

**IMS**

バージョン 15.1.0

システム・ユーティリティー

**IBM**



**IMS**

バージョン 15.1.0

システム・ユーティリティー

**IBM**

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、741 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、IMS 15 (プログラム番号 5635-A06)、IMS Database Value Unit Edition V15.01.00 (プログラム番号 5655-DS5)、IMS Transaction Manager Value Unit Edition V15.01.00 (プログラム番号 5655-TM4)、および新しい版で明記されていない限り、以降のすべてのリソースおよびモディフィケーションに適用されます。

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： SC27-6798-00

IMS

Version 15.1.0

System Utilities

(November 5, 2017 edition)

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 1974, 2017.

# 目次

本書について . . . . .	vii
前提知識 . . . . .	vii
新規および変更された情報の識別方法 . . . . .	vii
構文図の読み方 . . . . .	viii
IMS 15 のアクセシビリティ機能 . . . . .	ix

## 第 1 部 生成ユーティリティ . . . . . 1

### 第 1 章 アプリケーション制御ブロック保守ユーティリティ . . . . . 3

ACB 保守ユーティリティの例 . . . . .	14
DOPT PSB の管理 . . . . .	15
ACB ライブラリーを使用する IMS システム内の DOPT PSB の管理 . . . . .	16
IMS 管理の ACB 環境での DOPT PSB の管理 . . . . .	16

### 第 2 章 データベース記述 (DBD) 生成ユーティリティ . . . . . 19

各データベース・タイプの DBD 生成 . . . . .	24
DBD 生成の入力レコード構造 (DEDB の DBD を除く) . . . . .	31
DEDB の DBD 生成の入力レコード構造 . . . . .	33
DBD 生成のコーディング規則 . . . . .	34
DBDGEN ステートメント . . . . .	34
DBD ステートメント . . . . .	36
DATASET ステートメント . . . . .	58
AREA ステートメント . . . . .	76
SEGM ステートメント . . . . .	79
LCHILD ステートメント . . . . .	112
FIELD ステートメント . . . . .	122
XDFLD ステートメント . . . . .	144
DFSMARSH ステートメント . . . . .	150
DFSMAP ステートメント . . . . .	157
DFSCASE ステートメント . . . . .	161
DBDGEN、FINISH、および END ステートメント . . . . .	165
DBDGEN ユーティリティの例 . . . . .	165
副次索引または論理関係がない例 . . . . .	166
論理関係のある例 . . . . .	174
副次索引のある例 . . . . .	180
DBDGEN プロシージャの実行 . . . . .	186

### 第 3 章 MFS 装置特性テーブル・ユーティリティ (DFSUTB00) . . . . . 189

DFSUTB00 ユーティリティの実行 . . . . .	192
-------------------------------	-----

### 第 4 章 MFS 言語ユーティリティ (DFSUPAA0) . . . . . 195

ユーティリティ制御ステートメントおよび構文規則 . . . . .	202
-----------------------------------	-----

制御ステートメントの要約 . . . . .	206
メッセージ定義ステートメント . . . . .	208
フォーマット定義ステートメント . . . . .	222
区画セット定義ステートメント . . . . .	280
テーブル定義ステートメント . . . . .	283
コンパイル・ステートメント . . . . .	285
標準モードでのユーティリティの実行 . . . . .	291
バッチ・モードでのユーティリティの実行 . . . . .	297
テスト・モードでのユーティリティの実行 . . . . .	300
MFS ライブラリー・バックアップ・プロシージャ . . . . .	303
MFS 復元プロシージャ . . . . .	305

### 第 5 章 プログラム仕様ブロック (PSB) 生成ユーティリティ . . . . . 307

ユーティリティ制御ステートメント . . . . .	311
代替 PCB ステートメント . . . . .	311
全機能または高速機能データベースの PCB ステートメント . . . . .	315
GSAM PCB ステートメント . . . . .	330
SENSEG ステートメント . . . . .	332
SENFLD ステートメント . . . . .	336
PSBGEN ステートメント . . . . .	338
END ステートメント . . . . .	343
PSBGEN ユーティリティの例 . . . . .	343
PSBGEN プロシージャの実行 . . . . .	362

## 第 2 部 IMS カタログ・ユーティリティ . . . . . 365

### 第 6 章 ACB Generation and Catalog Populate ユーティリティ (DFS3UACB) . . . . . 367

### 第 7 章 IMS Catalog Alias Names ユーティリティ (DFS3ALI0) . . . . . 387

第 8 章 IMS Catalog Copy ユーティリティ (DFS3CCE0、DFS3CCI0) . . . . . 389	
IMS Catalog Export ユーティリティ (DFS3CCE0) . . . . .	389
IMS Catalog Import ユーティリティ (DFS3CCI0) . . . . .	399

第 9 章 IMS Catalog Directory Recovery ユーティリティ (DFS3RU00) . . . . .	415
---	-----

第 10 章 IMS Catalog Partition Definition Data Set ユーティリティ (DFS3UCD0) . . . . .	421
---	-----

第 11 章 IMS Catalog Populate ユー ティリティ (DFS3PU00) . . . . .	425
--	-----

第 12 章 IMS Catalog Library Builder ユーティリティ (DFS3LU00) . . . . .	443
--	-----

第 13 章 IMS Catalog Record Purge ユーティリティ (DFS3PU10) . . . . .	451
---	-----

---

第 3 部 分析ユーティリティおよび び報告書 . . . . .	465
--------------------------------------	-----

第 14 章 高速機能ログ分析ユーティ リティー (DBFULTA0) . . . . .	467
高速機能報告書タイプ . . . . .	476

第 15 章 ファイル選択およびフォー ット設定印刷ユーティリティ (DFSERA10) . . . . .	491
--	-----

DFSERA10 ユーティリティの例 . . . . .	501
DFSERA10 ユーティリティ・モジュール . . . . .	508
レコード・フォーマット設定・印刷モジュール (DFSERA30) . . . . .	508
プログラム分離トレース・レコード・フォー ット設定・印刷モジュール (DFSERA40) . . . . .	517
DL/I 呼び出しイメージ・キャプチャー・モジ ュール (DFSERA50) . . . . .	520
IMS トレース・テーブル・レコード・フォー ット設定・印刷モジュール (DFSERA60) . . . . .	521
拡張選択モジュール (DFSERA70) . . . . .	521
OM 監査証跡フォーマット設定・印刷モジ ュール (CSLULALE) . . . . .	524

第 16 章 IMS モニター報告書印刷ユ ーティリティ (DFSUTR20) . . . . .	527
DFSUTR20 ユーティリティの例 . . . . .	530

第 17 章 ログ・トランザクション分析 ユーティリティ (DFSILTA0) . . . . .	531
--	-----

第 18 章 オフライン・ダンプ・フォー マッター・ユーティリティ (DFSOFMD0) . . . . .	537
DFSOFMD0 ユーティリティの実行 . . . . .	539

第 19 章 統計分析ユーティリティ (DFSISTS0) . . . . .	545
DFSISTS0 ユーティリティの例 . . . . .	555

---

第 4 部 ログ・ユーティリティ	563
------------------	-----

第 20 章 ログ保存ユーティリティ (DFSUARC0) . . . . .	565
DFSUARC0 ユーティリティの例 . . . . .	576

第 21 章 ログ・マージ・ユーティリ ティー (DFSMTMG0) . . . . .	579
---	-----

第 22 章 ログ・リカバリー・ユーテ ィリティー (DFSULTR0) . . . . .	585
DFSULTR0 ユーティリティの例 . . . . .	601

---

第 5 部 サービス・ユーティリ ティ . . . . .	607
----------------------------------	-----

第 23 章 バッチ SPOC ユーティリ ティ (CSLUSPOC) . . . . .	609
バッチ SPOC ユーティリティの例 . . . . .	612

第 24 章 データベース・リカバリー 管理ユーティリティ (DSPURX00) . . . . .	615
DSPURX00 ユーティリティの例 . . . . .	618
エントリー・ポイント DSPURXRT を使用した ユーティリティの呼び出し . . . . .	619

第 25 章 動的 SVC ユーティリ ティ (DFSUSVC0) . . . . .	623
DFSUSVC0 ユーティリティの例 . . . . .	625

第 26 章 グローバル・オンライン変 更ユーティリティ (DFSUOLC0) . . . . .	627
DFSUOLC0 ユーティリティの例 . . . . .	633

第 27 章 MFS サービス・ユーティ リティー (DFSUTSA0) . . . . .	635
---	-----

第 28 章 複数システム検査ユーテ ィリティー (DFSUMSV0) . . . . .	651
--	-----

第 29 章 オンライン変更コピー・ユ ーティリティ (DFSUOCU0) . . . . .	665
OLCUTL プロシージャ . . . . .	672
IMS>.MODSTAT データ・セットの初期設定 . . . . .	673

第 30 章 スプール SYSOUT 印刷ユ ーティリティ (DFSUPRT0) . . . . .	677
DFSUPRT0 ユーティリティの例 . . . . .	680

第 31 章 時間制御操作検査ユーティリ ティ (DFSTVER0) . . . . .	681
DFSTVER0 ユーティリティーの例 . . . . .	683
<hr/>	
第 6 部 動的リソース定義ユーティ リティー . . . . .	687
第 32 章 Repository to RDDS ユーテ ィリティー (CSLURP20) . . . . .	689
CSLURP20 ユーティリティーの例 . . . . .	692
第 33 章 RDDS to Repository ユーテ ィリティー (CSLURP10) . . . . .	695
CSLURP10 ユーティリティーの例 . . . . .	699
第 34 章 RDDS コピー・ユーティリテ ィ (DFSURCP0) . . . . .	701
DFSURCP0 ユーティリティーの例 . . . . .	703
第 35 章 Create RDDS from Log Records ユーティリティー (DFSURCL0) . . . . .	705
DFSURCL0 ユーティリティーの例 . . . . .	711
第 36 章 Create RDDS from MODBLKS ユーティリティー (DFSURCM0) . . . . .	717
DFSURCM0 ユーティリティーの例 . . . . .	722

第 37 章 DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティー (DFSURST0) . . . . .	725
DFSURST0 ユーティリティーの例 . . . . .	730
第 38 章 RDDS 抽出ユーティリティー (DFSURDD0) . . . . .	733
DFSURDD0 ユーティリティーの例 . . . . .	736
<hr/>	
第 7 部 付録 . . . . .	739
特記事項 . . . . .	741
プログラミング・インターフェース情報 . . . . .	743
商標 . . . . .	743
製品資料に関するご使用条件 . . . . .	744
IBM オンライン・プライバシー・ステートメント . . . . .	744
参考文献 . . . . .	747
索引 . . . . .	X-1





---

## 本書について

これらのトピックでは、IMS™ リソースの生成、IMS カタログの操作、IMS アクティビティの分析、IMS ロギングの管理、IMS データベース・リカバリー管理 (DBRC) 機能の実行、IMS ネットワーク・サービスの保守、および動的リソース定義 (DRD) の使用のために、IMS システムで使用できるユーティリティの参照情報を提供しています。

この情報は、IBM® Knowledge Center で参照できます。

---

## 前提知識

本書を使用する際には、z/OS®、および IMS の概念、機能、アクセス方法の知識が必要です。前提資料は次のとおりです。

- IMS V15 コミュニケーションおよびコネクション
- IMS V15 データベース管理
- IMS V15 システム管理

z/OS の詳細については、IBM Knowledge Center の「z/OS basic skills」トピックを参照してください。

IMS の基本概念を理解するには、「*An Introduction to IMS*」(IBM Press 出版)をお読みになると役立ちます。

IBM では、IMS の学習に役立つような講習会や自習講座を数多く提供しています。利用可能な講習の詳しいリストについては、IBM Skills Gateway にアクセスして、IMS を検索してください。

---

## 新規および変更された情報の識別方法

IMS ライブラリーの PDF 資料のほとんどの新規および変更された情報は、左マージン内の文字 (改訂マーカ) によって示されています。「リリース計画」、ならびに「*Program Directory*」および「*Licensed Program Specifications*」の第 1 版 (-00) には、改訂マーカは含まれていません。

改訂マーカは、以下の一般的な規則に従っています。

- 技術的な変更のみにマークが付けられています。形式上の変更や文法的な変更には、マークは付けられていません。
- 段落、構文図、リスト項目、操作手順、または図などの要素の一部が変更された場合、その要素の一部だけの変更であっても、要素全体に改訂マーカが付けられています。
- トピックの変更が 50% を超えた場合には、そのトピック全体に改訂マーカが付けられています (そのため、新規トピックではなくても、新規トピックのように見えることがあります)。

改訂マーカは情報に加えられたすべての変更を示しているとは限りません。削除されたテキストとグラフィックスには、改訂マーカでマークを付けることはできないためです。

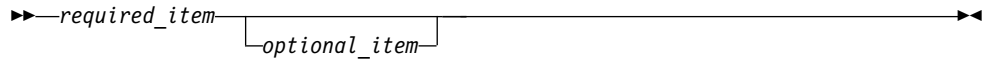
## 構文図の読み方

本書で使用されている構文図には、以下の規則が適用されています。

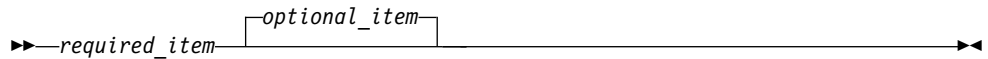
- 構文図は、経路を示す線に沿って、左から右、上から下に読み取ります。以下の規則が使用されます。
  - >>--- 記号は、構文図の始まりを示します。
  - ---> 記号は、構文図が次の行に続くことを示します。
  - >--- 記号は、この構文図が直前の行から続いていることを示します。
  - --->< 記号は、構文図の終わりを示します。
- 必須項目は、水平線 (メインパス) 上に表示されます。



- オプション項目は、メインパスより下に示されます。

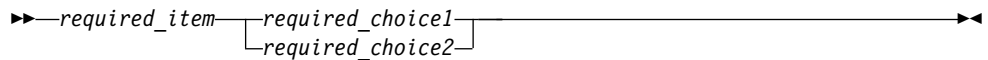


メインパスより上にオプション項目が示されている場合は、その項目が構文エレメントの実行に影響することはなく、読みやすくするためのみの表記です。

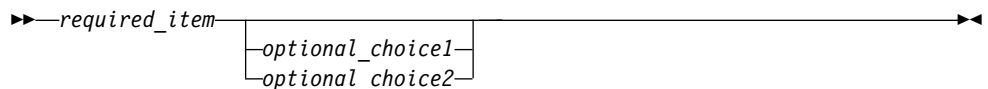


- 複数の項目から選択できる場合は、縦方向に並べて (スタック) 示されます。

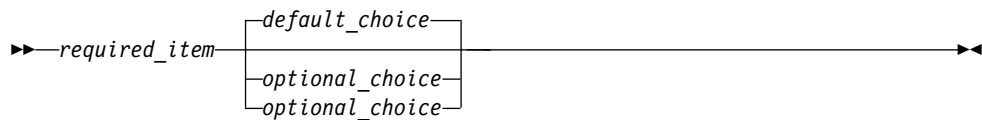
それらの項目の中から 1 つを選択する必要がある場合は、スタックの中の 1 つの項目がメインパス上に表示されます。



それらの項目から 1 つを選択することがオプションである場合は、スタック全体がメインパスの下に表示されます。



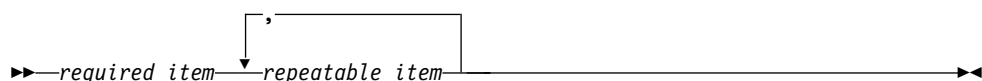
デフォルト項目が含まれている場合、その項目はメインパスより上に示され、他の選択項目はメインパスより下に示されます。



- メインパスの上方にある左に戻る矢印線は、項目が反復可能であることを示します。



反復矢印線にコンマが含まれている場合は、反復項目をコンマで区切る必要があります。

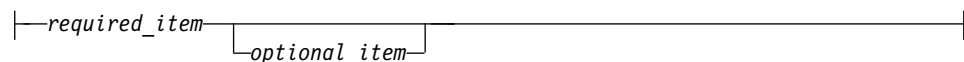


スタック上方の反復矢印線は、スタック内の項目を反復できることを示しています。

- 1 つの構文図を複数のフラグメントに分割しなければならない場合もあります。構文フラグメントはメインの構文図とは別に示されますが、フラグメントの内容は、図のメインパス上にあるものとして読む必要があります。



#### fragment-name:



- IMS では、b 記号は、該当位置に空白が 1 つあることを示します。
- キーワード、および該当する場合はキーワードの最小の省略語は、大文字で表されます。これらは、示されているとおりに入力する必要があります。変数は、すべて小文字のイタリック文字で示されます (例えば、*column-name*)。これらは、ユーザーが指定する名前または値を表します。
- キーワードとパラメーターは、構文図で間に句読点が表示されていない場合は、少なくとも 1 つのスペースで分離します。
- 句読記号、括弧、算術演算子、およびその他の記号は、構文図で示されたとおりに入力します。
- 脚注は、例えば (1) のように、数字を括弧で囲んで示してあります。

## IMS 15 のアクセシビリティ機能

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーが情報技術製品を快適に使用できるようにサポートします。

## アクセシビリティ機能

以下のリストは、IMS 15 を含む z/OS 製品の主なアクセシビリティ機能を示しています。これらの機能は、以下をサポートしています。

- キーボードのみの操作。
- スクリーン・リーダー (読み上げソフトウェア) およびスクリーン拡大鏡によって通常使用されるインターフェース。
- 色、コントラスト、フォント・サイズなど表示属性のカスタマイズ。

## キーボード・ナビゲーション

IMS 15 ISPF パネル機能には、キーボードまたはキーボード・ショートカット・キーを使用してアクセスできます。

TSO/E または ISPF を使用して IMS 15 ISPF パネルをナビゲートする詳細については、「z/OS TSO/E 入門」、「z/OS TSO/E ユーザーズ・ガイド」、および「z/OS 対話式システム生産性向上機能 (ISPF) ユーザーズ・ガイド 第 1 巻」を参照してください。上記の資料には、キーボード・ショートカットまたはファンクション・キー (PF キー) の使用方法を含む、各インターフェースのナビゲート方法が記載されています。それぞれの資料では、PF キーのデフォルトの設定値とそれらの機能の変更方法についても説明しています。

## 関連のアクセシビリティ情報

IMS 15 のオンライン資料は、IBM Knowledge Center で参照できます。

## IBM におけるアクセシビリティ

IBM のアクセシビリティに対する取り組みについては、*IBM Human Ability and Accessibility Center* ([www.ibm.com/able](http://www.ibm.com/able)) を参照してください。

---

## 第 1 部 生成ユーティリティー

生成ユーティリティーは、IMS システムを生成および構成するために使用します。

各トピックでは、ユーティリティーがどのように機能するのかを説明し、その使用における要件と制約事項を定義しています。また、例を提供しています。



---

## 第 1 章 アプリケーション制御ブロック保守ユーティリティー

ACB ライブラリーを使用してランタイム・アプリケーション制御ブロックを管理する IMS システムでは、アプリケーション制御ブロック (ACB) 保守ユーティリティーを使用して、命令実行と直接アクセスの待ち時間を削減し、アプリケーション・スケジューリングのパフォーマンスを向上させます。

このユーティリティーは、カタログを使用してランタイム・アプリケーション制御ブロックを管理する IMS システムでは使用されない場合があります。これは、必要なアプリケーション制御ブロックをオフラインで事前作成する機能を持っています。このため、アプリケーションのスケジュールの際に、そのアプリケーション制御ブロックを直接読み込むことができ、制御をアプリケーション・プログラムに迅速に渡すことができます。

アプリケーション・プログラムの実行をスケジュールするとき、IMS はまず、事前に作成した利用可能なデータベース記述子および PSB の制御ブロックを持っていないければなりません。これらの制御ブロックは DBDGEN および PSBGEN プロシージャによって作成されます。

これらの制御ブロックはマージして、アプリケーション制御ブロック (ACB) と呼ばれる IMS 内部形式に拡張する必要があります。マージと拡張の処理は、ブロック作成 と呼ばれています。

DB/DC 環境に必要なアプリケーション制御ブロックは、GPSB を使用するアプリケーション・プログラムの場合を除いて、事前作成する必要があります。これは、バッチ環境の場合にはオプションです。バッチ環境で IMS.ACBLIB を使用すると、PSBLIB と DBDLIB から動的に ACB を作成するよりも、必要な仮想記憶域は少なくて済みます。

ACB 保守ユーティリティーは、事前作成のブロック (ACB) ライブラリー (IMS.ACBLIB) を保守します。ACB ライブラリーは、プログラム (PSB) とデータベース (DBD) 記述の統合ライブラリーです。制御ステートメントを使用して、すべての PSB の全制御ブロック、特定 PSB の全制御ブロック、または特定の DBD を参照するすべての PSB の全制御ブロックを作成するよう保守ユーティリティーに指示することができます。

ACB 保守ユーティリティーは IMS カタログへのデータの取り込みは行いません。ACB 保守ユーティリティーが ACB を作成した後に IMS カタログにデータを取り込むには、IMS Catalog Populate ユーティリティー (DFS3PU00) を使用します。

ACB 保守ユーティリティーと IMS Catalog Populate ユーティリティーの両方を実行する代わりに、ACB Generation and Catalog Populate ユーティリティー (DFS3UACB) を使用することができます。このユーティリティーは、ACB の作成と IMS カタログへのデータの取り込みを単一のジョブ・ステップで行います。

サブセクション:

- 4 ページの『制約事項』

- 5 ページの『前提条件』
- 5 ページの『要件』
- 5 ページの『推奨事項』
- 6 ページの『入力と出力』
- 8 ページの『JCL 仕様』
- 9 ページの『ユーティリティ制御ステートメント』
- 13 ページの『戻りコード』

## 制約事項

アプリケーション・プログラムで必要とするものが 1 つの入出力 PCB と 1 つの変更可能代替 PCB のみの場合は、ACB 生成を実行する必要はありません。そのようなアプリケーション (普通は DCCTL 環境で使用) は、GPSB を使用して実行に必要なリソースを定義できます。

GSAM の PSB および DBD は、ACB 生成を使用して事前定義できません。GSAM 用の制御ブロックは標準 IMS データ・セット制御ブロックとは異なるためです。GSAM および非 GSAM データベースを参照する PSB は、ACB 生成を使用して事前定義し、非 GSAM データベース用の制御ブロックを作成することができます。

ACB 保守ユーティリティは、一部の IMS システム・リソースを使用しますが、システム全体は使用しません。IMS.PSBLIB および IMS.DBDLIB はデータ・セットを共用しています。IMS.ACBLIB は、専用でなければなりません。このユーティリティは、アクティブ IMS システムに並行して割り振られていない ACB ライブラリーを使用してのみ実行できます。

IMS.ACBLIB は、修正されるので、このプログラムの実行時に他の目的で使用することはできません。IMS.ACBLIB は区分データ・セットであり、ディレクトリー内の必要なリンク情報を伝えます。保守の目的で、オペレーティング・システム (IEHMOVE) およびデータ・セット (IEBCOPY) ユーティリティを使用できます。

異常終了と /ERE の間に、FP DBD をアクティブ ACBLIB に追加してはなりません。IMS の異常終了後にアクティブ ACBLIB に追加された FP DBD には、/ERE 後にアクセスできません。

高速機能副次索引データベースは、シンボリック・ポインターのみをサポートします。HISAM または SHISAM 副次索引データベースの LCHILD ステートメントに PTR=SYMB が指定されていない場合、ACB 保守ユーティリティはメッセージ DFS2292E を発行します。1 次 DEDB データベースとその副次索引データベースは ACBLIB から削除されます。

高速機能副次索引のユーザー区画グループの場合、同じユーザー区画グループ内に含まれる副次索引データベースは、HISAM のみ、または SHISAM のみのいずれかでなければなりません。LCHILD ステートメントでは、メッセージの DBD dbdname で識別されている同じユーザー区画グループに HISAM 副次索引データベースと SHISAM 副次索引データベースの両方が含まれています。1 次 DEDB データベースとその副次索引データベースは ACBLIB から削除されます。



PROCSEQD オペランドが指定された PCB において、ルート・セグメントからの物理パスに沿ってターゲット・セグメントの直接的な親となるセグメント以外のセグメントに対して、またはターゲット・セグメントの子セグメントに対して SENSEG ステートメントが指定されると、ACB 保守ユーティリティは無効な SENSEG ステートメント指定を検出します。ACB 保守ユーティリティはメッセージ DFS2295E を発行します。ACBLIB 内で、メッセージ DFS2295E で識別された PSB が削除されます。

高速機能 HISAM 副次索引データベースまたは高速機能 SHISAM 副次索引データベースではユーザー区画化が要求されます。しかし、PCB ステートメントの PROCSEQD= パラメーターに指定されたユーザー区画データベースが、1 次 DEDB データベース DBD 内の LCHILD ステートメント上の NAME= パラメーターで定義されたユーザー区画グループ内の最初のユーザー区画ではありません。ACB 保守ユーティリティはメッセージ DFS2366E を発行します。ACBLIB 内で、この 1 次 DEDB データベースとその副次索引データベースが削除されます。

PSB では 1 次 DEDB データベースの PCB ステートメントに PSELOPT= パラメーターが指定されており、ユーザー区画化は要求されていません。1 次 DEDB データベースでは、1 次 DEDB DBD 内の LCHILD ステートメント上の NAME= パラメーターに副次索引データベースが 1 つだけ指定されています。ACB 保守ユーティリティはメッセージ DFS2367E を発行します。ACBLIB 内で、メッセージで識別された PSB が削除されます。

## 前提条件

ACB 保守ユーティリティは、IMS.PSBLIB 内の PSB または IMS.DBDLIB 内の DBD を変更しません。PSB または DBD を変更する場合に、関連する PSB または DBD にも変更が必要であれば、このユーティリティを実行する前にそれらの変更を行う必要があります。IMS.ACBLIB での追加、変更、および削除は、オンライン変更ユーティリティとコマンドを使用すれば、IMS を停止せずに行うことができます。

PSB の変更を行うと、影響を受けるアプリケーション・プログラムの修正も行わなければならない場合があります。例えば、DBD でセグメント名が変更されると、そのセグメントに依存するすべての PSB の SENSEG ステートメントを変更する必要があります。

このデータベースを使用するアプリケーション・プログラムも修正しなければならない場合があります。

## 要件

IMS は、データ・セット許可に関する z/OS 規則に準拠しています。ある IMS ジョブ・ステップの許可を与える場合は、そのジョブ・ステップで使用されるすべてのライブラリーも許可する必要があります。ある IMS バッチ領域を無許可で実行するときは、IMS.SDFSRESL に許可不要ライブラリーを連結してください。

## 推奨事項

IMS システム内で IMS カタログを使用可能にする場合は、ACBCATWK DD ステートメントによって出力データ・セットを指定して、ACB 保守ユーティリティが

現在の実行中に生成する ACB メンバーのリストを記録するようにしてください。生成された ACB メンバーに関するこのレコードを DFS3PU00 ユーティリティへの入力として提供することで、IMS カタログへのデータの取り込みに要する時間が大幅に短縮されます。

## 入力と出力

次の図は、入出力データ・セットと命名要件との機能上の関連性を示したものです。ACB 保守ユーティリティは、IMS.DBDLIB データ・セット、IMS.PSBLIB データ・セット、SYSIN 制御ステートメント、COMPCTL IEBCOPY 制御ステートメント、および SYSPRINT メッセージから入力を受け取ります。ACB 保守ユーティリティは、SYSUT3 および SYSUT4 IEBCOPY ユーティリティ・データ・セット、および IMS.ACBLIB データ・セットに出力を出します。

IMS カタログが使用可能になっている IMS システムでは、ACB 保守ユーティリティがオプションで、生成された ACB メンバーのリストを ACBCATWK DD ステートメントによって参照されているデータ・セットに出力することができます。生成された ACB メンバーのリストを DFS3PU00 ユーティリティが入力として読み取ることで、IMS カタログへのデータの取り込みに要する時間が大幅に短縮されます。

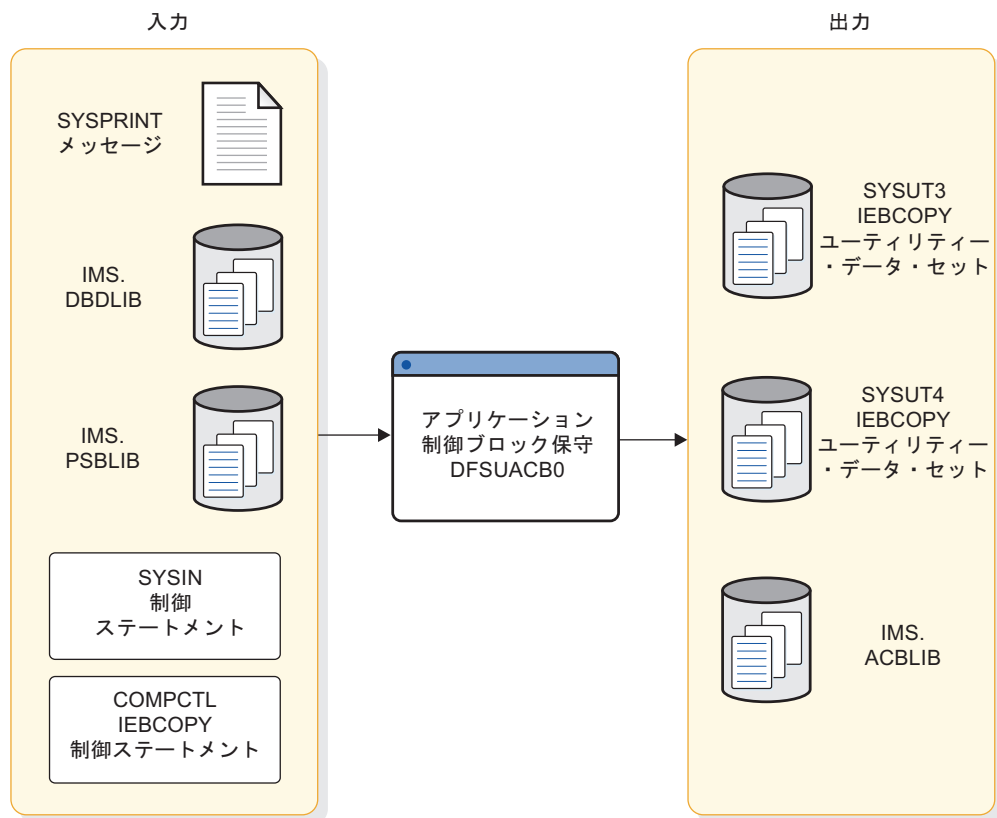


図 1. ACB 保守ユーティリティの入力と出力

### ACB 生成プロシージャー

システム定義の一部として、次の図に示すプロシージャーが作成されます。これは、IMS システム定義のステージ 2 で IMS.PROCLIB プロシージャー・ライブラリーに入れられます。

次の例は ACBLIB の保守のためのプロシージャーです。

```
//      PROC SOUT=A,COMP=,RGN=4M,SYS2=
//G     EXEC PGM=DFSRR00,PARM='UPB,&COMP',
//      REGION=&RGN
//SYSPRINT DD SYSOUT=&SOUT
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//IMS    DD DSN=IMS.&SYS2.PSBLIB,DISP=SHR
//      DD DSN=IMS.&SYS2.DBDLIB,DISP=SHR
//IMSACB DD DSN=IMS.&SYS2.ACBLIB,DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4 DD UNIT=SYSDA,SPACE=(256,(100,100)),
//      DCB=KEYLEN=8
//COMPCTL DD DISP=SHR,
//      DSN=IMS.&SYS2.PROCLIB(DFSACBCP)
```

この図では、IMS データ・セットの高位修飾子は IMS です。この高位修飾子は、IMS 生成で与えられるデフォルトです。ただし、ご使用のシステムでの IMS 生成でデフォルトが使用されなかった場合には、IMS データ・セット名の高位修飾子は IMS ではない場合があります。

#### ACB 生成 JCL ステートメント

下記は、ACB 生成プロシージャーを呼び出すのに使用できる JCL ステートメントの例です。

```
//ACBGEN JOB
//      EXEC ACBGEN
//SYSIN  DD *
BUILD PSB=(MYPSB)
```

ACB 生成プロシージャーでは、以下のシンボリック変数を使用しています。

#### SOUT=

SYSOUT クラスを指定します。デフォルトは A です。

#### COMP=

PRECOMP、POSTCOMP、これらの任意の組み合わせにより、必要なインプレスでの圧縮が行われます。デフォルトはありません。

#### RGN=

ACB ユーティリティーを実行するための領域サイズを指定します。この領域サイズは、生成するブロックのサイズによって異なり、通常は 100 から 150 KB です。デフォルトは 4 MB です。

#### SYS2=

オプションの第 2 レベル DS 名修飾子を指定します。指定する場合は、このパラメーターに後書きピリオドを含め、引用符で囲む必要があります。例えば、次のとおりです。

```
SYS2='IMSA.'
```

## JCL 仕様

### EXEC ステートメント

EXEC ステートメントの最初の部分は次の形式でなければなりません。

```
PGM=DFSRR00
```

パラメーター・フィールドは次の形式でなければなりません。

```
PARM='UPB,PRECOMP,POSTCOMP'
```

ここで、PRECOMP は IMS.ACBLIB データ・セットをブロックの作成前に圧縮することを要求し、POSTCOMP はブロックの作成後に圧縮することを要求します。

「UPB」は、ブロック保守ユーティリティーが制御を受け取ることを示します。このパラメーターは必須です。PRECOMP および POSTCOMP はオプションで、任意に組み合わせて指定できます。

### DD ステートメント

#### ACBCATWK

ACB 生成中に ACB ライブラリーに書き込まれる ACB メンバーのリストが入る、オプションの作業データ・セットを定義します。

ACBCATWK データ・セットは、ACB メンテナンス・ユーティリティーの出力データ・セットおよび DFS3PU00 ユーティリティーの入力データ・セットです。

ACBCATWK データ・セットは、DFS3PU00 ユーティリティーのパフォーマンスを向上させるために指定します。DFS3PU00 ユーティリティーは名前のリストを使用して、IMS カタログ内のどのレコードを挿入または更新する必要があるか判別します。ACBCATWK データ・セットが指定されない場合、DFS3PU00 ユーティリティーは、IMSACBxx DD ステートメントで参照されている ACB ライブラリー内のすべてのメンバーを処理します。

#### COMPCTL DD

PRECOMP または POSTCOMP を指定した場合に、IEBCOPY が使用する制御入力データ・セットを定義します。

EXEC ステートメントのパラメーターで PRECOMP と POSTCOMP の両方が要求される場合は、このデータ・セットを再読み取りオプションでクローズできなければなりません。

このデータ・セットには、次の形式の制御ステートメントが含まれている必要があります。

```
COPY INDD=IMSACB,OUTDD=IMSACB
```

#### DFSRESLB DD

IMS SVC モジュールが入っている許可ライブラリーを指します。IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) で許可する必要があります。この DD ステートメントには、許可ライブラリー中になければならない IMS SVC モジュール用の許可ライブラリーを指定します。JOBLIB または STEPLIB ステートメントは、IMS バッチには許可は不要です。

#### **IMS DD**

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。

#### **IMSACB DD**

単一 ACB ライブラリー・データ・セットを定義します。

制約事項: このデータ・セットは修正されるため、他のジョブと共用することはできません。

#### **STEPLIB DD**

IMS.SDFSRESL を指します。ここには、IMS 中核と必須の IMS モジュールが入っています。IMS.SDFSRESL に連結された無許可ライブラリーがあるために STEPLIB が無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

#### **SYSIN DD**

入力制御ステートメント・データ・セットを定義します。これらのデータ・セットは、テープ・ボリューム、直接アクセス装置、またはカード読取装置に置くか、入力ストリームで経路指定することができます。入力は、80 の倍数でブロック化できます。実行時に、このユーティリティーは必要な数の制御ステートメントを処理できます。

#### **SYSPRINT DD**

出力メッセージ・データ・セットを定義します。

SYSPRINT DD ステートメントで DASD データ・セットまたはテープ・データ・セットを参照する場合、そのデータ・セットのブロック・サイズを **DCB** パラメーターの **BLKSIZE** サブパラメーターで制御できます。**BLKSIZE** 値を指定する場合、その値は正確に 121 の倍数とする必要があります。そうでない場合はシステム ABEND013-20 が発生する可能性があります。DASD データ・セットの **BLKSIZE** を省略すると、システムが決定するブロック・サイズが使用されます。**LRECL** パラメーターに指定される値に関係なく、ユーティリティーは常にレコード長 121 を使用します。

#### **SYSUT3 DD**

PRECOMP または POSTCOMP のいずれかが EXEC ステートメントで指定されている場合に必要な作業データ・セットを定義します。

#### **SYSUT4 DD**

機能は SYSUT3 と同じです。

#### **DFSACBCP 制御ステートメント**

SMP/E 処理によって作成される DFSPROCB JCL を使用すると、以下の制御ステートメントが作成されます。

```
COPY INDD=IMSACB,OUTDD=IMSACB
```

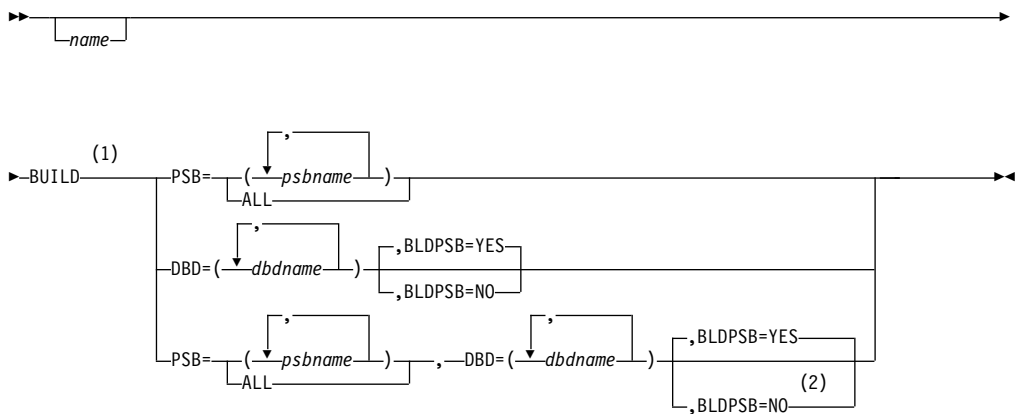
ACB 生成プロシージャは、DFSACBCP を用いて ACBLIB を圧縮します。

#### **ユーティリティー制御ステートメント**

ユーティリティー JCL に、ACB メンバーを作成または削除する制御ステートメントを指定します。制御ステートメントは以下のガイドラインに準拠している必要があります。

- ステートメントは 1 から 71 桁に収め、カード・イメージでコーディングします。
- 制御ステートメントには、任意に名前 (1 桁目から開始) を含めることができます。
- ステートメントを継続するためには、72 桁目に非空白文字を入力し、次の行の 16 桁目からステートメントを開始します。
- 命令フィールドの前後には、1 つ以上の空白を入れなければなりません。
- パラメーターは、1 つ以上の PSB 名または DBD 名から構成され、その前後にも 1 つ以上の空白を入れなければなりません。
- コンマ、括弧、空白は、区切り文字としてのみ使用できます。
- コメントを制御ステートメントの最後のパラメーターの後に書くことができます。パラメーターとコメントは 1 つ以上の空白で分離します。

### ACB 保守ユーティリティー構文: **BUILD** 形式



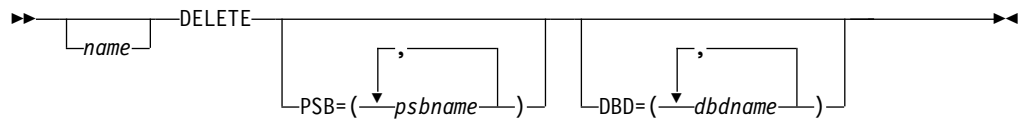
#### 注:

- 1 ACB メンテナンス・ユーティリティーの SYSIN 入力制御ステートメントに、先入れ先出し (FIFO) プロセスはありません。同じアプリケーション制御ブロック (ACB) 生成ジョブの SYSIN 制御ステートメントで、BUILD PSB= と BUILD DBD= の両方のパラメーターを指定すると、DBD= オペランドが先にブロック・ビルダー・ユーティリティー・プログラムに渡されます。SYSIN 制御ステートメントのどこに DBD= オペランドが入力されたかに関係なく、ACBLIB データ・セットで DBD の準備ができていない場合、DFS0586I が出されます。
- 2 同じステートメント内にパラメーター PSB=ALL と BLDPSB=NO を指定した場合、IMS はすべての PSB を作成します (BLDPSB=NO は無視されます)。同様に、同じ ACBGEN ジョブで 1 つの DBD に BLDPSB=NO パラメーターを指定し、別の DBD に BLDPSB=YES パラメーターを指定した場合、IMS は変更された DBD を参照するすべての PSB を作成し、BLDPSB=NO の指定を無視します。

次の例では、CUSTOMER および ORDER DBD に関連付けられているすべての PSB は、CUSTOMER DBD に BLDPSB=NO が指定されている場合でも再作成されます。

BUILD DBD=(CUSTOMER),BLDPSB=NO  
BUILD DBD=(ORDER),BLDPSB=YES

ACB 保守ユーティリティー構文: **DELETE** 形式



ACB 保守ユーティリティー・パラメーター

#### **BUILD**

指定した PSB (指定した DBD を指す) 用にブロックを作成することを指定します。

#### **DELETE**

ブロックを ACBLIB データ・セットから削除することを指定します。指定した PSB、および指定した DBD を指すすべての PSB が削除されます。

ACBLIB データ・セットからブロックを削除しても、IMS カタログ内の対応するレコードは削除されません。

#### **PSB=ALL**

現在 IMS.PSBLIB 内にあるすべての PSB 用に、そのブロックを作成することを指定します。このパラメーターは、初期 IMS.ACBLIB を作成するために使用します。PSB=ALL パラメーターを指定すると、すべての PSB および DBD (および他のモジュール) は ACBLIB データ・セットから削除され、それらのスペースは再利用できるようになります。その後、ACB 生成が PSBLIB データ・セット内のすべての PSB に関して実行されます。このパラメーターを DELETE ステートメントと一緒に使用しないでください。

制約事項: BUILD PSB=ALL パラメーターを SYSIN 制御ステートメントに指定する場合には、すべての PSB を単一の PSBLIB データ・セット内に置く必要があります。連結された PSBLIB は、IMS DD ステートメントでは認識されません。

#### **PSB=(psbname)**

この制御ステートメントで指定されているすべての PSB について、ブロックを作成または削除することを指定します。このタイプの制御ステートメントは必要なだけ実行要求することができます。このパラメーターは、新しい PSB を IMS.ACBLIB に追加したり、使用しなくなった PSB を削除したりします。単一のパラメーターを与える場合には、括弧を省略することができます。

#### **DBD=(dbdname)**

この DBD について、およびこの DBD を直接または論理関係によって間接的に参照するすべての PSB について、ブロックを作成または削除することを指定します。作成する DBD は、既に IMS.ACBLIB データ・セット内に存在している必要があります。参照する PSB も、既に IMS.ACBLIB データ・セット内に存在している必要があります。IMS.PSBLIB データ・セットに新規に追加された PSB は、PSB オペランドによって参照される必要があります。PSB を削除しても、その PSB が参照している DBD は削除されないため、このパラメーターを使用して特定の DBD を削除することができます。ただし、DBD を削除または

作成すると、指定した DBD を参照する IMS.ACBLIB データ・セット内のすべての PSB が、要求タイプに基づいて再作成または削除されることになります。単一のパラメーターを与える場合には、括弧を省略することができます。

例 1: PSB-a で DBD-a および DBD-b を参照します。DBDGEN が DBD-a および DBD-b に関して行われ、更新済み DBD は DBDLIB にあります (しかし、ACBLIB にはまだありません)。DBD-a を ACB 生成で指定することによって、DBD-a が ACBLIB 内に再ビルドされ、参照 PSB (この場合は、PSB-a) があれば、いずれもやはり再ビルドされます。確かに PSB-a は再ビルドされはしても、DBD-b が明確に ACBLIB 内に再ビルドされたわけではないため、ACBLIB は使用不能です。DBD-b を ACBLIB 内に再ビルドするためには、ACB 生成でそれを明示的に指定する必要があります。たとえ参照 PSB が完全に更新されている場合でも、やはり更新済み DBD を ACB 生成で明示的に指定する必要があります。

このプログラムが処理するすべての PSB は、IMS.ACBLIB データ・セット内にメンバーを生成します。PSB が参照する DBD は、特定の DBD が初めて処理されたとき、あるいはある DBD 名が制御ステートメントに現れたときに、メンバーを生成します。同じ DBD を参照するすべての PSB は、参照されている DBD に PSB を接続するために、ディレクトリー項目内の情報を伝えます。

論理 DBD は IMS.ACBLIB 内にメンバーを持っておらず、BUILD または DELETE 制御ステートメントで参照することはできません。

例 2: 以下の例では、BLDPSB パラメーターの使用方法を示します。

- CUSTOMER という名前の DBD が変更されたため、CUSTOMER を参照するすべての PSB を再作成する必要があります。

```
BUILD DBD=CUSTOMER,BLDPSB=YES
```

- ORDER および INVENTORY という名前の DBD が変更されたため、これらの DBD を参照するすべての PSB を再作成する必要があります。

```
BUILD DBD=(ORDER,INVENTORY),BLDPSB=YES
```

DBD を IMS.DBDLIB 内で置換する場合には、それを BUILD DBD 制御ステートメントの中にも含める必要があります。この方法のみが BUILD PSB=ALL を行わずに、IMS.ACBLIB 内で DBD を置換できる有効な方法です。

DBDLIB の修正 DBD を参照する BUILD PSB を実行すると、ACBLIB 内で置換された PSB は、更新版の DBD を含むことになります。この BUILD PSB が、変更された DBD についての BUILD DBD の前にくると、ACBLIB は、別の版の DBD を持つ PSB を含むことになります。BUILD PSB に指定された PSB は更新された DBD を含みますが、作成されていない PSB は古い DBD を参照することになります。ACBLIB 上の PSB 用の DBD が、アクセスされるデータベースと一致しない場合には、結果は予測できません。(例えば、変更された DBD でセグメント・コードが追加されるか、削除されたために U852 異常終了が起きます。) したがって、後で使用するために DBDGEN を実行する場合は、データベースに変更が反映されない限り、変更された DBD を参照する PSB を作成してはなりません。

物理 DBD が変更され、しかもそれが BUILD DBD ステートメントで参照されている場合は、変更された DBD (1 次索引と副次索引を含めて) と論理的に関



連するすべての物理 DBD も BUILD DBD ステートメントで参照する必要があります。しかし、それらの DBD と論理的に関連する DBD を再作成する必要はありません。

次の図に、一部の物理データベース間の関係が図示してあります。ここで A が変更された DBD です。次のような関係が存在しています。

- B と C は、A と論理的に関連しています。
- D は B と論理的に関連しています。
- E は C と論理的に関連しています。
- D と E は、A と論理的に関連していないので、BUILD DBD ステートメントで参照されていません。

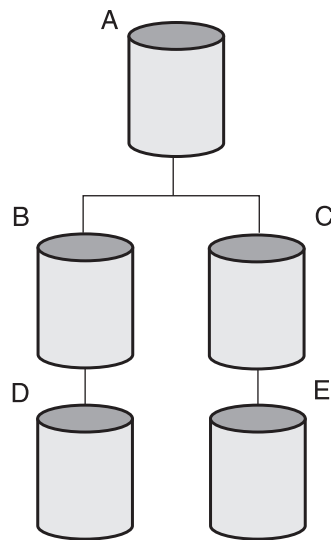


図 2. 論理的に関連した物理データベースの例

#### **BLDPSB=**YES | NO

BUILD DBD=(*dbdname*) ステートメント内で、変更された DBD を参照するすべての PSB を ACBGEN で再作成するかどうかを指定します。

#### **YES**

BUILD DBD=(*dbdname*) ステートメント上で、変更された DBD を参照するすべての PSB を ACBGEN で再作成するよう指示します。デフォルトは BLDPSB=YES です。

**NO** 変更された DBD がデータベースの物理構造を変更していない場合、その DBD を参照する PSB を ACBGEN で再作成しないよう指示します。高速機能 DEDB の場合、PSB が再作成されるのは、セグメント数またはデータベースのセグメント内のフィールド数あるいはその両方が変更される時に限られます。高速機能 MSDB の場合、データベースの物理構造が変更されても、参照する PSB は再作成されません。


## 戻りコード


ACB 生成プロシージャは、次のコードを戻します。

コード 意味

- 0 全ての操作が正常に終了
- 4 1 つ以上の警告メッセージが出ている
- 8 1 つ以上のブロックを作成できなかった
- 16 重大エラーでプログラムが中止された

関連概念:

 アプリケーション制御ブロック (ACBGEN) の作成 (データベース管理)

 ACBLIB データ・セットの割り振り (システム定義)

関連資料:

367 ページの『第 6 章 ACB Generation and Catalog Populate ユーティリティー (DFS3UACB)』

425 ページの『第 11 章 IMS Catalog Populate ユーティリティー (DFS3PU00)』

## ACB 保守ユーティリティーの例

これらの例では、ACB 保守ユーティリティーを使用して PSB のブロックを作成または削除する方法を示します。

### すべての PSB 用のブロックを作成する例

この例では、現在 IMS.ACBLIB 内に存在するすべてのブロックが削除され、それらのスペースを再利用して、現在 IMS.PSBLIB 内に存在するすべての PSB 用に新しいブロックが作成されます。このオプションは、通常は、IMS.ACBLIB データ・セットの初期作成に使用します。スペースがまだ ACBLIB に割り振られていなければ、スペース・パラメーターおよび DISP=NEW を IMSACB DD ステートメントに指定してください。

```
//BLDBLKS JOB
//*
//STEP EXEC ACBGEN,SOUT=A
//SYSIN DD *
        BUILD PSB=ALL
/*
```

### 特定の PSB 用のブロックを作成する例

この例では、PSB1、PSB2、および PSB3 のブロックを作成します。IMS.ACBLIB 内の他の PSB はすべて未変更のままです。これらの PSB が参照する DBD が IMS.ACBLIB 内に存在しなければ、それらは追加されます。さらに、DBD5 および DBD6 を ACBLIB から削除します。ブロックの作成後に IMS.ACBLIB は圧縮され、削除は実行されます。

```
//BLDBLKS JOB
//*
//STEP EXEC ACBGEN,SOUT=A,COMP=POSTCOMP
//SYSIN DD *
        BUILD PSB=(PSB1,PSB2,PSB3)
        DELETE DBD=(DBD5,DBD6)
/*
```

## PSB 削除とブロック再作成の例

この例では、IMS.ACBLIB データ・セットから PSB1 を削除し、DBD4 を参照する IMS.ACBLIB データ・セット内のすべての PSB にそのブロックを再作成させます。PSB1 が DBD4 を参照する場合は、PSB1 は IMS.ACBLIB から削除されたばかりなので、ブロックは再作成されません。PSB1 は IMS.PSBLIB から削除されません。IMS.ACBLIB は、ブロック作成の前後で圧縮されます。

```
//BLDBLKS JOB
/*
//STEP EXEC ACBGEN,SOUT=A,COMP='PRECOMP,POSTCOMP'
//SYSIN DD *
        DELETE PSB=PSB1
        BUILD DBD=DBD4
/*
```

---

## DOPT PSB の管理

ACB 保守ユーティリティーによって生成される動的オプション (DOPT) PSB を管理する方法は、IMS システムがアクティブ ACB のリポジトリとして ACB ライブラリーまたは IMS カタログのどちらを使用するかによって異なります。


ACB ライブラリーを使用する IMS システムでは、DOPT PSB 用のブロックが、1 次 ACBLIB データ・セットの後に連結された ACBLIB データ・セットに含まれていなければなりません。

IMS カタログを使用する IMS システムでは、DOPT PSB 用のブロックが他のすべてのアクティブ ACB ブロックと共に IMS カタログに格納されます。データ・セットを連結する必要はありません。

DOPT PSB 用に生成される PSB 定義およびブロックは、非 DOPT PSB 用のブロックの場合と同じです。この PSB は、APPLCTN ステージ 1 システム定義マクロ内か、あるいは IMS タイプ 2 コマンド CREATE PGM または UPDATE PGM 内で、DOPT パラメーターによって動的 PSB として定義されます。


PSB に関連付けられているプログラムが BMP 従属領域または JBP 従属領域に合わせてスケジュールされている場合は、プログラム作成ユーザー出口 (PGMCREAT) を使用し、PGMCREAT パラメーター・リストに PGMCR\_PF1\_DOPTY ビットを設定することによって PSB を動的と定義することもできます。

関連概念:


 [オンライン・アプリケーション・プログラムの宣言 \(システム定義\)](#)

関連資料:

425 ページの『第 11 章 IMS Catalog Populate ユーティリティー (DFS3PU00)』

 [APPLCTN マクロ \(システム定義\)](#)

 [CREATE PGM コマンド \(コマンド\)](#)

 [PGMCREAT ユーザー出口ルーチン・タイプ \(出口ルーチン\)](#)

## ACB ライブラリーを使用する IMS システム内の DOPT PSB の管理

ACB ライブラリー (ACBLIB) データ・セットを使用する IMS システムで動的オプション (DOPT) PSB を使用するには、1 次 ACBLIB データ・セットを連結する必要があります。ACBLIB データ・セット連結内の最初の (または 1 次) データ・セットには、すべての非動的 (非 DOPT) PSB 用のブロックが含まれている必要があります。それ以降の DOPT ACBLIB データ・セットにはすべての動的オプション (DOPT) PSB 用のブロックが含まれている必要があります。

注: アクティブおよび非アクティブの DOPT ACBLIB データ・セットには必ず異なる名前を付けて、DOPT ACBLIB データ・セットに行われた変更がオンライン変更によって確実に検出されるように図る必要があります。

1 次 ACBLIB データ・セットは、連結の最初の DD ステートメントです。1 つの PSB または DBD を連結データ・セットに BUILD するとき、DD ステートメントを 1 つだけ ACB 保守ユーティリティに与えてください。

システム初期設定時には、1 次または DOPT ACBLIB データ・セットのいずれかに、すべての非動的 PSB とすべての DBD が作成されていなければなりません。


トランザクション・スケジュール時まで、スケジュールされる DOPT PSB は、DOPT ACBLIB データ・セットに作成されている必要があります。DOPT PSB を 1 次 ACBLIB データ・セットの中に作成してはなりません。

システム内のすべての PSB が DOPT PSB である場合は、1 次 ACBLIB をダミー PDS データ・セットにしてください。DOPT ACBLIB には、すべての DBD および PSB のブロックを入れます。BMP、MPP、または IFP JCL の中で DIRCA サイズ・パラメーターを設定してください。

システム内の一部 (全部ではない) の PSB が DOPT PSB である場合は、両方の ACBLIB データ・セットが DBD と PSB のブロックを含むことになります。PSB を 1 つの ACBLIB データ・セット内に BUILD 作成すると、その PSB が参照する DBD のブロックもそのデータ・セット内に作成されます。その DBD が既に別の ACBLIB データ・セット内に作成されている場合には、その DBD に関して 2 組のブロックを持つことになります。DL/I は、BLDL を実行してその DBD のブロックを使用する際に、1 次 ACBLIB 内のブロックのセットを使用します。

DOPT PSB を使用するプログラムの終了処理の間に、PSB は PSB プールから削除されます。

関連資料:

 プロシージャの DBLDL= パラメーター (システム定義)

## IMS 管理の ACB 環境での DOPT PSB の管理

ACB の IMS 管理が使用可能な場合、ACB 保守ユーティリティによって生成される動的オプション (DOPT) PSB のブロックは、他のタイプの PSB のブロックと同じ方法で (つまり、IMS カタログに) 保管できます。IMS は ACB ライブラリーを使用せず、別個の連結データ・セットに DOPT PSB を配置する必要もありません。

ACB 保守ユーティリティーによって生成される新規または変更済みの DOPT PSB 用の ACB ブロックを IMS カタログに追加するには、それらが含まれている ACBLIB データ・セットを IMS Catalog Populate ユーティリティー (DFS3PU00) に入力として提供します。

入力 ACBLIB に DOPT PSB のみが含まれている場合、IMS Catalog Populate ユーティリティーで、それらの DOPT PSB をすぐに使用可能にすることができます。IMPORT DEFN コマンドを発行したり IMS を再始動したりする必要はありません。

入力 ACBLIB に DBD および非 DOPT PSB が DOPT PSB と共に含まれている場合は IMPORT DEFN SOURCE(CATALOG) コマンドを発行することによって、また、IMS Catalog Populate ユーティリティーが UPDATE オプションを指定して実行されている場合には IMS を再始動することによって、DOPT PSB を他のリソース・タイプと共にアクティブにする必要があります。

DOPT PSB のみを追加するには、以下のようにします。

1. DOPT PSB を ACBLIB データ・セット内に生成します。
2. DBD または 非 DOPT PSB 用のブロックを ACBLIB データ・セットから削除します。
3. 以下を指定して、IMS Catalog Populate ユーティリティー用の JCL を作成します。
  - a. IMSACB01 DD ステートメントに DOPT PSB が含まれている ACBLIB。ACBLIB データ・セットを連結する場合は、DBD または非 DOPT PSB を含む ACBLIB データ・セットを組み込まないでください。
  - b. **UPDATE** オプションと **SHARE** オプションを指定した **MANAGEDACBS** 制御ステートメント。


入力 ACBLIB に DBD または 非 DOPT PSB が含まれている場合に **SHARE** オプションを指定すると、これらのリソースは IMS システムでアクティブになりません。IMS システムが再始動されるか、他のイベントによってシステムがメモリーにリソースをロードするまで、IMS システムでは新規リソースを使用できません。

**SHARE** オプションを使用するには、その前に、PDSE データ・セットの拡張共用をサポートするように z/OS システムを構成しておく必要があります (z/OS SYS1.PARMLIB データ・セットの IGDSMSxx メンバーで PDSESHARING(EXTENDED) を指定します)。


4. IMS Catalog Populate ユーティリティーを実行します。

**UPDATE** オプションおよび **SHARE** オプションが指定されていると、IMS Catalog Populate ユーティリティーは、リソースのステージングを事前に行わずに DOPT PSB 用の ACB を IMS カタログ・ディレクトリーに直接追加します。ユーザーが IMPORT DEFN SOURCE(CATALOG) コマンドを発行する必要もありません。DOPT PSB は、共用モードで IMS カタログ・データ・セットに追加されるので、オンラインの IMS システムは IMS Catalog Populate ユーティリティーが DOPT PSB を追加している間も IMS カタログへのアクセスを継続することができます。

関連タスク:

 z/OS: 複数システム環境における拡張 PDSE 共用の指定

関連資料:

 IMS Catalog Populate ユーティリティ (DFS3PU00) (システム・ユーティリティ)

---

## 第 2 章 データベース記述 (DBD) 生成ユーティリティ

DBD ライブラリーを使用する IMS システムでは、データベース記述生成 (DBDGEN) ユーティリティを使用してデータベースを定義し、アプリケーション・プログラムでデータベースを使用できるようにします。

このユーティリティは、カタログを使用してランタイム・アプリケーション制御ブロックを管理する IMS システムでは使用されない場合があります。

データベース記述 (DBD) は、アプリケーション・プログラムが必要とするデータベース情報のすべてを含んでいる DL/I 制御ブロックです。

データベース記述 (DBD) は、特殊マクロ命令をコーディングすることにより作成します。これらのマクロは、DBDGEN ユーティリティへの入力になります。

各物理データベースの記述に使用できる物理 DBD は 1 つのみです。それ以上使用すると、0850、0852、または 0853 などのユーザー異常終了が起きます。実行時に、DL/I は DBD を使用して、内部制御ブロックの集合を作成します。

DBDGEN ユーティリティは、各 DBD を次のデータベース情報を使って定義します。

- セグメント・タイプ
- セグメント・タイプ間の物理関係と論理関係
- データベース編成とアクセス方式
- データベースの物理的特性
- 選択された出口ルーチンの名前およびデータ・オプションを定義する
- データベースおよびそのデータベースに保管されているデータを記述するメタデータ

サブセクション:

- 『制約事項』
- 『前提条件』
- 20 ページの『要件』
- 20 ページの『推奨事項』
- 20 ページの『入力と出力』

### 制約事項

現在、DBDGEN ユーティリティについて文書化されている制約事項はありません。

### 前提条件

現在、DBDGEN ユーティリティについて文書化されている前提条件はありません。

## 要件

DBDGEN 入力の構造化には厳密な規則があります。データベースごとに別々の入力セットが必要です。

## 推奨事項

現在、DBDGEN ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

DBDGEN プログラムは、いくつかのタイプの制御ステートメントを受け入れません。

- DBD ステートメントは、記述するデータベースの名前を指定し、データベース編成に関する情報を DL/I に与えます。
- DATASET ステートメントは、非 DEDB DBDGEN 入力レコード構造の中でのみ使用します。DATASET ステートメントは、データベース内のデータ・セット・グループを定義します。1 つ以上の DATASET ステートメントが DBD ステートメントの後に続きます。
- AREA ステートメントは、DEDB DBDGEN 入力レコード構造の中でのみ使用します。AREA ステートメントは、データベース内の区域を定義します。1 つ以上の AREA ステートメントが DBD ステートメントの後に続きます。
- SEGM ステートメントは、指定されたデータベースのセグメントを定義します。SEGM ステートメントは、次のステートメントと一緒に使用されます。
  - FIELD
  - XDFLD
  - LCHILD
  - DFSMARSH
  - DFSMAP
  - DFSCASE

各ステートメントで、セグメントまたはセグメント内のフィールドのさまざまな側面を定義します。

- DBDGEN ステートメントは、DBDGEN 制御ステートメントの終わりを表します。
- FINISH は、互換性を保つために入力ストリーム内に残されているオプションのステートメントです。
- END ステートメントは、入力ステートメントの終わりに達したことを z/OS アセンブラーに伝えます。

3 つのタイプの印刷出力および 1 つのロード・モジュール (IMS.DBDLIB という名前の区分データ・セットのメンバーになる) が、DBD 生成で作成されます。これらの出力のそれぞれについては、以下のセクションで説明します。

制御ステートメントのリスト



これは、このジョブ・ステップへの入力ステートメント・イメージをリストしたものです。

## 診断

各ステートメントの処理中にエラーが検出されると、診断メッセージが出されます。これらのメッセージは、読み取られた最後のステートメントのイメージの直後に印刷されます。このメッセージは、直前のステートメント、または前のステートメントのグループのいずれかを指し示します。各ステートメントごとに複数のメッセージを印刷することも可能です。

この場合には、それらのメッセージは互いに出力リスト上に続きます。すべてのステートメントが読み取られた後に、デック全体の妥当性についてさらに検査が行われます。これによって、1 つ以上の診断メッセージがさらに出される場合があります。

エラーが検出されると、診断メッセージが印刷され、ステートメントがリストされ、残りの出力が抑制されます。しかし、DBD 生成の実行が終了する前に、すべての制御ステートメントが読み取られて、検査されます。ステートメント・エラーが検出された場合、DBD 生成のバインド・ステップは処理されません。

## アセンブラー・リスト

DBD 生成の実行で作成される DBD マクロ展開のアセンブラー言語リストが提供されます。アセンブラー言語の PRINT NOGEN ステートメントを含めれば、このリストの印刷出力をなくすことができます。

オペレーティング・システムのアクセス方式として VSAM を使用するデータベースについての DBD 生成では、アセンブラー・リストのページに、データベースのデータ・セットを VSAM に定義するときに必要な一部のパラメーターの推奨値が記載されます。これらの推奨値以外に、CONTROLINTERVALSIZE と RECORDSIZE の値を、パフォーマンスの向上などの特別の理由で指定することができます。すべての ESDS 定義に合わせて RECORDSIZE を変更する必要があります。

制御インターバル・サイズを指定しなかった場合 ( 68 ページの表 4 の GSAM 行の SIZE パラメーターを参照)、このアセンブラー・リストで推奨されたサイズがデフォルトとして使用されます。以下に、HISAM データベースの場合に作成される出力の例を示します。示されているパラメーターは、アクセス方式サービス・プログラムの制御ステートメントで必要な形式です。最初の DEFINE は、キー順データ・セット (KSDS) 用のパラメーターを示し、2 番目の DEFINE は、入力順データ・セット (ESDS) 用のパラメーターを示しています。

VSAM データ・セットを完全に定義するときは、DBD 生成で提供されるものに、データ・セット名のパラメーター (NAME)、スペース割り振りのパラメーター (CYL)、およびボリューム割り当てのパラメーター (VOLUMES) を追加する必要があります。FREESPACE および WRITECHECK などのオプション・パラメーターは、必要であれば含めることができます。

## DBD 生成からのアクセス方式サービス・プログラムのパラメーターの例

/DBD コマンドを使用して、VSAM データベースのオフライン・ダンプを使用可能にするためには、そのデータベースのデータ・セットに対して VSAM DEFINE 操作で SHARE OPTIONS(3) を使用する必要があります。DBD 生成からのアクセス方式サービス・プログラムのパラメーターの例を次に示します。

```

**,* * * * *
**,*
**,*      RECOMMENDED VSAM DEFINE CLUSTER PARAMETERS
**,*
**,* * * * *
**,* * * * *
**,*      *NOTE 1
**,*      DEFINE CLUSTER (NAME(DDI311) -
**,*          INDEXED KEYS (6, 10) -
**,*          RECORDSIZE (680,680) -
**,*          DATA (CONTROLINTERVALSIZE (4096))
**,* *NOTE 1: SHOULD SPECIFY DSNAME FOR DDI311
**,* * * * *
**,* * * * *
**,*      *NOTE 2
**,*      DEFINE CLUSTER (NAME(DDI301) NONINDEXED -
**,*          RECORDSIZE (680,680) -
**,*          CONTROLINTERVALSIZE (4096))
**,* *NOTE 2: SHOULD SPECIFY DSNAME FOR DDI301
**,* * * * *

```

## セグメント・フラグのコード

DBD 生成で生成されたものの確認のために、セグメント・フラグがその特定の DBD 生成出力の中に印刷されます。フラグを解釈すれば、どのポインター・オプションが生成されたか、指定されたセグメントの挿入、削除、および置換の規則、物理子ポインターがこのセグメントの接頭部に確保されているかどうか、そしてセグメントに関連付けられた物理子の数がわかります。セグメント・フラグは、アセンブラ言語の定数定義 (DC) ステートメントとして、出力の中に示されます。この定数は、8 桁の 16 進数とそれに続く SEGMENT FLAGS という記述で定義されます。定数の各対の数字は、16 進バイトです。定数を解釈するときは、次の図に示されているように、最初の 6 桁を 2 進数に変換し、最後の 2 桁を 10 進数に変換してください。

バイト	変換後の値	説明
0		生成されるポインター位置
	1.....	CTR (カウンター)
	.1.....	物理兄弟順方向
	.11.....	物理兄弟順方向および物理兄弟逆方向
	...1....	物理親
	...1...	Logical twin forward
	...11..	Logical twin forward and backward
	.....1.	論理親
	.1.....1	階層順方向
	.11.....1	階層順方向および逆方向
1		セグメント処理規則
	10.....	挿入は物理
	01.....	挿入は仮想

	11.....	挿入は論理
	..10....	Insert nonsequential last
	..01....	Insert nonsequential first
	..11....	Insert nonsequential here at current position
	....10..	Replace physical
	....01..	Replace virtual
	....11..	Replace logical
	.....10	Delete physical
	.....01	Delete virtual
	.....11	Delete logical
	.....00	Bivirtual delete
2	..XX.XXX	予約済み
	1.....	セグメントは対
	.1.....	セグメントは FP DEDB の直接従属
	....1...	Segment's parent has two physical child ポインターを持つ。階層ポインターの指定なし
3	0-254	物理子ポインターが指し示す、 このセグメントの物理子の数

### セグメント接頭部のフォーマットの説明

これらの値を 2 進数と 10 進数に変換して表示すると、次のようになります。

バイト 0	バイト 1	バイト 2	バイト 3
FE	FD	08	0A
11111110	11111101	00001000	10

#### バイト 0

セグメントには、カウンター、物理兄弟順方向および逆方向、論理兄弟順方向および逆方向、物理親、および論理親のポインターがあります。

#### バイト 1

指定された挿入規則および置き換え規則は論理であり、指定された削除規則は仮想です。非順次は現在位置に挿入されます。

#### バイト 2

このセグメントの親の中に、物理子ポインター用の 4 バイトのフィールドが 2 つ確保されています。

#### バイト 3

このセグメントは、10 個の物理子の親です。

DBD 生成の出力には、次のステートメントが入っています。

```
DC X'FEFD080A' SEGMENT FLAGS
```

### ロード・モジュール




DBD 生成は、2 ステップからなるオペレーティング・システム・ジョブです。ステップ 1 はマクロ・アセンブリーの実行で、ステップ 2 への入力となるオブジェクト・モジュールを生成します。ステップ 2 はオブジェクト・モジュールのバインドで、IMS.DBDLIB ライブラリーのメンバーになるロード・モジュールを作成します。

### DBD 生成エラー条件

データベースのタイプごとに示されているオペランドまたはパラメーター以外のものを指定するか、必要なオペランドまたはパラメーターを省略すると、以下の 1 つ以上の条件が起こります。

- DBD 生成で以下の診断メッセージが出されます。
  - 定義中のデータベース・タイプのものではないオペランドまたはパラメーターにフラグを付けるメッセージ
  - 定義中のデータベース・タイプに必須のオペランドまたはパラメーターが省略されていることを指摘するメッセージ
- DBD 生成は完了しますが、DL/I は、定義されたデータベース・タイプにとって該当しないオペランドまたはパラメーターが指定されたため、生成された制御情報を無視します。
- DBD 生成は完了しますが、DL/I は、定義されたデータベースを作成しアクセスすることができません。その理由は、(a) データベースを相互に関連付けようとしたときに、矛盾する制御情報が指定されていた、または (b) アプリケーション・プログラムから見たときのデータベースのセグメント関係の記述が DBD 生成で正しく定義されていないためです。
- DBD 生成は完了し、DL/I はデータベースを作成しアクセスします。しかし、提供される結果が、望んでいたものとは異なっています。欠落制御情報または矛盾する制御情報を見つけた場合に DL/I が実行したデフォルトのアクションが、DBD 生成時に考慮されていなかったアクションである場合に、この条件が起こる可能性があります。

関連概念:

-  DBDGEN ユーティリティーの入力としてのデータベース記述のコーディング (データベース管理)
-  アプリケーション制御ブロック (ACBGEN) の作成 (データベース管理)
-  ACBLIB データ・セットの割り振り (システム定義)

---

## 各データベース・タイプの DBD 生成

DBDGEN ユーティリティーは、このユーティリティーを使用しているデータベースのタイプに基づいて、データベースの DBD を生成します。

下記のデータベース・タイプには DBDGEN ユーティリティーを使用します。

- HSAM (SHSAM を含む)
- GSAM
- HISAM (SHISAM を含む)
- HDAM
- PHDAM
- HIDAM
- PHIDAM
- MSDB
- DEDB
- 索引

- HIDAM の 1 次
- 2 次
- PSINDEX
- 論理

## HSAM/SHSAM の DBD 生成

HSAM データベースの DBD を生成するときは、以下のものを指定します。

- 1 つのデータ・セット・グループ。
- アプリケーションがデータベースからデータを検索するときに使用する入力データ・セットの DD 名。
- データベースをロードするときに使用する出力データ・セットの DD 名。
- データベースの 1 から 255 個のセグメント・タイプ。
- 各セグメント・タイプ内の 0 から 255 個のフィールド。データベース内のフィールド数は最大 1000 フィールド。

HSAM データベースには、以下のものを指定できません。

- データベース内のセグメント間での階層ポインターまたは物理子または物理兄弟ポインターの使用。
- セグメント間の論理関係または索引関係の使用。

オプションで、1 つの固定長セグメント・タイプのみを含むことができる単純 HSAM (SHSAM) データベースを定義することができます。この場合、セグメント・タイプのオカレンスに接頭部を組み込みません。

SHSAM データベースの DBD を生成するときは、以下のものを指定します。

- 1 つのデータ・セット・グループ。
- アプリケーションがデータベースからデータを検索するときに使用する入力データ・セットの DD 名。
- データベースをロードするときに使用する出力データ・セットの DD 名。
- 単一のセグメント・タイプ内の 0 から 255 個のフィールド。

## GSAM の DBD 生成

GSAM データベースを DBD 生成するときは、以下のものを指定します。

- 1 つのデータ・セット・グループ。
- アプリケーションがデータベースからデータを検索するときに使用する入力データ・セットの DD 名。
- データベースをロードするときに使用する出力データ・セットの DD 名。
- オプションで、仮想セグメントを記述する SEGM ステートメント。
- オプションで、外部名のみを使用して定義されている FIELD ステートメント。これらのフィールドは IMS によって検索できないため、SSA に指定することはできません。これは通常、IMS に定義されていない COBOL コピーブックのフィールド・メタデータを格納するために使用されます。

以下のものは指定できません。

- 物理セグメントを記述する SEGMENT ステートメント
- IMS 検索可能フィールドを記述する FIELD ステートメント
- セグメント間の論理関係または索引関係の使用。

可変長 GSAM/BSAM データベースの場合、IMS はアプリケーションによって渡される GSAM レコード内のレコード長値に、2 バイトを追加します。これは、レコードが入出力装置に書き込まれるときに、BSAM レコード記述子ワード (RDW) を形成する ZZ フィールドを格納するためです。

次の図は、4 つの GSAM レコード (IMS セグメント) が 1 つの 32,760 バイト・ブロックに正確に収まることを示しています。

```
//IDASD DD DUMMY
//ODASD DD UNIT=SYSDA,VOL=SER=000000,DISP=(,KEEP),
//      SPACE=(TRK,(5,1)),DSN=GSAM.VARIABLE1,
//      DCB=(RECFM=VB,BLKSIZE=32760,LRECL=32756)
//SYSIN DD *,DCB=BLKSIZE=80
S 1 1 1 1 1 DBDNAME
L      ISRT
L V8187 DATA 1ST RECORD LOADED TO GSAM
L      ISRT
L V8187 DATA 2ND RECORD LOADED TO GSAM
L      ISRT
L V8187 DATA 3RD RECORD LOADED TO GSAM
L      ISRT
L V8187 DATA 4TH RECORD LOADED TO GSAM
```

## HISAM/SHISAM の DBD 生成

HISAM または SHISAM データベースを DBD 生成するときは、以下のものを指定します。

- 1 つのデータ・セット・グループ。
- 1 つの VSAM キー順データ・セット (KSDS) と 1 つの VSAM 入力順データ・セット (ESDS) の DD 名。HISAM は、1 つのデータ・セット・グループしかサポートしません。したがって、HISAM データベースでは 2 次データ・セット・グループを持つことはできません。
- オプションで、1 つの固定長セグメント・タイプのみを含むことができる単純 HISAM (SHISAM) データベースを定義することができます。この場合、セグメント・タイプのおカレンスに接頭部を組み込みません。SHISAM データベースに指定する論理レコード長は、指定されたセグメント長と同じにするか、それより長くする必要があります。
- 少なくとも 1 つのセグメント・タイプ、データベースでは最大 255 個のセグメント・タイプ。
- セグメント・タイプごとに 0 から 255 個、データベースでは最大 1000 個のフィールド。そのうちの 1 つは、ルート・セグメント・オカレンスを索引付けするためにルート・セグメント・タイプの固有のシーケンス・フィールドでなければなりません。
- セグメント・タイプにつき最大 32 個、データベースでは最大 1000 個の副次索引関係 (オプション)。
- HISAM データベース内のセグメントが HISAM データベース内の別のセグメントを指すときには、シンボリック・ポインター・オプションを使用する論理関係 (オプション)、および HISAM データベース内のセグメントが HDAM または

HIDAM データベース内のセグメントを指すときには、直接またはシンボリック・ポインター・オプションを使用する論理関係 (オプション)。

- セグメント編集/圧縮の出口ルーチン (オプション)。これを使用すると、ユーザー提供ルーチンでは、補助記憶装置との間での出入りの際にセグメント・タイプの各オカレンスを操作できます。
- データ・キャプチャー出口ルーチン (オプション)。これを使用すると、Db2<sup>®</sup> for z/OS ユーザーは、更新後の IMS データにアクセスできます。この出口ルーチンは、SHISAM でも使用できます。

制約事項: HISAM データベース内のセグメント間の階層ポインターまたは物理子または物理兄弟ポインターの使用は、指定することはできません。

## HDAM/PHDAM の DBD 生成

HDAM データベースおよび PHDAM データベースの DBD を生成するときは、以下のものを指定します。

- ルート・セグメント・オカレンスを配置するために使用するユーザー提供のランダム化モジュールの名前。
- 1 から 10 個のデータ・セット・グループ。
- 各データ・セット・グループ内のフリー・スペースの分布。
- 定義するデータ・セット・グループごとの OSAM または ESDS データ・セットの DD 名。
- データ・セット・グループごとに少なくとも 1 つのセグメント・タイプ、データベースでは最大 255 個のセグメント・タイプ。
- セグメント編集/圧縮出口ルーチン (オプション)。これを使用すると、ユーザー提供ルーチンは、補助記憶装置との間での出入りの際にセグメント・タイプの各オカレンスを操作できます。
- データベース内のセグメント間での階層ポインターまたは物理子または物理兄弟ポインターの使用。
- 直接アドレスかシンボリック・ポインター・オプション、またはそのどちらかを使用するセグメント間の論理関係 (オプション)。
- セグメント・タイプごとに 0 から 255 個、データベースでは最大 1000 個のフィールド。
- セグメント・タイプにつき最大 32 個、データベースでは最大 1000 個の副次索引関係 (オプション)。
- データ・キャプチャー出口ルーチン (オプション)。これを使用すると、Db2 for z/OS ユーザーは、更新後の IMS データにアクセスできます。

## PHDAM の DBDGEN

- DD 名およびデータ・セットは、PHDAM データベースの DBDGEN に含まれません。その他のデータベース定義は、単にデータの階層構造と関係を定義するためのものです。
- DBDGEN は、個々の区画の定義は行いません。

## HIDAM および PHIDAM の DBD 生成

HIDAM データベースおよび PHIDAM データベースの DBD を生成するときは、以下のものを指定します。

- 1 から 10 個のデータ・セット・グループ。
- 各データ・セット・グループ内のフリー・スペースの分布。
- 定義するデータ・セット・グループごとの OSAM または ESDS データ・セットの DD 名 (HDAM データベースのみ)。
- データ・セット・グループごとに少なくとも 1 つのセグメント・タイプ、データベースでは最大 255 個のセグメント・タイプ。
- セグメント編集/圧縮出口ルーチン (オプション)。これを使用すると、ユーザー提供ルーチンは、補助記憶装置との間での出入りの際にセグメント・タイプの各オカレンスを操作できます。
- セグメント・タイプにつき最大 32 個、データベースでは最大 1000 個の副次索引関係 (オプション)。
- データベース内のセグメント間での階層ポインターまたは物理子または物理兄弟ポインターの使用。
- 直接アドレスかシンボリック・ポインター・オプション、またはそのどちらかを使用するセグメント間の論理関係 (オプション)。
- セグメント・タイプごとに 0 から 255 個、データベースでは最大 1000 個のフィールド。そのうちの 1 つは、ルート・セグメント・オカレンスを索引付けするためにルート・セグメント・タイプの固有のシーケンス・フィールドでなければなりません。
- データ・キャプチャー出口ルーチン (オプション)。これを使用すると、Db2 for z/OS ユーザーは、更新後の IMS データにアクセスできます。

### PHIDAM の DBDGEN:

- DD 名およびデータ・セットは、PHIDAM データベースの DBDGEN に含まれません。その他のデータベース定義は、単にデータの階層構造と関係を定義するためのものです。
- DBDGEN は、個々の区画の定義は行いません。

## MSDB の DBD 生成

MSDB の DBD を生成するときは、以下のものを指定する必要があります。

- 1 つのデータベース名。
- 1 つのデータ・セット・グループ。
- データベースの 1 つのセグメント・タイプ。
- データベース内の 0 から 255 個のフィールド。

以下のものは指定できません。

- セグメント間の論理関係または索引関係。
- 副次索引で使用するフィールド。
- 配列または構造として定義されたフィールド。



既存の MSDB の DBD が変更されると、データベースのセグメントが変更されていなくても、ヘッダー情報 (BHDR) が変更される場合があります。この結果、MSDBCPx データ・セットからのロードが試行されるため、メッセージ DFS2593I が出力されます。この場合には、MSDBCPn データ・セット内のヘッダーは、無効であるか、長さが誤っています。MSDB PROCLIB メンバーに ABND=y を指定している場合には、U1012 異常終了も生じます。DBD を修正後、ウォーム・スタートまたはコールド・スタートのどちらかで MSDBLOAD オプションを使用して、MSDBINIT データ・セットから MSDB をロードし、これらの問題を取り除いてください。

## DEDB の DBD 生成

DEDB の DBD を生成するときは、以下のものを指定する必要があります。

- 1 つのデータベース名。
- データベース内の 1 から 2048 個の区域。
- データベースの 1 から 127 個のセグメント・タイプ。
- セグメント・タイプごとに 0 から 255 個、データベースでは最大 1000 個のフィールド。そのうちの 1 つは、ルート・セグメント・タイプの固有のシーケンス・フィールドでなければなりません。
- 区域を記述するために使用する DD 名または区域名。
- データ・キャプチャー出口ルーチン (オプション)。これを使用すると、Db2 for z/OS ユーザーは、更新後の IMS データにアクセスできます。

親のそれぞれの子タイプに、最高 8 個のサブセット・ポインターを任意に指定することができます。

セグメント・タイプ間の論理関係または索引関係は指定できません。

## 索引、PSINDEX DBD、および FPINDEX DBD の生成

HIDAM 1 次索引の DBD 生成は、1 つの索引セグメント・タイプ (HIDAM ルート・セグメント・タイプのオカレンスを索引付けする) からなる索引データベースを作成します。PHIDAM には、1 次索引の DBD はありません。索引セグメントは次のものを含みます。

- 索引付けするルート・セグメント・オカレンスのシーケンス・フィールド・キー。
- その接頭部内に、そのルート・セグメント・オカレンスへの直接アドレス・ポインター。

HIDAM 1 次索引の DBD を生成するときは、以下のものを指定する必要があります。

- 1 つのデータベース名。
- 1 つのデータ・セット・グループ。1 つの KSDS の DD 名を指定する必要があります。
- 1 つのセグメント・タイプ。
- HIDAM 1 次索引データベースと HIDAM データベースのルート・セグメント・タイプとの間で必要な索引関係。

- セグメント・タイプ内にシーケンス・フィールドとして 1 つのフィールド。

制約事項:

- 副次索引の場合とは異なり、この他の FIELD ステートメントを指定することはできません。
- DBDGEN を使用して個々の区画を定義することはできません。
- 非固有副次索引 (PSINDEX) データベースは、HALDB に関してはサポートされていません。

副次索引の DBD 生成1 から 16 の索引ポインター・セグメント・タイプからなる副次索引データベースを作成します。これらは、HISAM、SHISAM、HDAM、PHDAM、HIDAM、または PHIDAM データベース内のターゲット・セグメントの索引付けに使用されます。

全機能副次索引の DBD を生成するときは、次のものを指定する必要があります。

- 1 つのデータベース名。
- 1 つのデータ・セット・グループ。すべての索引ポインター・セグメント・キーが固有である場合には、1 つの KSDS の DD 名を指定する必要があります。索引ポインター・セグメント・キーが固有でない場合には、1 つの KSDS および 1 つの ESDS の DD 名をそれぞれ指定する必要があります。副次索引は VSAM を使用しなければなりません。
- 1 つのセグメント・タイプ。
- 各セグメント・タイプごとに 1 つのフィールド

高速機能副次索引の場合、DBD ステートメントでは NAME= パラメーターに副次索引データベースの名前を指定します。ACCESS= パラメーターには、以下のいずれかの値を指定する必要があります。

**ACCESS=(INDEX,VSAM),FPINDEX=YES**

新しい高速機能副次索引データベースの DBD ステートメント上の HISAM 副次索引データベース。

**ACCESS=(INDEX,SHISAM),FPINDEX=YES**

新しい高速機能副次索引データベースの DBD ステートメント上の SHISAM 副次索引データベース。

1 次 DEDB データベースを副次索引によって定義するには、その 1 次 DEDB データベースの DBD 内の索引付けフィールドに LCHILD および XDFLD ステートメントを追加します。

HISAM 副次索引データベースまたは SHISAM 副次索引データベースで LCHILD ステートメントの NAME= パラメーターに 2 つ以上のユーザー区画データベースが定義されている場合、1 次 DEDB データベース内の XDFLD ステートメントの PSELRTN= パラメーターにユーザー区画選択出口を指定します。サンプルのユーザー区画選択出口は DBFPSE00 です。データの終わりに達したことを示す GB 状況コードが返される前に、単一のユーザー区画と複数のユーザー区画のどちらが使用されるようにするかを制御する PSELOPT=MULT|SNGL パラメーターを、XDFLD ステートメントまたは PCB ステートメントで PROCSEQD= パラメーターを使用して指定することができます。PSELOPT=MULT|SNGL は、PCB ステートメントで PROCSEQD= パラメーターを使用して明示的に指定する必要があります。PCB

ステートメントではデフォルト PSELOPT=MULT は適用されません。この値が XDFLD ステートメントの PSELOPT=MULT|SNGL をオーバーライドするためです。

## 論理 DBD の生成

論理 DBD 生成は、複数の論理セグメント・タイプからなる論理データベースを作成します。論理セグメント・タイプは、論理データベースに定義されたセグメント・タイプであり、物理データベース (1 つ以上) に定義されている 1 つのセグメント・タイプまたは 2 つのセグメント・タイプの連結を表します。

論理データベースの DBD を生成するときは、以下のものを指定する必要があります。

- 1 つのデータベース名。
- 1 つの論理データ・セット・グループ。
- 1 から 255 個のセグメント・タイプ。各セグメント・タイプは、論理セグメント・タイプの名前、および物理データベース (論理セグメント・タイプを処理するための呼び出しが出されたときに処理される) 内のセグメント・タイプ (1 つ以上) の名前を定義します。

論理データベースを作成するために使用する論理関係は、物理データベース (1 つ以上) で定義しておかなければなりません。

論理データベース内のセグメントに必要なすべてのフィールドは、物理データベース内に定義しておかなければなりません。

## DBD 生成の入力レコード構造 (DEDB の DBD を除く)

DBDGEN プログラムは制御ステートメントを受け入れます。制御ステートメントは、必ず特定の順序で SYSIN 入力ストリームに追加する必要があります。

以下の図は、DBD 生成入力 of 構造化のための規則を示したものです。

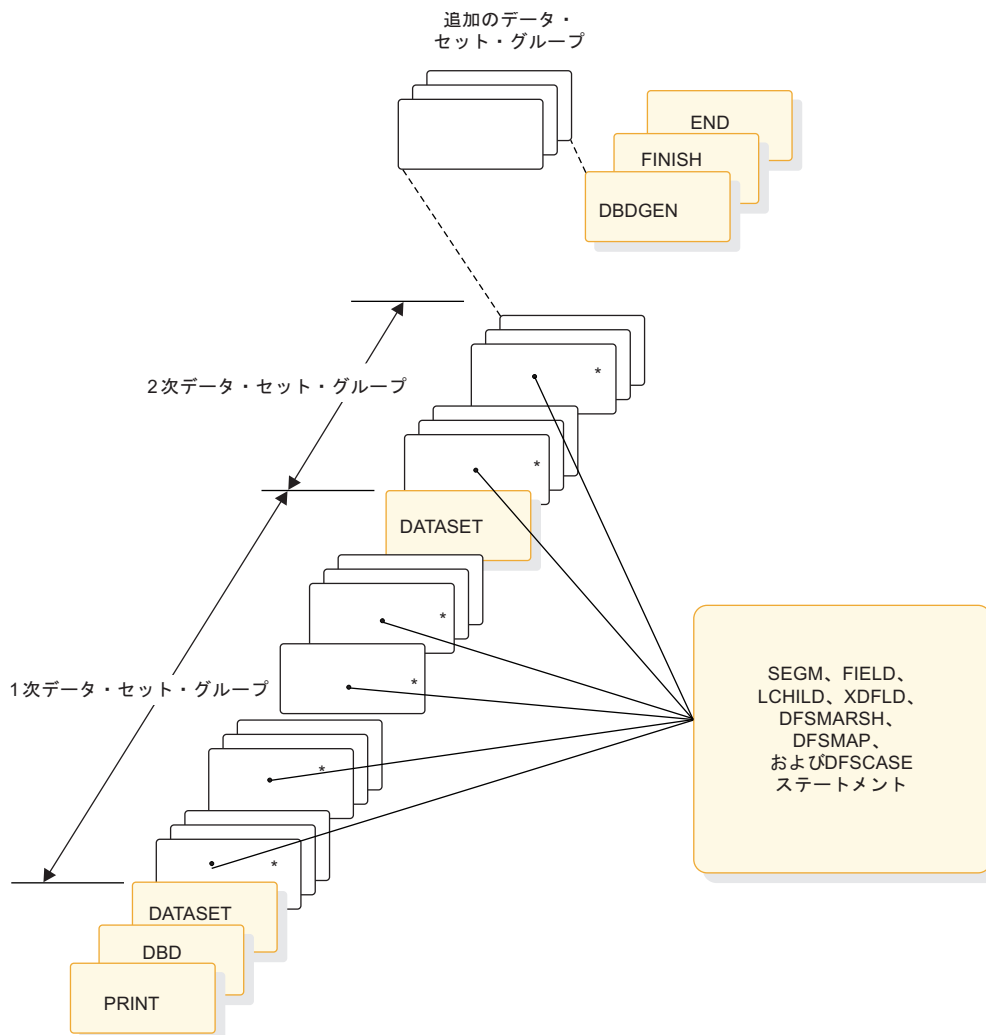


図 3. DBDGEN 入力レコード構造 (DEDB を除く)

例外: この入力レコード構造は、すべての DBD (DEDB の DBD は除く) に適用されます。

PRINT ステートメントはオプションです。このステートメントを含める場合には、入力デッキの最初のステートメントになります。PRINT を含めない場合には、DBD 制御ステートメントが入力デッキの最初になります。1 つ以上の DATASET ステートメントが DBD ステートメントの後に続きます。各 DATASET ステートメントの後には、そのデータ・セット・グループ内で定義されている可能性がある、SEGM、LCHILD、FIELD、XDFLD、DFSMARSH、DFSMAP、および DFSCASE ステートメントが続きます。少なくとも 1 つの SEGM ステートメントが各 DATASET ステートメントの後に続く必要があります。DBDGEN 入力レコード・セット内の SEGM ステートメントは、定義されるデータベース内のセグメントと同じ階層順に配置する必要があります。

FIELD ステートメントと LCHILD ステートメントは、適用する SEGM ステートメントの後に続きます。FIELD ステートメントがセグメント内のシーケンス・フィールドを定義している場合、このステートメントは、SEGM ステートメントの後に続いているすべての XDFLD ステートメントまたは他の FIELD ステートメント

よりも前になければなりません。LCHILD ステートメントは、論理親、HIDAM および PHIDAM ルート、索引ターゲットおよび索引ポインター・セグメント・タイプを定義している SEGM の後に続きます。副次索引関係を定義する場合は、その関係を確立する LCHILD ステートメントの後に対応する XDFLD ステートメントが続く必要があります。それら 2 つのステートメントの間に無関係な LCHILD ステートメントを入れることはできません。XDFLD ステートメントは、副次索引の索引ターゲット・セグメント・タイプを定義する SEGM の後に続きます。データベースごとに別々の入力レコード・セットが必要です。

フィールドに追加のメタデータを定義するために DFSMARSH ステートメントが使用されている場合、その DFSMARSH ステートメントは対応する FIELD ステートメントの後に続ける必要があります。IMSは、DFSMARSH ステートメントを、入力内でその DFSMARSH ステートメントの前に置かれている最後の FIELD ステートメントに関連付けます。

セグメント内で代替フィールド・マッピングを定義するために DFSMAP および DFSCASE ステートメントが使用されている場合、DFSMAP ステートメントの DEPENDSON パラメーターによって参照されている FIELD ステートメントは、入力内でその DFSMAP ステートメントの前に配置する必要があります。

要件: DBDGEN ステートメントは必須です。

FINISH を使用する場合には、END ステートメントの前に置きます。END は、入力レコード構造の最後のステートメントになります。

## DEDB の DBD 生成の入力レコード構造

DEDB の DBD 生成の入力レコード・セットの構造は、DATASET ステートメントの代わりに AREA ステートメントが使用されること以外は、他のタイプの DBD 生成と基本的には同じです。

AREA ステートメントはすべて、DBD ステートメントの直後にくる必要があります。SEGM ステートメントおよびその関連 FIELD ステートメントは、最後の AREA ステートメントの後に階層順に続きます。また、SEGM ステートメントは、定義されるデータベース内のセグメントと同じ階層順に配置する必要があります。

DEDB の DBD 生成では、次のことに注意してください。

- データ・セット・グループの概念は使えません。
- 副次索引が使えます。
- データベース間の論理関係は使えません。
- LCHILD ステートメントと XDFLD ステートメントが使えます。
- 順次従属セグメントには従属セグメントを付けられません。
- データベースごとに別々の入力レコード・セットが必要です。

以下の図に、DEDB の DBD を生成する入力レコード・セットを構成するときの規則を示します。

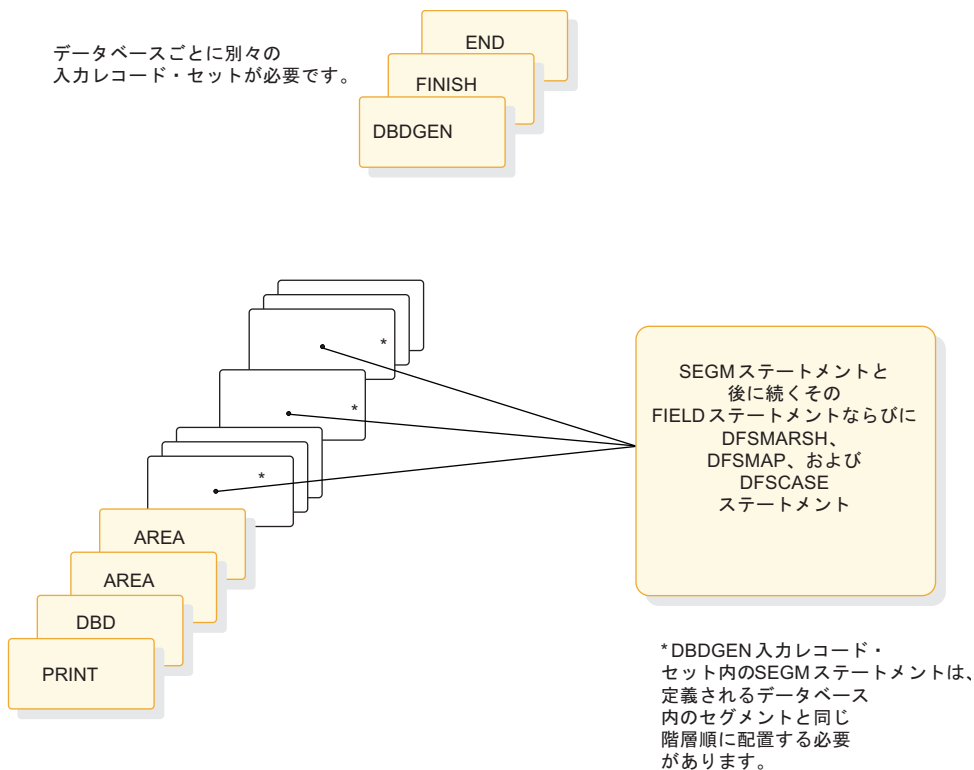


図 4. DEDB の DBDGEN 入力レコード構造

## DBD 生成のコーディング規則

DBD 生成ステートメントは、アセンブラー言語のマクロ命令です。各制御ステートメントは、命令コード (例えば、レコード・タイプ・コード) で区別しなければなりません。

関連概念:

➡ MVS マクロ命令の作成

## DBDGEN ステートメント

DBDGEN ユーティリティは、データベースを定義するための入力データとして、ステートメント命令タイプを使用します。

次のリストは、DBDGEN ユーティリティが受け入れるマクロ・ステートメントを示しています。

### PRINT

アセンブリー・リストがあれば、その印刷を制御するこのステートメントはオプションです。

**DBD** データベース名を定義する。このステートメントはすべてのデータベース・タイプで必須です。

全機能データベース名および DEDB データベース名はすべて固有でなければなりません。

**DATASET**

データベース内のデータ・セット・グループを定義する

**AREA**

高速機能 DEDB データベース内のエリアを定義する  
すべての DEDB エリアの名前は固有でなければなりません。

**SEGM**

データ・セット・グループ内または区域内のセグメント・タイプを定義する

**LCHILD**

セグメント・タイプ間の論理関係または索引関係を定義します。

**FIELD**

セグメント・タイプ内のフィールドを定義する

各 DBD 生成につき、FIELD と XDFLD のステートメントの合計は最大 1000 個。

**XDFLD**

副次索引で使用するフィールドを定義する

各 DBD 生成につき、FIELD と XDFLD のステートメントの合計は最大 1000 個。

**DFSMARSH**

フィールドのマーシャル属性を定義する。

**DFSMAP**

DFSMAP ステートメントは、セグメント内の代替フィールド・マッピングを定義するために、DFSCASE ステートメントを、セグメント・インスタンス内で有効な特定の DFSCASE ステートメントを識別するセグメント内の制御フィールドと関連付けます。このステートメントは、セグメントで代替フィールド・マッピングが使用されている場合にのみ必須です。

**DFSCASE**

代替フィールド・マッピングを使用するセグメント・タイプのマップ・ケースを定義します。このステートメントは、セグメントで代替フィールド・マッピングが使用されている場合にのみ必須です。

**DBDGEN**

DBDGEN ステートメントは、DBD の定義に使用される DBD 生成ステートメントの終了を示します。このステートメントは必須です。

**FINISH**

FINISH ステートメントは、オプションであり、互換性の目的で保持されています。

**END**

END ステートメントは、入力ステートメントの終わりをアセンブラーに伝えます。このステートメントは必須です。

以下の表に、データベースを定義する場合に DBDGEN ユーティリティへの入力として使用される、ステートメント命令のタイプが示してあります。さらに、ステートメントの各タイプの通常の使用法および各 DBD 生成で使用される各タイプの数も示しています。

各データベース・タイプに必要な DBDGEN ステートメント・セットは、それぞれ異なる場合があります。次の表で各ステートメントに示されている数は、各データベース・タイプでそのステートメントが必須であるか、オプションであるか、または適用外であるかを示しています。

表 1. DBD 生成ステートメント命令の要約：

マクロ	各 DBD 生成での使用個数											
	HSAM/ SHSAM	GSAM	HISAM/ HDAM	PHDAM	HIDAM	PHIDAM	MSDB	DEDB	索引	PSINDEX	論理	
PRINT	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1	0 から 1
DBD	1	1	1	1	1	1	1	1	1	1	1	1
DATASET	1	1	1/1 から 10	N/A	1 から 10	N/A	1	0	1	N/A	1	1
AREA	0	0	0	0	0	0	0	1-2048	0	0		
SEGM	1 から 255 <sup>3</sup>	0 から 1	1 から 255	1 から 255	1 から 255	1 から 255	1	1 から 127	1 <sup>1</sup>	1 <sup>1</sup>	1 から 255	
LCHILD	0	0	0 から 255	0 から 255	1 から 255	1 から 255	0	0	1 <sup>1</sup>	1 <sup>1</sup>	0	
FIELD	0 から 1000	0 から 1000	1 から 1000	0 から 1000	1 から 1000	1 から 1000	0 から 255	1 から 1000	1 <sup>2</sup>	1 <sup>2</sup>	0	
XDFLD	0	0	0 から 1000	0 から 1000	0 から 1000	0 から 1000	0	0 から 1000	0	0	0	
DFSMARSH	0 から 1000	0 から 1,000	0 から 1000	0 から 1000	0 から 1000	0 から 1000	0 から 255	0 から 1000	1 <sup>2</sup>	1 <sup>2</sup>	0	
DFSMAP	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし
DFSCASE	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし	0 以上。設定された制限なし
DBDGEN	1	1	1	1	1	1	1	1	1	1	1	1
FINISH	1	1	1	1	1	1	1	1	1	1	1	1
END	1	1	1	1	1	1	1	1	1	1	1	1

注:

1. 副次索引データベースごとに最大 16 個。
2. 副次索引データベースごとに最大 1000 個。
3. SHSAM データベースには SEGM ステートメントを 1 つしか指定できません。

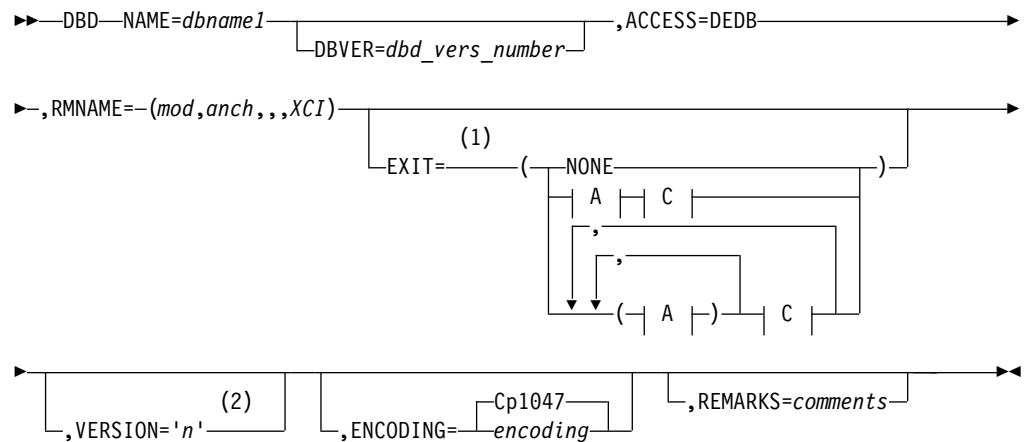
## DBD ステートメント

DBD ステートメントは、記述するデータベースの名前を指定し、データベース編成に関する情報を DL/I に与えます。制御ステートメント入力デッキには、1 つの DBD 制御ステートメントだけしか入れられません。

データベース・タイプごとの DBD マクロ命令の形式を、以下の例で示します。



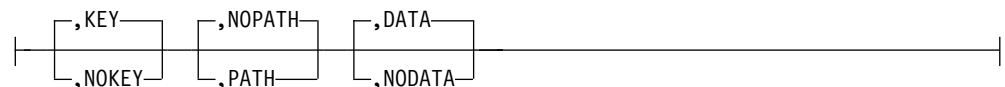
## DEDB データベースの DBD ステートメント



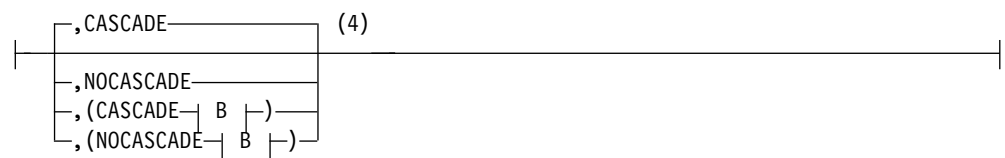
### A:



### B:



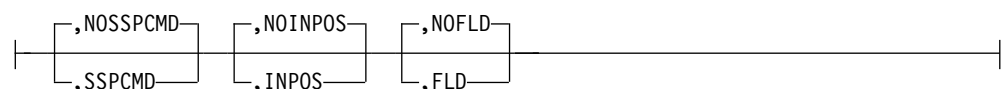
### C:



### D:



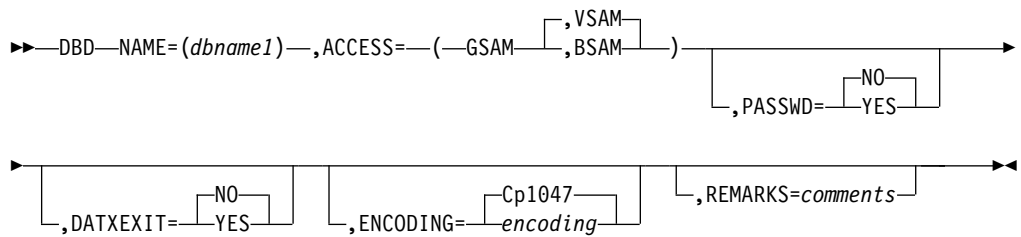
### E:



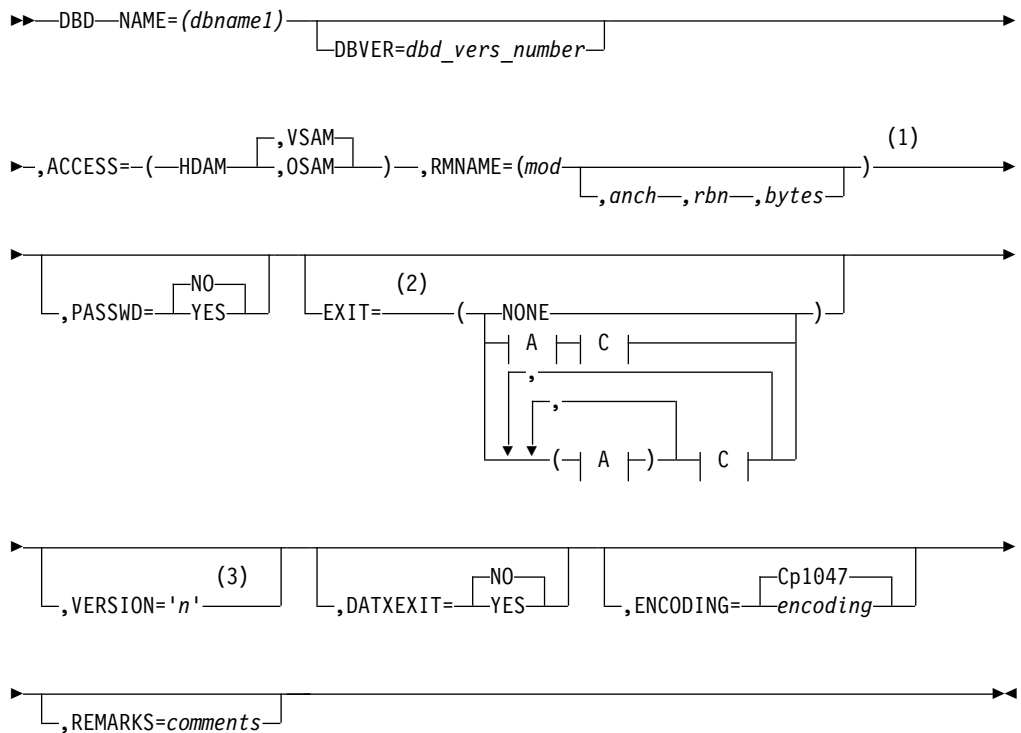
注:

- 1 データ・キャプチャー出口ルーチンに使用します。1 つの DBD ステートメントに複数の出口ルーチンを指定することができます。
- 2 デフォルトは、自動 DBDGEN タイム・スタンプです。
- 3 ロギングのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメータは LOG です。出口ルーチン名を指定すると、デフォルト・ロギング・パラメータは NOLOG になります。
- 4 CASCADE オプションの制御に使用します。

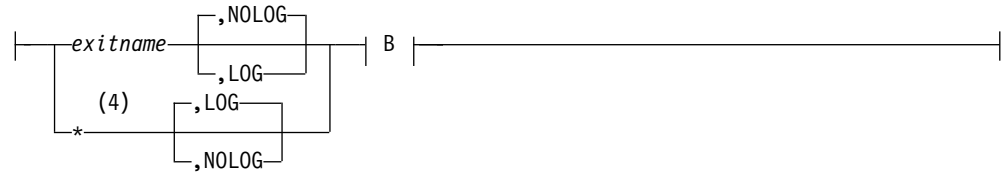
### GSAM データベースの DBD ステートメント



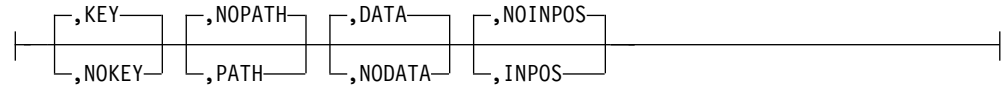
### HDAM データベースの DBD ステートメント



**A:**



**B:**



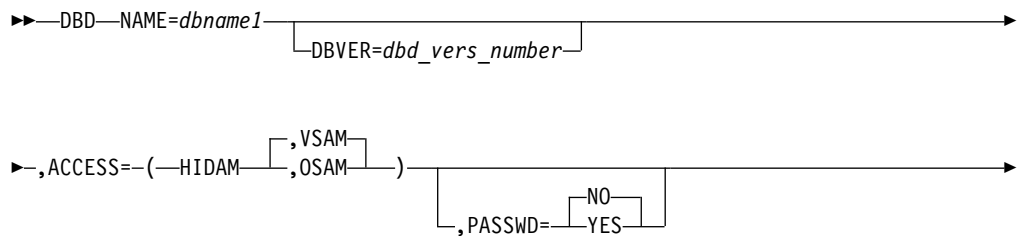
**C:**

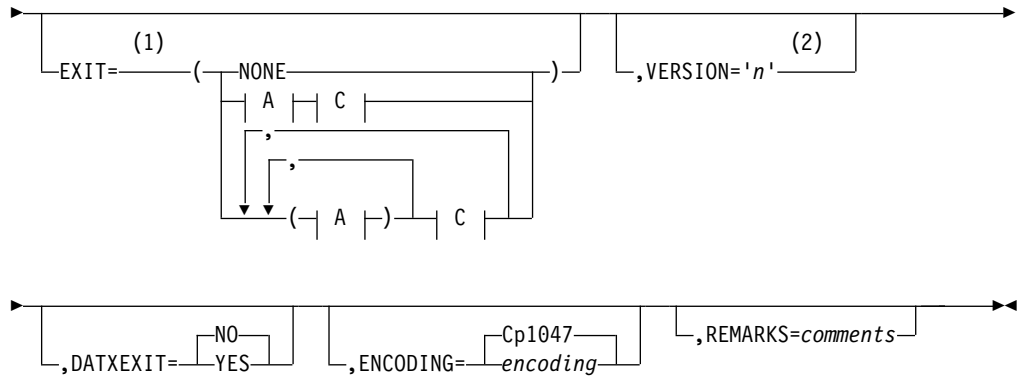


注:

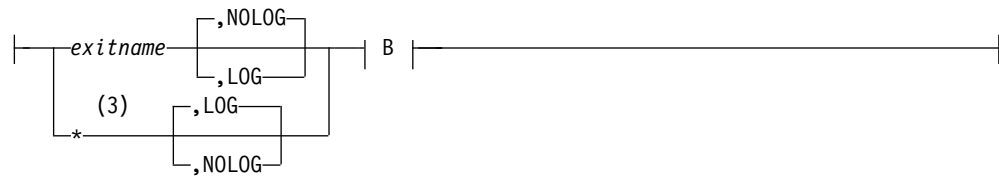
- 1 anch や rbn などのオプションのオペランドが、特定のランダム化モジュールで必要となることがあります。使用しているランダム化モジュールに関する文書を参照してください。
- 2 データ・キャプチャー出口ルーチンに使用します。1 つの DBD ステートメントに複数の出口ルーチンを指定することができます。
- 3 デフォルトは、自動 DBDGEN タイム・スタンプです。
- 4 ログのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメータは LOG です。出口ルーチン名を指定すると、デフォルト・ロギング・パラメータは NOLOG になります。
- 5 CASCADE オプションの制御に使用します。

### HIDAM データベースの DBD ステートメント

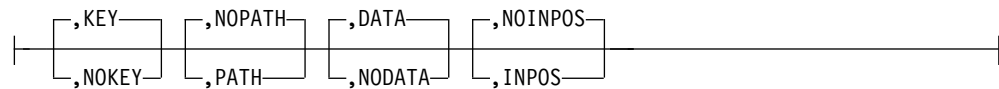




**A:**



**B:**



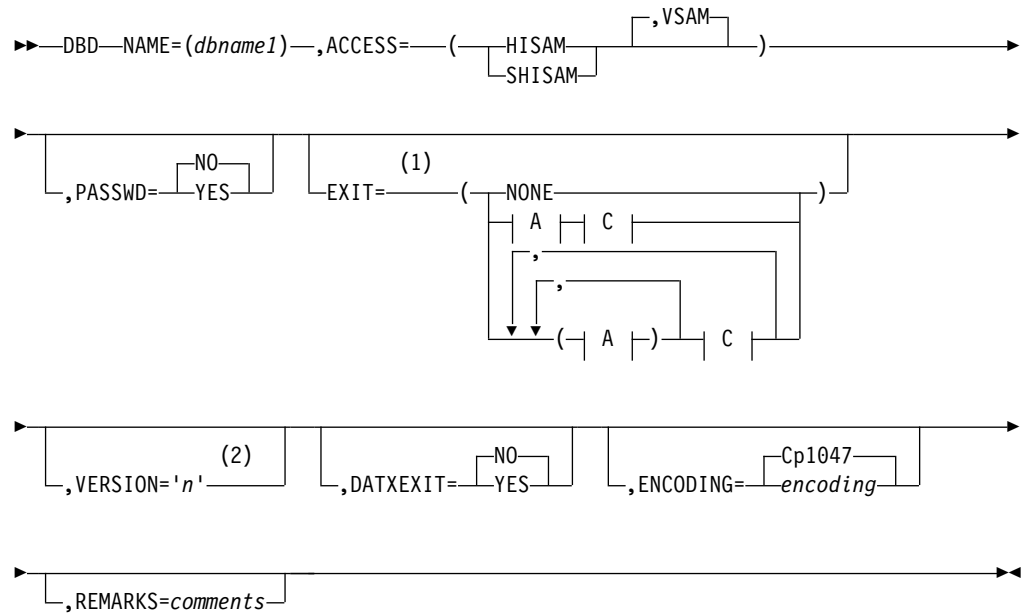
**C:**



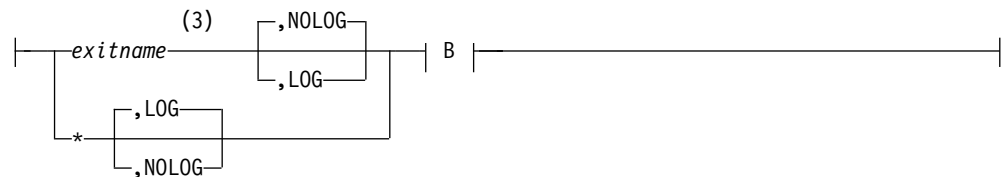
注:

- 1 データ・キャプチャー出力ルーチンに使用します。1 つの DBD ステートメントに複数の出力ルーチンを指定することができます。
- 2 デフォルトは、自動 DBDGEN タイム・スタンプです。
- 3 ログイングのみが要求されているため出力ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのログイング・パラメーターは LOG です。出力ルーチン名を指定すると、デフォルト・ログイング・パラメーターは NOLOG になります。
- 4 CASCADE オプションの制御に使用します。

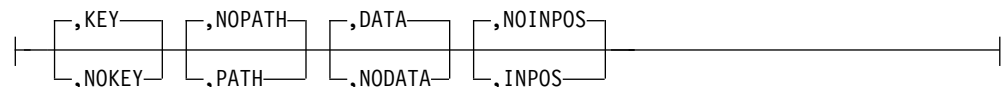
## HISAM/SHISAM データベースの DBD ステートメント



### A:



### B:



### C:



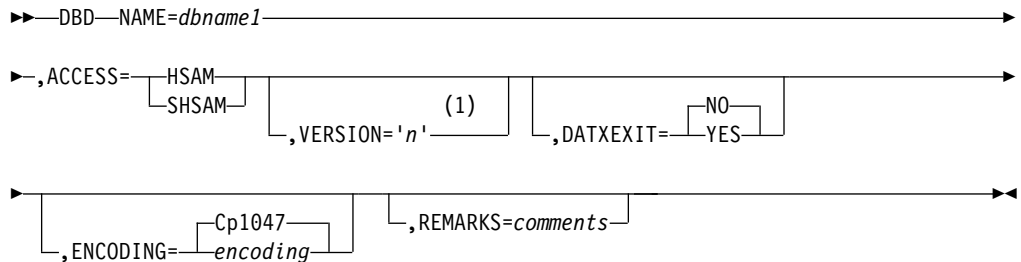
### 注:

- 1 データ・キャプチャー出口ルーチンに使用します。1 つの DBD ステートメントに複数の出口ルーチンを指定することができます。
- 2 デフォルトは、自動 DBDGEN タイム・スタンプです。
- 3 ロギングのみが要求されているため出口ルーチン名が不要な場合は、出口名に

アスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメータは LOG です。出力ルーチン名を指定すると、デフォルト・ロギング・パラメータは NOLOG になります。

4 CASCADE オプションの制御に使用します。

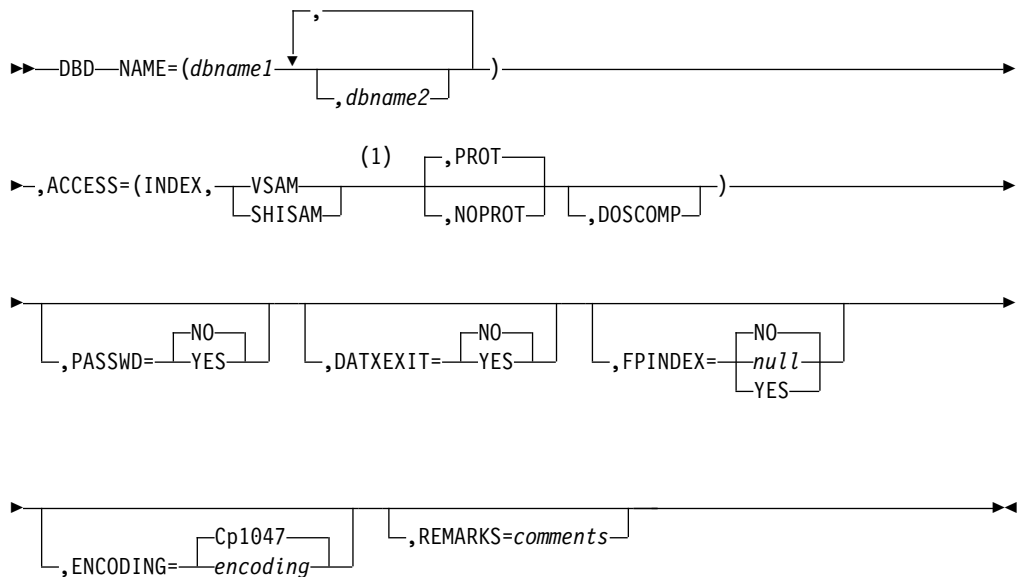
### HSAM/SHSAM データベースの DBD ステートメント



注:

1 デフォルトは、自動 DBDGEN タイム・スタンプです。

### INDEX データベースの DBD ステートメント

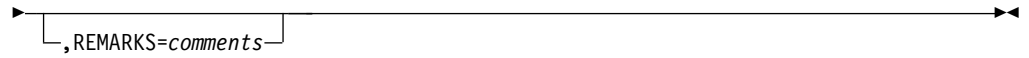


注:

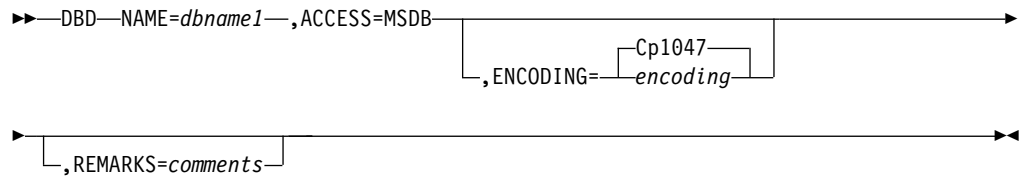
1 全機能副次索引では、VSAM を使用する必要があります。高速機能副次索引では VSAM と HISAM のいずれも使用できます。

### LOGICAL データベースの DBD ステートメント

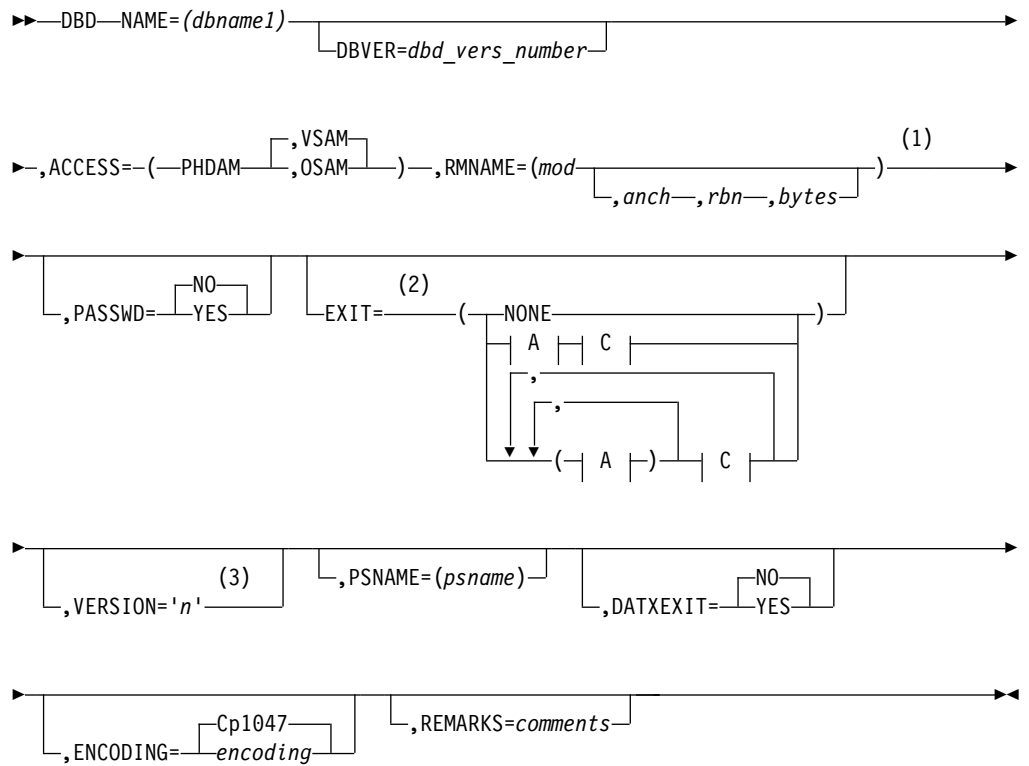




### MSDB データベースの DBD ステートメント



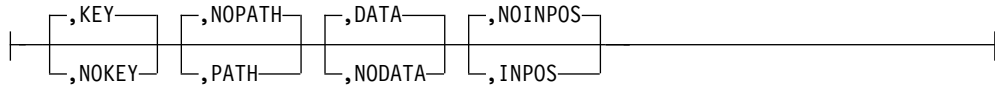
### PHDAM データベースの DBD ステートメント



**A:**



**B:**



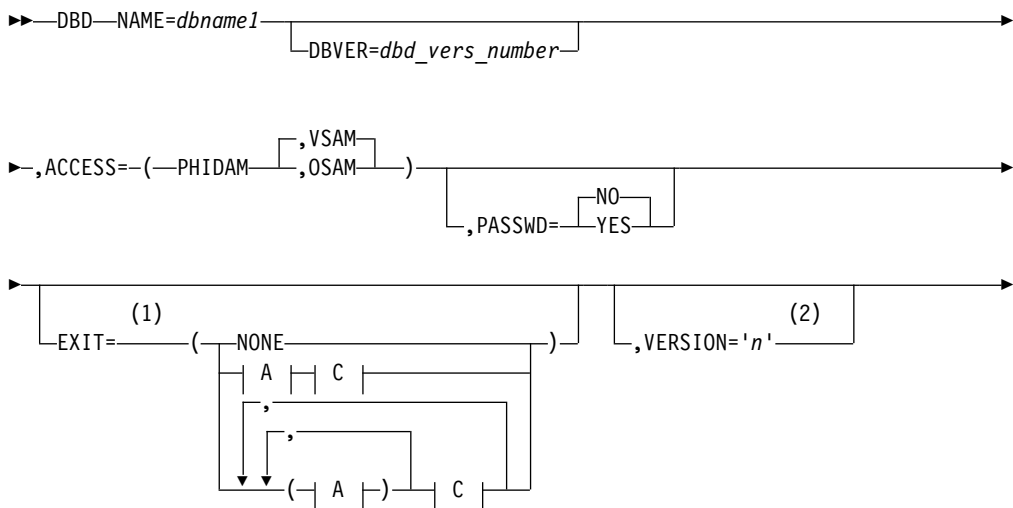
**C:**



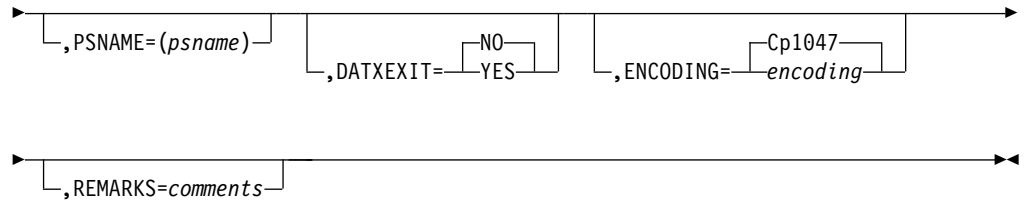
注:

- 1 `anch` や `rbn` などのオプションのオペランドが、特定のランダム化モジュールで必要となることがあります。使用しているランダム化モジュールに関する文書を参照してください。
- 2 データ・キャプチャー出口ルーチンに使用します。1 つの DBD ステートメントに複数の出口ルーチンを指定することができます。
- 3 デフォルトは、自動 DBDGEN タイム・スタンプです。
- 4 ログイングのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのログイング・パラメータは `LOG` です。出口ルーチン名を指定すると、デフォルト・ログイング・パラメータは `NOLOG` になります。
- 5 `CASCADE` オプションの制御に使用します。

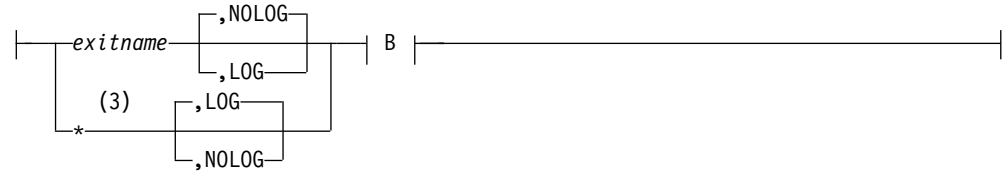
## PHIDAM データベースの DBD ステートメント



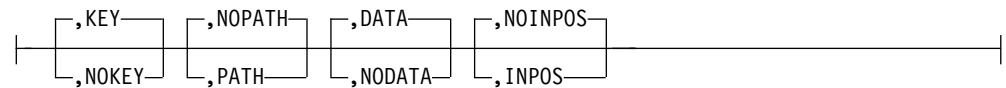




**A:**



**B:**



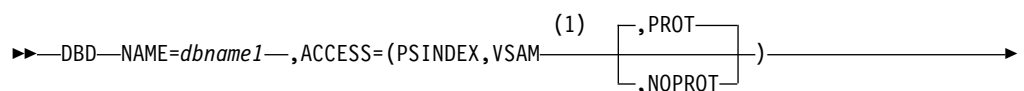
**C:**

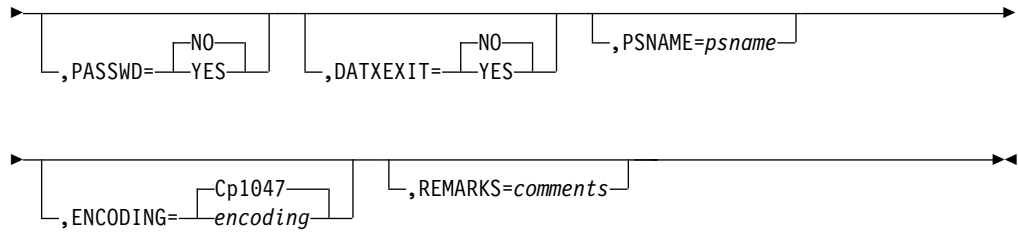


注:

- 1 データ・キャプチャー出口ルーチンに使用します。1 つの DBD ステートメントに複数の出口ルーチンを指定することができます。
- 2 デフォルトは、自動 DBDGEN タイム・スタンプです。
- 3 ログのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメーターは LOG です。出口ルーチン名を指定すると、デフォルト・ロギング・パラメーターは NOLOG になります。
- 4 CASCADE オプションの制御に使用します。

### PSINDEX データベースの DBD ステートメント





注:

- 1 副次索引は VSAM を使用しなければなりません。

## DBD ステートメント・パラメーターの説明

### DBD

このステートメントが DBD 制御ステートメントであることを表します。

### NAME=

記述するデータベースの名前を指定します。名前は、1 から 8 文字の英数字とすることができます。既存の PSB またはプログラム・ビューと同じ名前をデータベースに付けないでください。既存の名前を使用すると、予測不能な結果が起こる可能性があります。

既存の名前を使用すると、ACB の生成時にエラーが発生します。

この名前は、最初の DATASET 制御ステートメントの DD1 パラメーターで指定されたものと同じにすることができます。

共用副次索引データベースの場合には、最高 16 個の副次索引 DBD の名前を指定できます。

### DBVER=

同じデータベースにアクセスするためにアプリケーション・プログラムが複数の DBD を使用している場合に、DBD の特定のバージョンを識別する 1 から 2147483647 までの数値。

DBVER キーワードに指定される値は、IMS カタログのデータベース・レコードに保管されている以前のバージョンの DBD における DBVER の最高値に 1 を足した値でなければなりません。

DBVER キーワードはオプションです。データベースのバージョン管理方式が使用不可である場合でも、DBVER キーワードを省略した場合の DBD のバージョンは 0 です。

DBVER キーワードは、以下のデータベース・タイプにのみ有効です。

- DEDB
- HDAM
- HIDAM
- PHDAM
- PHIDAM

### ACCESS=

このデータベースに使用する DL/I のアクセス方式およびオペレーティング・システムのアクセス方式を指定します。また、このキーワードは、副次索引データベースを HALDB として定義します。別のアクセス方式は次のとおりです。

#### **HSAM**

階層順次アクセス方式 (HSAM)。HSAM を指定しても、HSAM データベースに定義されているセグメント・タイプが 1 つだけしかない場合には、このパラメーターはデフォルトの SHSAM になります。

#### **SHSAM**

1 つの固定長セグメント・タイプのみを含む単純 HSAM データベース。単純 HSAM データベースが定義されている場合には、セグメント・タイプのおカレンスに接頭部を付けて IMS がデータベースを処理できるようにする必要はありません。

#### **GSAM**

汎用順次アクセス方式 (GSAM)。オペレーティング・システム・アクセス方式として BSAM または VSAM を指定できます。デフォルトは VSAM です。GSAM を指定すると、DBD で SEGM 制御ステートメントを使用することはできません。

#### **HISAM**

階層索引順次アクセス方式 (HISAM)。オペレーティング・システム・アクセス方式として VSAM を指定できます。これがデフォルトです。

#### **SHISAM**

1 つの固定長セグメント・タイプのみを含む単純 HISAM データベース。この単純 HISAM データベースは、オペレーティング・システム・アクセス方式に VSAM が指定されている場合にのみ指定できます。単純 HISAM データベースが定義されている場合には、セグメント・タイプのおカレンスに接頭部を付けて IMS がデータベースを処理できるようにする必要はありません。

#### **HDAM**

階層直接アクセス方式 (HDAM)。オペレーティング・システムのアクセス方式として、OSAM または VSAM を指定できます。デフォルトは VSAM です。

#### **PHDAM**

区分階層直接アクセス方式 (PHDAM)。オペレーティング・システムのアクセス方式として、OSAM または VSAM を指定できます。デフォルトは VSAM です。

#### **HIDAM**

階層索引直接アクセス方式 (HIDAM)。オペレーティング・システムのアクセス方式として、OSAM または VSAM を指定できます。デフォルトは VSAM です。

#### **PHIDAM**

区分化階層索引直接アクセス方式 (PHIDAM)。オペレーティング・システムのアクセス方式として、OSAM または VSAM を指定できます。デフォルトは VSAM です。

## **MSDB**

主記憶データベース (MSDB)。

## **DEDB**

高速処理データベース (DEDB)。

## **INDEX**

**HIDAM** データベース内のルート・セグメント・タイプのオカレンスの 1 次索引が作成されます。あるいは **HISAM**、**HDAM**、または **HIDAM** データベース内のセグメント・タイプの副次索引が作成されます。**HIDAM** データベースの 1 次索引または副次索引の場合は、オペレーティング・システム・アクセス方式として **VSAM** を指定する必要があります。

**INDEX** パラメーターは、**DEDB** データベースの副次索引を作成する場合にも使用されます。このような場合、**VSAM** および **SHISAM** はどちらも有効なオペレーティング・システム・アクセス・タイプです。

**PHIDAM** データベースの 1 次索引を定義する場合、**INDEX** パラメーターは使用しません。

## **PSINDEX**

**PHDAM** データベースおよび **PHIDAM** データベースのセグメント・タイプの区分副次索引を作成します。オペレーティング・システム・アクセス方式として **VSAM** を指定する必要があります。デフォルトは **VSAM** です。

## **LOGICAL**

論理データベースは、1 つ以上の物理データベースの一部または全部の論理連結で構成されます。論理データベースは、既存の物理データベースを参照する必要があります。

## **PROT | NOPROT**

副次索引データベースで索引ポインター保護を使用するかどうかを指定します。このオプション・パラメーターは、**IMS** で使用される索引ポインター・セグメント内のすべてのフィールドの保全性を確保します。このパラメーターを使用すると、索引ポインター・セグメント内のフィールド (索引ポインター・セグメントのユーザー・データ部分のフィールドは除く) での置換操作をアプリケーション・プログラムは行えなくなります。索引ポインター・セグメントの削除操作は引き続き使用可能です。索引ポインター・セグメントに対する削除が出されると、索引ポインター・セグメント内の索引ターゲット・セグメント・ポインターが削除されます。しかし、最初に索引ポインター・セグメントの作成の原因となった索引ソース・セグメントは削除されません。

索引ポインター保護を使用しない場合、アプリケーション・プログラムでは、索引ポインター・セグメント内のすべてのフィールド (定数フィールド、検索フィールド、およびサブシーケンス・フィールドは除く) を置換できます。どの条件下でも、索引データベースへの挿入は無効です。

デフォルトで、副次索引データベースでは索引ポインター保護を使用します。

## **DOSCOMP**

これが **DLI/DOS** 索引データベースであるかどうかを指定します。データベースが索引であって、**DLI/DOS** を使用して作成されたものである場合には、これを指定する必要があります。**DLI/DOS** 索引データベースは、接頭部の一部としてセグメント・コードを含みます。データベースが **DLI/DOS** 索引データベ

ースであることを指定すると、IMS は定義されたデータベース内にこのコードが存在することを見込んで、このコードを保存するような方法で処理を行います。これには、挿入される新しいセグメントにセグメント・コードを指定することが含まれます。DLI/DOS データベースは VSAM を使用する必要があり、PHDAM、PHIDAM、または PSINDEX データベースにすることはできません。

#### **PSNAME=**

PSINDEX、PHDAM、または PHIDAM データベースの HALDB 区画を選択するモジュールを指定します。このパラメーターは、HALDB 区画選択出口ルーチンのモジュール名です。このパラメーターは、データベースのアクセス・タイプが PSINDEX、PHDAM、または PHIDAM である場合のみ有効です。

例外: ルート・キー範囲で HALDB 区画メンバーシップを定義している場合は、ユーザー提供の HALDB 区画選択ルーチンは必要ありません。

#### **RMNAME=**

DEDB に格納されているデータ、あるいは HDAM または PHDAM データベースの 1 次データ・セット・グループに格納されているデータの管理に使用するモジュール名を指定します。このパラメーターが有効なのは、データベース・アクセス・タイプが HDAM、PHDAM、または DEDB である場合のみです。ランダム化モジュールは、DEDB、HDAM、または PHDAM データベースへのルート・セグメントの配置、またはそれらからのルート・セグメントの取り出しを制御します。ランダム化モジュールと呼ばれる 1 つ以上のモジュールは、IMS システム内で利用できます。ある特定のデータベースは、それに関連したランダム化モジュールを 1 つしか持つことができません。汎用モジュール (ユーザー提供のパラメーターを使用して、ある特定のデータベースのランダム化を実行するもの) を作成して、いくつかのデータベースで利用することができます。ランダム化モジュールの目的は、DEDB、HDAM、または PHDAM データベースへのルート・セグメントの配置、またはそれらからのルート・セグメントの取り出しのためにアプリケーション・プログラムが指定する値を、相対ブロック番号およびアンカー・ポイント番号に変換することです。2 ステージ・ランダムマイザーを選択すると、1 つの区域内でランダム化を実行することができます。2 ステージ・ランダムマイザーを選択すると、/DBRECOVERY コマンドで DEDB の区域をすべて停止させなくても、1 つの区域内のルート・アンカー・ポイント数を変更することができます。

PHDAM データベースでは、ランダムマイザーのモジュール名および値が、区画ごとのデフォルトになります。HALDB 区画定義時に、区画ごとに異なるランダムマイザー名と値を設定することができます。HALDB 区画選択は、ランダム化モジュールを呼び出す前に行われます。ランダム化モジュールは、1 つの区画内でのみ位置を選択します。

#### **mod**

モジュール名は、この DEDB、PHDAM、または HDAM データベース内のセグメントの格納およびアクセスに使用するユーザー提供ランダム化モジュールの名前 (1 文字から 8 文字の英数字) です。モジュール名パラメーターにランダムマイザーの名前を指定し、アンカー・ポイント・パラメーターに 2 を指定して、2 ステージ・ランダムマイザーを選択します。

### **anch**

anch 値の目的は、高速機能 DEDB データベースを定義するか、全機能 HDAM または PHDAM データベースを定義するかによって異なります。

このパラメーターは符号なしの 10 進整数でなければなりません。このパラメーターのデフォルト値は 1 です。

DEDB データベースの場合、anch の値はランダムマイザーのタイプを指定します。値 1 は、単一ステージ・ランダムマイザーを示します。値 2 は、2 ステージ・ランダムマイザーを示します。その他の値は無効です。

HDAM および PHDAM データベースの場合、anch の値は、HDAM または PHDAM データベースのルート・アドレス可能域内の各制御インターバルまたは制御ブロックに必要なルート・アンカー・ポイントの数を指定します。標準的な値は 1 から 5 であり、この値は 255 を超えることはできません。

HDAM または PHDAM データベースにアクセスするときに、ユーザー・ランダム化ルーチンが、このパラメーターに指定された数値より大きいアンカー・ポイント番号を生成する場合、制御インターバルまたは制御ブロック内で最高の番号のアンカー・ポイントが使用されます。ランダム化ルーチンが IMS アンカー・ポイント番号 0 を生成した場合には、IMS は制御インターバルまたは制御ブロックのアンカー・ポイント 1 を使用します。

### **rbn**

このデータベースに関してランダム化モジュールに作成させる相対ブロック番号の最大値を指定します。このパラメーターは、HDAM または PHDAM データベースの場合のみ使用します。この値により、HDAM または PHDAM データベースのルート・アドレス可能域内の制御インターバルまたは制御ブロックの数が決まります。このパラメーターは、 $2^{24}-1$  を超えない符号なしの 10 進整数にする必要があります。このパラメーターを省略すると、ランダム化モジュールが作成する相対ブロック番号で上限の検査は行われません。このパラメーターを指定したにもかかわらず、指定されたランダム化モジュールがこのパラメーターよりも大きい相対ブロック番号を生成する場合、ルート・アドレス可能域内の最高位の制御インターバルまたは制御ブロックが IMS により使用されます。ユーザーのランダム化モジュールがブロック番号 0 を生成する場合、制御インターバルまたは制御ブロック 1 が IMS により使用されます。

HDAM または PHDAM データ・セットでは、最初のビットマップは、データ・セットの先頭にあるエクステンツの先頭ブロックにあります。

HDAM または PHDAM データベースでは、データ・セット・グループごとに指定されるデータ・セットの先頭にあるエクステンツの最初の制御インターバルまたは制御ブロックが、ビットマップに使用されます。VSAM データ・セットでは、2 番目の制御インターバルがビットマップに使用され、最初の制御インターバルは予約されます。IMS は、ランダムマイザーが計算したそのブロックに 1 を加えます。

### **bytes**

別のデータベース・レコードの呼び出しによって分断されない一連の挿入でルート・アドレス可能域に格納できる、データベース・レコードの最大バイト数を指定します。このパラメーターは、HDAM または PHDAM データベースの場合のみ使用します。このパラメーターを省略すると、このデータ

ベースのルート・セグメント・アドレス可能域に挿入できるデータベース・レコードの最大バイト数は、無制限になります。bytes パラメーターは、 $2^{24}-1$  を超えない符号なしの 10 進整数にする必要があります。最大相対ブロック番号パラメーターを省略すると、このパラメーターは無視されます。この場合、ルート・アドレス可能域に挿入できるデータベース・レコードのバイト数に制限はありません。

このパラメーターを HDAM または PHDAM データベースに対して指定した場合に、データベース・レコードの長さがそれより大きいと、レコードの超過部分は、現行のファイルの終わり (EOF) に続くオーバーフロー域に挿入されます。この操作には、このパラメーターの値を超えるすべてのデータベース・レコードの超過部分を入れる十分なスペースが、現行の EOF の後に使用可能でなければなりません。現行の EOF の後のオーバーフロー域に十分なスペースがないと、データベース・レコードはデータベースにランダムに挿入されることとなります。

### **XCI**

この DEDB が、ランダムマイザーを呼び出すときに、拡張呼び出しインターフェースを使用するかどうかを指定します。このオプションでは、3 つの異なる方法でランダムマイザーを呼び出すことができます。IMS の初期設定時、または /START DB コマンド実行時に、IMS は、まずランダムマイザーをロードし、次にランダムマイザーに INIT 呼び出しを行って、その初期設定ルーチンを呼び出します。/DBR DB コマンド実行の過程で IMS は TERM 呼び出しを行い、終了ルーチンを呼び出してから、ランダムマイザーをアンロードします。アプリケーションがルート・セグメントに GU または ISRT 呼び出しを出すと、通常のランダム化呼び出しがランダムマイザーに対して行われます。XCI オプションは、DEDB の場合にのみ有効です。

### **PASSWD=**

IMS 以外のプログラムが偶発的に IMS データベースにアクセスするのを防ぎます。

### **YES**

PASSWDYES を指定すると、このデータベースのデータ・セットを開くときに、DL/I はデータベース名を VSAM パスワードとして使用します。このパラメーターは、VSAM をアクセス方式として使用するデータベースにのみ有効です。データベース名を LOGICAL または DEDB のデータベース・タイプのパスワードとして使用することはできません。ユーザーが、z/OS アクセス方式サービス・プログラムの DEFINE ステートメントを使用してこのデータベースの VSAM データ・セットを定義する場合には、制御レベル (CONTROLPW) またはマスター・レベル (MASTERPW) のパスワードは、この DBD の DBDNAME と同じでなければなりません。この DBD に関連付けられたすべてのデータ・セットでは、同じパスワードを使用する必要があります。

IMS DB/DC システムでは、すべての VSAM OPEN がパスワード検査をう回するため、オペレーター・パスワード・プロンプトは回避されます。IMS DB システムでは、VSAM パスワード検査が実行されます。バッチ環境では、自動パスワード保護が指定されていないときに、データベース名と

等しくないパスワードにより制御レベル (CONTROLPW) でデータ・セットがパスワード保護されている場合、オペレーター・パスワード・プロンプトが出されます。

**NO PASSWDNO** を指定すると、データベース名を VSAM パスワードとして使用してはならないことを示します。これがデフォルトの動作です。

#### **EXIT=**

データ・キャプチャー出口ルーチンを使用することを指定します。1 つの DBD ステートメントに複数の出口ルーチン名を指定できます。出口ルーチンごとに異なるデータ・オプションを選択できます。パラメーター内にリストされた出口ルーチンの順序により、セグメントについて呼び出される出口ルーチンの順序が決まります。

**EXIT=** パラメーターは、DBD ステートメントに指定された場合、SEGM ステートメントに **EXIT=** パラメーターがない物理データベース内のすべてのセグメントに適用されます。この出口ルーチンがサポートする物理データベースは、次のとおりです。

- HISAM
- SHISAM
- HDAM
- PHDAM
- HIDAM
- PHIDAM
- DEDB

サポートされるデータベース編成またはサポートされるセグメント・タイプに出口ルーチンを指定しなければ、DBDGEN は失敗します。

**EXIT=** パラメーターは、SEGM ステートメントに指定することもできます。

#### **exit\_name**

データを処理する出口ルーチンの名前を指定します。このパラメーターは必須です。この名前は、標準命名規則に従う必要があります。最大 8 文字の英数字を使用できます。出口ルーチン名の代わりにアスタリスク (\*) を指定して、ロギングのみを行いたいことを示すことができます。これを指定した場合には、ロギング・パラメーターのデフォルトは LOG です。出口ルーチンを指定した場合のロギング・パラメーター・デフォルトは、NOLOG です。次に示すオペランドはすべてオプションです。

#### **KEY | NOKEY**

出口ルーチンに物理連結キーを渡すかどうかを指定します。このキーは、アプリケーションによって更新される物理セグメントを識別します。

##### **KEY**

出口ルーチンに物理連結キーを渡すことを指定します。

**KEY** がデフォルトです。

##### **NOKEY**

出口ルーチンに物理連結キーが必要ないときに指定します。



## **DATA | NODATA**

物理セグメント・データが更新のために出口ルーチンに渡されるかどうかを指定します。

### **DATA**

物理セグメント・データが更新のために出口ルーチンに渡されることを指定します。DATA が指定されていて、セグメント編集/圧縮出口ルーチンも使用される場合には、渡されるデータは拡張データです。

DATA がデフォルトです。

### **NODATA**

出口ルーチンがセグメント・データを必要としない場合に指定します。NODATA は、物理セグメント・データを保管することにより生じるオーバーヘッドを回避する場合に使用します。

## **NOPATH | PATH**

出口ルーチンが、物理ルート of の階層パスにあるセグメントからのデータを必要とするかどうかを指定します。

### **NOPATH**

出口ルーチンが、物理ルート of の階層パスにあるセグメントからのデータを必要としないことを示します。NOPATH は、パス・データの検索に必要な処理時間をなくするための効率的な方法です。

NOPATH はデフォルトです。

### **PATH**

更新セグメントのために物理ルート of の階層パスにあるセグメントのデータを、出口ルーチンに渡す必要がある場合に、これを指定します。アプリケーションが、挿入、置換、または削除の目的で複数のセグメントを別々にアクセスできるようにするには、PATH を使用します。

Db2 for z/OS 1 次キーを構成するためにパスのセグメントからの情報が必要なときに、PATH オプションを使用できます。Db2 for z/OS 1 次キーは、従属セグメント更新の伝搬要求で使用されます。通常、この種のセグメント情報が必要なのは、親セグメントがキー情報を含んでおり、親セグメントに収納できない追加データを従属セグメントが含んでいる場合です。

PATH は、追加処理が必要な場合にも使用できます。例えば、D コマンド・コードを呼び出さなかった場合などに、1 つの呼び出しで複数のセグメントにアクセスしないということがあります。この場合、アプリケーションが別の呼び出しで各セグメントにアクセスする際、追加処理が必要になります。

## **DLET | NODLET**

DLET 呼び出しで X'99' ログ・レコードが書き込まれるかどうかを指定します。

### **DLET**

DLET 呼び出しで X'99' ログ・レコードが書き込まれます。

DLET はデフォルトです。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **NODLET**

DLET 呼び出しで X'99' ログ・レコードが書き込まれません。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **BEFORE | NOBEFORE**

REPL 呼び出しで、変更前データが X'99' ログ・レコードに組み込まれるかどうかを指定します。

#### **BEFORE**

REPL 呼び出しで、変更前データが X'99' ログ・レコードに組み込まれます。

BEFORE はデフォルトです。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **NOBEFORE**

REPL 呼び出しで、変更前データが X'99' ログ・レコードに組み込まれません。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **CASCADE | NOCASCADE**

DL/I がこのセグメントを削除するときに、出口ルーチンが呼び出されるかどうかを指定します。

#### **CASCADE**

アプリケーションが親セグメントを削除したため、DL/I がこのセグメントを削除するときに、出口ルーチンが呼び出されることを示します。CASCADE を使用すると、定義されたセグメントのためにデータが確実に取り込まれます。

CASCADE はデフォルトです。

CASCADE パラメーターには、3 つのサブオプションがあります。これらのサブオプションは、データが出口ルーチンに渡される方法を制御します。サブオプションを指定する場合には、CASCADE パラメーターとサブオプションを小括弧で囲む必要があります。

#### **KEY | NOKEY**

物理連結キーを出口に渡すかどうかを指定します。

#### **KEY**

物理連結キーを出口に渡します。このキーは、カスケード削除で削除されるセグメントを識別します。

KEY がデフォルトです。

#### **NOKEY**

削除するセグメントの物理連結キーを出口ルーチンが必要としないときに指定することができます。

### **DATA | NODATA**

カスケード削除の場合に、セグメント・データを出口ルーチンに渡すかどうかを指定します。

#### **DATA**

カスケード削除の場合に、セグメント・データを出口ルーチンに渡します。また、DATA は削除されるセグメントを識別します (物理連結キーで識別できない場合)。

DATA がデフォルトです。

#### **NODATA**

出口ルーチンがセグメント・データを必要としない場合に指定します。NODATA により、物理セグメント・データを保管した場合に生じるかなりのストレージ要件およびパフォーマンス要件が削減されます。

### **NOPATH | PATH**

出口ルーチンが、物理ルート of 階層パスにあるセグメント・データを必要とするかどうかを指定します。

#### **NOPATH**

出口ルーチンが物理ルート of 階層パスにあるセグメント・データを必要としないことを示します。NOPATH は、カスケード削除に必要な相当量のパス・データを除去するために使用します。

NOPATH はデフォルトです。

#### **PATH**

カスケード削除のために、アプリケーションが複数のセグメントを別々にアクセスするときに指定します。

### **NOCASCADE**

DL/I がこのセグメントを削除するときに、出口ルーチンは呼び出されないことを示します。従属セグメントを持たないセグメントを削除する場合には、カスケード削除は不要です。

### **LOG | NOLOG**

データ・キャプチャーの制御ブロックまたはデータを IMS システム・ログに書き込むかどうかを指定します。

#### **LOG**

データ・キャプチャーの制御ブロックとデータを IMS システム・ログに書き込むことを要求します。

#### **NOLOG**

データ・キャプチャーの制御ブロックとデータを IMS システム・ログに書き込まないことを示します。

### **NOSSPCMD | SSPCMD**

高速機能のサブセット・ポインター (SSP) に関連するコマンド・コードをキャプチャーするかどうかを示すオプション・パラメーター。デフォルトはNOSSPCMD です。このオプションは、サブセット・ポインターが使用されるセグメントでのみ指定することをお勧めします。

次の表は、それぞれの DL/I 呼び出しでキャプチャーされるコマンド・コードを示しています。

表 2. DL/I 呼び出しでキャプチャーされるコマンド・コード

DL/I 呼び出し	詳細
G* (Get 呼び出し)	M、S、W、Z、R は、M、S、W、Z のうち少なくとも 1 つが同じ SSA 上で指定されている場合にキャプチャーされ、PATH が要求されている場合は PATH データとともにキャプチャーされます。
REPL	M、S、W、Z
DLET	Z
ISRT	M、S、W、Z、R は、セグメントが挿入される場合、または挿入されないセグメントの SSA に R が指定されたが PATH データが要求されている場合にキャプチャーされます。

#### NOINPOS | INPOS

ISRT 呼び出しで次のツイン・データをキャプチャーすることを要求するオプション・パラメーター。デフォルトは NOINPOS です。挿入されたセグメントの後に続くツインのツイン・データは、INPOS が指定されていて、以下の条件が該当する場合にキャプチャーされます。

- 非固有セグメントの ISRT が作成されている。
- HERE の ISRT 規則が使用されている。

新規セグメントが唯一のツイン・インスタンスであるか、ツイン・チェーン内の最後のツイン・インスタンスである場合、ツイン・データはキャプチャーされません。

#### NOFLD | FLD

DEDB FLD 呼び出しによって行われる更新をキャプチャーすることを要求するオプション・パラメーター。このオプションは、DEDB にのみ有効です。キャプチャーされる情報は、LOG オプションが指定されている場合、X'9904' ログ・レコードに記録されるだけです。データ・キャプチャー出口ルーチンには渡されません。

#### VERSION=

DBD の識別に使用される文字ストリングを指定します。出口ルーチンにはこの文字ストリングが渡されるので、出口ルーチンはデータベースの更新に使用する DBD のバージョンを判別できます。

#### 文字ストリング

文字ストリングの長さは 255 バイトまでです。適切な値が挿入されたかどうかを確認するための検査は行われません。したがって、DBD の変更時には必ず可変長文字ストリングを更新することが重要です。

文字ストリングを指定しないと、13 文字のタイム・スタンプが DBDGEN により生成されます。これは、DBDGEN が完了した日時を表します。形式は次のとおりです。

MM/DD/YYHH.MM

ここで、  
**MM** 月  
**DD** 日  
**YY** 年の最後の 2 桁  
**HH** 24 時間クロックの時間  
**MM** 分

#### **DATXEXIT=**

ユーザー出口 DFSDBUX1 を、このデータベースの処理中にアプリケーションで使用できるようにします。デフォルトは NO です。

このデータベースを処理中に、アプリケーションがデータ変換ユーザー出口ルーチン (DFSDBUX1) を使用できるようにします。デフォルトは DATXEXITNO です。

DATXEXITYES を指定すると、各データベース呼び出しの始めと終わりにユーザー出口 DFSDBUX1 が呼び出されます。DFSDBUX1 がロードされない場合、IMODULE が呼び出されてこれをロードします。

DATXEXITNO を指定した場合でも、DFSDBUX1 が SDFSRESL にあれば、DFSDBUX1 ユーザー出口ルーチンを呼び出すことができます。データベース定義に対して DFSDBUX1 を再度呼び出す必要がない場合、X'FF' が JCB の SRCHFLAG フィールドに戻され、DFSDLA00 は、出口を必要としていないことを示すマークをデータベース定義に動的に付けます。この場合は、DMB が DMB プールから除去されない限り、IMS セッションの期間中に、そのデータベース定義に対してユーザー出口が再度呼び出されることはありません。

#### **FPINDEX=**

索引データベースが 1 次高速機能 DEDB データベースの副次索引であるかどうかを指定します。デフォルトでは、索引データベースは副次索引ではありません。

有効な値は NO、*null*、および YES です。NO と *null* は同等です。デフォルトは NO (または *null*) です。

#### **ENCODING=**

このデータベース内のすべての文字データのデフォルト・エンコードを指定する、1 文字から 25 文字のオプション・フィールド。

デフォルトのコード・ページは Cp1047 です。これは、EBCDIC エンコードを指定します。

この値に以下の文字を含むことはできません。

- 単一引用符または二重引用符
- ブランク
- より小 (<) およびより大 (>) 記号
- アンパーサンド (&)

この値は、個々のセグメントまたはフィールドでオーバーライドできます。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### **CORRECT**


```
REMARKS='These remarks apply to the XYZ application'
```

#### **INCORRECT**

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

関連情報:

 DBD770 (メッセージおよびコード)

## **DATASET ステートメント**

DATASET ステートメントは、データベース内のデータ・セット・グループを定義します。

1 つの DBD 生成に少なくとも 1 つの DATASET ステートメントが必要です。ただし、HALDB データベース、DEDB データベース、および LOGICAL データベースの場合を除きます。HALDB データベースでは、DATASET ステートメントの代わりに SEGM ステートメントの DSGROUP パラメーターを使用して、データ・セット・グループを定義します。DEDB データベースでは、DATASET ステートメントの代わりに AREA ステートメントを使用して、データ・セット・グループを定義します。

使用する DATASET ステートメントの最大数は、定義するデータベースのタイプによって異なります。データベースの中には、1 つのデータ・セット・グループしか持つことができないものもあります。高速処理データベースには、1 から 2048 個の区域を定義することができます。HDAM および HIDAM データベースは、1 から 10 個のデータ・セット・グループに分けることができ、それらは 59 ページの『データベースを複数のデータ・セット・グループに分割するための規則』の規則に従う必要があります。

DBDGEN 入力デックでは、そのデータ・セット・グループに入れられるすべてのセグメントの SEGM ステートメントの前に DATASET ステートメントがきます。DBD 生成の最初の DATASET ステートメントは、1 次データ・セット・グループを定義します。後続の DATASET ステートメントは、2 次データ・セット・グループを定義します。

例外: 優先順位に関する唯一の例外は、DATASET ステートメントのラベル・フィールドが使用されている場合です。この例外については、60 ページの『ラベル・フィールドの使用』を参照してください。

オペランドを持たない後続のラベル付き DATASET マクロには、コメントを追加してはなりません。

## データベースを複数のデータ・セット・グループに分割するための規則

HDAM および HIDAM データベースは、次の制限に従って、最大 10 個のデータ・セット・グループに分割できます。各 DATASET ステートメントにより、別個のデータ・セット・グループが作成されます。ただし、60 ページの『ラベル・フィールドの使用』で説明されているケースは除きます。最初の DATASET ステートメントは、1 次データ・セット・グループを定義します。後続の DATASET ステートメントは、2 次データ・セット・グループを定義します。

HDAM または HIDAM データベースでは、DATASET ステートメントを使用して、データベース階層の任意のレベルの複数データ・セット・グループに分割することができます。ただし、次の制限を満たす必要があります。物理親およびその物理子は、別々のデータ・セット・グループ内にある場合、次の図に示されているように、階層ポインターではなく、物理子または物理兄弟ポインターによって接続する必要があります。

セグメント A (1 次データ・セット・グループ内のルート・セグメント) とセグメント B (2 次データ・セット・グループ内の第 1 レベルの従属) は、物理子を使用して接続しなければなりません。セグメント C (1 次データ・セット・グループ内の第 1 レベルの従属) とセグメント D (2 次データ・セット・グループ内の 2 次レベルの従属) も、物理子を使用して接続しなければなりません。同じ親の下にあるセグメント B と D の複数のオカレンス間には、物理兄弟ポインターを使用して接続しなければなりません。

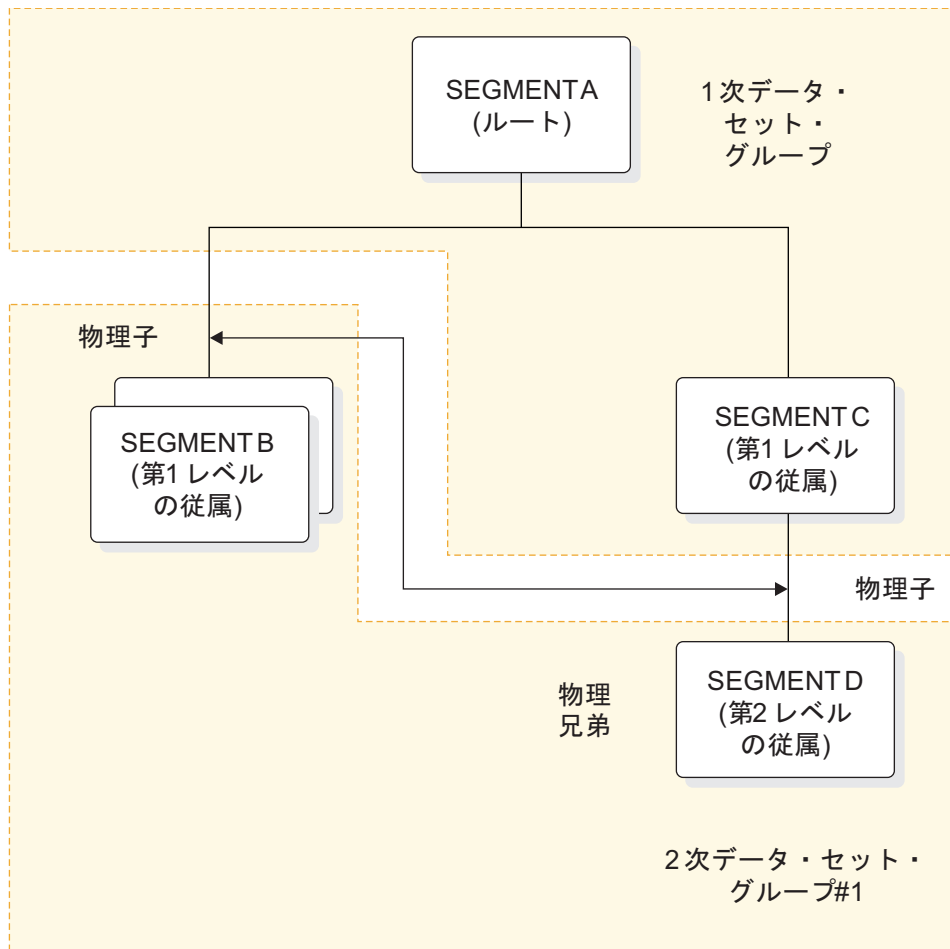


図 5. 物理子ポインターと物理兄弟ポインターによる接続

## ラベル・フィールドの使用

HDAM または HIDAM データベースでは、データ構造内の階層位置によってではなく、セグメント・サイズおよびアクセス頻度に応じて、セグメントをデータ・セット・グループに入れるほうが望ましい場合があります。セグメントを定義する SEGM ステートメントは階層順序で配置する必要があるという DBD 生成規則に従いながら、このことを達成するために、DATASET ステートメントのラベル・フィールドを使用します。

DATASET ステートメントにコーディングされた識別ラベルは、同じラベルを追加の DATASET ステートメントにコーディングすることで参照されます。共通ラベルを持つ最初の DATASET ステートメントのみが、データ・セット・グループの物理的特性を定義するオペランドを含むことができます。同じラベルを持つ DATASET ステートメントの後に続く SEGM ステートメントで定義されるすべてのセグメントは、そのラベルを持つ最初の DATASET ステートメントによって定義されるデータ・セット・グループに入れられます。

以下のような制限はありますが、アセンブラ言語の CSECT ステートメントとほぼ同じように、この機能を使用することができます。



- オペランドを指定した DATASET ステートメントのラベル・フィールドのラベルを、オペランドを指定する別の DATASET ステートメントで使用できません。
- ラベルは、英数字であって、アセンブラ言語ステートメントで有効なラベルでなければなりません。
- ラベルが付いていない DATASET ステートメントにはオペランドを指定しなければなりません。

60 ページの図 5 に関して次の表では、同じサイズのセグメント・タイプを同じデータ・セット・グループに入れるときに、DATASET ステートメントのラベル・フィールドをどのように使用するかを示しています。

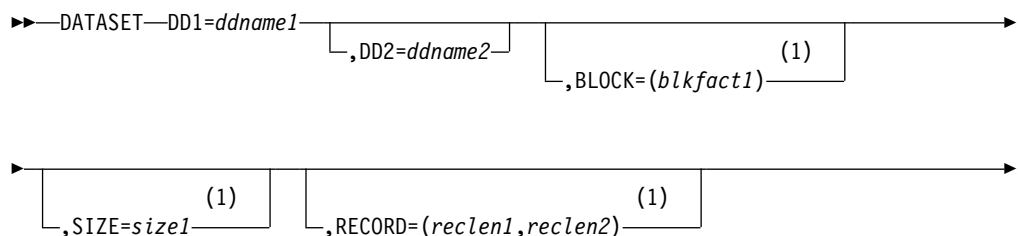
表 3. セグメント・タイプをグループ化するためのラベル・フィールドの使い方

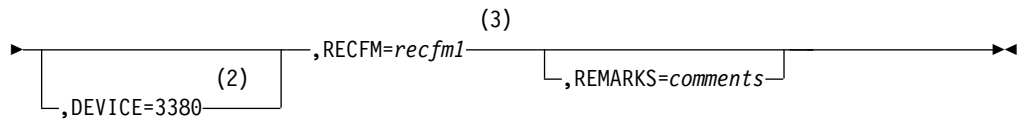
ラベル	操作	パラメーター
	DBD	NAME=HDBASE,ACCESS=HDAM, RMNAME=(RANDMODL,1,500,824)
DSG1	DATASET	DD1=PRIMARY,BLOCK=1648
	SEGM	NAME=SEGMENTA,BYTES=100
DSG2	DATASET	DD1=SECOND,BLOCK=3625
	SEGM	NAME=SEGMENTB,BYTES=50,PARENT=SEGMENTA
DSG1	DATASET	
	SEGM	NAME=SEGMENTC,BYTES=100,PARENT=SEGMENTA
DSG2	DATASET	
	SEGM	NAME=SEGMENTD,BYTES=50,PARENT=SEGMENTC
	DBDGEN	
	FINISH	
	END	

SEGMENTA および SEGMENTC という名前のセグメントは、1 次データ・セット・グループ中に存在します。SEGMENTB および SEGMENTD という名前のセグメントは、2 次データ・セット・グループ中に存在します。

各データベース・タイプの DATASET ステートメントの形式を、以下の例に示します。

### GSAM データベースの DATASET ステートメント

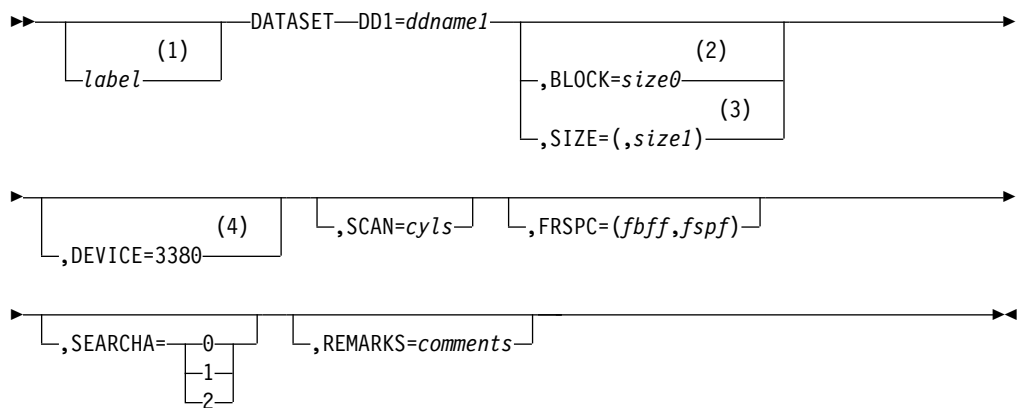




注:

- 1 値を指定しない場合、使用する値を DBDGEN ユーティリティが生成します。
- 2 DEVICE パラメーターは、無視されます。
- 3 RECFM は、GSAM データベースのみに有効です。

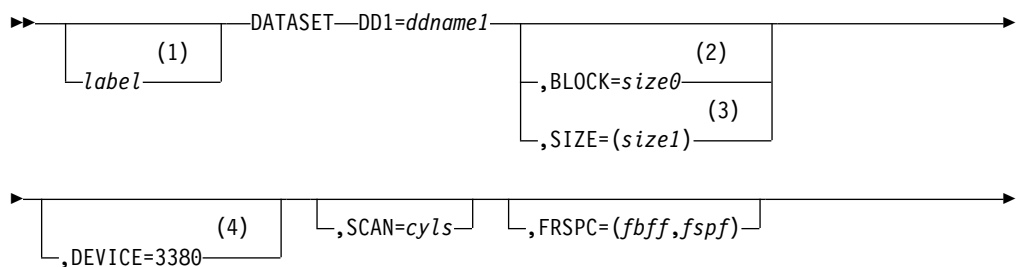
### HDAM データベースの DATASET ステートメント

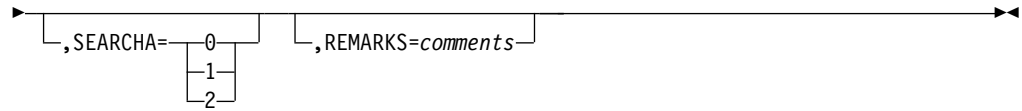


注:

- 1 同じ label が指定された DATASET ステートメントが複数ある場合、共通した label を持つ最初の DATASET ステートメントのみがデータ・セット・グループの物理的特性を定義するオペランドを含むことができます。
- 2 値を指定しない場合、使用する値を DBDGEN ユーティリティが生成します。
- 3 VSAM では、SIZE キーワードの有効なパラメーター指定は、512 バイト、1 KB、2 KB、4 KB、8 KB、および 28 KB までの 2 KB の倍数です。
- 4 DEVICE パラメーターは、無視されます。

### HIDAM データベースの DATASET ステートメント

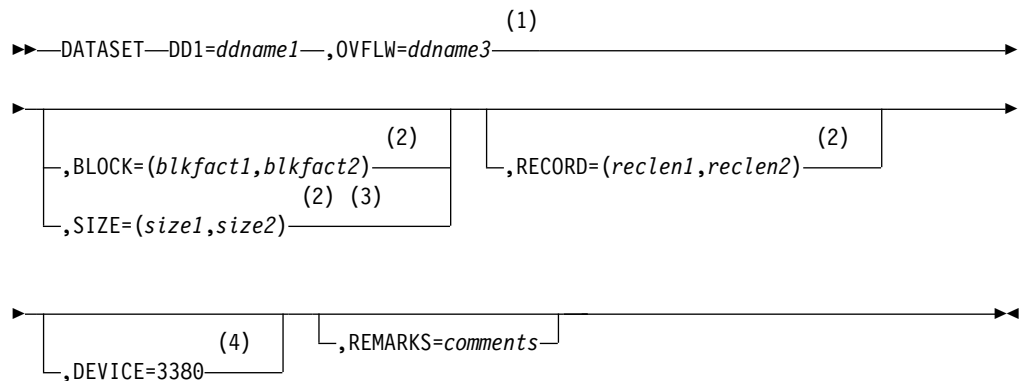




注:

- 1 同じ *label* が指定された DATASET ステートメントが複数ある場合、共通した *label* を持つ最初の DATASET ステートメントのみがデータ・セット・グループの物理的特性を定義するオペランドを含むことができます。
- 2 値を指定しない場合、使用する値を DBDGEN ユーティリティが生成します。
- 3 VSAM では、SIZE キーワードの有効なパラメーター指定は、512 バイト、1 KB、2 KB、4 KB、8 KB、および 28 KB までの 2 KB の倍数です。
- 4 DEVICE パラメーターは、無視されます。

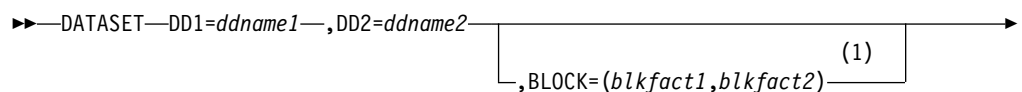
### HISAM/SHISAM データベースの DATASET ステートメント

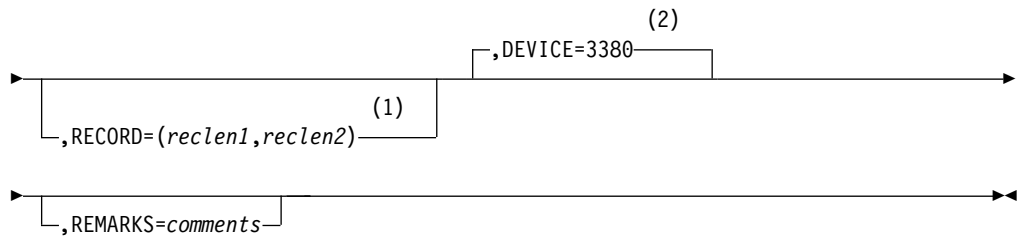


注:

- 1 HISAM データベースが、1 つのセグメント・タイプだけしか定義していない場合は、OVFLW を指定する必要がありません。単純 HISAM データベースでは、OVFLW は無効です。
- 2 値を指定しない場合、使用する値を DBDGEN ユーティリティが生成します。
- 3 SIZE キーワードの有効なパラメーター指定は、512 バイト、1 KB、2 KB、4 KB、8 KB、および 28 KB までの 2 KB の倍数です。
- 4 DEVICE パラメーターは、無視されます。

### HSAM/SHSAM データベースの DATASET ステートメント

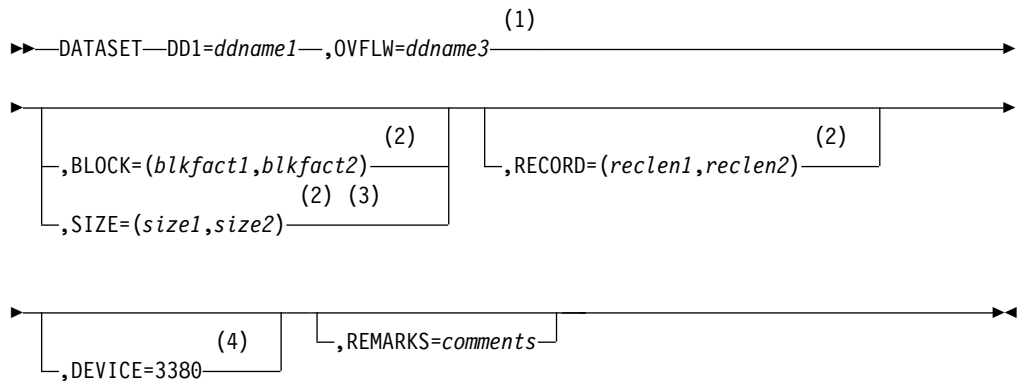




注:

- 1 値を指定しない場合、使用する値を DBDGEN ユーティリティが生成します。
- 2 DEVICE パラメーターは、無視されます。

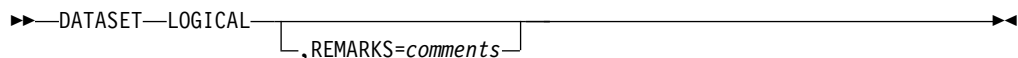
### INDEX データベースの DATASET ステートメント



注:

- 1 すべての索引セグメントのキーが固有であれば、OVFLW を指定する必要はありません。
- 2 値を指定しない場合、使用する値を DBDGEN ユーティリティが生成します。
- 3 SIZE キーワードの有効なパラメーター指定は、512 バイト、1 KB、2 KB、4 KB、8 KB、および 28 KB までの 2 KB の倍数です。
- 4 DEVICE パラメーターは、無視されます。

### LOGICAL データベースの DATASET ステートメント



### MSDB データベースの DATASET ステートメント



## LOGICAL

パラメーターは無効

HSAM、SHSAM または GSAM データベースの場合は、アプリケーション・プログラムがデータベースからデータを検索するときこの入力データ・セットを使用します。

### DEVICE=

この区域内のデータ・セットが格納される物理ストレージの装置タイプを指定します。デフォルトは 3380 です。これ以外の装置をコーディングすると、デフォルトは無視されます。

### DD2=

出力データ・セットの DD 名 (1 から 8 文字の英数字) を指定します。HSAM または SHSAM データベースには必須ですが、GSAM データベースにはオプションです。これを省略すると、ddname1 が想定されます。この出力データ・セットは、データベースのロード時に IMS が使用します。

### label

DATASET ステートメントにコーディングされた識別ラベルは、同じ *label* を追加の DATASET ステートメントにコーディングすることで参照されます。共通ラベルを持つ最初の DATASET ステートメントのみが、データ・セット・グループの物理的特性を定義するオペランドを含むことができます。同じラベルを持つ DATASET ステートメントの後に続く SEGM ステートメントで定義されるすべてのセグメントは、そのラベルを持つ最初の DATASET ステートメントによって定義されるデータ・セット・グループに入れられます。

### OVFLW=

このデータ・セット・グループ内のオーバーフロー・データ・セットの DD 名 (1 から 8 文字の英数字) を指定します。次のものについてはこのパラメーターを指定する必要があります。

- 固有でないキーが付いている索引ポインター・セグメントが入っている索引データベース。
- HISAM データベースのすべてのデータ・セット・グループ。ただし、HISAM データベース内に 1 つのセグメント・タイプしか定義されていない場合は除きます。

DD1、DD2、または OVFLW サブパラメーターで使用される DD 名は、IMS システムまたはアカウントの中で固有である必要があります。2 つまたはそれ以上の DBD の中に固有でない DD 名があると、データベースが破壊される可能性があります。データベースの破壊につながる可能性のある 1 つの状態は、両方の DD 名が同時に誤って使用される場合です (両方が、データ通信システムの 2 つの異なるメッセージ領域で使用されるか、あるいはデータベース専用システムのバッチ DL/I 領域で使用される 1 つの PSB の 2 つの PCB で使用される場合)。

次のような制約事項があります。

- 単純 HISAM データベースを定義するときは、OVFLW パラメーターを使用できません。
- 1 つのセグメント・タイプだけを入れる HISAM データベースを定義するときは、OVFLW パラメーターを指定する必要はありません。

- すべての索引セグメントが索引のキー順データ・セット内に挿入されるので、索引 DBD には DATASET ステートメントで OVFLW パラメーターを指定する必要はありません。
- ACCESS=(INDEX,SHISAM) を指定する場合、OVFLW パラメーターは無効です。

#### **BLOCK=**

ブロック化因数 (blkfact1、blkfact2) を指定して、HSAM、SHSAM、GSAM、HISAM、SHISAM、および索引データベースのデータ・セット・グループのデータ・セットで使用できるようにするか、またはオーバーヘッドを含まない (size0) 制御インターバル・サイズまたはブロック・サイズを指定して HDAM および HIDAM データベースのデータ・セット・グループのデータ・セットを使用できるようにします。

アクセス方式として VSAM を使用する HISAM、SHISAM、および索引データベースの場合は、BLOCK= パラメーターの代わりに SIZE= パラメーターを使用して制御インターバル・サイズを指定してください。HISAM、SHISAM、または索引データベースに SIZE= キーワードを指定した場合は、BLOCK= キーワードは無効です。

SIZE キーワードと BLOCK キーワードは、以下のデータベース・アクセス・タイプでは同時には使用できません。

- HDAM
- HIDAM
- HISAM
- INDEX
- SHISAM

RECORD= および BLOCK= オペランドを使用する場合は、結果として得られる制御インターバル・サイズは、8192 バイトより小さい場合は 512 の倍数でなければなりません。指定したレコード長と指定したブロック化因数の積に VSAM オーバーヘッドを加えたものが 512 の倍数でなく、8192 バイトより小さい場合には、制御インターバル・サイズは、それより大きな最も近い 512 の倍数の値に切り上げて得られます。8192 から 30720 バイト (最大許容サイズ) の制御インターバル・サイズは、2048 バイトの倍数でなければなりません。

RECORD= オペランドと BLOCK= オペランドの積に VSAM オーバーヘッドを加えたものが、8192 から 30720 バイトの値であっても、2048 の倍数ではない場合は、制御インターバル・サイズは、それより大きな最も近い 2048 の倍数の値に切り上げて得られます。

VSAM オーバーヘッドは、ブロック化因数が 1 の場合は 7 バイトであり、そうでない場合は 10 バイトです。OSAM データ・セットの最大ブロック・サイズは 32KB です。

HDAM および HIDAM データベースの場合は、BLOCK= パラメーターを使用すれば、制御インターバルまたはブロックのサイズについての DBDGEN の計算を変更することができます。ただし、BLOCK= パラメーターに指定された値のほかに、DBDGEN は、ルート・アンカー・ポイント、フリー・スペース・アンカー・ポイント、およびアクセス方式オーバーヘッドのためのスペースを追加します。結果として得られるブロックまたは制御インターバルのサイズは、SIZE= パラメーターの説明の中の式を参照するか、または DBDGEN の出力を

調べることにより判別できます。SIZE= が指定されていない場合に、アクセス方式が VSAM であれば、DBDGEN は、CI 内の未使用スペースを CI 内の各論理レコードに均等に分配することにより、最適な VSAM LRECL 値を計算します。SIZE= が指定されているか、あるいはデータベースが SHISAM である場合は、こうしたことは行われません。

次の表では、BLOCK= と RECORD= オペランドの使い方を説明しています。

表 4. BLOCK および RECORD オペランド

データベース・タイプ	BLOCK および RECORD オペランドの使い方
HSAM/SHSAM	<p><b>BLOCK=</b></p> <p>blkfact1 は、入力データ・セットに適用され、常に 1。</p> <p>blkfact2 は、出力データ・セットに適用され、常に 1。</p> <p><b>RECORD=</b></p> <p>reclen1 は、入力レコードの長さ。</p> <p>reclen2 は、出力レコードの長さ。</p> <p>HSAM/SHSAM は、常に非ブロック化される。LRECL と BLKSIZE は等しい。</p>
GSAM	<p><b>BLOCK=</b></p> <p>blkfact1 は、入出力データ・セットに適用される。</p> <p>blkfact2 は、無効なサブパラメーター。</p> <p><b>RECORD=</b></p> <p>reclen1 は、LRECL の長さであるか、可変長レコードの最大サイズ。</p> <p>reclen2 は、可変長レコードの最小サイズ。</p> <p><b>SIZE=</b></p> <p>size1 は、入出力データ・セットの BLKSIZE。</p> <p>size2 は、無効なサブパラメーター。</p>
HISAM/SHISAM	<p><b>BLOCK=</b></p> <p>blkfact1 は、1 次データ・セットのブロック化因数。</p> <p>blkfact2 は、オーバーフロー・データ・セットのブロック化因数。</p> <p><b>RECORD=</b></p> <p>reclen1 は、データ・セットの論理レコード長。</p> <p>reclen2 は、オーバーフロー・データ・セットの論理レコード長。</p>
HIDAM, HDAM	<p><b>BLOCK=</b></p> <p>size0 は、OSAM または VSAM データ・セット・グループのオーバーヘッドを含まないサイズ。</p> <p><b>RECORD=</b></p> <p>無視されます。</p>
MSDB	BLOCK= および RECORD= オペランドは無効。
DEDB	BLOCK= および RECORD= オペランドは無効。



表 4. BLOCK および RECORD オペランド (続き)

データベース・タイプ	BLOCK および RECORD オペランドの使い方
INDEX	<p><b>BLOCK=</b></p> <p>blkfact1 は、1 次データ・セットのブロック化因数。</p> <p>blkfact2 は、オーバーフロー・データ・セットのブロック化因数。</p> <p><b>RECORD=</b></p> <p>reclen1 は、1 次データ・セットの論理レコード長。</p> <p>reclen2 は、オーバーフロー・データ・セットの論理レコード長。</p>
LOGICAL	BLOCK= および RECORD= オペランドは無効。

注: DATASET ステートメントで reclen1 と reclen2 の両方を指定するときは、GSAM の場合を除き、reclen2 は reclen1 と等しいかそれ以上でなければなりません。

### SIZE=

DBDGEN ユーティリティによる制御インターバルまたはブロック・サイズの計算方法をオーバーライドします。SIZE= に指定した値が、アクセス方式サービス・プログラムを使用して VSAM に定義された制御インターバル・サイズと異なる場合には、DL/I は VSAM に定義された値を使用します。

DL/I DBD では、DBDGEN を使用しなくても、アクセス方式サービス・プログラムを使用して制御インターバル・サイズを VSAM に再定義すれば、DBD を効果的に修正できます。このようにすれば、DBDGEN を使用しなくても、データベースを新しい装置にマイグレーションすることができます。これを使用すると、指定した値にオーバーヘッドは加えられず、指定した値は IMS によって妥当性が検査されません。

VSAM データ・セットでは、指定する値が 8192 より小さい場合には、その値は 512 の倍数でなければなりません。その値が 512 の倍数でない場合は、DBDGEN はそれより大きな最も近い 512 の倍数に指定値を丸めて、警告メッセージを出します。8192 から 30720 バイト (最大許容サイズ) の範囲の指定値は、2048 の倍数にする必要があります。2048 の倍数でないと、DBDGEN はそれより大きな最も近い 2048 の倍数に指定値を丸めて、警告メッセージを出します。

HISAM、SHISAM、HIDAM 1 次索引、および副次索引データベースでは、size1 でデータ・セット・グループ内の 1 次データ・セットの制御インターバル・サイズまたはブロック・サイズを指定し、size2 でオーバーフロー・データ・セットの制御インターバル・サイズまたはブロック・サイズを指定します。

HDAM および HIDAM データベースでは、size1 パラメーターだけが使用されます。size1 パラメーターは、データ・セット・グループ内のデータ・セットの制御インターバル・サイズまたはブロック・サイズを指定します。

HISAM または索引データベースに SIZE を指定する場合には、RECORD パラメーターも指定する必要があります。指定するサイズ値は、関係しているデータ・セットを VSAM がオープンできるように、RECORD パラメーターの倍数にする必要があります。データベースに指定できる最小ブロック・サイズまたは最小制御インターバル・サイズを示す式を以下に示します。

OSAM データ・セットの最大ブロック・サイズは 32767 バイトです。ブロック・サイズが偶数長の OSAM データ・セットには、8 ギガバイトのサイズ限界があります。データベースをイメージ・コピーで保管する場合に、ブロック・サイズとして指定できる最大量は、32752 バイトです。イメージ・コピー処理モジュール DFSUDMP0 は、接頭部のダブル・ワード位置合わせのために、15 バイトをブロック・サイズに追加しますが、ブロック・サイズは 32767 を超えることはできません。DBDGEN ユーティリティーでブロック・サイズを指定する場合の最大値は 32752 バイトです。

**重要:** HISAM 1 次データ・セット・グループ、HIDAM 1 次索引データ・セット・グループ、および副次索引データ・セット・グループの SIZE= の計算方法 HISAM または INDEX データベースの 1 次データ・セット・グループの場合に、1 次データ・セットに指定できる最小制御インターバル・サイズは、1 次サイズによって与えられ、オーバーフロー・データ・セットの場合には、オーバーフロー・サイズによって与えられます。オーバーフロー・データ・セットは、データ・セット・グループに必ずしも必要ではありません。

- 1 次サイズ  $\geq$  ROOTSEG + OVERHEAD + VSAM CONTROL
- オーバーフロー・サイズ  $\geq$  MAXSEG + OVERHEAD + VSAM CONTROL

#### ROOTSEG=

セグメント接頭部を含む最大ルート・セグメント・サイズ。INDEX VSAM ルート・セグメント接頭部は、DL/I DOS を使用して作成された場合を除き、セグメント・コードを含みません。

#### OVERHEAD=

必要なバイト数は次のとおりです。

- 7 データベースに物理セグメント・タイプが複数入っている場合、OSAM に使用。
- 3 データベースに物理セグメント・タイプが 1 つしか入っていない場合、OSAM に使用。
- 4 ルート・セグメント・キーが固有でない INDEX VSAM データベースに使用。
- 0 DL/I DOS で作成されていない、固有のルート・セグメント・キーを持つ INDEX VSAM データベースに使用。

他のすべての VSAM データベースには 5 バイト。

#### VSAM CONTROL=

必要なバイト数は次のとおりです。

- 0 ブロック化因数が 1 の OSAM に使用。
- 7 ブロック化因数が 1 の VSAM に使用。
- 10 それ以外のすべての場合に使用。

#### MAXSEG=

セグメント接頭部が入っているこのデータ・セット・グループ内の最長セグメントの長さ (単位はバイト)。

**HIDAM 1 次データ・セット・グループの SIZE= の計算方法**

HDAM データベースの 1 次データ・セット・グループに対して指定できる最小のブロック・サイズまたは制御インターバル・サイズは、DBD ステートメントの RMNAME パラメーターの rbn パラメーターを指定するかどうかによって異なります。

- rbn を指定する場合は、次の 2 つの条件を満たしていなければなりません。
  - サイズ  $\geq$  (RAPS\*4) + FSEAP + 2 + VSAM CONTROL
  - サイズ  $\geq$  MAXSEG + FSEAP + VSAM CONTROL
- rbn を指定しない場合は、次の条件を満たしていなければなりません。
  - サイズ  $\geq$  MAXSEG + (RAPS\*4) + FSEAP + VSAM CONTROL

**RAPS=**

データベースのルート・アドレス可能域に対して指定するルート・アンカー・ポイントの個数。

**FSEAP=**

フリー・スペース・エレメント・アンカー・ポイントの場合は 4 バイト。

**VSAM CONTROL=**

OSAM の場合は 0 バイト、VSAM の場合は 7 バイト。

**MAXSEG=**

セグメント接頭部が入っているこのデータ・セット・グループ内の最長セグメントの長さ (単位はバイト)。

**HDAM 2 次データ・セット・グループの SIZE= の計算方法**

指定するブロック・サイズまたは制御インターバル・サイズは、次のようにする必要があります。

サイズ  $\geq$  MAXSEG + FSEAP + VSAM CONTROL

**MAXSEG=**

セグメントが入っているこのデータ・セット・グループ内の最長セグメントの長さ (単位はバイト)。

**FSEAP=**

フリー・スペース・エレメント・アンカー・ポイントの場合は 4 バイト。

**VSAM CONTROL=**

OSAM の場合は 0 バイト、VSAM の場合は 7 バイト。

**HIDAM データ・セット・グループの SIZE= の計算方法**

HIDAM データベースのデータ・セット・グループに対して指定できる最小のブロック・サイズまたは制御インターバル・サイズは、指定するアクセス方式によって異なります。さらに、1 次データ・セット・グループのブロック・サイズまたは制御インターバル・サイズは、ルート・セグメント・タイプに指定されるポインター・タイプによっても異なります。

HIDAM データベースのルート・セグメント・タイプに順方向のみの階層ポインターまたは物理兄弟ポインターを指定すると、1 次データ・セット・グループに対して指定するブロック・サイズまたは制御インターバル・サイズは、次の値でなければなりません。

サイズ  $\geq$  MAXSEG + FSEAP + RAP + VSAM CONTROL

1 次/2 次データ・セット・グループの他の条件のもとでは、指定するブロック・サイズまたは制御インターバル・サイズは、次のようにする必要があります。

サイズ  $\geq$  MAXSEG + FSEAP + VSAM CONTROL

**MAXSEG=**

セグメント接頭部が入っているこのデータ・セット・グループ内の最長セグメントの長さ (単位はバイト)。

**FSEAP=**

フリー・スペース・エレメント・アンカー・ポイントの場合は 4 バイト。

**VSAM CONTROL=**

OSAM の場合は 0 バイト、VSAM の場合は 7 バイト。

**RAP=**

1 つのルート・アンカー・ポイントにつき 4 バイト。

**RECORD=(reclen1,reclen2)**

このデータ・セット・グループに使用するデータ管理論理レコード長を指定します。このパラメーターは、オプションですが、ACCESS=LOGICAL が DBD ステートメントで指定されている場合は指定できません。reclen1 および reclen2 は数値にする必要があります。reclen2 の値は、常に reclen1 に等しいかそれより大きくなければなりません (ただし、GSAM データベースの場合を除きます)。パラメーターの各パラメーターの意味は、定義されるデータベースのタイプによって異なります。

単純 HISAM (SHISAM) データベースの場合は、指定する論理レコード長は、指定するセグメント長と同じにする必要があります。HISAM および INDEX DBD の許容最小論理レコード長は、VSAM CONTROL が無視されることを除けば、DATASET SIZE= パラメーターに記述された最小のブロック・サイズまたは制御インターバル・サイズと同じです。さらに、HISAM および INDEX DBD の VSAM KSDS および ESDS 両方の場合は、指定する論理レコード長を偶数値にする必要があります。

VSAM 1 次索引 (INDEX, VSAM) データベースの場合は、オーバーフロー論理レコード長 (reclen2) パラメーターを定義してはなりません。すべての索引セグメントはキー順データ・セットに挿入されるからです。

GSAM データベースの場合は、reclen1 で、固定長レコードの論理レコードのサイズ、または可変長または未定義レコードの最大サイズを指定します。reclen2 の値には、可変長または未定義レコードの最小サイズを指定します。可変長 GSAM/BSAM データベースの場合、IMS はアプリケーションによって渡される GSAM レコード内のレコード長値に、2 バイトを追加します。これは、レコードが入出力装置に書き込まれるときに、BSAM レコード記述子ワード (RDW) を形成する ZZ フィールドを格納するためです。

**RECFM=**

データ・セット内のレコードのフォーマットを指定します。レコード・フォーマットは、次のように定義されている文字を使用して指定します。

**F** レコードは固定長。

**FB** レコードは固定長でブロック化。

- V      レコードは可変長。
- VB     レコードは可変長でブロック化。
- U      レコードは未定義長。

このパラメーターは、GSAM データベースの場合にのみ有効です。

#### SCAN=cyls

セグメント挿入操作時に使用可能ストレージ・スペースを検索するときに、スキャンする直接アクセス装置のシリンダーの数を指定します。このパラメーターはオプションです。これは、この DBD 生成が HIDAM または HDAM データベースを定義するときのみ使用されます。これを指定する場合は、cyls は 255 を超えない 10 進整数でなければなりません。通常値は 0 から 5 です。デフォルト値は 3 です。SCAN=0 を指定すると、現行シリンダーのスペースのみがスキャンされます。

スキャンは、現行シリンダー位置から両方向に実行されます。スキャン限界値が原因で、現行エクステンツの外側の領域がスキャンに含まれる場合には、IMS は、スキャンが現行エクステンツの境界を超えないようにスキャン限界を調整します。このパラメーターで定義されたシリンダー範囲内にセグメント挿入のためのスペースが見つからない場合は、データベースのデータ・セット・グループの現在の終わりのスペースが使用されます。

#### FRSPC=

HDAM または HIDAM データベース内にフリー・スペースをどのように分散させるかを指定します。空きブロック頻度係数 (free block frequency factor) (fbff) は、このデータ・セット・グループ内の n 番目ごとの制御インターバルまたはブロックが、データベースのロードまたは再編成時にフリー・スペースとして残されることを指定します (ここで、fbff=n)。fbff の範囲には、fbff=1 を除いて、0 から 100 のすべての整数値が含まれます。fspf はフリー・スペースのパーセント係数 (free space percentage factor) です。これは、フリー・スペースとしてこのデータ・セット・グループに残す各制御インターバルまたはブロックの最小パーセントを指定します。fspf の範囲は 0 から 99 です。fbff および fspf のデフォルト値は 0 です。フリー・スペースのパーセントとセグメント・サイズの合計が制御インターバル・サイズまたはブロック・サイズを超える場合は、サイズ超過のセグメントにフラグを立てた警告メッセージを DBDGEN が出します。サイズ超過のセグメントをロードすると、「fspf」の指定は無視され、1 つの制御インターバルまたはブロックを使用して、サイズ超過のセグメントをロードします。

第 1 パラメーター FBFF を指定するとき値を小さくすると、データベース内のフリー・スペースの頻度が増大するという理解してください。例えば、値を 2 にすると、各データの後にフリー・スペース・ブロックが生じることとなります。この場合には、再編成ユーティリティまたはロード・ユーティリティの実行時に、フリー・スペース・ブロックのために余分の処理が必要になるので、システム・パフォーマンスが低下することとなります。

#### SEARCHA=

IMS が HD データベースにセグメントを挿入するときに使用する、HD スペース検索アルゴリズムのタイプを指定します。

- 0    IMS が使用する HD スペース検索アルゴリズムを選択することを指定しま

す。これはデフォルトです。このリリースでは、IMS は、SEARCHA=2 が指定された場合と同じアルゴリズムを使用します。

- 1 IMS が使用する HD スペース検索アルゴリズムは、2 番目に望ましいブロックまたは CI 内のスペースを検索しないという指定です。
- 2 IMS が使用する HD スペース検索アルゴリズムは、2 番目に望ましいブロックまたは CI 内のスペースを検索するという指定です。

#### **REL=**

MSDB が非端末関連 (NO または TERM) MSDB であるか、端末関連 (FIXED および DYNAMIC) MSDB であるかを定義します。非端末関連 MSDB のセグメントには所有権はありません。

IMS の第 5 版またはそれ以上の版の ETO では、端末関連キーの付いた MSDB はサポートされていません。これ以外のタイプの MSDB は継続してサポートされています。

端末関連 MSDB では、各セグメントは異なる LTERM に割り当てられます。LTERM 名は、セグメント・キーですが、セグメント内には含まれていません。各 LTERM は、1 つの MSDB につき 1 つのセグメントのみを所有し、所有者しかセグメントを変更できません。

**NO** 端末関連キーのない非端末関連 MSDB を指定します。キーおよびシーケンス・フィールドは、セグメントの一部に含まれます。

#### **TERM**

端末関連キーのある非端末関連 MSDB を指定します。キーは LTERM 名 (セグメントの一部ではない) であり、シーケンス・フィールドはありません。

#### **FIXED**

端末関連固定 MSDB を指定します。LTERM 名はセグメント・キーです。セグメントの更新を行うことができます。セグメントの挿入および削除は行えません。

#### **DYNAMIC**

端末関連動的 MSDB を指定します。LTERM 名はセグメント・キーです。セグメントの挿入および削除を行えます。1 つの同期処理インターバル内で、単一の LTERM から同じ MSDB に行える挿入または削除は 1 つのみです。

#### **search field name**

1 から 8 文字の英数字名を指定します。この名前は、FIELD ステートメントに定義された他のフィールド名と同じにはなりません。

シーケンス・フィールドは、LTERM 名をセグメント・キー (REL=TERM、FIXED、または DYNAMIC) として使用して MSDB について定義することはできないので、修飾呼び出しを行えるように検索フィールド名が用意されています。SSA で有効な値は LTERM 名のみです。したがって、検索フィールドは 8 バイト文字フィールドとして扱われ、他の定義は用意されていません。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### **CORRECT**

```
REMARKS='These remarks apply to the XYZ application'
```

#### **INCORRECT**

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## **IMS データ・セット・グループのデータ・セット**

各 IMS データベース内の非 HALDB データ・セットの DD ステートメントは、そのデータベースにアクセスする各ジョブで用意する必要があります。メッセージ処理プログラムまたはバッチ・メッセージ処理プログラムが使用するデータベースの場合は、IMS 制御領域用の JCL に DD ステートメントを含める必要があります。バッチ処理環境で排他使用されるデータベースの場合は、バッチ処理領域用の JCL に DD ステートメントを含める必要があります。z/OS オンライン環境では、データベースを動的に割り振ることができます。

HALDB データ・セットの場合は DD ステートメントは必要ありません。動的に割り振られるからです。

### **VSAM で必要な DD ステートメント**

データベース用のオペレーティング・システム・アクセス方式が VSAM の場合は、KSDS ごとに 1 つの DD ステートメントと、ESDS ごとに 1 つの DD ステートメントが必要です。DD ステートメントで必要なパラメーターの形式は次のとおりです。

```
//DDname DD DISP=SHR,DSNAME=
```

VSAM データ・セットはすべてカタログされているため、UNIT=、VOLSER=、および SPACE= パラメーターは、指定する必要はありません。

HISAM データベースの場合は、2 つの DD ステートメントが必要です。1 つは KSDS 用で、もう 1 つは ESDS 用です。HISAM データベースが 1 つのセグメント・タイプだけしか定義していない場合には、KSDS DD ステートメントのみ必要となります。

HDAM または HIDAM データベースの場合は、データ・セット・グループごとに 1 つの DD ステートメントが必要です。HIDAM データベースの 1 次索引の場合は、KSDS 用に 1 つの DD ステートメントが必要です。

固有キーを持つ副次索引データベースの場合は、KSDS 用に 1 つの DD ステートメントが必要です。

固有でないキーを持つ副次索引データベースの場合は、2 つの DD ステートメントが必要です。1 つは KSDS 用であり、もう 1 つは ESDS 用です。非固有キーのある副次索引データベースは、HALDB ではサポートされていないことに注意してください。VSAM データ・セットを定義する DD ステートメントに加えて、IMS VSAM バッファ・プールを定義するパラメータを収容するデータ・セットを指定する DD ステートメントを、バッチ領域用として用意する必要があります。この DD ステートメントの DD 名は DFSVSAMP です。IMS のオンライン実行では、この情報は、IMS.PROCLIB データ・セットのメンバー (メンバー名 DFSVSMxx) 内に用意されています。

### OSAM で必要な DD ステートメント

HSAM または SHSAM の場合は、入力または出力用の DD ステートメントを下記の形式で用意しなければなりません。

```
//DDname DD DSNAME= ,UNIT= ,VOL=SER= ,  
//          DISP= ,DCB=
```

DD ステートメントが HSAM または SHSAM 出力データ・セット用である場合は、データ・セットを事前割り振りしておくか、あるいは SPACE= パラメータを指定しておく (直接アクセス記憶装置を使用するとき) 必要があります。

RECFM=FB はオプションですが、指定する場合は、ロード時に指定しなければなりません。RECFM=F を指定してはなりません。

OSAM データ・セットの場合は、DCB パラメータの LRECL、BLKSIZE、および BUFL サブパラメータを指定しないでください。この情報は DBD から入手されるもので、変更することができません。

HDAM または HIDAM の場合は、各データ・セット・グループの OSAM データ・セット用の DD ステートメントが必要です。その形式は次のとおりです。

```
//dd1 DD DSNAME= ,UNIT= ,VOL=SER= ,  
//          DISP= ,DCB=(DSORG=PS[,OPTCD=W])
```

HDAM または HIDAM データベースを作成するときは、OSAM データ・セットを事前に割り振っておくか、SPACE= パラメータを指定する必要があります。

モデル DSCB を使用して世代別データ・セットを記述する場合は、モデル DSCB に LRECL、RECFM、および BLKSIZE パラメータを指定しないでください。この情報は DBD から入手されるもので、変更することができません。

## AREA ステートメント

DEDB データベースでは、AREA ステートメントを使用してデータベース内の区域を定義します。



DEDB 用の DBDGEN 入力デッキでは、DBD ステートメントと最初の SEGM ステートメントの間にすべての AREA ステートメントを入れる必要があります。少なくとも 1 つの AREA ステートメントが必要であり、複数の区域を定義するのに最大 2048 個の AREA ステートメントを使用することができます。

制約事項: AREA ステートメントを HALDB データベースに使用することは許されません。区分は DBDGEN の外側に定義されます。

```

(1)
▶▶—AREA—DD1=ddname1—,SIZE=size1———,UOW=(number1,overflow1)—————▶
▶—,ROOT=(number2,overflow2)—┌,DEVICE=3380┐—————▶
└,REMARKS=comments┘

```

注:

- 1 DEDB SIZE キーワードの有効なパラメーター指定は、512 バイト、1 KB、2 KB、4 KB、8 KB、および 28 KB までの 4 KB の倍数です。将来の互換性を確保するために、4 KB の倍数である CI サイズのみを使用してください。

## AREA ステートメント・パラメーターの説明

### AREA

このステートメントが DEDB の AREA 制御ステートメントであることを表します。

### DD1=

定義する区域の DD 名を指定します。ddname1 は、1 から 8 文字の英数字 (A-Z、0-9、#、@、\$) 名でなければなりません。このパラメーターは、単一エリア・データ・セットの場合は区域名でも DD 名でも構いませんが、複数エリア・データ・セットの場合は区域名のみが可能です。データベースが DBRC に登録されている場合は、このパラメーターに区域名を指定してください。

### DEVICE=

この区域内のデータ・セットが格納される物理ストレージの装置タイプを指定します。デフォルトは 3380 です。これ以外の装置をコーディングすると、デフォルトは無視されます。

### SIZE=

制御インターバルを指定します。サイズは、512 バイト、1 KB、2 KB、4 KB、8 KB のいずれか、および 28 KB までの 4 KB の倍数にすることができます。将来の互換性のために、4 KB の倍数である CI サイズのみを使用してください。デフォルト値は許されていません。

制約事項: 2319 装置には 4 KB を指定できません。

DEDB の場合には、DBDGEN SIZE= を、VSAM に定義された制御インターバル・サイズと一致させる必要があります。IMS はデータ・セットにアクセスする際にこの値を使用するからです。制御インターバル・サイズが VSAM データ・セットの中で変更される場合、その区域の DBD を新しい SIZE= 値に変更しなければなりません。

**UOW=**

作業単位 (UOW) の制御インターバルの数を指定します。UOW= パラメーターには、2 つのオペラント、number1 と overflow1 があります。

**number1**

作業単位 (UOW) の制御インターバルの数を指定します。この値は、2 から 32767 の値にする必要があります。

**overflow1**

UOW のオーバーフロー・セクション中の制御インターバルの数を指定します。overflow1 は 1 より大きいまたは等しい値にすることができますが、少なくとも number1 の指定値よりも 1 だけ小さな値です。

1 つの UOW 内のルート・アンカー・ポイント (RAP) の合計数は、number1 から overflow1 を引いて得られます。1 つの UOW 内の RAP 数にルート・アドレス可能部分の UOW 数を掛けると、1 つの区域内的 RAP の合計数が求まります。

**REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

**CORRECT**

```
REMARKS='These remarks apply to the XYZ application'
```

**INCORRECT**

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

**ROOT=**

DEDB エリアの特性を指定します。ROOT= パラメーターには、2 つのオペラント、number2 と overflow2 があります。

**number2**

区域のルート・アドレス可能部分と独立オーバーフロー用の予約区域に割り振られる合計スペースを指定します。これは、UOW で表されます。VSAM データ・セットの残りの部分は、順次従属データ用に予約されます。この値は、2 より大きく、32767 より小さい値とし、VSAM データ・セットの実際のスペース量より大きくすることはできません。

## overflow2

独立オーバーフロー用に予約するスペース (UOW) を指定します。これは少なくとも 1 であり、number2 の指定値よりも小さくする必要があります。独立オーバーフローは UOW を含んでいませんが、スペース割り振りの単位として UOW サイズが使用されます。

再編成 UOW は、DEDB 初期設定ユーティリティーにより自動的に割り振られます。VSAM スペース定義には、この追加の UOW を含めてください。つまり、必要な合計スペースは、ルート・アドレス可能域、独立オーバーフロー、および再編成用の 1 つの追加 UOW になります。再編成 UOW は、高速 DEDB 直接再編成ユーティリティーでは使用されませんが、IMS の他の機能で使用される場合があります。

例：この例では、2048\*64\*936 バイトが割り振られ、残りの区域は順次従属セグメント用に残されます。

```
AREA DD1=XX,SIZE=2048,  
      UOW=(64,14),  
      ROOT=(936,36)
```

1 つの制御インターバル当たり 1 つのルート・アンカー・ポイント (RAP) しかないため、区域内の RAP の合計数は  $(64-14)*(936-36) = 45000$  RAP になります。

DBDGEN により独立オーバーフローに割り振られるスペース量は、IMS がオンラインの間に増やすことができます。

## SEGM ステートメント

SEGM ステートメントは、セグメント・タイプ、そのセグメントのデータベース階層での位置、そのセグメントの物理的特性、およびそのセグメントと他のセグメントとの関連を定義します。

GSAM データベース以外は、各 DATASET ステートメントの直後に少なくとも 1 つの SEGM ステートメントがなければなりません。SEGM ステートメントにより定義されるセグメントは、DATASET ステートメントで定義されるデータ・セット・グループに入れられます。MSDB および DEDB 以外は、DBD 生成で最大 255 個の SEGM ステートメントを指定できます。MSDB の場合に指定できる SEGM ステートメントは、1 つのみです。DEDB の場合には、1 から 127 個の SEGM ステートメントを、最後の AREA ステートメントの直後に続ける必要があります。DBD 生成で他の SEGM ステートメントを与えることはできません。SEGM ステートメントは、階層順序で入力ファイルに配置する必要があります。最大 15 の階層レベルを定義できます。

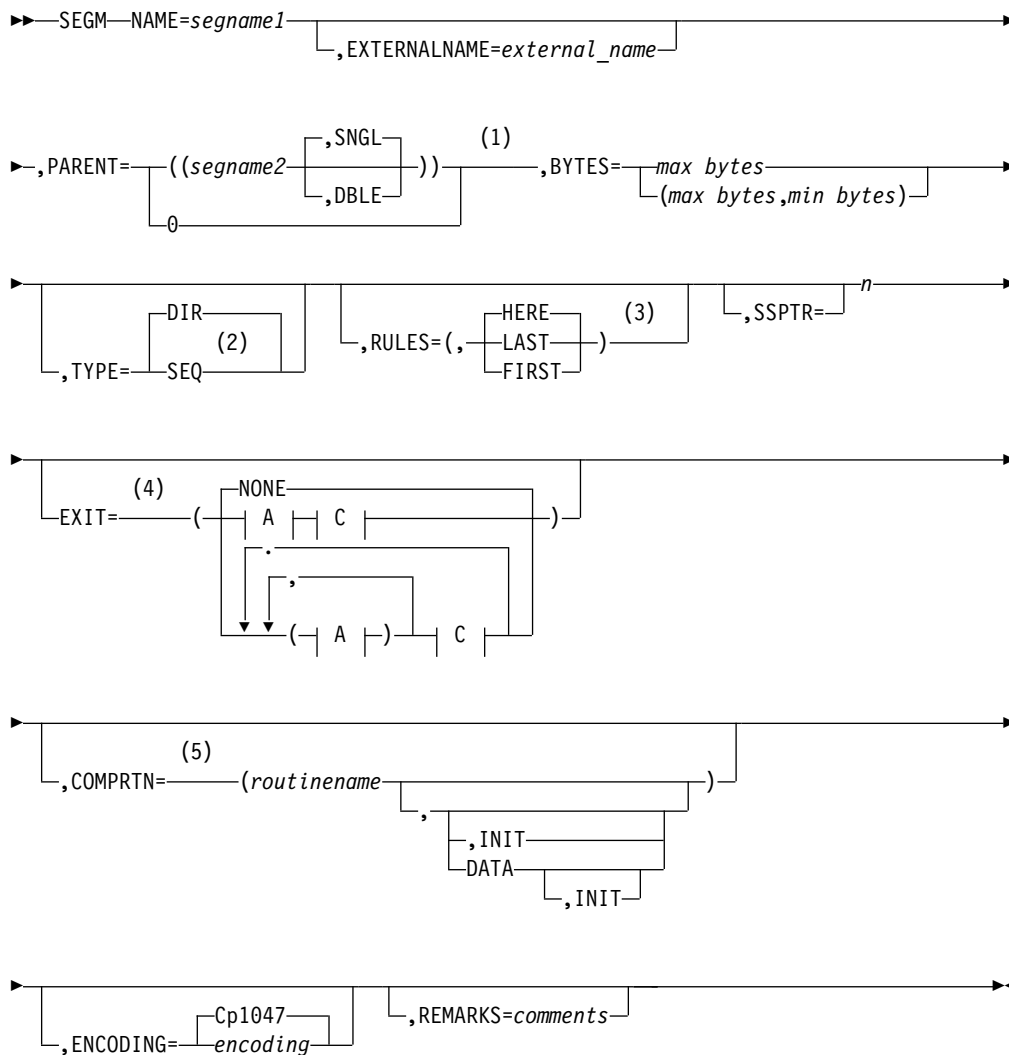
GSAM データベースの場合、DBD 生成に SEGM ステートメントを 1 つ指定することができます。この SEGM ステートメントは、検索不可フィールド (これ以外の場合は COBOL コピーブックでのみマップされる) の定義に使用できる仮想セグメントを定義します。結果フィールドのメタデータは、IMS カタログに格納されます。IMS カタログは、IMS Universal ドライバー経由で GSAM データベースにアクセスするアプリケーション・プログラムおよび製品で使用するためのものです。

SEGM ステートメントは、FIELD、XDFLD、および LCHILD ステートメントと一緒に使用して、全体として 1 つのセグメントを IMS に定義します。FIELD ステ

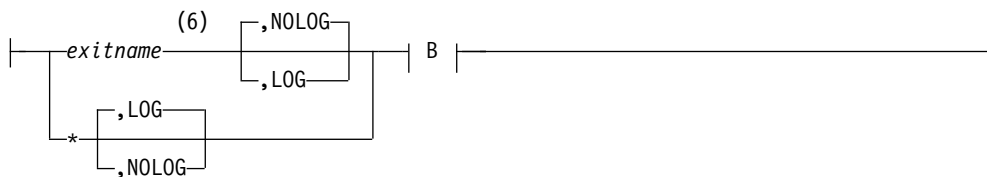
ートメントはセグメント内のフィールドを定義し、XDFLD ステートメントは副次索引に使用されるフィールドを定義し、LCHILD ステートメントはセグメント間の索引関係または論理関係を定義します。

データベース・タイプごとの SEGM ステートメントの形式を、以下の例に示します。

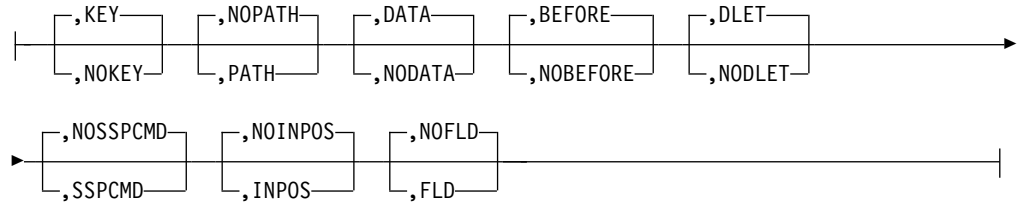
### DEDB データベースの SEGM ステートメント



**A:**



**B:**



**C:**



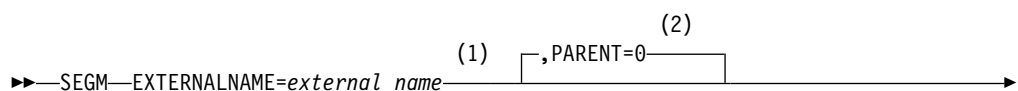
**D:**



注:

- 1 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。
- 2 順次従属タイプについては、SEGMENT ステートメントに TYPE=SEQ を指定する必要があります。
- 3 セグメント・タイプに固有のシーケンス・フィールドがない場合に必要です。デフォルトは HERE です。高速機能順次従属セグメント処理を使用する場合は、FIRST の挿入規則が常に使用され、これを変更することはできません。DEADB の直接従属セグメント処理の場合のデフォルトは、HERE です。
- 4 データ・キャプチャー出口ルーチンに使用します。1 つの SEGMENT ステートメントに複数の出口ルーチンを指定することができます。
- 5 セグメント編集/圧縮出口ルーチンに使用されます。
- 6 ロギングのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメータは LOG です。出口ルーチン名を指定すると、デフォルト・ロギング・パラメータは NOLOG になります。
- 7 CASCADE オプションの制御に使用します。

## GSAM データベースの SEGMENT ステートメント

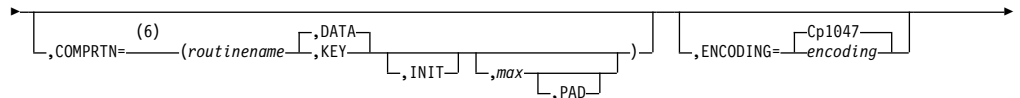
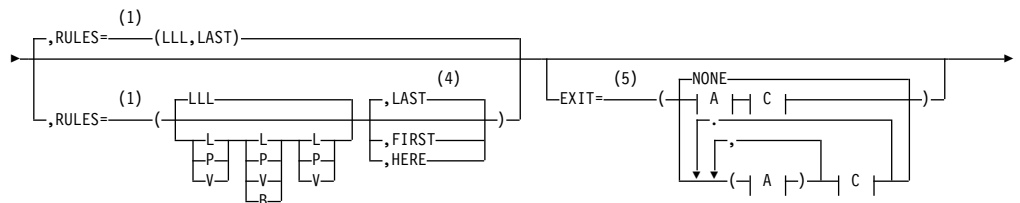
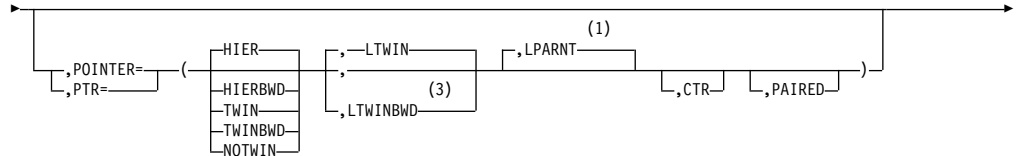
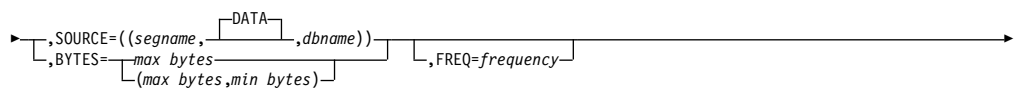
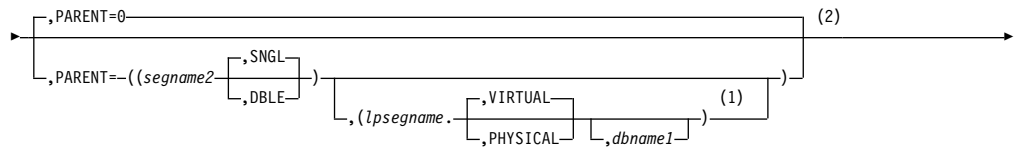
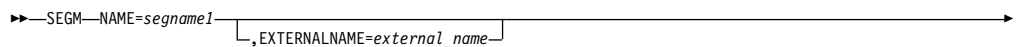




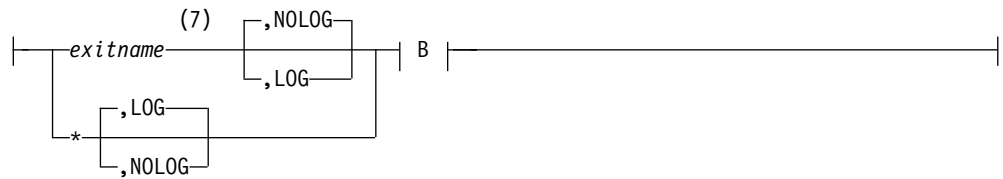
注:

- 1 NAME= は GSAM には無効です。EXTERNALNAME は必須です。
- 2 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。

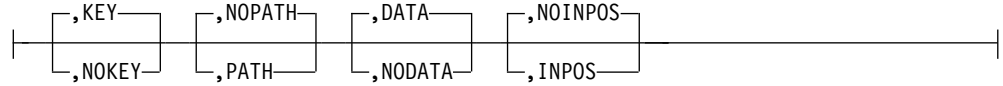
## HDAM データベースの SEGM ステートメント



A:



**B:**



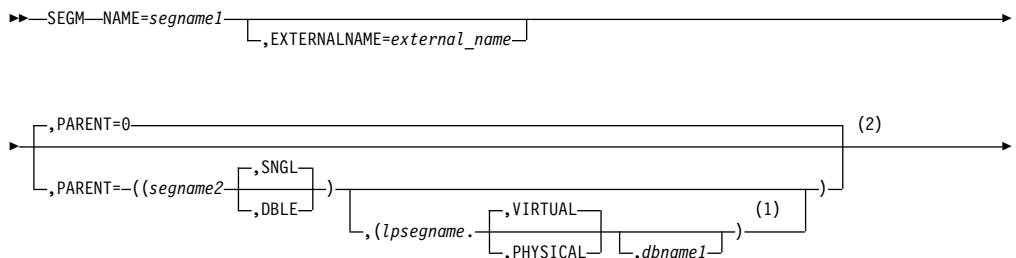
**C:**

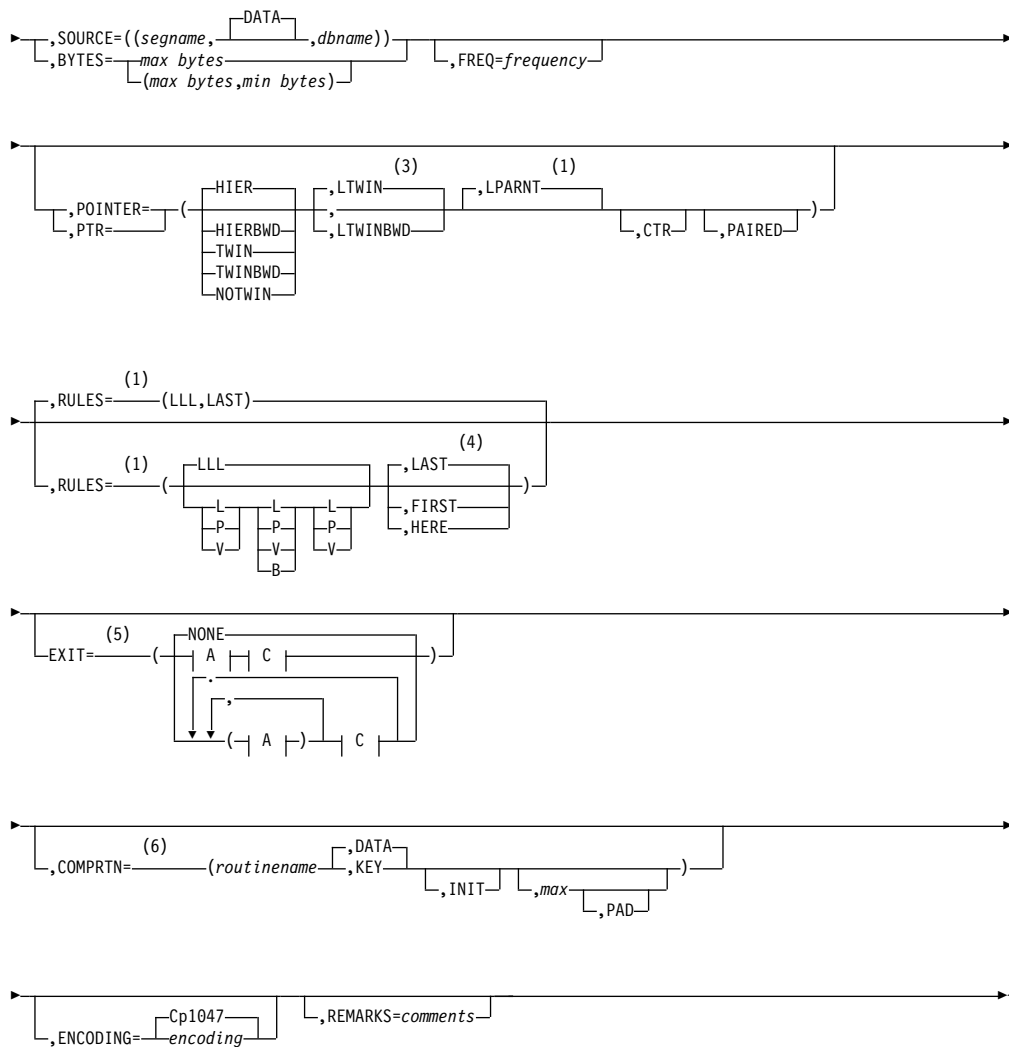


注:

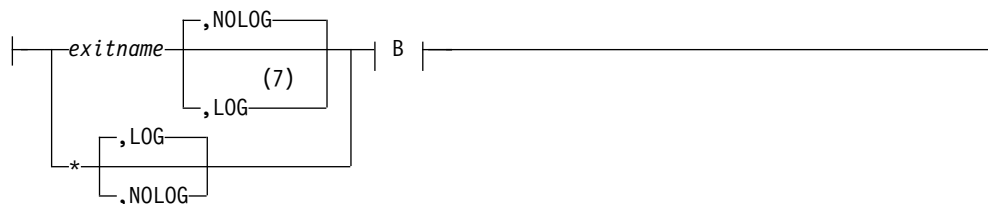
- 1 HDAM の論理関係にはオプションです。
- 2 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。
- 3 HDAM の論理関係には必要ですが、それ以外の場合はオプションです。
- 4 セグメント・タイプに固有のシーケンス・フィールドがない場合に必要です。デフォルトは LAST です。
- 5 データ・キャプチャー出力ルーチンに使用します。1 つの SEGM ステートメントに複数の出力ルーチンを指定することができます。
- 6 セグメント編集/圧縮出力ルーチンに使用されます。
- 7 ロギングのみが要求されているため出力ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメータは LOG です。
- 8 CASCADE オプションの制御に使用します。

## HIDAM データベースの SEGM ステートメント

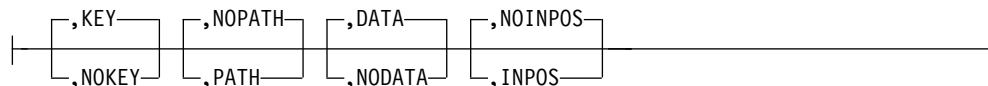




**A:**

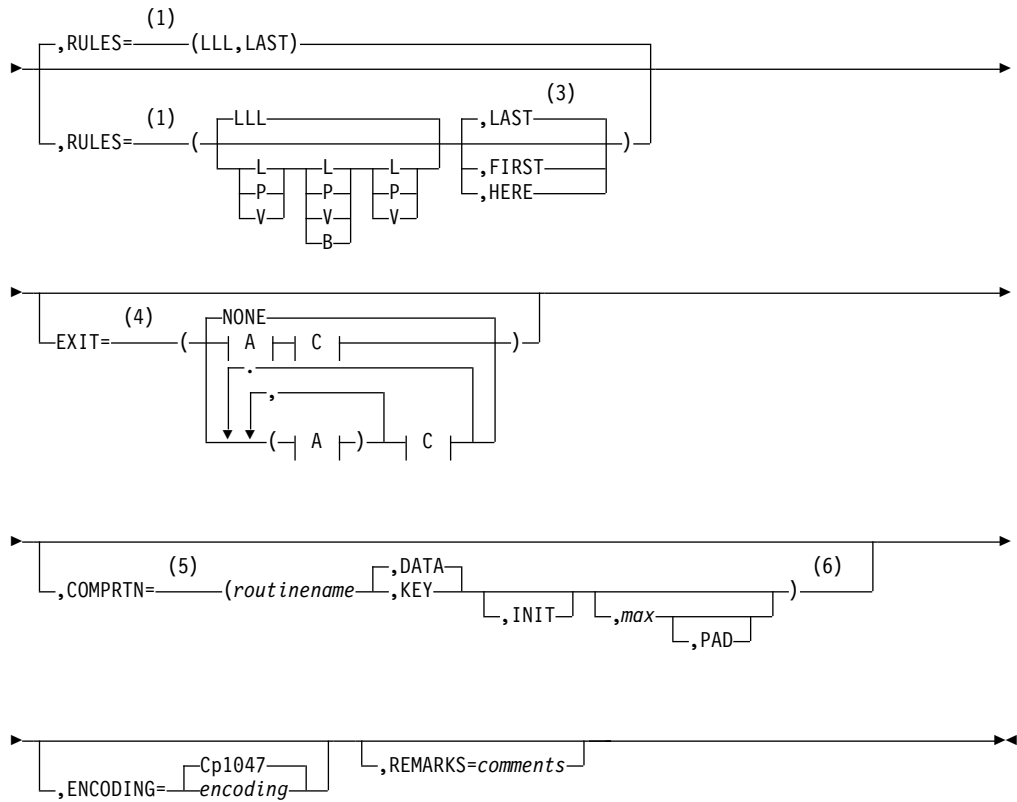


**B:**

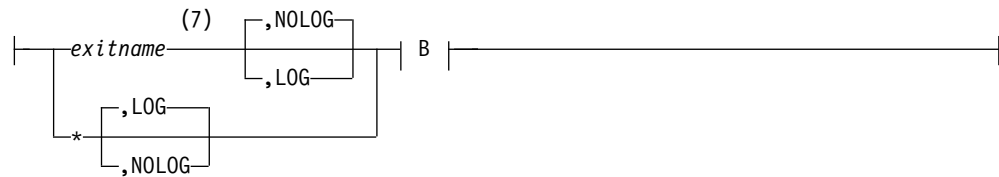




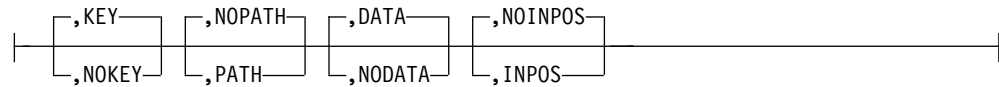




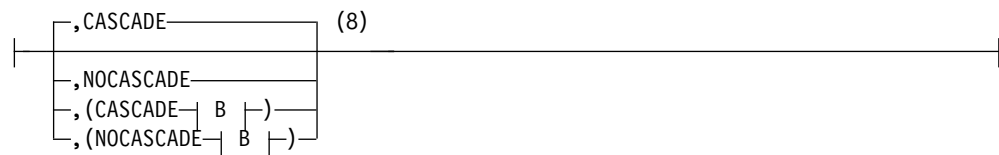
**A:**



**B:**



**C:**

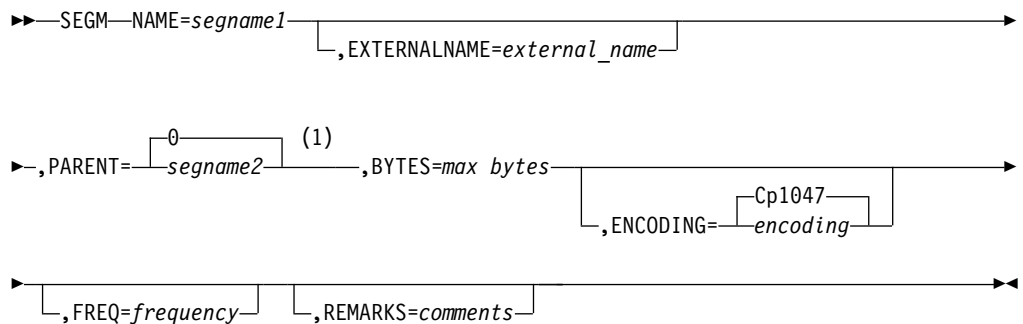


**注:**

1 HISAM の論理関係には必要ですが、それ以外の場合はオプションです。

- 2 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。
- 3 セグメント・タイプに固有のシーケンス・フィールドがない場合に必要です。デフォルトは LAST です。
- 4 データ・キャプチャー出力ルーチンに使用します。1 つの SEGM ステートメントに複数の出力ルーチンを指定することができます。
- 5 セグメント編集/圧縮出力ルーチンに使用されます。
- 6 単純 HISAM データベースには、可変長セグメントとセグメント編集/圧縮を指定することはできません。
- 7 ロギングのみが要求されているため出力ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメーターは LOG です。
- 8 CASCADE オプションの制御に使用します。

