```

SORT FIELDS=(1,30,CH,A)
OUTREC OVERLAY=(32:SEQNUM,5,ZD,START=1000,INCR=10,
 RESTART=(1,15))

```

出力レコードは次のようになります。

|            |              |       |
|------------|--------------|-------|
| Arizona    | Phoenix      | 01000 |
| Arizona    | Tuscon       | 01010 |
| California | Davis        | 01000 |
| California | Morgan Hill  | 01010 |
| California | San Jose     | 01020 |
| New York   | Albany       | 01000 |
| New York   | Armonk       | 01010 |
| New York   | Buffalo      | 01020 |
| New York   | Poughkeepsie | 01030 |

州 (State) が変更されるたびに、シーケンス番号が再度 1000 から開始することに注意してください。

IFTHEN 文節をシーケンス番号と一緒に指定する場合、シーケンス番号は、その IFTHEN 文節によって選択されたレコードのサブセットに対してのみ増分されます。例えば、上記に示されている入力レコードに次のステートメントを指定したとします。

```

OPTION COPY
OUTREC IFTHEN=(WHEN=(1,15,CH,EQ,C'Arizona'),
 OVERLAY=(32:SEQNUM,2,ZD)),
 IFTHEN=(WHEN=(1,15,CH,EQ,C'California'),
 OVERLAY=(32:SEQNUM,4,ZD)),
 IFTHEN=(WHEN=(1,15,CH,EQ,C'New York'),
 OVERLAY=(32:SEQNUM,3,ZD))

```

出力レコードは次のようになります。

|            |              |      |
|------------|--------------|------|
| New York   | Albany       | 001  |
| California | Morgan Hill  | 0001 |
| New York   | Buffalo      | 002  |
| Arizona    | Tuscon       | 01   |
| California | San Jose     | 0002 |
| New York   | Poughkeepsie | 003  |
| Arizona    | Phoenix      | 02   |
| California | Davis        | 0003 |
| New York   | Armonk       | 004  |

INREC、OUTREC、および OUTFIL OUTREC を使用してシーケンス番号を作成する方法については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

## VB データ・セットについての考慮事項

SORT、MERGE、INCLUDE、OMIT、および SUM ステートメントのところで学習した同じ VB データ・セットに関する考慮事項が、INREC および OUTREC ステートメントにも適用されますが、さらに次のような考慮事項が適用されます。



### RDW

BUILD、FIELDS、または IFTHEN BUILD パラメーターを使用する場合は、INREC または OUTREC ステートメントのフィールドは、VB レコードの RDW を表すために、1,4 (開始位置が 1、長さが 4) を指定することによって開始する必要があります。以下に例を示します。

```
INREC FIELDS=(1,4,8,25)
```

最初のフィールドに、RDW の直後の入力データ・バイト (すなわち、位置 5 から始まる) を含めたい場合は、1,4 ではなく 1,n (開始位置が 1、長さが n で、n は 5 以上) を使用できます。以下に例を示します。

```
OUTREC BUILD=(1,10,21,30)
```

(RDW の位置 1-2 からの) レコード長を表示したい場合は、1,4 を使用して、レコード長を表す 1,2 が後ろに続いた RDW を表すことができます。以下に例を示します。

```
INREC FIELDS=(1,4,C'Record length is ',1,2,BI,M11)
```

INREC または OUTREC の OVERLAY または IFTHEN OVERLAY パラメーターを使用する場合、位置 1-4 の RDW をオーバーレイしてはなりません。したがって、すべての項目は位置 5 から、または位置 5 の後から開始する必要があります。以下に例を示します。

```
INREC OVERLAY=(5:5,3,TRAN=UTOL,21:18,2,11:C'***')
```

このステートメントは、位置 5-7 の入力フィールドを小文字に変換します。最初の項目に c: を指定し忘れると、c はデフォルトで 1 に解釈されます。この結果、RDW をオーバーレイするのでエラー・メッセージが表示されます。以下に例を示します。

```
INREC OVERLAY=(5:3,TRAN=UTOL) Results in an error message
```

これは、出力位置 1-3 を、入力位置 5-7 の小文字変換でオーバーレイするものと解釈されます。位置 1-4 をオーバーレイできないので、エラー・メッセージが表示されます。

OVERLAY 項目に 1:、2:、3:、または 4: を使用しないように注意してください。例えば、次のステートメントの結果、エラー・メッセージが表示されます。これは 2 番目の項目が RDW をオーバーレイするからです。

```
OUTREC OVERLAY=(8:5,3,TRAN=UTOL, ok
2:18,2) Results in an error message
```

### 開始位置および桁

BUILD、FIELDS、OVERLAY、IFTHEN BUILD、または IFTHEN OVERLAY パラメーターのために入力フィールドをコーディングするときは、4 バイトの RDW のための 4 を開始位置に追加するのを忘れないでください。出力桁をコーディングするときも同様です。例えば、次の OUTREC ステートメントは、VB 入力レコードの最初データ・バイトから 6 番目のデータ・バイトのフィールドを、VB 出力レコードの位置 21-26 に入れます。

```
OUTREC FIELDS=(1,4,21:5,6)
```



## 可変データ

BUILD、FIELDS、または IFTHEN BUILD パラメーターを使用して、各 VB 入力レコードの終わりの可変データを VB 出力レコードに組み込みたい場合は、長さを指定せずに可変入力データの開始位置だけを指定します。以下に例を示します。

```
OUTREC FIELDS=(1,4,C'Fixed data ',5,10,C' Variable data ',15)
```

各 VB レコードの終わりの可変データには、以下のものも使用できます。

- p,HEX - 16 進表示
- p,TRAN=UTOL - 大文字から小文字へ
- p,TRAN=LTOU - 小文字から大文字へ
- p,TRAN=ALTSEQ - ALTSEQ テーブルを使用した変換

例えば、次の OUTREC ステートメントは、VB レコードの内容全体を 16 進数で表示します。

```
OUTREC FIELDS=(1,4,1,4,HEX,5,HEX)
```

OVERLAY または IFTHEN パラメーターでは、長さを指定せずに開始位置を指定してはなりません。OVERLAY または IFTHEN OVERLAY で次のいずれかを指定すると、DFSORT はエラー・メッセージを出します。

- p
- p,HEX
- p,TRAN=UTOL
- p,TRAN=LTOU
- p,TRAN=ALTSEQ .

以下に例を示します。

```
INREC OVERLAY=(5:C'***',
21:21,HEX) Results in an error message
```

### 要約

本章では、OUTREC ステートメントと INREC ステートメントの BUILD、FIELDS、OVERLAY、および IFTHEN パラメーターを使用して、固定位置/長さの入力レコードを出力用にさまざまな方法で再フォーマットする方法を説明しました。OUTFIL ステートメントだけでなく、OUTREC および INREC ステートメントを使用して、フィールドを削除する方法、フィールドを再配列する方法、フィールドをオーバーレイする方法、分離文字とシーケンス番号を挿入する方法、文字と数値データをさまざまな形式に変換する方法、算術演算を実行する方法、およびレコードごとに異なる方法で変更する方法を学習しました。

## レコードの再フォーマット

---

## 第 6 章 可変フィールドを持つレコードの再フォーマット

直前の章では、固定フィールドを持つレコードを再フォーマットする方法について説明しました。固定フィールドとは、どのレコードでも同じ位置で開始し、同じ長さを持つフィールドです。この章では、可変フィールドを持つレコードを同じように再フォーマットする方法を説明します。可変フィールドとは、レコードごとに開始位置と長さが異なるフィールド（コンマ区切り値 (CSV) など）です。

---

### BUILD および OVERLAY での %nn 解析済みフィールドの使用

位置/長さが可変のフィールドにはさまざまなタイプがあります。区切り文字で区切られたフィールド、コンマ区切り値 (CSV)、タブ区切り値、ブランク区切り値、キーワード区切り値などです。例えば、次のようにコンマ区切り値を持つ 4 つのレコードがあるとします。

```
Wayne,M,-53,-1732,Gotham
Summers,F,+7258,-273,Sunnydale
Kent,M,+213,-158,Metropolis
Prince,F,-164,+1289,Gateway
```

各レコードには、コンマで区切られた 5 つの可変フィールドがあることに注意してください。これらのフィールドは、どのレコードでも開始位置と終了位置が同じであるわけではなく、レコードごとに長さが異なります。したがって、INREC、OUTREC、または OUTFIL ステートメントの BUILD (または FIELDS または OUTREC) もしくは OVERLAY オペランドで、これらのフィールドに対して単に開始位置と長さ (p,m) を指定することはできません。しかし、INREC、OUTREC、または OUTFIL ステートメントの PARSE オペランドを使用すると、各可変入力フィールドから固定解析済みフィールドに関連データを抽出する方法を DFSORT に指示する規則を定義することができます。その後、固定入力フィールドを使用する場合と同じように、BUILD または OVERLAY オペランドで固定解析済みフィールドを使用できます。

%nn 名を使用して可変フィールドを固定解析済みフィールドに変換するために、解析済みフィールドを定義します (ここで、nn は 00 から 99 にすることができます)。1 回の実行ごとに最大 100 個の解析済みフィールドを定義し、使用することができます。各 %nn 解析済みフィールドは 1 回だけ定義する必要があります。%nn 解析済みフィールドは、BUILD または OVERLAY オペランドで使用する前に、PARSE オペランドで定義されなければなりません。

次の出力レコードを生成するために、CSV レコードを再フォーマットしたいものとします。

|         |        |            |
|---------|--------|------------|
| Wayne   | -178.5 | Gotham     |
| Summers | 698.5  | Sunnydale  |
| Kent    | 5.5    | Metropolis |
| Prince  | 112.5  | Gateway    |

次の OUTREC ステートメントを使用すると、可変フィールドを解析し、再フォーマットすることができます。

```

OUTREC PARSE=(%01=(ENDBEFR=C',' ,FIXLEN=8),
 %=(ENDBEFR=C','),
 %03=(ENDBEFR=C',' ,FIXLEN=5),
 %04=(ENDBEFR=C',' ,FIXLEN=5),
 %05=(FIXLEN=10)),
BUILD=(%01,14:%03,SFF,ADD,%04,SFF,EDIT=(SIIT.T),SIGNS=(,-),
 25:%05)

```

**PARSE** オペランドは、次のように各可変フィールドを固定解析済みフィールドに抽出する方法を定義します。

- **%01** 解析済みフィールドは、最初の可変フィールドを 8 バイトの固定解析済みフィールドに抽出するのに使用されます。ENDBEFR=C',' は、次のコンマ (最初の可変フィールドの後のコンマ) の前のバイトでデータの抽出を停止することを DFSORT に指示します。FIXLEN=8 は、%01 解析済みフィールドの長さが 8 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は Wayne を 8 バイト %01 解析済みフィールドに抽出します。Wayne は 5 文字でしかなく、一方、%01 解析済みフィールドの長さは 8 バイトであるので、DFSORT は %01 解析済みフィールドの右側に 3 つのブランクを埋め込みます。また、ENDBEFR=C',' は、最初の可変フィールドの後のコンマをスキップオーバーしてから、2 番目の可変フィールドを解析することも、DFSORT に指示します。
- **%** 解析済みフィールドは、2 番目の可変フィールドに対して何も抽出することなく、2 番目の可変フィールドをスキップするのに使用します。このフィールドは出力レコードに必要なないので、% を使用して無視することができます。したがって、最初のレコードに対して M を無視します。ENDBEFR=C',' は、2 番目の可変フィールドの後のコンマをスキップオーバーしてから、3 番目の可変フィールドを解析することを DFSORT に指示します。
- **%03** 解析済みフィールドは、3 番目の可変フィールドを 5 バイトの固定解析済みフィールドに抽出するのに使用されます。ENDBEFR=C',' は、次のコンマ (3 番目の可変フィールドの後のコンマ) の前でデータの抽出を停止することを DFSORT に指示します。FIXLEN=5 は、%03 解析済みフィールドの長さが 5 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は -53 を 5 バイト %03 解析済みフィールドに抽出します。-53 は 3 文字でしかなく、一方、%03 解析済みフィールドの長さは 5 バイトであるので、DFSORT は %03 解析済みフィールドの右側に 2 つのブランクを埋め込みます。また、ENDBEFR=C',' は、3 番目の可変フィールドの後のコンマをスキップオーバーしてから、4 番目の可変フィールドを解析することも、DFSORT に指示します。
- **%04** 解析済みフィールドは、4 番目の可変フィールドを 5 バイトの固定解析済みフィールドに抽出するのに使用されます。ENDBEFR=C',' は、次のコンマ (4 番目の可変フィールドの後のコンマ) の前でデータの抽出を停止することを DFSORT に指示します。FIXLEN=5 は、%04 解析済みフィールドの長さが 5 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は -1732 を 5 バイト %04 解析済みフィールドに抽出します。-1732 は 5 文字なので、5 バイトの %05 解析済みフィールドを埋め尽くし、埋め込みは必要ありません。また、ENDBEFR=C',' は、4 番目の可変フィールドの後のコンマをスキップオーバーしてから、5 番目の可変フィールドを解析することも、DFSORT に指示します。

- **%05** 解析済みフィールドは、5 番目の可変フィールドを 10 バイトの固定解析済みフィールドに抽出するのに使用されます。FIXLEN=10 は、%05 解析済みフィールドの長さが 10 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は Gotham と 4 つのブランクを 10 バイト %01 解析済みフィールドに抽出します。

**BUILD** オペランドは、あらかじめ抽出された固定解析済みフィールドを使用して、次のように出力レコードを作成します。

- **%01** は、最初の可変フィールドから抽出された 8 バイトの固定長データを、出力レコードの位置 1-8 にコピーします。最初のレコードの場合、位置 1-8 には 'Wayne ' が入ります。
- **14:%03,SFF,ADD,%04,SFF,EDIT=(SIIT.T),SIGNS=(,-)** は、3 番目の可変フィールドから抽出された 5 バイトの固定長データを、4 番目の可変フィールドから抽出された 5 バイト固定長データに追加します。編集された 6 バイトの結果が、出力レコードの位置 14-19 に置かれます。最初のレコードの場合、位置 14-19 には -178.5 (-53 + -1732 = -1785 を -178.5 に編集) が含まれます。%03 および %04 解析済みフィールドの右側にブランクが埋め込まれるので、SFF フォーマットを使用して、符号と数字を正しく処理する必要があることに注意してください。
- **25:%05** は、5 番目の可変フィールドから抽出された 10 バイトの固定長データを、出力レコードの位置 25-34 にコピーします。最初のレコードの場合、位置 25-34 に 'Gotham ' が入ります。

---

## IFTHEN での %nn 解析済みフィールドの使用

レコードのタイプごとに異なる可変の位置/長さフィールドがある場合、それらのフィールドを %nn 固定解析済みフィールドに変換して、IFTHEN 文節で使用することができます。WHEN=INIT 文節で %nn 解析済みフィールドを定義する場合、後続の IFTHEN 文節だけでなく、その文節の IFTHEN BUILD または IFTHEN OVERLAY サブオペランドで、そのフィールドを使用することができます。

WHEN=(logexp)、WHEN=ANY、または WHEN=NONE 文節で %nn 解析済みフィールドを定義する場合、その文節の IFTHEN BUILD または IFTHEN OVERLAY サブオペランドでそのフィールドを使用できます。

次の入力レコードがあるとします。

```
1/Sam/Charlie;27
2/Bill/48
1/Frank/Vicky;02
1/William/Dale;86
2/Helen/15
```

「1」で始まるレコードには、1 つの固定フィールド（「1」）と 3 つの可変フィールドがあり、「2」で始まるレコードには、1 つの固定レコード（「2」）と 2 つの可変フィールドがあることに注意してください。各レコード・タイプの可変フィールドでは、どのレコードでも開始位置と終了位置が同じであるわけではなく、レコードが異なれば長さも異なります。

次の出力レコードを生成するために、これらの入力レコードを再フォーマットしたいものとします。

|   |         |         |     |
|---|---------|---------|-----|
| 1 | Sam     | Charlie | 2.7 |
| 2 | Bill    |         | 4.8 |
| 1 | Frank   | Vicky   | 0.2 |
| 1 | William | Dale    | 8.6 |
| 2 | Helen   |         | 1.5 |

次の INREC ステートメントを使用すると、固定フィールドと可変フィールドを解析し、再フォーマットすることができます。

```
INREC IFOUTLEN=50,
 IFTHEN=(WHEN=INIT,
 PARSE=(%00=(ABSPOS=3,ENDBEFR=C'/',FIXLEN=8))),
 IFTHEN=(WHEN=(1,1,CH,EQ,C'1'),
 PARSE=(%01=(ABSPOS=3,STARTAFT=C'/',ENDBEFR=C';',
 FIXLEN=8),%02=(FIXLEN=2))),
 BUILD=(1,1,4:%00,18:%01,30:%02,ZD,EDIT=(T.T))),
 IFTHEN=(WHEN=(1,1,CH,EQ,C'2'),
 PARSE=(%03=(ABSPOS=3,STARTAFT=C'/',FIXLEN=2))),
 BUILD=(1,1,4:%00,30:%03,ZD,EDIT=(T.T)))
```

WHEN=INIT 文節の PARSE サブオペランドは、すべてのレコードに対して、%00 解析済みフィールドを使用して、最初の可変フィールドを 8 バイトの固定解析済みフィールドに抽出します。ABSPOS=3 は、位置 3 で開始することを DFSORT に指示します。ENDBEFR=C'/' は、次のスラッシュ (最初の可変フィールドの後のスラッシュ) の前でデータの抽出を停止することを DFSORT に指示します。FIXLEN=8 は、%00 解析済みフィールドの長さが 8 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は Sam を 8 バイト %00 解析済みフィールドに抽出します。Sam は 3 文字でしかなく、一方、%00 解析済みフィールドの長さは 8 バイトであるので、DFSORT は %00 解析済みフィールドの右側に 5 つのブランクを埋め込みます。WHEN=INIT 文節は、後続の文節に使用できるように %00 を定義します。( %00 は、この WHEN=INIT 文節の BUILD または OVERLAY サブオペランドで使用できますが、この場合は、他の文節に %00 を定義するだけです。)

最初の WHEN=(logexp) 文節の PARSE サブオペランドは、「1」レコードに対して、%01 解析済みフィールドを使用して、2 番目の可変フィールドを 8 バイト固定解析済みフィールドに抽出します。また、%02 解析済みフィールドを使用して、3 番目の可変フィールドを 2 バイトの固定解析済みフィールドに抽出します。

%01 の場合、ABSPOS=3 は、位置 3 から開始するように DFSORT に指示します。STARTAFT=C'/' は、次のスラッシュ (最初の可変フィールドの後のスラッシュ) の後からデータの抽出を開始することを DFSORT に指示します。ENDBEFR=C';' は、次のセミコロン (2 番目の可変フィールドの後のセミコロン) の前でデータの抽出を停止することを DFSORT に指示します。FIXLEN=8 は、%01 解析済みフィールドの長さが 8 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は Charlie を 8 バイト %01 解析済みフィールドに抽出します。Charlie は 7 文字でしかなく、一方、%01 解析済みフィールドの長さは 8 バイトであるので、DFSORT は %01 解析済みフィールドの右側に 1 つのブランクを埋め込みます。

%02 の場合、FIXLEN=2 は、%02 解析済みフィールドの長さが 2 バイトであることを DFSORT に指示します。したがって、最初のレコードについて、DFSORT は 27 を 2 バイト %02 解析済みフィールドに抽出します。

最初の WHEN=(logexp) 文節の BUILD サブオペランドは、WHEN=INIT 文節によって定義された %00 解析済みフィールド、ならびに最初の WHEN=(logexp) 文節によって定義された %01 および %02 解析済みフィールドを使用して、各「1」出力レコードを作成します。

2 番目の WHEN=(logexp) 文節の PARSE サブオペランドは、「2」レコードに対して、%03 解析済みフィールドを使用して、2 番目の可変フィールドを 2 バイトの固定解析済みフィールドに抽出します。

%03 の場合、ABSPOS=3 は、位置 3 から開始するように DFSORT に指示します。STARTAFT=C'/' は、次のスラッシュ (最初の可変フィールドの後のスラッシュ) の後からデータの抽出を開始することを DFSORT に指示します。FIXLEN=2 は、%03 解析済みフィールドの長さが 2 バイトであることを DFSORT に指示します。したがって、2 番目のレコードについて、DFSORT は 48 を 2 バイト %03 解析済みフィールドに抽出します。

2 番目の WHEN=(logexp) 文節の BUILD サブオペランドは、WHEN=INIT 文節によって定義された %00 解析済みフィールド、および 2 番目の WHEN=(logexp) 文節によって定義された %03 解析済みフィールドを使用して、各「2」出力レコードを作成します。

---

## BUILD および OVERLAY で %nn フィールドを使用できる項目

p,m を使用するのと同じように、BUILD または OVERLAY で %nn 解析済みフィールドを使用できます。 %nn を使用できる項目のリストは次のとおりです (詳しくは、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照)。

- %nn
- %nn,HEX
- %nn,TRAN=LTOU
- %nn,TRAN=UTOL
- %nn,TRAN=ALTSEQ
- %nn,f,edit
- %nn,f,to
- %nn,f in arexp,edit
- %nn,f in arexp,to
- %nn,Y2x
- %nn,Y2x,edit
- %nn,Y2x,to
- %nn,Y2x(c)
- %nn,Y2xP
- %nn,lookup
- %nn as set field in lookup
- %nn as set field in NOMATCH
- %nn,justify
- %nn,squeeze
- RESTART=(%nn) in SEQNUM



## PARSE パラメーター

PARSE で次のパラメーターを使用すると、可変の位置/長さデータを %nn 固定解析済みフィールドに抽出するための規則を定義できます。

- **FIXLEN=m:** この %nn 固定解析済みフィールドに対して抽出された可変データを含む、固定域の長さ (m) を指定します。
- **ABSPOS=p:** 入力位置 p からデータの抽出を開始します。
- **ADDPOS=x:** 現在位置 + x からデータの抽出を開始します。
- **SUBPOS=y:** 現在位置 - y からデータの抽出を開始します。
- **STARTAFT=string:** 文字ストリングまたは 16 進数ストリングの終わりの後のバイトから、データの抽出を開始します。
- **STARTAFT=BLANKS:** 次のブランク・グループの終わりの後から、データの抽出を開始します。
- **STARTAT=string:** 文字ストリングまたは 16 進数ストリングの最初のバイトから、データの抽出を開始します。
- **STARTAT=BLANKS:** 最初のブランク・グループの先頭から、データの抽出を開始します。
- **STARTAT=NONBLANK:** 次の非ブランクからデータの抽出を開始します。
- **ENDBEFR=string:** 文字ストリングまたは 16 進数ストリングの先頭の前のバイトで、データの抽出を停止します。
- **ENDBEFR=BLANKS:** 次のブランク・グループの前のバイトで、データの抽出を停止します。
- **ENDAT=string:** 文字ストリングまたは 16 進数ストリングの最後のバイトで、データの抽出を停止します。
- **ENDAT=BLANKS:** 次のブランク・グループの終わりで、データの抽出を停止します。
- **PAIR=APOST:** アポストロフィ (') のペアの間にあるストリングまたはブランクを検索しません。
- **PAIR=QUOTE:** 引用符 (") のペアの間にあるストリングまたはブランクを検索しません。

%nn 解析済みフィールドの定義と使用法の詳細、および詳しい例については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

### 要約

この章では、INREC、OUTREC、および OUTFIL ステートメントの BUILD、OVERLAY、および IFTHEN パラメーターで %nn 固定解析済みフィールドとして位置/長さが可変のフィールドを使用する方法の概要を説明しました。



---

## 第 7 章 複数の出力データ・セットおよび報告書の作成

OUTFIL 制御ステートメントを使用して、ソート済み、マージ済み、またはコピー済みの入力を 1 回受け渡すだけで、1 つ以上の出力データ・セットおよび報告書を作成することができます。それぞれの OUTFIL ステートメントで異なる OUTFIL パラメーターを使用できるので、単一の DFSORT アプリケーションで、同一の出力データ・セットを作成することもできますし、全く異なる出力データ・セットを作成することもできます。それぞれの OUTFIL ステートメントでは、以下を含む、広範囲にわたるタスクを実行できます。

- STARTREC および ENDREC パラメーターを使用してレコードの順次サブセットを選択する。
- SAMPLE パラメーターを使用してレコードのサンプルを選択する。
- INCLUDE および OMIT パラメーターを使用してレコードのサブセットを選択する。これらのパラメーターは、INCLUDE および OMIT ステートメントのすべての機能を持っています。
- SAVE パラメーターを使用して「廃棄」レコードを選択する。このパラメーターは、STARTREC、ENDREC、SAMPLE、INCLUDE、または OMIT パラメーターの結果として選択されなかったレコードを保管します。
- PARSE、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターを使用して、位置/長さが固定のフィールドまたは位置/長さが可変のフィールドを持つレコードを再フォーマットする。これらのパラメーターには、OUTREC ステートメントの PARSE、BUILD、FIELDS、OVERLAY、または IFTHEN パラメーターのすべての機能があります。また、これらのパラメーターを使用すると、各入力レコードから複数の出力レコードやブランクの出力レコードを作成することもできます。
- REPEAT パラメーターを使用してレコードを繰り返す。
- SPLIT、SPLITBY、および SPLITIR パラメーターを使用して出力データ・セットの間でレコードを分割する。
- BUILD、OUTREC、OVERLAY、IFTHEN、HEADERn、TRAILERn、SECTIONS、LINES、NODETAIL、BLKCCH1、BLKCCH2、BLKCCT1、および REMOVECC パラメーターを使用して、最高 3 レベル (報告書、ページ、およびセクション) の明細報告書を作成する。報告書には、ヘッダー、データ、ブランク行、トレーラー、統計、セクション、タイム・スタンプ、ページ番号、およびカウントなどのエレメントを含めることができます。
- FTOV パラメーターを使用して FB レコードを VB レコードに変換する、または VTOF パラメーターを使用して VB レコードを FB レコードに変換する。

OUTFIL ステートメントは、その他の任意の DFSORT ステートメント (例えば、INCLUDE、OMIT、INREC、SORT、MERGE、OPTION、SUM、および OUTREC) と一緒に使用できます。193 ページの『付録 C. 制御ステートメントの処理順序』に示されているように、OUTFIL 処理は、他のステートメントに対するすべての処理が終わった後で、最後に実行されます。したがって、他のステートメントに対する処理で作成された出力レコードは、OUTFIL 処理の入力レコードとなります。例

例えば、OUTFIL は、SORT ステートメントが指定された場合はソート済みレコードを処理し、OPTION COPY ステートメントが指定された場合はコピー済みレコードを処理します。

各種の DFSORT ステートメントをさまざまな方法で組み合わせることにより、たくさんの柔軟性が得られます。実際には、INREC ステートメント、OUTREC ステートメント、および OUTFIL ステートメントを使用して、入力レコードを 3 回再フォーマットできます。さらに、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保持し、異なる INCLUDE または OMIT および OUTREC、OVERLAY、または IFTHEN パラメーターを指定した OUTFIL ステートメントを使用して、残りのレコードの異なるサブセットを、さまざまな方法で再フォーマットされた異なる OUTFIL データ・セットに書き込むことができます。

## 複数の同一コピーの作成

データ・セットのバックアップを、ローカル・サイトではディスクに、リモート・サイトではテープに作成したいとします。これは、OPTION COPY ステートメントと一緒に、OUTFIL と FNAMES パラメーターを使用することにより行うことができます。FNAMES パラメーターは、出力データ・セットに対する JCL 内の DD ステートメントをそれぞれの DD 名によって識別します。サンプル JCL および DFSORT 制御ステートメントを、以下に示します。

```
//COPY JOB A492,PROGRAMMER
//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//BACKUP DD DSN=A123456.BOOKS.BACKUP,DISP=OLD
//NEWYORK DD DSN=BOOKS,UNIT=3490,DISP=(,KEEP),VOL=SER=REMOT1,LABEL=(,SL)
//SANJOSE DD DSN=BOOKS,UNIT=3490,DISP=(,KEEP),VOL=SER=REMOT2,LABEL=(,SL)
//SYSIN DD *
 OPTION COPY
 OUTFIL FNAMES=(BACKUP,NEWYORK,SANJOSE)
/*
```

注: 上記の JCL には、2 つのテープ・データ・セットが含まれています。この例を実行する場合は、ユーザー独自のテープ・データ・セット情報で置き換えてください。

以下は、OUTFIL ステートメントを作成するためのステップです。

表 42. 複数の出力データ・セット・ジョブを得るための OUTFIL ステートメントの作成

| ステップ | アクション                                                                                                      |
|------|------------------------------------------------------------------------------------------------------------|
| 1    | 少なくとも 1 つのブランクを入れてから、 <b>OUTFIL</b> を入力する                                                                  |
| 2    | 少なくとも 1 つのブランクを入れてから、 <b>FNAMES=</b> を入力する                                                                 |
| 3    | それぞれの出力データ・セットの DD 名をコンマで区切り、括弧で囲んで入力する。この例では、DD 名は <b>BACKUP</b> 、 <b>NEWYORK</b> 、および <b>SANJOSE</b> です。 |

この OUTFIL ステートメントは、データ・セットのバックアップ・コピーを 3 部作成します (オンサイト・ディスク・データ・セットが 1 つと、リモート・サイトに送信できる 2 つのテープ)。

このタスクは、OUTFIL の FILES パラメーターを使用して行うこともできます。FILES では、固有の DD ステートメントを指名する代わりに、SORTOFd または SORTOFdd DD ステートメントを割り当てます。FILES を使用する場合はサンプル JCL および DFSORT 制御ステートメントを、以下に示します。

```
//COPY JOB A492,PROGRAMMER
//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//SORTOF1 DD DSN=A123456.BOOKS.BACKUP,DISP=OLD
//SORTOF2 DD DSN=BOOKS,UNIT=3490,DISP=(,KEEP),VOL=SER=REMOT1
//SORTOF3 DD DSN=BOOKS,UNIT=3490,DISP=(,KEEP),VOL=SER=REMOT2
//SYSIN DD *
 OPTION COPY
 OUTFIL FILES=(1,2,3)
/*
```

FNAMES と FILES は共に同じ出力を作成しますが、FNAMES では、1 または 2 文字のあいまいな接尾部ではなく、OUTFIL DD ステートメントに対して 8 文字までの記述名を使用できます。FNAMES が推奨パラメーターであり、本章の残りの複数出力例ではすべて FNAMES を使用しています。

注: OUTFIL ステートメントに対して FNAMES または FILES パラメーターを指定しなかった場合、デフォルトの OUTFIL DD 名は SORTOUT です。

## 相対レコード番号による選択および抽出

入力レコードをコピー、ソート、またはマージすると、結果の出力レコードは、最初のレコードの 1 から始まり、最後のレコードの *n* (コピー、ソート、またはマージされたレコードの合計数) で終わる暗黙の相対レコード番号を持っています。

表 43 は、SORT.BRANCH データ・セットをコピーして作成した出力レコードの City (市区町村) フィールドと、各レコードの相対レコード番号 (省略形は RRN) を示しています (出力データ・セットでは、相対レコード番号は実際には**現れません**)。

表 43. コピーの場合の相対レコード番号

| City          | RRN |
|---------------|-----|
| 1        15   |     |
| Los Angeles   | 1   |
| San Francisco | 2   |
| Fort Collins  | 3   |
| Sacramento    | 4   |
| Sunnyvale     | 5   |
| Denver        | 6   |
| Boulder       | 7   |
| Morgan Hill   | 8   |
| Vail          | 9   |
| San Jose      | 10  |
| San Diego     | 11  |
| Aspen         | 12  |

表 44 は、City (市区町村) に基づいて SORT.BRANCH データ・セットをソートすることによって作成した出力レコードの City フィールドと、各レコードの RRN を示しています。コピーした出力データ・セットとソートした出力データ・セットでは、出力レコードの順序が異なるので RRN が異なることに注意してください (例えば、'Denver' を含むレコードの相対レコード番号は、コピーしたデータ・セットでは 6、ソートしたデータ・セットでは 3 です)。

表 44. ソートの場合の相対レコード番号

| City            | RRN |
|-----------------|-----|
| 1            15 |     |
| Aspen           | 1   |
| Boulder         | 2   |
| Denver          | 3   |
| Fort Collins    | 4   |
| Los Angeles     | 5   |
| Morgan Hill     | 6   |
| Sacramento      | 7   |
| San Diego       | 8   |
| San Francisco   | 9   |
| San Jose        | 10  |
| Sunnyvale       | 11  |
| Vail            | 12  |

これらの相対レコード番号を使用して、異なる OUTFIL データ・セットに書き込む順次レコードのグループを選択することができます。STARTREC=n は、相対レコード n から OUTFIL 処理を開始するよう DFSORT に指示します。ENDREC=m は、相対レコード m で OUTFIL 処理を終了するよう DFSORT に指示します。以下のステートメントは、この働きを示します。

```
SORT FIELDS=(1,15,CH,A)
OUTFIL FAMES=OUT1,ENDREC=4
OUTFIL FAMES=OUT2,STARTREC=2,ENDREC=6
OUTFIL FAMES=OUT3,STARTREC=7,ENDREC=9
OUTFIL FAMES=OUT4,STARTREC=10
```

OUT1 の場合は STARTREC が指定されていないので、DFSORT は RRN 1 から始まり、RRN 4 で終了します。OUT2 の場合、DFSORT は RRN 2 から始まり、RRN 6 で終了します。(RRN 2、3、および 4 は OUT1 と OUT2 の両方に書き込まれることに注意してください。) OUT3 の場合、DFSORT は RRN 7 から始まり、RRN 9 で終了します。OUT4 の場合は ENDREC が指定されていないので、DFSORT は RRN 10 から始まり、最後のレコード (RRN 12) で終了します。それぞれの OUTFIL データ・セットの RRN は、次のようになります。

```
OUT1: 1, 2, 3, 4
OUT2: 2, 3, 4, 5, 6
OUT3: 7, 8, 9
OUT4: 10, 11, 12
```

