



IBM Systems - iSeries

プログラミング

DDS 物理ファイルと論理ファイル

バージョン 5 リリース 4







**IBM Systems - iSeries**

**プログラミング**

**DDS 物理ファイルと論理ファイル**

**バージョン 5 リリース 4**

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、105 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、IBM i5/OS (プロダクト番号 5722-SS1) のバージョン 5、リリース 4、モディフィケーション 0 に適用されます。また、改訂版で断りがない限り、それ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も適用されます。このバージョンは、すべての RISC モデルで稼働するとは限りません。また、CISC モデルでは稼働しません。

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

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

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

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

(URL は、変更になる場合があります)

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： IBM Systems - iSeries  
Programming DDS for physical and logical files  
Version 5 Release 4

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.2

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

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

© Copyright International Business Machines Corporation 2001, 2006. All rights reserved.

© Copyright IBM Japan 2006

# 目次

物理ファイルおよび論理ファイルの DDS	1
V5R4 の新機能	1
印刷可能 PDF	1
DDS を使用する物理ファイルおよび論理ファイルの定義	2
DDS を使用する物理ファイルの定義	2
DDS を使用する論理ファイルの定義	2
物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)	5
物理ファイルおよび論理ファイルのキーワード項目 (45 から 80 桁目)	36
物理ファイルおよび論理ファイルの ABSVAL (絶対値) キーワード	37
物理ファイルおよび論理ファイルの ALIAS (別名) キーワード	38
ALL (すべて) キーワード - 論理ファイルのみ	38
物理ファイルおよび論理ファイルの ALTSEQ (代替照合順序) キーワード	39
ALWNULL (ヌル値の許容) キーワード - 物理ファイルのみ	40
物理ファイルおよび論理ファイルの CCSID (コード化文字セット識別コード) キーワード	41
物理ファイルおよび論理ファイルの CHECK (検査) キーワード	43
物理ファイルおよび論理ファイルの CHKMSGID (検査メッセージ ID) キーワード	44
物理ファイルおよび論理ファイルの CMP (比較) キーワード	44
物理ファイルおよび論理ファイルの COLHDG (列見出し) キーワード	45
物理ファイルおよび論理ファイルの COMP (比較) キーワード	45
CONCAT (連結) キーワード - 論理ファイルのみ	49
物理ファイルおよび論理ファイルの DATFMT (日付形式) キーワード	51
物理ファイルおよび論理ファイルの DATSEP (日付区切り記号) キーワード	54
物理ファイルおよび論理ファイルの DESCEND (降順) キーワード	55
DFT (デフォルト) キーワード - 物理ファイルのみ	55
物理ファイルおよび論理ファイルの DIGIT (ディジット) キーワード	57
DYNSLT (動的選択) キーワード - 論理ファイルのみ	58
物理ファイルおよび論理ファイルの EDTCDE (編集コード) および EDTWRD (編集語) キーワード	61
物理ファイルおよび論理ファイルの FCFO (変更順検索) キーワード	62
物理ファイルおよび論理ファイルの FIFO (先入れ先出し) キーワード	62

物理ファイルおよび論理ファイルの FLTPCN (浮動小数点の精度) キーワード	63
物理ファイルおよび論理ファイルの FORMAT (様式) キーワード	63
JDFTVAL (結合におけるデフォルト値) キーワード - 結合論理ファイルのみ	64
JDUPSEQ (結合における重複レコードの順序付け) キーワード - 結合論理ファイルのみ	66
JFILE (結合ファイル) キーワード - 結合論理ファイルのみ	67
JFLD (結合フィールド) キーワード - 結合論理ファイルのみ	69
JOIN (結合) キーワード - 結合論理ファイルのみ	70
JREF (結合における参照) キーワード - 結合論理ファイルのみ	73
物理ファイルおよび論理ファイルの LIFO (後入れ先出し) キーワード	74
物理ファイルおよび論理ファイルの NOALTSEQ (非代替照合順序) キーワード	75
PFILE (物理ファイル) キーワード - 論理ファイルのみ	75
物理ファイルおよび論理ファイルの RANGE (範囲) キーワード	77
REF (参照) キーワード - 物理ファイルのみ	79
REFACCPH (アクセス・パス定義参照) キーワード - 論理ファイルのみ	80
REFFLD (参照フィールド) キーワード - 物理ファイルのみ	81
物理ファイルおよび論理ファイルの REFSHIFT (シフト参照) キーワード	82
RENAME (名前変更) キーワード - 論理ファイルのみ	83
物理ファイルおよび論理ファイルの SIGNED (符号付き) キーワード	84
SST (サブストリング) キーワード - 論理ファイルのみ	85
物理ファイルおよび論理ファイルの TEXT (テキスト) キーワード	87
物理ファイルおよび論理ファイルの TIMFMT (時刻形式) キーワード	88
物理ファイルおよび論理ファイルの TIMSEP (時刻区切り記号) キーワード	89
TRNTBL (変換テーブル) キーワード - 論理ファイルのみ	89
物理ファイルおよび論理ファイルの UNIQUE (固有) キーワード	91
物理ファイルおよび論理ファイルの UNSIGNED (符号なし) キーワード	92
物理ファイルおよび論理ファイルの VALUES (値) キーワード	93

物理ファイルおよび論理ファイルの VARLEN (可変長フィールド) キーワード . . . . .	95
物理ファイルおよび論理ファイルの ZONE (ゾーン) キーワード . . . . .	96
データベース・ファイルの Unicode に関する考慮事項 . . . . .	96
長さ (30 から 34 桁目) . . . . .	97
データ・タイプ (35 桁目) . . . . .	98
小数点以下の桁数 (36 から 37 桁目). . . . .	98
キーワードに関する考慮事項 (45 から 80 桁目) . . . . .	98
データベース・ファイルの DBCS に関する考慮事項 . . . . .	100

DBCS を使用するデータベース・ファイルの定位置項目に関する考慮事項 . . . . .	100
DBCS を使用するデータベース・ファイルのキーワードに関する考慮事項 . . . . .	101
DBCS データが含まれるデータベース・ファイルを記述する場合の追加考慮事項. . . . .	103
<b>付録. 特記事項 . . . . .</b>	<b>105</b>
プログラミング・インターフェース情報 . . . . .	106
商標 . . . . .	106
使用条件 . . . . .	107

---

## 物理ファイルおよび論理ファイルの DDS

データ記述仕様 (DDS) を使用して、物理データベース・ファイルおよび論理データベース・ファイルを定義することができます。このトピックでは、これらのファイルを定義する定位置項目およびキーワード項目をコーディングするさいに必要な情報について説明します。

注: 本書のコード例を使用すると、お客様は、104 ページの『コード・ライセンスに関する特記事項』に記載されている条件に同意されたことになります。

---



### V5R4 の新機能

この項では、V5R4 の物理ファイルおよび論理ファイルの DDS の変更点の重点を述べます。

- | キーの合計の長さが、32 768 に変更されました。ただし、FCFO キーワードを指定する場合には、キーの
- | 合計の長さは 32 763 を超えてはなりません。

#### 新機能または変更点を見つける方法

テクニカルな変更内容を見つけやすくするために、本書では、以下のイメージを使用しています。

-  イメージは、新規情報または変更情報が始まる位置を示します。
-  イメージは、新規情報または変更情報が終わる位置を示します。

このリリースでの、その他の新規情報または変更情報を見つけるには、ユーザーへのメモを見つけてください。

---

### 印刷可能 PDF

本書の PDF を表示およびプリントするには、以下の説明を使用してください。


本書の PDF バージョンを表示あるいはダウンロードするには、物理ファイルおよび論理ファイルの DDS を選択します。

#### PDF ファイルの保存

表示用または印刷用の PDF ファイルをワークステーションに保存するには、次のようにします。

1. ブラウザーで PDF ファイルを右マウス・ボタンでクリックします (上記のリンクを右マウス・ボタンでクリックする)。
2. PDF をローカルで保管するオプションをクリックします。
3. PDF ファイルを保管するディレクトリーを指定します。
4. 「保存」をクリックします。

#### Adobe Reader のダウンロード

これらの PDF を表示または印刷するには、システムに Adobe Reader がインストールされていることが必要です。Adobe Web サイト ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  から、無償コピーをダウンロードできます。

---

## DDS を使用する物理ファイルおよび論理ファイルの定義

このトピックでは、DDS を使用して物理ファイルおよび論理ファイルをコーディングする方法、および、これらのファイルが使用する定位置項目について説明します。

### DDS を使用する物理ファイルの定義

物理ファイルを DDS に定義するには、ファイル・レベル項目、レコード・レベル項目、フィールド・レベル項目、キー・フィールド・レベル項目を使用します。

物理ファイルには、レコード様式を 1 つだけ含めることができます。物理ファイルを定義するには、以下の手順を実行します。

1. レコード様式は、次の 2 つのうちのいずれかの方法で指定します。

- 新しいレコード様式を定義する。

その新しいレコード様式については、必要に応じてフィールド仕様およびキー・フィールド仕様を指定します。

- 既存のレコード様式を共有する。

そのためには、**FORMAT** キーワードを用いて、別の物理ファイルですでに定義されているレコード様式を使用することをオペレーティング・システムに指示します。**FORMAT** キーワードを使用する場合には、既存のレコード様式にキー・フィールド・レベルの仕様が指定されていたとしても、その仕様が再度指定しなければなりません (ただしキー順アクセス・パスが必要な場合に限り)。)

2. 物理ファイルを定義するための項目は、次の順序で指定します。

- a. ファイル・レベル項目
- b. レコード・レベル項目
- c. フィールド・レベル項目
- d. キー・フィールド・レベル項目

注: ファイル名は、DDS ではなく物理ファイル作成 (CRTPF) コマンドによって指定します。

指定できるフィールドの数は、1 つのレコード様式について最高 8000 個です。レコード様式内のフィールドのいずれかが、日付、時刻、時刻スタンプ、可変長であるか、またはヌル値を許容するときに、フィールドの実際の最大数は 8000 より少なくなる場合もあります。フィールドの最大数はレコード様式内にあるフィールドの数およびフィールドの組み合わせによって異なります。レコード様式内の最大バイト数は可変長フィールドが含まれていなければ 32 766 です。可変長フィールドが含まれれば 32 740 です。

#### 関連情報

DDS キーワードおよびパラメーター値の規則

### DDS を使用する論理ファイルの定義

論理ファイルは、アプリケーション・プログラムによる読み取り操作のさいに、データ・レコードがどのように選択され、かつ定義されるかを決めます。論理ファイルは、単一様式論理ファイル、複数様式論理ファイル、または結合論理ファイル のいずれかにすることができます。

単一様式論理ファイルは、1 つのレコード様式を含み、**PFILE** キーワードで指定された 1 つのファイルを持っています。複数様式論理ファイルは、複数のレコード様式を含んでいるか、**PFILE** キーワードで指定された複数のファイルを持っています。結合論理ファイルは、1 つのレコード様式を含み、**JFILE** キーワードで指定された最高 256 個のファイルを持っています。



## 関連概念

4 ページの『DDS の論理ファイルのレコード様式の指定』

1 つの論理ファイルにレコード様式を 2 つ以上指定する場合には、レコード様式ごとに PFILE キーワードを指定しなければなりません。

## DDS の単一様式論理ファイルおよび複数様式論理ファイル

単一様式論理ファイルおよび複数様式論理ファイルの場合には、レコード・レベルで PFILE キーワードを指定しなければなりません。複数様式論理ファイルでは、あるレコード様式は、そのレコード様式について PFILE キーワードで指定したすべての物理ファイルに共通するフィールドだけしか使用することができません。

単一様式論理ファイルまたは複数様式論理ファイルを定義するためには、各項目を次の順序で指定します。

1. 任意: ファイル・レベル項目
2. レコード・レベル項目
3. 任意: フィールド・レベル項目
4. 任意: キー・フィールド・レベル項目
5. 任意: 選択および除外フィールド・レベル項目

ファイル内の各レコード様式について、2 から 5 の指定を繰り返してください。

次の例では、複数様式論理ファイルのコーディング例を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* LOGICAL FILE EXAMPLE
00020A* INVENTORY FORMAT
00030A      R  INVFMT                PFILE(INVENTORY)
00040A      K  ITEM
00050A*
00060A* ORDER FORMAT
00070A      R  ORDFMT                PFILE(ORDER)
00080A      TEXT('ORDER ANALYSIS')
00090A      ITEM
00100A      ORDER                    10
00110A      SUPPLY                    +2
00120A      SHPDAT                    CONCAT(SHPMO SHPDA SHPYR)
00130A      QTY                        5P      RENAME(QTYDUE)
00140A      K  ITEM
00150A      K  SHPYR
00160A      K  SHPMO
00170A      K  SHPDA
00180A      O  QTYDUE                CMP(LT 1)
00190A*
00200A* ACCOUNTING FORMAT
00210A      R  ACTFMT                PFILE(ACCOUNTS)
00220A      FORMAT(ACCOUNTL)
00230A      K  ITEM
A
```

## 関連概念

14 ページの『複数のレコード様式がある DDS 論理ファイル』

1 つの論理ファイルについて複数のレコード様式を指定する場合には、その論理ファイル内の各レコード様式に対して、少なくとも 1 つはキー・フィールドを指定しなければなりません。

## 関連資料

5 ページの『物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)』

DDS 形式の最初の 44 桁は定位置項目と呼ばれます。

7 ページの『物理ファイルおよび論理ファイルの名前 (19 から 28 桁目)』

この欄には、レコード名およびフィールド名を指定します。

26 ページの『物理ファイルおよび論理ファイルの長さ (30 から 34 桁目)』

この欄は、物理ファイル・フィールドまたは論理ファイル・フィールドの長さを指定するのに使用します。

29 ページの『物理ファイルおよび論理ファイルのデータ・タイプ (35 桁目)』

物理ファイルでは、この欄にはデータベース内でのフィールドのデータ・タイプを指定します。論理ファイルの場合にこの欄にデータ・タイプを指定するのは、この論理ファイルの基礎となっている物理ファイル内の対応するフィールドのデータ・タイプを指定変更または修正したい場合に限りです。

## DDS での論理ファイルの結合

結合論理ファイルでは、複数の物理ファイルからのいくつかの異なるフィールドを組み合わせて、1 つのレコードが形成されます。結合論理ファイルの場合、レコード・レベルで **JFILE** キーワードを指定しなければなりません。

結合論理ファイルを定義するためには、各項目を次の順序で指定します。

注: 結合論理ファイルでは、レコード様式は 1 つしか使用できないため、上記の各項目は一度だけ指定します。

1. **任意:** ファイル・レベル項目を指定する
2. レコード・レベル項目を指定する
3. 結合レベル項目を指定する
4. フィールド・レベル項目を指定する
5. **任意:** キー・フィールド・レベル項目を指定する
6. **任意:** 選択/除外フィールド・レベル項目を指定する

結合論理ファイルのコードの例を次に示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* Joins fields from two physical files into one record format
00020A      R RECORD1          JFILE(PF1 PF2)
00030A      J                  JOIN(PF1 PF2)
00040A                      JFLD(NAME NAME)
00050A      NAME              JREF(1)
00060A      ADDR
00070A      PHONE
      A
```

### 関連資料

5 ページの『物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)』

DDS 形式の最初の 44 桁は定位置項目と呼ばれます。

7 ページの『物理ファイルおよび論理ファイルの名前 (19 から 28 桁目)』

この欄には、レコード名およびフィールド名を指定します。

### 関連情報

DDS キーワードおよびパラメーター値の規則

## DDS の論理ファイルのレコード様式の指定

1 つの論理ファイルにレコード様式を 2 つ以上指定する場合には、レコード様式ごとに **PFILE** キーワードを指定しなければなりません。

レコード様式内のフィールドを指定するには、次の 3 通りの方法があります。

- レコード様式名および PFILE キーワードを指定する。
- レコード様式名、PFILE または JFILE のいずれかのキーワード、および少なくとも 1 つのフィールドを指定する。
- レコード様式名、PFILE キーワード、および FORMAT キーワードを指定する。

3 ページの『DDS の単一様式論理ファイルおよび複数様式論理ファイル』に、3 つのフィールド指定方法が示されています。

レコード様式内のフィールドを指定する上記の 3 つの方法のいずれについても、次のアクセス・パス 指定を使用することができます。

- キー・フィールドを指定しない (到着順アクセス・パス)。

DYNSLT キーワードを指定しない場合には、選択/除外フィールドを指定することはできません。論理ファイルについては、PFILE キーワードで、物理ファイルを 1 つとそのレコード様式を 1 つだけ指定することができます。

- 1 つまたは複数のキー・フィールドを指定する (キー順アクセス・パス)。

論理ファイル内に複数のレコード様式を指定する場合には、各レコード様式について少なくとも 1 つのキー・フィールドを指定しなければなりません。ファイル中のどのレコード様式に対しても、選択/除外フィールドを指定することができます。

- REFACCPATH キーワードを指定する (キー順アクセス・パス)。

他の物理ファイルまたは論理ファイルのアクセス・パス情報が、定義中のファイルの中に複写されません。

指定できるフィールドの数は、1 つのレコード様式について最高 8000 個です。レコード様式内のフィールドのいずれかが、日付、時刻、時刻スタンプ、可変長であるか、またはヌル値を許容するときに、フィールドの実際の最大数は 8000 より少なくなる場合もあります。フィールドの最大数はレコード様式内にあるフィールドの数およびフィールドの組み合わせによって異なります。レコード様式内の最大バイト数は可変長フィールドが含まれなければ 32 766 です。可変長フィールドが含まれれば 32 740 です。

#### 関連概念

2 ページの『DDS を使用する論理ファイルの定義』

論理ファイルは、アプリケーション・プログラムによる読み取り操作のさいに、データ・レコードがどのように選択され、かつ定義されるかを決めます。論理ファイルは、単一様式論理ファイル、複数様式論理ファイル、または結合論理ファイル のいずれかにすることができます。

#### 関連資料

26 ページの『物理ファイルおよび論理ファイルの長さ (30 から 34 桁目)』

この欄は、物理ファイル・フィールドまたは論理ファイル・フィールドの長さを指定するのに使用します。

## 物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)

DDS 形式の最初の 44 桁は定位置項目と呼ばれます。

次の例は、物理ファイルのいくつかの定位置項目を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* PHYSICAL FILE CODING EXAMPLE
00020A                                REF(INVENCTL/INVENTORY)
00030A                                UNIQUE
00040A      R ORDFMT                    TEXT('Format for Purchase Orders')
00050A          ORDNR          7 0      COLHDG('Order' 'Number')
00060A          ITMNR          R  10
00070A          SUPNR          R  +2      REFFLD(SUPID SUPLIB/SUPMST)
00080A          QTYORD          5B
00090A      K ORDNBR
00100A      K ITMNR                    ABSVAL
      A

```

図 1. 物理ファイルのコーディングの例

### 関連タスク

3 ページの『DDS の単一様式論理ファイルおよび複数様式論理ファイル』

単一様式論理ファイルおよび複数様式論理ファイルの場合には、レコード・レベルで PFILE キーワードを指定しなければなりません。複数様式論理ファイルでは、あるレコード様式は、そのレコード様式について PFILE キーワードで指定したすべての物理ファイルに共通するフィールドだけしか使用することができません。

4 ページの『DDS での論理ファイルの結合』

結合論理ファイルでは、複数の物理ファイルからのいくつかの異なるフィールドを組み合わせ、1 つのレコードが形成されます。結合論理ファイルの場合、レコード・レベルで JFILE キーワードを指定しなければなりません。

### 関連資料

7 ページの『物理ファイルおよび論理ファイルの名前 (19 から 28 桁目)』

この欄には、レコード名およびフィールド名を指定します。

29 ページの『物理ファイルおよび論理ファイルのデータ・タイプ (35 桁目)』

物理ファイルでは、この欄にはデータベース内でのフィールドのデータ・タイプを指定します。論理ファイルの場合にこの欄にデータ・タイプを指定するのは、この論理ファイルの基礎となっている物理ファイル内の対応するフィールドのデータ・タイプを指定変更または修正したい場合に限りです。

32 ページの『物理ファイルおよび論理ファイルの小数点以下桁数 (36 および 37 桁目)』

この欄には、パック 10 進数フィールド、ゾーン 10 進数フィールド、2 進数フィールド、または浮動小数点数フィールドにおける小数点の位置を指定します。

### 物理ファイルおよび論理ファイルの順序番号 (1 から 5 桁目)

この欄には、仕様書の各行の順序番号を指定します。順序番号はオプションで、文書化の目的にだけ使用されます。

### 物理ファイルおよび論理ファイルの仕様書タイプ (6 桁目)

これが DDS 仕様書であることを示すために、この欄に A を入力します。仕様書タイプはオプションで、文書化の目的にだけ使用されます。

### 物理ファイルおよび論理ファイルの注記 (7 桁目)

7 桁目にアスタリスク (\*) を入力すると、その行は注記として扱われます。注記テキストの記入には 8 から 80 桁目を使用します。

空白行 (7 から 80 桁目に文字がまったく指定されていない行) も、注記として扱われます。注記は DDS のどこにでも入れることができ、ソース・ファイルにだけ保管されます。注記は、コンパイル用コンピュータ印刷出力には印刷されますが、拡張コンパイル用コンピュータ印刷出力には印刷されません。

### 物理ファイルおよび論理ファイルの条件付け (8 から 16 桁目)

この欄は物理ファイルまたは論理ファイルでは使用しません。注記行のテキストを記入するために使用する場合を除き、この欄は空白のままにしてください。

### 物理ファイルおよび論理ファイルの名前または仕様のタイプ (17 桁目)

この欄には名前のタイプ、または論理ファイルの場合は仕様のタイプを指定するための値を入力します。名前タイプを指定すると、名前は 19 から 28 桁目で指定されます。

物理ファイルの場合に有効な項目は次のとおりです。

項目 意味

R レコード様式名

空白

フィールド名

K キー・フィールド名

注: 物理ファイルにはレコード様式を 1 つしか含めることができないので、R は 1 つだけ指定します。

論理ファイルの場合に有効な項目は次のとおりです。

項目 意味

R レコード様式名

J 結合仕様

空白

フィールド名または選択/除外の AND 条件

K キー・フィールド名

S 選択フィールド名

O 除外フィールド名

#### 関連資料

『物理ファイルおよび論理ファイルの名前 (19 から 28 桁目)』

この欄には、レコード名およびフィールド名を指定します。

70 ページの『JOIN (結合) キーワード - 結合論理ファイルのみ』

これは結合レベル・キーワードで、このキーワードを指定した結合仕様により結合される 2 つ (対) のファイルを識別するために使用します。

### 物理ファイルおよび論理ファイル用に予約済み (18 桁目)

この欄はどのファイル・タイプにも適用されません。注記のテキストの記入に使用する場合を除き、この桁は空白にしておいてください。

### 物理ファイルおよび論理ファイルの名前 (19 から 28 桁目)

この欄には、レコード名およびフィールド名を指定します。

以下のアイテムの名前を指定できます。

- この物理ファイルのレコード様式またはこの論理ファイルの 1 つまたは複数のレコード様式。
- レコード様式を構成する 1 つまたは複数のフィールド名 (レコード・レベルで FORMAT または PFILE キーワードを指定しない場合)。
- キー・フィールドとして使用される 1 つまたは複数のフィールド。
- 論理ファイルの場合、選択/除外仕様で使用される 1 つまたは複数のフィールド。

注: ファイル名は、DDS ではなく物理ファイル作成 (CRTPF) コマンドで指定します。

名前は 19 桁目から記入しなければなりません。

フィールド名または選択/除外の AND 条件を指定する場合を除き、17 桁目に名前のタイプを指定しなければなりません。

### 関連タスク

3 ページの『DDS の単一样式論理ファイルおよび複数様式論理ファイル』

単一样式論理ファイルおよび複数様式論理ファイルの場合には、レコード・レベルで PFILE キーワードを指定しなければなりません。複数様式論理ファイルでは、あるレコード様式は、そのレコード様式について PFILE キーワードで指定したすべての物理ファイルに共通するフィールドだけしか使用することができません。

4 ページの『DDS での論理ファイルの結合』

結合論理ファイルでは、複数の物理ファイルからのいくつかの異なるフィールドを組み合わせて、1 つのレコードが形成されます。結合論理ファイルの場合、レコード・レベルで JFILE キーワードを指定しなければなりません。

### 関連資料

7 ページの『物理ファイルおよび論理ファイルの名前または仕様のタイプ (17 桁目)』

この欄には名前のタイプ、または論理ファイルの場合は仕様のタイプを指定するための値を入力します。名前タイプを指定すると、名前は 19 から 28 桁目で指定されます。

5 ページの『物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)』

DDS 形式の最初の 44 桁は定位置項目と呼ばれます。

### 関連情報

DDS キーワードおよびパラメーター値の規則

### レコード様式:

17 桁目に R を指定した場合には、19 から 28 桁目に指定した名前はレコード様式名になります。

### 物理ファイルのレコード様式

物理ファイルの場合、レコード様式名は 1 つしか指定できません。レコード様式名は、次の 2 つのうちのいずれかの形で指定します。

- この物理ファイル内で個々のフィールド名が指定される新しいレコード様式の名前として指定。

レコード様式の名前には、物理ファイル作成 (CRTPF) コマンドで指定されるファイル名と同じ名前を使用することもできます。ただし、名前が固有のものでないと警告メッセージが出されます。これは、ある種の高水準言語処理プログラムでは、レコード様式名とファイル名を同じにすることができないためです。この種の高水準言語には RPG などがあります。レコード様式名とフィールド名は、システムから見て固有のものである必要はありません。つまり、他のファイルの中に同じ名前があってもかまいません。

- 別の物理ファイルの中ですでに定義されているレコード様式の名前として指定。

この場合は、FORMAT キーワードを指定しなければなりません。フィールド名および属性は指定しません。

## 単一様式論理ファイルおよび複数様式論理ファイルのレコード様式

複数のレコード様式名を指定することができます。ただし、各レコード様式名は、そのファイル内でそれぞれ固有の名前でなければなりません。例外については、該当する高水準言語の解説書を参照してください。

レコード様式名は、次の 3 つのうちのいずれかの形で指定します。

- PFILE キーワードに指定した最初の物理ファイルのレコード様式名。

レコード様式について個々のフィールド名を指定せず、しかも FORMAT キーワードも指定していない場合には、必ずこの名前を指定しなければなりません。

- この論理ファイル内で個々のフィールド名が指定される新しいレコード様式名。

この場合、すべてのフィールドについて名前を指定しなければなりません。名前のない物理ファイル・フィールドは、この論理ファイル・レコード様式には組み込まれません。RENAME キーワードおよび CONCAT キーワードのパラメーターとして指定された物理ファイル・フィールドは、この論理ファイル・レコード様式に組み込まれます。SST キーワードのパラメーターとして指定された物理ファイル・フィールドは、ほかに特別の指定がない限り、論理ファイル・レコード様式に組み込まれません。

- 物理ファイルまたは論理ファイルの中ですでに記述されているレコード様式名。

この場合には、フィールドの名前および属性は指定せずに、FORMAT キーワードを指定しなければなりません。

レコード様式名は、ファイル作成コマンドで指定されるファイルの名前と同じにすることができます。ただし、それが固有の名前でないと警告メッセージが送られます。高水準言語プロセッサによっては (たとえば RPG など)、レコード様式名をファイル名と同じにすることはできません。

PFILE キーワードをレコード様式名とともに使用すれば、そのレコード様式に関連付ける物理ファイルを指定することができます。1 つのレコード様式について、複数の物理ファイルを PFILE キーワードで指定することができます。フィールドが定義されておらず、FORMAT キーワードも指定されていない場合には、PFILE キーワードに指定した最初のファイルの様式が、すべての物理ファイルの様式として使用されます (この様式は、フィールド属性の参照、および属性と名前の検査に使用されます。)

## 結合論理ファイル

1 つの結合論理ファイルについて、レコード様式名は 1 つだけしか指定できません。レコード様式名は、この論理ファイル内に個々のフィールド名が指定される新しいレコード様式の名前として指定します。結合論理ファイルのレコード様式内の各フィールドは、19 から 28 桁目に指定した名前により識別しなければなりません。RENAME、CONCAT、および SST の各キーワードのパラメーターとして指定された物理ファイル・フィールドは、このレコード様式内の別の箇所でのフィールド名を指定した場合に限り、論理ファイルのレコード様式に組み込まれます。

JFILE キーワードは、レコード・レベルでは必須です。このキーワードには、レコード様式により結合されるいくつかの物理ファイルを指定します。

### 関連資料

63 ページの『物理ファイルおよび論理ファイルの FORMAT (様式) キーワード』

これは、レコード・レベル・キーワードで、以前に定義されたレコード様式のフィールド仕様を、この

レコード様式で共用することを指定するために使用します。定義するレコード様式の名前は、以前に定義されたレコード様式の名前と同じでなければなりません。

#### フィールド名:

17 桁目をブランクのままにした場合には、19 から 28 桁目に指定する名前はフィールド名になります。

FORMAT キーワードを指定する場合には、フィールド名を指定することはできません。

物理ファイルでは、どのフィールドにも名前を付けることが必要です。名前は、いずれもそのレコード様式内で固有のものでなければなりません。フィールド名は、DDS で指定する順序で物理バッファ内に現れます。

単一样式論理ファイルまたは複数様式論理ファイルを定義している場合には、この論理ファイルの基礎となる物理ファイルのレコード様式をそのまま使用することができます。したがって、その場合には、フィールド名を指定する必要はありません。

物理ファイルのレコード様式をそのまま使用しない場合には、論理ファイル内に指定する各フィールドには名前を付けることが必要です。単一样式論理ファイルまたは複数様式論理ファイルの場合には、各フィールド名は、そのレコード様式内で固有の名前でなければならず、物理ファイルのレコード様式のフィールドに対応していなければなりません。論理ファイルを使用するプログラムには、フィールド名の指定順序と同じ順序でフィールドが渡されます。

論理ファイル・レコード様式内のフィールドに付ける名前には、通常、物理ファイル・レコード様式内の対応するフィールドの名前と同じものを使用します。両者が違っている場合には、RENAME キーワードを用いて、2 つの名前を同じにすることが必要です。物理ファイルの複数のフィールドを連結したものを、論理ファイルのレコード様式内の 1 つのフィールドにすることもできます。SST キーワードを用いて、物理ファイルのフィールドのサブストリングを論理ファイル様式に記述することもできます。

**注:** 論理ファイルにおけるフィールド名の指定順序は、重要な意味を持っています。論理ファイル内の 1 つのレコード様式に、(RENAME または CONCAT を用いて) 同一物理フィールドを 2 回以上指定した場合には、論理ファイルでのフィールドの指定順序が、物理ファイルへのデータの転送順序となります。したがって、論理ファイル内で最後に指定された時点のフィールドの値が、その物理レコード内の値となります。

#### 関連資料

49 ページの『CONCAT (連結) キーワード - 論理ファイルのみ』

これはフィールド・レベル・キーワードで、物理ファイルのレコード様式からの 2 つ以上のフィールドを、定義中の論理ファイルのレコード様式内で連結して 1 つのフィールドにしたい場合に使用します。連結後にできるフィールドの名前を 19 から 28 桁目に指定しなければなりません。

#### キー・フィールド名:

17 桁目に K を指定した場合には、19 から 28 桁目に指定する名前はキー・フィールド名になります。

この名前は、物理ファイル・レコード様式内にあるフィールド名のどれかでなければなりません。このフィールドの内容は、データベースからのレコードの検索順序を決めるために使用されます。キーの指定は任意です。キー・フィールドを指定しなかった場合には、デフォルトの順序は到着順 (レコードがファイルに入れられた順序) になります。

キー・フィールド (および必要に応じて、選択/除外フィールド) は、論理ファイル・メンバーのレコード様式に対するキー順アクセス・パスを定義するために使用します。論理ファイル・メンバーには、論理ファイ



ル作成 (CRTLFL) コマンドまたは論理ファイル・メンバー追加 (ADDLFM) コマンドの DTAMBR5 パラメーターに指定した物理ファイル・メンバーが含まれます。

ファイルからの読み取りのレコードの順序を変更するためには、順序付けキーワードを指定します。順序付けキーワードには、ALTSEQ、NOALTSEQ、SIGNED、UNSIGNED、ABSVAL、ZONE、DIGIT、DESCEND、FCFO、FIFO、および LIFO があります。

キー・フィールドについて順序付けキーワードを指定しなかった場合には、デフォルト値により、そのキー・フィールドについての順序は昇順になります。文字、16 進数、日付、時刻、および時刻スタンプ・フィールドのデフォルトの属性は UNSIGNED です。数字フィールドのデフォルトの属性は、次に示すゾーン 10 進数フィールド (35 桁目が S) の場合を除き、SIGNED になります。

- ファイル・レベルで ALTSEQ を指定した場合には、そのファイルのすべてのゾーン 10 進数キー・フィールドは、デフォルトにより UNSIGNED になります。
- ゾーン 10 進数キー・フィールドについて DIGIT または ZONE を指定した場合には、そのフィールドは、デフォルトにより UNSIGNED になります。

1 つの論理ファイルについて複数のレコード様式を指定する場合、または PFILE キーワードで複数の物理ファイルを指定する場合には、その論理ファイルのすべてのレコード様式に対して少なくとも 1 つは、キー・フィールドを指定しなければなりません。

1 つのキーは複数のキー・フィールドを持つことができます。これを複合キーと呼びます。複合キーには、重要度 (高いものから低いもの) 順にキー・フィールド名を指定し、別々の行に個々のキー・フィールド名を指定します。

図 2 に、2 つのレコード様式を持つ複数様式論理ファイルを示します。レコード様式の 1 つは、複合キーを使用しています。この例では、RECORD1 には FIELD1 という単一キー・フィールドがあります。RECORD2 には、FIELD4 および FIELD5 から成る複合キーがあります。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1          PFILE(PF1)
00020A          FIELD1
00030A          FIELD2
00040A          FIELD3
00050A          K FIELD1
00060A*
00070A          R RECORD2          PFILE(PF2)
00080A          FIELD4
00090A          FIELD5
00100A          K FIELD4
00110A          K FIELD5
      A

```

図 2. 2 つのレコード様式を持つ複数様式論理ファイルの指定

キー・フィールドを指定しなかった場合には、定義中の論理ファイルは到着順アクセス・パスを持つこととなります。

- | キーを構成するフィールドは 120 に制限されています。キーの合計の長さは 32 768 バイトを超えてはな
- | りません。(FCFO キーワードを指定する場合には、キーの合計の長さは 32 763 バイトを超えてはなりま
- | せん。DDM 環境では、キーの長さは 12 000 に制限されます。) キーの合計の長さとは、各キー・フィ
- | ールドの長さを合計したものです。ヌル値を許容するキー・フィールドがある場合には、ヌル値を許容する
- | キー・フィールド 1 つにつき 1 バイトを加算してください。オペレーティング・システムでは、キーにヌ
- | ル値が入っているかどうかを判別するために 1 バイトだけ余計に使用されます。可変長のキー・フィール