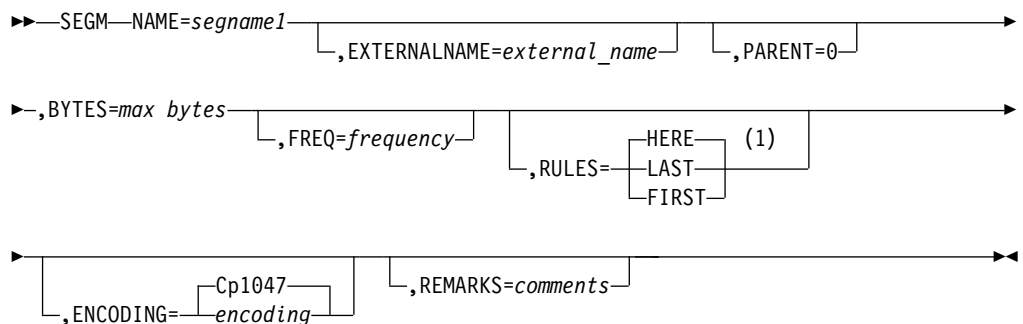
### HISAM データベースの SEGM ステートメント



注:

- 1 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。

### INDEX データベースの SEGM ステートメント

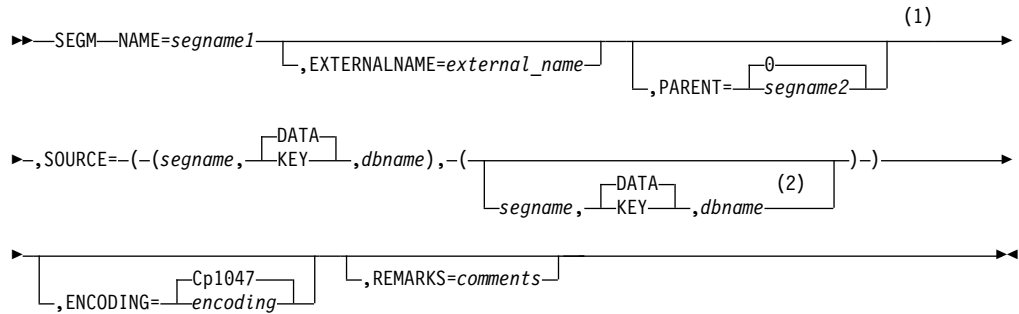


注:

- 1 セグメント・タイプに固有のシーケンス・フィールドがない場合に必要です。デフォルトは HERE です。高速機能順次従属セグメント処理を使用する場合

は、FIRST の挿入規則が常に使用され、これを変更することはできません。  
 DEDB の直接従属セグメント処理の場合のデフォルトは、HERE です。

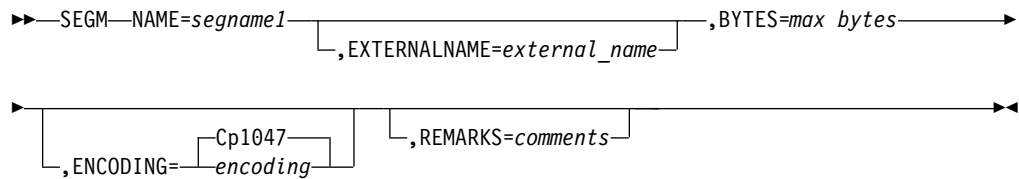
## LOGICAL データベースの SEGM ステートメント



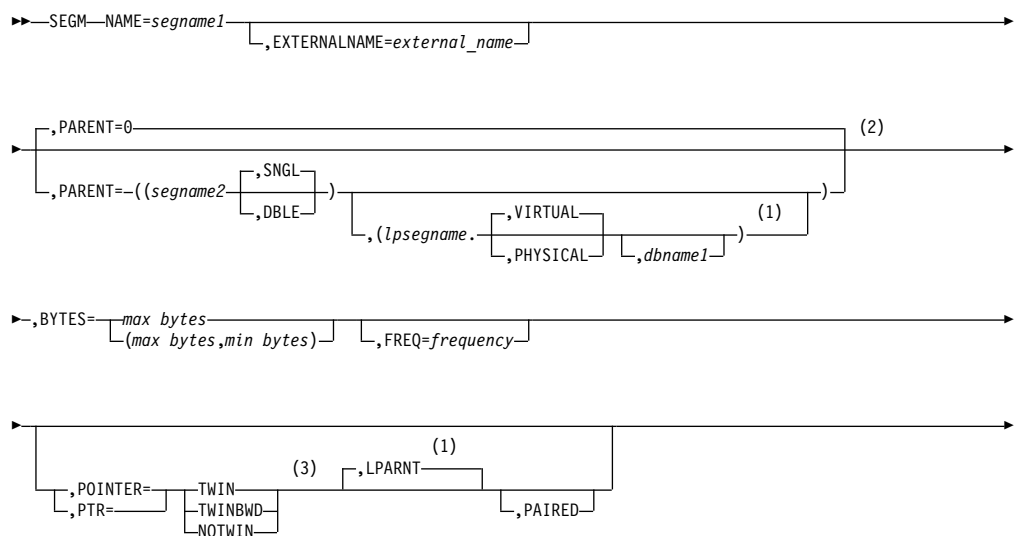
注:

- 1 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。
- 2 連結セグメント・タイプを定義する場合は必須です。

## MSDB データベースの SEGM ステートメント



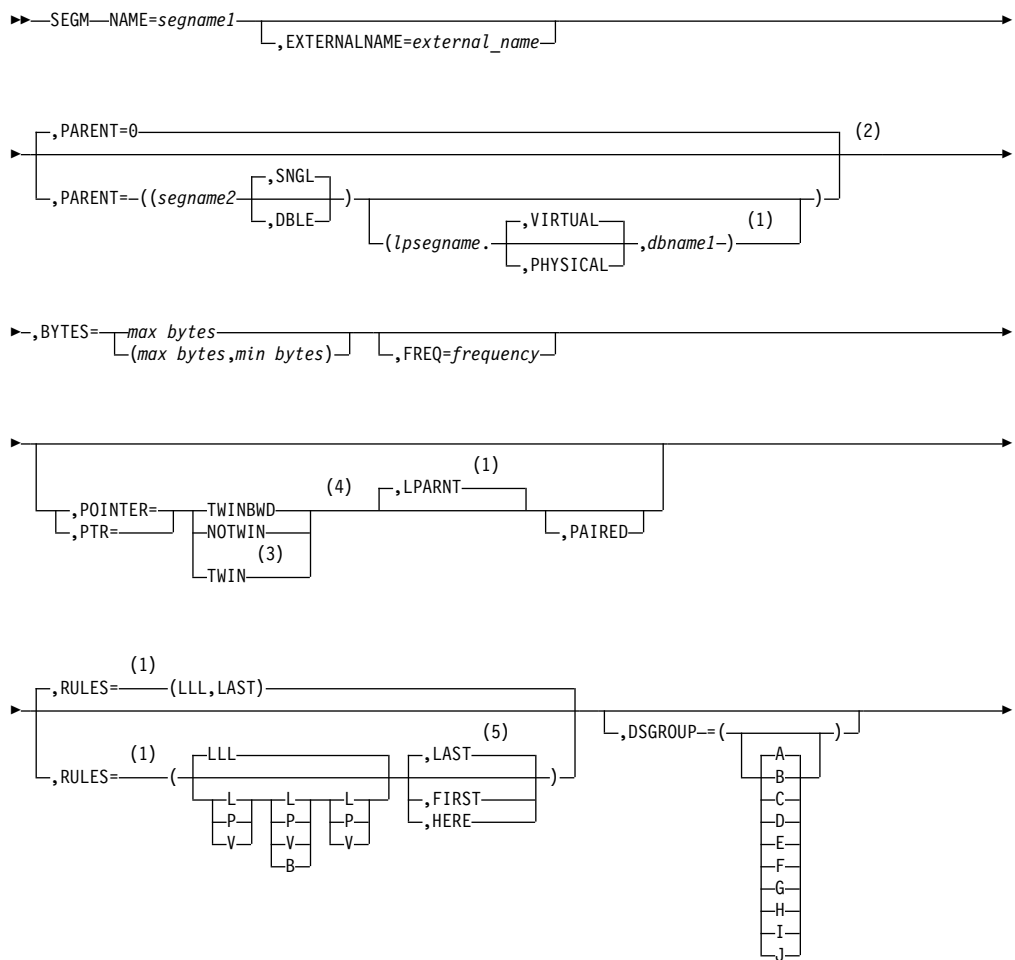
## PHDAM データベースの SEGM ステートメント

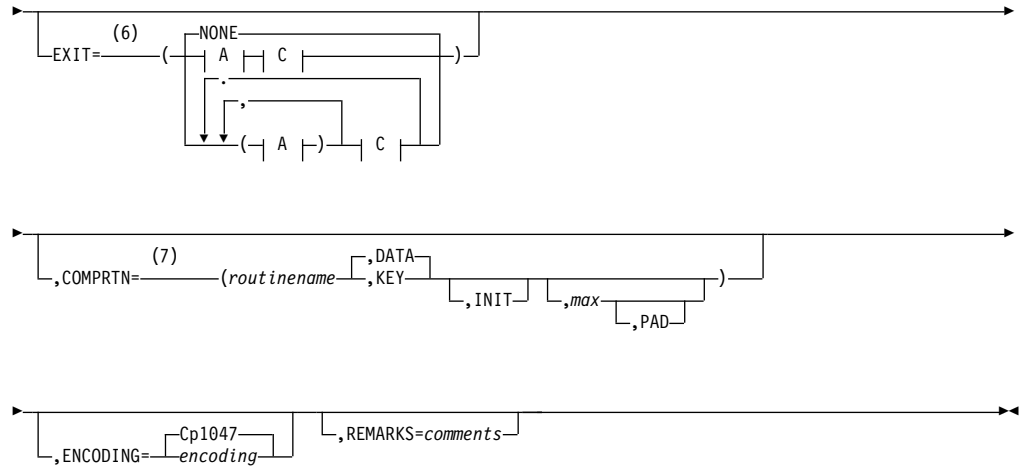




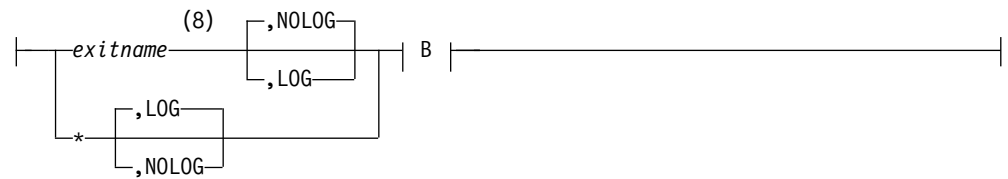
- 2 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。
- 3 PHDAM の論理関係には必要ですが、それ以外の場合はオプションです。
- 4 セグメント・タイプに固有のシーケンス・フィールドがない場合に必要です。デフォルトは LAST です。
- 5 データ・キャプチャー出口ルーチンに使用します。1 つの SEGM ステートメントに複数の出口ルーチンを指定することができます。
- 6 セグメント編集/圧縮出口ルーチンに使用されます。
- 7 ロギングのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメーターは LOG です。
- 8 CASCADE オプションの制御に使用します。

### PHIDAM データベースの SEGM ステートメント

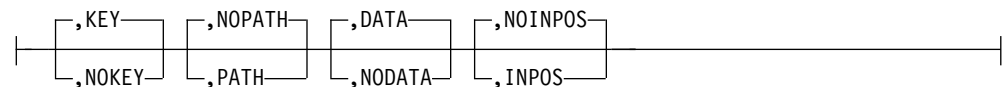




**A:**



**B:**



**C:**

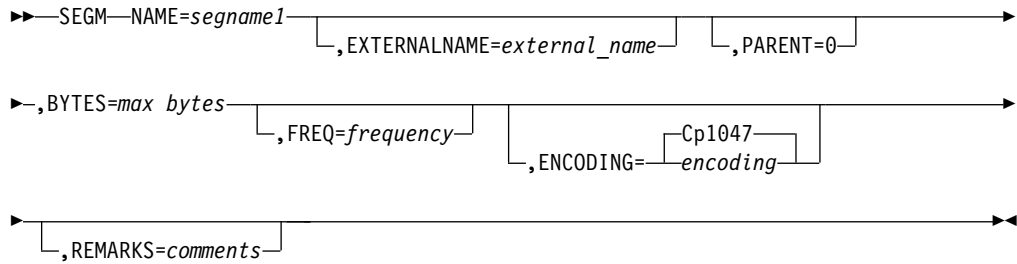


**注:**

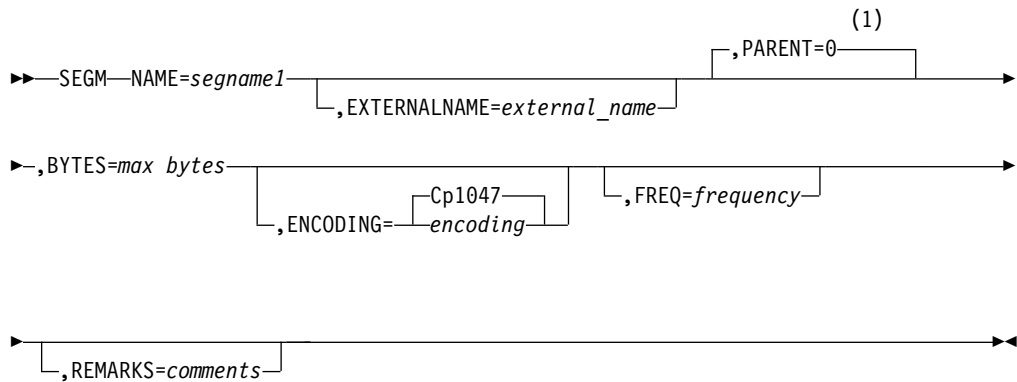
- 1 PHIDAM の論理関係にはオプションです。
- 2 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。
- 3 ルート・セグメントに TWIN を使用することは許されません。
- 4 PHIDAM の論理関係には必要ですが、それ以外の場合はオプションです。
- 5 セグメント・タイプに固有のシーケンス・フィールドがない場合に必要です。デフォルトは LAST です。
- 6 データ・キャプチャー出口ルーチンに使用します。1 つの SEGM ステートメントに複数の出口ルーチンを指定することができます。

- 7 セグメント編集/圧縮出口ルーチンに使用されます。
- 8 ロギングのみが要求されているため出口ルーチン名が不要な場合は、出口名にアスタリスク (\*) を指定します。この場合のデフォルトのロギング・パラメータは LOG です。
- 9 CASCADE オプションの制御に使用します。

## PSINDEX データベースの **SEGM** ステートメント



## SHSAM データベースの **SEGM** ステートメント



注:

- 1 データベースのルート・セグメント・タイプについては、PARENT= キーワードを省略するか、または PARENT=0 を指定することができます。

## **SEGM** ステートメント・パラメータの説明

SEGM ステートメントでは、マクロ定義の中に示されているキーワードの代わりに、下記の省略語を使用することができます。

表 5. キーワードの省略語

キーワード	省略語
POINTER	PTR
FIRST	F
LAST	L
HERE	H



表 5. キーワードの省略語 (続き)

キーワード	省略語
KEY	K
DATA	D
VIRTUAL	V
PHYSICAL	P

### SEGM

このステートメントがセグメント定義ステートメントであることを表します。

#### NAME=

定義するセグメント・タイプの名前を指定します。

指定した名前は、このセグメントへのすべての参照で DL/I およびアプリケーション・プログラムにより使用されます。DBD 生成では、重複セグメント名は許されません。*segname1* パラメータは、1 から 8 文字の英数字値にする必要があります。各文字は、A から Z、0 から 9、\$, #、または @ でなければなりません。

制約事項: 名前の先頭文字を数値にすることはできません。

GSAM データベースは、SEGM ステートメントの NAME= パラメータをサポートしません。

#### PARENT=

定義するセグメント・タイプの物理親と論理親 (それらがある場合) の名前を指定します。

0 ルート・セグメント・タイプの場合、PARENT= キーワードを省略するか、または PARENT=0 を指定します。

#### *segname2*

従属セグメント・タイプの場合、このセグメントの物理親の名前を指定します。

#### SNGL | DBLE

定義されるセグメント・タイプの物理親のすべてのオカレンスに入れる、物理子ポインタのタイプを指定します。SNGL および DBLE は、PHDAM、PHIDAM、HDAM、HIDAM、または DEDB データベース内のセグメントに対してのみ指定することができ、物理親が階層ポインタ (PTR=HIER または HIERBWD) を指定している場合は、それらは無視されます。

SNGL は、4 バイトの物理子の最初のポインタを、定義されるセグメント・タイプの物理親のすべてのオカレンスに入れます。SNGL はデフォルトです。

DBLE は、4 バイトの物理子の最初のポインタおよび 4 バイトの物理子の最終ポインタを、定義されるセグメント・タイプの物理親のすべてのオカレンスに入れます。

#### *lpsegname*

定義するセグメント・タイプの論理親がある場合、その名前を指定します。

このオペランドは物理データベースの DBDGEN 時にものみ使用され、論理子セグメント・タイプを定義する SEGM ステートメント上で指定する必要があります。

#### VIRTUAL | PHYSICAL

論理親の連結キー (LPCK) を論理子セグメントの一部として保管するかどうかを指定します。論理子セグメントに対してのみパラメーターを指定してください。PHYSICAL を指定した場合は、LPCK は、各論理子セグメントと一緒に保管されます。VIRTUAL を指定した場合は、LPCK は論理子セグメントに保管されません。論理親が HISAM データベース内にある論理子セグメントに対しては、PHYSICAL を指定する必要があります。また、論理親の連結キーの任意の部分を使用して物理兄弟にチェーンに配列されている論理子セグメントに対しても指定する必要があります。

- PHDAM および PHIDAM
  - PHDAM および PHIDAM に対しては PHYSICAL がデフォルトです。
  - VIRTUAL を PHDAM または PHIDAM に対して指定した場合、それは無視され、PHYSICAL が使用されます。
- HDAM および HIDAM
  - HDAM および HIDAM に対しては VIRTUAL がデフォルトです。
  - HDAM および HIDAM データベース内のシンボリック・ポインターは、LPCK を使用し、PHYSICAL の指定を必要とします。

#### *dbname1*

論理親が定義されているデータベースの名前を指定します。論理親が論理子と同じデータベース内に存在する場合は、*dbname1* を省略することができます。

#### BYTES=

セグメント・タイプのデータ部分の長さ (単位はバイト) を、符号なしの 10 進整数で指定します。このパラメーターは必須です。論理子であるセグメントの場合、PARENT パラメーター内で VIRTUAL または PHYSICAL のいずれかが指定されているか、あるいはそれがデフォルトであると、この長さには論理親の連結キーが含まれます。

#### 固定長セグメントの場合の *maxbytes* および *minbytes*

固定長セグメントの場合、*maxbytes* パラメーターは、セグメントのデータ部分に使用するストレージ量を指定します。*minbytes* パラメーターは、固定長セグメント (固定長圧縮セグメントを含む) には指定できません。セグメント・タイプに指定する最大長は、使用する記憶装置の最大レコード長から接頭部またはレコード・オーバーヘッドを引いた値を超えてはなりません。

VSAM の場合の最大レコード長は 30713 バイトですが、テープの場合の最大長は 32760 バイトです。*maxbytes* に指定できる最小の長さは、セグメント・タイプに定義するすべてのフィールドが入る大きさでなければなりません。セグメントが論理子セグメント・タイプの場合、その長さは、論理親の連結キーを入れるのに十分な大きさにする必要があります。

MSDB の場合、*maxbytes* 値は、32000 バイトを超えない固定長セグメントのデータ部分の長さを指定します。指定する値は、4 の倍数にする必要があります。

### 可変長セグメントの場合の *maxbytes* および *minbytes*

*minbytes* パラメーターを含む場合には、セグメント・タイプを可変長として定義します。*maxbytes* フィールドでは、このセグメント・タイプのおカレンスの最大長を指定します。*maxbytes* パラメーターに許される最大値と最小値は、固定長セグメントに関して説明した値と同じです。

セグメントが圧縮ルーチンによって処理される場合は、異常終了 0799 を回避するために、セグメント長を指定された最大定義より長くできるかどうかを示す制御情報を収容できるように *maxbytes* フィールドを設定します。拡張を可能にするには、*maxbytes* に 10 バイトの任意の値を追加します。

*minbytes* パラメーターでは、可変長セグメントが使用する最小ストレージ量を指定します。*minbytes* の最大値は、*maxbytes* に指定する値です。

*minbytes* の最小値は次のものでなければなりません。

- セグメント・タイプを編集/圧縮ルーチンで処理しない場合、または編集/圧縮ルーチンで処理はするがキー圧縮オプションを指定していない場合は、セグメント・タイプにシーケンス・フィールドを指定してあれば、*minbytes* にはシーケンス・フィールド全体を入れられる値を指定しなければなりません。
- キー圧縮オプションを使う編集/圧縮ルーチンで処理するセグメント・タイプ、または順序付けられていないセグメントの場合の最小値は、4 です。

HSAM、SHSAM、または SHISAM データベースのセグメントは可変長にすることができないため、これらのデータベースでは *minbytes* パラメーターは無効です。

高速機能 DEDB では、セグメントは、2 バイト・フィールド (この 2 バイトの長さフィールドを含むセグメントの長さを定義) で始まり、FIELD ステートメントで指定するユーザー・データが後に続きます。*minbytes* の値には、4 (最小値) から *maxbytes* (最大値) を指定できますが、*minbytes* 値は、このセグメントのシーケンス・フィールドを入れるのに十分な大きさでなければなりません (つまり、 $minbytes \geq START - 1 + SEGM$  ステートメントに続くシーケンス・フィールドの BYTES)。例えば、シーケンス・フィールドの長さが 20 バイトで  $START = 7$  のセグメントでは、最小 *minbyte* 値は 26 です。該当する任意の DL/I 呼び出しでは、実際のセグメント長は、シーケンス・フィールドを含んだ長さ  $maxbytes$  の値の間になります。*maxbytes* の値は、制御インターバル・サイズ - 120 を超えてはなりません。

### TYPE=

DEDB 従属セグメントのタイプを記述します。ルート・セグメントに指定してはなりません。

### SEQ

セグメントが順次従属セグメント・タイプであることを指定します。1 つの DEDB につき 1 つの順次従属セグメントしか許されていません。指定する場合には、これが最初の従属セグメント・タイプでなければなりません。

## DIR

セグメントが直接従属セグメント・タイプであることを指定します。DIR はデフォルトです。

## FREQ=

このセグメントが物理親の各オカレンスごとに何回現れる可能性があるかを示す見積回数を指定します。値は、1 から 16777215 までの整数で、符号なしの 10 進数でなければなりません。このセグメントがルート・セグメントの場合、「frequency」は、定義されるデータベース内に現れるデータベース・レコードの最大数の見積もりです。従属セグメントに適用される場合の FREQ= パラメーターの値は、データベースの各データ・セット・グループの論理レコード長および物理ストレージ・ブロック・サイズを判別するのに使用されます。

DBDGEN ユーティリティが、推奨論理レコード長を計算する過程で、アセンブラ・エラー・メッセージの IF0110 ARITHMETIC OVERFLOW または IEV103 MULTIPLICATION OVERFLOW が出される可能性があります。このエラーが、HSAM、SHSAM、または HISAM DBD の生成中に発生する場合は、論理レコード長および物理ブロック・サイズの判別が必要と考えられます。

FREQ= は、高速機能 DEDB または MSDB データベース内のセグメントには無効です。

## INPOS|NOINPOS

ISRT 上の次のツイン・データをキャプチャーするオプション・パラメーター。デフォルトは NOINPOS です。挿入されたセグメントの後に続くツインのツイン・データは、INPOS が指定されていて、以下の条件が該当する場合にキャプチャーされます。

- 非固有セグメントの ISRT が作成されている。
- HERE の ISRT 規則が使用されている。

新規セグメントが唯一のツイン・インスタンスであるか、ツイン・チェーン内の最後のツイン・インスタンスである場合、ツイン・データは存在しません。

## POINTER=

定義するセグメント・タイプのオカレンスの接頭部に、ポインター・フィールドを予約することを指定します。これらのフィールドは、このセグメントを隣接した親セグメントおよび兄弟セグメントに関連付けるために使用されます。

POINTER= パラメーターは、基本的には HDAM、HIDAM、PHDAM、および PHIDAM データベースの場合に使用されます。さらに、HDAM または HIDAM データベースのセグメント・タイプと論理関係がある HISAM データベースに定義されるセグメント・タイプに使用できます。

重要: HSAM または SHSAM データベースにセグメント・タイプを定義する場合には、POINTER= パラメーターを省略する必要があります。定義するセグメント・タイプが HISAM データベースの中のもので、論理関係を持っていない場合は、POINTER= パラメーターを省略してください。

次のリストでは、キーワード・オプションの一部の一般属性について説明しています。

- 選択したキーワード・オプションは、任意の順序で指定することができますが、コンマで区切る必要があります。
- キーワード・オプションは、1 回のみ指定することができます。

- キーワードはすべてオプションです。
- 各行から 1 つのキーワード・オプションが選択できます。
- キーワード・オプションまたはその省略語を選択できます。

表 6. *POINTER*= キーワードと省略語

キーワード・オプション	省略語
HIER	H
HIERBWD	HB
TWIN	T
TWINBWD	TB
NOTWIN	NT
LTWIN	LT
LTWINBWD	LTB
PAIRED	
LPARNT	LP
CTR	C

*POINTER*= パラメーターのキーワード・オプションの意味は次のとおりです。

#### **HIER [H]**

定義するセグメント・タイプのおカレンスの接頭部に、4 バイトの階層順方向ポインター・フィールドを予約します。HALDB では HIER はサポートされません。

#### **HIERBWD [HB]**

定義するセグメント・タイプのおカレンスの接頭部に、4 バイトの階層順方向ポインター・フィールドと 4 バイトの階層逆方向ポインター・フィールドを予約します。逆方向階層ポインターを用いると、削除のパフォーマンスが上がります。HALDB では HIERBWD はサポートされません。

#### **TWIN [T]**

定義するセグメントの接頭部に、4 バイトの物理兄弟順方向ポインター・フィールドを予約します。

#### **TWINBWD [TB]**

定義するセグメントの接頭部に、4 バイトの物理兄弟順方向ポインター・フィールドと 4 バイトの逆方向物理兄弟ポインター・フィールドを予約します。逆方向物理兄弟ポインターを使用すると、削除パフォーマンスが向上します。

推奨事項: このオプションは、HIDAM および PHIDAM データベースのルート・セグメントに使用するようお勧めします。

#### **NOTWIN [NT]**

定義するセグメント・タイプのおカレンスの接頭部に、物理兄弟順方向ポインターのスペースを予約しないようにします。

NOTWIN は、次の場合に従属セグメント・タイプに対して指定できます。

- 物理親に階層ポインターが指定されていない。

- 物理親セグメント・タイプのおカレンスの物理子として保管されている従属セグメント・タイプのおカレンスが 1 つ以下である。

さらに、NOTWIN は、HDAM および PHIDAM データベースのルート・セグメント・タイプにも指定できますが、それはランダム化モジュールが同義語 (同じブロックおよびアンカー・ポイントを持っている異なる値のキー) を生成しない場合に限られます。

NOTWIN が従属セグメント・タイプについて指定されている場合に、従属セグメントの 2 番目のおカレンスを、ある与えられた物理親セグメントの物理子としてロードまたは挿入しようとする、次のようになります。

- 初期ロード時に 2 番目のおカレンスを挿入しようとする、LB 状況コードが戻されます。
- 初期ロード後に 2 番目のおカレンスを挿入しようとする、II 状況コードが戻されます。

同義語をロードまたは挿入する試みはリジェクトされて、LB 状況コードまたは II 状況コードが出されます。

#### **LTWIN [LT]**

実論理子を定義する場合に限り、仮想対の論理関係に使用します。これは、定義する論理子セグメント・タイプのおカレンスの接頭部の中に、4 バイトの論理兄弟順方向ポインター・フィールドを予約します。このパラメーターを指定できるのは、定義するセグメント・タイプが論理子であって、HDAM または HIDAM データベースに定義する場合のみです。PAIRED を指定した場合、LTWIN パラメーターは無効です。HALDB では LTWIN はサポートされません。

#### **LTWINBWD [LTB]**

実論理子を定義する場合に限り、仮想対の論理関係に使用します。これは、定義する論理子セグメント・タイプのおカレンスの接頭部に、4 バイトの物理兄弟順方向ポインター・フィールドと 4 バイトの逆方向物理兄弟ポインター・フィールドを予約します。このパラメーターを指定できるのは、定義するセグメントが論理子であって、HDAM または HIDAM データベースに定義する場合のみです。PAIRED を指定した場合、LTWIN パラメーターは無効です。HALDB では LTWINBWD はサポートされません。

LTWIN ではなく LTWINBWD を使用すると、論理子セグメントを削除するときのパフォーマンスが上がります。

#### **LPARNT [LP]**

このパラメーターを指定できるのは、定義されるセグメント・タイプが論理子であり、しかも論理親が HDAM、HIDAM、PHDAM、または PHIDAM データベース内に存在する場合に限られます。論理親が HISAM データベース内にある場合は、このパラメーターを省略し、定義するセグメントの PARENT= パラメーターに PHYSICAL を指定してください。

HDAM、HIDAM、および HISAM データベースの場合、LPARNT は、定義するセグメント・タイプのおカレンスの接頭部に 4 バイトの論理親ポインター・フィールドを予約します。

PHDAM および PHIDAM データベースの場合、LPARNT は、定義するセグメント・タイプのおカレンスの接頭部に 28 バイトの拡張ポインター・セットを予約します。

### CTR [C]

定義するセグメント・タイプのおカレンスの接頭部に、4 バイトのカウンター・フィールドを予約します。カウンターが必要になるのは、HISAM、HDAM、または HIDAM データベースの中の論理親セグメントが、論理子ポインターで自分に接続されていない論理子セグメントを持っている場合です。ユーザーがこのパラメーターを指定しなくても、カウンターは、必要とされるすべてのセグメントの中に、DBD 生成時に自動的に入れられます。しかし、後で DBD 生成を行うのを避けるために、ユーザーは将来のカウンターの必要性を予想して、このパラメーターを使用してセグメント・タイプのおカレンスの接頭部の中にカウンター・フィールドを確保しておくことができます。HALDB では CTR はサポートされません。

### PAIRED

このセグメントが両方向論理関係を持つことを示します。このパラメーターは、以下のタイプに指定します。

- 仮想論理子セグメント・タイプ
- 両方向論理関係にある両方の物理対の論理子セグメント・タイプ

PAIRED を指定した場合、LTWIN および LTWINBWD パラメーターは無効になります。

### POINTER= パラメーターのデフォルト値

HIDAM または HDAM の DBD における POINTER= パラメーターのデフォルト・オプションは、次のとおりです。

PTR=(TWIN,LTWIN,LPARNT)

### LTWIN

SEGM ステートメントの PARENT= パラメーターで論理親の名前 (lpsegname) を指定した場合のデフォルトです。

### LPARNT

SEGM ステートメントの PARENT= パラメーターで VIRTUAL を選択した場合のデフォルトです。

INDEX、HISAM、HSAM、または SHSAM の DBD では、POINTER= パラメーターのデフォルト・オプションは、ポインター・フィールドなしです。

POINTER= パラメーターを SEGM ステートメントで明示的に指定すると、セグメントは指定されたポインターと、IMS が正しい操作のために必要とするポインターを持ちます。例えば、LTWIN および LPARNT ポインターが必要に応じて作成されます。デフォルトが使用されるのは、パラメーターをまったく指定しなかった場合のみです。以下の表は、さまざまなタイプの DBD 生成について、POINTER= パラメーターの各パラメーターの使用法を示したものです。

表 7. POINTER= パラメーターの使用法 (論理関係なし)

目的	キーワー ド・パラメ ーター	論理セグメント GSAM MSDB DEDB	セグメント定義					INDEX PSINDEX
			HSAM SHSAM SHISAM	HISAM	HDAM HIDAM	PHDAM PHIDAM	データベース・タイプの中に入れる物理セグメント	
階層内の次のセグメントへのポインター	HIER	INVALID	VALID	IGN	VALID	IGN	IGN	
階層内の次のセグメントと前のセグメントへのポインター	HIERBWD	INVALID	INVALID	IGN	VALID	IGN	IGN	
物理兄弟の次のオカレンスへのポインター	TWIN	INVALID	INVALID	IGN	VALID	VALID	IGN	
物理兄弟の次のオカレンスと前のオカレンスへのポインター	TWINBWD	INVALID	INVALID	IGN	VALID	VALID	IGN	
接頭部のカウンター・フィールド	CTR	INVALID	INVALID	VALID	VALID	IGN	IGN	
論理兄弟の次のオカレンスへのポインター	LTWIN	INVALID	INVALID	IGN	VALID <sup>1</sup>	IGN	IGN	
論理兄弟の次のオカレンスと前のオカレンスへのポインター	LTWINBWD	INVALID	INVALID	IGN	VALID <sup>1</sup>	IGN	IGN	
論理親セグメントへのポインター	LPARNT	INVALID	INVALID	VALID <sup>2</sup>	VALID <sup>3</sup>	VALID <sup>3</sup>	IGN	
HS-HS または HS-HD または HD-HD の論理関係	PAIRED	INVALID	INVALID	VALID <sup>4</sup>	VALID <sup>5</sup>	VALID <sup>5</sup>	IGN	

キー:

- INVALID - このパラメーターを指定することはできません。
- IGN - このパラメーターを指定しても、無視されます。
- VALID - このパラメーターは有効であり、以下の注に示されているとおりに使用されます。



表 7. POINTER= パラメーターの使用法 (論理関係なし) (続き)

目的	キーワー ド・パラメ ーター	論理セグメント GSAM MSDB DEDB	セグメント定義 データベース・タイプの中に入れる物理セグメント				
			HSAM SHSAM SHISAM	HISAM	HDAM HIDAM	PHDAM PHIDAM	INDEX PSINDEX

注:

1. 定義する論理子セグメントが論理関係を持つときに使用します。HDAM、HIDAM、PHDAM、または PHIDAM 内のセグメントであって、論理親が直接アドレス (論理子ポインター) によって論理子と関連付けられている場合は、これを指定してください。
2. HDAM、HIDAM、PHDAM、または PHIDAM データベース内に論理親が定義されていて、HISAM データベース内に論理子セグメントを定義するときを使用することができます。
3. HDAM、HIDAM、PHDAM、または PHIDAM データベース内に論理親が定義されていて、HDAM、HIDAM、PHDAM、または PHIDAM データベース内に論理子セグメントを定義するときを使用することができます。
4. HDAM、HIDAM、PHDAM、または PHIDAM データベース内に論理親が定義されていて、論理関係が両方向で、HISAM データベース内に論理子セグメントを定義するときを使用することができます。
5. 両方とも物理的に存在する 2 つの論理子セグメントで、または仮想論理子用の SEGM ステートメントで、両方向論理関係を定義するときを使用します。

**RULES=**

定義するセグメント・タイプのおカレンスの挿入、削除、および置換に使用される規則を指定します。

**path type values**

セグメントの挿入、削除、または置換のときに使用するパス・タイプを指定します。

最初の欄はセグメントの挿入に適用され、2 番目の欄はセグメントの削除に適用され、3 番目の欄はセグメントの置換に適用されます。3 つの欄はそれぞれ、同じ文字または異なる文字を含むことができます。これらのパラメーターは、論理子セグメント、およびそれらの物理親セグメントと論理親セグメントに指定します。論理関係を持たないすべてのセグメント・タイプについては、これらを省略する必要があります。値の P は物理パスを、L は論理パスを、V は仮想パスを、B は両方向仮想パスを表します。

**FIRST | LAST | HERE**

この SEGM ステートメントが定義するセグメント・タイプの新しいおカレンスが、物理データベースに挿入される場所を指定します (物理兄弟順序を確立します)。この値は、シーケンス・フィールドを持たないセグメント、またはシーケンス・フィールドが固有でないセグメントを処理する場合にのみ使用します。固有のシーケンス・フィールドが定義されているセグメント・タイプに指定された場合、この値は無視されます。

HDAM および PHDAM のルートを除いて、FIRST、LAST、または HERE の規則は、データベースの初期ロードには適用されず、セグメントはロード・モードに示された順序でロードされます。初期ロードまたは HD 再ロードで、固有のシーケンス・フィールドが HDAM のルートに関して定義されていない場合は、FIRST、LAST、または HERE の挿入規則によりルートがチェーニングされる順序が決まります。したがって、HDAM また

は PHDAM データベースの再ロードでは、HERE または FIRST が使用されると、順序付けられていないルートが反転されます。

DEDB セグメント以外は、LAST がデフォルトです。

高速機能順次従属セグメント処理の場合は、FIRST の挿入規則が常に使用され、これを変更することはできません。直接従属セグメント処理の場合は、FIRST、LAST、または HERE を指定できます。デフォルトは HERE です。

#### **FIRST**

シーケンス・フィールドが定義されていないセグメントの場合には、既存のすべての物理兄弟の前に新しいオカレンスが挿入されます。固有でないシーケンス・フィールドが定義されているセグメントの場合には、同じシーケンス・フィールド値を持つ既存のすべての物理兄弟の前に新しいオカレンスが挿入されます。

#### **LAST**

シーケンス・フィールドが定義されていないセグメントの場合には、既存のすべての物理兄弟の後ろに新しいオカレンスが挿入されます。固有でないシーケンス・フィールドが定義されているセグメントの場合には、同じシーケンス・フィールド値を持つ既存のすべての物理兄弟の後ろに新しいオカレンスが挿入されます。

#### **HERE**

シーケンス・フィールドのないセグメントの場合には、位置が確立されている物理兄弟の直前に新しいオカレンスが挿入されます。挿入するセグメントの物理兄弟に位置が確立されていなければ、既存のすべての物理兄弟の前に新しいオカレンスが挿入されます。固有でないシーケンス・フィールドが定義されているセグメントの場合には、同じシーケンス・フィールド値を持ち、位置が確立されている物理兄弟の直前に新しいオカレンスが挿入されます。同じシーケンス・フィールド値を持っている物理兄弟に位置が確立されていなければ、同じシーケンス・フィールド値を持つすべての物理兄弟の前に新しいオカレンスが挿入されます。挿入位置は、直前の DL/I 呼び出しで確立された位置により異なります。

物理パスに対して出される挿入呼び出しでは、指定した挿入規則よりコマンド・コードの L (last) が優先されるため、新しいオカレンスは LAST の挿入規則に従って挿入されます。

#### **DSGROUP=**

PHDAM および PHIDAM データベースについて、複数データ・セット・グループを指定します。形式は DSGROUP=c です。ここで、c は A から J の文字です。これによって、PHDAM および PHIDAM データベースを最大 10 個のデータ・セット・グループに分割できます。すべてのセグメントで、デフォルトは A (1 区分当たり 1 個のデータ・セット) です。ルート・セグメントに指定する場合、DSGROUP=A でなければなりません。

制約事項: A から J のシーケンスにギャップがあってはなりません。例えば、DSGROUP=C を SEGM ステートメントで指定した場合、DSGROUP=B を指定した SEGM ステートメントが少なくとも 1 個必要で、各 HALDB 区画は A、B、および C データ・セットを持つこととなります。


## SOURCE=

次の 2 つの目的で使用します。

- 定義する仮想論理子セグメント・タイプで表される実論理子セグメント・タイプを識別するため。
- 論理データベース内に定義されるセグメント・タイプで表される、物理データベース内のセグメント・タイプ (複数も可) を識別するため。

制約事項: PHDAM および PHIDAM データベースでは物理対しかサポートしないため、SOURCE キーワードは使用できません。

仮想論理子を定義するステートメントは、次のようになります。

```
►►SOURCE=((segname, , dbname))◄◄
```

### *segname*

実論理子の名前を指定します。



### DATA

*segname* のキーとデータの両方の部分を、セグメントの組み立てに使用することを示します。このパラメーターは必須です。

### *dbname*

実論理子が入っている物理データベースの名前を指定します。

論理データベースのセグメント・タイプを定義するステートメントは、次のようになります。

```
►►SOURCE=--((segname, , dbname), --((segname, , dbname)--)◄◄
```

### **(*segname*, KEY | DATA, *dbname*)**

最初のオカレンスは、論理セグメントとして定義されている物理データベース内のセグメントを指すか、またはこの論理データベースの連結セグメント・タイプの最初の部分として使用する物理データベース内の論理子セグメント・タイプを指します。

### *segname*

物理データベース内の論理子セグメント・タイプを指します。

### KEY

*segname* に指定されているセグメントのキー部分を、キー・フィードバック域に入れることを指定します。*segname* で表されている論理セグメント・タイプを処理するための呼び出しを発行するときは、このセグメントをユーザーの入出力域に入れてはなりません。

### DATA

*segname* で指定したセグメントのキー部分をキー・フィードバック域に入れる必要のあることと、*segname* で表されている論理セグメント・タイプを処理するための呼び出しを発行するときは、このセグメントをユーザーの入出力域に入れる必要があることを指定します。

### **dbname**

*segname* を含む物理データベースの名前を指定します。( *segname*, *KEY|DATA*, *dbname* ) の 2 番目のオカレンスは、この論理データベース内の連結セグメントの目標親部分に使用される、物理データベース内の論理親セグメント・タイプまたは物理親セグメント・タイプを指し示します。この 2 番目のオカレンスの各パラメーターの説明は、最初のオカレンスの説明と同じです。

( *segname*, *KEY | DATA*, *dbname* ) の最初のオカレンスが仮想論理子を指している場合、2 番目のオカレンス (指定する場合) は、実論理子の物理親を指していなければなりません。

ソース・セグメントを使用して、連結セグメントを表すときは、検索呼び出しの際にこの 2 つのセグメントのいずれを (または両方を) ユーザーの入出力域に入れるかを、*KEY* と *DATA* のパラメーターで指定します。*DATA* を指定すると、セグメントはユーザーの入出力域に入られます。*KEY* を指定すると、セグメントはユーザーの入出力域には入れられず、シーケンス・フィールド・キー (存在している場合) が PCB のキー・フィードバック域に入られます。連結セグメントのキーは、論理子がどのパスからアクセスされるかによって、論理子のキー、つまり物理兄弟シーケンス・フィールドか論理兄弟シーケンス・フィールドのいずれかになります。*KEY* および *DATA* パラメーターは、検索タイプの呼び出しのみに適用されます。

挿入呼び出しでは、ユーザーの入出力域には常に論理子セグメントと、挿入規則が物理でない限り、論理親セグメントが入っていなければなりません。*KEY* がセグメントについて指定されている場合でも、参照されたセグメントを含んでいる論理データベースに対して呼び出しが出された場合には、そのセグメントを含んでいるデータベースを IMS が使用できなければなりません。*SOURCE=* セグメント指定の最初のオカレンスが論理子を指している場合は、連結セグメントの目標親を指す 2 番目のオカレンスも指定する必要があります。明示的に指定されていない場合は、デフォルトによりブロックの作成時に *KEY* パラメーターで組み込まれます。

論理 DBD 生成で定義するセグメントは、1 つ以上の物理 DBD 生成で前もって定義されているセグメントの物理定義を使わなければなりません。

*SEGM* ステートメントで *INDEX* データ・セット内のセグメントを定義する場合は、*SOURCE=* パラメーターは無効です。

### **SSPTR=**

サブセット・ポインターの数を指定します。0 から 8 の値を指定できます。0 を指定するか、あるいは *SSPTR* を指定しない場合は、サブセット・ポインターを使用しないこととなります。

### **EXIT=**

データ・キャプチャー出口ルーチンを使用することを指定します。単一の *SEGM* ステートメントに複数の出口ルーチン名を指定できます。出口ルーチンごとに異なるデータ・オプションを選択できます。パラメーター内にリストされた出口ルーチンの順序により、呼び出される出口ルーチンの順序が決まります。

*SEGM* ステートメントで指定する場合、*EXIT=* パラメーターは DBD での指定を変更したり、このパラメーターを特定のセグメントに限定することができます。*EXIT=* パラメーターは、指定した物理データベース内の特定のセグメント

に適用されます。しかし、論理子セグメントに適用する場合には、出口ルーチンは仮想論理子ではなく、実論理子に指定する必要があります。この出口ルーチンがサポートする物理データベースは、次のとおりです。

- HDAM
- HIDAM
- PHDAM
- PHIDAM
- HISAM
- SHISAM
- DEDB

サポートされるデータベース編成またはサポートされるセグメント・タイプに出口ルーチンを指定しなければ、DBDGEN は失敗します。

DBD ステートメントにも EXIT= パラメーターを指定することができます。

#### **exit\_name**

データを処理する出口ルーチンの名前を指定します。このパラメーターは必須です。この名前は、標準命名規則に従う必要があります。最大 8 文字の英数字を使用できます。出口ルーチン名の代わりにアスタリスク (\*) を指定して、ロギングのみを行いたいことを示すことができます。これを指定した場合には、ロギング・パラメーターのデフォルトは LOG です。出口ルーチン名を指定すると、ロギング・パラメーターは NOLOG になります。

次に示すオペランドはオプションです。

#### **NONE**

DBD ステートメントで指定された出口ルーチンを無効にします。DBD の出口名がその特定セグメントに適用されないようにするには、SEGM ステートメントでこれを指定する必要があります。

EXIT=NONE は、仮想論理子に対して DBD で指定した出口を明示的に無効にします。

#### **LOG | NOLOG**

データ・キャプチャーの制御ブロックとデータを IMS システム・ログに書き込むかどうかを指定します。

#### **LOG**

データ・キャプチャーの制御ブロックとデータを IMS システム・ログに書き込むことを要求します。

#### **NOLOG**

データ・キャプチャーの制御ブロックとデータを IMS システム・ログに書き込まないことを示します。

#### **KEY | NOKEY**

物理連結キーを出口ルーチンに渡すかどうかを指定します。

#### **KEY**

出口ルーチンに物理連結キーを渡すことを指定します。このキーは、アプリケーションによって更新される物理セグメントを識別します。

KEY がデフォルトです。

## **NOKEY**

出口ルーチンには物理連結キーが必要でないことを指定します。

## **DATA | NODATA**

物理セグメント・データを更新のためにデータ・キャプチャー出口ルーチンに渡すかどうかを指定します。

### **DATA**

物理セグメント・データを、更新のためにデータ・キャプチャー出口ルーチンに渡します。DATA が指定されていて、セグメント編集/圧縮出口ルーチンも使用されている場合には、渡されるデータは拡張データです。

DATA がデフォルトです。

### **NODATA**

出口ルーチンがセグメント・データを必要としない場合に指定します。NODATA は、物理セグメント・データを保管することにより生じるオーバーヘッドを回避する場合に使用します。

## **NOPATH | PATH**

出口ルーチンが、物理ルートの階層パスにあるセグメントからのデータを必要とするかどうかを指定します。

### **NOPATH**

出口ルーチンが、物理ルートの階層パスにあるセグメントからのデータを必要としないことを示します。NOPATH は、パス・データの検索に必要な処理時間を回避するための効率的な方法です。

NOPATH はデフォルトです。

### **PATH**

物理ルートの階層パスにあるセグメントのデータを出口ルーチンに渡して、セグメントを更新する必要がある場合に、これを指定します。アプリケーションが、挿入、置換、または削除の目的で複数のセグメントを別々にアクセスできるようにするには、PATH を使用します。

Db2 for z/OS 1 次キーを構成するためにパスのセグメントからの情報が必要なときに、PATH オプションを使用できます。Db2 for z/OS 1 次キーは、従属セグメント更新の伝搬要求で使用されます。通常、この種のセグメント情報が必要なのは、親セグメントがキー情報を含んでおり、親セグメントに収納できない追加データを従属セグメントが含んでいる場合です。

PATH は、追加処理が必要な場合にも使用できます。例えば、D コマンド・コードを呼び出さなかった場合などに、1 つの呼び出しで複数のセグメントにアクセスしないということがあります。この場合、アプリケーションが別の呼び出しで各セグメントにアクセスする際、追加処理が必要になります。

## **DLET | NODLET**

DLET 呼び出しで X'99' ログ・レコードが書き込まれるかどうかを指定します。

### **DLET**

DLET 呼び出しで X'99' ログ・レコードが書き込まれます。

DLET はデフォルトです。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **NODLET**

DLET 呼び出しで X'99' ログ・レコードが書き込まれません。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **BEFORE | NOBEFORE**

REPL 呼び出しで、変更前データが X'99' ログ・レコードに組み込まれるかどうかを指定します。

##### **BEFORE**

REPL 呼び出しで、変更前データが X'99' ログ・レコードに組み込まれます。

BEFORE はデフォルトです。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

##### **NOBEFORE**

REPL 呼び出しで、変更前データが X'99' ログ・レコードに組み込まれません。

このパラメーターを SEGM ステートメントに指定すると、DBD ステートメントの指定がそれによってオーバーライドされます。

#### **CASCADE | NOCASCADE**

DL/I がこのセグメントを削除するときに、出口ルーチンが呼び出されるかどうかを指定します。

##### **CASCADE**

アプリケーションが親セグメントを削除したため、DL/I がこのセグメントを削除するときに、出口ルーチンが呼び出されることを示します。CASCADE を使用すると、定義されたセグメントのためにデータが確実に取り込まれます。

CASCADE はデフォルトです。

CASCADE パラメーターには、3 つのサブオプションがあります。これらのサブオプションは、データが出口ルーチンに渡される方法を制御します。サブオプションを指定する場合には、CASCADE パラメーターとサブオプションを小括弧で囲む必要があります。

#### **KEY | NOKEY**

物理連結キーを出口ルーチンに渡すかどうかを指定します。

##### **KEY**

物理連結キーを出口ルーチンに渡します。このキーは、カスケード削除で削除されるセグメントを識別します。

KEY がデフォルトです。

#### **NOKEY**

削除するセグメントの物理連結キーを出口ルーチンが必要としないときに指定することができます。

#### **DATA | NODATA**

セグメント・データを出口ルーチンに渡すかどうかを指定します。

##### **DATA**

カスケード削除の場合に、セグメント・データを出口ルーチンに渡します。また、DATA は削除されるセグメントを識別します (物理連結キーで識別できない場合)。

DATA がデフォルトです。

##### **NODATA**

出口ルーチンがセグメント・データを必要としない場合に指定します。NODATA により、物理セグメント・データを保管した場合に生じるかなりのストレージ要件およびパフォーマンス要件が削減されます。

#### **NOPATH | PATH**

出口ルーチンが、物理ルートの階層パスにあるセグメント・データを必要とするかどうかを指定します。

##### **NOPATH**

出口ルーチンが物理ルートの階層パスにあるセグメント・データを必要としないことを示します。NOPATH は、カスケード削除に必要な相当量のパス・データを除去するために使用します。

NOPATH はデフォルトです。

##### **PATH**

カスケード削除のために、アプリケーションが複数のセグメントを別々にアクセスするときに指定します。

#### **NOCASCADE**

DL/I がこのセグメントを削除するときに、出口ルーチンは呼び出されないことを示します。従属セグメントを持たないセグメントを削除する場合には、カスケード削除は不要です。

#### **NOSSPCMD | SSSPCMD**

高速機能のサブセット・ポインター (SSP) に関連するコマンド・コードをキャプチャーするかどうかを示すオプション・パラメーター。デフォルトはNOSSPCMD です。このオプションは、サブセット・ポインターが使用されるセグメントでのみ指定することをお勧めします。

次の表は、それぞれの DL/I 呼び出しでキャプチャーされるコマンド・コードを示しています。



表 8. DL/I 呼び出しでキャプチャーされるコマンド・コード

DL/I 呼び出し	詳細
G* (Get 呼び出し)	M、S、W、Z、R は、M、S、W、Z のうち少なくとも 1 つが同じ SSA 上で指定されている場合にキャプチャーされ、PATH が要求されている場合は PATH データとともにキャプチャーされます。
REPL	M, S, W, Z
DLET	Z
ISRT	M、S、W、Z、R は、セグメントが挿入される場合、または挿入されないセグメントの SSA に R が指定されたが PATH データが要求されている場合にキャプチャーされます。

### NOINPOS | INPOS

ISRT 呼び出しで次のツイン・データをキャプチャーすることを要求するオプション・パラメーター。デフォルトは NOINPOS です。挿入されたセグメントの後に続くツインのツイン・データは、INPOS が指定されていて、以下の条件が該当する場合にキャプチャーされます。

- 非固有セグメントの ISRT が作成されている。
- HERE の ISRT 規則が使用されている。

新規セグメントが唯一のツイン・インスタンスであるか、ツイン・チェーン内の最後のツイン・インスタンスである場合、ツイン・データはキャプチャーされません。

### NOFLD | FLD

DEDB FLD 呼び出しによって行われる更新をキャプチャーすることを要求するオプション・パラメーター。このオプションは、DEDB にのみ有効です。キャプチャーされる情報は、LOG オプションが指定されている場合、X'9904' ログ・レコードに記録されるだけです。データ・キャプチャー出口ルーチンには渡されません。

### COMPRTN=

DEDB または全機能データベースにセグメント編集/圧縮の出口ルーチンを選択します。

全機能データベースのセグメント編集/圧縮の場合

SOURCE キーワードを指定している場合は、このキーワードを指定してはなりません。DL/I COMPRTN キーワードは、

MSDB、HSAM、SHSAM、SHISAM、INDEX、および論理データベースの DBDGEN の実行時には無効です。これは、すべてのデータベースの論理子セグメントについても無効です。HISAM データベースに使用する場合は、HISAM ルート・セグメントのシーケンス・フィールド・オフセットを変更してはなりません。さらに、セグメント編集/圧縮オプションが指定されているセグメント・タイプに指定できる最小セグメント長は、4 バイトです。

要確認: セグメント編集/圧縮出口ルーチンを使用しており、しかもセグメントを可変長として定義した場合は、可変長セグメントの圧縮時には、DBD で定義

されたセグメントの最小の長さまでヌル・バイトが埋め込まれることに注意してください。最小のセグメントの長さが、圧縮長をオーバーライドします。これにより、過度に圧縮されたセグメントのロード時に、追加スペースが提供されません。

#### **routinename**

ユーザー提供の編集/圧縮出口ルーチンの名前を指定します。この名前は、1 から 8 文字の英数字値でなければならないが、IMS.SDFSRESL 内の他の名前と同じであってはならず、また、DBDNAME と同じであってなりません。

#### **DATA**

示された出口ルーチンがデータ・フィールドのみを圧縮または修正することを指定します。シーケンス・フィールドを修正してはならないだけでなく、セグメントの開始点に対してのシーケンス・フィールドの位置を変更するデータ・フィールドも修正してはなりません。圧縮ルーチンの名前が指定されていても、パラメーターが選択されていない場合は、DATA がデフォルトです。

#### **KEY**

指名されたセグメント内のどのフィールドも、出口ルーチンによって圧縮または修正が可能であることを指定します。このパラメーターは、HISAM データベースのルート・セグメントについては無効です。

#### **INIT**

セグメント出口ルーチンが初期設定と終了処理制御を必要とすることを示します。このパラメーターを指定すると、データベースのオープン後およびデータベースのクローズ後に編集/圧縮ルーチンに制御が渡されます。

#### **max**

圧縮出口ルーチン間に固定長セグメントの長さを増やすことのできる最大バイト数を指定します。1 から 32 767 バイトを指定できます。max のデフォルトは 10 です。

#### **PAD**

MAX で指定した数値は埋め込み用に使用し、MAX としては使用しないことを示します。1 から 32 767 までの数値は、挿入されたセグメントの圧縮後の長さが PAD 値よりも短いときに、そのサイズまで挿入セグメントが埋め込まれることを示します。

### **DEDB のセグメント編集/圧縮の場合**

#### **routinename**

ユーザー提供のセグメント編集/圧縮出口ルーチンの z/OS ロード・モジュール名を指定します。ルーチン名が必要です。

#### **DATA**

セグメントのユーザー・データ部分のみが圧縮されることを指定します。DATA はデフォルトです。

制約事項: KEY パラメーターは、DEDB についてはサポートされていません。KEY パラメーターを指定すると、エラー・メッセージが出され、DBDGEN は終了します。

## INIT

この指定により、データベースの最初の区域がオープンされた直後にセグメント圧縮出口ルーチンが制御を獲得し、データベースの最後の区域がクローズされる直前に制御を戻すことができます。DBDGEN で指定された値の範囲内にセグメント長がある限り、セグメントの圧縮または拡張についてのフィールド修飾の検査中にエラーは起こりません。

制約事項: COMPRTN= キーワードは、セグメントの終わりに置かれた固有キー・フィールドを含んでいる DEDB セグメントに対しては禁止されています。それらのタイプのセグメントを処理するために COMPRTN= を使用すると、DBDGEN は失敗し、メッセージ DGEN440 が発行されます。

## ENCODING=

セグメント内の文字データのエンコードを指定する、1 から 25 文字のオプション・フィールド。

ENCODING キーワードに指定する値には、以下の文字を含めることはできません。

- 単一および二重引用符
- ブランク
- より小 (<) およびより大 (>) 記号
- アンパーサンド (&)

このセグメントでは、SEGM ステートメントの ENCODING パラメーターの値は DBD ステートメントの ENCODING パラメーターの値をオーバーライドします。ENCODING パラメーターが SEGM ステートメントに指定されていない場合、デフォルト値は DBD ステートメントの ENCODING パラメーターの値か、または ENCODING が DBD ステートメントに指定されていなかった場合は、EBCDIC エンコードを指定する値 Cp1047 になります。

この値は、DFSMARSH ステートメントの ENCODING パラメーターによって個々のフィールドでオーバーライドできます。

## EXTERNALNAME=

NAME= パラメーターのオプションの別名。Java™ アプリケーション・プログラムは、セグメントを参照する場合にこの外部名を使用します。

GSAM データベースには、SEGM ステートメントの EXTERNALNAME パラメーターが必要です。

外部名は 1 から 128 文字の大文字の英数字ストリングとして指定します。外部名には下線文字を含めることができます。

SEGM ステートメントに指定する外部名は、DBD 内で固有でなければなりません。

EXTERNALNAME パラメーターのデフォルト値は NAME パラメーターの値です。

制約事項: 外部名は予約済み SQL キーワードにすることも、DFS で開始することもできません。

EXTERNALNAME パラメーターが指定されておらず、NAME パラメーターに予約済み SQL キーワードが指定されている場合、EXTERNALNAME は NAME 値に「\_TBL」を付加したうえで、その NAME 値をデフォルトの外部名として受け入れます。

#### REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### CORRECT


```
REMARKS='These remarks apply to the XYZ application'
```

#### INCORRECT

```
REMARKS='These remarks apply to the 'XYZ' application'
```


- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。


関連概念:

 GSAM データ・セット特性の指定元 (アプリケーション・プログラミング)


関連資料:

58 ページの『DATASET ステートメント』

 セグメント編集/圧縮出口ルーチン (出口ルーチン)

 IMS JDBC ドライバーにより制限されるポータブル SQL キーワード (アプリケーション・プログラミング)

関連情報:

 0799 (メッセージおよびコード)

## LCHILD ステートメント

LCHILD ステートメントは、1 つの

DEDB、HISAM、HIDAM、HDAM、PHDAM、または PHIDAM データベースでの 2 つのセグメント・タイプ間の論理関係、またはこれらのいずれか 2 つのデータベースでのセグメント・タイプ間の論理関係を定義します。

制約事項: PHIDAM データベースの 1 次索引に対して、LCHILD ステートメントを指定してはなりません。

## 論理関係

DBDGEN 入力デッキ内で、論理親セグメント・タイプを定義する SEGM ステートメントの後に、その論理親の論理子であるセグメント・タイプ (仮想論理子セグメント・タイプは除く) ごとに 1 つの LCHILD ステートメントが必要です。これらの LCHILD ステートメントは、論理親とその論理子セグメント・タイプの間関係を確立します。仮想論理子セグメント・タイプを定義する SEGM ステートメントの SOURCE= パラメーターは、論理親と仮想論理子セグメント・タイプの間同様の関係を確立します。

## HIDAM 1 次索引関係

2 つの LCHILD ステートメントを用いて、HIDAM 1 次索引データベースと HIDAM データベースのルート・セグメント・タイプとの間に必要な索引関係を確立します。

HIDAM データベースの DBD 生成においてルート・セグメント・タイプを定義する SEGM ステートメントの後に、索引データベース内の索引ポインター・セグメント・タイプを指定する LCHILD ステートメントが必要です。HIDAM 1 次索引データベースの DBD 生成において索引ポインター・セグメントを定義する SEGM ステートメントの後に、HIDAM データベース内のルート・セグメント・タイプを指定する LCHILD ステートメントが必要です。

## 副次索引関係

2 つの LCHILD ステートメントを使用して、それぞれの副次索引関係を確立します。

索引ターゲット・セグメント・タイプを定義する SEGM ステートメントの後に、その索引ターゲット・セグメント・タイプを指す索引ポインター・セグメント・タイプごとに 1 つの LCHILD ステートメントが必要です。索引ターゲット・セグメント・タイプ用の SEGM に続くそれぞれの LCHILD ステートメントは、その索引ターゲットを指す索引ポインター・セグメント・タイプを識別します。

高速機能 DBD では、単一の SEGM ステートメントの下に複数の LCHILD ステートメントを指定することがサポートされます。DEDB DBD 内には任意の数の LCHILD ステートメントを指定できます。これは、各ソース・セグメントからそれぞれ取得される同じ長さの検索フィールドがあり、これにより、単一の副次索引を指す複数の副次索引ポインター・セグメントが形成されるためです。

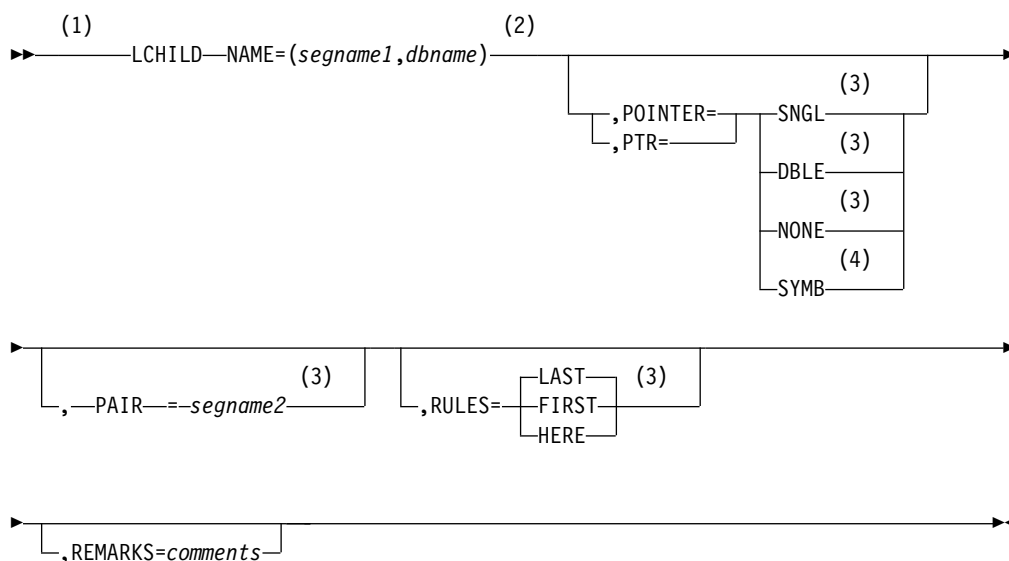
単一のソース・セグメントからの単一のターゲット・セグメントに対し、同じセグメント名を持つ複数の副次索引セグメントを定義するには、ターゲット・セグメントの SEGM ステートメントの下に 2 つ以上の LCHILD/XDFLD ステートメントを定義します。

最大 255 個の LCHILD ステートメントを、単一の DBD 生成で使用できます。LCHILD ステートメントを続けることができるのは、SEGM ステートメント、FIELD ステートメント、XDFLD ステートメント、または別の LCHILD ステートメントの後のみです。論理関係および索引関係は HSAM または SHSAM データベースに定義してはならないため、ACCESS=HSAM または ACCESS=SHSAM の場合には LCHILD ステートメントは無効です。

高速機能副次索引では、LCHILD ステートメントの PAIR および RULES オペランドはサポートされません。LCHILD ステートメントの PAIR= および RULES= パラメーターは論理関係に使用され、1 次 DEDB データベース DBD の LCHILD ステートメントでは無効なパラメーターです。

データベース・タイプごとの LCHILD ステートメントの形式を、以下の例に示します。

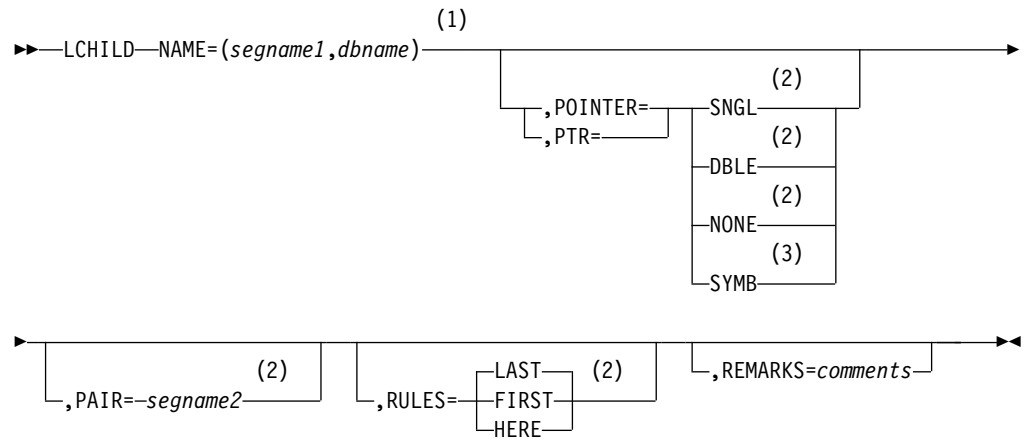
### HISAM データベースの LCHILD ステートメント



注:

- 1 HISAM 副次索引データベースまたは SHISAM 副次索引データベースに 2 つ以上のユーザー区画データベースがある場合、NAME= パラメーターに 2 つ以上のユーザー区画副次索引データベース名を指定します。
- 2 論理関係または副次索引付けに使用します。
- 3 論理関係に使用します。
- 4 物理データベースを定義する際に、索引ターゲット・セグメント・タイプにシンボリック・ポインター設定を指定する場合には、そのセグメント・タイプの副次索引にシンボリック・ポインター設定を指定してください。SYMB を副次索引のターゲット・セグメントに指定する場合は、PTR=SYMB を INDEX DBD の LCHILD ステートメントにも指定します。高速機能副次索引の場合は、LCHILD ステートメントに PTR=SYMB パラメーターを明示的に指定する必要があります。高速機能副次索引ではシンボリック・ポインターのみがサポートされますが、PTR=SNGL がデフォルトであるためです。

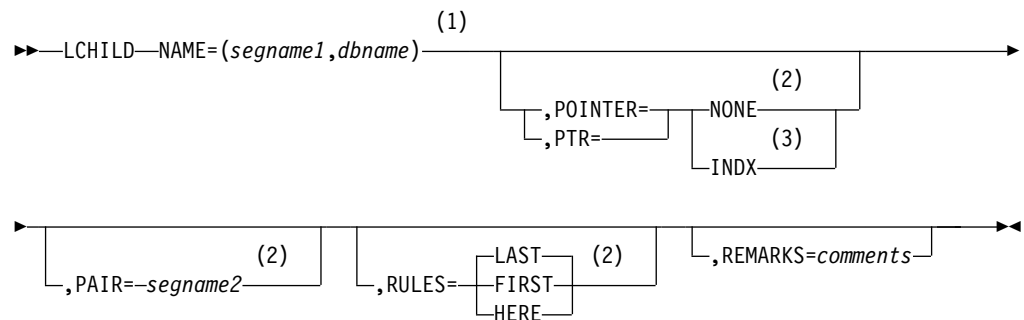
### HDAM データベースの LCHILD ステートメント



注:

- 1 論理関係または副次索引付けに使用します。
- 2 HDAM、HISAM、および HIDAM の論理関係に使用します。
- 3 物理データベースを定義する際に、索引ターゲット・セグメント・タイプにシンボリック・ポインター設定を指定する場合には、そのセグメント・タイプの副次索引にシンボリック・ポインター設定を指定してください。SYMB を副次索引のターゲット・セグメントに指定する場合は、PTR=SYMB を INDEX DBD の LCHILD ステートメントにも指定します。

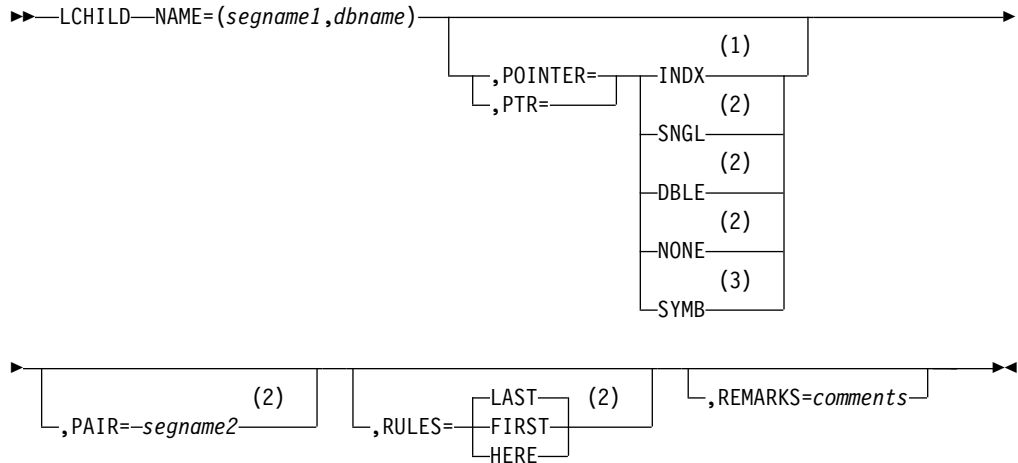
### PHDAM データベースの LCHILD ステートメント



注:

- 1 論理関係または副次索引付けに使用します。
- 2 HDAM、HISAM、および HIDAM の論理関係に使用します。
- 3 HIDAM の DBD 生成の実行時に、HIDAM 1 次索引関係を確立する LCHILD ステートメントに必要です。副次索引のターゲット・セグメントに PTR=INDX を指定する場合は、PTR は省略するか、INDEX DBD の LCHILD ステートメントに PTR=SNGL として指定する必要があります。

## HIDAM データベースの LCHILD ステートメント

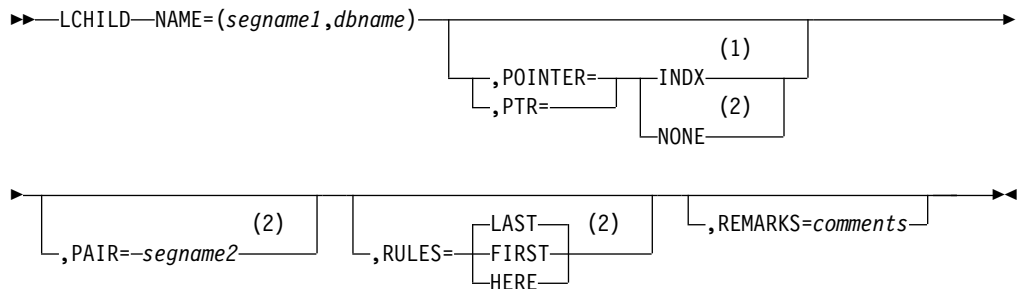


注:

- 1 HIDAM の DBD 生成の実行時に、HIDAM 1 次索引関係を確立する LCHILD ステートメントに必要です。副次索引のターゲット・セグメントに PTR=INDX を指定する場合は、PTR は省略するか、INDEX DBD の LCHILD ステートメントに PTR=SNGL として指定する必要があります。
- 2 HDAM、HISAM、および HIDAM の論理関係に使用します。
- 3 物理データベースを定義する際に、索引ターゲット・セグメント・タイプにシンボリック・ポインター設定を指定する場合には、そのセグメント・タイプの副次索引にシンボリック・ポインター設定を指定してください。SYMB を副次索引のターゲット・セグメントに指定する場合は、PTR=SYMB を INDEX DBD の LCHILD ステートメントにも指定します。

## PHIDAM データベースの LCHILD ステートメント

制約事項: PHIDAM データベースの 1 次索引に対して、LCHILD ステートメントを入力してはなりません。



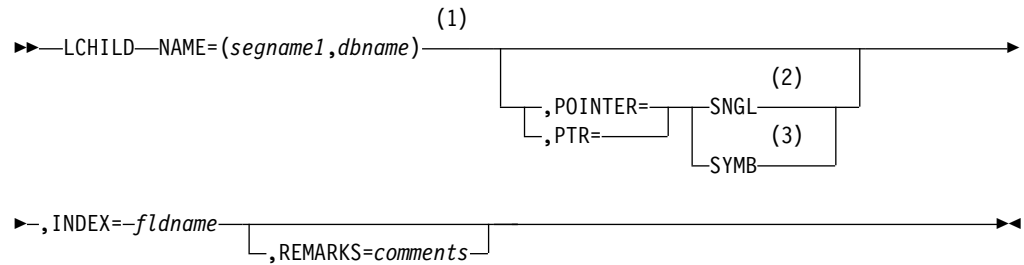
注:

- 1 HIDAM の DBD 生成の実行時に、HIDAM 1 次索引関係を確立する LCHILD ステートメントに必要です。副次索引のターゲット・セグメントに PTR=INDX を指定する場合は、PTR は省略するか、INDEX DBD の LCHILD ステートメントに PTR=SNGL として指定する必要があります。



2 HDAM、HISAM、および HIDAM の論理関係に使用します。

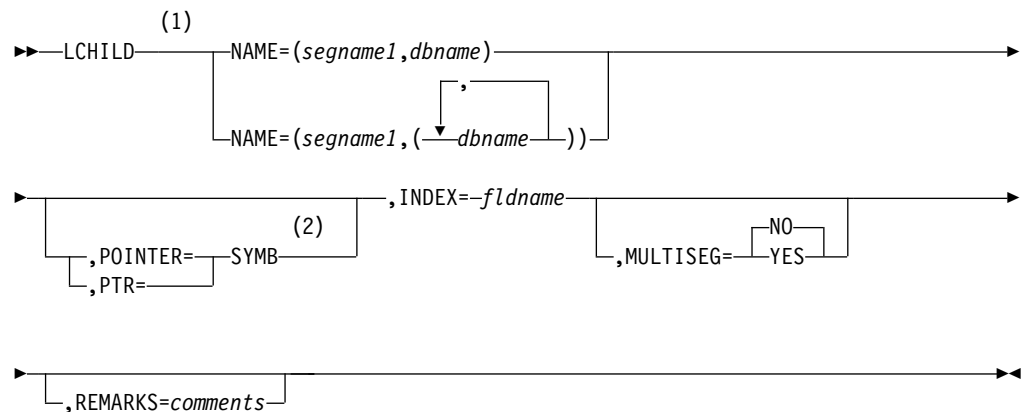
### 全機能副次索引データベースの場合の INDEX データベースの LCHILD ステートメント



注:

- 1 1 次索引付けおよび副次索引付け。
- 2 HIDAM データベースの 1 次索引に必要。
- 3 物理データベースを定義する際に、索引ターゲット・セグメント・タイプにシンボリック・ポインター設定を指定する場合には、そのセグメント・タイプの副次索引にシンボリック・ポインター設定を指定してください。SYMB を副次索引のターゲット・セグメントに指定する場合は、PTR=SYMB を INDEX DBD の LCHILD ステートメントにも指定します。

### DEDB 副次索引データベースの場合の INDEX データベースの LCHILD ステートメント



注:

- 1 1 次索引付けおよび副次索引付け。
- 2 高速機能副次索引では、シンボリック・ポインターを使用する必要があります。



## SNGL

論理関係、または直接アドレス・ポインターが作成する索引関係に使用します。SNGL は、DBDGEN 入力デックの先行 SEGM ステートメントで定義されたセグメント・タイプの各オカレンスに、論理子先頭ポインター・フィールドが確保されることを指定します。先行 SEGM で論理親を定義している場合には、このポインター・フィールドに論理子セグメント・タイプの最初のオカレンスを指す直接アドレス・ポインターが入ります。先行 SEGM で HIDAM 1 次索引データベース・セグメント・タイプを定義している場合には、このポインター・フィールドに、HIDAM データベース・ルート・セグメントを指す直接アドレス・ポインターが入ります。先行 SEGM で副次索引データベース内の索引ポインター・セグメント・タイプを定義している場合には、このポインター・フィールドに、索引ターゲット・セグメントを指す直接アドレス・ポインターが入ります。

**DBLE** 論理親セグメントに予約する 2 個の 4 バイト・ポインター・フィールド (論理子先頭および論理子最終) を指定するのに使用します。この 2 つのポインターは、論理親のもとにある論理子セグメント・タイプの最初と最後のオカレンスを指します。論理子最終ポインターは、論理子が順序付けられておらず、かつ RULES= パラメーターが LAST の場合に役立ちます。

## NONE

論理親から論理子セグメントへの論理関係をインプリメントしないか、あるいは論理子直接アドレス・ポインターでインプリメントしない場合、これを使用します。この場合は、論理親から論理子への関係が存在しないか、あるいは物理対のセグメントを使用して関係が維持されません。論理親セグメントにポインター・フィールドは予約されません。

**INDX** HIDAM データベースの LCHILD ステートメントに指定します。これは HIDAM データベースの DBD 生成の過程で、HIDAM ルート・セグメント・タイプと HIDAM 1 次索引の間の索引関係を確立するのに使用します。索引ターゲット・セグメント・タイプと副次索引の間の索引関係を確立するターゲット・データベースについての DBD の LCHILD ステートメントにも、INDX を指定できます。これらの場合には、1 次索引 DBD または副次索引 DBD の LCHILD ステートメントで PTR= パラメーターを省略するか、PTR=SNGL を指定してください。HIDAM 1 次索引の LCHILD ステートメントは、副次索引の LCHILD ステートメントの前に置く必要があります。

要件: ターゲット・データベースが HALDB の場合、DBD ステートメントの ACCESS パラメーターで PSINDEX パラメーターを使用して、索引データベースを HALDB 索引として定義する必要があります。

## SYMB

副次索引のターゲット・データベースの DBD 生成に使用します。これは、索引ターゲット・セグメントの連結キーを、直接ポインターではなく索引ポインター・セグメントに入れることを指定します。索引ターゲット・セグメント・タイプが HISAM データベースにある場合には、

SYMB を指定する必要があります。索引ターゲット・セグメントが HDAM または HIDAM データベースにある場合は、SYMB はオプションです。

さらに、INDEX DBDGEN で SYMB パラメーターを使用すると、索引ポインター・セグメントの接頭部の中に 4 バイトの直接アドレス索引ターゲット・セグメント・ポインター (これは、索引ポインターがシンボリックのときには使用されない) 用のスペースが予約されません。

#### **PAIR=**

両方向論理関係の場合にのみ *segname2* を指定します。*segname2* パラメーターは、*segname1* で指定された論理子セグメントと物理的または仮想的に対にされている論理子セグメントの名前です。*segname2* パラメーターは、1 から 8 文字の英数字値にする必要があります。

制約事項: PHDAM および PHIDAM データベースでは物理対しかサポートしないため、これらのデータベースを使用するときには仮想対に対してこのパラメーターを使用できません。

#### **INDEX=**

索引 DBD 生成の場合にのみ、LCHILD ステートメントに指定します。*fldname* パラメーターには、HIDAM データベースの 1 次索引の DBD 生成時に指定した HIDAM ルート・セグメント・タイプのシーケンス・フィールドの名前、または副次索引データベースの DBD 生成時に指定した索引ターゲット・セグメント・タイプの索引フィールド (XDFLD ステートメントにより定義) の名前を指定します。このパラメーターは、PHIDAM データベースの 1 次索引には必要ありません。

#### **RKSIZE=**

ターゲット・データベースのルート・キー・サイズを指定します。このパラメーターは、区分副次索引 (PSINDEX) データベースの場合のみ必須で、他のタイプのデータベースでは無効です。

#### **RULES=**

仮想論理子にシーケンス・フィールドが定義されていないか、固有でないシーケンス・フィールドが定義されている場合、論理関係に使用します。これらの条件下では、FIRST、LAST、または HERE の規則により、順序が制御されます。すなわち、論理関係にある実論理子のオカレンスは、論理子ポインターおよび論理兄弟ポインターにより、論理親から順序付けられます (これにより、論理兄弟順序が確立されます)。

制約事項: PHDAM および PHIDAM データベースでは物理対しかサポートしないため、これらのデータベースを使用するときには仮想対に対してこのパラメーターを使用できません。

#### **FIRST**

論理子にシーケンス・フィールドが指定されていない場合に、新しいオカレンスが論理子の既存の最初のオカレンスの前に挿入されることを示します。固有でないシーケンス・フィールドが論理子に指定されている場合、新しいオカレンスは、同じキーを持つ既存のすべてのオカレンスの前に挿入されます。

**LAST** 論理子にシーケンス・フィールドが指定されていない場合に、新しいオ

カレンスが論理子の既存の最後のオカレンスの後ろに挿入されることを示します。固有でないシーケンス・フィールドが論理子に指定されている場合には、新しいオカレンスは、同じキーを持つ既存のすべてのオカレンスの後ろに挿入されます。LAST はデフォルト・オプションです。

**HERE** 挿入は、直前の DL/I 呼び出しで確立された位置によって異なることを示します。シーケンス・フィールドが定義されていない場合には、セグメントは、直前の呼び出しで位置が確立された論理兄弟の前に挿入されます。直前の呼び出しで位置が確立されていない場合には、新しい兄弟は、既存のすべての論理兄弟の前に挿入されます。固有でないシーケンス・フィールドが定義されている場合は、同じシーケンス・フィールド値を持つ直前の呼び出しで位置が確立された論理兄弟の前にセグメントが挿入されます。同じシーケンス・フィールド値を持っている論理兄弟に位置が確立されていなければ、同じシーケンス・フィールド値を持つすべての兄弟の前にセグメントが挿入されます。

論理子の新しいオカレンスが、その物理親から挿入される場合には、論理兄弟チェーン上の論理子には前の位置は存在しません。したがって、シーケンス・フィールドが定義されていなければ、新しいオカレンスは、論理兄弟チェーン上の既存のすべてのオカレンスの前に置かれるか、あるいは、固有でないシーケンス・フィールドが定義されている場合には、同じシーケンス・フィールド値を持つ既存のすべてのオカレンスの前に置かれます。

論理パスに対して出される挿入呼び出しでは、指定した挿入規則よりコマンド・コードの L (LAST) が優先されるため、新しいオカレンスは LAST の挿入規則に従って挿入されます。

#### **MULTISEG=**

複数副次索引セグメント・グループに属する 1 組の LCHILD および XDFLD ステートメントを識別します。MULTISEG= パラメーターの有効な値は YES または NO です。NO はデフォルトです。

**YES** LCHILD/XDFLD ステートメントの組を複数副次索引セグメント・グループのメンバーとして識別します。

**NO** LCHILD/XDFLD ステートメントの組が複数副次索引セグメント・グループに属していないことを識別します。

制約事項: MULTISEG= パラメーターは、DEDB データベースの LCHILD ステートメントでのみ有効です。DEDB データベースでないデータベースに MULTISEG= を指定すると、DBDGEN ユーティリティーはエラー・メッセージを発行して終了します。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### CORRECT

```
REMARKS='These remarks apply to the XYZ application'
```

#### INCORRECT

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## FIELD ステートメント

FIELD ステートメントは、セグメント・タイプ内のフィールドを定義します。フィールドは、フィールドのセンシティブティを定義する際に PSB によって参照されるか、または DL/I 呼び出しセグメント検索指数でアプリケーション・プログラムにより参照されます。

FIELD ステートメントは、以下の条件を満たしていなければなりません。

- DBD 生成のすべてのセグメントについて、**NAME** パラメーターと XDFLD ステートメントの組み合わせを含む最大 1,000 個の FIELD ステートメントを定義できます。
- DBD 生成のすべてのセグメントについて、最大 20,000 個の XDFLD ステートメントとすべての FIELD ステートメントの組み合わせを定義できます。
- すべてのセグメント・タイプについて、最大 255 個の FIELD ステートメントと XDFLD ステートメントの組み合わせを定義できます。

HISAM、HIDAM、PHIDAM、HIDAM 1 次索引、SHISAM、DEDB、および非端末関連 MSDB の各データベースのルート・セグメント・タイプには、固有のシーケンス・フィールドを定義する必要があります。HDAM データベース内のルート・セグメント・タイプは、キー・フィールドを定義する必要はありません。キー・フィールドが定義されても、それが固有である必要はありません。

HDAM、HIDAM、PHDAM、および PHIDAM データベースで固有の副次索引を定義するために /SX を使用すると、索引ソース・セグメントの 4 バイトの RBA が索引レコードのキーの一部として組み込まれることとなります。

HISAM、HDAM、HIDAM、PHDAM、および PHIDAM データベースで固有の副次索引を定義するために /CK を使用しても、同じこととなります。PSINDEX では、/SX を指定すると、8 バイトの ILK が 4 バイトの RBA の代わりに使用されます。

PSINDEX の各項目にも、ターゲット・セグメントのルート・キーが含まれます。

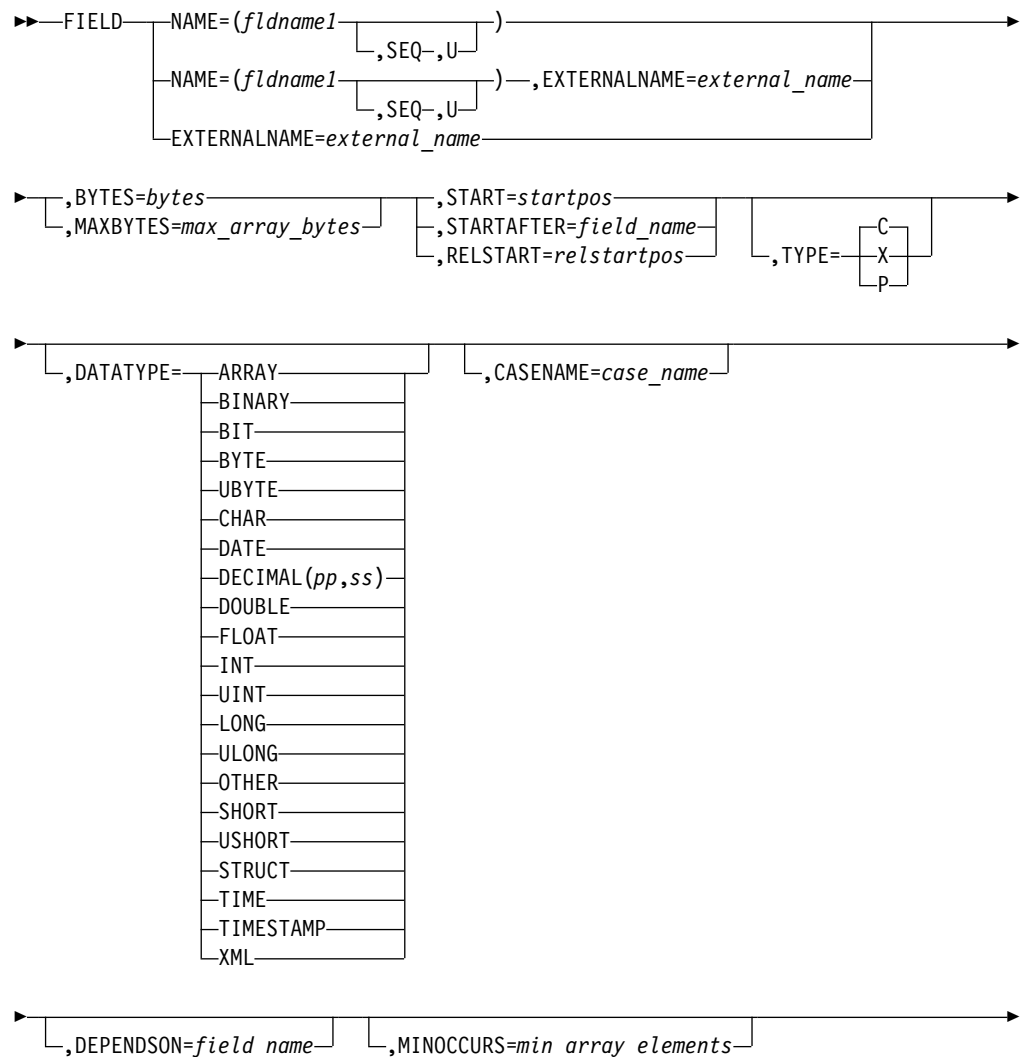
DBD 生成では、次の目的で FIELD ステートメントを使用します。

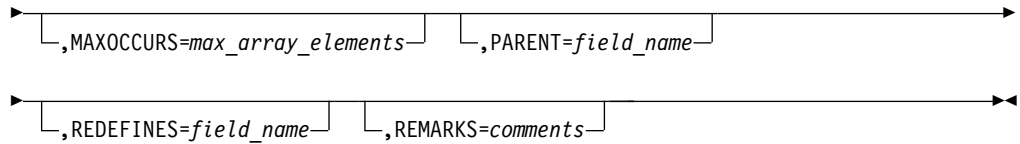
- セグメント・タイプをその物理親セグメントからアクセスするときにそのセグメント・タイプを見たものとしてそのセグメント・タイプのフィールドを定義するため。
- 仮想対の論理関係の実論理子セグメント・タイプをその論理親からアクセスするときに見たものとしてそのセグメント・タイプのフィールドを定義するため。  
FIELD ステートメントは、仮想論理子を定義する SEGM ステートメントの直後に続ける必要があります。
- 副次索引に使用するシステム関連フィールドを定義するため。

下記の構文図の後に、131 ページの『FIELD ステートメント・パラメーターの説明』が記載されています。

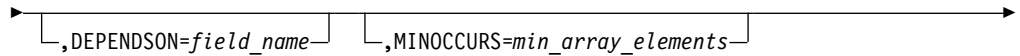
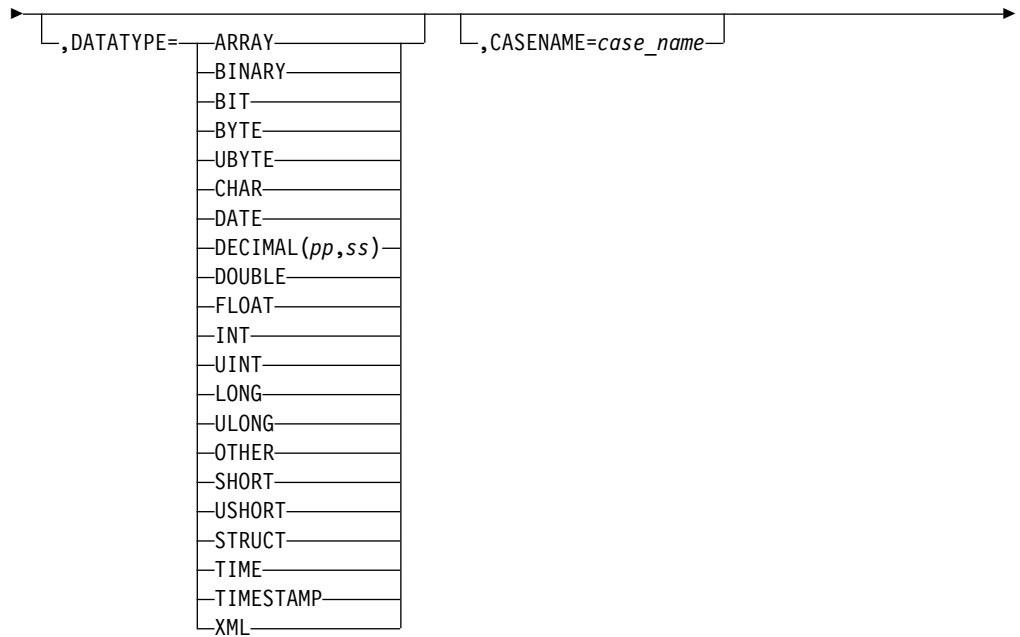
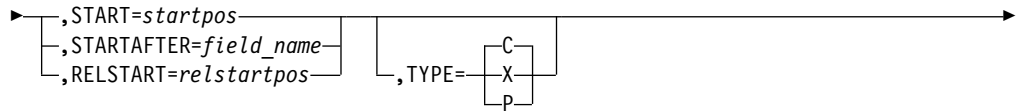
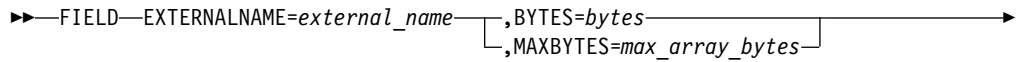
データベース・タイプごとの FIELD ステートメントの形式を、以下の構文図に示します。

### DEDB データベースの FIELD ステートメント

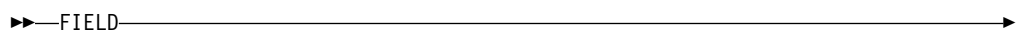




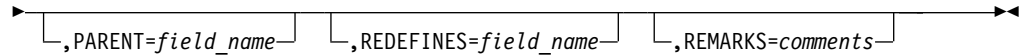
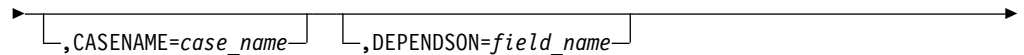
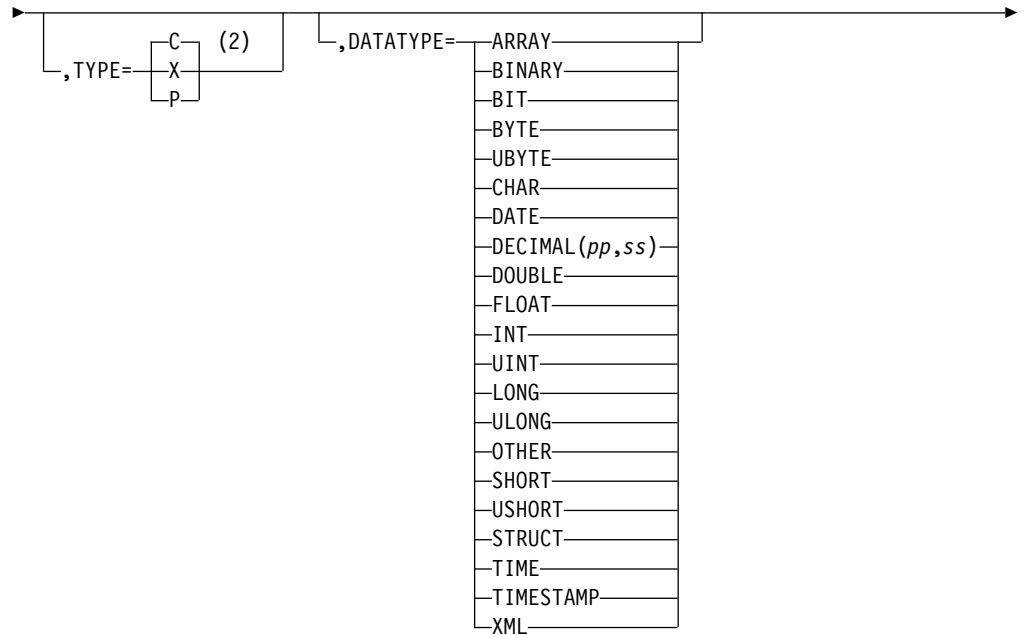
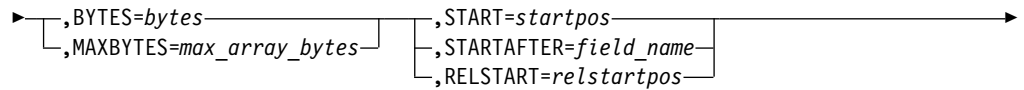
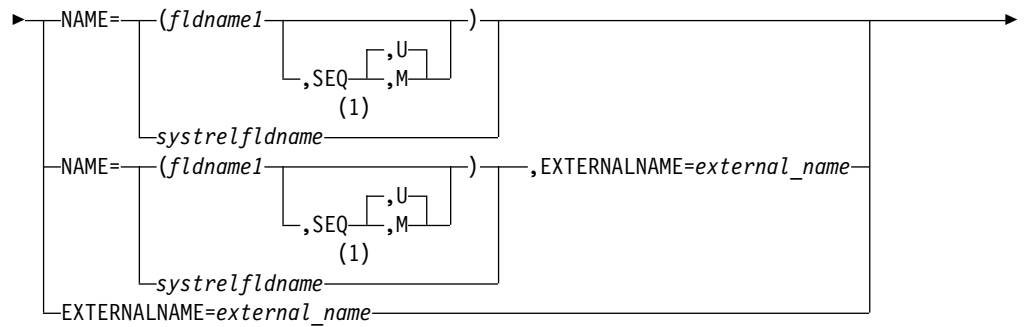
## GSAM データベースの FIELD ステートメント



## HDAM および PHDAM データベースの FIELD ステートメント





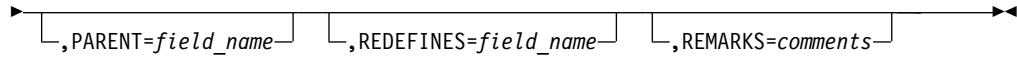
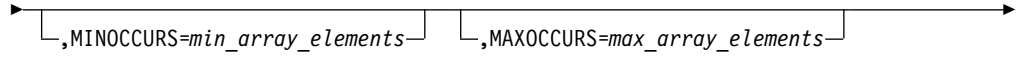
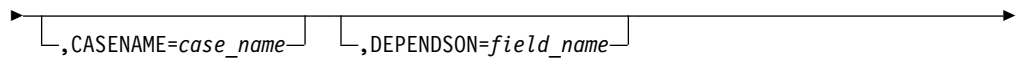
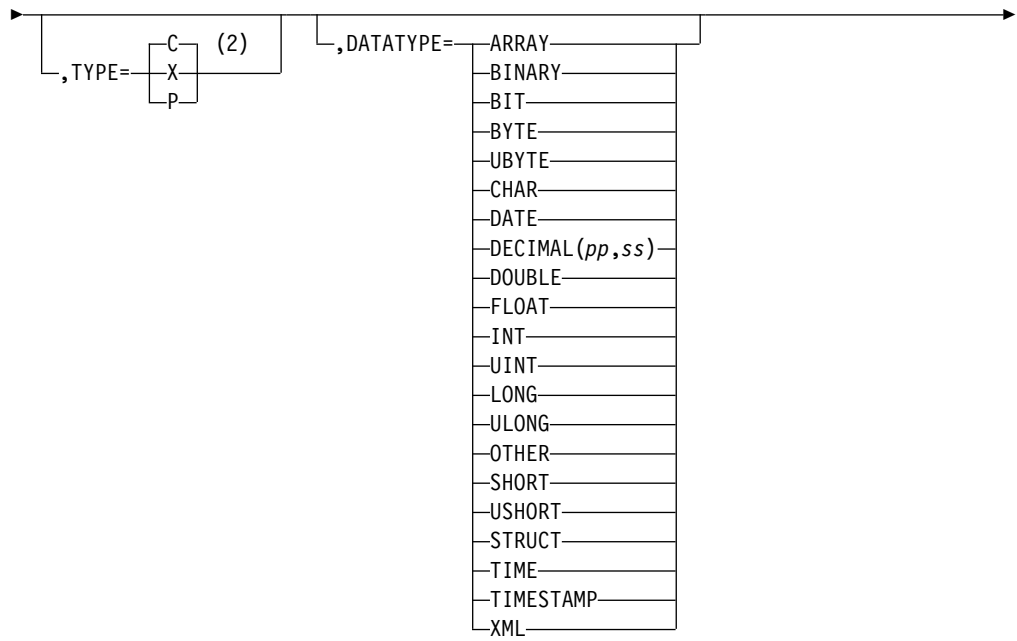
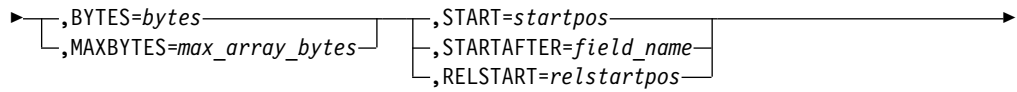
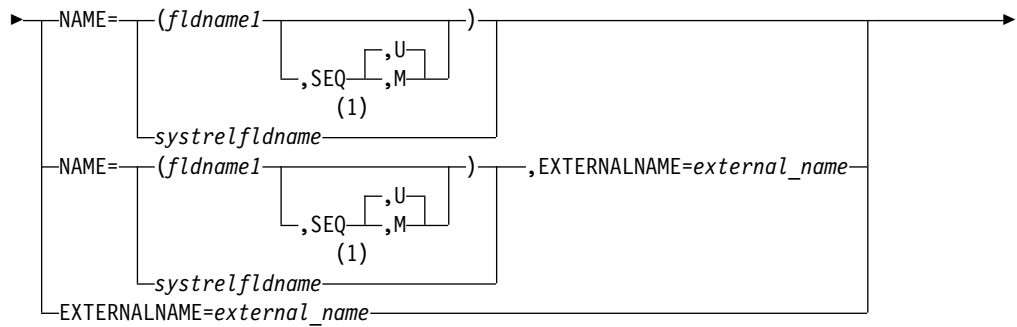


注:

- 1 副次索引として使用するシステム関連フィールド。
- 2 systrelfldname を指定したフィールドでは、TYPE= パラメーターは無視されま  
す。

## HIDAM および PHIDAM データベースの FIELD ステートメント

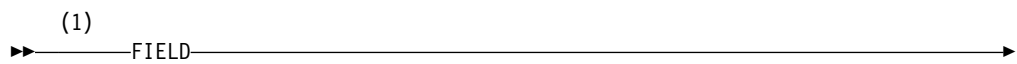


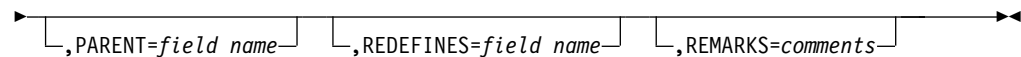
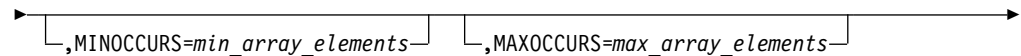
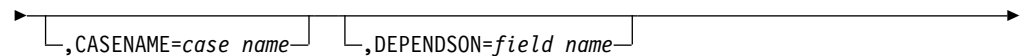
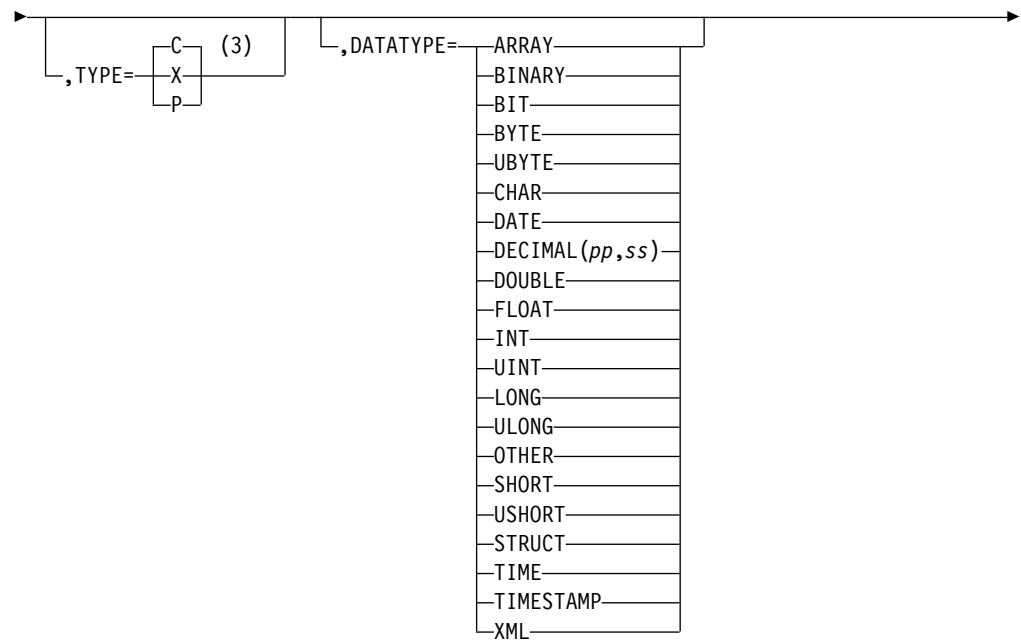
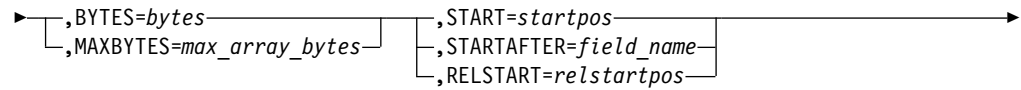
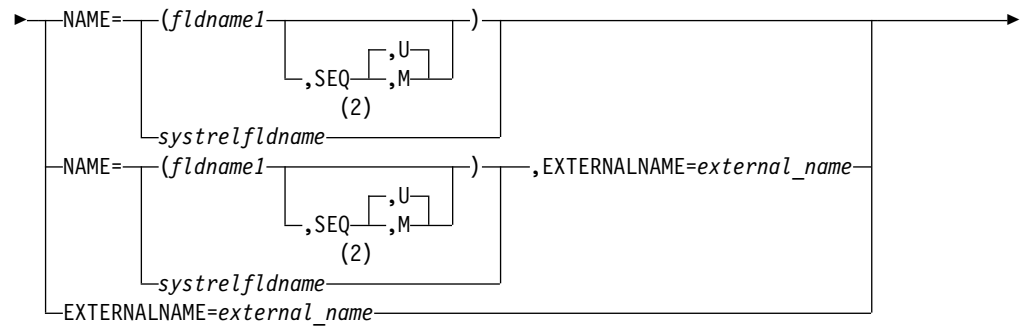


注:

- 1 副次索引として使用するシステム関連フィールド。
- 2 *systrelfldname* を指定したフィールドでは、*TYPE=* パラメーターは無視されま  
す。

### HISAM データベースの FIELD ステートメント



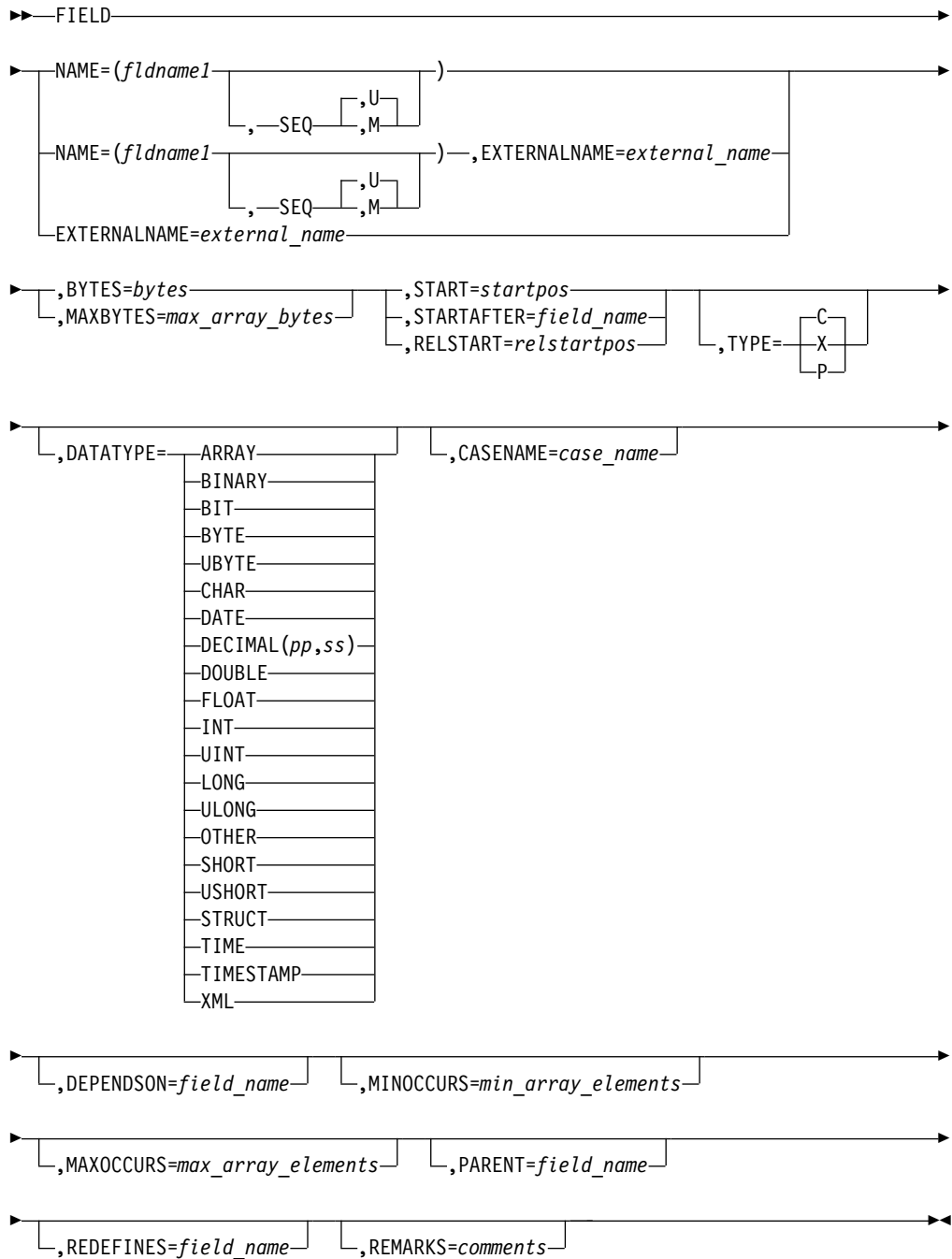


注:

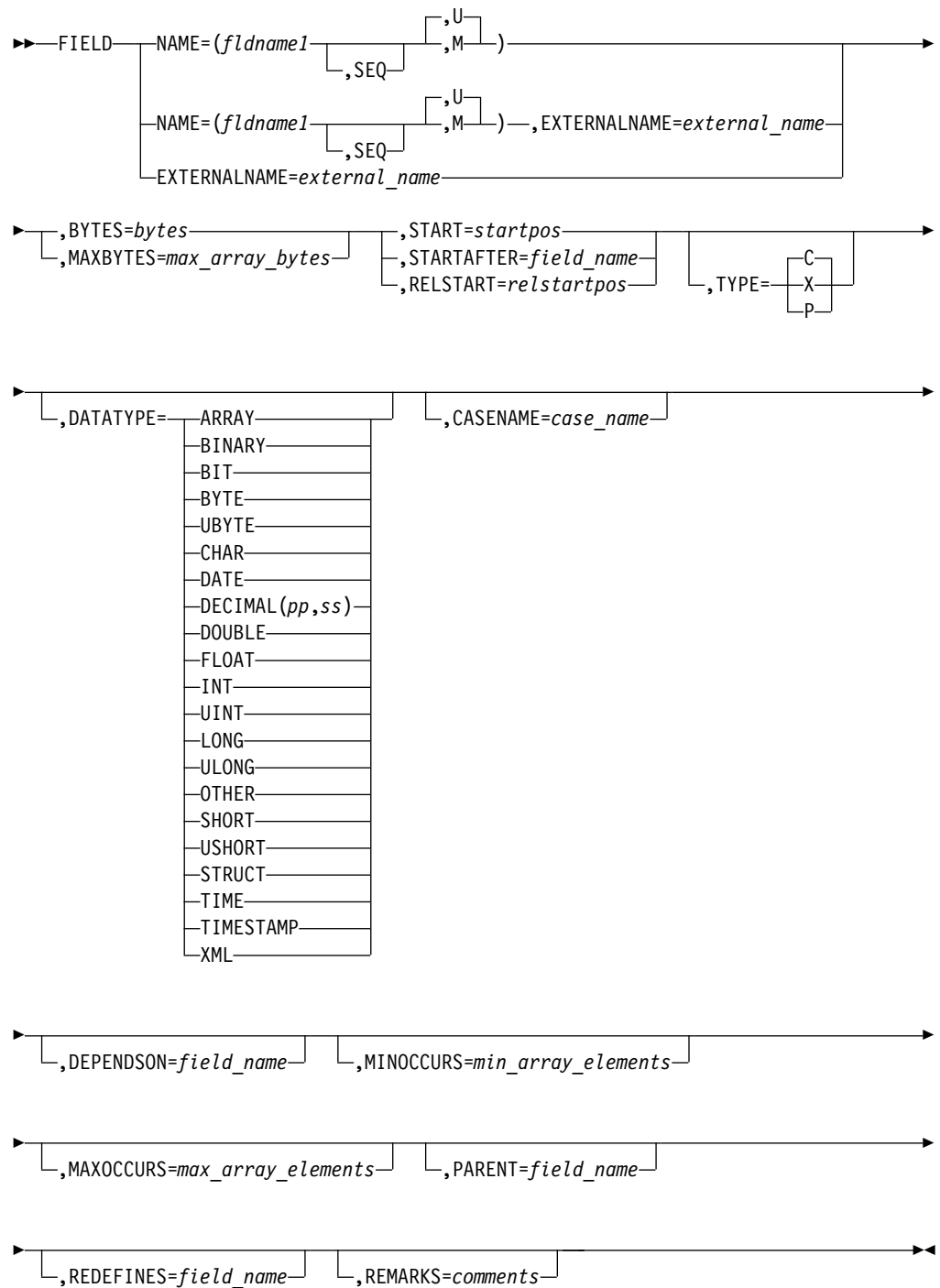
- 1 systrelfldname フィールドに対してコーディングできるのは、CK のみです。
- 2 副次索引として使用するシステム関連フィールド。

- 3 *systrelfldname* を指定したフィールドでは、TYPE= パラメーターは無視されます。

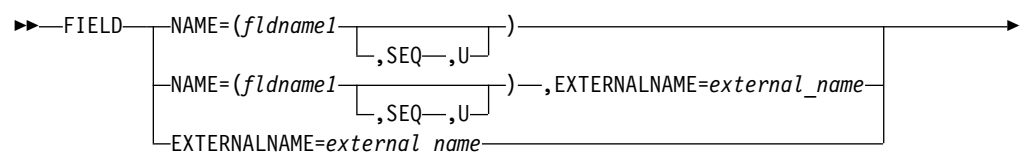
## HSAM/SHSAM データベースの FIELD ステートメント



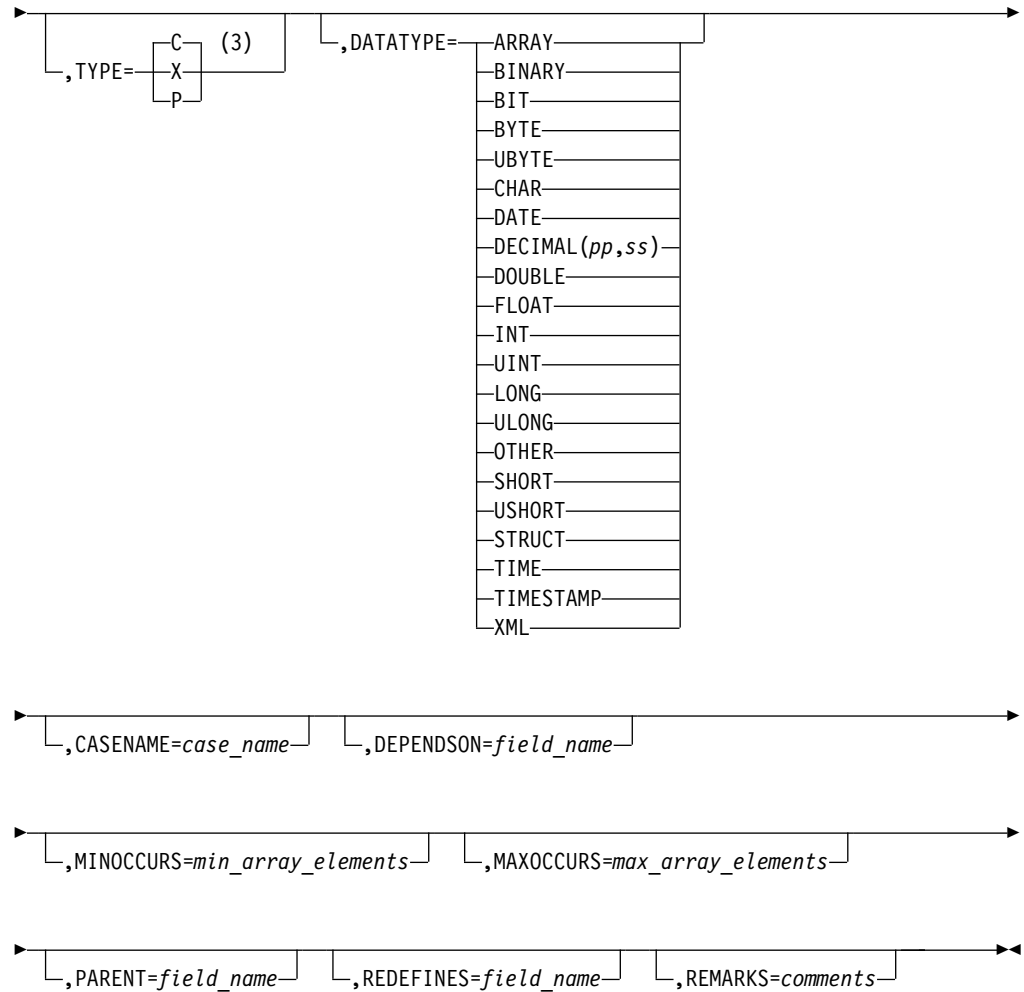
## INDEX/PSINDEX データベースの FIELD ステートメント



## MSDB データベースの FIELD ステートメント







注:

- 1 `sysrefldname` フィールドに対してコーディングできるのは、CK のみです。
- 2 副次索引として使用するシステム関連フィールド。
- 3 `sysrefldname` を指定したフィールドでは、`TYPE=` パラメーターは無視されません。

## FIELD ステートメント・パラメーターの説明

### BYTES=

定義するフィールドの長さをバイトで指定します。システム関連フィールド以外のフィールドでは、`BYTES` は、値が 255 を超えない有効な自己定義項でなければなりません。

索引ソース・セグメント・タイプの連結キーまたはその連結キーの一部をシステム関連フィールドとして定義する場合には、指定する値を 255 より大きくすることができますが、索引ソース・セグメントの連結キーの長さを超えてはなりません。

フィールドのバイト長が 255 を超える可能性があるのは、IMS によって列が検索不可として定義されている場合です。これらの列は、1 次キーとして定義することはできず、INTERNALNAME キーワードを指定することもできません。

/SX システム関連フィールドの長さは常に 4 バイトです。したがって、BYTES パラメーターは指定しても無視されます。

このフィールドが STRUCT または ARRAY によって構造または配列として定義されている場合、BYTES に指定する値は、その構造または配列に含まれるすべてのフィールドのバイト数の合計以上でなければなりません。

XML の場合、BYTES パラメーターはオプションであり、BYTES の有効値の範囲は 0 からそのセグメントの最大サイズまでです。XML の場合に BYTES パラメーターを省略すると、BYTES および MAXBYTES は許可されません。

#### **CASENAME=**

セグメント内のフィールドに代替マッピングが定義されている場合に、このフィールドが属するマップ・ケースの名前。CASENAME は、FIELD ステートメントを先行する DFSCASE ステートメント (このフィールドが属するマップ・ケースを定義するステートメント) に関連付ける場合にのみ、有効かつ必須になります。CASENAME の値は、DFSCASE ステートメントの NAME パラメーターに指定された値と一致している必要があります。

#### **DATATYPE=**

フィールドの外部データ型を指定する、3 から 9 文字のオプションの英数字フィールド。

DATATYPE パラメーターに DECIMAL が指定されている場合、デフォルトの INTERNALTYPECONVERTER は符号付き PACKEDDECIMAL です。

DATATYPE パラメーターに DATE、TIME、または TIMESTAMP が指定されている場合は、DFSMARSH ステートメントの INTERNALTYPECONVERTER パラメーターに LONG か CHAR を指定するか、または USERTYPECONVERTER を指定する必要があります。このフィールドに DFSMARSH ステートメントが含まれていない場合、INTERNALTYPECONVERTER=LONG がデフォルトになります。LONG が使用されると、その値は 1970 年 1 月 1 日からのミリ秒数として DASD に保管されます。

DATATYPE パラメーターに XML を指定すると、デフォルトの INTERNALTYPECONVERTER は XML\_CLOB です。これは、DATATYPE=XML が指定されている場合にのみ有効な値です。

DATATYPE パラメーターに STRUCT または ARRAY が指定されている場合、デフォルトの INTERNALTYPECONVERTER はそれぞれ STRUCT または ARRAY です。これらは、DATATYPE パラメーターにこのいずれかが指定されている場合にのみ、それぞれ有効な値になります。

その他の DATATYPE の値ではすべて、その値がデフォルトの INTERNALTYPECONVERTER として使用されます。

TYPE=C の場合、DATATYPE のデフォルトは CHAR になります。TYPE パラメーターの他の指定ではすべて、DATATYPE のデフォルトは BINARY になります。

有効な値は、以下のとおりです。



## ARRAY

ARRAY が指定された場合は、以下のようになります。

- NAME パラメーターはサポートされません。
- EXTERNALNAME パラメーターは必須です。
- BYTES または MAXBYTES パラメーターに指定するバイト値は、その配列に含まれるすべてのフィールドのバイト数の合計以上でなければなりません。

MSDB データベース・タイプでは ARRAY データ型はサポートされません。

## BINARY

TYPE=P または TYPE=X が指定されている場合、BINARY が DATATYPE パラメーターのデフォルト値です。

## BIT

BIT を指定する場合は、BYTES=1 も指定する必要があります。

## BYTE

BYTE を指定する場合は、BYTES=1 も指定する必要があります。

## UBYTE

UBYTE を指定する場合は、BYTES=1 も指定する必要があります。

## CHAR

TYPE=C が指定されている場合、CHAR が DATATYPE パラメーターのデフォルト値です。

## 日付

DATE が指定されている場合、INTERNALTYPECONVERTER=CHAR または USERTYPECONVERTER=*convertername* のいずれかを含む DFSMARSH ステートメントも指定していない限り、BYTES=8 を同時に指定する必要があります。

## DECIMAL(*pp,ss*)

**pp** 精度。0 より大きい 1 から 2 バイトの数値フィールド。

**ss** 位取り。0 以上の、1 から 2 バイトの数値フィールド。ss に指定する値は、pp の値より大きくてはなりません。

BYTES パラメーターには、使用されている 10 進数フォーマットと一致する値を指定する必要があります。

デフォルトの 10 進数フォーマットは符号付きパック 10 進数です。符号付きパック 10 進数フォーマットの場合に BYTES パラメーターとして必要な値を計算するには、 $length = \text{ceiling}((pp + 1) / 2)$  という式を使用します。

デフォルトの 10 進数フォーマットは、INTERNALTYPECONVERTER パラメーターを指定して変更できます。

INTERNALTYPECONVERTER=ZONEDDECIMAL と指定してゾーン 10 進フォーマットを使用する場合は、 $length = pp$  という式を使用して、BYTES パラメーターの値を計算します。

**DOUBLE**

DOUBLE を指定する場合は、BYTES=8 も指定する必要があります。

**FLOAT**

FLOAT を指定する場合は、BYTES=4 も指定する必要があります。

**INT**

INT を指定する場合は、BYTES=4 も指定する必要があります。

**UINT**

UINT を指定する場合は、BYTES=4 も指定する必要があります。

**LONG**

LONG を指定する場合は、BYTES=8 も指定する必要があります。

**ULONG**

ULONG を指定する場合は、BYTES=8 も指定する必要があります。

**OTHER**

ユーザー定義のデータ型を使用することを指定します。OTHER を指定する場合は、ユーザー提供のタイプ・コンバーターを指定する USERTYPECONVERTER パラメーターを含む DFSMARSH ステートメントも指定する必要があります。

**SHORT**

SHORT を指定する場合は、BYTES=2 も指定する必要があります。

**USHORT**

USHORT を指定する場合は、BYTES=2 も指定する必要があります。

**STRUCT**

STRUCT を指定するときは、この構造フィールドに子として動的配列フィールドが含まれる場合、SEQ パラメーターを同時に指定することはできません。動的配列フィールドは、DATATYPE=ARRAY と、さらに DEPENDSON や MAXBYTES パラメーターなどを指定して定義されます。

また、BYTES または MAXBYTES パラメーターに指定するバイト値は、その構造に含まれるすべてのフィールドのバイト数の合計以上でなければなりません。

MSDB データベース・タイプでは STRUCT データ型はサポートされません。

**TIME**

TIME が指定されている場合、INTERNALTYPECONVERTER=CHAR または USERTYPECONVERTER=*convertername* のいずれかを含む DFSMARSH ステートメントも指定していない限り、BYTES=8 を同時に指定する必要があります。

**TIMESTAMP**

TIMESTAMP が指定されている場合、INTERNALTYPECONVERTER=CHAR または USERTYPECONVERTER=*convertername* のいずれかを含む DFSMARSH ステートメントも指定していない限り、BYTES=8 を同時に指定する必要があります。

## XML

制約事項: NAME パラメーターが指定されている場合、DATATYPE=XML はサポートされません。

## DEPENDSON

動的配列内のエレメントの数を定義するフィールドの名前を指定します。参照されるフィールドの FIELD ステートメントは、DEPENDSON パラメーターを指定する FIELD ステートメントの前に配置する必要があります。指定する名前は、参照されるフィールドの定義内の EXTERNALNAME パラメーターの値 (明示的に定義された値であるか、デフォルトで受け入れた値であるかに関係なく) であることが必要です。

DEPENDSON パラメーターは、ARRAY も指定されている場合にのみ有効です。MINOCCURS の値と MAXOCCURS の値が異なる場合、DEPENDSON は必須です。

DEPENDSON パラメーターによって参照されるフィールドは、以下のいずれかの DATATYPE 値を指定して定義する必要があります。

- INT
- SHORT
- LONG
- (pp) または (pp,ss) が指定された DECIMAL。ss は 0 または 00。

MSDB データベース・タイプでは DEPENDSON パラメーターはサポートされません。

## EXTERNALNAME=

NAME= パラメーターのオプションの別名。Java アプリケーション・プログラムは、フィールドを参照する場合にこの外部名を使用します。外部名は IMS カタログ内にのみ保管され、定義するデータベースには保管されません。

EXTERNALNAME パラメーターは、NAME パラメーターが指定されていない場合、または GSAM データベースにこのフィールドが定義されている場合にのみ必須です。NAME パラメーターが指定されていない場合、このフィールドを検索できません。

外部名は 1 から 128 文字の大文字の英数字ストリングとして指定します。外部名には下線文字を含めることができます。

外部名は、セグメント内で固有でなければなりません。

EXTERNALNAME パラメーターのデフォルト値は NAME パラメーターの値です。

制約事項: 外部名は予約済み SQL キーワードにすることも、DFS で開始することもできません。

EXTERNALNAME が指定されておらず、NAME パラメーターに予約済み SQL キーワードが指定されている場合、EXTERNALNAME は NAME 値に「\_COL」を付加したうえで、その NAME 値をデフォルトの外部名として受け入れます。

IMS Universal ドライバーによって制限されている予約済み SQL キーワードのリストについては、IMS JDBC ドライバーにより制限されるポータブル SQL キーワード (アプリケーション・プログラミング)

#### M または U

このトピックで後に出てくる『U または M』の項目を参照してください。

#### MINOCCURS=

DECIMAL データ・タイプの場合、デフォルトの INTERNAL TYPECONVERTER は符号付き PACKEDDECIMAL です。

#### MAXOCCURS=

ARRAY の場合のみに、ARRAY 内のエレメントの最大数を指定する必須の数値。MAXOCCURS は必ず MINOCCURS 以上とし、ゼロであってはなりません。

#### MAXBYTES=

フィールド・インスタンスのバイト長が動的配列内のエレメント数に応じて異なる可能性がある場合に、フィールドの最大サイズをバイト単位で指定します。MAXBYTES と BYTES は同時には指定できません。

MAXBYTES の値は、このフィールドの下にネストされている、すべてのフィールドのバイト値の合計として可能性のある最大値以上でなければなりません。

MAXBYTES パラメーターが必須かつ有効であるのは、以下の場合のみです。

- フィールドが動的配列として定義されている場合。フィールドが動的配列であるのは、配列内のエレメント数とそのフィールドのインスタンスごとに異なる可能性がある場合です。動的配列の定義内で、**DEPENDSON** パラメーターがセグメント定義内の別のフィールドを参照し、それにより、動的配列の特定のインスタンスの配列エレメントの数が定義されます。
- 動的配列として定義済みのネストされたフィールドを含む、静的配列または構造として定義されたフィールドの場合。

MSDB データベース・タイプでは **MAXBYTES** パラメーターはサポートされません。

#### NAME=*fldname1*

セグメント・タイプ内のこのフィールドの名前を指定します。アプリケーション・プログラムは DL/I 呼び出し SSA で、その指定した名前を参照できます。フィールド名はセグメント定義内で固有でなければなりません。*fldname1* 値は、1 から 8 文字の英数字値にする必要があります。

以下のタイプのフィールドでは **NAME** パラメーターは必須です。

- **SEQ** パラメーターを指定するキー順フィールド・タイプ
- セグメント検索索引数 (SSA) によって参照されるフィールド・タイプ
- PSB 内の SENFLD ステートメントによって参照されるフィールド・タイプ
- XDFLD ステートメントによって参照されるフィールド・タイプ

その他のフィールド・タイプでは、**EXTERNALNAME** パラメーターが指定されていれば **NAME** パラメーターは省略できます。**NAME** パラメーターを省略すると、データベースのデータ管理ブロック (DMB) 内のストレージを節約できます。ただし、このフィールドでの検索を可能にするには **NAME** パラメーターを指定する必要があります。

以下のタイプのフィールドでは **NAME** パラメーターを指定できません。

- **GSAM** データベースに定義されているフィールド。代わりに **EXTERNALNAME** パラメーターを使用してください。
- 配列として定義されているフィールド。配列として定義されているフィールドには、フィールド定義に **ARRAY** が含まれています。
- 配列エレメントとして定義されているフィールド。配列エレメントであるフィールドは、**FIELD** ステートメントの **PARENT** パラメーターに配列フィールドの名前を指定します。
- 1 つ以上のネストされた動的配列を含む構造として定義されたフィールド。構造として定義されているフィールドには、フィールド定義に **DATATYPE=STRUCT** が含まれています。
- 動的配列を含む構造内に一緒に含まれているフィールド。構造内に含まれているフィールドは、**FIELD** ステートメントの **PARENT** パラメーターに構造フィールドの名前を指定します。
- セグメント内で動的配列の後に続くフィールド。動的配列の後に続くフィールドは、**STARTAFTER** パラメーターを指定します。
- 別のフィールドの開始位置に相対させた開始位置を指定する **RELSTART** パラメーターを含むフィールド。
- **XML** によって定義されたフィールド。

#### **PARENT=**

このフィールドを含む構造または配列として定義されているフィールドの名前を指定します。参照されるフィールドは、**DATATYPE=ARRAY** または **DATATYPE=STRUCT** のいずれかを指定して定義されている必要があります。

#### **REDEFINES=**

再定義フィールドを定義する **FIELD** ステートメントの **EXTERNALNAME** パラメーターに指定されている、再定義フィールドの名前。この値は、1 から 128 文字の英数字ストリングとして指定できます。

再定義フィールドが **EXTERNALNAME** パラメーターを指定していない場合、**NAME** パラメーターの値を使用できます。再定義フィールドが **NAME** パラメーターと **EXTERNALNAME** パラメーターの両方を指定し、それぞれの値が異なる場合、**EXTERNALNAME** パラメーターの値を使用する必要があります。

**DBD** 生成の入力順序では、再定義されるフィールドの **FIELD** ステートメントは、**REDEFINES** パラメーターを指定する **FIELD** ステートメントの前に配置する必要があります。

このフィールドの長さは、再定義されるフィールドの長さ (各 **FIELD** ステートメントの **BYTES** パラメーターに指定されたもの) と同じでなければなりません。

**ARRAY** として定義されたフィールド、または **ARRAY** を含むフィールドは再定義できません。

**MSDB** データベース・タイプでは **REDEFINES** パラメーターはサポートされません。

**RELSTART=**

配列 (または状況によっては構造) のエレメントとして定義されているフィールドの開始位置を指定します。有効な値は 1 から 32767 です。

RELSTART に指定する値は、配列または構造の開始位置に相対させた、そのフィールドの開始バイト・オフセットです。例えば、配列内の最初のフィールドは、そのフィールドを含む配列がセグメントのバイト 50 で開始する場合でも、通常は RELSTART 1 を指定します。

配列フィールドを親として指定するフィールドでは、RELSTART は必須です。構造を親として指定するフィールドでは、その構造フィールドが RELSTART または STARTAFTER を使用して定義されている場合、RELSTART は必須です。

次の例では、フィールド DYNARRAY は動的配列です。フィールド STRUCT01 は構造です。フィールド FLD03 および FLD04 は両方とも STRUCT01 を親として指定しています。セグメント内で動的配列が STRUCT01 の前に配置されているため、FLD03 と FLD04 の開始オフセットは STRUCT01 の開始位置との相対によってのみ指定できます。

```
FIELD EXTERNALNAME=ARRAYNUM,DATATYPE=DECIMAL(7,0),START=1,BYTES=4
FIELD EXTERNALNAME=DYNARRAY,DATATYPE=ARRAY,START=5,MAXBYTES=100
      MINOCCURS=10,MAXOCCURS=50,DEPENDSON=ARRAYNUM
FIELD EXTERNALNAME=FLD01,RELSTART=1,BYTES=2,PARENT=DYNARRAY
FIELD EXTERNALNAME=FLD02,STARTAFTER=DYNARRAY,BYTES=10
FIELD EXTERNALNAME=STRUCT01,DATATYPE=STRUCT,STARTAFTER=FLD02,BYTES=10
FIELD EXTERNALNAME=FLD03,RELSTART=1,BYTES=5,PARENT=STRUCT01
FIELD EXTERNALNAME=FLD04,RELSTART=6,BYTES=5,PARENT=STRUCT01
```

START、STARTAFTER、および RELSTART は同時には指定できません。

**REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

**CORRECT**

```
REMARKS='These remarks apply to the XYZ application'
```

**INCORRECT**

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## SEQ

NAME のサブパラメーターである SEQ は、このフィールドがセグメント・タイプ内のシーケンス・フィールドであることを表します。キーワード SEQ を含んでいる FIELD ステートメントは、DBD 生成入力デッキの SEGM ステートメントに続く最初の FIELD ステートメントにする必要があります。

実論理子セグメントのシーケンス・フィールドが、論理親の連結キーのいずれかの部分で構成されている場合には、論理子と論理親との連結キーを論理子がストレージ内に含めることができるように、SEGM ステートメントに PHYSICAL パラメーターを指定する必要があります。

一般に、1 つのセグメントが持つことのできるシーケンス・フィールドは 1 つのみです。ただし、仮想対の両方向論理関係の場合は、次に示すように、複数の FIELD ステートメントを使用して、仮想論理子セグメント・タイプに対して論理シーケンス・フィールドを定義できます。

論理親から論理子セグメントにアクセスする際に、実論理子セグメントのフィールド (1 つ以上) 内のデータによって決まる順序で実論理子セグメントを検索する必要がある場合には、シーケンス・フィールドを仮想論理子セグメント・タイプに指定する必要があります。このシーケンス・フィールドには、論理親から見たときのこのセグメントの任意の部分 (すなわち、実論理子の物理親の連結キーとそれに続く交差データ) を入れることができます。不連続の断片の論理親セグメントからアクセスされる論理子セグメントのシーケンス・フィールドを記述しなければならない場合があるため、SEQ パラメーターが存在する FIELD ステートメントが複数あることが許されています。各ステートメントは、固有の *fldname1* パラメーターを含んでいる必要があります。

シーケンス・フィールドを SSA 中の修飾として定義することができますが、後続のすべてのシーケンス・フィールドは指定されたフィールドの一部と見なされます。したがって、SSA で名前を指定したフィールドの長さは、指定したフィールドと後続のすべてのシーケンス・フィールドを連結した長さになります。この「分散」シーケンス・フィールドが許されるのは、仮想論理子セグメントのシーケンス・フィールドを指定する場合のみです。最初のシーケンス・フィールドが SSA 内の「分散」シーケンス・フィールドに含まれていなければ、DL/I は、引数を、シーケンス・フィールド指定ではなくデータ・フィールド指定として取り扱います。DL/I は、データ・フィールド指定の評価の際に、兄弟チェーン上のすべてのセグメント・インスタンスを調べる必要があります。シーケンス・フィールド指定が評価される際には、検索は、SSA 値よりも大きなシーケンス・フィールド値に達するまで、兄弟チェーンに沿って続行されます。その地点に達すると、検索は停止します。

MSDB では、DATASET ステートメントで REL=NO (端末関連キーのない非端末関連 MSDB) を指定する場合は、キーワードの SEQ を指定しなければなりません。それ以外の場合は、このキーワードは無効です。

DEDB では、SEQ をルート・セグメントで使用する必要がありますが、これは任意の直接従属セグメントで指定できます。

### 制約事項:

- SEQ は、順次従属セグメントには指定できません。
- あるフィールドが動的配列として定義されたフィールドを含む構造として定義されている場合、そのフィールドには SEQ を指定できません。構造フィ

ールドは DATATYPE=STRUCT によって定義されます。動的配列フィールドは、DATATYPE=ARRAY と、さらに DEPENDSON や MAXBYTES パラメーターなどによって定義されます。

#### **START=**

定義するフィールドの開始位置をセグメントの先頭からの相対的なバイトで指定します。START の値は、値が 32767 を超えない数値項でなければなりません。セグメントの最初のバイトの開始位置は 1 です。可変長セグメントの場合、最初の 2 バイトはセグメントの長さを含みます。したがって、実際、ユーザー・データ・フィールドが始まるのは 3 バイトからです。フィールドをオーバーラップすることが許されています。論理子セグメントを定義する場合、セグメント・タイプの最初の  $n$  個のバイトは、論理親または物理親の連結キーです。位置 1 から開始するフィールドは、そのフィールドの全部または一部を定義します。位置  $n+1$  から開始するフィールドは、交差データで始まります。

START をシステム関連フィールドに使用すると、連結キーの一部を索引ソース・セグメント・タイプ内のフィールドとして記述できます。この方法で使用すると、START は、連結キーの関連部分の開始位置を、連結キーの先頭から相対的に指定します。この場合、連結キーの最初のバイトは、位置 1 であると見なされます。これは、値が、連結キーの長さ + 1 を超えない数値項でなければなりません。BYTES パラメーターに指定された値を引いてください。/SX システム関連フィールドの開始位置パラメーターは無視されます。

START、STARTAFTER、および RELSTART は同時には指定できません。

XML の場合、START パラメーターはオプションであり、START 0 の指定が可能です。XML の場合に START パラメーターを省略すると、START 0 がデフォルトになります。

#### **STARTAFTER=**

フィールドが動的配列の後に開始するために、そのフィールドの開始バイト・オフセットを計算できない場合に、セグメント内でこのフィールドの直前に配置されるフィールドの名前を指定します。この名前は、INTERNALNAME キーワードで指定されている名前と同じではありません。

STARTAFTER が必須かつ有効なのは、動的配列として定義されたフィールドが前のオフセットでそのフィールドの前に配置されているために、そのフィールドの開始位置を計算できない場合のみです。

動的配列があると、セグメント内の後続のフィールドの開始オフセットの計算が不可能になります。これは、動的配列のバイト長がセグメントのインスタンスごとに異なる場合があるためです。動的配列フィールドの列は、DEPENDSON および MAXBYTES パラメーターを含めることによって識別できます。

STARTAFTER パラメーターは、配列フィールドを親として定義するフィールドでは指定できません。RELSTART パラメーターを代わりに指定してください。

START、STARTAFTER、および RELSTART は同時には指定できません。

#### **sysrelfldname**

副次索引専用として使用できるシステム関連フィールドを定義します。システム関連フィールドには次の 2 つのタイプがあります。



- 先行 SEGM ステートメントで定義された索引ソース・セグメント・タイプの連結キーの全部または一部。このタイプのシステム関連フィールドの名前は、最大 8 文字にすることが可能で、/CK の 3 文字で始めなければなりません。4 番目から 8 番目の文字で、定義するフィールドを一意的に識別することができます。この名前は、セグメント・タイプに定義された他のすべてのフィールドの中で固有にする必要があります。このタイプのシステム関連フィールドを定義するのは、索引ソース・セグメントの連結キー、あるいは索引ポインター・セグメントのサブシーケンス・フィールドまたは重複データ・フィールド内の連結キーの一部を使用できるようにするためです。

次の表に示す連結キーがあるとします。

表 9. 索引ソース・セグメント・タイプの連結キーの例

ルート・キー (10 バイト)	従属キー (3 バイト)	従属キー (3 バイト)	従属キー (3 バイト)
-----------------	--------------	--------------	--------------

3 つのシステム関連フィールドが、ルート・キーのバイト 2 から 8、2 番目のキーのバイト 1 と 4 番目のキーのバイト 2 および 3 から構成されているものとすると、FIELD ステートメントでこれを指定するときは次のようになります。

```
NAME=/CK1
BYTES=7
START=2
```

```
NAME=/CK2
BYTES=1
START=11
```

```
NAME=/CK3
BYTES=2
START=18
```

次に、定義した 3 つのシステム関連フィールドを、索引ポインター・セグメントのサブシーケンスまたは重複データ・フィールドの中で使用するために、これらのシステム関連フィールドの名前を、XDFLD ステートメントのサブシーケンスまたは重複データ・フィールドのリストの中に指定することができます。

- 2 番目のタイプのシステム関連フィールドは、副次索引でシーケンス・フィールド・キーが固有になるように、索引ソース・セグメント・タイプ内に定義します。このタイプのシステム関連フィールドに指定する名前は、/SX という文字で始める必要があります。また指定する名前は最大 8 文字です。このタイプのシステム関連フィールドを索引ソース・セグメント・タイプに定義すると、IMS は固有の 4 バイト値を生成し、索引ソース・セグメントから生成された索引ポインター・セグメントのサブシーケンス・フィールドにそれを入れます。

XDFLD ステートメントでは、索引ポインター・セグメントのサブシーケンス・フィールドまたは DDATA フィールドのいずれか、またはこの両方に指定するフィールド・リストの中に、/CK フィールドを入れることができます。/SX フィールドは、索引ポインター・セグメントのサブシーケンス・フィールドに指定するフィールド・リストにしか入れることができません。

高速機能副次索引の場合、/CK フィールドのみが有効であり、/SX フィールドは無効です。

#### TYPE=

このフィールド内のデータのマスクまたは埋め込みに IMS が使用する、文字のタイプを決定します。

DATATYPE パラメーターが明示的に設定されていない場合は TYPE パラメーターが DATATYPE のデフォルト値も決定しますが、そうでない場合は、TYPE がデータの保管方法や変換方法、またはアプリケーション・プログラムへのデータの表示方法に影響を及ぼすことはありません。

例えば、フィールド・レベル・センシティブティーを使用するアプリケーション・プログラムがこのフィールドのセンシティブティーを認識できない場合、IMS は X'00' または X'40、あるいは MSDB ではハーフワード・バイナリー・データまたはフルワード・バイナリー・データによって、フィールド内のデータをマスクすることができます。

アプリケーション・プログラムがセグメント内の 1 つ以上のフィールドのセンシティブティーを認識した場合、IMS は以下のいずれかの条件が満たされるとフィールドをマスクします。

- 挿入呼び出し時に、アプリケーション・プログラムがセンシティブティーを認識できないフィールドがセグメントに含まれている。
- 可変長セグメントを既存セグメントより長いセグメントに置き換えるための呼び出し時に、セグメントの増加部分にアプリケーション・プログラムがセンシティブティーを認識できないフィールドが含まれている。
- このフィールドが入っていない可変長セグメントを検索するための呼び出し時。

英数字フィールド (TYPE=C) が物理セグメント内に部分的に存在している場合には、データはユーザーの入出力域のフィールドに移され、右側がブランクで埋められます。部分的に存在する 16 進数フィールドまたはパック 10 進数フィールドは、ユーザーに示されるときには充てん値に置き換えられます。

DL/I 呼び出しはすべて、バイトごとに 2 進数でフィールド比較を実行します。フィールド内に含まれているデータが、このパラメーターで指定されているタイプであるか確認するための検査が、IMS によって行われることはありません。ただし、定義されているフィールドがフィールド・センシティブティーで使用されているか、MSDB にある場合は、その限りではありません。

TYPE パラメーターには以下の値を指定できます。

- X** 16 進データを指定します。X を指定すると、IMS がフィールドの未使用バイトを充てんする必要がある場合、IMS は値を右寄せし、値の左側の未使用バイトを X'00' によって充てんします。例えば、5 バイト・フィールド内の 3 バイト値 X'543210' は、X'0000543210' として書き出されます。
- P** パック 10 進数データ。P を指定すると、IMS がフィールドの未使用バイトを充てんする必要がある場合、IMS は値を右寄せし、値の左側の未使用バイトを X'00' によって充てんします。例えば、5 バイト・フィールド内の 3 バイト値 X'54321C' は、X'000054321C' として書き出されます。

**C** 英数字データまたはデータのタイプの組み合わせを指定します。C を指定すると、IMS がフィールドの未使用バイトを充てんする必要がある場合、IMS は値を左寄せし、値の右側の未使用バイトを X'40' によって充てんします。例えば、5 バイト・フィールド内の 3 バイト値 X'F5F4F3' は、X'F5F4F34040' として書き出されます。

**F** 2 値フルワード・データを指定します。

セグメント定義内で算術フィールドに別のフィールドをオーバーラップさせることはできません。すなわち、TYPE=F を指定するフィールドの開始と終了のバイト・オフセットの間で、別のフィールドを開始または終了するように定義することはできません。

**H** 2 値ハーフワード・データを指定します。

セグメント定義内で算術フィールドに別のフィールドをオーバーラップさせることはできません。すなわち、TYPE=H を指定するフィールドの開始と終了のバイト・オフセットの間で、別のフィールドを開始または終了するように定義することはできません。

MSDB データベースではタイプ X、C、P、H、および F が有効ですが、以下の規則が適用されます。

- C または X のフィールドにのみ、他のフィールドを入れることができます。
- 算術の定義 (タイプは P、H、F) が 1 つだけであれば、1 つのフィールドで複数の定義をすることができます。
- 算術フィールドの一部分を入れるフィールドには、算術フィールド全体を入れなければなりません。
- シーケンス・フィールドは、TYPE=C または X でなければなりません。
- シーケンス・フィールドは他のフィールドの一部分であってはなりません。
- 算術フィールドの SSA と FSA の比較では、論理比較演算ではなく算術比較演算を使用します。
- 初期ロードおよび呼び出し処理ルーチンは、桁数および X タイプと P タイプのフィールドが有効かどうかをテストします。
- MSDB のフィールド長には、以下の規則が適用されます。
  - TYPE=X: BYTES=1 から 256
  - TYPE=P: BYTES=1 から 16
  - TYPE=C: BYTES=1 から 256
  - TYPE=F: BYTES=4
  - TYPE=H: BYTES=2
  - フィールド・タイプ F および H には、明示的長さ指定が必要です。
  - フィールドが比較演算や算術演算に関係していて、長さがフルワードまたはハーフワードである場合には、パフォーマンスの最適化のためにフィールドを適切な境界に位置合わせする必要があります。セグメントの先頭は、フルワード境界に位置合わせします。
- FIELD ステートメントの *systrelfldname* を、/SX または /CK のいずれかで定義した場合は、TYPE=パラメーターは無視され、タイプは設定されません。

## U または M

NAME のサブパラメーターである U と M は、指定するシーケンス (SEQ) フィールドのタイプを修飾します。

パラメーター U は、セグメント・タイプのオカレンスのシーケンス・フィールドでは固有値のみを使用できることを示します。従属セグメント・タイプでは、ある物理親セグメントのもとにある各オカレンスのシーケンス・フィールドは、固有値を含んでいなければなりません。

パラメーター M は、セグメント・タイプのオカレンスのシーケンス・フィールドで重複値を使用できることを示します。ルート・セグメント・タイプでは、各オカレンスのシーケンス・フィールドは、HDAM の場合を除いて、固有値を含んでいる必要があります。HDAM データベースのルート・セグメント・タイプはキー・フィールドを必要としません。キー・フィールドが定義されても、それが固有である必要はありません。

セグメントにシーケンス・フィールドが定義されていないか、固有でないシーケンス・フィールドが定義されていると、そのセグメントの SEGM または LCHILD ステートメントで指定した FIRST、LAST、または HERE の規則に従ってセグメントのオカレンスが挿入されます。

推奨事項: 論理関係に関与しているすべてのセグメントに、固有のシーケンス・フィールドを使用してください。これには、物理子セグメントと論理子セグメントだけでなく、物理親と論理親も含まれます。仮想論理子セグメント・タイプの複数のシーケンス・フィールドは、固有または非固有として一様に定義する必要があります。

端末関連キーのない非端末関連 MSDB では、ルート・シーケンス・フィールドに固有 (U) 値を指定する必要があります。DEDB では、ルート・セグメントのシーケンス・フィールドに固有 (U) 値を指定する必要があります。DEDB 内の従属セグメントには、キーは不要です。ただし、キーを定義する場合には固有にしなければなりません。

関連概念:

 生成ユーティリティに対する DBD および PSB メタデータの定義 (データベース管理)

関連資料:

150 ページの『DFSMARSH ステートメント』

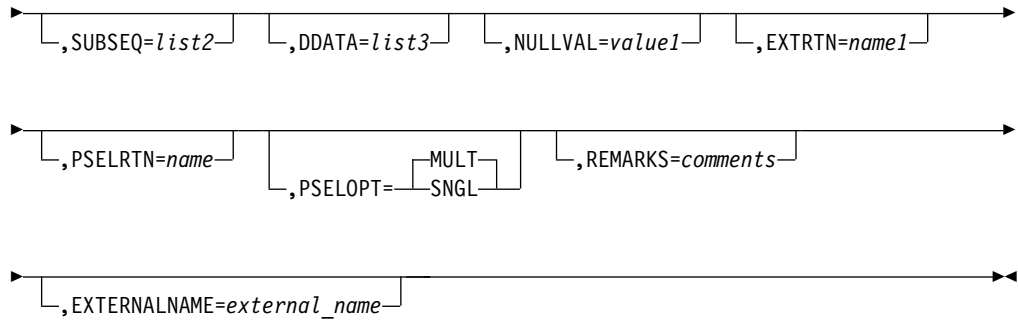
## XDFLD ステートメント

XDFLD ステートメントは、副次索引関係にのみ使用します。XDFLD ステートメントは、索引ターゲット・セグメント・タイプに関連付けられる索引フィールドの名前を定義し、索引ソース・セグメント・タイプを識別し、副次索引を作成する際に使用する索引ソース・セグメント・フィールドを識別します。

さらに、索引ポインター・セグメントの作成の抑制に関する情報が、このステートメントで与えられます。

制約事項: このステートメントは、ACCESS=INDEX、SHSAM、SHISAM、HSAM、または MSDB が指定されている DBD のセグメントの参照には使用できません。

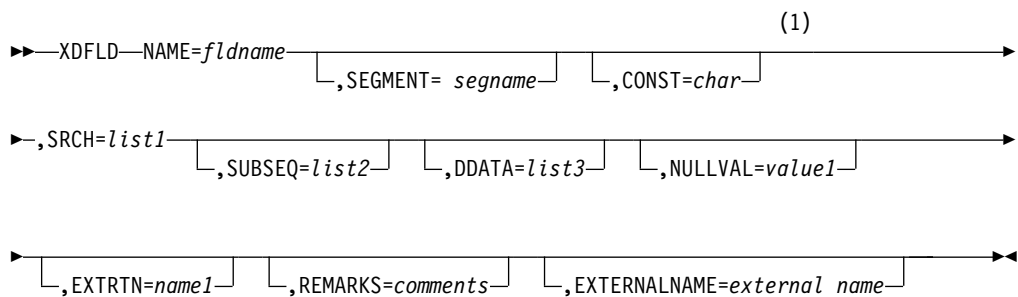




注:

- 1 単純 HISAM データベースの DBD 生成では、XDFLD ステートメントを使用しません。
- 2 SEARCH および SUBSEQUENCE フィールドの合計長は、240 バイトを超えてはなりません。

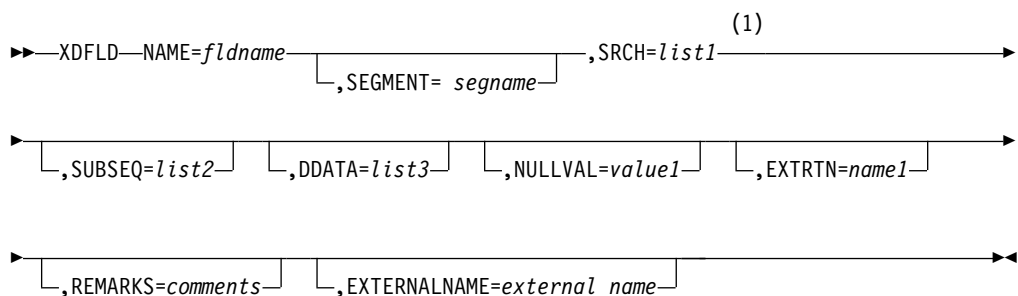
### HDAM および HIDAM データベースの XDFLD ステートメント



注:

- 1 CONSTANT、SEARCH、および SUBSEQUENCE フィールドの合計長は、240 バイトを超えてはなりません。

### PHDAM および PHIDAM データベースの XDFLD ステートメント



注:

- 1 SEARCH および SUBSEQUENCE フィールドの合計長は、240 バイトを超えてはなりません。

## XDFLD ステートメント・パラメーターの説明

### NAME=

索引ターゲット・セグメントの索引データ・フィールドの名前を指定します。指定した名前は、索引ポインター・セグメント・タイプの検索フィールドを、索引ターゲット・セグメント・タイプ内のフィールドとして実際に表します。指定した名前を使用すると、索引ポインター・セグメントの検索フィールド・キーで索引ターゲット・セグメント・タイプを呼び出す SSA を修飾できます。これにより、副次索引内のデータに基づいた 1 次または 2 次処理シーケンスで、索引ターゲット・セグメント・タイプのオカレンスにアクセスできます。fldname は、1 から 8 文字の英数字値にする必要があります。

指定した名前は、副次索引の内容に基づいて索引ターゲット・セグメントのオカレンスにアクセスするのに使用されるため、その名前は、索引ターゲット・セグメント・タイプに指定されるすべてのフィールド名の間で固有でなければなりません。

### SEGMENT=

この副次索引関係に索引ソース・セグメント・タイプを指定します。segname は、索引ターゲット・セグメント・タイプより階層の下にある、その後で定義されるセグメント・タイプの名前でなければなりません。あるいは、索引ターゲット・セグメント・タイプ自身の名前にすることができます。指定するセグメント名が、論理子セグメントであってはなりません。このパラメーターを省略すると、索引ターゲット・セグメント・タイプは索引ソース・セグメントと見なされます。

### CONST=

特定の副次索引の索引ポインター・セグメントのすべてを示すために使用する 1 つの文字を指定します。このパラメーターはオプションです。このパラメーターの目的は、複数の副次索引が同じ副次索引データベース内に存在している場合に、それぞれの副次索引に関連したすべての索引ポインター・セグメントを識別することです。char は、1 バイトの自己定義項でなければなりません。

制約事項: CONST は HALDB または DEDB データベースに対してサポートされません。

### SRCH=

副次索引の検索フィールドに使用する、索引ソース・セグメントのフィールドを指定します。list1 は、FIELD ステートメントで定義された、索引ソース・セグメント・タイプ内の 1 から 5 個のフィールド名のリストでなければなりません。複数の名前が含まれている場合には、それらをコマンドで区切って、括弧で囲む必要があります。リスト内の名前の順序は、索引ポインター・セグメントの検索フィールド内で連結されているフィールド値の順序です。関係しているフィールドの長さの合計は、索引ターゲット・セグメントの索引付きフィールドの長さになります (これは、セグメント検索指数に反映されていなければなりません)。

高速機能副次索引の場合、連結キーの一部を SRCH パラメーターの値として使用することができます。これを行うには、/CK オペランド、開始バイト、および長さを括弧で囲んで指定します。例えば、SRCH=(/CK,1,3),(/CK,5,2),(/CK,9,2) または SRCH=(/CK,1,3),FLDNM) と指定します。

### SUBSEQ=

副次索引のサブシーケンス・フィールドとして使用する索引ソース・セグメントのフィールド (ある場合) を指定します。 *list2* は、FIELD ステートメントで定義された、索引ソース・セグメント内の 1 から 5 個のフィールド名のリストでなければなりません。複数の名前が含まれている場合には、それらをコンマで区切って、括弧で囲む必要があります。リスト内の名前順序は、索引ポインター・セグメントのサブシーケンス・フィールド内で連結されているフィールド値の順序です。このパラメーターはオプションです。

高速機能副次索引の場合、連結キーの一部を SUBSEQ パラメーターの値として使用することができます。これを行うには、/CK オペランド、開始バイト、および長さを括弧で囲んで指定します。例えば、SUBSEQ=((/CK,1,3),(/CK,5,2),(/CK,9,2)) または SUBSEQ=((/CK,1,3),FLDNM) と指定します。

### DDATA=

副次索引の重複データ・フィールドに使用する索引ソース・セグメントのフィールド (ある場合) を指定します。 *list3* は、FIELD ステートメントで定義された、索引ソース・セグメント内の 1 から 5 個のフィールド名のリストでなければなりません。複数の名前が含まれている場合には、それらをコンマで区切って、括弧で囲む必要があります。リスト内の名前順序は、索引ポインター・セグメントの重複データ・フィールド内で連結されているフィールド値の順序です。このパラメーターはオプションです。

### NULLVAL=

索引ポインター・セグメントの検索フィールドで使用する索引ソース・セグメントのデータに、指定の値が入っているときに、索引ポインター・セグメントが作成されないようにします。

*value1* パラメーターは、1 バイトの自己定義項 (X'10'、C'Z'、5、または B'00101101')、または BLANK、ZERO のワードいずれかでなければなりません。BLANK は、C' ' または X'40' と同じです。ZERO は X'00' または 0 と同じですが、C'0' と同じではありません。パック 10 進値が必要な場合には、有効な数値およびゾーン数字または符号数字を持つ 16 進項 (パックの正の 3 の場合 X'3F' または負の 9 の場合 X'9D') として指定する必要があります。

SRCH= パラメーターに指定された索引ソース・セグメントの各フィールドが、すべてのバイトにこのパラメーターの値を持っている場合は、索引付けは実行されません。例えば、NULLVAL=C'9' が指定されていると、関連する索引には、値 C'9999...9' で索引付けされる項目はありません。

パック 10 進数フィールドの場合は、多少異なります。パック 10 進数フィールドでは、検索フィールドを構成する各フィールドは、別々のパック値であると見なされます。

例: 検索フィールドが 3 つの 2 バイト・パック・ソース・フィールドで構成されている場合に NULLVAL=X'9F' が指定されているとすると、X'9F' を含む索引項目はすべて抑制されるため、検索フィールド値 X'999F999F999F' を持つ索引項目は存在しないことになります。

さらに、同じ NULLVAL=X'9F' であっても、検索フィールドが 6 バイト・フィールド 1 つで、値が X'9999999999F' であるときには、索引付けは行われません。



検査される符号の形式は、指定された形式のみです。

例: X'9C' を指定すると、X'9F' 抑制は行われません。

NULLVAL= と EXTRTN= の両方のオペランドを指定するときには、そのいずれでも抑制されない限り、セグメントの索引付けが実行されます。

#### **EXTRTN=**

選択した索引ポインター・セグメントの作成を抑制するときには使用する、ユーザー提供の索引保守出口ルーチンの名前を指定します。パラメーター (name1) には、索引付きデータベースで行われる変更のために、DL/I が索引項目の挿入、削除、または置換を行おうとするときにいつも制御を受け取るユーザー提供のルーチンの名前を指示する必要があります。この出口ルーチンは、影響を受ける索引ソース・セグメントを検査し、索引ポインター・セグメントを生成すべきかどうかを判別できます。

NULLVAL= と EXTRTN= の両方のオペランドを指定するときには、そのいずれでも抑制されない限り、セグメントの索引付けが実行されます。

#### **EXTERNALNAME=**

**NAME=** パラメーターのオプションの別名になります。Java アプリケーション・プログラムは、フィールドを参照する場合にこの外部名を使用します。外部名は IMS カタログ内にものみ保管され、データベースには保管されません。

**EXTERNALNAME** パラメーターはオプションです。 **NAME** パラメーターは指定する必要があります。

外部名は 1 から 26 文字の大文字の英数字ストリングとして指定します。さらに次の文字も使用できます。

- \_ (下線)
- \$
- #
- @

外部名は、セグメント内で固有でなければなりません。

**EXTERNALNAME** パラメーターのデフォルト値は **NAME** パラメーターの値です。

注: 外部名は予約済み SQL キーワードにすることも、DFS で開始することもできません。

**EXTERNALNAME** が指定されておらず、**NAME** パラメーターに予約済み SQL キーワードが指定されている場合、**EXTERNALNAME** は **NAME** 値に「\_IDX」を付加したうえで、その **NAME** 値をデフォルトの外部名として受け入れます。IMS Universal ドライバーによって制限されている予約済み SQL キーワードのリストについては、『IMS JDBC ドライバーにより制限されるポータブル SQL キーワード』を参照してください。

#### **PSELRTN=**

HISAM または SHISAM 高速機能副次索引データベースでユーザー区画化が要求されている場合に、ユーザー区画選択出口ルーチンの名前を識別します。

#### **PSELOPT=**

ユーザー区画グループ内のユーザー区画データベースが、そのユーザー区画データベースのデータの終わりに達する前に、SSA 処理を行わない修飾 GN 呼び出しでどのように論理的にグループ化されるかを示します。ユーザー区画データベ

ースは、LCHILD ステートメントの NAME= パラメーターでユーザー区画グループの一部として定義されます。このパラメーターは高速機能副次索引データベースのみに適用されます。

PSELOPT= パラメーターは、PSB 内で PROCSEQD= パラメーターが指定された PCB ステートメントにも指定することができます。PROCSEQD= パラメーターが指定された PCB ステートメントでは PSELOPT= パラメーターのデフォルトはありません。XDFLD ステートメントと、PROCSEQD オペランドが指定された PCB ステートメントの両方に PSELOPT= パラメーターが指定されている場合、PCB ステートメントの PSELOPT= パラメーターが優先されます。

#### **MULT**

ユーザー・データ区画グループ内の選択されたユーザー区画とその後続のユーザー区画データベースを、1 次 DEDB データベース DBD の LCHILD ステートメントの NAME= パラメーターで物理的に定義されていると示します。

#### **SNGL**

選択されたユーザー区画データベースのみが使用されることを示します。PSELOPT=MULT は、XDFLD ステートメントの PSELOPT= パラメーターのデフォルトです。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### **CORRECT**

```
REMARKS='These remarks apply to the XYZ application'
```

#### **INCORRECT**

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

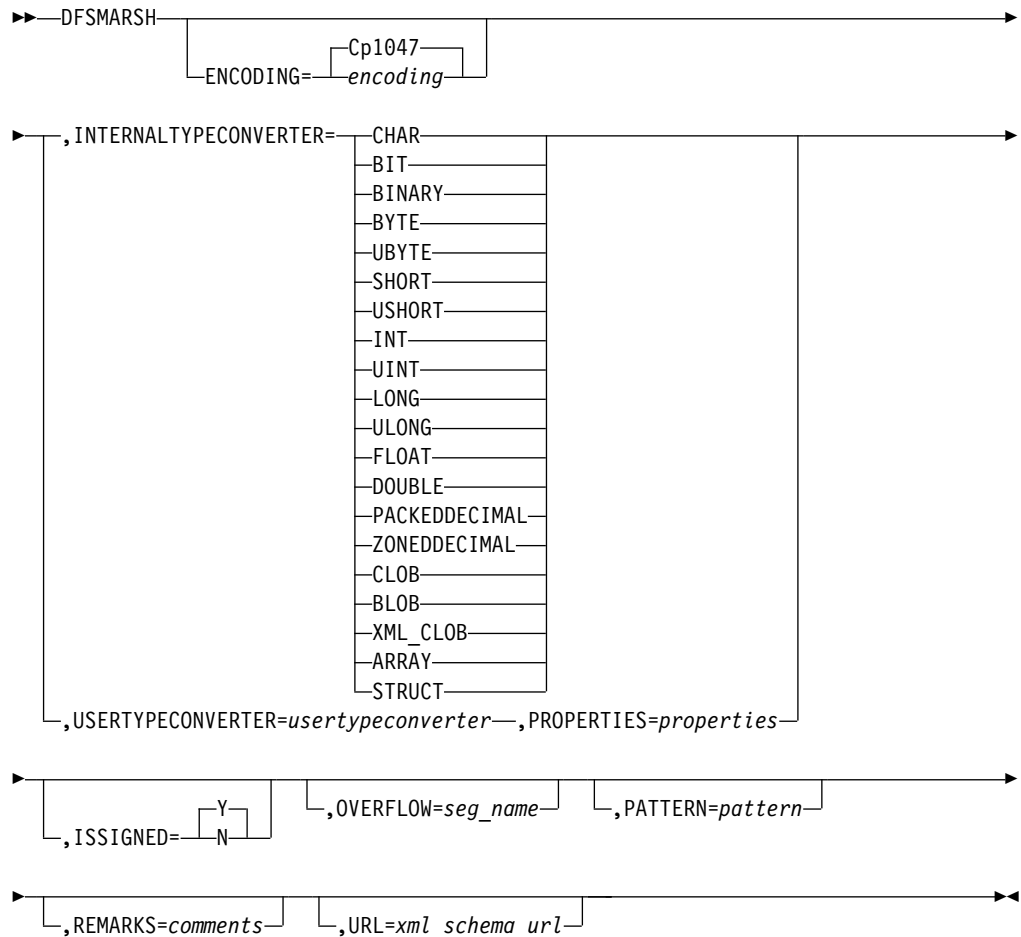
## **DFSMARSH ステートメント**

DFSMARSH ステートメントは、フィールドのマーシャル属性を定義します。

DFSMARSH ステートメントは、DBD 生成ユーティリティへの入力内で、その適用先の FIELD ステートメントの直後に続ける必要があります。

## すべてのデータベース・タイプの DFSMARSH ステートメント構文図

すべてのデータベース・タイプでの DFSMARSH ステートメントの形式を、次の構文図に示します。



## DFSMARSH ステートメント・パラメーターの説明

### ENCODING=

フィールド内の文字データのエンコードを指定する、1 から 25 文字のオプション・フィールド。 ENCODING が DFSMARSH ステートメントで有効なのは、INTERNALTYPECONVERTER=CHAR の場合のみです。

ENCODING キーワードに指定する値には、以下の文字を含めることはできません。

- 単一および二重引用符
- ブランク
- より小 (<) およびより大 (>) 記号
- アンパーサンド (&)

ENCODING パラメーターが DFSMARSH ステートメントに指定されていない場合、デフォルト値は、SEGM ステートメントの ENCODING パラメーター、または SEGM ステートメントに ENCODING が指定されていない場合は

DBD ステートメントの ENCODING パラメーターに指定された値によって決まります。SEGM ステートメントにも DBD ステートメントにも ENCODING パラメーターが指定されていない場合、ENCODING パラメーターのデフォルト値は EBCDIC エンコードを指定する Cp1047 です。

#### **INTERNALTYPECONVERTER=**

IMS データをアプリケーション・プログラムが要求するデータ型に変換するために IMS が使用する、内部変換ルーチンを指定します。

必ず INTERNALTYPECONVERTER または USERTYPECONVERTER のいずれか (両方ではなく) を指定します。INTERNALTYPECONVERTER と USERTYPECONVERTER は同時には指定できません。

DFSMARSH ステートメントに INTERNALTYPECONVERTER パラメーターをコーディングする場合、明示的に値をコーディングする必要があります。この値をブランクのままにしたり、ヌル値を指定したりすることはできません。

INTERNALTYPECONVERTER パラメーターの有効な値は以下のとおりです。

#### **ARRAY**

#### **BINARY**

#### **BIT**

BIT を指定する場合は、対応する FIELD ステートメントに BYTES=1 を指定することも必要です。

#### **BLOB**

#### **BYTE**

BYTE を指定する場合は、対応する FIELD ステートメントに BYTES=1 を指定することも必要です。

#### **UBYTE**

UBYTE を指定する場合は、BYTES=1 も指定し、さらに DATATYPE=BYTE または DATATYPE=UBYTE のいずれかを対応する FIELD ステートメントで指定する必要があります。

#### **CHAR**

#### **CLOB**

#### **DOUBLE**

DOUBLE を指定する場合は、対応する FIELD ステートメントに BYTES=8 を指定することも必要です。

#### **FLOAT**

FLOAT を指定する場合は、対応する FIELD ステートメントに BYTES=4 を指定することも必要です。

#### **INT**

INT を指定する場合は、対応する FIELD ステートメントに BYTES=4 を指定することも必要です。

#### **UINT**

UBYTE を指定する場合は、BYTES=4 も指定し、さらに DATATYPE=INT または DATATYPE=UINT のいずれかを対応する FIELD ステートメントで指定する必要があります。

**LONG**

LONG を指定する場合は、対応する FIELD ステートメントに BYTES=8 を指定することも必要です。

**ULONG**

ULONG を指定する場合は、BYTES=8 も指定し、さらに DATATYPE=LONG または DATATYPE=ULONG のいずれかを対応する FIELD ステートメントで指定する必要があります。

**PACKEDDECIMAL****SHORT**

SHORT を指定する場合は、対応する FIELD ステートメントに BYTES=2 を指定することも必要です。

**USHORT**

USHORT を指定する場合は、BYTES=2 も指定し、さらに DATATYPE=SHORT または DATATYPE=USHORT のいずれかを対応する FIELD ステートメントで指定する必要があります。

**STRUCT****XML\_CLOB****ZONEDDECIMAL**

INTERNALTYPECONVERTER を指定する場合、この DFSMARSH ステートメントの適用先の FIELD ステートメントに DATATYPE パラメーターを指定することも必要です。

INTERNALTYPECONVERTER パラメーターに指定する値は、DATATYPE パラメーターに指定された値と一致している必要があります。ほとんどの場合、INTERNALTYPECONVERTER には DATATYPE パラメーターに指定したものと同一値を指定する必要があります。次の表に、この規則で有効な例外を示します。

表 10. DATATYPE 値に基づく追加の有効値

DATATYPE 値	有効な INTERNALTYPECONVERTER 値
BINARY	BINARY、BLOB、CLOB、および XML_CLOB
BYTE	BYTE、UBYTE
日付	LONG、CHAR
DECIMAL(pp,ss)	PACKEDDECIMAL、ZONEDDECIMAL、BINARY
INT	INT、UINT
LONG	LONG、ULONG
SHORT	SHORT、USHORT
TIME	LONG、CHAR
TIMESTAMP	LONG、CHAR
XML	XML_CLOB

DATATYPE パラメーターに DATE、TIME、TIMESTAMP のいずれかが指定されている場合に INTERNALTYPECONVERTER=LONG を指定すると、その値は 1970 年 1 月 1 日からのミリ秒数として DASD に保管されます。

DFSMARSH ステートメントをコーディングしない場合、IMS は内部で INTERNALTYPECONVERTER パラメーターの値を、FIELD ステートメントの DATATYPE パラメーターの値によって決まるデフォルト値に設定します。

ほとんどの場合、INTERNALTYPECONVERTER のデフォルト値は DATATYPE パラメーターの値と同じです。次の表に、この規則の例外を示します。

表 11. DFSMARSH ステートメントが指定されない場合の INTERNALTYPECONVERTER のデフォルト値

DATATYPE 値	INTERNALTYPECONVERTER のデフォルト値
日付	LONG
DECIMAL(pp,ss)	符号付き PACKEDDECIMAL
TIME	LONG
TIMESTAMP	LONG
XML	XML_CLOB

#### ISSIGNED=

DATATYPE=DECIMAL の場合にのみ有効です。

有効な値は Y または N です。デフォルトは Y です。

#### OVERFLOW=

XML 文書の保持用に定義されたフィールドに収まらない XML 文書の一部の保管に使用できる、従属セグメントの 1 から 8 文字の名前。

この従属セグメントの親は、XML データ・フィールドを含むセグメントです。親セグメントの名前は、従属セグメントを定義する SEGM ステートメントの PARENT パラメーターに指定する必要があります。

OVERFLOW パラメーターは、XML\_CLOB データに対して DATATYPE=XML が指定されているフィールドにのみ適用されます。

#### PATTERN=

日付、時刻、およびタイム・スタンプの Java データ型用のパターンを指定する、1 から 50 文字のオプション・フィールド。

PATTERN パラメーターが適用されるのは、FIELD ステートメントの DATATYPE キーワードに DATE、TIME、または TIMESTAMP が指定され、DFSMARSH ステートメントの INTERNALTYPECONVERTER キーワードに CHAR が指定されている場合のみです。PATTERN はこれ以外のデータ型では無効です。

パターンは大/小文字が区別され、必ず単一引用符で囲む必要があります。

キーワード値の区切り文字として使用される単一引用符を除き、PATTERN キーワードに指定する値に以下の文字を含めることはできません。

- 単一および二重引用符
- より小 (<) およびより大 (>) 記号

- アンパーサンド (&)

指定できるパターンは、Java クラス `java.text.SimpleDateFormat` によって定義されます。DBD 生成ユーティリティでは、PATTERN に入力した値が Java で定義されたパラメーターに従っているかどうかはチェックされません。

例えば、`yyyy.MM.dd` という Java 形式を入力すると、結果として得られる時刻形式は「2013.01.01」になります。

#### PROPERTIES=

USERTYPECONVERTER パラメーターに指定されたユーザー・タイプ・コンバーターに対してプロパティを指定します。これらのプロパティが、ユーザー・タイプ・コンバーターに渡されます。

USERTYPECONVERTER が指定されている場合にのみ有効です。

PROPERTIES キーワードに指定する名前とプロパティは、大/小文字の区別があります。

PROPERTIES キーワードでは以下の文字はサポートされません。

- 単一および二重引用符
- ブランク
- より小 (<) およびより大 (>) 記号
- アンパーサンド (&)

プロパティ名の最大長は 128 文字です。プロパティ値の最大長も 128 文字です。

形式は次のとおりです。

```
PROPERTIES=(property1_name=property1_value,property2_name=property2_value)
```

以下に例を示します。

```
PROPERTIES=(DOG=BUTCH,CAT=LUCY)
```

#### REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### CORRECT

```
REMARKS='These remarks apply to the XYZ application'
```

#### INCORRECT

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。

- より大 ( > ) 記号。
- アンパーサンド (&)。

#### **URL=**

このフィールドを記述する XML スキーマを参照する URL の、1 から 256 文字のオプション・フィールド。例えば、URL=MySchema.xsd です。

URL キーワードに指定する値には、以下の文字を含めることはできません。

- 単一および二重引用符
- ブランク
- より小 ( < ) およびより大 ( > ) 記号
- アンパーサンド (&)

URL パラメーターは、XML\_CLOB データに対して DATATYPE=XML が指定されているフィールドにのみ適用されます。

#### **USERTYPECONVERTER=**

型変換に使用されるユーザー提供 Java クラスの、1 から 256 文字の完全修飾 Java クラス名。

USERTYPECONVERTER キーワードに指定する値には、以下の文字を含めることはできません。

- 単一および二重引用符
- ブランク
- より小 ( < ) およびより大 ( > ) 記号
- アンパーサンド (&)

INTERNALTYPECONVERTER と同時には指定できません。

以下に例を示します。

```
USERTYPECONVERTER=class://com.ibm.ims.dli.types.PackedDateConverter
```

### **DFSMARSH ステートメントの例**

以下の一連の例は、さまざまな DATATYPE およびタイプ・コンバーター指定に可能な DFSMARSH ステートメントの使用方法を示したものです。

#### **DATATYPE=DATE:**

```
FIELD    EXTERNALNAME=XDATE,
          BYTES=8,
          START=84,
          DATATYPE=DATE
DFSMARSH ENCODING=Cp1047,
          INTERNALTYPECONVERTER=CHAR,
          PATTERN='MMdyyy'
```

#### **DATATYPE=TIME:**

```
FIELD    EXTERNALNAME=XTIME,
          BYTES=6,
          START=92,
          DATATYPE=TIME
DFSMARSH ENCODING=Cp1047,
          INTERNALTYPECONVERTER=CHAR,
          PATTERN='HHmmss'
```



#### **DATATYPE=TIMESTAMP:**

```
FIELD    EXTERNALNAME=XTIMESTAMP,  
          BYTES=16,  
          START=84,  
          DATATYPE=TIMESTAMP  
DFSMARSH ENCODING=Cp1047,  
          INTERNALTYPECONVERTER=CHAR,  
          PATTERN='MMddyyyHHmssff'
```

#### **DATATYPE=ZONEDDECIMAL:**

```
FIELD NAME=ORDPRICE,  
      BYTES=10,  
      START=21,  
      DATATYPE=DECIMAL(10,2)  
DFSMARSH INTERNALTYPECONVERTER=ZONEDDECIMAL,  
          ISSIGNED=Y
```


#### **DATATYPE=PACKEDDECIMAL:**

```
FIELD    EXTERNALNAME=XPACKEDDEC1,  
          BYTES=4,  
          START=60,  
          DATATYPE=DECIMAL(7,2)  
DFSMARSH INTERNALTYPECONVERTER=PACKEDDECIMAL,  
          ISSIGNED=Y
```

#### **USERTYPECONVERTER=:**

```
FIELD    EXTERNALNAME=PACKEDDATEFIELD,  
          BYTES=5,  
          START=40,  
          DATATYPE=DATE  
DFSMARSH USERTYPECONVERTER=class://com.ibm.ims.dli.types.PackedDateConverter,  
          PROPERTIES=(ZONE=PACIFIC, DAYLIGHTSAVINGS=TRUE)
```

関連タスク:

 アプリケーション・プログラムのデータ・タイプの指定 (データベース管理)

関連資料:

122 ページの『FIELD ステートメント』

## **DFSMAP ステートメント**

DFSMAP ステートメントを使用すると、セグメント内のフィールドの代替マッピングが可能になります。

DFSMAP ステートメントは 1 つ以上のマップ・ケースから成るグループを定義し、それらのケースを制御フィールドに関連付けます。制御フィールドは、特定のセグメント・インスタンスで使用されるマップ・ケースを識別します。

すべてのデータベース・タイプでの DFSMAP ステートメントの形式を、次の構文図に示します。

すべてのデータベース・タイプの **DFSMAP** ステートメント構文図

```
▶▶—DFSMAP—NAME=map_name—,DEPENDINGON=field_name—┌───────────┐──▶▶  
└───────────┘ ┌───────────┘  
REMARKS=remarks
```

## DFSMAP ステートメント・パラメーターの説明

### DEPENDINGON=

特定のセグメント・インスタンスに使用されるマップ・ケースを判別する値が格納された、このセグメント内の制御フィールドの外部名。このマップの DFSCASE ステートメントの CASEID に対応する値が制御フィールドに含まれない場合、そのマップはこのセグメント・インスタンスでは使用されていません。制御ステートメントを定義する FIELD ステートメントに EXTERNALNAME パラメーターが明示的にコーディングされていない場合は、DEPENDINGON フィールドに NAME パラメーターの値を指定します。

### NAME=

このマップの名前を定義する、1 から 128 文字の必須の英数字フィールド。ブランクはサポートされません。

### REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### CORRECT

```
REMARKS='These remarks apply to the XYZ application'
```

#### INCORRECT

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## マッピング例: DFSMAP および DFSCASE

以下の例は、自動車保険契約、住宅保険契約、船舶保険契約の 3 つの異なるタイプの保険契約に関するデータの保管に使用される単一のセグメントを定義するために、DBD ソース内でマッピングを使用する方法を示しています。それぞれの保険契約タイプごとに、そのタイプに固有の情報を保持するための異なるフィールドが必要です。

DBD ソースでは、各保険契約タイプ用のフィールドが、異なる DFSCASE ステートメントによってマップされています。この例の 3 つのマップ・ケースには、それぞれ AUTOMAP、HOMEMAP、BOATMAP の名前が付けられています。任意の DFSCASE ステートメントによって定義されたマップを構成するフィールドは、FIELD ステートメント内の CASENAME パラメーターでそれぞれが属している

DFSCASE ステートメントの名前を指定しています。DFSCASE ステートメントは、セグメント CUSTOMERPOLICY 内の DFSMAP ステートメント POLICYMAPS によってグループ分けされます。

各マップ・ケースの CASEID パラメーターで指定されている値は、そのマップ・ケースを一意的に識別し、制御フィールド値の役割を果たします。セグメント・インスタンスが最初にデータベースに挿入される時点で、そのセグメント・インスタンスが使用するマップ・ケースの ID が制御フィールドに挿入されます。この例では、制御フィールドの名前は POLICYTYPE です。実行時には、アプリケーション・プログラムがデータベースからセグメントを取り出す際に制御フィールド値を評価して、フィールドの正しいマッピングを判別しなければなりません。

```

DBD      NAME=POLICYDB,          C
          ENCODING=CP1047,      C
          ACCESS=(DEDB),        C
          RMNAME=(RMOD3),       C
          PASSWD=NO
AREA     DD1=PLCYAR01,          C
          DEVICE=3330,          C
          SIZE=(2048),          C
          UOW=(15,10),          C
          ROOT=(10,5),          C
          REMARKS='AREA NUMBER 1 FOR POLICYDB DATABASE'
SEGM     NAME=CUSTOMER,         C
          PARENT=0,             C
          BYTES=(390,20)
FIELD    NAME=(CUSTKEY,SEQ,U),  C
          BYTES=12,             C
          START=1,              C
          TYPE=C
SEGM     NAME=POLICY,           C
          EXTERNALNAME=CUSTOMERPOLICY, C
          ENCODING=CP1047,      C
          PARENT=CUSTOMER,      C
          BYTES=(900),          C
          TYPE=DIR,              C
          RULES=(LLL,HERE)
*****
*          CONTROL FIELD:
*****
FIELD    EXTERNALNAME=POLICYTYPE, C
          BYTES=4,              C
          START=1,              C
          DATATYPE=CHAR
*****
*          DFSMAP STATEMENT:
*****
DFSMAP   NAME=POLICYMAPS,       C
          DEPENDINGON=POLICYTYPE
*****
*          DFSCASE STATEMENT 1:
*****
DFSCASE  NAME=AUTOMAP,          C
          CASEID=AUTO,          C
          CASEIDTYPE=C,         C
          MAPNAME=POLICYMAPS,   C
          REMARKS='DEFINES THE FIELDS OF AN AUTO INSURANCE POLICY'
FIELD    EXTERNALNAME=AUTOMAKE, C
          CASENAME=AUTOMAP,     C
          BYTES=15,             C
          START=5,              C
          DATATYPE=CHAR
FIELD    EXTERNALNAME=MODEL,    C
          CASENAME=AUTOMAP,     C

```


```

        BYTES=15,           C
        START=20,          C
        DATATYPE=CHAR
FIELD   EXTERNALNAME=YEAR,   C
        CASENAME=AUTOMAP,   C
        BYTES=4,            C
        START=35,           C
        DATATYPE=CHAR
*****
*       DFSCASE STATEMENT 2:
*****
DFSCASE NAME=HOMEMAP,      C
        CASEID=HOME,       C
        CASEIDTYPE=C,      C
        MAPNAME=POLICYMAPS, C
        REMARKS='DEFINES THE FIELDS OF A HOME INSURANCE POLICY'
FIELD   EXTERNALNAME=DWELLING_TYPE, C
        CASENAME=HOMEMAP,  C
        BYTES=20,          C
        START=5,           C
        DATATYPE=CHAR
FIELD   EXTERNALNAME=ROOMS,   C
        CASENAME=HOMEMAP,   C
        BYTES=5,            C
        START=25,           C
        DATATYPE=CHAR
FIELD   EXTERNALNAME=SQ_FOOT, C
        CASENAME=HOMEMAP,  C
        BYTES=6,            C
        START=30,           C
        DATATYPE=CHAR
*****
*       DFSCASE STATEMENT 3:
*****
DFSCASE NAME=BOATMAP,      C
        CASEID=BOAT,       C
        CASEIDTYPE=C,      C
        MAPNAME=POLICYMAPS, C
        REMARKS='DEFINES THE FIELDS OF A BOAT INSURANCE POLICY'
FIELD   EXTERNALNAME=CLASS,  C
        CASENAME=BOATMAP,  C
        BYTES=10,           C
        START=5,            C
        DATATYPE=CHAR
FIELD   EXTERNALNAME=LENGTH, C
        CASENAME=BOATMAP,  C
        BYTES=6,            C
        START=15,           C
        DATATYPE=CHAR
FIELD   EXTERNALNAME=BOATMAKE, C
        CASENAME=BOATMAP,  C
        BYTES=10,           C
        START=21,           C
        DATATYPE=CHAR

DBDGEN
FINISH
END

```

関連タスク:

 セグメントの代替フィールド・マップの定義 (データベース管理)

関連資料:

122 ページの『FIELD ステートメント』

161 ページの『DFSCASE ステートメント』

## DFSCASE ステートメント

DFSCASE ステートメントはマップ・ケースを定義します。マップ・ケースとは FIELD ステートメントの集合であり、この集合全体が、セグメント定義内の特定のバイト範囲に対し、オプションの代替フィールド・レイアウトを定義します。

1 つのセグメント内の同じバイト範囲をマップするマップ・ケースは、DFSMAP ステートメントによってグループ化されます。また DFSMAP ステートメントは、マップ・ケースを、セグメント定義で別個に定義された制御フィールドにリンクします。

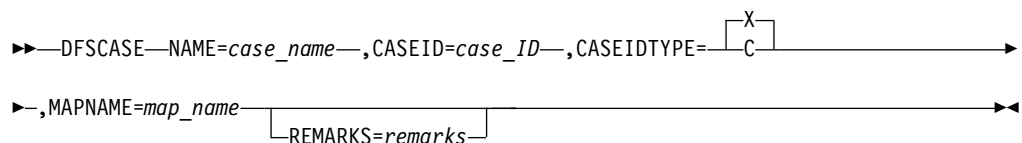
各マップ・ケースには固有の ID があります。セグメントのインスタンスでは、有効なマップ・ケースの ID が、セグメント作成時に制御フィールドに保管されます。

IMS Universal ドライバーが使用されていない限り、マップ・ケースによって定義されたフィールド・レイアウトを、このバイト範囲にアクセスするアプリケーション・プログラムに対し、COBOL コピーブックまたはその他のプログラミング成果物を使用して定義する必要があります。セグメント・インスタンスへのアクセス時に、アプリケーション・プログラムは制御フィールドの値を確認して、どのコピーブックを使用するかを決定します。

アプリケーション・プログラムが IMS Universal ドライバーを使用して IMS にアクセスする場合は、アプリケーション・プログラムに対してフィールド・レイアウトを定義するための追加のプログラミング成果物は必要ありません。

すべてのデータベース・タイプでの DFSCASE ステートメントの形式を、次の構文図に示します。

### すべてのデータベース・タイプの DFSCASE ステートメント構文図



## DFSCASE ステートメント・パラメーターの説明

### CASEID

ケースの固有 ID を定義する 1 から 128 バイトのフィールド。

セグメントのフィールド構造の一部または全部がこのケースによってマップされている場合、そのセグメントのインスタンスはユーザー定義制御フィールドにこの CASEID 値を指定します。

CASEIDTYPE=C の場合、CASEID フィールドには英数字、\_、@、\$、および # を含めることができます。単一引用符はサポートされていますが、必要はありません。ブランクはサポートされません。

CASEIDTYPE=X の場合、CASEID パラメーターで有効な文字は 0 から 9 および A から F のみです。

CASEID 値の長さは、ユーザー定義の制御フィールドの長さでサポートされているものでなければなりません。CASEIDTYPE=C の場合、CASEID 値の長さは、必ず制御フィールドの BYTES パラメーターに指定された値以下とします。CASEIDTYPE=X の場合、CASEID 値の長さは、必ず制御フィールドの BYTES パラメーターに指定された値の正確に 2 倍とします。

ケース ID は、そのケースの所属先マップ内で固有でなければなりません。

#### **CASEIDTYPE**

CASEID パラメーターで指定した値のデータ型を定義します。有効な値は、Cp1047 (EBCDIC 文字エンコード) を指定する C、および 16 進数を指定する X です。

CASEIDTYPE に C か X のどちらを指定するかに応じて、CASEID 値の有効な長さは異なる方法で計算されます。この長さが有効なのは、DFSMAP ステートメントの DEPENDINON パラメーターが参照するフィールドの BYTES パラメーターに指定された長さと、その長さが一致する場合です。

CASEIDTYPE=C の場合、CASEID 値の長さは、必ず BYTES パラメーターに指定された値以下とします。CASEIDTYPE=X の場合、CASEID 値の長さは、必ず BYTES パラメーターに指定された値の正確に 2 倍とします。

#### **MAPNAME**

DFSMAP ステートメントの NAME パラメーターに指定されている、このケースの所属先マップの名前。このフィールドは必須です。

#### **NAME**

このケースの名前を定義する、1 から 128 文字の必須の英数字フィールド。ブランクはサポートされません。

ケース名はセグメント内で固有でなければなりません。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### **CORRECT**

```
REMARKS='These remarks apply to the XYZ application'
```

#### **INCORRECT**

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## マッピング例: DFSMAP および DFSCASE

以下の例は、自動車保険契約、住宅保険契約、船舶保険契約の 3 つの異なるタイプの保険契約に関するデータの保管に使用される単一のセグメントを定義するために、DBD ソース内でマッピングを使用する方法を示しています。それぞれの保険契約タイプごとに、そのタイプに固有の情報を保持するための異なるフィールドが必要です。

DBD ソースでは、各保険契約タイプ用のフィールドが、異なる DFSCASE ステートメントによってマップされています。この例の 3 つのマップ・ケースには、それぞれ AUTOMAP、HOMEMAP、BOATMAP の名前が付けられています。任意の DFSCASE ステートメントによって定義されたマップを構成するフィールドは、FIELD ステートメント内の CASENAME パラメーターでそれぞれが属している DFSCASE ステートメントの名前を指定しています。DFSCASE ステートメントは、セグメント CUSTOMERPOLICY 内の DFSMAP ステートメント POLICYMAPS によってグループ分けされます。

各マップ・ケースの CASEID パラメーターで指定されている値は、そのマップ・ケースを一意的に識別し、制御フィールド値の役割を果たします。セグメント・インスタンスが最初にデータベースに挿入される時点で、そのセグメント・インスタンスが使用するマップ・ケースの ID が制御フィールドに挿入されます。この例では、制御フィールドの名前は POLICYTYPE です。実行時には、アプリケーション・プログラムがデータベースからセグメントを取り出す際に制御フィールド値を評価して、フィールドの正しいマッピングを判別しなければなりません。

```
DBD      NAME=POLICYDB,          C
          ENCODING=CP1047,      C
          ACCESS=(DEDB),        C
          RMNAME=(RMOD3),       C
          PASSWD=NO
      AREA DD1=PLCYAR01,         C
          DEVICE=3330,          C
          SIZE=(2048),          C
          UOW=(15,10),          C
          ROOT=(10,5),          C
          REMARKS='AREA NUMBER 1 FOR POLICYDB DATABASE'
      SEGM NAME=CUSTOMER,        C
          PARENT=0,             C
          BYTES=(390,20)
      FIELD NAME=(CUSTKEY,SEQ,U), C
          BYTES=12,             C
          START=1,              C
          TYPE=C
      SEGM NAME=POLICY,          C
          EXTERNALNAME=CUSTOMERPOLICY, C
          ENCODING=CP1047,      C
          PARENT=CUSTOMER,      C
          BYTES=(900),          C
          TYPE=DIR,             C
          RULES=(LLL,HERE)
*****
*      CONTROL FIELD:
*****
      FIELD EXTERNALNAME=POLICYTYPE, C
          BYTES=4,              C
          START=1,              C
          DATATYPE=CHAR
*****
*      DFSMAP STATEMENT:
```

```

*****
      DFSMAP  NAME=POLICYMAPS,                                C
              DEPENDONGON=POLICYTYPE
*****
*      DFSCASE STATEMENT 1:
*****
      DFSCASE  NAME=AUTOMAP,                                  C
              CASEID=AUTO,                                    C
              CASEIDTYPE=C,                                  C
              MAPNAME=POLICYMAPS,                            C
              REMARKS='DEFINES THE FIELDS OF AN AUTO INSURANCE POLICY'
      FIELD    EXTERNALNAME=AUTOMAKE,                          C
              CASENAME=AUTOMAP,                               C
              BYTES=15,                                       C
              START=5,                                        C
              DATATYPE=CHAR
      FIELD    EXTERNALNAME=MODEL,                             C
              CASENAME=AUTOMAP,                               C
              BYTES=15,                                       C
              START=20,                                       C
              DATATYPE=CHAR
      FIELD    EXTERNALNAME=YEAR,                              C
              CASENAME=AUTOMAP,                               C
              BYTES=4,                                        C
              START=35,                                       C
              DATATYPE=CHAR
*****
*      DFSCASE STATEMENT 2:
*****
      DFSCASE  NAME=HOMEMAP,                                  C
              CASEID=HOME,                                    C
              CASEIDTYPE=C,                                  C
              MAPNAME=POLICYMAPS,                            C
              REMARKS='DEFINES THE FIELDS OF A HOME INSURANCE POLICY'
      FIELD    EXTERNALNAME=DWELLING_TYPE,                      C
              CASENAME=HOMEMAP,                               C
              BYTES=20,                                       C
              START=5,                                        C
              DATATYPE=CHAR
      FIELD    EXTERNALNAME=ROOMS,                             C
              CASENAME=HOMEMAP,                               C
              BYTES=5,                                        C
              START=25,                                       C
              DATATYPE=CHAR
      FIELD    EXTERNALNAME=SQ_FOOT,                          C
              CASENAME=HOMEMAP,                               C
              BYTES=6,                                        C
              START=30,                                       C
              DATATYPE=CHAR
*****
*      DFSCASE STATEMENT 3:
*****
      DFSCASE  NAME=BOATMAP,                                  C
              CASEID=BOAT,                                    C
              CASEIDTYPE=C,                                  C
              MAPNAME=POLICYMAPS,                            C
              REMARKS='DEFINES THE FIELDS OF A BOAT INSURANCE POLICY'
      FIELD    EXTERNALNAME=CLASS,                             C
              CASENAME=BOATMAP,                               C
              BYTES=10,                                       C
              START=5,                                        C
              DATATYPE=CHAR
      FIELD    EXTERNALNAME=LENGTH,                           C
              CASENAME=BOATMAP,                               C
              BYTES=6,                                        C
              START=15,                                       C
              DATATYPE=CHAR

```



```

FIELD    EXTERNALNAME=BOATMAKE,          C
          CASENAME=BOATMAP,              C
          BYTES=10,                      C
          START=21,                      C
          DATATYPE=CHAR


```

```

DBDGEN
FINISH
END

```

関連タスク:

 セグメントの代替フィールド・マップの定義 (データベース管理)

関連資料:

122 ページの『FIELD ステートメント』

157 ページの『DFSMAP ステートメント』

## DBDGEN、FINISH、および END ステートメント

すべての DBD 生成ユーティリティの制御ステートメントには、END ステートメントが続いていなければなりません。

ユーティリティ・ステートメントがさらに 3 つあります。2 つは必須であり (DBDGEN と END)、1 つはオプションです (FINISH)。

DBDGEN ステートメントは、DBD の定義に使用される DBD 生成ステートメントの終了を示します。このステートメントは必須です。次の例は、すべてのデータベース・タイプに該当する DBDGEN ステートメントの形式を示しています。

```

▶▶—DBDGEN—▶▶

```

FINISH ステートメントは、オプションであり、互換性の目的で保持されています。次の例は、すべてのデータベース・タイプに該当する FINISH ステートメントの形式を示しています。

```

▶▶—FINISH—▶▶

```

END ステートメントは、入力ステートメントの終わりをアセンブラーに伝えます。このステートメントは必須です。次の例は、すべてのデータベース・タイプに該当する END ステートメントの形式を示しています。

```

▶▶—END—▶▶

```

## DBDGEN ユーティリティの例

これらの例では、DBDGEN ユーティリティをさまざまなデータベース・タイプの DBD の生成に使用する方法を示します。

アプリケーション・プログラムでは、前述のどのデータベースにもデータベース PCB を介して操作を行うことができます。データベース・ステートメントでの DBDNAME= パラメーターの値は、DBD 生成の DBD ステートメントでの

NAME= パラメーター値に等しい値にしてください。PSB 生成でデータベース・ステートメントに続く SENSEG ステートメントは、指定された DBD 生成の SEGM ステートメントで定義されたセグメントを参照する必要があります。

アプリケーション・プログラムが HIDAM データベースを使用する場合は、ステートメント上の DBDNAME= パラメーターの値を、HIDAM の DBD 生成の DBD ステートメントで指定する NAME= パラメーターと同じ値にしてください。

HIDAM の DBD の LCHILD ステートメントは、必要な INDEX DBD および索引データベースとの関係を IMS に与えます。INDEX DBD 名を、データベース PCB の DBDNAME= パラメーターに指定してはなりません。

## 副次索引または論理関係がない例

DBD 生成の例には、2 次索引または論理関係がない HSAM、HISAM、HDAM、HIDAM、1 次 HIDAM 索引、GSAM、MSDB および DEDB データベースを定義する場合に必要なステートメントが示されています。

2 種類のデータ構造を次に示します。1 つは給与計算インベントリーのデータ構造で使用される階層順序を表し、NAME、ADDRESS、および PAYROLL を含んでいます。もう 1 つはスキル・インベントリーのデータ構造で使用される階層順序を表し、SKILL、NAME、EXPERIENCE、および EDUCATION を含んでいます。

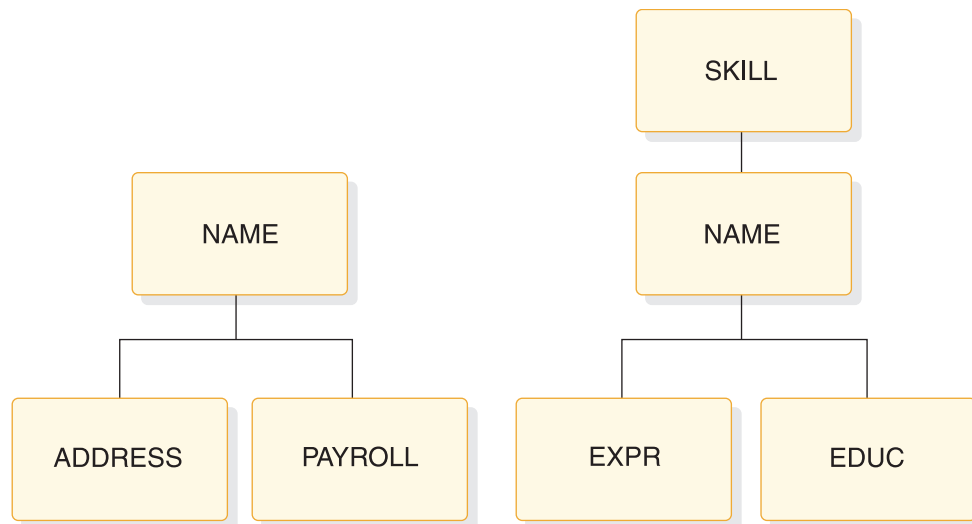


図 6. 給与計算インベントリーおよびスキル・インベントリーのデータ構造

## HSAM の DBD 生成の例

以下の例は、スキル・インベントリーと給与計算のデータ構造を HSAM データベースとして定義する DBD 生成のステートメントです。

スキル・インベントリー・データベースの **HSAM** の DBD 生成

```
DBD NAME=SKILLINV,ACCESS=HSAM
DATASET DD1=SKILLHSAM,DD2=HSAMOUT,BLOCK=1,
RECORD=3000
```

```
SEGM NAME=SKILL,BYTES=31,FREQ=100,PARENT=0
FIELD NAME=TYPE,BYTES=21,START=1,TYPE=C
FIELD NAME=STDCODE,BYTES=10,START=22,TYPE=C
```

```

SEGM  NAME=NAME,BYTES=20,FREQ=500,PARENT=SKILL
FIELD  NAME=STDCLEVL,BYTES=20,START=1,TYPE=C

SEGM  NAME=EXPR,BYTES=20,FREQ=10,PARENT=NAME
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C

SEGM  NAME=EDUC,BYTES=75,FREQ=5,PARENT=NAME
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C

DBDGEN
FINISH
END

```

### 給与データベースの *HISAM* の *DBD* 生成

```

DBD  NAME=PAYROLDB,ACCESS=HSAM
DATASET DD1=PAYROLL,DD2=PAYOUT,BLOCK=1,RECORD=1000,

SEGM  NAME=NAME,BYTES=150,FREQ=1000,PARENT=0
FIELD  NAME=(EMPLOYEE,SEQ,U),BYTES=60,START=1,TYPE=C
FIELD  NAME=MANNBR,BYTES=15,START=61,TYPE=C
FIELD  NAME=ADDR,BYTES=75,START=76,TYPE=C

SEGM  NAME=ADDRESS,BYTES=200,FREQ=2,PARENT=NAME
FIELD  NAME=HOMEADDR,BYTES=100,START=1,TYPE=C
FIELD  NAME=COMAILOC,BYTES=100,START=101,TYPE=C

SEGM  NAME=PAYROLL,BYTES=100,FREQ=1,PARENT=NAME
FIELD  NAME=HOURS,BYTES=15,START=51,TYPE=P
FIELD  NAME=BASICPAY,BYTES=15,START=1,TYPE=P

DBDGEN
FINISH
END

```

### **HISAM** の **DBD** 生成の例

次の例は、スキル・インベントリーと給与計算のデータ構造を *HISAM* データベースとして定義する *DBD* 生成ステートメントを示します。

#### スキル・インベントリー *SKILLINV* データベースの *HISAM* の *DBD* 生成

```

DBD  NAME=SKILLINV,ACCESS=HISAM
DATASET DD1=SKLHISAM,OVFLW=HISAMOVF,

SEGM  NAME=SKILL,BYTES=31,FREQ=100
FIELD  NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD  NAME=STDCODE,BYTES=10,START=22,TYPE=C

SEGM  NAME=NAME,BYTES=20,FREQ=500,PARENT=SKILL
FIELD  NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C

SEGM  NAME=EXPR,BYTES=20,FREQ=10,PARENT=NAME
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C

SEGM  NAME=EDUC,BYTES=75,FREQ=5,PARENT=NAME
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C

DBDGEN
FINISH
END

```

## 給与データベースの *HISAM* の *DBD* 生成

```
DBD  NAME=PAYROLDB,ACCESS=HISAM
DATASET DD1=PAYROLL,OVFLW=PAYROLOV,

SEGM  NAME=NAME,BYTES=150,FREQ=1000,PARENT=0
FIELD NAME=(EMPLOYEE,SEQ,U),BYTES=60,START=1,TYPE=C
FIELD NAME=MANNBR,BYTES=15,START=61,TYPE=C
FIELD NAME=ADDR,BYTES=75,START=76,TYPE=C

SEGM  NAME=ADDRESS,BYTES=200,FREQ=2,PARENT=NAME
FIELD NAME=HOMEADDR,BYTES=100,START=1,TYPE=C
FIELD NAME=COMAiloc,BYTES=100,START=101,TYPE=C

SEGM  NAME=PAYROLL,BYTES=100,FREQ=1,PARENT=NAME
FIELD NAME=HOURS,BYTES=15,START=51,TYPE=P
FIELD NAME=BASICPAY,BYTES=15,START=1,TYPE=P

DBDGEN
FINISH
END
```

## *HDAM* の *DBD* 生成の例

以下の例は、スキル・インベントリーのデータ構造を *HDAM* データベースとして定義するときに必要なステートメントです。最初の例は、階層ポインターを使用するデータベースを定義するもので、2 番目の例は、物理子ポインターと物理兄弟ポインターを使用するデータベースを定義するものです。3 番目の例は、*VERSION=* および *EXIT=* パラメーターを使用するデータベースを定義するものです。

### 階層ポインターを持つスキル・インベントリー *SKILLINV* データベースの *HDAM* の *DBD* 生成

```
DBD  NAME=SKILLINV,ACCESS=HDAM,RMNAME=(DFSHDC40,20,500,824)
DATASET DD1=SKILHDAM,BLOCK=4096,SCAN=0

SEGM  NAME=SKILL,BYTES=31,PTR=H,PARENT=0
FIELD NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD NAME=STDCODE,BYTES=10,START=22,TYPE=C

SEGM  NAME=NAME,BYTES=20,PTR=H,PARENT=SKILL
FIELD NAME=(STDCLEVEL,SEQ,U),BYTES=20,START=1,TYPE=C

SEGM  NAME=EXPR,BYTES=20,PTR=H,PARENT=NAME
FIELD NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD NAME=CLASSIF,BYTES=10,START=11,TYPE=C

SEGM  NAME=EDUC,BYTES=75,PTR=H,PARENT=NAME
FIELD NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD NAME=SCHOOL,BYTES=65,START=11,TYPE=C

DBDGEN
FINISH
END
```

### 物理子ポインターと物理兄弟ポインターを持つスキル・インベントリー・データベースの *HDAM* の *DBD* 生成

```
DBD  NAME=SKILLINV,ACCESS=HDAM,RMNAME=(DFSHDC40,20,500,824)
DATASET DD1=SKILHDAM,BLOCK=4096,SCAN=0

SEGM  NAME=SKILL,BYTES=31,PTR=T,PARENT=0
FIELD NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD NAME=STDCODE,BYTES=10,START=22,TYPE=C
```

```

SEGM  NAME=NAME,BYTES=20,PTR=T,PARENT=((SKILL,SNGL))
FIELD  NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C

SEGM  NAME=EXPR,BYTES=20,PTR=T,PARENT=((NAME,SNGL))
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C

SEGM  NAME=EDUC,BYTES=75,PTR=T,PARENT=((NAME,SNGL))
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C

DBDGEN
FINISH
END

```

**EXIT=** と **VERSION=** パラメーターを持つスキル・インベントリ **SKILLINV** データベースの **HDAM** の **DBD** 生成

```

DBD    NAME=SKILLINV,ACCESS=HDAM,RMNAME=(DFSHDC40,20,500,824),VERSION=CCCCC
DATASET DD1=SKILHDAM,BLOCK=4096,SCAN=0

```

```

SEGM  NAME=A,BYTES=8,PTR=H,PARENT=0,EXIT=(EXITA)
FIELD  NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD  NAME=STDCODE,BYTES=10,START=22,TYPE=C

```

```

SEGM  NAME=B,BYTES=20,PTR=H,PARENT=SKILL,(EXIT=(EXITB,(CASCADE,KEY)))
FIELD  NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C

```

```

SEGM  NAME=C,BYTES=8,PTR=H,PARENT=A,EXIT=((EXITA,PATH),(EXITC))
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C

```

```

SEGM  NAME=EDUC,BYTES=75,PTR=H,PARENT=NAME
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C

```

```

DBDGEN
FINISH
END

```

## HIDAM の DBD 生成の例

HIDAM データベースは、ルート・セグメント・タイプのシーケンス・フィールドで索引付けされています。HIDAM データベースと HIDAM 1 次索引データベースを定義する際、HIDAM ルート・セグメント・タイプと HIDAM 1 次索引データベースに定義されているセグメント・タイプとの間に索引関係が設定されます。以下の例は、HIDAM ルート・セグメント・タイプと HIDAM 1 次索引データベース中の索引セグメント・タイプの間索引関係を確立するために必要なステートメントをまとめたものです。索引関係に関するオペランドのみを示してあります。

### HIDAM 1 次索引関係

HIDAM:	INDEX:
DBD NAME=dbd1,ACCESS=HIDAM	DBD NAME=dbd2,ACCESS=INDEX
SEGM NAME=seg1,BYTES=, POINTER=	SEGM NAME=seg2,BYTES=
LCHILD NAME=(seg2,dbd2), PTR=INDX	LCHILD NAME=(seg1,dbd1), INDEX=f1d1

```
FIELD NAME=(f1d1,SEQ,U),      FIELD NAME=(f1d2,SEQ,U),
BYTES=,START=                  BYTES=,START=
```

次の例は、スキル・インベントリーのデータ構造を 2 つの HIDAM データベースとして定義するステートメントを示しています。最初の例は、階層ポインターで定義されており、2 番目の例は物理子ポインターおよび物理兄弟ポインターで定義されています。HIDAM データベースはルート・セグメント・タイプのシーケンス・フィールドで索引付けされているので、INDEX の DBD 生成が必要です。次の例は、2 つの HIDAM の DBD 生成と索引の DBD 生成のためのステートメントを示したものです。

#### **HIDAM** データベース **SKILLINV** の **INDEX** の **DBD** 生成

```
DBD  NAME=INDEXDB,ACCESS=INDEX
DATASET DD1=INXDDB1,
SEGM  NAME=INDEX,BYTES=21,FREQ=10000
LCHILD NAME=(SKILL,SKILLINV),INDEX=SKILL
FIELD NAME=(INDXSEQ,SEQ,U),BYTES=21,START=1
DBDGEN
FINISH
END
```

#### 階層ポインターを持つスキル・インベントリー・データベースの **HIDAM** の **DBD** 生成

```
DBD  NAME=SKILLINV,ACCESS=HIDAM
DATASET DD1=SKLHIDAM,BLOCK=4096,SCAN=0

SEGM  NAME=SKILL,BYTES=31,PTR=H,PARENT=0
FIELD NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD NAME=STDCODE,BYTES=10,START=22,TYPE=C
LCHILD NAME=(INDEX,INDEXDB),PTR=INDX

SEGM  NAME=NAME,BYTES=20,PTR=H,PARENT=SKILL
FIELD NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C

SEGM  NAME=EXPR,BYTES=20,PTR=H,PARENT=NAME
FIELD NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD NAME=CLASSIF,BYTES=10,START=11,TYPE=C

SEGM  NAME=EDUC,BYTES=75,PTR=H,PARENT=NAME
FIELD NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD NAME=SCHOOL,BYTES=65,START=11,TYPE=C

DBDGEN
FINISH
END
```

#### 物理子ポインターと物理兄弟ポインターを持つスキル・インベントリー **SKILLINV** データベースの **HIDAM** の **DBD** 生成

```
DBD  NAME=SKILLINV,ACCESS=HIDAM
DATASET DD1=SKLHIDAM,BLOCK=4096,SCAN=0

SEGM  NAME=SKILL,BYTES=31,PTR=T,PARENT=0
LCHILD NAME=(INDEX,INDEXDB),PTR=INDX

FIELD NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD NAME=STDCODE,BYTES=10,START=22,TYPE=C

SEGM  NAME=NAME,BYTES=20,PTR=T,PARENT=((SKILL,SNGL))
FIELD NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C
```

```

SEGM  NAME=EXPR,BYTES=20,PTR=T,PARENT=((NAME,SNGL))
FIELD NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD NAME=CLASSIF,BYTES=10,START=11,TYPE=C

```

```

SEGM  NAME=EDUC,BYTES=75,PTR=T,PARENT=((NAME,SNGL))
FIELD NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD NAME=SCHOOL,BYTES=65,START=11,TYPE=C

```

```

DBDGEN
FINISH
END

```

## PHDAM の DBD 生成の例

次の例は、PHDAM データベースの、物理子ポインターと物理兄弟ポインターを持つスキル・インベントリー・データベースの DBD 生成を示しています。

```

DBD    NAME=SKILLINV,ACCESS=(PHDAM,OSAM),RMNAME=(DFSHDC40,20,500,824)
SEGM   NAME=SKILL,BYTES=31,PTR=T,PARENT=0
FIELD  NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD  NAME=STDCODE,BYTES=10,START=22,TYPE=C
SEGM   NAME=NAME,BYTES=20,PTR=T,PARENT=((SKILL,SNGL))

FIELD  NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C
SEGM   NAME=EXPR,BYTES=20,PTR=T,PARENT=((NAME,SNGL))
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C
SEGM   NAME=EDUC,BYTES=75,PTR=T,PARENT=((NAME,SNGL))
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C
DBDGEN
END

```

## PHIDAM の DBD 生成の例

次の例は、PHIDAM データベースの、物理子ポインターと物理兄弟ポインターを持つスキル・インベントリー・データベースの DBD 生成を示しています。索引基本定義は必要ありません。

```

DBD    NAME=SKILLINV,ACCESS=PHIDAM
SEGM   NAME=SKILL,BYTES=31,PTR=T,PARENT=0
FIELD  NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD  NAME=STDCODE,BYTES=10,START=22,TYPE=C
SEGM   NAME=NAME,BYTES=20,PTR=T,PARENT=((SKILL,SNGL))
FIELD  NAME=(STDCLEVL,SEQ,U),BYTES=20,START=1,TYPE=C
SEGM   NAME=EXPR,BYTES=20,PTR=T,PARENT=((NAME,SNGL))
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C
SEGM   NAME=EDUC,BYTES=75,PTR=T,PARENT=((NAME,SNGL))
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C
DBDGEN
FINISH
END

```

## GSAM の DBD 生成の例

次の例は、GSAM データベースの入力および出力データ・セットを定義する DBD 生成ステートメントを示します。 NONE:





```

DBD    NAME=MSDBLM06,ACCESS=MSDB
DATASET REL=(DYNAMIC,FIELDLDM)
SEGM   NAME=LDM,BYTES=52
FIELD  NAME=FIELDSEQ,BYTES=4,START=1,TYPE=C
FIELD  NAME=FIELDX01,BYTES=2,START=5,TYPE=X
FIELD  NAME=FIELDX01,BYTES=2,START=5,TYPE=X
FIELD  NAME=FIELDX01,BYTES=2,START=5,TYPE=X
FIELD  NAME=FIELDH01,BYTES=2,START=7,TYPE=H
FIELD  NAME=FIELDH01,BYTES=2,START=7,TYPE=H
FIELD  NAME=FIELDH01,BYTES=2,START=7,TYPE=H
FIELD  NAME=FIELDF01,BYTES=4,START=9,TYPE=F
FIELD  NAME=FIELDF01,BYTES=4,START=9,TYPE=F
FIELD  NAME=FIELDF01,BYTES=4,START=9,TYPE=F
FIELD  NAME=FIELDH01,BYTES=2,START=13,TYPE=H
FIELD  NAME=FIELDH01,BYTES=2,START=13,TYPE=H
FIELD  NAME=FIELDH01,BYTES=2,START=13,TYPE=H
FIELD  NAME=FIELDP01,BYTES=2,START=13,TYPE=P
FIELD  NAME=FIELDP01,BYTES=2,START=13,TYPE=P
FIELD  NAME=FIELDP01,BYTES=2,START=13,TYPE=P
FIELD  NAME=FIELDP01,BYTES=2,START=13,TYPE=P
FIELD  NAME=FIELDP01,BYTES=2,START=13,TYPE=P
FIELD  NAME=FIELDP01,BYTES=2,START=13,TYPE=P
FIELD  NAME=FIELDH02,BYTES=2,START=32,TYPE=H
FIELD  NAME=FIELDH02,BYTES=2,START=32,TYPE=H
FIELD  NAME=FIELDH02,BYTES=2,START=32,TYPE=H
FIELD  NAME=FIELDF02,BYTES=4,START=34,TYPE=F
FIELD  NAME=FIELDF02,BYTES=4,START=34,TYPE=F
FIELD  NAME=FIELDF02,BYTES=4,START=34,TYPE=F
FIELD  NAME=FIELDX03,BYTES=12,START=38,TYPE=X
FIELD  NAME=FIELDX03,BYTES=12,START=38,TYPE=X
FIELD  NAME=FIELDX03,BYTES=12,START=38,TYPE=X
FIELD  NAME=FIELDH03,BYTES=2,START=50,TYPE=H
DBDGEN
FINISH
END

```

## 高速処理データベースの DBD 生成の例

次の例は、高速処理データベースの DBD を定義するために必要な DBD 生成ステートメントです。

```

DEDB1    DBD    NAME=DEDB0001,ACCESS=DEDB,RMNAME=RMOD1
AREA0    AREA  DD1=DB1AREA0,          AREA 0
           SIZE=1024,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA1    AREA  DD1=DB1AREA1,          AREA 1
           SIZE=1024,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA2    AREA  DD1=DB1AREA2,          AREA 2
           SIZE=1024,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA3    AREA  DD1=DB1AREA3,          AREA 3
           SIZE=4096,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA4    AREA  DD1=DB1AREA4,          AREA 4
           MODEL=1,SIZE=2048,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA5    AREA  DD1=DB1AREA5,          AREA 5
           SIZE=4096,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA6    AREA  DD1=DB1AREA6,          AREA 6
           SIZE=1024,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
AREA7    AREA  DD1=DB1AREA7,          AREA 7
           SIZE=2048,
           ROOT=(10,5),          5 UOW'S/AREA
           UOW=(15,10)          5 A.P.'S + 10 DEP. OFLOW.
ROOTSEG  SEGM  NAME=ROOTSEG1,PARENT=0,BYTES=(300,50)
ROOTFLD  FIELD NAME=(ROOTKEY1,SEQ,U),BYTES=8,START=3,TYPE=C
SDSEG    SEGM  NAME=SDSEGNM1,PARENT=ROOTSEG1,BYTES=(300,50),
           TYPE=SEQ
SDFLD    FIELD NAME=SDSCFLD1,BYTES=10,START=3,TYPE=C
DDSEG    SEGM  NAME=DDSEGNM1,PARENT=ROOTSEG1,
           BYTES=(40,15),TYPE=DIR
DDFLD1   FIELD NAME=(DD1FLD1,SEQ,U),BYTES=4,START=6

```

```
DDFLD2    FIELD  NAME=DD1FLD2,BYTES=5,START=10,TYPE=P
          DBDGEN
          FINISH
          END
```

## DEDB のサブセット・ポインタの DBD 生成の例

次の例は、サブセット・ポインタ付き DEDB を定義するために必要な DBD 生成ステートメントの例です。

```
DBD NAME=DEDBDB,ACCESS=DEDB,RMNAME=DBFHD040
AREA DD1=DEDBDD,SIZE=1024,
      ROOT=(10,5),UOW=(15,10)
SEGM NAME=A,BYTES=(48,27),PARENT=0
FIELD NAME=(A1,SEQ,U),BYTES=10,START=3,TYPE=C
SEGM NAME=B,BYTES=(24,11),PARENT=((A,SNGL)),TYPE=DIR,SSPTR=5
FIELD NAME=(B1,SEQ,U),BYTES=5,START=3,TYPE=C
FIELD NAME=B2,BYTES=5,START=10,TYPE=C
SEGM NAME=C,BYTES=(34,32),PARENT=((B,DBLE)),RULES=(,HERE),TYPE=DIR
FIELD NAME=(C1,SEQ,U),BYTES=20,START=3,TYPE=C
SEGM NAME=D,BYTES=(52,33),PARENT=((A,DBLE)),TYPE=DIR,SSPTR=3
FIELD NAME=(D1,SEQ,U),BYTES=2,START=3,TYPE=C
SEGM NAME=B,BYTES=(52,33),PARENT=((A,DBLE)),RULES=(,FIRST),TYPE=DIR
FIELD NAME=(B1,SEQ,U),BYTES=2,START=3,TYPE=C
DBDGEN
FINISH
END
```

注: SSPTR=n (n はサブセット・ポインタの個数を表します。)

## 論理関係のある例

定義できる論理関係のタイプには、単一方向、物理対の両方向、仮想対の両方向の 3 つがあります。

次の図は、IMS データベースで定義できる 3 タイプの論理関係を示しています。この図に続く各表では、各タイプの関係を定義するために必要なステートメントを示します。示されているのはこの関係に関連したオペランドのみで、各タイプの関係は、DBD1 および DBD2 という名前の 2 つのデータベースのセグメント間で定義されていると見なされています。

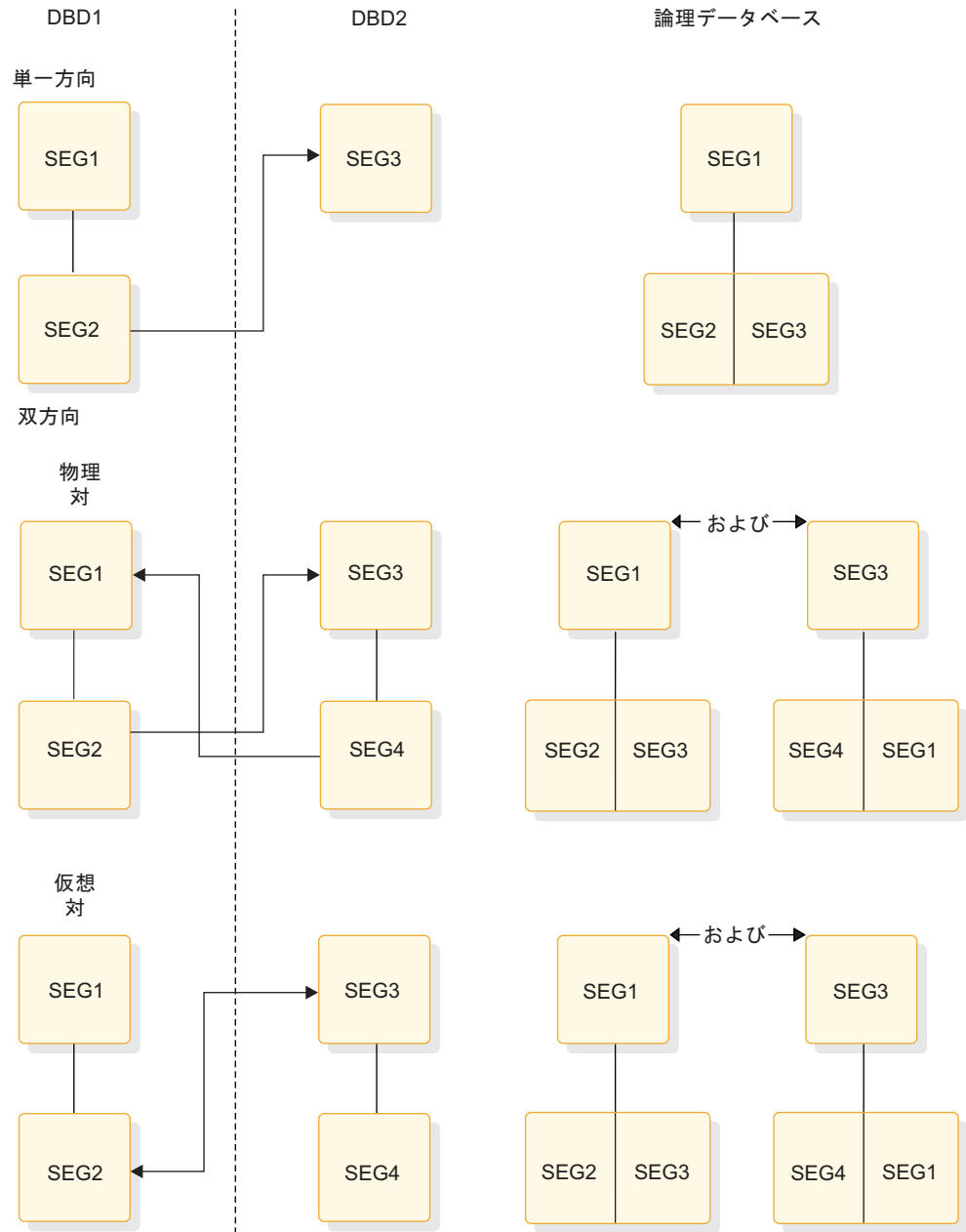


図 7. 単一方向論理関係、物理対の両方向論理関係、および仮想対の両方向論理関係の比較

次の各表では、各タイプ関係を定義するために必要なステートメントを示します。示されているのはこの関係に関連したオペランドのみで、各タイプ関係は、DBD1 および DBD2 という名前の 2 つのデータベースのセグメント間で定義されていると見なされています。

表 12. 単一方向論理関係を定義するために必要なステートメント

DBD1 用のステートメント	DBD2 用のステートメント
<pre>SEGM NAME=SEG1,PARENT= ,BYTES=,FREQ= ,POINTER=,RULES=</pre>	<pre>SEGM NAME=SEG3,PARENT= ,BYTES=,FREQ=,POINTER= ,RULES=</pre>

表 12. 単一方向論理関係を定義するために必要なステートメント (続き)

DBD1 用のステートメント	DBD2 用のステートメント
<pre> SEGMENT NAME=SEG2   ,PARENT=((SEG1,)   ,SEG3,PHYSICAL,DBD2))<sup>1</sup>   ,BYTES=,FREQ=   ,POINTER=(LPARNT)<sup>1</sup>   ,RULES=           </pre>	<pre> LCHILD NAME=(SEG2,DBD1)           </pre>
<p>注:</p> <p>1. シンボリック・ポインターまたは直接論理親ポインターを指定します。直接アクセス・ポインターを指定できるのは、論理親が<sup>3</sup> HDAM、HIDAM、PHDAM、または HIDAM データベース内にある場合のみです。</p>	

表 13. 物理対の両方向論理関係を定義するために必要なステートメント

DBD1 用のステートメント	DBD2 用のステートメント
<pre> SEGMENT NAME=SEG1,PARENT=   ,BYTES=,FREQ=,   ,POINTER=,RULES= LCHILD NAME=(SEG4,DBD2)   ,PAIR=SEG2 SEGMENT NAME=SEG2   ,PARENT=((SEG1,)   ,(SEG3,PHYSICAL,DBD2))<sup>1</sup>   ,BYTES=,FREQ=   ,POINTER=(LPARNT,PAIRED)<sup>1</sup>   ,RULES=           </pre>	<pre> SEGMENT NAME=SEG3,PARENT=   ,BYTES=,FREQ=   ,POINTER=,RULES= LCHILD NAME=(SEG2,DBD1)   ,PAIR=SEG4 SEGMENT NAME=SEG4   ,PARENT=((SEG3,)   ,(SEG1,PHYSICAL,DBD1))<sup>1</sup>   ,BYTES=,FREQ=   ,POINTER=(LPARNT,PAIRED)<sup>1</sup>   ,RULES=           </pre>
<p>注:</p> <p>1. シンボリック・ポインターまたは直接論理親ポインターを指定します。直接アクセス・ポインターを指定できるのは、論理親が<sup>3</sup> HDAM、HIDAM、PHDAM、または PHIDAM データベース内にある場合のみです。</p>	

