

**Adabas**

ユーティリティ

バージョン 8.1.3

June 2008

---

Adabas

This document applies to Adabas Version 8.1.3 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © Software AG 1971-2008. All rights reserved.

The name Software AG™, webMethods™, Adabas™, Natural™, ApplinX™, EntireX™ and/or all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

# 目次

1 ユーティリティ	1
2 表記規則	3
コントロールステートメント構文	4
構文形式	5
コントロールステートメント規則	7
パラメータ値	8
3 ADAACK ユーティリティ：アドレスコンバータのチェック	11
4 機能概要	13
5 ACCHECK：データストレージに対するアドレスコンバータのチェック	15
オプションパラメータ	16
例	17
6 JCL/JCS の必要項目と指定例	19
BS2000	20
z/OS	21
z/VM	22
VSE	23
7 ADACDC ユーティリティ：変更データの取得	25
8 機能概要	27
処理フェーズと結果ファイル	28
プライマリ入力データ	31
プライマリ出力ファイル	32
トランザクションファイル	33
9 ユーティリティの実行	35
オプションパラメータ	36
ADACDC での ISNREUSE の使用	38
10 オペレーティングシステムに応じた考慮事項	41
z/OS	42
VSE	43
BS2000	43
11 ADACDC ユーザー出口	45
ユーザー出口のインストール	46
ユーザー出口インターフェイス	65
ユーザー出口呼び出し	48
12 例	51
13 JCL/JCS の必要項目と指定例	53
BS2000	54
z/OS	55
z/VM	56
VSE	58
14 ADACMP ユーティリティ：データの圧縮／圧縮解除	61
15 機能概要	63
COMPRESS 機能の概要	64
DECOMPRESS 機能の概要	65

16 入力データの必要条件 .....	67
入力データの構造 .....	68
マルチプルバリューフィールドカウント .....	68
ピリオディックグループカウント .....	70
可変長フィールドのサイズ .....	73
17 処理 .....	75
セグメント化されたレコードに関する考慮事項 .....	76
データの検証 .....	82
データ圧縮 .....	82
非圧縮データの LOB 値と値参照の表現 .....	84
圧縮レコード内の 191 を超える MU および PE オカレンスの特定 .....	85
再スタートに関する考慮事項 .....	85
ユーザー出口 6 .....	85
18 COMPRESS：Adabas ファイルの圧縮 .....	87
基本パラメータとサブパラメータ .....	88
オプションのパラメータとサブパラメータ .....	90
ADACMP COMPRESS 例 .....	97
19 DECOMPRESS：Adabas ファイルの圧縮解除 .....	99
オプションのパラメータとサブパラメータ .....	185
マルチクライアントファイルの圧縮解除 .....	105
ADACMP DECOMPRESS .....	106
20 フィールド定義ステートメント .....	107
FNDEF：フィールド定義およびグループ定義 .....	108
FNDEF：ピリオディックグループ定義 .....	125
COLDE：照合ディスクリプタ定義 .....	128
HYPDE：ハイパーディスクリプタ定義 .....	131
PHONDE：フォネティックディスクリプタ .....	135
SUBDE：サブディスクリプタ定義 .....	136
SUBFN：サブフィールド定義 .....	138
SUPDE：スーパーディスクリプタ定義 .....	139
SUPFN：スーパーフィールド定義 .....	148
21 JCL/JCS の必要項目と指定例 .....	151
ユーザー出口を使用した ADACMP .....	152
BS2000 .....	153
z/OS .....	156
z/VM .....	160
VSE .....	162
22 COMPRESS 機能の出力 .....	165
圧縮されたデータレコード .....	166
拒否されたデータレコード .....	166
ADACMP レポート .....	173
23 DECOMPRESS 機能の出力 .....	175
拒否されたデータレコード .....	176
24 ADACNV ユーティリティ：データベースの変換 .....	181
25 機能概要 .....	183

データベースステータス .....	184
手順 .....	216
26 CONVERT:上位バージョンへのデータベースの変換 .....	187
オプションパラメータ .....	188
変換の考慮点 .....	189
例 .....	190
27 REVERT:下位バージョンへのデータベースの逆変換 .....	191
基本パラメータとサブパラメータ .....	192
オプションパラメータ .....	192
逆変換の考慮点 .....	193
例 .....	194
28 JCL/JCS の必要項目と指定例 .....	195
BS2000 .....	196
z/OS .....	244
z/VM .....	200
VSE .....	253
29 ADADBS ユーティリティ:データベースサービス .....	203
30 機能概要 .....	205
TEST パラメータによる構文チェック .....	206
31 ADD:データセットの追加 .....	207
アソシエータまたはデータストレージデータセット .....	208
基本パラメータとサブパラメータ .....	209
オプションパラメータ .....	209
例 .....	209
32 ALLOCATE:ファイルエクステンツの割り当て .....	211
基本パラメータ .....	212
オプションパラメータ .....	212
例 .....	284
33 CHANGE:フィールドの標準長の変更 .....	215
基本パラメータ .....	216
オプションパラメータ .....	217
例 .....	218
34 CVOLSER:指定されたディスクボリューム内のAdabas ボリュームの出力 .....	219
基本パラメータ .....	220
オプションパラメータ .....	220
例 .....	221
35 DEALLOCATE:ファイルエクステンツの割り当て解除 .....	223
基本パラメータ .....	224
オプションパラメータ .....	224
例 .....	225
36 DECREASE:アソシエータデータストレージの縮小 .....	227
基本パラメータ .....	228
オプションパラメータ .....	228
例 .....	229
手順 .....	229

37 DELCP：チェックポイントレコードの削除 .....	231
基本パラメータ .....	232
オプションパラメータ .....	232
例 .....	233
38 DELETE：ファイルの削除 .....	235
基本パラメータ .....	236
オプションパラメータ .....	236
例 .....	237
39 DSREUSE：データストレージブロックの再使用 .....	239
基本パラメータ .....	295
オプションパラメータ .....	240
例 .....	241
40 ENCODEF：ファイルエンコードの変更 .....	243
基本パラメータ .....	244
オプションパラメータ .....	245
例 .....	245
41 INCREASE：アソシエータデータストレージの増加 .....	247
基本パラメータ .....	248
オプションパラメータ .....	248
例 .....	249
一般手順 .....	249
オペレーティングシステム固有の手順 .....	249
42 ISNREUSE：ISN の再使用 .....	255
基本パラメータ .....	256
オプションパラメータ .....	256
例 .....	257
43 MODFCB：ファイルパラメータの修正 .....	259
基本パラメータ .....	260
オプションパラメータ .....	260
例 .....	262
44 MUPEX：MU フィールドおよび PE フィールドの最大数の設定 .....	263
構文 .....	265
基本パラメータ .....	265
オプションパラメータ .....	265
例 .....	266
45 NEWFIELD：新しいフィールドの追加 .....	267
基本パラメータ .....	268
オプションパラメータ .....	269
例 .....	270
46 ONLINVERT：オンラインインバートの開始 .....	271
基本パラメータ .....	272
オプションパラメータ .....	273
例 .....	274
47 ONLREORFASSO：ファイルのアソシエータのオンラインリオーダ開始 .....	275
基本パラメータ .....	276

オプションパラメータ .....	277
例 .....	277
48 ONLREORFDATA：ファイルデータのオンラインリオーダの開始 .....	279
基本パラメータ .....	280
オプションパラメータ .....	281
例 .....	282
49 ONLREORFILE：ファイルのアソシエータとデータのオンラインリオーダの開 始 .....	283
基本パラメータ .....	284
オプションパラメータ .....	285
例 .....	286
50 OPERCOM：Adabasオペレータコマンド .....	287
クラスタ環境での OPERCOM コマンドの使用 .....	288
オプションパラメータ .....	289
オペレータコマンド .....	290
51 PRIORITY：ユーザープライオリティの変更 .....	305
基本パラメータ .....	379
オプションパラメータ .....	306
例 .....	382
52 RECORDSPANNING：レコードのスパニングの有効化／無効化 .....	309
構文 .....	311
基本パラメータ .....	311
オプションパラメータ .....	311
例 .....	312
53 RECOVER：スペースの回復 .....	313
オプションパラメータ .....	314
54 REFRESH：ファイルを 0 件状態にする .....	315
基本パラメータ .....	386
オプションパラメータ .....	386
例 .....	317
55 REFRESHSTATS：統計値のリフレッシュ .....	319
オプションパラメータ .....	320
例 .....	321
56 RELEASE：ディスクリプタの解除 .....	323
基本パラメータ .....	324
オプションパラメータ .....	325
例 .....	394
57 RENAME：ファイルデータベースの名前の変更 .....	327
基本パラメータ .....	328
オプションパラメータ .....	328
例 .....	329
58 RENUMBER：ファイル番号の変更 .....	331
基本パラメータ .....	332
オプションパラメータ .....	332
例 .....	333

59 REPLICATION：レプリケーションのアクティブ化／非アクティブ化 .....	335
基本パラメータ .....	336
オプションパラメータ .....	337
例 .....	338
60 RESETDIB：アクティブなユーティリティリストのエントリのリセット .....	339
基本パラメータ .....	340
オプションパラメータ .....	341
例 .....	341
61 RESETPPT：PPT のリセット .....	343
構文 .....	345
基本パラメータ .....	345
オプションパラメータ .....	345
62 SPANCOUNT：スパンドレコード数のカウント .....	347
構文 .....	348
基本パラメータ .....	348
オプションパラメータ .....	348
例 .....	349
63 TRANSACTIONS：中断トランザクションの再開 .....	351
基本パラメータ .....	353
オプションパラメータ .....	405
例 .....	354
64 UNCOUPLE：ファイルのアンカップル .....	355
基本パラメータ .....	356
オプションパラメータ .....	406
例 .....	357
65 JCL/JCS の必要項目と指定例 .....	359
ユーザー出口を使用した照合 .....	360
BS2000 .....	360
z/OS .....	361
z/VM .....	362
VSE .....	363
66 ADADCK ユーティリティ：データストレージのチェック .....	365
67 機能概要 .....	367
68 DSCHECK 機能：データストレージのチェック .....	369
オプションのパラメータとサブパラメータ .....	370
例 .....	372
69 JCL/JCS の必要項目と指定例 .....	373
BS2000 .....	374
z/OS .....	375
z/VM .....	376
VSE .....	377
70 ADADEF ユーティリティ：データベースの定義 .....	379
71 機能概要 .....	381
データベースコンポーネント .....	382
チェックポイントファイル .....	382

72 DEFINE：データベースおよびチェックポイントファイルの定義 .....	383
基本パラメータ .....	386
オプションパラメータ .....	387
例 .....	392
73 MODIFY：エンコードの変更 .....	393
オプションパラメータ .....	394
例 .....	396
74 NEWWORK：定義ワークファイル .....	397
基本パラメータ .....	398
オプションパラメータ .....	398
例 .....	399
75 JCL/JCS の必要項目と指定例 .....	401
BS2000 .....	402
z/OS .....	403
z/VM .....	405
VSE .....	407
76 ADAFRM ユーティリティ：Adabas データセットのフォーマット .....	409
77 機能概要 .....	411
ステートメントの制約 .....	412
フォーマットオペレーション .....	412
78 データベースコンポーネントのフォーマット .....	413
フォーマットモード .....	414
構文 .....	414
基本パラメータ .....	416
オプションパラメータ .....	416
例 .....	418
79 JCL/JCS の必要項目と指定例 .....	419
BS2000 .....	420
z/OS .....	422
z/VM .....	423
VSE .....	425
80 ADAICK ユーティリティ：インデックスおよびアドレスコンバータのチェッ ク .....	427
81 機能概要 .....	429
82 ACCHECK：アドレスコンバータのチェック .....	431
基本パラメータ .....	432
オプションパラメータ .....	432
出力例 .....	433
83 ASSOPRINT：アソシエータブロックの出力／ダンプ .....	435
基本パラメータ .....	436
オプションパラメータ .....	436
84 BATCH：出力幅を 1 行当たり 132 文字に設定 .....	437
オプションパラメータ .....	438
85 DATAPRINT：データストレージブロックの出力／ダンプ .....	439
基本パラメータ .....	440

オプションパラメータ .....	440
86 DSCHECK：データストレージレコードの出力／ダンプ .....	441
基本パラメータ .....	442
オプションパラメータ .....	442
出力例 .....	443
87 DUMP：ダンプ出力形式 .....	445
オプションパラメータ .....	446
88 FCBPRINT：ファイルコントロールブロックの出力／ダンプ .....	447
基本パラメータ .....	448
オプションパラメータ .....	448
出力に関する考慮事項 .....	449
89 FDTPRINT：フィールド定義テーブルの出力／ダンプ .....	451
基本パラメータ .....	452
オプションパラメータ .....	452
90 GCBPRINT：ジェネラルコントロールブロック（GCB）の出力／ダンプ .....	453
オプションパラメータ .....	454
91 ICHECK：インデックスおよびアドレスコンバータのチェック .....	455
基本パラメータ .....	456
オプションパラメータ .....	530
92 INT：解釈済みの出力形式 .....	457
オプションパラメータ .....	458
93 NIPRINT：ノーマルインデックスの出力／ダンプ .....	459
基本パラメータ .....	460
オプションパラメータ .....	460
94 NOBATCH：出力幅を1行当たり80文字に設定 .....	461
オプションパラメータ .....	462
95 NODUMP：ダンプ出力形式の省略 .....	463
オプションパラメータ .....	464
96 NOINT：解釈済みの形式の省略 .....	465
オプションパラメータ .....	466
97 PPTPRINT：Parallel Participant Table の出力／ダンプ .....	467
オプションパラメータ .....	468
出力例 .....	468
98 UIPRINT：アッパーインデックスの出力／ダンプ .....	471
基本パラメータ .....	472
オプションパラメータ .....	472
99 例 .....	473
100 JCL/JCS の必要項目と指定例 .....	475
ユーザー出口を使用した照合 .....	476
BS2000 .....	476
z/OS .....	478
z/VM .....	479
VSE .....	480
101 ADAINV ユーティリティ：インバーテッドリストの管理 .....	481
102 機能概要 .....	483

103 COUPLE：定義ファイルカップリングディスクリプタ .....	485
基本パラメータ .....	486
オプションパラメータ .....	487
例 .....	488
ファイルカップリング用の一時スペース .....	488
アソシエータカップリングリスト .....	489
スペースカップリングリスト .....	490
スペース割り当て .....	491
104 INVERT：ディスクリプタの作成 .....	493
基本パラメータ .....	494
オプションのパラメータとサブパラメータ .....	495
INVERT 機能のスペース割り当て .....	497
例 .....	498
105 JCL/JCS の必要項目と指定例 .....	499
ユーザー出口を使用した照合 .....	500
BS2000 .....	500
z/OS .....	503
z/VM .....	505
VSE .....	507
106 ADALOD ユーティリティ：ファイルローダー .....	509
107 機能概要 .....	511
108 LOAD：ファイルのロード .....	513
基本パラメータ .....	515
オプションのパラメータとサブパラメータ .....	517
例 .....	530
LOAD データとスペースの要件 .....	531
拡張ファイルのロード .....	535
マルチクライアントファイルのロード .....	537
109 UPDATE：レコードの追加／削除 .....	539
基本パラメータ .....	660
オプションのパラメータとサブパラメータ .....	542
例 .....	547
ISN の指定形式 .....	548
UPDATE データとスペースの要件 .....	550
拡張ファイルの一括更新 .....	551
110 ローダーストレージの要件および使用 .....	553
111 TEMP データセットスペースの使用状況 .....	555
シーケンシャル TEMP データセット .....	556
112 ADALOD スペース／統計レポート .....	559
113 JCL/JCS の必要項目と指定例 .....	561
ユーザー出口を使用した照合 .....	562
BS2000 .....	562
z/OS .....	566
z/VM .....	568
VSE .....	571

114 ADAMER ユーティリティ：ADAM 見積り .....	575
115 機能概要 .....	577
116 ADAM アクセス回数の見積り .....	579
基本パラメータ .....	580
オプションパラメータ .....	580
例 .....	582
117 ADAMER 出力レポートの内容 .....	583
118 JCL/JCS の必要項目と指定例 .....	585
BS2000 .....	586
z/OS .....	587
z/VM .....	588
VSE .....	742
119 ADAORD ユーティリティ：リオーダ機能 .....	591
120 機能概要 .....	593
リオーダ機能 .....	594
再構築機能 .....	595
格納機能 .....	595
スペース割り当て .....	596
Adabas 8 の考慮事項 .....	596
121 REORASSO：アソシエータのリオーダ .....	597
オプションのパラメータとサブパラメータ .....	763
例 .....	603
122 REORDATA：データストレージのリオーダ .....	605
オプションのパラメータとサブパラメータ .....	606
例 .....	609
123 REORDB：データベースのリオーダ .....	611
オプションのパラメータとサブパラメータ .....	614
例 .....	620
124 REORFASSO：1つのファイルに対するアソシエータのリオーダ .....	623
基本パラメータ .....	625
オプションパラメータ .....	625
例 .....	628
125 REORFDATA：1つのファイルに対するデータストレージのリオーダ .....	631
基本パラメータ .....	632
オプションパラメータ .....	633
例 .....	636
126 REORFILE：ファイルのリオーダ .....	637
基本パラメータ .....	640
オプションパラメータ .....	640
例 .....	814
127 RESTRUCTUREDB: データベースの再構築 .....	647
オプションのパラメータとサブパラメータ .....	649
例 .....	652
128 RESTRUCTUREF：個別のファイルの再構築 .....	653
基本パラメータ .....	655

オプションパラメータ .....	655
例 .....	658
129 STORE：ファイルの格納 .....	659
オプションのパラメータとサブパラメータ .....	662
例 .....	668
130 JCL/JCS の必要項目と指定例 .....	671
BS2000 .....	672
z/OS .....	677
z/VM .....	680
VSE .....	683
131 ADAPLP ユーティリティ：プロテクションログ／ワークの出力 .....	687
132 機能概要 .....	689
133 ADAPLP の構文と例 .....	691
オプションのパラメータとサブパラメータ .....	693
例 .....	696
134 JCL/JCS の必要項目と指定例 .....	699
BS2000 .....	700
z/OS .....	704
z/VM .....	707
VSE .....	709
135 ADAPRI ユーティリティ：選択した Adabas ブロックの出力 .....	713
136 機能概要 .....	715
137 ADAPRI の構文と例 .....	717
基本パラメータ .....	718
オプションパラメータ .....	718
例 .....	719
138 JCL/JCS の必要項目と指定例 .....	721
BS2000 .....	722
z/OS .....	723
z/VM .....	724
VSE .....	725
139 ADARAI ユーティリティ：Adabas Recovery Aid .....	727
140 機能概要 .....	729
概念およびコンポーネント .....	730
141 CHKDB：データベースステータスのチェック .....	733
142 DISABLE：リカバリロギングの非アクティブ化 .....	735
143 LIST：現在の RLOG 世代の表示 .....	737
BS2000 における LIST の追加情報 .....	739
構文 .....	934
オプションパラメータ .....	740
例 .....	741
144 PREPARE：RLOG の初期化および開始 .....	751
構文 .....	752
基本パラメータ .....	753
オプションパラメータ .....	753

例 .....	753
145 RECOVER：リカバリジョブストリームの構築 .....	755
リカバリ処理 .....	756
最適化されたりカバリ処理 .....	758
要件 .....	758
制約 .....	759
リカバリに必要な入力 .....	760
リカバリ操作の出力 .....	760
RECOVER 機能の実行 .....	761
ファイルレベルのリカバリ .....	762
構文 .....	961
オプションのパラメータとサブパラメータ .....	764
例 .....	968
スケルトンジョブ制御 .....	767
JCL を変更するユーザー出口 .....	770
リカバリ前のチェック .....	770
RECOVER 機能またはリカバリジョブストリームの再スタート .....	771
146 REMOVE：Recovery Aid の削除 .....	773
例 .....	774
147 JCL/JCS の必要項目と指定例 .....	775
BS2000 .....	776
z/OS .....	792
z/VM .....	798
VSE .....	800
148 ADAREP ユーティリティ：データベースステータスレポート .....	803
149 機能概要 .....	805
150 レポート構文 .....	807
オプションパラメータ .....	808
例 .....	811
151 セーブテープ入力の処理 .....	813
プロテクションログの入力 .....	814
入力テープのチェック .....	815
同時パラメータ .....	815
デルタセーブテープのレポート .....	816
レポートレイアウト .....	816
152 レポート内容 .....	819
データベースの全般的な情報 .....	820
ファイル情報 .....	831
チェックポイント情報 .....	843
153 JCL/JCS の必要項目と指定例 .....	849
BS2000 .....	850
z/OS .....	851
z/VM .....	852
VSE .....	853
154 ADARES ユーティリティ：データベースリカバリ .....	857

155 機能概要 .....	859
ADARES の使用について .....	860
156 BACKOUT 機能 .....	865
157 BACKOUT：シーケンシャルプロテクションログ（SIBA）を使用した更新のバックアウト .....	867
基本パラメータ .....	869
オプションのパラメータとサブパラメータ .....	869
例 .....	873
158 BACKOUT DPLOG/MPLOG：デュアル／マルチプロテクションログを使用した更新のバックアウト .....	875
機能の実行 .....	876
構文 .....	877
基本パラメータ .....	878
オプションパラメータ .....	878
例 .....	882
159 CLCOPY：デュアルコマンドログのコピー .....	883
オプションパラメータ .....	884
例 .....	885
160 COPY：シーケンシャルプロテクションログまたはセーブテープのコピー .....	887
オプションパラメータ .....	889
例 .....	890
161 MERGE CLOG：ニュークリアスクラスタコマンドログのマージ .....	891
基本パラメータ .....	892
162 PLCOPY：プロテクションログのシーケンシャルデータセットへのコピー .....	893
オプションパラメータ .....	894
例 .....	896
163 REGENERATE：更新の再生成 .....	899
構文 .....	900
基本パラメータ .....	901
オプションのパラメータとサブパラメータ .....	901
例 .....	906
164 REPAIR：データストレージブロックの修復 .....	909
構文 .....	910
基本パラメータ .....	910
オプションパラメータ .....	911
例 .....	911
165 マルチスレッド処理統計値 .....	913
166 JCL/JCS の必要項目と指定例 .....	915
BS2000 .....	916
z/OS .....	924
z/VM .....	931
VSE .....	935
167 ADASAV ユーティリティ：データベースまたはファイルの保存／リストア .....	941
168 機能概要 .....	943
RESTONL 機能とRESTORE 機能 .....	944

Adabas リリースのサポート .....	945
169 RESTONL：オンラインソースからのデータベースのリストア .....	947
条件 .....	948
結果 .....	949
構文 .....	949
オプションのパラメータとサブパラメータ .....	949
例 .....	951
170 RESTONL GCB：オンラインソースからのデータベース増分のリストア .....	953
条件 .....	954
結果 .....	955
構文 .....	955
オプションのパラメータとサブパラメータ .....	956
例 .....	958
171 RESTONL FILES：オンラインソースから元の RABN へのファイルのリ ストア .....	959
条件 .....	960
結果 .....	961
構文 .....	962
オプションのパラメータとサブパラメータ .....	962
例 .....	965
172 RESTONL FMOVE 機能：オンラインソースから別の RABN へのファイルのリ ストア .....	967
条件 .....	968
結果 .....	969
構文 .....	970
オプションパラメータ .....	971
例 .....	977
173 RESTORE：オフラインソースからのデータベースのリストア .....	979
条件 .....	980
結果 .....	981
構文 .....	981
オプションパラメータ .....	981
例 .....	983
174 RESTORE GCB：オフラインソースからのデータベース増分のリストア .....	985
条件 .....	986
結果 .....	987
構文 .....	987
オプションパラメータ .....	988
例 .....	989
175 RESTORE FILES 機能：オフラインソースから元の RABN へのファイルのリ ストア .....	991
条件 .....	992
結果 .....	993
構文 .....	993
オプションパラメータ .....	994

例 .....	996
176 RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリス ストア .....	997
条件 .....	998
結果 .....	999
構文 .....	999
オプションパラメータ .....	1000
例 .....	1006
177 RESTPLOG：プロテクションログのみのリストア .....	1009
基本パラメータ .....	1010
オプションパラメータ .....	1011
例 .....	1011
178 SAVE：データベースのセーブ .....	1013
構文 .....	1015
オプションパラメータ .....	1015
例 .....	1017
179 SAVE FILES：指定ファイルのセーブ .....	1019
構文 .....	1021
オプションパラメータ .....	1022
例 .....	1024
180 JCL/JCS の必要項目と指定例 .....	1025
BS2000 .....	1026
z/OS .....	1032
z/VM .....	1037
VSE .....	1040
181 ADASEL ユーティリティ：プロテクションデータの選択 .....	1045
182 機能概要 .....	1047
183 ADASEL 構文 .....	1049
SELECT ステートメント .....	1051
追加の選択条件 .....	1052
date-time .....	1053
WITH 節 .....	1053
IF ステートメント .....	1054
value-criterion .....	1056
output-instruction .....	1061
184 グローバルパラメータによる ADASEL デフォルトの変更方法 .....	1069
185 JCL/JCS の必要項目と指定例 .....	1071
BS2000 .....	1072
z/OS .....	1073
z/VM .....	1074
VSE .....	1075
186 ADAULD ユーティリティ：ファイルのアンロード .....	1077
187 機能概要 .....	1079
188 UNLOAD FILE：指定したファイルのアンロード .....	1081
基本パラメータ .....	1082

オプションのパラメータとサブパラメータ .....	1082
例 .....	1088
189 ADAULD 入力処理 .....	1091
入力セーブテープの処理 .....	1092
190 ADAULD 出力処理 .....	1095
191 ADAULD ユーザー出口 9 .....	1097
192 JCL/JCS の必要項目と指定例 .....	1099
BS2000 .....	1100
z/OS .....	1102
z/VM .....	1104
VSE .....	1106
193 ADAVAL ユーティリティ：データベースの整合性チェック .....	1109
194 機能概要 .....	1111
195 VALIDATE：整合性チェックデータストレージとアソシエータ .....	1113
基本パラメータ .....	1114
オプションパラメータ .....	1115
196 ADAVAL 出力例 .....	1117
197 JCL/JCS の必要項目と指定例 .....	1119
ASSO、DATA、WORK データセット .....	1120
ユーザー出口を使用した照合 .....	1120
大きなファイルのソート .....	1121
BS2000 .....	1121
z/OS .....	1122
z/VM .....	1123
VSE .....	1124
198 ADAWRK ユーティリティ：ワークエリアリカバリレポート .....	1127
199 機能概要 .....	1129
200 ユーティリティ構文 .....	1131
201 レポートの解説 .....	1137
レポートの 1 ページ目 .....	1138
サマリレポート .....	1139
ファイルレポート .....	1142
トランザクションレポート .....	1146
チェックポイントレコードのレポート .....	1184
202 JCL/JCS の必要項目と指定例 .....	1185
z/OS .....	1186
VSE .....	1189
z/VM .....	1191
203 ADAZAP ユーティリティ：ASSO、DATA、WORK データセットの表示または 修正 .....	1193
204 機能概要 .....	1195
205 ADAZAP 構文 .....	1197
基本パラメータ .....	1198
オプションパラメータ .....	1198
例 .....	1199

---

206 JCL/JCS の必要項目と指定例 .....	1201
BS2000 .....	1202
z/OS .....	1203
z/VM .....	1204
VSE .....	1205
A Adabas シーケンシャルファイル .....	1207
シーケンシャルファイル表 .....	1208
オペレーティングシステムに応じた考慮事項 .....	1210
B VSE 用のライブラリおよびファイルプロシージャの例 .....	1221
Adabas のライブラリ (ADAVvLIB) .....	1222
Adabas のファイル (ADAVvFIL) .....	1223
目次 .....	1225



# 1 ユーティリティ

Adabas ユーティリティについては、それぞれ個別の章で説明します。単一機能のユーティリティについては、最初に、ユーティリティのステートメントおよび使用可能なすべてのパラメータを含む構文図を示しています。複数機能のユーティリティについては、最初に機能の概要について説明し、その後、個々の機能の構文図と説明を示しています。

各機能説明には、次の項目が含まれています。

- 構文図およびすべてのパラメータ
- 各パラメータの詳細説明（コーディング規則、制限、デフォルトなど）
- ユーティリティ機能例

機能説明に続いて、BS2000、z/OS、z/VM、VSEの各オペレーティングシステムについてジョブ制御例が示されています。



**Note:** Adabas マニュアルでは、DD 接頭辞を含まない VSE データセット名を記述する必要があるため、DD で始まるデータセット名は、データセット名の DD と残りの部分をスラッシュで区切って示しています。スラッシュはデータセット名の一部ではありません。

このドキュメントは次の部分で構成されています。

● ADAACK ユーティリティ：アドレスコンバータのチェック	ADAACK ユーティリティについて説明します。
● ADACDC ユーティリティ：変更データの取得	ADACDC ユーティリティについて説明します。
● ADACMP ユーティリティ：データの圧縮／圧縮解除	ADACMP ユーティリティについて説明します。
● ADACNV ユーティリティ：データベースの変換	ADACNV ユーティリティについて説明します。
● ADADBS ユーティリティ：データベースサービス	ADADBS ユーティリティについて説明します。
● ADADCK ユーティリティ：データストレージのチェック	ADADCK ユーティリティについて説明します。

## ユーティリティ

● ADADEF ユーティリティ：データベースの定義	ADADEF ユーティリティについて説明します。
● ADAFRM ユーティリティ：Adabas データセットのフォーマット	ADAFRM ユーティリティについて説明します。
● ADAICK ユーティリティ：インデックスおよびアドレスコンバータのチェック	ADAICK ユーティリティについて説明します。
● ADAINV ユーティリティ：インバーテッドリストの管理	ADAINV ユーティリティについて説明します。
● ADALOD ユーティリティ：ファイルローダー	ADALOD ユーティリティについて説明します。
● ADAMER ユーティリティ：ADAM 見積り	ADAMER ユーティリティについて説明します。
● ADAORD ユーティリティ：リオーダ機能	ADAORD ユーティリティについて説明します。
● ADAPLP ユーティリティ：プロテクションログ／ワークの出力	ADAPLP ユーティリティについて説明します。
● ADAPRI ユーティリティ：選択した Adabas ブロックの出力	ADAPRI ユーティリティについて説明します。
● ADARAI ユーティリティ：Adabas Recovery Aid	ADARAI ユーティリティについて説明します。
● ADAREP ユーティリティ：データベースステータスレポート	ADAREP ユーティリティについて説明します。
● ADARES ユーティリティ：データベースリカバリ	ADARES ユーティリティについて説明します。
● ADASAV ユーティリティ：データベースまたはファイルの保存／リストア	ADASAV ユーティリティについて説明します。
● ADASEL ユーティリティ：プロテクションデータの選択	ADASEL ユーティリティについて説明します。
● ADAULD ユーティリティ：ファイルのアンロード	ADAULD ユーティリティについて説明します。
● ADAVAL ユーティリティ：データベースの整合性チェック	ADAVAL ユーティリティについて説明します。
● ADAWRK ユーティリティ：ワークエリアリカバリレポート	ADAWRK ユーティリティについて説明します。
● ADAZAP ユーティリティ：ASSO、DATA、WORK データセットの表示または修正	ADAZAP ユーティリティについて説明します。
● ADABAS シーケンシャルファイル	Adabas ユーティリティで使用されるシーケンシャルファイル、およびオペレーティングシステム別のファイルの特徴およびデバイス定義について説明します。
● VSE 例のライブラリおよびファイル	VSE の各例からアクセスできるプロシージャライブラリにカタログする必要がある、Adabas のライブラリおよびファイル (ADAVvFIL) を示します。

## 2 表記規則

---

■ コントロールステートメント構文 .....	4
■ 構文形式 .....	5
■ コントロールステートメント規則 .....	7
■ パラメータ値 .....	8

このdocumentでは、次のトピックについて説明します。

- [コントロールステートメント構文](#)
- [構文形式](#)
- [コントロールステートメント規則](#)
- [パラメータ値](#)

## コントロールステートメント構文

ユーティリティのコントロールステートメントは次のフォーマットになります。

```
utility function parameter-list
```

ここでは次の内容を表しています。

ユーティリティ	実行するユーティリティの名前です。ユーティリティの名前の例として次のようなものがあります。  ADAORD ADADBS ADAINV
function	実行する特定のユーティリティ操作の名前です。例としては、次のようなものがあります。  ADAORD REORDATA ADADBS ADD ADAINV COUPLE 暗黙的な機能を持つ単一機能ユーティリティ（ADASEL、ADAULD など）には、ほとんどの場合、機能値がないか、またはオプションの機能値があります。
parameter-list	機能に続くパラメータのリストです。  リスト内のパラメータは、ほとんど常に次のフォーマットを持つキーワードです。  <i>parameter=value</i>  パラメータには1つ以上のオペランドがあり、キーワードパラメータは任意の順序で指定できます。  ほとんどのパラメータには、オペランドの値を選択または指定する必要があります。一部のオペランドは位置が固定されています（value1, value2, ..., valuex）。つまり、テキストに示された特定の順序に値が並ぶ必要があります。すべてのパラメータはコンマで区切る必要があります。

	このドキュメントのステートメント構文の説明では、パラメータを縦に並べて（上下に並べて）示すか、または縦線（ ）で区切っています。各リストには、利用可能なすべてのパラメータが示されており、ここから1つ以上を選択できるか、選択する必要があります。リストのパラメータはコンマで区切る必要がありますが、パラメータが上下に並べて示されているときは、これらのコンマは構文ステートメントでは省略されています。
--	---

## 構文形式

次の表は、Adabas ステートメントの構文図で使用される形式を示しています。

表記規則	説明	例
大文字、 太字	太字の大文字で表示される構文要素は Adabas キーワードです。これらのキーワードを指定する際に、表示されているとおりに入力する必要があります。	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>ADADBS CHANGE FILE = file-number</b> </div> <p>構文要素 ADADBS、CHANGE、および FILE は Adabas キーワードです。</p>
小文字、 斜体、標準 フォント	斜体のかかった標準フォントの小文字で表示される構文要素は、ユーザーが入力すべき項目を表します。	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>ADADBS CHANGE FILE = file-number</b> </div> <p>構文要素 <i>file-number</i> では、ユーザーがどのような種類の値を入力する必要があるかを表しています。この例では、ADADBSCHANGE オペレーションに関するファイルの数を入力する必要があります。</p>
大文字と 小文字が 混在、標準 フォント	大文字と小文字が混在した標準フォント（太字でも斜体でもない）で表示される構文要素は、他の Adabas コントロールステートメントにより設定される項目です。この表記は通常、Adabas の構文で、パラメータのデフォルト値をどのように決定するのかを表します。	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>[SORTDEV = { device-type   ADARUN-device }]</b> </div> <p>構文要素 "ADARUN-device" は、別のデバイスタイプが指定されない場合は ADARUN DEVICE パラメータが表すデバイスタイプが使用されることを示します。リテラル "ADARUN-device" を、SORTDEV パラメータに指定することはできません。</p>

表記規則	説明	例
下線	<p>下線は次の2つの目的に使用されます。</p> <ol style="list-style-type: none"> <li>1. 該当する場合は、デフォルト値を表します。それ以外の場合は、付帯的なパラメータの記述の中でデフォルト値が示されます。</li> <li>2. キーワードの短縮形を表します。</li> </ol>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>[LRECL = {record-buffer-length   4000 }]</code> </div> <p>上の例では、4000が、他のレコードバッファ長を指定していない場合に使用される LRECL パラメータのデフォルト値です。</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>DEVICE</code> </div> <p>上の例では、DEVICE パラメータの短縮形は DE です。</p>
縦線 ( )	<p>縦線は、相互に排他的な選択肢の区切りを表します。</p> <p><b>注意:</b> 角カッコや中カッコを使用した、より複雑な構文では、相互排他的な選択肢が入れ子になって記述されます。</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>ADAORD { RESTRUCTUREF   REF }</code> </div> <p>上の例では、この ADAORD 関数に RESTRUCTUREF または REF を選択する必要があります。デフォルト値はありません。</p>
角カッコ ([ ])	<p>角カッコは、選択型の要素を表します。角カッコ内で複数の要素が、入れ子になって記述されたり、縦線で区切られていたりするときには、その中の1つだけを使用できます。</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>SORTSEQ = { descriptor [ ,MU ] [ ,NU ] ISN [ , STARTISN = value ] physical-sequence }</code> </div> <p>上の例では、SORTSEQ パラメータおよび MU、NU、STARTISN の各サブパラメータが選択肢です。</p> <p><b>注意:</b> SORTSEQ パラメータの相互排他的な選択肢は、入れ子になって記述されています。</p>
中カッコ ( { } )	<p>中カッコは、必須の要素を表します。角カッコ内で複数の要素が、入れ子になって記述されたり、縦線で区切られていたりするときには、その中の1つだけを使用する必要があります。</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>SUSPEND [TTSYN = {time-available-to-sync   ADARUN-TT }] [TRESUME = {time-until-resume   120 }] RESUME</code> </div> <p>この例では、SUSPEND および RESUME パラメータのいずれかが必要です。</p>
字下げ	<p>字下げは、パラメータのサブパラメータを表します。</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>SUSPEND [TTSYN = {time-available-to-sync   ADARUN-TT }] [TRESUME = {time-until-resume   120 }] RESUME</code> </div> <p>この例では、TTSYN および TRESUME が SUSPEND パラメータのサブパラメータです。</p>

表記規則	説明	例
省略記号 (...)	省略記号は、繰り返して使用できる要素を表します。他の区切り文字が省略記号の前に記述されている場合は、その区切り文字を使用して、繰り返される要素を区切る必要があります。	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">[FIELD = 'field-name [ , option]... ' ] ...</div> <p>この例では、FIELD パラメータを繰り返して使用できます。また、このフィールドに関連するオプションも繰り返して使用できます。</p>
その他の区切り文字と記号	その他のすべての区切り文字と記号は、表示されたとおりに入力する必要があります。	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">[FIELD = 'field-name [ , option]... ' ] ...</div> <p>この例では、フィールド定義とそれに関連するオプションの前後に単一引用符を指定する必要があります。また、それらのオプションをコンマで区切る必要があります。</p>

## コントロールステートメント規則

ユーティリティコントロールステートメントの構造には、次の規則が適用されます。

1. 各コントロールステートメントの第1～6桁にはユーティリティ名を指定する必要があります。
2. ユーティリティ名とその後に続く機能名との間には、1つ以上の空白を指定する必要があります。
3. 各キーワードパラメータエントリおよびキーワードエントリ中の複数の値は、コンマで区切る必要があります。
4. 等号記号(=)の前後に空白を指定することはできません。
5. ステートメントの最後のパラメータエントリの後にコンマを付けるかどうかは任意です。
6. コントロールステートメントの処理は、72桁目またはパラメータリスト開始後に空白が現れた時点で終了します。73桁から80桁のエントリは処理されません。
7. 1桁目にアスタリスク(\*)のあるステートメントはコメントとみなされ、処理されません。
8. 複数のコントロールステートメントを継続して指定するには、第1～6桁に指定したユーティリティ名の後に少なくとも空白を1つ以上指定し、その後に、新しいコントロールステートメントと追加パラメータを指定します。

## パラメータ値

パラメータの等号記号 (=) の後に、変数（上記の例およびこのドキュメントでは斜体で表現されています）の値を実際に指定する場合は、次の構文に従います。

```
parameter = value
parameter = value-list
parameter = value-range
```

ここで、*value* については次のセクションで説明します。パラメータ *value-list* および *value-range* は *value* の変形であり、テキストで説明されている個々のパラメータ規則に応じて *value* の代わりに使用したり、一緒に使用することができます。

### value

*value* パラメータは、英数字または 16 進数の数字または文字列で構成できます。オプションのキーワードパラメータの一部では、パラメータが指定されない場合にデフォルト値が使用されます。

#### 英数字値

英数字値は次のいずれかの方法で指定します。

値の構成要素	前後のアポストロフィ
大文字、小文字、数字、マイナス符号 (-) のみ	任意
アポストロフィ自体を含むその他の文字（アポストロフィは2回入力する必要があります）	必須

#### 数値

数値は次のように指定します。

表現する値	指定する内容
ブロックまたはシリンダの数	ブロックを指定するときは、値の直後に文字 B を指定する必要があります。それ以外の場合はシリンダであるとみなされます。  SIZE=200B（ブロック 200 個） SIZE=200（シリンダ 200 個）
Adabas ファイル	1~4 桁の数字（先行ゼロを使用可能）  FILE=3

表現する値	指定する内容
	FILE=03 FILE=162
デバイスタイプ	使用するデバイスタイプのモデル番号に対応する 4 桁の数字  DEVICE=3380
フィールド名またはディスクリプタ	フィールド名またはディスクリプタに対応する 2 文字のフィールド名  FIELD1=NA

パラメータ記述で指定されている場合は、16 進値も受け付けられます。16 進値は、インジケータ X に続けてアポストロフィで囲む必要があります。

```
X'0002DC9F'
```

### value-list

value,... (数値)

```
BITRANGE=2,10,2
```

または

'value,...' (英数字値)

```
UQDE='AA,AC,AE'
```

### value-range

value - value, ...

```
ISN=600-900,1000-1200
```

値リストまたは値範囲内の個々の値は、対応するパラメータに指定された値に関連付けられていることがあり、このような場合に、位置が固定されていることがあります。例としては、次のようなものがあります。

```
ADADBS UNCOUPLE FILES=13,20,PASSWORD='PW13,PW20'
```

ADADBS UNCOUPLE 機能に対して、パスワードで保護されているファイル 13 および 20 のカップリングを解除するように指示します。

パスワード (PASSWORD パラメータで指定) は、FILES パラメータ内のそれぞれ対応するファイルと同じ順序で指定する必要があります。

ファイル 13 がパスワード保護されていない場合は、PASSWORD パラメータも、プレースホルダコマを使用して次のように指定する必要があります。

```
... PASSWORD=' ,PW20'
```

パスワード "PW20" を、FILES 値リストのファイル 20 に対応する位置に置きます。または、FILES でファイル 20 を最初に指定する必要があります。

# 3 ADAACK ユーティリティ：アドレスコンバータの チェック

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ACCHECK：データストレージに対するアドレスコンバータのチェック
- JCL/JCS の必要項目と指定例



## 4 機能概要

---

ADAACK では、特定ファイルまたはファイル範囲、あるいは特定 ISN または ISN 範囲についてアドレスコンバータをチェックします。スパンドレコードが使用中の場合、ADAACK では、渡されたすべての ISN をプライマリ ISN とみなし、それに応じて処理を実行します。範囲には、すべてのファイルまたはすべての ISN を含めることができます。通常、ADAACK は ADAICK と組み合わせて使用します。

チェック対象のファイルでスパンドレコードが有効になっている場合、セカンダリ ISN をセカンダリレコードの RABN にマッピングするために使用されるセカンダリアドレスコンバータも自動的にチェック対象となります。スパンドレコードの詳細は、「スパンドレコード」を参照してください。

ADAACK では各アドレスコンバータエレメントをチェックし、ファイルコントロールブロック (FCB) 内で指定されたデータストレージエクステンツの使用部分内に、データストレージ RABN があるかどうかを判別します。

ADAACK では各データストレージブロック (指定 ISN 範囲内) 内の各レコードの ISN をチェックし、その ISN に対するアドレスコンバータエレメントに正しいデータストレージ RABN が含まれていることを確認します。このことは、次のように実行されます。

1. 指定 ISN 範囲 (ISN 数の 4 倍) のメインメモリが割り当てられます。範囲の指定がない場合、全範囲 (MINISN から TOPISN まで) がチェックされます。

アドレスコンバータがデータベースからメモリ内のこのエリアに読み込まれます。

2. 使用されている各データストレージブロックが (FCB のデータストレージエクステンツに従って) 読み込まれ、メモリ内のアドレスコンバータに対してチェックされます。アドレスコンバータ内の各 ISN には、それに対応する正しいデータストレージレコードが正確に 1 つある必要があります。
3. データストレージ内に存在しなかった ISN は、メモリ内のアドレスコンバータにないかどうかチェックされます。

ファイルが大きい場合は、ADAACK の実行に長い時間がかかることがあります。ADAACK では、20 データストレージブロックを処理するたびに、メッセージ行を出力します。

使用されているすべてのデータストレージブロックが読み込まれるため、ISN 範囲は実行時間に影響を及ぼしません。



### Notes:

1. ADAACK 実行時には、Adabas ニュークリアスがアクティブである必要はありません。
2. 保留中の自動再スタート条件は無視されます。
3. ADAACK では、並行して更新処理を行う場合、ニュークリアスとの同期は取りません。
4. このユーティリティの用途は、診断のみにしてください。

ADAACK でエラーが発生すると、常にコンディションコード 8 が返されます。

# 5 ACCHECK：データストレージに対するアドレスコン バータのチェック

---

▪ オプションパラメータ .....	16
▪ 例 .....	17

```
ADAACK ACCHECK [FILE={ file | file - file | all-files }]
                [ISN={ isn - isn | all-isn }]
                [NOOPEN]
                [NOUSERABEND]
```

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

---

### FILE：チェック対象ファイル

チェック対象のファイル、ファイル範囲、またはすべてのファイルを指定します。デフォルトでは、データベース内の全ファイルがチェック対象となります。

### ISN：チェック対象 ISN 範囲

チェック対象の ISN 範囲またはすべての ISN を指定します。デフォルトでは、全範囲（MINISN から TOPISN まで）がチェック対象となります。

スパンドレコードが使用中の場合、ADAACK は、渡されたすべての ISN をプライマリ ISN とみなし、それに応じて処理を実行します。ISN がスパンドデータストレージレコードのプライマリ ISN の場合、Adabas では、セカンダリアドレスコンバータ（AC2）内のスパンドレコードについて、適切なセグメントレコードを自動的にチェックします。

特定の ISN についてのエラー情報を出力する場合、ADAACK ユーティリティでは、レコードがスパンされている場合に、問題がプライマリ ISN とセカンダリ ISN のどちらかなのかを示すようになりました。

### NOOPEN：オープン再同期の回避

通常、ADAACK では、開始するときにニュークリアスに対してユーティリティの OPEN コールを発行し、影響を受けるファイルのブロックが、ニュークリアスバッファプール内に残っていないことを確認します。しかし、これはまた他のユーザーに対してそのファイルをロックすることにもなります。NOOPEN を指定すると、ADAACK によって OPEN コールが発行されなくなります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

## 例

---

### 例 1

```
ADAACK ACCHECK
```

データベース内の全ファイルをチェックします。

### 例 2

```
ADAACK ACCHECK FILE=12,ISN=1-8000
```

ファイル 12 の ISN1 から 8000 までをチェックします。

### 例 3

```
ADAACK ACCHECK FILE=8-10
```

ファイル 8 から 10 までの全 ISN をチェックします。



## 6 JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	20
▪ z/OS .....	21
▪ z/VM .....	22
▪ VSE .....	23

ここでは、BS2000、z/OS、z/VM、およびVSEシステムにおける ADAACK の実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAACK パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT DDPRINT		メッセージおよびコード
ADAACK メッセージ	SYSLST DDDRUCK		メッセージおよびコード

### ADAACK JCL の例 (BS2000)

SDF フォーマットの場合：

```

/.ADAACK LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A A C K ADDRESS CONVERTER CHECK
/REMARK *
/REMARK *
/ASS-SYSLST L.ACK.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAyyyyy.DATA,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAACK,DB=yyyyy,IDTNAME=ADABAS5B
ADAACK ACHECK FILE=ffff
/LOGOFF SYS-OUTPUT=DEL
    
```

ISP フォーマットの場合：

```

/.ADAACK LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A A C K ADDRESS CONVERTER CHECK
/REMARK *
/REMARK *
/SYSFILE SYSLST=L.ACK.DATA
/FILE ADAvrs.MOD ,LINK=DDLIB
/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAACK,DB=yyyyyy,IDTNAME=ADABAS5B
ADAACK ACHECK FILE=ffff
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ADAACK メッセージ	DDDRUCK	プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAACK パラメータ	DDKARTE	リーダー	

### ADAACK JCL の例 (z/OS)

```

//ADAACK JOB
//*
//* ADAACK:
//* ADDRESS CONVERTER CHECK
//*
//ACK EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAACK,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*

```

```
//DDKARTE DD *
ADAACK ACHECK FILE=ffff
/*
```

この例については、MVSJOBS データセットの ADAACK を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ADAACK メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAACK パラメータ	DDKARTE	ディスク/端末/リーダー	

### ADAACK JCL の例 (z/VM)

```
DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDPRINT,DSN=ADAACK.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAACK.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNACK.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAACK.CONTROL,MODE=A
ADARUN
```

### RUNACK CONTROL A1 の内容

```
ADARUN PROG=ADAACK,DEVICE=dddd,DB=yyyyy
```

### ADAACK CONTROL A1 の内容

```
ADAACK ACHECK FILE=ffff
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	
データストレージ	DATARn	ディスク	*	
ADAACK メッセージ		プリンタ	SYS009	メッセージとコード
ADARUN メッセージ		プリンタ	SYSLST	メッセージとコード
ADARUN パラメータ	CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADAACK パラメータ		リーダー	SYSIPT	

\* すべてのプログラマ論理ユニットを使用できます。

### ADAACK JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

```
* $$ JOB JNM=ADAACK,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
* ADDRESS CONVERTER CHECK
// JOB ADAACK
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAACK,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAACK ACHECK FILE=27
/*
/&
* $$ EOJ
```

この例については、メンバ ADAACK.X を参照してください。



# 7 ADACDC ユーティリティ：変更データの取得

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ユーティリティの実行
- オペレーティングシステムに応じた考慮事項
- ADACDC ユーザー出口
- 例
- JCL/JCS の必要項目と指定例



## 8 機能概要

---

■ 処理フェーズと結果ファイル .....	28
■ プライマリ入力データ .....	31
■ プライマリ出力ファイル .....	32
■ トランザクションファイル .....	33

ADACDC ユーティリティでは、1つ以上のシーケンシャルプロテクションログを入力として使用し、入力プロテクションログがカバーしている期間中に加えられたすべてのデータベース変更のデルタを含むプライマリ出力ファイルを最終的に出力として生成します。変更のデルタとは、該当期間中に各 ISN に加えられた最新の変更のことであり、プライマリ出力ファイルに出力されます。



### Notes:

1. ADACDC の実行時に ISN パラメータが指定されている場合、デルタは生成されません。代わりに、個別の削除トランザクションと挿入トランザクションがそれぞれプライマリ出力ファイルに書き込まれ、変更のデルタは生成されません。
2. 現時点では、スパンドレコードは ADACDC ユーティリティでサポートされていません。ただし、IGNORESPANNED パラメータが ADACDC の実行時に指定された場合、ADACDC 処理でスパンドレコードは無視され、警告メッセージが発行され、処理が続きます。リターンコード "4" が返されます。

ADACDC の実行時に出力されたプライマリ出力ファイル内のデータはデータウェアハウスの代入プロセスの入力として定期的に変更され、データベース全体のコピーではなく、データベースに対する変更がデータウェアハウスデータベースに適用されます。これにより、データウェアハウスがより頻繁に、より短時間で更新されるため、格納される情報の精度を高めることができます。

ADACDC ユーティリティを実行するには、次の要件を満たしている必要があります。

- 外部ソースが利用可能であり、オペレーティングシステムに標準ソースとしてインストールされている必要があります。詳細は、「[オペレーティングシステムに応じた考慮事項](#)」を参照してください。
- さらに、外部ソースから、処理対象のレコードが属すファイルの FDT を含むデータベースアソシエータにアクセスできる必要があります。

ADACDC は、このソートパッケージを使用して ISN 順に出力を生成します。したがって、すべての変更が ISN 順にプライマリ出力ファイルに書き込まれます。

このchapterでは、次のトピックについて説明します。

## 処理フェーズと結果ファイル

---

ADACDC では、2つのフェーズでシーケンシャルプロテクションログを処理します。フェーズ 1 とフェーズ 2 は、別々に実行することも両方を同時に実行する（デフォルト）こともできます。

- フェーズ2を個別に実行する場合、または両フェーズを同時に実行する場合には、データは圧縮解除されてプライマリ出力ファイルに書き込まれます。

- フェーズ1のみを実行する場合は、データは抽出ファイルに書き込まれます。次に、この抽出ファイルをフェーズ2で複数回処理することによって、レコードが圧縮解除されてプライマリ出力ファイルに書き込まれます。

プライマリ出力ファイル内のデータレコードは非圧縮形式ですが、抽出ファイル内のデータレコードは圧縮形式になります。これらの形式の詳細は、『*Adabas ユーティリティマニュアル*』のADACMPの圧縮／圧縮解除に関するセクションを参照してください。

プライマリ出力ファイルと抽出ファイルは、可変長レコードを含む標準ファイルであり、オペレーティングシステムに依存します。

## フェーズ1と抽出ファイル

フェーズ1で、プロテクションログからの更新が分析され、CDCE と呼ばれる標準構造の接頭辞が付けられます。ファイルのレコードは、定数のCDCE接頭辞の後に、圧縮されたレコード情報が含まれているという形式となっています。

通常、これらのレコードは、ファイルの各ISNの最新の更新を確定するために外部ソートルーチンに渡されます。特定のファイルとISNの組み合わせの最新の変更のみが、抽出ファイルに書き込まれます。ただし、ADACDCの実行時にISNパラメータが指定されている場合、更新は引き続きISN順でソートされますが、集約は行われません。代わりに、ISNのすべての変更トランザクションが抽出ファイルに記録されます。

フェーズ1を個別に実行すると、抽出ファイルが作成されます。このファイルを使用すると、PLOGデータを一度処理することによって、必要に応じて、ファイル選択条件などに基づいて複数のプライマリ出力ファイルを生成できます。このオプションは、目的に応じて異なるファイル変更が必要となる場合に役立ちます。

フェーズ1処理の実行中に、出力用に抽出ファイルがオープンされます。レコードがソート処理から出力されると、次のいずれかの条件を満たしている場合に、各ファイルとISNの組み合わせに対する更新内容が抽出ファイルに書き込まれます。

- 更新がETユーザーによるもので、完了トランザクションに属していること
- 更新がEXUユーザーによって行われており、完了コマンドに属していること
- NOETが指定されていること

指定されたファイルを制御するユーティリティ処理がない場合は、期間中の上記以外のファイルとISNの組み合わせに対する全更新が破棄されます（「[プライマリ出力ファイルへのチェックポイントの書き込み](#)」を参照）。

 **Note:** ADACDCユーザー出口（後述の説明参照）がファイルとISNの既存の組み合わせでレコードを追加する場合、同一ファイル上に重複した組み合わせを持つことができません。ユーザー出口によって追加または修正されたレコードについては、そのことがCDCE構造にマークされます。

## フェーズ2または両フェーズとプライマリ出力ファイル

ADACDCの両フェーズを同時に実行するとき、またはフェーズ2処理だけを実行するとき、プライマリ出力ファイルが使用されます。

- 両方のフェーズを同時に実行する場合は、ソート処理の出力を使用して、プライマリ出力ファイルが直接オープンされ、作成されます。この場合、処理はフェーズ1処理で作成された抽出ファイルに対して行われます。
- フェーズ2のみを実行するときは、プライマリ出力ファイルは抽出ファイルからの入力を使用して作成されます。

ファイルの各レコードは、接頭辞 CDCO の後に、圧縮解除されたレコード情報が含まれるという形式となっています。何らかの理由でレコードを圧縮解除できない場合は、警告メッセージが発行され、圧縮レコードがプライマリ出力ファイルに書き込まれます。レコードの圧縮解除がエラーになったときは、CDCO 構造のフラグでユーザープログラムに通知されます。

 **Note:** ADACDC ユーザー出口（後述の説明参照）がファイルと ISN の既存の組み合わせでレコードを追加する場合、同一ファイル上に重複した組み合わせを持つことができません。ユーザー出口によって追加または修正されたレコードは、CDCO 構造にマークされます。

## プライマリ出力ファイルへのチェックポイントの書き込み

ADACDC ユーティリティの主要な用途は、一定の期間内に修正されたファイル内の各 ISN について、最新の変更内容をまとめた出力ファイルを提供することです。実行時に ISN パラメータが指定されている場合、ADACDC ユーティリティの主要な用途は、一定の期間内に修正されたファイル内の各 ISN の変更内容を含む出力ファイルを提供することです。

ファイルの内容は、単純に変更される以外に、ユーティリティが実行されることによって、根本的に変更される場合があります。例えば、ファイルが削除されると、ファイル内の全 ISN も削除されます。この場合、単に ISN の最新の変更内容を提供しても、ファイルの状態は正確に表されません。

このため、次のチェックポイントが必要に応じてプライマリ出力ファイルに記録され、書き込まれます。この場合、出力レコードには関連するインジケータが付けられます。

ADASAV RESTORE FILE	File created
ADAORD STORE FILE	File created
ADALOD LOAD FILE	File created
ADALOD UPDATE FILE	File updated

ADADBS DELETE FILE	File deleted
ADADBD REFRESH FILE	File deleted

これらの処理はファイルとファイルの状態に根本的な影響を及ぼすことがあるため、他の更新に関係する処理が行われたときは、チェックポイントがプライマリ出力ファイルに書き込まれません。

ADACDC では、上記の各チェックポイントまで全 ISN の最新の更新内容を保持します。つまり、プライマリ出力ファイルに複数のチェックポイントが書き込まれる場合は、ファイルと ISN の 1 つの組み合わせが複数回記録される場合があります。ファイルを削除、再作成、または大量更新する前にファイルのビューを完成し、そのコピーを作成することが要求される多くのデータウェアハウスパッケージに対して、このことが役立ちます。

## プライマリ入力データ

プライマリ入力データは、データベースから直接作成されたか、または ADARES PLCOPY によって非シーケンシャルプロテクションログから作成された、シーケンシャルプロテクションログで構成されます。

 **Note:** 現時点では、スパンドレコードは ADACDC ユーティリティでサポートされていません。ただし、IGNORESPANNED パラメータが ADACDC の実行時に指定された場合、ADACDC 処理でスパンドレコードは無視され、警告メッセージが発行され、処理が継続します。リターンコード "4" が返されます。

ADACDC では、このデータを処理することによって、次のことが保証されます。

- 読み込まれた新 PLOG ブロックの PLOG 番号が変わっていない場合は、新 PLOG ブロック番号は前の PLOG ブロック番号に 1 を加算した番号になります。
- PLOG 番号自体が変わっている場合は、新 PLOG 番号は前の PLOG 番号より大きくなり、新 PLOG ブロック番号は 1 になります。

 **Note:** PLOG 番号が変わって前との番号差が 1 より大きい場合は、警告メッセージが発行され、処理が継続されます。オンラインセーブを使用しているときにこのような状態になりますが、エラーではありません。

これらのチェックのいずれかでエラーが発生すると、ユーティリティの実行は終了します。

## プライマリ出力ファイル

プライマリ出力ファイルは、入力プロテクションログがカバーしている期間中に追加、更新、または削除された全データベースレコードを含むシーケンシャルファイルです。

レコードが複数回変更された場合、最後の変更のみが出力ファイルに出力されます。ADACDCは、ソート処理を使用して、同一レコードに対する複数回の変更を識別します。ただし、ADACDCの実行時にISNパラメータが指定されている場合、ISNのすべての変更がプライマリ出力ファイルに出力されます。ADACDCは、引き続きソート処理を使用して、プライマリ出力ファイルをISN順に書き込みます。

各プライマリ出力ファイルのレコードは、固定長のレコード接頭辞と、その後続く非圧縮形式のデータベースレコードで構成されます。非圧縮データの形式は、**ADACMP DECOMPRESS**機能の出力に対応します。

プライマリ出力レコードの接頭辞はCDCO DSECTで記述されます。構造は次のとおりです。

バイト	説明	
0~1	レコード長 (2進数)	
2~3	ゼロ	
4~7	定数 'CDCO'	
8~9	データベース ID	
10~11	ファイル番号	
12~15	更新レコードの ISN	
16~19	非圧縮データ長 (バイト単位)	
20~47	最新のレコード更新ユーザーのコミュニケーション ID (28 バイト)	
48	変更インジケータ	
	X'04'	レコードの追加
	X'08'	レコードの更新
	X'0C	レコードの削除
	X'10'	ファイルの生成
	X'14'	ファイルの更新
49	フラグ (ビットごとに設定)	
	X'80'	ユーザー出口によるレコードの追加
	X'40'	ユーザー出口によるレコードの修正
	X'20'	レコードの非圧縮エラー (圧縮のまま)
50	データベースバージョンインジケータ	

バイト	説明
51	予約済み（将来的に使用される予定）
52～59	4 バイトの STCK（4 バイトの 16 進カウンタが続く）ユーザーは、必要に応じて、この 8 バイトのフィールドでソートして、プライマリ出力ファイルのレコードを PLOG 順に戻すことができます。この処理が必要な場合についての詳細は、この section の「 <a href="#">ADACDC での ISNREUSE の使用</a> 」を参照してください。
60～67	予約済み（将来的に使用される予定）
68-...	非圧縮レコードデータ

## トランザクションファイル

ADACDCでは、入力データチェック結果を複数回の実行をまたいで保持するために、処理する最新のデータベース ID、PLOG 番号、および PLOG ブロック番号を含む情報をトランザクションファイルのトランザクション制御レコードに保存します。RESETTXF オプションの指定がない場合、この情報は最新の入力を確認するために使用されます。「[RESETTXF：入力トランザクションファイルのリセット](#)」（[ADACDC オプションパラメータに関するセクション](#)）を参照してください。

実際には ADACDC は 2 つの異なるトランザクションファイル、すなわち入力トランザクションファイルと出力トランザクションファイルを識別します。どちらのトランザクションファイルも可変長レコードを含む標準ファイルであり、オペレーティングシステムに依存します。

### 入力トランザクションファイルの処理

入力処理の段階で、ADACDC は、ソートプログラムへの入力トランザクションファイルを処理します。

入力トランザクションファイルの制御レコードに従って、0 以上のレコードが検索されます。これらのレコードは、ユーティリティの前の実行で完了していないトランザクションに関するデータベース更新です。これらのレコードは入力の一部として再処理され、通常、ユーティリティの次回実行時にシーケンシャルプロテクションログを指定すると、該当トランザクションは完了状態になります。したがって、プロテクションログ内の順序は非常に重要です。順序が正しくないと、更新は未処理のままになります。

トランザクションファイルには、最新のフェーズ 1 の実行時に NOET オプションが指定されたかどうかも記録されます。ADACDC では、このオプションが前回実行したとき以降変更されたことを検出すると、入力トランザクションファイルの制御レコードの情報を使用します。ただし、他のレコードのトランザクション情報はすべて無視されます。これは、NOET が変更されると、プライマリ出力ファイルまたは抽出ファイルに書き込まれるデータに矛盾が発生する場合があります。この状況が発生すると、ADACDC から警告が発行されます。

### 出力トランザクションファイルの処理

ソートプログラムからの出力処理が開始されると、入力トランザクションファイルがクローズされ、出力トランザクションファイルがオープンされます。そして、出力トランザクションファイルに制御レコードが書き込まれ、その後、未完了トランザクションまたは未完了コマンド（NOET オプションが有効な場合、または EXU ユーザーの場合）に関係する更新が書き込まれます。処理が終了すると、トランザクションファイルがクローズされます。

### 単一トランザクションファイルの使用

入力トランザクションファイルと出力トランザクションファイルとして同一ファイルを使用できます。ただし、出力トランザクションファイルへの書き込み時（すなわち、ソートユーティリティからの出力の処理中）にエラーが発生すると、この時点では入力トランザクションファイルが存在しないため、期待する結果を得ることができません。