1 ドがある場合は、可変長フィールド 1 つにつき 2 バイトを加算してください。オペレーティング・システムでは、フィールドの割り振られた長さを保管するために 2 バイトだけ余分に使用されます。

論理ファイルに複数のレコード様式を指定した場合には、始めの \*NONE キー・フィールド位置に 1 バイトの追加が必要です。追加されたキー・フィールド位置のおののおのについても 1 バイトの追加が必要です。オペレーティング・システムでは、別の物理ファイルのレコードに重複したキー値がある場合に 1 バイトだけ余計に使用されます。

たとえば、あるキーが、FIELD A、FIELD B、および FIELD C というフィールドにより、この順序で構成されるものとすれば、DDS は 図 3 に示されるようになります。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* SAMPLE COMPOSITE KEY (PHYSICAL FILE)
00020A      R RECORD
00030A      FIELD A          3 0
00040A      FIELD B          3 0
00050A      FIELD C          3 0
00060A      FIELD D          3 0
00070A      K FIELD A
00080A      K FIELD B
00090A      K FIELD C
      A

```

注: 行 00070 から 00090 までが複合キーを構成しています。

図 3. 複合キー

レコードは、下記の順序で順序付けされます。

1. まず、FIELD A の内容に基づいて順序付けされます。
2. 2 つ以上のレコードについて FIELD A の値が同じであった場合には、オペレーティング・システムにより、FIELD B の値に基づいてそれらのレコードが順序付けされます。
3. FIELD A および FIELD B のどちらについても値が同じであるレコードが 2 つ以上ある場合には、それらのレコードは FIELD C の値に基づいて順序付けされます。

次のようなファイルがあるとします。

レコード	FIELD A	FIELD B	FIELD C
1	333	99	67
2	444	10	45
3	222	34	23
4	222	12	01
5	222	23	45
6	111	06	89
7	222	23	67

すべてのフィールドについて昇順の順序付けを適用するものとすれば、レコードは次の順序で検索されます。

レコード	FIELD A	FIELD B	FIELD C
6	111	06	89
4	222	12	01
5	222	23	45
7	222	23	67
3	222	34	23

レコード	FIELD A	FIELD B	FIELD C
1	333	99	67
2	444	10	45

以下の情報に注意してください。

- レコード 3、4、5、および 7 は FIELD A の内容が同じなので、これらのレコードの順序を決定するフィールドは FIELD B です。
- この 4 つのレコードのうち、5 と 7 は FIELD B の値が同じなので、これらのレコードの順序を決定するフィールドは FIELD C です。
- FIELD C の値も同じであった場合、レコードは先入れ先出し法 (FIFO)、後入れ先出し法 (LIFO)、または変更順検索 (FCFO) のいずれかの順序で検索されます。順序を決める場合には、FIFO キーワード、LIFO キーワード、または FCFO キーワードのいずれかを指定します。キー値の重複を避けるためには、UNIQUE キーワードを指定してください。

負 (-) の値をとるキー・フィールドを含む例については、84 ページの『物理ファイルおよび論理ファイルの SIGNED (符号付き) キーワード』を参照してください。

物理ファイル作成 (CRTPF) コマンドで FILETYPE(\*SRC) を指定した場合、またはソース物理ファイル作成 (CRTSRCPF) コマンドの場合には、キー・フィールド仕様について特別の制約があります。

論理ファイルでフィールドをキー・フィールドとして指定する場合には、次の規則があります。

- 単一様式論理ファイルおよび複数様式論理ファイルの場合には、キー・フィールド名と定義したフィールドとの突き合わせは、次の検索順序に従って行われます。
  - DDS の 19 から 28 桁目に指定されたフィールド。
  - CONCAT キーワードまたは RENAME キーワードのパラメーターとして指定されたフィールド。

同じフィールド名が 2 回以上指定されている場合には、最初に指定されたフィールドが使用されます。

CONCAT キーワードまたは RENAME キーワードに指定したフィールド名と、それに対応する 19 から 28 桁目のフィールド名との両方を、ともにキー・フィールドとして指定することはできません。

SST キーワードのパラメーターとして指定したフィールド名は、論理ファイル様式内の別の箇所では定義されていない限り、キー・フィールドとして使用することはできません。

- 結合論理ファイルの場合には、指定するキー・フィールド名は、19 から 28 桁目にフィールド・レベルで指定されていて、しかも 1 次ファイル (JFILE キーワードに最初に指定した物理ファイル) に定義されているフィールドでなければなりません。

注: CONCAT、RENAME、または SST のいずれかのキーワードのパラメーター値として指定してあっても、結合論理ファイルの 19 から 28 桁目に指定していないフィールドは、キー・フィールドとして指定することはできません。

数字フィールドと、文字フィールドまたは 16 進数フィールドとを連結する場合には、その数字フィールドをキー・フィールドとして指定することはできません。ゾーン 10 進数フィールドとその他の数値データ・タイプのフィールドを連結する場合には、ゾーン 10 進数ではない方の数字フィールドをキー・フィールドとして指定することはできません。

14 ページの図 4 に、連結フィールドがキー・フィールドとして使用できる場合と使用できない場合を示します。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
00010A	R RECORD1	PFILE(PF1)
00020A	FLD1	
00030A	FLD2	
00040A	Z	CONCAT(ZFLD PFLD)
00050A	A	CONCAT(AFLD NFLD)
00060A	K ZFLD	
00070A	K AFLD	
A		

図 4. 正しい連結フィールドと正しくない連結フィールド

物理ファイル PF1 において、ZFLD はゾーン 10 進数で、PFLD はパック 10 進数です。したがって、フィールド Z はゾーン 10 進数フィールドであり、PFLD をキー・フィールドとして使用することはできません。ZFLD と Z は、同じレコード様式内でなければ、キー・フィールドとして使用することができます。

同じく物理ファイル PF1 において、AFLD は文字フィールドで、NFLD は数字フィールドです。したがって、A は文字であり、NFLD をキー・フィールドとして使用することはできません。AFLD と A は、同じレコード様式内でなければ、キー・フィールドとして使用することができます。

#### 関連資料

49 ページの『CONCAT (連結) キーワード - 論理ファイルのみ』

これはフィールド・レベル・キーワードで、物理ファイルのレコード様式からの 2 つ以上のフィールドを、定義中の論理ファイルのレコード様式内で連結して 1 つのフィールドにしたい場合に使用します。連結後にできるフィールドの名前を 19 から 28 桁目に指定しなければなりません。

#### DDS アクセス・パス・キーワード:

1 つまたは複数のアクセス・パス・キーワードを指定することにより、オペレーティング・システムによるキー値の作成方法と使用方法を変えることができます。

アクセス・パス・キーワードは次のとおりです。

ファイル・レベル	キー・フィールド・レベル
ALTSEQ	DESCEND
FCFO	DIGIT
FIFO	SIGNED
LIFO	UNSIGNED
REFACCPH	ZONE
UNIQUE	

複合キーを構成する個々のキー・フィールドには、それぞれ別のアクセス・パス・キーワードを指定することができます。

#### 複数のレコード様式がある DDS 論理ファイル:

1 つの論理ファイルについて複数のレコード様式を指定する場合には、その論理ファイル内の各レコード様式に対して、少なくとも 1 つはキー・フィールドを指定しなければなりません。

この場合、各キーについて同数のキー・フィールドを指定する必要はありません。また、1 つのレコード様式内に指定するキー・フィールドは、同じ論理ファイルの他のレコード様式内の対応するキー・フィールド

と同じフィールド属性および同じアクセス・パス・キーワードを持っていなければなりません。フィールド・タイプおよび長さが同じであっても、可変長キー・フィールドについては、固定長キー・フィールドに位置合せすることができません。

論理ファイル・メンバーが各レコード様式のレコードを順序付けるための単一アクセス・パスを持つことができるようにするために、各レコード様式ごとに 1 つのキーが必要です。レコードは、論理ファイルの基礎となる各種の物理ファイル・メンバーから戻されるさいに、論理ファイル・メンバーのアクセス・パスのキー・フィールドの値に従って組み合わせられます。

論理ファイル・メンバーのレコードを順序付けするさいに、オペレーティング・システムは、キー・フィールドの値を連結することによって、各レコードについてのキー値を作成します。このキー値は、プログラムが使用するアクセス・パスを作成するために使用されます。

複合キーを構成する各キー・フィールドには、それぞれキー位置があります。最初に指定したキー・フィールドは位置 1 に置かれ、2 番目に指定したキー・フィールドは位置 2 に置かれ、後続のキー・フィールドも同様の順序で位置が決まります。論理ファイルに対する入出力操作では、i5/OS™ プログラムは、データベースに書き込んだりデータベースから読み取ったりするレコードについて、キー値の比較を行います。複数のレコード様式を持つ論理ファイルを作成する場合は、(異なるキー・フィールドを指定するかどうかにかかわらず) オペレーティング・システムは キー位置属性検査を行います。キー位置属性検査に合格するためには、同じキー位置にある異なるレコード様式のキー・フィールドが、同じデータ・タイプ、同じ長さ、同じ小数点以下の桁数を持ち、キー・フィールド・レベルで指定された同じアクセス・パス・キーワードを持っていなければなりません。これによって、入出力操作時に行われる比較から意味のあるレコード順序が確保されます。

いくつかの浮動小数点数フィールドをキー・フィールドとして使用する場合には、それらの浮動小数点数フィールドは、同じデータ・タイプと精度を持たなければなりません。ただし、長さおよび小数点以下の桁数は同じである必要はありません。

次の例では、FIELD1、FLD1、および F1 は同じ属性を持っていなければならず、FIELD2、FLD2、および F2 は同じ属性を持っていなければなりません。FIELD1、FLD1、および F1 はキー位置 1 になり、FIELD2、FLD2、および F2 はキー位置 2 になります。一方のレコード様式のキー・フィールドの数を他方のレコード様式より多くすることも可能であり、この場合、多い分のフィールドについては、キー位置属性検査は必要ありません。FLD3 がこの種のフィールドに該当します。

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
00010A	R	RECORD1	PFILE(PF1)
00020A		FIELD1	5 0
00030A		FIELD2	10
00040A		FIELD3	10
00050A	K	FIELD1	
00060A	K	FIELD2	DESCEND
00070A*			
00080A	R	RECORD2	PFILE(PF2)
00090A		FLD1	5 0
00100A		FLD2	10
00110A		FLD3	20
00120A	K	FLD1	
00130A	K	FLD2	DESCEND
00140A	K	FLD3	
A*			
A	R	RECORD3	PFILE(PF3)
A		F1	5 0
A		F2	10
A		F3	30
A	K	F1	
A	K	F2	DESCEND
A			
A			

図5. キー・フィールド属性検査

複数のレコード様式がある論理ファイルのキー・フィールドの例では、ITEM という名前のフィールドがそれぞれのキーに指定されます。レコード様式 INVFMT および ACTFMT については、ITEM が、指定されている唯一のキー・フィールドです。レコード様式 ORDFMT については複合キーが指定されています。この複合キーには、ITEM、SHPYR、SHPMO、および SHPDA が含まれています。キーに使用される各フィールドは、フィールド・レベルでも存在するものでなければなりません。したがって、ITEM は、INVFMF のためにこの論理ファイルに複写できるように、物理ファイル INVENTORY のレコード様式に含まれていなければなりません。ITEM は、また、ACTFMT のためにもこの論理ファイルに複写できるように、論理ファイル ACCOUNTL のレコード様式にも含まれていなければなりません。さらに、物理ファイル ACCOUNTS にも ITEM が含まれていなければなりません。

#### 関連タスク

3 ページの『DDS の単一样式論理ファイルおよび複数様式論理ファイル』

単一样式論理ファイルおよび複数様式論理ファイルの場合には、レコード・レベルで PFILE キーワードを指定しなければなりません。複数様式論理ファイルでは、あるレコード様式は、そのレコード様式について PFILE キーワードで指定したすべての物理ファイルに共通するフィールドだけしか使用することができません。

#### DDS ファイルを作成する際のキー・フィールドでの \*NONE の使用:

同じキー位置を持つキー・フィールドを比較すべきでない場合が 2 つあります。

次の 2 つの条件に該当する場合です。

- 各キー・フィールドのフィールド属性 (データ・タイプ、長さ、小数点以下の桁数、またはフィールド・レベルのアクセス・パス・キーワード) が同じでない場合。
- 各キー・フィールドの属性は同じであるが、キー・フィールドを組み合わせる順序付けることを避けたい場合。

キー・フィールドの相互の無用な比較を避けるためには、キー・フィールドのどれか 1 つの代わりに \*NONE を指定し、その位置にあったキー・フィールドを次のキー位置に移します。オペレーティング・シ

システムは、\*NONE の前後のキー位置の値を比較しますが、論理ファイルの DDS にレコード様式が指定されている順序と同じ順序で、対象となるレコードを検索します。

後続行で \*NONE を 2 回以上指定して、キー・フィールドを適切なキー位置にずらし、アプリケーションにとって適切な形でキー・フィールド属性の比較が行われるようにすることができます。

図 6 に、キー・フィールドとしての \*NONE を示します。

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
A          R RECORD1          PFILE(PF1)
A          FIELD1             6A
A          FIELD2             4A
A          FIELD3             10A
A          K FIELD1
A          K FIELD2
A          K FIELD3
A
A          R RECORD2          PFILE(PF2)
A          FLD1               3A
A          FLD2               4A
A          FLD3               12A
A          K *NONE
A          K FLD2
A
A          R RECORD3          PFILE(PF3)
A          F1                 6A
A          F2                 4A
A          F3                 10A
A          K F1
A          K F2
A          K F3
A

```

図 6. キー・フィールドとしての \*NONE の指定

図 6 では、FIELD2、FLD2、および F2 の属性は同じでなければなりません。2 番目のレコードの最初のキー・フィールドに \*NONE を指定してあるので、FIELD1 と F1 (最初のレコードの最初のキー・フィールド、および 3 番目のレコードの最初のキー・フィールド) は、属性が同じでなければなりません。FIELD3 と F3 も同じ属性を備えていなければなりません。この 2 つのフィールドに対応するフィールドは、2 番目のレコード様式にはありません。

#### 関連資料

『キー・フィールドの指定 (例 1)』

この例の論理ファイルでは、CLSHST (研修記録) と JOBHST (課程記録) という 2 つの異なるレコード様式に従って、2 つの物理ファイルのレコードが処理されます。

18 ページの『キー・フィールドの指定 (例 2)』

この例では、別の論理ファイルを通して、例 1 と同じ 2 つの物理ファイルが処理されます。ただし、この論理ファイルの 2 番目のレコード様式には、キー位置 2 に \*NONE が指定されています。

#### キー・フィールドの指定 (例 1):

この例の論理ファイルでは、CLSHST (研修記録) と JOBHST (課程記録) という 2 つの異なるレコード様式に従って、2 つの物理ファイルのレコードが処理されます。

この論理ファイルでは、キー位置 1 に受講者番号 (EMPNBR) を指定することにより、2 つの物理ファイルにあるレコードを組み合わせ、EMPNBR に従って順序付けすることができます。

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R CLSHST                PFILE(CLSHSTP)
00020A      K EMPNBR  1
00030A      K CLSDTE  2
00040A*
00050A      R JOBHST                PFILE(JOBHSTP)
00060A      K EMPNBR  1
00070A      K JOBDETE  2
  A

```

図7. キー・フィールドの指定 (例 1)

レコード様式	キー位置	
	1	2
CLSHST	EMPNBR	CLSDTE
JOBHST	EMPNBR	JOBDETE

EMPNBR のキー値が同じであるレコードは、すべて同じ受講者に関するものです。同一受講者の全レコードを組み合わせて順序付けし、研修および課程を 1 つの記録としてまとめたい場合には、DDS ファイルを作成する際のキー・フィールドでの \*NONE の使用のトピックに示すように、2 つのレコード様式のキー位置 2 に CLSDTE (研修の日付) および JOBDETE (課程の日付) を指定します。

課程の日付と研修の日付は、その課程または研修が開始された日付 (月/年) を表すものとします。たとえば、3 人の受講者のレコードは、次の順序で検索されます。

EMPNBR	CLSDTE	JOBDETE	説明
1005	3/79		研修終了
1005	4/79		新しい課程を始めるために中止
1005		4/79	課程終了
1005	6/79		研修終了
1006		1/79	課程終了
1006		2/79	課程終了
1006	3/79		研修終了
1006	5/79		転属
1007		1/79	課程終了
1007		4/79	課程終了
1007		7/79	課程終了
1007	8/79		病気のため中止

上記のような報告書を作成することにより、各受講者の継続的な記録をとることができます。

### 関連概念

16 ページの『DDS ファイルを作成する際のキー・フィールドでの \*NONE の使用』  
 同じキー位置を持つキー・フィールドを比較すべきでない場合が 2 つあります。

### キー・フィールドの指定 (例 2):

この例では、別の論理ファイルを通して、例 1 と同じ 2 つの物理ファイルが処理されます。ただし、この論理ファイルの 2 番目のレコード様式には、キー位置 2 に \*NONE が指定されています。



```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R CLSHST                PFILE(CLSHSTP)
00020A      K EMPNBR  1
00030A      K CLSDTE  2
00040A*
00050A*
00060A      R JOBHST                PFILE(JOBHSTP)
00070A      K EMPNBR  1
00080A      K *NONE  2
00090A      K JOBDATE  3
      A

```

図 8. キー・フィールドの指定 (例 2)

レコード様式	キー位置		
	1	2	3
CLSHST	EMPNBR	CLSDTE	*NONE
JOBHST	EMPNBR	*NONE	JOBDATE

DDS ファイルの作成の際のキー・フィールドでの \*NONE の使用のトピックの例に示したように、2 つの物理ファイルにあるすべてのレコードは、まず受講者番号 (EMPNBR) に従って組み合わせられ順序付けされます。しかし、各受講者のレコードについては、まず研修の日付 (CLSDTE) に、次に課程の日付 (JOBDATE) に従って、組み合わせと順序付けが行われます。その例に使用したものと同一連のレコードは、今度は次の順序で検索されます。

EMPNBR	CLSDTE	JOBDATE	説明
1005	3/79		研修終了
1005	4/79		新しい課程を始めるために中止
1005	6/79		研修終了
1005		4/79	課程終了
1006	3/79		研修終了
1006	5/79		転属
1006		1/79	課程終了
1006		2/79	課程終了
1007	8/79		病気のため中止
1007		1/79	課程終了
1007		4/79	課程終了
1007		7/79	課程終了

いくつかの連続したレコード様式と同じキー位置に \*NONE がある場合には、これらの様式はそれぞれ前後のレコード様式と関連しあって 1 つのセットを形成し、順序付けのさいにはセット全体で 1 つのレコード様式としての役割を果たします。\*NONE のあとに指定されたキー・フィールドは、セット内のレコード様式の組み合わせと順序付けのために使用されます。次の例は、複数のレコード様式が 1 つのセットとしてどのような働きをするかを示しています。

#### 関連概念

16 ページの『DDS ファイルを作成する際のキー・フィールドでの \*NONE の使用』  
 同じキー位置を持つキー・フィールドを比較すべきでない場合が 2 つあります。

#### キー・フィールドの指定 (例 3):

この例では、5 つの物理ファイルを対象とする受講者論理ファイルについて考えてみましょう。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R  EMPMST                PFILE(EMPMSTP)
00020A      K  EMPNBR  1
00030A*
00040A      R  CLSREG                PFILE(CLSREGP)
00050A      K  EMPNBR  1
00060A      K  CLSDTE  2
00070A*
00080A      R  CLSHST                PFILE(CLSHSTP)
00090A      K  EMPNBR  1
00100A      K  CLSDTE  2
00110A*
00120A      R  JOBHST                PFILE(JOBHSTP)
00130A      K  EMPNBR  1
00140A      K  *NONE  2
00150A      K  JOBDTE  3
00160A*
00170A      R  ACTHST                PFILE(ACTHSTP)
00180A      K  EMPNBR  1
00190A      K  *NONE  2
00200A      K  ACTDTE  3
  A

```

図9. キー・フィールドの指定 (例 3)

レコード様式	キー位置	1	2	3
EMPMST	EMPNBR		*NONE	*NONE
CLSREG	EMPNBR		CLSDTE	*NONE
CLSHST	EMPNBR		CLSDTE	*NONE
JOBHST	EMPNBR		*NONE	JOBDTE
ACTHST	EMPNBR		*NONE	ACTDTE

レコードの組み合わせと順序付けは、次のように行われます。

1. すべてのレコードが受講者番号に従って組み合わせられ、順序付けされる。
2. 特定の受講者について、次のようにレコードが順序付けされる。
  - a. マスター・レコード (EMPMST 様式)
  - b. CLSREG 様式および CLSHST 様式のレコード。これは、CLSDTE (キー位置 2) の値に基づいて、組み合わせられ順序付けされます。
  - c. JOBHST 様式および ACTHST 様式のレコード。これは、JOBDTE および ACTDTE (キー位置 3) の値に基づいて、組み合わせられ順序付けされます。

キー定義に \*NONE を指定すると、順序付けは次のように行われます。

- 2 つの隣接する様式 CLSHST および JOBHST において、2 番目のキー位置の指定は、一方が \*NONE であるのに対してもう一方はフィールド名 CLSDTE です。したがって、その前のキー位置 (位置 1) のあとで、2 つの様式の間境界ができます。この境界より前にある様式のレコードは、EMPNBR の値だけに基づいて、境界より下にある様式のレコードと組み合わせられ順序付けされます。
- 様式 EMPMST の 2 番目のキー位置にも暗黙の \*NONE があるので、ここにも同様の境界ができます。
- 様式 JOBHST および ACTHST はいずれもキー位置 2 に \*NONE が指定されているので、これらの様式は 2 つで 1 つのセットとなります。したがって、これらの 2 つの様式のレコードは、JOBDTE の値と ACTDTE の値の比較に基づいて組み合わせられ、順序付けされます。

前のキー仕様によって定義されるレコード順序は、完全に、様式の指定順序によって決まります。たとえば、JOBHST が CLSHST の前に指定されていた場合、キー位置 2 の順序は次のようになります。

\*NONE, CLSDTE, \*NONE, CLSDTE, \*NONE

この場合には、様式 CLSREG の CLSDTE の値は様式 CLSHST の CLSDTE の値と組み合わせられて順序付けされることはなく、また JOBDATE も ACTDTE と組み合わせられて順序付けされることはありません。

#### キー・フィールドの指定 (例 4):

この例では、1 人の受講者が同じ研修を再度受けた場合を想定しています。EMPNUM および CLSDTE の値が両方とも同じである 2 つのレコードの順序付けを行うために、レコード様式 CLSHST には、3 番目のキー・フィールド DATE が指定されます。

しかし、他の様式の位置 3 には JOBDATE および ACTDTE があるため、次の使用可能キー位置 (位置 3) に DATE を指定することはできません。位置 3 に DATE を指定すると DATE の属性と CLSHST および JOBHST の属性との比較が行われ、キー定義が拒否されます。

必要な順序付けが行われるようにするためには、DATE の前に \*NONE を指定し、DATE をキー位置 4 にずらします。

DATE フィールドは、図 10 に示すように、位置 4 にきます。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R EMPMST                PFILE(EMPMSTP)
00020A      K EMPNBR
00030A*
00040A      R CLSREG                PFILE(CLSREGP)
00050A      K EMPNBR
00060A      K CLSDTE
00070A*
00080A      R CLSHST                PFILE(CLSHSTP)
00090A      K EMPNBR
00100A      K CLSDTE
00110A      K *NONE 1
00120A      K DATE 1
00130A*
00140A      R JOBHST                PFILE(JOBHSTP)
00150A      K EMPNBR
00160A      K *NONE
00170A      K JOBDATE
00180A*
00190A      R ACTHST                PFILE(ACTHSTP)
00200A      K EMPNBR
00210A      K *NONE
00220A      K ACTDTE
A

```

図 10. キー・フィールドの指定 (例 4)

レコード様式	キー位置	1	2	3	4
EMPMST	EMPNUM		*NONE	*NONE	*NONE
CLSREG	EMPNUM		CLSDTE	*NONE	*NONE
CLSHST	EMPNUM		CLSDTE	*NONE	DATE
JOBHST	EMPNUM		*NONE	JOBDATE	*NONE
ACTHST	EMPNUM		*NONE	ACTDTE	*NONE

キー位置 4 に DATE を指定すると、物理ファイル CLSHSTP 内のレコードのうち、EMPNBR および CLSDTE の値が両方とも同じであるすべてのレコードが、DATE の値に従って組み合わされ順序付けされます。

注: 上記の各例では、順序付けを確認するために実際に値をキーに入れてありますが、複数のレコード様式を持つ論理ファイルについて必要上 \*NONE を指定した場合には、必ずしも重複キー値を予測できるわけではありません。

### 選択/除外フィールド名:

選択/除外フィールドは、ユーザー・プログラムがこのレコード様式を用いてレコードを検索するさいにどのようにレコードを選択または除外するかを、オペレーティング・システムに指示するために使用します。選択/除外の対象となるレコードは、このレコード様式についての PFILE キーワードまたは JFILE キーワードで指定された物理ファイルのレコードだけです。

論理ファイルの場合の選択/除外フィールドには、次の規則が適用されます。

- 選択/除外フィールドの指定ができるのは、キー・フィールドも同時に指定した場合、または当該ファイルについて DYNSLT キーワードを指定した場合だけです。アプリケーションでキー・フィールドが必要とされない場合には、キー・フィールドとしての要件を満たすために、\*NONE をキー・フィールドとして指定することができます。
- 単一樣式論理ファイルおよび複数様式論理ファイルの場合、オペレーティング・システムは、次の検索順序を使用して、選択/除外フィールド名と定義されたフィールドとの突き合わせを行います。
  - DDS の 19 から 28 桁目に指定されたフィールド。
  - CONCAT キーワードまたは RENAME キーワードのパラメーターとして指定されたフィールド。

同じフィールド名が 2 回以上指定されている場合には、最初に指定されたフィールドが使用されます。

CONCAT キーワードまたは RENAME キーワードに指定したフィールド名と、それに対応する 19 から 28 桁目のフィールド名の両方を、ともに選択/除外フィールドとして指定することはできません。

SST キーワードのパラメーターとして指定したフィールド名は、論理ファイルのレコード様式内の他の箇所では定義されていない限り、選択/除外フィールドとして使用することはできません。

結合論理ファイルの場合には、ここに指定する選択/除外フィールド名は、19 から 28 桁目にフィールド・レベルで指定されているものでなければなりません。

選択/除外フィールドを使用するときには、17 桁目に S または O を指定してください。S または O のいずれかを指定すると、選択/除外比較ステートメントは OR 関係になります。OR 関係で結ばれた選択/除外比較ステートメントは、システムによって、互いに独立に処理されます。つまり、選択または除外の比較条件が適合すると、レコードは選択されるか除外されるかのどちらかになります。比較条件が適合しないと、システムは次の比較に移ります。

17 桁目にブランクを指定すると、選択/除外比較ステートメントは AND 関係になります。AND 関係で結ばれた比較条件をすべて満たさないと、レコードは選択または除外されません。23 ページの図 11 および 24 ページの図 12 を参照してください。19 から 28 桁目にはフィールド名を指定します。このフィールドの実行時の内容と、このフィールドについて指定された選択/除外キーワードとが照らし合わされて、当該レコードを選択すべきか除外すべきかが決定されます。選択/除外キーワードには、COMP、RANGE、および VALUES があります。1 つのファイルについての最後の選択/除外仕様は、ALL キーワードを用いて指定することができます。ただし、この場合には、フィールド名を指定してはなりません。

選択/除外フィールドは、物理ファイルのレコード様式および論理ファイルのレコード様式の両方に入っているものでなければなりません。選択/除外ステートメントは、当該のレコード様式についてのすべてのフィールド・レベルおよびキー・フィールド・レベルの項目の後に指定しなければなりません。また、同一レコード様式について、選択および除外の両方を指定することができます。以下の情報に注意してください。

- 1 つのレコード様式について選択および除外の両方を指定する場合には、それを指定する順序が重要な意味を持ちます。

選択/除外ステートメントは、それを指定した順序に従って処理されます。あるレコードが 1 つのステートメントを満足すると、そのレコードは指定に従って選択または除外され、そのレコードについては残りの選択/除外ステートメントとの照合は行われません。24 ページの図 13 を参照してください。

- 選択および除外の両方のステートメントを指定した場合には、指定した基準のいずれにも合致しないレコードを、選択の対象とするか除外の対象とするかを指定することができます。
- ALL キーワードを指定しなかった場合、どの値にも合致しないレコードに対しては、最後に指定したステートメントのタイプ (選択または除外) と逆の処置がとられます。つまり、選択が指定されていれば、選択基準に合致しないレコードが除外され、除外が指定されていれば、除外基準に合致しないレコードが選択されます。

1 つの論理ファイル内で指定できる選択/除外ステートメントの数には制限があります。多数の選択/除外ステートメントを指定したために、ファイルが作成できなくなった場合には、重要度の低い方から順に次のような変更を加えることによって、ファイルのオーバーヘッドを軽減してください。

- ファイル内のレコード様式の数減らす。
- PFILE キーワードまたは JFILE キーワードに指定した物理ファイルの数減らす。
- 選択/除外仕様で用いるフィールドの数減らす (同じフィールドは、複数回指定した場合でも 1 個として数えます)。

浮動小数点数フィールドを選択/除外フィールドとして指定することはできません。

選択/除外でアクセス・パスをもつことは可能であり、到着順でファイルが処理されます。たとえば、CPYF を FROMRCD(1) で指定したり、高水準言語でキー処理を要求できない場合があります。この場合には、処理方法は、DYNSLT キーワードを指定した場合と同じになります。

図 11 に、AND 関係で結んだ選択ステートメントを用いて選択/除外フィールドを指定する方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1                PFILE(PARTS)
00020A      PNO
00030A      DSC
00040A      UPR
00050A      QOH
00060A      K PNO
00070A      S UPR                    COMP(GT 5.00)
00080A      QOH                    COMP(LT 10)
00090A      0                       ALL
      A

```

図 11. 選択/除外フィールドの指定 (例 1)

図 11 では、2 つの選択ステートメントを満足した場合に限り、レコードが選択されます。最初のステートメントでは、フィールド UPR の値が 5.00 より大きいレコードを選択し、2 番目のステートメントでは、フィールド QOH の値が 10 より小さいレコードを選択します。フィールド QOH については 17 桁目に

S が指定されていません。したがって、これらの選択ステートメントは互いに AND 関係にあります。あるレコードがプログラムによって読み取られるためには、そのレコードが、指定された条件を両方とも満たしていなければなりません。

図 12 に、AND 関係で結ばれた 2 つの選択ステートメントと OR 関係にある除外ステートメントを用いて、選択/除外フィールドを指定する方法を示します。

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R RECORD1      PFILE(PARTS)
00020A      PNO
00030A      DSC
00040A      UPR
00050A      QOH
00060A      K PNO
00070A      O DSC          COMP(EQ 'HAMMER')
00080A      S UPR          COMP(GT 5.00)
00090A      QOH          COMP(LT 10)
00100A      0            ALL
      A

```

図 12. 選択/除外フィールドの指定 (例 2)

図 12 では、次の 2 つの条件の両方を満足すれば、レコードはプログラムに渡されます。

- フィールド DSC の値が HAMMER と等しくない。
- フィールド UPR の値が 5.00 より大きく、かつ、フィールド QOH の値が 10 より小さい。

図 13 に同じ選択/除外の論理を指定するためのいくつかの方法を示します。

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      S ST          COMP(EQ 'NY')
00020A      REP          COMP(EQ 'JSMITH') 1
00030A      YEAR        COMP(LT 78)
00040A      0            ALL
      A
00050A      O YEAR      COMP(GE 78)
00060A      S ST          COMP(EQ 'NY') 2
00070A      REP          COMP(EQ 'JSMITH')
00080A      0            ALL
      A
00090A      O REP      COMP(NE 'JSMITH')
00100A      O ST      COMP(NE 'NY') 3
00110A      S YEAR      COMP(LT 78)
00120A      0            ALL
      A

```

図 13. 選択/除外フィールドの指定 (例 3)

図 13 では、ニューヨーク州の JSMITH という名前の営業担当員について、1978 年より前のすべてのレコードを選択しようとしています。この場合に、3 つのコーディング方法があります。

1. すべてのレコードについて、ST、REP、および YEAR の各選択フィールドが比較され、選択または除外が行われます。
2. 1978 年以降のすべてのレコードは、最初の比較で除外されます。次に、1978 年より前のレコードだけについて、ST および REP が比較されます。この場合には、2 つの選択フィールドだけについて、条件を満足していればよいわけです。この方法の方が、1 の方法より効率的です。

3. ニューヨークの JSMITH に関連のないレコードはすべて、最初と 2 番目の比較で除外されます。その上で、残った全レコードについて、YEAR の値の比較が行われます。これは、1 または 2 よりも効率のよい方法です。

#### 関連資料

38 ページの『ALL (すべて) キーワード - 論理ファイルのみ』

これは選択/除外フィールド・レベル・キーワードで、当該論理ファイルについての他のすべての選択/除外仕様の処理が終わったあとの処置を指定するために使用します。

### 物理ファイルおよび論理ファイルの参照 (29 桁目)

この桁は、物理ファイルへの参照を指定する場合にのみ使用します。

論理ファイルの場合は、この桁を空白のままにしておいてください。すべての論理ファイルには、指定されているすべてのフィールドに対する参照機能が自動的に備わっています。論理ファイル内で明示指定されていない属性は、物理ファイルのレコード様式内の対応するフィールドから取り出されて適用されます。

物理ファイルでは、以前に定義されている名前付きのフィールド (参照フィールド と呼ばれます) の属性を参照したい場合には、この欄に R を指定します。物理ファイルの場合は、REF キーワードまたは REFFLD キーワードを指定します。参照フィールドには、定義中の物理ファイルですでに定義されているフィールドを用いることも、以前に作成したデータベース・ファイルのフィールドを用いることもできます。参照されるフィールド属性としては、フィールドの長さ、データ・タイプ、および小数点以下の桁数のほか、ALIAS、COLHDG、DATFMT、DATSEP、FLTPCN、REFSHIFT、TEXT、TIMFMT、TIMSEP、VARLEN、編集、および妥当性検査の各キーワードがあります。

R を指定しない場合には、このフィールドについてフィールド属性を指定しなければなりません。

注: DATFMT キーワードが参照フィールド上で、\*ISO、\*EUR、\*USA、または \*JIS に指定変更されると、DATSEP キーワードは参照されません。