表 14. 仮想対の両方向論理関係を定義するために必要なステートメント

DBD1 用のステートメント	DBD2 用のステートメント
<pre> SEGMENT NAME=SEG1,PARENT=   ,BYTES=,FREQ=   ,POINTER=,RULES=           </pre>	<pre> SEGMENT NAME=SEG3,PARENT=   ,BYTES=,FREQ=   ,POINTER=,RULES=  SEGMENT NAME=SEG4   ,PARENT=SEG3   ,POINTER=PAIRED   ,SOURCE=((SEG2,DATA,DBD1))           </pre>
<pre> SEGMENT NAME=SEG2   ,PARENT=((SEG1,)   ,(SEG3,PHYSICAL,DBD2))<sup>1</sup>   ,BYTES=,FREQ=   ,POINTER=(LTWIN,LPARNT)<sup>2</sup>   ,RULES=           </pre>	<pre> LCHILD NAME=(SEG2,DBD1)   ,POINTER=SNGL<sup>3</sup>   ,PAIR=SEG4   ,RULES=<sup>3</sup>           </pre>

表 14. 仮想対の両方向論理関係を定義するために必要なステートメント (続き)

DBD1 用のステートメント

DBD2 用のステートメント

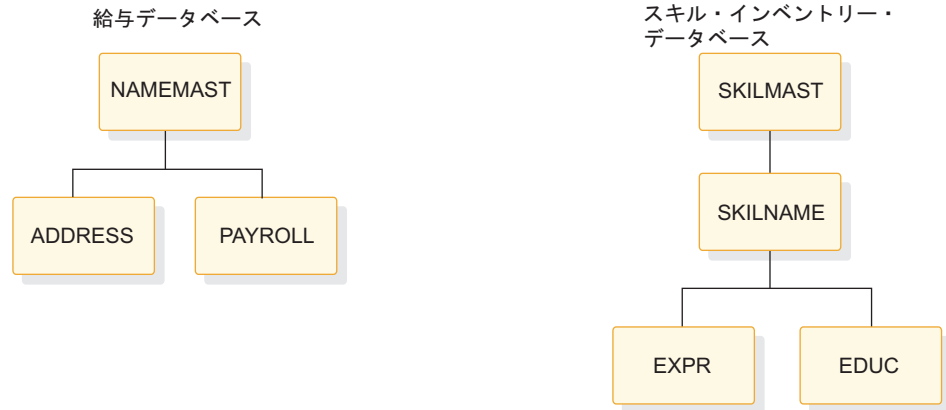
注:

1. シンボリック・ポインターまたは直接論理親ポインターを指定します。直接アクセス・ポインターを指定できるのは、論理親が HDAM、HIDAM、PHDAM、または PHIDAM データベース内にある場合のみです。
2. 論理兄弟ポインターには、LTWIN または LTWINBWD を指定してください。
3. 論理子ポインターには DNGL または DBLE を指定してください。パラメーター LCHILD RULES= は、仮想論理子にシーケンス・フィールドが定義されていなく、固有のシーケンス・フィールドが定義されていないとき、または、論理子セグメントがないときに使用します。

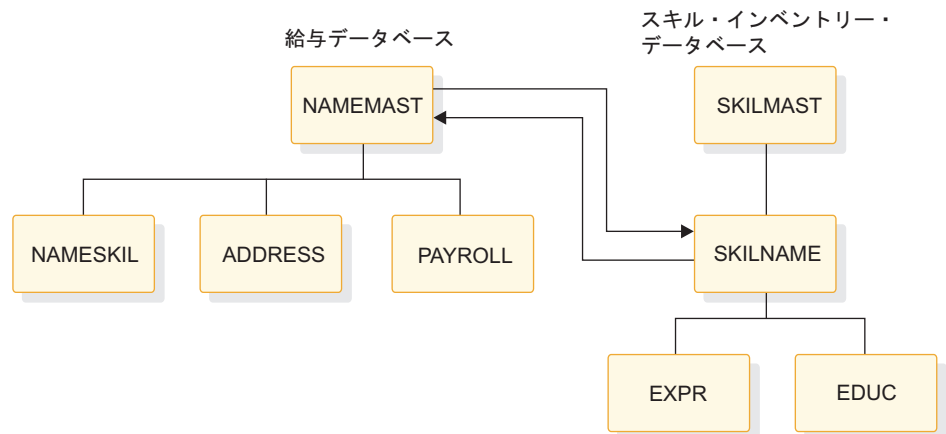
次の図の仮想対の両方向論理関係区域では、HISAM データベースが仮想対の論理関係に参加できるのは、実論理子が HDAM、HIDAM、PHDAM、または PHIDAM データベース内にあり、かつその論理親が HISAM データベース内にあるときのみに限られます。

次の図は、論理関係と論理データベースを定義する方法を例示したものです。パート 1 は、給与データベースおよびスキル・インベントリー・データベースの物理データ構造を表しています。パート 2 は、物理データ構造間の NAMEMAST (給与データベース内) と SKILNAME (スキル・インベントリー・データベース内) の論理関係を表しています。パート 3 は、論理関係の結果定義できる論理データベース (SKILL と NAME) を表しています。この新規データベースは、NAMEMAST 構造と SKILNAME 構造の両方からのセグメントを含みます。DBD 生成ステートメントの例を、図の後に示してあります。

## パート 1: 物理データベース



## パート 2: 論理関係



## パート 3: 論理データベース

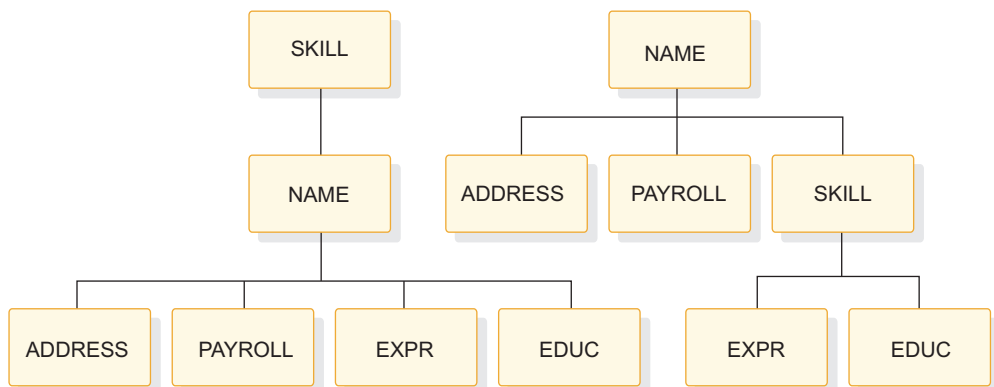


図 8. 物理データベース間の論理関係とその結果定義できる論理データベース

## DBD 生成ステートメントの例

以下の例は、以下のものを定義するときに必要な DBD 生成ステートメントです。

- 給与計算インベントリーおよびスキル・インベントリーのデータ構造 (2 つのデータベース間に仮想対の双方向論理関係を持つ、HIDAM および HDAM データベースとして、前述の図のパート 2 に示されています。)
- 論理データ構造 (前述の図のパート 3 で論理データベースとして示されています。)

```

DBD    NAME=PAYROLDB,ACCESS=HIDAM
DATASET DD1=PAYHIDAM,BLOCK=4096,SCAN=0
SEGM   NAME=NAMEMAST,PTR=TWINBWD,RULES=(VVV),           X
        BYTES=150
LCHILD NAME=(INDEX,INDEXDB),PTR=INDX
LCHILD NAME=(SKILNAME,SKILLINV),PAIR=NAMESKIL,PTR=DBLE
FIELD  NAME=(EMPLOYEE,SEQ,U),BYTES=60,START=1,TYPE=C
FIELD  NAME=MANNBR,BYTES=15,START=61,TYPE=C
FIELD  NAME=ADDR,BYTES=75,START=76,TYPE=C
SEGM   NAME=NAMESKIL,PARENT=NAMEMAST,PTR=PAIRED,       X
        SOURCE=((SKILNAME,DATA,SKILLINV))
FIELD  NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD  NAME=STDLEVL,BYTES=20,START=22,TYPE=C
SEGM   NAME=ADDRESS,BYTES=200,PARENT=NAMEMAST
FIELD  NAME=(HOMEADDR,SEQ,U),BYTES=100,START=1,TYPE=C
FIELD  NAME=COMAILOC,BYTES=100,START=101,TYPE=C
SEGM   NAME=PAYROLL,BYTES=100,PARENT=NAMEMAST
FIELD  NAME=(BASICPAY,SEQ,U),BYTES=15,START=1,TYPE=P
FIELD  NAME=HOURS,BYTES=15,START=51,TYPE=P
DBDGEN
FINISH
END

DBD    NAME=SKILLINV,ACCESS=HDAM,RMNAME=(DFSHDC40,20,500,824)
DATASET DD1=SKILHDAM,BLOCK=4096,SCAN=0
SEGM   NAME=SKILMAST,BYTES=31,PTR=TWINBWD
FIELD  NAME=(TYPE,SEQ,U),BYTES=21,START=1,TYPE=C
FIELD  NAME=STDCODE,BYTES=10,START=22,TYPE=C
SEGM   NAME=SKILNAME,                                   X
        PARENT=((SKILMAST,DBLE),(NAMEMAST,P,PAYROLDB)),
        BYTES=80,PTR=(LPARNT,LTWINBWD,TWINBWD),
        RULES=(VVV)                                   X
FIELD  NAME=(EMPLOYEE,SEQ,U),START=1,BYTES=60,TYPE=C
FIELD  NAME=(STDLEVL),BYTES=20,START=61,TYPE=C
SEGM   NAME=EXPR,BYTES=20,PTR=T,                       X
        PARENT=((SKILNAME,SNGL))
FIELD  NAME=PREVJOB,BYTES=10,START=1,TYPE=C
FIELD  NAME=CLASSIF,BYTES=10,START=11,TYPE=C
SEGM   NAME=EDUC,BYTES=75,PTR=T,                       X
        PARENT=((SKILNAME,SNGL))
FIELD  NAME=GRADLEVL,BYTES=10,START=1,TYPE=C
FIELD  NAME=SCHOOL,BYTES=65,START=11,TYPE=C
DBDGEN
FINISH
END

DBD    NAME=LOGICDB,ACCESS=LOGICAL
DATASET LOGICAL
SEGM   NAME=SKILL,SOURCE=((SKILMAST,,SKILLINV))
SEGM   NAME=NAME,PARENT=SKILL,                         X
        SOURCE=((SKILNAME,,SKILLINV),(NAMEMAST,,PAYROLDB))
SEGM   NAME=ADDRESS,PARENT=NAME,SOURCE=((ADDRESS,,PAYROLDB))
SEGM   NAME=PAYROLL,PARENT=NAME,SOURCE=((PAYROLL,,PAYROLDB))
SEGM   NAME=EXPR,PARENT=NAME,SOURCE=((EXPR,,SKILLINV))
SEGM   NAME=EDUC,PARENT=NAME,SOURCE=((EDUC,,SKILLINV))
DBDGEN
FINISH
END

BD     NAME=LOGIC1,ACCESS=LOGICAL
DATASET LOGICAL
SEGM   NAME=NAME,SOURCE=((NAMEMAST,,PAYROLDB))
SEGM   NAME=ADDRESS,PARENT=NAME,SOURCE=((ADDRESS,,PAYROLDB))
SEGM   NAME=PAYROLL,PARENT=NAME,SOURCE=((PAYROLL,,PAYROLDB))
SEGM   NAME=SKILL,PARENT=NAME,                         X
        SOURCE=((NAMESKIL,,PAYROLDB),(SKILMAST,,SKILLINV))
SEGM   NAME=EXPR,SOURCE=((EXPR,,SKILLINV)),PARENT=SKILL


```

```

SEGM  NAME=EDUC,SOURCE=((EDUC,,SKILLINV)),PARENT=SKILL
DBDGEN
FINISH
END

```

関連概念:

 論理関係の作成 (データベース管理)

## 副次索引のある例

以下の各例では、索引付きデータベースのセグメント・タイプと 2 次索引データベースのセグメント・タイプとの間に 2 次索引関係を確認するために必要なステートメントを示します。

索引ターゲット・セグメント・タイプと索引ソース・セグメント・タイプが同じ場合に必要となるステートメントは、以下の表に示してあります。

表 15. 索引ソース・セグメント・タイプと索引ターゲット・セグメント・タイプが同じである場合

索引付き DBD	索引 DBD
DBD NAME=DBD1,ACCESS=	DBD NAME=DBD2,ACCESS=INDEX
.	.
.	.
SEGM NAME <sup>1</sup> =SEG1,PARENT=	SEGM NAME=SEG3,PARENT=0,BYTES=
,BYTES	
FIELD NAME=(FLD2,SEQ,...),BYTES=	FIELD NAME=(FLD2,SEQ,...),BYTES=
FIELD NAME=FLD1,BYTES=	,START=1
,START	
LCHILD NAME=(SEG3,DBD2),	LCHILD NAME=(SEG1,DBD1),
POINTER <sup>2</sup> =INDX	INDEX=XDNAME,POINTER <sup>2</sup> =SNGL
XDFLD NAME=XDNAME,SRCH=FLD1	

注:

1. 索引ターゲット・セグメント・タイプは、ルート・セグメント・タイプでも従属セグメント・タイプでも構いませんが、論理子セグメント・タイプまたは論理子セグメント・タイプの従属にしてはなりません。索引ソース・セグメント・タイプは、論理子セグメント・タイプにしてはなりません。
2. 例では、索引 DBD の索引ポインター・セグメント・タイプの直接ポインターで示されています。シンボリック・ポインター設定が必要な場合には、POINTER=SYMB を両方の LCHILD ステートメントに指定してください。索引ターゲット・セグメント・タイプが HISAM データベース内にある場合にはシンボリック・ポインター設定が必要です。

以下の表では、索引ターゲット・セグメント・タイプと索引ソース・セグメント・タイプが異なっています。この表と前述の表では、副次索引関係に関連したオペランドのみが示されています。



表 16. 索引ソース・セグメント・タイプと索引ターゲット・セグメント・タイプが異なる場合

索引付き DBD	索引 DBD
DBD NAME=DBD1,ACCESS= . .	DBD NAME=DBD2,ACCESS=INDEX . .
SEGM NAME <sup>1</sup> =SEG1,BYTES=,PARENT=  LCHILD NAME=(SEG4,DBD2), POINTER <sup>2</sup> =INDX	SEGM NAME=SEG4,PARENT=0,BYTES=  FIELD NAME=(FLD4,SEQ,...) ,START=1,BYTES=
XDFLD NAME=XDNAME,SEGMENT=SEG3, SRCH=FLD3,...	LCHILD NAME=(SEG1,DBD1), INDEX=XDNAME,POINTER <sup>2</sup> =SNGL
SEGM NAME=SEG2,BYTES= ,PARENT=SEG1	
SEGM NAME <sup>1</sup> =SEG3 ,PARENT=SEG2	
FIELD NAME=FLD3,BYTES= ,START=	

注:

1. 索引ターゲット・セグメント・タイプは、ルート・セグメント・タイプでも従属セグメント・タイプでも構いません。しかし、論理子セグメント・タイプまたは論理子セグメント・タイプの従属にはなりません。索引ソース・セグメント・タイプは、論理子セグメント・タイプにはなりません。
2. 例では、索引 DBD の索引ポインター・セグメント・タイプの直接ポインターで示されています。シンボリック・ポインター設定が必要な場合には、POINTER=SYMB を両方の LCHILD ステートメントに指定してください。索引ターゲット・セグメント・タイプが HISAM データベース内にある場合にはシンボリック・ポインター設定が必要です。

### 全機能副次索引データベースの DBD の例

以下の図は、2 つの副次索引データベースで索引付けされたデータベース DTA1 を示しています。最初の副次索引 X1 では、索引ターゲット・セグメントおよび索引ソース・セグメントについて同じセグメントを使用し、2 番目の副次索引 X2 では、索引ソース・セグメントと索引ターゲット・セグメントとが異なっています。

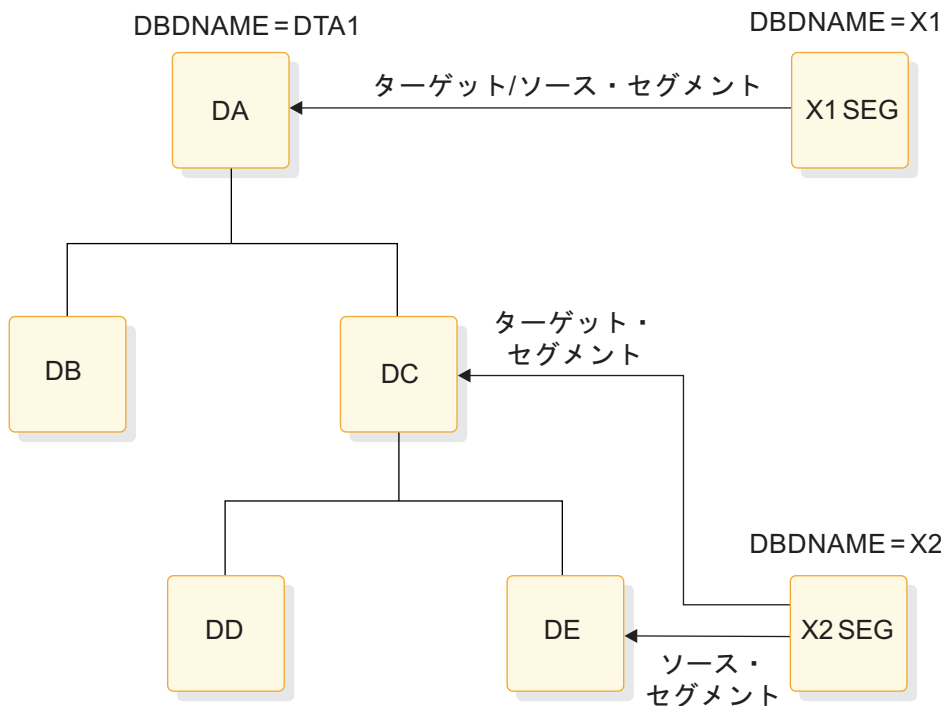


図 9. 2 つの副次索引による索引付きデータベース

次の図は、索引付きデータベースを定義する DBD 生成ステートメントです。

```

DBD    NAME=DTA1,ACCESS=HDAM,RMNAME=(DFSHDC40,20,500,824)
DATASET DD1=D1
SEGM   NAME=DA,PARENT=0,BYTES=15
FIELD  NAME=(DAF1,SEQ),BYTES=5,START=1
LCHILD NAME=(X1SEG,X1),PTR=INDX
XDFLD  NAME=DAF1X,SRCH=DAF1
SEGM   NAME=DB,PARENT=DA,BYTES=20
FIELD  NAME=(DBF1,SEQ),BYTES=5,START=1
SEGM   NAME=DC,PARENT=DA,BYTES=20
FIELD  NAME=(DCF1,SEQ),BYTES=5,START=1
LCHILD NAME=(X2SEG,X2),PTR=SYMB
XDFLD  NAME=DCF1X,SRCH=DEF1,SEGMENT=DE
SEGM   NAME=DD,PARENT=DC,BYTES=25
FIELD  NAME=(DDF1,SEQ),BYTES=5,START=1
SEGM   NAME=DE,PARENT=DC,BYTES=25
FIELD  NAME=(DEF1,SEQ),BYTES=5,START=1
DBDGEN
FINISH
END

```

次の図は、副次索引データベース X1 を定義する DBD 生成ステートメントです。

```

DBD    NAME=X1,ACCESS=INDEX
DATASET DD1=X1P
SEGM   NAME=X1SEG,BYTES=5,PARENT=0
FIELD  NAME=(X1F1,SEQ,U),START=1,BYTES=5
LCHILD NAME=(DA,DTA1),INDEX=DAF1X,POINTER=SNGL
DBDGEN
FINISH
END

```

次の図は、副次索引データベース X2 を定義する DBD 生成ステートメントです。

```

DBD    NAME=X2,ACCESS=INDEX
DATASET DD1=X2P
SEGM   NAME=X2SEG,BYTES=5,PARENT=0

```

```

FIELD  NAME=(X2F1,SEQ,U),START=1,BYTES=5
LCHILD NAME=(DC,DTA1),INDEX=DCF1X,POINTER=SYMB
DBDGEN
FINISH
END

```

## 高速機能副次索引データベースの DBD の例

次の例は、固有キー・ポインター・セグメントを使用する HISAM 副次索引データベース DBD を示しています。

```

DBDX  DBD      NAME=NAMEXDB,ACCESS=(INDEX,VSAM),FPINDEX=YES
      DATASET DD1=NAMEKSDS
      SEGM     NAME=NAMEXSEG,PARENT=0,BYTES=15
      FIELD   NAME=(NAMEKEY,SEQ,U),BYTES=10,START=1
      LCHILD  NAME=(COURSE,EDUCDB),INDEX=NAMEINDX,PTR=SYMB
      DBDGEN
      FINISH
      END

```

次の例は、非固有キー・ポインター・セグメントを使用する HISAM 副次索引データベース DBD を示しています。

```

DBDX  DBD      NAME=NAMEXDB,ACCESS=(INDEX,VSAM),FPINDEX=YES
      DATASET DD1=NAMEKSDS,OVFLW=NAMEESDS
      SEGM     NAME=NAMEXSEG,PARENT=0,BYTES=15
      FIELD   NAME=(NAMEKEY,SEQ,M),BYTES=10,START=1
      LCHILD  NAME=(COURSE,EDUCDB),INDEX=NAMEINDX,PTR=SYMB
      DBDGEN
      FINISH
      END

```

次の例は、固有キー・ポインター・セグメントを使用する SHISAM 副次索引データベース DBD を示しています。

```

DBDX  DBD      NAME=NAMEXDB,ACCESS=(INDEX,SHISAM),FPINDEX=YES
      DATASET DD1=NAMEKSDS
      SEGM     NAME=NAMEXSEG,PARENT=0,BYTES=15
      FIELD   NAME=(NAMEKEY,SEQ,U),BYTES=10,START=1
      LCHILD  NAME=(COURSE,EDUCDB),INDEX=NAMEINDX,PTR=SYMB
      DBDGEN
      FINISH
      END

```

以下の 3 つの例は、副次索引、複数の副次索引セグメント、およびユーザー区画がそれぞれ定義された DEDB DBD 定義を示しています。

次の例は、副次索引が定義された 1 次 DEDB データベース DBD を示しています。

1 次 DEDB EDUCDB データベースには、NAMEXDB、CLASSXDB、および INSTSXDB の 3 つの副次索引データベースが定義されています。NAMEXDB 副次索引のターゲット・セグメントはルート・セグメントです。NAMEXDB のターゲット・セグメントである COURSE セグメントは、ソース・セグメントと同じです。CLASSXDB および INSTSXDB 副次索引のターゲット・セグメントはルート・セグメントではありません。CLASSXDB のターゲット・セグメントである CLASS セグメントは、ソース・セグメントと同じです。INSTSXDB のターゲット・セグメントである INSTRUCT セグメントは、ソース・セグメントである COURSE セグメントと同じではありません。

```

DBD1  DBD      NAME=EDUCDB,ACCESS=DEDB,RMNAME=RMOD3
      AREA    NAME=EDAREA1,SIZE=1024,UOW=(100,10),ROOT=(236,36)

      SEGM    NAME=COURSE,PARENT=0,BYTES=100
      FIELD   NAME=(COURNO,SEQ,U),BYTES=5,START=1
      FIELD   NAME=COURNAME,BYTES=10,START=15

      LCHILD  NAME=(NAMEXSEG,NAMESXDB),PTR=SYMB
      XDFLD   NAME=NAMEINDX,SRCH=COURNAME

      SEGM    NAME=CLASS,BYTES=50,PARENT=COURSE
      FIELD   NAME=(CLASSNO,SEQ,U),BYTES=4,START=7
      FIELD   NAME=CLASNAME,BYTES=10,START=15

      LCHILD  NAME=(CLASXSEG,CLASSXDB),PTR=SYMB
      XDFLD   NAME=CLASINDX,SRCH=CLASNAME

      LCHILD  NAME=(INSTXSEG,INSTSXDB),PTR=SYMB
      XDFLD   NAME=INSTINDX,SEGMENT=INSTRUCT,SRCH=INSTNAME

      SEGM    NAME=INSTRUCT,BYTES=50,PARENT=CLASS
      FIELD   NAME=(INSTNO,SEQ,U),BYTES=6,START=1
      FIELD   NAME=INSTPHNO,BYTES=10,START=11
      FIELD   NAME=INSTNAME,BYTES=20,START=21

      SEGM    NAME=STUDENT,BYTES=50,PARENT=CLASS
      FIELD   NAME=(STUDNO,SEQ,U),BYTES=6,START=1
      FIELD   NAME=STUDPHNO,BYTES=10,START=11
      FIELD   NAME=STUDNAME,BYTES=20,START=21
      FIELD   NAME=ENRLDATE,BYTES=6,START=41

      DBDGEN
      FINISH
      END

```

次の例は、複数の副次索引セグメントが定義された DEDB データベース DBD を示しています。LCHILD ステートメントは、同じ副次索引セグメント名 (2 次データベース内の NAMEXSEG セグメント) および同じ副次索引データベース名 (NAMESXDB データベース) を定義します。XDFLD ステートメントは、同じソース・セグメント (COURSE セグメント) からの異なる検索フィールド (COURNAME および COURSECT) を含む複数の副次索引セグメントに、同じシーケンス・フィールド名 (NAMEINDX) を定義します。

COURSE のターゲット・セグメントは、コース名 (COURNAME) の副次索引またはコース・セクション番号 (COURSECT) の副次索引のいずれかを使用して見つけることができます。

複数の副次索引セグメント (COURNAME および COURSECT) の検索キーの長さは、同一でなければなりません。この例では、いずれも 10 バイトです。

```

DBD1  DBD      NAME=EDUCMDB,ACCESS=DEDB,RMNAME=RMOD3
      AREA    NAME=EDMAREA1,SIZE=1024,UOW=(100,10),ROOT=(236,36)

      SEGM    NAME=COURSE,PARENT=0,BYTES=100
      FIELD   NAME=(COURNO,SEQ,U),BYTES=5,START=1
      FIELD   NAME=COURNAME,BYTES=10,START=15
      FIELD   NAME=COURSECT,BYTES=10,START=25

      LCHILD  NAME=(NAMEXSEG,NAMESXDB),PTR=SYMB,MULTISEG=YES
      XDFLD   NAME=NAMEINDX,SRCH=COURNAME
      LCHILD  NAME=(NAMEXSEG,NAMESXDB),PTR=SYMB,MULTISEG=YES

```

```

XDFLD  NAME=NAMEINDX,SRCH=COURSECT

SEGM   NAME=CLASS,BYTES=50,PARENT=COURSE
FIELD  NAME=(CLASSNO,SEQ,U),BYTES=4,START=7
FIELD  NAME=CLASNAME,BYTES=10,START=15

SEGM   NAME=INSTRUCT,BYTES=50,PARENT=CLASS
FIELD  NAME=(INSTNO,SEQ,U),BYTES=6,START=1
FIELD  NAME=INSTPHNO,BYTES=10,START=11
FIELD  NAME=INSTNAME,BYTES=20,START=21

SEGM   NAME=STUDENT,BYTES=50,PARENT=CLASS
FIELD  NAME=(STUDNO,SEQ,U),BYTES=6,START=1
FIELD  NAME=STUDPHNO,BYTES=10,START=11
FIELD  NAME=STUDNAME,BYTES=20,START=21
FIELD  NAME=ENRLDATE,BYTES=6,START=41
DBDGEN
FINISH
END

```

次の例は、HISAM 副次索引データベースまたは SHISAM 副次索引データベースのユーザー区画化を使用して副次索引が定義された、1 次 DEDB データベース DBD を示しています。LCHILD ステートメントには NAMSXDB1 および NAMSXDB2 の 2 つのユーザー区画が指定されています。

PSELRTN=DBFPSE00 はユーザー区画選択出口です。ユーザー区画選択オプションは PSELOPT=SNGL です。これは、そのユーザー区画グループ内の選択されたユーザー区画データベースのみが 1 次 DEDB データベースへのアクセスに使用されることを示します。PROCSEQD= パラメーターが指定された PCB を使用して、SSA を使用しない後続の修飾 GN 呼び出しが発行されると、選択されたユーザー区画内の最後のポインター・セグメントが 1 次 DEDB データベース内のセグメントへのアクセスに使用された後に、データベースの終わりを示す GB 状況コードがアプリケーションに返されます。

```

DBD1  DBD  NAME=EDUCUDB,ACCESS=DEDB,RMNAME=RMOD3
      AREA NAME=EDUAREA1,SIZE=1024,UOW=(100,10),ROOT=(236,36)

      SEGM NAME=COURSE,PARENT=0,BYTES=100
      FIELD NAME=(COURNO,SEQ,U),BYTES=5,START=1
      FIELD NAME=COURNAME,BYTES=10,START=15

      LCHILD NAME=(NAMEXSEG,(NAMESXB1,NAMSXDB2)),PTR=SYMB
      XDFLD  NAME=XNAME,SRCH=COURNAME,PSELRTN=DBFPSE00,PSELOPT=SNGL

      SEGM NAME=CLASS,BYTES=50,PARENT=COURSE
      FIELD NAME=(CLASSNO,SEQ,U),BYTES=4,START=7
      FIELD NAME=CLASNAME,BYTES=10,START=15


      SEGM NAME=INSTRUCT,BYTES=50,PARENT=CLASS
      FIELD NAME=(INSTNO,SEQ,U),BYTES=6,START=1
      FIELD NAME=INSTPHNO,BYTES=10,START=11
      FIELD NAME=INSTNAME,BYTES=20,START=21

      SEGM NAME=STUDENT,BYTES=50,PARENT=CLASS
      FIELD NAME=(STUDNO,SEQ,U),BYTES=6,START=1
      FIELD NAME=STUDPHNO,BYTES=10,START=11
      FIELD NAME=STUDNAME,BYTES=20,START=21
      FIELD NAME=ENRLDATE,BYTES=6,START=41

      DBDGEN
      FINISH
      END

```

関連概念:

 副次索引の作成 (データベース管理)

---

## DBDGEN プロシージャールの実行

DBDGEN プロシージャールの実行は、データベース定義ブロックを生成する 2 ステップのアセンブルおよびバインド・プロシージャールです。システム定義のステージ 2 で、DBDGEN プロシージャールが IMS.PROCLIB ライブラリーに入れられます。

### DBDGEN ユーティリティー用の JCL

次の例は、DBDGEN ユーティリティー用の JCL を示しています。

```
//      PROC MBR=TEMPNAME,SOUT=A,RGN=0M,SYSDA=
//C      EXEC PGM=ASMA90,REGION=&RGN,
//          PARM=(OBJECT,NODECK,NODBCS,
//              'SIZE(MAX,ABOVE)')
//SYSLIB DD DSN=IMS.&SYSDA.SDFSMAC,DISP=SHR
//SYSLIN DD UNIT=SYSDA,DISP=(,PASS),
//          SPACE=(80,(100,100),RLSE),
//          DCB=(BLKSIZE=80,RECFM=F,LRECL=80)
//SYSPRINT DD SYSOUT=&SOUT,DCB=BLKSIZE=1089,
//          SPACE=(121,(300,300),RLSE,,ROUND)
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),
//          SPACE=(CYL,(10,5))
//L      EXEC PGM=IEWL,PARM='XREF,LIST',
//          COND=(0,LT,C),REGION=4M
//SYSLIN DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=&SOUT,DCB=BLKSIZE=1089,
//          SPACE=(121,(90,90),RLSE)
//SYSLMOD DD DISP=SHR,
//          DSN=IMS.&SYSDA.DBDLIB(&MBR)
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),
//          SPACE=(1024,(100,10),RLSE),DISP=(,DELETE)
```

### DBDGEN を呼び出すプロシージャール

DBDGEN に対する要求を処理する場合は、DBD 生成制御ステートメントを作成し、DBDGEN プロシージャールを呼び出す JCL (以下の図に示されている) に追加する必要があります。

```
//DBDGEN JOB
//      EXEC DBDGEN,MBR=
//C.SYSIN DD *

          DBD
          DATASET
          SEGM
          FIELD          DBD generation control statements
          LCHILD
          XDFLD
          DBDGEN
          FINISH
          END

/*
```

### JCL パラメーター

**MBR=**

生成する DBD の名前を指定します。この名前は、DBD ステートメントの NAME= キーワードに指定した最初の名前と同じにしてください。最初のデー

データベース名は、DBD メンバー名になり、共用副次索引の場合には追加名が別名として追加されます。データベース PCB がこの DBD 生成と関係している場合には、DBD ステートメントの NAME= キーワードに指定する名前の 1 つは、データベース PCB ステートメントの DBDNAME= キーワードに使用する名前であればなりません。共用副次索引以外では、データベース PCB ステートメントの DBDNAME= キーワードに使用される名前は、MBR= キーワード値に使用される名前と同じでなければなりません。

#### **RGN=**

この実行の領域サイズを指定します。デフォルトは 256 KB です。

#### **SOUT=**

SYSOUT DD ステートメントに割り当てられるクラスを指定します。

#### **SYS2=**

XRF 複合システムで「オプションの複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS2='IMSA.' のように、パラメーターを後書きピリオドも含めて引用符で囲みます。

#### ステップ C

ステップ C は、アセンブリー・ステップです。このステップには、次の DD ステートメントが必要です。

#### **SYSIN DD**

ステップ C への入力データ・セットを定義します。プロシーチャーを呼び出す際には、これらの DD ステートメントを指定する必要があります。

#### ステップ L

ステップ L は、バインド・ステップです。

例: 次に示すようにバインド EXEC ステートメントの PARM リストに AMODE=31 を追加することによって、デフォルトの AMODE=24、RMODE=24 ではなく、AMODE=31、RMODE=24 を使用して実行できます。

```
//L      EXEC  PGM=IEWL,PARM='XREF,LIST,AMODE=31',  
//              COND=(0,LT,C),REGION=120K
```

AMODE または RMODE に別の値を指定しなければ、デフォルト値が有効になります。常に、RMODE=24 を指定してバインド・ステップを実行する必要があります。このステップには、次の DD ステートメントが必要です。

#### **IMS.DBDLIB DD**

バインダー用の出力区分データ・セット、IMS.DBDLIB を定義します。





---

## 第 3 章 MFS 装置特性テーブル・ユーティリティ (DFSUTB00)

メッセージ形式サービス装置特性テーブル (MFSDCT) ユーティリティ (DFSUTB00) は、IMS システム定義を実行せずに IMS.PROCLIB ライブラリーの記述子メンバーに新しい画面サイズを定義するために使用します。新しい画面サイズの定義は、以前定義された画面サイズに追加されます。

MFSDCT (DFSUTB00) ユーティリティ・プロシージャは、以下のステップで構成されます。

1. DFSUTB00 プログラムが実行され、以下のいくつかの機能が開始されます。DFSUTB00 プログラムは、以下を行います。
  - PROCLIB から 1 つまたは 2 つの記述子メンバーを読み取り、入力として新しい装置記述子だけを使用する。
  - それぞれの装置記述子について、DCTENTRY ステートメントを作成する。
  - 必要に応じて JOBLIB/STEPLIB データ・セットから (通常 IMS.SDFSRESL ライブラリーから) 既存の装置特性テーブルをロードし、次に、DCT 項目ごとに DCENTRY ステートメントを作成する。
  - アセンブラーを呼び出し、入力としてそれらの DCTENTRY ステートメントと、DCTBLD マクロおよび MFSINIT マクロを渡す。
  - アセンブラーからの出力を、更新された装置特性テーブルまたは新しい装置特性テーブルとして、またデフォルトの MFS 形式定義の新しいセットとして使用可能にする。(この出力は後で行われる処理のために分割され、別個のファイルに入れられます。)
2. アセンブラーが呼び出され、新しい装置特性テーブルがアセンブルされる。
3. バインダーが呼び出され、新しい装置特性テーブルが IMS.SDFSRESL にバインドされる。
4. MFS 言語ユーティリティのフェーズ 1 で、新しいデフォルトの MFS 形式制御ブロックを生成する。
5. MFS 言語ユーティリティのフェーズ 2 で、新しいデフォルトの MFS 形式制御ブロックを IMS.FORMAT ライブラリーに入れる。

サブセクション:

- 『制約事項』
- 190 ページの『前提条件』
- 190 ページの『要件』
- 190 ページの『推奨事項』
- 190 ページの『JCL 仕様』
- 192 ページの『戻りコード』

### 制約事項

このユーティリティには、以下の制約事項が適用されます。

- PROCLIB から 1 つまたは 2 つの記述子メンバーを読み取る場合に、ユーティリティーは他の記述子をすべて無視します。
- 少なくとも 1 つの装置記述子を指定しないと、ユーティリティーは終了します。

## 前提条件

現在、DFSUTB00 ユーティリティーについて文書化されている前提条件はありません。

## 要件

DFSUTB00 ユーティリティーを実行するには、MFS 装置記述形式の要件を満たす必要があります。

MFS 装置特性テーブル・ユーティリティーは、システム定義を行わずに、MFS 装置記述子を使用して DCT の画面サイズを更新し、新しいデフォルトの MFS 形式を生成します。

## 推奨事項

現在、DFSUTB00 ユーティリティーについて文書化されている推奨事項はありません。

## JCL 仕様

MFS DCT プロシージャには、プロシージャ・ステートメント、EXEC ステートメント、DD ステートメント、および MFS 装置記述が必要です。

プロシージャ・ステートメント

プロシージャ・ステートメントは、以下の例に示す形式でなければなりません。

```
PROC RGN=4M,SOUT=A,SYS2=,PXREF=NOXREF,
PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
COMPR=NOCOMPRESS,COMPR2=COMPRESS,
LN=55,SN=8,DEVCHAR=0,COMPR3=NOCOMPRESS,
DIRUPDT=UPDATE,DCTSUF=,
DSCTSUF=,DSCMSUF=,FMTMAST=N
```

オプションのキーワード・パラメーターに加えて、指定する残りのパラメーターに応じて、以下のパラメーターを指定することが必要になる場合もあります。(x は、メンバー名に追加する英数字の接尾部です。)

### DCTSUF=x

DFSUDT0 に付加する接尾部を指定します。名前 DFSUDT0x は、新しい定義が追加された装置特性テーブルを識別します。この接尾部は、IMSGEN マクロの SUFFIX= キーワードで指定された値に対応しています。接尾部を指定しないと、装置記述子だけからまったく新しい装置特性テーブルが作成されます。

### DSCTSUF=x

DFSDSCT に付加する接尾部を指定します。名前 DFSDSCTx は、記述子メンバーを識別します。この接尾部は、IMS プロシージャの DSCT= キーワードで指定された値に対応しています。DSCMSUF= が指定されていない場合、このパラメーターが必要です。

### **DSCMSUF=x**

DFSDSCM に付加する接尾部を指定します。名前 DFSDSCMx は、記述子メンバーを識別します。この接尾部は、IMSGEN マクロの SUFFIX= キーワードで指定された値に対応しています。DSCTSUF= が指定されていない場合、このパラメーターが必要です。

### **FMTMAST=Y/N**

マスター端末で MFS に対する IMS 提供のサポートを使用するか (Y)、または使用しないか (N) を指定します。

### **EXEC** ステートメント

EXEC ステートメントは、装置特性テーブルを作成することを決定します。また、所要の記述子メンバーの名前と、更新された装置特性テーブルまたは新しい装置特性テーブルの名前も指定します。このプロシーチャーの 5 つのステップは、それぞれ異なるプログラムを指定して実行します。

以下の図は、MFSDCT (DFSUTB00) ユーティリティの 5 つのステップを示します。

```
//S1 EXEC PGM=DFSUTB00,REGION=&RGN,  
// PARM=('DCTSUF=&DCTSUF,DSCTSUF=&DSCTSUF'  
// 'DSCMSUF=&DSCMSUF,DEVCHAR=&DEVCHAR')  
//S2 EXEC PGM=ASMA90,REGION=&RGN,  
// PARM=('OBJECT,NODECK,NOLIST',  
// COND=(0,LT)'  
//S3 EXEC PGM=IEWL,  
// PARM=('SIZE=880K,64K),NCAL,LET,REUS,XREF,LIST',  
// REGION=&RGN,  
// COND=(0,LT)  
//S4 EXEC PGM=DFSUPAA0,REGION=&RGN,  
// PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,&COMPR,;  
// 'LINECNT=&LN,STOPRC=&SN,DEVCHAR=&DEVCHAR'),  
// COND=(0,LT)  
//S5 EXEC PGM=DFSUNUB0,REGION=&RGN,  
// PARM=(&COMPR2,&COMPR3,&DIRUPDT,;  
// 'DEVCHAR=&DEVCHAR'),COND=((0,LT,S1),  
// (0,LT,S2),(0,LT,S3),(8,LT,S4))
```

### **DD** ステートメント

MFSDCT プロシーチャーのステップ 1 では、以下の DD 名が使用されます。

#### **DCT**

更新済みまたは新規の装置特性テーブルのための一時データ・セットを、DD 名 SYSLIN でアセンブラーからの出力として定義し (ステップ 2)、DD 名 DCT で使用してバインダーへの入力として定義します (ステップ 3)。

#### **DCTIN**

装置特性テーブルのための一時データ・セットをアセンブラーへの入力として定義します (ステップ 2)。

#### **DCTLNK**

ステップ 3 用にバインド制御ステートメントのための一時データ・セットを定義します。

## DEFLTS

MFS 言語ユーティリティーの入力に対するデフォルトの MFS 形式定義のため  
の一時データ・セットを定義します (ステップ 4)。

## PROCLIB

記述子メンバー DFSDSCMx と DFSDSCTx が入っているライブラリーを定義  
します。

## STEPLIB

DFSUTB00 プログラムと、DCTSUF= パラメーターに指定された装置特性テー  
ブルが入っているライブラリーを定義します。

## SYSIN

生成された DCENTRY ステートメントを含む一時ファイルを定義します。

## SYSLIN

更新済みまたは新規の装置特性テーブルのための一時データ・セットを、アセン  
ブラーからの出力として定義し (ステップ 2)、DD 名 DCT で使用してバイン  
ダーへの入力として定義します (ステップ 3)。

## SYSLIB

IMS マクロと z/OS マクロが入っているライブラリーを定義します。

## SYSPRINT

ステップ 1 で印刷されたすべての出力 (エラー・メッセージとステップ 2 と 3  
からの出力も含む) 用のデータ・セットを定義します。

## SYSPUNCH

アセンブラーから出力されたオブジェクト・モジュールを含む一時ファイルを定  
義します。出力は装置特性テーブルと、その直後にデフォルトの MFS 形式定  
義が続く形になります。

## SYSUT1

アセンブラーおよびバインダー作業データ・セットを定義します。

## SYSLMOD

新しい装置特性テーブルまたは変更された装置特性テーブルを入れる  
IMS.SDFSRESL データ・セットを定義します。

## 戻りコード

戻りコードはエラー・メッセージに基づいています。

---

## DFSUTB00 ユーティリティーの実行

メッセージ形式サービス装置特性テーブル・プロシーチャーを実行することによ  
り、MFS 装置特性テーブル・ユーティリティーを呼び出すことができます。

### メッセージ形式サービス装置特性テーブル・プロシーチャー

```
| //          PROC RGN=4M,SOUT=A,SYS2=,PXREF=NOXREF,  
| //          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,  
| //          COMPR=NOCOMPRESS,COMPR2=COMPRESS,  
| //          LN=55,SN=8,DEVCHAR=0,COMPR3=NOCOMPRESS,  
| //          DIRUPDT=UPDATE,DCTSUF=,  
| //          DSCTSUF=,DSCMSUF=,FMMAST=N  
| //S1       EXEC PGM=DFSUTB00,REGION=&RGN,  
| //          PARM=('DCTSUF=&DCTSUF,DSCTSUF=&DSCTSUF',
```

```

//          'DSCMSUF=&DSCMSUF,DEVCHAR=&DEVCHAR'
//          'FMTMAST=&FMTMAST')
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SYSLIB  DD DSN=IMS.ADFSMAC,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//          DD DSN=SYS1.SDFSMAC,DISP=SHR
//PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSIN   DD DSN=&&SYSIN,UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSPUNCH DD DSN=&&SYSPUNCH,UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT1  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//DCTIN   DD DSN=&&DCTIN,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//DEFLTS  DD DSN=&&DEFLTS,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//DCTLNK  DD DSN=&&DCTLNK,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(1,1)),
//          DCB=BLKSIZE=800
//S2      EXEC PGM=ASMA90,REGION=& RGN,
//          PARM='OBJECT,NODECK,NOLIST',
//          COND=(0,LT)
//SYSLIB  DD DSN=IMS.ADFSMAC,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//SYSLIN  DD DSN=&&DCT,DISP=(NEW,PASS)
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DCB=BLKSIZE=800<
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(BLKSIZE=605),
//          SPACE=(605,(100,50),RLSE,,ROUND)
//SYSUT1  DD UNIT=SYSDA,DISP=(,DELETE),
//          SPACE=(CYL,(15,15))
//SYSIN   DD DSN=&&DCTIN,DISP=(OLD,DELETE)
//S3      EXEC PGM=IEWL,
//          PARM=('SIZE=(880K,64K)',NCAL,LET,REUS,
//          XREF,LIST),
//          REGION=&RGN,
//          COND=(0,LT)
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//          SPACE=(605,(10,10),RLSE,,ROUND)
//SYSLMOD DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SYSUT1  DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSPUNCH)),
//          SPACE=(CYL,(10,1))
//SYSLIN  DD DSN=&&DCTLNK,DISP=(SHR,DELETE)
//DCT     DD DSN=&&DCT,DISP=(SHR,DELETE)
//S4      EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,;
//          &COMPR,'LINECNT=&LN,STOPRC=&SN',
//          'DEVCHAR=&DEVCHAR'),COND=(0,LT)
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN   DD DSN=&&DEFLTS,DISP=(OLD,DELETE)
//REFIN   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFOUT  DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFRD   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//SYSTEXT DD DSN=&&TXTPASS,UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//DUMMY   DD DISP=SHR,
//          DSN=IMS.&SYS2.PROCLIB(REFCPY)

```

```

|          //UTPRINT DD SYSOUT=&SOUT
|          //SYSPRINT DD SYSOUT=&SOUT,
|          //          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
|          //SYSUDUMP DD SYSOUT=&SOUT
|          //SEQBLKS DD DSN=&&BLKS,DISP=(NEW,PASS),
|          //          UNIT=SYSDA,SPACE=(CYL,(1,1))
|          //S5      EXEC PGM=DFSUNUB0,REGION=&RGN,
|          //          PARM=(&COMPR2,&COMPR3,&DIRUPDT,;
|          //          'DEVCHAR=&DEVCHAR'),COND=((0,LT,S1),
|          //          (0,LT,S2),(0,LT,S3),(8,LT,S4))
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //SEQBLKS DD DSN=&&BLKS,DISP=(OLD,DELETE)
|          //UTPRINT DD SYSOUT=&SOUT,
|          //          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
|          //SYSUDUMP DD SYSOUT=&SOUT
|          //FORMAT  DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
|          //DUMMY    DD DISP=SHR,
|          //          DSN=IMS.&SYS2.PROCLIB(FMTCPY)
|          //SYSPRINT DD SYSOUT=&SOUT
|          //SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
|          //SYSUT4  DD UNIT=SYSDA,SPACE=(CYL,(1,1))

```

---

## 第 4 章 MFS 言語ユーティリティー (DFSUPAA0)

MFS 言語ユーティリティー (DFSUPAA0) は、メッセージ形式サービス (MFS) 制御ブロックを作成し、保管するために使用します。

制御ブロックの中間テキスト・ブロック (ITB) フォームは IMS.REFERAL ライブラリーに入っています。制御ブロックは、IMS.FORMAT ライブラリーに入れられ、IMS の通常の操作の過程で使用されます。

定義: 形式セットは、1 つの形式と、SOR= オペランドでこの形式を参照するすべてのメッセージで構成されています。

MFS 言語ユーティリティーには、標準、バッチ、およびテストの 3 つの操作モードがあります。どのようなモードでも、この言語ユーティリティーはオフラインで実行され、同一の制御ステートメントを受け入れ、同じ種類の ITB と制御ブロックを生成します。MFS ライブラリーの使用方法はモードによって異なります。したがって、それぞれのモードで使用されるプロシージャが異なります。

標準モードでは、IMS.REFERAL に書き込まれた ITB が、MFSUTL プロシージャによって制御ブロックに変換され、ステージング・ライブラリー IMS.FORMAT に入れられます。制御ブロックがアクティブ・ライブラリーではなく、ステージング・ライブラリーに入れられるので、標準モードはオンライン IMS 制御領域と並行して実行できます。

バッチ・モードが標準モードと異なるのは、MFSBTCH1 プロシージャでは、作成された制御ブロックを特殊なライブラリー IMS.MFSBATCH に入れておき、後で MFSBTCH2 プロシージャによって (別のジョブで)、ステージング・ライブラリー IMS.FORMAT に転送するという点にあります。

テスト・モードでは、MFSTEST プロシージャが制御ブロックを作成し、別の IMS.TFORMAT ライブラリーにこれらの制御ブロックを入れます。制御ブロックは、オンライン操作を妨げることなくテストできるので、オンライン IMS 制御領域と並行して操作できます。

IMS システム定義のステージ 2 では、以下のプロシージャが生成されます。

### **MFSUTL**

MFS オンライン制御ブロックを作成して、それらのブロックを IMS.FORMAT ライブラリーに入れる MFS 言語ユーティリティーの 2 ステップの標準モード実行プロシージャ。

### **MFSBTCH1**

MFS オンライン・ブロックを作成し、累積する MFS 言語ユーティリティーの 1 ステップのバッチ・モード実行プロシージャ。

### **MFSBTCH2**

累積された MFS オンライン制御ブロックを (MFSBTCH1 から) IMS.FORMAT ライブラリーに入れる MFS 言語ユーティリティーの 1 ステップのバッチ・モード実行プロシージャ。

## **MFSBACK**

MFS ライブラリーのバックアップをとる 2 ステップの実行プロシージャ。オプションの MFSTEST 機能が使用される場合、MFSBACK には、さらにテスト・ライブラリーのバックアップをとるステップが含まれます。

## **MFSREST**

MFS ライブラリーを復元する 2 ステップの実行プロシージャ。オプションの MFSTEST 機能が使用される場合、MFSREST には、さらにテスト・ライブラリーを復元するステップが含まれます。

## **MFSRVC**

MFS ライブラリーを維持する 1 ステップの実行プロシージャ。

システム定義の過程で MFSTEST モードが選択されている場合は、次に示す追加のプロシージャが生成されます。

## **MFSTEST**

MFS オンライン・ブロックを作成して、それらを IMS.TFORMAT ライブラリーに入れる MFS 言語ユーティリティの 2 ステップのテスト・モード実行プロシージャ。

MFS 言語ユーティリティには、新しい制御ブロックまたは代替の制御ブロックを作成するプロシージャに加えて、MFS ライブラリーのバックアップ操作および復元操作を行う MFSBACK および MFSREST プロシージャも含まれます。

削除およびリスト表示の操作はサービス・ユーティリティによって行われます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 197 ページの『要件』
- 197 ページの『推奨事項』
- 197 ページの『JCL 仕様』

## **制約事項**

MFSTEST プロシージャを、それ自体、または MFS ライブラリーを使用する他のプロシージャやプログラムと並行して実行しないでください。IMS.TFORMAT の制御ブロックをテストするには、端末オペレーターは /TEST MFS コマンドを入力します。次に、IMS.TFORMAT のテスト制御ブロックが (必要であれば、アクティブ形式ライブラリーのオンライン制御ブロックも) テスト操作のためにバッファーに読み込まれます。テストが正常に終了した後、制御ブロックをステージング・ライブラリー IMS.FORMAT に入れるには、MFSUTL プロシージャを使用してソース・ステートメントを再コンパイルします。

## **前提条件**

現在、DFSUPAA0 ユーティリティについて文書化されている前提条件はありません。



## 要件

現在、DFSUPAA0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSUPAA0 ユーティリティについて文書化されている推奨事項はありません。

## JCL 仕様

DFSUPAA0 ユーティリティには、MFSUTL、MFSBTCH1、および MFSTEST プロシーチャーの EXEC ステートメントと DD ステートメントが必要です。

ステップ 1 (S1) を実行する場合 (MFSUTL、MFSBTCH1、MFSTEST プロシーチャーで) に、EXEC ステートメントの PARM キーワードには、次のパラメーターの指定が可能です。

### **PXREF= NOXREF | XREF**

ソートされた相互参照リストを生成するか (XREF)、生成しないか (NOXREF) を指定します。デフォルト値は NOXREF です。ソートされた相互参照リストには、すべてのラベルと関連する参照のリストが含まれています。

### **PCOMP= NOCOMP | COMP**

エラー・リカバリー処理または置換処理によってステートメントが変更された後で、複合バージョンすなわち最終バージョンのステートメントを印刷するか (COMP または COMPOSITE)、印刷しないか (NOCOMP) を指定します。デフォルト値は NOCOMP です。複合ステートメントは、エラー・リカバリーの過程で行われた構文上の仮定を反映しています。セマンティックの仮定は複合ステートメントには組み込まれず、中間テキスト・ブロックに反映されます。MFLD/DFLD ステートメントの反復生成機能を使用した場合、COMP を指定すると、生成されたステートメントも印刷されます。NOCOMP を指定すると、この印刷は抑止されます。

### **PSUBS= NOSUBS | SUBS**

ステートメントのオペランド・フィールドで置換変数が検出された場合に、その置換変数とその等価の値を印刷するか (SUBS または SUBSTITUTE)、印刷しないか (NOSUBS) を指定します。デフォルト値は NOSUBS です。

### **PDIAG= NODIAG | DIAG**

XREF、COMP、および SUBS オプションをすべてオンに設定するか (DIAG または DIAGNOSTIC)、オンに設定しないか (NODIAG) を指定します。DIAG または DIAGNOSTIC を指定すると、さらに診断情報が印刷されます。デフォルト値は NODIAG で、XREF、COMP、および SUBS オプションには影響しませんが、診断情報の印刷を抑止します。

### **COMPR= NOCOMPRESS | COMPRESS**

新しい ITB が追加される前に IMS.REFERAL ライブラリーを圧縮するか (COMPRESS)、圧縮しないか (NOCOMPRESS) を指定します。デフォルト値は NOCOMPRESS です。

### **LN= 55 | nn**

ページ当りの印刷行数を指定します。デフォルト値は 55 です。

**SN= 08 | nn**

重大度コードの比較値を指定します。MSG、FMT、および TABLE ブロックでエラーの重大度がこの比較値に等しいか、または超えるブロックは、IMS.REFERAL ライブラリーには書き込まれません。デフォルト値は 08 です。

**DEVCHAR= 0 | x**

DFSUDT0 に付加する英数字の接尾部 (x) を指定します。名前 DFSUDT0 は、所定の装置特性テーブルを識別します。この接尾部 (x) は、IMSGEN マクロの SUFFIX= キーワードで指定された値に対応しています。デフォルトはゼロ (0) です。

MFSRVC プロシージャを実行する場合、パラメーターを 1 つ指定できます。LIST 制御ステートメントのパラメーター DEVCHAR に接尾部の指定がない場合に、パラメーター DEVCHAR=0 または x を指定すると、装置特性テーブルに使用される英数字の接尾部 (x) が指定されます。デフォルトはゼロです。

MFSUTL または MFSBTCH2 プロシージャでステップ 2 (S2) を実行する場合には、EXEC ステートメントの PARM キーワードに次の 3 つのパラメーターを指定できます。

**COMPR2= COMPRESS | NOCOMPRESS**

新しい制御ブロックが追加される前に IMS.FORMAT ライブラリーを圧縮するか (COMPRESS)、圧縮しないか (NOCOMPRESS) を指定します。デフォルト値は COMPRESS です。

**COMPR3= COMPREND | NOCOMPREND**

すべての形式ブロックが追加または置換され、索引ディレクトリー (\$\$IMSDIR) が更新された後で、DD 名 FORMAT のデータ・セットを圧縮するか (COMPREND)、圧縮しないか (NOCOMPREND) を指定します

**DIRUPDT= UPDATE | NOUPDATE**

形式ライブラリーからブロックが削除された後で、特殊索引ディレクトリー (\$\$IMSDIR) を自動的に更新するか (UPDATE)、更新しないか (NOUPDATE) を指定します。NOUPDATE を指定すると、\$\$IMSDIR の更新は行われません。デフォルトは UPDATE です。

MFSTEST プロシージャのステップ 2 (S2) を実行する場合には、PARM='TEST' パラメーターを指定しなければなりません。

指定できる EXEC ステートメントの他のパラメーターを次に示します。

**RGN=**

このユーティリティーを実行する領域のサイズを指定します。デフォルトは 360K です。

**SOUT=**

SYSOUT クラスを指定します。デフォルトは A です。

**SNODE=**

MFS ユーティリティー・データ・セット名に割り当てることができるノードを指定します。デフォルト値は IMS です。

**SOR=**

SYSIN または SYSLIB の MFS ユーティリティ・ライブラリーに割り当てることができるライブラリー名を指定します。デフォルトは NOLIB です。

**MBR=**

SYSIN の MFS ユーティリティ・メンバーに割り当てることができるメンバー名を指定します。デフォルトは NOMBR です。

**EXEC** ステートメント

MFS 言語ユーティリティでサポートされている EXEC ステートメント・パラメーターには、コンパイル上の各種制御機能があります。

EXEC ステートメントの形式は次のとおりです。

```
//S1      EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,
//          &COMPR,'LINECNT=&LN,STOPRC=&SN',
//          'DEVCHAR=&DEVCHAR')
```

プリプロセッサとフェーズ 1 に対して EXEC ステートメントに指定されるパラメーターには、次の働きがあります。

- 印刷出力を制御します。
- 参照ライブラリー (IMS.REFERAL) を圧縮します。
- 診断情報を要求します。
- どの MFS 装置特性テーブルを使用するか指定します。
- 指定のレベルのエラーを含んでいる制御ブロックが IMS.REFERAL に書き込まれないようにします。

フェーズ 2 に対しても EXEC ステートメントにパラメーターを指定して、IMS.FORMAT と IMS.REFERAL を圧縮するかどうか、および削除後に \$IMSDIR を自動更新するかどうかを指定することができます。

DEVCHAR パラメーターは、使用する MFS 装置特性テーブルの接尾部を指定します。装置特性テーブルにアクセスするには、MFS 言語ユーティリティへの入力での DEV TYPE=3270-An (n は 1 から 15) が指定されていなければなりません。

MFS 言語ユーティリティでサポートされている EXEC ステートメント・パラメーターには、コンパイル上の各種制御機能があります。指定できるパラメーターは次のとおりです。

**NOXREF | XREF**

ソートされた相互参照リストを生成するか (XREF)、生成しないか (NOXREF) を指定します。ソートされた相互参照リストには、すべてのラベルと関連する参照のリストが含まれています。デフォルトは NOXREF です。

**NOCOMP | COMP**

エラー・リカバリー処理または置換処理によってステートメントが変更された後で、複合バージョンすなわち最終バージョンのステートメントを印刷するか (COMP または COMPOSITE)、印刷しないか (NOCOMP) を指定します。複合ステートメントは、エラー・リカバリーの過程で行われた構文上の仮定を反映しています。セマンティックの仮定は複合ステートメントには組み込まれず、中間テキスト・ブロックに反映されます。デフォルトは NOCOMP です。

## **NOSUBS | SUBS**

置換変数 (EQU オペランド) を含んでいるステートメントを印刷するか (SUBS または SUBSTITUTE)、印刷しないか (NOSUBS) を指定します。デフォルトは NOSUBS です。

## **NODIAG | DIAG**

XREF、COMP、および SUBS オプションをオンに設定し、診断情報を印刷するか (DIAG または DIAGNOSTIC)、しないか (NODIAG) を指定します。デフォルトは NODIAG で、XREF、COMP、および SUBS オプションの設定には影響しませんが、診断情報の印刷を抑止します。

## **NOCOMPRESS | COMPRESS**

新しい ITB が追加される前に IMS.REFERAL ライブラリーを圧縮するか (COMPRESS)、圧縮しないか (NOCOMPRESS) を指定します。デフォルトは NOCOMPRESS です。

## **DIRUPDT= UPDATE | NOUPDATE**

形式ライブラリーから 1 つ以上のブロックが削除された後、特殊索引ディレクトリー (\$\$IMSDIR) を自動的に更新するか (UPDATE)、更新しないか (NOUPDATE) を指定します。NOUPDATE を指定すると、\$\$IMSDIR の更新は行われません。デフォルトは UPDATE です。

## **LINECNT=*nn***

ページ当りの印刷行数を指定します。デフォルトは 55 です。

## **STOPRC=*nn***

重大度コードの比較値を指定します。MSG、FMT、および TABLE ブロックでエラーの重大度がこの比較値に等しいか、または超えるブロックは、IMS.REFERAL ライブラリーには書き込まれません。デフォルトは 08 です。

## **DEVCHAR=*n*|*x***

DEV TYPE=3270-An でロードされる装置特性テーブルの名前 (DFSUDT0x) の最後の 1 文字として使用する英数字の接尾部 (x) を指定します。デフォルトはゼロ (DFSUDT00) です。

定義ステートメントは、示された順序で説明します。DO と ENDDO コンパイル・ステートメントについては、通常コーディングされる位置、つまり、MFLD または DFLD ステートメントの前後で説明します。コンパイル・ステートメントの形式は、関連機能 (あれば) ごとの順序で説明します。つまり、ALPHA、COPY、EQU、および RESCAN (等価設定処理)、STACK および UNSTACK (SYSIN/SYSLIB レコードのスタッキング)、TITLE、PRINT、SPACE、および EJECT (SYSPRINT リスト制御)、END の順になります。

## **MFSUTL および MFSTEST 領域パラメーターの見積もり**

MFSUTL および MFSTEST プロシーチャーを呼び出す EXEC ステートメントの RGN= パラメーターに指定すべき主記憶域の必要量を見積もる際に、以下のステップが役に立ちます。

1. ステートメント基底カウントを計算します。MFS 言語ユーティリティーへの入力の場合、処理される装置形式のうち (ステートメントの数が) 最大の形式と、この形式に関連する最大メッセージ記述子を判別します。これらの 2 つの制御ブロックに含まれるステートメントの合計数を加えて、ステートメント基底カウントを算出します。

特定のユーザー指定の MSG ITB または FMT ITB を処理する際に、このユーティリティは IMS.REFERAL データ・セットから保管されたすべての関連する MSG ITB または FMT ITB を再処理し、関連するすべてのオンライン・ブロック間でリンク互換性を確保します。これらの再処理された ITB も、ステートメント基底カウントを算出する処理のために、分析する必要があります。

- 領域の必要量を見積もります。ステートメント基底カウントに 214 を掛けた値に 300000 を加えます。これにより算出された値を、2048 の倍数で最も近い値に切り上げます。切り上げた値が、主ストレージの必要量の見積もりとなります。この値を、MFSUTL および MFSTEST プロシーチャーを呼び出す EXEC ステートメントの RGN= パラメーターに指定してください。

ステートメント基底カウントに関連して多くのリテラル DFLD ステートメントを持つ複雑な形式では、この見積もりを超えることもあります。

## **DD ステートメント**

MFSUTL、MFSBTCH1、MFSBTCH2、および MFSTEST プロシーチャーで使用されるデータ・セット名は、インストール要件に適合します。DD 名と DD 名が参照するデータ・セットは次のとおりです。

### **REFIN**

### **REFOUT**

### **REFRD**

MFSTEST プロシーチャーで使用される場合を除き、MFS 参照ライブラリーを参照します。MFSTEST では、REFIN と REFRD は MFS 参照ライブラリーを参照し、REFOUT は一時データ・セットです。

### **FORMAT**

MFS 制御ブロック・ライブラリーを参照します。MFSTEST では、この DD 名は MFS テスト制御ブロック・ライブラリーを参照します。

### **SYSLIB**

入力のコピー元のオプションのユーザー・ライブラリーを参照します。

### **SYSIN**

順次データ・セット、または区分データ・セットのメンバーのいずれかの入力データ・セットを参照します。

### **DUMMY**

MFS 参照ライブラリーおよび制御ブロック・ライブラリーの圧縮に使用される制御ステートメントが入っている IMS プロシーチャー・ライブラリーを参照します。

### **SYSUT3**

### **SYSUT4**

データ・セットの圧縮の過程で作業データ・セットとして使用されるデータ・セットの DD 名です。

MFS 参照ライブラリーも MFS 制御ブロック・ライブラリーも圧縮されない場合は、DUMMY、SYSUT3、および SYSUT4 はすべて省略することができます。

#### UTPRINT

MFR 参照ライブラリーの圧縮の過程でメッセージ用に使用され、MFS 言語ユーティリティーのフェーズ 2 の処理の過程では MFS エラーおよび状況メッセージ用に使用されます。

MFSRVC プロシージャで使用されるデータ・セットを参照する DD 名は次のとおりです。データ・セット名はインストール要件に適合するように変更できます。

#### REFIN

MFS 参照ライブラリーを参照します。

#### FORMAT

MFS 制御ブロック・ライブラリーを参照します。

#### SYSIN

順次データ・セット、または区分データ・セットのメンバーのいずれかの入力データ・セットを参照します。


#### SYSSNAP

特定の重大エラーが検出された場合に、SNAP マクロからの出力の受け取りに使用されるデータ・セットを指します。

#### SYSPRINT

出力の宛先を参照します。出力先がデータ・セット (SYSOUT= の代わりに) の場合、このデータ・セットに DISP=MOD を使用します。

関連概念:

 MFS 装置特性テーブル (アプリケーション・プログラミング API)

---

## ユーティリティー制御ステートメントおよび構文規則

MFS 言語ユーティリティーで使用する制御ステートメントは、次の 2 つの主なカテゴリーに分けられます。定義ステートメントおよびコンパイル・ステートメント

MFS 言語ユーティリティーで使用する制御ステートメントは、次の 2 つの主なカテゴリーに分けられます。

- 定義ステートメント。メッセージ形式、装置形式、区画セット、オペレーター制御テーブルの定義に使用されます。
- コンパイル・ステートメント。定義ステートメントのコンパイルと SYSPRINT リストの制御に使用されます。

定義ステートメントとコンパイル制御ステートメントを用いて、ユーティリティーに実行させる機能と、各種オプションを指定します。

定義およびコンパイル制御機能は次のとおりです。

- SYSPRINT リスト制御

コンパイル・リストのフォーマット設定には、XREF、SUBS、COMP、DIAG、LINECNT の各パラメーターが用意されています。

- SYSIN/SYSLIB レコードのスタッキングとアンスタッキング

1 つ以上の SYSIN/SYSLIB レコードを処理し、後にコンパイルで再使用できるようプロセッサ・ストレージにとどめておくための制御ステートメントが用意されています。反復されるステートメント・グループでは、これらのステートメントを COPY の代替機能として使用できます。

MFLD ステートメントと DFLD ステートメントは、前に DO ステートメント、後ろに ENDDO ステートメントを置くことで、繰り返し生成できます。DFLD の反復生成では、行番号と桁番号に増分を加えていくことができます。

- 英字の生成

ALPHA ステートメントでは、文字群に新しい文字を追加し、それを英字と指定できます。

- COPY

COPY ステートメントでは、区分データ・セットのメンバーをユーティリティー・プリプロセッサの入力ストリームの中へコピーできます。

制御ステートメントは、アセンブラーに似た言語で書かれます。標準形式は次のとおりです。

## MFS 言語ユーティリティーの制御ステートメント構文

<i>label</i>	<i>operation</i>	<i>operand</i>	<i>comments</i>
--------------	------------------	----------------	-----------------

### **label**

ステートメントを指定します。オプションとされているときは、省略できます。名前を指定するときは、それをステートメントの先頭 (1 桁目) から始め、名前の後ろに 1 つ以上の空白を置かなければなりません。名前には 1 から 8 文字 (FMT ラベルの場合は 1 から 6 文字) の英数字を使用できます。第 1 文字は英字でなければなりません。

### **operation**

制御ステートメントの種類を指定します。普通は 10 桁目から始まります。前後に 1 つ以上の空白が必要です。

### **operand**

1 つ以上のパラメーターから成ります。パラメーターには、定位置パラメーターとキーワード・パラメーターがあります。MFS 制御ステートメントの定位置パラメーターは、常にオペランドの先頭に置かれます。通常、16 桁目から始まります。キーワード・パラメーターでは、位置は重要ではありません。1 つのオペランド内では、パラメーターはコンマで区切られます。したがって、制御ステートメントの構文説明の中で、頭にコンマを持つパラメーターはキーワード・パラメーターです。オペランド・フィールド自体の前後にも、1 つ以上の空白が必要です。

### **comments**

ユーティリティー制御ステートメントに書けますが、オペランド・フィールドの最後のパラメーターと 1 つ以上の空白で区切らなければなりません (ステートメントにオペランドがないときは、ステートメントとコメントを 1 つ以上の空白で区切ります)。コメント行を開始するには、1 桁目にアスタリスクを置きます。

行の継続には、72 桁目にブランク以外の文字を入れます。現在行がコメント行なら、継続行は任意の桁から開始して構いません。

ほかに、次の点に留意してください。

- 継続行の数に制限はありません。
- オペランド・フィールドに含まれる文字の数に制限はありません。個々のオペランド項目は、256 文字を超えることができませんが、中に埋め込まれたり末尾に置かれたりしている 2 つ目の引用符は、その数に含まれません。
- リテラル内部に標準外の文字が見付かると、重大度 4 の警告メッセージが出されます。標準外の文字はリテラル内にそのまま維持されます。
- 現在行が制御ステートメントの場合、継続行は 16 桁目から始めなければなりません。
- リテラル内に 1 個のアンパーサンド文字を作り出すには、1 個のアンパーサンドが必要です。

定義ステートメントとコンパイラー・ステートメントでの指定に加え、EXEC ステートメントの PARM キーワードに指定できるパラメーターもいくつかあって、これでプリプロセッサとフェーズ 1 の現行コンパイルを制御できます。フェーズ 2 でも、1 個のパラメーターを指定できます。

以下に示す 5 つの特別規則では、例に実際の MFS コードを使用しています。

1. ステートメントのコーディングで、等号または左括弧の直後にコンマが来るときは、そのコンマを省略できます。

`,FTAB=(,FORCE)` は、`,FTAB=(FORCE)` のようにコーディングできます。

2. ステートメントのコーディングで、等号の直後に括弧で囲まれた単一項目が来るときは、括弧を省略できます。

`,FTAB=(,FORCE)` は、`,FTAB=,FORCE` のようにコーディングできます。

3. 上記規則 1 と規則 2 は、同じ単一項目に任意の順序で適用できます。

`,FTAB=(,FORCE)` は、`,FTAB=FORCE` のようにコーディングできます。

4. キーワードを指定するときは、どのような場合であれ、そのキーワードの直後に少なくとも 1 個のパラメーターを指定しなければなりません。

`,FTAB=` と `,FTAB=,LDEL='**'` は、どちらも認められません。

5. ラベルとステートメント・タイプ名の間、およびステートメント・タイプ名とそのパラメーターの間には、ブランクが必要です。それ以外の場所では、明示的に `b` という記号で表現しない限り、ブランクは使用できません。

`DEV ,PAGE` は正しく、`DEV,PAGE` と `,FTAB= ,(MIX)` は誤りです。

## 構文エラー

MFS 言語ユーティリティーは、ソース・ステートメント中に構文エラーを見付けると、そのエラーからのリカバリーを試みます。そのリカバリーでの前提事項が正しいかどうかは保証の限りではありません。また、IMS のリリースごとに前提事項は異なります。リカバリーのための前提事項は、(1) 誤った項目が見付かったとき何が



必要とされていたか、(2) 誤った項目の直前の項目の右側には何を指定できるか、(3) 誤った項目の左側には何を指定できるか、に基づいて決められます。

エラー・リカバリーの処理では、診断メッセージに次の記号が用いられます。

**;** ソース・ステートメントの終わりに達したことを表します。位置マーカーは、スキャンされた最後のソース項目の直後の位置を指し示します。

**\$L\$** リテラル・オペランド項目を表します。

**\$V\$** ID オペランド項目 (英字で、後ろに英数字が続く場合もある) を表します。

**\$I\$** 数値オペランド項目を表します。

**\$A\$** 英数字オペランド項目 (数字で、後ろに英数字が続く場合もある) を表します。

**\$D\$** 区切りオペランド項目を表します。

ほとんどのエラー・リカバリー・メッセージは、重大度コードが 4 です。これは、警告レベルのエラーを表します。項目が削除されたり、構文スキャンが打ち切られたりすると、ステートメントの有効な処理ができず、重大度コードが 8 になります。

## 無効なステートメント・シーケンス

MSG、FMT、PDB、または TABLE の各定義ステートメントを処理する言語ユーティリティ・プリプロセッサ・ルーチン群は、階層的に編成されています。あるレベルのルーチンはそのレベルのステートメントを処理し、次のステートメントの読み取りで、どのルーチンに制御を渡すべきか決定します。

新しく読み取ったステートメントが 1 つ下のレベルのステートメントであれば (例えば、DEV ステートメントの次に DIV ステートメントがあれば)、1 つ下のレベルのルーチン (例えば、DIV ステートメント・プロセッサ) が呼び出されます。

新しく読み取ったステートメントが 1 つ下のレベルのステートメントでないときは、次の 3 つのルーチンのうちいずれかに制御が渡されます。

- 欠落しているステートメントを扱う 1 つ下のレベルのルーチン (例えば、DEV ステートメントの次に DPAGE ステートメントがあったときは、DIV プロセッサ)
- 新しく読み取ったステートメントが現在のプロセッサと同じレベルであれば (例えば、一連の DFLD ステートメントの場合)、同じレベルのルーチン
- 新しく読み取ったステートメントが、同じレベルでも 1 つ下のレベルでもないときは (例えば、DFLD ステートメントに続く DEV ステートメント、無効なステートメント、順序の狂ったステートメントなど)、1 つ上のレベルの (つまり、現在のルーチンを呼んだ) ルーチン

こうして、MSG、FMT、PDB、または TABLE 定義の階層構造が無効であったり、ステートメント演算子にミススペルがあったりすると、上記 (3) が適用され、1 つずつ上のレベルのルーチンに制御が返されます。最高レベルでは、FMT、MSG、TABLE、PDB、END のいずれかのステートメントしか、プリプロセッサに受け入れられません。したがって、次の FMT、PDB、MSG、TABLE、または END に

至るまでにあるすべてのステートメントはフラッシュされ (つまり処理されない)、適切なエラー・メッセージで通知されます。

## 制御ステートメントの要約

メッセージ形式、装置形式、区画セット、オペレーター制御テーブルの定義には、それぞれに異なる階層的な定義ステートメント・セットが用いられます。

### メッセージ定義ステートメント・セット

メッセージ形式の定義に用いられます。次のステートメントがあります。

**MSG** メッセージ定義の開始を表します。

### LPAGE

関連するセグメント/フィールド定義グループを指定します。

### PASSWORD

IMS パスワードとして使う 1 つ以上のフィールドを指定します。

**SEG** メッセージ・セグメントを指定します。

**DO** 後続 MFLD ステートメントの反復処理を要求します。

### MFLD

メッセージ・フィールドを定義します。MFLD ステートメントの反復処理を呼び出すには、DO ステートメントと ENDDO ステートメントを使用できます。反復処理の実行には、MFLD ステートメントの前に DO ステートメント、後ろに ENDDO ステートメントを置いてください。

### ENDDO

先行 MFLD ステートメントの反復処理を終了します。

### MSGEND

メッセージ定義の終わりを表します。

### フォーマット定義ステートメント・セット

装置形式の定義に用いられます。次のステートメントから成っています。

**FMT** フォーマット定義の開始を表します。

**DEV** 装置タイプと動作オプションを指定します。

**DIV** フォーマットが入力用か、出力用か、両用かを指定します。

### DPAGE

メッセージ・フィールドの LPAGE グループに対応する装置フィールド・グループを指定します。

### PPAGE

リモート・アプリケーション・プログラムに一度に送信できる、論理的に関連したレコードのグループを指定します。

**DO** 後続 RCD ステートメントまたは DFLD ステートメントの反復処理を要求します。

**RCD** リモート・アプリケーション・プログラムに 1 つのレコードとして送信される、関連した装置フィールドのグループを指定します。

## **DFLD**

装置フィールドを定義します。DFLD ステートメントの反復処理を呼び出すには、DO ステートメントと ENDDO ステートメントを使用できます。反復処理の実行には、DFLD ステートメントの前に DO ステートメント、後ろに ENDDO ステートメントを置いてください。

## **ENDDO**

後続 RCD ステートメントまたは DFLD ステートメントの反復処理を終了します。

## **FMTEND**

フォーマット定義の終わりを表します。

## **区画定義ステートメント・セット**

区画セット (区画記述子ブロック) の定義に用いられます。次のステートメントから成っています。

**PDB** 区画セット定義の始まりを表します。その区画セットを記述するいくつかのパラメーターを指定できます。

**PD** 区画記述子を定義します。区画記述子には、区画の記述に必要なパラメーターが含まれます。

## **PDBEND**

区画セット定義の終わりを表します。

## **テーブル定義ステートメント・セット**

オペレーター制御テーブルの定義に用いられます。次のステートメントがありません。

## **TABLE**

テーブル定義の開始を表します。

**IF** 条件付きテストと、テスト後の処置を定義します。

## **TABLEEND**

テーブル定義の終わりを表します。

## **コンパイル・ステートメント**

各種機能に用いられます。MFS 言語ユーティリティーでサポートされているコンパイル・ステートメントをアルファベット順にリストします。

## **ALPHA**

フィールド名とリテラルを定義する上で英字と見なす 1 組の文字を定義します。

## **COPY**

SYSLIB DD ステートメントで表される区分データ・セットのメンバーを、プリプロセッサの入力ストリームの中へコピーします。

**DO** MFLD または DFLD 定義ステートメントの反復処理を要求します。

## **EJECT**

SYSPRINT リストを次ページへページ替えします。

**END** SYSIN 処理するデータの終わりを指定します。

## ENDDO

MFLD、RCD、または DFLD 定義ステートメントの反復処理を終了します。

**EQU** 記号を番号、英数字 ID、またはリテラルと等価に設定します。

## PRINT

SYSPRINT オプションを制御します。

## RESCAN

EQU 処理を制御します。

## SPACE

SYSPRINT リストで行をスキップします。

## STACK

再使用のためプロセッサ・ストレージに保持しておく 1 つ以上の SYSIN または SYSLIB レコードを指定します。

## TITLE

SYSPRINT リストの表題を指定します。

## UNSTACK

これまでにスタックされている SYSIN または SYSLIB レコードを取り出します。

コンパイル・ステートメントは、一連の制御ステートメントにおける論理的に適切な地点に挿入されます。例えば、TITLE は先頭に置き、EJECT は MSG、FMT、TABLE といった各ステートメントの前に置けます。

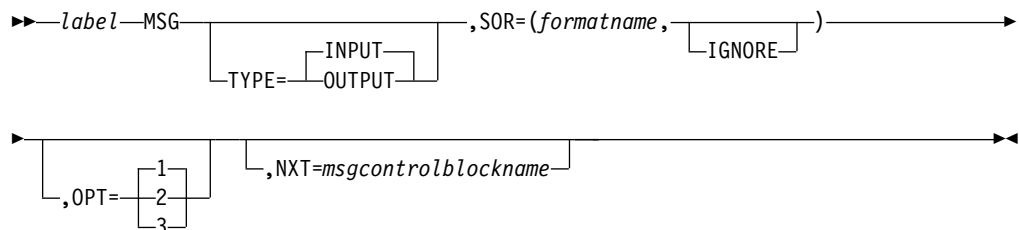
## メッセージ定義ステートメント

メッセージ定義ステートメントには、MSG ステートメント、LPAGE ステートメント、PASSWORD ステートメント、SEG ステートメント、DO ステートメント、MFLD ステートメント、ENDDO ステートメント、および MSGEND ステートメントがあります。

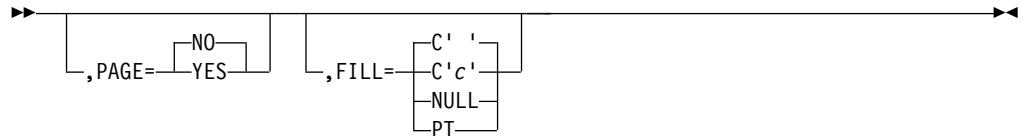
### MSG ステートメント

MSG ステートメントは、メッセージの入力定義または出力定義を開始し、それに名前を付けます。

### MSG TYPE=INPUT または OUTPUT のときの形式



### MSG TYPE=OUTPUT のみのときの形式



## パラメーター

### label

1 から 8 文字の英数字名を指定してください。このラベルは、他のメッセージ記述子の NXT オペランドで参照できます。

### TYPE=

この定義をメッセージ INPUT 制御ブロックまたは OUTPUT 制御ブロックに指定します。デフォルトは INPUT です。

### SOR=

FMT ステートメントのソース名を指定します。その FMT ステートメントが DEV ステートメントと共に、このメッセージ記述子で処理される端末またはリモート・プログラム・データ・フィールドを定義します。TYPE=OUTPUT のときに IGNORE を指定すると、MFS は、装置形式定義で FEAT= オペランドに IGNORE が指定されている装置に対して指定されたデータ・フィールドを使用します。TYPE=INPUT で IGNORE を指定するのは、対応するメッセージ出力記述子に IGNORE が指定されている場合に限ってください。SOR=IGNORE を使用するとき、メッセージ入力記述子とメッセージ出力記述子の両方に IGNORE を指定しなければなりません。

### OPT=

MFS がメッセージの編集に使用するメッセージ・フォーマット設定オプションを指定します。デフォルトは 1 です。

### NXT=

このメッセージ記述子によってメッセージを処理した結果、次のメッセージとして予想されるメッセージのマッピングに使用されるメッセージ記述子の名前を指定します。TYPE=INPUT なら、NXT= にはメッセージ出力記述子を指定します。TYPE=OUTPUT なら、NXT= にはメッセージ入力記述子を指定します。ISC 出力では、NXT= が ATTACH FM ヘッダーの RDPN になります。

TYPE=OUTPUT で、SOR= オペランドに指定された *formatname* に 3270 または 3270P 装置タイプの形式が含まれているときは、NXT= で参照される *msgcontrolblockname* も同じ *formatname* を使用しなければなりません。

### PAGE=

この制御ブロックを使って編集するメッセージに、オペレーター論理ページング(前方・後方ページング)を行うか (YES)、行わないか (NO) を指定します。このオペランドは、TYPE=OUTPUT の場合のみ有効です。デフォルトは NO です。つまり、物理ページの前方ページングだけを行います。

### FILL=

出力装置フィールドに使う充てん文字を指定します。このオペランドは、TYPE=OUTPUT の場合のみ有効です。デフォルトは C' ' です。この充てん文字の指定は、FMT 定義の DPAGE ステートメントに FILL=NONE が指定されていないと、無視されます。3270 出力に EGCS フィールドがあるときは、FILL=PT または FILL=NULL だけを指定してください。FILL=PT では、出力

フィールド (1 バイトまたは 2 バイト) にデータが送られる場合のみ、そのフィールドが消去されます。したがって、アプリケーション・プログラム・メッセージで MFLD が省略されていると、DFLD は消去されません。DPM-Bn では、OFTAB が指定されていると、FILL= は無視され、FILL=NULL と見なされます。

### C' '

文字 ' ' は装置フィールドを埋めるために使用されるデフォルトです。空白文字は、有効な印刷可能文字である X'40' と解釈されます。

### C'c'

文字 'c' で装置フィールドを埋めます。3270 ディスプレイ装置では、X'3F' より小さな値を指定しておく、制御文字は X'00'、その他の非図形文字は X'40' に変更されます。その他の装置では、FILL=C'c' に X'3F' より小さな値が指定されると、その指定は無視され、デフォルトで X'3F' と見なされます (これは FILL=NULL 指定と同等です)。


C'c' を X'36' と指定すると、その値が X'0' または X'40' に変更されます。これは X'36' が有効な印刷可能文字ではないためです。

### NULL

フィールドを充てん文字で埋めないことを意味します。3270 と SLU 2 ディスプレイ以外の装置では、装置フィールドがメッセージ・データでいっぱいにならないと、その行が「圧縮行」となります。

**PT** 3270 と SLU 2 ディスプレイ以外の装置では、NULL と同じです。3270 と SLU 2 ディスプレイでは、装置フィールド (DFLD) をいっぱいに満たさない出力フィールドがあると、その後ろにプログラム・タブ文字が挿入され、フィールドの既存のデータを消去します。

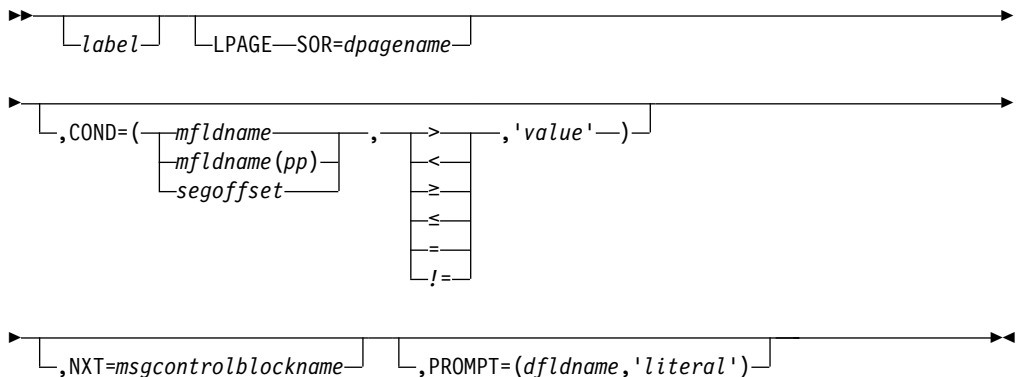
関連資料:

 MFS 出力メッセージの形式 (アプリケーション・プログラミング)

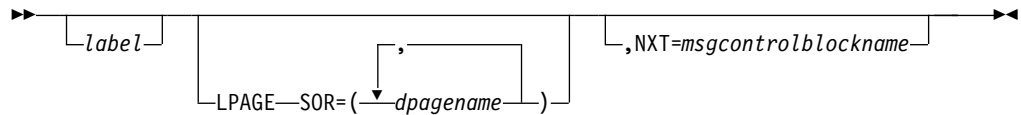
## LPAGE ステートメント

LPAGE ステートメントの使用はオプションです。1 つの論理ページを構成するセグメント・グループを定義します。

### MSG TYPE=OUTPUT のときの形式



## MSG TYPE=INPUT のときの形式



### パラメーター

#### label

このステートメントを一意的に示すための 1 から 8 文字の英数字名を指定します。

#### SOR=

この論理ページの装置形式を定義している DPAGE ステートメントの名前を指定します。TYPE=INPUT で、入力メッセージを作成するためのデータ・ソースとして複数の DPAGE が使われるときは、複数の *dpagename* を指定できます。

#### COND=

条件付きテストを指定します。テストが成功なら、この LPAGE に続くセグメント定義とフィールド定義が、この論理ページの出力編集に使用されます。論理ページの最初のセグメントの指定部分が調べられ、それが指定されたリテラル値より大きい (>)、小さい (<)、以上か (≥)、以下か (≤)、等しい (=)、等しくない (≠) を見て、編集にこの LPAGE を使うかどうか判断されます。MSG 定義の最後の LPAGE ステートメントには、COND= は不要です。

調べる部分の指定には、フィールド名 (*mfldname*)、フィールド・オフセット (*mfldname(pp)*) (*pp* は、指定されたフィールド内でのオフセット値)、セグメント・オフセット (*segoffset*) のいずれかを使用します。*mfldname(pp)* を使用する場合、*pp* は 1 より大きいと等しくなければなりません。比較の長さは、指定されたリテラルの長さと同じです。先行の MSG ステートメントで OPT=3 が指定されている場合、調べる対象は MFLD ステートメントで定義されている 1 つのフィールドの内部でなければなりません。

*segoffset* の指定では、ゼロを起点とするオフセット値を使用します。したがって、オフセット指定では、セグメントの LLZZ を忘れてはなりません (つまり、最初のデータ・バイトはオフセット 4 にあります)。

*pp* の指定では、指定されたフィールドについて 1 を起点とするオフセット値を使用します (つまり、フィールド中最初のデータ・バイトはオフセット 0 ではなく 1 にあります)。

指定の *mfldname* が ATTR=YES で定義されているときは、*pp* オフセットを使用しなければなりません。指定するオフセットの最小値は 3 です。つまり、2 バイトの属性に続いて、最初のデータ・バイトはオフセット 3 にあります。

ATTR=*nm* が指定されているときは、オフセットの最小値が *nm* の 2 倍に 1 を加えた値になります。したがって、ATTR=2 なら *pp* は少なくとも 5、ATTR=(YES,2) なら *pp* は少なくとも 7 です。

すべての LPAGE で条件付きテストが失敗に終わったときは、この MSG 定義の最後の LPAGE が編集に使用されます。

LPAGE 選択にコマンド・データ・フィールドを使うとき、つまり、/FORMAT *modname...(data)* とするときは、LPAGE *mfllname* パラメーターで指定する MFLD が、MOD の関連 LPAGE の最初 8 バイト内になければなりません。

#### **NXT=**

この論理ページが処理される時、次のメッセージのマッピングに使用するメッセージ記述子の名前を指定します。この名前は、先行 MSG ステートメントに指定されているどの *NXT=msgcontrolblockname* にも優先します。

#### **PROMPT=**

DFLD の名前を指定します。MFS は、出力メッセージの最後の論理ページをフォーマット設定しているとき、指定されたリテラルをその DFLD に挿入します。いったん表示されたプロンプト・リテラルは、FILL=NULL であれば、応答によって画面の再フォーマット設定が促されない限り画面上にとどまります。

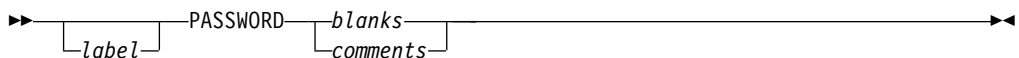
### **PASSWORD ステートメント**

PASSWORD ステートメントは、IMS パスワードとして使う 1 つ以上のフィールドを指定します。

PASSWORD ステートメントを使用する場合、入力 LPAGE または MSG 定義では、PASSWORD ステートメントと関連 MFLD を、最初の SEG ステートメントより前に置かなければなりません。PASSWORD ステートメントの後ろには最大 8 個の MFLD ステートメントを指定できますが、パスワード全体の長さが 8 文字を超えることはできません。充てん文字は X'40' でなければなりません。オプション 1 または 2 のメッセージでは、編集後のデータの最初の 8 文字が IMS パスワードとして使用されます。オプション 3 のメッセージでは、編集後の最初のフィールドのデータ内容が IMS パスワードとして使用されます。

3270 入力では、DFLD ステートメントでもパスワードを定義できます。2 つのパスワード方式を共に使用した場合は、MSG 定義で指定されたパスワードが使用されます。

#### フォーマット



#### パラメーター

##### **label**

このステートメントを一意的に示すための 1 から 8 文字の英数字名を指定します。

### **SEG ステートメント**

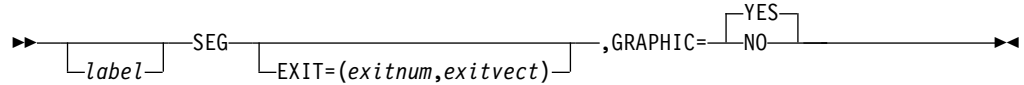
SEG ステートメントは、メッセージ・セグメントを指定するもので、アプリケーション・プログラムが複数セグメント・メッセージの処理を要求している場合のみ必要とされるステートメントです。

出力メッセージ・セグメント群は、指定されたキュー・バッファの長さを越えることができません。TYPE=INPUT の MSG で、入力メッセージの宛先が単一セグメントのコマンドまたはトランザクションと定められているときは、定義するセグ

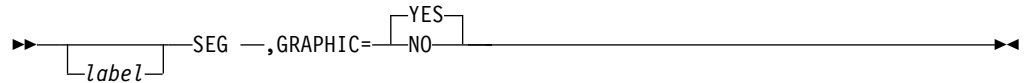


メントは 1 つにしてください。複数のセグメントを定義し、単一セグメントのコマンドまたはトランザクションの入力にその定義を使用するときは、入力が編集後にはセグメントが 1 つだけになるよう十分に注意してください。

### MSG TYPE=INPUT のときの形式



### MSG TYPE=OUTPUT のときの形式



### パラメーター

#### label

このステートメントを一意的に示すための 1 から 8 文字名を指定します。

#### EXIT=

このメッセージ・セグメントに対するセグメント編集出口ルーチンのインターフェースを記述します。 *exitnum* は出口ルーチン番号、*exitvect* は、このセグメントに対して出口ルーチンが呼び出されたときに、それに引き渡される値です。*exitnum* は 0 から 127、*exitvect* は 0 から 255 です。DIV ステートメントで NOSEGEXIT が指定されていない限り (DPM 装置のみ)、入力セグメントの処理が完了すると、SEG 出口が呼び出されます。

#### GRAPHIC=

MSG TYPE=INPUT で、宛先定義がこのセグメントでの上段変換を要求している場合に、IMS はそれを行うか (YES)、行わないか (NO) を指定します (TRANSACTION マクロまたは NAME マクロの EDIT= パラメーターを参照)。デフォルトは YES です。入力セグメント・データが非図形形式 (バック 10 進数、EGCS、バイナリー、など) になっているときは、GRAPHIC=NO を指定してください。GRAPHIC=NO とすると、このセグメント内の MFLD では FILL=NULL が無効となります。

GRAPHIC=YES が指定され、かつ入力メッセージの宛先定義で上段変換が要求されているときは、次のリストに示す変換が行われます。

変換前 変換後

a から z

A から Z

X'81' から X'89'

X'C1' から X'C9'

X'91' から X'99'

X'D1' から X'D9'

X'A2' から X'A9'

X'E2' から X'E9'

GRAPHIC=YES と定義されているセグメント中に、FILL=NULL と指定された MFLD があるときは、圧縮によって 16 進文字 X'3F' がセグメントから排除されます。SEG ステートメントで GRAPHIC=NO と FILL=NULL が指定されていると、非図形データ・ストリーム中にあるすべての X'3F' が圧縮によってセグメントから排除され、望ましくない結果が生じることがあります。非図形データは、出力では固定長出力フィールドとして送られることが望ましく、FILL=NULL の使用はお勧めできません。

MSG TYPE=OUTPUT での GRAPHIC= キーワードは、DPM にしか適用されません。IMS アプリケーション・プログラムからのデータに非図形制御文字 (X'00' から X'3F') が含まれているとき、それをブランクで置き換えるか (YES)、置き換えないか (NO) を指定します。デフォルト値は YES です。NO を指定すると、IMS アプリケーション・プログラムから送られてくるすべてのビット・ストリングが、無修正のまま MFS を通過して、リモート・プログラムに送られます。

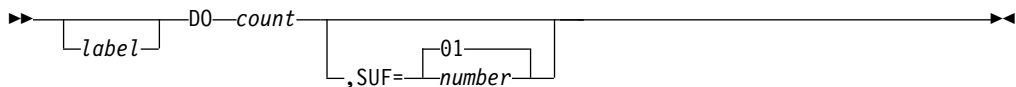
制約事項: GRAPHIC=NO が指定されていると、オプション 1 と 2 を使用している IMS アプリケーション・プログラムは、LPAGE の内部にあるセグメントを省略することができません。また、ヌル文字 (X'3F') を使用しているセグメントの内部にあるフィールドを切り捨てたり、省略したりすることもできません。

## DO ステートメント

DO ステートメントは ENDDO ステートメントと対で用いられ、これらのステートメント間で MFLD ステートメントの反復生成が行われます。

DO の使用はオプションですが、DO を含んでいるメッセージは、必ずそれと対をなす ENDDO を含んでいなければなりません。

### フォーマット



### パラメーター

#### label

1 から 8 文字の英数字名を指定できます。使用されません。

#### count

後続の MFLD ステートメントの生成回数を指定します。指定できる最大 count は 99 です。99 を超える count が指定されると、右端の 2 桁だけが使用され (例えば、103 が指定されれば、03 が使用され)、エラー・メッセージが出ます。

#### SUF=

1 桁または 2 桁の接尾部を指定します。生成される最初の MFLD ステートメント・グループの MFLD label と dfldname にこの接尾部が付加されます。デフォルトは 01 です。MFS は、後続のステートメント・グループが生成されるごとに、接尾部の値を 1 ずつ大きくしていきます。



使用されると、そのラベルが 6 文字で切り捨てられ、2 桁のシーケンス番号が付加されて、8 文字の名前を形成します。この場合、エラー・メッセージは出ません。

### ***dflname***

DEV ステートメントまたは DFLD ステートメントを使用して定義された装置フィールド名を指定します。その装置フィールドから入力データが取り出され、あるいはそこへ出力データが書き込まれます。メッセージ出力制御ブロックの定義でこのパラメーターを省略すると、アプリケーション・プログラムから与えられたデータが出力装置上に表示されません。MFS の反復生成機能 (DO ステートメントと ENDDO ステートメント) を使用するときには、*dflname* を最大 6 文字に制限してください。ステートメントが反復生成されるごとに、2 桁のシーケンス番号 (01 から 99) が *dflname* に付加されます。指定する *dflname* が 6 バイトを超えているとき反復生成を行うと、*dflname* が 6 文字で切り捨てられ、そこに 2 桁のシーケンス番号が付加されて、8 文字の名前を形成します。この場合、エラー・メッセージは出ません。このパラメーターは、次のいずれかの形式で指定できます。

### ***dflname***

入力データの取り出し元または出力データの書き込み先となる装置フィールド名を指定します。

### **'literal'**

入力メッセージにリテラル値を挿入するときは、ここに指定できます。

### **(*dflname*, 'literal')**

TYPE=OUTPUT の場合、指定の DFLD に書き込むリテラル・データを記述します。この形式を使用するときは、アプリケーション・プログラムから与えられる出力メッセージ・セグメントにリテラルのためのスペースを割り振っておいてはなりません。

TYPE=INPUT の場合、メッセージ・フィールドに書き込むリテラル・データを記述します。このメッセージ・フィールドに入るべきデータを装置から受け取らない場合に、このリテラル・データがメッセージ・フィールドに書き込まれます。DEV ステートメントの PFK パラメーターでこの *dflname* が使用されると、このリテラルが PF キーのリテラルまたは制御機能で常に置き換えられます。しかし、PFK パラメーターでこの *dflname* が指定されていても、その PF キーが使用されないと、MFLD ステートメントに指定されていたリテラルがメッセージ・フィールドに移動されます。物理ページングが使用されるときは、リテラルがフィールドに挿入されるものの、その論理ページの最後の物理ページが表示されるまでは処理されません。

どちらの場合も、LTH= オペランドが指定されていると、その LTH= 指定に合わせて、リテラルの長さについて切り捨てまたは埋め込みが行われます。指定されたリテラルの長さが、定義されているフィールドの長さに満たないと、リテラルは、TYPE=OUTPUT ならブランクで埋め込まれ、TYPE=INPUT なら指定の充てん文字 (FILL=) で埋め込まれます。TYPE=INPUT で充てん文字が指定されていないときは、そのリテラルは、ブランク (デフォルト) で埋め込まれます。リテラル値の長さは、256 バイトを超えないようにしてください。

**(dfldname, system-literal)**

システム・リテラルのリストからいずれかの名前を指定します。システム・リテラルは通常のリテラルと同じ働きをしますが、リテラル値が、装置への転送に先立つフォーマット設定で作成される点が異なります。LTH=、ATTR=、JUST= の各オペランドは指定できません。この形式を使用するときは、アプリケーション・プログラムから与えられる出力メッセージ・セグメントにリテラルのためのスペースを割り振っておいてはなりません。

次の表に、システム・リテラルとその長さおよび形式を示しています。

表 17. システム・リテラルの長さ形式

システム・リテラル名	作成されるリテラル		
	長さ	フォーマット	注
LTSEQ	5	<i>nnnnn</i>	1
LTNAME	8	<i>aaaaaaaa</i>	1
TIME	8	HH:MM:SS	
DATE1 または YYDDD	6	YY.DDD	
DATE2 または MMDDYY	8	MM/DD/YY	
DATE3 または DDMMYY	8	DD/MM/YY	
DATE4 または YYMMDD	8	YY/MM/DD	
DATE1Y4 または YYYYDDD または DATEJUL	8	YYYY.DDD	
DATE2Y4 または MMDDYYYY または DATEUSA	10	MM/DD/YYYY	
DATE3Y4 または DDMMYYYY または DATEEUR	10	DD/MM/YYYY	
DATE4Y4 または YYYYMMDD または DATEISO	10	YYYY/MM/DD	
LPAGENO	4	<i>nnnn</i>	2
LTMSG	14	MSG WAITING Qx	3

表 17. システム・リテラルの長さ形式 (続き)

		作成されるリテラル	
システム・リテラル名	長さ	フォーマット	注

注:

1. LTSEQ は、論理端末の出力メッセージのシーケンス番号です。作成される値は、論理端末デキュー・カウント + 1 となります。IMS コールド・スタートまたは /NRESTART BUILDQ に続く最初の出力メッセージのシーケンス番号は 00001 です。IMS で作成されるメッセージには、この番号を変えないものがあります。

LTNAME は、論理端末 (LTERM) 名です。このメッセージは、その LTERM 用にフォーマット設定されます。

端末入力に応じて IMS 制御領域が作成するメッセージ (エラー・メッセージ、ほとんどのコマンド応答) には、LTSEQ も LTNAME も入りません。これらのメッセージは、IMS メッセージ出力記述子 DFSMO1 を使用します。与えられる値は、それぞれ 00000 とブランクです。

2. LPAGENO では、メッセージの現在の論理ページ番号がシステム・リテラルとして与えられます。この番号は、オペレーター論理ページ要求として入力されたページ番号に対応しており、それに対して作成されるリテラルは 4 桁の番号です。先行ゼロはブランクに変換されます。
3. LTMSG では、この出力メッセージが端末に送信されたときに、その端末のメッセージ・キューに既にほかのメッセージが待機していると、「MSG Waiting Qx」というリテラル (x はメッセージ・キュー番号で、1、2、3、4 のいずれか) が LTMSG フィールドで送信されます。現行キュー以外のどのキューにもメッセージがなければ、ブランクが LTMSG フィールドで送信されます。

待機中のメッセージは、一般に、現在のメッセージがデキューされると送信されます。その場合、Q1 で待機中のメッセージは無条件で送信されます。Q2 で待機中のメッセージは、端末が排他モードであれば、Q1 のメッセージが送信される時に送信されます。Q2 で待機中のメッセージは、送信が会話状況によって阻止されない限り、また、Q3 または Q4 で待機中のメッセージは、送信が排他状況または会話状況によって阻止されない限り、送信されます。ほかのキューでメッセージが待機しているとき、端末が会話状態にあれば、会話を一時保留にしてそのメッセージを表示できます。また、端末が排他モードにあれば、端末を排他モードから切り替えることでそのメッセージを表示できます。応答モード・トランザクションを入力中であれば、端末から応答モード・トランザクション入力を入力する前に、そのメッセージを表示できます。

このシステム・リテラルは、会話型モードで使用することをお勧めします。ISC サブシステムでの使用はお勧めできません。

### (,SCA)

この出力フィールドを、出力装置上には表示されないシステム制御域として定義します。このようなフィールドは、1 つの論理ページ (LPAGE) に 1 つだけで、しかも、そのページの最初のメッセージ・セグメントになければなりません。論理ページが定義されていないときは、SCA フィールドは 1 つしか定義できず、しかも出力メッセージの最初のセグメントになければなりません。この指定は、先行 MSG ステートメントで TYPE=OUTPUT が指定されている場合のみ有効です。

### LTH=

入力時にアプリケーション・プログラムに引き渡すフィールド、または出力時に

アプリケーション・プログラムから受け取るフィールドの長さを指定します。デフォルト値は最小値で 1、最大値は 8000 です。(メッセージの最大長が 32767 を超えてはなりません。)

入力フィールドの定義には  $(pp,nn)$  の形式を使用できます。ただし、 $(pp,nn)$  形式を使用するときは、フィールド名を最初の定位置パラメーターに指定しなければなりません。 $pp$  には、入力データ・フィールドのどのバイトをメッセージ・フィールドの最初のデータ・バイトと見なすかを指定します。例えば、 $pp$  を 2 とすれば、入力データの第 1 バイトが無視され、第 2 バイトがこのフィールドの最初のバイトになります。 $pp$  の値は 1 以上でなければなりません。 $nn$  には、アプリケーション・プログラムに渡すフィールドの長さを指定します。

定位置パラメーターに  $(SCA)$  を指定したときは、 $LTH=$  の値には少なくとも 2 を指定してください。

定位置オペランドにリテラルを指定したときは、 $LTH=$  を省略できます ( $TYPE=INPUT$ )。その場合は、*literal* に指定された長さが使用されます。リテラル・フィールドに  $LTH=$  を指定すると、指定されたりテラルがその長さに合わせて切り捨てられたり、ブランクで埋め込まれたりします。DO ステートメントと ENDDO ステートメントの間に MFLD ステートメントがあると、その MFLD ソース・ステートメントに  $LTH=$  指定があるかどうかにかかわらず、生成される MFLD ステートメントには長さの値が印刷されます。

#### JUST=

データ・フィールドを左寄せ (L) または右寄せ (R) し、装置形式制御ブロックが必要とするデータ量または提供するデータ量に応じて、右端または左端を切り捨てます。デフォルトは L です。

#### ATTR=

アプリケーション・プログラムが 3270 属性と拡張属性 ( $nn$ ) を変更してよいか (YES)、よくないか (NO) を指定します。

YES なら、3270 属性データ用に 2 バイトを確保しておかなければなりません。出力時には、ここがアプリケーション・プログラムによって埋められ、入力時にはブランクに初期化されます。この 2 バイトは、 $LTH=$  指定に入れておく必要があります。

$nn$  に指定する値は、動的に変更できる拡張属性の数です。 $nn$  は、1 から 6 の値をとることができます。無効な値を指定すると、デフォルトで 1 となります。属性 1 つにつき 2 バイトを拡張属性データ用に追加確保する必要があります。出力時には、ここがアプリケーション・プログラムによって埋められ、入力時にはブランクに初期化されます。これらの属性バイトは、MFLD  $LTH=$  指定に入れておく必要があります。

以下の例は、有効な ATTR= 指定と、各指定で予約しなければならないバイト数を示します。

MFLD     ,ATTR=(YES, $nn$ )  
          2 + (2 ×  $nn$ )

MFLD     ,ATTR=(NO, $nn$ )  
          2 ×  $nn$

MFLD     ,ATTR=( $nn$ )  
          2 ×  $nn$

**MFLD** ,ATTR=YES  
2

**MFLD** ,ATTR=NO  
0

出力メッセージで、定位置パラメーターによってリテラル値が指定されていると、ATTR=YES と *nn* は無効です。

他の IMS ISC サブシステムに送信されるフィールドの属性は、受信側サブシステムの形式で ATTR= がどう指定されていても、MFS によって入力データとして扱われます。例えば、ATTR=(YES,1),LTH=5 と定義されているメッセージ・フィールド (MFLD) の内容が次のようであるとします。

00A0C2F1C8C5D3D3D6

受信側サブシステムの MFLD が ATTR= 指定なしの LTH=9 と定義されていると、アプリケーション・プログラムは次の内容を受け取ります。

00A0C2F1C8C5D3D3D6

受信側サブシステムの MFLD が ATTR=(YES,1) で、LTH=13 と定義されていると、アプリケーション・プログラムは次の内容を受け取ります。

4040404000A0C2F1C8C5D3D3D6

受信側サブシステムの MFLD が ATTR=(YES,1) で、LTH=5 と定義されていると、アプリケーション・プログラムは次の内容を受け取ります。

4040404000A0C2F1C8

属性データの上段変換を阻止するには、入力 SEG ステートメントで GRAPHIC=NO を指定してください。

#### **FILL=**

装置から受け取ったデータの長さがこのフィールドの長さに満たないとき、フィールドの残りを埋めるのに使う文字を指定します。このフィールドへのデータがないときも、この文字が埋め込まれます (ただし、MSG ステートメントでオプション 3 が指定されている場合を除きます)。このオペランドは、TYPE=INPUT の場合のみ有効です。デフォルトは X'40' です。

#### **X'hh'**

16 進表現 *hh* の文字で、フィールドを埋めます。 FILL=X'3F' は FILL=NULL と同じです。

#### **C'c'**

文字 *c* でフィールドを埋めます。

#### **NULL**

フィールドから欠落しているデータ分だけ左へメッセージ・セグメントを圧縮します。

#### **EXIT=**

このメッセージ・フィールドに対するフィールド編集出口ルーチンのインターフェースを記述します。 *exitnum* には出口ルーチン番号を指定します。 *exitvect* は、このフィールドに対して出口ルーチンが呼び出されたとき、それに引き渡される値です。 *exitnum* の値は 0 から 127、 *exitvect* の値は 0 から 255 です。 MFS 編集後の (しかし、オプション 1 と 2 では NULL 圧縮前の) フィ



ールドのアドレスが、そのフィールドに対して定義されているベクトルと共に編集出口ルーチンに渡されます。(DPM 装置に対して NOFLDEXIT が指定されていると、出口ルーチンは呼び出されません。) 出口ルーチンは、戻りコードとして 0 から 255 の値を返します。MFS は、各セグメントに対して返されてきたコードのうち最大の値を保管しておき、セグメント編集ルーチンが使用できるようにします。同じ MFLD ステートメントに 'literal' が指定されていると、EXIT= は無効です。

生成された **MFLD** ステートメントの印刷:

EXEC ステートメントのパラメーター・リストに **COMP** を指定しておく、生成された MFLD ステートメントがシンボリック・ソース形式で印刷されます。

こうすることで、中間テキスト・ブロックを解釈することなく、MFLD ステートメント生成の結果を見ることができます。

生成された MFLD ステートメントごとに、次の各項目が印刷されます。

- 生成されたステートメント・シーケンス番号と、それに続く + (正符号)。正符号は、その MFLD ステートメントが DO ステートメント処理の結果として生成されたものであることを表します。
- MFLD ステートメント・ラベル (あれば)。付加された接尾部を含みます。
- ステートメント演算子である MFLD。
- *dflname* (あれば)。付加された接尾部を含みます。
- ECGS リテラルでは、G、SO、SI がありません。すべての指定を印刷できるだけのスペースがないと、リテラルが切り捨てられます。切り捨てがあったことは、リテラルの一部に続いて省略符号 (...) が示されることでわかります。この省略符号は切り捨てられた部分を表します。
- システム・リテラル名 (あれば)。

*dflname* とリテラルが共にあるときは、括弧でくくられます。

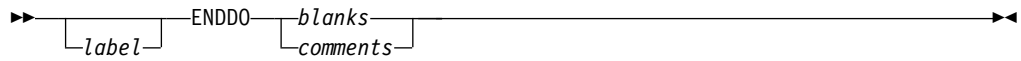
- (,SCA) (あれば)。
- フィールド長。LTH=*nnnn* (または LTH=(*pppp,nnnn*)) の形式が使われます。
- JUST=L または R (あれば)。
- ATTR=YES (あれば)。
- ATTR=*nn* (あれば)。

これ以外のオペランドは、たとえソース MFLD ステートメントに指定されていても印刷されません。

## **ENDDO** ステートメント

ENDDO ステートメントは、反復生成すべき MFLD ステートメントのグループの終わりを示します。

生成された MFLD ステートメントは、ENDDO ステートメントの直後に印刷されます。ENDDO は、DO ステートメントが指定されている場合に必要です。



**label**

1 から 8 文字の英数字名を指定できます。使用されません。

**MSGEND ステートメント**

MSGEND ステートメントは、メッセージの入力定義または出力定義の終わりを示します。定義の最後のステートメントとして必要です。

これが実行依頼されたジョブの終わりなら、この後ろにさらに END コンパイル・ステートメントが必要です。



**label**

1 から 8 文字の英数字名を指定できます。使用されません。

**フォーマット定義ステートメント**

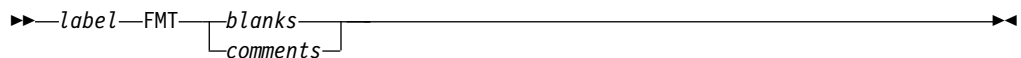
フォーマット定義ステートメントには、FMT ステートメント、DEV ステートメント、DIV ステートメント、DPAGE ステートメント、PPAGE ステートメント、DO ステートメント、RCD ステートメント、DFLD ステートメント、ENDDO ステートメント、および FMTEND ステートメントがあります。

**FMT ステートメント**

FMT ステートメントは、フォーマット定義を開始し、それに名前を付けます。このフォーマット定義には 1 つ以上の装置形式が含まれますが、その装置形式間の違いは、DEV ステートメントで指定される装置タイプとフィーチャーだけです。

フォーマット定義に含まれる各装置形式では、装置またはリモート・プログラムに送信され、あるいはそこから受信されるデータのレイアウトが指定されます。

フォーマット



パラメーター

**label**

1 から 8 文字の英数字名。MSG ステートメントの SOR= オペランドにあるメッセージ記述子によって参照される名前であり、必須です。

label に指定された名前は、作成された装置出力形式および装置入力形式のブロックが IMS.FORMAT ライブラリーに格納される時、そのメンバー名の一部となります。

DEV TYPE=DPM-An で、DIV OPTIONS=MSG なら、label に指定された名前は、出力メッセージ・ヘッダーに含まれるデータ名としてリモート・プログラムに送信されます。

DEV TYPE=DPM-Bn で、DIV OPTIONS=(MSG,DNM) なら、label に指定された名前は、DD FM ヘッダーに含まれるデータ構造名として他のサブシステムに送信されます。

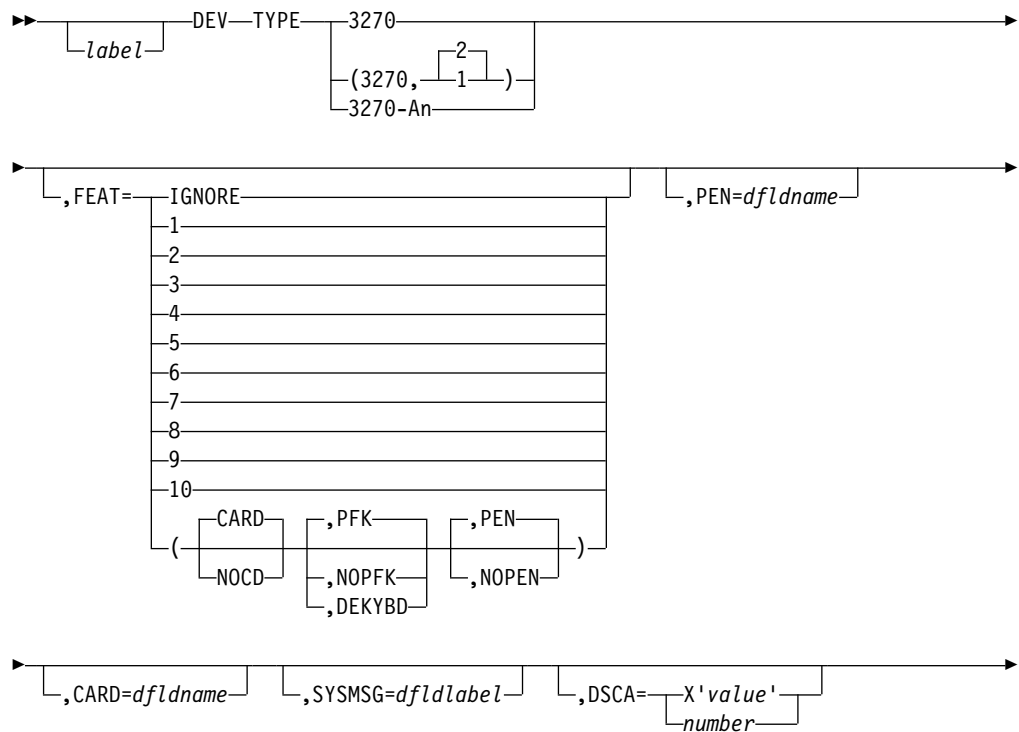
### DEV ステートメント

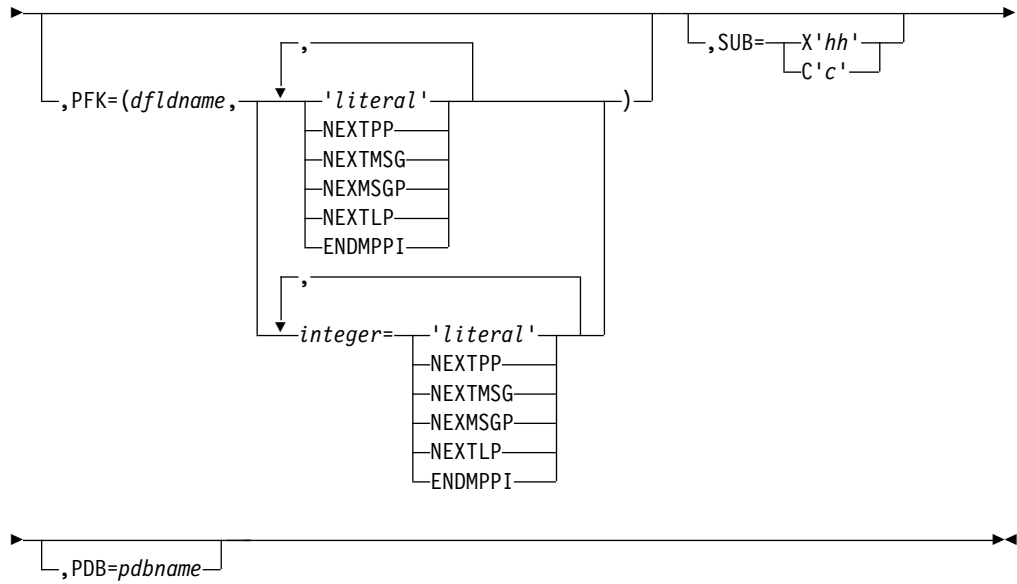
DEV ステートメントは、ある装置の装置特性、またはある装置タイプの特徴を定義します。

この DEV ステートメントには DFLD ステートメントが続き、指定された特性に基づいてマッピングされます。このマッピングは、次の DEV ステートメントまたは FMTEND ステートメントまで続きます。DPM 装置では、DEV ステートメントに DPM プログラム・タイプ番号と、フィーチャー・セット番号 (こちらはオプション) が指定されます。

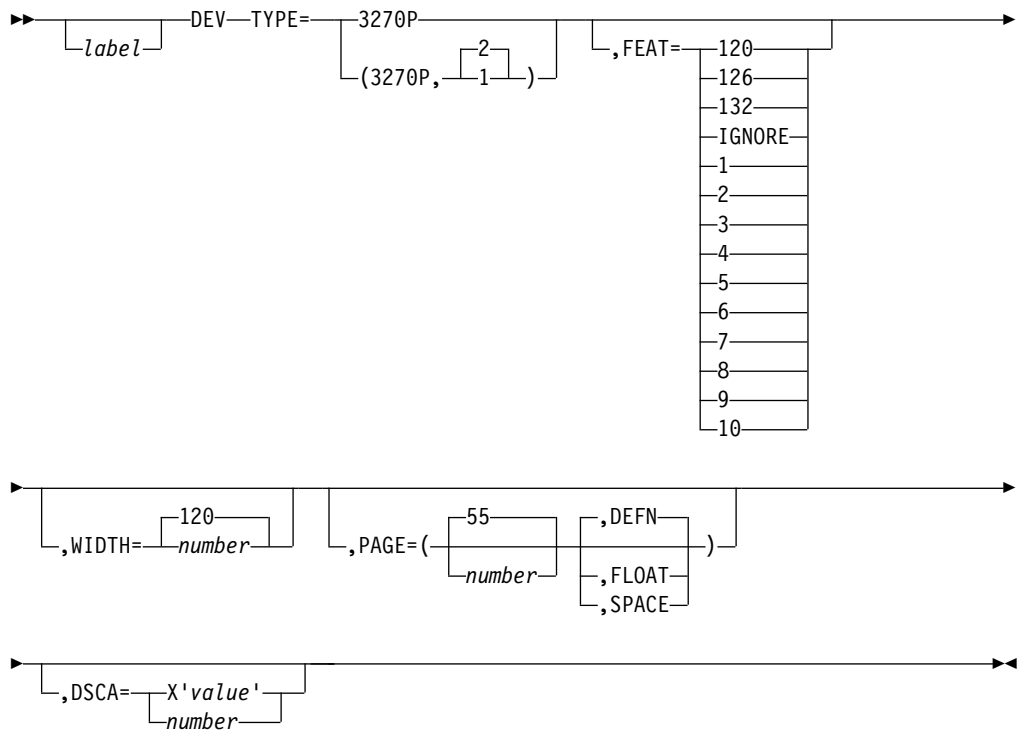
重要: DEV ステートメントを使用する前に、TYPE= オペランドの説明をお読みください。

### 3270 ディスプレイのときの形式

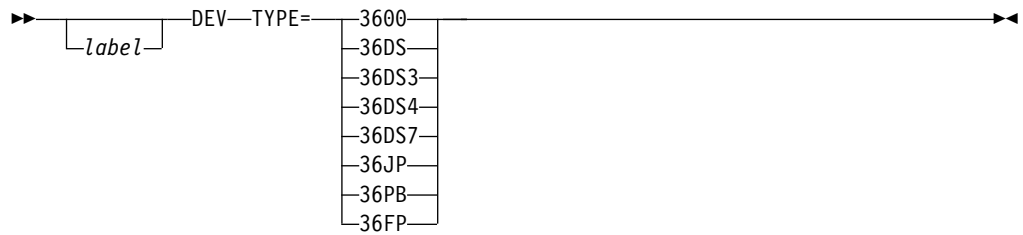




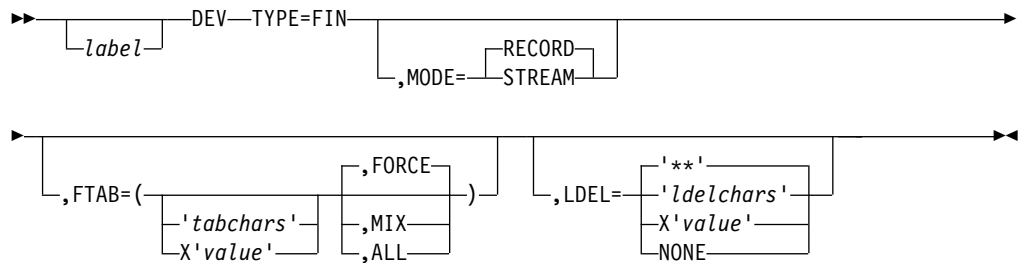
### 3270 プリンターのときの形式



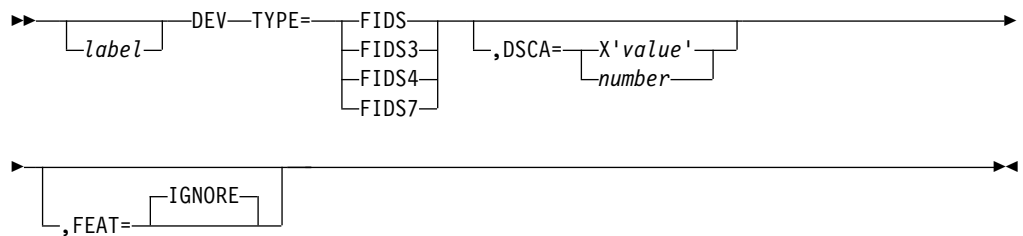
金融機関ワークステーション (3600 または 4700) のときの形式



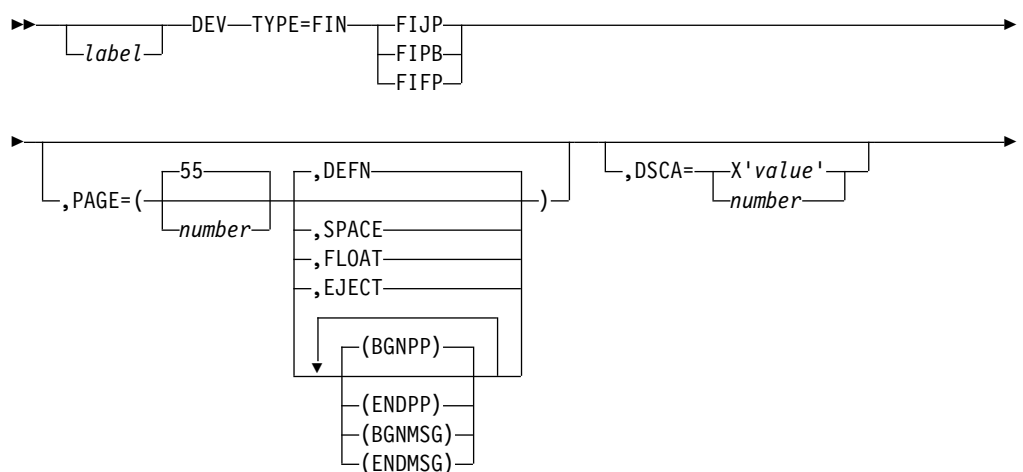
DEV TYPE=FIN のときの形式

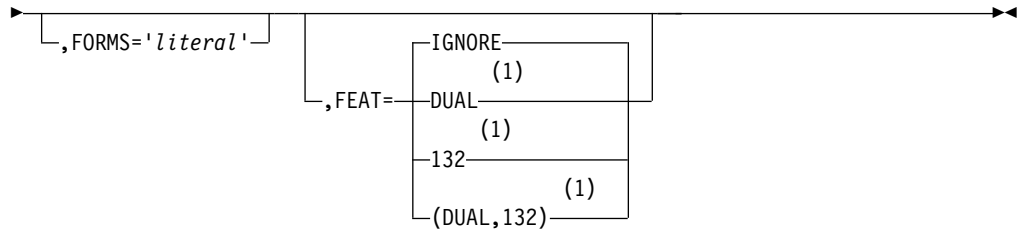


DEV TYPE=FIDS、FIDS3、FIDS4、FIDS7 のときの形式



DEV TYPE=FIJP、FIPB、FIFP のときの形式

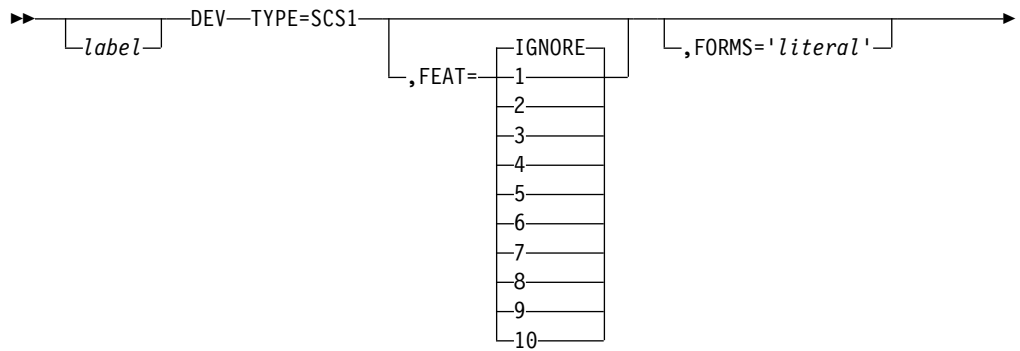




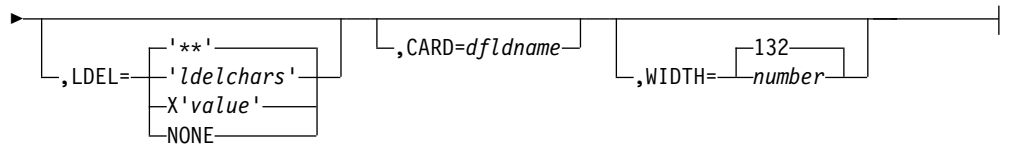
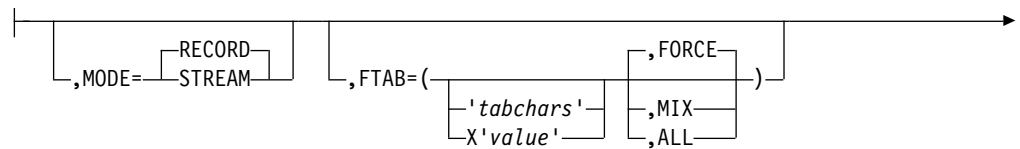
注:

1 FIFP のみ

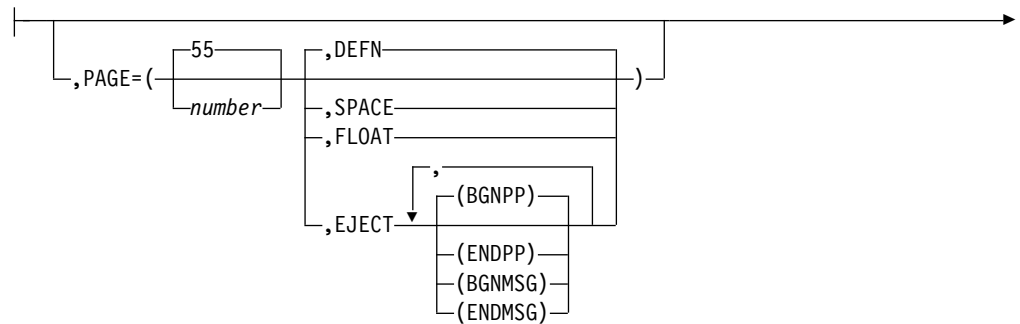
**SCS1** のときの形式

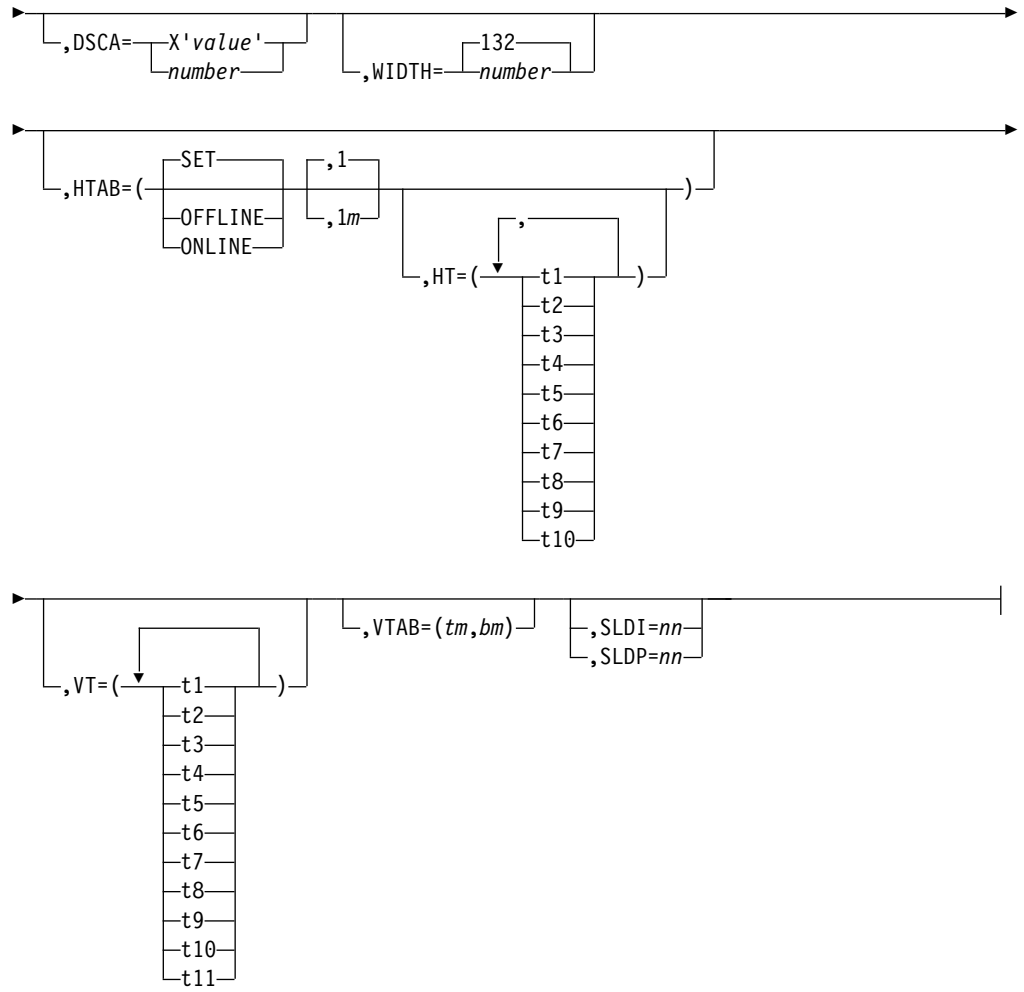


**<FOR INPUT>:**

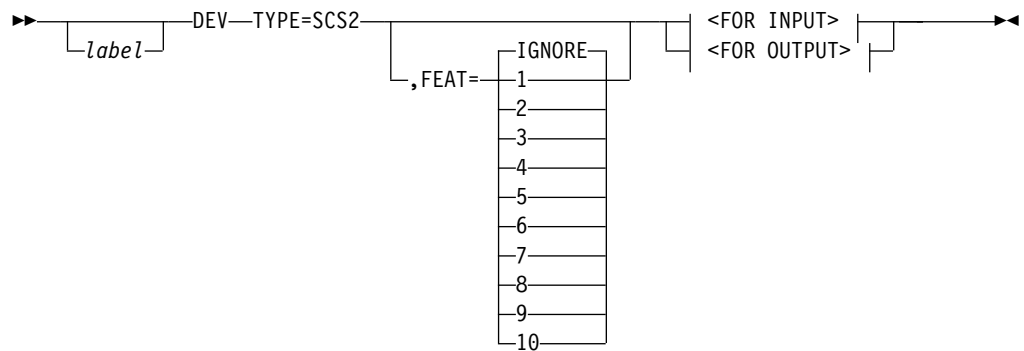


**<FOR OUTPUT>:**

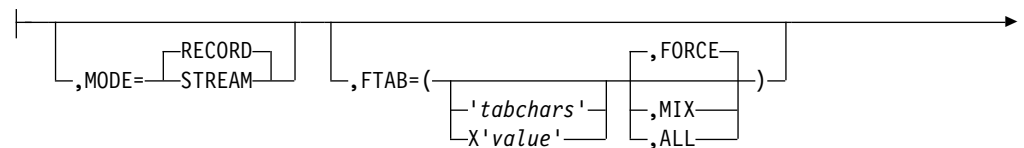


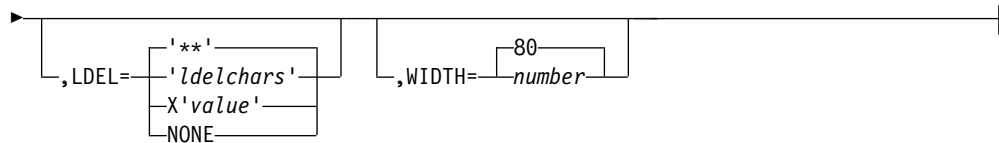


### SCS2 のときの形式

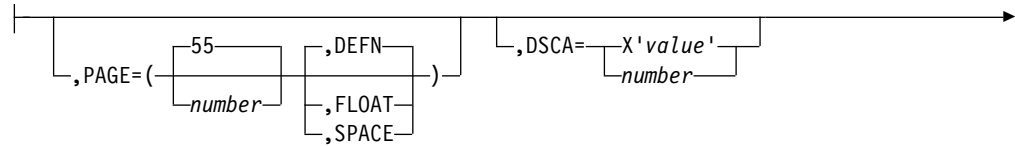


### <FOR INPUT>:

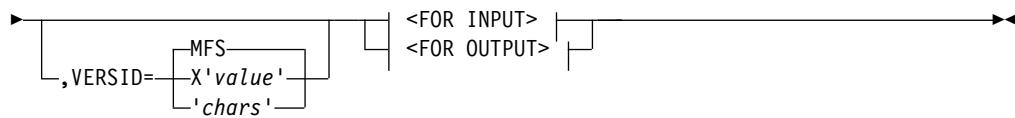
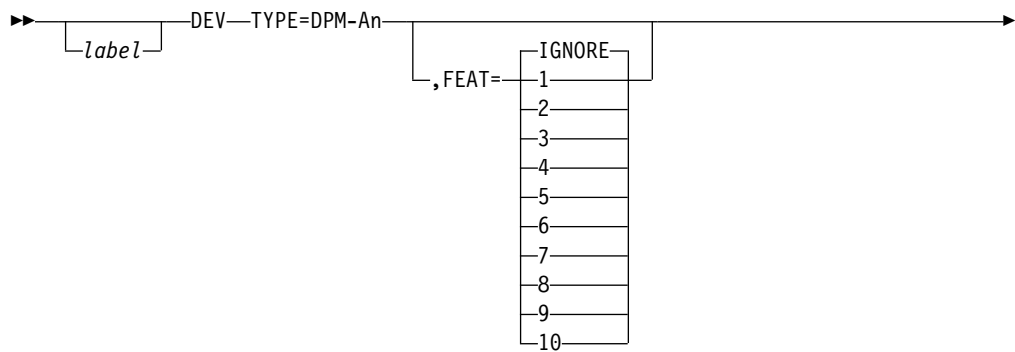




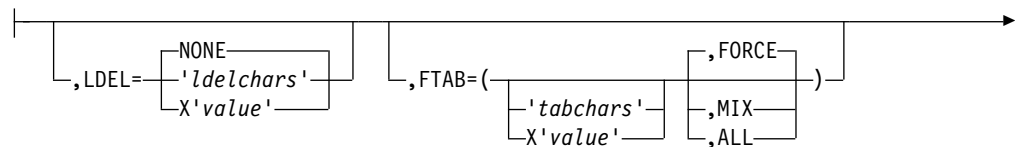
**<FOR OUTPUT>:**



**DPM-An のときの形式**

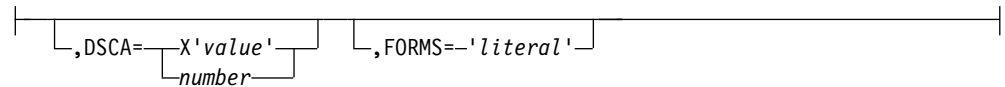


**<FOR INPUT>:**

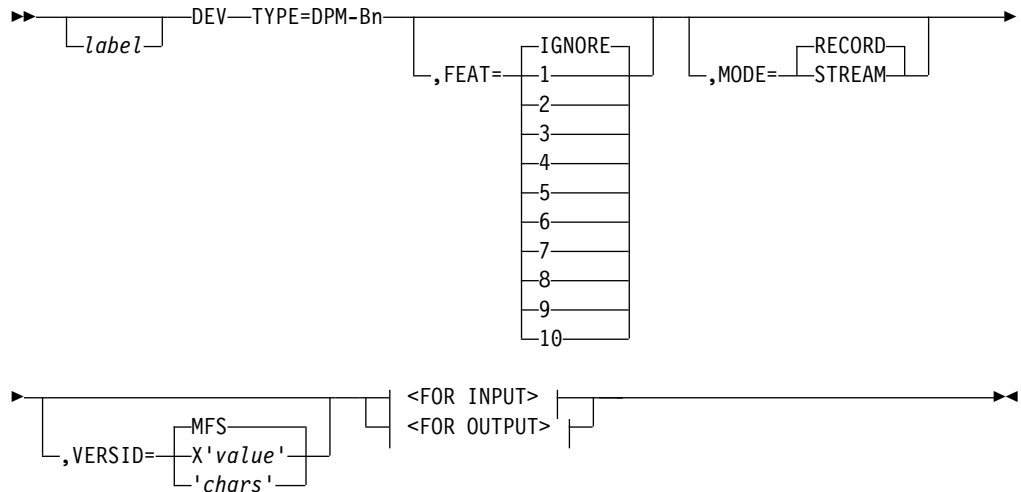




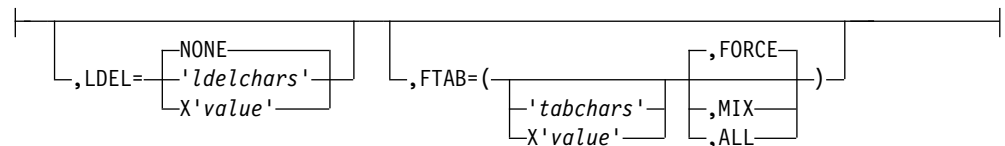
## <FOR OUTPUT>:



## DPM-Bn のときの形式



## <FOR INPUT>:



## <FOR OUTPUT>:



## パラメーター

### label

このステートメントを一意的に示すための 1 から 8 文字の英数字名を指定します。指定はオプションです。

### TYPE=

このフォーマット記述を使用する装置の装置タイプと型式番号を指定します。3275 に接続した 3284-3 プリンターは、TYPE=3270P としてのみサポートされます。3284-3 のフォーマットの定義で指定する型式番号は、関連する 3275 の型式番号です。

TYPE=3270-An は、IMS システム定義時に定められた画面サイズを持ち、フィークャー番号が  $n=1-15$  の 3270 および SLU 2 ディスプレイのシンボル名で

す。この指定があると、MFS 言語ユーティリティーは MFS 装置特性テーブル (DFSUDT0x) を読み、画面サイズを取り出します。

TYPE=DPM-Bn は、装置を ISC ノードとして指定します。n で指定される装置タイプは、システム定義 TERMINAL マクロのコンポーネント指定 (COMPT=) と一致しなければなりません。

装置と型式に基づいて、次のとおり指定してください。

**TYPE=** 装置と型式

**3270,1** 3275-1

3276-1、11 (IMS システム定義で 3270 型式 1 と定義)

3277-1

3278-1 (IMS システム定義で 3277 型式 1 と定義)

SLU 2 (480 文字)

**3270,2** 3275-2 SLU 2 (1920 文字)

(IMS システム定義時に 'mod 2' と定義された、画面区域が 1920 文字のディスプレイ)

**3270-An**

3270-An (IMS システム定義時に TYPE=3270-An と定義されたすべての 3270 または SLU 2 ディスプレイに適用されます)

3270-An として定義できる 3270 装置、ならびに装置タイプのシンボル名と画面サイズの標準的関連付けを、以下に示しておきます。

装置 画面サイズと定義

**3180** 24×80 画面サイズ、3270-A2 と定義

**327X-1、11**

12×80 画面サイズ、3270-A1 と定義

**327X-2、12**

24×80 画面サイズ、3270-A2 と定義

**327X-3、13**

32×80 画面サイズ、3270-A3 と定義

**327X-4、14**

43×80 画面サイズ、3270-A4 と定義

**3278-5**

27×132 画面サイズ、3270-A7 と定義

**3290** 62×160 画面サイズ、3270-A8 と定義

または

24×80 画面サイズ、3270-A2 と定義

**5550** 3270 漢字エミュレーションまたは 3270 PC。24×80 画面サイズで、3270-A2 と定義

**3270P,1**

3284-1

3286-1  
 3287 (480 文字印刷、SLU 1 や SLU 4 としては接続されていない)  
 3289 (480 文字印刷、SLU 1 や SLU 4 としては接続されていない)

**3270P,2**  
 3284-2  
 3286-2  
 3287 (1920 文字印刷、SLU 1 や SLU 4 としては接続されていない)  
 3289 (1920 文字印刷、SLU 1 や SLU 4 としては接続されていない)

**FIN** 金融機関アプリケーション・プログラム (入力のみ)

**FIDS** 金融機関表示コンポーネント (6×40。例えば、3604-1 または -2)

**FIDS3**  
 金融機関表示コンポーネント (12×40。例えば、3604-3)

**FIDS4**  
 金融機関表示コンポーネント (16×64。例えば、3604-4)

**FIDS7**  
 金融機関表示コンポーネント (24×80。例えば、3604-7)

**FIJP** 金融機関ジャーナル印刷装置

**FIPB** 金融機関通帳印刷装置

**FIFP** 金融機関事務管理印刷装置

**SCS1** 次のコンソール・キーボード・プリンター  
 NTO  
 3771  
 3773  
 3774  
 3775  
 3776  
 5553  
 5557  
 SLU 1 (印刷データ・セットまたはバルク・プリンター)  
 SLU 4  
 3289 と 3287 (IMS に SLU 1 として接続しているとき)

**SCS2** 3521 カード・パンチ  
 3501 カード・リーダー  
 2502 カード・リーダー  
 SLU 1 (送信データ・セット)  
 SLU 4

**DPM-An**  
 SLU P (n は 1 から 15)

## DPM-Bn

ISC (n は 1 から 15)

### MODE=

フィールド・スキャンをどう行うかを指定します。デフォルト値は RECORD です。MODE= は、DPM-An では入力のみ、DPM-Bn では入力と出力で有効です。DPM-Bn で、入力モードと出力モードが同じでないときは、各 DIV ステートメントの前に DEV ステートメントが必要です。

### RECORD

フィールドは、装置またはプログラムから送られてくる個々のレコード (装置からの 1 行、リモート・プログラムからの 1 伝送) の内部にあるがままに定義されます。DPM-Bn では、可変長、可変ブロック化 (VLVB) フォーマットのレコードにはレコード・モードを指定しなければなりません。

### STREAM

フィールドは、連続したフィールド・ストリームとして定義されます。つまり、レコードの境界は MFS スキャンに影響しません。フィールドは、所定の順序で入力される限り、レコード間で分割されていても構いませんし、どのレコードから入力されていても構いません。DPM-Bn では、チェーニングした要求/応答単位 (RU) にはストリーム・モードを指定しなければなりません。

### FTAB=

フィールド・タブ (FTAB) 文字を指定します。入力したデータの長さが定義されたフィールド長に満たないとき、あるいはフィールドに入るべきデータがないときは、ユーザーまたはリモート・プログラムはこの文字を使って入力フィールドを終わらせることができます。

- FIN、DPM-An、DPM-Bn では、最大 8 個の FTAB 文字 (16 個の 16 進数字) を指定できます。少なくとも 1 文字 (2 個の 16 進数字) を指定してください。
- SCS1 では、最大 4 個の FTAB 文字 (8 個の 16 進数字) を指定できます。NL、LF、HT、および VT の各文字は常に FTAB 文字ですから、指定する必要はありません。
- SCS2 では、最大 3 個の FTAB 文字 (6 個の 16 進数字) を指定できます。NL、CR、LF、HT、および VT の各文字は常に FTAB 文字ですから、指定する必要はありません。しかし、入力がカード・リーダーから行われるときは、ホレリス・コードがカードにパンチされていないと、MFS はその文字を受け取ることができません。

FTAB 文字が定義されていない場合、各装置入力フィールドは定義されたとおりの長さであると見なされます。レコード・モードでは、レコードの終わりに達すると、現行フィールドが終了し、そのレコードで定義されているすべての後続レコードが装置データなし (メッセージ充てん) として処理されます。ストリーム・モードでは、入力メッセージを構成するすべての伝送が伝送の境界に影響されない 1 つのデータ・フィールド・ストリームとして扱われます。DPM 入力では、FTAB が定義されていなかったり使用されなかったりすると、各装置入力フィールドは定義されたとおりの長さであると見なされます。ただし、NULL=DELETE が指定されている場合を除きます。NULL=DELETE の場合

は、フィールドの末尾に NULL があるか、フィールド全体が NULL であると、定義された長さまでメッセージ充てん文字でフィールドへの埋め込みが行われます。

このオペランドで FTAB 文字を定義するときは、FORCE、MIX、ALL のいずれかも指定できます。デフォルトは FORCE です。

#### **FORCE**

ユーザーまたはリモート・プログラムが FTAB 文字を入力するまで、FTAB が不要であることを表します。レコード・モードでは、1 つのフィールドに FTAB を使用したら、現行レコード中の残りの各フィールドも、その長さにかかわらず、FTAB で終わらせなければなりません。ストリーム・モードでは、1 つのフィールドに FTAB を使用したら、そのメッセージ中の残りの各フィールドも FTAB で終わらせなければなりません。

#### **MIX**

FTAB はなくて構いませんが、入力データが定義されたフィールド長に満たないときは、その入力フィールドを終わらせるのに使用できます。

#### **ALL**

フィールド長にかかわらず、すべてのフィールドを終わらせるのに FTAB を使用します。ただし、特定のモード(MODE=) 依存状態を除きます。レコード・モードでは、レコードで定義もしくは入力されている最後のフィールドには FTAB が不要です。ストリーム・モードでは、メッセージで定義もしくは入力されている最後のフィールドには FTAB が不要です。

#### **LDEL=**

2 個の文字または 4 個の 16 進数字を指定します。これを入力データ中のレコードの最後の 2 文字として入力すると、そのレコードが廃棄されます。

NONE を指定すると、IMS はレコード削除処理をバイパスします。ただし、最初のレコードだけは例外で、これは最後の 2 文字がアスタリスク (\*\*) なら常に削除されます。DPM 装置では NONE がデフォルトです。その他の装置では、\*\* がデフォルトです。

#### **PAGE=**

出力パラメーターを次のとおり指定します。

##### ***number***

プリンター装置では、1 印刷ページの印刷行数を指定します。カード装置では、DPAGE または物理ページ (DFLD ステートメントで *pp* パラメーターが使用されている場合) 当たりのパンチ・カード枚数を指定します。この値は妥当性検査に使用されます。指定する値は 1 以上、256 未満でなければなりません。デフォルトは 55 です。

SCS1 プリンターでは、VTAB= が指定されていると、PAGE= の最小値が 3 になります。

#### **DEFN**

DFLD ステートメントで定義されているとおりに行を印刷し、カードにパンチします (出力ページから行/カードを取り除いたり、追加したりしません)。

**SPACE**

どの出力ページにも、*number* パラメーターで指定されたとおりの行数/カード枚数を含みます。

**FLOAT**

フォーマット設定後にデータを含んでいない (すべてブランクまたは NULL の) 行/カードを削除します。

3270P 装置と SCS1 装置では、データを含まない (ブランクか NULL だけの) 行でも、次の条件下では削除できません。

- その行に、1 つ以上の行密度設定 (SLDx) 指定があるとき
- 拡張属性を持つと定められているフィールドが複数の行にまたがっているとき

**EJECT**

プリンター装置で、ページ替え操作を実行します。EJECT は、TYPE=FIJP、FIPB、FIFP、SCS1 のいずれかでのみ有効です。SCS1 に EJECT を指定すると、MFS は垂直紙送り機構があるものと見なします。サブリストのデフォルトは BGNPP です。

ページ替えを実行できる時点は、次のサブリストに示すとおりです。

**BGNPP**

出力の各物理ページの前でページ替えします。

**ENDPP**

物理ページの印刷が終わるたびにページ替えします。

**BGNMSG**

メッセージ中のデータの印刷を開始する前にページ替えします。

**ENDMSG**

メッセージ・データをすべて印刷した後でページ替えします。

**DSCA=**

この装置形式を使用する出力メッセージのデフォルト・システム制御域 (DSCA) を指定します。メッセージ出力記述子に SCA が指定されていて、その指定内容に DSCA と矛盾するところがあるときは、DSCA が優先します。通常は、両方の SCA に指定されている機能が共に実行されます。SCS1 と SCS2 では、DSCA= オペランドが指定されていても、無視されます。3270P では、DSCA= オペランドが指定されていても無視されます。ただし、「音響装置アラーム」のビット設定は例外で、DSCA/SCA オプションでこのビットが指定されていると、当該装置に送られます。TYPE=DPM-An または DPM-Bn では、DSCA/SCA 情報が DFLD 定義で要求されている場合のみ、それがリモート・プログラムまたは ISC サブシステムに送られます。

ここに指定する値は 65535 を超えない 10 進数か、X'hhhh' でなければなりません。値が指定されると、それが内部で X'hhhh' に変換されます。

DSCA フィールドの 2 バイトは、以下の表または 236 ページの表 20 に示すとおりでなければなりません。

以下の表に、3270 ディスプレイ、SLU 2 装置、TYPE=DPM-An または DPM-Bn での DSCA ビット設定を示します。

表 18. 3270 ディスプレイ、SLU 2 装置、TYPE=DPM-An または DPM-Bn の DSCA フィールドのビット設定

バイト	ビット	設定
0	0-7	0
1	0	1
	1	強制フォーマット書き込み (装置バッファを消去し、すべての必要データを書き込む)
	2	無保護フィールドを消去してから、書き込む
	3	音響装置アラーム
	4	出力を候補ポインターにコピーする。DPM-Bn ではビット 1 から 4 を無視
	5	B'0'- 3270 では、出力を送信するとき、画面を保護する。DPM では、要求時ページングを実行できる。B'1'- 3270 では、出力を送信するとき、画面を保護しない。DPM-B では、自動ページングを実行できる。
	6-7	0。ただし、区画形式モードの 3290 を除く

3275 ディスプレイで、バイト 1 のビット 5 が B'1' (画面を保護しない) に設定されているとき、入力と出力が同時に発生すると (競合)、装置の接続が切断されます。3275 以外の装置では、SCA オプションが無視されます。バイト 1 のビット 5 が B'0' に設定されていれば、アプリケーション・プログラムは SCA 値を B'1' に設定することで、出力の自動ページ化を要求できます。この要求は、出力メッセージの最初の LPAGE の最初のセグメントにある場合のみ、認められます。

バイト 0 がゼロ以外の値であるか、バイト 1 のビット 6 または 7 がゼロ以外の値であると、MFS はその値をゼロに変更します。ただし、区画形式モードの 3290 を除きます。

区画形式モードの 3290 では、バイト 1 のビット 6 に特別な意味があります。出力メッセージの DOF が最後のメッセージの DOF と同じなら、その出力メッセージが送信される前に、DSCA のバイト 1、ビット 6 が検査され、区画の消去/非消去オプションが確認されます。ビット 6 の設定の意味を次の表に示します。

表 19. 3290 区画形式モードのビット設定

バイト	ビット	設定	意味
1	6	B'1'	出力メッセージを送信する前に、すべての区画を消去する
		B'0111'	既存の区画を消去しない

デフォルトは B'0' (消去しない) です。ビット 6 が定義されていれば、既存のすべての区画が消去された後、出力が指定の区画ページング・オプションに従って送信されます。ビット 6 が定義されていなければ、出力が指定の区画ページング・オプションに従って送信され、出力を受信しない区画は、出力送信以前の状態から変わりません。

以下の表に、TYPE=FIDS、FIDS3、FIDS4、FIDS7、FIJP、FIPB、FIFP での DSCA ビット設定を示します。

表 20. TYPE=FIDS、FIDS3、FIDS4、FIDS7、FIJP、FIPB、FIFP での DSCA ビット設定

バイト	ビット	設定
0	0-7	0
1	0	1
	1-2	FIN 出力装置には適用されない
	3	出力メッセージ・ヘッダーに「装置アラーム」を設定
	4	FIN 出力装置には適用されない
	5-7	0

FIN 装置では、バイト 0 がゼロ以外の値であるか、バイト 1 のビット 1、2、5、6、または 7 がゼロ以外の値であると、MFS はその値をゼロに変更します。

バイト 1 のビット 1、2、4 は、3270 と SLU 2 でのみ意味があり、FIN には適用されません。FIN 出力装置用に編集されるメッセージでは、これらのビットがオンになっていても無視されます。

3270 と FIN 装置では、指定された機能が実行されます。DPM 装置では、ユーザー定義の装置フィールド (DFLD) を通じてリモート・プログラムに指定が与えられます。

#### **FEAT=**

この装置またはプログラム・グループのフィーチャーを指定します。

#### **IGNORE**

この装置では、フィーチャーを無視します。

#### **120 | 126 | 132**

3284 と 3286 装置タイプ (TYPE=3270P) の行長を指定します。

#### **CARD**

装置には 3270 オペレーター ID カード読取機構が付いています。NOCD は、CARD フィーチャーがないことを表します。

#### **DEKYBD**

データ入力キーボードが付いています。これがあるということは、PFK フィーチャーがあることも意味します。したがって、DEKYBD を指定したときは、PFK が無効となります。NOPFK は、PFK フィーチャーも DEKYBD フィーチャーもないことを意味します。

#### **PFK**

装置には、PF キーが付いています。NOPFK は、PFK フィーチャーも DEKYBD フィーチャーもないことを意味します。

#### **PEN**

選択ライト・ペン検知機構があります。NOPEN は、PEN フィーチャーがないことを意味します。

#### **DUAL**

FIFP 装置に複式紙送り機構が付いています。

#### **132**

FIFP 装置に拡張印刷機能が付いています。



## 1|2|3|4|5|6|7|8|9|10

SCS1、SCS2、3270P、DPM-An、または DPM-Bn 装置タイプに、ユーザー定義のフィーチャーが付いています。

SCS1、SCS2、3270P の各装置では、特殊な装置特性を持っている装置を FEAT= でグループ分けできます。例えば、印刷位置が最大 80 で、VFC を持たない装置を FEAT=1 で 1 つのグループとし、印刷位置が最大 132 で、VFC を持つ装置を FEAT=2 でもう 1 つのグループとする、などです。最小限の機能しか持っていない装置は、FEAT=IGNORE で 1 つのグループにまとめてください。3270P 装置に WIDTH= を指定するときは、FEAT=(1...10) も指定しなければなりません。FEAT=(1...10) を指定して、WIDTH= を指定しないと、WIDTH= はデフォルトで 120 となります。

DEV TYPE=DPM-An または DPM-Bn では、ユーザー定義装置形式グループを FEAT= で指定します。こうすることで、共通のフィーチャーや依存関係を持つ複数のプログラムを同時に選択できます。

IGNORE 指定時には、FEAT= オペランドでは他の値をコーディングできません。システム定義時に TERMINAL マクロで FEAT=IGNORE を指定していないときは、MSG ステートメントの、IGNORE 指定を持つ装置形式の SOR= オペランドで IGNORE を指定しなければなりません。FEAT=IGNORE でない限り、FEAT= には、IMS システム定義時に TERMINAL マクロで指定したとおり正確に指定してください。違っていると、DFS057 エラー・メッセージが出されます。3270 装置に対して FEAT=IGNORE または 1-10 を指定した場合でも、PEN=、CARD=、PFK= の各オペランドは指定できます。TYPE=3270P で FEAT=IGNORE なら、MFS は 120 文字の行幅を認めます。

CARD、PFK、DEKYBD、PEN の各フィーチャー値は、3270 ディスプレイでのみ有効です。DUAL は、TYPE=FIFP でのみ有効です。FEAT= オペランドを省略した場合のデフォルト・フィーチャーは、3270 ディスプレイでは CARD、PFK、PEN です。デフォルトの行幅は、TYPE=3270P では 120、TYPE=FIFP では 80 です。

1、2、3、4、5、6、7、8、9、10 は、DEV TYPE=3270、3270P、3270-An、SCS1、SCS2、DPM-An、DPM-Bn でのみ有効な値です。3270 ディスプレイでは、特定のフィーチャーまたはハードウェア・データ・ストリーム依存関係を持つ装置のグループ分けに、FEAT= の指定に 1 から 5 を使用できます。

制約事項: このキーワードの指定は任意です。3270 ディスプレイの他フィーチャー指定との併用はできません。

3290 と 3180 の両方に同じフォーマットを使うときは、FEAT= オペランドに指定する値を装置タイプに合わせて変えなければなりません。各装置タイプ用に選択する FEAT パラメーター値は、IMS システム定義の TERMINAL マクロにも指定されていなければなりません。

FIN、FIDS、FIDS3、FIDS4、FIDS7、FIJP、FIPB では、FEAT が常に IGNORE です。FIFP では、132 と DUAL を指定する場合を除き、IGNORE を使用します。

フィーチャー・オペランド値の指定順序は任意です。また、必要な値を指定するだけで構いません。下線のある値はデフォルトですから、指定を省略できます。垂直リストからは、1 つの値だけを指定してください。

例: FEAT= 指定の用途をいくつか示します。

- TYPE=DPM-A1,FEAT=1 では、DPAGE ページング・オプションとシミュレートされた属性を持つ装置形式のグループを作れます。
- TYPE=DPM-A5,FEAT=2 では、ページング・オプションがなく、ビット・ストリング属性 (MFS によって解釈されない) を持つ装置形式のグループを作れます。
- TYPE=DPM-B1,FEAT=IGNORE では、PPAGE ページング・オプションと最小限のプログラム要件だけを持つ装置形式を指定できます。

#### **PFK=**

最初のサブパラメーターには、PF キー・リテラルまたは制御機能データを収める入力フィールドの名前を指定します。残りのサブパラメーターには、そのフィールドに収めるリテラル・データか、対応 PF キーが押されたときに実行される制御機能を、定位置形式またはキーワード形式で指定します。

最初のサブパラメーターの名前 (PF キー・リテラルまたは制御機能データを含んでいる入力フィールドの名前) は、MFLD ステートメントで参照できます。DEV 定義内の DFLD ステートメントのラベルとして使用してはなりません。残りのサブパラメーターは、定位置形式とキーワード形式のどちらでも指定できます。サブパラメーターをキーワード形式で指定するときは、1 ~ 36 の整数 (1 と 36 を含む) を使用します。重複指定はできません。1 つの DEV ステートメントでは、一方の PFK= オペランド形式 (定位置かキーワードか) しか使用できません。このオペランドは、3270 ディスプレイでのみ有効です。実際の形式ブロックが作成される時点で、各リテラルに埋め込みが行われ、リスト中最大のリテラルの長さまで右端にブランクが追加されます。リテラルの最大長は 256 バイトです。

装置が IMS コピー機能をサポートしているときは、PFK12 でコピー機能が呼び出され、DEV ステートメントの PFK12 定義は無視されます。それ以外では、PFK12 定義が有効です。

FEAT=NOPFK が指定されていると、その指定が PFK に変更されます。PFK は最大 36 個まで、ユーザーにおいて定義できます。

指定できる制御機能は、次のとおりです。

#### **NEXTPP-PAGE ADVANCE**

現在の出力メッセージにおける次の物理ページを要求します。出力メッセージが進行中でなければ、特に応答はありません。

#### **NEXTMSG-MESSAGE ADVANCE**

進行中の出力メッセージがあれば、それをデキューし、キュー中の次の出力メッセージ (あれば) を送信するよう要求します。

#### **NEXTMSGP-MESSAGE ADVANCE PROTECT**

進行中の出力メッセージがあれば、それをデキューし、次の出力メッセージを送信するか、次のメッセージがなければ、そのことを伝える通知メッセージを返すよう、要求します。

#### **NEXTLP-NEXT LOGICAL PAGE**

現在のメッセージにおける次の論理ページを要求します。

#### **ENDMPPI-END MULTIPLE PAGE INPUT**

複数の物理ページから成る入力メッセージの終わりを指定します。

#### **PEN=**

入力フィールドの名前を指定します。スペースまたはヌル指定文字を持つフィールドで即時ライト・ペン検出が起ると、このフィールドにリテラル・データが入ります。リテラル・データは、DFLD ステートメント上で PEN= オペランドにより定義されます (DFLD ステートメントの PEN= オペランドを参照)。この名前は、MFLD ステートメントで参照されることがありますから、この DEV 定義内の DFLD ステートメントのラベルとして使用しないでください。PEN= オペランドは、3270 ディスプレイでのみ有効です。FEAT=NOOPEN が指定されていると、その指定が PEN に変更されます。

スペースまたはヌル指定文字で定義されているフィールドで即時検出が起こったとき、別のフィールドが選択もしくは変更されているか MOD 属性を持っていたり、その DFLD に対して PEN= オペランドが指定されていないかたりすると、PEN= フィールド名に疑問符 (?) が挿入されます。

即時検出が起こらなかつたり、アンパーサンド (&) 指定文字で定義されているフィールドで起こったりすると、PEN= オペランドが、MFLD ステートメントで指定された充てん文字で埋め込まれます。

#### **CARD=**

オペレーター ID カード・データが入力されたとき、それを受け入れる入力フィールドの名前を指定します。この名前は、MFLD ステートメントで参照されることがありますから、この DEV 定義内の DFLD ステートメントのラベルとして使用しないでください。このオペランドは、3270 ディスプレイまたは SCS1 が指定された場合のみ有効です。3270 ディスプレイに対して FEAT=NOCD が指定されていると、それが CARD に変更されます。このカード・フィールド名を参照する入力 MFLD に磁気カード入力のデータが渡される際には、事前にすべての制御文字がデータから取り除かれます。

3270 ディスプレイでは、磁気カード・データと制御文字を入れるのに十分な大きさの無保護フィールドを DFLD ステートメントで定義しておかなければなりません。カーソルをこのフィールドに位置付けてから、カードをリーダーに通し、カード情報を入力してください。カード・データは、DFLD ステートメントに使用された名前ではなく、CARD= フィールド名と論理的に関連付けられます。

TYPE=SCS1 の装置では、オペレーター ID (OID) 文字を持つカード・データだけがこのフィールド名と関連付けられます。OID 文字を持つカードは、データ入力中いつでも入力できます。MFS は、OID 文字を持たないデータを見付けると、それをキーボードから入力されたデータであるかのように扱います。

#### **SYSMSG=**

IMS システム・メッセージの表示先となる装置フィールドがどの DFLD ステートメントで定義されているかを、そのラベルで指定します。このオペランドは、3270 ディスプレイが指定されている場合のみ有効です。この DEV 定義に含まれる各 DPAGE 内のどの物理ページにも、このラベルを持つ DFLD を定義しておいてください。メッセージの切り捨てが起こらないよう、SYSMSG の DFLD は少なくとも LTH=79 でなければなりません。ここで参照される DFLD は、MFLD ステートメントからも参照されることがあります。

#### **FORMS=**

1 から 16 バイトのリテラルを指定します。FIN では、この FMT を使用して装置に送られる各メッセージの出力メッセージ・ヘッダーに、このリテラルが挿

入されます。FIN アプリケーション・プログラムはこのデータに基づいて、そのメッセージに必要な特殊用紙が装置に取り付けられているかどうか、ページ・サイズが正しく、用紙位置決めが適切かどうかを確認できます。

SLU 1 印刷データ・セット・コンポーネントまたは SLU 4 への SCS1 出力では、IMS 出力を受け取るデータ・セットがこのリテラルで指定されます。しかし、IMS に対して SLU 1 または SLU 4 と定義されているプログラマブル型 3770 では、端末がリテラルを無視し、すべての印刷データ・セット出力が SYS.INTR データ・セットに書き込まれます。非プログラマブル型 3770、SLU 1 非 PDS コンポーネント、SLU 4 へのすべての SCS1 出力では、リテラルが無視されます。

装置タイプ DPM-An では、このリテラルが出力メッセージ・ヘッダーに挿入されます。DPAGE または PPAGE ページング・オプションが指定されているときは、このリテラルが、独立に送信される特殊用紙出力メッセージ・ヘッダーの一部となり、その後 (リモート・プログラムからのページング要求に続いて) DPAGE または PPAGE 出力メッセージ・ヘッダーとデータ・レコードが送信されます。デフォルトの MSG オプションが選択されていれば、リテラルを挿入した出力メッセージ・ヘッダーが最初のレコードとして送信され、その後ろにデータ・レコードが続きます。

#### **WIDTH=**

この装置タイプの最大行幅を次のいずれかとして指定します。

- 入力データまたは出力データ 1 行当たりの印刷位置数
- 入力データまたは出力データのカード 1 枚当たりのパンチ位置数
- カード・リーダー入力データのカード幅

デフォルトは、SCS1 入出力では 132、SCS2 入出力では 80、3270P 出力では 120 です。指定する値は、SCS1 では 255、SCS2 では 249 を超えることができません。行幅は、HTAB= キーワード (SCS1 と SCS2 のみ) で左マージン値が指定されているかどうかにかかわらず、1 桁目を基準にして指定されます。指定する行幅は 1 以上でなければなりません。

3270P 装置に WIDTH を指定するときは、FEAT=(1..10) も指定しなければなりません。FEAT=(1..10) を指定して、WIDTH= を指定しないと、デフォルトで WIDTH=120 と見なされます。

#### **HTAB=**

TYPE=SCS1 のとき、次のことを指定します。

- 装置上のどこに MFS が水平タブ・ストップを設定するか
- MFS が出力メッセージにタブ制御文字を挿入して水平タブを行うかどうか、およびその時期。
- 装置上のどこに MFS が左マージンを設定するか。

HTAB= を指定しないと、水平タブは行われず、左マージン位置は 1 桁目と見なされます。

#### **SET | ONLINE | OFFLINE**

MFS が装置の水平方向のフォーマット制御機構を設定します。MFS が装置の水平方向のフォーマット制御機構を設定すると、最大行幅、左右マージン、水平タブ・ストップの各特性が設定されます。HTAB= キーワードが指定されていれば、デフォルトは SET です。

## SET

MFS は水平タブ・ストップを設定しますが、出力メッセージにタブ制御文字を挿入しません。以後の入力には水平タブを実行できます。

## ONLINE

MFS は指定の (HT=) 位置に水平タブ・ストップを設定し、オンライン処理時に水平タブ制御文字を挿入します。

## OFFLINE

MFS は指定の (HT=) 位置に水平タブ・ストップを設定し、フォーマットのオフライン・コンパイル時に水平タブ制御文字を挿入します。

## 1|1m (left margin)

左マージンの桁位置を指定します。デフォルトは 1 です。指定する値は、WIDTH= 値より小さくなければなりません。

## HT=

1 から 10 個の水平タブ・ストップ位置を指定します。1 桁目を基準にして、左マージン値以上、かつ WIDTH= 値未満の値を指定してください。

## VT=

MFS は、指定された位置にタブ制御文字を挿入します。1 から 11 個の垂直タブ・ストップ位置を指定できます。VTAB= が指定されている場合、指定する VT= 値は 1 行目を基準とし、VTAB= キーワードに指定されている下部マージン以下の値でなければなりません。また、VTAB= が指定されていなければ、VT= 値は、PAGE= キーワードに指定されたページ行数以下でなければなりません。最大値は 255 です。255 より大きな値を指定しても、255 と見なされ、エラー・メッセージは出ません。VT= は、TYPE=SCS1 でのみ有効です。PAGE=(n,FLOAT) が指定されていると、VT= は無効です。

VTAB= も指定されている場合に限り、X'00' も有効なタブ・ストップと認められます。

VT= は、VTAB= および PAGE= と共に、ページの垂直方向のフォーマット設定データ・ストリームを形成します。VTAB= キーワードの *tm* は 1 以上で、VT= キーワードの *t1* 未満でなければなりません。VTAB= キーワードの *bm* は、VT= キーワードの *t11* 以上で、PAGE= キーワードに指定された最大ページ長以下でなければなりません。

## VTAB=

SCS1 プリンターでは、上部 (*tm*) と下部 (*bm*) のページ・マージンを指定します。VTAB= は、VT= および PAGE= と共に、ページの垂直方向のフォーマット設定データ・ストリームを形成します。*tm* は 1 以上で、VT= キーワードの *t1* 未満でなければなりません。*tm* の最大値は 253 です。

*bm* は、VT= キーワードの *t11* 以上で、PAGE= キーワードに指定された最大ページ長以下でなければなりません。*bm* は、*tm* より少なくとも 2 大きくなければなりません。VTAB= が指定されている場合、PAGE= 値は 3 以上でなければなりません。

VTAB= キーワードに指定された上部マージン (*tm*) が 1 に等しくないとき、垂直方向フォーマット設定 (SVF) データ・ストリームの後ろに用紙送り (FF) が挿入されます。

PAGE=(n,FLOAT) が指定されていると、VTAB= は無効です。

#### **SLDI=**

SCS1 プリンターでは、出力メッセージの行密度を 1 インチ当たりの行数で指定します。(SLDP= も参照してください)。SLDI= は、DFLD ステートメントでも指定できます。SLDI= 値は 1 から 72 で、対象装置のアーキテクチャーと整合していなければなりません (該当する装置またはコンポーネントの解説書を参照してください)。

SLDI= が DEV ステートメントと DFLD ステートメントの両方に指定されていると、2 つの SLD データ・ストリームが作成されます。1 つはメッセージの先頭で送信され、行密度を設定します。もう 1 つはメッセージ内部 (SLDI= 指定があるフィールドの直前で、すべての垂直タブと改行文字の後ろ) で送信されます。

制約事項: 1 つの DEV ステートメントに SLDI= と SLDP= の両方を指定することはできません。

メッセージ内の SLDI= 指定は、メッセージの先頭で設定された行密度を変更します。この新しい行密度は、明示的にリセットされるまで有効です。

#### **SLDP=**

SCS1 プリンターでは、出力メッセージの行密度を 1 インチ当たりのポイント数で指定します。(SLDI= も参照してください)。SLDP= は、DFLD ステートメントでも指定できます。SLDP= 値は 1 から 72 で、対象装置のアーキテクチャーと整合していなければなりません (該当する装置またはコンポーネントの解説書を参照してください)。

SLDP= が DEV ステートメントと DFLD ステートメントの両方に指定されていると、2 つの SLD データ・ストリームが作成されます。1 つはメッセージの先頭で送信され、行密度を設定します。もう 1 つはメッセージ内部 (SLDP= 指定があるフィールドの直前で、すべての垂直タブと改行文字の後ろ) で送信されます。

制約事項: 1 つの DEV ステートメントに SLDP= と SLDI= の両方を指定することはできません。

メッセージ内の SLDP= 指定は、メッセージの先頭で設定された行密度を変更します。この新しい行密度は、明示的にリセットされるまで有効です。

ヒント: 行密度設定 (SLDx) キーワードを定義する際には、用紙位置決めが正しく行われるようにしてください。SLDx= の定義が不適切だと、用紙位置決めが失われることがあります。

#### **VERSID=**

バージョン ID として任意の 2 文字または 2 バイトの 16 進値を指定します。MFS を指定するか、VERSID キーワードそのものを指定しないと、MFS がバージョン ID を計算します。デフォルトは MFS です。

MFS によるバージョン ID の計算には、FMT 定義がコンパイルされた日の日付とタイム・スタンプが使われます。その値は MFS 言語ユーティリティー出力に印刷されますから、フォーマット定義でそれを参照できます。

#### **SUB=**

MFS が、入力データ・ストリーム中の X'3F' 文字を置き換えるのに使う文字を指定します。このパラメーターに X'3F' を指定するか、パラメーターそのもの

を指定しないとき、また、受信した入力に MFS 編集をバイパスするときは、文字変換が起きません。ここに指定する SUB 文字は、データ・ストリーム中のどこにも含まれてはなりません。つまり、図形文字であってはなりません。

**X'hh'**

入力データ・ストリーム中のすべての X'3F' を、16 進表現が 'hh' となる文字で置き換えます。

**C'c'**

入力データ・ストリーム中のすべての X'3F' を、文字 'c' で置き換えます。

**PDB=**

(区画形式モードの 3290 または 3180 では) 出力メッセージまたは入力メッセージの区画セットを記述するのに使われる、区画記述子ブロックの名前を指定します。このパラメータは、TYPE=3270-An を指定する DEV ステートメントでのみ有効です。

関連概念:

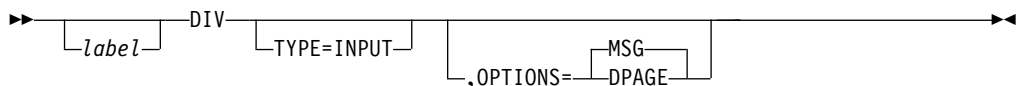
➡ MFS メッセージ形式 (アプリケーション・プログラミング API)

**DIV ステートメント**

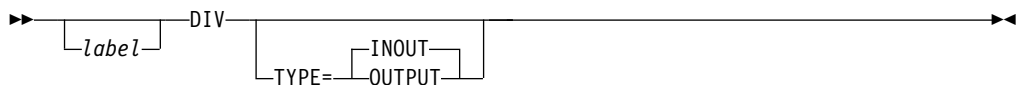
DIV ステートメントは、DIF または DOF 内の装置形式を定義します。フォーマットは入力用、出力用、入出力用があり、複数の物理ページから構成できます。

DEV TYPE=SCS1、SCS2、または DPM-AN では、DIV TYPE=OUTPUT と DIV TYPE=INPUT という 2 つの DIV ステートメントを定義できます。その他の装置タイプでは、DEV ごとに 1 つの DIV ステートメントしか認められません。

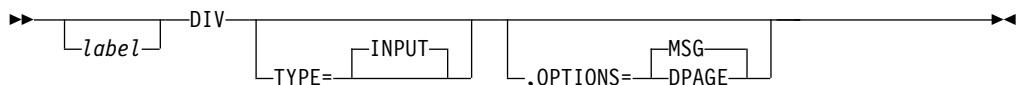
**DEV TYPE=SCS1、SCS2 のいずれかで、DIV TYPE=INPUT のときの形式**



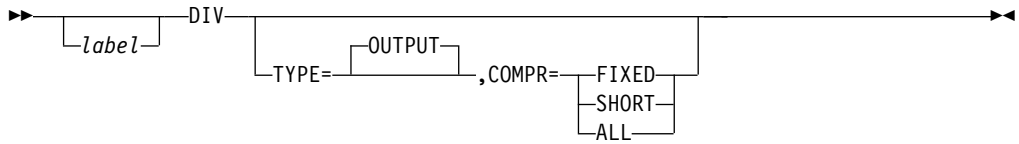
**DEV TYPE=3270 または 3270-An のときの形式**



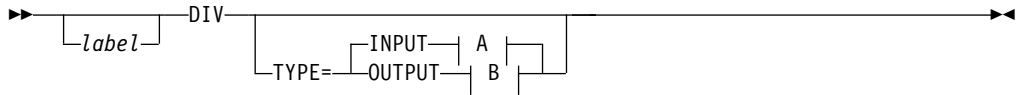
**DEV TYPE=FIN のときの形式**



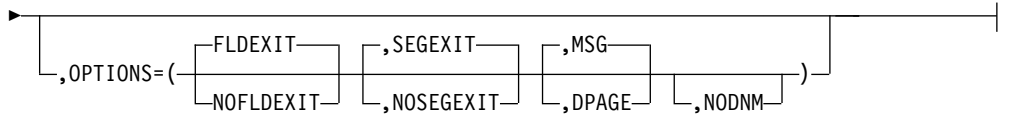
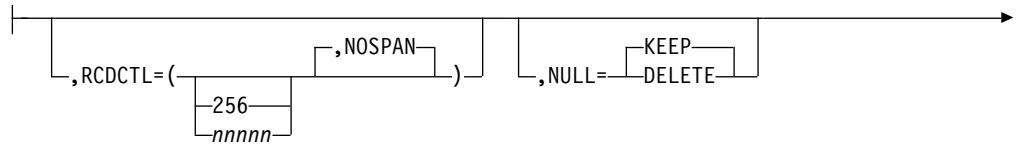
**DEV TYPE=SCS1、SCS2、3270P、FIDS、FIDS3、FIDS4、FIDS7、FIJP、FIPB、FIFP のいずれかで、DIV TYPE=OUTPUT のときの形式**



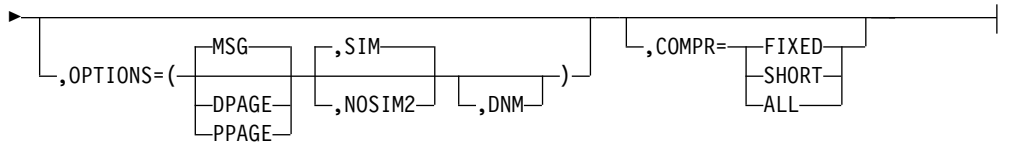
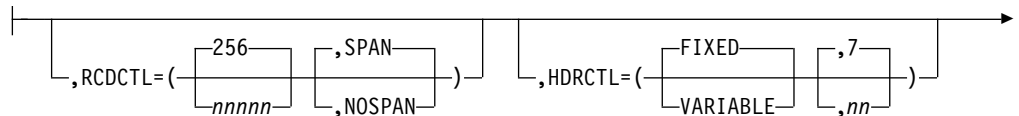
**DEV TYPE=DPM-An のときの形式**



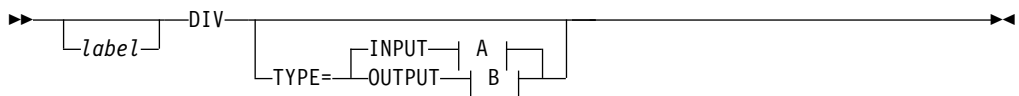
**A:**



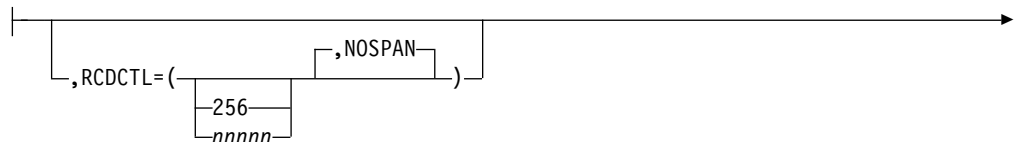
**B:**



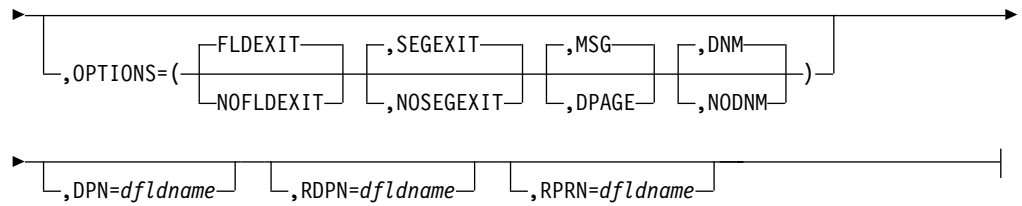
**DEV TYPE=DPM-Bn のときの形式**



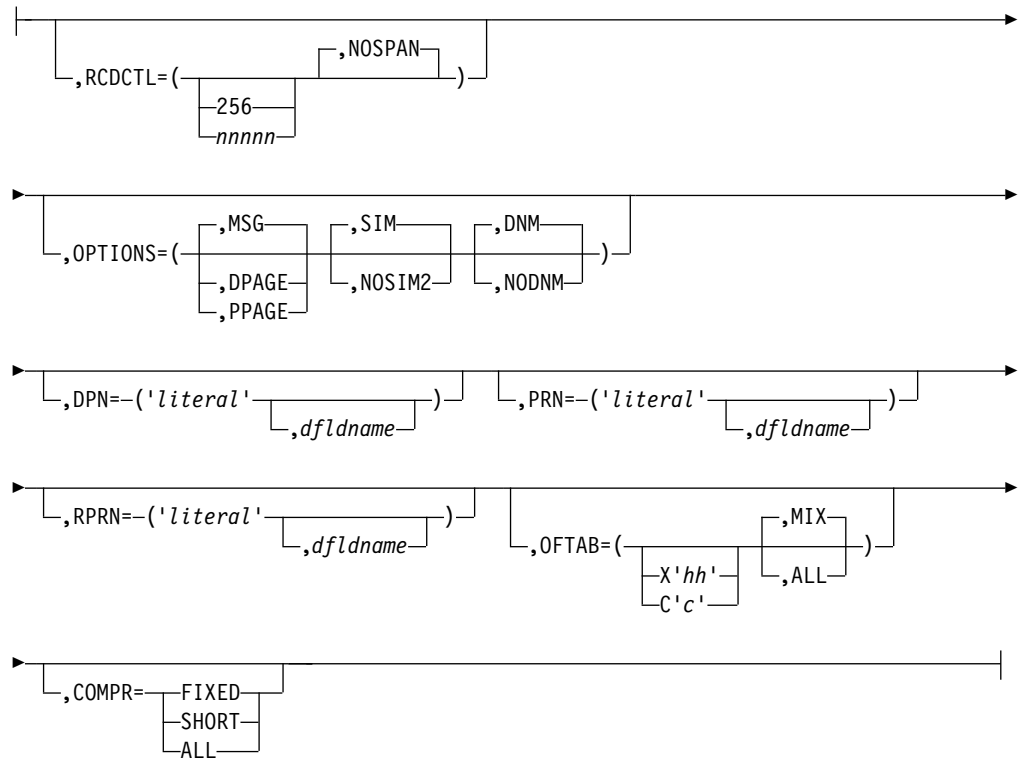
**A:**







**B:**



## パラメーター

### label

このステートメントを一意的に示すための 1 から 8 文字の英数字名を指定します。

### TYPE=

入力専用フォーマット (INPUT)、出力専用フォーマット (OUTPUT)、両用フォーマット (INOUT) を記述します。

DIV TYPE=OUTPUT または TYPE=INPUT が指定されていると、特定の DEV ステートメント・キーワードが適用されます。

例えば、DEV TYPE=SCS1 のとき WIDTH=80 を指定すると、出力時にはフィールドを 1 から 80 桁目に印刷でき、入力時には 1 から 80 桁目から受信できることを意味します。DEV TYPE=SCS2 で WIDTH=80 とすると、カード・リーダーとカード・パンチの両方で位置の数が同じであることを意味します。

DEV TYPE=SCS1 で、WIDTH=80 と HTAB=(SET,5) を指定すると、出力時にはフィールドを 5 から 80 桁目に印刷でき、入力時には 5 から 80 桁目から受信できることを意味します。この場合、入力時の DFLD POS=(1,5) または

POS=5 は、1 桁目を指定し、左マージンを 1 に指定したのと同じことです。データの入力は、現在の左マージンがどこに設定されているかにかかわらず、同じに行われます。

#### **RCDCTL=**

このパラメーターは、DEV ステートメントに MODE=RECORD が指定されている場合のみ有効です。DEV TYPE=DPM-An または DPM-Bn の場合にのみ、入力または出力伝送レコードの最大長を指定します。また DPM-An では、フィールドが複数のレコードにまたがれるか (SPAN)、またがれないか (NOSPAN) を指定します。RCDCTL の値は 32000 以下で、メッセージ出力ヘッダーの長さ以上にする必要があります (DPM-An については、HDRCTL の説明を参照してください)。デフォルト値は 256 です。たとえ同じフォーマット定義の中で RCD ステートメントが使われていても、RCDCTL はレコード定義を作成します。

- DEV TYPE=DPM-An または TYPE=Bn で、DIV TYPE=INPUT の場合

入力フォーマット定義では、フィールドがレコード境界にまたがってはいならず、したがって、RCDCTL 値で指定された長さに収まっていなければなりません。NOSPAN がデフォルトです。

- DEV TYPE=DPM-An または Bn で、DIV TYPE=OUTPUT の場合

指定する RCDCTL サイズは、IMS システム定義時の OUTBUF= マクロで指定された出力バッファ・サイズ以下でなければなりません。RCDCTL サイズが OUTBUF に指定された値より大きいと、1 つのレコードに複数回の出力伝送が必要となる場合があります。リモート・プログラムに好ましくない結果をもたらすことがあります。定義されたレコードにフィールド群がぴったり収まらず、しかも NOSPAN が指定されていると、レコードが完全に埋まらないことがあります。

RCDCTL では、1 つのフィールドがレコード境界にまたがることのできるか (SPAN) (DPM-An のみ)、できないか (NOSPAN) も指定されます。SPAN が指定されると (DPM-An のみ)、レコード境界にまたがるフィールドが発生しますから (しかし、PPAGE 境界にまたがることはありません)、部分フィールドを関連付けたり、それぞれを別個に処理したりするためのロジックを、リモート・プログラムに組み込む必要があります。

NOSPAN が指定されると、すべてのフィールドが完全にレコード内に収まります。指定された RCDCTL 値より大きな長さを持つフィールドはありません。

OPTIONS=MSG では、最初のデータ・フィールドがメッセージの最初のフィールドです。OPTIONS=DPAGE では最初のデータ・フィールドが DPAGE の最初のフィールド、OPTIONS=PPAGE では PPAGE の最初のフィールドです。最初のデータ・フィールドが出力メッセージ・ヘッダーと同じレコードに収まらない場合、OPTIONS=DPAGE または PPAGE が指定されていれば、最初のデータ・レコードは次の伝送で送信されます。出力メッセージ・ヘッダーは単独で送信されます (OPTIONS=MSG では常に単独)。

#### **NULL=**

DEV TYPE=DPM-An かつ DIV TYPE=INPUT の場合にのみ、フィールドの末

尾にあるヌルを MFS に無視させるか (KEEP)、探して置き換えさせるか (DELETE) を指定します。 NULL=DELETE なら、MFS は末尾にヌルを持つ (または、ヌルだけから成る) 入力メッセージ・フィールドを探し、そのヌルをメッセージ定義で指定された充てん文字で置き換えます。

#### **OPTIONS=**

DIV TYPE=INPUT なら、OPTIONS キーワードには、呼び出す出口ルーチン、要求するページングまたはデリバリーの種類、そして DPM-Bn の場合にのみ、データのマッピングに使用する DPAGE データ名の選択を指定します。

DIV TYPE=OUTPUT なら、OPTIONS= キーワードには、要求するページングまたはデリバリーの種類、要求する属性処理の種類、そして DPM-Bn の場合にのみ、データのマッピングに使用する DPAGE データ名の選択を指定します。

DPM 出力メッセージでは、選択するオプションにより、リモート・プログラムまたは ISC サブシステムに送信するレコードの作成方法が決まります。また、処理とロジックが、IMS アプリケーション・プログラムとリモート・プログラムまたは ISC サブシステムとの間で配分されます。

- DEV TYPE=DPM-An または TYPE=DPM-Bn で、DIV TYPE=INPUT の場合

#### **FLDEXIT|NOFLDEXIT**

#### **SEGEXIT|NOSEGEXIT**

この装置タイプからの入力データは、まずリモート・プログラムで部分的に編集してから、IMS に送ることができます。入力フォーマット定義では、このパラメーターによって、MSG 定義の MFLD ステートメントと SEG ステートメントで指定した出口ルーチンを、この DPM フォーマットに対して呼び出すか (それぞれ FLDEXIT と SEGEXIT)、呼び出さないか (それぞれ NOFLDEXIT と NOSEGEXIT) を指定します。NOFLDEXIT または NOSEGEXIT を指定すると、当該出口ルーチンがバイパスされます。FLDEXIT と SEGEXIT がデフォルトです。

#### **MSG**

1 つの DPAGE から入力メッセージを作成できます。

#### **DPAGE**

複数の DPAGE から入力メッセージを作成できます。MFS 定義で複数 DPAGE 入力が必要されていないと、複数の DPAGE からメッセージを作成することはできません。この場合は次のようになります。

1 個の DPAGE が伝送され、そこに当該 DPAGE 定義を超える量のデータが含まれていると、その入力メッセージは拒否され、エラー・メッセージが出されます。

複数の DPAGE が伝送されると、その入力メッセージは拒否され、エラー・メッセージが出されます。

#### **NODNM (DPM-An のみ)**

#### **DNM/NODNM (DPM-Bn のみ)**

DNM (データ名) が指定されるか、デフォルトで DNM と見なされると (DPM-Bn のみ)、次の条件が満たされるとき、その DPAGE が選択されて、現在の (または唯一の) データ伝送のマッピングに使われます。

DPAGE データ名がメッセージ・ヘッダーの DSN パラメーターとして与えられ、かつ

その DPAGE データ名が、定義されている DPAGE データ名に一致するとき

この条件が満たされないと、最後に定義された DPAGE 名がデータのマッピングに使用されます。ただし、その DPAGE が条件付きと定義されている場合を除きます。

NODNM (データ名なし) が指定されると (DPM-An でも DPM-Bn でも)、MFS は受信したデータと COND= パラメーターに条件付きテストを行い、DPAGE を選択します。

- DEV TYPE=SCS1、SCS2、FIN で、DIV TYPE=INPUT の場合

#### MSG

1 つの DPAGE から入力メッセージを作成できます。

#### DPAGE

複数の DPAGE から入力メッセージを作成できます。MFS 定義で複数 DPAGE 入力が要求されていないと、複数の DPAGE からメッセージを作成することはできません。この場合は次のようになります。

- 1 個の DPAGE が伝送され、そこに当該 DPAGE 定義を超える量のデータが含まれていると、その入力メッセージは拒否され、エラー・メッセージが相手サブシステムに送られます。
- 複数の DPAGE が伝送されると、その入力メッセージは拒否され、エラー・メッセージが相手サブシステムに送られます。

- DEV TYPE=DPM-An または TYPE=DPM-Bn で、DIV TYPE=OUTPUT の場合

#### MSG

これがデフォルトです。IMS は 1 つのメッセージ中のすべての DFLD を 1 つのメッセージ・グループとして送信します。メッセージの先頭には出力メッセージ・ヘッダーが置かれ、すべての DFLD が送信されます。DPM-Bn では、ヘッダー中のデータ構造名はあってもなくても構いません。

#### DPAGE

IMS は、1 つの論理ページにグループ化されたすべての DFLD を送信します。この論理ページは、1 つ以上のレコードとして送信されます。DPAGE で PPAGE ステートメントが定義されていると、その PPAGE ステートメントの 1 つ 1 つが新しいレコードを開始します。リモート・プログラムからページング要求があったときは、論理ページが追加送信されます。各論理ページの前には出力メッセージ・ヘッダーが置かれ、DPAGE のラベルがそのヘッダーに挿入されます。DPM-Bn では、DD ヘッダーにデータ構造名を含めるかどうかはオプションで、この構造名は DNM/NODNM の指定によって異なります。

#### PPAGE

IMS は、1 つの表示ページ (PPAGE) にグループ化された DFLD を、1 つのチェーンとして送信します。表示ページは、1 つ以上のレコードから成るグループとして送信されます。リモート・プログラムから IMS へ

ページング要求が送られてくれば、表示ページが追加送信されます。各表示ページの前には出力メッセージ・ヘッダーが置かれ、PPAGE ステートメントのラベルがそのヘッダーに挿入されます。DPM-Bn では、DD ヘッダーにデータ構造名を含めるかどうかはオプションで、この構造名は DNM/NODNM の指定によって異なります。

#### **SIM/NOSIM2**

MFS に属性をシミュレートさせるか (SIM)、させないか (NOSIM2) を指定します。SIM がデフォルトで、MFS は IMS アプリケーション・プログラムによって指定された属性をシミュレートし、そのシミュレートされた属性を、ATTR=YES または YES,nn を指定している対応 DFLD に挿入します。フィールドの最初のバイトが、シミュレートされた属性を入れるのに用いられます。対応 DFLD で ATTR=YES または YES,nn を指定している場合に、MFLD が 3270 属性情報を (その ATTR=YES または YES,nn オペランドによって) その DFLD に与えないときは、フィールドの最初のバイトで空白が送信されます。リモート・プログラムまたは ISC サブシステムのアプリケーション設計者は、シミュレートされた属性をそのプログラムまたはサブシステムの中で解釈しなければなりません。

NOSIM2 が指定されると、MFS は 2 バイトのビット・ストリングをリモート・プログラムまたはサブシステムに送信します。このビット・ストリングは、IMS アプリケーション・プログラムから与えられたままの形で送信されます。3270 拡張バイトがあれば (ATTR=YES,nn)、2 バイトの 3270 属性ストリングに続いて、それがアプリケーション・プログラムから与えられたままの形で送信されます。MFLD から属性情報が提供されないときは、フィールド・データに先行する 2 バイトに 2 進ゼロが挿入され、送信されます。

さらに詳しくは、DFLD ステートメントの ATTR= を参照してください。

#### **DNM (DPM-An のみ)**

DEV ステートメントの FORMS= キーワードで使用し、メッセージ・ヘッダーに入れるリテラルを指定できます。このパラメーターはオプションです。

#### **DNM/NODNM (DPM-Bn のみ)**

DNM を指定するか、デフォルトで使用すると、MFS は次の項目を DD ヘッダーに含めます。

- FMT 名 (OPTIONS=MSG なら)
- DPAGE 名 (OPTIONS=DPAGE なら)
- PPAGE 名 (OPTIONS=PPAGE なら)

NODNM を指定すると、DD ヘッダーにはデータ構造名 (DSN) が含まれません。

#### **HDRCTL=**

DEV TYPE=DPM-An で DIV TYPE=OUTPUT の場合にのみ、出力メッセージ・ヘッダーの特性を指定します。

#### **FIXED**

完全に埋め込まれた出力メッセージ・ヘッダーをリモート・プログラムに送

信します。固定出力メッセージ・ヘッダーの構造は、この FMT 定義に基づいて作成されるすべての DPM 出力メッセージで同じです。基本となる DPM 出力メッセージ・ヘッダーは、長さが 7 で、バージョン ID を含んでいます。

#### VARIABLE

MIDNAME と DATANAME の末尾ブランクを省略し、それに合わせて長さフィールドを調整します。MIDNAME を使用しないときは、MIDNAME フィールド自体もその長さも存在しません。

**nn** ヘッダーの最小長を指定します。つまり、MFS フィールドを持たない基本ヘッダーの長さです。デフォルトは 7 で、DPM の基本メッセージ・ヘッダーの長さです。7 以外の値を指定すると、リモート・プログラムで間違った結果を引き起こすことがあります。

RDPN=、DPN=、PRN=、および RPRN= として参照される各パラメーターは、ISC ATTACH 機能管理ヘッダーと、それと同等の ISC SCHEDULER 機能管理ヘッダーの両方を参照します。

#### RDPN=

DIV TYPE=INPUT では dfldname を指定することにより、推奨する戻り宛先処理名 (RDPN) を、この dfldname を参照する入力メッセージ MFLD に含めることができます。dfldname を指定しないと、入力メッセージに RDPN は挿入されません。

#### DPN=

DIV TYPE=OUTPUT では 'literal' 指定により、このリテラルを出力 ATTACH メッセージ・ヘッダーの DPN として使うよう、MFS に要求できます。リテラルは、8 文字を超えてはなりません。dfldname も指定したときは、その dfldname を参照する MFLD に指定されているデータが、出力 ATTACH メッセージ・ヘッダーの DPN として使用されます。そして、その dfldname を参照する出力メッセージ MFLD が存在しないときに、'literal' が使用されます。dfldname を参照する MFLD 中のデータが 8 文字より大きいと、最初の 8 文字だけが使用されます。

#### PRN=

DIV TYPE=INPUT では dfldname を指定することにより、推奨する基本リソース名 (PRN) を、この dfldname を参照する入力メッセージ MFLD に含めることができます。dfldname を指定しないと、アプリケーション・プログラムへの入力メッセージに PRN は挿入されません。

DIV TYPE=OUTPUT では 'literal' 指定により、このリテラルを出力 ATTACH メッセージ・ヘッダーの PRN として使うよう、MFS に要求できます。リテラルは、8 文字を超えてはなりません。dfldname も指定したときは、その dfldname を参照する MFLD に指定されているデータが、出力 ATTACH メッセージ・ヘッダーの PRN として使用されます。そして、その dfldname を参照する出力メッセージ MFLD が存在しないときに、'literal' が使用されます。dfldname を参照する MFLD 中のデータが 8 文字より大きいと、最初の 8 文字だけが使用されます。

#### RPRN=

DIV TYPE=INPUT では dfldname を指定することにより、推奨する戻り基本リソース名 (RPRN) を、この dfldname を参照する入力メッセージ MFLD に

含めることができます。 dfldname を指定しないと、アプリケーション・プログラムあての入力メッセージに RPRN は挿入されません。

DIV TYPE=OUTPUT では 'literal' 指定により、このリテラルを出力 ATTACH メッセージ・ヘッダーの推奨戻り基本リソース名 (RPRN) として使うよう、MFS に要求できます。リテラルは、8 文字を超えてはなりません。 dfldname も指定したときは、その dfldname を参照する MFLD に指定されているデータが、出力 ATTACH メッセージ・ヘッダーの RPRN として使用されます。そして、その dfldname を参照する出力メッセージ MFLD が存在しないときに、'literal' が使用されます。 dfldname を参照する MFLD 中のデータが 8 文字より大きいと、最初の 8 文字だけが使用されます。

#### **OFTAB=**

メッセージの出力データ・ストリームに出力フィールド・タブ分離文字を挿入するよう、MFS に指示します。 OPTIONS=DNM で OFTAB なら、OFTAB 文字が DD ヘッダーに挿入され、標識が MIX または ALL に設定されます。 OPTIONS=NODNM の場合、DD ヘッダーは送信されません。

#### **X'hh'**

16 進表現が "hh" となる文字を、出力フィールド・タブ分離文字に使用します。 X'3F' や X'40' の指定は無効です。

#### **C'c'**

文字 "c" を出力フィールド・タブ分離文字に使用します。 C" の指定は無効です。

制約事項: IMS アプリケーション・プログラムからのデータ・ストリーム中に、ここに指定した文字があってはなりません。もしあると、ブランク (X'40') に変更されます。

出力フィールド・タブ分離文字を定義するときは、MIX と ALL の一方も指定できます。デフォルト値は MIX です。

#### **MIX**

データをまったく含まなかったり、定義された DFLD 長に満たないデータしか含まなかったりするフィールドがあれば、そこに出力フィールド・タブ分離文字を挿入します。

#### **ALL**

データ長に関係なく、すべてのフィールドに出力フィールド・タブ分離文字を挿入します。

#### **COMPR=**

短フィールド、固定長フィールド、あるいはアプリケーション・プログラムから与えられるすべてのフィールドから末尾ブランクを取り除くよう、MFS に要求します。

DPM-AN 装置では、次の条件がすべて該当するとき、セグメントの最後から末尾ブランクが取り除かれます。

1. FILL=NULL または FILL=PT が指定されている。
2. マッピング中の現行セグメントに対して GRAPHIC=YES が指定されている。
3. MSG セグメントで OPT=1 または OPT=2 が指定されている。

条件 1、2、および 3 が満たされると、末尾空白が次のとおり置き換えられます。

**FIXED**

固定長フィールドの末尾空白をヌルで置き換えます。

**SHORT**

アプリケーション・プログラムによって短縮されたフィールドの末尾空白をヌルで置き換えます。

**ALL**

すべてのフィールドの末尾空白をヌルで置き換えます。

この後、レコードの最後ですべての末尾ヌルが圧縮されます。さらに詳しくは、FILL= オペランドの説明を参照してください。

DPM-BN 装置では、次の条件がすべて該当するとき末尾空白が取り除かれます。

1. 現行 DIV ステートメントに OFTAB が指定されているか、FILL=NULL または FILL=PT が指定されている。
2. マッピング中の現行セグメントに対して GRAPHIC=YES が指定されている。
3. MSG セグメントで OPT=1 または OPT=2 が指定されている。

条件 1、2、および 3 が満たされると、末尾空白が次のとおり除去されます。

**FIXED**

固定長フィールドから末尾空白を取り除きます。

**SHORT**

アプリケーション・プログラムによって短縮されたフィールドから末尾空白を取り除きます。

**ALL**

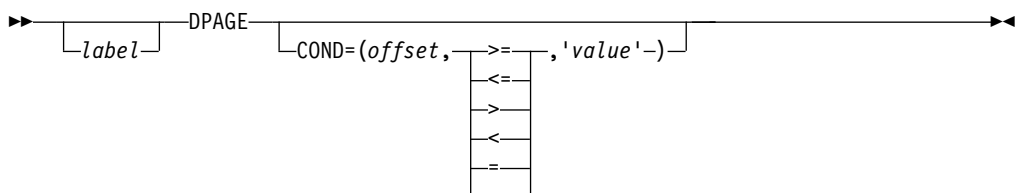
すべてのフィールドから末尾空白を取り除きます。

**DPAGE ステートメント**

DPAGE ステートメントは、装置形式の論理ページを定義します。

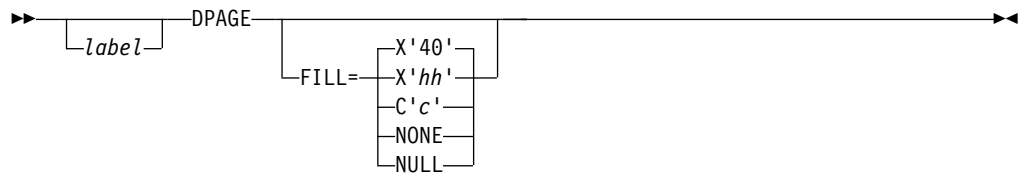
この装置形式 (FMT) を参照しているメッセージ記述子の中に、LPAGE ステートメントを含んでいるものがないか、特に装置オプションが必要とされないときは、このステートメントを省略できます。

**DEV TYPE=DPM-An、DPM-Bn のいずれかで、DIV TYPE=INPUT のときの形式**

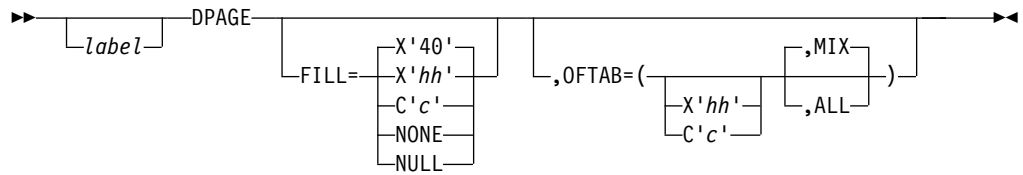




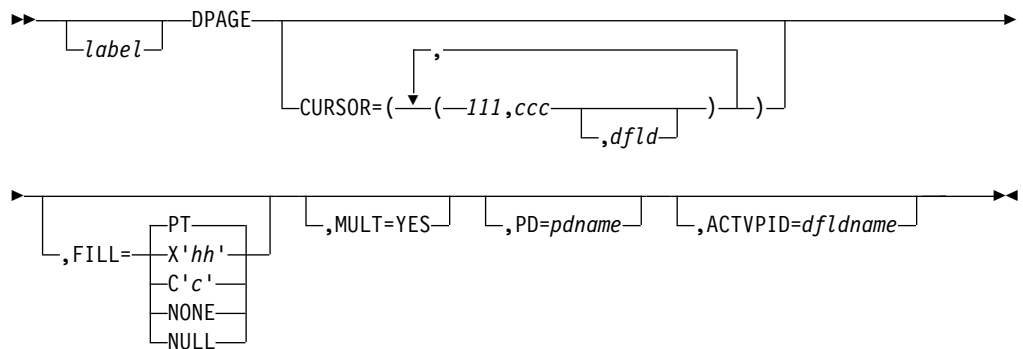
**DEV TYPE=DPM-An で、DIV TYPE=OUTPUT のときの形式**



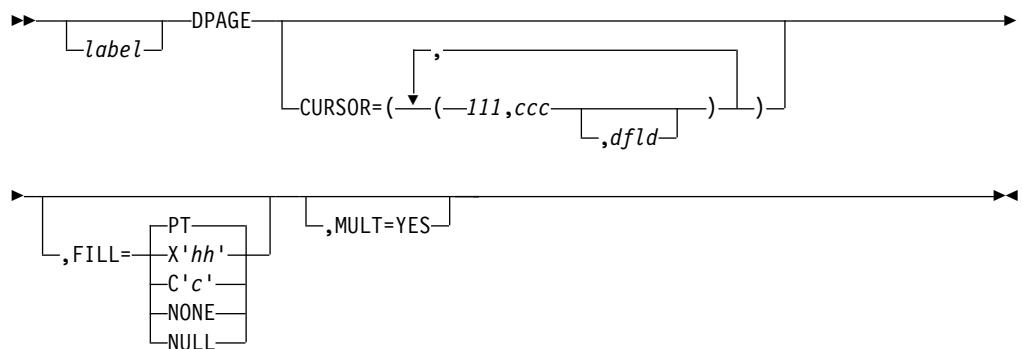
**DEV TYPE=DPM-Bn で、DIV TYPE=OUTPUT のときの形式**



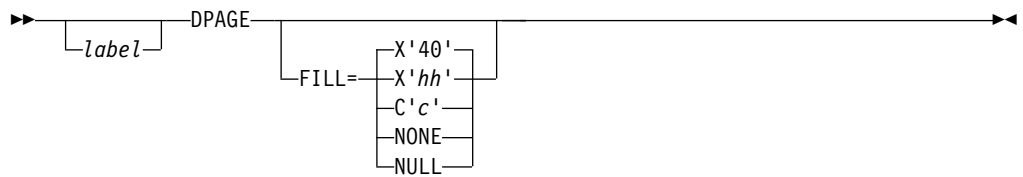
**DEV TYPE=3270-An のときの形式**



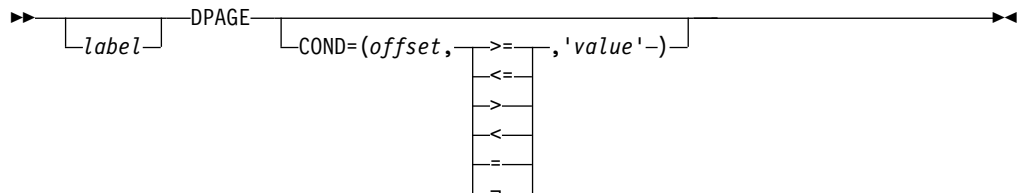
**DEV TYPE=3270 のときの形式**



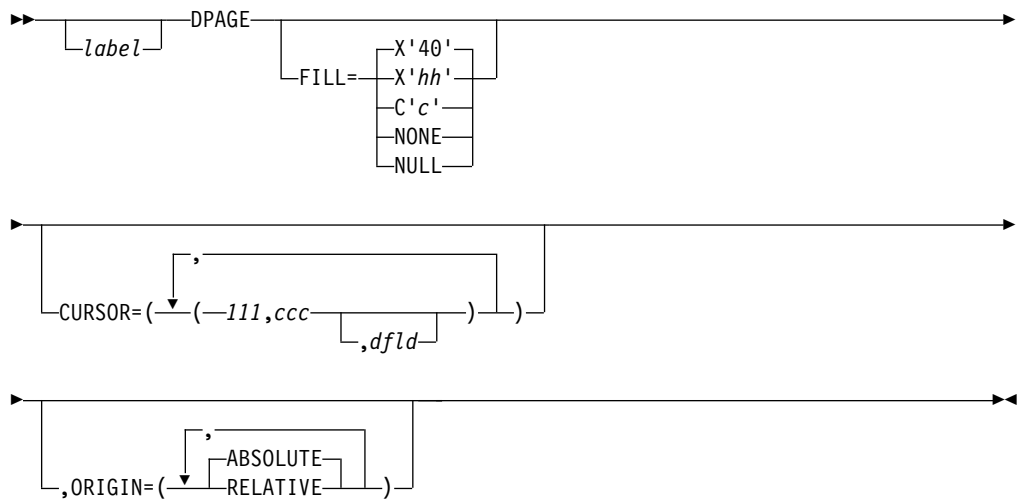
**DEV TYPE=3270P のときの形式**



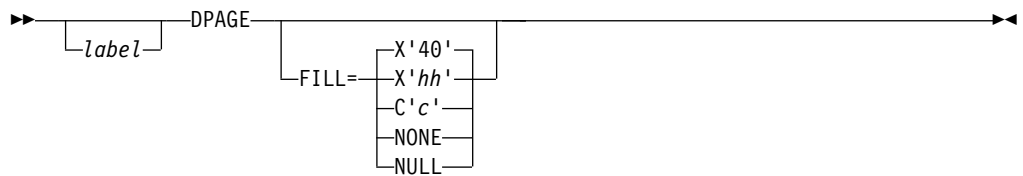
**DEV TYPE=FIN のときの形式**



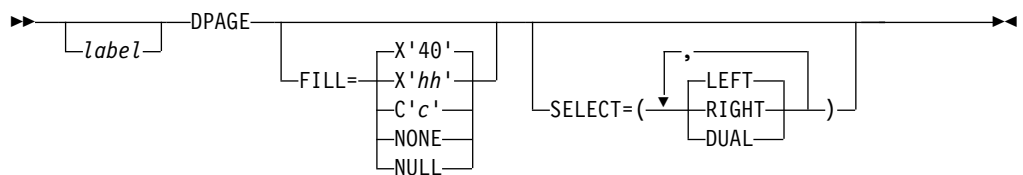
**DEV TYPE=FIDS、FIDS3、FIDS4、FIDS7 のときの形式**



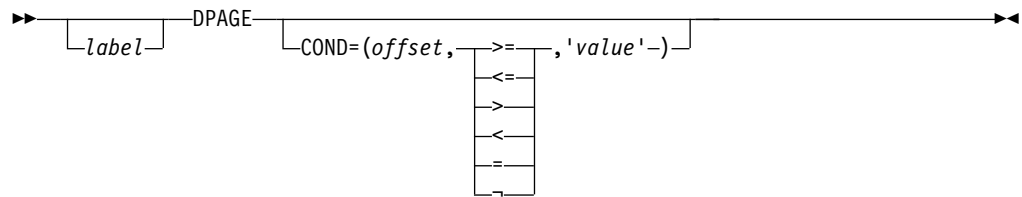
**DEV TYPE=FIJP または FIPB のときの形式**



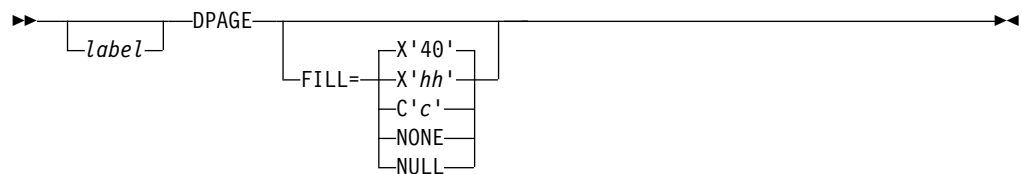
**DEV TYPE=FIFP のときの形式**



## DEV TYPE=SCS1 または SCS2 で、DIV TYPE=INPUT のときの形式



## DEV TYPE=SCS1 または SCS2 で、DIV TYPE=OUTPUT のときの形式



### パラメーター

#### label

この装置形式が LPAGE SOR= 参照を含んでいるとき、あるいはこの装置に対して 1 つの DPAGE ステートメントのみが定義されているとき、1 から 8 バイトの英数字名を指定できます。1 つの FMT 定義内で複数の DEV ステートメントが定義されているときは、そのそれぞれが同じラベルを持つ DPAGE ステートメントを含んでいなければなりません。

装置タイプが DPM-An で、DIV ステートメントで OPTIONS=DPAGE が指定されているときは、この名前がデータ名として、出力メッセージ・ヘッダーに挿入されてリモート・プログラムに送られます。ラベルの指定を省略すると、MFS は診断名を生成し、それをヘッダーに含めてリモート・プログラムに送信します。DPAGE ステートメントそのものを省略すると、FMT ステートメントのラベルが出力メッセージ・ヘッダーに挿入されて送信されます。OPTIONS=DNM なら、FMT ステートメントのラベルが DSN として DD ヘッダーに挿入されて送信されます。

#### COND=

最初の入力レコードに対して行う条件付きテストを指定します。指定する値は、ゼロを基準とするオフセット値です。オフセット指定では、入力レコードの LLZZ フィールドを忘れてはなりません (例えば、最初のデータ・バイトはオフセット 4 にあります)。条件が満たされると、この DPAGE 以後に定義されている DFLD が入力のフォーマット設定に使用されます。どの条件も満たされないときは、最後に定義されている DPAGE が使用されます。ただし、その最後に定義されている DPAGE に COND= が指定されている場合を除きます。最後の DPAGE に COND= パラメーターが指定されていて、しかもその条件が満たされないと、入力メッセージは拒否されます。メッセージ入力定義では、複数の LPAGE 定義が認められています。

このキーワードが指定され、DIV ステートメントに OPTIONS=NODNM が指定されていると、この指定が DPAGE 選択に使用されます。このキーワードが

指定され、DIV ステートメントに `OPTIONS=DNM` が指定されていると、`COND=` 指定は無視され、DD ヘッダーにあるデータ構造名が `DPAGE` 選択に使用されます。

`COND=` 比較の際、金融機関、SCS1、または SCS2 のキーボードから入力された小文字データが、大文字に変換されることはありません。したがって、リテラル・オペランドも小文字で指定しなければなりません。

#### **FILL=**

出力装置フィールドに使う充てん文字を指定します。デフォルト値は、3270 ディスプレイを除くすべての装置タイプで `X'40'`、3270 ディスプレイでは `PT` です。3270 出力に `EGCS` フィールドがあるときは、`FILL=PT` または `FILL=NULL` だけを指定してください。 `FILL=PT` では、出力フィールド (1 バイトまたは 2 バイト) にデータが送られる場合のみ、そのフィールドが消去されます。したがって、アプリケーション・プログラム・メッセージで `MFLD` が省略されていると、`DFLD` は消去されません。 `DPM-Bn` では、`OFTAB` が指定されていると、`FILL=` は無視され、`FILL=NULL` と見なされます。

#### **NONE**

装置フィールドを満たすのにメッセージ出力記述子にある充てん文字を使うときは、これを指定してください。

#### **X'hh'**

16 進表現が「hh」となる文字で装置フィールドを埋めます。

#### **C'c'**

文字 'c' で装置フィールドを埋めます。

#### **NULL**

フィールドを充てん文字で埋めないことを意味します。3270 ディスプレイ以外の装置では、装置フィールドがメッセージ・データでいっぱいにならないと、その行が「圧縮行」となります。

`DPM-An` 装置では、リモート・プログラムまたはサブシステムあてに送信するレコードに末尾ヌル (`X'3F'`) があると、それがすべて取り除かれます。取り除かれる末尾ヌルは、最初の非ヌル文字までで、非ヌル文字に挟まれているヌル文字はそのまま送信されます。レコード全体がヌルながら、その後ろにさらにデータ・レコードが続くときは、ヌルを 1 個だけ含んだレコードがリモート・プログラムに送信されます。レコード全体がヌルで、その後ろにさらにレコードが続いていて、しかも `OPTIONS=MSG` または `DPAGE` であるか、`PPAGE` で `OPTIONS=PPAGE` であれば、その `DPAGE` または `PPAGE` の末尾に至るまでのすべてのヌル・レコードが削除されます。

**PT** 3270 ディスプレイ以外では、`NULL` と同じです。3270 ディスプレイでは、装置フィールド (`DFLD`) をいっぱい満たさない出力フィールドがあると、その後ろにプログラム・タブ文字が挿入され、フィールドの既存のデータを消去します。それ以外は、`FILL=NULL` と同じです。

3270 ディスプレイ装置では、`X'3F'` より小さな値を指定しておく、制御文字は `X'00'`、その他の非図形文字は `X'40'` に変更されます。その他の装置では、

FILL=X'hh' または FILL=C'c' に X'3F' より小さな値が指定されると、その指定は無視され、デフォルトで X'3F' と見なされます (これは FILL=NULL 指定と同等です)。

#### **MULT=YES**

この DPAGE には複数の物理ページから成る入力メッセージが認められます。

#### **CURSOR=**

物理ページにおけるカーソルの位置を指定します。1 つの論理ページまたはメッセージが複数の物理ページから成っているときは、複数のカーソル位置が必要になることもあります。値 III は行番号、ccc は桁を指定します。III も ccc もともに 1 以上でなければなりません。カーソル位置は、定義済みのフィールド上にあるか、デフォルトで設定されなければなりません。III,ccc のデフォルト値は、3270 ディスプレイでは 1,2 です。金融機関ディスプレイ・コンポーネントでは、カーソル位置が特に指定されない限り、MFS はカーソルの位置付けを行いません。つまり、カーソルは、通常、装置上の出力データの最後に置かれず。金融機関ディスプレイ・コンポーネントでは、ORIGIN= パラメーターがどう指定されていても、すべてのカーソル位置が絶対位置になります。

dffd パラメーターは、入力時にはアプリケーション・プログラムにカーソル情報を与えるための手段、出力時にはアプリケーション・プログラムがカーソル位置を指定するための手段として使われます。

ヒント: 出力時のカーソルの位置付けには、カーソル属性機能 (MFLD ステートメントで ATTR=YES を指定) を使用してください。

dffd パラメーターには、カーソル位置を含んでいるフィールドの名前を指定します。この名前は MFLD ステートメントで参照することができ、この DEV 定義にある DFLD ステートメントのラベルとして使用してはなりません。このフィールドの形式は 2 個の 2 進ハーフワードから成り、一方に行番号、他方に桁番号が入ります。メッセージ入力記述子がこのフィールドを参照するときは、そこにはメッセージ項目のカーソル位置が含まれます。メッセージ出力記述子がこのフィールドを参照するときは、アプリケーション・プログラムがそこに行番号と桁番号を含む 2 個の 2 進ハーフワードとしてカーソル位置を指定します。指し示されたフィールドに 2 進ゼロが入っていると、指定された III,ccc が出力時のカーソル位置として使われます。入力時にこのフィールドに 2 進ゼロが入っていると、それはカーソル位置が定義されていないことを意味します。この dffd を参照する入力 MFLD は、GRAPHIC=NO 指定のあるセグメント内に定義するか、EXIT=(0,2) として、2 進数を 10 進数に変換しなければなりません。

#### **ORIGIN=**

定義された各物理ページについて、金融機関ディスプレイにおけるページの位置付け方法を指定します。デフォルト値は ABSOLUTE です。

#### **ABSOLUTE**

旧画面を消去し、1 行目の 1 桁目にページを位置付けます。DFLD ステートメントに指定された行と桁が、画面上のデータの実際の行と桁になります。

#### **RELATIVE**

出力時にカーソルが位置付けられていた行の次の行の 1 桁目を、ページ起

点とします。装置へのあらゆる出力に一貫性を持たせるように計画しておかないと、好ましくない結果になることがあります。

#### **OFTAB=**

記述する DPAGE の出力データ・ストリームに、DPAGE ステートメントで指定された出力フィールド・タブ分離文字を挿入するよう、MFS に指示します。

##### **X'hh'**

16 進表現が「hh」となる文字を、出力フィールド・タブ分離文字に使用します。X'3F' や X'40' の指定は無効です。

##### **C'c'**

文字 'c' を出力フィールド・タブ分離文字に使用します。C' ' の指定は無効です。

制約事項: IMS アプリケーション・プログラムからのデータ・ストリーム中に、ここに指定した文字があってはなりません。もしあると、ブランク (X'40') に変更されます。

出力フィールド・タブ分離文字を定義するときは、MIX と ALL の一方も指定できます。デフォルト値は MIX です。

##### **MIX**

データをまったく含まなかったり、定義された DFLD 長に満たないデータしか含まなかったりするフィールドがあれば、そこに出力フィールド・タブ分離文字を挿入します。

##### **ALL**

データ長に関係なく、すべてのフィールドに出力フィールド・タブ分離文字を挿入します。

#### **SELECT=**

先行 DEV ステートメントで FEAT=DUAL を指定された FIFP 装置に対して、キャリッジ選択を行います。正しい用紙が取り付けられ、左マージンが適切に設定されていることを確認しておいてください。デフォルト値は LEFT です。

##### **LEFT**

この DPAGE で定義された対応物理ページを左プラテンへ送ります。

##### **RIGHT**

この DPAGE で定義された対応物理ページを右プラテンへ送ります。

##### **DUAL**

この DPAGE で定義された対応物理ページを左右両方のプラテンへ送ります。

#### **PD=**

(区画形式モードの 3180 と 3290 では) この DPAGE ステートメントと関連付けた区画の区画記述子名を指定します。このパラメーターにより、メッセージの論理ページと該当する区画の間でマッピングが行われます。PD の名前は、DEV ステートメントで指定された PDB ステートメントに含まれていなければなりません。

#### **ACTVPID=**

(区画形式モードの 3290 では) 活動化する区画の区画 ID 番号 (PID) を含んで

いる、メッセージ中の出力フィールドの名前を指定します。この *dfldname* は MFLD ステートメントで参照しなければならず、DEV 定義内の DFLD ステートメントのラベルとして使用しないでください。アプリケーション・プログラムは、活動化する区画の PID をこのフィールドに格納します。PID の形式は 2 バイトの 2 進数で、その値は X'0000' から X'000F' でなければなりません。

3180 では、このオペランドを指定しないでください。この装置では 1 つの区画のみ認められるので、アクティブ区画を指定する必要はありません。

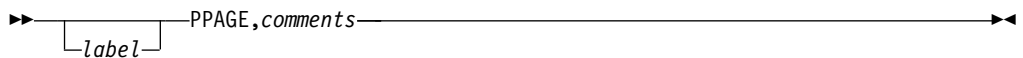
## PPAGE ステートメント

PPAGE ステートメントは、DPM-An または DPM-Bn の装置タイプでのみ使用でき、表示ページの先頭を定義します。

表示ページとは、この定義の DIV ステートメントで OPTIONS=PPAGE が指定されているとき、ページング要求に応じてリモート・プログラムに引き渡されるデータの単位を言います。DPM-Bn MODE=RECORD の場合にのみ、OPTIONS=MSG または DPAGE が指定されていると、DIV ステートメントのそれらのオプションで説明されているとおりにページングが行われ、次いで、PPAGE ステートメントが新しいレコードの始まりを定義します (つまり、RCD ステートメントと同等です)。

入力 DPAGE では、PPAGE ステートメントが 1 つしか認められず、DPAGE ステートメントと最初の DFLD ステートメントの間に置かれます。出力 DPAGE では、OPTIONS=PPAGE と定義されているメッセージの DPAGE に 2 つの PPAGE ステートメントが連続していると、PPAGE ラベルをデータ名とする出力メッセージ・ヘッダーだけがリモート・プログラムに送られます。ただし、DPM-Bn で OPTIONS=(PPAGE,DNM) の場合を除きます。DPM-Bn では、DIV TYPE=OUTPUT に対して OPTIONS=(PPAGE, NODNM) が指定されている場合、DFLD ステートメントなしの PPAGE ステートメントは認められません。警告メッセージが出され、その PPAGE ステートメントは無視されます。OPTIONS=MSG または DPAGE では、連続した PPAGE ステートメントは無視されます。

## フォーマット



## パラメーター

### *label*

1 から 8 文字の英数字名を指定してください。OPTIONS=PPAGE なら、このラベルがデータ名 (DPM-An) またはデータ構造名 (DPM-Bn) としてメッセージ出力ヘッダーまたは DD ヘッダーに挿入されて送信され、リモート・プログラムに対してこの表示ページのデータ構造を伝えます。ラベルの指定を省略すると、MFS が診断ラベルを生成し、それをヘッダーに含めてリモート・プログラムに送信します。

ヒント: ユーザー定義ラベルを指定してください。MFS が生成する名前は、MFS 定義の再コンパイルで変化することがあるからです。

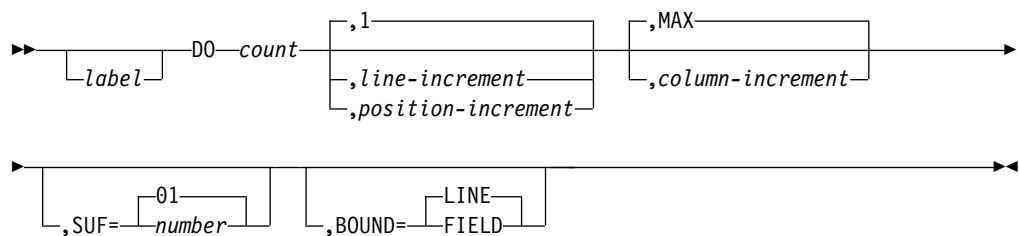
指定するラベルは、少なくとも当該 FMT 定義の内部で固有である必要があります。また、この表示ページに含まれているデータのフォーマット設定に使う DSECT を、リモート・プログラムがこのラベルに基づいて指定するのであれば、IMS システム内部で固有であることが望まれます。

## DO ステートメント

DO ステートメントは ENDDO ステートメントと対で用いられ、このステートメント間で DFLD ステートメントと RCD ステートメントの反復生成が行われます。

DO を使用するときは、DFLD の命名に制約があります。

### フォーマット



### パラメーター

#### label

1 から 8 文字の英数字名を指定できます。使用されません。

#### count

ステートメントの生成回数を指定します。

#### line-increment

最初のサイクル以後、行位置をどれだけずつ大きくするか指定します。最初のサイクルでは、DFLD ステートメントの POS= キーワードに指定された *lll* 値が用いられます。デフォルトは 1 です。このパラメーターは、DEV TYPE が DPM-An または DPM-Bn の場合は指定しません。

#### position-increment

最初のサイクル以後、位置パラメーターをどれだけずつ大きくするか指定します。最初のサイクルでは、DFLD ステートメントの POS= オペランドに指定された *nnn* 値が使用されます。位置増分は、MODE=STREAM が指定されているとき、入力装置形式で使用されます。このパラメーターは、DEV TYPE が DPM-An または DPM-Bn の場合は指定されません。

#### MAX

1 つのサイクルが終わるごとに行増分を加えますが、DFLD の桁値はどのサイクルでも一定に保ちます。装置形式で MODE=STREAM が指定されているとき、または、装置タイプが DPM-An または DPM-Bn であるときは、このパラメーターは使用されません。指定しても、無視されます。

#### column-increment

最初のサイクル以後、桁位置をどれだけずつ大きくするか指定します。最初のサイクルでは、DFLD ステートメントの POS= キーワードに指定された *ccc* 値が



用いられます。デフォルトは MAX です。装置タイプが DPM-An または DPM-Bn のとき、あるいは装置形式で MODE=STREAM が指定されているときは、このパラメーターは使用されず、指定しても無視されます。

#### SUF=

2 桁の接尾部を指定します。生成される最初の DFLD ステートメント・グループの *dflename* にこの接尾部が付加されます。デフォルトは 01 です。MFS は、後続のステートメント・グループが生成されるごとに、接尾部を 1 ずつ大きくしていきます。