このため、ユーティリティを実行する前に、エラーが発生した場合にリストアできるように、トランザクションファイルをバックアップしておく必要があります。

または、プログラムによるファイルの更新時には常にファイルの新バージョンを作成する機能がオペレーティングシステムに備えられている場合、その機能を使用して新バージョンを作成することもできます。

# 9 ユーティリティの実行

---

■ オプションパラメータ .....	36
■ ADACDC での ISNREUSE の使用 .....	38

```
ADACDC [FILE= filelist ]  
        [IGNORESPANNED]  
        [ISN]  
        [NOET]  
        [PHASE={ 1 | 2 | BOTH }]  
        [RESETTXF]
```

ADACDCユーティリティを初めて実行する場合、下記の構文を使用して、次回以降の実行の入力のために有効なトランザクションファイルを作成する入力トランザクションファイル（CDCTXI）を指定しないでおくか、またはダミーを指定してください。

```
ADACDC RESETTXF,PHASE=BOTH
```

RESETTXF オプションを指定すると、存在しないまたはダミーの入力トランザクションファイルが無視され、プライマリ入力データが読み込まれ、入力データに基づいてプライマリ出力が生成されます。

初回実行時に入力トランザクションファイルが作成された後は、このユーティリティの実行にはユーティリティ名ADACDCのみが必要になります。PHASEパラメータのデフォルト値はBOTHです。パラメータオプションについては、以下で説明されています。

## オプションパラメータ

---

### FILE：処理するファイル

FILEパラメータを使用すると、ユーティリティによって処理され、出力ファイルに書き込まれるファイルを制限できます。

- PHASE=1 の場合は、指定ファイルに関するレコードのみが抽出ファイルに書き込まれます。
- PHASE=2 または BOTH の場合は、指定ファイルに関するレコードのみがプライマリ出力ファイルに書き込まれます。



**Note:** PHASE=2 のときは、PHASE=1 で作成された抽出ファイルを入力として指定する必要があります。

このパラメータを指定しない場合、ユーティリティによって全ファイルが処理されます。

### IGNORESPANNED：スパンドレコードの無視

現時点では、スパンドレコードはADACDCユーティリティでサポートされていません。ただし、IGNORESPANNEDパラメータがADACDCの実行時に指定された場合、ADACDC処理でスパンドレコードは無視され、警告メッセージが発行され、処理が継続します。リターンコード "4" が返されます。

**ISN**：削除トランザクションと挿入トランザクションを別々に記録

ADACDC 処理では、通常、同じ ISN のすべての削除トランザクションと挿入トランザクションが統合され、その ISN の更新トランザクションが1つ生成されます。ただし、ISN パラメータを指定すると、削除トランザクションと挿入トランザクションは、それぞれ個別にプライマリ出力ファイル (CDCOUT) に記録されます。このため、ISN パラメータを指定して ADACDC ユーティリティを使用すると、プライマリ出力ファイルで生成されるレコード数が大幅に増加する場合があります。

**NOET**：ET 処理しない

通常、ADACDC では、完了トランザクションの一部であるレコードまたは完了コマンドの一部であるレコード (EXU ユーザーの場合) のみが処理されます。

PHASE=1 または PHASE=BOTH のときに NOET オプションを指定すると、完了トランザクションは処理されません。入力、NOET オプション付きまたはなしでプロテクションログ (PLOG) をすでに処理したフェーズ1からの抽出ファイルであるため、PHASE=2 の場合には NOET は効果がありません。

NOET オプションを指定した場合、データベースに加えられた変更が PLOG で検出されると、この変更内容は即時に処理され、抽出ファイル (PHASE=1) またはプライマリ出力ファイル (PHASE=BOTH) に書き込まれます。

 **Caution:** このオプションを指定すると、入力 PLOG の終了時点で完了していないトランザクションに関する更新が、プライマリ出力ファイルに書き込まれます。

**PHASE**：実行フェーズ

PHASE パラメータでは、ユーティリティで必要となる入力および生成する出力を決定します。

- PHASE=1 の場合は、シーケンシャル PLOG 入力を読み込まれ、PHASE=2 で処理される中間抽出ファイルが生成されます。
- PHASE=2 の場合は、PHASE=1 で生成された抽出ファイルを読み込まれ、ファイル変更のデルタを含むプライマリ出力ファイルが生成されます。
- PHASE=BOTH (デフォルト) の場合は、シーケンシャル PLOG 入力を読み込まれ、変更のデルタを含むプライマリ出力ファイルが直接生成されます。このとき、抽出ファイルの読み込みおよび書き込みは行われません。

詳細は、「[処理フェーズと結果ファイル](#)」セクションを参照してください。

**RESETXF**：入力トランザクションファイルのリセット

ADACDC は、PLOG が PLOG ブロック／番号順に読み込まれるように、プライマリ入力データをチェックします。チェックエラーが発生すると、ユーティリティの実行は終了します。

ADACDC は、チェック結果を複数回の実行にまたがって維持するために、入力トランザクションファイルおよび出力トランザクションファイルを保存します。これらのファイルは、ユーティリティを次回実行するとき、未完了のトランザクションまたはコマンド (EXU ユーザーの場合) に関するレコード更新を追跡するために使用されます。通常、このような未

完了のトランザクションまたはコマンドは、次のシーケンシャルPLOGを指定してユーティリティを実行すると、完了します。

ただし、PLOGをPLOG順序どおり以外に処理する必要が発生し、トランザクションファイルの情報を削除しても安全な場合は、RESETTXF オプションを使用してトランザクションファイルのリセットできます。この場合、今回の実行ではチェックは行われず、未完了のトランザクションおよびコマンドはすべて無視されます。ADACDCでは、入力トランザクションファイルの情報は無視され、ジョブの終了時に出力トランザクションファイルに書き込まれます。



**Caution:** PLOGシーケンスが中断された場合、トランザクションファイルに記録されている未完了トランザクションに関するレコード更新は、そのまま保持されます。

## ADACDCでのISNREUSEの使用

---

標準のADACDC処理では、プライマリ出力ファイルはISN順で生成されます。通常、この処理は正常に行われます。ただし、ADADBSまたはADALOD ISNREUSEオプションを指定してデータベースファイルを作成した場合、エラー（レスポンス98）が発生することがあります。これらのエラーは、ISNが再利用されたことにより、異なるISNで同一のユニークディスクリプタキー（UQ）が使用された複数のトランザクションがPLOGに存在する可能性がある場合に発生します。

▶ **手順9.1.** これらの問題を解決するには、次の手順を実行する必要があります。

- 1 ISNパラメータを指定してADACDCユーティリティを実行します。これにより、プライマリ出力ファイルには、ISN別に変更がまとめられる代わりに、変更の詳細なリストが出力されるようになります。このように実行した後も、プライマリ出力ファイル内のデータはISN順になります。
- 2 このデータをデータウェアハウスまたはその他のアプリケーションに適用する前に、プライマリ出力ファイル（CDCOUT）をPLOG順にソートします。このソートは、CDCOUTデータに対して8バイトのオフセット52で実行する必要があります。

データがPLOG順にソートされたら、そのデータをデータウェアハウスまたはその他のアプリケーションに適用することができます。

- 3 他のADACDCジョブを実行する前に、トランザクションファイル（CDCTXI/CDCTXO）をPLOG順にソートし直します。この処理を行わない場合は、ADACDCの今後の実行に影響を及ぼす可能性があります。

トランザクションファイル（CDCTXI/CDCTXO）を正しい順序にするには、実際にはソートを2回実行する必要があります。その手順を次に示します。

1. 1回目のソートでは、CDCEレコードをPLOG順にソートします。このソートは、4バイトのオフセット68、4バイトのオフセット16の順で実行する必要があります。例としては、次のようなものがあります。

```
SORT FIELDS=(68,4,BI,A,16,4,BI,A),RECORD TYPE=V,LENGTH=32756
```

2. 2回目のソートでは、トランザクションファイルをCDCC、CDCE、CDCX順に戻します。このソートは、4バイトのオフセット4で実行する必要があります。
- 2回のソートを実行したら、今後のADACDCジョブでそのトランザクションファイルを処理できるようになります。



# 10 オペレーティングシステムに応じた考慮事項

---

■ z/OS .....	42
■ VSE .....	43
■ BS2000 .....	43

ソート要件を満たすため、ADACDCユーティリティでは、オペレーティングシステムの標準のソート機能が使用されます。各オペレーティングシステムで次の事項を考慮する必要があります。

このchapterでは、次のトピックについて説明します。

 **Note:** ISNを再利用できるファイルに対してADACDCを実行する場合（ADADBSまたはADALODISNREUSEを指定した場合）、プラットフォームとは関係なく、プライマリ出力ファイルとトランザクションファイルの特別なソートが必要になります。詳細は、「[ADACDCでのISNREUSEの使用](#)」を参照してください。

## z/OS

---

ソート機能を起動するための追加ジョブステップは必要ありません。ただし、ソートするデータ量によっては、ADACDCジョブステップでワークファイルまたは他のソート固有機能のためのDDステートメントが別途必要になります。詳細についてはソートのマニュアルを参照してください。

 **Note:** 通常、SYSOUT DDステートメントが指定されている場合は、ソートパッケージによって概要情報が提供されます。

ADACDCは、ソートを呼び出して制御を渡す際に、デフォルトでSORTというロードモジュール名を使います。ソートモジュール名が異なる場合には、次の方法で外部ソートプログラムの名称を指定して、Adabas オプションモジュール ADAOPD の再アセンブルとリンク設定を行う必要があります。

1. OPDOS メンバを修正して、パラメータ SORTPGM= でソートプログラムの名称を指定します。
2. ASMLOPD メンバを修正および実行して、モジュール ADAOPD のアセンブルとリンク設定を行います。

## VSE

外部ソートを呼び出すには、ADACDCユーティリティジョブでパーティションエリアにスペースを確保する必要があります。このため、次のいずれかの方法で、EXECステートメントにSIZEパラメータを指定する必要があります。

```
// EXEC ADARUN,SIZE=(ADARUN,128K)
```

または

```
// EXEC ADARUN,SIZE=(AUTO,128K)
```

ソート機能を起動するための追加ジョブステップは必要ありません。ただし、ソートするデータ量によっては、ADACDCジョブステップでワークファイルまたは他のソート固有機能のための、ソート関連のDLBLステートメントが別途必要になります。詳細についてはソートのマニュアルを参照してください。

ADACDCは、ソートを呼び出して制御を渡す際に、デフォルトでSORTというロードモジュール名を使います。ソートモジュール名が異なる場合には、パラメータSORTPGMで正しいソートモジュールの名称を指定して、Adabas オプションモジュール ADAOPD のアセンブルとリンク設定をやり直す必要があります。VSE の Adabas インストールに関するドキュメントの VSE システムのインストールに関するセクションで、オプションテーブルの修正、アセンブル、リンク設定の説明を参照してください。

## BS2000

大規模なソート処理の場合、Siemens 外部ソートが呼び出されることがあります。次のジョブカードが必要になります。

```
/SET-FILE-LINK BLSLIBnn,$.SORTLIB  
/SET-FILE-LINK SORTWK1,#SORTWK,BUF-LEN=STD(2),OPEN-MODE=OUTIN
```

```
/CREATE-FILE #SORTWK,PUB(SPACE=( &PRIM,&SEC )  
/START-PROGRAM . . . . .,RUN-MODE=ADVANCED,ALT-LIBRARY=YES
```

ここでは次の内容を表しています。

nn	00~99の値
#SORTWK	BS2000 コマンドで作成
&PRIM	割り当てるプライマリ PAM ページの数
&SEC	割り当てるセカンダリ PAM ページの数



**Note:** SORTWK1 ファイルのサイズは、ソート対象のデータ量によって変わります。

# 11 ADACDC ユーザー出口

---

■ ユーザー出口のインストール .....	46
■ ユーザー出口インターフェイス .....	65
■ ユーザー出口呼び出し .....	48

ADACDCでは、処理のさまざまなポイントでユーザー出口を呼び出し、ユーザーに処理中断の機会を提供します。

このchapterでは、次のトピックについて説明します。

## ユーザー出口のインストール

---

▶手順 11.1. ユーザー出口のインストール手順は次のとおりです。

- 1 ADACDC で使用するユーザー出口をコンパイルして、モジュール名を ADACDCUX とします。
- 2 上記のモジュールを ADACDC で使用できるようにします。

サンプルのユーザー出口 ADACDCUX がソースデータセットに用意されています。このサンプルは、ユーザー出口インターフェイスを使用したレコードの追加、削除、更新方法のみを示しています。

## ユーザー出口インターフェイス

---

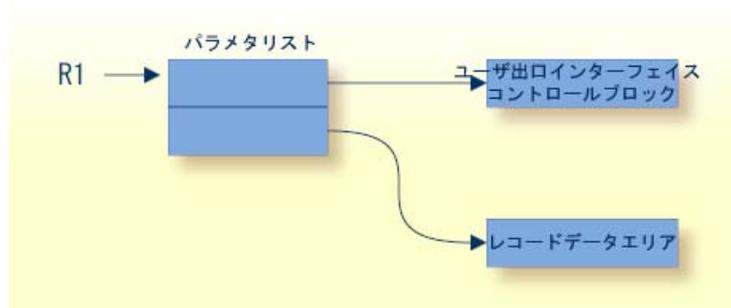
ユーザー出口をコールするには、レジスタを次のように設定してください。

R1	ユーザーパラメータリスト
R13	72 バイトの標準レジスタセーブエリア
R14	リターンアドレス
R15	エントリポイント

ユーザーパラメータリストには、次の2つのポイントが含まれています。

- 1つ目は、CDCU DSECT によってマップされる ADACDC ユーザー出口パラメータリストへのポイントです。
- 2つ目は、処理対象のレコードが渡される適切なユーザー出口のレコードエリアへのポイントです。

ユーザー出口で実行するアクションは、CDCUFUNC で指定します。一方、ユーザー出口が ADACDC に対して命令するリターン時のアクションは、CDCURESP フィールドで指定します。



### ADACDC ユーザー出口

ADACDC ユーザー出口インターフェイスコントロールブロック (CDCU DSECT) の構造は、次のとおりです。

バイト	説明	
0~3	定数 'CDCU'	
4~7	ユーザー出口が使用する	
8~11	第 2 パラメータ内のレコード長	
12	機能 ID :	
	X'00'	初期化
	X'04'	SORT 入力に渡す前
	X'08'	抽出ファイルに書き込む前
	X'0C'	プライマリ出力ファイルに書き込む前
X'10'	終了	
13	ユーザー出口からのレスポンスコード :	
	X'00'	正常処理
	X'04'	レコードの無視
	X'08'	レコードの更新済み
X'0C'	新規レコードの挿入	
14~31	予約済み (将来的に使用される予定)	

## ユーザー出口呼び出し

ユーザー出口の各種呼び出しとその目的について説明します。

### 初期化呼び出し (CDCUFUNC=CDCUINIT)

ADACDCでは、初期化中に、その後の処理で必要となるエリアの設定を行うためにユーザー出口を呼び出します。ユーザーコントロールブロックにリンクする必要がある場合は、CDCU内に CDCUUSER フィールドが提供されます。

レコードエリアポインタが指すデータは、この呼び出しには関係ありません。

### 終了呼び出し (CDCUFUNC=CDCUTERM)

ADACDCでは、終了処理中に、オープンしているファイルをクローズしたり ADACDC 実行後も保持されているエリアを解放するためにユーザー出口を呼び出します。例えば、CDCUUSERにリンクポインタが設定されている場合に、このエリアを解放して CDCUUSER フィールドを空値に設定します。

レコードエリアポインタが指すデータは、この呼び出しには関係ありません。

### **SORT** 入力呼び出し (CDCUFUNC=CDCUINPT)

ADACDCは、レコードを SORT ルーチンに入力として引き渡す前に、ユーザー出口を呼び出します。

レコードエリアポインタは、ユーザー出口によって返される圧縮データレコードをポイントします。圧縮データレコードには、CDCE コントロールブロックによって接頭辞が付けられます。

ユーザー出口は次の中から選択できます。

- 処理を正常に続行する。
- レコードの無視を要求する。
- 現在のレコードを更新する。
- レコードを追加する。この場合、レコードエリアポインタがポイントするレコードが SORT ルーチンに引き渡されます。これ以降、次のいずれかの状況になるまで、出口が呼び出されるたびに同じレコードが渡されます。
  - 処理を正常に続行する。
  - 今のレコードを無視または更新する。この場合、次の入力レコードの処理に進みます。

### 抽出出力呼び出し (CDCUFUNC=CDCUWRTE)

ADACDCは、フェーズ1の処理中にレコードを抽出ファイルに書き込む前に、ユーザー出口を呼び出します。この出口は、フェーズ1の処理中にのみ呼び出され、他の場合とは関係ありません。

レコードエリアポインタは、書き込まれる圧縮データレコードをポイントします。圧縮データレコードには、CDCE コントロールブロックによって接頭辞が付けられます。

ユーザー出口は次の中から選択できます。

- 処理を正常に続行する。
- レコードの無視を要求する。
- 現在のレコードを更新する。
- レコードを追加する。この場合、リターン時にレコードエリアポインタがポイントするレコードが抽出ファイルに書き込まれます。これ以降、次のいずれかの状況になるまで、出口が呼び出されるたびに同じレコードが渡されます。
  - 処理を正常に続行する。
  - 今のレコードを無視または更新する。この場合、次のレコードの抽出ファイルへの書き込み処理に進みます。

#### プライマリ出力呼び出し (CDCUFUNC=CDCUWRTO)

ADACDC は、レコードをプライマリ出力ファイルに書き込む前に、ユーザー出口を呼び出します。この出口は、フェーズ 1 の処理中は呼び出されません。

レコードエリアポインタは、書き込まれる非圧縮データレコードをポイントします。非圧縮データレコードには、CDCO コントロールブロックによって接頭辞が付けられます。

ユーザー出口は次の中から選択できます。

- 処理を正常に続行する。
- レコードの無視を要求する。
- 現在のレコードを更新する。
- レコードを追加する。この場合、リターン時にレコードエリアポインタがポイントするレコードがプライマリ出力ファイルに書き込まれます。これ以降、次のいずれかの状況になるまで、出口が呼び出されるたびに同じレコードが渡されます。
  - 処理を正常に続行する。
  - 今のレコードを無視または更新する。この場合、次のレコードの出力ファイルへの書き込み処理に進みます。

#### レコードの更新または追加

出口からのレコードを更新または追加するときは、次の点を考慮してください。

- 入力抽出出口ポイントおよび出力抽出出口ポイントで、CDCELEN/CDCERECL フィールドが CDCEDATA フィールドの後のデータの長さを正しく反映している必要があります。
- 入力抽出出口ポイントおよび出力抽出出口ポイントで、CDCOLEN/CDCORECL フィールドが CDCODATA フィールドの後のデータの長さを正しく反映している必要があります。
- ISN 更新を置き換えない場合は、入力出口ポイントで、追加するレコードの ISN がユニークである必要があります。
- ISN を置き換える場合は、入力出口ポイントで、ISN の全オカレンスを調べてから ISN の最後のオカレンスを更新するか、または ISN を置き換えるレコードを追加する必要があります。

- 抽出出力出口ポイントまたはプライマリ出力出口ポイントでレコードを追加するときは、ファイルとISNの組み合わせが重複していると、プライマリ出力ファイルに重複情報が書き込まれ、処理ルーチンに混乱を与える場合があるということに注意してください。

# 12 例

---

```
ADACDC RESETTXF, PHASE=BOTH
```

入力トランザクションファイル上の情報を無視し、プライマリ入力データを読み込んで、その入力データに基づいてプライマリ出力を生成します。

これ以降の入力用に有効なトランザクションを作成するために、ユーティリティを初めて実行するときは、この構文を使用して、入力トランザクションファイル（CDCTXI）を削除するか、またはダミーを指定してください。

```
ADACDC PHASE=1  
ADACDC FILE=20  
ADACDC FILE=40-50
```

プライマリ入力データを読み込み、ファイル20および40～50のみを処理します。これらのファイルの各 ISN の最新の更新だけを抽出ファイルに書き込みます。

```
ADACDC PHASE=2  
ADACDC FILE=44-47
```

すでに作成された抽出ファイルを読み込み、ファイル44～47の全レコードをプライマリ出力ファイルに書き込みます。



# 13 JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	54
▪ z/OS .....	55
▪ z/VM .....	56
▪ VSE .....	58

このセクションでは、ADACDC を BS2000、z/OS、z/VM、および VSE システムで実行するために必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	GCB および FDT エントリを読み取るのに必要
プロテクションログ	DDSIIN/ DDSIINnn	テープ/ディスク	シーケンシャルログ (PHASE=2 のときは不要)
抽出ファイル	CDCEXT	テープ/ディスク	出力抽出ファイルまたは入力抽出ファイル (PHASE=BOTH のときは不要)
入力トランザクションファイル	CDCTXI	テープ/ディスク	RESETTXF が指定されたとき、または PHASE=2 のときは不要
出力トランザクションファイル	CDCTXO	テープ/ディスク	PHASE=2 のときは不要
プライマリ出力ファイル	CDCOUT	テープ/ディスク	PHASE=1 のときは不要
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADACDC パラメータ	SYSDTA/ DDKARTE	ディスク/端末/リーダー	ユーティリティ
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADACDC メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

### ADACDC JCL の例 (BS2000)

```

/.ADACDC LOGON
/REMA ADACDC: CAPTURE DELTA CHANGES
/REMA
/ASS-SYSOUT EXAMPLE.ADACDC.SYSOUT
/MODIFY-TEST-OPTION DUMP=YES
/DELETE-FILE EXAMPLE.OUTPUT.TRANS.FILE
/SET-JOB-STEP
/DELETE-FILE EXAMPLE.OUTPUT.PRIMARY.FILE
/SET-JOB-STEP
/CREATE-FILE EXAMPLE.OUTPUT.TRANS.FILE,PUB(SPACE=(48,48))
/CREATE-FILE EXAMPLE.OUTPUT.PRIMARY.FILE,PUB(SPACE=(48,48))
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDASSOR1,EXAMPLE.DByyyyy.ASSOR1
    
```

```

/SET-FILE-LINK DDSIIN,EXAMPLE.DByyyyyy.PLOG000
/SET-FILE-LINK DDSIIN01,EXAMPLE.DByyyyyy.PLOG001
/SET-FILE-LINK DDSIIN02,EXAMPLE.DByyyyyy.PLOG002
/SET-FILE-LINK DDSIIN03,EXAMPLE.DByyyyyy.PLOG003
/SET-FILE-LINK CDCTXI,EXAMPLE.INPUT.TRANS.FILE
/SET-FILE-LINK CDCTXO,EXAMPLE.OUTPUT.TRANS.FILE
/SET-FILE-LINK CDCOUT,EXAMPLE.OUTPUT.PRIMARY.FILE
/SET-FILE-LINK DDRUCK,EXAMPLE.ADACDC.DRUCK
/SET-FILE-LINK DDPRINT,EXAMPLE.ADACDC.PRINT
/SET-FILE-LINK DDLIB,ADABAS.Vvrs.MOD
/START-PROGRAM *M(ADABAS.Vvrs.MOD,ADARUN)
ADARUN
PROG=ADACDC,MODE=MULTI, IDTNAME=xxxxxxxx,DEVICE=dddd,DBID=yyyyy
ADACDC
PHASE=BOTH,FILE=1,10,20-30
/LOGOFF SYS-OUTPUT=DEL
NOSPOOL

```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	GCB および FDT エントリを読み取るのに必要
プロテクションログ	DDSIIN	テープ/ディスク	シーケンシャルログ (PHASE=2 のときは不要)
入力トランザクションファイル	CDCTXI	テープ/ディスク	RESETTXF が指定されたとき、または PHASE=2 のときは不要
出力トランザクションファイル	CDCTXO	テープ/ディスク	PHASE=2 のときは不要
抽出ファイル	CDCEXT	テープ/ディスク	出力抽出ファイルまたは入力抽出ファイル (PHASE=BOTH のときは不要)
プライマリ出力ファイル	CDCOUT	テープ/ディスク	PHASE=1 のときは不要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADACDC パラメータ	DDKARTE	リーダー	ユーティリティ
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADACDC メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADACDC JCL の例 (z/OS)

この例については、MVSJOBS データセットの ADACDC を参照してください。

```
//ADACDC      JOB
//*
//*      ADACDC: CAPTURE DELTA CHANGES
//*
//CDC         EXEC PGM=ADARUN
//STEPLIB    DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1    <=== ASSO
//DDSIIN     DD   DSN=EXAMPLE.DByyyyy.PLOG(-3),DISP=SHR  <== PLOG TAPE
//           DD   DSN=EXAMPLE.DByyyyy.PLOG(-2),DISP=SHR  <== PLOG TAPE
//           DD   DSN=EXAMPLE.DByyyyy.PLOG(-1),DISP=SHR  <== PLOG TAPE
//           DD   DSN=EXAMPLE.DByyyyy.PLOG(0),DISP=SHR   <== PLOG TAPE
//CDCTXI     DD   DSN=EXAMPLE.input.trans.file,DISP=SHR
//CDCTXO     DD   DSN=EXAMPLE.output.trans.file,DISP=OLD
//CDCOUT     DD   DSN=EXAMPLE.output.primary.file,DISP=OLD
//DDDRUCK    DD   SYSOUT=X
//DDPRINT    DD   SYSOUT=X
//SYSUDUMP   DD   SYSOUT=X
//DDCARD     DD   *
ADARUN PROG=ADACDC,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE    DD   *
ADACDC PHASE=BOTH,FILE=1,10,20-30
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	GCB および FDT エントリを読み取るのに必要
プロテクションログ (PLOG)	DDSIIN	ディスク/テープ	シーケンシャルログ (PHASE=2 のときは不要)
入力トランザクションファイル	CDCTXI	ディスク/テープ	RESETTXF が指定されたとき、または PHASE=2 のときは不要
出力トランザクションファイル	CDCTXO	ディスク/テープ	PHASE=2 のときは不要
抽出ファイル	CDCEXT	ディスク/テープ	出力抽出ファイルまたは入力抽出ファイル (PHASE=BOTH のときは不要)
プライマリ出力ファイル	CDCOUT	ディスク/テープ	PHASE=1 のときは不要

データセット	DD 名	ストレージ	追加情報
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADACDC パラメータ	DDKARTE	リーダー	ユーティリティ
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADACDC メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADACDC JCL の例 (z/VM)

```

/*                                     */
/*      ADACDC JCL Example (z/VM)    */
/*                                     */
DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
/*                                     */
DATADEF DDSIIN,DSN=ADACDC.PLOG,MODE=A
DATADEF CDCTXI,DSN=ADACDC.INNPUT,MODE=A
DATADEF CDCTXO,DSN=ADACDC.OUTPUT,MODE=A
DATADEF CDCOUT,DSN=ADACDC.PRIMARY,MODE=A
/*                                     */
DATADEF DDPRINT,DSN=ADACDC.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADACDC.DDDRUCK,MODE=A
/*                                     */
DATADEF DDCARD,DSN=RUNCDC.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADACDC.CONTROL,MODE=A

EXECOS ADARUN
RCODE = RC
EXIT RCODE

```

## RUNCDC CONTROL A1 の内容

```
ADARUN PROG=ADACDC,DEVICE=dddd,DB=yyyyyy
```

## ADACDC CONTROL A1 の内容

ADACDC PHASE=BOTH, FILE=1, 10, 20-30

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	GCB および FDT エントリを読み取るのに必要
プロテクションログ	SIIN	テープ ディスク	SYS010 *	シーケンシャルログ (PHASE=2 のときは不要)
入力トランザクション	CDCTXI	テープ ディスク	SYS015 *	RESETTXF が指定されているとき、または PHASE=2 のときは不要
出力トランザクション	CDCTXO	テープ ディスク	SYS016 *	PHASE=2 のときは不要
抽出	CDCEXT	テープ ディスク	SYS017 *	出力抽出ファイルまたは入力抽出ファイル (PHASE=BOTH のときは不要)
プライマリ出力	CDCOUT	テープ ディスク	SYS018 *	PHASE=1 のときは不要
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	オペレーションマニュアル
ADACDC パラメータ	-	リーダー	SYSIPT	ユーティリティ
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADACDC メッセージ	-	プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

## ADACDC JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

この例については、メンバ ADACDC.X を参照してください。

```
* $$ JOB JNM=ADACDC,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADACDC
*       CAPTURE DELTA CHANGES
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,TAPE
// PAUSE MOUNT LOAD INPUT FILE ON TAPE cuu
// TLBL SIIN,'EXAMPLE.DByyy.PLOG'
// MTC REW,SYS010
// DLBL CDCTXI,'EXAMPLE.INPUT.TRANS.FILE',,SD
// EXTENT SYS015
// ASSGN SYS015,DISK,VOL=DISK01,SHR
// DLBL CDCTXO,'EXAMPLE.OUTPUT.TRANS.FILE',,SD
// EXTENT SYS016,,,sssss,nnnnn
// ASSGN SYS016,DISK,VOL=DISK02
// DLBL CDCOUT,'EXAMPLE.OUTPUT.TRANS.FILE',,SD
// EXTENT SYS018,,,sssss,nnnnn
// ASSGN SYS018,DISK,VOL=DISK04
// EXEC ADARUN,SIZE=ADARUN
ADARUN DBID=yyyyy,DEVICE=dddd,PROG=ADACDC,SVC=xxx,MODE=MULTI
/*
ADACDC PHASE=BOTH,FILE=1,10,20-30
/*
/&
* $$ EOJ
```



# 14 ADACMP ユーティリティ：データの圧縮／圧縮解除

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- 入力データの必要条件
- 処理（セグメント化されたレコードに関する考慮事項、再スタートに関する考慮事項、ユーザー出口6などの説明を含む）
- COMPRESS 機能：Adabas ファイルの圧縮
- DECOMPRESS 機能：Adabas ファイルの圧縮解除
- フィールド定義ステートメント
- JCL/JCS の必要項目と指定例
- COMPRESS 機能の出力
- DECOMPRESS 機能の出力



# 15 機能概要

---

- COMPRESS 機能の概要 ..... 64
- DECOMPRESS 機能の概要 ..... 65

このchapterでは、次のトピックについて説明します。

## COMPRESS 機能の概要

---

COMPRESS 機能を使用すると、データベースにロードされるデータレコードを編集して圧縮できます。

入力データ → ADACMP COMPRESS → ADALOD LOAD

以下からのデータレコードを入力として使用できます。

- ユーザーから提供される物理シーケンシャルデータセット（固定長または可変長のレコード）
- 既存の Adabas ファイル（つまり、ADACMP DECOMPRESS または ADAULD UNLOAD から）

入力データの論理構造と特性は、[フィールド定義ステートメント](#)を使用して記述します。

- FNDEF ステートメントは、フィールド（またはフィールドグループ）の定義に使用します。
- SUBFN と SUPFN ステートメントは、それぞれサブフィールドとスーパーフィールドの定義に使用します。
- COLDE、HYPDE、PHONDE、SUBDE、およびSUPDE ステートメントは、それぞれ照合ディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタ、サブディスクリプタ、およびスーパーディスクリプタの定義に使用します。

指定されたフィールド定義は、ファイルの Adabas フィールド定義テーブル（FDT）の作成に使用されます。または、フィールド定義を指定する代わりに、既存の Adabas FDT を使用することもできます（[FDT パラメータ](#)を参照）。

入力レコードのフィールドを入力レコード内での位置とは異なる順番で処理する場合、または1つ以上のフィールドをスキップする場合は、[FORMAT パラメータ](#)を使用して、入力フィールドの順番と位置を示すことができます。

ADACMP COMPRESS 機能では、入力データは次のように処理されます。

- 数字データの正当性がチェックされます。
- 英数字フィールドから末尾の空白が削除されます。
- 数字フィールドから先行ゼロが削除されます。
- アンパック形式の数字フィールドはパック形式にされます。

[固定 \(FI\) オプション](#)で定義されたフィールドは圧縮されません。

ユーザー出口を使用して、入力データをさらに編集することができます。詳細は、「[ユーザー出口6](#)」を参照してください。

ADALOD ユーティリティの入力として使用される ADACMP COMPRESS 機能の出力には、FDT、圧縮レコード以外に、ユーティリティレポートに表示されるデータストレージスペースの要件 (ADALOC DSSIZE パラメータ) および中間データセットとソートデータセットのサイズ見積もり (TEMPSIZE と SORTSIZE) が含まれます。

ADACMP COMPRESS 機能のレポートには、次の情報も示されます。

- 処理されたレコードの数
- 拒否されたレコードの数
- 圧縮率 (パーセンテージ)

拒否されたレコードを含むデータセットも生成されます。

## DECOMPRESS 機能の概要

DECOMPRESS 機能では、個々のファイルを圧縮解除します。

入力データ → ADACMP DECOMPRESS → 非圧縮レコード

入力データは、次の方法で既存の Adabas ファイル内のデータレコードから圧縮解除することができます。

- ADAULD (ファイルアンロード) ユーティリティを使用してアンロード
- 直接 (個別のファイルアンロードは行わない)

Adabas ファイルを直接圧縮解除する場合は、ADACMP DECOMPRESS の INFILE パラメータを使用します。圧縮解除処理の一部として、ターゲットファイルが FDT 情報なしでアンロードされます。これにより、大きなファイルを圧縮解除する場合に時間を短縮できます。

ISN パラメータが指定された場合は、ADACMP DECOMPRESS 機能の出力には ISN が含まれます。ファイルのデータ構造またはフィールド定義に対して必要な変更を加えると、Adabas 以外のプログラムの入力として、または COMPRESS 機能の入力として、DECOMPRESS 出力を使用できるようになります。



# 16 入力データの必要条件

---

■ 入力データの構造 .....	68
■ マルチプルバリューフィールドカウント .....	68
■ ピリオディックグループカウント .....	70
■ 可変長フィールドのサイズ .....	73

このchapterでは、次のトピックについて説明します。

## 入力データの構造

---

ADACMP入力データは、シーケンシャルデータセットまたはシーケンシャルファイル内に含まれている必要があります。インデックス付きのシーケンシャル入力やVSAMの入力は使用できません。

レコード長は固定、可変、あるいは不定長でもかまいません。最大入力レコード長は、オペレーティングシステムによって決定されます。最大圧縮レコード長は、使用しているデータストレージのブロックサイズおよびそのファイルに設定されている最大圧縮レコード長によって制限されます（ADALODユーティリティの **MAXRECL** パラメータを参照）。入力レコードの形式は、ブロック形式か非ブロック形式のいずれかです。

各レコード内のフィールドは、指定したフィールド定義ステートメント（または既存のAdabas FDTを使用する場合は指定したFDT）に従って構造化されている必要があります。ユーザー出口ルーチンを使用する場合、ユーザー出口処理後の構造がフィールド定義に一致する必要があります。入力レコードに対応するフィールド定義ステートメントがない場合、末尾の情報は無視され、ADACMPの出力にも含まれません。

また、レコードが含まれていないデータセットの入力も許可されます。

パラメータNUMREC=0を指定することにより、入力データセットの指定を省略することができます。

## マルチプルバリューフィールドカウント

---

各レコードのマルチプルバリューフィールド内の値の個数をフィールド定義ステートメントで指定するか、または各入力レコードの値の先頭にバリューカウントを指定する必要があります。フィールド定義ステートメントで指定する場合、指定可能なマルチプルバリューカウントの最小値は1で、最大値は65,534または191（ADACMP COMPRESSの **MUPEX** パラメータと **MUPECOUNT** パラメータの設定によって異なる）です。入力レコードに最小のカウント（個数）を指定するとき、値を含まないマルチプルバリューフィールドを示すゼロ（0）を指定できます。

各レコード内の値の個数が一定である場合、マルチプルバリューフィールドを定義するフィールド定義ステートメントにその個数が指定できます（**オカレンス**の指定）。この場合、入力レコード中にはカウントバイトを付けません。既存のFDTからのレコード定義を使用する場合、または前のDECOMPRESS処理からの入力データを使用する場合、データにはすでに長さの値が含まれています。フィールド定義ステートメントにはカウントを指定しません。

指定したカウントは、そのフィールドに **NU** オプションが指定されている場合、ADACMP によって修正されることがあります。つまり、空値が省略され、カウントフィールドがそれによって調整されます。このことは、バリューカウントを一連の各値の前に指定するか、またはフィールド定義ステートメントで指定するかにかかわらず、同様です。「**MU：マルチプルバリューフィールド**」セクションを参照してください。

### 例1：オカレンス数が異なるマルチプルバリューフィールドカウント

フィールド定義：

```
ADACMP FNDEF=' 01, MF, 5, A, MU, NU '
```

各レコードには異なる数のMF値が含まれ、一連の各オカレンスの前にカウントが指定されています。

	ADACMP 実行前	ADACMP 実行後
入力レコード 1 (3つの値)	MF カウント =3 AAAA BBBB CCCC	MF カウント =3 AAAA BBBB CCCC
入力レコード 2 (2つの値)	MF カウント =2 AAAA BBBB	MF カウント =2 AAAA BBBB
入力レコード 3 (3つの値)	MF カウント =3 AAAA bbbb CCCC	MF カウント =2 AAAA CCCC
入力レコード 4 (値なし)	MF カウント =0	MF カウント =0
入力レコード 5 (1つの値)	MF カウント =1 bbbb	MF カウント =0

### 例2：オカレンス数が同じマルチプルバリューフィールドカウント

フィールド定義：

```
ADACMP FNDEF=' 01, MF, 4, A, MU(3), NU '
```

各レコードには3つのMF値が含まれ、これらの値はフィールド定義ステートメントで指定されています。

	ADACMP 実行前	ADACMP 実行後
入力レコード 1	AAAA BBBB CCCC	MF カウント =3 AAAA BBBB CCCC
入力レコード 2	AAAA BBBB bbbb	MF カウント =2 AAAA BBBB
入力レコード 3	AAAA bbbb CCCC	MF カウント =2 AAAA CCCC
入力レコード 4	bbbb bbbb bbbb	MF カウント =0

## ピリオディックグループカウント

---

各ピリオディックグループでは、レコード内のフィールドの繰り返し（オカレンス）カウントを指定する必要があります。カウントは、すべてのレコードについてフィールド定義ステートメント内で指定するか、または1バイトまたは2バイトのバイナリ値（**MUPECOUNT**パラメータの値によって異なる）として、すべてのレコードの各オカレンスグループの前に指定します。フィールド定義ステートメントでカウントを指定する場合、入力レコードにはカウントバイトを付けません。フィールド定義ステートメントで指定する場合、指定可能なカウントの最小値は1で、最大値は65,534または191（ADACMP COMPRESSの**MUPEX**パラメータと**MUPECOUNT**パラメータの設定によって異なる）です。レコードに最小のカウントを指定するときは、オカレンスを含まないピリオディックグループを示す値としてゼロ（0）を指定できます。

指定したオカレンスカウントは、ピリオディックグループ内の全フィールドが**NU**オプションを指定して定義されている場合は、ADACMPによって修正されることがあります。つまり、特定オカレンス内の全フィールドが空値であり、空値でないオカレンスが後にない場合、そのオカレンスは削除され、ピリオディックグループオカレンスカウントがそれに従って調整されません。

## 例1：オカレンス数が異なるピリオディックグループカウント

フィールド定義：

```
ADACMP COMPRESS
ADACMP FNDEF=' 01,GA,PE'
ADACMP FNDEF=' 02,A1,4,A,NU'
ADACMP FNDEF=' 02,A2,4,A,NU'
```

入力記録には、GAの可変個のオカレンスが含まれます（**MUPEX**パラメータは指定されおらず、その結果**MUPECOUNT**パラメータがデフォルトの"1"に設定されているため、最大191オカレンスまで許可されます）。オカレンスカウントは、入力記録内の各オカレンスグループの前に指定されています。

	ADACMP 実行前	ADACMP 実行後
入力記録 1	GA カウント =2	GA カウント =2
	GA (第1オカレンス) A1=AAAA A2=BBBB	A1=AAAA A2=BBBB
	GA (第2オカレンス) A1=CCCC A2=DDDD	A1=CCCC A2=DDDD
入力記録 2	GA カウント =1	GA カウント =0
	GA (第1オカレンス) A1=bbbb A2=bbbb	削除 削除
入力記録 3	GA カウント =3	GA カウント =3
	GA (第1オカレンス) A1=AAAA A2=bbbb	A1=AAAA A2= 削除
	GA (第2オカレンス) A1=BBBB A2=bbbb	A1=BBBB A2= 削除
	GA (第3オカレンス) A1=CCCC A2=bbbb	A1=CCCC A2= 削除
入力記録 4	GA カウント =0	GA カウント =0

例 2：オカレンス数が同じピリオディックグループカウント

フィールド定義：

```
ADACMP FNDEF=' 01,GA,PE(3) '
ADACMP FNDEF=' 02,A1,4,A,NU '
ADACMP FNDEF=' 02,A2,4,A,NU '
```

全入力レコードにはGAの3つのオカレンスが生まれ、これらのオカレンスはフィールド定義ステートメントで指定されています。

	ADACMP 実行前	ADACMP 実行後
入力レコード 1		GA カウント =3
	GA (第 1 オカレンス) A1=AAAA A2=bbbb	A1=AAAA A2 削除
	GA (第 2 オカレンス) A1=BBBB A2=bbbb	A1=BBBB A2 削除
	GA (第 3 オカレンス) A1=CCCC A2=bbbb	A1=CCCC A2 削除
入力レコード 2		GA カウント =2 (注参照)
	GA (第 1 オカレンス) A1=bbbb A2=bbbb	A1= 削除 A2= 削除
	GA (第 2 オカレンス) A1=BBBB A2=bbbb	A1=BBBB A2= 削除
	GA (第 3 オカレンス) A1=bbbb A2=bbbb	A1= 削除 A2= 削除
入力レコード 3	全オカレンスが 空値	GA カウント =0 全オカレンスが削除される



**Note:** 第 1 オカレンスは、空値でないオカレンスが後に続くため、カウントに含まれます。第 3 オカレンスは、空値でないオカレンスが後に続かないため、カウントに含まれません。

### 例 3 : PE グループへのフィールドの追加

AW という PE にフィールド AY を追加します。

古い FDT	新しい FDT
01 AA,8,A,DE,UQ	01 AA,8,A,DE,UQ
01 AW,PE	01 AW,PE
02 AX,8,U,NU	02 AX,8,U,NU
02 AT,8,U,NU	02 AT,8,U,NU
01,AZ,3,A,DE,MU,NU	02 AY,8,U,NU
	01,AZ,3,A,DE,MU,NU

 **Note:** 現在 PE に存在しているすべてのフィールドを指定する必要があります。

1. PE の最大オカレンス（2 の結果など）を決定します。
2. フォーマットパラメータを指定してファイルを圧縮解除します。
3. INFILE=xx,FORMAT='AA,AX1-2,AT1-2,AZ' で圧縮解除します。
4. 再度圧縮します。

```
ADACMP COMPRESS FILE=32
ADACMP FORMAT='AA,AX1-2,AT1-2,AZ'
ADACMP FNDEF='01,AA,8,A,DE,UQ'
ADACMP FNDEF='01,AW,PE(2)'
ADACMP FNDEF='02,AX,8,U,NU'
ADACMP FNDEF='02,AT,8,U,NU'
ADACMP FNDEF='02,AY,8,U,NU'
ADACMP FNDEF='01,AZ,3,A,DE,MU,NU'
```

## 可変長フィールドのサイズ

可変長フィールド（フィールド定義に長さパラメータの指定なし）の各値の先頭には、値の長さ（長さバイトを含む）を示す1バイトのバイナリカウントを付ける必要があります。可変長（つまり、長さ 0）で指定された LA フィールドの先頭には、2 バイトの長さ（長さバイトを含む）を付ける必要があります。可変長で指定された LB フィールドの先頭には、4 バイトの長さ（長さバイトを含む）を付ける必要があります。

### 可変長フィールドのサイズの例

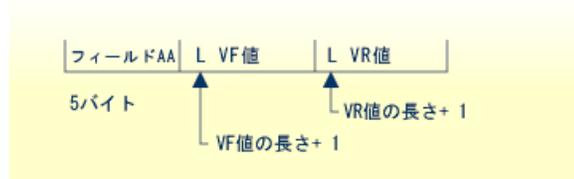
フィールド定義：

```
ADACMP FNDEF='01,AA,5,A,DE'
```

```
ADACMP FNDEF='01,VF,0,A'
```

```
ADACMP FNDEF='01,VR,0,A'
```

入力レコード：



# 17 処理

---

■ セグメント化されたレコードに関する考慮事項 .....	76
■ データの検証 .....	82
■ データ圧縮 .....	82
■ 非圧縮データの LOB 値と値参照の表現 .....	84
■ 圧縮レコード内の 191 を超える MU および PE オカレンスの特定 .....	85
■ 再スタートに関する考慮事項 .....	85
■ ユーザー出口 6 .....	85

このchapterでは、次のトピックについて説明します。

## セグメント化されたレコードに関する考慮事項

---

ADACMP の非圧縮レコードが長すぎてシーケンシャルデータセットで許容されている最大レコード長に収まらない場合（32 KB 以下）、ADACMP では、このレコードを複数の物理レコードにセグメント化することができます。1つの非圧縮論理レコードを1つ以上の非圧縮物理レコードにスパンすることができます。

また、ADACMP ユーティリティでは、非圧縮出力に特別なヘッダー（ADAH および ADAC）を作成することができます。これらの特別なヘッダーは、ADACMP 処理でのみ使用されます。論理レコード内のペイロードデータの位置、および同じ論理レコード内の複数の物理レコード間の関係は、これらのヘッダーから識別されます。ADAH ヘッダーが論理レコードの最初の物理レコードに使用され、ADAC ヘッダーが論理レコードを構成する2番目以降の物理レコードに使用されます。非圧縮論理レコードのセグメント化が必要ない場合（1つの物理レコードに収まる場合）、ADAH ヘッダーのみが作成されます。ADAC ヘッダーは不要です。

ADACMP 圧縮解除ロジックおよびレコードスパンニングで発生するレコードのセグメンテーションと混同しないでください。スパンドレコードも複数の物理レコード（1つのプライマリレコードと複数のセカンダリレコード）で構成されますが、圧縮レコードになります。また、各スパンドレコードには、標準スパンドレコードヘッダーが自動的に割り当てられます。このヘッダーは、ADACMP を使用して非圧縮レコード用に作成可能な ADAH および ADAC ヘッダーとは異なります。ADACMP によって生成されるセグメント化されたレコードには、標準スパンドレコードヘッダーは含まれません。スパンドレコードの詳細については、「スパンドレコード」を参照してください。

このsectionでは、次のトピックについて説明します。

- [ADACMP ヘッダーの作成とサポート](#)
- [ADAH および ADAC ヘッダーの説明](#)

### ADACMP ヘッダーの作成とサポート

ADACMP DECOMPRESS の HEADER パラメータでは、圧縮解除ロジックで出力にヘッダーを作成するかどうかを制御します。ADACMP COMPRESS の HEADER パラメータでは、圧縮ロジックで非圧縮入力の一部として ADACMP ヘッダーを許容するかどうかを制御します。

## ADAH および ADAC ヘッダーの説明

このsectionでは、次のトピックについて説明します。

- ADAH ヘッダー
- ADAC ヘッダー
- 例

### ADAH ヘッダー

ADACMP ヘッダーを使用する場合、論理レコードの最初の物理レコードは、次の情報を含む ADAH ヘッダーで始まります。

- 文字 "ADAH"
- ADAH ヘッダーの長さ
- これが論理レコード内で最後の物理レコードであるのか、同じ論理レコードの別の物理レコードがこのレコードに続くのかを示す継続インジケータです。
- レコード（ヘッダーを含む）の全体の長さ 最初の物理レコードが書き込まれるときにレコードの全体の長さが不明な場合、値がゼロになることがあります。
- この物理レコード内のペイロードデータ（論理レコードのセグメント）の長さ ペイロードデータの長さを表し、ADAH ヘッダーの長さは含まれません。長さは、物理レコードの長さからヘッダーの長さを引いた値以下である必要があります。この値よりも小さい場合は、物理レコード内のその他のデータ（ペイロードデータ長でカバーされないデータ）はすべて無視されます。

ペイロードデータは、ADAH ヘッダーの後に続きます。

ADAH DSECT は、分散 Adabas 8 MVSSRCE ライブラリのメンバ ADAH 内にあります。

### ADAC ヘッダー

ADACMP 非圧縮レコードのセグメント化を行った場合、および ADACMP ヘッダーが要求された場合は、論理レコードを構成する 2 番目以降のすべての物理レコードの先頭に、次の情報を含む ADAC ヘッダーが置かれます。

- 文字 "ADAC"
- ADAC ヘッダーの長さ
- これが論理レコード内で最後の物理レコードであるのか、同じ論理レコードの別の物理レコードがこのレコードに続くのかを示す継続インジケータです。
- 論理レコード内のこのセカンダリレコードのシーケンス番号 論理レコードの 2 番目の物理レコードは最初のセカンダリレコードになるため、そのシーケンス番号は "1" です。シーケンス番号は、番号抜けのない昇順になります。

- この物理レコードに含まれるペイロードデータ（セグメント）の論理レコード内のオフセット  
このオフセットは、論理レコード内の先行する各物理レコードのペイロードデータ長の合計です。
- この物理レコード内のペイロードデータ（セグメント）の長さ ペイロードデータの長さを表します。ADAC ヘッダーの長さは含まれません。長さは、物理レコードの長さからヘッダーの長さを引いた値以下である必要があります。この値よりも小さい場合は、物理レコード内のその他のデータ（ペイロードデータ長でカバーされないデータ）はすべて無視されます。

ペイロードデータは、ADAC ヘッダーの後に続きます。

ADAC DSECT は、分散 Adabas 8 MVSSRCE ライブラリのメンバ ADAC 内にあります。

例

次の表は、非圧縮データの3つの論理レコードを7つの物理レコードにスパンした例を示しています。

 **Note:** ADAH および ADAC ヘッダーの DSECT は、分散 Adabas 8 MVSSRCE ライブラリのメンバ ADAH および ADAC 内にあります。

論理レコード	物理レコードヘッダー	ヘッダーフィールド		説明
		フィールド	値	
1	ADAH	ADAHEYE	ADAH	ADAH ヘッダーのアイキャッチャー
		ADAHLEN	32	ADAH ヘッダー長
		ADAHIND	C	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	バイナリの 0 を含んでいる必要があります。
		ADAHTLEN	50000	論理レコードの全体の長さ。最初のセグメントが書き込まれるときに全体の長さが不明な場合、この値は 0 になることがあります。
		予約済み	0	予約済み
		ADAHSLEN	27962	このセグメントの長さ（ペイロードデータの長さ）。ADAHLEN と ADAHSLEN の値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
	ADAHDATA	'Record 1 - payload data part 1'	ペイロードデータ	
ADAC	ADACEYE	ADAC	ADAC ヘッダーのアイキャッチャー	

論理レコード	物理レコードヘッダー	ヘッダーフィールド		説明
		フィールド	値	
		ADACLEN	32	ADAC ヘッダー長
		ADACIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み
		ADACSEQ	1	論理レコードにおける継続レコードのシーケンス番号（最初の ADAC レコードのシーケンス番号は "1"）。
		ADACOFFS	27962	論理レコードにおけるセグメントのオフセット（最初のペイロードデータバイトのオフセットは "0"）。
		予約済み	0	予約済み
		ADACSLLEN	22038	このセグメントの長さ（ペイロードデータの長さ）。ADACLEN と ADACSLLEN の値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
		ADACDATA	'Record 1 - payload data part 2'	継続レコードのペイロードデータ
2	ADAH	ADAHEYE	ADAH	ADAH ヘッダーのアイキャッチャー
		ADAHLEN	32	ADAH ヘッダー長
		ADAHIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	バイナリの 0 を含んでいる必要があります。
		ADAHTLEN	25000	論理レコードの全体の長さ。最初のセグメントが書き込まれるときに全体の長さが不明な場合、この値は 0 になることがあります。
		予約済み	0	予約済み
		ADAHSLEN	25000	このセグメントの長さ（ペイロードデータの長さ）。ADAHLEN と ADAHSLEN の値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
		ADAHDATA	'Record 2 - payload data'	ペイロードデータ
3	ADAH	ADAHEYE	ADAH	ADAH ヘッダーのアイキャッチャー
		ADAHLEN	32	ADAH ヘッダー長

論理レコード	物理レコードヘッダー	ヘッダーフィールド		説明
		フィールド	値	
		ADAHIND	C	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	バイナリの0を含んでいる必要があります。
		ADAHTLEN	0	論理レコードの全体の長さ。最初のセグメントが書き込まれるとき、全体の長さが不明な場合は、この値はゼロになることがあります（論理レコード3の場合など）。
		予約済み	0	予約済み
		ADAHSLEN	27962	このセグメントの長さ（ペイロードデータの長さ）。ADAHLENとADAHSLENの値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
		ADAHDATA	'Record 3 - payload data part 1'	ペイロードデータ
ADAC		ADACEYE	ADAC	ADACヘッダーのアイキャッチャー
		ADACLEN	32	ADACヘッダー長
		ADACIND	C	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み
		ADACSEQ	1	論理レコードにおける継続レコードのシーケンス番号（最初のADACレコードのシーケンス番号は"1"）。
		ADACOFFS	27962	論理レコードにおけるセグメントのオフセット（最初のペイロードデータバイトのオフセットは"0"）。
		予約済み	0	予約済み
		ADACSLLEN	27962	このセグメントの長さ（ペイロードデータの長さ）。ADACLENとADACSLLENの値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
		ADACDATA	'Record 3 - payload data part 2'	継続レコードのペイロードデータ
ADAC		ADACEYE	ADAC	ADACヘッダーのアイキャッチャー
		ADACLEN	32	ADACヘッダー長

論理レコード	物理レコードヘッダー	ヘッダーフィールド		説明
		フィールド	値	
		ADACIND	C	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み
		ADACSEQ	2	論理レコードにおける継続レコードのシーケンス番号（最初の ADAC レコードのシーケンス番号は "1"）。
		ADACOFFS	55924	論理レコードにおけるセグメントのオフセット（最初のペイロードデータバイトのオフセットは "0"）。
		予約済み	0	予約済み
		ADACSLLEN	27962	このセグメントの長さ（ペイロードデータの長さ）。ADACLEN と ADACSLLEN の値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
		ADACDATA	'Record 3 - payload data part 3'	継続レコードのペイロードデータ
	ADAC	ADACEYE	ADAC	ADAC ヘッダーのアイキャッチャー
		ADACLEN	32	ADAC ヘッダー長
		ADACIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコードセグメント E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み
		ADACSEQ	3	論理レコードにおける継続レコードのシーケンス番号（最初の ADAC レコードのシーケンス番号は "1"）。
		ADACOFFS	83886	論理レコードにおけるセグメントのオフセット（最初のペイロードデータバイトのオフセットは "0"）。
		予約済み	0	予約済み
		ADACSLLEN	16114	このセグメントの長さ（ペイロードデータの長さ）。ADACLEN と ADACSLLEN の値の合計が、物理レコードの最小長です。物理レコードはこの値よりも長くすることができます。この場合、超過したデータに意味はなく、無視されます。
		ADACDATA	'Record 3 - payload data part 4'	継続レコードのペイロードデータ

## データの検証

ADACMPでは、U（アンパック形式）またはP（パック形式）フォーマットで定義されたフィールドをチェックし、フィールド値が数字であり、かつ正しいフォーマットであることを確認します。値が空値である場合、この空値はそのフィールドに指定されたフォーマットに対応している必要があります（「[フィールド定義ステートメント](#)」セクションの「[SQL 空値の表現](#)」を参照）。

英数字 (A)	空白 (16 進数 X'40')
2 進数 (B)	バイナリの 0 (16 進数 X'00')
固定小数点 (F)	バイナリの 0 (16 進数 X'00')
浮動小数点 (G)	バイナリの 0 (16 進数 X'00')
パック形式 (P)	符号付きパック 10 進数のゼロ (右端の下位バイトに、X'00' に続けて X'0F'、X'0C'、または X'0D' を伴う)
アンパック形式 (U)	符号付アンパック 10 進数のゼロ (右端の下位バイトに、X'F0' に続けて X'C0' または X'D0' を伴う)

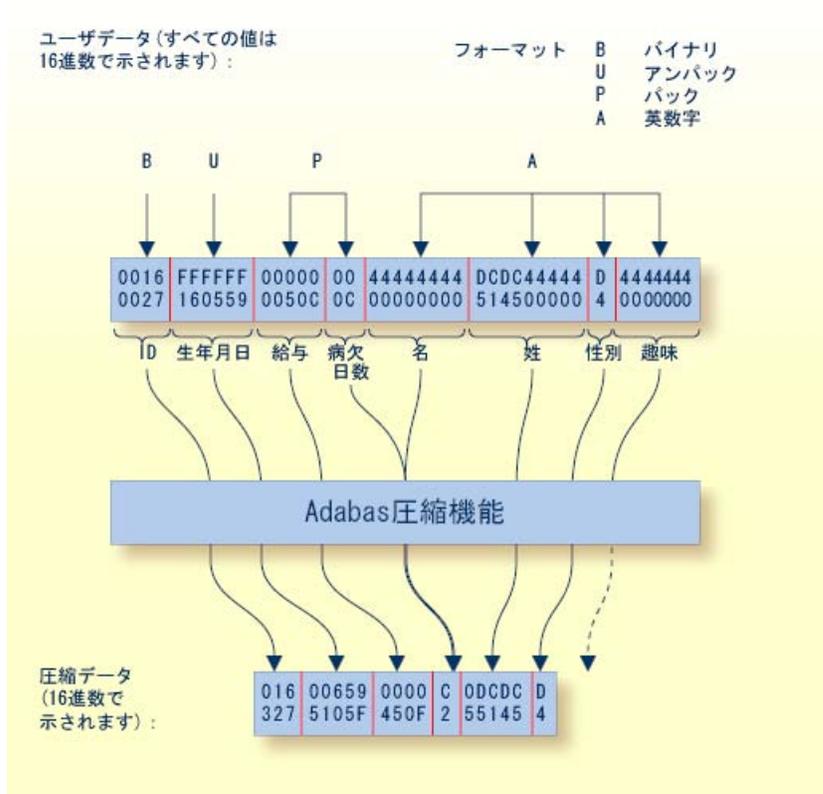
無効なデータを含むレコードは、すべて ADACMP エラー (DDFEHL) データセットに書き込まれ、圧縮データセットには書き込まれません。

## データ圧縮

各フィールドの値は ([FI オプション](#)の指定がない限り) 次のように圧縮されます。

- A フォーマットで定義されたフィールドの末尾の空白は削除されます。
- 数字フィールド (B、F、P、または U フォーマットで定義されたフィールド) の先行ゼロは削除されます。
- フィールドが U フォーマット (アンパック形式) で定義されている場合、値は P フォーマット (パック形式) に変換されます。
- 浮動小数点 (G フォーマット) フィールドの末尾のゼロは削除されます。
- フィールドが NU オプションを指定して定義され、かつ値が空値である場合は、1 バイトのインジケータが格納されます。このインジケータバイトでは、16 進数の X'C1' は 1 個の空値フィールドが後に続くことを示し、X'C2' は 2 個の空値フィールドが後に続くことを示し、以下同様に、最大で 63 個の空値フィールドを表現します。64 個以上の場合、インジケータバイトが繰り返されます。SQL 空値 (NC オプションフィールド) の圧縮については、「[フィールド定義ステートメント](#)」セクションの「[SQL 空値の表現](#)」を参照してください。
- レコードの末尾にある空値フィールドは格納されないため、圧縮されません。

## データ圧縮の例



## ADACMP 圧縮

下図は、次のフィールド定義とその値が ADACMP によってどのように処理されるかを示しています。

```
FNDEF='01, ID, 4, B, DE'
FNDEF='01, BD, 6, U, DE, NU'
FNDEF='01, SA, 5, P'
FNDEF='01, DI, 2, P, NU'
FNDEF='01, FN, 9, A, NU'
FNDEF='01, LN, 10, A, NU'
```

```
FNDEF='01,SE,1,A,FI'  
FNDEF='01,HO,7,A,NU'
```

## 非圧縮データの LOB 値と値参照の表現

このsectionでは、ラージオブジェクト (LB) フィールドの値、LB フィールドの値参照、および 32 KB より長い論理レコードを ADACMP COMPRESS 機能の入力データセットでどのように表現する必要があり、これらの項目が ADACMP DECOMPRESS 機能の出力データセットでどのように表現されるかについて説明します。

- ラージオブジェクト (LB) フィールドの値
- ラージオブジェクト (LB) フィールドの値参照

### ラージオブジェクト (LB) フィールドの値

FORMAT パラメータを指定せずに ADACMP を実行した場合、非圧縮データ内の各ラージオブジェクト (LB) フィールド値の先頭に4バイトの長さフィールドが付けられます。長さの値は、固有の LB フィールド値の長さ、長さフィールド自体の4バイトを加えた値になります。NB オプションを指定せずに定義されたフィールドの場合、空の LB フィールド値は、値5と1つの空白を含む長さフィールドで構成されます。NB オプションを指定して定義されたフィールドの場合、空の LB フィールド値は、値4のみを含む長さフィールドで構成されます。

ADACMP COMPRESS を使用して LB フィールドを含む FDT を定義する場合、非圧縮入力の各 LB フィールド値は 253 バイトと等しいかそれより小さくする必要があります。

### ラージオブジェクト (LB) フィールドの値参照

LOBVALUES=NO を指定して ADACMP DECOMPRESS 機能を実行し、LOB ファイルグループの基本ファイル内のレコードのみを圧縮解除した場合 (関連する LOB ファイルに格納されている LB フィールド値はすべて省略)、LOB ファイル内の LB フィールド値に対する基本ファイルレコード内の各参照は、非圧縮出力で次のように表現されます。

- LB フィールド値の4バイトの長さフィールドには、LB フィールド値への参照の存在を示す X'FFFFFFFF' (高い値) が含まれます。
- インジケータの後には、LB フィールド値参照に関する2バイトの長さフィールド (長さバイトを含む) が続きます。長さの値は、固有の LB フィールド値参照の長さ、長さフィールド自体の2バイトを加えた値になります。
- 長さフィールドの後には、固有の LB フィールド値参照が続きます。

ADACMP COMPRESS 機能で LOBVALUES=NO を指定した場合、処理対象の基本ファイルに関連付けられている LOB ファイルに格納されている LB フィールド値の場所で、同じ構造が要求されます。

ADACMP COMPRESS の入力として使用する LB フィールド値参照は、ADACMP DECOMPRESS から取得する必要があります。COMPRESS を使用して新しい LB フィールド値参照を取り入れる適切な方法はありません。これは、LOB ファイル内の既存の LB フィールド値を正しく参照できないためです。

## 圧縮レコード内の 191 を超える MU および PE オカレンスの特定

圧縮レコード内に 191 を超える MU および PE オカレンスがある場合は、そのオカレンスカウンターの先頭に x'C0' バイトが付加されることによって、そのことが示されます。このバイトは、レコードの圧縮時に ADACMP ユーティリティまたはニュークリアスにより設定されます。x'C0' インジケータバイトの後に、後続の MU または PE オカレンスカウントに使用されるカウントバイトの数を示すバイトが続きます。例えば、次のインジケータの場合を考えます。