相対レコード番号を使用してレコードの「サンプル」を取ることもできます。SAMPLE=n は、STARTREC=x で示されたレコードで始まり、ENDREC=y で示されたレコードで終わるまで、n 番目のレコードごとに処理するよう DFSORT に指示します。SAMPLE=(n,m) は、STARTREC=x で示されたレコードから始まり、

ENDREC=y で示されたレコードまで、n 番目のレコードごとに m 個のレコードを処理するよう DFSORT に指示します。以下のステートメントは、この働きを示しています。

```
OPTION COPY
OUTFIL FNames=OUT1,SAMPLE=3
OUTFIL FNames=OUT2,STARTREC=4,SAMPLE=2,ENDREC=10
OUTFIL FNames=OUT3,STARTREC=2,SAMPLE=(5,3)
```

OUT1 の場合、DFSORT は RRN 1 から始まり、RRN 10 まで 3 番目のレコードごとにレコードを書き込みます (RRN 13 が次のレコードですが、12 レコードしかありません)。OUT2 の場合、DFSORT は RRN 4 から始まり、RRN 10 まで 1 レコードおきにレコードを書き込みます。OUT3 の場合、DFSORT は RRN 2 から始めて 3 レコードを書き込み、レコード 2 の後は RRN 12 まで、5 番目のレコードごとに 3 レコードを書き込みます。それぞれの OUTFIL データ・セットのレコードは、次のようになります。

```
OUT1: 1, 4, 7, 10
OUT2: 4, 6, 8, 10
OUT3: 2, 3, 4, 7, 8, 9, 12
```

---

## 廃棄レコードの組み込み、省略、および保管

29 ページの『第 3 章 レコードの組み込みまたは省略』では、INCLUDE および OMIT ステートメントを使用して、ソート、コピー、またはマージを行う入力レコードのサブセットだけを選択する方法を学習しました。INCLUDE および OMIT パラメーターを使用することによっても、個々の OUTFIL 出力データ・セットに合わせてレコードのサブセットを同じように選択できます。異なる OUTFIL ステートメントでは、異なる INCLUDE または OMIT パラメーターを使用できます。

INCLUDE および OMIT ステートメントの COND パラメーターで有効な論理式はすべて、OUTFIL の INCLUDE および OMIT パラメーターでも有効です。ただし、INCLUDE および OMIT ステートメントの場合のように、OUTFIL ステートメントで FORMAT パラメーターを指定することはできません。INCLUDE および OMIT ステートメントの場合の VB データ・セットに関する考慮事項は、INCLUDE および OMIT パラメーターにも適用されます。

ブックストア・データ・セットをタイトルによってソートし、English (英文学)、History (歴史学)、および Psychology (心理学) 学部ごとに別個の出力データ・セットを作成したいとします。この 3 つの出力データ・セットを作成するには、以下の OUTFIL ステートメントを使用できます。

```
SORT FIELDS=(1,75,CH,A)
OUTFIL FNames=ENGLOUT,INCLUDE=(110,5,CH,EQ,C'ENGL')
OUTFIL FNames=HISTOUT,INCLUDE=(110,5,CH,EQ,C'HIST')
OUTFIL FNames=PSYCHOUT,INCLUDE=(110,5,CH,EQ,C'PSYCH')
```

SORT ステートメントは、タイトル (Title) フィールドによって入力レコードをソートします。

最初の OUTFIL ステートメントは、English 学部のレコードを ENGLOUT データ・セットに書き込みます。結果は 94 ページの表 45 に示されています。

表 45. ENGLOUT データ・セット内の出力レコード

| Book Title                            | Course Department |
|---------------------------------------|-------------------|
| 1 75                                  | 110 114           |
| EDITING SOFTWARE MANUALS              | ENGL              |
| INKLINGS: AN ANTHOLOGY OF YOUNG POETS | ENGL              |
| MODERN ANTHOLOGY OF WOMEN POETS       | ENGL              |
| SHORT STORIES AND TALL TALES          | ENGL              |
| THE COMPLETE PROOFREADER              | ENGL              |

2 番目の OUTFIL ステートメントは、History 学部のレコードを HISTOUT データ・セットに書き込みます。結果は表 46 に示されています。

表 46. HISTOUT データ・セット内の出力レコード

| Book Title                | Course Department |
|---------------------------|-------------------|
| 1 75                      | 110 114           |
| CRISES OF THE MIDDLE AGES | HIST              |
| EIGHTEENTH CENTURY EUROPE | HIST              |
| THE INDUSTRIAL REVOLUTION | HIST              |

3 番目の OUTFIL ステートメントは、Psychology 学部のレコードを PYSCHOUT データ・セットに書き込みます。結果は表 47 に示されています。

表 47. PYSCHOUT データ・セット内の出力レコード

| Book Title                        | Course Department |
|-----------------------------------|-------------------|
| 1 75                              | 110 114           |
| ADVANCED TOPICS IN PSYCHOANALYSIS | PSYCH             |
| INTRODUCTION TO PSYCHOLOGY        | PSYCH             |

上記の例では、English、History、および Psychology 学部の本を対象とした出力データ・セットを作成しました。残りの本を収容する追加の出力データ・セット (RESTOUT) を作成したい場合は、以下のステートメントを使用できます。

```

SORT FIELDS=(1,75,CH,A)
OUTFIL FNames=ENGLOUT,INCLUDE=(110,5,CH,EQ,C'ENGL')
OUTFIL FNames=HISTOUT,INCLUDE=(110,5,CH,EQ,C'HIST')
OUTFIL FNames=PYSCHOUT,INCLUDE=(110,5,CH,EQ,C'PSYCH')
OUTFIL FNames=RESTOUT,
 INCLUDE=(110,5,CH,NE,C'ENGL',AND,
 110,5,CH,NE,C'HIST',AND,
 110,5,CH,NE,C'PSYCH')

```

これは、学部が 3 つだけの場合は有用ですが、学部の数が増えると、RESTOUT に対する INCLUDE ステートメントはもっと複雑になります。

OUTFIL ステートメントのグループによって廃棄されたレコードを保持しておくには、DFSORT の SAVE パラメーターを使用した方が簡単です。SAVE パラメーターは、INCLUDE、OMIT、STARTREC、ENDREC、または SAMPLE パラメーターの結果として他の OUTFIL データ・セットのために選択されなかったレコードを、その OUTFIL データ・セットのために選択します。したがって、上記のステートメントの代わりに以下のステートメントを使用すれば、同じ結果が得られます。



```

SORT FIELDS=(1,75,CH,A)
OUTFIL FNames=ENGLOUT,INCLUDE=(110,5,CH,EQ,C'ENGL')
OUTFIL FNames=HISTOUT,INCLUDE=(110,5,CH,EQ,C'HIST')
OUTFIL FNames=PSYCHOUT,INCLUDE=(110,5,CH,EQ,C'PSYCH')
OUTFIL FNames=RESTOUT,SAVE

```

4 番目の OUTFIL ステートメントは、English、History、または Psychology で使用されなかったレコードを RESTOUT データ・セットに書き込みます。結果は表 48 に示されています。

表 48. RESTOUT データ・セット内の出力レコード

| Book Title                    | Course | Department |
|-------------------------------|--------|------------|
| 1 75                          | 110    | 114        |
| COMPUTER LANGUAGES            | COMP   |            |
| COMPUTERS: AN INTRODUCTION    | COMP   |            |
| INTRODUCTION TO BIOLOGY       | BIOL   |            |
| LIVING WELL ON A SMALL BUDGET |        |            |
| NUMBERING SYSTEMS             | COMP   |            |
| PICK'S POCKET DICTIONARY      |        |            |
| STRATEGIC MARKETING           | BUSIN  |            |
| SUPPLYING THE DEMAND          | BUSIN  |            |
| SYSTEM PROGRAMMING            | COMP   |            |
| VIDEO GAME DESIGN             | COMP   |            |

#### これまで

これまでは、入力データ・セットの複数の同一コピーを作成する方法、出力する入力レコードを相対レコード番号によって選択し抽出する方法、および出力するよう選択した入力レコードを組み込む、省略する、または保管する方法を学習しました。次に、OUTFIL を使用してレコードの再フォーマット、繰り返し、および分割を行う方法を学習します。

## 再フォーマット

45 ページの『第 5 章 固定フィールドを持つレコードの再フォーマット』と 83 ページの『第 6 章 可変フィールドを持つレコードの再フォーマット』では、ソート、コピー、またはマージの処理中に、INREC および OUTREC ステートメントを使用してさまざまな方法で入力レコードを再フォーマットする方法を学習しました。また、PARSE、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターを使用して同じように OUTFIL 出力レコードを再フォーマットすることもできます。INREC および OUTREC ステートメントの場合の VB データ・セットに関する考慮事項は、PARSE、BUILD、OUTREC、OVERLAY および IFTHEN パラメーターにも適用されます。

OUTFIL ステートメントごとに、さまざまな PARSE、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターを使用できます。これらのパラメーターを使用すると、INREC および OUTREC ステートメントの場合と同じ広範囲にわた

るタスクを実行することができます (45 ページの『第 5 章 固定フィールドを持つレコードの再フォーマット』と 83 ページの『第 6 章 可変フィールドを持つレコードの再フォーマット』を参照)。

さらに、BUILD、OUTREC、または IFTHEN BUILD パラメーターを使用することにより、それぞれの入力レコードから 1 つ以上の出力レコードを作成し、レコードの前、中間、または後にブランク行を挿入することもできます。

出版社 FERN および WETH に関して別個のデータ・セットを作成したいとします。WETH では、1 行に販売部数とインストラクターの名前を入れます。FERN では、1 行にコース番号を入れ、別の行に価格を入れます。ブランク行を 2 行使用して、各本の情報を区切ることにします。この 2 つの出力データ・セットを作成するには、以下のステートメントを使用できます。

```
OPTION COPY
OUTFIL FNames=WETHBKS,
 INCLUDE=(106,4,CH,EQ,C'WETH'),
 OUTREC=(C'Sold ',166,4,BI,EDIT=(IIT),
 C' copies of Wethman, Inc. book for instructor ',
 160,2,X,145,15)
OUTFIL FNames=FERNBKS,
 INCLUDE=(106,4,CH,EQ,C'FERN'),
 OUTREC=(3:C'Fernall Brothers book #',SEQNUM,2,ZD,
 C' for course ',115,5,/,
 5:C'costs ',170,4,BI,EDIT=($IIT.TT),2/)
```

最初の OUTFIL ステートメントは、INCLUDE パラメーターを使用して出版社 WETH の本を選択し、OUTREC パラメーターを使用してそれぞれの本に関する出力レコードと必須情報を作成します。2 番目の OUTFIL ステートメントは、INCLUDE パラメーターを使用して出版社 FERN の本を選択し、別の OUTREC パラメーターを使用してそれぞれの本に関する出力レコードと必須情報を作成します。OUTREC パラメーターでは、INREC および OUTREC ステートメントのところで前述したように、フィールド、定数、シーケンス番号、および数値編集が使用されます。

/、n/ および /.../ は、OUTFIL ステートメントの BUILD、OUTREC、または IFTHEN BUILD パラメーターで使用できますが、OUTFIL ステートメントの OVERLAY または IFTHEN OVERLAY パラメーター、もしくは INREC および OUTREC ステートメントのパラメーターでは使用できません。BUILD、OUTREC または IFTHEN BUILD パラメーターの始まりまたは終わりで n/ を使用すると、n ブランク行が挿入されます。BUILD、OUTREC または IFTHEN パラメーターの中間で n/ を使用すると、n-1 ブランク行が挿入されます。

したがって、FERNBKS に対する OUTREC パラメーターでは、中間の 115,5 の後の / (0 ブランク行) は、それぞれの FERN 入力レコードから 2 つの出力レコードを作成するために使用され、終わりの 2/ (2 ブランク行) は、それぞれの FERN 本に関する情報の後に 2 ブランク行を挿入するために使用されています。

WETHBKS として作成される結果は、次のようになります。

```
Sold 26 copies of Wethman, Inc. book for instructor CL CHATTERJEE
Sold 9 copies of Wethman, Inc. book for instructor ST GOODGOLD
Sold 23 copies of Wethman, Inc. book for instructor DC SMITH
Sold 21 copies of Wethman, Inc. Book for instructor HR BISCARDI
```



FERNBKS として作成される結果は、次のようになります。

```
Fernall Brothers book #01 for course 00032
costs $26.00
```

```
Fernall Brothers book #02 for course 00032
costs $3.60
```

```
Fernall Brothers book #03 for course 10347
costs $6.25
```

```
Fernall Brothers book #04 for course 30975
costs $26.00
```

DFSORT は自動的に、それぞれの OUTFIL データ・セットの LRECL を再フォーマットされた出力レコード長に設定します。それぞれの入力レコードから複数の出力レコードを作成するために n/ を使用した場合、LRECL は最長出力レコードに設定されます。

WETHBKS の場合、出力レコードは 71 バイトの長さなので、DFSORT は LRECL を 71 に設定します。FERNBKS の場合、最初の出力レコードは 44 バイトの長さで、2 番目の出力レコードは 17 バイトの長さなので、DFSORT は LRECL を 44 に設定します。再フォーマットされた長さが、LRECL に必要な長さよりも短い場合は、c:X を使用して LRECL を c に増やすことができます。例えば、FERNBKS の LRECL を 44 ではなく 80 にしたい場合は、FERNBKS に対する OUTREC パラメーターの最後の行を次のように変更できます。

```
5:C'costs ',170,4,BI,EDIT=($IIT.TT),80:X,2/)
```

## 繰り返し

OUTFIL の REPEAT=n パラメーターを使用すると、それぞれの出力レコードを n 回繰り返すことができます。シーケンス番号を付けて BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターを指定した場合は、繰り返されるそれぞれのレコードごとにシーケンス番号は増分されますが、それ以外の場合は繰り返されるレコードは同じものです。例えば、以下のステートメントを使用すると、繰り返されたレコードを持つ 2 つの出力データ・セット (コース名だけを持つデータ・セットと、コース名およびシーケンス番号を持つデータ・セット) を作成できます。

```
OPTION COPY
INCLUDE COND=(110,5,CH,EQ,C'PSYCH')
OUTFIL FNames=OUT1,
 REPEAT=3,
 BUILD=(120,25)
OUTFIL FNames=OUT2,
 REPEAT=3,
 BUILD=(120,25,X,SEQNUM,4,ZD)
```

OUT1 として作成される結果は、次のようになります。

```
PSYCHOLOGY I
PSYCHOLOGY I
PSYCHOLOGY I
PSYCHOANALYSIS
PSYCHOANALYSIS
PSYCHOANALYSIS
```

OUT2 として作成される結果は、次のようになります。

|                |      |
|----------------|------|
| PSYCHOLOGY I   | 0001 |
| PSYCHOLOGY I   | 0002 |
| PSYCHOLOGY I   | 0003 |
| PSYCHOANALYSIS | 0004 |
| PSYCHOANALYSIS | 0005 |
| PSYCHOANALYSIS | 0006 |

IFTHEN 文節をシーケンス番号と一緒に指定する場合、シーケンス番号は、その IFTHEN 文節によって選択されたレコードのサブセットに対してのみ増分されます。例えば、次のステートメントを指定するものとします。

```
OPTION COPY
INCLUDE COND=(110,5,CH,EQ,C'PSYCH')
OUTFIL REPEAT=3,
IFTHEN=(WHEN=(120,7,CH,EQ,C'PSYCHOL'),
 BUILD=(120,25,X,SEQNUM,4,ZD)),
IFTHEN=(WHEN=NONE,BUILD=(120,25,X,SEQNUM,2,ZD))
```

SORTOUT として作成される結果は、次のようになります。

|                |      |
|----------------|------|
| PSYCHOLOGY I   | 0001 |
| PSYCHOLOGY I   | 0002 |
| PSYCHOLOGY I   | 0003 |
| PSYCHOANALYSIS | 01   |
| PSYCHOANALYSIS | 02   |
| PSYCHOANALYSIS | 03   |

---

## 分割

データ・セット内に何枚のレコードがあるかはわからないが、レコードをできるだけ均等に 2 つの出力データ・セットの間で分割したいと仮定します。OUTFIL の SPLIT パラメーターを使用すると、レコードがなくなるまで、最初のレコードを OUTPUT1 に、2 番目のレコードを OUTPUT2 に、3 番目のレコードを OUTPUT1 に、4 番目のレコードを OUTPUT2 に、といった具合に入れることができます。SPLIT は、FNAMES で指定されたデータ・セットの間で、一度に 1 枚ずつレコードを分割します。以下のステートメントは、2 つの OUTFIL データ・セットの間でレコードを分割します。

```
OPTION COPY
OUTFIL FNAMES=(OUTPUT1,OUTPUT2),SPLIT
```

入力レコードが 17 枚の場合、OUTPUT1 として作成される結果は、次のようになります。

```
Record 01
Record 03
Record 05
Record 07
Record 09
Record 11
Record 13
Record 15
Record 17
```

OUTPUT2 として作成される結果は、次のようになります。

```
Record 02
Record 04
Record 06
Record 08
```

```
Record 10
Record 12
Record 14
Record 16
```

同様に、OUTFIL の SPLITBY=n パラメーターは、FNAMES で指定されたデータ・セットの間で、一度に n 枚ずつレコードを分割します。以下のステートメントは、3 つの OUTFIL データ・セットの間で、一度に 4 枚ずつレコードを分割します。

```
OPTION COPY
OUTFIL FNAMES=(OUT1,OUT2,OUT3),SPLITBY=4
```

入力レコードが 17 枚の場合、OUT1 として作成される結果は、次のようになります。

```
Record 01
Record 02
Record 03
Record 04
Record 13
Record 14
Record 15
Record 16
```

OUT2 として作成される結果は、次のようになります。

```
Record 05
Record 06
Record 07
Record 08
Record 17
```

OUT3 として作成される結果は、次のようになります。

```
Record 09
Record 10
Record 11
Record 12
```

SPLIT と SPLITBY=n はどちらも、最後の ddname にレコードを書き込んだ後、最初の ddname から再び開始します。これにより、1 つ以上の OUTFIL データ・セットで不連続レコードが生じる場合があります。例えば、SPLITBY=4 を使用する上記の例では、OUT1 にはレコード 1-4 と 13-16 があります。各 OUTFIL データ・セットに連続レコードのみが必要な場合は、OUTFIL の SPLIT1R=n パラメーターを使用できます。このパラメーターは、各 OUTFIL データ・セットに n 個のレコードを書き込んだ後、残りのレコードを最後の OUTFIL データ・セットに書き込みます。したがって、SPLIT および SPLITBY=n は、OUTFIL データ・セット間で何度も循環しますが、SPLIT1R=n は、OUTFIL データ・セット間で 1 回だけ循環します。

次のステートメントでは、各 OUTFIL データ・セット内に連続レコードのみがあるようにレコードが分割されることが保証されます。

```
OPTION COPY
OUTFIL FNAMES=(OUTA,OUTB,OUTC),SPLIT1R=5
```

入力レコードが 17 枚の場合、OUTA として作成される結果は、次のようになります。

## OUTFIL の使用

Record 01  
Record 02  
Record 03  
Record 04  
Record 05

OUTB として作成される結果は、次のようになります。

Record 06  
Record 07  
Record 08  
Record 09  
Record 10

OUTC として作成される結果は、次のようになります。

Record 11  
Record 12  
Record 13  
Record 14  
Record 15  
Record 16  
Record 17

### これまで

これまでは、入力データ・セットの複数の同一コピーを作成する方法、出力する入力レコードを相対レコード番号によって選択し抽出する方法、出力するよう選択した入力レコードを組み込む、省略する、または保管する方法、ならびに、出力する入力レコードの繰り返し方法と分割方法を学習しました。次に、各種の報告書の作成方法を学習します。

---

## 報告書の作成: OUTFIL 対 ICETOOL

報告書を作成するには、OUTFIL ステートメントと ICETOOL の DISPLAY オペレーターの両方を使用できます。一方が行わない報告書作成機能を他方が実行しますが、一般には、この両方の違いは制御と労力にあります。OUTFIL では、報告書の外観をより多く制御できますが、情報のあらゆる断片を報告書のどこに表示させるかおよびその概観を指定する必要があるため、ユーザーには相当な労力が必要になります。ICETOOL では、情報を報告書のどこに表示させるかおよびその概観を判別する作業のほとんどがユーザーに代わって行われるので、報告書の概観はそれほど制御できません。

ICETOOL の方が OUTFIL よりも使用しやすいため、ICETOOL を報告書作成の最初的手段として選択してください。ただし、特定の報告書作成機能が必要な場合や、ICETOOL で提供されるよりも概観を制御した報告書が必要な場合は、いつでも OUTFIL を使用できます。次の節では、OUTFIL を使用した報告書の作成方法を学習します。ICETOOL の DISPLAY オペレーターを使用して報告書を作成するチュートリアルについては、133 ページの『第 11 章 ICETOOL ユーティリティの使用』を参照してください。

## OUTFIL を使用した報告書の作成

OUTFIL では、OUTFIL パラメーターの BUILD、OUTREC、OVERLAY、IFTHEN、HEADERn、TRAILERn、SECTIONS、LINES、NODETAIL、BLKCCH1、BLKCCH2、BLKCCT1、および REMOVECC を使用することによって、1 つ以上の明細報告書を作成することができます。以下の一部または全部を報告書に含めることができます。

- カバー・シート (報告書ヘッダー)
- 各ページの上部のヘッダー (ページ・ヘッダー)
- 各ページの下部のトレーラー (ページ・トレーラー)
- 各セクションの始まりのヘッダー (セクション・ヘッダー)
- 各セクションの終わりのトレーラー (セクション・トレーラー)
- 要約シート (報告書トレーラー)

報告書には、ユーザーが指定するさまざまなエレメント (現在日付、現在時刻、編集済みまたは変換済みのページ番号、文字ストリング、ブランク行など) を含めることができます。また、報告書には、入力レコードから得たさまざまなエレメント (文字フィールド、未編集、編集済み、または変換済みの数値入力フィールド、編集済みまたは変換済みのレコード・カウント、および数値入力フィールドの編集済みまたは変換済みの合計、最大、最小、および平均など) を含めることもできます。

## データ

報告書のデータは、OUTFIL 出力レコードから構成されます。

INCLUDE、OMIT、INREC、SUM および OUTREC の各ステートメントと、OUTFIL パラメーター STARTREC、SAMPLE、ENDREC、INCLUDE、OMIT、SAVE、PARSE、BUILD、OUTREC、OVERLAY、および IFTHEN を使用して、報告書に出力させるデータ・レコードとその概観を決定することができます。

以下のステートメントは、作成できる最も単純な OUTFIL 報告書を示しています。

```
OPTION COPY
OUTFIL FNAME=RPT1,LINES=10
```

LINES=10 は、ページ当たり 10 行の報告書を作成することを示します。これらのステートメントは、次の 2 ページの結果を RPT1 として作成します。

```
1Data line 01
Data line 02
Data line 03
Data line 04
Data line 05
Data line 06
Data line 07
Data line 08
Data line 09
Data line 10
1Data line 11
Data line 12
Data line 13
Data line 14
Data line 15
```

注: 下記のいくつかの例では、ポイントを説明するために小さな LINES 値を使用しています。しかし、LINES のデフォルトは 60 であり、この値がそれに近い値を使用するのが普通です。

報告書の場合、OUTFIL は、各出力行の最初のバイトに ANSI 紙送り制御文字を入れ、その行で取るべき処置をプリンターに指示します。Data line 01 および Data line 11 の '1' は、これらの行がそれぞれ新規ページで始まることをプリンターに指示します。これは、DFSORT が LINES=10 をプリンターのページ当たり行数 10 に変換する方法です。DFSORT が OUTFIL 報告書に使用するその他の ANSI 紙送り制御文字には、次のものがあります。すなわち、シングルのスペースを表すブランク (出力行の前にブランク行なし)、ダブルスペースを表す '0' (出力行の前に 1 ブランク行)、およびトリプル・スペースを表す '-' (出力行の前に 2 ブランク行) です。

ディスプレイで報告書を表示した場合、ANSI 紙送り制御は意味を持たず、レコードに実際に含まれていても、通常は表示されません。出力レコード内で ANSI 紙送り制御文字が必要ない場合は、OUTFIL の REMOVECC パラメーターを使用すれば除去できます (後で詳しく説明します)。

一般に、報告書のデータ・レコードを再フォーマットするには、PARSE、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターを使用します。SORT.BRANCH データ・セットから、各ブランチの売り上げと利益または損益を示す、1 ページ当たり 9 行の報告書を印刷したいとします。以下のステートメントを使用すると、読みやすい形式の売り上げフィールドおよび利益フィールドと一緒に、各ブランチをソート順に示すことができます。

```
SORT FIELDS=(1,15,CH,A)
OUTFIL FNames=RPT2,LINES=9,
OUTREC=(1,15,X,
 22,6,PD,EDIT=(SIII,IIT),SIGNS=(,-),X,
 28,6,PD,EDIT=(SIII,IIT),SIGNS=(,-))
```

RPT2 として作成される 2 ページの結果は、次のようになります。

|               |        |        |
|---------------|--------|--------|
| 1Aspen        | 25,800 | 5,200  |
| Boulder       | 33,866 | 7,351  |
| Denver        | 31,876 | 6,288  |
| Fort Collins  | 12,300 | -2,863 |
| Los Angeles   | 22,530 | -4,278 |
| Morgan Hill   | 18,200 | 3,271  |
| Sacramento    | 42,726 | 8,276  |
| San Diego     | 32,940 | 8,275  |
| San Francisco | 42,820 | 6,832  |
| 1San Jose     | 27,225 | 8,264  |
| Sunnyvale     | 16,152 | -978   |
| Vail          | 23,202 | 5,027  |

## ヘッダー

上記の RPT2 の出力は報告書のように見えませんが、OUTFIL の HEADER2 パラメーターを使用してページ見出しを追加することにより修正できます。このパラメーター (および後述の HEADER1 と HEADER3 パラメーター) を使用すると、文字ストリング、16 進数ストリング、入力フィールド、現在日付、現在時刻、ページ番号、およびブランク行によって複数行見出しを指定できます。

以下のステートメントは、ページ見出し付きの報告書を作成します。

```

SORT FIELDS=(1,15,CH,A)
OUTFIL FNames=RPT3,LINES=15,
 HEADER2=(/,3:'Branch Revenue Report',
 30:'Page',PAGE,45:DATE=(MD4-),2/,
 3:'Branch',25:'Revenue',50:'Profit',/,
 3:'-----',25:'-----',50:'-----'),
 OUTREC=(3:1,15,X,
 25:22,6,PD,EDIT=(SIII,IIT),SIGNS=(,-),X,
 50:28,6,PD,EDIT=(SIII,IIT),SIGNS=(,-))

```

HEADER2 パラメーターのすべてのエレメントは、PAGE および DATE=(MD4-) パラメーターを除いて、前の OUTREC ステートメントの説明のところですでに紹介しました。

PAGE は、先頭ページの 1 から始まり、その後の各ページごとに 1 ずつ増分するページ番号に先行ブランクを付けて、6 文字のページ番号を印刷するように、DFSORT に指示します。3 ページ目の場合、PAGE により ' 3' になります。先行ゼロを付けた 3 文字のページ番号を印刷したい場合は、PAGE=(EDIT=(TTT)) または PAGE=(M11,LENGTH=3) または PAGE=(TO=ZD,LENGTH=3) を使用できます。10 ページ目の場合、これらはすべて '010' になります。変換および編集について詳しくは、54 ページの『数値フィールドから異なるフォーマットへの変換』および 55 ページの『数値フィールドの編集』を参照してください。

DATE=(MD4-) は、mm-dd-yyyy 形式の日付を印刷するよう DFSORT に指示します。ヘッダーおよびトレーラーで印刷できる日付形式の詳細については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターの場合のように、ヘッダー (HEADERn パラメーター) またはトレーラー (TRAILERn パラメーター) の文字ストリングを C で始める必要はありませんが、要望があればそうすることもできます。

ヘッダーまたはトレーラーの始まりまたは終わりで n/ を使用すると、n ブランク行が印刷されます。ヘッダーまたはトレーラーの途中で n/ を使用すると、n-1 ブランク行が印刷されます。

RPT3 として作成される 2 ページの結果は、次のようになります。

```

1
0 Branch Revenue Report Page 1 04-16-2005
 Branch Revenue Profit

 Aspen 25,800 5,200
 Boulder 33,866 7,351
 Denver 31,876 6,288
 Fort Collins 12,300 -2,863
 Los Angeles 22,530 -4,278
 Morgan Hill 18,200 3,271
 Sacramento 42,726 8,276
 San Diego 32,940 8,275
 San Francisco 42,820 6,832
 San Jose 27,225 8,264
1
0 Branch Revenue Report Page 2 04-16-2005
 Branch Revenue Profit

 Sunnyvale 16,152 -978
 Vail 23,202 5,027

```



この報告書を印刷した場合、ANSI 紙送り制御文字は、最初のバイトが '1' のときはページ替えし、最初のバイトが '0' のときは行の前にブランク行を挿入するように、プリンターに指示します。したがって、ヘッダーの 'Branch Revenue Report' 行と "Branch" 行の間にはブランク行が入り、これは OUTREC パラメーターの 2/ に相当します。しかし、この報告書をディスプレイで表示した場合、ANSI 紙送り制御文字は無視されるので、ディスプレイでは、'0' 行はその前にブランク行を持つことはありません。報告書を印刷またはディスプレイで表示するときにブランク行を強制的に出力させるには、HEADER2 の 2 番目の行を次のように変更できます。

```
30:'Page',PAGE,45:DATE=(MD4-),/,X,/,
```

これは、プリンターのためのブランク行を挿入するのに、'0' の ANSI 紙送り制御文字を使用する代わりに追加のブランク出力行を書き込むように DFSORT に強制します。プリンターまたはディスプレイでは、ブランク行が出力されます。

OUTFIL の HEADER1 パラメーターは HEADER2 パラメーターによく似ています。ただし HEADER1 は、データの先頭ページの前の別個のページに報告書見出しを作成します。次の HEADER1 パラメーターを OUTFIL ステートメントに追加できます。

```
HEADER1=(20:'Cover sheet for Branch Revenue Report',3/,
 20:'Printed on ',DATE=(MD4/),' at ',TIME),
```

この別個のページは、RPT3 の報告書の先頭、前に示された他の 2 ページの前に印刷されます。したがって、報告書の最初の 2 ページは次のように開始します。

```
1 Cover sheet for Branch Revenue Report
- Printed on 04/16/2005 at 16:42:20

1
Branch Revenue Report Page 1 04-16-2005
...
```

報告書見出しには数行しかないので、最初のページ見出しのために新規ページにページ替えすることに注意してください。報告書見出しと同じページでページ見出しを開始したい場合は、OUTFIL の BLKCCH2 パラメーターを使用できます。このパラメーターは、最初のページ見出しの先頭行の「1」をブランクで置き換えて、ページ替えを避けることを、DFSORT に指示します。BLKCCH2 を使用すると、報告書の先頭ページは次のように開始します。

```
1 Cover sheet for Branch Revenue Report
- Printed on 04/16/2005 at 16:42:20

Branch Revenue Report Page 1 04-16-2005
...
```

また、OUTFIL の BLKCCH1 パラメーターを使用すると、報告書見出しの先頭行の「1」をブランクで置き換えて、報告書見出しのページ替えを避けることを、DFSORT に指示することもできます。

以下に、HEADER1、HEADER2、および BLKCCH2 を使用する報告書のサンプル JCL ステートメントと制御ステートメントを示しています。

```
//HDRPT JOB A492,PROGRAMMER
//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.BRANCH,DISP=SHR
```



```
//RPT3 DD SYSOUT=A
//SYSIN DD *
SORT FIELDS=(1,15,CH,A)
OUTFIL FNAMES=RPT3,LINES=15,BLKCCCH2,
 HEADER1=(20:'Cover sheet for Branch Revenue Report',3/,
 20:'Printed on ',DATE=(MD4/),' at ',TIME),
 HEADER2=(/,3:'Branch Revenue Report',
 30:'Page',PAGE,45:DATE=(MD4-),2/,
 3:'Branch ',25:' Revenue',50:' Profit',/,
 3:'-----',25:'-----',50:'-----'),
 OUTREC=(3:1,15,X,
 25:22,6,PD,EDIT=(SIII,IIT),SIGNS=(,-),X,
 50:28,6,PD,EDIT=(SIII,IIT),SIGNS=(,-))
/*
```

もちろん、SYSOUT=A の代わりに、RPT3 の一時データ・セットまたは永続データ・セットを使用できます。以下に例を示します。

```
//RPT3 DD DSN=88MYRPT,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=SYSDA
```

## トレーラーおよび統計

OUTFIL の TRAILER1 (報告書トレーラー) および TRAILER2 (ページ・トレーラー) パラメーターを使用して、報告書にトレーラーを追加することができます。必要なヘッダーとトレーラーの任意の組み合わせを使用できます。

HEADERn パラメーターと同様に、TRAILER1 および TRAILER2 パラメーター (および後述の TRAILER3 パラメーター) を使用すると、文字ストリング、16 進数ストリング、入力フィールド、現在日付、現在時刻、ページ番号、およびブランク行によって複数行見出しを指定することができます。さらに、編集済みまたは変換済みのレコード・カウント、および数値フィールドの編集済みまたは変換済みの合計、最大、最小、および平均をトレーラーに含めることもできます。