指定された接尾部が 2 桁を越えていると、MFS は右端の 2 桁だけを使用します。

生成される接尾部が最終的に 2 桁を超えるような count が指定されると、MFS は許可される最大値まで count を小さくします。例えば、count が 8 で、SUF=95 だと、発生する接尾部のうち 100、101、102 は無効です。この場合、MFS は count を 5 に減らしてステートメントを処理し、エラー・メッセージを出します。

#### BOUND=

行位置と桁位置をいつ更新するか指定します。デフォルトは LINE です。装置形式で MODE=STREAM が指定されているとき、または、装置タイプが DPM-An または DPM-Bn であるときは、このパラメーターは使用されません。指定しても、無視されます。

#### LINE

反復の前にすべてのフィールドを調べます。桁増分のために、DFLD ステートメント・グループ中のいずれかのフィールドが行に収まらない事態が生じると、すべてのフィールドの桁位置値が初期値にリセットされ、行位置値が line-increment 値だけ大きくなります。

#### FIELD

ステートメントが繰り返されるたび、桁位置値を column-increment 値だけ大きくします。MAX が指定されているか、新しい桁位置値が装置の最大行長に達すると、行位置値が line-increment 値だけ大きくなり、桁位置値が初期値にリセットされます。

### 行と桁の増分の例

次の例は、行と桁を増やす方法を示しています。

```
D0      20,1,38
A1  DFLD  POS=(9,6),LTH=6
B1  DFLD  POS=(9,27),LTH=3
```

この例で、A1 と B1 は、行増分 (1) と桁増分 (38) ずつ増加します。生成は、コンパイラーによって次のように行われます。

- セット内の各桁値に桁増分を加算します。位置は (9,44) および (9,65) になります。
- これらの新しい桁値を使用するフィールドの中に、この装置の行サイズ制限を超えるものがあるかどうかテストします。この例では、3270 型式 2 で制限を 80 としています。
- 行幅の違反はないので、新しい桁値および同じ行値を使用して A2 と B2 を生成します。

- 桁増分をもう一度加算します。位置は (9,82) および (9,103) になります。
- フィールドが行幅を超えるので、桁値が元の値にリセットされて (9,6) および (9,27) になり、行増分が適用されます。その結果、位置は (10,6) および (10,27) になります。
- 新しい行値と、元のステートメントにある桁値を使用して、A3 と B3 を生成します。

反復回数が 20 回になるまで、生成がこのようにして継続されます。

### 生成された DFLD ステートメントの印刷

EXEC ステートメントのパラメーター・リストに COMP を指定しておくこと、生成された DFLD ステートメントをシンボリック・ソース形式で印刷できます。こうすることで、中間テキスト・ブロックを解釈することなく、DFLD ステートメント生成の結果を見ることができます。

生成された DFLD ステートメントごとに、次の各項目が印刷されます。

- 生成されたステートメント・シーケンス番号と、それに続く正符号 (+)。正符号は、その DFLD ステートメントが DO ステートメント処理の結果として生成されたものであることを表します。
- DFLD ステートメント・ラベル (あれば)。付加された接尾部を含みます。
- ステートメント演算子である DFLD。
- EGCS リテラルでは、G、SO、SI がありません。すべての指定を印刷できるだけのスペースがないと、リテラルが切り捨てられます。切り捨てがあったことは、リテラルの一部と 3 個のピリオド (...) で示されます。このピリオドが、切り捨てられた部分を表します。
- ATTR=(YES,*nn*) (あれば)。
- ATTR=YES (あれば)。
- ATTR=*nn* (あれば)。
- ATTR=(...) (属性があれば)。
- EATTR=(...) (あれば)。
- RECORD または STREAM 形式の POS= キーワード。行と桁、またはストリーム位置が、それぞれの増分で更新されます。装置タイプが DPM-An または DPM-Bn のときは印刷されません。
- SCA (あれば)。
- フィールド長。形式は LTH=*nnnn*。

これ以外のオペランドは、たとえソース DFLD ステートメントに指定されていても印刷されません。

装置タイプ DPM-An または DPM-Bn では、RCD ステートメントが DO ステートメントと ENDDO ステートメントの間に来ることがあります。その場合は、その RCD ステートメントに続く DFLD の反復生成ごとに、新しいレコード境界が作成されます。例えば、次の例だと、DFLD の A01、B01、C01 がレコード 1、A02、B02、C02 がレコード 2、A03、B03、C03 がレコード 3 に入ります。

```

DO 3
RCD
A  DFLD   LTH=10
B  DFLD   LTH=10
C  DFLD   LTH=10
ENDDO

```

あるいは、RCD ステートメントが DO ステートメントの直前に置かれることもあります。その場合は、DO ステートメント後の最初の DFLD ステートメントから新しいレコード境界が始まり、ENDDO ステートメント (または最大レコード長) に達するまで終わりません。例えば、次の例だと、DFLD D01 から新しいレコードが始まり、E01、D02、E02 がすべてそこに含まれます。

```

RCD
DO 2
D  DFLD   LTH=10
E  DFLD   LTH=10
ENDDO

```

## RCD ステートメント

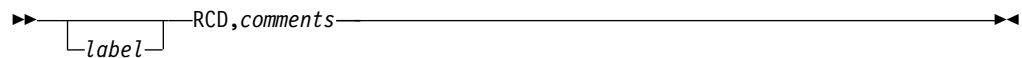
RCD は、レコードにおける DFLD の配置を左右するステートメントであり、DEV TYPE=DPM-An または DPM-Bn でのみ有効です。

DFLD ステートメントの前に置かれ、リモート・プログラムへ送る新しい伝送レコードを開始します。RCD ステートメントの後ろに続く DFLD 群は、次の RCD ステートメントが現れるか、最大レコード長に達するまで (あるいは、NOSPAN が指定されていれば、フィールド全体を現行レコードに収められなくなるまで)、1 つの伝送レコードに挿入されます。

RCD ステートメントは、PPAGE、DO、DFLD、ENDDO の各ステートメントの後ろに配置することもできます。RCD ステートメントを 2 つ続けると、最初の RCD ステートメントだけが有効と見なされます。

RCD ステートメントは、STREAM モードでは使用できません。

## フォーマット



## パラメーター

### *label*

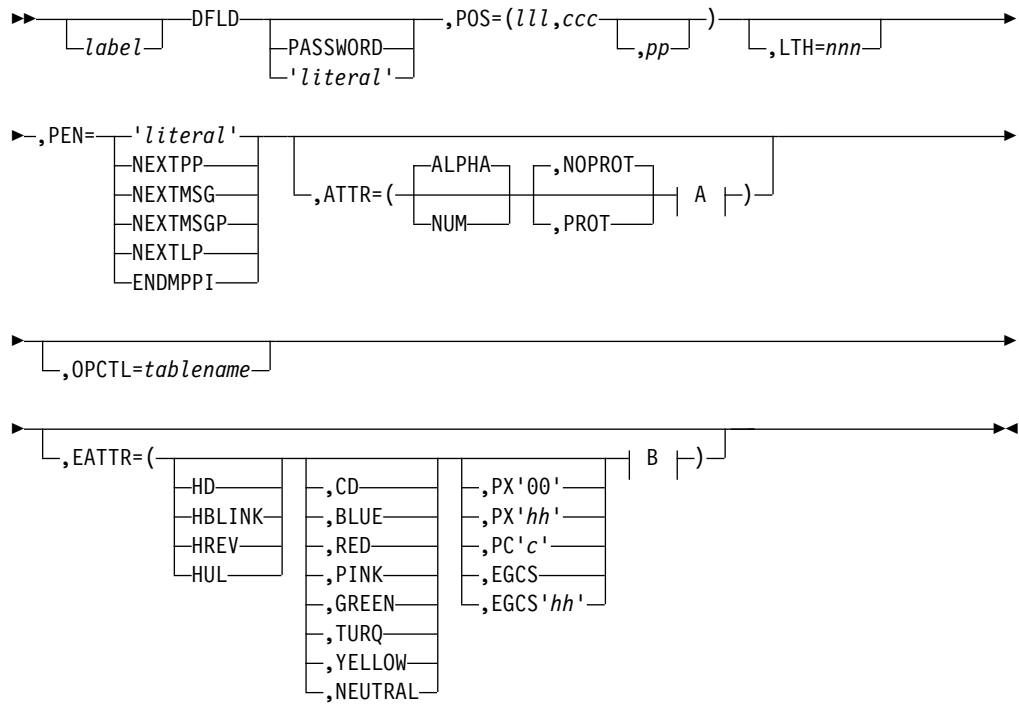
1 から 8 文字の英数字名を指定できます。使用されません。

## DFLD ステートメント

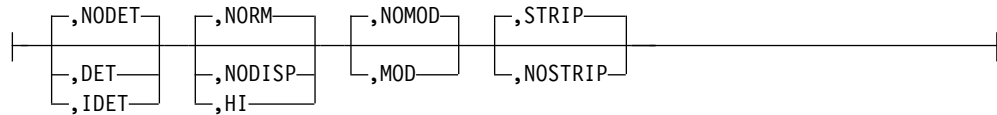
DFLD ステートメントは、端末やリモート・プログラムから読み取られ、あるいは書き込まれるフィールドを装置形式内に定義します。

IMS またはリモート・アプリケーション・プログラムにとって関係のある部分だけを定義してください。形式内のヌル・スペースは特に定義する必要がありません。

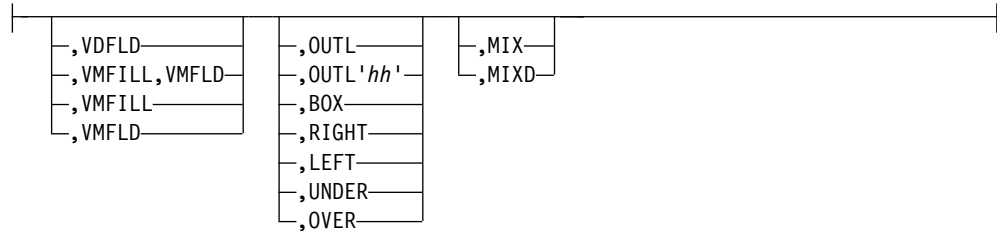
## DEV TYPE=3270 または 3270-An のときの形式



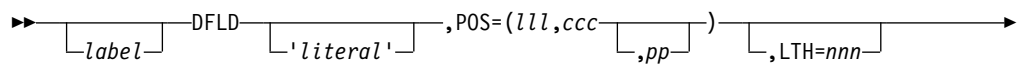
### A:

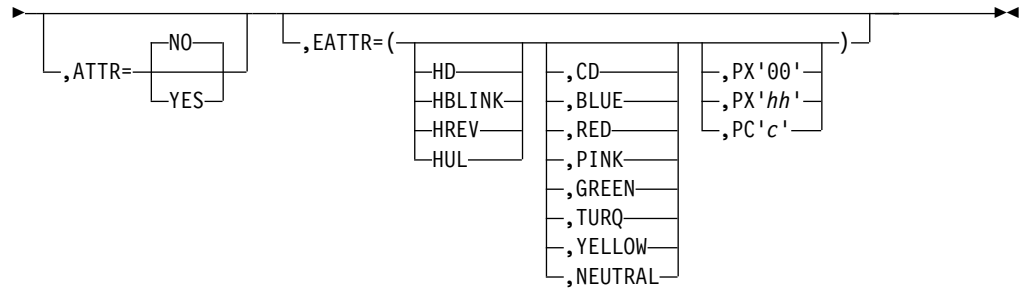


### B:

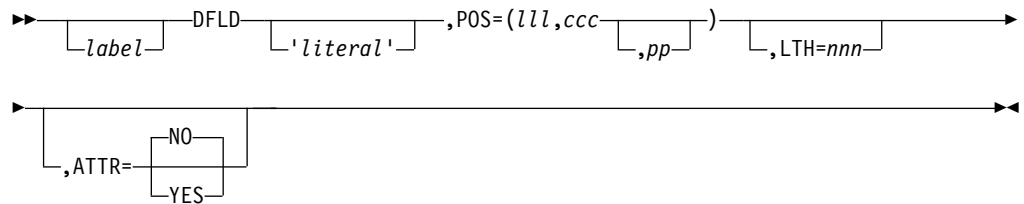


## DEV TYPE=3270P のときの形式

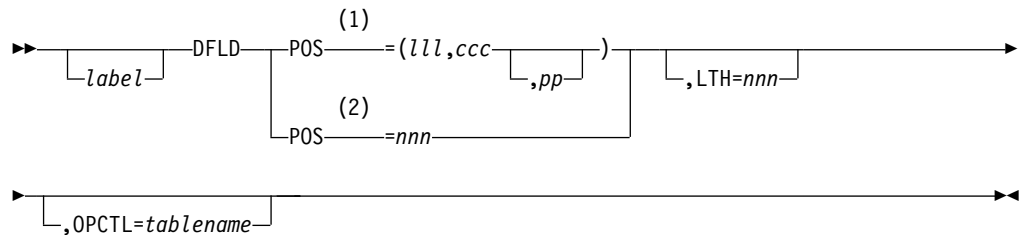




**DEV TYPE=FIDS、FIDS3、FIDS4、FIDS7、FIFP、FIJP、FIPB のときの形式**



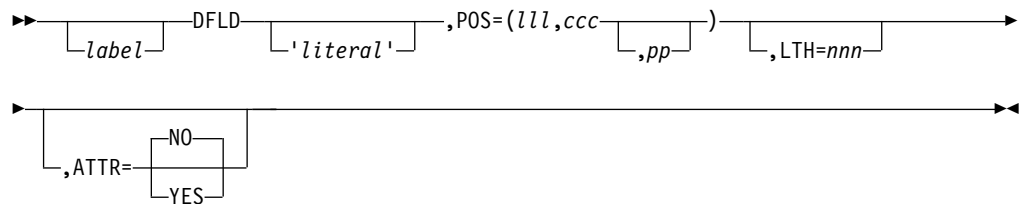
**DEV TYPE=FIN のときの形式**



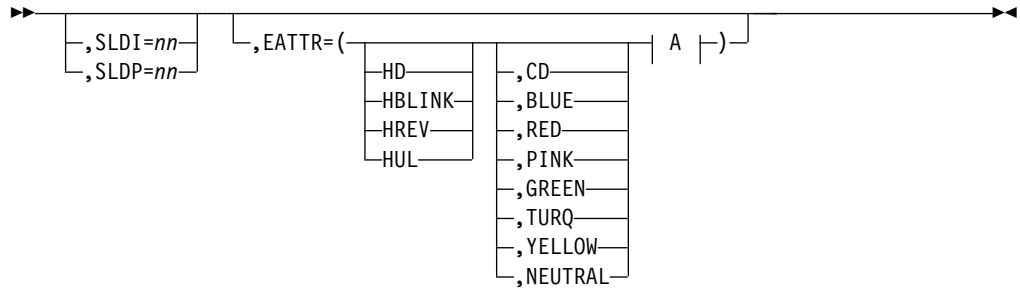
注:

- 1 MODE=RECORD のみ
- 2 MODE=STREAM のみ

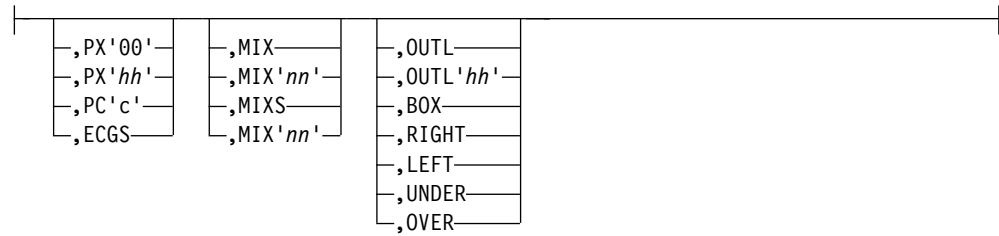
**DEV TYPE=SCS1 または SCS2 で、DIV TYPE=OUTPUT のときの形式**



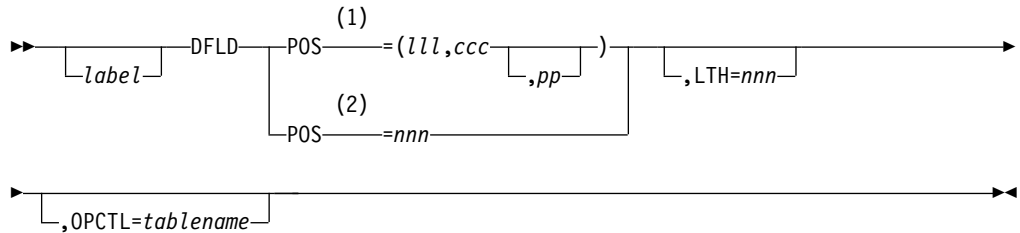
**SCS1 のみのときの形式**



**A:**



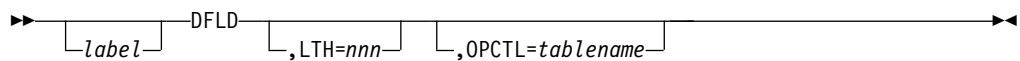
**DEV TYPE=SCS1 または SCS2 で、DIV TYPE=INPUT のときの形式**



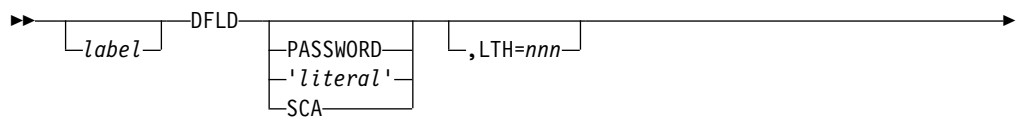
注:

- 1 MODE=RECORD のみ
- 2 MODE=STREAM のみ

**DEV TYPE=DPM-An または DPM-Bn で、DIV TYPE=INPUT のときの形式**



**DEV TYPE=DPM-An または DPM-Bn で、DIV TYPE=OUTPUT のときの形式**





## パラメーター

### **label**

1 から 8 文字の英数字名を指定できます。メッセージ記述子は、端末またはリモート・プログラムとのデータ転送でこのラベル (dfldname) を参照します。MFS の反復生成機能 (DO ステートメントと ENDDO ステートメント) を使用するときは、この dfldname を最大 6 文字に制限してください。ステートメントが反復生成されるごとに、2 桁のシーケンス番号 (01 から 99) がラベルに付加されます。ここに指定されたラベルが 6 文字を超えているときに反復生成が使用されると、そのラベルが 6 文字で切り捨てられ、2 桁のシーケンス番号が付加されて、8 文字の名前を形成します。この場合、エラー・メッセージは出ません。

PASSWORD、SCA、または '*literal*' が指定されていると、ラベルの指定は無効であり、エラー・メッセージが出ます。DIV ステートメントに DPN、PRN、RDPN、または RPRN の dfldname 指定があると、dfldname を現行 DIV ステートメントの DFLD ラベルとして使うことはできません。

### **PASSWORD**

このフィールドを、入力メッセージ用 IMS パスワードの位置情報として指定します。

IMS では、大/小文字混合または大文字のパスワードをサポートしています。

ヒント: 入力メッセージ定義の PASSWORD 機能を使用してください。

PASSWORD を指定すると、この DFLD ステートメントで記述されているフィールドをメッセージ記述子で参照できなくなります。また、PASSWORD を指定するときは、label 指定を省略しなければなりません。

### **'literal'**

装置に示すリテラル文字ストリングを指定します。literal の長さは、3270 ディスプレイ装置では 256 バイトを超えてはなりません。また、FIDS と FIDS3 では 40 バイト、FIDS4 では 64 バイト、FID57 では 80 バイト、3270P では 256 バイト、すべてのプリンター装置とパンチ装置では行幅を、いずれも超えてはなりません。DPM では、literal の長さが RCDCTL オペランドに指定された値を超えてはなりません。

3270 ディスプレイでは、リテラル・フィールドが PROT 属性を (指定の有無にかかわらず) 持っています。ALPHA が指定されないと、NUM と見なされません。

制約事項: literal を指定すると、この DFLD ステートメントで記述されるフィールドをメッセージ記述子で参照できなくなります。また、literal を指定するときは、label 指定を省略しなければなりません。

## SCA

DPM 定義でのみ使用されるパラメーターです。IMS アプリケーション・プログラムから送られてくるか、DSCA に指定された SCA 情報を、この DFLD で送信します。

SCA を指定するときは、*label* を指定しないでください。

## POS=

このフィールドの最初のデータ位置をディスプレイ・フォーマットの行 (III)、桁 (ccc)、物理ページ (pp) で定義します。pp の指定がないと、1 と見なされます。

DEV TYPE=FIN、FIDS、FIDS3、FIDS4、FIDS7、FIJP、FIPB、FIFP、SCS1、SCS2 の場合

### 111,ccc

このフィールドの、レコード内におけるレコード番号と位置を指定します。MODE=RECORD なら、この指定形式を使用しなければなりません。III と ccc は、ともに 1 以上でなければなりません。

### nnn

STREAM モードの入力で、このフィールドの開始位置を指定します。特に指定しないと、このフィールドは先行フィールドの直後から始まります。これが最初のフィールドであれば、左マージンから始まります。MODE=STREAM が指定され、POS= が指定されているときは、この指定形式が必要です。nnn は 1 以上でなければなりません。

### 111,ccc,pp

出力フィールドに対して、行番号と桁番号、さらにオプションで物理ページ番号を指定します。III、ccc、pp は、いずれも 1 以上でなければなりません。

DEV TYPE=3270、3270-An、3270P の場合

### 111,ccc,pp

出力フィールドに対して、行番号と桁番号、さらにオプションで物理ページ番号を指定します。III、ccc、pp は、いずれも 1 以上でなければなりません。

3270 ディスプレイでは、POS=(1,1) を指定しないでください。末尾から先頭へ折り返すようにフィールドを定義しないでください。

制約事項: 3270 のいくつかの型式では、フィールドが 1 行目の 2 桁目から始まり、しかも英字属性と保護属性を共に持っているとき、ディスプレイ画面をコピーできないことがあります。

DEV TYPE=DPM-An または DPM-Bn の場合

### DPM 装置の場合

POS= キーワードが無視されます。

## LTH=

フィールドの長さを指定します。定位置パラメーターに '*literal*' が指定されているときは、このオペランドの指定を省略してください。その場合、*literal* の長さがフィールド長として使われます。このオペランドと '*literal*' を共に指定し



て、両者の長さが違っていると、出力フォーマット設定が予測できないものになるおそれがあります。LTH= に指定する値は、装置の物理ページ・サイズを超えてはなりません。

LTH= に指定できる最大長は、すべての装置で 8000 文字です。ただし、3270 ディスプレイ、3604 ディスプレイ、RCDCT=NOSPAN の DPM を除きます。3270 ディスプレイの最大長は、画面サイズ - 1 です。例えば、480 文字ディスプレイなら、最大長は 479 文字となります。FIDS ディスプレイ・コンポーネントでは、最大長は 240 文字です。また、FIDS3 では 480 文字、FIDS4 では 1024 文字、FIDS7 では 1920 文字となります。長さとして 0 を指定しないでください。DPM で RCDCT=NOSPAN が指定されていて、RCDCTL が 8000 未満の場合、長さは RCDCTL 値以下でなければなりません。SCA と LTH= をともに指定するときは、LTH を 2 としなければなりません。

POS= と LTH= には、3270 ディスプレイ装置や ATTR=YES 指定のある DFLD 用に予約されている属性文字位置は含まれません。このバイトは、アプリケーション・プログラムからはアクセスできませんが、画面/印刷ページでは各表示/印刷フィールドの前の位置を占めますから、ディスプレイ/プリンター・フォーマットの設計では、このバイトを含めておくことを忘れてはなりません。

3270 プリンター用の DFLD 定義では、ハードウェア属性文字が使用されません。したがって、ATTR=YES が指定されていない限り、フィールドは属性文字が入らないように並置して定義する必要があります。しかし、3270P と定義されているプリンターでは、印刷行の最後の桁 (FEAT=、WIDTH=、装置のデフォルト幅のいずれかによります) は使用できません。行の最後の桁は、IMS による紙送り制御のために予約されています。このため、印刷行が 120 (FEAT=120) で、DFLD に POS=(1,1),LTH=120 が指定されていると、最初の行に 119 文字が印刷され、2 行目に残りの 1 文字が印刷されます。

DPM 定義では、DIV ステートメントに OPTIONS=NOSIM2 が指定され、ATTR=YES か YES,*mn* が指定されていると、DFLD の長さに 2 バイト + 拡張属性が追加されます。最初の 2 バイトは、2 進表現の 3270 属性 (保護、数字、など) 用に予約されています。OPTIONS=SIM が指定されていれば、ATTR=YES または YES,*nn* 指定を持つ DFLD の長さに 1 バイト (または、1 バイト + 拡張属性) が追加されます。つまり、フィールドの最初の 1 バイトは、シミュレートされた属性用に予約されています。

検出可能フィールド (DET または IDET) には、1 バイトの検出指定文字と 3 バイトの埋め込み文字用に、POS と LTH に 4 個の文字位置を用意しておかなければなりません。ただし、その検出可能フィールドが表示行上の最後のフィールドであるときは、検出指定文字用に 1 個の文字位置を用意するだけで構いません。検出指定文字はフィールド・データの前、埋め込み文字 (必要なら) はフィールド・データの後ろに置かれます。検出指定文字と必要な埋め込み文字は、アプリケーション・プログラムが用意するか、フィールド・データと共に MFLD リテラルとして与えなければなりません。埋め込み文字は、装置上の先行フィールドでも必要とされることがあります。

#### **ATTR=**

下に示す DEV TYPE と DIV TYPE の組み合わせについて表示属性を定義します。

- DEV TYPE=3270 または 3270-An

属性キーワードは、必要なものだけを任意の順序で指定できます。下線のあるキーワードはデフォルトなので、指定を省略できます。

2 つのユーザー定義フィールドが 2 つ以上の文字で分離されていると、MFS はディスプレイ・バッファ中でのそのスペースを表現するために、未定義のフィールドを 1 つ生成します。この未定義フィールドの表示属性は、NUM、PROT、NODISP となります。

#### **ALPHA | NUM**

フィールドに数値属性を与えるかどうかを指定します。数値属性は、3275/3277 または 3276/3278 が数値ロック機能 (データ入力キーボードの自動上段シフト) を使用することを指定します。フィールドに NUM と PROT を指定すると、自動スキップ機能が使用されます。つまり、無保護フィールドの最終文字位置に文字が入力されると、カーソルが自動的にその (NUM 属性と PROT 属性を指定された) フィールドをスキップし、次の無保護フィールドの最初の文字位置に移動します。埋め込まれた無保護フィールドの後ろに上記の ATTR= パラメーターで説明した未定義フィールドがある場合は、自動スキップ機能が使用されます。このパラメーターを PROT パラメーターと併用すると、COPY 機能をロックできます。詳しくは「PROT」を参照してください。

#### **NOPROT | PROT**

フィールドがユーザーによる変更から保護されているかどうか指定します。リテラル・フィールドには常に PROT が適用され、NOPROT を指定しても無視されます。

リモート 3270 端末の IMS コピー機能は、ディスプレイの 1 行目 1 桁目にある属性バイトの値を保護と英字に設定することでロックできます。コピー機能がロックされると、この機能を使ってディスプレイの内容をプリンターにコピーすることができなくなります。3274 および 3276 制御装置の「ローカル・コピー機能」は、この属性設定ではロックできません。「ローカル・コピー機能」は、プリント・キーにより呼び出されます。

#### **NODET|DET|IDET**

ライト・ペン操作で検出できるフィールドかどうかを指定します。DET は据え置き検出可能フィールド、IDET は即時検出可能フィールドを表します。LTH= オペランドのところで説明したとおり、適切な指定文字と埋め込み文字を用意してください。3270 ディスプレイ装置では、1 つの行における最後の検出可能フィールドの前に、いくつかの検出可能フィールドを置けるのか、またはいくつかの検出可能フィールドと検出不能フィールドを混在させられるのか、が制約されています。

#### **NORM|NODISP|HI**

フィールドの表示輝度を、通常 (NORMAL)、高輝度 (HI)、表示不可 (NODISP) のいずれかに指定します。NODISP を指定するときは、DET または IDET を指定できません。

高輝度 (HI) フィールドを定義するとき、最初のデータ・バイトに検出指定文字を入れておくと、その高輝度 (HI) フィールドが検出可能になります。

#### **NOMOD|MOD**

このフィールドにフィールド変更属性バイトがあると見なすかどうかを指定します。MOD では、たとえフィールドが変更されていなくても、端末はそのフィールドがユーザーによって変更されたものと見なします (つまり、フィールド変更属性バイトに変更データ・タグ (MDT) が設定されます)。これを PROT 属性と混同しないでください。PROT は、ユーザーによる変更を禁止する属性です。リテラル・フィールドでは MOD が無視されます。

MOD が指定されると、MFS からこの物理ページに出力が送られてくるたびに、変更属性が設定されます (ただし、動的属性変更によって指定変更される場合を除きます)。

#### **STRIP|NOSTRIP**

入力フィールドをアプリケーション・プログラムに引き渡す前に、その前にあるペン検出指定バイトを取り除く (STRIP) かどうか指定します。ライト・ペン検出可能フィールドに EGCS 属性が指定されているときは、DFLD ステートメントで ATTR=NOSTRIP を指定し、さらに入力データから 2 個の指定文字を取り去る、またはそれをバイパスするよう、アプリケーション・プログラムを設計してください。ATTR=STRIP を指定 (もしくは、デフォルトで使用) すると、MFS は最初の指定文字だけを取り除きます。フィールド中の最後の文字が失われる (切り捨てられる) ことがあります。

- DIV TYPE=OUTPUT で、DEV TYPE=3270P、FIDS、FIDS3、FIDS4、FIDS7、FIFP、FIJP、FIPB、FIS1、SCS2

出力メッセージにこのフィールドの属性情報が含まれているとき、その属性情報の表示にこのフィールドの第 1 バイトを使用するか (YES)、しないか (NO) を、属性キーワードで指定します。デフォルトは NO です。ATTR=YES が指定された場合、MFS プリプロセッサが内部でキーワード値を調整するため、LTH= キーワードおよび POS= キーワードでは、シミュレートされた属性バイトを考慮する必要はありません。ATTR=YES が指定されたときにとられる処置は次のとおりです。

#### **CURSOR**

(FIDS、FIDS3、FIDS4、FIDS7 の ABSOLUTE 出力のみ)。カーソルがこのフィールドの先頭に位置付けられます。

#### **NODISP**

他の属性指定がどうであれ、データが送信されません。

**HI** 第 1 バイトにアスタリスク (\*) が置かれます。

#### **MODIFIED**

第 1 バイトに下線文字 (\_) が置かれます。

#### **HI** で、**MODIFIED**

第 1 バイトに感嘆符 (!) が置かれます。

出力メッセージから属性情報が与えられないとき、第 1 バイトはブランクになります。

- DIV TYPE=OUTPUT で、DEV TYPE=DPM-An、DPM-Bn、3270P、FIDS、FIDS3、FIDS4、FIDS7、FIFP、FIJP、FIPB、FIS1、SCS2

このフィールドの最初の 1 バイトまたは 2 バイトに既存の 3270 属性が入っているか (YES)、いないか (NO)、また、拡張属性 (nn) があるかどうか、を属性キーワードで指定します。キーワードは、次のとおりさまざまな組み合わせで使用できます。

#### YES

既存の 3270 属性を IMS アプリケーション・プログラムからリモート・プログラムへ伝えるのに、このフィールドの最初の 1 バイトまたは 2 バイトを使用します。DIV ステートメントで SIM が指定されていればシミュレート形式、NOSIM2 が指定されていれば 2 進形式が使われます。(SIM なら MFS が属性をシミュレートし、NOSIM2 なら MFS が入力されたままのビット群を引き渡します。)

したがって、ATTR=YES が指定され、かつ OPTIONS=SIM または OPTIONS= 指定が省略されると、DFLD の長さに 1 バイトが追加されます。OPTIONS=NOSIM2 なら、DFLD の長さに 2 バイトが追加されます。これらのバイトは、リモート・プログラムに送信する属性バイトとして予約されています。

**NO** 既存の 3270 属性を IMS アプリケーション・プログラムからリモート・プログラムへ (シミュレート形式または 2 進形式で) 伝えるのに、このフィールドの最初の 1 バイトまたは 2 バイトを使用しません。これはデフォルトです。

**nn** 動的に変更できる拡張属性の数を、1 から 4 の範囲で指定します。無効な指定があると、1 と見なされます。指定された属性 1 個につき 2 バイト (2 x nn) が DFLD の長さに追加されます。追加されるバイトはデータに先行し、既存の 3270 属性バイトのために予約されたバイトの後ろにある場合と (YES)、後ろにあってはならない場合 (NO) があります。これらのバイトは、IMS アプリケーション・プログラムからリモート・プログラムへ拡張属性 (2 進形式) を引き渡すために使用されます。属性は、常に IMS アプリケーション・プログラムから与えられたまま送信され、シミュレートされたり、妥当性を検査されたりすることはありません。

#### YES,nn

2 つを組み合わせ、YES,nn とすれば、属性と拡張属性の両方が送信されます。この場合、説明したように SIM と NOSIM2 のどちらが指定されたかによって、次のようになります。

SIM との組み合わせでは、シミュレートされた 3270 属性 (1 バイト) に加えて、このフィールドの拡張属性 (2 x nn バイト) も、IMS アプリケーション・プログラムからリモート・プログラムへ送信することを意味します。これらの属性をリモート・プログラムへ渡すのに用いられるバイトの総数は、1 + (2 x nn) です。

NOSIM2 との組み合わせでは、2 進形式の 3270 属性 (2 バイト) に加えて、このフィールドの拡張属性 (2 x nn バイト) も、IMS アプリケー

ション・プログラムからリモート・プログラムへ送信することを意味します。これらの属性はすべて 2 進形式であり、その全部をリモート・プログラムへ渡すのに用いられるバイトの総数は  $2 + (2 \times nn)$  です。

#### **NO,nn**

NO,nn の組み合わせは、拡張属性だけを送信することを意味します。したがって、2 進形式で送信されるバイトの総数は、 $(2 \times nn)$  となります。

有効な指定と、そのとき確保しなければならないバイト数は、次のとおりです。

```
For DIV ,OPTION=NOSIM2 then:
DFLD  ,ATTR=(YES,nn)    2 + (2 × nn)
DFLD  ,ATTR=(NO,nn)    2 × nn
DFLD  ,ATTR=(,nn)      2 × nn
DFLD  ,ATTR=YES        2
DFLD  ,ATTR=NO         0
For DIV ,OPTION=SIM or not specified then:
DFLD  ,ATTR=(YES,nn)    1 + (2 × nn)
DFLD  ,ATTR=(NO,nn)    2 × nn
DFLD  ,ATTR=YES        1
DFLD  ,ATTR=NO         0
```

#### **EATTR=**

出力 DFLD でのみ有効で、DEV TYPE=3270、3270-An、3270P、SCS1 のとき、このフィールドの拡張属性を定義します。

すべての拡張属性がすべての装置タイプに適用できるとはかぎりません。指定と装置タイプの関係が妥当であることを確認するには、当該装置のコンポーネントの解説書を参照してください。

オペランドでは、次の項目を指定します。

- フィールド強調表示の追加
- フィールド・カラー
- フィールドの枠取り
- 入力制御
- 実施する妥当性検査
- プログラム式シンボル・バッファのローカル ID

プログラム式シンボル・バッファから文字が選択され、フィールドに挿入されます。これらのオペランドは、どのような順序で指定しても構いません。オペランドで装置のデフォルト値が選択されると、それがデータ・ストリーム中に場所を確保する目的に使われ、指定された属性をアプリケーション・プログラムから変更できるようになります。

フィールドの強調表示方法を追加するには、次の値を使います。

**HD** 装置デフォルト

**HBLINK**

明滅

**HREV**

反転表示

**HUL** 下線

フィールドのカラーを指定するには、次の値を使います。

- BLUE
- RED
- PINK
- GREEN
- TURQ (ターコイズ)
- YELLOW
- CD
- NEUTRAL

最後の 2 つのオペランドは次のように使われます。

**CD** デフォルトを指定します。

#### **NEUTRAL**

装置依存色を指定します。つまり、NEUTRAL で実際に表示される色は装置ごとに異なります。一般に、1 面のプログラム式シンボルなら、ディスプレイでは白、プリンターでは黒です。3 面のプログラム式シンボルなら、ディスプレイでもプリンターでもマルチカラーとなります。

PX'00'、PX'hh'、PC'c'、EGCS、および EGCS'hh' の 5 つのオペランドは相互に排他的です。つまり、フィールドはこれらの特性の 1 つを持つことができますが、複数の特性を併せ持つことはできません。MFS はどの 3270 装置についても、指定された文字セットが正しくロードされているかどうかを確認はしません。IMS アプリケーション・プログラムは MFS バイパスを用いて、プログラム式シンボル・バッファをロードできます。

#### **PX'00' | PX'hh' | PC'c'**

既にロードされているプログラム式シンボル・バッファに対して指定されているローカル ID か、EGCS プログラム式シンボル・バッファに対応する値を指定します。

#### **PX'00'**

無指定と同じですが、この場合、アプリケーション・プログラムがプログラム式シンボル属性を動的に変更することにより、このフィールドに対してプログラム式シンボル・バッファを指定できます。

#### **PX'hh'**

X'40' から X'FE' の 16 進文字です。

#### **PC'c'**

X'40' から X'FE' の 16 進文字です。

#### **EGCS | EGCS'hh'**

3270 ディスプレイに関する出力 DFLD でのみ有効です。SCS1 装置タイプでは EGCS のみ有効で、EGCS 'hh' は指定できません。

DFLD ステートメントに拡張図形文字セット・リテラルを指定すると、強制的に拡張図形文字セット属性が与えられます。つまり、3270 ディスプレイでは EATTR=EGCS'hh' を指定する必要がありませんし、SCS1 装置タイプでは EATTR=EGCS を指定する必要がありません。3270 ディスプレイでは、プログラム式シンボル値 X'F8' が設定されます。

制約事項: SCS1 DFLD 拡張図形文字セット属性は、IMS アプリケーション・プログラムで変更できません。

3283 型式 52 用に EGCS フィールドを定義するときは、長さを偶数値にしてください。1 つの EGCS フィールドが複数の装置行にまたがるときは、どの装置行にも偶数個の印刷位置が確保されるように WIDTH= と POS= を指定しなければなりません。

#### **EGCS**

このフィールドのフィールド属性を拡張図形文字セットに設定します。また、2 バイト文字セットにも設定します。

#### **EGCS'hh'**

'hh' は、使用するプログラム式シンボル値です。'hh' の値は、X'40' から X'FE' の任意の 16 進値か、X'00' です。3270 ディスプレイの拡張図形文字セット指定から 'hh' を省略すると、プログラム式シンボル値 X'F8' が指定されたものと見なされます。SCS1 装置に hh を指定しても、無視されます。

IMS アプリケーション・プログラムからの動的変更によって拡張図形文字セット・データも受け入れる EBCDIC フィールドを定義するには、プログラム式シンボル属性を EGCS'00' と指定してください。

#### **VDFLD|VMFILL|VMFLD|VMFILL,VMFLD**

フィールドに対して行う妥当性検査の種類を、次のとおり指定します。

#### **VDFLD**

Default

#### **VMFILL**

全桁入力必須

#### **VMFLD**

必須フィールド

#### **VMFILL,VMFLD**

全桁入力必須と必須フィールドの組み合わせ

フィールドが保護されている (ATTR=PROT) か、妥当性検査属性を持つリテラルであると、妥当性検査属性の指定がリセットされ、メッセージが出されます。

フィールドの枠取りの指定には、次の値が使用されます。

#### **OUTL'hh'**

フィールドの枠取り値 'hh' によるフィールドの枠取り

#### **OUTL**

装置デフォルト

#### **BOX** 枠

#### **RIGHT、LEFT、UNDER、OVER**

個々にも指定でき、組み合わせても指定できる罫線

フィールドの枠取り値 'hh' は、X'00' から X'0F' の 2 桁の 16 進数です。それ以外の値が指定されると、装置デフォルトの X'00' と見なされます。以下の表に、各種フィールドの枠取りパターンの値を示しておきます。

表 21. フィールドの枠取り値

値	UNDER	RIGHT	OVER	LEFT
00				
01	X			
02		X		
03	X	X		
04			X	
05	X		X	
06		X	X	
07	X	X	X	
08				X
09	X			X
0A		X		X
0B	X	X		X
0C			X	X
0D	X		X	X
0E		X	X	X
0F	X	X	X	X

3270 ディスプレイと SCS1 プリンターでのフィールドの枠取りは、アプリケーション・プログラムのコードで動的に変更できます。左線、右線、上線、下線の位置は、装置ごとに異なります。

IBM 5550 ファミリーの装置を 3270 として使用する場合のフィールドの枠取りについて、以下に簡単に説明します。

#### 3270 ディスプレイ

左線と右線は、3270 の基本属性バイトの位置に印刷されます。現在行の上線と、1 つ前の行の下線は、同じ線です。

24 行目の下線は、アプリケーション・プログラム域とメッセージ域を分ける分離線と同じです。

#### SCS1 プリンター

左線と右線は、MFS が現行フィールドの前後に確保しているバイトに印刷されます。現在行の上線と、1 つ前の行の下線は、同じ線です。ページの最終行に下線が指定されると、そのページの最終行に下線、次ページの第 1 行に上線が描かれます。

隣接する 2 つのフィールドの間に 1 バイトの間隔があると、前のフィールドの右線と後ろのフィールドの左線が同じになります。

#### MIX|MIXD|MIX'*nn*'|MIXS|MIXS'*nn*'

DBCS/EBCDIC 混合フィールドを指定します。

#### 3270 ディスプレイ

**MIX** DBCS/EBCDIC 混合フィールド (DBCS/EBCDIC mixed field)



## MIXD

装置デフォルト

3270 ディスプレイの入力制御は、アプリケーション・プログラムから動的に変更できます。

## SCS1 プリンター

**MIX** DBCS/EBCDIC 混合フィールドで、SO/SI のブランク印刷。

**MIXS** DBCS/EBCDIC 混合フィールドで、SO/SI のブランク印刷は抑制。

### MIX'*nn*'

「*nn*」は SO/SI の対の最大数です。DBCS/EBCDIC 混合フィールドで、SO/SI のブランク印刷。

### MIXS'*nn*'

「*nn*」は SO/SI の対の最大数です。DBCS/EBCDIC 混合フィールドで、SO/SI のブランク印刷は抑制。

'*nn*' は、MFS メッセージ・エディターが使用するバッファ情報です。01 から 31 の 2 桁の 10 進数でなければなりません。MIX または MIXS を指定した場合、MFS デフォルト値が次のように計算されます。

**MIX** DFLD 長を 5 で割って 1 を加えた値と、31 のいずれか小さい方。

**MIXS** DFLD 長を 3 で割って 1 を加えた値と、31 のいずれか小さい方。

1 つのフィールドが継続行にまたがるときは、いずれかの方法でフィールド長から得られた値 '*nn*' + 1 が各行に割り当てられます。

SCS1 プリンターで、継続行にまたがる DBCS/EBCDIC 混合データが DBCS 文字で分割されると、MFS は最後の文字をブランクで置き換え、その文字を次行の先頭に置きます。その結果、印刷位置が 1 つ失われます。

## PEN=

このフィールドが検出されたときに選択すべきリテラル、あるいは実行すべきオペレーター制御機能を指定します。(1) '*literal*' が指定され、(2) フィールドが即時検出可能 (ATTR= オペランド) で、(3) ヌルまたはスペース指定文字を含んでいるときは、そのフィールドが検出されると、先行 DEV ステートメントの PEN オペランドで参照されているフィールドに、指定されたリテラルが置かれます (ただし、他の装置フィールドがどれも変更されない場合)。ほかに変更されている装置フィールドがあると、リテラルの代わりに疑問符 (?) が置かれます。リテラルの長さは、256 バイトを超えてはなりません。

(1) 制御機能が指定され、(2) フィールドが即時検出可能 (ATTR= オペランド定義) で、(3) ヌルまたはスペース指定文字を含んでいるときは、そのフィールドが検出されて、ほかに変更されている装置フィールドがなければ、指定された制御機能が実行されます。ほかに変更されている装置フィールドがあると、疑問符 (?) が置かれ、その制御機能は実行されません。指定できる制御機能は、次のとおりです。

#### **NEXTPP-PAGE ADVANCE**

現在の出力メッセージにおける次の物理ページを要求します。出力メッセージが進行中でなければ、特に応答はありません。

#### **NEXTMSG-MESSAGE ADVANCE**

進行中の出力メッセージがあれば、それをデキューし、キュー中の次の出力メッセージ (あれば) を送信するよう要求します。

#### **NEXTMSGP-MESSAGE ADVANCE PROTECT**

進行中の出力メッセージがあれば、それをデキューし、次の出力メッセージを送信するか、次のメッセージがなければ、そのことを伝える通知メッセージを返すよう、要求します。

#### **NEXTLP-NEXT LOGICAL PAGE**

現在のメッセージにおける次の論理ページを要求します。

#### **ENDMPPI-END MULTIPLE PAGE INPUT**

複数の物理ページから成る入力メッセージの終わりを指定します。

ENDMPPI は、既に受信されたデータがある場合のみ有効です。データ入力がないときに、複数ページ入力 (MPPI) を終わらせる機能ではありません。

#### **OPCTL=**

この装置フィールドが受信されたときにオペレーター制御要求の有無を調べるのに使う、TABLE ステートメントで定義されたテーブルの名前を指定します。OPCTL 処理は、入力装置データの処理時に実行されます。制御機能が選択されていれば、ほとんどの場合、その制御機能が即時に実行されます。IMS 入力メッセージは作成されません。

#### **SLDI=**

SCS1 プリンターでは、出力メッセージの行密度を 1 インチ当たりの行数で指定します。(SLDP= も参照してください)。SLDI= は、DEV ステートメントでも指定できます。SLDI= の値は 1 から 72 であれば有効です。指定する値は、対象装置のアーキテクチャーと整合していなければなりません (当該装置またはコンポーネントの解説書を参照してください)。

SLDI= が DEV ステートメントと DFLD ステートメントの両方に指定されていると、2 つの SLD データ・ストリームが作成されます。1 つはメッセージの先頭で送信され、行密度を設定します。もう 1 つはメッセージ内部 (SLDI= 指定があるフィールドの直前で、すべての垂直タブと改行文字の後ろ) で送信されます。メッセージ内の SLDI= 指定は、メッセージの先頭で設定された行密度を変更します。この新しい行密度は、明示的にリセットされるまで有効です。

#### **SLDP=**


SCS1 プリンターでは、出力メッセージの行密度を 1 インチ当たりのポイント数で指定します。(SLDI= も参照してください)。SLDP= は、DEV ステートメントでも指定できます。SLDP= の値は 1 から 72 であれば有効です。指定する値は、対象装置のアーキテクチャーと整合していなければなりません (当該装置またはコンポーネントの解説書を参照してください)。

SLDP= が DEV ステートメントと DFLD ステートメントの両方に指定されていると、2 つの SLD データ・ストリームが作成されます。1 つはメッセージの先頭で送信され、行密度を設定します。もう 1 つはメッセージ内部 (SLDP=

指定があるフィールドの直前で、すべての垂直タブと改行文字の後ろ) で送信されます。メッセージ内の SLDP= 指定は、メッセージの先頭で設定された行密度を変更します。この新しい行密度は、明示的にリセットされるまで有効です。

**重要:** 行密度設定 (SLDx) キーワードの定義では、用紙位置決めが正しく行われるよう注意してください。SLDx= の定義が不適切だと、用紙位置にずれが生じることがあります。また、SLDI= と SLDP= は併用できないことにも注意してください。DO ステートメントと ENDDO ステートメントに挟まれる DFLD ステートメントには、SLDI= も SLDP= も指定できません。

関連資料:

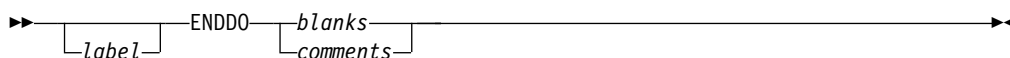
 MFS 出力メッセージの形式 (アプリケーション・プログラミング)

## ENDDO ステートメント

ENDDO ステートメントは、反復生成すべき DFLD ステートメントのグループの終わりを示します。

生成された DFLD ステートメントは、ENDDO ステートメントの直後に印刷されます。この定義で入力された DO ステートメントの 1 つ 1 つに対して ENDDO ステートメントが必要です。

フォーマット



パラメーター

### *label*

1 から 8 文字の英数字名を指定できます。使用されません。

## FMTEND ステートメント

FMTEND ステートメントは、装置形式定義を終了させます。装置形式定義の最後のステートメントとして必要です。

これが SYSIN 処理への入力の終わりなら、FMTEND ステートメントの後ろにさらに END コンパイル・ステートメントが必要です。

フォーマット



パラメーター

### *label*

1 から 8 文字の英数字名を指定できます。使用されません。

## 区画セット定義ステートメント

区画セット定義ステートメントには、PDB ステートメント、PD ステートメント、および PDBEND ステートメントがあります。

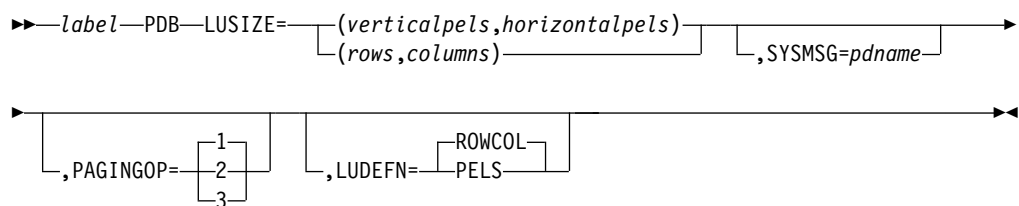
### PDB ステートメント

PDB ステートメントは、区画形式モードの 3290 装置または 3180 装置に対して、区画セット (区画記述子ブロック) の定義を開始します。

PDB ステートメントには、区画セット全体の特性を記述するいくつかのパラメーターが含まれています。メッセージの論理ページのフォーマットに区画セットが必要とされるときは、DEV ステートメントの PDB キーワードでこの PDB ステートメントの名前が参照されます。

どの PDB にも、少なくとも 1 個の PD ステートメントを指定しなければなりません。ただし、区画形式モードの 3180 では、各 PDB 内に指定できる PD ステートメントが 1 つに限られます。これは、3180 では 1 つの区画しか指定できないためです。区画形式モードの 3180 と 3290 にはこれ以外にも指定上の違いがあります。

### フォーマット



### パラメーター

#### *label*

PDB の名前。1 から 8 文字の英数字名 (pdbname) を指定してください。

#### **LUSIZE=**

PDB 定義の対象である LU ディスプレイの物理サイズを記述します。

LUDEFN=PELS なら、サイズは画素 (ピクセル) 数で指定されます。

LUDEFN=ROWCOL なら、サイズは行数と列数で指定されます (これがデフォルト値です)。3180 では、LUSIZE の指定に行数と列数を使用しなければなりません。

#### **SYMSMSG=**

システム・メッセージを表示する区画の名前 (*pdname*) を指定します。システム・メッセージ区画には、フィールドを 1 つだけ定義してください。システム・メッセージの端が切り捨てられないよう、この DFLD では少なくとも LTH=79 を指定しなければなりません。

現行 PDB でシステム・メッセージ区画を定義しているのであれば、すべてのシステム・メッセージがこの区画に送られます。システム・メッセージ区画を特に定義しないものの、現行 DOF で SYMSMSG フィールドを定義しているのであれば、システム・メッセージは、アクティブ区画のシステム・メッセージ・フィールドに送られます。さらに、現行 PDB でシステム・メッセージ用の区画を定

義しておらず、DOF にもそれ用のフィールドが定義されていないときは、現在の区画形式モードがシステム・メッセージによって破壊され、3290 は標準形式モードに戻ります。

#### PAGINGOP=

区画でのページ表示アルゴリズムをオプション番号 (1、2、3) で指定します。この 3 つのアルゴリズムは、区画セット中の各区画に対してメッセージの初期ページを表示するそれぞれの方法を指定します。また、3290 装置からページング要求を入力したときのページング動作も指定します。

3180 形式では、デフォルトの 1 を受け入れるか、このオペランドで指定しなければなりません。

#### LUDEFN=

PDB ステートメントの LUSIZE パラメーターと PD ステートメントの VIEWLOC パラメーターを、行数と列数で指定するのか、ピクセル数で指定するのかを定めます。すべての PD ステートメントが同じセル・サイズを使用しているのであれば、LUDEFN の指定は任意です。デフォルト (ROWCOL) で構いません。3180 形式では、ROWCOL を指定するか、デフォルトとして受け入れなければなりません。

同じ PDB 内の複数の PD ステートメント間で指定のセル・サイズが異なっているときは、必ず PELS を指定してください。

### PD ステートメント

PD ステートメントは、1 つの区画とその表示スペースを定義します。

PDB ステートメントで記述されるどの区画セットにも、少なくとも 1 個の PD ステートメントがなければなりません。ただし、区画形式モードの 3180 では、各 PDB 内に指定できる PD ステートメントが 1 つに限られます。

#### フォーマット

```
▶▶—label—PD—PID=nn—,VIEWPORT=(rrrrr,ccccc)—,VIEWLOC=————▶▶
▶ [ (rrrrr,ccccc) —————▶▶
  [ (verticalpels,horizontalpels) ] [ ,PRESPACE=(rrrrr,ccccc) ]
▶ [ ,WINDOWOF=rrrrr ] [ ,CELLSIZE=(hh,vv) ] [ ,SCROLLI=rows ]▶▶
```

#### パラメーター

##### label

1 から 8 文字の英数字名 (pdname) を指定しなければなりません。この名前は DPAGE ステートメントによって参照され、論理ページと区画の関連付けに使われます。

##### PID=

区画の区画 ID 番号を指定します。3290 形式では、00 から 15 の値が有効です。どの区画にも固有の PID がなければなりません。3180 形式では、指定すべき区画は 1 つだけなので、00 の値を使用します。

#### VIEWPORT=

区画のビューポート・サイズを指定します。 *rrrrr* は行数、*ccccc* は列数を表します。 3180 装置には次の制約があります。

- 列数が 80 以上なら、行数は 43 以下でなければなりません。
- 列数が 80 より大きく、132 以下なら、行数は 27 以下でなければなりません。

#### VIEWLOC=

ディスプレイ画面上におけるビューポートの位置を、画面の左上隅からの距離オフセットで指定します。 PDB ステートメントの LUDEFN パラメーターが ROWCOL なら、距離は行数と列数で表されます。 *rrrrr* は行数、*ccccc* は列数を表します。 LUDEFN パラメーターが PELS なら、距離は画面上端からのピクセル数と、画面左端からのピクセル数で表されます。 3180 用の形式を定義するときは、VIEWLOC を行数と列数で指定しなければなりません。

#### PRESpace=

表示スペース・バッファのサイズを行数と列数で指定します。 *rrrrr* は行数、*ccccc* は列数を表します。このパラメーターを指定しないと、VIEWPORT パラメーターに指定されたビューポートのサイズがデフォルトとなります。このパラメーターを指定する場合、列数パラメーターの指定はオプションです。指定を省略すると、VIEWPORT パラメーターの列数指定がデフォルトとして使われます。列数を指定するときは、VIEWPORT パラメーターの列数と同じにしなければなりません。

3180 形式でこのオペランドを指定するときは、行数と列数の積が 7680 より大きくなってはなりません。

#### WINDOWOF=

表示スペースの上端からビュー・ウィンドウの上端までの初期オフセット値を行数で指定します。これにより、表示スペースの表示すべき部分が、画面上のビューポートにマッピングされます。対話式処理でのオフセット変更には、スクロールを使用します。WINDOWOF のデフォルト値はゼロです。

#### CELLSIZE=

文字セルに含まれる水平ピクセル数と垂直ピクセル数を指定します。指定の順序が、MFS では例外的であることに注意してください。つまり、最初に文字セルの幅、ついで高さを指定します。これは、MFS での通常の指定順序とは逆です。

3290 では、6 × 12 ピクセルがデフォルトです (小さな文字の場合)。3290 で指定できる値は、6 × 12 から 12 × 31、または 00 × 00 です。00 × 00 を指定しておく、3290 装置が最も読みやすいセル・サイズを選択します。ただし、その場合は、MFS がビューポートの位置および起こりうるオーバーラップに関して妥当性を検査できません。したがって、ビューポートのサイズ指定と位置指定を正確に行ってください。

3180 では、このオペランドを、使用可能な画面区域のサイズに応じて、次のとおり指定してください。

- CELLSIZE=(12,12)
  - 24 × 80
  - 32 × 80

- 43 × 80
- CELLSIZE=(10,16)
- 27 × 132

#### SCROLLI=

スクロール機能を使用したとき、何行分のスクロールが起こるかを指定します。デフォルトのスクロール移動量は 1 行です。スクロール移動量がビューポートのサイズより大きいと、表示スペースの一部が画面上で見えなくなります。スクロール移動量に 0 を指定すると、スクロール機能が無効となります。

#### PDBEND ステートメント

PDBEND ステートメントは、区画セット定義 (区画記述子ブロック) を終わらせます。定義の最終ステートメントとして必要です。

これが SYSIN 処理への入力の終わりなら、PDBEND ステートメントの後ろにさらに END コンパイル・ステートメントが必要です。

フォーマット

```

▶▶PDBEND blanks
      └──comments──┘

```

### テーブル定義ステートメント

テーブル定義ステートメントには、TABLE ステートメント、IF ステートメント、および TABLEEND ステートメントがあります。

#### TABLE ステートメント

TABLE ステートメントは、オペレーター制御テーブルの定義を開始し、それに名前を付けます。このテーブルは、DFLD ステートメントの OPCTL キーワードで参照できます。

TABLE ステートメントと、それに続く IF ステートメントおよび TABLEEND ステートメントは、MSG 定義または FMT 定義の外になければなりません。

フォーマット

```

▶▶tablename TABLE blanks
      └──comments──┘

```

パラメーター

#### *tablename*

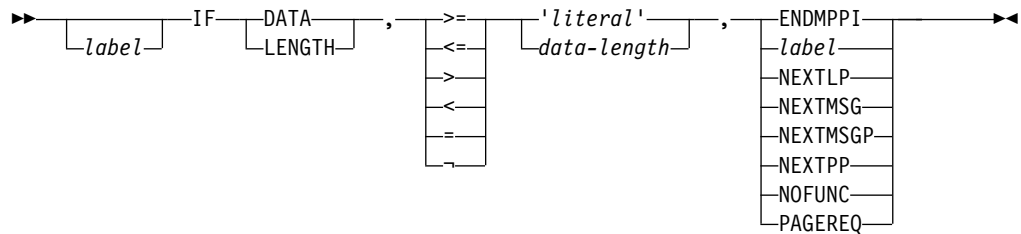
テーブル名として、1 から 8 バイトの英数字名を指定します。

#### IF ステートメント

IF ステートメントは、先行 TABLE ステートメントで名付けられたテーブルに 1 項目を定義します。

どの IF ステートメントも 1 つの条件付き操作を定義し、その条件が真だったとき実行される関連した制御機能もしくは分岐機能を指定しています。

## フォーマット



## パラメーター

### **label**

1 から 8 文字の英数字名を指定できます。先行の IF ステートメントに分岐機能が含まれているときは、このラベルが必要です。

### **DATA**

実行すべき条件付き操作を指定します。装置からこのフィールドに与えられたデータに対して、この操作が実行されます。

### **LENGTH**

条件付き操作で、このフィールドに入力された文字の数をテストします。このフィールドのサイズ制限は、DFLD の場合と同じです。

### **=、<、>、≠、≤、≥**

条件関係を指定します。その関係が真であれば、指定の制御機能が呼び出されます。

### **'literal'**

リテラル・ストリングです。入力データをこれと比較します。比較は、上段シフト文字に変換される前の入力に対して行われます。'literal' を指定するときは、最初のオペランドに DATA を指定しなければなりません。入力データの長さが literal ストリングの長さに等しくない、小さいほうの長さに合わせて比較が行われます。ただし、条件関係が ≠ で、データ長がゼロのときは、制御機能が実行されます。入力が下段シフト文字のときは、ALPHA ステートメントを使用し、リテラルを下段シフト文字で指定してください。

### **data-length**

整数値を指定します。このフィールドへの入力データの文字数とその整数値と比較されます。

### **ENDMPPI-END MULTIPLE PAGE INPUT**

複数の物理ページから成る入力の終わりを指定します (これが、作成中のメッセージの終わりです)。

### **label**

指定のラベルを持つ IF ステートメントでテストを続けます (分岐)。label は、この TABLE 定義で、現行ステートメントより後ろにある IF ステートメントのラベルでなければなりません (つまり、順方向の分岐機能でなければなりません)。



#### NEXTLP-NEXT LOGICAL PAGE

現在のメッセージにおける次の論理ページを要求します。

#### NEXTMSG-MESSAGE ADVANCE

進行中の出力メッセージがあれば、それをデキューし、キュー中の次の出力メッセージ (あれば) を送信するよう要求します。

#### NEXTMSGP-MESSAGE ADVANCE PROTECT

進行中の出力メッセージがあれば、それをデキューし、次の出力メッセージを送信するか、次のメッセージがなければ、そのことを伝える通知メッセージを返すよう、要求します。

#### NEXTPP-PAGE ADVANCE

現在の出力メッセージにおける次の物理ページを要求します。出力メッセージが進行中でなければ、特に応答はありません。

#### NOFUNC

条件付き機能テストを終了します。

#### PAGEREQ-LOGICAL PAGE REQUEST

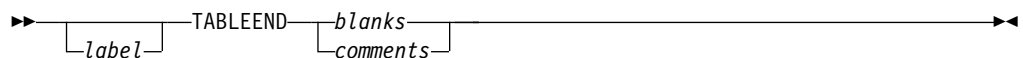
入力データの 2 番目から最後の文字までを論理ページ要求と見なします。

### TABLEEND ステートメント

TABLEEND ステートメントは、テーブル定義の終わりを指定します。

これが SYSIN 処理への入力の終わりならば、TABLEEND ステートメントの後ろにさらに END コンパイル・ステートメントが必要です。

#### フォーマット



#### パラメーター

##### *label*

1 から 8 文字の英数字名を指定できます。使用されません。

### コンパイル・ステートメント

コンパイル・ステートメントには、ALPHA ステートメント、COPY ステートメント、EQU ステートメント、RESCAN ステートメント、STACK ステートメント、UNSTACK ステートメント、TITLE ステートメント、PRINT ステートメント、SPACE ステートメント、EJECT ステートメント、および END ステートメントがあります。

#### ALPHA ステートメント

ALPHA ステートメントでは、MFS 言語ユーティリティーが英字と見なす文字セットを指定します。ここに指定された文字は、有効なフィールド名とリテラルの定義に使用されます。

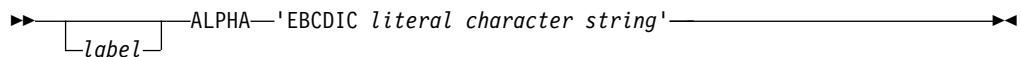
制約事項: 次の文字は、ALPHA で指定しても英字にはなりません。

b  $\phi$  \* < ( + !! \* ) ; , .  
 - / , % \_ > ? :  
 ' = "  
 0 から 9 まで

文字 A から Z、&; (X'50')、#、\$、@ は、MFS 言語ユーティリティでは常に英字と見なされます。

上に示されている文字はすべて、標準文字と呼ばれます。したがって、上記以外のすべての文字は標準外文字です。

### フォーマット



### パラメーター

#### **label**

1 から 8 文字の英数字名を指定できます。使用されません。

#### **'literal character string'**

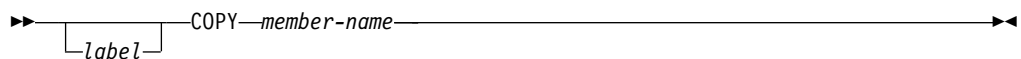
MFS 言語ユーティリティが英字と見なす文字を指定します。ALPHA ステートメントに EGCS リテラルを指定すると、エラー・メッセージが出ます。

### **COPY** ステートメント

COPY ステートメントは、SYSLIB DD ステートメントで表される区分データ・セットのメンバーのコピーを呼び出します。

コピーされたメンバーは、別のメンバーのネストされたコピーを要求できます。コピー要求のネスト・チェーンでは、コピーされるメンバーが既に上位レベルに存在してはなりません。コピーで使用できるネスト・レベルには、言語ユーティリティ・プリプロセッサが使用できるストレージ容量以外に制限はありません。各 COPY ステートメントのレベルは、印刷される各 COPY レコードの右側に印刷されています。

### フォーマット



### パラメーター

#### **label**

1 から 8 文字の英数字名を指定できます。使用されません。

#### **member-name**

ユーティリティ・プリプロセッサの入力ストリームの中へコピーする区分データ・セット・メンバーの名前を指定します。

### **EQU** ステートメント

EQU ステートメントは、置換変数としてのシンボルを定義します。



## EQU ステートメントの例

次の EQU ステートメントを見てください。

```
A EQU ATTR
AE EQU 'ATTR='
P EQU '(PROT,NUM)'
EP EQU '=(PROT,NUM)'
```

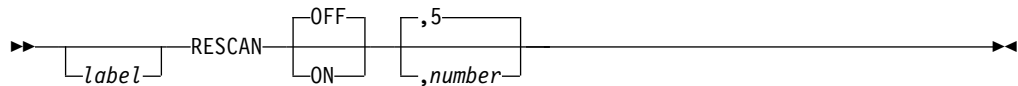
以下の例では同じ結果が生じます。

```
ATTR=(PROT,NUM)
ATTR=P
AE.P
A.EP
A=P
```

## RESCAN ステートメント

RESCAN ステートメントは、置換モードでの EQU ステートメントの動作を制御します。

フォーマット



パラメーター

### *label*

1 から 8 文字の英数字名を指定できますが、使用されません。

### OFF|ON

さらに置換が必要かどうかの確認に、置換テキストを再スキャンするか (ON)、しないか (OFF) を指定します。数値が指定されている場合を除き、デフォルトは OFF です。

ON を指定すると、置き換えられたテキストの内部でさらに (最大出現回数まで) 置換が起動されます。

### 5|*number*

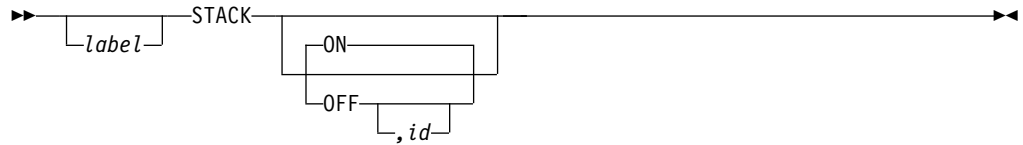
1 回の再スキャン置換で何回の再置換を認めるか指定します。デフォルトは 5 回です。最大数の「*number*」によって指定された回数を超える再帰的置換が試みられると、エラー・メッセージが出され、置換が打ち切られます。RESCAN ON,0 は RESCAN OFF と解釈されます。

## STACK ステートメント

STACK ステートメントは、1 つ以上の SYSIN または SYSLIB レコードを指定し、そのレコードを処理後も、後で再利用できるようにプロセッサ・ストレージに保管 (スタック) しておくように要求します。

SYSIN/SYSLIB レコードのスタックには、STACK ステートメントと UNSTACK ステートメントが含まれてはなりません。印刷されたレコードの右側に S の文字があるときは、そのレコードが将来の利用のためにスタックされていることを表します。

#### フォーマット



#### パラメーター

##### **label**

1 から 8 文字の名前を指定できますが、使用されません。

**ON** SYSIN/SYSLIB レコードのスタックの始まりを指定します。ON がデフォルトです。特に指定しなくても、スタッキングが開始されます。

##### オフ

SYSIN/SYSLIB レコードのスタックの終わりを指定します。

**id** レコード・スタックの名前を 1 から 8 文字の英数字で指定します。コンパイルでスタックを 1 つだけ使用するのであれば、ID は不要です。MFS が 8 個のブランクから成る ID をスタックに割り当てます。

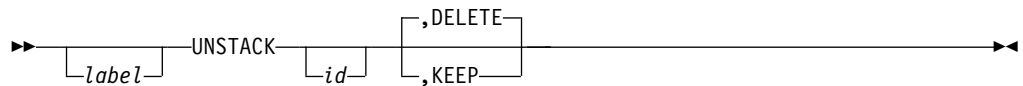
複数のスタッキング操作が要求されたときは、すべてのスタックに固有の ID が必要です。無名のスタックが 1 つだけ許されます。

#### UNSTACK ステートメント

UNSTACK ステートメントは、既に処理されている SYSIN/SYSLIB レコード・スタックの取り出しを要求し、その取り出されたスタックを処理後に削除するかどうか指定します。

印刷されたレコードの右側に U の文字があるときは、そのレコードが処理のためにプロセッサ・ストレージのスタックから読み取られていることを表します。

#### フォーマット



#### パラメーター

##### **label**

1 から 8 文字の英数字名を指定できます。使用されません。

**id** 取り出して処理するスタックの ID (1 から 8 文字) を指定します。ID が指定されないと、MFS は、8 個のブランクから成る ID を持つスタックを取り出します。



## フォーマット



## パラメーター

### **label**

1 から 8 文字の英数字名を指定できます。使用されません。

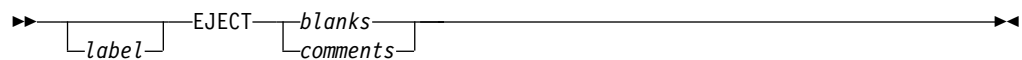
### **1|number**

このステートメント以後、何行スキップするかを指定します。デフォルトは 1 です。

## **EJECT** ステートメント

EJECT ステートメントは、出力の書き出しでページ替えに使用されます。EJECT ステートメントは印刷されます。

## フォーマット



## パラメーター

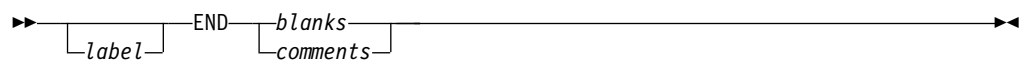
### **label**

1 から 8 文字の英数字名を指定できます。使用されません。

## **END** ステートメント

END ステートメントは、SYSIN 処理への入力の終わりを指定します。このステートメントを省略すると、自動的に END ステートメントが作成され、エラー・メッセージが出されます。

## フォーマット



## パラメーター

### **label**

1 から 8 文字の英数字名を指定できます。使用されません。

---

## 標準モードでのユーティリティーの実行

DFSUPAA0 ユーティリティーは、ここで説明する MFSUTL プロシーチャーを使用した 2 ステップ・モードの操作を使って、標準モードで実行することができます。

以下の図は、MFSUTL プロシージャラーを使用した 2 ステップ標準モードの操作の概要を示しています。

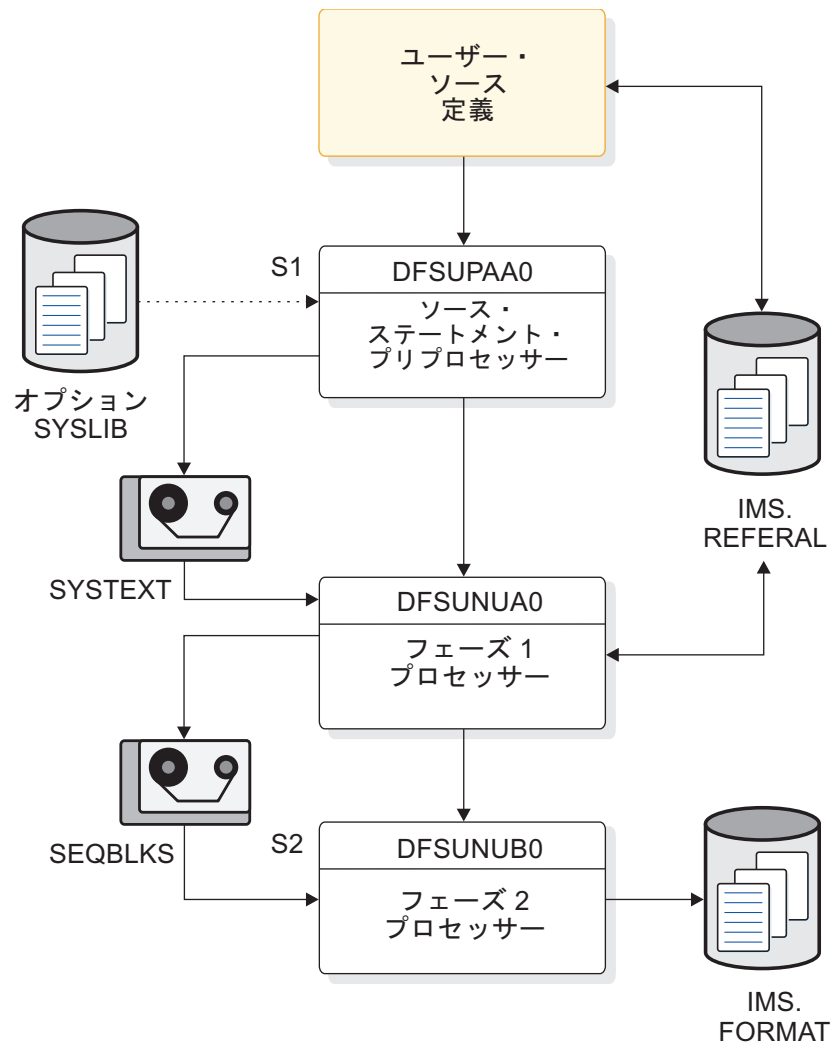


図 10. MFSUTL プロシージャラーの全体のフロー

次のトピックでは、ステップ 1 (S1) とステップ 2 (S2) の詳細について説明します。

### MFSUTL プロシージャラーの記述

**ステップ 1 (S1):** ソース・ステートメント・プリプロセッサ

MFS 言語ユーティリティーのプリプロセッサは、MFS 言語ソース・ステートメントに対する非アセンブラー、非マクロ型のプリプロセッサです。

プリプロセッサの実行により、中間テキスト・ブロック (ITB) が生成されます。それらの ITB は残りのユーティリティー・フェーズによって処理され、メッセージ (MSG) 制御ブロックおよび形式 (FMT) 制御ブロックが生成されます。IMS は、それらの制御ブロックを使用して、装置の表示とアプリケーション・プログラムへの提示のためにメッセージを形式化します。



プリプロセッサの主要な機能は、ユーザーの指定したソース定義の構文チェックと関連する妥当性チェックを行うことです。プリプロセッサは MSG、FMT、区画記述子ブロック (PDB)、および処理済みの TABLE ソース定義ごとに ITB を生成し、それらを IMS.REFERAL に保管します。

IMS.REFERAL の区分データ・セット (PDS) ディレクトリーには、MSG、FMT、PDB、および TABLE の ITB ごとに 1 つの項目が含まれています。さらに、IMS.REFERAL に保管された既知のすべての FMT ITB と MSG ITB の相互関係が PDS ディレクトリーに記録されます。

プリプロセッサは、以下の順序で実行されます。

1. プリプロセッサは、IMS.REFERAL に保管された既知のすべての FMT ITB と MSG ITB の相互関係を表す制御テーブルを作成します。EXEC ステートメントで圧縮機能を要求すると、プリプロセッサは IMS.REFERAL を圧縮します。
2. プリプロセッサは、ユーザー指定の FMT 定義および MSG 定義をその制御テーブルに追加し、それらの定義を制御テーブルに照らして解決します。この結果、ある定義が処理で指定されると、形式セットの制御ブロックがすべてフェーズ 1 で確実に再処理されます。この解決により、形式セット全体ではなく、変更する必要があるメッセージまたは形式のソースだけをコンパイルできるようになります。
3. 処理する定義を判別する解決機能が実行された後、プリプロセッサは、これらの FMT 定義および MSG 定義の制御ステートメントをフェーズ 1 処理用の SYSTEXT データ・セットに入れます。

PDB 定義を変更すると、その PDB を参照するそれぞれの形式セットを再コンパイルしなければならない場合があります。

以下の処置を行うと、形式セット ITB のフェーズ 1 の再処理により、IMS.FORMAT ライブラリー用の新しい FMT 制御ブロックおよび MSG 制御ブロックが作成されます。

- 制御ブロックとして IMS.FORMAT に既に存在している FMT 定義の修正。
- 制御ブロックとして IMS.FORMAT に既に存在している MSG 定義の修正。
- 形式セットへの新しい MSG の追加。
- 別の FMT への MSG 定義の再割り当て。この再割り当てにより、古い形式セットと新しい形式セットが再処理されます。

MSG 定義で (SOR= キーワードによって) 参照する FMT 名が、まだ MFS 言語ユーティリティーに指定されていない場合、プリプロセッサは MSG ITB を IMS.REFERAL ライブラリーに保管します。FMT 定義が指定されるまで、MSG 制御ブロックは作成されません。FMT 定義が指定されると、形式セットが処理され、IMS.FORMAT 用の新しい MSG 制御ブロックと FMT 制御ブロックが作成されます。

同様に、FMT 定義が MFS 言語ユーティリティーに指定されたが、この定義を参照する MSG がない場合、FMT ITB は IMS.REFERAL に保管されます。MSG 定義が少なくとも 1 つ指定されるまで、FMT 制御ブロックは作成されません。FMT 定

義が指定されると、形式セットが処理され、IMS.FORMAT 用の新しい MSG 制御ブロックと FMT 制御ブロックが作成されます。

IMS の各エラー・メッセージは、戻りコード 4、8、12、16、または 20 と共に戻され、この順序でエラーの重大度が増すことを示します。

FMT、MSG、PDB、または TABLE の ITB を作成するステートメントの処理時にエラー条件が検出されると、エラーを示すメッセージに関連付けられた最も高い重大度コードが保持されます。このコードは ITB を IMS.REFERAL ライブラリーに書き込むかどうかを判別する際に使用されます。プリプロセッサは、処理された定義ごとに発行された最も高い値の戻りコードを保持します。EXEC ステートメントの STOPRC パラメーターに比較値 (許容できない最も低い値の戻りコード) を指定することができます。戻りコードがこの STOPRC の値より大きいか等しい場合には、ITB は IMS.REFERAL には書き込まれません。例えば、STOPRC の値が 4 の場合、戻りコード 0 の ITB だけが書き込まれることとなります。STOPRC を指定しないと、値 8 が想定され、戻りコード 0 または 4 の ITB だけが書き込まれます。

プリプロセッサは、ジョブ実行中に処理されたすべての ITB で最も高い値の戻りコードも保持します。最も高い値の戻りコードが 16 またはそれ以上の場合、あるいは IMS.REFERAL に ITB がまったく書き込まれなかった場合には、プリプロセッサはフェーズ 1 に制御を渡しません。戻りコードが 16 またはそれ以上の場合には、プリプロセッサは、その戻りコードに等しい完了コードを伴って、制御を z/OS に戻します。

#### ステップ 1 (S1): フェーズ 1 プロセッサ

プリプロセッサはフェーズ 1 プロセッサを呼び出します。まず最初に、フェーズ 1 プロセッサはプリプロセッサにより SYSTEXT データ・セットに入れられた制御ステートメントを使用して、今回の実行で処理される FMT ITB と MSG ITB をすべて表すモジュール・テーブルを作成します。モジュール・テーブルを作成した後、フェーズ 1 プロセッサは IMS.REFERAL から ITB を読み取って、MSG 定義および FMT 定義ごとに制御ブロックを作成します。入力形式定義で制御機能の TABLE が要求されると、フェーズ 1 プロセッサは、IMS.REFERAL から TABLE ITB を入手して、機能を装置入力形式 (DIF) に作成します。

形式定義で HALDB 区画セットが要求されると、フェーズ 1 プロセッサは IMS.REFERAL から PDB ITB を入手して、区画制御機能を装置出力形式 (DOF) に作成します。

フェーズ 1 プロセッサは、新しく作成された制御ブロックを SEQBLKS データ・セットに入れます。処理されるメンバーごとに、SEQBLKS データ・セットに制御レコードがあります。このレコードはメンバー、メンバーのサイズ、作成の日時を示します。この制御レコードの後に、フェーズ 1 プロセッサによって作成された制御ブロックのイメージが続きます。

制御ブロックの作成の過程でエラーが検出されると、エラーのある定義に対するエラー制御レコードが SEQBLKS データ・セットに入れられます。このレコードは、エラーのあるメンバー、およびそのエラー制御レコードの作成日時を示します。さ

らに、フェーズ 1 プロセッサは z/OS に完了コード 12 を戻します。ステップ 2 を強制的に実行した場合、フェーズ 2 プロセッサは作成エラーのある制御ブロックを削除します。

フェーズ 1 プロセッサは、MFS 言語ユーティリティーの実行中に処理されたすべての ITB について、高い値の戻りコードを維持します。z/OS に戻る前に、フェーズ 1 プロセッサは、この高い値の戻りコードとプリプロセッサの高い値の戻りコードを比較します。2 つの戻りコードのいずれかの高い方の値が、ステップ 1 の完了コードとして z/OS に戻されます。

#### ステップ 2 (S2): フェーズ 2 プロセッサ

フェーズ 1 プロセッサが終了すると、フェーズ 2 プロセッサがジョブ・ステップとして制御を受け取ります。フェーズ 2 プロセッサは、2 パス・モードで動作し、新しい制御ブロックを IMS.FORMAT ライブラリーに入れます。最初のパスでは、フェーズ 2 プロセッサは SEQBLKS データ・セットを読み取り、フェーズ 1 プロセッサで作成された MOD、MID、DOF、および DIF のすべての名前を含む内部テーブルを作成します。フェーズ 1 プロセッサの実行時に作成エラーが生じた形式記述またはメッセージ記述の名前も、この内部テーブルに追加されます。IMS.FORMAT ライブラリーの制御ブロックで、IMS.FORMAT 内で置換されるブロック、またはフェーズ 1 で作成エラーが生じたブロックは、IMS.FORMAT から削除されます。

圧縮機能を要求すると、フェーズ 2 プロセッサは IMS.FORMAT ライブラリーを圧縮します。これにより、制御ブロック・メンバーの追加に使用可能なライブラリーのスペースが最大になります。さらに、フェーズ 1 プロセッサがすべての関連メンバーを再処理するために、関連制御ブロックをグループ化して、オンライン実行中に制御ブロックを取り出すときのシーク時間を低減することができます。

2 番目のパスでは、SEQBLKS データ・セットがモジュール・テーブルと共に再処理され、新しい制御ブロックが IMS.FORMAT に書き込まれ、ディレクトリー更新のためにこれらの制御ブロックに対して STOW が実行されます。

主記憶索引ディレクトリー \$\$IMSDIR に項目を持つ制御ブロックが削除され、かつ置換されていない場合、索引項目は削除しなければなりません。この更新は自動的に行うことができますが、大規模な形式ライブラリーが存在し、削除されているブロックが比較的少ない場合には、このような自動更新は非効率的です。これを避けるために、MFSUTL と MFSBTCH2 の両方のプロシージャーに、DIRUPDT=UPDATE|NOUPDATE パラメーターを使用することができます。デフォルトは UPDATE です。UPDATE を指定するか、またはデフォルトとして使用した場合、\$\$IMSDIR は自動的に更新されます。NOUPDATE を指定すると、更新処理はバイパスされます。この場合は、MFS サービス・ユーティリティーを使用して、\$\$IMSDIR からブロックを削除しなければなりません。

この実行で作成される新しい制御ブロック用に索引項目が \$\$IMSDIR に追加される場合は、MFS サービス・ユーティリティーの INDEX 機能を使用しなければなりません。

フェーズ 2 プロセッサは、MFS 言語ユーティリティーの特定の実行について、すべての制御ブロックが IMS.FORMAT に維持されているかに基づいて、ステップ 2 の完了コードを z/OS へ渡します。

## JCL の要件

次の図は、MFSUTL プロシージャの JCL を示しています。

```
//          PROC RGN=4M,SOUT=A,SYS2=,
//          SNODE='IMS',
//          SOR=NOLIB,MBR=NOMBR,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,COMPR2=COMPRESS,
//          LN=55,SN=8,DEVCHAR=0,COMPR3=NOCOMPRESS,
//          DIRUPDT=UPDATE
//S1       EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,
//          &COMPR,'LINECNT=&LN,STOPRC=&SN',
//          'DEVCHAR=&DEVCHAR')
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN   DD DISP=SHR,
//          DSN=&SNODE.&SOR.(&MBR)
//REFIN   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFOUT  DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFRD   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//SYSTEXT DD DSN=&&TXTPASS,UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//DUMMY   DD DISP=SHR,
//          DSN=IMS.&SYS2.PROCLIB(REFCPY)
//UTPRINT DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT&SOUT
//SEQBLKS DD DSN=&&BLKS,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//S2       EXEC PGM=DFSUNUB0,REGION=&RGN,
//          PARM=(&COMPR2,&COMPR3,&DIRUPDT,
//          'DEVCHAR=&DEVCHAR'),COND=(8,LT,S1)
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SEQBLKS DD DSN=&&BLKS,DISP=(OLD,DELETE)
//UTPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//FORMAT  DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
//DUMMY   DD DISP=SHR,
//          DSN=IMS.&SYS2.PROCLIB(FMTCPY)
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

DISP=OLD の指定は必須です。

制約事項: DD DUMMY の指定はサポートされていません。

### REFCPY 制御ステートメント

MFSUTL プロシージャは、この制御ステートメントを使用して REFERAL を圧縮します。

```
COPY INDD=REFOUT,OUTDD=REFOUT
```

## *FMTCPY* 制御ステートメント

MFSUTL プロシージャは、この制御ステートメントを使用して *FORMAT* を圧縮します。

```
COPY INDD=FORMAT,OUTDD=FORMAT
```

---

## バッチ・モードでのユーティリティーの実行

DFSUPAA0 ユーティリティーは、2 つの必須プロシージャ *MFSBTCH1* および *MFSBTCH2* を使用して、バッチ・モードで実行することができます。

バッチ・モードは、メッセージ記述子と装置形式を累積データ・セット *IMS.MFSBATCH* に入れる処理を一括して行う機能を提供します。このデータ・セットは、別個のジョブによって *MFS* ステージング・ライブラリー *IMS.FORMAT* に適用することができます。バッチ累積データ・セットを使用する場合は、*IMS.FORMAT* に入れられる前に、累積するすべての制御ブロックを維持できるだけの大きさに *IMS* システム・データ・セット *IMS.MFSBATCH* を割り振り、カタログを作成する必要があります。次の図は、バッチ・モードの概要を示しています。

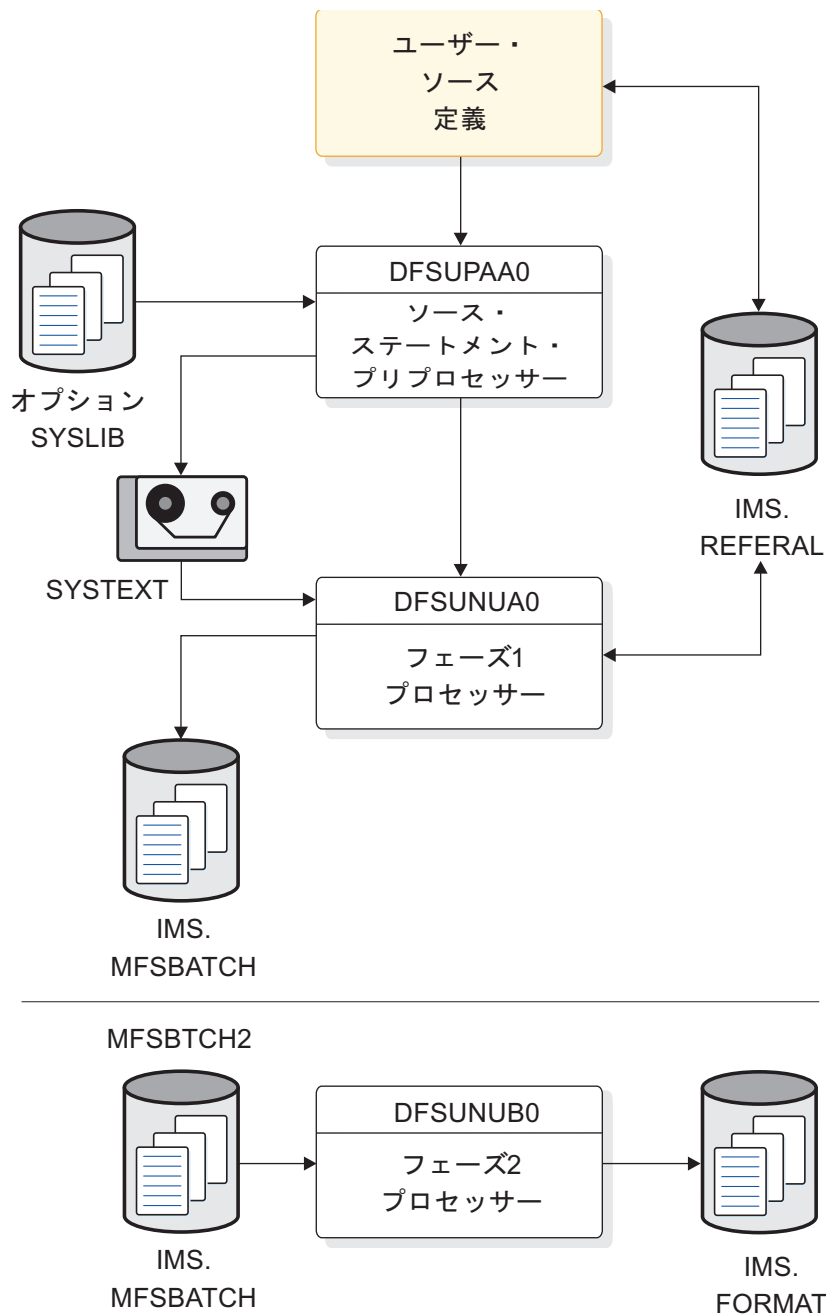


図 11. MFSBTCH1 および MFSBTCH2 プロシーチャーの全体のフロー

### MFSBTCH1 プロシーチャーの記述

このプロシーチャーは、292 ページの『MFSUTL プロシーチャーの記述』のステップ 1 と同じです。ただし、新しく構成された制御ブロックまたはエラー制御レコード (あるいはその両方) が SEQBLKS 累積データ・セット IMS.MFSBATCH に追加される点が異なります。

JCL の要件

以下の例は、MFSBTCH1 プロシージャの JCL を示しています。EXEC パラメータと DD ステートメントの詳細については、195 ページの『第 4 章 MFS 言語ユーティリティ (DFSUPAA0)』を参照してください。

```
//          PROC RGN=4M,SOUT=A,SYS2=,
//          SNODE='IMS',
//          SOR=NOLIB,MBR=NOMBR,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,LN=55,SN=8,DEVCHAR=0
//S1       EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,
//          &COMPR, 'LINECNT=&LN,STOPRC=&SN',
//          'DEVCHAR=&DEVCHAR')
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN   DD DISP=SHR
//          DSN=&SNODE..&SOR.(&MBR),
//REFIN   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFOUT  DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFRD   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//SYSTEM DD DSN=&&TXTPASS,UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//DUMMY   DD DISP=SHR
//          DSN=IMS.&SYS2.PROCLIB(REFCPY),
//UTPRINT DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//SEQBLKS DD DISP=(MOD,KEEP)
//          DSN=IMS.&SYS2.MFSBATCH
```

#### REFCPY 制御ステートメント

MFSBTCH1 プロシージャは、この制御ステートメントを使用して REFERAL を圧縮します。

```
COPY INDD=REFOUT,OUTDD=REFOUT
```

#### MFSBTCH2 プロシージャの記述

このプロシージャは、292 ページの『MFSUTL プロシージャの記述』のステップ 2 と同じです。ただし、次の段落で説明する点が異なります。

名前が重複する制御ブロックが検出されると、最後に検出されたブロックだけが記録されます。これにより、同じ名前を持つ制御ブロックが MFSBTCH1 プロシージャで複数回処理された場合に、IMS.FORMAT には最後に作成された制御ブロックが追加されます。IMS.FORMAT で制御ブロックが置換される場合、それらはまず IMS.FORMAT から削除されます。したがって、最後に作成された制御ブロックが作成時エラーとなり、このときに同じ名前のブロックが IMS.FORMAT に存在していると、この制御ブロックが IMS.FORMAT から削除されることとなります。

2 番目のパスでは、IMS.MFSBATCH がモジュール・テーブルと共に再処理され、新しい制御ブロックと、重複する制御ブロックのうち最後に現れたブロックが IMS.FORMAT に書き込まれ、ディレクトリ更新のためにこれらのブロックに対して STOW が実行されます。

このステップの終わりで、SEQBLKS データ・セットは MFSBTCH1 プロシージャーによる以後の使用が可能になるように空にされます。

### JCL の要件

以下の例は、MFSBTCH2 プロシージャーの JCL を示しています。EXEC パラメーターと DD ステートメントの詳細については、195 ページの『第 4 章 MFS 言語ユーティリティ (DFSUPAA0)』を参照してください。

```
//          PROC RGN=4M,SOUT=A,COMPR2=COMPRESS,
//          COMPR3=NOCOMPRED,DIRUPDT=UPDATE,SYS2=
//S2       EXEC PGM=DFSUNUB0,REGION=&RGN,
//          PARM='&COMPR2,&COMPR3,&DIRUPDT'
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SEQBLKS DD DISP=(OLD,KEEP),
//          DSN=IMS.&SYS2.MFSBATCH
//UTPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//FORMAT  DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
//DUMMY   DD DISP=SHR,
//          DSN=IMS.&SYS2.PROCLIB(FMTCPY)
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

### FMTCPY 制御ステートメント

MFSBTCH2 プロシージャーは、この制御ステートメントを使用して FORMAT を圧縮します。

```
COPY INDD=FORMAT,OUTDD=FORMAT
```

---

## テスト・モードでのユーティリティの実行

DFSUPAA0 ユーティリティは、ここで説明する MFSTEST プロシージャーを使用したテスト・モードの操作で実行することができます。

次の図は、MFSTEST プロシージャーを使用するテスト・モードの操作の概要を示しています。



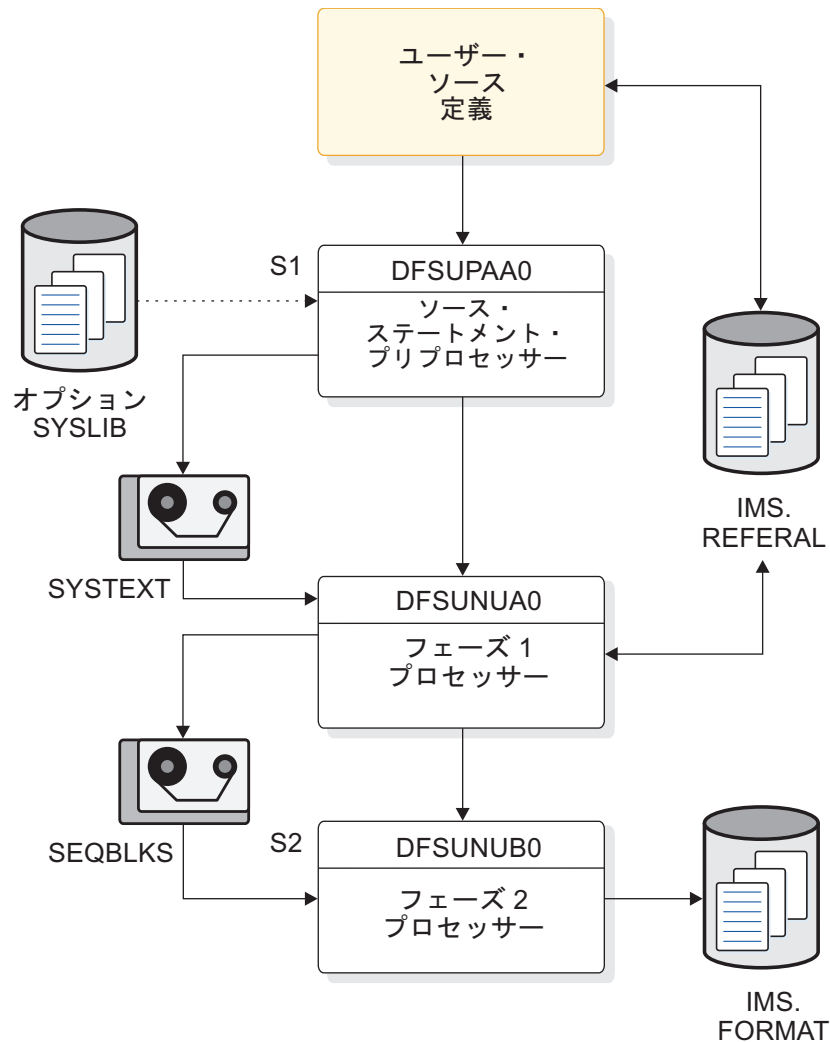


図 12. MFSTEST プロシージャの全体のフロー

## MFSTEST プロシージャの記述

MFSTEST プロシージャは、IMS 制御領域がアクティブなときに、メッセージ記述子と装置形式を作成するのに使用することができます。/TEST MFS コマンドを使用して、オンラインの実稼働アクティビティーを妨げることなく、MFSTEST により作成した制御ブロックをチェックすることができます。テストした制御ブロックは、MFSUTL プロシージャを使用してステージング・ライブラリーに入れることができます。MFSTEST プロシージャはステージング・ライブラリーを変更しないので、ITB と制御ブロックはすべてそのまま変更されません。

MFSTEST プロシージャを使用した場合は、MFS 言語ユーティリティーの 2 ステップの実行が異なります。

**ステップ 1 (S1):** ソース・ステートメント・プリプロセッサ

ソース・ステートメント・プリプロセッサは、ITB を一時ライブラリーに入れる点を除いて、MFSUTL プロシージャと同様に動作します。ヒストリー参照ライブラリー IMS.REFERAL の内容は、新しい MSG、FMT、PDB、または TABLE の ITB、あるいはこのテスト・モードの実行の結果としての新しい関係を反映するよう

には変更されません。IMS.REFERAL ライブラリーは、必要なすべての MSG ITB および FMT ITB が確実に処理されるようにする解決機能を実行するために、読み取り専用でのみ使用されます。

#### ステップ 1 (S1): フェーズ 1 プロセッサー

フェーズ 1 プロセッサーの動作は、MFSUTL プロシージャーによる動作と同様ですが、プリプロセッサーによって作成された連結一時ライブラリーと IMS.REFERAL ライブラリーから必要な MSG と FMT の ITB を読み取る点が異なります。

フェーズ 1 プロセッサーは、この実行で定義されるすべての MSG、FMT、PDB、および TABLE を、プリプロセッサーが作成した一時ライブラリーから入手します。関連する ITB が存在する場合、追加のブロックが IMS.REFERAL から入手されます。

#### ステップ 2 (S2): フェーズ 2 プロセッサー

フェーズ 2 プロセッサーは特殊形式ライブラリー IMS.TFORMAT に対して動作します。IMS.TFORMAT は、端末が MFSTEST モードの場合に、MFS 制御ブロックへのアクセスのために IMS 制御領域により使用されます。この実行の過程で新しいバージョンが作成されるか、またはエラーが検出された場合、フェーズ 2 プロセッサーはこのライブラリーから制御ブロックを削除します。次に、フェーズ 2 プロセッサーは、この実行の過程で作成された新しい制御ブロックを、オンライン・テストで使用できるライブラリーに挿入します。

IMS.TFORMAT は、IMS 制御領域がそこから並行して読み取りを行っている可能性があるため、圧縮されません。

推奨事項: IMS 制御領域が実行中でないときに、このデータ・セットを定期的に圧縮してください (IMS.TFORMAT に DISP=OLD を使用します)。

テスト形式データ・セットに \$\$IMSDIR がある場合、テスト・プロシージャーはそれを削除します。

## JCL の要件

次の図は、MFSTEST プロシージャーの JCL を示しています。

```
//          PROC RGN=4M,SOUT=A,SYS2=,
//          SNODE='IMS',
//          SOR=NOLIB,MBR=NOMBR,PXREF=NOXREF,
//          PCOMP=NOCOMP,PSUBS=NOSUBS,PDIAG=NODIAG,
//          COMPR=NOCOMPRESS,LN=55,SN=8,DEVCHAR=0
//S1       EXEC PGM=DFSUPAA0,REGION=&RGN,
//          PARM=(&PXREF,&PCOMP,&PSUBS,&PDIAG,
//          &COMPR,'LINECNT=&LN,STOPRC=&SN',
//          'DEVCHAR=&DEVCHAR')
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//*SYSLIB - USER OPTION
//SYSIN   DD DISP=SHR,
//          DSN=&SNODE..&SOR.(&MBR)
//REFIN   DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//REFOUT  DD DSN=&&TEMPPDS,
//          DCB=IMS.&SYS2.REFERAL,
//          UNIT=SYSDA,SPACE=(CYL,(5,1,10))
//REFRD   DD DSN=*.REFOUT,VOL=REF=*.REFOUT,
```

```

//          DISP=(OLD,DELETE)
//          DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//SYSTEXT DD DSN=&&TXTPASS,UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=800
//SYSPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//SEQBLKS DD DSN=&&BLKS,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//S2       EXEC PGM=DFSUNUB0,REGION=&RGN,
//          PARM='TEST,DEVCHAR=&DEVCHAR',
//          COND=(8,LT,S1)
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SEQBLKS DD DSN=&&BLKS,DISP=(OLD,DELETE)
//UTPRINT DD SYSOUT=&SOUT,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD SYSOUT=&SOUT
//FORMAT  DD DSN=IMS.&SYS2.TFORMAT,DISP=SHR

```

DISP=OLD の指定は必須です。

制約事項: DD DUMMY の指定はサポートされていません。

---

## MFS ライブラリー・バックアップ・プロシージャー

MFSBACK プロシージャーは、ユーティリティ・ライブラリーのバックアップ操作を実行します。

**重要:** このプロシージャーを使用する場合は、IMS.REFERAL ライブラリーと IMS.FORMAT ライブラリーが確実に同じレベル、つまり同時にダンプおよび復元されるようにしてください。この 2 つのライブラリーの内容を記述する関連情報が IMS.REFERAL PDS ディレクトリーにあるため、この操作を実行することが重要です。すべてのライブラリーが確実に同一レベルに復元されるようにするには、復元操作に先立って、すべての MFS データ・セットを消去して再割り振りしてください。ライブラリーが同一レベルに復元されない場合、予期しない動作が発生する可能性があります。

### MFSBACK プロシージャー

次の図は、MFSBACK プロシージャーの JCL を示しています。オプションの MFSTEST 機能も組み込まれています。DISP=OLD の指定はすべて必須です。

制約事項: DISP=OLD の指定が必須のステートメントでは、DD DUMMY の指定はサポートされません。

IMS.REFERAL ライブラリーのブロック・サイズを指定する場合は、800 バイトを指定する必要があります。

```

//          PROC  NODE='IMS',
//          TAPE=MFSDBS,SOUT=A,DSN=FORMAT,SYS2=
//*
//*****
//*
//* PROCEDURE KEYWORDS FOR // EXEC STATEMENT:
//*
//* NODE=  PREFIX LEVEL TO BE USED FOR
//*          ACCESS TO IMS MFS LIBRARIES.
//*
//* SYS2=  SECOND PREFIX LEVEL TO BE USER FOR

```



## MFS 復元プロシージャ

MFSREST プロシージャは、ユーティリティ復元操作を実行します。

**重要:** このプロシージャを使用する場合は、IMS.REFERAL ライブラリーと IMS.FORMAT ライブラリーが確実に同じレベル、つまり同時にダンプおよび復元されるようにしてください。この 2 つのライブラリーの内容を記述する関連情報が IMS.REFERAL PDS ディレクトリーにあるため、この操作を実行することが重要です。すべてのライブラリーが確実に同一レベルに復元されるようにするには、復元操作に先立って、すべての MFS データ・セットを消去して再割り振りしてください。ライブラリーが同一レベルに復元されない場合、予期しない動作が発生する可能性があります。

### MFSREST プロシージャ

次の図は、MFSREST プロシージャの JCL を示しています。オプションの MFSTEST 機能も組み込まれています。DISP=OLD の指定はすべて必須です。

**制約事項:** DISP=OLD の指定が必須のステートメントでは、DD DUMMY の指定はサポートされません。

```
//          PROC  NODE='IMS',
//          TAPE=MFSDBS,SOUT=A,DSN=FORMAT,SYS2=
//*
//*****
//*
//* PROCEDURE KEYWORDS FOR // EXEC STATEMENT:
//*
//* NODE=  PREFIX LEVEL TO BE USED FOR
//*        ACCESS TO IMS MFS LIBRARIES.
//*
//* SYS2=  SECOND PREFIX LEVEL TO BE USED FOR
//*        ACCESS TO IMS MFS LIBRARIES.
//*
//* TAPE=  RESTORE TAPE SERIAL NUMBER.
//*
//* SOUT=  SPECIFIES THE PRINT OUTPUT CLASS
//*        TO BE USED FOR PRINTED OUTPUT
//*        DURING THE RESTORE OPERATION.
//*
//*
//*****
//*
//MOVE1 EXEC PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3  DD SPACE=(CYL,(1,1)),UNIT=SYSDA
//REFERAL DD DSN=&NODE.&SYS2.REFERAL,DISP=OLD
//TAPEIN  DD UNIT=2400,LABEL=(1,SL),DISP=(OLD,KEEP),
//          VOL=(,RETAIN,SER=&TAPE),
//          DSN=&NODE.&SYS2.REFERAL,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//*****
//*
//* //MOVE1.SYSIN DD * MUST BE SUPPLIED BY THE
//* USER WITH THE APPROPRIATE COPY CONTROL
//* STATEMENT AS SHOWN BELOW:
//*
//* COPY OUTDD=REFERAL,INDD=TAPEIN
//*
//*****
//*
//MOVE2 EXEC PGM=IEBCOPY,PARM='SIZE=100K'
```

```

//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//FORMAT DD DSN=&NODE..&SYS2.&DSN,DISP=OLD
//TAPEIN DD UNIT=2400,LABEL=(2,SL),
//          VOL=(,RETAIN,REF=*.MOVE1.TAPEIN),
//          DSN=&NODE..&SYS2.&DSN,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//          DISP=(OLD,KEEP)
//*
//*****
//* //MOVE2.SYSIN DD * MUST BE SUPPLIED WITH *
//* THE APPROPRIATE COPY CONTROL STATEMENT *
//* AS SHOWN BELOW: *
//* *
//* COPY OUTDD=FORMAT,INDD=TAPEIN *
//* *
//*****
//*
//MOVE3 EXEC PGM=IEBCOPY,PARM='SIZE=100K'
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//FORMAT DD DSN=&NODE..&SYS1.TFORMAT,DISP=SHR
//TAPEIN DD UNIT=2400,LABEL=(3,SL),
//          VOL=REF=*.MOVE1.TAPEIN,
//          DSN=&NODE..&SYS2.TFORMAT,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//          DISP=(OLD,KEEP)
//*
//*****
//* //MOVE3.SYSIN DD * MUST BE SUPPLIED WITH *
//* THE APPROPRIATE COPY CONTROL STATEMENT *
//* AS SHOWN BELOW: *
//* *
//* COPY OUTDD=FORMAT,INDD=TAPEIN *
//* *
//*****
//*
```

---

## 第 5 章 プログラム仕様ブロック (PSB) 生成ユーティリティー

プログラム仕様ブロック (PSB) 生成ユーティリティーは、アプリケーション・プログラムの特性と論理端末および論理データ構造の使用を記述したマクロ命令から PSB を生成します。

IMS システムで ACB の IMS 管理が使用可能でない限り、IMS のもとでアプリケーション・プログラムを実行するには、事前に PSB 生成ユーティリティーが必要になります。ACB 保守ユーティリティーは、データベース記述子 (DBD) と一緒に PSB を使用して、実行時に使用するアプリケーション制御ブロック (ACB) を作成します。

IMS システムで ACB の IMS 管理が使用可能な場合、PSB 生成ユーティリティーを使用する代わりに、CREATE PROGRAMVIEW などの SQL 動的定義言語 (DDL) ステートメントを使用して、アプリケーション・プログラムの特性を IMS に対して記述します。

PSB 生成ステートメントでは、使用する IMS リソースの識別と、その特性を指定します。これらのプログラム連絡ブロック (PCB) は、アプリケーション・プログラムが使用するメッセージ宛先とデータベースを表します。さらに、アプリケーション・プログラム自体の特性を与えるステートメントも必要です。メッセージ・プログラム、バッチ・プログラム、または各高速機能プログラムごとに 1 つの PSB が必要です。PSB の名前およびそれに関連するアプリケーション・プログラムは、通信システム内で同じである必要があります。

1 つの入出力 PCB と 1 つの変更可能代替 PCB のみが必要な場合には、生成済みの PSB (GPSB) を使用して、アプリケーション・プログラムに必要なリソースを記述できます。GPSB は、すべてのオンライン環境で使用できますが、通常は DCCTL アプリケーション・プログラムで使用されます。GPSB のために PSBGEN を実行する必要はありません。

**重要:** PROCOPT 値によって、BMP アプリケーションに、データベースでのセグメントの挿入、置換、または削除を許可する場合は、変更をコミットせずに、BMP アプリケーションが更新するデータベースおよび HALDB 区画の総数が、両方で 300 を超えないようにしてください。

非コミット更新があるすべての全機能データベース・タイプが、この制限値 300 に不利に働きます。非コミット更新の件数に関する 300 というデータベース制限に近づく最も一般的な原因は、HALDB データベースの区画の数であるため、HALDB データベースを設計するには十分に注意してください。

サブセクション:

- 308 ページの『制約事項』
- 308 ページの『前提条件』
- 308 ページの『要件』
- 308 ページの『推奨事項』

- 『入力と出力』
- 311 ページの『JCL 仕様』

### 制約事項

現在、PSB 生成ユーティリティーについて文書化されている制約事項はありません。

### 前提条件

現在、PSB 生成ユーティリティーについて文書化されている前提条件はありません。

### 要件

現在、PSB 生成ユーティリティーについて文書化されている要件はありません。

### 推奨事項

現在、PSB 生成ユーティリティーについて文書化されている推奨事項はありません。

### 入力と出力

PSB 生成ユーティリティーにより、作成された PSB は PSB ライブラリーに入れます。各 PSB は、オペレーティング・システムの区分データ・セット IMS.PSBLIB のメンバーです。IMS バッチ実行 (DL/I 領域タイプ) の場合には、必要な PCB PSB は PSBLIB からロードされ、DL/I データベース PCB ステートメントの処理に必要な拡張 PSB がそれから作成されます。ACB 生成を実行して、拡張 PSB を ACBLIB に事前に組み込む必要があります。PSBLIB は、ACB 生成プロセスへの入力として使用されます。バッチ実行でもまた、JCL 実行ステートメントに領域タイプ「DBB」を指定することにより、ACBLIB からの事前組み込みブロックを使用することができます。オンライン領域 (BMP) 内で実行されているアプリケーションで、1 つ以上の GSAM PCB が定義された PSB を参照している場合、IMS は PSBLIB のある ACBLIB を使用して内部制御ブロックを構築します。その場合、PSB は ACBLIB と PSBLIB の両方で同じものとして定義してある必要があります。

PSB 生成で使用する 6 つのタイプのステートメントは次のものです。

- 入力メッセージのソース以外の出力メッセージの宛先用の PCB ステートメント。これらのステートメントは代替 PCB と呼ばれ、これらは、メッセージ処理プログラム、バッチ・メッセージ処理プログラム、および高速機能プログラム (これらは IMS メッセージ・キューとのインターフェースを持ちます) の中で使用されます。
- DL/I データベースおよび高速機能データベース用の PCB ステートメント。これらのステートメントは、メッセージ処理プログラム、バッチ処理プログラム、および高速機能処理プログラムが、データベースとのインターフェースを定義するとき使用します。



- アプリケーション・プログラム・センシティブな、データベース内のセグメント用の SENSEG ステートメント。これらのステートメントは、メッセージ処理プログラム、バッチ処理プログラム、高速機能処理プログラムが、論理データ構造を定義するときに使用します。
- アプリケーション・プログラムに依存する、セグメント内のフィールド用の SENFLD ステートメント。
- 各 PSB 用の PSBGEN ステートメント。このステートメントは、関連するアプリケーション・プログラムの特性を表すときに使用されます。
- PSBGEN ステートメントごとに必要なアセンブラ言語の END ステートメント。

PSB 生成に使用されるステートメントのリストには、入力メッセージ・ソース用の PCB は含まれていません。この目的のために、入出力 PCB が IMS オンライン制御プログラム中核内に存在しています。メッセージ処理に使用するアプリケーション・プログラムに入ったときに、入力メッセージのソースを指す PCB ポインターが、PCB アドレス・ポインターのリスト内の最初の項目として用意されます。PCB リストの残りの項目は、関連する PSB 内で定義された PCB と直接関係しており、PSB 生成時に定義された順序と同じ順序でアプリケーション・プログラムに定義する必要があります。DL/I メッセージおよびデータベース呼び出しを行う際には、すべての PCB をアプリケーション・プログラムで使用できます。特定の DL/I 呼び出しでは 1 つの PCB のみが使用されます。

PCB の名前を定義し (PCBNAME=name)、LIST=NO を指定すると、アプリケーション・プログラムに渡される PCB のリストから、代替 PCB、DL/I PCB、高速機能 PCB、および GSAM PCB を除外することができます。アプリケーション・インターフェース・ブロック (AIB) を使用して呼び出す場合には、PCB の名前を指定する必要があります。AIB は、すべてのタイプの PCB に使用できます。

バッチ処理領域のメッセージ処理プログラムまたはバッチ・メッセージ処理プログラムをテストするときは、PSBGEN ステートメントの CMPAT オプションを使用します。CMPAT=YES を指定すると、IMS はメッセージ処理領域で実行されているかのように、アプリケーションに PCB を提供します。CMPAT を使用すると、バッチ実行とオンライン実行の間にプログラムを再コンパイルする必要がなくなります。

バッチ・プログラムの場合、PSBGEN ステートメントの CMPAT オプションで要求しない限り、入出力 PCB はリスト内に存在しません。したがって、CMPAT=YES を指定しなければ、プログラムに与えられる PCB のリストは PSB 内の PCB と直接関係しています。バッチ処理領域でのバッチ処理の場合、TP PCB を PSB 内に入れてはなりません。

TM バッチ環境では、CMPAT=YES が暗黙指定され、PSBGEN でこれを変更することはできません。DCCTL バッチ領域で実行されるアプリケーション・プログラム用の PCB リストは、必ず 1 つの入出力 PCB を含んでいます。

IMS バッチ・メッセージ処理領域内で動作するバッチ・プログラムに関連する PSB の中に、代替 PCB を指定できます。これらの PCB は、出力メッセージ・キューイング用に利用できます。バッチ・メッセージ処理領域内で動作するバッチ・プロ

グラムは、入力メッセージ・キューのメッセージにアクセスできます。メッセージ処理プログラムの場合と同様に、1 つの入出力 PCB が必ず用意されています。

高速機能領域で実行される高速機能プログラムに関連する PSB の中に、代替 PCB および変更可能代替 PCB を指定できます。同じ PTERM を持つ応答代替 PCB を使用すれば、高速機能出力メッセージを、別のコンポーネントが端末に接続されている元の PTERM に送り戻すことができます。代替 PCB (非応答モード) を使用すれば、任意の端末または IMS メッセージ・キューに出力メッセージを送信できます。

DL/I 呼び出しを行い PCB 情報 (状況コードおよびフィードバック情報) の問い合わせを行うために、アプリケーション・プログラムへの入り口でアプリケーション・プログラムに渡される PCB リストを、アプリケーション・プログラム内に定義された名前参照できます。PCB のアドレスは、アプリケーション・プログラムから IMS への DL/I 呼び出しの中の 2 番目のパラメーターにすることができます。PCB アドレスは、入力メッセージのソース、出力メッセージの宛先、またはデータベースを表すことができます。DL/I 呼び出しが完了すると、PCB には、その呼び出しに関連する状況およびフィードバック情報が入ります。

#### 出力メッセージと統計

PSB 生成では 3 つのタイプの印刷出力と 1 つのロード・モジュールが作成されます。このロード・モジュールは、区分データ・セット (IMS.PSBLIB) のメンバーになります。出力のタイプは次のとおりです。

#### 制御ステートメントのリスト

これは、このジョブ・ステップへの入力ステートメント・イメージをリストしたものです。

#### 診断

制御ステートメントの処理中にエラーが検出されると、エラーが検出される前に読み取られた最後の制御ステートメントのイメージの直後に、診断メッセージが印刷されます。このメッセージは、直前の制御ステートメントまたは先行する制御ステートメントのグループを指し示すことができます。制御ステートメントごとに複数のメッセージを印刷することも可能です。この場合、それらのメッセージは、出力リスト上でそれぞれの後に続きます。すべての制御ステートメントが読み取られた後に、デック全体の論理についてさらに検査が行われます。これによって、さらに、1 つ以上の診断メッセージが出される場合があります。

エラーが検出されると、診断メッセージが印刷され、制御ステートメントがリストされ、残りの出力は抑制されます。しかし、PSB 生成の実行が終了する前に、すべての制御ステートメントが読み取られて、検査されます。制御ステートメント・エラーが検出された場合は、PSB 生成のバインド・ステップは実行されません。

#### アセンブラー・リスト

PRINT NOGEN を指定しない限り、PSB 生成の実行で作成される PSB のオペレーティング・システム・アセンブラー言語リストが出力されます。

#### ロード・モジュール

PSB 生成は、2 ステップからなるオペレーティング・システム・ジョブです。ステップ 1 は、オブジェクト・モジュールを生成するマクロ・アセンブリーの

実行です。ステップ 2 は、オブジェクト・モジュールのバインドであり、IMS.PSBLIB ライブラリーのメンバーになるロード・モジュールを生成します。

## JCL 仕様



PSB 生成ユーティリティーは、標準オペレーティング・システム・ジョブとして実行されます。EXEC ステートメントおよびユーティリティー制御ステートメントを定義する必要があります。

EXEC ステートメント

次の形式でなければなりません。

```
// EXEC PSBGEN,MBR=APPLPGM1
```

関連概念:

-  アプリケーション制御ブロック (ACBGEN) の作成 (データベース管理)
-  ACBLIB データ・セットの割り振り (システム定義)

---

## ユーティリティー制御ステートメント

ユーティリティー制御ステートメントは PSBGEN ユーティリティーの必須ステートメントとオプション・ステートメントを定義します。

入出力 PCB の PSB 生成では、PCB ステートメントは不要です。IMS がそれを自動的に作成します。このことが当てはまるのは、メッセージ処理アプリケーション・プログラム、バッチ処理アプリケーション・プログラム (IMS バッチ・メッセージ処理領域で動作し、IMS メッセージ・キューから入力メッセージを入手する必要があるもの)、および高速機能アプリケーション・プログラム (IMS 高速機能従属領域で動作するもの) です。IMS DB バッチ処理領域で動作するバッチ処理アプリケーション・プログラムは、PSBGEN マクロ・ステートメントで特別に要求しない限り、入出力 PCB を持ちません。

## 代替 PCB ステートメント

代替 PCB では、現行入力メッセージのソース以外の宛先を記述します。

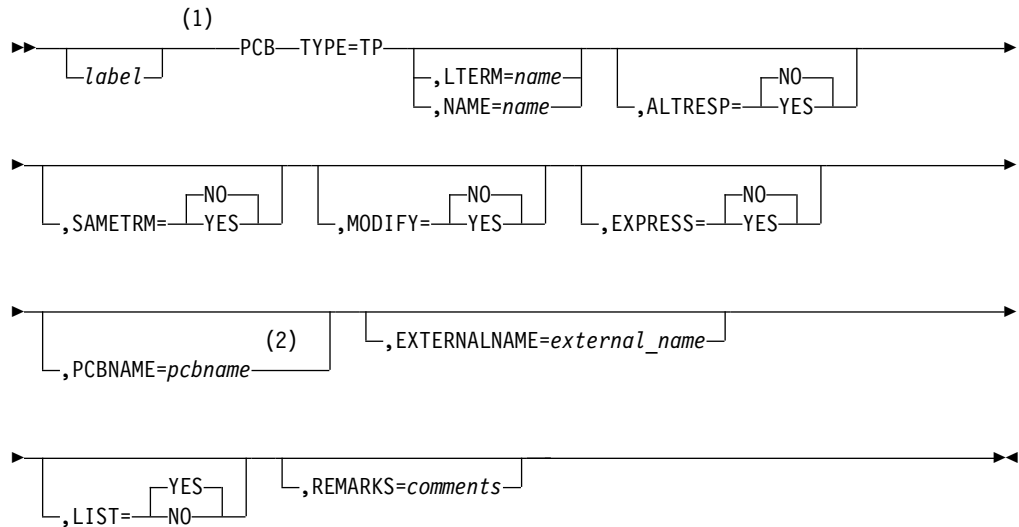
このステートメント命令により、アプリケーション・プログラムでは、入力メッセージのソース以外の宛先に出力メッセージを送信できます。

要件: 出力を送る宛先ごとに 1 つの PCB ステートメントが必要です。

出力メッセージは、出力端末か、あるいは別のプログラムが処理する入力トランザクション・キューのいずれかに送信することができます。それぞれの出力メッセージの宛先に、別個の代替 PCB 宛先が必要です。出力で応答する必要があるものが入力ソース端末のみならば、このタイプの PCB ステートメントを含めてはなりません。メッセージ処理プログラム、バッチ・メッセージ処理プログラム、および高速機能プログラムは、それぞれ関連する PSB の中に代替 PCB ステートメントを持つことができます。代替 PCB は、高速機能トランザクションへのメッセージの送

信に使用することはできません。しかし、高速機能アプリケーション・プログラムは、代替 PCB を使用して、任意の端末または IMS トランザクションにメッセージを経路指定することができます。

代替 PCB ステートメントは、PSB 生成制御カード・デッキの最初のものでなければならず、その後には IMS データベースに関連する PCB を識別するステートメントが続きます。以下の図は、代替 PCB ステートメントの形式を示しています。



注:

- 1 *label* および PCBNAME を同時に指定することはできません。*label* または PCBNAME= パラメーターのどちらかを使用してください。
- 2 *label* および PCBNAME を同時に指定することはできません。*label* または PCBNAME= パラメーターのどちらかを使用してください。

#### **label**

アセンブラ言語のステートメントで有効な、長さが 1 から 8 文字の長さの英数字ラベルを指定します。PSB 内の PCB ステートメントのラベルは、固有である必要があります。

例外: PCBNAME= パラメーターを指定する場合は、このパラメーターを指定しないでください。

#### **PCB**

PCB ステートメントであることを示します。

#### **TYPE=TP**

代替 PCB のすべてに必須のキーワード・パラメーターです。

#### **LTERM=|NAME=**

出力メッセージの宛先のパラメーターです。「name」はメッセージの実際の宛先で、論理端末名 (LTERM=) またはトランザクション・コード名 (NAME=) です。この名前がトランザクション・コード名の場合は、この PCB への出力メッセージは、NAME パラメーターで指定されたトランザクション・コードを処理するのに使用されるプログラムへ入力するために、エンキューされます。この

名前は、1 から 8 文字の長さの英数字でなければならず、論理端末名またはトランザクション・コードとしてユーザーの IMS システム定義に指定する必要があります。 LTERM= または NAME= パラメーターは、MODIFY=YES を指定した場合以外は指定する必要があります。

#### **EXTERNALNAME=**

PCB ラベルまたは PCBNAME= パラメーターのオプションの別名。Java アプリケーション・プログラムは、PCB を参照する場合にこの外部名を使用します。

外部名は 1 から 128 文字の大文字の英数字ストリングとして指定します。外部名には下線文字を含めることができます。

外部名は PSB 内で固有でなければなりません。

EXTERNALNAME を指定しない場合、デフォルトの外部名は PCB ラベルか PCBNAME (いずれか指定されている方) になります。

PCB ラベルも PCBNAME も指定されていない場合、EXTERNALNAME はデフォルトでブランクになります。

制約事項: 外部名は予約済み SQL キーワードにすることも、DFS で開始することもできません。

EXTERNALNAME パラメーターが指定されておらず、NAME パラメーターに予約済み SQL キーワードが指定されている場合、EXTERNALNAME は NAME 値に「\_SCH」を付加したうえで、その NAME 値をデフォルトの外部名として受け入れます。

#### **ALTRESP=**

応答モード、会話型モード、または排他モードで端末に応答するために、入出力 PCB の代わりにこの代替 PCB を使用できるか (YES)、できないか (NO) を指定します。デフォルトは NO です。ALTRESP=YES は、代替 PCB についてのみ有効です。

#### **SAMETRM=**

応答代替 PCB に名前を指定された論理端末が、入力メッセージの発信元となった論理端末と同じ物理端末に割り当てられているかどうかを IMS が検証するか (YES)、またはしないか (NO) を指定します。デフォルトは NO です。応答モードの端末で動作するプログラムおよび会話型プログラムが使用する応答代替 PCB については、SAMETRM=YES を指定する必要があります。排他モードの出力専用装置にメッセージを送信するために代替応答 PCB を使用する場合には、SAMETRM=NO を指定する必要があります。

#### **MODIFY=**

代替 PCB を修正できる (YES) ようにするかどうかを指定します。この機能により、この PCB に関連する宛先名を動的に修正することができます。デフォルト値は NO です。MODIFY=YES を指定した場合は、NAME= または LTERM= パラメーターを省略してください。

#### **EXPRESS=**

アプリケーション・プログラムが異常終了した場合に、この代替 PCB からのメッセージを送信するか (YES)、バックアウトするか (NO) を指定します。

**YES** この指定は、プログラムが異常終了するか、ROLL または ROLB 呼び

出しを出した場合でも、EXPRESS メッセージを宛先端末に送信できることを示します。これらの条件下にあるすべての PCB について (高速でも非高速でも)、挿入されても伝送可能になっていないメッセージは取り消されますが、伝送可能にされたメッセージは取り消されません。

非高速 PCB の場合は、プログラムが同期点 (コミット・ポイント) に達するまで、メッセージを宛先に伝送することはできません。同期点が起こるのは、プログラムが終了するか、CHKP 呼び出しを出すか、あるいは次の入力メッセージを要求する (トランザクションが MODE=SNGL で定義されているとき) 場合です。

高速 PCB の場合は、IMS が完全なメッセージを持っていることを認識している場合に、メッセージは宛先に伝送できます。メッセージは、PURG 呼び出しがその PCB を使用して行われるか、あるいはプログラムが次の入力メッセージを要求するときに、利用可能です。

IMS システム定義で PSB が高速機能アプリケーションとして定義されているときは、EXPRESS=YES を指定しても、応答代替 PCB については、実行時に無視されます。

**NO** この指定は、アプリケーション・プログラムが異常終了すると、メッセージがバックアウトされることを表します。NO はデフォルトです。

#### **PCBNAME=**

PCB の名前を指定します。PCB の名前は、標準命名規則に従った 8 バイトの英数字の文字ストリングでなければなりません。PCB の名前は、PSB 内で固有でなければなりません。

例外: ラベルを使用する場合は、このパラメーターを指定しないでください。

#### **LIST=**

入り口でアプリケーション・プログラムに渡される PCB リストに、名前を指定した PCB を含めるかどうかを指定します。指定した PCB を PCB リストに含めるときは、YES を指定します。指定した PCB を PCB リストから除外するときは、NO を指定します。YES はデフォルトです。

PCB リストから PCB を除外するときは、PCBNAME= パラメーターで PCB に名前を指定する必要があります。アプリケーション・プログラムが PCB のアドレスを必要としない場合には、LIST=NO を指定します。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

### CORRECT

REMARKS='These remarks apply to the XYZ application'

### INCORRECT

REMARKS='These remarks apply to the 'XYZ' application'

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## 全機能または高速機能データベースの PCB ステートメント

PSB 生成入力レコードの 2 番目のタイプのステートメントは、DL/I データベースまたは高速機能データベース用の PCB の記述を指定します。

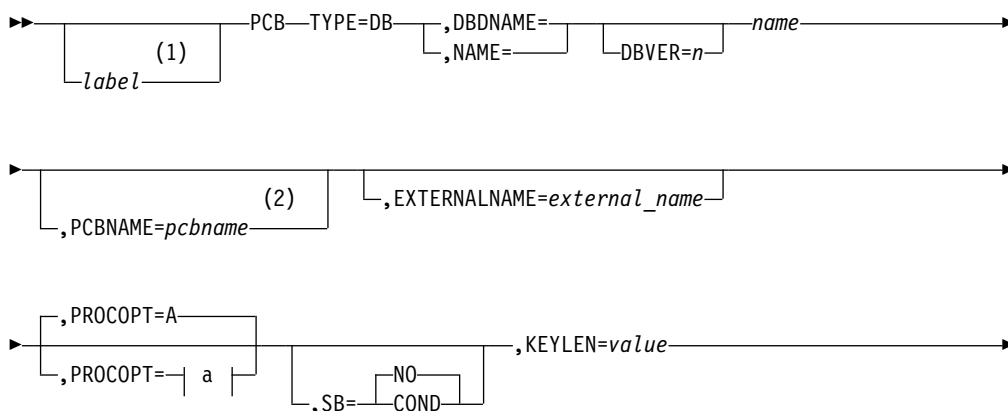
通常は、1 つ以上のデータベース PCB が PSB 内に含まれますが、2 番目のタイプのステートメントは必ずしも必要ではありません。例えば、メッセージ通信プログラムまたは会話型メッセージ・プログラムでは、DL/I データベースへのアクセスを要求することができません。そのため、データベース PCB は不要です。

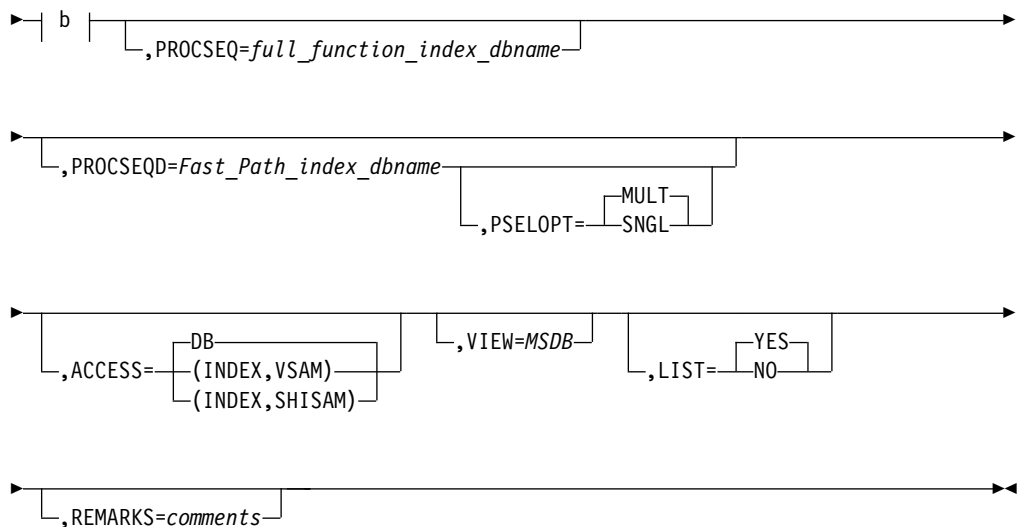
DCCTL 環境では、データベース PCB (GSAM PCB は除く) はサポートされていませんが、PSBGEN に含めることができます。DCCTL 環境で実行されるアプリケーション・プログラムが、データベース PCB を使おうとすると、AD 状況コードを受け取ります。

PSBGEN に定義できるデータベース PCB の最大数は、代替端末 PCB を含め 2500 です。これは、すべての IMS 領域タイプ (MSG、DL/I など) 内で実行されるアプリケーション・プログラムに対する最大値です。

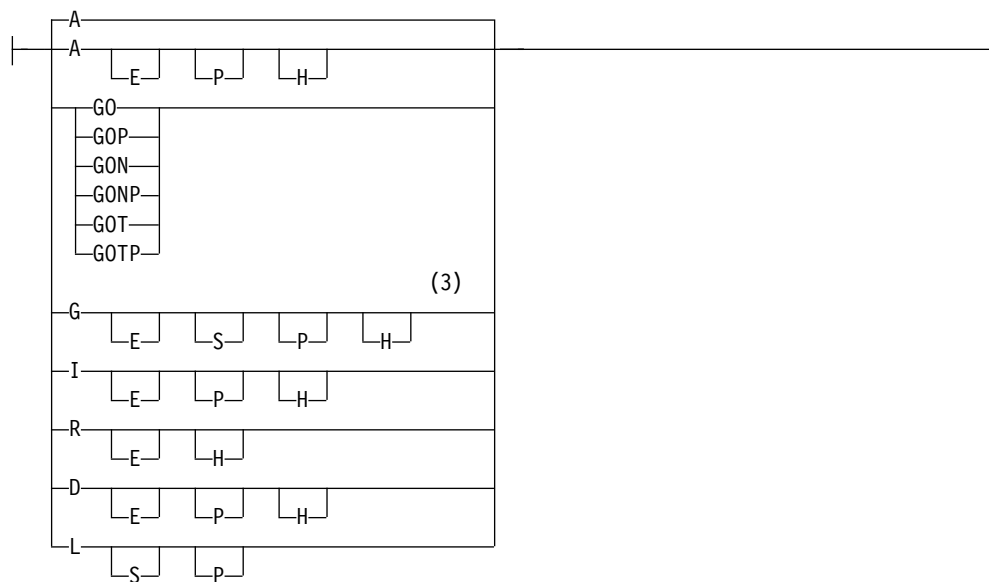
高速機能副次索引 PCB が PSB 内の唯一の PCB である場合、関連する DEDB PCB をその PSB に含める必要があります。最小の DEDB PCB には、関連する DEDB データベースのルート・セグメント用の SENSEG ステートメントを指定する必要があります。

以下の図は、DL/I データベースの PCB ステートメントの形式を示しています。





**a:**



**b:**



注:

- 1 *label* および PCBNAME を同時に指定することはできません。label または PCBNAME= パラメーターのどちらか 1 つのみを使用してください。
- 2 *label* および PCBNAME を同時に指定することはできません。label または PCBNAME= パラメーターのどちらか 1 つのみを使用してください。



- 3 これらのオペランドは、任意の組み合わせが選択できます。G、I、R、および D が選択されている場合は、代わりに A を使用します (A = G、I、R、および D の組み合わせ)。

#### **label**

特定の PCB を DFSCTL ファイルの SBPARM 制御ステートメントで参照するために使用するオプション・ラベル。指定する場合は、アセンブラ言語ステートメントで有効な 1 から 8 バイトまでの英数字からなる文字ストリングにする必要があります。PSB 内の PCB ステートメントのラベルは、固有でなければなりません。

例外: PCBNAME= を使用する場合は、このパラメーターを指定しないでください。

#### **TYPE=DB**

すべての DL/I データベースの PCB に必須のキーワード・パラメーターです。

#### **DBDNAME= または NAME=**

物理 DBD または論理 DBD の名前を指定するパラメーターです。これは、この論理データ構造のデータベース・セグメントの基本ソースに使用されます。この PCB の下で 1 つ以上の SENSEG ステートメントで定義される論理構造は、関連するアプリケーション・プログラムがセンシティブであるデータ・セグメントの階層セットです。データ・セグメントのこの論理階層は、物理階層として存在する場合もあり、存在しない場合もあります。物理階層として存在するかどうかは、SENSEG ステートメントで定義されたセグメントの関係、およびデータベース記述 (DBD) で定義された 1 つ以上のデータベースにそれらのセグメントが存在するかどうかによって決まります。このステートメントに続き、しかも次の PCB または PSBGEN ステートメントの前にある SENSEG ステートメントはすべて、この PCB の DBDNAME= または NAME= パラメーターで指定された DBD で定義されているセグメントを参照している必要があります。

キーワード DBDNAME および NAME は同義です。DBDNAME の方が記述的です。NAME は、以前のリリースとの互換性のために保持されています。

#### **DBVER=n**

データベースのバージョン管理が使用可能な場合、このアプリケーション・プログラムで必要なデータベース定義 (DBD) のバージョン番号を指定します。

指定される数値は、DBD で定義され、IMS カタログに保管されているバージョン番号と一致している必要があります。データベース・バージョン番号の有効値は、0 から 2147483647 です。

PSB 内の複数の PCB が同じデータベースを参照している場合、それぞれの PCB で、同じ DBD バージョン番号を指定する必要があります。

#### **PCBNAME=**

PCB の名前を指定します。PCB の名前は、標準命名規則に従った 8 バイトの英数字の文字ストリングでなければなりません。

例外: PCB ステートメントに *label* を指定する場合は、このパラメーターを指定しないでください。

## PROCOPT=

この PCB で宣言されているセンシティブ・セグメントに関する処理オプションを指定します。これらの指定オプションは、関連するアプリケーション・プログラムで使用できます。このパラメーターで最大 4 つのオプションを使用できます。パラメーター中の文字の意味は次のとおりです。

- A** すべてのオプション。デフォルトでは、PROCOPT=A には G (取得)、I (挿入)、R (置換)、および D (削除) の各オプションが含まれています。PROCOPT=A はデフォルトの設定値です。
- G** Get オプション。
- I** 挿入オプション。デフォルトでは、PROCOPT=I には高速機能 DEDB の G (取得) オプションが含まれています。その他のデータベース・タイプの G オプションは、PROCOPT=I には含まれていません。
- R** 置換オプション。デフォルトでは、PROCOPT=R には G (取得) オプションが含まれています。
- D** 削除オプション。デフォルトでは、PROCOPT=D には G (取得) オプションが含まれています。
- P** パス CALL です。コマンド・コード D を使用する場合は必須です。ただし、フィールドに依存しないバッチ・プログラムでの ISRT 呼び出しの場合は除きます。DEDB を処理する際にコマンド・コード D を使用する場合は、PROCOPT=P は必須ではありません。P は、A (すべて)、G (取得)、I (挿入)、D (削除)、および L (ロード) の各オプションと共に使用されます。
- O** PCB に O オプションを使用すると、IMS は戻されたセグメントの所有権を検査しません。したがって、保全性なしの読み取りプログラムでは、別のプログラムによって更新されたセグメントを取得する可能性があります。更新プログラムが異常終了してバックアウトした場合は、現在も過去もデータベースに存在していないセグメントを保全性なしの読み取りプログラムが取得することになります。あるセグメントが削除され、同じ位置に同じタイプの別のセグメントが挿入された場合は、セグメント・データ、およびアプリケーションへ返された後続のすべてのデータが、異なるデータベース・レコードから取得される可能性があります。したがって、O オプションを使用する場合は、そのオプションで読み取られたデータに基づいて更新を行わないでください。O は、GO、GON、GONP、GOT、GOTP、または GOP いずれかのみとして指定する必要があります。

IMS はこれらのエラー・タイプのいくつかを認識し、それらを異常終了 U0849 に変換します。ただし、PROCOPT GOx の場合に起きるその他の条件は、保全性なしの読み取りが原因であるために検出されません。この場合、ループ、ハング、およびシステムの異常終了が発生するおそれがあります。この PROCOPT を使用する場合は、システム設計を慎重に検討して、並行更新アクティビティーがこのような種類の条件の発生リスクを高める可能性があるかどうかを判断する必要があります。

- N** 読み取り専用アプリケーション・プログラムが起こす異常終了の数を減らします。読み取り専用アプリケーション・プログラムは、別のアプリケーション・プログラムが更新しているデータを参照できます。このこ

とが行われるときには、そのデータを指す無効なポインターが存在する場合があります。無効なポインターが検出されると、読み取り専用アプリケーション・プログラムは異常終了します。N を指定すると、この状態を回避できます。代わりに、GG 状況コードがプログラムに戻されます。プログラムは、処理を終了するか、別のセグメントを読み取って処理を続行するか、あるいは別のパスを使用してデータにアクセスするかを決定する必要があります。N を指定する場合は、GON、GONH、または GONP として指定する必要があります。

**T** T を指定すると DL/I により自動的に操作が再試行される点を除いて、N オプションと同じです。再試行が失敗すると、GG 状況コードがアプリケーション・プログラムに戻されます。T を指定する場合、GOT、GOTH、または GOTP として指定する必要があります。

**E** これを指定すると、オンライン・プログラムがデータベースまたはセグメントを排他使用できるようになります。これは、G、I、D、R、および A と共に使用します。

制約事項: DEDB の場合、PROCOPT=E は許可されません。

**L** データベースのロードのためのロード・オプション (HIDAM および PHIDAM を除く)。

**GS** 昇順でのみセグメントを読み取ります (HSAM の場合のみ)。HSAM データベースの場合に GS を指定すると、DL/I IMS 領域において基本順次アクセス方式 (BSAM) ではなく、待機順次アクセス方式 (QSAM) を使用してセグメントが読み取られます。

**LS** 昇順でのみセグメントをロードします (HIDAM、HDAM、PHIDAM、PHDAM の場合)。このロード・オプションは、HIDAM および PHIDAM の場合は必須です。HIDAM および PHIDAM データベースの場合には LS を指定する必要があるため、ルート・セグメント・シーケンス・フィールドの索引がデータベースのロード時に作成されます。

**H** 特定の PSB を使用するアプリケーション・プログラムについて高速順次処理を指定します。PROCOPT=H の使用には、以下の制限が適用されます。

- DEDB のみに使用できます。
- PCB レベルでは使えますが、セグメント・レベルでは使えません。
- 他の高速機能処理オプションと一緒に使用する必要があります。
- PROCOPT には、H を含めて最大 4 つまでオプションを指定できます。
- BMP の場合にのみ指定できます。
- 1 つの PSB につき、データベース当たり 1 つの PROCOPT=H PCB しか使用できません。HSSP を使用した BMP が、同じ PSB 内の同じデータベースに関して、PROCOPT=H を持つ複数の PCB を使用すると、最初に使用された PCB 以外の PCB を使用するすべてのデータベース呼び出しで FH 状況コードを受け取ります。SETO ステートメントでキーワード NOPROCH を使用すると、この制限を軽減することができます。

- PROCSEQD=Fast\_Path\_index\_dbdname が指定されている場合、PROCOPT=H は使用できません。
- PROCOPT=H を PROCOPT=GO と共に使用することはできません。

H は A、G、I、R および D と共に使用されます。

PROCOPT パラメーターを指定しない場合、デフォルトの PROCOPT=A が使用されます。置換機能と削除機能は、GET 機能も暗黙指定します。

このプログラムがポインターに従って実行されているときに、別のプログラムがそのポインターを更新すると、PROCOPT=GO で、検索モジュール (DFSDLR00) からのユーザー異常終了 (U8XX) が起こる可能性があります。無効な圧縮セグメントが検出されると、VLEXP ルーチンまたは検索モジュールで、異常終了 U0800 または U0852 も起こる可能性があります。ポインターは、挿入機能および削除機能の実行時、および可変長セグメントの置換時に更新されます。このタイプの異常終了の数を減らすため、N または T を指定した PROCOPT= パラメーターをコーディングしてください。

注:

1. PROCOPT が L または LS であり、しかも HISAM または HIDAM データベースを明示的に参照するか、INDEX データベースを暗黙的に参照する PCB が PSB 内にある場合は、同じ PSB 内にある他の PCB が、L または LS 以外の PROCOPT で、リストされたデータベースのいずれかを明示的にせよ、暗黙的にせよ、参照することはできません。PROCOPT が L または LS で、PHIDAM データベースを明示的に参照する PCB が PSB にある場合は、同じ PSB 内の他の PCB が、L または LS の PROCOPT で、同じ PHIDAM データベースを参照することはできません。その PCB 内の SENSEG ステートメントには、INDICES= オペランドを含めることはできません。
2. 複数データ・セット・グループを持つデータベースを参照する PCB に L を指定する場合は、その PCB に、データベース内のデータ・セット・グループごとに最低 1 つの SENSEG ステートメントを含める必要があります。
3. データベースが VSAM を使用しているときに、PROCOPT=L を指定した PCB を使用して最初の ISRT 呼び出しを出す場合には、VSAM データ・セットは空でなければなりません。それが空でないと、オープン・エラーが起こります。

推奨事項: データベースが OSAM を使用している場合は、新しく割り振った空のデータ・セットを使用してください。

データ・セットが空でない場合は、ロードはデータ・セットの先頭から開始され、既存のデータに上書きされることとなります。

4. PCB に 'O' オプションを指定する場合は、SENSEG ステートメントの PROCOPT には、I、R、D、または A を指定してはなりません。
5. オンライン・アプリケーション・プログラムは、PSB 内の PCB で参照される SHSAM または HSAM データベースを常に排他使用します。オンライン環境では、これらの同じ SHSAM または HSAM データベースにアクセスする他のアプリケーション・プログラムを同時にスケジュールすることができません。
6. オンライン・データベース・イメージ・コピー・ユーティリティーがこの PCB を参照する場合には、PROCOPT= L または LS の値は無効です。コピーする

データベースが HIDAM または PHIDAM データベースの索引部分である場合には、PROCOPT=G および PROCOPT=GO のみが有効です。PROCOPT=E を指定すると、オンライン・イメージ・コピー・ユーティリティーはデータベースの排他制御を要求しなくても、排他制御で実行されます。

7. データベース調査ユーティリティー機能がこの PCB を参照する場合は、PROCOPT=G を指定しなければなりません。
8. 連結セグメントの場合、PROCOPT= パラメーターは、連結セグメントの論理子セグメントを制御します。連結セグメントの論理親は、SEGM ステートメントの RULES= パラメーターで規定されます。
9. PROCOPT=E は、PCB 内で指定されたデータベースにのみ適用されます。アプリケーションが明示的に使用しない副次索引を排他使用できるようにするには、副次索引データベース用に PROCOPT=E を指定した別の PCB を追加します。

#### SB=

順次バッファリング (SB) を使用してバッファーに入れる PCB を指定します。これはオプション・パラメーターです。バッチおよび BMP に関して DFSSBUX0 でデフォルト・オプションを SB=COND に変更していない限り、デフォルトは SB=NO です。

#### COND

SB を条件付きで活動化することを指定します。IMS は、DB データ・セットに対するこの PCB の入出力参照パターンの統計をモニターします。IMS は、順次入出力参照パターンと妥当なアクティビティー率を検出する場合、SB をアクティブにし、必要なバッファーを獲得します。

**NO** この DB PCB に SB を使用しないことを指定します。

ヒント: 短時間実行の MPP、高速機能プログラム、および CICS® プログラムでは、SB= キーワードを省略するか、SB=NO を指定する必要があります。

#### KEYLEN=

アプリケーション・プログラムが論理データ構造内で使用する、センシティブ・セグメントの階層パス用の一番長い連結キーをバイトで示した値です。以下の図に示す IMS データベースは、セグメント A から H までとセグメント J を含みます。セグメント A、B、C、D、F、および J のそれぞれのキー・フィールド長は 10 バイトです。セグメント E のキー・フィールド長は 250 バイトです。セグメント G のキー・フィールド長は 40 バイトです。セグメント H のキー・フィールド長は 50 バイトです。次の表に KEYLEN= の指定方法を示します。

表 22. KEYLEN の決定方法

データベース階層のパス	連結キーの長さのパス
A+B+C=	30 バイト
A+B+D=	30 バイト
A+E=	260 バイト
A+F+G+H+J=	120 バイト

KEYLEN=260 バイトを指定します。

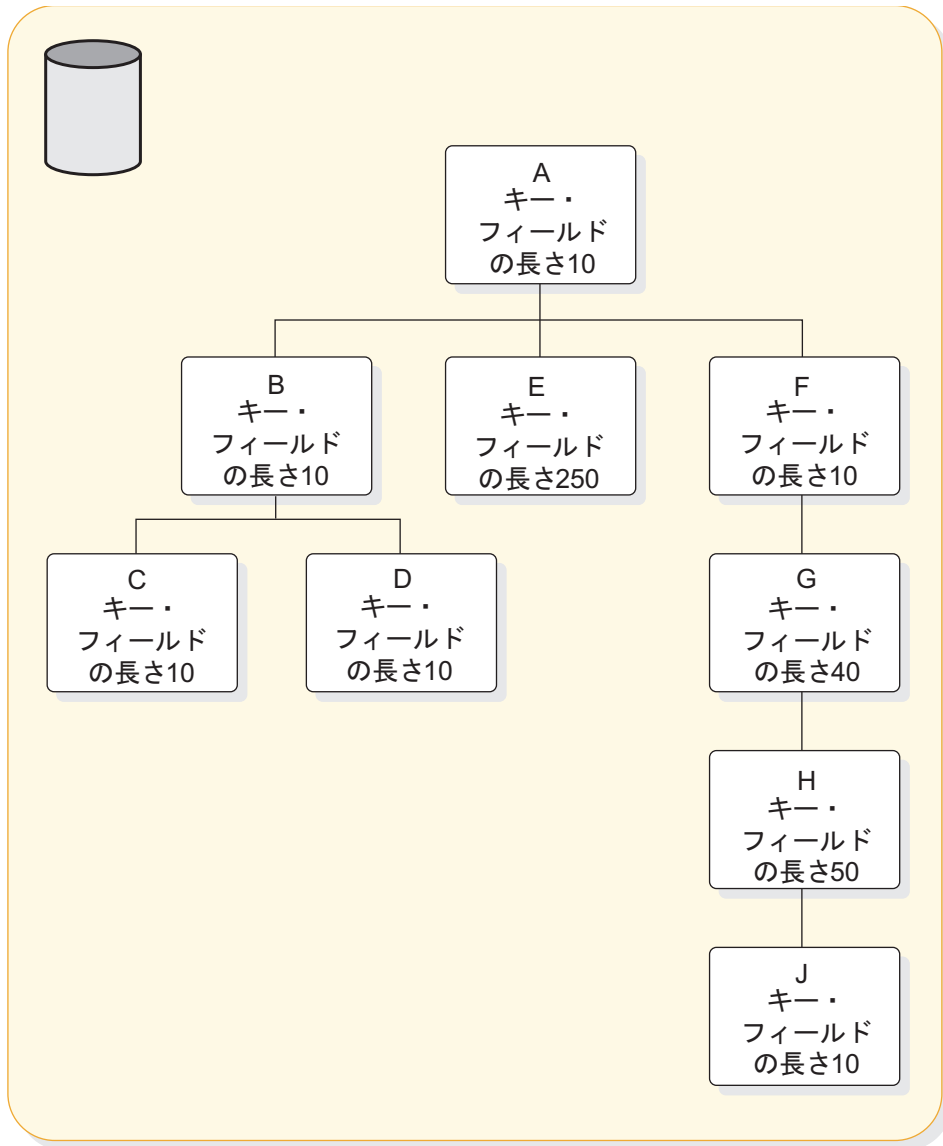


図 13. KEYLEN の定義

端末関連キーのない非端末関連 MSDB では、その値は、DBD 生成のシーケンス・フィールドの BYTES パラメーターの値以上で、しかも 1 から 240 バイトの値でなければなりません。

端末関連 MSDB (キーとして LTERM 名を使用する) のこの値は、8 でなければなりません。

#### POS=

論理データ構造の単一または複数の位置付けを指定します。単一または複数の位置付けは、呼び出し機能にバリエーションを与えます。

単一位置付けと複数位置付けの間のパフォーマンスの変化はほとんどありません。HSAM は複数位置付けをサポートしていません。

POS=SINGLE または S がデフォルトです。

例外: 従属セグメントが 2 つを超える DEDB では、デフォルトは POS=MULTIPLE または M です。

DEDB に PCB ステートメントで POS の値をコーディングしても、従属セグメントの個数に基づいて選択されるデフォルトが変更されることはありません。

#### PROCSEQ=

DBDNAME パラメーターに名前が指定されたデータベースを 2 次処理シーケンスで処理するとき使用する 2 次索引の名前を指定します。このパラメーターはオプションです。これが有効であるのは、このデータベースに 2 次索引が存在している場合のみです。このパラメーターを使用する場合は、後に続く SENSEG ステートメントが、索引付きデータベースの中のセグメント・タイプの 2 次処理シーケンス階層を反映している必要があります。例えば、最初の SENSEG ステートメントでは、PARENT=0 パラメーターで索引先セグメントの名前を指定する必要があります。

*full\_function\_index\_dbname* は副次索引 DBD の名前ではなりません。

2 次処理シーケンスの場合には、処理オプション L および LS は無効です。索引ターゲット・セグメントや、その逆親のいずれかを、挿入したり削除したりすることはできません。ブロックを作成する際に、これらのセグメントの処理オプションに I または D が含まれていると、処理オプションがこの制限を反映するように変更されていることを示す警告メッセージが出されます。

#### PROCSEQD=

1 次 DEDB データベース内のセグメントにアクセスするために使用する、副次索引データベースの名前を指定します。このパラメーターはオプションです。これが有効であるのは、このデータベースに 2 次索引が存在している場合のみです。このパラメーターを使用する場合は、後続の SENSEG ステートメントを 1 次 DEDB の物理シーケンス内でコーディングする必要があります。コーディングする最小の SENSEG ステートメントは、物理ルート・セグメントから索引セグメント (ターゲット・セグメントとも呼びます) に至るパス内にあるものです。

区分副次索引データベースの場合、*Fast\_Path\_index\_dbname* は LCHILD ステートメント内にリストされた最初の *dbname* でなければなりません。

2 次処理シーケンスの場合には、処理オプション L および LS は無効です。索引ターゲット・セグメントや、その逆親のいずれかを、挿入したり削除したりすることはできません。ブロックを作成する際に、これらのセグメントの処理オプションに I または D が含まれていると、処理オプションがこの制限を反映するように変更されていることを示す警告メッセージが出されます。

HISAM または SHISAM 副次索引データベースについてユーザー区画化が要求されている場合、PROCSEQD= パラメーターは、1 次 DEDB データベース DBD 内の LCHILD ステートメントの NAME= パラメーターで定義されている、ユーザー区画グループ内の最初の区画データベースの名前を指定します。XDFLD ステートメントの PSELRTN= パラメーター内のユーザー区画選択出口は、実際に使用する区画データベースを副次索引キー値に基づいて決定します。

#### PSELOPT=

ユーザー区画グループ内のユーザー区画データベースが、そのユーザー区画データベースのデータの終わりに達する前に、SSA 処理を行わない修飾 GN 呼び出しでどのように論理的にグループ化されるかを示します。ユーザー区画データベ

ースは、LCHILD ステートメントの NAME= パラメーターでユーザー区画グループの一部として定義されます。このパラメーターは高速機能副次索引データベースのみに適用されます。

PSELOPT= パラメーターは、XDFLD ステートメントに指定することもできます。PROCSEQD= パラメーターが指定された PCB ステートメントでは PSELOPT= パラメーターのデフォルトはありませんが、XDFLD ステートメントの PSELOPT= パラメーターでは PSELOPT=MULT がデフォルトです。

XDFLD ステートメントと、PROCSEQD オペランドが指定された PCB ステートメントの両方に PSELOPT= パラメーターが指定されている場合、PCB ステートメントの PSELOPT= パラメーターが優先されます。

#### **MULT**

ユーザー・データ区画グループ内の選択されたユーザー区画とその後続のユーザー区画データベースを、1 次 DEDB データベース DBD の LCHILD ステートメントの NAME= パラメーターで物理的に定義されているとおりに示します。PSELOPT=MULT は、XDFLD ステートメントの PSELOPT= パラメーターのデフォルトです。

#### **SNGL**

選択されたユーザー区画データベースのみが使用されることを示します。

#### **ACCESS=**

副次索引データベースがその 1 次 DEDB データベースへのアクセスに使用されるか、または副次索引データベースが別個の論理データベースとして処理されるかを指定します。

**DB** 1 次 DEDB データベースがその副次索引シーケンスを使用してアクセスされることを指定します。ACCESS=DB がデフォルトです。

#### **(INDEX,VSAM) | (INDEX,SHISAM)**

ユーザー区画グループ内の 1 つ以上のユーザー区画データベースが別個の論理データベースとしてアクセスされることを指定します。

HISAM 副次索引データベースの場合は、高速機能副次索引データベースの DBD ステートメントに ACCESS=(INDEX,VSAM) を指定します。

SHISAM 副次索引データベースの場合は、高速機能副次索引データベースの DBD ステートメントに ACCESS=(INDEX,SHISAM) を指定します。

#### **VIEW=MSDB**

MSDB コミット・ビューの指定に使用します。既存のアプリケーションでは、MSDB コミット・ビューまたはデフォルトの DEDB コミット・ビューを使用することができます。DEDB に関して MSDB コミット・ビューを使用するには、ステートメントで VIEW=MSDB を指定します。VIEW=MSDB を指定しなかった場合は、DEDB では DEDB コミット・ビューを使用します。MSDB を DEDB にマイグレーションする場合は、既存のアプリケーション・プログラムのいずれにも変更を加える必要はありません。



VIEW=MSDB を指定する PCB を使用して REPL 呼び出しを発行する場合は、セグメントにキーが必要です。コマンド・コード 'D' が指定されている場合は、パス内のいずれのセグメントにもこの条件が当てはまります。そうでない場合は、状況 AM が戻されます。

#### **LIST=**

入り口でアプリケーション・プログラムに渡される PCB リストに、名前を指定した PCB を含めるかどうかを指定します。指定した PCB を PCB リストに含めるときは、YES を指定します。指定した PCB を PCB リストから除外するとき、NO を指定します。YES がデフォルトです。

PCB リストから PCB を除外するときには、label または PCBNAME= パラメーターで PCB に名前を指定する必要があります。アプリケーション・プログラムが PCB のアドレスを必要としない場合には、LIST=NO を指定します。

#### **EXTERNALNAME=**

PCB ラベルまたは PCBNAME= パラメーターのオプションの別名。Java アプリケーション・プログラムは、PCB を参照する場合にこの外部名を使用します。

外部名は 1 から 128 文字の大文字の英数字ストリングとして指定します。外部名には下線文字を含めることができます。

外部名は PSB 内で固有でなければなりません。

EXTERNALNAME を指定しない場合、デフォルトの外部名は PCB ラベルか PCBNAME (いずれか指定されている方) になります。

PCB ラベルも PCBNAME も指定されていない場合、EXTERNALNAME はデフォルトで空白になります。

制約事項: 外部名は予約済み SQL キーワードにすることも、DFS で開始することもできません。

EXTERNALNAME パラメーターが指定されておらず、NAME パラメーターに予約済み SQL キーワードが指定されている場合、EXTERNALNAME は NAME 値に「\_SCH」を付加したうえで、その NAME 値をデフォルトの外部名として受け入れます。

#### **REMARKS=**

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### **CORRECT**


```
REMARKS='These remarks apply to the XYZ application'
```

## INCORRECT

REMARKS='These remarks apply to the 'XYZ' application'

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

関連資料:

 IMS JDBC ドライバーにより制限されるポータブル SQL キーワード (アプリケーション・プログラミング)

## 高速機能データベースの処理オプション

高速機能データベースの処理オプションは、高速機能データベースのタイプ (非端末関連または固定端末関連の MSDB、動的端末関連 MSDB、または DEDB) によって異なります。

非端末関連または固定端末関連 MSDB では、G または R の処理オプションのみ有効です。

- G** GET 機能。
- R** 置換機能。G も含みます。

動的端末関連 MSDB では、G、I、R、D、A の処理オプションか、G、I、R、および D を任意に組み合わせたものが有効です。

- G** GET 機能。
- I** 挿入機能。
- R** 置換機能。G も含みます。
- D** 削除機能。G も含みます。
- A** 全機能。G、I、R、および D の機能を含みます。

DEDB では、G、I、R、D、A、P、N、T、O、および H の処理オプションが有効です。

- G** GET 機能。
- I** 挿入機能。
- R** 置換機能。G も含みます。
- D** 削除機能。G も含みます。
- A** 全機能。G、I、R、および D を含みます。
- P** 位置機能。DEDB を処理する際にコマンド・コード D を使用する場合は必須ではありません。これは、バッチ・メッセージ・プログラム (BMP) の場合にのみ有効です。別のタイプの領域 (IFP 領域など) にこのオプションを指定すると、無視されます。このオプションでは、ルート・セグメントで G(H)U、G(H)N、または ISRT の実行時に UOW 境界が交差していると、GC 状況コードが戻されます。また、データベース位置付けは、有効な SYNC 呼び出し後も保持され、GC 状況コードを受け取った直後に SYNC が出されると、ブランクの状況コードが戻されます。SYNC 処理が失敗し

た場合、または ROLB 呼び出しの場合、位置は前回の有効な同期点に設定されます。あるいは、有効な同期点がなければ、データベースの先頭に設定されます。GC 状況が前に起きていない場合の SYNC または ROLB 呼び出しでも、位置はデータベースの先頭に設定されます。

DEDB の呼び出しで D コマンド・コードを使用する場合は、プログラムの PCB 内に P 処理オプションを指定する必要はありません。

**N** 読み取り専用アプリケーション・プログラムが起こす異常終了の数を減らします。読み取り専用アプリケーション・プログラムは、別のアプリケーション・プログラムが更新しているデータを参照できます。このことが行われると、そのデータを指す無効なポインターが存在する場合があります。無効なポインターが検出されると、読み取り専用アプリケーション・プログラムは異常終了します。N を指定すると、この状態を回避できます。代わりに、GG 状況コードがプログラムに戻されます。プログラムは、処理を終了するか、別のセグメントを読み取って処理を続行するか、あるいは別のパスを使用してデータにアクセスすることができます。N を指定する場合は、GON、GONH、または GONP として指定する必要があります。

**O** 読取専用。可用性を調べるためにエンキューしてはなりません。DEDB の場合に PROCOPT=GO、GON、または GOT を選択すると、健全性のない読み取りが有効になります。検索データの健全性を保持するのに、ロッキング・メカニズムは使用されません。O は、GO、GON、GOT のいずれかとして指定する必要があり、H と一緒に使用することはできません。

O を指定すると、IMS は制御インターバル (CI) を 1 回だけ読み取り、その CI のコピーを同じ同期点インターバルでの次の 50 回の参照に対して使用します。参照が 50 回行われると、参照カウンターがリセットされ、CI が再度読み取られます。この動作によって、別のスレッドで更新された整合性のないデータへのアクセスが原因で発生するセグメント・チェーン・ループを防止しています。

このプログラムがポインターに従って実行しているときに、別のプログラムがそのポインターを更新すると、PROCOPT=GO で、ユーザー異常終了 (U1026) が起こる可能性があります。異常終了のもう 1 つの例として、別のプログラムが長さを変更したときに、このプログラムが移動したセグメントの再読み取りを行う場合があります。次の例では、異常終了 U1026 が発生したり、古いデータが検索される場合を例示していますので、参考にご覧ください。

例 1 : 1 つの領域が、更新 PCB と PROCOPT=GO PCB の両方を使用して、同じセグメントを更新して読み取る場合は、以下の手順で行うと、PROCOPT=GO PCB (MLTE) の制御ブロックへのポインター・エラーは発生しません。更新 PCB (PCBA) を呼び出してから、読み取り PCB (PCBGO) を呼び出すようにします。

1. 領域 1 の PCBGO が CI を読み取り、このセグメントの位置を MLTE に設定します。バッファ内のデータが EPSTGOBF にリンクされません。
2. 領域 1 はセグメントの更新を行うために呼び出しを行います。領域 1 の PCBA はバッファから EPSTGOBF をスチールします。領域 1 の PCBA はセグメントの古い位置を保管し、セグメントを更新します。領

域 1 は、セグメントが移動していても、PCBGO MLTE を更新します。これは、GO MLTE のセグメントの位置が、保管した古いセグメントの位置と一致するからです。

3. 領域 1 の PCBGO はもう一度セグメントを参照して、更新済みセグメントをリトリートします。

例 2 : 2 つの領域が、同一のセグメントを更新し、更新 PCB と PROCOPT=GO PCB の両方を使用する場合、以下の手順で行うと、PROCOPT=GO PCB (MLTE) の制御ブロックへのポインター・エラーは発生しません。しかし、PROCOPT=GO PCB は、他の領域から更新したセグメントにアクセスすることはできません。

1. 領域 1 の PCBGO が CI を読み取り、このセグメントの位置を MLTE に設定します。バッファは EPSTGOBF にリンクされます。
2. 領域 2 の PCBA はロック付きで CI を読み取り、長さを変更したセグメントと置き換えます。セグメントの位置が変わり、領域 1 の PCBGO MLTE に設定した位置にある更新された CI の FSE になります。領域 1 は EPSTGOBF にリンクしたバッファに古いデータをまだ持っています。
3. 領域 2 の変更によってバッファは更新されていないため、領域 1 の PCBGO はもう一度セグメントを参照して、古いセグメントをリトリートします。

例 3 : 2 つの領域が、同じセグメントを更新し、更新 PCB と PROCOPT=GO PCB の両方を使用する場合、以下の手順に従うと、PROCOPT=GO PCB (MLTE) の制御ブロックのポインター・エラーは発生しません。しかし、PROCOPT=GO PCB は、自分の領域から更新したセグメントにアクセスすることはできません。

1. 領域 1 の PCBGO が CI を読み取り、このセグメントの位置を MLTE に設定します。バッファは EPSTGOBF にリンクされます。
2. 領域 2 の PCBA はロック付きで CI を読み取り、長さを変更したセグメントと置き換えます。セグメントの位置が変わり、領域 1 の PCBGO MLTE に設定した位置にある更新された CI の FSE になります。領域 1 は EPSTGOBF にリンクしたバッファに古いデータをまだ持っています。
3. 領域 1 はセグメントの更新を行うために呼び出しを行います。領域 1 は領域 2 のロックが解除されるのを待ちます。更新されたセグメントは別のブロック上に存在しているため、領域 1 は EPSTGOBF に複写バッファを検出できず、古いバッファが EPSTGOBF にリンクされたままになっています。領域 1 は、そのバッファにある更新された CI を読み取ります。領域 1 の PCBA はその場所でセグメントを更新します。セグメントが移動していても、領域 1 は PCBGO MLTE を更新しません。これは、MLTE 内の位置がセグメントの位置と一致しなくなっているためです。このように、2 つの複写バッファが存在するようになります。一方には、EPSTGOBF にリンクした古いデータが入っており、もう一方には、EPSTXCOC にリンクした更新済み情報が入っています。
4. 領域 1 の PCBGO はそのセグメントを参照して、古いデータをリトリートします。

例 4 : 2 つの領域が、同一のセグメントを更新し、更新 PCB と PROCOPT=GO PCB の両方を使用する場合は、以下の手順に従うと、PROCOPT=GO PCB (MLTE) の制御ブロックへのポインター・エラーが発生します。

1. 領域 1 の PCBGO が CI を読み取り、このセグメントの位置を MLTE に設定します。バッファは EPSTGOBF にリンクされます。
2. 領域 2 の PCBA はロック付きで CI を読み取り、長さを変更したセグメントと置き換えます。同じブロック内でセグメントの位置が変更され、領域 1 の PCBGO MLTE に設定された位置に更新済み CI の FSE を作成します。領域 1 は EPSTGOBF にリンクされたバッファに古いデータをまだ持っています。
3. 領域 1 はセグメントの更新を行うために呼び出しを行います。領域 1 は領域 2 のロックが解除されるのを待ちます。領域 1 の PCBA は EPSTGOBF から離してバッファをスチールし、更新済み CI を読み取り、これを領域 1 のバッファに移動します。領域 1 の PCBA はその場所でセグメントを更新します。セグメントが移動していても、領域 1 は PCBGO MLTE を更新しません。これは、MLTE 内の位置がセグメントの位置と一致なくなっているためです。
4. 領域 1 の PCBGO は、セグメントをもう一度参照し、異常終了 U1026 を受け取ります。これは、以前セグメントがあった (MLTE の位置) 場所に FSE が存在するようになったためです。

このタイプの異常終了の数を減らすため、N または T を指定した PROCOPT= パラメーターをコーディングしてください。

**T** 機能は、N オプションとまったく同じです。T を指定する場合、GOT、GOTH、または GOTP として指定する必要があります。

**H** HSSP。G と P を含みます。

バッチ指向 BMP のような、入力 LTERM がないプログラムから端末関連の動的 MSDB への DLET または ISRT 呼び出しでは、指定した処理オプションに関係なく、状況コードが AM になります。

置換機能は、GET 機能も含みます。参照されたセグメントがルート・セグメントまたは直接従属セグメントである場合、A は G、I、R、および D を暗黙指定します。順次従属セグメントの場合には、処理オプションの G、I、および GI のみ有効です。

処理オプションの P が有効であるのは、IMS バッチ・メッセージ・プログラムが使用するルート・セグメントに指定された場合のみです。処理オプションの P が IFP 領域など他のタイプの領域に指定された場合は、無視されます。このオプションでは、ルート・セグメントで G(H)U、G(H)N、または ISRT の実行時に UOW 境界が交差していると、GC 状況コードが戻されます。また、データベース位置付けは、有効な SYNC 呼び出し後も保持され、GC 状況コードを受け取った直後に SYNC が出されると、ブランクの状況コードが戻されます。SYNC 処理が失敗した場合、または ROLB 呼び出しの場合、位置は前回の有効な同期点に設定されません。あるいは、有効な同期点がなければ、データベースの先頭に設定されます。GC 状況が前に起きていない場合の SYNC または ROLB 呼び出しでも、位置はデータベースの先頭に設定されます。

DEDB の呼び出しで D コマンド・コードを使用する場合は、プログラムの PCB 内に P 処理オプションを指定する必要はありません。

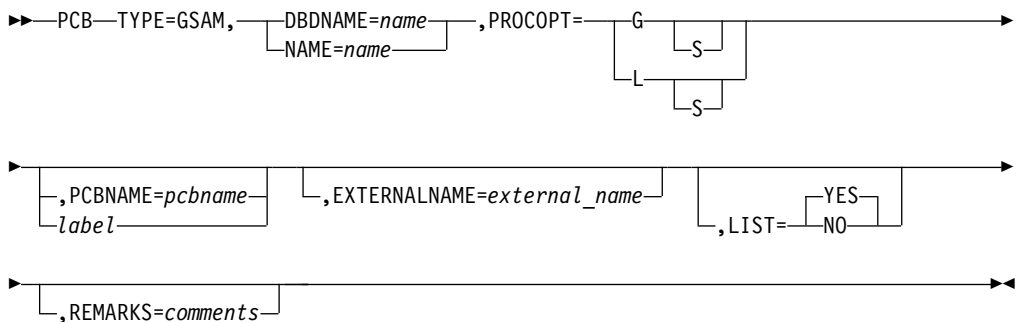
PROCOPT H は、O と共に使用することはできません。

無効な処理オプションを指定すると、PSBGEN はそれらのオプションを受け入れませんが、アプリケーション制御ブロック保守ユーティリティーは失敗します。エラーは PSBGEN では現れませんが、ACBGEN で現れます。

## GSAM PCB ステートメント

GSAM PCB ステートメントは、割り振られる GSAM データベース PCB の名前を指定します。

以下の図は、GSAM データベースの PCB ステートメントの形式を示しています。



### TYPE=GSAM

従属領域に割り振られて処理される GSAM データベース PCB のすべてに必須のキーワード・パラメーターです。

### DBDNAME= または NAME=

データ・セット記述の 1 次ソースとして使用する GSAM DBD を指定する名前に必須のキーワード・パラメーターです。SENSESEG ステートメントを、この PCB ステートメントに続けてはなりません。

### PROCOPT=

この PCB で宣言された、関連するアプリケーション・プログラムで使用できるデータ・セットの処理オプションの必要パラメーターです。このパラメーターには次の文字を指定してください。

- G** GET 機能。
- L** LOAD 機能。
- S** 大規模順次アクティビティ。GSAM 複数バッファリング・オプション (BUFFIO) を使用します。

PSB 生成で TYPE=TP または DB の PCB ステートメントがあれば、その後ろにこの GSAM PCB ステートメントを続けなければなりません。次の規則があります。

### TP PCB

第 1

## DB PCB (複数の場合もある)

第 2 版

## GSAM PCB

最後

### PCBNAME=

PCB の名前を指定します。PCB の名前は、標準命名規則に従った 8 バイトの英数字の文字ストリングでなければなりません。PCB の名前は、PSB 内で固有でなければなりません。

例外: PCB ステートメントに *label* を指定する場合は、このパラメーターを指定しないでください。

### *label*

アセンブラ言語ステートメントで有効な、長さが 1 から 8 文字までの英数字ラベルを指定します。PSB 内の PCB ステートメントのラベルは、固有でなければなりません。

例外: PCBNAME= を使用する場合は、このパラメーターを指定しないでください。

### EXTERNALNAME=

PCB ラベルまたは PCBNAME= パラメーターのオプションの別名。Java アプリケーション・プログラムは、PCB を参照する場合にこの外部名を使用します。

外部名は 1 から 128 文字の大文字の英数字ストリングとして指定します。外部名には下線文字を含めることができます。

外部名は PSB 内で固有でなければなりません。

EXTERNALNAME を指定しない場合、デフォルトの外部名は PCB ラベルか PCBNAME (いずれか指定されている方) になります。

PCB ラベルも PCBNAME も指定されていない場合、EXTERNALNAME はデフォルトでブランクになります。

制約事項: 外部名は予約済み SQL キーワードにすることも、DFS で開始することもできません。

EXTERNALNAME パラメーターが指定されておらず、NAME パラメーターに予約済み SQL キーワードが指定されている場合、EXTERNALNAME は NAME 値に「\_SCH」を付加したうえで、その NAME 値をデフォルトの外部名として受け入れます。

### LIST=

入り口でアプリケーション・プログラムに渡される PCB リストに、名前を指定した PCB を含めるかどうかを指定します。指定した PCB を PCB リストに含めるときは、YES を指定します。指定した PCB を PCB リストから除外するとき、NO を指定します。YES はデフォルトです。

PCB リストから PCB を除外するとき、PCBNAME= パラメーターで PCB に名前を指定する必要があります。アプリケーション・プログラムが PCB のアドレスを必要としない場合には、LIST=NO を指定します。

## REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

### CORRECT

```
REMARKS='These remarks apply to the XYZ application'
```

### INCORRECT

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## SENSEG ステートメント

階層的に関連したデータ・セグメントのセットを定義するには、SENSEG ステートメントをデータベース PCB ステートメントと共に使用します。

このセットは、この PCB を介してプログラムがセンシティブなセグメントを表します。このセグメント・セットは、物理的に 1 つのデータベース内に入れることも、複数の物理データベースから得ることもできます。1 つ以上の SENSEG PCB ステートメントを含めることができます。各 SENSEG ステートメントは、それが関連している PCB ステートメントの直後に続ける必要があります。アプリケーション・プログラムがセンシティブであるセグメントごとに 1 つの SENSEG ステートメントが必要です。必要なセグメントへの階層パス内のすべてのセグメントを指定する必要があります。最大限 30,000 の SENSEG ステートメントが単一の PSB 生成で定義できます。30,000 もの SENSEG ステートメント数は、非実用的です。SENSEG ステートメント数がこれほど多くなると、通常使用可能な容量のストレージには収まらないからです。

PCB ステートメントの後に並べる SENSEG ステートメントの順序により、セグメントの論理アクセス順序が決まります。HSAM または HISAM データベースを使用する場合には、SENSEG ステートメント順序は、PROCSEQ パラメーターが PCB ステートメント内で使用されていない限り、DBDGEN に定義されたセグメントの物理順序に従う必要があります。

PROCSEQ パラメーターが PCB ステートメントで使用されている場合は、SENSEG ステートメント・シーケンスに、PROCSEQ パラメーターで指定されている 2 次処理シーケンスが反映されます。HDAM、HIDAM、PHDAM、および PHIDAM データベースの場合には、同じレベルのセグメントの SENSEG ステート



メントが DBD と同じ順序である必要はありません。親セグメントが階層ポインター設定を使用しない従属セグメントの順序は、物理順序とは異なることがあります。

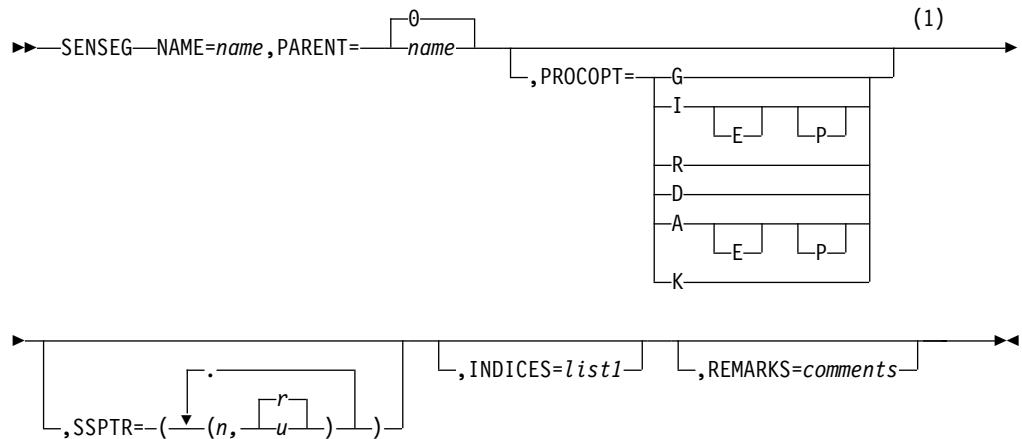
PCB ステートメントに PROCSEQD パラメーターを指定する場合、1 次 DEDB データベースの物理構造順序を使用して SENSEG ステートメントを指定する必要があります。PROCSEQD オペランドが指定されるすべての PCB は、それぞれ 2 つの PCB としてカウントされ、PSBGEN ごとの PCB の制限である 2500 に近づいていきます。

ターゲット・セグメントがルート・セグメントである場合、副次索引を使用せずに DEDB データベースに指定するときと同じ方法で SENSEG セグメントを指定します。ルート・セグメントの下のすべてのセグメントは、1 次 DEDB データベースの物理構造全体で副次索引を使用してアクセス可能です。

ターゲット・セグメントがルート・セグメントでない場合、ルート・セグメントからターゲット・セグメントに至る物理パスに沿って、ターゲット・セグメントの直接的な親であるすべてのセグメントに対して SENSEG ステートメントを指定する必要があります。ルート・セグメントからの物理パスに沿ってターゲット・セグメントの直接的な親であるセグメントと、ターゲット・セグメントのすべての子セグメントのみが、1 次 DEDB データベースが副次索引を使用してアクセスされるときに 1 次 DEDB データベースの物理構造内でアクセス可能です。

DEDB データベースの場合、PCB ステートメントに PROCSEQD パラメーターが指定されていて、ターゲット・セグメントがルート・セグメントでない場合、物理ルート・セグメントからターゲット・セグメントへ直接つながる線にあるすべての SENSEG ステートメントをコーディングする必要があります。SENSEG ステートメントの順序も、PROCSEQD GN 処理が、ターゲット・セグメントから始まりルートへと上る論理順序でナビゲートされている場合でも、物理順序でなければなりません。

SENSEG ステートメントの形式は次のとおりです。



注:

- 1 これらは任意の組み合わせで選択できます。G、I、R、および D をすべて選択する場合には、代わりに A を使用してください (A =G、I、R、および D を組み合わせたもの)。

**NAME=**

DBD 生成時に SEGM ステートメントで定義したセグメント・タイプの名前。  
このフィールドは、1 から 8 文字の英数字です。

**PARENT=**

このセグメントの親のセグメント・タイプ名です。

要件: このパラメーターは、すべての従属セグメントに必須です。

このフィールドは、1 から 8 文字までの英数字または 0 です。この SENSEG ステートメントがルート・セグメント・タイプをセンス可能 (センシティブ) と定義している場合には、このパラメーターはゼロでなければなりません。

PARENT=0 がデフォルトです。

**PROCOPT=**

関連付けられているアプリケーション・プログラムがこのセンシティブ・セグメントを使用する場合の有効な処理オプションを表します。このパラメーターは、PCB ステートメントの PROCOPT= パラメーターと同じ意味を持ちます。このパラメーターの有効オプションの他に、PCB ステートメントでは使用できず、SENSEG ステートメントで使用できるオプションが 1 つあります。PROCOPT が K である場合は、キー・センシティブィーのみを示します。SSA を持たない GN 呼び出しは、データ・センシティブ・セグメントのみにアクセスできます。キー・センシティブ・セグメントが SSA での検索用に指定されている場合は、そのセグメントはユーザーの入出力域には移されません。キーは、PCB のキー・フィールドバック域の該当するオフセット位置に置かれます。この PROCOPT= パラメーターを指定しない場合、PCB PROCOPT パラメーターがデフォルトとして使用されます。PCB ステートメントと SENSEG ステートメントに指定した処理オプションが異なる場合、それらのオプションに互換性があれば、SENSEG PROCOPT が PCB PROCOPT をオーバーライドします。先行する PCB ステートメントに PROCOPT= L または LS を指定した場合には、このパラメーターを省略する必要があります。

PROCOPT= L または LS を指定する場合は、仮想論理子のセグメント・タイプには SENSEG ステートメントを指定してはなりません。置換機能と削除機能は、GET 機能も暗黙指定します。

セグメントに PROCOPT=K の指定があれば、非修飾 GN (GET NEXT) 呼び出しは、PROCOPT が K 以外のセンシティブ・セグメントにスキップします。

SENSEG PROCOPT は PCB PROCOPT を変更します。PROCOPT=E が PCB の中で指定されている場合は、SENSEG PROCOPT にも E を指定する必要があります (その SENSEG 専用にスケジュールする場合)。

SENSEG ステートメントの中で処理オプション N または T を指定しても無効です。これらの処理オプションは PCB ステートメントの中でしか指定できません。

DEDB 順次従属セグメントの処理オプションは、G または I のいずれかでなければなりません。これらの値のいずれかを PCB ステートメントで指定しない場合は、SENSEG PCB ステートメントに PROCOPT=G または I を指定する必要があります。

連結セグメントの場合、PROCOPT= パラメーターは、連結セグメントの論理子セグメントを制御します。連結セグメントの論理親は、SEGM PCB ステートメントの RULES= パラメーターで制御されます。

#### SSPTR=

サブセット・ポインタの数とサブセット・ポインタのセンシティブティーを指定します。最大 8 個のサブセット・ポインタを定義できます。サブセット・ポインタの数 (第 1 パラメーター) は、1 から 8 でなければなりません。サブセット・ポインタのセンシティブティー (第 2 パラメーター) は、R (読み取りセンシティブ) または U (更新) でなければなりません。第 1 パラメーターと第 2 パラメーターを指定しない場合、ポインタにはセンシティブティーがありません。n のみを指定した場合は、ポインタは読み取りセンシティブです。SSPTR=R はデフォルトです。

処理オプションが A、R、I、または D でない場合は、U (更新センシティブティー) を使用できません。

#### INDICES=

どの 2 次索引に検索フィールドを含めるかを指定します。この検索フィールドは索引先セグメント・タイプの SSA を修飾するために使用されます。

INDICES= パラメーターは、索引先セグメント・タイプにのみ指定できます。これを指定すると、索引先セグメント・タイプの呼び出しの SSA を、指定された各 2 次索引に含まれている索引先セグメント・タイプの検索フィールド上で修飾できます。

#### 制約事項:

- 索引先セグメント・タイプの呼び出しの SSA は、副次索引の検索フィールドでは修飾できません。ただし、索引セグメント・タイプの場合の SENSEG ステートメントの INDICES= パラメーターか、または PCB ステートメントの PROCSEQ= パラメーターに、その副次索引が指定されている場合は別です。
- INDICES= パラメーターは、高速機能副次索引ではサポートされません。

*list1* には、最大 32 個の副次索引の DBD 名を指定できます。複数の名前を指定する場合には、それらの名前をコンマで区切って、そのリストを括弧で囲む必要があります。

#### REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力される

と、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

**CORRECT**

REMARKS='These remarks apply to the XYZ application'

**INCORRECT**

REMARKS='These remarks apply to the 'XYZ' application'

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパサンド (&)。

以下の図は、セグメント A から F を含む、セグメント定義のデータ構造です。

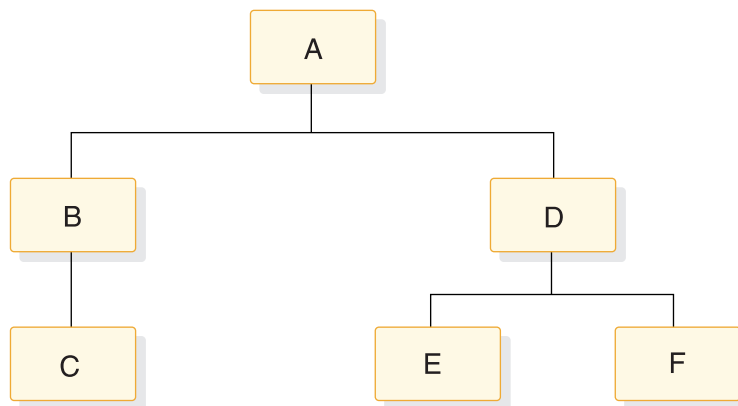


図 14. セグメント定義のデータ構造

これらのセグメントのすべては、1 つの DBD 内で定義されます。先行する PCB ステートメントで PROCOPT=L、LS、I、または D を指定した場合は、SENSEG PCB ステートメントで INDICES= を指定してはなりません。

このデータ構造を表す PCB と SENSEG ステートメント全体は、次のようになります。

Col. 10	Col. 16	72.
PCB	TYPE=DB,DBDNAME=DATABASE, PROCOPT=A,KEYLEN=22	X
SENSEG	NAME=A,PARENT=0,PROCOPT=G	
SENSEG	NAME=B,PARENT=A,PROCOPT=G	
SENSEG	NAME=C,PARENT=B,PROCOPT=I	
SENSEG	NAME=D,PARENT=A,PROCOPT=A	
SENSEG	NAME=E,PARENT=D,PROCOPT=G	
SENSEG	NAME=F,PARENT=D,PROCOPT=A	

**SENFLD ステートメント**

SENFLD ステートメントを SENSEG ステートメントと一緒に使用すれば、アプリケーション・プログラム・センシティブな、セグメント内のフィールドを示すことができます。



### REPLACE= または REPL=

このフィールドを置き換え呼び出しで変更するかどうかを指定します。NO または N を指定できます。指定しない場合は、REPLACE=YES (または Y) がデフォルトになります。

### REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### CORRECT

```
REMARKS='These remarks apply to the XYZ application'
```

#### INCORRECT

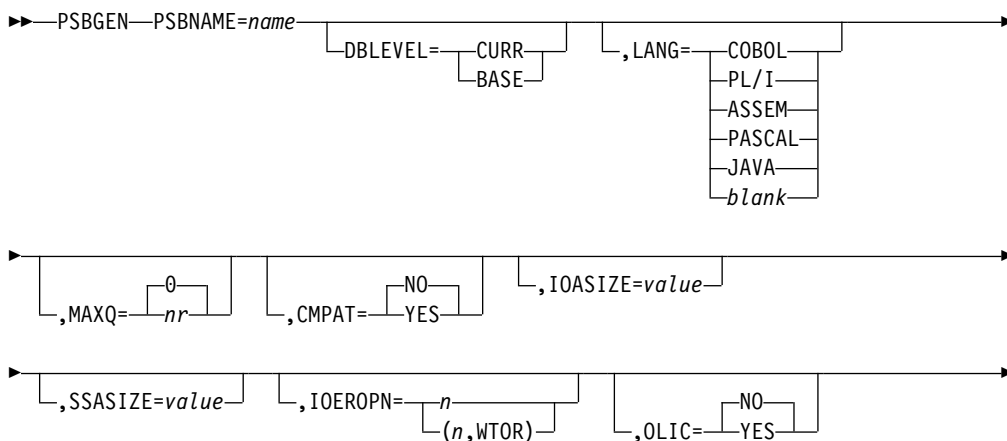
```
REMARKS='These remarks apply to the 'XYZ' application'
```

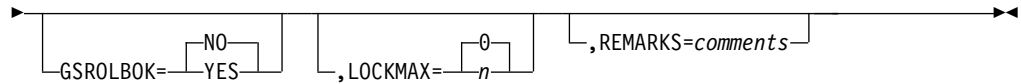
- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

## PSBGEN ステートメント

PSBGEN ステートメントは、アプリケーション・プログラムの特性を指定します。

以下の構文図は、PSBGEN ステートメントの形式を示しています。





#### PSBNAME=

この PSB のパラメーター (英数字名) を指定します。PSBNAME の名前は、標準命名規則に従った 8 バイトの英数字の文字ストリングでなければなりません。この名前は、ライブラリー IMS.PSBLIB 内の PSB のロード・モジュール名になります。プログラムがメッセージ処理領域内で実行される場合には、この名前は、IMS.PGMLIB という名前のプログラム・ライブラリー内のプログラム・ロード・モジュール名と同じでなければなりません。名前の中で特殊文字は使用できません。

DBD に、既存の PSB と同じ名前を付けてはなりません。既存名を使用すると、予測不能な結果が起こることがあります。ACB 生成時にエラーが起こります。

#### DBLEVEL=

データベースのバージョン管理が使用可能な場合、特定のデータベース・バージョンを要求しないアプリケーション・プログラムに対してデータを返すのに使用する DBD のバージョンを指定します。この PSB を使用するすべてのアプリケーション・プログラムでは、ここで指定する値が、IMS.PROCLIB データ・セットの DFSDExxx メンバーで指定される DBLEVEL のシステム・デフォルトをオーバーライドします。

DBLEVEL の有効値は以下のとおりです。

#### CURR

DBLEVEL=CURR が指定されていて、アプリケーション・プログラムがデータベースにアクセスする際の呼び出しでバージョン番号を指定しない場合、IMS は、ACB ライブラリー内のアクティブ・メンバーによって定義されたデータベース構造を使用してアプリケーション・プログラムにデータを返します。通常、アクティブ ACB メンバーが、物理データベースの実際の現行構造を定義します。

#### BASE

DBLEVEL=BASE が指定されていて、アプリケーション・プログラムがデータベースのバージョン番号を指定しない場合、IMS は、DBVER=0 を含む、IMS カタログ内の DBD レコードによって定義されたデータベース構造を使用してアプリケーション・プログラムにデータを返します。

DBVER=0 は、DBD の後続バージョンで DBVER 値が明示的に指定される前の DBD のデフォルト・バージョンです。IMS カタログ内にバージョン 0 の DBD インスタンスが複数存在する場合、IMS は最新のタイム・スタンプを持つインスタンスを使用します。

個々のアプリケーション・プログラムで、DBLEVEL 値で指定した番号以外の DBD バージョン番号を要求することができます。これを行うには、PCB ステートメントの DBVER キーワードか、または INIT VERSION 呼び出しで DBD バージョン番号を指定します。

データベースのバージョン管理が使用不可の場合、DBLEVEL パラメーターは無視されます。

**LANG=**

メッセージ処理プログラムまたはバッチ処理プログラムの作成に使用するコンパイラ言語を示すオプションのキーワード。OLICYES を指定した場合、LANGPLI は無効です。アプリケーション・プログラムが C 言語で作成されている場合には、LANGASSEM を指定します。

CICS および z/OS 用の Language Environment (言語環境プログラム) は LANGPASCAL をサポートしていません。

アプリケーションが JMP 領域内で IMS 用の Java クラス・ライブラリーを使用している場合は、LANGJAVA を指定する必要があります。

PLICALLA エントリー・ポイントを使用して互換モードで実行される IMS PL/I アプリケーションを使用している場合は、LANGPLI を指定する必要があります。

**MAXQ=**

同期点と同期点の間に出すことができる、Qx コマンド・コードによるデータベース呼び出しの最大数。最大数は 32,767 です。デフォルトはゼロです。

**CMPAT=**

BMP または MSG と、バッチ DL/I パラメーター・リストとの間の互換性を提供します。YES に設定される場合、この PSB は、どのように使用されているかにかかわらず、常に入出力 PCB があるように取り扱われます。NO に設定される場合、BMP または MSG 領域についてのみ入出力 PCB が PSB に追加されます。デフォルトは NO です。

**IOASIZE=**

アプリケーション・プログラムで使用する最大入出力域のサイズ (バイト数) を指定します。このサイズ指定を使用して、このアプリケーション・プログラムのスケジューリング時に、ユーザー入出力域のデータに関する制御領域のコピーを保持するために PSB プールに予約する主記憶域の量が決められます。この値を指定しない場合、IMS は最大のデフォルト入出力域サイズを計算します。デフォルトのサイズは、可能な限り最長のパス CALL 内にあるすべてのセンシティブ・セグメントの合計の長さです。(アプリケーション・プログラムが、セグメント中のすべてのフィールドにセンシティブでなくても、セグメントの全長を使用する必要があります。) 指定する値はバイト単位で、最大が 256000 です。ただし、1 つのパス CALL でアプリケーションに戻される全連結セグメントの合計の長さは、65535 バイトを超えてはなりません。

PSB がフィールド・センシティブ・セグメントを含んでいる場合に、IOASIZE が指定されていると、指定値は、ACBGEN ユーティリティーが計算した IOASIZE よりも大きい場合にのみ使用されます。使用される IOASIZE の値は、指示されます。このプール所要量の主要なコンポーネントは、IOASIZE と SSASIZE です。

この PSB で STAT 呼び出しまたはテスト・プログラム (DFSDDLTO) を使用する場合は、IOASIZE を 600 バイトより大きくしなければなりません。

この PSB で CMD または GCMD 呼び出し (自動化操作プログラム・インターフェース・アプリケーション・プログラムから) を使用する場合には、IOASIZE は 132 バイト以上にする必要があります。



拡張チェックポイント・リスタートを使用する場合は、IOASIZE には、下記の大きい方の値以上の値を指定しなければなりません。

- 以前のチェックポイントがある DL/I データベース (この PSB にチェックポイントがある場合) の位置変更を行っているときに、再始動時に出される GU 呼び出しからのデータを受け取るために必要な入出力域。
- 以前のチェックポイントがある GSAM データ・セットで使用される最大 LRECL。

XRST CALL の第 3 パラメーター (I/O AREA LEN) が指す値、またはこのパラメーターの値のいずれか大きい方の値が使用されます。

#### **SSASIZE=**

アプリケーション・プログラムで使用するすべての SSA の最大合計長を指定します。IMS は、このサイズ指定を使用して、このアプリケーション・プログラムの実行中に、ユーザーの SSA スtringのコピーを保持するために PSB 作業プールに予約する主記憶域量を決めます。この値を指定しなければ、ACB ユーティリティ・プログラムが、デフォルトとして使用される最大 SSA サイズを計算します。計算されたサイズは、この PSB 内の任意の PCB 内の最大レベル数に 280 を掛けたものです。指定する値はバイト単位で、最大が 256000 です。

制約事項: DBCTL なしの CICS のもとで IMS を実行するとき、PSB 作業プール所要量は 64 KB を超えることはできません。

このプール所要量の主要なコンポーネントは、IOASIZE と SSASIZE です。PSB を ACBLIB に組み込んだとき、ACB 生成のメッセージ DFS0589I は PSB の合計作業プール・スペース所要量を示します。

重要: 高速機能副次索引の呼び出しでは、SSASIZE ワークエリアが、SUBSEQ フィールドからの追加ストレージおよび修飾子の数を収容する変換済み SSA を保持します。DL/I 呼び出しが開始されると、この変換済み SSA は全機能データベースに渡されます。

デフォルトの SSASIZE は、ACBGEN で定義されているデフォルトの SSA サイズに 840 バイトを加えた値に指定されます。

SSASIZE を指定した場合、またはデフォルトを使用していて SSASIZE の大きさが不足している場合、AU 状況コードが発行されます。この問題を修正するには、PSB での SSASIZE の値を大きくして PSBGEN および ACBGEN を再実行します。

#### **IOEROPN n**

バッチ・タイプ領域 (DLI または DBB) の場合にのみ適用できます。このパラメーターは、CICS には無効です。n サブパラメーターは条件コードであり、IMS が正常終了し、アプリケーション・プログラムの実行中にデータベースで 1 つ以上の入力エラーまたは出力エラーが起きたときにオペレーティング・システムに戻されます。n サブパラメーターは 0 から 4095 の数値です。

n=451 の場合は、IMS は条件コードをオペレーティング・システムに渡さずに、U451 異常終了を出して終了します。n=451 であって、IMS またはアプリケーション・プログラムが U451 以外で異常終了し、しかも入出力エラーが起きた場合には、プログラマー向け書き込みのメッセージ DFS0426I が出力されま

す。このメッセージは、実行時に入出力エラーが起きたこと、および U451 異常終了が起きたこと (実際の異常終了が起きていない場合) を示します。

n=451 の場合、オペレーターが DFS0451A メッセージに対して CONT を応答したとしても、IMS は異常終了 U0451 で終了します。

IOEROPN パラメーターを使用すれば、入出力エラーが起きたときの固有の JCL 条件コードを設定でき、後続のジョブ・ステップでその条件コードをテストできます。このパラメーターを指定しない場合には、アプリケーション・プログラムから渡される戻りコードはオペレーティング・システムに渡され、データベースの入出力エラーを示すのは状況コードおよびコンソール・メッセージのみになります。

#### **OLIC=**

この PSB のユーザーが、この PSB に指定されているデータベースに対して BMP として実行される調査ユーティリティ機能またはオンライン・データベース・イメージ・コピー・ユーティリティを実行する許可を与えられているかどうかを示します。YES を指定すると、オンライン・イメージ・コピー・ユーティリティおよび調査ユーティリティ機能を使用でき、NO を指定すると、オンライン・イメージ・コピー・ユーティリティおよび調査ユーティリティ機能を使用できません。NO がデフォルトです。PSB 内の任意のデータベース PCB (TYPE パラメーターに DB が指定されている PCB) が L または LS 処理オプションを指定する場合、このパラメーターは無効です。

例外: このパラメーターは、CICS、GSAM、HSAM、MSDB、または DEDB データベースに対して使用できません。

#### **GSROLBOK=**

次の場合に、内部 ROLB 呼び出しを行って非 GSAM データベース更新をロールバックするかどうかを制御します。

- アプリケーションが非メッセージ・ドリブン BMP であるとき。
- PSB に、GSAM データベースの PCB が含まれているとき。
- スレッド作成時または SQL 呼び出し時に、Db2 for z/OS がデッドロックを報告するとき。

YES は、内部 ROLB 呼び出しが行われること、およびデッドロックに関する SQL コードがアプリケーション・プログラムに戻されることを意味します。

NO は、ROLB 呼び出しではなく、ユーザー異常終了 777 が発生することを意味します。デフォルトは NO です。

#### **LOCKMAX=**

アプリケーション・プログラムが一度に獲得できるロックの最大数を示します。 $n$  は、0 から 255 の数値です。 $n$  は、1000 の単位で指定します。例えば、LOCKMAX=5 と指定すると、一度に最大 5000 個のロックが可能であることを示します。

デフォルトは 0 です。これは、一度に可能なロックの最大数がないことを示します。

コミットなしでアプリケーション・プログラムが延長して実行される場合は、データベースのレコードおよび変更に関して IMS が行うロックは累積可能で

す。LOCKMAX パラメーターを使用すれば、単一のアプリケーション・プログラムがすべてのロッキング・ストレージを使い切るために起こる他のプログラムの異常終了を防ぐことができます。

LOCKMAX=0 (限界を完全にオフにする) を指定するか、または従属領域 (BMP、MPP、または IFP) またはバッチ (DBB または DLI) に対して LOCKMAX=1 から 32767 を指定すれば、PSBGEN ステートメントで指定された LOCKMAX 値をプログラム実行時に変更できます。値は、1000 の単位です。この方法を使用すれば、PSBGEN ステートメントの LOCKMAX パラメーターに指定できる最大値 255 を超えることができます。

#### REMARKS=

オプションのユーザー・コメント。1 から 256 文字フィールド。

コメントにコンマや空白スペースなどの特殊文字が含まれる場合、コメント・ストリング全体を単一引用符で囲みます。

REMARKS キーワードで指定される値に、以下の文字を含むことはできません。

- 単一引用符 (コメント・ストリング全体を囲むために使用されている場合を除く)。単一引用符がコメント・ストリング全体の末尾より前に入力されると、そのコメント・ストリングの残りの部分は切り捨てられます。以下の例では、REMARKS キーワードでの単一引用符の正しい使い方と誤った使い方を示します。

#### CORRECT

```
REMARKS='These remarks apply to the XYZ application'
```

#### INCORRECT

```
REMARKS='These remarks apply to the 'XYZ' application'
```

- 二重引用符。
- より小 (<) 記号。
- より大 (>) 記号。
- アンパーサンド (&)。

メッセージ出力用の PCB ステートメントおよびデータベース用の PCB ステートメントを複数指定することができますが、PSB 生成 PCB ステートメント・デック内の PSBGEN は 1 つのみです。PSBGEN ステートメントは、デック内で END ステートメントの前にくる最後のステートメントにしなければなりません。

## END ステートメント

すべての PSB 生成ユーティリティーの制御ステートメントには、END ステートメントが続いていなければなりません。

END ステートメントは、マクロ・アセンブラーが必要とするもので、アセンブリ・データの最後を示すためのものです。

---

## PSBGEN ユーティリティーの例

以下の各例では、PSBGEN ユーティリティーを使用して PSB を生成する方法を示します。

## PSB 生成の例

この例は、階層データ構造を処理するメッセージ処理プログラムの PSB 生成です。データ構造に含まれるセグメントは、PARTMAST、CPWS、POLN、OPERTON、INVSTAT、および OPERSGMT です。

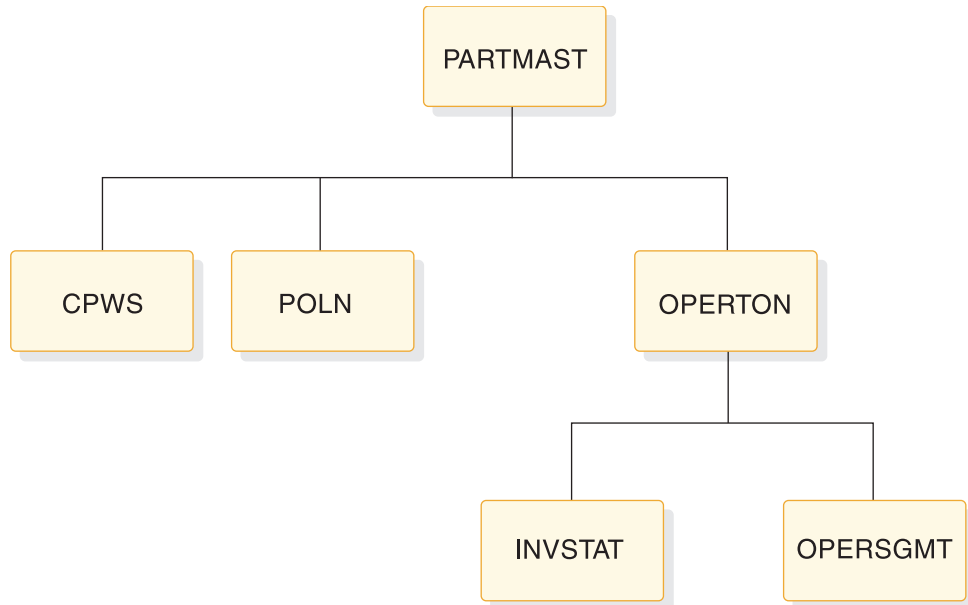


図 15. 階層データ構造の例

### 例 1

この例は、入力のソースを表す端末ばかりでなく、論理端末の OUTPUT1 と OUTPUT2 にも送信される出力メッセージを示しています。

```
//PSBGEN JOB
//      EXEC   PSBGEN,MBR=APPLPGM1
//C.SYSIN DD   *

      PCB     TYPE=TP,NAME=OUTPUT1,PCBNAME=OUTPCB1
      PCB     TYPE=TP,NAME=OUTPUT2,PCBNAME=OUTPCB2
      PCB     TYPE=DB,DBDNAME=PARTMSTR,PROCOPT=A,KEYLEN=100
      SENSEG  NAME=PARTMAST,PARENT=0,PROCOPT=A
      SENSEG  NAME=CPWS,PARENT=PARTMAST,PROCOPT=A
      SENSEG  NAME=POLN,PARENT=PARTMAST,PROCOPT=A
      SENSEG  NAME=OPERTON,PARENT=PARTMAST,PROCOPT=A
      SENSEG  NAME=INVSTAT,PARENT=OPERTON,PROCOPT=A
      SENSEG  NAME=OPERSGMT,PARENT=OPERTON
      PSBGEN  LANG=COBOL,PSBNAME=APPLPGM1
      END

/*
```

### 例 2

この例は、バッチ・プログラムで使用するステートメントを示しています。この PSB を使用するプログラムは、バッチ環境の通信 PCB を参照しません。

```
//PSBGEN JOB
//      EXEC   PSBGEN,MBR=APPLPGM2
//C.SYSIN DD   *
```

```

PCB    TYPE=DB,DBDNAME=PARTMSTR,PROCOPT=A,KEYLEN=100
SENSEG NAME=PARTMAST,PARENT=0,PROCOPT=A
SENSEG NAME=CPWS,PARENT=PARTMAST,PROCOPT=A
SENSEG NAME=POLN,PARENT=PARTMAST,PROCOPT=A
SENSEG NAME=OPERTON,PARENT=PARTMAST,PROCOPT=A
SENSEG NAME=INVSTAT,PARENT=OPERTON,PROCOPT=A
SENSEG NAME=OPERSGMT,PARENT=OPERTON
PSBGEN LANG=COBOL,PSBNAME=APPLPGM2
END

```

/\*

### 例 3

この例は、PSB 生成が、バッチ・メッセージ処理プログラムについて実行されることを示しています。GSAM PCB は、報告書ファイルを生成するためにアプリケーション・プログラムが使用します。

```

//PSBGEN JOB
//      EXEC   PSBGEN,MBR=APPLPGM3
//C.SYSIN DD   *

```

```

PCB    TYPE=TP,NAME=OUTPUT1
PCB    TYPE=TP,NAME=OUTPUT2
PCB    TYPE=DB,DBDNAME=PARTMSTR,PROCOPT=A,KEYLEN=100
SENSEG NAME=PARTMAST,PARENT=0,PROCOPT=A
SENSEG NAME=CPWS,PARENT=PARTMAST,PROCOPT=A
PCB    TYPE=GSAM,DBDNAME=REPORT,PROCOPT=LS
PSBGEN LANG=COBOL,PSBNAME=APPLPGM3
END

```

/\*

### 例 4

この例は、PSB 生成が、バッチ・プログラムについて実行されることを示しています。PCB には名前 (PRTMASTR) が付けられています。PCB の名前は、AIBTDLI インターフェースを使用する DL/I 呼び出しで使用されます。

```

//PSBGEN JOB
//      EXEC   PSBGEN,MBR=APPLPGM4
//C.SYSIN DD   *

```

```

PCB    TYPE=DB,DBDNAME=PARTMSTR,PROCOPT=A,KEYLEN=100,PCBNAME=PARTMSTR
SENSEG NAME=PARTMAST,PARENT=0,PROCOPT=A
SENSEG NAME=CPWS,PARENT=PARTMAST,PROCOPT=A
SENSEG NAME=POLN,PARENT=PARTMAST,PROCOPT=A
SENSEG NAME=OPERTON,PARENT=PARTMAST,PROCOPT=A
SENSEG NAME=INVSTAT,PARENT=OPERTON,PROCOPT=A
SENSEG NAME=OPERSGMT,PARENT=OPERTON
PSBGEN LANG=COBOL,PSBNAME=APPLPGM4
END

```

/\*

### 例 5

この例は、PSB 生成が、バッチ・プログラムについて実行されることを示しています。ラベル (PARTROOT) は、PCB 内の唯一のルート・セグメントを示すのに使用されます。PCB のアドレスは、入り口でアプリケーションに渡される PCB リストから除外されます。

```

//PSBGEN JOB
//      EXEC   PSBGEN,MBR=APPLPGM5
//C.SYSIN DD   *

```

```

PARTROOT PCB TYPE=DB,DBDNAME=PARTMSTR,PROCOPT=A,LIST=NO
SENSEG NAME=PARTMAST,PARENT=0,PROCOPT=A
PSBGEN LANG=COBOL,PSBNAME=APPLPGM5
END
/*

```

## フィールド・レベル・センシティブィー PSB 生成の例

以下の図は、フィールド・レベル・センシティブィーを使用するバッチ・プログラム用の PCB を示しています。この図では、セグメントの階層順を示します。employee セグメントが第 1 レベルです。office セグメントと employee project セグメントは 2 次レベルです。階層構造外であるが、2 次レベルで、project セグメントが employee project セグメントに接続しています。

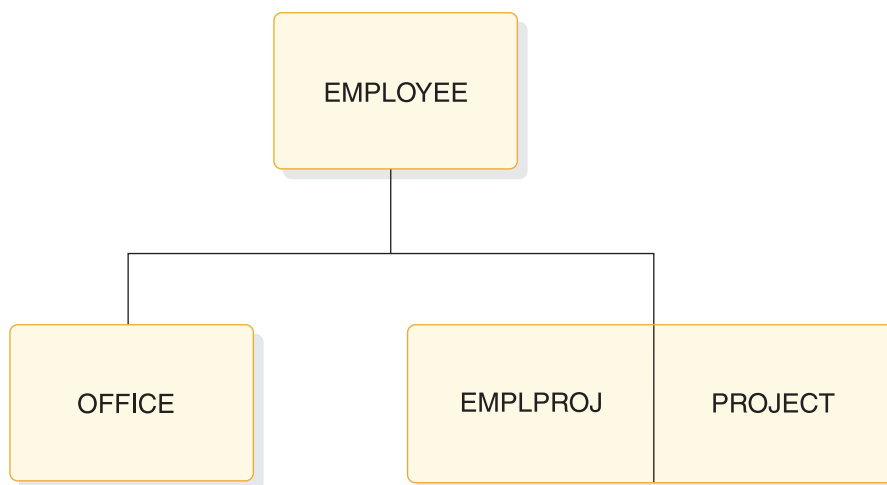


図 16. フィールド・レベル・センシティブィー PSB 生成の例

SEGMENT NAME	FIELD NAME	START LOCATION	LENGTH
EMPLOYEE	EMPSSN	1	9
	EMPLNAME	10	10
	EMPFNAME	20	9
	EMPMI	29	1
	EMPADDR	30	30
OFFICE	OFNUMBER	1	5
	OFFPHONE	6	7
EMPLPROJ	EPFUNCTN	1	20
	EPTIMEST	21	5
	EPTIMCUR	26	5
PROJECT	PROJNUM	1	8
	PROJTTL	9	20
	PROJSTRT	29	8
	PROJEND	37	8
	PROJSTAT	45	1

```

//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

```

```

PCB TYPE=DB,NAME=FISDBD1,PROCOPT=GRP,KEYLEN=20
SENSEG NAME=EMPLOYEE,PARENT=0
SENFLD NAME=EMPLNAME,START=13,REPL=NO

```

```

SENFLD  NAME=EMPFNAME,START=1,REPL=NO
SENFLD  NAME=EMPMI,START=11
SENSEG  NAME=OFFICE,PARENT=EMPLOYEE
SENSEG  NAME=EMPLPROJ,PARENT=EMPLOYEE
SENFLD  NAME=PROJNUM,START=1
SENFLD  NAME=PROJTITLE,START=10
SENFLD  NAME=EPFUNCTN,START=35
SENFLD  NAME=EPTIMEST,START=60
SENFLD  NAME=EPTIMCUR,START =70
PSBGEN  LANG=ASSEM,PSBNAME=APPLPGM1
END

```

/\*

## 高速機能 PSB 生成の例

以下の 2 つの例は、高速機能 PSB 生成のサンプルです。

### 例 1

これは、8 個の PCB が入っている MSDB PSB のステートメントの例です。

```

//PSBGEN  JOB
//          EXEC    PSBGEN,MBR=APPLPGM1
//C.SYSIN  DD      *

PCB      TYPE=DB,DBDNAME=MSDBLM01,PROCOPT=R,      NONTERMINAL-RELATED      X
          KEYLEN=4                                END OF PCB STATEMENT
SENSEG   NAME=LDM,PARENT=0                        (DEFAULT)
PCB      TYPE=DB,DBDNAME=MSDBLM02,PROCOPT=R,      NONTERMINAL-RELATED      X
          KEYLEN=1
SENSEG   NAME=LDM,PARENT=0
PCB      TYPE=DB,DBDNAME=MSDBLM03,PROCOPT=R,      NONTERMINAL-RELATED      X
          KEYLEN=2
SENSEG   NAME=LDM,PARENT=0
PCB      TYPE=DB,DBDNAME=MSDBLM04,PROCOPT=R,      NONTERMINAL-RELATED      X
          KEYLEN=8                                TERM KEYS
SENSEG   NAME=LDM,PARENT=0
PCB      TYPE=DB,DBDNAME=MSDBLM05,PROCOPT=R,      FIXED RELATED            X
          KEYLEN=8
SENSEG   NAME=LDM,PARENT=0
PCB      TYPE=DB,DBDNAME=MSDBLM06,PROCOPT=A,      DYNAMIC RELATED         X
          KEYLEN=8
SENSEG   NAME=LDM,PARENT=0
PCB      TYPE=DB,DBDNAME=MSDBLM06,PROCOPT=R,      DYNAMIC RELATED         X
          KEYLEN=8
SENSEG   NAME=LDM,PARENT=0
PCB      TYPE=DB,DBDNAME=MSDBLM06,PROCOPT=G,      DYNAMIC RELATED         X
          KEYLEN=8
SENSEG   NAME=LDM,PARENT=0
PSBGEN   LANG=ASSEM,PSBNAME=APPLPGM1              END OF PSBGEN MACRO
END      END OF PSB GEN
/*

```

### 例 2

これは、DEDB サブセット・ポインターのステートメントの例です。

```

//PSBGEN  JOB
//          EXEC    PSBGEN,MBR=APPLPGM1
//C.SYSIN  DD      *

PCB      TYPE=DB,DBDNAME=MSDBLM01,PROCOPT=R,      NONTERMINAL-RELATED      X
PCB      TYPE=DB,DBDNAME=X,PROCOPT=A,KEYLEN=100
SENSEG   NAME=A,PARENT=C
SENSEG   NAME=B,PARENT=A,SSPTR=((1,R),(2,U),(5))
SENSEG   NAME=C,PARENT=B

```

```

SENSEG NAME=D,PARENT=A,SSPTR=((2,R))
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*

```

注:

1. SSPTR=((n,r))
  - n** この SENSEG 内のサブセット・ポインター数。
  - r** ポインターのセンシティビティ (R : 読み取り、U : 更新)
2. n も r も省略すると、ポインターにはセンシティビティがありません。
3. n を指定し r を指定しないと、デフォルトは R です (読み取りセンシティブ)。

### 追加の PSB 生成の例

#### 例 1

以下の図は、バッチ・プログラムに対して実行されている PSB 生成を示しています。この図では、セグメントの階層順を示します。Skill セグメントが第 1 レベルです。Name セグメント (これは payroll と skill に分かれています) は 2 次レベルです。Address、Payroll、Expr、および Educ は 3 次レベルです。

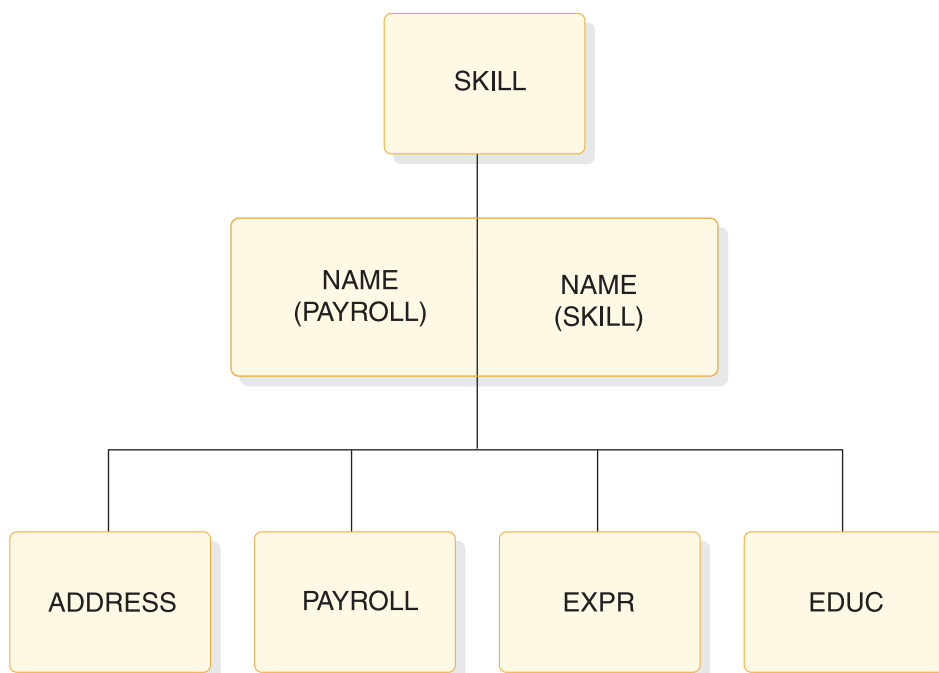


図 17. DL/I データベース・ステートメントの定義に使用される PSBGEN ステートメント (例 1)

```

//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

PCB TYPE=DB,DBDNAME=LOGIC1;PROCOPT=G,KEYLEN=151,POS=M
SENSEG NAME=SKILL,PARENT=0,PROCOPT=A
SENSEG NAME=NAME,PARENT=SKILL,PROCOPT=A

```



```

SENSEG    NAME=ADDRESS,PARENT=NAME,PROCOPT=A
SENSEG    NAME=PAYROLL,PARENT=NAME,PROCOPT=A
SENSEG    NAME=EXPR,PARENT=NAME,PROCOPT=A
SENSEG    NAME=EDUC,PARENT=NAME,PROCOPT=A
PSBGEN    LANG=COBOL,PSBNAME=APPLPGM1
END
/*

```

例 2:

以下の図は、バッチ・プログラムに対して実行されている PSB 生成を示しています。この図では、セグメントの階層順を示します。NAME セグメントが第 1 レベルです。NAMESK、ADDRESS、および PAYROLL セグメントは 2 次レベルです。Expr および Educ セグメントは 3 次レベルで、NAMESK セグメントと結合しています。この図では、NAMESK セグメントは NAMESKIL と SKILL に分離されていますが、SENSEG ステートメントでこれらを別のセグメントとして定義するわけではありません。

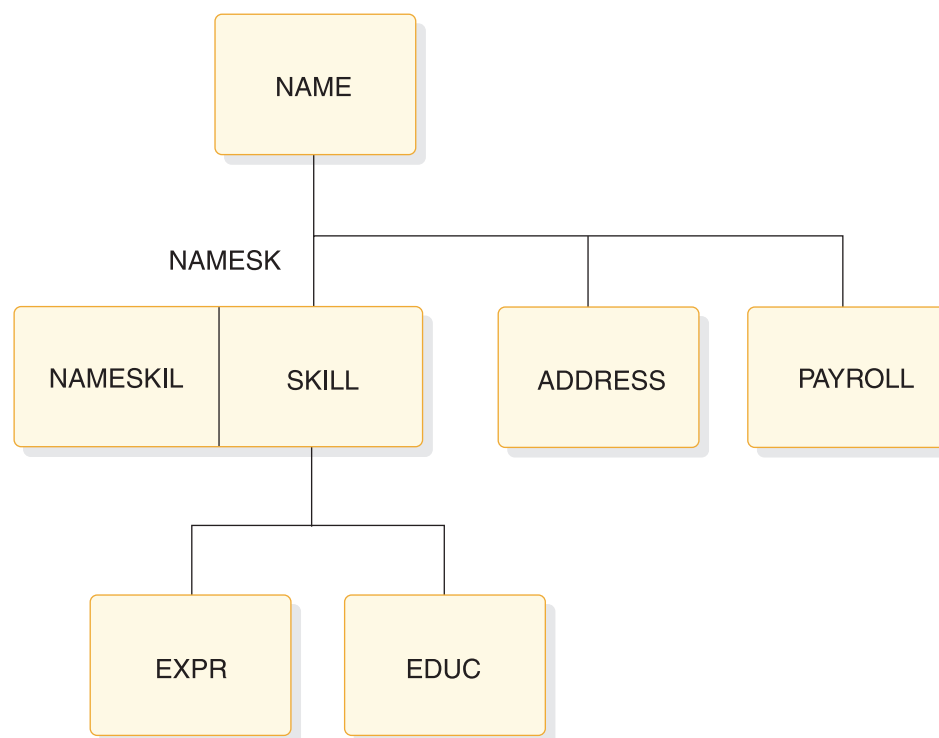


図 18. DL/I データベースの PCB ステートメントを定義する際に使用される PSBGEN PCB ステートメント (例 2)

```

//PSBGEN    JOB
//          EXEC    PSBGEN,MBR=APPLPGM1
//C.SYSIN   DD     *

PCB        TYPE=DB,DBDNAME=LOGICDB,PROCOPT=A,KEYLEN=241,POS=M
SENSEG     NAME=NAME,PARENT=0,PROCOPT=G
SENSEG     NAME=NAMESK,PARENT=NAME,PROCOPT=G
SENSEG     NAME=EXPR,PARENT=NAMESK,PROCOPT=G
SENSEG     NAME=EDUC,PARENT=NAMESK,PROCOPT=G
SENSEG     NAME=ADDRESS,PARENT=NAME,PROCOPT=G

```

```

SENSEG    NAME=PAYROLL,PARENT=NAME,PROCOPT=G
PSBGEN    LANG=PL/I,PSBNAME=APPLPGM1
END
/*

```

例 3:

以下の図は、DL/I データベース内でのセグメント相互間の論理関係を定義する PSB を示しています。この図は、セグメント PARTMAST (親セグメント)、CPWS、POLN、INVSTAT、および OPERSGMT (これらはすべて、PARTMAST の第 1 レベル子セグメント) の階層順を示しています。代替ステートメントで、出力を論理端末 "OUTPUT" に送信します。PSBGEN ステートメントで、この JCL を APPLPGM1 として IMS.PSBLIB ライブラリーに保管します。

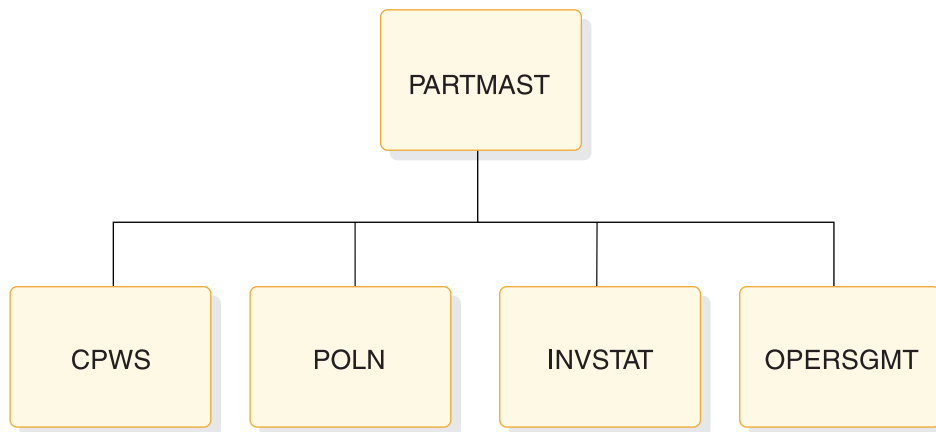


図 19. DL/I データベースの PCB ステートメントを定義する際に使用される PSBGEN PCB ステートメント (例 3)

```

//PSBGEN  JOB
//          EXEC   PSBGEN,MBR=APPLPGM1
//C.SYSIN  DD   *

PCB        TYPE=TP,LTERM=OUTPUT
PCB        TYPE=DB,DBDNAME=PARTMSTR,PROCOPT=GIDR,KEYLEN=100
SENSEG     NAME=PARTMAST,PARENT=0,PROCOPT=A
SENSEG     NAME=CPWS,PARENT=PARTMAST,PROCOPT=A
SENSEG     NAME=POLN,PARENT=PARTMAST,PROCOPT=A
SENSEG     NAME=INVSTAT,PARENT=PARTMAST,PROCOPT=A
SENSEG     NAME=OPERSGMT,PARENT=PARTMAST
PSBGEN     LANG=COBOL,PSBNAME=APPLPGM1
END
/*

```

例 4:

以下の図は、DL/I データベース PO DB の POMSTR セグメントと POLNITEM セグメントとの間の論理関係の定義に使用される JCL を示しています。代替ステートメントで、トランザクション・コード名 "out1" および "out2" を使用して、出力をアプリケーションに送信します。

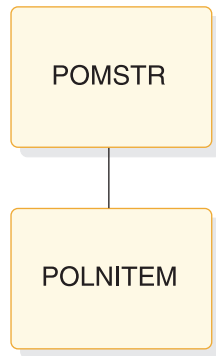


図 20. 論理関係および作成出力を定義する際に使用される *PSBGEN PCB* ステートメント

```

//PSBGEN  JOB
//        EXEC  PSBGEN,MBR=APPLPGM1
//C.SYSIN  DD   *

PCB      TYPE=TP,NAME=OUT1
PCB      TYPE=TP,NAME=OUT2
PCB      TYPE=DB,DBDNAME=PODB,PROCOPT=GID,KEYLEN=200
SENSEG   NAME=POMSTR
SENSEG   NAME=POLNITEM,PARENT=POMSTR
PSBGEN   LANG=COBOL,PSBNAME=APPLPGM1
END
/*
  
```

### アプリケーション・データベースのサンプル問題の例

例 5 から 10 では、それぞれの JCL で作成される論理データベースの基礎として、DBDNAME=DI21PART を使用します。このデータベースは、セグメント PARTROOT、STANINFO、STOKSTAT、CYCCOUNT、および BACKORDR を含みます。PARTROOT は親セグメントです。STANINFO および STOKSTAT は PARTROOT の子セグメントです。CYCCOUNT および BACKORDR は STOKSTAT の子セグメントです。

#### 例 1:

以下の図は、メッセージ通信または会話型メッセージング・プログラムのいずれかを示しています。この JCL は、ロード・モジュール DFSSAM01 として IMS.PSBLIB ライブラリーに保管されます。

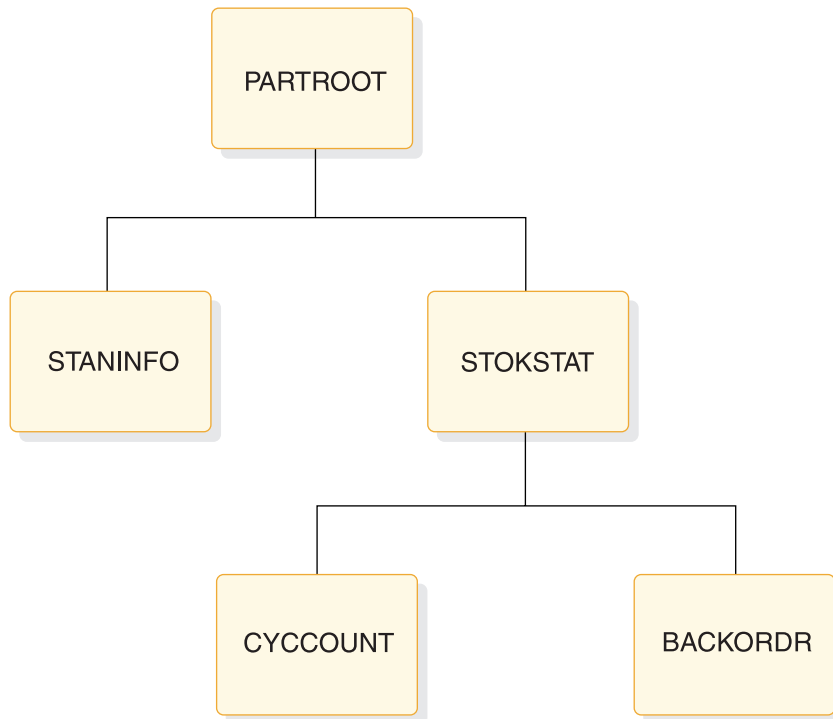


図 21. メッセージ通信または会話型メッセージ・プログラム用のデータ構造および JCL

```

//PSBGEN JOB
//      EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

SENSEG  NAME=BACKORDR,PARENT=STOKSTAT
PSBGEN  LANG=COBOL,PSBNAME=APPLPGM1
END
/*
  
```

例 2:

以下の図に示されている JCL は、PARTROOT セグメントと STANINFO セグメントの間の論理関係を定義しています (図では陰影を付けて示してある)。この JCL は、ロード・モジュール DFSSAM02 として IMS.PSBLIB ライブラリーに保管されます。