ファイル・レベルおよびレコード・レベルの場合は、29 桁目は空白でなければなりません。

参照フィールドが定義中のファイル内のフィールドである場合には、参照フィールドの名前として、定義中のフィールドの名前と同じ名前を指定することはできません。この 2 つの名前が同じである場合には、参照フィールドを定義しているファイルの名前を REF キーワードまたは REFFLD キーワードのパラメータ値として指定してください。両者の名前が同じでない場合には、REFFLD キーワードで参照フィールドの名前を指定します。

参照フィールドの属性の一部を指定変更したい場合には、変更したい属性を定義中のフィールドに指定します。さらに、下記の点を考慮してください。

- 定義中のフィールドで編集コード (EDTCDE) または編集語 (EDTWRD) を指定した場合は、編集仕様は参照フィールドから複写されません。
- 定義中のフィールドで CHECK (AB、ME、MF、M10、M10F、M11、M11F、VN、または VNE)、CHKMSGID、COMP、RANGE、または VALUES を指定した場合は、妥当性検査仕様は参照フィールドから複写されません。
- 定義中のフィールドでデータ・タイプ、フィールドの長さ、または小数点以下の桁数を指定した場合は、編集キーワードおよび妥当性検査キーワードは参照フィールドから複写されません。

注: 物理ファイルの作成後は、参照ファイルを削除または変更しても、その物理ファイル内のフィールド記述が影響を受けることはありません。参照ファイルに対する変更を反映させるためには、物理ファイルを削除して再作成する必要があります。

## 関連資料

79 ページの『REF (参照) キーワード - 物理ファイルのみ』

これはファイル・レベル・キーワードで、フィールド記述が検索されるファイルの名前を指定するために使用します。

81 ページの『REFFLD (参照フィールド) キーワード - 物理ファイルのみ』

これはフィールド・レベル・キーワードで、次の 3 つのいずれかの条件のもとでフィールドを参照するために使用します。

## 関連情報

DDS ファイルの REF キーワードおよび REFFLD キーワードの指定方法

## 物理ファイルおよび論理ファイルの長さ (30 から 34 桁目)

この欄は、物理ファイル・フィールドまたは論理ファイル・フィールドの長さを指定するのに使用します。

物理ファイルでは、この欄で名前付きのフィールドの長さを指定します (長さを参照フィールドから参照する場合を除く)。数値タイプのフィールドの場合は数字の桁数を、文字タイプのフィールドの場合は文字数を指定します。

論理ファイルでは、この欄は論理フィールドの長さを指定するのに使用します。長さは、この論理ファイルの基礎となっている物理ファイルに対応するフィールドの長さを指定変更または修正したい場合に限り、指定してください。この欄が空白であれば、定義中のフィールドの長さは、論理ファイルの基礎となっている物理ファイル内の対応するフィールドの長さと同じになります。物理ファイルのフィールドが可変長であり、長さを空白にしたとき、そのフィールドは論理ファイルでも可変長になります。長さを指定した場合、VARLEN キーワードを指定していなければ論理ファイルのそのフィールドは固定長になります。さらに、SST (サブstring) キーワードを使用して、別のフィールドのサブセットである文字stringを指定することにより、論理ファイル・フィールドの長さを制御することができます。

長さを指定する場合には、右寄せで指定しなければなりません。先行ゼロはオプションです。

図 14 に、物理ファイルでのフィールド長の指定の正しい例と正しくない例を示します。

```
|...+....1....+....2....+....3....+....4....+....5
00010A          FIELD1      7
  A
00020A          FIELD2      7
  A
00030A          FIELD3    R  +7
  A
```

注: FIELD1 は、フィールドの長さの正しくない指定例を示しています。FIELD2 および FIELD3 は、フィールドの長さの正しい指定例を示しています。

図 14. 物理ファイルの場合の長さ指定の正しい例と正しくない例

有効な長さ指定は次のとおりです。

データ・タイプ	有効な長さ
文字	1 から 32 766 文字
16 進数	1 から 32 766 バイト
2 進数	1 から 18 桁
2 進文字	1 から 32 766 文字
ゾーン 10 進数	1 から 63 桁
パック 10 進数	1 から 63 桁
浮動小数点 (単精度)	1 から 9 桁



データ・タイプ	有効な長さ
浮動小数点 (倍精度)	1 から 17 桁
日付	6、8、または 10 文字
時刻	8 文字
時刻スタンプ	26 文字

データ・タイプが L (日付)、T (時刻)、または Z (時刻スタンプ) のフィールドの桁数はシステムが判別します。30 から 34 桁目にフィールド長を入力してはなりません。

日付および時刻のフィールドの長さには区切り記号も含まれます。

時刻スタンプには次のような既定の形式があります。

YYYY-MM-DD-hh.mm.ss.uuuuuu

単精度の場合は最大 9 桁、倍精度の場合は最大 17 桁を指定することができます。i5/OS プログラムでサポートされる浮動小数点の精度は、単精度の場合は 7 桁、倍精度の場合は 15 桁です。

記憶域内では、1 つのレコードの中の全フィールドが占める合計バイト数が 32 766 以下でなければなりません。レコード様式の合計の桁数を判別するための規則については、表 1 を参照してください。

システムは、実際に記憶域内で占められるバイト数を次のようにして判別します。

データ・タイプ	記憶域内で占有されるバイト数
文字	文字数
16 進数	バイト数
2 進数	
1 から 4 桁	2 バイト
5 から 9 桁	4 バイト
10 から 18 桁	8 バイト
2 進文字	文字数
ゾーン 10 進数	桁数
パック 10 進数	(桁数/2) + 1 (小数部分は切り捨て)
浮動小数点 (単精度)	4 バイト
浮動小数点 (倍精度)	8 バイト
日付	DATFMT キーワードを指定しない場合は 10 文字。 DATFMT キーワードを指定する場合は 6、8 または 10 文字
時刻	8 文字
時刻スタンプ	26 文字

注: システムでは、データ・タイプがゾーン 10 進数の場合よりパック 10 進数の場合の方が、算術演算が効率よく行われます。

表 1 は、様式の合計長を判別するための規則を説明します。

表 1. 様式の合計長の判別に関する規則

状況	処置
レコード様式に可変長フィールドは含まれていますか？	1. 様式の合計長に、追加の 24 バイトを加算してください。 2. 様式の長さ、可変長フィールド 1 つ当たり追加の 2 バイトを加算してください。
レコード様式にヌル値が許されるフィールドは含まれていますか？	様式内のフィールド数の合計を 8 で除算し、余りがでたら 1 バイト切り上げ、様式の長さに加算してください。

物理ファイルの参照フィールド (29 桁目が R) の長さまたは論理ファイルのフィールドの長さを指定変更するためには、新しい値を指定するか、長さの増減を指定します。長さを増やすためには、+n を指定します (n は増加長)。長さを減らすためには、-n を指定します (n は減少長)。たとえば、+4 は、ある数字フィールドが参照フィールドより 4 桁長くなることを示します。物理ファイルのフィールドの長さを指定変更する方法については、5 ページの『物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)』の例を参照してください。

物理ファイルのレコード様式内の対応するフィールドのデータ・タイプが、小数点以下の桁数が 1 以上の 2 進数である場合には、論理ファイルで長さの指定変更を行うことはできません。また、記述中のフィールドが、対応する物理レコード様式からのいくつかのフィールドを連結したものである場合には、論理ファイルで長さを指定することはできません。物理フィールドの長さの合計は、システムによって計算されます。

30 から 34 桁目に値を指定した場合、ユーザーのプログラムには指定された長さ分のデータが渡されません。しかし、対応する物理ファイルのフィールドの長さは変更されません。このため、データ変換エラーが起こることがあります。ファイルにメンバーを追加しようとする、またはファイルのメンバーを開こうとすると、オペレーティング・システムからマッピング・エラー・メッセージが送られます。次の場合にも、オペレーティング・システムからユーザーのプログラムにマッピング・エラー・メッセージが送られます。

- 論理ファイルからの読み取りを行う場合に、その論理ファイルの中で、対応する物理ファイルでの指定値より長さを減らすことが指定されている場合。
- 論理ファイルへの書き込みを行う場合に、その論理ファイルの中で、対応する物理ファイルでの指定値より長さを増やすことが指定されている場合。

たとえば、物理ファイルのフィールドの長さが 4 文字と定義されているときに、論理ファイル内でのフィールド定義によりその長さを 2 文字に減らした場合には、プログラムは、物理ファイル内の AB という値を読み取ることはできますが、ABCD という値は読み取ることができません。この場合、書き込みは必ず正常に行われます。文字フィールドの場合には、物理ファイルのフィールドでは、データが左寄せされ、右側が空白で埋め込まれます。数字フィールドには、物理ファイルのフィールドでは、データが右寄せされ、左側がゼロで埋め込まれます。

30 から 34 桁目の指定は、フィールド仕様の場合に限り有効です。キー・フィールド、選択/除外フィールド、結合レコード、レコード・レベル、およびファイル・レベルでは、この欄は空白にしなければなりません。

**注:** 高水準言語では、フィールドの長さについて制限がある場合があります。そのような高水準言語で使用するファイルについては、その制約を守らなければなりません。

#### 関連概念

4 ページの『DDS の論理ファイルのレコード様式の指定』

1 つの論理ファイルにレコード様式を 2 つ以上指定する場合には、レコード様式ごとに PFILE キーワードを指定しなければなりません。

#### 関連タスク

3 ページの『DDS の単一樣式論理ファイルおよび複数様式論理ファイル』

単一樣式論理ファイルおよび複数様式論理ファイルの場合には、レコード・レベルで PFILE キーワードを指定しなければなりません。複数様式論理ファイルでは、あるレコード様式は、そのレコード様式について PFILE キーワードで指定したすべての物理ファイルに共通するフィールドだけしか使用することができません。

#### 関連資料

85 ページの『SST (サブストリング) キーワード - 論理ファイルのみ』

これはフィールド・レベルのキーワードで、既存の文字、16 進数、ゾーン・フィールド、または図形のサブセットである文字ストリングを指定するために使用します。

69 ページの『JFLD (結合フィールド) キーワード - 結合論理ファイルのみ』

これは結合レベル・キーワードで、結合元 フィールドと結合先 フィールドの名前を識別するために使用します。これらのフィールドの値は、複数の物理ファイルを 1 つの結合論理ファイルで結合するために用いられます。結合元フィールドおよび結合先フィールドは、どちらも結合フィールドと呼ばれています。

## 物理ファイルおよび論理ファイルのデータ・タイプ (35 桁目)

物理ファイルでは、この欄にはデータベース内でのフィールドのデータ・タイプを指定します。論理ファイルの場合にこの欄にデータ・タイプを指定するのは、この論理ファイルの基礎となっている物理ファイル内の対応するフィールドのデータ・タイプを指定変更または修正したい場合に限りです。

この欄がブランクであれば、定義中のフィールドのデータ・タイプは、この論理ファイルの基礎となる物理ファイル内の対応するフィールドのデータ・タイプと同じものになります。

有効なデータ・タイプ項目は、次のとおりです。

項目	意味
P	パック 10 進数
S	ゾーン 10 進数
B	2 進数
F	浮動小数点数
A	文字
H	16 進数
L	日付
T	時刻
Z	時刻スタンプ
5	2 進文字

注: データ・タイプ J (専用)、E (択一)、O (混用) および G (図形) は、DBCS を使用する DDS データベース・ファイルをサポートします。G (図形) データ・タイプは、UCS-2 または UTF-16 を使用する DDS データベース・ファイルもサポートします。A (文字) データ・タイプも、UTF-8 を使用するデータベース・ファイルをサポートします。

物理ファイルでは、データ・タイプを指定せず、また参照フィールドからの複製もしない場合は、オペレーティング・システムにより次のようなデフォルトのデータ・タイプが割り当てられます。

- 小数点以下の桁数 (36 から 37 桁目) がブランクであれば、A (文字)。
- 小数点以下の桁数 (36 から 37 桁目) に 0 から 63 までの数字が指定されていれば、P (パック 10 進数)。

注:

1. パック 10 進数フィールド、ゾーン 10 進数フィールド、または 2 進数フィールドの場合に、それが整数フィールドであることを示すには 37 桁目に 0 を指定します。

2. 単精度の浮動小数点数フィールドの場合は、35 桁目に F を指定します。倍精度を指定したい場合、または前に指定した浮動小数点数フィールドの精度を変更したい場合は、FLTPCN キーワードを使用します。
3. フィールドの内容をシステムに解釈させる必要がないことを指示するには、35 桁目に H (16 進数) を指定します。ほとんどの場合、16 進数フィールドは、16 進数フィールドの内容が文字セットまたはコード・ページに変換されない点を除き、文字フィールドとして扱われます。

次の表は、物理ファイルと論理ファイルのフィールドのデータ・タイプ間で有効なデータ変換のタイプを示します。表内で、有効な変換は X または表の注の参照番号で示してあります。

物理ファイル・ フィールドのデ ータ・タイプ	論理ファイル・フィールドのデータ・タイプ																
	A	H	S	P	B	F	L	T	Z	UTF8	UTF16	UCS2	O	J	E	G	5
文字 (A)	X	X	1							X	X	X	X		X		X
16 進数 (H)	X	X	1										X	X	X		X
ゾーン (S)	1	1	X	X	2	X											1
パック (P)			X	X	2	X											
2 進数 (B)			2	2	3	2											
浮動小数点 (F)			X	X	2	X											
日付 (L)	6,7		6	6			X										
時刻 (T)			4					X									
タイム・スタ ンプ (Z)							5	5	X								
UTF-8	X									X	X	X	X				X
UTF-16	X									X	X	X	X				X
UCS-2	X									X	X	X	X				X
混用 (O)		X								X	X	X	X				X
専用 (J)		X											X	X	X	X	X
択一 (E)		X											X		X		X
図形 (G)										X	X	X	X	X	X	X	
2 進文字	X	X	1										X	X	X		X

**注:**

1. 文字数 (またはバイト数) と数字の桁数が等しく、かつ文字 (または 16 進数) フィールドが可変長フィールドとして定義されていない場合にだけ有効。
2. 2 進数フィールドの小数点以下の桁数がゼロの場合にだけ有効。
3. 両方のフィールドの小数点以下の桁数が同じ場合にだけ有効。
4. システムによってフィールド長が生成されるため、30 から 34 桁目に長さを入力しないでください。この長さには区切り記号は含まれていません。
5. フィールドが入力専用である場合にだけ有効。
6. 論理ファイル・フィールドの場合、これらのデータ・タイプにフィールド長 (30 から 34 桁) を指定することができます。長さを指定しない場合には、システムがデフォルトの長さを生成します。これらのデータ・タイプに有効な長さについては、DATFMT キーワードとともに説明されています。
7. DBCS フィールド・タイプを DATE フィールドにマップすることはできません。

**関連タスク**

3 ページの『DDS の単一样式論理ファイルおよび複数様式論理ファイル』

単一样式論理ファイルおよび複数様式論理ファイルの場合には、レコード・レベルで PFILE キーワードを指定しなければなりません。複数様式論理ファイルでは、あるレコード様式は、そのレコード様式について PFILE キーワードで指定したすべての物理ファイルに共通するフィールドだけしか使用することができません。

#### 関連資料

100 ページの『データ・タイプ (35 桁目)』

4 つの DBCS データ・タイプ、すなわち J (専用)、E (択一)、O (混用) および G (図形) のいずれでも使用することができます。

98 ページの『データ・タイプ (35 桁目)』

Unicode データとして有効なデータ・タイプは G (グラフィック) データ・タイプと A (文字) タイプです。

5 ページの『物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)』

DDS 形式の最初の 44 桁は定位置項目と呼ばれます。

69 ページの『JFLD (結合フィールド) キーワード - 結合論理ファイルのみ』

これは結合レベル・キーワードで、結合元 フィールドと結合先 フィールドの名前を識別するために使用します。これらのフィールドの値は、複数の物理ファイルを 1 つの結合論理ファイルで結合するために用いられます。結合元フィールドおよび結合先フィールドは、どちらも結合フィールドと呼ばれています。

#### DDS ファイルでの数値データ・タイプの変換:

物理ファイルのレコード様式からのデータ・タイプの変換は、数値タイプの範囲内であれば許されます。たとえば、物理ファイルの 2 進数フィールドを、論理ファイルではゾーン 10 進数フィールドに変換することができます。

#### DDS ファイル内のゾーン 10 進数と文字または 16 進数との間の変換:

ゾーン 10 進数フィールドと、文字フィールドまたは 16 進数フィールドとは、長さが同じであれば、一方から他方へ変換することができます。

プログラム内での該当フィールドのデータ・タイプは、論理ファイルで指定されたデータ・タイプです。受け渡しされるデータに数字 (0 から 9) だけが含まれている場合には、入出力操作時にエラーが起こることはありません。しかし、プログラムは、0 から 9 以外の文字を文字フィールドまたは 16 進数フィールドからゾーン 10 進数フィールドに渡す入出力操作を行うことはできません。この場合には、オペレーティング・システムからメッセージが送られ、該当の入出力操作は完了できません。

たとえば、フィールドが物理ファイル内のゾーン 10 進数であるとし、プログラムに渡すデータのタイプとして文字タイプ (A) を指定する場合には、そのフィールドが論理ファイルを介して物理ファイルに戻される時点で、フィールドに含まれているのが必ず数字 (0 から 9) だけになるように、十分注意しなければなりません。

もう 1 つの例として、フィールドが物理ファイル内の文字フィールドである場合を考えてみましょう。そのフィールドをゾーン 10 進数フィールドとして指定し、論理ファイル内でキー・フィールドとして指定した場合には、該当の物理ファイルの全レコードに数字 (0 から 9) だけが含まれていなければ、論理ファイルを作成することはできません。

#### DDS 内のフィールドの浮動小数点からパック 10 進数、ゾーン 10 進数、または 2 進数のフィールドへの変換:

物理ファイルの浮動小数点数フィールドを論理ファイルのパック 10 進数、ゾーン 10 進数、または 2 進数のフィールドに変換したい場合には、長さおよび小数点以下の桁数を明示指定しなければなりません。浮動小数点数データを固定小数点フォーマットに変換する場合には、長さおよび小数点以下の桁数として指定する値が、そのデータを収容できるだけの大きさを備えているかどうかを確認してください。

物理ファイルの長さおよび小数点以下の桁数は、単なる表示値であり、実際の数値の大きさを表すものではありません。

#### DDS 内のフィールドを連結するときのデータ・タイプの変換:

定義中のフィールドが、関連する物理ファイルからのフィールド (CONCAT キーワードで指定したもの) を連結したものである場合には、データ・タイプを指定することはできません。オペレーティング・システムは、連結しようとするフィールドのデータ・タイプに基づいてデータ・タイプを割り当てます。

その場合の一般的な規則は次のとおりです。

- 連結の対象に 1 つまたは複数の 16 進数 (H) フィールドが含まれている場合は、結果のデータ・タイプは 16 進数 (H) です。
- 連結の対象に 1 つまたは複数の文字 (A) フィールドが含まれており、16 進数フィールドが含まれていない場合は、結果のデータ・タイプは文字 (A) です。
- 連結の対象が数値 (S、P、B) フィールドだけである場合は、結果のデータ・タイプはゾーン 10 進数 (S) です。
- 連結に UTF-8 フィールドが含まれている場合、結果は UTF-8 になります。
- 連結に UCS-2 または UTF-16 フィールドが含まれている場合は、リストに UTF-16 があれば結果は UTF-16 になり、UTF-16 がなければ、UCS-2 になります。
- 連結に 2 進文字フィールドが含まれている場合、結果は 2 進文字になります。

#### DDS 内のフィールドをサブストリング化するときのデータ・タイプの変換:

定義中のフィールドが、論理ファイルまたは関連の物理ファイルのフィールドのサブストリング (SST キーワードで指定済み) である場合には、オリジナルのフィールドは文字、16 進数、ゾーン、DBCS グラフィック、または 2 進文字 (A、H、S、または G) でなければなりません。

論理ファイル・フィールドのデータ・タイプを指定しない場合、次の表に示すように変換されます (ソース・フィールド・タイプは、物理ファイル・フィールドまたは前に論理ファイル・ソースで定義された論理ファイル・フィールドのタイプです)。

ソース・フィールド・タイプ	結果の論理ファイル・フィールド
A	A
H	H
S	A
G	G
2 進文字	2 進文字

#### 物理ファイルおよび論理ファイルの小数点以下桁数 (36 および 37 桁目)

この欄には、パック 10 進数フィールド、ゾーン 10 進数フィールド、2 進数フィールド、または浮動小数点数フィールドにおける小数点の位置を指定します。

定義中のフィールドについて CONCAT キーワードを指定する場合には、ここで小数点以下の桁数を指定することはできません。物理ファイル内で小数点以下の桁数を持つフィールドは、連結フィールドに組み込むことはできません。

注: 高水準言語では、小数点以下の桁数について、桁数および値についての特殊な制限がある場合があります。そのような高水準言語で使用するファイルについては、その制約を守らなければなりません。

## 物理ファイルでの小数点以下の桁数の使用

物理ファイルでは、この欄には、パック 10 進数フィールド、ゾーン 10 進数フィールド、2 進数フィールド、または浮動小数点数フィールドにおける小数点の位置を指定します。つまり、小数点の右側の小数桁数を、0 から 63 までの 10 進数で指定します。(この数は、フィールド桁数として指定した桁数を超えるものであってはなりません。) 物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目) のトピックは、小数点以下の桁数フィールドをどのようにコーディングするかを示します。フィールド桁数が 2 進数フィールドの 9 よりも大きい場合、小数点以下の桁数の値は 0 でなければなりません。

データは、実際には小数点なしでシステムに保管され、単に暗黙の小数点が存在するだけです。たとえば、1.23 という値は 123 と保管されます。したがって、編集を指定していなければ、表示装置ファイルまたは印刷装置ファイルには 123 の値がそのまま現れます。

参照フィールド (29 桁目が R) の桁数を変更するためには、新しい値を指定するか、または桁数の増減を指定します。桁数を増やすためには、 $+n$  ( $n$  は増加桁数) を指定します。桁数を減らすためには、 $-n$  ( $n$  は減少桁数) を指定します。たとえば、+4 は、小数点以下の桁数を参照フィールド内の桁数より 4 桁分多くすることを示します。小数点以下の桁数が許容最大数を超過していると、エラー・メッセージが送られます。

## 論理ファイルでの小数点以下の桁数の使用

論理ファイルで小数点以下の桁数を指定するのは、その論理ファイルの基礎になっている物理ファイルの対応するフィールドの小数点以下の桁数を指定変更または修正したい場合に限りです。この欄を空白にすれば、定義中のフィールドは、この論理ファイルの基礎となる物理ファイルの対応するフィールドの小数点以下の桁数と同じになります。

パック 10 進数フィールドまたはゾーン 10 進数フィールドの小数点の位置を指定変更または修正するには、小数点の右側の桁数 (つまり小数点以下の桁数) を示す 0 から 63 までの範囲内の値を指定します。ここに指定する値は、フィールド長として指定した値を超えてはなりません。物理ファイル内の対応するフィールドが 2 進数 (データ・タイプ B) で、その小数点以下の桁数が 1 桁以上である場合には、小数点以下の桁数を指定変更または修正することはできません。論理ファイルのフィールドが 2 進数で (論理ファイルの 35 桁目に B が指定されていて)、対応する物理ファイル内のフィールドが 2 進数でない場合には、その 2 進数フィールドについては、小数点以下の桁数はゼロでなければなりません。

新しい値を指定するか、あるいは桁数の増減を指定して、フィールドの桁数を指定変更することができます。桁数を増やすためには、 $+n$  ( $n$  は増加桁数) を指定します。桁数を減らすためには、 $-n$  ( $n$  は減少桁数) を指定します。たとえば、+4 は、小数点以下の桁数を参照フィールド内の桁数より 4 桁分多くすることを示します。

36 から 37 桁目に値を指定し、プログラムが論理ファイルのフィールドを介して物理ファイルのフィールドに対するデータの読み書きを行う場合には、オペレーティング・システムによりデータの小数点での位置合わせが行われます。このために、場合によっては小数部分の値の切り捨てまたはデータ変換エラーが起きることがあります。小数部分の値の切り捨てが起こるのは、次のような場合です。

- 論理ファイルからの読み取りを行う場合であって、その論理ファイルの中で、対応する物理ファイルの指定値よりも小数点以下の桁数を減らすことが指定されている場合。

- 論理ファイルへの書き込みを行う場合であって、その論理ファイルの中で、対応する物理ファイルの指定値よりも小数点以下の桁数を増やすことが指定されている場合。

たとえば、物理ファイルのフィールドの長さが 4 桁で、小数点以下の桁数が 2 桁として定義されており、一方、論理ファイルにおいて対応するフィールドの小数点以下の桁数が 0 桁に減らされたとすれば、物理ファイル内の 0.20 という値は、論理ファイルでは 0 になります。また、物理ファイル内の 2.52 という値は、論理ファイルでは 2 になります。

小数部分の値が切り捨てられた場合には、フィールドの左側がゼロで埋められます。

データ変換エラーは次のような場合に起こることがあります。

- 論理ファイルへの書き込みを行う場合であって、その論理ファイルの中で、対応する物理ファイルの指定値よりも小数点以下の桁数を減らすことが指定されている場合。
- 論理ファイルからの読み取りを行う場合であって、その論理ファイルの中で、対応する物理ファイルの指定値よりも小数点以下の桁数を増やすことが指定されている場合。

データ変換エラーは、小数点の左側に、使用できる桁数よりも多い桁数を転送しようとするとき起こります。たとえば、上記の例と同様に物理ファイルのフィールドが 4 桁で、そのうち小数点以下の桁数が 2 桁であり、一方、論理ファイルのフィールドについて指定した小数点以下の桁数が 0 であったとします。この場合、3322 という値は、論理ファイルには書き込むことができますが、物理ファイルには収容することができません。この値が物理ファイルでは収まらないのは、小数点の左には 2 桁しか許されないからです。

データ変換エラーが生じないようにするためには、論理ファイルのフィールドの長さ (30 から 34 桁目) を、小数点以下の桁数の増減分だけ増加または減少させます。

#### 関連資料

5 ページの『物理ファイルおよび論理ファイルの定位置項目 (1 から 44 桁目)』  
DDS 形式の最初の 44 桁は定位置項目と呼ばれます。

69 ページの『JFLD (結合フィールド) キーワード - 結合論理ファイルのみ』  
これは結合レベル・キーワードで、結合元 フィールドと結合先 フィールドの名前を識別するために使用します。これらのフィールドの値は、複数の物理ファイルを 1 つの結合論理ファイルで結合するために用いられます。結合元フィールドおよび結合先フィールドは、どちらも結合フィールドと呼ばれています。

### 物理ファイルおよび論理ファイルの使用目的 (38 桁目)

この欄は、名前を付けたフィールドを、入力専用にするか、入出力共用フィールドにするか、あるいは非入出力フィールドにするかを指定するのに使用します。

物理ファイルの場合に指定できる項目は次のとおりです。

項目 意味

ブランク

デフォルトにより B (入力および出力の両方が可能)

**B** 入力と出力の両方が可能

デフォルト値が唯一の有効な値と同じであるため、この欄への入力はありません。

38 桁目の項目は REF または REFFLD キーワードにより参照されません。したがって、物理ファイルのフィールドについて 38 桁目に B を指定しても、表示装置ファイルでそのフィールドが参照されるさいには、その指定は効力はありません。



論理ファイルに有効な項目は、次に説明されているとおりです。

### ブランク (デフォルト)

38 桁目をブランクにすると、次のデフォルト値が使用されます。

- 単一样式論理ファイルおよび複数様式論理ファイル (PFILE がレコード・レベルで指定されたもの) の場合には、当該フィールドは入出力共用 (B) フィールドになります。
- 結合論理ファイル (JFILE がレコード・レベルで指定されたもの) の場合には、当該フィールドは入力専用 (I) フィールドになります。

### B (入出力共用)

38 桁目に B を指定すると、当該フィールドは入出力共用 フィールドとなり、入力操作および出力操作の両方に使用することができます。つまり、プログラムでは、そのフィールドからデータを読み取ることも、そのフィールドにデータを書き込むこともできます。結合論理ファイルは読み取り専用ファイルであるため、結合論理ファイルに対して入出力共用フィールドを指定することはできません。

### I (入力専用)

38 桁目に I を指定すると、そのフィールドは入力専用 となり、入力操作だけに使用できるフィールドになります。つまり、プログラムでは、そのフィールドからデータを読み取ることはできますが、フィールド内のデータを変更することはできません。入力専用フィールドの代表的な例としては、アクセス・パスの保守を軽減するために使用されるキー・フィールド、ユーザーが見ることはできるが変更はできない重要フィールド (従業員レコードの給与フィールドなど)、および SST キーワードまたは TRNTBL キーワードが指定されているフィールドなどが挙げられます。

入力専用フィールドを指定したレコード様式に対してプログラムで変更を行った場合には、入力専用フィールドは更新されず、メッセージも出されません。入力専用フィールドを指定したレコード様式に対してプログラムで出力操作を行った場合には、入力専用フィールドにデフォルト値 (DFT (デフォルト) を参照) が入ります。

入力専用フィールドは、物理ファイル内に指定することはできません。

### N (非入出力)

38 桁目に N を指定した場合には、そのフィールドが非入出力 フィールド (入力用でも出力用でもないフィールド) となります。このフィールドは、結合論理ファイルの場合にだけ有効です。非入出力フィールドは、結合論理ファイル内の結合フィールドとして使用することができますが、非入出力フィールドをプログラムで読み取ることはできません。

非入出力フィールドは、物理ファイル内の結合されるフィールドの属性が合致しない場合に使用します。この場合には、一方または両方の結合フィールドを再定義しなければなりません。しかし、再定義したフィールドをレコード様式に入れる必要がない (すなわち、アプリケーション・プログラムで再定義したフィールドを見る必要がない) 場合もあります。このような場合に、再定義する結合フィールドについてこの欄に N を指定することによって、それらの結合フィールドがレコード様式に入らないようにすることができます。

38 桁目に N を指定したフィールドは、プログラムにより使用されるバッファには入りません。ただし、ファイル・フィールド記述表示 (DSPFFD) コマンドを用いれば、フィールド記述は表示されず。

非入出力フィールドは、選択/除外フィールドまたはキー・フィールドとして使用することはできません。

38 桁目の項目は、REF キーワードまたは REFFLD キーワードを使用して行う参照の対象にはなりません。したがって、論理ファイル内のフィールドについて 38 桁目に B または I を指定しても、そのフィールドが表示装置ファイルで参照されるさいには、その指定は効力はありません。

## 関連資料

55 ページの『DFT (デフォルト) キーワード - 物理ファイルのみ』

これは、フィールド・レベル・キーワードで、フィールドのデフォルト値を指定するために使用します。

69 ページの『JFLD (結合フィールド) キーワード - 結合論理ファイルのみ』

これは結合レベル・キーワードで、結合元 フィールドと結合先 フィールドの名前を識別するために使用します。これらのフィールドの値は、複数の物理ファイルを 1 つの結合論理ファイルで結合するために用いられます。結合元フィールドおよび結合先フィールドは、どちらも結合フィールドと呼ばれています。

## 物理ファイルおよび論理ファイルの位置 (39 から 44 桁目)

この欄は物理ファイルまたは論理ファイルでは使用しません。注記行のテキストを記入するために使用する場合を除き、この欄はブランクのままにしてください。

---

## 物理ファイルおよび論理ファイルのキーワード項目 (45 から 80 桁目)

この項では、物理ファイルおよび論理ファイルの記述を行う場合に使用できるキーワード項目について説明します。キーワードは、45 から 80 桁目 (機能欄) に記入します。

ほとんどのキーワードは、物理ファイルと論理ファイルの両方で使用できます。ただし、物理ファイルでのみ使用できるキーワード、および、論理ファイルでのみ使用できるキーワードがあります。そのような場合、その制限は、キーワードの表題および本文の両方に注記があります。

### 具体的な制約事項

単一様式論理ファイルおよび複数様式論理ファイルにのみ有効なキーワードは次のとおりです。

- PFILE
- REFACCPH

結合論理ファイルにのみ有効なキーワードは次のとおりです。

- JDFTVAL
- JDUPSEQ
- JFILE
- JFLD
- JOIN
- JREF

DDS を使用してソース・ファイル (通常、DDS ではなく CRTSRCPF コマンドで作成される) を記述するとき、または論理ファイルが、ソース・ファイルとして使用される物理ファイルに基づいている場合には、下記のキーワードは使用できません。

ABSVAL	NOALTSEQ
ALTSEQ	SIGNED
DESCEND	UNIQUE
FCFO	VARLEN
FIFO	ZONE
LIFO	

## 関連資料

101 ページの『DBCS を使用するデータベース・ファイルのキーワードに関する考慮事項』  
DBCS データを含むフィールドについては、数値データに使用するための DDS キーワード は指定し  
ないでください。システムは DBCS データを文字データと同じに取り扱うので、DBCS データに対し  
て算術演算を行うことはできません。

## 関連情報

DDS キーワードおよびパラメーター値の規則

## 物理ファイルおよび論理ファイルの ABSVAL (絶対値) キーワード

これはキー・フィールド・レベル・キーワードで、この数字フィールドに関連する値の順序付けを行うさい  
に、フィールドの符号を無視するようオペレーティング・システムに指示するために使用します。このキー  
ワードにはパラメーターはありません。

次の例は、いずれもゾーン 10 進数キー・フィールドのある 6 つのレコードを示しています。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

順序付けキーワードまたは ALTSEQ キーワードを指定していない場合には、デフォルトによりそのキー・  
フィールドの属性は SIGNED になります。この場合には、レコードの順序付けは次のようになります。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
3	98-	F9D8
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8
6	99	F9F9

ABSVAL キーワードを指定した場合には、負のフィールドの絶対値が使用され、レコードの順序は次のよ  
うになります。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
2	00	F0D0
5	20	F2F0
4	97	F9F7
1	98	F9F8
3	98-	F9D8
6	99	F9F9

ABSVAL キーワードは、文字、日付、時刻、時刻スタンプ、および 16 進数データ・タイプのフィールド  
に対しては無効です。このキーワードを DIGIT、SIGNED、UNSIGNED、または ZONE キーワードととも  
に使用することはできません。

ABSVAL (キー・フィールド・レベルのキーワード) を使用すると、ALTSEQ (ファイル・レベル・キーワード) は無視されます。キー・フィールドに ABSVAL を指定すると、ファイル・レベルで ALTSEQ が指定されていても、そのキー・フィールドについては NOALTSEQ が効力を持つことになります。これは、NOALTSEQ キーワードが指定されていても起こります。