以下のステートメントは、全体のレコード・カウントと各ブランチからもたらされた売り上げの各種の全体的な統計を別個の報告書トレーラー・ページに収めた、報告書を作成します。

```
SORT FIELDS=(1,15,CH,A)
OUTFIL FNAMES=RPT4,
 HEADER2=(1:'BRANCH',18:'REVENUE',28:'PROFIT'),
 OUTREC=(1,15,X,
 22,6,PD,EDIT=(SIII,IIT),SIGNS=(,-),X,
 28,6,PD,EDIT=(SIII,IIT),SIGNS=(,-),
 50:X),
 TRAILER1=(2/,
 3:'Overall results for branches on ',DATE=(MD4/),' :',2/,
 3:COUNT=(EDIT=(IIT)), ' branches are included',2/,
 5:'Total revenue = ',
 TOT=(22,6,PD,M12,LENGTH=10),/,
 5:'Average revenue = ',
 AVG=(22,6,PD,M12,LENGTH=10),/,
 5:'Lowest revenue = ',
 MIN=(22,6,PD,M12,LENGTH=10),/,
 5:'Highest revenue = ',
 MAX=(22,6,PD,M12,LENGTH=10))
```

OUTFIL 報告書の場合、DFSORT は、ヘッダー・レコードまたはトレーラー・レコードがデータ・レコードよりも長いと終了します。この場合、DATE=(MD4/) の付いた TRAILER1 行は 46 バイトの長さなので、データ・レコードは少なくとも 46 バイトの長さでなければなりません。OUTREC パラメーターで設定されたデータ長

## OUTFIL の使用

を 42 バイトから 50 バイトに増やすには、示されたように OUTREC パラメータの終わりに 50:X を指定することができます。

COUNT=(EDIT=(IIT)) は、データ・レコードのカウントを、先行ゼロが抑止された 3 桁として示します。TOT=(22,6,PD,M12,LENGTH=10) は、合計売り上げを、10 バイトの M12 パターン (すなわち、SI,III,IIT) として示します。同様に、AVG、MIN、および MAX は、それぞれ平均、最小、および最大を、10 バイトの M12 パターンとして示します。変換および編集について詳しくは、54 ページの『数値フィールドから異なるフォーマットへの変換』および 55 ページの『数値フィールドの編集』を参照してください。

LINES=n が指定されていないので、デフォルトのページ当たり行数 60 が使用されます。

RPT4 として作成される 2 ページの結果は、次のようになります。

| 1BRANCH       | REVENUE | PROFIT |
|---------------|---------|--------|
| Aspen         | 25,800  | 5,200  |
| Boulder       | 33,866  | 7,351  |
| Denver        | 31,876  | 6,288  |
| Fort Collins  | 12,300  | -2,863 |
| Los Angeles   | 22,530  | -4,278 |
| Morgan Hill   | 18,200  | 3,271  |
| Sacramento    | 42,726  | 8,276  |
| San Diego     | 32,940  | 8,275  |
| San Francisco | 42,820  | 6,832  |
| San Jose      | 27,225  | 8,264  |
| Sunnyvale     | 16,152  | -978   |
| Vail          | 23,202  | 5,027  |

1  
0 Overall results for branches on 04/16/2005:  
0 12 branches are included  
0 Total revenue = 329,637  
Average revenue = 27,469  
Lowest revenue = 12,300  
Highest revenue = 42,820

2 番目のページは、TRAILER1 に対応して作成される報告書トレーラー・ページであり、これは通常、新規ページで始まります。ただし、OUTFIL の BLKCCT1 パラメーターを使用すると、報告書トレーラー用の強制改ページを避けることができます。例えば、OUTFIL ステートメントの先頭行を次のように変更とします。

OUTFIL FNAMES=RPT4,BLKCT1,

RPT4 として作成される 1 ページの結果は、次のようになります。

| 1BRANCH       | REVENUE | PROFIT |
|---------------|---------|--------|
| Aspen         | 25,800  | 5,200  |
| Boulder       | 33,866  | 7,351  |
| Denver        | 31,876  | 6,288  |
| Fort Collins  | 12,300  | -2,863 |
| Los Angeles   | 22,530  | -4,278 |
| Morgan Hill   | 18,200  | 3,271  |
| Sacramento    | 42,726  | 8,276  |
| San Diego     | 32,940  | 8,275  |
| San Francisco | 42,820  | 6,832  |
| San Jose      | 27,225  | 8,264  |
| Sunnyvale     | 16,152  | -978   |
| Vail          | 23,202  | 5,027  |

0 Overall results for branches on 04/16/2005:  
0 12 branches are included

```

0 Total revenue = 329,637
 Average revenue = 27,469
 Lowest revenue = 12,300
 Highest revenue = 42,820

```

## データまたは紙送り制御文字がない場合

報告書に、実際のデータ行ではなく全体統計ページを含めたい場合には、OUTFIL の NODETAIL パラメーターを OUTFIL ステートメントに追加することができます。NODETAIL は、データ・レコードを報告書データ・セットに書き込むのではなく通常の方法で処理するよう DFSORT に指示します。

報告書では ANSI 紙送り制御文字が不要であれば、OUTFIL の REMOVECC パラメーターを OUTFIL ステートメントに追加できます。REMOVECC は、各レコードの最初のバイトから紙送り制御文字を除去するよう DFSORT に指示します。その結果、データは、報告書の桁 2 ではなく桁 1 から始まります。

2 つの出力データ・セットを作成したいと仮定します。最初のデータ・セットは、入力データ・セット内のレコードの合計カウントを示す単一出力レコードを持っていない必要があります。2 番目のデータ・セットは、今日の日付を ddmmmyyyy の形式で示す単一出力レコードを持っていない必要があります。以下のステートメントを使用すると、この 2 つの出力データ・セットを作成できます。

```

OPTION COPY
OUTFIL FNames=BRANCHCT,NODETAIL,REMOVECC,
 TRAILER1=(COUNT=(TO=ZD,LENGTH=10))
OUTFIL FNames=CURDT,NODETAIL,REMOVECC,
 HEADER1=(DATENS=(DM4))

```

入力データ・セットのレコード数が 12 の場合、BRANCHCT として作成される単一レコードは次のようになります (カウントは桁 1-10 に示されます)。

```
00000000012
```

ジョブが 2004 年 7 月 15 日に実行された場合、CURDT として作成される単一レコードは次のようになります (ddmmmyyyy 形式の日付は桁 1-8 に示されます)。

```
15072004
```

## ヘッダーとトレーラーに関する VB データ・セットの考慮事項

HEADERn または TRAILERn パラメーターで出力桁を指定するときは、BUILD、OUTREC、または IFTHEN OUTREC パラメーターの場合のように、RDW のための 4 を桁に追加しないでください。次の OUTFIL ステートメントを例にとります。

```

OUTFIL REMOVECC,
 HEADER2=(1:C'PAGE HEADER'),
 OUTREC=(1,4,5:C'OUTREC FOR DATA')

```

'PAGE HEADER' スtringと 'OUTREC FOR DATA' スtringは共に、それぞれの出力レコードの位置 5 から始まります。HEADER2 の場合は、RDW がヘッダー・レコードに現れる場合でも、RDW のための 4 を出力桁に追加してはならないので、1: を使用します。OUTREC の場合は、RDW のための 4 を出力桁に追加する必要があるため、5: を使用します。この違いだけに注意を払い、それに応じて出力桁を指定してください。

HEADERn または TRAILERn パラメーターで入力フィールドを指定するときは、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターの場合と同様、RDW のための 4 を開始位置に追加する必要があります。しかし、HEADERn または TRAILERn パラメーターには、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターの場合のように、入力レコードの可変部分 (すなわち、長さを持たない位置) は指定できません。

## セクション

コンピューター (Computer) 学部の本に関する情報、英文学 (English) 学部の本に関する情報、および歴史学 (History) 学部の本に関する情報をそれぞれ 1 ページに印刷したいとします。それぞれの学部の本ごとに、そのコース番号 (Course Number)、在庫数 (Number in Stock)、そのときまでの販売数 (Number Sold Y-to-D)、および価格 (Price) を示し、さらに、3 つの数値 (BI) フィールドのそれぞれについての学部別合計を示すことにします。以下のステートメントを使用すると、このすべての情報を持つ報告書を作成できます。

```
INCLUDE COND=(110,4,SS,EQ,C'COMP,ENGL,HIST')
INREC FIELDS=(1:110,4, Course Department
CHANGE=(16,
C'COMP',C'Computer Science',
C'ENGL',C'English',
C'HIST',C'History'),
17:115,5, Course Number
22:162,4, Number in Stock
26:166,4, Number Sold Y-to-D
30:170,4, Price
50:X) Ensure data length greater than header/trailer length
SORT FIELDS=(1,16,CH,A,17,5,CH,A)
OUTFIL FAMES=RPT5,
SECTIONS=(1,16,SKIP=P,
HEADER3=(3:X,/,
3:'Department: ',1,16,/,X,/,
3:'Number',12:'In Stock',23:'Sold YTD',34:' Price',/,
3:'-----',12:'-----',23:'-----',34:'-----'),
TRAILER3=(3:'=====',12:'=====',23:'=====',34:'====='),/,
3:'Totals',
15:TOT=(22,4,BI,EDIT=(IIIIIT)),
26:TOT=(26,4,BI,EDIT=(IIIIIT)),
34:TOT=(30,4,BI,EDIT=(IIIT.TT))),
OUTREC=(3:17,5,
15:22,4,BI,EDIT=(IIIIIT),
26:26,4,BI,EDIT=(IIIIIT),
34:30,4,BI,EDIT=(IIIT.TT))
```

INCLUDE ステートメントおよび INREC ステートメントは、ソートの前に不要なレコードおよびフィールドを除去します。また、INREC ステートメントは、学部 ID (COMP、ENGL、HIST) をもっと読みやすいストリングに変更します。以前に学習したように、INREC ステートメントはさまざまなフィールドの開始位置を変更するので、INREC の後で処理されるステートメント (この場合は SORT と OUTFIL) では新規の位置を使用する必要があります。

OUTFIL の SECTION パラメーターは、報告書を学部コースによってセクションに分割するために使用されます。1,16 は、桁 1-16 の学部値が変わるたびに新規セクションを開始するよう DFSORT に指示します。SORT ステートメントは学部フィールドに基づいてソートし、それぞれの学部値のすべてのレコードを一緒に集めるので、それぞれの学部値ごとに 1 つのセクションだけが作成されます。レコードがまだセクション・フィールドによってソートされていない場合は、SORT ステート

メントを使用してレコードをセクション・フィールドでソートしてください。他のフィールドでも同様にソートできますが、最初のソート・フィールドはセクション・フィールドでなければなりません。

SKIP=P は、各セクションを新規ページで開始するよう DFSORT に指示します。あるいは、セクションを同じページに出力させたい場合は、SKIP=nL を使用できます (n は各セクション間の行数です)。

HEADER3 は、HEADER1 と HEADER2 がそれぞれ報告書ヘッダーとページ・ヘッダーを作成する同じ方法で、セクション・ヘッダーを作成します。同様に、TRAILER3 は、TRAILER1 と TRAILER2 がそれぞれ報告書トレーラーとページ・トレーラーを作成する同じ方法で、セクション・トレーラーを作成します。

OUTFIL の OUTREC パラメーターは、報告書のデータ・レコードを再フォーマットするために使用されます。

RPT5 として作成される 3 ページの結果は、次のようになります。

```
1
 Department: Computer Science

 Number In Stock Sold YTD Price
 ----- -
00032 5 29 26.00
00032 6 27 3.60
00032 20 26 18.99
00103 4 23 31.95
00205 10 10 21.99
=====
Totals 45 115 102.53
```

```
1
 Department: English

 Number In Stock Sold YTD Price
 ----- -
10054 10 9 15.20
10347 7 19 6.25
10347 13 32 14.50
10856 1 26 4.50
10856 2 32 5.95
=====
Totals 33 118 46.40
```

```
1
 Department: History

 Number In Stock Sold YTD Price
 ----- -
50420 15 9 7.95
50521 14 17 12.00
50632 23 21 17.90
=====
Totals 52 47 37.85
```

データ行ではなく、それぞれの学部の合計を示す 2 番目の報告書を作成したいとします。これを行うには、NODETAIL (データなし)、HEADER2 (ページ・ヘッダー)、SECTIONS (セクション)、および TRAILER3 (セクション・トレーラー) パラメーターを指定した 2 番目の OUTFIL ステートメントを使用することができます。この場合、データ行は印刷されないで、OUTREC パラメーターを使用して再フォーマットする必要はありません。この 2 つの報告書を作成する完全なジョブの JCL および制御ステートメントは、次のようになります。

## OUTFIL の使用

```
//SCTNRPT JOB A492,PROGRAMMER
//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//RPT5 DD SYSOUT=A
//RPT6 DD SYSOUT=A
//SYSIN DD *
INCLUDE COND=(110,4,SS,EQ,C'COMP,ENGL,HIST')
INREC FIELDS=(1:110,4, Course Department
CHANGE=(16,
C'COMP',C'Computer Science',
C'ENGL',C'English',
C'HIST',C'History'),
17:115,5, Course Number
22:162,4, Number in Stock
26:166,4, Number Sold Y-to-D
30:170,4, Price
50:X) Ensure data length greater than header/trailer length
SORT FIELDS=(1,16,CH,A,17,5,CH,A)
OUTFIL FAMES=RPT5,
SECTIONS=(1,16,SKIP=P,
HEADER3=(3:X,/,
3:'Department: ',1,16,/,X,/,
3:'Number',12:'In Stock',23:'Sold YTD',34:' Price',/,
3:'-----',12:'-----',23:'-----',34:'-----'),
TRAILER3=(3:'=====',12:'=====',23:'=====',34:'=====',/,
3:'Totals',
15:TOT=(22,4,BI,EDIT=(IIIIIT)),
26:TOT=(26,4,BI,EDIT=(IIIIIT)),
34:TOT=(30,4,BI,EDIT=(IIIT.TT)))),
OUTREC=(3:17,5,
15:22,4,BI,EDIT=(IIIIIT),
26:26,4,BI,EDIT=(IIIIIT),
34:30,4,BI,EDIT=(IIIT.TT))
OUTFIL FAMES=RPT6,
NODETAIL,
HEADER2=(3:'Department ',22:'In Stock',
31:'Sold YTD',42:' Price',/,
3:'-----',22:'-----',
31:'-----',42:'-----'),
SECTIONS=(1,16,SKIP=0L,
TRAILER3=(3:1,16,
25:TOT=(22,4,BI,EDIT=(IIIIIT)),
34:TOT=(26,4,BI,EDIT=(IIIIIT)),
42:TOT=(30,4,BI,EDIT=(IIIT.TT))))
/*
```

RPT5 として作成される 3 ページの報告書は、前に示されたとおりです。 RPT6 として作成される 1 ページの結果は、次のようになります。

| 1 | Department       | In Stock | Sold YTD | Price  |
|---|------------------|----------|----------|--------|
|   | -----            | -----    | -----    | -----  |
|   | Computer Science | 45       | 115      | 102.53 |
|   | English          | 33       | 118      | 46.40  |
|   | History          | 52       | 47       | 37.85  |

### これまで

これまでは、入力データ・セットの複数の同一コピーを作成する方法、出力する入力レコードを相対レコード番号によって選択し抽出する方法、出力するよう選択した入力レコードを組み込む、省略する、または保管する方法、出力する入力レコードの再フォーマット方法、繰り返し方法、および分割方法、ならびに、さまざまなタイプの報告書の作成方法を学習しました。次に、固定長レコードと可変長レコードの間の変換について学習します。

## FB から VB への変換

OUTFIL の FTOV パラメーターを使用すると、FB データ・セットを VB データ・セットに変換できます。それぞれの VB 出力レコードは、4 バイトの RDW とその後に FB 入力レコードからの対応するデータを持っているので、RDW 内の長さは FB レコードに 4 を加えた長さになります。

以下の JCL および DFSORT 制御ステートメントは、ブックスストア・データ・セットのレコードを FB から VB に変換します。

```
//FBVB JOB A92,PROGRAMMER
//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//VBOUT DD DSN=A123456.SORT.VSAMP,DISP=(NEW,CATLG,DELETE),
// UNIT=3390,SPACE=(CYL,(5,5))
//SYSIN DD *
OPTION COPY
OUTFIL FNames=VBOUT,FTOV
/*
```

SORT.SAMPIN の LRECL は 173 バイトなので、SORT.VSAMP のそれぞれの VB レコードは 177 バイトになり (173 の FB レコード長に RDW の 4 を加えた長さ)、SORT.VSAMP には 177 の LRECL が与えられます。

また、OUTFIL ステートメントで FTOV と一緒に PARSE、BUILD、OUTREC、OVERLAY、または IFTHEN パラメーターを使用することもできます。再フォーマットのすべての機能が使用可能です (入力フィールド、ストリング、編集、など)。FTOV では、FB レコード (RDW なし) の場合に指定する BUILD、OUTREC、OVERLAY または IFTHEN パラメーターの、入力位置と出力桁を指定します。DFSORT は、FB レコードが再フォーマットされた後に RDW を追加します。

FTOV と BUILD を一緒に使用した例を、以下に示します。

```
OUTFIL FTOV,BUILD=(1:120,25,32:C'in ',110,5)
```

VB 出力レコードは次のようになります。

```
Positions 1-2: Length (in RDW) = hex 002B = 43
Positions 3-4: Zeros (in RDW) = hex 0000 = 0
Positions 5-29: Input positions 120-144
Positions 30-35: Blanks
Positions 36-38: 'in '
Positions 39-43: Input positions 110-114
```



## OUTFIL の使用

それぞれの VB 出力レコードは 43 バイトの長さであり、出力データ・セットには 43 の LRECL が与えられます。

前の例では、VB 出力レコードはすべて同じ長さです。OUTFIL の VLTRIM パラメーターを使用すると、FB 入力レコードからいろいろな長さの VB 出力レコードを作成することができます。LRECL=20 の FB 入力データ・セットがあり、次のような 20 バイトの各レコードがあるとします。

```
ABC
ABCDEF
AC
ABCDEFGHI
```

以下のステートメントは、それぞれ 20 バイトの FB 入力レコードを 24 バイトの VB 出力レコードに変更し、VB 出力データ・セットに対して LRECL=24 を設定します。

```
OPTION COPY
OUTFIL FTOV
```

VB 出力レコードは次のようになります。

| Length |   | X'0000' |   | Data      |    |
|--------|---|---------|---|-----------|----|
| 1      | 2 | 3       | 4 | 5         | 24 |
| 24     |   | 0       |   | ABC       |    |
| 24     |   | 0       |   | ABCDEF    |    |
| 24     |   | 0       |   | AC        |    |
| 24     |   | 0       |   | ABCDEFGHI |    |

それぞれの VB 出力レコードの終わりから末尾ブランクを除去するには、VLTRIM=C' ' (または VLTRIM=X'40') を次のように使用します。

```
OUTFIL FTOV,VLTRIM=C' '
```

結果の VB 出力データ・セットには、最大レコード長が 24 であることを示す LRECL=24 がやはり設定されますが、VB 出力レコードは右側にブランクを埋め込まれません。すなわち、次のようないろいろなレコード長になります。

| Length |   | X'0000' |   | Data      |    |
|--------|---|---------|---|-----------|----|
| 1      | 2 | 3       | 4 | 5         | 24 |
| 7      |   | 0       |   | ABC       |    |
| 10     |   | 0       |   | ABCDEF    |    |
| 6      |   | 0       |   | AC        |    |
| 13     |   | 0       |   | ABCDEFGHI |    |

VLTRIM には任意の文字または 16 進バイト値を使用できます。例えば、VLTRIM=C'\*' は末尾のアスタリスクを除去し、VLTRIM=X'00' は末尾の 2 進ゼロを除去します。



## VB から FB への変換

OUTFIL の VTOF および BUILD または OUTREC パラメーターを使用することにより、VB データ・セットを FB データ・セットに変換できます。

PARSE、BUILD、または OUTREC のすべての機能が使用可能です (入力フィールド、ストリング、編集、など)。VTOF では、VB 入力レコード (RDW 付き) の入力位置と FB 出力レコード (RDW なし) の出力桁を指定します。DFSORT は、FB 出力レコードには RDW を含めません。

**重要:** VTOF と一緒に OVERLAY または IFTHEN を指定することはできません。

以下の JCL および DFSORT 制御ステートメントは、LRECL=104 の VB データ・セットを LRECL=100 の FB データ・セットに変換します。

```
//VFBFB JOB A92,PROGRAMMER
//S1 EXEC PGM=SORT
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=A123456.VBIN,DISP=SHR
//FBOUT DD DSN=A123456.FBOUT,DISP=(NEW,CATLG,DELETE),
// UNIT=3390,SPACE=(CYL,(5,5))
//SYSIN DD *
 OPTION COPY
 OUTFIL FNames=FBOUT,VTOF,OUTREC=(5,100)
/*
```

VB 入力レコードの位置 5 (RDW の後) から始まるデータの最高 100 バイトまでが、FB 出力レコードの位置 1 から出力されます。FB 出力レコードはすべて 100 バイトの長さになります。デフォルトでは、VB 入力レコードが 100 データ・バイトよりも短い場合、DFSORT は、100 バイトになるまで出力データの右側にブランクを埋め込みます。しかし、OUTFIL の VLFILL パラメーターを使用すれば、埋め込み文字を望みの任意の文字に変更できます。例えば、VLFILL=C'\*' はアスタリスクで埋め込み、VLFILL=X'00' は 2 進ゼロで埋め込みます。

```
 SORT FIELDS=(25,10,CH,A)
* Pad short fields on the right with blanks (default is VLFILL=C' ')
 OUTFIL FNames=PADB,VTOF,
 OUTREC=(41,40,C'Blank padding',11,20)
* Pad short fields on the right with asterisks
 OUTFIL FNames=PADA,VTOF,
 OUTREC=(5,60,C'Asterisk padding',61,40),VLFILL=C'*.
* Pad short fields on the right with binary zeros
 OUTFIL FNames=PADZ,VTOF,
 OUTREC=(21,60,C'Binary zero padding'),VLFILL=X'00'
```

LRECL=73 is set for the PADB output data set, which has 73 byte FB records. Short input fields are padded with blanks on the right as needed.

LRECL=116 is set for the PADA output data set, which has 116 byte FB records. Short input fields are padded with asterisks on the right as needed.

LRECL=79 is set for the PADZ output data set, which has 79 byte FB records. Short input fields are padded with binary zeros on the right as needed.

### 要約

本章では、OUTFIL ステートメントとさまざまなパラメーターを使用して複数の出力データ・セットおよび報告書を作成するいろいろな方法を説明しました。

次の章では、プログラムから DFSORT を呼び出す方法を紹介します。

---

## 第 8 章 プログラムからの DFSORT の呼び出し

本章には、プログラミング・インターフェース情報を記載してあります。

PGM=SORT または PGM=ICEMAN によって直接 DFSORT プログラム制御ステートメントを処理するほか、DFSORT は、COBOL、PL/I、アセンブラー、またはその他のプログラムからも呼び出すことができます。本章では、COBOL を使用したソート・マージおよび PL/I を使用したソートを中心に説明します。本章の例では、COBOL 環境が使用可能であると想定しています。

これらの言語を使用する際の制約事項およびアセンブラー・プログラムからの DFSORT の呼び出しについては、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

---

### 制御ステートメントの受け渡し

COBOL または PL/I を使用して DFSORT を呼び出すと、これらのプログラム・プロダクトがユーザーに代わって、SORT または MERGE 制御ステートメントと RECORD 制御ステートメントを作成し受け渡します。

INCLUDE、OMIT、SUM、INREC、OUTREC、および OUTFIL のようなその他の DFSORT 制御ステートメントを、COBOL または PL/I プログラムで使用することができます。これらの DFSORT ステートメントには、SYSIN データ・セットを使用するのと同じ方法で、SORTCNTL または DFSPARM データ・セットを使用してください。例えば、SORTCNTL DD ステートメントを使用すると、英文学 (English) 学部の本だけを選択する INCLUDE 制御ステートメントを受け渡すことができます。

```
//EXAMP JOB A492,PROGRAMMER
.
.
.
//SORTCNTL DD *
 INCLUDE COND=(110,5,CH,EQ,C'ENGL')
/*
```

DFSORT を直接呼び出す、すなわち PGM=SORT または PGM=ICEMAN によって呼び出すときは、SYSIN データ・セットまたは DFSPARM データ・セットのいずれかまたは両方を DFSORT 制御ステートメントに使用することができます。

DFSORT をプログラムから呼び出すときは、SORTCNTL データ・セットまたは DFSPARM データ・セットのいずれかまたは両方を DFSORT 制御ステートメントに対して使用することができます。

COBOL で DFSORT を使用する場合は、SORT-CONTROL および SORT-RETURN 特殊レジスターを熟知している必要があります。詳細については、ご使用のサイトで利用可能なコンパイラー・バージョンに関する「COBOL プログラマーズ・ガイド」を参照してください。

## COBOL プログラムからの DFSORT の呼び出し

COBOL プログラムから DFSORT を呼び出すには、COBOL ステートメントの SORT と MERGE を使用します。この節では、COBOL の SORT および MERGE ステートメントを使用するサンプル・プログラムを示します。詳細については、ご使用のサイトで利用可能なコンパイラ・バージョンに関する「COBOL プログラマーズ・ガイド」を参照してください。

### レコードのソート

117 ページの図 2 のサンプル COBOL プログラムは、DFSORT を呼び出して、ブックストア・マスター・ファイル (MASTER-FILE) をタイトルによって昇順にソートします。ソート済みマスター・ファイルは SORTED-MASTER-FILE に書き込まれます。

以下は、サンプル COBOL プログラムを呼び出す JCL です。

```
//EXAMP JOB A492,PROGRAMMER
//BOOKS EXEC PGM=COBOLPGM
//STEPLIB DD DSN=USER.PGMLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//MASTIN DD DSN=A123456.MASTER,DISP=OLD
//MASTOUT DD DSN=A123456.OUTB,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA
//PRINTFL DD SYSOUT=A
```

PGM=SORT または PGM=ICEMAN と共に JCL EXEC ステートメントを使用することによって DFSORT を実行するための JCL (19 ページの『データ・セットを直接ソートするための JCL』を参照) と対比した場合、上記の JCL には次のような違いがあります。

- EXEC ステートメント上のプログラム名は、COBOL プログラムのプログラム名です。
- STEPLIB DD ステートメントは、COBOL プログラムを含むライブラリーを定義します。
- 入力ファイルに対する DD ステートメントの名前が SORTIN である必要はありません。
- 出力ファイルに対する DD ステートメントの名前が SORTOUT である必要はありません。

ソートの制御フィールドと順序は、SORT 制御ステートメントによって指定するのではなく、COBOL プログラム自体の中で指定することに注意してください。117 ページの図 2 は、サンプル COBOL プログラムを示しています。

---

```

IDENTIFICATION DIVISION.
PROGRAM-ID.
 COBOLPGM.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
 SELECT SD-FILE ASSIGN TO
 DUMMYNM.
 SELECT MASTER-FILE ASSIGN TO
 MASTIN.
 SELECT SORTED-MASTER-FILE ASSIGN TO
 MASTOUT.
 SELECT PRINT-FILE ASSIGN TO
 PRINTFL.
DATA DIVISION.
FILE SECTION.
SD SD-FILE
 DATA RECORD IS SD-RECORD.
01 SD-RECORD.
 05 TITLE-IN PICTURE X(75).
 05 AUTH-LN-IN PICTURE X(15).
 05 AUTH-FN-IN PICTURE X(15).
 05 PUB-IN PICTURE X(4).
 05 COUR-DEPT-IN PICTURE X(5).
 05 COUR-NO-IN PICTURE X(5).
 05 COUR-NAM-IN PICTURE X(25).
 05 INST-LN-IN PICTURE X(15).
 05 INST-INIT-IN PICTURE X(2).
 05 NO-STOCK-IN PICTURE 9(8) BINARY.
 05 NO-SOLD-IN PICTURE 9(8) BINARY.
 05 PRICE-IN PICTURE 9(8) BINARY.

```

---

図 2. SORT コマンドを持つサンプル COBOL プログラム (1/2)

```
FD MASTER-FILE
 DATA RECORD IS MASTER-RECORD.
01 MASTER-RECORD.
 05 FILLER PICTURE X(173).

FD SORTED-MASTER-FILE
 DATA RECORD IS SORTED-MASTER-RECORD.
01 SORTED-MASTER-RECORD.
 05 FILLER PICTURE X(173).

FD PRINT-FILE
 DATA RECORD IS OUTPUT-REPORT-RECORD.
01 OUTPUT-REPORT-RECORD.
 05 REPORT-OUT PICTURE X(120).
.
.
.

PROCEDURE DIVISION.
.
.
.

SORT-ROUTINE SECTION.
 SORT SD-FILE
 ASCENDING KEY TITLE-IN
 USING MASTER-FILE
 GIVING SORTED-MASTER-FILE.
 IF SORT-RETURN > 0
 DISPLAY "SORT FAILED".
.
.
.

SORT-REPORT SECTION.
 print a report on PRINT-FILE using SORTED-MASTER-FILE.
.
.
.
 STOP RUN.
```

---

図 2. SORT コマンドを持つサンプル COBOL プログラム (2/2)

## レコードのマージ

120 ページの図 3 のサンプル COBOL プログラムは DFSORT を呼び出し、事前ソート済みブックストア・マスター・ファイル (MASTER-FILE) を別の事前ソート済みファイル (NEW-BOOKS-FILE) とマージして、新規のマスター・ファイル (MERGED-FILE) を作成します。

このプログラムの JCL は、次のとおりです。

---

```
//EXAMP JOB A492,PROGRAMMER
//BOOKS EXEC PGM=COBOLP
//STEPLIB DD DSN=USER.PGMLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//MASTERFL DD DSN=A123456.MASTER,DISP=OLD
//NEWBOOKS DD DSN=A123456.NEW,DISP=OLD
//MERGEDFL DD DSN=A123456.OUTC,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA
```

---

120 ページの図 3 は、サンプル COBOL プログラムを示しています。

```
IDENTIFICATION DIVISION.
PROGRAM-ID.
 COBOLP.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
 SELECT SD-FILE ASSIGN TO
 DUMMYNM.
 SELECT MASTER-FILE ASSIGN TO
 MASTERFL.
 SELECT NEW-BOOKS-FILE ASSIGN TO
 NEWBOOKS.
 SELECT MERGED-FILE ASSIGN TO
 MERGEDFL.
DATA DIVISION.
FILE SECTION.
SD SD-FILE
 DATA RECORD IS SD-RECORD.
01 SD-RECORD.
 05 TITLE-KEY PICTURE X(75).
 05 FILLER PICTURE X(98).

FD MASTER-FILE
 DATA RECORD IS MASTER-RECORD.
01 MASTER-RECORD.
 05 FILLER PICTURE X(173).

FD NEW-BOOKS-FILE
 DATA RECORD IS NEW-BOOKS-RECORD.
01 NEW-BOOKS-RECORD.
 05 FILLER PICTURE X(173).

FD MERGED-FILE
 DATA RECORD IS MERGED-RECORD.
01 MERGED-RECORD.
 05 FILLER PICTURE X(173).
.
.
.

PROCEDURE DIVISION.
.
.
.

MERGE-ROUTINE SECTION.
 MERGE SD-FILE
 ASCENDING KEY TITLE-KEY
 USING MASTER-FILE NEW-BOOKS-FILE
 GIVING MERGED-FILE.
 IF SORT-RETURN > 0
 DISPLAY "MERGE FAILED".
 STOP RUN.
```

---

図 3. MERGE コマンドを持つサンプル COBOL プログラム



## COBOL FASTSRT によるソート

117 ページの図 2 に示されたレコードをソートするための COBOL プログラムの場合、入力 (MASTER-FILE からの) および出力 (SORTED-MASTER-FILE への) は COBOL FASTSRT オプションに適格です。このコンパイル時 FASTSRT オプションを使用すると、COBOL ではなく DFSORT が入出力処理を行うので、ソートがかなり高速で実行されるようになります。FASTSRT の詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」およびご使用のサイトで利用可能なコンパイラ・バージョンに関する「COBOL プログラマーズ・ガイド」を参照してください。

注: COBOL では、ソート入力とソート出力を別々に評価し、FASTSRT に適格かどうかを調べます。入力または出力プロシージャがあるためにソートの入力または出力が適格とならない場合には、このようなプロシージャを置き換え、DFSORT 制御ステートメントを使用して同じことを行うことができます。例えば、制御ステートメント OUTREC を使用すると、レコードを出力データ・セットに書き込む前にどのように再フォーマットするかを指示することができます。

## PL/I プログラムからの DFSORT の呼び出し

DFSORT を呼び出す場合、PL/I プログラムでは、SORT 制御ステートメント、RECORD 制御ステートメント、および DFSORT が使用できる主記憶域量を受け渡す必要があります。RECORD 制御ステートメントでは、入力データ・セットのレコード・タイプと長さを指定します。RECORD 制御ステートメントの後に、DFSORT のための主記憶域値を MAX またはバイト数で表した値のいずれかとして指定します。(主記憶域値として MAX を指定した場合に、DFSORT のパフォーマンスが最良になります。)

DFSORT 制御ステートメントは、SORTCNTL または DFSPARM データ・セット内に指定することもできます。

以下は、122 ページの図 4 に示されたプログラム用の JCL です。英文学 (English) 学部の本だけを選択する INCLUDE 制御ステートメントを指定するために、SORTCNTL データ・セットが使用されています。

```
//EXAMP JOB A492,PROGRAMMER
//BOOKS EXEC PGM=PLIPGM
//STEPLIB DD DSN=USER.PGMLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//SORTOUT DD DSN=A123456.SORT.SAMPOUT,DISP=OLD
//SORTCNTL DD *
 INCLUDE COND=(110,5,CH,EQ,C'ENGL')
/*
//SYSPRINT DD SYSOUT=A
```