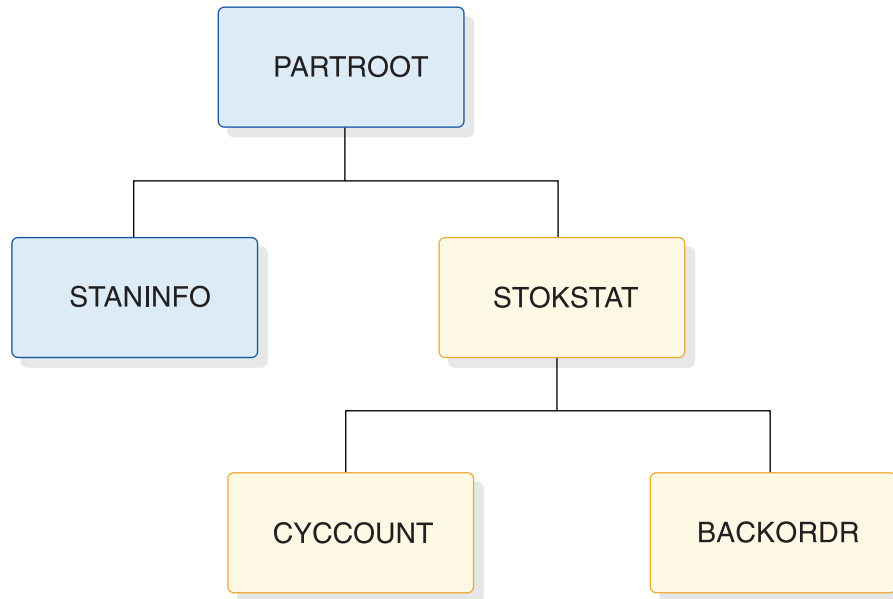


図 22. データベース DI21PART 内の論理関係についてのデータ構造および JCL

```

//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *
  
```

```

PCB TYPE=DB,DBDNAME=DI21PART,PROCOPT=G,KEYLEN=19
SENSEG NAME=PARTROOT
SENSEG NAME=STANINFO,PARENT=PARTROOT
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*
  
```

例 3:

以下の図は、DL/I データベース DI21PART の論理構造全体を定義しています。この JCL は、ロード・モジュール DFSSAM03 として IMS.PSBLIB ライブラリーに保管されます。

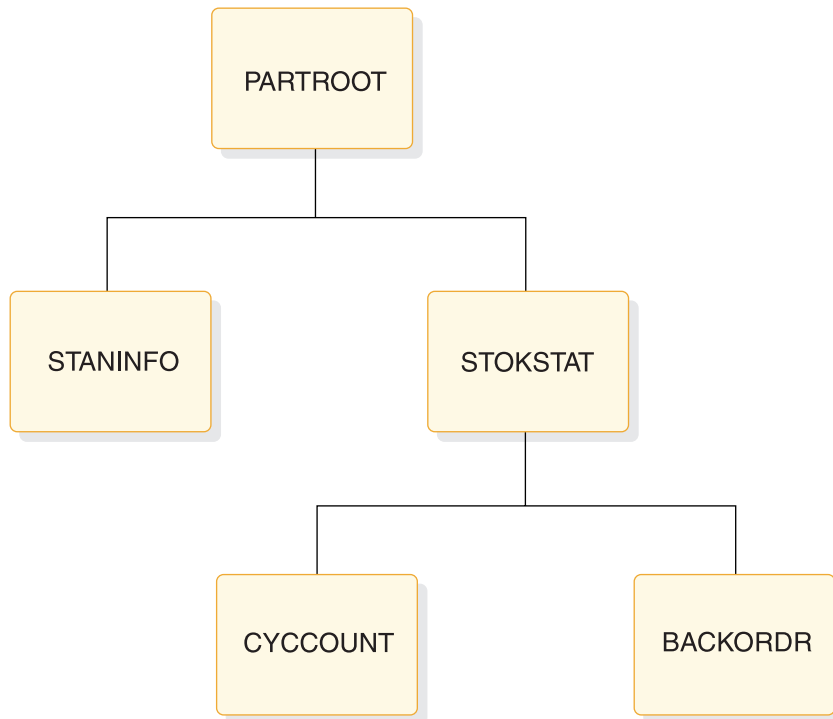


図 23. DL/I データベース DI21PART から定義された論理データベースのデータ構造および JCL

```

//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

PCB TYPE=DB,DBDNAME=DI21PART,PROCOPT=G,KEYLEN=43
SENSEG NAME=PARTROOT
SENSEG NAME=STANINFO,PARENT=PARTROOT
SENSEG NAME=STOKSTAT,PARENT=PARTROOT
SENSEG NAME=CYCCOUNT,PARENT=STOKSTAT
SENSEG NAME=BACKORDR,PARENT=STOKSTAT
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*
  
```

例 4:

以下の図は、PARTROOT セグメントと STOKSTAT セグメントの間の論理関係を定義しています (図では陰影を付けて示してある)。JCL は、論理端末 HOWARD へも出力し、JCL をロード・モジュール DFSSAM03 として IMS.PSBLIB ライブラリーに保管します。

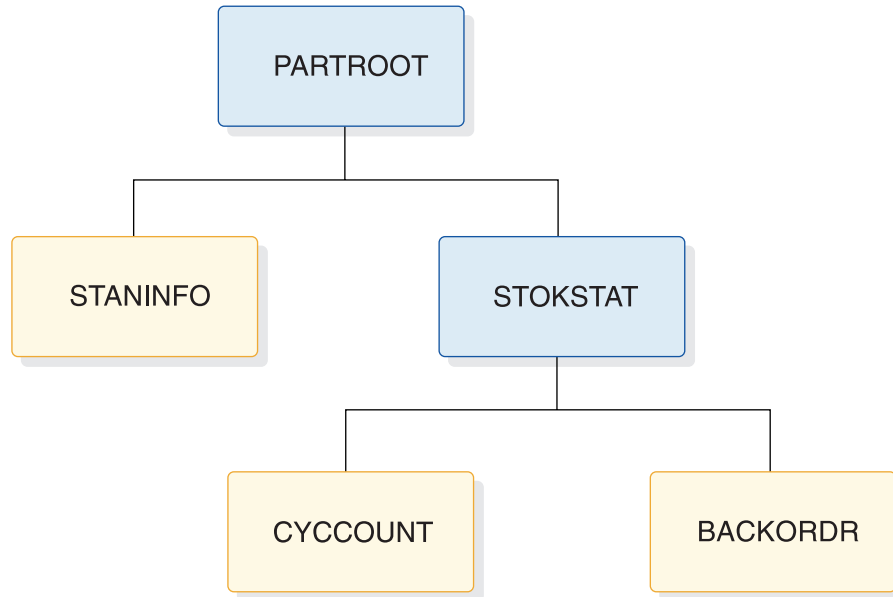


図 24. データベース DI21PART 内で出力を作成する論理関係についてのデータ構造および JCL (パート 1)

```

//PSBGEN JOB MSGLEVEL=1
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

PCB TYPE=TP,LTERM=HOWARD
PCB TYPE=DB,DBDNAME=DI21PART,PROCOPT=A,KEYLEN=33
SENSEG NAME=PARTROOT
SENSEG NAME=STOKSTAT,PARENT=PARTROOT
PSBGEN LANG=COBOL,PSBNAME=DFSSAM05
END
/*
  
```

例 5:

以下の図は、例 8 と同じですが、この JCL をロード・モジュール DFSSAM06 として IMS.PSBLIB ライブラリーに保管する点が違います。

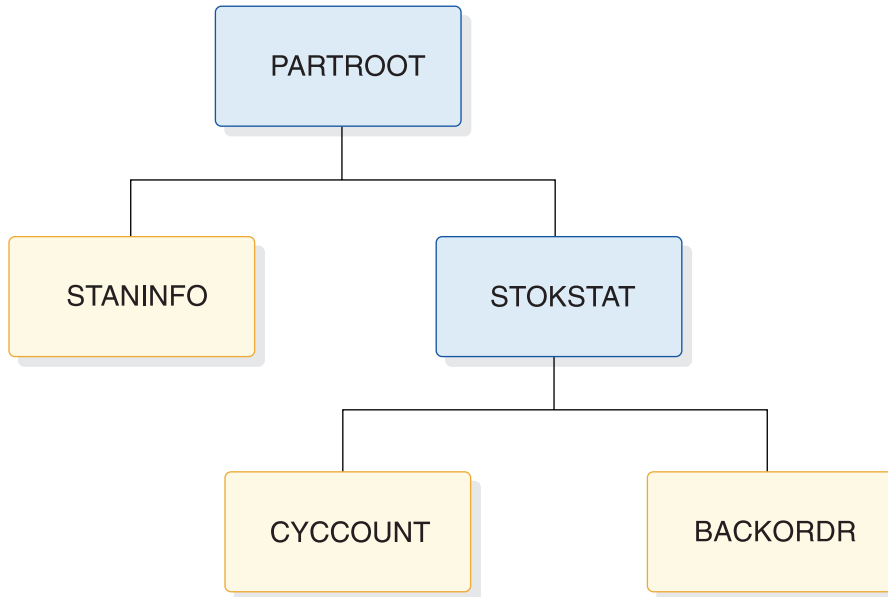


図 25. データベース DI21PART 内で出力を作成する論理関係についてのデータ構造および JCL (パート 2)

```

//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *
  
```

```

PCB TYPE=TP,LTERM=HOWARD
PCB TYPE=DB,DBDNAME=DI21PART,PROCOPT=A,KEYLEN=33
SENSEG NAME=PARTROOT
SENSEG NAME=STOKSTAT,PARENT=PARTROOT
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*
  
```

**例 6:**

以下の図は、DL/I データベース DI21PART の論理構造全体を定義しています。この JCL は、ロード・モジュール DFSSAM07 として IMS.PSBLIB ライブラリーに保管されます。



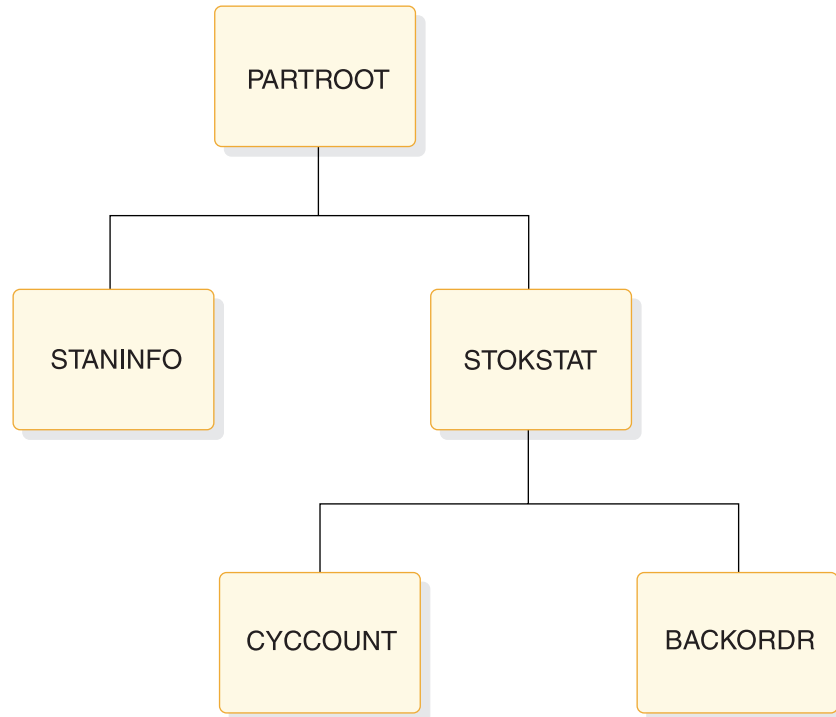


図 26. DL/I データベース DI21PART から定義された論理データベースのデータ構造および JCL

```

//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

PCB TYPE=DB,DBDNAME=DI21PART,PROCOPT=G,KEYLEN=43
SENSEG NAME=PARTROOT
SENSEG NAME=STANINFO,PARENT=PARTROOT
SENSEG NAME=STOKSTAT,PARENT=PARTROOT
SENSEG NAME=CYCCOUNT,PARENT=STOKSTAT
SENSEG NAME=BACKORDR,PARENT=STOKSTAT
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*
  
```

### 共用副次索引の例

この例のデータベース構造を次の図に示します。この図で示すデータベース DTA3 は、共用副次索引データベース X4 内の 3 つの副次索引 (X4、X5、および X6) によって索引付けされています。各副次索引は、索引ターゲット・セグメントと索引ソース・セグメントの両方に異なるセグメントを使用します。副次索引 X4 は、DTA3 のセグメント DA をターゲット/ソース・セグメントとして使用します。副次索引 X5 は、DTA3 のセグメント DC をターゲット/ソース・セグメントとして使用します。副次索引 X6 は、DTA3 のセグメント DE をターゲット/ソース・セグメントとして使用します。

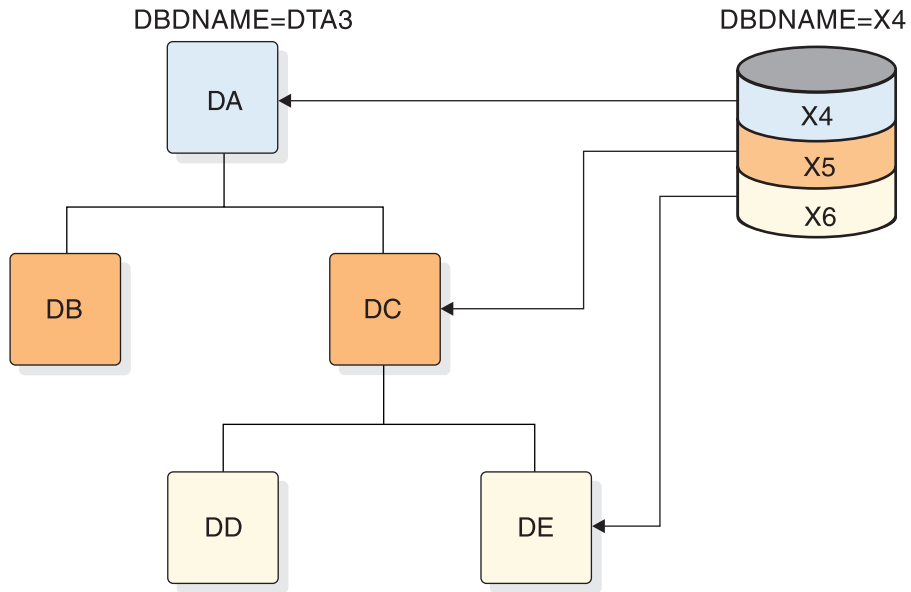


図 27. 共用副次索引データベース内の 3 つの副次索引で索引付けされているデータベース

DA 経由の索引のデータベース構造を次の図に示します。これはセグメント DA、DB、DC、DD、および DE を含みます。

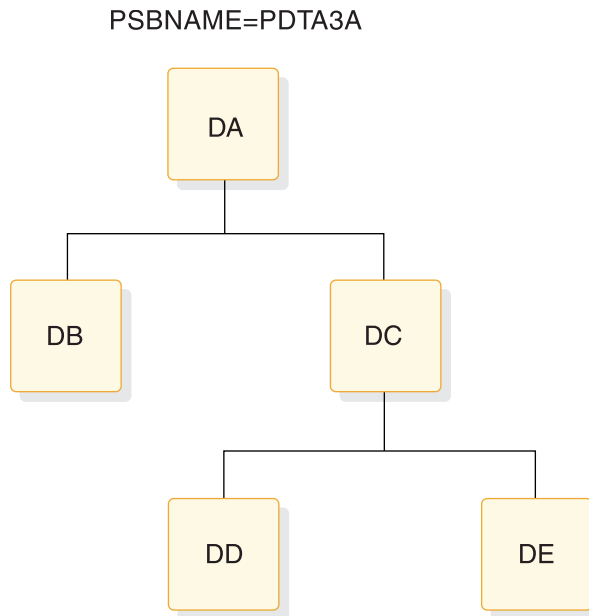


図 28. セグメント DA 経由の索引のデータ構造および JCL

```
//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *
```

```
PCB TYPE=DB,DBDNAME=DTA3,PROCOPT=A,KEYLEN=15,PROCSEQ=X4
SENSEG NAME=DA,PARENT=0
SENSEG NAME=DB,PARENT=DA
SENSEG NAME=DC,PARENT=DA,INDICES=X5
SENSEG NAME=DD,PARENT=DC
```

```

SENSEG    NAME=DE,PARENT=DC,INDICES=X6
PSBGEN    LANG=COBOL,PSBNAME=APPLPGM1
END
/*

```

DC 経由の索引のデータベース構造を次の図に示します。これは、セグメント DC、DA、DD、および DE を示しています。

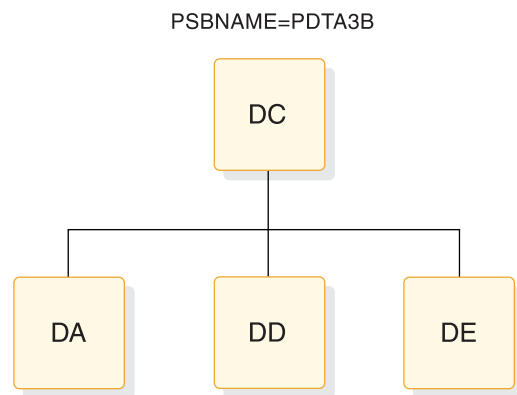


図 29. セグメント DC 経由の索引のデータ構造および JCL

```

//PSBGEN  JOB
//          EXEC    PSBGEN,MBR=APPLPGM1
//C.SYSIN  DD      *

PCB        TYPE=DB,DBDNAME=DTA3,PROCOPT=A,KEYLEN=15,PROCSEQ=X5
SENSEG     NAME=DC,PARENT=0
SENSEG     NAME=DA,PARENT=DC,INDICES=X4
SENSEG     NAME=DD,PARENT=DC
SENSEG     NAME=DE,PARENT=DC,INDICES=X6
PSBGEN     LANG=COBOL,PSBNAME=APPLPGM1
END
/*

```

さらにこのデータベース構造は、副構造として、DA 経由の索引のデータベース構造を含むことができます。

DE 経由の索引のデータベース構造を次の図に示します。これには、セグメント DE、DC、および DA が示されています。

PSBNAME=PDTA3B

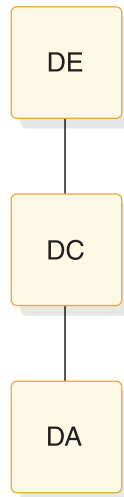


図 30. セグメント DE 経由の索引のデータ構造および JCL

```
//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

PCB TYPE=DB,DBDNAME=DTA3,PROCOPT=A,KEYLEN=15,PROCSEQ=X6
SENSEG NAME=DE,PARENT=0
SENSEG NAME=DC,PARENT=DE,INDICES=X5
SENSEG NAME=DA,PARENT=DC,INDICES=X4
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*
```

さらに、このデータベース構造は、DA と DC 経由の索引のデータベース構造を副構造として含むことができます。

INDEX データベースの PCB は、次のとおりです。

```
//PSBGEN JOB
// EXEC PSBGEN,MBR=APPLPGM1
//C.SYSIN DD *

PCB TYPE=DB,DBDNAME=X4,PROCOPT=A,KEYLEN=5
SENSEG NAME=X4A,PARENT=0
PCB TYPE=DB,DBDNAME=X5,PROCOPT=A,KEYLEN=5
SENSEG NAME=X5A,PARENT=0
PCB TYPE=DB,DBDNAME=X6,PROCOPT=A,KEYLEN=5
SENSEG NAME=X6A,PARENT=0
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
END
/*
```

### 高速機能副次索引の例

以下の例は、ターゲット・セグメントがルート・セグメントである場合に、PROCSEQD= パラメーターを使用する、PSB 内の高速機能副次索引 PCB を示しています。

PCB PCB2NDX を選択すると、1 次 DEDB データベース EDUCDB は、その高速機能副次索引データベース NAMESXDB を使用してアクセスされます。

PROCSEQD パラメーターは、1 次 DEDB データベース EDUCDB へのアクセスに使用する高速機能副次索引データベースの名前 NAMESXDB を指定します。

ターゲット・セグメントである COURSE セグメントがルート・セグメントです。ルート・セグメントの下のすべてのセグメントが、1 次 DEDB データベースの物理構造全体で NAMESXDB 副次索引を使用してアクセス可能です。

```
PCB2NDX
PCB      TYPE=DB,DBDNAME=EDUCDB,PROCOPT=A,KEYLEN=100,PROCSEQD=NAMESXDB
SENSEG   NAME=COURSE,PARENT=0,PROCOPT=GR
SENSEG   NAME=CLASS,PARENT=COURSE
SENSEG   NAME=INSTRUCT,PARENT=CLASS          <<-- target segment
SENSEG   NAME=STUDENT,PARENT=CLASS
PSBGEN   PSBNAME=NAMEXPSB,LANG=COBOL
END
```

以下の例は、ターゲット・セグメントがルート・セグメントでない場合に、PROCSEQD= パラメーターを使用する、PSB 内の高速機能副次索引 PCB を示しています。

PCB PCB3NDX を選択すると、1 次 DEDB データベース EDUCDB は、その高速機能副次索引データベース INSTSXDB を使用してアクセスされます。PROCSEQD パラメーターは、1 次 DEDB データベース EDUCDB へのアクセスに使用する高速機能副次索引データベースの名前 INSTSXDB を指定します。

ターゲット・セグメントである INSTRUCT セグメントはルート・セグメントではありません。ルート・セグメントからの物理パスに沿ってターゲット・セグメントの直接的な親であるセグメントと、ターゲット・セグメントのすべての子セグメントのみが、1 次 DEDB データベースが INSTSXDB 副次索引を使用してアクセスされるときに 1 次 DEDB データベースの物理構造内でアクセス可能です。

注: STUDENT セグメントにはアクセスできません。このセグメントが、ターゲット・セグメントである INSTRUCT セグメントの直接的な親セグメントでも子セグメントでもないためです。

```
PCB3NDX
PCB      TYPE=DB,DBDNAME=EDUCDB,PROCOPT=GR,KEYLEN=100,PROCSEQD=INSTSXDB
SENSEG   NAME=COURSE,PARENT=0
SENSEG   NAME=CLASS,PARENT=COURSE
SENSEG   NAME=INSTRUCT,PARENT=CLASS          <<-- target segment
PSBGEN   PSBNAME=NAMEXPSB,LANG=COBOL
END
```

以下の例は、ターゲット・セグメントがルート・セグメントでない場合に、PROCSEQD= パラメーターを使用する、PSB 内の高速機能副次索引 PCB を示しています。

PCB PCB4NDX を選択すると、1 次 DEDB データベース EDUCDB は、その高速機能副次索引データベース CLASSXDB を使用してアクセスされます。PROCSEQD パラメーターは、1 次 DEDB データベース EDUCDB へのアクセスに使用する高速機能副次索引データベースの名前 CLASSXDB を指定します。

ターゲット・セグメントである CLASS セグメントはルート・セグメントではありません。ルート・セグメントからの物理パスに沿ってターゲット・セグメントの直接的な親であるセグメントと、ターゲット・セグメントのすべての子セグメントのみが、1 次 DEDB データベースが CLASSXDB 副次索引データベースを使用して

アクセスされるときに 1 次 DEDB データベースの物理構造内でアクセス可能です。STUDENT セグメントはアクセス可能です。このセグメントが、ターゲット・セグメントである CLASS セグメントの子セグメントであるためです。

```
PCB4NDX
PCB      TYPE=DB,DBDNAME=EDUCDB,PROCOPT=GR,KEYLEN=100,PROCSEQD=CLASSXDB
SENSEG   NAME=COURSE,PARENT=0
SENSEG   NAME=CLASS,PARENT=COURSE
SENSEG   NAME=INSTRUCT,PARENT=CLASS
SENSEG   NAME=STUDENT,PARENT=CLASS
PSBGEN   PSBNAME=NAMEXPSB,LANG=COBOL
END
```

次の例は、高速機能副次索引をデータベースとして示しています。

```
PCB2XDB
PCB      TYPE=DB,DBDNAME=NAMESXDB,PROCOPT=GR,KEYLEN=10
SENSEG   NAME=NAMEXSEG,PARENT=0
PSBGEN   PSBNAME=DB2DXPSB,LANG=COBOL
END
```

---

## PSBGEN プロシーチャーの実行

IMS のシステム定義で、PSBGEN という名前のプロシーチャーが IMS.PROCLIB プロシーチャー・ライブラリーに入ります。このアセンブルとバインドからなる 2 ステップ・プロシーチャーにより、PSB が生成されます。

最初のステップ (ステップ C) は、オペレーティング・システム・アセンブリーで、プロシーチャーが呼び出された後に実行されます。2 番目のステップ (ステップ L) はバインドで、ステップ C からのアセンブリー出力を受け取り、PSB を IMS.PSBLIB に入れます。

### プロシーチャー・ステートメント

次の図はプロシーチャー・ステートメントを示しています。図の後に続くリストに、このステートメントで使用されているパラメーターが定義してあります。

```
//      PROC MBR=TEMPNAME,SOUT=A,RGN=0M,SYS2=
//C     EXEC PGM=ASMA90,REGION=&RGN,
//      PARM=(OBJECT,NODECK,NODBCS,
//      'SIZE(MAX,ABOVE)')
//SYSLIB DD DSN=IMS.&SYS2.SDFSMAC,DISP=SHR
//SYSLIN DD UNIT=SYSDA,DISP=(,PASS),
//      SPACE=(80,(100,100),RLSE),
//      DCB=(BLKSIZE=80,RECFM=F,LRECL=80)
//SYSPRINT DD SYSOUT=&SOUT,DCB=BLKSIZE=1089,
//      SPACE=(121,(300,300),RLSE,,ROUND)
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(CYL,(10,5))
//L     EXEC PGM=IEWL,PARM='XREF,LIST',
//      COND=(0,LT,C),REGION=4M
//SYSLIN DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=&SOUT,DCB=BLKSIZE=1089,
//      SPACE=(121,(90,90),RLSE)
//SYSLMOD DD DISP=SHR,
//      DSN=IMS.&SYS2.PSBLIB(&MBR)
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),
//      SPACE=(1024,(100,10),RLSE),DISP=(,DELETE)
```

**MBR=**

生成する PSB の名前です。この名前は、PSBGEN ステートメントの

PSBNAME= パラメーターに指定した名前と同じでなければなりません。この予防措置をとらないと、実行時にユーザー ABEND 929 が起こるか、あるいは ACB 生成「BUILD PSB」操作時にメッセージ DFS929I (「BLDL FAILED FOR MEMBER」) を受け取ることがあります。

#### **SOUT=**

SYSOUT クラスを指定します。デフォルトは A です。

#### **RGN=**

PSBGEN ユーティリティーの実行用の領域サイズを指定します。デフォルトは 512KB です。

#### **SYS2=**

XRF 複合システムで「オプションの複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS2='IMSA.' のように、パラメーターを後書きピリオドも含めて引用符で囲みます。

## **ステップ C**

ステップ C は、アセンブリー・ステップです。

#### **DD** ステートメント

##### **SYSIN DD**

ステップ C への入力データ・セットを定義します。プロシージャを呼び出す際には、これらの DD ステートメントを指定する必要があります。

## **ステップ L**

ステップ L は、バインド・ステップです。

例: このステップは、次に示すようにバインド EXEC ステートメントの PARM リストに AMODE=31 を追加することによって、デフォルトの AMODE=24、RMODE=24 ではなく、AMODE=31、RMODE=24 を使用して実行できます。

```
//L      EXEC  PGM=IEWL,PARM='XREF,LIST,AMODE=31',  
//              COND=(0,LT,C),REGION=120K
```

AMODE または RMODE に別の値を指定しなければ、デフォルト値が有効になります。常に、RMODE=24 を指定してバインド・ステップを実行する必要があります。

#### **DD** ステートメント

##### **SYSMOD DD**

バインダー用の出力区分データ・セット (IMS.PSBLIB) を定義します。

## **プロシージャの呼び出し**

PSBGEN プロシージャを呼び出すときは、以下の図の JCL ステートメントを使用します。

```
//PSBGEN  JOB  
//          EXEC  PROC=PSBGEN,MBR=TEMPNAME  
//C.SYSIN  DD    *
```

```
          PCB  
SENSEG   (The control statements for PSB generation)  
PSBGEN   PSBNAME=TEMPNAME  
END  
/*
```



---

## 第 2 部 IMS カタログ・ユーティリティー

IMS カタログ・ユーティリティーを使用して、IMS カタログに関するさまざまな作業を実行します。

各トピックでは、ユーティリティーがどのように機能するのかを説明し、その使用における要件と制約事項を定義しています。また、例を提供しています。



---

## 第 6 章 ACB Generation and Catalog Populate ユーティリティー (DFS3UACB)

ACB Generation and Catalog Populate ユーティリティー (DFS3UACB) を使用して、IMS.ACBLIB データ・セット内に ACB メンバーを生成し、対応するメタデータ・レコードを IMS カタログ内に作成します。また、ご使用の IMS システムで ACB を管理している場合は、結果として生じる ACB を、保留中の変更として IMS カタログのステージング・データ・セットに追加します。これらはすべて単一のジョブ・ステップで実行します。

カタログを使用して ACB を管理する IMS システムでは、ACB ライブラリーに ACB を生成する必要はありません。ただし、ツールまたはユーティリティーをまだ ACB の IMS 管理用に更新していない場合は、引き続き ACB の生成と IMS カタログへのデータの取り込みに DFS3UACB ユーティリティーを使用することができます。

IMS カタログへのデータの取り込みを ACB メンバーの生成と同じジョブ・ステップで行うと、IMS カタログと ACB ライブラリー相互の整合性を確保できます。

DFS3UACB ユーティリティーは ACB 保守ユーティリティーを呼び出して ACB メンバーを生成した後、同じジョブ・ステップで IMS Catalog Populate ユーティリティー (DFS3PU00) を呼び出して、生成された ACB メンバーに対応するレコードを IMS カタログに取り込みます。

SYSIN DD ステートメントを使用して ACB 保守ユーティリティーに制御ステートメントを指定する必要があります。ACB 保守ユーティリティーに制御ステートメントを指定しない場合、DFS3UACB は直ちに終了します。

ACB 保守ユーティリティーに対して実行パラメーターを指定することができます。これを行うには、この実行パラメーターを、DFS3UACB ユーティリティーの PARM パラメーター内の実行パラメーターとして指定します。これらのパラメーターで ACBCATWK DD ステートメントを使用している場合、POSTCOMP オプションは指定しないでください。

DFS3PPRM DD ステートメントを使用して、IMS カタログをサポートする DFSDFxxx PROCLIB メンバーの名前などの実行パラメーターを、DFS3PU00 ユーティリティーに指定する必要があります。DFS3UACB ユーティリティーは、データ取り込みフェーズの開始時に、これらの実行パラメーターを DFS3PU00 ユーティリティーに渡します。

ACB の IMS 管理が使用可能な場合は、生成された ACB をオンライン IMS システムで使用できるようにするために、SYSINP DD ステートメントで MANAGEDACBS= 制御ステートメントを指定する必要があります。DFS3PU00 ユーティリティーは、指定に応じて、ACB を後で活動化するために IMS カタログのステージング・データ・セットに追加するか、あるいは IMS システムがオフラインである場合または管理対象 ACB を初めてセットアップしている場合は、それら

を IMS ディレクトリー・データ・セットに直接追加します。IMS は、始動時にディレクトリー・データ・セットから ACB をロードします。

DFS3UACB ユーティリティーは、カタログへのデータの取り込みをロード・モードでも更新モードでも実行できます。ロード・モードを使用すると、IMS カタログ内の既存のレコードはすべて破棄されます。

ユーティリティー JCL に指定する PSB は、このユーティリティーが IMS カタログへのアクセスに使用するアクセス・モードを決定します。DFS3UACB ユーティリティー JCL には以下の PSB を指定できます。

- DFSCPL00。IMS カタログの初期ロードを実行します。
- DFSCP001。既存の IMS カタログにレコードを挿入します。
- DFSCP000。IMS カタログ・データ・セットのスペース所要量を見積もります。

サブセクション:

- 『制約事項』
- 『前提条件』
- 369 ページの『要件』
- 369 ページの『推奨事項』
- 369 ページの『入力と出力』
- 371 ページの『JCL 仕様』
- 375 ページの『ACB 生成制御ステートメント』
- 380 ページの『カタログ・データ追加制御ステートメント』
- 385 ページの『戻りコード』

## 制約事項

DFS3UACB ユーティリティーは、z/OS 制御下のスタンドアロン領域で実行されます。

DFS3UACB ユーティリティーは、EXEC= ステートメントの PARM= キーワードに指定された JCL パラメーターは受け入れません。IMS カタログへのデータの取り込みに必要な実行パラメーターを指定する場合は、代わりに DFS3PPRM DD ステートメントを使用します。

IMSplex 環境で、別々の LPAR にある複数バージョンの IMS に対して DFS3UACB ユーティリティーを実行している場合は、別々のジョブでユーティリティーを実行してください。ジョブごとに SCHENV パラメーターを使用して、各ジョブが必ず正しい LPAR に接続するようにしてください。

## 前提条件

初めて IMS カタログをロードする場合は、IMS カタログが正しく構成されていることを確認します。DFS3UACB ユーティリティーを使用して IMS カタログを初めてロードする場合は、事前に以下のステップを完了しておく必要があります。

- IMS カタログの DBD および PSB ロード・モジュールを DBD および PSB ライブラリー内に入れている。

- IMS カタログの ACB メンバーを生成し、アクティブな ACB ライブラリー内に入れている。
- IMS カタログの HALDB マスター・データベースおよび区画を、RECON データ・セット内、またはターゲット IMS カタログが DBRC によってサポートされていない場合は IMS カタログ区画定義データ・セット内で定義している。
- DFSDFxxx PROCLIB メンバーの CATALOG セクションを正しくコーディングしている。

SHARE オプションを使用して DOPT PSB をオンライン IMS カタログに追加するには、最初に Populate ユーティリティーで ACB の IMS 管理をセットアップする前に、PDSE データ・セットの拡張共用を z/OS 内で使用可能にする必要があります。詳細については、z/OS: 複数システム環境における拡張 PDSE 共用の指定を参照してください。

## 要件

IMS は、データ・セット許可に関する z/OS 規則に準拠しています。ある IMS ジョブ・ステップの許可を与える場合は、そのジョブ・ステップで使用されるすべてのライブラリーも許可する必要があります。ある IMS バッチ領域を無許可で実行するときは、IMS.SDFSRESL に許可不要ライブラリーを連結してください。

SYSIN DD ステートメントを使用して ACB 保守ユーティリティーに制御ステートメントを指定する必要があります。

DFS3UACB ユーティリティーが DFS3PU00 ユーティリティーに渡す実行パラメーターを、DFS3PPRM DD ステートメントを使用して指定する必要があります。

ACB の IMS 管理が使用可能な場合は、生成された ACB をオンライン IMS システムで使用できるようにするために、SYSINP DD ステートメントで MANAGEDACBS= 制御ステートメントを指定する必要があります。

## 推奨事項

既存の IMS カタログを更新する場合は、IMS カタログ・データ・セットのイメージ・コピーを作成することを検討してください。IMS カタログが DBRC に登録されている場合、DBRC コマンド GENJCL.IC を使用してカタログをバックアップできます。IMS カタログを IMS カタログ区画定義データ・セット内で定義した場合、標準のイメージ・コピー JCL を使用する必要があります。

ACBLIB メンバーをストレージ内に保持する必要があるため、ストレージ要件は ACBLIB メンバーの総数に大きく左右されます。異常終了 U1002 理由コード 4 が発行される原因となる GETMAIN 障害を防ぐには、ジョブの領域サイズの増加を検討してください。

## 入力と出力

DFS3UACB ユーティリティーは、以下の入力データ・セットと出力データ・セットを使用します。

必須データ・セットは以下のとおりです。

- IMS.ACBLIB データ・セット。このユーティリティーは、ACBLIB データ・セット内に ACB メンバーを作成した後、IMS カタログでのレコードの作成時にその ACBLIB データ・セットを入力として使用します。
- DBDLIB データ・セット。
- PSBLIB データ・セット。
- ACBCATWK データ・セット。ACBCATWK データ・セットは、入力データ・セットでもあり、出力データ・セットでもあります。ACB 保守ユーティリティーはこれを出力データ・セットとして使用して、生成される ACB メンバーを記録します。DFS3PU00 ユーティリティーは ACBCATWK データ・セットを入力として読み取ることで、IMS カタログへのデータ取り込み時のパフォーマンスを向上します。
- IMS.PROCLIB データ・セット。DFS3UACB ユーティリティーは IMS.PROCLIB データ・セットの DFSDFx<sup>xxx</sup> メンバーを読み取ります。
- SYSIN 制御ステートメント。
- DFS3PU00 ユーティリティーの実行パラメーターを含む入力データ・セットまたはインライン・ステートメント。DFS3UACB ユーティリティーはこのユーティリティーを内部で呼び出して、IMS カタログにデータを取り込みます。このデータ・セットまたはインライン・ステートメントは、DFS3PPRM DD ステートメントを使用して指定します。
- SYSPRINT メッセージ。

オプション・データ・セットは以下のとおりです。

- COMPCTL IEBCOPY 制御ステートメントの入力データ・セット。

DFS3UACB ユーティリティーの 1 次出力は、ACB ライブラリー・メンバーおよび IMS カタログのレコードです。このユーティリティーは ACB メンバーを非アクティブな IMS.ACBLIB データ・セットにロードします。カタログ・レコードは IMS カタログ・データ・セット (DFSCD000) に保管されます。

オプションで、DFS3UACB ユーティリティーは生成された ACB メンバーのリストを、ACBCATWK DD ステートメントによって参照されているデータ・セットに出力できます。生成された ACB メンバーのリストを維持することで、DFS3UACB ユーティリティーのデータ取り込みフェーズのパフォーマンスが大幅に向上します。

また、DFS3UACB ユーティリティーはメッセージと統計情報を SYSPRINT データ・セットに書き込みます。

DFS3UACB ユーティリティーは、SYSUT3 および SYSUT4 IEBCOPY ユーティリティー・データ・セットに出力を書き込みます。

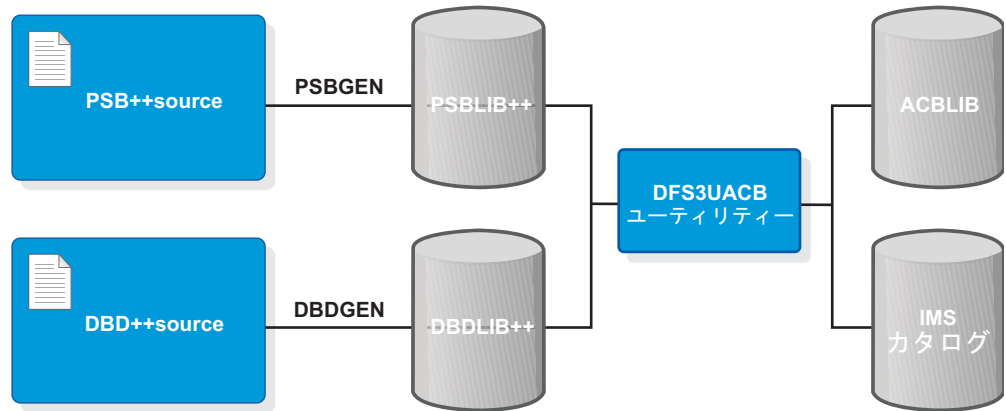


図 31. ACBGEN and Catalog Populate ユーティリティの入力と出力

## JCL 仕様

以下のサブセクションでは、DFS3UACB ユーティリティ JCL の例を 2 つ示します。1 つ目の例は、既存のカタログを更新するための JCL を示します。2 つ目の例は、IMS カタログをロードするための JCL を示します。IMS カタログをロードすると、IMS カタログの既存のレコードはすべて削除されます。

更新モードの場合の **DFS3UACB** ユーティリティ **JCL** ステートメント

以下は、DFS3UACB ユーティリティを使用して既存の IMS カタログに対して ACB メンバーとカタログ・レコードを生成するときを使用できる JCL ステートメントの例です。この JCL では、更新モードで IMS カタログにアクセスするために、DFS3PPRM DD ステートメントに PSB DFSCP001 が指定されています。

```

//ACBPOPUP JOB 'IMS SYSTEM',CLASS=K,MSGLEVEL=(1,1),REGION=0M
//*
//*****
//* DFS3UACB GENERATES ACB MEMBERS IN AN ACB LIBRARY BY CALLING THE
//* ACB MAINTENANCE UTILITY. IN THE SAME JOB STEP,
//* DFS3UACB INSERTS RECORDS IN THE EXISTING IMS CATALOG BY CALLING
//* THE IMS CATALOG POPULATE UTILITY (DFS3PU00)
//*****
//*
//ACBCATT EXEC PGM=DFS3UACB,REGION=0M
//*
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSABEND DD SYSOUT=*
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//ACBCATWK DD SPACE=(CYL,(1,1)),UNIT=SYSDA
//*
//*****
//* ACBGEN DATASETS
//*****
//IMSACB DD DSN=IMS.ACBLIB,DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4 DD UNIT=SYSDA,SPACE=(256,(100,100)),DCB=KEYLEN=30
//*****
//* ACBGEN INPUT PARMS TO UPDATE ACBLIB
//*****

```

```

//SYSIN DD *
BUILD PSB=psbname (same as needed for ACBGEN)
/*
//*****
//* POPULATE UTILITY DATASETS
//*****
//IMSACB01 DD DSN=*.IMSACB,DISP=OLD DO NOT REPLACE ASTERISK
//SYSINP DD * ISRTLIST DUPLIST /*
//DFSVSAMP DD ..... BUFFER POOL DEFINITIONS
//IEFRDRE DD ..... LOG DATASET FOR CATALOG UPDATES
//IEFRDER2 DD ..... LOG DATASET FOR CATALOG UPDATES
//*****
//* UPDATE INPUT PARMS FOR IMS CATALOG POPULATE UTILITY
//*****
//DFS3PPRM DD *
DLI,DFS3PU00,DFSCP001,,,,,,,,,Y,N,,,,,,,,,DFSDF=CAT
/*
//

```

ロード・モードの場合の **DFS3UACB** ユーティリティの **JCL** ステートメント

以下は、DFS3UACB ユーティリティを使用して ACB メンバーを生成し、カタログ・レコードを IMS カタログにロードするときに使用できる JCL ステートメントの例です。

この例では、ロード・モードで IMS カタログにアクセスするために、DFS3PPRM DD ステートメントに PSB DFSCPL00 が指定されています。

**重要:** DFS3UACB ユーティリティをロード・モードで実行すると、IMS カタログの既存のレコードはすべて削除されます。

```

//ACBPOPLD JOB 'IMS SYSTEM',CLASS=K,MSGLEVEL=(1,1),REGION=0M
/*
//*****
//* DFS3UACB GENERATES ACB MEMBERS IN AN ACB LIBRARY BY CALLING THE
//* ACB MAINTENANCE UTILITY. IN THE SAME JOB STEP,
//* DFS3UACB LOADS RECORDS IN THE IMS CATALOG BY CALLING
//* THE IMS CATALOG POPULATE UTILITY (DFS3PU00)
//*****
//*
//ACBCATT EXEC PGM=DFS3UACB,REGION=0M
/*
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSABEND DD SYSOUT=*
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//ACBCATWK DD SPACE=(CYL,(1,1)),UNIT=SYSDA
/*
//*****
//* DATASETS USED ONLY BY ACB MAINTENANCE UTILITY
//*****
//IMSACB DD DSN=IMS.ACBLIB,DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4 DD UNIT=SYSDA,SPACE=(256,(100,100)),DCB=KEYLEN=30
//*****
//* ACBGEN INPUT PARMS TO REBUILD ACBLIB
//*****
//SYSIN DD *
BUILD PSB=ALL
/*

```



```

//*****
//* DATASETS USED ONLY BY IMS CATALOG POPULATE UTILITY
//*****
//IMSACB01 DD DSN=*.IMSACB,DISP=OLD      DO NOT REPLACE ASTERISK
//DFSVSAMP DD .....      BUFFER POOL DEFINITIONS
//IEFRDER DD .....      LOG DATASET FOR CATALOG UPDATES
//IEFRDER2 DD .....      LOG DATASET FOR CATALOG UPDATES
//*****
//* LOAD INPUT PARMS FOR IMS CATALOG POPULATE UTILITY
//*****
//DFS3PPRM DD *
DLI,DFS3PU00,DFSCPL00,,,,,,,,,Y,N,,,,,,,,,DFSDF=CAT
/*
//

```

## DD ステートメント

### ACBCATWK

ACB 生成中に ACB ライブラリーに書き込まれる ACB メンバーのリストが入る、オプションの作業データ・セットを定義します。

ACBCATWK データ・セットは、ACB メンテナンス・ユーティリティの出力データ・セットおよび DFS3PU00 ユーティリティの入力データ・セットです。

ACBCATWK データ・セットは、DFS3PU00 ユーティリティのパフォーマンスを向上させるために指定します。DFS3PU00 ユーティリティは名前のリストを使用して、IMS カタログ内のどのレコードを挿入または更新する必要があるか判別します。ACBCATWK データ・セットが指定されない場合、DFS3PU00 ユーティリティは、IMSACBxx DD ステートメントで参照されている ACB ライブラリー内のすべてのメンバーを処理します。

### DFSVSAMP

バッファ・プール・パラメーター・データ・セットを定義します。

### IEFRDER DD

プライマリー IMS ログ・データ・セットを定義します。

### IEFRDER2 DD

セカンダリー IMS ログ・データ・セットを定義します。

### IMS DD

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。

### IMSACB DD

単一 ACB ライブラリー・データ・セットを定義します。

制約事項: このデータ・セットは修正されるため、他のジョブと共用することはできません。

### IMSACB01

IMS カタログにデータを追加するために使用される ACB メンバーが入った ACB ライブラリー・データ・セットを定義します。この DD ステートメントは必須です。

この DD ステートメントは、IMSACB DD ステートメントで定義されたものと同じデータ・セットを指定する必要があります。同じデータ・セットが確

実に参照されるようにするには、高位修飾子としてアスタリスクを使用してこの DD ステートメントをコーディングします (例: //IMSACB01 DD DSN=\*.ACBLIB,DISP=OLD)。

#### DFS3PPRM

DFS3PU00 ユーティリティの実行パラメーターを指定します。DFS3PPRM DD ステートメントによって指定する実行パラメーターには、以下の値が含まれます。

- 使用する DFS3PU00 ユーティリティの PSB
- DBRC を有効にするかどうか
- IRLM を有効にするかどうか
- IMS カタログ属性を含む DFSDFxxx PROCLIB メンバーの名前

DFS3PPRM DD ステートメントを省略すると、DFS3UACB ユーティリティは以下のデフォルト実行パラメーターを DFS3PU00 ユーティリティに渡します。

```
DLI,DFS3PU00,DFSCP001,,,,,,,,,,,,,Y,N,,,,,,,,,,,,,'DFSDF=CAT'
```

上記のデフォルト・パラメーターでは、DBRC を有効にし、IRLM は有効にせずに更新モードで DFS3PU00 ユーティリティが実行されます。このデフォルト値は、DFSDFxxx PROCLIB メンバーとして DFSDFCAT を指定しています。

IMS カタログをロードするには、DFS3PPRM DD ステートメントで定義されている実行パラメーターに DFSCPL00 PSB を指定する必要があります。

**重要:** DFS3PU00 ユーティリティの実行パラメーターに DFSCPL00 PSB を指定すると、ロード・プロセスの開始前に既存の IMS カタログはすべて削除されます。

IMS カタログでのデータ共有を有効にする場合、DFS3PPRM DD ステートメントに IRLM サポートを指定する必要があります。それには、IRLM サポートの位置に Y を指定し、その後の位置に IRLM ID を指定します。

DFS3PPRM DD ステートメントの次の例は、ロードの PSB DFSCPL00、IRLM サポートなし、DFSDF001 という名前の DFSDFxxx メンバーを指定しています。

```
//DFS3PPRM DD *  
DLI,DFS3PU00,DFSCPL00,,,,,,,,,,,,,Y,N,,,,,,,,,,,,,DFSDF=001
```

前の例と対照的に、DFS3PPRM DD ステートメントの次の例は、更新の PSB DFSCP001、IRLM IRL1 によるサポート、DFSDF002 という名前の DFSDFxxx メンバーを指定しています。

```
//DFS3PPRM DD *  
DLI,DFS3PU00,DFSCP001,,,,,,,,,,,,,Y,Y,IRL1,,,,,,,,,,,,,DFSDF=002
```

#### PROCLIB DD

ユーティリティに必要な、IMS カタログの各種属性を定義する DFSDFxxx メンバーが入った IMS.PROCLIB データ・セットを定義します。

#### SYSABEND DD

ダンプ・データ・セットを定義します。

### **SYSIN DD**

入力制御ステートメント・データ・セットを定義します。これらのデータ・セットは、テープ・ボリューム、直接アクセス装置、またはカード読取装置に置くか、入力ストリームで経路指定することができます。入力は、80 の倍数でブロック化できます。実行時に、このユーティリティーは必要な数の制御ステートメントを処理できます。

### **SYSINP DD**

80 文字の固定長レコードを持つオプションの制御ステートメント順次データ・セット。位置 1 から 72 までの文字だけが読み取られます。

1 つ以上のレコードで、制御ステートメント・パラメーターをブランクまたはコンマで区切って指定することができます。

SYSINP DD ステートメントで指定できる制御ステートメントの説明については、380 ページの『カタログ・データ追加制御ステートメント』を参照してください。

### **SYSPRINT DD**

出力メッセージ・データ・セットを定義します。

SYSPRINT DD ステートメントで DASD データ・セットまたはテープ・データ・セットを参照する場合、そのデータ・セットのブロック・サイズを **DCB** パラメーターの **BLKSIZE** サブパラメーターで制御できます。BLKSIZE 値を指定する場合、その値は正確に 121 の倍数とする必要があります、そうでない場合はシステム ABEND013-20 が発生する可能性があります。DASD データ・セットの BLKSIZE を省略すると、システムが決定するブロック・サイズが使用されます。LRECL パラメーターに指定される値に関係なく、ユーティリティーは常にレコード長 121 を使用します。

### **SYSUT3 DD**

PRECOMP または POSTCOMP のいずれかが EXEC ステートメントで指定されている場合に必要な作業データ・セットを定義します。

### **SYSUT4 DD**

機能は SYSUT3 と同じです。

## **ACB 生成制御ステートメント**

DFS3UACB ユーティリティーの ACB 生成フェーズでは、ユーティリティー JCL に制御ステートメントを指定して、ACB メンバーに対して実行するアクションを指定します。BUILD ステートメントと DELETE ステートメントを指定できます。

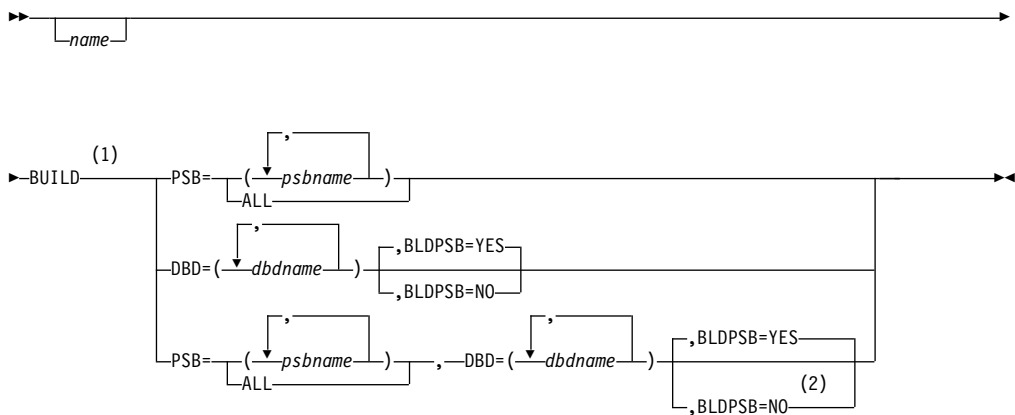
BUILD ステートメントでは、DFS3UACB は指定された ACB メンバーを作成し、対応するレコードを IMS カタログにロードおよび挿入します。DELETE ステートメントでは、DFS3UACB ユーティリティーは ACB ライブラリー・データ・セットの ACB メンバーのみを削除します。IMS カタログのレコードは削除されません。IMS カタログからのレコードの除去は、保存ポリシーによって制御されます。

制御ステートメントは以下のガイドラインに準拠している必要があります。

- ステートメントは 1 から 71 桁に収め、カード・イメージでコーディングします。

- 制御ステートメントには、任意に名前 (1 桁目から開始) を含めることができます。
- ステートメントを継続するためには、72 桁目に非ブランク文字を入力し、次の行の 16 桁目からステートメントを開始します。
- 命令フィールドの前後には、1 つ以上のブランクを入れなければなりません。
- パラメーターは、1 つ以上の PSB 名または DBD 名から構成され、その前後にも 1 つ以上のブランクを入れなければなりません。
- コンマ、括弧、ブランクは、区切り文字としてのみ使用できます。
- コメントを制御ステートメントの最後のパラメーターの後に書くことができます。パラメーターとコメントは 1 つ以上のブランクで分離します。

**DFS3UACB** ユーティリティの制御構文: **BUILD** の形式



注:

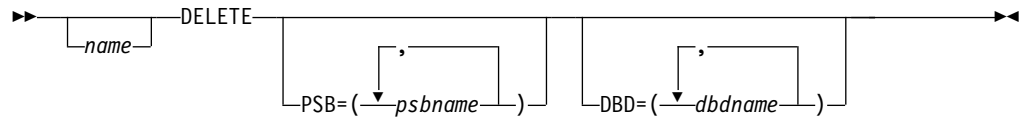
- 1 ACB メンテナンス・ユーティリティの SYSIN 入力制御ステートメントに、先入れ先出し (FIFO) プロセスはありません。同じアプリケーション制御ブロック (ACB) 生成ジョブの SYSIN 制御ステートメントで、BUILD PSB= と BUILD DBD= の両方のパラメーターを指定すると、DBD= オペランドが先にブロック・ビルダー・ユーティリティ・プログラムに渡されます。SYSIN 制御ステートメントのどこに DBD= オペランドが入力されたかに関係なく、ACBLIB データ・セットで DBD の準備ができていない場合、DFS0586I が出されます。
- 2 同じステートメント内にパラメーター PSB=ALL と BLDPSB=NO を指定した場合、IMS はすべての PSB を作成します (BLDPSB=NO は無視されます)。同様に、同じ ACBGEN ジョブで 1 つの DBD に BLDPSB=NO パラメーターを指定し、別の DBD に BLDPSB=YES パラメーターを指定した場合、IMS は変更された DBD を参照するすべての PSB を作成し、BLDPSB=NO の指定を無視します。

次の例では、CUSTOMER および ORDER DBD に関連付けられているすべての PSB は、CUSTOMER DBD に BLDPSB=NO が指定されている場合でも再作成されます。

```
BUILD DBD=(CUSTOMER),BLDPSB=NO
BUILD DBD=(ORDER),BLDPSB=YES
```

## ACB 保守ユーティリティー構文: DELETE 形式

DELETE ステートメントは ACB ライブラリーから ACB メンバーを除去しますが、対応するレコードを IMS カタログから除去することはありません。



## ACB 保守ユーティリティー・パラメーター

### BUILD

指定した PSB (指定した DBD を指す) 用にブロックを作成することを指定します。

### DELETE

ブロックを ACBLIB データ・セットから削除することを指定します。指定した PSB、および指定した DBD を指すすべての PSB が削除されます。

ACBLIB データ・セットからブロックを削除しても、IMS カタログ内の対応するレコードは削除されません。

### PSB=ALL

現在 IMS.PSBLIB 内にあるすべての PSB 用に、そのブロックを作成することを指定します。このパラメーターは、初期 IMS.ACBLIB を作成するために使用します。PSB=ALL パラメーターを指定すると、すべての PSB および DBD (および他のモジュール) は ACBLIB データ・セットから削除され、それらのスペースは再利用できるようになります。その後、ACB 生成が PSBLIB データ・セット内のすべての PSB に関して実行されます。このパラメーターを DELETE ステートメントと一緒に使用しないでください。

制約事項: BUILD PSB=ALL パラメーターを SYSIN 制御ステートメントに指定する場合には、すべての PSB を単一の PSBLIB データ・セット内に置く必要があります。連結された PSBLIB は、IMS DD ステートメントでは認識されません。

### PSB=(psbname)

この制御ステートメントで指定されているすべての PSB について、ブロックを作成または削除することを指定します。このタイプの制御ステートメントは必要に応じて実行要求することができます。このパラメーターは、新しい PSB を IMS.ACBLIB に追加したり、使用しなくなった PSB を削除したりします。単一のパラメーターを与える場合には、括弧を省略することができます。

### DBD=(dbdname)

この DBD について、およびこの DBD を直接または論理関係によって間接的に参照するすべての PSB について、ブロックを作成または削除することを指定します。作成する DBD は、既に IMS.ACBLIB データ・セット内に存在している必要があります。参照する PSB も、既に IMS.ACBLIB データ・セット内に存在している必要があります。IMS.PSBLIB データ・セットに新規に追加された PSB は、PSB オペランドによって参照される必要があります。PSB を削除しても、その PSB が参照している DBD は削除されないため、このパラメーターを使用して特定の DBD を削除することができます。ただし、DBD を削除または

作成すると、指定した DBD を参照する IMS.ACBLIB データ・セット内のすべての PSB が、要求タイプに基づいて再作成または削除されることになります。単一のパラメーターを与える場合には、括弧を省略することができます。

例 1: PSB-a で DBD-a および DBD-b を参照します。DBDGEN が DBD-a および DBD-b に関して行われ、更新済み DBD は DBDLIB にあります (しかし、ACBLIB にはまだありません)。DBD-a を ACB 生成で指定することによって、DBD-a が ACBLIB 内に再ビルドされ、参照 PSB (この場合は、PSB-a) があれば、いずれもやはり再ビルドされます。確かに PSB-a は再ビルドされはしても、DBD-b が明確に ACBLIB 内に再ビルドされたわけではないため、ACBLIB は使用不能です。DBD-b を ACBLIB 内に再ビルドするためには、ACB 生成でそれを明示的に指定する必要があります。たとえ参照 PSB が完全に更新されている場合でも、やはり更新済み DBD を ACB 生成で明示的に指定する必要があります。

このプログラムが処理するすべての PSB は、IMS.ACBLIB データ・セット内にメンバーを生成します。PSB が参照する DBD は、特定の DBD が初めて処理されたとき、あるいはある DBD 名が制御ステートメントに現れたときに、メンバーを生成します。同じ DBD を参照するすべての PSB は、参照されている DBD に PSB を接続するために、ディレクトリー項目内の情報を伝えます。

論理 DBD は IMS.ACBLIB 内にメンバーを持っておらず、BUILD または DELETE 制御ステートメントで参照することはできません。

例 2: 以下の例では、BLDPSB パラメーターの使用方法を示します。

- CUSTOMER という名前の DBD が変更されたため、CUSTOMER を参照するすべての PSB を再作成する必要があります。

```
BUILD DBD=CUSTOMER,BLDPSB=YES
```

- ORDER および INVENTORY という名前の DBD が変更されたため、これらの DBD を参照するすべての PSB を再作成する必要があります。

```
BUILD DBD=(ORDER,INVENTORY),BLDPSB=YES
```

DBD を IMS.DBDLIB 内で置換する場合には、それを BUILD DBD 制御ステートメントの中にも含める必要があります。この方法のみが BUILD PSB=ALL を行わずに、IMS.ACBLIB 内で DBD を置換できる有効な方法です。

DBDLIB の修正 DBD を参照する BUILD PSB を実行すると、ACBLIB 内で置換された PSB は、更新版の DBD を含むことになります。この BUILD PSB が、変更された DBD についての BUILD DBD の前にくると、ACBLIB は、別の版の DBD を持つ PSB を含むことになります。BUILD PSB に指定された PSB は更新された DBD を含みますが、作成されていない PSB は古い DBD を参照することになります。ACBLIB 上の PSB 用の DBD が、アクセスされるデータベースと一致しない場合には、結果は予測できません。(例えば、変更された DBD でセグメント・コードが追加されるか、削除されたために U852 異常終了が起きます。) したがって、後で使用するために DBDGEN を実行する場合は、データベースに変更が反映されない限り、変更された DBD を参照する PSB を作成してはなりません。

物理 DBD が変更され、しかもそれが BUILD DBD ステートメントで参照されている場合は、変更された DBD (1 次索引と副次索引を含めて) と論理的に関

連するすべての物理 DBD も BUILD DBD ステートメントで参照する必要があります。しかし、それらの DBD と論理的に関連する DBD を再作成する必要はありません。

次の図に、一部の物理データベース間の関係が図示してあります。ここで A が変更された DBD です。次のような関係が存在しています。

- B と C は、A と論理的に関連しています。
- D は B と論理的に関連しています。
- E は C と論理的に関連しています。
- D と E は、A と論理的に関連していないので、BUILD DBD ステートメントで参照されていません。

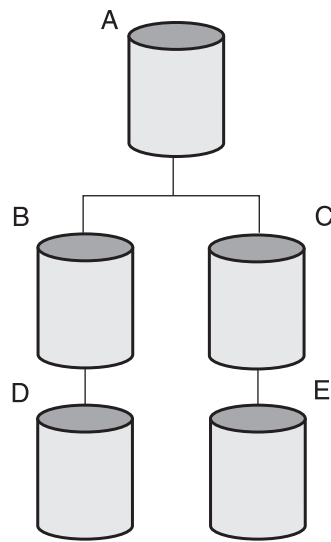


図 32. 論理的に関連した物理データベースの例

#### **BLDPSB=**YES | NO

BUILD DBD=(*dbdname*) ステートメント内で、変更された DBD を参照するすべての PSB を ACBGEN で再作成するかどうかを指定します。

#### **YES**

BUILD DBD=(*dbdname*) ステートメント上で、変更された DBD を参照するすべての PSB を ACBGEN で再作成するよう指示します。デフォルトは BLDPSB=YES です。

**NO** 変更された DBD がデータベースの物理構造を変更していない場合、その DBD を参照する PSB を ACBGEN で再作成しないよう指示します。高速機能 DEDB の場合、PSB が再作成されるのは、セグメント数またはデータベースのセグメント内のフィールド数あるいはその両方が変更されるときに限られます。高速機能 MSDB の場合、データベースの物理構造が変更されても、参照する PSB は再作成されません。

## カタログ・データ追加制御ステートメント

SYSINP DD ステートメントを使用して、1 つ以上のレコードで、以下の制御ステートメント・パラメーターをブランクまたはコンマで区切って指定することができます。

### DUPLIST

既に IMS カタログにあるインスタンスと重複するために IMS カタログに追加されない、入力 ACB ライブラリー内の各 DBD リソースまたは PSB リソースを DFS3PU00 ユーティリティでリストすることを指定します。IMS カタログ内のリソースの重複するインスタンスごとに、ユーティリティはメッセージ DFS4436I を出力します。

MANAGEDACBS= ステートメントが指定されており、ACB の IMS 管理が使用可能な場合、ユーティリティは、既に IMS ディレクトリーにあるリソースのインスタンスと重複するために IMS ディレクトリーまたはステージング・データ・セットに追加されない、各 DBD リソースまたは PSB リソースもリストします。

MANAGEDACBS= ステートメントで UPDATE パラメーターが指定された場合、ユーティリティは重複するインスタンスごとにメッセージ DFS4531I を出力します。

UPDATE パラメーターで LATEST が指定されたかデフォルトとして受け入れられた場合、ユーティリティは、IMS ディレクトリー内のリソース・インスタンスのタイム・スタンプがより後の時刻であるために IMS ディレクトリーに追加されないインスタンスごとに、メッセージ DFS4522I も出力します。

MANAGEDACBS= ステートメントで STAGE パラメーターが指定された場合、ユーティリティは、重複するインスタンスであるためにステージング・データ・セットにコピーされないリソースごとに、メッセージ DFS4542I を出力します。

STAGE パラメーターで LATEST が指定されたかデフォルトとして受け入れられた場合、ユーティリティは、IMS ディレクトリー内のリソース・インスタンスのタイム・スタンプがより後の時刻であるためにステージング・データ・セットに追加されないインスタンスごとに、メッセージ DFS4539I も出力します。

### ERRORMAX=*n*

*n* 個を超えるメッセージが、特定の DBD および PSB が IMS カタログに書き込まれるメタデータを持つことを妨げるエラーを示す場合、IMS Catalog Populate ユーティリティを終了します。メタデータの重複するインスタンスは、この制限にカウントされません。このオプションを省略する場合、制限はありません。

### RESOURCE\_CHKP\_FREQ=*n*

チェックポイントの間に挿入する DBD および PSB リソース・インスタンスの数を指定します。*n* は、1 から 99999999 までの 1 桁から 8 桁の数値です。デフォルト値は 100 です。

### SEGMENT\_CHKP\_FREQ=*n*

チェックポイントの間に挿入するセグメントの数を指定します。この数に達すると、IMS はチェックポイントを発行する前に、現在処理されているリソース・インスタンスのセグメントすべての挿入を完了します。*n* は、1 から 99999999 までの 1 桁から 8 桁の数値です。デフォルト値は 1000 です。



注: 最初のチェックポイント頻度数に達すると、チェックポイントが発行され、カウンターは 0 にリセットされます。

## ISRTLIST

ACB の IMS 管理が使用可能の場合、ユーティリティーは、IMS ディレクトリーに追加されるか、後で IMS ディレクトリーにインポートするためにステージング・データ・セットに保管される各 DBD リソースまたは PSB リソースもリストします。

ユーティリティーは、以下のメッセージを出力することにより、各リソースに対して実行されるアクションを識別します。

### DFS4520I

リソースは、新規リソースとして IMS ディレクトリーに追加されました。

### DFS4521I

リソースのこのインスタンスは、IMS ディレクトリー内のリソースの既存インスタンスに取って代わりました。

### DFS4537I

リソースは、ステージング・データ・セットへコピーされました。リソースは、後で IMS ディレクトリーにインポートされる時、IMS ディレクトリーに新規リソースとして挿入されます。

### DFS4538I

リソースは、ステージング・データ・セットへコピーされました。リソースは、後で IMS ディレクトリーにインポートされる時、IMS ディレクトリー内のリソースの既存インスタンスに取って代わります。

## MANAGEDACBS=

MANAGEDACBS 制御ステートメントは、以下のアクションを実行するために使用します。

- データベースおよびプログラム・ビューのランタイム・アプリケーション制御ブロック (ACB) を管理するために IMS をセットアップします。
- 新規 ACB または ACB ライブラリー・データ・セットから変更した ACB を使用して、ACB を管理する IMS システムを更新します。
- ACB を管理する IMS システムに後でインポートするために、ACB ライブラリーからステージング・データ・セットに ACB を保管します。

MANAGEDACBS ステートメントは、以下の構文図に従って指定することができます。

MANAGEDACBS= ステートメントで複数のパラメーターを指定する場合に限り、括弧が必要です。

以下のリストでは、MANAGEDACBS= ステートメントで指定できるパラメーターについて説明します。

## SETUP

アプリケーション制御ブロック (ACB) を管理するために IMS に必要な IMS ディレクトリー・データ・セットを作成します。IMS ディレクトリー・データ・セットのすべての既存インスタンスは置き換えられます。

指定した MANAGEDACBS=SETUP でユーティリティーが正常に実行されるまで、DFSDFxxx PROCLIB メンバーの CATALOG セクションに ACBMGMT=CATALOG を指定しないでください。ACB の IMS 管理を使用可能にする前に、IMS ディレクトリーを作成して、アクティブ ACB をロードしておく必要があります。

SETUP が指定された場合、ユーティリティーは、入力 ACB ライブラリー内にある ACB を IMS ディレクトリー・データ・セットに挿入します。

ユーティリティー JCL で IMS カタログ PSB DFSCP001 が指定された場合、ユーティリティーは、新規または変更されたすべての DBD および PSB を IMS カタログに挿入します。

ユーティリティー JCL で IMS カタログ PSB DFSCPL00 が指定された場合、ユーティリティーは既存のすべてのカタログ・レコードを削除し、IMS カタログを再ロードします。

ユーティリティーは、IMS ディレクトリーに ACB を追加するとき、IMS カタログ内の対応する DBD インスタンスおよび PSB インスタンスにアクティブとしてのフラグを立てます。

## STAGE

ACB を入力 ACB ライブラリーからステージング・データ・セットに保管します。ステージング・データ・セットに保管された ACB は、IMPORT DEFN SOURCE(CATALOG) コマンドで IMS システムに追加されるまで活性化されません。

STAGE を指定するときは、ユーティリティー JCL で IMS カタログ PSB DFSCP001 を指定する必要があります。

MANAGEDACBS=STAGE 制御ステートメントが指定されたときにステージング・データ・セットが存在する場合、ユーティリティーは既存のステージング・データ・セットを使用します。既にステージング・データ・セット内にある ACB はすべて保持されます。新規リソースをステージングするときには、ステージング・データ・セットをスクラッチして再作成するには、MANAGEDACBS=(STAGE,DELETE) を指定します。DELETE が指定される場合、既存のデータ・セット内にある ACB はすべて失われます。

STAGE が指定された場合、DFS3PU00 ユーティリティーは IMS システム内にまだ存在しない、ACB ライブラリー内のすべての ACB を、ステージング・データ・セットにコピーします。ACB ライブラリー内の ACB が既に IMS システム内に存在する場合、ユーティリティーは、その ACB のタイム・スタンプが UNCOND パラメーターや LATEST パラメーターの基準を満たすかどうかに基づいて、ACB をステージング・データ・セットに保管します。

IMS カタログの初期ロードを行うとき、または ACB の IMS 管理を使用可能にする前に、STAGE を指定しないでください。STAGE パラメーターは、既存の IMS ディレクトリー・データ・セットに対する更新のステージング専用です。

MANAGEDACBS= の後に複数のパラメーターを指定する場合には、括弧で囲む必要があります。例: (STAGE,LATEST)。STAGE のみを指定する場合は、括弧を省略することができます。

## DELETE

ステージング・データ・セットがどのオンライン IMS システムにも割り振られていない場合は、リソースをステージング・データ・セットに追加する前に、ステージング・データ・セットをスクラッチして再作成します。既存のステージング・データ・セット内にある ACB はすべて失われます。

## LATEST

IMS システム内に ACB が既に存在する場合、ACB ライブラリー内のインスタンスのタイム・スタンプが IMS システム内の ACB より新しい場合を除き、ACB ライブラリー内の ACB のインスタンスをステージング・データ・セットに保管しません。

これはデフォルトです。

## UNCOND

IMS システム内に ACB が既に存在する場合、ACB ライブラリー内の ACB のタイム・スタンプが IMS システム内の ACB のタイム・スタンプと同じである場合を除き、ACB ライブラリー内の ACB のインスタンスをステージング・データ・セットに無条件に保管します。

## UPDATE

既存の IMS ディレクトリー・システム・データ・セットを排他モードで直接更新します。ACB はステージング・データ・セットに入れられません。

推奨事項: UPDATE パラメーターを指定する場合、IMS ディレクトリーを使用する IMS システムをすべてシャットダウンしてください。UPDATE が指定される場合、IMS Catalog Populate ユーティリティには IMS ディレクトリーに対する排他的アクセスが必要です。

また、ユーティリティは、IMS ディレクトリーの更新時にオンライン IMS システムに通知しません。そのため、更新された ACB をロードするために、ユーティリティが IMS ディレクトリーを更新するときにオンラインになっている IMS システムを再始動する必要があります。

UPDATE が指定された場合、DFS3PU00 ユーティリティは、まだ IMS システム内にはない ACB を IMS ディレクトリーに無条件に挿入します。IMS システム内に ACB が存在する場合、その ACB のインスタンスは、ACB ライブラリー内のインスタンスのタイム・スタンプ、および UNCOND パラメーターまたは LATEST パラメーターの指定に従って置き換えられます。

IMS カタログの初期ロードを行うとき、または ACB の IMS 管理を使用可能にする前に、UPDATE を指定しないでください。UPDATE パラメーターは、既存の IMS ディレクトリー・データ・セットの更新専用です。

UPDATE を指定するときは、ユーティリティ JCL で IMS カタログ PSB DFSCP001 を指定する必要があります。

MANAGEDACBS の後に複数のパラメーターを指定する場合には、括弧で囲む必要があります。例: (UPDATE,LATEST)。UPDATE のみを指定する場合は、括弧を省略することができます。

### LATEST

IMS システム内に ACB が既に存在する場合、その ACB は、ACB ライブラリー内のインスタンスのタイム・スタンプが IMS システム内の ACB より新しい場合を除き、ACB ライブラリー内の ACB のインスタンスで置き換えられません。

これはデフォルトです。

### UNCOND

IMS システム内に ACB が既に存在する場合、その ACB は、ACB ライブラリー内の ACB のタイム・スタンプが IMS システム内の ACB のタイム・スタンプと同じである場合を除き、ACB ライブラリー内の ACB のインスタンスで無条件に置き換えられます。

### SHARE

動的オプション (DOPT) PSB の場合のみ、共用モードに必要な IMS ディレクトリー・データ・セットを割り振ることにより、オンライン処理を中断せずに DOPT PSB を IMS カタログに追加できるようにします。

ユーティリティーが IMS カタログをロードする場合は、SHARE を指定しないでください。これが該当するのは、ユーティリティー JCL の実行パラメーターに DFSCPL00 (IMS カタログのロード PSB) の指定がある場合です。

入力 ACBLIB データ・セット内に DOPT PSB ではないリソースが存在する場合は、SHARE を指定しないでください。

SHARE オプションを指定してユーティリティーを実行する前に、z/OS システム内で PDSE データ・セットの拡張共用を使用可能にする必要があります。z/OS 内で PDSE データ・セットの拡張共用を使用可能にするには、シスプレックス内の各システム上で、SYS1.PARMLIB 内の IGDSMSxx メンバーに PDESHPARING(EXTENDED) を指定します。SHARE の指定時に拡張 PDSE 共用が使用可能になっていない場合、ユーティリティーは必要な IMS カタログ・データ・セットを排他モードで割り振ります。これが原因で、他の IMS プロセスおよびアプリケーション・プログラムとの競合が発生する場合があります。

拡張 PDSE 共用の使用可能化について詳しくは、z/OS: 複数システム環境における拡張 PDSE 共用の指定を参照してください。

デフォルトでは、ACB の IMS 管理のために割り振られるシステム・データ・セット名は、IMS カタログの最初の区画が定義されるときに RECON データ・セットに指定されたデータ・セット名の接頭部から取られます。このデフォルトをオーバーライドするには、DFSDFxxx PROCLIB メンバーの CATALOG セクションの SYSDSNHLQ= ステートメントまたはカタログ定義出口ルーチン (DFS3CDX0) で HLQ 名の接頭部を指定できます。

### NODUPLIST

追加されなかったリソース・インスタンスのリストを出力しません。このパラメーターがデフォルトです。

### NOISRTLIST


挿入されたリソース・インスタンスのリストを出力しません。

## 戻りコード


DFS3UACB ユーティリティーは以下のコードを返します。

- 0       すべての操作が正常に終了
- 4       1 つ以上のレコードが IMS カタログにロードされませんでした。警告メッセージが発行されました。
- 16      重大エラーでプログラムが中止された

関連タスク:

 ACB Generation and Catalog Populate ユーティリティー (DFS3UACB) を使用した IMS カタログへのデータの追加 (システム定義)

関連資料:

 IMS PROCLIB データ・セットの DFSDFxxx メンバー (システム定義)  
3 ページの『第 1 章 アプリケーション制御ブロック保守ユーティリティー』



---

## 第 7 章 IMS Catalog Alias Names ユーティリティー (DFS3ALI0)

IMS Catalog Alias Names ユーティリティー (DFS3ALI0) を使用して、1 つの IMSplex 環境内で複数のカタログを定義します。

### 概説

このユーティリティーを使用して、カタログ別名の全リストを IMS に対して定義します。IMS がユーザー・データベース PCB のためにカタログ名の変換を実行できるように、この定義リストが必要になります。

カタログ・データベースに標準の接頭部 DFSC が使用されている場合、このユーティリティーを使用する必要はありません。

### 制約事項

このユーティリティーについて文書化されている制約事項はありません。

### 前提条件

カタログ別名の名前を使用するには、事前に、IMS システムの始動 JCL で使用されている IMS.PROCLIB データ・セットの DFSDFxxx メンバーを、CATALOG セクションを含めて構成する必要があります。

### 要件

ご使用の IMS システムが IMS カタログを使用している必要があります。

### 入力と出力

このユーティリティーは IMS カタログ別名の接頭部 (それぞれ 4 文字ずつ) のリストを受け取って、それらの名前をシステム DBD ライブラリーに追加します。

### JCL 仕様

**EXEC** ステートメント

```
//ALIAS EXEC PGM=DFS3ALI0
```

**DD** ステートメント

#### **JOBLIB/STEPLIB**

IMS.SDFSRESL データ・セットの位置。ここには、このユーティリティーの実行可能モジュールが含まれています。

#### **DFSRESLB**

IMS.SDFSRESL データ・セットの位置。ここには、このユーティリティーの実行可能モジュールが含まれています。また、カタログ DFSCD000 DBD も含まれています。

### **SYSPRINT**

このユーティリティによって生成されたメッセージを受け取るデータ・セット。 **SYSPRINT** データ・セットの DCB パラメーターは RECFM=FB および LRECL=121 です。

### **SYSLMOD**

別名の追加先となる DBD ライブラリーの位置。



### **SYSIN**

このユーティリティの入力パラメーターを含むデータ・セット。カタログ別名の接頭部の全リストを含みます。複数の項目はコンマで区切ります。


## サンプル **JCL** ステートメント

```
//ALIAS EXEC PGM=DFS3ALIO
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=IMS.DBDLIB,DISP=OLD
//SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
prefix1,prefix2,...
/*
```

関連タスク:

-  IMS カタログ別名の定義 (システム定義)
-  IMS カタログの共用 (システム定義)

関連資料:

-  IMS PROCLIB データ・セットの DFSDFxxx メンバー (システム定義)



---

## 第 8 章 IMS Catalog Copy ユーティリティ (DFS3CCE0、DFS3CCI0)

IMS Catalog Copy ユーティリティを使用して、IMS カタログのコピーを作成し、さらにオプションで、IMS カタログへのデータの取り込みに使用された ACB、DBD、および PSB ライブラリー・データ・セットのコピーを作成します。


ACB の IMS 管理が使用可能な場合は、IMS Catalog Copy ユーティリティを使用して、IMS カタログによって管理されるアクティブ ACB をコピーすることもできます。

IMS Catalog Copy ユーティリティは、テスト環境から実稼働環境への IMS カタログ・データベースのマイグレーションや、災害復旧時における実稼働環境から別のインストール済み環境への IMS カタログ・データベースのコピーといった目的に使用できます。


IMS Catalog Copy ユーティリティには、IMS Catalog Export ユーティリティ (DFS3CCE0) と IMS Catalog Import ユーティリティ (DFS3CCI0) が含まれています。

Catalog Copy ユーティリティはバッチ領域内でも、オンライン BMP としても実行できます。

関連概念:

 IMS カタログ・データ・セット (システム定義)

関連タスク:

 IMS カタログのコピー (システム定義)

関連資料:

425 ページの『第 11 章 IMS Catalog Populate ユーティリティ (DFS3PU00)』

 z/OS: IEBCOPY (ライブラリー・コピー) プログラム

---

## IMS Catalog Export ユーティリティ (DFS3CCE0)

IMS Catalog Export ユーティリティは、IMS カタログとそこに含まれる ACB、DBD、および PSB ライブラリーを、同一ジョブ・ステップでエクスポート・データ・セットにコピーします。

IMS 管理の ACB 環境では、IMS Catalog Export ユーティリティで、IMS カタログによって管理されるアクティブ ACB をエクスポートすることもできます。

IMS Catalog Export ユーティリティは、ACB、DBD、および PSB ライブラリー・データ・セットをコピーするために z/OS データ・セット・ユーティリティ

IEBCOPY を呼び出します。IEBCOPY では入力データ・セットの連結はサポートされません。そのため、1 つのジョブ・ステップでは各タイプのライブラリーの 1 つしかコピーできません。

IMS Catalog Export ユーティリティーは、IMS カタログをエクスポート・データ・セットにコピーしたときにエクスポート統計報告書を生成します。IEBCOPY ユーティリティーは、それ自体の統計に関する報告書を作成します。

IMS Catalog Export ユーティリティーはバッチ領域内でも、オンライン BMP としても実行できます。

IMS Catalog Export ユーティリティーは、DBD と PSB のすべてのレコードを IMS カタログにコピーします。ただし、エクスポート JCL の IMSACB DD ステートメントで ACB ライブラリーが指定されている場合、IMS Catalog Export ユーティリティーは、指定された ACB ライブラリー内の対応するメンバーのタイム・スタンプと一致するタイム・スタンプを持つ DBD または PSB レコード内のセグメントのみをコピーします。IMSACB DD ステートメントを省略すると、ユーティリティーは IMS カタログ内の各 DBD または PSB レコードのセグメントの、すべてのタイム・スタンプのバージョンをコピーします。

IMSACB DD ステートメントで ACB ライブラリーが指定されていても、IMS カタログ内の特定の DBD または PSB レコードに ACB メンバーが見つからない場合には、ユーティリティーは警告メッセージを出し、レコード内のセグメントのすべてのタイム・スタンプのバージョンをコピーします。

サブセクション:

- 『制約事項』
- 391 ページの『前提条件』
- 391 ページの『要件』
- 391 ページの『推奨事項』
- 391 ページの『入力と出力』
- 393 ページの『エクスポート JCL の仕様』
- 395 ページの『Export DD ステートメントの説明』
- 397 ページの『エクスポート統計報告書』
- 399 ページの『ライブラリー統計報告書』
- 399 ページの『戻りコード』

## 制約事項

IMS Catalog Export ユーティリティーは、システム・サービス・ユーティリティー IEBCOPY を使用して ACB、PSB、および DBD ライブラリーをコピーします。IEBCOPY ユーティリティーは入力データ・セットの連結をサポートしないため、IMS Catalog Export ユーティリティーは単一のジョブ・ステップで、各タイプのライブラリーごとにデータ・セットを 1 つしかコピーできません。

## 前提条件

IMS Catalog Export ユーティリティに固有の前提条件は、現在文書化されていません。

## 要件

IMS Catalog Export ユーティリティに固有の要件は、現在文書化されていません。

## 推奨事項

IMS Catalog Export ユーティリティに固有の推奨事項は、現在文書化されていません。

## 入力と出力

IMS カタログをコピーする場合、IMS Catalog Export ユーティリティ (DFS3CCE0) の 1 次入力は IMS カタログです。このユーティリティは IMS カタログ PSB DFSCP000 を使用して IMS カタログ・データベースを読み取ります。

IMS カタログのコピー中、IMS Catalog Export ユーティリティは IMS カタログへのアクセスに使用される PSB および DBD ライブラリーを読み取る必要があります。この目的で、その PSB および DBD ライブラリーを参照するために //IMS DD ステートメントが必要です。入力 IMS カタログ・データ・セットには DD ステートメントは不要です。

コピーする IMS カタログのカタログ・レコードに DBD および PSB メンバーの複数のタイム・スタンプ・バージョンが含まれる場合、IMSACB DD ステートメントに ACB ライブラリー・データ・セットを指定して、カタログ内の各 DBD または PSB メンバーのうち、対応する ACB ライブラリー・メンバーのタイム・スタンプと一致するタイム・スタンプ・バージョンのみをコピーすることができます。

IMS Catalog Export ユーティリティは IMSACB DD ステートメントを使用して、IMS カタログからコピーされる DBD および PSB メンバー・セグメントのタイム・スタンプ・バージョンの数を制限します。

ACB、DBD、および PSB ライブラリーをコピーするには、以下の DD ステートメントを使用して、それぞれのデータ・セットを入力として指定する必要があります。

- CCUACB DD ステートメント (ACB ライブラリー・データ・セットの場合)
- CCUDBD DD ステートメント (DBD ライブラリー・データ・セットの場合)
- CCUPSB DD ステートメント (PSB ライブラリー・データ・セットの場合)

コピー処理の整合性を維持するには、CCUACB、CCUDBD、および CCUPSB の各 DD ステートメントによって記述されるデータ・セットが、IMS および IMSACB DD ステートメントによって記述されるデータ・セットと同じでなければなりません。

IMS システムのアクティブ ACB が ACB ライブラリーを使用して管理されている場合は、IMS カタログ、ACB ライブラリー、DBD ライブラリー、および PSB ラ

イブラリーのエクスポートされたコピーの出力データ・セットを、以下の DD ステートメントを使用して JCL に指定する必要があります。

- CCUCATEX DD ステートメント (IMS カタログのエクスポート・データ・セットの場合)
- CCUACBEX DD ステートメント (ACB ライブラリーのエクスポート・データ・セットの場合)
- CCUDBDEX DD ステートメント (DBD ライブラリーのエクスポート・データ・セットの場合)
- CCUPSBEX DD ステートメント (PSB ライブラリーのエクスポート・データ・セットの場合)

ACB の IMS 管理が使用可能な場合は、IMS カタログによって管理されるアクティブ ACB をコピーするために必要なカタログ・システム・エクスポート・データ・セットに対して、CCUDIREX DD ステートメントを使用します。カタログ・システム・エクスポート・データ・セットには、IMS カタログ・ディレクトリー内にあるアクティブ ACB に関する情報、および IMS カタログ内にある対応する DBD レコードと PSB レコードに関する情報が含まれています。IMS カタログ・ディレクトリー内でアクティブな ACB が見つからない場合、IMS カタログ・ディレクトリーのコピーとエクスポートは行われません。

CCUDIREX DD ステートメントを使用する場合、以下の DD ステートメントは無視されます。

- CCUACBEX DD
- CCUACB DD
- CCUDBDEX DD
- CCUDBD DD
- CCUPSBEX DD
- CCUPSB DD

次の図に、IMS Catalog Export ユーティリティーの入力と出力を示します。

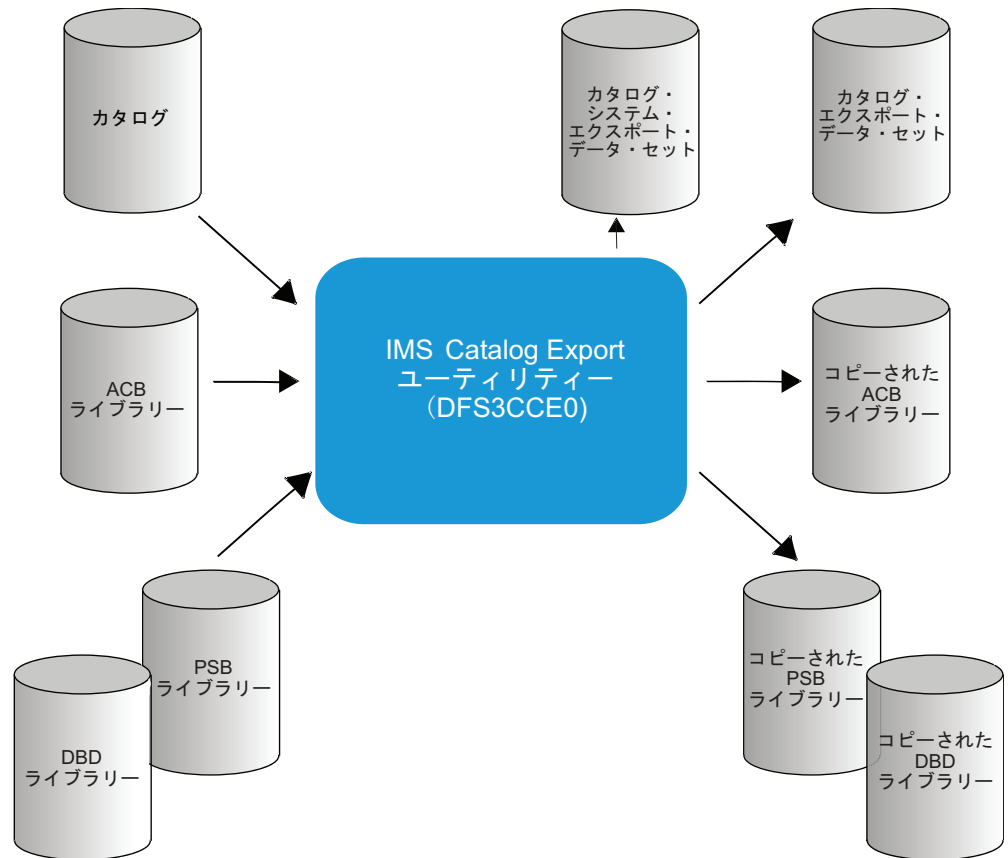


図 33. エクスポートでの IMS Catalog Export ユーティリティの入力と出力

## エクスポート JCL の仕様

### IMS Catalog Export ユーティリティの JCL ステートメント

次の JCL は、IMS カタログをエクスポート・データ・セットにコピーするときに使用できる JCL の例です。

```
//EXPRTCAT EXEC PGM=DFSRR00,
// PARM=(DLI,DFS3CCE0,DFSCP000,,,,,,,,,Y,{Y|N],[ir|lname],
//      ,,,,,,,,,,DFSDF=xxx')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//IMSACB DD ... ACBLIB
//IMS DD ... PSBLIB
// DD ... DBDLIB
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=* Dump data set
//SYSPRINT DD SYSOUT=* Messages
//IEFRDER DD ... Log data set
//DFSVSAMP DD ... Buffer pool parameters
//CCUCATEX DD ... Catalog export data set
//CCUACB DD ... Input data set for ACBLIB export function
//CCUACBEX DD ... Output data set for ACBLIB export function
//CCUDBD DD ... Input data set for DBDLIB export function
//CCUDBDEX DD ... Output data set for DBDLIB export function
//CCUPSB DD ... Input data set for PSBLIB export function
//CCUPSBEX DD ... Output data set for PSBLIB export function
```

ACB の IMS 管理が使用可能な場合には、例えば以下に示す JCL ステートメントを使用して、カタログ・システム・エクスポート・データ・セットを使用して IMS カタログ・ディレクトリーをコピーすることができます。CCUDIREX DD ステートメントを使用する場合、以下のステートメントは無視されます。

- CCUACB DD
- CCUACBEX DD
- CCUDBD DD
- CCUDBDEX DD
- CCUPSB DD
- CCUPSBEX DD

```
//EXPRTCAT EXEC PGM=DFSRR00,
// PARM=(DLI,DFS3CCE0,DFSCP000,,,,,,,,,Y,{Y|N],[ir|lname],
//          ,,,,,,,,,,'DFSDF=xxx')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//IMSACB DD ... ACBLIB
//IMS DD ... PSBLIB
// DD ... DBDLIB
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=* Dump data set
//SYSPRINT DD SYSOUT=* Messages
//IEFRDER DD ... Log data set
//DFSVSAMP DD ... Buffer pool parameters
//CCUCATEX DD ... Catalog export data set
//CCUACB DD ... Input data set for ACBLIB export function
//CCUACBEX DD ... Output data set for ACBLIB export function
//CCUDBD DD ... Input data set for DBDLIB export function
//CCUDBDEX DD ... Output data set for DBDLIB export function
//CCUPSB DD ... Input data set for PSBLIB export function
//CCUPSBEX DD ... Output data set for PSBLIB export function
//CCUDIREX DD ... Catalog directory export data set
```

カタログを共用しない場合、コピー・ユーティリティーは、そのカタログを使用している IMS で BMP ジョブによってそのカタログにアクセスしてコピーできます。以下に、BMP でカタログをコピーする場合のサンプル JCL ジョブ・ステップを示します。

```
//EXPRTCAT EXEC PGM=DFSRR00,
//          PARM='BMP,DFS3CCE0,DFSCP000,,,,,,,,,imsname'
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//IMSACB DD ... ACBLIB
//IMS DD ... PSBLIB
// DD ... DBDLIB
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=* Dump data set
//SYSPRINT DD SYSOUT=* Messages
//CCUCATEX DD ... Catalog export data set
//CCUACB DD ... Input data set for ACBLIB export function
//CCUACBEX DD ... Output data set for ACBLIB export function
//CCUDBD DD ... Input data set for DBDLIB export function
//CCUDBDEX DD ... Output data set for DBDLIB export function
//CCUPSB DD ... Input data set for PSBLIB export function
//CCUPSBEX DD ... Output data set for PSBLIB export function
```

IMS カタログ・ディレクトリー・データ・セットを共用しない場合、コピー・ユーティリティーは、そのデータ・セットにアクセスし、BMP ジョブを使用して、そのデータ・セットを使用している IMS にコピーできます。以下に、BMP でカタログ

をコピーする場合のサンプル JCL ジョブ・ステップを示します。CCUDIREX DD ステートメントを使用する場合、以下のステートメントは無視されます。

- CCUACB DD
- CCUACBEX DD
- CCUDBD DD
- CCUDBDEX DD
- CCUPSB DD
- CCUPSBEX DD

```
//EXPRTCAT EXEC PGM=DFSRR00,  
//          PARM='BMP,DFS3CCE0,DFSCP000,,,,,,,,,imsname'  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//IMSACB DD ... ACBLIB  
//IMS DD ... PSBLIB  
// DD ... DBDLIB  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//SYSABEND DD SYSOUT=* Dump data set  
//SYSPRINT DD SYSOUT=* Messages  
//CCUCATEX DD ... Catalog export data set  
//CCUACB DD ... Input data set for ACBLIB export function  
//CCUACBEX DD ... Output data set for ACBLIB export function  
//CCUDBD DD ... Input data set for DBDLIB export function  
//CCUDBDEX DD ... Output data set for DBDLIB export function  
//CCUPSB DD ... Input data set for PSBLIB export function  
//CCUPSBEX DD ... Output data set for PSBLIB export function  
//CCUDIREX DD ... Catalog directory export data set
```

## Export DD ステートメントの説明

### DFSRESLB DD

IMS SVC モジュールが入っている許可ライブラリーを指します。IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) で許可する必要があります。この DD ステートメントには、許可ライブラリー中になければならない IMS SVC モジュール用の許可ライブラリーを指定します。JOBLIB または STEPLIB ステートメントは、IMS バッチには許可は不要です。

### DFSVSAMP

バッファー・プール・パラメーター・データ・セットを定義します。

### IEFRDER DD

プライマリー IMS ログ・データ・セットを定義します。

### CCUACB DD

IMS Catalog Export ユーティリティの場合は、コピーする入力 ACB ライブラリー・データ・セットを定義します。

IMS Catalog Export ユーティリティは、CCUACB DD ステートメント上の ACB ライブラリー・データ・セットの連結をサポートしません。

IMS Catalog Export ユーティリティの JCL に CCUACB DD ステートメントと IMSACB DD ステートメントの両方が指定されている場合、両方のステートメントが同じ ACB ライブラリー・データ・セットを指していることを確認します。

**CCUACBEX DD**

ACB ライブラリーのコピーの出力データ・セットを定義します。

**CCUCATEX DD**

IMS カタログのコピーの出力データ・セットを定義します。

**CCUDBD DD**

IMS Catalog Export ユーティリティの場合は、コピーする入力 DBD ライブラリー・データ・セットを定義します。

**CCUDBDEX DD**

DBD ライブラリーのコピーの出力データ・セットを定義します。

**CCUDIREX DD**

IMS カタログのアクティブ ACB に関する情報を格納するための出力データ・セットを定義します。この情報は、ターゲット IMS システムにアクティブ ACB を取り込むために必要です。

データ・セット名の長さは、最長で 44 文字です。ただし、最後の修飾子を除くすべての修飾子の長さを 37 文字までとすることができます。修飾子は、エクスポートされた IMS ディレクトリー・データ・セットの高位修飾子として使用されます。

**CCUPSB DD**

IMS Catalog Export ユーティリティの場合は、コピーする入力 PSB ライブラリー・データ・セットを定義します。

**CCUPSBEX DD**

PSB ライブラリーのコピーの出力データ・セットを定義します。

**IMS DD**

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。

**IMSACB DD**

IMS カタログへのデータの取り込みに使用された、ACB ライブラリー・データ・セットを定義するオプションの DD ステートメント。

IMSACB DD ステートメントを指定して、レコード内の DBD または PSB メンバーのうち、対応する ACB ライブラリー・メンバーのタイム・スタンプと一致するタイム・スタンプ・バージョンのみをコピーします。IMSACB DD ステートメントを省略すると、IMS Catalog Export ユーティリティは各レコード内のメンバーのすべてのタイム・スタンプ・バージョンをコピーします。

複数の ACB ライブラリー・データ・セットを IMS Catalog Export ユーティリティへの入力として定義しなければならない場合、ACB ライブラリー・データ・セットを IMSACB DD ステートメントに連結する必要があります。

IMSACB DD ステートメントによって参照されているデータ・セットはコピーされません。ACB ライブラリー・データ・セットをコピーするには、それらのデータ・セットを CCUACB DD ステートメントによって参照し、対応する出力データ・セットを CCUACBEX DD ステートメントによって参照する必要があります。



IMS Catalog Export ユーティリティの JCL に CCUACB DD ステートメントと IMSACB DD ステートメントの両方が指定されている場合、両方のステートメントが同じ ACB ライブラリー・データ・セットを指していることを確認します。

#### **PROCLIB DD**

ユーティリティに必要な、IMS カタログの各種属性を定義する DFSDFxxx メンバーが入った IMS.PROCLIB データ・セットを定義します。

#### **STEPLIB DD**

IMS.SDFSRESL を指します。ここには、IMS 中核と必須の IMS モジュールが入っています。IMS.SDFSRESL に連結された無許可ライブラリーがあるために STEPLIB が無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

#### **SYSABEND DD**

ダンプ・データ・セットを定義します。

#### **SYSPRINT DD**

統計報告書およびユーティリティ・メッセージの出力データ・セットを定義します。SYSPRINT データ・セットを定義しない場合、このユーティリティが定義します。

### **エクスポート統計報告書**

IMS Catalog Export ユーティリティはエクスポート統計報告書を SYSPRINT データ・セットに出力します。

この報告書には、以下の 2 つのセクションがあります。最初のセクションには、このユーティリティが IMS カタログからエクスポート・データ・セットにコピーする DBD または PSB レコードがリストされます。コピーされる各 DBD または PSB レコードは、それぞれにレコード・タイプ、DBD または PSB 名、およびタイム・スタンプが指定された別個の項目としてリストされます。

IMS カタログ内の DBD または PSB メンバーのレコードにレコード・セグメントの複数のタイム・スタンプ・バージョンが含まれる場合、そのレコードは報告書内で複数の行に表示され、各行が DBD または PSB レコード内のセグメントの異なるタイム・スタンプ・バージョンを表します。DBD または PSB レコード内のセグメントの異なるタイム・スタンプ・バージョンを表す各項目は、タイム・スタンプ以外は同じです。このタイム・スタンプは、そのメンバーの対応するバージョンが ACB ライブラリー内で生成された時刻を反映します。

2 番目のセクションには、このユーティリティはエクスポート・データ・セットにコピーする各タイプのセグメントの総数をリストします。ヘッダー・セグメントは、DBD または PSB メンバー・レコードのルート・セグメントです。IMS カタログに、1 つの DBD または PSB メンバーに対して複数のタイム・スタンプ・バージョンが含まれているかどうかに関係なく、IMS カタログに含まれる各 DBD または PSB 名のルート・セグメントは 1 つのみです。DBD または PSB メンバーの異なるタイム・スタンプ・バージョンは、カタログ内でヘッダー・セグメント直下に兄弟セグメントとして保管されます。

次の例では、IMS Catalog Export ユーティリティーは 24 個の DBD および PSB メンバーのレコードをコピーしました。報告書の最初のセクションには 26 行表示されています。IMS カタログ内には、DBD DBOVLFPFC および PSB PSBCOMPT のメンバー・レコードにそれぞれ 2 つのタイム・スタンプ・バージョンがあるためです。

例の 2 番目のセクションでは、ヘッダー・セグメントの数は DBD および PSB メンバー名の実際の数に反映しています。それに対し、コピーされた DBD および PSB セグメントの合計は、コピーされたタイム・スタンプ・バージョンの総数を反映しています。

```
DFS4803I DBD DBOHIDK5 WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DBOVLFPFC WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DBOVLFPFC WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DBVHDJ05 WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DFSCD000 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DFSCX000 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DHVNTZ02 WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DIVNTZ02 WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DLOHIDK5 WITH TIME STAMP 1133521550000 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DLVHDJ05 WITH TIME STAMP 1133521550000 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DLVNTZX2 WITH TIME STAMP 1133521560000 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DLVNTZ02 WITH TIME STAMP 1133521560000 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD DXVNTZ02 WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD D2XHDJ05 WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
DFS4803I DBD D2XHIDK5 WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB CATCP000 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB DFSCPL00 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB DFSCP000 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB DFSCP001 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB DFSCP002 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB DFSCP003 WITH TIME STAMP 1133521563369 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB PLVAPZ12 WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB PLVAPZ22 WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB PSBCOMPT WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB PSBCOMPT WITH TIME STAMP 1133521561758 WAS COPIED TO THE EXPORT DATA SET.
DFS4804I PSB PSBEJK05 WITH TIME STAMP 1133521555167 WAS COPIED TO THE EXPORT DATA SET.
```

```
DFS4805I 00000024 HEADER SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000015 DBD SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000003 DBDRMK SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000010 DSET SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000329 SEGM SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000006 SEGMRMK SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000944 FLD SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000004 FLDRMK SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000944 MAR SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000038 LCHILD SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000003 XDFLD SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000011 PSB SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000004 PSBRMK SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000036 PCB SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000009 PCBMRK SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000860 SS SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
DFS4805I 00000030 DBDXREF SEGMENTS WERE COPIED TO THE EXPORT DATA SET.
```

CCUDIREX DD ステートメントを使用して IMS カタログのアクティブ ACB をコピーした場合、作成される出力データ・セットのリストが生成されます。この出力データ・セットは、IMS Catalog Import ユーティリティーの使用時にシステムで使用可能になっている必要があります。出力データ・セットのリストは、エクスポート・ジョブ出力の末尾に出力されます。以下の例は、IMS Catalog Import ユーティリティー用に使用可能でなければならない 3 つの出力データ・セットを示しています。

CATALOG COPY EXPORT UTILITY

DFS4813I THE FOLLOWING DATA SETS MUST BE MADE AVAILABLE FOR  
IMPORT OF THIS IMS CATALOG DIRECTORY:

IMSTESTS.DIR.CONTROL  
IMSTESTS.DIR.DX1001  
IMSTESTS.DIR.DX1002

## ライブラリー統計報告書

IMS Catalog Export ユーティリティーは、z/OS データ・セット・ユーティリティー IEBCOPY を呼び出して ACB、DBD、および PSB ライブラリーをコピーします。IEBCOPY ユーティリティーは、コピー操作に関する統計とメッセージを表示します。IEBCOPY ユーティリティーについて詳しくは、このトピックの末尾近くにある関連リンクを参照してください。

## 戻りコード

IMS Catalog Export ユーティリティーは以下のコードを返します。

- 0 すべての操作が正常に終了
- 4 1 つ以上の警告メッセージが出された
- 8 以上  
重大エラーでプログラムが中止された

---

## IMS Catalog Import ユーティリティー (DFS3CCI0)

IMS Catalog Import ユーティリティー (DFS3CCI0) を宛先環境で使用して、IMS カタログをロードまたは更新し、そこに含まれる ACB、DBD、および PSB ライブラリーをエクスポート・データ・セットから宛先データ・セットにコピーします。

発信元環境で ACB の IMS 管理が使用可能な場合、IMS Catalog Import ユーティリティーを使用して、IMS カタログ・ディレクトリー・データ・セット内のアクティブ ACB をインポートすることができます。IMS Catalog Import ユーティリティーは、それらの ACB を宛先環境の IMS カタログ・ディレクトリーにインストールできます。

IMS Catalog Import ユーティリティーは、ACB、DBD、および PSB ライブラリー・データ・セットをコピーするために z/OS データ・セット・ユーティリティー IEBCOPY を呼び出します。IEBCOPY では入力データ・セットの連結はサポートされません。そのため、1 つのジョブ・ステップでは各タイプのライブラリーの 1 つしかコピーできません。

IMS Catalog Import ユーティリティーは、宛先環境で IMS カタログをロードまたは更新したときにインポート統計報告書を生成します。IEBCOPY ユーティリティーは、それ自体の統計に関する報告書を作成します。

IMS Catalog Import ユーティリティーはバッチ領域内でも、オンライン BMP としても実行できます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 『入力と出力』
- 403 ページの『インポート JCL の仕様』
- 405 ページの『Import DD ステートメントの説明』
- 407 ページの『制御ステートメント』
- 412 ページの『インポート統計報告書』
- 414 ページの『ライブラリー統計報告書』
- 414 ページの『戻りコード』

### 制約事項

IMS Catalog Import ユーティリティは、システム・サービス・ユーティリティ IEBCOPY を使用して ACB、PSB、および DBD ライブラリーをコピーします。IEBCOPY ユーティリティは入力データ・セットの連結をサポートしないため、IMS Catalog Import ユーティリティは単一のジョブ・ステップで、各タイプのライブラリーごとにデータ・セットを 1 つしかコピーできません。

### 前提条件

IMS Catalog Import ユーティリティに固有の前提条件は、現在文書化されていません。

### 要件

IMS Catalog Import ユーティリティに固有の要件は、現在文書化されていません。

### 推奨事項

IMS Catalog Import ユーティリティに固有の推奨事項は、現在文書化されていません。

### 入力と出力

IMS Catalog Import ユーティリティ (DFS3CCI0) の 1 次入力は、IMS Catalog Export ユーティリティ (DFS3CCE0) によって生成され、IMS カタログのコピーが含まれているエクスポート・データ・セットです。このデータ・セットを識別するために CCUCATIM DD ステートメントが必要です。IMS で 宛先 IMS 環境の ACB を管理する場合、もう 1 つの重要な入力は、発信元 IMS 環境からの IMS 管理のアクティブ ACB に関する情報が含まれたエクスポート・データ・セットです。このデータ・セットを識別するために CCUDIRIM DD ステートメントが必要です。

IMS カタログのコピー中、IMS Catalog Export ユーティリティは IMS カタログへのアクセスに使用される PSB および DBD ライブラリーを読み取る必要があ

ります。この目的で、その PSB および DBD ライブラリーを参照するために //IMS DD ステートメントが必要です。入力 IMS カタログ・データ・セットには DD ステートメントは不要です。

IMS Catalog Copy ユーティリティは IMSACB DD ステートメントを使用して、IMS カタログからコピーされる DBD および PSB メンバー・セグメントのタイム・スタンプ・バージョンの数を制限します。

コピー処理の整合性を維持するには、CCUACB、CCUDBD、および CCUPSB の各 DD ステートメントによって記述されるデータ・セットが、IMS および IMSACB DD ステートメントによって記述されるデータ・セットと同じでなければなりません。

IMS Catalog Export ユーティリティで ACB ライブラリー、DBD ライブラリー、および PSB ライブラリーをコピーした場合、これらのライブラリーは以下の DD ステートメントを使用してインポート JCL で識別します。

- CCUACBIM DD ステートメント (ACB ライブラリーのエクスポート・データ・セットの場合)
- CCUDBDIM DD ステートメント (DBD ライブラリーのエクスポート・データ・セットの場合)
- CCUPSBIM DD ステートメント (PSB ライブラリーのエクスポート・データ・セットの場合)

IMS Catalog Import ユーティリティを使用して、ACB ライブラリー、DBD ライブラリー、および PSB ライブラリーを宛先ライブラリーにコピーするには、以下の DD ステートメントを使用してインポート JCL で宛先ライブラリーを識別します。

- CCUACB DD ステートメント (ACB ライブラリー・データ・セットの場合)
- CCUDBD DD ステートメント (DBD ライブラリー・データ・セットの場合)
- CCUPSB DD ステートメント (PSB ライブラリー・データ・セットの場合)

IMS ディレクトリーの IMS 管理のアクティブ ACB を IMS Catalog Export ユーティリティでコピーした場合は、インポート JCL で、CCUDIRIM DD ステートメントを使用して ACB のカタログ・システム・エクスポート・データ・セットを識別します。CCUDIRIM DD ステートメントは、また、IMS 管理の ACB 環境内の IMS ディレクトリー・データ・セットに ACB を追加するようユーティリティに指示します。

CCUDIRIM DD ステートメントを使用すると、以下の DD ステートメントは、指定されていても無視されます。

- CCUACBIM DD
- CCUACB DD
- CCUDBDIM DD
- CCUDBD DD
- CCUPSBIM DD
- CCUPSB DD

重要: CCUDIRIM DD ステートメントを使用する場合は、IMS Catalog Import ユーティリティの JCL で SYSINP DD ステートメントを使用して、MANAGEDACBS= 制御ステートメント・パラメーターを指定する必要があります。MANAGEDACBS= 制御ステートメント・パラメーターは、コピーされた ACB を使用して宛先システムの IMS ディレクトリー・データ・セットを作成または更新するように、IMS Catalog Import ユーティリティに指示します。MANAGEDACBS= 制御ステートメント・パラメーターの説明については、407 ページの『制御ステートメント』を参照してください。

IMS Catalog Import ユーティリティは、CCUDIRIM DD ステートメントを使用して、アクティブ ACB を表すセグメントのみを宛先環境の IMS ディレクトリーにコピーします。

次の図に、IMS Catalog Import ユーティリティの入力と出力を示します。

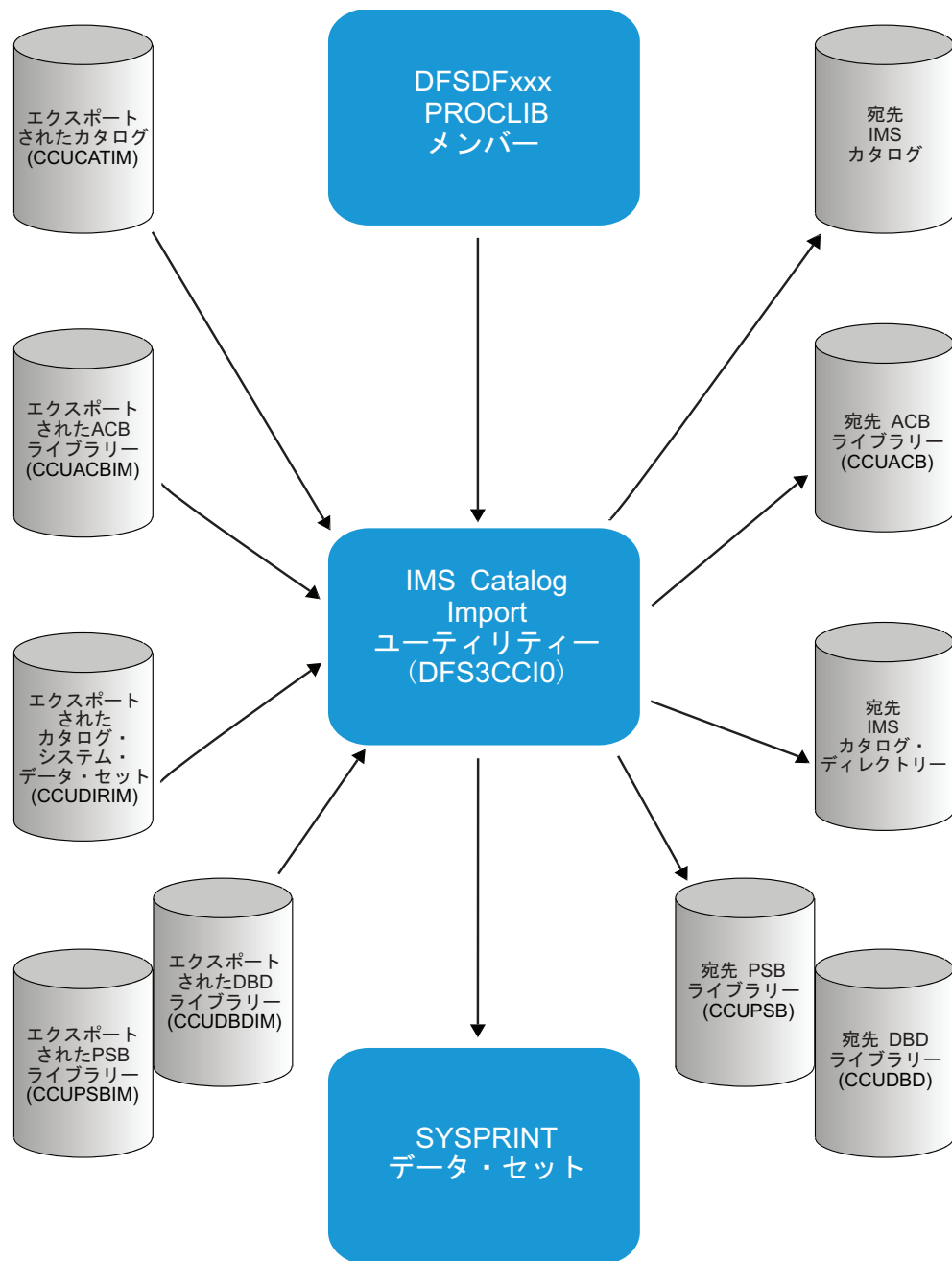


図 34. インポートでの IMS Catalog Import ユーティリティの入力と出力

## インポート JCL の仕様

既存の IMS カタログに更新モードでインポートする場合の JCL ステートメント

次のサンプル JCL ステートメントは、エクスポートされた DBD および PSB メタデータを既存のカタログまたは新規カタログにインポートするときに使用できます。指定された ACB、DBD、および PSB ライブラリーを指定された宛先 ACB、DBD、および PSB ライブラリーにコピーできます。

```
//IMPRTCAT EXEC PGM=DFS3CCI0,
// PARM=(DLI,DFS3CCI0,DFSCP001,,,,,,,,,Y,{Y|N],[irlname],
//                ,,,,,,,,,,'DFSDF=xxx')
```

```

//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=*      Dump data set
//SYSPRINT DD SYSOUT=*      Messages
//IEFRDER DD ...           Log data set with catalog updates
//DFSVSAMP DD ...           Buffer pool parameters
//IMSACB DD ...            ACBLIB
//IMS DD ...               PSBLIB
// DD ...                 DBDLIB
//CCUCATIM DD ...          Catalog import data set
//CCUACBIM DD ...          Input data set for ACBLIB import function
//CCUACB DD ...            Output data set for ACBLIB import function
//CCUDBDIM DD ...          Input data set for DBDLIB import function
//CCUDBD DD ...            Output data set for DBDLIB import function
//CCUPSBIM DD ...          Input data set for PSBLIB import function
//CCUPSB DD ...            Output data set for PSBLIB import function

```

カタログを共用しない場合、コピー・ユーティリティーは、そのカタログを使用している IMS で BMP ジョブによってそのカタログを更新できます。以下に、BMP でカタログを更新する場合のサンプル JCL ジョブ・ステップを示します。

```

//IMPRTCAT EXEC PGM=DFS3CCI0,
//          PARM=(BMP,DFS3CCI0,DFSCP001,,,,,,,,,imsname')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=*      Dump data set
//SYSPRINT DD SYSOUT=*      Messages
//IEFRDER DD ...           Log data set with catalog updates
//DFSVSAMP DD ...           Buffer pool parameters
//IMSACB DD ...            ACBLIB
//IMS DD ...               PSBLIB
// DD ...                 DBDLIB
//CCUCATIM DD ...          Catalog import data set
//CCUACBIM DD ...          Input data set for ACBLIB import function
//CCUACB DD ...            Output data set for ACBLIB import function
//CCUDBDIM DD ...          Input data set for DBDLIB import function
//CCUDBD DD ...            Output data set for DBDLIB import function
//CCUPSBIM DD ...          Input data set for PSBLIB import function
//CCUPSB DD ...            Output data set for PSBLIB import function

```

以下のサンプル JCL ステートメントを使用すると、IMS カタログ・ディレクトリを発信元環境から、すでにアクティブ ACB を管理している宛先環境にある既存の IMS カタログにインポートすることができます。

```

//IMPRTCAT EXEC PGM=DFS3CCI0,
//          PARM=(DLI,DFS3CCI0,DFSCP001,,,,,,,,,Y,{Y|N},[ir|lname],
//          ,,,,,,,,,,'DFSDF=xxx')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=*      Dump data set
//SYSPRINT DD SYSOUT=*      Messages
//IEFRDER DD ...           Log data set with catalog updates
//DFSVSAMP DD ...           Buffer pool parameters
//CCUCATIM DD ...          Catalog import data set
//CCUDIRIM DD ...          Catalog directory import data set
//SYSINP DD *              MANAGEDACBS=UPDATE

```

新規の IMS カタログにロード・モードでインポートする場合の JCL ステートメント



次の JCL は、IMS カタログをエクスポート・データ・セットからロードするときに使用できる JCL の例です。

IMS カタログのデータベース・データ・セットが存在しない場合、IMS Catalog Import ユーティリティの実行中に IMS が自動的に作成します。

データ・セットのサイズまたは配置を自分で制御したい場合は、IMS Catalog Populate ユーティリティ (DFS3PU00) を使用して、IMS カタログのデータ・セットに必要な DASD スペースを評価できます。詳細については、412 ページの『インポート統計報告書』を参照してください。

IMS カタログを更新またはロードした後は、GENJCLIC または標準イメージ・コピー JCL を使用して IMS カタログをバックアップします。

```
//IMPRTCAT EXEC PGM=DFS3CCI0,  
// PARM=(DLI,DFS3CCI0,DFSCPL00,,,,,,,,,Y,{Y|N],[ir|lname],  
//      ,,,,,,,,,,'DFSDF=xxx')  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//SYSABEND DD SYSOUT=*          Dump data set  
//SYSPRINT DD SYSOUT=*          Messages  
//IEFRDER DD ...                Log data set with catalog updates  
//DFSVSAMP DD ...              Buffer pool parameters  
//IMSACB DD ...                ACBLIB  
//IMS DD ...                   PSBLIB  
// DD ...                      DBDLIB  
//CCUCATIM DD ...              Catalog import data set  
//CCUACBIM DD ...              Input data set for ACBLIB import function  
//CCUACB DD ...                Output data set for ACBLIB import function  
//CCUDBDIM DD ...              Input data set for DBDLIB import function  
//CCUDBD DD ...                Output data set for DBDLIB import function  
//CCUPSBIM DD ...              Input data set for PSBLIB import function  
//CCUPSB DD ...                Output data set for PSBLIB import function
```

以下の JCL は、IMS ディレクトリーをカタログ・システム・エクスポート・データ・セットからロードするときに使用できる JCL の例です。

```
//IMPRTCAT EXEC PGM=DFS3CCI0,  
// PARM=(DLI,DFS3CCI0,DFSCPL00,,,,,,,,,Y,{Y|N],[ir|lname],  
//      ,,,,,,,,,,'DFSDF=xxx')  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//SYSABEND DD SYSOUT=*          Dump data set  
//SYSPRINT DD SYSOUT=*          Messages  
//IEFRDER DD ...                Log data set with catalog updates  
//DFSVSAMP DD ...              Buffer pool parameters  
//CCUCATIM DD ...              Catalog import data set  
//CCUDIRIM DD ...              Catalog directory import data set  
//SYSIN DD *                    MANAGEDACBS=SETUP
```

## Import DD ステートメントの説明

### DFSRESLB DD

IMS SVC モジュールが入っている許可ライブラリーを指します。IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) で許可する必要があります。この DD ステートメントには、許可ライブラリー中になければならない IMS SVC モジュール用の許可ライブラリーを指定します。JOBLIB または STEPLIB ステートメントは、IMS バッチには許可は不要です。

**DFSVSAMP**

バッファ・プール・パラメーター・データ・セットを定義します。

**IEFRDER DD**

プライマリー IMS ログ・データ・セットを定義します。

**CCUACB DD**

IMS Catalog Import ユーティリティの場合は、宛先 ACB ライブラリー・データ・セットを定義します。

**CCUACBIM DD**

インポートする ACB ライブラリーのコピーを保持する入力データ・セットを定義します。

**CCUCATIM DD**

インポートする IMS カタログのコピーを保持する入力データ・セットを定義します。

**CCUDBD DD**

IMS Catalog Import ユーティリティの場合は、宛先 DBD ライブラリー・データ・セットを定義します。

**CCUDBDIM DD**

インポートする DBD ライブラリーのコピーを保持する入力データ・セットを定義します。

**CCUDIRIM DD**

インポートする IMS カタログのアクティブ ACB に関する情報を保持する入力データ・セットを定義します。

**CCUPSB DD**

IMS Catalog Import ユーティリティの場合は、宛先 PSB ライブラリー・データ・セットを定義します。

**CCUPSBIM DD**

インポートする PSB ライブラリーのコピーを保持する入力データ・セットを定義します。

**IMS DD**

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。

**IMSACB DD**

ACB ライブラリーのコピーを保持する入力データ・セットを定義するオプションの DD ステートメント。発信元 IMS システムで IMS カタログのデータを取り込むために使用されます。

複数の ACB ライブラリー・データ・セットを入力として定義する場合、ACB ライブラリー・データ・セットを IMSACB DD ステートメントに連結する必要があります。

IMSACB DD ステートメントによって参照されているデータ・セットはインポートされません。ACB ライブラリー・データ・セットをインポートするには、それらのデータ・セットを CCUACBIM DD ステートメントを使用して参照し、対応する宛先データ・セットを CCUACB DD ステートメントによって参照する必要があります。

### **PROCLIB DD**

ユーティリティに必要な、IMS カタログの各種属性を定義する DFSDFxxx メンバーが入った IMS.PROCLIB データ・セットを定義します。

### **STEPLIB DD**

IMS.SDFSRESL を指します。ここでは、IMS 中核と必須の IMS モジュールが入っています。IMS.SDFSRESL に連結された無許可ライブラリーがあるために STEPLIB が無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

### **SYSABEND DD**

ダンプ・データ・セットを定義します。

### **SYSINP DD**

CCUDIRIM DD ステートメントを使用する場合は、80 文字の固定長レコードを持つ必須制御ステートメント順次データ・セットです。位置 1 から 72 までの文字だけが読み取られます。

1 つ以上のレコードで、制御ステートメント・パラメーターをブランクまたはコンマで区切って指定することができます。

CCUDIRIM DD ステートメントを指定する場合は、IMS Catalog Import ユーティリティの JCL で SYSINP DD ステートメントを使用して、MANAGEDACBS= 制御ステートメント・パラメーターを指定する必要があります。MANAGEDACBS= 制御ステートメント・パラメーターは、コピーされた ACB を使用して宛先システムの IMS ディレクトリー・データ・セットを作成または更新するように、IMS Catalog Import ユーティリティに指示します。

SYSINP DD ステートメントで指定できる制御ステートメントの説明については、『制御ステートメント』を参照してください。

### **SYSPRINT DD**

統計報告書およびユーティリティ・メッセージの出力データ・セットを定義します。SYSPRINT データ・セットを定義しない場合、このユーティリティが定義します。

## **制御ステートメント**

SYSINP DD ステートメントを使用して、1 つ以上のレコードで、以下の制御ステートメント・パラメーターをブランクまたはコンマで区切って指定することができます。

### **DUPLIST**

既に IMS カタログにあるインスタンスと重複するために IMS カタログに追加されない、入力 ACB ライブラリー内の各 DBD リソースまたは PSB リソースを DFS3PU00 ユーティリティでリストすることを指定します。IMS カタログ内のリソースの重複するインスタンスごとに、ユーティリティはメッセージ DFS4436I を出力します。

MANAGEDACBS= ステートメントが指定されており、ACB の IMS 管理が使用可能な場合、ユーティリティは、既に IMS ディレクトリーにあるリソースのインスタンスと重複するために IMS ディレクトリーまたはステージング・データ・セットに追加されない、各 DBD リソースまたは PSB リソースもリスト

します。

MANAGEDACBS= ステートメントで UPDATE パラメーターが指定された場合、ユーティリティは重複するインスタンスごとにメッセージ DFS4531I を出力します。

UPDATE パラメーターで LATEST が指定されたかデフォルトとして受け入れられた場合、ユーティリティは、IMS ディレクトリー内のリソース・インスタンスのタイム・スタンプがより後の時刻であるために IMS ディレクトリーに追加されないインスタンスごとに、メッセージ DFS4522I も出力します。

MANAGEDACBS= ステートメントで STAGE パラメーターが指定された場合、ユーティリティは、重複するインスタンスであるためにステージング・データ・セットにコピーされないリソースごとに、メッセージ DFS4542I を出力します。

STAGE パラメーターで LATEST が指定されたかデフォルトとして受け入れられた場合、ユーティリティは、IMS ディレクトリー内のリソース・インスタンスのタイム・スタンプがより後の時刻であるためにステージング・データ・セットに追加されないインスタンスごとに、メッセージ DFS4539I も出力します。

#### **ERRORMAX=*n***

*n* 個を超えるメッセージが、特定の DBD および PSB が IMS カタログに書き込まれるメタデータを持つことを妨げるエラーを示す場合、IMS Catalog Populate ユーティリティを終了します。メタデータの重複するインスタンスは、この制限にカウントされません。このオプションを省略する場合、制限はありません。

#### **RESOURCE\_CHKP\_FREQ=*n***

チェックポイントの間に挿入する DBD および PSB リソース・インスタンスの数を指定します。*n* は、1 から 99999999 までの 1 桁から 8 桁の数値です。デフォルト値は 100 です。

#### **SEGMENT\_CHKP\_FREQ=*n***

チェックポイントの間に挿入するセグメントの数を指定します。この数に達すると、IMS はチェックポイントを発行する前に、現在処理されているリソース・インスタンスのセグメントすべての挿入を完了します。*n* は、1 から 99999999 までの 1 桁から 8 桁の数値です。デフォルト値は 1000 です。

注: 最初のチェックポイント頻度数に達すると、チェックポイントが発行され、カウンターは 0 にリセットされます。

#### **ISRTLST**

ACB の IMS 管理が使用可能の場合、ユーティリティは、IMS ディレクトリーに追加されるか、後で IMS ディレクトリーにインポートするためにステージング・データ・セットに保管される各 DBD リソースまたは PSB リソースもリストします。

ユーティリティは、以下のメッセージを出力することにより、各リソースに対して実行されるアクションを識別します。

#### **DFS4520I**

リソースは、新規リソースとして IMS ディレクトリーに追加されました。

#### DFS4521I

リソースのこのインスタンスは、IMS ディレクトリー内のリソースの既存インスタンスに取って代わりました。

#### DFS4537I

リソースは、ステージング・データ・セットへコピーされました。リソースは、後で IMS ディレクトリーにインポートされる時、IMS ディレクトリーに新規リソースとして挿入されます。

#### DFS4538I

リソースは、ステージング・データ・セットへコピーされました。リソースは、後で IMS ディレクトリーにインポートされる時、IMS ディレクトリー内のリソースの既存インスタンスに取って代わります。

#### MANAGEDACBS=

MANAGEDACBS 制御ステートメントは、以下のアクションを実行するために使用します。

- データベースおよびプログラム・ビューのランタイム・アプリケーション制御ブロック (ACB) を管理するために IMS をセットアップします。
- 新規 ACB または ACB ライブラリー・データ・セットから変更した ACB を使用して、ACB を管理する IMS システムを更新します。
- ACB を管理する IMS システムに後でインポートするために、ACB ライブラリーからステージング・データ・セットに ACB を保管します。

MANAGEDACBS ステートメントは、以下の構文図に従って指定することができます。

MANAGEDACBS= ステートメントで複数のパラメーターを指定する場合に限り、括弧が必要です。

以下のリストでは、MANAGEDACBS= ステートメントで指定できるパラメーターについて説明します。

#### SETUP

アプリケーション制御ブロック (ACB) を管理するために IMS に必要な IMS ディレクトリー・データ・セットを作成します。IMS ディレクトリー・データ・セットのすべての既存インスタンスは置き換えられます。

指定した MANAGEDACBS=SETUP でユーティリティーが正常に実行されるまで、DFSDFxxx PROCLIB メンバーの CATALOG セクションに ACBMGMT=CATALOG を指定しないでください。ACB の IMS 管理を使用可能にする前に、IMS ディレクトリーを作成して、アクティブ ACB をロードしておく必要があります。

SETUP が指定された場合、ユーティリティーは、入力 ACB ライブラリー内にある ACB を IMS ディレクトリー・データ・セットに挿入します。

ユーティリティー JCL で IMS カタログ PSB DFSCP001 が指定された場合、ユーティリティーは、新規または変更されたすべての DBD および PSB を IMS カタログに挿入します。

ユーティリティー JCL で IMS カタログ PSB DFSCPL00 が指定された場合、ユーティリティーは既存のすべてのカタログ・レコードを削除し、IMS カタログを再ロードします。

ユーティリティーは、IMS ディレクトリーに ACB を追加するとき、IMS カタログ内の対応する DBD インスタンスおよび PSB インスタンスにアクティブとしてのフラグを立てます。

#### **STAGE**

ACB を入力 ACB ライブラリーからステージング・データ・セットに保管します。ステージング・データ・セットに保管された ACB は、IMPORT DEFN SOURCE(CATALOG) コマンドで IMS システムに追加されるまで活動化されません。

STAGE を指定するときは、ユーティリティー JCL で IMS カタログ PSB DFSCP001 を指定する必要があります。

MANAGEDACBS=STAGE 制御ステートメントが指定されたときにステージング・データ・セットが存在する場合、ユーティリティーは既存のステージング・データ・セットを使用します。既にステージング・データ・セット内にある ACB はすべて保持されます。新規リソースをステージングするときにはステージング・データ・セットをスクラッチして再作成するには、MANAGEDACBS=(STAGE,DELETE) を指定します。DELETE が指定される場合、既存のデータ・セット内にある ACB はすべて失われます。

STAGE が指定された場合、DFS3PU00 ユーティリティーは IMS システム内にまだ存在しない、ACB ライブラリー内のすべての ACB を、ステージング・データ・セットにコピーします。ACB ライブラリー内の ACB が既に IMS システム内に存在する場合、ユーティリティーは、その ACB のタイム・スタンプが UNCOND パラメーターや LATEST パラメーターの基準を満たすかどうかに基づいて、ACB をステージング・データ・セットに保管します。

IMS カタログの初期ロードを行うとき、または ACB の IMS 管理を使用可能にする前に、STAGE を指定しないでください。STAGE パラメーターは、既存の IMS ディレクトリー・データ・セットに対する更新のステージング専用です。

MANAGEDACBS= の後に複数のパラメーターを指定する場合には、括弧で囲む必要があります。例: (STAGE,LATEST)。STAGE のみを指定する場合は、括弧を省略することができます。

#### **DELETE**

ステージング・データ・セットがどのオンライン IMS システムにも割り振られていない場合は、リソースをステージング・データ・セットに追加する前に、ステージング・データ・セットをスクラッチして再作成します。既存のステージング・データ・セット内にある ACB はすべて失われます。

#### **LATEST**

IMS システム内に ACB が既に存在する場合、ACB ライブラリー内のインスタンスのタイム・スタンプが IMS システム内の ACB より新しい場合を除き、ACB ライブラリー内の ACB のインスタンスをステージング・データ・セットに保管しません。

これはデフォルトです。

## UNCOND

IMS システム内に ACB が既に存在する場合、ACB ライブラリー内の ACB のタイム・スタンプが IMS システム内の ACB のタイム・スタンプと同じである場合を除き、ACB ライブラリー内の ACB のインスタンスをステージング・データ・セットに無条件に保管します。

## UPDATE

既存の IMS ディレクトリー・システム・データ・セットを排他モードで直接更新します。ACB はステージング・データ・セットに入れられません。

推奨事項: UPDATE パラメーターを指定する場合、IMS ディレクトリーを使用する IMS システムをすべてシャットダウンしてください。UPDATE が指定される場合、IMS Catalog Populate ユーティリティーには IMS ディレクトリーに対する排他的アクセスが必要です。

また、ユーティリティーは、IMS ディレクトリーの更新時にオンライン IMS システムに通知しません。そのため、更新された ACB をロードするために、ユーティリティーが IMS ディレクトリーを更新するときにオンラインになっている IMS システムを再始動する必要があります。

UPDATE が指定された場合、DFS3PU00 ユーティリティーは、まだ IMS システム内にない ACB を IMS ディレクトリーに無条件に挿入します。IMS システム内に ACB が存在する場合、その ACB のインスタンスは、ACB ライブラリー内のインスタンスのタイム・スタンプ、および UNCOND パラメーターまたは LATEST パラメーターの指定に従って置き換えられます。

IMS カタログの初期ロードを行うとき、または ACB の IMS 管理を使用可能にする前に、UPDATE を指定しないでください。UPDATE パラメーターは、既存の IMS ディレクトリー・データ・セットの更新専用です。

UPDATE を指定するときは、ユーティリティー JCL で IMS カタログ PSB DFSCP001 を指定する必要があります。

MANAGEDACBS の後に複数のパラメーターを指定する場合には、括弧で囲む必要があります。例: (UPDATE,LATEST)。UPDATE のみを指定する場合は、括弧を省略することができます。

## LATEST

IMS システム内に ACB が既に存在する場合、その ACB は、ACB ライブラリー内のインスタンスのタイム・スタンプが IMS システム内の ACB より新しい場合を除き、ACB ライブラリー内の ACB のインスタンスで置き換えられません。

これはデフォルトです。

## UNCOND

IMS システム内に ACB が既に存在する場合、その ACB は、ACB ライブラリー内の ACB のタイム・スタンプが IMS システム内の ACB のタイム・スタンプと同じである場合を除き、ACB ライブラリー内の ACB のインスタンスで無条件に置き換えられます。

## SHARE

動的オプション (DOPT) PSB の場合のみ、共用モードに必要な IMS

ディレクトリー・データ・セットを割り振ることにより、オンライン処理を中断せずに DOPT PSB を IMS カタログに追加できるようにします。

ユーティリティーが IMS カタログをロードする場合は、SHARE を指定しないでください。これが該当するのは、ユーティリティー JCL の実行パラメーターに DFSCPL00 (IMS カタログのロード PSB) の指定がある場合です。

入力 ACBLIB データ・セット内に DOPT PSB ではないリソースが存在する場合は、SHARE を指定しないでください。

SHARE オプションを指定してユーティリティーを実行する前に、z/OS システム内で PDSE データ・セットの拡張共用を使用可能にする必要があります。z/OS 内で PDSE データ・セットの拡張共用を使用可能にするには、シスプレックス内の各システム上で、SYS1.PARMLIB 内の IGDSMSxx メンバーに PDSESHARING(EXTENDED) を指定します。SHARE の指定時に拡張 PDSE 共用が使用可能になっていない場合、ユーティリティーは必要な IMS カタログ・データ・セットを排他モードで割り振ります。これが原因で、他の IMS プロセスおよびアプリケーション・プログラムとの競合が発生する場合があります。

拡張 PDSE 共用の使用可能化について詳しくは、z/OS: 複数システム環境における拡張 PDSE 共用の指定を参照してください。

デフォルトでは、ACB の IMS 管理のために割り振られるシステム・データ・セット名は、IMS カタログの最初の区画が定義されるときに RECON データ・セットに指定されたデータ・セット名の接頭部から取られます。このデフォルトをオーバーライドするには、DFSDFxxx PROCLIB メンバーの CATALOG セクションの SYSDSNHLQ= ステートメントまたはカタログ定義出口ルーチン (DFS3CDX0) で HLQ 名の接頭部を指定できます。

#### **NODUPLIST**

追加されなかったリソース・インスタンスのリストを出力しません。このパラメーターがデフォルトです。

#### **NOISRTLST**

挿入されたリソース・インスタンスのリストを出力しません。

### **インポート統計報告書**

IMS Catalog Import ユーティリティーは、IMS カタログでロードまたは更新されるレコード・セグメントに関するインポート統計報告書を作成します。IMS Catalog Import ユーティリティーを分析専用モードで実行すると、現在このユーティリティーへの入力として使用されている ACB ライブラリーから IMS カタログがロードまたは更新された場合、この報告書は見込み統計のみを反映します。

報告書の最初のセクションは、このユーティリティーの現時点での実行中に挿入されたセグメントの要約です。IMS カタログに挿入されたセグメント・タイプごとに、この要約には以下の情報が列編成で表示されます。

- セグメント・コード (SC)
- セグメント名。通常これは、DBD または PSB ソース・ファイルのマクロに対応します。



- セグメントが保管されている IMS カタログ内のデータ・セット・グループ (DSG)
- リストされたセグメントの親セグメント
- IMS カタログにロードされた、そのタイプのセグメントの総数
- 親セグメントの下にある、そのタイプのセグメント・インスタンスの平均数

報告書の 2 番目のセクションは、このユーティリティーによって更新された既存の DBD および PSB レコードの数と、タイム・スタンプが ACB ライブラリー内の対応する ACB メンバーのタイム・スタンプと一致したために更新されなかった既存の DBD および PSB レコードの数の両方を示します。報告書のこのセクションは、IMS カタログ内でのタイム・スタンプが ACB ライブラリー内でのタイム・スタンプと一致したために、挿入の必要がない DBD および PSB セグメントの数も示します。

報告書の残りのセクションは、ストレージの見積もりを示します。

OSAM データ・セットのために、レポートのストレージ・セクションには指定されたサイズのブロックが一定数だけ示されます。間接リスト・データ・セット (ILDS)、1 次索引データ・セット、副次索引データ・セットを含む VSAM KSDS の場合、レポートには一定数の VSAM レコードが示されます。

これらの数は、DFS3PU00 ユーティリティーへの入力として提供する ACB ライブラリーから構築されるカタログ・レコードをロードするために必要な、スペースの大きさを反映した推定値です。IMS カタログ・データ・セットに必要なストレージの量を計算する場合には、拡張の余地があるよう、計算に十分な追加スペースを加えてください。

IMS に IMS カタログ・データ・セットを自動作成させる場合には、DFSDFxxx PROCLIB メンバーの IMS カタログ・セクション内の SPACEALLOC パラメーターに、ユーティリティーによって提供される推定値のパーセントで追加スペースを指定することができます。このパラメーターのデフォルト値は 500% です。

レポートでは、以下の略語が使用されます。

**DSG** データ・セット・グループ

**L** HALDB ILDS データ・セット。示されるレコード数は、IMS カタログが再編成される場合に作成できる間接リスト項目 (ILE) の、可能な数を表しています。

**SC** セグメント・コード。セグメント・タイプをロードするとき、IMS は固有の ID (整数 1 から 255) としてセグメント・コードを割り当てます。IMS は番号を昇順に割り当て、ルート・セグメント・タイプ (番号 1) から始まり、階層順にすべての従属セグメント・タイプにそれを続けていきます。

**SEGS** セグメント

**X** HALDB 区分 1 次索引。

CATALOG DFSCD000

PARTITION DFSCD01

NUMBER OF SEGMENTS INSERTED INTO THE CATALOG  
INSERTED

AVERAGE

SC	SEGMENT	SEGMENTS	DSG	PARENT	SEGS/PARENT
1	HEADER	4228	A		
2	DBD	2530	A	HEADER	0.6
3	CAPXDBD	7	D	DBD	0.0
5	DSET	2599	D	DBD	1.0
7	AREA	139	D	DBD	0.1
9	SEGM	16337	B	DBD	6.5
10	CAPXSEGM	1	D	SEGM	0.0
12	FLD	16426	C	SEGM	1.0
14	MAR	16426	C	FLD	1.0
17	LCHILD	2687	B	SEGM	0.2
20	XDFLD	134	B	LCHILD	0.0
33	PSB	1840	A	HEADER	0.4
35	PCB	9190	B	PSB	5.0
37	SS	75274	B	PCB	8.2
39	SF	1105	B	SS	0.0
41	DBDXREF	8886	D	PSB	4.8

SEGMENT	WITHIN EXISTING HEADER	DUPLICATES NOT INSERTED
DBD	71	0
PSB	72	0

ESTIMATED SPACE REQUIREMENT TO HOLD INSERTED SEGMENTS

DSG	BLKSIZE	BLOCKS
A	4096	596
B	4096	9343
C	4096	8214
D	4096	236

DSG	RECORDS
L	8886
X	4230

SECONDARY INDEX	RECORDS
DFSCX000	8886

## ライブラリー統計報告書

IMS Catalog Import ユーティリティーは、z/OS データ・セット・ユーティリティー IEBCOPY を呼び出して ACB、DBD、および PSB ライブラリーをコピーします。IEBCOPY ユーティリティーは、コピー操作に関する統計とメッセージを表示します。IEBCOPY ユーティリティーについて詳しくは、このトピックの末尾近くにある関連リンクを参照してください。

## 戻りコード

IMS Catalog Import ユーティリティーは以下のコードを返します。

- 0 すべての操作が正常に終了
- 4 1 つ以上の警告メッセージが出された
- 8 以上  
重大エラーでプログラムが中止された

---

## 第 9 章 IMS Catalog Directory Recovery ユーティリティー (DFS3RU00)

IMS ディレクトリー・データ・セットを再作成したり、IMS カタログからアクティブ・リソースを読み取ることによって IMS データベースおよびプログラムの ACB をリカバリーしたり、IMS ディレクトリー・データ・セットをクリーンアップしたりするには、IMS™ Catalog Directory Recovery ユーティリティー (DFS3RU00) を使用します。このユーティリティーは、DDL を使用している IMS 管理の ACB 環境で使用され、ACBLIB から IMS ディレクトリーを作成することはできません。

DFS3RU00 ユーティリティーには、以下の 4 つの基本機能があります。

- IMS ディレクトリー・データ・セットの再作成
- すべての IMS データベースおよびプログラムのための ACB のリカバリー
- IMS ディレクトリー・データ・セットへのオンライン・ランタイム・ブロックの書き込み
- IMS ディレクトリー・データ・セットのクリーンアップ

再作成、リカバリー、および書き込みの各機能は、SYSIN DD 制御ステートメントに MBR=ALL を指定することにより有効になります。

ディレクトリー・オンライン更新が異常終了したときに、DFS3RU00 ユーティリティーをサブミットして IMS ディレクトリー・データ・セットをクリーンアップすることもできます。

IMS カタログと IMS ディレクトリー・データ・セットの両方が失われる災害復旧シナリオの場合は、まず最初に IMS カタログ・データベースをリカバリーしてから、このユーティリティーを実行してディレクトリー・データ・セットを再作成して復元する必要があります。

サブセクション:

- 『制約事項』
- 416 ページの『前提条件』
- 416 ページの『要件』
- 416 ページの『推奨事項』
- 416 ページの『入力と出力』
- 416 ページの『JCL 仕様』
- 418 ページの『戻りコード』

### 制約事項

DFS3RU00 ユーティリティーは、DL/I バッチ領域として実行されます。主記憶データベース (MSDB) および共用副次索引データベースはサポートされません。

## 前提条件

災害復旧の場合は、最初に、IMS カタログをリカバリーする必要があります。IMS カタログは、DFS3RU00 ユーティリティへの入力として機能します。

IMPORT DEFN SOURCE(CATALOG) コマンドまたは DDL 定義の活動化プロセスがディレクトリー・データ・セットを更新中に DFS3RU00 ユーティリティがディレクトリー・データ・セットをクリーンアップしないようにします。

## 要件

DFS3RU00 ユーティリティでは、IMS.PROCLIB データ・セットの DFSDFxxx メンバーにアクセスする必要があります。

DFS3RU00 ユーティリティを実行する前に、IMS カタログ DBD (DFSCD000、DFSCX000) および PSB (DFSCP000、DFSCP001) がリカバリー済みであることを確認してください。そうであれば、リカバリーが失敗した場合にユーティリティを再実行できます。DFS3RU00 ユーティリティは、カタログ DBD および PSB をディレクトリー・データ・セットに書き込みます。ディレクトリー・データ・セットがすでに存在する場合には、IMS.PROCLIB データ・セットの DFSDFxxx メンバーに ACBMGMT=CATALOG を指定して、IMS カタログと、ACB の IMS 管理を使用可能にします。そうでない場合は、ACBMGMT=ACBLIB を指定してください。

## 推奨事項

DFS3RU00 ユーティリティを使用してディレクトリー・データ・セットをクリーンアップする場合は、JCL および IMS システムで内部リソース・ロック管理 (IRLM) を使用可能にして、ディレクトリー・データ・セットを更新するプロセスとの競合を回避します。

IRLM が使用されない場合は、QUERY MEMBER TYPE(IMS) SHOW() コマンドを発行して、IMPORT および DDL 定義の活動化処理の状況を確認します。

## 入力と出力

DFS3RU00 ユーティリティは、以下の入力を使用します。

- SYSIN DD ステートメント
- IMS.PROCLIB データ・セットの DFSDFxxx メンバーの CATALOG セクション

DFS3RU00 ユーティリティは、以下の出力を生成します。

- リカバリーされた 1 つ以上の IMS ディレクトリー・データ・セット
- ブートストラップ・データ・セット

## JCL 仕様

EXEC ステートメント

以下の JCL は、DFS3RU00 ユーティリティの実行に使用できる PARM パラメーターを示しています。IMS.PROCLIB データ・セット内の DFSDFxxx メンバーの 3 文字の接尾部を指定するには、DFSDF= パラメーターを使用します。

```
//BATCH EXEC PGM=DFSRR00,  
// PARM=(DLI,DFS3RU00,DFSCP000,,,,,,,,,Y,N,,,,,,,,,'DFSDF=xxx')
```

IRLM サポートを使用可能にするために、以下の例に示すように、EXEC ステートメントに PARM パラメーターを指定できます。

```
// PARM=(DLI,DFS3RU00,DFSCP000,,,,,,,,,Y,Y,irldid,,,,,,,,,'DFSDF=xxx')
```

**DD ステートメント**

#### **STEPLIB DD**

IMS 中核モジュールと必須の IMS モジュールが入っている IMS.SDFSRESL ライブラリーを指します。IMS.SDFSRESL に連結された無許可ライブラリーが含まれているために STEPLIB DD ステートメントが無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

#### **DFSRESLB DD**

IMS SVC モジュールが入っている許可ライブラリーを指します。

IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) による許可が必要です。JOB LIB DD および STEPLIB DD ステートメントは、IMS バッチには許可は不要です。

#### **PROCLIB DD**

IMS カタログ内にあるレコードのデフォルトの保存基準を定義する IMS.PROCLIB データ・セットの DFSDFxxx メンバーを指定します。

#### **IMS DD**

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。ACB の IMS 管理が使用可能である場合、IMS DD ステートメントは無視されます。

#### **IEFRDER DD**

プライマリー IMS ログ・データ・セットを定義します。

#### **DFSVSAMP DD**

バッファ・プール・パラメーター・データ・セットを定義します。

#### **SYSPRINT DD**

出力レポート・データ・セットを定義します。

#### **SYSIN DD**

入力制御ステートメントを定義します。ユーティリティは、次の SYSIN DD 制御ステートメントを受け入れます。

```
MBR=ALL
```

MBR=ALL を指定すると、ユーティリティは、IMS カタログからすべてのアクティブ IMS リソースのメタデータを読み取り、オンライン・ランタイム・ブロックをリカバリーし、それらを IMS ディレクトリー・データ・セットに書き込みます。

## 戻りコード

IMS Catalog Recovery ユーティリティは、以下のいずれかの戻りコードを使用して実行を完了します。

- 0 エラーまたは例外条件はありません。
- 4 データベース・タイプでのエラー、または IMS オンライン・ブロック作成障害。DFS3RU00 ユーティリティでは、MSDB および共用副次索引データベースはサポートされません。これらのアクセス・タイプ・データベースおよびプログラムを参照する ACB、またはブロック・ビルダー障害リソースの ACB を復元するためには、ACBGEN を実行してから、Catalog Populate ユーティリティ (DFS3PU00) を実行して、これらのメンバーをディレクトリー・データ・セットにリカバリーしてください。
- 8 制御ステートメントのエラー。
- 12 ジョブ報告書の記述を参照してください。

## JCL の例


以下のサンプル JCL は、すべてのデータベースおよびプログラムの ACB をリカバリーします。省略符号は、ユーザー値に置き換える必要があります。

```
//BATCH EXEC PGM=DFSRR00,  
// PARM=(DLI,DFS3RU00,DFSCP000,,,,,,,,,Y,N,,,,,,,,, 'DFSDF=001')  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//IMS DD DSN=IMS.PSBLIB,DISP=SHR  
// DD DSN=IMS.DBDLIB,DISP=SHR  
//IEFRDER DD ...  
//DFSVSAMP DD ...  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
MBR=ALL
```

以下のサンプル JCL は、ディレクトリー・データ・セットをクリーンアップします。PSB 名 DFS3DCU0 は、PARM= パラメーター・リストに指定する必要があります。制御カードは必要ありません。


```
//BATCH EXEC PGM=DFSRR00,  
// PARM=(DLI,DFS3RU00,DFS3DCU0,,,,,,,,,Y,irlmid,,,,,,,,, 'DFSDF=CT2')  
//STEPLIB DD DSN=IMSTESTL.TNUC0,DISP=SHR  
// DD DSN=IMSTESTL.TNUCT,DISP=SHR  
// DD DSN=IMSB LD.I14RTSMM.CRESLIB,DISP=SHR  
// DD DSN=IMSTESTG.IMSTESTS.DYNALLOC,DISP=SHR  
// DD DSN=IMSTESTG.IMSTESTL.DYNALLOC,DISP=SHR  
//DFSRESLB DD DSN=IMSTESTL.TNUC0,DISP=SHR  
// DD DSN=IMSB LD.I14RTSMM.CRESLIB,DISP=SHR  
// DD DSN=IMSTESTG.IMSTESTS.DYNALLOC,DISP=SHR  
// DD DSN=IMSTESTG.IMSTESTL.DYNALLOC,DISP=SHR  
//PROCLIB DD DSN=IMSVS.PROCLIB2,DISP=SHR  
// DD DSN=USER.PRIVATE.PROCLIB,DISP=SHR  
// DD DSN=IMSTESTG.I14RTSMM.PROCLIB,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSABEND DD SYSOUT=*  
//SYSOUT DD SYSOUT=A  
//
```

関連概念:

 アプリケーション制御ブロック (ACBGEN) の作成 (データベース管理)

関連タスク:

 [IMS が ACB を管理する場合の IMS カタログのリカバリー \(データベース管理\)](#)

 [IMS ディレクトリー・データ・セットのクリーンアップ \(データベース管理\)](#)

関連資料:

 [QUERY MEMBER コマンド \(コマンド\)](#)





---

## 第 10 章 IMS Catalog Partition Definition Data Set ユーティリティー (DFS3UCD0)

IMS Catalog Partition Definition Data Set ユーティリティー (DFS3UCD0) は、IMS カタログ区画定義データ・セットを作成して、データを取り込みます。このデータ・セットは、データベース区画の管理に DBRC が使用されない場合に IMS カタログ・データベース区画に関する情報を保管します。

### 制約事項

このユーティリティーについて文書化されている制約事項はありません。

### 前提条件

このユーティリティーについて文書化されている前提条件はありません。

### 要件

指定された区画定義データ・セットが空の場合、このユーティリティーがその区画定義データ・セットを作成します。そのデータ・セットに既にデータが含まれている場合、既存の情報は上書きされます。

### 推奨事項

可能であれば、IMS カタログ区画情報の管理には DBRC を使用してください。

### 入力と出力

DFS3UCD0 ユーティリティーは、以下の入力データ・セットと出力データ・セットを使用します。

- ユーティリティーからの入出力を含む DFSHDBSC データ・セット。
- ユーティリティーの入力データ・セットとして使用される IMS DBDLIB データ・セット。
- ユーティリティーからのメッセージを受け取る SYSPRINT データ・セット。
- ユーティリティー制御ステートメントを含む SYSIN データ・セット。

### JCL 仕様

**EXEC** ステートメント

```
//S1 EXEC PGM=DFS3UCD0
```

**DD** ステートメント

#### **JOBLIB/STEPLIB**

IMS.SDFSRESL データ・セットの位置。ここには、このユーティリティーの実行可能モジュールが含まれています。

#### **DFSHDBSC**

ユーティリティーへの入力およびユーティリティーからの出力に使用されるデータ・セット名。指定できるデータ・セット名は 1 つのみで、

連結データ・セットは無効です。このデータ・セットの DCB 値は RECFM=F および LRECL=80 です。

#### IMS

IMS DBDLIB データ・セット。

#### SYSPRINT

このユーティリティーによって生成されたメッセージを受け取るデータ・セット。 **SYSPRINT** データ・セットの DCB パラメーターは RECFM=FB および LCRECL=133 です。

#### SYSIN

このユーティリティーの入力パラメーターを含むデータ・セット。

**SYSIN** データ・セットの DCB パラメーターは RECFM=FB および LRECL=80 です。

このユーティリティーの **SYSIN** DD ステートメントでは、**HALDB** および **PART** ステートメントのみをサブミットします。

#### HALDB ステートメント

このステートメントは、IMS カタログの HALDB マスター名を指定します。

```
▶▶ HALDB=(NAME=name)▶▶
```

#### NAME

IMS カタログ区画定義データ・セットで定義している IMS カタログ HALDB の名前を指定します。

#### PART ステートメント

このステートメントは、IMS カタログ HALDB 内の区画 (1 つ以上) の構造を指定します。

```
▶▶ PART=(NAME=name, PART=name, DSNPREFIX=prefix  
        , KEYSTCHAR=value  
        , KEYSTHEX=value)  
        BLOCKSIZE(4096, value)  
        FBFF(0, value)  
        FSPF(0, value)  
▶▶
```

#### NAME

この区画の定義対象の HALDB の名前を識別します。

#### PART

HALDB 区画名を指定します。この値には、長さ 7 文字までの英数字を指定できます。先頭の文字は英字でなければなりません。

#### **DSNPREFIX**

区画に含まれる区画データ・セットのデータ・セット名接頭部を指定します。この値には 37 文字までを指定でき、有効な JCL データ・セット名でなければなりません。

#### **KEYSTCHAR**

この値は、この区画の HALDB 区画ハイ・キー値です。ハイ・キーの長さをルート・キーより長くすることはできません。ハイ・キーの長さがルート・キーより短い場合、定義されたルート・キーの長さの長さまで、ハイ・キー値に X'FF' バイトが繰り返し埋め込まれます。それぞれの区画ハイ・キー値は、HALDB 内で固有でなければなりません。この値は 256 文字までの長さにすることができます。

このパラメーターまたは **KEYSTHEX** が必須です。このパラメーターは **KEYSTHEX** と同時には指定できません。

#### **KEYSTHEX**

HALDB 区画ハイ・キー値を 16 進形式で指定します。512 文字までの長さにすることができます。このパラメーターまたは **KEYSTCHAR** が必須です。このパラメーターは **KEYSTCHAR** と同時には指定できません。

#### **BLOCKSZE**

OSAM データ・セットのブロック・サイズを指定します。この値は 32766 までの偶数でなければなりません。この値は OSAM データ・セットのみに使用します。このパラメーターのデフォルト値は 4096 です。このパラメーターには最大 10 個の値 (DBD で定義されている各データ・セット・グループごとに 1 つずつ) を指定できます。複数の値はコンマで区切ります。

#### **FBFF**

このデータ・セット・グループ内の n 番目ごとの制御インターバルまたはブロックが、データベースのロードまたは再編成操作時にフリー・スペースとして残されることを指定します。この値には、0 から 100 までの、1 を除く任意の整数を指定できます。デフォルト値は 0 です。0 は、ロードまたは再編成操作時にフリー・スペースを保持しないことを指定します。

#### **FSPF**

このデータ・セット・グループ内に保持する必要がある、フリー・スペースの最小パーセントを指定します。この値には 0 から 99 までの任意の数を指定できます。デフォルトは 0 です。

### 戻りコード

このユーティリティは、完了後に DFS4353I メッセージを生成します。このメッセージには以下のいずれかの戻りコードが含まれます。

- 0 処理は正常完了。
- 40 BPESTART マクロが正常に完了しませんでした。DFS4353I の理由コード・フィールドに BPESTART マクロからの戻りコードが含まれています。

- 44 ストレージを取得できない。DFS4353I の理由コードが取得できなかったストレージを示します。
- 1 BPEPARSE 文法
  - 2 構文解析出力ブロック
  - 3 HALDB 定義記述子ブロック
  - 4 DBD レコード・ストレージ
- 48 ユーティリティの SYSIN ステートメントの構文解析中にエラーが発生しました。DFS4353I の理由コード・フィールドに BPEPARSE マクロからの戻りコードが含まれています。
- 52 ユーティリティの SYSIN ステートメントの妥当性検査中にエラーが発生しました。
- 56 BPERDPDS マクロによるユーティリティの SYSIN ステートメントの読み取り中に、エラーが発生しました。DFS4353I の理由コード・フィールドに BPERDPDS マクロからの戻りコードが含まれています。
- 60 SYSPRINT のオープン中にエラーが発生しました。
- 76 HALDB 区画定義データ・セット (DFSHDBSC) を処理できませんでした。理由コードがこの処理エラーの発生時点を示しています。
- 72 (RC\_OPENDEFDS) - データ・セットのオープン中にエラーが発生しました。
  - 80 (RC\_CLOSEDEFDS) - データ・セットのクローズ中にエラーが発生しました。
  - 84 (RC\_RDJCBDEFDS) - データ・セットの処理中に RDJFCB マクロ・エラーが発生しました。
- 88 DBDLIB を処理できませんでした。理由コードが潜在的な原因を示しています。
- 1 IMS DD ステートメントが見つかりませんでした。
  - 2 IMS DD オープン・エラーが発生しました。
  - 3 指定されたデータベースが DBDLIB 内で見つかりませんでした。
  - 4 指定されたデータベースが HALDB ではありません。

### サンプル JCL ステートメント

```
//S1      EXEC PGM=DFS3UCD0,REGION=0M
//STEPLIB DD  DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSHDBSC DD DSN=...,DISP=
//SYSPRINT DD SYSOUT=*
//IMS     DD  DSN=IMS.DBDLIB,DISP=SHR
//SYSIN   DD  *
HALDB=(NAME=DFSCD000)
PART=(NAME=DFSCD000,PART=xxxxxxx,
      DSNPREFIX=xxxxxxx,
      KEYSTCHAR=xxxxxxx)
/*
```

---

## 第 11 章 IMS Catalog Populate ユーティリティ (DFS3PU00)

IMS Catalog Populate ユーティリティ (DFS3PU00) を使用して、ACB ライブラリー・データ・セットから IMS カタログのデータベース・データ・セットに DBD および PSB のインスタンスをロードまたは挿入し、これらの更新も行います。IMS システム内で ACB の IMS 管理が使用可能な場合、ユーティリティはさらに、IMS によって管理されている IMS ディレクトリー・データ・セットにアクティブなアプリケーション制御ブロック (ACB) を追加します。

DFS3PU00 ユーティリティを使用すると、IMS カタログを使用しなくても IMS カタログ・データ・セットの必要サイズを見積もることもできます。このユーティリティは、入力 ACB ライブラリー内のメンバーをスキャンして、サイズの見積もりを生成します。

DFS3PU00 ユーティリティをロード・モードで実行するか、更新モードで実行するか、あるいは分析専用モードで実行するかは、ユーティリティ JCL で指定する IMS カタログ PSB によって決定されます。

- ロード・モード (PSB DFSCPL00 で指定)
- 更新モード (PSB DFSCP001 で指定)
- 分析専用モード (DFSCP000 で指定)

ACB の IMS 管理が使用可能な場合は、生成された ACB をオンライン IMS システムで使用できるようにするために、SYSINP DD ステートメントで MANAGEDACBS= 制御ステートメントを指定する必要があります。DFS3PU00 ユーティリティは、指定に応じて、ACB を後で活動化するために IMS カタログのステージング・データ・セットに追加するか、あるいは IMS システムがオフラインである場合または管理対象 ACB を初めてセットアップしている場合は、それらを IMS ディレクトリー・データ・セットに直接追加します。IMS は、始動時にディレクトリー・データ・セットから ACB をロードします。

動的オプション (DOPT) PSB が入力 ACB ライブラリー内の唯一のリソース・タイプである場合、システムをオフラインにしなくても、ユーティリティで DOPT PSB を追加することができます。これにはいくつかの制約事項が適用されます。434 ページの『制御ステートメント』に記載されている SHARE キーワードを参照してください。

IMS カタログのいずれかのデータ・セット (IMS ディレクトリーのデータ・セットを含む) が作成されていない場合、DFS3PU00 ユーティリティが自動的に作成します。IMS ディレクトリー (内部 IMS 構造) は、IMS カタログ内でアクティブとしてフラグが立てられている、DBD インスタンスおよび PSB インスタンスのランタイム制御ブロックを保持するために使用されます。

このユーティリティが IMS カタログ・データ・セットに割り振るスペース量は、入力 ACB ライブラリーのメンバーと、DFSDFxxx PROCLIB メンバーのカタログ・セクション内の SPACEALLOC パラメーターに指定された値とに基づいて決まります。ユーティリティが IMS ディレクトリー・データ・セット (PDSE デー

タ・セット) に割り振るスペース量も、ACB ライブラリーのサイズに基づいて決まりますが、必要に応じて動的に増加されます。

DFS3PU00 ユーティリティーの出力には、IMS カタログにロードされるレコード・セグメントに関する統計を含む報告書があります。この報告書には、セグメントの数とタイプに関する情報、および IMS カタログの各データ・セット・グループに必要な DASD ストレージの見積もり量が記載されます。IMS カタログ・データ・セットの作成前に、これらのデータ・セットが使用する DASD ストレージの量を知る必要がある場合、IMS カタログへのデータの取り込みを行わずに DFS3PU00 ユーティリティーを実行して、統計報告書のみを生成することができます。DFS3PU00 ユーティリティーを分析専用モードで実行するには、ユーティリティー JCL にこのユーティリティーの PSB として DFSCP000 を指定します。

このユーティリティーは、ACB ライブラリーのメンバーを評価した後に報告書を生成します。この報告書は、DFS3PU00 ユーティリティーが IMS カタログのロード時に生成するものと同じ報告書です。

DFS3PU00 ユーティリティーは、カタログ・レコードを、1 つ以上の ACB ライブラリーの ACB メンバーと、(データベース・タイプに応じて) DBD ライブラリーおよび PSB ライブラリー内の関連する DBD および PSB メンバーとから作成します。これらのレコードには、アプリケーション・プログラムおよびデータベースのメタデータが含まれています。

DFS3PU00 ユーティリティーは DL/I 領域で実行できます。または、このユーティリティーが既存の IMS カタログを更新する場合は、BMP 領域でも実行できます。このユーティリティーを DL/I バッチ領域で実行し、IMS カタログを共用する場合は、ユーティリティー JCL の EXEC パラメーターに IRLM サポートを指定する必要があります。次の例では、2 番目の Y と *irlmid* 値が IRLM サポートを示しています。

```
PARM=(DLI,DFS3PU00,DFSCP001,,,,,,,,,Y,Y,irlmid,,,,,,,,,'DFSDF=001')
```

DFS3PU00 ユーティリティーを実行する代わりに、ACB Generation and Catalog Populate ユーティリティー (DFS3UACB) を使用して IMS カタログにデータを取り込むことができます。DFS3UACB ユーティリティーは、アプリケーションとデータベース用の ACB ライブラリーの生成と、その後の IMS カタログへのデータの取り込みの両方を、同じジョブ・ステップで行います。

サブセクション:

- 427 ページの『制約事項』
- 427 ページの『前提条件』
- 427 ページの『要件』
- 428 ページの『推奨事項』
- 428 ページの『入力と出力』
- 430 ページの『JCL 仕様』
- 434 ページの『制御ステートメント』
- 439 ページの『IMS Catalog Populate ユーティリティーの統計報告書』
- 441 ページの『戻りコード』

## 制約事項

DFS3PU00 ユーティリティーに固有の制約事項は、現在文書化されていません。

## 前提条件

DFS3PU00 ユーティリティーが新規または変更されたアプリケーション・プログラムとデータベース用のメタデータを IMS カタログにロードするには、事前に、その新規または変更されたアプリケーション・プログラムとデータベース用の DBD、PSB、および ACB のすべての生成プロセスを完了しておく必要があります。

IMS カタログを初めてロードする場合は、DFS3PU00 ユーティリティーを実行する前に以下のステップが完了していることを確認します。

- IMS カタログの DBD および PSB ロード・モジュールが DBD および PSB ライブラリーに追加されている。
- IMS カタログの ACB メンバーが生成され、IMS.ACBLIB データ・セットにロードされている。
- IMS カタログの HALDB マスター・データベースおよび区画が、RECON データ・セット内、またはターゲット IMS カタログが DBRC によってサポートされていない場合は IMS カタログ区画定義データ・セット内で定義されている。

SHARE オプションを使用して DOPT PSB をオンライン IMS カタログに追加する前に、PDSE データ・セットの拡張共用を z/OS 内で使用可能にする必要があります。詳細については、z/OS: 複数システム環境における拡張 PDSE 共用の指定を参照してください。

## 要件

DFS3PU00 ユーティリティーは以下のデータ・セットにアクセスする必要があります。

- IMS カタログを使用可能にし、IMS カタログの別名を定義する DFSDFxxx メンバーを含む IMS.PROCLIB データ・セット
- 1 つ以上の IMS.ACBLIB データ・セット
- ACB ライブラリー・メンバーが論理的に関連したデータベースを参照する場合は、IMS.DBDLIB データ・セット
- ACB ライブラリー・メンバーが GSAM データベースを参照する場合は、IMS.DBDLIB データ・セットおよび IMS.PSBLIB データ・セット

IMS.DBDLIB および IMS.PSBLIB データ・セットが入力として含まれる場合、これらは入力 IMS.ACBLIB データ・セットの作成元となった DBD および PSB ライブラリーでなければなりません。

PSBLIB または DBDLIB データ・セットの必須メンバーが見つからない場合、このユーティリティーはエラー・メッセージを発行し、参照する側の PSB のレコードは IMS カタログ内に作成されません。見つからなかったカタログ・レコードは後で追加できます。それには、正しい ACB ライブラリーとともに必要な PSB または DBD メンバーを提供し、ユーティリティー JCL に DFSCP001 PCB を指定して更新モードで DFS3PU00 ユーティリティーを再実行します。

ACB の IMS 管理が使用可能な場合は、生成された ACB をオンライン IMS システムで使用できるようにするために、SYSINP DD ステートメントで MANAGEDACBS= 制御ステートメントを指定する必要があります。

IMS カタログを DBRC に登録する場合、IMS カタログの初期ロード後に IMS カタログのイメージ・コピーを作成する必要があります。IMS カタログを DBRC に登録しない場合、IMS は初期ロード後にイメージ・コピーを要求できません。ただし、初期ロード後にイメージ・コピーを作成しない場合、カタログをリカバリーする方法は再ロードのみです。

## 推奨事項

既存の IMS カタログを更新する場合、更新の完了時に IMS カタログ・データ・セットのイメージ・コピーを作成します。IMS カタログが DBRC に登録されている場合、DBRC コマンド GENJCL.IC を使用してカタログをバックアップできます。IMS カタログを IMS カタログ区画定義データ・セット内で定義した場合、標準のイメージ・コピー JCL を使用する必要があります。

ACBLIB メンバーをストレージ内に保持する必要があるため、ストレージ要件は ACBLIB メンバーの総数に大きく左右されます。異常終了 U1002 理由コード 4 が発行される原因となる GETMAIN 障害を防ぐには、ジョブの領域サイズの増加を検討してください。

## 入力と出力

DFS3PU00 ユーティリティは常に、データベースとアプリケーション・プログラム用の ACB メンバーを含む ACB ライブラリー・データ・セットから、および IMS.PROCLIB データ・セットの DFSDFxxx メンバーから入力を読み取ります。

データベースで論理関係が使用されている場合、IMS Catalog Populate ユーティリティは IMS.DBDLIB データ・セットからも入力を読み取ります。

GSAM データベースを使用し、なおかつ ACB ライブラリー・メンバーが GSAM データベースを参照する場合、DFS3PU00 ユーティリティはさらに、IMS.DBDLIB データ・セットおよび IMS.PSBLIB データ・セットの両方から入力を読み取ります。

DFS3PU00 ユーティリティの出力には以下のものがあります。

- IMS カタログ・データ・セット (DFSCD000) 内のレコード・セグメント。
- SYSPRINT データ・セットに対するメッセージおよび統計情報。
- MANAGEDACBS=SETUP が指定された場合は、以下のようになります。
  - IMS カタログ内の新規または更新済みのレコード・セグメント
  - IMS ディレクトリー・データ・セット
  - IMS カタログ内のアクティブな DBD インスタンスおよび PSB インスタンスに対応する IMS ディレクトリー・データ・セット内の ACB
- MANAGEDACBS=STAGE が指定された場合は、以下のようになります。
  - IMS カタログ内の新規または更新済みのレコード・セグメント
  - IMS ディレクトリー・データ・セットのステージング・データ・セット内の ACB



- MANAGEDACBS=UPDATE が指定された場合は、以下のようになります。
  - IMS カタログ内の新規または更新済みのレコード・セグメント
  - IMS カタログ内のアクティブな DBD インスタンスおよび PSB インスタンスに対応する IMS ディレクトリー・データ・セット内の ACB

以下のデータベース・データ・セットのいずれかが存在しない場合、DFS3PU00 ユーティリティーが自動的に作成します。

- 以下の DFSCD000 データベース・データ・セット:
  - IMS カタログのセグメント用の 4 つのデータ・セット
  - 間接リスト・データ・セット (ILDS)
  - 1 次索引データ・セット
- DFSCX000 副次索引データ・セット
- MANAGEDACBS=SETUP ステートメントが指定されている場合は、IMS ディレクトリー・データ・セット (IMS ディレクトリー・ブートストラップ・データ・セットを含む)。

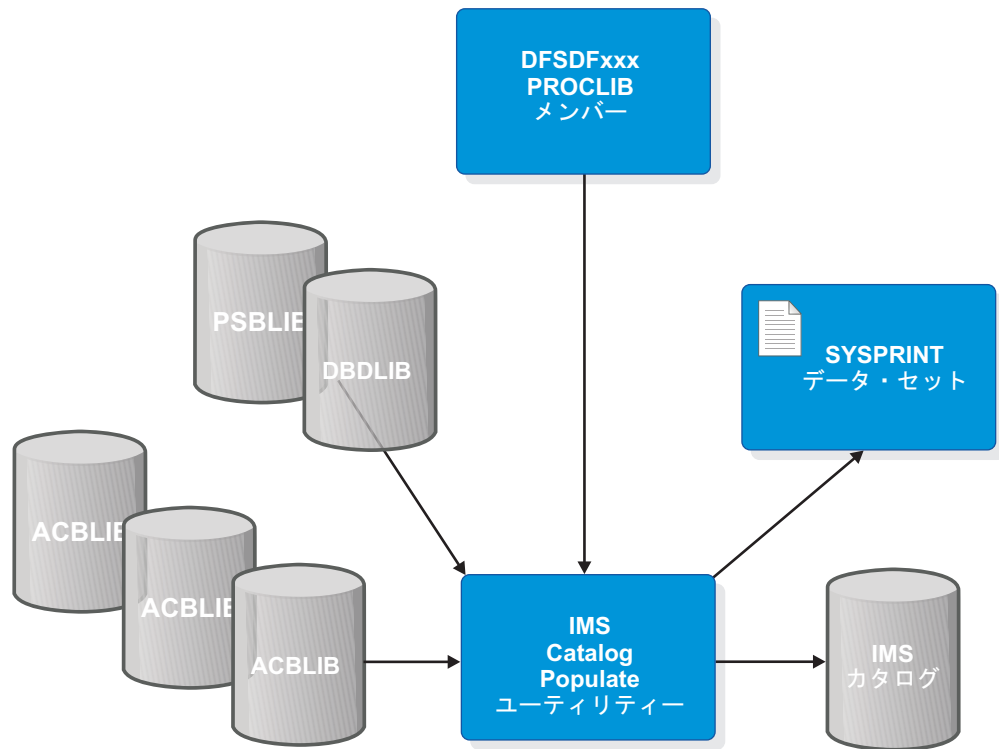


図 35. IMS Catalog Populate ユーティリティの入力と出力

## JCL 仕様

### DFS3PU00 ユーティリティ JCL ステートメント

次の JCL 例に示すように、DFS3PU00 ユーティリティの実行パラメータは IMS.PROCLIB データ・セットの DFSDFxxx メンバーを参照する必要があります。

IMSACB01 DD ステートメントにより、少なくとも 1 つの IMS.ACBLIB データ・セットをユーティリティ JCL に指定する必要があります。追加の ACBLIB データ・セットを含めるには、これらのデータ・セットを単一の DD ステートメント内で連結するか、または追加の IMSACBnn DD ステートメントを指定します (次の JCL 例に示すように、IMSACB02 および IMSACB03 DD ステートメントによって指定します)。追加の IMSACBnn DD 名には、末尾 2 つの文字位置に連続した番号を割り当てる必要があります。

名前が重複している重複 ACB メンバーは、ACB ライブラリーが単一の DD ステートメントで連結されているか、別個の DD ステートメントで個々に参照されているかに応じて異なる方法で処理されます。クローンの ACB ライブラリーを使用し

ている場合でも、いずれかのライブラリー内で重複 ACB メンバーが個々に生成されている可能性がある場合は、別個の DD ステートメントを使用してユーティリティー JCL でクローンの各 ACB ライブラリーを参照します。

ACB ライブラリーが別個の DD ステートメントによって個々に参照されている場合、DFS3PU00 ユーティリティーは名前が重複した ACB メンバーの ACB 生成タイム・スタンプを確認し、ACB 生成タイム・スタンプが前に処理された同じ名前の ACB メンバーと異なる場合にのみ、それらのメンバーを使用します。タイム・スタンプが同じ場合、名前が重複した ACB メンバーは無視されます。

ただし、ACB ライブラリーが単一の DD ステートメントで連結されている場合、DFS3PU00 ユーティリティーは名前が重複した ACB メンバーの ACB 生成タイム・スタンプを確認しません。連結された ACB ライブラリー内に重複したメンバー名が存在する場合、最初の ACB メンバーのみが使用され、名前が重複している後続の ACB メンバーは、タイム・スタンプが異なる場合でもすべて無視されます。

DLIBATCH または同等のプロシージャーには、ACB ライブラリーの生成に使用された PSB および DBD ライブラリー用の IMS DD ステートメントが既に含まれているはずです。また、このプロシージャーには、IMS.SDFSRESL データ・セット用の STEPLIB および DFSRESLB DD ステートメントと、IMS ログ・データ・セット用の IEFORDER および IEFORDER2 DD ステートメントも含まれているはずです。

以下は、DFS3PU00 ユーティリティーを使用した IMS カタログの初期ロードの実行に使用できる JCL ステートメントのサンプルです。この JCL は IMS カタログの PSB DFSCPL00 を指定します。この PSB が、レコードを IMS カタログにロードします。既存のレコードはすべて上書きされます。

```
//LOADCAT EXEC PGM=DFS3PU00,  
// PARM=(DLI,DFS3PU00,DFSCPL00,,,,,,,,,Y,N,,,,,,,,,'DFSDF=001')  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//IMS DD DSN=IMS.PSBLIB,DISP=SHR  
// DD DSN=IMS.DBDLIB,DISP=SHR  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//SYSABEND DD SYSOUT=* Dump data set  
//SYSPRINT DD SYSOUT=* Messages, statistics  
//IEFRDER DD ... Log data set  
//DFSVSAMP DD ... Buffer pool parameters  
//IMSACB01 DD ... First ACBLIB  
// DD ... Optional concatenated ACBLIB  
//IMSACB02 DD ... Optional additional ACBLIBs  
//IMSACB03 DD ...  
//SYSINP DD * ISRTLIST DUPLIST /*
```

以下は、DL/I バッチ・ジョブで DFS3PU00 ユーティリティを実行して IMS カタログを更新する場合に使用できる JCL ステートメントのサンプルです。この JCL は IMS カタログの PSB DFSCP001 を指定します。この PSB が、カタログ全体を置き換えずに IMS カタログにレコードを挿入します。IRLM サポートは示されていません。

```
//UPDTCAT EXEC PGM=DFS3PU00,
//                PARM=(DLI,DFS3PU00,DFSCP001,,,,,,,,,Y,N,,,,,,,,,
//                ,,'DFSDF=001')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=* Dump data set
//SYSPRINT DD SYSOUT=* Messages, statistics
//IEFRDER DD ... Log data set with catalog updates
//DFSVSAMP DD ... Buffer pool parameters
//IMSACB01 DD ... First ACBLIB
// DD ... Optional concatenated ACBLIB
//IMSACB02 DD ... Optional additional ACBLIBs
//IMSACB03 DD ...
```

以下は、BMP ジョブで DFS3PU00 ユーティリティを実行して IMS カタログを更新する場合に使用できる JCL ステートメントのサンプルです。この JCL は IMS カタログの PSB DFSCP001 を指定します。この PSB が、カタログ全体を置き換えずに IMS カタログにレコードを挿入します。

この例では、*imsid* が、ジョブが実行される IMS システムの ID です。

```
//UPDTCAT EXEC PGM=DFS3PU00,
//                PARM=(BMP,DFS3PU00,DFSCP001,,,,,,,,,imsid,,,,,)
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=* Dump data set
//SYSPRINT DD SYSOUT=* Messages, statistics
//IEFRDER DD ... Log data set with catalog updates
//DFSVSAMP DD ... Buffer pool parameters
//IMSACB01 DD ... First ACBLIB
// DD ... Optional concatenated ACBLIB
//IMSACB02 DD ... Optional additional ACBLIBs
//IMSACB03 DD ...
```

## DD ステートメント

### DFSRESLB DD

IMS SVC モジュールが入っている許可ライブラリーを指します。IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) で許可する必要があります。この DD ステートメントには、許可ライブラリー中になければならない IMS SVC モジュール用の許可ライブラリーを指定します。JOBLIB または STEPLIB ステートメントは、IMS バッチには許可は不要です。

### DFSVSAMP

バッファ・プール・パラメーター・データ・セットを定義します。

### IEFRDER DD

プライマリー IMS ログ・データ・セットを定義します。

### IEFRDER2 DD

セカンダリー IMS ログ・データ・セットを定義します。

### **IMS DD**

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。

この IMS DD ステートメントが必要になるのは、IMS カタログにロードされた IMS ACB ライブラリー・メンバーが GSAM データベースまたは論理的に関連したデータベースを参照する場合のみです。

### **IMSACB01 DD**

IMS カタログにデータを追加するために使用される ACB メンバーが入った ACB ライブラリー・データ・セットを定義します。この DD ステートメントは必須です。

### **IMSACBnn DD**

オプションの入力 ACB ライブラリー・データ・セットを追加で定義します。追加の ACB ライブラリーの DD 名には、*nn* の位置に連続した番号を割り当てる必要があります。

*nn* の値は、先行する IMSACB*nn* DD 名の *nn* の値に 1 を加えて決定されます。*nn* の有効値は 02 から 99 までです。

例えば、必須の IMSACB01 DD ステートメントの後に続く DD ステートメントの名前は、IMSACB02、IMSACB03、IMSACB04 などになります。

IMSACB*nn* DD 名の連続した番号が途切れていると、その途切れた部分より後の ACB ライブラリー DD ステートメントは無視されます。

### **PROCLIB DD**

ユーティリティに必要な、IMS カタログの各種属性を定義する DFSDFxxx メンバーが入った IMS.PROCLIB データ・セットを定義します。

### **STEPLIB DD**

IMS.SDFSRESL を指します。ここには、IMS 中核と必須の IMS モジュールが入っています。IMS.SDFSRESL に連結された無許可ライブラリーがあるために STEPLIB が無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

### **SYSABEND DD**

ダンプ・データ・セットを定義します。

### **SYSINP DD**

80 文字の固定長レコードを持つオプションの制御ステートメント順次データ・セット。位置 1 から 72 までの文字だけが読み取られます。

1 つ以上のレコードで、制御ステートメント・パラメーターをブランクまたはコンマで区切って指定することができます。

SYSINP DD ステートメントで指定できる制御ステートメントの説明については、434 ページの『制御ステートメント』を参照してください。

### **SYSPRINT DD**

出力メッセージ・データ・セットを定義します。

SYSPRINT DD ステートメントで DASD データ・セットまたはテープ・データ・セットを参照する場合、そのデータ・セットのブロック・サイズを **DCB** パラメーターの **BLKSIZE** サブパラメーターで制御できます。BLKSIZE 値を指定する場合、その値は正確に 121 の倍数とする必要があります、そうでない場合はシステム ABEND013-20 が発生する可能性があります。DASD データ・セット

の BLKSIZE を省略すると、システムが決定するブロック・サイズが使用されます。LRECL に指定する値に関係なく、ユーティリティーは常に 121 のレコード長を使用します。

## 制御ステートメント

SYSINP DD ステートメントを使用して、1 つ以上のレコードで、以下の制御ステートメント・パラメーターをブランクまたはコンマで区切って指定することができます。

### DUPLIST

既に IMS カタログにあるインスタンスと重複するために IMS カタログに追加されない、入力 ACB ライブラリー内の各 DBD リソースまたは PSB リソースを DFS3PU00 ユーティリティーでリストすることを指定します。IMS カタログ内のリソースの重複するインスタンスごとに、ユーティリティーはメッセージ DFS4436I を出力します。

MANAGEDACBS= ステートメントが指定されており、ACB の IMS 管理が使用可能な場合、ユーティリティーは、既に IMS ディレクトリーにあるリソースのインスタンスと重複するために IMS ディレクトリーまたはステージング・データ・セットに追加されない、各 DBD リソースまたは PSB リソースもリストします。

MANAGEDACBS= ステートメントで UPDATE パラメーターが指定された場合、ユーティリティーは重複するインスタンスごとにメッセージ DFS4531I を出力します。

UPDATE パラメーターで LATEST が指定されたかデフォルトとして受け入れられた場合、ユーティリティーは、IMS ディレクトリー内のリソース・インスタンスのタイム・スタンプがより後の時刻であるために IMS ディレクトリーに追加されないインスタンスごとに、メッセージ DFS4522I も出力します。

MANAGEDACBS= ステートメントで STAGE パラメーターが指定された場合、ユーティリティーは、重複するインスタンスであるためにステージング・データ・セットにコピーされないリソースごとに、メッセージ DFS4542I を出力します。

STAGE パラメーターで LATEST が指定されたかデフォルトとして受け入れられた場合、ユーティリティーは、IMS ディレクトリー内のリソース・インスタンスのタイム・スタンプがより後の時刻であるためにステージング・データ・セットに追加されないインスタンスごとに、メッセージ DFS4539I も出力します。

### ERRORMAX=*n*

*n* 個を超えるメッセージが、特定の DBD および PSB が IMS カタログに書き込まれるメタデータを持つことを妨げるエラーを示す場合、IMS Catalog Populate ユーティリティーを終了します。メタデータの重複するインスタンスは、この制限にカウントされません。このオプションを省略する場合、制限はありません。

### RESOURCE\_CHKP\_FREQ=*n*

チェックポイントの間に挿入する DBD および PSB リソース・インスタンスの数を指定します。*n* は、1 から 99999999 までの 1 桁から 8 桁の数値です。デフォルト値は 100 です。

### **SEGMENT\_CHKP\_FREQ=n**

チェックポイントの間に挿入するセグメントの数を指定します。この数に達すると、IMS はチェックポイントを発行する前に、現在処理されているリソース・インスタンスのセグメントすべての挿入を完了します。n は、1 から 99999999 までの 1 桁から 8 桁の数値です。デフォルト値は 1000 です。

注: 最初のチェックポイント頻度数に達すると、チェックポイントが発行され、カウンターは 0 にリセットされます。

### **ISRTLST**

ACB の IMS 管理が使用可能の場合、ユーティリティーは、IMS ディレクトリーに追加されるか、後で IMS ディレクトリーにインポートするためにステージング・データ・セットに保管される各 DBD リソースまたは PSB リソースもリストします。

ユーティリティーは、以下のメッセージを出力することにより、各リソースに対して実行されるアクションを識別します。

#### **DFS4520I**

リソースは、新規リソースとして IMS ディレクトリーに追加されました。

#### **DFS4521I**

リソースのこのインスタンスは、IMS ディレクトリー内のリソースの既存インスタンスに取って代わりました。

#### **DFS4537I**

リソースは、ステージング・データ・セットへコピーされました。リソースは、後で IMS ディレクトリーにインポートされる時、IMS ディレクトリーに新規リソースとして挿入されます。

#### **DFS4538I**

リソースは、ステージング・データ・セットへコピーされました。リソースは、後で IMS ディレクトリーにインポートされる時、IMS ディレクトリー内のリソースの既存インスタンスに取って代わります。

### **MANAGEDACBS=**

MANAGEDACBS 制御ステートメントは、以下のアクションを実行するために使用します。

- データベースおよびプログラム・ビューのランタイム・アプリケーション制御ブロック (ACB) を管理するために IMS をセットアップします。
- 新規 ACB または ACB ライブラリー・データ・セットから変更した ACB を使用して、ACB を管理する IMS システムを更新します。
- ACB を管理する IMS システムに後でインポートするために、ACB ライブラリーからステージング・データ・セットに ACB を保管します。

MANAGEDACBS ステートメントは、以下の構文図に従って指定することができます。

MANAGEDACBS= ステートメントで複数のパラメーターを指定する場合に限り、括弧が必要です。

以下のリストでは、MANAGEDACBS= ステートメントで指定できるパラメーターについて説明します。

## SETUP

アプリケーション制御ブロック (ACB) を管理するために IMS に必要な IMS ディレクトリー・データ・セットを作成します。IMS ディレクトリー・データ・セットのすべての既存インスタンスは置き換えられます。

指定した MANAGEDACBS=SETUP でユーティリティーが正常に実行されるまで、DFSDFxxx PROCLIB メンバーの CATALOG セクションに ACBMGMT=CATALOG を指定しないでください。ACB の IMS 管理を使用可能にする前に、IMS ディレクトリーを作成して、アクティブ ACB をロードしておく必要があります。

SETUP が指定された場合、ユーティリティーは、入力 ACB ライブラリー内にある ACB を IMS ディレクトリー・データ・セットに挿入します。

ユーティリティー JCL で IMS カタログ PSB DFSCP001 が指定された場合、ユーティリティーは、新規または変更されたすべての DBD および PSB を IMS カタログに挿入します。

ユーティリティー JCL で IMS カタログ PSB DFSCPL00 が指定された場合、ユーティリティーは既存のすべてのカタログ・レコードを削除し、IMS カタログを再ロードします。

ユーティリティーは、IMS ディレクトリーに ACB を追加するとき、IMS カタログ内の対応する DBD インスタンスおよび PSB インスタンスにアクティブとしてのフラグを立てます。

## STAGE

ACB を入力 ACB ライブラリーからステージング・データ・セットに保管します。ステージング・データ・セットに保管された ACB は、IMPORT DEFN SOURCE(CATALOG) コマンドで IMS システムに追加されるまで活動化されません。

STAGE を指定するときは、ユーティリティー JCL で IMS カタログ PSB DFSCP001 を指定する必要があります。

MANAGEDACBS=STAGE 制御ステートメントが指定されたときにステージング・データ・セットが存在する場合、ユーティリティーは既存のステージング・データ・セットを使用します。既にステージング・データ・セット内にある ACB はすべて保持されます。新規リソースをステージングするときにはステージング・データ・セットをスクラッチして再作成するには、MANAGEDACBS=(STAGE,DELETE) を指定します。DELETE が指定される場合、既存のデータ・セット内にある ACB はすべて失われます。

STAGE が指定された場合、DFS3PU00 ユーティリティーは IMS システム内にまだ存在しない、ACB ライブラリー内のすべての ACB を、ステージング・データ・セットにコピーします。ACB ライブラリー内の ACB が既に IMS システム内に存在する場合、ユーティリティーは、その ACB のタイム・スタンプが UNCOND パラメーターや LATEST パラメーターの基準を満たすかどうかに基づいて、ACB をステージング・データ・セットに保管します。



IMS カタログの初期ロードを行うとき、または ACB の IMS 管理を使用可能にする前に、STAGE を指定しないでください。STAGE パラメーターは、既存の IMS ディレクトリー・データ・セットに対する更新のステージング専用です。

MANAGEDACBS= の後に複数のパラメーターを指定する場合には、括弧で囲む必要があります。例: (STAGE,LATEST)。STAGE のみを指定する場合は、括弧を省略することができます。

#### **DELETE**

ステージング・データ・セットがどのオンライン IMS システムにも割り振られていない場合は、リソースをステージング・データ・セットに追加する前に、ステージング・データ・セットをスクラッチして再作成します。既存のステージング・データ・セット内にある ACB はすべて失われます。

#### **LATEST**

IMS システム内に ACB が既に存在する場合、ACB ライブラリー内のインスタンスのタイム・スタンプが IMS システム内の ACB より新しい場合を除き、ACB ライブラリー内の ACB のインスタンスをステージング・データ・セットに保管しません。

これはデフォルトです。

#### **UNCOND**

IMS システム内に ACB が既に存在する場合、ACB ライブラリー内の ACB のタイム・スタンプが IMS システム内の ACB のタイム・スタンプと同じである場合を除き、ACB ライブラリー内の ACB のインスタンスをステージング・データ・セットに無条件に保管します。

#### **UPDATE**

既存の IMS ディレクトリー・システム・データ・セットを排他モードで直接更新します。ACB はステージング・データ・セットに入れられません。

推奨事項: UPDATE パラメーターを指定する場合、IMS ディレクトリーを使用する IMS システムをすべてシャットダウンしてください。UPDATE が指定される場合、IMS Catalog Populate ユーティリティーには IMS ディレクトリーに対する排他的アクセスが必要です。

また、ユーティリティーは、IMS ディレクトリーの更新時にオンライン IMS システムに通知しません。そのため、更新された ACB をロードするために、ユーティリティーが IMS ディレクトリーを更新するときにオンラインになっている IMS システムを再始動する必要があります。

UPDATE が指定された場合、DFS3PU00 ユーティリティーは、まだ IMS システム内にはない ACB を IMS ディレクトリーに無条件に挿入します。IMS システム内に ACB が存在する場合、その ACB のインスタンスは、ACB ライブラリー内のインスタンスのタイム・スタンプ、および UNCOND パラメーターまたは LATEST パラメーターの指定に従って置き換えられます。

IMS カタログの初期ロードを行うとき、または ACB の IMS 管理を使用可能にする前に、UPDATE を指定しないでください。UPDATE パラメーターは、既存の IMS ディレクトリー・データ・セットの更新専用です。

UPDATE を指定するときは、ユーティリティー JCL で IMS カタログ PSB DFSCP001 を指定する必要があります。

MANAGEDACBS の後に複数のパラメーターを指定する場合には、括弧で囲む必要があります。例: (UPDATE,LATEST)。UPDATE のみを指定する場合は、括弧を省略することができます。

#### LATEST

IMS システム内に ACB が既に存在する場合、その ACB は、ACB ライブラリー内のインスタンスのタイム・スタンプが IMS システム内の ACB より新しい場合を除き、ACB ライブラリー内の ACB のインスタンスで置き換えられません。

これはデフォルトです。

#### UNCOND

IMS システム内に ACB が既に存在する場合、その ACB は、ACB ライブラリー内の ACB のタイム・スタンプが IMS システム内の ACB のタイム・スタンプと同じである場合を除き、ACB ライブラリー内の ACB のインスタンスで無条件に置き換えられます。

#### SHARE

動的オプション (DOPT) PSB の場合のみ、共用モードに必要な IMS ディレクトリー・データ・セットを割り振ることにより、オンライン処理を中断せずに DOPT PSB を IMS カタログに追加できるようにします。

ユーティリティーが IMS カタログをロードする場合は、SHARE を指定しないでください。これが該当するのは、ユーティリティー JCL の実行パラメーターに DFSCPL00 (IMS カタログのロード PSB) の指定がある場合です。

入力 ACBLIB データ・セット内に DOPT PSB ではないリソースが存在する場合は、SHARE を指定しないでください。

SHARE オプションを指定してユーティリティーを実行する前に、z/OS システム内で PDSE データ・セットの拡張共用を使用可能にする必要があります。z/OS 内で PDSE データ・セットの拡張共用を使用可能にするには、シスプレックス内の各システム上で、SYS1.PARMLIB 内の IGDSMSxx メンバーに PDSESHARING(EXTENDED) を指定します。SHARE の指定時に拡張 PDSE 共用が使用可能になっていない場合、ユーティリティーは必要な IMS カタログ・データ・セットを排他モードで割り振ります。これが原因で、他の IMS プロセスおよびアプリケーション・プログラムとの競合が発生する場合があります。

拡張 PDSE 共用の使用可能化について詳しくは、z/OS: 複数システム環境における拡張 PDSE 共用の指定を参照してください。

デフォルトでは、ACB の IMS 管理のために割り振られるシステム・データ・セット名は、IMS カタログの最初の区画が定義されるときに RECON データ・セットに指定されたデータ・セット名の接頭部から取られます。このデフォルトをオーバーライドするには、DFSDFxxx PROCLIB メンバーの CATALOG セクションの SYSDSNHLQ= ステートメントまたはカタログ定義出口ルーチン (DFS3CDX0) で HLQ 名の接頭部を指定できます。

## **NODUPLIST**

追加されなかったリソース・インスタンスのリストを出力しません。このパラメーターがデフォルトです。

## **NOISRTLST**

挿入されたリソース・インスタンスのリストを出力しません。

## **IMS Catalog Populate ユーティリティーの統計報告書**

DFS3PU00 ユーティリティーは、実行のつど、IMS カタログでロードまたは更新するレコード・セグメントに関する統計報告書を作成します。DFS3PU00 ユーティリティーを読み取り専用モードで実行すると、現在このユーティリティーへの入力として使用されている ACB ライブラリーから IMS カタログがロードまたは更新された場合、この報告書は見込み統計のみを反映します。

DFS3PU00 ユーティリティーを分析専用モードで実行するには、ユーティリティー JCL にこのユーティリティーの PSB として DFSCP000 を指定します。

報告書の最初のセクションは、このユーティリティーの現時点での実行中に挿入されたセグメントの要約です。IMS カタログに挿入されたセグメント・タイプごとに、この要約には以下の情報が列編成で表示されます。

- セグメント・コード (SC)
- セグメント名。通常これは、DBD または PSB ソース・ファイルのマクロに対応します。
- セグメントが保管されている IMS カタログ内のデータ・セット・グループ (DSG)
- リストされたセグメントの親セグメント
- IMS カタログにロードされた、そのタイプのセグメントの総数
- 親セグメントの下にある、そのタイプのセグメント・インスタンスの平均数

報告書の 2 番目のセクションは、このユーティリティーによって更新された既存の DBD および PSB レコードの数と、タイム・スタンプが ACB ライブラリー内の対応する ACB メンバーのタイム・スタンプと一致したために更新されなかった既存の DBD および PSB レコードの数の両方を示します。報告書のこのセクションは、ACBLIB タイム・スタンプが既にカタログ内にあったため、再度挿入する必要はなかった DBD および PSB セグメントの数も示します。

報告書の残りのセクションは、ストレージの見積もりを示します。

OSAM データ・セットのために、レポートのストレージ・セクションには指定されたサイズのブロックが一定数だけ示されます。間接リスト・データ・セット (ILDS)、1 次索引データ・セット、副次索引データ・セットを含む VSAM KSDS の場合、レポートには一定数の VSAM レコードが示されます。

これらの数は、DFS3PU00 ユーティリティーへの入力として提供する ACB ライブラリーから構築されるカタログ・レコードをロードするために必要な、スペースの大きさを反映した推定値です。IMS カタログ・データ・セットに必要なストレージの量を計算する場合には、拡張の余地があるよう、計算に十分な追加スペースを加えてください。

IMS に IMS カタログ・データ・セットを自動作成させる場合には、DFSDFxxx PROCLIB メンバーの IMS カタログ・セクション内の SPACEALLOC パラメータに、ユーティリティによって提供される推定値のパーセントで追加スペースを指定することができます。このパラメータのデフォルト値は 500% です。

レポートでは、以下の略語が使用されます。

**DSG** データ・セット・グループ

**L** HALDB ILDS データ・セット。示されるレコード数は、IMS カタログが再編成される場合に作成できる間接リスト項目 (ILE) の、可能な数を表しています。

**SC** セグメント・コード。セグメント・タイプをロードするとき、IMS は固有の ID (整数 1 から 255) としてセグメント・コードを割り当てます。IMS は番号を昇順に割り当て、ルート・セグメント・タイプ (番号 1) から始まり、階層順にすべての従属セグメント・タイプにそれを続けていきます。

**SEGS** セグメント

**X** HALDB 区分 1 次索引。

CATALOG DFSCD000

PARTITION DFSCD01

NUMBER OF SEGMENTS INSERTED INTO THE CATALOG

SC	SEGMENT	INSERTED SEGMENTS	DSG	PARENT	AVERAGE SEGS/PARENT
1	HEADER	4228	A		
2	DBD	2530	A	HEADER	0.6
3	CAPXDBD	7	D	DBD	0.0
5	DSET	2599	D	DBD	1.0
7	AREA	139	D	DBD	0.1
9	SEGM	16337	B	DBD	6.5
10	CAPXSEGM	1	D	SEGM	0.0
12	FLD	16426	C	SEGM	1.0
14	MAR	16426	C	FLD	1.0
17	LCHILD	2687	B	SEGM	0.2
20	XDFLD	134	B	LCHILD	0.0
33	PSB	1840	A	HEADER	0.4
35	PCB	9190	B	PSB	5.0
37	SS	75274	B	PCB	8.2
39	SF	1105	B	SS	0.0
41	DBDXREF	8886	D	PSB	4.8

SEGMENT	WITHIN EXISTING HEADER	DUPLICATES NOT INSERTED
DBD	71	0
PSB	72	0

ESTIMATED SPACE REQUIREMENT TO HOLD INSERTED SEGMENTS

DSG	BLKSIZE	BLOCKS
A	4096	596
B	4096	9343
C	4096	8214
D	4096	236

DSG	RECORDS
---	-----

L 8886  
X 4230


SECONDARY INDEX	RECORDS
----- DFSCX000	----- 8886

## 戻りコード


DFS3PU00 ユーティリティは以下のコードを返します。

- 0 すべての操作が正常に終了
- 4 1 つ以上のレコードを IMS カタログにロードできませんでした。警告メッセージが発行されました。
- 8 以上  
重大エラーでプログラムが中止された

関連タスク:

 IMS Catalog Populate ユーティリティ (DFS3PU00) を使用した IMS カタログへのデータの追加 (システム定義)

関連資料:

 IMS PROCLIB データ・セットの DFSDFxxx メンバー (システム定義)  
3 ページの『第 1 章 アプリケーション制御ブロック保守ユーティリティ』



---

## 第 12 章 IMS Catalog Library Builder ユーティリティー (DFS3LU00)

ACB の IMS 管理が使用可能な場合、IMS Catalog Library Builder ユーティリティー (DFS3LU00) を使用して、データベース記述子 (DBD)、プログラム仕様ブロック (PSB)、およびアプリケーション制御ブロック (ACB) を作成し、従来どおりこれらを必要とするツールまたはプロセスで使用できます。ユーティリティーは、IMS カタログを読み取ることによってこれらのリソースを作成します。

IMS Catalog Library Builder ユーティリティーでは、DBD Generation ユーティリティーおよび PSB Generation ユーティリティーへの入力用の DBD および PSB のマクロ・ステートメントを作成することもできます。

### 制約事項

なし。

### 前提条件

ACB の IMS 管理を使用可能にする必要があります。

### 要件

出力データ・セットが存在し、そのデータ・セットにユーティリティーで生成されたメンバーを保持するための十分な空き領域がなければなりません。

### 入力と出力

DFS3LU00 ユーティリティーは IMS カタログを読み取って、IMS カタログに格納されているデータベースおよびプログラム・ビューの定義から、以下のリソースのうちの 1 つ以上を作成します。

- 現在 IMS システムでアクティブになっている ACB のライブラリー。
- 現在 IMS システムでアクティブになっているデータベース用の DBD のライブラリー。
- 現在 IMS システムでアクティブになっているプログラム・ビュー用の PSB のライブラリー。
- 現在 IMS システムでアクティブになっているデータベースの DBD マクロ・ステートメント定義。このマクロ定義は、DBD Generation ユーティリティーへの入力として使用できます。
- 現在 IMS システムでアクティブになっているプログラム・ビューの PSB マクロ・ステートメント定義。このマクロ定義は、PSB Generation ユーティリティーへの入力として使用できます。

ユーティリティーで作成可能なリソースは、SYSIN DD ステートメントで制御ステートメント・パラメーターを指定して選択します。

入力として提供される各制御ステートメントについて、ユーティリティでは、制御ステートメントと同じ名前の DD ステートメントで識別される出力データ・セットが必要です。

## JCL 仕様

IMS Catalog Library Builder ユーティリティでは、DBD マクロ・ステートメント、PSB マクロ・ステートメント、および ACB、DBD、PSB の各ライブラリーを生成できます。このユーティリティでは、JCL に組み込む制御ステートメントおよび DD ステートメントに応じて、上記のさまざまなタイプの出力タイプの 1 つ、一部、またはすべてを 1 回の実行で生成できます。

組み込む DD ステートメントには連結を含めないでください。

JCL では、IMSCATHLQ パラメーターは、ユーティリティが DBD リソースおよび PSB リソースの作成に使用するデータベースおよびプログラム・ビューの定義が含まれている IMS カタログを識別します。IMS カタログの IMS ディレクトリー・データ・セットの高位修飾子または IMS ディレクトリー・データ・セットの高位修飾子を定義する CATDSDLQ DFSMDA メンバーの名前のどちらかを指定することができます。CATDSDLQ DFSMDA メンバーを指定する JCL の例については、CATDSDLQ DFSMDA メンバーを指定するための JCL を参照してください。

### DBD マクロ・ステートメントを作成するための JCL

```
*****
/* THIS STEP FILLS THE DBDSOR LIBRARY
*****
//LUDBDSOR EXEC PGM=DFS3LU00,REGION=0M
/*
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//LUSYSVRT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSABEND DD SYSOUT=*
//DBDSOR DD DSN=IMS.DFS3LU00.DBDSOR,DISP=OLD,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN DD *
DBDSOR
IMSCATHLQ=IMS.DFSCD000
/*
```

### PSB マクロ・ステートメントを作成するための JCL

```
*****
/* THIS STEP FILLS THE PSBSOR LIBRARY
*****
//LUPSBSOR EXEC PGM=DFS3LU00,REGION=0M
/*
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//LUSYSVRT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSABEND DD SYSOUT=*
//PSBSOR DD DSN=IMS.DFS3LU00.PSBSOR,DISP=OLD,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN DD *
PSBSOR
IMSCATHLQ=IMS.DFSCD000
/*
```



### ACB ライブラリーを作成するための JCL

```
//*****  
//* THIS STEP FILLS THE ACBLIB LIBRARY  
//*****  
//LUACBLIB EXEC PGM=DFS3LU00,REGION=0M  
//*  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//SYSPRINT DD SYSOUT=A  
//LUSYSPRT DD SYSOUT=A  
//SYSOUT DD SYSOUT=A  
//SYSABEND DD SYSOUT=*  
//ACBLIB DD DSN=IMS.DFS3LU00.ACBLIB,DISP=OLD  
//SYSIN DD *  
ACBLIB  
IMSCATHLQ=IMS.DFSCD000  
/*
```

### PSB ライブラリーを作成するための JCL

```
//*****  
//* THIS STEP FILLS THE PSBLIB LIBRARY  
//*****  
//LUPSBLIB EXEC PGM=DFS3LU00,REGION=0M  
//*  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//SYSLIB DD DISP=SHR,DSN=IMS.SDFSMAC  
//SYSPRINT DD SYSOUT=A  
//LUSYSPRT DD SYSOUT=A  
//SYSOUT DD SYSOUT=A  
//SYSABEND DD SYSOUT=*  
//PSBLIB DD DSN=IMS.DFS3LU00.PSBLIB,DISP=OLD  
//SYSIN DD *  
PSBLIB  
IMSCATHLQ=IMS.DFSCD000  
/*  
//*****  
//* SYSAIN is used to compile the PSB source.  
//*****  
//SYSAIN DD DSN=IMSTESTS.DFS3LU00.SYSAIN,DISP=OLD  
//*****  
//* SYSLIN is used to link the compiled PSB objects into libraries.  
//*****  
//SYSLIN DD DSN=IMSTESTS.DFS3LU00.SYSLIN,DISP=OLD  
//*
```

### DBD ライブラリーを作成するための JCL

```
//*****  
//* THIS STEP FILLS THE DBDLIB LIBRARY  
//*****  
//LUDBDLIB EXEC PGM=DFS3LU00,REGION=0M  
//*  
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR  
//SYSLIB DD DISP=SHR,DSN=IMS.SDFSMAC  
//SYSPRINT DD SYSOUT=A  
//LUSYSPRT DD SYSOUT=A  
//SYSOUT DD SYSOUT=A  
//SYSABEND DD SYSOUT=*  
//DBDLIB DD DSN=IMS.DFS3LU00.DBDLIB,DISP=OLD  
//SYSIN DD *
```

```

DBDLIB
IMSCATHLQ=IMS.DFSCD000
/*
//*****
//* SYSAIN is used to compile the DBD source.
//*****
//SYSAIN DD DSN=IMSTESTS.DFS3LU00.SYSAIN,DISP=OLD
//*****
//* SYSLIN is used to link the compiled DBD objects into libraries.
//*****
//SYSLIN DD DSN=IMSTESTS.DFS3LU00.SYSLIN,DISP=OLD
//*

```

### CATDSHLQ DFSMDA メンバーを指定するための JCL

次の例では、DFSMDA メンバーを含むデータ・セットが STEPLIB に連結されています。IMS カタログの別名と文字 HLQ が IMSCATHLQ パラメーターに指定されています。この値は、高位修飾子を定義する DFSMDA メンバーの名前と一致します。

この、DFS3LU00 ユーティリティ JCL に CATDSHQ DFSMDA メンバーを指定するための方式は、あらゆるタイプのリソースを作成するのに使用されます。

```

//*****
//* THIS STEP FILLS THE ACBLIB LIBRARY
//*****
//LUACBLIB EXEC PGM=DFS3LU00,REGION=0M
//*
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
// DD DSN=IMS.MDALIB,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//LUSYSVRT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSABEND DD SYSOUT=*
//ACBLIB DD DSN=IMS.DFS3LU00.ACBLIB,DISP=OLD
//SYSIN DD *
ACBLIB
IMSCATHLQ=DFSCHLQ
/*

```

### DD ステートメント

#### ACBLIB DD

ユーティリティが作成するメンバーを格納する出力 IMS ACBLIB データ・セットを定義します。

#### DBDLIB DD

ユーティリティが作成するメンバーを格納する出力 IMS DBDLIB データ・セットを定義します。

#### DBDSOR DD

ユーティリティが作成する DBD ソース・メンバーを格納する出力データ・セットを定義します。

DBDSOR DD ステートメントを使用して、以下の DCB 属性を指定する必要があります。

- 編成 . . . . : PO
- レコード・フォーマット . . . : F

- レコード長 . . . . : 80
- ブロック・サイズ . . . . . : 80
- データ・セット名タイプ : PDS

#### DFSRESLB DD

IMS SVC モジュールが入っている許可ライブラリーを指します。IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) で許可する必要があります。この DD ステートメントには、許可ライブラリー中になければならない IMS SVC モジュール用の許可ライブラリーを指定します。JOB LIB または STEPLIB ステートメントは、IMS バッチには許可は不要です。

#### LUSYSVRT DD

ユーティリティによって書き込まれる状況および要約統計情報を保持する出力データ・セットを定義します。このデータ・セットは、SYSOUT に送信することも、永続データ・セットとして定義することもできます。

以下の例は、処理される各メンバーの LUSYSVRT DD ステートメントに書き込み可能な状況メッセージのタイプを示しています。コンパイラおよびリンカーの出力には追加情報が含まれています。

```

COMPILE SUCCESS FOR MEMBER : mbr_name
LINK    SUCCESS FOR MEMBER : mbr_name
COMPILE FAILURE FOR MEMBER : mbr_name  RC = rc
LINK    FAILURE FOR MEMBER : mbr_name  RC = rc

```

ユーティリティは、実行が完了すると、LUSYSVRT DD ステートメントに要約統計情報も書き込みます。以下の例は、ユーティリティによって書き込まれる統計のタイプを示しています。

```

Processed xxxxxxxx out of yyyyyyyy members
Total catalog entries      : xxxxxxxx
Entry get failures        : xxxxxxxx
Entry get successes       : xxxxxxxx
xxxxxxx compiles attempted : yyyyyyyy
xxxxxxx compiles completed : yyyyyyyy
xxxxxxx compiles failed   : yyyyyyyy
xxxxxxx links attempted   : yyyyyyyy
xxxxxxx links successful  : yyyyyyyy
xxxxxxx links failed     : yyyyyyyy
xxxxxxx members created   : yyyyyyyy

```

#### PROCLIB DD

ユーティリティに必要な、IMS カタログの各種属性を定義する DFSDFxxx メンバーが入った IMS.PROCLIB データ・セットを定義します。

#### PSBLIB DD

ユーティリティが作成するメンバーを格納する出力 IMS PSBLIB データ・セットを定義します。

#### PSBSOR DD

ユーティリティが作成する PSB ソース・メンバーを格納する出力データ・セットを定義します。

PSBSOR DD ステートメントを使用して、以下のDCB 属性を指定する必要があります。

- 編成 . . . . : PO
- レコード・フォーマット . . . : F

- レコード長 . . . : 80
- ブロック・サイズ . . . . . : 80
- データ・セット名タイプ : PDS

#### STEPLIB DD

IMS.SDFSRESL を指します。ここには、IMS 中核と必須の IMS モジュールが入っています。IMS.SDFSRESL に連結された無許可ライブラリーがあるために STEPLIB が無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

#### SYSABEND DD

ダンプ・データ・セットを定義します。

#### SYSAIN DD

DBDLIB 制御ステートメントまたは PSBLIB 制御ステートメントが指定されている場合に、DBDLIB メンバーまたは PSBLIB メンバーを作成するコンパイラへの入力として使用する中間データ・セットを定義します。

LRECL=80,BLKSIZE=80,RECFM=F,DSORG=PS を指定してデータ・セットを定義します。

#### SYSIN DD

入力制御ステートメント・データ・セットを定義します。これらのデータ・セットは、テープ・ボリューム、直接アクセス装置、またはカード読取装置に置くか、入力ストリームで経路指定することができます。入力は、80 の倍数でブロック化できます。実行時に、このユーティリティーは必要な数の制御ステートメントを処理できます。

#### SYSLIB DD

DBDLIB 制御ステートメントまたは PSBLIB 制御ステートメントが指定されている場合に、IMS マクロ・ライブラリーを含む IMS.SDFSREMAC データ・セットを識別します。

#### SYSLIN DD

DBDLIB 制御ステートメントまたは PSBLIB 制御ステートメントを指定する場合は、DBDLIB メンバーまたは PSBLIB メンバーをリンクさせるバインダーへの入力として使用する中間データ・セットを定義してください。

LRECL=80,BLKSIZE=80,RECFM=F,DSORG=PS を指定してデータ・セットを定義します。

#### SYSPRINT DD

出力メッセージ・データ・セットを定義します。

SYSPRINT DD ステートメントで DASD データ・セットまたはテープ・データ・セットを参照する場合、そのデータ・セットのブロック・サイズを DCB パラメーターの BLKSIZE サブパラメーターで制御できます。BLKSIZE 値を指定する場合、その値は正確に 121 の倍数とする必要があります。そうでない場合はシステム ABEND013-20 が発生する可能性があります。DASD データ・セットの BLKSIZE を省略すると、システムが決定するブロック・サイズが使用されます。LRECL パラメーターに指定される値に関係なく、ユーティリティーは常にレコード長 121 を使用します。

## ユーティリティー制御ステートメント

DFS3LU00 ユーティリティーの SYSIN DD ステートメントには、以下の制御パラメーターを自由に組み合わせて指定できます。

### ACBLIB

IMS システム内のアクティブ ACB を ACB ライブラリー・データ・セットにコピーします。結果の ACB ライブラリーは、ACBLIB データ・セットを必要とする IMS システム、アプリケーション・プログラム、ユーティリティー、またはその他のプロセスで使用することができます。

### IMSCATHLQ=

ユーティリティーが ACB、DBD、および PSB の各リソースの取得元として使用できる IMS ディレクトリー・データ・セットの高位修飾子を指定する、必須制御ステートメントです。

IMS ディレクトリー・データ・セットの高位修飾子が CATDShLQ DFSMDA メンバーによって定義されている場合は、IMS カタログの別名に続けて HLQ を指定します。例えば、IMSCATHLQ=DFSCHLQ です。この値は、高位修飾子を定義する CATDShLQ DFSMDA メンバーの DDNAME パラメーターで指定された値とも一致する必要があります。

### DBDLIB

IMS カタログに格納されているアクティブ・データベース定義から、DBD ライブラリー・データ・セットにデータベース記述子 (DBD) メンバーを作成します。

DBDLIB 制御ステートメントを指定する場合は、以下の追加の DD ステートメントが必要です。

- SYSAIN DD
- SYSLIB DD
- SYSLIN DD

### MBR=

ユーティリティー出力の作成対象となる単一リソースの名前を指定します。指定する名前は、IMS カタログ内のアクティブ・リソースの名前と一致している必要があります。

MBR= パラメーターで指定するリソースのタイプは、MBR= パラメーターで指定した制御ステートメントによって定義されている出力タイプと整合性がなければなりません。例えば、DBDLIB 制御ステートメントまたは DBDSOR 制御ステートメントを指定する場合は、MBR= パラメーターで指定するリソース名は DBD にする必要があります。

MBR= パラメーターを指定できるのは、ユーティリティーの 1 回の実行について 1 回のみです。複数回指定されている場合、ユーティリティーは、MBR= パラメーターの最後のインスタンスで指定されたリソースのみを作成します。

### PSBLIB

IMS カタログに格納されているアクティブ・プログラム・ビュー (PSB) 定義から、PSB ライブラリー・データ・セットにプログラム仕様ブロック (PSB) メンバーを作成します。

PSBLIB 制御ステートメントを指定する場合は、以下の DD ステートメントも必要です。

- SYSAIN DD
- SYSLIB DD
- SYSLIN DD

#### **DBDSOR**

IMS カタログに格納されているアクティブ・データベース定義から、DBD Generation ユーティリティ用のマクロ・ステートメントを作成します。

#### **PSBSOR**

IMS カタログに格納されているアクティブ・プログラム・ビュー (PSB) 定義から、PSB Generation ユーティリティ用のマクロ・ステートメントを作成します。

---

## 第 13 章 IMS Catalog Record Purge ユーティリティー (DFS3PU10)

IMS Catalog Record Purge ユーティリティー (DFS3PU10) を使用して、DBD または PSB インスタンスを表すセグメント、DBD バージョンのすべてのインスタンス、あるいは DBD または PSB レコード全体を、IMS カタログから削除します。

PROCLIB データ・セットの DFSDFxxx メンバーに ACBMGMT=CATALOG が指定されている場合、IMS カタログから削除される DBD および PSB は、IMS によって IMS ディレクトリーからも自動的に削除されます。DFSDFxxx メンバーに ACBMGMT=CATALOG が指定されていない場合は、MANAGEDACBS ステートメントを使用して、IMS カタログから削除される DBD および PSB がユーティリティーによって IMS ディレクトリーから削除されるようにしてください。ACB が IMS ディレクトリーから削除されると、その ACB は IMS ディレクトリー・ステージング・データ・セットからも削除されます。

このユーティリティーは、分析、パージ、および更新という 3 つの基本機能を実行します。これらの機能は、ユーティリティーの 1 回の実行の中で、独立して実行することも順次に実行することもできます。分析機能か除去機能、あるいはその両方を選択するには、MODE 制御ステートメントを使用します。IMS カタログ内の DBD および PSB レコードの保存基準を設定または変更するには、UPDATE 制御ステートメントを使用します。

このユーティリティーの分析機能は、ユーザー定義の保存基準を読み取って、削除に適格である DBD および PSB インスタンスを識別することにより、IMS カタログにあるレコードを評価します。DELDBVER ステートメントを指定すると、このユーティリティーは指定された DBD バージョンのすべてのインスタンスを、削除に的確であると識別します。ユーティリティーは、削除可能な DBD または PSB インスタンスのそれぞれについて、報告書および DELETE ステートメントを作成します。

ユーティリティーの除去機能は、保存基準をチェックせずに DELETE ステートメントを処理することにより、DBD および PSB インスタンスを削除します。DELETE ステートメントに対して追加および編集を行って、この追加および編集を行わなければ削除に適格ではない DBD および PSB インスタンスを削除することができます。DELDBVER ステートメントをコード化して、指定された DBD バージョンのすべてのインスタンスを削除することができます。

ユーティリティーの更新機能では、個々の DBD および PSB レコードのヘッダー・セグメントにある保存基準を設定または変更します。ヘッダー・セグメントにある保存基準は、IMS PROCLIB データ・セットの DFSDFxxx メンバーで指定されたデフォルトの保存基準をオーバーライドします。更新機能と分析機能が同時に要求された場合、ユーティリティーは更新を実行してから分析を実行します。

DFS3PU10 ユーティリティーは、200 回更新するたびに CHKP (チェックポイント) 呼び出しを発行して、IMS カタログに対する更新をコミットします。その際に、メッセージ DFS4518I がユーティリティー出力で発行されます。

ユーティリティーのモードおよび機能は、以下の制御ステートメントで指定します。

- MODE ANALYSIS | PURGE | BOTH
- DELDBVER
- UPDATE DBD | PSB
- DELETE
- MANAGEDACBS

DFS3PU10 ユーティリティーは、DL/I 領域、DBB 領域、または BMP 領域で実行できます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 453 ページの『推奨事項』
- 453 ページの『入力と出力』
- 453 ページの『JCL 仕様』
- 455 ページの『SYSIN 制御ステートメント』
- 459 ページの『SYSUT1 制御ステートメント』
- 460 ページの『その他の使用法情報』
- 461 ページの『JCL の例』
- 462 ページの『戻りコード』

## 制約事項

このユーティリティーについて文書化されている制約事項はありません。

## 前提条件

このユーティリティーについて文書化されている前提条件はありません。

## 要件

DFS3PU10 ユーティリティーには、IMS カタログを使用可能にし、IMS カタログにあるレコードのデフォルトの保存基準を定義する DFSDFxxx メンバーを含む IMS.PROCLIB データ・セットへのアクセス権限が必要です。

DFS3PU10 ユーティリティーを BMP 領域で実行し、IMS カタログを共用する場合は、ユーティリティー JCL の EXEC パラメーターに IRLM サポートを指定する必要があります。



## 推奨事項

最初に、ユーティリティを実行して、各 DBD および PSB レコードに対して有効な保存基準に基づいて削除することができる DBD および PSB インスタンスのリストを取得します。次にそのリストを調べて、引き続き IMS アプリケーションに必要な DBD または PSB インスタンスがそこに含まれていないことを確認します。最後に、このユーティリティを実行して不要な DBD または PSB インスタンスを除去します。

## 入力と出力

DFS3PU10 ユーティリティは、以下の入力を受け入れます。

- 分析機能と更新機能は、SYSIN DD ステートメントからの制御ステートメントを読み取ります。
- 除去機能は、SYSUT1 データ・セットからの制御ステートメントを読み取ります。
- 分析機能は、IMS カタログのレコードを読み取ります。
- すべての機能は、IMS.PROCLIB データ・セットの DFSDFxxx メンバーの CATALOG セクションから読み取りを行います。

DFS3PU10 ユーティリティは、以下の出力を生成します。

- 分析機能は、SYSUT1 データ・セットに DELETE ステートメントを書き込みます。
- 分析機能は、削除に適格な DBD および PSB インスタンスのリストを、SYSPRINT データ・セットに書き込みます。
- 除去機能は、削除された DBD および PSB インスタンスのリストを、SYSPRINT データ・セットに書き込みます。
- 更新機能は、IMS カタログにある DBD および PSB レコードのヘッダー・セグメントを更新します。
- 除去機能は、IMS カタログからセグメントまたはレコードを削除します。

## JCL 仕様

### EXEC ステートメント

DFSDF= パラメーターは、IMS.PROCLIB データ・セットの DFSDFxxx メンバーの 3 文字の接尾部を指定します。

```
//BATCH EXEC PGM=DFSRR00,  
// PARM=(DLI,DFS3PU10,DFSCP001,,,,,,,,,Y,N,,,,,,,,,DFSDF=xxx')
```

IRLM サポートを指定するために、以下の例に示すように EXEC ステートメントで PARM パラメーターをコーディングすることができます。

```
PARM=(DLI,DFS3PU10,DFSCP001,,,,,,,,,Y,Y,irld,,,,,,,,,DFSDF=xxx')
```

BMP 領域でユーティリティを実行するには以下のようにします。

```
PARM=(BMP,DFS3PU10,DFSCP001,,,,,,,,,imsid,,,,,,,,)
```

### DD ステートメント

#### **STEPLIB DD**

IMS.SDFSRESL を指します。ここには、IMS 中核と必須の IMS モジュールが入っています。IMS.SDFSRESL に連結された無許可ライブラリーがあるために STEPLIB が無許可である場合には、DFSRESLB DD ステートメントを含める必要があります。

#### **DFSRESLB DD**

IMS SVC モジュールが入っている許可ライブラリーを指します。IMS バッチの場合は、SDFSRESL および、DFSRESLB DD ステートメント上でそれに連結されたデータ・セットは、許可プログラム機能 (APF) で許可する必要があります。この DD ステートメントには、許可ライブラリー中になければならない IMS SVC モジュール用の許可ライブラリーを指定します。JOBLIB または STEPLIB ステートメントは、IMS バッチには許可は不要です。

#### **PROCLIB DD**

IMS カタログにあるレコードのデフォルトの保存基準を定義する DFSDFxxx メンバーを含む、IMS.PROCLIB データ・セットを定義します。

#### **IMS DD**

IMS.PSBLIB および IMS.DBDLIB データ・セットを定義します。

#### **IEFRDER DD**

プライマリー IMS ログ・データ・セットを定義します。

#### **DFSVSAMP**

バッファ・プール・パラメーター・データ・セットを定義します。

#### **SYSPRINT**

MODE ANALYSIS を指定してこのユーティリティーが実行された場合、SYSPRINT データ・セットには、現在有効な保存基準に基づいて削除に適格である DBD および PSB インスタンスの名前とタイム・スタンプのリストが含まれます。MODE PURGE または MODE BOTH を指定してユーティリティーが実行された場合、SYSPRINT データ・セットには、削除された DBD および PSB インスタンスのリストが含まれます。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FBA および LRECL=121 ですが、これらのパラメーターを指定する必要はありません。

#### **SYSIN**

DFS3PU10 ユーティリティーの分析機能、除去機能、および更新機能によって読み取られるユーティリティー制御ステートメントを含む、物理順次データ・セット。SYSIN データ・セットの DCB パラメーターは RECFM=FB および LRECL=80 ですが、これらのパラメーターを指定する必要はありません。

#### **SYSUT1**

ユーティリティーの除去機能によって読み取られる DELETE 制御ステートメントまたは DELDBVER 制御ステートメント、あるいはその両方を含む物理順次データ・セット。ユーティリティーの分析機能は、DELETE ステートメントを SYSUT1 データ・セットに書き込みます。

MODE BOTH を指定した場合を除いて、SYSUT1 データ・セットの内容を編集して、生成された DELETE ステートメントを変更するか、DELETE ステートメントおよび DELDBVER ステートメントに追加を行うことができます。

SYSUT1 データ・セットの DCB パラメーターは RECFM=FB および LRECL=80 ですが、これらのパラメーターを指定する必要はありません。

## SYSIN 制御ステートメント

ユーティリティーでは、SYSIN DD ステートメントを使用して、以下のタイプの制御ステートメントを受け入れます。

- MODE ステートメント
- DELDBVER ステートメント
- UPDATE ステートメント
- MANAGEDACBS ステートメント

### MODE ステートメント

ユーティリティーが分析機能または除去機能を実行するか、あるいはその両方を実行するかどうかを指定します。MODE ステートメントを指定できるのは、1 回のみです。



### ANALYSIS

各レコードに対して有効な保存基準、および入力として含まれる DELDBVER ステートメントに基づいて、ユーティリティーは、削除に適切な DBD および PSB インスタンスに対する DELETE ステートメントを生成します。IMS カタログからは何も削除されません。

各カタログ・レコードの HEADER セグメントで設定されている保存基準を使用して、削除に適切であるかどうかを判別されます。レコードの HEADER セグメントに保存基準が含まれていない場合、このユーティリティーは IMS.PROCLIB データ・セットの DFSDFxxx メンバーの CATALOG セクションに含まれる保存基準を使用します。

DELETE ステートメントは SYSUT1 データ・セットに書き込まれ、既存の内容をすべて上書きします。分析が完了したら、DELETE ステートメントの確認、および必要に応じて編集を行うことができます。

UPDATE ステートメントはこのモードでサブミットできます。

UPDATE ステートメントは、このユーティリティーがカタログの HEADER 情報を調べる前に処理されます。

### PURGE

このユーティリティーは SYSUT1 データ・セット内の DELETE ステートメントを読み取り、一致する DBD および PSB インスタンスをデータベースから除去します。

UPDATE ステートメントはこのモードでは使用できません。

### BOTH

このユーティリティは ANALYSIS モードで実行され、現在の保存基準に照らして不要となったレコード・インスタンスを判別し、その後、PURGE モードで実行され、識別されたレコード・インスタンスを除去します。

UPDATE ステートメントはこのモードでサブミットできます。

UPDATE ステートメントは、このユーティリティが他のアクションを実行する前に処理されます。

#### DELDBVER ステートメント

指定されたバージョンの DBD のすべてのインスタンスを削除します。DELDBVER を指定すると、MODE ANALYSIS または MODE BOTH も指定する必要があります。

MODE ANALYSIS を指定すると、DELDBVER ステートメントは、いかなる保存基準があろうと、指定されたバージョン番号の DBD のすべての DBD インスタンスに対して、SYSUT1 データ・セット内に DELETE ステートメントを生成します。削除を提案された DBD インスタンスは、SYSPRINT データ・セットにもリストされます。

MODE BOTH を指定すると、DELDBVER ステートメントは、指定された DBD バージョンのすべてのインスタンスを削除します。削除される DBD インスタンスは、SYSPRINT データ・セットにリストされます。

MODE PURGE を指定すると、DELDBVER ステートメントを SYSIN データ・セットで指定することはできません。

DELDBVER ステートメントは、SYSUT1 データ・セットでも指定することができます。詳細については、459 ページの『SYSUT1 制御ステートメント』を参照してください。

次の図では、DELDBVER ステートメントの構文を示します。

►—DELDBVER—*member-name*—*version-number*—◄

DELDBVER ステートメントには以下のパラメーターを指定できます。

##### ***member-name***

バージョンを削除する DBD の 8 文字の名前。

##### ***version-number***

削除する DBD のバージョン番号。この値は、削除するバージョンの DBD 生成ステートメントの DBVER キーワードで指定されたバージョン番号と一致していなければなりません。

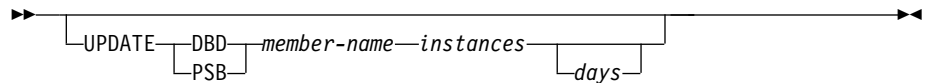
#### UPDATE ステートメント

このステートメントを使用して、カタログ・データベース内のレコードの保存基準を設定します。この情報は、関連するカタログ・レコードの HEADER セグメントに保管されます。

UPDATE ステートメントは任意の数をサブミットできます。1 つ以上の UPDATE ステートメントを指定して MODE ステートメントが指定されると、UPDATE ステートメントが最初に処理されます。

MODE ステートメントを指定せずに 1 つ以上の UPDATE ステートメントを指定すると、ユーティリティーは DELETE ステートメントを SYSUT1 データ・セットに書き込まず、DBD または PSB インスタンスも削除しません。

MODE PURGE を指定すると、UPDATE ステートメントを SYSIN データ・セットで指定することはできません。



### UPDATE DBD|PSB

DBD または PSB を指定します。ユーティリティー入力で、複数の UPDATE ステートメントを指定することができます。

#### *member-name*

IMS リソース名。これらの名前は 8 文字とし、ワイルドカードは以下の形式でサポートされます。

- ワイルドカード演算子 (\*) のみを指定すると、すべての DBD または PSB リソースの保存基準を更新できます。
- 接頭部を指定した後にワイルドカード演算子を指定 (ABC\*) すると、接頭部の値と一致する DBD または PSB レコードの保存基準を更新できます。

更新は英数字順に処理されます。先に行われた更新は後から行われた更新によってオーバーライドされます。例えば、以下の UPDATE ステートメントがサブミットされたとします。

```
UPDATE DBD DB* 1 365
UPDATE DBD * 1 20
UPDATE DBD DBOHIDK5 10 813
```