```
x'c0020204'
```

この例では、x'C0' はこれが拡張カウントであることを示し、x'02' はカウントバイトが 2 つあることを示します。また、x'0204' は、このフィールドに 516 個のオカレンスが存在することを示します。

## 再スタートに関する考慮事項

ADACMP には再スタート機能は備えられていません。ADACMP の実行が中断された場合、最初からやり直す必要があります。

## ユーザー出口 6

ADACMP COMPRESS 処理時の編集用に、ユーザー出口 6 と呼ばれるユーザー作成ルーチンを使用することができます。このルーチンは、アセンブラまたは COBOL で記述できます。記述したルーチンは、アセンブルまたはコンパイルした後、Adabas ロードライブラリ（またはそれに連結されたライブラリ）にリンクする必要があります。

ユーザー出口 6 は、次のように指定して呼び出します。

```
ADARUN UEX6=program
```

ここで、*program* はロードライブラリ内のルーチン名です。

ユーザー出口 6 の構造とパラメータの詳細は、ユーザー出口とハイパー出口に関する説明を参照してください。

# 18 COMPRESS : Adabas ファイルの圧縮

---

■ 基本パラメータとサブパラメータ .....	88
■ オプションのパラメータとサブパラメータ .....	90
■ ADACMP COMPRESS 例 .....	97

```

ADACMP COMPRESS {field-definition-statements | FDT = file-number}
  [CODE = cipher-code]
  [DATADEVICE = { device-type | ADARUN-device }]
  [DEVICE = { device-type-list | ADARUN-device }]
  [FACODE = file-alpha-EBCDIC-key]
  [FILE = { file-number | 0 }]
  [FWCODE = file-wide-key]
  [FUWCODE = { wide-key | UWCODE-definition }]
  [FORMAT = format]
  [HEADER = { YES | NO }]
  [LOBDEVICE = { device-type-list | DEVICE-parmvalue }]
  [LOBVALUES = { YES | NO }]
  [MAXLOGRECL = buffer-size]
  [MUPECOUNT = 1 | 2]
  [MUPEX]
  [NOUSERABEND]
  [NUMREC = number-of-records]
  [PASSWORD = "password"]
  [RECFM = { E | FB | V | VB | U }]
    [LRECL = record-length]
  [SPAN]
  [ { USERISN | MINISN = { start-isn | 1 } } ]
  [UACODE = userdata-alpha-key]
  [UWCODE = { userdata-wide-key | FWCODE-definition }]
  [UARC = { userdata-architecture-key | 2 }]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータとサブパラメータ

### *field-definition-statements*

ADACMP の入力として指定するフィールド定義ステートメントは、次の目的で使用されます。

- 入力レコードに含まれる各フィールドの長さフォーマットを指定します。これにより ADACMP は、編集および圧縮時に正しいフィールド長とフォーマットを判断できます。
- ファイルのフィールド定義テーブル (FDT) を作成します。このテーブルは、Adabas コマンド実行中にファイル内のすべてのフィールド (またはグループ) の論理構造と特徴を判断するために Adabas が使用します。

ADACMP COMPRESS では FDT パラメータとフィールド定義ステートメントのいずれかを指定する必要があります。両方を指定すると、フィールド定義ステートメントが無視されます。

この構文では、次のフィールド定義ステートメントを使用できます。

```
FNDEF = 'field-definition'  
[COLDE = 'collation-descriptor-definition']  
[HYPDE = 'hyperdescriptor-definition']  
[PHONDE = 'phoneticdescriptor-definition']  
[SUBDE = 'subdescriptor-definition']  
[SUBFN = 'subfield-definition']  
[SUPDE = 'superdescriptor-definition']  
[SUPFN = 'superfield-definition']
```

フィールド定義ステートメントとその構文の詳細は、「[フィールド定義ステートメント](#)」を参照してください。

#### FDT : 既存の Adabas フィールド定義テーブルの使用

使用する既存の Adabas フィールド定義テーブル (FDT) を指定します。FDT は、既存のファイルのものでも、ADADBS ユーティリティの KEEPFDI オプションで削除されたファイルのものでもかまいません。

ADACMP COMPRESS では FDT パラメータとフィールド定義ステートメントのいずれかを指定する必要があります。両方を指定すると、フィールド定義ステートメントが無視されます。

FDT パラメータが指定されている場合、入力データは、FORMAT パラメータが使用されていない限り、指定の FDT に定義された構造と一致している必要があります。FDT でマルチプルバリューフィールドまたはピリオディックグループを定義する場合、長さの値を定義するか、または FDT にその定義がすでに含まれている必要があります。「[マルチプルバリューフィールドカウント](#)」セクションおよび「[ピリオディックグループカウント](#)」セクションを参照してください。

FDT パラメータを使用した場合、指定したすべてのフィールド定義ステートメントは無視されます。

## オプションのパラメータとサブパラメータ

### CODE : サイファコード

データを暗号化してデータベースにロードする場合、このパラメータでサイファ（暗号）コードを指定する必要があります。暗号化についての詳細は『*Adabas Security Manual*』を参照してください。



**Note:** ファイルに LB フィールドが含まれている場合、ADACMP COMPRESS 機能で CODE パラメータを指定することはできません。

### DATADEVICE : デバイスタイプ

DATADEVICE パラメータは、スパンドレコードのセグメンテーションに使用されるデータストレージデバイスタイプを指定します。SPAN パラメータが指定された場合、ADACMP はスパンされた長い圧縮レコードをセグメントに分割します。このセグメントは、DATADEVICE パラメータで暗示的に指定されるデータストレージのブロックサイズよりわずかに小さくなります。

SPAN パラメータが指定されていない場合、DATADEVICE の値は不要です。ただし、値を指定して圧縮レコードのサイズを制限することはできます。この場合、指定したストレージデバイスタイプのブロックサイズを超過したすべてのレコードがDDFEHLエラーデータセットに書き込まれます。

### DEVICE : デバイスタイプ

DEVICE パラメータを指定した場合、ADACMP で、指定した各デバイスタイプに対する実行スペースの必要容量が計算され、表示されます。このレポートには、ファイルに MUXEX パラメータが設定されているかどうかを示す情報も含まれます。

DEVICE パラメータでデバイスタイプを指定しない場合、ADARUN デバイスタイプがデフォルトとして使用されます。

### FACODE : 英数字フィールドのエンコード

ファイル内の英数字フィールドに UES ファイルエンコードを定義する場合は、FACODE を指定する必要があります。英数字のエンコードは EBCDIC エンコードファミリに準拠する必要があります（例えば、スペース文字は X'40'）。

### FILE : ファイル番号

FDT にハイパーディスクリプタが含まれている場合、このパラメータを指定する必要があります。指定したファイル番号は、関連するハイパー出口の入力となります。ハイパー出口の詳細は、『*Adabas DBA リファレンスマニュアル*』を参照してください。

ユーザー出口 6 には常にこのファイル番号が渡されます。FILE を指定しない場合、値はゼロであると仮定されます。

### FORMAT : 入力レコードのフォーマット定義

このパラメータを使用して、入力レコードのフィールドの位置、フォーマット、および長さを示すフォーマット定義を行うことができます。指定するフォーマットは、『*Adabas コマ*

『アドリファレンス』に記載の更新コマンド用フォーマットバッファエントリの規則に準拠する必要があります。

変換規則は、『Adabas コマンドリファレンス』の Adabas 更新コマンドの記述に準じます。SQL 空値 (NC オプション) フィールドの変換については、「[NC : SQL 空値オプション](#)」を参照してください。FORMAT パラメータで省略したフィールドには空値が割り当てられます。

FORMAT パラメータを省略した場合、入力レコードは、指定したフィールド定義ステートメントの順番で処理されるか、既存の Adabas フィールド定義テーブルに従って処理されます (FDT パラメータを指定している場合)。

LOBVALUES パラメータが NO に設定されている場合、FORMAT パラメータで指定する定義で LB フィールドを使用することはできません。

#### **FUWCODE** : ワイド文字フィールドのデフォルトのユーザーエンコード

FUWCODE では、ファイルのワイド文字フィールドがデータベースにロードされる場合にデフォルトで使用されるユーザーエンコードを定義します。このパラメータを省略すると、データベースの UWCODE 定義のエンコードが使用されます。

#### **FWCODE** : ワイド文字フィールドのエンコード

圧縮ファイルに W フォーマットのフィールド (ワイド文字フィールド) が存在する場合、FWCODE を指定してファイルエンコードを定義する必要があります。

FWCODE では、ワイド文字フィールドの最大長 (バイト単位) も決定します。

#### **HEADER**

このオプションパラメータでは、ADACMP 圧縮ロジックで、非圧縮入力レコード内のセグメント化された ADACMP レコードヘッダーを要求するかどうかを指定します。有効な値としては YES と NO があり、NO がデフォルトです。

HEADER=NO は、フォーマットが Adabas 8 より前の Adabas バージョンの ADACMP によって許可および生成されます。HEADER=NO が指定されている場合、入力レコードには処理対象ファイルのフィールドに関する非圧縮データのみが含まれている必要があります。各データレコードは、シーケンシャル入力データセットの 1 つの物理レコードに収まる必要があります (32 KB 未満)。データがこのサイズより大きい場合、エラーとなったレコードが DDFEHL エラーデータセットに書き込まれます。

HEADER=YES は、Adabas 8 を実行している場合だけ指定できます。HEADER=YES が指定されている場合、各入力レコードは ADAH または ADAC ヘッダーで始まり、ADACMP によって処理される論理レコードとセグメント化された物理レコードが関連付けられている必要があります。各論理レコードは、32 KB より大きくすることができます。各物理レコードのヘッダーでは、論理レコード内で後続データの位置を定義します。ADAH および ADAC ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAH および ADAC 内にあります。

HEADER=YES が指定された場合、セグメント化された非圧縮レコードが論理レコードへとアセンブルされるときにエラーが発生することがあります。ADAH ヘッダーでエラーになっ

た場合、ADAHレコードは書き込まれますが、後続のADACレコードは次のADAHレコードが処理されるまで書き込まれません。ADACヘッダーでエラーになった場合、先行するADAHヘッダーはペイロードデータなしで書き込まれます。エラーになったADACレコードは完全に書き込まれます。後続のADACレコードは、次のADAHレコードが処理されるまで書き込まれません。拒否されたレコードおよびその結果として返される可能性のあるレスポンスコードの詳細は、「**COMPRESS 機能の出力**」を参照してください。

HEADER パラメータと SPAN パラメータを混同しないでください。SPAN パラメータでは、圧縮レコードがデバイスのデータストレージのブロックサイズを超過した場合にそのレコード自体をスパンするかどうかを制御します。スパンドレコードには、標準スパンドレコードヘッダーが含まれています。このヘッダーは、HEADER パラメータで要求される ADAH または ADAC ヘッダーとは異なります。スパンドレコードの詳細については、「スパンドレコード」を参照してください。

#### LOBDEVICE : LOB ファイルのデバイスタイプ

このオプションパラメータでは、ADACMP COMPRESS 機能によって生成される LOB ファイルをロードするために使用するデータストレージデバイスタイプを指定します。ADACMP では、指定したデバイスのブロックサイズに基づいて、LB フィールド値がセグメントに分割されます。このパラメータは、FDT に 1 つ以上のラージオブジェクト (LB オプション) フィールドが含まれている場合にのみ有効です。

LOBDEVICE を指定しない場合、ADACMP COMPRESS DEVICE パラメータで指定されたデバイスタイプが使用されます。DEVICE パラメータでもデバイスが指定されていない場合、ADARUN DEVICE パラメータの値が使用されます。

#### LOBVALUES : LB フィールドサイズインジケータ

このオプションパラメータでは、(253 バイトを超える) 長い LB フィールド値または (253 バイトまでの) 短い LB フィールド値を ADACMP COMPRESS 入力データとして要求するかどうかを指定します。このパラメータに有効な値としては "YES" と "NO" があり、"NO" がデフォルトです。

このパラメータに "YES" が指定されている場合は、非圧縮入力データに、253 バイトを超える LB フィールド値を含めることができます。この場合、2 番目のシーケンシャル出力データセットも JCL での実行用に提供されます。この 2 番目のデータセットは、JCL で DD コントロールステートメント DDAUSB1 を使用して識別されます。このデータセットを使用して、253 バイトを超える LB フィールド値の圧縮 LB セグメントレコードが格納されます。

このパラメータに "NO" が指定されている場合は、非圧縮入力データに、253 バイトまでの LB フィールド、および LOB ファイルに格納されている LB フィールド値への参照を含めることができます。この場合、ADACMP COMPRESS FORMAT パラメータで LB フィールドを指定することはできません。ADACMP での処理時に、入力に含まれている短い LB フィールド値と LB フィールド値参照が出力に書き込まれます。



**Note:** LOBVALUES=NO を指定した ADACMP DECOMPRESS 機能に続いて LOBVALUES=NO を指定した ADACMP COMPRESS 機能を実行すると、LOB ファイルグループ内の基本ファイルの FDT を変更できます。

**LRECL** : 入力レコード長 (VSE のみ)

RECFM=F または RECFM=FB を指定する場合には、このパラメータも指定して入力データのレコード長 (バイト単位) を定義する必要があります。それ以外の場合は、LRECL を指定しないでください。

z/OS の場合、レコード長は入力データセットラベルまたは DD ステートメントから取得されます。

BS2000 の場合、レコード長はカタログエントリまたは /FILE ステートメントから取得されます。

**MAXLOGRECLEN** : バッファサイズ

このオプションパラメータでは、セグメント化された非圧縮物理レコードを圧縮論理レコードにアSEMBルするために ADACMP で使用されるバッファのサイズ (バイト単位) を指定できます。このバッファは HEADER=YES も指定した場合に限り、割り当てられます。それ以外の場合、MAXLOGRECLEN の設定は無視されます。MAXLOGRECLEN のデフォルト値は 1,048,576 バイト (1 MB) です。

MAXLOGRECLEN で指定された値に文字 "K" が付いた場合、この値は 1024 倍されます。最小値は、32768 バイトです。



**Note:** MAXLOGRECLEN は、非圧縮入力データのスパニングに関係します。圧縮レコードのスパニングに関係する SPAN パラメータと混同しないでください。

**MINISN** : 開始 ISN

MINISN パラメータでは、自動的に割り当てる ISN の開始値を指定します。MINISN パラメータを指定しない場合、デフォルトで 1 が割り当てられます。USERISN を指定した場合、MINISN は指定できません。

**MUPECOUNT** : バリュウカウントフィールドのサイズの指定

MUPECOUNT パラメータは、COMPRESS 機能用の入力レコードのバリュウカウントフィールドのサイズを指定します。構文は次のとおりです。

```
MUPECOUNT={1 | 2}
```

"1" が指定された場合、入力データで MU または PE 値に先行する各バリュウカウントフィールドは 1 バイトである必要があります。"191" を超えることはできません。"2" が指定された場合、入力データで MU または PE 値に先行する各バリュウカウントフィールドは 2 バイトである必要があります。MUPEX パラメータが指定されている場合のみ、バリュウカウントは 191 を超えることができます。MUPEX を指定すると、最大のカウントは "65,534" になります。

MUPEX パラメータが設定されている場合、MUPECOUNT のデフォルトは "2" になり、MUPEX パラメータが設定されていない場合、MUPECOUNT のデフォルトは "1" になります。



**Note:** このオプションは Adabas 8 より前のリリースと互換性がありません。したがって、191 を超える PE グループオカレンスを持ったレコードをロードした場合、以前のバージョンへの下位互換はありません。ただし Adabas 8 より前の Adabas バージョ

ンで作成された ADACMP データセットは、バージョン 8 の ADALOD ユーティリティを使用すると、正常にロードされます。

圧縮レコード内の 191 より多い MU および PE オカレンスを識別する方法については、「[圧縮レコード内の 191 を超える MU および PE オカレンスの特定](#)」を参照してください。

#### MUPEX : 拡張ピリオディックグループカウントの有効化

MUPEX パラメータでは、ファイルで拡張 MU/PE 制限 (191 を超えるオカレンス) を許容するかどうかを指定します。このオプションが指定されない場合、指定できる MU フィールドの最大値および PE フィールドの最大値は 191 です。それ以外の場合、最大値は 65,534 です。

MUPEX パラメータを設定する場合には、SPAN パラメータも同様に設定することを検討してください。これにより、追加の MU および PE フィールドを圧縮するとき、圧縮レコードのサイズ超過による圧縮エラーを回避することができます。



**Note:** このオプションは Adabas 8 より前のリリースと互換性がありません。したがって、191 を超える PE グループオカレンスを持ったレコードをロードした場合、以前のバージョンへの下位互換はありません。ただし Adabas 8 より前の Adabas バージョンで作成された ADACMP データセットは、バージョン 8 の ADALOD ユーティリティを使用すると、正常にロードされます。

圧縮レコード内の 191 より多い MU および PE オカレンスを識別する方法については、「[圧縮レコード内の 191 を超える MU および PE オカレンスの特定](#)」を参照してください。

#### NOUSERABEND : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ utility TERMINATED DUE TO ERROR CONDITION (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

#### NUMREC : 圧縮解除するレコード数

処理する入力レコードの数を指定します。このパラメータを省略すると、入力データセット内のすべての入力レコードが処理されます。

入力データセットに多数のレコードがある場合、ADACMP の初回実行時にこのパラメータを使用することをお勧めします。このパラメータを指定すると、フィールド定義エラーや無効な入力データで多くのレコードが拒否される場合に、全レコードに対して不要な処理が行われることを回避できます。また、このパラメータは、テスト用に小さいファイルを作成する場合にも役立ちます。

NUMREC=0 を指定すると、入力データセットはオープンされません。

#### PASSWORD : FDT ファイルのパスワード

FDT パラメータを指定し、そのファイルがパスワード保護されている場合、このパラメータを使用して、そのファイルに有効なパスワードを指定する必要があります。

**RECFM** : 入力レコードフォーマット (**VSE** のみ)

このパラメータを使用して、次のように入力レコードフォーマットを指定する必要があります。

F	固定長、非ブロック形式 (LRECL パラメータの指定も必要)。
FB	固定長、ブロック形式 (LRECL パラメータの指定も必要)。
V	可変長、非ブロック形式。
VB	可変長、ブロック形式。
U	不定長

z/OS の場合、レコードフォーマットは入力データセットラベルまたは DD ステートメントから取得されます。

BS2000 の場合、レコードフォーマットはカタログエントリまたは FILE ステートメントから取得されます。

**SPAN** : スパンドレコードの有効化

SPAN パラメータでは、レコードの圧縮後にその圧縮レコードがデバイスのデータストレージのブロックサイズを超過した場合に、レコードをスパンすることが可能になります。

**UACODE** : 英数字フィールド入力のユーザーエンコード

UACODE では、英数字フィールドのシーケンシャル入力のユーザーエンコードを定義します。UACODE を指定する場合は、FACODE も指定する必要があります。

**UARC** : 非圧縮ユーザーデータ入力のアーキテクチャ

UARC パラメータでは、非圧縮ユーザーデータのシーケンシャル入力のアーキテクチャを指定します。"ユーザーデータアーキテクチャキー" は、次の値を合計した整数です。

バイト順	b=0	上位バイト 1 番目
	b=1	下位バイト 1 番目
エンコードファミリ	e=0	ASCII エンコードファミリ
	e=2	EBCDIC エンコードファミリ (デフォルト)
浮動小数点フォーマット	f=0	IBM370 浮動小数点フォーマット
	f=4	VAX 浮動小数点フォーマット
	f=8	IEEE 浮動小数点フォーマット

デフォルトは  $ARC = b + e + f = 2$  です。つまり、上位バイト 1 番目、EBCDIC エンコードファミリ、および、IBM370 浮動小数点フォーマット ( $b=0$ ,  $e=2$ ,  $f=0$ ) です。

Intel386 PC のユーザーデータの場合は、例えば  $b=1$ ,  $e=0$ ,  $f=8$ 、または  $ARC=9$  のようになります。

**USERISN** : ユーザー ISN 割り当て

各レコードの ISN 値はユーザーが定義します。このパラメータを省略した場合、各レコードの ISN 値は Adabas によって割り当てられます。

USERISN を指定する場合は、各レコードに割り当てる ISN 値を各データレコードの直前に 4 バイトの 2 進数で付ける必要があります。MINISN パラメータを指定した場合、USERISN は指定できません。

USERISN が、HEADER=YES とともに指定された場合、ADAH ヘッダーの直後に ISN が論理レコードの一部として続きます。

ユーザー定義の ISN 値を含む固定長または不定長入力レコードのフォーマットは、次のとおりです。

`userisn/data`

ユーザー定義の ISN 値を含む可変長入力レコードのフォーマットは、次のとおりです。

`length/xx/userisn/data`

ここでは次の内容を表しています。

<code>length</code>	物理レコード長を示す 2 バイトの 2 進数 (レコードデータの長さ + 8 バイト)
<code>xx</code>	バイナリの 0 を含む 2 バイトフィールド
<code>userisn</code>	レコードに割り当てる ISN 値を示す 4 バイトの 2 進数
<code>data</code>	入力レコードデータ

ISN 値は任意の順序で指定できますが、(ファイルに対して) ユニークであり、かつファイルに設定した MAXISN 値を超えないようにする必要があります ([ADALOD ユーティリティ](#) のドキュメントを参照)。

ADACMP では、ISN 値がユニークかどうか、および MAXISN を超えているかどうかをチェックしません。これらのチェックは ADALOD ユーティリティで行われます。

**UWCODE** : ワイド文字フィールド入力のユーザーエンコード

UWCODE では、ワイド文字フィールドのシーケンシャル入力のユーザーエンコードを定義します。UWCODE パラメータを指定する場合は、FWCODE も指定する必要があります。

ユーザー入力のすべてのワイド文字フィールドが同じコードページにエンコードされます。1 回の ADACMP の実行でフィールドごとに異なるエンコードを指定することはできません。

## ADACMP COMPRESS 例

### 例 1

```

ADACMP COMPRESS
ADACMP FNDEF='01,AA,7,A,DE,FI'           Field AA
ADACMP FNDEF='01,AB,15,A,DE,MU,NU'       Field AB
ADACMP FNDEF='01,GA'                     Group GA
ADACMP FNDEF='02,AC,15,A,NU'             Field AC
ADACMP FNDEF='02,AD,2,P,FI'             Field AD
ADACMP FNDEF='02,AE,5,P,NU'             Field AE
ADACMP FNDEF='02,AF,6,W'                 Field AF
ADACMP COLDE='7,Y1=AF'                   Collation descriptor Y1
ADACMP SUBDE='BB=AA(1,4)'                 Subdescriptor BB
ADACMP SUPDE='CC=AA(1,4),AD(1,1)'         Superdescriptor CC
ADACMP HYPDE='1,DD,4,A,MU=AB,AC,AD'       Hyperdescriptor DD
ADACMP PHONDE='EE(AA)'                   Phonetic descriptor EE
ADACMP SUBFN='FF=AA(1,2)'                 Subfield FF
ADACMP SUPFN='GG=AA(1,4),AD(1,1)'         Superfield GG
    
```

フィールド AA	レベル1、7バイトの英数字、ディスクリプタ、固定ストレージオプションで定義されています。
フィールド AB	レベル1、15バイトの英数字、ディスクリプタ、マルチプルバリューフィールド、空値省略で定義されています。
GA	フィールド AC、AD、AE、および AF を含むグループです。
BB	サブディスクリプタです（フィールド AA のバイト位置 1~4）。
CC	スーパーディスクリプタです（フィールド AA のバイト位置 1~4、およびフィールド AD のバイト位置 1）。
DD	フィールド AB、AC、および AD で構成されるハイパーディスクリプタです。ハイパー出口 1 に割り当てられています。
EE	フィールド AA から派生したフォネティックディスクリプタです。
FF	サブフィールドです（フィールド AA のバイト位置 1~2）。
GG	スーパーフィールドです（AA のバイト位置 1~4、および AD のバイト位置 1）。
Y1	フィールド AF の照合ディスクリプタです。照合ディスクリプタユーザー出口 7（CDX07）に割り当てられています。

### 例 2

```

ADACMP COMPRESS
ADACMP FORMAT='AG,6,U,AF,4X,AA,'         input record format
ADACMP FORMAT='AB,AC'                     continuation of FORMAT statement
ADACMP FNDEF='01,AA,10,A,NU'              field definitions
ADACMP FNDEF='01,AB,7,U,NU'
ADACMP FNDEF='01,AF,5,P,NU'
ADACMP FNDEF='01,AG,12,P,NU,DE'
    
```

```
ADACMP FNDEF='01,AC,3,A,NU,DE'
```

入力レコードフォーマットがFORMATパラメータで明示されています。ADACMPでは、このフォーマットを入力レコードのフィールド処理のベースとして使用します。ファイルのフィールド定義テーブル（FDT）は、FNDEFステートメントに指定した構造に対応します。

### 例 3

```
ADACMP COMPRESS
ADACMP FORMAT='AG,AF,4X,AA,AB,AC'      input record format
ADACMP FDT=8                            FDT same as file 8
```

入力レコードフォーマットがFORMATパラメータで明示されています。使用するFDTは、Adabasファイル8に現在定義されているものと同じです。

### 例 4

```
ADACMP COMPRESS NUMREC=2000,USERISN
ADACMP FNDEF='01,AA,7,A,DE,FI'          Field AA
ADACMP FNDEF='01,AB,15,A,DE,MU,NU'      Field AB
```

処理する入力レコード数は2,000件に制限されています。各レコードのISN値はユーザーが指定します。

### 例 5

```
ADACMP COMPRESS RECFM=FB,LRECL=100
ADACMP FNDEF='01,AA,7,A,DE,FI'          Field AA
ADACMP FNDEF='01,AB,15,A,DE,MU,NU'      Field AB
```

VSE入力ファイルには、固定長（ブロック形式）レコードが含まれています。レコード長は100バイトです。

# 19 DECOMPRESS : Adabas ファイルの圧縮解除

---

■ オプションのパラメータとサブパラメータ .....	185
■ マルチクライアントファイルの圧縮解除 .....	105
■ ADACMP DECOMPRESS .....	106

DECOMPRESS 機能では、次のいずれかの方法でデータを圧縮解除します。

- ADAULD UNLOAD ユーティリティ機能によってアンロードされた出力から
- 1つの圧縮 Adabas ファイルから直接 (INFILE パラメータでファイル番号が指定されている場合)

INFILE ファイルから直接データを圧縮解除する場合、DECOMPRESS では最初に ADAULD UNLOAD/MODE=SHORT 機能を実行します。これにより、ADAULD と ADACMP DECOMPRESS 処理を別々に実行するよりも、時間を短縮できます。

```

ADACMP DECOMPRESS [CODE = cipher-code ]
                    [FORMAT = output-record-format-definition ]
                    [HEADER = { YES | NO } ]
                    [INFILE = file-number ]
                    [ETID = owner-id ]
                    [LPB = { prefetch-buffer-size | based-on-ADARUN-lu } ]
                    [PASSWORD = ' password ' ]
                    [SORTSEQ = { descriptor [,NU] | ISN | physical-sequence } ]
                    [UTYPE = { EXF | EXU } ]
                    [ISN]
                    [LOBVALUES = { YES | NO } ]
                    [MAXLOGRECLN = buffer-size ]
                    [NOUSERABEND]
                    [NUMREC = { number-of-records | all-records } ]
                    [TRUNCATE]
                    [UACODE = userdata-alpha-key ]
                    [UWCODE = userdata-wide-key ]
                    [UARC = { architecture-key | 2 } ]

```

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

### CODE : サイファコード

圧縮解除するファイルが暗号化されている場合、このパラメータでそのサイファ (暗号) コードを指定する必要があります。暗号化についての詳細は『*Adabas Security Manual*』を参照してください。

### ETID : マルチクライアントファイルオーナー ID

ETIDでは、INFILEで指定されたマルチクライアントファイルのオーナーIDを指定します。ADACMP DECOMPRESSでは、ETIDで指定されたオーナーIDに割り当てられているマルチクライアントファイル内のレコードのみを選択して圧縮解除します。ETID値は、レコー

ドがマルチクライアントファイルにロードされたときに割り当てられていた値と同じである必要があります。

#### FORMAT : 出力レコードのフォーマット定義

FORMAT では、FDT で指定されたフォーマット以外のフォーマットに圧縮解除できます。このパラメータを使用して、既存のファイルの FDT、および特にピリオディック (PE) グループの構造を変更できます。

FORMAT パラメータの構文は、読み込みコマンドに使用するフォーマットバッファ構文と同じです。ただし、テキストは挿入できません (テキストは圧縮/圧縮解除できません)。詳細は、『Adabas コマンドリファレンス』を参照してください。



**Note:** FORMAT パラメータでは、圧縮解除時にすべての関連データフィールドが処理されたかどうかはチェックされません。

例えば、次のように定義されたマルチプルバリュー (MU) フィールドがあるとします。

```
01,AA,8,A,MU
```

このフィールドには 5 つのオカレンスが含まれ、ADACMP DECOMPRESS FORMAT パラメータが指定されています。

```
AA1-4
```

ここで、5 番目のフィールド値についての指定がない場合、先頭の 4 つの AA フィールド値のみが圧縮解除されます。これは、PE フィールドのオカレンスおよび長さの変更にも適用されます。

#### HEADER

このオプションパラメータでは、ADACMP 圧縮解除ロジックで、セグメント化された ADACMP レコードヘッダー (ADAH および ADAC) を非圧縮出力の一部として生成するかどうかを指定します。有効な値としては YES と NO があり、NO がデフォルトです。

HEADER=NO は、フォーマットが Adabas 8 より前の Adabas バージョンの ADACMP によって許可および生成されます。HEADER=NO を指定した場合、ADACMP によって生成される非圧縮出力レコードには、処理対象ファイルのフィールドに関するデータのみが含まれます。各データレコードは、シーケンシャル入力データセットの 1 つの物理レコードに収まる必要があります (32 KB 未満)。データがこのサイズより大きい場合、エラーとなったレコードが DDFEHL エラーデータセットに書き込まれます。

HEADER=YES は、Adabas 8 を実行している場合だけ指定できます。HEADER=YES を指定した場合、ADACMP によって生成される各非圧縮出力レコードは ADAH または ADAC ヘッダーで始まり、論理レコードと物理レコードが関連付けられます。各論理レコードは、32 KB より大きくすることができます。各物理レコードのヘッダーでは、論理レコード内で後続データの位置を定義します。ADACMP でセグメント化されたレコードの詳細は、「[セグメント化されたレコードに関する考慮事項](#)」を参照してください。ADAH および ADAC ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAH および ADAC 内にあります。

**INFILE** : 圧縮解除するファイルの番号

INFILEパラメータを使用すると、ADAULDユーティリティでファイルのアンロードを行わなくても、ファイルを圧縮解除できます。INFILEパラメータを指定しない場合、入力は一シケンシャル (DD/EBAND) ファイルから読み込まれます。INFILEパラメータをETIDパラメータとともに指定した場合、マルチクライアントファイル内のレコードを選択的に圧縮解除できます。マルチクライアントファイルを圧縮解除する場合は、「[マルチクライアントファイルの圧縮解除](#)」セクションを参照してください。

**ISN** : 非圧縮出力に ISN を含める

各レコードの ISN 値が各非圧縮レコードとともに出力されます。このパラメータを省略した場合、各レコードの ISN 値は出力されません。

ISNをHEADER=YESとともに指定した場合、ADAHヘッダーの直後にISN値が論理レコードの一部として続きます。ADAHヘッダーのDSECTは、分散Adabas MVSSRCEデータセットのメンバADAH内にあります。

**LPB** : プリフェッチバッファサイズ

LPBでは、ADACMP DECOMPRESS INFILE機能に対する内部プリフェッチバッファのサイズ(バイト単位)を指定します。指定可能最大値は、32,760バイトです。デフォルト値は、ニュークリアスに指定されたADARUNのLU値に応じて、Adabasによって算出されます。

**LOBVALUES** : LB フィールドサイズインジケータ

LOBVALUESパラメータは、ラージオブジェクト(LB)フィールドを含むファイルに対してのみ指定する必要があります。

このオプションパラメータでは、(253バイトを超える)長いLBフィールド値または(253バイトまでの)短いLBフィールド値をADACMP DECOMPRESS出力データとして要求するかどうかを指定します。このパラメータに有効な値としては"YES"と"NO"があり、"NO"がデフォルトです。

このパラメータに"NO"が指定されている場合は、非圧縮出力データに、253バイトまでのLBフィールド、およびLOBファイルに格納されているLBフィールド値への参照を含めることができます。この場合、ADACMP DECOMPRESS FORMATパラメータでLBフィールドを指定することはできません。ADACMP DECOMPRESSでの処理時には、基本ファイルのレコードのみが入力として読み込まれます。基本ファイルにLOBファイル内のLBフィールド値への参照が含まれている場合、それらの参照は読み込まれず、出力に単に複製されます。

このパラメータに"YES"が指定されている場合、非圧縮出力データに、レコードのLBフィールド値を含めることができます。この場合、INFILEパラメータも指定して、圧縮解除するデータを含むLOBファイルの基本ファイルのファイル番号を識別する必要があります。処理中にADACMP DECOMPRESSが基本ファイルからレコードを読み取り、圧縮解除すると、参照されているすべてのLBフィールド値をLOBファイルから読み取ります。



**Note:** LOBVALUES=NOを指定したADACMP DECOMPRESS機能に続いて

LOBVALUES=NOを指定したADACMP COMPRESS機能を実行すると、LOBファイルグループ内の基本ファイルのFDTを変更できます。

**MAXLOGRECLN** : バッファサイズ

このオプションパラメータを使用すると、非圧縮出力データの物理レコードを1つ以上スパンしている論理レコードをアSEMBLするために ADACMP で使用されるバッファのサイズ (バイト単位) を指定できます。このバッファは HEADER=YES も指定した場合に限り、割り当てられます。それ以外の場合、MAXLOGRECLN の設定は無視されます。MAXLOGRECLN のデフォルト値は 1,048,576 バイト (1 MB) です。

MAXLOGRECLN で指定された値に文字 "K" が付いた場合、この値は 1024 倍されます。最小値は、32768 バイトです。

**NOUSERABEND** : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ utility TERMINATED DUE TO ERROR CONDITION (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**NUMREC** : 圧縮解除するレコード数

NUMREC では、処理する入力レコードの数を指定します。このパラメータを省略すると、入力データセット内のすべての入力レコードが処理されます。

入力データセットに多数のレコードがある場合、ADACMP の初回実行時に NUMREC を使用することをお勧めします。このパラメータを指定すると、フィールド定義エラーや無効な入力データが原因で多くのレコードが拒否される場合に、全レコードに対して不要な処理が行われることを回避できます。また、NUMREC は、テスト用の小さなファイルの作成にも役立ちます。

**PASSWORD** : INFILE のパスワード

パスワード保護された Adabas ファイルから直接ファイルを圧縮解除する場合、PASSWORD パラメータで適切なパスワードを指定する必要があります。

**SORTSEQ** : INFILE ファイルの処理順序

SORTSEQ は、ファイルの処理順序を決めます。このパラメータを省略すると、レコードが物理順で処理されます。SORTSEQ は、INFILE が指定されている場合にのみ指定できます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューディスクリプタフィールド、またはピリオディックグループ内のディスクリプタは使用しないでください。

ディスクリプタ名が、空値省略 (NU) オプションで定義されたフィールドを参照している場合、ディスクリプタ名の後に "NU" を指定する必要があります。この場合、空値を含むディスクリプタのレコードは圧縮解除されません。この場合に NU を指定しないと (デフォルト)、ADACMP で NU ディスクリプタが拒否されます。

 **Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない (値を持たない) 場合は、空値省略のディスクリプタフィールドでなくても、そのレコー

ドはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

ISN を指定すると、ファイルが ISN 昇順に処理されます。また、Adabas チェックポイントまたはセキュリティファイルについては、SORTSEQ=ISN しか許可されません。

**TRUNCATE** : 余分な英数文字の切り捨て

TRUNCATE パラメータでは、圧縮解除時における圧縮英数文字データの文字の切り捨てを可能にします。TRUNCATE を指定した場合、FDT 定義より長い文字を含む英数文字フィールドが ADACMP DECOMPRESS 処理時に検出されると、余分な文字が切り捨てられます。

TRUNCATE を指定しない場合、余分な文字を含む英数文字レコードは DDFEHL データセットに書き込まれます。英数文字以外のフィールドは切り捨ての対象とはなりません。

**UACODE** : 英数文字フィールド出力のエンコードプロトコル

UACODE では、英数文字フィールドのシーケンシャル出力のエンコードを定義します。このパラメータを使用すると、圧縮されたシーケンシャル入力のヘッダーで引き渡された英数文字フィールドのユーザーエンコードを変更できます。

**UARC** : 非圧縮ユーザーデータ出力のアーキテクチャ

UARC パラメータでは、非圧縮ユーザーデータのシーケンシャル出力のアーキテクチャを指定します。このパラメータを使用すると、圧縮されたシーケンシャル入力のヘッダーで引き渡されたユーザーエンコードを変更できます。

"ユーザーデータアーキテクチャキー" は、次の値を合計した整数です。

バイト順	b=0	上位バイト 1 番目
	b=1	下位バイト 1 番目
エンコードファミリ	e=0	ASCII エンコードファミリ
	e=2	EBCDIC エンコードファミリ (デフォルト)
浮動小数点フォーマット	f=0	IBM370 浮動小数点フォーマット
	f=4	VAX 浮動小数点フォーマット
	f=8	IEEE 浮動小数点フォーマット

デフォルトは  $ARC = b + e + f = 2$  です。つまり、上位バイト 1 番目、EBCDIC エンコードファミリ、および、IBM370 浮動小数点フォーマット (b=0、e=2、f=0) です。

Intel386 PC のユーザーデータの場合は、例えば b=1、e=0、f=8、または  $ARC=9$  のようになります。

**UTYPE** : ユーザータイプ

INFILE パラメータで指定したファイルのアンロード処理時に有効にするユーザータイプを指定します。次のいずれかの値を指定できます。

EXF	ファイルの他ユーザーには、アクセスおよび更新を許可しません。
EXU	ファイルの他ユーザーには、アクセスのみを許可します。EXU がデフォルトです。

**UWCODE** : ワイド文字フィールド出力のエンコードプロトコル

UWCODE では、ワイド文字フィールドのシーケンシャル出力のエンコードを定義します。このパラメータを使用して、圧縮されたシーケンシャル入力のヘッダーで引き渡されたワイド文字フィールドのユーザーエンコードを変更できます。

## マルチクライアントファイルの圧縮解除

ADACMP では、Adabas データを圧縮解除して、シーケンシャルユーザーファイルに出力します。INFILE パラメータでマルチクライアントファイルを指定し正しい ETID 値を指定すれば、DECOMPRESS 機能は自動的に選択したレコードを圧縮解除します。

DECOMPRESS 機能は、オーナー ID (ある場合) をスキップします。マルチクライアントファイルの DECOMPRESS 処理の出力には、オーナー ID も ETID 情報も含まれていません。

DECOMPRESS 機能で処理するマルチクライアントファイルを INFILE パラメータで指定する場合、ETID パラメータを使用して圧縮解除を特定ユーザーのレコードのみに制限することが可能です。この場合は、特定ユーザーのレコードのみが ADACMP に読み込まれ、圧縮解除されます。マルチクライアントファイルの圧縮解除時に ETID パラメータを指定しなければ、ファイル内のすべてのレコードが圧縮解除されます。

## 例

ファイル 20 内の USER1 が所有するレコードのみが圧縮解除され、シーケンシャル出力ファイルに出力されます。

```
ADACMP  DECOMPRESS  INFILE=20,ETID=USER1
```

## ADACMP DECOMPRESS

---

### 例 1

DECOMPRESS 機能を実行します。入力データセットには、ADAULDユーティリティの前回実行時の出力を使用します。

```
ADACMP DECOMPRESS
```

### 例 2

Adabas ファイル 23 を圧縮解除します。非圧縮出力に、各レコードの ISN 値を含めます。

```
ADACMP DECOMPRESS INFILE=23,ISN
```

## 20 フィールド定義ステートメント

---

▪ FNDEF：フィールド定義およびグループ定義 .....	108
▪ FNDEF：ピリオディックグループ定義 .....	125
▪ COLDE：照合ディスクリプタ定義 .....	128
▪ HYPDE：ハイパーディスクリプタ定義 .....	131
▪ PHONDE：フォネティックディスクリプタ .....	135
▪ SUBDE：サブディスクリプタ定義 .....	136
▪ SUBFN：サブフィールド定義 .....	138
▪ SUPDE：スーパーディスクリプタ定義 .....	139
▪ SUPFN：スーパーフィールド定義 .....	148

ADACMP の入力として指定するフィールド定義は、次の目的で使用されます。

- 入力レコードに含まれる各フィールドの長さフォーマットを指定します。これにより ADACMP は、編集および圧縮時に正しいフィールド長とフォーマットを判断できます。
- ファイルのフィールド定義テーブル (FDT) を作成します。このテーブルは、Adabas コマンド実行中にファイル内のすべてのフィールド (またはグループ) の論理構造と特徴を判断するために Adabas が使用します。

フィールド定義を入力する場合は、次の構文に従ってください。最小 1 個、最大 926 個の定義を指定できます。

フィールドとグループ	FNDEF=' level , name [ , length , format ] [ , MU [ ( occurrences ) ] ] [ , option ] ... '
ピリオディックグループ	FNDEF=' level , name [ , PE [ ( occurrences ) ] ] '
照合ディスクリプタ	COLDE=' number , name [ , UQ [ , XI ] ] = parentfield '
ハイパーディスクリプタ	HYPDE=' number , name , length , format [ { , option } ... ] = { parentfield } , ... '
フォネティックディスクリプタ	PHONDE=' name ( field ) '
サブディスクリプタ	SUBDE=' name [ , UQ [ , XI ] ] = parentfield ( begin , end ) '
サブフィールド	SUBFN=' name = parentfield ( begin , end ) '
スーパーディスクリプタ	SUPDE=' name [ , UQ [ , XI ] ] = parentfield ( begin , end ) } , ... '
スーパーフィールド	SUPFN=' name = parentfield ( begin , end ) [ , parentfield ( begin , end ) ] ... '

各定義の右側にユーザーコメントを入力することができます。ユーザーコメントと定義の間には少なくとも 1 つの空白を指定する必要があります。

これらの各定義タイプについては、この chapter で説明されています。

## FNDEF：フィールド定義およびグループ定義

FNDEF パラメータでは、Adabas フィールド定義またはグループ定義を指定できます。フィールド定義およびグループ定義のエントリを構成するには、次の構文を使用します。

```
FNDEF = 'level, name [ , length, format ] [, MU [(occurrences)] ] [ , option ] ... '
```

レベル番号と名前は必須です。定義エントリの間には空白をいくつ入れても構いません。

各 FNDEF パラメータについては、このsectionで説明されています。

- level
- name
- length
- format
- occurrences
- フィールドオプション

MU パラメータについては、「[フィールドオプション](#)」で説明されています。

### level

レベル番号は 01～07 の範囲内の 1 桁または 2 桁の数字（先行のゼロは任意）を使用し、フィールドグルーピングと合わせて使用します。02 またはそれ以上のレベル番号のついたフィールドは、それより小さいレベル番号のついた直前のグループの一部であるとみなされます。

グループで定義すると、グループ名を使用して一連のフィールド（または1つのフィールドだけでもよい）を参照することができます。これは一連の連続フィールドを参照するのに便利で効果的な方法です。

レベル番号 01～06 は、グループの定義に使用することができます。グループは他のグループを含むことができます。ネスト構造のグループにレベル番号をつけるときは、レベル番号を飛ばしてはなりません。

次の例では、フィールド A1 および A2 はグループ GA に含まれます。フィールド B1 とグループ GC（フィールド C1 と C2 で構成）はグループ GB に含まれます。

FNDEF='01,GA'	グループ	
FNDEF='02,A1,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='02,A2,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='01,GB'	グループ	
FNDEF='02,B1,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='02,GC'	グループ（ネスト構造）	
FNDEF='03,C1,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='03,C2,...'		エレメンタリまたはマルチプルバリュースフィールド

**name**

フィールド（またはグループ）に割り当てる名前を指定します。

名前はファイル内でユニークでなければなりません。名前は2文字で指定しなければならず、最初の文字は英字で、2番目の文字は英字か数字でなければなりません。特殊文字は使用できません。

値 E0～E9 は編集マスク用に予約されており、使用してはなりません。

正しい名前例	誤りの名前例
AA	A（2文字ではない）
B4	E3（編集マスク）
S3	F*（特殊文字が含まれている）
WM	6M（最初の文字が英字ではない）

**length**

フィールドの長さ（バイト単位）を指定します。長さの値を使用して、次のことが行われます。

- 各入力レコード内に存在するフィールドの長さを ADACMP に示します。
- コマンド処理中に Adabas で使用される標準（デフォルト）長を定義します。

この標準長はフィールド定義テーブル（FDT）に格納され、フィールドの読み込み／更新時に、ユーザーから特に長さの指定がない場合に使用されます。

指定できる最大フィールド長は、フォーマット値に応じて異なります。

フォーマット	最大長
英数字（A）	253 バイト
2進数（B）	126 バイト
固定小数点（F）	4 バイト（常に2または4バイト）
浮動小数点（G）	8 バイト（常に4または8バイト）
パック10進数（P）	15 バイト
アンパック10進数（U）	29 バイト
ワイド文字（W）	253 バイト*

\*FWCODE 属性値によっては、W フィールドの最大長が253 バイトより小さくなる場合があります。例えば、FWCODE にデフォルト値（つまり、Unicode）を使用した場合、最大長は252 バイト（1文字が2バイト）になります。

標準長をグループ名に指定することはできません。

FI オプションを使用しない限り、標準長によってフィールド値のサイズが制限されることはありません（「[FI：固定ストレージ](#)」を参照）。読み込みまたは更新コマンドで、そのフォーマットに許される最大長までのフィールド長を指定して、フィールドの標準長を変更することができます。

フィールドの標準長がゼロの場合、そのフィールドは可変長フィールドとみなされます。可変長フィールドには標準（デフォルト）長はありません。固定小数点（F）フィールドに対して長さ指定を行う場合、2バイトまたは4バイトの長さのみを指定できます。浮動小数点（G）フィールドに対しては、4バイトまたは8バイトの長さのみを指定できます。

Adabas コマンドで可変長フィールドを長さ指定なしで参照すると、フィールド値の先頭に値の長さ（長さバイトを含む）を示す1バイトの2進数フィールドが付けられて返されます。この長さの値は、フィールド更新時に指定する必要があります。また、ADACMPで処理される入力レコード内のフィールドに対しても指定する必要があります。ロング英数字（LA）オプションを使用して定義されたフィールドの場合、フィールド値の先頭に値の長さ（長さバイトを含む）を示す2バイトの2進数フィールドが付けられます。

## format

フィールドの標準フォーマットを指定します（1文字のコードで表す）。

A	英数字（左詰め）
B	2進数（右詰め、符号なし、正の数）
F	固定小数点（右詰め、符号あり、2の補数表示）
G	浮動小数点（一般形式、符号あり）
P	パック10進数（右詰め、符号あり）
U	アンパック10進数（右詰め、符号あり）
W	ワイド文字（左詰め）

標準フォーマットは次の目的で使用されます。

- 各入力レコード内に存在するフィールドのフォーマットを ADACMP に示します。
- コマンド処理中に Adabas で使用される標準（デフォルト）フォーマットを定義します。この標準フォーマットはフィールド定義テーブル（FDT）に格納されており、フィールドの読み込み／更新時に、ユーザーが特にフォーマットを指定していない場合に使用されます。

標準フォーマットはフィールドに必ず指定しなければなりません。グループ名に指定することはできません。グループを読み込む（書き込む）とき、グループ内のフィールドは必ず個々の各フィールドの標準フォーマットに従って返されます（指定する必要があります）。指定したフォーマットに基づいて、フィールドに対する圧縮処理のタイプが決定されます。

固定小数点フィールドの長さは2バイトまたは4バイトです。正の値は一般的な形式で示され、負の値は2の補数で示されます。

## フィールド定義ステートメント

浮動小数点フォーマットで定義されたフィールドの長さは、4バイト（単精度）または8バイト（倍精度）で指定します。浮動小数点として定義されたフィールドの値を他のフォーマットに変換することは可能です。

バイナリフィールドをディスクリプタとして定義する場合、またはフィールドに正の数と負の数の両方が含まれている可能性がある場合は、Bフォーマットの代わりにFフォーマットを使用する必要があります。これは、Bフォーマットではすべての値が符号なし（正の数）とみなされるためです。

ワイド文字フィールドは、英数字フィールドと同様に、FDTで定義された標準長（バイト単位）または可変長になります。ワイド文字フィールドを可変長フォーマット以外で書き換える場合は、ユーザーエンコードとの互換性が必要です。例えば、Unicodeのユーザーエンコードでは、偶数の長さが要求されます。数値（U、P、B、F、G）からワイド文字フォーマットへの変換は許可されていません。

### occurrences

MUPEX オプションを指定している場合に、レコード内に含める MU フィールドのオカレンスの数を指定します。これは任意のパラメータです。

### フィールドオプション

オプションは2文字のコードで指定します。複数のコードを指定することができます（フィールドで適用可能な場合）。これらのコードは、コンマ（,）で区切って任意の順番で指定できます。

次の表は、フィールド定義およびグループ定義に使用できるオプションを示しています。特定のオプションの詳細を確認するには、表内でオプション名をクリックします。



**Note:** PEオプションは、別のAdabasフィールドオプションです。ただし、このオプションは、ピリオディックグループを指定する場合にのみ、使用することができます。詳細は、「[FNDEF：ピリオディックグループ定義](#)」を参照してください。

コード	説明
<a href="#">DE</a> ：ディスクリプタ	フィールドがディスクリプタ（キー）であることを示します。
<a href="#">FI</a> ：固定ストレージ	フィールド長が固定ストレージ長であることを示します。値は内部的な長さバイトなしで格納され、圧縮されません。また、定義したフィールド長を超えない必要があります。
<a href="#">LA</a> ：ロング英数字フィールド	このAまたはWフォーマットの可変長フィールドには、長さが16,381バイトまでの値を格納できます。
<a href="#">LB</a> ：ラージオブジェクトオプション	英数字フィールドには、2,147,483,643（約2 GB）までのデータを格納できます。
<a href="#">MU</a> ：マルチプルバリュースフィールド	フィールドには1レコードあたり65,534個までの値を格納できます。

コード	説明
<b>NB</b> ：空白圧縮オプション	LA および LB フィールドのどちらからも、末尾の空白を削除（圧縮）するべきではありません。
<b>NU</b> ：空値省略	フィールドの空値は省略されます。
<b>NV</b> ：変換なし	この A または W フォーマットのフィールドは、レコードバッファ内で変換されることなく処理されます。
<b>UQ</b> ：ユニークディスクリプタ	フィールドがユニークディスクリプタであることを示します。つまり、ファイル内の各レコードで、ディスクリプタに異なる値が格納される必要があります。
<b>XI</b> ：除外インスタンス番号	このフィールドでは、PE に設定された UQ オプションからインデックス（オカレンス）番号が除外されます。
<b>NC</b> ：SQL 空値オプション	<p>Adabas には、<b>NC</b> および <b>NN</b> の 2 つのデータ定義オプションが用意されています。これらのオプションを使用して、Software AG のメインフレーム Adabas SQL Gateway (ACE) および他の SQL (Structured Query Language) データベース問い合わせ言語に対応する SQL 互換の空値表現を指定することができます。これらのオプションの詳細は、「<a href="#">SQL 空値の表現</a>」を参照してください。</p> <p><b>NC</b> オプションを使用した場合、フィールドには、SQL 解釈で値のないフィールドを示す空値を格納することができます。つまり、フィールドの値は定義されません（カウントされません）。</p> <p><b>NN</b> オプションを使用する場合、NC オプションで定義されたフィールドには、常に値が定義されている必要があります。つまり、SQL 空値を格納することはできません。</p>
<b>NN</b> ：SQL 非空値オプション	

## DE：ディスクリプタ

DE は、フィールドがディスクリプタ（キー）であることを示します。エントリがフィールドのアソシエータインバーテッドリストに作成されることにより、該当フィールドは FIND コマンドのソートキーとして検索式で使用したり、論理的な順次読み込み処理を制御する方法として使用したり、ファイルカップリングの基準として使用したりできるようになります。

ディスクリプタオプションは慎重に使用する必要があります。特に、ファイルが大きく、ディスクリプタとみなされるフィールドが頻繁に更新される場合は、注意が必要です。

ディスクリプタフィールドの定義はレコード構造から独立していますが、ディスクリプタフィールドがレコードの最初に配列されず、論理的に物理レコードの終了を超えている場合、パフォーマンスの理由から、そのレコードに対するインバーテッドリストエントリが生成されないことに注意してください。このケースでインバーテッドリストエントリを生成するには、ファイルのアンロード（SHORT モード）、圧縮解除、そして再ロードが必要です。あるいは、アプリケーションプログラムを使用してフィールドをファイルの各レコードの最初に再配列します。

## FI：固定ストレージ

FI では、フィールド長が固定ストレージ長であることを示します。フィールドの値は内部的な長さバイトなしで格納され、圧縮されません。また、定義されたフィールド長を超えない必要があります。

空値を含む可能性の低い1バイトまたは2バイトの長さのフィールド（社員番号、性別など）、および圧縮できない値を含むフィールドに対しては、FI オプションを使用することをお勧めします。

マルチプルバリューフィールドやピリオディックグループ内のフィールドに対して FI オプションを使用することはお勧めしません。このようなフィールドの空値は省略（圧縮）されないため、ディスクストレージスペースが無駄になるだけでなく、処理時間も増加します。

FI オプションは、次のフィールドに対しては指定できません。

- U フォーマットのフィールド
- NC、NN、または NU オプションフィールド
- FNDEF ステートメントで長さがゼロ（0）に定義された可変長フィールド
- ピリオディック（PE）グループ内のディスクリプタ

FI オプションで定義されたフィールドは、フィールドの標準長を超える値によって更新することはできません。

### FI の使用例

	定義	ユーザーデータ	内部表現
FI オプションなし	FNDEF='01,AA,3,P'	33104C 00003C	0433104F (4 バイト) 023F (2 バイト)
FI オプションあり	FNDEF='01,AA,3,P,FI'	33104C 00003C	33104F (3 バイト) 00003F (3 バイト)

## LA：ロング英数字フィールド

LA（ロング英数字）オプションは、可変長の英数字フィールドおよびワイド文字フォーマットのフィールド（つまり、フィールド定義 FNDEF で長さがゼロに定義された A または W フォーマットのフィールド）に対して指定できます。LA オプションを指定したフィールドには、長さが 16,381 バイトまでの値を格納できます。

LA オプションを含む英字またはワイド文字フィールドは、このオプションを含まない英字またはワイド文字フィールドと同様に圧縮されます。LA オプションを含むフィールドで実際に可能な最大長は、圧縮レコードを格納するブロックのサイズで制限されます。

LA オプションが指定されているフィールドが更新される（または読み込まれる）場合、そのフィールド値はレコードバッファに指定され（または返され）ます。このとき、フィールド値の先頭には、値の長さ（長さバイトを含む）を示す2バイトが付けられます（フィールド長+2）。

LA オプションが指定されたフィールドに関する考慮事項を以下に示します。

- NU、NC/NN、NV、または MU オプションと一緒に指定できます。
- PE グループのメンバにすることができます。
- FI オプションと一緒に指定できません。
- ディスクリプタフィールドにすることはできません。
- サブ/スーパーフィールド、サブ/スーパーディスクリプタ、ハイパーディスクリプタ、またはフォネティックディスクリプタの親にすることはできません。
- サーチバッファには指定できません。指定すると、レスポンスコード 61 が返されます。

詳細は、「フォーマットバッファでの LA (ロング英数字) フィールドのフィールド長の指定」および「レコードバッファでの LA (ロング英数字) フィールドのフィールド長の指定」を参照してください。

#### LA の使用例

	定義	ユーザーデータ	内部表現
LA オプションなし	FNDEF='01,BA,0,A'	X'06',C'HELLO' --	X'06C8C5D3D3D6' (1 バイトの長さ) --
LA オプションあり	FNDEF='01,BA,0,A,LA'	X'0007',C'HELLO' X'07D2',C' ... (2000 データバイト) ...'	X'06C8C5D3D3D6' (1 バイトの長さ) X'87D2 ... (2000 データバイト) ...'