122 ページの図 4 に示されたサンプル PL/I プログラムは DFSORT を呼び出して、ブックストア・ファイルをタイトルによってソートします。このプログラムは、適切な SORT および RECORD 制御ステートメントと、DFSORT 主記憶域値として MAX を指定しています。

```
PLIPGM: PROC OPTIONS(MAIN);

 DCL 1 MASTER_RECORD,
 5 TITLE_IN CHAR(75),
 5 AUTH_LN_IN CHAR(20),
 .
 .
 .
 5 PRICE_IN BIN FIXED(31);

 DCL RETURN_CODE FIXED BIN(31,0);
 DCL MAXSTOR FIXED BIN(31,0);
 UNSPEC(MAXSTOR)='00000000'B||UNSPEC('MAX');
 .
 .
 .
 CALL PLISRTA (' SORT FIELDS=(1,75,CH,A) ',
 ' RECORD TYPE=F,LENGTH=(173) ',
 MAXSTOR,
 RETURN_CODE);
 IF RETURN_CODE ^= 0 THEN DO;
 PUT SKIP EDIT ('SORT FAILED')(A);
 CALL PLIRETC(RETURN_CODE);
 END;
 .
 .
 .
 CALL OUTPUT;
 .
 .
 .
 OUTPUT: PROCEDURE;
 .
 .
 .
 . Print a report from the sorted master file (SORTOUT)
 .
 .
 .
 END;
END PLIPGM;
```

---

図4. SORT コマンドを持つサンプル PL/I プログラム

### 要約

本章では、COBOL および PL/I から DFSORT を呼び出す方法を説明しました。

## 第 9 章 インストール・デフォルトの指定変更

IBM では、事前設定デフォルトを付けて DFSORT を出荷します。インストール時、システム・プログラマーはこれらのデフォルトを変更できます。例えば、IBM 提供のデフォルトは、DFSORT 制御ステートメントを DFSORT メッセージ・データ・セットにリストすることです。しかし、ご使用のサイトで、DFSORT 制御ステートメントをリストしないことをデフォルトにすることができます。

さらに、システム・プログラマーは、DFSORT が直接実行される場合に扱うジョブ用と DFSORT がプログラムから呼び出される場合に扱うジョブ用にそれぞれ別個にシステム・デフォルトを確立することもできます。したがって、DFSORT を直接実行するときは、デフォルトは DFSORT 制御ステートメントをリストすることであり、DFSORT をプログラムから呼び出すときは、デフォルトは DFSORT 制御ステートメントをリストしないことである、とすることもできます。

多くの場合、特定のインストール・デフォルトがジョブに適切でないときは、次のような方法で、デフォルトを一時的に指定変更することができます。

- DFSPARM データ・セット内の PARM パラメーターとして
- DFSPARM データ・セット内の OPTION、ALTSEQ、または DEBUG パラメーターとして
- JCL EXEC ステートメントの PARM パラメーターとして
- SYSIN または SORTCNTL データ・セット内の OPTION、ALTSEQ、または DEBUG パラメーターとして
- 呼び出し側プログラムから渡される OPTION、ALTSEQ、または DEBUG パラメーターとして

DFSPARM データ・セット内の OPTION、ALTSEQ、および DEBUG パラメーターは、DFSORT を直接実行するのかプログラムから呼び出すのかにかかわらず、インストール・デフォルトを指定変更する有用な方法です。

本章では、多くの使用可能デフォルトの一部を指定変更する方法、JCL EXEC ステートメントの PARM パラメーター、および DFSPARM データ・セット内の OPTION 制御ステートメントについて学習します。

次のような ICETOOL DEFAULTS ジョブを使用すると、ご使用のサイトで選択されているすべてのインストール・デフォルトをリストすることができます。

```
//S1 EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//DFLT DD SYSOUT=A
//TOOLIN DD *
 DEFAULTS LIST(DFLT)
/*
```

ICETOOL については、『第 10 章 ICETOOL ユーティリティの使用』で学習します。

指定変更できるすべてのパラメーターの詳細および指定変更する際に使用できるさまざまな方法については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。特に、その中の『付録 C DFSORT オプションの指定/指定変更』を参照してください。

---

### JCL EXEC ステートメントへの PARM パラメーターの指定

DFSORT を直接実行するときは、JCL EXEC ステートメントの PARM パラメーターを使用して、ジョブの特定のデフォルトを指定変更することができます。例えば、ご使用のサイトのデフォルトが DFSORT ステートメントをリストすること（インストール・パラメーター LIST=YES）であるのに、ユーザーのジョブではリストしたくない場合は、PARM フィールドに NOLIST を指定できます。

```
//S1 EXEC PGM=SORT,PARM='NOLIST'
```

これに対し、デフォルトが DFSORT ステートメントをリストしないこと（インストール・パラメーター LIST=NO）であるのに、リストさせたい場合は、PARM フィールドに LIST を指定できます。

```
//S1 EXEC PGM=SORT,PARM='LIST'
```

複数の PARM パラメーターを指定することもできます。例えば、DFSORT ステートメントをリストする場合に、SUM フィールドがオーバーフローした場合には DFSORT に 4 の戻りコードを設定させ、動的に割り振られた 6 個の SYSDA 作業データ・セットを使用し、エラーが検出された場合には DFSORT を ABEND させるには、次のように指定できます。

```
//S1 EXEC PGM=SORT,PARM='LIST,OVFL0=RC4,DYNALOC=(SYSDA,6),ABEND'
```

---

### DFSPARM への OPTION 制御ステートメントの指定

DFSORT を直接実行する場合でもプログラムから呼び出す場合でも、DFSPARM データ・セット内の OPTION ステートメントを使用して、ジョブの特定のデフォルトを指定変更することができます。これを行うには、//DFSPARM DD \* ステートメントの後に、単独でまたは他の DFSORT 制御ステートメントと一緒に、OPTION ステートメントを入れます。

OPTION ステートメントで指定変更したい特定のデフォルトは、等価の照合レコードを元の順序で書き込むかどうかを指定する EQUALS=NO インストール・オプションです。

ご使用のサイトのデフォルトが EQUALS=NO（推奨値）であり、一時的に指定変更したい場合（ユーザーのジョブでは等価の照合レコードが元の順序で書き込まれるようにしたい場合）には、次のように OPTION ステートメントに EQUALS を指定できます。

```
//DFSPARM DD *
 OPTION EQUALS
/*
```

あるいは、ご使用のサイトでインストール・デフォルトとして EQUALS=YES を確立しており、それを一時的に指定変更したい (ユーザー・ジョブでは等価の照合レコードをランダムに書き込みたい) 場合には、OPTION ステートメントに NOEQUALS を指定できます。

```
//DFSPARM DD *
 OPTION NOEQUALS
/*
```

DFSPARM データ・セットの中では、OPTION ステートメントに複数のパラメーターを指定でき、また追加の制御ステートメントを指定することもできます。例えば、DFSORT ステートメントをリストする場合に、SUM フィールドがオーバーフローした場合には DFSORT に 4 の戻りコードを設定させ、動的に割り振られた 6 個の SYSDA 作業データ・セットを使用し、エラーが検出された場合には DFSORT を ABEND させるには、次のように指定できます。

```
//DFSPARM DD *
 OPTION LIST,OVFL0=RC4,DYNALLOD=(SYSDA,6)
 DEBUG ABEND
/*
```

### 要約

本章では、JCL EXEC ステートメントの PARM パラメーターまたは OPTION 制御ステートメントを使用して、選択した DFSORT インストール・デフォルトを指定変更する方法を説明しました。



---

## 第 10 章 DFSORT の効率的な使用

DFSORT から最高のパフォーマンスが得られるのは、以下の指針に従った場合です。

- 十分な主記憶域を提供する。
- メモリー・オブジェクト・ソート、ハイパー・ソート、およびデータ・スペース・ソートを許可する。
- ワークスペースに高速ディスクを使用する。
- INREC を使用して不要なフィールドを除去する。
- INCLUDE または OMIT を使用して不要なレコードを除去する。
- STOPAFT および SKIPREC を使用して不要なレコードを除去する。
- SUM を使用してレコードを統合する。
- OUTFIL を使用して複数の出力データ・セットを作成する。
- DFSORT 制御ステートメントを使用してプログラム・ロジックを置き換える。
- COBOL の場合には FASTSORT を使用する。
- パフォーマンスを低下させる可能性のあるオプションを避ける。

追加の提案事項は、「z/OS DFSORT アプリケーション・プログラミング・ガイド」と「z/OS DFSORT チューニング・ガイド」に記載されています。

---

### 主記憶域を十分に提供する

デフォルトでは、DFSORT は、ソート、マージ、およびコピー・アプリケーションに 6 MB の主記憶域を使用することができ、また該当する場合にはそれ以上の主記憶域を使用します。ご使用のサイトで SIZE=MAX のインストール・デフォルトを使用していない場合は、ジョブに対して MAINSIZE=MAX パラメーターを設定することをお勧めします。例えば、次のようにして MAINSIZE=MAX を設定できます。

```
//DFSPPARM DD *
 OPTION MAINSIZE=MAX
/*
```

あるいは、MAINSIZE=*n*M パラメーター (*n* MB) を指定して、ジョブに必要なもっと多くの (または少ない) ストレージを DFSORT に与えることもできますが、MAINSIZE=MAX ではなく MAINSIZE=*n*M を使用する特別な理由がない限り、この方法はお勧めしません。例えば、次のようにして MAINSIZE=2M を設定できます。

```
//DFSPPARM DD *
 OPTION MAINSIZE=2M
/*
```

## メモリー・オブジェクト・ソート、ハイパー・ソート、およびデータ・スペース・ソートを許可する

デフォルトでは、DFSORT は、該当する場合には、ディスク作業データ・セットの代わりにまたは一緒に、メモリー・オブジェクト、ハイパースペース、またはデータ・スペースを使用して、ソート・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。ご使用のサイトで MOSIZE=0、HIPRMAX=0、または DSPSIZE=0 のインストール・デフォルトを使用している場合、大規模データ・セットをソートするときには、MOSIZE=MAX、HIPRMAX=OPTIMAL、および DSPSIZE=MAX パラメーターを指定する必要があるかもしれません。例えば、次のようにしてこれらのパラメーターを設定できます。

```
//DFSPPARM DD *
 OPTION MOSIZE=MAX,HIPRMAX=OPTIMAL,DSPSIZE=MAX
/*
```

## 作業データ・セットに高速ディスクを使用する

IBM のエンタープライズ・ストレージ・サーバー・サブシステムなどの高速ディスクを使用すると、作業データ・セットの最良のパフォーマンスが得られます。IBM の FICON 接続ソリューションなどの高速チャネルを活用すると、スループットをさらに改善することができます。DFSORT は、これらのテクノロジーによって提供される高速データ転送機能を活用するように最適化されます。可能であれば、テープを作業データ・セットとして使用することは避けてください。

## INREC を使用して不要なフィールドを除去する

レコードを再フォーマットする必要がある場合、INREC を使用してレコードを相当短くすれば、その結果として処理が速くなります。

INREC は、レコードの処理前にレコードを再フォーマットし、OUTREC は、処理後にレコードを再フォーマットします。したがって、レコードを短くするには INREC を使用し、レコードを長くするには OUTREC を使用する必要があります。INREC を使用する際に参照する制御ステートメントおよび対応するレコード位置の要約については、表 49 を参照してください。

表 49. INREC の場合の制御ステートメントおよび対応するレコード

| 制御ステートメント | 元のレコード | 再フォーマットされたレコード |
|-----------|--------|----------------|
| SORT      |        | ✓              |
| MERGE     |        | ✓              |
| SUM       |        | ✓              |
| OUTREC    |        | ✓              |
| OUTFIL    |        | ✓              |
| INCLUDE   | ✓      |                |
| OMIT      | ✓      |                |

INREC の詳細については、45 ページの『第 5 章 固定フィールドを持つレコードの再フォーマット』を参照してください。



---

## INCLUDE または OMIT を使用して不要なレコードを除去する

当然、入力レコードの数は処理時間に影響します。レコードが少なければ、DFSORT アプリケーションはそれだけ速く処理されます。可能であれば INCLUDE または OMIT を使用して、アプリケーションに関連するレコードだけを選択するようにすれば、パフォーマンスを向上させることができます。

INCLUDE および OMIT の詳細については、29 ページの『第 3 章 レコードの組み込みまたは省略』を参照してください。

---

## STOPAFT および SKIPREC を使用して不要なレコードを除去する

STOPAFT および SKIPREC オプションを使用することによっても、処理する入力レコードの数を減らすことができます。

- STOPAFT は、ソートまたはコピーのために受け入れるレコードの最大数を指定する場合に使用します。
- SKIPREC は、ソートまたはコピーが始まる前にスキップするレコード数を指定する場合に使用します。

これらのオプションの使用方法については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

---

## SUM を使用してレコードを統合する

該当する場合には、ジョブに対し SUM ステートメントを使用して以下を行うことによって、ソートまたはマージのパフォーマンスを向上させることができます。

- 等しい制御フィールドを持つ 2 つのレコードが見つかるたびにフィールドの内容を加算する。DFSORT は、結果を一方のレコードに入れ、他方を削除するので、出力レコードの数が削減されます。
- FIELDS=NONE を指定して重複制御フィールドを持つレコードを削除する。

SUM の詳細については、39 ページの『第 4 章 レコードの合計』を参照してください。

---

## OUTFIL を使用して複数の出力データ・セットを作成する

同じ入力データ・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、OUTFIL を使用すれば、入力データ・セットは 1 度だけしか読み取られないので、パフォーマンスが向上します。OUTFIL は、ソート、マージ、およびコピー・アプリケーションに使用することができ、洗練されたフィルター操作、編集、変換、検索と置換、および報告書作成の機能を提供します。

OUTFIL の詳細については、89 ページの『第 7 章 複数の出力データ・セットおよび報告書の作成』を参照してください。

---

## DFSORT 制御ステートメントを使用してプログラム・ロジックを置き換える

一般には、適切な DFSORT 制御ステートメントを使用する方がプログラム・ロジックよりも効率的です。可能であれば、プログラム・コードによって同じ機能を行う代わりに、DFSORT 制御ステートメントの INCLUDE、OMIT、INREC、OUTREC、SUM、および OUTFIL を使用するようにしてください。

---

## COBOL の場合には FASTSRT を使用する

COBOL では、FASTSRT コンパイル時オプションを使用することにより、DFSORT のパフォーマンスを向上させることができます。FASTSRT を使用すると、COBOL ではなく DFSORT が入出力処理を行います。FASTSRT オプションの詳細については、ご使用のサイトで利用可能なコンパイラ・バージョンに関する「COBOL プログラマーズ・ガイド」を参照してください。

---

## パフォーマンスを低下させる可能性のあるオプションを避ける

以下にリストしたパラメーターおよびオプションは、DFSORT パフォーマンスに悪影響を与える可能性があります。これらを使用するのは、必要がある場合のみに限定してください。これらのパラメーターおよびオプションの内容とパフォーマンスへの影響についての詳細は、「z/OS DFSORT アプリケーション・プログラミング・ガイド」および「z/OS DFSORT チューニング・ガイド」を参照してください。

- VERIFY パラメーター
- EQUALS パラメーター
- NOBLKSET パラメーター
- CKPT パラメーター
- EQUCOUNT パラメーター
- NOCINV パラメーター
- LOCALE パラメーター
- BSAM パラメーター
- NOASSIST パラメーター
- NOCFW パラメーター
- テープの作業データ・セット
- ユーザー出力ルーチン
- EFS プログラム
- ユーザー出力ルーチンの動的リンク・エディット
- MOSIZE、HIPRMAX、DSPSIZE、または MAINSIZE パラメーターに小さな値を使用すること

### 要約

本章では、各種の制御ステートメントおよびパラメーターの効率的な使用を含め、DFSORT パフォーマンスを向上させる方法を説明しました。

---

## 第 3 部 ICETOOL の使い方を学ぶ

|                                                   |     |
|---------------------------------------------------|-----|
| 第 11 章 ICETOOL ユーティリティーの使用 . . .                  | 133 |
| ICETOOL オペレーター . . . . .                          | 133 |
| サンプル入力データ・セット . . . . .                           | 134 |
| 必要な JCL ステートメントの作成 . . . . .                      | 135 |
| ICETOOL コメント・ステートメントおよびブラン<br>ク・ステートメント . . . . . | 136 |
| 数値フィールドの統計を印刷する . . . . .                         | 136 |
| オペレーター・ステートメントの継続 . . . . .                       | 138 |
| VB データ・セット・レコード長の統計 . . . . .                     | 139 |
| 同一のソート済みデータ・セットの作成 . . . . .                      | 139 |
| ソート済みデータ・セットのいろいろなサブセット<br>を作成する . . . . .        | 142 |
| 複数の未ソート・データ・セットを作成する . . . . .                    | 144 |
| ある範囲内の値をカウントする . . . . .                          | 145 |
| 簡潔報告書を印刷する . . . . .                              | 147 |
| 調整済み報告書を印刷する . . . . .                            | 148 |
| フォーマット項目を使用する . . . . .                           | 150 |
| 編集マスク . . . . .                                   | 150 |
| 先行ゼロ . . . . .                                    | 152 |
| 編集パターン . . . . .                                  | 152 |
| 統計なし . . . . .                                    | 153 |
| 除法 . . . . .                                      | 153 |
| 先行、浮動、および末尾の文字 . . . . .                          | 154 |
| 分割した報告書を印刷する . . . . .                            | 154 |
| フィールドの出現回数を印刷する . . . . .                         | 157 |
| フィールド出現回数によってレコードを選択する . . . . .                  | 159 |
| いろいろなデータ・セットからのフィールドを結合<br>する . . . . .           | 162 |
| いろいろなデータ・セットからのレコードを突き合<br>わせる . . . . .          | 165 |
| 完全な ICETOOL ジョブと TOOLMSG 出力 . . . . .             | 169 |



---

## 第 11 章 ICETOOL ユーティリティーの使用

ICETOOL は多目的の DFSORT ユーティリティーであり、DFSORT の諸機能を使用して、単一ステップで 1 つ以上のデータ・セットに複数の操作を実行します。

本章では、広範囲にわたる作業を行うのに使用できる ICETOOL の 13 個の「オペレーター」を紹介します。多くの ICETOOL オペレーターを使用する大規模な「メイン」の ICETOOL ジョブ、および特定のポイントを説明するもっと小さな追加のいくつかの ICETOOL ジョブを作成している間に、ICETOOL の JCL および制御ステートメントを学習することになります。ICETOOL の諸機能を完全に利用するには、ICETOOL のすべてのオペレーター、オペランド、および呼び出し方法を熟知している必要があります。これについては、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

注: おそらく、関連のないタスクは異なる ICETOOL ジョブで行うと思いますが、関連のないタスクを同じ ICETOOL ジョブで行うこともできます。本章で構成する ICETOOL ジョブでは、各種の ICETOOL オペレーターの使用を説明しやすいように両方の方法を使用しています。

---

### ICETOOL オペレーター

以下にリストした 13 個の ICETOOL オペレーターを使用して、さまざまな機能を実行できます。13 個の ICETOOL オペレーターをいろいろ組み合わせて使用することにより、多くの複雑なタスクを実行するアプリケーションを容易に作成することができます。

#### 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カウンを含むメッセージを印刷します。

COUNT を使用して、データ・セット内のレコード数の一致基準 (例えば、空、空でない、5000 レコードより小、等しい、またはより大、など) に基づいて RC=12、RC=4、または RC=0 を設定することもできます。

#### DEFAULTS

DFSORT のインストール・デフォルトを別個のリスト・データ・セットに印刷します。

#### DISPLAY

指定され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別個のリスト・データ・セットに印刷します。簡潔報告書、調整済み報告書、または分割した報告書を作成できます。

#### MODE

オペレーターのグループに対して設定またはリセットできる次の 3 つのモードがあります。

- STOP モード (デフォルト) は、エラーが検出された場合にそれ以降の操作を停止します。

- CONTINUE モードは、エラーが検出された場合でもそれ以降の操作を続けます。
- SCAN モードでは、実際に操作を行うことなく ICETOOL ステートメントを検査できます。

### **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およびそれが現れる回数を別個のリスト・データ・セットに印刷します。簡潔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、値カウントが指定の基準に一致するもの(例えば、重複値だけまたは非重複値だけ)に限定することができます。

### **RANGE**

指定された数値フィールドの指定された範囲内の値のカウントを含むメッセージをデータ・セットに印刷します。

### **SELECT**

指定された数値フィールドまたは文字フィールドの値が現れる回数の一致基準(例えば、重複値だけまたは非重複値だけ)に基づいて、データ・セットから、出力データ・セットに入れるレコードを選択します。選択されなかったレコードを、別個の出力データ・セットに保管することができます。

### **SORT**

データ・セットを 1 つ以上の出力データ・セットにソートします。

### **SPLICE**

レコードの指定されたフィールドのうち、情報は異なるが指定された数値フィールド値または文字フィールド値が同じである(すなわち、重複値)フィールドを一緒に接合します。複数のレコードの指定のフィールドを結合して 1 つの出力レコードを作成することができます。さまざまなデータ・セットのレコードのフィールドを接合できるので、SPLICE によっていろいろな「結合」および「突き合わせ」操作を行うことができます。

### **STATS**

指定された数値フィールドの最小、最大、平均、および合計を含むメッセージをデータ・セットに印刷します。

### **UNIQUE**

指定された数値フィールドまたは文字フィールドの固有値のカウントを含むメッセージを印刷します。

### **VERIFY**

データ・セット内の指定された 10 進数フィールドを調べ、各フィールドで見つかったそれぞれの無効値を識別するメッセージを印刷します。

---

## **サンプル入力データ・セット**

それぞれの ICETOOL オペレーター (DEFAULTS と MODE を除く) には、入力データ・セットが必要です。あるオペレーターで使用される入力データ・セットと別のオペレーターで使用される入力データ・セットが、同じであっても異なっても構いません。ICETOOL は、単一ステップで多くのデータ・セットを処理できます。

本章では、入力データ・セットとして、ブランチ・データ・セット SORT.BRANCH、ブックストア・データ・セット SORT.SAMPIN、および追加のブックストア・データ・セット SORT.SAMPADD を使用します。これらのサンプル・データ・セットの追加情報については、187 ページの『付録 A. サンプル・データ・セットの作成』および 189 ページの『付録 B. サンプル・データ・セットの記述』を参照してください。SORT.BRANCH から ICETOOL が作成する 2 つの一時データ・セットも入力として使用されます。

注: 本章では、SORT.BRANCH、SORT.SAMPIN、および SORT.SAMPADD 以外のデータ・セットも使用します。テキストで使用されているデータ・セットと一致させるには最初からデータ・セットを作成することもできますし、すでにお持ちのデータ・セットに同様の処理を行うこともできます。

---

## 必要な JCL ステートメントの作成

ICETOOL ジョブは、以下のものから構成されます。

1. すべての ICETOOL ジョブに必要な JCL ステートメント。
2. ICETOOL ジョブによって実行すべき操作を示すオペレーター・ステートメント。
3. 指定されたオペレーター・ステートメントの結果として必要となる JCL ステートメント。

ICETOOL ジョブを作成する際の最初のステップは、常に必要となる JCL を書くことです。以下に、ICETOOL ジョブに必要な JCL の骨組みを示しています。

```
//EXAMP JOB A492,PROGRAMMER
//TOOL EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//TOOLIN DD *
<ICETOOL statements go here>
/*
<Additional JCL statements go here>
```

- JOB ステートメントは、ジョブの開始を合図します。
- EXEC ステートメントは、ジョブ・ステップの開始を合図し、ICETOOL プログラムを実行するようオペレーティング・システムに指示します。
- TOOLMSG ステートメントは、ICETOOL メッセージ用の出力データ・セットを定義します。
- DFSMSG ステートメントは、DFSORT メッセージ用の出力データ・セットを定義します。
- TOOLIN ステートメントは、ICETOOL ステートメント (コメント、ブランク、およびオペレーター・ステートメント) の前に置かれます。ユーザーが作成する ICETOOL ステートメントは、TOOLIN の後に指定する必要があります。ユーザーが作成する追加の JCL ステートメントは、TOOLIN ステートメントの前または ICETOOL ステートメントの後に指定することができます。メイン ICETOOL ジョブでは、追加の JCL ステートメントは ICETOOL ステートメントの後に入れられます。

---

## ICETOOL コメント・ステートメントおよびブランク・ステートメント

コメント・ステートメントおよびブランク・ステートメントは、ICETOOL オペレーター・ステートメントの間のどこにでも入れることができます。

- コメント・ステートメントは、桁 1 のアスタリスク (\*) から始まり、ICETOOL オペレーター・ステートメントと一緒に印刷されます。
- ブランク・ステートメントは、桁 1 から 72 までブランクを含むもので、ICETOOL が適宜ブランク行を印刷した後は無視されます。

例のブランク・ステートメントとコメント・ステートメントは、次のようになっています。

```
* Statistics from all branches
```

以下は、これらのステートメントを作成するステップです。

表 50. ブランク・ステートメントとコメント・ステートメントを作成するステップ

| ステップ | アクション                               |
|------|-------------------------------------|
| 1    | TOOLIN DD ステートメントの後で、1 行スキップする。     |
| 2    | 桁 1 にアスタリスク (*) を入力し、その後にコメントを指定する。 |

完了すると、TOOLIN ステートメントは次のようになります。

```
//TOOLIN DD *
```

```
* Statistics from all branches
/*
```

メイン ICETOOL ジョブの場合、コメント・ステートメントは、その機能を記述する各オペレーターの前に入れられます。必ずしもこのようにする必要はありませんが、このようにしておくとうりです。

### これまで

これまでの説明では、ICETOOL ユーティリティーの基本部分を紹介しました。次に、以下のチュートリアルを使用することにより、ICETOOL オペレーターの大部分を学習することができます。以下の節にはメイン ICETOOL ジョブの各部分が含まれており、この章が終わるまでには完全なメイン ICETOOL ジョブが作成されるようになっています。章の終わりに、完全なメイン ICETOOL ジョブとその結果のメッセージを含む節があります。

---

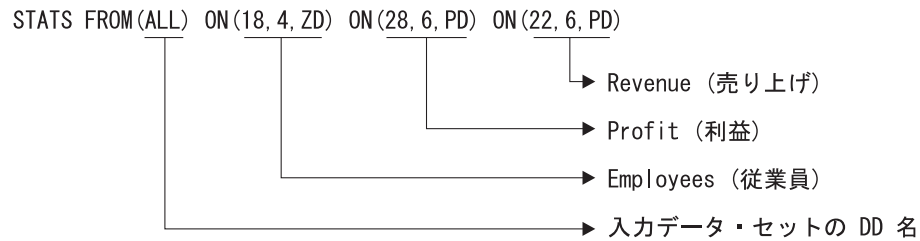
## 数値フィールドの統計を印刷する

数値フィールドを含むデータ・セットに作業を行っている場合は、このような 1 つ以上のフィールドに関する統計情報が必要になる場合があります。STATS オペレーターを使用すると、最大 10 個までの特定数値フィールドの最小、最大、平均、および合計値を調べることができます。

ブランチ・オフィス・データ・セットの従業員 (Employees)、利益 (Profit)、売り上げ (Revenue) の各フィールドの統計を印刷する STATS オペレーターは、次のよう



になります。



以下に、この STATS オペレーターを作成するためのステップを示します。

表 51. STATS オペレーターの作成ステップ

| ステップ | アクション                                                                                                                                                                                                                            |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | コメント・ステートメントの後に <b>STATS</b> を入力する (STATS の前に 1 つ以上のブランクを入れても構いません)。                                                                                                                                                             |
| 2    | 少なくとも 1 つのブランクを入れてから、 <b>FROM(ALL)</b> を入力する<br><br>FROM は、統計を印刷したい入力データ・セットの DD 名 (すなわち、DD ステートメントの名前) を指定します。この場合は、ALL が選択された DD 名ですが、任意の有効な 1 から 8 文字の DD 名を使用できます。                                                           |
| 3    | 少なくとも 1 つのブランクを入れてから、 <b>ON</b> を入力する<br><br>ON は、統計を印刷したい対象となるフィールドを定義します。                                                                                                                                                      |
| 4    | 以下のものをコンマで区切り、括弧で囲んで入力する<br>1. 入力レコードの先頭 (先頭位置はバイト 1) からの相対位置で表した、従業員フィールドが始まる場所。従業員フィールドはバイト 18 から始まります。<br>2. 従業員フィールドの長さ (バイト数)。従業員フィールドは 4 バイトの長さです。<br>3. データ・フォーマットを表すコード。従業員フィールドには、 <b>ZD</b> として指定されるゾーン 10 進データが入られます。 |
| 5    | 少なくとも 1 つのブランクを入れてから、 <b>ON</b> を入力する<br><br>ON は、統計を印刷したい対象となる別のフィールドを定義します。1 つの STATS ステートメントで最大 10 個のフィールドの統計を印刷できます。ON フィールドは、その統計を印刷したい同じ順序で指定してください。                                                                       |
| 6    | 利益フィールドの位置 (28)、長さ (6)、およびフォーマット (パック 10 進数を表す <b>PD</b> ) をコンマで区切り、括弧で囲んで入力する。                                                                                                                                                  |
| 7    | 少なくとも 1 つのブランクを入れてから、 <b>ON</b> を入力する。売り上げフィールドの位置 (22)、長さ (6)、およびフォーマット ( <b>PD</b> ) をコンマで区切り、括弧で囲んで入力する。<br><br>ステートメントは、桁 1 から 72 の間にコーディングしてください。                                                                           |

さらに、ALL の DD 名を使用して A123456.SORT.BRANCH データ・セットに対する DD ステートメントを作成し、これをジョブの終わりに入れる必要があります。

```
//ALL DD DSN=A123456.SORT.BRANCH,DISP=SHR
```

完了すると、TOOLIN ステートメントおよび ALL ステートメントは次のようになります。

```
//TOOLIN DD *

* Statistics from all branches
STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
/*
//ALL DD DSN=A123456.SORT.BRANCH,DISP=SHR
```

この STATS 操作が実行されると、結果は TOOLMSG データ・セットに入れられます。これまでに作成した ICETOOL ジョブを実行した場合、TOOLMSG 出力は次のようになります。

```
ICE600I 0 DFSORT ICETOOL UTILITY RUN STARTED

ICE650I 0 VISIT http://www.ibm.com/storage/dfsor FOR ICETOOL PAPERS, EXAMPLES AND MORE

ICE632I 0 SOURCE FOR ICETOOL STATEMENTS: TOOLIN

ICE630I 0 MODE IN EFFECT: STOP

 * Statistics from all branches
 STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
ICE627I 0 DFSORT CALL 0001 FOR COPY FROM ALL TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 0000000000000012
ICE607I 0 STATISTICS FOR (18,4,ZD) :
ICE608I 0 MINIMUM: +000000000000015, MAXIMUM: +000000000000035
ICE609I 0 AVERAGE: +000000000000024, TOTAL : +000000000000298
ICE607I 0 STATISTICS FOR (28,6,PD) :
ICE608I 0 MINIMUM: -000000000004278, MAXIMUM: +000000000008276
ICE609I 0 AVERAGE: +000000000004222, TOTAL : +0000000000050665
ICE607I 0 STATISTICS FOR (22,6,PD) :
ICE608I 0 MINIMUM: +000000000012300, MAXIMUM: +0000000000042820
ICE609I 0 AVERAGE: +000000000027469, TOTAL : +0000000000329637
ICE602I 0 OPERATION RETURN CODE: 00

ICE601I 0 DFSORT ICETOOL UTILITY RUN ENDED - RETURN CODE: 00
```

出力から、以下のことがわかります。

- メッセージ ICE628I は、処理されたレコードのカウントを与えます。
- メッセージ ICE607I、ICE608I、および ICE609I は、各 ON フィールドの数値統計を、指定された順に与えます。
- 各オペレーターの戻りコードはメッセージ ICE602I で与えられ、最高位の戻りコードはメッセージ ICE601I で与えられます。

## オペレーター・ステートメントの継続

STATS ステートメント (またはその他の ICETOOL オペレーター・ステートメント) を単一行の桁 1 から 72 までに収めることができない場合は、複数行にわたって継続することができます。オペレーターまたはオペランドの後にハイフン (-) を付けて行を終了すれば、次の行は継続行として扱われます。ハイフンに後に指定された文字は無視されます。

オペレーターと各オペランドは完全に 1 行 (桁 1 から 72 まで) に指定しなければならないことに注意してください。

以下に例を示します。

STATS            - this is the operator  
FROM(ALL) - ALL is the ddname for SORT.BRANCH  
ON(18,4,ZD)-  
ON(28,6,PD)-  
ON(22,6,PD)

## VB データ・セット・レコード長の統計

可変長レコード・データ・セットを処理しているときは、STATS オペレーターを使用することにより、以下の情報を簡単に入手できます。

- データ・セットの最も短いレコード (最小)
- データ・セットの最も長いレコード (最大)
- データ・セットのレコードの平均長 (平均)
- データ・セットの合計バイト数 (合計)

ICETOOL は、これらの統計を印刷するための特殊な ON(VLEN) フィールドを用意しています。ON(VLEN) は、その他の ON フィールドと同様に指定してください。

VB レコードに対して ON(p,m,f) フィールドをコーディングするときは、DFSORT 制御ステートメント・フィールドのところで前述したように、4 バイトの RDW のための 4 を開始位置に忘れずに追加してください。

次の STATS オペレーターは、レコード長および PD フィールドの最小、最大、合計、および平均値を、最初の 2 データ・バイトに報告します。

```
STATS FROM(VBIN) ON(VLEN) ON(5,2,PD)
```