さらに、IMS カタログ・データベースに以下のリソースのレコードが含まれているとします。

```
DBD CUSTDB
DBD DB1XYZ
DBD DB2XYZ
DBD DB3XYZ
DBD DBOHIDK5
DBD DBVHDJ05
DBD EMHDB1
```

このユーティリティーは、汎用ワイルドカードの更新を最初に処理し、すべての DBD レコードに保存値 VERSIONS=1 および DAYS=20 を割り当てます。次に、DB\* のステートメントを処理し、接頭部 DB を持つすべてのレコードに保存値 VERSIONS=1 および DAYS=365 を割り当てます。最後に、DBOHIDK5 のステートメントを処理し、その特定のリソースに保存値 VERSIONS=10 および DAYS=813 を割り当てます。より新しい更新がより古い更新をオーバーライドします。

#### *instances*

DBD または PSB レコードに保存しておく必要がある DBD または PSB のインスタンスの数。

DBD インスタンスで、データベースのバージョン管理を使用する場合、この値は、各バージョンの DBD で保存する必要がある DBD のインスタンスの数になります。

このパラメーターを 1 に設定すると、最新のインスタンス以外の DBD または PSB インスタンスが削除に適格とされます。

レコード内のインスタンスの数または DBD バージョンがこの値より小さい場合、削除に適格とされるインスタンスはありません。

レコード内のインスタンスの数または DBD バージョンがこの値より大きい場合、最も古いインスタンスが削除に適格とされますが、これはそのインスタンスが *days* の値 (設定されている場合) よりも古いものである場合のみです。

この値は、DBD および PSB レコードにある HEADER セグメントの RETNINST フィールドに保管されます。

### **days**

DBD または PSB のインスタンスを、DBD または PSB レコードから除去可能となるまでの間、保存しておかなければならない日数。この日数より古い DBD および PSB インスタンスのみが、削除に適格とされます。

DBD または PSB レコードのこのパラメーターを省略するか 0 に設定すると、DBD または PSB インスタンスの経過日数は保存基準として使用されません。この値は、HEADER レコード・セグメントの RETNDAYS フィールドに保管されます。

注: *days* パラメーターに 0 より大きい値が指定されると、以下の基準をすべて満たす場合にのみ、DBD または PSB インスタンスは削除に適格とされます。

- インスタンスの経過日数が *days* の値より大きい
- レコード内のインスタンスの数または DBD バージョンが、*instances* の値より大きい
- インスタンスのタイム・スタンプが、*instances* の値を満たすために保存されたインスタンスのタイム・スタンプよりも古い

例えば、レコード内の DBD または PSB インスタンスの数が *instances* の値より小さい場合、1 つ以上のインスタンスの経過日数が *days* の値より大きい場合でも、削除に適格なインスタンスはありません。同様に、レコード内のインスタンスの数が *instances* 値より大きいけれども、レコード内の各インスタンスの経過日数が *days* の値より小さい場合は、削除に適格なインスタンスはありません。

## **MANAGEDACBS**

IMS カタログから DBD インスタンスおよび PSB インスタンスを削除するときに、IMS ディレクトリー内の対応する DBD インスタンスおよび PSB インスタンスを削除するかどうかを指定します。

### **UPDATE**

IMS カタログから削除される DBD インスタンスおよび PSB インスタンスを、IMS ディレクトリーから削除します。IMS ディレクトリーが

アクティブになっている場合、つまり、有効な DFSDFxxx PROCLIB メンバーの <CATALOG> セクションに ACBMGMT=CATALOG が指定されている場合は、MANAGEDACBS UPDATE ステートメントが自動的に生成されます。MANAGEDACBS UPDATE ステートメントと MODE ANALYSIS ステートメントの両方が指定されている場合は、DBD インスタンスおよび PSB インスタンスは IMS ディレクトリーから削除されません。

## SYSUT1 制御ステートメント

SYSUT1 データ・セットの制御ステートメントは、DFS3PU10 ユーティリティーが DBD および PSB セグメント、DBD バージョン、あるいは IMS カタログの DBD または PSB レコード全体を削除するのに使用します。

DFS3PU10 ユーティリティーの分析機能は、SYSUT1 データ・セット内に制御ステートメントを生成します。この制御ステートメントは、除去機能への入力として使用することができます。

MODE ANALYSIS を指定すると、SYSUT1 の生成された制御ステートメントに追加または編集を行ってから、除去機能を実行することができます。ただし、MODE BOTH を指定すると、除去機能が SYSUT1 データ・セットを処理してセグメントとレコードを削除する前に、制御ステートメントを確認または編集することはできません。

MODE PURGE を指定した場合の処理では、SYSUT1 データ・セットをコード化したり、以前のユーティリティーの実行で分析機能が生成した SYSUT1 データ・セットを使用したりすることができます。

以下の制御ステートメントを、SYSUT1 データ・セットで指定することができます。

### DELDDBVER ステートメント

指定されたバージョンの DBD のすべてのインスタンスを削除します。

次の図では、DELDDBVER ステートメントの構文を示します。

▶▶—DELDDBVER—*member-name*—*version-number*—▶▶

DELDDBVER ステートメントには以下のパラメーターを指定できます。

#### ***member-name***

バージョンを削除する DBD の 8 文字の名前。

#### ***version-number***

削除する DBD のバージョン番号。この番号は、削除するバージョンの DBD 生成ステートメントの DBVER キーワードで指定されたバージョン番号と一致していなければなりません。

### DELETE ステートメント

IMS カタログ・データベースから削除する DBD または PSB インスタンス、あるいは DBD または PSB レコード全体を指定します。

各 DBD または PSB レコード内の個々のインスタンスは、DBD または PSB の名前、および ACB 生成タイム・スタンプで区別します。

ユーティリティーの分析機能は、各レコードに対して現在有効な保存基準、および SYSIN データ・セットの DELDBVER ステートメントに基づいて、自動的に DELETE ステートメントを生成します。

MODE ANALYSIS を指定すると、何かを削除する前に、SYSUT1 データの内容を確認および編集することができます。

インスタンスが削除に適格であるかどうかは、レコードの HEADER セグメント、または IMS.PROCLIB データ・セットの DFSDFxxx メンバーの CATALOG セクションからユーティリティーが読み取る保存基準によって決定されます。

除去モード時は、ユーティリティーは保存基準のチェックを行いません。DELETE ステートメントを手作業でコード化または編集すると、このコード化または編集を行わなければ削除に適格ではない DBD および PSB インスタンスを削除することができます。

以下の構文図は、DELETE ステートメントの形式を示しています。

```
▶▶—DELETE—DBD—member-name—timestamp—▶▶
      |
      └─PSB─┘
```

#### **DELETE DBD|PSB**

DBD または PSB のいずれかのリソースのレコード・インスタンスを削除できます。

#### ***member-name***

DBD または PSB リソースの IMS 名。これらの名前は 8 文字とし、ワイルドカードは以下の形式でサポートされます。

- ワイルドカード演算子 (\*) のみを指定すると、すべての DBD または PSB リソースを含めることができます。
- 接頭部を指定した後にワイルドカード演算子を指定 (ABC\*) すると、接頭部の値と一致する DBD または PSB リソースを含めることができます。

#### ***timestamp***

除去する特定の DBD または PSB インスタンスを識別する ACB タイム・スタンプ。

このタイム・スタンプの形式は yydddhhmsssthh です。

ヒント: ワイルドカード演算子 (\*) を指定すると、ルート・セグメントおよびヘッダー・セグメントを含め、DBD レコードまたは PSB レコード全体を IMS カタログから削除することができます。

### **その他の使用法情報**

DBD または PSB レコードの HEADER セグメントの RETNINST フィールドまたは RETNDAYS フィールドに保存基準が保管されていない場合、ユーティリティーは IMS.PROCLIB データ・セットの DFSDFxxx メンバーを使用して、保存基準を決定します。



このレコードの HEADER セグメントに RETNINST フィールドと RETNDAYS フィールドの両方に非ゼロ値が含まれる場合、ユーティリティーは、除去する DBD および PSB インスタンスを決定するのに DFSDFxxx メンバーの値を使用しません。代わりに、HEADER セグメントの RETNINST フィールド (最小インスタンス数) および RETNDAYS フィールド (最小日数) の値を使用します。

**重要:** RETNINST の値がゼロであり、RETNDAYS の値がゼロでない場合、ユーティリティーは、ACB ライブラリーのアクティブ・メンバーに対応するインスタンスを含む、DBD および PSB のすべてのインスタンスをレコードから除去する DELETE ステートメントを生成します。

DFSDFxxx メンバー内の DAYS パラメーターを構成することで、カタログ・レコード・インスタンスを保存する最小日数を設定する一方で、特定のレコードの時間ベースの保存を選択的に無効にすることをお勧めします。その場合、このユーティリティーを使用して、インスタンス数として 1 以上の値を、さらにそれらのレコードの日数として 0 を、それぞれ明示的に指定します。

例えば、DFSDFxxx メンバーに以下の保存情報が含まれている場合、レコード内のインスタンスの数が 5 を超えており、なおかつそれらのインスタンスの少なくとも 1 つの経過日数が 5 日以上になっている場合を除いて、削除に適格とされるインスタンスは存在しません。

```
RETENTION(INSTANCES=5,DAYS=5)
```

ただし、IMS カタログにあるレコードのサブセットに対して時間ベースの保存期間を使用不可にすることができます。これを行うには、DFS3PU10 ユーティリティーの UPDATE ステートメントを使用して、以下の例で示すように各レコードの HEADER セグメントで DAYS 値を直接 0 に設定します。

```
UPDATE DBD JK* 5 0
```

接頭部が JK の各 DBD レコードの場合、上記の例によって RETNINST=5 および RETNDAYS=0 が設定されます。これらのレコードでは、IMS Catalog Record Purge ユーティリティーが削除が適格であるかを決定する際に、経過日数が考慮されません。ユーティリティーが JK\* DBD インスタンスに対して DELETE ステートメントを生成するのは、各 JK\* DBD レコードに保管されているインスタンスが 5 つを超える場合のみです。最も古いレコード・インスタンスが最初に除去されます。

DFSDFxxx メンバーの CATALOG セクションに RETENTION 値が指定されていない場合、デフォルトの INSTANCES=2 および DAYS=0 が使用されます。

## JCL の例

以下のユーティリティー JCL の例では、保存基準を更新し、さらに削除に適格なすべての DBD および PSB インスタンスに対して DELETE ステートメントを生成します。

ユーティリティー実行の UPDATE フェーズにおいて、この JCL の例では、すべての PSB レコードの保存設定、および接頭部 JK\* に一致する DBD レコードの設定に対する保存設定が更新されます。

ANALYSIS フェーズにおいて、この JCL の例では、削除に適格なすべての DBD および PSB インスタンスに対して DELETE ステートメントを生成します。DELDBVER ステートメントは、DBD JKDBA020 のバージョン 2 のすべての DBD インスタンスに対して DELETE ステートメントを生成します。ユーティリティーは、SYSUT1 データ・セットが参照するデータ・セットに DELETE ステートメントを書き込みます。その後、このデータ・セットを編集するか、または MODE PURGE を使用した個別の実行への SYSUT1 入力として直接このデータ・セットを使用することができます。

```
//BATCH EXEC PGM=DFSRRC00,
// PARM=(DLI,DFS3PU10,DFSCP001,,,,,,,,,Y,N,,,,,,,,,'DFSDF=001')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//SYSUT1 DD ... DELETE statements
//IEFRDER DD ... Log data set
//DFSVSAMP DD ... Buffer pool parameters
//SYSPRINT DD SYSOUT=* Analysis or purge report
//SYSIN DD * Control statements
MODE ANALYSIS
UPDATE PSB * 5 365
UPDATE DBD JK* 5 365
DELDBVER JKDBA020 2
```

以下の JCL では、MODE PURGE を使用した実行への SYSUT1 入力として使用する、専用の DELETE および DELDBVER ステートメントをコード化する方法について示しています。

```
//BATCH EXEC PGM=DFSRRC00,
// PARM=(DLI,DFS3PU10,DFSCP001,,,,,,,,,Y,N,,,,,,,,,'DFSDF=001')
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.PROCLIB,DISP=SHR
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//IEFRDER DD ... Log data set
//DFSVSAMP DD ... Buffer pool parameters
//SYSPRINT DD SYSOUT=* Analysis or purge report
//SYSIN DD * SYSIN control statements
MODE PURGE
/*
//SYSUT1 DD * SYSUT1 control statements
DELETE DBD JKDBA600 1306514025679
DELETE DBD * 1300112161055
DELETE PSB * 1300112161055
DELETE PSB PSBJKE05 *
DELDBVER JKDBA020 2
/*
```

## 戻りコード

IMS Catalog Record Purge ユーティリティーは、以下のいずれかの戻りコードを発行して実行を完了します。

- 0 エラーまたは例外条件はありません。
- 4 警告のみ。
- 8 制御ステートメントのエラー。


12 制御ステートメントの組み合わせが無効です (メッセージ DFS4424E または DFS4433E で示されます)。

20 以下のいずれかのメッセージを生成するエラーが発生しました。


- DFS4420E
- DFS4421E
- DFS4422E
- DFS4423E
- DFS4427E
- DFS4485E

24 SYSPRINT DD ステートメントが欠落しているか、または無効です。

関連概念:

 IMS カタログからの DBD インスタンスおよび PSB インスタンスの削除 (データベース管理)

関連資料:

 DFSDFxxx メンバーの CATALOG および CATALOGxxxx セクション (システム定義)



---

## 第 3 部 分析ユーティリティーおよび報告書

IMS システムの分析や報告書を収集したり、書式を整えるためには、分析ユーティリティーを使用してください。

各トピックでは、ユーティリティーがどのように機能するのかを説明し、その使用における要件と制約事項を定義しています。また、例を提供しています。



## 第 14 章 高速機能ログ分析ユーティリティー (DBFULTA0)

高速機能ログ分析ユーティリティーは、IMS システム・ログに記録されたデータに基づいて、高速機能についての統計報告書を準備するときに使用します。

このユーティリティーはオフライン・ユーティリティーであり、3 つのデータ・セットを作成します。そのうちの 1 つには次のような 7 つの定様式報告書が含まれます。

- 例外トランザクションの詳細リスト
- IFP 領域のトランザクション・コード別の例外詳細要約
- IFP 領域のトランザクション・コード別の通過時間総合要約
- すべてのトランザクション・コードおよび PSB についてのリソースの使用量と競合の総合要約
- PST 別の IFP 領域の領域占有率の要約
- VSO アクティビティーの要約
- 分析の総括

これらの報告書は、システムのインストール、チューニング、および障害追及に役立ちます。このユーティリティーは、IMS モニターやログ・トランザクション分析ユーティリティーとは関連がありません。

以下の図に、高速機能トランザクション用に計算される 4 つの間隔を示します。

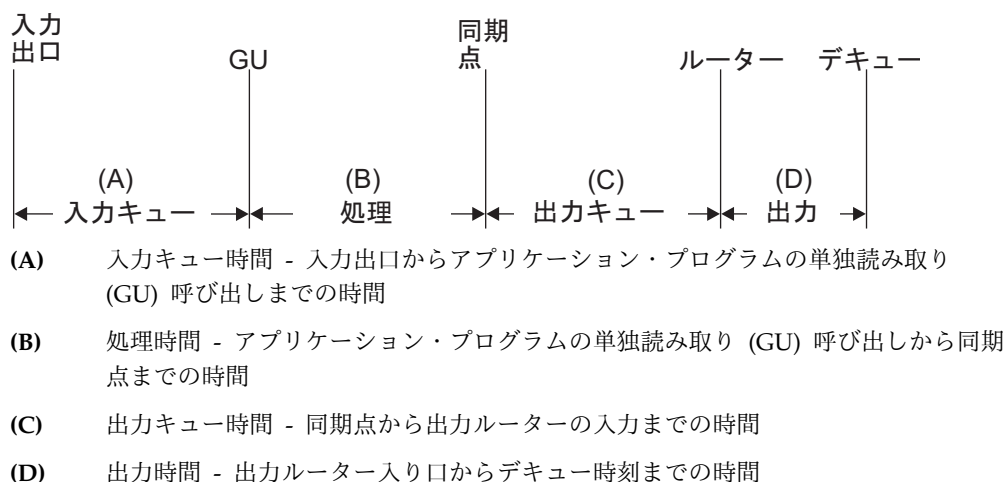


図 36. 高速機能トランザクションの間隔

ログに記録できる最大間隔は、65.535 秒です。ただし、時間幅の計算でフィールドがオーバーフローすると、報告書には計算のオーバーフローを示す 9999 が表示されます。IN-Q、PROC、および OUTQ の各フィールドでは、最大 9.999 秒を表すことができます。

この 4 つの間隔は、計算されて、高速機能ログ・レコードの予約フィールドに挿入され、通常のロギング・プロシーチャーの一部にされます。間隔 (A) と (B) は、それぞれ入力メッセージ (X'5901') ログ・レコードと出力メッセージ (X'5903') ログ・レコードに表示されます。間隔 (C) と (D) は、デキュー・ログ・レコード (X'5936') に表示されます。同期点は、間隔 (B) と (C) の間の境界にとられます。

高速機能ログ分析報告書には、その他に高速機能ログ・レコードからのパフォーマンス関連データ項目が含まれます。ログ・レコードに含まれ、報告もできるデータ項目の種類は、次のとおりです。

- 入力メッセージ (X'5901') ログ・レコード
  - トランザクションの宛先コード
  - トランザクションの入力端末の LTERM 名
  - 負荷平衡グループ・キュー・カウント
- 同期点 (X'5937') ログ・レコード
  - VSO 読み取りの回数
  - VSO 更新 (CI) の回数
  - ADS 読み取りの回数
  - ADS 更新 (CI) の回数
  - DEDB 呼び出しが行われた回数
  - MSDB 呼び出しが行われた回数
  - 制御インターバル (CI) の競合回数
  - 作業単位 (UOW) の競合回数
  - 使用された共通バッファの数
  - 共通バッファ待ちの回数
  - 専用バッファ待ちの回数

間隔 (A)、(B)、(C)、(D)、およびパフォーマンス関連項目を他のログ情報と組み合わせ、すべての報告書が作成されます。

サブセクション:

- 『制約事項』
- 469 ページの『前提条件』
- 469 ページの『要件』
- 469 ページの『推奨事項』
- 469 ページの『入力と出力』
- 470 ページの『JCL 仕様』
- 471 ページの『ユーティリティー制御ステートメント』
- 476 ページの『戻りコード』

## 制約事項

共通キュー・サーバー (CQS) のログ・レコードは、IMS ログ・レコードとは形式が異なるため、高速機能ログ分析ユーティリティーは、CQS ログを入力として使用できません。



## 前提条件

現在、DBFULTA0 ユーティリティーについて文書化されている前提条件はありません。

## 要件

現在、DBFULTA0 ユーティリティーについて文書化されている要件はありません。

## 推奨事項

現在、DBFULTA0 ユーティリティーについて文書化されている推奨事項はありません。

## 入力と出力

高速機能ログ分析ユーティリティーでは、以下の入力を使用します。

- IMS システム・ログ・データ・セット
- 実行パラメーターが指定されている制御ステートメント

高速機能ログ分析ユーティリティーの処理は、以下の 2 ステップから成っています。

1. 高速機能トランザクション詳細レコード (FPTDR) の作成
2. FPTDR の分析と報告書の印刷

このユーティリティーからの出力の基本単位は FPTDR です。処理される高速機能トランザクションごとに 1 つの FPTDR が作成されます。FPTDR は 143 バイトの EBCDIC 論理レコードであり、あるトランザクションに関連するデータ (1 つ以上のログ・レコードからコンパイルされたもの) と、このトランザクションが同期点の処理に入ったときの順序を示すシーケンス番号から成っています。各 FPTDR にデータを提供できる最後のログ・レコードは、トランザクションのデキュー・レコードです。

基本 FPTDR レコードは、例外トラフィック・データ・セットに書き込まれる際に、252 バイトに拡張されます。最初の 143 バイトは、合計トラフィック・データ・セットと同じです。

高速機能ログ分析ユーティリティーは、FPTDR を使用して、以下の 3 つの出力データ・セットを作成します。

- 合計トラフィック・データ・セット。通常はテープ・データ・セットまたは直接アクセス・データ・セットで、すべての FPTDR を含みます。このデータ・セットは、ソートおよび印刷のために、あるいは他の分析のために、後続のジョブ・ステップに渡すことができます。

このデータ・セットはオプションです。

- 例外トラフィック・データ・セット。通常は直接アクセス・データ・セットまたはテープ・データ・セットです。これは、例外として設定した FPTDR のみを含むので、例外トランザクションの詳細リスト報告書に表示されます。このデータ・セットは、ソートおよび印刷のために、あるいは他の分析のために、後続のジョブ・ステップに渡すことができます。

このデータ・セットはオプションです。

- 定様式報告書。通常、トランザクション詳細レコードを様々な組み合わせで作成された複数の報告書から成る印刷出力データ・セットです。

合計転送および例外トラフィック・データ・セットは、ユーティリティーでフォーマット設定された処理パフォーマンス・データを通知しやすくするために用意されたもので、ログ・データ・セットを用いる必要がありません。例えば、報告書を検査すると、回線アクティビティーに関連している可能性のある問題を分析するために、合計トラフィック・データ・セットを物理回線番号および端末順序でソートし、印刷する必要があることが分かる場合があります。DBFULTA0 のソース・コード内の内部 DSECT の FPDR が、これらのレコードをマップします。

レコードは、完了した順序で、合計トラフィック・データ・セットおよび例外トラフィック・データ・セットに書き込まれます。つまり、通常のトランザクション順序のデキュー・レコードの順序で書き込まれます。しかし、各トランザクションに割り当てられるシーケンス番号は、トランザクションが同期点処理に入った順序で決まります。

## JCL 仕様

高速機能ログ分析ユーティリティーは、標準オペレーティング・システム・ジョブとして実行されます。EXEC ステートメント、DD ステートメント、およびユーティリティー制御ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

高速機能ログ分析ユーティリティーを実行します。

```
//EXEC PGM=DBFULTA0
```

### DD ステートメント

#### STEPLIB DD

DBFULTA0 ロード・モジュールが入っているプログラム・ライブラリーを記述します。

```
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
```

#### SYSPRINT DD

DBFULTA0 の印刷出力 (報告書、メッセージ、およびパラメーター・ステートメント・イメージ) を受け取るデータ・セットを記述します。この DD ステートメントは必須です。

```
//SYSPRINT DD SYSOUT=A
```

#### SYSUT1 DD

DBFULTA0 の合計トラフィック出力を受け取るデータ・セットを記述します。これは、DBFULTA0 が形成するすべての高速機能トランザクション詳細レコードから成る順次データ・セットです。各レコードは EBCDIC 文字です。論理レコード長は 143 バイトです。ブロック・サイズ指定はオプションです。

BLKSIZE のデフォルト値は 1430 です。

```
//SYSUT1 DD DSN=&&TOTAL,DISP=(,PASS),UNIT=SYSDA,  
// SPACE=(CYL,(1,1)),DCB=BLKSIZE=2860
```

## SYSUT2 DD

DBFULTA0 の例外トラフィック出力を受け取るデータ・セットを記述します。これは、例外である高速機能トランザクション詳細レコードから成る順次データ・セットです。これは、例外トランザクションの詳細リストのコピーで、見出しと紙送り制御文字が抑制されています。論理レコード長は 252 バイトです。ブロック・サイズ指定はオプションです。BLKSIZE のデフォルト値は 2520 です。

```
//SYSUT2 DD DSN=&&EXCEP,DISP=(,PASS),UNIT=SYSDA,  
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=5040
```

## LOGTAPE DD

入力ログ・データ・セットを記述します。

```
//LOGTAPE DD DSN=IMS33.LOG,DISP=OLD,VOL=SER=XXXXXX,UNIT=XXXX
```

## SYSIN DD

入力制御データ・セットを記述します。このデータ・セットは、実行パラメータを指定するときに使用します。この DD ステートメントはオプションです。以下に、入力ストリームの例を示します。

```
//SYSIN DD *  
START=09:59:59          24-hour notation, note colons  
END=12:00:00  
LINECNT=45             lines per page for reports  
NOT-MESSAGE           include transactions that are not IFPs  
MAXDETAIL=5000        exceptions detail listing limit  
CALLS  
BUFFER  
VSO  
TT(*)=15.0  
TT(TCODE1)=3.0  
TT(TCODE2)=2.5  
TT(TCODE3)=1.0
```

## ユーティリティー制御ステートメント

SYSIN データ・セット内の制御ステートメントは、高速機能ログ分析ユーティリティーを制御します。分析が実行される高速機能実行の時間枠を指定できます。これは、開始時刻または終了時刻で表します。同期点のタイム・スタンプがこの間隔内に入るトランザクションが処理されます。間隔を指定しないと、ログ・データ・セット全体が処理されます。

指定した終了時間までログが処理された後、指定した分析時間枠の間に処理されたトランザクションに関連するデキュー・レコードを探すスキャンが続行されます。

マルチボリューム・ログ・データ・セットの処理は、マルチボリュームを //LOGTAPE DD ステートメントに指定するか、または DD ステートメントを連結することにより行います。

### 通過時間例外の指定

生成される印刷出力の量は、トランザクション・コードごとに例外通過時間値を指定することによって制限できます。この例外値より小さいトランザクション通過時間のオカレンスは、例外トランザクションの詳細リストには現れません。固有トランザクション・コードごとに、別々の例外通過時間を指定できます。さらに、個々

に指定されていないすべてのトランザクション・コードにグローバル値を指定することができます。例外基準を超えるトランザクションについては、別個の要約報告書が作成されます。

例外通過時間を指定しないか (デフォルトは 0)、または後続のジョブ・ステップで合計 FPTDR データ・セットを印刷すれば、ログ・データ・セットから処理されたすべてのトランザクションの詳細報告書を作成することができます。

例外トランザクションの詳細リスト報告書に印刷されるトランザクションの数に、上限を設定することができます。この限界を使用して、予期しない大量の出力リストが生成されるのを防ぐことができます。

分析パラメーター・ステートメントの形式

ステートメントはすべて 1 桁目から始めます。ステートメントは任意の順序で指定でき、検査のために SYSPRINT データ・セットにリストされます。

#### 開始日付の指定 (*STARTDAY*)

処理される最初のトランザクションの日付をユリウス暦形式で指定できます。それより前の日付のトランザクションは無視されます。開始時刻を同時に指定すると、その日付でも同期点時刻がそれより早いトランザクションも無視されます。このパラメーターの形式は次のとおりです。

`STARTDAY=YYDDD`

YYDDD は、年の最後の 2 桁、および 1 から 366 の通算日の順次番号です。

デフォルト値は IMS の開始された日付で、タイプ X'42' のログ・レコードから得られます。

#### 終了日付の指定 (*ENDDAY*)

処理される最後のトランザクションの日付をユリウス暦形式で指定できます。それより後の日付のトランザクションは無視されます。終了時刻を同時に指定すると、その日付でも同期点時刻がそれより後のトランザクションも無視されます。このパラメーターの形式は次のとおりです。

`ENDDAY=yyddd`

(yy は年の最後の 2 桁、ddd は 001 から 365 の通算日の順次番号です)

終了時刻を指定する場合、デフォルトは、タイプ X'42' のログ・レコードからの、IMS が開始された日付です。終了時刻が開始時刻より早い場合は、デフォルトは 1 日後になります。終了日付も終了時刻も指定しなければ、データ・セット全体が処理されます。

#### 開始時刻の指定 (*START*)

処理される最初のトランザクションの時刻を指定できます。同期点時刻がそれより前のトランザクションは無視されます。このパラメーターの形式は次のとおりです。(24 時間クロックでの時間、分、および秒。)

`START=HH:MM:SS[+|-}HH:MM]`

オプションの時間帯情報を指定する必要があるのは、夏時間への変更などのために、入力したその日の協定世界時とのオフセットが現在のオフセットと異なる場合のみです。

hh:mm:ss に続くオプションの時間帯情報には、次のものが入ります。

+ または -

UTC からの時間帯のオフセットの符号を指定します。

**HH** UTC からのオフセットの総時間数を指定します。

**MM** オフセットの分で指定します。MM は 00、15、30、45、またはブランクです。

デフォルトは 00:00:00 です。この場合には、分析はログ・データ・セット内の最初のトランザクションから始まります。

#### 終了時刻の指定 (**END**)

処理する最後のトランザクションの同期点時刻を指定できます。同期点時刻がそれより後のトランザクションは無視されます。このパラメーターの形式は次のとおりです。(24 時間クロックでの時間、分、および秒。)

END=HH:MM:SS[{|-}HH:MM]

オプションの時間帯情報を指定する必要があるのは、夏時間への変更などのために、入力したその日の UTC とのオフセットが現在のオフセットと異なる場合のみです。

hh:mm:ss に続くオプションの時間帯情報には、次のものが入ります。

+ または -

UTC からの時間帯のオフセットの符号を指定します。

**HH** UTC からのオフセットの総時間数を指定します。

**MM** オフセットの分で指定します。MM は 00、15、30、45、またはブランクです。

終了日付を指定しない場合は、デフォルトにより、分析はログ・データ・セットの最後のトランザクションで終了します。

ログ・データ・セットの日付は、パラメーター・ステートメントで明示的に指定されません。このデータは、JCL 要件に記載されているログ・データ・セットに関する指定で暗黙指定されています。ユリウス日付は、実行の開始時にログ・ヘッダー・レコードから読み取られ、検査のためにパラメーター要約の一部として印刷されます。

#### 例外通過時間の指定 (**TT**)

報告の目的のために例外と見なすことにした、各高速機能トランザクションの時間間隔を指定することができます。このパラメーターの形式は次のとおりです。(秒および 10 分の 1 秒)

TT (TRANCODE)=SS.T

トランザクション・コード (最高 8 文字) は、括弧で囲みます。100 個まで個々のトランザクション・コードを指定できます。例外通過時間のグローバル値は、TT(\*)=SS.T (秒および 10 分の 1 秒) と指定します。

この値は、個別に指定されていないすべてのトランザクション・コードに適用されます。個別に指定すると、グローバル値は変更されます。グローバル例外通過時間のデフォルト値は 0 です。例外通過時間の実際的な上限は、65.5 秒です。この制限は、高速機能ログ・レコード内の時間間隔 (A)、(B)、および (C) を表すためのフィールド・サイズによるものです。

非メッセージ・ドリブン・オプション (**NON-MESSAGE** または **NOT-MESSAGE**)

IFP ではない (つまり、BMP、MPP、ユーティリティー、および DBCTL スレッド) トランザクションを例外と見なし、例外トランザクションの詳細リスト報告書に含めるよう指定することができます。受け入れられる形式は次のとおりです。

NON-MESSAGE

または

NOT-MESSAGE

どちらの形式でも同じ結果が得られます。

例外トランザクションの詳細リスト報告書のサイズ制限 (**MAXDETAIL**)

例外トランザクションの詳細リストに印刷される行数を制限することができます。この限界に達した後も、分析は続行されます。しかし、例外トランザクションの詳細リストには、これ以上トランザクションは印刷されません。

このパラメーターの形式は次のとおりです。

MAXDETAIL=n

ここで、n は 7 桁を超えない整数です。デフォルト値は 1000 です。印刷出力行に限界が設定されても、例外詳細トラフィック・データ・セット (SYSUT2) に書き込まれる例外詳細レコードの数には影響しません。

**DL/I** 呼び出し指定 (**CALLS**)

DL/I 呼び出し数の印刷を指定することができます。この呼び出し数は、呼び出しタイプ (GU、REPL など) 別に印刷されます。このパラメーターの形式は次のとおりです。

CALLS

呼び出しについての情報は、タイプ X'5937' のログ・レコードから得られます。

バッファ使用指定 (**BUFFER**)

バッファの使用量をタイプ別に印刷することを指定できます。このパラメーターの形式は次のとおりです。

BUFFER

バッファの使用量について、以下の情報が収集されます。

- 使用した NBA バッファース数 (NBA)
- 使用したオーバーフロー・バッファース数 (OVFN)
- このトランザクションによってバッファース・スチールが呼び出された回数 (STEAL)
- バッファースが使用可能になるのをトランザクションが待った回数 (WAIT)
- OTHREAD へ送られたバッファースの数 (OTHR)
- MSDB および SDEP 処理で使用したバッファースの数 (NRDB)

バッファースについての情報は、タイプ X'5937' のログ・レコードから得られます。

#### 64 ビット高速機能バッファース・プールの指定 (FPBP64)

64 ビット高速機能バッファース・プールの使用量をタイプ別に印刷することを指定できます。このパラメーターの形式は次のとおりです。

FPBP64

プール・サイズが 512 から 28 KB の 64 ビット高速機能バッファース・プールの使用量について、以下の情報が収集されます。

- DL/I 呼び出し処理中にスレッドが取得した共通バッファースの数
- DL/I 呼び出し処理中にスレッドが取得したシステム・バッファースの数
- 高速機能データベースに書き込まれたデータを含む共通バッファースの数
- 高速機能データベースに書き込まれたデータを含むシステム・バッファースの数
- 領域のバッファース待ちの回数。

FPBP64 に関する情報は、タイプ X'5945' のログ・レコードから得られます。

#### データ・スペース使用の指定 (VSO)

データ・スペースの使用に関する情報をトランザクション別に印刷することを指定できます。このパラメーターの形式は次のとおりです。

VSO

データ・スペースの使用量について、以下の情報が収集されます。

- データ・スペースからの CI 読み取り要求が満たされる数 (VGET)
- データ・スペースへの更新の CI の数 (VPUT)。この数は、区域が非 VSO の場合に OTHREAD に送られたはずの CI の数を表します。
- DASD からデータ・スペースに読み取られた CI の数 (DGET)

データ・スペースについての情報は、タイプ X'5937' のログ・レコードから得られます。

#### ページ当たりの印刷行数の指定 (LINECNT)

印刷される報告書のページ当たりの印刷行数を指定できます。このパラメーターの形式は次のとおりです。

LINECNT=n

n は 5 より大きい整数です。指定した値は、表題とヘッダーに適用されるため、許容最小値が 6 になります。デフォルト値は、ページ当たり 55 行です。

各パラメーター・ステートメントは、検査のために読み取られたとおりに、SYSPRINT データ・セット内にリストされます。以下の例は、SYSIN データ・セットから読み取られたパラメーター・ステートメント、および SYSPRINT データ・セット内にリストされる方法の例を示したものです。

SPECIFIED INPUT PARAMETERS:

```
ANALYSIS START TIME: 00:00:00    DATE: 2010187
                        END TIME: 23:59:59
A MAXIMUM OF 1000 EXCEPTIONAL TRANSACTIONS WILL BE LISTED.
RATE CALCULATION ACTIVE: INTERVAL=86399 SECONDS.
TRANSIT TIME EXCEPTION VALUES:
```

TRANSACTION CODE	EXCEPTION VALUE IN SEC. (IN-Q THRU OUT-Q)
*GLOBAL*	0.0

すべてのパラメーター・ステートメントが読み取られると、指定したパラメーター、または指定しなかったパラメーターの代わりに使用されたデフォルト値のいずれかの要約表示を、ユーティリティーが印刷します。START パラメーターと END パラメーターを両方とも指定した場合は、RATE CALCULATION ACTIVE の行が表示され、領域占有率の要約報告書が生成されます。次の例はパラメーターの表示を示しています。日付情報は、ログ・バッファー制御レコード (X'42') から得られます。

```
LOG DATA SET ANALYSIS FOR IMS FAST PATH
PAGE 1
THE FOLLOWING PARAMETER CARDS WERE READ FROM SYSIN:
LINECNT=45
```

## 戻りコード


ユーザー異常終了コードは生成されません。

以下の戻りコードが作成されます。

コード 意味

- 0 分析が正常に終了しました。
- 4 分析が途中で終了し、結果が部分的に作成されています。
- 8 分析を実行できません。
- 12 DD 名 SYSPRINT をオープンできません。

関連資料:

 ログ・レコード (診断)

---

## 高速機能報告書タイプ

高速機能報告書はトランザクションの詳細を提供します。

### 合計トラフィックおよび例外トラフィック・データ・セットの形式

高速機能ログ分析ユーティリティーは、合計転送および例外トラフィック・データ・セットに書き込まれる高速機能トランザクション詳細レコードを収集します。



FPTDR ごとに 1 つの論理レコードが作成されます。データ・セット編成は固定ブロックで、SYSUT1 (合計トラフィック・データ・セット) の場合は LRECL=143、SYSUT2 (例外トラフィック・データ・セット) の場合は LRECL=252 です。BLKSIZE は、//SYSUT1 DD ステートメントおよび //SYSUT2 DD ステートメントに指定できます。ただし、デフォルト・ブロック化因数は 10 です。論理レコードは、各トランザクションに関連するデキュー・レコードの順序で書き込まれます。データ・コードは、すべての文字について EBCDIC です。各論理レコードの形式は、内部 DSECT FPDR によってマップされます。

編集されるフィールドの先行ゼロはすべて抑制されますが、小数点の左に必ず最低 1 つの非ブランク桁があります。

未使用のフィールド (例えば、デキュー情報がないレコードの出力時間フィールド) は、ブランクに設定されます。

同期点日付および IMS リリース・レベルのフィールドは、参考情報として SYSUT1 および SYSUT2 データ・セットに含まれていますが、定様式報告書には表示されません。

オーバーフロー値が入る 10 進整数フィールドは、すべてが 9 の値で示されます。オーバーフローを示すこの方法では、オーバーフローしたフィールドが高い方にソートされます。

## 例外トランザクションの詳細リスト報告書

IFP トランザクションごとに、何を例外通過時間値 (TT 入力パラメーター) と見なすかを、入力パラメーターで定義します。通過時間は、間隔 A、B、および C (467 ページの図 36 で定義) の合計として定義されます。出力時間 D は、これには含まれません。指定した例外値を超える通過時間のトランザクションは、例外トランザクションの詳細リストに含まれ、SYSUT2 データ・セットに書き込むことができます。

以下のトランザクションは報告書に表示されます。

- 通過時間が例外通過時間指定と等しいかそれより大きく、正常に処理された IFP トランザクション。

これには、デキュー・ログ・レコードが検出されなかったトランザクションが含まれます。これらのトランザクションについては、出力キュー時間 (したがって、合計通過時間も) が分からないためフォーマット設定されません。この条件は、報告書の中の TOTAL 欄に NO DEQ という文字で示されます。

- 同期点障害が生じたすべての IFP トランザクション。これには、最初のメッセージ GU の前の無効な作業、およびメッセージ GU で「QC」状況コードを受け取った後か、または「QC」状況コードを受け取らずにトランザクションが IMS に戻った場合に行われた無効な作業が含まれます。
- 「nonmessage」オプションを指定すると、非メッセージ・ドリブン・トランザクションが含まれます。



欄に非空白文字があると同期障害を示し、前の図では、この欄は空白です。次の図は、同期処理に失敗したトランザクションが記載された報告書の例を示しています。

DETAIL LISTING OF EXCEPTION TRANSACTIONS:

PAGE 4

SEQ NO.	TRANCODE OR PSB	SYNC TIME	POINT F	S CODE	ROUTING CODE	LOGICAL TERMINAL	PST ID	QUEUE COUNT	TRANSIT IN-Q	TIMES(MSEC) PROC OUTQ	- -OUT- DEDB ..ADS.. ..VSO.. MSDB BUF CONTENTIONS R P	CALL	RD	UPD	RD	UPD	CALL	USE	CI	UW	OB	BW	T T	
1	PBVD SAGR	16:26:20.55					1					32	16	0	16	16	0	16	0	0	0	0	0	B
	CALLS - GU	0 GN	0	GNP	0 GHU	0 GHN	0 GHNP	0 REPL	0 ISRT	32	DLET	0	FLD	0	POS	0	TOTAL	32						
	BUFFER - NBA=	16 OVFN=	0	STEAL=	0	WAIT=	0	OTHR=	0	NRDB=	0	PBUF=	0	PBWT=	0	ASIO=	0	AIOW=	0					
	VSO - VGET	0 VPUT	16	DGET	16																			
6	BMP255	16:27:08.37	L				1					42	13	0	0	0	10	0	0	0	0	0	0	
	CALLS - GU	0 GN	0	GNP	0 GHU	0 GHN	21 GHNP	0 REPL	0 ISRT	0	DLET	21	FLD	0	POS	0	TOTAL	42						
	BUFFER - NBA=	5 OVFN=	5	STEAL=	4	WAIT=	0	OTHR=	0	NRDB=	0	PBUF=	0	PBWT=	0	ASIO=	0	AIOW=	0					
	VSO - VGET	0 VPUT	0	DGET	0																			
7	BMP255	16:27:08.38	R				1					42	13	0	0	0	10	0	0	0	0	0	0	
	CALLS - GU	0 GN	0	GNP	0 GHU	0 GHN	21 GHNP	0 REPL	0 ISRT	0	DLET	21	FLD	0	POS	0	TOTAL	42						
	BUFFER - NBA=	5 OVFN=	5	STEAL=	4	WAIT=	0	OTHR=	0	NRDB=	0	PBUF=	0	PBWT=	0	ASIO=	0	AIOW=	0					
	VSO - VGET	0 VPUT	0	DGET	0																			

非空白・コード A から U の意味は次のとおりです。

- A MSDB 検査障害
- B MSDB 算術オーバーフロー
- C DEDB 順次従属域がいっぱいです。
- D DEDB 順次従属挿入でバッファのオーバーフローが起きました。
- E DEDB 順次従属バッファのオーバーフローが 3 回起きました。
- F DEDB エリアを使用できません。
- G 動的 MSDB 区域がいっぱいです。
- H 必要な MSDB セグメントが見つかりません。
- I DEDB FLD 呼び出し。CI のロックを獲得できませんでした。
- J DEDB FLD 呼び出し。デッドロックが起きました。
- K DEDB FLD 呼び出し。オーバーフローが起きました。
- L ROLB 呼び出し
- M DEDB FLD 呼び出し。検査が失敗しました。
- N DEDB FLD 呼び出し。CI 内のセグメントが削除されました。
- O リソース不足
- P /ERE 内の未完了条件
- Q RESYNC 打ち切りが要求されました。
- R リソースのデッドロック
- S データ・セット内にスペースがありません。
- U アプリケーション・プログラムが異常終了しました。

同期障害に関連する情報は、タイプ X'5938' のログ・レコードから得られます。

## ROUTING CODE

負荷平衡グループの ID

## LOGICAL TERMINAL

このトランザクションの入力 LTERM 名

## PST-ID

PST 番号

## QUEUE COUNT

このトランザクションが同期点処理に入ったときの負荷平衡グループ (BALG) キュー内のトランザクション数

## TRANSIT TIMES (ミリ秒単位の通過時間)

### IN-Q 時間間隔 A (ミリ秒単位の入力キュー時間)

トランザクションが以下のものである場合は、入力キュー時間には、共用 EMH 入出力通過時間の N/A がマークされます。

1. ローカルのみ
2. グローバルのみ、または DBFULTA0 が IMS バックエンドのログを読み取っているときに他の CPC で処理される最初のローカル・トランザクション

### PROC

時間間隔 B (ミリ秒単位の処理時間)

### OUTQ

時間間隔 C (ミリ秒単位の出力キュー時間)。出力キュー時間に関連した情報は、タイプ X'5936' のログ・レコード (端末出力デキュー・レコード) から得られます。

トランザクションが以下のものである場合は、入力キュー時間には、共用 EMH 入出力通過時間の N/A がマークされます。

1. ローカルのみ
2. グローバルのみ、または DBFULTA0 が IMS バックエンドのログを読み取っているときに他の CPC で処理される最初のローカル・トランザクション

### TOTAL

時間間隔 A、B、C の合計。これは、ユーティリティーについて定義された通過時間です。この合計値は、トランザクション・コードの例外値を超えます。

## OUT TIME

時間間隔 D。秒単位の (デキューまでの) 出力時間

## DEDB CALL

DEDB 呼び出しの合計数

## ADS READS & UPDATES

読み取りおよび更新が行われた CI の数

## VSO READS & UPDATES

データ・スペースから読み取りおよび更新が行われた CI の数

## MSDB CALL

この処理中の MSDB 呼び出しの回数

## BUF USE

共通バッファ・プールから使用されたバッファの合計数。この数には、MSDB と SDEP に使用された非関連バッファも含まれます。

## CONTENTIONS

- CI この処理の CI 待ちの回数
- UW この処理の UOW 待ちの回数
- OB オーバーフロー・バッファ割り振り待ちの回数。この数は、1 より大きくなることはありません。
- BW 共通バッファ待ちの回数
- RT 領域タイプで、次のいずれかです。
  - B BMP
  - I IFP
  - M MPP
  - U ユーティリティー
- PT 処理タイプで、次のいずれかです。
  - G 共用 EMH グローバル・メッセージ処理
  - H HSSP
  - R 再編成

次の行が得られるのは、オプションのユーティリティー制御ステートメントを指定した場合のみです。ただし、その情報は、SYSUT2 データ・セット中の FPTDR レコードの拡張部分では常に使用可能です。

## CALLS 行

CALLS 行は、DEDB 呼び出しのタイプ別の DL/I 呼び出し回数を含みます。CALLS に関連する情報は、タイプ X'5937' のログ・レコードから得られます。

DL/I 呼び出しのタイプには次のものがあります。

### GU CALL

GU 呼び出しの回数

### GN CALL

GN 呼び出しの回数

### GNP CALL

GNP 呼び出しの回数

### GHU CALL

GHU 呼び出しの回数

### GHN CALL

GHN 呼び出しの回数

### GHNP CALL

GHNP 呼び出しの回数

**REPL CALL**  
REPL 呼び出しの回数

**ISRT CALL**  
ISRT 呼び出しの回数

**DLET CALL**  
DLET 呼び出しの回数

**FLD CALL**  
FLD 呼び出しの回数

**POS CALL**  
POS 呼び出しの回数

**TOTAL**  
この処理の間の DL/I 呼び出しの数

#### **BUFFER 行**

BUFFER 行は、タイプ別のバッファ使用量を含みます。BUFFER に関連する情報は、タイプ X'5937' のログ・レコードから得られます。

バッファ使用には次のタイプがあります。

**NBA** この処理中に NBA ラッチ待ちが起きた回数

**OVFN**  
この処理中に使用したオーバーフロー・バッファの数

**STEAL**  
このトランザクションによってバッファ・スチールが呼び出された回数

**WAIT**  
バッファが使用可能になるのをトランザクションが待った回数

**OTHR**  
OTHREAD へ送られたバッファの数

**NRDB**  
MSDB および SDEP 処理で使ったバッファの数

**PBUF** トランザクション (1 つの作業単位) において、HSSP または高速 DEDB 直接再編成ユーティリティーが使用した専用バッファの数

**PBWT**  
トランザクション (1 つの作業単位) において、HSSP または高速 DEDB 直接再編成ユーティリティーが専用バッファを待った回数

**ASIO** トランザクション (1 つの作業単位) において、HSSP または高速 DEDB 直接再編成ユーティリティーが UOW 非同期先読みを行った回数

**AIOW**  
トランザクション (1 つの作業単位) において、HSSP または高速 DEDB 直接再編成ユーティリティーが完了すべき UOW 非同期先読みの回数

この数は、ゼロか 1 のいずれかです。

## VSO 行

VSO 行には、トランザクションによるデータ・スペース使用についての情報が入ります。VSO に関連する情報は、タイプ X'5937' ログ・レコードから得られます。

データ・スペース使用について収集される情報のタイプは次のとおりです。

### VGET

データ・スペースからの CI 読み取り要求が満たされる数

### VPUT

データ・スペースへの更新の CI の数

この数は、区域が非 VSO だった場合に OTHREAD に送られたはずの CI の数を表します。

### DGET

DASD からデータ・スペースに読み取られた CI の数

## SEMHB 行

SEMHB 行には、EMHQ での高速機能入力および出力メッセージの通過時間が入ります。SEMHB に関連する情報は、タイプ X'5936' ログ・レコードから得られます。

データ・スペース使用について収集される情報のタイプは次のとおりです。

### SHARED EMHB

共用 EMH グローバル・メッセージ処理

### IMSG TRANSIT

アプリケーション GU の前に、高速機能入力メッセージが EMHQ で費やした時間。時間の単位はミリ秒です。

### OMSG TRANSIT

アプリケーション GU の前に、高速機能出力メッセージが EMHQ で費やした時間。時間の単位はミリ秒です。

高速機能トランザクション・コードごとに、別々に例外通過時間値を指定できます。他のすべての指定されていないトランザクション・コードに適用されるグローバル値を指定できます。

## IFP 領域のトランザクション・コード別例外詳細要約報告書

例外トランザクションの詳細リストに選択された例外トランザクションについて、要約が作成されます。ただし、対象になるのは例外 IFP トランザクションのみです。他のトランザクション・タイプは、NON-MESSAGE オプションを指定した場合でも含められません。

デキュー・レコードが検出されなかったトランザクションは、この要約には含まれません。

以下の図は、トランザクション・コード別例外詳細要約の報告書の例です。

SUMMARY OF EXCEPTION DETAIL BY TRANSACTION CODE FOR IFP REGIONS												PAGE 6	
TRANS CODE	-NO.OF- -TRANS-	----TOTAL----		TRANSIT TIMES IN MILLI-SECONDS						INPUT MSG		OUTPUT MSG	
		-AVG-	-MAX-	--INPUT Q --	--PROCESS --	--OUTPUT Q--	LENG (CH)	LENG (CH)-	-AVG-	-MAX-	-AVG-	-MAX-	
TPCA	157837	381	889	293	682	40	405	47	325	94	94	100	100





## TRANCODE OR PSB

トランザクション・コードまたは PSB

## NO. OF TRANS

通過時間の値が算出されたトランザクション・コードのオカレンスの数

## DEDB CALLS

DEDB 呼び出しの数

## TOTAL

この処理時の DL/I 呼び出しの合計数

**GET** この処理時の「GET」DL/I 呼び出しの合計数  
(GU、GN、GNP、GHU、GHN、GHNP)

**UPD** この処理時の「UPDATE」DL/I 呼び出しの合計数  
(REPL、ISRT、DLET、FLD)

**AVG** 1 つの処理間隔当たりの平均呼び出し数

**MAX** 1 つの処理間隔当たりの最大呼び出し数

## MSDB CALLS (AVG MAX)

処理間隔ごとの MSDB 呼び出しの平均回数と最大回数

## ADS I/O

エリア・データ・セット入出力

**RDS** エリア・データ・セットについてのこの処理時に行われた  
「READ」DL/I 呼び出しの合計数  
(GU、GN、GNP、GHU、GHN、GHNP)

**UPD** エリア・データ・セットについてこの処理時に行われた  
「UPDATE」DL/I 呼び出しの合計数 (REPL、ISRT、DLET、FLD)

**AVG** 1 つの処理間隔当たりの平均呼び出し数

**MAX** 1 つの処理間隔当たりの最大呼び出し数

## VSO ACT

VSO のアクティビティー量

**RDS** データ・スペースからの CI 読み取り要求が満たされた合計数

**UPD** データ・スペースへの更新の CI の合計数

**AVG** 1 つの処理間隔当たりの平均呼び出し数

**MAX** 1 つの処理間隔当たりの最大呼び出し数

## COMMON BUFFER USAGE

バッファの使用量

**AVG** 1 つの処理間隔当たりの平均呼び出し数

**MAX** 1 つの処理間隔当たりの最大呼び出し数

**WTS** バッファが使用可能になるのをトランザクションが待った総数

**STL** このトランザクションについてバッファ・スチールが呼び出された  
総数

## TOTL SYNC FAIL

同期点処理に失敗したこのトランザクション・コードの累計オカレンス数

## CONTENTIONS

制御インターバルの競合回数

## TOT UOW

このトランザクション・コードで発生した作業単位競合の総数

## TOT OBA

このトランザクション・コードで発生したオーバーフロー・バッファ領域競合の総数

## CI/SEC

このトランザクション・コードでの 1 秒当たりの CI 競合の総数。時間間隔が 1 秒より小であれば、デフォルトの 1 秒になります。

## TRAN RATE/SEC

このトランザクション・コードの平均トランザクション率。時間間隔が 1 秒より小であれば、デフォルトの 1 秒になります。

## LGNR STATS

LGNR 指定に関連する統計

### NO. OF CI COMB

このトランザクション・コードについて、LGNR 指定を超過した総数。この数は、0 または 1 のいずれかです。

### NO. OF CI LOG'D

このトランザクション・コードについて、CI 全体がログに記録された総数。この数は 0 または 1 のいずれかで、1 になるのは「NO.OF CI COMB」も 1 の場合のみです。

## 領域占有率の要約報告書

指定した期間の間の IFP 領域のおおよその領域占有率の要約報告書が作成されます。時間間隔が 1 秒より小であれば、デフォルトの 1 秒になります。この情報は、適切な数の IFP 領域を作業負荷の処理に使用できるかどうかを調べる際に使用できます。

この報告書は、ユーティリティーに START および END パラメーターの両方を指定した場合にのみ生成されます。次の図は、PST 別 IFP 領域の領域占有率 (パーセント) の要約の例を示しています。

SUMMARY OF REGION OCCUPANCY (PERCENT) FOR IFP REGIONS BY PST

PAGE 9

MEASUREMENT INTERVAL= 120 SECONDS.

REGION 1	HAD 70% OCCUPANCY WITH 84.4 SEC OF TOTAL PROCESS TIME DURING 978 TRANSACTIONS. RELATED PSB=TPC
REGION 2	HAD 67% OCCUPANCY WITH 81.1 SEC OF TOTAL PROCESS TIME DURING 922 TRANSACTIONS. RELATED PSB=TPC
REGION 3	HAD 68% OCCUPANCY WITH 82.2 SEC OF TOTAL PROCESS TIME DURING 956 TRANSACTIONS. RELATED PSB=TPC
REGION 4	HAD 67% OCCUPANCY WITH 81.3 SEC OF TOTAL PROCESS TIME DURING 926 TRANSACTIONS. RELATED PSB=TPC
REGION 5	HAD 69% OCCUPANCY WITH 83.4 SEC OF TOTAL PROCESS TIME DURING 972 TRANSACTIONS. RELATED PSB=TPC
REGION 6	HAD 67% OCCUPANCY WITH 80.7 SEC OF TOTAL PROCESS TIME DURING 919 TRANSACTIONS. RELATED PSB=TPC
REGION 7	HAD 70% OCCUPANCY WITH 84.1 SEC OF TOTAL PROCESS TIME DURING 978 TRANSACTIONS. RELATED PSB=TPC
REGION 8	HAD 68% OCCUPANCY WITH 82.5 SEC OF TOTAL PROCESS TIME DURING 942 TRANSACTIONS. RELATED PSB=TPC
REGION 9	HAD 66% OCCUPANCY WITH 80.4 SEC OF TOTAL PROCESS TIME DURING 944 TRANSACTIONS. RELATED PSB=TPC
REGION 10	HAD 70% OCCUPANCY WITH 84.8 SEC OF TOTAL PROCESS TIME DURING 958 TRANSACTIONS. RELATED PSB=TPC

注: PSB 名は、PSB 名を判別するための十分なデータがログにない場合、報告書に「\*\*\*\*\*」と表示される可能性があります。

## VSO アクティビティの要約報告書

区域別の VSO パフォーマンス統計の要約報告書が作成されます。この報告書は、ディスクへの書き込みが行われた場合にのみ生成されます。以下の図はこの報告書の例です。

SUMMARY OF VSO ACTIVITY						PAGE 12
SHR(0/1) AREA	VSO GETS	VSO PUTS	DASD GETS	DASD PUTS	I/O SCHED	
BRANCH01	8092	8095	0	6012	2154	
TELLER01	8200	8198	0	8018	3752	
SHR(2/3) AREA	CF GETS	CF PUTS	READ HIT	READ XI	DASD GETS	DASD PUTS
AREAFR01	1234567	1234567	99%	99%	1234567	1234567
AREA2	1234567	1234567	N/A	N/A	1234567	1234567

VSO アクティビティの要約報告書の欄見出しは次のとおりです。

### VSO GETS

データ・スペースからの CI 読み取り要求が満たされた合計数

### VSO PUTS

データ・スペースへの更新の CI の合計数。この数は、区域が非 VSO であった場合に OTHREAD に送られたはずの CI の合計数です。

### DASD GETS

DASD からデータ・スペースに読み取られた CI の数

### DASD PUTS

データ・スペースから DASD に書き込まれた CI の数

### I/O SCHED

スケジュールされた入出力の合計数

### CF GETS

カップリング・ファシリティによって満たされた CI 読み取り要求の合計数

### CF PUTS

カップリング・ファシリティへの更新の CI の合計数

### READ-HIT

プールの検索のパーセントおよびバッファが検出された回数。これは、ルックアサイド・プールのみ有効です。

### READ-XI

バッファがプールで検出された回数のパーセントおよびバッファが無効だった回数。これは、ルックアサイド・プールのみ有効です。

### DASD GETS

DASD からカップリング・ファシリティに読み取られた CI の数

## DASD PUTS

カップリング・ファシリティから DASD に書き込まれた CI の数

### 分析の総括報告書

以下の図は、分析の総括報告書の例です。

RECAPITULATION OF THE ANALYSIS:

PAGE 13

(1) TOTAL NUMBER OF FAST PATH TRANSACTIONS EXAMINED (SYSUT1).....	157837
(2) NO. OF TRANSACTIONS INCLUDED IN THE EXCEPTION DETAIL DATA SET (SYSUT2)...	157837
BREAKDOWN BY EXCEPTION TYPE:	
(2.1) TRANSIT TIME.....	157837
(2.2) IFP SYNC FAILURE.....	0
(2.3) NO DEQUEUE RECORD.....	0
(2.4) MPP,BMP, DBCTL AND UTILITIES.....	N/A
(INC SYNC FAILURE)	
(3) NO. OF IFP TRANSACTIONS INCLUDED IN THE SUMMARY OF EXCEPTION DETAIL BY TRANSACTION (2.1)+(2.2).....	157837
(4) NO. OF TRANSACTIONS OR PSBS INCLUDED IN THE PROFILE SUMMARY FOR ALL REGIONS (INC SYNC FAILURE) BY PSB OR TRANCODE.....	157837
(5) NO. OF IFP TRANSACTIONS INCLUDED IN THE OVERALL SUMMARY BY TRANSACTION (1)-(2.3).....	157837
(6) NO. OF TIMES COMBINING CONSTANT WAS DOUBLED.....	0
(7) NO. OF TIMES ENTIRE CI LOGGED (LGNR EXCEEDED).....	0

見出しの意味は次のとおりです。

行 (1) 分析期間内に検査されて、ユーティリティによる報告の統計データの基礎として選択されたトランザクションの数。これは、高速機能リソースと関連するトランザクション、つまり、IFP、MPP、または BMP 領域からのトランザクション、または DBCTL トランザクションからのものを含みます。これらは、SYSUT1 DD ステートメントが指定されている場合は、合計トラフィック出力データ・セットに書き込まれたトランザクションでもありません。

行 (2) 検出され、SYSUT2 データ・セットに書き込まれた例外トランザクションの数。以下が含まれます。

- 通過時間が例外通過時間の指定値と同じかそれ以上である IFP トランザクション
- 同期点障害が生じたすべての IFP トランザクション
- デキュー・レコードが検出されなかったすべての IFP トランザクション
- ユーザーが NON-MESSAGE オプションを選択した場合には、すべての非メッセージ・ドリブン高速機能トランザクション。これには、MPP、BMP、ユーティリティ、および DBCTL トランザクションが含まれません。

行 (2.1)

通過時間が例外通過時間の指定値と同じかそれ以上である IFP トランザクションの数。この数は、IFP 領域のトランザクション・コード別例外詳細要約報告書の NO. OF TRANS 欄で報告されているトランザクション数と一致している必要があります。

行 (2.2)

同期点障害が起きた IFP トランザクションの数。この数は、IFP 領域のトランザクション・コード別例外詳細要約の SYNC FAIL 欄で報告されているトランザクションの数と一致している必要があります。

行 (2.3)

分析期間内にデキュー・レコードが検出されなかった IFP トランザクションの数

行 (2.4)

非メッセージ・ドリブン高速機能トランザクションの数。これには、分析期間内に検出された DBCTL スレッドからのトランザクション、および MPP、BMP、およびユーティリティー領域からのトランザクションすべてが含まれます。これは、NON-MESSAGE オプションを選択した場合にのみ報告されます。

NON-MESSAGE オプションを選択していない場合は、N/A (該当せずの意) の文字が印刷されます。

行 (3) IFP 領域のトランザクション・コード別例外詳細要約で報告された IFP トランザクションの数。この数は、正常に処理されたトランザクションおよび同期点障害が起きたトランザクションを含みます。これは、行 (2.1) および (2.2) で報告されている数の合計です。これには、デキュー・レコードが受け取られなかったトランザクションは含まれません。

行 (4) すべてのトランザクション・コードおよび PSB についてのリソースの使用量と競合の総合要約報告書に含まれているトランザクションの数。この数は、行 (1) の数と一致している必要があります。

行 (5) IFP 領域のトランザクション・コード別通過時間総合要約に示されているトランザクションの数。この数は、NO. OF TRANS 欄で報告されているトランザクションの数と一致している必要があります。

行 (6) すべてのトランザクション・コードについて、LGNR 指定を超過した合計回数

行 (7) すべてのトランザクション・コードについて、CI 全体がログに記録された合計回数



---

## 第 15 章 ファイル選択およびフォーマット設定印刷ユーティリティー (DFSERA10)

ファイル選択フォーマット設定印刷ユーティリティー (DFSERA10) は、IMS ログ・データ・セットのデータの検査と表示を支援します。

このユーティリティーは、次のことを行えます。

- ログ・データ・セット全体の印刷またはコピー
- 制御ステートメントの入力に基づいた、複数のログ・データ・セットからの印刷またはコピー
- Operations Manager (OM) ログ・レコードの印刷
- データ・セット内の順次位置に基づいた、ログ・レコードの選択と印刷
- 外部トレース・データ・セットの選択と印刷
- 時刻、日付、または識別の各フィールドの内容などの、レコードそのものに入っているデータに基づいた、ログ・レコードの選択と印刷
- 選択したログ・レコードをモジュールで特殊処理できるようにすること

入力と出力のオプション、選択範囲、および各種のフィールドとレコードの選択基準を定義するときは、一連の制御ステートメントを使用します。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 492 ページの『推奨事項』
- 492 ページの『入力と出力』
- 492 ページの『JCL 仕様』
- 494 ページの『ユーティリティー制御ステートメント』

### 制約事項

現在、DFSERA10 ユーティリティーについて文書化されている制約事項はありません。

### 前提条件

現在、DFSERA10 ユーティリティーについて文書化されている前提条件はありません。

### 要件

現在、DFSERA10 ユーティリティーについて文書化されている要件はありません。

## 推奨事項

現在、DFSERA10 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

すべてのデータ入力は、QSAM を使用して処理され、テープか直接アクセス装置に入れることができます。データ・セット編成は、物理順次でなければなりません。レコード・フォーマットは、固定長または可変長、ブロック化または非ブロック化、あるいは未定義長にすることができます。複数の入出力データ・セットを使用できるとともに、それらを別々の装置タイプに入れることができます。

制御情報を含んでいるデータ・セットのレコード長は、80 でなければなりません。これらのステートメントは、処理される時と同じ形式および順序で、出力印刷データ・セットに複製されます。エラー条件が検出されると、該当するステートメントに続いてエラー・メッセージが出されます。

出力データは、SYSPRINT データ・セットでフォーマット設定および印刷を行ったり、未変更で指定のデータ・セットへコピーする (あるいはその両方を行う) ことができます。

印刷されるデータは、32 バイトのセグメントにフォーマット設定され、16 進形式と EBCDIC 形式の両方で表示されます。このとき、各セグメントの前に 16 進数の相対オフセット値が付きます。

プログラムの制御のフローは、次の 2 段階があります。

- 制御ステートメントの処理。ここでは、レコード・テストと選択パラメーターが作成され、制御ステートメントのエラーが診断されます。
- レコード選択と出力の処理。ここでは、入力データを読み取って分析してから、選択パラメーターと比較して、そのレコードの出力の適用度を判別します。

最初のフェーズでは、パラメーター・ステートメントを読み取って検査し、必要なテストまたはテスト・シリーズを構成して、テスト・グループを作成します。このテスト・グループは、プログラムの次のフェーズに制御が渡されるときに、レコード選択で使用されます。この第 2 フェーズでは、入力データを読み取り、グループ内の各テストの結果によって後処理を決定します。ファイルの終わり条件を検出して、あるいは示されたレコード・カウントを満して、入力データの最後に達すると、プログラム制御はフェーズ 1 に戻り、そこで次のテスト・グループが作成されます。

## JCL 仕様

ファイル選択フォーマット設定印刷ユーティリティは、標準オペレーティング・システム・ジョブとして実行されます。JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

次の形式でなければなりません。



```
// EXEC PGM=DFSERA10
```

あるいは、EXEC ステートメントをカタログ式プロシージャに組み込むことができます。

#### **DD ステートメント**

##### **STEPLIB DD**

出口ルーチン・モジュールを入れる区分データ・セットを定義します。出口ルーチンを使用しない場合には、またはそのモジュールが LINKLIB に入れられている場合、このステートメントは不要です。

##### **SYSPRINT DD**

フォーマット設定済み印刷レコードおよび制御メッセージを入れるための出力データ・セットを記述します。これは、通常、SYSOUT=A と定義します。

このデータ・セットには、RECFM=FBA の DCB パラメーターが指定されます。LRECL およびブロック・サイズは、SYSPRINT DD ステートメントで指定でき、これは 133 の倍数でなければなりません。両方のデフォルトは 133 です。

**重要:** デフォルト以外の LRECL および EXITR オプション・ステートメントを指定する場合は、デフォルト以外の LRECL をサポートする EXITR モジュールを指定する必要があります。EXITR モジュールがデフォルト以外の LRECL をサポートしていない場合は、デフォルト値の 133 を使用する必要があります。

##### **SYSIN DD**

入力制御データ・セットを記述します。このファイルには、80 文字の固定長レコードが入っていないければなりません。

##### **入力またはデータの DD**

フォーマット設定済み印刷レコードの作成のために検査する入力データ・セットを定義します。

これらのデータ・セットは、直接アクセス装置またはテープ上の、標準ラベル付きファイルでなければなりません。これらは、DSORG=PS である限り、レコード・フォーマットは任意です (F、FB、V、VB、VBS、または U)。

RECFM=U のファイルを使用する場合は、DCB BLKSIZE パラメーターを指定する必要があります。これらのファイルは、QSAM を使用して処理されます。QSAM がサポートするファイルはいずれも、入力として記述できます。

CONTROL ステートメントの中で DD 名が指定されていない場合、使用されるデフォルト DD 名は SYSUT1 です。

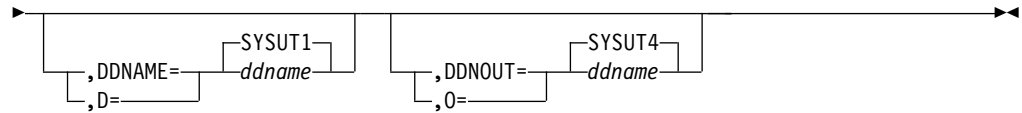
##### **出力またはデータの DD**

選択したレコードを入れるオプションの出力データ・セットを定義します。

DFSERA10 は、このデータ・セットの RECFM を、入力データ・セットの場合に指定された RECFM と等しい値に設定します。LRECL および BLKSIZE についても同様です (指定されていない場合)。

使用されるデフォルト DD 名は、SYSUT4 です。





**CONTROL または CNTL**

CONTROL ステートメントであることを表します。

**SKIP= または K=**

テストする最初のレコードを定義します。これより前のレコードはすべて無視されます。

このキーワードを指定しない場合、デフォルトのゼロが使用され、入力ファイルの最初のレコードがテストされます。

**aaa**

0 から 99999999 の範囲の値を指定する必要があり、コンマを入れてはなりません。

**STOPAFT= または H=**

テストする最後のレコードを定義します。現行のテスト・グループは、処理されるレコードをカウントして、この値に達すると終了します。

このキーワードを指定しない場合、デフォルトの 16777215 が使用されます。

STOPAFT パラメーターのデフォルト値として 16777215 が使用され、しかも EOF を示すメッセージ DFS707I が表示されない場合、16777215 よりあとのレコードは処理されていません。

**bbb**

1 から 99999999 の範囲の値を指定する必要があり、コンマを入れてはなりません。ゼロの値を指定すると、1 つのレコードが処理されます。

**EOF**

ファイルの終わり条件を表します。EOF パラメーターを使用すると、示された最大 99,999,999 個のレコードを超える数のレコードを処理できます。

**E** これを指定すると、レコードが選択基準を満たしている場合にのみ、レコードがカウントされて、一連のテストの終了が判定されます。指定しなければ、(SKIP 値の後) すべてのレコードがカウントされます。

**DDNAME= または D=**

現行のテスト・グループについての入力データ・セットを識別します。対応する DD ステートメントを用意する必要があります。

このキーワードを省略した場合は、デフォルトの SYSUT1 が使用されるため、該当する DD ステートメントを用意しなければなりません。

**DDNOUT= または O=**

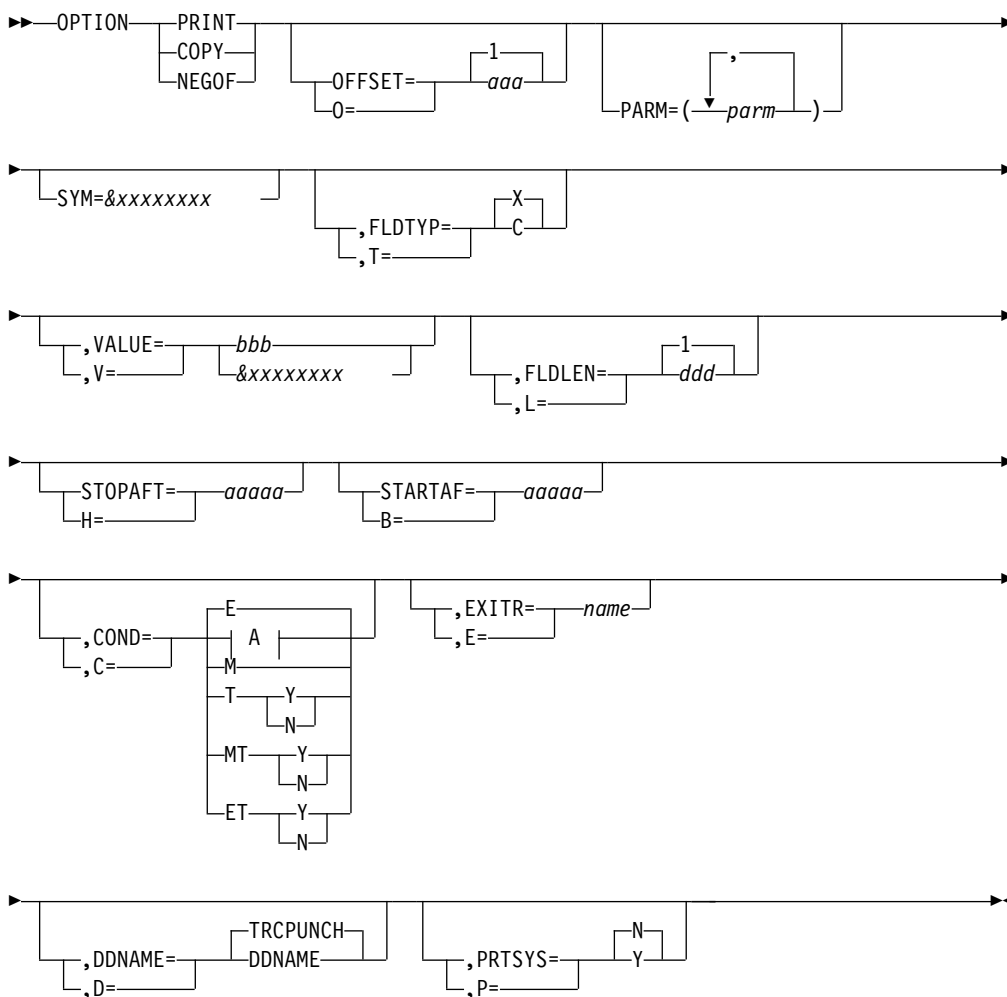
現行のテスト・グループについてのオプションの出力データ・セットを示します。

このキーワードは OPTION COPY 機能と一緒に使用します。これが必要になるのは、デフォルトの SYSUT4 以外の DD 名が必要な場合のみです。

(DDNOUT を指定しても、SYSUT4 が存在しても、このデータ・セットは使用されません。このデータ・セットは、OPTION COPY が COND=E と同時に指定されている場合にのみ使用されます。)

## OPTION ステートメント

OPTION ステートメントは、1 組のテストを作成します。1 つ以上の OPTION 機能を、必要な任意の組み合わせで指定して、各入力レコードに対して実行される出力処理および選択基準をさらに定義できます。EXITR および DDNAME キーワード・オペランド以外のキーワード・オペランドを省略すると、このプログラムのフェーズ 2 で処理されるすべてのレコードは SYSPRINT データ・セットに表示されるか、あるいは指定した出力データ・セットに転送されます。



**A:**



## オプション

各オプションには、次のような 2 つの異なる機能があります。

1. OFFSET キーワードのための開始位置を決定する
2. 実行する出力処理を決定する

個々のオプションを組み合わせて複数フィールド・テストを形成する場合は、OFFSET は変更されずにそのまま使用されます。ただし、出力処理は、COND=E キーワードが指定されている OPTION ステートメントによって決定されます。

### PRINT

選択されたすべてのレコードを SYSPRINT データ・セットに表示します。

### COPY

選択されたすべてのレコードを、指定した出力データ・セットに転送します。これらのレコードは、PRTSYS キーワードを使用すれば、SYSPRINT データ・セットに表示することもできます。

### NEGOF

OFFSET キーワード値を、ログ・レコードの最後からの負のオフセットとして使用します。この機能を使用して選択したすべてのレコードは、SYSPRINT データ・セットに表示されます。

## キーワード

以下のキーワードはすべてオプションです。

### OFFSET= または 0=

レコードの中のテストするフィールドの最初のバイトの位置を定義するのに使用します。デフォルトは、レコードの位置 1 です。

#### **aaa**

1 から、テストするレコードの長さまでの範囲の値を指定する必要があります。最大値は 32767 バイトです。論理レコード長を超えているかどうかを判別するための検査は実行されません。

DSECT を使用して制御レコードまたはブロック内の値を見つける場合は、OFFSET パラメーターの開始値を調整する必要があります。ほとんどの DSECT は相対値ゼロで開始しますが、OFFSET キーワードに指定する値は必ずバイト 1 からの相対位置で表されます。

### PARAM=

パラメーターを DFSERA70 出口ルーチンに渡すときに使用します。

### SYM=

値を記号で定義するときに使用します。このオプションは、VALUE キーワードに代わるもので、VALUE= キーワードと同じエレメント・テストで使用してはなりません。

#### **&xxxxxxxx**

記号の固有名です。「&」は認識文字です。「xxxxxxxx」は、1 文字から 8 文字の記号名です。これは、指定する SYM= ごとに固有でなければなりません。この記号は、テスト・シリーズの中の 1 つ以上の後続エレメントの中で、VALUE= オプションに使用できます。

**FLDTYP= または T=**

VALUE= フィールドのデータ・タイプを定義するのに使用します。

**X** データを 16 進文字のペアして処理することを定義します。テスト・データはパックされます (2 バイトが 1 つにされて、同等の 16 進数にされます)。これは、デフォルトです。

例 : VALUE=D9D6D6E3E2C5C7 (14 バイト) を FLDTYP=X パラメーターとともに指定すると、その結果の VALUE= の値は、EBCDIC では ROOTSET、16 進数では D9D6D6E3E2C5C7 になります。いずれの場合も、長さは 7 バイトです。

**C** データを EBCDIC として処理することを定義します。テスト・データは、カードに穿孔されて使用され、変更はされません。

**VALUE= または V=**

テスト・フィールドの文字を定義します。FLDTYPE=X を指定した場合には、このデータを、16 進文字のペアとして入力する必要があります。「マスクによるテスト」条件の場合は、1 つのペアがテスト用の 16 進数値を表す必要があります。FLDTYP=C を指定した場合には、このデータを、EBCDIC 文字として入力する必要があります。ブランクまたはコンマの文字をこのパラメーターに含める場合は、FLDTYP=X と同等の適切な 16 進数を使用する必要があります。

制約事項: このオプションを SYM= キーワードと同じエレメント・テストで使用してはなりません。

**bbb**

255 個の EBCDIC 文字または 510 個の 16 進文字を超えることはできません。このフィールドの長さは、FLDLEN= キーワード値によって決まり、このフィールド中の「非 nul」文字の数で決まるわけではありません。

**&xxxxxxxx**

前に指定された SYM= オプションの記号名です。各記号は、SYM= オプションで決まる関連した値を持ちます。

**FLDLEN= または L=**

テスト・フィールドから使用する文字数を定義します。

**ddd**

VALUE= キーワードに指定された文字数ではなく、使用する実際のバイト数を表します。このフィールドの許容値の範囲は、1 から 255 です。デフォルトは 1 です。

**STOPAFT= または H=**

単一テストまたは複数フィールド・テストの場合に、選択するレコードの数を定義します。このステートメントは、エレメント・テストごとに COND=E 制御ステートメントに指定できるのみです。

**aaaaa**

0 から 32767 のエレメントにします。

**STARTAF= または B=**

単一テストまたは複数フィールド・テストの場合の、スキップする選択レコードの数を定義します。このステートメントは、エレメント・テストごとに COND=E 制御ステートメントに指定できるのみです。

aaaaa

0 から 32767 のエレメントにします。

**COND= または C=**

テストのタイプ、およびグループ内の他のテストとの関連を定義します。このキーワードを指定しない場合、デフォルトは COND=E です。

- E** テスト・シリーズの最後の (または唯一の) エレメントを示します。この後に現れる OPTION 制御ステートメントは、新しいテストのシリーズになります。これにより、各レコードに対して種々のテストを実行でき、各テスト・シリーズをレコード内の別々のフィールドに使用できます。最終出力処理は、このキーワード値で定義された OPTION 機能によって決まります。
- I** VALUE= 値をテストします。テストが失敗すると、レコードは渡されません。このオプションは独立型で使用するか、E、M、または T パラメーターの前で使用することができます。
- M** これが複数フィールド・テストであることを示します。このシリーズ内のすべてのテストを満たしてからでなければ、このレコードの最終出力選択および処理を開始することはできません。
- T** これを指定すると、VALUE= バイトが、比較フィールドではなく、「マスクによるテスト」値として使用されます。VALUE= フィールドの最初のバイト (FLDTYP=X の場合は 2 つの 16 進文字) のみが使用されます。FLDTYP=C を使用すると、EBCDIC 文字に相当する 16 進文字がテスト値です。このパラメーターを使用する場合、FLDLN= キーワードを指定してはなりません。デフォルト長さ 1 が想定されます。
- Y** 「マスクによるテスト」の条件が満たされるためには、テスト・バイトの対応の各ビットごとに、レコード・テスト・フィールド内に 1 ビットがなければならないことを示します。これは、「1 の場合分岐」テストに相当します。
- N** 「マスクによるテスト」の条件が満たされるためには、テスト・バイトのいずれの対応ビットについても、レコード・テスト・フィールド内に 1 ビットがあってはならないことを示します。これは、「ゼロの場合分岐」テストに相当します。
- MT** 「マスクによるテスト」の OPTION を定義しますが、M パラメーターで説明されているような複数フィールド・テストのプロパティを備えています。T パラメーターではデフォルト「1,E」を想定しているため、「マスクによるテスト」値で始まる複数フィールド・テストについては、MT パラメーターを使用する必要があります。
- ET** 複数フィールドのテスト・シリーズが、「マスクによるテスト」条件で終了することを表します。

**EXITR= または E=**

候補レコードが現行テストのすべての選択基準を満たしたときに、制御を受け取る出口ルーチンのエントリー・ポイント名を指定します。

複数のテスト・グループが同じ出口ルーチンを指定した場合、グループごとにそのルーチンをストレージにロードしようとしています。したがって、そのルーチンは再入可能でなければなりません。入力でファイルの最後に達したときに、出口ル

ルーチンの最終呼び出しが行われます。パラメーター・フィールドにゼロがあるかどうかを検査して、ファイルの終わりに達したかどうかを判別できます。

出口ルーチンへのインターフェースは次のとおりです。

• **ENTRY:**

**REGISTERS**

**R1** パラメーター・リストへのポインターが入っています。

**R13** 空の保管域を指します。

**R14** リターン・アドレスが入ります。

**R15** 出口ルーチンの入り口アドレスが入ります。

**PARMLIST**

- パラメーター・リストは 2 ワードです。最初のワードは候補レコードへのポインターで、2 番目のワード (高位ビットがオン) は、SYSPRINT データ・セットの DCB へのポインターです。

**EXIT:**

出口ルーチンから戻ったときに、このレコードの処理を続行するかどうかを、レジスター 15 の内容で判別します。

ゼロ以外の値であれば、このレコードをこれ以上処理せず、次の入力レコードに対して選択テストを再び開始します。

ゼロの値はこのレコードが必要であることを示します。COND=E キーワードが入っている最後の OPTION ステートメントに基づいて、出力処理が決まります。

EXITR キーワードを省略した場合は、戻りコードのゼロを受け取ったときと同様に処理が続行されます。

**DDNAME= または D=**

DL/I 呼び出しトレース・ログ・レコード検索ルーチン (DFSERA50) が使用する出力データ・セットを定義します (このルーチンがユーザー出口ルーチンとして指定されている場合)。対応する DD ステートメントを用意する必要があります。

このキーワードを省略しても、DFSERA50 が出口ルーチンであれば、デフォルトの TRCPUNCH が使用されるため、適切な DD ステートメントを指定しなければなりません。

**PRTSYS= または P=**

選択されたレコードを SYSPRINT データ・セットに表示するときに使用します。

**N** 選択されたレコードを印刷しないことを指示します。

**Y** 出力データ・セットに転送されたすべてのレコードを、フォーマット設定して印刷することを指示します。

このキーワードが使用できるのは、OPTION COPY 機能と一緒にの場合のみです。N はデフォルトです。

**END** ステートメント



現行の入力ファイル用のすべてのテストの定義を完了したら、END ステートメントを用いてこれらのテストを実行します。

END は 1 桁目に入力します。10 桁目以降はコメント用に使用できます。

### COMMENT ステートメント

コメント・ステートメントはオプションです。これを使用すると、内容が SYSPRINT データ・セットに表示されます。

代替の方法として、1 桁目にアスタリスク (\*) を使用してコメントを示すことができます。ただし、アスタリスクを使用して挿入したコメントは、SYSPRINT データ・セット内に表示されません。

関連資料:

 リポジトリ・サーバーの監査ログ・レコードの印刷 (診断)

---

## DFSERA10 ユーティリティーの例

以下の例は、DFSERA10 の使用法をいくつか示しています。ほとんどの例が IMS ログ・データ・セットを参照していますが、このユーティリティーは、QSAM を使用して処理できる任意のデータ・セットに使用できます。

理解しやすいように、オプション・キーワードはすべて完全な形で指定されています。また、デフォルトを使用できる箇所でも多くがコーディングされています。省略語およびキーワードのデフォルトを使用すると、必要な入力的大幅に少なくなります。

### 例 1

この例では、IMS ログ・データ・セットからすべてのログ・レコードを印刷またはコピーするときに必要な JCL および制御ステートメントを示します。

```
//EXAMPLE1 JOB
//*
//          EXEC PGM=DFSERA10
//STEPLIB  DD DISP=SHR,DSN=IMS.SDFSRESL
//*
//SYSUT1   DD DISP=(OLD,KEEP),DSN=IMSLOG,
//          UNIT=TAPE,VOL=SER=123456
//SYSUT4   DD DISP=(NEW,PASS),DSN=EXAMPLE1.COPY1,
//          UNIT=SYSDA,SPACE=(CYL,(3,1)),
//          VOL=SER=IMSPAC
//*
//SYSPRINT DD SYSOUT=A
//*
//SYSIN    DD *
*-----*
*   CONTROL STATEMENT : DEFAULTS   *
*           INPUT = SYSUT1         *
*           OUTPUT = SYSPRINT      *
* SELECTION QUALIFIERS :          *
*   1. DEFAULT = ALL INPUT RECORDS *
*-----*
OPTION    PRINT
END
*-----*
*   CONTROL STATEMENT : DEFAULTS   *
```

```

*           INPUT = SYSUT1           *
*           OUTPUT = SYSUT4         *
* SELECTION QUALIFIERS :           *
*           1. DEFAULT = ALL INPUT RECORDS *
*-----*
OPTION COPY
END
/*
//

```

## 例 2

この例では、特定のタイプのすべてのログ・レコードを選択して印刷する 2 つの方法を示します。

- 1 つの選択限定子を指定: TYPE X'16' IN 5TH BYTE = (ALL /SIGN ON/OFF)
- 2 つの選択限定子を指定: TYPE X'16' IN 5TH BYTE = (LOG RECORD TYPE) および FLAG X'01' IN 6TH BYTE = 1 (/SIGN ON - ONLY)

```

//EXAMPLE2 JOB
/*
//          EXEC PGM=DFSERA10
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
/*
//SYSPRINT DD SYSOUT=A
/*
//LOGIN    DD DISP=(OLD,KEEP),DSN=IMSLOG,
//          UNIT=TAPE,VOL=SER=123456
/*
//SYSIN    DD *
*-----*
* CONTROL STATEMENT : SPECIFIED *
*           INPUT = LOGIN       *
*           OUTPUT = SYSPRINT   *
* SELECTION QUALIFIERS :       *
*           1. TYPE X'16' IN 5TH BYTE = (ALL /SIGN ON/OFF) *
*-----*
CONTROL CNTL DDN=LOGIN
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=16,FLDLLEN=1,COND=E
END
*-----*
* CONTROL STATEMENT : SPECIFIED *
*           INPUT = LOGIN       *
*           OUTPUT = SYSPRINT   *
* SELECTION QUALIFIERS :       *
*           1. TYPE X'16' IN 5TH BYTE = (LOG RECORD TYPE) *
*           2. FLAG X'01' IN 6TH BYTE = 1 (/SIGN ON - ONLY) *
*-----*
CONTROL CNTL DDN=LOGIN
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=16,FLDLLEN=1,COND=M
OPTION PRINT OFFSET=6,FLDTYP=X,VALUE=01,COND=ETY
END
/*
//

```

## 例 3

この例は、2 種類のログ・レコードを印刷またはコピーする方法を示しています。この両方のログ・レコードに共通するフィールド値 (USERID) がそれぞれに入っていますが、レコード・タイプによってそのオフセット位置が違います。

```

//EXAMPLE3 JOB
/*
//          EXEC PGM=DFSERA10

```

```

//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//*
//SYSPRINT DD SYSOUT=A
//*
//LOGIN DD DISP=(OLD,KEEP),DSN=IMSLOG,
// UNIT=TAPE,VOL=SER=123456
//LOGOUT DD DISP=(NEW,PASS),DSN=EXAMPLE3.COPY1,
// SPACE(CYL,(3,1)),UNIT=SYSDA,
// VOL=SER=IMSPAC
//*
//SYSIN DD *
-----*
* CONTROL STATEMENT : SPECIFIED *
* INPUT = LOGIN *
* OUTPUT = SYSPRINT *
* SELECTION QUALIFIERS : *
* 1. LOG RECORD TYPE X'16' *
* USERID IN 9TH BYTE (FROM BEGINNING OF RECORD) *
* 2. LOG RECORD TYPE X'50' *
* USERID IN 12TH BYTE (FROM END OF RECORD) *
-----*
CONTROL CNTL DDNAME=LOGIN
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=16,FLDLEN=1,COND=M
OPTION PRINT OFFSET=9,FLDTYP=C,VALUE=USERAAAA,FLDLEN=8,COND=E
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=50,FLDLEN=1,COND=M
OPTION NEGOF OFFSET=12,FLDTYP=C,VALUE=USERAAAA,FLDLEN=8,COND=E
END
-----*
* CONTROL STATEMENT : SPECIFIED *
* INPUT = LOGIN *
* OUTPUT = LOGOUT *
* SELECTION QUALIFIERS : *
* * THE SAME AS FOR THE 'PRINT' AND 'NEGOF' OPTIONS *
* ABOVE, BUT SINCE THE 'COPY' OPTION DEFINES AN OUTPUT*
* DATA SET OTHER THAN SYSPRINT, THIS OPTION MUST BE *
* CODED WITH THE 'COND=E' KEYWORD. *
-----*
CONTROL CNTL DDN=LOGIN,DDNOUT=LOGOUT
OPTION PRINT OFFSET=9,FLDTYP=C,VALUE=USERAAAA,FLDLEN=8,COND=M
OPTION COPY OFFSET=5,FLDTYP=X,VALUE=16,FLDLEN=1,COND=E
OPTION NEGOF OFFSET=12,FLDTYP=C,VALUE=USERAAAA,FLDLEN=8,COND=M
OPTION COPY OFFSET=5,FLDTYP=X,VALUE=50,FLDLEN=1,COND=E
END
/*
//

```

#### 例 4

この例では、指定したすべてのログ・レコード・タイプを選択します。それぞれには共通のユーザー ID 値が入っています。これらのレコードを印刷し、指定した出力データ・セットに転送します。

```

//EXAMPLE4 JOB
//*
// EXEC PGM=DFSERA10
//STEPLIB DD DSNAME=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//*
//LOGIN DD DISP=(OLD,KEEP),UNIT=TAPE
//LOGIN DD DISP=(OLD,KEEP),DSNAME=IMSLOG,
// UNIT=TAPE,VOL=SER=IMSPAC
//LOGOUT DD DISP=(NEW,PASS),DSNAME=EXAMPLE4.COPY1,
// SPACE=(CYL,(3,1)),UNIT=SYSDA,
// VOL=SER=IMSPAC
//

```

```

//SYSIN DD *
-----*
* CONTROL STATEMENT : SPECIFIED *
* INPUT = LOGIN *
* OUTPUT = (SYSPRINT AND LOGOUT) *
* * SINCE MULTIFIELD TESTS ARE BEING USED, *
* AND CONSIST OF MULTIPLE OPTION FUNCTIONS, *
* FINAL OUTPUT PROCESSING OF THE SELECTED RECORD *
* IS BASED UPON THE 'COPY' OPTION AND 'PRTSYS=Y' *
* KEYWORD BEING CODED WITH 'COND=E'. *
* SELECTION QUALIFIERS : *
* 1. USERID = USERBBBB *
* 2. LOG RECORD TYPES (X'16',X'50',X'51',X'52') *
-----*
CONTROL CNTL DDNAME=LOGIN,DDNOUT=LOGOUT
OPTION PRINT OFFSET=9,FLDTYP=C,VALUE=USERBBBB,FLDLLEN=8,COND=M
OPTION COPY PRTSYS=Y,OFFSET=5,FLDTYP=X,VALUE=16,FLDLLEN=1,COND=E
OPTION NEGOF OFFSET=12,FLDTYP=C,VALUE=USERBBBB,FLDLLEN=8,COND=M
OPTION COPY PRTSYS=Y,OFFSET=5,FLDTYP=X,VALUE=50,FLDLLEN=1,COND=E
OPTION NEGOF OFFSET=12,FLDTYP=C,VALUE=USERBBBB,FLDLLEN=8,COND=M
OPTION COPY PRTSYS=Y,OFFSET=5,FLDTYP=X,VALUE=51,FLDLLEN=1,COND=E
OPTION NEGOF OFFSET=12,FLDTYP=C,VALUE=USERBBBB,FLDLLEN=8,COND=M
OPTION COPY PRTSYS=Y,OFFSET=5,FLDTYP=X,VALUE=52,FLDLLEN=1,COND=E
END
/*
//

```

## 例 5

この例では、DFSERA10 の 1 回の実行で、選択されたログ・レコードを個々の出力データ・セットにコピーします。選択されたすべてのレコードが印刷されます。

```

//EXAMPLE5 JOB
/*
// EXEC PGM=DFSERA10
//STEPLIB DD DISP=SHR,DSNAME=IMS.SDFSRESL
/*
//SYSUT1 DD DISP=(OLD,KEEP),DSNAME=IMSL0G
/*
//SYSPRINT DD SYSOUT=A,UNIT=TAPE,VOL=SER=123456
/*
//LOGOUT1 DD DISP=(NEW,PASS),DSNAME=EXAMPLE5.COPY1,
// SPACE=(CYL,(3,1)),UNIT=SYSDA,
// VOL=SER=IMSPAC
//LOGOUT2 DD DISP=(NEW,PASS),DSNAME=EXAMPLE5.COPY2,
// SPACE=(CYL,(3,1)),UNIT=SYSDA,
// VOL=SER=IMSPAC
//LOGOUT3 DD DISP=(NEW,PASS),DSNAME=EXAMPLE5.COPY3,
// SPACE=(CYL,(3,1)),UNIT=SYSDA,
// VOL=SER=IMSPAC
/*
//SYSIN DD *
-----*
* CONTROL STATEMENT : SPECIFIED *
* INPUT = DEFAULT (SYSUT1) *
* OUTPUT = SYSPRINT AND (LOGOUT1,LOGOUT2,LOGOUT3) *
* SELECTION QUALIFIERS : *
* 1. LOG RECORD TYPE X'16' *
* 2. USERIDS = (USERAAAA,USERBBBB,USERCCCC) *
-----*
CONTROL CNTL DDNOUT=LOGOUT1
OPTION COPY OFFSET=9,FLDTYP=C,VALUE=USERAAAA,FLDLLEN=8,COND=M
OPTION COPY OFFSET=5,FLDTYP=X,VALUE=16,FLDLLEN=1,COND=E,PRTSYS=Y
END
-----*
CONTROL CNTL DDNOUT=LOGOUT2

```

```

OPTION COPY OFFSET=9,FLDTYP=C,VALUE=USERBBBB,FLDLEN=8,COND=M
OPTION COPY OFFSET=5,FLDTYP=X,VALUE=16,FLDLEN=1,COND=E,PRTSYS=Y
END
*-----*
CONTROL CNTL DDNOUT=LOGOUT3
OPTION COPY OFFSET=9,FLDTYP=C,VALUE=USERCCCC,FLDLEN=8,COND=M
OPTION COPY OFFSET=5,FLDTYP=X,VALUE=16,FLDLEN=1,COND=E,PRTSYS=Y
END
/*
//

```

## 例 6

この例は、OSAM イメージ・コピー・データ・セットのレコード 158、およびこのブロック番号を参照するログ・データ・セット上のすべてのタイプ X'50' のレコードを印刷するのに必要な、JCL と制御ステートメントを示しています (非ブロック化 OSAM を想定)。

```

//EXAMPLE6 JOB
//*
//          EXEC PGM=DFSERA10
//STEPLIB  DD DISP=SHR,DSNAME=IMS.SDFSRESL
//*
//SYSUT1   DD DISP=(OLD,KEEP),DSNAME=IMSL0G,
//          UNIT=TAPE,VOL=SER=123456
//*
//SYSPRINT DD SYSOUT=A
//*
//IMAGFILE DD DISP=(OLD,KEEP),DSNAME=OSAMIMAG,
//          UNIT=TAPE,VOL=SER=456789
//*
//SYSIN    DD *
*-----*
*      CONTROL STATEMENT : SPECIFIED      *
*          INPUT = IMAGFILE                *
*          OUTPUT = SYSPRINT               *
*      SELECTION QUALIFIERS :              *
*          1. OSAM RBN = 0000009E (RECORD NO. 158) *
*-----*
CONTROL CNTL STOPAFT=(1,E),DDNAME=IMAGFILE
OPTION PRINT OFFSET=1,FLDTYP=X,VALUE=0000009E,FLDLEN=4,COND=4
END
*-----*
*      CONTROL STATEMENT : DEFAULTS        *
*          INPUT = SYSUT1                   *
*          OUTPUT = SYSPRINT                *
*      SELECTION QUALIFIERS :              *
*          1. LOG RECORD TYPE X'50'         *
*          2. DATABASE NAME = DATABAS1     *
*          3. FLAG X'04' IN 7TH BYTE = 0 (OSAM DATA SET) *
*          4. OSAM RBN = 0000009E         *
*-----*
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=50,FLDLEN=1,COND=M
OPTION PRINT OFFSET=53,FLDTYP=C,VALUE=DATABAS1,FLDLEN=8,COND=M
OPTION PRINT OFFSET=7,FLDTYP=X,VALUE=04,COND=MTN
OPTION PRINT OFFSET=43,FLDTYP=X,VALUE=0000009E,FLDLEN=4,COND=E
END
/*
//

```

## 例 7

この例では、データベース名 (53 番目のバイトから始まる) が DB01DS01 ではない、タイプ X'50' のレコードとタイプ X'25' のレコードをすべて印刷するのに必要な JCL と制御ステートメントを示しています。

制御ステートメントの 2 番目のセットでは、最初のタイプ X'25' のレコードの 9 番目のバイトから始まるデータベース名を選択するために、シンボリック・キーワードを使用しています。次の制御ステートメント内の値に関して同じシンボリック名を使用すると、同じデータベース名を持つタイプ X'50' のレコードはすべて (最初のレコードは除く) が 53 バイト目から印刷されます。

```
//EXAMPLE7 JOB
//          EXEC PGM=DFSERA10
//STEPLIB DD  DISP=SHR,DSNAME=IMS.SDFSRESL
//*
//SYSPRINT DD  SYSOUT=A
//*
//LOGIN      DD  DISP=(OLD,KEEP),DSNAME=IMSL0G,
//            UNIT=TAPE,VOL=SER=123456
//*
//SYSIN      DD  *
*-----*
* CONTROL STATEMENT : SPECIFIED *
*           INPUT : LOGIN *
*           OUTPUT : SYSPRINT *
* SELECTION QUALIFIERS: *
*           1. LOG RECORD TYPE X'50' *
*           -= DB01DS01 STARTING IN THE 53rd BYTE *
*           (DATABASE NAME) PRINT 5 LOG RECORDS *
*           2. LOG RECORD TYPE X'25' *
*-----*
CONTROL  CNTL  DDNAME=LOGIN
OPTION   PRINT OFFSET=5,FLDTYP=X,VALUE=50,FLDLEN=1,COND=M
OPTION   PRINT OFFSET=53,FLDTYP=C,VALUE=DB01DS01,FLDLEN=8,STOPAFT=5, X
          COND=IE
OPTION   PRINT OFFSET=5,FLDTYP=X,VALUE=25,FLDLEN=1,COND=E
END
*-----*
* CONTROL STATEMENT : SPECIFIED *
*           INPUT : LOGIN *
*           OUTPUT : SYSPRINT *
* SELECTION QUALIFIERS: *
*           1. LOG RECORD TYPE X'25' *
*           DEFINE SYMBOL &DBNAME STARTING IN THE 9th BYTE *
*           (DATABASE NAME ) & PRINT 1 RECORD *
*           2. LOG RECORD TYPE X'50' *
*           USE SYMBOL &DBNAME FOR DATABASE NAME STARTING *
*           IN THE 53rd BYTE & SKIP THE FIRST SELECTED *
*           RECORD *
*-----*
CONTROL  CNTL  DDNAME=LOGIN
OPTION   PRINT OFFSET=5,FLDTYP=X,VALUE=25,FLDLEN=1,COND=M
OPTION   PRINT OFFSET=9,FLDTYP=C,FLDLEN=8,SYM=&DBNAME,STOPAFT=1,COND=E,
OPTION   PRINT OFFSET=5,FLDTYP=X,VALUE=50,FLDLEN=1,COND=M
OPTION   PRINT OFFSET=53,FLDTYP=C,VALUE=&DBNAME,FLDLEN=8,STARTAF=1, X
          COND=E,
END
/*
//
```

## 例 8

この例は、外部トレース・データ・セットを印刷するときに必要な JCL と制御ステートメントを示しています。

```
//EXAMPLE8 JOB MSGLEVEL=(1,1)
//*
//PRTTAB EXEC PGM=DFSERA10
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//*
//SYSUT1 DD DISP=SHR,DSN=IMS.EXTERNAL.TRACE
//*
//SYSPRINT DD SYSOUT=A
//*
//SYSIN DD *
*-----*
* CONTROL STATEMENT : SPECIFIED *
* INPUT = SYSUT1 *
* OUTPUT = SYSPRINT *
* SELECTION QUALIFIERS : *
* 1. Log record type X'67FA' in fifth and sixth *
* byte = (all trace log records) *
*-----*
CONTROL CNTL SKIP=0
OPTION PRINT OFFSET=5,VALUE=67FA,FLDLLEN=2,COND=E,EXITR=DFSERA60
END
/*
//
```

## 例 9

この例では、IMS ログからの 67FF レコードを印刷するときに必要な JCL および制御ステートメントを示します。

```
//EXAMPLE9 JOB
//*
// EXEC PGM=DFSERA10
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//*
//SYSPRINT DD SYSOUT=A
//*
//LOGIN DD DISP=SHR,DSN=IMSLOG
//*
//SYSIN DD *
CONTROL CNTL DD=LOGIN
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=67FF,FLDLLEN=2,COND=M
OPTION PRINT OFFSET=29,FLDTYP=X,VALUE=F0F8F3F2,FLDLLEN=4,COND=E,EXITR=DFSERA30
END
/*
//
```

## 例 10

この例は、OM ログ・レコードを印刷するときに必要な JCL と制御ステートメントを示しています。

```
//EXAMPLE10 JOB
//*
//STEP1 EXEC PGM=DFSERA10
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//*
//SYSUT1 DD DCB=(BLKSIZE=32760),DSN=SYSLOG.OM.AUDIT.TRAIL.LOG,
// SUBSYS=(LOGR,IXGSEXIT)
//*
//SYSPRINT DD SYSOUT=A
```

```

/**
//SYSIN      DD
CONTROL     CNTL   STOPAFT=EOF
OPTION      PRINT  EXITR=CSLULALE
END
/*
//

```

## DFSERA10 ユーティリティー・モジュール

ファイル選択フォーマット設定印刷ユーティリティーは、DFSERA10 ユーティリティーに戻りコードを渡す複数のモジュールを呼び出します。

以下のトピックでは、これらのモジュールに関する追加情報を提供します。

### レコード・フォーマット設定・印刷モジュール (DFSERA30)

レコード・フォーマット設定・印刷モジュール (DFSERA30) は、トレースおよび汎用サブレコード・タイプ (X'00' および X'01') および SNAP サブレコード・タイプ (X'FD' および X'FF') をフォーマット設定するときに使用します。

その他のログ・レコードは、z/OS ダンプ・フォーマットでフォーマット設定されます。DFSERA30 は、ファイル選択フォーマット設定印刷ユーティリティー (DFSERA10) の出口ルーチンです。このルーチンはログ・レコードをフォーマット設定するので、戻りコードを DFSERA10 へ渡します。この戻りコードは、ログ・レコードは処理されており、これ以上処理は必要ではないことを DFSERA10 に伝えます。

トレースおよび SNAP サブレコード・タイプの場合には、このモジュールは、ログ・レコード先行情報を作成し、その後そのログ・レコード内の各エレメントの定様式印刷出力を続けます。

DFSERA30 では、ダンプされる各レコード内の STCK 値を可読タイム・スタンプに変換し、この値をレコードのヘッダー行に印刷します。この値はハードウェア・クロックから得られたものであるため、次のことを心得ておく必要があります。

- 時刻は UTC (GMT) であり、地方時ではない。
- ハードウェア・クロックは、使用されるシステムに存在する可能性のある、うるう秒調整を含まない (CVT フィールド CVTISO を参照)。このため、レコードが書き込まれたとき、DFSERA30 によって出力される時刻が z/OS の報告する時刻と異なる場合があります。この差は、うるう秒調整量に等しい値です。

### ユーティリティー制御ステートメント

以下に、DFSERA30 出口ルーチンを使用してタイプ X'67' ログ・レコードをフォーマット設定するために必要な制御ステートメントを示します。

```

Column 1      Column 10                                     72
CONTROL       CNTL
OPTION        PRINT OFFSET=5,FLDLEN=2,VALUE=67aa,        X
              COND=E,EXITR=DFSERA30                       X
END
/*
//

```



この図で、aa はログ・レコード・サブタイプです。

**aa=01** TRACE ログ・レコード・サブタイプを指定します。

**aa=FD**

SNAP ログ・レコード・サブタイプを指定します。

**aa=FF** ABEND ログ・レコード・サブタイプを指定します。

以下に、DFSERA30 の出力の例を示します。AE9004 は、ログ・レコード作成時の LXB のストレージ・アドレスです。各行の第 2 欄は、LXB からの相対オフセットです。

```
DFSERA30 -- FORMATTED LOG PRINT
```

```
⋮
```

```
INTERNAL TRACE RECORD
```

```
⋮
```

```
LXB
```

AE9004	000000	807F0BC9	00093660	00AE9350	00AE92B0	00091E90	00AE991C	17000000	7F0C0000
AE9024	000020	80000000	520821CE	0008229C	000820C6	80082194	012141CE	60000054	0A000000
AE9044	000040	30000005	022140C6	600000CE	09000000	30000005	47000000	20000001	00000000
AE9064	000060	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
AE9084	000080	TO AE90C4	0000C0	SAME AS ABOVE					
AE90E4	0000E0	00000000	0C419317	F1044193	17F10441	9337E218	D243F510	A314A8C3	419101A2
AE9104	000100	02F30C41	93179101	A502F004	F30C4193	17F10441	93170000	00000000	00B66218

## デッドロック報告書

デッドロック報告書には、777 および 123 疑似異常終了と非メッセージ・ドリブン BMP でのデッドロックの結果として生じたデッドロックのリソースおよびリソース所有者に関する情報が入ります。(これらによって、'FD' 状況コードが出されます。)

DFSERA30 はこのデッドロック・ブロックを検出すると、そのブロックを印刷し、ブロック内のデータに基づいて報告書を作成します。過度にデッドロックが起きる場合には、デッドロック・ブロックとそれに基づく報告書によって、そのデッドロックに関係しているリソースを分析することができます。

外部サブシステムがデッドロックを検出したために IMS が 777 または 123 疑似異常終了でアプリケーションを異常終了した場合、デッドロック報告書には、そのサブシステム、ジョブ、およびデッドロックを受け取ったリカバリー単位を識別する情報が含まれます。

## BMP 領域と MPP 領域のデッドロック報告書

以下の図は、BMP 領域と MPP 領域に関係している単純なデッドロックについての DFSERA30 報告書の例です。リソース 1 と 2 を待機している MPP プログラムが被害の対象となっています。このプログラムは、キー 'KK360' についてのルート・ロックを要求しています。このロックの保有者は BMP プログラムです。この BMP プログラムは、キー 'KK130' についてのルート・ロックを要求しています。このロックの保有者は MPP プログラムです。

```
DEADLOCK ANALYSIS REPORT - LOCK MANAGER IS IRLM
```

```
RESOURCE DMB-NAME LOCK-LEN LOCK-NAME      - WAITER FOR THIS RESOURCE IS VICTIM  
01 OF 02 DHVNTZ02    08    00000BC4800501D7
```

```
KEY IS ROOT KEY OF DATA BASE RECORD ASSOCIATED WITH LOCK
```

KEY=(KK360)

```
IMS-NAME TRAN/JOB PSB-NAME DBD/PCB# PST# RGN CALL LOCK LOCKFUNC STATE
WAITER SYS3      NQF1      PMVAPZ12 DLVNTZ02 0002 MPP  GET  GRIDX 30400358 06-P
BLCKER SYS3      DDLKBMP1 PLVAPZ22      0027 0003 BMP  ---- - - - - - - - - - - - - - - 06-P
```

```
RESOURCE DMB-NAME LOCK-LEN LOCK-NAME
02 OF 02 DHVNTZ02      08      00000924800501D7
```

KEY IS ROOT KEY OF DATA BASE RECORD ASSOCIATED WITH LOCK  
KEY=(KK130)

```
IMS-NAME TRAN/JOB PSB-NAME DBD/PCB# PST# RGN CALL LOCK LOCKFUNC STATE
WAITER SYS3      DDLKBMP1 PLVAPZ22 DLVNTZ02 0003 BMP  GET  GRIDX 30400358 06-P
BLCKER SYS3      NQF1      PMVAPZ12      0006 0002 MPP  ---- - - - - - - - - - - - - - - 06-P
```

DEADLOCK ANALYSIS REPORT - END OF REPORT

この例では、X'30' は LOCK の欄で GRIDX として報告されています。提供される定様式デッドロック情報を完全に理解するためには、DL/I のロッキングに関する用語とデータ編成を多少理解している必要があります。

### 報告書の読み方

定様式報告書は、ロック名ごとに要約されます。これは、n 個のうちのロック 1 から始まり、ロックされているデータベース名、ロック名の長さ、およびロック名そのものを示します。ロック名はいくつかのコードから構成されます。これらのコードは、相対ブロック・アドレス (RBA)、ロックが全機能 (FF) データベースまたは高速機能 (DEDB) データベースのいずれで起きたか、さらに DEDB 内で起きた場合には、それが制御インターバル (CI) レベルであるかセグメント・レベルであるかといった、ロックに関する情報を提供します。

FF データベースの場合、RBA はロック名のバイト 1 から 4 に表示されます。例えば、ロック名 00000924800501D7 では、RBA= 924 です。

FP (高速機能) データベース内のロックの RBA の判別は、これより少し複雑になります。以下の表は、FP データベースのロック名は細分されることを示します。

表 23. FP データベース内のロック名

バイト位置	ロック情報
1	ロック ID
2 から 4	相対バイト・アドレス
5 から 6	DMCB 番号
7	エリア番号
8	高速機能 ID の C6

FP データベースでは、最初の 2 桁 (バイト 1) には、ロックがセグメント・レベルで起きた場合は数字の「80」が表示されます。この場合、次に表示される 3 バイトは 30 ビット RBA を示します。実際の RBA を知るには、表示された RBA に 4 を掛ける必要があります。

ロック名の最初のバイトは、ロックされるリソースを識別するロック ID です。DSECT は ADFS MAC(DBFEPST) 内にあります。これは、高速機能ロックに対し、可能なすべてのロック ID を提供します。

EPSTRSID DS	0F	RESOURCE NAME
EPSTLKID DC	X'00'	LOCK SUB ID
EPSTCILK EQU	X'00'	CI LOCK
EPSTSLLK EQU	X'80'	SLL LOCK - '80' ON &
EPSTSLLL EQU	X'40'	- '40' OFF
EPSTUWLK EQU	X'E4'	UOW LOCK, X'E4'=C'U'
EPSTDULK EQU	X'F0'	DUMMY LOCK
EPSTMDLK EQU	X'F1'	MSDB LOCK
EPSTOVLK EQU	X'F2'	BUFFER OVERFLOW LOCK
EPSTNRLK EQU	X'F3'	DBRC NOT REGISTERD DB
EPSTARLK EQU	X'F8'	AREA LOCK
EPSTCLMK EQU	X'FF'	COMMAND LOCK

ロックが CI レベルで起きた場合は、最初の 2 桁はコード X'00' を示します。この場合、次に表示される 3 バイトは 24 ビット RBA を示します。実際の RBA を知るには、表示された RBA に 256 (X'100') を掛ける必要があります。

さらに、FP データベース内で起きたロックでは、ロック名の最後の 2 桁 (バイト 8) には、コード「C6」が表示されます。

例えば、ロック名 80000C02800101C6 は、FP データベース内でセグメント・レベルで起こったロックで、RBA は「00003008」であることを示しています。

多くの場合、ロックは、ルート・キーが認識されているデータベース・レコードについてのものです。次の行には、ロックされているデータベース・レコードのルート・キーに関する情報が示されます。ルート・キーについて起こる可能性のある報告書のステートメントを以下に示します。

- KEY IS ROOT KEY OF DATA BASE RECORD ASSOCIATED WITH LOCK

このステートメントは最も一般的なものです。これは、後に続くキーが、ロックに関係しているデータベース・レコードのルート・キーであることを示します。この報告書ステートメントは、例えば、HIDAM または PHIDAM ルートが索引を使用して検索されるときに表示されます。ルートのロックが要求されたときに、キーが分かります。

- KEY FOR RESOURCE IS NOT AVAILABLE

これは、ロックされているデータベース・レコードのキーが利用不能であることを示します。この報告書ステートメントは、例えば、HDAM または PHDAM データベースの GN 呼び出しによって、DL/I が次のルート・アンカーをロックしたときに表示されます。このロック要求が、デッドロックに関係しているリソースの 1 つである場合には、ロックに関連するキーを印刷することはできません。

- LOCKING PRIOR ROOT FOR HIDAM ROOT INSERT, KEY DISPLAYED IS FOR NEXT HIGHER ROOT

これは、ルートが HIDAM または PHIDAM に挿入され、ルートに兄弟順方向ポインターと兄弟逆方向ポインターがある場合に生じる可能性があります。例え

ば、キー 10 と 12 があり、11 が挿入されている場合には、この報告書ステートメントが表示されます。表示されるキーはキー 12 ですが、ロックはキー 10 に対するものです。

- LOCKING ON NEXT HIDAM ROOT FOR GN CALL, KEY DISPLAYED IS FOR PRIOR HIDAM ROOT

順方向および逆方向に兄弟ポインターを設定して HIDAM または PHIDAM を使用し、キー 10、11、および 12 が存在し、位置がキー 10 にある場合に、このステートメントが表示されることがあります。GN 呼び出しでは 11 に対するロックが必要です。ロックが要求されたときには、キーが分からないので、前のルートのキーが表示されます。

- LOCKING ON HDAM ANCHOR, KEY DISPLAYED IS HDAM KEY REQUESTED

HDAM または PHDAM を使用しているときに表示される場合があります。ロックされる項目はアンカーです。アンカーがロックされると、検索されるキーは分かりませんが、要求されるキーは分かるので、それが表示されます。

ロックに DEDB が関与している場合は、次の報告書ステートメントが発行される可能性があります。

- SPECIAL INTERNAL LOCK FOR PROMOTE

高速機能が READ (01) レベルまたは EXCLUSIVE (04) レベルでロックを要求しています。このステートメントは、GU 呼び出しの直後に同じセグメントに対する GHU 呼び出しが続いた場合に発生する可能性があります。その結果としてのロックを 01 から 04 にレベル上げる要求が、デッドロックの原因となりました。

ロックの待機者および保持者/所有者の情報が次に印刷されます。待機および保持の作業単位は、IMSID、tranname または jobname、PSB 名、PST 番号、および領域タイプで識別されます。リストされる WAITER は、データベース・キー情報が関係している作業単位です。

WAITER (待機者) と HOLDER (保有者) の 2 行の情報にはいくつかの相違があります。現行の PCB 名、DL/I 呼び出し、およびロック要求は、待機者のみに関係します。ロックの保有者はこの情報を利用できません。ロックの保有者には、相対 DBPCB 番号が出力されます。

処理中の現行 DL/I 呼び出しは、以下のいずれかの呼び出しとして報告されます。

**GET** DL/I 呼び出しは、GU、GHU、GN、GHN、GNP、または GHNP でした。取り込まれた情報では、特定の GET 呼び出し機能の分類はできません。

**REPL** DL/I 呼び出しは REPL でした。

**ISRT** DL/I 呼び出しは、ISRT または ASRT でした。

**DLET** DL/I 呼び出しは DLET でした。

**POS** DL/I 呼び出しは、MSDB の POS 呼び出しでした。

ロック要求機能は、LOCK および LOCKFUNC の欄で報告されます。LOCKFUNC の最初のバイトは、便宜上変換されます。LOCKFUNC は、LRHPARM DSECT によってマップされる 16 進数機能、モード、状態、およびフラグです。

ロック要求機能を変換する理由は、ブロック・レベルのデータ共用で起きたデッドロック、異なる順序でのアプリケーション・プログラムによるデータ・アクセスで起きたデッドロック、あるいはそれらの両方によって起きたデッドロックを識別するためです。デッドロックの目的について、ロック要求機能を次の呼び出しのように要約することができます。ここで、*x* は通常、L、X、B、P、U、または W です。

#### **GBID<sub>x</sub>**

ブロック・ロックを獲得します。ブロック・レベル共用のみです。

#### **GZID<sub>x</sub>**

データ・セット使用中ロックを獲得します。データ・セットのオープン、クローズ、および拡張のシリアライズ化のためだけに使用します。デッドロックに巻き込まれた場合は、IMS コード内のエラーの可能性がります。

#### **GXID<sub>x</sub>**

データ・セット拡張ロックを獲得します。データ・セットの拡張をシリアライズ化するために使用します。ブロック・レベル共用の場合のみで、おそらくは HISAM データベースです。

#### **GRID<sub>x</sub>**

データベース・レコードのルートのロックを獲得します。

#### **GQCM<sub>x</sub>**

Q コマンド・コード・ロックを獲得します。これは、特定のデータに対するアプリケーションからのロックです。GQCM 機能は、全機能のみに適用されます (高速機能では、Q コマンド・コードが出されるときに新しいロックを獲得しません)。

#### **GSEG<sub>x</sub>**

従属セグメントについてのセグメント・ロックを獲得します。これは、IRLM がロック・マネージャーであるときには使用されません。

#### **GFPLL**

高速機能ロックを獲得します。

### **ロック状態**

ロック状態はロックのタイプまたはレベルであり、通常は数値で指定されます。ロックの状態を管理するために、IMS は内部リソース・ロック・マネージャー (IRLM) またはプログラム分離 (PI) ロック・マネージャーを使用します。この 2 つのロック・マネージャーがロックのレベルを反映するのに使用する状態は同じではありません。PI ロック・マネージャーは 4 つの状態をサポートし、IRLM は 11 の状態をサポートしますが、IMS が使用するはそのうちの 8 つだけです。

ロック状態は、数字ではなく名前で示されることがあります。PI がサポートしている 4 つの状態を表すのに使用されている名前は、次のとおりです。

#### **状態 1**

読み取り専用

- 状態 2  
読み取り
- 状態 3  
更新
- 状態 4  
排他的

以下の表は、着信ロック要求のレベルと PI ロック・マネージャーの使用時にロックが保有されるレベルの互換性を記述するマトリックスです。互換状態は「C」(ロック要求が許諾されることを意味する)で示され、非互換状態は「X」(互換要求が許諾されないことを意味する)で示されます。

表 24. PI ロックの互換性マトリックス

要求されるレベル	1	2	3	4
保有レベル 1	C	C	C	X
保有レベル 2	C	C	X	X
保有レベル 3	C	X	X	X
保有レベル 4	X	X	X	X

IRLM の場合の 8 つの状態およびそれらの特性は、2 つのマトリックスで定義されます。一方のマトリックスは、結果の状態を決めるときに使用され、もう一方のマトリックスは、要求および保有の作業単位の互換性を決めるときに使用されます。

結果の状態 の概念については、いくらかの説明が必要です。簡単に言えば、結果の状態とは、現行要求の許諾の結果であるロック状態、または現行要求が許諾されたものと想定して、後続のリクエストが置かれる「保有」状態のことです。IRLM では、1 つの特定の作業単位が 1 つのリソースを一度より多くロックできるため、ある作業単位が 2 回目にリソースをロックするときに異なる状態を指定した場合には、ロックの最終的な保有状態は、最初のロックで与えられた特権を失うことなく 2 番目の状態の特権を与えるものでなければなりません。

厳密に階層的な特権順序を持つ状態のセットが与えられた場合には、2 つの状態のうちの上位の状態を付与するだけで十分です。しかし、それぞれの上位の状態が、先行する状態の特権すべてを必ずしも含まないロッキング・プロトコルを可能にするためには、マトリックスで指定すると、結果の状態に他の 2 つの状態の特権の合計を与え、3 番目の状態にすることができます。そうすれば、要求は 3 番目の状態の要求として処理されます。以下の表は、結果の状態のマトリックスです。

表 25. IRLM の結果の状態のマトリックス

要求されるレベル	1	2	3	4	5	6	7	8
保有レベル 1	1	2	3	4	5	6	7	8
保有レベル 2	2	2	3	4	5	6	7	8
保有レベル 3	3	3	3	6	5	6	7	8
保有レベル 4	4	4	6	4	5	6	7	8
保有レベル 5	5	5	3	5	5	6	7	8
保有レベル 6	6	6	6	6	6	6	7	8
保有レベル 7	7	7	7	7	7	7	7	8

表 25. IRLM の結果の状態のマトリックス (続き)

要求されるレベル	1	2	3	4	5	6	7	8
保有レベル 8	8	8	8	8	8	8	8	8

次の表は互換性マトリックスを示しており、互換は「C」、非互換は「X」で表されています。

表 26. IRLM の互換性マトリックス

要求されるレベル	1	2	3	4	5	6	7	8
保有レベル 1	C	C	C	C	C	C	C	X
保有レベル 2	C	C	C	C	C	C	X	X
保有レベル 3	C	C	C	X	X	X	X	X
保有レベル 4	C	C	X	C	C	X	X	X
保有レベル 5	C	C	X	C	X	X	X	X
保有レベル 6	C	C	X	X	X	X	X	X
保有レベル 7	C	X	X	X	X	X	X	X
保有レベル 8	X	X	X	X	X	X	X	X

IRLM では、状態は専用の属性を持つことができます。専用属性に意味があるのは、ブロック・レベル・データ共用を使用する場合のみです。専用属性は、単一の IMS のさまざまなスレッドへのロックの認可には影響を与えません。専用属性は、ロックがこの IMS の専用でなければならない (その IMS のみに認可される) ことを示します。

制約事項: データを共用している別の IMS のスレッドは、ロックを認可されません。

専用は、'-P' とその後のロック状態で示されます。

### ブロックの限度

デッドロック・サイクル内の各リソースのデータを保持するときに、固定サイズのブロックが使用されます。このブロックは、9 つのリソースを含むサイクルのデータを保持できるだけの十分な大きさです。サイクルに 9 つのリソースを超えるリソースが含まれている場合には、メッセージでそのことが示され、最初の 9 つのみが報告されます。

データを保持するためのブロックの数は限られています。デッドロックが起きたときにすべてのブロックが使用中であると、メッセージでそのことが知らされ、そのデッドロックについてはデッドロック情報は提供されません。

### 収集された補足情報

定様式デッドロック報告書は、収集されてただちにログにスナップされた完全なデータの要約です。生データの情報をマップする、2 つのマクロ DSECT があります。DIPENTRY DSECT と DLKDL DSECT です。

## デッドロックの報告での例外

1 つのデッドロックのみ、報告形式が異なります。

```
LOCK 1                LOCK 2
PST 1 owns STATE SHR  PST 2 owns STATE UPD
PST 3 waits STATE UPD  PST 1 waits STATE UPD
PST 2 waits STATE SHR
```

PST 3 のアプリケーション 3 が、互換性がない状態で LOCK 1 を要求するようなことをしなければ、PST 2 は所有者の PST 1 と互換性がある状態で LOCK 1 を要求しているので、デッドロックは起きなかったはずです。

このデッドロックの報告で起きた例外は、LOCK 1 が 2 回リストされている点にあります。これは、PST 1 が所有していて PST 3 が待機している状態で 1 回リストされ、PST 2 が待機していて保有者情報のない状態で再びリストされています。報告書では、BLCKER の IMS-NAME フィールドに NOTAVAIL が表示されません。

## スナップからのデッドロック・エレメントのみの選択

777 疑似異常終了でスナップされるエレメントは、他の疑似異常終了の場合よりも数が少なくなりますが、そのスナップは、デッドロック・ブロックよりも多くのエレメントを含みます。スナップから特定のエレメントのみを選択することができます。以下の図は、任意の疑似異常終了スナップからデッドロック・エレメントのみを選択するときの DFSERA10 制御ステートメントを示しています。

```
//SYSIN    DD *
*-----*
*
*          CONTROL STATEMENT : DEFAULTS
*                INPUT = SYSUT1
*                OUTPUT = SYSPRINT
*
*          SELECTION QUALIFIERS :
*                1. LOG RECORD TYPE OF X'67FF'
*                2. NAME OF BLOCK WITHIN SNAP IS C'DEADLOCK'
*
*          EXIT ROUTINE = DFSERA30
*
*-----*
OPTION PRINT OFFSET=5,FLDLLEN=2,VALUE=67FF,FLDTYP=X,COND=M
OPTION PRINT OFFSET=33,FLDLLEN=8,VALUE=DEADLOCK,FLDTYP=C,COND=E,    X
          EXITR=DFSERA30
END
/*
//
```

## IMS のサブシステム検出デッドロック

外部サブシステムがデッドロックを検出したために IMS が U777 でアプリケーションを異常終了した場合、デッドロック報告書には、そのサブシステム、ジョブ、およびデッドロックを受け取ったリカバリー単位を識別する情報が含まれます。

以下の図は、外部サブシステムが検出したデッドロックに関する DFSERA30 報告書の例です。

```
PSEUDO ABEND RECORD ABEND NO = 0777 RECNO = 00000162 TIME 12:24:07.1 DATE 2006.292
DEADLOCK
```



```
EXTERNAL SUBSYSTEM SSOP      DETECTED A DEADLOCK DURING NORMAL CALL
REGION TYPE      : MPP
REGION NUMBER   : 0001
JOB NAME        : MPP1
PSB NAME        : DCSQL7B
SMB NAME        : TXSQL7B
RECOVERY TOKEN  : E2E8E2F3404040400000000500000000
```

関連概念:

 DB/DC および DCCTL でのロッキング・メカニズムとデータベース保全性 (システム管理)

## プログラム分離トレース・レコード・フォーマット設定・印刷モジュール (DFSERA40)

プログラム分離 (PI) トレース・レコード・フォーマット設定・印刷モジュールは、ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) から出力ルーチンとしてタイプ X'67FA' のログ・レコードを受け取り、それらのレコードを SYSPRINT データ・セット上でフォーマット設定します。

このトピックにはプロダクト・センシティブ・プログラミング・インターフェース情報が含まれています。

これらのログ・レコードは、PI (プログラム分離) トレース、DFSLRH00 に対するトレース PI エンキューおよびデキューの各呼び出し、および DL/I アナライザーへの DL/I 呼び出しで作成されます。DL/I アナライザーは、すべての DL/I 呼び出しを処理します。トレースがアクティブなときに、DL/I アナライザー呼び出しがトレースされます。標準 ENQ/DEQ 呼び出しは、DFSLR マクロ命令で呼び出されます。

PI トレースは、IMS オンライン環境で /TRACE コマンドによって実行されるか、あるいは LOCK=OUT を指定した OPTIONS ステートメントによって実行されます。

データ共用環境では、PI トレースがアクティブで、ログに記録されていれば、PI トレース・ロガーが IMS ロック・マネージャー (DFSLMGR0) によってアクティブにされ、IRLM への出口になります。

PI トレース・レコード・フォーマット設定・印刷モジュールは、ファイル選択フォーマット設定印刷ユーティリティの実行時にロードされるので、LINKLIB にあるか、JOBLIB または STEPLIB データ・セットになければなりません。

### 出力

以下に DFSERA40 からの出力例を示します。フィールド間のスペースは変更されません。

```
DATE: 05/11/10
MODULE PST TIME (*=ET) CALLR ACT LEV WC WFC SEQN FDBK RC PC ID= (RBA DMB DCB SUF) CLS TOKEN COMMENT
LRHO 01 GZIDB 0ABE
UNCHN 02 0AC0 61
LRHO 01 RZIDP 0AC1
PIEX 01 23:36:22.472 DLI TNFQ UPD 00 00 0AC3 0000 481075C5 8007 01
LRHO 01 TTLKX 0AC4
PIEX 01 23:36:22.472 DLI ENQ UPD 00 00 0AC5 0000 00000658 8006 01 00722050
LRHO 01 GRIDX 0AC6
```

PIEX	01	23:36:22.474	APP	ENQ	SHR	00	00	0ACA	0000	00000694	8006	01	0	007220DC
LRHO	01			GCCMX				0ACB						
DLA0	01	23:36:22.493		GU				0ACE		8				DL/I CALL
PIEX	01	23:36:22.493	DLI	IDEC	UPD	00	00	0ACF	0000	00000658	8006	01		
LRHO	01			RRIDX				0AD0						
LRHO	01			GZIDB				0AC1						
LRHO	01			RZIDB				0AD4						
PIEX	01	23:36:22.495	DLI	ENC	UPD	00	00	0AD6	0000	00001108	8006	01		00722050
LRHO	01			GRIDX				0AD7						
PIEX	01	23:36:22.496	DLI	IDEQ	UPD	00	00	0ADA	0000	00001108	8006	01		
LRHO	01			RRIDX				0ADB						
DLA0	03	23:36:23.614		GU				0ADE		1				DL/I CALL
LRHO	03			GZIDB				0AE4						
LRHO	03			RZIDB				0AE7						
PIEX	03	23:36:23.735	DLI	TENQ	UPD	00	00	0AE9	0000	48107105	8007	01		
LRHO	03			TTLKX				0AEA						
PIEX	03	23:36:23.736	DLI	ENC	UPD	00	00	0AEB	0000	00000408	8006	01		00722050
LRHO	03			GRIDX				0AEC						
PIEX	03	23:36:23.737	APP	ENQ	SHR	00	00	0AF0	0000	00000428	8006	01	0	00722014
LRHO	03			GQCMX				0AF1						
DLA0	03	23:36:23.834		GU				0AF5		2				DL/I CALL
PIEX	03	23:36:23.835	DLI	IDEQ	UPD	00	00	0AFA	0000	00000408	8006	01		
LRHO	03			FRIDX				0AFP						
LRHO	03			GZIDB				0AFC						
LRHO	03			RZIDB				0AFF						
PIEX	03	23:36:23.838	DLI	ENQ	UPD	00	00	0B01	0000	00001108	8006	01		00722050
LRHO	03			GRIDX				0B02						
PIEX	03	23:36:23.840	DLI	TDEQ	UPD	00	00	0B05	0000	00001108	8006	01		
LRHO	03			RRIDX				0B06						
DLA0	02	23:36:27.257		GHU				0B0F		4				DL/I CALL
PIEX	02	23:36:27.257	DLI	TDEQ	UPD	00	00	0B10	0000	0000087C	8006	01		
LRHO	02			RRIDX				0B11						
LRHO	02			GZIDB				0B12						
LRHO	02			RZIDB				0B15						
PIEX	02	23:36:27.263	DLI	TENQ	UPD	00	00	0B17	0000	481071C5	8007	01		
LRHO	02			TTLKX				0B18						
PIEX	02	23:36:27.263	DLI	ENQ	UPD	00	00	0B19	0000	00000408	8006	01		007220A0
LRHO	02			GRIDX				0B1A						
PIEX	02	23:36:27.265	DLI	TENQ	UPD	01	00	0B1E	1800 04	00000428	8006	01		00722014
PIEX	03	23:36:45.079	APP	CEQ	SHR	00	00	0B34	0000				0	
LRHO	03			RQCML				0B35						
PIEX	02	0:17.850*	DLI	UNK	RD			0B37		6F				SEQ2=0B1E
LRHO	02			TTLKL				0B38		04				
DLA0	02	23:36:45.982		GHU				0B3A		5				DL/I CALL
PIEX	02	23:36:45.982	DLI	TDEQ	UPD	00	00	0B3B	0000	00000408	8006	01		
LRHO	02			RRIDX				0B3C						
LRHO	02			GZIDB				0B3D						
LRHO	02			BZIDB				0B40						
PIEX	02	23:36:45.986	DLI	TENQ	UPD	00	00	0B42	0000	481075C5	8007	01		
LRHO	02			TTLKX				0B43						
PIEX	02	23:36:45.986	DLI	ENQ	UPD	00	00	0B44	0000	00000658	8006	01		007220A0
LRHO	02			GRIDX				0B45						

### 列見出しの説明

**日付** PI トレースの開始日付を表します。TIME フィールドは、この日付を基準にしたものです。

### MODULE

DFSLRH00 への DFSLR 呼び出しを出したモジュール、あるいは IRLM または DFSFXC10 を呼び出したモジュールを表します。示されている 4 文字は、モジュールの完全な名前 DFSxxxx0 の xxxx 部分です。

**PST** (PSTPSTNR からの) プログラム仕様テーブル (PST) の番号を表します。

**TIME** 呼び出しの時刻を、トレースが開始された日付を基準にして、HHH:MM:SS.UUU (UUU はミリ秒) で表します。戻りコード (RC) が 04 で、PI トレース・タイミングが呼び出し時にアクティブな場合は、この報告書のこの PST の次のレコードは、このフィールドのエンキュー待ちの経過時間を示します。時間は、MM:SS.UUU\* で示され、「\*」はこれが経過時間であることを示します。

### CALLR

呼び出し側のタイプ (DLI、FP、APP) を表します。

**ACT** 要求されたアクションを表します。

- LEV** この呼び出しの制御レベルを表します。
- RD** 読み取り専用
  - SH** 共用
  - UPD** 更新
  - EXC** 排他的
- WC** この PST を待機させた状態でこのリソースを保有している PST の個数
- WFC** この PST がこのリソースを解放するのを待っている PST の個数
- SEQN**  
対応する内部トレースのシーケンス番号を表します。
- FDBK**  
DFSFXC10 または IRLM からの 2 バイトのフィードバック情報です。
- RC** DFSFXC10 または IRLM からの戻りコードを表します。
- 00** 正常終了
  - 04** 呼び出し側は、要求したリソースの制御を得るために IMS を待機しなければなりません。
  - 08** デッドロック。要求は許可されません。このトランザクションで、内部疑似異常終了、バックアウト、および自動再スケジューリングが起きます。
  - 0C** 無効な呼び出し
- PC** エンキュー待機の後続く PST 通知コードを表します。このフィールドは、RC が 04 で、TIME フィールドの最後に「\*」がある場合にのみ存在します。
- 60** デッドロックが起きました。このトランザクションで、内部疑似異常終了、バックアウト、および自動再スケジューリングが起きます。
  - 61** /STO REG# ABDUMP コマンドに代わって、PST が PI ロックの PI 待ちチェーンから除去されました。
  - 6F** リソースの制御権を獲得しました。
- ID=** エンキューまたはデキューが行われているリソースの 8 バイト識別を表します。これには、4 バイトの RBA、2 バイトの DMB 番号、1 バイトの DCB 番号、および 1 バイトの SUF (接尾部) フィールドが含まれています。
- CLS** APP タイプの呼び出し側の場合は、要求された Q コマンド・コード・クラスを表します。LMGR トレースの場合は、CLASS パラメーターを表します。
- CLS は、全機能のみに適用されます (高速機能はロック・クラスをサポートしません)。
- TOKEN**  
この呼び出しでロックまたはエンキューされた制御ブロックのアドレス、または呼び出しタイプがアンロックまたは DEQ 呼び出しの場合には、ロック・マネージャーに渡される制御ブロックのアドレスです。

## COMMENT

トレースが DFSDLA00 から要求された場合は 'DL/I CALL' を指定します。他のコメントは、LMGR トレースに関するものです。

## ユーティリティー制御ステートメント

以下に、DFSERA40 に必要な制御ステートメントを示します。

```
桁 1          桁 10          桁 16          72
CONTROL      CNTL
OPTION       PRINT          OFFSET=5,VALUE=67FA,FLDLEN=2,      X
                                COND=E,EXITR=DFSERA40
END
/*
//
```

## DL/I 呼び出しイメージ・キャプチャー・モジュール (DFSERA50)

トレース・データが IMS ログ・データ・セットに送られる場合、ユーティリティー DFSERA10 および特殊 DL/I 呼び出しイメージ・キャプチャー・ルーチン DFSERA50 を使用してそれを検索できます。DFSERA50 は、検索される DL/I 呼び出しイメージ・キャプチャー・レコードの非ブロック化、フォーマット設定、および番号付けを行います。

DFSERA50 を使用するためには、順次出力データ・セットを定義する DD ステートメントを、DFSERA10 入力ストリームに挿入します。この DD ステートメントのデフォルト DD 名は TRCPUNCH です。このステートメントでは、BLKSIZE=80 を指定する必要があります。トレース項目シーケンス番号の最初の 3 バイトが PST 番号であるため、出力を幾つかの BMP アプリケーションと区別できます。

## ユーティリティー制御ステートメント

次の図は、DL/I 呼び出しイメージ・キャプチャー・データをフォーマット設定 (DL/I テスト・プログラム DFSDDL10 への入力として受け入れ可能な形式に) するための制御ステートメントを示しています。

```
Column 1      Column 10          72
OPTION        PRINT OFFSET=5,VALUE=5F,COND=M
OPTION        PRINT OFFSET=25,FLDTYP=C,VALUE=psbname,      X
                                FLDLEN=8,COND=E,EXITR=DFSERA50,DDNAME=OUTDDN
END
/*
//
```

DDNAME= パラメーターを使用して、DFSERA50 が使用する DD ステートメントの名前を指定します。OUTDDN DD ステートメントに定義するデータ・セットが、デフォルト TRCPUNCH DD ステートメントの代わりに使用されます。この例では、DD ステートメントは次のようになります。

```
//OUTDDN DD ...,DCB=(BLKSIZE=80),...
```

## IMS トレース・テーブル・レコード・フォーマット設定・印刷モジュール (DFSERA60)

IMS トレース・テーブル・レコード・フォーマット設定・印刷モジュール (DFSERA60) は、ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) からタイプ X'67FA' のログ・レコードを受け取り、それらのレコードを SYSPRINT データ・セット上でフォーマット設定します。DFSVSAMP データ・セットまたは DFSVSMnn PROCLIB メンバーに OPTION ステートメントを使用して、トレース・テーブルがログに書き込まれるよう指定すると、これらのログ・レコードが作成されます。

DFSERA60 は DFSERA10 の実行時にロードされるので、LINKLIB にあるか、あるいは JOBLIB または STEPLIB データ・セットになければなりません。

### ユーティリティ制御ステートメント

以下に、DFSERA60 を呼び出すために必要な制御ステートメントを示します。

```
Column 1      Column 10                                     72
CONTROL        CNTL
OPTION         PRINT OFFSET=5,FLDLEN=2,VALUE=67FA,      X
               COND=E,EXITR=DFSERA60
END
/*
//
```

## 拡張選択モジュール (DFSERA70)

拡張選択モジュール (DFSERA70) は、圧縮された IMS ログから拡張ログ・レコードを生成するために使用します。また、レコード自体の中に含まれているデータ、例えば、時刻、日付、または ID フィールドの内容などに基づいて、「5X」(DL/I 5X および高速機能 5950) ログ・レコードを選択およびフォーマット設定するためにも使用します。これらのレコードは、PARM TOKEN= の説明にリストされているすべてのログ・レコード・タイプと一緒にフォーマット設定され、いくつかのオプションのログ・フィールドを識別および強調するためにログ出力のフォーマットを変更します。

ルーチンの場合は、ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) の OPTION ステートメントの PARM= パラメーターのサブパラメーターとしてルーチンの検索条件を指定します。PARM= に指定可能なサブパラメーターは次のとおりです。

#### XFMT=

X'50' ログ・レコード・フォーマットを拡張して、特定のデータ入力項目の取得可能性を高めます。

Y データを別個の行に置き、データ入力項目の識別子を追加することで、特定のタイプの処理のログ・データを強調表示します。これは、データ共用、XRF バッファおよびロック・トラッキング、スペース管理、キー、バックアウト (取り消し)、およびリカバリー (再実行) の処理タイプを記述するログ・データに適用されます。処理タイプがこれらと関連しない場合、データ・セクションは省略されます。

これらのデータ・セクションは、レコードの生のログ・データの後に追加されます。各セクションは、ID とその後に 16 進ログ・データ、または文字ログ・データ、あるいはその両方を含みます。これらは次の項目を含んでいます (ここで、X は 16 進ログ・データを表し、C は文字ログ・データを表します)。

#### データ共用

```
DSHRDSSN XXXXXXXX DSHRLSN XXXXXXXXXXXX DSHRUSID  
XXXXXXXX RACF-UID CCCCCC XXXXXXXXXXXXXXXX
```

#### XRF バッファおよびロック・トラッキング

```
TRAKPLSZ XXXX TRAKBUFN XXXX TRAKHASH XXXXXXXX  
TRAKLOCK XXXXXXXX TRAKFLGS XX XX
```

#### スペース管理

```
SMGTFLGS XX XX SMGTROFF XXXX SMGTRLEN XXXX
```

#### キー

```
KSDS データベース・アクションを記述する文字ストリング  
LENGTH XXXX  
16 進数と文字の混合データの 1 つ以上の行
```

#### 取り消し

```
UNDO データベース・アクションを記述する文字ストリング  
LENGTH XXXX OFFSET XXXX  
16 進数と文字の混合データの 1 つ以上の行
```

#### 再実行

```
REDO データベース・アクションを記述する文字ストリング  
LENGTH XXXX OFFSET XXXX  
16 進数と文字の混合データの 1 つ以上の行
```

**N** データ共用、バッファおよびロック・トラッキング、スペース管理、キー、バックアウト、またはリカバリーのログ・データを強調表示しません。データは、レコードの生データの一部としてフォーマット設定されます。

N はデフォルトです。

#### **PST=pst\_number**

PST 番号のレコードを選択します。

#### **SYSID=system\_id**

リカバリー・トークンのシステム ID 部分のレコードを選択します。

#### **TOKEN=token**

リカバリー・トークンの 16 進数のトークン部分のレコードを選択します。

X'07'、 X'08'、 X'0A'、 X'13'、 X'27'、 X'28'、 X'31'、 X'32'、 X'35'、 X'37'、  
X'38'、  
X'39'、 X'3D'、 X'41'、 X'4C'、 X'50'、 X'56'、 X'59'、 X'5901'、 X'5903'、  
X'5937'、 および X'5938' のレコード・タイプを選択できます。

#### **PSB=psb\_name**

PSB 名のレコードを選択します。

#### **DBD=dbd\_name**

DBD 名のレコードを選択します。

**RBA=rba\_value**

RBA (lrecl) のレコードを選択します。

**BLOCK=block\_rba**

RBA (ブロック) のレコードを選択します。

**USERID=userid**

ユーザー ID のレコードを選択します。

**KEY=ksds\_key**

キーのレコードを選択します。

**OFFSET=offset**

バッファ内のデータの指定したオフセットを更新するレコードを選択します。

**UNDO=undo\_data**

指定した文字ストリングと一致するバックアウト・データのレコードを選択します。文字ストリングの最大長は 255 文字です。

**REDO=redo\_data**

指定した文字ストリングと一致するリカバリー・データを持つレコードを選択します。文字ストリングの最大長は 255 文字です。

**DATA=log\_data**

一致するデータ (圧縮データを含めて) が入っているレコードを選択します (すべてのログ・レコードを検索します)。文字ストリングの最大長は 255 文字です。

各サブパラメーターは英大文字でなければならず、ブランクを含みません。サブパラメーター・データは、文字または 10 進数でなければなりません。16 進データの前に X を付け、そのデータを単一引用符で囲む必要があります (例、X'0123')。

選択されたレコードは、テープまたは DASD に書き込むことができます。

複数のサブパラメーターを指定した場合は、すべての条件が満たされなければレコードは選択されません。一部の条件が満たされている場合には、複数のルーチンを使用してレコードを選択してください。

ログ印刷フォーマット設定は、DFSERA30 によって行われます。形式は、DFSERA30 が指定されたルーチンである場合のようになります。DFSERA70 をロードするときには、DFSERA30 が使用可能でなければなりません。

このルーチンは、認識できない文字または無効なパラメーター指定を無視します。

**例**

これらの例は、DFSERA70 モジュールを使用して、通常データまたは拡張データを持つレコードを印刷する方法を示しています。

以下の例に示すオプションは、X'50' または X'5950' データベース・レコードを含んだすべてのレコードを印刷し、データを X'5050' レコード内に展開します。

```
OPTION PRINT EXITR=DFSERA70
```

次の例に示すオプションは、拡張データを持つ X'50' データベース・レコードだけを印刷します。

```
OPTION PRINT O=5,V=50,EXITR=DFSERA70
```

以下の例に示すオプションは、拡張データと 67 の診断レコードを持つ X'50' データベース・レコードだけを印刷します。

```
OPTION PRINT O=5,V=67,EXITR=DFSERA30
OPTION PRINT O=5,V=50,EXITR=DFSERA70
```

以下に、APPLPSB という名前の PSB を使用して、PST 番号 X'A' の X'50' または X'5950' データベース・レコードを含むすべてのレコードを、通常形式で印刷するためのオプションを示します。

```
OPTION PRINT EXITR=DFSERA70,PARM=(XFMT=N,PST=X'A',PSB=APPLPSB)
```

以下に、X'2000' およびオフセット X'200' の RBA にある、X'50' または X'5950' データベース・レコードを含むすべてのレコードを通常形式で印刷するためのオプションを示します。

```
OPTION PRINT EXITR=DFSERA70,PARM=(XFMT=N,RBA=X'2000',OFFSET=X'200')
```

以下の例に示すオプションは、文字ストリング「aaaa」を含んでいるすべてのレコードを拡張形式で印刷します。

```
OPTION PRINT EXITR=DFSERA70,PARM=(XFMT=Y,DATA=aaaa)
```

以下の例に示すオプションは、トークン X'0001F8FF00000000' を持つすべてのタイプのログ・レコードを選択し、拡張形式でそれらのレコードを印刷します。

```
OPTION PRINT EXITR=DFSERA70,PARM=(XFMT=Y,TOKEN=X'0001F8FF00000000')
```

以下に、トークン X'0001F8E400000001' を持つ X'0A' ログ・レコードを選択し、通常形式でレコードを印刷するためのオプションを示します。

```
OPTION PRINT O=5,V=0A,T=X,EXITR=DFSERA70,PARM=(XFMT=N,TOKEN=X'0001F8E400000001')
```

## OM 監査証跡フォーマット設定・印刷モジュール (CSLULALE)

MVS™ システム・ロガーから OM ログ・レコードを印刷するには、IMS ファイル選択・フォーマット設定印刷ユーティリティ (DFSERA10) とモジュール CSLULALE を使用します。

### JCL 仕様

次の例では、MVS システム・ログからログ・レコードを印刷するために必要な JCL を示します。この JCL は MVS ロガーにデフォルトのログ・ストリーム・サブシステム出口ルーチン IXGSEXIT を起動させ、ログ・レコードをコピーさせます。出口ルーチンは、ログ・レコード 1 つにつき最大 32760 バイトのデータを返します。これは、OM がそれより大きいログ・レコードをサポートしている場合でも変わりません。必要であれば、別の出口ルーチンの名前を指定できます。

以下の JCL を使用して OM ログ・レコードを印刷します。

```
//CSLERA10 JOB  MSGLEVEL=1,MSGCLASS=A,CLASS=K
//STEP1 EXEC  PGM=DFSERA10
//STEPLIB DD   DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT DD  SYSOUT=A
//SYSUT1 DD   DSN=SYSLOG.OM.AUDIT.TRAIL.LOG,
//             SUBSYS=(LOGR,IXGSEXIT),
//             DCB=(BLKSIZE=32760)
//SYSIN DD *
```



```
CONTROL    CNTL H=EOF
OPTION     PRINT EXITR=CSLULALE,PARM=(F=option)
END
//
```

**DD** ステートメント

### STEPLIB

DSN= は IMS.SDFSRESL を指します。ここには、IMS ファイル選択およびフォーマット設定印刷ユーティリティ、DFSER10 が入っています。

### SYSPRINT

フォーマット設定済み印刷レコードおよび制御メッセージを入れるための出力データ・セットを記述します。これは、通常、SYSOUT=A と定義します。

このデータ・セットには、RECFM=FBA の DCB パラメーターが指定されます。LRECL およびブロック・サイズは、SYSPRINT DD ステートメントで指定でき、これは 133 の倍数でなければなりません。133 から 32760 までの任意の LRECL 値を指定できます。ブロック・サイズおよび LRECL のデフォルトは 133 です。

### SYSUT1

DSN= は、OM ログ・ストリーム名を指します。OM 始動パラメーター (CSLOIxxx PROCLIB メンバーの AUDITLOG= パラメーター) で指定した名前と同じ名前を使用します。

指定した時刻範囲へのログ・データの制限

SUBSYS ステートメントで FROM および TO パラメーターを使用して、印刷するログ・レコードを特定の時間間隔内のレコードだけに制限できます。例えば、次の DD カードを参照してください。

```
//SYSUT1 DD DSN=SYSLOG.OM.AUDIT.TRAIL.LOG,
//   SUBSYS=(LOGR,IXGSEXIT,
//   'FROM=(2010/042,11:00:00),TO=(2010/042,12:00:00)'),
//   DCB=(BLKSIZE=32760)
```

DD カードは DFSERA10 プログラムに、2010 年の通算日 42 の 11:00 から 12:00 までのログ・レコードだけを渡します。指定する日時は GMT (グリニッジ標準時) です。時刻値の秒フィールドはオプションです。

ローカルの日時を使用するには、ステートメントに LOCAL キーワードを追加します。以下の DD カードを参照してください。

```
//SYSUT1 DD DSN=SYSLOG.OM.AUDIT.TRAIL.LOG,
//   SUBSYS=(LOGR,IXGSEXIT,
//   'FROM=(2010/042,11:00:00),TO=(2010/042,12:00:00),LOCAL'),
//   DCB=(BLKSIZE=32760)
```

### ユーティリティ制御ステートメント

#### PARM=

パラメーターを CSLULALE 出口ルーチンに渡すときに使用します。

**F=** パラメーター名を指定します。有効なオプションは、次のとおりです。

#### WRAP

個々の行を折り返すことを指定します。WRAP はデフォルトです。

**BYCOL**


行を列によってグループ化することを指定します。

**BYRSC**

行をリソースによってグループ化することを指定します。

**H=** 印刷するログ・レコードの数を指定します。H=EOF はすべてのログ・レコードを印刷します。

関連資料:

 [z/OS: IXGSEXIT - ログ・ストリーム・サブシステム出口](#)

---

## 第 16 章 IMS モニター報告書印刷ユーティリティー (DFSUTR20)

IMS モニター (DFSMNTR0) で収集したデータを用いて、要約報告書とデータの分布表示を印刷するため、IMS モニター報告書印刷ユーティリティー (DFSUTR20) を使用します。

報告書の形式と、報告書に記載される情報の種類は、IMS DB モニター印刷ユーティリティー (DFSUTR30) で印刷されるものと同じであるか、類似しています。

サブセクション:

- 『制約事項』
- 『前提条件』
- 528 ページの『要件』
- 528 ページの『推奨事項』
- 528 ページの『入力と出力』
- 528 ページの『JCL 仕様』

### 制約事項

IMS モニター報告書印刷ユーティリティーには、下記の制限が適用されます。

- 特定の報告書内に通常見られるタイプの情報がモニターによって収集されなかった場合、その報告書内またはその報告書のセクション内に、通常なら含まれる情報が生成されません。例えば、チェックポイントが発生しなかった場合は、チェックポイントの見出しのみが印刷されます。
- モニター・トレース間隔の開始時と終了時にデータが収集される報告書では、その報告書は、それらの間隔で収集されるデータ、およびそれらの相違を示します。これらの報告書のデータは両方の間隔で必要なため、モニター・トレースの前に IMS 制御領域が終了する場合、これらの報告書は生成されません。
- 領域要約、領域待機、実行プロファイル、または呼び出し要約 (DB) の各報告書の使用を計画している場合は、合計 999999999 回を超える DL/I 呼び出しの間、オンのままにしておいてはなりません。999999999 回の DL/I 呼び出しの後に、これらの報告書のさまざまな合計フィールドで切り捨てが行われます。

IMS モニター報告書印刷ユーティリティー (DFSUTR30) で印刷される報告書で使用される用語の多くは、IMS モニター報告書印刷ユーティリティー (DFSUTR20) で印刷される報告書にも現れます。

### 前提条件

現在、DFSUTR20 ユーティリティーについて文書化されている前提条件はありません。

## 要件

現在、DFSUTR20 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSUTR20 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

モニター報告書印刷ユーティリティはバッチ・プログラムとして実行され、DASD またはテープ上の順次データ・セットが入力として使用されます。このデータ・セットの内容は、IMS オンライン実行時に、/TRACE SET ON MONITOR コマンドへの応答で、IMS モニター・モジュール (DFSMNTR0) によって作成されます。

## JCL 仕様

### EXEC ステートメント

プログラム名を指定します。このステートメントは、次のような形式でなければなりません。

```
// EXEC PGM=DFSUTR20,REGION=4096K
```

### JOB ステートメント

ジョブを開始します。

### DD ステートメント

#### STEPLIB DD

IMS.SDFSRESL を指します。ここには、IMS 中核と必要なアクション・モジュールが入っています。

#### //SYSPRINT DD

報告書と制御メッセージが入れられる出力データ・セットを指定します。通常、SYSOUT=A とコーディングします。このデータ・セットの DCB パラメーターは、RECFM=FBA および LRECL=133 です。BLKSIZE は、SYSPRINT DD ステートメントで指定でき、133 の倍数でなければなりません。BLKSIZE を指定しなければ、デフォルトの 133 が使用されます。

#### //SYSUT1 DD

分析する入力データ・セットを指定します。これは、モニター・モジュール DFSMNTR0 によって作成される、ラベル付き順次データ・セットです。(DD 名と DS 名は、IMS プロシージャ内の IMSMON です。)

#### //ANALYSIS DD

分析制御データ・セットを指定します。このファイルは、カード・イメージ形式でなければなりません。

分析制御データ・セット

分析制御データ・セットは、印刷する報告書を決定し、分布報告書の分布再定義が行えるようにします。

- 呼び出し要約報告書のみを印刷する場合は、この実行用の分析制御データ・セット内に ONLY DLI ステートメントを組み込んでください。ステートメントは、カード・イメージで 1 桁目から始まります。
- 呼び出し要約報告書を生成するときは、この実行用の分析制御データ・セットに DLI ステートメントを組み込んでください。このステートメントを組み込まない場合、デフォルト・オプションが使用されます。つまり、呼び出し要約報告書を除くすべての報告書が印刷されます。ステートメントは、カード・イメージで 1 桁目から始まります。
- オプションの分布付録報告書を作成するときは、分析制御データ・セットの任意の場所に DIS ステートメントを組み込んでください。このステートメントを組み込まなければ、要約報告書のみが印刷されます。ステートメントは、カード・イメージで 1 桁目から始まります。

上記のオプションのいずれも選択されなかった場合は、呼び出し要約報告書および配布付録報告書を除いて、すべての報告書が印刷されます。

#### 分布の再定義の指定

ユーザーが分布を再定義を指定するときの一般形式は、次のとおりです。

**Dn** n1,n2...

**Dn** 分布 ID (ID) で、1 桁目から始めます。

**n1** から **n9**

それぞれ 8 桁以下で、0 から 99999999 の正の数を指定します。

各再定義では、必要に応じて複数のステートメントを使用できます。継続ステートメントのフォーマットは、以下のような z/OS の規則に従います。

- 最初のステートメントの最後の値には、コンマと少なくとも 1 つの空白を続けなければなりません。
- 継続ステートメントの最初の値は、2 桁目以降で、10 桁目以前の位置から始めなければなりません。
- コメントの前に最低 1 個空白を付ければ、コメントを入れることができます。

領域の実行経過時間の分布 ID が D1 で、デフォルトの定義が次のようになっているものとします。

```
0 1 2 3 30 300 3000 30000 3000000 30000000 INF
```

これを、次のように再定義することができます。

```
0 1 2 5 30 40 50 60 3000000 30000000 INF
```


この再定義を行うには、分析制御データ・セット内に以下のレコードを入れます。

```
D1 1,2,5,30,40,50,60,3000000,30000000
```


数字は定位置パラメーターになっているため、次のように指定しても同じ再定義を行えます。

```
D1 ,,5,,40,50,60
```

関連概念:

 [IMS モニター報告書 \(システム管理\)](#)

関連資料:

 [データベース・モニター報告印刷ユーティリティ \(DFSUTR30\) \(データベース・ユーティリティ\)](#)

---

## DFSUTR20 ユーティリティの例

下記の JCL では、すべての報告書 (通し番号が IMSDA1 のテープからの呼び出し要約報告書を含む) を作成します。

```
//TRACE    JOB
//*
//          EXEC PGM=DFSUTR20,REGION=512K
//STEPLIB  DD DSN=IMS.&SYS2..SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSUT1   DD DSN=IMSMON,DISP=(OLD,KEEP),
//          UNIT=TAPE,VOL=SER=IMSDA1
//ANALYSIS DD *
DLI CALL REPORT
DISTRIBUTION
/*
```

次の例は、JCL を変更する場合に D30 と D2 の分布がどのように変更されるかを示しています。

```
          .
          .
          .
//ANALYSIS DD *
DLI CALL REPORT
DISTRIBUTION
D30 8000,24000,50000,75000
D2 1000,2000,3000,4000,5000,6000,7000,8000,9000
/*
```

---

## 第 17 章 ログ・トランザクション分析ユーティリティー (DFSILTA0)

ログ・トランザクション分析ユーティリティー (DFSILTA0) は、IMS ログ・データ・セット内のレコードに基づいて、IMS トランザクションの個々のオカレンスに関する情報を収集するときに使用します。

収集される情報には次のものがあります。

- トランザクション識別
- ソース
- トランザクション・プログラム名
- 従属領域
- 優先順位
- トランザクションのクラス

取り消されたメッセージは使用されません。

さらに、DFSILTA0 は以下のものを累積します。

- 各トランザクションが受け取られた時刻
- メッセージ単独読み取り (GU) 呼び出しの時刻
- トランザクション処理が終了した時刻
- 出力メッセージが出力キューに入れられた時刻
- 出力メッセージが端末への送信を開始された時刻

これらの時刻から、DFSILTA0 は以下のものを計算します。

- 全体の応答時間
- 入力キューに入っていた時間
- 処理時間
- 出力キューに入っていた時間

このような情報を利用して、システムのボトルネックを検出したり、トランザクション・クラスが正しく割り当てられているかどうかを評価することができます。IMS ログ・データのさらに小さな部分に対して統計分析ユーティリティーを実行している場合は、DFSILTA0 は、ユーザーの指定に合わせた新しいログを提供できます。DFSILTA0 は、IMS システム定義のときに IMS.SDFSRESL の中に置かれます。

サブセクション:

- 532 ページの『制約事項』
- 532 ページの『前提条件』
- 532 ページの『要件』
- 532 ページの『推奨事項』

- 『入力と出力』
- 533 ページの『JCL 仕様』

## 制約事項

ログ・トランザクション分析ユーティリティには、以下の制約があります。

- バッチ領域からのログ・データ・セットは使用されません。
- 取り消されたメッセージは使用されません。
- DFSILTA0 は、指定した時間またはチェックポイントの範囲に入るトランザクションごとに、GETMAIN ストレージ・プール内にキュー項目を作成します。これらのキュー項目は、ログ・トランザクション分析報告書の項目を完成するために必要なログ・レコードがすべてログで検出されるまでは、解放されたり、再利用されたりすることはありません。
- たくさんのトランザクションがエンキューされているときに、何らかの理由でこれらが処理されないと、ストレージの使用量とプロセッサ時間が増加します。
- 共通キュー・サーバー (CQS) のログ・レコードは、IMS ログ・レコードとは形式が異なるため、ログ・トランザクション分析ユーティリティは、CQS ログを入力として使用することはできません。
- このユーティリティは、このユーティリティのリリース・レベルと同じリリースの IMS で作成された入力ログ・データ・セットのみに機能します。

## 前提条件

現在、DFSILTA0 ユーティリティについて文書化されている前提条件はありません。

## 要件

現在、DFSILTA0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSILTA0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

DFSILTA0 への入力は次の 3 種類があります。

- IMS ログ・データ。これは必須です。
- 報告書表題ステートメント。これは、オプション表題データ・セットの記述情報を与えます。
- パラメーター。オプションの入力パラメーター ST=。これはログ・データ・セットのどの部分でトランザクションを探すかを指定します。

DFSILTA0 ユーティリティの出力を生成するすべての DD ステートメントは、オプションです。DD ステートメントを何も指定しなかった場合、ユーティリティは CPU 時間を浪費するだけで、何も出力を返しません。いずれかの DD ステートメントを指定した場合、DFSILTA0 は以下の出力を生成できます。

- 処理された期間の IMS ログ・レコードが入っている新規データ・セット



- 入力順の詳細報告書 (PRINTER DD ステートメントを指定した場合)
- 順番に並べた報告書を生成するためにソートできる報告書 (REPORT DD ステートメントを指定した場合)
- 見出し報告書 (HEADING DD ステートメントを指定した場合)

詳細報告書形式のフィールド名の開始位置と長さは、順番に並べた報告書を作成するためのオプションのソート・ステップで使用されます。

## JCL 仕様

ログ・トランザクション分析ユーティリティーは、標準オペレーティング・システム・ジョブとして実行されます。ユーザーは EXEC ステートメントと DD ステートメントを定義する必要があります。

### EXEC ステートメント

ログ・トランザクション分析ユーティリティーの DFSILTA0 を実行します。

```
//STEP0 EXEC PGM=DFSILTA0,PARM='ST'=(hhmmss+HHMM,,mm)
```

### ST=

開始時刻と終了時刻を指定します。ST パラメーターを省略すると、デフォルトは、最初に検出されたチェックポイントになります。ST= パラメーターの形式は次のとおりです。

```
ST=ALL
(hhmmss[+|-]HHMM),
c,mm,e)
```

ST= パラメーターには、ALL パラメーター以外に 4 つの定位置パラメーターがあることに注意してください。ALL パラメーターを除いて、これらのパラメーターは括弧で囲む必要があります。

### ALL

ログ・データ・セット全体を表します。

### hhmmss

時間、分、および秒を指定します。この時刻またはこの時刻以降に起きた最初のチェックポイントの後に発生したトランザクションのみが処理されます。デフォルトでは、この時刻から 10 分間処理が行われます。

ヒント: このパラメーターは常に、入力ログ上の最初のチェックポイントより後の時刻を参照するものと想定されます。ログの最初のチェックポイントで始まるトランザクションを処理する場合は、このパラメーターには値を指定しないでください。

### {+|-}HHMM

地方時を協定世界時 (UTC) に変換するときに使用する時間帯オフセットを指定します。

+ または -

オフセットの符号を指定します。hh および mm もブランクの場合のみブランクにできます。時間帯を指定する必要があるのは、入力日の

UTC とのオフセットが現在のオフセットと異なる場合のみです。この例として、夏時間への変更のためにオフセットが変更される場合があります。

**HH** オフセットの時間を、0 から 14 の数値またはブランク (mm もブランクの場合のみ) を指定します。

**MM** 00、15、30、45、またはブランクで、オフセットの分を指定します。

オフセットとして +|-0000 が指定される場合は、開始時間は UTC です。オフセットを指定しなかった場合は、z/OS オフセットからオフセットが取得されます。

**C** トランザクションの選択を停止する前に処理するチェックポイントの個数を指定します。C は 1 から 9 の数値です。

**MM** トランザクションを選択する分数を指定します。MM は 0 から 99 の数値です。

**E** 指定した開始時刻からデータ・セットの終わりまでを指定します。E はデフォルトです。

ログ・トランザクション分析ユーティリティーは、チェックポイントとチェックポイントの間のレコードをスキャンします。中間ログ・データ・セット上の最初のチェックポイントより前のレコードは、前のログのチェックポイントの参照によって分析されるだけです。

## **DD** ステートメント

### **STEPLIB DD**

IMS.SDFSRESL を指します。ここには、IMS 中核と必要なユーティリティー・モジュールが入っています。

```
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
```

### **HEADING DD**

オプションの見出し出力データ・セットを記述します。

```
//HEADING DD SYSOUT=A
```

### **PRINTER DD**

オプションの印刷報告書出力データ・セットを記述します。

```
//PRINTER DD SYSOUT=A
```

### **REPORT DD**

オプションの報告書出力データ・セットを記述します。このデータ・セットは、ソート・ステップに渡すことができます。報告書の項目見出しおよびチェックポイント・レコードは、このデータ・セットには含まれません。

```
//REPORT DD DSN=&&REPORT,DISP=(,PASS),UNIT=SYSDA,  
// SPACE=(CYL,(1,1))
```

### **LOGINxxx DD**

入力ログ・データ・セットを記述します。最後の 3 文字はオプションです。5 文字を超えて指定した場合、最後の文字が報告書の中でシステム ID として使用されます。

各 IMS システムからのログ・データは、単一の DD ステートメントへ割り当てられている必要があります。IMS から複数のログ・データ・セットを使用する場合、それらのデータ・セットは連続していなければならない、作成された順序でそれらを連結する必要があります。

```
//LOGIN DD DSN=IMS.LOG,DISP=OLD,VOL=SER=XXXXXX,  
// UNIT=YYYY
```

#### LOGOUT DD

オプションのログ・データ・セットを記述します。このログ・データ・セットは、統計分析ユーティリティへの入力として使用できます。

LOGOUT データ・セットの内容は、指定した間隔内で LOGIN に入っているものと同じですが、LOGIN の先頭位置のタイプ 6 のレコードは再コピーされます。

```
//LOGOUT DD DSN=IMS.LOGOUT,DISP=(,PASS),  
// VOL=SER=XXXXXX,UNIT=TAPE,DCB=(RECFM=VB,  
// LRECL=6004,BLKSIZE=6008)
```

#### TITLE DD

オプションの表題データ・セットを記述します。これによって、プリンター出力データ・セットの各ページに記述情報を含めることができます。

```
//TITLE DD *  
*** Descriptive information
```

ソート (SORT) ステップはオプションです。これは、順番に並べられた報告書の作成に使用します。

#### EXEC

ソート・プログラムを実行します。

```
//STEP1 EXEC PGM=SORT
```

#### SYSOUT DD

ソート用のメッセージ出力データ・セットを記述します。

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

#### SORTIN DD

ソート・プログラムの入力データ・セットを記述します。これは、REPORT DD ステートメントによって記述されたデータ・セットです。

```
//SORTIN DD DSN=&&REPORT,DISP=(OLD,DELETE)
```

#### SORTOUT DD

ソート・プログラムの出力データ・セットを記述します。これは、順番に並べられた報告書の印刷に使用します。

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

#### SORTWK01-12|DD

ソート・プログラムの作業データ・セットを記述します。最低 3 つのデータ・セットを使用する必要があります。これらは、テープでもディスクでも構いません。ディスクの場合の形式は、次のとおりです。

```
//SORTWKnn DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
```

#### SYSIN DD


ソート・プログラムの制御データ・セットを記述します。入力ストリーム内の制御データ・セットの形式は、次のとおりです。

```
//SYSIN DD *
```

例：以下は、領域内のメッセージ単独読み取り (GU) スケジュール時刻順に報告書を作成する、SORT 制御ステートメントの例です。

```
SORT FIELDS=(67,7,CH,A,55,2,CH,A),SIZE=E500
```

関連概念:

 統計分析報告書、ログ・トランザクション報告書、およびログ・レコード分析 (システム管理)

関連資料:

545 ページの『第 19 章 統計分析ユーティリティ (DFSISTS0)』

---

## 第 18 章 オフライン・ダンプ・フォーマッター・ユーティリティー — (DFSOFMD0)

オフライン・ダンプ・フォーマッター・ユーティリティー (DFSOFMD0) は、障害の影響を受けず、しかもダンプ処理に従属しないダンプ内の内部 IMS 制御ブロックをフォーマット設定するときに使用します。

このユーティリティーにより、特定の問題の分析に必要なデータ域のみを印刷してフォーマット設定するために、ダンプを調整することができます。オフライン・ダンプ・フォーマッター・ユーティリティーは、次の目的で使用します。

- オフライン・ダンプ・フォーマット設定に必要な環境の確立
- ダンプ・フォーマット設定制御ステートメントの読み取りと検査
- ダンプ・フォーマット設定モジュールの再配置またはロード
- オフライン・ダンプのフォーマット設定処理の指示

オフライン・ダンプ・フォーマッター・ユーティリティーは、対話式問題制御システム (IPCS) からの verb 出口として呼び出されます。

オフライン・ダンプ・フォーマッター・ユーティリティーのモジュールは、ダンプのフォーマット設定に使用されるモジュールが、ダンプされる IMS 制御ブロックのレベルと必ず一致するように、ダンプされるストレージ内に入れられます。これらのモジュールをダンプ・ストレージから再配置することもできますし、プログラム・ライブラリーから新しいコピーをロードすることもできます。

オフライン・ダンプ・フォーマッター・ユーティリティーは、IMS の複数のリリース・レベルがある場合でも、あるいは IMS のサポートされているどのバージョンをご使用であっても、使用できます。オフライン・ダンプ・フォーマッター・ユーティリティーのロード・モジュールは、別名と関連付けられており、この別名により異なるリリースからの IMS.SDFSRESL を IPCS TASKLIB に連結することができます。別名は次のとおりです。

別名   ロード・モジュール

### DFSOF111

DFSOFMD0

### DFSAB111

DFSABND0

IPCS TASKLIB 連結には、複数の実行ライブラリーを含めることができます。

サブセクション:

- 538 ページの『制約事項』
- 538 ページの『前提条件』
- 538 ページの『要件』
- 539 ページの『推奨事項』

- 539 ページの『入力と出力』

## 制約事項

オフライン・ダンプ・フォーマッター・ユーティリティには、以下の制約があります。

- 事前再編成ユーティリティおよびイメージ・コピー・ユーティリティなどのような、IMS オンライン定様式ダンプを現在作成中ではないバッチ領域に、オフライン・ダンプ・フォーマッター・ユーティリティを使用することはできません。これらには、IMS ダンプのフォーマット設定に必要な IMS 制御ブロックが含まれていないためです。

## 前提条件

現在、DFSOFMD0 ユーティリティについて文書化されている前提条件はありません。

## 要件

オフライン・ダンプ・フォーマッター・ユーティリティには、以下の制約があります。

- このユーティリティを実行するマシンは、IMS を実行するライセンスを持っていないければなりません。
- オフライン・ダンプ・フォーマッター・ユーティリティは、GETMAIN、ESTAE、および LOAD について z/OS サービスに依存しているため、IMS 制御ブロックの生成時に条件付きでアセンブルされます。DFSOFMD0 モジュールが IPCS によって LOAD SVC でロードされる場合は、そのモジュールは STEPLIB データ・セットまたはリンク・リスト・ライブラリー内になければなりません。
- DFSOFMD0 モジュールは、フォーマット設定する IMS システムと同じリリース・レベルでなければなりません。このモジュールは、フォーマット設定の対象である z/OS と同じレベルの z/OS 上でアセンブルする必要があります。この条件は、以前のリリースからの IMS.SDFSRESL を連結する場合でも適用されます。
- このユーティリティを実行するために使用する IPCS のバージョンは、ダンプされた z/OS システムと互換性がなければなりません。
- SYS1.DUMPxx データ・セットは、IMS SDUMP オプションを使用するシステムの IMS 制御領域、DL/I、DBRC、および IRLM アドレス・スペースの完全なダンプが入る十分な大きさでなければなりません。
- 高速機能ダンプをフォーマット設定するためには、高速機能で生成された IMS システムからのフォーマット設定モジュールを使用する必要があります。
- IMS 共用メッセージ・キューまたは共用 EMH キューを使用している場合は、ご使用の SYS1.DUMPxx データ・セットには、上記のアドレス・スペースに加えて、CQS アドレス・スペースのダンプも収められるだけの十分な大きさが必要です。共通サービス層 (CSL) を使用している場合は、ご使用の SYS1.DUMPxx データ・セットには、上記のアドレス・スペースに加えて、SCI アドレス・スペースのダンプも収められるだけの十分な大きさが必要です。

## 推奨事項

現在、DFSOFMD0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

このユーティリティには、下記の入力が必要です。

- 受け入れ可能な、以下のような機械可読ダンプ。
  - SDUMP
  - SYSMDUMP
  - 独立型ダンプ
  - z/OS DUMP コマンドによって要求されるダンプ
  - IMS システム・アドレス・スペースのその他のマシン可読ダンプ

ダンプは、キー 0 およびキー 7 CSA、CVT、SQA、および最低 1 つの CTL または DL/I SAS のアドレス・スペースを含む必要があります。CSA は、バッチ環境では不要です。

- IMS ダンプ形式制御データ・セット、または IPCS VERBX 制御ステートメントで指定した FMTIMS (オプション)
- IPCS 用の正しい VERBX 制御ステートメントの実行

このユーティリティの出力は、IMS ダンプの指定セクションの定様式ダンプです。

以前の IMS リリースからの実行ライブラリーを用いてオフライン・ダンプ・フォーマッターを使用する場合、CSA がバッチ SYSMDUMP と一緒に含まれていないと、フォーマッター・ダイアログ初期設定警告が出されます。オフライン・ダンプ・フォーマッターは、連結されるプログラム・ライブラリーのリリース・レベルを判別することはできませんが、ライブラリーが正しく連結されていると想定して続行されます。

---

## DFSOFMD0 ユーティリティの実行

IMS ダンプ・フォーマッターを使用すると、DFSFRMAT ファイルの複雑な編集を行わなくても、メニュー方式でオフライン・ダンプ・フォーマッター・ユーティリティを実行できます。

IPCS は、画面上のメニューを使用して、IMS ダンプ・フォーマッターを実行します。これらのメニューを使用すると、ダンプに入れられる情報を指定できます。IMS ダンプ・フォーマッターは、オフライン・ダンプ・フォーマッター・ユーティリティを呼び出して、必要なフォーマット設定タスクを実行します。出力は、ユーザーが端末で読むことができる形式で戻されます。

### IMS オンライン環境でのユーティリティの実行

IMS DB/DC、DCCTL、または DBCTL 環境でオフライン・ダンプ・フォーマッター・ユーティリティを使用するときは、IMS 開始パラメーター・オプションの FMTO=D を指定します。

SYSMDUMP DD ステートメントを使用することもできます。

## IMS バッチ環境でのユーティリティの実行

DBCTL、DB/DC、または DCCTL バッチ環境で IMS バッチ・ジョブ・ダンプをオフラインでフォーマット設定するために、z/OS SYSMDUMP を要求できます。z/OS は、IMS オフライン・ダンプ・フォーマッター・ユーティリティを使用してオフラインでフォーマット設定できるダンプを作成します。このユーティリティを使用する前に、IMS バッチ JCL プロシージャ内の SYSUDUMP または SYSABEND DD ステートメントを除去し、SYSMDUMP DD ステートメントを挿入する必要があります。

SYSMDUMP データ・セットが小さすぎるか、選択不可、または使用不可の場合、オペレーティング・システムは、バッチ・ジョブの使用可能なダンプを作成できないことがあります。

## IPCS の実行

IPCS のもとでオフライン・ダンプ・フォーマッター・ユーティリティを使用するときは、IMS ユーザー制御ステートメントを用意しなければなりません。

IMS ユーザー制御ステートメントのいくつかの例を以下に示します。

```
VERBX DFSOFMD0 'jjjjjjjj[,R][,D]' verbx_options
```

```
VERBX DFSOFMD0 'jjjjjjjj[,R][,H],FMTIMS(ALL)' verbx_options
```

```
VERBX DFSOF320 'jjjjjjjj,FMTIMS(SCD)' verbx_options
```

```
VERBX DFSOF320 'jjjjjjjj[,R][,N],FMTIMS(AUTO,MIN)' verbx_options
```

```
VERBX IMSDUMP 'jjjjjjjj[,R][,D],FMTIMS(SAVEAREA,DISP)' verbx_options
```

```
VERBX IMSDUMP 'jjjjjjjj[,R][,D]' verbx_options
```

```
VERBX IMSDUMP 'IMSDUMMY,R,FMTIMS(LOG)' verbx_options
```

```
VERBX IMSDUMP 'IMSDUMMY,R,FMTIMS(SAP,2A723C80)' verbx_options
```

```
VERBX IMSDUMP 'IMSDUMMY,R,FIMTIMS(SAVEAREA,SUM)' verbx_options
```

制御ステートメントのパラメーターは次のとおりです。

### jjjjjjjj

IMS CTL、DL/I、または IMS バッチ・アドレス・スペースの、ジョブ名または開始タスク名を表します。

**R** REFRESH を表します。これは、IMS ダンプ・フォーマッター・モジュールを現行プログラム・ライブラリーからロードするよう要求するためのオプション・パラメーターです。無効なダンプ・フォーマッター・ルーチンが存在している場合に R を指定しなければ、その無効なルーチンが現行ライブラリーの代わりにロードされることがあります。

**H** HALFLINE を表します。これは、IMS ダンプ・フォーマッターに画面幅（つまり、80 文字/行）に表示を制限することを要求するオプション・パラメーターです。

**N** NO HEADER を表します。これは、小さなデータ域ダンプをフォーマット設定



する際にヘッダー印刷の量を減らすためのオプション・パラメーターです。フォーマッターは、印刷ヘッダーおよびフッターをスキップし、初期設定を終了しなかったアドレス・スペースまたは脱落 IMS アドレス・スペースを記述するダンプ内容警告メッセージを抑制します。

- D DEBUG を表し、IMS オフライン・フォーマッターに、その ESTAE を作成しないよう要求し、IMS ダンプ・フォーマッターの異常終了のダンプを作成できるようにするオプション・パラメーターです。

#### **FMTIMS(options)**

FMTIMS verb を指定します。FMTIMS verb は、制御ステートメント、または IMS ダンプ形式制御データ・セット記述 (DFSFRMAT DD) のいずれかで指定する必要があります。FMTIMS は、フォーマット設定オプション (IPCS の現行パスでフォーマット設定される IMS ダンプのセクションを記述したもの) のサブセットを許可します。DFSFRMAT DD 記述は、このサブセットを記述します。

#### **verbx\_options**

有効な IPCS VERBX コマンド・オプションを指定します。

ユーザー制御ステートメントの中に FMTIMS を指定しない場合には、DFSFRMAT オプションを指定した IMS ダンプ形式制御ステートメントを指定する必要があります。

以下の例は、IMS ダンプ形式制御データ・セット情報を指定するための TSO ALLOCATE コマンドの例です。

```
ALLOC FI(DFSFRMAT) SHR DA('dump.control.dsname')
```

**DD** ステートメント

#### **INDEX DD**

定様式ダンプの前にダンプ索引を印刷するためのものです。

#### **DFSFRMAT DD**

IMS ダンプ形式制御データ・セットを記述します。このデータ・セットには、IPCS の現行パスでフォーマット設定される IMS ダンプのセクションを指定する制御ステートメントが入っています。このステートメントを指定しないと、フォーマット設定オプションはデフォルトとして SUMMARY が使用されます。

IMS ダンプ形式制御データ・セットは順次データ・セットで、固定または固定ブロック化のレコード・フォーマット (RECFM=F または FB) で定義する必要があります。レコード長は、任意の有効なサイズが可能です。データ・セットは 1 個の FMTIMS verb を含み、フォーマット設定される IMS のセクションを記述するサブセット・オプションがその後続きます。選択するオプションに MIN パラメーターを追加することにより、フォーマット設定されるサブセットの縮小版を要求できます。

AUTO オプションを指定すれば、ダンプ・フォーマッター・オプションの選択を IMS に行わせることができます。AUTO を指定すると、IMS は、障害のある ITASK を調べ、必要なダンプ・フォーマッター出力用に適切なオプション・セットを選択することにより、使用すべきオプションを判別します。

AUTO は、MIN または SUM 修飾子と一緒に指定できます。MIN または SUM を使用すると、その修飾子は AUTO が選択する各オプションに追加されます。

サブセット・オプションの組み合わせおよび指定順序は自由です。以下のサブセット・オプションはそれぞれ単独で指定したり、示されているように修飾子を付けることができますが、引数を追加する必要はありません。



- ALL または (ALL,MIN)
- 「AOI 自動化操作プログラム・インターフェース (Directed Message Manager)」
- AUTO、または (AUTO,MIN)、または (AUTO,SUM)
- CBT
- DB または (DB,MIN)
- DBRC
- DBRM データベース・リカバリー・マネージャー
- DC または (DC,MIN)
- DEDB または (DEDB,MIN)
- DISPATCH または (DISPATCH,MIN)
- EMH または (EMH,MIN)
- IRLM 制御ブロック・フォーマット設定
- LOG または (LOG,MIN)
- LUM
- MSDB または (MSDB,MIN)
- OTMA Open TM Access
- QM または (QM,MIN)
- RESTART
- SAVEAREA、または (SAVEAREA,MIN) または (SAVEAREA,SUM)
- SB または (SB,MIN)
- SCD または (SCD,MIN)
- SDE ストレージ記述子エレメント・ブロックおよびストレージ
- SMBS すべての SMB
- SPST
- SUBS
- SUMMARY または (SUMMARY,MIN)
- SYSTEM または (SYSTEM,MIN)
- UTIL

下記のサブセット・オプションには、以下で示すように、引数を付けるか修飾子を付ける必要があります。

- (CBTE,cbteid)
- (CLB,address)、(CLB,nodename)、(CLB,lterm name)、または (CLB,comm id)
- (DPST,address)、(DPST,number)、または (DPST,name)

- (LLB,link number)
- (LUB,lu name)
- (POOL,poolid) または (POOL,poolid,MIN)
- (SAP,sapaddr) または (SAP,ecbaddr)
- (SYSPST,system pst address) または (SYSPST,system pst name)
- (TRACE,name) または (TRACE,name,MIN)

関連概念:

-  z/OS: MVS 対話式問題管理システム (IPCS) の概要
-  IMS オフライン・ダンプ・フォーマッターを使用した IMS 問題の解決 (診断)



---

## 第 19 章 統計分析ユーティリティー (DFSISTS0)

統計分析ユーティリティー (DFSISTS0) は、いずれかの IMS システム・ログ (バッチ領域からのものは除く) の情報を分析するときに使用します。

DFSISTS0 ユーティリティーは IMS.SDFSRESL データ・セット内にあります。

IMS システム・ログの選択部分に対して統計分析ユーティリティーを実行するときは、ログ・トランザクション分析ユーティリティーの使用により、ユーザー独自の指定に合わせた新しいログを作成できます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 『入力と出力』
- 550 ページの『JCL 仕様』
- 552 ページの『ユーティリティー制御ステートメント』

### 制約事項

統計分析には、以下の制約があります。

- 共通キュー・サーバー (CQS) のログ・レコードは、IMS ログ・レコードとは形式が異なるため、CQS ログを入力として使用することはできません。
- このユーティリティーは、このユーティリティーのリリース・レベルと同じリリースの IMS で作成された入力ログ・データ・セットのみに機能します。

### 前提条件

現在、DFSISTS0 ユーティリティーについて文書化されている前提条件はありません。

### 要件

現在、DFSISTS0 ユーティリティーについて文書化されている要件はありません。

### 推奨事項

現在、DFSISTS0 ユーティリティーについて文書化されている推奨事項はありません。

### 入力と出力

ログ・レコード

以下のログ・レコードは、IMS 統計分析ユーティリティーに使用されています。

- 01 入力メッセージは、宛先メッセージ・キューに入れられる準備ができてい  
る。
- 03 出力メッセージ・セグメントは、宛先メッセージ・キューに入れられる準備  
ができています。
- 07 アプリケーション・プログラムが終了した。
- 31 アプリケーション・プログラムは、次のメッセージを検索する「単独読み取  
り (get unique)」を出す。
- 34 メッセージが取り消され、そのメッセージの一部は以前にログに記録されて  
いる。
- 35 メッセージが宛先メッセージ・キューに入れられた。
- 36 メッセージが宛先メッセージ・キューから取り出された。
- 48 可変長の埋め込みログ・レコード

以下で、各ログ・レコード・タイプについて詳細に説明します。

#### ログ・タイプ 01

メッセージが通信で完全に受信され、宛先キューに入れる準備ができたときに、ログ・タイプ 01 レコードが作成されます。宛先キューは、スケジューラー・メッセージ・ブロック (SMB) か、通信名テーブル (CNT) のいずれかです。SMB 宛先の場合は、トランザクション・コードが端末オペレーターによって入力されたこと、そしてアプリケーション・プログラムがこれからスケジュールされることを意味します。CNT の場合は、メッセージ通信が行われることを意味します。端末オペレーターが LTERM とメッセージを入力する場合には、アプリケーション・プログラムは不要です。メッセージは、出力の目的で、入力メッセージで名前の指定された LTERM に直接キューイングされます。

#### ログ・タイプ 03

メッセージのセグメントがアプリケーション・プログラムによって作成されて、宛先メッセージ・キューに入れる準備ができたときに、03 レコードが作成されます。宛先メッセージ・キューは、SMB または CNT のいずれかです。SMB が宛先である場合は、「プログラム間」メッセージ通信がアプリケーション・プログラムによって呼び出されます。セグメントの宛先が CNT であれば、アプリケーション・プログラムは出力メッセージを LTERM へ送信します。

タイプ 03 レコードでは、日付と時刻のフィールド (PDATE および PTIME) は 01 レコードから送られます。統計ユーティリティーの実行の際には、03 レコードと 36 レコードが相互に関連して応答時間が決まります。この時間は、メッセージが入力キューに入れられた時刻 (03 レコードから入手される) から、メッセージが出力キューから取り出された時刻 (関連する 36 レコードから入手される) までの時間です。

#### ログ・タイプ 07

このレコードは、システムのアプリケーション会計レコードです。タイプ 07 レコードは、メッセージ処理領域またはバッチ・メッセージ処理領域でアプリケーション・プログラムが終了するときに作成されます。

#### ログ・タイプ 31

タイプ 31 レコードは、アプリケーション・プログラムが次のメッセージを検索するために「単独読み取り (get unique)」を出したときに作成されます。

#### ログ・タイプ 34

タイプ 34 レコードは、メッセージが取り消され、そのメッセージの一部が既にログに記録されている場合に作成されます。

#### ログ・タイプ 35

タイプ 35 レコードは、メッセージ (入力または出力) が宛先キューに入れられたときに作成されます。

メッセージが非常に長いため、複数の入力メッセージ・バッファを必要とする場合は、レコードに日付と時刻が含まれます。この条件下では、タイプ 01 レコード内の日付と時刻は無効になります。

#### ログ・タイプ 36

タイプ 36 レコードは、メッセージ全体が送信されて、そのメッセージをキューから解放する準備ができたときに作成されます。ディスプレイ装置以外のすべての装置では、最後のセグメントが正常に端末に送信されると、ただちにメッセージをキューから解放する準備が整います。ディスプレイ装置の場合は異なります。表示出力が 1 ページだけしかない場合は、最後のセグメントが正常に送信された後でメッセージはデキューされます。

表示出力が複数ページの場合は、システム定義時に `TERMINAL` マクロで選択した `PAGEDEL` オプションにより、メッセージがキューから取り出される準備のできた状態になる時点が決まります。`option=PAGEDEL` (または `PAGEDEL=YES`) を指定すると、ユーザーが疑問符 (?), PA2 キー、または新しい入力トランザクションを入力すると、メッセージはデキューされます。

`option=NPGDEL` (または `PAGEDEL=NO`) の場合は、出力キューからメッセージを取り出してタイプ 36 レコードを作成するために、疑問符 (?) または PA2 キーを入力する必要があります。

応答時間に対する `option=NPGDEL` (または `PAGEDEL=NO`) の影響が、顕著になる場合があります。表示された現行メッセージを長時間そのままにするか、あるいはビデオ装置の電源をオフにすると、メッセージは出力キューから除去されず、タイプ 36 レコードは端末操作が再び開始されるまで作成されません。結果として、応答時間は、数時間または幾日にも及ぶように見えます。

#### ログ・タイプ 48

タイプ 48 のレコードは、GMT 時間からの時間帯オフセットを含む、可変長の埋め込みログ・レコードです。

#### 報告書

報告書は、`PRINTDCB DD` ステートメントが組み込まれている場合に生成されます。さまざまなタイプの統計報告書は、次に説明されているとおりです。

- キューイングされたが送信されなかったメッセージ (宛先別)

出力メッセージ (X'03') はログに現れますが、メッセージが端末に送信されたことを示すレコード (X'36') は現れません。出力は、シンボリック端末名別にソートされます。

- プログラム間メッセージ (宛先別)

出力メッセージ (X'03') は `SMB` に送信されます。出力は、宛先別にソートされます。

- 回線および端末報告書

回線および端末報告書は、回線と端末の負荷を時刻別に示します。(これを使用して、回線と端末の使用状況、トラフィックのピーク期間などを判別できます。)

回線および端末報告書には、各 LTERM から IMS への入力メッセージ (R)、X'01' と、IMS から各 LTERM への出力メッセージ (S)、X'03' がカウントされます。

- メッセージ通信は 2 つのメッセージとカウントされます。すなわち、1 つは発信元端末からのもの、もう 1 つは宛先端末へのものです。
- ブロードキャスト・メッセージは、発信元端末からの 1 つのメッセージ、およびそれぞれ宛先端末へ送られる 1 つのメッセージとカウントされます。

次の 4 つの報告書は、トランザクション・コードを扱ったものです。出力メッセージが別の端末からのコマンドによって生成された場合は、入力接頭部データが、メッセージ「THIS OUTPUT NOT RESULT OF INPUT」で置き換えられます。システムが生成する X'03' メッセージ (端末入力には依存しない) は、トランザクション・コード IMSSYS を持っています。ログに入れられていない入力メッセージによって、あるいは同じ端末からのコマンド (例えば、DISPLAY) によって出力メッセージが生成された場合には、トランザクション・コードは NOTAVA です。それ以外の場合については、トランザクション・コードは生成される X'01' ログ・レコードの中にあります。

- キューイングされたが送信されなかったメッセージ (トランザクション・コード別)

出力メッセージ (X'03') はログに現れますが、メッセージが端末に送信されたことを示すレコード (X'36') は現れません。出力は、トランザクション・コード別にソートされます。

- プログラム間メッセージ (トランザクション・コード別)

出力メッセージ (X'03') は SMB に送信されました。出力は、トランザクション・コード別にソートされます。

- トランザクション報告書

この報告書は、トランザクション・コード別および時刻別の負荷を示すものです。

- 「R」は、各論理端末から IMS に入力メッセージが入力された時刻を示します。
- 「S」は、各論理端末へ出力メッセージが出力された時刻を表します。

この報告書では、回線および端末報告書と同じメッセージがカウントされます。入力、トランザクション・コード別にソートされます。

トランザクション・コードの欄には、次の項目が入ります。

**(NOSORC)**

出力メッセージはコマンドにより生成されました。



## (NOTAVA)

出力メッセージは、入力ログに入っていない入力メッセージにより生成されました。

## (IMSSYS)

出力メッセージは IMS により生成されました。

- トランザクション応答報告書

2 つの応答時間を測定します。最初の行は、入力メッセージを完全に受信してから (エンキュー時間 X'35')、端末への応答メッセージが正常にデキューされるまで (X'36') の応答時間です。2 番目の行は、入力メッセージを完全に受信してから (エンキュー時間 X'35')、端末への応答メッセージが開始されるまで (GU 時間 X'31') の応答時間です。

単一トランザクションから複数の応答が出される場合があります。それらの応答は、元の入力メッセージの結果であるプログラム間通信トランザクションからの出力メッセージを含むことができます。

百分位数の報告書は、最小応答、最長応答、第 25 百分位数応答、第 50 百分位数応答、第 75 百分位数応答、および第 95 百分位数応答を示します。第  $n^{\text{th}}$  百分位数内の応答時間は、そのトランザクション・コードについて処理された応答時間の合計数の  $n\%$  以上です。例えば、'75% RESPONSE' の欄の 04.3S という数は、そのトランザクションの合計応答の 75% が、04.3 秒以下だったことを意味します。

- アプリケーション会計報告書

各アプリケーション・プログラムまたはトランザクション・コードに計算機使用料を割り振るための十分なデータを提供します。

この報告書には次の情報が含まれています。

- DL/I に対するすべての要求のカウンント
- プロセッサ作業時間の量

DL/I のサービス (メッセージまたはデータベースへのアクセス) の要求すべてがカウンントされます。これらのカウンントは、プログラム別、プログラム内のトランザクション・コード別、およびトランザクション・コード内の優先順位別に累算されます。

処理されたメッセージのカウンントと、「単独読み取り」のカウンントが含まれます。これらのカウンントは異なるものになります。「単独読み取り」は複数のファイルと非ゼロの戻りコードを返す可能性のあるメッセージの要求であるからです。処理されたメッセージには、戻りコードが含まれません。

CPU 時間は、スケジューリングの要求が行われたときに設定されます。その値は、各トランザクションの最大時間にトランザクションの最大数を掛けた値です。残りの時間は、スケジューリングの次の要求の前に要求されます。この時間は、プログラムが実行された実際の時間です。これには、データにアクセスするための待ち時間は含まれていません。アプリケーション・プログラムが BMP であって、TTIMER または STIMER マクロを出した場合には、この時間は不適切なことがあります。

平均プロセッサ時間は、合計メッセージ・プロセッサ時間をメッセージの数で割ったものです。これは丸められません。最終平均プロセッサ時間は、再計算された平均です。

不良完了コードの数は、アプリケーション・プログラムが異常終了した回数、またはアプリケーション・プログラムがゼロ以外の値をレジスタ 15 に入れて戻った回数を表します。

- **IMS 会計報告書**

IMS 制御領域の開始時と停止時を示します。

メッセージ選択/コピー (リスト) オプション

メッセージ選択/コピー (リスト) の実行は、オプションです。

メッセージ選択/コピー (リスト) プログラムは、IMSLOGO と IMSLOGP のどちらかの DD ステートメントが含まれている場合に実行されます。

メッセージは、トランザクション、端末、または入力された時刻の順にリストまたはコピーできます。

## **JCL 仕様**

**EXEC** ステートメント

統計分析ユーティリティの DFSISTS0 を実行します。

```
//STEP1 EXEC PGM=DFSISTS0,PARM='LINECNT=XX'
```

LINECNT を指定しない場合、デフォルトは 36 です。

**DD** ステートメント

**STEPLIB DD**

ユーティリティ・プログラムを含んでいるプログラム・ライブラリーを記述します。形式は次のとおりです。

```
//STEPLIB DD DSNAME=IMS.SDFSRESL,DISP=SHR
```

**//LOGINxxx DD**

入力ログ・データ・セットを記述します。最後の 3 文字はオプションです。

各 IMS システムからのログ・データは、単一の DD ステートメントへ割り当てられている必要があります。IMS から複数のログ・データ・セットを使用する場合、それらのデータ・セットは連続していなければならない、作成された順序でそれらを連結する必要があります。LOGINxxx DD ステートメントの形式は次のとおりです。

```
//LOGIN DD DSNAME=IMSLOG,DISP=OLD,VOL=SER=XXXXXX,UNIT=TAPE
```

**SORTWK01-32 DD**

ソート・プログラムの作業データ・セットを記述します。定義されるスペースは異なることがあります。データ・セットは少なくとも 3 つ必要です。これらは、テープまたはディスクのいずれかに置くことができます。ディスク・ソートの場合の形式は、次のとおりです。

```
//SORTWKnn DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
```

### **SORTLIB DD**

ソート・プログラムのモジュールを含んでいるライブラリーを記述します。形式は次のとおりです。

```
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
```

### **SORTOUT DD**

出力データ・セットを使用しません。しかし、DCB 情報をこの DD ステートメントに指定する必要があります。形式は次のとおりです。

```
//SORTOUT DD DUMMY,DCB=*.LOGIN
```

### **SYSPRINT DD**

制御メッセージ用の出力データ・セットを記述します。形式は次のとおりです。

```
//SYSPRINT DD SYSOUT=A
```

次のステップはソートで、すべての DD ステートメント (SORTIN と SORTOUT は除く) は前のソートと同じです。

### **PRINTDCB DD**

報告書の出力、通常は出力ストリームを記述します。これは、ブロック化でも非ブロック化でも構いません。QSAM を使用して入出力が実行され、QSAM がバッファを獲得するからです。この DD ステートメントはオプションです。PRINTDCB DD ステートメントの形式は次のとおりです。

```
//PRINTDCB DD SYSOUT=A,DCB=(BLKSIZE=133,  
// LRECL=133,RECFM=FA)
```

### **IMSLOGP DD**

「メッセージの選択とリスト」出力を記述します。この DD ステートメントはオプションです。ステートメントの形式は、DFSIST30 の PRINTDCB ステートメントと同じです。

### **IMSLOGO DD**

「メッセージの選択とコピー」出力を記述します。この DD ステートメントはオプションです。IMSLOGO DD ステートメントの形式は次のとおりです。

```
// IMSLOGO DD DSNAME=OUTPUT,UNIT=tttt,  
// DISP=(NEW,KEEP),DCB=(RECFM=VB,  
// LRECL=32763,BLKSIZE=32767)
```

IMSLOGO DD ステートメントには、DFSISTS0 ユーティリティーが使用する、メッセージ・レコード、タイプ 01 ログ・レコード、およびタイプ 03 ログ・レコードからのテキスト・セグメントが含まれます。次の表は、各レコードの最初の 85 バイト (X'55') に含まれるソート・フィールドの内訳を示しています。

表 27. IMSLOGO DD ステートメントの最初の 85 バイト (X'55') のソート・フィールド

オフセット (16 進数)	長さ	説明
0	4	IMSLOGO レコードの長さ
4	4	グレゴリオ暦の日付 (YYYYMMDD)
8	1	04
9	9	ゼロ
12	8	メッセージが作成された時刻

表 27. IMSLOGO DD ステートメントの最初の 85 バイト (X'55') のソート・フィールド (続き)

オフセット (16 進数)	長さ	説明
1A	2	内部の順序付けフィールド
1C	1	メッセージのタイプ:  <b>01</b> 端末などの外部ソースから IMS へのメッセージ  <b>03</b> 端末などの外部ターゲットを宛先とするメッセージ  <b>02</b> プログラム間通信。ここで、1 つのトランザクションが別のトランザクションを呼び出します。
1D	8	ゼロ
25	8	トランザクション・コード
2D	8	入力端末 ID
35	4	入力端末 ID からのこのメッセージのシーケンス番号
39	8	メッセージ入力時刻
41	8	出力端末 ID
49	4	出力端末 ID へのこのメッセージのシーケンス番号
4D	8	メッセージ出力時刻

オフセット 55 で開始するのはメッセージ・テキスト全文であり、長さが 32,767 バイト (X'7FFF') を超えることができないレコードに収めるために、必要に応じて切り捨てられます。

## ユーティリティー制御ステートメント

メッセージ選択/コピー (リスト) プログラムでは、SYSIN から読み取られた制御ステートメントに基づいてメッセージが選択されます。選択されたメッセージには、印刷または出力データ・セットへのコピー (あるいはその両方) が行われます。IMSLOGO DD ステートメントを含めると、出力データ・セットが作成されます。IMSLOGP DD ステートメントを含めると、選択されたメッセージが印刷されます。単一の実行内に両方の DD ステートメントを含めることができます。

制御ステートメントには下記の制約があります。

- すべての制御ステートメントは 1 桁目から始まり、その制御ステートメントを識別するキーワードを持ちます。
- キーワードの後に一連のパラメーターを続け、それらのパラメーターをコンマで区切り、括弧で囲みます。
- 制御ステートメントは、71 桁目を超えて続けることはできません。

- 同じキーワードを持ち、1 桁目から始まる複数の制御ステートメントを指定することができます。
- 括弧内ではすべてのパラメーターは定位置パラメーターで、欠落パラメーターはコンマで示す必要があります。
- 制御ステートメントで指定される基準のうち最低 1 つをメッセージが満たす場合に、そのメッセージは選択されます。

パラメーターの最後を \* で終了させることにより、名前のグループであることを示すことができます。

例：INV\* とすると、INV、INVENTORY、INVA、および INVB という名前が選択されます。

名前パラメーター ALL を使用すると、指定した名前ではなくすべての名前を選択することができます。

#### トランザクション・コード制御ステートメント

トランザクション・コード制御ステートメントの形式は、次のとおりです。

```
TRANS CODE=(TRANSCOD,I,0),(TRANSA,I),(INV*,,0),(ALL,I,0)
```

- 第 1 パラメーターは、1 から 8 文字のトランザクション・コードです。
- 第 2 パラメーターの I により、このコードを持つ入力メッセージが選択されます。
- 第 3 パラメーターの O により、このコードの結果生じる出力メッセージが選択されます。
- すべてのトランザクション・コードの ALL により、トランザクション・コードが選択されます。
- トランザクション・コード内にアスタリスクがあると、アスタリスクより前の文字のみが、入力トランザクション・コードにある対応する数の文字と比較されて、選択の判別が行われます。これを使用して、トランザクション・コードのグループを選択することができます。

#### シンボリック端末名制御ステートメント

以下に、シンボリック端末名制御ステートメントの例を示します。

```
SYM NAME=(TERMA,I,0,ALL),(TERM*,I,,ALL),(TERMINV,,0,ALL)
SYM NAME=(TERMPAY,I,0,TERMA)
SYM NAME=(ALL,,0,TERMA)
```

- 第 1 パラメーターは入力メッセージの発信元のシンボリック端末名で、1 から 8 文字の長さにすることができます。
- 第 2 パラメーターの I により、この端末からの入力を選択されます。
- 第 3 パラメーターの O により、この端末からの入力によって生成される出力が選択されます。
- 第 4 パラメーターは出力メッセージの宛先のシンボリック端末名で、1 から 8 文字の長さにすることができます。
- シンボリック端末名を ALL にすると、すべてのシンボリック端末名が選択されます。

- 端末名内にアスタリスクがあると、アスタリスクより前の文字のみが、入力端末名にある対応する数の文字と比較されて、選択の判別が行われます。これを使用して、端末名のグループを選択することができます。

#### 時間制御ステートメント

時間制御ステートメントの形式は次のとおりです。

TIME=(yyddd,hhmm[+|-]HHMM),yyddd,hhmm[+|-]HHMM)

- 第 1 パラメーターは、年 (YY) と年間通算日 (DDD) で表された開始日です。
- 第 2 パラメーターは、時間 (HH) と分 (MM)、およびオプションの時間帯オフセット情報 ({+|-}HHMM) で表された開始時刻です。
- 第 3 パラメーターは、年 (YY) と年間通算日 (DDD) で表された終了日です。
- 第 4 パラメーターは、時間 (HH) と分 (MM)、およびオプションの時間帯オフセット情報 ({+|-}HHMM) で表された終了時刻です。

第 2 パラメーターと第 4 パラメーターで使用するオプションの時間帯パラメーターは次のとおりです。

- {+|-} は、協定世界時 (UTC) からの時間帯オフセットの符号です。
- HH は、UTC からのオフセットの総時間数です。
- MM はオフセットの分であり、00、15、30、45、または空白を指定することができます。
- 時間制御ステートメントを含めた場合は、トランザクション・コード・ステートメントまたは端末制御ステートメントで指定され、しかも指定の時間内に入っているメッセージのみが選択されます。

#### 印刷不能文字制御ステートメント

印刷不能文字制御ステートメントの形式は次のとおりです。

NON PRINT=HEX

この制御ステートメントを含めた場合は、印刷不能文字が 16 進数で 2 行に印刷されます。すなわち、1 つの 16 進文字がもう一方の 16 進文字の上に印刷されます。このステートメントを含めなければ、印刷不能文字は空白として表示されます。

#### メッセージ選択出力順序ステートメント


メッセージ選択出力順序ステートメントのフォーマットは、以下のとおりです。

ORDER=CREATE|SOURCE|DEST

メッセージがリストされる場合、デフォルトの順序は、最初の入力メッセージの時刻の順序でリストすることです (「最初」とは、そのメッセージの結果として生成されたメッセージ・グループ内で「最初」の意味です。このグループは通常、入力メッセージとその応答から構成されますが、中間プログラム間通信が存在する場合は、それらの通信も含まれます)。このデフォルトの順序は CREATE です。

最初の入力メッセージの LTERM の順にメッセージをリストするには、SOURCE を指定し、初期メッセージ (メッセージ通信以外の何かの初期トランザクション) のターゲットの順にメッセージをリストするには、DEST を指定します。

関連概念:

 統計分析報告書、ログ・トランザクション報告書、およびログ・レコード分析 (システム管理)

関連資料:

531 ページの『第 17 章 ログ・トランザクション分析ユーティリティー (DFSILTA0)』

---

## DFSISTS0 ユーティリティーの例

以下の例は、//PRINTDCB DD ステートメントか //IMSLOGP DD ステートメント、またはその両方を指定して実行した統計分析ユーティリティー (DFSISTS0) によって生成された出力を示しています。

### 統計分析ユーティリティー用の JCL

以下の図は、統計分析ユーティリティーの実行用の JCL の例です。これは、日付別にソートされるすべての統計のジョブ・ストリームの例で、これにより日付制御のもとで報告書が作成されます。すべてのデータ・セット内の BLKSIZE と LRECL は、入力ログによって異なります。

```
//STATS    JOB 1,NAME,MSGCLASS=A,MSGLEVEL=1,PRTY=8
//STEPLIB  DD DSN=IMS.SDFSRESL,DISP=SHR
//*
//ST1      EXEC PGM=DFSISTS0
//SORTLIB  DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//LOGIN    DD DSN=IMSLOG,DISP=OLD,
//          UNIT=TAPE,VOL=SER=LOGTAP
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTOUT  DD DUMMY
//PRINTDCB DD SYSOUT=A
//IMSLOGP  DD SYSOUT=A
//SYSIN    DD *
TRANS CODE=(ALL,I,0)
NON PRINT=HEX
/*
//
```

### 統計報告書の例

以下に、DFSISTS0 によって生成される統計報告書のリストを示します。その後に報告書の例を示します。以下の例の右上隅に示されている報告書の日付は、これらの例のように、日付によるソートを指定した場合以外は表示されません。以下のような報告書が生成されます。

- キューイングされたが送信されなかったメッセージ (宛先別) ( 556 ページの『キューイングされたが送信されなかったメッセージ (宛先別)』)
- プログラム間メッセージ (宛先別) ( 556 ページの『プログラム間メッセージ (宛先別)』)
- 回線および端末 ( 557 ページの『回線および端末報告書』)

- キューイングされたが送信されなかったメッセージ (トランザクション・コード別) ( 558 ページの『キューイングされたが送信されなかったメッセージ (トランザクション・コード別)』)
- プログラム間メッセージ (トランザクション・コード別) ( 558 ページの『プログラム間メッセージ (トランザクション・コード別)』)
- トランザクション ( 559 ページの『トランザクション報告書』)
- トランザクション応答 ( 559 ページの『トランザクション応答報告書』)
- アプリケーション会計 ( 560 ページの『アプリケーション会計報告書』)

## 正規化浮動小数の使用例

統計報告書の数値が大きすぎて指定のスペースに収まらない場合、IMS は、数値全体の代わりに正規化浮動小数を使用します。

正規化浮動小数では、文字 E を使用して、「10 倍して X の累乗に上げる」ことを意味します。

例えば、25,000 は 2.5E4 と書くこともでき、3.7E5 は 370,000 に相当します。

## キューイングされたが送信されなかったメッセージ (宛先別)

M E S S A G E S -- Q U E U E D B U T N O T S E N T		D A T E 05/15/09	P A G E 00001
TOTAL			
DESTINATION	MESSAGES		
AZCOGH00	1		
CWARENS1	1		
DFSMTCNT	2		
F@MYDV21	1		
OQCCM211	1		

## プログラム間メッセージ (宛先別)

M E S S A G E S -- P R O G R A M T O P R O G R A M		D A T E 05/15/09	P A G E 00001
TOTAL			
DESTINATION	MESSAGES		
VGG034T1	2		
VJDBBAT	32		
VJDBI4T	3		
VJDBL1T	72		
VJDBL3T	4		
VJDBL4T	39		
VJDBMLR	11		
VJDBML2	14		
VJDBR1T	74		
VJMPCC1	1		
VJMPCC1B	8		
VJMPCC1C	4		
VJMPCC1D	7		
VJMPCC1E	5		
VJMPCC1F	3		
VJMPCC2	118		
VJMPCC3	8		
VODSWC	1		
VODSWT	73		
VOGSF07	4		
VOMPCC1	5		
VOMPCC1A	13		
VOMPCC1B	9		
VOMPCC1C	14		
VOMPCC1D	3		
VOMPCC1E	1		
VOMPCC1F	2		



VOMPCC2C	174
VOMPCC3	2
VOSSADD	72
VOSSBIR	2
VOSSFDI	17
VOSSIHR	2
VOSSLG	2
VOSSLPT	1
VOSSNFH	38
VOSSNMA	31
VOSSSCR	9
VOSSSMI	4
VOSSSUP2	1
VOSSTDC	1
VOS027T1	2
VOTABAD	8
VOTAFDI	3
VOTAFD02	8
VOTAMDS	12
VOTANSA	1
VOTAPAS	2
VOTAPTA	2
VOTARFT	1
VOTASEB	10
VOXPECIN	1
YXCAJFX	1

### 回線および端末報告書

LINE		AND TERMINAL		REPORT		DATE 05/15/09													PAGE 00001	
NODE	R/S	TOTAL	TOTAL	AVG	HOURLY DISTRIBUTION															
		MESSAGES	CHARACTERS	SIZE	00-07	07-08	08-09	09-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-24		
AADUGNA4	R	5	8,675	1,735	0	0	0	0	0	0	2	3	0	0	0	0	0	0		
	S	5	8,675	1,735	0	0	0	0	0	0	2	3	0	0	0	0	0	0		
ACRADD01	S	3	25,194	8,398	0	0	0	0	0	0	3	0	0	0	0	0	0	0		
ADEHORN0	R	1	1,735	1,735	0	0	0	0	0	0	0	1	0	0	0	0	0	0		
	S	2	8,675	4,337	0	0	0	0	0	0	0	2	0	0	0	0	0	0		
ADEHORN1	R	1	3,807	3,807	0	0	0	0	0	0	0	1	0	0	0	0	0	0		
	S	2	3,807	1,903	0	0	0	0	0	0	0	2	0	0	0	0	0	0		
ADHOWAR0	R	2	5,718	2,859	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
	S	2	5,718	2,859	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
ADHOWAR3	R	1	1,886	1,886	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	S	1	11,316	11,316	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
ADHOWAR4	R	2	6,908	3,454	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
	S	2	6,908	3,454	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
ADOVALIO	R	2	3,454	1,727	0	0	0	0	0	0	2	0	0	0	0	0	0	0		
ADUNCAN0	R	1	1,800	1,800	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	S	1	3,600	3,600	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
AJWHIT10	R	3	2,233	744	0	0	0	0	0	0	3	0	0	0	0	0	0	0		
	S	2	253	126	0	0	0	0	0	0	2	0	0	0	0	0	0	0		
AKCAMPB0	S	8	13,880	1,735	0	0	0	0	0	0	4	4	0	0	0	0	0	0		
ALOPEZ2	R	4	5,852	1,463	0	0	0	0	0	0	2	2	0	0	0	0	0	0		
ALUOMA1	R	1	1,758	1,758	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	S	1	2,158	2,158	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
AMRAMOS0	R	1	1,152	1,152	0	0	0	0	0	0	0	1	0	0	0	0	0	0		
	S	1	1,152	1,152	0	0	0	0	0	0	0	1	0	0	0	0	0	0		
ASMYTH0	R	5	11,701	2,340	0	0	0	0	0	0	2	3	0	0	0	0	0	0		
	S	5	12,224	2,444	0	0	0	0	0	0	2	3	0	0	0	0	0	0		
ASUTTON2	R	6	10,173	1,695	0	0	0	0	0	0	3	3	0	0	0	0	0	0		
	S	6	15,447	2,574	0	0	0	0	0	0	3	3	0	0	0	0	0	0		
ATFREVE0	R	3	4,762	1,587	0	0	0	0	0	0	3	0	0	0	0	0	0	0		
AZCOGH00	S	4	8,632	2,158	0	0	0	0	0	0	0	4	0	0	0	0	0	0		
BDUNKER0	R	7	9,821	1,403	0	0	0	0	0	0	6	1	0	0	0	0	0	0		
BEWILLIO	R	3	9,372	3,124	0	0	0	0	0	0	2	1	0	0	0	0	0	0		
	S	3	28,710	9,570	0	0	0	0	0	0	2	1	0	0	0	0	0	0		
BMANZANO	R	1	2,158	2,158	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	S	1	2,158	2,158	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
BRASMUS0	R	3	6,474	2,158	0	0	0	0	0	0	3	0	0	0	0	0	0	0		
	S	3	6,474	2,158	0	0	0	0	0	0	3	0	0	0	0	0	0	0		
BRICHINO	R	1	1,758	1,758	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	S	1	3,807	3,807	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
BXSMI255	R	6	15,540	2,590	0	0	0	0	0	0	2	4	0	0	0	0	0	0		
	S	6	15,540	2,590	0	0	0	0	0	0	2	4	0	0	0	0	0	0		
CALLRVW0	R	2	1,988	994	0	0	0	0	0	0	0	2	0	0	0	0	0	0		
	S	4	12,398	3,099	0	0	0	0	0	0	0	4	0	0	0	0	0	0		
CBUNNELO	R	2	3,454	1,727	0	0	0	0	0	0	0	2	0	0	0	0	0	0		
CDESJAR0	R	5	8,229	1,645	0	0	0	0	0	0	2	3	0	0	0	0	0	0		
CEWEBER1	R	4	6,768	1,692	0	0	0	0	0	0	0	4	0	0	0	0	0	0		
CFERRAN4	R	1	503	503	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
CGALDE0	R	1	1,735	1,735	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	S	1	86,750	86,750	0	0	0	0	0	0	1	0	0	0	0	0	0	0		

CGALDE1 R	1	1,735	1,735	0	0	0	0	0	0	0	1	0	0	0	0	0
CGOOLSB1 R	2	2,926	1,463	0	0	0	0	0	0	0	2	0	0	0	0	0
CGUCINS3 R	9	19,371	2,152	0	0	0	0	0	0	8	1	0	0	0	0	0
S	9	19,371	2,152	0	0	0	0	0	0	8	1	0	0	0	0	0
CKLARS00 R	1	503	503	0	0	0	0	0	0	0	1	0	0	0	0	0
CMEN00 R	1	3,807	3,807	0	0	0	0	0	0	1	0	0	0	0	0	0
S	1	3,807	3,807	0	0	0	0	0	0	1	0	0	0	0	0	0
COAAHS00 S	10	30,130	3,013	0	0	0	0	0	0	3	7	0	0	0	0	0
COJJHP00 S	4	8,632	2,158	0	0	0	0	0	0	4	0	0	0	0	0	0
COLAM030 R	20	31,330	1,566	0	0	0	0	0	0	9	11	0	0	0	0	0
CRUNNEL1 R	2	2,652	1,326	0	0	0	0	0	0	2	0	0	0	0	0	0

注:

1. LINE RTN = 回線相対端末番号
2. R/S = 受信/送信

キューイングされたが送信されなかったメッセージ (トランザクション・コード別)

M E S S A G E S -- Q U E U E D B U T N O T S E N T		D A T E 05/15/09		P A G E 00001	
TRANSACTION CODE	TOTAL MESSAGES				
VOMPCC1A	1				
VOS1SIGN	1				

プログラム間メッセージ (トランザクション・コード別)

M E S S A G E S -- P R O G R A M T O P R O G R A M		D A T E 05/15/09		P A G E 00001	
TRANSACTION CODE	TOTAL MESSAGES				
VJDBI4T	6				
VJDBL1T	144				
VJDBL3T	9				
VJDBL4T	88				
VJMPCC1	2				
VJMPCC1A	117				
VJMPCC1B	4				
VJMPCC1C	1				
VJMPCC1D	4				
VJMPCC1F	1				
VJMPCC3	2				
VOCITADD	5				
VOCITFCT	3				
VODSWT	73				
VOGSC02	3				
VOGSF05	3				
VOGSJFX	6				
VOMPCC1A	200				
VOMPCC1B	20				
VOMPCC2C	1				
VOSSADD	59				
VOSSCWL	1				
VOSSESO	11				
VOSSFDI	20				
VOSSIA	10				
VOSSNFH	10				
VOSSNMA	62				
VOSSSCR	9				
VOSSSMI	4				
VOSSSUP2	1				
VOSSTIM	1				
VOS019T1	30				
VOTABAD	3				
VOTAFDI	6				
VOTAFD02	12				
VOTANSA	1				
VOTAPAS	2				
VOTARFT	1				
VOTATREB	1				
VOTATTE	1				

## トランザクション報告書

TRANSACTION REPORT				DATE 05/15/09													PAGE 00001	
TRANSACTION	TOTAL	TOTAL	AVG	HOURLY DISTRIBUTION														
CODE	R/S	MESSAGES	CHARACTERS	SIZE	00-07	07-08	08-09	09-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-24
VGCAMSS	R	1	1,524	1,524	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VGGDCIS	R	1	1,263	1,263	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VGGDCMC	R	2	2,888	1,444	0	0	0	0	0	0	1	1	0	0	0	0	0	0
VGGDEWRA	R	2	3,640	1,820	0	0	0	0	0	0	0	2	0	0	0	0	0	0
VGGDISS	R	4	2,136	534	0	0	0	0	0	0	3	1	0	0	0	0	0	0
VGGDRIP	R	4	6,728	1,682	0	0	0	0	0	0	1	3	0	0	0	0	0	0
VGGDRIS	R	1	1,718	1,718	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VGGDRMC	R	4	6,868	1,717	0	0	0	0	0	0	3	1	0	0	0	0	0	0
VGGDSOIE	R	5	8,316	1,663	0	0	0	0	0	0	2	3	0	0	0	0	0	0
VGGDTGM1	R	1	1,753	1,753	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VGGDTGS1	R	1	1,718	1,718	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VGGDTLOG	R	47	62,741	1,334	0	0	0	0	0	0	24	23	0	0	0	0	0	0
VGGDTRAK	R	5	7,960	1,592	0	0	0	0	0	0	5	0	0	0	0	0	0	0
VGG015T1	R	4	4,912	1,228	0	0	0	0	0	0	4	0	0	0	0	0	0	0
VGG022TC	R	3	5,475	1,825	0	0	0	0	0	0	1	2	0	0	0	0	0	0
VGG022TE	R	1	1,825	1,825	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VGG022T1	R	7	5,879	839	0	0	0	0	0	0	3	4	0	0	0	0	0	0
VGG022T2	R	9	16,425	1,825	0	0	0	0	0	0	5	4	0	0	0	0	0	0
VGG043T1	R	7	10,789	1,541	0	0	0	0	0	0	3	4	0	0	0	0	0	0
VGS1SIGN	R	13	3,289	253	0	0	0	0	0	0	6	7	0	0	0	0	0	0
VJCRLCRE	R	1	674	674	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJCRLCRE	S	1	412	412	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJCRREF	R	1	681	681	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJCRREF	S	1	1,996	1,996	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJDBL4T	S	2	3,002	1,501	0	0	0	0	0	0	0	2	0	0	0	0	0	0
VJMPCC1	R	1	7,164	7,164	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VJMPCC1	S	1	576	576	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJMPCC1A	R	7	1,381	197	0	0	0	0	0	0	0	7	0	0	0	0	0	0
VJMPCC1B	S	1	3,850	3,850	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VJMPCC1C	S	3	24,497	8,165	0	0	0	0	0	0	0	3	0	0	0	0	0	0
VJMPCC1E	S	2	1,994	997	0	0	0	0	0	0	2	0	0	0	0	0	0	0
VJMPCC2	R	6	822	137	0	0	0	0	0	0	5	1	0	0	0	0	0	0
VJMPCC2	S	1	923	923	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJMPCC3	S	2	796	398	0	0	0	0	0	0	1	1	0	0	0	0	0	0
VJS1SIGN	R	1	253	253	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VJS1SIGN	S	1	253	253	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VOCAMSS	R	6	9,144	1,524	0	0	0	0	0	0	5	1	0	0	0	0	0	0
VOCAMSS	S	23	35,052	1,524	0	0	0	0	0	0	15	8	0	0	0	0	0	0
VOCICIQ1	R	61	16,403	268	0	0	0	0	0	0	31	30	0	0	0	0	0	0
VOCICIQ1	S	61	119,302	1,955	0	0	0	0	0	0	30	31	0	0	0	0	0	0
VOCICIQ2	R	11	4,785	435	0	0	0	0	0	0	5	6	0	0	0	0	0	0
VOCICIQ2	S	11	55,131	5,011	0	0	0	0	0	0	5	6	0	0	0	0	0	0
VOCIEAU1	R	1	248	248	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VOCIEAU1	S	1	408	408	0	0	0	0	0	0	1	0	0	0	0	0	0	0
VOCIGARM	R	22	18,843	856	0	0	0	0	0	0	12	10	0	0	0	0	0	0
VOCIGARM	S	22	8,002	363	0	0	0	0	0	0	12	10	0	0	0	0	0	0
VOCIGIQ1	R	28	7,532	269	0	0	0	0	0	0	13	15	0	0	0	0	0	0
VOCIGIQ1	S	28	424,865	15,173	0	0	0	0	0	0	13	15	0	0	0	0	0	0
VOCIMWL1	R	1	311	311	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VOCIMWL1	S	1	63,211	63,211	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VOCISRH1	R	30	8,278	275	0	0	0	0	0	0	16	14	0	0	0	0	0	0
VOCISRH1	S	30	30,910	1,030	0	0	0	0	0	0	16	14	0	0	0	0	0	0
VOCITADD	R	5	3,513	702	0	0	0	0	0	0	3	2	0	0	0	0	0	0
VOCITADD	S	5	7,659	1,531	0	0	0	0	0	0	3	2	0	0	0	0	0	0
VOCITFCT	R	4	3,274	818	0	0	0	0	0	0	2	2	0	0	0	0	0	0
VOCITFCT	S	4	8,082	2,020	0	0	0	0	0	0	2	2	0	0	0	0	0	0
VOCITIQ1	R	12	3,400	283	0	0	0	0	0	0	4	8	0	0	0	0	0	0
VOCITIQ1	S	12	26,918	2,243	0	0	0	0	0	0	4	8	0	0	0	0	0	0
VOCRLCRE	R	1	674	674	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VOCRLCRE	S	1	412	412	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VOCRREF	R	1	681	681	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VOCRREF	S	1	1,996	1,996	0	0	0	0	0	0	0	1	0	0	0	0	0	0

## トランザクション応答報告書

TRANSACTION RESPONSE REPORT			DATE 05/15/09				PAGE 00001
TRANSACTION	TOTAL	LONGEST	95%	75%	50%	25%	SHORTEST
CODE	RESPONSES	RESPONSE	RESPONSE	RESPONSE	RESPONSE	RESPONSE	RESPONSE
VJCRLCRE	1	.11S	.11S	.11S	.11S	.11S	.11S
VJCRREF	1	.08S	.08S	.08S	.08S	.08S	.08S
VJCRREF	1	.03S	.03S	.03S	.03S	.03S	.03S
VJMPCC1A	9	5.02S	5.01S	5.01S	.29S	.05S	.04S
VJMPCC1C	9	.36S	.31S	.16S	.06S	.03S	.01S
VJMPCC1C	1	.10S	.10S	.10S	.10S	.10S	.10S
VJMPCC1E	1	.02S	.02S	.02S	.02S	.02S	.02S
VJMPCC1E	1	.01S	.01S	.01S	.01S	.01S	.01S
VJMPCC2	1	.01S	.01S	.01S	.01S	.01S	.01S
VJMPCC2	1	.00S	.00S	.00S	.00S	.00S	.00S
VJS1SIGN	1	.18S	.18S	.18S	.18S	.18S	.18S



VOS041T	VOS041T1	8	3	6	0	4	18	0	9	7	0	2	9	3	0	0	0.0S	0.005S
VOS042T	VOS042T1	9	214	217	0	214	1,409	158	25	241	0	28	428	203	0	0	0.3S	0.001S
VOS043T	VOS043T1	9	52	54	0	194	5,390	1.6E4	57	100	0	48	388	51	0	0	2.0S	0.038S
VOS1SGN	VOS1SIGN	10	23	28	0	43	85	788	0	40	0	0	0	0	40	0	0.0S	0.002S
VOS102T	VOS102T1	8	4	7	0	4	24	0	0	12	0	0	6	4	4	0	0.0S	0.004S
VOS104T	VOS104T1	8	16	22	0	12	69	0	0	28	0	0	12	12	0	0	0.0S	0.002S
VOTAACR	VOTAACR	8	7	12	0	7	28	0	0	7	0	0	7	7	0	0	0.0S	0.003S
VOTABAD	VOTABAD	9	8	14	0	4	356	84	152	234	52	18	288	51	61	0	0.2S	0.031S
VOTADRP	VOTADRP	8	2	3	0	1	2	2,504	0	0	0	0	0	0	0	0	0.2S	0.129S
VOTAFDI	VOTAFDI	8	3	4	2	10	7	0	2	9	6	1	10	6	2	0	0.0S	0.006S
VOTAFDO	VOTAFDO2	8	8	13	17	23	51	5	4	75	46	10	60	47	23	0	0.1S	0.013S
VOTAFEB	VOTAFEB	8	2	3	0	2	10	0	0	4	0	1	3	2	1	0	0.0S	0.005S
VOTAMDS	VOTAMDS	8	12	20	0	0	0	0	0	0	0	0	0	0	0	0	0.0S	0.001S
VOTANSA	VOTANSA	8	1	2	0	1	2	0	1	3	0	0	7	1	0	0	0.0S	0.015S
VOTAPAS	VOTAPAS	9	2	4	0	3	8	0	1	11	10	2	17	10	3	0	0.0S	0.019S
VOTAPTA	VOTAPTA	8	2	4	4	0	0	0	0	2	0	0	0	2	0	0	0.0S	0.002S
VOTARFT	VOTARFT	8	2	4	7	4	7	0	0	11	6	2	11	4	2	0	0.0S	0.013S
VOTASEB	VOTASEB	8	10	14	0	10	15	0	0	0	0	0	0	0	0	0	0.0S	0.001S
VOTATBL	VOTATTE	12	1	2	0	2	38	5	3	10	0	1	19	1	2	0	0.0S	0.023S
VOTATEB	VOTATREB	12	3	4	0	4	14	0	0	9	4	3	8	7	1	0	0.0S	0.004S
VOTATSR	VOTATSR	8	1	2	0	2	5	0	0	5	2	1	3	0	0	0	0.0S	0.004S
VOTATST	VOTATST	8	1	2	0	0	2	0	0	3	0	3	0	1	0	0	0.0S	0.003S
VOXPECI	VOXPECIN	8	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0.0S	0.013S
SYSTEM TOTALS			3,097	3,431	3,238	7,999	2.5E4	3.7E4	1.9E4	1.0E4	2.5E4	945	1.2E4	2,964	2,730	0	14.0S	0.004S
I M S	ACCOUNTING	REPORT										D A T E	05/15/09				P A G E	00001

START TIME 19:59:34  
STOP TIME 20:00:22  
REPORT PERIOD IS FROM 05/15/09 TO 05/15/09.