## 例

次の例は、ABSVAL キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          ORDAMT          5 0
00020A          K ORDAMT          ABSVAL
A
```

## 物理ファイルおよび論理ファイルの ALIAS (別名) キーワード

これはフィールド・レベル・キーワードで、フィールドの別名 (代替名) を指定するために使用します。プログラムのコンパイルの時点で、DDS のフィールド名の代わりに、別名がプログラムに組み込まれます。

その別名が使用されるかどうかは、使用する高水準言語コンパイラーによって決まります。該当言語の ALIAS のサポートについては、その高水準言語の解説書を参照してください。

キーワードの形式は次のとおりです。

ALIAS (alternative-name)

別名は、他のすべての別名、または該当レコード様式内の DDS フィールド名と同じものであってはなりません。重複する別名が見つかった場合には、そのフィールド名または別名についてのエラー・メッセージが出されます。

別名は、DDS 内または他の i5/OS 機能で (たとえば、キー・フィールド名として、REFFLD キーワードに指定するフィールド名として、あるいはファイル・コピー (CPYF) コマンドで使用するフィールド名として) 使用することはできません。

ALIAS キーワードの指定されたフィールドを参照した場合、参照元のフィールドにこのキーワードが明示指定されていないと、ALIAS キーワードが参照元フィールドに複写されます。

## 例

次の例は、ALIAS キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          FIELDA          25A          ALIAS (CUSTOMERNAME)
A
```

この例では、FIELDA の別名は CUSTOMERNAME です。

### 関連情報

DDS キーワードおよびパラメーター値の規則

## ALL (すべて) キーワード - 論理ファイルのみ

これは選択/除外フィールド・レベル・キーワードで、当該論理ファイルについての他のすべての選択/除外仕様の処理が終わったあとの処置を指定するために使用します。

17 桁目に S を指定し、機能欄に ALL を指定すると、他の選択/除外規則に該当しないレコードはすべて選択するように、オペレーティング・システムに指示することになります。17 桁目に O を指定し、機能

欄に ALL を指定すると、他の選択/除外規則に該当しないレコードはすべて除外するように、オペレーティング・システムに指示することになります。ALL を指定する場合には、他の選択/除外キーワードの後に指定しなければなりません。ALL キーワードとともにフィールド名を指定することはできません。

このキーワードにはパラメーターはありません。

ALL キーワードを指定しなかった場合には、当該ファイルに対して行った最後の選択/除外仕様とは逆の処置が、デフォルトの処置として取られます。つまり、最後の指定が選択であれば、デフォルトの処置は「すべて除外」となります。最後の指定が除外であれば、デフォルトの処置は「すべて選択」となります。

## 例

次の例は、ALL キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      S ACT                      COMP(EQ 3000)
00020A      S ACT                      COMP(GT 3100)
00030A      O AMT                      COMP(LT 0)
00040A      0                          ALL
      A
```

### 関連資料

22 ページの『選択/除外フィールド名』

選択/除外フィールドは、ユーザー・プログラムがこのレコード様式を用いてレコードを検索するさいにどのようにレコードを選択または除外するかを、オペレーティング・システムに指示するために使用します。選択/除外の対象となるレコードは、このレコード様式についての PFILE キーワードまたは JFILE キーワードで指定された物理ファイルのレコードだけです。

## 物理ファイルおよび論理ファイルの ALTSEQ (代替照合順序) キーワード

これはファイル・レベル・キーワードで、このファイルについてキーが指定されている場合に、代替照合順序テーブルを用いてファイル・メンバーのレコードの検索順序を決めるよう、オペレーティング・システムに指示するために使用します。

キーワードの形式は次のとおりです。

```
ALTSEQ([library-name/]table-name)
```

代替照合順序テーブルの名前は必須パラメーター値であり、library-name はオプションです。ライブラリー名を指定しなかった場合、i5/OS オペレーティング・システムは、ファイル作成時にライブラリー・リスト (\*LIBL) を使用します。

次の場合には、ALTSEQ キーワードは無効です。

- 物理ファイル作成 (CRTPF) または論理ファイル作成 (CRTLF) コマンドで FILETYPE(\*SRC) を指定した場合。
- キー・フィールドのデータ・タイプが、パック 10 進数、2 進数、または浮動小数点数である場合。
- キー・フィールドに対して ABSVAL または SIGNED が指定されている場合。これらのフィールドでは、NOALTSEQ (キー・フィールド・レベル・キーワード) が指定されているものと見なされます (改めて指定する必要はありません)。NOALTSEQ は複合キーを構成するフィールドのうち代替順序を必要としないフィールドに指定することができます。
- 物理ファイル作成 (CRTPF) または論理ファイル作成 (CRTLF) コマンドで、SRTSEQ パラメーターに \*SRC 以外の値を指定する場合。

ALTSEQ キーワードを、REFACCPH キーワードと一緒に指定することはできません。

代替照合順序テーブルの使用権限を持っていないければなりません。代替照合順序テーブルはテーブル作成 (CRTTBL) コマンドを用いて作成されます。

ALTSEQ を指定すると、ゾーン・キー・フィールドはデフォルトにより符号なしでの順序になります。個々のキー・フィールドに SIGNED キーワードを指定すれば、このデフォルト値を指定変更することができます。

## 例

次の例は、論理ファイルの場合の ALTSEQ キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                ALTSEQ(TABLELIB/TABLE1)
00020A      R RECORD1                  PF1 (PF1)
00030A      :
00040A      :
00050A      :
00060A      NAME                        20
00070A      :
00080A      :
00090A      K NAME
      A
```

この例では、RECORD1 の様式を備えたレコードが、代替照合順序テーブル (ライブラリー TABLELIB 内の TABLE1) に従って、NAME というキーに基づいて順序付けされます。

## ALWNULL (ヌル値の許容) キーワード - 物理ファイルのみ

これはフィールド・レベル・キーワードで、当該フィールドにヌル値が許容されることを定義するために使用します。

このキーワードにはパラメーターはありません。

ALWNULL キーワードを指定するとき、30 から 34 桁目で指定できる最大長は 32 765 (フィールドが可変長でもあるなら 32 739) バイトです。

物理ファイルの場合に、\*JOB、\*MDY、\*DMY、\*YMD または \*JUL の値を持つ DATFMT キーワードを指定し、フィールドがヌル値を許容するときには、このフィールドについての DFT キーワードに有効な日付を指定しなければなりません。

## 例

次の例は、ALWNULL キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1
00020A      FIELD1      75A      ALWNULL
00030A      FIELD2      100A
00040A      FIELD3      L      ALWNULL
00050A                                DATFMT(*MDY)
00060A                                DFT('12/25/93')
      A
```

この例では、FIELD1 はヌル値を許容するフィールドとして定義されます。FIELD1 のデフォルト値はヌル値です。FIELD2 はヌル値を許容しないフィールドとして定義されます。FIELD2 のデフォルト値はブランクです。

## 物理ファイルおよび論理ファイルの CCSID (コード化文字セット識別コード) キーワード

これはファイル・レベルまたはフィールド・レベル・キーワードで、物理ファイルにはこれを使用し、論理ファイルにはフィールド・レベル・キーワードを使用して、文字フィールドについてのコード化文字セット ID (CCSID) を指定します。

キーワードの形式は次のとおりです。

```
CCSID(value [field-display-length | *MIN | *LEN display-positions]
      [*CONVERT | *NOCONVERT] [*NORMALIZE])
```

この値は最大 5 桁までの数値です。この値は、エンコード・スキーム識別コード、文字セット識別コード、コード・ページ識別コードと、当該フィールドのデータに使用されるコード化図形文字表記を個別に識別するその他の関連情報を 1 つの組にして示します。

論理ファイルの場合、次に示す特性が満たされなければ CCSID キーワードを論理ファイル・フィールドで使用することはできません。

- 論理ファイル CCSID キーワードに指定された値が Unicode エンコード・スキームを使用している場合は、UCS-2 レベル 1 または UTF-16 エンコード・スキームでは、フィールド・データ・タイプが G、UTF-8 エンコード・スキームでは、フィールド・データ・タイプが A でなければなりません。また、対応する物理ファイル・フィールドのタイプは A、G、または O でなければなりません。
- 論理ファイル CCSID キーワードに指定された値が Unicode エンコード・スキームを使用しない場合は、フィールド・データ・タイプは A、O、または G でなければなりません。さらに、対応する物理ファイル・フィールドを G タイプ・フィールドにして、CCSID キーワードに UCS-2 または UTF-16 値を指定するか、A (文字) タイプ・フィールドにして、UTF-8 CCSID を指定する必要があります。

**field-display-length** パラメーターはオプションで、当該フィールドが表示装置ファイル内のフィールドによって参照されるときにだけ使用されます。このパラメーターは、**value** パラメーターが UCS-2 または UTF-16 であるときのみ有効です。**field-display-length** パラメーターを利用することによって、UCS-2 または UTF-16 データに応じて、フィールド・サイズを制御できます。

**field-display-length** の代わりに、特殊値 **\*MIN** を指定することができます。この特殊値は、表示装置ファイルの DDS レコード様式の参照フィールドによって使用される場合のみ、物理ファイルで定義できます。この値は、表示位置について定義されたフィールド表示桁数の指定に使用されます。この値を指定すると、画面上のフィールド桁数が、DDS で定義されたフィールド桁数と等しくなります。

**field-display-length** の代わりに、**display-positions** 値と一緒に特殊値 **\*LEN** を指定することもできます。この特殊値は、表示装置ファイルの DDS レコード様式の参照フィールドによって使用される場合のみ、物理ファイルで定義できます。この値は、表示位置について定義されたフィールド表示桁数の指定に使用されません。この値を指定すると、画面上のフィールド桁数が、**display-positions** 値と等しくなります。

**\*CONVERT** パラメーターはオプションです。このパラメーターは、印刷装置ファイルの DDS レコード様式の参照フィールドによって使用される場合のみ、物理ファイルで定義できます。このパラメーターは、フィールドを印刷する際に、UCS-2 または UTF-16 データを、CRTPRTF、CHGPRTF または OVRPRTF コマンドの **CHRID** コマンド・パラメーターで指定されたターゲット CCSID に変換するように指定します。このパラメーターを指定しないと、キーワードはデフォルトにより **\*CONVERT** に設定されます。

**\*NOCONVERT** を指定すると、UCS-2 または UTF-16 データはターゲット CCSID に変換されません。

**\*NORMALIZE** パラメーターはオプションですが、UTF-8 および UTF-16 データを使用しているときの結果が予測しやすくなります。このパラメーターを使用すると、UTF-8 データと UTF-16 データ内の文字

を結合できます。この文字の結合サポートによって、結果が複数の文字で構成されるようにできます。データ・ストリングの先頭文字の後に、スペースなしで最大 300 個の異なるアクセント付き文字 (ウムラウトやアクセントなど) を続けることができます。結果の文字がすでに文字セット内に定義されているものである場合は、正規化によって、結合文字のストリングが定義済み文字の 16 進数値に置き換えられます。結果の文字が定義済み文字でない場合は、結合文字ストリングは正規化が行われても変更されません。たとえば、後に揚音文字 (X'0301') が続く UTF-16 漢字ストリング 'e' (X'0065') の正規化を行うと、文字 é (X'00E9') に置換されます。

\*NORMALIZE パラメーターを使用できるのは、フィールド・レベルで CCSID キーワードが使用されている場合のみです。このキーワードを指定しないと、システムは、UTF-8 および UTF-16 フィールドに挿入したデータ、またはこれらのフィールドを更新したデータはすでに正規化されているものと想定します。

\*NORMALIZE は、CCSID キーワードの UTF-16 値 (グラフィックス・フィールド) または UTF-8 値 (文字フィールド) でのみ有効です。

物理ファイルのファイル・レベルで指定した場合、CCSID キーワードは当該ファイル内の各文字フィールドに適用されます。ただし、すでに CCSID キーワードが指定してある文字フィールドには適用されません。ファイル・レベルの CCSID が UCS-2 または UTF-16 である場合、それは CCSID キーワードを持たない任意の G フィールドに適用されます。物理ファイル・フィールドの CCSID 値が UCS-2 エンコード・スキームを使用している場合には、このフィールドのデータ・タイプはタイプ G でなければなりません。また、物理ファイル・フィールドの CCSID 値が UTF-8 エンコード・スキームを使用している場合には、このフィールドのデータ・タイプは文字でなければなりません。物理ファイル・フィールドの CCSID 値が UTF-16 エンコード・スキームを使用している場合には、このフィールドのデータ・タイプはタイプ G でなければなりません。

CCSID キーワードがファイル・レベルで指定されておらず、しかも文字フィールドすべてに CCSID キーワードが指定されているわけではない場合、それらのフィールドには、ファイル作成時にジョブのデフォルトの CCSID が割り当てられます。

## 例

次の例は、物理ファイルでの CCSID キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                CCSID(285)
00020A      R RECORD1
00030A      FIELD1      75G      CCSID(13488)
00040A      FIELD2      150A
00050A      FIELD3      20A
00060A      FIELD4      10A      CCSID(1208 *NORMALIZE)
00070A      FIELD5      10G      CCSID(1200)
A
```

FIELD1 には 13488 という UCS-2-ccsid の値が割り当てられます。FIELD2 および FIELD3 には 285 という CCSID の値が割り当てられます。FIELD4 には、UTF-8 CCSID 値 1208 が割り当てられ、そのデータは、ファイル内に挿入またはファイルを更新する前に正規化されます。FIELD5 には、UTF-16 CCSID 値として 1200 が割り当てられ、そのデータは、ファイル内に挿入またはファイルを更新する前には正規化されません。

次の例は、対応する論理ファイルでの CCSID キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00000A
00010A      R RECORD1
00020A      FIELD1      75A      CCSID(37)
00030A      FIELD2      150G     CCSID(13488 80)
```



00040A	FIELD3	20A	
00050A	FIELD4	10G	CCSID(1200 *NORMALIZE)
00060A	FIELD5	10A	
A			

論理ファイルの FIELD1 には 37 という SBCS CCSID の値が割り当てられます。物理ファイル・フィールドに UCS-2 データが入っているため、FIELD1 の物理ファイルと論理ファイルとの間で変換が行われます。論理ファイルの FIELD2 には 13488 という UCS-2-ccsid の値が割り当てられます。論理ファイルに UCS-2 データが入っているため、FIELD2 の物理ファイルと論理ファイルとの間で変換が行われます。FIELD3 には CCSID は指定されません。FIELD4 には、UTF-16 CCSID 値 1200 が割り当てられます。物理ファイル・フィールドに UTF-8 文字データが入っているため、物理ファイルと FIELD4 の論理ファイルの間で変換が行われます。データは正規化されます。FIELD5 には、ファイルが作成されたジョブの CCSID が割り当てられます。物理ファイル・フィールドに UTF-16 文字データが入っているため、物理ファイルと FIELD5 の論理ファイルの間で変換が行われます。データは正規化されません。

#### 関連情報

CCSID (コード化文字セット識別コード) キーワード

i5/OS グローバリゼーション

## 物理ファイルおよび論理ファイルの CHECK (検査) キーワード

これはフィールド・レベル・キーワードで、表示装置ファイルの妥当性検査を指定するために使用します。

キーワードの形式は次のとおりです。

CHECK(edit-check-code [. . .])

CHECK キーワードは、定義中の物理または論理ファイルには影響を与えません。表示装置ファイルに入力可能フィールドを定義する時点で、29 桁目に R を指定し、REF キーワードまたは REFFLD キーワードを使用することによって、定義するフィールドを参照することができます。表示装置ファイル作成時に、オペレーティング・システムは、CHECK キーワードおよびその他のフィールド属性を、物理ファイルまたは論理ファイルの該当フィールドから表示装置ファイルのフィールドに複写します。CHECK キーワード(およびその他のすべての妥当性検査キーワードならびに CHKMSGID キーワード) は、表示装置ファイルのフィールドに他の妥当性検査キーワードを指定することによって指定変更することができます。

このキーワードを物理および論理ファイルで指定する場合の規則は、表示装置ファイルで指定する場合によく似ています。ただし、物理および論理ファイルで指定できるのは下記の各コードだけです。

#### コード 意味

- AB** ブランク使用可能
- ME** 入力必須
- MF** 全桁入力必須
- M10** IBM® モジュラス 10 自己検査アルゴリズム
- M10F** IBM モジュラス 10 自己検査アルゴリズム
- M11** IBM モジュラス 11 自己検査アルゴリズム
- M11F** IBM モジュラス 11 自己検査アルゴリズム
- VN** 名前の妥当性検査
- VNE** 拡張名の妥当性検査

浮動小数点数フィールド (35 桁目が F) については、CHECK(AB)、CHECK(VN)、CHECK(VNE)、および CHECK(M10)、CHECK(M11)、CHECK(M10F)、または CHECK(M11F) キーワードを指定することはできません。また、16 進数フィールド (35 桁目が H) については、CHECK キーワードは指定できません。日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に CHECK キーワードを指定してはなりません。

#### 関連情報

表示装置ファイルの参照 (29 桁目)

表示装置ファイルの CHECK (チェック) キーワード

## 物理ファイルおよび論理ファイルの CHKMSGID (検査メッセージ ID) キーワード

これはフィールド・レベル・キーワードで、妥当性検査キーワードに関連付けるエラー・メッセージを識別するために使用します。

CHKMSGID キーワードを指定しない場合、システム提供メッセージが使用されます。CHKMSGID キーワードを指定した定義中のフィールドを、あとで表示装置ファイル作成時に参照すると、その妥当性検査情報と CHKMSGID キーワードが表示装置ファイルに複写されます。画面からの入力の検査中に妥当性検査エラーが見つかったら、CHKMSGID キーワードで指定したエラー・メッセージがメッセージ行に表示されます。

CHKMSGID キーワードは、定義中の物理 / 論理ファイルには影響を与えません。

キーワードの形式は次のとおりです。

```
CHKMSGID(message-id [library/]message-file [message-data-field])
```

message-data-field パラメーターを指定しても、物理ファイルまたは論理ファイル内にその名前のフィールドを定義する必要はありません。しかし、CHKMSGID キーワードが指定されているフィールドを表示装置ファイル作成時に参照する場合は、その表示装置ファイル内に (CHKMSGID キーワードが指定されているフィールドと同一のレコード様式内に) そのメッセージ・データ・フィールドを定義しなければなりません。

CHKMSGID を指定できるのは、VALUES、RANGE、CMP、COMP、CHECK(M10)、CHECK(M11)、CHECK(VN)、または CHECK(VNE) のいずれかのキーワードも指定するフィールドだけです。

#### 関連情報

表示装置ファイルの CHKMSGID (検査メッセージ識別) キーワード

## 物理ファイルおよび論理ファイルの CMP (比較) キーワード

このキーワードは COMP キーワードと同義です。

キーワードの形式は次のとおりです。

```
CMP(relational-operator value)
```

COMP キーワードを使用することをお勧めします。

#### 関連資料

45 ページの『物理ファイルおよび論理ファイルの COMP (比較) キーワード』

これはフィールド・レベル・キーワードで、定義中のフィールドがあとで表示装置ファイルの作成時に

参照される場合に、そのフィールドの妥当性検査を指定するために使用します。論理ファイルでは、選択/除外フィールド・レベルでもこのキーワードを指定することができます。COMP は、CMP と同義です。

## 物理ファイルおよび論理ファイルの COLHDG (列見出し) キーワード

これはフィールド・レベル・キーワードで、テキスト管理機能、QUERY ユーティリティ、データ・ファイル・ユーティリティ (DFU)、および画面設計機能 (SDA) によってこのフィールドのラベルとして使用される列見出しを指定するために使用します。

キーワードの形式は次のとおりです。

```
COLHDG('line-1' ['line-2' ['line-3']])
```

20 文字以内の行を最大 3 行まで指定することができます。列見出しの各行は、単一引用符 (') で囲まなければならない。列見出しの中でさらに単一引用符を使用したい場合には、二重単一引用符 (' ') を使用します。列見出しの 1 行目、2 行目、および 3 行目を区別するために、各行の間に 1 つまたは複数の空白を使用します。

物理ファイルで、COLHDG キーワードを指定せず、参照フィールドから取り出されなかった場合には、フィールド名が使用されます。論理ファイルで COLHDG を指定しなかった場合には、物理ファイルの列見出しが使用されますが、フィールドが連結フィールドである場合は例外で、この場合にはデフォルトによりフィールド名が使用されます。

COLHDG を指定してあって、TEXT を指定していない場合には、列見出しの始めから 50 桁目までがテキストとして使用されます。たとえば、COLHDG('Order' 'Date') は、TEXT('Order Date') と同じこととなります。

### 例

次の例は、物理ファイルでの COLHDG キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00150A          ORDDAT          5  0          COLHDG('Order' 'Date')
00160A          NAME            20          COLHDG('Customer's Name')
00170A          CITY            20          COLHDG('Customer' 'City' 'Field')
      A
```

Order Date は数字フィールド (以下の図で NNNNN と示されている) であるため、ORDDAT については小数点以下の桁数またはデータ・タイプを指定しなければなりません。

次の画面は、テキスト管理機能、QUERY ユーティリティ、DFU、または SDA の実行時に、列見出しがどのように表示されるかを示しています。

```
Customer
Order          City
Date           Customer's Name  Field
NNNNN         XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
```

## 物理ファイルおよび論理ファイルの COMP (比較) キーワード

これはフィールド・レベル・キーワードで、定義中のフィールドがあとで表示装置ファイルの作成時に参照される場合に、そのフィールドの妥当性検査を指定するために使用します。論理ファイルでは、選択/除外フィールド・レベルでもこのキーワードを指定することができます。COMP は、CMP と同義です。

キーワードの形式は次のとおりです。

COMP (relational-operator value)

選択/除外フィールド・レベルでは、このキーワードの形式は次のとおりです。

COMP (relational-operator field-name)

有効な関係演算子は次のとおりです。

### 関係演算子

#### 意味

<b>EQ</b>	に等しい
<b>NE</b>	に等しくない
<b>LT</b>	より小さい
<b>NL</b>	より小さくない
<b>GT</b>	より大きい
<b>NG</b>	より大きくない
<b>LE</b>	より小さいかまたは等しい
<b>GE</b>	より大きいまたは等しい

value パラメーターは、フィールド・レベルまたは選択/除外フィールド・レベルで指定することができます。field-name パラメーターは、選択/除外フィールド・レベルでだけ指定することができます。

### 例 1

次の例に、文字ストリングおよび数値ストリングをパラメーターとする COMP キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD          PFILE(PF1)
00020A
00030A          FIELDA             1 0      COMP(NE 0)      1
00040A          FIELDB             1        COMP(NE 'A')    1
00050A          FIELDC
00060A          FIELDD
00070A          FIELDE
00080A          K FIELDB
00090A          S FIELDC           COMP(EQ FIELDD)  2
00100A          S FIELDA           COMP(NE 0)      2
00110A          S FIELDE           COMP(NE *NULL)  2
00120A          O FIELDB           COMP(GE 'A')    2
      A
```

- 1 COMP は、FIELDA および FIELDB を参照する表示装置ファイルについての妥当性検査キーワードとして、FIELDA および FIELDB に対して指定されています。
- 2 COMP は、FIELDC、FIELDA、FIELDDB、および FIELDE に対する選択/除外キーワードとして指定されています。物理ファイル PF1 のレコードは、次の比較に基づいて、この論理ファイルのレコード様式を介して取り出されます。
  - FIELDC: FIELDC が FIELDDD に等しい場合に、レコードが選択されます。
  - FIELDA: FIELDC についてのテストに合格しなかったレコードは、FIELDA がゼロでない場合に限り選択されます。
  - FIELDE: FIELDA テストに合格しなかったレコードは、FIELDE がヌル値でない場合に限り選択されます。

## 例 2

次の例に、16 進数文字ストリングをパラメーターとする COMP キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RCD1      PFILE(PF1)
00020A      CODEA
00030A      FLD1
00040A      FLD2
00050A      K FLD1
00060A      S CODEA      COMP(EQ X'51')
      A
```

ここでは、COMP は、CODEA (1 バイトのフィールド) についての選択/除外キーワードとして指定されています。物理ファイル PF1 のレコードは、フィールド CODEA の値が 16 進数 51 である場合に限り、このレコード様式を介して取り出されます。

### 関連資料

44 ページの『物理ファイルおよび論理ファイルの CMP (比較) キーワード』  
このキーワードは COMP キーワードと同義です。

## フィールド・レベルの COMP の指定

フィールド・レベルでは COMP が定義中の物理または論理ファイルに影響を及ぼすことはありません。

しかし、表示装置ファイルに入力可能フィールドを定義する時点で、29 桁目に R を指定し、REF キーワードまたは REFFLD キーワードを指定することによって、記述しているフィールドを参照することができます。表示装置ファイル作成時に、オペレーティング・システムは、COMP キーワードおよびその他のフィールド属性を、論理ファイルの該当フィールドから表示装置ファイルのフィールドに複写します。COMP キーワード (およびその他のすべての妥当性検査キーワードならびに CHKMSGID キーワード) は、表示装置ファイルのフィールドに他の妥当性検査キーワードを指定することによって指定変更することができます。

フィールド・レベルの COMP キーワードには、フィールド名をパラメーター値として指定することはできません。

フィールド・レベルの COMP キーワードに \*NULL をパラメーター値として指定することはできません。

浮動小数点数フィールド (35 桁目が F) または 16 進数フィールド (35 桁目が H) に対しては、COMP キーワードは指定できません。日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に COMP キーワードを指定してはなりません。

物理ファイルまたは論理ファイルでのこのキーワードの指定に関する規則は、表示装置ファイルの場合と同じです。

## 物理ファイルおよび論理ファイルの数字フィールドの定義

ワークステーション・ユーザーがデータを入力すると、オペレーティング・システムは、フィールドの小数点以下の桁数に従って、入力された文字の位置合わせを行います。フィールドがユーザーのプログラムに渡される時点で、先行ブランクと後書きブランクはゼロで埋め込まれます。小数点文字を入力しなかった場合には、入力された右端の文字の右側に、オペレーティング・システムが小数点文字を入れます。たとえば、長さ (34 桁目で指定) が 5 で、小数点以下の桁数 (37 桁目で指定) が 2 の数字フィールドの場合には、1.2 は 001.20 として、100 は 100.00 として解釈されます。

### 関連情報

表示装置ファイルの参照 (29 桁目)

表示装置ファイルの COMP (比較) キーワード

## 選択/除外フィールド・レベルの COMP の指定

選択/除外フィールド・レベルでは、パラメーターにフィールド名、値、または \*NULL のいずれかを指定することができます。

選択/除外フィールドが 2 進文字フィールドである場合は、フィールド名パラメーターも 2 進文字フィールドでなければなりません。2 進文字の選択/除外フィールドの比較では、オペランドの実際の長さを考慮する必要もあります。比較するオペランドの実際の長さが等しい場合にのみ、それらのオペランドが対等に比較されます。長いオペランドの長さが短いオペランドの長さと同じになるまでは、短いオペランドは長いオペランドよりも小さいと見なされます。

値を指定する場合には、次の規則が適用されます。

- 定義しているフィールドが文字フィールドの場合には、文字定数または 16 進数文字ストリングを指定しなければなりません。

文字ストリングは、単一引用符を使用して指定します (45 ページの『物理ファイルおよび論理ファイルの COMP (比較) キーワード』トピックの例 1 を参照)。

16 進数文字ストリングは、X の後に、数字 (0 から 9) と文字 (A から F) の組み合わせを単一引用符 (') で囲んで指定します。単一引用符内の 16 進数字の数は、指定したフィールドの長さの 2 倍でなければなりません (45 ページの『物理ファイルおよび論理ファイルの COMP (比較) キーワード』トピックの例 2 を参照)。

- 定義しているフィールドが数字フィールドの場合には、例 2 に示すように、数値ストリング (単一引用符で囲まない 0 から 9 の数字) を指定しなければなりません。
- 定義しているフィールドが日付フィールドの場合には、DATFMT キーワードで指定されているのと同じ形式で有効な日付を指定し、DATSEP キーワードで指定されているのと同じ区切り記号を使用してください。

たとえば、DATFMT に \*MDY が指定され、DATSEP に「/」が指定されている場合、COMP(EQ '12/15/91') は有効な値です。

- 定義しているフィールドが時刻フィールドの場合には、TIMFMT キーワードで指定されているのと同じ様式で有効な時刻を指定し、TIMSEP キーワードで指定されているのと同じ区切り記号を使用してください。

たとえば、TIMFMT で \*ISO が指定されている場合、COMP(EQ '11.00.00') は有効な値です。\*ISO のデフォルトの区切り記号はピリオド (.) です。

- 定義しているフィールドが時刻スタンプ・フィールドの場合には、次の形式でデフォルト値を指定しなければなりません。

```
COMP(EQ 'YYYY-MM-DD-HH.MM.SS.UUUUUU')
```

\*NULL を指定する場合、関係演算子は EQ または NE でなければなりません。

COMP は、定義中のレコード様式に対する入力命令をユーザーのプログラムが送るときに、この論理ファイルの基礎となっている物理ファイルから取り出されたレコードを選択または除外するために使用します。オペレーティング・システムは、ユーザーが指定した値、またはユーザーがその名前を指定したフィールドの値、あるいはヌル値 (\*NULL を指定した場合) のいずれかと、選択/除外フィールドの値とを比較した結果に基づいて、レコードを選択または除外します。

## CONCAT (連結) キーワード - 論理ファイルのみ

これはフィールド・レベル・キーワードで、物理ファイルのレコード様式からの 2 つ以上のフィールドを、定義中の論理ファイルのレコード様式内で連結して 1 つのフィールドにしたい場合に使用します。連結後にできるフィールドの名前を 19 から 28 桁目に指定しなければなりません。

キーワードの形式は次のとおりです。

```
CONCAT(field-1 field-2...)
```

物理ファイルのフィールドの名前は、連結したい順に指定し、各フィールド名をブランクで区切ります。

論理ファイルの 1 つのレコード様式の中で、同じ物理フィールドを (RENAME または CONCAT を使用して) 複数回指定した場合、更新操作または挿入操作では、論理ファイル内でそのフィールドの指定順序と同じ順序でその物理ファイルにデータが渡されます。したがって、その物理フィールドの最後の指定が、物理レコードに入れられる値であり、その物理フィールドをキー・フィールドとするすべてのキーで使用される値です。同じフィールドのそれ以前の値はすべて無視されます。

CONCAT キーワードを用いて定義したフィールド、または CONCAT キーワードのパラメーター値として指定したフィールドをキー・フィールドとして使用したい場合は、キー・フィールド名の項を参照してください。

2 進文字フィールドは、別の 2 進文字フィールドとのみ連結できます。UTF-8 フィールドは、別の UTF-8 フィールドとのみ連結できます。UCS-2 および UTF-16 フィールドは、同じタイプのフィールドでのみ、または相互に連結できます。

小数点以下の桁数としてゼロ以外の値が指定されているフィールドを連結フィールドに含めることはできません。小数点以下の桁数がゼロのフィールドは、連結フィールドに含めることができ、そのフィールドは整数フィールドとして取り扱われます。

i5/OS プログラムは、連結したすべてのフィールドの合計長 (数字および文字) として、連結フィールドの長さを割り当てます。

オペレーティング・システムは、連結するフィールドに基づいて、フィールドを固定長フィールドまたは可変長フィールドになるように割り当てます。その場合の一般的な規則は次のとおりです。

- 可変長フィールドを固定長フィールドまたは別の可変長フィールドに連結する場合、連結後のフィールドは、可変長フィールドとなります。
- 固定長フィールドを固定長フィールドに連結する場合、連結後のフィールドは、CONCAT キーワードと同じフィールドで VARLEN キーワードが指定されていない場合は、固定長フィールドとなります。

**注:** 連結の結果が、ヌル値を許容する可変長フィールド、UCS-2 フィールド、UTF-16 フィールド、UTF-8 フィールド、または 2 進文字フィールドである場合には、CONCAT フィールドは入力専用 (38 桁目が I) でなければなりません。論理ファイル・レコード様式に連結が含まれる場合、基本ファイルの物理ファイル・レコード様式からのヌル値を許容するフィールドは含めることはできません。

オペレーティング・システムは、連結しようとするフィールドのデータ・タイプに基づいてデータ・タイプを割り当てます。その場合の一般的な規則は次のとおりです。

- 連結の対象に 1 つまたは複数の 16 進数 (H) フィールドが含まれている場合は、結果のデータ・タイプは 16 進数 (H) です。

- 連結の対象に 1 つまたは複数の文字 (A) フィールドが含まれており、16 進数フィールドが含まれていない場合は、結果のデータ・タイプは文字 (A) です。
- 連結の対象が数値 (S、P、B) フィールドだけである場合は、結果のデータ・タイプはゾーン 10 進数 (S) です。
- 連結に UTF-8 フィールドが含まれている場合、結果は UTF-8 フィールドになります。
- 連結に UCS-2 または UTF-16 フィールドが含まれている場合は、リストに 1 つでも UTF-16 があれば結果は UTF-16 になり、そうでなければ、UCS-2 になります。
- 連結に 2 進文字フィールドが含まれている場合、結果は 2 進文字になります。

数字フィールドを連結する場合には、連結される右端のフィールドの符号が連結フィールドの符号として使用されます。その他のフィールドの符号は無視されますが、連結フィールド内にそのまま残っています。したがって、連結の最後にあるフィールド以外のフィールドに負の値が入る可能性がある場合には、適切な処置 (連結フィールドをパック 10 進数に変換するなど) を行うことにより、最後のフィールド以外のフィールドの符号を削除しなければなりません。

連結フィールドの最大長は、連結フィールドのデータ・タイプおよび連結対象フィールドの長さによって異なります。連結フィールドがゾーン 10 進数 (S) である場合には、その合計長は 63 バイト以下でなければなりません。また、それが文字 (A) または 16 進数 (H) の場合には、合計長は 32 766 バイト以下でなければなりません。連結フィールドが可変長フィールドである場合、その合計長は 32 740 (ヌル値も許容するフィールドの場合は 32 739) を超えてはなりません。

連結フィールドに浮動小数点数、日付、時刻、または時刻スタンプ・フィールドを含めることはできません。

結合論理ファイルの場合には、連結するフィールドはすべて同じ物理ファイルからのフィールドでなければなりません。CONCAT キーワードで最初に指定するフィールドにより、どの物理ファイルを使用するかが識別されます。そのため、最初のフィールドは、結合論理ファイルの基礎となる物理ファイルの中で固有になっていなければならないか、またはどの物理ファイルを使用するか指定するために CONCAT キーワードに続いて JREF キーワードも指定しなければなりません。