### これまで

これで、ICETOOL の STATS オペレーターを使用して数値フィールドおよび VB レコード長の統計を印刷する方法がわかりました。次の節では、ICETOOL の SORT オペレーターの使用法を学習します。

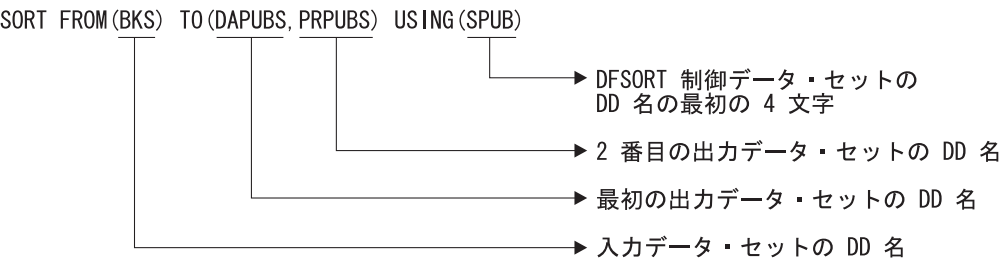
## 同一のソート済みデータ・セットの作成

ICETOOL の SORT オペレーターを使用することにより、ソート済み出力データ・セットを作成できます。単一の SORT オペレーターでは、1 つの出力データ・セットまたは 10 個までの同一の出力データ・セットを作成することができます。INCLUDE または OMIT ステートメントを使用すると、入力レコードのサブセットを選択できます。INREC または OUTREC ステートメントを使用すると、入力レコードのフィールドを再配置できます。OUTFIL ステートメントを使用すると、レコードの異なるサブセットまたはフィールドの異なる配置を使用して出力データ・セットをいくつでも作成できます。

この例では、サンプル・ブックストア・データ・セット (SORT.SAMPIN) と追加のブックストア・データ・セット (SORT.SAMPADD) の両方を入力として使用します。

出版社 VALD および WETH から本を選択し、出版社とタイトルによってソートし、それらをディスクと印刷データ・セットに書き込む SORT オペレーターは、次

のようになっています。



以下に、この SORT オペレーターを作成するためのステップを示します。

表 52. SORT オペレーターの作成ステップ

| ステップ | アクション                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | コメント・ステートメントを作成する (オプション)。<br><br>* Books from VALD and WETH                                                                                                                                                                                                                                                                                                                                                                                        |
| 2    | コメント・ステートメントの後に <b>SORT</b> を入力する                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 3    | 少なくとも 1 つのブランクを入れてから、 <b>FROM(BKS)</b> を入力する<br><br>BKS は、ソートしたい入力データ・セットの DD 名を指定します。                                                                                                                                                                                                                                                                                                                                                              |
| 4    | 少なくとも 1 つのブランクを入れてから、 <b>TO(DAPUBS,PRPUBS)</b> を入力する<br><br>TO は、ソート済みレコードのサブセットを収容する出力データ・セットの DD 名を指定します。 DFSORT で許可される任意のタイプ (永続、一時、ディスク、テープ、印刷など) の同一の出力データ・セットを、最大 10 個まで作成することができます。<br><br>この場合、DAPUBS は一時ディスク・データ・セットとして選択された DD 名、PRPUBS は印刷データ・セットとして選択された DD 名です。希望する任意の有効な 1 から 8 文字の DD 名を使用できます。<br><br>ICETOOL は自動的に OUTFIL を使用して、入力データ・セットの 1 回の受け渡しから両方の出力データ・セットを作成します。                                                           |
| 5    | 少なくとも 1 つのブランクを入れてから、 <b>USING(SPUB)</b> を入力する<br><br>USING は、DFSORT 制御ステートメントが入っているデータ・セットの DD 名の最初の 4 文字を指定します。この場合、選択された 4 文字は SPUB ですが、DD 名として有効であれば希望する任意の 4 文字を使用できます。 DD 名の最後の 4 文字は常に CNTL なので、この場合、完全な DD 名は SPUBCNTL となります。<br><br>SORT オペレーターの場合は、入力データ・セットのソート方法を DFSORT に指示するために、DFSORT 制御ステートメント・データ・セット (SPUBCNTL) に SORT 制御ステートメントを指定する必要があります。また、INCLUDE、OMIT、INREC、OUTREC、および OUTFIL のような追加の DFSORT 制御ステートメントを、必要に応じて指定することもできます。 |

SORT オペレーターを伴う JCL ステートメントを作成するには、次のように行います。

表 53. SORT オペレーターのための JCL ステートメントを作成するステップ

| ステップ | アクション                                                                                                                                                                                                                                                       |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | <p>入力データ・セットに対する DD ステートメントを作成し、それらをジョブの終わりに入れる。2 つの入力データ・セットを連結するには、2 番目の DD ステートメントの DD 名フィールドをブランクのままにする必要があります。</p> <pre>//BKS    DD DSN=A123456.SORT.SAMPIN,DISP=SHR //        DD DSN=A123456.SORT.SAMPADD,DISP=SHR</pre>                              |
| 2    | <p>ディスクおよび印刷出力データ・セットに対する DD ステートメントを作成し、それらをジョブの終わりに入れる。</p> <pre>//DAPUBS DD DSN=&amp;&amp;DSRT,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=SYSDA //PRPUBS DD SYSOUT=A</pre>                                                                                      |
| 3    | <p>DFSORT 制御ステートメント・データ・セットに対する DD ステートメントを作成し、それをジョブの終わりに入れる。</p> <pre>//SPUBCNTL DD *</pre>                                                                                                                                                               |
| 4    | <p>出版社とタイトルによって入力データ・セットをソートする SORT 制御ステートメント、および出版社が VALD と WETH の本だけを選択する INCLUDE ステートメントを作成し、それらを SPUBCNTL ステートメントの後に入れる。</p> <pre>SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),           FORMAT=CH</pre> |

完了すると、この SORT オペレーターの TOOLIN ステートメントと DD ステートメントは、次のようになります。

```
* Books from VALD and WETH
SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
/*
//BKS DD DSN=A123456.SORT.SAMPIN,DISP=SHR
// DD DSN=A123456.SORT.SAMPADD,DISP=SHR
//DAPUBS DD DSN=&&DSRT,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=SYSDA
//PRPUBS DD SYSOUT=A
//SPUBCNTL DD *
SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),
 FORMAT=CH
/*
```

**ヒント:** OR で結合された 2 つの条件を使用して上記の INCLUDE ステートメントを作成する代わりに、37 ページの『INCLUDE および OMIT の場合のサブストリング検索』で学習した INCLUDE のサブストリング検索機能を使用して、次のように 1 つの条件で INCLUDE ステートメントを作成することもできます。

```
INCLUDE COND=(106,4,SS,EQ,C'VALD,WETH')
```

表 54 は、結果の出力データ・セットに現れるレコードの Book Title (ブック・タイトル) および Publisher (出版社) フィールドを示しています。実際のレコードにはすべてのフィールドが含まれています。

表 54. 出版社 VALD および WETH の本

| Book Title | Publisher |
|------------|-----------|
| 1 75       | 106 109   |

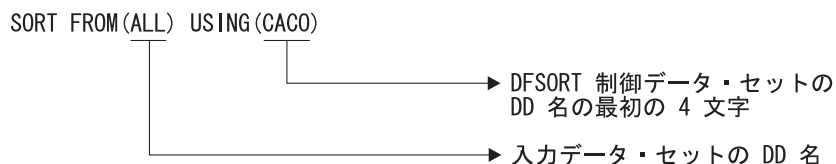
表 54. 出版社 VALD および WETH の本 (続き)

| Book Title                   | Publisher |
|------------------------------|-----------|
| CELLS AND HOW THEY WORK      | VALD      |
| COMPLETE SPANISH DICTIONARY  | VALD      |
| EDITING SOFTWARE MANUALS     | VALD      |
| FREUD'S THEORIES             | VALD      |
| INTRODUCTION TO BIOLOGY      | VALD      |
| NOVEL IDEAS                  | VALD      |
| SHORT STORIES AND TALL TALES | VALD      |
| STRATEGIC MARKETING          | VALD      |
| VIDEO GAME DESIGN            | VALD      |
| ZEN BUSINESS                 | VALD      |
| ANTICIPATING THE MARKET      | WETH      |
| CIVILIZATION SINCE ROME FELL | WETH      |
| COMPUTERS: AN INTRODUCTION   | WETH      |
| EIGHTEENTH CENTURY EUROPE    | WETH      |
| GUIDE TO COLLEGE LIFE        | WETH      |
| GUNTHER'S GERMAN DICTIONARY  | WETH      |
| REBIRTH FROM ITALY           | WETH      |
| SYSTEM PROGRAMMING           | WETH      |
| THE INDUSTRIAL REVOLUTION    | WETH      |

## ソート済みデータ・セットのいろいろなサブセットを作成する

同じ入力データ・セットからレコードのサブセットを作成したい場合は、**SORT** または **COPY** オペレーターと共に **OUTFIL** ステートメントを使用することができます。 **OUTFIL** ステートメントは出力データ・セットの **DD** 名を指定するので、**TO** オペランドは不要です。 **OUTFIL** のすべての機能は、**SORT** および **COPY** オペレーターを通して使用可能です。

カリフォルニアおよびコロラドにあるブランチ・オフィス用に、市区町村によってソートした別個のディスクおよびテープ・データ・セットを作成する **SORT** オペレーターは、次のようになります。



以下に、この **SORT** オペレーターを作成するためのステップを示します。

表 55. **SORT** オペレーターの作成ステップ

| ステップ | アクション                                                                                    |
|------|------------------------------------------------------------------------------------------|
| 1    | コメント・ステートメントを作成する (オプション)。<br><br>* Separate output for California and Colorado branches |
| 2    | コメント・ステートメントの後に <b>SORT</b> を入力する                                                        |

表 55. SORT オペレーターの作成ステップ (続き)

| ステップ | アクション                                                                                                                                                 |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3    | <p>少なくとも 1 つのブランクを入れてから、<b>FROM(ALL)</b> を入力する</p> <p>ALL は、ソートしたい入力データ・セットの DD 名を指定します。STATS オペレーターで A123456.SORT.BRANCH に対して使用した同じ DD 名を使用できます。</p> |
| 4    | <p>少なくとも 1 つのブランクを入れてから、<b>USING(CACO)</b> を入力する</p> <p>CACOCNTL データ・セットには、SORT および OUTFIL ステートメントが入っています。</p>                                        |

SORT オペレーターを伴う JCL ステートメントを作成するには、以下のステップに従ってください。

表 56. SORT オペレーターのための JCL ステートメントを作成するステップ

| ステップ | アクション                                                                                                                                                                                                                                                                                                                                                                                         |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | <p>DFSORT 制御ステートメント・データ・セットに対する DD ステートメントを作成し、それをジョブの終わりに入れる。</p> <pre>//CACOCNTL DD *</pre>                                                                                                                                                                                                                                                                                                 |
| 2    | <p>市区町村によって入力データ・セットをソートする SORT 制御ステートメント、カリフォルニア・ブランチだけを選択する OUTFIL ステートメント、およびコロラド・ブランチだけを選択する OUTFIL ステートメントを作成し、それらを CACOCNTL ステートメントの後に入れる。</p> <pre> SORT FIELDS=(1,15,CH,A) OUTFIL FNames=(CADASD,CATAPE),INCLUDE=(16,2,CH,EQ,C'CA') OUTFIL FNames=(CODASD,COTAPE),INCLUDE=(16,2,CH,EQ,C'CO') </pre>                                                                                   |
| 3    | <p>ディスクおよびテープ出力データ・セットに対する DD ステートメントを作成し、それらをジョブの終わりに入れる。</p> <pre> //CADASD DD DSN=&amp;&amp;CA,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390 //CATAPE DD DSN=CA.BRANCH,UNIT=3480,VOL=SER=111111, // DISP=(NEW,KEEP),LABEL=(,SL) //CODASD DD DSN=&amp;&amp;CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390 //COTAPE DD DSN=CO.BRANCH,UNIT=3480,VOL=SER=222222, // DISP=(NEW,KEEP),LABEL=(,SL) </pre> |

完了すると、この SORT オペレーターの TOOLIN ステートメントと DD ステートメントは、次のようになります。

```

* Separate output for California and Colorado branches
SORT FROM(ALL) USING(CACO)
/*
//CACOCNTL DD *
 SORT FIELDS=(1,15,CH,A)
 OUTFIL FNames=(CADASD,CATAPE),INCLUDE=(16,2,CH,EQ,C'CA')
 OUTFIL FNames=(CODASD,COTAPE),INCLUDE=(16,2,CH,EQ,C'CO')
/*
//CADASD DD DSN=&&CA,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//CATAPE DD DSN=CA.BRANCH,UNIT=3480,VOL=SER=111111,
// DISP=(NEW,KEEP),LABEL=(,SL)
//CODASD DD DSN=&&CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//COTAPE DD DSN=CO.BRANCH,UNIT=3480,VOL=SER=222222,
// DISP=(NEW,KEEP),LABEL=(,SL)

```



表 57 は、最初の OUTFIL ステートメントを使用した結果として CADASD データ・セット (&&CA) および CATAPE データ・セット (CA.BRANCH) に現れるレコードを示しています。

表 57. 市区町村によってソートされたカリフォルニアに関するレコード

| City          | State        | Employees    | Revenue      | Profit       |
|---------------|--------------|--------------|--------------|--------------|
| <b>1 15</b>   | <b>16 17</b> | <b>18 21</b> | <b>22 27</b> | <b>28 33</b> |
| Los Angeles   | CA           | 32           | 22530        | -4278        |
| Morgan Hill   | CA           | 15           | 18200        | 3271         |
| Sacramento    | CA           | 29           | 42726        | 8276         |
| San Diego     | CA           | 22           | 32940        | 8275         |
| San Francisco | CA           | 35           | 42820        | 6832         |
| San Jose      | CA           | 21           | 27225        | 8264         |
| Sunnyvale     | CA           | 18           | 16152        | -978         |

表 58 は、2 番目の OUTFIL ステートメントを使用した結果として CODASD データ・セット (&&CO) および COTAPE データ・セット (CO.BRANCH) に現れるレコードを示しています。

表 58. 市区町村によってソートされたコロラドに関するレコード

| City         | State        | Employees    | Revenue      | Profit       |
|--------------|--------------|--------------|--------------|--------------|
| <b>1 15</b>  | <b>16 17</b> | <b>18 21</b> | <b>22 27</b> | <b>28 33</b> |
| Aspen        | CO           | 20           | 25800        | 5200         |
| Boulder      | CO           | 32           | 33866        | 7351         |
| Denver       | CO           | 33           | 31876        | 6288         |
| Fort Collins | CO           | 22           | 12300        | -2863        |
| Vail         | CO           | 19           | 23202        | 5027         |

171 ページの図 6 に、完全な TOOLMSG 出力を示しています。

#### これまで

このチュートリアルではこれまでに、ICETOOL の STATS オペレーターを使用して数値フィールドの統計を印刷する方法、および ICETOOL の SORT オペレーターを使用して入力データ・セットをソートし複数の出力データ・セットを作成する方法を学習しました。次に、ICETOOL の COPY オペレーターを学習します。

## 複数の未ソート・データ・セットを作成する

入力データ・セットの未ソート・コピーを作成したい場合は、ICETOOL の COPY オペレーターを使用できます。COPY オペレーターは DFSORT ステートメントを必要としません。しかし、該当する場合には、DFSORT ステートメント (例えば、INCLUDE、OMIT、INREC、OUTREC、または OUTFIL) を指定することができます。

以下に、COPY オペレーター・ステートメントの例を 2 つとそれに付随する JCL ステートメントを示しています。



```
//TOOLIN DD *
COPY FROM(ALL) TO(D1,D2,D3)
COPY FROM(ALL) TO(P1) USING(COPY)
/*
//ALL DD DSN=A123456.SORT.BRANCH,DISP=SHR
//D1 DD DSN=A123456.SORT.COPY1,DISP=OLD
//D2 DD DSN=A123456.SORT.COPY2,DISP=OLD
//D3 DD DSN=A123456.SORT.COPY3,DISP=OLD
//P1 DD SYSOUT=*
//COPYCNTL DD *
INCLUDE COND=(16,2,CH,EQ,C'CA')
/*
```

最初の COPY オペレーターは、A123456.SORT.COPY1、A123456.SORT.COPY2、および A123456.SORT.COPY3 内に、A123456.SORT.BRANCH の同一コピーを作成します。

2 番目の COPY オペレーターは、カリフォルニアのブランチのために A123456.SORT.BRANCH レコードを印刷します。結果の印刷出力では、文字フィールドだけしか判読可能でないことに注意してください。読みやすいフォーマットで数値フィールドを表示する方法については、本章で後述します。

コピーの方がソートよりも効率的であるので、可能であれば、SORT オペレーターではなく COPY オペレーターを使用するようにしてください。

#### これまで

本章ではこれまでに、3 つの重要な ICETOOL オペレーターである STATS、SORT、および COPY について学習しました。次のチュートリアルでは、ICETOOL の RANGE オペレーターの使用法を示します。

## ある範囲内の値をカウントする

ICETOOL の RANGE オペレーターを使用すると、定義した範囲内にある特定の数値フィールドの値の数をカウントすることができます。範囲は、以下のもので定義できます。

| オペランド                          | 比較                |
|--------------------------------|-------------------|
| <b>EQUAL</b>                   | 値に等しい             |
| <b>NOTEQUAL</b>                | 値に等しくない           |
| <b>LOWER</b>                   | 値より小さい            |
| <b>HIGHER</b>                  | 値より大きい            |
| <b>HIGHER</b> および <b>LOWER</b> | ある値より大きい、別の値より小さい |

利益が -1500 より大きく +8000 より小さいカリフォルニア・ブランチの数のカウントを印刷するには、次のような RANGE ステートメントを作成します。

RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)

→ 範囲の上限  
→ 範囲の下限  
→ Profit (利益)  
→ 入力データ・セットの DD 名

CADASD で定義された入力データ・セットは、以前に (SORT オペレーターのところで) カリフォルニア・ブランチ用に作成した同じデータ・セットです。

HIGHER(-1500) は、-1500 より大きな利益フィールド内の値をカウントするように指示し、その一方で LOWER(+8000) は、+8000 より小さな利益フィールド内の値をカウントするように指示します。負の限界の場合は、数値の前に負符号 (-) を指定する必要があります。正の限界の場合は、数値の前に正符号 (+) を指定しても省略しても構いません。したがって、HIGHER(8000) と HIGHER(+8000) は同じです。

従業員数が 32 人より少ない (カリフォルニアおよびコロラドの) ブランチの数のカウントを印刷するには、次のような RANGE ステートメントを作成します。

RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)

→ 範囲の上限  
→ Employees (従業員)  
→ 入力データ・セットの DD 名

CADASD および ALL は以前に定義してあるので、ICETOOL ジョブに新規の JCL ステートメントを追加する必要はありません。

これらの RANGE オペレーターが実行されると、結果は TOOLMSG データ・セットに入れられます。これらの RANGE オペレーターに対して作成される TOOLMSG 出力は、次のようになります。

```
* California branches profit analysis
RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
ICE627I 0 DFSORT CALL 0004 FOR COPY FROM CADASD TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000007
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (28,6,PD) : 000000000000003
ICE602I 0 OPERATION RETURN CODE: 00
```

```
* Branches with less than 32 employees
RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)
ICE627I 0 DFSORT CALL 0005 FOR COPY FROM ALL TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000012
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (18,4,ZD) : 000000000000008
ICE602I 0 OPERATION RETURN CODE: 00
```

出力から、以下のことがわかります。

- メッセージ ICE628I は、処理されたレコードのカウントを与えます。
- メッセージ ICE631I は、指定された範囲内の値のカウントを与えます。利益が -1500 より大きく +8000 より小さいカリフォルニア・ブランチは 3 つ、従業員数が 32 人より少ないブランチは 8 つありました。

- 各オペレーターの戻りコードは、メッセージ ICE602I で与えられます。

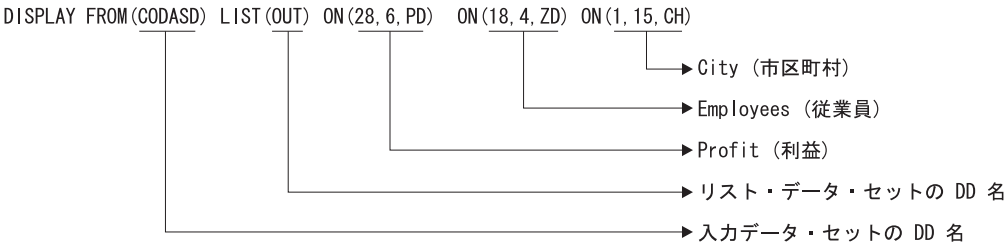
これまで

ここでは、ICETOOL の RANGE オペレーターを使用して、特定のフィールドに関してある範囲内の値の数をカウントする方法について学習しました。次に、ICETOOL の DISPLAY オペレーターを学習します。

## 簡潔報告書を印刷する

数値フィールドは、印刷されたときに判読可能でないフォーマット (2 進数、固定小数点、または 10 進数) になっていることがしばしばあります。ICETOOL の DISPLAY オペレーターを使用すると、入力データ・セットからの 20 までの数値フィールドと文字フィールドを読みやすい形式で示す簡潔報告書を印刷することができます。指定したフィールドは、定義したリスト・データ・セットに印刷されます。数値フィールドの場合は、それぞれの数値の 10 進値に伴って正符号 (+) または負符号 (-) が適宜印刷されます。

コロラド・ブランチの利益 (Profit)、従業員 (Employees)、市区町村 (City) の各フィールドを示すリスト・データ・セットを印刷するには、次の DISPLAY オペレーターを作成します。



CODASD で定義された入力データ・セットは、以前に (SORT オペレーターのところで) コロラド・ブランチ用に作成した同じデータ・セットです。LIST は、フィールドの印刷先のリスト・データ・セットの DD 名を指定します。この場合は、OUT がリスト・データ・セットとして選択された DD 名ですが、希望する任意の有効な 1 から 8 文字の DD 名を使用することができます。ON フィールドは、その値をリスト・データ・セットに印刷させたい同じ順序で指定してください。

OUT は前に定義されていないので、OUT に対する JCL ステートメントをジョブの終わりに追加する必要があります。

```
//OUT DD SYSOUT=A
```

この DISPLAY オペレーターが実行されると、OUT データ・セットは次のようになります。

| (28,6,PD)         | (18,4,ZD)        | (1,15,CH)    |
|-------------------|------------------|--------------|
| +0000000000005200 | +000000000000020 | Aspen        |
| +0000000000007351 | +000000000000032 | Boulder      |
| +0000000000006288 | +000000000000033 | Denver       |
| -0000000000002863 | +000000000000022 | Fort Collins |
| +0000000000005027 | +000000000000019 | Vail         |

注: 出力には、実際には、101 ページの『OUTFIL を使用した報告書の作成』で前述したように、報告書の最初のバイトに ANSI 紙送り制御文字が含まれています。しかし、ICETOOL は常に、ブランク行を印刷するのに、'0' および ' ' の ANSI 紙送り制御文字を使用する代わりに、報告書に実際のブランク行を挿入します。したがって、OUTFIL 報告書の場合に行うように、ICETOOL 報告書を表示するときは何を行わなくてもブランク行を強制的に表示することができます。ANSI 紙送り制御文字はこの節では示されていませんが、報告書には存在します。

利益、従業員、市区町村の各値は、上部に各列の見出しが付いたページの別個の列に印刷されます。複数のページを印刷する場合、DISPLAY は各ページの上部に見出しを書き込みます。見出しを印刷したくない場合は、DISPLAY の NOHEADER オペランドを使用すれば見出しの印刷を抑止できます。

DISPLAY には 2 つの特殊な ON フィールドがあり、以下の目的に使用できます。

- **ON(VLEN)** は、可変長レコード・データ・セットの場合に使用でき、各レコードの長さを印刷します。
- **ON(NUM)** は、各レコードの相対レコード番号 (1 から始まる) を印刷します。

ON(VLEN) または ON(NUM) フィールドは、その他の ON フィールドと同様に使用してください。

---

## 調整済み報告書を印刷する

前のチュートリアルでは、DISPLAY オペレーターを使用して数値フィールドと文字フィールドの簡潔リストを印刷する方法を紹介しました。DISPLAY の追加オペランドを使用することにより、さまざまな調整済み報告書フォーマットで数値フィールドと文字フィールドを示すリスト・データ・セットを作成することができます。以下を指定できます。

- タイトル・エレメント (TITLE、DATE、PAGE、および TIME オペランド) およびタイトル・エレメント間のスペーシング (TBETWEEN オペランド)
- 1、2、または 3 行フィールド見出し (HEADER オペランド)
- 列のスペーシング (INDENT、BETWEEN、および STATLEFT オペランド)
- フィールド・フォーマット (BLANK および PLUS オペランド)、および個々の ON フィールドのフォーマット項目
- 統計 (TOTAL、AVERAGE、MAXIMUM、および MINIMUM オペランド)
- ページ当たり行数 (LINES オペランド)

これらのオペランドおよび DISPLAY で使用できるその他のオペランドの詳細については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

タイトル行、フィールド見出し、合計、平均、および最小を持つ、市区町村 (City)、利益 (Profit)、従業員 (Employees) の各フィールドを示すコロラド・ブランチの報告書を印刷するには、次の DISPLAY オペレーターを作成します。

```

* Print a report for the Colorado branches
DISPLAY FROM(CODASD) LIST(RPT) -
 ▶ データ・セットの DD 名

DATE TITLE(' Colorado Branches Report') PAGE -
 ▶ タイトル行エレメント

HEADER(' City') HEADER(' Profit') HEADER(' Employees') -
 ▶ フィールド見出し

ON(1, 15, CH) ON(28, 6, PD) ON(18, 4, ZD) BLANK BETWEEN(5) -
 ▶ 列間のスペース
 ▶ 代替印刷フォーマット
 ▶ フィールド

TOTAL(' Total') AVERAGE(' Average') MINIMUM(' Lowest')
 ▶ 統計

```

CODASD は、前に作成したコロラド・ブランチ・データ・セットの DD 名です。  
RPT は、報告書の印刷先のリスト・データ・セットの DD 名です。

DATE、TITLE、および PAGE は、タイトル行に含めるエレメントとその配置を示します。これらのオペランドは、それらをリスト・データ・セットに印刷させたい同じ順序で指定してください。

各 HEADER は、対応するフィールドに使用する見出しを示します。ON フィールドとそれに対応する HEADER スtringは、その値をリスト・データ・セットに印刷させたい同じ順序で指定してください。

BLANK は、数値を、正符号の場合はブランク、負符号の場合は -、および先行ゼロを付けずに印刷することを指定します。BETWEEN は、列間のスペース数を指定します (デフォルトの 3 を指定変更します)。HEADER、BLANK、および BETWEEN は、より報告書らしいフォーマットを作成するために、フィールドの位置調整と列幅、およびフィールド間のスペーシングを変更します。

TOTAL、AVERAGE、および MINIMUM を使用すると、示された統計が、指定の String で識別され、報告書の終わりに作成されます。

RPT は前に定義されていないので、RPT に対する JCL ステートメントをジョブの終わりに追加する必要があります。

```
//RPT DD SYSOUT=A
```

この DISPLAY オペレーターが実行されると、結果の RPT データ・セットは次のようになります。

```

10/21/03 Colorado Branches Report - 1 -

City Profit Employees

Aspen 5200 20
Boulder 7351 32
Denver 6288 33
Fort Collins -2863 22
Vail 5027 19

```

|         |       |     |
|---------|-------|-----|
| Total   | 21003 | 126 |
| Average | 4200  | 25  |
| Lowest  | -2863 | 19  |

タイトル行と見出し行は各ページの上部に印刷されます。文字データは左寄せされ、数値データは右寄せされ、ゼロは抑止されます。統計は、データ列の後に印刷されます。

## フォーマット項目を使用する

前のチュートリアルでは、BLANK オペランドを使用して、報告書のすべての数値の表示方法を変更しました。フォーマット項目を使用すると、個々の数値フィールドおよび関連する統計の報告書における外観を変更することができます。例えば、区切り文字、桁数、小数点、小数点以下の桁数、符号、先行ゼロ、除算、先行ストリング、浮動ストリング、および末尾ストリングに関して変更できます。フォーマット項目を使用して、文字フィールドの先行ストリングまたは末尾ストリングを挿入することもできます。

フォーマット項目は、次のように、コンマで区切って ON オペランドの一部として指定します。 ON(p,m,f,formatting)、ON(VLEN,formatting)、および ON(NUM,formatting)。

## 編集マスク

33 個の事前定義編集マスクから選択できます。以下の表は、使用可能なマスクを紹介し、それぞれのマスクで値 12345678 と -1234567 がどのように印刷されるかを示しています。パターンでは、以下のものが使用されています。

- **d** は、10 進数字 (0-9) を表します。
- **w** は、正の値の場合はブランクとなり、負の値の場合は - となる先行符号を表します。
- **x** は、正の値の場合はブランクとなり、負の値の場合は - となる末尾符号を表します。
- **y** は、正の値の場合はブランクとなり、負の値の場合は ( となる先行符号を表します。
- **z** は、正の値の場合はブランクとなり、負の値の場合は ) となる末尾符号を表します。

表 59. 編集マスク・パターン

| マスク | パターン                                   | 12345678   | -1234567   |
|-----|----------------------------------------|------------|------------|
| A0  | wdddddddddddddddddddddddddddd          | 12345678   | -1234567   |
| A1  | wd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd     | 12,345,678 | -1,234,567 |
| A2  | wd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd     | 12.345.678 | -1.234.567 |
| A3  | wd ddd ddd ddd ddd ddd ddd ddd ddd ddd | 12 345 678 | -1 234 567 |
| A4  | wd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd | 12'345'678 | -1'234'567 |
| A5  | d ddd ddd ddd ddd ddd ddd ddd ddd dddx | 12 345 678 | 1 234 567- |

表 59. 編集マスク・パターン (続き)

| マスク | パターン                                     | 12345678    | -1234567    |
|-----|------------------------------------------|-------------|-------------|
| B1  | wddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,d   | 1,234,567.8 | -123,456.7  |
| B2  | wddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,d   | 1.234.567,8 | -123.456,7  |
| B3  | wddd ddd ddd ddd ddd ddd ddd ddd ddd,d   | 1 234 567,8 | -123 456,7  |
| B4  | wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,d   | 1'234'567.8 | -123'456.7  |
| B5  | wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,d   | 1'234'567,8 | -123'456,7  |
| B6  | ddd ddd ddd ddd ddd ddd ddd ddd ddd,dx   | 1 234 567,8 | 123 456,7-  |
| C1  | wdd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd.dd   | 123,456.78  | -12,345.67  |
| C2  | wdd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,dd   | 123.456,78  | -12.345,67  |
| C3  | wdd ddd ddd ddd ddd ddd ddd ddd ddd,dd   | 123 456,78  | -12 345,67  |
| C4  | wdd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,dd   | 123'456.78  | -12'345.67  |
| C5  | wdd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,dd   | 123'456,78  | -12'345,67  |
| C6  | dd ddd ddd ddd ddd d ddd ddd ddd ddd,dxx | 123 456,78  | 12 345,67-  |
| D1  | wd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd.ddd   | 12,345.678  | -1,234.567  |
| D2  | wd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,ddd   | 12.345,678  | -1.234,567  |
| D3  | wd ddd ddd ddd ddd ddd ddd ddd ddd,ddd   | 12 345,678  | -1 234,567  |
| D4  | wd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd.ddd   | 12'345.678  | -1'234.567  |
| D5  | wd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddd   | 12'345,678  | -1'234,567  |
| D6  | d ddd ddd ddd ddd ddd ddd ddd ddd,dddx   | 12 345,678  | 1 234,567-  |
| E1  | yd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,dddz  | 12,345,678  | (1,234,567) |
| E2  | yd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,dddz  | 12.345,678  | (1.234,567) |
| E3  | yd ddd ddd ddd ddd ddd ddd ddd ddd dddz  | 12 345 678  | (1 234 567) |
| E4  | yd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'dddz      | 12'345'678  | (1'234'567) |
| F1  | ydd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,dddz | 123,456.78  | (12,345.67) |
| F2  | ydd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,dddz | 123.456,78  | (12.345,67) |
| F3  | ydd ddd ddd ddd ddd ddd ddd ddd ddd dddz | 123 456,78  | (12 345,67) |
| F4  | ydd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'dddz     | 123'456.78  | (12'345.67) |
| F5  | ydd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'dddz     | 123'456,78  | (12'345,67) |
| G1  | wddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd     | 1,234.5678  | -123.4567   |
| G2  | wddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd     | 1.234,5678  | -123,4567   |
| G3  | wddd ddd ddd ddd ddd ddd ddd ddd ddd     | 1 234,5678  | -123,4567   |
| G4  | wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd     | 1'234.5678  | -123.4567   |
| G5  | wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd     | 1'234,5678  | -123,4567   |
| G6  | ddd ddd ddd ddd ddd ddd ddd ddd ddd,dddx | 1 234,5678  | 123,4567-   |



前の報告書で Profit (利益) フィールドに E1 編集マスクを使用するには、単に ON(28,6,PD) を ON(28,6,PD,E1) に変更すれば済みます。RPT データ・セットは、次のようになります。

10/21/03                      Colorado Branches Report                      - 1 -

| City         | Profit  | Employees |
|--------------|---------|-----------|
| Aspen        | 5,200   | 20        |
| Boulder      | 7,351   | 32        |
| Denver       | 6,288   | 33        |
| Fort Collins | (2,863) | 22        |
| Vail         | 5,027   | 19        |
| Total        | 21,003  | 126       |
| Average      | 4,200   | 25        |
| Lowest       | (2,863) | 19        |