END OF REPORTS

\* 次の条件がすべて満たされている場合、2番目の挿入は、1ユーザーが出した挿入としてカウントされます。

1. 新しい HIDAM または PHIDAM ルート
2. 重複キーではない (II の状況が戻されない)

\*\* DFSIST30 への入力を日付制御を用いてソートしておかないと、これらの日付は表示されません。

### メッセージ選択/コピー (リスト) オプション報告書

以下の図は、メッセージ選択/コピー (リスト) オプションによって作成された報告書の例を示したものです。

MESSAGES

INPUT PREFIX	TRANSACTION CODE	LINE NO	RELA TERM	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME	OUTPUT PREFIX	NODE NAME	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME	
	THIS OUTPUT NOT RESULT OF INPUT								DSWP5008	00017	PDSW5008	09.107	15.54.1	
OUTPUT	SEG=001	LEN=0001	*F*											
INPUT PREFIX	TRANSACTION CODE	LINE NO	RELA TERM	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME	OUTPUT PREFIX	NODE NAME	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME	
	THIS OUTPUT NOT RESULT OF INPUT								DSWP5008	00019	PDSW5008	09.107	15.54.4	
OUTPUT	SEG=001	LEN=0009	*88-3-2000*											
OUTPUT	SEG=002	LEN=0248	*WITHDRAWAL		\$300.00	FDEPOSIT	\$6704.62	FSAVINGS	444.44	F	CHECKING	\$9800.50	F*	
			*OVERDRAFT		\$30.32	FVISA	\$2020.20	FMASTER	CHRG	\$105.00	F	CARLOAN	\$1040.00	F*
			*TRANSFER C-5 \$50.00 FCHRISTMAS CLUB \$94.60*											
INPUT PREFIX	TRANSACTION CODE	NODE NAME	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME	OUTPUT PREFIX	NODE NAME	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME		
	DE1Q	DSWP0056	00015	PDSW0056	09.107	15.53.51		DSWP0056	00016	PDSW0056	07.107	15.54.2		
		03	0	18D										
INPUT	SEG=001	LEN=0016	*1BDE1Q 3Y43A*											
INPUT	SEG=002	LEN=0230	*23(9) WITHDRAW OF \$300DEPOSIT OF \$6704.62SAVINGS DEPOSIT OF \$444.44CHECKING TRANSFER OF \$9800.500V*											
			*ERDRAFT OF \$30.32VISA ENTRY OF \$2020.20MASTER CHARGE OF \$105.00CAR LOAN OF \$140.00TRANSFER C-S OF \$*1											
			*50.00CHRISTMAS CLUB OF \$94.60Y											
INPUT PREFIX	TRANSACTION CODE	NODE NAME	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME								
	DE1Q	DSWP0084	00017	PDSW5008	09.107	15.54.25								
OUTPUT	SEG=001	LEN=0031	** DATA SUCCESSFULLY RECEIVED +F*											

INPUT PREFIX	TRANSACTION CODE	LINE NO	RELA TERM	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME	OUTPUT PREFIX	NODE NAME	SEQ NO	SYMBOLIC ADDRESS	DATE	TIME
	DE1Q <sup>2</sup>	DSWP016		00018	PDSW0116	09.107	15.54.36		DSWP0116	00016	PDSW0116	07.107	15.54.36

注:

1. 230 文字の報告書メッセージを示します。
2. トランザクション・コード「DE1Q」で生成され、相対端末 DSWP0116 に伝送された 31 文字のメッセージを示します。

---

## 第 4 部 ログ・ユーティリティー

ログ・ユーティリティーは、SLDS を生成するとき、システム・ログ・データ・セットからデータ・セットを生成するとき、エラーを含んでいるログ・データ・セット、または正常に終了しなかったログ・データ・セットから使用可能なログ・データ・セットを生成するときに使用します。

各トピックでは、ユーティリティーがどのように機能するのかを説明し、その使用における要件と制約事項を定義しています。また、例を提供しています。





## 第 20 章 ログ保存ユーティリティー (DFSUARCO)

ログ保存ユーティリティー (DFSUARCO) は、いっぱいになった OLDS またはバッチ IMS SLDS から SLDS を生成するときに使用できます。

IMS DB は、テープまたは DASD に入れることができる SLDS にログ・レコードを書き込みます。これにより、IMS バッチ・ユーザーは、DASD にログを記録し、SLDS を作成し、後でその SLDS を DASD またはテープにコピーすることができます。

ログ保存ユーティリティーには、以下に示すオプションの機能があります。これらの機能は、ユーティリティー制御ステートメントで指定する必要があります。

### RLDS (リカバリー・ログ・データ・セット) を作成する場合

DB リカバリーに必要なすべてのログ・レコードを含んでいる出力データ・セットの作成を要求できます。この出力データ・セットを、リカバリー・ログ・データ・セット (RLDS) と呼びます。入力データ・セットが DB リカバリー用のレコードを含んでいる場合には、RLDS は DBRC によって認識され、DB リカバリーおよび変更累積用の JCL を作成する際に、これが SLDS の代わりに GENJCL によって使用されます。DB リカバリーに必要なレコードが入力データ・セットに含まれていない場合、RLDS はヌル・データ・セットです。この場合には、DBRC は、RLDS DSNNAME およびボリューム通し番号の代わりに SLDS のデータ・セット名とボリューム通し番号を記録し、ヌル RLDS の代わりに GENJCL の SLDS を使用します。

### SLDS のログ・レコードの省略

一般に、SLDS は OLDS からのすべてのログ・レコードを含みますが、SLDS から一部のタイプのログ・レコードを省略する必要がある場合には、これらのログ・レコードを NOLOG パラメーターを使って SLDS 制御ステートメントに指定する必要があります。SLDS は、データベース・リカバリーと IMS 再始動に必要なレコードを含んでいなければなりません。必要なレコード・タイプが省略するものとして指定されていると、ログ保存ユーティリティーは、エラー・メッセージを出して終了します。

### ユーザー・データ・セットへのログ・レコードのコピー

ログ保存ユーティリティーは、選択したログ・レコードを複数のユーザー・データ・セットに直接コピーすることができます。SYSIN 制御ステートメントに、選択するログ・レコード、およびログ・レコードを書き込むデータ・セットの DD 名を指定することができます。

### ユーザー出口ルーチンの指定

ログ保存ユーティリティーに複数のユーザー出口ルーチンを指定することができます。ログ保存ユーティリティーは、初期設定時、入力ログ読み取り時、および終了時に、各ユーザー出口ルーチンに制御を渡します。ユーザー出口ルーチンでは、ログ・レコードの処理や、データ・セットの作成を行うことができます。

### 強制ボリュームの終わり (EOV) の指定

テープ上の複式 SLDS 中の対応ボリュームが確実に同じレコードを含むように

(したがって交換可能であるように) するために、ボリュームに書き込むブロックの数を指定できます。指定の数のログ・ブロックが書き込まれたときに、両方の SLDS で強制的に EOV になります。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 『入力と出力』
- 569 ページの『JCL 仕様』
- 571 ページの『ユーティリティー制御ステートメント』
- 576 ページの『戻りコード』

## 制約事項

現在、DFSUARC0 ユーティリティーについて文書化されている制約事項はありません。

## 前提条件

現在、DFSUARC0 ユーティリティーについて文書化されている前提条件はありません。

## 要件

現在、DFSUARC0 ユーティリティーについて文書化されている要件はありません。

## 推奨事項

現在、DFSUARC0 ユーティリティーについて文書化されている推奨事項はありません。

## 入力と出力

ログ保存ユーティリティーには、OLDS と SLDS の 2 種類の入力があります。このユーティリティーは、このユーティリティーのリリース・レベルと同じリリースの IMS で作成されたログ・データ・セットのみを受け入れます。

### OLDS 入力

入力に使用される OLDS は、正常にクローズされていなければなりません。入力 OLDS の RECON 内の状況は、'ARCHIVE NEEDED' でなければなりません。

1 つの OLDS 内でエラーが起こると、保存ジョブは終了します。ログ・リカバリー・ユーティリティーを実行してその OLDS をリカバリーし、ログ保存ユーティリティーを再実行してください。

IMS オンライン実行時に重複 OLDS が使用された場合、その両方がログ保存ユーティリティーへの入力として使用されます。1 次 OLDS 内でエラーが検出されると、保存ユーティリティーは 2 次 OLDS へ切り替えます。レコードが 2 次

OLDS 内にあれば、保存ジョブは続行されます。エラーが同じブロック内で検出された場合には、保存ジョブは終了します。ログ・リカバリー・ユーティリティを実行してその OLDS をリカバリーし、ログ保存ユーティリティを再実行してください。重複 OLDS の一方が利用できない場合、例えば、状況が 'ARCHIVE NEEDED' ではない場合には、利用可能な OLDS のみが入力として使用されます。利用不能な OLDS は無視されます。

重複 OLDS が入力として使用される場合に、1 次 OLDS の最初のブロックにエラーがあると、ログ保存ユーティリティは失敗して終了します。2 次 OLDS が正しい場合でも、両方の OLDS の最初のブロックにシーケンス・エラーが示されます。ログ保存ユーティリティは、1 次 OLDS の最初のブロックをアンカー・ポイントとして使用します。このブロックにエラーがあると、そのブロックから収集されたデータを 2 次 OLDS と比較して検査することができません。どちらかの OLDS の最初のブロックにエラーが存在している場合は、ログ・リカバリー・ユーティリティを実行して、その OLDS をリカバリーしてから、ログ保存ユーティリティを再実行してください。

複数の OLDS を入力 OLDS として指定する場合は、それらは連続して作成されていない必要があります。別々の IMS システム実行で作成された OLDS を、同時に入力にすることはできません。

入力内のいずれかの OLDS がリカバリー・ポイント (リカバリー・ポイントとは、OLDS を強制的に切り替える /DBRECOVERY および /DBDUMP コマンドを実行して得られるもの) で終了していると、保存ユーティリティは次のことを行います。

- SLDS と RLDS のうちの少なくとも 1 つが DASD に入っていれば、出力データ・セットはクローズされ、リカバリー・ポイントで終了した OLDS の処理後に、保存ジョブは終了します。処理されていない可能性のある残りの OLDS の状態は、まだ ARCHIVE NEEDED のままです。
- すべての SLDS と RLDS がテープに入っている場合、IMS は、すべての SLDS と RLDS にボリュームの終わりを強制し、保存ジョブは SLDS と RLDS の次のボリュームを使用して続行されます。

DBRC は、入力 OLDS を検査します。OLDS 指定にエラーがあると、ログ保存ユーティリティは終了して、エラー・メッセージが出されます。

### SLDS 入力

入力に使用される単一 SLDS は、IMS バッチ環境で作成された DASD 上の SLDS です。また、この SLDS は正常にクローズされていなければなりません。EXEC パラメーターに DBRC=NO が指定されていると、テープ SLDS 入力が許されます。保存済みの SLDS を使用して、これをもう一度保存することができます。しかし、これは本来の使用法ではありません。

複式 SLDS を入力として使用することができます。1 次 SLDS 内でエラーが検出されると、ログ保存ユーティリティは 2 次 SLDS に切り替えます。レコードが 2 次 SLDS 内にあれば、保存ジョブは続行されます。単一 SLDS 内にエラー、または複式 SLDS 内の同じブロックにエラーがあると、保存ジョブは終了します。ログ・リカバリー・ユーティリティを実行してその SLDS をリカバリーし、ログ保存ユーティリティを再実行してください。

入力、DBRC が存在している状態で作成された DASD SLDS からのものである場合には、ログ保存ユーティリティーは、新しい SLDS 情報で既存の SLDS レコードを更新するよう DBRC に知らせます。バッチ SLDS の保存ジョブ用の JCL を作成する必要があります。

## 出力

ログ保存ユーティリティーは、SLDS、オプションの RLDS、およびユーザー・データ・セットを出力として作成する他に、SYSPRINT にリストを作成します。

SYSPRINT には以下のものが入ります。

- 制御ステートメントのリスト
- チェックポイント・タイム・スタンプ ID のリスト
- 説明メッセージのリスト
- 保存結果のリスト

以下の図は、制御ステートメントの SYSPRINT リストの例です。

```
*****LOG ARCHIVE UTILITY CONTROL STATEMENT*****
SLDS -
  NOLOG(10,16,5F,67,69) FEOV(08000)
COPY DDNOUT1(DATASET1) -
  RECORD(OFFSET(5) FLDTYP(X) VALUE(16) FLDLEN(1) COND(E)) -
  RECORD(OFFSET(5) FLDTYP(X) VALUE(50) FLDLEN(1) COND(E)) -
  RECORD(OFFSET(5) FLDTYP(X) VALUE(51) FLDLEN(1) COND(E)) -
  RECORD(OFFSET(5) FLDTYP(X) VALUE(52) FLDLEN(1) COND(E))
EXIT NAME(UEXIT01)
```

以下の図は、チェックポイント・タイム・スタンプ ID のリストの例です。

```
USER CHECKPOINT RECORD - yyyy.ddd hh:mm:ss.t CHKPT-id region-id prg-name (v1)(v2)
SYSTEM CHECKPOINT RECORD - yyyy.ddd hh:mm:ss.t chkpt-id (v1)(v2) CHECKPOINT XXX (RESTART TTT)
```

チェックポイント・ログ・レコード (X'18' と X'4001') が検出されたときには、SYSPRINT リストにその手前の出力行の 1 行が印刷されます。日付、時刻、およびチェックポイント ID が両方について示されています。X'18' レコードについては、領域 ID とプログラム名が示され、X'4001' レコードについては、チェックポイント要求タイプが示されます。ここで、XXX は要求されたチェックポイントのタイプを英語で示したものです。出力 1 次 SLDS ボリュームのボリューム通し番号 (v1)、およびチェックポイントのコピー先となる 2 次 SLDS (複式出力の場合) も示されます。最初の X'4001' レコードについては再始動タイプも示されますが、TTT は実行された再始動のタイプを英語で示したものです。

以下の図は、説明メッセージの SYSPRINT リストの例です。

```
*** END OF VOLUME FORCED ON SLDS. PRIMARY(volser) SECONDARY(volser) ***
*** WRITE ERROR ON SLDS|USER|RLDS ddname ***
*** OUT-OF-SPACE on SLDS|USER|RLDS ddname ***
*** NO RECORD FOUND FOR SLDS|USER|RLDS ddname ***
```

以下の図は、保存結果の SYSPRINT リストの例です。

```
*** LOG ARCHIVE UTILITY (DFSUARCO)                **hh:mm yy.ddd **
      COPIED LOG RECORDS

FROM      DDNAME=ddname VOLSER=volser          DDNAME=ddname VOLSER=volser
          (for primary input)                  (for secondary input)
          .
          .
```

```

TO PRIMARY SLDS DSNAME=dsname
  VOLSER = volser volser volser .....
TO SECONDARY SLDS DSNAME=dsname
  VOLSER = volser volser volser .....

```

SLDS DOES NOT CONTAIN THE FOLLOWING LOG RECORDS:  
 'xx' 'xx' 'xx' 'xx' .....

```

TO PRIMARY RLDS DSNAME=dsname
  VOLSER = volser volser volser .....
TO SECONDARY RLDS DSNAME=dsname
  VOLSER = volser volser volser .....

```

## JCL 仕様

このログ保存ユーティリティーは z/OS バッチ・ジョブとして実行され、複数のログ保存ユーティリティー・ジョブを並行して実行できます。JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) が必要です。複式出力が要求されている場合は、SLDS は、1 次データ・セットと 2 次データ・セットから成ります。

### EXEC ステートメント

実行するユーティリティー、およびオプションの実行パラメーターを定義します。形式は次のとおりです。

```
//STEP EXEC PGM=DFSUARCO
          PARM= 'nnnn, DBRC=nnn, IMSPLEX=imsplex_name, DBRCGRP=xxx'
```

#### PARM=

サブシステム ID を示し、DBRC を指定するかどうかを示します。

#### nnnn

1 から 4 文字の長さの IMS サブシステム ID を示します。これは、入力データ・セットが OLDS の場合には指定する必要があります。これは、OLDS 内のデータを作成したオンライン IMS システムの IMSID と同じ値です。

#### DBRC=YES|NO

このユーティリティーのこの実行で DBRC を使用しないことを明示的に宣言するときは、DBRC=NO (または N) を指定します。

このユーティリティーのこの実行で DBRC を使用することを明示的に宣言するときは、DBRC=YES (または Y) を指定します。

DBRC= を指定しない場合、デフォルトの YES になります。

#### IMSPLEX=*imsplex\_name*

どの IMSplex DBRC を結合するのかわを示します。IMSPLEX= はオプション・パラメーターです。

#### DBRCGRP=*xxx*

DBRC グループで使用される RECON データ・セットで定義されている DBRC グループ ID を指定します。

### DD ステートメント

## STEPLIB DD

ログ保存プログラムが入っているプログラム・ライブラリーおよび任意のユーザー出口ルーチンを指定します。

## DFSOLPnn DD (1 次 OLDS 用)

## DFSOLSnn DD (2 次 OLDS 用)

入力に使用する OLDS を記述します。重複 OLDS を指定することができます。重複 OLDS の場合は、1 次 OLDS と 2 次 OLDS の接尾部が一致している必要があります。*nn* (接尾部) の値は 00 から 99 で、ログ・データがオンライン実行によって作成されたときに使用された DD 名と同じでなければなりません。入力として使用されるすべての OLDS は、オンライン実行時に連続して使用されていなければなりません。

DBRC=Y の場合は、DD ステートメント内で OLDS DD ステートメントを任意の順序で指定できます。

DBRC=N の場合は、OLDS DD ステートメントを作成された順序で指定する必要があります。また、重複 OLDS を一連の 1 次ペアおよび 2 次ペアとして指定する必要があります。

DCB BUFNO キーワードには、2 から 99 までのバッファ読み取りを指定することができます。

## DFSSLDSP DD (入力の 1 次 SLDS 用)

## DFSSLDSS DD (入力の 2 次 SLDS 用)

バッチ SLDS を指定します。任意に、ユーザーはバッチ SLDS に複式 SLDS を指定できます。入力に使用される SLDS と OLDS を同時に指定することはできません。2 から 99 個の読み取りバッファを指定できます。

## DFSSLOGP DD (出力の 1 次 SLDS 用)

## DFSSLOGS DD (出力の 2 次 SLDS 用)

出力に使用される SLDS を定義します。その形式は、使用する装置タイプによって異なります。SLDS が DASD 上にある場合は、保存されるログを収納するだけの十分なスペースを割り振る必要があります。SLDS ブロック・サイズを指定できますが、これは、入力データ・セットのブロック・サイズとは異なったものにすることができます。これを指定しなければ、入力データ・セットのブロック・サイズが使用されます。2 次 SLDS はオプションで、複式保存を指定します。入力がバッチ SLDS であり、DBRC が存在している状態でログ保存ユーティリティを実行する場合は、複式 SLDS レコードが DBRC に既に認識されている場合にのみ複式出力を作成できます。

複式 SLDS が作成される場合、それらは異なるブロック・サイズを持つことができます。しかし、FEOV が指定されていても、両方のデータ・セットのブロック・サイズが等しく、両方がテープに割り振られている場合以外は無視されません。テープを指定する場合は、そのラベルは標準ラベルである必要があります。2 から 99 個の書き込みバッファを指定できます。

制約事項: これらの DD ステートメントでは、JCL パラメーター FREE=CLOSE を使用しないでください。データ・セットは動的に割り振り解除され、FREE=CLOSE の使用は予測不能の結果を生じます。

## ddname DD (RLDS またはユーザー出力データ・セット、あるいはその両方用)

ユーザー・データ・セットまたはリカバリー・ログ・データ・セット (RLDS)、

あるいはその両方を定義します。データ・セットが DASD 上にある場合は、それにコピーされるレコードを収容するための十分なスペースを割り振る必要があります。データ・セットは、RECFM=VB で作成されます。ブロック・サイズを指定でき、それを入力データ・セットのブロック・サイズとは異なるものにすることができますが、最長レコードを入れることができるほど十分大きいものにする必要があります。これを指定しなければ、入力データ・セットのブロック・サイズが使用されます。複式データ・セットが作成される場合、それらのブロック・サイズを異なるものにすることができます。2 から 99 個の書き込みバッファを指定できます。

#### **SYSPRINT**

出力メッセージ・データ・セットを定義します。

#### **SYSUDUMP**

ダンプ・データ・セットを定義します。

#### **SYSIN DD**

制御ステートメントを指定します。

#### **RECON1 DD**

最初の DBRC (データベース・リカバリー管理) RECON データ・セットを定義します。この RECON1 データ・セットは、IMS 制御領域で使用される RECON1 データ・セットと同じでなければなりません。

#### **RECON2 DD**

2 番目の DBRC RECON データ・セットを定義します。この RECON2 データ・セットは、IMS 制御領域で使用される RECON2 データ・セットと同じでなければなりません。

#### **RECON3 DD**

RECON1 または RECON2 でエラーが検出されたときに使用するオプションの DBRC RECON データ・セットを定義します。この RECON3 データ・セットは、IMS 制御領域で使用される RECON3 データ・セットと同じでなければなりません。

動的割り振りを使用している場合は、これらの RECON データ・セットの DD 名 を使用しないでください。

### **ユーティリティー制御ステートメント**

制御ステートメントはすべてオプションです。制御ステートメントは、次の場合に使用します。

- ユーザー出口ルーチンを使用する場合
- RLDS を作成する場合
- 特定のレコードをユーザー・データ・セットに入れる場合
- 特定のレコードを SLDS へのコピーから除去する場合
- テープ出力ボリュームを強制的に重複して作成する場合

3 つのタイプの制御ステートメントがあり、各ステートメントは、命令コードとパラメーターから成っています。制御ステートメントの使用規則は次のとおりです。

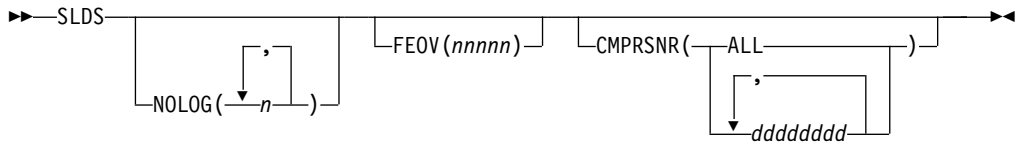
- 制御ステートメントは、1 から 72 桁目にフリー・フォームで入れることができます。パラメーターの指定順序は自由です。

- 命令コードとパラメーターはそれぞれ、ブランク、コンマ、またはコメントで区切らなければなりません。
- 1つの制御ステートメントに複数行を使用できます。継続文字 (+ および -) を、1桁目と72桁目の間にすることができます。(+)を使用する場合、行はブランクなしで連結されます。(-)を使用する場合、行はブランク付きで連結されます。
- どのパラメーターの値も、1組の括弧で囲まなければなりません。

### SLDS ステートメント

SLDS ステートメントは、SLDS に書き込まれないログ・レコード・タイプを指定します。さらに、テープ出力ボリュームについてボリュームの終わりが強制実行されることも指定します。これを省略すると、ログ・レコードはすべて SLDS にコピーされます。1つの SLDS 制御ステートメントだけしか指定できません。

SLDS 制御ステートメントの形式は次のとおりです。



### NOLOG

SLDS にコピーされないログ・レコード・タイプを定義します。NOLOG サブパラメーターの値は、16進数で指定する必要があります。例えば、SLDS NOLOG (19,1A,1B) のように指定します。

SLDS は、データベース・リカバリーとシステム再始動に必要なレコードを含んでいなければなりません。必要なレコード・タイプが省略すべきものとして指定されていると、ログ保存ユーティリティーは、エラー・メッセージを出して終了します。

### FEOV

重複出力テープ・ボリュームを指定します。このパラメーターは、複式テープ SLDS 環境でのみ必要です。これを指定すると、マルチボリューム・データ・セットの対応ボリュームに確実に同じレコードが含まれます (結果として、それらが交換可能になります)。

*nnnnn* は、テープ SLDS に書き込まれるブロック数を示します。これらのブロックが書き込まれるたびに、1次 SLDS と 2次 SLDS の両方に FEOV が出されます。ブロック番号は、5桁の10進数で指定されます。両方の SLDS のブロック・サイズが等しくない場合は、FEOV パラメーターは無視されます。

### CMPRSNR

リカバリー不能全機能データベース用のデータベース更新ログ・レコードのファイル属性指定を決定します。CMPRSNR を指定すると、ログ・レコードをアーカイブにコピーするデフォルトの動作がオーバーライドされます。

#### CMPRSNR(ALL)

ログ・レコードが圧縮されることを示します。レコードを圧縮すると、レコードが最小サイズのプレースホルダー・レコードに置き換えられます。

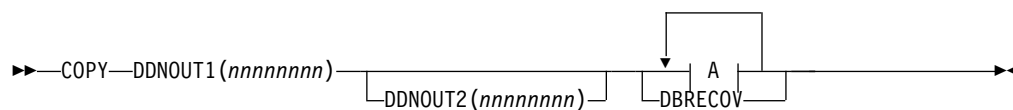


### CMPRSNR(dddddddd)

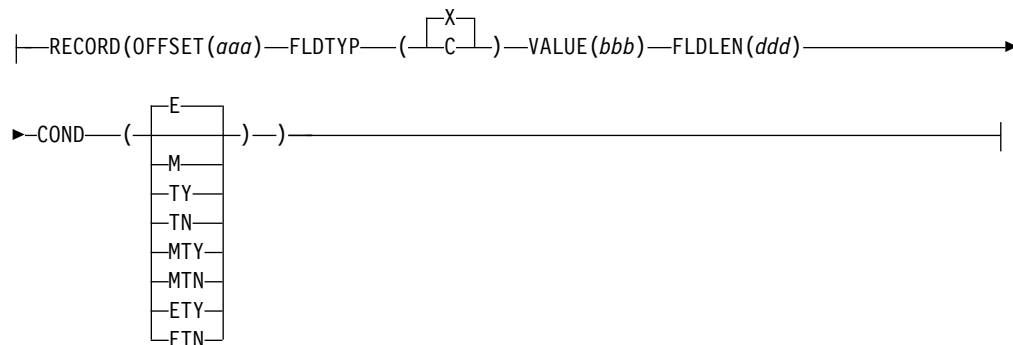
データベース名が *dddddddd* ストリングと一致する場合にのみログ・レコードが圧縮されることを示す 8 文字のストリング。

### COPY ステートメント

COPY ステートメントは、保存時にユーザー・データ・セットまたは RLDS を作成するときに使用します。COPY ステートメントの形式は次のとおりです。



### A:



COPY ステートメント内のキーワードの代わりに、下記の省略語を使用することができます。

キーワード

省略語

**OFFSET**

O

**FLDTYP**

T

**VALUE**

V

**FLDLLEN**

L

**COND**

C

**DDNOUT1**

**DDNOUT2**

データ・セットの DD 名を識別します。DDNOUT2 は、二重コピーが作成される場合にのみ必要です。DD ステートメントは、JCL に組み込む必要があります。nnnnnnnnn は DD 名の値です。

## RECORD

指定したデータ・セットに書き出すレコードを選択する際の条件を指定します。

### OFFSET(*aaa*)

レコードの中のテスト対象のフィールドの先頭を定義します。デフォルトは、レコードの位置 1 です。

*aaa* は、1 から、テストされるレコードの長さまでの範囲の値 (レコードの長さを含む) です。最大値は 32767 バイトです。論理レコード長を超えているかどうかを判別するための検査は実行されません。OFFSET キーワードに指定する値は、必ずバイト 1 からの相対位置で表されます。

### FLDTYP(*X*) | (*C*)

VALUE フィールド内のデータのタイプを定義します。X または C の値を指定する必要があります。

X は、データが 16 進文字のペアして処理されることを指定します。テスト・データは、パックされ (2 バイトが 1 つにされ)、同等の 16 進数にされます。X がデフォルトです。

C は、データを EBCDIC として処理することを定義します。

### VALUE(*bbb*)

FLDTYP(X) が指定されている場合は 16 進数で指定でき、FLDTYP(C) が指定されている場合は EBCDIC で指定できます。EBCDIC では、この値を引用符で囲んで指定します。文字ストリングが分離文字のブランクまたはコンマを含んでいる場合には、引用符の表記が必要です。引用符内には任意の文字を指定できます。(引用符内の二重引用符は、1 個の単一引用符を表します。) 負符号 (-) が最後の非ブランク文字であれば、値が次の行に継続されると見なされます。

制約事項: *bbb* の値は、255 個の EBCDIC 文字または 510 個の 16 進文字を超えることはできません。

このフィールドの長さは FLDLEN 値によって決まり、このフィールドの「非ヌル」文字の数で決まるわけではありません。

### FLDLEN(*ddd*)

テスト・フィールドから使用する文字数を定義します。

*ddd* は、VALUE に指定された文字数ではなく使用される実際のバイト数を表します。このフィールドの許容値の範囲は、1 から 255 です。デフォルトは 1 です。

### COND(*x*)

テストのタイプ、およびグループ内の他のテストとの関連を定義します。デフォルトは、COND(E) です。COND(*x*) を単独で指定することも、複数のオプションを結合することもできます。

**E** テスト・シリーズの最後の (または唯一の) エlementを示します。この後に現れるレコード制御ステートメントはいずれも、新しいシリーズのテストになります。これにより、各レコードに対して種々のテストを実行でき、各テスト・シリーズをレコード内の別々のフィールドに使用できます。

**M** これが複数フィールドのテストであることを示します。各入力レコード

に複数のテストが行われます。このシリーズ内のすべてのテストを満たしてからでなければ、このレコードの最終的な出力選択および処理を開始することはできません。

- T** これを指定すると、VALUE バイトが、比較フィールドではなく「マスクによるテスト」値として使用されます。VALUE フィールドの最初のバイト (FLDTYP(X) の場合は 2 つの 16 進文字) のみを使用されます。FLDTYP(C) を使用する場合は、EBCDIC 文字に相当する 16 進文字がテスト値です。このパラメーターを使用する場合は、FLDLEN キーワードを指定してはなりません。デフォルト長さ 1 が想定されます。
- Y** 「マスクによるテスト」のテスト・バイトの対応ビットごとに、レコード・テスト・フィールド中にビットがなければならぬことを示します。これは、「1 の場合分岐」テストに相当します。
- N** 「マスクによるテスト」のテスト・バイトに対応するビットに、レコード・テスト・フィールドのビットが立たないことを示します。これは、「ゼロの場合分岐」テストに相当します。
- MT** 複数フィールド・テストの特性で「マスクによるテスト」オプションを定義します。このパラメーターは、「マスクによるテスト」値で始まる複数フィールド・テストに使用する必要があります。
- ET** 複数フィールドのテスト・シリーズが、「マスクによるテスト」条件で終了することを表します。

#### **DBRECOV**

データベース・リカバリーに必要なすべてのログ・レコードを、指定した出力データ・セットにコピーします。この出力データ・セットは、DBRC に認識されており、DB リカバリーまたは変更累積用の JCL の作成時に作成された SLDS の代わりに GENJCL 処理で使用します。この出力データ・セットは、リカバリー・ログ・データ・セット (RLDS) です。DB リカバリーに必要なレコードがない場合には、RLDS はヌル・データ・セットです。この場合、DBRC は、RLDS DSNAME およびボリューム通し番号の代わりに、DSNAME およびボリューム通し番号を記録し、このヌル RLDS の代わりに SLDS を GENJCL 用として使用します。

DDNOUT1 は、COPY 制御ステートメントの必要パラメーターです。COPY 制御ステートメントには、必要なだけ RECORD パラメーターを指定できます。

RECORD パラメーターを指定しなければ、指定したデータ・セットにすべてのログ・レコードがコピーされます。

ある特定の COPY ステートメントでは、RECORD パラメーターと DBRECOV パラメーターを同時に指定することはできません。複数の COPY 制御ステートメントを指定できますが、DBRECOV パラメーターを指定した COPY ステートメントは 1 つしか許されません。

2 つの COPY ステートメントで、同じ出力データ・セットを指定してはなりません。

**EXIT** ステートメント

EXIT ステートメントでは、ユーザー出口ルーチンを使用することを指定します。

EXIT ステートメントの形式は次のとおりです。

▶—EXIT—NAME(nnnnnnnn)————▶

#### NAME(nnnnnnnn)

ユーザー出口のメンバー名を指定します。ユーザー出口ルーチンは、保存ユーティリティー・プログラムから LOAD でアクセスされます。できれば、JOBLIB または STEPLIB にバインドしてください。

複数の EXIT 制御ステートメントまたは複数の NAME パラメーターを指定することができます。

### 戻りコード

ログ保存ユーティリティーは、以下の戻りコードを戻します。

コード 意味

- 0 保存処理は正常に終了しました。
- 4 以下の一方または両方が起こると、この戻りコードが戻されます。
  - 保存処理は正常終了したが、すべての OLDS が保存されたわけではない場合。リカバリー・ポイントが検出され、ジョブが強制終了しました。残りの未保存の OLDS についてログ保存ユーティリティーを再実行してください。SYSPRINT メッセージを参照してください。
  - 保存ユーティリティーへの入力として指定された OLDS が、このジョブが実行されたときに既に保存されていた場合。SYSPRINT メッセージは、既に保存されていた OLDS がどれかを示します。
- 8 保存処理は失敗して終了しました。メッセージ DFS3263I または DFS3062I がその理由を示します。
- U3274 異常終了 - DBRC 内部障害です。メッセージ DFS3274I と種々の DSPxxxx メッセージがその理由を示します。

---

## DFSUARC0 ユーティリティーの例

以下の各例では、DFSUARC0 ユーティリティーで COPY 制御ステートメントを使用して RLDS と FEOV を作成し、SLDS での整合性を確保する方法を示します。

### 例 1

以下は、COPY 制御ステートメントを使用して RLDS を作成するログ保存ユーティリティーの JCL の例を示しています。

```
//ARCHIVE JOB MSGCLASS=A,CLASS=A,MSGLEVEL=(1,1)
//*
//ARC1 EXEC PGM=DFSUARC0,PARM='SYSA'
//STEPLIB DD DSN=IMS.&SYS2..SDFSRESL,DISP=SHR
/* COPY FROM 3 OLDS TO A SLDS */
/* RLDS AND A USER DATA SET ARE ALSO CREATED */
//DFSOLP00 DD DSN=OLP900,DISP=SHR,DCB=(BUFNO=20)
//DFSOLP01 DD DSN=OLP901,DISP=SHR,DCB=(BUFNO=20)
//DFSOLP02 DD DSN=OLP902,DISP=SHR
```

```

//DFSSLOGP DD DSN=SLDSP.D82001.N001,DISP=(,KEEP),
//          UNIT=TAPE,VOL=(,,99),LABEL=(,SL)
//RLSDDD1  DD DSN=RLDSP.D82001.N001,DISP=(,KEEP),
//          UNIT=TAPE,VOL=(,,99),LABEL=(,SL)
//USERDD1  DD DSN=USER.D82001.N001,DISP=(,KEEP),
//          UNIT=3350,VOL=USER01,SPACE=(CYL,5)
//RECON1   DD DSN=RECON1,DISP=SHR
//RECON2   DD DSN=RECON2,DISP=SHR
//RECON3   DD DSN=RECON3,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN    DD *
COPY DDNOUT1 (RLSDDD1) DBRECOV
/* THIS USER DATA SET CONTAINS          */
/* X'A5', X'A6', AND X'A7' LOG RECORDS    */
COPY DDNOUT1 (USERDD1) -
RECORD (0(5) T(X) V(A5) L(1) C(E)) -
RECORD (0(5) T(X) V(A6) L(1) C(E)) -
RECORD (0(5) T(X) V(A7) L(1) C(E))
EXIT NAME (UEXIT01)

```

## 例 2

以下は、SLDS 内の整合性を保つために FEOV を使用するログ保存ユーティリティ  
 一の JCL を示しています。

```

//ARCHIVE2 JOB MSGCLASS=A,CLASS=A,MSGLEVEL=(1,1)
//*
//ARC2 EXEC PGM=DFSUARC0,PARM='SYSA'
//STEPLIB DD DSN=IMS.&SYS2..SDFSRESL,DISP=SHR
//* COPY FROM 2 OLDS TO DUAL SLDS          */
//DFSOLP02 DD DSN=OLP902,DISP=SHR
//DFSOLP00 DD DSN=OLP900,DISP=SHR
//DFSOLS00 DD DSN=OLS900,DISP=SHR
//DFSOLS02 DD DSN=OLS902,DISP=SHR
//DFSSLOGP DD DSN=SLDSP.D82001.N001,DISP=(,KEEP),
//          UNIT=TAPE,VOL=(,,99),LABEL=(,SL)
//DFSSLOGS DD DSN=SLDSS.D82001.N001,DISP=(,KEEP),
//          UNIT=TAPE,VOL=(,,99),LABEL=(,SL)
//RECON1   DD DSN=RECON1,DISP=SHR
//RECON2   DD DSN=RECON2,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN    DD *
SLDS FEOV (08000)
/* THE SLDS ARE FORCED EOVS AFTER 8000 LOG BLOCKS */
/* ARE WRITTEN.                                     */
/*

```



---

## 第 21 章 ログ・マージ・ユーティリティー (DFSLTMG0)

ログ・マージ・ユーティリティー (DFSLTMG0) は、複数の IMS システムからのシステム・ログ・データ・セット (SLDS) をマージすることにより、1 つのデータ・セットを生成するために使用します。結果としてのデータ・セットは、ログ・トランザクション分析ユーティリティーへの入力として使用されます。

ログ・マージ・ユーティリティーは、レコード内のシステム・クロック値に基づいてログ・レコードを識別し、その後、それらのレコードを昇順でマージします。

ログ・マージ・ユーティリティーは、最大 9 個の IMS システム・ログをマージすることができます。それぞれのログは、同じ時間幅の間に実行された、一意的に識別される IMS システムの出力です。ログ・マージ・ユーティリティーへの入力の順序は、LOG01、LOG02、LOG03、...、LOG09 です。

DFSLTMG0 は、IMS システム定義時に、IMS.SDFSRESL の中に置かれます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 580 ページの『入力と出力』
- 582 ページの『JCL 仕様』

### 制約事項

共通キュー・サーバー (CQS) のログ・レコードが IMS ログ・レコードとは異なる形式を持つため、ログ・マージ・ユーティリティーは、CQS ログを入力として使用できません。

### 前提条件

現在、DFSLTMG0 ユーティリティーについて文書化されている前提条件はありません。

### 要件

現在、DFSLTMG0 ユーティリティーについて文書化されている要件はありません。

### 推奨事項

現在、DFSLTMG0 ユーティリティーについて文書化されている推奨事項はありません。

## 入力と出力

ログ・マージ・ユーティリティーへの入力は、最大 9 個の別々の IMS システムからのログ、および制御ステートメントから構成されています。単一システムからのログは、時間の順序で連結された一連のログから成っています。ログ・レコード間で互換性のあるシステム・クロック値が確実に生成されるようにするため、ログ・レコードは、同期した外部または内部クロックを持つプロセッサで実行されている IMS システムからのものであることが必要です。システム・クロック値は、時刻 (TOD) クロックとも呼ばれる 8 バイトのフィールドで、それぞれのログ・レコードの末尾に格納されます。

DFSLTMG0 は、START と STOP 制御ステートメントで指定された時刻の間に作成されたログ・レコードのマージ・データ・セットを、出力として生成します。この時刻は、協定世界時です (Universal Time Coordinated - UTC) です。

制約事項: マージ出力を、データベース・リカバリー・ユーティリティーへの入力として使用してはなりません。

### ログ・マージの制御

ログ出力を制御するには、次のようにします。

- 調べたい論理リンク・パスに関与している必要なシステムを選択します。
- ログ・トランザクション分析ユーティリティーを使用するときに調べたい必要なシステムからのログを選択します。
- 同じ時間スパンをカバーするように、各システムの一連の入力ログを調整します。
- 特定の時間間隔についてシステム間処理をサンプリングする必要がある場合は、ログ・マージ・ユーティリティーの制御ステートメントの開始および停止時刻を指定します。

開始日付 (ユリウス暦) と時刻の両方、または時刻のみを示すことができます。これらの時刻は、最初のシステム・ログ (LOG01 DD ステートメントで指定) に適用されます。他のログ・アクティビティーは、最初のシステム・ログの最初と最後のイベント間で発生する場合に収集されます。

- 特定タイプのログ・レコードのみをマージするときには、『ログ・レコードの選択』の項に記載されているキーワードを使って制御ステートメントで指定します。
- トランザクション分析ステップに適したログ・レコードを選択するために、MSG を指定します。(デフォルトではすべてのレコードになりますが、その場合は、ユーティリティー入力にいくつかのシステムの DL/I アクティビティーが含まれ、処理時間が延びることがあります。)

### 制御ステートメントの形式

#### START

開始時刻の指定に使用します。このステートメントは必須です。



表 28. START ステートメントの形式

位置	長さ	値
1	5	START
6	1	ブランク
7	可変	<p>yyddd,hhmsstt[{ + -}HHMM] ただし、hhmsstt の末尾桁はすべて省略できます。hhmsstt に続くオプションの時間帯情報には、次のものが入ります。</p> <p>+ または - 協定世界時 (UTC) からの時間帯のオフセットの符号を指定します。</p> <p><b>HH</b> UTC からのオフセットの総時間数を指定します。HH は 0 から 14 の数値です。</p> <p><b>MM</b> オフセットの分で指定します。MM は 00、15、30、45、またはブランクです。</p> <p>オプションの時間帯情報を指定する必要があるのは、夏時間への変更などのために、その日入力した UTC が現在のオフセットと異なる場合のみです。</p>

### STOP

停止時刻を指定する必要がある、その時刻は、LOG01 で指定した時刻フィールドからの相対的な時刻でなければなりません。

表 29. STOP ステートメントの形式

位置	長さ	値
1	4	STOP
5	1	ブランク

表 29. STOP ステートメントの形式 (続き)

位置	長さ	値
6	可変	<p>yyddd,hhmsstt[{ + -}HHMM] ただし、hhmsstt の末尾桁はすべて省略できます。hhmsstt に続くオプションの時間帯情報には、次のものが入ります。</p> <p>+ または - UTC からの時間帯のオフセットの符号を指定します。</p> <p><b>HH</b> UTC からのオフセットの総時間数を指定します。HH は 0 から 14 の数値です。</p> <p><b>MM</b> オフセットの分で指定します。MM は 00、15、30、45、またはブランクです。</p> <p>オプションの時間帯情報を指定する必要があるのは、夏時間への変更などのために、その日入力した UTC が現在のオフセットと異なる場合のみです。</p>

#### ログ・レコードの選択

この制御ステートメントは、特定タイプのログ・レコードのみをマージするために使用します。フォーマットはフリー・フォームで、1 桁目から始まります。次のリストにあるキーワードであれば、いずれも任意の組み合わせで使用できますが、構文上次の制限があります。

- キーワードの後ろにブランクを入れると、この制御ステートメントの処理は終了します。
- キーワードの後ろにコンマを入れると、この制御ステートメントの処理が継続されます。

キーワード  
意味

**ALL** ログ・レコードのすべてのタイプを選択します (これは、制御ステートメントがない場合のデフォルトです)。

**MSG** ログ・トランザクション分析ユーティリティ (DFSILTA0) に必要なすべてのログ・レコード (X'01'、 X'03'、 X'06'、 X'07'、 X'08'、 X'3x' シリーズ、 X'40'、 X'42'、 X'47') を選択します。

**3X** X'30' から X'3F' の範囲内のすべてのログ・レコードを選択します。

**XX** ログ・レコードのタイプ XX のものを選択します。

#### JCL 仕様

**EXEC** ステートメント

ログ・マージ・ユーティリティの DFSLTMG0 を実行します。

```
//STEP0 EXEC PGM=DFSLTMG0
```

## DD ステートメント

### STEPLIB DD

IMS.SDFSRESL を指します。ここには、IMS 中核と必要なアクション・モジュールが入っています。

```
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
```

### PRINT DD

制御ステートメントとエラー・メッセージ用の SYSPRINT データ・セットを指定します。

```
//PRINT DD SYSOUT=A
```

### LOG01 DD

最初の入力ログ・データ・セットを記述します。

```
//LOG01 DD DSN=IMS.LOGA,DISP=OLD,  
//          VOL=SER=XXXXXX,UNIT=TAPE
```

### LOG02 DD

2 番目の入力ログ・データ・セットを記述します。

```
//LOG02 DD DSN=IMS.LOGB,DISP=OLD,  
//          VOL=SER=XXXXXX,UNIT=TAPE
```

### LOGOUT DD

出力データ・セットを記述します。

```
//LOGOUT DD DSN=IMS.LOGOUT,DISP=(,PASS),  
//          VOL=SER=YYYYYY,UNIT=TAPE,  
//          DCB=(RECFM=VBS,LRECL=6000,BLKSIZE=6008)
```

### SYSIN DD

制御ステートメントのデータ・セットを記述します。

```
//SYSIN DD *  
START 75332,0830  
STOP 75332,1030  
MSG
```



---

## 第 22 章 ログ・リカバリー・ユーティリティー (DFSULTR0)

ログ・リカバリー・ユーティリティー (DFSULTR0) は、読み取りエラーを含んでいるログ・データ・セット、または正常に終了しなかったログ・データ・セットから使用可能なログ・データ・セットを生成するときに使用します。

OLDS とバッチまたはオンライン SLDS の両方を、ログ・リカバリー・ユーティリティーを使用してリカバリーすることができます。

このユーティリティーには、次の 4 つの操作モードがあります。

**CLS** 先行書き込みデータ・セット (WADS) から、または次 OLDS から OLDS をクローズします。

CLS モードでは、OLDS のみを処理します。SLDS をクローズするときには、DUP モードを使用します。CLS モードでは、ユーザー作成のロガー・ユーザー出口 (LOGWRT) があれば、ログ・リカバリー・ユーティリティーの実行時に呼び出されます。ロガー・ユーザー出口 (LOGWRT) は、初期設定呼び出しで 1 回、書き込まれるデータのログ・バッファごとの書き込み呼び出しで 1 回、そして終了呼び出しで 1 回呼び出されます。

制約事項: ログ・リカバリー・ユーティリティーは、複数の LOGWRT ユーザー出口の呼び出しをサポートしません。

**DUP** SLDS または OLDS を処理します。DUP モードでは、エラー ID レコードが入っている中間ログ、またはファイル終了マークが入っているクローズされたバッチ SLDS を作成します。

SLDS を安全にクローズするためには、DUP モードを実行してから REP モードを実行します。あるいは、ゼロ以外の ERRRC と、オリジナル・エラーの発生時に戻されたログ・シーケンス番号 (LSN) を用いて DUP モードを実行します。システムがメッセージ DFS616I を出すことがあり、これに障害発生時点の LSN が示されます。DFS616I が出ない場合は、DUP モードを実行してから REP モードを実行して、SLDS を安全にクローズする必要があります。

重要: 実稼働環境で SLDS をクローズするときに、LSN なしで DUP モードを実行してはなりません。ただし、REP モードも実行する場合は別です。LSN または REP モードと併用せずに DUP モードを使用すると、データが失われることがあります。

**REP** 中間ログを読み取り、エラー ID レコードをユーザー指定データで置き換えて、新しいログを作成します。

**PSB** OLDS と SLDS を混合したものから、「アクティブ PSB」報告書を生成することができます。

入力ログ・データ・セットの有効なデータ・セット属性は次のとおりです。

- RECFM=VB
- BLKSIZE は 8 より大きい

- LRECL は 5 以上で、しかも (BLKSIZE - 4) 以下

ログ・リカバリー・ユーティリティーは、以下のタイプのエラーを検出します。

- 入力ログ・データ・セットの読み取り時の入出力エラー
- ログ・レコードまたはログ・ブロック長のエラー
- ログ・レコード、ログ・ブロック、または OLDS 書き込みタイム・スタンプ内のシーケンス・エラー (OLDS リカバリーのみ)
- ログ・レコードのシーケンス・エラー (SLDS リカバリーのみ)

サブセクション:

- 『制約事項』
- 『前提条件』
- 587 ページの『要件』
- 587 ページの『推奨事項』
- 589 ページの『出力』
- 594 ページの『JCL 仕様』
- 597 ページの『ユーティリティー制御ステートメント』
- 600 ページの『戻りコード』

## 制約事項

DFSULTR0 ユーティリティーには下記の制約があります。

- 単一ロギングが使用され、しかも DBRC がアクティブな場合には、単一ログのみをログ・リカバリー・ユーティリティーへ入力として与えることができ、単一ログのみを DUP および REP モードからの出力として作成することができます。その他の場合には、DBRC は異常終了します。
- 重複ロギングが使用され、しかも DBRC がアクティブな場合には、重複ログのみをログ・リカバリー・ユーティリティーへの入力として与えることができます (PSB モードの場合は別です。これは単一ログ入力のみを受け入れます)。その他の場合には、DBRC RECON 更新は不適切になります。重複ログを入力として与える場合には、重複ログを、DUP および REP モードからの出力として作成する必要があります。1 次および 2 次 DSNAMES を DD ステートメントに正しく指定する必要があります。

## 前提条件

DFSULTR0 ユーティリティーを実行する前に、あらかじめ必要な一定のタスクまたは条件が満たされていることを確認してください。

OLDS および SLDS は、クローズしてからでないと、ユーティリティーへの入力として使用できません。

### OLDS のリカバリー

OLDS はクローズしてからでなければ、保存したり、任意のユーティリティーへの入力として使用することができません。使用中の OLDS は、通常シャットダウン時または緊急時再始動時に自動的にクローズされます。緊急時再始動が失敗したと

き、または書き込みエラーが検出されたために OLDS がクローズされないときは、ログ・リカバリー・ユーティリティーを使用してクローズしなければなりません。

RECON 内のゼロの停止時刻は、ログ・リカバリー・ユーティリティーを CLS モードで実行する必要があることを示します。これは、できる限り DUP の前に実行する必要がありますが、REP の後でも実行できます。

### SLDS のリカバリー

SLDS は、クローズしてからでなければ、ユーティリティーまたは IMS 再始動への入力として使用できません。ログ・リカバリー・ユーティリティーは、バッチ IMS システムによって作成された SLDS をクローズします。

### 要件

現在、DFSULTR0 ユーティリティーについて文書化されている要件はありません。

### 推奨事項

現在、DFSULTR0 ユーティリティーについて文書化されている推奨事項はありません。

### 入力

ログ・リカバリー・ユーティリティーは、単一ログと重複ログの両方を入力として使用します。このユーティリティーは、このユーティリティーのリリース・レベルと同じリリースの IMS で作成された入力ログ・データ・セットのみを受け入れません。

#### 単一ログの入力

CLS モードでは、ユーティリティーは次のことを行います。

1. 入力ログを読み取ります。
2. エラーがなければ、使用するログを作成します。
3. WADS を入力として使用するときは、アクティブ PSB の報告書を作成します。報告書の生成には入力ログのみが使用されるため、報告書が完全でない場合があります。PSB モードを使用することで、完全な報告書を生成できます。

DUP モードでは、ユーティリティーは次のことを行います。

1. 入力ログを読み取ります。
2. エラーがなければ、使用するログを作成します。
3. エラーを検出したときは、中間ログとエラー・リストを作成します。

DUP モードで作成された中間ログを使用する REP モードの場合、ユーティリティーは次のことを行います。

1. 中間ログを読み取ります。
2. 正しいブロックを出力ログにコピーします。
3. ユーザー指定の制御ステートメントに従って、エラー・ブロックを正しいブロックに置き換えます。
4. 使用可能ログを作成します。

PSB モードでは、ユーティリティーは次のことを行います。

1. 入力ログを読み取ります。
2. アクティブ PSB の報告書を作成します。

#### 重複ログの入力

以下の説明の中の用語、「1 次」および「2 次」は、重複ログ・データ・セットの 2 つのログを区別するためのものです。

CLS モードでは、ユーティリティーは次のことを行います。

1. 入力ログを読み取ります。
2. 両方の OLDS の同じ位置にエラーが検出されなければ、使用可能ログを作成します。エラーが一方の OLDS で (別のものはない) に検出された場合は、エラー・ブロック ID が NONE のエラー・リストが作成され、ユーティリティーは処理を続けます。その場合、作成された OLDS の対を IMS の再始動または保存 (これも OLDS の対の片方のみでのエラーを許容する) への入力として使用できることもありますが、エラーを除去するために、DUP モード処理が必要です。
3. WADS を入力として使用するとき、アクティブ PSB の報告書を作成します。報告書の生成には入力ログのみが使用されるため、報告書が完全でない場合があります。PSB モードを使用することで、完全な報告書を生成できます。

DUP モードでは、ユーティリティーは次のことを行います。

1. 1 次ログを読み取り、その内容を新しいシステム・ログにコピーします。エラー・ブロックを検出すると、DUP モードは、ログ・エラーを検出した 2 次ログに読み取り操作を位置付けます。それから DUP モードは 2 次ログを読み取り、その内容を同じ新しいシステム・ログにコピーします。今度は 2 次ログでエラーが検出されると (ただし、同じ位置ではない)、DUP モードは、前にエラーを検出した 1 次ログに読み取り操作を位置付けます。この処理は、完全に新しいシステム・ログが作成されるまで続きます。以下の図は、重複ロギングを使用する場合の DUP モードと REP モードを図示したものです。



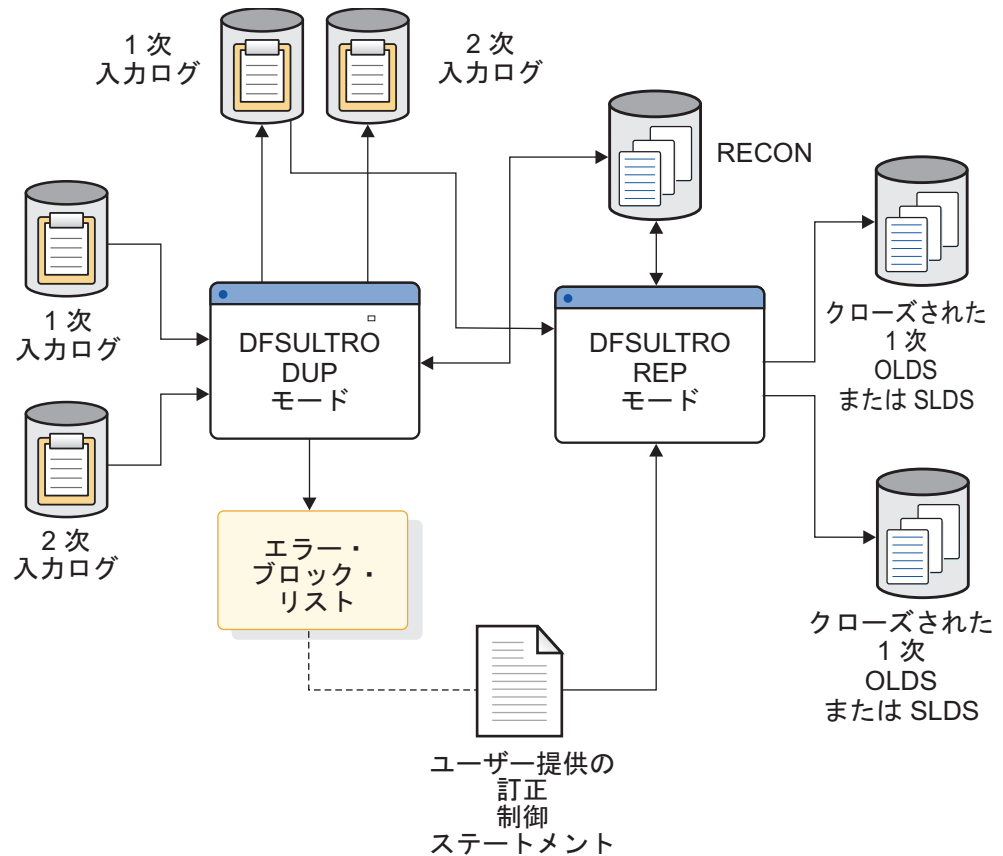


図 37. 重複ロギングを使用する場合の DUP モードと REP モード

2. 両方のログの同じ位置でエラーを検出した場合は、両方のエラー・ブロックを中間ログにコピーし、エラー・ブロックを一意的に識別します。中間ログ・データ・セットは、すべての有効なログ・ブロック、エラー・ブロック、およびエラー ID レコードを含みます。
3. REP モードで必要なユーザー指定制御ステートメントを作成するときに役立つように、文字形式と 16 進数形式でエラー・ブロックのリストを作成します。

#### 重複ログを入力として使用する REP モードの場合

1. DUP モードで作成された中間ログを読み取ります。
2. 正しいブロックをコピーします。
3. 制御ステートメントに従って、エラー・ブロックを正しいブロックに置き換えます。
4. 使用可能ログを作成します。

複式システム・ログ入力を使用していて、両方の入力ログの同じ位置にエラーがない場合は、DUP モードで作成されたログは正しいので、REP モードは必要ありません。

#### 出力

使用可能ログ、アクティブ PSB 報告書、および中間ログの他に、ログ・リカバリー・ユーティリティーは以下のものも作成します。

- 中間ログ・エラー ID レコード
- エラー・ブロックのリスト (SYSPRINT)
- REP モード検査メッセージ
- データ・レコードのダンプ

#### 中間ログ・エラー ID レコード

以下の図は、重複ログ入力から生成された中間ログのエラー ID レコードを図示したものです。この例では、1 次ログおよび 2 次ログの両方の BLK2 にエラーがあります。中間ログでは、最初のエラー ID は BLK2B を表し、2 番目のエラー ID は BLK2A を表します。REP モード時に、BLK2A または BLK2B が、制御ステートメントに基づいて正しいブロックに置き換えられます。さらにこの例では、REP モード実行後の有効なログを示しています。

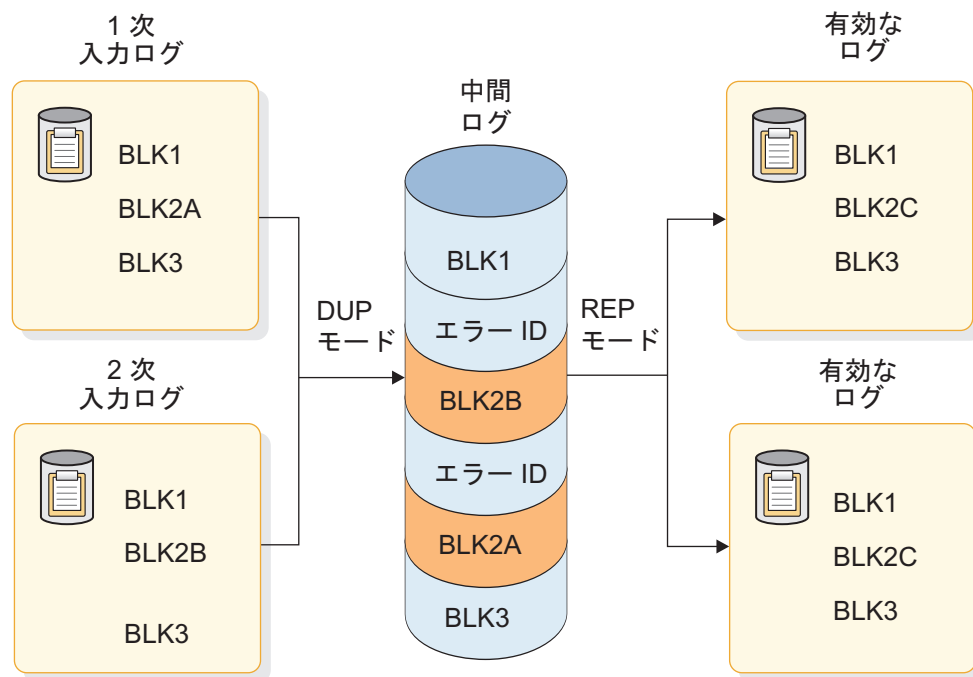


図 38. 中間ログのエラー ID レコード

#### エラー・ブロックのリスト (SYSPRINT)

エラー・ブロック・リストは、CLS モードおよび DUP モードの実行時に検出されたエラーを含みます。REP モードから生じた検証メッセージとその後のデータ・レコードのダンプも含みます。

次に示すのは、CLS モードおよび DUP モード両方のエラー・リストです。

```
pppppppppppppppppppp ON ddddddd BLOCK# bbbbbbb ** ERROR-ID=xnnnnn **
ssssssssssssssssss--gghhijj
```

エラー・ブロックのリストの各フィールドは次のとおりです。

**pppppppppppppppppppp**

エラーのタイプを示すメッセージ接頭語です。次のタイプのエラーを示します。

**PERMANENT I/O ERROR**

データ・チェックでも、長さエラーでも、連続データ・チェックでもないエラーが起こり、入力ログ用の SYNAD 出口に入りました。

**DATA CHECK**

データ・チェック・エラーで入力ログ用の SYNAD 出口に入りました。

**END-OF-DATA**

入力ログ用の EODAD 出口に入りました。これはエラーではなく、この入力データ・セットの処理が終了したことを示します。代替ログへのスワップが成功すると、処理は代替ログで続行されます。

**BLOCK LENGTH ERROR**

ブロック記述子ワード (BDW) 内の長さが有効ではありません。

**BLOCK TOD ERROR**

OLDS ブロック接尾部内の時刻 (TOD) が、昇順になっていません。

**BLOCK SEQ ERROR**

OLDS ブロック接尾部内のブロック・シーケンス番号が、昇順になっていません。

**RECORD LENGTH ERROR**

レコード RDW 内の長さが正しくありません。

**RECORD SEQ ERROR**

レコード・シーケンス番号が、昇順になっていません。

**ddddddd**

エラーが検出されたデータ・セットの DD 名です。次のリストに考えられる DD 名が示してあります。

**IEFRDER**

1 次入力 SLDS

**IEFRDER2**

2 次入力 SLDS

**DFSOLP**

1 次入力 OLDS

**DFSOLS**

2 次入力 OLDS

**bbbbbbb**

エラーのあるブロックの相対ブロック番号 (16 進数) です。ブロックは、最初の入力ボリュームの最初のブロックを 0000001 としてカウントを開始します。

- x A または B で、エラーが現行ログと代替ログのどちらで起きたかを表します。処理開始時には、1 次ログが現行ログで、2 次ログが代替ログです。エラーのために処理が代替にスワップされると、役割が逆になって、処理が続行されます。代替ログでのエラーは、必ず現行ログでのエラーの前に報告されます。

**nnnnn**

エラーを識別するシーケンス番号です。

xnnnnn は 'NONE' です。これは、重複 OLDS で CLS モード処理が、ある特定の位置で一方の OLDS (その他のものではない) にエラーを検出した場合で

す。これらの条件の下でエラーがリストされる理由は、たとえエラーを修正しないで OLDS を再始動や保存に使用できるとしても、エラーを修正するため DUP モードを使用を望むかもしれない状況をユーザーに警告することです。

#### SSSSSSSSSSSSSSSSSSSS

エラーをさらに識別するメッセージ接尾部です。この接尾部には次のものがあります。

#### ORIGINAL BDW X'ssss'

BDW 内の元のブロック長が正しくないため、変更されました。変数 *sss* は、16 進表記で表された元の値です。

#### RCD AT OFFST X'oooo'

ログ・レコードのレコード記述子ワード (RDW) 内に、無効な長さが入っています。変数 *oooo* は、ブロックの開始位置からエラーのある RDW までの、16 進数で表したオフセット (ゼロを基準にしたもの) です。

#### ffffffff TO tttttttt

ブロック・シーケンス、ブロック TOD、またはレコード・シーケンス・エラーが起きました。変数 *fffffff* は、最後の正しい値 (または正しい値と見なされるもの) です。変数 *ttttttt* は、エラー値です。シーケンス・エラーが起きた後は、次のブロック内の最初のレコード・シーケンス番号、ブロック・シーケンス番号、およびブロック TOD は正しいと見なされ、残りのレコードが検査される新しいシーケンスを開始します。ログ・リカバリー・ユーティリティは、正しいデータのシーケンスの切れ目を報告します。報告書を分析し、有効なデータと無効なデータを判別する必要があります。

- gg** ブランクまたは NS です。これは、複数の特殊接尾部値の最初のもので、NS は、複式 SLDS 入力の場合にのみ適用されます。2 つの入力ログは、同じブロックから開始しません。両方の入力ログに共通の最初のブロックが読み取られるまで、「代替」ログにスワップすること (または対応する出力データ・セットに書き込むこと) はできません。
- hh** ブランクまたは CE です。これは、特殊接尾部値の 2 番目のものです。CE は、これが連続エラーであることを表します。中間にある正しいブロックが読み取られないで、2 番目から *n* 番目のエラーが起きました。
- ii** ブランクまたは SA です。これは、特殊接尾部値の 3 番目のものです。SA は、代替ログが既にデータの終わり (END-OF-DATA) に達しているか、または永続入出力エラー (PERMANENT I/O ERROR) を検出したため、代替ログにスワップすることができないことを示します。
- jj** ブランクまたは SO です。これは、特殊接尾部値の最後のものです。SO は、代替ログへのスワップ時に、代替ログが END-OF-DATA に達したか、または PERMANENT I/O ERROR を検出したことを示します。この場合には、通常、処理は元の現行ログに戻ります。しかし、現行ログは既に END-OF-DATA に達しているか、PERMANENT I/O ERROR が検出されています。したがって、現行ログに戻ることができません。

#### REP モード検査メッセージ

REP モードでの処理時に、中間ログ・データ・セットのデータを正しく置き換えると、以下のメッセージが印刷されます。

```
DATA REPLACED IN RECORD Axxxxx ... replacement data text...
```

ただし、xxxxx はエラー ID です。

制御ステートメントの形式にエラーがあると、以下のメッセージが印刷されます。

```
ERROR IN CONTROL STATEMENT FORMAT ... text of control statement...
```

データ・レコードのダンプ

検査メッセージに続くデータ・レコードのダンプは、レコードの 16 進表示です。16 進表示は、データ・レコードの 1 印刷行につき 4 行で印刷されます。

- 最初の行には、エラーがあるブロック内の位置 (1 から開始) と、そのバイトの EBCDIC 表示が印刷されます。
- 2 番目の行には、各ログ・レコードの最初のバイト位置を示すアスタリスクが印刷されます。
- 3 番目の行には、ゾーン部分の表示のみが印刷されます。
- 4 番目の行には、数字部分の表示のみが印刷されます。

次の図は、印刷出力のフォーマットを示しています。

```
000001      q   RRE  b           // EBCDIC representation
              *                   // first byte of a log record
              2000020049 00DDC40809 // high-order hexadecimal digit
              00000D0008 029954024F // low-order hexadecimal digit
```

アクティブ領域メッセージ

CLS モードで WADS を指定した場合は、システム障害時のアクティブ PSB が印刷されます。障害時のアクティブ PSB ごとに 1 行が印刷されます。PSB にバックアウトが必要な場合は、出力の PSB 行の下にデータベース名がリストされます。次の図はこの出力のフォーマットを示しています。

\*\*\*\*\* RECOVERY REQUIREMENTS \*\*\*\*\*

<u>PSB NAME</u>	<u>RECOVERY TOKEN</u>	<u>DATABASE</u>	<u>DSID</u>	<u>ACTION</u>	<u>Required</u>
PPPPPPPP	EEEEEEEEHHHHHHHHHHHHHHHHHH				
		DDDDDDDD			
NNN	MMMMMMM			SSSSSSSS	

END OF REPORT

この報告書内のフィールドの意味は次のとおりです。

**PPPPPPPP**

PSB 名

**EEEEEEEE**

リカバリー・トークンの EBCDIC 部分

**HHHHHHHHHHHHHHHHHH**

8 バイト (16 文字) のリカバリー・トークンの 16 進数部分

**DDDDDDDD**

データベース名。

## SSSSSSSS

データベース名の状況。データベース名が DDDDDDDDD フィールド内にならない場合は、次のメッセージのいずれかが出されます。

No database names found (データベース名が検出されません)

DBNAME list may be incomplete (DBNAME リストが不完全な可能性があります)

NNN エリア・データ・セットを示す高速機能データ・セット ID

## MMMMMMMM

出されたメッセージ。次のメッセージのいずれかが出されます。

Backout is required (バックアウトが必要です)

Redo is required (やり直してください)

Databases are in doubt (データベースが正しいか疑問です)

アクティブ領域報告書は、PSB モードでも作成されます。

## JCL 仕様

DFSULTR0 を実行するには、次の JCL が必要です。

**EXEC** ステートメント

ログ・リカバリー・ユーティリティー (DFSULTR0) を呼び出します。次の形式でなければなりません。

```
//STEP EXEC PGM=DFSULTR0,PARM='IMSID=iiiiiii,  
//          DBRC=ddd, IMSPLEX=imsplex_name, DBRCGRP=xxx'
```

### **IMSID=iiiiiii**

入力 OLDS を作成したオンライン・システムの IMSID を指定します。

要件: CLS モードでは、IMSID= は必須です。OLDS 入力および DBRC=YES (指定された場合またはデフォルトとして使用される場合) の DUP モードでも、IMSID= が必須です。

IMSID= を指定しても、不要な場合には無視されます。

### **DBRC=YES|NO**

DBRC= のデフォルトを IMS システム定義時に IMSCTRL マクロで確立しないことを表します。

このユーティリティーのこの実行で DBRC を使用しないことを明示的に宣言するときは、DBRC=NO (または N) を指定します。

このユーティリティーのこの実行で DBRC を使用することを明示的に宣言するときは、DBRC=YES (または Y) を指定します。CLS モードでは DBRC=YES が必須です (デフォルトです)。DUP および REP モードでは DBRC=YES はオプションです。

推奨事項: DBRC をアクティブにして DUP モードを実行すると、REP モードも DBRC をアクティブにして実行する必要があります。

**IMSPLEX=imsplex\_name**

どの IMSplex DBRC を結合するのかが示します。IMSPLEX= はオプション・パラメーターです。

**DBRCGRP=xxx**

DBRC グループで使用される RECON データ・セットで定義されている DBRC グループ ID を指定します。

パラメーターのデフォルトを使用するときは、PARM フィールドのパラメーター (キーワードも含めて) を完全に省略しなければなりません。

入力パラメーターを何も指定しなければ、デフォルトは、IMSID= (指定されていない) および DBRC=YES になります。

**DD ステートメント**

DD ステートメントは、ログ・リカバリー・ユーティリティーの特定の実行に必要な場合にのみ使用されます。

## 制約事項:

- DFSOLP (および DFSOLS) DD ステートメントを使用して、OLDS 入力を指定してください。
- IEFRDER (および IEFRDER2) DD ステートメントを使用して、SLDS 入力を指定してください。
- IEFRDER (および IEFRDER2) DD ステートメントも入っている実行では、DFSOLP (および DFSOLS) DD ステートメントを指定してはなりません。
- DFSNOLP (および DFSNOLS) DD ステートメントも入っている実行では、DFSWADSn DD ステートメントを指定してはなりません。
- IEFRDER (および IEFRDER2) DD ステートメントも入っている実行では、DFSWADSn、DFSNOLP (および DFSNOLS)、またはこの組み合わせを指定してはなりません。
- DFSNOLP (および DFSNOLS) DD ステートメントも入っている実行では、DFSPOLP (および DFSPOLS) DD ステートメントを指定しないでください。
- キーワード NOWADS も入っている実行では、DFSWADSn DD ステートメントを指定しないでください。

**STEPLIB DD**

ログ・リカバリー・ユーティリティーのモジュールが入っている IMS.SDFSRESL を指します。

**SYSPRINT DD**

システム・メッセージ・データ・セットを定義します。

**SYSUDUMP DD**

ダンプ・データ・セットを定義します。

SYSUDUMP ステートメントは、このユーティリティーの末尾にある例には組み込まれていません。

**DFSOLP DD**

1 次、または唯一の、入力 OLDS を定義します。

**DFSOLS DD**

2 次入力 OLDS を定義します。重複 OLDS を使用する場合にのみ、このステートメントを含めてください。

**DFSWADS $n$  DD**

WADS データ・セットを定義します。ここで、 $n$  は 0 から 9 の値です。オンライン実行時に使用されるすべての WADS を指定できますが、障害時にオンライン・システムで使用中的のもののみが必要です。この DD ステートメントは、WADS から OLDS をクローズする際に必要です。オンライン・システムで WADS を使用中でなければ、DFSWADS $n$  DD ステートメントは使用されません。

**DFSNOLP DD**

1 次、または唯一の、次の OLDS を定義します。次の OLDS とは、OLDS で書き込みエラーが起きた直後に、オンライン IMS システムによって書き込まれる OLDS です。

**DFSNOLS DD**

2 次の次の OLDS を定義します。重複 OLDS を使用する場合にのみ、このステートメントを含めてください。

**DFSPOLP DD**

指定された、クローズされる OLDS の前に IMS オンライン・サブシステムが使用した 1 次 OLDS を定義します。前に OLDS がなければ、この DD ステートメントを使用してはなりません。

**DFSPOLS DD**

指定された、クローズされる OLDS の前に IMS オンライン・サブシステムが使用した 2 次 OLDS を定義します。前の重複 OLDS を使用する場合にのみ、このステートメントを含めてください。

**IEFRDER DD**

1 次、または唯一の、入力 SLDS を定義します。DUP モードのすべての入力 SLDS ログは、同じブロック・サイズにしてください。IEFRDER は、PSB モードでの OLDS ログと SLDS ログの連結を指定するときに使用します。ログの連結を指定するときは、ログの名前を昇順で与える必要があります。

**IEFRDER2 DD**

2 次入力 SLDS を定義します。重複ログを使用する場合にのみ、このステートメントを含めてください。データ・セットが不要な場合には、このステートメントを省略してください。DD DUMMY や DSNAME=NULLFILE を使用してはなりません。

**NEWRDER DD**

新しいログまたは中間ログ用の、1 次、または唯一の、出力データ・セットを定義します。

**NEWRDER2 DD**

新しいログまたは中間ログ用の 2 次出力データ・セットを定義します。DBRC がアクティブで、しかも重複ログが入力として使用される場合に、このステートメントは必要です。DBRC がアクティブでなければ、このステートメントは不要です。DD DUMMY や DSNAME=NULLFILE を使用してはなりません。



### RECON1 DD

最初の DBRC RECON データ・セットを定義します。動的割り振りを使用する場合は、このステートメントは不要です。

### RECON2 DD

2 番目の DBRC RECON データ・セットを定義します。動的割り振りを使用する場合は、このステートメントは不要です。

### RECON3 DD

RECON1 または RECON2 でエラーが検出されたときに使用するオプションの DBRC RECON データ・セットを定義します。この RECON データ・セットは、制御領域が使用する RECON データ・セットと同じでなければなりません。動的割り振りを使用する場合は、このステートメントは不要です。

### SYSIN DD

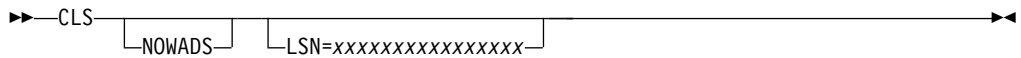
ログ・リカバリー入力制御ステートメントが入っている制御データ・セットを定義します。

## ユーティリティー制御ステートメント

ログ・リカバリー・ユーティリティーのユーティリティー制御ステートメントは、CLS モード、 DUP モード、 REP モード、または PSB モードのどれが使用されるかによって異なります。

CLS モード - WADS または NEXT OLDS からの OLDS のクローズ

この制御ステートメントの形式は次のとおりです。



### CLS

CLS モードであることを表します。

要件: CLS モードでは、DBRC が必要です。

WADS からクローズするときは、前の OLDS が使用可能であれば、前の OLDS に書き出された最後のブロック (ブロック・シーケンス番号は、OLDS の切り替え時に DBRC に渡され、RECON に保管される) の接尾部が取得されます。ブロックの接尾部を用いて、クローズされる OLDS のシーケンス検査の基準が確立されます。

WADS からクローズするときは、最初のエラーが検出されるか EOF があるかと、WADS から OLDS をクローズしようとしています。シーケンス・エラーが検出されると、CLS モードは失敗します。最初のエラーのブロックを含んでいるリストが生成されます。

次の OLDS からクローズするときは、次の OLDS の最初のブロックのシーケンス番号 (BSN) が判別されます。入力 OLDS でブロック BSN-1 が検出されると、入力 OLDS はクローズされます。ブロック BSN-1 が検出される前に EOF またはエラーが検出されると、CLS モードは失敗します。

### NOWADS

OLDS をクローズするときに、WADS の使用を抑止します。このキーワードを

使用するときは、JCL から DFSWADS<sub>n</sub> DD カードを除去する必要があります。そうでないと、ユーザー異常終了 U3271 が発生します。

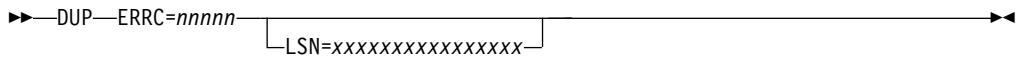
**重要:** WADS が使用不能な場合にのみ、NOWADS を使用してください。できれば、このキーワードは使用しないでください。ログ・レコードが失われ、データ安全性が損なわれる可能性があります、リカバリーが完全でないことがあります。

**LSN=xxxxxxxxxxxxxxxx**

DUP または CLS モードの処理で使用されるオプション・パラメーターです。入力ログで検出される必要のあるログ・シーケンス番号を指定します。ユーティリティーが成功していた (戻りコード 0 または 4) が、検出された最後のログ・シーケンス番号が xxxxxxxx より小さい場合には、ユーティリティーは戻りコード 8 で終了し、DBRC には正常終了は通知されず、メッセージ DFS3271I が出されます。xxxxxxxxxxxxxxxx の値は、16 文字の 16 進文字でなければなりません。

**DUP** モード - **OLDS** または **SLDS** のリカバリー (中間ログの作成)

この制御ステートメントの形式は次のとおりです。



**DUP**

DUP モードであることを表します。

**ERRC=nnnnn**

事前定義された数の入出力エラーまたはシーケンス・エラーが入力ログ・データ・セットで検出された後に、DUP モードを終了させるのに使用されます。*nnnnn* は、エラーの数 (00000 から 99999) を指定します。値を指定しないか、またはキーワードを省略すると、デフォルトは 99999 です。このフィールドには、先行ゼロ付きの 5 桁の数字を入れる必要があります。

*nnnnn* を 00000 に指定すると、最初のエラーが検出されたときに、DUP モードが終了し、中間ログがクローズされます。エラー ID レコードおよびエラー・ブロックは、中間ログに書き込まれません。REP モードは不要です。

ERRC=00000 を使用すれば、REP モードを実行しなくても、SLDS がクローズされます。最初のエラーのブロックを含むリストを生成することができます。最初のエラーが検出されると、最初のエラーの後の部分に新しいデータが存在しないことを確かめるために追加の検査が行われます。

**重要:** 実稼働環境で DUP ERRC=00000 を実行する場合は注意してください。これらの検査は安全ではないので、伴う危険性を明確に理解した上でのみ、ERRC=00000 を指定してください。例えば、正しいデータの間でログをクローズすると、正しいデータが破棄されることがあります。*nnnnn* の値を 00000 より大きくし、REP モードも実行する方が安全です。

ゼロより大きな ERRC 値を指定すると、EOF が検出されるか ERRC に達したときに DUP モードは終了します (ERRC は各ブロック読み取りの前に検査さ

れます)。エラーが検出されると、エラー ID レコードおよびエラー・ブロックが中間ログに書き込まれるため、REP モードが必要です。検出されたエラーを含むリストが生成されます。

OLDS または SLDS をリカバリーするには、ゼロより大きな ERRC 値を指定します。

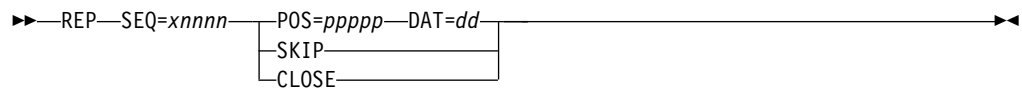
#### **LSN=xxxxxxxxxxxxxxxx**

DUP または CLS モードの処理で使用されるオプション・パラメーターです。入力ログで検出される必要のあるログ・シーケンス番号を指定します。ユーティリティーが成功していた (戻りコード 0 または 4) が、検出された最後のログ・シーケンス番号が xxxxxxxx より小さい場合には、ユーティリティーは戻りコード 8 で終了し、DBRC には正常終了は通知されず、メッセージ DFS3271I が出力されます。xxxxxxxxxxxx の値は、16 文字の 16 進文字でなければなりません。

#### **REP モード - OLDS または SLDS のリカバリー (新しいログの作成)**

このモードでは、DUP モードで作成された中間ログを読み取り、正しいブロックをコピーし、REP ステートメントに基づいてエラー・ブロックを正しいブロックに置き換えます。(1 次入力データ・セットのみが REP モード時に読み取られます。) 出力ログ・データ・セットは、新しい OLDS ログまたは SLDS ログです。最低 1 つの制御ステートメントが必要ですが、いくつでも含めることができます。

制御ステートメントの形式は次のとおりです。



#### **REP**

REP モードであることを表します。

#### **SEQ=xnnnnn**

変更するブロックの識別番号を示します。この番号は、DUP モードのリスト出力に示されます。

#### **POS=pppppp**

置き換えられるデータの開始位置を、1 からの相対位置で指定します。

#### **DAT=dd**

dd は、置換データを表す 2 から 50 個の 16 進文字 (0 から 9、A から F) を指定します。

#### **SKIP**

出力ログにこのデータ・ブロックを入れないことを指示します。

#### **CLOSE**

このエラー・ブロックの直前で出力ログをクローズすることを指示します。

REP モードの CLOSE オプションと、CLS モードを使用して WADS または次の OLDS から OLDS をクローズする処理を混同しないようにしてください。

REP ステートメントを使用するときには、次の規則が適用されます。

- 少なくとも制御ステートメントを 1 つ用意しなければなりません。
- 前のブロックでログがクローズされていない限り、DUP モード出力に記載されている各エラー・ブロックに、少なくとも 1 つ制御ステートメントを用意しなければなりません。
- 複数の REP ステートメントを用意するときは、識別番号 (SEQ=) をブロック番号の昇順にしなければなりません。
- データが正しくても、ブロックがエラーとして識別される場合は、そのブロック用に 1 つの制御ステートメントを与える必要があります。正しいブロックの最初の 4 バイトを、既存のデータに置き換えてください。これは、通常、入出力エラーの後の最初のブロックの場合です。
- DUP モードで重複ログを使用している場合は、Annnnnn または Bnnnnnn の 2 つのエラー・ブロックのうち的一方のみにステートメントを用意してください。選択されなかったブロックは無視され、出力ログには書き出されません。
- リカバリー中のログが、WADS または次の OLDS から正しくクローズされていない OLDS である場合は、REP モードの出力を入力として使用して、ログ・リカバリー・ユーティリティを CLS モードで再実行する必要があります。

#### PSB モード — アクティブ PSB の報告書の印刷

このユーティリティの前回の実行で以下のメッセージが出ているときは、PSB モードを使用します。

```
DFS3272I X'47' LOG RECORD NOT FOUND.
ACTIVE PSB MESSAGES NOT GENERATED.
```

アクティブ領域メッセージを入手するときは、ログ・リカバリー・ユーティリティを PSB モードで再実行する必要があります。PSB モードは、どの PSB がアクティブかを判別するために、いつでも使用できます。

PSB モードは、オープンされている OLDS については、実行してはなりません。生成されるリストは不完全なものになります。

この制御ステートメントの形式は次のとおりです。

▶▶—PSB—————▶▶

#### PSB

PSB モードであることを表します。

#### 戻りコード

ログ・リカバリー・ユーティリティは、以下の戻りコードを戻します。

コード 意味

- 0 正常終了。CLS モードを実行して、WADS で OLDS を終了する場合は、エラー・メッセージを無視してください。
- 4 報告書が不完全である可能性があります。この状況は、ログ・リカバリーで特定のログ・レコードが検出されなかった場合に発生します。

ログ・リカバリーで 47 レコードが全部揃って検出されなかった場合は、メッセージ DFS3272I が出力され、報告書には ACTIVE PSB MESSAGES NOT GENERATED と出力される以外に出力が生成されません。これを示しているのが次のメッセージです。

```
DFS3272I X'47' LOG RECORD NOT FOUND.  
ACTIVE PSB MESSAGES NOT GENERATED.
```


ログ・リカバリーで PSB のリカバリー単位に対する 5607 レコードが検出されなかった場合は、そのリカバリー単位の DB 状況に DBNAME LIST MAY BE INCOMPLETE と出力されます。

どちらの場合も、以前のログ・データを取り込んで、必要なレコードが含まれるようにする必要があります。CLS モードを実行していた場合は、PSB モードを実行して最後の OLDS に加えてすべてのログを取り込む必要があります。

- 8 正常終了していません。問題がログのリリース・レベルとユーティリティーのリリース・レベルの不一致による場合は、このエラー・コードにはメッセージ DFS3062I が伴います。

後続のジョブ・ステップの EXEC ステートメントで COND= パラメーターを用いれば、これらの戻りコードを調べることができます。

関連資料:

 [ロガー・ユーザー出口 \(LOGWRT\) \(出口ルーチン\)](#)

---

## DFSULTR0 ユーティリティーの例

以下の各例では、DFSULTR0 ユーティリティーを使用して OLDS または SLDS をさまざまなモードでリカバリーする方法を示します。

### 例 1

次の例は、CLS モードで、WADS から OLDS をクローズする方法を示したものです。入力データ・セットは、同じ場所でクローズされます。DBRC RECON データ・セットは、更新されて「close time」になります。

```
//EXAMPL01 JOB .....  
//*  
//DFSULTR0 EXEC PGM=DFSULTR0,PARM='IMSID=iiiiiii'  
//*  
//* NOTE - IMSID= is required  
//* NOTE - Defaults are DBRC=YES  
//* NOTE - DBRC=NO is not valid.  
//*  
//SYSPRINT DD SYSOUT=A  
//DFSOLP DD ..... Primary OLDS to be closed  
//DFSOLS DD ..... Secondary OLDS to be closed  
//DFSPOLP DD ..... Primary prior OLDS  
//DFSPOLS DD ..... Secondary prior OLDS  
//DFSWADSn DD ..... WADS used by on-line system  
//RECONn DD ..... DBRC RECON data set(s)  
//* (can be dynamically allocated)  
//SYSIN DD *  
CLS
```

入力 OLDS または前の OLDS が作成されたときに WADS を使用していなかった場合は、DFSWADSn DD ステートメントを除去し、制御ステートメントに NOWADS キーワードを追加してください。

前の OLDS を使用できない場合は、DFSPOLP (および DFSPOLS) DD ステートメントを除去してください。

## 例 2

次の例は、CLS モードで、次の OLDS から OLDS をクローズする方法を示したものです。入力データ・セットは、同じ場所でクローズされます。DBRC RECON データ・セットは更新され、フラグはオフにされます。

```
//EXAMPL02 JOB .....
/*
//DFSULTR0 EXEC PGM=DFSULTR0,PARM='IMSID=iiiiiii'
/*
/* NOTE - IMSID= is required
/* NOTE - Defaults are DBRC=YES
/* NOTE - DBRC=NO is not valid
/*
//SYSPRINT DD SYSOUT=A
//DFSOLP DD ..... OLDS to be closed from next-OLDS
//DFSNOLP DD ..... next-OLDS
//RECONn DD ..... DBRC RECON data set(s)
/* (can be dynamically allocated)
//SYSIN DD *
CLS
```

## 例 3

次の例は、OLDS のリカバリーの 2 ステップのうちの最初として DUP モードを使用する方法を示したものです。入力データ・セットは、中間データ・セットにコピーされます。中間ログ・レコードは、DBRC RECON 内に作成されます。

```
//EXAMPL03 JOB .....
/*
//DFSULTR0 EXEC PGM=DFSULTR0,PARM='IMSID=iiiiiii'
/*
/* NOTE - IMSID= is required
/* NOTE - Defaults are DBRC=YES
/*
//SYSPRINT DD SYSOUT=A
//DFSOLP DD ..... Primary OLDS to be recovered
//DFSOLS DD ..... Secondary OLDS to be recovered
//NEWRDER DD ..... Primary interim data set
//NEWRDER2 DD ..... Secondary interim data set
//RECONn DD ..... DBRC RECON data set(s)
/* (can be dynamically allocated)
//SYSIN DD *
DUP ERRC=nnnn
```

ゼロより大きい ERRC 値を指定した場合 (デフォルトは 99999)、エラー・ブロックは出力データ・セットに書き込まれ、エラーのブロックについてのリストが生成されます。エラーを訂正するため、そしてエラー・ブロックを除去するためには、REP モードが必要です。エラーが検出されず、実行が正常に行われる場合は、REP モードは不要です。

ERRC=00000 を指定した場合は、DFSOLP (および DFSOLS) で EOF または最初のエラーが検出されると、NEWRDER (および NEWRDER2) はクローズされま

す。実行が正常に終了する場合は、REP モードは不要です。実行が正常に終了しない場合は、ゼロより大きな ERRC 値を指定して DUP モードを再実行しなければならず、REP モードが必要です。

リカバリー中のログが、WADS または次の OLDS から正しくクローズされていない OLDS である場合は、REP モードの出力 (または、エラーが検出されなかった場合は DUP モードの出力) を入力として使用して、ログ・リカバリー・ユーティリティーを CLS モードで再実行しなければなりません。

#### 例 4

次の例は、OLDS のリカバリーの 2 ステップのうち 2 番目のものとして REP モードを使用する方法を示したものです。入力データ・セットは、新しい OLDS にコピーされます。コピー処理時に、REP 制御ステートメントの指示に従って、エラー・ブロックが除去され、エラーのブロックが訂正されます。DBRC RECON 内の中間データ・セット情報は削除されます。DBRC RECON 内の元の OLDS 情報は、出力データ・セット情報で置き換えられます。

```
//EXAMPL05 JOB .....
/*
//DFSULTR0 EXEC PGM=DFSULTR0,PARM='IMSID=iiiiiii'
/*
/* NOTE - IMSID= is required
/* NOTE - Defaults are DBRC=YES
/*
//SYSPRINT DD SYSOUT=A
//DFSOLP DD ..... Primary interim data set
//DFSOLS DD ..... Secondary interim data set
//NEWRDER DD ..... Primary recovered OLDS
//NEWRDER2 DD ..... Secondary recovered OLDS
//RECONn DD ..... DBRC RECON data set(s)
/* (can be dynamically allocated)
//SYSIN DD *
REP SEQ=A00001 POS=000018 DAT=83 (EXAMPLE ONLY)
REP SEQ=A00002 SKIP
REP SEQ=A00003 CLOSE
```

リカバリー中のログが、WADS または次の OLDS から正しくクローズされていない OLDS である場合は、REP モードの出力を入力として使用して、ログ・リカバリー・ユーティリティーを CLS モードで再実行する必要があります。

#### 例 5

次の例は、SLDS のリカバリーの 2 ステップのうち最初のものとして DUP モードを使用する方法を示したものです。入力データ・セットは、中間データ・セットにコピーされます。中間ログ・レコードは、DBRC RECON 内に作成されます。

```
//EXAMPL04 JOB .....
/*
//DFSULTR0 EXEC PGM=DFSULTR0
/* (PARAM NOT REQUIRED - SEE NOTES BELOW)
/*
/* NOTE - IMSID= is ignored
/* NOTE - Defaults are DBRC=YES
/*
//SYSPRINT DD SYSOUT=A
//IEFRDER DD ..... Primary SLDS to be recovered
//IEFRDER2 DD ..... Secondary SLDS to be recovered
//NEWRDER DD ..... Primary interim data set
//NEWRDER2 DD ..... Secondary interim data set
```

```
//RECONn DD ..... DBRC RECON data set(s)
//*      (can be dynamically allocated)
//SYSIN DD *
DUP ERRC=nnnn
```

ゼロより大きい ERRC 値を指定した場合 (デフォルトは 99999) は、エラー・ブロックが出力データ・セットに書き込まれ、エラーのブロックについてのリストが生成されます。エラーを訂正するため、そしてエラー・ブロックを除去するためには、REP モードが必要です。エラーが検出されず、実行が正常に行われる場合は、REP モードは不要です。

## 例 6

次の例は、SLDS のリカバリーの 2 ステップのうち 2 番目のものとして REP モードを使用する方法を示したものです。入力データ・セットは、新しい SLDS にコピーされます。コピー処理時に、REP 制御ステートメントの指示に従って、エラー・ブロックが除去され、エラーのブロックが訂正されます。DBRC RECON 内の中間データ・セット情報は削除されます。DBRC RECON 内の元の SLDS 情報は、出力データ・セット情報で置き換えられます。

```
//EXAMPL06 JOB .....
//*
//DFSULTR0 EXEC PGM=DFSULTR0
//*      (PARAM NOT REQUIRED - SEE NOTES BELOW)
//*
//* NOTE - IMSID= is ignored
//* NOTE - Defaults are DBRC=YES
//*
//SYSPRINT DD SYSOUT=A
//IEFRDER DD ..... Primary interim data set
//IEFRDER2 DD ..... Secondary interim data set
//NEWRDER DD ..... Primary recovered SLDS
//NEWRDER2 DD ..... Secondary recovered SLDS
//RECONn DD ..... DBRC RECON data set(s)
//*      (can be dynamically allocated)
//SYSIN DD *
REP SEQ=A00001 POS=000018 DAT=83 (EXAMPLE ONLY)
REP SEQ=A00002 SKIP
REP SEQ=A00003 CLOSE
```

## 例 7

次の例は、メッセージ DFS3272I X'47' LOG RECORD NOT FOUND を受け取った後に、「アクティブ PSB」のリストを生成する方法を示したものです。ACTIVE PSB MESSAGES NOT GENERATED.PSB モードが使用されます。

```
//EXAMPL07 JOB .....
//*
//DFSULTR0 EXEC PGM=DFSULTR0
//*
//* NOTE - IMSID= is ignored
//* NOTE - DBRC=YES is not valid
//* NOTE - Defaults are DBRC=NO
//*
//SYSPRINT DD SYSOUT=A
//*
//* NOTE - The first log data set in the IEFRDER DD statement should
be the latest log data set containing the X'47' record.
//*
//IEFRDER DD ..... next or prior OLDS or SLDS
//      DD ..... next or prior OLDS or SLDS
//      DD ..... next or prior OLDS or SLDS
```



```

:
//          DD ..... latest OLDS or SLDS
//*
//*
//SYSIN    DD *
PSB

```

入力ログは、作成された順に連結しなければならず、しかもログ・レコードの内容がオーバーラップしていたり、抜けていたりしてはなりません。

## 例 8

次の例は、入力ログの連結 (OLDS と SLDS) から、「アクティブ PSB」のリストを作成する方法を示したものです。PSB モードが使用されます。

```

//EXAMPL08 JOB .....
//*
//DFSULTR0 EXEC PGM=DFSULTR0
//*
/* NOTE - IMSID= is ignored
/* NOTE - DBRC=YES is invalid
/* NOTE - Defaults are DBRC=NO
/*
//SYSPRINT DD SYSOUT=A
//IEFRDR   DD ..... OLDS or SLDS
:
:
:          DD ..... OLDS or SLDS
:
//          DD ..... OLDS or SLDS
/*
/*          (see note below)
//SYSIN    DD *
PSB

```

要件: 入力ログは、作成される順序で連結する必要があります。OLDS と SLDS が混合している場合は、ログ・レコードの内容にオーバーラップする部分があってはなりません。

## 例 9

次の例では、DUP モードおよび ERRRC=00000 で、IMS バッチによって作成された SLDS をクローズする方法を示します。入力データ・セットは、出力データ・セットにコピーされ、出力データ・セットでクローズされます。DBRC RECON 内の入力 SLDS 情報は、出力データ・セット情報で置き換えられます。

```

//EXAMPL09 JOB .....
//*
//DFSULTR0 EXEC PGM=DFSULTR0
/*          (PARAM NOT REQUIRED - SEE NOTES BELOW)
/*
/* NOTE - IMSID= is ignored
/* NOTE - Defaults are DBRC=YES
/*
//SYSPRINT DD SYSOUT=A
//IEFRDER  DD ..... Primary SLDS to be closed
//IEFRDER2 DD ..... Secondary SLDS to be closed
//NEWRDER  DD ..... Primary output SLDS
//NEWRDER2 DD ..... Secondary output SLDS
//RECONn   DD ..... DBRC RECON data set(s)
/*          (can be dynamically allocated)
//SYSIN    DD *
DUP ERRRC=00000

```

ERRC=00000 を指定した場合は、IEFRDER (および IEFRDER2) で EOF または最初のエラーが検出されると、NEWORDER (および NEWORDER2) はクローズされます。実行が正常に終了する場合は、REP モードは不要です。実行が正常に終了しない場合は、ゼロより大きな ERRC 値を指定して DUP モードを再実行しなければならず、REP モードが必要です。

入力 SLDS (IEFRDER, IEFRDER2) がマルチボリューム・テープ・データ・セットである場合は、最後のボリュームのみを DD ステートメントに指定する必要があります。また、出力 DD ステートメント (NEWORDER, NEWORDER2) 上のデータ・セット名 (DSN) は、入力と同じにしてください。実行が正常に終了する場合は、DBRC RECON 内のボリューム情報のみが置き換えられます。ERRC=00000 が必要です。

---

## 第 5 部 サービス・ユーティリティー

サービス・ユーティリティーは、IMS の保守や操作機能を実行するために使用します。

各トピックでは、ユーティリティーがどのように機能するのかを説明し、その使用における要件と制約事項を定義しています。また、例を提供しています。



---

## 第 23 章 バッチ SPOC ユーティリティー (CSLUSPOC)

バッチ SPOC (単一制御点) ユーティリティーは、IMSplex のメンバーに IMS オペレーター・コマンドをサブミットするために使用します。

バッチ SPOC ユーティリティーでは、プログラム・パラメーターは IMSplex 環境を定義し、ユーティリティーへの入力に IMS オペレーター・コマンドです。このユーティリティーからの出力は、SYSPRINT ファイルに書き込まれるコマンド応答です。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 610 ページの『入力と出力』
- 611 ページの『JCL 仕様』
- 612 ページの『戻りコード』

### 制約事項

バッチ SPOC ユーティリティーでは、OM API がサポートしているコマンドを使用する必要があります。それらのコマンドはすべてタイプ 2 コマンドであり、タイプ 1 コマンドのみのサブセットです。サポートされるコマンドについては、OM API でサポートされるコマンドとキーワード (コマンド)を参照してください。

シーケンス番号がない固定長レコードを使用した SYSIN 入力では、レコードの最後の 8 バイトにすべての数値パラメーターを使用しないでください。そのパラメーターはシーケンス番号と見なされ、切り捨てられるからです。

可変長レコードを使用した SYSIN 入力では、シーケンス番号は許可されません。

### 前提条件

現在、バッチ SPOC ユーティリティーについて文書化されている前提条件はありません。

### 要件

バッチ SPOC ユーティリティーは、IMSplex 環境でのみ機能します。

### 推奨事項

現在、バッチ SPOC ユーティリティーについて文書化されている推奨事項はありません。

## 入力と出力

バッチ SPOC ユーティリティの入力は、SYSIN DD ステートメントです。SYSIN DD ステートメントの例は次のとおりです。

```
//SYSIN DD *
  QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)
```

ユーザーは、実行したいコマンドが入っている SYSIN ファイルを提供します。コマンドは、順次に行われます。1 つのコマンドが完了すると次のコマンドが実行され、SYSIN ファイル内のすべてのレコードが処理されるまで、それが続きます。SYSIN 制御ステートメントの継続は、行の最後の非空白文字が正符号 (+) または負符号 (-) であることによって指定されます。正符号は次行から先行スペースを除去し、負符号は先行スペースを保持します。

SYSIN ファイルの中には、コメントを組み込むことができます。コメントには、以下の構造を使用します。

- コメント行で、コメントを 1 桁目のアスタリスク (\*) で開始し、同じ行にコマンドを入力しません。例えば、次のようになります。

```
* this is a comment
```

コメントのみの行の場合、バッチ SPOC ユーティリティはそのコメント行を処理のために OM に送信しません。

- コマンドと同じ行では、コメントは 『/\*』 で開始し、『\*/』 で終了します。以下に例を示します。

```
/* this is a comment */
```

以下の位置にあるコメントがサポートされます。

```
QRY IMSPLEX SHOW(ALL) /* This is a comment */
```

以下の位置にあるコメントはサポートされません。

- コメントが同じ行のコマンドの前にある場合。例えば、次のとおりです。

```
/* This is a comment */ QRY IMSPLEX SHOW(ALL)
```

- コメントが同じ行のコマンドの中央にある場合。例えば、次のとおりです。

```
QRY /* This is a comment */ IMSPLEX SHOW(ALL)
```

- コメントが 1 つのブロックの複数の行にまたがる場合。例えば、次のとおりです。

```
/* This is comment line number 1,
  this is comment line number 2 */
```

コメントが同じ行のコマンドの後にある場合、バッチ SPOC ユーティリティは入力ストリング全体を Operations Manager (OM) に送信し、コメントは IMS によって解析されます。

バッチ SPOC ユーティリティの出力は、SYSPRINT ファイル内に置かれたフォーマット済みコマンド応答です。

複数のコマンドを発行した場合、応答はコマンドが SYSIN ファイル内で指定されたのと同じ順序で表示されます。デフォルトのレコード長は 133 です。コマンド応答は TSO SPOC 表示によく似た形式でフォーマット設定されます。長すぎるレコ

ードは、次の行へ折り返されます。SYSPRINT ファイル内で長いレコードを許可するために、DCB 情報を JCL またはデータ・セット割り振りの中で指定できます。

## JCL 仕様

バッチ SPOC ユーティリティは、標準 JCL ステートメントを使用して呼び出されます。

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//SPOC      EXEC PGM=CSLUSPOC,PARM=('IMSPLEX=plex,ROUTE=sysid,WAIT=05:00,F=option')
```

パラメーター・キーワード

### IMSPLEX

IMSpIex の名前に 1 から 5 文字の接尾部を指定する必須パラメーター

**F** SPOC 出力の印刷フォーマットを指定するオプション・パラメーター。以下のいずれか 1 つの値を指定できます。

### WRAP

必要に応じて次の行に折り返します。これはデフォルトです。

### BYCOL

データの行が列単位でグループ化されます。

### BYRSC

データの行がリソース単位でグループ化されます。

### ROUTE

コマンドを実行する IMSpIex メンバーの SYSID を指定するオプション・パラメーター。ROUTE を指定しない場合、IMSpIex のすべてのメンバーがコマンドを実行します。複数のメンバーを指定する場合は、リストを小括弧で囲み、名前をコンマで区切ります。例えば、次のとおりです。

```
// PARM=('IMSPLEX=PLEX1,WAIT=30,ROUTE=(IMSZ,IMSA)')
```

ROUTE=\* を指定すると、コマンドは IMSpIex 内のすべての登録済みコマンド処理クライアントに送付されます。ROUTE=% を指定すると、IMSpIex 内の登録済みコマンド処理クライアントの中で、コマンドが登録されており、さらにマスター機能を持つ 1 つのクライアントにのみコマンドが送付されます。

Operations Manager がそのコマンド処理クライアントを選択します。

### WAIT

個々のコマンドの待ち時間を指定するオプション・パラメーター。WAIT 値は分と秒 (MMM:SS) または秒のみ (SSSS) で指定します。OM は、IMSpIex の全メンバーから応答を受信すると即座に単一応答を返します。この間隔が満了した場合、OM は IMSpIex メンバーからのいくつかの応答を返し、それに加えて、一部が応答しなかったことを示します。バッチ SPOC ユーティリティは、各コマンドが完了するのを待ってから、次のコマンドを発行します。デフォルトの WAIT 値は 5 分 (5:00) です。WAIT 時間は、SYSIN ファイル内のすべてのコマンドに適用されます。ユーザーは、待ち時間にゼロ秒を指定できます。この場合、バッチ SPOC はコマンドを発行しますが、応答を待機しません。

## 戻りコード

以下の戻りコードが作成されます。

コード 意味

- 0 ユーティリティーが正常に完了しました。
- 4 警告メッセージが発行されました。出力ファイルを確認してください。
- 8 問題が検出されました。出力ファイルを確認してください。1 つ以上の IMS オペレーター・コマンドが失敗しました。必要なコマンドを使用して、ユーティリティーを再実行してください。
- 12 内部エラーバッチ SPOC ユーティリティーがコマンドまたはコマンド応答を処理できませんでした。返されるエラー・メッセージ CSLM0142E に追加情報があります。

---

## バッチ SPOC ユーティリティーの例

これらの例では、バッチ SPOC ユーティリティーを使用して定様式のコマンド応答を作成する方法を示します。

### JCL の例

次の例は、複数のコマンドを持つサンプル・バッチ・ジョブの単純な呼び出しを示していますが、ユーザーは、他の有効な JCL を使用して、このユーティリティーを呼び出すこともできます。

```
//SPOCJOB JOB ,
// MSGCLASS=H,NOTIFY=&SYSUID,USER=&SYSUID/*
//SPOC EXEC PGM=CSLUSPOC,
// PARM=('IMSPLEX=PLEX1,ROUTE=IMS3,WAIT=30,F=WRAP')
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  QRY IMSPLEX SHOW(JOB,TYPE, +
                    STATUS)

  QRY TRAN NAME(INV1*) SHOW(ALL) /* inventory appl */
/*EOF
```

### 出力例

SYSPRINT が SYSOUT ファイルの場合は、システム表示および検索機能 (SDSF) を使用して、バッチ・ジョブ出力を表示することができます。次の例は、サンプルのバッチ・ジョブ出力を示しています。

```
=====
Log for. . . : QRY IMSPLEX SHOW(JOB,TYPE,STATUS)

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS3
Start time. . . . : 2010.132 15:36:28.11
Stop time . . . . : 2010.132 15:36:29.17
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : SYS3

IMSpIex MbrName CC Member JobName Type Status
CSLPLEX1 OM10M 0 USRT002 USRT002 AOP ACTIVE
```



```

CSLPLEX1 OM10M 0 OM10M OM1 OM READY,ACTIVE
CSLPLEX1 OM10M 0 RM1RM RM1 RM READY,ACTIVE
CSLPLEX1 OM10M 0 SCI1SC SCI1 SCI READY,ACTIVE
CSLPLEX1 OM10M 0 IMS3 IMS3 IMS READY,ACTIVE
CSLPLEX1 OM10M 0 SYS1 SYS1 IMS READY,ACTIVE
=====

```

### 待ち時間を指定しない出力の例

待ち時間を指定しなかった場合 (WAIT=0)、コマンド応答は利用不可であり、印刷されません。SYSPRINT ファイルには、各コマンドの簡単な要約情報だけが入ります。次の例は、応答がないサンプルのバッチ・ジョブ出力を示しています。

```

=====
Log for. . : QRY IMSPLEX SHOW(JOB,TYPE,STATUS)

IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2010.075 15:36:28.11
=====

```



---

## 第 24 章 データベース・リカバリー管理ユーティリティ (DSPURX00)

データベース・リカバリー管理ユーティリティを使用して、RECON データ・セットを構築および保守するコマンドを出し、RECON データ・セットに情報を追加し、ユーティリティ用のジョブを生成します。

データベース・リカバリー管理ユーティリティに実行依頼されるコマンドは、同じ汎用フォーマットを持っています。それぞれのコマンドは verb と修飾子 (ピリオドで区切られている) から構成され、その後にパラメーターが続きます。

さらに、データベース・リカバリー管理ユーティリティは、CHANGE.RECON UPGRADE コマンドを使用して、サポートされている、以前の版の IMS から RECON データ・セットをアップグレードするのにも使用できます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 『入力と出力』
- 617 ページの『JCL 仕様』

### 制約事項

現在、DSPURX00 ユーティリティについて文書化されている制約事項はありません。

### 前提条件

現在、DSPURX00 ユーティリティについて文書化されている前提条件はありません。

### 要件

現在、DSPURX00 ユーティリティについて文書化されている要件はありません。

### 推奨事項

現在、DSPURX00 ユーティリティについて文書化されている推奨事項はありません。

### 入力と出力

次の図では、データベース・リカバリー管理ユーティリティの入出力の要件を示します。図の後には、その入出力の詳細を説明する注記が続きます。

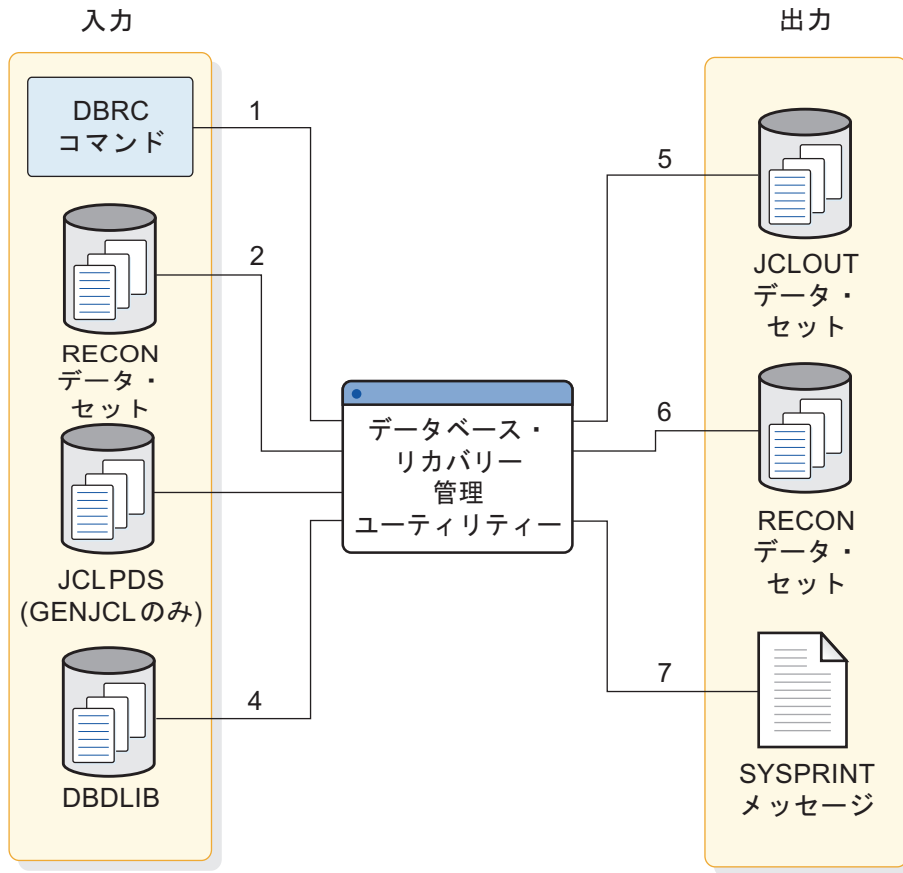


図 39. データベース・リカバリー管理ユーティリティの入出力

図に関する注:

1. DBRC コマンド (データベース・リカバリー管理ユーティリティへの入力)
2. RECON データ・セット (データベース・リカバリー管理ユーティリティへの入力)
3. PDS。これには、DBRC がジョブを生成するために使用する、ユーティリティの JCL および制御ステートメントが含まれます (データベース・リカバリー管理ユーティリティへの入力)
4. DBRC の制御下に置かれるデータベースのデータベース記述が入ったデータ・セット (データベース・リカバリー管理ユーティリティへの入力となる場合があります) ACB の IMS 管理が使用可能な場合、データベース記述子は、DBDLIB データ・セットではなく IMS カタログから取得されます。
5. GENJCL コマンドによって作成されたジョブ (データベース・リカバリー管理ユーティリティからの出力)
6. RECON データ・セット。これはユーティリティによって更新されている場合があります (データベース・リカバリー管理ユーティリティからの出力)
7. 次のうちの 1 つ以上 (データベース・リカバリー管理ユーティリティからの出力):
  - 入力コマンドのリスト
  - その実行に関連した通知メッセージ、または何らかの障害を説明する診断メッセージおよび戻りコード

- GENJCL コマンドの場合に作成された各ジョブのリスト

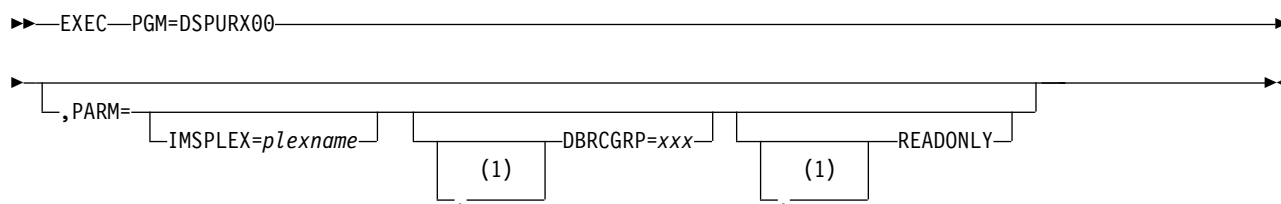
## JCL 仕様

データベース・リカバリー管理ユーティリティは標準 z/OS ジョブとして実行されます。

### EXEC ステートメント

実行されるプログラムを示します。

EXEC ステートメントには、3 つのオプション・パラメーター IMSPLEX、DBRCGRP、および READONLY を指定できます。これらのパラメーターを指定する場合は、次のフォーマットを使用する必要があります。



注:

- 1 複数のオプション・パラメーターを指定する場合は、コンマで区切る必要があります。

### IMSPLEX

DBRC が結合する IMSplex を指定します。

IMSPLEX パラメーター は、DBRC を使用するすべてのジョブ・ステップで指定できます。

### DBRCGRP

DBRC グループで使用される RECON データ・セットで定義されている DBRC グループ ID を指定します。

### READONLY

ユーザーに読み取り専用権限を与えることを指定します。READONLY モードのときにステートメントを更新しようとすると、エラーが発生します。

### DD ステートメント

#### STEPLIB

IMS.SDFSRESL を指します。ここには、IMS 中核と必要なアクション・モジュールが入っています。

#### SYSPRINT

DBRC 診断メッセージとリスト出力の宛先を定義します。宛先をテープまたは DASD データ・セット、またはプリンターにすることも、あるいは出力ストリーム (SYSOUT) を通じて経路指定することもできます。

#### RECON

RECON データ・セットを動的に割り当てるために、RECON1、RECON2、および RECON3 の DD ステートメントは省略します。

**IMS** ACB、DBD、および PSB ライブラリーを使用する IMS システムで、IMS DBDLIB データ・セットを定義します。ACB の IMS 管理が使用可能な場合、IMS DD ステートメントは無視されます。

ACB、DBD、および PSB ライブラリーを使用する IMS システムでは、以下のコマンドに IMS DD ステートメントが必要です。

- INIT.PART
- INIT.DBDS
- NOTIFY.REORG
- HALDB を初期設定する場合は、INIT.DB
- DBDS DD 名またはエリア名を変更する場合は、CHANGE.DBDS
- HALDB 区画の KEYSTRNG 値を変更する場合は、CHANGE.PART

ACB の IMS 管理が使用可能な場合、上記のコマンドは、RECON データ・セットに登録されているか、またはコマンド自体に指定されている IMS カタログからデータベース定義を取得します。


**JCLPDS**、または **JCLPDS** パラメーターでユーザーが指定する **DD** 名 基幹 JCL メンバーを含む PDS を定義します。これは GENJCL コマンドの場合にのみ必要です。

**JCLOUT**、または **JCLOUT** パラメーターでユーザーが指定する **DD** 名 生成される JCL を受け取るデータ・セットを定義します。これは GENJCL コマンドの場合にのみ必要です。

#### **SYSIN**

入力コマンドのソースを定義します。SYSIN は、テープ・データ・セットまたは DASD データ・セット、あるいはカード読取装置でもかまいませんが、入力ストリーム (DD \* または DD DATA) を通じて経路指定することもできます。

関連概念:

 データ・セット命名規則 (システム管理)

---

## DSPURX00 ユーティリティーの例

RECON データ・セットを初期設定し、1 つのデータベースを 2 つの DBDS に登録することができます。

### データベース・リカバリー管理ユーティリティーの入出力

次の図は、RECON データ・セットを初期設定し、1 つのデータベースを 2 つの DBDS に登録するジョブの例です。

```
//INITRCON JOB
//INIT04 EXEC PGM=DSPURX00
//STEPLIB DD DSN=IMS.SDFSRESL
//SYSPRINT DD SYSOUT=A
//*
//IMS DD DSN=IMS.DBDLIB
//JCLPDS DD DSN=IMS.JCLPDS
//JCLOUT DD DSN=IMS.JCLOUT
//SYSIN DD *
INIT.RECON SSID(IMS3)
```

```

INIT.DB   DBD(DBDESDS1)  SHAREVL(2)
INIT.DBDS DBD(DBDESDS1)  DDN(DDNESDSA)  GENMAX(3)  -
          REUSE DSN(IMS.DBDESDS1.DDNESDSA.DSN)  -
          ICJCL(MYIC) RECOVJCL(MYRECOV)  -
INIT.IC   DBD(DBDESDS1)  DDN(DDNESDSA)  -
          ICDSN(IMS.*.ICDSN1)
INIT.IC   DBD(DBDESDS1)  DDN(DDNESDSA)  -
          ICDSN(IMS.*.ICDSN2)  ICDSN2(IMS.*.ICDSN2)
INIT.IC   DBD(DBDESDS1)  DDN(DDNESDSA)  -
          ICDSN(IMS.*.ICDSN3)
/*
INIT.DBDS DBD(DBDESDS1)  DDN(DDNESDSB)  GENMAX(4)  -
          NOREUSE DSN(IMS.DBDESDS1.DDNESDSB.DSN)
/*
INIT.CAGRP GRPNAME(CAGRP1)  GRPMAX(2)  REUSE  -
          GRPMEM((DBDESDS1,DDNESDSA),(DBDESDS1,DDNESDSB))
INIT.CA   GRPNAME(CAGRP1)  CADSN(IMS.*.CADSN1)  -
          VOLLIST(CAVOL1,CAVOL2,CAVOL3)  FILESEQ(4)
INIT.CA   GRPNAME(CAGRP1)  CADSN(IMS.*.CADSN2)  -
          VOLLIST(CAVOL4)
/*

```

---

## エントリー・ポイント **DSPURXRT** を使用したユーティリティの呼び出し

エントリー・ポイント DSPURXRT を使用して、SYSPRINT、SYSIN、IMS、RECON1、RECON2、および RECON3 の各データ・セットに代替 DD 名を指定することができます。

ユーザーの呼び出し側プログラムがこれらの DD 名をすでに使用している場合は、異なる DD 名を使用します。また、データベース・リカバリー管理ユーティリティに異なるデータ・セットを使用させます。

RECON1 DD ステートメントが無い場合は、RECON データ・セットが動的に割り振られます。また、JCLOUT および JCLPDS DD ステートメントも置換できません。

特に断りがない限り、DBRC を呼び出す各プログラムは、EXEC ステートメントに指定されたパラメーターを DBRC に渡す必要がありません。DBRC が z/OS から渡されたパラメーター・リストを見つけ出すので、呼び出し側プログラムでこのパラメーター・リストを変更する必要はありません。

エントリー・ポイント DSPURXRT は 31 ビット・モードで呼び出す必要があります。DBRC は、31 ビット・モードで実行し、レジスター 15 に戻りコードを入れて、31 ビット・モードで呼び出し側に戻します。CALL ではパラメーターを 2 つまで渡すことができます。

- 最初のパラメーターは、オプションのアドレスを指定します。このオプションは、EXEC ステートメントの PARM パラメーターで指定できるものです。DSPURXRT エントリー・ポイントでサポートされるオプションは READONLY だけです。このオプションでは、アプリケーションが DBRC 情報に対する読み取り専用アクセスを必要とするかどうかを指定します。

- 2 番目のパラメーターは、DSPURXRT エントリー・ポイント処理の中で使用される、標準データ・セットの代替 DD 名のリストのアドレスを指定します。標準 DD 名を使用する場合は、2 番目のパラメーターが 2 進ゼロのハーフワードを指すか、または省略されます。

エントリー・ポイント DSPURXRT は、ユーザー・プログラムで ATTACH、LINK、または LOAD および CALL マクロ命令を使用して呼び出します。呼び出しマクロを実行する前に、プログラムは該当のレジスターとオペランド・リストの内容を初期化します。レジスターの内容は、以下のような標準リンケージ規則に従っています。

- レジスター 1 には、引数リストのアドレスが含まれています。
- レジスター 13 には、保管域のアドレスが含まれています。
- レジスター 14 には、戻り点のアドレスが含まれています。
- レジスター 15 には、エントリー・ポイント DSPURXRT のアドレスが含まれています。

レジスター 1 で示される引数リストは、パラメーターに対する最大 2 個のポインターで構成されます。リストの最後のワードでは、高位ビットがオンになり、最後のワードであることを示します。最初のワードは、エントリー・ポイント DSPURXRT に渡されるオプションのアドレスです。オプション・リストは、オプション・リストの全長が入っているハーフワード (長さにはこのハーフワードも含まれる) で構成されます。オプションが渡されない場合、この長さはゼロです。IMSPLEX= および DBRCGRP= オプションは、EXEC JCL ステートメントで指定されるパラメーターとしてのみサポートされます。

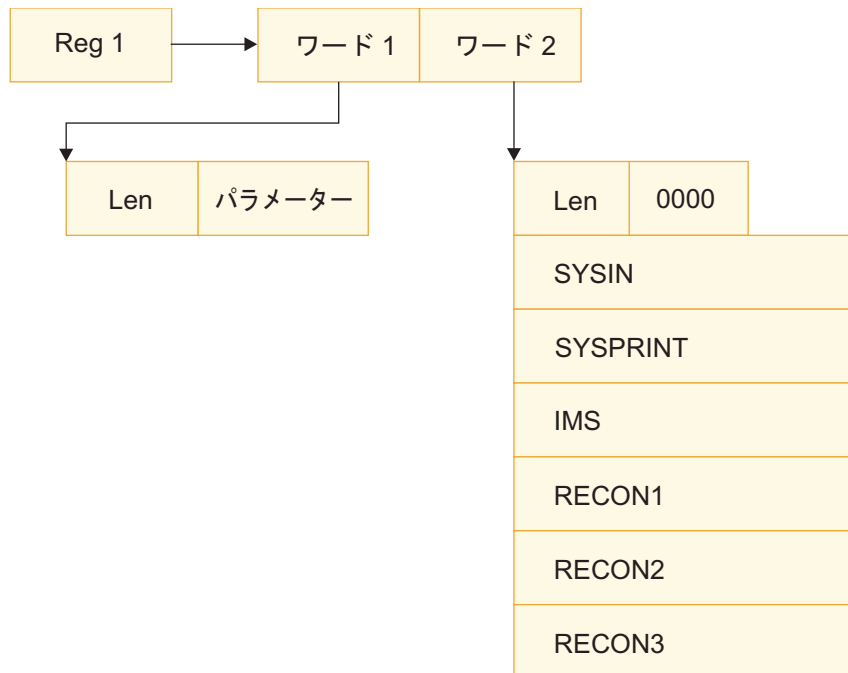


図 40. データベース・リカバリー管理ユーティリティを呼び出すために必要なレジスターとオペランド・リストの内容



図の 2 番目のワードは、DD 名のリストのアドレスです。DD 名は左寄せし、必要な場合は右側をブランクで埋めます。

DD 名は次のとおりです。

- DD 名リストの全長が入っているハーフワード (長さにはこのハーフワードも含まれる)。
- 予約ハーフワード。
- SYSIN データ・セットの代わりに使用する DD 名が入っている 8 バイト・フィールド。または、SYSIN を置き換えない場合は、8 バイトのブランク (X'40')。
- SYSPRINT データ・セットの代わりに使用する DD 名が入っている 8 バイト・フィールド。または、SYSPRINT を置き換えない場合は、8 バイトのブランク (X'40')。
- IMS データ・セットの代わりに使用する DD 名が入っている 8 バイト・フィールド。または、IMS を置き換えない場合は、8 バイトのブランク (X'40')。
- RECON1 データ・セットの代わりに使用する DD 名が入っている 8 バイト・フィールド。または、RECON1 を置き換えない場合は、8 バイトのブランク (X'40')。
- RECON2 データ・セットの代わりに使用する DD 名が入っている 8 バイト・フィールド。または、RECON2 を置き換えない場合は、8 バイトのブランク (X'40')。
- RECON3 データ・セットの代わりに使用する DD 名が入っている 8 バイト・フィールド。または、RECON3 を置き換えない場合は、8 バイトのブランク (X'40')。

関連資料:

 z/OS: MVS ATTACH マクロ

 z/OS: MVS CALL マクロ

 z/OS: MVS LINK マクロ

 z/OS: MVS LOAD マクロ



---

## 第 25 章 動的 SVC ユーティリティー (DFSUSVC0)

動的監視プログラム呼び出し (SVC) ユーティリティーを使用すると、z/OS SVC テーブルを変更して新しいコピーの SVC モジュールを指し示すことにより、z/OS オペレーティング・システムの IPL を行わなくても、更新されたバージョンの IMS タイプ 2 SVC または DBRC タイプ 4 SVC をインストールできます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 624 ページの『要件』
- 624 ページの『推奨事項』
- 624 ページの『入力と出力』
- 624 ページの『JCL 仕様』
- 625 ページの『戻りコード』

### 制約事項

動的 SVC ユーティリティーには、下記の制約事項が適用されます。

- IMS タイプ 2 SVC モジュールを更新しようとするときに、このタイプ 2 SVC を使用する IMS イメージ (制御領域、バッチ、またはユーティリティー) を、アクティブにすることはできません。これと同じ制約事項は、DBRC タイプ 4 SVC モジュールには適用されません。

### 前提条件

動的 SVC ユーティリティーには、以下の前提条件があります。

- IMS.SDFSRESL は、置換される正しい SVC 番号を反映している必要があります。誤ったライブラリーを指し示し、それによって別の SVC 番号が (あるいは非 IMS SVC 番号さえもが) このライブラリーへ関連付けられると、エラーが発生する可能性があります。

SVC 番号を指定して、番号が DFSUSVC0 ユーティリティーの IMS.SDFSRESL に含まれるようにするには、DFSIDEF0 モジュールを試用する必要があります。DFSUSVC0 ユーティリティーのタイプ 2 SVC 番号を指定するには、DFSIDEF0 モジュールの DFSIDEF マクロに DFSIDEF TYPE=PARM,SVC2=xxx を指定します。ここで、xxx はタイプ 2 SVC 番号です。DFSUSVC0 ユーティリティーのタイプ 4 SVC 番号を指定するには、DFSIDEF0 モジュールの DFSIDEF マクロに DFSIDEF TYPE=PARM,SVC4=yyy を指定します。ここで、yyy はタイプ 4 SVC 番号です。

警告: このユーティリティーを使用する前に、システム管理者に確認してください。

## 要件

動的 SVC ユーティリティには、以下の要件があります。

- JCL は、IMS.SDFSRESL を参照する DFSRESLB DD ステートメントを含む必要があります。
- 更新された SVC モジュール (IMS タイプ 2 SVC か DBRC タイプ 4 SVC、またはその両方) は、DFSRESLB DD ステートメントで指定された IMS.SDFSRESL に入っている必要があります。
- SVC 番号と新しい SVC モジュールを含む IMS.SDFSRESL は、APF 許可ライブラリーでなければなりません (標準 IMS インストール)。
- このユーティリティ・プログラムは、APF 許可ライブラリー (通常は、IMS.SDFSRESL ですが、これは必要条件ではありません) の中に存在する必要があります。
- すべての IMS SVC が、既に z/OS に対して定義されている必要があります。
- 動的 SVC ユーティリティを使用して更新バージョンの SVC をインストールする場合は、従来どおり IMS タイプ 2 SVC モジュールを z/OS 中核に追加し、タイプ 4 SVC モジュールを SYS1.LPALIB ライブラリーまたは MLPA ライブラリーに追加する必要があります。これを行わないと、z/OS の IPL により、IMS システムは旧バージョンの SVC モジュールの使用に戻ります。この場合は、動的 SVC ユーティリティを使用して、新規バージョンの IMS SVC モジュールを再インストールする必要があります。

## 推奨事項

現在、DFSUSVC0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

このユーティリティへの入力は、更新された IMS タイプ 2 SVC モジュールであるか、更新された DBRC タイプ 4 SVC モジュールであるか、その両方です。更新された SVC モジュールは、DFSRESLB DD ステートメントによって指し示されるライブラリーの中に存在する必要があります。

ユーティリティはどの SVC を更新すべきかを判別し、動的に z/OS SVC テーブルを変更して、新しい SVC モジュールを指し示します。

## JCL 仕様

動的 SVC ユーティリティは、標準 z/OS ジョブとして実行されます。以下のものを指定する必要があります。

- JOB ステートメント
- EXEC ステートメント
- 入力を定義する DD ステートメント

### EXEC ステートメント

このユーティリティは、z/OS ジョブとして実行されます。

EXEC ステートメントは、以下のいずれかの形式であることが必要です。

- //STEP001 EXEC PGM=DFSUSVC0 または
- //STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(2)' または
- //STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(4)' または
- //STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(2,4)'

EXEC ステートメントにより、更新する対象が IMS タイプ 2 SVC モジュールなのか、DBRC タイプ 4 SVC モジュールなのか、あるいはその両方なのかを指定できます。SVCTYPE=(2) を指定すると、IMS タイプ 2 SVC モジュールが更新されます。SVCTYPE=(4) を指定すると、DBRC タイプ 4 SVC が更新されます。SVCTYPE=(2,4) を指定すると、IMS タイプ 2 SVC と DBRC タイプ 4 SVC の両方のモジュールが更新されます。SVCTYPE= パラメーターに値を指定しなかった場合は、デフォルトとして IMS タイプ 2 SVC モジュールが更新されます。

**DD** ステートメント

#### **STEPLIB DD**

実際の DFSUSVC0 ユーティリティーが入っている許可ライブラリーを指し示します。許可ライブラリーは、ご使用の IMS.SDFSRESL に入っている必要があります。

```
//STEPLIB DD DSN=SOME.APF.AUTHORIZED.DATASET,DISP=SHR
```

#### **DFSRESLB DD**

更新された SVC モジュールと IMS タイプ 2 および DBRC タイプ 4 SVC の番号が入っている許可ライブラリーを指し示します。

```
//DFSRESLB DD DSN=SOME.IMS.SDFSRESL,DISP=SHR
```


### 戻りコード

以下の戻りコードが作成されます。

コード 意味

- 0 動的インストールが正常に終了しました。指定されたすべての SVC ルーチンは正しく更新されました。
- 8 指定された SVC ルーチンの少なくとも 1 つのインストールが失敗しました。

関連タスク:

 タイプ 2 SVC モジュールのインストール (システム管理)

---

## DFSUSVC0 ユーティリティーの例

これらの例では、DFSUSVC0 ユーティリティーを使用して、IMS タイプ 2 SVC および DBRC タイプ 4 SVC を置き換える方法を示します。

以下に、IMS タイプ 2 SVC および DBRC タイプ 4 SVC を置き換えるために必要な JCL を示します。

### IMS タイプ 2 SVC の置き換えの例

```
//SVCINIT JOB MSGLEVEL=1,TIME=1440
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(2) '
//STEPLIB DD DSN=SOME.APF.AUTHORIZED.DATASET,
//          DISP=SHR
//DFSRESLB DD DSN=SOME.IMS.SDFSRESL,
//          DISP=SHR
```

### DBRC タイプ 4 SVC の置き換えの例

```
//SVCINIT JOB MSGLEVEL=1,TIME=1440
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(4) '
//STEPLIB DD DSN=SOME.APF.AUTHORIZED.DATASET,
//          DISP=SHR
//DFSRESLB DD DSN=SOME.IMS.SDFSRESL,
//          DISP=SHR
```

### 両方の SVC モジュールの置き換えの例

```
//SVCINIT JOB MSGLEVEL=1,TIME=1440
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(2,4) '
//STEPLIB DD DSN=SOME.APF.AUTHORIZED.DATASET,
//          DISP=SHR
//DFSRESLB DD DSN=SOME.IMS.SDFSRESL,
//          DISP=SHR
```

---

## 第 26 章 グローバル・オンライン変更ユーティリティー (DFSUOLC0)

グローバル・オンライン変更ユーティリティーを使用して、OLCSTAT データ・セットの初期設定、再作成、またはアンロックを行います。

グローバル・オンライン変更ユーティリティーを使用して OLCSTAT データ・セットを初期設定するには、MODBLKS データ・セットを定義していない IMS の場合でも、最初に MDBS=A または MDBS=B を定義する必要があります。IMSplex でグローバル・オンライン変更を使用可能にするには、IMSplex 内の最初の IMS を初めてコールド・スタートする前に、グローバル・オンライン変更ユーティリティーを使用して OLCSTAT を初期設定する必要があります。グローバル・オンライン変更ユーティリティーを使用すれば、エラーが使用不可にならしめた OLCSTAT データ・セットを再作成することができます。

グローバル・オンライン変更コマンド INITIATE OLC PHASE(PREPARE) と、それに続けて INITIATE OLC PHASE(COMMIT) を使用すれば、非アクティブ・ライブラリーがアクティブ・ライブラリーになります。

OLCSTAT データ・セットはグローバル・オンライン変更状況を含んでいます。これには、変更 ID、アクティブ・オンライン変更ライブラリー、ロック・フィールド、最後のオンライン変更、およびオンライン変更ライブラリーを使う現行 IMS のリストが含まれます。IFP 領域が稼働しているときは、既存のアクティブ経路コードのため、OLC コミットは停止します。したがって、コミットの前に、IFP 領域はすべて終了しておく必要があります。

オンライン変更コピー・ユーティリティーは OLCSTAT DD ステートメントをサポートし、グローバル・オンライン変更状況データ・セット名を識別します。OLCSTAT データ・セットは、ローカル・オンライン変更で使用する MODSTAT データ・セットに似ています。

**重要:** 再作成機能とアンロック機能の使用には細心の注意を払ってください。一連のエラーによって OLCSTAT データ・セットがロック状態のままにされ、オンライン変更がまったく行われなくなったときにのみ、アンロック機能を使用してください。有効な OLCSTAT データ・セットの内容を不注意で破棄すると、グローバル・オンライン変更および追加 IMS システムの初期設定は、OLCSTAT データ・セットが再初期設定されるまで行えません。

OLCSTAT データ・セットが失われた場合に対処する、OLCSTAT データ・セット・リカバリー手順を確立してください。グローバル・オンライン変更が正常終了する都度、以下のデータを記録してください。

- 変更 ID
- アクティブ・オンライン変更ライブラリーの接尾部
- オンライン変更ライブラリーを使う現行 IMS システムのリスト

OLCSTAT データ・セットが破棄された場合は、記録されたデータを使用してグローバル・オンライン変更ユーティリティの初期設定機能を実行し、OLCSTAT データ・セットを再初期設定します。

OLCSTAT データ・セット内のオンライン変更ライブラリーの現行値を示す DFS3499 メッセージは、DFS994 チェックポイント・メッセージの後に続きます。初期設定時の DFS3410 メッセージも、OLCSTAT データ・セットからの現行オンライン変更ライブラリーを示します。

DFSUOLC0 を使用して OLCSTAT データ・セット・ヘッダーの IMS バージョンを更新することもできます。前のレベルよりも高い IMS の新バージョンへアップグレードする場合、OLCSTAT ヘッダーの IMS バージョンが IMS コールド・スタート時に自動的に更新されます。ただし、下位の IMS レベルへフォールバックする場合は、この自動更新は完了しません。この場合は、OLCSTAT ヘッダーの IMS バージョンを下位 IMS レベルに更新するためには、グローバル・オンライン変更ユーティリティを実行させる必要があります。

UNL 機能を使用して OLCSTAT データ・セットをアンロックしようとして、グローバル・オンライン変更中にすべての IMS が失敗した場合は、IMS を始動する必要があります。OLCSTAT データ・セットはすでにロックはされていないので、IMS は始動します。

OLCSTAT データ・セットのロックを保持したままオンライン変更を行っている最中にグローバル・オンライン変更コマンドのマスターが失敗し、その後で、UNL 機能を使用して OLCSTAT データ・セットをアンロックする場合、次に発行するグローバル・オンライン変更コマンドは、オンライン変更の状態によって異なります。

- グローバル・オンライン変更が、OLCSTAT データ・セットが更新されたコミット 1 フェーズを渡した場合、オンライン変更がコミットされ、発行可能なコマンドは、クリーンアップによってコミットを完了する別の INIT OLC PHASE(COMMIT) コマンドのみになります。
- OLCSTAT データ・セットが更新されたコミット・フェーズ 1 の前にグローバル・オンライン変更が失敗した場合、グローバル・オンライン変更はコミットされません。INIT OLC PHASE(COMMIT) コマンドを指定してコミットを再試行するか、TERMINATE OLC コマンドを指定してグローバル・オンライン変更を中止することができます。

グローバル・オンライン変更の状態を確認するには、QUERY MEMBER TYPE(IMS) SHOW(STATUS) コマンドを発行して、グローバル・オンライン変更に関係するすべての IMS の状態を表示します。詳細については、グローバル・オンライン変更コマンドの取扱エラー (システム管理)を参照してください。

サブセクション:

- 629 ページの『制約事項』
- 629 ページの『前提条件』
- 629 ページの『要件』
- 629 ページの『推奨事項』
- 629 ページの『JCL 仕様』



## 制約事項

現在、DFSUOLC0 ユーティリティについて文書化されている制約事項はありません。

## 前提条件

現在、DFSUOLC0 ユーティリティについて文書化されている前提条件はありません。

## 要件

現在、DFSUOLC0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSUOLC0 ユーティリティについて文書化されている推奨事項はありません。

## JCL 仕様

次の JCL は DFSUOLC プロシージャと一緒に実行され、デフォルトである VERS=2 を使用してユーティリティを呼び出します。

```
//DFSUOLC0 JOB
//STEP1 EXEC DFSUOLC, FUNC=, ACBS=, MDDBS=, FMDS=, MDID=, PLEX=
//SYSIN DD *
//
```

次の JCL は DFSUOLC プロシージャと一緒に実行され、VERS=1 を使用してユーティリティを呼び出します。

```
//DFSUOLC0 JOB
//STEP1 EXEC DFSUOLC, FUNC=, ACBS=, MDDBS=, FMDS=, MDID=, PLEX=, VERS=1
//SYSIN DD *
//
```

### プロシージャ・ステートメント

以下の図に示されている JCL では、DFSUOLC プロシージャを呼び出す場合に使用されるステートメントを示しています。このプロシージャは、システム定義のステージ 2 で生成され、IMS.PROCLIB に置かれます。

```
//PROC FUNC=, ACBS=, MDDBS=, FMDS=, MDID=, PLEX=, VERS=, SOUT=A
//STEP1 EXEC PGM=DFSUOLC0,
//          PARM=(&FUNC, &ACBS, &MDDBS, &FMDS, &MDID, &PLEX, &VERS)
//OLCSTAT DD DSN=IMSPLEX.OLCSTAT, DISP=OLD
//SYSPRINT DD SYSOUT=&SOUT
//SYSIN DD DUMMY
```

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
PGM=DFSUOLC0, PARM=(&FUNC, &ACBS, &MDDBS, &FMDS, &MDID, &PLEX, &VERS)
```

グローバル・オンライン変更ユーティリティ (DFSUOLC0) には、以下のパラメーターを指定できます。

**ACBS** アクティブ ACB ライブラリーの IMS JCL IMSACB DD ステートメント

の接尾部を指定します。この接尾部は A または B です。A は、IMSACBA がアクティブ・ライブラリーの DD ステートメントであることを示し、B は、IMSACBB がアクティブ・ライブラリーの DD ステートメントであることを示します。

#### **FMTS**

アクティブ MFS 形式 (FORMAT) ライブラリーの IMS JCL FORMAT DD ステートメントの接尾部を指定します。この接尾部は A または B です。A は、FORMATA がアクティブ・ライブラリーの DD ステートメントであることを示し、B は、FORMATB がアクティブ・ライブラリーの DD ステートメントであることを示します。FORMAT は、オンライン・システムが形式ライブラリーとして使用するオンライン MFS 定義を含んでいます。MFS がサポートする端末および MFS 言語ユーティリティ・プログラムでは、それらを使用する必要があります。このパラメーターは、IMSplex 内の IMS が MFS 形式ライブラリーを使用しない場合でも必要です。

#### **FUNC**

グローバル・オンライン変更ユーティリティで実行する機能を指定します。

#### **ADD**

オンライン変更ライブラリーを使う現行 IMS システムのリストに、1 つ以上の IMS メンバーを追加します。OLCSTAT データ・セットがエラーの影響を受けたために使用不可になり、OLCSTAT データ・セットの内容を再作成するときは、IMS を追加してください。追加する IMS システムは、SYSIN DD カードで指定する必要があります。

オンライン変更ライブラリーを使う現行 IMS システムを追加します。例えば、現在、立ち上がっている IMS システムを追加します。

**重要:** オンライン変更ライブラリーに対してカレントでない IMS を追加して、その IMS をウォーム・スタートした場合は、そのウォーム・スタートは失敗することがあります。

#### **DEL**

オンライン変更ライブラリーを使う現行 IMS システムのリストから、1 つ以上の IMS システムを削除します。

IMS を再度立ち上げる予定がない場合は、その IMS を削除してください。そうすれば、INITIATE OLC コマンドを FRCABND または FRCNRML のキーワードと一緒に指定する必要がありません。削除する IMS システムは、SYSIN DD カードで指定する必要があります。

#### **INI**

OLCSTAT データ・セットを初期設定する機能です。ACBS、MDBS、FMTS、および MDID も指定する必要があります。1 つ以上の IMS システムのリストを SYSIN DD ステートメントで指定できます (オプション)。SYSIN DD ステートメントで IMS システムを指定しなければ、OLCSTAT データ・セットから IMS システムのリストが削除されます。

IMSpdex 内の最初の IMS を初めてコールド・スタートする前に、OLCSTAT データ・セットを初期設定するために INI 機能が必要です。

OLCSTAT レコードの内容が失われ、再構成する必要がある場合は、グローバル・オンライン変更ユーティリティの INI 機能を実行して、オンライン変更 ID とオンライン変更ライブラリー DD 名の正しい値を指定してその内容を構成する必要があります。オンライン変更ライブラリーを使う現行 IMS システムを、SYSIN DD ステートメントで追加することもできます。障害発生時に OLCSTAT データ・セットの内容を再構成できるように、現行のオンライン変更ライブラリーと変更 ID を記録してください。

#### UNL

オンライン変更中にすべての IMS システムが失敗したとき、OLCSTAT データ・セットのロックをリセットする機能です。

グローバル・オンライン変更ユーティリティの UNL 機能は、IMSpdex 内のすべての IMS システムがオンライン変更中に失敗した場合に、OLCSTAT データ・セットのロックをリセットするために必要です。オンライン変更は、OLCSTAT データ・セット内のロック・フィールドを設定し、オンライン変更中に他の IMS システムが初期設定できないようにします。グローバル・オンライン変更が進行中 (準備フェーズからコミット・フェーズまで) の間は、OLCSTAT データ・セットのロックが設定されているので、IMS の初期設定は失敗します。オンライン変更中にすべての IMS システムが失敗した後で、ある IMS が初期設定を試行しても、OLCSTAT データ・セットのロックは設定されたままなので、IMS の初期設定はリジェクトされます。この場合、グローバル・オンライン変更ユーティリティで UNL 機能を実行し、OLCSTAT データ・セットのロックをリセットする必要があります。OLCSTAT データ・セットのロックをリセットしないと、IMS は初期設定できません。UNL 機能を使うことはめったにありません。これが必要になるのは、オンライン変更中にすべての IMS システムが失敗した場合のみです。

#### MDBS

アクティブ MODBLKS データ・セットの IMS JCL MODBLKS DD ステートメントの接尾部を指定します。この接尾部は A または B です。A は、MODBLKSA がアクティブ・ライブラリーの DD ステートメントであることを示し、B は、MODBLKSA がアクティブ・ライブラリーの DD ステートメントであることを示します。

DRD が使用可能の場合、MODBLKS DD カードを定義せずに IMS を開始できます。しかし、グローバル・オンライン変更ユーティリティを使用して OLCSTAT データ・セットを初期設定するには、MODBLKS データ・セットを定義していない IMS の場合でも、最初に MDBS=A または MDBS=B を定義する必要があります。MODBLKS データ・セットが IMS に対して定義されていない場合、OLCSTAT データ・セット内の MODBLKS 値は無視されます。

#### MDID

INI 機能で使う変更 ID (オンライン変更状況 ID) を指定します。これは、

実施されたグローバル・オンライン変更の数がゼロであることを示すために、ゼロに初期設定しておきます。変更 ID は、1 つ以上のオンライン変更について IMS がダウンしたかどうかを判別するためと、IMS が実施できる再始動の種類を判別するために使用されます。この変更 ID は、以下のことを行うために、IMS の内部処理で使用されます。

- IMS をコールド・スタートする必要があるかどうかを判別するため。

最後のグローバル・オンライン変更に IMS が関与していれば、その変更 ID は OLCSTAT データ・セット内の変更 ID と一致します。この IMS はウォーム・スタートができます。最後のグローバル・オンライン変更に IMS が関与していなければ、その変更 ID は OLCSTAT データ・セット内の変更 ID と一致しません。その IMS の再始動タイプが、実施された最後のオンライン変更と矛盾しなければ、ウォーム・スタートができます。複数のグローバル・オンライン変更について IMS がダウンした場合は、コールド・スタートをしなければなりません。

- 緊急時再始動処理中にセキュリティ状況をリカバリーするため。

**PLEX** UNL 機能で使用する z/OS システム間カップリング・ファシリティー CSL IMSplex グループ名を表す 1 から 5 文字の ID を指定します。PLEX は、UNL 機能に必要です。同じ IMSplex 共用グループ内にあり、データベースまたはメッセージ・キューを共用するすべての OM、RM、SCI、IMS および ODBM IMSplex メンバーは、同じ ID を指定する必要があります。CSLSIxxx、CSLOIxxx、CSLRIxxx、および DFSCGxxx PROCLIB メンバーの IMSPLEX= パラメーターについても、同じ ID を使用する必要があります。

## SOUT

SYSOUT DD ステートメントに割り当てるクラスを指定します。

STEPLIB DD ステートメントは、IMS.SDFSRESL を識別します。IMS.SDFSRESL には、IMS 必須モジュールが入っています。この IMS.SDFSRESL は、IMSplex で使用可能な最高のレベルであることが必要です。

SYSUDUMP DD ステートメントは、このプログラムのダンプ・データ・セットを定義します。

SYSPRINT DD ステートメントは、メッセージ出力データ・セットを定義します。

OLCSTAT DD ステートメントは OLCSTAT (グローバル・オンライン変更状況) データ・セット名を識別します。OLCSTAT DD ステートメントは必須です。

SYSIN DD ステートメントは、定義、追加、または削除する IMS システムのリストを含んでいます。1 行に 1 つの IMS ID を指定します。

ADD 機能と共に指定された SYSIN DD ステートメントは、1 つ以上の IMS システムを、OLCSTAT データ・セット内の IMS システムの既存リストに追加します。

DEL 機能と共に指定された SYSIN DD ステートメントは、1 つ以上の IMS システムを、OLCSTAT データ・セット内の IMS システムの既存リストから削除します。

INI 機能と共に指定された SYSIN DD ステートメントは、IMS システムの新規リストを定義します。既に存在している IMS レコードは、消去されます。

**VERS** OLCSTAT の初期設定に使用する OLCSTAT のバージョンを指定します。有効な値は 1 と 2 です。デフォルト値は 2 です。

- 1 ヘッダーについて、OLCSTAT をバージョン 1 フォーマットに初期設定するよう指示します。

OLCSTAT 内に IMS 15 システムと一緒に 1 つ以上の IMS V9 または IMS V10 システムが存在する場合は、VERS=1 を使用してください。

- 2 ヘッダーについて、OLCSTAT をバージョン 2 フォーマットに初期設定するよう指示します。

OLCSTAT 内のすべての IMS システムが IMS 15 の場合は、VERS=2 を使用してください。TYPE(ACBMBR) メンバー・オンライン変更を利用するには、VERS=2 が必須です。

関連資料:

665 ページの『第 29 章 オンライン変更コピー・ユーティリティー (DFSUOCU0)』

---

## DFSUOLC0 ユーティリティーの例

これらの例では、DFSUOLC プロシージャーを使用して DFSUOLC0 ユーティリティーを呼び出し、OLCSTAT データ・セットを初期設定する方法を示します。

### グローバル・オンライン変更ユーティリティーの例 1

以下に示す例は、最初の IMS を初めてコールド・スタートする前に OLCSTAT データ・セットを初期設定するグローバル・オンライン変更ユーティリティーの JCL です。

```
//DFSUOLC0 JOB
//STEP1 EXEC DFSUOLC, FUNC=INI, ACBS=A, MDBS=A, FMST=A, MDID=0
//SYSIN DD *
/*
//
```

### グローバル・オンライン変更ユーティリティーの例 2

以下に示す例は、OLCSTAT データ・セット・ヘッダーを初期設定するグローバル・オンライン変更ユーティリティーの JCL です。OLCSTAT データ・セット・ヘッダーを初期設定するときに、IMS システムのリストを組み込む必要性はめったにありません。1 例として、OLCSTAT データ・セットが使用不可になった場合には、OLCSTAT ヘッダーを初期設定する必要があります。オンライン変更ライブラリーを使う現行 IMS システムが分かっているならば、それらの IMS システムをリストに組み込むことができます。IMSID を指定しなければ、OLCSTAT データ・セット・レコードに IMSID はリストされません。

```
//DFSUOLC0 JOB
//STEP1 EXEC DFSUOLC, FUNC=INI, ACBS=A, MDBS=A, FMST=A, MDID=0
//SYSIN DD
```

```
IMSA  
IMSB  
/*  
//
```

### グローバル・オンライン変更ユーティリティーの例 3

次の例は、IMS を初めてコールド・スタートする前に、OLCSTAT データ・セットをバージョン 1 に初期設定するグローバル・オンライン変更ユーティリティーの JCL を示しています。

```
//DFSUOLC0 JOB  
//STEP1 EXEC DFSUOLC, FUNC=INI, ACBS=A, MDBS=A, FMDS=A, MDID=0, VERS=1  
//SYSIN DD *  
//
```

### グローバル・オンライン変更ユーティリティーの例 4

次の例は、OLCSTAT データ・セット・ヘッダーをバージョン 1 に初期設定し、オンライン変更ライブラリーの現行の IMS 15 システムのリストを追加するグローバル・オンライン変更ユーティリティーの JCL を示しています。

```
//DFSUOLC0 JOB  
//STEP1 EXEC DFSUOLC, FUNC=INI, ACBS=A, MDBS=A, FMDS=A, MDID=0, VERS=1  
//SYSIN DD  
IMSA  
IMSB  
//
```

---

## 第 27 章 MFS サービス・ユーティリティー (DFSUTSA0)

MFS サービス (MFSRVC) ユーティリティー (DFSUTSA0) は、MFS 中間テキスト・ブロックと制御ブロックが MFS 言語ユーティリティー (DFSUPAA0)によって処理および保管された後で、これらのブロックの制御と保守を行うのに使用します。

中間テキスト・ブロック (ITB) は、IMS.REFERAL ライブラリーに保管されているメッセージ定義、形式定義、区画セット定義、またはテーブル・ソース言語定義です。制御ブロックは、IMS.FORMAT、IMS.FORMATA、IMS.FORMATB、または IMS.TFORMAT のいずれかのライブラリーに保管されているメッセージ定義または形式定義です。

サービス・ユーティリティーは、以下の機能を行います。

- INDEX は、IMS.FORMAT 制御ブロックへの高速アクセスを可能にする特殊なディレクトリーを作成します。
- DELETE は、特殊索引ディレクトリー (\$\$IMSDIR) の指定された内容を削除します。
- SCRATCH は、IMS.FORMAT および IMS.REFERAL ライブラリーおよびそれらのディレクトリーの指定された内容を消去します (SCRATCH は、IMS.TFORMAT にも機能します)。
- RELATE は、IMS.REFERAL ライブラリーの内容の解釈リストを作成します。
- LIST は、以下のいずれかの解釈リストを作成します。
  - IMS.FORMAT または IMS.TFORMAT ライブラリーの内容
  - IMS.FORMAT ライブラリーの特殊索引ディレクトリーの内容
  - IMS.SDFSRESL ライブラリーの MFS 装置特性テーブル (DFSUDT0x) の内容

サブセクション:

- 636 ページの『制約事項』
- 636 ページの『前提条件』
- 636 ページの『要件』
- 636 ページの『推奨事項』
- 636 ページの『入力と出力』
- 636 ページの『JCL 仕様』
- 638 ページの『ユーティリティー制御ステートメント』
- 642 ページの『INDEX 機能の記述』
- 643 ページの『DELETE 機能の記述』
- 643 ページの『SCRATCH 機能の記述』
- 645 ページの『RELATE 機能の記述』
- 646 ページの『LIST 機能の記述』

- 647 ページの『機能の記述のパラメーター』
- 648 ページの『MFS 装置特性テーブルの LIST』
- 649 ページの『LIST DEVCHAR 出力』
- 650 ページの『戻りコード』

## 制約事項

このユーティリティには、以下の制約事項が適用されます。

- MFS サービス・ユーティリティを MFS 言語ユーティリティ (MFSUTL プロシージャ) と同時に実行することは、両方が同じデータ・セットにアクセスしている場合は、行わないでください。
- MFS サービス・ユーティリティを IMS オンライン制御領域と同時に実行することは、両方がアクティブ形式ライブラリーにアクセスしている場合は、行わないでください。

## 前提条件

現在、DFSUTSA0 ユーティリティについて文書化されている前提条件はありません。

## 要件

現在、DFSUTSA0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSUTSA0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

現在、DFSUTSA0 の入力または出力ステートメントはありません。

## JCL 仕様

DFSUTSA0 ユーティリティには、プロシージャ・ステートメント、EXEC ステートメント、および DD ステートメントが必要です。

次の図は、MFS ライブラリーを保守する 1 ステップのプロシージャを示しています。このプロシージャは、システム定義のステージ 2 で IMS.PROCLIB に入れます。

```
//          PROC DEVCHAR=0,SYS2=,SOUT=A
//MFSRVC EXEC PGM=DFSUTSA0,REGION=4M,PARM='DEVCHAR=&DEVCHAR'
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//*
//*          PRINT FILES
//*
//SYSPRINT DD SYSOUT=&SOUT
//*          DCB=(RECFM=VBA,LRECL=137)
//SYSSNAP DD SYSOUT=&SOUT
//*          DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
//SYSUDUMP DD SYSOUT=&SOUT
//*
//*          REFERAL LIBRARY
```



```

//*
//REFIN DD DSN=IMS.&SYS2.REFERAL,DISP=OLD
//*
//* ON-LINE FORMAT LIBRARY
//*
//FORMAT DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
//*
//*
//* //SYSIN DD * MUST BE SUPPLIED BY
//* USER WITH INPUT CONTROL CARD STREAM
//*
//* ALL DISP=OLD SPECIFICATIONS OF THIS
//* PROCEDURE ARE REQUIRED .....
//*
//*
```

プロシージャ・ステートメント

プロシージャ・ステートメントは、次の形式でなければなりません。

DEVCHAR=0,SYS2=,SOUT=A

#### **DEVCHAR=**

装置特性テーブルを指定します。デフォルトは 0 です。

#### **SOUT=**

SYSOUT クラスを指定します。デフォルトは A です。

#### **SYS2=**

XRF 複合システムで「オプションの複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS2='IMSA.' のように、オペランドを後書きピリオドも含めて引用符で囲みます。

プロシージャの呼び出しの例

MFSRVC プロシージャを呼び出すための JCL ステートメントを、次に示します。

```

//MFSRVC JOB MSGLEVEL=1
// EXEC MFSRVC
//SYSIN DD *
        END
/*
```

#### **SYSIN DD**

入力制御ステートメント・データ・セットを定義します。

**EXEC** ステートメント

EXEC ステートメントは次の形式になっていなければなりません。

PGM=DFSUTSA0,REGION=250K,PARM='DEVCHAR=&DEVCHAR'

#### **REGION=**

MFS サービス・ユーティリティーを実行する領域のサイズを指定します。デフォルトは 4 MB です。

#### **PARM=**

PARM= フィールドは、以下の形式でなければなりません。

PARM='DEVCHAR=&DEVCHAR'

ただし、&DEVCHAR は、リストする装置特性テーブルです。

## DD ステートメント

### STEPLIB DD

IMS.SDFSRESL を指します。ここには、IMS 中核と必要なアクション・モジュールが入っています。

### SYSPRINT DD

出力メッセージ・データ・セットを定義します。このデータ・セットには、プリンターを指定することができます。あるいは、このデータ・セットを出力ストリームに入れて送ることもできます。DISP=(MOD,...) を指定する場合は、テープ・ボリュームまたは直接アクセス装置を指定できます。出力は 121 の倍数としてブロック化できます。

### SYSIN

順次データ・セット、または区分データ・セットのメンバーのいずれかの入力データ・セットを参照します。

### SYSSNAP

特定の重大エラーが検出された場合に、SNAP マクロからの出力の受け取りに使用されるデータ・セットを指します。

## ユーティリティー制御ステートメント

MFS サービス・ユーティリティー・プログラムで使用される制御ステートメントは、プリプロセッサに対する入力ソース・ステートメントと同じ構文と多くの同じキーワードを使用します。

例外: ステートメントまたはその一部にコメントが付けられる場合は例外です。

要件: 1 つ以上の FMT ブロックに許容される操作は非常に広範囲にわたるため、ユーティリティーではステートメントのコメントを /\* で開始する必要があります (例えば、LIST ALL /\* THIS STATEMENT...)

## 定位置パラメーター

### ALL

指定可能な場合は、関連する演算子に対してサポートされている機能をすべて呼び出すことを暗黙に指定します。例えば、INDEX ALL は、IMS.FORMAT に存在するすべての MID、MOD、DIF、および DOF の名前を、ブロックへの直接アクセスのためにオンライン制御領域により使用される特殊索引ディレクトリー (\$\$IMSDIR) に暗黙に挿入します。

### PDIR

IMS.FORMAT の PDS ディレクトリー項目に対して行われる関連操作、あるいは LIST または SCRATCH で使用される場合は、IMS.TFORMAT の PDS ディレクトリーに対して行われる関連操作の呼び出しを暗黙に指定します。例えば、LIST PDIR を指定すると、IMS.FORMAT または IMS.TFORMAT の内容が解釈された形式でリストされます。

### INDEX

オンライン制御領域で使用される特殊索引ディレクトリー (\$\$IMSDIR) に対する操作の呼び出しを指示します。

## REFER

ヒストリー中間テキスト・ストレージ・ライブラリーとして MFS 言語ユーティリティーで使用される IMS.REFERAL に対する操作の呼び出しを指示します。

## FORMAT

IMS.FORMAT に対する操作、あるいは LIST または SCRATCH で使用される場合は IMS.TFORMAT に対する操作の呼び出しを指示します。

## DEVCHAR

LIST 機能でのみ有効です。このパラメーターを指定すると、EXEC パラメーターの接尾部 (DEVCHAR=接尾部) によって識別される装置特性テーブルが印刷されます。EXEC パラメーターが指定されていない場合は、DFSUDT00 の内容が印刷されます。

## キーワード

### MSG=*name* | ALL

機能の呼び出しを指示します。

#### 名前

特定のメッセージ制御ブロック、MID、または MOD に対する機能の呼び出しを指示します。名前には、英字で始まる 1 から 8 文字の英数字の値を指定します。

#### ALL

すべてのメッセージ記述子に対する操作の呼び出しを指示します。

### FMT=*name* | ALL

操作の呼び出しを指示します。

#### 名前

特定の装置形式に対する操作の呼び出しを指示します。

#### ALL

すべての装置形式に対する操作の呼び出しを指示します。ALL 以外に何も修飾しない場合、操作は IMS.FORMAT のすべての FMT 制御ブロックに対して行われます。

FMT 制御ブロックは、IMS.FORMAT の複数の FMT 制御ブロック (DIF と DOF) から構成することができます。さらに修飾しない限り、同一のルート名を持つすべての FMT 制御ブロックに対して操作が行われます。「name」には、英字で始まる 1 から 6 文字の英数字の値を指定できます。

### TBL=*name*

SCRATCH 機能に対して、IMS.REFERAL ライブラリーから TBL ITB を消去するように指示します。このキーワードは、REFER 定位置パラメーターの指定がある場合に SCRATCH ユーティリティー制御ステートメントでのみ有効です。サービス・ユーティリティーの RELATE 機能を使用して、IMS.REFERAL に入っているすべての TBL の名前を入手できます。

### PDB=*name*

SCRATCH 機能に対して、IMS.REFERAL ライブラリーから PDB ITB を消去するように指示します。このキーワードは、REFER 定位置パラメーターの指定がある場合に SCRATCH ユーティリティー制御ステートメントでのみ有効で

す。サービス・ユーティリティの RELATE 機能を使用して、IMS.REFERAL に入っているすべての PDB の名前を入手できます。

**DEVCHAR=x**

「=」の後に指定される接尾部によって識別される装置特性テーブルを印刷します。このパラメーターは LIST 機能でのみ有効です。

**DEV=**

指定された FMT 制御ブロックを、特定の装置、2 次論理装置タイプ、またはリモート・プログラム、あるいはすべて (ALL) に適用されるように修飾します。

指定 装置

**3270** 3270 または SLU 2 ディスプレイ装置

**3270-A**

IMS システム定義の過程で装置シンボル名を用いて定義されたすべての 3270 または SLU 2 ディスプレイ装置

**3270-A1,...,A15**

IMS システム定義の過程で特定の装置シンボル名を用いて定義された単一の 3270 または SLU 2 ディスプレイ装置

**3270P** 3270 印刷装置

**FIN** 金融機関アプリケーション・プログラム

**FIDS** 金融機関表示コンポーネント (6x40) (例 : 3604-1 または 3604-2)

**FIDS3**

金融機関表示コンポーネント (12x40) (例 : 3604-3)

**FIDS4**

金融機関表示コンポーネント (16x64) (例 : 3604-4)

**FIDS7**

金融機関表示コンポーネント (24x80) (例 : 3604-7)

**FIJP** 金融機関ジャーナル印刷装置

**FIPB** 金融機関通帳印刷装置

**FIFP** 金融機関事務管理印刷装置

**DPM-A**

システム定義の過程でこの装置シンボル名を用いて定義されたすべての SLU P 装置

**DPM-B**

システム定義の過程でこの装置シンボル名を用いて定義されたすべての ISC ノード

**DPM-A1,...,A15**

SLU P

**DPM-B1,...,B15**

ISC ノード

**SCS1** 次のコンソール・キーボード/印刷装置 : 3767、NTO、3771、3773、3774、3775、3776、3777、および SLU 1 (印刷データ・セット) または SLU 4

**SCS2** 3521 カード・パンチ、3501 カード・リーダー、2502 カード・リーダー、および SLU 1 (伝送データ・セット) または SLU 4

**ALL** 指定の装置すべて

#### **MDL=1 | 2 | ALL**

FMT 制御ブロックの操作を決定します。

- 1 FMT 制御ブロックの操作を 3270/3270P の型式 1 の装置の制御ブロックに限定します。
- 2 FMT 制御ブロックの操作を 3270/3270P の型式 2 の装置の制御ブロックに限定します。

#### **ALL**

3270/3270P の型式 1 および型式 2 の両方の装置の制御ブロックを FMT 制御ブロックの操作の対象にします。

このキーワードは、3270 および 3270P 装置にのみ適用されます。

#### **DIV=INPUT | OUTPUT | INOUT**

FMT 制御ブロックの操作を決定します。

##### **INPUT**

FMT 制御ブロック操作を DIF に限定します。

##### **OUTPUT**

FMT 制御ブロック操作を DOF に限定します。

##### **INOUT**

DIF と DOF の両方を FMT 制御ブロックの操作の対象にします。

#### **FEAT=IGNORE | n[nn] | (list) | ANY**

FMT 制御ブロックの操作を決定します。

##### **IGNORE**

FMT 制御ブロックの操作を、MFS 言語ユーティリティーに対する DEV ステートメントで FEAT=IGNORE が指定された制御ブロックに限定します。

##### **n[nn]**

次のいずれかを指定します。

- 120、126、132 のいずれかの印刷行
- ユーザー定義の機能 1 から 10

##### **(list)**

FMT 制御ブロックの操作を、特定の機能の組み合わせを持つ制御ブロックに限定します。「list」に許可される指定は次のとおりです。

DEV=FIFP の場合:

DUAL  
132  
(DUAL,132)

DEV=3270 または DEV=3270-An の場合:

PEN            ,PFK            ,CARD  
**NOPE**        DEKYBD        **NOCD**  
                 **NOPEK**

実際のコーディングで指定を区切るのに必要な箇所にものみコマを入れてください。機能を指定する場合、その位置は関係しません。少なくとも 1 つの代替パラメーターをコーディングする必要があります。

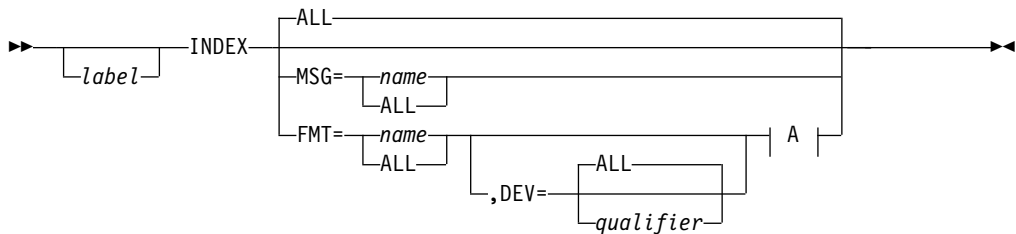
NOPE、NOPEK、または NOCD パラメーターの 1 つ、2 つ、あるいは 3 つのいずれかを指定しても、あるいはまったく指定しなくても、機能の値は同じ値になります。

#### ANY

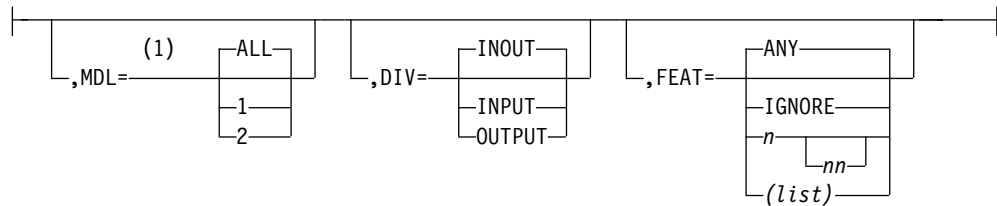
すべての制御ブロックに対して、機能指定に関する制約なしに、FMT 制御ブロックの操作を指示します。

### INDEX 機能の記述

INDEX 機能は、指定された制御ブロック (または関連する一連の制御ブロック) の名前を特殊索引ディレクトリー (\$\$IMSDIR) に入れます。この索引ディレクトリーの使用により、制御ブロックへの高速オンライン・アクセスが可能になります。



#### A:



#### 注:

1 MDL= は、3270 および 3270P 装置にのみ適用されます。

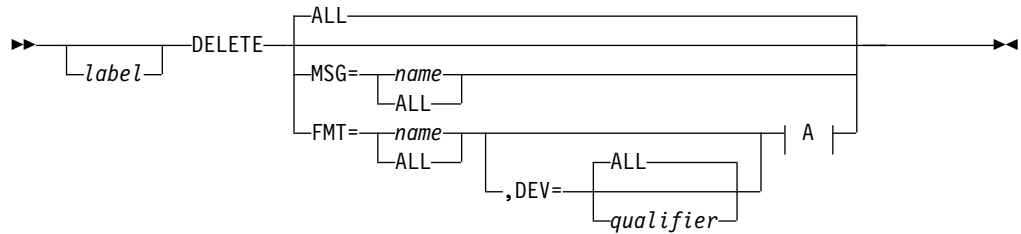
\$\$IMSDIR が INDEX 機能によって作成されると、それはオンライン IMS 制御領域の初期設定時に MFS バッファ・プールに読み込まれ、そのバッファ・プールに永続的に常駐します。要求された制御ブロックが MFS バッファ・プールにない場合、MFS プール・バッファ・マネージャーは次に \$\$IMSDIR の項目を検索します。該当する項目が見つかった場合、MFS プール・マネージャーはその制御ブロックの直接読み取りを出すことができます。

この特殊索引ディレクトリーを使用して 2 つの直接アクセス・ストレージ読み取り操作を 1 つの読み取り操作に置換すると、パフォーマンスは改善されますが、これはオンライン制御領域でのストレージ・コスト (MFS バッファ・プールで索引が

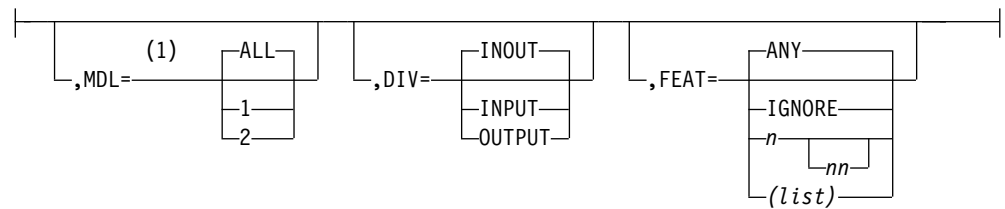
付けられた制御ブロック当たり 14 バイト) と比較検討しなければなりません。全体に占める割合が小さく、使用頻度の最も高い制御ブロックのみを索引付けの対象にすることをお勧めします。

## DELETE 機能の記述

DELETE 機能は、制御ブロックまたは一連の制御ブロックの名前を指定します。オンラインの MFS プール・マネージャーにより使用される特殊索引ディレクトリー (\$\$IMSDIR) からその指定した名前が削除されます。



**A:**

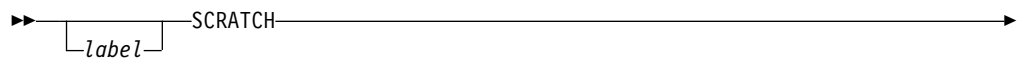


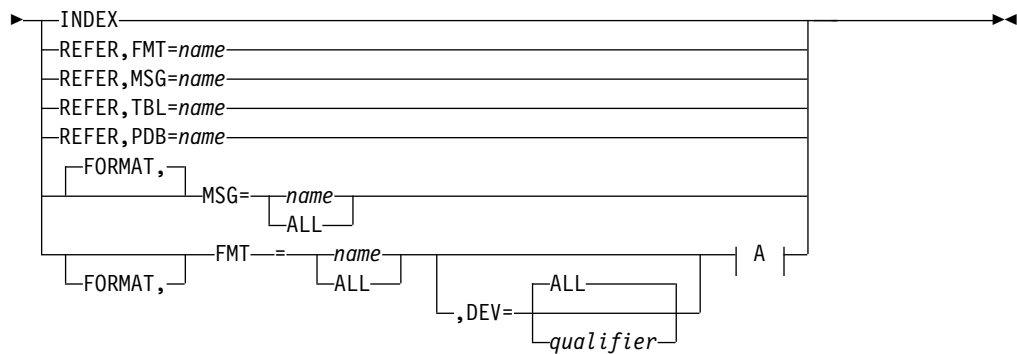
注:

- 1 MDL= は、3270 および 3270P 装置にのみ適用されます。

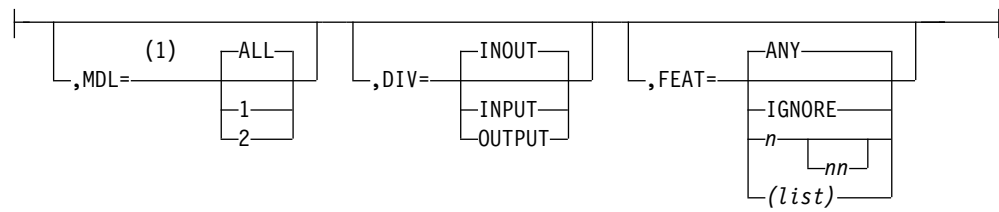
## SCRATCH 機能の記述

SCRATCH 機能は、メッセージ、形式、区画セット、またはテーブルの ITB を IMS.REFERAL から消去します。この機能を使用して、メッセージ記述子、装置形式、索引ディレクトリーを IMS.FORMAT または IMS.TFORMAT から消去することができます。





**A:**



注:

1 MDL= は、3270 および 3270P 装置にのみ適用されます。

SCRATCH 機能を効率的に使用するには、IMS.REFERAL の ITB と IMS.FORMAT または IMS.TFORMAT の制御ブロックとの間の関係を理解していただければなりません。次の図に示すように、形式セットは 1 つの形式とそれに関連するすべてのメッセージから構成されます。ここで、SOR= パラメーターはソースとしての形式を指定し、DFSDF2 はメッセージ DFSMI2、DFSMO2、および DFSMO3 のソースとして指定された形式です。この形式セットの ITB 形式は IMS.REFERAL に保管され、制御ブロック形式は IMS.FORMAT に保管されます。形式 ITB は、さまざまな装置タイプおよび装置機構の多くの装置形式に対応できます。



## IMS.REFERAL

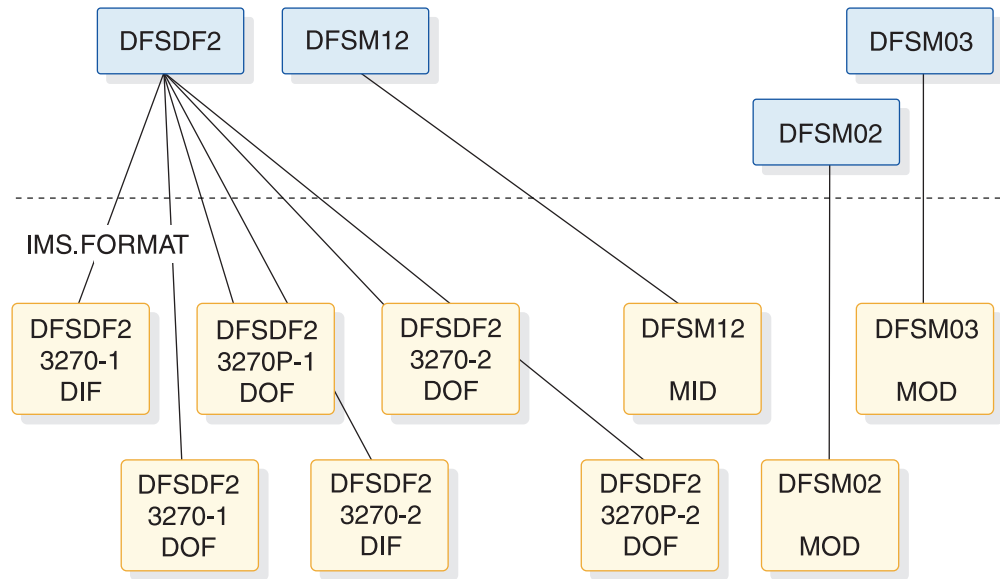


図 41. IMS.REFERAL にある ITB と、3270 形式 DFSDF2 およびその形式セットに対する制御ブロックの関係

制御ブロックを IMS.FORMAT から消去する操作は、一時的な操作です。つまり、次回その形式セットのメンバーのいずれかが言語ユーティリティによって処理されると、その制御ブロックはステージング・ライブラリーに復元されます。したがって、3270-1 の DFSDF2 DIF が IMS.FORMAT から消去され、その後 DFSDF2 あるいは関連メッセージのいずれか 1 つが言語ユーティリティによって処理された場合、3270-1 の DIF は再度 IMS.FORMAT に入れられます。

制御ブロックを完全に除去するには、その制御ブロックの ITB も除去する必要があります。3270-1 ソースを組み込まずに DFSDF2 形式定義を再コンパイルすると、DFSDF2 の 3270-1 制御ブロックを削除できます。

もう 1 つの方法として、MFS サービス・ユーティリティを使用して参照データ・セットから DFSDF2 ITB をすべて消去する方法があります。

ユーザーは、DELETE ALL または SCRATCH INDEX のいずれかを使用して、索引ディレクトリーを消去することができます。

例外: SCRATCH 機能は、MFS 装置特性テーブルには適用されません。

## RELATE 機能の記述

RELATE 機能は IMS.REFERAL ライブラリーのすべての FMT ITB、MSG ITB、PDB ITB、および TABLE ITB のリストを出力します。

```

▶▶-----RELATE-----▶▶
   [label]                [comments]
  
```

FMT ITB と MSG ITB が関連付けられます。各 FMT ITB 名に続いて、3 スペースの字下げの後に、SOR= パラメーターによって対応する FMT を参照する MSG

ITB の名前が出力されます。MSG ITB 項目は、メッセージが INPUT と OUTPUT のどちらであるかを示します。IMS.REFERAL には存在しないが、1 つ以上の MSG ITB によって参照される FMT ITB 項目の場合は、右隣に \*\*NOT DEFINED と示されます。PDB ITB と TABLE ITB は、FMT ITB と MSG ITB の後にリストされます。

次の図は、IMS.REFERAL の内容と、FMT ITB と MSG ITB の関係を示しています。例えば、最初の項目では、FMT 名が DFSDF1 で、DFSDF1 に関連する MSG が DFSMI1 (入力 MSG) と DFSMO1 (出力 MSG) の 2 つであることを示しています。

```

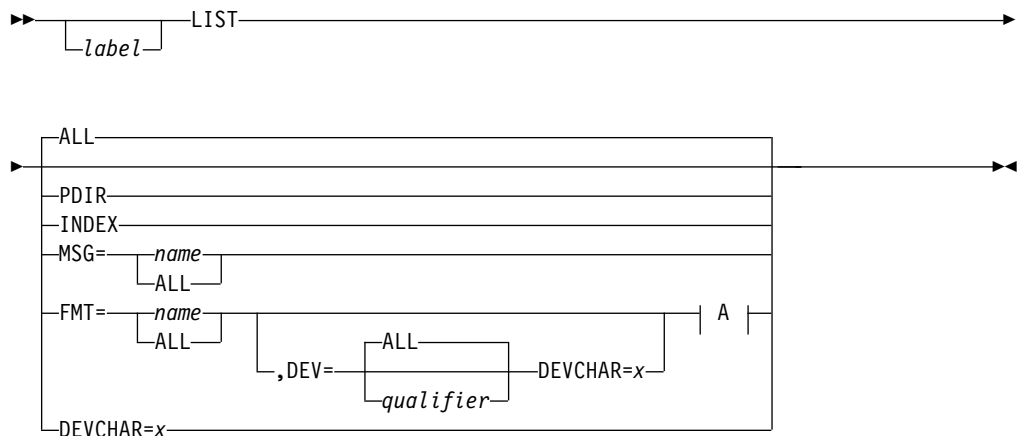
RELATE
DFSDF1
  DFSMI1      INPUT
  DFSMO1      OUTPUT
DFSDF2
  DFSDSP01    OUTPUT
  DFSMI2      INPUT
  DFSMO2      OUTPUT

  DFSMO3      OUTPUT
  DFSMO5      OUTPUT
DFSDF3
  DFSMI4      INPUT
  DFSMO4      OUTPUT
DSFDF5
  DFSMSTRI    INPUT
DFS1209I     PROCESSING TERMINATED BY EOD ON SYSIN

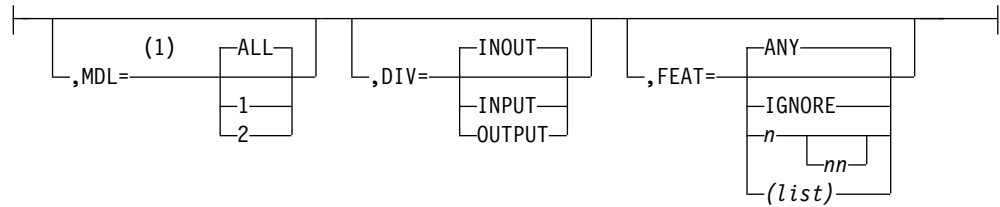
```

## LIST 機能の記述

LIST 機能は、テスト・ライブラリー IMS.TFORMAT、ステージング・ライブラリー IMS.FORMAT、または装置特性テーブル DFSUDT0X のいずれかの内容の解釈リストを出力します。



**A:**



注:

- 1 MDL= は、3270 および 3270P 装置にのみ適用されます。

## 機能の記述のパラメーター

ステージング・ライブラリーとテスト・ライブラリーの場合、INDEX、DELETE、SCRATCH、RELATE、および LIST 制御ステートメントを次のように修飾することにより、リストする内容の選択が可能です。

### ALL

PDS ディレクトリーと索引ディレクトリーの両方を指定します。デフォルトは ALL です。

### PDIR

PDS ディレクトリーを指定します。

### INDEX

索引ディレクトリー \$IMSDIR を指定します。

### MSG

1 つまたはすべてのメッセージ記述子に関する PDS ディレクトリー項目を指定します。

### FMT

1 つ以上の装置形式に関する PDS ディレクトリー項目を指定します。

以下のようにイベントがチェーニングして起こると、特殊なケースになります。

1. 索引ディレクトリー (\$IMSDIR) が存在しない時点でこのサービス・ユーティリティーが開始される。
2. その後で、索引ディレクトリーが INDEX 機能により作成される。
3. LIST PDIR または LIST ALL が呼び出される。

このような場合、PDS ディレクトリーの内容が含まれる出力には、索引ディレクトリーの項目は含まれません。これは、サービス・ユーティリティーが終了するまで、索引ディレクトリーがストレージに維持され、形式ライブラリーに書き込まれないために発生します。

出力リストには、制御ブロックごとに以下のフィールドを持つ 1 行が含まれます。

### NAME

MSG または FMT ステートメントで指定された制御ブロックの名前を表します。

### TYPE

制御ブロックのタイプを表します。

**DIF**  
装置入力形式

**DOF**  
装置出力形式

**MSG**  
メッセージ入力または出力制御ブロック

**DEV**  
FMT 制御ブロックの場合に、制御ブロックが適用される装置タイプを表します。

**MDL**  
FMT 制御ブロックの場合に、制御ブロックが適用される装置タイプの型式を表します。このフィールドは、3270 装置と 3270P 装置で使用されます。

**TTR**  
IMS.FORMAT 内の制御ブロックの位置を表します (16 進数)。このフィールドは、IMS.FORMAT の項目に対してのみ有効です。LIST INDEX には無効です。

**SIZE**  
制御ブロックのサイズをバイト数で表します (16 進数)。

**FEAT**  
FMT 制御ブロックの場合に、制御ブロックが適用される装置タイプ、型式、特殊機構を、16 進の非解釈表記で表します。この情報はライブラリーの FMT 名に追加されるので、この数値によって出力リストの照合順序が決定されます。

**FEATURES**  
FMT 制御ブロックの場合に、その制御ブロックが適用される特定の装置機構を表します。

以下の図は、645 ページの図 41 で説明している制御ブロックに対する LIST 機能の出力リストの例を示しています。

NAME	TYPE	DEV	MDL	TTR	SIZE	FEAT	INTERPRETED FEATURES
DFSDF2	DIF	3270	1	000705	000001AE	007F	IGNORE
DFSDF2	DOF	3270	1	000703	00000279	007F	IGNORE
DFSDF2	DOF	3270P	1	00070B	000001A5	017F	IGNORE
DFSDF2	DIF	3270	2	000709	000000B6	027F	IGNORE
DFSDF2	DOF	3270	2	000707	00000296	027F	IGNORE
DFSDF2	DOF	3270P	2	00070D	00000195	037F	IGNORE
DFSM12	MSG			00070F	0000005E		
DFSM02	MSG			000711	00000006		
DFSM03	MSG			000713	000000F3		

## MFS 装置特性テーブルの LIST

IMS システム定義の過程で作成された装置特性テーブル DFSUdT0x に対して、LIST DEVCHAR または DEVCHAR=0 または x (x はテーブルの接尾部) を指定すると、指定された MFS 装置特性テーブルのすべての項目をリストした解釈リストが出力されます。このリストには、テーブルの項目ごとに、3270 装置または SLU 2 装置の装置シンボル名、画面サイズ、および機構が示されます。接尾部が、LIST ステートメントの DEVCHAR パラメーターまたは EXEC ステートメントの PARM オペランドに指定されていない場合、テーブル DFSUdT00 がリストされま

す。LIST ステートメントの DEVCHAR パラメーターに接尾部が指定されていないが、EXEC ステートメントには指定がある場合、テーブル DFSUDT0x (x は EXEC ステートメントに指定されているテーブル接尾部) がリストされます。図 42 と図 43 に、LIST DEVCHAR 機能の例を示しておきます。

図 42. DEVCHAR= を指定した LIST DEVCHAR 機能の例

```
//MFSRVC EXEC PGM=DFSUTSA0, PARM='DEVCHAR=3'
//SYSIN DD *
LIST DEVCHAR=3
LIST DEVCHAR=7
```

最初の LIST ステートメントは、装置特性テーブル DFSUDT03 の内容をリストします。

2 番目の LIST ステートメントは、装置特性テーブル DFSUDT07 の内容をリストします。

図 43. LIST DEVCHAR 機能の例

```
//MFSRVC EXEC PGM=DFSUTSA0
//SYSIN DD *
LIST DEVCHAR=0
LIST DEVCHAR=7
```

最初の LIST ステートメントは、装置特性テーブル DFSUDT00 の内容をリストします。

2 番目の LIST ステートメントは、装置特性テーブル DFSUDT07 の内容をリストします。

## LIST DEVCHAR 出力

出力リストには、指定された装置特性テーブルの項目ごとに、以下のフィールドを含む 1 行が示されます。

### Symbolic Name

IMS システム定義の TYPE または TERMINAL マクロの TYPE= オペランドで、3270 ディスプレイに対して定義したシンボル名

### Screen Size

IMS システム定義の TYPE または TERMINAL マクロの SIZE= オペランドで定義した 3270 ディスプレイの画面の行および桁

### Device Features

IMS システム定義の TYPE または TERMINAL マクロの FEAT= オペランドで指定されている機構

以下の図は、LIST DEVCHAR 出力の例です。

SYMBOLIC NAME	SCREEN SIZE LINES COLS	DEVICE FEATURES
3270-A01	12 80	CARD,PFK,PEN
3270-A02	24 80	IGNORE
3270-A03	32 80	CARD,DEKYBD,PEN

## 戻りコード

エラー・メッセージ ERR TYPE x IN REFERAL LIBRARY, FUNCT=RELATE は、MFS REFERAL ライブラリーに問題が存在する場合に出されます。エラー・タイプは次のとおりです。

- 1 ディレクトリーのブロック長エラー
- 2 ユーザー付加部分の長さエラー
- 3 不明ブロック・エラー
- 4 重複 DUMMY FORMAT エラー
- 5 重複 FORMAT エラー
- 6 TABLE ブロック・エラー
- 7 メッセージ入力ブロック・エラー
- 8 REFIN OPEN エラー
- 9 REFIN OPEN SYNAD がとられた
- A PDB ブロック・エラー
- B FORMAT ブロック・エラー
- C メッセージ出力ブロック・エラー

SYSIN、SYSPRINT、REFIN <REFERAL> ファイルにエラーが検出された場合、戻りコードはそれぞれ 4、8、12 に設定されます。

RELATE 機能にエラーが検出された場合、後続の機能は無視されます。

---

## 第 28 章 複数システム検査ユーティリティ (DFSUMSV0)

複数システム検査ユーティリティ (DFSUMSV0)を使用して、マルチシステム環境で IMS システムのシステム定義の整合性と互換性を検査します。このユーティリティは、すべての複数システム結合 (MSC) リンク・タイプの IMS MSC で使用します。

このユーティリティは、IMS システムの正常な動作を妨げる可能性があるエラーを識別します。

このユーティリティを使用しない場合には、システム定義の互換性を手動で検査しなければなりません。

サブセクション:

- 『制約事項』
- 652 ページの『前提条件』
- 652 ページの『要件』
- 652 ページの『推奨事項』
- 653 ページの『入力』
- 656 ページの『出力』
- 662 ページの『JCL 仕様』
- 663 ページの『ユーティリティ制御ステートメント』
- 664 ページの『戻りコード』

### 制約事項

複数システム検査ユーティリティには下記の制約があります。

- システム間通信 (ISC) に関連するエラーを検出することはできません。
- CICS はサポートしません。
- 指定ルーティングに関連するエラーを検出することはできません。
- 検査ユーティリティの 1 回の実行で検査できる IMS システムの数は、2 から 936 です。
- このユーティリティは、共用キュー・グループ内の IMS サブシステムの検査に使用することはできません。同じ共用キュー・グループ内の IMS サブシステム相互間の MSC リンクはサポートされず、メッセージ「DFS2149 - RESTART ABORTED」が出される原因になります。
- リモート論理端末 (LTERM) の定義が誤っている場合、ユーティリティはリモート LTERM に対してのみエラーを認識してエラー・メッセージを発行し、ターゲット領域にあるローカル LTERM は検出されません。
- IMS は、初期設定される IMS システム内で定義された MSC リンクに関連する MSC 記述子だけを処理します。その他の MSC 記述子はすべて無視されます。
- 異なるリリース・レベルを持つ IMS システムを検査するときは、最新のリリース・レベルからのユーティリティを使用してください。

- DFSUMSV0 は、動的リソース定義 (DRD) によって定義されたりソースを検査することができません。別の方法として、/MSVERIFY コマンドを使用して、それらのリソースを検査できます。

## 前提条件

複数システム検査ユーティリティには下記の前提条件があります。

このユーティリティを実行する前に、マルチシステム構成全体に組み込まれるすべてのマルチシステム制御ブロックにある IMS 定義エラーを、すべて解決しなければなりません。すべてのシステム定義を完成した後、すべての IMS マルチシステム定義からのすべてのマルチシステム制御ブロックを、IMS.SDFSRESL またはそれ以外の何らかのユーザー指定ライブラリーにバインドする必要があります。ユーティリティはマルチシステム制御ブロックにアクセスできるようになります。問題判別を行う場合、この検査ユーティリティの実行時に、すべての IMS マルチシステム制御ブロックを取めたアセンブリー・リストを使用可能にしてください。アセンブリー・リストを使用しないと、マルチシステム検査処理を実行した結果示される不整合や非互換性の解決が困難になります。

検査ユーティリティを実行する前に、検査されるすべてのシステムのマルチシステム制御ブロック・モジュールを、IMS.SDFSRESL または他のユーザー指定のライブラリーにロードしておかなければなりません。以下の図は、マルチシステム制御ブロック・モジュールを IMS.SDFSRESL にロードするためのサンプル・ジョブ・ストリームを示しています。

```
//STEP1 EXEC PGM=IEWL,PARM='SIZE=(500K),NCAL,LET,REUS,XREF,LIST'  
//SYSPRINT DD SYSOUT=A  
//SYSLMOD DD DSN=IMS.SDFSRESL,DISP=OLD  
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(100,10))  
//SYSLIN DD *  
PLACE OBJECT MODULES HERE  
NAME DFSMSxxx(R)  
/*
```

ユーザーは、複数のオブジェクト・モジュールを用意しなければなりません。先行のマルチシステム制御ブロック・モジュールを識別する NAME ステートメント (xxx を制御ブロック名の 3 桁の接尾部で置き換える) を、それぞれのオブジェクト・デッキの後に指定してください。IMS.SDFSRESL 以外のライブラリーを使用する場合、そのライブラリーに応じて //SYSLMOD ステートメントを修正してください。

## 要件

このユーティリティには、以下の要件が適用されます。

検査ユーティリティは、SYSIN データ・セットの 1 つ以上の制御ステートメントを入力として要求します。

## 推奨事項

DFSUMSV0 ユーティリティには、その使用と実行に関連して、いくつかの推奨事項があります。



オンライン実行を行う前に、この複数システム検査ユーティリティを実行して、定義されたリンクとルーティング・パスをすべて検査してください。

IMSCtrl マクロで SYSTEM キーワードに MSVERIFY パラメーターが指定されている場合は、IMS マルチシステム制御ブロックと検査ユーティリティだけが生成されます。

## 入力

このユーティリティは、次の 2 つのフェーズで入力を処理します。

- 入力の妥当性検査
- マルチシステム制御ブロックの検査

### 入力の妥当性検査

入力 (SYSIN データ・セット内の制御ステートメント) の検査が完了した後、有効なマルチシステム制御ブロック名のリストが印刷されます。次に、検査ユーティリティは、マルチシステム制御ブロック名が IMS.SDFSRESL PDS ディレクトリーにあるかどうかを判別します。ユーティリティは、ディレクトリーに見付からなかった制御ブロック名をすべて印刷します。エラーが検出された場合、検査ユーティリティは実行を終了して完了コード 12 を戻します。エラーが検出されなかった場合、ユーティリティはマルチシステム制御ブロックを実ストレージにロードします。

### マルチシステム制御ブロックの検査

このユーティリティは、各マルチシステム制御ブロックの以下の特定の部分を検査します。

- パートナー ID および割り当てられた物理リンク
- リモート SYSID からローカル SYSID へのパス
- リモートおよびローカルのトランザクション属性
- 対応する論理端末の有無

### パートナー ID および割り当てられた物理リンク

論理リンク定義のパートナー ID は、以下に該当するか否かを確認するために検査されます。

- 1 つのシステムだけで参照されているのではない
- 2 つのシステムだけで参照されている

MSLINK マクロの PARTNER キーワードで定義された各パートナー ID は、他のすべてのマルチシステム制御ブロックの他のすべてのパートナー ID に対比してチェックされます。エラーが検出されると、該当するメッセージが印刷されます。エラーの論理リンクは、検査処理の以降のステップでは未定義のリンクとして処理されます。

パートナー ID を検査して、両方のシステムで該当の論理リンクに対して MSPLINK (物理リンク) が定義されている場合、この物理リンク属性のタイプとバッファー・サイズが検査されます。物理リンクのタイプは、次のとおりです。

- 実ストレージ間
- チャンネル間
- 仮想記憶通信アクセス方式
- 伝送制御プロトコル/インターネット・プロトコル

互換性のない物理リンクが検出されると、両方の物理リンク定義の属性が表示され、ユーザーがエラーを判別するのに役立ちます。MSPLINK が 1 つの論理リンクに対してのみ定義されている場合、相手側が定義されていないことを示す通知メッセージが印刷されます。システム定義で IMS に対して定義されている MSC 物理リンクと論理リンクに加えて、ユーザーは MSC 記述子を使用して IMS に対してリモート名を識別することができます。MSC 記述子によって、各リモート・リソースと生成された MSNAME マクロのリンク名が関連付けられます。

リモート **SYSID** からローカル **SYSID** へのパス

SYSID に対する SYSID テーブル項目について、その SYSID 番号のマルチシステム制御ブロックがすべて検査され、パス・エラーの有無が検査されます。パス・エラーとは、SYSID がリモートとして定義されているマルチシステム制御ブロックと、SYSID がローカルとして定義されているマルチシステム制御ブロックとの間のパスが不完全であることです。

不完全なパスは、以下のような理由で生じる場合があります。

- 中間システムのマルチシステム制御ブロックの SYSID に、MSNAME ブロックのアドレスが含まれていない (未定義の SYSID)。
- MSNAME ブロックが割り当てられている論理リンク・ブロックにローカル SYSID がなく、この論理リンク・ブロック自体に戻るパスがある (ループ状態)。
- MSNAME ブロックが関連付けられている論理リンク・ブロックのパートナー ID が無効である。無効なパートナー ID とは、他のどのようなマルチシステム制御ブロックにも定義されていないパートナー ID、あるいは複数のマルチシステム制御ブロックで定義されているパートナー ID です。
- SYSID 番号が複数のシステムでローカルとして定義されている。
- SYSID 番号がいずれのシステムでもローカルとして定義されていない。

すべての論理リンク・パスが検査されるまで、SYSID は番号順にスキャンされます。パスの検査が終わると、検査ユーティリティーが警告メッセージを表示することがあります。このようなメッセージは、ローカル SYSID への無効なパスの原因となるので、この SYSID 番号 -MSNAME を割り当ててはならない論理リンク番号を示します。

リモートおよびローカルのトランザクション属性

各マルチシステム制御ブロックでリモート・トランザクション定義がスキャンされます。各リモート・トランザクション定義は、リモート SYSID とローカル SYSID を参照します。各リモート・トランザクション・コードは、参照するリモート SYSID がローカルとして定義されているシステムのトランザクション・コードと比

較されます。一致するトランザクション・コードがない場合、エラー・メッセージが印刷されます。一致するコードが検出されると、2つのマルチシステム制御ブロックでその属性が検査されます。

以下のトランザクション・コード属性は、両方のシステムで一致していなければなりません。

- ローカル
- リモート
- リカバリー可能
- リカバリー不能
- 会話型
- 固定長スクラッチ域
- 固定長スクラッチ域の長さ
- 非照会
- 照会
- 単一セグメント
- 複数セグメント
- 非高速機能

トランザクション属性に矛盾がある場合、エラー・メッセージが印刷されます。両方のシステムでトランザクションに対して指定されている属性が表示されます。

ローカル・システムから対応するリモート・トランザクションへ戻るパスが存在していることを確認するために、ローカル・トランザクションからの戻りが検査されます。エラーがあると、メッセージが印刷されます。

対応する論理端末の有無

各マルチシステム制御ブロックでリモート LTERM 定義がスキャンされます。各 LTERM は、MSNAME ブロックに関連付けられています。各 MSNAME ブロックにはリモート SYSID 定義とローカル SYSID 定義が入っています。

リモート LTERM が検出されると、検査ユーティリティーは MSNAME のリモート SYSID がローカルとして定義されているシステムで、対応する LTERM が同じ名前で作られていることを確認します。対応する LTERM 定義がない場合、エラー・メッセージが印刷されます。LTERM が検出された場合には、検査が続行されます。

IMS システム定義の過程で定義されたマルチシステム LTERM の場合、リモート LTERM は、リモート・システムの IMS システム定義で定義するか、またはリモート・システムの拡張端末オプション (ETO) MSC 記述子によって定義することができます。ETO MSC 記述子を使用する場合には、リモート IMS システムの ETO 機能が初期設定されるまでは、そのリモート LTERM は存在しません。したがって、このユーティリティーは、欠落したリモート LTERM についてのエラー・メッセージを印刷し、該当の LTERM が動的リソースである可能性を示します。

ユーティリティーは、宛先 LTERM からの戻りパスをチェックして、対応するリモート LTERM に戻るローカル・システムからのパスがあることを確認します。エラーがあると、メッセージが印刷されます。

そのシステムへの戻りパスは、LTERM がリモートとして定義されていたマルチシステム制御ブロック内の MSNAME ブロックでローカルとして定義されている SYSID です。

すべての検査作業が終了すると、ユーティリティーは、システムの構成を視覚化する際に補助としてパス・マップを印刷します。

## 出力

検査ユーティリティーの出力には、通知メッセージ、警告メッセージ、エラー・メッセージ、およびパス・マップが含まれます。

あるエラーがさらに他のエラー条件を発生させる可能性があるので、まず最初により小さい番号のメッセージを分析し、該当するエラーを訂正します。パス・マップは、検査ユーティリティーが検査したルーティング・パスのマトリックス形式の要約です。このマップは、(マルチシステム制御ブロック名による昇順で) 検査される最初の 18 以下のシステムについて作成されます。19 以上のシステムを検査する場合は、すべてのシステムに対して検査が行われ、すべてのシステムについて該当のメッセージが作成されますが、パス・マップは最初の 18 のシステムだけを対象として作成されます。

次の図は、DFSMS001、DFSMS002、および DFSMS003 という名前のマルチシステム制御ブロックを持つ 3 つのシステムから成る構成の、エラー・メッセージとパス・マップの例を示しています。(説明を簡単にするため、以降の説明では特定のシステムを参照する場合にこれらの名前を使用しています。) このパス・マップとエラー・メッセージはより大きい報告書から抽出したものであるため、すべてのエラー・メッセージが示されているわけではありません。

この図には 3 つのセクションがあります。上のセクションはエラー・メッセージのリストであり、中央と下のセクションにはパス・マップが示されています。中央のセクションは SYSID を特定のシステムに関連付けます。下のセクションでは、特定のシステムの間論理リンクとパートナー ID を関連付けます。

DFS2311X PARTNER ID AC DEFINED MORE THAN TWICE AND IGNORED.

DFS2317X PARTNER ID AJ IN LOGICAL LINK 009 WITHIN DFSMS001 IS NOT DEFINED IN A SECOND SYSTEM.

DFS2311X PARTNER ID AK DEFINED MORE THAN TWICE AND IGNORED.

DFS2311X PARTNER ID SA DEFINED MORE THAN TWICE AND IGNORED.

DFS2311X PARTNER ID SB DEFINED MORE THAN TWICE AND IGNORED.

DFS2311X PARTNER ID SC DEFINED MORE THAN TWICE AND IGNORED.

	....MS001	A ....MS002	....MS003	....
0001	....	....	....	....
0002	....	....	....	....
0003	B ....	....	....	....
0004	....	....	....	....
0005	....	....	....	....
0006	....	....	....	....
0007	....	....	....	....
*0008	....	....** AM	....	....
0009	....LOCAL	....001 AL	....	....
*0010	....LOCAL	....*** AK	....LOCAL	....
0011	....LOCAL	....001 AB	....002 BC	....
0012	....LOCAL	....001 AB	....002 BC	....
*0013	....LOCAL	....*** AC	....*** AC	....
0014	....LOCAL	....001 AD	....002 BE	....
0015	....LOCAL	....001 AE	....002 BF	....
0016	....LOCAL	....001 AF	....002 BG	....
*0017	....LOCAL	....*** AC	....002 BH	....
*0018	....LOCAL	....LOCAL	....001 CA	....
*0019	....LOCAL	....LOCAL	....002 BF	....
*0020	....*** AK	....LOCAL	....*** AK	....
0021	....002 AB	....LOCAL	....002 BC	....
0022	....002 AB	....LOCAL	....002 BC	....
*0023	....*** AC	....LOCAL	....002 BD	....
0024	....002 AD	....LOCAL	....002 BE	....
0025	....002 AE	....LOCAL	....002 BF	....

図 44. エラー・メッセージとマルチシステム・パス・マップの例

0026	....002 AF	....LOCAL	....002 BG	....
0027	....002 AG	....LOCAL	....002 BH	....
*0028	....	....001 AG	....	....
*0029	....002 AF	....001 AF	....LOCAL	....
0030	....	....	....	....
0031	....002 AB	....003 BC	....LOCAL	....
0032	....002 AB	....003 BC	....LOCAL	....
*0033	....*** AC	....003 BD	....LOCAL	....
0034	....002 AD	....003 BE	....LOCAL	....
0035	....002 AE	....003 BF	....LOCAL	....
*0036	....*** AK	....003 BG	....LOCAL	....
0037	....	....003 BH	..	....
0038	....003 CA	....	....LOCAL	....
0039	....	....	....	....
0040	....002 AL	....LOCAL	....	....
*0041	....LOCAL	....	....002 BF	....
0042	....002 AH	....LOCAL	....	....
*0043	....002 AH	....001 AH	....	....
0044	....LOCAL	....001 AH	....	....
*0045	....LOCAL	....LOCAL	....	....
*0046	....LOCAL	....LOCAL	....	....
0047	....	....	....	....
0048	....	....	....	....
0049	....	....	....	....
0050	....	....	....	....
*0051	....002 AG	....	....LOCAL	....
0052	....	....	....	....
*0053	....002 AI	....	....	....
*0054	....** AJ	....	....	....
0055	....	....LOCAL	....	....
0056	....	....	....	....
0057	....	....	....	....
0058	....	....	....	....
0059	....	....	....	....
*0060	....*** SA	....*** SA	....	....
*0061	....LOCAL	....LOCAL	....	....
*0062	....*** SB	....*** SB	....	....
0063	....LOCAL	....001 AI	....	....
*0064	....*** SC	....*** SC	....	....
*0065	....LOCAL	....LOCAL	....	....
0066	....	....LOCAL	....	....
0067	....	....	....	....
0068	....	....	....	....
0069	....	....	....	....

0070	.....	.....	.....	.....
0071	.....LOCAL	.....001 BW	.....	.....
0072	.....LOCAL	.....001 BX	.....	.....
0073	.....LOCAL	.....001 BY	.....	.....
0074	.....LOCAL	.....001 TA	.....	.....
0075	.....LOCAL	.....001 TB	.....	.....
0076	.....LOCAL	.....	.....001 TC	.....
0077	.....LOCAL	.....001 XA	.....	.....
0078	.....LOCAL	.....001 XB	.....	.....
0079	.....LOCAL	.....	.....001 XC	.....
*0080	.....*** AK	.....003 BF	.....LOCAL	.....
0081	.....	.....	.....	.....
0082	.....	.....	.....LOCAL	.....
0083	.....	.....	.....	.....
0084	.....002 TA	.....LOCAL	.....	.....
0085	.....002 TB	.....LOCAL	.....	.....
0086	.....	.....LOCAL	.....002 TD	.....
0087	.....002 XA	.....LOCAL	.....	.....
0088	.....002 XB	.....LOCAL	.....	.....
0089	.....	.....LOCAL	.....002 XD	.....
*0090	.....*** AK	.....LOCAL	.....	.....
0091	.....002 BW	.....LOCAL	.....	.....
0092	.....002 BX	.....LOCAL	.....	.....
0093	.....002 BY	.....LOCAL	.....	.....
0094	.....	.....LOCAL	.....002 BP	.....
0095	.....	.....LOCAL	.....002 BQ	.....
0096	.....	.....LOCAL	.....002 BR	.....
0097	.....002 TB	.....LOCAL	.....001 TC	.....
0098	.....	.....	.....	.....
0099	.....	.....	.....	.....
0100	.....LOCAL	.....	.....001 BS	.....
0101	.....LOCAL	.....	.....001 BT	.....
0102	.....LOCAL	.....	.....001 BU	.....
0103	.....LOCAL	.....001 AN	.....	.....
0104	.....002 AN	.....LOCAL	.....	.....
0105	.....	.....	.....	.....
0106	.....	.....	.....	.....
0107	.....	.....	.....	.....
0108	.....	.....	.....	.....
0109	.....	.....	.....	.....
0110	.....003 BS	.....	.....LOCAL	.....
0111	.....003 BT	.....	.....LOCAL	.....
0112	.....003 BU	.....	.....LOCAL	.....

```

0113      ....      ....003 BP      .... LOCAL      ....
0114      ....      ....003 BQ      .... LOCAL      ....
0115      ....      ....003 BR      .... LOCAL      ....
0116      ....003 TC      ....      .... LOCAL      ....
0117      ....      ....003 TD      .... LOCAL      ....
0118      ....003 XC      ....      .... LOCAL      ....
0119      ....      ....003 XD      .... LOCAL      ....
0120      ....003 TC      ....001 TB      .... LOCAL      ....
*2036      .... LOCAL      ....003 BF      ....002 BG      ....
-----
VTAM      ... AB.001-----AB.001.      .....
* CTC      ... AC.002-----AC.002-----AC.002.      .....
MTM      ... AD.003-----AD.003.      .....
          ... AE.004-----AE.007.      .....
          ... AF.005-----AF.009.      .....
          ... AG.006-----AG.010.      .....
          ... AH.007-----AH.011.      .....
          ... AI.008-----AI.012.      .....
* MTM      ... AJ.009.      .....
*VTAM      ... AK.010-----AK.013-----AK.011.      .....
VTAM      ... AL.011-----AL.014.      .....
VTAM      ... AN.028-----AN.033.      .....
MTM      ... BS.017-----BS.013.      .....
CTC      ... BT.019-----BT.015.      .....
VTAM      ... BU.021-----BU.017.      .....
MTM      ... BW.016-----BW.021.      .....
CTC      ... BX.018-----BX.023.      .....
VTAM      ... BY.020-----BY.025.      .....
VTAM      ... CA.012-----CA.007.      .....
*VTAM      ... SA.013-----SA.018-----SA.008.      .....
*VTAM      ... SB.014-----SB.019-----SB.009.      .....
*VTAM      ... SC.015-----SC.020-----SC.010.      .....
TCP      ... TA.022-----TA.027.      .....
TCP      ... TB.023-----TB.028.      .....
TCP      ... TC.024-----TC.019.      .....
-----
*      = AN ERROR WAS DETECTED ON THIS LINE.
**     = NO PARTNER FOR ID.
***    = MULTI-PARTNERS FOR ID.

DFS2399I JOB TERMINATED - RETURN CODE 12

```

図に関する注:

英字 意味

- A** 一番上の行は、このユーティリティの実行によって検査されたシステムのマルチシステム制御ブロック名 (DFS 接頭部は含まない) を示しています。
- B** 図の中央のセクションの最初の列には、マルチシステム構成で定義されてい



る SYSID がすべて示されます。SYSID 番号の前にアスタリスクがある場合、マトリックスの該当の行にエラーがあることを示します。

- C** 図の中央のセクションの各項目は、SYSID (列 B) をマルチシステム制御ブロック名 (行 A) に関連付けます。
- ほとんどの項目には、システム (行 A) に論理的にリンクするように定義されたシステムのマルチシステム制御ブロック名の 4 桁の接尾部と、この論理リンクに定義された 2 文字のパートナー ID が含まれています。
  - DFSMS002 の SYSID 0041 などのブランクの項目は、システムに対してその SYSID が定義されていないことを示します。SYSID 0001 から 0007 に対する項目はすべてブランクになっています。これは、これらの SYSID がマルチシステム制御ブロックのいずれにも指定されていなかったことを示します。
  - LOCAL を指定する項目、例えば、DFSMS001 の SYSID 0009 などは、該当のシステムがその SYSID をローカルとして定義していることを示します。
  - エラーはアスタリスクによって示されます。SYSID の前にアスタリスクが 1 つある場合は、その印刷行に 1 つ以上のエラーが検出されたことを示します。DFSMS002 の SYSID 0008 のように、項目の接尾部が 2 つのアスタリスク (\*\*) で置き換えられている場合には、検査ユーティリティーがこのシステムに対するパートナーを検出しなかったことを示します。DFSMS002 の SYSID 0017 のように、3 つのアスタリスク (\*\*\*) が示されている場合は、3 つ以上のパートナーが検出されたことを示します。1 つの SYSID が複数のシステムでローカルとして定義されている場合、該当する印刷行は、1 つのアスタリスクと、LOCAL を指定する該当の行の複数の項目により識別されます。SYSID 0010 は、このようなエラーの例を示しています。

- D** 下のセクションの最初の列には、物理リンクのタイプが以下によって示されます。

**CTC** チャンネル間アダプター

**MTM** 実ストレージ間

**TCP** 伝送制御プロトコル/インターネット・プロトコル (TCP/IP)

**VTAM®**

仮想記憶通信アクセス方式

この列の項目は、物理リンクが定義されていない場合はブランクであり、E により識別される論理リンクの場合には、最初に検出された物理リンクに従って割り当てられます。リンク・タイプの前のアスタリスク (物理リンクが定義されていない場合は、アスタリスクだけ) は、該当の印刷行にエラーがあることを示します。

- E** パートナー ID と相対論理リンク番号によって論理リンクを識別します。パートナー ID は表の上の部分に示されているパートナー ID に直接関連しています。

検査ユーティリティーは、パートナー・システムを破線で結ぶことによってそれらに関連付けます。

この図では、パートナー関係 AC、AJ、AK、SA、SB、および SC がエラーになっています。このエラーについては表の上の部分で既に識別されていますが、下の部分ではさらに明確に示されています。パートナー ID BC は 3 つ以上の定義を持ち、一方、AX は 1 つの定義だけを持っています。

## JCL 仕様

DFSUMSV0 ユーティリティには、プロシージャー・ステートメント、EXEC ステートメント、およびプロシージャーの呼び出しが必要です。

以下の図は、複数システム検査ユーティリティの実行に使用するプロシージャーの例を示しています。IMSMSV は、SMP/E 処理中に作成され、IMS.SDFSPROC に入れられます。

```
//          PROC   DSN='IMS.SDFSRESL',
//                REG=32K,CLASS=A,PARM='ALL',
//                UNIT=SYSDA,SER=,DSM='IMS.MODBLKS'
//MSVERIFY EXEC  PGM=DFSUMSV0,PARM='&ALL',REGION=&REG
//STEPLIB DD      DSN=&DSM,DISP=SHR,UNIT=&UNIT,VOL=SER=&SER
//          DD      DSN=&DSN,DISP=SHR,UNIT=&UNIT,VOL=SER=&SER
//SYSOUT  DD      SYSOUT=&CLASS
```

プロシージャー・ステートメント

プロシージャー・ステートメントは、次の形式でなければなりません。

```
PROC   DSN='IMS.SDFSRESL',
        REG=32K,CLASS=A,ALL=ALL,
        UNIT=SYSDA,SER=,DSM='IMS.MODBLKS'
```

### ALL=

通知メッセージ DFS2327I を含むすべてのメッセージを印刷することを指定します。デフォルトは ALL です。

### CLASS=

SYSOUT クラスを指定します。デフォルトは A です。

### DSM=

IMS.MODBLKS のデータ・セット名を指定します。

### DSN=

検査ユーティリティ・プログラム (DFSMSV00) とその制御ブロックが入っているデータ・セット名を指定します。デフォルトは IMS.SDFSRESL です。

### REG=

このユーティリティを実行する領域のサイズを指定します。デフォルトは 32 KB です。

### SER=

DSN パラメーターで指定されたデータ・セットが入っている DASD のボリューム通し番号を指定します。データ・セットがカタログ式データ・セットの場合、SER= を指定する必要はありません。

### UNIT=

STEPLIB UNIT TYPE を指定します。

EXEC ステートメント

実行ステートメントは必ず PGM=DFSUMSV0 の形式で指定してください。  
PARM= フィールドは、以下の形式でなければなりません。

```
PARM='&ALL',REGION=&REG
```

ユーティリティーを実行するための EXEC ステートメントに PARM=ALL を指定すると、通知メッセージ DFS2327I がユーティリティー出力の一部として出力されます。この通知メッセージは、参照される論理リンクに対して //ASSIGN を使用して SYSID/MSNAME を割り当てないように警告します。これは、この割り当てではこの SYSID レベルのローカル SYSID へのパスは提供されないためです。

以下に、DFSUMSV0 ユーティリティーを実行するために必要な JCL を示します。

```
//STEP1 EXEC IMSMSV
//SYSIN DD *
        INPUT FOR IMS MULTISYSTEM VERIFICATION UTILITY
/*
```

### ユーティリティー制御ステートメント

制御ステートメントには、IMSCTRL マクロの MSVID キーワードで指定された 1 から 3 桁の接尾部が含まれていなければなりません。各制御ステートメントには、1 つ以上のこのような接尾部を任意の順序で指定できます。入力ステートメントのスキャンは、ブランクが検出されると終了します。1 桁目がブランクの場合、入力ステートメントはコメント・ステートメントとして扱われます。制御ステートメントに複数の接尾部が指定されている場合、2 番目以降の接尾部はコンマによって前の接尾部と区切らなければなりません。指定する必要があるのは接尾部の有効数字だけです。

制御ステートメントの接尾部はそれぞれ、そのステートメントで完了している必要があり、次の制御ステートメントに継続することはできません。

入力データの例を次に示します。

- 1、255、6、009、80、02、198 は有効です。
- 0、677、0040、NYC、1A、0001、5,5 は無効です。

無効な項目はそれぞれ印刷され、エラーのタイプが識別されます。

以下に例を示します。

**0** 1 から 676 の範囲にありません。

**677** 1 から 676 の範囲にありません。

**0040** 3 桁を超えています。

**NYC** 数字ではありません。

**5,5** 入力データが重複しています。

3 つのシステムに有効な入力、次のとおりです。

```
STATEMENT 1,5,255
or
STATEMENT 001,005,255
or
STATEMENT 255,01,5
```


## 戻りコード

コード 意味


0 通知メッセージと警告メッセージだけが印刷される

12 マルチシステムの実行前に解決しなければならないエラーが検出された

関連資料:

 [IMSCTRL マクロ \(システム定義\)](#)

関連情報:

 [DFS2149 \(メッセージおよびコード\)](#)

---

## 第 29 章 オンライン変更コピー・ユーティリティ (DFSUOCU0)

オンライン変更コピー・ユーティリティ (DFSUOCU0) は、IMS または IMSplex をローカル・オンライン変更またはグローバル・オンライン変更のために準備するプロセスの 1 つのステップとして使用します。

オンライン変更コピー・ユーティリティは、新規定義が入っているソース・ライブラリーをターゲット・ライブラリーにコピーします。オンライン変更コマンド・シーケンスを発行して、オンライン変更の準備をしてコミットすると、非アクティブ・ライブラリーがアクティブ・ライブラリーになります。このユーティリティは、z/OS シリアライゼーション・サービスを使用して、コピーが行われている間に他のユーティリティがステージング・ライブラリー (コピー元) または非アクティブ・ライブラリー (コピー先) を更新できないようにします。オンライン変更コピー・ユーティリティは、ターゲット・ライブラリーをクリアし、ソース・ライブラリーの内容を移すための IEBCOPY を呼び出します。

オンライン変更コピー・ユーティリティは、最初のコールド・スタートに先立って、IMS のインストール時に、ステージング・ライブラリーの内容をアクティブ・ライブラリーにコピーすることができます。これを行うためには、このユーティリティを呼び出す際に出力 DD 名のパラメーターを指定する必要があります。それは、IMS.MODSTAT (グローバル・オンライン変更が使用可能な場合は、OLCSTAT データ・セット) の初期の内容がアクティブ・ライブラリーを指定するからです。

オンライン変更コピー・ユーティリティ (DFSUOCU0) は、フォールバックの目的に使用できるアクティブ ACB ライブラリーのバックアップ・コピーを作成するために使用できます。このバックアップ・コピーを作成するには、以下のすべてのエレメントを指定します。

- コピー・タイプ・パラメーターの TYPE=ACTVACB。これは、オンライン変更コピー・ユーティリティでアクティブ ACB ライブラリーからメンバーをコピーすることを指定します。

**重要:** TYPE=ACTVACB を指定してオンライン変更コピー・ユーティリティを実行する前に、OLCSTAT ライブラリーを開始しておく必要があります。

- ターゲット・ライブラリー・パラメーターの OUT=O。これは、オンライン変更コピー・ユーティリティでアクティブ ACB ライブラリーの内容を、非アクティブ ACB ライブラリー以外のライブラリーにコピーすることを指定します。
- IMSACBO DD ステートメント。これは、バックアップ・コピーを作成する場所を指定します。

IMSplex の中に IMS サブシステムのクローンが存在せず、しかもライブラリーが共用されていない場合は、IMSplex の中のすべての IMS にオンライン変更コピー・ユーティリティを実行しなければならないことがあります。IMS サブシステムのクローン作成が行われ、ライブラリーが共用されている IMSplex では、オンライン変更コピー・ユーティリティを実行する必要がある可能性があるのは、最高位の IMS レベルの 1 つの IMS.SDFSRESL に対して 1 回だけです。

サブセクション:

- 『制約事項』
- 『前提条件』
- 667 ページの『要件』
- 667 ページの『推奨事項』
- 667 ページの『JCL 仕様』

## 制約事項

オンライン変更コピー・ユーティリティには下記の制約があります。

- ACBLIB、FORMAT、および MODBLKS ライブラリーのいずれかが複数の IMS システムによって共有されている場合、このユーティリティの実行中は、すべてのシステムが同じライブラリーを使用しなければなりません。
- XRF 環境では、ACBLIB、FORMAT、および MODBLK の各データ・セットは、エラー保護のため共有非二重 DASD 上になければなりません。同じ追加または変更を、IMS データ・セットの別個の重複コピーに行ってください。
- オンライン変更コピー・ユーティリティでは、新しい IMS モジュールを IMS.SDFSRESL データ・セットに追加することが必要となるような追加または変更を行うことはできません。
- このユーティリティを使用して、MSDB の追加、変更、または削除を行うことはできません。ただし、MSDB についての PSB 関連の変更は、DBD の変更が含まれていない場合に限り、ACBLIB に対しては行うことができます。
- このユーティリティを使用して、HALDB データベースの追加、変更、または削除を行うことはできません。これらを行うことができるのは、区画定義ユーティリティを使用した場合のみです。ただし、HALDB データベースについての PSB 関連の変更は、DBD の変更が含まれていない場合に限り、ACBLIB に対して行うことができます。
- 動的リソース定義 (DRD) を使用可能に設定 (MODBLKS=DYN を DFSCGxxx IMS.PROCLIB メンバーの中か、DFSDFxxx IMS.PROCLIB メンバーの COMMON\_SERVICE\_LAYER セクションの中で指定) してあるときは、オンライン変更コピー・ユーティリティが不要になる場合があります。オンライン変更機能は使用不可であるため、保管してあるリソース定義をコピーして、オンライン変更の準備をする必要がありません。オンライン変更コピー・ユーティリティを使用して、保管してあるリソース定義を、IMS コールド・スタートによってロードされる MODBLKS データ・セットにコピーできます。

オンライン変更コピー・ユーティリティを完了前に取り消してしまうと、ACBLIB、FMTLIB、または MODBLKS データ・セットの状況は予測できません。このユーティリティによって変更されるデータ・セットは、このユーティリティがデータ・セットの排他制御権を持つとただちにクリアされ、その後新しい情報がそのデータ・セットに書き込まれます。このユーティリティを正常終了の前に取り消した場合、データ・セット内の情報は使用できません。

## 前提条件

現在、DFSUOCU0 ユーティリティについて文書化されている前提条件はありません。

## 要件

オンラインで変更を行うためには、下記のライブラリーのコピーが 3 つ必要です。

### ACBLIB

DMB や PSB などの、データベースおよびプログラムの記述子

### FORMAT

MFS 言語ユーティリティーとサービス・ユーティリティーによって作成された制御ブロック

### MODBLKS

変更するリソース用の制御ブロックのサブセット

各ライブラリーの 1 つのコピーは、オフライン機能専用で使用されます。このライブラリーには接尾部がなく、ステージング・ライブラリーと呼ばれています。

各ライブラリーの他の 2 つのコピーには、接尾部 A または B が付いています。IMS オンライン・システムが一度に使用するのは、これらのライブラリーのうちの 1 つだけです。使用中のライブラリーを、アクティブ・ライブラリーと呼びます。使用中でないもう一方のライブラリーを、非アクティブ・ライブラリーと呼びます。

オンライン変更コピー・ユーティリティーでは、ステージング・ライブラリーの内容を MODSTAT データ・セット (ローカル・オンライン変更) または OLCSTAT データ・セット (グローバル・オンライン変更) 中の情報に基づいて、非アクティブ・ライブラリーにコピーできます。

この同じシリアライゼーションのメソッドは、アクティブ・ライブラリーが IMS オンライン・システムによって使用されている間、このユーティリティーがアクティブ・ライブラリーを更新するのを防ぎます。

## 推奨事項

ACB ライブラリー・メンバーのオンライン変更操作を実行する前に、すべてのアクティブ ACB ライブラリー・メンバーのバックアップ・コピーを作成してください。

## JCL 仕様

プロシージャ・ステートメント

このユーティリティーを呼び出し、オプションの IEBCOPY パラメーターを組み込むには、プロシージャ・ステートメントを次の書式にする必要があります。

```
OLCUTL PROC TYPE=,IN=,OUT=,SOUT=A,SYS=,SYS2=,OLCGLBL='DUMMY,',OLCLOCL=
```

**OLCGLBL=**

**OLCLOCL=**

これらのパラメーターは、OLCSTAT DD カードまたは MODSTAT/  
MODSTAT2 DD カードを生成します。

**OLCGLBL='DUMMY,',OLCLOCL=**

システム定義のステージ 2 によって生成されます。このパラメーターを指定す

ると、OLCUTL プロシージャは、ローカル・オンライン変更用に以下の DD ステートメントを備えたデフォルトとしてセットアップされます。

```
//MODSTAT DD &OLCLOCL.DSN=IMS.&SYS.MODSTAT,DISP=SHR
//MODSTAT2 DD &OLCLOCL.DSN=IMS.&SYS.MODSTAT2,DISP=SHR
//OLCSTAT DD &OLCGLBL.DSN=IMSPLEX.OLCSTAT,DISP=OLD
```

グローバル・オンライン変更の場合は、OLCGLBL= と OLCLOCL= のパラメーターを以下のように設定してください。

```
OLCGLBL=,OLCLOCL='DUMMY,'
```

これらのパラメーターは、グローバル・オンライン変更で使用される以下のような DD ステートメントを生成します。

```
//MODSTAT DD DUMMY,DSN=IMS.&SYS..MODSTAT,DISP=SHR
//MODSTAT2 DD DUMMY,DSN=IMS.&SYS..MODSTAT2,DISP=SHR
//OLCSTAT DD DSN=IMSPLEX.OLCSTAT,DISP=OLD
```

#### **SOUT=**

SYSOUT DD ステートメントに割り当てられるクラスを指定します。

#### **SYS=**

XRF 複合システムの「必須共用」に指定されたデータ・セットに対して、2 次レベルの dsname 修飾子 (オプション) を指定します。これを指定する場合は、パラメーターを引用符で囲むとともに、パラメーターに後書きピリオドを含める必要があります (例、SYS='IMSA.')