#### LB : ラージオブジェクトフィールドオプション

フィールドにラージオブジェクト (LB) オプションを指定すると、ラージオブジェクトフィールドにすることができます。LB フィールドには、2,147,483,643 バイト (約 2 GB) までのデータを格納できます。現時点では、LB フィールド全体を格納し、取り出せるだけで、LB フィールドの一部を格納し、取り出すことはできません。

LB フィールドのフォーマットは "A" (英数字) であり、そのデフォルトフィールド長は現在、0 として定義されている必要があります。

LB フィールドは、次のものにすることはできません。

- ディスクリプタ、または特別なディスクリプタ (フォネティック、サブ、スーパー、またはハイパーディスクリプタ) の親
- FI または LA オプションとともに定義する。

LA および LB のどちらとしてフィールドを定義するかを決定する場合は、「LA および LB フィールドの違い」を参考にしてください。

- サーチバッファまたはフォーマットバッファのフォーマット選択条件で、指定されたフィールド

LB フィールドは次のいずれかです。

- MU、NB、NC、NN、NU、または NV オプションのいずれかで定義されたフィールド
- 単一グループまたは PE グループの一部にする。

LB フィールド定義に NB（非空白圧縮）フィールドオプションがあるかどうかは、文字を含む LB フィールドで末尾の空白が Adabas により削除されたかどうかを示します。

バイナリと文字の両データを含む LB フィールドがサポートされています。NV および NB オプションの両方で定義された LB フィールドには、バイナリラージオブジェクトデータを格納できます。これは、Adabas ではバイナリ LB フィールドが変更されることがないためです。格納されているのと同じ LB バイナリバイト文字列が、LB フィールドが読み込まれたときに取り出されます。また、バイナリ値を含む LB フィールドは NV および NB オプションで定義されているため、Adabas では、何らかの文字コードページに従って LB フィールドバイナリ値が変換されることも、バイナリ値を含む LB フィールドで末尾の空白が削除されることもありません。



**Note:** バイナリ値を含む LB フィールドの定義には、フォーマット B は使用されません。これは、一部の環境ではフォーマット B が、バイト順の異なるバイトスワップを意味するためです。バイトスワッピングは、バイナリ LB フィールドに適用されません。

次の表は、LB フィールドの FDT 定義の有効な例を示しています。

FDT の指定	説明
1, L1, 0, A, LB, NU	フィールド L1 は、空値省略の文字型のラージオブジェクトフィールドです。
1, L2, 0, A, LB, NV, NB, NU, MU	フィールド L2 は空値省略、マルチプルバリュー、バイナリフィールド型のラージオブジェクトフィールドです。

LB フィールドを扱うコマンドの対象は、LOB ファイルグループの基本ファイルにする必要があります。LOB ファイルに対するユーザーコマンドは拒否されます。

LB フィールドの基本的な情報については、「ラージオブジェクト (LB) フィールドの基本」を参照してください。

### MU：マルチプルバリューフィールド

マルチプルバリューフィールドオプションでは、フィールドの同一レコードに複数の値を格納できることを示します。MUPEX パラメータを指定し、MUPECOUNT パラメータを "2" に設定した場合、フィールドには 1 レコードあたり 65,534 までの値を格納できます。これらのパラメータを設定しない場合、フィールドには 1 レコードあたり 191 までの値を格納できます。ADACMP の各入力レコードには、最低 1 つの値（空値でも可）が含まれている必要があります。

値は、フィールドに指定された他のオプションに従って格納されます。最初の値の先頭には、フィールドに現在存在する値の数を示すカウントフィールドが付けられます。格納される値の数は、ADACMP 入力レコードに含まれる値の数に、フィールドを後で更新したときに追加され

た値を加え、省略された値を差し引いた数と等しくなります（このことは、フィールドが NU オプションを指定して定義されている場合にのみ適用されます）。

ADACMP の各入力レコードに含まれる値の数が一定の場合、その数を MU 定義ステートメントで MU(*n*) の形式で指定できます。*n* は各入力レコードに含まれる値の数です。次の例では、各入力レコードにマルチプルバリュースフィールド AA の 3 つの値が含まれています。

```
FNDEF='01,AA,5,A,MU(3)'
```

値にゼロ (MU(0)) を指定した場合、入力レコードにはマルチプルバリュースフィールドの値が含まれていないことを示します。

値の数がすべての入力レコードで一定でない場合、各入力レコードの最初の値の先頭に、そのレコードに含まれる値の数を示す 1 バイトまたは 2 バイトの 2 進数のカウントフィールド (**MUPECOUNT** パラメータの設定によって異なる) が付けられます（「[入力データの必要条件](#)」も参照）。

FDT を指定している場合 (**FDT パラメータ**の説明を参照)、フィールドカウントは各入力レコードに 1 バイトまたは 2 バイトのバイナリ値として含まれる必要があります (**MUPECOUNT** パラメータの設定によって異なる)。

入力レコードが DECOMPRESS 機能を使用して作成された場合、入力レコードには必要なすべてのカウントフィールドがすでに含まれています。この場合、フィールド定義ステートメントでカウントを指定する必要はありません。

入力または更新時に含まれるすべての値が圧縮されます (**FI オプション**を一緒に指定している場合を除く)。FI オプションと MU オプションを一緒に使用する場合は、注意が必要です。これは、圧縮可能な値が多数存在すると、大量のディスクストレージが浪費される可能性があるためです。

NU オプションと MU オプションが一緒に指定されている場合は、空値が論理的および物理的の両形式で圧縮されます。すべての値（空値を含む）の位置的關係は、通常、MU オカレンスで維持されます（オカレンスが NU オプションで定義されている場合を除く）。多数の空値が MU フィールドグループに存在する場合、NU オプションを指定することによって、フィールドで必要となるディスクストレージを軽減できます。ただし、値の相対位置を維持する必要がある場合は、このオプションを使用しないでください。

NC (または NC/NN) オプションは、MU フィールドに対しては指定できません。

圧縮レコード内の 191 より多い MU および PE オカレンスを識別する方法については、「[圧縮レコード内の 191 を超える MU および PE オカレンスの特定](#)」を参照してください。

**MU** の使用例 (NU 指定あり)

```
FNDEF='01,AA,5,A,MU,NU'
```

元の内容 (L は値の長さ) :

- ファイルのロード後：

3	L 値 A	L 値 B	L 値 C
count	AA1	AA2	AA3

- 値 B を空値に更新後：

2	L 値 A	L 値 C
count	AA1	AA2

### MU の使用例 (NU 指定なし)

```
FNDEF=' 01,AA,5,A,MU'
```

元の内容 (L は値の長さ)：

- ファイルのロード後：

3	L 値 A	L 値 B	L 値 C
count	AA1	AA2	AA3

- 値 B を空値に更新後：

3	L 値 A	L 値 B	L 値 C
count	AA1	AA2	AA3

### NB：空白圧縮

NB オプションを **LA** および **LB** フィールドとともに使用すると、空白圧縮を制御できます。NB オプションを指定すると、フィールドの末尾の空白が削除されなくなります。指定しない場合は、英数字またはワイド文字フィールドの値の格納時に末尾の空白が削除されます。フィールドに NB オプションを指定した場合、そのフィールドに NU オプションまたは NC オプションも指定する必要があります。NB 処理には、NC または NU の使用も同時に必要となります。



**Note:** NB オプションを使用せずにフィールドを指定すると、格納したときと取り出したときのフィールドの長さが異なる可能性があります。取り出したときの非 NB フィールドの長さは、空白圧縮のために、格納したときにそのフィールドに指定した長さよりも短くなる可能性が高くなります。これは、値が実際は文字列ではなく、空白を表す文字コードで終わるバイナリ値であった場合に問題となることがあります。したがって、格納したときと取り出したときのフィールドの長さを同じにしたい場合は、NB オプションを使用します。

**NU：空値省略**

NUでは、フィールド内の空値を省略します。

標準圧縮（NUまたはFIを指定していない場合）では、空値を2バイトで表します（値の長さが1バイト、値自体（この場合、空値）が1バイト）。空値省略では、空値フィールドを1バイトの空値フィールドインジケータで表します。空値自体は格納されません。

空値を含む一連の連続フィールドにNUオプションを指定した場合、フィールドは1バイトの空値フィールド（バイナリ11nnnnnn）インジケータで表されます。ここで、nnnnnnは、合計63までの空値を含むフィールドの連続バイト数です。このため、NUオプションで定義されたフィールドはできるだけグルーピングする必要があります。

ディスクリプタにNUオプションを指定した場合、ディスクリプタの空値はインバーテッドリストに格納されません。したがって、検索コマンドでこのディスクリプタを使用した場合、および空値を検索値として使用した場合、常にレコードは選択されません。ディスクリプタの空値を含むデータストレージにレコードが存在する場合でも同様です。NUオプションで定義されたディスクリプタを使用して、論理順読み込み（L3/L6）コマンドで論理順を制御する場合、ディスクリプタの空値を含むレコードは読み込まれません。

ディスクリプタをファイルカップリングの基準として使用し、そのディスクリプタに多数の空値が存在する場合、NUオプションを指定して、カップリングリストの合計サイズを軽減する必要があります。

NUオプションは、NC/NN組み合わせオプションまたはFIオプションで定義されたフィールドに対しては指定できません。

**NUの使用例**

	定義	ユーザーデータ	内部表現
標準圧縮	FNDEF='01,AA,2,B'	0000	0200 (2 バイト)
FI オプションあり	FNDEF='01,AA,2,B,FI'	0000	0000 (2 バイト)
NU オプションあり	FNDEF='01,AA,2,B,NU'	0000	C1 (1 バイト) *

\* C1 は1つの空値フィールドを示します。

### NV：変換なし

英数字（A）フォーマットまたはワイド文字フォーマット（W）のフィールド用の無変換オプションです。このオプションを指定すると、該当フィールドがレコードバッファ内で変換なしで処理されます。

NVオプションで定義されたフィールドは、ユーザーとの間で変換されません。フィールドではファイルエンコードの特徴（つまり、デフォルトの空白）が保持されます。

- A フィールドの場合は、常に、EBCDIC の空白（X'40'）になります。
- W フィールドの場合は、常に、W フォーマットのファイルエンコードの空白になります。

NVオプションは、変換しても意味のないデータ、またはアプリケーションで保管されているままのデータが必要とされるために変換すべきでないデータを含むフィールドに使用します。

NVフィールドのフィールド長は、ユーザーアーキテクチャでバイトスワップが採用されている場合、バイトスワップされます。

NVフィールドについては、A フォーマットを W フォーマットに変換したり、W フォーマットを A フォーマットに変換したりすることはできません。

### UQ：ユニークディスクリプタ

UQ では、フィールドがユニークディスクリプタであることを示します。ユニークディスクリプタには、ファイル内の各レコードで異なる値が含まれている必要があります。FNDEF ステートメントでは、DE オプションを指定している場合にのみ UQ オプションを指定できます。UQ オプションは、SUBDE、SUPDE、および HYPDE ステートメントでも使用できます。

フィールドを ADAM ディスクリプタとして使用する場合、UQ オプションを指定する必要があります（[ADAMER ユーティリティ](#)を参照）。

ユニーク値のチェックは、ADACMP では行われません。このチェックを行うには、ADALOD ユーティリティまたは ADAINV ユーティリティで INVERT 機能を実行します。ファイルのロード中にユニークでない値が検出されると、ADALOD はエラーメッセージを出力して終了します。

ADAINV と ADALOD は拡張ファイルチェーン内のファイルごとに別々に実行する必要があるため、チェーン全体でユニークであるかどうかをチェックすることはできません。

ただし、Adabas では、拡張ファイルチェーン間でユニークディスクリプタの値をチェックできます。追加（N1/N2）または更新（A1）対象の値がチェーン内のすべてのファイル間でユニークでない場合、レスポンスコード 198 が返されます。

## XI：除外インスタンス番号

デフォルトでは、ユニークディスクリプタ（UQ）として定義されたピリオディックグループ（PE）内のフィールドのオカレンス番号がディスクリプタ値の一部として含まれます。つまり、異なるレコード内の別々のピリオディックグループオカレンスに同じフィールド値が含まれることもあります。

XI オプションを使用すると、値をユニークに保つために、ディスクリプタ値からオカレンス番号を除外できます。XI オプションを設定した場合、すべてのレコード内の PE フィールドの全オカレンスで、フィールド値が1つまでに限定されます。

## SQL 空値の表現

Adabas には、NC および NN の2つのデータ定義オプションがあります。これらのオプションを使用すると、Software AG のメインフレーム Adabas SQL Gateway（ACE）および他の SQL（Structured Query Language）データベース問い合わせ言語に対応する SQL 互換の空値表現を指定することができます。

NC および NN オプションは、以下で定義されたフィールドには適用できません。

- Adabas 空値省略（NU）
- 固定小数点データタイプ（FI）
- マルチプルバリュー（MU）
- ピリオディックグループ内（PE）
- グループフィールドとして

また、NN オプションは、NC オプションを指定したフィールドに対してのみ指定できます。

サブフィールド／スーパーフィールドまたはサブディスクリプタ／スーパーディスクリプタの親フィールドでは、NC オプションを指定できます。ただし、単一のスーパーフィールドまたはディスクリプタの親フィールドでは、NU および NC フィールドを組み合わせることはできません。1つの親フィールドが NC の場合、他の親フィールドを NU フィールドに指定することはできません。また、その逆についても同様です。

## 例

フィールド AA を定義し、NC および NN オプションを割り当てるための正しい ADACMP COMPRESS FNDEF ステートメントを以下に示します。

```
ADACMP FNDEF='01,AA,4,A,NN,NC,DE'
```

NC/NN オプションを不適切な方法で使用すると、ADACMP ユーティリティで ERROR-127 が発生します（次の例を参照）。

不適切な例	理由
ADACMP FNDEF='01,AA,4,A,NC,NU'	NU オプションと NC オプションは互換性がありません。
ADACMP FNDEF='01,AB,4,A,NC,FI'	NC オプションとおよび FI オプションは互換性がありません。
ADACMP FNDEF='01,PG,PE' ADACMP FNDEF='02,P1,4,A,NC'	PE グループ内で NC オプションを指定することはできません。

このsectionでは、次のトピックについて説明します。

- [NC：SQL 空値オプション](#)
- [空値インジケータ値](#)
- [NN：SQL 非空値オプション](#)

### NC：SQL 空値オプション

NC（非カウント）オプションを使用しない場合、空値は、フィールドのフォーマットに応じて、ゼロまたは空白になります。

NCオプションを使用した場合、レコードバッファに指定されたゼロまたは空白は、[空値インジケータ値](#)に従って、真のゼロまたは空白として（つまり、有意性のある空値として）解釈されるか、または未定義値として（つまり、真のSQLまたは有意性のない空値として）解釈されます。

NCオプションで定義されたフィールドに、レコードバッファに指定された値が含まれていない場合、フィールド値は常に SQL 空値として扱われます。

真のSQL空値として解釈された場合、その空値はSQL解釈で値のないフィールドを示します。つまり、フィールド値は格納されていない（定義されていない）ことを示します。

空値インジケータ値には、空値の内部 Adabas 表現を示す役割があります。詳細は、次のsectionの「[空値インジケータ値](#)」、および「[サーチバッファ](#)」を参照してください。

NC フィールドを含むレコードを圧縮または圧縮解除する際には次の規則が適用されます。

1. FORMAT パラメータを指定している場合、ADACMP は、更新タイプのコマンドに対するニュークリアスの動作と同じように動作します。『[Adabas コマンドリファレンス](#)』を参照してください。

2. FORMAT パラメータを指定していない場合

- **圧縮の場合：**

NC フィールドの値のみが入力レコードに含まれます。空値インジケータバイト（2 バイト）は省略する必要があります。値は、空値インジケータバイトがゼロに設定されている場合と同じように圧縮されます。この方法で NC フィールドに空値を割り当てることはできません。

例

フィールド定義テーブル (FDT) の定義	FNDEF=' 01,AA,4,A,NC'
入力レコードの内容:	MIKE

■ 圧縮解除の場合:

NC フィールドの値に有意性がない場合、レコードは DDFEHL (または FEHL) に書き込まれ、レスポンスコード 55 が返されます。

NC フィールドの値に有意性がある場合、値は通常どおり圧縮解除されます。空値インジケータバイトはありません。

例

フィールド定義テーブル (FDT) の定義	FNDEF=' 01,AA,4,A,NC'
出力レコードの内容	MIKE

空値インジケータ値

空値インジケータ値は、データフォーマットにかかわらず、常に 2 バイトであり、固定小数点フォーマットです。フィールド値が追加または変更されたときにレコードバッファに指定され、フィールド値が読み込まれたときにレコードバッファに返されます。

更新 (Ax) コマンドまたは追加 (Nx) コマンドでは、フォーマットバッファのフィールド指定に対応するレコードバッファ位置に空値インジケータ値を設定する必要があります。設定は次のいずれかにする必要があります。

16 進数値	意味
X'FFFF'	フィールド値を "未定義"、つまり有意性のない空値に設定します。レコードバッファ内のフィールドに値が指定されていないこと、バイナリの 0 が指定されていること、および空白が指定されていることの違いは、無視されます。すべてが "値が指定されていない" として解釈されます。
0000	レコードバッファ内のフィールドに値が指定されていない場合、バイナリの 0 が指定されている場合、および空白が指定されている場合は、有意性のある空値が指定されているものとして解釈されます。

## フィールド定義ステートメント

読み込み (Lx) コマンド、または読み込みを伴う検索 (フォーマットバッファエントリを含む Sx) コマンドでは、フォーマットバッファのフィールドの位置に対応するレコードバッファ位置に空値インジケータ値が返されたかどうかをプログラムで確認する必要があります。空値インジケータ値は、次の値のいずれかになります。これらの値はそれぞれ、選択したフィールドに含まれる実際の値の意味を示します。

16進数値	意味
X'FFFF'	フィールド内のゼロまたは空白には有意性がありません。
0000	フィールド内のゼロまたは空白は有意性のある値 (つまり、真のゼロまたは空白) です。
xxxx	フィールドは切り捨てられます。空値インジケータ値には、データベースレコードに格納されたとおりの値の全長 (xxxx) が含まれます。

例

NC オプションで定義された 2 バイトの Adabas バイナリフィールド AA での空値のフィールド定義を以下に示します。

01,AA,2,B,NC

値のタイプ	空値インジケータ値 (レコードバッファ)	データ	Adabas の内部表現
ゼロ以外の値	0 (バイナリの値に有意性あり)	0005	0205
空白	0 (バイナリの空値に有意性あり)	0000 (ゼロ)	0200
空値	FFFF (バイナリの空値に有意性なし)	(無関係)	C1

### NN : SQL 非空値オプション

NN (非空値または空値指定不可) オプションは、データフィールドに NC オプションも指定している場合にのみ指定できます。NN オプションでは、NC フィールドには必ず値 (ゼロまたは空白を含む) が定義されている必要があることを示します。フィールドには、"値を指定する" 必要があります。

NN オプションを使用すると、レコードが追加または更新されたときにフィールドが未定義のまま残らなくなります。フィールドには、有意性のある値が必ず設定されている必要があります。そうでない場合、Adabas からレスポンスコード 52 が返されます。

次の例は、NN オプションを指定した場合と指定しない場合に、2 バイトの Adabas 英数字フィールド AA で有意性のない空値がどのように処理されるかを示しています。

例

NN オプションを指定した場合、または指定しない場合に、2 バイトの Adabas 英数字フィールド AA では、有意性のない空値は次のように処理されます。

オプション	フィールド定義	空値インジケータ値	Adabas の内部表現
NN 指定あり	01,AA,2,A,NC,NN	FFFF (有意性のない空値)	なし (レスポンスコード 52 が返される)
NN 指定なし	01,AA,2,A,NC	FFFF (有意性のない空値)	C1

## FNDEF：ピリオディックグループ定義

ピリオディックグループ定義のエントリを構成するには、次の構文を使用します。

```
FNDEF = 'level, name [, PE [(occurrences) ] ]'
```

レベル番号と名前は必須です。定義エントリの間には空白をいくつ入れても構いません。

これらの定義の各 FNDEF パラメータについては、この section で説明されています。

- level
- name
- PE：ピリオディックグループ
- occurrences

### level

レベル番号は 01～07 の範囲内の 1 桁または 2 桁の数字（先行のゼロは任意）を使用し、フィールドグルーピングと合わせて使用します。02 またはそれ以上のレベル番号のついたフィールドは、それより小さいレベル番号のついた直前のグループの一部であるとみなされます。

グループで定義すると、グループ名を使用して一連のフィールド（または1つのフィールドだけでもよい）を参照することができます。これは一連の連続フィールドを参照するのに便利で効果的な方法です。

レベル番号 01～06 は、グループの定義に使用することができます。グループは他のグループを含むことができます。ネスト構造のグループにレベル番号をつけるときは、レベル番号を飛ばしてはなりません。

## フィールド定義ステートメント

次の例では、フィールド A1 および A2 はグループ GA に含まれます。フィールド B1 とグループ GC（フィールド C1 と C2 で構成）はグループ GB に含まれます。

FNDEF='01,GA'	グループ	
FNDEF='02,A1,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='02,A2,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='01,GB'	グループ	
FNDEF='02,B1,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='02,GC'	グループ（ネスト構造）	
FNDEF='03,C1,...'		エレメンタリまたはマルチプルバリュースフィールド
FNDEF='03,C2,...'		エレメンタリまたはマルチプルバリュースフィールド

### name

フィールド（またはグループ）に割り当てる名前を指定します。

名前はファイル内でユニークでなければなりません。名前は 2 文字で指定しなければならず、最初の文字は英字で、2 番目の文字は英字か数字でなければなりません。特殊文字は使用できません。

値 E0～E9 は編集マスク用に予約されており、使用してはなりません。

正しい名前例	誤りの名前例
AA	A（2 文字ではない）
B4	E3（編集マスク）
S3	F*（特殊文字が含まれている）
WM	6M（最初の文字が英字ではない）

### PE：ピリオディックグループ

ピリオディックグループフィールドオプションでは、グループフィールドの後に、特定レコードで複数回使用できるピリオディックグループ定義が続くことを示します。 **MUPEX** パラメータを指定し、**MUPECOUNT** パラメータを "2" に設定した場合、ピリオディックグループは 1 レコードあたり最大で 65,534 回使用できます。これらのパラメータを設定しない場合、ピリオディックグループは 1 レコードあたり最大で 191 回使用できます。各 ADACMP 入力レコードには、1 つ以上のオカレンス（オカレンス内の値がすべて空値の場合でも可）が存在する必要があります。

ピリオディックグループ：

- 1 つ以上のフィールドで構成されます。最大で 254 のエレメンタリフィールドを指定できます。ディスクリプタまたはマルチプルバリュースフィールド（あるいはその両方）、および他

のグループを指定できますが、ピリオディックグループに別のピリオディックグループを含めることはできません。

- **MUPEX** パラメータを指定し、**MUPECOUNT** パラメータを "2" に設定した場合、ピリオディックグループは特定グループ内で 0~65,534 回使用できます。ただし、各 ADACMP 入力レコードには、1つ以上のオカレンス（オカレンス内の値がすべて空値の場合でも可）が存在する必要があります。これらのパラメータを設定しない場合、ピリオディックグループは特定グループ内で 0~191 回使用できます。
- 01 レベルで定義する必要があります。ピリオディックグループ内のすべてのフィールドが直後に続き、さらにレベル 02 以上で定義されている必要があります（最大で 7 まで 1 ずつ増加）。次の 01 レベル定義は、現在のピリオディックグループの終了を示します。
- グループ名とともにのみ指定できます。長さパラメータとフォーマットパラメータをグループ名とともに指定することはできません。

圧縮レコード内の 191 より多い MU および PE オカレンスを識別する方法については、「[圧縮レコード内の 191 を超える MU および PE オカレンスの特定](#)」を参照してください。

ピリオディックグループ定義の 2 つの例を以下に示します。

ピリオディックグループ **GA** :

```
FNDEF=' 01 ,GA,PE'
FNDEF=' 02 ,A1,6,A,NU'
FNDEF=' 02 ,A2,2,B,NU'
FNDEF=' 02 ,A3,4,P,NU'
```

この例では、ピリオディックグループ GA はフィールド A1、A2、および A3 で構成されます。レコード内のピリオディックグループのオカレンス数は、1 バイトまたは 2 バイトのバイナリ値として定義されます（**MUPECOUNT** パラメータの設定によって異なる）。すべてのレコードで、各オカレンスグループの前に指定します。

ピリオディックグループ **GB** :

```
FNDEF=' 01 ,GB,PE(3)'
FNDEF=' 02 ,B1,4,A,DE,NU'
FNDEF=' 02 ,B2,5,A,MU(2),NU'
FNDEF=' 02 ,B3'
FNDEF=' 03 ,B4,20,A,NU'
FNDEF=' 03 ,B5,7,U,NU'
```

この例では、ピリオディックグループ GB はフィールド B1、B2、およびグループ B3（フィールド B4 および B5 を含む）で構成されます。レコードには、ピリオディックグループの 3 つのオカレンスを含めることができます。

### **occurrences**

MUPEX オプションを指定している場合に、レコード内に含める PE フィールドのオカレンスの数を指定します。これは任意のパラメータです。

## **COLDE：照合ディスクリプタ定義**

---

照合ディスクリプタオプションを使用すると、ユーザー指定のアルゴリズムに基づいてディスクリプタ値をソート（照合）できます。

値は、特別な照合ディスクリプタユーザー出口（CDX01～CDX08）でコーディングされたアルゴリズムに基づきます。各照合ディスクリプタはユーザー出口に割り当てられなければならないが、単一のユーザー出口は複数の照合ディスクリプタを処理します。

例



照合出口機能は次のイベントで呼び出されます。

#### INITIALIZE 機能

- ニュークリアスセッションの開始
- 照合出口が定義されたときのユーティリティの初期化 (ADARUN パラメータ)

#### ENCODE 機能

- 親の値の更新/挿入/削除 (ニュークリアス)
- 照合ディスクリプタを検索値に指定して検索 (ニュークリアス)
- レコードの圧縮 (ADACMP)

#### DECODE 機能

- 出口が DECODE 機能をサポートする場合のみ、照合 DE によるインデックス (L9) の読み込み (ニュークリアス)

ユーザー出口に対して指定する入力パラメータについては、「照合ディスクリプタ出口01~08」を参照してください。入力パラメータには次のものがあります。

- 入力文字列のアドレスと長さ
- 出力エリアのアドレスとサイズ
- 返される出力文字列長のフルワードのアドレス

ユーザー出口によって、返される出力文字列の長さが設定されます。

詳細は、「CDXnn：照合ディスクリプタユーザー出口」を参照してください。



#### Notes:

1. 照合ディスクリプタは、英数字 (A) またはワイド文字 (W) の親フィールドに対して定義できます。フォーマット、長さ、およびオプション (UQ および XI を除く) は、COLDE パラメータで定義された親フィールドから取得されます。ユニークディスクリプタ (UQ) およびインデックス除外 (XI) オプションは、照合ディスクリプタ自体に対して別々に定義されません。

## フィールド定義ステートメント

2. 照合ディスクリプタ値を使用した検索は、標準ディスクリプタの場合と同様の方法で実行されます。
3. 照合ディスクリプタ値が正しく作成されているかどうかは、ユーザーが確認する必要があります。データストレージに対する照合ディスクリプタ値の完全性をチェックする標準的な方法はありません。メンテナンスユーティリティ ADAICK は、インデックスの構造だけをチェックし、内容はチェックしません。ユーザーは、それぞれの値定義規則を設定し、値が正しいかどうかをチェックします。
4. ファイルに複数の照合ディスクリプタが含まれている場合、割り当てられている出口は、照合ディスクリプタ名のアルファベット順で呼び出されます。

### 照合ディスクリプタの構文

照合ディスクリプタは、次の構文を使用して定義します。

```
COLDE = 'number, name [ , UQ [ , XI ] ] = parent-field'
```

上記の意味は次に示すとおりです。

<i>number</i>	照合ディスクリプタに割り当てるユーザー出口番号です。Adabasニュークリアスでは、この番号を使用して、呼び出す照合ディスクリプタユーザー出口を決定します。
<i>name</i>	照合ディスクリプタに使用する名前です。照合ディスクリプタの命名規則は、Adabasフィールド名の命名規則と同一です。
<i>UQ</i>	ユニークディスクリプタオプションを照合ディスクリプタに割り当てることを示します。
<i>XI</i>	インデックス（オカレンス）番号を除外することによって、照合ディスクリプタをユニークに保つことを示します。
<i>parent-field</i>	エレメンタリ A または W フィールドの名前です。照合ディスクリプタには、1つの親フィールドを含めることができます。フィールド名とアドレスはユーザー出口に渡されます。

MU、NU、およびPEオプションは親フィールドから取得され、照合ディスクリプタに暗黙的に設定されます。

親フィールドをNUオプションで指定した場合、空値フィールドを含むレコードについては照合ディスクリプタのインバーテッドリストにエントリが作成されません。これは、他の照合ディスクリプタ要素の値の有無にかかわらず同様です。

親フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

照合ディスクリプタ定義の例：

フィールド定義：

```
FNDEF='01, LN, 20, A, DE, NU'      Last-Name
```

照合ディスクリプタ定義：

```
COLDE='1, Y2=LN'
```

- 照合ディスクリプタユーザー出口 1 (CDX01) がこの照合ディスクリプタに割り当てられます。名前は Y2 です。
- 照合ディスクリプタの長さフォーマットは親フィールドから取得されます。長さフォーマットはそれぞれ 20 と英数字です。照合ディスクリプタは、空値省略 (NU) のマルチプルバリュー (MU) フィールドです。
- 照合ディスクリプタの値は、親フィールド LN から派生します。

## HYPDE：ハイパーディスクリプタ定義

ハイパーディスクリプタオプションを使用すると、ユーザー指定のアルゴリズムに基づいてディスクリプタ値を生成できます。

値は、特別なハイパーディスクリプタユーザー出口 (HEX01~HEX31) でコーディングされたアルゴリズムに基づきます。各ハイパーディスクリプタはユーザー出口に割り当てられる必要があり、単一のユーザー出口で複数のハイパーディスクリプタが処理されます。

例

```
ハイパーディスクリプタHI,File=20 }
ハイパーディスクリプタHI,File=30 } → ハイパーディスクリプタUserExit4 (HEX4)
ハイパーディスクリプタY7,File=30 }
ハイパーディスクリプタFA,File=9   → ハイパーディスクリプタUserExit9 (HEX9)
```

ハイパーディスクリプタ値がAdabasニュークリアスまたはADACMPユーティリティによって処理されるとき、常に出口が呼び出されます。

ユーザー出口に対して指定する入力パラメータを以下に示します。

- ハイパーディスクリプタ名
- ファイル番号
- データストレージレコードからフィールド名とPEインデックス（該当する場合）とともに取得されるフィールドのアドレス これらのアドレスは圧縮後のフィールド値を指します。これらのフィールドの名前は、ADACMPまたはADAINVのHYPDEパラメータを使用して定義する必要があります。

ユーザー出口からはディスクリプタ値（DVT）が圧縮形式で返される必要があります。ハイパーディスクリプタに割り当てられているオプション（PE、MU）に応じて、値が返されないか、1つ以上の値が返されます。

入力値に割り当てられている元のISNは変更可能です。

ハイパーディスクリプタユーザー出口の詳細は、「ハイパーディスクリプタ出口01-31」を参照してください。



### Notes:

1. ハイパーディスクリプタのフォーマット、長さ、およびオプションについては、ユーザーが定義します。HYPDEパラメータで定義された親フィールドからは取得されません。
2. ハイパーディスクリプタ値を使用した検索は、標準ディスクリプタの場合と同様の方法で実行されます。
3. ハイパーディスクリプタ値が正しく作成されているかどうかは、ユーザーが確認する必要があります。データストレージに対するハイパーディスクリプタ値の完全性をチェックする標準的な方法はありません。メンテナンスユーティリティADAICKは、インデックスの構造だけをチェックし、内容はチェックしません。ユーザーは、それぞれの値定義規則を設定し、値が正しいかどうかをチェックします。
4. ハイパーディスクリプタがパック形式またはアンパック形式で定義されている場合、Adabasで返された値の正当性がチェックされます。パック値の符号ハーフバイトには、A、C、E、F（正）またはB、D（負）を格納できます。符号はAdabasでFまたはDに変換されます。
5. ファイルに複数のハイパーディスクリプタが含まれている場合、割り当てられている出口は、ハイパーディスクリプタ名のアルファベット順で呼び出されます。

ハイパーディスクリプタの構文

ハイパーディスクリプタは、次の構文を使用して定義します。

**HYPDE = 'number, name, length, format [ { , option } ... ] = { parent-field }, ...'**

ここでは次の内容を表しています。

<i>number</i>	ハイパーディスクリプタに割り当てるユーザー出口番号です。Adabas ニュークリアスでは、この番号を使用して、呼び出すハイパーディスクリプタユーザー出口を決定します。	
<i>name</i>	ハイパーディスクリプタに使用する名前です。ハイパーディスクリプタの命名規則は、Adabas フィールド名の命名規則と同一です。	
<i>length</i>	ハイパーディスクリプタのデフォルトの長さです。	
<i>format</i>	ハイパーディスクリプタのフォーマットです。	
	フォーマット	最大長
	英数字 (A)	253 バイト
	2 進数 (B)	126 バイト
	固定小数点 (F)	4 バイト (常に 4 バイト)
	浮動小数点 (G)	8 バイト (常に 4 または 8 バイト)
	パック 10 進数 (P)	15 バイト
	アンパック 10 進数 (U)	29 バイト
<b>注意:</b> ハイパーディスクリプタにはワイド文字 (W) フォーマットは使用できません。		
<i>option</i>	ハイパーディスクリプタに割り当てるオプションです。ハイパーディスクリプタとともに使用できるオプションを以下に示します。	
	MU	マルチプルバリューフィールド
	NU	空値省略
	PE	ピリオディックグループのフィールド
	UQ	ユニークディスクリプタ
<i>parent-field</i>	エレメンタリフィールドの名前です。ハイパーディスクリプタには、1~20 の親フィールドを含めることができます。フィールド名とアドレスはユーザー出口に渡されます。 <b>注意:</b> ハイパーディスクリプタの親フィールドは、W (ワイド文字) フォーマットにはできません。	

親フィールドを NU オプションで指定した場合、空値フィールドを含むレコードについてはハイパーディスクリプタのインバーテッドリストにエントリが作成されません。このことは、他のハイパーディスクリプタエレメントの値の有無にかかわらず同様です。

## フィールド定義ステートメント

親フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

ハイパーディスクリプタ定義の例：

フィールド定義：

```
FNDEF='01, LN, 20, A, DE, NU'   Last-Name
FNDEF='01, FN, 20, A, MU, NU'   First-Name
FNDEF='01, ID, 4, B, NU'       Identification
FNDEF='01, AG, 3, U'           Age
FNDEF='01, AD, PE'             Address
FNDEF='02, CI, 20, A, NU'       City
FNDEF='02, ST, 20, A, NU'       Street
FNDEF='01, FA, PE'             Relatives
FNDEF='02, NR, 20, A, NU'       R-Last-Name
FNDEF='02, FR, 20, A, MU, NU'   R-First-Name
```

ハイパーディスクリプタ定義：

```
HYPDE='2, HN, 60, A, MU, NU=LN, FN, FR'
```

- ハイパーディスクリプタユーザー出口2がこのハイパーディスクリプタに割り当てられます。名前は HN です。
- ハイパーディスクリプタは、空値省略 (NU) のマルチプルバリュー (MU) フィールドで、その長さは 60、フォーマットは英数字です。
- ハイパーディスクリプタの値は、フィールド LN、FN、および FR から派生します。

ADACMPHYPDE=ステートメントは、次の例に示すように、別の行に続けることができます。そのためには、まず、最初の行の引数全体の後、閉じ引用符の前にマイナス (-) を指定します。その後、次の行のステートメント名 (ADACMP) の直後に、残りの位置引数を引用符で囲んで入力します。

```
ADACMP HYPDE='1, HY, 20, A=AA, BB, CC, -'
ADACMP 'DD, EE, FF'
```

## PHONDE：フォネティックディスクリプタ

FINDコマンドでフォネティックディスクリプタを使用すると、類似したフォネティック値を含むすべてのレコードが返されます。ディスクリプタのフォネティック値は、フィールド値の先頭20バイトに基づきます。英字の値のみが考慮され、数値、特殊文字、および空白は無視されます。英数字の大文字と小文字は、内部的に同一のものとして処理されます。

フォネティックディスクリプタは、次の構文を使用して定義します。

```
PHONDE = 'name (field)'
```

ここでは次の内容を表しています。

<i>name</i>	フォネティックディスクリプタに使用する名前です。フォネティックディスクリプタの命名規則は、Adabas フィールド名の命名規則と同一です。
<i>field</i>	フォネティック処理されるフィールドの名前です。

フィールドは次の要件を満たす必要があります。

- エレメンタリフィールドまたはマルチプルバリューフィールドであること
- 英数字フォーマットで定義されていること

フィールドはディスクリプタとして指定できます。

フィールドは次のようには指定できません。

- サブディスクリプタ、スーパーディスクリプタ、またはハイパーディスクリプタ
- ピリオディックグループ内
- 複数のフォネティックディスクリプタのソースフィールドとして
- フォーマット W (ワイド文字)

フィールドを NU オプションで定義した場合、空値フィールド (指定のバイト位置内) を含むレコードについてはフォネティックディスクリプタのインバーテッドリストにエントリが作成されません。フォーマットはフィールドの場合と同一です。

フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバー

## フィールド定義ステートメント

テッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

フォネティックディスクリプタ定義の例：

フィールド定義：

```
FNDEF='01,AA,20,A,DE,NU'
```

フォネティック定義：

```
PHONDE='PA(AA)'
```

## SUBDE：サブディスクリプタ定義

サブディスクリプタは、エレメンタリフィールドの一部から作成されるディスクリプタです。そのエレメンタリフィールドは、ディスクリプタ自体であってもなくてもかまいません。サブディスクリプタはサブフィールドとしても使用できます。つまり、レコードの出力フォーマットを制御するために、フォーマットバッファに指定できます。

サブディスクリプタは、次の構文を使用して定義します。

```
SUBDE = 'name [, UQ [ XI ] ] = parent-field (begin, end)'
```

ここでは次の内容を表しています。

<i>name</i>	サブディスクリプタ名です。サブディスクリプタの命名規則は、Adabas フィールド名の命名規則と同一です。
<i>UQ</i>	サブディスクリプタをユニークとして定義することを示します（ <a href="#">オプション UQ の定義</a> を参照）。
<i>XI</i>	インデックス（オカレンス）番号を除外することによって、サブディスクリプタをユニークに保つことを示します（ <a href="#">オプション XI の定義</a> を参照）。
<i>parent-field</i>	サブディスクリプタの派生元フィールドの名前です。
<i>begin</i>	サブディスクリプタ定義を開始する親フィールド内の相対バイト位置です。
<i>end</i>	サブディスクリプタ定義を終了する親フィールド内の相対バイト位置です。

\* カウントは、英数字またはワイド文字フィールドの場合は左から右へと1から行われ、数値またはバイナリフィールドの場合は右から左へと1から行われます。親フィールドをPフォーマットで定義した場合、出力サブディスクリプタ値の符号は下位バイトの下位4ビット（つまり、バイト1）から取得されます。

サブディスクリプタの親フィールドは次のように指定できます。

- ディスクリプタ
- エレメンタリフィールド
- マルチプルバリューフィールド（ただし、マルチプルバリューフィールドの特定のオカレンスは不可）
- ピリオディックグループ内（ただし、ピリオディックグループの特定のオカレンス内は不可）

サブディスクリプタの親フィールドは次のようには指定できません。

- サブ／スーパーフィールド、サブディスクリプタ、スーパーディスクリプタ、またはフォネティックディスクリプタ
- フォーマット G（浮動小数点）

親フィールドを NU オプションで定義した場合、空値フィールド（指定のバイト位置内）を含むレコードについてはサブディスクリプタのインバーテッドリストにエントリが作成されません。フォーマットは親フィールドの場合と同一です。

親フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

サブディスクリプタ定義の例 1：

親フィールド定義：

```
FNDEF=' 01,AR,10,A,NU'
```

サブディスクリプタ定義：

```
SUBDE='SB=AR(1,5)'
```

サブディスクリプタ SB の値は、親フィールド AR のすべての値の先頭 5 バイト（左から右にカウント）から派生します。すべての値は文字フォーマットで示されます。

AR 値	SB 値
DAVENPORT	DAVEN
FORD	FORD
WILSON	WILSO

サブディスクリプタ定義の例 2：

親フィールド定義：

## フィールド定義ステートメント

---

```
FNDEF='02,PF,6,P'
```

サブディスクリプタ定義：

```
SUBDE='PS=PF(4,6)'
```

サブディスクリプタ PS の値は、親フィールド PF のすべての値の 4~6 バイト（右から左にカウント）から派生します。すべての値は 16 進数で示されます。

PF 値	PS 値
00243182655F	02431F
00000000186F	0F（注参照）
78426281448D	0784262D

 **Note:** 親フィールド PF に NU オプションが指定されている場合、この値に対応する PS 値は作成されません。

サブディスクリプタ定義の例 3：

ソースフィールド定義：

```
FNDEF='02,PF,6,P'
```

サブディスクリプタ定義：

```
SUBDE='PT=PF(1,3)'
```

PT の値は、PF のすべての値の 1~3 バイト（右から左にカウント）から派生します。すべての値は 16 進数で示されます。

PF 値	PT 値
00243182655F	82655F
00000000186F	186F
78426281448D	81448D

## SUBFN：サブフィールド定義

---

サブフィールドには、次のような特徴があります。

- Adabas 読み込みコマンドで読み込むことができるエレメンタリフィールドの一部です。
- 更新できません。
- ADAINV INVERT SUBDE=... を使用してサブディスクリプタに変更できます。

サブフィールドは、次の構文を使用して定義します。

```
SUBFN = ' name = parent-field (begin, end)'
```

ここでは次の内容を表しています。

<i>name</i>	サブフィールド名です。サブフィールドの命名規則は、Adabas フィールド名の命名規則と同一です。
<i>parent-field</i>	サブフィールドの派生元フィールドの名前です。
<i>begin*</i>	サブフィールド定義を開始する親フィールド内の相対バイト位置です。
<i>end*</i>	サブフィールド定義を終了する親フィールド内の相対バイト位置です。

\* カウントは、英数字またはワイド文字フィールドの場合は左から右へと1から行われ、数値またはバイナリフィールドの場合は右から左へと1から行われます。親フィールドをPフォーマットで定義した場合、出力サブフィールド値の符号は下位バイトの下位4ビット（つまり、バイト1）から取得されます。

サブフィールドの親フィールドは次のように指定できます。

- マルチプルバリューフィールド
- ピリオディックグループ内

サブフィールドの親フィールドはGフォーマット（浮動小数点）にはできません。

サブフィールド定義の例

```
SUBFN=' X1=AA(1,2)'
```

## SUPDE：スーパーディスクリプタ定義

スーパーディスクリプタは、複数のフィールド、フィールドの一部、またはこれらの組み合わせから作成されるディスクリプタです。

スーパーディスクリプタの定義に使用される各ソースフィールド（フィールドの一部）は、親と呼ばれます。スーパーディスクリプタの定義には、2~20の親フィールドまたはフィールドの一部を使用できます。合計サイズは、253以下である必要があります。

スーパーディスクリプタはユニークディスクリプタとして定義できます。

スーパーディスクリプタはスーパーフィールドとしても使用できます。つまり、レコードの出力フォーマットを決定するために、フォーマットバッファに指定できます。

このsectionでは、次のトピックについて説明します。

- SUPDE 構文
- スーパーディスクリプタに対する Adabas コマンドインターフェイス
- スーパーディスクリプタのフォーマット変換
- SUPDE の例

### SUPDE 構文

スーパーディスクリプタは、次の構文を使用して定義します。

```
SUPDE = ' name [ , UQ [ XI ] ] = { parent-field ( begin , end ) } , ...'
```

ここでは次の内容を表しています。

name	スーパーディスクリプタ名です。スーパーディスクリプタの命名規則は、Adabas の名前の命名規則と同一です。
UQ	スーパーディスクリプタをユニークとして定義することを示します（ <a href="#">オプション UQ の定義</a> を参照）。
XI	インデックス（オカレンス）番号を除外することによって、スーパーディスクリプタをユニークに保つことを示します（ <a href="#">オプション XI の定義</a> を参照）。
parent-field	スーパーディスクリプタ要素の派生元の親フィールドの名前です。最大で20の親フィールドを指定できます。
begin*	スーパーディスクリプタ要素を開始するフィールド内の相対バイト位置です。
end*	スーパーディスクリプタ要素を終了するフィールド内の相対バイト位置です。

\* カウントは、英数字またはワイド文字フォーマットで定義されたフィールドの場合は左から右へと1から行われ、数値またはバイナリフォーマットで定義されたフィールドの場合は右から左へと1から行われます。すべての親フィールド（FI で定義された親フィールドを除く）について、親フィールドのデータタイプで許可されている範囲内の開始値と終了値が有効です。

スーパーディスクリプタの親フィールドは次のように指定できます。

- エレメンタリフィールド
- 最大1つの MU フィールド（ただし、特定の MU フィールド値は不可）
- ピリオディックグループ内（ただし、特定のオカレンス内は不可）
- ディスクリプタ

スーパーディスクリプタの親フィールドは次のようには指定できません。

- スーパーディスクリプタ、サブディスクリプタ、またはフォネティックディスクリプタ

- フォーマット G (浮動小数点)
- NC オプションフィールド (別の親フィールドが NU オプションフィールドの場合)
- ロング英数字 (LA) フィールド

親フィールドを NC または NU オプションで指定した場合、空値フィールドを含むレコードについてはスーパーディスクリプタのインバーテッドリストにエントリが作成されません。つまり、親の値が空値で、かつ NC/NU オプションが指定されている場合、値は作成されません。このことは、他のスーパーディスクリプタエレメントの値の有無にかかわらず同様です。

親フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

スーパーディスクリプタ値の合計長は、253 バイト (英数字) または 126 バイト (2 進数) を超えない必要があります。

A (英数字) または W (ワイド文字) 親フィールドから派生したスーパーディスクリプタエレメントが存在しない場合、スーパーディスクリプタのフォーマットは B (2 進数) です。A または W 親フィールドから派生したスーパーディスクリプタエレメントが存在する場合、スーパーディスクリプタのフォーマットには、最後の A または W エレメントが反映されます。例えば、最後の A または W エレメントが W の場合、スーパーディスクリプタのフォーマットは W になります。

バイナリ形式のすべてのスーパーディスクリプタが符号なしの数として処理されます。

ADACMP SUPDE=ステートメントは、別の行に続けることができます。そのためには、最初の行の閉じ引用符の直前にある引数の後にマイナス (-) を指定します。その後、次の行のステートメント名 (ADACMP) の直後に、残りの位置引数を引用符で囲んで入力します。例としては、次のようなものがあります。

```
ADACMP SUPDE='SI=AA(10,20),BB(20,21),-'
ADACMP      'CC(12,13),DD(14,15)'
```

## スーパーディスクリプタに対する Adabas コマンドインターフェイス

スーパーディスクリプタに対するインターフェイスとして次のコマンドを使用できます。

Adabas コマンド	スーパーディスクリプタ値は次のいずれかです。
N1 または A1	親フィールドでスーパーディスクリプタの作成が強制される場合に、挿入処理および更新処理を行うと特定のフィールドを使用して暗黙的に構築されます。
Sx または L3	検索式および論理読み込みのバリュースタックに指定されます。
L9	レコードバッファに返されます。

## スーパーディスクリプタのフォーマット変換

スーパーディスクリプタには、次のように計算される最終のスーパーディスクリプタフォーマットが適用されます。

スーパーディスクリプタのフォーマット	使用される状況：
A (英字)	少なくとも 1 つの親フィールドがフォーマット A の場合
B (2 進数)	他のすべてがスーパーディスクリプタフィールドの場合

このsectionでは、次のトピックについて説明します。

- [更新時のフォーマット変換](#)
- [バリュースタックでのフォーマット変換](#)
- [出力のフォーマット変換 \(L9 コマンド\)](#)

 **Note:** 変換が実行されるのは、親フィールドが NV オプションで設定されている場合のみです。

### 更新時のフォーマット変換

スーパーディスクリプタは、すべての環境で同じ照合順序が適用されるように構築される必要があります。ただし、このsectionで説明されているとおり、一部の組み合わせでは問題が発生します。

- IBM (EBCDIC) および UNIX (ASCII) 環境での英数字 (フォーマット A) の値
- IBM (EBCDIC) および UNIX (ASCII) 環境での数値 (フォーマット U)
- ビッグエンディアンおよびリトルエンディアンストレージフォーマットのバイナリ値 (フォーマット B、F、G)

## ■ パック値の各種符号（フォーマット P）

### IBM（EBCDIC）および UNIX（ASCII）環境での英数字（フォーマット A）の値

UNIX データベースに対する挿入または更新コールが IBM メインフレーム環境からのものである場合、すべての英字フィールドの値が EBCDIC から ASCII に変換されます。その結果として、スーパーディスクリプタの親の値が自動的に ASCII に変換されます。この場合、アプリケーションで特定のソート順（例えば、大文字と小文字の使用）が要求されるとき、処理が失敗する可能性があります。EBCDIC フォーマットでは、小文字は大文字の前をきて、ASCII フォーマットでは、この順序が逆になります（大文字が小文字の前をきます）。

次の 2 つの方法のいずれかを使用して、この問題を解決できます。

- EBCDIC-ASCII 矛盾を持つ親フィールドに対して NV オプションを使用します。これにより、EBCDIC-ASCII 変換が無効になります。
- スーパーディスクリプタの代わりにハイパーディスクリプタを使用します。

### IBM（EBCDIC）および UNIX（ASCII）環境での数値（フォーマット U）

UNIX データベースに対する挿入または更新コールが IBM メインフレーム環境からのものである場合、すべての数字フィールドの値が EBCDIC から ASCII に変換されます。その結果として、スーパーディスクリプタの親の値が自動的に ASCII に変換されます。最終のスーパーディスクリプタによって A（英字）または B（2 進数）フォーマットが要求された場合でも同様です。

### ビッグエンディアンおよびリトルエンディアンストレージフォーマットのバイナリ値（フォーマット B、F、G）

プラットフォームによっては、バイナリバイト順序がビッグエンディアンフォーマットまたはリトルエンディアンフォーマットで格納されることがあります。例えば、IBM および HP-UX プロセッサでは、ビッグエンディアンフォーマット（バイトは右から左に評価）で使用され、Intel プロセッサではリトルエンディアン順（バイトは左から右に評価）が使用されます。

Adabas では、スワップアーキテクチャのバイナリ値（左から右に評価されるリトルエンディアンバイナリ値）を含むスーパーディスクリプタを変換し、それらをインデックスに格納する前に標準ソート順にします。

- 親の長さが 1 を超えるバイナリフィールドを 1 つ以上含む英字のスーパーディスクリプタの場合、バイナリの親の値がスワップされます。
- バイナリのスーパーディスクリプタの場合、親エントリの順序がスワップされ、バイナリ以外の親の値がスワップされます。

### パック値の各種符号（フォーマット P）

パック値の符号情報の表現は、プラットフォームによって異なります。Adabasがオープンシステム上にある場合は、正の値（A、C、またはF）がCに変換され、負の値（BまたはD）がDに変換されます。Adabasがメインフレーム上にある場合は、正の値はFで表されます。この結果、パック値がスーパーディスクリプタで使用されている場合は、パック値の符号から意味が失われる（標準のビットパターンになる）ため、照合順序の問題が発生します。この問題が発生した場合、正のパック値が負のパック値としてソートされる可能性があります。

さらに、パック値がスーパーディスクリプタに組み込まれている場合、メインフレーム向けのAdabasでは負のパック値が正のパック値の前にソートされ、オープンシステム向けのAdabasでは正のパック値が負のパック値の前にソートされます。

これらの問題を解決するために、スーパーディスクリプタの代わりにハイパーディスクリプタを使用することをお勧めします。

### バリュースワップでのフォーマット変換

スーパーディスクリプタがバリュースワップに指定されている場合、関連するインデックスエン트리と一致するように変換されます。

### 出力のフォーマット変換（L9 コマンド）

L9 コマンドで取得されたスーパーディスクリプタ値は、レコードバッファに返される前に変換される必要があります。英字フィールドは、必要に応じてASCIIからEBCDICに変換されます。また、必要に応じて、スーパーディスクリプタのバイナリ部分がスワップされます。スーパーディスクリプタのパック値部分のパック符号は変換されません。

### SUPDE の例

- [スーパーディスクリプタ定義の例 1](#)
- [スーパーディスクリプタ定義の例 2](#)
- [スーパーディスクリプタ定義の例 3](#)
- [スーパーディスクリプタ定義の例 4](#)

## ■ スーパーディスクリプタ定義の例 5

### スーパーディスクリプタ定義の例 1

フィールド定義：

FNDEF=' 01, LN, 20, A, DE, NU'	Last-Name
FNDEF=' 01, FN, 20, A, MU, NU'	First-Name
FNDEF=' 01, ID, 4, B, NU'	Identification
FNDEF=' 01, AG, 3, U'	Age
FNDEF=' 01, AD, PE'	Address
FNDEF=' 02, CI, 20, A, NU'	City
FNDEF=' 02, ST, 20, A, NU'	Street
FNDEF=' 01, FA, PE'	Relatives
FNDEF=' 02, NR, 20, A, NU'	R-Last-Name
FNDEF=' 02, FR, 20, A, MU, NU'	R-First-Name

スーパーディスクリプタ定義：

```
SUPDE=' SD=LN(1,4),ID(3,4),AG(2,3)'
```

スーパーディスクリプタ SD を作成します。スーパーディスクリプタの値は、フィールド LN のバイト 1~4（左から右にカウント）、フィールド ID のバイト 3~4（右から左にカウント）、およびフィールド AG のバイト 2~3（右から左にカウント）から派生します。すべての値は 16 進数で示されます。

LN	ID	AG	SD
C6D3C5D4C9D5C7	00862143	F0F4F3	C6D3C5D40086F0F4
D4D6D9D9C9E2	02461866	F0F3F8	D4D6D9D90246F0F3
D7C1D9D2C5D9	00000000	F0F3F6	ID により値は格納されない
404040404040	00432144	F0F0F0	LN により値は格納されない
C1C1C1C1C1	00000144	F1F1F1	C1C1C1C10000F1F1
C1C1C1C1C1	00860000	F0F0F0	C1C1C1C10086F0F0

英数字フォーマットで定義された親フィールドから 1 つ以上のエレメントが派生しているため、SD のフォーマットは英数字になります。

### スーパーディスクリプタ定義の例 2

フィールド定義：

```
FNDEF='01, LN, 20, A, DE, NU'   Last-Name
FNDEF='01, FN, 20, A, MU, NU'   First-Name
FNDEF='01, ID, 4, B, NU'       Identification
FNDEF='01, AG, 3, U'          Age
FNDEF='01, AD, PE'            Address
FNDEF='02, CI, 20, A, NU'      City
FNDEF='02, ST, 20, A, NU'      Street
FNDEF='01, FA, PE'            Relatives
FNDEF='02, NR, 20, A, NU'      R-Last-Name
FNDEF='02, FR, 20, A, MU, NU'  R-First-Name
```

スーパーディスクリプタ定義：

```
SUPDE='SY=LN(1,4),FN(1,1)'
```

スーパーディスクリプタ SY をフィールド LN および FN (マルチプルバリューフィールド) から作成します。すべての値は文字フォーマットで示されます。

LN	FN	SY
FLEMING	DAVID	FLEMD
MORRIS	RONALD RON	MORRR MORRR
WILSON	JOHN SONNY	WILSJ WILSS

英数字フォーマットで定義された親フィールドから1つ以上のエレメントが派生しているため、SY のフォーマットは英数字になります。

### スーパーディスクリプタ定義の例 3

フィールド定義：

```
FNDEF='01, PN, 6, U, NU'
FNDEF='01, NA, 20, A, DE, NU'
FNDEF='01, DP, 1, B, FI'
```

スーパーディスクリプタ定義：

```
SUPDE='SZ=PN(3,6),DP(1,1)'
```

スーパーディスクリプタ SZ を作成します。スーパーディスクリプタの値は、フィールド PN のバイト 3～6（右から左にカウント）、およびフィールド DP のバイト 1 から派生します。すべての値は 16 進数で示されます。

PN	DP	SZ
F0F2F4F6F7F2	04	F0F2F4F604
F8F4F0F3F9F8	00	F8F4F0F300
F0F0F0F0F1F1	06	F0F0F0F006
F0F0F0F0F0F1	00	F0F0F0F000
F0F0F0F0F0F0	00	PN により値は格納されない
F0F0F0F0F0F0	01	PN により値は格納されない

英数字フォーマットで定義された親フィールドから派生したエレメントはないため、SZ のフォーマットはバイナリになります。最後の 2 つの値については、空値は格納されません。スーパーディスクリプタオプションが NU（PN フィールドから）で、PN フィールド値にアンパック形式のゼロ（X'F0'）、つまり空値が格納されているためです。

#### スーパーディスクリプタ定義の例 4

フィールド定義：

```
FNDEF=' 01 , PF , 4 , P , NU '  
FNDEF=' 01 , PN , 2 , P , NU '
```

スーパーディスクリプタ定義：

```
SUPDE=' SP=PF ( 3 , 4 ) , PN ( 1 , 2 ) '
```

スーパーディスクリプタ SP を作成します。スーパーディスクリプタの値は、フィールド PF のバイト 3～4（右から左にカウント）、およびフィールド PN のバイト 1～2（右から左にカウント）から派生します。すべての値は 16 進数で示されます。

PF	PN	SP
0002463F	003F	0002003F
0000045F	043F	0000043F
0032464F	000F	PN により値は格納されない
0038000F	044F	0038044F

英数字フォーマットで定義された親フィールドから派生したエレメントはないため、SP のフォーマットはバイナリになります。

### スーパーディスクリプタ定義の例 5

フィールド定義：

```
FNDEF=' 01,AD,PE'  
FNDEF=' 02,CI,4,A,NU'  
FNDEF=' 02,ST,5,A,NU'
```

スーパーディスクリプタ定義：

```
SUPDE=' XY=CI(1,4),ST(1,5)'
```

スーパーディスクリプタ XY をフィールド CI および ST から作成します。すべての値は文字フォーマットで示されます。

CI	ST	XY
(第 1 オカレンス) BALT	(第 1 オカレンス) MAIN	BALTMAIN
(第 2 オカレンス) CHI	(第 2 オカレンス) SPRUCE	CHI SPRUC
(第 3 オカレンス) WASH	(第 3 オカレンス) 11TH	WASH11TH
(第 4 オカレンス) DENV	(第 4 オカレンス) bbbbb	ST により値は格納されない

英数字フォーマットで定義された親フィールドから1つ以上のエレメントが派生しているため、XY のフォーマットは英数字になります。

## SUPFN：スーパーフィールド定義

---

スーパーフィールドは、Adabas 読み込みコマンドで読み込むことができる複数のフィールド、フィールドの一部、またはこれらの組み合わせで構成されるフィールドです。スーパーフィールドに対して次の処理を行うことはできません。

- 更新する。
- NC オプションで定義されたフィールドで構成する（別の親フィールドが NU オプションで定義されている場合）。
- ディスクリプタとして使用する。

ADAINV ユーティリティの機能 INVERT SUPDE=... を使用すると、スーパーフィールドをスーパーディスクリプタに変更できません。

スーパーフィールドは、次の構文を使用して定義します。

```
SUPFN = 'name = parent-field (begin, end)[, parent-field (begin, end)]...'
```

ここでは次の内容を表しています。

<i>name</i>	スーパーフィールド名です。スーパーフィールドの命名規則は、Adabas の名前の命名規則と同一です。
<i>parent-field</i>	スーパーフィールド要素の派生元フィールドの名前です。
<i>begin</i> *	スーパーフィールド要素を開始するフィールド内の相対バイト位置です。
<i>end</i> *	スーパーフィールド要素を終了するフィールド内の相対バイト位置です。

\* カウントは、英数字またはワイド文字フォーマットで定義されたフィールドの場合は左から右へと1から行われ、数値またはバイナリフォーマットで定義されたフィールドの場合は右から左へと1から行われます。

スーパーフィールドの親フィールドは次のように指定できます。

- マルチプルバリューフィールド
- ピリオディックグループ内

スーパーフィールドの親フィールドは、フォーマット G（浮動小数点）にはできません。

スーパーフィールド値の合計長は、253 バイト（英数字）または 126 バイト（2 進数）を超えない必要があります。

A（英数字）または W（ワイド文字）親フィールドから派生したスーパーフィールド要素が存在しない場合、スーパーフィールドのフォーマットは B（2 進数）です。A または W 親フィールドから派生したスーパーフィールド要素が存在する場合、スーパーフィールドのフォーマットは、最後の A または W エlement に合わせられます。例えば、最後の A または W エlement が W の場合、スーパーフィールドのフォーマットは W になります。

スーパーフィールド定義の例

```
SUPFN=' X2=AA(1,2),AB(1,4),AC(1,1) '
```

---

# 21 JCL/JCS の必要項目と指定例

---

■ ユーザー出口を使用した ADACMP .....	152
■ BS2000 .....	153
■ z/OS .....	156
■ z/VM .....	160
■ VSE .....	162

このセクションでは、ADACMP を BS2000、z/OS、z/VM、および VSE システムで実行するために必要なジョブ制御情報について説明します。また、各ジョブストリームの例を示します。

 **Note:** リカバリログがアクティブな場合、実行ログが RLOG に記録されるユーティリティで使用されるシーケンシャルデータセットが、リカバリ処理用に維持され、かつ使用できる状態になっている必要があります（例えば、ADALOD LOAD 処理の DD/EBAND 入力）。

この chapter では、次のトピックについて説明します。

## ユーザー出口を使用した ADACMP

### ユーザー出口を使用した圧縮

レコードが ADACMP COMPRESS ユーティリティで処理される前に、ユーザー出口 6 を使用してそのレコードに対してユーザー処理を実行することができます。これは、ADACMP DECOMPRESS ユーティリティから出力された非圧縮レコードの順序および内容を制御するためにも使用できます。このように使用すると、ユーザー出口により、どの非圧縮レコードが ADACMP により DDAUSBA データセットに書き込まれるかが制御されます。ユーザー出口 6 の詳細は、「ユーザー出口 6 (データ圧縮前のユーザー処理)」を参照してください。

ADACMP 実行中にユーザー出口 6 を使用するには、実行時にロード可能なユーザー出口ルーチンが必要です。つまり、ユーザー出口ルーチンをアセンブルして、次の Adabas ライブラリにリンクしておく必要があります。

- BS2000、z/OS、z/VM 用のロードライブラリ（またはそれに連結されたライブラリ）
- VSE 用のコアイメージライブラリ、またはコアイメージライブラリ検索チェーン内のライブラリ

ADACMP COMPRESS ユーティリティジョブでは、以下を指定する必要があります。

```
ADARUN UEX6 = exit-name
```

上記の意味は次に示すとおりです。

<code>exit-name</code>	ユーザー出口で制御されるユーザールーチン名。最大 8 文字。
------------------------	--------------------------------

詳細は、「`UEXn : ユーザー出口`」を参照してください。

## ユーザー出口を使用した照合

ADACMP 実行中に照合ユーザー出口を使用するには、ADARUN ユーティリティの実行用に `CDXnn` パラメータを指定する必要があります。

UES (ユニバーサルエンコーディングサポート) と組み合わせて使用する場合、照合ディスクリプタユーザー出口パラメータのフォーマットは次のようになります。

**ADARUN CDX<sub>nn</sub>= exit-name**

上記の意味は次に示すとおりです。

<code>nn</code>	照合ディスクリプタ出口の番号。01 から 08 までの 2 桁の 10 進数。
<code>exit-name</code>	照合ディスクリプタ出口で制御を持つユーザールーチン名。最大 8 文字。

照合ディスクリプタ出口に指定できるプログラムは 1 つだけです。最大 8 個の照合ディスクリプタ出口を任意の順序で指定できます。詳細は、「`CDXnn : 照合ディスクリプタユーザー出口`」を参照してください。

## BS2000

データセット	リンク名	ストレージ	追加情報
ユーザー入力データ (COMPRESS 機能)	DDEBAND	テープ/ディスク	
圧縮データ (DECOMPRESS 機能)	DDEBAND	テープ/ディスク	INFILE パラメータを指定している場合は不要
ラージオブジェクト (LB) フィールドを含むファイルを有するデータベースの圧縮データ (COMPRESS 機能)	DDAUSB1	テープ/ディスク	この追加データセットは、圧縮されたラージオブジェクトレコードを受け取ります。このレコードは、最初の出力データセット (DDAUSBA) の圧縮レコードが基本ファイルにロードされる時

データセット	リンク名	ストレージ	追加情報
			に LOB ファイルにロードされます。
圧縮データ (COMPRESS 機能)	DDAUSBA	テープ/ディスク	
圧縮解除データ (DECOMPRESS 機能)	DDAUSBA	テープ/ディスク	
拒否データ	DDFEHL	テープ/ディスク	
ECS エンコードオブジェクト	DDECSOJ	テープ/ディスク	UES (ユニバーサルエンコーディングサポート) が必要です。
ADARUN パラメータ	DDCARD		Adabas オペレーション
ADACMP パラメータおよびデータ定義	SYSDTA/ DDKARTE		Adabas ユーティリティ
ADARUN メッセージ	SYSOUT/ DDPRINT	プリンタ/ディスク	Adabas メッセージおよびコード
ADACMP レポート	SYSLST/ DDDRUCK	プリンタ/ディスク	Adabas メッセージおよびコード

## JCL の例 (BS2000)

### ADACMP COMPRESS

SDF フォーマットの場合：

```

/.ADACMP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A C M P COMPRESS
/REMARK *
/DELETE-FILE CMP.AUS
/SET-JOB-STEP
/DELETE-FILE CMP.FEHL
/SET-JOB-STEP
/CREATE-FILE CMP.AUS,PUB(SPACE=(48,48))
/SET-JOB-STEP
/CREATE-FILE CMP.FEHL,PUB(SPACE=(48,48))
/SET-JOB-STEP

/ASS-SYSLST L.CMP
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDEBAND,CMP.EIN
/SET-FILE-LINK DDAUSBA,CMP.AUS
/SET-FILE-LINK DDFEHL,CMP.FEHL
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
    
```

```
ADARUN PROG=ADACMP,DB=yyyyy,IDTNAME=ADABAS5B
ADACMP COMPRESS NUMREC=1000,FDT=1,USERISN,DEVICE=dddd,eeee
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADACMP LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A C M P COMPRESS
/REMARK *
/ER CMP.AUS
/STEP
/ER CMP.FEHL
/STEP
/SYSFILE SYSLST=L.CMP
/FILE ADA.MOD,LINK=DDLIB
/FILE CMP.EIN,LINK=DDEBAND
/FILE CMP.AUS,LINK=DDAUSBA,SPACE=(48,48)
/FILE CMP.FEHL,LINK=DDFEHL,SPACE=(48,48)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADACMP,DB=yyyyy,IDTNAME=ADABAS5B
ADACMP COMPRESS NUMREC=1000,FDT=1,USERISN,DEVICE=dddd,eeee
/LOGOFF NOSPOOL
```

## ADACMP DECOMPRESS

### SDF フォーマットの場合：

```
/.ADACMP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A C M P DECOMPRESS
/REMARK *

/DELETE-FILE CMP.AUS
/SET-JOB-STEP
/DELETE-FILE CMP.FEHL
/SET-JOB-STEP
/CREATE-FILE CMP.AUS,PUB(SPACE=(48,48))
/SET-JOB-STEP
/CREATE-FILE CMP.FEHL,PUB(SPACE=(48,48))
/SET-JOB-STEP
/ASS-SYSLST L.DEC
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDEBAND,CMP.EIN
```

```

/SET-FILE-LINK DDAUSBA,CMP.AUS
/SET-FILE-LINK DDFEHL,CMP.FEHL
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADACMP,DB=yyyyy,IDTNAME=ADABAS5B
ADACMP DECOMPRESS
/LOGOFF SYS-OUTPUT=DEL
    
```

ISP フォーマットの場合：

```

/.ADACMP LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A C M P DECOMPRESS
/REMARK *
/ER CMP.AUS
/STEP
/ER CMP.FEHL
/STEP
/SYSFILE SYSLST=L.CMP.DEC

/FILE ADA.MOD,LINK=DDLIB
/FILE CMP.EIN,LINK=DDEBAND
/FILE CMP.AUS,LINK=DDAUSBA,SPACE=(48,48)
/FILE CMP.FEHL,LINK=DDFEHL,SPACE=(48,48)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADACMP,DB=yyyyy,IDTNAME=ADABAS5B
ADACMP DECOMPRESS
/LOGOFF NOSPOOL
    
```

z/OS

データセット	DD名	ストレージ	追加情報
ユーザー入力データ (COMPRESS 機能)	DDEBAND	テープ/ディスク	
圧縮データ (DECOMPRESS 機能)	DDEBAND	テープ/ディスク	INFILE パラメータを指定している場合は不要
ラージオブジェクト (LB) フィールドを含むファイルを有するデータベースの圧縮データ (COMPRESS 機能)	DDAUSB1	テープ/ディスク	この追加データセットは、圧縮されたラージオブジェクトレコードを受け取ります。このレコードは、最初の出力データセット (DDAUSBA) の圧縮レコードが基本ファイルにロードされる時

データセット	DD 名	ストレージ	追加情報
			に LOB ファイルにロードされます。
圧縮データ (COMPRESS 機能)	DDAUSBA	テープ/ディスク	
圧縮解除データ (DECOMPRESS 機能)	DDAUSBA	テープ/ディスク	
拒否データ	DDFEHL	テープ/ディスク	
ECS エンコードオブジェクト	DDEC SOJ	テープ/ディスク	UES (ユニバーサルエンコーディングサポート) に必要です。
ADACMP レポート	DDDRUCK	プリンタ	
ADARUN メッセージ	DDPRINT	プリンタ	
ADARUN パラメータ	DDCARD	リーダー	
ADACMP パラメータおよびデータ定義	DDKARTE	リーダー	

## JCL の例 (z/OS)

MVSJOBS データセットで、COMPRESS の例については ADACMP および ADACMPs、DECOMPRESS の例については ADACMPD を参照してください。