## 例

以下に CONCAT キーワードの指定方法の例を示します。

### 例 1

以下の例では、MTH、DAY、および YEAR は物理ファイル内のフィールドであり、論理ファイル内で DATE という 1 つのフィールドに連結されます。

	....1....	....2....	....3....	....4....	....5....	....6....	....7....	....8
00010A		R RECORD1			PF1(PF1)			
00020A		DATE			CONCAT(MTH DAY YEAR)			
A								

### 例 2

次の例では、プログラムで DATE が 01 03 81 から 02 05 81 に変更されても、物理レコードに置かれる値は変わりません。これは、最後に指定されたフィールドが、MTH (値 01)、DAY (値 03)、および YEAR (値 81) であるためです。ただし、MTH、DAY、および YEAR が新しい値に変更された場合には、物理レコード内の DATE の値も変更されます。



```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R RECORD2      PFILE(PF1)
00020A      DATE          CONCAT(MTH DAY YEAR)
00030A      MTH
00040A      DAY
00050A      YEAR
      A

```

### 例 3

次の例では、物理ファイルのフィールドが論理ファイル内で複数のフィールドに連結されます。

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R RECORD3      PFILE(PF1)
00020A      DATE          CONCAT(MTH DAY YEAR)
00030A      CMPDAT        CONCAT(DAY MTH YEAR)
      A

```

### 例 4

この例で、PF1 からのフィールドが次のとおりであったとします。

- FIXED1 は固定長フィールド。
- FIXED2 は固定長フィールド。
- VARLEN1 は可変長フィールド。

この場合、結果的に各フィールドは次のようになります。

- FIELD1 は可変長フィールド。
- FIELD2 は固定長フィールド。
- FIELD3 は可変長フィールド。

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R RECORD4      PFILE(PF1)
00020A      FIELD1        CONCAT(FIXED1 VARLEN1)
00030A      FIELD2        CONCAT(FIXED1 FIXED2)
00040A      FIELD3        CONCAT(FIXED1 FIXED2)
00050A      VARLEN
      A

```

#### 関連概念

10 ページの『フィールド名』

17 桁目をブランクのままにした場合には、19 から 28 桁目に指定する名前はフィールド名になります。

10 ページの『キー・フィールド名』

17 桁目に K を指定した場合には、19 から 28 桁目に指定する名前はキー・フィールド名になります。

#### 関連資料

101 ページの『CONCAT (連結) キーワード』

これはフィールド・レベル・キーワードで、物理ファイルのレコード様式の複数のフィールドを組み合わせて、定義中の論理ファイルのレコード様式内で 1 つのフィールドとして用いたい場合に使用します。

## 物理ファイルおよび論理ファイルの DATFMT (日付形式) キーワード

これはフィールド・レベル・キーワードで、日付フィールドの形式を指定するために使用します。このキーワードは、日付フィールド (データ・タイプ L) か、または対応する物理ファイルのフィールドが日付フィ

ールド (データ・タイプ L) である論理ファイルのゾーン・フィールド (データ・タイプ S)、パック 10 進数フィールド (データ・タイプ P)、または文字フィールド (データ・タイプ A) にだけ使用できます。

キーワードの形式は次のとおりです。

DATFMT(date-format)

date-format パラメーターで日付の形式を指定します。次の表に、物理ファイル・フィールドに使用できる日付形式とそれぞれのデフォルトの区切り記号の値を示します。

形式名	日付形式パラメーター	日付形式および区切り記号	フィールド長	例
ジョブのデフォルト値	*JOB <sup>1</sup>			
月/日/年	*MDY <sup>1</sup>	mm/dd/yy	8	06/21/90
日/月/年	*DMY <sup>1</sup>	dd/mm/yy	8	21/06/90
年/月/日	*YMD <sup>1</sup>	yy/mm/dd	8	90/06/21
年間通算日	*JUL <sup>1</sup>	yy/ddd	6	90/172
国際標準化機構	*ISO	yyyy-mm-dd	10	1990-06-21
IBM 米国規格	*USA	mm/dd/yyyy	10	06/21/1990
IBM 欧州規格	*EUR	dd.mm.yyyy	10	21.06.1990
日本工業規格西暦	*JIS	yyyy-mm-dd	10	1990-06-21

注: この形式が指定されており、かつフィールドがヌル値を許容するときには、このフィールドの DFT キーワードに有効な日付を指定しなければなりません。

物理ファイル・フィールドの DATFMT キーワードに適用されるその他の属性は、以下のとおりです。

- 日付 (L) データ・タイプには DATFMT キーワードしか指定できません。
- DATFMT キーワードを指定しない場合、デフォルト値は \*ISO です。
- フィールド長値と小数点以下の桁数値はブランクでなければなりません。

次の表に、論理ファイルに使用できる日付形式とそれぞれのデフォルトの区切り記号の値を示します。

形式名	日付形式パラメーター	日付形式	ゾーンまたは文字フィールド長	ゾーンまたは文字の例	パック・フィールド長	パックスの例 (16 進数)
ジョブのデフォルト値	*JOB					
月/日/年	*MDY	mmddy	6,0	062196	6,0 または 7,0	'0062196F'X
日/月/年	*DMY	ddmmy	6,0	210696	6,0 または 7,0	'0210696F'X
年/月/日	*YMD	yyymmdd	6,0	960621	6,0 または 7,0	'0960621F'X
月/日/年 (年は 4 桁)	*MDYY <sup>1</sup>	mmddyyy	8,0	06211996	8,0 または 9,0	'006211996F'X
日/月/年 (年は 4 桁)	*DMYY <sup>1</sup>	ddmmyyy	8,0	21061996	8,0 または 9,0	'021062006F'X
年/月/日 (年は 4 桁)	*YYMD <sup>1</sup>	yyyymmdd	8,0	19960621	8,0 または 9,0	'019960621F'X
年間通算日	*JUL	yyddd	5,0	96172	5,0	'96172F'X
ユリウス暦 (年は 4 桁)	*LONGJUL <sup>1</sup>	yyyyddd	7,0	1996172	7,0	'1996172F'X

形式名	日付形式パラメーター	日付形式	ゾーンまたは文字フィールド長	ゾーンまたは文字の例	パック・フィールド長	パックの例 (16 進数)
世紀/日/月/年	*CMDY <sup>1</sup>	cmmdyy	7,0	0062196	7,0	'0062196F'X
世紀/日/月/年	*CDMY <sup>1</sup>	cddmmyy	7,0	1210696	7,0	'1210696F'X
世紀/年/月/日	*CYMD <sup>1</sup>	cyymmdd	7,0	1960621	7,0	'1960621F'X
月/年	*MY <sup>1,2</sup>	mmyy	4,0	0696	4,0 または 5,0	'00696F'X
年/月	*YM <sup>1,2</sup>	yymm	4,0	9606	4,0 または 5,0	'09606F'X
月/年 (年は 4 桁)	*MY <sup>1,2</sup>	mmyyyy	6,0	061996	6,0 または 7,0	'0061996F'X
年/月 (年は 4 桁)	*YM <sup>1,2</sup>	yyymm	6,0	199606	6,0 または 7,0	'0199606F'X
国際標準化機構	*ISO	yyyymmdd	8,0	19960621	8,0 または 9,0	'019960621F'X
IBM 米国規格	*USA	mmddyyyy	8,0	19960621	8,0 または 9,0	'006211996F'X
IBM 欧州規格	*EUR	ddmmyyyy	8,0	21061996	8,0 または 9,0	'021061996F'X
日本工業規格西暦	*JIS	yyyymmdd	8,0	19960621	8,0 または 9,0	'019960621F'X

**注:**

- これらの DATFMT は日付 (L) タイプ・フィールドには有効ではありません。対応する物理ファイルのフィールドが日付タイプ・フィールドである論理ファイルのゾーン 10 進数、パック 10 進数、または文字タイプにのみ有効です。
- DATFMT に「日」の値が指定されていない場合、暗黙に当該月の「1 日」を指定しているものとします。

論理ファイル・フィールドの DATFMT キーワードに適用されるその他の属性は、以下のとおりです。

- 論理ファイル・フィールドのパック (P)、ゾーン (S)、文字 (A)、および日付 (L) データ・タイプには、DATFMT キーワードを指定できます。
- パック、文字、およびゾーンの論理ファイル・フィールドにフィールド長を指定することはできませんが、上記の表にリストされている有効な値でなければなりません。
- DATFMT キーワードを指定しておらず、かつデータ・タイプが L の場合、デフォルト値は対応する物理ファイル・フィールドからの日付形式とフィールド長になります。
- パックおよびゾーン・データ・タイプの場合、小数点以下の桁数 (36, 37 桁) はブランクにしなければなりません。
- パック・データ・タイプの場合、場合によっては、特定の 1 つの形式に対して 2 種類の長さを指定できることがあります。値が大きい長さの方が、パフォーマンスはよくなります。値を 1 つしか指定しない場合、値が小さい長さがデフォルト値として使用されます。

物理ファイル・フィールドおよび論理ファイル・フィールドの両方に適用される DATFMT キーワードの属性には、以下の点が含まれます。

- \*JOB を指定した場合にはジョブ属性がデフォルト値となり、フィールド長は区切り記号なしのジョブ属性に基づきます。

- DFT キーワードが指定されていない場合は、現在の日付がデフォルト値です。
- \*ISO、\*USA、\*EUR、または \*JIS を指定した場合、DATSEP キーワードは使用できません。これらの日付形式には既定の区切り記号があります。
- DATFMT キーワードは、日付フィールドについてのジョブ属性を指定変更します。システム・デフォルト値を変更するものではありません。

## 例

次の例は、DATFMT キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A          DATFLD1          L          DATFMT(*JUL)
00040A          DATFLD2          L          DATFMT(*EUR)
          A
```

この例では、現在の日付が 1990 年 6 月 21 日で、現在のシステムの日付形式値が MDY で、システムの日付区切り記号が / であるとする、DATFLD1 には 90/172 (1990 年の 172 番目の日) が入ります。DATFLD2 には 21.06.1990 が入ります。

## 物理ファイルおよび論理ファイルの DATSEP (日付区切り記号) キーワード

これはフィールド・レベル・キーワードで、日付フィールドの区切り記号を指定するために使用します。このキーワードは日付フィールド (データ・タイプ L) にだけ使用することができます。

キーワードの形式は次のとおりです。

```
DATSEP(*JOB | 'date-separator')
```

date-separator パラメーターにより、年、月、および日の値の間に入れる区切り記号の値を指定します。有効な値はスラッシュ (/)、ダッシュ (-)、ピリオド (.), コンマ (,), またはブランク ( ) のいずれかです。このパラメーターは単一引用符 (') で囲まなければなりません。

\*JOB を指定した場合、ジョブ属性がデフォルト値となります。

物理ファイルでは、DATSEP キーワードを指定しない場合、ジョブ属性がデフォルト値となります。

論理ファイルでは、DATSEP キーワードを指定しない場合、物理ファイルの日付区切り記号がデフォルト値となります。物理ファイル・フィールド (DATFMT キーワードに \*ISO、\*USA、\*EUR、または \*JIS が指定されている) に DATSEP キーワードを指定しない場合には、DATSEP のデフォルト値はジョブ属性です。

DATFMT キーワードに \*ISO、\*USA、\*EUR、または \*JIS の日付形式値を指定した場合には、DATSEP キーワードを指定することはできません。これらの形式には既定の日付区切り記号があります。

DATSEP キーワードは、ジョブ属性を指定変更します。システム・デフォルト値を変更するものではありません。

## 例

次の例は、DATSEP キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD1
00030A          DATFLD2          L          DATFMT(*DMY) DATSEP('-')
00040A          DATFLD4          L          DATSEP(' ')
          A

```

この例では、現在の日付が 1990 年 6 月 21 日で、現在のシステムの日付形式値が MDY で、システムの日付区切り記号が '/' であるとする、DATFLD2 には 21-06-90 が入ります。DATFLD4 には 06 21 90 が入ります。

## 物理ファイルおよび論理ファイルの DESCEND (降順) キーワード

これはキー・フィールド・レベル・キーワードで、該当の文字、16 進数、または数字のキー・フィールドを降順に検索することを指定するために使用します。

デフォルトの順序は昇順です。DESCEND キーワードを使って保管されたデータの例については、84 ページの『物理ファイルおよび論理ファイルの SIGNED (符号付き) キーワード』を参照してください。

このキーワードにはパラメーターはありません。

### 例

次の例は、論理ファイルの場合の DESCEND キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          K ITEM
00020A          K BALDUE          DESCEND
          A

```

## DFT (デフォルト) キーワード - 物理ファイルのみ

これは、フィールド・レベル・キーワードで、フィールドのデフォルト値を指定するために使用します。

キーワードの形式は次のとおりです。

```
DFT('value' | numeric-value | X'hexadecimal-value' | *NULL)
```

このキーワードを指定しなかった場合、文字フィールドおよび 16 進数フィールドはブランクがデフォルト値になり、数字フィールドはゼロがデフォルト値になります。ただし、フィールドに ALWNULL キーワードを指定した場合、文字フィールド、16 進数フィールド、および数字フィールドのデフォルト値はヌル値になります。

指定する値については以下の規則があります。

- 定義中のフィールドが文字フィールドである場合には、文字定数、16 進値、あるいは \*NULL を指定してください。文字ストリングは単一引用符で囲んで指定してください。フィールドが可変長 (VARLEN) の場合には、ストリングの長さは、割り振られた長さ以下でなければなりません。

16 進値は、X の後に、数字 (0 から 9) と文字 (A から F) の組み合わせを単一引用符で囲んで指定します。単一引用符内の 16 進数の桁数は、指定したフィールドの長さの 2 倍でなければなりません。フィールドが可変長 (VARLEN) の場合には、単一引用符内の 16 進数の桁数は、割り振られた長さの 2 倍でなければなりません。

- 定義中のフィールドが 16 進数フィールドである場合は、文字定数、16 進値、あるいは \*NULL を指定してください。

注: 文字定数を指定した場合、その文字定数の 16 進数表現がデフォルト値となります。

文字ストリングは単一引用符で囲んで指定してください。フィールドが可変長 (VARLEN) の場合には、ストリングの長さは、割り振られた長さ以下でなければなりません。

16 進値は、X の後に、数字 (0 から 9) と文字 (A から F) の組み合わせを単一引用符で囲んで指定します。単一引用符内の 16 進数の桁数は、指定したフィールドの長さの 2 倍でなければなりません。フィールドが可変長 (VARLEN) の場合には、単一引用符内の 16 進数の桁数は、割り振られた長さの 2 倍でなければなりません。

- 定義中のフィールドが数字フィールドである場合には、数値 (単一引用符で囲まない数字 0 から 9) か、または \*NULL を指定してください。36 から 37 桁目の指定がゼロでない場合には、DDS の適切な位置に数字定数とともに小数点文字を指定してください。
- \*NULL を指定する場合は、そのフィールドに ALWNULL キーワードも指定しなければなりません。
- 何も値を指定しない場合 (DFT('')) には、これは長さが 0 のストリングをデフォルト値とすることを意味し、そのフィールドが可変長である (VARLEN キーワードも指定する) ときにだけ有効です。
- 定義しているフィールドが日付フィールドの場合には、DATFMT キーワードで指定されているのと同じ形式で有効な日付を指定し、DATSEP キーワードで指定されているのと同じ区切り記号を使用してください。たとえば、DATFMT で \*MDY が指定され、DATSEP で 「/」 が指定されているときは、DFT('12/15/91') は有効な値です。

DFT キーワードが指定されていない場合は、現在の日付がデフォルト値です。

- 定義しているフィールドが時刻フィールドの場合には、TIMFMT キーワードで指定されているのと同じ様式で有効な時刻を指定し、TIMSEP キーワードで指定されているのと同じ区切り記号を使用してください。たとえば、TIMFMT で \*ISO が指定されている場合は、DFT('11.00.00') は有効な値です。\*ISO のデフォルトの区切り記号はピリオド (.) です。

DFT キーワードが指定されていない場合は、現在の時刻がデフォルト値です。

- 定義しているフィールドが時刻スタンプ・フィールドの場合には、次の形式でデフォルト値を指定しなければなりません。

```
DFT('YYYY-MM-DD-HH.MM.SS.UUUUUU')
```

DFT キーワードが指定されていない場合は、現在の時刻がデフォルト値です。

指定した値は、以下の場合にフィールドに割り当てられます。

- プログラムが、この物理ファイルを基礎にする論理ファイルに対して出力操作を行う場合に、その論理ファイルのレコード様式に該当フィールドの名前が指定されていない場合。
- このファイルのメンバーに対して物理ファイル・メンバーの初期設定 (INZPFM) コマンドを使用した場合。
- 複写先ファイルに、複写元ファイルに存在しないフィールドがある場合に、FMTOPT(\*MAP) を指定したファイル・コピー (CPYF) コマンドを使用した場合。

指定した値は、プログラムが結合論理ファイルに対して入力操作を行うさいに、次のすべての条件が満たされていれば、プログラムに渡されます。

- 結合論理ファイルについて JDFTVAL キーワードが指定されている。
- 定義中のファイルが、結合論理ファイルの 2 次ファイルとして指定されている。
- 入力操作が実行され、2 次ファイルへの関係でレコードが作成されなかった。

入力操作では、このキーワードは物理ファイルに影響を与えません。

## 例

次の例は、DFT キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A          CHARFLD1      20A      DFT('Sample field')
00030A          CHARFLD2      5A       DFT(X'D985955185')
00040A          HEXFLD1       3H      DFT('ABC')
00050A          HEXFLD2       3H      DFT(X'C1C2C3')
00060A          NUMFLD1       5S 0   DFT(99999)
00070A          NUMFLD2       5S 2   DFT(999.99)
00080A          NUMFLD3       5S 2   DFT(999)
00090A          NUMFLD4       5S 2   DFT(*NULL)
00100A          ALWNULL
00110A          NUMFLD5       5S 2   DFT(999.99)
00120A          ALWNULL
00130A          DATFLD1       L       DATFMT(*MDY) DATSEP('-')
00140A          DFT('12-31-91')
00150A          TIMFLD1       T       DFT('11.15.00')
      A
```

この例では、CHARFLD1 のデフォルト値は 'Sample field' です。CHARFLD2 のデフォルト値は 16 進数 D985955185 です。HEXFLD1 のデフォルト値は C1C2C3 (文字定数の 16 進数表現) です。HEXFLD2 のデフォルト値は C1C2C3 です。NUMFLD1 のデフォルト値は 99999 です (フィールドの小数点以下の桁数がゼロなので小数点文字は不要です)。NUMFLD2 のデフォルト値は 999.99 です。NUMFLD3 のデフォルト値は 999 です (小数部分を指定する必要がない場合には小数点文字は不要です)。NUMFLD4 のデフォルト値はヌル値です (DFT(\*NULL) を指定する場合は、そのフィールドに ALWNULL キーワードを必ず指定しなければなりません)。NUMFLD5 のデフォルト値は 999.99 で、このフィールドにはヌル値も許容されます。DATFLD1 のデフォルト値は 12-31-91 です。TIMFLD1 のデフォルト値は 11.15.00 (\*ISO 形式) です。

### 関連資料

34 ページの『物理ファイルおよび論理ファイルの使用目的 (38 桁目)』

この欄は、名前を付けたフィールドを、入力専用にするか、入出力共用フィールドにするか、あるいは非入出力フィールドにするかを指定するのに使用します。

64 ページの『JDFTVAL (結合におけるデフォルト値) キーワード - 結合論理ファイルのみ』

これは結合論理ファイルで使用するファイル・レベル・キーワードで、2 次ファイルとの結合の結果、目的のレコードが得られない場合に、システムがフィールドにデフォルト値を与えることを指定するために使用します。JDFTVAL は、結合論理ファイルに対してだけ有効です。

## 物理ファイルおよび論理ファイルの DIGIT (ディジット) キーワード

これは、キー・フィールド・レベル・キーワードで、このキー・フィールドに関連する値が構成される際に、キー・フィールドの各バイトのゾーン部分 (下位 4 ビット) だけを使用することを指定するために使用します。ゾーン部分はゼロで埋められます。

このキーワードにはパラメーターはありません。

DIGIT キーワードは 1 つのキー・フィールド全体に適用されるもので、単にフィールド内の 1 桁だけに提供されるものではありません。このキーワードは、文字、16 進数、またはゾーン 10 進数のデータ・タイプのフィールドについてだけ有効です。

このキーワードを、ABSVAL、SIGNED、または ZONE の各キーワードとともに使用することはできません。

キー・フィールドに DIGIT を指定した場合には、フィールドの値は、符号付きデータではなく、符号なしの 2 進データのストリングと見なされます。これは、ゾーン 10 進数フィールドについてのデフォルト値です。

## 例

次の例は、DIGIT キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00040A          K ORDTYP          DIGIT
  A
```

この例では、ORDTYP が 3 バイトのフィールドの場合、3 つの異なるレコードにおけるこのフィールドの値は、たとえば下記のようになります。

値	16 進数	キーとして使用される部分
C4J	C3F4D1	341
CMA	D3D4C1	341
3D1	F3C4F1	341

## DYNSLT (動的選択) キーワード - 論理ファイルのみ

これはファイル・レベル・キーワードで、選択/除外仕様によりファイルに指定した選択および除外のテストを処理時点で行うことを指定するために使用します。このキーワードは、アクセス・パスの選択/除外ではなく、動的な選択/除外を指定します。

このキーワードにはパラメーターはありません。

DYNSLT キーワードが指定されている論理ファイルに対してプログラムが入力操作を行うと、関連した物理ファイルのすべてのレコードについて、選択/除外値が有効であるかどうかのテストがシステムにより行われます。値を満たすレコードだけがプログラムに渡されます。各レコードをテストすることで入出力のパフォーマンスが低下する場合がありますが、ファイルのアクセス・パスの保守を行うよりは効率のよい場合もあります。これは、読み取り回数の少ない論理ファイルの場合であって、特に、その基礎となっている物理ファイルが頻繁に更新されるような場合に適しています。また、選択されるレコードの占める割合が大きいファイルの場合も、一般に動的選択/除外を用いた方が効率が良くなります。

キー順序アクセス・ファイルの場合には、ファイル作成時にアクセス・パスが作成され、論理ファイル作成 (CRTLF) コマンドまたは論理ファイル変更 (CHGLF) コマンドの MAINT パラメーターに従って、そのアクセス・パスの保守が行われます。DYNSLT キーワードは、キー順序アクセス・ファイルのアクセス・パスの保守には影響を及ぼしません。

DYNSLT キーワードを指定したすべての単一樣式論理ファイルについては、選択/除外フィールドを指定するためにキー・フィールドを指定する必要はありません。しかし、DYNSLT キーワードを指定したすべての複数様式論理ファイルについては、少なくとも 1 つのキー・フィールドを指定しなければなりません。このキー・フィールドに \*NONE を指定することができます。

フィールドの選択または除外を行う必要があり、次のいずれかの条件に該当する場合には、DYNSLT キーワードを使用しなければなりません。

- 論理ファイルが到着順である (キー・フィールドが指定されていない)。このトピックの例 1 を参照してください。
- 論理ファイルが、JDFTVAL キーワードの指定された結合論理ファイルである。



- 論理ファイルが結合論理ファイルであり、選択/除外フィールドが、その論理ファイルの基礎となっている複数の物理ファイルから取り出されたフィールドであり、かつ、次のいずれかの条件に該当している。
  - 選択/除外フィールドが、同じ選択ステートメントまたは除外ステートメント上で使用されている。このトピックの例 3 を参照してください。
  - 選択/除外フィールドが、選択ステートメントと除外ステートメントの両方で使用されている。このトピックの例 4 を参照してください。
  - 選択/除外フィールドが、OR 関係にある複数の選択ステートメントで使用されている。
  - 選択/除外フィールドが、AND 関係にある複数の除外ステートメントで使用されている。

DYNSLT キーワードは、REFACCPH キーワードとともに指定することはできません。

結合論理ファイルの場合、選択/除外フィールドは、JFILE キーワードで指定されたどの物理ファイルの中にあるフィールドであってもかまいません。したがって、結合論理ファイルでは、フィールドの出所を明確にし、あいまいさを避けるために、JREF キーワードを使用してください。

## 例

以下に、DYNSLT キーワードの指定方法の例を示します。

### 例 1

次の例に、到着順の動的選択の指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                DYNSLT
00020A      R RECORD1                 PFILE(PF1)
00030A      FLD1
00040A      FLD2
00050A      S FLD1                    COMP(GT 2)

```

この例では、キー・フィールドがないため、DYNSLT キーワードは必須です。

レコードは到着順で論理ファイルからプログラムに渡されます。物理ファイル PF1 には、次のレコードがあるものとします。

#### FLD1 FLD2

```

1      aaaa
2      dddd
3      jjjj
4      bbbb

```

プログラムが入力操作を行うさいに、システムは、選択/除外値に従って最初の 2 つのレコードをテストしますが、これらのレコードは、プログラムには渡されません。プログラムに渡されるのは、最後の 2 つのレコードだけです。

#### FLD1 FLD2

```

3      jjjj
4      bbbb

```

## 例 2

次の例に、キー順アクセス・パスを用いた動的選択の指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                DYNSLT
00020A      R RECORD1                 PFILE(PF1)
00030A      FLD1
00040A      FLD2
00050A      K FLD1
00060A      S FLD2                    COMP(GT 'bbbb')
      A
```

この例では、DYNSLT キーワードの指定は必須ではありません。レコードは、論理ファイルからキー順でプログラムに渡されます。物理ファイル PF1 には、次のレコードがあるものとします。

### FLD1 FLD2

```
1      aaaa
2      dddd
3      jjjj
4      bbbb
```

前述のプログラムがレコードを要求すると、システムは、選択/除外値に従ってそのレコードの FLD2 値をテストします。その結果、プログラムには次のレコードだけが渡されます。

### FLD1 FLD2

```
2      dddd
3      jjjj
```

## 例 3

次に示すのは、2 つの物理ファイルから取り出したフィールドを比較する選択/除外仕様を指定した結合論理ファイルの例です。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                DYNSLT
00020A      R RECORD1                 JFILE(PF1 PF2)
00030A      J                          JFLD(FLD1 FLD3)
00040A      FLD1                       JREF(PF1)
00050A      FLD2                       JREF(PF1)
00060A      FLD3                       JREF(PF2)
00070A      FLD4                       JREF(PF2)
00080A      S FLD1                     COMP(GT FLD4)
      A
```

この例では、FLD1 および FLD2 は 1 次ファイル (PF1) のフィールドであり、FLD3 および FLD4 は 2 次ファイル (PF2) のフィールドです。選択仕様により、1 次ファイルの FLD1 と 2 次ファイルの FLD4 とが比較されます。したがって、DYNSLT キーワードは必須です。

## 例 4

次に示すのは、複数の物理ファイルのフィールドを使用する選択仕様および除外仕様を指定した結合論理ファイルの例です。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                DYNSLT
00020A      R JREC                     JFILE(PF1 PF2)
```

```

00030A      J                JOIN(PF1 PF2)
00040A      J                JFLD(FLD1 FLD2)
00050A      FLD1            JREF(PF1)
00060A      FLD2            JREF(PF1)
00070A      FLD3            JREF(PF2)
00080A      K FLD1
00090A      S FLD1            COMP(GT 0)
00100A      O FLD3            COMP(GT 4)
      A

```

この例では、FLD1 および FLD3 は異なる物理ファイルのフィールドであり、選択ステートメントおよび除外ステートメントの両方に指定されています。したがって、DYNSLT キーワードは必須です。

## 物理ファイルおよび論理ファイルの EDTCDE (編集コード) および EDTWRD (編集語) キーワード

これらはフィールド・レベル・キーワードで、定義中のフィールドがあとで表示装置ファイルまたは印刷装置ファイル作成時に参照される場合に、そのフィールドの編集を指定するために使用します。EDTCDE および EDTWRD キーワードは物理および論理ファイルに影響を及ぼすことはありません。

EDTCDE キーワードの形式は次のとおりです。

```
EDTCDE(edit-code [* | floating-currency-symbol])
```

EDTWRD キーワードの形式は次のとおりです。

```
EDTWRD('edit-word')
```

表示装置ファイルに入力可能フィールドを定義するときに、29 桁目に R を指定し、REF または REFFLD キーワードを指定することによって、定義しているフィールドを参照します。表示装置ファイル作成時に、オペレーティング・システムは、EDTCDE または EDTWRD キーワードおよびその他のフィールド属性を、物理ファイルまたは論理ファイルの該当フィールドから表示装置ファイルのフィールドに複写します。表示装置ファイルまたは印刷装置ファイルで新たに編集キーワードを指定すれば、EDTCDE または EDTWRD キーワードを指定変更することができます。表示装置ファイルまたは印刷装置ファイルに DLTEDT キーワードを指定すれば、該当のフィールドに関する編集指定がすべて削除されます。

浮動小数点数フィールド (35 桁目が F) または 16 進数フィールド (35 桁目が H) については、EDTCDE または EDTWRD キーワードを指定することはできません。日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に EDTCDE または EDTWRD キーワードを指定してはなりません。

物理ファイルまたは論理ファイルでのこれらのキーワードの指定に関する規則は、表示装置ファイルの場合と同じです。

### 例

次の例は、物理ファイルの場合の EDTCDE および EDTWRD キーワードの指定方法を示します。

```

|...+.....1....+.....2....+.....3....+.....4....+.....5....+.....6....+.....7....+.....8
A          R RECORD
A
A          PRICE          5 2      EDTCDE(J)
A
A          SALES          7 2      EDTCDE(K $)
A
A          SALARY        8 2      EDTCDE(1 *)
A

```

```

A          BALANCE          7 2      EDTWRD('$ 0. &CR')
A
A          DATE             6 0      EDTCDE(Y)
A

```

この例では、PRICE、SALES、SALARY、および DATE の各フィールドについて編集が指定されています。これらのフィールドを表示装置ファイルまたは印刷装置ファイルで参照する場合には、新たに編集を指定する必要はありません。つまり、上記の各フィールドを参照するアプリケーション相互間で、これらのフィールドの編集が標準化されます。

#### 関連情報

表示装置ファイルの参照 (29 桁目)

表示装置ファイルの EDTCDE (編集コード) キーワード

表示装置ファイルの EDTWRD (編集語) キーワード

## 物理ファイルおよび論理ファイルの FCFO (変更順検索) キーワード

これはファイル・レベル・キーワードで、同じ物理または論理ファイル・メンバーから重複キー値を持つ複数のレコードを検索する場合に、最初にキーの値が変更されたレコードを最初に検索するレコードにするときに使用します。これを変更順検索 (FCFO) と呼びます。

このキーワードにはパラメーターはありません。

FCFO キーワードを、FIFO、LIFO、UNIQUE、または REFACCPH キーワードと同時に指定することはできません。

FCFO、LIFO、FIFO、または UNIQUE をいずれも指定しなかった場合には、重複キー値を持つレコードは先入れ先出し (FIFO)、後入れ先出し (LIFO)、または変更順検索 (FCFO) の順序で検索されますが、どの順序でキーが検索されるかは特定できません。

FCFO キーワードを指定すると、レコードは、レコードのキー値の変更時点の順序で順序付けされます。FIFO キーワードまたは LIFO キーワードの場合は、相対レコード番号に従ってレコードが順序付けされます。

FCFO キーワードを含むファイルには、キー・フィールドを少なくとも 1 つは指定しなければなりません。物理ファイル作成 (CRTPF) または論理ファイル作成 (CRTLF) コマンドで FILETYPE(\*SRC) を指定する場合は、FCFO キーワードは指定できません。

#### 例

次の例は、物理ファイルでの FCFO キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          FCFO
00020A          R CUSREC          TEXT('CUSTOMER RECORD')
00030A          CUSNAMEF          10A
00040A          CUSNAMEM          1A
00050A          CUSNAMEL          10A
00060A          K CUSNAMEL
A

```

## 物理ファイルおよび論理ファイルの FIFO (先入れ先出し) キーワード

これはファイル・レベル・キーワードで、同じ物理または論理ファイル・メンバーから重複キー値を持つレコードを検索する場合に、先入れ先出し順序 (FIFO) で検索することを指定するために使用します。

このキーワードにはパラメーターはありません。

FIFO キーワードは、FCFO、LIFO、UNIQUE、および REFACCPH キーワードと同時に使用することはできません。

FCFO、LIFO、FIFO、または UNIQUE のいずれも指定しなかった場合は、重複キー値を持つレコードは FIFO、LIFO、FCFO の順序で取得されますが、取得されるレコードの順序は保証されません。

FIFO キーワードを含むファイルには、少なくとも 1 つのキー・フィールドを指定しなければなりません。物理ファイル作成 (CRTPF) または論理ファイル作成 (CRTLF) コマンドで FILETYPE(\*SRC) を指定する場合は、FIFO キーワードは指定できません。

## 例

次の例は、物理ファイルでの FIFO キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                FIFO
00020A      R CUSREC                    TEXT('CUSTOMER RECORD')
00030A      CUSNAMEF                    10A
00040A      CUSNAMEM                    1A
00050A      CUSNAMEL                    10A
00060A      K CUSNAMEL
      A
```

## 物理ファイルおよび論理ファイルの FLTPCN (浮動小数点の精度) キーワード

これはフィールド・レベル・キーワードで、浮動小数点数フィールドの精度を指定するために使用します。

キーワードの形式は次のとおりです。

```
FLTPCN(*SINGLE | *DOUBLE)
```

\*SINGLE は単精度、\*DOUBLE は倍精度です。このキーワードは、浮動小数点数フィールド (データ・タイプ F) に対してだけ有効です。

FLTPCN キーワードを指定しなかった場合には、デフォルトによって単精度が使用されます。単精度のフィールドは、最大 9 桁まで指定することができ、倍精度のフィールドは 17 桁まで指定することができます。9 (単精度の場合) または 17 (倍精度の場合) を超えるフィールドの長さを指定すると、エラー・メッセージが送られ、ファイルは作成されません。

## 例

次の例は、FLTPCN キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00090A      FIELDA                    17F 4      FLTPCN(*DOUBLE)
      A
```

この例では、FIELDA は倍精度の浮動小数点数フィールドです。

## 物理ファイルおよび論理ファイルの FORMAT (様式) キーワード

これは、レコード・レベル・キーワードで、以前に定義されたレコード様式のフィールド仕様を、このレコード様式で共用することを指定するために使用します。定義するレコード様式の名前は、以前に定義されたレコード様式の名前と同じでなければなりません。

キーワードの形式は次のとおりです。

FORMAT([library-name/]database-file-name)

database-file-name パラメーターは必須です。これは、定義済みのレコード様式が取り出される物理または論理ファイルの名前です。

library-name はオプションです。ライブラリー名を指定しなかった場合には、ファイル作成時点で有効なライブラリー・リスト (\*LIBL) が使用されます。