## 先行ゼロ

デフォルトでは、編集マスクを使用するときには先行ゼロは表示されませんが、次の 2 番目の ON フィールドのように、フォーマット項目に LZ を追加することによってこれを変更することができます。

HEADER('No leading zeros','(without LZ)') ON(28,6,PD,E1) -  
 HEADER('Leading zeros','(with LZ)') ON(28,6,PD,E1,LZ)

次の結果の報告書は、SORT.BRANCH からの Profit 値の表示方法が、E1 編集マスクと共に LZ を使用した場合と使用しなかった場合でどのように異なるかを示しています。2 行見出しを作成するために、HEADER('string1','string2') を使用していることに注意してください。

| No leading zeros<br>(without LZ) | Leading zeros<br>(with LZ) |
|----------------------------------|----------------------------|
| (4,278)                          | (00,000,004,278)           |
| 6,832                            | 00,000,006,832             |
| (2,863)                          | (00,000,002,863)           |
| 8,276                            | 00,000,008,276             |
| (978)                            | (00,000,000,978)           |
| 6,288                            | 00,000,006,288             |
| 7,351                            | 00,000,007,351             |
| 3,271                            | 00,000,003,271             |
| 5,027                            | 00,000,005,027             |
| 8,264                            | 00,000,008,264             |
| 8,275                            | 00,000,008,275             |
| 5,200                            | 00,000,005,200             |

## 編集パターン

編集マスクは、符号なし数値データ (電話番号、日付、時刻、識別番号、など) には特に有用ではありません。代わりに、編集パターンを使用することにより、これらのタイプの数値の報告書への表示方法を変更できます。編集パターンを指定するには、E'pattern' を使用します。パターンの中では、数値から数字 (0-9) を出力させる場所では 9 を使用し、他の文字を出力させる場所では他の任意の文字を使用します。



桁 21-28 に *mmddyyyy* 形式の 8 バイトの ZD 日付がある場合は、  
ON(21,8,ZD,E'99/99/9999') を使用すれば、この日付を *mm/dd/yyyy* として表示できます。03122004 という 8 バイトの値は、03/12/2004 として表示されます。

桁 31-40 に *aaapppnnnn* 形式の 10 バイトの ZD 電話番号がある場合は、  
ON(31,10,ZD,E'(999)-999-9999') を使用すれば、これを (aaa)-ppp-nnnn として表示できます。0123456789 という 10 バイトの値は、(012)-345-6789 として表示されます。

## 統計なし

デフォルトでは、TOTAL、MAXIMUM、MINIMUM、および AVERAGE (さらに、後半で学習する BTOTAL、BMINIMUM、BMAXIMUM、および BAVERAGE) を使用して統計を要求すると、すべての数値 ON フィールドについての統計が表示されます。NOST フォーマット項目を使用すれば、意味のない、電話番号のような数値フィールドの統計を抑止することができます。以下の DISPLAY オペレーターは、SORT.BRANCH データ・セットからの報告書を印刷します。その際、Revenue および Profit フィールドは合計しますが、Employees フィールドは合計しません。

```
DISPLAY FROM(IN) LIST(RPT3) -
 HEADER('City') ON(1,15,CH) -
 HEADER('Employees') ON(18,4,ZD,NOST) -
 HEADER('Revenue') ON(22,6,PD) -
 HEADER('Profit') ON(28,6,PD) -
 TOTAL('Totals')
```

RPT3 データ・セットは、次のようになります。

| City          | Employees | Revenue | Profit |
|---------------|-----------|---------|--------|
| -----         | -----     | -----   | -----  |
| Los Angeles   | 32        | 22530   | -4278  |
| San Francisco | 35        | 42820   | 6832   |
| Fort Collins  | 22        | 12300   | -2863  |
| Sacramento    | 29        | 42726   | 8276   |
| Sunnyvale     | 18        | 16152   | -978   |
| Denver        | 33        | 31876   | 6288   |
| Boulder       | 32        | 33866   | 7351   |
| Morgan Hill   | 15        | 18200   | 3271   |
| Vail          | 19        | 23202   | 5027   |
| San Jose      | 21        | 27225   | 8264   |
| San Diego     | 22        | 32940   | 8275   |
| Aspen         | 20        | 25800   | 5200   |
| Totals        |           | 329637  | 50665  |

## 除法

以下の 10 通りの除法項目から選択できます。

- /D - 10 による除算
- /C - 100 による除算
- /K - 1000 による除算
- /DK - 10000 による除算 (10\*1000)
- /CK - 100000 による除算 (100\*1000)
- /M - 1000000 による除算 (1000\*1000)
- /G - 1000000000 による除算 (1000\*1000\*1000)
- /KB - 1024 による除算
- /MB - 1048576 による除算 (1024\*1024)

- /GB - 1073741824 による除算 (1024\*1024\*1024)

HEADER('Profit/(Loss) in K\$') および ON(28,6,PD,E1,/K) を使用した場合、  
SORT.BRANCH からの Profit 値は、次のようになります。

```
Profit/(Loss) in K$

(4)
6
(2)
8
0
6
7
3
5
8
8
5
```

## 先行、浮動、および末尾の文字

次のように、数値フィールドには浮動文字を追加でき、数値フィールドおよび文字フィールドには先行文字と末尾文字を追加できます。

- **F'string'** - フォーマット済み数値データの最初の非ブランク文字の左側に現れる浮動ストリング。
- **L'string'** - 文字または数値データ列の先頭に現れる先行ストリング。
- **T'string'** - 文字または数値データ列の終わりに現れる末尾ストリング。

HEADER('Profit') および ON(28,6,PD,A1,F'\$',T'\*\*) を使用した場合、  
SORT.BRANCH からの Profit 値は次のようになります。

```
Profit

$-4,278**
$6,832**
$-2,863**
$8,276**
$-978**
$6,288**
$7,351**
$3,271**
$5,027**
$8,264**
$8,275**
$5,200**
```

## 分割した報告書を印刷する

前のチュートリアルでは、DISPLAY オペレーターを使用して調整済み報告書を印刷する方法を紹介しました。DISPLAY の BREAK オペランドを使用すると、ソート済みの文字または数値の分割フィールドによってセクションに分割した報告書を作成することができます。ON(p,m,f,formatting) の場合と同様に、BREAK(p,m,f,formatting) の場合も一緒にフォーマット項目を使用することができます。また、分割タイトルのストリング (BTITLE オペランド) や、個々のセクションの統計 (BTOTAL、BAVERAGE、BMAXIMUM、および BMINIMUM オペランド) を指定することもできます。



この DISPLAY オペレーターが実行されると、次のような SECTIONS データ・セットについての 3 ページの報告書が作成されます。

BOOKS FOR INDIVIDUAL PUBLISHERS

- 1 -

PUBLISHER: VALD

| TITLE OF BOOK                | PRICE OF BOOK |
|------------------------------|---------------|
| CELLS AND HOW THEY WORK      | \$24.95       |
| COMPLETE SPANISH DICTIONARY  | \$6.50        |
| EDITING SOFTWARE MANUALS     | \$14.50       |
| FREUD'S THEORIES             | \$12.50       |
| INTRODUCTION TO BIOLOGY      | \$23.50       |
| NOVEL IDEAS                  | \$24.50       |
| SHORT STORIES AND TALL TALES | \$15.20       |
| STRATEGIC MARKETING          | \$23.50       |
| VIDEO GAME DESIGN            | \$21.99       |
| ZEN BUSINESS                 | \$12.00       |
| AVERAGE FOR THIS PUBLISHER   | \$17.91       |
| TOTAL FOR THIS PUBLISHER     | \$179.14      |

BOOKS FOR INDIVIDUAL PUBLISHERS

- 2 -

PUBLISHER: WETH

| TITLE OF BOOK                | PRICE OF BOOK |
|------------------------------|---------------|
| ANTICIPATING THE MARKET      | \$20.00       |
| CIVILIZATION SINCE ROME FELL | \$13.50       |
| COMPUTERS: AN INTRODUCTION   | \$18.99       |
| EIGHTEENTH CENTURY EUROPE    | \$17.90       |
| GUIDE TO COLLEGE LIFE        | \$20.00       |
| GUNTHER'S GERMAN DICTIONARY  | \$10.88       |
| REBIRTH FROM ITALY           | \$25.60       |
| SYSTEM PROGRAMMING           | \$31.95       |
| THE INDUSTRIAL REVOLUTION    | \$7.95        |
| AVERAGE FOR THIS PUBLISHER   | \$18.53       |
| TOTAL FOR THIS PUBLISHER     | \$166.77      |

BOOKS FOR INDIVIDUAL PUBLISHERS

- 3 -

| TITLE OF BOOK              | PRICE OF BOOK |
|----------------------------|---------------|
| AVERAGE FOR ALL PUBLISHERS | \$18.20       |
| TOTAL FOR ALL PUBLISHERS   | \$345.91      |

#### これまで

ここでは、ICETOOL の DISPLAY オペレーターを使用して、簡潔報告書、調整済み報告書、および分割した報告書を印刷する方法を学習しました。次に、ICETOOL の OCCUR オペレーターについて学習します。

---

## フィールドの出現回数を印刷する

ICETOOL の OCCUR オペレーターを使用すると、ON フィールド値によってソートした、さまざまな ON フィールド値の出現回数を示す簡潔報告書または調整済み報告書を印刷することができます。1 度だけしか現れない値は、「非重複」値と呼ばれます。複数回現れる値は、「重複」値と呼ばれます。

以下のオペランドを使用することにより、データ・セット内のすべての値をリストしたり、重複値と非重複値のいろいろな組み合わせをリストしたりすることができます。

- **ALLDUPS** - 重複値のみをリストする
- **NODUPS** - 非重複値のみをリストする
- **EQUAL** - 指定した回数だけ現れる値のみをリストする
- **HIGHER** - 指定した回数より多く現れる値のみをリストする
- **LOWER** - 指定した回数より少なく現れる値のみをリストする

OCCUR 報告書のセットアップは、DISPLAY 報告書のセットアップに似ています。OCCUR の ON フィールドは、その値をリスト・データ・セットに印刷させたい同じ順序で指定してください。OCCUR には 2 つの特殊な ON フィールドがあり、以下の目的に使用できます。

- **ON(VALCNT)** は、各フィールド値が現れる回数を印刷するのに使用できます。
- **ON(VLEN)** は、可変長レコードのレコード長に使用できます。

最大 10 個の ON フィールドを使用できます。ON(VALCNT) および ON(VLEN) 以外のすべての ON フィールド値は、出現のソートと出現回数のカウントに使用されます。例えば、ON(1,4,CH) を使用する場合、'ABCD+01' と 'ABCD-01' はその順序で (1,4,CH によって) ソートされ、'ABCD' の出現回数 2 回としてカウントされます。しかし、ON(1,4,CH) と ON(5,3,FS) を使用する場合、'ABCD-01' は 'ABCD+01' より前に (1,4,CH,A および 5,3,FS,A によって) ソートされ、'ABCD-01' の出現回数 1 回および 'ABCD+01' の出現回数 1 回としてカウントされます。

DISPLAY 報告書の場合と同様に、OCCUR 報告書に対して以下のものを使用できます。

- タイトル・エレメント (TITLE、DATE、PAGE、および TIME オペランド) およびタイトル・エレメント間のスペーシング (TBETWEEN オペランド)
- フィールド見出し (HEADER オペランド)
- 列のスペーシング (INDENT および BETWEEN オペランド)
- フィールド・フォーマット (BLANK および PLUS オペランド) と、ON(p,m,f,formatting)、ON(VALCNT,formatting)、および ON(VLEN,formatting) といった個々の ON フィールドのフォーマット項目
- ページ当たり行数 (LINES オペランド)

これらのオペランドおよび OCCUR で使用できるその他のオペランドの詳細については、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

この例では、入力としてサンプル・ブックストア・データ・セットを使用します。各出版社から使用されている異なる本の数を示す報告書を印刷するには、次の OCCUR ステートメントを作成します。

```
* Print the count of books in use from each publisher
OCCUR FROM(BKIN) LIST(PUBCT) BLANK -
 └─ 代替印刷フォーマット
 └─ データ・セットの DD 名

TITLE(' Books from Publishers') DATE(DMY.) -
 └─ タイトル行エレメント

HEADER(' Publisher') HEADER(' Books Used') -
 └─ フィールド見出し

ON(106,4,CH) ON(VALCNT,N05)
 └─ 出版社とカウント
```

BKIN は、サンプル・ブックストア・データ・セットの DD 名です。PUBCT は、報告書の印刷先のリスト・データ・セットの DD 名です。BLANK は、使用するフィールド・フォーマットを指定します。

TITLE は、タイトル行に現れるタイトル・ストリングを示します。DATE は、タイトル行に現れる日付のフォーマット (dd.mm.yy - dd は 2 桁の日、mm は 2 桁の月、yy は年の最後の 2 桁) を示します。

HEADER ストリングは、ON フィールドに対応しています。ON(VALCNT,N05) は、OCCUR で使用される特殊な ON フィールドであり、出現カウントを印刷します。N05 は、カウントの桁数を 5 に設定するために使用されるフォーマット項目であり、VALCNT のデフォルトである 15 桁を指定変更します。

A123456.SORT.SAMPIN および PUBCT データ・セットに対する DD ステートメントを作成し、それらをジョブの終わりに入れてください。

```
//BKIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//PUBCT DD SYSOUT=A
```

この OCCUR オペレーターが実行されると、PUBCT データ・セットは次のようになります。

```
Books from Publishers 21.10.03

Publisher Books Used

COR 7
FERN 4
VALD 5
WETH 4
```

各出版社の名前と共に、その出版社がサンプル・ブックストア・データ・セットに現れた回数 (これは、その出版社からの異なる本の数に等しい) が印刷されます。

#### これまで

ここでは、ICETOOL の OCCUR オペレーターを使用して、異なるフィールド値が現れる回数を示す簡潔報告書または調整済み報告書を印刷する方法を学習しました。次に、ICETOOL の SELECT オペレーターの使用法を学習します。

## フィールド出現回数によってレコードを選択する

ICETOOL の SELECT オペレーターを使用すると、ON フィールド値によってソートした、異なる ON フィールド値が現れる回数に従って選択したレコードを持つ出力データ・セットを作成することができます。OCCUR オペレーターの場合と同様、1 度だけしか現れない値は非重複値と呼ばれ、複数回現れる値は重複値と呼ばれます。

最大 10 個の ON フィールドを使用できます。ON フィールド値はすべて、出現のソートと出現回数のカウントに使用されます。例えば、ON(1,4,CH) を使用する場合、'ABCD+01' と 'ABCD-01' はその順序で (1,4,CH,A によって) ソートされ、('ABCD') の出現回数 2 回としてカウントされます。しかし、ON(1,4,CH) と ON(5,3,FS) を使用する場合、'ABCD-01' は 'ABCD+01' より前に (1,4,CH,A および 5,3,FS,A によって) ソートされ、'ABCD-01' の出現回数 1 回および 'ABCD+01' の出現回数 1 回としてカウントされます。

以下のオペランドを使用することにより、重複値および非重複値を持つレコードのいろいろな組み合わせを選択することができます。

- **FIRST** - 各値の最初のレコードのみを保持する (すなわち、非重複値を持つレコード、および重複値の最初のレコード)
- **LAST** - 各値の最後のレコードのみを保持する (すなわち、非重複値を持つレコード、および重複値の最後のレコード)
- **FIRSTDUP** - 重複値の最初のレコードのみを保持する
- **LASTDUP** - 重複値の最後のレコードのみを保持する
- **ALLDUPS** - 重複値を持つレコードのみを保持する
- **NODUPS** - 非重複値を持つレコードのみを保持する
- **EQUAL** - 指定した回数だけ現れる値を持つレコードのみを保持する
- **HIGHER** - 指定した回数より多く現れる値を持つレコードのみを保持する
- **LOWER** - 指定した回数より少なく現れる値を持つレコードのみを保持する

選択されたレコードは、TO オペランドで識別された出力データ・セットに書き込まれます。該当する場合には、上記のオペランドと一緒に DISCARD オペランドを使用することにより、**選択されなかったレコード**を DISCARD オペランドで識別した別個の出力データ・セットに保管することができます。TO データ・セットだけ、DISCARD データ・セットだけ、またはこの両方を作成できます。

USING データ・セットを使用して、SELECT 操作のための DFSORT INCLUDE、OMIT、INREC、および OUTFIL ステートメントを指定することができます。



INCLUDE または OMIT および INREC ステートメントの処理は、SELECT 処理の前に実行されます。OUTFIL ステートメントの処理は、SELECT 処理の後に実行されます。

5 冊以上の異なる本が使用されている出版社のレコードを収めた出力データ・セットを作成するには、次の SELECT ステートメントを作成します。

\* Separate output containing records for publishers  
\* with more than 4 books in use  
SELECT FROM(BKIN) TO(BKOUT) ON(106, 4, CH) HIGHER(4)

レコードの選択基準  
Publisher (出版社)  
出力データ・セットの DD 名  
入力データ・セットの DD 名

BKIN は、サンプル・ブックストア・データ・セットの DD 名です。BKOUT は、5 回以上現れる各出版社 (Publisher) フィールド値のレコード (この場合は、COR および VALD のすべてのレコード) を収める出力データ・セットの DD 名です。

A123456.BOOKS1 データ・セットに対する DD ステートメントを作成し、これをジョブの終わりに入れてください。

```
//BKOUT DD DSN=A123456.BOOKS1,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(3,3)),UNIT=3390
```

表 60 は、結果の出力データ・セット内のレコードの Book Title (ブック・タイトル) および Publisher (出版社) フィールドを示しています。実際のレコードにはすべてのフィールドが含まれています。

表 60. 5 冊以上の本が使用されている出版社からの本

| Book Title                            |    | Publisher |
|---------------------------------------|----|-----------|
| 1                                     | 75 | 106 109   |
| LIVING WELL ON A SMALL BUDGET         |    | COR       |
| SUPPLYING THE DEMAND                  |    | COR       |
| INKLINGS: AN ANTHOLOGY OF YOUNG POETS |    | COR       |
| PICK'S POCKET DICTIONARY              |    | COR       |
| MODERN ANTHOLOGY OF WOMEN POETS       |    | COR       |
| INTRODUCTION TO PSYCHOLOGY            |    | COR       |
| CRISES OF THE MIDDLE AGES             |    | COR       |
| VIDEO GAME DESIGN                     |    | VALD      |
| EDITING SOFTWARE MANUALS              |    | VALD      |
| STRATEGIC MARKETING                   |    | VALD      |
| SHORT STORIES AND TALL TALES          |    | VALD      |
| INTRODUCTION TO BIOLOGY               |    | VALD      |

複数の本を使用するコースと 1 冊の本だけを使用するコースの両方に関して、コース名と著者のラストネームだけを持つレコードを収めた別個の出力データ・セットを作成するには、次の SELECT ステートメントを作成します。



```

* Course name and author's last name for courses with more than one book
* and only one book
SELECT FROM(BKIN) TO(DUP) DISCARD(NODUP)

```

非重複出力データ・セットの DD 名  
 重複出力データ・セットの DD 名  
 入力データ・セットの DD 名

```

ON(120,25,CH) ALLDUPS USING(CTL1)

```

DFSORT 制御データ・セットの最初の 4 文字  
 レコードの選択基準  
 Course name (コース名)

BKIN は、サンプル・ブックストア・データ・セットの DD 名です。DUP は、複数の本を持つコースのレコードを収める出力データ・セットの DD 名です。NODUP は、本を 1 冊だけ使用するコースのレコードを収める出力データ・セットの DD 名です。

以下は、ジョブに必要な完全な JCL (制御ステートメントを含む) です。

```

//SEL2 JOB A492,PROGRAMMER
//EXTRACT EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//BKIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//TOOLIN DD *
* Course name and author's last name for courses with more than one book
* and only one book
SELECT FROM(BKIN) TO(DUP) DISCARD(NODUP) -
ON(120,25,CH) ALLDUPS USING(CTL1)
/*
//CTL1CNTL DD *
 OMIT COND=(120,25,CH,EQ,C' ')
 OUTFIL FNames=DUP,OUTREC=(120,25,2X,145,15)
 OUTFIL FNames=NODUP,OUTREC=(120,25,2X,145,15)
/*
//DUP DD SYSOUT=*
//NODUP DD SYSOUT=*

```

OMIT ステートメントは、SELECT 処理の前に、コース名がブランクのレコードを除去します。DUP に対する OUTFIL ステートメントは、複数の本を持つコースに関する選択レコードを再フォーマットし、コース名と著者のラストネームだけを持つようにします。NODUP に対する OUTFIL ステートメントは、1 冊の本だけを持つコースに関する選択レコードを再フォーマットし、コース名と著者のラストネームだけを持つようにします。

以下は、DUP レコードを現れるとおりに示しています。

|                    |            |
|--------------------|------------|
| INTRO TO COMPUTERS | CHATTERJEE |
| INTRO TO COMPUTERS | CHATTERJEE |
| INTRO TO COMPUTERS | CHATTERJEE |
| MODERN POETRY      | FRIEDMAN   |
| MODERN POETRY      | FRIEDMAN   |
| WORLD HISTORY      | GOODGOLD   |
| WORLD HISTORY      | WILLERTON  |

以下は、NODUP レコードを現れるとおりに示しています。

|                    |           |
|--------------------|-----------|
| ADVANCED MARKETING | LORCH     |
| BIOLOGY I          | GREENBERG |
| DATA MANAGEMENT    | SMITH     |
| EUROPEAN HISTORY   | BISCARDI  |
| FICTION WRITING    | BUCK      |
| MARKETING          | MAXWELL   |
| PSYCHOANALYSIS     | NAKATSU   |
| PSYCHOLOGY I       | ZABOSKI   |
| TECHINICAL EDITING | MADRID    |
| TECHNICAL EDITING  | MADRID    |
| VIDEO GAMES        | NEUMANN   |

"TECHINICAL EDITING" と "TECHNICAL EDITING" は異なるので ("TECHINICAL" と誤ってつづられたために)、この両方が NODUP データ・セットに含められたことに注意してください。

#### これまで

本章ではこれまで、数値フィールドの統計を印刷する方法、ソート済みデータ・セットと未ソート・データ・セットの作成方法、特定フィールドに関してある範囲内の数値フィールドのカウントを入手する方法、入力データ・セットからフィールドを印刷する方法、報告書の印刷方法、フィールド出現回数のカウントを印刷する方法、およびフィールド出現回数に基づいて出力レコードを選択する方法について学習しました。次に、ICETOOL の SPLICE オペレーターの使用法を学習します。

## いろいろなデータ・セットからのフィールドを結合する

ICETOOL の SPLICE オペレーターを使用すると、情報は異なるが同じ ON フィールド値を持つレコードの 50 までのフィールドを一緒に接合することにより、さまざまな方法で出力レコードを作成できます。出力レコードは、ON フィールド値によってソートされます。いろいろな入力データ・セットからのレコードを接合できるので、さまざまな「結合」および「突き合わせ」操作を実行できます。

最大 10 個の ON フィールドを使用できます。ON フィールド値はすべて、ソートと接合に使用されます。例えば、ON(1,4,CH) を使用する場合、'ABCD+01' と 'ABCD-01' はその順序で (1,4,CH,A によって) ソートされ、'ABCD' の出現回数 2 回としてカウントされます。しかし、ON(1,4,CH) と ON(5,3,FS) を使用する場合、'ABCD-01' は 'ABCD+01' より前に (1,4,CH,A および 5,3,FS,A によって) ソートされ、'ABCD-01' の出現回数 1 回および 'ABCD+01' の出現回数 1 回としてカウントされます。

2 つの入力データ・セットに結合または突き合わせ操作を行うには、接合するレコードのペアごとにフィールドを適切に位置合わせする必要があります。一般には、一方または両方の入力データ・セットを一時データ・セットにコピーし再フォーマットすることによってこれを行うので、一時データ・セットと一緒に接合することができます。

163 ページの表 61 と 163 ページの表 62 に示された 2 つの入力データ・セット REGION.IN1 と REGION.IN2 があると想定します。

表 61. 結合用の REGION.IN1 データ・セット

| Region | Headquarters | Regional Director |
|--------|--------------|-------------------|
| 1 5    | 6 20         | 21 35             |
| East   | Philadelphia | C. Kent           |
| West   | San Jose     | B. Wayne          |
| North  | Boston       | P. Parker         |
| South  | Charlotte    | D. Prince         |

表 62. 結合用の REGION.IN2 データ・セット

| Office | Region | Employees | Evaluation | Established |
|--------|--------|-----------|------------|-------------|
| 1 4    | 5 9    | 10 13     | 14 23      | 24 27       |
| 0001   | East   | 0050      | Fair       | 1983        |
| 0001   | South  | 0023      | Good       | 1976        |
| 0002   | South  | 0068      | Fair       | 1978        |
| 0002   | East   | 0125      | Excellent  | 1986        |
| 0001   | West   | 0052      | Good       | 1995        |
| 0003   | East   | 0028      | Good       | 1994        |
| 0002   | West   | 0105      | Excellent  | 2001        |
| 0003   | South  | 0054      | Fair       | 1992        |
| 0001   | North  | 0200      | Fair       | 1991        |
| 0004   | South  | 0070      | Good       | 2002        |

これらの 2 つの入力データ・セットから、出力データ・セット REGION.OUT を作成します。REGION.IN2 内のそれぞれのレコードごとに、REGION.IN1 内の対応する Region を検索し、2 つのレコードのフィールドを組み合わせて REGION.OUT の 1 つの出力レコードを形成します (表 63 を参照)。

表 63. 結合用の REGION.OUT データ・セット

| Office | Region | Regional Director | Employees | Evaluation | Headquarters |
|--------|--------|-------------------|-----------|------------|--------------|
| 1 4    | 5 9    | 10 24             | 25 28     | 29 38      | 39 53        |
| 0001   | East   | C. Kent           | 0050      | Fair       | Philadelphia |
| 0002   | East   | C. Kent           | 0125      | Excellent  | Philadelphia |
| 0003   | East   | C. Kent           | 0028      | Good       | Philadelphia |
| 0001   | North  | P. Parker         | 0200      | Fair       | Boston       |
| 0001   | South  | D. Prince         | 0023      | Good       | Charlotte    |
| 0002   | South  | D. Prince         | 0068      | Fair       | Charlotte    |
| 0003   | South  | D. Prince         | 0054      | Fair       | Charlotte    |
| 0004   | South  | D. Prince         | 0070      | Good       | Charlotte    |
| 0001   | West   | B. Wayne          | 0052      | Good       | San Jose     |
| 0002   | West   | B. Wayne          | 0105      | Excellent  | San Jose     |

REGION.IN1 および REGION.IN2 から REGION.OUT を作成するには、以下の ICETOOL ステートメントおよび JCL ステートメントを使用してください。

```
* Reformat REGION.IN1 to T1 so it can be spliced
COPY FROM(REGNIN1) TO(T1) USING(CTL1)
* Reformat REGION.IN2 to T1 so it can be spliced
COPY FROM(REGNIN2) TO(T1) USING(CTL2)
* Splice records in T1 with matching ON fields
SPICE FROM(T1) WITHALL -
```

```

 ON(5,5,CH) - Region
 WITH(1,4) - Office
 WITH(25,4) - Employees
 WITH(29,10) - Evaluation
 TO(REGNOUT)
/*
//REGNIN1 DD DSN=A123456.REGION.IN1,DISP=SHR
//REGNIN2 DD DSN=A123456.REGION.IN2,DISP=SHR
//T1 DD DSN=*&&T1,UNIT=3390,SPACE=(CYL,(5,5)),DISP=(MOD,PASS)
//REGNOUT DD DSN=A123456.REGION.OUT,DISP=(NEW,CATLG,DELETE),UNIT=3390,
// SPACE=(CYL,(5,5))
//CTL1CNTL DD *
* Move REGION.IN1 fields to their locations for the
* output data set
 OUTREC FIELDS=(5:1,5, Region
 10:21,15, Regional Director
 39:6,15) Headquarters
/*
//CTL2CNTL DD *
* Move REGION.IN2 fields to their locations for the
* output data set
 OUTREC FIELDS=(1:1,4, Office
 5:5,5, Region
 25:10,4, Employees
 29:14,10, Evaluation
 53:X)
/*

```

最初の COPY オペレーターは、表 64 に示されているように、REGION.IN1 からのフィールドを持つ一時データ・セット T1 にレコードを書き込みます。

表 64. REGION.IN1 からの T1 データ・セット・フィールド

| Blanks | Region | Regional Director | Blanks | Blanks | Headquarters |
|--------|--------|-------------------|--------|--------|--------------|
| 1 4    | 5 9    | 10 24             | 25 28  | 29 38  | 39 53        |

2 番目の COPY オペレーターは、表 65 に示されているように、REGION.IN2 からのフィールドを持つ一時データ・セット T1 の終わり (MOD) にレコードを書き込みます。

表 65. REGION.IN2 からの T1 データ・セット・フィールド

| Office | Region | Blanks | Employees | Evaluation | Blanks |
|--------|--------|--------|-----------|------------|--------|
| 1 4    | 5 9    | 10 24  | 25 28     | 29 38      | 39 53  |

Region フィールドは、SPLICE でレコードを突き合わせる際に使用する ON フィールドであるので、両方のタイプの再フォーマット済みレコードの同じ位置 (位置 5-9) に移動する必要があります。出力レコードに入りたいその他のフィールドはすべて、結合レコードの中で出力させたい位置に移動する必要があります。その他のタイプのレコード内で対応するフィールドには、レコードごとにブランクを入れます。例えば、再フォーマットされた REGION.IN1 レコードの位置 10-24 には、Regional Director フィールドが表示されますが、再フォーマットされた REGION.IN2 レコードの対応する位置 10-24 にはブランクが表示されます。また、再フォーマットされた REGION.IN2 レコードの位置 1-4 には、Office フィールドが表示されますが、再フォーマットされた REGION.IN1 レコードの対応する位置 1-4 にはブランクが表示されます。

FROM(T1) オペランドは、SPLICE 操作の入力が T1 データ・セットであることを DFSORT に指示します。このデータ・セットには、再フォーマットされた REGION.IN1 レコードと REGION.IN2 レコードがその順序で入っています。入力レコードの順序は重要です。というのは、それぞれの突き合わせ ON フィールドの最初のレコードを基底レコードとして、2 番目またはその後のレコードが接合されることになるからです。

この場合、再フォーマットされたすべての REGION.IN1 レコードは、再フォーマットされたすべての REGION.IN2 レコードの前にあります。したがって、SPLICE オペレーターは、ON フィールドが一致するときはいつでも、再フォーマットされた REGION.IN2 レコードからの WITH フィールドを、再フォーマットされた REGION.IN1 レコードに接合します。WITHALL は、最初のレコードを、ON フィールドが一致する他のすべてのレコードと接合するよう SPLICE に指示します。WITHALL を指定しなかった場合は、最初のレコードと最後のレコードだけが接合されます。

これは、使用可能な一部のオペランドを使用して、SPLICE で行える多数の結合操作タイプの一例にすぎません。

また、USING データ・セットを使用して、SPLICE 操作のための DFSORT INCLUDE、OMIT、INREC、および OUTFIL ステートメントを指定することもできます。INCLUDE または OMIT および INREC ステートメントの処理は、SPLICE 処理の前に実行されます。OUTFIL ステートメントの処理は、SPLICE 処理の後に実行されます。

SPLICE で使用できるすべてのオペランドの詳細、およびさまざまな結合操作の例については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

---

## いろいろなデータ・セットからのレコードを突き合わせる

SPLICE は、結合操作だけでなく突き合わせ操作にも使用できます。

SORT.SAMPIN と SORT.SAMPADD 内のコースを調べ、次のような 3 つの出力データ・セットを作成したいとします。

- COURSE.MATCH (SORT.SAMPIN と SORT.SAMPADD の両方に現れるコースを示すもの)
- COURSE.INONLY (SORT.SAMPIN のみに現れるコースを示すもの)
- COURSE.ADDONLY (SORT.SAMPADD のみに現れるコースを示すもの)

これらの 3 つのデータ・セットを作成する JCL および ICETOOL ステートメントは、次のとおりです。

```
//S1 EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//F1IN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//F2IN DD DSN=A123456.SORT.SAMPADD,DISP=SHR
//T1 DD DSN=%%T1,DISP=(MOD,PASS),UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//MATCH DD DSN=COURSE.MATCH,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(5,5)),UNIT=3390
//F1ONLY DD DSN=COURSE.INONLY,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(5,5)),UNIT=3390
//F2ONLY DD DSN=COURSE.ADDONLY,DISP=(NEW,CATLG,DELETE),
```

```

// SPACE=(CYL,(5,5)),UNIT=3390
//TOOLIN DD *
* Copy needed File1 fields to T1 and add '11' id
 COPY FROM(F1IN) TO(T1) USING(MATA)
* Copy needed File2 fields to T1 and add '22' id
 COPY FROM(F2IN) TO(T1) USING(MATB)
* Splice second id character from overlay to base for
* matching course numbers. Use resulting spliced or
* unspliced id to write records to MATCH, F1ONLY
* or F2ONLY as appropriate.
 SPLICE FROM(T1) TO(MATCH) ON(1,5,CH) -
 WITH(32,1) KEEPNOUDUPS USING(MATC)
/*
//MATACNTL DD *
* Extract needed File1 fields and add '11' id
 OUTREC FIELDS=(1:115,5, Course number
 6:120,25, Course name
 31:C'11') '11' id for File1 records
/*
//MATBCNTL DD *
* Extract needed File2 fields and add '22' id
 OUTREC FIELDS=(1:115,5, Course number
 6:120,25, Course name
 31:C'22') '22' id for File2 records
/*
//MATCCNTL DD *
* A '12' id indicates a match between File1 and File2.
* Remove the '12' id and write the records to MATCH.
 OUTFIL FNAMES=MATCH,INCLUDE=(31,2,CH,EQ,C'12'),OUTREC=(1,30)
* A '11' id indicates a record in File1 only.
* Remove the '11' id and write the records to F1ONLY.
 OUTFIL FNAMES=F1ONLY,INCLUDE=(31,2,CH,EQ,C'11'),OUTREC=(1,30)
* A '22' id indicates a record in File2 only.
* Remove the '22' id and write the records to F2ONLY.
 OUTFIL FNAMES=F2ONLY,INCLUDE=(31,2,CH,EQ,C'22'),OUTREC=(1,30)
/*