### ADACMP COMPRESS

この例は、MVSJOBS データセットのメンバ ADACMP 内にあります。

```
//ADACMP      JOB
//*
//*      ADACMP COMPRESS
//*      COMPRESS A FILE
//*
//CMP         EXEC PGM=ADARUN
//STEPLIB    DD  DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//*
//DDDRUCK    DD  SYSOUT=X
//DDPRINT    DD  SYSOUT=X
//SYSUDUMP   DD  SYSOUT=X
//DDEBAND    DD  DISP=OLD,DSN=EXAMPLE.DByyyyy.INPUT,UNIT=TAPE, <===
//            VOL=SER=TAPE01                        <===
//DDAUSBA    DD  DISP=(NEW,KEEP),DSN=EXAMPLE.DByyyyy.COMP01,UNIT=DISK, <==
//            VOL=SER=DISK01,SPACE=(TRK,(200,10),RLSE)
//DDFEHL     DD  DISP=(NEW,KEEP),DSN=EXAMPLE.DByyyyy.FEHL,UNIT=DISK, <===
//            VOL=SER=DISK01,SPACE=(TRK,1)
//DDCARD     DD  *
ADARUN PROG=ADACMP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
```

```

/*
//DDKARTE DD *
ADACMP COMPRESS FILE=1
ADACMP FNDEF='01,AA,008,B,DE'
ADACMP FNDEF='01,BA,020,A,NU,DE'
ADACMP FNDEF='01,BB,015,A,NU,DE'
ADACMP FNDEF='01,BC,001,A,FI'
ADACMP FNDEF='01,CA,001,A,NU,DE'
ADACMP FNDEF='01,CB,002,U,NU,DE'
ADACMP FNDEF='01,CC,010,A,NU,DE'
ADACMP FNDEF='01,CD,002,U,NU,DE'
ADACMP FNDEF='01,DA,005,U,NU'
ADACMP FNDEF='01,DB,020,A,NU,DE'
ADACMP FNDEF='01,DC,015,A,NU,DE'
ADACMP FNDEF='01,DD,002,A,NU,DE'
ADACMP FNDEF='01,DE,005,U,NU,DE'
ADACMP FNDEF='01,DF,008,A,NU,DE'
ADACMP FNDEF='01,FA,020,A,NU,DE'
ADACMP FNDEF='01,FB,006,U,NU,DE'
ADACMP FNDEF='01,FC,006,U,NU'
ADACMP FNDEF='01,GA,002,U,NU'
ADACMP FNDEF='01,HA,002,U,NU'
ADACMP FNDEF='01,IA,002,U,NU'

ADACMP FNDEF='01,KA,002,U,NU'
ADACMP FNDEF='01,LA,030,A,NU,DE'
ADACMP SUBDE='SB=DE(3,5)'
ADACMP SUPDE='SP=CA(1,1),CB(1,2),CD(1,2)'
ADACMP PHONDE='PA(BA)'
/*

```

次の例は、MVSJOBS データセットのメンバ ADACMPS 内にあります。この例は、ADACMP スパンドレコードパラメータと拡張 MU/PE 制限パラメータの使用方法を示しています。

```

//ADACMP JOB
//*
//* ADACMP COMPRESS
//* COMPRESS A FILE
//* USING MUPEX AND SPAN OPTIONS WITH 2-BYTE MU/PE COUNTS
//*
//CMP EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.VVRS.LOAD <=== ADABAS LOAD
//*
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDEBAND DD DISP=OLD,DSN=EXAMPLE.DBYYYYY.INPUT,UNIT=TAPE, <===
// VOL=SER=TAPE01 <===
//DDAUSBA DD DISP=(NEW,KEEP),DSN=EXAMPLE.DBYYYYY.COMP01,UNIT=DISK, <
// VOL=SER=DISK01,SPACE=(TRK,(200,10),RLSE)

```

```

//DDFEHL      DD  DISP=(NEW,KEEP),DSN=EXAMPLE.DBYYYYY.FEHL,UNIT=DISK, <==
//              VOL=SER=DISK01,SPACE=(TRK,1)
//DDCARD      DD  *
ADARUN  PROG=ADACMP,MODE=MULTI,SVC=XXX,DEVICE=3390,DBID=YYYYY
/*
//DDKARTE     DD  *
ADACMP COMPRESS FILE=1
ADACMP FNDEF='01,AA,008,B,DE'
ADACMP FNDEF='01,BA,020,A,NU,DE'
ADACMP FNDEF='01,BB,015,A,NU,DE'
ADACMP FNDEF='01,BC,001,A,FI'
ADACMP FNDEF='01,CA,001,A,NU,DE'
ADACMP FNDEF='01,CB,002,U,NU,DE'
ADACMP FNDEF='01,CC,010,A,NU,DE'
ADACMP FNDEF='01,CD,002,U,NU,DE'
ADACMP FNDEF='01,DA,005,U,NU'
ADACMP FNDEF='01,DB,020,A,NU,DE'
ADACMP FNDEF='01,DC,015,A,NU,DE'
ADACMP FNDEF='01,DD,002,A,NU,DE'
ADACMP FNDEF='01,DE,005,U,NU,DE'
ADACMP FNDEF='01,DF,008,A,NU,DE'
ADACMP FNDEF='01,FA,020,A,NU,DE'
ADACMP FNDEF='01,FB,006,U,NU,DE'
ADACMP FNDEF='01,FC,006,U,NU'
ADACMP FNDEF='01,GA,002,U,NU'
ADACMP FNDEF='01,HA,002,U,NU'
ADACMP FNDEF='01,IA,002,U,NU'
ADACMP FNDEF='01,KA,002,U,NU'
ADACMP FNDEF='01,LA,030,A,NU,DE'
ADACMP FNDEF='01,MA,010,A,MU,NU,DE'
ADACMP FNDEF='01,PB,PE'
ADACMP FNDEF='02,P1,008,A,NU'
ADACMP FNDEF='02,P2,002,A,NU'
ADACMP FNDEF='02,P3,020,A,NU'
ADACMP SUBDE='SB=DE(3,5)'
ADACMP SUPDE='SP=CA(1,1),CB(1,2),CD(1,2)'
ADACMP PHONDE='PA(BA)'
ADACMP MUPEX                                <== EXTENDED MU/PE FILE
ADACMP MUPECOUNT=2                        <== 2-BYTE MU/PE COUNTS IN INPUT
ADACMP SPAN                                <== SPANNED RECORD FILE
ADACMP DATADEVICE=3390                    <== DATA STORAGE DEVICE TYPE
/*
//

```

## ADACMP DECOMPRESS

次の例は、MVSJOBS データセットのメンバ ADACMPD 内にあります。

```
//ADACMP      JOB
//*
//*      ADACMP COMPRESS
//*      DECOMPRESS A FILE
//*
//DECMP      EXEC PGM=ADARUN
//STEPLIB    DD  DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//*
//DDASSOR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <===DATA
//DDWORKR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <===WORK
//DDDRUCK    DD  SYSOUT=X
//DDPRINT    DD  SYSOUT=X
//SYSUDUMP   DD  SYSOUT=X
//DDEBAND    DD  DISP=OLD,DSN=EXAMPLE.DByyyyy.COMP01,UNIT=TAPE,
//            VOL=SER=TAPE01
//DDAUSBA    DD  DISP=(NEW,KEEP),DSN=EXAMPLE.DByyyyy.DECOMP01,UNIT=DISK,
//            VOL=SER=DISK01,SPACE=(TRK,(200,10),RLSE)
//DDFEHL     DD  DISP=(NEW,KEEP),DSN=EXAMPLE.DByyyyy.FEHL,UNIT=DISK,
//            VOL=SER=DISK01,SPACE=(TRK,1)
//DDCARD     DD  *
ADARUN PROG=ADACMP,MODE=MULTI,SVC=xxxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE    DD  *
ADACMP DECOMPRESS INFILE=1
/*
```

## z/VM

データセット	DD名	ストレージ	追加情報
ユーザー入力データ (COMPRESS 機能)	DDEBAND	テープ/ディスク	
圧縮データ (DECOMPRESS 機能)	DDEBAND	テープ/ディスク	INFILEパラメータを指定している場合は不要
ラージオブジェクト (LB) フィールドを含むファイルを有するデータベースの圧縮データ (COMPRESS 機能)	DDAUSB1	テープ/ディスク	この追加データセットは、圧縮されたラージオブジェクトレコードを受け取ります。このレコードは、最初の出力データセット (DDAUSBA) の圧縮レコードが基本ファイルにロードされるときに LOB ファイルにロードされます。

データセット	DD名	ストレージ	追加情報
圧縮データ (COMPRESS 機能)	DDAUSBA	テープ/ディスク	
圧縮解除データ (DECOMPRESS 機能)	DDAUSBA	テープ/ディスク	
拒否データ	DDFEHL	テープ/ディスク	
ECS エンコードオブジェクト	DDECSOJ	テープ/ディスク	UES (ユニバーサルエンコーディングサポート) に必要
ADACMP レポート	DDDRUCK	ディスク/端末/プリンタ	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	
ADACMP コントロールカードおよびデータ定義	DDKARTE	ディスク/端末/リーダー	

## JCL の例 (z/VM)

### ADACMP COMPRESS

```

DATADEF DDEBAND,DSN=FILE015.CMPD015,MODE=A
DATADEF DDAUSBA,DSN=FILE015.LODD015,MODE=A
DATADEF DDFEHL,DSN=FILE015.CMPERROR,MODE=A
DATADEF DDDRUCK,DSN=ADACMP.DDDRUCK,MODE=A
DATADEF DDPRINT,DSN=ADACMP.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDCARD,DSN=RUNCMP.CONTROL,MODE=A
DATADEF DDKARTE,DSN=FILE001.CMPC015,MODE=A
ADARUN

```

### RUNCMP CONTROL A1 の内容

```
ADARUN PROG=ADACMP,DEVICE=dddd,DB=yyyyyy
```

### FILE001 CMPC015 A1 の内容

```
ADACMP COMPRESS
NUMREC=1000,FDT=1,USERISN,DEVICE=dddd,eeee
```

## VSE

ファイル	ファイル名	ストレージ	論理ユニット	追加情報
ユーザー入力データ (COMPRESS 機能)	EBAND	テープ ディスク	SYS010 *	
圧縮データ (DECOMPRESS 機能)	EBAND	テープ ディスク	SYS010 *	INFILE パラメータを指定している場合は不要
ラージオブジェクト (LB) フィールドを含むファイルを有するデータベースの圧縮データ (COMPRESS 機能)	AUSB1	テープ ディスク		この追加データセットは、圧縮されたラージオブジェクトレコードを受け取ります。このレコードは、最初の出力データセット (AUSBA) の圧縮レコードが基本ファイルにロードされるときに LOB ファイルにロードされます。
圧縮データ (COMPRESS 機能)	AUSBA	テープ ディスク	SYS012 *	
圧縮解除データ (DECOMPRESS 機能)	AUSBA	テープ ディスク	SYS012 *	
拒否データ	FEHL	テープ ディスク	SYS017 *	
ECS エンコードオブジェクト	EC SOJ	テープ ディスク	SYS020 *	UES (ユニバーサルエンコーディングサポート) が必要です。
ADACMP レポート	-	プリンタ	SYS009	
ADARUN メッセージ	-	プリンタ	SYSLST	
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADACMP コントロールカードおよびデータ定義	-	リーダー	SYSIPT	

\* すべてのプログラマ論理ユニットを使用できます。

## JCS の例 (VSE)

VSE プロシージャの内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

COMPRESS の例についてはメンバ ADACMP.X、DECOMPRESS の例についてはメンバ ADACMPD.X を参照してください。

### ADACMP COMPRESS

```
* $$ JOB JNM=ADACMP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
*      COMPRESS A FILE
// JOB ADACMP
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,TAPE
// PAUSE MOUNT LOAD INPUT FILE ON TAPE cuu
// TLBL EBAND,'EXAMPLE.DByyyyy.UNCOMP01'
// MTC REW,SYS010
// DLBL AUSBA,'EXAMPLE.DByyyyy.COMP01',,SD
// EXTENT SYS016,,,,sssss,nnnnn
// ASSGN SYS016,DISK,VOL=DISK01,SHR
// DLBL FEHL,'EXAMPLE.DByyy.FEHL',,SD
// EXTENT SYS017,,,,sssss,nnnnn
// ASSGN SYS017,DISK,VOL=DISK02,SHR
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADACMP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADACMP COMPRESS FILE=1
ADACMP FNDEF='01,AA,008,B,DE'
ADACMP FNDEF='01,BA,020,A,NU,DE'
ADACMP FNDEF='01,BB,015,A,NU,DE'
ADACMP FNDEF='01,BC,001,A,FI'
ADACMP FNDEF='01,CA,001,A,NU,DE'
ADACMP FNDEF='01,CB,002,U,NU,DE'
ADACMP FNDEF='01,CC,010,A,NU,DE'
ADACMP FNDEF='01,CD,002,U,NU,DE'
ADACMP FNDEF='01,DA,005,U,NU'
ADACMP FNDEF='01,DB,020,A,NU,DE'
ADACMP FNDEF='01,DC,015,A,NU,DE'
ADACMP FNDEF='01,DD,002,A,NU,DE'
ADACMP FNDEF='01,DE,005,U,NU,DE'
ADACMP FNDEF='01,DF,008,A,NU,DE'
ADACMP FNDEF='01,FA,020,A,NU,DE'
ADACMP FNDEF='01,FB,006,U,NU,DE'
ADACMP FNDEF='01,FC,006,U,NU'

ADACMP FNDEF='01,GA,002,U,NU'
ADACMP FNDEF='01,HA,002,U,NU'
```

```

ADACMP FNDEF='01,IA,002,U,NU'
ADACMP FNDEF='01,KA,002,U,NU'
ADACMP FNDEF='01,LA,030,A,NU,DE'
ADACMP SUBDE='SB=DE(3,5)'
ADACMP SUPDE='SP=CA(1,1),CB(1,2),CD(1,2)'
ADACMP PHONDE='PA(BA)'
/*
/&
* $$ EOJ

```

## ADACMP DECOMPRESS

```

* $$ JOB JNM=ADACMPD,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
*      DECOMPRESS A FILE
// JOB ADACMPD
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,TAPE
// PAUSE MOUNT LOAD INPUT FILE ON TAPE cuu
// TLBL EBAND,'EXAMPLE.DByyyyy.COMP01'
// MTC REW,SYS010
// DLBL AUSBA,'EXAMPLE.DByyyyy.DECOMP01',,SD
// EXTENT SYS016,,,,sssss,nnnnn
// ASSGN SYS016,DISK,VOL=DISK01,SHR
// DLBL FEHL,'EXAMPLE.DByyy.FEHL',,SD
// EXTENT SYS017,,,,sssss,nnnnn
// ASSGN SYS017,DISK,VOL=DISK02,SHR
*
* *****
*      REMEMBER TO CUSTOMIZE PARAMETERS OF ADABAS UTILITY
* *****
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADACMP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADACMP DECOMPRESS INFILE=1
/*
/&
* $$ EOJ

```

## 22 COMPRESS 機能の出力

---

■ 圧縮されたデータレコード .....	166
■ 拒否されたデータレコード .....	166
■ ADACMP レポート .....	173

このchapterでは、次のトピックについて説明します。

## 圧縮されたデータレコード

ADACMPで処理、編集、および圧縮されたデータレコードは、ファイル定義情報とともに可変長のブロックレコード形式でシーケンシャルデータセットに出力されます。このデータセットは、ADALODユーティリティの入力として使用できます。いくつかのADACMP実行の出力を合わせて、ADALODの入力として使用することもできます。

出力データセットに1つもレコードが含まれていない場合（入力データセットに1つもレコードを指定しなかったか、またはすべてのレコードが拒否された場合）、その出力もADALODユーティリティの入力として使用できます。この場合、ファイルに割り当てられているアソシエータスペースの量が十分であることを確認する必要があります。これは、入力レコード値のサンプルなしでは、ADALODユーティリティで正しい見積もりを行うことができないためです（詳細は、[ADALOD ユーティリティ](#)を参照）。

圧縮レコード内の191より多いMUおよびPEオカレンスを識別する方法については、「[圧縮レコード内の191を超えるMUおよびPEオカレンスの特定](#)」を参照してください。

## 拒否されたデータレコード

ADACMPの圧縮処理中に拒否されたレコードは、すべてDD/FEHLエラーデータセットに書き込まれます。レコードは、可変長ブロックフォーマットで出力され、複数の物理レコードにセグメント化されることがあります。各論理拒否レコードの前には、初期ADAF拒否レコードヘッダーがあります。論理レコードおよびADAFヘッダーがDD/FEHL物理レコード長に収まらない場合、ADAN拒否レコードヘッダーは、論理拒否レコードを構成する残りの物理拒否レコードセグメントに先行します。

これら2つのヘッダーの機能は、次のとおりです。

- ADAFヘッダーは、エラーの状況と適切な情報を示します。
- ADANヘッダーのほうが小さく、拒否レコードの継続およびADAH/ADACヘッダーのエラーレポートに使用されます。

ADAFおよびADANヘッダーのDSECTは、分散Adabas MVSSRCEデータセットのメンバADAFおよびADAN内にあります。

従来、ADACMPエラーに生成されるDD/FEHLエラーデータセットは、FEHL物理レコード長を超える拒否レコードを切り捨てていました。バージョン8では、拒否されたレコードは切り捨てられずにセグメント化されます。この変更により、DD/FEHL LRECL設定には、最低500バイトが必要となります。

HEADER=YESが指定された場合、セグメント化された非圧縮レコードが論理レコードへとアセンブルされるときにエラーが発生することがあります。ADAHヘッダーでエラーになった場合、ADAHレコードは書き込まれますが、後続のADACレコードは次のADAHレコードが処理されるまで書き込まれません。ADACヘッダーでエラーになった場合、先行するADAHヘッダーはペイロードデータなしで書き込まれます。エラーになったADACレコードは完全に書き込まれません。後続のADACレコードは、次のADAHレコードが処理されるまで書き込まれません。

以下に示すレスポンスコードが返されることがあります。

X'E7' (231)	入力レコードが非常に短い場合 (COMPRESS)
X'E8' (232)	出力レコード長のエラー (COMPRESS)
X'E9'(233)	<p>無効な ADAH スパンドレコードヘッダーが見つかった場合 以下に示すサブコードは詳細を表します。</p> <ul style="list-style-type: none"> <li>■ 1 – 正しくない ADAH アイキャッチャー</li> <li>■ 2 – 正しくない ADAH ヘッダー長</li> <li>■ 3 – 予期しない継続インジケータ</li> <li>■ 4 – バイナリの 0 に設定されていない予約済みエリア</li> <li>■ 5 – 無効なセグメント長</li> <li>■ 6 – 合計ペイロードデータ長が MAXLOGRECLEN 設定を超過</li> </ul> <p>スパンドレコードおよび ADAH ヘッダーの詳細は、「スパンドレコード」を参照してください。ADAH ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAH 内にあります。</p>
X'EA'(234)	<p>無効な ADAC スパンドレコードヘッダーが見つかった場合 以下に示すサブコードは詳細を表します。</p> <ul style="list-style-type: none"> <li>■ 1 – 正しくない ADAC アイキャッチャー</li> <li>■ 2 – 正しくない ADAC ヘッダー長</li> <li>■ 3 – 予期しない継続インジケータ</li> <li>■ 4 – バイナリの 0 に設定されていない予約済みエリア</li> <li>■ 5 – 無効なセグメント長</li> <li>■ 6 – 予期しない継続レコードシーケンス番号</li> <li>■ 7 – 無効なセグメントオフセット</li> <li>■ 8 – 累積ペイロードデータ長が ADAH の指定合計長を超過</li> <li>■ 9 – 累積ペイロードデータ長が MAXLOGRECLEN 設定を超過</li> </ul> <p>スパンドレコードおよび ADAC ヘッダーの詳細は、「スパンドレコード」を参照してください。ADAC ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAC 内にあります。</p>

レコード内の最初の無効なフィールドのみが検出され、参照されます。他のエラーがあっても、次に実行するまで検出されません。

### 拒否されたデータレコードの例

次の表は、ADACMPの圧縮処理中に拒否された4つのレコードのFEHL出力を示しています。拒否レコード1および3にはFEHLレコード（ADAF）が1つだけ含まれています。拒否レコード2は2つのFEHLレコード（ADAFおよびADAN）にセグメント化されています。拒否レコード4は3つのFEHLレコード（1つのADAFレコードおよび2つのADANレコード）にセグメント化されています。



**Note:** ADAF および ADAN ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAF および ADAN 内にあります。

拒否レコード	FEHLレコード	FEHL フィールド		説明
		フィールド	値	
1	ADAF	ADAFEYE	ADAF	ADAF ヘッダーのアイキャッチャー
		ADAFLEN	72	ADAF ヘッダー長
		ADAFTYPE	R	タイプ。正しい値は次のとおりです。 H：ADAH ヘッダー P：ADAH ヘッダーとペイロード R：論理レコード
		ADAFIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコード E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み
		ADAFSLEN	22000	セグメント長（ヘッダーに続くペイロードデータの長さ）
		ADAFTOTL	22000	合計長。 <b>注意:</b> すべての ADANSLEN フィールドおよび ADAFSLEN の値の合計は、ADAFTOTL の値と等しくなる必要があります（この例のレコード 1 の場合、 $0 + 22000 = 22000$ ）。
		ADAFISN	1	レコードの ISN
		ADAFLNUM	1	論理レコード番号
		ADAFPNUM	1	物理レコード番号
		ADAFEOFF	5000	論理レコードのエラーオフセット
		ADAFPEX	0	PE インデックス
		ADAFFN	ZA	フィールド名
		ADAFRSP	41	レスポンスコード

拒否 レコー ド	FEHL レコー ド	FEHL フィールド		説明
		フィールド	値	
		ADAFSUB	2	サブコード
		予約済み	0	予約済み
		ADAFDATA	'Record 1 -- Payload Data'	拒否入力データ
2	ADAF	ADAFEYE	ADAF	ADAF ヘッダーのアイキャッチャー
		ADAFLEN	72	ADAF ヘッダー長
		ADAFTYPE	R	タイプ。正しい値は次のとおりです。  H：ADAH ヘッダー P：ADAH ヘッダーとペイロード R：論理レコード
		ADAFIND	C	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコード (ADAN) E：論理レコードの終わり (最終セグメント)
		予約済み	0	予約済み
		ADAFSLEN	27962	セグメント長 (ヘッダーに続くペイロードデータの長さ)
		ADAFTOTL	50000	合計長。  <b>注意:</b> すべての ADANSLEN フィールドおよび ADAFSLEN の値の合計は、ADAFTOTL の値と等しくなる必要があります (この例のレコード 2 の場合、22038 + 27962 = 50000)。
		ADAFISN	2	レコードの ISN
		ADAFLNUM	2	論理レコード番号
		ADAFPNUM	3	物理レコード番号
		ADAFEOFF	35000	論理レコードのエラーオフセット
		ADAFPEX	0	PE インデックス
		ADAFFN	ZA	フィールド名
		ADAFRSP	41	レスポンスコード
		ADAFSUB	2	サブコード
		予約済み	0	予約済み
		ADAFDATA	'Record 2 -- Payload data part 1'	拒否入力データ
	ADAN	ADANEYE	ADAN	ADAN ヘッダーのアイキャッチャー
		ADANLEN	24	ADAN ヘッダー長

拒否 レ コー ド	FEHL レコー ド	FEHL フィールド		説明
		フィールド	値	
		ADANTYPE	R	タイプ。正しい値は次のとおりです。 C：ADAC ヘッダー D：ADAC ヘッダーとペイロード P：ADAH レコードセグメント R：論理レコードセグメント
		ADANIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコード (ADAN) E：論理レコードの終わり (最終セグメント)
		予約済み	0	予約済み
		ADANSLEN	22038	セグメント長 (ヘッダーに続くペイロードデータの長さ)
		ADANOFF	27962	論理レコードのエラーオフセット。
		ADANDATA	'Record 2 -- Payload data part 2'	継続される拒否入力データ
3	ADAF	ADAFEYE	ADAF	ADAF ヘッダーのアイキャッチャー
		ADAFLEN	72	ADAF ヘッダー長
		ADAFTYPE	P	タイプ。正しい値は次のとおりです。 H：ADAH ヘッダー P：ADAH ヘッダーとペイロード R：論理レコード
		ADAFIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコード (ADAN) E：論理レコードの終わり (最終セグメント)
		予約済み	0	予約済み
		ADAFSLEN	20000	セグメント長 (ヘッダーに続くペイロードデータの長さ)
		ADAFTOTL	20000	合計長。  <b>注意:</b> すべての ADANSLEN フィールドおよび ADAFSLEN の値の合計は、ADAFTOTL の値と等しくなる必要があります (この例のレコード 3 の場合、0 + 20000 = 20000)。
		ADAFISN	3	レコードの ISN
		ADAFLNUM	3	論理レコード番号
		ADAFPNUM	4	物理レコード番号
		ADAFEOFF	0	論理レコードのエラーオフセット
		ADAFPEX	0	PE インデックス
		ADAFFN		フィールド名

拒否 レコー ド	FEHL レコー ド	FEHL フィールド		説明	
		フィールド	値		
		ADAFRSP	233	レスポンスコード	
		ADAFSUB	1	サブコード	
		予約済み	0	予約済み	
		ADAFDATA	ADAH および ペイロードデー タ	拒否入力データ	
4	ADAF	ADAFEYE	ADAF	ADAF ヘッダーのアイキャッチャー	
		ADAFLEN	72	ADAF ヘッダー長	
		ADAFTYPE	H	タイプ。正しい値は次のとおりです。  H：ADAH ヘッダー P：ADAH ヘッダーとペイロード R：論理レコード	
		ADAFIND	C	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコード (ADAN) E：論理レコードの終わり (最終セグメント)	
		予約済み	0	予約済み	
		ADAFSLEN	32	セグメント長 (ヘッダーに続くペイロードデータの長さ)	
		ADAFTOTL	10064	合計長。  <b>注意:</b> すべての ADANSLEN フィールドおよび ADAFSLEN の値の合計は、ADAFTOTL の値と等しくなる必要があります (この例のレコード 4 の場合、 $32 + 32 + 10000 = 10064$ )。	
		ADAFISN	4	レコードの ISN	
		ADAFLNUM	4	論理レコード番号	
		ADAFPNUM	8	物理レコード番号	
		ADAFEOFF	0	論理レコードのエラーオフセット	
		ADAFPEX	0	PE インデックス	
		ADAFFN		フィールド名	
		ADAFRSP	234	レスポンスコード	
		ADAFSUB	3	サブコード	
		予約済み	0	予約済み	
		ADAFDATA	ADAH ヘッ ダー	拒否入力データ	
		ADAN	ADANEYE	ADAN	ADAN ヘッダーのアイキャッチャー
			ADANLEN	24	ADAN ヘッダー長

拒否 レ コー ド	FEHL レコー ド	FEHL フィールド		説明	
		フィールド	値		
		ADANTYPE	C	タイプ。正しい値は次のとおりです。  C：ADAC ヘッダー D：ADAC ヘッダーとペイロード P：ADAH レコードセグメント R：論理レコードセグメント	
		ADANIND	C	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコード（ADAN） E：論理レコードの終わり（最終セグメント）	
		予約済み	0	予約済み	
		ADANSLEN	32	セグメント長（ヘッダーに続くペイロードデータの長さ）	
		ADANOFF	32	論理レコードのエラーオフセット。	
		ADANDATA	ADAC ヘッ ダー	継続される拒否入力データ	
		ADAN	ADANEYE	ADAN	ADAN ヘッダーのアイキャッチャー
		ADAN	ADANLEN	24	ADAN ヘッダー長
		ADAN	ADANTYPE	D	タイプ。正しい値は次のとおりです。  C：ADAC ヘッダー D：ADAC ヘッダーとペイロード P：ADAH レコードセグメント R：論理レコードセグメント
		ADAN	ADANIND	E	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコード（ADAN） E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み	
		ADANSLEN	10000	セグメント長（ヘッダーに続くペイロードデータの長さ）	
		ADANOFF	64	論理レコードのエラーオフセット。	
		ADANDATA	ADAC および ペイロードデー タ	継続される拒否入力データ	

## ADACMP レポート

ADACMPでは、圧縮レコードに必要なデータストレージのおよそのスペース量（ブロック単位およびシリンダ単位）を計算します。この情報は、DEVICEパラメータで指定するさまざまなデバイスタイプに必要なスペースを、5%～30%の各データストレージパディングファクタについて示した表の形で出力されます。

ADACMP レポートの出力例を以下に示します。

### PARAMETERS:

```
ADACMP COMPRESS NUMREC=1000
ADACMP FNDEF='01,AA,8,B,DE'
ADACMP FNDEF='01,BA,6,A,NU'
ADACMP FNDEF='01,BB,8,P,NU'
ADACMP FNDEF='01,AD,1,A,FI'
ADACMP SUBDE='CA=BA(1,3)'
```

### COMPRESS PROCESSING STATISTICS:

```
NUMBER OF RECORDS READ          1,000
NUMBER OF INCORRECT RECORDS      0
NUMBER OF COMPRESSED RECORDS    1,000

RAW DATA                        24,000 BYTES
COMPRESSED DATA                 16,656 BYTES
COMPRESSION RATE                 31.9 %
LARGEST COMPRESSED RECORD        20 BYTES
```

### DATASTORAGE SPACE REQUIREMENTS:

I	DEVICE	I	PADDING	I	BLOCKSIZE	I	NUMBER OF	I
I		I	FACTOR	I	BYTES	I	BLOCKS	CYLS
I	3380	I		I	4,820	I		I
I		I	5%	I	4,578	I	4	1 I
I		I	10%	I	4,337	I	4	1 I
I		I	15%	I	4,096	I	5	1 I
I		I	20%	I	3,856	I	5	1 I
I		I	25%	I	3,615	I	5	1 I
I		I	30%	I	3,373	I	5	1 I
I		I		I		I		I

### TEMP SPACE ESTIMATION:

I	DEVICE	I	BLOCKSIZE	I	NUMBER OF	I
---	--------	---	-----------	---	-----------	---

## COMPRESS 機能の出力

```

I          I    BYTES  I      BLOCKS      CYLS  I
I-----I-----I-----I-----I
I  3380  I    7,476  I          5          1  I
I-----I-----I-----I-----I

```

THE LARGEST DESCRIPTOR IS AA, IT WILL OCCUPY 1 TEMP BLOCKS

SORT SPACE ESTIMATION:

```

I DEVICE I BLOCKSIZE I    LWP          I      NR OF      I
I          I (BYTES)  I (BYTES)          I  BLOCKS    CYLS I
-----I-----I-----I-----I-----I
I  3380  I    7476  I  139264 (MINIMUM) I      2      1  I
I          I          I  1048576 (DEFAULT) I      2      1  I
I          I          I  139264 (OPTIMUM) I      2      1  I
I-----I-----I-----I-----I

```

圧縮率は、圧縮ルーチンに実際に入力されたデータ量に基づいて計算されます。フォーマット要素 nX（フィールドに空白を入力するために使用）でスキップしたフィールドはカウントされません。

ADACMP COMPRESS の実行で SPAN を指定した場合、スパンドータストレージレコードの統計情報も出力されます。

Spanned Record Statistics:

```

-----
Number of Non-Spanned records          2
Number of Spanned records              8
Min Number of Segments                 2
Max Number of Segments                 5
Avg Number of Segments                 4

Max MU Count                           0
Max PE Count                           300

```

ラージオブジェクト（LB）フィールドが圧縮される場合、LBフィールド（レポートで"LOB"として表示される）の統計情報が出力されます。

Large Objects Statistics:

```

-----
Number of bytes for largest LOB        10,124,996
Total number of LOBs                   5
Total number of outsourced LOBs        5
Number of LOB file records             10,085

```

# 23 DECOMPRESS 機能の出力

---

- 拒否されたデータレコード ..... 176

ADACMPDECOMPRESS機能では、各レコードを圧縮解除し、そのレコードをシーケンシャルデータセットに格納します。レコードは可変長のブロック形式で出力されます。各非圧縮レコードは、次のフォーマットに従って、ISN オプションの指定あり、または指定なしで出力されます。

```
length xx [ISN] data
```

ここでは次の内容を表しています。

<i>length</i>	データの長さを示す 2 バイトの 2 進数+8 (ISN パラメータの指定がない場合は +4)
<i>xx</i>	バイナリの 0 を含む 2 バイトフィールド
ISN	レコードの ISN 値を示す 4 バイトの 2 進数
<i>data</i>	非圧縮データレコード

データレコードのフィールドの順番は、ファイルのアンロード時のフィールド定義テーブル (FDT) と同じです。標準長とフォーマットはフィールドごとに適用されます。

フィールド値が標準長を超えた場合、TRUNCATE パラメータが指定されていると、そのフィールドが英数字フォーマットである場合、値は標準長まで切り捨てられます。それ以外のフォーマットの場合、そのレコードは ADACMP によって DD/FEHL エラーデータセットに書き込まれます (次のchapterの「拒否されたデータレコード」を参照)。

レコード内のマルチプルバリューフィールドやピリオディックグループのカウンタバイトは、すべて非圧縮データ出力に含まれます。ADACMP では、MU フィールドまたは PE グループが空の場合、カウンタ 1 が生成されます。これにより、DECOMPRESS 処理の出力を後の COPMPRESS 処理の入力として使用できるようになります。

## 拒否されたデータレコード

ADACMP では、圧縮レコードにフィールド定義テーブル (FDT) 内で定義されている標準長を超えるフィールドが含まれていると、そのレコードが拒否されます (TRUNCATE パラメータを指定している場合を除く)。

ADACMP の圧縮解除処理中に拒否されたレコードは、すべて DD/FEHL エラーデータセットに書き込まれます。レコードは、可変長ブロックフォーマットで出力され、複数の物理レコードにセグメント化されることがあります。各論理拒否レコードの前には、初期 ADAF 拒否レコードヘッダーがあります。論理レコードおよび ADAF ヘッダーが DD/FEHL 物理レコード長に収まらない場合、ADAN 拒否レコードヘッダーは、論理拒否レコードを構成する残りの物理拒否レコードセグメントに先行します。

これら 2 つのヘッダーの機能は、次のとおりです。

- ADAF ヘッダーは、エラーの状況と適切な情報を示します。

- ADANヘッダーのほうが小さく、拒否レコードの継続および ADAH/ADACヘッダーのエラーレポートに使用されます。

ADAF および ADAN ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAF および ADAN 内にあります。

従来、ADACMP エラーに生成される DD/FEHL エラーデータセットは、FEHL 物理レコード長を超える拒否レコードを切り捨てていました。バージョン 8 では、拒否されたレコードは切り捨てられずにセグメント化されます。この変更により、DD/FEHL LRECL 設定には、最低 500 バイトが必要となります。

以下に示すレスポンスコードが返されることがあります。

X'E7' (231)	入力レコードが非常に短い場合 (DECOMPRESS)
X'E8' (232)	出力レコード長のエラー (DECOMPRESS)

 **Notes:**

1. レコード内の最初の無効なフィールドのみが検出され、DD/FEHL で参照されます。レコード内の他のエラーは検出も記録もされません。

### 拒否されたデータレコードの例

次の表は、ADACMP の圧縮解除処理中に拒否された 2 つのレコードの FEHL 出力を示しています。拒否レコード 1 には、FEHL レコード (ADAF) が 1 つのみ含まれています。拒否レコード 2 は、2 つの FEHL レコード (ADAF および ADAN) にセグメント化されています。

 **Note:** ADAF および ADAN ヘッダーの DSECT は、分散 Adabas MVSSRCE データセットのメンバ ADAF および ADAN 内にあります。

拒否レコード	DDFEHL レコード	DDFEHL フィールド		説明
		フィールド	値	
1	ADAF	ADAFEYE	ADAF	ADAF ヘッダーのアイキャッチャー
		ADAFLEN	72	ADAF ヘッダー長
		ADAFTYPE	R	タイプ。正しい値は次のとおりです。 H：ADAH ヘッダー P：ADAH ヘッダーとペイロード R：論理レコード
		ADAFIND	E	継続インジケータ。正しい値は次のとおりです。 C：後続の継続レコード E：論理レコードの終わり (最終セグメント)

拒否 レ コー ド	DDFEHL レコー ド	DDFEHL フィールド		説明
		フィールド	値	
		予約済み	0	予約済み
		ADAFSLEN	22000	セグメント長
		ADAFTOTL	22000	合計長
		ADAFISN	1	レコードの ISN
		ADAFNUM	1	論理レコード番号
		ADAFNUM	1	物理レコード番号
		ADAFEOFF	5000	論理レコードのエラーオフセット
		ADAFPEX	0	PE インデックス
		ADAFFN	ZA	フィールド名
		ADAFRSP	41	レスポンスコード
		ADAFSUB	2	サブコード
		予約済み	0	予約済み
		ADAFDATA	'Record 1 -- Payload Data'	拒否入力データ
2	ADAF	ADAFEYE	ADAF	ADAF ヘッダーのアイキャッチャー
		ADAFLEN	72	ADAF ヘッダー長
		ADAFTYPE	R	タイプ。正しい値は次のとおりです。  H：ADAH ヘッダー P：ADAH ヘッダーとペイロード R：論理レコード
		ADAFIND	C	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコード (ADAN) E：論理レコードの終わり (最終セグメント)
		予約済み	0	予約済み
		ADAFSLEN	27962	セグメント長
		ADAFTOTL	50000	合計長
		ADAFISN	2	レコードの ISN
		ADAFNUM	2	論理レコード番号
		ADAFNUM	3	物理レコード番号
		ADAFEOFF	35000	論理レコードのエラーオフセット
		ADAFPEX	0	PE インデックス
		ADAFFN	ZA	フィールド名
		ADAFRSP	41	レスポンスコード
		ADAFSUB	2	サブコード

拒否 レ コー ド	DDFEHL レコー ド	DDFEHL フィールド		説明
		フィールド	値	
		予約済み	0	予約済み
		ADAFDATA	'Record 2 -- Payload data part 1'	拒否入力データ
	ADAN	ADANEYE	ADAN	ADAN ヘッダーのアイキャッチャー
		ADANLEN	24	ADAN ヘッダー長
		ADANTYPE	R	タイプ。正しい値は次のとおりです。  C：ADAC ヘッダー D：ADAC ヘッダーとペイロード P：ADAH レコードセグメント R：論理レコードセグメント
		ADANIND	E	継続インジケータ。正しい値は次のとおりです。  C：後続の継続レコード（ADAN） E：論理レコードの終わり（最終セグメント）
		予約済み	0	予約済み
		ADANSLEN	22038	セグメント長  <b>注意:</b> ADANSLEN および ADAFSLEN の値の合計は、 ADAFTOTL の値と等しくなる必要があります（この例の 場合、22038 + 27962 = 50000）。
		ADANOFF	27962	論理レコードのエラーオフセット。
		ADANDATA	'Record 2 -- Payload data part 2'	継続される拒否入力データ

---

# 24 ADACNV ユーティリティ：データベースの変換

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- **CONVERT**：上位バージョンへのデータベースの変換
- **REVERT**：下位バージョンへのデータベースの変換
- **JCL/JCS** の必要項目と指定例



# 25 機能概要

---

■ データベースステータス .....	184
■ 手順 .....	216

ADACNV ユーティリティを使用すると、バージョン 5.2 以上の Adabas データベースをより上位のバージョンに変換 (CONVERT) したり、上位バージョンの Adabas データベースをより下位のバージョンに逆変換 (REVERT) できます。

 **Caution:** データベースを変換する前に、すべてのアクティブなニュークリアスおよびユーティリティジョブを正常に終了させる必要があります。

データベースの完全性を確保するために、最初に ADACNV は、変更するブロックを中間ストレージ、すなわちシーケンシャルデータセット DD/FILEA に書き込みます。変更されたすべてのブロックが DD/FILEA に書き込まれると、非復帰点に到達し、変更されたブロックはデータベースに書き込まれます。ADACNV が非復帰点の後に異常終了すると、RESTART パラメータが指定されている場合は、ADACNV の実行が開始されます。このとき、ADACNV では最初に DD/FILEA の内容を読み取ってデータベースに書き込みます。

TEST パラメータは、変換または逆変換できるか調べるためのパラメータで、変更をデータベースに書き込みません。したがって、TEST パラメータを使用するときは、ADACNV の実行前にデータベース上の全動作を終了させる必要はありません。

## データベースステータス

---

内部的には、ユーティリティは、ターゲットバージョンに達するまで、一度に 1 バージョンずつ変換または逆変換を行います。つまり、TEST パラメータを指定しないで ADACNV を実行する場合は、現在のデータベースとターゲットデータベースの間で、変換または逆変換に必要なすべての要件が確実に満たされていることが重要です。

変換または逆変換を開始する前に、ADACNV によってデータベースのステータスがチェックされます。

- DIB は空である必要があります。つまり、アクティブまたは異常終了した Adabas ニュークリアスまたはユーティリティが存在しないようにする必要があります。RESTART が指定されている場合、DIB にはタイムスタンプを含む ADACNV エントリが含まれている必要があります。
- バージョン 5.2 から 5.3 以上のバージョンに変換する場合は、24 バイトの拡張ヘッダに対応するための十分な空きスペースがチェックポイントブロック 8 にあることが必要です。バージョン 5.2 への逆変換する場合、チェックポイントブロック 20~24 は空である必要があります。
- WORK データセットには、保留中の自動再スタートが存在しないようにする必要があります。

このチェックが正常に行われると、ADACNV はデータベースをロックして DIB エントリを作成します。

逆変換の場合は、ターゲットバージョンにない機能が使用されていないかどうか ADACNV によってチェックされ、使用されている場合はメッセージが出力されます。



**Note:** ロードされないが、フィールド定義テーブル (FDT) ブロックに FDT が格納されているファイルがデータベース内に存在する可能性があります。データベースのバージョン 8 への変換中に、ADACNV がこのような FDT を検出した場合、後処理の一部としてこれらの FDT は削除されます。

## 手順

---

Adabas データベースを変換または逆変換するには、次の手順に従います。

1. ニュークリアスがアクティブである場合は、ADAEND を使用して終了します。
2. ADARESPLCOPY/CLCOPY を使用してすべてのプロテクションログとコマンドログをコピーします。

インストール環境によっては、この手順はユーザー出口 2 によって自動的に実行される場合があります。

ログがコピーされるまで待機します。

3. 必要に応じて、データベースの全体またはデルタのバックアップを行います。
4. ADACNV ユーティリティを実行します。
5. 変換または逆変換を行ったバージョンのニュークリアスを開始します。



**Important:** データベースの保全性を確保するには、DD/FILEA が恒久的に定義されており、ADACNV が正常に終了した場合にのみ削除される必要があります。DD/FILEA データセットは、ジョブの終了時に自動的に削除されるテンポラリデータセットとして定義しないようにする必要があります。



# 26 CONVERT:上位バージョンへのデータベースの変換

---

■ オプションパラメータ .....	188
■ 変換の考慮点 .....	189
■ 例 .....	190

CONVERT 機能は、最新のニュークリアスセッションの Adabas バージョンから開始します。

```
ADACNV CONVERT [IGNPPT]
                [NOUSERABEND]
                [PLOGDEV = { multiple-PLOG-device-type | ADARUN-device }]
                [RESTART]
                [TEST]
                [TOVERS = { target-version | ADACNV-version }]
```

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

---

### IGNPPT : Parallel Participant Table PLOG エントリを無視

Parallel Participant Table (PPT) 構造を使用する Adabas バージョンから Adabas の上位バージョンに変換するとき、コピー／マージされていない現バージョンから1つ以上のプロテクションログ (PLOG) が検出されると、エラーが出力され、変換は失敗します。

コピー／マージされていない PLOG があるにもかかわらず処理を続行する場合は、IGNPPT を指定してください。

 **Note:** DDPLOG<sub>x</sub> ステートメントが、PPT の DDPLOG<sub>x</sub> データセットに加えて JCL で指定されている場合は、これらのデータセットは空である必要があります。空でない場合は、エラーが出力されます。IGNPPT は PPT 処理のみに関連しています。PLOG データセットを JCL に指定する場合は、これらのデータセットは空である必要があります。

### NOUSERABEND : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### PLOGDEV : マルチプロテクションログのデバイスタイプ

PLOGDEV には、変換されるマルチプロテクションログデータセットが含まれている物理デバイスタイプを指定します。PLOGDEV を指定しない場合、ADARUN DEVICE パラメータで指定されたデバイスタイプが使用されます。

**RESTART**：非復帰点以降の復帰

ADACNV が非復帰点以降、言い換えると全変更ブロックを DD/FILEA に書き込んだ後に異常終了した場合に、RESTART パラメータは、ADACNV に DD/FILEA の内容の読み取りから始めてデータベースへの書き込みを続行するように指示します。

**TEST**：変換テスト

TEST パラメータを使用すると、変換処理が可能であるかどうかについて、実際にデータベースに変更内容を書き込むことなくテストできます。

**TOVERS**：変換先バージョン

ADACNV を実行した後に得ることができる Adabas データベースの 2 文字のバージョン（バージョンおよび改訂レベル）です。TOVERS パラメータを

- 指定する場合は、ソースバージョンよりも上位のバージョンを指定する必要があります。
- 指定しない場合、ADACNV は、実行されているバージョンをターゲットバージョンとして使用します。

バージョンは *vr* の形式で指定し、バージョンと改訂レベルを表します（"74" または "81" など）。



**Note:** ロードされないが、フィールド定義テーブル（FDT）ブロックに FDT が格納されているファイルがデータベース内に存在する可能性があります。データベースのバージョン 8 への変換中に、ADACNV がこのような FDT を検出した場合、後処理の一部としてこれらの FDT は削除されます。

## 変換の考慮点

以下では、ADACNV によってどのような手順で変換が実行されるかについて、概要を説明します。

### 全バージョン

- WORK データセットおよびマルチ PLOG データセット（提供されている場合）のデータプロテクションエリアがバイナリの 0 に設定されます。

### バージョン 6.1 から 6.2

- Adabas Delta Save DLOG エリアヘッダーに正しいバージョンが設定されます。

### バージョン 6.2 から 7.1

- Adabas Delta Save DLOG エリアヘッダーに正しいバージョンが設定されます。

### バージョン 7.1 からバージョン 8

CONVERT を使用して、Adabas 7.1 から Adabas 8 へデータベースを直接変換することはできません。これを行うには、最初にデータベースを Adabas 7.2 または 7.4 に変換し、次に Adabas 8 に変換する必要があります。

すべてのバージョンからバージョン 8

ロードされないが、フィールド定義テーブル (FDT) ブロックに FDT が格納されているファイルがデータベース内に存在する可能性があります。データベースのバージョン 8 への変換中に、ADACNV がこのような FDT を検出した場合、後処理の一部としてこれらの FDT は削除されます。

## 例

---

```
ADACNV CONVERT TOVERS=81
```

最新のニュークリアスセッションで選択された Adabas のバージョンをバージョン 8.1 データベースに変換します。

# 27 REVERT：下位バージョンへのデータベースの逆 変換

---

- 基本パラメータとサブパラメータ ..... 192
- オプションパラメータ ..... 192
- 逆変換の考慮点 ..... 193
- 例 ..... 194

REVERT 機能は、最新のニュークリアスセッションの Adabas バージョンから開始します。

```
ADACNV REVERT [TOVERS = target-version ]  
               [IGNPPT]  
               [NOUSERABEND]  
               [PLOGDEV = { multiple-PLOG-device-type | ADARUN-device }]  
               [RESTART]  
               [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータとサブパラメータ

---

### TOVERS：変換先バージョン

ADACNV を実行した後に得ることができる Adabas データベースの 2 文字のバージョン（バージョンおよび改訂レベル）です。TOVERS パラメータの値はソースバージョンよりも下位のバージョンである必要があります。

バージョンは *vr* の形式で指定し、バージョンと改訂レベルを表します（61または81など）。

## オプションパラメータ

---

### IGNPPT：Parallel Participant Table PLOG エントリを無視

Parallel Participant Table (PPT) 構造を使用する Adabas バージョンから Adabas の下位バージョンに逆変換するとき、コピー／マージされていない現バージョンから 1 つ以上のプロテクションログ (PLOG) が検出されると、エラーが出力され、変換は失敗となります。

IGNPPT を指定すると、コピー／マージされていない PLOG があっても、ユーティリティによる処理は続行されます。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PLOGDEV**：マルチプロテクションログのデバイスタイプ

PLOGDEVには、変換されるマルチプロテクションログデータセットが含まれている物理デバイスタイプを指定します。PLOGDEVを指定しない場合、ADARUN DEVICE パラメータで指定されたデバイスタイプが使用されます。

**RESTART**：非復帰点以降の復帰

ADACNV が非復帰点以降、言い換えると全変更ブロックを DD/FILEA に書き込んだ後に異常終了した場合に、RESTART パラメータは、ADACNV に DD/FILEA の内容の読み取りから始めてデータベースへの書き込みを続行するように指示します。

**TEST**：変換テスト

TEST パラメータを使用すると、変換処理が可能であるかどうかについて、実際にデータベースに変更内容を書き込むことなくテストできます。

## 逆変換の考慮点

以下では、ADACNVによってどのような手順で変換が実行されるかについて、概要を説明します。

**全バージョン**

- ターゲットバージョンでサポートされていない Adabas 機能が現バージョンで使用されている場合は逆変換を実行できません。このステートメントは、データベースの構造に影響を与える Adabas のすべての機能に適用されます。

**バージョン 8 からすべての下位バージョン**

データベースによって Adabas 8 の次の拡張機能のいずれかが使用されている場合は、ADACNV によってデータベースを Adabas 8 以前のバージョンに逆変換することができません。

- 5 個を超える ASSO、DATA、または DSST エクステンント
- 5 個を超えるファイルエクステンント
- スパンドレコードの使用が可能なファイル
- 191 を超える MU および PE オカレンスが可能なファイル
- ラージオブジェクト (LB) フィールドを使用するファイル
- NB (空白圧縮なし) オプションが設定されたフィールドを持つファイル
- 2 バイトファイル番号を持つシステムファイル

下位変換を完了するには、最初に、Adabas データベースからこれらの新機能を含むファイルを削除する必要があります。

Adabas 8 で提供されている次のその他の新機能を使用しても、Adabas 7.4 への下位変換が可能です。ただし、新機能は Adabas 7.4 では使用できません。

- ACBX インターフェイス経由（長いバッファ、セグメント化されたバッファを使用する場合など）で発行された Adabas コマンド
- 新しいフォーマットバッファ機能（長さインジケータなど）を使用したコマンド

### バージョン 8 からバージョン 7.1

REVERT を使用して Adabas 8 から直接 Adabas 7.1 には逆変換することはできません。最初に、データベースを Adabas 7.4 に逆変換し、その後、7.4 から 7.1 に逆変換します。

### バージョン 7.1 から 6.2

- バージョン 7.1 では、フリースペーステーブル (FST) が 1RABN (RABN10) から 5RABN (RABN 10~14) に拡張されます。ADACNV により、すべての FST エントリが 1 RABN に収まるかどうかチェックされます。収まらない場合は、最小の FST エクステントが削除されます。この処理は FST が 1 ASSO ブロックに収まるまで繰り返し実行されます。該当するメッセージが出力されます。
- Adabas Delta Save DLOG エリアヘッダーに正しいバージョンが設定されます。

### バージョン 6.2 から 6.1

- Adabas Delta Save DLOG エリアヘッダーに正しいバージョンが設定されます。

## 例

---

```
ADACNV REVERT TOVERS=73
```

最新のニュークリアスセッションで選択された Adabas バージョンをバージョン 7.3 の Adabas データベースに逆変換します。

# 28

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	196
▪ z/OS .....	244
▪ z/VM .....	200
▪ VSE .....	253

このセクションでは、ADACNV with BS2000、z/OS、z/VM、および VSE システムの実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1	ディスク	
マルチプロテクションログ	DDPLOGRn	ディスク	
中間ストレージ	DDFILEA	テープ/ディスク	注参照
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADACNV パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/ DDPRINT		メッセージおよびコード
ADACNV メッセージ	SYSLST/ DDDRUCK		メッセージおよびコード

 **Note:** 中間ストレージが読み取られる回数は定義されていません。このストレージがテープ/カセット上にある場合、テープが解放されることを防止するために ADARUN パラメータ TAPEREL=NO を使用する必要があります。Software AG では、テープ解放コマンドをジョブに配置し、ジョブが終了したときにテープ/カセットユニットが解放されるようにすることをお勧めします。次の例を参照してください。

### ADACNV JCL 例 (BS2000)

#### 中間ディスクファイルストレージ使用

SDF フォーマットの場合：

```

/.ADACNV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A C N V CONVERT THE DATABASE TO NEW VERSION
/REMARK *
/DELETE-FILE ADAyyyyy.FILEA
/SET-JOB-STEP
/CREATE-FILE ADAyyyyy.FILEA,PUB(SPACE=(4800,480))
/SET-JOB-STEP
/ASS-SYSLST L.CNV.DATA
/ASS-SYSDTA *SYSCMD
    
```

```

/SET-FILE-LINK DDLIB,ADAhrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAyyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR1,ADAyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADAyyyyy.PLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDFILEA,ADAyyyyy.FILEA
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADACNV,DB=yyyyy,IDTNAME=ADABAS5B
ADACNV CONVERT TOVERS=vr
/LOGOFF SYS-OUTPUT=DEL

```

### ISP フォーマットの場合：

```

/.ADACNV LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A C N V CONVERT THE DATABASE TO NEW VERSION
/REMARK *
/SYSFILE SYSLST=L.CNV.DATA
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyy.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyy.WORK,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyy.PLOGR1,LINK=DDPLOGR1,SHARUPD=YES
/FILE ADAyyyyy.PLOGR2,LINK=DDPLOGR2,SHARUPD=YES
/FILE ADAyyyyy.FILEA,LINK=DDFILEA,SPACE=(4800,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADACNV,DB=yyyyy,IDTNAME=ADABAS5B
ADACNV CONVERT TOVERS=vr
/LOGOFF NOSPOOL

```

### 中間テープ／カセットファイルストレージ使用

#### SDF フォーマットの場合：

```

/.ADACNV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A C N V CONVERT THE DATABASE TO NEW VERSION
/REMARK * INTERMEDIATE TAPE/CASSETTE STORAGE
/REMARK *
/DELETE-FILE ADAyyyyy.FILEA
/SET-JOB-STEP
/CREATE-FILE ADAyyyyy.FILEA,TAPE(DEV-TYPE=T-C1,VOL=ADA001)
/SET-JOB-STEP
/ASS-SYSLST L.CNV.DATA

```

```

/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADA vrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAyyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR1,ADAyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADAyyyyy.PLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDFILEA,ADAyyyyy.FILEA,TAPE(FILE-SEQ=1),OPEN-MODE=OUTIN
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADACNV,DB=yyyyy,IDTNAME=ADABAS5B,TAPEREL=NO
ADACNV CONVERT TOVERS=vr
/SET-JOB-STEP
/REMARK * NOW RELEASE THE TAPE
/REM-FILE-LINK DDFILEA,UNL-REL-TAPE=YES
/LOGOFF SYS-OUTPUT=DEL

```

**ISP** フォーマットの場合：

```

/.ADACNV LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A C N V CONVERT THE DATABASE TO NEW VERSION
/REMARK * INTERMEDIATE TAPE/CASSETTE STORAGE
/REMARK *
/SYSFILE SYSLST=L.CNV.DATA
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyy.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyy.WORK,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyy.PLOGR1,LINK=DDPLOGR1,SHARUPD=YES
/FILE ADAyyyyy.PLOGR2,LINK=DDPLOGR2,SHARUPD=YES
/FILE ADAyyyyy.FILEA,LINK=DDFILEA,DEVICE=T C1,VOLUME=ADA001

/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADACNV,DB=yyyyy,IDTNAME=ADABAS5B,TAPEREL=NO
ADACNV CONVERT TOVERS=vr
/STEP
/REMARK * NOW RELEASE THE TAPE
/REL DDFILEA,UNLOAD

```

/LOGOFF NOSPOOL

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	
マルチプロテクションログ	DDPLOGRn	ディスク	
中間ストレージ	DDFILEA	テープ/ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADACNV パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADACNV メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADACNV JCL 例 (z/OS)

この例については、MVSJOBS データセットの ADACNV を参照してください。

```
//ADACNV      JOB
//*
//*      ADACNV:
//*      EXAMPLE HOW TO USE ADACNV TO CONVERT DATABASE
//*      TO A DIFFERENT VERSION
//*
//CNV        EXEC PGM=ADARUN
//STEPLIB    DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1  <=== ASSO
//DDDATAR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1  <=== DATA
//DDWORKR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1  <=== WORK
//DDPLOGR1   DD   DSN=EXAMPLE.DByyyyy.PLOGR1,DISP=SHR  <=== PLOG 1
//DDPLOGR2   DD   DSN=EXAMPLE.DByyyyy.PLOGR2,DISP=SHR  <=== PLOG 2

//DDFILEA    DD   DSN=EXAMPLE.DByyyyy.FILEA,           <=== INTERMEDIATE FILE
//            UNIT=SYSDA,SPACE=(TRK,(150,150),RLSE),
//            DISP=(NEW,CATLG)
//DDDRUCK    DD   SYSOUT=X
//DDPRINT    DD   SYSOUT=X
//SYSUDUMP   DD   SYSOUT=X
```

## JCL/JCS の必要項目と指定例

```
//DDCARD DD *
ADARUN PROG=ADACNV,SVC=xxx,DE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADACNV CONVERT TOVERS=vr
/*
```

## z/VM

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	
マルチプロテクションログ	DDPLOGRn	ディスク	
中間ストレージ	DDFILEA	テープ/ディスク	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADACNV パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADACNV メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

## ADACNV JCL 例 (z/VM)

```
DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDWORKR1,DSN=ADABASV.V.WORK,VOL=WORKV1
DATADEF DDPLOGR1,DSN=ADABASV.V.PLOGR1,VOL=PLOGV
DATADEF DDPLOGR2,DSN=ADABASV.V.PLOGR2,VOL=PLOGV
DATADEF DDFILEA,DSN=ADACNV.FILEA,MODE=A
DATADEF DDPRINT,DSN=ADACNV.DDPRINT,MODE=A
DATADEF DUMP,DUMMY

DATADEF DDDRUCK,DSN=ADACNV.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNCNV.CONTROL,MODE=A
DATADEF DDKARTE,DSN=CONVERT.CONTROL,MODE=A
ADARUN
```

## RUNCNV CONTROL A1 の内容

```
ADARUN PROG=ADACNV,DEVICE=dddd,DB=yyyyyy
```

## CONVERT CONTROL A1 の内容

```
ADACNV CONVERT TOVERS=vr
```

## VSE

ファイル	ファイル名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	
データストレージ	DATARn	ディスク	*	
ワーク	WORKRn	ディスク	*	
マルチプロテクションログ	PLOGRn	ディスク	*	
中間ストレージ	FILEA	テープ ディスク	SYS010 *	
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	オペレーションマニュアル
ADACNV パラメータ	-	リーダー	SYSIPT	
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADACNV メッセージ	-	プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADACNV JCS 例 (VSE)

VSE プロシージャの内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

この例については、メンバー ADACNV.X を参照してください。

```
* $$ JOB JNM=ADACNV,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
*      CONVERT DATABASE TO NEW VERSION
// JOB ADACNV
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// DLBL FILEA,'ADACNV.WORK.FILE',0,SD
// EXTENT SYS015,,,,ssss,nnnn
// ASSGN SYS015,DISK,VOL=vvvvvv,SHR
// EXEC ADARUN,SIZE=ADARUN
ADARUN DBID=yyyyy,DEVICE=dddd,PROG=ADACNV,SVC=xxx
/*
ADACNV CONVERT TOVERS=vr
/*
/&
* $$ EOJ
```

# 29 ADADBS ユーティリティ：データベースサービス

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ADD：データセットの追加
- ALLOCATE：ファイルエクステントの割り当て
- CHANGE：フィールドの標準長の変更
- CVOLSER：指定ボリューム内の *Adabas* エクステント情報の出力
- DEALLOCATE：ファイルエクステントの割り当て解除
- DECREASE：アソシエータデータストレージの縮小
- DELCP：チェックポイントレコードの削除
- DELETE：ファイル削除
- DSREUSE：データストレージブロックの再利用
- ENCODEF：ファイルエンコードの変更
- INCREASE：アソシエータデータストレージの拡張
- ISNREUSE：ISN の再使用
- MODFCB：ファイルパラメータの変更
- MUPEX：MU フィールドおよび PE フィールドの最大数の設定
- NEWFIELD：新フィールドの追加
- ONLINVERT：オンラインインバート処理の開始
- ONLREORFASSO：ファイルのアソシエータのオンラインリオーダ開始
- ONLREORFDATA：ファイルデータのオンラインリオーダの開始
- ONLREORFILE：ファイルのアソシエータとデータのオンラインリオーダの開始
- OPERCOM：*Adabas* オペレータコマンド
- PRIORITY：ユーザープライオリティの変更

- **RECORDSPANNING**：レコードのスパニングの有効化／無効化
- **RECOVER**：スペースの回復
- **REFRESH**：ファイルを 0 件状態に設定
- **REFRESHSTATS**：統計値のリフレッシュ
- **RELEASE**：ディスクリプタの解除
- **RENAME**：ファイルデータベース名の変更
- **RENUMBER**：ファイル番号の変更
- **REPLICATION**：レプリケーションのアクティブ化または非アクティブ化
- **RESETDIB**：アクティブなユーティリティリストのエントリのリセット
- **RESETPPT**：PPT のリセット
- **SPANCOUNT**：スパンドレコード数のカウント
- **TRANSACTIONS**：トランザクションの中断および再開
- **UNCOUPLE**：ファイルのカップリング解除
- **JCL/JCS** の必要項目と指定例

# 30 機能概要

---

- TEST パラメータによる構文チェック ..... 206



**Note:** すべての ADADBS 機能は Adabas Online System (AOS) を使用して実行することもできます。Adabas Recovery Aid を使用する場合は、変更処理には AOS を使用することをお勧めします。これは、AOS ではリカバリ処理に必要なチェックポイントを書き込むためです。

1 回の ADADBS 実行中に、任意の数の機能を実行できます。

## TEST パラメータによる構文チェック

---

ADADBS 機能には構文チェック専用モードが組み込まれました。TEST パラメータを指定すると、実際の ADADBS 機能を実行せずにチェックのみが行われます。

ADADBS ユーティリティでは複数の機能を実行できます。このため、ADADBS では、次に指定された ADADBS 機能までパラメータが読み取られ、読み取られたばかりの機能/パラメータが実行されます。その後、ADADBS により、再び次の機能まで機能およびパラメータが読み取られます (以下同様)。したがって、TEST パラメータは、機能が実行されないことを保証するために、最初の機能/パラメータグループの前またはその中に指定する必要があります。以下に例を示します。

```
ADADBS TEST  
ADADBS DELETE FILE=1  
ADADBS DELETE FILE=2
```

# 31      ADD : データセットの追加

---

■ アソシエータまたはデータストレージデータセット .....	208
■ 基本パラメータとサブパラメータ .....	209
■ オプションパラメータ .....	209
■ 例 .....	209

ADD 機能では、新しいデータセットをアソシエータまたはデータストレージに追加します。