FORMAT キーワードを指定した場合には、当該レコード様式についてフィールド仕様を指定することはできません。キー仕様を指定し、また、選択/除外仕様もこのファイルで使用したい場合には、それも指定しなければなりません。(それらは以前に定義したレコード様式と同じでも異なってもかまいません。)

結合論理ファイルでは、FORMAT キーワードは無効です。また、FORMAT キーワードのパラメーター値として、結合論理ファイルを指定することはできません。

レコード様式を共用しているデータベース・ファイルが削除された場合でも、そのレコード様式は、それを使用しているファイルが 1 つでもある間は、存在し続けます。たとえば、FILE2 の RECORD は、FILE1 の RECORD の仕様を共有するため FORMAT キーワードを使用します。ファイルはどちらも作成が完了しています。FILE1 を削除し、別の DDS を用いて作成し直した場合にも、RECORD は FILE2 に依然として存在しており、それは、FORMAT キーワードを使用することにより、他のファイルで、元のレコード様式を得るために参照することができます。

このキーワードで分散データ管理 (DDM) ファイルを指定することはできません。

## 例

次の例は、論理ファイルの場合の FORMAT キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD          PFILE(FILE2)
00020A          A                 FORMAT(FILE1)
```

この例では、この論理ファイルのレコード様式は、ファイル FILE1 ですでに指定されているレコード様式と同じです。このレコード様式の名前 (RECORD) は、FILE1 のレコード様式と同じでなければなりません。

### 関連概念

8 ページの『レコード様式』

17 桁目に R を指定した場合には、19 から 28 桁目に指定した名前はレコード様式名になります。

## JDFTVAL (結合におけるデフォルト値) キーワード - 結合論理ファイルの み

これは結合論理ファイルで使用するファイル・レベル・キーワードで、2 次ファイルとの結合の結果、目的のレコードが得られない場合に、システムがフィールドにデフォルト値を与えることを指定するために使用します。JDFTVAL は、結合論理ファイルに対してだけ有効です。

このキーワードにはパラメーターはありません。

システムのデフォルト値は、文字フィールドおよび 16 進数フィールドの場合にはブランク、数字フィールドの場合にはゼロです。物理ファイルのフィールドに対して DFT キーワードを指定することによって、特定のフィールドのデフォルト値を変更することができます。

JDFTVAL を指定すると、プログラムは、対応するレコードが 2 次ファイルにないレコードも検索します。JDFTVAL の指定がない場合には、2 次ファイルに対応するレコードのない 1 次ファイルのレコードは無視されます。

3 つ以上のファイルを結合しており、結合フィールドとして使用するフィールドに対して JDFTVAL キーワードを指定した場合には、2 次ファイルにレコードがないフィールドのデフォルト値が、フィールドの値と同様の方法で処理されます。たとえば、レコードは、そのデフォルト値に基づいて、選択または除外されます。また、そのフィールドが他の 2 次ファイルに結合するための結合フィールドとして使用されている場合には、他の 2 次ファイルからのレコードが、デフォルト値に基づいてプログラムに戻されます。

## 例

次の例は、JDFTVAL キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                JDFTVAL
00020A                                JFILE(PF1 PF2)
00030A      R RECORD1                  JOIN(PF1 PF2)
00040A      J                          JFLD(NAME NAME)
00050A      NAME                        JREF(1)
00060A      ADDR
00070A      BAL
      A

```

この例では、PF1 が 1 次ファイルで、PF2 が 2 次ファイルです。PF1 および PF2 には、次のレコードがあるものとします。

PF1 NAME	ADDR	PF2 NAME	BAL
Anne	120 1st St.	Anne	5.00
Doug	40 Pillsbury	Doug	6.50
Mark	2 Lakeside Dr.	Sue	2.00
Sue	120 Broadway		

結合論理ファイル内で JDFTVAL を指定してあれば、プログラムは次のレコードを読み取ります (レコードは到着順に示されています)。

NAME	ADDR	BAL
Anne	120 1st St.	5.00
Doug	40 Pillsbury	6.50
Mark	2 Lakeside Dr.	0.00
Sue	120 Broadway	2.00

結合論理ファイル内で JDFTVAL を指定していない場合には、プログラムは 3 つのレコードだけを読み取ります (Mark についてはレコードは見つかりません)。

この例で、JREF(1) の代わりに JREF(2) が指定されていた場合には、プログラムに戻されるレコードは次のように違ってきます。

NAME	ADDR	BAL
Anne	120 1st St.	5.00
Doug	40 Pillsbury	6.50
	2 Lakeside Dr.	0.00
Sue	120 Broadway	2.00

## 関連資料

55 ページの『DFT (デフォルト) キーワード - 物理ファイルのみ』  
これは、フィールド・レベル・キーワードで、フィールドのデフォルト値を指定するために使用します。

## JDUPSEQ (結合における重複レコードの順序付け) キーワード - 結合論理ファイルのみ

これは結合レベル・キーワードで、プログラムが結合論理ファイルを読み取るさいに、値が重複する結合フィールドを持つレコードをどのような順序でプログラムに渡すかを指定するために使用します。

キーワードの形式は次のとおりです。

JDUPSEQ(sequencing-field-name [\*DESCEND])

このキーワードは、固有レコードの順序には影響を及ぼしません。このキーワードを指定しなかった場合には、結合フィールドに重複する値を持つレコードをシステムがプログラムに渡す順序は特定されません。

1 つの結合仕様で複数の JDUPSEQ キーワードを指定した場合には、JDUPSEQ キーワードを指定した順序によって、重複するレコードの検索順序が決まります。これは、重複キーを持つレコードの検索順序を決めるために、さらに別のキー・フィールドの指定を追加する場合に似ています。

このキーワードは、結合論理ファイルに対してだけ有効です。

1 つの結合仕様では、JFLD キーワードで結合先 フィールドとして指定したすべてのフィールドと、JDUPSEQ キーワードで指定したすべてのフィールドの長さの合計が、120 バイトを超えてはなりません。

順序付けフィールド名は、(1) この結合仕様の結合先 ファイルに存在するフィールドであり、かつ、(2) この結合仕様の JFLD キーワードの結合先 フィールドとして指定されていないフィールドでなければなりません。順序付けフィールド名として連結フィールドまたは SST フィールドを指定することもできます。順序付けフィールド名は、結合論理ファイルのレコード様式内に指定する必要はありません。

必要に応じて、\*DESCEND を指定することにより、重複するレコードの検索順序を変更することができます。\*DESCEND を指定しなかった場合には、重複レコードは次のデフォルトの順序で検索されます。

- 順序付けフィールドが数字フィールドである場合には、符号も含めた昇順。
- 順序付けフィールドが文字フィールドである場合には、昇順。

### 例

次の例は、JDUPSEQ キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R JREC                JFILE(PF1 PF2)
00020A          J                    JOIN(PF1 PF2)
00030A          JFLD(NAME1 NAME2)
00040A          JDUPSEQ(PHONE)
00050A          NAME1
00060A          ADDR
00070A          PHONE
```

この例で、PF1 および PF2 に次のレコードがあるものとします。

PF1 NAME1	ADDR	PF2 NAME2	TELEPHONE
Anne	120 1st St.	Anne	555-1111
Doug	40 Pillsbury	Anne	555-6666
Mark	2 Lakeside Dr.	Anne	555-2222



PF1 NAME1	ADDR	PF2 NAME2	TELEPHONE
		Doug	555-5555

PF2 には、Anne について、それぞれ異なる電話番号を示す 3 つのレコードがあります。例のように JDUPSEQ を指定すると、これらのレコードは次の順序で検索されます。

NAME	ADDR	TELEPHONE
Anne	120 1st St.	555-1111
Anne	120 1st St.	555-2222
Anne	120 1st St.	555-6666
Doug	40 Pillsbury	555-5555

JDUPSEQ キーワードがレコードの順序に影響を及ぼすのは、重複がある場合だけであることに注意してください。

次の例では、例 1 と同じ物理ファイルを基にした論理ファイルを想定しています。PF2 には、Anne について、それぞれ異なる電話番号を示す 3 つのレコードがあります。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC      JFILE(PF1 PF2)
00020A      J          JOIN(PF1 PF2)
00030A      JFLD(NAME1 NAME2)
00040A      JDUPSEQ(PHONE *DESCEND)
00050A      NAME1
00060A      ADDR
00070A      PHONE
A

```

JDUPSEQ に \*DESCEND を指定した場合には、レコードは次の順序で検索されます。

NAME1	ADDR	TELEPHONE
Anne	120 1st St.	555-6666
Anne	120 1st St.	555-2222
Anne	120 1st St.	555-1111
Doug	40 Pillsbury	555-5555

リストには、Anne の電話番号が降順に示されています。

## JFILE (結合ファイル) キーワード - 結合論理ファイルのみ

これは、レコード・レベル・キーワードで、定義中の結合論理ファイルを介してアクセスされるデータが入っている物理ファイルを識別するために使用します。

キーワードの形式は次のとおりです。

```
JFILE([library-name/]physical-file-name [..256])
```

このキーワードは、ファイルを結合論理ファイルとして識別する点を除けば、PFILE キーワードと同じです。JFILE キーワードは、PFILE キーワードと同時に定義することはできません。

JFILE キーワードの指定は、結合論理ファイルにおけるレコード・レベルでは必須です。JFILE キーワードは、少なくとも 2 つの物理ファイル名を指定しなければなりません。同じファイル名を 2 回以上指定することもできます。

最初に指定したファイルは 1 次ファイルと呼ばれ、このファイルから結合が開始されます。その他のファイルはすべて、2 次ファイルと呼ばれます。2 次ファイルは 255 まで指定することができます (すなわち、JFILE キーワードには合計 256 個のファイルを指定することができます)。

JFILE キーワードで分散データ管理 (DDM) ファイルが指定できるのは、論理ファイルをリモート・システムで作成している場合だけです。詳しくは、分散データ管理の項を参照してください。

JFILE キーワードに物理ファイルを指定する順序については、次の各事項を考慮してください。

- 各物理ファイルのレコード数が異なっている場合には、JFILE キーワードには、レコード数の少ない物理ファイルから順に (左から右へと) 指定します。1 次ファイルのレコード数は、2 次ファイルのレコード数に等しいか、またはそれより少なくなければなりません。こうすることにより、ファイルを読み取るパフォーマンスが向上します。
- 結合仕様で指定する 1 次ファイルおよび 2 次ファイルは、一定の順序になっていなければなりません。この順序は、JFILE キーワードに指定したファイルの順序により決まります。JOIN (結合) キーワードの説明箇所にある例 3 を参照してください。
- JOIN キーワードおよび JREF キーワードでは、JFILE キーワードで指定したファイルを識別するのに相対ファイル番号を使用することができます。JFILE キーワードで最初に指定したファイルが相対ファイル番号 1 になり、2 番目のファイルが相対ファイル番号 2 になります。以下同様にして、256 までの相対ファイル番号を指定することができます。JOIN キーワードおよび JREF キーワードで、ファイル名の代わりに相対ファイル番号を使用する場合には、JFILE キーワードでのファイルの指定順序が、JOIN キーワードおよび JREF キーワードの指定方法に影響することになります。

注: 物理ファイルの名前が固有でない場合には、相対ファイル番号を指定しなければなりません。

## 例

以下に JFILE キーワードの指定方法の例を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC      JFILE(PF1 PF2)
00020A      J           JOIN(PF1 PF2)
00030A      J           JFLD(NAME1 NAME2)
      A
```

この結合論理ファイルでは、PF1 が 1 次ファイルで、PF2 が 2 次ファイルです。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC      JFILE(MYLIBA/PHYSICAL1 +
00020A      J           MYLIBB/PHYSICAL2 MYLIBC/PHYSICAL3)
00030A      J           JOIN(1 2)
00040A      J           JFLD(FIELD1 FIELD2)
00050A      J           JOIN(1 3)
00060A      J           JFLD(FIELD1 FIELD2)
      A
```

この結合論理ファイルでは、ライブラリー MYLIBA のファイル PHYSICAL1 が 1 次ファイルです。ライブラリー MYLIBB のファイル PHYSICAL2 およびライブラリー MYLIBC のファイル PHYSICAL3 は 2 次ファイルです。

### 関連資料

70 ページの『JOIN (結合) キーワード - 結合論理ファイルのみ』

これは結合レベル・キーワードで、このキーワードを指定した結合仕様により結合される 2 つ (対) のファイルを識別するために使用します。

### 関連情報

## JFLD (結合フィールド) キーワード - 結合論理ファイルのみ

これは結合レベル・キーワードで、結合元 フィールドと結合先 フィールドの名前を識別するために使用します。これらのフィールドの値は、複数の物理ファイルを 1 つの結合論理ファイルで結合するために用いられます。結合元フィールドおよび結合先フィールドは、どちらも結合フィールドと呼ばれています。

キーワードの形式は次のとおりです。

JFLD(from-field-name to-field-name)

結合フィールドは、この結合仕様の JOIN キーワードで指定した物理ファイルのフィールドに対応していなければなりません。JFLD キーワードに指定する名前は、結合論理ファイルで名前を変更した場合を除き、物理ファイルで指定されている名前と同じでなければなりません。JOIN キーワードを指定しなかった場合には、JFILE キーワードが使用されます。

このキーワードは、結合論理ファイルに対してだけ有効です。

1 つの結合仕様には、少なくとも 1 つの JFLD キーワードが必須です。結合仕様は、17 桁目の J により識別されます。結合論理ファイルには少なくとも 1 つの結合仕様が必須なので、結合論理ファイルには少なくとも 1 つの JFLD キーワードを指定しなければなりません。

結合フィールドは、結合論理ファイルのレコード様式のフィールドとして指定する必要はありません。

物理ファイルの結合にさらに多くの結合フィールドを指定する必要がある場合には、複数の JFLD キーワードを指定します。

JFLD キーワードに指定するフィールド名は、対応する結合レコード様式に、または (JFILE キーワードに指定した) 物理ファイルの 1 つに、フィールド・レベルで指定されていなければなりません。

i5/OS オペレーティング・システムは、次の検索順序に従って、結合フィールド名と定義されたフィールドとの突き合わせを行います。

1. 該当の結合論理ファイルのフィールド・レベルの 19 から 28 桁目で指定されているフィールド。

**注:** CONCAT、RENAME、または SST キーワードが指定されているフィールドは、結合フィールドとして使用することができます。しかし、CONCAT、RENAME、または SST キーワードの中で指定されているフィールドは、結合フィールドとして使用することはできません。

2. JOIN キーワードで指定した物理ファイルのフィールド。

結合フィールドを指定する場合の規則は次のとおりです。

- 結合元 フィールドは、JOIN キーワードで指定した結合元 ファイルのフィールドでなければなりません。
- 結合先 フィールドは、JOIN キーワードで指定した結合先 ファイルのフィールドでなければなりません。
- 結合フィールドは、結合レコード様式に定義する必要はありません。
- 結合元 および結合先 フィールドは、同じフィールド属性 (長さ、データ・タイプ、および小数点以下の桁数) を持っていないければなりません。同じ名前である必要はありません。結合フィールドの定義が物理ファイル相互間で異なっている場合には、一方または両方のフィールドを再定義しなければなりません。フィールドを再定義すると、データ変換エラーが起こることがあります。

注: 文字フィールドは同じ長さである必要はありません。短い方の結合フィールドにはブランクが埋め込まれて、長い方の結合フィールドと同じ長さになります。

- 1 つの結合仕様において、JFLD キーワードの結合先 フィールドとして指定するすべてのフィールドと、JDUPSEQ キーワードで指定するすべてのフィールドの合計長は、120 バイト以下でなければなりません。
- 2 進文字フィールドは、別の 2 進文字フィールドとのみ結合できます。

## 例

以下に JFLD キーワードの指定方法の例を示します。

### 例 1

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R JREC          JFILE(PF1 PF2)
00020A          J              JOIN(PF1 PF2)
00030A          JFLD(NAME1 NAME2)
A
```

この結合論理ファイルでは、JFLD キーワードによって、物理ファイル PF1 の NAME1 が物理ファイル PF2 の NAME2 と結合されることが指定されています。

### 例 2

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R JREC          JFILE(PF1 PF2)
00020A          J              JOIN(PF1 PF2)
00030A          JFLD(NAME1 NAME2)
00040A          JFLD(ADDR1 ADDR2)
A
```

この結合論理ファイルでは、JFLD キーワードによって、物理ファイル PF1 の NAME1 および ADDR1 が物理ファイル PF2 の NAME2 および ADDR2 と結合されることが指定されています。

### 関連資料

26 ページの『物理ファイルおよび論理ファイルの長さ (30 から 34 桁目)』

この欄は、物理ファイル・フィールドまたは論理ファイル・フィールドの長さを指定するのに使用します。

29 ページの『物理ファイルおよび論理ファイルのデータ・タイプ (35 桁目)』

物理ファイルでは、この欄にはデータベース内でのフィールドのデータ・タイプを指定します。論理ファイルの場合にこの欄にデータ・タイプを指定するのは、この論理ファイルの基礎となっている物理ファイル内の対応するフィールドのデータ・タイプを指定変更または修正したい場合に限りです。

32 ページの『物理ファイルおよび論理ファイルの小数点以下桁数 (36 および 37 桁目)』

この欄には、パック 10 進数フィールド、ゾーン 10 進数フィールド、2 進数フィールド、または浮動小数点数フィールドにおける小数点の位置を指定します。

34 ページの『物理ファイルおよび論理ファイルの使用目的 (38 桁目)』

この欄は、名前を付けたフィールドを、入力専用にするか、入出力共用フィールドにするか、あるいは非入出力フィールドにするかを指定するのに使用します。

## JOIN (結合) キーワード - 結合論理ファイルのみ

これは結合レベル・キーワードで、このキーワードを指定した結合仕様により結合される 2 つ (対) のファイルを識別するために使用します。

キーワードの形式は次のとおりです。

JOIN(from-file to-file)

このキーワードは、結合論理ファイルに対してだけ有効です。

結合するファイルを指定するには、ファイル名または相対ファイル番号のいずれかを使用することができます。ただし、JFILE キーワードで同じファイルを複数回指定した場合には、相対ファイル番号を指定しなければなりません。

ファイル名を指定する場合には、JFILE キーワードに一度しか指定されていないファイルだけを選んで指定しなければなりません。また、各 JFILE キーワードでは、結合元 ファイルが、結合先 ファイルよりも前に指定されなければなりません。

相対番号を指定する場合、各番号は JFILE キーワードに指定した個々のファイルを表します。以下の値は、有効な値です。

ファイル

有効な値

結合元ファイル番号

1 から 255

結合先ファイル番号

2 から 256

結合元ファイル の番号は、必ず、結合先ファイル の番号より小さくなければなりません。

結合元 ファイルおよび結合先 ファイルの指定順序には、特別な規則があります。詳しくは、このトピックの例 3 を参照してください。

1 つの結合論理ファイル内では、各 2 次ファイルはそれぞれ一度だけ結合先 ファイルとして使用することができます。

## 物理ファイルおよび論理ファイルの結合仕様

結合仕様を指定するには、次のようにします。

1. レコード・レベルの直後 (19 から 28 桁目の欄に最初のフィールド名を指定する前) の行の 17 桁目に J を指定します。17 桁目の J は、結合仕様の始めを示します。
2. JOIN キーワードを指定します。JOIN キーワードは、JFILE キーワードに指定したファイルが 2 つだけの場合にはオプションです。JFILE キーワードに 3 つ以上の物理ファイルを指定した場合には、各 2 次ファイルについて 1 つの JOIN キーワードが必要です。
3. 各結合仕様に対して、少なくとも 1 つの JFLD キーワードを指定します。
4. 17 桁目に他の結合仕様の始めを示す J を指定するか、あるいは、19 から 28 桁目にフィールド名を指定すると、それが 1 つの結合仕様の終わりを示すものとなります。

JFILE キーワードで指定した 2 次ファイルのそれぞれについて、結合仕様が 1 つずつ必要です。したがって、結合論理ファイルには、少なくとも 1 つは結合仕様が含まれていなければなりません。

JOIN キーワードは、1 つの結合仕様で一度しか指定することができません。

## 例

以下に JOIN キーワードの指定方法の例を示します。

### 例 1

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R RECORD1      JFILE(PFA PFB PFC)
00020A      J              JOIN(PFA PFB)
00030A      J              JFLD(NAME1 NAME2)
00040A      J              JOIN(PFA PFC)
00050A      J              JFLD(NAME1 NAME3)
00060A      NAME1
A
```

この例では、PFA は PFB に結合され、PFC にも結合されます。

### 例 2

次の例は、相対ファイル番号を使用して JOIN を指定する方法を示します。

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R RECORD1      JFILE(PFA PFB PFC)
00020A      J              JOIN(1 2)
00030A      J              JFLD(NAME1 NAME2)
00040A      J              JOIN(1 3)
00050A      J              JFLD(NAME1 NAME3)
00060A      NAME1
A
```

例 2 は例 1 と同じことを意味しています。PFA は、JFILE キーワードに指定された最初の物理ファイルであり、相対ファイル番号 1 を持っています。PFB および PFC は、JFILE キーワードに指定された 2 番目のファイルと 3 番目のファイルです。PFB の相対ファイル番号は 2 で、PFC の相対ファイル番号は 3 です。

### 例 3

次の例は、関連する物理ファイルの順序を示します。

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A      R J3          JFILE(VENDORS PARTS PARTWARE +
00020A      1            WAREHOUSE
00030A      J              JOIN(1 2) 2
00040A      J              JFLD(VNBR VNUM)
00050A      J              JOIN(2 3) 3
00060A      J              JFLD(PNBR PNBR)
00070A      J              JOIN(3 4) 3
00080A      J              JFLD(WNBR WNBR)
00090A      VNAME
00100A      VAD1
00110A      VAD2
00120A      PNBR          JREF(2)
00130A      WNBR          JREF(4)
00140A      BIN
00150A      QOH
A
```

この例の結合論理ファイルは 4 つの物理ファイルを基礎としています。JFILE キーワードで最初に指定されている VENDORS ファイルは 1 次ファイルで、その相対ファイル番号は 1 です。PARTS ファイル、PARTWARE ファイル、および WAREHOUSE ファイルは 2 次ファイルです。PARTS の相対ファイル番号は 2 で、PARTWARE の相対ファイル番号は 3 で、WAREHOUSE の相対ファイル番号は 4 です。

JOIN キーワードで指定されている番号の順序に注意してください。

- 1 最初の JOIN キーワードの最初のパラメーター値 (最初の結合元 ファイル) は、1 次ファイルでなければなりません。
- 2 JOIN キーワードに指定する 2 番目のパラメーター値 (結合先 ファイル) は、JFILE キーワード上での 2 次ファイルの指定順序に従っていなければなりません。相対ファイル番号の代わりにファイル名を指定するのであれば、その順序は次のようになります。

```
J JOIN(VENDORS PARTS)
J JOIN(PARTS PARTWARE)
J JOIN(PARTWARE WAREHOUSE)
```
- 3 JOIN キーワードでは、結合元 ファイルおよび結合先 ファイルを昇順で指定しなければなりません。

注: ファイルは、結合元 ファイルとして 2 回以上指定することができます。たとえば、上記の JOIN キーワードのパラメーターは次のように指定することができます。

```
J JOIN(1 2)
J JOIN(2 3)
J JOIN(2 4)
```

ただし、ファイルは、結合先 ファイルとしては 1 回だけしか指定できません。

#### 関連資料

7 ページの『物理ファイルおよび論理ファイルの名前または仕様のタイプ (17 桁目)』

この欄には名前のタイプ、または論理ファイルの場合は仕様のタイプを指定するための値を入力します。名前タイプを指定すると、名前は 19 から 28 桁目で指定されます。

67 ページの『JFILE (結合ファイル) キーワード - 結合論理ファイルのみ』

これは、レコード・レベル・キーワードで、定義中の結合論理ファイルを介してアクセスされるデータが入っている物理ファイルを識別するために使用します。

## JREF (結合における参照) キーワード - 結合論理ファイルのみ

これは結合論理ファイルのフィールド・レベル・キーワードで、名前が複数の物理ファイルで指定されているフィールドについて使用します。このキーワードは、そのフィールドがどの物理ファイルに含まれているかを示します。

キーワードの形式は次のとおりです。

```
JREF(file-name | relative-file-number)
```

このキーワードには、物理ファイル名またはその相対ファイル番号のいずれかを指定することができます。JFILE キーワードで同じ物理ファイルの名前を 2 回指定した場合には、相対ファイル番号を指定しなければなりません。相対ファイル番号は、JFILE キーワードで指定した物理ファイルに対応します。たとえば、JREF(1) を指定すると、フィールドは、JFILE キーワードに指定した最初の物理ファイルに関連付けられます。JREF(2) を指定すると、フィールドは、JFILE キーワードに指定した 2 番目の物理ファイルに関連付けられます。このトピックの例 2 を参照してください。

このキーワードは結合論理ファイルに対してだけ有効です。

結合論理ファイルは、複数 (最高 256 個) の物理ファイルを基にしています。結合論理ファイルのレコード様式で指定する各フィールド名は、いずれも、その結合論理ファイルの基礎となっている物理ファイルのフィールドの 1 つを固有に識別するものでなければなりません。たとえば、結合論理ファイルが 2 つの物理ファイルを基礎としており、各物理ファイルに NAME という名前のフィールドがある場合には、そのフィールドがどちらの物理ファイルのものであるかを識別するために、JREF キーワードを指定しなければなりません。

物理ファイル (JFILE キーワードで指定されたもの) の間でフィールド名が固有な場合には、このキーワードの指定はオプションです。たとえば、結合論理ファイルが 2 つの物理ファイルに関連付けられており、物理ファイルのどちらか 1 つにだけフィールド NAME1 が入っている場合には、JREF キーワードを指定する必要はありません。

結合論理ファイルがただ 1 つの物理ファイルに関連付けられている場合 (JFILE キーワードで同じファイル名を 2 回指定している場合) には、フィールドごとに JREF キーワードを指定しなければなりません。

### 例 1

以下に JREF キーワードの指定方法の例を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JOINREC      JFILE(PFA PFB PFC)
00020A      :
00030A      :
00040A      :
00050A      NAME          JREF(PFB)
      A
```

この例では、JREF キーワードはファイル名を伴って指定されており、NAME は PFA と PFB の両方に入っています。JREF (PFB) の指定により、このフィールドを PFB と関連付けています。

### 例 2

次の例は、相対ファイル番号を使用して JREF を指定する方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JOINREC      JFILE(PFA PFB PFC)
00020A      :
00030A      :
00040A      :
00050A      NAME          JREF(2)
      A
```

例 2 は例 1 と同じことを意味しています。例 2 では、NAME は PFA と PFB の両方に入っています。JREF(2) の指定により、このフィールドを PFB (JFILE キーワードで指定した 2 番目の物理ファイル) に関連付けています。

## 物理ファイルおよび論理ファイルの LIFO (後入れ先出し) キーワード

これはファイル・レベル・キーワードで、同一の物理ファイル・メンバーから重複キー値を持つ複数のレコードを検索する場合に、後入れ先出し (LIFO) の順序で検索することを指定するために使用します。

このキーワードにはパラメーターはありません。

LIFO は、FCFO、FIFO、UNIQUE または REFACCPH キーワードと同時に使用できません。

変更順検索 (FCFO)、先入れ先出し法 (FIFO)、LIFO、または UNIQUE のいずれも指定しなかった場合は、重複キー値を持つレコードは FIFO、LIFO、FCFO の順序で取得されますが、取得されるレコードの順序は保証されません。

LIFO キーワードの指定があるファイルには、少なくとも 1 つのキー・フィールドを指定しなければなりません。物理ファイル作成 (CRTPF) または論理ファイル作成 (CRTLF) コマンドで FILETYPE(\*SRC) を指定する場合は、LIFO キーワードは指定できません。



## 例

次の例は、物理ファイルでの LIFO キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                LIFO
00020A      R CUSREC                    TEXT('CUSTOMER RECORD')
00030A          CUSNAMEF                10A
00040A          CUSNAMEM                 1A
00050A          CUSNAMEL                10A
00060A      K CUSNAMEL
      A
```

## 物理ファイルおよび論理ファイルの NOALTSEQ (非代替照合順序) キーワード

これはキー・フィールド・レベル・キーワードで、ファイル・レベルで指定された ALTSEQ キーワードをこのキー・フィールドには適用しないことを指定するために使用します。キー・フィールドに ABSVAL または SIGNED を指定する場合は、そのキー・フィールドに NOALTSEQ キーワードが指定されているかどうかにかかわらず、自動的に NOALTSEQ が効力を持つこととなります。

このキーワードにはパラメーターはありません。

## 例

次の例は、NOALTSEQ キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                ALTSEQ(TABLELIB/TABLE1)
00020A      R DSTR
00030A          :
00040A          :
00050A          CODE                    1
00060A          NAME                    20
00070A          :
00080A          :
00090A      K CODE
00100A      K NAME                      NOALTSEQ
      A
```

レコード様式 DSTR のレコードが、複合キー CODE および NAME に従って順序付けされます。CODE は代替照合順序 (TABLELIB の TABLE1) により順序付けされます。NAME は EBCDIC 照合順序により順序付けされます。NOALTSEQ が指定されているために、NAME フィールドの順序は変更されません。

## PFIL (物理ファイル) キーワード - 論理ファイルのみ

これはレコード・レベル・キーワードで、定義中のレコード様式を介してアクセスされるデータが入っている 1 つまたは複数の物理ファイルを指定するために使用します。

キーワードの形式は次のとおりです。

```
PFIL([library-name/]physical-file-name [.32])
```

PFIL キーワードは、単一樣式論理ファイルまたは複数様式論理ファイルのすべてのレコード様式について必須です。このキーワードは、このファイルを単一樣式論理ファイルまたは複数様式論理ファイルとして識別する点を除けば、JFIL キーワードと同じです。PFIL キーワードは、JFIL キーワードと同時に指定することはできません。1 つの論理ファイルでは、PFIL キーワードを用いて最高 32 個の物理ファイル名を指定することができます。最大数の物理ファイルを使用する場合には、1 つのレコード様式で (つまり PFIL キーワードを 1 つだけ使用して) 32 個の物理ファイル名を指定することもできますし、32 個の

レコード様式に 1 つずつ物理ファイル名を指定することもできます。あるいは、複数のレコード様式に適当に分散させて物理ファイル名を指定することもできます (各レコード様式に指定する物理ファイルの数は同数である必要はありません)。どの方法をとっても、指定できる物理ファイル名は最高 32 個です。論理ファイル作成時に複数の物理ファイルを指定する場合の制約事項については、該当の高水準言語の解説書を参照してください。

各物理ファイル名について、ライブラリー名はオプションです。ライブラリー名を指定しなかった場合には、ファイル作成時点で有効なライブラリー・リスト (\*LIBL) が使用されます。

複数様式論理ファイルの 1 つのレコード様式に複数の物理ファイル名を指定する場合には、その論理ファイルのレコード様式のすべてのフィールドは、指定するすべての物理ファイルに入っていない限りなりません。この種のファイルは、RPG で外部記述することはできません。なぜなら、様式名の重複を招く結果となるためです。PFILE キーワードに指定された物理ファイルが複数の場合に、そのうちの一部の物理ファイルだけに入っているフィールドをプログラムでアクセスするためには、次のいずれかの作業を行います。

- 結合論理ファイルを指定する。この方法の場合には、PFILE キーワードではなく、JFILE キーワードを使用します。
- 論理ファイルに、最初の物理ファイルに入っていないフィールドを含む別のレコード様式を指定する。

たとえば、FLD1 と FLD2 は物理ファイル PF1、PF2、および PF3 のすべてに入っているが、FLD3 は PF3 だけに入っているという場合には、PF1 および PF2 を基礎とする論理ファイル・レコード様式で FLD3 を指定することはできません。FLD3 にアクセスしたい場合には、FLD3 を含む別のレコード様式を指定するようにするか、または結合論理ファイルを使用します。

単一样式または複数様式の論理ファイルを用いて、異なる物理ファイルのフィールドを 1 つのレコード様式に組み込むことはできません。これを行うには、結合論理ファイルを使用します。単一样式または複数様式の論理ファイルの 1 つのレコード様式を介して読み取るレコードに含めることができるのは、1 つの物理ファイルから取り出したデータだけです。また、論理ファイルの 1 つのレコード様式を介して書き込むレコードは、1 つの物理ファイルにだけ保管されます。

PFILE キーワードで分散データ管理 (DDM) ファイルが指定できるのは、論理ファイルをリモート・システムで作成している場合だけです。

## 例

以下に PFILE キーワードの指定方法の例を示します。

### 例 1

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A          R LOGRCD1          PFILE(PF1)
  A
```

この例では、LOGRCD1 は PF1 の中にあるフィールドだけを使用できます。

### 例 2

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
00010A          R LOGRCD2          PFILE(PF1 PF2)
  A              :
  A              :
00020A          R LOGRCD3          PFILE(PF1 PF2 PF3)
  A              :
  A              :
  A              :
```

この例では、LOGRCD2 は PF1 と PF2 に共通のフィールドを使用しなければならず、LOGRCD3 は PF1、PF2、および PF3 に共通のフィールドを使用しなければなりません。

### 例 3

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R LOGRCD4      PFILE(PF1)
  A          :
  A          :
00020A      R LOGRCD5      PFILE(PF2)
  A          :
  A          :
00030A      R LOGRCD6      PFILE(LIB1/PF6)
  A

```

この例では、LOGRCD4、LOGRCD5、および LOGRCD6 は、固有のフィールドを持つことができます。LOGRCD6 には、修飾物理ファイル名が指定されています。

## 物理ファイルおよび論理ファイルの RANGE (範囲) キーワード

このキーワードは、フィールド・レベル、選択または除外フィールド・レベル (またはその両方) で指定します。

キーワードの形式は次のとおりです。

RANGE(low-value high-value)

### 例 1

次の例は、RANGE キーワードに文字および数値ストリングを指定する方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD      PFILE(PF1)
  A
00020A      FIELD A       1 0  RANGE(2 5)  1
00030A      FIELD B       1  RANGE('2' '5')
00040A      FIELD C
00050A      K FIELD D
00060A      S FIELD A     RANGE(1  4)  2
  A

```

この例では、RANGE (1) は、FIELD A および FIELD B を参照する表示装置ファイルについての妥当性検査キーワードとして、FIELD A および FIELD B に対して指定されています。表示装置ファイルでは、この RANGE は、FIELD A または FIELD B に 2、3、4、または 5 だけを入力することをワークステーション・ユーザーに要求しています。FIELD A は数字フィールドで、FIELD B は文字フィールドです。ユーザーが指定するフィールドのタイプは、プログラムの作成に使用されている高水準言語により異なります。

RANGE (2) は、FIELD A についての選択/除外キーワードとして指定されています。FIELD A の値が 1、2、3、または 4 の場合にだけ、基礎となる物理ファイル PF1 からのレコードが、この論理ファイルのレコード様式を介して検索されます。

### 例 2

次の例では、16 進数文字ストリングを使用して RANGE キーワードを指定しています。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RCD1        PFILE(PF1)
00020A      CODE A
00030A      FLD1