```

最初の COPY オペレーターは、表 66 に示されているように、SORT.SAMPIN からのコース番号 (Course Number) フィールドとコース名 (Course Name) フィールド、および '11' という ID からなる一時データ・セット T1 にレコードを書き込みます。

表 66. SORT.SAMPIN からの T1 データ・セット・フィールド

| Course Number | Course Name | '11'  |
|---------------|-------------|-------|
| 1 5           | 6 30        | 31 32 |

2 番目の COPY オペレーターは、表 67 に示されているように、SORT.SAMPADD からのコース番号 (Course Number) フィールドとコース名 (Course Name) フィールド、および '22' という ID からなる一時データ・セット T1 の終わり (MOD) にレコードを書き込みます。

表 67. SORT.SAMPADD からの T1 データ・セット・フィールド

| Course Number | Course Name | '22'  |
|---------------|-------------|-------|
| 1 5           | 6 30        | 31 32 |

SPLICE オペレーターは、コース番号 (ON フィールド) を突き合わせます。一致が見つかり、2 番目の ID 文字が基底レコードに接合されます。WITHALL が指定されていないので、それぞれの一致の最後のレコードだけがそれぞれの一致の最初

のレコードと接合されます。KEEPNODUPS は、接合されないレコードを保持するよう DFSORT に指示します。KEEPNODUPS を指定しなかった場合、一方のデータ・セットに 1 度だけ現れ、他方のデータ・セットには現れないコース番号に関するレコードは、COURSE.INONLY または COURSE.ADDONLY に書き込まれる代わりに削除されます。

以下に、可能なすべての一致と不一致の場合に何が起こるかを示しています。

- ID '11' で 1 回以上現れ、ID '22' でも 1 回以上現れるコース番号の場合は、最後の '22' レコードからの 2 番目の '2' が、最初の '11' レコードと接合され、'12' という ID が得られます。したがって、'12' という ID は、SORT.SAMPIN と SORT.SAMPADD の両方に現れるコース番号を表します。
- ID '11' で 1 回だけ現れる (ID '22' では現れない) コース番号の場合は、ID '11' は変更されません。ID '11' で複数回現れ (ID '22' では現れない) コース番号の場合は、最後の '11' レコードからの 2 番目の '1' が、最初の '11' レコードと接合され、'11' という ID が得られます。したがって、'11' という ID は、SORTIN.SAMPIN だけに現れるコース番号を表します。
- ID '22' で 1 回だけ現れる (ID '11' では現れない) コース番号の場合は、ID '22' は変更されません。ID '22' で複数回現れ (ID '11' では現れない) コース番号の場合は、最後の '22' レコードからの 2 番目の '2' が、最初の '22' レコードと接合され、'22' という ID が得られます。したがって、'22' という ID は、SORTIN.SAMPADD だけに現れるコース番号を表します。

SORT.SAMPIN と SORT.SAMPADD の両方に現れるコース番号に関する情報を含む COURSE.MATCH に対して作成される出力を、表 68 に示しています。

表 68. COURSE.MATCH 出力

| Course Number | Course Name        |
|---------------|--------------------|
| 1 5           | 6 30               |
| 00103         | DATA MANAGEMENT    |
| 00205         | VIDEO GAMES        |
| 10054         | FICTION WRITING    |
| 30016         | PSYCHOLOGY I       |
| 30975         | PSYCHOANALYSIS     |
| 50420         | WORLD HISTORY      |
| 50521         | WORLD HISTORY      |
| 50632         | EUROPEAN HISTORY   |
| 70124         | ADVANCED MARKETING |
| 70251         | MARKETING          |

SORT.SAMPIN だけに現れるコース番号に関する情報を含む COURSE.INONLY に対して作成される出力を、表 69 に示しています。

表 69. COURSE.INONLY 出力

| Course Number | Course Name |
|---------------|-------------|
| 1 5           | 6 30        |



表 69. COURSE.INONLY 出力 (続き)

| Course Number | Course Name        |
|---------------|--------------------|
| 00032         | INTRO TO COMPUTERS |
| 10347         | TECHNICAL EDITING  |
| 10856         | MODERN POETRY      |
| 80521         | BIOLOGY I          |

SORT.SAMPADD だけに現れるコース番号に関する情報を含む  
COURSE.ADDONLY に対して作成される出力を、表 70 に示しています。

表 70. COURSE.ADDONLY 出力

| Course Number | Course Name       |
|---------------|-------------------|
| 1 5           | 6 30              |
| 70255         | BUSINESS THEORY   |
| 80522         | BIOLOGY II        |
| 80523         | INTRO TO GENETICS |

#### これまで

本章ではこれまで、数値フィールドの統計を印刷する方法、ソート済みデータ・セットと未ソート・データ・セットの作成方法、特定フィールドに関してある範囲内の数値フィールドのカウントを入手する方法、入力データ・セットからフィールドを印刷する方法、報告書の印刷方法、フィールド出現回数のカウントを印刷する方法、フィールド出現回数に基づいて出力レコードを選択する方法、いろいろなデータ・セットのフィールドを結合する方法、およびいろいろなデータ・セットのレコードを突き合わせる方法について学習しました。本章の残りの部分では、完全なメイン ICETOOL ジョブとその結果として生じる TOOLMSG 出力を示します。



---

## 完全な ICETOOL ジョブと TOOLMSG 出力

以下に、本章で作成した完全な ICETOOL ジョブを示します。

---

```
//EXAMP JOB A492,PROGRAMMER
//TOOL EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//TOOLIN DD *
```

\* Statistics from all branches  
STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)  
\* Books from VALD and WETH  
SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(S PUB)  
\* Separate output for California and Colorado branches  
SORT FROM(ALL) USING(CACO)  
\* California branches profit analysis  
RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)  
\* Branches with less than 32 employees  
RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)  
\* Print profit, employees, and city for each Colorado branch  
DISPLAY FROM(CODASD) LIST(OUT) ON(28,6,PD) ON(18,4,ZD) ON(1,15,CH)  
\* Print a report for the Colorado branches  
DISPLAY FROM(CODASD) LIST(RPT) -  
DATE TITLE('Colorado Branches Report') PAGE -  
HEADER('City') HEADER('Profit') HEADER('Employees') -  
ON(1,15,CH) ON(28,6,PD) ON(18,4,ZD) BLANK BETWEEN(5) -  
TOTAL('Total') AVERAGE('Average') MINIMUM('Lowest')  
\* Print a report of books for individual publishers  
DISPLAY FROM(DAPUBS) LIST(SECTIONS) -  
TITLE('BOOKS FOR INDIVIDUAL PUBLISHERS') PAGE -  
HEADER('TITLE OF BOOK') ON(1,35,CH) -  
HEADER('PRICE OF BOOK') ON(170,4,BI,C1,F'\$') -  
BTITLE('PUBLISHER:') BREAK(106,4,CH) -  
BAVERAGE('AVERAGE FOR THIS PUBLISHER') -  
BTOTAL('TOTAL FOR THIS PUBLISHER') -  
AVERAGE('AVERAGE FOR ALL PUBLISHERS') -  
TOTAL('TOTAL FOR ALL PUBLISHERS')

---

図 5. 完全な ICETOOL ジョブ (1/3)

---

```

* Print the count of books in use from each publisher
OCCUR FROM(BKIN) LIST(PUBCT) BLANK -
 TITLE('Books from Publishers') DATE(DMY.) -
 HEADER('Publisher') HEADER('Books Used') -
 ON(106,4,CH) ON(VALCNT,N05)
* Separate output containing records for publishers
* with more than 4 books in use
SELECT FROM(BKIN) TO(BKOUT) ON(106,4,CH) HIGHER(4)
* Reformat REGION.IN1 to T1 so it can be spliced
COPY FROM(REGNIN1) TO(T1) USING(CTL1)
* Reformat REGION.IN2 to T1 so it can be spliced
COPY FROM(REGNIN2) TO(T1) USING(CTL2)
* Splice records in T1 with matching ON fields
SPLICE FROM(T1) WITHALL -
 ON(5,5,CH) - Region
 WITH(1,4) - Office
 WITH(25,4) - Employees
 WITH(29,10) - Evaluation
 TO(REGNOUT)

/*
//ALL DD DSN=A123456.SORT.BRANCH,DISP=SHR
//BKS DD DSN=A123456.SORT.SAMPIN,DISP=SHR
// DD DSN=A123456.SORT.SAMPADD,DISP=SHR
//DAPUBS DD DSN=&&DSRT,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=SYSDA
//PRPUBS DD SYSOUT=A
//SPUBCNTL DD *
SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),
FORMAT=CH

/*
//CACOCNTL DD *
SORT FIELDS=(1,15,CH,A)
OUTFIL FNAMES=(CADASD,CATAPE),INCLUDE=(16,2,CH,EQ,C'CA')
OUTFIL FNAMES=(CODASD,COTAPE),INCLUDE=(16,2,CH,EQ,C'CO')

/*
//CADASD DD DSN=&&CA,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//CATAPE DD DSN=CA.BRANCH,UNIT=3480,VOL=SER=111111,
// DISP=(NEW,KEEP),LABEL=(,SL)
//CODASD DD DSN=&&CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//COTAPE DD DSN=CO.BRANCH,UNIT=3480,VOL=SER=222222,
// DISP=(NEW,KEEP),LABEL=(,SL)
//OUT DD SYSOUT=A
//RPT DD SYSOUT=A
//SECTIONS DD SYSOUT=A
//BKIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//PUBCT DD SYSOUT=A
//BKOUT DD DSN=A123456.BOOKS1,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(3,3)),UNIT=3390
//REGNIN1 DD DSN=A123456.REGION.IN1,DISP=SHR
//REGNIN2 DD DSN=A123456.REGION.IN2,DISP=SHR
//T1 DD DSN=&&T1,UNIT=3390,SPACE=(CYL,(5,5)),DISP=(MOD,PASS)
//REGNOUT DD DSN=A123456.REGION.OUT,DISP=(NEW,CATLG,DELETE),UNIT=3390,
// SPACE=(CYL,(5,5))
//CTL1CNTL DD *
* Move REGION.IN1 fields to their locations for the
* output data set
OUTREC FIELDS=(5:1,5, Region
 10:21,15, Regional Director
 39:6,15) Headquarters

/*

```

---

図5. 完全な ICETOOL ジョブ (2/3)

---

```
//CTL2CNTL DD *
* Move REGION.IN2 fields to their locations for the
* output data set
 OUTREC FIELDS=(1:1,4, Office
 5:5,5, Region
 25:10,4, Employees
 29:14,10, Evaluation
 53:X)
/*
```

---

図5. 完全な ICETOOL ジョブ (3/3)

以下は、このジョブを実行することで作成される完全な TOOLMSG データ・セットです。

---

```
ICE600I 0 DFSORT ICETOOL UTILITY RUN STARTED

ICE650I 0 VISIT http://www.ibm.com/storage/dfsor FOR ICETOOL PAPERS, EXAMPLES AND MORE

ICE632I 0 SOURCE FOR ICETOOL STATEMENTS: TOOLIN

ICE630I 0 MODE IN EFFECT: STOP

 * Statistics from all branches
 STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
ICE627I 0 DFSORT CALL 0001 FOR COPY FROM ALL TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000012
ICE607I 0 STATISTICS FOR (18,4,ZD) :
ICE608I 0 MINIMUM: +000000000000015, MAXIMUM: +000000000000035
ICE609I 0 AVERAGE: +000000000000024, TOTAL : +000000000000298
ICE607I 0 STATISTICS FOR (28,6,PD) :
ICE608I 0 MINIMUM: -0000000000004278, MAXIMUM: +0000000000008276
ICE609I 0 AVERAGE: +0000000000004222, TOTAL : +00000000000050665
ICE607I 0 STATISTICS FOR (22,6,PD) :
ICE608I 0 MINIMUM: +0000000000012300, MAXIMUM: +00000000000042820
ICE609I 0 AVERAGE: +0000000000027469, TOTAL : +00000000000329637
ICE602I 0 OPERATION RETURN CODE: 00

 * Books from VALD and WETH
 SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
ICE606I 0 DFSORT CALL 0002 FOR SORT FROM BKS TO OUTFIL USING SPUBCNTL COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

 * Separate output for California and Colorado branches
 SORT FROM(ALL) USING(CACO)
ICE606I 0 DFSORT CALL 0003 FOR SORT FROM ALL TO OUTFIL USING CACOCNTL COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

 * California branches profit analysis
 RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
ICE627I 0 DFSORT CALL 0004 FOR COPY FROM CADASD TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000007
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (28,6,PD) : 000000000000003
ICE602I 0 OPERATION RETURN CODE: 00

 * Branches with less than 32 employees
 RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)
ICE627I 0 DFSORT CALL 0005 FOR COPY FROM ALL TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000012
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (18,4,ZD) : 000000000000008
ICE602I 0 OPERATION RETURN CODE: 00
```

---

図6. 完全な TOOLMSG データ・セット (1/3)

---

```

* Print profit, employees, and city for each Colorado branch
 DISPLAY FROM(CODASD) LIST(OUT) ON(28,6,PD) ON(18,4,ZD) ON(1,15,CH)
ICE643I 0 WIDTH OF REPORT IS 0121 BYTES
ICE627I 0 DFSORT CALL 0006 FOR COPY FROM CODASD TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN OUT DATA SET
ICE628I 0 RECORD COUNT: 000000000000005
ICE602I 0 OPERATION RETURN CODE: 00

 * Print a report for the Colorado branches
 DISPLAY FROM(CODASD) LIST(RPT) -
 DATE TITLE('Colorado Branches Report') PAGE -
 HEADER('City') HEADER('Profit') HEADER('Employees') -
 ON(1,15,CH) ON(28,6,PD) ON(18,4,ZD) BLANK BETWEEN(5) -
 TOTAL('Total') AVERAGE('Average') MINIMUM('Lowest')
ICE643I 0 WIDTH OF REPORT IS 0121 BYTES
ICE627I 0 DFSORT CALL 0007 FOR COPY FROM CODASD TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN RPT DATA SET
ICE628I 0 RECORD COUNT: 000000000000005
ICE602I 0 OPERATION RETURN CODE: 00

 * Print a report of books for individual publishers
 DISPLAY FROM(DAPUBS) LIST(SECTIONS) -
 TITLE('BOOKS FOR INDIVIDUAL PUBLISHERS') PAGE -
 HEADER('TITLE OF BOOK') ON(1,35,CH) -
 HEADER('PRICE OF BOOK') ON(170,4,BI,C1,F'$') -
 BTITLE('PUBLISHER:') BREAK(106,4,CH) -
 BAVERAGE('AVERAGE FOR THIS PUBLISHER') -
 BTOTAL('TOTAL FOR THIS PUBLISHER') -
 AVERAGE('AVERAGE FOR ALL PUBLISHERS') -
 TOTAL('TOTAL FOR ALL PUBLISHERS')
ICE643I 0 WIDTH OF REPORT IS 0121 BYTES
ICE627I 0 DFSORT CALL 0008 FOR COPY FROM DAPUBS TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN SECTIONS DATA SET
ICE628I 0 RECORD COUNT: 000000000000019
ICE602I 0 OPERATION RETURN CODE: 00

 * Print the count of books in use from each publisher
 OCCUR FROM(BKIN) LIST(PUBCT) BLANK -
 TITLE('Books from Publishers') DATE(DMY.) -
 HEADER('Publisher') HEADER('Books Used') -
 ON(106,4,CH) ON(VALCNT)
ICE643I 0 WIDTH OF REPORT IS 0121 BYTES
ICE627I 0 DFSORT CALL 0009 FOR SORT FROM BKIN TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN PUBCT DATA SET
ICE628I 0 RECORD COUNT: 000000000000020
ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 000000000000004
ICE602I 0 OPERATION RETURN CODE: 00

 * Separate output containing records for publishers
 * with more than 4 books in use
 SELECT FROM(BKIN) TO(BKOUT) ON(106,4,CH) HIGHER(4)
ICE627I 0 DFSORT CALL 0010 FOR SORT FROM BKIN TO BKOUT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000020
ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 000000000000012
ICE602I 0 OPERATION RETURN CODE: 00

 * Reformat REGION.IN1 to T1 so it can be spliced
 COPY FROM(REGNIN1) TO(T1) USING(CTL1)
ICE606I 0 DFSORT CALL 0011 FOR COPY FROM REGNIN1 TO T1 USING CTL1CNTL COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

ICE601I 0 DFSORT ICETOOL UTILITY RUN ENDED - RETURN CODE: 00

```

---

図6. 完全な TOOLMSG データ・セット (2/3)

---

```
 * Reformat REGION.IN2 to T1 so it can be spliced
 COPY FROM(REGNIN2) TO(T1) USING(CTL2)
ICE606I 0 DFSORT CALL 0012 FOR COPY FROM REGNIN2 TO T1 USING CTL2CNTL COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

 * Splice records in T1 with matching ON fields
 SPLICE FROM(T1) WITHALL -
 ON(5,5,CH) - Region
 WITH(1,4) - Office
 WITH(25,4) - Employees
 WITH(29,10) - Evaluation
 TO(REGNOUT)
ICE627I 0 DFSORT CALL 0013 FOR SORT FROM T1 TO REGNOUT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000014
ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 000000000000010
ICE602I 0 OPERATION RETURN CODE: 00
```

---

図 6. 完全な *TOOLMSG* データ・セット (3/3)

#### 要約

本章では、13 個の ICETOOL オペレーターを紹介し、これらのオペレーターを使用してさまざまなタスクを実行する方法について説明しました。DFSORT の ICETOOL の完全な情報については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。



---

## 第 4 部 シンボルの使い方を学ぶ

|                                                |     |
|------------------------------------------------|-----|
| 第 12 章 シンボルの定義および使用 . . . . .                  | 177 |
| SYMNAMES データ・セットの作成 . . . . .                  | 177 |
| フィールドを表すシンボルの定義 . . . . .                      | 178 |
| DFSORT ステートメントでフィールドにシンボルを<br>使用する方法 . . . . . | 179 |
| ICETOOL オペレーターでフィールドにシンボルを<br>使用する方法 . . . . . | 180 |
| 定数を表すシンボルの定義と使用 . . . . .                      | 181 |





---

## 第 12 章 シンボルの定義および使用

シンボル とは、フィールドまたは定数を表すのに使用できる名前 (何か意味のあるものが望ましい) です。マッピングとも呼ばれるシンボルのセットを使用して、特定のタイプのレコードの情報のような、関連するフィールドおよび定数のグループを記述することができます。このようなマッピングを使用することにより、フィールド、定数、および出力桁をそのシンボルで参照できるので、使用したいフィールドの位置、長さ、およびフォーマット、定数の値、または出力桁の位置を知っていなくても済むようになります。

ユーザーまたはご使用のサイトで DFSORT のシンボル処理機能を使用することによって、ユーザー独自のレコードのフィールドを表すシンボルおよびこれらのフィールドに関連した定数を表すシンボルを作成できます。そうすると、DFSORT 制御ステートメントおよび ICETOOL オペレーターでそれらのシンボルを使用できるようになります。

さらに、RACF<sup>®</sup>、DCOLLECT、および DFSMSrmm<sup>™</sup> のような他のプロダクトで作成されたレコードに使用するシンボルの事前定義セットを入手することもできます。他のプロダクトで作成されたレコードの DFSORT シンボル・マッピングの入手方法、およびこれらのシンボルを使用する例については、次の URL にある DFSORT ホーム・ページにアクセスしてください。

<http://www.ibm.com/storage/dfsor>t

この節の章では、ブックストア・データ・セットのシンボルを定義する方法と、それらをいくつかの DFSORT 制御ステートメントおよび ICETOOL オペレーターで使用方法について説明します。本書では記述されていませんが、シンボル定数でシステム・シンボル (例えば、&SYSPLEX. や &JOBNAME.) を使用することもできます。シンボル (システム・シンボルを含む) を DFSORT および ICETOOL でどのように使用できるかについて詳しくは、「z/OS DFSORT アプリケーション・プログラミング・ガイド」を参照してください。

---

### SYMNAMES データ・セットの作成

DFSORT および ICETOOL は、SYMNAMES DD ステートメントで指定されたデータ・セットから、使用すべきシンボルを入手します。DFSORT JCL および制御ステートメントを含むデータ・セットを作成するのと同じ方法で、使用したい SYMNAMES データ・セットを、RECFM=FB および LRECL=80 で作成してください。次に、ISPF EDIT のようなエディターを使用して、下記に示すように、シンボルを定義する SYMNAMES ステートメントを作成してください。

SYMNAMES データ・セットを作成した後は、定義したシンボルを使用したい対象となる任意の DFSORT または ICETOOL アプリケーションでそのデータ・セットを使用できます。エディターを使用することにより、いつでも SYMNAMES データ・セットに SYMNAMES ステートメントを追加したり、SYMNAMES ステートメントを削除または変更することができます。

本章では、SYMNAMES ステートメントを入れるのに、SORT.SYMBOLS という名前のデータ・セットを作成したと想定しています。

## フィールドを表すシンボルの定義

189 ページの『付録 B. サンプル・データ・セットの記述』に、ブックストア・データ・セットのフィールドを示しています。これらのフィールドを表すシンボルを定義するには、それぞれのシンボルに対する SYMNAMES ステートメントを SORT.SYMBOLS 内に作成します。シンボル Title および Author\_Last\_Name に対する SYMNAMES ステートメントは、次のようになります。

```
* Symbols for fields
Title,1,75,CH
Author_Last_Name,*,15,CH
```

Title フィールドのシンボル Title および Author's Last Name フィールドのシンボル Author\_Last\_Name を定義する SYMNAMES ステートメントを作成するには、以下のステップに従ってください。

表 71. フィールドを表すシンボルの定義ステップ

| ステップ | アクション                                                                                                                                                                                                                                                |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | コメント・ステートメントを作成する (オプション)。 <p>* Symbols for fields</p>                                                                                                                                                                                               |
| 2    | <b>Title</b> を入力し、その後にコンマを続ける。 <p>これは、Title フィールドに使用するシンボルとなります。シンボルは 1 から 50 文字で、大文字 (A-Z)、小文字 (a-z)、数値 (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 (-) から構成することができます。ただし、先頭文字は数値であってはなりません。Title、TITLE、および title は、3 つの異なるシンボルです。</p>       |
| 3    | Title フィールドの位置を入力し、その後にコンマを続ける。 <p>Title フィールドの位置は <b>1</b> です。</p>                                                                                                                                                                                  |
| 4    | Title フィールドの長さを入力し、その後にコンマを続ける。 <p>Title フィールドの長さは <b>75</b> です。</p>                                                                                                                                                                                 |
| 5    | Title フィールドのフォーマットを入力し、その後にブランクを続ける。 <p>Title フィールドのフォーマットは <b>CH</b> です。</p>                                                                                                                                                                        |
| 6    | <b>Author_Last_Name</b> を入力し、その後にコンマを続ける。 <p>これは、Author's Last Name フィールドに使用するシンボルとなります。</p>                                                                                                                                                         |
| 7    | 位置の代わりに * を入力し、その後にコンマを続ける。 <p>位置を表すアスタリスク (*) は、このフィールドが前のフィールドの直後に続くことを示します。この場合、* は、Author_Last_Name シンボルの位置として自動的に <b>76</b> を割り当てます。* の代わりに 76 を指定することもできますが、フィールドが隣接しているときは * の方が望ましい指定です。* を使用すると、フィールドのシンボルが他のフィールドのシンボルを変更することなく挿入されます。</p> |

表 71. フィールドを表すシンボルの定義ステップ (続き)

| ステップ | アクション                                                                                                   |
|------|---------------------------------------------------------------------------------------------------------|
| 8    | Author's Last Name フィールドの長さを入力し、その後にコンマを続ける。<br><br>Author's Last Name フィールドの長さは <b>15</b> です。          |
| 9    | Author's Last Name フィールドのフォーマットを入力し、その後にブランクを続ける。<br><br>Author's Last Name フィールドのフォーマットは <b>CH</b> です。 |

ブックストア・データ・セット内のその他のフィールドのシンボルを定義する SYMNAMES ステートメントは、次のようになります。

```
Author_First_Name,*,15,CH
Publisher,*,4,CH
Course_Department,*,5,CH
Course_Number,*,5,CH
Course_Name,*,25,CH
Instructor_Last_Name,*,15,CH
Instructor_Initials,*,2,CH
Number_in_Stock,*,4,BI
Number_Sold_YTD,*,4,BI
Price,*,4,BI
```

#### これまで

これまで、SYMNAMES データ・セットの作成方法およびこれを使用してフィールドのシンボルを定義する方法を学習しました。次に、定義したフィールドのシンボルの使用方法を学習します。

## DFSORT ステートメントでフィールドにシンボルを使用する方法

ブックストア・データ・セットのシンボルを SORT.SYMBOLS に定義したので、DFSORT 制御ステートメントにおいてフィールドが現れるところではどこでも、これらのシンボルを使用することができます。

以下に、シンボルを使用する DFSORT アプリケーションの例を示します。このアプリケーションは、SORT.SAMPIN からコース 00032 および 10347 に関する本を選択し、それらをタイトル、インストラクター、および価格でソートします。

```
//SYM1 JOB A492,PROGRAMMER
//SORTIT EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SYMNAMES DD DSN=A123456.SORT.SYMBOLS,DISP=SHR
//SYMNOUT DD SYSOUT=*
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//SORTOUT DD DSN=A123456.SORT.SAMPOUT,DISP=SHR
//SYSIN DD *
 INCLUDE COND=(Course_Number,EQ,C'00032',OR,
 Course_Number,EQ,C'10347')
 SORT FIELDS=(Title,A,
 Instructor_Last_Name,A,Instructor_Initials,A,
 Price,A)
/*
```

SYMNAMES DD ステートメントは、このアプリケーションに使用する SYMNAMES データ・セットを指定します。

SYMNOUT DD ステートメントは、DFSORT に元の SYMNAMES ステートメントとそれらから作成したシンボル・テーブルをリストさせる先のデータ・セットを指定します。この情報を表示する必要がない場合は、SYMNOUT データ・セットを省略できます。この SYMNAMES データ・セットの場合、SYMNOUT リストは次のようになります。

```
----- ORIGINAL STATEMENTS FROM SYMNAMES -----
* Symbols for fields
Title,1,75,CH
Author_Last_Name,*,15,CH
Author_First_Name,*,15,CH
Publisher,*,4,CH
Course_Department,*,5,CH
Course_Number,*,5,CH
Course_Name,*,25,CH
Instructor_Last_Name,*,15,CH
Instructor_Initials,*,2,CH
Number_in_Stock,*,4,BI
Number_SoTd_YTD,*,4,BI
Price,*,4,BI

----- SYMBOL TABLE -----
Title,1,75,CH
Author_Last_Name,76,15,CH
Author_First_Name,91,15,CH
Publisher,106,4,CH
Course_Department,110,5,CH
Course_Number,115,5,CH
Course_Name,120,25,CH
Instructor_Last_Name,145,15,CH
Instructor_Initials,160,2,CH
Number_in_Stock,162,4,BI
Number_SoTd_YTD,166,4,BI
Price,170,4,BI
```

INCLUDE および SORT ステートメントは、位置、長さ、およびフォーマットの代わりに、フィールドにシンボルを使用しています。

使用する任意の %nn 解析済みフィールドに対してシンボルを定義することができます。例えば、次の SYMNAMES ステートメントを指定して、%01 および %02 に対してシンボルを定義できます。

```
First_name,%01
Last_name,%02
```

その後、これらのシンボルを次の OUTREC ステートメントで使用できます。

```
OUTREC PARSE=(First_name=(ABSPOS=9, FIXLEN=12, ENDBEFR=C', '),
 Last_name=(FIXLEN=15, ENDBEFR=C', ')),
 BUILD=(First_name,15:Last_name)
```

---

## ICETOOL オペレーターでフィールドにシンボルを使用する方法

ICETOOL オペレーターおよび関連する DFSORT 制御ステートメントにおいても、フィールドが現れるところではどこでも、SORT.SYMBOLS からのシンボルを使用することができます。以下に、フィールドのシンボルを使用する ICETOOL アプリケーションの例を示しています。このアプリケーションは、179 ページの

『DFSORT ステートメントでフィールドにシンボルを使用する方法』の DFSORT 例と同じことを行うほか、選択された本のうち価格が 7 ドルより下の冊数と 20 ドルより上の冊数を表示します。

```
//SYM2 JOB A492,PROGRAMMER
//TOOL EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//SYMNAMES DD DSN=A123456.SORT.SYMBOLS,DISP=SHR
//SYMNOUT DD SYSOUT=*
//IN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//OUT DD DSN=A123456.SORT.SAMPOUT,DISP=OLD
//TOOLIN DD *
 SORT FROM(IN) TO(OUT) USING(CTL1)
 RANGE FROM(OUT) ON(Price) LOWER(+700)
 RANGE FROM(OUT) ON(Price) HIGHER(+2000)
//CTL1CNTL DD *
 INCLUDE COND=(Course_Number,EQ,C'00032',OR,
 Course_Number,EQ,C'10347')
 SORT FIELDS=(Title,A,
 Instructor_Last_Name,A,Instructor_Initials,A,
 Price,A)
/*
```

SYMNAMES DD ステートメントは、このアプリケーションに使用する SYMNAMES データ・セットを指定します。

SYMNOUT DD ステートメントは、ICETOOL に元の SYMNAMES ステートメントとそれらから作成したシンボル・テーブルをリストさせる先のデータ・セットを指定します。SYMNAMES データ・セットは、179 ページの『DFSORT ステートメントでフィールドにシンボルを使用する方法』で示された DFSORT アプリケーションの場合と同じであるので、SYMNOUT リストも同じになります。

RANGE オペレーターと INCLUDE および SORT ステートメントは、位置、長さ、およびフォーマットの代わりに、フィールドにシンボルを使用しています。

#### これまで

DFSORT 制御ステートメントおよび ICETOOL オペレーターにおいてフィールドの代わりにシンボルを使用する方法を学習しました。次に、定数を表すシンボルの定義方法と使用方法を学習します。

## 定数を表すシンボルの定義と使用

DFSORT 制御ステートメントおよび ICETOOL オペレーターにおいて、10 進定数、文字定数、16 進定数、またはビット定数が現れるところではどこでも、シンボルを使用することができます。

180 ページの『ICETOOL オペレーターでフィールドにシンボルを使用する方法』の ICETOOL 例は、次の RANGE オペレーターと

```
RANGE FROM(OUT) ON(Price) LOWER(+700)
RANGE FROM(OUT) ON(Price) HIGHER(+2000)
```

次の INCLUDE ステートメントを使用しています。

```
INCLUDE COND=(Course_Number,EQ,C'00032',OR,
 Course_Number,EQ,C'10347')
```

+700 と +2000 は 10 進定数です。C'00032' と C'10347' は文字定数です。これらの定数の代わりにシンボルを使用するようにすると、作成する ICETOOL オペレーターおよび DFSORT 制御ステートメントでそれらが分かりやすくなります。これらの定数に対する SYMNAMES ステートメントは、次のようになります。

```
* Symbols for constants
Discount,+700
Premium,+2000
Beginning_Economics,C'00032'
Advanced_Sociology,C'10347'
```

4 つの定数を表すこれらのシンボルを SORT.SYMBOLS データ・セットに追加するには、以下のステップに従ってください。

表 72. 定数を表すシンボルの定義ステップ

| ステップ | アクション                                                                                                |
|------|------------------------------------------------------------------------------------------------------|
| 1    | コメント・ステートメントを作成する (オプション)。 <p>* Symbols for constants</p>                                            |
| 2    | <b>Discount</b> を入力し、その後にコンマを続ける。 <p>これは、10 進定数 +700 に使用するシンボルとなります。</p>                             |
| 3    | 定数を入力し、その後にブランクを続ける。 <p>定数は <b>+700</b> です。<b>700</b> も使用できますが、正の 10 進定数には + 符号を使用することをお勧めします。</p>   |
| 4    | <b>Premium</b> を入力し、その後にコンマを続ける。 <p>これは、10 進定数 <b>+2000</b> に使用するシンボルとなります。</p>                      |
| 5    | 定数を入力し、その後にブランクを続ける。 <p>定数は <b>+2000</b> です。<b>2000</b> も使用できますが、正の 10 進定数には + 符号を使用することをお勧めします。</p> |
| 6    | <b>Beginning_Economics</b> を入力し、その後にコンマを続ける。 <p>これは、文字定数 C'00032' に使用するシンボルとなります。</p>                |
| 7    | 定数を入力し、その後にブランクを続ける。 <p>定数は <b>C'00032'</b> です。<b>'00032'</b> または <b>c'00032'</b> も使用できます。</p>       |
| 8    | <b>Advanced_Sociology</b> を入力し、その後にコンマを続ける。 <p>これは、文字定数 C'10347' に使用するシンボルとなります。</p>                 |
| 9    | 定数を入力し、その後にブランクを続ける。 <p>定数は <b>C'10347'</b> です。<b>'10347'</b> または <b>c'10347'</b> も使用できます。</p>       |