```
ADADBS ADD {ASSOSIZE = size [ ASSODEV = {device-type | ADARUN-device }]}|
          DATASIZE = size [ DATADEV = {device-type | ADARUN-device }]}
          [NOUSERABEND]
          [TEST]
```

このchapterでは、次のトピックについて説明します。

## アソシエータまたはデータストレージデータセット

---

アソシエータまたはデータストレージの場合、追加するデータセットは、現在使用中のデバイスタイプと同じデバイスタイプ上にある場合と、異なるデバイスタイプ上にある場合があります。これで、最大で 99 個の物理エクステントがアソシエータおよびデータストレージデータセットに設定されました。ただし、すべてのアソシエータ、データストレージ、およびデータストレージスペーステーブル (DSST) のエクステントのエクステント記述がジェネラルコントロールブロック (GCB) に収まる必要があるため、実際の最大数は、これよりも少なくなる可能性があります。例えば、標準の 3390 デバイスタイプでは、アソシエータ、データストレージ、および DSST の各エクステントの個数は 75 を超えることができます (他のタイプのエクステント数が少ない場合は、1つのタイプのエクステントをより多く持てます)。

 **Note:** アソシエータおよびデータストレージデータセットのサイズは別々に追加する必要があります。1回の操作で両方を追加することはできません。

アソシエータまたはデータストレージデータセットに対する ADD 操作が完了すると、ADD 機能によって現在のニュークリアスセッションが自動的に終了されます。これは、新しいセッションが開始する前に、必要なアソシエータまたはデータストレージを ADAFRM でフォーマットすることを可能にします。ニュークリアスが停止したことが、メッセージによって通知されます。

### 手順

▶ **手順31.1.アソシエータまたはデータストレージに新しいデータセットを追加するには、次の手順に従います。**

- 1 ADD 機能を実行します。
- 2 オペレーティングシステムでデータセットを割り当て、ADAFRM ユーティリティを使用して追加スペースをフォーマットします。
- 3 すべての Adabas ニュークリアスおよび Adabas ユーティリティ実行プロシージャに必要な JCL/JCS を追加します。

## 基本パラメータとサブパラメータ

### ASSODEV | DATADEV：デバイスタイプ

新しいデータセットに使用するデバイスタイプです。これらのパラメータは、ADARUN DEVICE パラメータで指定されたデバイスタイプと異なるデバイスタイプが使用される場合にのみ必要です。

VSAM データセットには、ダイナミックデバイスタイプである DDxxxxR1=9999、DDxxxxR2=8888、... DDxxxxR5=5555 を使用します。例えば、DDDATAR3 を追加する場合は、DATADEV=7777 です。

### ASSOSIZE | DATASIZE：追加するデータセットのサイズ

新しいデータセットに含めるシリンダの数です。

## オプションパラメータ

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定パラメータの構文だけがテストされ、値や変数の正当性はチェックされません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

3350 ディスクに 800 シリンダを含む新しいデータセットをデータストレージに追加します。

```
ADADBS ADD DATASIZE=800, DATADEV=3350
```

100 シリンダを含む新しいデータセットを、アソシエータの既存デバイスタイプでアソシエータに追加します。

```
ADADBS ADD ASSOSIZE=100
```



# 32 ALLOCATE：ファイルエクステンツの割り当て

---

■ 基本パラメータ .....	212
■ オプションパラメータ .....	212
■ 例 .....	284

ALLOCATE 機能を使用して以下を割り当てることができます。アドレスコンバータ、データストレージ、ノーマルまたは特定サイズのアップパーインデックスエクステント1回のADADBS実行で1つのエクステントのみを割り当てることができます。

```
ADADBS ALLOCATE FILE = file-number
                { ACSIZE | DSSIZE | NISIZE | UISIZE } = size
                [DEVICE = device-type | ADARUN-device ]
                [NOUSERABEND]
                [PASSWORD = 'password' ]
                [STARTRABN = start-rabn ]
                [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**FILE**：エクステントを割り当てるファイル

FILE にはエクステントを割り当てるファイルの数を指定します。

**ACSIZE | DSSIZE | NISIZE | UISIZE**：エクステントのタイプと大きさ

これらのパラメータを使用して、割り当てるエクステントのタイプとサイズを指定します。1回のADADBS ALLOCATE ステートメントではエクステントタイプとサイズを1つのみ指定できます。値は、シリンダ単位またはブロック単位で指定できます。サイズをブロックで指定する場合、数字の後に "B" を付加する必要があります (2000B など)。



**Note:** Adabas によって、MAXISN のデータベース設定を超過するアドレスコンバータのサイズ (ACSIZE) を割り当てようとしているかどうかを検出されます。この割り当てを実行しようとする、Adabas では、MAXISN までの ACSIZE のみが割り当てられます。

## オプションパラメータ

---

**DEVICE**：デバイスタイプ

ファイル割り当てに使用するデバイスタイプです。このパラメータは、ADARUN DEVICE パラメータで指定されたデバイスタイプと異なるデバイスタイプを使用する場合にのみ必要です。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルのパスワードを指定します。ファイルがパスワード保護されている場合、このパラメータは必須です。

**STARTRABN**：エクステントの開始 RABN

割り当てられるエクステントの開始 RABN です。このパラメータを省略すると、ADADBSによって開始 RABN が割り当てられます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

30 ブロックのアドレスコンバータエクステントをファイル 15 に割り当てます。

```
ADADBS ALLOCATE FILE=15,ACSIZE=30B
```



# 33 CHANGE：フィールドの標準長の変更

---

■ 基本パラメータ .....	216
■ オプションパラメータ .....	217
■ 例 .....	218

CHANGE 機能を使用して以下を変更できます。

- Adabas フィールドの標準長
- 通常の英数字 (A) フィールドからロング英字 (LA) フィールドへ、または
- デフォルトフィールドフォーマットをアンパック (U) からパック (P) へ

1 回の機能実行では、これらの変更の 1 つのみを実行できます。

この機能によって、データストレージ内のレコードへの修正は行われません。したがって、Adabas の定義に基づく新しい標準長とレコード内の実際のバイト数が違うため、ユーザーは、結果が正しくないフィールドを参照しないようにする必要があります。

Adabas 拡張ファイルのフィールド長を変更する場合は、拡張ファイルの個々のコンポーネントファイルに対して変更を実行する必要があります。コンポーネントファイルでの CHANGE オペレーションの実行ごとに変更を確認するメッセージが出力され、コンディションコード 4 が返されます。

```
ADADBS CHANGE FILE = file-number
                FIELD = 'field-name'
                { FORMAT=P | LENGTH=new-length | OPTION=LA }
                [ NOUSERABEND ]
                [ PASSWORD = 'password' ]
                [ TEST ]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：フィールドを含むファイル

長さを変更するフィールドを含むファイルです。Adabas システムファイルは指定できません。

### FIELD：変更するフィールド

標準長を変更するフィールドです。このフィールドとして、FIオプションで定義されたフィールド、または長さゼロが定義されたフィールド（可変長フィールド）は使用できません。フィールド名は単一引用符（'）で囲んで指定します。

### FORMAT=P：新しいフィールドフォーマット

新しい標準フィールドフォーマットです。フィールドフォーマットの変更は、アンパック (U) からパック (P) への変更でのみサポートされています。サブ/スーパー/ハイパーディスクリプタの親フィールドは変更できません。

FORMAT、LENGTH、OPTION のうちの 1 つを指定する必要があり、1 つだけ指定可能です。

**LENGTH**：新しいフィールド長

フィールドの新しい標準長です。長さゼロは指定できず、長さがゼロで定義された既存のフィールド（可変長フィールドなど）を標準長に再定義することもできません。

FORMAT、LENGTH、OPTION のうちの 1 つを指定する必要があり、1 つだけ指定可能です。

**OPTION=LA**：新しいフィールドオプション

新しいフィールドオプションです。フィールドオプションの変更は、通常の英数字（A）からロング英字（LA）への変更でのみサポートされています。

FORMAT、LENGTH、OPTION のうちの 1 つを指定する必要があり、1 つだけ指定可能です。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

変更するフィールドを含むファイルのパスワードです。ファイルがパスワード保護されている場合、このパラメータは必須です。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

ファイル 5 のフィールド AB の標準長を 11 バイトに変更します。

```
ADADBS CHANGE FILE=5,FIELD='AB',LENGTH=11
```

# 34 CVOLSER：指定されたディスクボリューム内の

## Adabas ボリュームの出力

---

■ 基本パラメータ .....	220
■ オプションパラメータ .....	220
■ 例 .....	221

CVOLSER 機能はディスクボリュームに含まれる Adabas ファイルエクステンツの出力に使用します。

```
ADADBS CVOLSER VOLSER = volume-serial-number
[NOUSERABEND]
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**VOLSER**：ボリュームシリアル番号

VOLSER は使用するディスクボリュームのボリュームシリアル番号です。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

ディスクボリューム DISK02 に含まれる Adabas ファイルエクステントを出力します。

```
ADADBS CVOLSER VOLSER=DISK02
```



# 35 DEALLOCATE : ファイルエクステンツの割り当て 解除

---

■ 基本パラメータ .....	224
■ オプションパラメータ .....	224
■ 例 .....	225

DEALLOCATE 機能を使用して次の割り当てを解除できます。アドレスコンバータ、データストレージ、ノーマルインデックスまたはアップーインデックスエクステント 1 回の ADADBS 実行で割り当てを解除できるのは 1 つのエクステントのみです。

```
ADADBS DEALLOCATE FILE = file-number
                   { ACSIZE | DSSIZE | NISIZE | UISIZE } = size
                   [NOUSERABEND]
                   [PASSWORD = 'password' ]
                   [STARTRABN = start-rabn ]
                   [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**ACSIZE | DSSIZE | NISIZE | UISIZE**：エクステントのタイプと大きさ

これらのパラメータには、割り当てを解除するエクステントのタイプとサイズを指定します。エクステントタイプとサイズは 1 つのみ指定できます。サイズは RABN ブロック数の後に "B" を付加したもの（DSSIZE=20B など）である必要があります、また、エクステントの最後の未使用 RABN の数を超過しないようにする必要があります。

**FILE**：エクステントの割り当てを解除するファイル

FILE には、割り当てを解除するエクステントを指定します。10 進値で指定します。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

スペースの割り当てを解除するファイルのパスワードです。ファイルがパスワード保護されている場合、このパラメータは必須です。パスワードは単一引用符でくくります。

**STARTRABN**：エクステントの開始 **RABN**

割り当て解除を実行するエクステントの最初のRABNです。このパラメータを省略すると、ファイルの最後のエクステントの割り当てが解除されます。アドレスコンバータでは、最後のエクステントのみが割り当て解除できます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

ファイル 15 に対して、30 ブロックのアドレスコンバータエクステントの割り当てを解除します。

```
ADADBS DEALLOCATE FILE=15,ACSIZE=30B
```



# 36      DECREASE : アソシエータデータストレージの縮小

---

■ 基本パラメータ .....	228
■ オプションパラメータ .....	228
■ 例 .....	229
■ 手順 .....	229

DECREASE 機能では、現在アソシエータまたはデータストレージに使用されている最終のデータセットのサイズを縮小します。解放するスペースは、フリースペーステーブル内で利用可能でなければなりません。

DECREASE 機能は、指定された物理エクステンツスペースをすべて割り当て解除するわけではありません。

```
ADADBS DECREASE { ASSOSIZE | DATASIZE } = sizeB
                 [NOUSERABEND]
                 [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### ASSOSIZE | DATASIZE：縮小するブロック

ASSOSIZE/DATASIZEには、アソシエータまたはデータストレージデータセットによって縮小されるブロックの数を指定します（10進値の後に"B"を付加します）。ASSOSIZEまたはDATASIZEのいずれかが指定できますが、両方は指定できません。ASSOSIZEとDATASIZEの両方を指定する場合は、個々に別のADADBS DECREASE ステートメントで入力する必要があります。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

アソシエータを 100 ブロック縮小し、データストレージを 200 ブロック縮小します。

```
ADADBS DECREASE ASSOSIZE=100B  
ADADBS DECREASE DATASIZE=200B
```

## 手順

---

▶手順 36.1. スペースの割り当てを解除するには、次の手順に従います。

- 1 DECREASE 機能でデータベースを縮小します。
- 2 ADASAV SAVE でデータベースを保存します。
- 3 ADAFRM でデータセットを再フォーマットします。
- 4 ADASAV でデータベースをリストアします。



# 37 DELCP：チェックポイントレコードの削除

---

■ 基本パラメータ .....	232
■ オプションパラメータ .....	232
■ 例 .....	233

DELCP 機能では、チェックポイントレコードを削除します。

ADADBS DELCP の実行後、残りのレコードには、新しい ISN が割り当てられます。この ISN には、チェックポイントレコードの削除時に使用された ISN も含まれます。より低い ISN が割り当てられますが、チェックポイントの時間的な順序は保持されます。

```
ADADBS DELCP TODATE = yyyymmdd  
[NOUSERABEND]  
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### TODATE：レコード削除の最終日付

TODATE では、チェックポイント情報を削除する最終日付を指定します。TODATE = で指定された日付以降の日付を持つチェックポイント情報は、削除されません。TODATE = は必須指定であり、デフォルトの日付は存在しません。日付は年（4桁）、月（2桁）、日（2桁）の順に 10 進値で指定します。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTESTパラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

1996年2月1日以前の全チェックポイントレコードを削除します。

```
ADADBS DELCP TODATE=19960201
```



# 38 DELETE：ファイルの削除

---

■ 基本パラメータ .....	236
■ オプションパラメータ .....	236
■ 例 .....	237

DELETE 機能はデータベースから Adabas ファイルを削除します。

```
ADADBS DELETE { FILE = fnr [KEEPFDT] [PASSWORD = 'password' ] }  
                [NOUSERABEND]  
                [TEST]
```

Adabas ファイルがデータベースから削除されると、ファイルに割り当てられたすべての論理エクステントの割り当てが解除されます。解放されたスペースは、新しいファイルまたは既存ファイルの新しいエクステントに使用できます。

削除するファイルはカップリングされていない必要があります。Adabas 拡張ファイルを指定すると、拡張ファイル全体（アンカーおよびコンポーネントファイル）が削除されます。

DELETE 機能が完了するとき、オペレータコマンド LOCKU または LOCKF で設定されたロックはリセットされます。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：削除するファイル

FILE には、削除する Adabas ファイルの番号を指定します。ADALOD コーティリティの SYSDFILE オプションを使用してロードされたチェックポイント、セキュリティ、トリガ、およびその他のあらゆるファイルは、ADADBS DELETE のみが Adabas ユーザーである場合にのみ指定できます。これらのファイルを削除することによって、削除終了時に Adabas が自動的に終了します。Adabas 拡張ファイルを削除するには、ファイル番号（およびアンカーファイル）を指定します。

## オプションパラメータ

---

### KEEPFDT：フィールド定義テーブルの保持

ADADBSDELETE では、KEEPFDT パラメータが指定されている場合、後で ADACMP で使用できるように、削除されたファイルのフィールド定義テーブル（FDT）を保持します。このパラメータが指定されると、削除中のファイルと同じ番号のファイルは、新しいファイルの FDT が削除されたファイルの FDT と同じか、またはロードオペレーションによって IGNFDT パラメータが新しいファイルの FDT を受け入れるように指定のされた場合は、後からロードすることのみが可能です。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

PASSWORDでは、削除するファイルのパスワードを指定します。ファイルがパスワード保護されている場合、このパラメータは必須です。

**TEST**：構文テスト

TESTパラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTESTパラメータを使用する詳細については、「[TESTパラメータによる構文チェック](#)」の項を参照してください。

## 例

ファイル 6 を削除します。

```
ADADBS DELETE FILE=6
```

パスワード保護ファイル10を削除します。フィールド定義テーブルは保持されます。ファイル番号 10 は、別の ADALOD LOAD コマンドが IGNFDT オプションを使用して発行されるまでは、再び使用できません。

```
ADADBS DELETE  
FILE=10,KEEPFDT,PASSWORD='FILE10'
```



# 39 DSREUSE : データストレージブロックの再使用

---

■ 基本パラメータ .....	295
■ オプションパラメータ .....	240
■ 例 .....	241

DSREUSE 機能で、データストレージブロックの割り当てを制御します。

```
ADADBS DSREUSE FILE = file-number
                MODE = { ON | OFF }
                [NOUSERABEND]
                [PASSWORD = 'password' ]
                [RESET]
                [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILE は DSREUSE 設定を適用するファイルの番号です。

ブロックの再使用は、当初、ADALOD FILE 機能でデータベースにファイルをロードするとき、または ADADEF DEFINE 機能でシステムファイルを定義するときに決定されます。いずれの場合も、これらの機能で特に指定されない限り、ブロック再使用のデフォルトは "YES" になります。

### MODE：再使用モード

有効にするデータストレージブロック割り当てモードです。MODE=OFF の場合、ADADBS DSREUSE 機能がキャンセルされ、レコード削除の結果解放されるデータストレージブロックが再使用できなくなります。MODE=ON の場合、データストレージブロックは再使用できます。MODE= パラメータにはデフォルトはなく、指定する必要があります。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

PASSWORD は、ファイルのセキュリティパスワードを指定しますが、ファイルがパスワード保護されている場合、このパラメータは必須です。

**RESET**：スペースポインタリセット

RESET パラメータを指定すると、ファイルの先頭から新しいデータストレージスペースが検索されます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

ファイル 6 のデータストレージブロックは再使用しません。

```
ADADBS DSREUSE FILE=6,MODE=OFF
```



# 40 ENCODEF：ファイルエンコードの変更

---

■ 基本パラメータ .....	244
■ オプションパラメータ .....	245
■ 例 .....	245

```
ADADBS ENCODER FILE = file-number
          FACODE = alpha-key | UWCODER = wide-key
          [NOUSERABEND]
          [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FILE はエンコードを変更するファイル番号です。

### FACODE：ファイル内の英数字フィールドのエンコード

FACODE パラメータでは、ファイルに保存される英数字フィールドのエンコードを定義します。ロード済みのファイルに適用できます。英数字のエンコードは EBCDIC に従わなければなりません（例えばスペース文字は 16 進数 X'40'）。DBCS オンリーの場合を除いて、2 バイト文字セット（DBCS）タイプのエンコードがサポートされます。提供されるコードページについては、「提供される UES エンコード」を参照してください。

FACODE および UWCODER は、相互排他的なパラメータであり、一方を指定した場合は、他方を指定しないでください。ただし、いずれか 1 つは指定する必要があります。

### UWCODER：ファイル内のワイド文字フィールドのユーザーエンコード

UWCODER パラメータでは、ファイルに保存されるワイド文字フィールドのユーザーエンコードを定義します。ロード済みのファイルに適用できます。ワイドファイルエンコードが変更されるわけではありません。

ワイド文字フィールドのエンコードを変更するには、ファイルをアンロードし、圧縮解除してから圧縮して、再ロードする必要があります。提供されるコードページについては、「提供される UES エンコード」を参照してください。

FACODE および UWCODER は、相互排他的なパラメータであり、一方を指定した場合は、他方を指定しないでください。ただし、いずれか 1 つは指定する必要があります。

## オプションパラメータ

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

次の例では、ADADBS ENCODEF を使用して、ファイル 1425 の英数字フィールドのエンコードをユーザーコードページ 285（CECP：英国、充填文字 X'40' との互換 EBCDIC）に変更します。ファイル 401 では、ADADBS ENCODEF を使用してワイドフィールドのエンコードをユーザーコードページ 3396（IBM、CCSID 4396、1880 ユーザー定義文字を含む日本語ホスト 2 バイト文字）を変更します。UWCODE が変更されるため、ファイル 401 をアンロードし、圧縮解除してから圧縮して、再ロードする必要があることに注意してください。

```
ADADBS ENCODEF FILE=1425,FACODE=285
```

```
ADADBS ENCODEF FILE=401,UWCODE=3396
```



# 41 INCREASE : アソシエータデータストレージの増加

---

■ 基本パラメータ .....	248
■ オプションパラメータ .....	248
■ 例 .....	249
■ 一般手順 .....	249
■ オペレーティングシステム固有の手順 .....	249

INCREASE 機能は、現在アソシエータまたはデータストレージに使用されている最終のデータセットのサイズを拡張します。この機能は、アソシエータに対しては何回でも実行できます。データストレージスペーステーブル (DSST) の最大数が 99 であることによって、データストレージの増加がある程度制限されます。この最大数に達した場合は、ADAORD ユーティリティの REORASSO または REORDB 機能を使用して 99 のデータストレージエクステントのすべてを組み合わせる単一のエクステントにする必要があります。

### Notes:

1. アソシエータおよびデータストレージデータセットのサイズは別々に拡張する必要があります。1 回の操作で両方を拡張することはできません。
2. INCREASE 操作が完了すると、INCREASE 機能によって、現在のニュークリアセッションが自動的に終了されます。これは、新しいセッションを開始する前に、必要なアソシエータまたはデータストレージを ADAFRM でフォーマットすることを可能にします。ニュークリアが停止されたことを示す情報メッセージが出力されます。

```
ADADBS INCREASE { ASSOSIZE | DATASIZE } = size  
[ NOUSERABEND ]  
[ TEST ]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### ASSOSIZE | DATASIZE：増加するサイズ

アソシエータまたはデータストレージデータセットによって必要とされるブロックまたはシリンドラの追加数です。ブロックを指定する場合は、値の後に "B" (DATASIZE=50B など) を付加します。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エ

ラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

#### TEST：構文テスト

TEST パラメータを使用すると、実際に操作を実行せずに構文をテストできます。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。このパラメータの使用に関する詳細については、「[TEST パラメータによる構文チェック](#)」を参照してください。

## 例

アソシエータを 400 シリンダ増加します。

```
ADADBS INCREASE ASSOSIZE=400
```

## 一般手順

▶手順41.1.アソシエータまたはデータストレージのサイズを増加する一般的な手順は、次のとおりです。

- 1 ADASAV ユーティリティを使用してデータベースをバックアップします。このステップは任意ですが、実行することをお勧めします。
- 2 ADADBS INCREASE 機能を実行します。
- 3 ADAFRM ユーティリティで、データセットに追加されている新しいスペースをフォーマットします。

## オペレーティングシステム固有の手順

### z/OS システム

z/OS では、JCL で DISP=MOD パラメータを指定することで同じデータセットをフォーマットできます。増加しているデータセットの SPACE パラメータは次のように指定します。

```
SPACE=(CYL,(0,n))
```

ここでは、 $n$  が追加するスペースの量（シリンダ数）です。また、ADAFRM コントロールステートメントでは、追加するシリンダ数も指定します。フォーマットするデータセットの増加部分が新しいボリュームに含まれる場合、JCL の VOL パラメータには、データセットを含む全ボリュームへの参照が含まれている必要があります。

### 例 1：OS 単一ボリューム INCREASE

現在 300 シリンダを含むアソシエータデータセットに 400 シリンダを追加します。INCREASE 機能のコントロールステートメントは次のとおりです。

```
ADADBS INCREASE ASSOSIZE=400
```

次の JCL 例では、ADAFRM を使用してアソシエータデータセットを増加します。

```
//DDASSOR1 DD  
DSN=... ,DISP=MOD,SPACE=(CYL,(0,400))
```

実際の ADAFRM コントロールステートメントは次のとおりです。

```
ADAFRM ASSOFRM SIZE=400
```

### 例 2：OS 複数ボリューム INCREASE

例 1 での増加を複数のボリュームに対して行う場合は、JCS でボリュームを指定します。

```
//DDASSOR1 DD DSN=...  
//  
DISP=(MOD,CATLG),VOL=SER=(V1,V2,...),SPACE=(CYL,(0,400))...
```

INCREASE ステップの後で、FORMAT ステップの前に、次のステップを組み込んで正しいカタログエントリを実行します。

```
//UNCATLG EXEC PGM=IEFBR14  
//DDASSOR1 DD DSN=... ,DISP=(SHR,UNCATLG)
```

## VSE システム

▶ **手順 41.2. アソシエータまたはデータストレージの増加には、次の手順をお勧めします。**

- 1 現データベースを保存します。
- 2 ADAEND オペレータコマンドで Adabas セッションを通常終了させます。
- 3 同じボリュームに新しいエクステントを追加するために、データベースを定義する JCS を更新します。

同一または異なる VSE ボリューム上の新しいアソシエータまたはデータエクステントを ADADBS INCREASE で増加し、ADAFRM でフォーマットする前に、そのボリュームの目次 (VTOC) を更新して新しいエクステントを加える必要があります。

1つのボリュームエクステンツの場合は、次の例のようなジョブを使用して VTOC を更新してください。

```
* $$ JOB JNM=jobname
* $$ LST ...
* $$ PCH ...
// ASSGN SYS001,DISK,VOL=volume,SHR
// DLBL ASSOEXT,'dsname',99/365,DA
// EXTENT SYS001,volumel,1,0,starttrack1,trackcount1
// EXTENT SYS001,volumel,1,1,starttrack2,trackcount2

// EXEC ASSEMBLY,GO
MODVTOC CSECT
        BALR 9,0
        BCTR 9,0
        BCTR 9,0
        USING MODVTOC,9
        OPEN ASSOEXT
        CLOSE ASSOEXT
        EOJ RC=0
ASSOEXT DTFPH TYPEFLE=OUTPUT,DEVADDR=SYS001,DEVICE=DISK,MOUNTED=ALL
        END

/*
/&
* $$ EOJ
```

2つのボリュームエクステンツの場合は、次の例のようなジョブを使用してください。

```
* $$ JOB JNM=jobname
* $$ LST ...
* $$ PCH ...
// ASSGN SYS001,DISK,VOL=volume1,SHR
// ASSGN SYS002,DISK,VOL=volume2,SHR
// DLBL ASSOEXT,'dsname',99/365,DA
// EXTENT SYS001,volumel,1,0,starttrack1,trackcount1
// EXTENT SYS002,volume2,1,1,starttrack2,trackcount2
// EXEC ASSEMBLY,GO
MODVTOC CSECT
        BALR 9,0
        BCTR 9,0
        BCTR 9,0
        USING MODVTOC,9
        OPEN ASSOEXT
        CLOSE ASSOEXT
        EOJ RC=0
ASSOEXT DTFPH TYPEFLE=OUTPUT,DEVADDR=SYS001,DEVICE=DISK,MOUNTED=ALL
        END

/*
```

```
/&
* $$ EOJ
```

 **Note:** このジョブによって VSE エラーメッセージ 4733D がコンソールに送られ、オペレータに対するレスポンスが要求されます。JCS がチェックされると、オペレータレスポンスは DELETE になっている必要があります。

- 4 ADADBS INCREASE 処理を実行します。
- 5 新エクステントをフォーマットするために、新しい ADAFRM ジョブを実行します。次の例で示されているように、ADAFRM ジョブでは FROMRABN パラメータを指定する必要があります。

```
ADAFRM ASSOFRM SIZE=size, FROMRABN=rabn-number
```

ここでは、*size* がデータセットを増加するシリンダまたはブロックの数で、*rabn-number* が新しいエクステントにおける最初の RABN です。

- 6 Adabas ニュークリアスを開始します。

 **Note:** VM 環境では、マルチボリューム、マルチエクステントファイルに対して特定の制約が適用されます。これらの制約が侵害されると、VSE エラー 4n83I（無効な論理ユニット）が発生します。これらの制約の詳細については、適切な IBM マニュアルを参照してください。

## z/VM システム

z/VM には、データベースを増加させるための 2 つの手順があります。1 つの手順では ADAMAINT および INCREASE EXEC を使用し、もう 1 つの手順ではマニュアル操作を順次行います。

### ▶手順 41.3. EXEC プロシージャ

- 1 ユーザーの CMS 環境を修正するために ADAMAINT EXEC をコールします。

```
ADFnnnnn EXEC, DBnnnnn VOLUMES volume=vol-id, ...
```

ADAMAINT により、既存の ASSO/DATA/WORKR<sub>x</sub> へのミニディスクの追加、または新しい ASSO/DATA/WORKR<sub>x</sub> の定義が必要となります。

- 2 INCREASE EXEC をコールします。この EXEC では、自動的に LINK、ADADBS ADD または INCREASE が（ADAMAINT でのユーザー指定に従って）実行され、次に ADAFRM を使用して新しいエリアがフォーマットされます。

## ▶手順 41.4. マニュアル操作

- 1 必要とするサイズよりも1シリンダ（または疑似シリンダ）大きい新規ミニディスクを定義します。
- 2 FORMAT コマンドを発行します。

```
FORMATcuu T nnn
```

ここでは、*cuu* が新しいミニディスクの仮想ユニットアドレスであり、*nnn* は CKD デバイスに対しては "1"、FBA デバイスに対しては "20" です。ボリュームラベルが要求された場合には、6 文字までの英数字でユニークな名前を指定する必要があります。

- 3 次のコマンドでミニディスクを予約します。

```
RESERVE file-name file-type T
```

ここでは、*file-name* および *file-type* がプライマリミニディスク上のファイルに使用されるファイル名およびファイルタイプと一致する必要があります。

- 4 このドキュメントで説明されている手順に従って ADADBS INCREASE を実行します。
- 5 ADAEND で Adabas セッションを終了させます。
- 6 ADAREP レポートを作成し、新規エクステント内の最初の RABN を検索します。これはデータベースの物理レイアウト部分に配置されている場合があります。このエクステント上の RABN 範囲は VOLSER NUMBER xxxxxxx で示されます。
- 7 必要に応じて、ディレクトリまたは PROFILE EXEC に新規ミニディスク用の CP LINK ステートメントを追加します。このデータベースにマルチライトアクセスを行う他のすべての仮想マシン（DBA マシンなど）に対して、PROFILE EXEC または CP ディレクトリエントリを更新します。
- 8 更新が必要な EXEC に対しては、ファイルに対する DATADEF ステートメントを修正します。Software AG の標準の EXEC が使用されている場合、これらの DATADEF ステートメントは ADFnnnnn EXEC にあり、nnnnn は 5 桁のデータベース ID を表します。

DATADEF ステートメントを変更するには、次の行を配置します。

```
volx = vol-id
```

*volx* がアソシエータの場合は "a"、DATA の場合は "d" である場合、*vol-id* は前のボリュームリストです。この行を変更します。

```
volx =(vol-id,vo-label)
```

ここでは、*vo-label* は、ステップ 2 で FORMAT コマンドの入力中に指定されたボリュームラベルです。

- 9 ファイルに対して次のように ADAFRM ユーティリティを実行します。

```
ADAFRM xxxxFRM SIZE=size, FROMRABN=rabn-number
```

ここでは次の内容を表しています。

<i>xxxx</i>	ASSO または DATA のいずれか
<i>size</i>	ミニディスクから 1 シリンダ (または疑似シリンダ) を引いたサイズ
<i>rabn-number</i>	ステップ 6 で作成されたレポートでの表示のとおり新しいエクステントにおける最初の RABN

## BS2000 システム

▶手順 41.5. 次の手順を使用して BS2000 システムでデータベースを増加します。

- 1 セクション「[一般手順](#)」の説明に従って ADADBS INCREASE を実行します。
- 2 ADAEND で Adabas セッションを終了させます。
- 3 ADAREP ユーティリティを実行してデータベースレポートを生成します。レポートを使用して、レポートの "Physical Layout of the Database" 部分で新しいエクステントの最初の RABN を検索します。RABN 範囲は "VOLSER NUMBER" 列に示されます。
- 4 BS2000 MODIFY-FILE-ATTRIBUTE コマンドでデータセットを増加します。例としては、次のようなものがあります。

```
/MODIFY-FILE-ATTRIBUTE ADA99.ASSO,PUB(SPACE=REL(400))
```

 **Note:** 古い ISP フォーマットでは、FILE コマンド、例えば、 /FILE ADA99.ASSO, SPACE=400 で実行しました。

- 5 ADAFRM ユーティリティを実行して新しいスペースをフォーマットします。ステップ 4 で追加されるスペースの例：

```
ADAFRM ASSOFRM SIZE=400B, FROMRABN=rabn-number
```

*rabn-number* で新しいエクステントの最初の RABN がレポートでの表示通りに指定されています。

## 42 ISNREUSE : ISN の再使用

---

■ 基本パラメータ .....	256
■ オプションパラメータ .....	256
■ 例 .....	257

ISNREUSE 機能では、削除したレコードの ISN を新しいレコードに再割り当てできるかどうかを制御します。

```
ADADBS ISNREUSE FILE = file-number
                MODE = { ON | OFF }
                [NOUSERABEND]
                [PASSWORD = 'password' ]
                [RESET]
                [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILE には、ISNREUSE 設定を変更するファイル番号を指定します。チェックポイントファイルは指定できません。

### MODE：再使用モード

MODE で ISN 再使用モードを有効にできます。MODE=OFF の場合、Adabas では、削除されたレコードレコードの ISN が新しいレコードに再使用されません。新レコードにはそれぞれ次に大きい未使用 ISN が割り当てられます。MODE=ON の場合、Adabas では、削除レコードの ISN を再使用できます。MODE パラメータにはデフォルトはなく、指定する必要があります。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### PASSWORD：ファイルパスワード

PASSWORD は、ファイルのセキュリティパスワードを指定しますが、ファイルがパスワード保護されている場合、このパラメータは必須です。

**RESET : ISN** ポインタのリセット

RESET パラメータでは、ファイルの開始において未使用の ISN が検索されます。

**TEST** : 構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定パラメータの構文だけがテストされ、値や変数の正当性はチェックされません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

ファイル 7 内の削除レコードの ISN は新しいレコードに割り当てることができます。

```
ADADBS ISNREUSE FILE=7,MODE=ON
```



# 43      MODFCB：ファイルパラメータの修正

---

■ 基本パラメータ .....	260
■ オプションパラメータ .....	260
■ 例 .....	262

MODFCB 機能で、Adabas 非システムファイルの各種のパラメータを修正します。

```
ADADBS MODFCB FILE = file-number
[ASSOPFAC = new-padding-factor ]
[DATAPFAC = new-padding-factor ]
[LOBFILE = file-number ]
[MAXDS = maximum-secondary-allocation ]
[MAXNI = maximum-secondary-allocation ]
[MAXUI = maximum-secondary-allocation ]
[MAXRECL = maximum-compressed-record-length ]
[NOUSERABEND]
[PASSWORD = 'password' ]
[PGMREFRESH = { YES | NO } ]
[RPLUPDATEONLY = { YES | NO } ]
[TEST]
```



**Note:** LOBFILE パラメータが指定された場合、同じ実行で FILE パラメータ（必須）と、任意の PASSWORD および TEST パラメータのみを指定できます。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILE は修正する Adabas ファイル番号です。Adabas システムファイル番号は指定できません。ファイルにラージオブジェクト（LB）フィールドがある場合は、これが、基本ファイルのファイル番号です。

## オプションパラメータ

---

### ASSOPFAC/DATAPFAC：ファイルパディングファクタ

ASSOPFAC/DATAPFAC では、アソシエータおよびデータストレージに対して有効とするパディングファクタ（1～90）をそれぞれ指定します。既存ブロックでは元のパディングファクタが保持されます（「ADAORD ユーティリティ」参照）。

スパンドレコードを含むファイルについては、データストレージパディングファクタ (DATAPFAC) 設定は無視されます。データストレージのスパンが有効になっているファイルに DATAPFAC が指定されると、次の警告が表示されます。

```
*****DATAPFAC is ignored for files with spanned data storage enabled*****
```

リターンコード 4 (CC=4) も返されます。

#### LOBFILE：LOB ファイル番号

LOBFILE では、LOB ファイル (基本ファイルに関連付けられた LOB ファイル) の番号を指定します。このパラメータは (基本ファイル番号に設定された) FILE パラメータと組み合わせて、2つのファイルを同期しているものとして扱う場合に役に立ちます。

 **Note:** LOBFILE パラメータが指定された場合、同じ実行で FILE パラメータ (必須) と、任意の PASSWORD および TEST パラメータのみを指定できます。

詳細については、「ラージオブジェクト (LB) フィールドの基本」を参照してください。

#### MAXDS/MAXNI/MAXUI：最大 2 次割り当て

データストレージ (MAXDS)、ノーマルインデックス (MAXNI)、およびアッパーインデックス (MAXUI) に対する 2 次エクステントごとの最大ブロック数です。

指定値は後に "B" を付加してブロックが指定されている必要があります (MAXDS=8000B など)、65535B を超過する数は使用できません。

パラメータの 1 つが指定されていないか、または "0B" が指定されている場合、コンポーネントに対する最大 2 次エクステントの割り当ては制限されません。

いかなる場合も、Adabas では、これらのパラメータに対して最小 2 次割り当てが行われません。

MAXDS=6B MAXNI=6B MAXUI=15B

これらの最小割り当てよりも低い値を指定すると、最小値が使用されます。

#### MAXRECL：最大圧縮レコード長

ファイルに許可する最大圧縮レコード長です。指定値には指定されたファイル内の現在の最大レコードサイズよりも小さい値は使用できません。

#### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### **PASSWORD**：ファイルパスワード

このパラメータは、FILEパラメータで指定されたファイルのパスワードが保護されている場合に必要です。

### **PGMREFRESH**：プログラムによるファイルのリフレッシュ

PGMREFRESH オプションでは、特殊 E1 コマンドの発行によって、ユーザープログラムでファイルリフレッシュ機能を実行できるようにするかどうかを設定します。パラメータが指定されない場合、オプションでは現在のステータスであるオン (YES) またはオフ (NO) のいずれかの状態が保持されます。

### **RPLUPDATEONLY**

RPLUPDATEONLY パラメータを ADADBS MODFCB 機能で使用し、Adabas データベースファイルを、Adabas-to-Adabas レプリケーションの一部として、または他の手段も使用して Event Replicator Server によってのみ更新できるようにするかどうかを指定します。このパラメータは任意指定です。有効な値は "YES" または "NO" です。"YES" の値はファイルが Event Replicator を介してのみ処理できることを、NO の値はファイルが Event Replicator 処理を含む通常的手段で更新できることを示します。ADADBSMODFCB 機能で RPLUPDATEONLY パラメータに指定された値がない場合はデフォルト値はなく、ファイルに以前使用された値が使用されます。

### **TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTESTパラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

ファイル203に対して次のような修正を行います。アソシエータパディングファクタは5に設定します。データストレージパディングファクタは5に設定し、最大データストレージ2次エクステンションは100ブロックに設定します。

```
ADADBS MODFCB
FILE=203,ASSOPFAC=5,DATAPFAC=5,MAXDS=100B
```

# 44 MUPEX : MU フィールドおよび PE フィールドの最大数の設定

---

- 構文 ..... 265
- 基本パラメータ ..... 265
- オプションパラメータ ..... 265
- 例 ..... 266

MUPEX 機能はファイルの MU フィールドまたは PE グループの指定に使用します。この機能はディスクリプタで PE フィールドを使用するファイルで実行できないか、またはシステムファイルで実行できません。また、データストレージでの圧縮レコードの最大サイズによって、オカレンスの最大数も制限されます。

ディスクリプタに PE グループを含むファイルに対する MU/PE カウントを増加させるには、PE グループに関係するあらゆるディスクリプタを（ADADBS RELEASE 機能を使用して）最初に解放する必要があります。ディスクリプタが解放されると、MUPEX 機能を使用して MUPECOUNT パラメータを "2" に設定でき、（ADADBS INVERT 機能を使用して）ディスクリプタが再インポートされます。

このchapterでは、次のトピックについて説明します。

## 構文

```
ADADBS MUPEX FILE = file-number
MUPECOUNT = { 1 | 2 }
[PASSWORD = 'password' ]
[NOUSERABEND]
[TEST]
```

## 基本パラメータ

### FILE：ファイル番号

FILE では、オカレンスの最大数を設定するファイルを指定します。MU および PE の制限を設定するデータベース中のファイルの有効なファイル番号を指定します。

システムファイルは変更できません。

### MUPECOUNT：カウントフィールドのサイズ

MUPECOUNT では、ファイルに許容される MU および PE オカレンスの最大数を設定します。有効な値は、"1" または "2" です。

"1" の指定は、ファイルで発生する MU フィールドおよび PE グループの数が 191 を超過しないことを示します。"2" の指定は、ファイルに最大 65,534 の MU フィールドおよび PE グループが含まれることを示します。

MUPECOUNT パラメータに "1" を設定するには、ファイルが空であるか、PE フィールドが設定されていない必要があります。MUPECOUNT を "2" に設定するには、ファイルが空でない場合は、DE フィールドの一部である PE フィールドをファイルに含めることができません。

## オプションパラメータ

### NOUSERABEND：ABEND なし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザー ABEND 34（ダンプあり）またはユーザー ABEND 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力しても ABEND しません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エ

ラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### **PASSWORD**：ファイルパスワード

MU および PE フィールドカウントを制限するファイルのパスワードです。ファイルがパスワード保護されている場合、このパラメータは必須です。パスワードは単一引用符でくくります。また、この機能が正しく動作するには、パスワードによって更新権限が与えられる必要があります。

### **TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。Adabas の機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

次の例で、ファイル 17 に保管できる MU および PE フィールド要素の最大数は 65,534 です。

```
ADADBS MUPEX FILE=17,MUPECOUNT=2
```

# 45 NEWFIELD：新しいフィールドの追加

---

■ 基本パラメータ .....	268
■ オプションパラメータ .....	269
■ 例 .....	270

NEWFIELD 機能では、ファイルに1つまたは複数のフィールドを追加します。新フィールド定義がフィールド定義テーブル（FDT）の末尾に追加されます。



**Note:** ディスクリプタフィールドの定義はレコード構造から独立していますが、ディスクリプタフィールドがレコードの最初に配列されず、論理的に物理レコードの終了を超えている場合、パフォーマンスの理由から、そのレコードに対するインバーテッドリストエントリが生成されないことに注意してください。このケースでインバーテッドリストエントリを生成するには、ファイルのアンロード（SHORT モードで）、圧縮解除、および再ロードが必要です。あるいは、アプリケーションプログラムを使用してフィールドをファイルの各レコードの最初に再配列します。

NEWFIELD では、実際のデータストレージデータを新しいフィールドに指定できません。データは、後から Adabas 追加／更新または Natural コマンドを使用して指定できます。

Adabas 拡張ファイルにフィールドを追加するときは、個別コンポーネントファイルそれぞれにフィールドを追加する必要があります。コンポーネントファイルでの NEWFIELD オペレーションの実行ごとに変更を確認するメッセージおよびコンディションコード 4 が返されます。

```
ADADBS NEWFIELD FILE = file-number
      [FNDEF = 'Adabas-field-definition' ]
      [NOUSERABEND]
      [PASSWORD = 'password' ]
      [SUBFN = 'name = parent-field (begin , end)' ]
      [SUPFN = 'name = {parent-field (begin , end) } ...' ]
      [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILEでは、フィールドを追加するファイル番号を指定します。ファイルとして、Adabas システムファイルは使用できません。

## オプションパラメータ

### FNDEF：Adabas フィールド定義

FNDEF では、Adabas フィールド（データ）定義を指定します。追加の各フィールドごとに1つのFNDEF ステートメントが必要です。フィールド定義エントリを構成するのに使用する構文は次のとおりです。

```
FNDEF = 'level, name [ , length, format ] [, MU [(occurrences)] ] [, option ] ... '
```

```
FNDEF = 'level, name [ , PE [(occurrences)] ]'
```

各定義は、ADACMP ドキュメントで、ADACMP ユーティリティに関して説明されているフィールド定義構文に従っている必要があります（「[FNDEF：フィールド定義およびグループ定義](#)」と「[FNDEF：ピリオディックグループ定義](#)」の「フィールド定義ステートメント」セクション）。

次の制約に注意してください。

- サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、またはフォネティックディスクリプタは指定できません。
- 文字列の記述やシーケンス番号は指定できません。
- MU または PE フィールドを追加するときにオカレンス番号を指定しても無視されます。

最初のFNDEF ステートメントにレベル番号を設定するときは、次のルールが適用されます。

1. レベル番号 01 は常に許可されます。
2. レベル番号 02 以降は、このフィールドの既存グループへの追加を意味します。したがって、次のルールが適用されます。
  - 通常のグループ（ピリオディックグループ以外）であれば、フィールドを追加できます。
  - グループがPEグループである場合、ファイルのファイルコントロールブロック（FCB）がない場合、すなわち、ファイルがKEEPFDT オプションで削除されているか、または、Adabas Online System FDT 定義機能を使用してFDTが定義されているが、ファイル定義機能が実行されていない場合のいずれかに限って、ファイルを追加できます。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ

ラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルパスワード ファイルがパスワード保護されている場合、このパラメータは必須です。

**SUBFN/ SUPFN**：サブフィールドまたはスーパーフィールドの追加

これらのパラメータは、サブフィールドやスーパーフィールドの追加に使用できます。各定義は、ADACMP ユーティリティで説明されたサブフィールド/スーパーフィールドの定義構文に従っている必要があります。詳細については、「[COMPRESS機能：Adabas ファイルの圧縮](#)」を参照してください。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

ファイル 24 にグループ AB (フィールド AC および AX で構成) を追加します。

```
ADADBS NEWFIELD FILE=24
ADADBS FNDEF='01,AB'
ADADBS FNDEF='02,AC,3,A,DE,NU'
ADADBS FNDEF='02,AX,5,P,NU'
```

# 46 ONLINVERT：オンラインインバートの開始

---

- 基本パラメータ ..... 272
- オプションパラメータ ..... 273
- 例 ..... 274

ONLINVERT 機能では、オンラインインポート処理を開始します。

```
ADADBS ONLINVERT FILE = file-number
      { FIELD = 'field-name [ , option, ... ]' |
        SUPDE = ' name [ , UQ [ , XI ] ] = parent-field (begin, end), ...' |
        SUBDE = ' name [ , UQ [ , XI ] ] = parent-field (begin, end)' |
        PHONDE = ' phonde-name (parent-field)' |
        HYPDE = 'nr, name, length, format [ , options] = parent-field, ...' |
        COLDE = 'nr, name [ , UQ [ , XI ] ] = parent-field' }
      [CODE = cipher-code ]
      [PASSWORD = 'password' ]
      [NOUSERABEND]
      [TEST]
      [WAIT]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

ファイルは、新しいディスクリプタを作成するファイル番号です。拡張ファイルチェーンのコンポーネントファイルを指定すると、そのチェーンの全コンポーネントファイルにディスクリプタが追加されます。

### FIELD/ SUBDE/ SUPDE/ PHONDE/ HYPDE/ COLDE：ディスクリプタ定義

インポートするディスクリプタのタイプ定義には、これらのパラメータを1つだけ指定します。オンラインインポート機能を使用すると、ファイルごとに1つのディスクリプタを1回にインポートできます。

FIELDパラメータを使用してフィールドをディスクリプタとして定義します。照合ディスクリプタにはCOLDEを、ハイパーディスクリプタにはHYPDEを、フォネティックディスクリプタにはPHONDEを、サブディスクリプタにはSUBDEを、スーパーディスクリプタにはSUPERDEを使用します。

FIELDでは、インポートする既存のフィールドを指定します。フィールドはエレメンタリフィールドでもマルチプルバリューフィールドでもよく、また（フィールドがFIオプションで定義されていない限り）ピリオディックグループ内のものでも構いません。

ディスクリプタフィールドをユニーク指定したい場合は、フィールド名に続けて"UQ"を指定します。ピリオディックグループに属するフィールドはユニークディスクリプタとして定義できません。ディスクリプタの一意性を、排除されるインデックス（オカレンス番号）で決定する必要がある場合は、同様に、"XI"も指定してください。

サブフィールドまたはスーパーフィールドをインバートする場合は、フィールドの作成時と同じ親フィールドを SUBDE または SUPDE で指定しないとエラーになります。開始および終了の値は元のフィールド定義から取得されます。

親フィールドを NU オプションで指定すると、空値フィールドを含むレコードについてはインバーテッドリストにエントリが作成されません。スーパーディスクリプタとハイパーディスクリプタは、他のディスクリプタエレメントの値に関係なく、エントリが作成されません。

親フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

各ディスクリプタの構文、サブパラメータ、およびコーディングについての詳細は、「[ADACMP ユーティリティ](#)」のディスクリプションを参照してください。

## オプションパラメータ

---

### CODE：サイファコード

FILE パラメータで指定したファイルが暗号化されている場合、サイファコードを CODE パラメータで指定しなければなりません。

### PASSWORD：ファイルパスワード

FILE で指定されたファイルのパスワードがセキュリティ保護されている場合、PASSWORD パラメータを使用してファイルのパスワードを指定する必要があります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### WAIT：プロセス終了待機

このオンライン処理が終わるまで ADADBS が次の機能や終了の処理を行わない場合に、WAIT を指定します。

WAIT を指定しないと、ADADBS はオンライン処理を開始したらただちに次の処理を実行します。

## 例

---

この処理の終了を待たずに、ファイル 10 のフィールド AA をディスクリプタにするオンライン処理を開始します。

```
ADADBS ONLINVERT FILE=10,FIELD=AA
```

# 47 ONLREORFASSO：ファイルのアソシエータのオンラインリオーダー開始

---

■ 基本パラメータ .....	276
■ オプションパラメータ .....	277
■ 例 .....	277

ONLREORFASSO 機能では、指定されたファイルのアソシエータをリオーダするオンライン処理を開始します。

 **Notes:**

1. オンラインリオーダ処理では既存のファイルエクステントは変更されず、エクステント内のファイルインデックスのみ再編成されます。
2. オンラインインデックスリオーダ処理では、最大限のブロック（ASSO パディングファクタに従って）外へインデックス要素を移動するのではなく、最大限になっていないブロックへ移動するだけです。
3. 解放されたインデックスブロックは未使用の RABN チェーンに挿入されません。
4. この機能は、Adabas Cluster Services 環境または Adabas Parallel Services 環境で利用できません。

```
ADADBS ONLREORFASSO FILE = file-number
                        [ASSOPFAC = asso-padding-factor]
                        [PASSWORD = password ]
                        [NOUSERABEND]
                        [TEST]
                        [WAIT]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILE は、ステートメント内の後続のパラメータを適用するファイルを指定します。

1回のONLREORFASSOオペレーション内では、複数のファイルおよび関連パラメータを指定できます。この場合、ファイルは、指定した順序でリオーダされます。

Adabas 拡張ファイルのコンポーネントファイルを指定した場合は、指定したコンポーネントファイルのアソシエータだけがリオーダされます。他のコンポーネントファイルはまったく影響を受けません。

Adabas チェックポイント、セキュリティファイル番号を指定することはできません。

## オプションパラメータ

### ASSOPFAC：アソシエータパディングファクタ

ASSOPFAC では、リオーダ処理中に使用されない各アソシエータブロック内のパーセントであるアソシエータブロックパディングファクタを定義します。1 から 90 までの範囲の値が指定できます。パディング後の空きバイト数は、最も大きいディスクリプタ値 + 10 より大きくなければなりません。

このパラメータを省略すると、ファイルの現在のパディングファクタが使用されます。

### PASSWORD：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### WAIT：プロセス終了待機

このオンライン処理が終わるまで ADADBS が次の機能や終了の処理を行わない場合に、WAIT を指定します。

WAIT を指定しないと、ADADBS はオンライン処理を開始したらただちに次の処理を実行します。

## 例

最初にファイル 10 のアソシエータを、次にファイル 11 のアソシエータをリオーダするオンライン処理を開始します。ファイル 11 のアソシエータパディングファクタは 5 パーセントです。

```
ADADBS ONLREORFASSO FILE=10
ADADBS FILE=11,ASSOPFAC=5
```



# 48 ONLREORFDATA：ファイルデータのオンラインリ オーダの開始

---

■ 基本パラメータ .....	280
■ オプションパラメータ .....	281
■ 例 .....	282

ONLREORFDATA 機能では、指定されたファイルのデータストレージをリオーダするオンライン処理を開始します。

 **Notes:**

1. オンラインリオーダ処理では既存のファイルエクステン트는変更されず、エクステント内のファイルのデータストレージレコードのみ再編成されます。
2. この機能は、Adabas Cluster Services 環境または Adabas Parallel Services 環境で利用できません。

```
ADADBS ONLREORFDATA FILE = file-number
                    [DATAPFAC = data-padding-factor ]
                    [SORTSEQ = { ISN | de-name | physical-sequence } ]
                    [PASSWORD = password ]
                    [NOUSERABEND]
                    [TEST]
                    [WAIT]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILE は、ステートメント内の後続のパラメータを適用するファイルを指定します。

1 回の ONLREORFDATA オペレーション内では、複数のファイルおよびパラメータを指定できます。この場合、ファイルは、指定した順序でリオーダされます。

Adabas 拡張ファイルのコンポーネントファイルを指定した場合は、指定したコンポーネントファイルのデータストレージだけがリオーダされます。他のコンポーネントファイルは影響を受けません。

Adabas チェックポイント、セキュリティファイル番号を指定することはできません。

## オプションパラメータ

### DATAFAC：データストレージパディングファクタ

DATAFACでは、データストレージパディングファクタを指定します。指定値は、ファイルのリオーダ時に使用しないままとなる各データストレージブロックのパーセントを表します。1～90の範囲で値を指定できます（データストレージパディングファクタの設定と使用の詳細については、「[ADALOD ユーティリティ](#)」を参照してください）。

このパラメータを省略すると、ファイルの現在のパディングファクタが使用されます。

### SORTSEQ：ファイルリオーダ順序

SORTSEQは、ファイルの処理順序を決めます。このパラメータを省略すると、レコードが物理順で処理されます。

 **Note:** 単一データストレージブロック内のレコードは、指定された順番に従ってはソートされません。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。

ISN を指定すると、ファイルが ISN 昇順に処理されます。

### PASSWORD：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### WAIT：プロセス終了待機

このオンライン処理が終わるまで ADADBS が次の機能や終了の処理を行わない場合に、WAIT を指定します。

WAIT を指定しないと、ADADBS はオンライン処理を開始したらただちに次の処理を実行します。

## 例

---

最初にファイル10のデータストレージを、次にファイル11のデータストレージをリオーダするオンライン処理を開始します。ファイル11のデータストレージパディングファクタは5パーセントです。

```
ADADBS ONLREORFDATA FILE=10
ADADBS          FILE=11 ,DATAPFAC=5
```

# 49 ONLREORFILE：ファイルのアソシエータとデータのオンラインリオーダの開始

---

■ 基本パラメータ .....	284
■ オプションパラメータ .....	285
■ 例 .....	286

## ONLREORFILE：ファイルのアソシエータとデータのオンラインリオーダの開始

ONLREORFILE 機能では、指定されたファイルのアソシエータおよびデータストレージをリオーダするオンライン処理を開始します。



### Notes:

1. オンラインリオーダ処理では既存のファイルエクステン트는変更されず、エクステント内のファイルのインデックスとデータストレージレコードのみが再編成されます。
2. この機能は、Adabas Cluster Services 環境または Adabas Parallel Services 環境で利用できません。
3. 解放されたインデックスブロックは未使用の RABN チェーンに挿入されません。

```
ADADBS ONLREORFILE  FILE = file-number
                     [ASSOPFAC = asso-padding-factor]
                     [DATAPFAC = data-padding-factor]
                     [SORTSEQ = { ISN | de-name | physical-sequence  }]
                     [PASSWORD = password ]
                     [NOUSERABEND]
                     [TEST]
                     [WAIT]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FILE は、ステートメント内の後続のパラメータを適用するファイルを指定します。

1回のONLREORFILE オペレーション内では、複数のファイルおよび関連パラメータを指定できます。この場合、ファイルは、指定した順序でリオーダされます。

Adabas 拡張ファイルのコンポーネントファイルを指定した場合は、指定したコンポーネントファイルのアソシエータとデータストレージのみがリオーダされます。他のコンポーネントファイルは影響を受けません。

Adabas チェックポイント、セキュリティファイル番号を指定することはできません。

## オプションパラメータ

### ASSOPFAC：アソシエータパディングファクタ

ASSOPFAC は、リオーダ処理中に使用されないアソシエータブロック内のパーセントを表すアソシエータブロックパディングファクタを定義します。1 から 90 までの範囲の値が指定できます。パディング後の空きバイト数は、最も大きいディスクリプタ値 + 10 より大きくなければなりません。

このパラメータを省略すると、ファイルの現在のパディングファクタが使用されます。

### DATAPFAC：データストレージパディングファクタ

DATAPFAC では、新しいデータストレージパディングファクタを指定します。指定値は、ファイルのリオーダ時に使用しないままとなる各データストレージブロックのパーセントを表します。1~90 の範囲の値を指定できます（データストレージパディングファクタの設定と使用の詳細については、「[ADALOD ユーティリティ](#)」を参照してください）。

このパラメータを省略すると、ファイルの現在のパディングファクタが使用されます。

### SORTSEQ：ファイルリオーダ順序

SORTSEQ は、ファイルの処理順序を決めます。このパラメータを省略すると、レコードが物理順で処理されます。

 **Note:** 単一データストレージブロック内のレコードは、指定された順番に従ってはソートされません。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。

ISN を指定すると、ファイルが ISN 昇順に処理されます。

### PASSWORD：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## ONLREORFILE：ファイルのアソシエータとデータのオンラインリオーダの開始

---

### WAIT：プロセス終了待機

このオンライン処理が終わるまで ADADBS が次の機能や終了の処理を行わない場合に、WAIT を指定します。

WAIT を指定しないと、ADADBS はオンライン処理を開始したらただちに次の処理を実行します。

## 例

---

最初にファイル10のアソシエータとデータストレージを、次に、ファイル11のアソシエータとデータストレージをリオーダするオンライン処理を開始します。ファイル10のアソシエータパディングファクタは5パーセント、ファイル11のデータストレージパディングファクタは10パーセントです。

```
ADADBS ONLREORFILE FILE=10,ASSOPFAC=5
ADADBS FILE=11,DATA PFAC=10
```

# 50

## OPERCOM : Adabasオペレータコマンド

---

■ クラスタ環境での OPERCOM コマンドの使用 .....	288
■ オプションパラメータ .....	289
■ オペレータコマンド .....	290

OPERCOM 機能は、Adabas ニュークリアスにオペレータコマンドを発行します。

Adabas クラスタ環境では、OPERCOM コマンドは、単一クラスタニュークリアスまたはクラスタのアクティブな全ニュークリアスに向けて行うことができます。特定のニュークリアスが指定されない場合、コマンドはデフォルトでローカルニュークリアスに送られます。

Adabas は、コマンド実行を確認できるように、オペレータにメッセージを発行します。

```
ADADBS OPERCOM operator-command  
[NOUSERABEND]  
[NUCID = {nuc-id | 0}]  
[TEST]
```

このセクションでは、いくつかのオペレータコマンドが、Adabas クラスタ環境で発行されるときに異なって動作するので、オプションパラメータの後に個々のオペレータコマンドについて説明します。

このchapterでは、次のトピックについて説明します。

## クラスタ環境での OPERCOM コマンドの使用

---

いくつかの ADARUN パラメータはグローバルパラメータ、つまり、クラスタの全ニュークリアスに対して同じ値を持たなければなりません。これらの一部はセッション初期化時に設定され、変更できません。その他は実行中のシステムで修正できます。これらの修正可能なグローバルパラメータ値を変更する OPERCOM コマンドは、クラスタ環境では特殊な方法で操作されます。

Adabas クラスタニュークリアスによって1以上のグローバルパラメータが変更される場合、これらのパラメータによってパラメータ変更ロックが確保され、ローカルパラメータエリアでの変更と他のクラスタニュークリアスへの変更通知が行われ、リプライが待たれます。他のクラスタニュークリアスでは独自のローカルパラメータエリアでの変更が行われ、承認メッセージが送信されます。

## オプションパラメータ

### GLOBAL : アクティブな全クラスタニュークリアスの操作

ADAEND、CANCEL、FEOFCL、FEOFPL、および HALT の 5 つの OPERCOM コマンドでは、GLOBAL オプションを使用してクラスタのアクティブな全ニュークリアスに渡る操作を実行します。例

```
ADADBS OPERCOM ADAEND, GLOBAL
```

他のすべての OPERCOM コマンドでは、クラスタのアクティブな全ニュークリアスに渡って操作するために NUCID=0 オプションを使用します。

### NOUSERABEND : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### NUCID : クラスタニュークリアス ID

Adabas ニュークリアスクラスタのすべてのニュークリアス実行で、ADADBS のような Adabas ユーティリティの実行が可能です。

一部の例外はありますが、NUCID パラメータを使用すると、ローカルに実行された ADADBS OPERCOM オペレーションによってコマンドが発行されたかのように、ADADBSOPERCOM コマンドを実行のためにクラスタ内の特定ニュークリアスに送ることができます。NUCID=0 を指定することによってほとんどの OPERCOM コマンドをクラスタの全ニュークリアスにルーティングできます。

NUCID がクラスタ環境で指定されない場合、コマンドはローカルニュークリアスにルーティングされます。



**Note:** ADADBSOPERCOM および Adabas オンラインシステム (AOS) では、NUCID パラメータの 0 の値はコマンドがクラスタの全ニュークリアス (グローバル) に適用されることを示します。NUCID パラメータのゼロ以外の値は、コマンドが指定されたニュークリアスにのみ適用されることを示します。

### TEST : 構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定されたパラメータの構文のみがテストされ、値や変数の正当性はテストできません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## オペレータコマンド

### ADAEND

```
ADAEND [, GLOBAL ]
```

このコマンドでは、Adabas セッションを正常終了させます。このコマンドの発行後に、新しいユーザーは受け付けられません。各ユーザーの現在の論理トランザクションが終了するまで、ET ロジックの更新が継続されます。前述したように、すべてのアクティビティが完了すると、Adabas セッションは終了します。

ニュークリアクラスタ環境では、GLOBAL オプションは、アクティブな全クラスタニュークリアスの Adabas セッションを終了するために使用することができます。

### ALOCKF



**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

```
ALOCKF = file-number
```

EXU、EXF、または UTI ユーザーが指定ファイルを排他制御できるように、ファイルを事前ロックします。事前ロックによって、新しいトランザクションがファイルを使用するのを防ぎます。すべての現在使用中のユーザーがそのファイルの使用を止めると、排他制御ユーザーはファイルをロックできます。それまで排他制御ユーザーは待つ必要があります。

ユーティリティを実行せずに事前ロックを解除する方法については、RALOCKF コマンドを参照してください。

このコマンドは次の場合は利用できません。

- シングルユーザーモード、または
- 読み取り専用ニュークリアスの場合

## CANCEL

```
CANCEL [, GLOBAL ]
```

Adabas セッションを即時にキャンセルします。すべてのコマンド処理は直ちに中断されます。保留中の自動再スタートが有効となり、次の Adabas セッションの初期化中に自動再スタートルーチンが実行されます。

ニュークリアスクラスタ環境では、GLOBAL オプションは、アクティブな全クラスタニュークリアスの Adabas セッションをキャンセルするために使用することができます。

## CLOGMRG

```
CLOGMRG = { YES | NO }
```

ニュークリアスクラスタ環境で、自動的なコマンドログマージ (ADARUNCLOGMRG パラメータ値) のオンまたはオフを切り換えます。

CLOGMRG コマンドは、定義ではグローバルであり、クラスタの全ニュークリアスに影響します。NUCID が指定された場合は、無視されます。

## CLUFREEUSER

```
CLUFREEUSER = [, TNA = max-time] [, UID = userid ] [, FORCE] [, GLOBAL] [, NUCID = nucid ]
```



**Note:** CLUFREEUSER コマンドは、クラスタ環境でのみ有効です。このコマンドは、ローカルニュークリアスに対してのみ発行することができます。また、GLOBAL オプションを指定した場合は、クラスタのすべてのアクティブおよび非アクティブなニュークリアスに対して発行することができます。