```

```
00040A          FLD2
00050A          K FLD1
00060A          S CODEA          RANGE(X'51' X'54')
A
```

この例では、RANGE は CODEA (1 バイトのフィールド) に対する選択/除外キーワードとして指定されています。フィールド CODEA の値が 16 進数の 51 から 54 の範囲内にある場合にだけ、物理ファイル PF1 からのレコードがこのレコード様式を介して検索されます。

## フィールド・レベルの RANGE の指定

フィールド・レベルでは、このキーワードは、定義中のフィールドがあとで表示装置ファイルの作成時に参照される場合に、そのフィールドに対して行われる妥当性検査を指定するために使用します。

RANGE が定義中の物理および論理ファイルに影響を及ぼすことはありません。表示装置ファイルで入力可能フィールドを定義する場合に、29 桁目に R を指定し、REF キーワードまたは REFFLD キーワードを指定することによって、定義するフィールドを参照することができます。表示装置ファイル作成時に、オペレーティング・システムは、RANGE キーワードおよびその他のフィールド属性を、物理ファイルおよび論理ファイルの該当フィールドから表示装置ファイルのフィールドに複写します。RANGE キーワード (およびその他のすべての妥当性検査キーワードならびに CHKMSGID キーワード) は、表示装置ファイルのフィールドに新たに妥当性検査キーワードを指定することによって指定変更することができます。

物理ファイルまたは論理ファイルでのこのキーワードの指定に関する規則は、表示装置ファイルの場合と同じです。

RANGE キーワードは、浮動小数点フィールド (35 桁目が F) または 16 進数フィールド (35 桁目が H) には指定できません。また、日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に RANGE キーワードを指定してはなりません。

### 関連情報

表示装置ファイルの参照 (29 桁目)

表示装置ファイルの RANGE (範囲) キーワード

## 選択/除外フィールド・レベルの RANGE の指定

選択/除外フィールド・レベルでは、このキーワードは、選択/除外フィールドの指定されたレコード様式を使用する入力命令がユーザーのプログラムから送られた場合に、物理ファイルから検索されたレコードを選択または除外するために使用されます。

この場合には次の規則が適用されます。

- 定義しているフィールドが文字フィールドの場合には、文字ストリングまたは 16 進数文字ストリングを指定しなければなりません。

文字ストリングは単一引用符で囲んで指定します (77 ページの『物理ファイルおよび論理ファイルの RANGE (範囲) キーワード』トピックの例 1 を参照)。

16 進数文字ストリングは、X の後に、数字 (0 から 9) と文字 (A から F) の組み合わせを単一引用符で囲んで指定します。単一引用符内の 16 進数字の数は、指定したフィールドの長さの 2 倍でなければなりません (77 ページの『物理ファイルおよび論理ファイルの RANGE (範囲) キーワード』トピックの例 2 を参照)。

- 定義しているフィールドが数字フィールドの場合には、数値 (単一引用符で囲まない 0 から 9 の数字) を指定しなければなりません。77 ページの『物理ファイルおよび論理ファイルの RANGE (範囲) キーワード』トピックの例 1 を参照してください。

- 定義しているフィールドが日付フィールドの場合には、DATFMT キーワードで指定されているのと同じ形式で有効な日付を指定し、DATSEP キーワードで指定されているのと同じ区切り記号を使用してください。たとえば、DATFMT で \*MDY が指定され、DATSEP で「/」が指定されている場合、RANGE('12/15/91' '12/31/91') は有効な値です。
- 定義しているフィールドが時刻フィールドの場合には、TIMFMT キーワードで指定されているのと同じ様式で有効な時刻を指定し、TIMSEP キーワードで指定されているのと同じ区切り記号を使用してください。たとえば、TIMFMT で \*ISO が指定されている場合、RANGE('11.00.00' '12.00.00') は有効な値です。\*ISO のデフォルトの区切り記号はピリオド (.) です。
- 定義しているフィールドが時刻スタンプ・フィールドの場合には、次の形式でデフォルト値を指定しなければなりません。

```
RANGE('YYYY-MM-DD-HH.MM.SS.UUUUUU' 'YYYY-MM-DD-HH.MM.SS.UUUUUU')
```

## REF (参照) キーワード - 物理ファイルのみ

これはファイル・レベル・キーワードで、フィールド記述が検索されるファイルの名前を指定するために使用します。

キーワードの形式は次のとおりです。

```
REF([library-name/]database-file-name [record-format-name])
```

REF は、以前に定義されたレコード様式からフィールド属性を取り出して適用するためのものです。各フィールド記述で同じファイルを参照する場合には、それぞれのフィールドに REFFLD キーワードを指定するのではなく、REF キーワードでファイル名を 1 回だけ指定してください。複数のファイルを参照する場合には、REFFLD キーワードを使用します。REF キーワードは 1 回しか指定できません。

このキーワードでは、database-file-name は必須のパラメーター値です。library-name および record-format-name はオプションです。

ライブラリー名を指定しなかった場合には、ファイル作成時点で有効なライブラリー・リスト (\*LIBL) が使用されます。複数のレコード様式がある場合には、このキーワードのパラメーター値としてレコード様式名を指定してください。レコード様式名を指定しなかった場合には、各レコード様式が順番に検索され、最初に出てきた該当フィールド名が使用されます。

このキーワードでは、分散データ管理 (DDM) ファイルを指定することもできます。

DDM ファイルを使用する場合は、データベース・ファイル名およびライブラリー名は、ソース・システム上の DDM ファイル名およびライブラリー名です。また、レコード様式名は、ターゲット・システム上のリモート・ファイル内のレコード様式名です。

注: 対話式データ定義ユーティリティ (IDDU) ファイルは、参照ファイルとしては使用できません。

### 例

次の例は、REF キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                REF(FILE1)
00020A          R RECORD
00030A          FLD1      R
      A
```

この例では、FLD1 の属性は、FILE1 の最初の (または唯一の) FLD1 と同じになります。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                REF(LIB1/FILE1 RECORD2)
00020A          R RECORD
00030A          FLD1          R
      A

```

この例では、FLD1 の属性が、LIB1 の FILE1 の RECORD2 の FLD1 の属性と同じになります。

#### 関連資料

25 ページの『物理ファイルおよび論理ファイルの参照 (29 桁目)』  
この桁は、物理ファイルへの参照を指定する場合にのみ使用します。

#### 関連情報

DDS ファイルの REF キーワードおよび REFFLD キーワードの指定方法

## REFACCPATH (アクセス・パス定義参照) キーワード - 論理ファイルのみ

これは、ファイル・レベル・キーワードで、この論理ファイルに関するアクセス・パス情報を他の物理ファイルまたは論理ファイルから複製することを指定するために使用します。

アクセス・パス情報には、キー情報、選択および除外情報、代替照合順序情報、動的選択情報、およびキーの順序付け情報 (FCFO、FIFO、LIFO、および UNIQUE キーワードで指定される) が含まれます。

キーワードの形式は次のとおりです。

```
REFACCPATH([library-name/]database-file-name)
```

アクセス・パスを定義するファイルの名前が、このキーワードのパラメーター値になります。

REFACCPATH キーワードを含むファイルには、キー・フィールド、選択フィールド、または除外フィールドを含めることはできません。

定義中のファイルのレコード様式には、この論理ファイルの基礎となる物理ファイル中のレコード様式より少ないフィールドを入れることも、多くのフィールドを入れることもできます。

REFACCPATH キーワードに指定されたファイルが単一樣式または複数様式の論理ファイルの場合には、指定されたファイルおよび REFACCPATH キーワードを含んでいるファイルは、PFILE キーワードと同じ順序で指定されている同じ物理ファイルに基づいていなければなりません。

REFACCPATH キーワードは、結合論理ファイルには使用できません。次に示す条件のすべてに該当する場合に限り、結合論理ファイルを REFACCPATH キーワードのパラメーター値として指定することができます。

- 作成中のファイルが単一樣式論理ファイルである。
- PFILE キーワードで指定した物理ファイルが、該当の結合論理ファイルの JFILE キーワードで最初に指定されているファイルである。
- 該当の結合論理ファイルにキー・フィールドが指定されており、選択/除外フィールドは指定されていない。

このキーワードで分散データ管理 (DDM) ファイルを指定することはできません。

REFACCPATH キーワードは、DYNSTLT、ALTSEQ、FCFO、FIFO、LIFO、または UNIQUE キーワードと同時に指定することはできません。

## 例

次の例は、REFACCPH キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A* ORDER HEADER LOGICAL FILE (ORDHDR11)
00040A                                REFACCPH(DSTLIB/ORDHDRL)
00050A      R ORDHDR                    PFILE(ORDHDRP)
      A
```

## REFFLD (参照フィールド) キーワード - 物理ファイルのみ

これはフィールド・レベル・キーワードで、次の 3 つのいずれかの条件のもとでフィールドを参照するために使用します。

条件は次のとおりです。

- 参照フィールドの名前が、19 から 28 桁目の名前と異なる場合。
- 参照フィールドの名前は 19 から 28 桁目の名前と同じであるが、参照フィールドのレコード様式、ファイル、またはライブラリーが、REF キーワードでの指定と異なる場合。
- 参照フィールドが参照元フィールドと同じ DDS ソース・ファイル内にある場合。

キーワードの形式は次のとおりです。

```
REFFLD([record-format-name/]referenced-field-name
[/*SRC | [library-name/]database-file-name]])
```

referenced-field-name は、定義中のフィールドの名前と同じであっても必須です。record-format-name は、参照ファイルに複数のレコード様式がある場合に指定します。参照されるフィールドが、定義中のフィールドと同じ DDS ソース・ファイル内にある場合には、(database-file-name ではなく) \*SRC を使用してください。\*SRC は、database-file-name および library-name を指定しなかった場合のデフォルト値です。

**注:** 同じ DDS ソース・ファイル内のフィールドを参照する場合、参照されるフィールドは、定義中のフィールドより前に置かれていなければなりません。

特定のデータベース・ファイルを検索する場合は、データベース・ファイル名 (および、必要であれば、そのライブラリー名) を指定します。

29 桁目は R でなければなりません。参照されるフィールドで指定されるキーワードの一部は、定義中のフィールドには組み込みません。詳しくは、物理ファイルおよび論理ファイルの参照 (29 桁目) の項を参照してください。

同じ DDS ソース・ファイルの中で、ファイル・レベルで REF を指定し、フィールド・レベルで REFFLD を指定した場合には、REFFLD の指定の方が優先されます。検索順序は、REF キーワードと REFFLD キーワードの両方の影響を受けます。

このキーワードでは、分散データ管理 (DDM) ファイルを指定することもできます。

DDM ファイルを使用する場合は、データベース・ファイル名およびライブラリー名は、ソース・システム上の DDM ファイル名およびライブラリー名です。参照フィールド名およびレコード様式名は、ターゲット・システム上のリモート・ファイル内のフィールド名およびレコード様式名です。

**注:** 対話式データ定義ユーティリティ (IDDU) ファイルは、参照ファイルとしては使用できません。

## 例

次の例は、REFFLD キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R FMAT1
00020A          ITEM          5
00030A          ITEM1        R          REFFLD(ITEM)
00040A          ITEM2        R          REFFLD(FMAT1/ITEM)
00050A          ITEM3        R          REFFLD(ITEM FILEX)
00060A          ITEM4        R          REFFLD(ITEM LIBY/FILEX)
00070A          ITEM5        R          REFFLD(FMAT1/ITEM LIBY/FILEX)
00080A          ITEM6        R          REFFLD(ITEM *SCR)
      A
```

REF キーワードが指定されていないので、行 00030 および 00040 では、デフォルト値を取り、これらが指定されている DDS ソース・ファイルの検索が行われます。行 00080 では、パラメーター \*SRC によりこのソース・ファイルが明示指定されています。

### 関連資料

25 ページの『物理ファイルおよび論理ファイルの参照 (29 桁目)』  
この桁は、物理ファイルへの参照を指定する場合にのみ使用します。

### 関連情報

DDS ファイルの REF キーワードおよび REFFLD キーワードの指定方法

## 物理ファイルおよび論理ファイルの REFSHIFT (シフト参照) キーワード

これはフィールド・レベル・キーワードで、表示装置ファイルまたは (DFU) 操作でフィールドが参照される場合に、そのフィールドのキーボード・シフトを指定するために使用します。

キーワードの形式は次のとおりです。

REFSHIFT(keyboard-shift)

表示装置ファイルに入力可能フィールドを定義するときに、29 桁目に R を指定し、REF または REFFLD キーワードを指定することによって、定義しているフィールドを参照します。表示装置ファイル作成時に、オペレーティング・システムは、REFSHIFT キーワードおよびその他のフィールド属性を、論理ファイルの該当フィールドから表示装置ファイルのフィールドに複写します。新たに編集キーワードを指定することによって、表示装置ファイルまたは印刷装置ファイルに指定されている編集を指定変更できます。DLTEDIT キーワードを指定して、フィールドについての編集をすべて削除することができます。

このキーワードで指定したパラメーターの値が、データベース・ファイルに指定されているデータ・タイプに代わって、表示装置ファイルのキーボード・シフトの値 (35 桁目) になります。物理ファイルまたは論理ファイルにおいて、REFSHIFT キーワードが指定されたフィールドを参照した場合には、その REFSHIFT キーワードが、表示装置ファイルの新しいフィールドに複写されます。ただし、新しいフィールドに指定されたフィールド属性 (データ・タイプなど) が REFSHIFT キーワードで指定したキーボード・シフトと矛盾している場合には、このキーワードは無視されます。

このキーワードは、データ・タイプ A、S、B、または P のフィールドに対して有効です。パラメーター値として、データ・タイプと矛盾しないキーボード・シフトの値を選択してください。各データ・タイプに対応する有効なパラメーターは次のとおりです。

- 文字フィールド (A) : REFSHIFT(A | X | W | N | I | D | M)
- 数字フィールド (S、B、P): REFSHIFT(S | Y | N | I | D)



## 例

次の例は、物理ファイルでの REFSHIFT キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD
00020A          FIELDA          5          REFSHIFT(X)
00030A          FIELDN          4P         REFSHIFT(N)
      A
```

示されているように、ファイル (FILE1) のフィールド FIELDA および FIELDN に REFSHIFT キーワードが指定されています。REFSHIFT キーワードは、これらのフィールドが表示装置ファイルから参照された場合に使用されます。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          REF(FILE1)
00020A          R RECORD
00030A          FIELDA  R          1  2
      A          FIELDN  R          2  2
      A
```

この例では、表示装置ファイルで FILE1 が参照されています (REF キーワード)。この表示装置ファイル内のフィールド FIELDA および FIELDN では、FILE1 中の FIELDA および FIELDN が参照されます。FILE1 のフィールドには REFSHIFT キーワードが指定されているので、表示装置ファイルでは、その REFSHIFT キーワードで指定されているキーボード・シフトが使用され、各フィールドの属性は次のようになります。

- FIELDA については、35 桁目のキーボード・シフト属性は X。
- FIELDN については、35 桁目のキーボード・シフト属性は N。

### 関連情報

表示装置ファイルの参照 (29 桁目)

表示装置ファイルのデータ・タイプ / キーボード・シフト (35 桁目)

## RENAME (名前変更) キーワード - 論理ファイルのみ

これはフィールド・レベルのキーワードで、定義している論理レコード様式内のフィールド名を対応する物理ファイルのフィールド名と異なる名前にしたい場合に使用します。

キーワードの形式は次のとおりです。

```
RENAME(physical-file-field-name)
```

物理ファイルのレコード様式で使われている名前が、このキーワードのパラメーター値になります。物理ファイルのレコード様式内の 1 つのフィールドを、定義中のレコード様式内では名前を変更して複数のフィールドにすることもできます。

フィールドの名前変更は次のような場合に行います。

- 使用したい複数のプログラムで、同じフィールドに対してそれぞれ別の名前が使われている場合。
- 物理ファイルのレコード様式の 1 つのフィールドを、論理ファイルのレコード様式の複数のフィールドにマッピング (対応づけ) したい場合。
- 使用中の高水準言語で、異なる名前を持つ 2 つのフィールドが 1 つのデータ記憶域を共有することが許されない場合 (RPG はこのような言語の 1 つです)。このような場合には、RENAME キーワードを指定することにより、どちらのフィールドも同じデータ記憶域にアクセスできるようになります。

論理ファイルの 1 つのレコード様式の中で、同じ物理ファイルのフィールドを、RENAME または CONCAT を使用して複数回指定した場合には、論理ファイルにフィールドを指定した順序と同じ順序で更新操作または挿入操作時にその物理ファイルにデータが渡されます。したがって、その物理フィールドの最後の指定が、物理レコードに入れられる値であり、その物理フィールドをキー・フィールドとするすべてのキーで使用される値です。同じフィールドの前の値はすべて重ね書きされ、効力がなくなります。

## 例

以下に RENAME キーワードの指定方法の例を示します。

次の例では、物理ファイル (PF1) 内の QTYDUE フィールドが、論理ファイルでは QTY という名前に変更されます。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00050A          R RCD1          PFILE(PF1)
00060A          QTY            RENAME(QTYDUE)
      A

```

次の例では、論理ファイル内の名前変更されたフィールド (QTY) がキー・フィールドとして使用されています。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00050A          R RCD2          PFILE(PF2)
00060A          :
      A          :
00130A          QTY            RENAME(QTYDUE)
00140A          K QTY
      A

```

## 物理ファイルおよび論理ファイルの SIGNED (符号付き) キーワード

これはキー・フィールド・レベルのキーワードで、この数値キー・フィールドに関連する値の順序付けを行う場合に、オペレーティング・システムが値の符号 (正か負か) を評価の対象とすることを指定するために使用します。

このキーワードにはパラメーターはありません。

次の例は、いずれもゾーン 10 進数キー・フィールドのある 6 つのレコードを示しています。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

デフォルト (順序付けキーワードも ALTSEQ キーワードも指定されていない) の場合は、キー・フィールドの属性は SIGNED になります。レコードは、下記の順序で順序付けされます。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
3	98-	F9D8
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
6	99	F9F9

SIGNED と DESCEND の両方を指定した場合には、レコードは下記の順序で順序付けされます。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
6	99	F9F9
1	98	F9F8
4	97	F9F7
5	20	F2F0
2	00	F0F0
3	98-	F9D8

このキーワードはデータ・タイプが文字、日付、時刻、時刻スタンプ、または 16 進数のフィールドに対しては無効です。このキーワードを ABSVAL、DIGIT、UNSIGNED、または ZONE の各キーワードと同時に使用することはできません。

SIGNED (キー・フィールド・レベルのキーワード) を使用すると、ALTSEQ (ファイル・レベル・キーワード) は無視されます。キー・フィールドに SIGNED を指定すると、ファイル・レベルで ALTSEQ が指定されていても、そのキー・フィールドについては NOALTSEQ が自動的に効力を持ちます。このことは NOALTSEQ の指定の有無にかかわらず起こります。

## 例

次の例は、物理ファイルでの SIGNED キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD
00020A      FLDA          7S 2
00030A      FLDB
00040A      K FLDA          SIGNED
      A
```

## SST (サブストリング) キーワード - 論理ファイルのみ

これはフィールド・レベルのキーワードで、既存の文字、16 進数、ゾーン・フィールド、または図形のサブセットである文字ストリングを指定するために使用します。

キーワードの形式は次のとおりです。

SST(field-name starting-position [length])

field-name パラメーターには、サブストリングの取り出し元のフィールドの名前を指定します。このフィールドは、同じ論理ファイル様式の中で SST フィールド (現在定義中のフィールド) より前に定義されているか、あるいは、PFILE または JFILE キーワードで指定した物理ファイルの中に存在していなければなりません。このフィールドを見つけるために、システムは一致するフィールド名を次のようにして検索します。

- 最初に、システムはこの SST フィールドより前の論理ファイル様式の 19 から 28 桁目に指定されているフィールド名を調べます。
- 一致するフィールド名が論理ファイル様式の 19 から 28 桁目に見つからない場合、システムは次の規則に従い、PFILE キーワードまたは JFILE キーワードで指定された物理ファイルの中から当該のフィールド名を検索します。

- 論理ファイルが単一様式論理ファイルまたは複数様式論理ファイルの場合には、当該フィールドは、PFILE キーワードで指定したすべてのファイルに入っていないければなりません。
- 論理ファイルが結合論理ファイルであり、SST フィールドについて JREF キーワードが指定されている場合には、当該フィールドは、JREF キーワードで参照される JFILE に入っていないければなりません。
- 論理ファイルが結合論理ファイルであり、SST フィールドについて JREF キーワードが指定されていない場合には、当該フィールドは、1 つの JFILE だけに入っていないければなりません。

サブストリングは、SST キーワードで指定する開始桁から始まります。その長さは、キーワードの 3 番目のパラメーターとして指定するか、またはフィールドの長さ (DDS の 30 から 34 桁目) で指定します。開始桁は必須パラメーターですが、長さはオプションです。

注: 開始桁および長さの値は両方とも正の整数値でなければならず、定義するサブストリングは、SST キーワードで指定したフィールドの長さより大きくすることはできません。

この場合には次の規則が適用されます。

- SST キーワードのフィールドが 2 進文字である場合、結果のフィールドも 2 進文字であり、SST キーワードのフィールドが 16 進数である場合は、結果のフィールドも 16 進数です。また、SST キーワードのフィールドが DBCS グラフィックである場合は、結果のフィールドも DBCS グラフィックです。それ以外の場合、結果のフィールドは必ず文字フィールドです。DDS にデータ・タイプが指定されていないと、結果のフィールドのデータ・タイプは、次の表に示すサブストリング化されたフィールドによって決まります (ソース・フィールド・タイプ は、物理ファイル・フィールド、または前に論理ファイル・ソースで定義された論理ファイル・フィールドのタイプです)。

ソース・フィールド・タイプ	結果の論理ファイル・フィールド
A	A
H	H
S	A
G	G
2 進文字	2 進文字

- 結果のフィールドの使用目的は、入力専用 (I) または非入出力 (N) のいずれかでなければなりません。
- 結果のフィールドの長さはオプションです。フィールドの長さか、キーワードの length パラメーターのいずれかを指定しなければなりません。両方を指定する場合には、両方が等しい値でなければなりません。フィールドの長さを指定しなかった場合には、キーワードの length パラメーターの値が割り当てられます。
- SST キーワードは、CONCAT、RENAME、または TRNTBL キーワードと同時に、同じフィールドに対して指定することはできません。
- このキーワードに指定したフィールドを、CONCAT、TRNTBL、または SST キーワードを使用して定義することはできません。

## 例

以下に SST キーワードの指定方法の例を示します。

次の例は、結合論理ファイルでの SST キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
  A           R RECORD1           JFILE(PF1 PF2)
  A           J                   JOIN(1 2)
```

```

A          JFELD(CITY CITY)
A          ADDRESS      JREF(2)
A          CITY          I  SST(ADDRESS 21 10)
A          JREF(2)
A          SYEAR         I  SST(SALESDATE 5)
A          NAME          JREF(1)
A          CUSTNAME      I  SST(NAME 11 10) JREF(2)
A          K SYEAR
A

```

この例には、下記のことが示されています。

- CITY は、論理ファイル様式のフィールド ADDRESS のサブストリングであり、PF1 の CITY と結合されます。
- CUSTNAME は、論理ファイル様式のフィールド NAME に別の JREF が指定されているため、PF2 のフィールド NAME のサブストリングです。
- SYEAR は、キー・フィールドであり、固有のフィールド名 SALESDATE が PF1 の中に存在していなければなりません。
- SST キーワードを指定したフィールドの使用目的 (38 桁目) は、I (入力専用) でなければなりません。これは、結合論理ファイルであるため、使用目的のデフォルト値は I です。

次の例は、単一または複数様式論理ファイルの場合の SST キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R REC1        PFILE(PFA)
A          LASTNAME      I  SST(NAME 10 10)
A          K LASTNAME
A

```

この例では、LASTNAME フィールドは PFA の NAME のサブストリングです。単一样式論理ファイルまたは複数様式論理ファイルの SST フィールドには、38 桁目に使用目的 I を指定しなければなりません。

#### 関連資料

26 ページの『物理ファイルおよび論理ファイルの長さ (30 から 34 桁目)』

この欄は、物理ファイル・フィールドまたは論理ファイル・フィールドの長さを指定するのに使用します。

## 物理ファイルおよび論理ファイルの TEXT (テキスト) キーワード

これはレコード・レベルまたはフィールド・レベルのキーワードで、レコード様式またはフィールドに、プログラムの文書化に使用されるテキスト記述 (または注記) を指定するために使用します。

キーワードの形式は次のとおりです。

```
TEXT('description')
```

テキストは、単一引用符で囲まなければなりません。50 文字を超えるテキストを指定した場合には、高水準言語コンパイラーが使用するのは最初の 50 文字だけです。

**注:** 論理ファイルに TEXT キーワードを指定してフィールドは指定しない場合には、物理ファイルのテキスト・キーワードが使用されます (指定されている場合)。

#### 例

次の例は、レコード・レベルおよびフィールド・レベルでの TEXT キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R CUSMST          TEXT('Customer Master Record')
00020A          FLD1              3 0      TEXT('ORDER NUMBER FIELD')
A

```

## 物理ファイルおよび論理ファイルの TIMFMT (時刻形式) キーワード

これはフィールド・レベル・キーワードで、時刻フィールドの形式を指定するために使用します。このキーワードは、時刻フィールド (データ・タイプ T) か、または対応する物理ファイルのフィールドが時刻フィールド (データ・タイプ T) であるゾーン・フィールド (データ・タイプ S) に使用することができます。

キーワードの形式は次のとおりです。

TIMFMT(time-format)

次の表は使用できる時刻形式とそのデフォルトの区切り記号を説明したものです。

形式名	時刻形式パラメーター	時刻形式および区切り記号	フィールド長	例
時:分:秒	*HMS	hh:mm:ss	8	14:00:00
国際 標準化機構	*ISO	hh.mm.ss	8	14.00.00
IBM 米国規格	*USA	hh:mm AM または hh:mm PM	8	2:00 pm
IBM 欧州規格	*EUR	hh.mm.ss	8	14.00.00
日本工業規格西暦	*JIS	hh:mm:ss	8	14:00:00

物理ファイルに対して TIMFMT キーワードを指定しない場合には、\*ISO がデフォルト値となります。

論理ファイルに対して TIMFMT キーワードを指定しない場合には、物理ファイルの時刻形式がデフォルト値となります。

\*ISO、\*USA、\*EUR、または \*JIS の時刻形式パラメーターの値を指定した場合には、TIMSEP キーワードは指定できません。これらの形式には既定の時刻区切り記号があります。

TIMFMT キーワードは、時刻フィールドについてのジョブ属性を指定変更します。システム・デフォルト値を変更するものではありません。

### 例

次の例は、TIMFMT キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A          TIMFLD1          T          TIMFMT(*ISO)
00040A          TIMFLD2          T          TIMFMT(*USA)
A

```

この例で、現在の時刻が午後 2 時、システムの時刻形式が hhmmss、システムの時刻区切り記号が ':' である場合、TIMFLD1 には 14.00.00 が入ります。TIMFLD2 には 2:00 PM が入ります。

## 物理ファイルおよび論理ファイルの TIMSEP (時刻区切り記号) キーワード

これはフィールド・レベルのキーワードで、時刻フィールドに使用する区切り記号を指定するために使用します。このキーワードは時刻フィールド (データ・タイプ T) にだけ使用することができます。

キーワードの形式は次のとおりです。

```
TIMSEP(*JOB | 'time-separator')
```

**time-separator** パラメーターを使用して、時、分、秒の値の間に入れる区切り記号を指定します。有効な値はコロン (:), ピリオド (.), コンマ (,), およびブランク ( ) です。このパラメーターは単一引用符で囲まなければならない。

\*JOB を指定した場合、ジョブ属性がデフォルト値となります。

物理ファイルでは、TIMSEP キーワードを指定しない場合、ジョブ属性がデフォルト値となります。

論理ファイルでは、TIMSEP キーワードを指定しない場合、物理ファイルの時刻区切り記号がデフォルト値となります。物理ファイル・フィールド (TIMFMT キーワードに \*ISO、\*USA、\*EUR、または \*JIS が指定されている) に TIMSEP キーワードを指定しない場合には、TIMSEP のデフォルト値はジョブ属性です。

TIMFMT キーワードに、\*ISO、\*USA、\*EUR、または \*JIS 時刻形式を指定した場合には、TIMSEP キーワードを指定することはできません。これらの形式には既定の時刻区切り記号があります。

DFT キーワードが指定されていない場合は、現在の時刻がデフォルト値になります。

TIMSEP キーワードは、時刻フィールドについてのジョブ属性を指定変更します。システム・デフォルト値を変更するものではありません。

### 例

次の例は、TIMSEP キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A          TIMFLD1          T          TIMSEP(' ')
00040A          TIMFLD2          T          TIMSEP('.')
```

この例で、現在の時刻が午後 2 時、システムの時刻形式が hhmmss、システムの時刻区切り記号が ':' である場合、TIMFLD1 には 14 00 00 が入ります。TIMFLD2 には 14.00.00 が入ります。

## TRNTBL (変換テーブル) キーワード - 論理ファイルのみ

これはフィールド・レベルのキーワードで、PFILE キーワードまたは JFILE キーワードに指定した物理ファイルとプログラムとの間でこのフィールドの受け渡しが行われるさいに使用される変換テーブルの名前を指定するために使用します。

このフィールドは文字フィールドでなければならない、その長さを論理ファイルで定義し直すことはできません。TRNTBL キーワードとともに CONCAT キーワードを指定する場合には、CONCAT キーワードで指定するフィールドは、すべて文字フィールドでなければならない。

キーワードの形式は次のとおりです。

```
TRNTBL([library-name/]translation-table-name)
```

translation-table-name は必須パラメーター値であり、library-name はオプションです。ライブラリー名を指定しなかった場合には、オペレーティング・システムにより、ファイル作成時に有効なライブラリー・リスト (\*LIBL) が使用されます。

translation-table-name に ICU テーブルの名前を指定することはできません。

このキーワードは、入力専用 (38 桁目に I が指定されている) フィールドまたは非入出力 (38 桁目に N が指定されている) フィールドである文字フィールドに対してだけ有効です。

16 進数フィールド (35 桁目が H) に対しては、TRNTBL キーワードは指定できません。また、日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に TRNTBL キーワードを指定してはなりません。

同一論理ファイル内の異なるフィールドについて、最高 99 個の異なる変換テーブルを指定することができます。

変換は、物理ファイルから該当のフィールドを読み取る時点で行われます。したがって、論理ファイルで指定されているすべての機能 (キー・フィールドの順序付け、選択/除外処理、またはレコードの連結など) は、変換後のデータに基づいて行われます。

TRNTBL キーワードは、論理ファイルから戻されるレコード内のデータを変更します。ALTSEQ キーワードは、論理ファイルから戻されるレコードの順序だけを変更します。

TRNTBL キーワードは、次の点を除いて、CHRID キーワードとよく似ています。

- TRNTBL キーワードの場合は、使用する変換テーブルの名前を指定しますが、CHRID キーワードの場合はこれは指定しません。
- TRNTBL キーワードの場合は、プログラムが論理ファイルを読み取る時点で、プログラムへの入力としてのデータを変更します。CHRID キーワードの場合は、特定の装置に表示または印刷するデータを変更します。プログラムが更新後のデータを使用する (たとえば IF-THEN-ELSE ステートメントや、COBOL の SORT ステートメントなどで) 場合には、TRNTBL キーワードを使用します。プログラムが論理ファイルからのデータだけを処理する場合には、プログラムで使用する表示装置ファイルまたは印刷装置ファイルに CHRID キーワードを指定する必要はありません。

論理ファイル作成 (CRTLF) コマンドで FILETYPE(\*SRC) を指定した場合には、TRNTBL キーワードは無効です。

#### 注:

1. TRNTBL キーワードを指定する場合には、論理ファイルの該当フィールドの長さは、物理ファイルの対応するフィールドの長さと同じでなければなりません。
2. ファイルの作成時点で、ユーザーは、所要の変換テーブルについての使用権限を持っていないとできません。変換テーブルは、テーブル作成 (CRTTBL) コマンドを用いて作成します。
3. TRNTBL キーワードで指定する変換テーブルは、論理ファイル作成時点でだけ参照されます。したがって、変換テーブルに対する変更は、論理ファイルを再作成するまでは、論理ファイルに影響を及ぼすことはありません。

#### 例

次の例は、TRNTBL キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1          PFILE(PF1)
00020A          CHAR1              I   TRNTBL(LIB1/TBL1)
```



```

00030A          CHAR2          A I      TRNTBL(LIB2/TBL2)
00040A          NUM1
00050A          NUM2
      A

```

この例では、フィールド CHAR1 は、ライブラリー LIB1 のテーブル TBL1 を用いて変換されます。フィールド CHAR2 は、ライブラリー LIB2 のテーブル TBL2 を用いて変換されます。フィールド CHAR2 は、TRNTBL キーワードが指定できるように、文字フィールド (35 桁目が A) として論理ファイル内で再定義されています。フィールド NUM1 と NUM2 は、物理ファイル PF1 の数字フィールドであり、これらのフィールドに対して TRNTBL キーワードを指定することはできません。

## 物理ファイルおよび論理ファイルの UNIQUE (固有) キーワード

これはファイル・レベルのキーワードで、当該物理および論理ファイルの同一メンバー内に重複するキーを持つレコードがあってはならないことを指定するために使用します。

パラメーターを使用してヌルのキー値を重複キーと見なすかどうかを指定することができます。新しいレコードの挿入または追加あるいは既存のレコードの更新で、重複キーが生じる可能性がある場合は、その操作は拒否されます。このような書き込みまたは更新命令を出したアプリケーション・プログラムに対しては、エラー・メッセージが送られます。ワークステーション・ユーザーがデータ・ファイル・ユーティリティー (DFU) を使用していると、メッセージはワークステーションに表示されます。当該ファイルに重複キーを持つレコードを複写するファイル・コピー・コマンドは、正常に完了しません。

このキーワードの形式は次のとおりです。