これで、RANGE オペレーターは次のように書くことができます。

```
RANGE FROM(OUT) ON(Price) LOWER(Discount)
RANGE FROM(OUT) ON(Price) HIGHER(Premium)
```

INCLUDE ステートメントは次のように書くことができます。

```
INCLUDE COND=(Course_Number,EQ,Beginning_Economics,OR,
 Course_Number,EQ,Advanced_Sociology)
```

#### 要約

本章では、フィールドおよび定数を表すシンボルを定義し、それらを DFSORT 制御ステートメントおよび ICETOOL オペレーターで使用方法を説明しました。DFSORT シンボルの詳細については、「*z/OS DFSORT アプリケーション・プログラミング・ガイド*」を参照してください。

これで、「*z/OS DFSORT: ご使用の手引き*」のチュートリアルが完了しました。この後の付録には、サンプル・データ・セットに関する重要な情報および DFSORT が各種の制御ステートメントを処理する順序を記載しています。





---

## 第 5 部 付録



---

## 付録 A. サンプル・データ・セットの作成

本書の多くの例では、入力データ・セットとして A123456.SORT.SAMPIN、A123456.SORT.SAMPADD、および A123456.SORT.BRANCH を使用し、出力データ・セットとして A123456.SORT.SAMPOUT を使用しています。これらのデータ・セットを使用する本書内の例を試してみたい場合は、以下の ICEDCRE ジョブを実行することにより、自分のユーザー ID を持つ独自の userid.SORT.SAMPIN、userid.SORT.SAMPADD、userid.SORT.BRANCH、および userid.SORT.SAMPOUT データ・セットを作成してください。

ICEDCRE JOB ステートメントの代わりにユーザー自身の JOB ステートメントを使用します。4 つの DD ステートメントでは 'userid' の代わりにユーザー自身のユーザー ID を使用します。

DFSORT ジョブを実行するのに JOBLIB または STEPLIB を使用する場合は、JOBLIB または STEPLIB を ICEDCRE ジョブに追加します。SYSALLDA ではなく、UNIT に値を入りたい場合は、DD ステートメントの該当パラメーターを変更します。また、該当する場合には、DD ステートメントに VOL=SER=vvvvvv パラメーターを追加します。

```
//ICEDCRE JOB A492,PROGRAMMER
//SAMPCOPY EXEC PGM=ICESAMP,PARM=(BOOKS)
//SYSPRINT DD SYSOUT=*
//SAMPLE DD DSN=userid.SORT.SAMPIN,DISP=(NEW,CATLG),
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=173,BLKSIZE=1730),
// UNIT=SYSALLDA
//*
//ADD DD DSN=userid.SORT.SAMPADD,DISP=(NEW,CATLG),
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=173,BLKSIZE=1730),
// UNIT=SYSALLDA
//*
//OUTPUT DD DSN=userid.SORT.SAMPOUT,DISP=(NEW,CATLG),
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=173,BLKSIZE=1730),
// UNIT=SYSALLDA
//*
//BRANCH DD DSN=userid.SORT.BRANCH,DISP=(NEW,CATLG),
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=33,BLKSIZE=330),
// UNIT=SYSALLDA
//*
//SYSIN DD DUMMY
```

このジョブが正常に実行された後で、これらの 4 つのデータ・セットのデータ・セット名の A123456 を自分のユーザー ID で置き換えることができます。例えば、次のような DD ステートメントの場合は、

```
//SORTIN DD DSN=A123456.SORT.SAMPIN,DISP=SHR
```

代わりに、次のようなユーザー独自の DD ステートメントを使用することができます。

```
//SORTIN DD DSN=userid.SORT.SAMPIN,DISP=SHR
```

## サンプル・データ・セットの使用

注: 一部の例では、A123456.SORT.SAMPIN、A123456.SORT.SAMPOUT、A123456.SORT.SAMPADD、および A123456.SORT.BRANCH 以外のデータ・セットが使用されています。テキストで使用されているデータ・セットと一致させるには最初からデータ・セットを作成することもできますし、すでにお持ちのデータ・セットに同様の処理を行うこともできます。

## 付録 B. サンプル・データ・セットの記述

サンプル・データ・セット SORT.SAMPIN、SORT.SAMPADD、および SORT.BRANCH のレコードのフィールドをこのページに、レコードの内容を後続ページに記載しています。

表 73 は、サンプル・ブックストア・データ・セット SORT.SAMPIN および追加のブックストア・データ・セット SORT.SAMPADD の各フィールドの長さやデータ・フォーマットを示しています。これらのデータ・セットは両方とも、173 バイトの長さ (LRECL=133) の固定長レコード (RECFM=FB) です。

表 73. SORT.SAMPIN および SORT.SAMPADD のフィールド記述

| フィールド                  | 長さ | データ・フォーマット |
|------------------------|----|------------|
| Title                  | 75 | CH         |
| Author's Last Name     | 15 | CH         |
| Author's First Name    | 15 | CH         |
| Publisher              | 4  | CH         |
| Course Department      | 5  | CH         |
| Course Number          | 5  | CH         |
| Course Name            | 25 | CH         |
| Instructor's Last Name | 15 | CH         |
| Instructor's Initials  | 2  | CH         |
| Number In Stock        | 4  | BI         |
| Number Sold Y-to-D     | 4  | BI         |
| Price                  | 4  | BI         |

表 74 は、サンプル・ブランチ・データ・セット SORT.BRANCH の各フィールドの長さやデータ・フォーマットを示しています。このデータ・セットは、33 バイトの長さ (LRECL=33) の固定長レコード (RECFM=FB) です。

表 74. SORT.BRANCH のフィールド記述

| フィールド     | 長さ | データ・フォーマット |
|-----------|----|------------|
| City      | 15 | CH         |
| State     | 2  | CH         |
| Employees | 4  | ZD         |
| Revenue   | 6  | PD         |
| Profit    | 6  | PD         |

このサンプル・データ・セットのレコードの内容は後続ページに示されています。横の各行はそれぞれ 1 つのレコードを表し、各列はそれぞれ 1 つのフィールドを表します。見出しは、各フィールドのフィールド名、開始位置、および終了位置を示します。もちろん、これらの見出しがデータ・セット内で実際に現れるわけでは

## サンプル・ブックストア・データ・セット

ありません。数値フィールド (BI、ZD、PD) は、表示された場合は実際には判読可能ではありませんが、等価の判読可能値として示されています。

### サンプル・データ・セット - SORT.SAMPIN

| Book Title                            |    | Author's<br>Last Name |    | Author's<br>First Name |     | Publisher |     | Course<br>Department |     |
|---------------------------------------|----|-----------------------|----|------------------------|-----|-----------|-----|----------------------|-----|
| 1                                     | 75 | 76                    | 90 | 91                     | 105 | 106       | 109 | 110                  | 114 |
| COMPUTER LANGUAGES                    |    | MURRAY                |    | ROBERT                 |     | FERN      |     | COMP                 |     |
| LIVING WELL ON A SMALL BUDGET         |    | DEWAN                 |    | FRANK                  |     | COR       |     |                      |     |
| SUPPLYING THE DEMAND                  |    | MILLER                |    | TOM                    |     | COR       |     | BUSIN                |     |
| VIDEO GAME DESIGN                     |    | RASMUSSEN             |    | LORI                   |     | VALD      |     | COMP                 |     |
| INKLINGS: AN ANTHOLOGY OF YOUNG POETS |    | WILDE                 |    | KAREN                  |     | COR       |     | ENGL                 |     |
| COMPUTERS: AN INTRODUCTION            |    | DINSHAW               |    | JOKII                  |     | WETH      |     | COMP                 |     |
| PICK'S POCKET DICTIONARY              |    | GUSTLIN               |    | CAROL                  |     | COR       |     |                      |     |
| EDITING SOFTWARE MANUALS              |    | OJALVO                |    | VICTOR                 |     | VALD      |     | ENGL                 |     |
| NUMBERING SYSTEMS                     |    | BAYLESS               |    | WILLIAM                |     | FERN      |     | COMP                 |     |
| STRATEGIC MARKETING                   |    | YAEGER                |    | MARK                   |     | VALD      |     | BUSIN                |     |
| THE INDUSTRIAL REVOLUTION             |    | GROSS                 |    | DON                    |     | WETH      |     | HIST                 |     |
| MODERN ANTHOLOGY OF WOMEN POETS       |    | COWARD                |    | PETER                  |     | COR       |     | ENGL                 |     |
| INTRODUCTION TO PSYCHOLOGY            |    | DUZET                 |    | LINDA                  |     | COR       |     | PSYCH                |     |
| THE COMPLETE PROOFREADER              |    | GREEN                 |    | ANN                    |     | FERN      |     | ENGL                 |     |
| SYSTEM PROGRAMMING                    |    | CAUDILLO              |    | RAUL                   |     | WETH      |     | COMP                 |     |
| SHORT STORIES AND TALL TALES          |    | AVRIL                 |    | LILIANA                |     | VALD      |     | ENGL                 |     |
| INTRODUCTION TO BIOLOGY               |    | WU                    |    | CHIEN                  |     | VALD      |     | BIOL                 |     |
| ADVANCED TOPICS IN PSYCHOANALYSIS     |    | OSTOICH               |    | DIANNE                 |     | FERN      |     | PSYCH                |     |
| EIGHTEENTH CENTURY EUROPE             |    | MUNGER                |    | ALICE                  |     | WETH      |     | HIST                 |     |
| CRISES OF THE MIDDLE AGES             |    | BENDER                |    | GREG                   |     | COR       |     | HIST                 |     |

### サンプル・データ・セット - SORT.SAMPADD

| Book Title                   |    | Author's<br>Last Name |    | Author's<br>First Name |     | Publisher |     | Course<br>Department |     |
|------------------------------|----|-----------------------|----|------------------------|-----|-----------|-----|----------------------|-----|
| 1                            | 75 | 76                    | 90 | 91                     | 105 | 106       | 109 | 110                  | 114 |
| GUNTHER'S GERMAN DICTIONARY  |    | WILLIS                |    | GUNTER                 |     | WETH      |     |                      |     |
| COMPLETE SPANISH DICTIONARY  |    | ROBERTS               |    | ANGEL                  |     | VALD      |     |                      |     |
| ANOTHER ITALIAN DICTIONARY   |    | UNDER                 |    | JOAN                   |     | COR       |     |                      |     |
| FRENCH TO ENGLISH DICTIONARY |    | JONES                 |    | JACK                   |     | FERN      |     |                      |     |
| GUIDE TO COLLEGE LIFE        |    | LAMB                  |    | CHARLENE               |     | WETH      |     |                      |     |
| THE ANIMAL KINGDOM           |    | YOUNG                 |    | KEVIN                  |     | COR       |     | BIOL                 |     |
| A SMALLER WORLD: MICROBES    |    | BEESELY               |    | GEORGE                 |     | FERN      |     | BIOL                 |     |
| DNA: BLUEPRINT FOR YOU       |    | IAVERS                |    | ILSE                   |     | FERN      |     | BIOL                 |     |
| CELLS AND HOW THEY WORK      |    | JETTS                 |    | PETER                  |     | VALD      |     | BIOL                 |     |
| KNOW YOUR CONSUMER           |    | ZANE                  |    | JENNIFER               |     | COR       |     | BUSIN                |     |
| ANTICIPATING THE MARKET      |    | ALLEN                 |    | CLYDE                  |     | WETH      |     | BUSIN                |     |
| ZEN BUSINESS                 |    | WILLIAMS              |    | KATIE                  |     | VALD      |     | BUSIN                |     |
| THE ART OF TAKEOVERS         |    | HUNT                  |    | ROBERT                 |     | FERN      |     | BUSIN                |     |
| THE TOY STORE TEST           |    | LITTLE                |    | MARIE                  |     | COR       |     | COMP                 |     |
| NOVEL IDEAS                  |    | PETERS                |    | SETH                   |     | VALD      |     | ENGL                 |     |
| POLITICS AND HISTORY         |    | TOMPSOM               |    | KEN                    |     | FERN      |     | HIST                 |     |
| CIVILIZATION SINCE ROME FELL |    | PIERCE                |    | NICOLE                 |     | WETH      |     | HIST                 |     |
| REBIRTH FROM ITALY           |    | FISH                  |    | JOHN                   |     | WETH      |     | HIST                 |     |
| FREUD'S THEORIES             |    | GOOLE                 |    | APRIL                  |     | VALD      |     | PSYCH                |     |
| MAP OF THE HUMAN BRAIN       |    | WINTER                |    | POLLY                  |     | COR       |     | PSYCH                |     |
| QUEUE THEORY                 |    | FOX                   |    | THAD                   |     | FERN      |     | BUSIN                |     |
| DESIGNING APPLICATIONS       |    | STEVENS               |    | NOAH                   |     | COR       |     | COMP                 |     |

サンプル・データ・セット - SORT.SAMPIN (続き)

| Course<br>Number | Course<br>Name     | Instructor's<br>Last Name | Instructor's<br>Initials | Number<br>In Stock | Number Sold<br>Year-to-Date | Price   |
|------------------|--------------------|---------------------------|--------------------------|--------------------|-----------------------------|---------|
| 115 119          | 120 144            | 145 159                   | 160 161                  | 162 165            | 166 169                     | 170 173 |
| 00032            | INTRO TO COMPUTERS | CHATTERJEE                | CL                       | 5                  | 29                          | 2600    |
|                  |                    |                           |                          | 14                 | 1                           | 9900    |
| 70251            | MARKETING          | MAXWELL                   | RF                       | 0                  | 32                          | 1925    |
| 00205            | VIDEO GAMES        | NEUMANN                   | LB                       | 10                 | 10                          | 2199    |
| 10856            | MODERN POETRY      | FRIEDMAN                  | KR                       | 2                  | 32                          | 595     |
| 00032            | INTRO TO COMPUTERS | CHATTERJEE                | CL                       | 20                 | 26                          | 1899    |
|                  |                    |                           |                          | 46                 | 38                          | 295     |
| 10347            | TECHNICAL EDITING  | MADRID                    | MM                       | 13                 | 32                          | 1450    |
| 00032            | INTRO TO COMPUTERS | CHATTERJEE                | AN                       | 6                  | 27                          | 360     |
| 70124            | ADVANCED MARKETING | LORCH                     | MH                       | 3                  | 35                          | 2350    |
| 50420            | WORLD HISTORY      | GOODGOLD                  | ST                       | 15                 | 9                           | 795     |
| 10856            | MODERN POETRY      | FRIEDMAN                  | KR                       | 1                  | 26                          | 450     |
| 30016            | PSYCHOLOGY I       | ZABOSKI                   | RL                       | 26                 | 15                          | 2200    |
| 10347            | TECHNICAL EDITING  | MADRID                    | MM                       | 7                  | 19                          | 625     |
| 00103            | DATA MANAGEMENT    | SMITH                     | DC                       | 4                  | 23                          | 3195    |
| 10054            | FICTION WRITING    | BUCK                      | GR                       | 10                 | 9                           | 1520    |
| 80521            | BIOLOGY I          | GREENBERG                 | HC                       | 6                  | 11                          | 2350    |
| 30975            | PSYCHOANALYSIS     | NAKATSU                   | FL                       | 1                  | 12                          | 2800    |
| 50632            | EUROPEAN HISTORY   | BISCARDI                  | HR                       | 23                 | 21                          | 1790    |
| 50521            | WORLD HISTORY      | WILLERTON                 | DW                       | 14                 | 17                          | 1200    |

サンプル・データ・セット - SORT.SAMPADD (続き)

| Course<br>Number | Course<br>Name     | Instructor's<br>Last Name | Instructor's<br>Initials | Number<br>In Stock | Number Sold<br>Year-to-Date | Price   |
|------------------|--------------------|---------------------------|--------------------------|--------------------|-----------------------------|---------|
| 115 119          | 120 144            | 145 159                   | 160 161                  | 162 165            | 166 169                     | 170 173 |
|                  |                    |                           |                          | 3                  | 16                          | 1088    |
|                  |                    |                           |                          | 8                  | 4                           | 650     |
|                  |                    |                           |                          | 26                 | 6                           | 925     |
|                  |                    |                           |                          | 7                  | 17                          | 1100    |
|                  |                    |                           |                          | 20                 | 1                           | 2000    |
| 80522            | BIOLOGY II         | HAROLD                    | LM                       | 2                  | 30                          | 3000    |
| 80522            | BIOLOGY II         | HAROLD                    | LM                       | 9                  | 20                          | 1955    |
| 80523            | INTRO TO GENETICS  | ABRAHAM                   | NG                       | 15                 | 21                          | 2195    |
| 80523            | INTRO TO GENETICS  | ABRAHAM                   | NG                       | 8                  | 46                          | 2495    |
| 70251            | MARKETING          | MAXWELL                   | RF                       | 16                 | 3                           | 4500    |
| 70124            | ADVANCED MARKETING | LORCH                     | MH                       | 20                 | 25                          | 2000    |
| 70255            | BUSINESS THEORY    | SCHOFFE                   | KN                       | 12                 | 12                          | 1200    |
| 70255            | BUSINESS THEORY    | SCHOFFE                   | KN                       | 17                 | 15                          | 615     |
| 00205            | VIDEO GAMES        | NEUMANN                   | LB                       | 15                 | 12                          | 2600    |
| 10054            | FICTION WRITING    | BUCK                      | GR                       | 11                 | 25                          | 2450    |
| 50521            | WORLD HISTORY      | WILLERTON                 | DW                       | 3                  | 17                          | 995     |
| 50420            | WORLD HISTORY      | GOODGOLD                  | ST                       | 6                  | 15                          | 1350    |
| 50632            | EUROPEAN HISTORY   | BISCARDI                  | HR                       | 10                 | 20                          | 2560    |
| 30975            | PSYCHOANALYSIS     | NAKATSU                   | FL                       | 12                 | 15                          | 1250    |
| 30016            | PSYCHOLOGY I       | ZABOSKI                   | RL                       | 6                  | 9                           | 895     |
| 70255            | BUSINESS THEORY    | SCHOFFE                   | KN                       | 2                  | 20                          | 1500    |
| 00103            | DATA MANAGEMENT    | SMITH                     | DC                       | 7                  | 15                          | 1435    |

## サンプル・ブックストア・データ・セット

### サンプル・データ・セット - SORT.BRANCH

| City          |    | State | Employees |  | Revenue | Profit |  |
|---------------|----|-------|-----------|--|---------|--------|--|
| 1             | 15 | 16 17 | 18 21     |  | 22 27   | 28 33  |  |
| Los Angeles   |    | CA    | 32        |  | 22530   | -4278  |  |
| San Francisco |    | CA    | 35        |  | 42820   | 6832   |  |
| Fort Collins  |    | CO    | 22        |  | 12300   | -2863  |  |
| Sacramento    |    | CA    | 29        |  | 42726   | 8276   |  |
| Sunnyvale     |    | CA    | 18        |  | 16152   | -978   |  |
| Denver        |    | CO    | 33        |  | 31876   | 6288   |  |
| Boulder       |    | CO    | 32        |  | 33866   | 7351   |  |
| Morgan Hill   |    | CA    | 15        |  | 18200   | 3271   |  |
| Vail          |    | CO    | 19        |  | 23202   | 5027   |  |
| San Jose      |    | CA    | 21        |  | 27225   | 8264   |  |
| San Diego     |    | CA    | 22        |  | 32940   | 8275   |  |
| Aspen         |    | CO    | 20        |  | 25800   | 5200   |  |

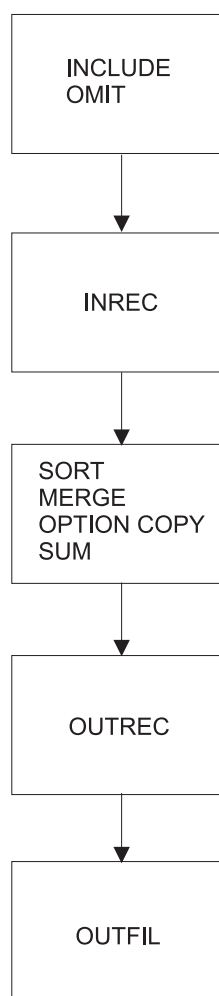


---

## 付録 C. 制御ステートメントの処理順序

以下のフローチャートは、制御ステートメントが処理される順序を示しています。  
(SUM は、SORT または MERGE と同時に処理されます。SUM は COPY と一緒には使用されません。)

ユーザーはどんな順序でステートメントを書いても構いませんが、DFSORT は常に下記に示された順序でステートメントを処理します。



---

図 7. 制御ステートメントの処理順序



---

## 付録 D. アクセシビリティ

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。z/OS のアクセシビリティの主要機能により、ユーザーは以下のことができるようになります。

- 画面読み上げ機能および画面拡大機能などの支援機能の使用
- キーボードのみを使用して、特定の機能または画面を使用したのと同等の機能を操作
- 色、コントラスト、フォント・サイズなど表示属性のカスタマイズ

---

### 支援機能の使用

画面読み上げ機能などの支援機能は、z/OS のユーザー・インターフェースを使用して機能します。この支援機能を使用して z/OS インターフェースにアクセスする場合、その特定情報については支援機能の資料を参照してください。

---

### ユーザー・インターフェースのキーボード・ナビゲーション

ユーザーは、TSO/E または ISPF を使用して z/OS ユーザー・インターフェースにアクセスできます。TSO/E および ISPF インターフェースへのアクセス方法については、*z/OS TSO/E 入門*、*z/OS TSO/E ユーザーズ・ガイド*、および *z/OS ISPF ユーザーズ・ガイド 第 1 巻* を参照してください。上記の資料には、キーボード・ショートカットまたはファンクション・キー (PF キー) の使用方法を含む TSO/E および ISPF の使用方法が記載されています。それぞれの資料では、PF キーのデフォルトの設定値とそれらの機能の変更方法についても説明しています。

---

### z/OS 情報

z/OS の情報は、以下の URL のインターネット・ライブラリーにある BookServer/Library Server の各バージョンの z/OS 資料に、スクリーン・リーダーを使用してアクセスすることができます。

<http://www.ibm.com/systems/z/os/zos/bkserv/>



---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-8711  
東京都港区六本木 3-2-12  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation  
Mail Station P300  
2455 South Road

Poughkeepsie, NY 12601-5400  
USA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

---

## プログラミング・インターフェース情報

本書の情報は、DFSORT のプログラミング・インターフェースとして使用されることを意図して記述されたものではありません。

本書には、お客様が DFSORT のサービス入手するプログラムを書くのに使用できる、意図されたプログラミング・インターフェースも記載されています。この情報は、章またはセクションの始まるの文によって、あるいは下記の表示によって、その出現箇所を識別できます。

——— プログラミング・インターフェース情報 ———

——— プログラミング・インターフェース情報 の終り ———

---

## 商標

以下は、International Business Machines Corporation の商標です。

|              |         |
|--------------|---------|
| DFSORT       | IBMLink |
| Hipersorting | RACF    |
| Hiperspace   | z/OS    |
| IBM          | z/VM    |

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アクセシビリティ 195  
印刷, 報告書の  
    DISPLAY オペレーター  
        (ICETOOL) 147  
    OUTFIL ステートメント 101  
インストール・デフォルトの指定変更 123  
オーバーフロー  
    説明 42  
    防止 75  
大文字から小文字へ 53

## [カ行]

可変フィールド 83  
紙送り制御, ANSI 102, 104, 107  
キーボード 195  
許容される比較 33  
繰り返し 97  
継続, ステートメントの  
    DFSORT 19  
    ICETOOL 138  
結合 162  
検索  
    サブストリング 37  
降順  
    ソート 16  
    例 4  
向上, パフォーマンスの  
    回避すべきオプション 130  
    高速ディスク 128  
    主記憶域の割り振り 127  
    ハイパースペース 128  
FASTSORT コンパイル時オプション 130  
INCLUDE ステートメント 129  
INREC ステートメント 128  
JCL (ジョブ制御言語) 130  
OMIT ステートメント 129  
SKIPREC オプション 129  
STOPAFT オプション 129  
SUM ステートメント 129  
高速ディスク 128

固定小数点データ 6  
固定フィールド 45  
コピー, レコードの 26  
小文字から大文字へ 53

## [サ行]

再配列, フィールドの 48  
再フォーマット, レコードの 45, 83  
作業用ストレージ・データ・セット  
    効率的使用のための装置 128  
    必要な数 19  
削除, フィールドの 46  
削除, レコードの 39  
作成, 複数データ・セット・コピーの 90  
作成, DFSORT ジョブの 7  
サブストリング検索 37  
算術演算 59  
サンプル・データ・セット 8, 91, 134, 187, 189  
シーケンス番号 76  
実行, DFSORT ジョブの 7  
指定, COPY の  
    COPY オペレーター (ICETOOL) 144  
    MERGE ステートメント 26  
    OPTION ステートメント 26  
    SORT ステートメント 26  
指定変更, インストール・デフォルトの 123  
主記憶域の割り振り 127  
出現  
    OCCUR オペレーター  
        (ICETOOL) 157  
    SELECT オペレーター  
        (ICETOOL) 159  
    SPLICE オペレーター  
        (ICETOOL) 162  
出力レコードの再フォーマット 45  
順序, 制御ステートメントの 31  
順序付け  
    降順 4  
    昇順 4  
    制御ステートメント 31  
    DFSORT デフォルト 10  
ショートカット・キー 195  
昇順  
    ソート 16  
    例 4  
除外, レコードの 32  
処理順序  
    フローチャート 193

処理順序 (続き)  
    INREC と OUTREC の違い 45  
    INREC に関する特別な考慮事項 74, 75  
身体障害 195  
シンボル 177  
数値テスト 36  
制御ステートメント  
    順序付け 31  
    処理順序フローチャート 193  
DFSORT 7  
制御フィールド  
    一般情報 8  
    オーバーラップ 17  
    結合 19  
    合計 39  
    再配列 48  
    重複 39  
    重複レコードの削除 41  
    等価照合 10, 39  
    複数 17  
セクション 108, 154  
ゼロ 48  
選択, レコードの 159  
ソート, レコードの  
    DFSORT 15  
    ICETOOL 139  
ゾーン 10 進数データ 5

## [タ行]

短フィールド 25, 38, 43  
抽出 91  
重複フィールド  
    OCCUR オペレーター  
        (ICETOOL) 157  
    SELECT オペレーター  
        (ICETOOL) 159  
    SPLICE オペレーター  
        (ICETOOL) 162  
    SUM ステートメント 121  
    UNIQUE オペレーター  
        (ICETOOL) 134  
突き合わせ 165  
データの位置調整 61  
データの左/右詰め (ブランクなどを削除) 64  
データ・スペース・ソート 128  
データ・セット  
    コピー  
        定義 6

データ・セット (続き)  
コピー (続き)  
    DFSORT による 26  
    ICETOOL による 144  
ソート  
    降順 16  
    昇順 16  
    定義 4  
    複数の制御フィールドによる 17  
    DFSORT による 15  
    ICETOOL による 139  
    複数コピーの作成 90, 91  
    マージ  
        定義 5  
        DFSORT による 21  
データ・フォーマット 5  
定数  
    再フォーマット 51  
    比較 35  
デフォルト  
    指定変更 123  
        OPTION ステートメントの使用 124  
        PARM パラメーターの使用 124  
    等しいレコードの順序 10  
統計、数値フィールドの印刷 136, 149  
統計の印刷、数値フィールドの 105

## [ハ行]

廃棄レコードの保管 93  
ハイパー・ソート 128  
バック 10 進数データ 5  
比較  
    演算子 31  
    許容される 33  
    数値 36  
    フィールドと定数 29, 33  
    フィールドとフィールド 29, 33  
非重複フィールド  
    OCCUR オペレーター (ICETOOL) 157  
    SELECT オペレーター (ICETOOL) 159  
    SPLICE オペレーター (ICETOOL) 162  
    SUM ステートメント 121  
    UNIQUE オペレーター (ICETOOL) 134  
等しい制御フィールド 10  
フィールド  
    再フォーマット 48  
    数値の統計の印刷 136  
    不要なフィールドの削除 46  
    編集 48, 150  
    文字および数値の印刷 147

フィールドと定数の比較 29, 33  
フィールドとフィールドの比較 29, 33  
フォーマット 6  
フォーマット、定数を書く際の 35  
フォーマット、フィールドの 45, 150  
複数データ・セット 90  
複数の制御フィールド 17  
浮動符号データ 6  
ブランク 49  
分割 98  
変換、数値フィールドの 54  
編集、フィールドの 45, 150  
編集マスク  
    ICETOOL 150  
    OUTFIL 57  
報告書  
    DISPLAY オペレーター (ICETOOL) 147  
    OUTFIL ステートメント 100  
保管、廃棄レコードの 93

## [マ行]

マージ、レコードの 21  
マッピング  
    シンボル 177  
メッセージ検索ツール、LookAt xiii  
メモリー・オブジェクト・ソート 128  
文字ストリングのフォーマット 51  
文字データ 5

## [ヤ行]

要約フィールド 39

## [ラ行]

ルックアップおよび変更 60  
レコード  
    合計 39  
    コピー  
        定義 6  
        COPY ステートメントの作成 26  
    出現回数によって選択 159  
    ソート  
        降順 16  
        昇順 16  
        定義 4  
        複数の制御フィールドによる 17  
        SORT ステートメントの作成 15  
マージ  
    定義 5  
    プログラム制御ステートメントの使用 21

レコード (続き)  
    INREC および OUTREC で再配列するときの考慮事項 74, 75  
    OUTREC による再配列 29  
    OUTREC による再配列および再フォーマット 48  
    OUTREC による再フォーマット 29  
レコード長の変更 47  
レコードの解析 83

## [数字]

16 進数ストリングのフォーマット 51  
16 進数表示 58  
2 進数データ 5  
2 進ゼロ 48

## A

ASCII データ・フォーマット・コード 5

## B

BI フォーマット 5

## C

CH フォーマット 5  
COBOL  
    サンプル・プログラム 116, 119  
    DFSORT ステートメントの受け渡し 115  
    DFSORT の呼び出し 116  
    FASTSORT コンパイル時オプション 130  
    MERGE ステートメント 116  
    SORT ステートメント 116  
COPY オプション 26  
COPY オペレーター (ICETOOL) 144  
COUNT オペレーター (ICETOOL) 133

## D

DEFAULTS オペレーター (ICETOOL) 133  
DFSORT FTP サイト 3  
DFSORT Web サイト 3  
DFSORT ステートメントの受け渡し  
    COBOL プログラムから 115  
    PL/I プログラムから 115  
DFSORT の呼び出し  
    COBOL プログラム 116  
    PL/I プログラム 121  
DFSPARM 20



DISPLAY オペレーター (ICETOOL) 147

## E

EBCDIC データ・フォーマット・コード  
5

## F

FASTSRT  
コンパイル時オプション 130  
COBOL 121  
FB から VB へ 111  
FI フォーマット 6  
FS フォーマット 6  
FTP サイト 3

## I

ICETOOL  
オペレーターの要約 133  
完全なサンプル・ジョブ 169  
ジョブ  
エレメント 135  
サンプル 169  
ジョブの作成 135  
シンボル 180  
ステートメント  
オペレーターの継続 138  
コメント 136  
ブランク 136  
定義 133  
要件  
入力データ・セット 134  
JCL 135  
COPY オペレーター 144  
COUNT オペレーター 133  
DEFAULTS オペレーター 133  
DISPLAY オペレーター 147  
JCL ステートメント 135  
MODE オペレーター 133  
OCCUR オペレーター 157  
RANGE オペレーター 145  
SELECT オペレーター 159  
SPLICE オペレーター 162  
STATS オペレーター 136  
TOOLMSG 出力例 169  
UNIQUE オペレーター 134  
VERIFY オペレーター 134  
INCLUDE ステートメント 29  
INREC ステートメント 45  
ISCI データ・フォーマット・コード 5

## J

JCL (ジョブ制御言語) 130  
一般情報 20  
コピーの実行 26  
出力データ・セットの選択 20  
ステートメント 20  
ソートの実行 19  
入力データ・セットの選択 20  
プログラムからの DFSORT の呼び出し 116, 119, 121  
マージの実行 23  
ICETOOL 135  
JOB ステートメント 19

## L

LookAt メッセージ検索ツール xiii

## M

MERGE ステートメント 22  
MERGE ステートメント、COBOL 116  
MODE オペレーター (ICETOOL) 133

## O

OCCUR オペレーター (ICETOOL) 157  
OMIT ステートメント 29  
OPTION COPY 26  
OPTION ステートメント 26, 124  
OUTFIL  
編集マスク・パターン 57  
OUTFIL ステートメント 89  
OUTREC ステートメント 45

## P

PARM パラメーター 124  
PD フォーマット 5  
PL/I  
サンプル・プログラム 121  
DFSORT ステートメントの受け渡し  
115  
DFSORT の呼び出し 121

## R

RANGE オペレーター (ICETOOL) 134,  
145  
RECORD ステートメント 121

## S

SELECT オペレーター (ICETOOL) 159  
SFF フォーマット 5  
SKIPREC オプション 129  
SORT オペレーター (ICETOOL) 139  
SORT ステートメント 15  
SORT ステートメント、COBOL 116  
SORT-CONTROL 特殊レジスター  
(COBOL) 116  
SORT-RETURN 特殊レジスター  
(COBOL) 116  
SPLICE オペレーター (ICETOOL) 162  
STATS オペレーター (ICETOOL) 136  
STOPAFT オプション 129  
SUM ステートメント 39

## U

UFF フォーマット 5  
UNIQUE オペレーター (ICETOOL) 134

## V

VB から FB へ 113  
VB データ・セット 24, 37, 42, 79, 107,  
111, 113, 139  
VERIFY オペレーター (ICETOOL) 134

## W

Web サイト 3

## Z

ZD フォーマット 5







プログラム番号: 5694-A01

Printed in Japan

SD88-6335-02



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