ニュークリアスでユーザーキューエレメント (UQE) との関係がなくなった共通ストレージに残るユーザーテーブルエレメント (UTE) を削除します。

TNA	タイムアウト値を指定する秒単位での 10 進数です。他の条件が満たされると、指定された時間中に使用されない UTE は削除されることがあります。TNA= が指定されていないと、UTE は、その最新の使用に関係なく削除されることがあります。	
UID	次の文字列または 16 進数のバイト文字列です。	
	<code>cccccccc</code>	引数は、1~8 桁の文字の、数字、またはアポストロフィで囲まれない組み込み (-) 記号。
	<code>'cccccccc'</code>	引数は、アポストロフィで囲まれた 1~8 文字。
	<code>X'xxxxxxxxxxxxxxxxxx'</code>	引数は、"X" が先頭で、単一引用符に囲まれた 2~16 桁 (偶数) の 16 進数です。
	文字、数字、または組み込み (-) 記号以外の文字を含んでいる場合、文字列はアポストロフィで囲まなければなりません。指定された文字列が 8 文字未満の長さであれば、暗黙的に空白が埋め込まれます。指定された 16 進数の文字列が 16 桁の 16 進数より短い場合、暗黙的にバイナリの 0 で埋め込まれます。UID 文字列が '*' (または X'5C') で終わっている場合、前の文字はユーザー ID の接頭辞として扱われます。引数一致には、接頭辞のみが必要です。ユーザーの 28 バイトのコミュニケーション ID の最後の 8 バイトが特定のユーザー ID またはユーザー ID 接頭辞に一致している場合、そのユーザーの UTE は他の条件が満たされると、削除されます。UID= が指定されていないと、UTE はそのユーザー ID に関係なく削除されることがあります。	
FORCE	ユーザーがレスポンスコード 9、サブコード 20 の予定であっても、残された UTE は削除されます。FORCE が指定されない場合、そのような UTE は削除されません。FORCE パラメータを使う前に、削除される UTE を所有しているユーザーが、そのどのトランザクションもオープンしたままにはならないことを、あらかじめ周知しておいてください。	
GLOBAL	Adabas クラスタ全体に渡って残された UTE は、UQE との関係がなくなっており、指定された他のパラメータに従ったときに適格であっても削除されます。さらに、他の規則を条件として、残された UTE は、割り当てられたニュークリアスがそれらの最後の使用以後に終了していた場合に削除されます。GLOBAL が指定されないときは、ローカルニュークリアスに割り当てられ、ニュークリアスの開始後に使用された UTE のみが削除に適格です。	
NUCID	コマンドがクラスタの特定のニュークリアスによって処理されることを示すために使用されます。	

CT

**CT = timeout-limit**

ADARUN CT パラメータの値をダイナミックに上書きします。つまり、Adabas コマンドが完了してからリージョン間コミュニケーション (使用している特定のオペレーティングシス

テムに依存する) を経由して Adabas コマンドがユーザーに結果を返すまでに経過する時間の最大秒数をダイナミックに上書きします。最小設定は 1、最大設定は 16777215 です。

ニュークリアスクラスタ環境では、CT コマンドは、グローバルに作用し、クラスタの全ニュークリアスに影響します。NUCID が指定された場合は、無視されます。

### DAUQ

直近 15 分以内に少なくとも 1 つの Adabas コマンドを実行したユーザーのユーザーキューエレメント (UQE) を表示します。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

### DCQ

ポストされたコマンドキューエレメント (CQE) をすべて表示します。各 CQE のユーザー ID、ジョブ名、およびバッファ長が表示されます。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

### DDIB

データ保全ブロック (DIB) を表示します。このブロックは、どの Adabas ユーティリティがアクティブであり、各ユーティリティがどのようなリソースを使用しているかを示すエントリを持っています。DDIB 機能は、ニュークリアスがアクティブなときでもアクティブでないときでも実行できます。

ニュークリアスクラスタ環境では、DDIB コマンドによって表示される情報はグローバルです。コマンドはどのニュークリアスでも実行することができます。

### DDSF

Adabas Delta Save ファシリティ (DSF) ステータス表示 Adabas ニュークリアスでは、ADADBS ジョブプロトコルと同様にオペレータコンソールで DSF ステータスが表示されます。

この機能は、ニュークリアスがパラメータ ADARUN DSF=YES で実行されている場合のみ使用できます。

ニュークリアスクラスタ環境では、DDSF コマンドによって表示される情報はグローバルです。コマンドはどのニュークリアスでも実行することができます。

## DFILES

```
DFILES = { n | n1-nx | n1, n2, n3, n4, n5 }
```

指定ファイルに対するアクセス、更新、EXU、およびUTIユーザーの数を表示します。ユーザータイプはファイルごとに合計され、ファイル別にリストされます。

ニュークリアスクラスタ環境では、NUCID=0オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

## DFILUSE

```
DFILUSE = file-number
```

現行のセッション中に、指定されたファイルに対して処理されたコマンドのカウンタを表示します。

ニュークリアスクラスタ環境では、NUCID=0オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

## DHQA

DHQA ホールドキューエレメント (HQE) を 1000 まで表示します。

ニュークリアスクラスタ環境では、NUCID=0オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

## DLOCKF

ロックされたファイルを表示します。

ニュークリアスクラスタ環境では、NUCID=0オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

## DNC

ポストされたコマンドキューエレメント (CQE) の数を表示します。

ニュークリアスクラスタ環境では、NUCID=0オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

**DNH**

現在ホールドキュー内にある ISN の個数を表示します。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

**DNU**

現在のユーザー数を表示します。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

**DONLSTAT**

**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

アクティブなりオーダまたはインバートの各オンライン処理のステータスとプロセス ID を表示します。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

**DPARM**

現在有効な Adabas セッションパラメータを表示します。

**DRES**

割り当てられたプールスペースおよび現在のセッション中にこれまでに到達した最高使用レベル ('上限') を、レコードカウントと次のリソースのパーセントによって表示します。

- アタッチドバッファ (AB) - サポートされない現在の割り当て
- コマンドキュー (CQ)
- フォーマットプール (FP)
- ホールドキュー (HQ)
- ISN のテーブル用プール (TBI)
- シーケンシャルコマンドテーブル用プール (TBQ または TBLES)
- ユーザーキュー (UQ)
- ユニークディスクリプタプール (DUQPOOL)
- セキュリティプール
- ユーザーキューファイルリストプール
- ワークプール (WP)

- グローバルトランザクション ID (XID、Adabas Transaction Manager の場合は 0 以外のみ) 用プール
- クラスタブロック更新の再実行プール (0 より大きい ADARUN LRDP のクラスタニュークリアスにのみ 0 以外)

実際の値は、ニュークリアスメッセージ ADAN28 中に表示されます。詳しくは『Adabas メッセージおよびコードマニュアル』を参照してください。

#### DSTAT

現在の Adabas ニュークリアスオペレーティングステータスを表示します。

#### DTH

スレッドステータスを表示します。

#### DUQ

アクティブまたは非アクティブを 5 つまで表示 ユーザーキューエレメント

#### DUQA

すべてのユーザーキューエレメント (UQE) を表示します。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

#### DUQE

**DUQE = X' user-id'**

指定したユーザー ID (Adabas に割り当てられたもの) のユーザーキューエレメントを次のとおり表示します。

```
DUQE=X' A3CF2'
```

ユーザー ID は、16 進形式で入力する必要があります。ユーザー ID にジョブ名は使用できません。

ニュークリアスクラスタ環境では、ユーザー ID がクラスタに対してユニークではないため、NUCID を常に指定する必要があります。

#### DUUQE

ユーティリティのユーザーキューエレメント (UQE) を表示します。

ニュークリアスクラスタ環境では、NUCID=0 オプションは、アクティブな全クラスタニュークリアスの情報を表示するために使用することができます。情報は、ニュークリアスごとに次々に表示されます。

**FEOFCL****FEOFCL [ , GLOBAL ]**

現在使用中のデュアルコマンドログまたはマルチコマンドログを閉じ、他のデュアルコマンドログまたは他のマルチコマンドログに切り換えます。このコマンドは、デュアルまたはマルチコマンドログを使用している場合にのみ有効です。

ニュークリアスクラスタ環境では、GLOBALオプションを使用して、すべてのクラスタニュークリアスで同時にデュアルコマンドログまたはマルチコマンドログを切り替えることができます。

**FEOFPL****FEOFPL [ , GLOBAL ]**

現在使用中のデュアルコマンドログまたはマルチデータプロテクションログを閉じ、他のデュアルコマンドログまたは他のマルチプロテクションログに切り換えます。このコマンドは、デュアルまたはマルチデータプロテクションログを使用している場合にのみ有効です。

ニュークリアスクラスタ環境では、GLOBALオプションを使用して、すべてのクラスタニュークリアスで同時にデュアルコマンドログまたはマルチプロテクションログを切り替えることができます。

**HALT****HALT [ , GLOBAL ]**

Adabasセッションを停止します。アクティブな各ETロジックユーザーに対してはBT（バックアウトランザクション）コマンドが発行されます。Adabasセッションはその後終了し、ダンプは生成されません。

ニュークリアスクラスタ環境では、GLOBALオプションは、アクティブな全クラスタニュークリアスのAdabasセッションを停止するために使用できます。

**LOCKF**

```
LOCKF = file-number
```

指定ファイルをロックします。指定したファイルは全セキュリティレベルでロックされます。

**LOCKU**

```
LOCKU = file-number
```

指定ファイルをユーティリティ以外のすべての使用に対してロックします。Adabasユーティリティではファイルを正常に使用できます。

**LOCKX**

```
LOCKX = file-number
```

すべてのユーザー（EXUまたはEXFユーザーを除く）に対して指定ファイルをロックします。EXUおよびEXFユーザーは指定ファイルを通常どおり使用できます。EXUユーザーがOPコマンドを発行すると、自動的にロックは解除されます。

**LOGGING**

コマンドロギングを開始します。

**LOG<sub>xx</sub>**

各コマンドログについて、xxによる指定どおりにロギングを開始します（xxが次の1つに当てはまる場合）。

CB	Adabas コントロールブロック
FB	Adabas フォーマットバッファ
IB	Adabas ISN バッファ
IO	Adabas I/O アクティビティ
RB	Adabas レコードバッファ
SB	Adabas サーチバッファ
UX	Adabas パラメータリストの7番目のパラメータに渡されたユーザーデータ

VB	Adabas バリュースタック
----	-----------------

**NOLOGGING**

コマンドのロギングを停止します。

**NOLOG<sub>xx</sub>**

*xx* のロギングを停止または防止します (*xx* が次の 1 つである場合)。

CB	Adabas コントロールブロック
FB	Adabas フォーマットバッファ
IB	Adabas ISN バッファ
IO	Adabas I/O アクティビティ
RB	Adabas レコードバッファ
SB	Adabas サーチバッファ
UX	Adabas パラメータリストの 7 番目のパラメータに渡されたユーザーデータ
VB	Adabas バリュースタック

**ONLRESUME**

**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

**ONLRESUME = X'identifier'**

中断されたオンラインリオーダー処理またはオンラインインバート処理を再開します。

クラスタ環境では、オンラインプロセスIDはクラスタに対してユニークでないので、NUCIDを常に指定しなければなりません。

**ONLSTOP**

**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

```
ONLSTOP = X'identifier'
```

オンラインリオーダー処理またはオンラインインバート処理を完全に停止します。この処理は、一貫性のある状態を保つために次の割り込みポイントまで続行してから、必要な後処理を行った後で終了します。

クラスタ環境では、オンラインプロセスIDはクラスタに対してユニークでないので、NUCIDを常に指定しなければなりません。

## ONLSUSPEND



**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

```
ONLSUSPEND = X'identifier'
```

オンラインリオーダー処理またはオンラインインバート処理を中断します。この処理は、一貫性のある状態を保つために次の割り込みポイントまで続行してから、コマンドの後処理を行ったうえで、処理対象として選択できない状態になります。オンライン処理がニュークリアスのリソースを使い過ぎる場合に、このコマンドを使用すると役に立ちます。

クラスタ環境では、オンラインプロセスIDはクラスタに対してユニークでないので、NUCIDを常に指定しなければなりません。

## RALOCKF



**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

```
RALOCKF = n
```

ユーティリティを実行することなく、指定されたファイルの事前ロック（ALOCKF コマンドを参照してください）を解除します。

## RALOCKFA



**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

ユーティリティを実行することなく、すべてのファイルの事前ロック（ALOCKF コマンドを参照）を解除します。

### RDUMPST

オンラインダンプステータスを終了します。このコマンドは、通常、ADASAV ユーティリティのオンライン実行が異常終了した場合に使用します。

### READONLY

 **Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

**READONLY = { YES | NO }**

READONLY ステータスをオンまたはオフにします。

ニュークリアスクラスタ環境では、READONLY コマンドは定義によりグローバルで、クラスタの全ニュークリアスに影響します。NUCID が指定された場合は、無視されます。

### REVIEW

 **Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

**REVIEW = { NO | LOCAL | *hub-id* }**

AdabasReview を停止します。ハブモードからローカルモードに変更します。ニュークリアスと通信する Adabas Review のハブを指定または変更します。

### STOPF

**STOPF = *file-number* [, PURGE]**

指定ファイルを使用中であるユーザーを停止します。ファイル番号は1つのみを指定できます。EXF ユーザーおよび UTI ユーザーは対象外です。

任意指定の PURGE パラメータは、ADARUN OPENRQ=NO が指定されたとき、中止されたユーザーキューエレメントをユーザーキューから削除します。PURGE パラメータの使用例を以下に示します。

```
ADADBS OPERCOM STOPF=5,PURGE
```

## STOPI

```
STOPI = time [, PURGE]
```

指定された *time* (秒) 内にコマンドを実行していないユーザーを停止します。EXF ユーザーおよび UTI ユーザーは対象外です。

任意指定の PURGE パラメータは、ADARUN OPENRQ=NO が指定されたとき、中止されたユーザーキューエレメントをユーザーキューから削除します。PURGE パラメータの使用例を以下に示します。

```
ADADBS OPERCOM STOPI=3600,PURGE
```

## STOPU

```
STOPU = { X'user-id' | job-name }
```



**Note:** STOPU=X'userid' コマンドは、ADAORD または ADAINV 処理に使用できません。代替手段として、ONLSTOP=X'identifier' コマンドを参考にしてください。

Adabasによって割り当てられたユーザーID (表示コマンドで表示された形式のID) を持つユーザーを停止するか、または指定されたジョブ名 (*job-name*) を持つ全ユーザーを停止します。

STOPUにより、非アクティブまたはタイムアウトしたユーザーがクリアされ、そのユーザーのユーザーキューエレメント (UQE) は削除されます。プログラム/ユーザーがETロジックユーザーで、ETステータスになく、STOPUが発行される前に終了していない場合、Adabasによって、この時点までにトランザクションによって作成されたすべての更新がバックアウトされ、ホールドされているレコードはすべて解放されます。トランザクションが続行される場合、STOPU後の変更のみが完了されます。

ユーザーIDは、16進形式で指定する必要があります。例えば、次のように指定します。

```
STOPU=X'A3CF2'
```

ニュークリアスクラスタ環境では、ユーザーIDがクラスタに対してユニークではないため、NUCIDを常に指定する必要があります。

**SYNCC**

ニュークリアスですべての ET ユーザーの再同期を実行します。ニュークリアスは、すべての ET ユーザーが ET ステータスに到達するのを待ちます。

**TNAu**

**TNAu = time**

ユーザーの非アクティビティタイムリミットを設定します (u の 1 つが次の場合)。

A	アクセスオンリー (ACC) ユーザー
E	ET ロジックユーザー
X	排他制御 (EXF/EXU) ユーザー

指定する場合、time は 0 よりも大きな値である必要があり、これによって ADARUN 値が上書きされます。

ニュークリアスクラスタ環境では、TNAu コマンドは定義によりグローバルで、クラスタの全ニュークリアスに影響します。NUCID が指定された場合は、無視されます。

**TT**

**TT = time**

ET ロジックユーザーに対するトランザクションタイムリミット (秒) を設定します。指定する場合、この値は 0 よりも大きな値である必要があり、これによって ADARUN 値が上書きされます。ニュークリアスクラスタ環境で、TT コマンドはグローバルに作用し、クラスタの全ニュークリアスに影響します。NUCID が指定された場合は、無視されます。

**UNLOCKF**

**UNLOCKF = file-number**

指定ファイルのロックを解除します。ファイル使用をロック前の状態に戻します。

## UNLOCKU

```
UNLOCKU = file-number
```

指定ファイルのユーティリティ使用ロックを解除します。ユーティリティ以外のユーザーに対してユーティリティロック前の状態に戻します。

## UNLOCKX

```
UNLOCKX = file-number
```

指定ファイルのロックを解除します。ファイル使用をロック前の状態に戻します。

## UTIONLY



**Note:** 現在 Adabas Parallel Services クラスタニュークリアスでの使用には利用できません。

```
UTIONLY = { YES | NO }
```

UTIONLY ステータスをオンまたはオフにします。

ニュークリアスクラスタ環境では、UTIONLY コマンドは定義によりグローバルで、クラスタの全ニュークリアスに影響します。NUCID が指定された場合は、無視されます。

# 51      PRIORITY：ユーザープライオリティの変更

---

■ 基本パラメータ .....	379
■ オプションパラメータ .....	306
■ 例 .....	382

PRIORITY機能は、ユーザーのAdabasプライオリティの設定または変更に使用できます。ユーザープライオリティは0（最低プライオリティ）から255（最高プライオリティ）の範囲で設定できます。

ユーザーはAdabasコントロールブロックに指定された同じユーザーIDで識別されます（OPコマンド、追加1フィールド）。

```
ADADBS PRIORITY USERID = 'user-id'  
[NOUSERABEND]  
[PRTY = { n | 255 }]  
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### USERID：ユーザーID

プライオリティを変更するユーザーのチェックポイントファイルにおけるユーザーIDです。このユーザーのレコードが存在しない場合、新しいレコードがチェックポイントファイルに追加されます。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### PRTY：ユーザープライオリティ

ユーザーに対して有効とするプライオリティです。最低プライオリティ0から最高プライオリティ 255 までの範囲の値が指定できます。デフォルトは 255 です。この値は、リージョン間コミュニケーションメカニズムによってオペレーティングシステムプライオリティに追加されます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

**例**

```
ADADBS PRIORITY USERID='USER24',PRTY=7
```

ユーザー ID が "USER24" であるユーザーのプライオリティ割り当てを "7" に設定します。



# 52 RECORDSPANNING：レコードのスパニングの有効化／無効化

---

▪ 構文 .....	311
▪ 基本パラメータ .....	311
▪ オプションパラメータ .....	311
▪ 例 .....	312

RECORDSPANNING 機能を使用して、ファイルのレコードスパニングを有効化または無効化できます。レコードスパニングが有効化されると、ファイル中の圧縮レコードのサイズがデータストレージの最大ブロックサイズを超過する場合があります。

このchapterでは、次のトピックについて説明します。

## 構文

```
ADADBS RECORDSPANNING FILE = file-number
                        MODE = { ON | OFF }
                        [PASSWORD = 'password' ]
                        [NOUSERABEND]
                        [TIMELIMIT = { 60 | timelimit } ]
                        [TEST]
```

## 基本パラメータ

### FILE：ファイル番号

FILE では、レコードスパニングを有効化または無効化するファイル番号を指定します。レコードスパニングを制御するデータベース中の有効なファイル番号を指定します。

### MODE：レコードスパニングの ON/OFF の切り替え

MODE では、ファイルに対してレコードスパニングを有効にするかどうかを指定します。有効な値は "ON" および "OFF" です。値 "ON" ではレコードスパニングがオンに、値 "OFF" ではレコードスパニングがオフに切り替えられます。

## オプションパラメータ

### NOUSERABEND：ABEND なし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザー ABEND 34（ダンプあり）またはユーザー ABEND 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力しても ABEND しません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### PASSWORD：ファイルパスワード

レコードスパニングを有効化または無効化するファイルのパスワードです。ファイルがパスワード保護されている場合、このパラメータは必須です。パスワードは単一引用符でくくります。また、この機能が正しく動作するには、パスワードによって更新権限が与えられる必要があります。

### **TIMELIMIT**：秒数

ADADBS RECORDSPANNING 機能を実行できる最大秒数です。ファイルに対してレコードスパニングがオフになっている場合、Adabasによって、ファイルにスパンドレコードがないことを確認する必要があります。この分析には一定の時間がかかることから、TIMELIMIT パラメータを使用して、ADADBS RECORDSPANNING 機能実行までに待つ時間の長さを指定します。ファイルスキャンに TIMELIMIT パラメータで指定した時間よりも長い時間がかかる場合、機能は中止され、適切なエラーメッセージが出力されます。

デフォルトは 60 秒です。

### **TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。Adabasの機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

次の例では、ファイル 17 に対してレコードスパニングをオンにします。ファイル 17 のレコードがデータストレージブロックの境界を超過しました。

```
ADADBS RECORDSPANNING MODE=ON,FILE=17
```

# 53 RECOVER : スペースの回復

---

- オプションパラメータ ..... 314

RECOVER 機能は、フリースペーステーブル (FST) を再構築することで、割り当てられたスペースを復元します。RECOVER では、使用可能スペース全体からファイル、DSST、および代替 RABN エクステンントを差し引きます。

```
ADADBS RECOVER [NOUSERABEND]
                [TEST]
```

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

# 54 REFRESH：ファイルを0件状態にする

---

■ 基本パラメータ .....	386
■ オプションパラメータ .....	386
■ 例 .....	317

REFRESH 機能は、ファイルにロードされたレコードを0件の状態にセットし、アドレスコンバータ、データストレージ、ノーマルインデックス、およびアップパーインデックスに割り当てられた第1エクステントを空の状態にセットし、また他のエクステントの割り当てを解除します。

REFRESH 機能が完了すると、オペレータコマンド LOCKU または LOCKF で前に設定されたロックはすべてリセットされます。

```
ADADBS REFRESH FILE = file-number
[NOUSERABEND]
[PASSWORD = 'password' ]
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**FILE**：ファイル番号

FILE では、空のステータスに設定するファイルを指定します。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータは必須です。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS 機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

ファイル 116 を 0 件状態にセットします。

```
ADADBS REFRESH FILE=116
```



# 55

## REFRESHSTATS：統計値のリフレッシュ

---

- オプションパラメータ ..... 320
- 例 ..... 321

REFRESHSTATS 機能は、現セッションで Adabas ニュークリアスに保守された、統計の値をリセットします。パラメータを使用して、統計値の特殊なグループに機能を制限することができます。

REFRESHSTATS を起動すると、Adabas によってニュークリアスシャットダウン統計が DD/PRINT に自動的に書き込まれます。

 **Important:** Adabas 統計値のリフレッシュは対応する値に影響します。Adabas Statistics Facility (ASF) フィールド値を含む 通常、ニュークリアスのスタートからの期間を反映するこれらの値は最終リフレッシュ後の時間を参照します。ASF ユーザーにとって、値のリフレッシュ前に、適切な ASF 機能でニュークリアスレコードを保管することは有効な場合があります。

```
ADADBS REFRESHSTATS [ALL]
                        [CMDUSAGE]
                        [COUNTERS]
                        [FILEUSAGE]
                        [NUCID = nucid ]
                        [NOUSERABEND]
                        [POOLUSAGE]
                        [THREADUSAGE]
```

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

---

### ALL：全統計値

CMDUSAGE、COUNTERS、FILEUSAGE、POOLUSAGE、および THREADUSAGE の組み合わせに対する省略形として ALL キーワードを使用できます。

オプションキーワードを1つも指定しない場合、ALL がデフォルトのオプションです。

### CMDUSAGE：コマンド使用カウンタ

CMDUSAGE パラメータは、Lx、Sx、または A1 などの Adabas ダイレクトコールコマンドのカウンタをリセットするために指定します。

### COUNTERS：使用頻度カウンタ

COUNTERS パラメータでは、ローカルまたはリモートコール、フォーマット変換、フォーマット上書き、自動再スタート、プロテクションログスイッチ、バッファフラッシュ、およびコマンドスローバックに対するカウンタフィールドのリセットを指定します。

**FILEUSAGE**：ファイル当たりのコマンドのカウンタ

FILEUSAGE パラメータでは、各ファイルに対するコマンドカウンタをリセットします。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**NUCID**：クラスタニュークリアス ID

Adabas ニュークリアスクラスタのすべてのニュークリアス実行で、ADADBS のような Adabas ユーティリティの実行が可能です。NUCID パラメータは、ADADBS REFRESHSTATS オペレーションをローカルに実行することによってコマンドが発行されたかのように、ADADBS REFRESHSTATS 機能を実行のためにクラスタの特定ニュークリアスに向けることを可能にします。

NUCID=0 を指定すると、統計値はクラスタのアクティブな全ニュークリアスのためにリフレッシュされます。

**POOLUSAGE**：ニュークリアスプールの高水位標

POOLUSAGE パラメータでは、ワークプール、コマンドキュー、またはユーザーキューなどのニュークリアスプールに対する高水位標をリセットします。

**THREADUSAGE**：スレッド当たりのコマンドカウンタ

THREADUSAGE パラメータでは、各 Adabas スレッドに対するコマンドカウンタのリセットを指定します。

## 例

ファイル 116 を 0 件状態にセットします。

```
ADADBS REFRESHSTATS
CMDUSAGE , POOLUSAGE , NUCID=3
```

NUCID=3 を持つ Adabas クラスタニュークリアスのためのシャットダウン統計値が DD/PRINT に書き込まれると、ニュークリアスのコマンドカウンタおよびプール高水位標はリセットされません。



# 56 RELEASE : ディスクリプタの解除

---

■ 基本パラメータ .....	324
■ オプションパラメータ .....	325
■ 例 .....	394

RELEASE 機能は、ディスクリプタをディスクリプタ状態から解除します。

この機能では、現在このディスクリプタのアソシエータインバーテッドリストを占有しているすべてのスペースを解放します。このスペースは、リオーダまたは ADALOD UPDATE の実行によって、このファイルに再使用できるようになります。データストレージには何の変更も行われません。

Adabas 拡張ファイルのディスクリプタスペースを解放するときは、拡張ファイルの個別コンポーネントファイルそれぞれに RELEASE 機能を実行します。コンポーネントファイルでの RELEASE オペレーションの実行ごとに変更を確認するメッセージが出力され、コンディションコード 4 が返されます。

```
ADADBS RELEASE FILE = file-number
                DESCRIPTOR = 'name'
                [NOUSERABEND]
                [PASSWORD = 'password' ]
                [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILEでは、解除するディスクリプタを含むファイルを指定します。ファイルとして Adabas システムファイルは指定できません。

### DESCRIPTOR：解放するディスクリプタ

DESCRIPTORでは、解除するディスクリプタを指定します。あらゆるディスクリプタタイプが指定できます。現在ファイルカップリングの基準として使用されているディスクリプタは指定できません。解放中のディスクリプタが ADAM ディスクリプタである場合、そのファイルは、ADAM ファイルとしては処理できなくなります。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### PASSWORD：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータは必須です。パスワードは単一引用符でくくります。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

---

```
ADADBS RELEASE FILE=31,DESCRIPTOR='AA'
```

ファイル 31 のディスクリプタ AA をディスクリプタステータスから解放します。



# 57 RENAME：ファイルデータベースの名前の変更

---

■ 基本パラメータ .....	328
■ オプションパラメータ .....	328
■ 例 .....	329

RENAME 機能は、ファイルまたはデータベースに割り当てられている名前の変更に使用します。

```
ADADBS RENAME NAME = 'name'  
[FILE = file-number ]  
[NOUSERABEND]  
[PASSWORD = 'password' ]  
[TEST]
```

このchapterでは、次のトピックについて説明します。

### 基本パラメータ

---

**NAME**：新しいファイル名

NAME はファイルに割り当てられる新しい名前です。名前は引用符に囲んで指定します ('RESERVATIONS' など)。最大 16 文字まで使用できます。

### オプションパラメータ

---

**FILE**：ファイル番号

FILE は、名前を変更するファイル番号です。0 が指定されるか、または指定が省略された場合にはデータベースの名前が変更されます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルのパスワードを指定します。ファイルがパスワード保護されている場合、このパラメータは必須です。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS 機能で TEST

パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

ファイル 2 の名前を "INVENTORY" に変更します。

```
ADADBS RENAME FILE=2,NAME='INVENTORY'
```

データベースの名前が "RESERVATIONS" に変更されます。

```
ADADBS RENAME NAME='RESERVATIONS',FILE=0
```



# 58

## RENUMBER：ファイル番号の変更

---

■ 基本パラメータ .....	332
■ オプションパラメータ .....	332
■ 例 .....	333

RENUMBER 機能では、Adabas ファイルの番号を変更します。

```
ADADBS RENUMBER FILES = current-number, new-number
[NOUSERABEND]
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**FILES**：現在のファイル番号、新しいファイル番号

ファイルに現在設定されている番号、およびファイルにこれから割り当てる新しい番号です。新しい番号がすでに別のファイルに割り当てられている場合、RENUMBER機能は実行されません。

Adabasシステムファイルは使用できません。ファイルは、セキュリティ保護および別のファイルへのカップリングが実行できず、拡張ファイルの一部として使用できません。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTESTパラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

---

## 例

---

ファイル 4 のファイル番号を 40 に変更します。

```
ADADBS RENUMBER FILES=4,40
```



# 59 REPLICATION：レプリケーションのアクティブ化

## ／非アクティブ化

---

■ 基本パラメータ .....	336
■ オプションパラメータ .....	337
■ 例 .....	338

REPLICATION 機能では、Adabas データベースファイルのレプリケーションをアクティブ化または非アクティブ化します。この機能は Adabas データベースとは使用しないでください。Event Replicator データベースとの併用は有効ではありません。

```
ADADBS REPLICATION FILE = file-number
                        {ON | OFF | MODIFY [ACTIVATE | DEACTIVATE] }
                        [DSBI = { ON | OFF } ]
                        [KEY = field | NOKEY]
                        [TARGET = dbid ]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

レプリケーションをアクティブ化または非アクティブ化するデータベースファイルのファイル番号 このパラメータは必須です。デフォルトはありません。



**Note:** レプリケーションは、Adabas システムファイルまたはサイファファイルに対してはオンにできません。

### ON、OFF、MODIFY：レプリケーションの設定

ON、OFF、または MODIFY パラメータの 1 つを設定する必要があります。デフォルト値はありません。各パラメータの説明は次のとおりです。

- ON では、指定されたファイルのレプリケーションをオンにします。
- OFF では、指定されたファイルのレプリケーションをオフにします。
- MODIFY では、レプリケーションされたファイルに対して、1 つ以上のレプリケーション関連値 (KEY、TARGET、DSBI など) を修正できます。ACTIVATE および DEACTIVATE パラメータでは、レプリケーションされたファイルのレプリケーションステータスを変更できます。

## オプションパラメータ

### ACTIVATE：レプリケーションのアクティブ化

ACTIVATE パラメータは、MODIFY が指定された場合にのみ使用できます。このパラメータでは、レプリケーションが定義されたファイルのレプリケーションをアクティブ化します。ファイルに対してレプリケーションがすでに非アクティブ化されている必要があります。

### DEACTIVATE：レプリケーションの非アクティブ化

DEACTIVATE パラメータは、MODIFY が指定されている場合にのみ使用できます。このパラメータでは、レプリケーションが定義されたファイルのレプリケーションを非アクティブ化します。ファイルに対してレプリケーションがすでにアクティブ化されている必要があります。

### DSBI：ビフォーイメージの定義

DSBI パラメータでは、ファイルでのレコードの更新中に、データストレージのビフォーイメージをレプリケーションのために収集するかどうかを定義します。DSBI パラメータを設定するためのルールは次のとおりです。

- 設定可能値："ON"、"OFF"
- "ON" または "MODIFY" が指定された場合、このパラメータは任意です。
- パラメータのデフォルト値は "ON" です（ADADBS REPLICATION ON が指定されている場合）。
- ADADBS REPLICATION OFF が指定されると、パラメータは使用できません。

レプリケーション中の Adabas データベース処理におけるこのパラメータの使用の詳細については、『*Event Replicator for Adabas Concepts*』のニュークリアス処理に関する説明を参照してください。

### KEY：プライマリキーの定義

KEY パラメータでは、レプリケーションのプライマリキーを定義します。KEY パラメータを設定するためのルールは次のとおりです。

- ON または MODIFY が指定された場合、このパラメータは任意です。
- OFF が指定された場合、このパラメータは指定できません。
- パラメータは指定できません（OFF または NOKEY が指定されている場合）。



**Note:** 指定されたフィールド名は、ファイルにおけるディスクリプタである必要があります。この場合、フィールドはディスクリプタ、サブディスクリプタ、スーパーディスクリプタなどの場合があるため、"ディスクリプタ" は総称的に使用されます。

### NOKEY：プライマリキー設定の削除

NOKEY パラメータでは、MODIFY 指定時にプライマリキー設定を削除します。NOKEY パラメータを設定するためのルールは次のとおりです。

- OFF が指定された場合、このパラメータは指定できません。

- ON が指定され、NOKEY または KEY のいずれもが指定されていない場合、パラメータはデフォルト設定となります。
- パラメータは指定できません (KEY が指定されている場合)。

### TARGET：Event Replicator ターゲット ID

TARGET では、Event Replicator ターゲットデータベース ID を定義します。このパラメータは、ON が指定された場合は必須、MODIFY が指定された場合はオプションとなり、OFF が指定された場合は使用できません。

## 例

---

```
ADADBS REPLICATION FILE=33,ON,TARGET=206,KEY=AA
```

または

```
ADADBS REPLICATION FILE=70,OFF
```

または

```
ADADBS REPLICATION FILE=71,MODIFY  
ADADBS ACTIVATE
```

または

```
ADADBS REPLICATION FILE=72,MODIFY  
ADADBS DEACTIVATE
```

# 60 RESETDIB : アクティブなユーティリティリストの エントリのリセット

---

■ 基本パラメータ .....	340
■ オプションパラメータ .....	341
■ 例 .....	341

RESETDIB機能では、アクティブなユーティリティリストのエントリ（データ保全ブロックまたはDIB）をリセットします。

Adabasにより、DIBの各Adabasユーティリティによって使用されるファイルのリストが更新されます。DDIBオペレータコマンド（またはAdabas Online System）を使用すると、どのファイルがどのジョブに使用されるかを決定するために、このブロックを表示できます。ユーティリティは正常終了時には、DIBからエントリを削除します。ユーティリティが異常終了した場合（例えば、オペレータによるジョブのキャンセル）、そのユーティリティが使用しているファイルの状態は使用中のままです。DBAは、RESETDIB機能を使用してこのようなファイルを解放できます。



**Note:** RESETDIB機能はアクティブなニュークリアスがあってもなくても実行できます。異常終了したADAORD REORDB、REORDATA、REORASSOからDIBを削除するには、アクティブなニュークリアスなしでADADBS RESETDIBを実行する必要があります。

```
ADADBS RESETDIB { JOBNAME = 'job-name' [ IDENT = identifier ] }  
                  IDENT = identifier  
                  [NOUSERABEND]  
                  [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### JOBNAME：ジョブ名

このパラメータでは、リセットするエントリのジョブ名を指定します。ジョブ名がユニークでない場合は、IDENTパラメータも同時に指定する必要があります。

### IDENT：ユーティリティ実行ID

ユーティリティの実行を識別するユニークな番号です。単独で指定することもできますが、複数のユーティリティ実行に同じ名前が使用されているときのジョブ名としても指定できます。IDはオペレータコマンドDDIBまたはAdabas Online Systemを使用して取得できます。

## オプションパラメータ

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

ジョブ "JOB1" の DIB ブロックエントリを削除します。

```
ADADBS RESETDIB JOBNAME='JOB1'
```

IDENT=127 である "JOB2" の DIB ブロックエントリを削除します。

```
ADADBS RESETDIB  
JOBNAME='JOB2', IDENT=127
```



# 61 RESETPPT : PPT のリセット

---

▪ 構文 .....	345
▪ 基本パラメータ .....	345
▪ オプションパラメータ .....	345

RESETPPT 機能は、ASSO データセット上の PPT ブロックをリセットします。PPT ブロックには、ニュークリアスにより使用される PLOG、CLOG、および WORK データセットの情報が含まれます。この情報は、WORK、PLOG、および CLOG データセットでの整合性チェックの実行に使用するのに加え、自動再起動が必要かどうかの決定にも使用します。

PPT のリセットにはリスクがあります。

- 自動再起動が保留されているときに PPT をリセットしようとする、ニュークリアスによる現在の障害からの回復が防止されます。
- PLOG および CLOG がコピー対象として保持されているときに PPT をリセットしようとする、新しい PPT 情報が書き込まれるまで、PLOG および CLOG データセットにおけるすべての整合性チェックが無効化されます。この結果、データセットが上書きされ、データは失われます。

これらのリスクがあるため、Software AG では、最大の注意を払い、Software AG 顧客サポート担当の指示に従う場合にのみ、RESETPPT を使用することをお勧めします。

 **Caution:** この機能を実行する前にニュークリアスを停止している必要があります。それ以外の場合はエラーが発生します。

このchapterでは、次のトピックについて説明します。

## 構文

```
ADADBS RESETPPT [NOUSERABEND]
                  [TEST]
```

## 基本パラメータ

必須のパラメータはありません。

## オプションパラメータ

### NOUSERABEND : ABEND なし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザー ABEND 34 (ダンプあり) またはユーザー ABEND 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力しても ABEND しません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### TEST : 構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。Adabas の機能で TEST パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。



## 62 SPANCOUNT : スパンドレコード数のカウント

---

▪ 構文 .....	348
▪ 基本パラメータ .....	348
▪ オプションパラメータ .....	348
▪ 例 .....	349

SPANCOUNT 機能では、スパニングされたファイルのレコード数をカウントおよび表示します。

このchapterでは、次のトピックについて説明します。

## 構文

---

```
ADADBS SPANCOUNT FILE = file-number
[PASSWORD = 'password' ]
[NOUSERABEND]
[TIMELIMIT = { 60 | timelimit } ]
[TEST]
```

## 基本パラメータ

---

### FILE：ファイル番号

FILE では、スパンドレコードをカウントするファイル番号を指定します。スパンドレコードをカウントするデータベース中の有効なファイル番号を指定します。

## オプションパラメータ

---

### NOUSERABEND：ABEND なし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザー ABEND 34（ダンプあり）またはユーザー ABEND 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力しても ABEND しません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### PASSWORD：ファイルパスワード

スパンドレコードをカウントするファイルのパスワードです。ファイルがパスワード保護されている場合、このパラメータは必須です。パスワードは単一引用符でくくります。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。Adabasの機能でTEST

パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

**TIMELIMIT**：秒数

ADADBS SPANCOUNT 機能を実行できる最大秒数です。機能に TIMELIMIT パラメータで指定した時間よりも長い時間がかかる場合、機能は中止され、該当するエラーメッセージが出力されます。

デフォルトは 60 秒です。

## 例

次の例では、ファイル 17 のスパンドレコードをカウントします。

```
ADADBS SPANCOUNT FILE=17
```

結果は次のとおりです。

スパニングされないレコード	15132
プライマリレコード	19345
セカンダリレコード	43897

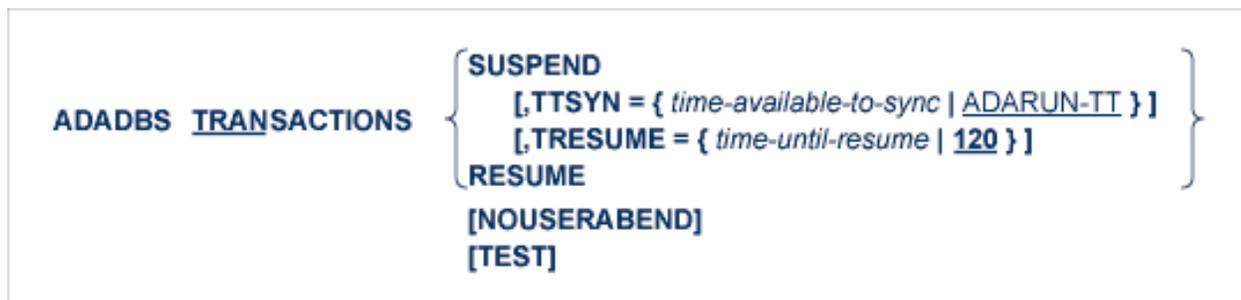


# 63 TRANSACTIONS : 中断トランザクションの再開

---

■ 基本パラメータ .....	353
■ オプションパラメータ .....	405
■ 例 .....	354

TRANSACTIONS 機能は、更新トランザクション処理を中断および再開するために使用できません。つまり、リカバリ可能な開始ポイントである静止状態に到達します。



SUSPEND 機能が送信されると、新しい更新トランザクションがユーザーキューに保持されます。トランザクションの実行は、TTSYNパラメータで割り当てられた時間内に実行できる場合は、終了できます。この時間を超過するトランザクションはバックアウトされます。クラスタ環境では、すべてのクラスタニュークリアスが同様に静止されます。

 **Note:** ADADBS TRANSACTIONS の SUSPEND または RESUME 機能を使用しようとしていて、ログオンに Natural Security が使用されている場合は、FSEC Natural Security システムファイルの FSEC Natural プロファイルパラメータを RO（読み取り専用）設定で設定する必要があります。FSECパラメータに ROが指定されていない場合、Natural Security のログオンプログラムの一部で FSEC への更新が必要なことから Adabas が中断されている間には Natural にログオンできません。

静止が成功すると、DASD ファイルがバッファ内容で最新になるように、すべてのニュークリアスに対してフラッシュされます。チェックポイント SYNC-73 は書き込まれ、ADADBS は通知されます。

この時点で、IBM の FlashCopy や Storage Tek の SnapShot のような非 Software AG の高速バックアップ製品を実行して、データベースからコピーを実行できます。すなわち、コピーによって配列ストレージデバイスの電子メモリで高速バックアップ製品によって作成されたデータがポイントされます。

 **Caution:** 正規の Software AG データベース（またはデルタ）セーブの代替としてデータベース高速コピーを使用することはお勧めしません。Software AG では、データベース高速コピーに含まれるデータセットの制御ができないだけでなく、高速コピーの成功も保証できません。さらに、デルタセーブによって影響される DSF ステータス変更がオリジナルの代わりにデータベースコピーで発生するため、デルタセーブをデータベースのコピー上で実行できません。

TRESUME タイムアウトおよび RESUME 機能の発行前に COPY が完了すると、ニュークリアスにより SYNS-74 チェックポイントが書き込まれ、中断状態を抜けて更新処理が再開されます。データベースは COPY 処理の期間全体に渡って有効な状態にありました。

COPY が TRESUME タイムアウトの前に完了しない場合、Adabas では自動的に中断状態を抜けて更新処理が再開されます。その後 RESUME 機能が発行されると、Adabas ではレスポンスコー

ドによって拒否され、ADADBSはエラーメッセージで異常終了します。つまり、更新処理が中断状態だった間にCOPYによって生成されたものは無効であり使用できません。AdabasによってCOPY処理が進行中にデータベースの更新が再開された可能性があることためです。

そのように作成されたデータベースコピーがリカバリに使用される場合、COPYの時点でデータベースをリストアする必要を取り除いて、後続の再生成は、SUSPEND機能の終了後に書き込まれたSYNC-73チェックポイントで開始する必要があります。

**!** **Important:** 他のジョブステップ、そしてRESUME機能が続くSUSPEND機能のジョブでは、その間のジョブステップには更新タイプのコマンドや機能を使用できません。そうでないと、ジョブ実行は、ニュークリアスが中断状態をタイムアウトするまで引き延ばされます。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

**SUSPEND：**トランザクションの中断とデータベースの静止

このパラメータを使用して、更新トランザクション処理を中断し、データベースを静止します。

**RESUME：**前に中断されたトランザクション処理の再開

このパラメータを使用して、前に中断された更新トランザクション処理を再開します。

Adabasが中断されていない状態または中断された状態でなくなったときに、このパラメータを使用すると、この機能はエラーで終了します。

## オプションパラメータ

**TRESUME**

このパラメータを使用して、システムが中断後ニュークリアスによって自動的に通常の更新トランザクションが開始されるまでに静止状態を保持する時間（秒）を指定します。このパラメータが指定されない場合、デフォルトは120秒で、最大は86400秒、またはおよそ24時間です。カウントは、ニュークリアスが正常に静止したときに始まります。

**TTSYN**

このパラメータを使用して、システム静止のために実行されている更新トランザクションの強制終了とバックアウトをニュークリアスが実行する前に、すべてのETユーザーがETステータスに到達するのを待機する最大時間を指定します。このパラメータが指定されない場合、デフォルトはADARUN TTの値です。

**NOUSERABEND：**アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド34（ダンプあり）またはユーザーアベンド35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベン  
ドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ  
ラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了しま  
す。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメ  
ータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTEST  
パラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参  
照してください。

## 例

---

現在実行している更新トランザクションが終了するために 300 秒、およびそれ以後 Adabas が自  
動的に正常な処理を再開する前の中断に 150 秒を許しているデータベースを静止します。

```
ADADBS TRANSACTIONS SUSPEND,TTSYN=300,TRESUME=150
```

# 64 UNCOUPLE：ファイルのアンカップル

---

■ 基本パラメータ .....	356
■ オプションパラメータ .....	406
■ 例 .....	357

UNCOUPLE 機能は、2 ファイル間の関係の除去に使用します。

```
ADADBS UNCOUPLE FILES = number, number
[NOUSERABEND]
[PASSWORD = 'password' ]
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**FILES**：アンカップルするファイル

FILES ではアンカップルする 2 つのファイルを指定します。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

PASSWORD は 1 以上のファイルのセキュリティパスワードを指定します。ファイルのいずれかがパスワード保護されている場合に必要です。両方のファイルがパスワード保護されている場合、パスワードは両方のファイルに適用されます。また、パスワードは単一引用符で囲む必要があります。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。ADADBS機能でTESTパラメータを使用する詳細については、「[TEST パラメータによる構文チェック](#)」の項を参照してください。

## 例

ファイル 62 と 201 をアンカップルします。どちらか片方、または両方が、パスワード "PAIR05" で保護されています。

```
ADADBS UNCOUPLE  
FILES=62,201,PASSWORD='PAIR05'
```



# 65 JCL/JCS の必要項目と指定例

---

■ ユーザー出口を使用した照合 .....	360
■ BS2000 .....	360
■ z/OS .....	361
■ z/VM .....	362
■ VSE .....	363

このセクションでは、ADADBS with BS2000、z/OS、z/VM、および VSE システムの実行に必要なジョブ制御情報を説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## ユーザー出口を使用した照合

ADADBS ONLINVERT 実行中に照合ユーザー出口を使用するには、ユーティリティ実行に対して ADARUN CDXnn パラメータを指定する必要があります。

UES（ユニバーサルエンコーディングサポート）と合わせて使用する場合の照合ディスクリプタユーザー出口の構文は次のとおりです。

```
ADARUN CDXnn= exit-name
```

ここでは次の内容を表しています。

nn	照合ディスクリプタ出口の番号。01 から 08 までの 2 桁の 10 進数。
exit-name	照合ディスクリプタ出口で制御を持つユーザールーチン名。最大 8 文字。

照合ディスクリプタ出口に指定できるプログラムは 1 つだけです。最大 8 個の照合ディスクリプタ出口を任意の順序で指定できます。詳細は『Adabas DBA リファレンスマニュアル』を参照してください。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn		非アクティブニュークリアスを伴う OPERCOM DDIB または RESETDIB に必要です。
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADADBS パラメータ	SYSDTA/ DDKARTE		ユーティリティ
ADARUN メッセージ	SYSOUT/ DDPRI NT		メッセージおよびコード
ADADBS メッセージ	SYSLS T/ DDDRUCK		メッセージおよびコード

## ADADBS JCL 例 (BS2000)

SDF フォーマットの場合：

```

/.ADADBS LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A D B S ALL FUNCTIONS
/REMARK *
/ASS-SYSLST L.DBS.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAhrs.MOD
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADADBS,DB=yyyyy,IDTNAME=ADABAS5B
ADADBS REFRESH FILE=1
/LOGOFF SYS-OUTPUT=DEL

```

ISP フォーマットの場合：

```

/.ADADBS LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A D B S ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.DBS
/FILE ADA.MOD,LINK=DDLIB
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADADBS,DB=yyyyy,IDTNAME=ADABAS5B
ADADBS REFRESH FILE=1
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	非アクティブニュークリアスを伴う OPERCOMDDIB または RESETDIB 機能に必要です。
ADADBS メッセージ	DDDRUCK	プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADADBS パラメータ	DDKARTE	リーダー	

## ADADBS JCL 例 (z/OS)

この例については、MVSJOBS データセットの ADADBS を参照してください。

```
//ADADBS      JOB
//*
//*          ADADBS:
//*          DATA BASE SERVICES (BATCH)
//*
//DBS          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
//SYSUDUMP    DD   SYSOUT=X
//DDCARD      DD   *
ADARUN  PROG=ADADBS,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE     DD   *
ADADBS      REFRESH FILE=1
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク/端末/リーダー	非アクティブニュークリアスを伴う OPERCOM DDIB または RESETDIB 機能に必要です。
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADADBS パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADADBS メッセージ	DDDRUCK	ディスク/端末/プリンタ	

## ADADBS JCL 例 (z/VM)

この例については、MVSJOBS データセットの ADADBS を参照してください。

```
DATADEF DDPRINT,DSN=ADADBS,DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADADBS.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNDBS.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADADBS.CONTROL,MODE=A
ADARUN
```

### RUNDBS CONTROL A1 の内容

```
ADARUN PROG=ADADBS,DEVICE=dddd,DB=yyyyy
```

### ADADBS CONTROL A1 の内容

```
ADADBS REFRESH FILE=1
```

## VSE

ファイル	ファイル名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	非アクティブニュークリアスを伴う OPERCOMDDIB または RESETDIB 機能に必要です。
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADADBS パラメータ	-	リーダー	SYSIPT	ユーティリティ
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADADBS メッセージ	-	プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

## ADADBS JCS 例 (VSE)

VSE プロシージャの内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

この例については、MVSJOBS データセットの ADADBS.X を参照してください。

```
* $$ JOB JNM=ADADBS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
*      DATABASE SERVICES (BATCH)
// JOB ADADBS
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADADBS,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADADBS REFRESH FILE=1
/*
/&
* $$ EOJ
```

# 66 ADADCK ユーティリティ：データストレージの チェック

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- DSCHECK 機能：データストレージのチェック
- JCL/JCS の必要項目と指定例



# 67 機能概要

---

ADADCKでは、データベース内の指定ファイル（複数ファイルも可）のデータストレージおよびデータストレージスペーステーブル（DSST）をチェックします。

ADADCKでは、使用されている各データストレージブロックを（ファイルコントロールブロックのデータストレージエクステンツに従って）読み込み、次のチェックを実行します。

- ブロック長が許容範囲内かどうか（4ブロック長物理ブロックサイズ）
- データストレージブロックの全レコードの長さ合計に4を加えた長さがブロック長と等しいかどうか
- レコード長がファイルの最大圧縮レコード長を超えるレコードまたは長さが0のレコードがあるかどうか
- 1つのブロック内に重複した ISN があるかどうか
- スパンドレコードが使用されている場合は、ヘッダー内の ISN は有効かどうか 各ヘッダーには、チェーン内のプライマリレコードの ISN、チェーン内の前のスパンドレコードの ISN、およびチェーン内の次のスパンドレコードの ISN が含まれているかどうか
- スパンドレコードが使用されている場合は、プライマリおよびセカンダリスパンドレコードの ID ビットが正しいかどうか ビットの1つのみがオンにされているかどうか
- 関連する DSST エLEMENT の値は正しいかどうか 値が正しくない場合は、DSST の REPAIR が必要です（「[REPAIR パラメータ](#)」参照）。



## Notes:

1. ADADCK 実行時には、Adabas ニュークリアスがアクティブである必要はありません。
2. ニュークリアスがアクティブな場合、NOOPENパラメータが指定されていない限り、ADADCK によって ADADCK のオペレーションとアクティブニュークリアスの同期が取られます。
3. 保留中の自動再スタート条件は無視されます。
4. このユーティリティの用途は、診断のみにしてください。

エラーが発生すると、ADADCKによりコンディションコード4または8が返されます。

### ADADCK およびスパンドレコード

スパンドレコードが使用されており、指定されたデータストレージ RABN のいずれかがチェーンの一部である場合は、スパンドレコード全体がチェックされます。FROMRABN および TORABN が指定されており、ファイルにスパンドデータストレージレコードが含まれている場合は、FROMRABN パラメータがプライマリレコードの RABN をポイントするようにすることをお勧めします。このようにしない場合は、警告メッセージが出力されることがあります。この場合、ADADCK 実行結果のコンディションコードは0となります。指定された FROMRABN がプライマリレコードではなくセカンダリレコードをポイントしているためにチェーン全体をチェックできないことは、単に警告によって示されます。

また、セカンダリ ISN が、FROMRABN および TORABN パラメータ指定で指定されたデータ RABN 以外のデータ RABN に存在している場合もあります。これにより、セカンダリスパンドレコードチェーンをチェックするために、追加のデータストレージ RABN を読み込むことが必要になる場合があります。パフォーマンスを最適化するために、ADADCK によってセカンダリ ISN テーブルがオンザフライで作成されます。見つからないセカンダリ ISN がある場合、不明なセカンダリ ISN の場所を特定するために、データストレージ全体の検索が試みられます。

1つのスパンドデータストレージファイルに関してチェックされる ISN の最大数は MAXPISN の設定によって決定されますが、デフォルトは1000です。ファイルに MAXPISN パラメータ設定を超過する数のプライマリ ISN が含まれている場合、処理は続行されますが、次の警告メッセージが表示されます。

```
*** Warning ***
    More than MAXPISN primary spanned ISNs.  Only
    the first MAXPISN ISNs will be checked.  Run

    ADADCK again specifying a different

    FROMRABN to check the remaining RABNs or
    specify a higher MAXPISN value.
    Any errors reported after this warning
    may be due to the table limitation.
```

この警告メッセージが表示された場合、この警告の後に報告されるあらゆるエラーまたは CC=8 コンディションは、1000 のサイズ制限によってユーティリティによる ISN の追跡ができないことによって発生している可能性があります。ファイルが損傷し、ADADCK 処理に使用される内部テーブルが ISN のすべてを保持できないことがエラーの原因となっている場合があります。ファイルが正常な状態であることを確認するには、ADAVL ユーティリティを実行します。

# 68

## DSCHECK 機能：データストレージのチェック

---

- オプションのパラメータとサブパラメータ ..... 370
- 例 ..... 372

```
ADADCK DSCHECK [FILE = { file [FROMRABN = DS-blknum ] [TORABN = DS-blknum ] | file - file }]  
                [MAXPISN = { num | 1000 }]  
                [NOOPEN]  
                [NOUSERABEND]  
                [REPAIR]  
                [USAGE]
```

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

---

### FILE：チェックするファイル

チェックするファイル（または単一のファイル範囲）です。指定がない場合は、データベース内の全ファイルがチェックされます。

### FROMRABN：データストレージブロック番号

チェックを開始するデータストレージブロックの RABN です。このパラメータは、チェック対象のファイルが1つだけの場合にのみ適用可能です。つまり、1回の ADADCK の実行で FROMRABN/TORABN の範囲を1つだけ指定できます。

複数のファイルに対して複数の FROMRABN/TORABN 範囲を1回の ADADCK 要求で指定すると、実行時に最後の範囲だけが使用されます。また、FROMRABN/TORABN パラメータで指定した範囲が任意のファイル DS エクステンツの範囲外である場合、ADADCK はエクステンツ内のブロックをチェックしません。このため、複数の FROMRABN/TORABN の範囲が指定された場合、またはこの範囲が任意のファイル DS エクステンツの範囲外である場合、Adabas は警告メッセージを発行します。

このパラメータがない場合は、ファイルに対して最初に割り当てられたデータストレージエクステンツの先頭からチェックが開始されます。

### MAXPISN

スパンドデータストレージファイルに対してチェックされるプライマリ ISN の最大数です。デフォルトは 1000 です。ファイルに MAXPISN パラメータ設定を超過する数のプライマリ ISN が含まれている場合、処理は続行されますが、次の警告メッセージが表示されます。

```

*** Warning ***
    More than MAXPISN primary spanned ISNs.  Only
    the first MAXPISN ISNs will be checked.  Run

    ADADCK again specifying a different

    FROMRABN to check the remaining RABNs or
    specify a higher MAXPISN value.
    Any errors reported after this warning
    may be due to the table limitation.

```

**NOOPEN**：オープン同期の回避

通常、ADADCKは開始するときニュークリアスに対してユーティリティのOPEN コールを実行し、影響を受けるファイルのブロックが、ニュークリアスバッファプール内に残っていないことを確認します。しかし、これはまた他のユーザーに対してそのファイルをロックすることにもなります。NOOPENを指定すると、ADADCKによってOPEN コールが発行されず、他のユーザがファイルを使用できます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**REPAIR**：データストレージスペーステーブルの修復

このパラメータが指定されている場合、ADADCKによって無効なデータストレージスペーステーブル（DSST）が検出されると、自動的にテーブルが修復されます。

**TORABN**：終了データストレージブロック番号

チェックを終了するデータストレージブロックのRABNを指定します。このパラメータは、チェック対象のファイルが1つだけの場合にのみ適用可能です。つまり、1回のADADCKの実行でFROMRABN/TORABNの範囲を1つだけ指定できます。

複数のファイルに対して複数のFROMRABN/TORABN範囲を1回のADADCK要求で指定すると、実行時に最後の範囲だけが使用されます。また、FROMRABN/TORABNパラメータで指定した範囲が任意のファイルDSエクステンツの範囲外である場合、ADADCKはエクステンツ内のブロックをチェックしません。このため、複数のFROMRABN/TORABNの範囲が指定された場合、またはこの範囲が任意のファイルDSエクステンツの範囲外である場合、Adabasは警告メッセージを発行します。

このパラメータを省略した場合は、ファイルに対して最後に割り当てられたデータストレージエクステンツの最後でチェックが終了します。

**USAGE**：データストレージブロック使用率の出力

このパラメータが指定されると、ADADCKによって、各データストレージブロックで使用されているバイト数、ブロックサイズ、およびブロックの使用率が棒グラフとして出力されます。

## 例

---

ファイル 20 のデータストレージおよび DSST をチェックし、データストレージブロックの使用状態を示す棒グラフを出力します。必要に応じてスペーステーブルを修復します。

```
ADADCK DSCHECK FILE=20, USAGE, REPAIR
```

ファイル 8 から 12 までのデータストレージおよび DSST をチェックします。

```
ADADCK DSCHECK FILE=8-12
```

ファイル 12 のデータストレージおよび DSST を、878 から 912 の RABN 範囲でチェックします。

```
ADADCK DSCHECK FILE=12,  
FROMRABN=878, TORABN=912
```

# 69

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	374
▪ z/OS .....	375
▪ z/VM .....	376
▪ VSE .....	377

このセクションでは、ADADCK with BS2000、z/OS、z/VM、および VSE システムの実行に必要なジョブ制御情報を説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSOR <sub>n</sub> または DDASSO <sub>nn</sub>	ディスク	
データストレージ	DDDATAR <sub>n</sub> または DDDATA <sub>nn</sub>	ディスク	
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADADCK パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT DDPRINT		メッセージおよびコード
ADADCK メッセージ	SYSLST DDDRUCK		メッセージおよびコード

### ADADCK JCL 例 (BS2000)

SDF フォーマットの場合：

```

/.ADADCK LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK *A D A D C K DATA STORAGE CHECK
/REMARK *
/REMARK *
/ASS-SYSLST L.DCK.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB, ADAvrs.MOD
/SET-FILE-LINK DDASSOR1, ADAyyyyy.ASSO, SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1, ADAyyyyy.DATA, SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD, ADARUN), PR-MO=ANY
ADARUN PROG=ADADCK, DB=yyyyy, IDTNAME=ADABAS5B
ADADCK DSCHECK FILE=27
/LOGOFF SYS-OUTPUT=DEL
    
```

ISP フォーマットの場合：

```

/.ADADCK LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK *A D A D C K DATA STORAGE CHECK
/REMARK *
/REMARK *
/SYSFILE SYSLST=L.DCK.DATA
/FILE ADA.MOD,LINK=DDLIB

/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADADCK,DB=yyyyyy,IDTNAME=ADABAS5B
ADADCK DSCHECK FILE=27
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSOR <sub>n</sub> または DDASSO <sub>nn</sub>	ディスク	
データストレージ	DDDATAR <sub>n</sub> または DDDATA <sub>nn</sub>	ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADADCK パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADADCK メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

### ADADCK JCL 例 (z/OS)

この例については、MVSJOBS データセットの ADADCK を参照してください。

```

//ADADCK      JOB
//*
//*          ADADCK:
//*          DATA STORAGE CHECK
//*
//DCK         EXEC PGM=ADARUN
//STEPLIB    DD  DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDDRUCK    DD  SYSOUT=X
//DDPRINT    DD  SYSOUT=X

```

```
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADADCK,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADADCK DSCHECK FILE=27
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSOR <sub>n</sub> または DDASSO <sub>nn</sub>	ディスク	
データストレージ	DDDATAR <sub>n</sub> または DDDATA <sub>nn</sub>	ディスク	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADADCK パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADADCK メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

### ADADCK JCL 例 (z/VM)

```
DATADEF DDASSOR1,DSN=ADABASv.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASv.DATA,VOL=DATAV1
DATADEF DDPRINT,DSN=ADADCK.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADADCK.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNDCK.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADADCK.CONTROL,MODE=A
```

```
ADARUN
```

RUNDCK CONTROL A1 の内容

```
ADARUN PROG=ADADCK,DEVICE=dddd,DB=yyyyy
```

ADADCK CONTROL A1 の内容

```
ADADCK DSCHECK FILE=27
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSOR <sub>n</sub> または ASSO <sub>nn</sub>	ディスク	*	
データストレージ	DATAR <sub>n</sub> または DATA <sub>nn</sub>	ディスク	*	
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADADCK パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	メッセージおよびコード
ADADCK メッセージ		プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADADCK JCS 例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

この例についてはメンバ ADADCK.X を参照してください。

```
* $$ JOB JNM=ADADCK,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADADCK
*      DATA STORAGE CHECK
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADADCK,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADADCK DSCHECK FILE=27
/*
/&
* $$ EOJ
```

# 70 ADADEF ユーティリティ：データベースの定義

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- **DEFINE**：データベースおよびチェックポイントファイルの定義
- **MODIFY**：エンコードの変更
- **NEWWORK**：ワークファイルの定義
- **JCL/JCS** の必要項目と指定例



# 71 機能概要

---

- データベースコンポーネント ..... 382
- チェックポイントファイル ..... 382

次のデータベース属性は ADADEF で定義されます。

- データベース名および ID
- データベースコンポーネント（アソシエータ、データストレージ、および WORK）
  - デバイスタイプ
  - サイズ
- チェックポイントシステムファイル
- データベースのデフォルトエンコード

## データベースコンポーネント

---

各データベースコンポーネント（アソシエータ、データストレージ、および WORK）は、ADADEF を使用して定義する前に、ADAFRM ユーティリティでフォーマットする必要があります。また、ADADEF ユーティリティを使用すると、既存データベースに新しい WORK データセットを定義することもできます。

Recovery Aid 機能を使用するシステムには、リカバリログ（RLOG）データセットが必要です。このデータセットは、最初に ADAFRM を使用してフォーマットし、次に、ADARAI ユーティリティで定義する必要があります。

## チェックポイントファイル

---

Adabas は、Adabas CL および ET コマンドで提供されたチェックポイントデータとユーザーデータを保存するために、チェックポイントシステムファイルを使用します。ファイルは、ADADEF DEFINE（データベース）機能を使用して指定する必要があります。