```
UNIQUE[(*INCNUL | *EXCNUL)]
```

このパラメーターの指定はオプションです。指定した場合、このパラメーターにより、ヌルのキー値を重複と見なすかどうかが決まります。\*INCNUL がデフォルト値で、これは重複の判別のさいにヌル値を含めることを示します。\*EXCNUL を指定した場合は、重複の判別のさいにヌル値を含めないことを示します。

物理ファイルを基礎とする論理ファイルについて UNIQUE キーワードが指定されている場合は、その物理ファイルのメンバーはいずれも重複キー値を持つことはできません。

物理または論理ファイルに対して UNIQUE キーワードを指定した場合には、そのファイルを作成する物理ファイル作成 (CRTPF) または論理ファイル作成 (CRTLF) コマンドに MAINT(\*IMMED) パラメーター値を指定しなければなりません。これは、変更が行われるとただちにアクセス・パスも更新されることを意味します。

UNIQUE キーワードを指定しなかった場合には、重複キー値を持つレコードは指定した順序で処理されます。FIFO キーワードを指定した場合には、先入れ先出しの順序となり、LIFO キーワードを指定した場合には、後入れ先出しの順序となります。FCFO キーワードを指定した場合には、変更順検索の順序になります。FIFO、LIFO、または FCFO のいずれも指定しない場合には、レコードがどの順序で順序付けされるかは特定できません。

UNIQUE キーワードを、FIFO、LIFO、FCFO、または REFACCPH キーワードとともに指定することはできません。

### 例

次の例は、論理ファイルの場合の UNIQUE キーワードの指定方法を示します。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A*
00020A* SAMPLE LOGICAL FILE (CUSMSTL)
00030A*
00040A
00050A          R CUSREC          UNIQUE
00060A          PFIL(CUSMSTP)
00070A          CUST              TEXT('Logical File Master Record')
00080A          NAME
00090A          ADDR
00100A          K CUST
      A

```

## 物理ファイルおよび論理ファイルの UNSIGNED (符号なし) キーワード

これはキー・フィールド・レベルのキーワードで、数字フィールドを符号なしの 2 進データのストリングとして順序付けすることを指定するために使用します。文字、日付、時刻、時刻スタンプ、および 16 進数フィールドは、デフォルト値により、符号なしの値として扱われます。

このキーワードにはパラメーターはありません。

UNSIGNED は、キー・フィールドのデータ・タイプに関係なく、物理または論理ファイルのキー・フィールドについて使用することができます。UNSIGNED キーワードは、SIGNED キーワードおよび ABSVAL キーワードと同時に指定することはできません。

次の場合には、UNSIGNED キーワードがデフォルト値になります。

- ゾーン 10 進数キー・フィールドに対して、ファイル・レベルで ALTSEQ が指定されている場合。
- ゾーン 10 進数キー・フィールドに対して、ZONE または DIGIT が指定されている場合。
- すべての文字フィールドおよび 16 進数フィールド。

注: 浮動小数点数フィールドについて UNSIGNED を指定することはできますが、その結果は予測できません。

次の例は、ゾーン 10 進数キー・フィールドを持つ 6 つのレコードの例です。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

UNSIGNED を指定した場合には、レコードの順序は次のようになります。

レコード	数字キー・フィールド (ゾーン 10 進数)	16 進数表現
2	00	F0F0
5	20	F2F0
3	98-	F9D8
4	97	F9F7
1	98	F9F8
6	99	F9F9

## 例

次の例は、物理ファイルでの UNSIGNED キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORDA
00020A      FLDA          7S 2
00030A      FLDB          5
00040A      K FLDA          UNSIGNED
          A
```

## 物理ファイルおよび論理ファイルの VALUES (値) キーワード

このキーワードは、フィールド・レベル、選択または除外フィールド・レベル (またはその両方) で指定します。

キーワードの形式は次のとおりです。

VALUES(value-1 [value-2...[value-100]])

## 例

以下に VALUES キーワードの指定方法の例を示します。

### 例 1

次の例では、文字値および数値を使用します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1      PFILE(PF1)
00020A      FIELDA          1 0      VALUES(1 6 9)      1
00030A      FIELDB          1          VALUES('A' 'B' 'C') 1
00040A      K FIELDA
00050A      S FIELDB          VALUES('A' 'B')      2
00060A      S FIELDA          VALUES(1 6)          2
          A
```

- VALUES は、FIELDA および FIELDB を参照する表示装置ファイルについての妥当性検査キーワードとして、FIELDA および FIELDB に対して指定されています。
- VALUES は、また、FIELDA および FIELDB に対し、選択/除外キーワードとしても指定されています。物理ファイル PF1 からのレコードは、次のフィールドの値に従って、この論理ファイルのレコード様式を介して検索されます。
  - FIELDB: FIELDB が A または B に等しい場合に限り、レコードが検索されます。
  - FIELDA: FIELDA が 1 または 6 に等しい場合に限り、まだ選択されていないレコードが選択されます。

### 例 2

次の例では、16 進値を使用します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RCD1          PFILE(PF1)
00020A      CODEA
00030A      FLD1
00040A      FLD2
00050A      K FLD1
00060A      S CODEA          VALUES(X'51' X'54' X'AE')
          A
```

この例では、VALUES は、CODEA (1 バイトのフィールド) に対する選択/除外キーワードとして指定されています。フィールド CODEA の値が 16 進数の 51、54、または AE の場合にだけ、物理ファイル PF1 からのレコードがこのレコード様式を介して検索されます。

## フィールド・レベルの VALUES の指定

フィールド・レベルでは、このキーワードは、定義中のフィールドがあとで表示装置ファイルの作成時に参照される場合に、そのフィールドに対して行われる妥当性検査を指定するために使用します。

VALUES が定義中の物理および論理ファイルに影響を及ぼすことはありません。表示装置ファイルで入力可能フィールドを定義する場合に、29 桁目に R を指定し、REF キーワードまたは REFFLD キーワードを指定することによって、定義するフィールドを参照することができます。表示装置ファイル作成時に、オペレーティング・システムは、VALUES キーワードおよびその他のフィールド属性を、物理ファイルまたは論理ファイルの該当フィールドから表示装置ファイルのフィールドに複写します。VALUES キーワード (およびその他のすべての妥当性検査キーワードならびに CHKMSGID キーワード) は、表示装置ファイルのフィールドに新たに妥当性検査キーワードを指定することによって指定変更することができます。

物理ファイルまたは論理ファイルでのこのキーワードの指定に関する規則は、表示装置ファイルの場合と同じです。

浮動小数点数フィールド (35 桁目が F) または 16 進数フィールド (35 桁目が H) に対しては、VALUES キーワードは指定できません。日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に VALUES キーワードを指定してはなりません。

### 関連情報

表示装置ファイルの参照 (29 桁目)

表示装置ファイルの VALUES (値) キーワード

## 選択/除外フィールド・レベルの VALUES の指定

選択/除外フィールド・レベルでは、このキーワードは、選択/除外フィールドの指定されたレコード様式を使用する入力命令がユーザーのプログラムから送られた場合に、物理ファイルから検索されたレコードを選択または除外するために使用されます。

この場合には次の規則が適用されます。

- 定義しているフィールドが文字フィールドの場合には、文字ストリングまたは 16 進数文字ストリングを指定しなければなりません。

文字ストリングは単一引用符で囲んで指定します (93 ページの『物理ファイルおよび論理ファイルの VALUES (値) キーワード』トピックの例 1 を参照)。

16 進数文字ストリングは、X の後に、数字 (0 から 9) と文字 (A から F) の組み合わせを単一引用符で囲んで指定します。単一引用符内の 16 進数字の数は、指定したフィールドの長さの 2 倍でなければなりません (93 ページの『物理ファイルおよび論理ファイルの VALUES (値) キーワード』トピックの例 2 を参照)。

- 定義しているフィールドが数字フィールドの場合には、数値 (単一引用符で囲まない 0 から 9 の数字) を指定しなければなりません。93 ページの『物理ファイルおよび論理ファイルの VALUES (値) キーワード』のトピックの例 1 を参照してください。

- 定義しているフィールドが日付フィールドの場合には、DATFMT キーワードで指定されているのと同じ形式で有効な日付を指定し、DATSEP キーワードで指定されているのと同じ区切り記号を使用してください。たとえば、DATFMT で \*MDY が指定され、DATSEP で「/」が指定されているときは、VALUES('12/15/91' '12/31/91') は有効な値です。
- 定義しているフィールドが時刻フィールドの場合には、TIMFMT キーワードで指定されているのと同じ様式で有効な時刻を指定し、TIMSEP キーワードで指定されているのと同じ区切り記号を使用してください。たとえば、TIMFMT で \*ISO が指定されている場合は、VALUES('11.00.00' '12.00.00') は有効な値です。\*ISO のデフォルトの区切り記号はピリオド (.) です。
- 定義しているフィールドが時刻スタンプ・フィールドの場合には、次の形式でデフォルト値を指定しなければなりません。  
VALUES('YYYY-MM-DD-HH.MM.SS.UUUUUU' 'YYYY-MM-DD-HH.MM.SS.UUUUUU')

## 物理ファイルおよび論理ファイルの VARLEN (可変長フィールド) キーワード

これは、フィールド・レベルのキーワードで、当該フィールドを可変長フィールドとして定義するために使用します。

可変長フィールドを使用すると、フィールドのデータが通常は一定の長さに納まるが、時としてそれより長くなるという場合に、記憶域の効率が良くなります。当該フィールドの最大長は 30 から 34 桁目に指定します。割り振り長 (つまり標準的な一定の長さ) はパラメーターで指定することができます。

キーワードの形式は次のとおりです。

VARLEN[(allocated-length)]

allocated-length パラメーターはオプションです。このパラメーターは、当該ファイルの固定部分の中で当該フィールドに割り振るバイト数 (図形フィールドの場合は 2 バイト文字) を指定するために使用します。allocated-length パラメーターを指定しない場合、このフィールドのデータはファイルの可変長部分に保管されます。

allocated-length パラメーターに使用できる値は、1 から当該フィールドの 30 から 34 桁目で指定された最大長までです。

論理ファイルの場合、このキーワードにはパラメーターはありません。

VARLEN キーワードは文字フィールドと図形フィールドにだけ使用することができます。

VARLEN キーワードを指定する場合、30 から 34 桁目に指定できる最大長は 32 740 (ヌル値を許容するフィールドの場合は 32 739) です。グラフィックス・フィールドの場合、指定できる最大長は 16 370 です。

ある可変長フィールドに DFT キーワードを指定する場合、デフォルト値の長さは、そのフィールドの割り振り長以下でなくてはなりません。デフォルト値が割り振り長より長い場合には、ファイルの作成時にエラー・メッセージが出されます。

可変長フィールドのデフォルト値として 16 進値を指定する場合、16 進文字の文字数は、当該フィールドの割り振り長の 2 倍に等しくなければなりません。

allocated-length パラメーターに値を指定する場合を除いて、DFT キーワードを VARLEN キーワードと同じフィールドに使用することはできません。

日付、時刻、または時刻スタンプ・フィールド (35 桁目が L、T、または Z) に VARLEN キーワードを指定してはなりません。

## 例

次の例は、物理ファイルでの VARLEN キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A          FIELD1      100A      VARLEN(30)
00030A          FIELD2      200A      VARLEN
A
```

この例では、FIELD1 は最大長が 100 で割り振り長が 30 の可変長フィールドとして定義されています。FIELD2 は最大長が 200 で割り振り長を持たない可変長フィールドとして定義されています。

## 物理ファイルおよび論理ファイルの ZONE (ゾーン) キーワード

これは、キー・フィールド・レベルのキーワードで、このキー・フィールドに関連する値が構成される際に、キー・フィールドの各バイトのゾーン部分 (上位 4 ビット)だけを使用することを指定するために使用します。ディジット部分はゼロで埋め込まれます。

このキーワードにはパラメーターはありません。

このキーワードは、フィールド内の特定の桁ではなく、キー・フィールド全体に対して適用されるもので、データ・タイプが文字、16 進数、またはゾーン 10 進数のフィールドに対してだけ有効です。

ZONE は ABSVAL、SIGNED、または DIGIT キーワードと同時に指定することはできません。

キー・フィールドに対して ZONE を指定した場合には、フィールドの値は符号なしの 2 進データのストリングと見なされます (ゾーン 10 進数フィールドについてのデフォルト値である符号付きデータ・ストリングとは見なされません)。

## 例

次の例は、ZONE キーワードの指定方法を示します。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          K CODE          ZONE
A
```

この例では、CODE が 1 バイトのフィールドの場合、3 つの異なるレコードのフィールド値は次のようになります。

値	16 進数	キーとして使用される部分
A	C1	C
B	C2	C
E	C5	C

---

## データベース・ファイルの Unicode に関する考慮事項

Unicode は、出力する文字およびテキストの万国共通のエンコード・スキームで、各国間でのデータ交換を可能にします。以下のトピックでは、データベース・ファイルの記述に使用される DDS の 30 から 37 桁目および 45 から 80 桁目の指定方法について説明します。説明のない桁については、Unicode に関する特別な考慮事項はありません。

Unicode フィールドには、IBM iSeries™ サーバーで使用されるすべてのタイプの文字 (DBCS 文字も含まれる) を含めることができます。Unicode データは、1 単位のテキストを表現できる最小バイトの組み合わせを表すコード単位 で構成されます。

物理ファイルおよび論理ファイルの DDS でサポートされる Unicode には、次の 3 つの変換様式 (エンコード方式) があります。

- **UTF-8** は、既存の ASCII ベース・システムで使用しやすいように設計された 8 ビットのエンコード方式です。UTF-8 データは、文字データ・タイプとして保管されます。UTF-8 様式データの CCSID 値は 1208 です。

UTF-8 コード単位の長さは 1 バイトです。UTF-8 文字は 1、2、3、または 4 コード単位の長さになることができます。UTF-8 データ・ストリングには、サロゲート文字や結合文字などを含む任意の文字を含めることができます。

- **UTF-16** は 100 万を超える文字にコード値を割り当てるための 16 ビット・エンコード方式で、UCS-2 のスーパーセットです。UTF-16 データは、グラフィック・データ・タイプとして保管されます。UTF-16 様式データの CCSID 値は 1200 です。

UTF-16 コード単位の長さは 2 バイトです。UTF-16 文字の長さは、1 または 2 コード単位 (2 または 4 バイト) です。UTF-16 データ・ストリングには、サロゲートや結合文字などを含む任意の文字を含めることができます。

- **UCS-2** は 2 オクテットで表現される汎用文字セットです。つまり、1 文字は 16 ビットで表されます。UCS-2 データは、グラフィック・データ・タイプとして保管されます。UCS-2 様式データの CCSID 値は 13488 です。

UCS-2 は UTF-16 のサブセットで、Unicode で定義されたすべての文字をサポートするわけではありません。UTF-16 が結合文字とサロゲートもサポートすること以外は、UCS-2 と UTF-16 は同じです。UCS-2 タイプの方が使用可能なデータベース機能が豊富なため、結合文字およびサロゲートのサポートが必要ない場合は、このタイプの使用を選択できます。

注: 以下のトピックで、UTF-16 に言及する内容は、UCS-2 も対象としています。

## 長さ (30 から 34 桁目)

この欄にはフィールドの長さを指定します。UTF-16 のデータを含むフィールドの長さは、1 から 16 383 までのコード単位です。UTF-8 のデータを含むフィールドの長さは、1 から 32 766 までのコード単位です。

Unicode データを含むフィールドのプログラムの長さを決定するときには、以下の規則を考慮してください。

- UTF-16 コード単位の長さはいずれも 2 バイトです。
- フィールドの長さは UTF-16 コード単位の数で指定します。たとえば、UTF-16 コード単位を 3 個含むフィールドのデータは、6 バイトです。
- UTF-8 コード単位の長さはいずれも 1 バイトです。UTF-8 文字は 1、2、3、または 4 コード単位の長さになることができます。
- Unicode データと EBCDIC との間の変換後のデータは、変換前の元のデータの長さと同くなるか、それより長くなったり短くなったりします。たとえば、UTF-16 コード単位 1 つは、2 バイトのデータで構成されています。この文字は、1 バイトのデータから成る 1 つの SBCS 文字、2 バイトのデータから成る 1 つの図形 DBCS 文字、4 バイトのデータから成る 1 つのシフト文字付き DBCS 文字に変換されます。したがって、(物理ファイル内の) Unicode フィールドを、論理ファイル内の異なるタイプの

フィールドに変換するときには、論理ファイル内のフィールドは、VARLEN キーワードを指定して定義することをお勧めします。論理ファイル・フィールドの長さは、Unicode フィールドの変換結果の最大サイズを保持するのに十分な大きさに定義しておく必要があります。これには、将来の拡張の可能性も考慮しておきます。

論理ファイルでは、長さを指定せずに UTF-16 から EBCDIC への変換が行われる場合には、以下の場合を除いて、論理ファイル・フィールドに対応する物理ファイル・フィールドの長さが用いられます。

- 物理ファイル・フィールドで UTF-16 の使用が可能で、論理ファイル・フィールドのデータ・タイプが O である場合には、論理ファイル・フィールドの長さは物理ファイル・フィールドの 2 倍になります。

## データ・タイプ (35 桁目)

Unicode データとして有効なデータ・タイプは G (グラフィック) データ・タイプと A (文字) タイプです。

### G (図形)

タイプ G を CCSID キーワードとともに使用すると、フィールドが UTF-16 データを含むことになります。

通常は、G を指定すると、フィールドは図形 DBCS EBCDIC データを含むことになります。CCSID キーワードとともに指定すると、フィールドは UTF-16 データを含むようになります。物理ファイルと論理ファイル内の対応するフィールド間で変換が必要なときには、データは UTF-16 CCSID の文字と対応するフィールドの CCSID の文字の間でマッピングされます。

### A (文字)

タイプ A を CCSID キーワード (1208) とともに使用すると、フィールドが UTF-8 データを含むことになります。

通常は、A を指定すると、フィールドには EBCDIC データが含まれることになります。CCSID キーワードとともに指定すると、フィールドは UTF-8 データを含むようになります。物理ファイルと論理ファイル内の対応するフィールド間で変換が必要なときには、データは UTF-8 CCSID の文字と対応するフィールドの CCSID の文字の間でマッピングされます。

### 関連資料

29 ページの『物理ファイルおよび論理ファイルのデータ・タイプ (35 桁目)』

物理ファイルでは、この欄にはデータベース内でのフィールドのデータ・タイプを指定します。論理ファイルの場合にこの欄にデータ・タイプを指定するのは、この論理ファイルの基礎となっている物理ファイル内の対応するフィールドのデータ・タイプを指定変更または修正したい場合に限りです。

## 小数点以下の桁数 (36 から 37 桁目)

Unicode データを使用する場合には、この欄はブランクのままにしておいてください。

## キーワードに関する考慮事項 (45 から 80 桁目)

一部の DDS キーワードで Unicode データがどのように使われているかについて説明します。

CCSID キーワードは、A タイプまたは G タイプのフィールドに Unicode データを含められるようにするために使用します。

CCSID パラメーターには、Unicode のエンコード・スキームを使用する CCSID がなければなりません。このキーワードは、物理ファイルと論理ファイルの両方で使用できます。



論理ファイルの場合、次に示す特性が満たされなければ CCSID キーワードを論理ファイル・フィールドで使用することはできません。

- 論理ファイル CCSID キーワードに指定された値が Unicode エンコード・スキームを使用する場合は、フィールド・データ・タイプは UTF-8 の場合は A、UCS-2/UTF-16 の場合は G でなければなりません。また、対応する物理ファイル・フィールドのタイプも A、G、または O でなければなりません。物理ファイル・フィールドで CCSID キーワードを指定している場合には、65 535 以外の値を含んでいなければなりません。
- 論理ファイル CCSID キーワードに指定された値が Unicode エンコード・スキームを使用しない場合は、フィールド・データ・タイプは A、O、または G でなければなりません。また、対応する物理ファイル・フィールドも Unicode フィールドでなければなりません。論理ファイル・フィールドで指定した CCSID キーワードは、65 535 以外の値を含んでいなければなりません。

DFT キーワードを Unicode 可能フィールドに指定する場合、SBCS、シフト文字付き DBCS、またはシフト文字付き DBCS 図形文字ストリングを含めることができます。

COMP キーワードを使用できるのは、別の Unicode 使用可能フィールド内のデータを比較する場合に限ります。長さが同じである 2 つの UTF-8 データ・ストリングは、文字境界に関係なく、16 進数値として相互に比較することができます。

Unicode CCSID タグが付いている選択フィールドまたは除外フィールドに、COMP、RANGE および VALUES キーワードで文字リテラルを指定できます。リテラルの最大長は、DDS 仕様書の 30 から 34 桁目に定義されている Unicode コード単位の数です。

Unicode フィールドで VARLEN キーワードを使用できます。

論理ファイルは、UTF-8 および UTF-16 データ・キーを持つことができます。

## Unicode フィールドの連結

Unicode フィールドは CONCAT キーワードで使用できます。この場合には次の規則が適用されます。

- CONCAT キーワードのパラメーターは UCS-2 グラフィック・フィールド、UTF-16 フィールド、またはそれぞれのタイプの混合でかまいません。これ以外のフィールド・タイプを UCS-2 または UTF-16 フィールドに連結することはできません。
  - パラメーターに UTF-16 が含まれている場合の連結結果は UTF-16 です。
  - それ以外の場合、結果は UCS-2 です。
- UTF-8 フィールドは、別の UTF-8 フィールドとのみ連結できます。それ以外のフィールド・タイプは、UTF-8 フィールドと連結できません。連結結果は UTF-8 になります。
- 結果フィールドは入力専用フィールドでなければなりません。DDS ソース・ステートメントの 38 桁目に I を指定してください。

## 結合論理ファイル

UTF-8 フィールドは、結合論理ファイル内で、他の UTF-8 フィールドと結合される場合があります。UTF-16 フィールドは結合可能です。例えば、UTF-8 フィールドまたは UTF-16 フィールドを CHAR フィールドと結合するには、論理ファイル内で、UTF-8、UTF-16、または CHAR フィールドを同じタイプ・フィールドに再定義する必要があります。

## 論理ファイル内の選択/除外フィールド

UTF-8 フィールドと UTF-16 フィールドは選択/除外フィールドとして指定できます。

---

## データベース・ファイルの DBCS に関する考慮事項

以下のトピックでは、データベース・ファイルについての一般的な考慮事項とともに、物理ファイルおよび論理ファイルの定位置項目およびキーワード項目についての DBCS に関する考慮事項について説明します。

これらのトピックで説明している機能は、DBCS および非 DBCS の両システムでサポートされています。

### 関連情報

DDS ファイルで DBCS テキストを使用する場合の一般的考慮事項

## DBCS を使用するデータベース・ファイルの定位置項目に関する考慮事項

以下のトピックでは、データベース・ファイルの長さ、データ・タイプ、および 10 進数の定位置項目についての DBCS に関する考慮事項について説明します。説明のない定位置項目については、DBCS に関する特別な考慮事項はありません。

### 長さ (30 から 34 桁目)

シフト文字付き DBCS データを含むフィールドの長さは、4 から 32 766 バイトです (フィールドが可変長の場合には、4 から 32 740 バイト)。DBCS グラフィック・フィールドの長さは、1 から 16 383 文字です (フィールドが可変長の場合には、1 から 16 370 文字)。

DBCS データを含むフィールドの長さを決めるさいには、以下の規則を考慮に入れてください。

- 各 DBCS 文字は、長さ 2 バイトです。
- DBCS グラフィック・フィールドの場合には、フィールドの長さは DBCS 文字の数で指定されます。
- データ・タイプ J、E、または O のフィールドの場合には、フィールドの長さに両方のシフト制御文字も含めます。これらの文字は、合わせて 2 バイトです。
- データ・タイプとして J または E を指定するフィールドの長さは偶数でなければなりません。

たとえば、3 個の DBCS 文字を含むシフト文字付きフィールドのデータの長さは、シフトイン文字およびシフトアウト文字それぞれ 1 個ずつを含めて、合計 8 バイトになります。

$(3 \text{ 文字} \times 2 \text{ バイト}) + (\text{シフトアウト} + \text{シフトイン}) = 8$

3 個の DBCS 文字を含む DBCS グラフィック・フィールドの長さは合計 6 バイトになります。

$(3 \text{ 文字} \times 2 \text{ バイト}) = 6$

### データ・タイプ (35 桁目)

4 つの DBCS データ・タイプ、すなわち J (専用)、E (択一)、O (混用) および G (図形) のいずれでも使用することができます。

#### J (専用)

DBCS データのみを入れることのできるフィールド。

#### E (択一)

DBCS データまたは英数字データのどちらか一方だけを入れることのできるフィールド。

#### O (混用)

DBCS データおよび英数字データの両方を入れることのできるフィールド。DBCS データと英数字データとの切り替えにはシフト制御文字を使用します。

#### G (図形)

シフト制御文字なしの DBCS データのみを入れることのできるフィールド。

## 関連資料

29 ページの『物理ファイルおよび論理ファイルのデータ・タイプ (35 桁目)』  
物理ファイルでは、この欄にはデータベース内でのフィールドのデータ・タイプを指定します。論理ファイルの場合にこの欄にデータ・タイプを指定するのは、この論理ファイルの基礎となっている物理ファイル内の対応するフィールドのデータ・タイプを指定変更または修正したい場合に限りです。

## 小数点以下の桁数 (36 から 37 桁目)

DBCS データを使用する場合には、この欄はブランクのままにしておいてください。

## DBCS を使用するデータベース・ファイルのキーワードに関する考慮事項

DBCS データを含むフィールドについては、数値データに使用するための DDS キーワードは指定しないでください。システムは DBCS データを文字データと同じに取り扱うので、DBCS データに対して算術演算を行うことはできません。

DBCS データ・フィールド (35 桁目に指定されているデータ・タイプが O、J、E、または G のいずれかであるもの) については、次の DDS キーワードは使用してはなりません。

ABSVAL	CHECK(VNE)	REFSHIFT
ALTSEQ	DATFMT	SIGNED
CHECK(M10)	DATSEP	SST
CHECK(M10F)	DIGIT	TIMFMT
CHECK(M11)	EDTCDE	TIMSEP
CHECK(M11F)	EDTWRD	TRNTBL
CHECK(VN)	FLTPCN	ZONE

注:

1. SST キーワードはデータ・タイプが G のフィールドで使用できます。
2. REFSHIFT キーワードは、データ・タイプが O、J、または E のフィールドで使用できます。

## 関連資料

36 ページの『物理ファイルおよび論理ファイルのキーワード項目 (45 から 80 桁目)』  
この項では、物理ファイルおよび論理ファイルの記述を行う場合に使用できるキーワード項目について説明します。キーワードは、45 から 80 桁目 (機能欄) に記入します。

## CONCAT (連結) キーワード

これはフィールド・レベル・キーワードで、物理ファイルのレコード様式の複数のフィールドを組み合わせ、定義中の論理ファイルのレコード様式内で 1 つのフィールドとして用いたい場合に使用します。

連結後にできるフィールドの名前を 19 から 28 桁目に指定しなければなりません。物理ファイルのフィールドの名前は、連結したい順に指定し、各フィールド名をブランクで区切ります。

次のような規則および制約事項があります。

- オペレーティング・システムは、連結したすべてのフィールドの合計長 (数字および文字) として、連結フィールドの長さを割り当てます。

注: データ・タイプ J のフィールドの場合、連結するフィールド間にあるシフトアウトとシフトインの対は、結果のフィールドからは除去されます。結果のデータ・タイプが 16 進数である場合には、最初の 16 進数フィールドの前にある DBCS フィールドについてシフトアウト文字とシフトイン文字の対が除去されます。

- DBCS グラフィック・フィールドは他の図形データ・タイプ・フィールドとのみ連結することができます。
- オペレーティング・システムは、連結しようとするフィールドのデータ・タイプに基づいてデータ・タイプを割り当てます。連結にシフト文字付きの DBCS フィールドが含まれる場合の一般的な規則は次のとおりです。
  - 連結の対象に 1 つまたは複数の 16 進数 (H) フィールドが含まれている場合は、結果のデータ・タイプは 16 進数 (H) です。
  - 連結するすべてのフィールドが DBCS 専用 (J) である場合には、結果のデータ・タイプは DBCS 専用 (J) です。
  - 連結に 1 つまたは複数の DBCS (O, E, J) フィールドが含まれるが、16 進数フィールドは含まれない場合には、結果のデータ・タイプは DBCS 混用 (O) です。
- オペレーティング・システムは、連結するフィールドに基づいて、フィールドを固定長フィールドまたは可変長フィールドになるように割り当てます。その場合の一般的な規則は次のとおりです。
  - 可変長フィールドを、固定長フィールドまたは別の可変長フィールドのいずれかに連結すると、結果は可変長フィールドになります。
  - 固定長フィールドを固定長フィールドに連結すると、結果は固定長フィールドになります。ただし、CONCAT キーワードを指定したフィールドと同じフィールドに、VARLEN キーワードも指定した場合を除きます。
- 連結フィールドの最大長は、連結フィールドのデータ・タイプおよび連結対象フィールドの長さによって異なります。連結フィールドがゾーン 10 進数 (S) である場合は、その合計長は 63 バイトを超えることはできません。文字 (A) または DBCS (O, J) の場合には、合計長は 32 766 バイトを超えることはできません。連結フィールドが可変長の場合には、その合計長は 32 740 バイト (ヌル値も使用可能なフィールドの場合は 32 739 バイト) を超えることはできません。

連結フィールドが DBCS 図形 (G) フィールドの場合には、合計長は 16 383 文字を超えることはできません。連結フィールドが可変長の場合には、その合計長は 16 370 文字を超えることはできません。

- 結合論理ファイルの場合には、連結するフィールドはすべて同じ物理ファイルからのフィールドでなければなりません。CONCAT キーワードで最初に指定するフィールドにより、どの物理ファイルを使用するかが識別されます。したがって、最初に指定するフィールドは、論理ファイルの基礎となるすべての物理ファイルを通じて固有のものでなければならず、そうでない場合には、どの物理ファイルを使用するのかを JREF キーワードによって指定しなければなりません。
- 連結された 1 つまたは複数のフィールドが DBCS フィールドの場合には、フィールド名も 19 から 28 桁目または DBCS 連結前に指定された RENAME キーワードか CONCAT キーワードに指定されていない限り、CONCAT キーワードのどのフィールドもキー、選択、または除外として指定できません。
- 連結したフィールドの使用目的は I (入力専用) でなければなりません。
- REFSHIFT は、データ・タイプ O または J が割り当てられた連結フィールドに指定することはできません。

## 例

次の例は、DDS コーディング仕様書での CONCAT キーワードの指定方法を示しています。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD 1          PFILE(PF1)
A          FLD1                I    CONCAT(PFLD1 PFLD2)
A          FLD2                I    CONCAT(PFLD1 PFLD2 PFLD3)
A          FLD3                I    CONCAT(PFLD4 PFLD5)
A
```

この例で、PF1 からのフィールドが次のとおりであったとします。

- データ・タイプ J の PFLD1
- データ・タイプ J の PFLD2
- データ・タイプ E の PFLD3
- データ・タイプ G の PFLD4 および PFLD5

結果のフィールドは次のようになります。

- データ・タイプ J の FLD1
- データ・タイプ O の FLD2
- データ・タイプ G の FLD3

#### 関連資料

49 ページの『CONCAT (連結) キーワード - 論理ファイルのみ』

これはフィールド・レベル・キーワードで、物理ファイルのレコード様式からの 2 つ以上のフィールドを、定義中の論理ファイルのレコード様式内で連結して 1 つのフィールドにしたい場合に使用します。連結後にできるフィールドの名前を 19 から 28 桁目に指定しなければなりません。

## DBCS データが含まれるデータベース・ファイルを記述する場合の追加考慮事項

DBCS フィールドを含むデータベース・ファイルを記述するさいには、以下の考慮事項に注意してください。

- DDS で DBCS フィールドを記述した場合、システムはそのファイルを DBCS ファイルとして扱います。したがって、それが DBCS ファイルであることを示すために、ファイル作成コマンドで IGCDTA(\*YES) を指定する必要はありません。
- 物理ファイルのフィールドのデータ・タイプは、論理ファイルのフィールドを参照する場合には、次のように変更できます。

物理ファイルのデータ・タイプ	論理ファイルのデータ・タイプ
J	J、O、E、H、G
O	O、H
E	O、E、H
A	A、O、E、H
H	J、O、E、A、H
G	G、O、J、E

注: 物理ファイル・データ・タイプが文字 (A) または 16 進数 (H)、および論理ファイル・データ・タイプが DBCS 専用 (J)、または DBCS 択一 (E) の場合には、物理ファイル・フィールド長 (30 から 34 桁目) は 4 以上の偶数でなければなりません。

- DDS では、DBCS キー・フィールドは文字フィールド (35 桁目にデータ・タイプとして O が指定されているもの) として扱われます。
- DDS では、EBCDIC 照合順序に従って DBCS データが分類されます。
- キー・フィールドの順序付けキーワードのうち文字フィールドに対して使用できるものは、次のキーワードを除いて DBCS フィールドに対しても使用することができます。

ALTSEQ  
DIGIT  
ZONE

- シフト文字付き DBCS データは、注記および文字ストリングの使用が可能なところであればどこでも使用することができます。
- データ・タイプが J 以外のシフト文字付き DBCS フィールドは、文字フィールド (データ・タイプ A) と比較することができます。
- DBCS グラフィック・フィールドは、他の図形フィールドとのみ比較することができます。
- 次の妥当性検査キーワードは、DBCS フィールドで指定することができます。

COMP  
RANGE  
VALUES

- 物理ファイルに VARLEN キーワードを指定する場合、シフト文字付き DBCS フィールドの割り振り長として指定できる最小長は 4 です。DBCS グラフィック・フィールドの割り振り長として指定できる最小長は 1 です。

#### 関連情報

DBCS 文字ストリング

---

## コード・ライセンスに関する特記事項

IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用权を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、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。

IBM、そのプログラム開発者、または供給者は、いかなる場合においてもその予見の有無を問わず、以下に対する責任を負いません。

1. データの喪失、または損傷
2. 直接損害、特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。

---

## 付録. 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す。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
Licensing

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとしします。

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。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してな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  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用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、IBM 機械コードのご使用条件、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。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_年を入れる\_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

---

## プログラミング・インターフェース情報

本書 (プログラミング DDS 物理ファイルと論理ファイル) には、プログラムを作成するユーザーが IBM i5/OS のサービスを使用するためのプログラミング・インターフェースが記述されています。

---

## 商標

以下は、IBM Corporation の商標です。



- | e(ロゴ)server
- | eServer
- | i5/OS
- | IBM
- | IBM (ロゴ)
- | iSeries

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

---

## 使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

**個人使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

**商業的使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、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なしで提供されます。







Printed in Japan