#### **SYS2=**

XRF 複合システムで「オプションの複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS2='IMSA.' のように、パラメーターを後書きピリオドも含めて引用符で囲みます。

#### **EXEC** ステートメント

EXEC ステートメントは、どのコピーを作成するか、およびどのデータ・セットを入力および出力用として使用するかを決めます。このステートメントのフォーマットには、ターゲット・ライブラリーの後に、オプションの IEBCOPY パラメーターを WORK、SIZE、LIST の順に指定して組み込むことができます。

パラメーターを省略する場合は、その場所にコンマを入れます。例えば、WORK と SIZE を指定せずに LIST パラメーターだけを指定するには、次の構文を使用します。.,&LIST

IEBCOPY オプションのリストは、IEBCOPY キーワード、等号、パラメーター値、およびコンマを含み、64 バイトを超えることができません。IEBCOPY パラメーターは、IEBCOPY ユーティリティーが呼び出された時点で IEBCOPY ユーティリティーに渡されます。

EXEC ステートメントは次の形式になっていなければなりません。

```
PGM=DFSUOCU0,PARM=(&TYPE,&IN,&OUT,&WORK,&SIZE,&LIST)
```

#### **&TYPE**

入力ライブラリーからターゲット・ライブラリーにコピーするライブラリーのタ



イブを指定します。コピー・タイプは、ACB、ACTVACB、FORMAT、MODBLKS ライブラリーのいずれかにすることができます。ACTVACB は固有であり、他の 3 つとは異なります。

パラメーター  
意味

#### ACTVACB

フォールバックの目的で、アクティブ ACB ライブラリー・メンバーをコピーしてバックアップを作成することを指定します。&TYPE パラメーター ACTVACB を指定した場合は、入力ライブラリー・パラメーター (&IN) は必要ないかサポートされません。&IN パラメーターを &TYPE パラメーター ACTVACB と一緒にコーディングした場合、IMS は情報メッセージ (DFS3469I) を発行します。

#### &IN

入力として使用するライブラリー DD 名を定義します。

パラメーター  
意味

**S** IMS ステージング・ライブラリー (IMSACB、FORMAT、または MODBLKS)

**I** ユーザー入力ライブラリー (IMSACBI、FORMATI、または MODBLKSI)

I パラメーターを使用すると、ステージング・ライブラリー以外の入力ライブラリーを使用できます。

#### &OUT

出力に使用するライブラリー DD 名を定義します。

パラメーター  
意味

**A** IMS A ライブラリー (IMSACBA、FORMATA、または MODBLKSA)

**B** IMS B ライブラリー (IMSACBB、FORMATB、または MODBLKSB)

**G** OLCSTAT データ・セットを使用して、ユーティリティーが決めるターゲット・ライブラリー (非アクティブ)。ターゲットは、IMS オンライン・システムが現在使用していないライブラリーです。OLCSTAT データ・セットの割り振りは、OLCSTAT DD ステートメントによって、または SYSIN DD ステートメントの桁 1 から OLCSTAT データ・セット名を動的に指定することによって行うことができます。

G を ACTVACB パラメーターと一緒に指定した場合、オンライン変更コピー・ユーティリティーはアクティブ ACB ライブラリーの内容を非アクティブ ACB ライブラリーにコピーします。

この場合、OLCUTL JCL には、アクティブ ACB ライブラリーのバックアップ・コピーのデータ・セット名を指定する DD ステートメントが含まれている必要があります。

**O** ユーザー出力ライブラリー (IMSACBO、FORMATO、または MODBLKSO)

O を ACTVACB パラメーターと一緒に指定した場合、オンライン変更コピー・ユーティリティはアクティブ ACB ライブラリーの内容を非アクティブ ACB ライブラリー以外のライブラリーにコピーします。

この場合、OLCUTL プロシージャ JCL には、IMSACBO のデータ定義名 (DD 名) を持つ DD ステートメント (アクティブ ACB ライブラリーのバックアップ・コピーのデータ・セット名を指定します) が含まれている必要があります。

O パラメーターを使用すると、アクティブまたは非アクティブ・データ・セット以外のターゲット・データ・セットを選択できます。

**U** MODSTAT データ・セットを使用して、ユーティリティが決めるターゲット・ライブラリー (非アクティブ)。ターゲットは、IMS オンライン・システムが現在使用していないライブラリーです。

**推奨事項:** 不適切な選択を行うと、IMS がアクティブ・ライブラリーをオーバーレイするおそれがあるので、オンライン操作時には出力ライブラリーに A または B パラメーターを使用しないようにしてください。

ローカル・オンライン変更をサポートする IMS には、U パラメーターを指定してください。IMSplex には、グローバル・オンライン変更をサポートする G パラメーターを使用するようお勧めします。

#### **&WORK**

作業パラメーターを IEBCOPY ユーティリティへ渡すオプション・パラメーター。作業パラメーターは、ディレクトリー項目、内部テーブル、および入出力バッファー用の作業域を要求するための仮想ストレージ・バイト数を渡します。

#### **&SIZE**

サイズ・パラメーターを IEBCOPY ユーティリティへ渡すオプション・パラメーター。サイズ・パラメーターは、IEBCOPY ユーティリティがバッファーとして使用できる仮想ストレージの最大バイト数を指定します。

#### **&LIST**

リスト・パラメーターを IEBCOPY ユーティリティへ渡すオプション・パラメーター。LIST=NO を指定すると、正常にコピーされた各メンバーごとに発行される IEBCOPY IEB1541 メッセージが抑止されます。

#### **DD ステートメント**

##### **IMSACB DD**

##### **IMSACBA DD**

##### **IMSACBB DD**

ステージング、アクティブ、または非アクティブ ACBLIB を定義します。

##### **IMSACBI DD**

ユーザー定義の入力 ACBLIB。

##### **IMSACBO DD**

ユーザー定義の出力 ACBLIB。

##### **FORMAT DD**

##### **FORMATA DD**

**FORMATB DD**

ステージング、アクティブ、または非アクティブの MFS 形式ライブラリーを定義します。

**FORMATI DD**

ユーザー定義の入力 FORMAT ライブラリー。

**FORMATO DD**

ユーザー定義の出力 FORMAT ライブラリー。

**OLCSTAT DD**

IMS でグローバル・オンライン変更を行えるように、グローバル・オンライン変更状況データ・セット名を定義します。OLCSTAT データ・セットは、ローカル・オンライン変更で使用する MODSTAT データ・セットに似ています。ローカル・オンライン変更を使用可能にする場合、またはグローバル・オンライン変更を使用可能にし、OLCSTAT データ・セットを動的に割り振る場合は、OLCSTAT DD を定義しません。

OLCSTAT データ・セットを動的に割り振るには、OLCSTAT DD ステートメントによって OLCSTAT データ・セットを定義するのではなく、SYSIN DD ステートメントを &OUT オプション G に指定できます。OLCSTAT データ・セットは、SYSIN DD ステートメントの後のデータとして桁 1 から定義する必要があります。

OLCSTAT DD ステートメントと SYSIN DD ステートメントによって OLCSTAT データ・セットを定義すると、SYSIN DD ステートメントは無視されます。

複数ステップのバッチ・ジョブにより、INITIATE OLC コマンドを発行する CSLUSPOC ステップの後のステップで DFSUOCU0 ユーティリティを呼び出す場合、DFSUOCU0 ユーティリティは OLCSTAT データ・セットを動的に割り振る必要があります。

**MODBLKS DD****MODBLKSA DD****MODBLKSB DD**

ステージング、アクティブ、または非アクティブ・システム定義ライブラリーを定義します。

**MODBLKSI DD**

ユーザー定義の入力 MODBLKS ライブラリー。

**MODBLKSO DD**

ユーザー定義の出力 MODBLKS ライブラリー。

**MODSTAT DD****MODSTAT2 DD**

ローカル・オンライン変更の変更状況データ・セット名を定義します。これは、オンライン IMS が初期設定時に使用するアクティブ・データ・セット (および XRF を使用する場合は、非アクティブ・データ・セット) です。

**SYSUDUMP DD**

このプログラム用のダンプ・データ・セットを定義します。このデータ・セットは、プリンター、磁気テープ、または直接アクセス装置に置くか、出力ストリームで経路指定することができます。

### **SYSPRINT DD**

メッセージ出力データ・セットを定義します。このデータ・セットは、プリンター、磁気テープ、または直接アクセス装置に置くか、出力ストリームで経路指定することができます。この DD ステートメントは必ず含める必要があります。

### **SYSUT3 DD**

必要な作業データ・セットを定義します。

### **SYSUT4 DD**

機能は SYSUT3 と同じです。

### **COPYCTL DD**

IEBCOPY の呼び出し前に作成するコピー制御データ・セットを定義します。

関連資料:



z/OS: IEBCOPY (ライブラリー・コピー) プログラム

627 ページの『第 26 章 グローバル・オンライン変更ユーティリティ (DFSUOLC0)』

---

## **OLCUTL プロシージャ**

OLCUTL プロシージャは、ライブラリー・タイプ、入力ライブラリー、出力ライブラリー用に選択したパラメーターと IEBCOPY パラメーターを指定して、DFSUOCU0 ユーティリティを呼び出すために使用します。

OLCUTL プロシージャはシステム定義によって生成され、システム定義のステージ 2 で IMS.PROCLIB データ・セットの中に配置されます。

### **OLCUTL プロシージャの呼び出し**

以下に、オンライン変更コピー・ユーティリティ OLCUTL プロシージャを呼び出すために使用する OLCUTL ステートメントを示します。このプロシージャはシステム定義によって生成され、システム定義のステージ 2 で IMS.PROCLIB に入れます。

推奨事項: OLCUTL は、ターゲット・ライブラリー・データ・セットをクリアし、ソース・ライブラリーの内容を移すための IEBCOPY を呼び出します。スペース不足のために IEBCOPY が異常終了した場合、ターゲット・ライブラリーの内容は予測できません。これを避けるために、ソース・ライブラリーとターゲット・ライブラリーには同量のスペースを割り振ってください。

```
//          PROC TYPE=,IN=,OUT=,SOUT=A,SYS=,SYS2=,OLCGLBL='DUMMY,',OLCLOCL=
//S          EXEC PGM=DFSUOCU0,PARM=(&TYPE,&IN,&OUT)
//STEPLIB   DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//MODBLKS   DD DSN=IMS.&SYS2.MODBLKS,DISP=SHR
//MODBLKSA  DD DSN=IMS.&SYS2.MODBLKSA,DISP=SHR
//MODBLKSB  DD DSN=IMS.&SYS2.MODBLKSB,DISP=SHR
//IMSACB    DD DSN=IMS.&SYS2.ACBLIB,DISP=SHR
//IMSACBA   DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
//IMSACBB   DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR
//FORMAT    DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
//FORMATA   DD DSN=IMS.&SYS2.FORMATA,DISP=SHR
//FORMATB   DD DSN=IMS.&SYS2.FORMATB,DISP=SHR
//MODSTAT   DD &OLCLOCL.DSN=IMS.&SYS.MODSTAT,
//          DISP=SHR
//MODSTAT2  DD &OLCLOCL.DSN=IMS.&SYS.MODSTAT2,
```

```
//          DISP=SHR
//OLCSTAT DD &OLCGLBL.DSN=IMSPLEX.OLCSTAT,
//          DISP=OLD
//SYSUDUMP DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//COPYCTL DD DSN=&&COPYCTL,DISP=(NEW,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
```

## ステージング・ライブラリーのコピー

以下に、ステージング・ライブラリーを MODSTAT データ・セットによって示された非アクティブ・ライブラリーにコピーする JCL ステートメントを示します。

```
//* COPY MODBLKS TO MODBLKSA
//*
//STEP01 EXEC PROC=OLCUTL,SOUT='*',TYPE=MODBLKS,IN=S,OUT=U,,LIST=NO
//*
//* COPY ACBLIB TO ACBLIBA
//*
//STEP03 EXEC PROC=OLCUTL,SOUT='*',TYPE=ACB,IN=S,OUT=U
//*
//* COPY FORMAT TO FORMATA
//*
//STEP04 EXEC PROC=OLCUTL,SOUT='*',TYPE=FORMAT,IN=S,OUT=U
//*
//* COPY FORMAT TO FORMAT WITH IEBCOPY PARAMETERS SPECIFIED
//*
//STEP05 EXEC PROC=OLCUTL,SOUT='*',TYPE=FORMAT,IN=S,OUT=U,WORK=2M,SIZE=2M,LIST=NO
//*
//*
```

## メンバー OLC 用のアクティブ ACB ライブラリーのバックアップ

以下に、アクティブ ACB ライブラリーを非アクティブ ACB ライブラリーにコピーする方法を示します (OUT=G)。

```
//DFSUOCU0 JOB LINK,MSGLEVEL=1,REGION=640K,CLASS=N,
            USER=USRT001
//STEP1 EXEC OLCUTL,TYPE=ACTVACB,OUT=G,SOUT=*,
//          OLCLOCL='DUMMY,',OLCGLBL=,SYS=
//
```

以下に、OLCUTL プロシージャの IMSACBO DD ステートメントで指定した ACB ライブラリーにアクティブ ACB ライブラリーをコピーする方法を示します (OUT=O)。

```
//DFSUOCU0 JOB LINK,MSGLEVEL=1,REGION=640K,CLASS=N,
            USER=USRT001
//STEP1 EXEC OLCUTL,TYPE=ACTVACB,OUT=O,SOUT=*,
//          OLCLOCL='DUMMY,',OLCGLBL=,SYS=
//
//
```

---

## IMS>.MODSTAT データ・セットの初期設定

INITMOD プロシージャは、ローカル・オンライン変更が有効になっている IMS の場合に、IMS.MODSTAT データ・セットを初期設定します。MODSTAT データ・セットは、最初の IMS コールド・スタートの前、または、IMS.MODSTAT が現行 DD 名を含んでいない場合はその他のコールド・スタートの前に、初期設定する必要があります。

グローバル・オンライン変更が使用可能になっている IMS は、MODSTAT データ・セットを使用しません。

推奨事項: グローバル・オンライン変更を使用可能にする場合は、MODSTAT DD カードを定義しないで、MODSTAT データ・セットを定義する必要がなくなるようにしてください。

システム定義のステージ 2 で、INITMOD プロシージャは IMS.PROCLIB プロシージャ・ライブラリーに入れられます。以下に、INITMOD プロシージャに必要な JCL を示します。

```
//INIT1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT2 DD DSN=IMS.&SYS.MODSTAT&SF,DISP=OLD
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,
// DSN=IMS.&SYS2.PROCLIB(DFSMREC)
```

## プロシージャ・ステートメント

プロシージャ・ステートメントは、次の形式でなければなりません。

```
PROC SYS=,SYS2=,SF=,SOUT=A
```

### **SYS=**

XRF 複合システムの「必須共用」に指定されたデータ・セットに対して、2 次レベルの dsname 修飾子 (オプション) を指定します。これを指定する場合は、パラメーターを引用符で囲むとともに、パラメーターに後書きピリオドを含める必要があります (例、SYS='IMSA.')

### **SYS2=**

XRF 複合システムで「オプションの複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS2='IMSA.' のように、パラメーターを後書きピリオドも含めて引用符で囲みます。

### **SF=**

MODSTAT データ・セット名の接尾部を指定します (SF= または SF=2)。

### **SOUT=**

SYSOUT DD ステートメントに割り当てられるクラスを指定します。

## DFSMREC 制御ステートメント

INITMOD プロシージャは、この制御ステートメントを使用して、MODSTAT データ・セットを初期設定します。DFSMREC は、MODSTAT レコードのデータを含みます。この制御ステートメントは、SMP/E 処理時に作成される DFSPROCB JCL を使用すると、作成されます。このステートメントは、次のような形式でなければなりません。

```
0,MODBLKSA,IMSACBA,FORMATA
```

値は、コンマで区切る必要があり、組み込みブランクは入れません。

0 MODSTAT の ID で、制限なしの可変長です。この正の値は、ゼロに初期設定

され、緊急時再始動処理時にセキュリティー状況のリカバリーのために IMS 内部処理で使用されます。任意の IMS コールド・スタート時に、これをゼロに初期設定できます。

#### **MODBLKSA | MODBLKSB**

アクティブ MODBLKS データ・セット (IMS システム定義出力が入っている IMS.MODBLKSA または IMS.MODBLKSB データ・セット) の DD 名です。

DRD が使用可能な場合、MODBLKS DD カードを定義せずに IMS を開始できます。INITMOD プロシーチャーを使用するには、MODBLKS データ・セットを定義していない IMS の場合でも、MODBLKSA または MODBLKSB を定義して MODSTAT データ・セットを初期設定する必要があります。

MODBLKS データ・セットが IMS に対して定義されていない場合、IMS は MODSTAT データ・セット内の MODBLKSA または MODBLKSB 定義を無視します。

INITMOD プロシーチャーを使用するには、MODBLKS データ・セットを定義していない IMS の場合でも、MODBLKSA または MODBLKSB を定義して MODSTAT データ・セットを初期設定する必要があります。

#### **IMSACBA | IMSACBB**

アクティブ IMSACB データ・セット (IMS.ACBLIBA または IMS.ACBLIBB ライブラリー) の IMS プロシーチャー内の DD 名です。

#### **FORMATA | FORMATB**

アクティブ FORMAT データ・セット (オンライン・システムによってフォーマット・ライブラリーとして使用されるオンライン MFS 定義が入っている IMS.FORMATA または IMS.FORMATB データ・セット) の DD 名です。MFS がサポートする端末および MFS 言語ユーティリティー・プログラムでは、それらを使用する必要があります。これらのライブラリーの 1 つがアクティブな場合 (つまり、オンライン・システムによって使用されている場合) は、次のオンライン変更の実行時に使用できるように、IMS.FORMAT の内容が別のライブラリー (つまり、非アクティブ・ライブラリー) にコピーされます。それらの DD 名は、それぞれ FORMATA および FORMATB にする必要があります。MFS を定義しなかった場合、IMS は、この DD 名を無視します。

DBCTL の場合、FORMAT DD カードを定義せずに、DBCTL IMS を開始できます。INITMOD プロシーチャーを使用するには、MODSTAT データ・セットを使用していない DBCTL IMS の場合でも、FORMATA または FORMATB を定義して MODSTAT データ・セットを初期設定する必要があります。

DBCTL IMS は、MODSTAT データ・セット内の FORMATA または FORMATB の定義を無視します。

INITMOD プロシーチャーを使用するには、MODSTAT データ・セットを定義していない IMS の場合でも、FORMATA または FORMATB を定義して MODSTAT データ・セットを初期設定する必要があります。

IMS.MODSTAT レコードの内容が失われ、再構成する必要がある場合、あるいは INITMOD プロシーチャーによるデフォルト初期設定を使用しない場合には、IEBGENER ジョブを実行して、DD 名およびオンライン変更 ID の正しい値を指定してその内容を構成する必要があります。新しい IMS.MODSTAT データ・セットの属性は、RECFM=F および BLKSIZE=80 でなければなりません。

以下に、MODSTAT ID を 0 に、DD 名を MODBLKSA、IMSACBA、および FORMATA に初期設定する方法を示します。

```
//INIT1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT2 DD DSN=IMS.&SYS.MODSTAT&SF,DISP=OLD
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,
//          DSN=IMS.&SYS2.PROCLIB(DFSMREC)
./          ADD NAME=DFSMREC
./          NUMBER NEW1=10,INCR=10
0,MODBLKSA,IMSACBA,FORMATA
```

あるいは、INITMOD プロシージャの SYSUT1 および SYSUT2 DD ステートメントを変更して、前述の IEBGENER サンプル・ジョブと同じ目的のために使用することもできます。

MODSTAT レコードの現行値を示す DFS3499 メッセージは、DFS994 チェックポイント・メッセージの後に続きます。初期設定時の DFS3410 メッセージも、MODSTAT レコード・データを示します。



---

## 第 30 章 スプール SYSOUT 印刷ユーティリティ (DFSUPRT0)

システム定義時にスプール SYSOUT 用の通信回線が定義されている場合、オンライン制御プログラムによって生成されたメッセージを一連のデータ・セットからシステム出力装置にコピーするには、スプール SYSOUT 印刷ユーティリティ (DFSUPRT0) を使用します。

スプール・データ・セットもシステム出力装置も QSAM を使用して処理されます。スプール・データ・セットのブロック化因数は、オンライン制御プログラムによって決定されます。システム出力装置のブロック化は、JCL の SYSPRINT DD ステートメントで指定できます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 『入力と出力』
- 678 ページの『JCL 仕様』
- 680 ページの『戻りコード』

### 制約事項

スプール SYSOUT 印刷ユーティリティは CICS をサポートしていません。

### 前提条件

現在、DFSUPRT0 ユーティリティについて文書化されている前提条件はありません。

### 要件

現在、DFSUPRT0 ユーティリティについて文書化されている要件はありません。

### 推奨事項

開始した印刷ユーティリティは、/START LINE コマンドを発行してスプール SYSOUT を使用可能にする前に完了しておくようにします。

### 入力と出力

印刷ユーティリティからの出力には、状況情報のページと、それに続いて FULL として示され、発生順に印刷されたスプール・データ・セットの内容が含まれます。

### 出力例

以下の図は、スプール SYSOUT 印刷ユーティリティーの出力の例を示しています。

```
DFSUPRT0 - SYSOUT PRINT UTILITY    TIME  9:35:1   DATE   10.056

DDNAME    STATUS    CREATED TIME    DATE    DATASET NAME
SPOOL1    FULL      9:33:239      10.244  IMSTESTL.IMS01.SPOOL1
SPOOL2    INUS      :00:000       0.000   IMSTESTL.IMS01.SPOOL2
SPOOL3    AVAL      :00:000       0.000   IMSTESTL.IMS01.SPOOL3
```

報告書のフィールドの意味は、次のとおりです。

#### DDNAME

ユーザー提供の DDNAME

#### STATUS

FULL - データ・セットを印刷する場合

INUS - オンラインで書き込み中の場合

AVAL - 使用されない場合

#### CREATED TIME

作成時刻 (24 時間表示) (HH:MM:SST)

日付 作成の年間通算日 (YY.DDD)

#### DATASET NAME

割り当てられたデータ・セットの DSNAME

スプール・データ・セットに入っているシステム・メッセージには、通常、印刷不能な制御文字 (通常、改行文字記号 X'15') が含まれています。UCS プリンターを SYSOUT 装置として使用する場合、これらのメッセージは関係のない英字として印刷されます (UCS パラメーター要求に対して、大文字変換モード操作が指定されている場合)。

#### JCL 仕様

DFSUPRT0 ユーティリティーには、プロシージャー・ステートメント、EXEC ステートメント、および DD ステートメントが必要です。

以下の図に示した DFSWT $nnn$  プロシージャーは、システム定義の過程でスプール SYSOUT オプションによって作成されたデータ・セットを印刷するオンライン・プログラムとして、スプール SYSOUT 印刷ユーティリティー・プログラム (DFSUPRT0) を実行します。このユーティリティー・プログラムは、オンライン制御プログラムにより生成されたメッセージを、一連のデータ・セットからシステム出力装置にコピーします。これらのプロシージャーはシステム定義時に作成され、SYSDEF のステージ 2 で IMS.PROCLIB プロシージャー・ライブラリーに入れます。 $nnn$  は、UNITYPE=SPOOL の回線グループの数によって決まります。

```
//      PROC  SOUT=A,RGN=30K,SYS1=,SYS2=
//PRINT EXEC PGM=DFSUPRT0,REGION=&RGN
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//*
//SYSPRINT DD SYSOUT=&SOUT,DCB=BLKSIZE=1410
//SYSUDUMP DD SYSOUT=&SOUT
/*
```

```
//SPOOLn DD DISP=SHR,DSN=IMS.&SYS1.SYS01
//SPOOLn DD DISP=SHR,DSN=IMS.&SYS1.SYS02
//SPOOLn DD DISP=SHR,DSN=IMS.&SYS1.SYS03
//*
```

### プロシージャー・ステートメント

このステートメントは次の形式になっていなければなりません。

```
PROC SOUT=A,RGN=30K,SYS1=,SYS2=
```

#### **SOUT=**

SYSOUT DD ステートメントに割り当てられるクラスを指定します。

#### **RGN=**

IMS 制御プログラムに割り振られる z/OS 領域のサイズを指定します。

#### **SYS1=**

XRF 複合システムで「必須の複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS1='IMSA.' のように、オペランドを後書きピリオドも含めて引用符で囲みます。

#### **SYS2=**

XRF 複合システムで「オプションの複製」として指定されているデータ・セットに、オプションの第 2 レベルの DS 名修飾子を指定します。指定する場合には、SYS2='IMSA.' のように、オペランドを後書きピリオドも含めて引用符で囲みます。

### EXEC ステートメント

このステートメントは、以下の形式にすることができます。

```
EXEC PGM=DFSUPRT0
```

あるいは、必要な JCL ステートメントが含まれているプロシージャーをこのステートメントに指定することもできます。実行に適切な領域サイズは通常 30 KB です。

### DD ステートメント

#### **STEPLIB DD**

印刷ユーティリティーが入っているライブラリーを定義します。この DD ステートメントは通常 DSNAME=IMS.SDFSRESL,DISP=SHR です。

#### **SYSPRINT DD**

出力先のシステム出力装置を定義します。レコード形式は VBM です。ブロック・サイズの指定がない場合、または 141 未満のブロック・サイズが指定された場合には、デフォルトのブロック・サイズ 141 が想定されます。ブロック・サイズには、QSAM に対して有効な値、または 142 以上の値を指定できます。論理レコードの長さには任意の値を指定できます。論理レコード長の指定がない場合のデフォルトは、指定されたブロック・サイズから 4 を引いた値になります (ブロック・サイズのデフォルト 141 を使用した場合、論理レコード長は 137 になります)。

DFSUPRT0 は、拡張アドレス・ボリューム (EAV) 上のシリンダー管理域内の出力データ・セットをサポートします。これらのデータ・セットは、サイズが

65535 トラックを超え、相対トラック (TTT) エクステンション・サイズが 3 バイトであり、DD ステートメントで DSNTYPE = LARGE を指定して割り振られたものです。EAV ボリューム上のデータ・セットは、65535 トラックを超える大きいデータ・セットの場合も、それより小さいデータ・セットの場合もあります。

#### SPOOLnn DD

印刷されるスプール・データ・セットを記述します (nn は有効な英数字の ID です)。この DD ステートメントは通常 DSNAME=IMS.SYSnn です。ここで、「nn」はシステム定義によって割り当てられます。DCB 情報をコーディングしないでください。既にコーディングされている場合は、RECFM=VBM を指定してください。

DFSUPRT0 は、拡張アドレス・ボリューム (EAV) 上のシリンダー管理域内の入力データ・セットをサポートします。これらのデータ・セットは、サイズが 65535 トラックを超え、相対トラック (TTT) エクステンション・サイズが 3 バイトであり、DD ステートメントで DSNTYPE = LARGE を指定して割り振られたものです。

#### 戻りコード

コード 意味

- 0 正常終了
- 4 印刷用に割り振られたデータ・セットがない
- 8 SYSPRINT DD ステートメントが欠落している
- 12 SYSPRINT データ・セットでの入出力エラー

---

## DFSUPRT0 ユーティリティの例

DFSWTnnn プロシージャは、スプール SYSOUT オプションによって作成されたデータ・セットを印刷する IMSWTnnn ジョブとして呼び出すことができます。

| IMSWTnnn メンバーのジョブ・クラスおよびメッセージ・クラスは、  
| IMS.SDFSPROC データ・セットにある IMSWT000 メンバーの **CLASS=** パラメータ  
| ーおよび **MSGCLASS=** パラメーターによって決定されます。

このジョブは、プロシージャ DFSWTnnn を実行し、それにより、スプール SYSOUT ユーティリティ・プログラム (DFSUPRT0) が呼び出され、スプール SYSOUT データ・セットが印刷されます。

このジョブは、IMS.JOBS データ・セットにコピーして実行しなければなりません。

```
//SPRT0 JOB 1,IMS,CLASS=A,MSGCLASS=A,MSGLEVEL=1  
// EXEC DFSWTnnn
```

---

## 第 31 章 時間制御操作検査ユーティリティ (DFSTVER0)

時間制御操作 (TCO) 検査ユーティリティ (DFSTVER0) は、TCO スクリプト・メンバーにエラーがないようにするために使用します。スクリプト・メンバーをオンラインで実行する前にこのユーティリティを実行すると、このユーティリティはオンライン実行時に検出されるすべてのスクリプト・メンバーを検出します。

このユーティリティは、以下の事項についての報告書を生成します。

- エラー
- 統計
- タイマー・エレメント (時間スケジュール要求)
- メッセージ
- 要約

例外: ストレージの不足によって発生したエラーは検出されません。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 682 ページの『入力と出力』
- 682 ページの『JCL 仕様』
- 683 ページの『戻りコード』

### 制約事項

このユーティリティには、以下の制約事項が適用されます。

DFSTVER0 ユーティリティを実行する前に、TCO スクリプト・メンバーを TCO スクリプト・ライブラリーに追加する必要があります。

### 前提条件

現在、DFSTVER0 ユーティリティについて文書化されている前提条件はありません。

### 要件

現在、DFSTVER0 ユーティリティについて文書化されている要件はありません。

### 推奨事項

現在、DFSTVER0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

検査するメンバーごとに入力制御ステートメントを割り当てることにより、複数のメンバーを同時に検査することができます。

### JCL 仕様

TCO 検査ユーティリティーは、標準 z/OS ジョブとして実行されます。次のものがが必要です。

- ユーザーが定義する JOB ステートメント
- EXEC ステートメント
- DD ステートメント

#### EXEC ステートメント

このユーティリティーを実行する領域のサイズは、時間スケジュール要求セットおよびメッセージ・セットに対して 4 KB の増分値で獲得されます。したがって、ユーティリティーを実行するたびに 8 KB が獲得されます。各タイミング・エレメントは 4 KB のストレージの 32 バイトを使用します。メッセージが使用するストレージの容量は、指定するセグメントの数によって異なります。例として示したジョブでは、ストレージの容量を増やさずに、さらに時間スケジュール要求とメッセージを追加することができます。

EXEC ステートメントは次の形式になっていなければなりません。

```
// EXEC PGM=DFSTVER0
```

#### DD ステートメント

##### STEPLIB DD

TCO モジュールが入っている IMS.SDFSRESL を指します。

##### SYSPRINT DD

ユーティリティーが生成した報告書を格納する出力データ・セット。

##### SYSUDUMP

ダンプ・データ・セットを定義します。

##### SYSIN DD

80 文字の制御ステートメントが入っている入力制御データ・セットを記述します。ユーザーがこのユーティリティーで検査しようとする TCO スクリプト・メンバー名は、各ステートメントの 1 から 8 桁目で指定します。

メッセージ・セグメントのサイズを変更するには、CONT キーワードにパラメーターを指定します。デフォルトのセグメント連結回数は 9 です。CONT パラメーターは、特定のスクリプトの新しいセグメント連結回数を示す 1 から 99 までの 1 または 2 桁の数値です。

##### DFSTCF DD

TCO スクリプト・メンバーが入っている IMS.TCFSLIB を指します。データ・セット TCFSLIB または他の有効な DS 名を指定できます。

## 戻りコード

TCO 検査ユーティリティーは、検査処理の状況を示すコードを戻します。戻りコードは、次のとおりです。

コード 意味

- 0 検査中のスクリプトにはエラーが検出されなかった。各スクリプトごとに、5 つの出力報告書が生成されます。
- 4 検査 JCL で指定された CONT パラメーターにエラーがあった
- 6 スクリプトに構文エラーがあった
- 8 以下のいずれかのエラーが生じた
  - ストレージを獲得できない
  - SYSIN データ・セットをオープンできない
  - SYSPRINT データ・セットをオープンできない
  - 検査 JCL に DFSTCF DD ステートメントがない
- 10 入出力にエラーがある
- 12 スクリプト・メンバーが見つからない
- 14 スクリプト・メンバーのデータ・セットをオープンできない

ユーティリティーから 0 より大きい戻りコードが戻される場合は、検査中の 1 つ以上のスクリプトにエラーがあります。

DFSTVER0 ユーティリティーの報告書は、エラーのない報告書について発行されます。エラーのあるスクリプトについては、エラー報告書が出力されます。ただし、エラーのタイプによっては、エラー報告書が生成されない場合があります。

---

## DFSTVER0 ユーティリティーの例

これらの例では、DFSTVER0 ユーティリティーを使用してサンプル・スクリプト・メンバーを作成し、メンバーを検査し、報告書を作成する方法を示します。

### サンプル・スクリプト・メンバーの例

以下に、サンプル・スクリプト・メンバー (DFSTCF10) を示します。

```
/BRO LTERM CTRL
DFSTCF10 LOADED.
*TIME DFSTXITB S ***** 00001500
/ASS LTERM LOG27403 TO LINE 31 PTERM 1 ; 00001600
/START LINE 2 PTERM ALL; 00001700
/START LINE 26 PTERM ALL; 00001800
/START LINE 18 PTERM ALL; 00001900
/STA DB MSDBLM01,MSDBLM02,MSDBLM03,MSDBLM04,MSDBLM05; 00002000
/STA DB MSDBLM06,MSDBLM07,MSDBLM08; 00002100
*TIME DFSTXITB S ***** 00002200
/START REGION MSDBMTX3; 00002300
/START REGION MSDBMTY3; 00002400
/START REGION MSDBMTZ1; 00002500
*TIME DFSTXITB S ***** 00002600
PTERM01 BEGIN PTERM1; 00002700
PTERM03 BEGIN PTERM3; 00002800
/STOP REGION 1; 00002900
*TIME DFSTXITB S ***** 00003000
00003100
```

```

DFSTCF LOAD DFSTCF1A;                                00003200
*TIME   DFSTXITB      0004 S                        ****      00003300
*TIME   DFSTXTIB      0004 S                        ****      00003400
/*

```

## 検査の例

以下に、メンバー DFSTCF01、DFSTCF02、および DFSTCF10 の TCO 検査ユーティリティ・プロシージャの例を示します。

```

/*
/*      THIS JCL IS USED TO VERIFY THE USER SUPPLIED SCRIPTS
/*
// EXEC PGM=DFSTVER0
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DFSTCF DD DSN=IMS.TCFLIB,DISP=SHR
//SYSIN DD *
DFSTCF01 CONT 5
DFSTCF02 CONT 20
DFSTCF10      (ONE OR MORE CARDS SPECIFYING MEMBER NAMES TO BE VERIFIED)
/*

```

この場合、3つのスクリプト・メンバー DFSTCF01、DFSTCF02、および DFSTCF10 が、DFSTCF DD ステートメントで参照されている IMS.TCFLIB に入っています。DFSTVER0 ユーティリティ DFSTVER0 は、STEPLIB DD ステートメントで参照されている IMS.SDFSRESL に入っています。このユーティリティによって生成された報告書は、クラス A 出力用に割り当てたデバイスへ送られます。

## エラー報告書の例

以下に、TCO 検査ユーティリティによって生成されたエラー報告書の例を示します。このエラー報告書では、DFSTCF10 スクリプト・メンバーのシーケンス番号 00003400 のステートメントに指定されている出口ルーチンの名前のつづりが誤っていることが示されています。この出口ルーチンを IMS.SDFSRESL ライブラリーで検出できなかったため、この報告書でエラーとして報告されています。このステートメントは、タイマー・エレメント報告書にある時間スケジュール要求テーブルから削除されます。

```

ERROR REPORT FOR MEMBER DFSTCF10

DFS3360E USER EXIT DFSTXTIB REQUESTED NOT FOUND, SEQUENCE NUMBER= 00003400

```

## 統計報告書の例

以下に、TCO 検査ユーティリティによって生成された統計報告書の例を示します。いずれかの時間要求ステートメント (IMS.SDFSRESL にある) で指定されている出口ルーチンは DFSTXITB だけです。

```

STATISTICS REPORT FOR MEMBER DFSTCF10 PROGRAM EXITS REQUIRED IN IMS.SDFSRESL DFSTXITB

```

## タイマー・エレメント報告書の例

以下に、TCO 検査ユーティリティによって生成されたタイマー・エレメント報告書の例を示します。



## TIMER ELEMENTS REPORT

TIME OF ACTIVATION	EXIT CALLED	ATTRIBUTES	PARM
STARTUP	DFSTXITB	RES	**** MSG SET
STARTUP	DFSTXITB	RES	**** MSG SET
STARTUP	DFSTXITB	RES	**** MSG SET
STARTUP	DFSTXITB	RES	**** MSG SET
0957	DFSTXITB	RES SNGL	**** MSG SET

この報告書の各列の意味は、次のとおりです。

### Time of Activation

STARTUP、あるいはそれが実際の実行である場合は時間要求が処理される時刻のいずれかを示します。

### Exit Called

この時間要求に対して呼び出される出口を示します。

### Attributes

以下の属性が可能です。

#### RES

常駐の出口ルーチンを示します。

#### DYN

動的にロードされる出口ルーチンを示します。

#### CONT

毎日同時刻の実行を示します。

#### SNGL

単一の実行を示します。

### PARM

この時間要求のメッセージ・セットを示す \*\*\*\*MSG SET か、あるいはスクリプト・メンバーのこの時間要求の 56 から 71 桁目で指定されている 16 バイトのデータのいずれかを示します。

## メッセージ・テーブル報告書の例

以下に、TCO 検査ユーティリティーによって生成されたメッセージ・テーブル報告書の例を示します。この報告書は、DFSTCF10 スクリプト・メンバーに 5 つのメッセージ・セットがあることを示しています。1 桁目のアスタリスクは、新しいメッセージを示します。それぞれのメッセージ・セットはブランク行で区切られます。

### MESSAGE TABLE REPORT

EACH LINE IS A SEGMENT

\* IN COLUMN 1 SIGNIFIES START OF NEW MESSAGE

\* IN COLUMN 121 SIGNIFIES SEGMENT IS TRUNCATED

\*/BRO LTERM CTRL  
DFSTCF10 LOADED.

\*/ASS LTERM LOG27403 TO LINE 31 PTERM 1 ;  
\*/START LINE 2 PTERM ALL;  
\*/START LINE 26 PTERM ALL;  
\*/START LINE 18 PTERM ALL;  
\*/STA DB MSDBLM01,MSDBLM02,MSDBLM03,MSDBLM04,MSDBLM05;

```

*/STA DB MSDBLM06,MSDBLM07,MSDBLM08;

*/START REGION MSDBMTX3;
*/START REGION MSDBMTY3;
*/START REGION MSDBMTZ1;

*PTERM01 BEGIN PTERM1;
*PTERM03 BEGIN PTERM3;
*/STOP REGION 1;

*DFSTCF LOAD DFSTCF1A;

```

メッセージ・セットは、1 つ以上のメッセージから成ります。メッセージ・セットは、単一セグメントまたは複数セグメントから成ります。この報告書の例では、次のとおりです。

- 最初のメッセージ・セットは単一の複数セグメント・メッセージです。
- 2 番目、3 番目、および 4 番目のメッセージ・セットは、複数の単一セグメント・メッセージです。
- 最後のメッセージ・セットは、単一の単一セグメント・メッセージです。

### 要約報告書の例

以下に、TCO 検査ユーティリティーによって生成された要約報告書の例を示します。要約報告書には、スクリプト・メンバーで見付かった時間スケジュール要求の数とメッセージの数のリストが出力されます。また、検査されるメンバーの時間スケジュール要求に指定された出口ルーチンの数と、ストレージの使用量の要約も出力されます。

SUMMARY REPORT

# ELEMENTS	# MSGS	#EXIT ROUTINES	STORAGE SIZE
00005	00014	00001	08324

---

## 第 6 部 動的リソース定義ユーティリティー

動的リソース定義 (DRD) ユーティリティーは、リソース定義データ・セット (RDDS) の作成、1 つの RDDS から別の RDDS または IMSRSC リポジトリへの内容のコピー、RDDS を作成するためのデータの再フォーマット設定などの作業を実行する場合に使用します。



---

## 第 32 章 Repository to RDDS ユーティリティー (CSLURP20)

Repository to RDDS ユーティリティー (CSLURP20) を使用して、リソース定義データ・セット (RDDS) を IMSRSC リポジトリから生成します。このユーティリティーは、バックアップやマイグレーションを行う場合、またはリソース定義と記述子定義を 1 つのリポジトリから別のリポジトリにコピーする場合にも使用できます。

生成される RDDS は非システム RDDS です。この RDDS を使用して、リソース定義と記述子定義を IMPORT コマンドによって IMS にインポートすることができます。生成された RDDS のリソース定義と記述子定義は、DFSURDD0 などの RDDS ユーティリティーへの入力としても使用できます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 690 ページの『推奨事項』
- 690 ページの『インターフェース』
- 690 ページの『入力と出力』
- 690 ページの『JCL 仕様』
- 691 ページの『ユーティリティー制御ステートメント』
- 691 ページの『戻りコード』

### 制約事項

現在、CSLURP20 ユーティリティーについて文書化されている制約事項はありません。

### 前提条件

現在、CSLURP20 ユーティリティーについて文書化されている前提条件はありません。

### 要件

このユーティリティーは Resource Manager (RM) および Structured Call Interface (SCI) との接続を行います。このユーティリティーが実行される z/OS イメージでは、アクティブな SCI 領域が使用可能でなければなりません。アクティブな RM 領域および SCI 領域がシスプレックス内の z/OS イメージで使用可能でなければなりません。そのイメージは、必ずしもこのユーティリティーが実行される z/OS イメージと同じでなくてもかまいません。

RM のバージョンは IMS バージョン 12 以降に対応し、リポジトリで使用できるように設定されている必要があります。

## 推奨事項

現在、CSLURP20 ユーティリティーについて文書化されている推奨事項はありません。

## インターフェース

CSLURP20 は、標準 JCL によって開始されるオフラインのバッチ・ユーティリティーです。

## 入力と出力

CSLURP20 ユーティリティーへの入力は、IMS リソース定義と記述子定義を含むリポジトリ、および RDDS データ・セット名です。

CSLURP20 ユーティリティーからの出力は非システム RDDS です。この RDDS には、要求された IMS システム用のリポジトリのリソース定義と記述子定義が含まれています。

指定された RDDS データ・セットではリソース定義と記述子定義が上書きされません。

指定された RDDS はデータが書き込まれる前にフォーマットされるため、最終的にその RDDS の既存の内容は上書きされます。保管されているすべてのリソース定義がこのユーティリティーによってリポジトリから読み取られるまで、RDDS はフォーマットされません。RDDS ヘッダーには、リポジトリからのシステム・タイプ情報が含まれます。

このユーティリティーの出力には、リポジトリから書き込まれたリソース定義の数をリストする要約セクションが含まれます。

ユーティリティーによって発行されたメッセージは、SYSPRINT データ・セットに書き込まれます。出力定義の処理中のエラー、リポジトリからの読み取り中のエラー、または RDDS への書き込み中のエラーが発生した場合、このユーティリティーは戻りコード 8 を発行して終了します。

CSLURP20 ユーティリティーのエラーは、SYSPRINT データ・セット内で CSL26xxE メッセージによって報告されます。

## JCL 仕様

JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1      EXEC PGM=CSLURP20
```

### DD ステートメント

## JOBLIB / STEPLIB DD

ユーティリティの実行可能モジュールが入っている IMS.SDFSRESL を指します。連結データ・セットは無効です。

## RDDSDSN DD

このユーティリティからの出力として使用される RDDS の DSN を定義します。この DD ステートメントは必須です。RDDS として指定できる DSN は 1 つだけです。連結データ・セットは無効です。

## SYSPRINT DD

このユーティリティによって生成されたメッセージを受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FB,LRECL=133 です。

## SYSIN DD

ユーティリティの入力パラメーターのソースを定義します。この DD ステートメントは必須です。SYSIN データ・セットの DCB パラメーターは、RECFM=FB,LRECL=80 です。

例えば、次のようになります。

```
//SYSIN          DD      *  
IMSPLEX(NAME=PLEX1 IMSID(IMS1))
```

SYSIN DD ステートメントの入力パラメーターの形式は次のとおりです。

▶▶—IMSPLEX(NAME=*plxnm*—IMSID(*imsx*)—)—————▶▶

### NAME=

CSLURP20 ユーティリティが RM および SCI の登録に使用する IMSplex を指定します。この値は 1 から 5 文字の値で、文字 CSL に付加されて IMSplex 名を形成します。指定する値は、RM および SCI 領域の IMSPLEX= に指定された値と同じでなければなりません。このパラメーターは必須であり、反復不可です。

### IMSID()

CSLURP20 ユーティリティがリポジトリから読み取るリソース定義と記述子定義がある IMS ID を指定します。このパラメーターは 1 から 4 文字です。このパラメーターは必須であり、反復不可です。

## ユーティリティ制御ステートメント

現在、CSLURP20 ユーティリティについて文書化されているユーティリティ制御ステートメントはありません。

## 戻りコード

CSLURP20 ユーティリティのエラーは、SYSPRINT データ・セット内で CSL26xxE メッセージによって報告されます。完了時、このユーティリティは以下のいずれかの戻りコードを発行して終了します。

コード 意味

- 0 ユーティリティは正常に完了しました。
- 8 ユーティリティは強制終了エラーを検出しました。

関連概念:

- IMSRSC リポジトリの管理 (システム管理)
- IMSRSC リポジトリの概要 (システム定義)
- 動的リソース定義環境の保守 (システム定義)

関連タスク:

- IMSRSC リポジトリの使用からのフォールバック (システム定義)
- IMPORT コマンドを使用した、RDDS からのリソース定義および記述子定義のインポート (システム定義)
- 1 つ以上の IMS システムが停止しているが、RM システムがアクティブな場合の IMSRSC リポジトリ・データ・セットのリカバリー (オペレーションおよびオートメーション)
- IMSRSC リポジトリにおけるリソース定義および記述子定義の作成 (システム定義)

関連資料:

- CSL メッセージ (メッセージおよびコード)

---

## CSLURP20 ユーティリティの例

これらの例では、Repository to RDDS ユーティリティ (CSLURP20) を実行するためのサンプル JCL と、要約報告書の例を示します。

### JCL の例

次のサンプル JCL を使用して、CSLURP20 ユーティリティを実行できます。

```
//RPO2RDDS JOB      ,USER,CLASS=A,MSGCLASS=X,NOTIFY=USER
//JOB LIB DD        DSN=IMSV12.RESLIB
//STEP1 EXEC       PGM=CSLURP20
//SYSUDUMP DD      SYSOUT=*
//SYSPRINT DD      SYSOUT=*
//*
//*****
//* SPECIFY A VALID RDDS DSN                               */
//*****
//RDDS DSN DD      DSN=TEST.NONSYS.RDDS,DISP=OLD
//*
//*****
//* IMSID MUST BE SPECIFIED ON SYSIN                       */
//*****
//SYSIN DD *
      IMSPLEX(NAME=PLEX1 IMSID(IMS1))
/*
//
```

### 要約出力の例

次の例は、CSLURP20 ユーティリティの要約出力を示しています。

Sample Output for CSLURP20  
-----




```
CSL2603I CSLURP20 IS PROCESSING RDDS TEST.NONSYS.RDDS
CSL2618I CSLURP20 IS PROCESSING PLEX=CSLPLEX1, IMSID LIST FROM SYSIN IMS1
CSL2620I CSLURP20 SUCCESSFUL REGISTRATION WITH RM, RMNAME=RM1RM
CSL2625I CSLURP20 WRITE TO RDDS SUCCESSFUL FOR RDDSDSN=TEST.NONSYS.RDDS
        FROM REPOTYPE=IMSRSC  REPONAME=IMS_REPOS
```

\*\*\*\* SUMMARY \*\*\*\*

DB	COUNT	: 828
DBDESC	COUNT	: 0
PGM	COUNT	: 599
PGMDESC	COUNT	: 0
RTC	COUNT	: 89
RTCDESC	COUNT	: 0
TRAN	COUNT	: 664
TRANDESC	COUNT	: 0

関連概念:

 [動的リソース定義の概要 \(システム定義\)](#)



---

## 第 33 章 RDDS to Repository ユーティリティー (CSLURP10)

RDDS to Repository ユーティリティー (CSLURP10) を使用して、リソース定義データ・セット (RDDS) の内容を IMSRSC リポジトリにコピーします。

このユーティリティーを使用すると、RDDS のリソース定義と記述子定義によるリポジトリへの初回のデータ取り込みまたはリポジトリの更新を行うことができます。

CSLURP10 ユーティリティーが RDDS の内容をリポジトリにコピーするとき、以下のように、Resource Manager (RM) でリソース妥当性検査が実行されます。

- 新しいトランザクション定義または宛先コード定義がリポジトリに追加されると、リソース妥当性検査が行われ、関連するプログラムの定義がリポジトリに存在しているか、または CSLURP10 ユーティリティーによってリポジトリに追加中であるかが確認されます。
- リポジトリ内の既存のプログラム定義、トランザクション定義、または宛先コード定義が更新されると、リソース妥当性検査が行われ、更新される属性が、関連するどのリソース定義の属性とも競合しないことが確認されます。例えば、トランザクションが高速機能専用 (FP(E)) に変更されると、関連するプログラムが FP(E) として定義されていることを確認する検査が行われます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 696 ページの『要件』
- 696 ページの『推奨事項』
- 696 ページの『インターフェース』
- 696 ページの『入力と出力』
- 697 ページの『JCL 仕様』
- 698 ページの『ユーティリティー制御ステートメント』
- 698 ページの『戻りコード』

### 制約事項

現在、CSLURP10 ユーティリティーについて文書化されている制約事項はありません。

### 前提条件

現在、CSLURP10 ユーティリティーについて文書化されている前提条件はありません。

## 要件

RDDS はシステム RDDS と非システム RDDS のいずれの場合もありますが、正常なエクスポート操作、またはいずれかの RDDS 作成ユーティリティから取得した有効なリソース定義セットを含んでいる必要があります。

このユーティリティは RM および Structured Call Interface (SCI) との接続を行います。このユーティリティが実行される z/OS イメージでは、アクティブな SCI 領域が使用可能でなければなりません。アクティブな RM 領域および SCI 領域がシスプレックス内の z/OS イメージで使用可能でなければなりません。そのイメージは、必ずしもこのユーティリティが実行される z/OS イメージと同じでなくてもかまいません。

RM のバージョンは IMS バージョン 12 以降に対応し、リポジトリで使用できるように設定されている必要があります。

## 推奨事項

現在、CSLURP10 ユーティリティについて文書化されている推奨事項はありません。

## インターフェース

CSLURP10 ユーティリティは標準 JCL によって開始します。

## 入力と出力

CSLURP10 ユーティリティへの入力はシステムまたは非システム RDDS です。この RDDS には IMS リソース定義と記述子定義が含まれています。RDDS は、正常なエクスポート操作から取得した有効なデータを含んでいるか、いずれかの RDDS 作成ユーティリティで生成されたものでなければなりません。

EXPORTNEEDED を指定した Create RDDS from Log Records ユーティリティ (DFSURCL0) で非システム RDDS を作成し、IMS がシャットダウンまたは強制終了してコールド・スタートが必要になる前に IMSRSC リポジトリにエクスポートされなかった新規または変更済みリソースのみを保持するようにした場合は、IMS がコールド・スタートする前に RDDS to Repository ユーティリティ (CSLURP10) の非システム RDDS が定義をリポジトリに書き込むように指定することができます。

CSLURP10 ユーティリティからの出力は、入力 RDDS からコピーされたリソース定義によって更新されたリポジトリです。

重複したリソース定義がある場合 (非システム RDDS 内に存在する可能性があります)、入力 RDDS 内で検出された最後の定義のみがリポジトリに書き込まれます。

このユーティリティの出力には、リポジトリに書き込まれたリソース定義の数とリポジトリに書き込まれなかった検出済みの重複の数をリストする、要約セクションが含まれます。

ユーティリティによって発行されたメッセージは、SYSPRINT データ・セットに書き込まれます。入力定義の処理中のエラーまたはリポジトリーへの書き込み中のエラーが発生した場合、このユーティリティは戻りコード 8 を発行して終了します。

CSLURP10 ユーティリティのエラーは、SYSPRINT データ・セット内で CSL26xxE メッセージによって報告されます。

## JCL 仕様

JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1 EXEC PGM=CSLURP10,MEMLIMIT=4G
```

### DD ステートメント

#### JOBLIB / STEPLIB DD

ユーティリティの実行可能モジュールが入っている IMS.SDFSRESL を指します。連結データ・セットは有効です。

#### RDDSDSN DD

このユーティリティへの入力として使用される RDDSDSN の DSN を定義します。この DD ステートメントは必須です。RDDSDSN として指定できる DSN は 1 つだけです。連結データ・セットは無効です。

#### SYSPRINT DD

このユーティリティによって生成されたメッセージを受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FB,LRECL=133 です。

#### SYSIN DD

ユーティリティの入力パラメーターのソースを定義します。この DD ステートメントは必須です。SYSIN データ・セットの DCB パラメーターは、RECFM=FB,LRECL=80 です。

例えば、次のようになります。

```
//SYSIN DD *  
IMSPLEX(NAME=PLEX1 IMSID(IMS1,IMS2))
```

SYSIN DD ステートメントの入力パラメーターの形式は次のとおりです。

▶▶IMSPLEX(NAME=*plxnm*-IMSID(*imsx*)-)▶▶

#### NAME=

CSLURP10 ユーティリティが RM および SCI の登録に使用する IMSplex を指定します。この値は 1 文字から 5 文字で構成され、文字 CSL に付加されて IMSplex 名を形成します。指定する値は、RM および SCI

領域の IMSPLEX= に指定された値と同じでなければなりません。このパラメーターは必須であり、反復不可です。

### IMSID()

CSLURP10 ユーティリティーがリポジトリに書き込む IMS ID を指定します。このパラメーターは 1 から 4 文字で、反復可能です。指定できる IMS ID の最大数は 32 個です。SYSIN DD に IMS ID を指定しない場合、CSLURP10 はデフォルトとして RDDS ヘッダー内の IMS ID を使用します。このパラメーターはオプションです。

RDDS 内のリソース定義と記述子定義は、IMSID キーワードに指定された IMS ID のリストに適用されます。指定されているすべての IMS システムに、RDDS の同じリソース定義と記述子定義があります。RDDS が MSC システムから取得したものであり、複数の IMS ID が指定されている場合、すべての IMS システムには同じ SIDR および SIDL 定義があります。このユーティリティーを使用して定義を RDDS からリポジトリにエクスポートするか書き込むときに、指定された IMS ID の IMS は必ずしもアクティブでなくてかまいません。

### ユーティリティー制御ステートメント

現在、CSLURP10 ユーティリティーについて文書化されているユーティリティー制御ステートメントはありません。





### 戻りコード

CSLURP10 ユーティリティーのエラーは、SYSPRINT データ・セット内で CSL26xxE メッセージによって報告されます。完了時、このユーティリティーは以下のいずれかの戻りコードを発行して終了します。


コード 意味

- 0 ユーティリティーは正常に完了しました。
- 8 ユーティリティーは強制終了エラーを検出しました。返された CSL26xxE メッセージに、エラーの詳細が記載されています。


関連概念:

-  IMSRSC リポジトリの管理 (システム管理)
-  IMSRSC リポジトリの概要 (システム定義)
-  IMSRSC リポジトリへの MODBLKS リソース定義および記述子定義のエクスポート (システム定義)
-  動的リソース定義環境の保守 (システム定義)

関連タスク:

-  IMSRSC リポジトリにおけるリソース定義および記述子定義の作成 (システム定義)

関連資料:

-  CSL メッセージ (メッセージおよびコード)

## CSLURP10 ユーティリティーの例

これらの例では、RDDS to Repository ユーティリティー (CSLURP10) を実行するためのサンプル JCL と、要約報告書の例を示します。

### JCL の例

次のサンプル JCL を使用して、CSLURP10 ユーティリティーを実行できます。

```
//RDDS2RPO JOB ,USER,CLASS=A,MSGCLASS=X,NOTIFY=USER
//JOB LIB DD DSN=IMSV12.RESLIB
//STEP1 EXEC PGM=CSLURP10,MEMLIMIT=4G
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//*****
//* SPECIFY A VALID RDDS DSN FOR INPUT. CONCATENATION IS NOT */
//* ALLOWED. */
//*****
//RDDS DSN DD DSN=TEST.IMS.RDDS,DISP=SHR
//*
//*****
//* IMSID MAY BE SPECIFIED ON SYSIN OR WILL DEFAULT TO THE */
//* IMSID ON THE RDDS HEADER RECORD. WHEN USING SYSIN, COMMA */
//* SEPERATED IMSIDS MAY BE SPECIFIED. */
//*****
//SYSIN DD *
IMSPLEX(NAME=PLEX1 IMSID(IMS1))
/*
//
```

### 要約出力の例

次の例は、CSLURP10 ユーティリティーの要約出力を示しています。

Sample Output for CSLURP10


-----

```
CSL2603I CSLURP10 IS PROCESSING RDDS TEST.IMS.RDDS
CSL2618I CSLURP10 IS PROCESSING PLEX=CSLPLEX1, IMSID LIST FROM SYSIN, IMS1
CSL2620I CSLURP10 SUCCESSFUL REGISTRATION WITH RM, RMNAME=RM1RM
CSL2600I CSLURP10 WRITE TO REPOSITORY WAS SUCCESSFUL FOR REPOTYPE=IMSRSC , REPONAME=IMS_REPOS
```

\*\*\*\* SUMMARY \*\*\*\*

DB	COUNT	: 828
DBDESC	COUNT	: 0
PGM	COUNT	: 599
PGMDESC	COUNT	: 0
RTC	COUNT	: 89
RTCDESC	COUNT	: 0
TRAN	COUNT	: 664
TRANDESC	COUNT	: 0
DB	DUPLICATES:	0
DBDESC	DUPLICATES:	0
PGM	DUPLICATES:	0
PGMDESC	DUPLICATES:	0
RTC	DUPLICATES:	0
RTCDESC	DUPLICATES:	0
TRAN	DUPLICATES:	0
TRANDESC	DUPLICATES:	0

関連概念:

 [動的リソース定義の概要 \(システム定義\)](#)





---

## 第 34 章 RDDS コピー・ユーティリティー (DFSURCP0)

RDDS コピー・ユーティリティー (DFSURCP0) は、リソース定義データ・セット (RDDS) の内容を別の RDDS にコピーするために使用します。

新しい RDDS に関連した IMSID を提供するために、制御ステートメントが読み取られます。RDDS コピー・アクティビティーがカウントされ、要約報告書に組み込まれます。

DFSURCP0 ユーティリティーによる RDDS 作成の始めに、初期設定の状況を示している RDDS ヘッダー・レコードが書き込まれます。ユーティリティーが正常に完了した場合、ヘッダーは、RDDS が良好であることとコピーが行われた時刻を反映するように更新されます。

IMSID 制御ステートメントが指定された場合、RDDS ヘッダーはその IMSID を示します。この制御ステートメントが指定されなかった場合、IMSID は元のソース RDDS から保持されます。

RETAINTIME 制御ステートメントが指定された場合、RDDS ヘッダーには元のソース RDDS からのタイム・スタンプが格納されます。この制御ステートメントが指定されなかった場合、タイム・スタンプはコピーが完了した時刻を反映します。

これらのステップを実行するための JCL は、手動で作成できます。また、DFSURCP0 ユーティリティーを呼び出した場合、ISPF パネルで自動的に生成されます。

サブセクション:

- 『制約事項』
- 『前提条件』
- 702 ページの『要件』
- 702 ページの『推奨事項』
- 702 ページの『入力と出力』
- 702 ページの『JCL 仕様』
- 703 ページの『ユーティリティー制御ステートメント』
- 703 ページの『戻りコード』

### 制約事項

現在、DFSURCP0 ユーティリティーについて文書化されている制約事項はありません。

### 前提条件

現在、DFSURCP0 ユーティリティーについて文書化されている前提条件はありません。

## 要件

現在、DFSURCP0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSURCP0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

DFSURCP0 ユーティリティへの入力は、既存の RDDS (RDDSIN DD ステートメント) です。制御ステートメントは、CONTROL データ・セットから読み取られます。

リソース定義レコードの新しいコピーは、RDDS データ・セット (RDDSDSN DD ステートメント) に書き込まれます。ユーティリティによって発行されたメッセージは、SYSPRINT データ・セットに書き込まれます。要約報告書が REPORT データ・セットに書き込まれます。

## JCL 仕様

DFSURCP0 ユーティリティは、標準オペレーティング・システム・ジョブとして実行されます。JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1 EXEC PGM=DFSURCP0
```

### DD ステートメント

#### JOBLIB / STEPLIB DD

ユーティリティの実行可能モジュールが入っている IMS.SDFSRESL を指します。連結データ・セットを使用できます。

#### RDDSIN DD

ターゲット RDDS にコピーされるソース RDDS を定義します。この DD ステートメントは必須です。

#### RDDSDSN DD

ソース RDDS からリソース定義レコードを受け取るターゲット RDDS を定義します。この DD ステートメントは必須です。RDDSDSN データ・セットの DCB パラメーターは、RDDSIN DD ステートメントのソース RDDS からの DCB パラメーターと同じものであることが必要です。

#### SYSPRINT DD

ユーティリティによって生成されたメッセージを受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

## REPORT DD

ユーティリティによって生成された要約報告書を格納するデータ・セットを定義します。この DD ステートメントは必須です。REPORT データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

要約報告書には、以下のような情報が含まれています。

- ユーティリティへ提供された制御ステートメントのイメージ
- RDDSDSN および RDDSDSN DD ステートメントに関連したデータ・セット名
- RDDSDSN DD ステートメントから読み取られた項目の数

## CONTROL DD

ユーティリティの入力パラメーターのソースを定義します。この DD ステートメントは必須です。CONTROL データ・セットの DCB パラメーターは、RECFM=FB、LRECL=80 です。

## ユーティリティ制御ステートメント

### IMSID

ユーティリティによって生成される RDDSDSN データ・セットに関連した IMS サブシステムの名前を指定します。この名前は、IMS サブシステムに関連した 4 文字の ID です。

### RETAINTIME

RDDSDSN ヘッダー内のタイム・スタンプをどこから取得するかを示します。RETAINTIME 制御ステートメントが指定された場合、RDDSDSN ヘッダーには元のソース RDDSDSN からのタイム・スタンプが格納されます。この制御ステートメントが指定されなかった場合、タイム・スタンプはコピーが行われた時刻です。


## 戻りコード

以下の戻りコードが作成されます。

コード 意味

- 0 ユーティリティが正常に完了しました。
- 8 ユーティリティは強制終了エラーを検出しました。

関連概念:

 動的リソース定義の概要 (システム定義)

---

## DFSURCP0 ユーティリティの例

これらの例では、DFSURCP0 ユーティリティを実行するためのサンプル JCL と、要約報告書の例を示します。

### JCL の例

次のサンプル JCL を使用して、このユーティリティを実行できます。

```

//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOBLIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURCP0
//RDDSDSN DD DSN=[RDDSD source data set name],DISP=SHR
//RDDSDSN DD DSN=[RDDSD target data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//SYSPRINT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//REPORT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//CONTROL DD *
IMSID=imid
/*

```

## 要約出力の例

次の例では、サンプルの要約報告書を示し、報告書の解釈方法について説明します。

```

* *****
* RDDSD COPY / DFSURCP0 DATE: 2011/125 TIME: 23:05 PAGE: 1
* *****
*
* CONTROL: IMSID=SYS3
* CONTROL: RETAIN TIME
*
* RDDSD INPUT DATA SET NAME :
* IMSTESTL.IMS1.RDDSD1
*
* RDDSD OUTPUT DATA SET NAME:
* IMSTESTL.IMS1.RDDSD2
*
* IMSID USED . . . . : SYS3
*
* RDDSD COPY TIME . . : 2011.126 05:33:40.605599-UTC
* RDDSD BLKSIZE . . . : 32760
*
* *****
* RDDSD COPY INFORMATION SUMMARY
*
* TOTAL # OF NON COMMENT CONTROL STATEMENTS READ . . : 2
* TOTAL # OF RDDSD RECORDS READ . . . . . : 28
* TOTAL # OF RDDSD RECORDS WRITTEN . . . . . : 28
* TOTAL # OF RDDSD RECORDS UPDATED . . . . . : 1
* *****

```

---

## 第 35 章 Create RDDS from Log Records ユーティリティー (DFSURCL0)

Create RDDS from Log Records ユーティリティー (DFSURCL0) は、チェックポイント・ログ・レコード (X'40') およびタイプ 2 コマンド・レコード (X'22') から非システム・リソース定義データ・セット (RDDS) を作成するために使用します。非システム RDDS には、IMS 内のすべてのリソース定義か、あるいはパラメーターの STARTTIME、STOPTIME、または EXPORTNEEDED が指定されている場合は定義のサブセットを含めることができます。

RDDS に関連した IMSID に制御ステートメントが提供され、RDDS を作成するためにチェックポイント・ログ・レコードに関連したチェックポイント ID が再フォーマット指定されます。

チェックポイントは X'4001' レコード (チェックポイントの開始) で始まり、X'4098' レコード (チェックポイントの終わり) で終了します。完全なチェックポイント (この両方のレコードを含んでいるログ・データ・セットの連結) だけが処理対象として考慮されます。ユーティリティーによって検出された最初のチェックポイント・レコードが X'4001' でない場合は、最初の X'4001' レコードが検出されるまで、すべてのチェックポイント・レコードが無視されます。チェックポイントが CHKPTID 制御ステートメントによって明示的に選択されていても、X'4001' レコードが存在しない場合は、メッセージ DFS3997E が生成され、ユーティリティーは失敗します。ログ・データ・セット上の最後のチェックポイントは、CHKPTID 制御ステートメントが指定されていない場合は、暗黙的に選択されます。チェックが明示的または暗黙的に選択され、そのチェックポイントの X'4098' レコードが検出されない場合は、メッセージ DFS3991E が生成され、ユーティリティーは失敗します。

考慮されるログ・データ・セットの部分を制限するために、STARTTIME および STOPTIME 制御ステートメントによって時刻範囲を指定できます。時刻範囲を指定しなかった場合、処理は連結されたログ・データ・セット内の最初のレコードから始まり、ログ・データ・セット内の最後のレコードで終了します。

チェックポイントを指定するために、STARTTIME および STOPTIME パラメーターが直接使用されることはありません。これらのパラメーターは、チェックポイントとタイプ 2 コマンドのログ・レコードが評価される連結されたログ・データ・セットの境界として使用されます。STARTTIME と STOPTIME は 協定世界時 (UTC) として指定され、STCK 値に変換されます。それらの STCK 値が厳密に指定されている必要はなく、正しい精度レベルまでタイム・スタンプが埋め込まれます。

EXPORTNEEDED パラメーターを指定すると、IMSRSC リポジトリにエクスポートされていない、IMS ログ内のリソース定義のみを含む RDDS を生成できます。

以下のリソースおよび記述子の定義は、RDDS には書き込まれません。

- IMS で定義されたリソースおよび記述子

- HALDB 区画
- CPIC トランザクション
- エクスポート・オプションで作成されたのではなく、DFSINSX0 出口ルーチンによって作成された IMS リソース

ヘッダー・レコードおよび RDDS フォーマットのレコードは、RDDS データ・セットに書き込まれます。リソースと記述子、および RDDS の作成時に実行されたアクティビティがカウントされ、要約報告書に組み込まれます。

DFSURCL0 ユーティリティによる RDDS 作成の開始時に、INIT という RDDHD\_RDDSTAT 値を持つヘッダー・レコードが書き込まれます。ユーティリティが正常に完了した場合、ヘッダーは GOOD という RDDHD\_RDDSTAT 値を含むように更新されます。STOPTIME パラメーターを指定した場合、その値は RDDS\_STCKE に反映されます。そうでない場合、RDDS\_STCKE 値には、RDDS に追加された最後のログ・レコードの Store Time Clock (STCK) 値が反映されません。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 707 ページの『入力と出力』
- 708 ページの『JCL 仕様』
- 709 ページの『ユーティリティ制御ステートメント』
- 711 ページの『戻りコード』

## 制約事項

現在、DFSURCL0 ユーティリティについて文書化されている制約事項はありません。

## 前提条件

現在、DFSURCL0 ユーティリティについて文書化されている前提条件はありません。

## 要件

現在、DFSURCL0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

- 多くの場合、チェックポイントを指定する必要はありません。指定しなかった場合、ユーティリティはログの連結内の最後のチェックポイントを選択します。
- 特定のチェックポイントを検出したい場合は、CHKPTID パラメーターを使用します。CHKPTID 値は YYDDD/HHMMSS のフォーマットで指定します。

- STARTTIME および STOPTIME パラメーターを使用した場合、評価されるログ・レコードの範囲が狭くなります。同じチェックポイント処理を使用してください。
- 大部分の処理では、STARTTIME および STOPTIME パラメーターは必要ありません。これらのパラメーターは、以下の状況で役立つ場合があります。
  - STARTTIME パラメーターは、大きなログ・データ・セットまたは複数のログ・データ・セットを連結してあり、処理を迅速にするためにユーティリティーでデータをスキップできるようにしたい場合に役立つことがあります。また、このパラメーターは、明示的なチェックポイント ID が不明で、チェックポイントが発生する前の時点が分かっている場合にも便利です。
  - STOPTIME パラメーターは、選択したチェックポイントの後に発生するいくつかの X'22' レコードを除外したい場合に役立つことがあります。

推奨事項: EXPORTNEEDED パラメーターを指定する場合は、STARTTIME または STOPTIME を指定しないでください。これにより、IMSRSC リポジトリにエクスポートされていないすべてのリソース定義を RDDS に書き込むことができます。

STARTTIME および STOPTIME パラメーターで指定する値は、ログ・レコードの接尾部の STCK 値に基づいている必要があります。接尾部の STCK 値以外のログ・レコードからの時刻値を使用しないでください。他の時刻値は、その使用状況や、うるう秒に関する調整のために、異なっている場合があるからです。

適切な STCK 値を判別するには、DFSERA30 モジュールを使用してログ・レコードをフォーマット設定するか、それと等価のログ・フォーマット設定ユーティリティーを使用して STCK 値を表示します。

次の例は、DFSERA30 モジュールからの出力を使用して STCK 値を判別する方法を示しています。

```
DFSERA30 - FORMATTED LOG PRINT
22 RECORD - 2008-06-25 20:25:04.339239 UTC <--- STCK 値
00000000 000000 00F80000 2208C023 C3D9C540 40404040
00000020 000020 00000000 2008177F 20253133 8995028D
00000040 000040 00000000 00000000 000000A0 00000008
00000060 000060 00000000 00000060 00000000 00000000
00000080 000080 8CF8C000 000A0700 00FF0000 00020000
000000A0 0000A0 00000000 00000000 00000000 00000000
```

Create RDDS from Log Records ユーティリティー (DFSURCL0) に EXPORTNEEDED 制御ステートメントを指定する場合は、入力として使用する IMS ログを生成した IMS のバージョンと同じバージョンの DFSURCL0 ユーティリティーを実行することをお勧めします。例えば、IMS 14 DFSURCL0 ユーティリティーは IMS 14 ログ・データ・セットで使用してください。そうでないと、予測不能な結果が生じる可能性があります。

## 入力と出力

DFSURCL0 ユーティリティーへの入力は IMS ログで、これには RDDS 内のデータを作成するために使用されたログ・レコードが入っています。制御ステートメントは (CONTROL DD ステートメントによって指定された) CONTROL データ・セットから読み取られます。

DFSURCL0 ユーティリティは、チェックポイント・ログ・レコード (X'4004'、X'4006'、X'4007'、および X'4083') とタイプ 2 コマンド・レコード (X'2208'、X'2205'、X'2206'、および X'2207') の内容を、対応するランザクション、データベース、プログラム、およびルーティング・コード・リソースの RDDS フォーマットのレコードに変換します。ヘッダーおよびリソース・レコードは、RDDS データ・セット (RDDSDSN DD ステートメント) に書き込まれます。ユーティリティによって発行されたメッセージは、SYSPRINT データ・セットに書き込まれます。要約報告書が REPORT データ・セットに書き込まれます。

## JCL 仕様

DFSURCL0 ユーティリティは、標準オペレーティング・システム・ジョブとして実行されます。JOB ステートメント、EXEC ステートメント、および DD ステートメントを定義する必要があります。

**EXEC** ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1 EXEC PGM=DFSURCL0
```

**DD** ステートメント

### JOBLIB / STEPLIB DD

このユーティリティの実行可能モジュールが入っている IMS.SDFSRESL データ・セットを指します。連結データ・セットを使用できます。

### SYSUT1 DD

ユーティリティへの入力として使用される IMS ログ・データ・セットを定義します。この DD ステートメントは必須です。連結データ・セットを使用できます。

### RDDSDSN DD

ユーティリティによって生成されたリソース定義レコードを受け取る RDDS を定義します。この DD ステートメントは必須です。RDDSDSN データ・セット用の DCB パラメーターは、LRECL=32756、BLKSIZE=32760、および RECFM=VB です。

### WORKFILE DD

中間的な報告書の詳細を、それが REPORT DD ステートメントで印刷用に選択されるまで保持するデータ・セットを定義します。この DD ステートメントは必須です。WORKFILE データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。

### REPORT DD

ユーティリティによって生成されたメッセージを受け取るデータ・セットを定義します。この DD ステートメントは必須です。REPORT データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。

### CONTROL DD

ユーティリティの入力パラメーターのソースを定義します。この DD ステートメントは必須です。CONTROL データ・セットの DCB パラメーターは、RECFM=FB、LRECL=80 です。



## ユーティリティー制御ステートメント

### IMSID

ユーティリティーによって生成される RDDS に関連した IMS サブシステムの名前を指定します。このパラメーターに関連付けられる値は、IMS サブシステムに関連した 4 文字の ID です。このパラメーターはオプションです。このパラメーターを省略した場合、IMSID は X'4001' チェックポイント・レコードから抽出されます。

### CHKPTID

RDDS データ・セットの作成の基礎として使用されるチェックポイントを選択します。このチェックポイント ID に関連したチェックポイント・レコードが処理され、このチェックポイントに続く後続のすべての X'22' レコードが、次のチェックポイント ID、STOPTIME 指定、またはログの終わりまで処理されます。このパラメーターはオプションです。これを省略した場合は、ログ内にある最後のチェックポイント ID が使用されます。

チェックポイント ID の値は、以下の複数の場所で検出できます。

- IMS システム・ログ内の DFS3804I LATEST RESTART CHKPT メッセージは、IMS がチェックポイントをとるたびに生成されます。このメッセージからチェックポイント ID を抽出してください。CHKPTID パラメーターの構文は、メッセージ内のチェックポイント ID のフォーマットに一致します。
- X'4001' チェックポイントのレコード。この場合、チェックポイント ID はオフセット X'0C' の位置に出現します。

### EXPORTNEEDED

IMS ログから、およびまだ IMSRSC リポジトリにエクスポートされていないリソース定義に関する IMS ログ内の X'22' ログ・レコードから、非システム RDDS データ・セットを作成することを指定します。

出力の非システム RDDS は、以下のいずれかの目的に使用できます。

- IMS コールド・スタート前に定義を IMSRSC リポジトリに書き込むために、CSLURP10 ユーティリティーで使用できます。
- IMS コールド・スタート後に IMSRSC リポジトリからの IMPORT コマンドに指定できます。

EXPORTNEEDED 制御ステートメントと一緒に STARTTIME または STOPTIME、あるいはこの両方を指定すると、このリポジトリにエクスポートする必要があり、なおかつ、作成または更新の時間が指定された範囲の STARTTIME と STOPTIME にあるリソース定義のみを含む、非システム RDDS が作成されます。

出力の非システム RDDS には、IMS ログ内の一部のリソース定義が含まれない可能性があるため、この RDDS を IMS コールド・スタートには使用できません。

EXPORT コマンドによってリソース定義が IMS に正常に書き込まれても、IMS がシステム・チェックポイントを取っていなかった場合、その後 IMS が異常終了するエラーが発生すると、それらのリソース定義は IMS ログ内で要エクスポートのマークが付けられたままになります。このシナリオでは、IMS をコールド・スタートする必要があり、EXPORTNEEDED を

指定した DFSURCL0 ユーティリティを実行すると、このユーティリティによって生成される RDDS には、前回正常に実行された EXPORT コマンドから得られたリソース定義が含まれます。以下にこのシナリオのタイムラインを示します。

1. ユーザーがリソースと記述子を作成または更新します。
2. ユーザーが IMS チェックポイントを取ります。前のステップで作成されたリソースが、「要エクスポート」インディケーター付きで IMS ログに書き込まれます。
3. ユーザーが追加のリソース定義を更新します。
4. ユーザーがコマンド EXPORT DEFN TARGET(REPO) OPTION(CHANGESONLY) を使用して、リソース定義と記述子定義を IMS から IMSRSC リポジトリにエクスポートします。ステップ 1 と 3 で作成または変更されたすべてのリソースがリポジトリに書き込まれます。
5. 別のチェックポイントが取られる前に IMS が異常終了します。
6. ユーザーが EXPORTNEEDED 制御ステートメントを指定して DFSURCL0 ユーティリティを実行します。ステップ 2 でのログ内の IMS チェックポイントおよびステップ 3 でのログ・レコードから、RDDS が作成されます。ステップ 4 でエクスポートされたリソースは、引き続き DFSURCL0 によって作成された RDDS 内に存在します。
7. IMS のコールド・スタートが必要になったために、ステップ 5 より後に変更が行われていないことを確認するには、ステップ 6 で作成された RDDS を使用して、リポジトリ内のリソース定義が、IMS システムが異常終了してコールド・スタートが必要になった時点より前の IMS システム内のリソース定義と一致していることを確認する必要があります。

#### **NODETAIL**

X'22' レコード内のデータによって変更されたリソースの名前が要約報告書に含まれないことを示します。このパラメーターはオプションです。このパラメーターを省略した場合、変更されたすべてのリソースの名前が報告書に組み込まれます。

#### **STARTTIME**

処理を開始するログ上の地点の判別に使用するタイム・スタンプを指定します。レコード上の時刻を判別するために、ログ・レコードの接尾部内の STCK が使用されます。タイム・スタンプがそれより前のログ・レコードは無視されます。処理は、STARTTIME 指定に続く次の開始チェックポイント・レコード (X'4001') で始まります。開始チェックポイント・レコードの後にチェックポイントの途中で発生するタイム・スタンプ、後続のすべてのチェックポイント・レコード、および X'22' レコードは、次の開始チェックポイント・レコードが検出されるまで無視されます。

このパラメーターはオプションです。これを省略した場合、処理はログ内の最初のレコードから開始されます。このパラメーターは、指定した開始時刻より前のレコードを無視する特定の理由がある場合にだけ、指定してください。

CHKPTID と STARTTIME の両方のパラメーターを指定し、STARTTIME で指定した時刻が、指定した CHKPTID ステートメントの開始チェックポイント・レコードより後の時刻である場合、指示されたチェックポイントは検出されず、処理は戻りコード 8 で終了します。

STARTTIME パラメーターのフォーマットは 16 文字のストリングで、YYYYDDDDHHMMSSTHM です。STARTTIME 値は UTC で指定する必要があります。データが 16 文字より少ない場合は、右側にゼロが埋め込まれます。

### STOPTIME

処理を停止するログ上の地点の判別に使用するタイム・スタンプを指定します。レコード上の時刻を判別するために、ログ・レコードの接尾部内の STCK 値が使用されます。タイム・スタンプがそれより後のログ・レコードは無視されます。このパラメーターはオプションです。これを省略した場合、処理はログに含まれる最後のレコードまで続行されます。

STOPTIME パラメーターのフォーマットは 16 文字のストリングで、YYYYDDDDHHMMSSTHM です。STOPTIME 値は UTC で指定する必要があります。データが 16 文字より少ない場合は、右側に次のような埋め込みが行われます。

- HHMMSS の値が省略されていれば、必要に応じて値「235959」から埋め込まれます。
- それ以外の位置には 9 が埋め込まれます。


### 戻りコード

以下の戻りコードが作成されます。


コード 意味

- 0 ユーティリティーが正常に完了しました。
- 8 ユーティリティーは強制終了エラーを検出しました。

関連概念:

 [動的リソース定義の概要 \(システム定義\)](#)

関連タスク:

 1 つ以上の IMS システムが停止しているが、RM システムがアクティブな場合の IMSRSC リポジトリ・データ・セットのリカバリー (オペレーションおよびオートメーション)

---

## DFSURCL0 ユーティリティーの例

これらの例では、DFSURCL0 ユーティリティーを実行するためのサンプル JCL と、要約報告書の例を示します。

### JCL の例

次のサンプル JCL を使用して、このユーティリティーを実行できます。

```

//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOB LIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURCL0
//SYSUT1 DD DSN=[Log data set name(s)],DISP=SHR
//RDDSDSN DD DSN=[RDDSDSN data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//WORKFILE DD DSN=[Workfile data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//REPORT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//CONTROL DD *
IMSID=imid
CHKPTID=yyddd/hhmmss
EXPORTNEEDED
/*
//

```

## 要約報告書の例

次の例では、サンプルの要約報告書を示し、報告書の解釈方法について説明します。

```

* *****
* RDDS CREATION / DFSURCL0 DATE: 2011/126 TIME: 11:48 PAGE: 1
* *****
*
* CONTROL: IMSID=SYS3
* CONTROL: CHKPTID=11126/113514
*
*
* RDDS DATA SET NAME:
* IMSTESTL.IMS1.RDDSDSN
*
* INPUT LOG DATA SET NAME(S):
* IMSTESTL.IMS01.OLDSPO
*
* *****
* RDDS CREATION UTILITY INFORMATION SUMMARY
*
* LIST OF CHKPTIDS CONTAINED ON LOG:
* CHKPTID: 11126/113514
* RDDS DATASET BUILT FROM CHKPTID: 11126/113514
* CREATED DB DBX
* CREATED DB DBY
* CREATED DB DBZ
* CREATED PGM PGMX
* CREATED PGM PGMY
* UPDATED PGM PGMY
* CREATED TRAN TRANX
* CREATED TRAN TRANY
* DELETED TRAN TRANY
* CREATED RTC RTCX
* CREATED RTC RTCY
* UPDATED RTC RTCX
* CHKPTID: 11126/114444
*
* IMSID USED: SYS3
*
* TIME STAMP ON FIRST LOG RECORD . . . . . : 2011126183457995
* TIME STAMP ON LAST LOG RECORD . . . . . : 2011126184444866
*
* TOTAL # OF NON COMMENT CONTROL RECORDS READ . . . . . : 2
* TOTAL # OF LOG RECORDS READ . . . . . : 703
* TOTAL # OF LOG RECORDS SKIPPED. . . . . : 0

```

```

*
* SUPPLIED CHKPTID: 11126/113514 WAS FOUND ON LOG AND PROCESSED
* STATISTICS FOR THIS CHKPTID:
*
* TRANSACTION MODEL : DFSDSTR1
* DATABASE MODEL    : DFSDSDB1
* PGM MODEL         : DFSDSPG1
* ROUTE CODE MODEL  : DBFDSRT1
*
* RESULT OF PROCESSING CHECKPOINT AND TYPE-2 COMMAND RECORDS:
* TOTAL # OF TRAN RSC DEFS READ FROM CHKP RECS . . . . . 668
* TOTAL # OF TRAN RSC DEFS CREATED FROM X22 RECS . . . . . 2
* TOTAL # OF TRAN RSC DEFS UPDATED FROM X22 RECS . . . . . 0
* TOTAL # OF TRAN RSC DEFS DELETED FROM X22 RECS . . . . . 1
* TOTAL # OF TRAN RSC DEFS NOT EXPORTED FROM RECORDS . . . 0
* TOTAL # OF TRAN RSC DEFS WRITTEN TO RDDS . . . . . 669
*
* TOTAL # OF TRAN DESC DEFS READ FROM CHKP RECS . . . . . 1
* TOTAL # OF TRAN DESC DEFS CREATED FROM X22 RECS . . . . . 0
* TOTAL # OF TRAN DESC DEFS UPDATED FROM X22 RECS . . . . . 0
* TOTAL # OF TRAN DESC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF TRAN DESC DEFS NOT EXPORTED FROM RECORDS . . . 1
* TOTAL # OF TRAN DESC DEFS WRITTEN TO RDDS . . . . . 0
*
* TOTAL # OF DB RSC DEFS READ FROM CHKP RECS . . . . . 836
* TOTAL # OF DB RSC DEFS CREATED FROM X22 RECS . . . . . 3
* TOTAL # OF DB RSC DEFS UPDATED FROM X22 RECS . . . . . 0
* TOTAL # OF DB RSC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF DB RSC DEFS NOT EXPORTED FROM RECORDS . . . 0
* TOTAL # OF DB RSC DEFS WRITTEN TO RDDS . . . . . 839
*
* TOTAL # OF DB DESC DEFS READ FROM CHKP RECS . . . . . 1
* TOTAL # OF DB DESC DEFS CREATED FROM X22 RECS . . . . . 0
* TOTAL # OF DB DESC DEFS UPDATED FROM X22 RECS . . . . . 0
* TOTAL # OF DB DESC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF DB DESC DEFS NOT EXPORTED FROM RECORDS . . . 1
* TOTAL # OF DB DESC DEFS WRITTEN TO RDDS . . . . . 0
*
* TOTAL # OF PGM RSC DEFS READ FROM CHKP RECS . . . . . 605
* TOTAL # OF PGM RSC DEFS CREATED FROM X22 RECS . . . . . 2
* TOTAL # OF PGM RSC DEFS UPDATED FROM X22 RECS . . . . . 1
* TOTAL # OF PGM RSC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF PGM RSC DEFS NOT EXPORTED FROM RECORDS . . . 1
* TOTAL # OF PGM RSC DEFS WRITTEN TO RDDS . . . . . 606
*
* TOTAL # OF PGM DESC DEFS READ FROM CHKP RECS . . . . . 1
* TOTAL # OF PGM DESC DEFS CREATED FROM X22 RECS . . . . . 0
* TOTAL # OF PGM DESC DEFS UPDATED FROM X22 RECS . . . . . 0
* TOTAL # OF PGM DESC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF PGM DESC DEFS NOT EXPORTED FROM RECORDS . . . 1
* TOTAL # OF PGM DESC DEFS WRITTEN TO RDDS . . . . . 0
*
* TOTAL # OF RTC RSC DEFS READ FROM CHKP RECS . . . . . 94
* TOTAL # OF RTC RSC DEFS CREATED FROM X22 RECS . . . . . 2
* TOTAL # OF RTC RSC DEFS UPDATED FROM X22 RECS . . . . . 1
* TOTAL # OF RTC RSC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF RTC RSC DEFS NOT EXPORTED FROM RECORDS . . . 0
* TOTAL # OF RTC RSC DEFS WRITTEN TO RDDS . . . . . 96
*
* TOTAL # OF RTC DESC DEFS READ FROM CHKP RECS . . . . . 1
* TOTAL # OF RTC DESC DEFS CREATED FROM X22 RECS . . . . . 0
* TOTAL # OF RTC DESC DEFS UPDATED FROM X22 RECS . . . . . 0
* TOTAL # OF RTC DESC DEFS DELETED FROM X22 RECS . . . . . 0
* TOTAL # OF RTC DESC DEFS NOT EXPORTED FROM RECORDS . . . 1
* TOTAL # OF RTC DESC DEFS WRITTEN TO RDDS . . . . . 0
*
* TOTAL # OF RDDSDSN RECORDS WRITTEN . . . . . 217
* TOTAL # OF RESOURCE DETAIL LINES WRITTEN . . . . . 12
*
* CHECKPOINT LOG RECORDS:
* TOTAL # OF X4001 LOG RECORDS READ . . . . . 1

```

```

*
* SIZE OF X4004 RECORDS (BYTES) . . . . . 3944
* TOTAL # OF X4004 LOG RECORDS READ . . . . . 43
* TOTAL # OF RESOURCE X4004 LOG RECORDS CREATED . . . . . 1
* TOTAL # OF DESCRIPTOR X4004 LOG RECORDS CREATED . . . . . 0
* TOTAL # OF X4004 LOG RECORDS DISCARDED . . . . . 1
* TOTAL # OF TRAN/(SMB) RECORDS WRITTEN . . . . . 43
*
* SIZE OF X4006 RECORDS (BYTES) . . . . . 928
* TOTAL # OF X4006 LOG RECORDS READ . . . . . 106
* TOTAL # OF RESOURCE X4006 LOG RECORDS CREATED . . . . . 1
* TOTAL # OF DESCRIPTOR X4006 LOG RECORDS CREATED . . . . . 0
* TOTAL # OF X4006 LOG RECORDS DISCARDED . . . . . 1
* TOTAL # OF DB/(DDIR) RECORDS WRITTEN . . . . . 106
*
* SIZE OF X4007 RECORDS (BYTES) . . . . . 964
* TOTAL # OF X4007 LOG RECORDS READ . . . . . 56
* TOTAL # OF RESOURCE X4007 LOG RECORDS CREATED . . . . . 1
* TOTAL # OF DESCRIPTOR X4007 LOG RECORDS CREATED . . . . . 0
* TOTAL # OF X4007 LOG RECORDS DISCARDED . . . . . 1
* TOTAL # OF PGM/(PDIR) RECORDS WRITTEN . . . . . 56
*
* SIZE OF X4083 RECORDS (BYTES) . . . . . 1016
* TOTAL # OF X4083 LOG RECORDS READ . . . . . 10
* TOTAL # OF RESOURCE X4083 LOG RECORDS CREATED . . . . . 1
* TOTAL # OF DESCRIPTOR X4083 LOG RECORDS CREATED . . . . . 0
* TOTAL # OF X4083 LOG RECORDS DISCARDED . . . . . 1
* TOTAL # OF RTC/(RCTE) RECORDS WRITTEN . . . . . 10
*
* TOTAL # OF X4098 LOG RECORDS READ . . . . . 1
*
* TYPE-2 COMMAND LOG RECORDS:
* TOTAL # OF X22 LOG RECORDS READ . . . . . 12
* TOTAL # OF X2202 (DB UPD ACTCTYPE) RECS . . . . . 0
* TOTAL # OF X2205 (DB) RECORDS READ . . . . . 3
* TOTAL # OF X2206 (PGM) RECORDS READ . . . . . 3
* TOTAL # OF X2207 (RTC) RECORDS READ . . . . . 3
* TOTAL # OF X2208 (TRAN) RECORDS READ . . . . . 3
* TOTAL # OF X22 IMPORT RECORDS READ . . . . . 0
* TOTAL # OF IMPORT (DB) RECORDS READ . . . . . 0
* TOTAL # OF IMPORT (PGM) RECORDS READ . . . . . 0
* TOTAL # OF IMPORT (RTC) RECORDS READ . . . . . 0
* TOTAL # OF IMPORT (TRAN) RECORDS READ . . . . . 0
* TOTAL # OF OTHER X22 RECORDS READ . . . . . 0
*
* OTHER STATISTICS:
* TOTAL # OF X45FF LOG RECORDS READ . . . . . 2
*
* *****

```

この要約報告書の例では、以下のとおりです。

- 処理されるチェックポイントを指定するために、キーワード  
CHKPTID=11126/113514 を含んでいる制御ステートメントが提供されます。
- 対象のログ・データ・セットが識別されます。
- ログに、RDDS の作成に使用されたチェックポイント 11126/113514 が含まれて  
いました。タイプ 2 コマンド・ログ・レコードに基づいて作成、更新、または  
削除された特定のリソースが識別されています。
- ログに関連した IMSID は SYS3 として識別されました。
- 最初と最後に読み取られたログ・レコードのタイム・スタンプが示されていま  
す。
- ユーティリティー処理を指示するために、1 つの制御ステートメントが読み取ら  
れました。
- 703 個のログ・レコードが読み取られました。

- CHKPTID 制御ステートメントによって指定されたチェックポイント ID がログ上で検出され、処理されました。
- 指示されたチェックポイント ID について、有意のチェックポイント・レコードの明細と、それらのレコードの属性が次のように提示されています。
  - 1 つの X'4001' レコードが読み取られました。
  - X'4004'、X'4006'、X'4007'、および X'4083' のそれぞれのレコードでは、以下のとおりです。
    - レコードのサイズが示されました。
    - 読み取られたレコードの数が示されました。
    - タイプ 2 コマンド・ログ・レコードから抽出されたデータに基づいて、新規リソースを格納するために作成された可能性のある RDDS レコードの数が示されました。
    - 評価から廃棄された可能性のある RDDS レコードの数が示されました。レコードが廃棄される理由は、以下のとおりです。
      - それらのレコードには、IMS の内部で定義されたりソースと記述子だけが入っている。
      - レコード上に含まれていたすべてのリソースは、タイプ 2 コマンド・ログ・レコードの処理の結果として削除された。
    - 示されたりソース・タイプについて書き込まれた RDDS レコードの数。
- 1 つの X'4098' レコードが読み取られました。
- タイプ 2 コマンド・レコードに関連した統計が提示されました。
  - 総数で 12 個のタイプ 2 コマンド・ログ・レコードが読み取られました。
  - X'2205'、X'2206'、X'2207'、X'2208' のそれぞれのリソース・レコードに含まれていたデータに基づいて、以下の統計が示されました。
    - 読み取られたレコードの数
    - RDDS に書き込まれたリソースの総数
    - 作成されたリソースの数
    - 更新されたリソースの数
    - 削除されたリソースの数
- 2 つの X'45FF' レコード (システム統計レコード) が読み取られました。
- 217 レコードが (RDDSDSN DD ステートメントによって識別された) RDDS に書き込まれました。
- 作成、更新、および削除されたリソースを識別する 12 行が要約報告書に書き込まれました。





---

## 第 36 章 Create RDDS from MODBLKS ユーティリティー (DFSURCM0)

Create RDDS from MODBLKS ユーティリティー (DFSURCM0) は、MODBLKS データ・セットからデータを抽出し、そのデータを再フォーマット設定して RDDS を作成するために使用します。

リソース定義データ・セット (RDDS) に関連した IMSID と MODBLKS メンバーに関連した接尾部を識別するために、制御ステートメントが読み取られます。DFSMBx、DFSDDIRx、DFSPDIRx、および DFSRCTEx の各 MODBLKS PDS メンバーからのデータは、RDDS へのロード用にフォーマット設定されます。RDDS ヘッダー・レコードが作成されます。そのヘッダー・レコードとチェックポイントに似たレコードが RDDS に書き込まれます。RDDS 作成アクティビティーがカウントされ、要約報告書に組み込まれます。

トランザクションに編集ルーチンが (例えば、TRANSACT マクロ上の EDIT パラメーターで) 割り当てられている場合、DFSURCM0 ユーティリティーは以下のようにしてそのルーチンを識別します。

- DFSURST0 ユーティリティーによって作成された、トランザクション ID と編集ルーチンの相互参照ファイルを使用します。
- IMS 中核モジュールから、情報を抽出します。

特定の状況では、すべての IMS サブシステムに以下の MODBLKS メンバー・タイプのすべてが含まれているわけではありません。

- SMB ブロックは、DB DC / DCCTL についてのみ評価され、必須です。
- DDIR ブロックは、DB DC / DBCTL についてのみ評価され、必須です。
- PDIR ブロックは、すべてのシステム・タイプについて評価され、必須です。
- RCTE ブロックは、DB DC / DCCTL についてのみ評価されますが、高速機能をサポートしているシステム上でのみ検出されます。

中核モジュール (DFSVNUCx) は、以下の 3 つのソースのいずれかから、以下の検索順序で抽出されます。

- SDFSRESL DD ステートメント

DFSURCM0 ユーティリティーを実行するための実行可能ファイルがリンク・リストに入っている場合に (このとき JOBLIB および STEPLIB DD ステートメントは JCL ストリーム内にコーディングされない)、または対象の中核が JOBLIB または STEPLIB DD ステートメント内で指定されたものと異なる IMS.SDFSRESL データ・セットに含まれている場合に、ターゲットの中核が入っている IMS.SDFSRESL データ・セットを識別するオプションのステートメント。

- STEPLIB DD ステートメント

DFSURCM0 ユーティリティー STEPLIB DD ステートメントの実行可能ファイルが入っている IMS.SDFSRESL データ・セットを識別します。

- JOBLIB DD ステートメント

DFSURCM0 ユーティリティの実行可能ファイルが入っている IMS.SDFSRESL データ・セットを識別します。

STEPLIB と JOBLIB の両方の DD ステートメントを指定した場合、STEPLIB が JOBLIB より優先し、STEPLIB の内容が評価されます。

RDDS 作成の開始時に、INIT という RDDHD\_RDDSTAT 値を持つヘッダー・レコードが書き込まれます。このユーティリティが正常に完了した場合、ヘッダーは GOOD という RDDHD\_RDDSTAT 値を含むように更新され、RDDS STCKE 値はユーティリティの実行時間を反映するように更新されます。

DFSURCM0 ユーティリティを使用すると、以下の準備手順を追加することにより、システム生成マクロから RDDS を生成することができます。

- ステージ 1 を事前構文解析してパラメーターを変更し、DFSURST0 ユーティリティを使用してサブシステム情報を抽出する
- IMS システム・マクロからステージ 1 IMS MODBLKS 生成を行い、ステージ 2 を作成する
- 作業データ・セット (OBJLIB、MODBLKS) を割り振る
- 割り振った作業データ・セットを参照するために、ステージ 2 JCL を編集する
- 生成されたステージ 2 を実行して、MODBLKS に似たデータ・セットを作成する
- MODBLKS に似たデータ・セットを DFSURCM0 への入力として使用する

これらのステップを実行する JCL ステートメントは、手動で作成できます。あるいは、「Create RDDS from MODBLKS」の ISPF パネルを呼び出したときに自動的に作成されます。

このユーティリティを使用して、以下のタスクを実行することにより、MODBLKS またはシステム生成マクロから CREATE コマンドを生成することもできます。

- MODBLKS またはシステム生成マクロから RDDS データ・セットを作成する。
- RDDS 抽出ユーティリティを使用し、作成された RDDS および OUTPUT=CMD 制御ステートメントをユーティリティへの入力として使用して、CREATE コマンドを作成する。

サブセクション:

- 719 ページの『制約事項』
- 719 ページの『前提条件』
- 719 ページの『要件』
- 719 ページの『推奨事項』
- 719 ページの『入力と出力』
- 719 ページの『JCL 仕様』
- 721 ページの『ユーティリティ制御ステートメント』
- 721 ページの『戻りコード』

## 制約事項

現在、DFSURCM0 ユーティリティについて文書化されている制約事項はありません。

## 前提条件

現在、DFSURCM0 ユーティリティについて文書化されている前提条件はありません。

## 要件

現在、DFSURCM0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSURCM0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

DFSURCM0 ユーティリティへの入力は MODBLKS データ・セットまたは MODBLKS に似たデータ・セットであり、SMB、DDIR、PDIR、および RCTE リソースのリストが入っているメンバーを伴っています。制御ステートメントは、CONTROL データ・セットから読み取られます。

DFSURCM0 ユーティリティは MODBLKS データ・セット内のリソース定義を取得して、それらの定義を RDDDS によってサポートされている定義フォーマットに変換し、新しい定義を RDDDS 内に保管します。ヘッダーおよびリソース定義レコードは、(RDDSDSN DD ステートメントによって指定された) RDDDS に書き込まれます。ユーティリティによって発行されたメッセージは、SYSPRINT データ・セットに書き込まれます。要約報告書が REPORT データ・セットに書き込まれます。

## JCL 仕様

DFSURCM0 ユーティリティは、標準オペレーティング・システム・ジョブとして実行されます。JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1 EXEC PGM=DFSURCM0
```

### DD ステートメント

#### JOBLIB / STEPLIB DD

ユーティリティの実行可能モジュールが入っている IMS.SDFSRESL データ・セットを指します。このデータ・セットには、トランザクション入力編集ルーチンの名前の抽出元となる中核モジュール DFSVNUCx も入っています。連結データ・セットを使用できます。

### **SDFSRESL DD**

トランザクション入力編集ルーチンの名前の抽出元となる中核モジュール DFSVNUCx が入っている IMS.SDFSRESL データ・セットを指します。このデータ・セットは、JOB LIB または STEPLIB DD ステートメントによって参照されるデータ・セットと同じデータ・セットの場合があります。IMS.SDFSRESL データ・セットは、オプションのデータ・セットです。これを省略した場合は、JOB LIB または STEPLIB DD から中核モジュールが取得されます。この DD ステートメントを指定するのは、指定する IMS.SDFSRESL データ・セットが JOB LIB または STEPLIB DD ステートメントで指定したものと異なる場合か、DFSURCM0 ユーティリティの実行可能モジュールをリンク・リストから取得する場合 (この場合は、JOB LIB および STEPLIB DD ステートメントを指定しません) だけにしてください。

### **MODBLKS DD**

ユーティリティへの入力として使用される MODBLKS データ・セットを定義します。この DD ステートメントは必須です。連結データ・セットを使用できません。

### **RDDSDSN DD**

ユーティリティによって生成されたリソース定義レコードを受け取る RDDSDSN を定義します。この DD ステートメントは必須です。RDDSDSN データ・セット用の DCB パラメーターは、LRECL=32756、BLKSIZE=32760、RECFM=VB です。

### **SYSPRINT DD**

ユーティリティによって生成されたメッセージを受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

### **REPORT DD**

ユーティリティによって生成された要約報告書を格納するデータ・セットを定義します。この DD ステートメントは必須です。REPORT データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

要約報告書には、以下のような機能と統計が含まれています。

- ユーティリティへ提供された制御ステートメントのイメージ
- 各リソース・タイプの MODBLKS データ・セット・メンバーの名前とサイズ
- 各リソース・タイプの MODBLKS データ・セットから読み取られた項目の数
- 各リソース・タイプの RDDSDSN に書き込まれたレコードとリソースの数

### **CONTROL DD**

ユーティリティの入力パラメーターのソースを定義します。この DD ステートメントは必須です。CONTROL データ・セットの DCB パラメーターは、RECFM=FB、LRECL=80 です。

## ユーティリティー制御ステートメント

以下の制御ステートメントを明示的に指定することも、DFSURST0 ユーティリティーを DFSURCM0 ユーティリティーの実行前に実行する場合は、DFSURST0 によって生成される制御ステートメントを DFURCM0 への入力として使用することもできます。

DFSURST0 によって生成された制御ステートメントを使用する場合は、DFSURST0 JCL の OUTPARMS DD ステートメントに示されたデータ・セットを、DFSURCM0 ユーティリティー JCL 内の CONTROL DD ステートメントとして連結します。

### IMSID

ユーティリティーによって生成される RDDS データ・セットに関連した IMS サブシステムの名前を指定します。このパラメーターに関連付けられる値は、IMS サブシステムに関連した 4 文字の ID です。

### SYSTYPE

RDDS の使用対象となる IMS サブシステムのタイプを表す 1 文字から 4 文字の値を指定します。このパラメーターはオプションです。SYSTYPE のデフォルト値は DBDC です。

有効な SYSTYPE 値は、以下のとおりです。

#### DBDC

DB/DC システム

DB DBCTL システム

DC DCCTL システム

### SUFFIX

MODBLKS データ・セット内のメンバーに関連した接尾部を表す 1 文字の値を指定します。このパラメーターは、「SUF」と省略できます。このパラメーターはオプションです。SUFFIX のデフォルト値は 0 です。

### EDITRTN

トランザクション ID とそれに関連した編集ルーチンのペアを指定します。値はペアで指定され、トランザクション ID と編集ルーチン名は、間にスペースを入れずにコンマで分離されます。このパラメーターはオプションです。EDITRTN 指定がないすべてのトランザクション ID では、関連する編集ルーチンは識別されません。

次の例は EDITRTN パラメーターを示しています。

```
EDITRTN=ICS,DFSCSMB0
```

ICS はトランザクション名で、DFSCSMB0 は編集ルーチンです。どちらも、ステージ 1 の TRANSACT マクロ上で以下のように指定されたものです。

```
TRANSACT CODE=ICS,PRTY=(5,12,5),PROCLIM=10,100), MODE=SNGL,EDIT=,DFSCSMB0)
```


## 戻りコード

以下の戻りコードが作成されます。

コード 意味

- 0 ユーティリティーが正常に完了しました。
- 8 ユーティリティーは強制終了エラーを検出しました。

関連概念:

 動的リソース定義の概要 (システム定義)

## DFSURCM0 ユーティリティーの例

これらの例では、DFSURCM0 ユーティリティーを実行するためのサンプル JCL と、要約報告書の例を示します。

### JCL の例

次のサンプル JCL ステートメントを使用して、このユーティリティーを実行できます。

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOBLIB DD DSN=[Library data set name],DISP=SHR
//S1 EXEC PGM=DFSURCM0
//MODBLKS DD DSN=[MODBLKS data set name],DISP=SHR
//RDDSDSN DD DSN=[RDDSDSN data set name],DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//SYSPRINT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//REPORT DD SYSOUT=*,
// DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//CONTROL DD *
IMSID=imid
SUF=x
/*
```

この例では、SDFSRESL DD ステートメントが指定されていないので、中核モジュール DFSVNUCV は JOBLIB DD ステートメントで指定されたデータ・セットから取得されます。

### 要約報告書の例

次の例では、サンプルの要約報告書を示し、報告書の解釈方法について説明します。

```
* *****
* RDDS BUILD FROM MODBLKS / DFSURCM0 DATE: 2011/125 TIME: 22:33 PAGE: 1
* *****
*
* CONTROL: IMSID=SYS3
* CONTROL: SYSTYPE=DBDC
* CONTROL: SUF=C
*
* RDDS DATA SET NAME:
* IMSTESTL.IMS1.RDDS1
*
* MODBLKS DATA SET NAME:
* IMSBLD.I12RTS1B.COMBLKS1
*
* IMSID USED . . . : SYS3
* SUFFIX USED . . . : C
* SYSTYPE USED . . . : DBDC
*
* TRAN MEMBER NAME. . . . . : DFSSMB0C
```

```

*      SIZE OF MEMBER . . . . . : 122176
*      SIZE OF MODBLKS BLOCK. . . : 184
*      SIZE OF CHKP BLOCK . . . . : 184
*      SIZE OF RSCX EXTENSION . . : 60
*      TOTAL NUMBER OF BLOCKS . . : 664
*      BLOCKS PER RDDS RECORD . . : 134
*
* DB  MEMBER NAME. . . . . : DFSDDIRC
*      SIZE OF MEMBER . . . . . : 139104
*      SIZE OF MODBLKS BLOCK. . . : 168
*      SIZE OF CHKP BLOCK . . . . : 51
*      SIZE OF RSCX EXTENSION . . : 60
*      TOTAL NUMBER OF BLOCKS . . : 828
*      BLOCKS PER RDDS RECORD . . : 64
*
* PGM MEMBER NAME. . . . . : DFSPDIRC
*      SIZE OF MEMBER . . . . . : 67200
*      SIZE OF MODBLKS BLOCK. . . : 112
*      SIZE OF CHKP BLOCK . . . . : 24
*      SIZE OF RSCX EXTENSION . . : 60
*      TOTAL NUMBER OF BLOCKS . . : 599
*      BLOCKS PER RDDS RECORD . . : 88
*
*      * BLOCK FOR DBF#FPU0 PDIR WAS REMOVED
*
* RTC MEMBER NAME. . . . . : DFSRCTEC
*      SIZE OF MEMBER . . . . . : 4272
*      SIZE OF MODBLKS BLOCK. . . : 48
*      SIZE OF CHKP BLOCK . . . . : 28
*      SIZE OF RSCX EXTENSION . . : 60
*      TOTAL NUMBER OF BLOCKS . . : 89
*      BLOCKS PER RDDS RECORD . . : 88
*
* RDDS BUILD TIME : 2011.126 05:33:40.605599-UTC
* RDDS BLKSIZE   : 32760
*
* *****
*      RDDS BUILD INFORMATION SUMMARY
*
* TOTAL # OF NON COMMENT CONTROL STATEMENTS READ . . : 3
* TOTAL # OF TRANSACT EDIT ROUTINE CONTROL STMTS . . : 0
* TOTAL # OF RDDS RECORDS WRITTEN. . . . . : 29
* TOTAL # OF TRAN ENTRIES IN MODBLKS . . . . . : 664
* TOTAL # OF TRAN RDDS RECORDS WRITTEN . . . . . : 5
* TOTAL # OF DB ENTRIES IN MODBLKS . . . . . : 828
* TOTAL # OF DB RDDS RECORDS WRITTEN . . . . . : 13
* TOTAL # OF PGM ENTRIES IN MODBLKS . . . . . : 599
* TOTAL # OF PGM RDDS RECORDS WRITTEN . . . . . : 7
* TOTAL # OF RTC ENTRIES IN MODBLKS . . . . . : 89
* TOTAL # OF RTC RDDS RECORDS WRITTEN . . . . . : 2
* *****

```

この例では、次のとおりです。

- ログに関連した IMSID は、制御ステートメント上で SYS3 として識別されました。
- MODBLKS データ・セット内のモジュールに関連した接尾部は、制御ステートメント上で C として指定されました。
- システム・タイプは、制御ステートメント上で DBDC として指定されました。
- 処理されたリソースに関連した MODBLKS メンバーごとに、以下の情報が提示されました。
  - 処理された MODBLKS メンバーに関連したリソース・タイプと名前
  - メンバーのサイズが以下のように示されました。
    - レコードのサイズ

- MODBLKS メンバー内の各リソースのブロックのサイズ
  - RDDS レコードに格納される各リソースのブロックのサイズ
  - RDDS レコード内の各リソースに関連したリソース拡張域のブロックのサイズ
  - RDDS データ・セット内にこのリソース・タイプ用に作成されるリソース・ブロックの合計数
  - このリソース・タイプ用に作成される RDDS レコード内のリソース・ブロックの最大数
- ユーティリティの処理に関連した一般統計が、以下のように提示されました。
    - 3 つの制御ステートメントが読み取られ、処理されました。
    - トランザクション編集ルーチンを識別する制御ステートメントは指定されませんでした。
    - MODBLKS データ・セットから処理された 4 つのリソース・タイプのそれぞれについて、以下の情報が提示されました。
      - 対応する MODBLKS メンバー内で検出された、指定されたリソース・タイプの項目の数
      - このリソース・タイプ用に書き込まれた RDDS レコードの数



---

## 第 37 章 DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティー (DFSURST0)

DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティー (DFSURST0) は、ステージ 1 ソース・コードに含まれているマクロから選択したパラメーターを再フォーマット設定または抽出するために使用します。

DFSURST0 ユーティリティーは、ステージ 1 の実行後、選択されたステージ 1 マクロ指定のパラメーターを変更することによってステージ 2 JCL を生成し、IMS サブシステムに関する情報の収集を自動化します。

ステージ 1 のステートメントの変更を指示するために、ユーティリティーへ制御ステートメントが提供されます。ステージ 1 のソースおよびコピー・メンバーが、ユーティリティーへの入力として読み取られます。それらの要素は結合され、処理され、出力ファイルへ書き込まれます。その後、出力ファイルがシステム生成プロセスへの入力として使用されます。

以下のマクロについてのステージ 1 ソース・コード指定は、このユーティリティーによって変更されます。

### IMSCTRL マクロ

IMSCTRL マクロの SYSTEM ステートメントは、以下のように処理されます。

1. SYSTEM ステートメントが修正され、実行される IMS 生成のタイプが、元の指定から MODBLKS に変更されます。元の指定は、ALL、BATCH、CTLBLKS、MODBLKS、MSVERIFY、NUCLEUS、ON-LINE のいずれかのタイプです。
2. IMS システムのタイプが抽出され、DFSURCM0 ユーティリティーへの入力として使用できるよう、制御ステートメントとしてフォーマット設定されます。IMS システムのタイプは、DB/DC、DBCTL、DCCTL のいずれかです。

ステージ 1 ソース・コードには、IMSCTRL マクロの 1 つのオカレンスがあります。

### IMSGEN マクロ

IMSGEN マクロは、以下のように処理されます。

- IMSGEN マクロ上の MODGEN ステートメントは、z/OS コンパイル時システム・マクロを含んでいるデータ・セットを指定するように変更されます。このデータ・セットは、ステージ 1 の正常な実行によって生成されるステージ 2 JCL のアSEMBル・ステップで、SYSLIB DD ステートメントに連結されます。
- IMSGEN マクロ上の OBJDSET ステートメントは、ステージ 1 の正常な実行によって生成されるステージ 2 JCL のアSEMBル・ステップで、SYSLIN DD ステートメントとして指定されるデータ・セットを指定するように変更されます。

- IMSGEN マクロ上の USERLIB ステートメントは、ステージ 1 の正常な実行によって生成されるステージ 2 JCL の MODBLKS リンク・エディット・ステップで、USERLIB DD ステートメントとして指定されるデータ・セットを指定するように変更されます。
- IMSGEN マクロ上の NODE ステートメントの第 2 パラメーターは、MODBLKS データ・セットの高位修飾子を指定するように変更されます。このデータ・セットには、ステージ 2 の正常な実行のリンク・エディットされた結果が保管されます。
- IMSGEN マクロ上の NODE ステートメントの第 3 パラメーターは、以下のデータ・セットの高位修飾子を指定するように変更されます。
  - ADFSMAC データ・セット。これには、ステージ 1 の正常な実行によって生成されるステージ 2 JCL のアSEMBル・ステップで、SYSLIB DD ステートメントに連結される IMS コンパイル時マクロが入っています。
  - ADFSLOAD データ・セット。これには、ステージ 1 の正常な実行によって生成されるステージ 2 JCL の MODBLKS バインド・ステップで、ADFSLOAD DD ステートメントによって識別される IMS バインド時ルーチンが入っています。
- IMS サブシステムの SUFFIX 値が IMSGEN マクロから抽出され、DFSURCM0 ユーティリティへの入力として使用される制御ステートメントとしてフォーマット設定されます。

ステージ 1 ソース・コードには、IMSGEN マクロの 1 つのオカレンスがあります。

#### TRANSACT マクロ

TRANSACT マクロは、以下のように処理されます。

- TRANSACT マクロ上の CODE および EDIT ステートメントを調べて、編集マクロの適用対象となるトランザクションのリストが判別されます。
- CODE ステートメントは、TRANSACT マクロのオカレンスに関連したトランザクション ID のリストを判別します。
- EDIT ステートメントは、CODE ステートメントで示されたトランザクション ID のリストに関連した編集ルーチンを判別します。
- ステージ 1 ソース・コードは、TRANSACT マクロの複数のオカレンスを持つことができます。TRANSACT マクロの 1 つの指定には、CODE 指定が 1 つだけ入っていますが、その指定に複数のトランザクション ID スキャンを関連付けることができます。1 つの TRANSACT マクロ指定には、EDIT ステートメントのオプションの指定が 1 つだけ存在します。

サブセクション:

- 727 ページの『制約事項』
- 727 ページの『前提条件』
- 727 ページの『要件』
- 727 ページの『推奨事項』
- 727 ページの『入力と出力』

- 728 ページの『JCL 仕様』
- 730 ページの『戻りコード』

## 制約事項

現在、DFSURST0 ユーティリティについて文書化されている制約事項はありません。

## 前提条件

現在、DFSURST0 ユーティリティについて文書化されている前提条件はありません。

## 要件

現在、DFSURST0 ユーティリティについて文書化されている要件はありません。

## 推奨事項

現在、DFSURST0 ユーティリティについて文書化されている推奨事項はありません。

## 入力と出力

DFSURST0 ユーティリティへの入力は、IMS サブシステムのステージ 1 マクロ・ソースが入っているデータ・セットです。このステージ 1 ソースは COPY メンバーを参照し、それらのメンバーが入っている区分データ・セット (PDS) も入力として使用されます。制御ステートメントは、CONTROL データ・セットから読み取られます。

DFSURST0 ユーティリティはステージ 1 ソースを読み取り、新しいバージョンのステージ 1 ソースを書き込みます。新しいソースには、選択されたステートメントの指定に対する変更が入っています。ステージ 1 ソースから抽出されたデータが入っている制御ステートメントが生成され、このデータをあとで他の DRD ユーティリティで使用できます。トランザクションと編集ルーチンの相互参照が作成され、後続の DFSURCM0 ユーティリティで使用されます。ユーティリティによって発行されたメッセージと要約報告書は、SYSPRINT データ・セットに書き込まれます。

## **OUTPARMS** 生成の制御ステートメント

以下の制御ステートメントが DFSURST0 によって生成され、DFSURCM0 ユーティリティで使用されます。

### **IMSID**

IMSID パラメーターは、ユーティリティによって生成される RDDDS データ・セットに関連付けられる IMS サブシステムの名前を指定します。このパラメーターに関連付けられる値は、IMS サブシステムに関連した 4 文字の ID です。

### **SYSTYPE**

SYSTYPE パラメーターは、このユーティリティによって生成される

RDDS データ・セットに関連付けられる IMS サブシステムのタイプを指定します。これは、1 文字から 4 文字までの値です。

有効な値は、以下のとおりです。

- DB - DBCTL システム
- DC - DCCTL システム
- DBDC - DB/DC システム

#### **SUFFIX**

SUFFIX パラメーターは、MODBLKS データ・セット内のメンバーに関連した接尾部を指定します。これは 1 文字の値です。

#### **EDITRTN**

EDITRTN パラメーターは、トランザクション ID とそれに関連した編集ルーチンのペアを指定します。値はペアで指定され、トランザクション ID と編集ルーチン名は、間にスペースを入れずにコマンドで分離されます。

#### **JCL 仕様**

DFSURST0 ユーティリティは、標準オペレーティング・システム・ジョブとして実行されます。JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

**EXEC** ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1 EXEC PGM=DFSURST0
```

**DD** ステートメント

#### **JOBLIB / STEPLIB DD**

ユーティリティの実行可能モジュールが入っている IMS.SDFSRESL データ・セットを指します。連結データ・セットを使用できます。

#### **STAGE1IN DD**

ユーティリティへの入力として使用されるステージ 1 ソース・マクロ・データ・セットを定義します。この DD ステートメントは必須です。ソースが PDS メンバー内にある場合、DD ステートメント上でメンバー名を含める必要があります。

#### **COPYFILE DD**

STAGE1IN DD ステートメントで識別されたステージ 1 ソース内に COPY ステートメントがコーディングされている場合、COPYFILE DD はステージ 1 ソース・マクロ・メンバーが入っている PDS を定義します。この DD ステートメントは、COPY ステートメントがコーディングされている場合にのみ必須です。

#### **OUTFILE DD**

ユーティリティによって生成された変更済みステージ 1 ソースを受け取るデータ・セットを定義します。この DD ステートメントは必須です。OUTFILE データ・セットの DCB パラメーターは、RECFM=FB、LRECL=80 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

### **OUTPARMS DD**

ユーティリティーによって生成された制御ステートメントを受け取るデータ・セットを定義します。この DD ステートメントは必須です。OUTPARMS データ・セットの DCB パラメーターは、RECFM=FB、LRECL=80 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

この DD ステートメントで参照されたデータ・セットは、DFSURCM0 ユーティリティー JCL ストリーム内で CONTROL DD ステートメントとして使用されます。

### **SYSPRINT DD**

ユーティリティーによって生成されたメッセージと要約報告書を受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FBA、LRECL=133 です。BLKSIZE は、この DD ステートメントで指定しなければなりません。

### **CONTROL DD**

ユーティリティーの入力パラメーターのソースを定義します。この DD ステートメントは必須です。CONTROL データ・セットの DCB パラメーターは、RECFM=FB、LRECL=80 です。

## **ユーティリティー制御ステートメント**

### **MODGEN**

ステージ 1 の正常な実行によって生成されるステージ 2 JCL のアセンブル・ステップで、SYSLIB DD ステートメントに連結される z/OS コンパイル時システム・マクロが入っているデータ・セットを指定します。値は、そのデータ・セットに関連した 44 文字の名前です。このパラメーターはオプションです。これを省略した場合は、MODGEN の既存のステージ 1 指定が使用されます。既存のステージ 1 指定が存在しない場合のデフォルトは、SYS1.MODGEN です。

### **USERLIB**

これに指定したデータ・セットは、ステージ 1 の正常な実行によって生成されるステージ 2 JCL の MODBLKS リンク・エディット・ステップで、USERLIB DD ステートメントを指定します。値は、そのデータ・セットに関連した 44 文字の名前です。このパラメーターはオプションです。これを省略した場合は、USERLIB の既存のステージ 1 指定が使用されます。

### **OBJDSET**

これに指定したデータ・セットは、ステージ 1 の正常な実行によって生成されるステージ 2 JCL の MODBLKS バインド・ステップで、OBJDSET DD ステートメントを指定します。値は、そのデータ・セットに関連した 44 文字の名前です。このパラメーターはオプションです。これを省略した場合は、NODE ステートメントの第 2 パラメーターに対する既存のステージ 1 指定が使用されます。

### **MODBLKSHLQ**

ステージ 2 の正常な実行のバインド結果を格納する MODBLKS データ・セットの高位修飾子を指定します。値は、そのデータ・セットに関連した 44 文字の名前です。これは、ユーティリティーによって生成されるステージ 1 ソース内の NODE ステートメントの第 3 パラメーターにマップされます。

重要: このパラメーターは、MODBLKS データ・セットが誤って上書きされないようにするために必要です。

### IMSHLQ

ステージ 1 の正常な実行によって生成されるステージ 2 JCL に組み込まれる IMS ADFSMAC および ADFSLOAD データ・セットの高位修飾子を指定します。値は、このデータ・セットに関連した最低位修飾子に先行する 44 文字の名前の一部です。これは、ユーティリティーによって生成されるステージ 1 ソース内の NODE ステートメントの第 3 パラメーターにマップされます。このパラメーターはオプションです。これを省略した場合は、NODE ステートメントの第 3 パラメーターに対する既存のステージ 1 指定が使用されます。

**ASM** ステージ 1 で IMSGEN マクロ上の ASM ステートメント (ステージ 2 アセンブリー・ステップ用に生成されるアセンブラー JCL を指定します) に関連した値を変更します。この値によって、ステージ 1 の ASM ステートメント指定に関連した値が置き換えられます。複数のパラメーターを指定する場合は、パラメーターを小括弧で囲み、コンマで分離する必要があります。このパラメーターはオプションです。これを省略した場合は、ASM ステートメントの既存のステージ 1 指定が使用されます。


### 戻りコード

以下の戻りコードが作成されます。

コード 意味

- 0 ユーティリティーが正常に完了しました。
- 8 ユーティリティーは強制終了エラーを検出しました。

関連概念:

 動的リソース定義の概要 (システム定義)

---

## DFSURST0 ユーティリティーの例

これらの例では、DFSURST0 ユーティリティーを実行するためのサンプル JCL と、要約報告書の例を示します。

### JCL の例

次のサンプル JCL を使用して、このユーティリティーを実行できます。

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOB LIB DD DSN=[library data set name],DISP=SHR
//S1 EXEC PGM=DFSURST0
//STAGE1IN DD DSN=[Stage 1 input data set name],DISP=SHR
//COPYFILE DD DSN=[Copy PDS data set name],DISP=SHR
//OUTFILE DD DSN=[Stage 1 output data setname],
// DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//OUTPARMS DD DSN=[OUTPARMS control statements data setname],
// DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=[Volume name],
// SPACE=(TRK,(1,1),RLSE),
```

```
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSPRINT DD  SYSOUT=*,
//          DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA)
//CONTROL DD *
      MODGEN=SYS1.MACLIB
      USERLIB=USERID.DFSURST0.USERLIB
      OBJDSET=USERID.DFSURST0.OBJDSET
      MODBLKSHLQ=USERID.DFSURST0
      IMSHLQ=IMSB LD.HMK1010
//*
//
```

この例では、ASM パラメーターが指定されていないので、ASM ステートメントの既存のステージ 1 指定が使用されます。

## 要約報告書の例

次の例では、サンプルの要約報告書を示し、報告書の解釈方法について説明します。

```
DFSURST0 CONTROL:    MODGEN=SYS1.MACLIB
DFSURST0 CONTROL:    USERLIB=USERID.DFSURST0.USERLIB
DFSURST0 CONTROL:    OBJDSET=USERID.DFSURST0.OBJDSET
DFSURST0 CONTROL:    MODBLKSHLQ=USERID.DFSURST0
DFSURST0 CONTROL:    IMSHLQ=IMSB LD.HMK1010
COPY MEMBER ARSALL   PROCESSED
ORIGINAL IMSCTRL: ALL
  RECORDS PROCESSED . . . : 13
COPY MEMBER SAMPDBD PROCESSED
  RECORDS PROCESSED . . . : 15
COPY MEMBER IVP#DBD PROCESSED
  RECORDS PROCESSED . . . : 8
COPY MEMBER GENIDENT PROCESSED
  RECORDS PROCESSED . . . : 8
DFSURST0: STAGE1 PRE-PARSER UTILITY: SUMMARY REPORT
          DATE: 2008/059 TIME: 16:46
INPUT LOG DATA SET NAME(S)
USERID.STAGE1(ARSS1 )
NUMBER OF CONTROL RECORDS READ . . . . : 5
NUMBER OF STAGE1 RECORDS READ . . . . : 1132
NUMBER OF COPY STATEMENTS READ . . . . : 4
NUMBER OF LINES FROM COPY MEMEBERS . . : 44
NUMBER OF EXCLUDED JCL STATEMENTS. . . : 0
NUMBER OF STAGE1 COMMENTS IGNORED. . . : 216
NUMBER OF STAGE1 RECORDS WRITTEN . . . : 1177
NUMBER OF IMSCTRL STATEMENTS . . . . . : 1
NUMBER OF TRANSACT STATEMENTS. . . . . : 7
NUMBER OF IMSGEN STATEMENTS. . . . . : 1
NUMBER OF OUTPARM RECORDS WRITTEN. . . : 3
```

この例では、次のとおりです。

- ユーティリティーに対して提供された制御ステートメントのイメージが表示されています。
- COPYFILE DD ステートメントで示されたファイルから読み取られた各 COPY メンバーの名前と、そのメンバーから読み取られたレコードの数が表示されています。
- 常にユーティリティーによって MODBLKS に変更されますが、IMSCTRL マクロ上の SYSTEM キーワードに関連した TYPE パラメーターの元の指定は、ALL です。

- ステージ 1 システム生成マクロが入っているデータ・セットの名前は、データ・セット USERID.STAGE 内のメンバー ARSS1 です。
- 5 つの制御ステートメントが読み取られ、処理されました。
- 識別されたステージ 1 データ・セット・メンバーから 1132 レコードが読み取られました。
- COPYFILE DD ステートメントで示されたファイルから、4 つの COPY メンバーが読み取られました。
- COPY メンバーから 44 レコードが読み取られました。
- ステージ 1 データ・セット・メンバーの内容は、ステージ 1 マクロだけを含んでいました。ソースに JCL は含まれていませんでした。ソースに JCL があり、例えば、ステージ 1 マクロが //SYSIN DD \*ステートメントの直後に埋め込まれていたとすれば、その JCL は、ユーティリティーが作成する変更されたステージ 1 ソースから除外されます。
- ステージ 1 ソース内で 216 のコメントが検出されました。これらは、処理では無視されましたが、ユーティリティーが作成するステージ 1 ソースに組み込まれます。
- 変更されたソースとして、1177 レコードが書き込まれました。
- 入力ステージ 1 ソース内で 1 つの IMSCTRL マクロ・ステートメントが検出されました。
- 入力ソース内で 7 つの TRANSACT マクロ・ステートメントが検出されました。
- 入力ソース内で 1 つの MSGEN マクロ・ステートメントが検出されました。
- あとで DFSURCM0 ユーティリティーの制御ステートメントとして使用される可能性がある 3 つのレコードが、OUTPARM DD ステートメントで識別されたファイルに書き込まれました。



---

## 第 38 章 RDDS 抽出ユーティリティ (DFSURDD0)

RDDS 抽出ユーティリティは、リソース定義データ・セット (RDDS) に保管されているリソース定義と記述子定義を、IMS ステージ 1 マクロ・ステートメントまたは IMS タイプ 2 CREATE コマンドに変換するときに使用します。RDDS 抽出ユーティリティは、RDDS の内容をリストする報告書を生成する場合にも使用できます。

RDDS 抽出ユーティリティは、オフラインのバッチ・ユーティリティです。

サブセクション:

- 『制約事項』
- 『前提条件』
- 『要件』
- 『推奨事項』
- 『入力と出力』
- 734 ページの『JCL 仕様』
- 735 ページの『ユーティリティ制御ステートメント』
- 735 ページの『戻りコード』

### 制約事項

現在、DFSURDD0 ユーティリティについて文書化されている制約事項はありません。

### 前提条件

現在、DFSURDD0 ユーティリティについて文書化されている前提条件はありません。

### 要件

現在、DFSURDD0 ユーティリティについて文書化されている要件はありません。

### 推奨事項

現在、DFSURDD0 ユーティリティについて文書化されている推奨事項はありません。

### 入力と出力

RDDS 抽出ユーティリティへの入力は、IMS のリソース定義と記述子定義が入っている RDDS データ・セットです。RDDS には、正常なエクスポート操作で取得されたか、またはいずれかの RDDS 作成ユーティリティで生成された、有効なデータが含まれている必要があります。

生成されたステージ 1 マクロ・ステートメントおよび CREATE コマンドは、SYSOUT データ・セットに書き込まれます。このユーティリティによって発行されたメッセージは、SYSPRINT データ・セットに書き込まれます。

リソース定義をステージ 1 マクロ・ステートメントに変換する場合は、RDDS 内のリソース定義を表す DATABASE、APPLCTN、RTCODE、および TRANSACT の各マクロ・ステートメントがこのユーティリティの出力になります。リソース記述子定義は、ステージ 1 マクロ・ステートメントに変換されません。

リソース定義を IMS タイプ 2 CREATE コマンドに変換する場合は、CREATE DB、CREATE DBDESC、CREATE PGM、CREATE PGMDESC、CREATE RTC、CREATE RTCDESC、CREATE TRAN、および CREATE TRANDESC の各コマンドが RDDS 抽出ユーティリティの出力になります。

APPLCTN マクロまたは CREATE PGM コマンドで LANG=ASSEM パラメーターを指定してプログラム定義を RDDS にエクスポートすると、このユーティリティによって必ず APPLCTN マクロ・ステートメントに LANG=COBOL が生成されるか、CREATE PGM コマンドに LANG(COBOL) が生成されます。ASSEM と COBOL の両プログラム言語は、IMS では同じように扱われます。

このユーティリティを使用すると、リソース定義と記述子定義をステージ 1 マクロまたはタイプ 2 CREATE コマンドに変換するほかに、RDDS の内容をリストする報告書も生成できます。SYSIN DD ステートメントで OUTPUT=QUERY 制御ステートメントを指定すると、ステージ 1 マクロ・ステートメントおよび CREATE コマンドが生成される際に組み込まれるフィールドに加えて、次のフィールドが追加で抽出されます。

- リソース定義または記述子定義が最後にアクセス、作成、更新、またはインポートされた時間
- RDDS ヘッダー情報 (RDDS データ・セット名、状況、タイプ、およびタイムスタンプの情報)

この報告書は SYSOUT データ・セットに書き込まれます。

## JCL 仕様

RDDS 抽出ユーティリティは、標準オペレーティング・システム・ジョブとして実行されます。JOB ステートメント、EXEC ステートメント、および DD ステートメント (入力と出力を定義するもの) を定義する必要があります。

### EXEC ステートメント

EXEC ステートメントの形式は次のとおりです。

```
//S1      EXEC PGM=DFSURDD0,MEMLIMIT=12G
```

RDDS 抽出ユーティリティは、64 ビット・ストレージを取得します。EXEC ステートメントに MEMLIMIT パラメーターを指定して、2 GB 境界より上に使用可能な仮想ストレージが十分に確保されるようにする必要があります。MEMLIMIT パラメーターは 12 GB 以上に設定してください。

### DD ステートメント

## **JOBLIB / STEPLIB DD**

ユーティリティの実行可能モジュールが入っている IMS.SDFSRESL データ・セットを指します。この DD ステートメントは必須です。連結データ・セットを使用できます。

## **RDDSDSN DD**

このユーティリティへの入力として使用される RDDS を定義します。この DD ステートメントは必須です。RDDS として指定できるデータ・セットは 1 つだけです。連結データ・セットは指定できません。

## **SYSOUT DD**

このユーティリティによって生成された出力を受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSOUT データ・セットの DCB パラメーターは、LRECL=80,RECFM=FB です。

SYSOUT には、IMS ステージ 1 マクロ・ステートメント、IMS タイプ 2 CREATE コマンド、または照会報告書のいずれかを指定できます。

## **SYSPRINT DD**

このユーティリティによって生成されたメッセージを受け取るデータ・セットを定義します。この DD ステートメントは必須です。SYSPRINT データ・セットの DCB パラメーターは、RECFM=FB,LRECL=133 です。

## **SYSIN DD**

ユーティリティの入力パラメーターのソースを定義します。この DD ステートメントは必須です。SYSIN データ・セットの DCB パラメーターは、RECFM=FB,LRECL=80 です。

## **ユーティリティ制御ステートメント**

### **出力**

OUTPUT パラメーターは、生成される出力の形式を指定します。このパラメーターは必須です。OUTPUT パラメーターの有効な値は次のとおりです。

### **MAC**

RDDSDSN 内のデータを IMS ステージ 1 マクロ・ステートメントに変換します。

### **CMD**

RDDSDSN 内のデータをタイプ 2 CREATE コマンドに変換します。

### **QUERY**

RDDSDSN 内のデータを報告書として生成します。

照会報告書は次の部分に分かれています。

- RDDSDSN ヘッダーの内容の表示
- RDDSDSN 内で記述されている各リソースの属性の表示
- 各リソース・タイプの項目数の合計
- 各リソース・タイプの重複項目数の合計


## **戻りコード**

以下の戻りコードが作成されます。

コード 意味

- 0 ユーティリティーが正常に完了しました。
- 4 ユーティリティーが正常に完了しましたが、アプリケーションにトランザクションが定義されていないか、または高速機能専用アプリケーションに宛先コードが定義されていません。
- 8 SYSPRINT データ・セットをオープンする際にエラーが発生しました。
- 12 SYSOUT データ・セットをオープンする際にエラーが発生しました。
- 16 SYSIN データ・セットを読み取る際にエラーが発生しました。
- 20 SYSIN データを解析する際にエラーが発生しました。
- 24 MVS GETMAIN マクロを使用してストレージを取得する際にエラーが発生しました。
- 28 64 ビット・ストレージからバッファを取得する際にエラーが発生しました。
- 32 RDJFCB 要求が RDDSDS データ・セットの DS 名 (データ・セット名) を返すことができませんでした。
- 36 RDDSDS に複数のデータ・セットが定義されています。
- 40 RDDSDSN DD ステートメントに定義されている RDDSDS データ・セットをオープンする際にエラーが発生しました。
- 44 RDDSDSN DD ステートメントに定義されている RDDSDS データ・セットを読み取る際にエラーが発生しました。
- 48 RDDSDSN DD ステートメントに定義されているデータ・セットが RDDSDS ではありません。
- 52 RDDSDSN DD ステートメントに定義されている RDDSDS に、正常なエクスポート操作で得られたデータが含まれていません。
- 56 BPE 限定機能サービス (LFS) を開始する際にエラーが発生しました。

関連概念:

 動的リソース定義の概要 (システム定義)

## DFSURDD0 ユーティリティーの例

以下の各例では、RDDSDS 抽出ユーティリティーを実行する 2 組のサンプル JCL ステートメントと照会報告書の例を示します。

### 出力が **MAC** に設定された **JCL** の例

次のサンプル JCL を使用すると、IMS ステージ 1 マクロ・ステートメントを出力に指定して、このユーティリティーを生成できます。

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOBLIB DD DSN=library data set name,DISP=SHR
//S1 EXEC PGM=DFSURDD0,MEMLIMIT=12G
//RDDSDSN DD DSN=RDDSDS data set name,DISP=SHR
//SYSOUT DD DSN=output data set name,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=Volume name,
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
```

```
//SYSIN DD *
OUTPUT=MAC
/*
//
```

## 出力が **QUERY** に設定された **JCL** の例

次のサンプル JCL を使用すると、照会報告書を出力に指定して、このユーティリティーを生成できます。

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOB LIB DD DSN=library data set name,DISP=SHR
//S1 EXEC PGM=DFSURDD0,MEMLIMIT=12G
//RDDSDSN DD DSN=RDDS data set name,DISP=SHR
//SYSOUT DD DSN=output data set name,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=Volume name,
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
OUTPUT=QUERY
/*
//
```

## 照会報告書の例

次に示す照会報告書の例は、SYSIN DD ステートメントで OUTPUT=QUERY が指定された場合に生成されます。

```
RDDS HEADER RECORD +
HEADER_LENGTH(168) VERSION(1) STATUS(GOOD) +
IMSID(SYS3) IMSTYPE(DBDC) SYSTEM_RDDS?(Y) +
TIMESTAMP(2008.058 21:50:07.695470-UTC) +
data set_NAME(USERID.TEST.RDDS2 )
DB NAME(AUTODB) ACCTYPE(UPD) RESIDENT(N) GLOBAL DMB(0000) +
LOCAL DMB(0001) MODELNAME() MODELTYPE() TMC(2007.311 16:18:42.49-UTC) +
TMAC() TMUP() TIMP()
DB NAME(AUTODBH) ACCTYPE(UPD) RESIDENT(N) GLOBAL DMB(0000) +
LOCAL DMB(0002) MODELNAME() MODELTYPE() TMC(2007.311 16:18:42.49-UTC) +
TMAC() TMUP() TIMP()
DB NAME(BANKATMS) ACCTYPE(EXCL) RESIDENT(N) GLOBAL DMB(0000) +
LOCAL DMB(0003) MODELNAME() MODELTYPE() TMC(2007.311 16:18:42.49-UTC) +
TMAC() TMUP() TIMP()
PGM NAME(EMHPSB2) BMPTYPE(N) DOPT(N) FP(E)GPSB(N) +
RESIDENT(N) SCHDTYPE(PARALLEL) TRANSTAT(N) MODELNAME() +
MODELTYPE() TMC(2008.354 22:17:41.80-UTC) TMAC() +
TMUP() TIMP()
TRAN NAME(EMHTX2) AOCMD(N) CLASS(1) CMTMODE(SNGL) +
CONV(N) DCLWA(Y) DIRROUTE(N) EDITUC(Y) FP(E) INQ(N) +
LPRI(1) MAXRGN(0) MSGTYPE(SNGLSEG) NPRI(1) PARLIM(65535) +
PGM(EMHPSB2) PLCT(65535) PLCTTIME(6553500) RECOVER(Y) +
REMOTE(N) RESP(Y) SEGNO(0) SEGSZ(0) SERIAL(N) +
TRANSTAT(N) WFI(N) MODELNAME() MODELTYPE() TMC(2008.354 22:17:42.74-UTC) +
TMAC() TMUP() TIMP()
TRAN NAME(EMHTX3) AOCMD(N) CLASS(1) CMTMODE(SNGL) +
CONV(N) DCLWA(Y) DIRROUTE(N) EDITUC(Y) FP(E) INQ(N) +
LPRI(1) MAXRGN(0) MSGTYPE(SNGLSEG) NPRI(1) PARLIM(65535) +
PGM(EMHPSB2) PLCT(65535) PLCTTIME(6553500) RECOVER(Y) +
REMOTE(N) RESP(Y) SEGNO(0) SEGSZ(0) SERIAL(N) +
TRANSTAT(N) WFI(N) MODELNAME() MODELTYPE() TMC(2008.354 22:17:42.74-UTC) +
TMAC() TMUP() TIMP()
* RTC NAME(EMHTX2) INQ(N) PGM(EMHPSB2) MODELNAME() MODELTYPE() +
TMC(2008.354 22:17:42.73-UTC) TMAC() TMUP() +
TIMP()
RTC NAME(EMHTX22) INQ(N) PGM(EMHPSB2) MODELNAME() +
MODELTYPE() TMC(2008.354 22:17:42.73-UTC) TMAC() +
TMUP() TIMP()
* RTC NAME(EMHTX3) INQ(N) PGM(EMHPSB2) MODELNAME() MODELTYPE() +
```

```

      TMC(2008.354 22:17:42.73-UTC) TMC() TMUP() +
      TIMP()
RTC NAME(EMHTX32) INQ(N) PGM(EMHPSB2) MODELNAME() +
MODELTYPE() TMC(2008.354 22:17:42.73-UTC) TMC() +
TMUP() TIMP()
TRAN NAME(TSTAD2R2) AOCMD(N) CLASS(1) CMTMODE(MULT) +
CONV(N) DCLWA(Y) DIRROUTE(N) EDITUC(Y) INQ(N) +
LCT(65535) LPRI(1) MAXRGN(0) MSGTYPE(MULTSEG) +
NPRI(1) PARLIM(65535) PGM(AD2TP) PLCT(65535) +
PLCTTIME(6553500) RECOVER(Y) REMOTE(N) RESP(Y) +
SEGNO(0) SEGSZ(0) SERIAL(N) TRANSTAT(N) WFI(N) +
MODELNAME() MODELTYPE() TMC(2007.319 18:50:47.98-UTC) +
TMC() TMUP() TIMP(2008.012 01:13:32.20-UTC)
PGM NAME(APOL1) BMPTYPE(N) DOPT(Y) FP(N)GPSB(N) RESIDENT(N) +
SCHDTYPE(SERIAL) TRANSTAT(N) MODELNAME() MODELTYPE() +
TMC(2007.319 18:50:47.28-UTC) TMC() TMUP(2008.002 22:50:56.69-UTC) +
TIMP(2008.012 01:13:32.20-UTC)
PGM NAME(FPPSB02) BMPTYPE(N) DOPT(N) FP(E)GPSB(N) +
RESIDENT(Y) SCHDTYPE(PARALLEL) TRANSTAT(N) MODELNAME() +
MODELTYPE() TMC(2007.311 16:18:42.48-UTC) TMC() +
TMUP() TIMP()
**** SUMMARY ****
TRAN      COUNT :      0
TRANDESC  COUNT :      0
DB        COUNT :    2045
DBDESC    COUNT :      0
PGM       COUNT :    5506
PGMDESC   COUNT :      0
RTC       COUNT :      0
RTCDESC   COUNT :      0

TRAN      DUPLICATES: 0
TRANDESC  DUPLICATES: 0
DB        DUPLICATES: 0
DBDESC    DUPLICATES: 0
PGM       DUPLICATES: 0
PGMDESC   DUPLICATES: 0
RTC       DUPLICATES: 0
RTCDESC   DUPLICATES: 0

```

この照会報告書の例では、以下のとおりです。

- 各リソース定義および各リソース記述子タイプに対して、次のカウントが表示されます。
  - RDDS に含まれているリソースの数
  - RDDS 内に同じリソース名を持つ別の項目があるリソース項目の数
- 報告書には、次の 2 種類の宛先コード・タイプのレコードが取り込まれます。
  - ステージ 1 入力または CREATE コマンドに明示的に組み込まれた結果の宛先コード
  - 高速機能専用トランザクションに関連付けられている宛先コード。報告書では、このタイプの宛先コードの先頭にアスタリスク (\*) が付加されます。

---

## 第 7 部 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。本書の他言語版を IBM から入手できる場合があります。ただし、ご利用にはその言語版の製品もしくは製品のコピーを所有していることが必要な場合があります。

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。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

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、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*IBM Director of Licensing*  
*IBM Corporation*  
*North Castle Drive, MD-NC119*  
*Armonk, NY 10504-1785*  
*US*

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せんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

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。これらの名前はすべて架空のものであり、類似する個人や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。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

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) (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\_.

---

## プログラミング・インターフェース情報

本書では、IMS によって提供されるプロダクト・センシティブ・プログラミング・インターフェースとそれに関連する情報を記述しています。

プロダクト・センシティブ・プログラミング・インターフェースにより、お客様のインストール済み環境で、このソフトウェア製品の診断、修正、モニター、修復、調整、またはチューニングなどの作業を実行することができます。これらのインターフェースを使用すると、IBM のソフトウェア製品の詳細設計や実装に対する依存関係が生じます。このためプロダクト・センシティブ・プログラミング・インターフェースは上記の特別な目的にだけ使用してください。詳細設計やその実現方法に依存しているので、このようなインターフェースに合わせて作成したプログラムは、新しい製品のリリース、バージョンで実行するとき、または保守サービスの結果として、変更が必要になることがあります。プロダクト・センシティブ・プログラミング・インターフェースとそれに関連する情報は、セクションやトピックの単位の場合はその冒頭で識別され、それ以外の場合は「プロダクト・センシティブ・プログラミング・インターフェース」というマーケティングで識別されます。IBM では、上記の冒頭部での識別の記述、およびその記述を参照する本書内のすべての記述を、そのような記述によって示される全体コピーまたは部分コピーに含めるよう求めています。

---

## 商標

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com)<sup>®</sup> は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

---

## 製品資料に関するご使用条件

これらの資料は、以下のご使用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。

### 適用される条件

このご使用条件は、IBM Web サイトのすべてのご利用条件に追加して適用されます。

### 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。

### 権利

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。

---

## IBM オンライン・プライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。

IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

この「ソフトウェア・オファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/privacy/details>) を参照してください。



---

## 参考文献

この参考文献のリストには、IMS 15 ライブラリーのすべての資料が記載されています。

表題	頭字語	資料番号
IMS V15 アプリケーション・プログラミング	APG	SC27-6778
IMS V15 アプリケーション・プログラミング API	APR	SC27-6779
IMS V15 コマンド 第 1 巻: IMS コマンド A-M	CR1	SC27-6780
IMS V15 コマンド 第 2 巻: IMS コマンド N-V	CR2	SC27-6781
IMS V15 コマンド 第 3 巻: IMS コンポーネント および z/OS コマンド	CR3	SC27-6782
IMS V15 コミュニケーションおよびコネクション	CCG	SC27-6783
IMS V15 データベース管理	DAG	SC27-6784
IMS V15 データベース・ユーティリティー	DUR	SC27-6785
IMS Version 15 Diagnosis	DGR	GC27-6786
IMS V15 出口ルーチン	ERR	SC27-6787SC43- 3856
IMS V15 インストール	INS	SC27-6788
IMS Version 15 Licensed Program Specifications	LPS	GC27-6799
IMS V15 メッセージおよびコード 第 1 巻: DFS メッセージ	MC1	GC27-6789
IMS V15 メッセージおよびコード 第 2 巻: DFS 以外メッセージ	MC2	GC27-6790
IMS V15 メッセージおよびコード 第 3 巻: IMS 異常終了コード	MC3	GC27-6791
IMS V15 メッセージおよびコード 第 4 巻: IMS コンポーネント・コード	MC4	GC27-6792
IMS V15 オペレーションおよびオートメーション	OAG	SC27-6793
IMS V15 リリース計画	RPG	GC27-6794
IMS V15 システム管理	SAG	SC27-6795
IMS V15 システム定義	SDG	GC27-6796
IMS V15 システム・プログラミング API	SPR	SC27-6797
IMS V15 システム・ユーティリティー	SUR	SC27-6798





# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アクセシビリティ  
キーボード・ショートカット ix  
機能 ix  
アクティブ領域メッセージ  
ログ・リカバリー・ユーティリティ  
(DFSULTR0) 593  
アプリケーション制御ブロック (ACB)  
推奨事項 5  
制約事項 4  
前提条件 5  
保守  
概要 3  
制御ステートメント 9, 375  
入力と出力 6  
要件 5  
ACBLIB ライブラリー 3  
アプリケーション制御ブロック保守ユーティリティ 3  
アプリケーション・プログラム  
メタデータ  
ACB Generation and Catalog  
Populate ユーティリティ  
(DFS3UACB) 367  
IMS カタログ、ロード 425  
IMS カタログへのロード 425  
BMP  
データベース非コミット更新の制限  
307  
位置調整  
指定 219  
印刷ページのフォーマット制御  
下部マージン 241  
行密度 241, 242  
上部マージン 241  
エラー・ブロックのリスト (SYSPRINT)  
フィールドの説明 590  
オフライン・ダンプ・フォーマッター・ユーティリティ (DFSOFMD0)  
概要 537  
環境  
DB バッチ 540  
DBCTL 539  
DB/DC 539

オフライン・ダンプ・フォーマッター・ユーティリティ (DFSOFMD0) (続き)  
環境 (続き)  
DCCTL 539  
TM  
バッチ 540  
推奨事項 539  
制約事項 538  
前提条件 538  
ダンプ形式制御データ・セット  
サブセット・オプション 541  
説明 541  
DD ステートメント 541  
ダンプ・フォーマッター 537  
入力と出力 539  
マイグレーションの考慮事項 537  
要件 538  
ロード・モジュール 537  
IPCS 540  
SDUMP 539  
オペレーター制御テーブル  
作成に使用される言語ステートメント  
IF 283  
TABLE 283  
TABLEEND 285  
オンライン変更コピー・ユーティリティ (DFSUOCU0)  
アクティブ・ライブラリー (active library) 667  
概要 665  
使用されるライブラリー 667  
推奨事項 667  
ステージング・ライブラリー 667  
制約事項 666  
前提条件 666  
取り消し 666  
非アクティブ・ライブラリー (inactive library) 667  
プロシージャー・ステートメント 667, 674  
要件 667  
DD ステートメント 670  
DFSMBREC 制御ステートメント 674  
EXEC ステートメント 668  
INITMOD プロシージャー 674  
JCL 673  
MSDB 666  
OLCUTL プロシージャー 672  
オンライン・データベース・イメージ・コピー・ユーティリティ (DFSUICP0)  
必要な PSBGEN の指定 363

## [カ行]

拡張選択出口ルーチン (DFSERA70)  
概要 521  
例 523  
カタログ (catalog)  
コピー 389, 399  
単一ステップでの ACBGEN とデータの取り込み 367  
データの取り込み 425  
定義 387, 421  
ユーティリティ  
DFS3UCD0 387, 421  
ロード 425  
ACB Generation and Catalog  
Populate ユーティリティ  
(DFS3UACB) 367  
ACBGEN 中のデータの取り込み 367  
ACBGEN 中のデータのロード 367  
DBRC、使用しない 387, 421  
IMS カタログ・ユーティリティ 365  
カタログ・データベース  
セグメント、削除 451  
レコードの除去 451  
画面フォーマットの設定  
画面サイズの指定 229  
キーボード・ショートカット ix  
強制 EOJ  
ログ保存ユーティリティ 566  
区画  
データベース非コミット更新の制限  
307  
区画セットの作成に使用される言語ステートメント  
PD 281  
PDB 280  
PDBEND 283  
区分階層直接アクセス方式  
(PHDAM) 171  
区分化階層索引直接アクセス方式  
(PHIDAM) 171  
グローバル・オンライン変更ユーティリティ (DFSUOLCO)  
概要 627  
推奨事項 629  
制約事項 629  
前提条件 629  
パラメーター 629  
要件 629  
例 633  
JCL 629

グローバル・オンライン変更ユーティリティー (DFSUOLC0) (続き)  
  OLCSTAT データ・セット 627  
形式セット 195  
検査処理  
  複数システム検査ユーティリティー (DFSUMSV0) 653  
高可用性ラージ・データベース (HALDB) 区画  
  データベース非コミット更新の制限 307  
高速機能  
  AREA ステートメント  
    キーワード 77  
    形式 61  
    説明 77  
  DEDB の DBD 生成  
    説明 29  
  MSDB PSB 生成  
    例 347  
  MSDB の DBD 生成  
    説明 28  
高速機能 DEDB  
  高速処理データベース (DEDB)  
    DBD 生成 36  
高速機能 (Fast Path)  
  ログ分析ユーティリティー (DBFULTA0)  
    ログ間隔 467  
  AREA ステートメント  
    DEDB の DBD 生成のレコード 33  
  DEDB PSB 生成  
    代替 PCB ステートメント 311  
  DEDB の DBD 生成  
    入力レコード構造 33  
    例 173  
  MSDB PSB 生成  
    代替 PCB ステートメント 311  
  MSDB の DBD 生成  
    例 172  
高速機能 MSDB  
  主記憶データベース (MSDB)  
    DBD 生成 36  
  DEDB (高速処理データベース)  
    DBD 生成 36  
高速機能ログ分析ユーティリティー (DBFULTA0)  
  エラー処理 476  
  概要 467  
  高速機能報告書タイプ 476  
  推奨事項 469  
  制約事項 468  
  前提条件 469  
  入力と出力 469  
  報告書  
    通過時間総合要約 484

高速機能ログ分析ユーティリティー (DBFULTA0) (続き)  
  報告書 (続き)  
    トランザクション・コード別例外詳細要約報告書 483  
  分析の総括 488  
  リソースの使用量と競合の総合要約 484  
  領域占有率の要約 486  
  例外トランザクションの詳細リスト 477  
  VSO アクティビティーの要約 487  
ユーティリティー制御ステートメント 471  
  要件 469  
  JCL の要件 470  
高速処理データベース (DEDB) (data entry database (DEDB))  
  DBD 生成 36  
構文図  
  読み方 viii  
コピー機能  
  リモート端末 270  
コンパイル・ステートメント  
  ステートメントの要約 207  
COPY 285  
EJECT 291  
END 291  
EQU 286  
EQU 処理 286  
PRINT 290  
RESCAN 288  
RESCAN ステートメント 288  
SPACE 290  
STACK ステートメント 289  
SYSIN 202  
SYSLIB 202  
SYSPRINT 202  
TITLE 290  
UNSTACK 289

## [サ行]

サービス・ユーティリティー  
  オンライン変更コピー・ユーティリティー (DFSUOCU0) 607  
  グローバル・オンライン変更ユーティリティー (DFSUOLC0) 607  
  時間制御操作検査ユーティリティー (DFSTVER0) 607  
  スプール SYSOUT 印刷ユーティリティー (DFSUPRT0) 607  
  データベース・リカバリー管理ユーティリティー (DSPURX00) 607  
  動的 SVC ユーティリティー (DFSUSVC0) 607

サービス・ユーティリティー (続き)  
  バッチ SPOC ユーティリティー (CSLUSPOC) 607  
  複数システム検査ユーティリティー (DFSUMSV0) 607  
  リソース定義データ・セット抽出ユーティリティー (DFSURDD0) 607  
  MFS サービス・ユーティリティー (DFSUTSA0) 607  
索引データベース  
  定義  
    DFSCASE ステートメント 161  
    DFSMAP ステートメント 157  
    DFSmarsh ステートメント 150  
  DBD (データベース記述) 生成 36  
  DFSCASE ステートメント 161  
  DFSMAP ステートメント 157  
  DFSmarsh ステートメント 150  
索引の DBD 生成  
  概要 29  
  副次索引 30  
  論理 DBD 31  
  HIDAM 1 次索引 29  
時間スケジュール要求テーブル  
  時間制御操作検査ユーティリティー (DFSTVER0) 684  
時間制御操作検査ユーティリティー (DFSTVER0)  
  出力  
    エラー報告書 684  
    時間スケジュール要求報告書 (タイマー・エレメント報告書) 684  
    説明 682  
    統計報告書 684  
    メッセージ・テーブル報告書 685  
    要約報告書 686  
  推奨事項 681  
  制約事項 681  
  説明 681  
  前提条件 681  
  戻りコード 683  
  要件 681  
  EXEC ステートメント 682  
  JCL 仕様 682  
システム・チェックポイント  
  データベース非コミット更新の制限 307  
システム・リテラル  
  その他のフォーマット、CA パラメーター (MFLD ステートメント) 217  
自動化操作プログラム・インターフェース (AOI) 340  
充てん文字  
  出力装置フィールド  
    指定 209

充てん文字 (続き)  
   入力メッセージ・フィールド  
   指定 220  
 主記憶データベース (MSDB) (main  
 storage database (MSDB))  
   DBD 生成 36  
 出力メッセージ  
   ヘッダー  
   構造と内容 247  
 商標 741, 743  
 省略、SLDS のログ・レコードの  
   ログ保存ユーティリティ 565  
 スクリプト・メンバー、エラー 681  
 スプール SYSOUT 印刷ユーティリティ  
 (DFSUPRT0)  
   システム・メッセージ 677  
   出力例 677  
   推奨事項 677  
   制約事項 677  
   説明 677  
   前提条件 677  
   ブロック化因数 677  
   要件 677  
   DD ステートメント 679  
   DFSWTnnn プロシージャ 678  
   EXEC ステートメント 679  
   IMSWTnnn プロシージャ 680  
   PROC ステートメント  
   説明 679  
 すべてのトランザクション・コードおよび  
   PSB についてのリソースの使用量と競合  
   の総合要約報告書  
   高速機能ログ分析ユーティリティ  
   484  
 制御ステートメントのリスト  
   アセンブラー・リスト 21  
   診断 21  
 制御ブロック  
   作成、MFS 言語ユーティリティに  
   よる 294  
   MFS 言語ユーティリティ 195  
 生成ユーティリティ  
   アプリケーション制御ブロック保守ユ  
   ーティリティ 1  
   データベース記述 (DBD) 生成ユーテ  
   ィリティ 1  
   プログラム仕様ブロック (PSB) 生成ユ  
   ーティリティ 1  
   MFS 言語ユーティリティ  
   (DFSUPAA0) 1  
   MFS 装置特性テーブル・ユーティリ  
   ティ (DFSUTB00) 1

選択  
   リカバリー・トークンごとのログ・レ  
   コード  
   すべてのレコード・タイプを選択す  
   る例 524  
   特定レコード・タイプを選択する例  
   524  
   TOKEN= サブパラメーター 522  
   X'50' ログ・レコードの拡張ログ・フ  
   ォーマット設定  
   XFMT= サブパラメーター 521  
 選択ペン、3270  
   フィールド検出可能性の指定 270  
 装置特性テーブル 229  
 属性シミュレーション  
   指定 269  
 属性データ  
   出力装置フィールド  
   指定 269  
   ATTR= オペランド (MFLD ステ  
   ートメント) 219  
   入力メッセージ・フィールド  
   ATTR= オペランド (MFLD ステ  
   ートメント) 219

## [タ行]

代替 PCB ステートメント、PSB 生成  
 311  
 タブ  
   フィールド・タブ 232  
 単一ログの入力  
   ログ・リカバリー・ユーティリティ  
   587  
 ダンプ形式制御データ・セット  
   サブセット・オプション 541  
   説明 541  
   DD ステートメント 541  
 チェックポイント  
   システム  
   データベース非コミット更新の制限  
   307  
 データベース  
   メタデータ  
   ACB Generation and Catalog  
   Populate ユーティリティ  
   (DFS3UACB) 367  
   IMS カタログ、ロード 425  
   IMS カタログへのロード 425  
   IMS カタログ  
   コピー 389, 399  
 データベース記述 (DBD) 19  
   生成  
   索引データベースと PSINDEX デ  
   ータベース 29, 30  
   論理セグメント・タイプ 31

データベース記述 (DBD) (続き)  
   生成 (続き)  
   DEDB データベース 29  
   GSAM データベース 25  
   HDAM データベース 27  
   HDAM データベースと PHDAM  
   データベース 27  
   HIDAM データベースと PHIDAM  
   データベース 28  
   HISAM データベース 26, 27  
   HSAM データベース 25  
   MSDB 28  
   PHDAM データベース 27  
   SHISAM データベース 26  
   SHSAM データベース 25  
 データベース記述規則、DBD 生成 34  
 データベース記述生成ユーティリティ  
   制御ステートメント、入力順序 31  
 データベース・リカバリー管理ユーティリ  
 ティ (DSPURX00)  
   推奨事項 615  
   制約事項 615  
   説明 615  
   前提条件 615  
   ユーティリティの呼び出し 619  
   要件 615  
   例 618  
 データ・キャプチャー出口ルーチン  
   EXIT= パラメーター 52  
 デッドロック報告書  
   ロック結果の状態 514  
   U777 および U123 異常終了 509  
 統計分析ユーティリティ (DFSISTS0)  
   概要 545  
   作成される報告書の説明と例  
   アプリケーション会計報告書 549,  
   560  
   回線および端末報告書 557  
   キューイングされたが送信されな  
   かったメッセージ (宛先別) 547,  
   556  
   キューイングされたが送信されな  
   かったメッセージ (トランザクシ  
   ョン・コード別) 558  
   トランザクション報告書 559  
   プログラム間メッセージ (宛先別)  
   556  
   プログラム間メッセージ (トランザ  
   クション・コード別) 558  
   メッセージ選択/コピー (リスト)  
   (DF) によって生成されるメッセ  
   ージ 552, 561  
   IMS 会計報告書 550, 560  
   出力 545  
   推奨事項 545  
   制御ステートメント 552

統計分析ユーティリティ (DFSISTS0)  
(続き)  
 制約事項 545  
 説明 545  
 前提条件 545  
 入力 545  
 プログラム・モジュール  
 メッセージ選択/コピー (リスト)  
 (DFSIST40) 550  
 ユーティリティ制御ステートメント  
 印刷不能文字 554  
 時間 554  
 シンボリック端末名 553  
 説明 552  
 トランザクション・コード 553  
 要件 545  
 例 555  
 JCL の要件  
 ジョブ・ストリーム の例 555  
 説明 550  
 動的 SVC ユーティリティ  
 (DFSUSVC0)  
 エラー処理 625  
 概要 623  
 出力 624  
 推奨事項 624  
 制約事項 623  
 前提条件 623  
 入力 624  
 戻りコード 625  
 要件 624  
 例 625  
 DD ステートメント 625  
 JCL の要件 624  
 動的リソース定義 (DRD) (dynamic  
 resource definition (DRD))  
 ユーティリティ 689  
 特記事項  
 商標 741, 743  
 特記事項 741  
 トランザクション応答報告書  
 統計分析ユーティリティ  
 (DFSISTS0)  
 報告書 549  
 トランザクション・コード属性  
 (MSC) 654

## [ハ行]

バージョン ID  
 指定 242  
 パスワード、IMS  
 指定 267  
 バックアップ操作、MFS ライブラリー  
 303

バッチ SPOC ユーティリティ  
 (CSLUSPOC)  
 推奨事項 609  
 制約事項 609  
 説明 609  
 前提条件 609  
 入力と出力 610  
 パラメーター・キーワード 611  
 戻りコード 612  
 要件 609  
 例 612  
 出力例 612  
 待ち時間を指定しない出力の例 613  
 JCL の例 612  
 EXEC ステートメント 611  
 JCL 仕様 611  
 バッチ・メッセージ処理 (BMP)  
 アプリケーション・プログラム  
 データベース非コミット更新の制限  
 307  
 バッチ・モード (MFS 言語ユーティリ  
 ティ) 297  
 反復処理 (MFLD/DFLD)  
 DO ステートメント 215, 260  
 ENDDO ステートメント 221, 279  
 PRINT GEN の働き 290  
 RCD ステートメントと DFLD 263  
 非メッセージ・ドリブン・オプション  
 高速機能ログ分析ユーティリティ  
 474  
 ファイル選択およびフォーマット設定印刷  
 ユーティリティ (DFSERA10)  
 オプションのキーワード  
 OFFSET= 497  
 概要 491  
 拡張選択出口ルーチン  
 (DFSERA70) 521  
 プログラム分離 (PI) トレース・レコ  
 ード・フォーマット設定・印刷モジ  
 ュール (DFSERA40)  
 概要 517  
 例 501  
 レコード・フォーマット設定・印刷モ  
 ジュール (DFSERA30)  
 説明 508  
 DL/I 呼び出しイメージ・キャプチャ  
 ー・モジュール (DFSERA50) 520  
 IMS トレース・テーブル・レコード・  
 フォーマット設定・印刷モジュール  
 (DFSERA60) 521  
 JCL の要件  
 例 501  
 OPTION ステートメント  
 PARM= パラメーターのサブパラ  
 メーター 521

ファイル選択フォーマット設定印刷ユー  
 ティリティ (DFSERA10)  
 オプションのキーワード 497  
 B= 498  
 COND= 499  
 C= 499  
 DDNAME= 500  
 D= 500  
 EXITR= 499  
 E= 499  
 FLDLEN= 498  
 FLDTYP= 498  
 H= 498  
 L= 498  
 O= 497  
 PARM= 497  
 PRSYS= 500  
 P= 500  
 STARTAF= 498  
 STOPAFT= 498  
 SYM= 497  
 T= 498  
 VALUE= 498  
 V= 498  
 出力 492  
 推奨事項 492  
 制御ステートメント  
 説明 494  
 COMMENT 501  
 CONTROL 494  
 END 500  
 OPTION 496  
 制約事項 491  
 前提条件 491  
 入力 492  
 プログラム分離 (PI) トレース・レコ  
 ード・フォーマット設定・印刷モジ  
 ュール (DFSERA40)  
 サンプル 517  
 制御ステートメント 520  
 要件 491  
 レコード・フォーマット設定・印刷モ  
 ジュール (DFSERA30)  
 制御ステートメント 508  
 COPY オプション 497  
 JCL の要件  
 説明 492  
 例 506  
 DD ステートメント 492  
 NEGOF オプション 497  
 PRINT オプション 497  
 フィールド・タブ 232  
 指定 232  
 復元操作、MFS ライブラリー 305  
 副次索引  
 DBD 生成 30

複数システム検査ユーティリティ  
(DFSUMSV0)  
 検査処理 653  
 出力メッセージ 656  
 処理フェーズ 653  
 推奨事項 652  
 制約事項 651  
 説明 651  
 前提条件 652  
 トランザクション・コード属性  
 システム相互間での整合性 654  
 説明 654  
 リモート 654  
 ローカル 654  
 入力の妥当性検査 653  
 パートナー ID 653  
 物理リンク 653  
 プロシージャ、実行 662  
 プロシージャの呼び出し 663  
 マルチシステム制御ブロックの検査  
 653  
 マルチシステム・パス・マップ 656  
 ユーティリティ制御ステートメント  
 663  
 説明 663  
 要件 652  
 論理端末 651, 655  
 論理リンク 653  
 EXEC ステートメント 662  
 IMSMSV プロシージャ 662  
 MSC (複数システム結合機能) 651  
 PROC ステートメント  
 説明 662  
 SYSID パス  
 説明 654  
 リモート 654  
 ローカル 654  
 物理リンク、MSC 653  
 プログラム出力  
 ログ保存ユーティリティ 568  
 プログラム仕様ブロック (PSB) 生成ユー  
 ティリティ  
 高速機能データベース  
 処理オプション 326  
 プログラム制御ブロック (PCB)  
 DB  
 データベース非コミット更新の制限  
 307  
 プログラム分離 (PI) トレース・レコー  
 ド・フォーマット設定・印刷モジュール  
 (DFSERA40)  
 出力例 517  
 制御ステートメント 520

プログラム分離トレース・レコード・フォ  
 ーマット設定・印刷モジュール  
 (DFSERA40)  
 概要 517  
 デッドロック報告書 509  
 プログラム・タブ機能  
 充てん文字 (fill character) 209  
 プロシージャ  
 ACBGEN 6  
 DBDGEN 186  
 INITMOD 674  
 OLCUTL 672  
 PSBGEN 362  
 分割、データベースを複数のデータ・セッ  
 ト・グループに  
 DBD 生成 59  
 分析の総括報告書  
 高速機能ログ分析ユーティリティ  
 488  
 分析ユーティリティおよび報告書  
 オフライン・ダンプ・フォーマッタ  
 ー・ユーティリティ  
 (DFSOFMD0) 465  
 高速機能ログ分析ユーティリティ  
 (DBFULTA0) 465  
 統計分析ユーティリティ  
 (DFSISTS0) 465  
 ファイル選択およびフォーマット設定  
 印刷ユーティリティ  
 (DFSERA10) 465  
 ログ・トランザクション分析ユーティ  
 リティ (DFSILTA0) 465  
 IMS モニター報告書印刷ユーティリテ  
 ー (DFSUTR20) 465  
 変換、文字の  
 入力メッセージの指定 242  
 変更、オンラインで  
 オンライン変更コピー・ユーティリテ  
 ー 665  
 グローバル・オンライン変更コピー・  
 ユーティリティ 627  
 変更データ・タグ (MDT) 271  
 保護、画面の  
 DLFID ステートメントでのパラメータ  
 ー指定 270  
 保護解除、画面の  
 DLFID ステートメントでのパラメータ  
 ー指定 270  
 保存ユーティリティ 565

## [マ行]

マーシャル属性 150  
 マルチシステム制御ブロック 653  
 マルチシステム・パス・マップ、  
 MSC 656

メタデータ  
 アプリケーション  
 ACB Generation and Catalog  
 Populate ユーティリティ  
 (DFS3UACB) 367  
 IMS カタログへのロード 425  
 データベース  
 ACB Generation and Catalog  
 Populate ユーティリティ  
 (DFS3UACB) 367  
 IMS カタログへのロード 425

## [ヤ行]

ユーザー・データ・セットへのログ・レコ  
 ードのコピー 565  
 ユーティリティ  
 オフライン・ダンプ・フォーマッタ  
 ー・ユーティリティ  
 (DFSOFMD0) 537  
 オンライン変更コピー・ユーティリテ  
 ー (DFSUOCU0) 665  
 グローバル・オンライン変更ユーティ  
 リティ (DFSUOLC0) 627  
 高速機能ログ分析ユーティリティ  
 (DBFULTA0) 467  
 時間制御操作検査ユーティリティ  
 (DFSTVER0) 681  
 スプール SYSOUT 印刷ユーティリテ  
 ー (DFSUPRT0) 677  
 統計分析ユーティリティ  
 (DFSISTS0) 545  
 動的 SVC ユーティリティ  
 (DFSUSVC0) 623  
 バッチ SPOC ユーティリティ  
 (CSLUSPOC) 609  
 ファイル選択フォーマット設定印刷ユ  
 ーティリティ (DFSERA10) 491  
 複数システム検査ユーティリティ  
 (DFSUMSV0) 651  
 ログ保存ユーティリティ  
 (DFSUARC0) 565  
 ログ・トランザクション分析ユーティ  
 リティ (DFSILTA0) 531  
 ログ・マージ・ユーティリティ  
 (DFSMTMG0) 579  
 ログ・リカバリ・ユーティリティ  
 (DFSULTR0) 585  
 Catalog Directory Recovery ユーティ  
 リティ (DFS3RU00) 415  
 Create RDDS from Log Records ユ  
 ーティリティ (DFSURCL0) 705,  
 711  
 Create RDDS from MODBLKS ユ  
 ティリティ (DFSURCM0) 717

ユーティリティ (続き)

CSLULALE (OM 監査証跡フォーマット設定・印刷モジュール) 524

CSLURP10 (RDDS to Repository ユーティリティ) 695

CSLURP20 (Repository to RDDS ユーティリティ) 689

CSLUSPOC (バッチ SPOC ユーティリティ) 609

DBDGEN  
指定する情報 19  
使用するデータベース 24  
制御ステートメント 19

DFS3UCD0 387, 421

DFSURCL0 (Create RDDS from Log Records ユーティリティ) 705

DFSURCM0 (Create RDDS from MODBLKS ユーティリティ) 717

DFSURCP0 (RDDS コピー・ユーティリティ) 701

DFSURDD0 (RDDS 抽出ユーティリティ) 733

DFSURST0 (DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティ) 725

DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティ (DFSURST0) 725

IMS モニター報告書印刷ユーティリティ (DFSUTR20) 527

MFS 言語ユーティリティ (DFSUPAA0) 195

MFS サービス・ユーティリティ (DFSUTSA0) 635

MFS 装置特性テーブル・ユーティリティ (DFSUTB00) 189

OM 監査証跡フォーマット設定・印刷モジュール (CSLULALE) 524

PSBGEN 307

RDDS to Repository ユーティリティ (CSLURP10) 695

RDDS コピー・ユーティリティ (DFSURCP0) 701

RDDS 抽出ユーティリティ (DFSURDD0) 733

Repository to RDDS ユーティリティ (CSLURP20) 689

## [ラ行]

ラベル・フィールド  
DBD 生成 60

リテラル・フィールド  
出力メッセージ (output message)  
長さの指定 218

リテラル・フィールド (続き)  
出力メッセージ (output message) (続き)  
パスワード・パラメーターの長さ 267  
リテラルの切り捨て 262

領域占有率の要約報告書  
高速機能ログ分析ユーティリティ 486

例  
トークンを持つすべてのログ・レコード・タイプを選択 524  
トークンを持つ特定ログ・レコード・タイプを選択 524  
ファイル選択フォーマット設定印刷ユーティリティ 501

例外トランザクションの詳細リスト報告書  
高速機能ログ分析ユーティリティ 477

レコード・フォーマット設定・印刷モジュール (DFSERA30)  
概要 508  
制御ステートメント 508  
デッドロック報告書 509

レコード・フォーマット設定・印刷モジュール (DFSERS30)  
サブシステム検出デッドロック 516  
収集された補足情報 515  
デッドロック報告書 509  
デッドロック・ブロックのみの選択 516  
特殊な状態 515  
ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) 508  
報告書の読み方 510  
報告の例外 516  
ロック結果の状態 514  
ロック状態 513

連結 EQU 288

ログ分析ユーティリティ  
高速機能 (DBFULTA0) 467

ログ保存ユーティリティ (DFSUARC0)  
エラー処理 576  
オプションの機能 565  
概要 565  
強制ポリリュームの終わりの指定 566  
省略、SLDS のログ・レコードの 565  
推奨事項 566  
制御ステートメント 571  
制約事項 566  
前提条件 566  
バッチ DASD SLDS の保存 565  
プログラム出力 568  
ユーザー出口ルーチンの指定 565  
ユーザー・データ・セットへのログ・レコードのコピー 565

ログ保存ユーティリティ (DFSUARC0) (続き)  
要件 566  
例 576  
COPY ステートメント 573  
DD ステートメント 569  
EXIT ステートメント 575  
JCL の要件 569  
OLDS 入力 566  
RLDS (リカバリー・ログ・データ・セット) 565  
RLDS を作成する場合 565  
SLDS ステートメント 572  
SLDS 入力 567

ログ・トランザクション分析ユーティリティ (DFSILTA0)  
推奨事項 532  
制約事項 532  
説明 531  
前提条件 532  
パラメーターの説明 533  
プログラムからの出力 532  
プログラムへの入力 532  
要件 532

ログ・マージ・ユーティリティ (DFSMTMG0)  
概要 579  
推奨事項 579  
制御ステートメントの形式 580  
制約事項 579  
前提条件 579  
入力と出力 580  
要件 579  
ログ・マージの制御 580  
DD ステートメント 583  
JCL の要件 582  
MSC (複数システム結合機能) 580

ログ・ユーティリティ  
ログ保存ユーティリティ (DFSUARC0) 563  
ログ・マージ・ユーティリティ (DFSMTMG0) 563  
ログ・リカバリー・ユーティリティ (DFSULTR0) 563  
ログ・リカバリー・ユーティリティ (DFSULTR0)  
アクティブ PSB 報告書の印刷 600  
アクティブ領域メッセージ 593  
新しいログの作成 599  
エラー処理 600  
エラー・ブロックのリスト (SYSPRINT) 590  
概要 585  
出力 589  
推奨事項 587  
制御ステートメント 597

ログ・リカバリー・ユーティリティ  
(DFSULTR0) (続き)  
 制約事項 586  
 前提条件 586  
 単一ログの入力 587  
 中間ログの作成 598  
 中間ログ・エラー ID レコード 590  
 重複ログの入力  
 CLS モード 588  
 DUP モード 588  
 REP モード 588  
 データ・レコードのダンプ 593  
 入力 587  
 モード 585, 589  
 要件 587  
 例 601  
 CLS モード 585  
 CLS モードのエラー・リスト 590  
 DD ステートメント 595  
 DUP モード 585  
 DUP モードのエラー・リスト 590  
 JCL の要件 594  
 OLDS のリカバリー 586  
 PSB モード 585  
 REP モード 585  
 REP モード検査メッセージ 592  
 SLDS のリカバリー 587  
 ログ・レコード  
 統計分析ユーティリティ 545  
 論理  
 端末  
 マルチシステム制御ブロック 655  
 リンク  
 マルチシステム制御ブロック 653  
 論理端末、マルチシステム制御ブロック  
 651

## [数字]

3270 情報表示システム  
 コピー機能  
 リモート端末 270  
 選択ペン  
 指定 270

## A

ACB Generation and Catalog Populate  
 ユーティリティ (DFS3UACB) 367  
 概要 367  
 出力 369  
 推奨事項 369  
 制御ステートメント  
 形式 376  
 構文 376

ACB Generation and Catalog Populate  
 ユーティリティ (DFS3UACB) (続き)  
 制御ステートメント (続き)  
 パラメーター 377  
 要件 375  
 制約事項 368  
 前提条件 368  
 入力 369  
 戻りコード 385  
 要件 369  
 BUILD ステートメント  
 構文 376  
 パラメーター 377  
 DD ステートメント 373  
 DELETE ステートメント  
 構文 376  
 パラメーター 377  
 JCL 371  
 ACB (アプリケーション制御ブロック) 3  
 ACB 保守ユーティリティ (DFSRR00)  
 出力 6  
 制御ステートメント  
 形式 10  
 要件 9  
 BUILD 11  
 BUILD DBD 11  
 DELETE 11  
 説明 3  
 入力 6  
 戻りコード 13  
 例 14  
 ACBGEN プロシージャ 7  
 DFSACBCP 制御ステートメント 9  
 IMS.ACBLIB 3  
 JCL 7  
 ACB ライブラリー  
 コピー、IMS Catalog による 389, 399  
 ACBGEN 3  
 ACBGEN and Catalog Populate ユーテ  
 イリティ (DFS3UACB)  
 参照: ACB Generation and Catalog  
 Populate ユーティリティ  
 (DFS3UACB)  
 ACBGEN プロシージャ 7  
 DD ステートメント 8  
 EXEC ステートメント 8  
 ACBLIB データ・セット  
 ACB 保守ユーティリティ 3  
 ACCESS= パラメーター  
 DBD ステートメント 47  
 AOI (自動化操作プログラム・インターフ  
 ェース)  
 IOASIZE 要件 340  
 AREA ステートメント  
 キーワード 77  
 形式 77

AREA ステートメント (続き)  
 構文 77  
 パラメーターの説明 77  
 ATTR= オペランド (DFLD ステートメン  
 ト)  
 コピー・ロック 269  
 指定 269  
 パラメーター  
 NODET/DET/IDET 270  
 NODISP|HI 270  
 NOMOD|MOD 271  
 NOPROT/PROT 270  
 STRIP|NOSTRIP 271  
 YES,nn 272  
 ATTR= オペランド (MFLD ステートメン  
 ト)  
 指定 219

## B

BLOCK= パラメーター  
 DATASET ステートメント 67  
 BMP (バッチ・メッセージ処理)  
 アプリケーション・プログラム  
 データベース非コミット更新の制限  
 307  
 B= パラメーター  
 DFSERA10 OPTION ステートメント  
 498

## C

Catalog Directory Recovery ユーティリ  
 ティー  
 DFS3RU00 415  
 Catalog Record Purge ユーティリティ  
 451  
 Catalog ユーティリティ  
 Record Purge ユーティリティ 451  
 COMP= オペランド (DIV ステートメン  
 ト)  
 指定 251  
 COMP= パラメーター  
 ACBGEN プロシージャ 7  
 COND= オペランド (LPAGE ステートメ  
 ント) の指定 211  
 COND= パラメーター  
 ファイル選択フォーマット設定印刷ユ  
 ーティリティ (DFSERA10) 499  
 CONTROL ステートメント  
 ファイル選択フォーマット設定印刷ユ  
 ーティリティ (DFSERA10) 494  
 COPY オプション  
 ファイル選択フォーマット設定印刷ユ  
 ーティリティ (DFSERA10) 497

- COPY ステートメント  
 ログ保存ユーティリティ 573
- COPY ステートメント (言語ユーティリティ) 285
- Create RDDS from Log Records ユーティリティ (DFSURCL0)  
 推奨事項 706  
 制約事項 706  
 説明 705  
 前提条件 706  
 入力と出力 707  
 戻りコード 711  
 ユーティリティ制御ステートメント 709  
 要件 706  
 例  
 要約報告書の例 712  
 JCL の例 711  
 DD ステートメント 708  
 EXEC ステートメント 708  
 JCL 仕様 708
- Create RDDS from MODBLKS ユーティリティ (DFSURCM0)  
 推奨事項 719  
 制約事項 719  
 説明 717  
 前提条件 719  
 入力と出力 719  
 戻りコード 721  
 ユーティリティ制御ステートメント 721  
 要件 719  
 例  
 要約報告書の例 722  
 JCL の例 722  
 DD ステートメント 719  
 EXEC ステートメント 719  
 JCL 仕様 719
- CSLULALE (OM 監査証跡フォーマット設定・印刷モジュール)  
 指定した時刻範囲へのログ・データの制限 525  
 説明 524  
 ユーティリティ制御ステートメント 525  
 DD ステートメント 525  
 JCL 仕様 524
- CSLURP10 (RDDS to Repository ユーティリティ)  
 推奨事項 696  
 制約事項 695  
 説明 695  
 前提条件 695  
 入力と出力 696  
 戻りコード 698
- CSLURP10 (RDDS to Repository ユーティリティ) (続き)  
 ユーティリティ制御ステートメント 698  
 要件 696  
 例  
 要約出力の例 699  
 JCL の例 699  
 DD ステートメント 697  
 EXEC ステートメント 697  
 JCL 仕様 697
- CSLURP20 (Repository to RDDS ユーティリティ)  
 推奨事項 690  
 制約事項 689  
 説明 689  
 前提条件 689  
 入力と出力 690  
 戻りコード 691  
 ユーティリティ制御ステートメント 691  
 要件 689  
 例  
 要約出力の例 692  
 JCL の例 692  
 DD ステートメント 690  
 EXEC ステートメント 690  
 JCL 仕様 690
- CSLUSPOC (バッチ SPOC ユーティリティ)  
 推奨事項 609  
 制約事項 609  
 説明 609  
 前提条件 609  
 入力と出力 610  
 パラメーター・キーワード 611  
 戻りコード 612  
 要件 609  
 例 612  
 出力例 612  
 待ち時間を指定しない出力の例 613  
 JCL の例 612  
 EXEC ステートメント 611  
 JCL 仕様 611
- C= パラメーター  
 C= パラメーター  
 ファイル選択フォーマット設定印刷  
 ユーティリティ 499
- D**
- DATASET ステートメント  
 キーワード 65  
 形式 61  
 説明 58
- DATASET ステートメント (続き)  
 データベース  
 GSAM 61  
 HDAM 62  
 HIDAM 62  
 HISAM 63  
 HSAM 63  
 INDEX 64  
 LOGICAL 64  
 MSDB 64  
 パラメーターの説明 65
- DATXEXIT パラメーター  
 DBD ステートメント 57
- DBD 生成の指定  
 高速機能副次索引データベース 30  
 索引データベースと PSINDEX データベース 29  
 全機能副次索引データベース 30  
 論理データベース 31  
 GSAM データベース 25  
 HIDAM データベースと PHIDAM データベース 28  
 HISAM データベース 26  
 HSAM データベース 25, 29  
 MSDB データベース  
 ヘッダー情報 (BHDR) 29  
 SHISAM データベース 26  
 SHSAM データベース 25
- DBD 生成の入力レコード構造 (DEDB 以外)  
 要件 33  
 例外 32
- DBD (データベース記述) 生成  
 アセンブラの各種リスト 21  
 エラー条件 23  
 記述規則 34  
 コーディング規則 34  
 高速機能 DEDB 29  
 高速機能 MSDB 28  
 高速機能データベース (Fast Path databases) 36
- 索引の生成  
 副次索引 30  
 論理 31  
 HIDAM の 1 次 29
- 出力  
 アセンブラ・リスト 21  
 診断 21  
 セグメント・フラグのコード 21  
 タイプ 20  
 例 22  
 ロード・モジュール 23
- 診断 21  
 推奨事項 20  
 ステートメント・タイプの要約 31



- DBD (データベース記述) 生成 (続き)  
 制御インターバル・サイズ最小値の指定、データベースの 67  
 制御ステートメントの形式  
 END 165  
 制約事項 19  
 前提条件 19  
 入力レコード構造 33  
 パラメーターの説明 36  
 プロシージャ 186  
 ブロック・サイズ最小値の指定、データベースの 67  
 要件 20  
 ラベル・フィールド 60  
 例  
 共用副次索引 181  
 高速機能 DEDB 173  
 高速機能 MSDB 172  
 高速機能副次索引 183  
 索引の生成 165, 170  
 副次索引 180  
 副次索引関係または論理関係 165  
 論理関係 174  
 GSAM 171  
 HDAM 168  
 HIDAM 169  
 HISAM 167  
 HSAM 166  
 AREA ステートメント  
 キーワード 77  
 形式 77  
 構文 77  
 説明 77  
 DATASET ステートメント  
 形式 61  
 説明 58  
 分割、データベースを複数のデータ・セット・グループに 59  
 ラベル・フィールド 60  
 DBD ステートメント 36  
 DBDGEN ステートメント 165  
 DBDGEN の概要  
 構成 19  
 使用するデータベース 24  
 制御ステートメント 19  
 DD ステートメント 75, 76  
 DEDB データベース 36  
 DEDB のオプションの指定  
 DBDGEN ステートメント 165  
 END ステートメント 165  
 FINISH ステートメント 165  
 DFSCASE ステートメント  
 キーワード 161  
 説明 161  
 パラメーター 161
- DBD (データベース記述) 生成 (続き)  
 DFSMAP ステートメント  
 キーワード 158  
 説明 157  
 パラメーター 158  
 DFSMARSH ステートメント  
 キーワード 151  
 説明 150  
 パラメーター 151  
 END ステートメント 165  
 FIELD ステートメント  
 キーワード 131  
 形式 131  
 説明 122  
 FINISH ステートメント 165  
 GSAM データベース 25  
 GSAM (汎用順次アクセス方式) 36  
 HDAM データベース 27, 36  
 HIDAM データベース 28, 36  
 HISAM データベース 26, 36  
 HSAM データベース 25, 36  
 LCHILD ステートメント  
 形式 117  
 説明 112  
 副次索引関係を定義 113  
 論理関係を定義 113  
 1 次索引関係の定義 113  
 MSDB データベース 36  
 SEGM ステートメント  
 キーワード 92  
 キーワードの省略語 92  
 説明 79  
 ポインター・キーワードのオプションと省略語 96  
 SHISAM データベース 26  
 SHSAM データベース 25  
 XDFLD ステートメント  
 キーワード 144  
 形式 144  
 説明 144  
 DBD ライブラリー  
 コピー、IMS Catalog による 389, 399  
 DBDGEN  
 プロシージャ  
 JCL パラメーター 186  
 DBDGEN ユーティリティ 19  
 制御ステートメント、入力順序 31  
 DBD= キーワード  
 ACB Generation and Catalog  
 Populate ユーティリティ (DFS3UACB) 377  
 ACB 保守ユーティリティ 11  
 DBFULTA0 (高速機能ログ分析ユーティリティ)  
 エラー処理 476  
 概要 467
- DBFULTA0 (高速機能ログ分析ユーティリティ) (続き)  
 高速機能報告書タイプ 476  
 推奨事項 469  
 制約事項 468  
 前提条件 469  
 入力と出力 469  
 報告書  
 通過時間総合要約 484  
 トランザクション・コード別例外詳細要約報告書 483  
 分析の総括 488  
 リソースの使用量と競合の総合要約 484  
 領域占有率の要約 486  
 例外トランザクションの詳細リスト 477  
 VSO アクティビティの要約 487  
 ユーティリティ制御ステートメント 471  
 要件 469  
 JCL の要件 470  
 DBRC (データベース・リカバリー管理) 615  
 DD ステートメント  
 DFSUPRT0 679  
 DD1= パラメーター  
 AREA ステートメント 77  
 DDNAME= キーワード  
 制御ステートメント  
 DFSERA10 CONTROL 495  
 DFSERA10 OPTION 500  
 DDNOUT= キーワード  
 DFSERA10 制御ステートメント 495  
 DEDB (高速処理データベース) 定義  
 セグメント 80  
 フィールド 123  
 DBD 生成 29  
 DFSCASE ステートメント 161  
 DFSMAP ステートメント 157  
 DFSMARSH ステートメント 150  
 命名 37  
 DFSCASE ステートメント 161  
 DFSMAP ステートメント 157  
 DFSMARSH ステートメント 150  
 DELETE 機能 (MFS サービス・ユーティリティ DFSUTSA0) 643  
 DEV ステートメント  
 指定 223  
 FEAT= オペランド 236  
 FTAB= オペランド 232  
 PFK= オペランド 238  
 SLDI= オペランド 241  
 SLDP= オペランド 242  
 SUB= オペランド 242

DEV ステートメント (続き)  
 VERSID= オペランド 242  
 VTAB= オペランド 241

DFLD (装置フィールド・ステートメント)  
 215  
 生成された DFLD ステートメントの  
 印刷 262  
 反復処理 215, 263  
 PASSWORD パラメーター 267

DFS3CCE0  
 出力 391  
 推奨事項 391  
 制約事項 390  
 前提条件 391  
 統計報告書、エクスポート 397  
 入力 391  
 戻りコード 399  
 要件 391  
 JCL 393, 404

DFS3CCE0 ユーティリティ  
 概要 389

DFS3CCIO  
 出力 400  
 推奨事項 400  
 制約事項 400  
 前提条件 400  
 統計報告書、インポート 412  
 入力 400  
 戻りコード 414  
 要件 400  
 JCL 403, 404

DFS3CCIO ユーティリティ  
 概要 389

DFS3LU00 443

DFS3PU00  
 概要 425  
 出力 428  
 推奨事項 428  
 制約事項 427  
 前提条件 427  
 統計報告書 439  
 入力 428  
 戻りコード 441  
 要件 427  
 DD ステートメント 395, 432  
 JCL 430

DFS3PU10 (IMS Catalog Record Purge  
 ユーティリティ) 451

DFS3RU00 (Catalog Directory Recovery  
 ユーティリティ) 415

DFS3UACB ユーティリティ  
 概要 367  
 出力 369  
 推奨事項 369  
 制約事項 368  
 前提条件 368

DFS3UACB ユーティリティ (続き)  
 入力 369  
 戻りコード 385  
 要件 369  
 DD ステートメント 373  
 JCL 371

DFS3UCD0 387, 421

DFSACBCP 制御ステートメント 9

DFSCASE ステートメント  
 キーワード 161  
 パラメーター 161

DFSERA10 (ファイル選択フォーマット設  
 定印刷ユーティリティ)  
 オプションのキーワード 497  
 B= 498  
 COND= 499  
 C= 499  
 DDNAME= 500  
 D= 500  
 EXITR= 499  
 E= 499  
 FLDLEN= 498  
 FLDTYP= 498  
 H= 498  
 L= 498  
 OFFSET= 497  
 O= 497  
 PARM= 497  
 PRSYS= 500  
 P= 500  
 STARTAF= 498  
 STOPAFT= 498  
 SYM= 497  
 T= 498  
 VALUE= 498  
 V= 498  
 概要 491  
 出力 492  
 推奨事項 492  
 制御ステートメント  
 説明 494  
 COMMENT 501  
 CONTROL 494  
 END 500  
 OPTION 496  
 制約事項 491  
 前提条件 491  
 入力 492  
 要件 491  
 例 501  
 COPY オプション 497  
 JCL の要件  
 説明 492  
 DD ステートメント 492  
 NEGOF オプション 497  
 PRINT オプション 497

DFSERA10 ユーティリティ・モジュー  
 ル  
 拡張選択モジュール (DFSERA70) 508  
 プログラム分離トレース・レコード・  
 フォーマット設定・印刷モジュール  
 (DFSERA40) 508  
 レコード・フォーマット設定・印刷モ  
 ジュール (DFSERA30) 508  
 DL/I 呼び出しイメージ・キャプチャ  
 ー・モジュール (DFSERA50) 508  
 IMS トレース・テーブル・レコード・  
 フォーマット設定・印刷モジュール  
 (DFSERA60) 508  
 OM 監査証跡フォーマット設定・印刷  
 モジュール (CSLULALE) 508

DFSERA30 (レコード・フォーマット設  
 定・印刷モジュール)  
 概要 508  
 サブシステム検出デッドロック 516  
 収集された補足情報 515  
 制御ステートメント 508  
 デッドロック報告書 509  
 デッドロック・ブロックのみの選択  
 516  
 特殊な状態 515  
 報告書の読み方 510  
 報告の例外 516  
 ロック状態 513

DFSERA40 (プログラム分離トレース・レ  
 コード・フォーマット設定・印刷モジュー  
 ル)  
 概要 517  
 出力例 517  
 制御ステートメント 520  
 デッドロック報告書 509

DFSERA50 (DL/I 呼び出しイメージ・キ  
 ャプチャー・モジュール)  
 概要 520  
 制御ステートメント 520

DFSERA60 (IMS トレース・テーブル・  
 レコード・フォーマット設定・印刷モジ  
 ュール)  
 概要 521  
 制御ステートメント 521

DFSERA70 (拡張選択出口ルーチン)  
 概要 521  
 例 523

DFSILTA0 (ログ・トランザクション分析  
 ユーティリティ)  
 推奨事項 532  
 制約事項 532  
 前提条件 532  
 プログラムからの出力 532  
 プログラムへの入力 532  
 要件 532

- DFSIST30 (報告書作成プログラム)  
作成される報告書の説明と例  
回線および端末報告書 548  
キューイングされたが送信されなかったメッセージ (宛先別) 547  
キューイングされたが送信されなかったメッセージ (トランザクション・コード別) 548  
トランザクション応答報告書 549  
プログラム間メッセージ (宛先別) 547  
プログラム間メッセージ (トランザクション・コード別) 548  
統計分析ユーティリティ 547  
プログラム・モジュール  
報告書作成プログラム (DFSIST30) 547
- DFSIST40 (メッセージ選択/コピー (リスト))  
統計分析ユーティリティ 550
- DFSISTS0 (統計分析ユーティリティ)  
推奨事項 545  
制約事項 545  
前提条件 545  
要件 545  
例 555
- DFSMTMG0 (ログ・マージ・ユーティリティ)  
概要 579  
推奨事項 579  
制御ステートメントの形式 580  
制約事項 579  
前提条件 579  
入力と出力 580  
要件 579  
ログ・マージの制御 580  
DD ステートメント 583  
JCL の要件 582  
MSC (複数システム結合機能) 580
- DFSMAP ステートメント  
キーワード 158  
パラメーター 158
- DFSMTARSH ステートメント  
キーワード 151  
パラメーター 151
- DFSMTNTR0、データ通信モニター 527
- DFSMTREC 制御ステートメント 674
- DFSOFMD0 (オフライン・ダンプ・フォーマッター・ユーティリティ)  
概要 537  
環境  
DB バッチ 540  
DBCTL 539  
DB/DC 539  
DCCTL 539  
TM バッチ 540
- DFSOFMD0 (オフライン・ダンプ・フォーマッター・ユーティリティ) (続き)  
推奨事項 539  
制約事項 538  
前提条件 538  
ダンプ形式制御データ・セット  
サブセット・オプション 541  
説明 541  
DD ステートメント 541  
ダンプ・フォーマッター 537  
入力と出力 539  
マイグレーションの考慮事項 537  
要件 538  
ロード・モジュール 537  
IPCS 540  
SDUMP 539
- DFSMTVER0 (時間制御操作検査ユーティリティ)  
出力  
エラー報告書 684  
時間スケジュール要求テーブル 684  
時間スケジュール要求報告書 (タイム・エレメント報告書) 684  
説明 682  
統計報告書 684  
メッセージ・テーブル報告書 685  
要約報告書 686  
推奨事項 681  
制約事項 681  
前提条件 681  
要件 681  
EXEC ステートメント 682  
JCL 仕様 682
- DFSUARCO (ログ保存ユーティリティ)  
エラー処理 576  
オプションの機能 565  
概要 565  
強制ボリュームの終わりの指定 566  
省略、SLDS のログ・レコードの 565  
推奨事項 566  
制御ステートメント 571  
制約事項 566  
前提条件 566  
バッチ DASD SLDS の保存 565  
プログラム出力 568  
ユーザー出口ルーチンの指定 565  
ユーザー・データ・セットへのログ・レコードのコピー 565  
要件 566  
例 576  
COPY ステートメント 573  
DD ステートメント 569  
EXIT ステートメント 575  
JCL の要件 569  
OLDS 入力 566
- DFSUARCO (ログ保存ユーティリティ) (続き)  
RLDS (リカバリー・ログ・データ・セット) 565  
RLDS を作成する場合 565  
SLDS ステートメント 572  
SLDS 入力 567
- DFSUDT0x (装置特性テーブル) 648  
画面サイズの指定 229  
装置特性テーブル 229
- DFSULTR0 (ログ・リカバリー・ユーティリティ)  
エラー・ブロックのリスト (SYSPRINT) 590  
概要 585  
推奨事項 587  
制約事項 586  
前提条件 586  
単一ログの入力 587  
中間ログ・エラー ID レコード 590  
重複ログの入力  
CLS モード 588  
DUP モード 588  
REP モード 588  
入力 587  
モード 589  
要件 587  
CLS モード 585  
DUP モード 585  
OLDS のリカバリー 586  
PSB モード 585  
REP モード 585  
SLDS のリカバリー 587
- DFSUMSV0 (複数システム検査ユーティリティ)  
推奨事項 652  
ユーティリティ制御ステートメント 663  
要件 652  
EXEC ステートメント 662
- DFSUOCU0 (オンライン変更コピー・ユーティリティ)  
アクティブ・ライブラリー (active library) 667  
概要 665  
使用されるライブラリー 667  
推奨事項 667  
ステー징・ライブラリー 667  
制約事項 666  
前提条件 666  
取り消し 666  
非アクティブ・ライブラリー (inactive library) 667  
プロシージャ・ステートメント 667, 674  
要件 667

- DFSUOCU0 (オンライン変更コピー・ユーティリティ) (続き)
- DD ステートメント 670
  - DFSMREC 制御ステートメント 674
  - EXEC ステートメント 668
  - INITMOD プロシージャ 674
  - JCL 673
  - MSDB 666
  - OLCUTL プロシージャ 672
- DFSUOLC0 (グローバル・オンライン変更ユーティリティ)
- 概要 627
  - 推奨事項 629
  - 制約事項 629
  - 前提条件 629
  - パラメーター 629
  - 要件 629
  - 例 633
  - JCL 629
  - OLCSTAT データ・セット 627
- DFSUPAA0 (MFS 言語ユーティリティ)
- 区画セット定義ステートメント 280
    - PD ステートメント 281
    - PDB ステートメント 280
    - PDBEND ステートメント 283
  - コンパイル・ステートメント 285
    - ALPHA ステートメント 285
    - COPY ステートメント 286
    - EJECT ステートメント 291
    - END ステートメント 291
    - EQU ステートメント 287
    - PRINT ステートメント 290
    - RESCAN ステートメント 288
    - SPACE ステートメント 291
    - STACK ステートメント 289
    - TITLE ステートメント 290
    - UNSTACK ステートメント 289
  - 推奨事項 197
  - 制御ブロック 195
  - 制約事項 196
  - 前提条件 196
  - テーブル定義ステートメント 283
    - IF ステートメント 284
    - TABLE ステートメント 283
    - TABLEEND ステートメント 285
  - テスト・モード (MFSTEST プロシージャ)
    - ソース・ステートメント・プリプロセッサ 301
    - フェーズ 1 プリプロセッサ 302
    - フェーズ 2 302
  - 標準モード (MFSUTL プロシージャ)
    - フェーズ 1 294
    - フェーズ 2 295
  - フォーマット定義ステートメント 222
- DFSUPAA0 (MFS 言語ユーティリティ) (続き)
- DEV ステートメント 223
  - DFLD ステートメント 263
  - DIV ステートメント 243
  - DO ステートメント 260
  - DPAGE ステートメント 252
  - ENDDO ステートメント 279
  - FMT ステートメント 222
  - FMTEND ステートメント 279
  - PPAGE ステートメント 259
  - RCD ステートメント 263
- フォーマット・セット (format set) 195
- メッセージ定義ステートメント 208
    - DO ステートメント 214
    - ENDDO ステートメント 221
    - LPAGE ステートメント 210
    - MFLD ステートメント 215
    - MSG ステートメント 208
    - MSGEND ステートメント 222
    - PASSWORD ステートメント 212
    - SEG ステートメント 212
  - モード 195
  - 要件 197
- DFSUPRT0 (スプール SYSOUT 印刷ユーティリティ)
- 推奨事項 677
  - 制約事項 677
  - 前提条件 677
  - 要件 677
  - DD ステートメント 679
- DFSURCL0 (Create RDDS from Log Records ユーティリティ)
- 推奨事項 706
  - 制約事項 706
  - 説明 705
  - 前提条件 706
  - 入力と出力 707
  - 戻りコード 711
  - ユーティリティ制御ステートメント 709
  - 要件 706
  - 例
    - 要約報告書の例 712
    - JCL の例 711
  - DD ステートメント 708
  - EXEC ステートメント 708
  - JCL 仕様 708
- DFSURCM0 (Create RDDS from MODBLKS ユーティリティ)
- 推奨事項 719
  - 制約事項 719
  - 説明 717
  - 前提条件 719
  - 入力と出力 719
- DFSURCM0 (Create RDDS from MODBLKS ユーティリティ) (続き)
- 戻りコード 721
  - ユーティリティ制御ステートメント 721
  - 要件 719
  - 例
    - 要約報告書の例 722
    - JCL の例 722
  - DD ステートメント 719
  - EXEC ステートメント 719
  - JCL 仕様 719
- DFSURCP0 (RDDS コピー・ユーティリティ)
- 推奨事項 702
  - 制約事項 701
  - 説明 701
  - 前提条件 701
  - 入力と出力 702
  - 戻りコード 703
  - ユーティリティ制御ステートメント 703
  - 要件 702
  - 例
    - 要約出力の例 704
    - JCL の例 703
  - DD ステートメント 702
  - EXEC ステートメント 702
  - JCL 仕様 702
- DFSURDD0 (RDDS 抽出ユーティリティ)
- 推奨事項 733
  - 制約事項 733
  - 説明 733
  - 前提条件 733
  - 入力と出力 733
  - 戻りコード 735
  - ユーティリティ制御ステートメント 735
  - 要件 733
  - 例
    - 照会報告書の例 737
    - JCL の例 736
  - DD ステートメント 734
  - EXEC ステートメント 734
  - JCL 仕様 734
- DFSURST0 (DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティ)
- 推奨事項 727
  - 制約事項 727
  - 説明 725
  - 前提条件 727
  - 入力と出力 727
  - 戻りコード 730
  - ユーティリティ制御ステートメント 729

DFSURST0 (DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティ) (続き)

要件 727  
例

要約報告書の例 731

JCL の例 730

DD ステートメント 728

EXEC ステートメント 728

JCL 仕様 728

DFSUSVC0 (動的 SVC ユーティリティ)

エラー処理 625

概要 623

出力 624

推奨事項 624

制約事項 623

前提条件 623

入力 624

戻りコード 625

要件 624

例 625

DD ステートメント 625

JCL の要件 624

DFSUTB00

推奨事項 190

前提条件 190

DFSUTB00 (MFS 装置特性テーブル・ユーティリティ)

ユーティリティの実行 192

要件 190

DD ステートメント 191

DFSUTR20 (IMS モニター報告書印刷ユーティリティ)

概要 527

推奨事項 528

制約事項 527

前提条件 527

入力 528

分析制御データ・セット 528

要件 528

用語の定義 527

JCL の要件 528

JCL の例 530

DFSUTSA0 (MFS サービス・ユーティリティ)

推奨事項 636

前提条件 636

ユーティリティ制御ステートメントのキーワード 639

ユーティリティ制御ステートメントのパラメーター 638

要件 636

DD ステートメント 638

EXEC ステートメント 637

DFSWTnnn プロシージャ (スプール SYSOOUT 印刷ユーティリティ)

DFSUPRT0) 678

DIF (装置入力形式)

作成に使用される言語ステートメント

要約 206

DEV 223

DFLD 263

DIV 243

DO 260

DPAGE 252

ENDDO 279

FMT 222

FMTEND 279

PPAGE 259

RCD 263

DIV ステートメント

COMPR= オペランド 251

HDRCTL= オペランド 247

DL/I 呼び出しイメージ・キャプチャー・モジュール (DFSERA50)

概要 520

制御ステートメント 520

ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) 520

DOF (装置出力形式)

作成に使用される言語ステートメント

要約 206

DEV 223

DFLD 263

DIV 243

DO 260

DPAGE 252

ENDDO 279

FMT 222

FMTEND 279

PPAGE 259

RCD 263

DRD

ユーティリティ 689

DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティ (DFSURST0)

推奨事項 727

制約事項 727

説明 725

前提条件 727

入力と出力 727

戻りコード 730

ユーティリティ制御ステートメント 729

要件 727

例

要約報告書の例 731

JCL の例 730

DD ステートメント 728

EXEC ステートメント 728

DRD IMS SYSGEN ステージ 1 事前構文解析ユーティリティ (DFSURST0) (続き)

JCL 仕様 728

DSPURX00 (データベース・リカバリー管理ユーティリティ) 615

推奨事項 615

制約事項 615

前提条件 615

ユーティリティの呼び出し 619

要件 615

例 618

D= キーワード

制御ステートメント

DFSERA10 CONTROL 495

DFSERA10 OPTION 500

## E

EJECT ステートメント (言語ユーティリティ) 291

END ステートメント (言語ユーティリティ) 291

ENDDO ステートメント

終わりの指定

DFLD ステートメント 279

MFLD ステートメント 221

EQU 処理 286

EQU ステートメント (言語ユーティリティ・ステートメント) 286

EXEC ステートメント

スプール SYSOOUT 印刷ユーティリティ (DFSUPRT0) 679

MFS 装置特性テーブル

(DFSUTB00) 191

TCO 検査ユーティリティ

(DFSTVER0) 682

EXEC ステートメントのオペランド

DEVCHAR= 200

EXITR= キーワード

DFSERA10 OPTION 制御ステートメント 499

EXIT= パラメーター

データ・キャプチャー出口ルーチン 52

DBD ステートメント 52

E= キーワード

DFSERA10 OPTION 制御ステートメント 499

## F

FEAT= オペランド (DEV ステートメント) の指定 236

FIELD ステートメント

キーワード 131

## FIELD ステートメント (続き)

形式 131

索引データベース 128

説明 122

DEDB データベース 123

GSAM データベース 124

HDAM データベースと PHDAM データベース 124

HIDAM データベースと PHIDAM データベース 125

HISAM データベース 126

HSAM データベース 128

MSDB データベース 129

SHISAM データベース 130

SHSAM データベース 128

## FILL= オペランド

MFLD ステートメントの指定 220

MSG ステートメントの指定 209

## FLDLN= キーワード

DFSERA10 OPTION 制御ステートメント 498

## FLDTYP= キーワード

DFSERA10 OPTION 制御ステートメント 498

FMT ステートメントの指定 222

## FMTCPY 制御ステートメント

MFSBTCH2 プロシージャ 300

MFSUTL プロシージャ 297

FMTEND ステートメントの指定 279

## FRSPC= キーワード

DATASET ステートメント 73

## FTAB= オペランド (DEV ステートメント)

指定 232

## G

### GSAM (汎用順次アクセス方式)

DBD 生成 36

指定 25, 47

例 171

PCB 生成

例 345

## H

### HALDB (高可用性ラージ・データベース) 区画

データベース非コミット更新の制限 307

### HDAM (階層直接アクセス方式)

定義

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

### HDAM (階層直接アクセス方式) (続き)

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

### HDAM データベース

DBD (データベース記述) 生成 36

### HDRCTL= オペランド

DIV ステートメントの指定 247

### HIDAM (階層索引直接アクセス方式) 定義

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

### HIDAM データベース

DBD (データベース記述) 生成 36

### HISAM (階層索引順次アクセス方式) 定義

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

### HISAM データベース

DBD (データベース記述) 生成 36

### HSAM (階層順次アクセス方式) 定義

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

### HSAM データベース

DBD (データベース記述) 生成 36

### H= ステートメント

DFSERA10 OPTION 制御ステートメント 498

## I

### IF ステートメント

指定 283

### IFP 領域のトランザクション・コード別の通過時間総合要約報告書

高速機能ログ分析ユーティリティ 484

### IFP 領域のトランザクション・コード別例外詳細要約報告書

高速機能ログ分析ユーティリティ 483

### IMS Catalog Copy ユーティリティ (DFS3CCE0、DFS3CCIO)

概要 389

### IMS Catalog Export ユーティリティ (DFS3CCE0)

概要 389

出力 391

推奨事項 391

制約事項 390

前提条件 391

統計報告書、エクスポート 397

入力 391

戻りコード 399

要件 391

DD ステートメント 395

JCL 393

### IMS Catalog Import ユーティリティ (DFS3CCIO)

概要 399

出力 400

推奨事項 400

制約事項 400

前提条件 400

統計報告書、インポート 412

入力 400

戻りコード 414

要件 400

DD ステートメント

SYSINP 407

JCL 403, 404

SYSINP DD ステートメント 407

### IMS Catalog Library Builder ユーティリティ (DFS3LU00) 443

### IMS Catalog Populate ユーティリティ (DFS3PU00)

概要 425

出力 428

推奨事項 428

制約事項 427

前提条件 427

統計報告書 439

入力 428

戻りコード 441

要件 427

DD ステートメント 432

MANAGEDACBS ステートメント、SYSINP 381, 409, 435

SYSINP 375, 433

JCL 430

SYSINP DD ステートメント 375, 433

MANAGEDACBS ステートメント 381, 409, 435

### IMS Catalog Record Purge ユーティリティ (DFS3PU10) 451

### IMS カタログ

コピー 389, 399

IMS カタログ (続き)  
サイズ変更 439  
セグメント、削除 451  
単一ステップでの ACBGEN とデータの取り込み 367  
データの取り込み 425  
定義 387, 421  
統計 439  
ユーティリティ  
DFS3UCD0 387, 421  
レコードの除去 451  
ロード 425  
ACB Generation and Catalog  
Populate ユーティリティ  
(DFS3UACB) 367  
ACB ライブラリー  
コピー 389, 399  
ACBGEN 中のデータの取り込み 367  
ACBGEN 中のデータのロード 367  
DBD ライブラリー  
コピー 389, 399  
DBRC、使用しない 387, 421  
PSB ライブラリー  
コピー 389, 399  
IMS カタログ・ユーティリティ 365  
IMS ダンプ・フォーマッター 539  
IMS トレース・テーブル・レコード・フォーマット設定・印刷モジュール (DFSERA60)  
概要 521  
制御ステートメント 521  
ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) 521  
IMS のサブシステム検出デッドロック 516  
IMS パスワード  
指定 267  
PASSWORD ステートメント 212  
IMS モニター報告書  
オフライン・ダンプ・フォーマッター・ユーティリティ  
(DFSOFMD0) 537  
ログ保存ユーティリティ  
(DFSUARC0) 565  
DB/DC  
統計分析ユーティリティ  
(DFSISTS0) 545  
ログ・トランザクション分析ユーティリティ (DFSILTA0) 531  
ログ・マージ・ユーティリティ  
(DFSMTMG0) 579  
ログ・リカバリー・ユーティリティ  
(DFSULTR0) 585  
IMS モニター報告書印刷ユーティリティ (DFSUTR20)  
概要 527

IMS モニター報告書印刷ユーティリティ (DFSUTR20) (続き)  
推奨事項 528  
制約事項 527  
前提条件 527  
入力 528  
分析制御データ・セット 528  
要件 528  
用語の定義 527  
JCL の要件 528  
JCL の例 530  
IMSMSV プロシージャ (複数システム検査ユーティリティ DFSUMSV0) 662  
IMSWTnnn プロシージャ (スプールSYSOUT 印刷ユーティリティ DFSUPRT0) 680  
IMS.FORMAT ライブラリー  
バックアップおよび復元操作 303, 305  
IMS.REFERAL ライブラリー  
バックアップおよび復元操作 303, 305  
INDEX 機能 (MFS サービス・ユーティリティ DFSUTSA0) 642  
INITMOD プロシージャ  
プロシージャ・ステートメント 674  
DFSMMREC 制御ステートメント 674  
MODSTAT レコード 674  
IPCS (対話式問題制御システム)  
オフライン・ダンプ・フォーマッター 540  
オフライン・ダンプ・フォーマッターのユーザー制御ステートメント 540  
IMS ダンプ・フォーマッター 539

## J

JUST= オペランド (MFLD ステートメント) の指定 219

## L

LCHILD ステートメント  
索引データベース 117  
HDAM データベース 114  
HIDAM データベース 116  
HISAM データベース 114  
PHDAM データベース 115  
PHIDAM データベース 116  
PSINDEX データベース 118  
LIST 機能 (MFS サービス・ユーティリティ DFSUTSA0) 646  
LOG パラメーター  
DBD ステートメント 55  
LOGICAL パラメーター  
DATASET ステートメント 65

LPAGE  
オペランド 211  
COND= 211  
SOR= 211  
出力  
条件付き選択 211  
LTH= オペランド (MFLD ステートメント) の指定 218  
LUSIZE= オペランド (PDB ステートメント) の指定 280  
L= キーワード  
DFSERA10 OPTION 制御ステートメント 498

## M

MANAGEDACBS ステートメント 381, 409, 435  
MBR= パラメーター  
DBD 生成 186  
MFLD (メッセージ・フィールド・ステートメント)  
生成された MFLD ステートメントの印刷 221  
反復処理 214, 215  
ATTR= オペランド 219  
FILL= オペランド 220  
JUST= オペランド 219  
LTH= オペランド 218  
MFS 言語ユーティリティ  
コンパイル・ステートメント 202  
要約 207  
ALPHA 202  
COPY 285  
EJECT 291  
END 291  
EQU 286  
RESCAN 288  
SPACE 290  
SYSIN 202  
SYSLIB 202  
SYSPRINT 202  
TITLE 290  
UNSTACK 289  
連結 EQU 288  
MFS 言語ユーティリティ (DFSUPAA0)  
区画セット定義ステートメント 280  
PD ステートメント 281  
PDB ステートメント 280  
PDBEND ステートメント 283  
コンパイル・ステートメント 285  
ALPHA ステートメント 285  
COPY ステートメント 286  
EJECT ステートメント 291  
END ステートメント 291  
EQU ステートメント 287

- MFS 言語ユーティリティ  
(DFSUPAA0) (続き)  
コンパイル・ステートメント (続き)  
PRINT ステートメント 290  
RESCAN ステートメント 288  
SPACE ステートメント 291  
STACK ステートメント 289  
TITLE ステートメント 290  
UNSTACK ステートメント 289  
推奨事項 197  
制御ブロック 195  
制約事項 196  
説明 195  
前提条件 196  
テーブル定義ステートメント 283  
IF ステートメント 284  
TABLE ステートメント 283  
TABLEEND ステートメント 285  
テスト・モード (MFSTEST プロシ  
ジャー)  
説明 300  
ソース・ステートメント・プリプロ  
セッサ 301  
フェーズ 1 プリプロセッサ 302  
フェーズ 2 302  
領域パラメータの見積もり 200  
バッチ・モード  
説明 297  
MFSBTCH1 プロシジャー 298  
MFSBTCH2 プロシジャー 299  
標準モード (MFSUTL プロシジャー  
ー)  
フェーズ 1 294  
フェーズ 2 295  
領域パラメータの見積もり 200  
フォーマット定義ステートメント 222  
DEV ステートメント 223  
DFLD ステートメント 263  
DIV ステートメント 243  
DO ステートメント 260  
DPAGE ステートメント 252  
ENDDO ステートメント 279  
FMT ステートメント 222  
FMTEND ステートメント 279  
PPAGE ステートメント 259  
RCD ステートメント 263  
フォーマット・セット (format  
set) 195  
メッセージ定義ステートメント 208  
DO ステートメント 214  
ENDDO ステートメント 221  
LPAGE ステートメント 210  
MFLD ステートメント 215  
MSG ステートメント 208  
MSGEND ステートメント 222  
PASSWORD ステートメント 212
- MFS 言語ユーティリティ  
(DFSUPAA0) (続き)  
メッセージ定義ステートメント (続き)  
SEG ステートメント 212  
モード 195  
要件 197  
DD 名 (MFSRVC)  
FORMAT 202  
REFIN 202  
SYSIN 202  
SYSPRINT 202  
SYSSNAP 202  
DD 名 (MFSULT、MFSBTCH1、  
MFSBTCH2)  
DUMMY 201  
FORMAT 201  
REFIN 201  
REFOUT 201  
REFRD 201  
SYSIN 201  
SYSLIB 201  
SYSUT3 201  
SYSUT4 201  
UTPRINT 202  
FMTCPY 制御ステートメント  
MFSBTCH2 プロシジャー 300  
MFSUTL プロシジャー 297  
JCL のパラメータの説明  
PCOMP= 197  
PSUBS= 197  
PXREF= 197  
JCL の要件  
MFSBACK プロシジャー 303  
MFSBTCH1 プロシジャー 298  
MFSBTCH2 プロシジャー 300  
MFSREST プロシジャー 305  
MFSTEST プロシジャー 302  
MFSUTL プロシジャー 296  
REFCPY 制御ステートメント  
MFSBTCH1 プロシジャー 299  
MFSULT プロシジャー 296  
MFS サービス・ユーティリティ  
(DFSUTSA0)  
推奨事項 636  
制約事項 636  
説明 635  
前提条件 636  
ユーティリティ制御ステートメント  
638  
ユーティリティ制御ステートメン  
トのキーワード 639  
FMT= 639  
ユーティリティ制御ステートメン  
トのパラメータ 638  
要件 636  
DD ステートメント 638
- MFS サービス・ユーティリティ  
(DFSUTSA0) (続き)  
DELETE 機能 643  
EXEC ステートメント 637  
INDEX 機能 642  
LIST 機能  
出力 648  
LIST 機能の出力 646  
MFSRVC プロシジャー 636  
PROC ステートメント  
説明 637  
RELATE 機能 645  
SCRATCH 機能 643  
MFS 装置特性テーブル・ユーティリ  
ティ (DFSUTB00)  
推奨事項 190  
制約事項 189  
説明 189  
前提条件 190  
ユーティリティの実行 192  
要件 190  
DD ステートメント 191  
EXEC ステートメント  
説明 191  
MFS 記述子形式 190  
MFSUCT プロシジャー 190  
PROC ステートメント  
説明 190  
MFSBACK プロシジャー (MFS 言語ユ  
ーティリティ)  
説明 303  
JCL の要件 303  
MFSBTCH1 プロシジャー (MFS 言語  
ユーティリティ)  
説明 298  
JCL の要件 298  
MFSBTCH2 プロシジャー (MFS 言語  
ユーティリティ)  
説明 299  
JCL の要件 300  
MFSUCT  
MFS 装置特性テーブル・ユーティリ  
ティ (DFSUTB00) 189  
MFSREST プロシジャー (MFS 言語ユ  
ーティリティ)  
説明 305  
JCL の要件 305  
MFSRVC プロシジャー (MFS サービ  
ス・ユーティリティ) 636  
MFSTEST プロシジャー (MFS 言語ユ  
ーティリティ)  
ステップ 1 (ソース・ステートメン  
ト・プリプロセッサ) 301  
ステップ 1 (フェーズ 1) 302  
ステップ 2 (フェーズ 2) 302  
領域パラメータの見積もり 200



MFSTEST プロシージャ (MFS 言語ユーティリティ) (続き)  
 JCL の要件 302

MFSUTL プロシージャ (MFS 言語ユーティリティ)  
 ステップ 1 (フェーズ 1) 294  
 ステップ 1 (プリプロセッサ) 294  
 ステップ 2 (フェーズ 2) 295  
 領域パラメーターの見積もり 200  
 JCL の要件 296

MID (メッセージ入力記述子)  
 作成に使用される言語ステートメント 206  
 要約 206  
 DO 214  
 ENDDO 221  
 LPAGE 210  
 MFLD 215  
 MSG 208  
 MSGEND 221  
 PASSWORD 212  
 SEG 212

MOD (メッセージ出力記述子)  
 作成に使用される言語ステートメント 206  
 要約 206  
 DO 214  
 ENDDO 221  
 LPAGE 210  
 MFLD 215  
 MSG 208  
 MSGEND 221  
 PASSWORD 212  
 SEG 212

MODEL= パラメーター  
 DATASET ステートメント 65

MODSTAT レコード、INITMOD プロシージャ 674

MSC (複数システム結合機能) 651  
 ログ・マージ・ユーティリティ  
 出力 580  
 入力 580

MSDB (主記憶データベース)  
 定義  
 DFSCASE ステートメント 161  
 DFSMAP ステートメント 157  
 DFSMARSH ステートメント 150  
 DFSCASE ステートメント 161  
 DFSMAP ステートメント 157  
 DFSMARSH ステートメント 150

MSDB の DBD 生成  
 説明 28

MSG ステートメント  
 FILL= オペランド 209

MSGEND ステートメント  
 指定 221

## N

NAME パラメーター  
 DBD ステートメント 46

NEGOF オプション  
 ファイル選択フォーマット設定プログラム (DFSERA10) 497

NOLOG パラメーター  
 DBD ステートメント 55

NULL  
 圧縮の指定 251  
 充てん文字  
 出力装置フィールド 209  
 入力メッセージ・フィールド 220

## O

OFFSET= キーワード  
 DFSERA10 OPTION 制御ステートメント 497

OLCSTAT データ・セット  
 初期設定 627  
 説明 627  
 リカバリー手順 627

OLCUTL プロシージャの処理 672

OLDS (オンライン・ログ・データ・セット)  
 終了 567  
 重複 OLDS 566  
 リカバリー・ポイント 567  
 ログ保存ユーティリティへの入力 566  
 ログ・リカバリー・ユーティリティを使用したリカバリー 586

OM 監査証跡フォーマット設定・印刷モジュール (CSLULALE)  
 指定した時刻範囲へのログ・データの制限 525  
 説明 524  
 ユーティリティ制御ステートメント 525  
 DD ステートメント 525  
 JCL 仕様 524

OSAM データ・セット・ブロック・サイズ 70

OVFLW= パラメーター  
 DATASET ステートメント 66

O= キーワード  
 制御ステートメント  
 DFSERA10 CONTROL 495  
 DFSERA10 OPTION 497

## P

PARM= キーワード  
 サブパラメーター  
 TOKEN= サブパラメーター 522  
 XFMT= サブパラメーター 521  
 DFSERA10 OPTION 制御ステートメント 497

PASSWD= パラメーター  
 DBD ステートメント 51

PASSWORD ステートメントの指定 212

PASSWORD パラメーター (DFLD ステートメント) の指定 267

PCB ステートメント  
 全機能または高速機能データベース 315  
 データベース PCB のサイズ 315  
 GSAM 330  
 SENSEG ステートメント 332

PCB (プログラム制御ブロック)  
 DB  
 データベース非コミット更新の制限 307

PD ステートメント (区画定義)  
 指定 281

PDB (区画記述子ブロック)  
 作成に使用される言語ステートメント  
 要約 207  
 PDBEND 207  
 LUSIZE= オペランド 280

PDBEND ステートメントの指定 283

PF キー (3270)  
 指定 238

PFK= オペランド (DEV ステートメント) の指定 238

PHDAM (区分化階層直接アクセス方式)  
 定義  
 DFSCASE ステートメント 161  
 DFSMAP ステートメント 157  
 DFSMARSH ステートメント 150

DBD 生成  
 例 171

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

PHIDAM (区分化階層索引直接アクセス方式)  
 定義  
 DFSCASE ステートメント 161  
 DFSMAP ステートメント 157  
 DFSMARSH ステートメント 150

DBD 生成  
 例 171

DFSCASE ステートメント 161

DFSMAP ステートメント 157

DFSMARSH ステートメント 150

PPAGE ステートメントの指定 259  
PRINT オプション  
    ファイル選択フォーマット設定印刷ユーティリティ (DFSERA10) 497  
PRINT ステートメント (言語ユーティリティ) 290  
PROC ステートメント  
    スプール SYSOUT 印刷ユーティリティ (DFSUPRT0) 679  
    複数システム検査ユーティリティ (DFSUMSV0) 662  
    MFS 装置特性テーブル・ユーティリティ (DFSUTB00) 190  
PRTSYS= キーワード  
    DFSERA10 OPTION 500  
PSB 生成ユーティリティ  
    高速機能データベース  
        処理オプション 326  
PSB (プログラム仕様ブロック)  
    実行 363  
    制御ステートメントの形式  
        全機能または高速機能データベースの PCB 315  
        代替 PCB (alternate PCB) 311  
        END 343  
        GSAM PCB 330  
        I/O PCB 311  
        PSBGEN 338  
        SENFLD 337  
        SENSEG 332  
    生成  
        概要 307  
        制御ステートメント 311  
        制御ステートメントの形式 15, 16  
    説明 307  
    全機能または高速機能データベースの PCB  
        PROCOPT パラメーター 315  
    動的オプション  
        IMS 管理の ACB 環境での追加 17  
    例  
        アプリケーション・データベース 351  
        階層データ構造の例 344  
        共用副次索引 357  
        高速機能 (Fast Path) 347  
        フィールド・レベル・センシティブティ 346  
        論理データベース 348  
        GSAM 345  
DEDB のオプションの指定  
    END ステートメント 343  
    PSBGEN ステートメント 338  
    SENSEG ステートメント 332  
PCB (プログラム連絡ブロック) 307  
PCBNAME= パラメーター 345

PSB ライブラリー  
    コピー、IMS Catalog による 389, 399  
PSBGEN  
    ステートメント  
        データベース PCB の最大数 315  
        PSB 生成 338  
        プロシージャ 362  
PSBGEN ユーティリティ  
    規則 308  
    推奨事項 308  
    生成  
        入力と出力 308  
    制約事項 308  
    前提条件 308  
    要件 308  
    6 つの入出力ステートメントのタイプ 308  
PSB= キーワード  
    ACB Generation and Catalog Populate ユーティリティ (DFS3UACB) 377  
PSB= パラメーター  
    ACB 保守ユーティリティ 11  
PT (プログラム・タブ) 機能  
    充てん文字 (fill character) 209  
P= キーワード  
    DFSERA10 OPTION 制御ステートメント 500

## R

RCD ステートメントの指定 263  
RDDS to Repository ユーティリティ (CSLURP10)  
    推奨事項 696  
    制約事項 695  
    説明 695  
    前提条件 695  
    入力と出力 696  
    戻りコード 698  
    ユーティリティ制御ステートメント 698  
    要件 696  
    例  
        要約出力の例 699  
        JCL の例 699  
    DD ステートメント 697  
    EXEC ステートメント 697  
    JCL 仕様 697  
RDDS コピー・ユーティリティ (DFSURCP0)  
    推奨事項 702  
    制約事項 701  
    説明 701  
    前提条件 701  
    入力と出力 702

RDDS コピー・ユーティリティ (DFSURCP0) (続き)  
    戻りコード 703  
    ユーティリティ制御ステートメント 703  
    要件 702  
    例  
        要約出力の例 704  
        JCL の例 703  
    DD ステートメント 702  
    EXEC ステートメント 702  
    JCL 仕様 702  
RDDS 抽出ユーティリティ (DFSURDD0)  
    推奨事項 733  
    制約事項 733  
    説明 733  
    前提条件 733  
    入力と出力 733  
    戻りコード 735  
    ユーティリティ制御ステートメント 735  
    要件 733  
    例  
        照会報告書の例 737  
        JCL の例 736  
    DD ステートメント 734  
    EXEC ステートメント 734  
    JCL 仕様 734  
RECFM= パラメーター  
    DATASET ステートメント 72  
RECON データ・セット  
    アップグレードする 615  
    バージョン・マイグレーション 615  
RECORD= パラメーター  
    DATASET ステートメント 72  
REFCPY 制御ステートメント  
    MFSBTCH1 プロシージャ 299  
    MFSULT プロシージャ 296  
RELATE 機能 (MFS サービス・ユーティリティ DFSUTSA0) 645  
REL= パラメーター  
    DATASET ステートメント 74  
Repository to RDDS ユーティリティ (CSLURP20)  
    推奨事項 690  
    制約事項 689  
    説明 689  
    前提条件 689  
    入力と出力 690  
    戻りコード 691  
    ユーティリティ制御ステートメント 691  
    要件 689  
    例  
        要約出力の例 692

Repository to RDDS ユーティリティ  
(CSLURP20) (続き)  
例 (続き)  
JCL の例 692  
DD ステートメント 690  
EXEC ステートメント 690  
JCL 仕様 690  
RESCAN ステートメント (言語ユーティ  
リティ) 288  
RGN= パラメーター  
プロシージャ  
ACBGEN 7  
DBDGEN 187  
RLDS (リカバリー・ログ・データ・セッ  
ト)  
作成 565  
ログ保存ユーティリティへの出力  
568  
RMNAME= パラメーター  
DBD ステートメント 49  
ROOT= パラメーター  
AREA ステートメント 78

## S

SCA (システム制御域)  
指定 217  
SCAN= パラメーター  
DATASET ステートメント 73  
SCRATCH 機能 (MFS サービス・ユーテ  
ィリティ DFSUTSA0) 643  
SDUMP  
オフライン・ダンプ・フォーマッター  
539  
SEARCHA= パラメーター  
DATASET ステートメント 73  
SEG ステートメント  
EXIT= オペランド 212  
GRAPHIC= オペランド 212  
SEGM ステートメント  
キーワードの省略語 92  
形式 92  
説明 79  
データベース  
DEDB 80  
GSAM 81  
HDAM 82  
HIDAM 83  
HISAM 85  
HSAM 87  
INDEX 87  
MSDB 88  
PHDAM 88  
PHIDAM 90  
PSINDEX 92  
SHSAM 92

SEGM ステートメント (続き)  
ポインター・キーワードのオプション  
と省略語 96  
SENSEG ステートメント  
PSB 生成 332  
SHISAM (単純階層索引順次アクセス方  
式)  
定義  
DFSCASE ステートメント 161  
DFSMAP ステートメント 157  
DFSMARSH ステートメント 150  
DFSCASE ステートメント 161  
DFSMAP ステートメント 157  
DFSMARSH ステートメント 150  
SHSAM (単純階層順次アクセス方式)  
定義  
DFSCASE ステートメント 161  
DFSMAP ステートメント 157  
DFSMARSH ステートメント 150  
DFSCASE ステートメント 161  
DFSMAP ステートメント 157  
DFSMARSH ステートメント 150  
SIZE= パラメーター  
AREA ステートメント 77  
DATASET ステートメント 69  
SKIP= パラメーター  
ファイル選択フォーマット設定印刷ユ  
ーティリティ (DFSERA10) 495  
SLDI= オペランド (DEV ステートメン  
ト) の指定 241  
SLDP= オペランド (DEV ステートメン  
ト) の指定 242  
SLDS (システム・ログ・データ・セット)  
バッチ保存 565  
ログ保存ユーティリティへの出力  
568  
ログ保存ユーティリティへの入力  
567  
ログ・リカバリー・ユーティリティ  
を使用したリカバリー 587  
ログ・レコードの省略 565  
SLDS (システム・ログ・データ・セット)  
ステートメント  
ログ保存ユーティリティ 572  
SLU  
MFS におけるタイプ 2 の動作の定義  
コピー機能 270  
SOR= オペランド (LPAGE ステートメン  
ト) の指定 211  
SOUT= パラメーター  
プロシージャ  
ACBGEN 7  
DBDGEN 187  
SPACE ステートメント (言語ユーティ  
ティ) 290

STACK ステートメント (MFS 言語ユー  
ティリティ) 289  
STARTAF= ステートメント  
DFSERA10 OPTION 制御ステートメ  
ント 498  
STOPAFT= ステートメント  
ファイル選択フォーマット設定印刷ユ  
ーティリティ (DFSERA10) 495  
DFSERA10 OPTION 制御ステートメ  
ント 498  
SUB= オペランド (DEV ステートメント)  
指定 242  
SVC ユーティリティ 623  
SYM= オプション  
DFSERA10 OPTION 制御ステートメ  
ント 497  
SYS2= パラメーター  
プロシージャ  
ACBGEN 7  
DBDGEN 187  
SYSID パス、MSC 654  
SYSINP DD ステートメント 375, 407,  
433  
SYSIN/SYSLIB レコードのスタッキング  
とアンスタッキング  
STACK ステートメント 289  
UNSTACK 289  
SYSMDUMP  
オフライン・ダンプ・フォーマッター  
540  
SYSPRINT  
ログ保存ユーティリティ 568  
SYSPRINT リスト制御  
コンパイル・ステートメント 202  
EJECT ステートメント 291  
PRINT ステートメント 290  
SPACE ステートメント 290  
TITLE ステートメント 290

## T

TABLE ステートメントの指定 283  
TABLEEND ステートメントの指定 285  
TCO エラー報告書  
時間制御操作検査ユーティリティ  
(DFSTVER0) 684  
TCO 検査プロシージャ  
時間制御操作検査ユーティリティ  
(DFSTVER0) 682  
TCO 検査ユーティリティ (DFSTVER0)  
DD ステートメント 682  
TCO 時間スケジュール要求テーブル  
時間制御操作検査ユーティリティ  
(DFSTVER0) 684

TCO スクリプト・ライブラリー  
時間制御操作検査ユーティリティー  
(DFSTVER0) 681

TCO タイマー・エレメント報告書  
時間制御操作検査ユーティリティー  
(DFSTVER0) 684

TCO 統計報告書  
時間制御操作検査ユーティリティー  
(DFSTVER0) 684

TCO メッセージ・テーブル報告書  
時間制御操作検査ユーティリティー  
(DFSTVER0) 685

TCO 要約報告書  
時間制御操作検査ユーティリティー  
(DFSTVER0) 686

TITLE ステートメント (言語ユーティリ  
ティー) 290

TOKEN= サブパラメーター 522

T= キーワード  
DFSERA10 OPTION 制御ステートメ  
ント 498

## U

UNSTACK ステートメント (言語ユーテ  
ィリティー) 289

UOW= パラメーター  
AREA ステートメント 78

## V

VALUE= パラメーター  
DFSERA10 OPTION 制御ステートメ  
ント 498

VERSID= オペランド (DEV ステートメ  
ント) の指定 242

VERSION パラメーター  
DBD ステートメント 56

VSO アクティビティの要約報告書  
高速機能ログ分析ユーティリティー  
487

VTAB= オペランド (DEV ステートメン  
ト)  
指定 241

V= パラメーター  
DFSERA10 OPTION 制御ステートメ  
ント 498

## W

WADS (先行書き込みデータ・セット)  
データ・セット 596  
CLS モード 585, 593  
NOWADS 598

## X

XDFLD ステートメント  
キーワード 144  
形式 144  
説明 144  
パラメーターの説明 147  
HDAM データベース 146  
HISAM データベース 145  
PHDAM データベース 146  
XFMT= サブパラメーター 521

## [特殊文字]

/TEST MFS コマンド 196





プログラム番号: 5635-A06  
5655-DS5  
5655-TM4

Printed in Japan

SC43-4270-00



日本アイ・ビー・エム株式会社  
〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IMS バージョン 15.1.0

システム・ユーティリティー