# 72 DEFINE : データベースおよびチェックポイント ファイルの定義

---

■ 基本パラメータ .....	386
■ オプションパラメータ .....	387
■ 例 .....	392

データベースとチェックポイントファイルは同時に定義する必要があります。

データベースパラメータには、必須の ASSOSIZE、DATASIZE、および WORKSIZE パラメータと、ASSODEV から WORKDEV までの任意パラメータ（インデントされていない）があります。これらのパラメータは、構文図に示されています。

データベース定義には FILE=...,CHECKPOINT,... ステートメントも必要です。チェックポイントファイルパラメータ（構文図で FILE ステートメントの下にインデントして示されています）は、次の FILE ステートメントの直後に指定する必要があります。例を参照してください。

```

ADADEF DEFINE  ASSOSIZE = size-list
DATASIZE = size-list
WORKSIZE = size
FILE = file-number, CHECKPOINT
    DSSIZE = size
    MAXISN = maximum-number-of-records-expected
    [ACRABN = starting-rabn ]
    [ASSOPFAC = { Associator-padding-factor | 10 } ]
    [ASSOVOLUME = 'Associator-extent-volume' ]
    [DATAPFAC = { Data-Storage-padding-factor | 10 } ]
    [DATAVOLUME = 'Data-Storage-extent-volume' ]
    [DSDEV = device-type ]
    [DSRABN = starting-rabn ]
    [DSREUSE = { NO | YES } ]
    [ISNSIZE = { 3 | 4 } ]
    [MAXDS = maximum-Data-Storage-secondary-allocation ]
    [MAXNI = maximum-normal-index-secondary-allocation ]
    [MAXUI = maximum-upper-index-secondary-allocation ]
    [NAME = { 'file-name' | CHECKPOINT } ]
    [NIRABN = starting-rabn ]
    [NISIZE = size ]
    [UIRABN = starting-rabn ]
    [UISIZE = size ]
    [ASSODEV = { device-type-list | ADARUN-device } ]
    [DATODEV = { device-type-list | ADARUN-device } ]
    [DBIDENT = { database-id | ADARUN-dbid } ]
    [DBNAME = { database-name | GENERAL-DATABASE } ]
    [FACODE = { alpha-EBCDIC-key | 37 } ]
    [FWCODE = { wide-key | 4095 } ]
    [MAXFILES = { maximum-number-of-files | 255 } ]
    [NOUSERABEND]
    [OVERWRITE]
    [RABNSIZE = { 3 | 4 } ]
    [UACODE = { alpha-ASCII-key | 437 } ]
    [UES = { YES | NO } ]
    [UWCODE = { wide-key | FWCODE-definition } ]
    [WORKDEV = { device-type-list | ADARUN-device } ]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### ASSOSIZE/ DATASIZE/ WORKSIZE：データベースサイズ

ASSO-/DATA-/WORKSIZE では、アソシエータ、データストレージ、または WORK に割り当てるブロックまたはシリンダの数を指定します。ブロック値の後には "B" を付加する必要があります。付加しない場合、値はシリンダの数とみなされます。

アソシエータまたはデータストレージが複数のデータセットに含まれる場合には、各データセットのサイズを指定する必要があります。同時に ASSODEV または DATADEV パラメータに複数のエクステントを指定する場合、対応する ASSOSIZE または DATASIZE パラメータには、位置が固定されたオペランドとして対応する順番でエクステントサイズを指定する必要があります（例を参照）。

指定できる最小の WORKSIZE は 300 ブロックです。



**Note:** ASSOSIZE または DATASIZE を指定しないと、ADADEF DEFINE 機能を実行できません。WORKSIZE を指定しないと、WORK データセットには 3 シリンダが割り当てられます。通常、3 シリンダではデータベースを起動するために十分ではないため、WORKSIZE は必須パラメータとみなされています。

### DSSIZE：データストレージサイズ

DSSIZE では、チェックポイント/データストレージに割り当てるブロックまたはシリンダの数を指定します。ブロック数を指定する場合、指定値の後に "B" を付加します（DSSIZE=80B など）。

DSSIZE および MAXDS パラメータで指定するチェックポイントファイルのサイズは、次の条件によって決定されます。

- 格納する ET データ量
- チェックポイント情報を保持させるユーティリティ実行の回数
- ユーザー ID の数

### FILE ... CHECKPOINT パラメータ

FILE...CHECKPOINT パラメータには、チェックポイントシステムファイルに使用するファイル番号を指定します。このパラメータは必須であり、ファイル番号は 5000 以下にする必要があります。

Adabas は、AdabasCL および ET コマンドで提供されたチェックポイントデータとユーザーデータを保存するために、チェックポイントシステムファイルを使用します。

### MAXISN：使用する最大 ISN

ファイルに割り当てることのできる最大 ISN です。この値を使用して、アドレスコンバータのスペース割り当てが決定されます。MAXISN を決定するときは、環境に対する ET データおよびチェックポイントデータの重要性を考慮してください。

Adabas では、チェックポイントデータより ET データが重要であるとみなします。チェックポイントシステムファイルの ET データの ISN 範囲がすべて使用されると、すぐに最初のチェックポイント ISN が削除され、ET データに与えられます。これは進行中のプロセスです。MAXISN に到達すると、すぐに新しいアドレスコンバータエクステン트가割り当てられ、チェックポイントデータに与えられます。チェックポイントデータは Adabas Online System 機能 DELCP を使用して 1 つ 1 つ削除できます。



**Note:** チェックポイントによるデータの処理方法は、今後の Adabas のリリースに従います。

## オプションパラメータ

### ACRABN/DSRABN/NIRABN/UIRABN：開始 RABN

これらのパラメータを使用すると、個々のエリアに対する割り当てを、指定された RABN で開始できます。

- ACRABN アドレスコンバータ
- DSRABN データストレージ
- NIRABN ノーマルインデックス
- UIRABN アッパーインデックス

### ASSODEV/DATADEV/WORKDEV：デバイスタイプ

ASSO-/DATA-/WORKDEV には、アソシエータ、データストレージ、および WORK に割り当てるデバイスタイプを指定します。使用されるデバイスタイプが ADARUN DEVICE パラメータで指定されたデバイスタイプと異なる場合のみ、これらのパラメータが必要です。

WORKDEV が指定されている場合、デバイスタイプは 1 つだけ使用できます。アソシエータ (ASSODEV) またはデータストレージ (DATADEV) が複数のデータセットに含まれる場合は、両方のエクステン트가 ADARUN DEVICE タイプにある場合でも、各データセットのデバイスタイプを指定する必要があります。

VSAM データセットで複数のエクステン트가使用される場合、ASSODEV および DATADEV は DD/xxxxR1=9999; DD xxxxR2=8888; ... DD/xxxxR5=5555 のダイナミックデバイスタイプを反映している必要があります。例えば、DDDATAR1 および DDDATAR2 を定義する場合は、DATADEV=9999,8888 になります。

指定されたデバイスタイプへのスペース割り当ては、同一ジョブの同一 ADADEF ステートメントまたは別の ADADEF ステートメントで ASSOSIZE、DATASIZE、および WORKSIZE パラメータを対応させることによって指定する必要があります。ASSODEV または DATADEV パラメータで複数のエクステン트를同一デバイスタイプまたは異なるデバイスタイプで指定する (例: DATADEV=3380,3350) 場合、ASSOSIZE または DATASIZE パラメータは対応する順番で関連エクステンツサイズを指定する必要があります。

### ASSOPFAC/ DATAPFAC：パディングファクタ

ASSOPFAC では、各アソシエータ RABN ブロックで今後のエントリ用に確保するスペースのパーセンテージ（パディングスペース）を定義します。このスペースは、今後のディスクリプタの拡張や ISN の追加に使用されます。指定できる値（%）は 1～90 です。この値は、ブロックオーバーフローによって既存のアドレスブロックが 2 つの新ブロックに分割される時にオーバーヘッドを生じるのを防止できるほど十分な大きさにすべきです。ASSOPFAC が指定されていない場合は、ADADEF により、パディングファクタは 10 % とみなされます。

DATAPFAC では、各データストレージ RABN ブロックで今後のエントリ用に確保するスペースのパーセンテージ（パディングスペース）を定義します。このスペースは、既存のデータレコードへの変更によってブロック内でより多くのスペースが必要になったときに使用されます。既存のブロックに入りらなくなっ切らなくなった更新済みのレコードは、別のブロックに移動する必要があります。指定できる値（%）は 1～90 です。この値は、ブロックオーバーフローによって既存のアドレスブロックが 2 つの新ブロックに分割される時にオーバーヘッドを生じるのを防止できるほど十分な大きさにすべきです。DATAPFAC が指定されていない場合は、ADADEF により、パディングファクタは 10 % とみなされます。

### ASSOVOLUME/ DATAVOLUME：エクステントボリューム



**Note:** ASSOVOLUME および DATAVOLUME の値は単一引用符で囲む必要があります。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UI エクステント）を割り当てるボリュームを指定します。

ファイルのデータストレージスペースエクステントを割り当てるときに使用するボリューム番号を指定します。

要求されたブロック数が指定のボリュームに見つからない場合、ADADEF では、ASSOVOLUME や DATAVOLUME パラメータの値に関係なく、再度割り当てを試行します。

ACRABN、UIRABN、または NIRABN が指定された場合、ADADEF では、対応するエクステントタイプを割り当てるとき、ASSOVOLUME の値が無視されます。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。

ASSOVOLUME か DATAVOLUME またはその両方が指定されない場合、ファイルのアソシエータやデータストレージスペースは、ADADEF のデフォルト割り当てルールに従ってそれぞれ割り当てられます。

### DBIDENT: データベース ID

DBIDENT では、データベースに割り当てられる ID 番号を指定します。1～65535 の範囲内の値を指定できます。このパラメータが指定されない場合は、ADARUN DBID パラメータで指定された値が使用されます。

複数データベースを作成する場合は、各データベースを一意的に識別するために、DBIDENT パラメータが必要です。

**DBNAME**：データベース名

DBNAMEはデータベースに割り当てられる名前です。この名前はADAREPユーティリティで作成するデータベースステータスレポートのタイトルに表示されます。最大16文字まで指定できます。名前にハイフン(-)以外の特殊文字または空白が含まれる場合は、名前を単一引用符で囲む必要があります。

このパラメータが指定されない場合は、デフォルト値の"GENERAL-DATABASE"が割り当てられます。

**DSDEV**：データストレージのデバイスタイプ

DSDEVでは、チェックポイントファイルのデータストレージに使用されるデバイスタイプを指定します。デフォルト値はありません。DSDEVを指定しない場合、任意のデバイスタイプが使用されます。

**DSREUSE**：ストレージの再利用

DSREUSEでは、チェックポイントファイルで有効になったデータストレージスペースを再利用できるようにするかどうかを指定します。デフォルトはYESです。

**FACODE**：英数字フィールドのエンコード

FACODEパラメータは、データベース内の全ファイルの英数字フィールドに対してデフォルトエンコードを指定します。英数字のエンコードはEBCDICに従わなければなりません（例えばスペース文字は16進数X'40'）。モードタイプまたはシフトタイプの2バイト文字セット(DBCS)のエンコードがサポートされています。固定タイプのDBCS(DBCSオンリー)のエンコードはサポートされていません。デフォルトエンコードキーは37です。

データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。ファイルがロードされるとファイルのFCBにエンコードが格納されます。

このパラメータで設定したデフォルトエンコードは、ADADEF MODIFY機能を使って変更できます。データベース全体の設定を変更しても、ロード済みのファイルのエンコードには影響を及ぼしません。

**FWCODE**：ワイド文字フィールドのエンコード

FWCODEパラメータは、データベース内の全ファイルのワイド文字(W)形式フィールドに対してデフォルトエンコードを指定します。デフォルトは4095(Unicode)です。

FWCODEパラメータを使うと、すべてのユーザーエンコードに対して広範囲なコードポイントのスーパーセットを定義するワイド文字エンコードを設定できます。例えば、Host-DBCSとShift-JISのコードポイントは約10,000ですが、Unicodeのコードポイントは約50,000です。

データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。ファイルがロードされるとファイルのFCBにエンコードが格納されます。

このパラメータで設定したデフォルトエンコードは、ADADEF MODIFY 機能を使って変更できます。データベース全体の設定を変更しても、ロード済みのファイルのエンコードには影響を及ぼしません。

### ISNSIZE：3 または 4 バイト ISN

ISNSIZE は、ファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを示します。デフォルトは 3 バイトです。

### MAXDS/MAXNI/MAXUI：最大 2 次割り当て

MAXDS/NI/UI では、データストレージ、ノーマルインデックス、およびアップパーインデックスの 2 次エクステンメントごとの最大ブロック数をそれぞれ指定します。ブロックの場合、指定値の後に "B" を付加する必要があります (MAXDS=8000B など)。65535B を超える値は使用できません。

### MAXFILES：最大ファイル番号

MAXFILES パラメータは、データベースの中にロードできる最大ファイル数を指定します。このパラメータの最小値は 3 です。許可された最大値は 5000 または ASSOR1 ブロックサイズより 1 小さい値のうち、どちらか小さい方の値です。例えば、2003 は、ASSOR1 が 3380 DASD 上に格納されたデータベースの最大 MAXFILES 値です。

指定される値によって、データベース作成時に割り当てられるファイルコントロールブロックおよびフィールド定義テーブルの数が決定します。各ファイルコントロールブロックにはアソシエータブロックが 1 つ、各フィールド定義テーブルにはアソシエータブロックが 4 つ必要となります。

このパラメータを省略すると、255 が値として割り当てられます。

データベースが作成された後は、MAXFILES の値は、ADAORD ユーティリティの REORASSO または REORDB 機能を実行することによってのみ変更できます。

### NAME：チェックポイントファイルの名前

NAME では、定義されているチェックポイントファイルの名前を指定します。この名前は、ADAREP ユーティリティで作成されるデータベースステータスレポートに表示されます。許可される最大文字数は 16 です。デフォルトファイル名は CHECKPOINT です。

### NISIZE：ノーマルインデックスサイズ

NISIZE は、ノーマルインデックスに対して割り当てられるブロックまたはシリンダ数を指定します。ブロック数を指定する場合、指定値の後に "B" を付加します (NISIZE=80B など)。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**OVERWRITE**：既存データベースの上書き

既存データベースを上書きするには、OVERWRITE を指定します。新しくフォーマットされたデータセットでデータベースを作成するときは、OVERWRITE は指定できません。

**RABNSIZE**：3 バイトまたは 4 バイト **RABN**

RABNSIZE では、データベースの RABN の長さを指定します。24 ビット RABN には 3 を、31 ビット RABN には 4 を指定します。デフォルトは 3 です。

**UACODE**：英数字フィールドのユーザーエンコード

パラメータ UACODE では、ASCII ユーザーの場合の英数字フィールドに対するデフォルトエンコードを指定します。英数字のエンコードは ASCII に従わなければなりません（例えばスペース文字は 16 進数 X'20'）。マルチバイト文字セットのエンコードがサポートされています。デフォルトエンコードは 437 です。

UACODE 値は、ロードされているファイルには格納されません。

このパラメータで設定したユーザーセッション用のデフォルトエンコードは、OP コマンドを使って書き換えることができます。ADADEF MODIFY 機能を使用して変更できます。

**UES**：ユニバーサルエンコーディングサポート

パラメータ UES を設定することで、データベースのユニバーサルエンコーディングサポートをアクティブにできます。xxCODE形式のパラメータ（FACODE、FWCODE、UACODE、UWCODE）が有効である場合は、UES=YES と見なされます。

UES を無効にするには、UES=NO を明示的に設定しなければなりません。

ADADEF MODIFY 機能を使用して、このパラメータのデフォルト設定を変更できます。

**UISIZE**：アッパーインデックスサイズ

UISIZE は、アッパーインデックスに対して割り当てられるブロック数またはシリンダ数を指定します。ブロック数を指定する場合、指定値の後に "B" を付加します（UISIZE=80B など）。

**UWCODE**：ワイド文字フィールドのユーザーエンコード

UWCODE パラメータは、ワイド文字（W）形式フィールドに対してユーザーエンコードを指定します。このパラメータを指定しない場合は、FWCODE の現在の値がデフォルト値となります。

データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。ファイルがロードされるとファイルの FCB にエンコードが格納されます。

このパラメータで設定したユーザーセッション用のデフォルトエンコードは、OP コマンドを使って書き換えることができます。ADADEF MODIFY 機能を使用して、デフォルト設定を変更できます。データベース全体の設定を変更しても、ロード済みのファイルのエンコードには影響を及ぼしません。

## 例

### 例 1

```
ADADEF DEFINE
ADADEF ASSOSIZE=200,DATASIZE=600,WORKSIZE=50
ADADEF DBIDENT=1,DBNAME=DATABASE-1
ADADEF MAXFILES=150
ADADEF FILE=1,CHECKPOINT
ADADEF NAME='DB1-CHECKPOINT',MAXISN=5000
ADADEF DSSIZE=2,NISIZE=50B,UISIZE=10B
```

アソシエータ、データストレージ、および WORK のサイズはそれぞれ 200、600、および 50 シリンダです。データベースの ID 番号は 1 で、データベース名は DATABASE-1 です。データベースにロードできる最大ファイル数（および最大ファイル番号）は 150 です。ファイル 1 は Adabas チェックポイントファイル用に確保されます。最初のシステムファイルの名前は DB1-CHECKPOINT になります。このファイルのデータストレージのサイズは 2 シリンダ、ノーマルインデックスのサイズは 50 ブロック、アップパーインデックスのサイズは 10 ブロック、MAXISN は 5000 になります。

### 例 2

```
ADADEF DEFINE
ADADEF ASSODEV=3380,DATADEV=3380,3390,WORKDEV=3380
ADADEF ASSOSIZE=100,DATASIZE=200,300,WORKSIZE=25
ADADEF DBIDENT=2,DBNAME='DATABASE_2'
ADADEF MAXFILES=255
ADADEF FILE=255,CHECKPOINT,MAXISN=5000
ADADEF DSSIZE=3,NISIZE=100B,UISIZE=20B
```

アソシエータは 3380 デバイスタイプに作成し、100 シリンダを割り当てます。データストレージは 2 つのデータセットから構成されます。最初のデータセットには、最初の DATADEV (3380) デバイスタイプで 200 シリンダを割り当てます。2 番目のデータセットには、2 番目の DATADEV (3390) デバイスタイプで 300 シリンダを割り当てます。ワークスペースには WORKDEV デバイス (3380) で 25 シリンダを割り当てます。

データベースの ID 番号は 2 で、データベース名は DATABASE\_2 です。データベースには最大 255 個のファイルをロードできます。Adabas チェックポイントファイルはこのステップ中にロードされます。

# 73 MODIFY：エンコードの変更

---

■ オプションパラメータ .....	394
■ 例 .....	396

MODIFY 機能は ADADEF DEFINE によるエンコードセットの変更に使われます。オプションのエンコードパラメータを少なくとも1つ指定する必要があります。

フィールドの実際のエンコードは FCB に格納されているため、FACODE、FWCODE、または UWCODE パラメータを変更しても、ファイルが影響を受けることはありません。データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。

```
ADADEF MODIFY [FACODE = { alpha-EBCDIC-key | current-setting }]  
              [FWCODE = { wide-key | current-setting }]  
              [NOUSERABEND]  
              [UACODE = { alpha-ASCII-key | current-setting }]  
              [UES = { YES | NO }]  
              [UWCODE = { wide-key | current-setting }]
```

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

---

### FACODE：英数字フィールドのエンコード

FACODE パラメータは、データベース内の全ファイルの英数字フィールドに対してデフォルトエンコードを指定します。英数字のエンコードは EBCDIC に従わなければなりません（例えばスペース文字は 16 進数 X'40'）。モードタイプまたはシフトタイプの 2 バイト文字セット（DBCS）のエンコードがサポートされています。固定タイプの DBCS（DBCS オンリー）のエンコードはサポートされていません。デフォルトエンコードキーは現在の設定です。

データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。ファイルがロードされるとファイルの FCB にエンコードが格納されます。データベース全体の設定を変更しても、ロード済みのファイルのエンコードには影響を及ぼしません。

### FWCODE：ワイド文字フィールドのエンコード

FWCODE パラメータは、データベース内の全ファイルのワイド文字（W）形式フィールドに対してデフォルトエンコードを指定します。デフォルトエンコードは、現在の設定です。

FWCODE パラメータを使うと、すべてのユーザーエンコードに対して広範囲なコードポイントのスーパーセットを定義するワイド文字エンコードを設定できます。例えば、Host-DBCS と Shift-JIS のコードポイントは約 10,000 ですが、Unicode のコードポイントは約 50,000 です。

データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。ファイルがロードされるとファイルのFCBにエンコードが格納されます。データベース全体の設定を変更しても、ロード済みのファイルのエンコードには影響を及ぼしません。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**UACODE**：英数字フィールドのユーザーエンコード

パラメータ UACODE では、ASCII ユーザーの場合の英数字フィールドに対するデフォルトエンコードを指定します。英数字のエンコードは ASCII に従わなければなりません（例えばスペース文字は 16 進数 X'20'）。マルチバイト文字セットのエンコードがサポートされています。デフォルトエンコードは、現在の設定です。

UACODE 値は、ロードされたファイルには格納されません。ユーザーセッション用のデフォルトエンコードは、OP コマンドを使用して上書きできます。

**UES**：ユニバーサルエンコーディングサポート

パラメータ UES を使用して、既存データベースのユニバーサルエンコーディングサポートを有効化または無効化できます。無効化は、ワイド文字（W）形式フィールドを含むファイルがロードされない場合に限って指定できます。

xxCODE 形式のパラメータ（FACODE、FWCODE、UACODE、UWCODE）が有効である場合は、UES=YES と見なされます。

UES を無効にするには、UES=NO を明示的に設定しなければなりません。

**UWCODE**：ワイド文字フィールドのユーザーエンコード

UWCODE パラメータは、ワイド文字（W）形式フィールドに対してユーザーエンコードを指定します。このパラメータを指定しない場合は、デフォルト値として現在の設定が使用されます。

データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。ファイルがロードされるとファイルのFCBにエンコードが格納されます。データベース全体の設定を変更しても、ロード済みのファイルのエンコードには影響を及ぼしません。

ユーザーセッション用のデフォルトエンコードは、OP コマンドを使用して上書きできます。

## 例

---

### 例 1

既存データベースのユニバーサルエンコーディングサポートを無効化します。データベースにはワイド文字（W）形式が含まれます。

```
ADADEF MODIFY UES=NO
```

### 例 2

データベース内の全ファイルのワイド文字（W）フィールドに対するデフォルトエンコードを、現在の設定からコードページ 835（6204のユーザー定義文字を含む繁体字中国語ホストダブルバイト）に変更します。

```
ADADEF MODIFY FWCODE=835
```

ワイド文字フィールドの実際のエンコードはFCBに格納されているため、このように変更しても、ロード済みのファイルが影響を受けることはありません。データベース全体の設定の目的は、ファイルロード時のデフォルトとして使用することです。

# 74 NEWWORK：定義ワークファイル

---

■ 基本パラメータ .....	398
■ オプションパラメータ .....	398
■ 例 .....	399

次のパラメータを使用して WORK データセットを定義します。

```
ADADEF NEWWORK WORKSIZE = size
[NOUSERABEND]
[WORKDEV = {device-type-list | ADARUN-device }]
```



**Notes:**

1. この機能の実行中は Adabas ニュークリアスをアクティブにできません。古い WORK は JCL/JCS で指定する必要があります。
2. 保留中の自動再スタートが存在する場合は、ADADEF NEWWORK 機能は実行できません。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**WORKSIZE**：WORK データセットサイズ

WORK データセットに割り当てるブロックまたはシリンダの数です。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**WORKDEV**：デバイスタイプ

新しい WORK データセットに割り当てるデバイスタイプです。

このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

## 例

---

50シリンダのサイズで新しいWORKデータセットを定義します。デバイスタイプはADARUN  
DEVICE パラメータから取得されます。

```
ADADEF NEWWORK  
ADADEF WORKSIZE=50
```



# 75 JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	402
▪ z/OS .....	403
▪ z/VM .....	405
▪ VSE .....	407

このセクションでは、ADADEF with BS2000、z/OS、z/VM、および VSE システムの実行に必要なジョブ制御情報を説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADADEF パラメータ	SYSDTA/ DDKARTE		ユーティリティ
ADARUN メッセージ	SYSOUT/ DDPRINT		メッセージおよびコード
ADADEF メッセージ	SYSLST/ DDDRUCK		メッセージおよびコード

### ADADEF JCL 例 (BS2000)

#### データベース定義

SDF フォーマットの場合：

```

/.ADADEF LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A D E F DEFINE DATABASE
/REMARK *
/ASS-SYSLST L.DEF.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO
/SET-FILE-LINK DDDATAR1,ADyyyyyy.DATA
/SET-FILE-LINK DDWORKR1,ADyyyyyy.WORK
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADADEF,DB=yyyyyy, IDTNAME=ADABAS5B
ADADEF DEFINE DBNAME=EXAMPLE-DB
ADADEF ASSOSIZE=100,DATASIZE=200,WORKSIZE=40
ADADEF MAXFILES=120
ADADEF FILE=1,CHECKPOINT
ADADEF NAME= CHECKPOINT ,MAXISN=5000,UISIZE=10B
ADADEF DSSIZE=500B,NISIZE=100B
    
```

```
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADADEF LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A D E F DEFINE DATABASE
/REMARK *
/SYSFILE SYSLST=L.DEF.DEFI
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1
/FILE ADAyyyyyy.DATA ,LINK=DDDATAR1
/FILE ADAyyyyyy.WORK ,LINK=DDWORKR1
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADADEF,DB=yyyyyy,IDTNAME=ADABAS5B
ADADEF DEFINE DBNAME=EXAMPLE-DB
ADADEF ASSOSIZE=100,DATASIZE=200,WORKSIZE=40
ADADEF MAXFILES=120
ADADEF FILE=1,CHECKPOINT
ADADEF NAME= CHECKPOINT ,MAXISN=5000,UISIZE=10B
ADADEF DSSIZE=500B,NISIZE=100B
/LOGOFF NOSPOOL
```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク (現在)	DDWORKR1 DDWORKR4	ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADADEF パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADADEF メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADADEF JCL 例 (z/OS)

## データベース定義

```
//ADADEF      JOB
//*
//*      ADADEF:
//*      DEFINE THE PHYSICAL LAYOUT OF THE DATABASE
//*      DEFINE THE NUCLEUS SYSTEMFILE: CHECKPOINT FILE
//*
//DEF          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDDRUCK    DD   SYSOUT=X
//DDPRINT    DD   SYSOUT=X
//SYSUDUMP   DD   SYSOUT=X
//DDCARD     DD   *
ADARUN PROG=ADADEF,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE    DD   *
ADADEF DEFINE DBNAME=EXAMPLE-DB,DBIDENT=YYYYY
ADADEF      ASSOSIZE=100,DATASIZE=200,WORKSIZE=40
ADADEF      MAXFILES=120
*
ADADEF FILE=19,CHECKPOINT
ADADEF      NAME='CHECKPOINT',MAXISN=5000
ADADEF      DSSIZE=100B,NISIZE=3B,UISIZE=3B
/*
```

この例については、MVSJOBS データセットの ADADEF を参照してください。

## 新しいワークの定義

```
//ADADEFNW   JOB
//*
//*      ADADEF: DEFINE NEW WORK
//*
//DEF          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDDRUCK    DD   SYSOUT=X
```

```
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADADEF,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADADEF NEWWORK WORKSIZE=60,WORKDEV=eeee
/*
```

この例については、MVSJOBS データセットの ADADEFNW を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADADEF パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADADEF メッセージ	DDDRUCK	ディスク/端末/プリンタ	

### ADADEF JCL 例 (z/VM)

#### データベース定義

```
DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASV.V.ASSO,VOL=DATAV1
DATADEF DDWORKR1,DSN=ADABASV.V.WORK,VOL=WORKV1
DATADEF DDPRINT,DSN=ADADEF.DDPRINT,MODE=A
DATADEF DUMP,DUMMY

DATADEF DDDRUCK,DSN=ADADEF.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNDEF.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADADEF.CONTROL,MODE=A
ADARUN
```

#### RUNDEF CONTROL A1 の内容

```
ADARUN  PROG=ADADEF,DEVICE=dddd,DB=yyyyyy
```

### ADADEF CONTROL A1 の内容

```
ADADEF  DEFINE  DBNAME=EXAMPLE-DB
ADADEF          ASSOSIZE=100,DATASIZE=200,WORKSIZE=40
ADADEF          MAXFILE=120
*
ADADEF  FILE=1,CHECKPOINT
ADADEF  NAME='CHECKPOINT',MAXISN=5000,UISIZE=10B
ADADEF          DSSIZE=500B,NISIZE=100B
```

### 新しいワークの定義

```
DATADEF  DDASSOR1,DSN=ADABASv. ASSO,VOL=ASSOV1
DATADEF  DDDATAR1,DSN=ADABASv. ASSO,VOL=DATAV1
DATADEF  DDWORKR1,DSN=ADABASv. WORK,VOL=WORKV1
DATADEF  DDPRINT,DSN=ADADEF.DDPRINT,MODE=A
DATADEF  DUMP,DUMMY
DATADEF  DDDRUCK,DSN=ADADEF.DDDRUCK,MODE=A
DATADEF  DDCARD,DSN=RUNDEF.CONTROL,MODE=A
DATADEF  DDKARTE,DSN=ADADEF.CONTROL,MODE=A
ADARUN
```

### RUNDEF CONTROL A1 の内容

```
ADARUN  PROG=ADADEF,DEVICE=dddd,DB=yyyyyy
```

### ADADEF CONTROL A1 の内容

```
ADADEF NEWWORK WORKSIZE=60,WORKDEV=eeee
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	
データストレージ	DATARn	ディスク	*	
ワーク (現在)	WORKR1	ディスク	*	
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADADEF パラメータ	-	リーダー	SYSIPT	
ADARUN メッセージ	-	プリンタ	SYSLST	
ADADEF メッセージ	-	プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADADEF JCS 例 (VSE)

VSE プロシージャの内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

#### データベース定義

この例については ADADEF.X を参照してください。

```
* $$ JOB JNM=ADADEF,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADADEF
*      DEFINE THE PHYSICAL LAYOUT OF THE DATABASE
*      DEFINE THE NUCLEUS SYSTEMFILE: CHECKPOINT FILE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADADEF,MODE=SINGLE,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADADEF DEFINE DBNAME=EXAMPLE-DB,DBIDENT=yyyyy
ADADEF      ASSOSIZE=100,DATASIZE=200,WORKSIZE=40
ADADEF      MAXFILES=120
*
```

```
ADADEF FILE=19,CHECKPOINT
ADADEF NAME='CHECKPOINT',MAXISN=5000
ADADEF DSSIZE=100B,NISIZE=3B,UISIZE=3B
/*
/&
* $$ EOJ
```

### 新しいワークの定義

この例については ADADEFNW.X を参照してください。

```
* $$ JOB JNM=ADADEFNW,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADADEFNW
* DEFINE NEW WORK
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADADEF,MODE=SINGLE,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADADEF NEWWORK WORKSIZE=60,WORKDEV=eeee
/*
/&
* $$ EOJ
```

# 76 ADAFRMユーティリティ：Adabas データセットのフォーマット

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- データベースコンポーネントのフォーマット
- JCL/JCS の必要項目と指定例



# 77 機能概要

---

- ステートメントの制約 ..... 412
- フォーマッティングオペレーション ..... 412

プライマリ Adabas ダイレクトアクセス (DASD) データセットは、ADAFRM ユーティリティを使用してフォーマットする必要があります。

これらのデータセットには、アソシエータ、データストレージ、および WORK データセット以外に中間ストレージ (一時、ソート、コマンド/プロテクション/リカバリロギング) データセットが含まれます。

Adabas ニュークリアスまたは Adabas ユーティリティで新しいデータセットを使用するには、その前にフォーマットを実行しておく必要があります。ADADBSINCREASE または ADD 機能でデータセットを拡張後に、新たに RABN もフォーマットする必要があります。

ADAFRM には、既存のアソシエータ、データストレージ、または WORK のブロック/シリンダをバイナリの 0 (NULL) にリセットする機能もあります。リセットすると、既存のアソシエータ、データストレージ、または WORK データセットの指定されたブロックは、バイナリの 0 で埋められます。

最後に、Adabas 8 以降を使用している場合は、このユーティリティを使用して、PLOG 全体を再フォーマットすることなく、PLOG から複数 PLOG ヘッダをクリアできます。詳細については、の ADAFRM **FROMRABN**、**NUMBER**、および **SIZE** パラメータのドキュメントに記載されているように、FROMRABN、NUMBER、および SIZE パラメータを組み合わせて使用方法について参照してください。

## ステートメントの制約

---

同じジョブの中で複数の ADAFRM 機能 (ASSOFRM や DATAFRM など) を実行することができます。しかし、各機能は個別のステートメントで分けて指定しなければなりません。詳細については、このセクションの最後にある例を参照してください。

## フォーマットオペレーション

---

ADAFRM によるフォーマットは 2 つの基本オペレーションにより構成されます。

1. 指定されたトラック/シリンダでのブロック (RABNS と呼ばれるもの) 作成
2. 作成したブロックへのバイナリ 0 の適用 (NULL)

# 78 データベースコンポーネントのフォーマッティン グ

---

▪ フォーマッティングモード .....	414
▪ 構文 .....	414
▪ 基本パラメータ .....	416
▪ オプションパラメータ .....	416
▪ 例 .....	418

このchapterでは、次のトピックについて説明します。

### フォーマットモード

ADAFRM には複数のフォーマットモードがあります。

1. **新しいデータセットをフォーマットします (...FRM 機能)**。機能名と NUMBER パラメータによって指定されたデータセットのみがアクセスされ、フォーマットされます。新しいデータセットをフォーマットする場合は、FROMRABN パラメータを指定できません。
2. **既存 (一部) データセットをフォーマットします (ASSOFRM、DATAFRM、WORKFRM、および TEMPFRM 機能)**。ここでは、z/OS プラットフォームを除いて、FROMRABN パラメータを指定する必要があります。WORK およびデータストレージをフォーマットするとき (WORKFRM および DATAFRM)、ADAFRM ジョブ制御にはアソシエータデータセットも含まれる必要があります。

このフォーマットモードは、ASSO および DATA の ADADBS INCREASE 機能の組み合わせで使用されます。より大きい WORK が必要な場合は、ADADEF NEWWORK を使用する必要があります。

3. **既存 (ブロック) データセットを再フォーマットします (...RESET 機能)**。このモードでは、データベース内のアソシエータ、データストレージ、および WORK のすべてのデータセットがアクセス用に開かれます。これらの機能のために FROMRABN パラメータを指定する必要があります。
4. **Adabas8以降を使用している場合は、このユーティリティを使用して、PLOG 全体を再フォーマットすることなく、PLOG から複数 PLOG ヘッダをクリアできます。** 詳細については、の ADAFRM **FROMRABN**、**NUMBER**、および **SIZE** パラメータのドキュメントに記載されているように、FROMRABN、NUMBER、および SIZE パラメータを組み合わせる方法について参照してください。

### 構文

アソシエータ (ASSO..) またはデータストレージ (DATA..) データセットをフォーマットする場合は、この構文を使用します。

```
ADAFRM { ASSOFRM | DATAFRM } SIZE = size
      [DEVICE = { device-type | ADARUN-device } ]
      [ { FROMRABN = { starting-rabn | NEXT } |
        NUMBER = { dataset-number | 1 } } ]
      [NOUSERABEND]
```

WORK (WORK..) データセット、コマンドログ (CLOG..) 、プロテクションログ (PLOG..) 、またはソート (SORT..) データセットをフォーマットする場合は、この構文を使用します。



**Note:** Adabas 8 以降を使用している場合は、この構文を使用して、PLOG 全体を再フォーマットすることなく、PLOG から複数 PLOG ヘッダをクリアすることもできます。詳細については、の ADAFRM *FROMRABN*、*NUMBER*、および *SIZE* パラメータのドキュメントに記載されているように、FROMRABN、NUMBER、および SIZE パラメータを組み合わせて使用する方法について参照してください。

```
ADAFRM { WORKFRM | CLOGFRM | PLOGFRM | SORTFRM }
      SIZE = size
      [DEVICE = { device-type | ADARUN-device }]
      [ { FROMRABN = starting-rabn | NUMBER = { dataset-number | 1 } } ]
      [NOUSERABEND]
```

リカバリログ (RLOG..) データセットをフォーマットするには、この構文を使用します。

```
ADAFRM RLOGFRM SIZE = size
      [DEVICE = { device-type | ADARUN-device }]
      [NOUSERABEND]
```

一時 (TEMP..) データセットをフォーマットするには、この構文を使用します。

```
ADAFRM TEMPFRM SIZE = size
      [DEVICE = { device-type | ADARUN-device }]
      [FROMRABN = starting-rabn ]
      [NOUSERABEND]
```

既存のアソシエータ、データストレージ、または WORK データセットのブロックを再フォーマットするには、この構文を使用します。

```
ADAFRM { ASSORESET | DATARESET | WORKRESET }  
      SIZE = size  
      FROMRABN = start-rabn  
      [NOUSERABEND]
```

## 基本パラメータ

**SIZE**：フォーマットするエリアのサイズ

SIZE では、フォーマット（またはリセット）するエリアのサイズを指定します。ブロック（10 進値の後に "B" を付加したもの）またはシリンダを指定できます。RLOGFRM 機能については、ADARAI ユーティリティの PREPARE 機能で RLOGSIZE によって指定されたサイズと同じサイズである必要があります。「[基本パラメータ](#)」を参照してください。

Adabas 8 以降を使用していて、PLOG 全体を再フォーマット化せずに PLOG から複数の PLOG ヘッダーをクリアする場合は、**SIZE** パラメータの値を "1" に設定し、**FROMRABN** および **NUMBER** パラメータの両方に値を指定します。

## オプションパラメータ

**DEVICE**：デバイスタイプ

DEVICE は、フォーマットするエリアが含まれる物理デバイスタイプです。DEVICE が指定されない場合は、ADARUN DEVICE パラメータで指定されたデバイスタイプが使用されます。

**FROMRABN**：開始 RABN

FROMRABN では、フォーマットまたはリセットを開始する RABN を指定します。このパラメータは既存のデータセットにのみ指定でき、NUMBER は FROMRABN と同じ ADAFRM ジョブでは指定できません。

xxxxFRM 機能で FROMRABN が指定されると、FROMRABN ポイントでフォーマットが開始され、（サイズがブロックで指定されたかブロックに変換されたとみなして）FROMRABN + SIZE で求められた RABN 前の最大の完全トラックまでフォーマットは続行されます。つまり、指定された範囲（FROMRABN + SIZE）内の最終トラックは、全トラックの RABN がその範囲内にある場合にのみフォーマットされます。

Adabas 8 以降を使用していて、PLOG 全体を再フォーマット化せずに PLOG から複数の PLOG ヘッダーをクリアする場合は、**SIZE** パラメータの値を "1" に設定し、**FROMRABN** および **NUMBER** パラメータの両方に値を指定します。

ASSOまたはDATA データセットのサイズを拡張するとき、VSE、z/VM、およびBS2000でのみFROMRABNをオプションとして利用できます。指定するRABNは、ADADBSを使用した論理拡張(ADAFRMによる論理拡張の前に実行する必要があります)以前に割り当てられた最大のRABNよりも大きなRABNにする必要があります。FROMRABN=NEXTの指示により、ADAFRMでは最初の未フォーマットRABNがFROMRABNの値として使用されます。次に、ADAFRMによって、NEXT値で決定されたフォーマットのためのブロック範囲がフリースペーステーブル(FST)に含まれていることが確認されます。含まれていない場合、ADAFRMはERROR-126で終了します。z/OSでは、VTOCの最後のレコードポインタを機能FROMRABNによって修正することができないため、FROMRABNは既存ブロックの再フォーマットにのみ使用してください。「[ADADBS INCREASEの例](#)」を参照してください。

このパラメータは、ASSORESET、DATARESET、およびWORKRESET機能に必要です。機能ASSORESETを使用して指定する場合、FROMRABNは30より大きい値である必要があります。

#### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド34(ダンプあり)またはユーザーアベンド35(ダンプなし)で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"(エラーのためユーティリティを終了します)を表示してコンディションコード20で終了します。

#### NUMBER：データセット番号

NUMBERでは、フォーマットする非シーケンシャルコマンドログ、非シーケンシャルプロテクションログ、アソシエータ、データストレージ、およびソートデータセットを選択します。デフォルトは1(最初のデータセット)です。指定できる値:

- ASSO または DATA (データストレージ) には 1 から 99 の値を指定できます。
- プロテクションログ (PLOG) または コマンドログ (CLOG) には 1 から 8 の値を指定できます。
- リカバリログ (RLOG) には 1 だけを指定できます。
- SORT には 1 または 2 を指定できます (VSE システムでは 1 のみ)。
- WORK または TEMP には 1 または デフォルト値を指定できます。

同じジョブで他のADAFRMステートメントによってFROMRABNの値が指定されている場合、ADAFRM...FRM機能ステートメントではNUMBER値を指定できません(また、NUMBER値へのデフォルト設定も行われません)。

NUMBERは、関連するデータ定義(DD)ステートメントの接尾辞番号と一致する必要があります。指定できるステートメントとその例については、「[JCL/JCSの必要項目と指定例](#)」を参照してください。

Adabas 8 以降を使用していて、PLOG 全体を再フォーマット化せずに PLOG から複数の PLOG ヘッダーをクリアする場合は、**SIZE** パラメータの値を "1" に設定し、**FROMRABN** および **NUMBER** パラメータの両方に値を指定します。

### 例

---

#### 例 1

アソシエータ用に 50 シリンダ、データストレージ用に 200 シリンダ、WORK 用に 10 シリンダ、リカバリログ (RLOG) 用に 2 シリンダをフォーマットします。

```
ADAFRM ASSOFRM SIZE=50,DEVICE=3380
ADAFRM DATAFRM SIZE=200,DEVICE=3380
ADAFRM WORKFRM SIZE=10,DEVICE=3380
ADAFRM RLOGFRM SIZE=2
```

#### 例 2

非シーケンシャルコマンドログデータセット 1 用に 1 シリンダ、非シーケンシャルコマンドログデータセット 2 用に 1 シリンダをフォーマットします。

```
ADAFRM CLOGFRM SIZE=1,DEVICE=3350,NUMBER=1
ADAFRM CLOGFRM SIZE=1,DEVICE=3350,NUMBER=2
```

#### 例 3

既存 WORK データセットの最初の 2 ブロックをバイナリ 0 にリセットします。

```
ADAFRM WORKRESET FROMRABN=1,SIZE=2B
```

#### 例 4

データストレージデータセットが 3380 ディスク上にあるとした場合 (9 ブロック/トラック、15 トラック/シリンダ)、データセットの先頭位置から見てシリンダ位置 201 に相当する位置から 100 シリンダがフォーマットされます。

```
ADAFRM DATAFRM SIZE=100,FROMRABN=26992
```

#### 例 5

VSE、z/VM、または BS2000 では、データベースのアソシエータで 200 シリンダ拡張されたときのみ、このジョブによってデータベースで新しいスペースがフォーマットされます。サポートされているすべてのプラットフォームに共通する詳細な例については、[オペレーティングシステム固有のプロシージャ](#)のセクションにある ADADBS INCREASE の例を参照してください。

```
ADAFRM ASSOFRM SIZE=200,FROMRABN=NEXT
```

# 79 JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	420
▪ z/OS .....	422
▪ z/VM .....	423
▪ VSE .....	425

このセクションでは、BS2000、z/OS、z/VM、および VSE システムで ADAFRM を実行するために必要なジョブ制御情報を説明し、各ジョブストリームの例を示します。

 **Note:** オプションの Recovery Aid (RLOG) で実行している場合は、すべての中間データセットがジョブ制御でもカタログされる必要があります。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	フォーマット対象のデータセット
データストレージ	DDDATARn		
ワーク	DDWORKR1 DDWORKR4		
一時	DDTEMPR1		
ソート	DDSORTRn		
マルチコマンドログ	DDCLOGRn		
マルチプロテクションログ	DDPLOGRn		
リカバリログ	DDRLOGR1		
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAFRM パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/ DDPRINT		メッセージおよびコード
ADAFRM メッセージ	SYSLST/ DDDRUCK		メッセージおよびコード

### ADAFRM JCL 例 (BS2000)

SDF フォーマットの場合：

```

/.ADAFRM LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A F R M ALL FUNCTIONS
/REMARK *

/ASS-SYSLST L.FRM
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyy.ASSO,OPEN-MODE=OUTIN,BUFF-LEN=STD(1)
/SET-FILE-LINK DDDATAR1,ADAyyyyy.DATA,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
/SET-FILE-LINK DDWORKR1,ADAyyyyy.WORK,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
    
```

```

/SET-FILE-LINK DDTEMPR1,ADAyyyyy.TEMP,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
/SET-FILE-LINK DDSORTR1,ADAyyyyy.SORT,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
/SET-FILE-LINK DDPLOGR1,ADAyyyyy.PLOGR1,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
/SET-FILE-LINK DDPLOGR2,ADAyyyyy.PLOGR2,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
/SET-FILE-LINK DDRLOGR1,ADAyyyyy.RLOGR1,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN  PROG=ADAFRM,DB=yyyyy, IDTNAME=ADABAS5B
ADAFRM  ASSOFRM  SIZE=100
ADAFRM  DATAFRM SIZE=200
ADAFRM  WORKFRM  SIZE=40
ADAFRM  SORTFRM  SIZE=25
ADAFRM  TEMPFRM  SIZE=10
ADAFRM  PLOGFRM  SIZE=40,NUMBER=1
ADAFRM  PLOGFRM  SIZE=40,NUMBER=2
ADAFRM  RLOGFRM  SIZE=10
/LOGOFF  SYS-OUTPUT=DEL

```

### ISP フォーマットの場合：

```

/.ADAFRM LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A F R M ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.FRM
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO ,LINK=DDASSOR1,OPEN=OUTIN,BLKSIZE=(STD,1)
/FILE ADAyyyyy.DATA ,LINK=DDDATAR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/FILE ADAyyyyy.WORK ,LINK=DDWORKR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/FILE ADAyyyyy.TEMP ,LINK=DDTEMPR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/FILE ADAyyyyy.SORT ,LINK=DDSORTR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/FILE ADAyyyyy.PLOGR1,LINK=DDPLOGR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/FILE ADAyyyyy.PLOGR2,LINK=DDPLOGR2,OPEN=OUTIN,BLKSIZE=(STD,2)
/FILE ADAyyyyy.RLOGR1,LINK=DDRLOGR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/EXEC (ADARUN,ADA.MOD)
ADARUN  PROG=ADAFRM,DB=yyyyy, IDTNAME=ADABAS5B
ADAFRM  ASSOFRM  SIZE=100
ADAFRM  DATAFRM SIZE=200
ADAFRM  WORKFRM  SIZE=40

ADAFRM  SORTFRM  SIZE=25
ADAFRM  TEMPFRM  SIZE=10
ADAFRM  PLOGFRM  SIZE=40,NUMBER=1
ADAFRM  PLOGFRM  SIZE=40,NUMBER=2
ADAFRM  RLOGFRM  SIZE=10

```

/LOGOFF NOSPOOL

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	フォーマット対象のデータセット
データストレージ	DDDATARn		
ワーク	DDWORKR1 DDWORKR4		
一時	DDTEMPR1		
ソート	DDSORTRn		
マルチコマンドログ	DDCLOGRn		
マルチプロテクションログ	DDPLOGRn		
リカバリログ	DDRLOGR1		
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAFRM パラメータ	DDKARTE	ディスク	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAFRM メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

### ADAFRM JCL 例 (z/OS)

この例については、MVSJOBS データセットの ADAFRM を参照してください。

```
//ADAFRM      JOB
//*
//*      ALLOCATE AND FORMAT THE DATABASE COMPONENTS
//*
//*      MORE THAN ONE DATA SET CAN BE FORMATTED IN A SINGLE RUN
//*
//*
//FRM        EXEC PGM=ADARUN
//STEPLIB    DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD                <=== ADABAS LOAD
//*
//DDASSOR1   DD   DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.ASSOR1, <=== ASSO
//            SPACE=(CYL,(0,100)),UNIT=DISK,VOL=SER=VOL001
//DDDATAR1   DD   DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.DATAR1, <=== DATA
//            SPACE=(CYL,(0,200)),UNIT=DISK,VOL=SER=VOL002
//DDWORKR1   DD   DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.WORKR1, <=== WORK
//            SPACE=(CYL,(0,40)),UNIT=DISK,VOL=SER=VOL003
```

```
//DDSORTR1 DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.SORTR1, <=== SORT
//          SPACE=(CYL,(0,100)),UNIT=DISK,VOL=SER=VOL003
//DDTEMPR1 DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.TEMPR1, <=== TEMP
//          SPACE=(CYL,(0,100)),UNIT=DISK,VOL=SER=VOL003
//DDPLOGR1 DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.PLOGR1, <=== PLOG1
//          SPACE=(CYL,(50)),UNIT=DISK,VOL=SER=VOL003
//DDPLOGR2 DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.PLOGR2, <=== PLOG2
//          SPACE=(CYL,(50)),UNIT=DISK,VOL=SER=VOL003
//DDCLOGR1 DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.CLOGR1, <=== CLOG1
//          SPACE=(CYL,(50)),UNIT=DISK,VOL=SER=VOL003
//DDCLOGR2 DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyy.CLOGR2, <=== CLOG2
//          SPACE=(CYL,(50)),UNIT=DISK,VOL=SER=VOL003
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAFRM,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAFRM ASSOFRM SIZE=100,DEVICE=dddd
ADAFRM DATAFRM SIZE=200,DEVICE=dddd
ADAFRM WORKFRM SIZE=40,DEVICE=dddd
ADAFRM SORTFRM SIZE=100,DEVICE=dddd
ADAFRM TEMPFRM SIZE=100,DEVICE=dddd
ADAFRM PLOGFRM SIZE=50,NUMBER=1,DEVICE=dddd
ADAFRM PLOGFRM SIZE=50,NUMBER=2,DEVICE=dddd
ADAFRM CLOGFRM SIZE=50,NUMBER=1,DEVICE=dddd
ADAFRM CLOGFRM SIZE=50,NUMBER=2,DEVICE=dddd
/*
```

## z/VM

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	フォーマット対象のデータセット
データストレージ	DDDATARn		
ワーク	DDWORKR1 DDWORKR4		
一時	DDTEMPR1		
ソート	DDSORTRn		
マルチコマンドログ	DDCLOGRn		
マルチプロテクションログ	DDPLOGRn		
リカバリログ	DDRLOGR1		
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル

データセット	DD名	ストレージ	追加情報
ADAFRM パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAFRM メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

## ADAFRM JCL 例 (z/VM)

```

DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDWORKR1,DSN=ADABASV.V.WORK,VOL=WORKV1
DATADEF DDSORTR1,DSN=ADABASV.V.SORT,VOL=SORTV1
DATADEF DDTEMPR1,DSN=ADABASV.V.TEMP,VOL=TEMPV1
DATADEF DDPLOGR1,DSN=ADABASV.V.PLOG1,VOL=PLOGV1
DATADEF DDPLOGR2,DSN=ADABASV.V.PLOG2,VOL=PLOGV2
DATADEF DDRLOGR1,DSN=ADABASV.V.RLOG1,VOL=RLOGV1
DATADEF DDPRINT,DSN=ADAFRM.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAFRM.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNFRM.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAFRM.CONTROL,MODE=A
ADARUN
    
```

## RUNFRM CONTROL A1 の内容

```

ADARUN PROG=ADAFRM,DEVICE=dddd,DB=yyyyy
    
```

## ADAFRM CONTROL A1 の内容

```

ADAFRM ASSOFRM SIZE=100
ADAFRM DATAFRM SIZE=200
ADAFRM WORKFRM SIZE=40
ADAFRM SORTFRM SIZE=25
ADAFRM TEMPFRM SIZE=10
ADAFRM PLOGFRM SIZE=40
ADAFRM PLOGFRM SIZE=40,NUMBER=2
    
```

```
ADAFRM RLOGFRM SIZE=10
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	フォーマットするファイル
データストレージ	DATARn			
ワーク	WORKR1			
一時	TEMPR1			
ソート	SORTR1			
マルチコマンドログ	CLOGRn			
マルチプロテクションログ	PLOGRn			
リカバリログ	RLOGR1			
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADAFRM パラメータ	-	リーダー	SYSIPT	
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADAFRM メッセージ	-	プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADAFRM JCS 例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

この例については、メンバ ADAFRM.X を参照してください。

```
* $$ JOB JNM=ADAFRM,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAFRM
*          FORMAT THE DATABASE COMPONENTS
?/ EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAFRM,MODE=SINGLE,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAFRM ASSOFRM SIZE=100,DEVICE=dddd
ADAFRM DATAFRM SIZE=200,DEVICE=dddd
```

```
ADAFRM WORKFRM SIZE=40,DEVICE=dddd
ADAFRM SORTFRM SIZE=100,DEVICE=dddd
ADAFRM TEMPFRM SIZE=100,DEVICE=dddd
ADAFRM PLOGFRM SIZE=50,NUMBER=1,DEVICE=dddd
ADAFRM PLOGFRM SIZE=50,NUMBER=2,DEVICE=dddd
ADAFRM CLOGFRM SIZE=50,NUMBER=1,DEVICE=dddd
ADAFRM CLOGFRM SIZE=50,NUMBER=2,DEVICE=dddd
/*
/&
* $$ EOJ
```

# 80 ADAICK ユーティリティ：インデックスおよびアドレスコンバータのチェック

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ACCHECK：アドレスコンバータのチェック
- ASSOPRINT：アソシエータブロックの出力／ダンプ
- BATCH：出力を1行当たり132文字に設定
- DATAPRINT：データストレージブロックの出力／ダンプ
- DSCHECK：データストレージレコード内容の出力／ダンプ
- DUMP：ダンプ省略の解除
- FCBPRINT：ファイルコントロールブロックの出力／ダンプ
- FDTPRINT：フィールド定義テーブルの出力／ダンプ
- GCBPRINT：ジェネラルコントロールブロックの出力／ダンプ
- ICHECK：インデックスおよびアドレスコンバータのチェック
- INT：フォーマット出力省略の解除
- NIPRINT：ノーマルインデックスの出力／ダンプ
- NOBATCH：出力を1行当たり80文字に設定
- NODUMP：ダンプの省略
- NOINT：フォーマット出力の省略
- PPTPRINT：Parallel Participant Table の出力／ダンプ
- UIPRINT：アップパーインデックスの出力／ダンプ
- 例
- JCL/JCS の必要項目と指定例



# 81 機能概要

---

ADAICKでは次の物理構造をチェックします。アソシエータこのチェックでは、ディスクリプタ値の構成と、ジェネラルコントロールブロック（GCB）およびファイルコントロールブロック（FCB）に定義されたアソシエータエクステンントに従って、インデックスの整合性チェックが行われます。

ADAICKユーティリティは、診断のためにのみ使用してください。

ADAICKでは次の機能が実行できます。

- 指定ファイルのインデックスやアドレスコンバータをチェックします。
- データベース内の ASSO または DATA ブロックの内容を出力／ダンプします。
- ノーマルインデックス（NI）およびアップパーインデックス（UI）の内容を出力／ダンプします。
- GCB、FCB、FDT、PPTの内容を一定のフォーマットで出力／ダンプします。

ADAICKユーティリティ実行でスパンドレコードに ISN を指定するときは、必ず、スパンドレコードのセカンダリ ISN ではなく、プライマリ ISN を指定してください。ADAICKユーティリティ処理は、指定されたすべての ISN がプライマリ ISN であることを前提としています。セカンダリ ISN は自動的に処理されます。



## Notes:

1. ADAICKはアクティブ Adabas ニュークリアスがあってもなくても実行できます。
2. 保留中の自動再スタート条件は無視されます。
3. ニュークリアスがアクティブの場合、NOOPENパラメータが指定されていない限り、ADAICKによって ADAICK のオペレーションとアクティブなニュークリアスが同期されます。
4. データベースでスパンドレコードが使用される場合、ADAICK DSCHECKによって出力されたレポートにより、プライマリおよびセカンダリスパンドレコード ISN が識別されます。



## 82 ACCHECK：アドレスコンバータのチェック

---

■ 基本パラメータ .....	432
■ オプションパラメータ .....	432
■ 出力例 .....	433

ACCHECK 機能では、指定されたデータベースファイルのアドレスコンバータがチェックされます。さらに、ファイルにスパンドレコードが含まれている場合は、セカンダリアドレスコンバータが自動的にチェックされます。

```
ADAICK ACCHECK FILE = file-number  
[NOPEN]  
[NOUSERABEND]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：チェックするファイル

チェックするファイルです。ファイル番号は ADAICK を最初に実行するときに必要です。

次の実行で FILE が指定されない場合は、ADAICK によって最後に使用されたファイルがチェックされます。

## オプションパラメータ

---

### NOOPEN：オープン同期の回避

開始時において、ADAICK は通常、ユーティリティの OPEN コールをニュークリアスに対して発行し、関係するファイルのブロックがニュークリアスバッファプール内に残っていないことを確認します。しかし、これはまた他のユーザーに対してそのファイルをロックすることにもなります。NOOPEN を指定すると、ADAICK は OPEN コールを発行しなくなります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

## 出力例

ADAICK ACCHECK から生成された次の出力サンプルは、スパンドレコードを使用し、このため、セカンダリアドレスコンバータを含んでいる Adabas 8 データベースを対象としています。

```

FILE 002 AC CHECK
FILE 002          AC                      LOW          HIGH
      NR OF          THRU
FILE 002          ISN-ISN                RABN          DS RABN
      DS RABN      RECORDS ISN (DEC)
FILE 002          00000001-0000034F 000000E1      000001D9 0000023C
      25          847
FILE 002          00003850-00003B9F 000000F2      00000000 00000000
      25          15,263
FILE 002 AC2 CHECK
FILE 002          AC2                    LOW AC2      HIGH AC2
      NR OF          THRU
FILE 002          AC2 ISN-ISN            RABN          DS RABN      DS RABN
RECORDS AC2 ISN
FILE 002          00000001-0000034F 000000F3 000001DA 0000023F          100
      848

```



# 83 ASSOPRINT : アソシエータブロックの出力／ダン プ

---

■ 基本パラメータ .....	436
■ オプションパラメータ .....	436

```
ADAICK ASSOPRINT RABN = {rabn | rabn-rabn}
[NOUSERABEND]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### RABN：処理対象の RABN

出力／ダンプする RABN（または RABN 範囲）。ADAICK でブロックに格納された情報のタイプを識別できる場合（UI、NI など）、フォーマットされた出力が生成されます。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 84 BATCH：出力幅を1行あたり132文字に設定

---

- オプションパラメータ ..... 438

```
ADAICK BATCH [NOUSERABEND]
```

ADAICK をバッチモードで使用する場合は、この機能を使用して出力幅を 1 行当たり 132 文字に設定できます。出力幅の再設定については、「[NOBATCH 機能](#)」を参照してください。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 85 DATAPRINT：データストレージブロックの出力／ ダンプ

---

■ 基本パラメータ .....	440
■ オプションパラメータ .....	440

ADAICK DATAPRINT では、指定レコードのデータストレージブロックを出力（またはダンプ）できます。スパンドレコード RABN が要求されると、出力では、レコードがプライマリ（"プライマリ ISN"）またはセカンダリ（"AC2 ISN"）レコードとして識別されます。さらに、マスタ ISN（スパンドレコードのプライマリ ISN）と次の ISN の番号がリストされます。次の ISN 番号は常に次のセカンダリレコード ISN です（最終レコードの場合はゼロになります）。

スパンドデータストレージレコードがプライマリまたはセカンダリ ISN ビットセットで適切にフラグ設定されていないと、"ERRORSPAN\*" 句が出力に表示される場合があります。コンディションコード "4" が示されます。

```
ADAICK DATAPRINT RABN = { rabn | rabn-rabn }  
[NOUSERABEND]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### RABN：処理対象の RABN

出力／ダンプする RABN（または RABN 範囲）。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 86 DSCHECK : データストレージレコードの出力／ダンプ

---

■ 基本パラメータ .....	442
■ オプションパラメータ .....	442
■ 出力例 .....	443

```
ADAICK DSCHECK FILE = file-number  
[ISN = isn-of-record]  
[NOOPEN]  
[NOUSERABEND]
```

Adabas 8 以降を実行している場合、スパンドデータストレージレコードが使用されていると、ADAICK DSCHECK からの出力でプライマリおよびセカンダリ ISN が識別されます。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

レコードを出力／ダンプするファイルの番号です。ファイル番号は ADAICK を最初に実行するときが必要です。

次の実行で FILE が省略されると、最後に ADAICK がアクセスしたファイルが使用されません。

## オプションパラメータ

---

### ISN：データストレージレコードの ISN

出力するデータストレージレコードの ISN です。ISN が指定されない場合は、DSCHECK 機能により最終 ISN の次のレコードが出力されます。

データストレージレコードがスパンドレコードである場合は、スパンドレコードのセカンダリ ISN ではなく、必ずプライマリ ISN を指定してください。ADAICK ユーティリティ処理は、指定されたすべての ISN がプライマリ ISN であることを前提としています。セカンダリ ISN は自動的に処理されます。

### NOOPEN：オープン再同期の回避

開始時において、ADAICK は通常、ユーティリティの OPEN コールをニュークリアスに対して発行し、関係するファイルのブロックがニュークリアスバッファプール内に残っていないことを確認します。しかし、これはまた他のユーザーに対してそのファイルをロックすることにもなります。NOOPEN を指定すると、ADAICK は OPEN コールを発行しなくなります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベン  
ドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ  
ラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了しま  
す。

## 出力例

スパンドレコードを使用する Adabas 8 のデータベースに対して ADAICK DSCHECK から生成  
された出力の例です。レポートではプライマリおよびセカンダリスパンドレコード ISN が識別  
されることに注意してください。

```

000001D9 0004 0000 LEN 130D AB IS THE LAST FIELD IN THE RECORD
000001D9
000001D9 000A 0006 ISN 00000001 Primary ISN=1
000001D9 000E 000A FLAGS 0081
000001D9 0010 000C Next ISN 00000001
000001D9 0014 0010 Primary 00000001
000001DA 0004 0000 LEN 1303 BA IS THE LAST FIELD IN THE RECORD
000001DA
000001DA 000A 0006 AC2 ISN 00000001 AC2 ISN=1
000001DA 000E 000A FLAGS 0041
000001DA 0010 000C Next ISN 00000002
000001DA 0014 0010 Primary 00000001

```



# 87 DUMP : ダンプ出力形式

---

- オプションパラメータ ..... 446

```
ADAICK { DUMP | NODUMP } [NOUSERABEND]
```

この機能では、ダンプ形式でのブロックの出力が要求されます。ダンプ形式での出力の省略についての詳細は、「**NODUMP 機能**」を参照してください。この機能は、**ASSOPRINT**、**DATAPRINT**、**FCBPRINT**、**FDTPRINT**、**GCBPRINT**、および **PPTPRINT** 機能と組み合わせて使用する必要があります。

INT 機能と DUMP 機能の両方を指定した場合は、ブロックが両方の形式で出力されます。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 88 FCBPRINT：ファイルコントロールブロックの出力

## ／ダンプ

---

■ 基本パラメータ .....	448
■ オプションパラメータ .....	448
■ 出力に関する考慮事項 .....	449

```
ADAICK FCBPRINT FILE = file-number  
[NOPEN]  
[NOUSERABEND]
```

ファイルコントロールブロック（FCB）はダンプ／出力されます。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FCBを出力／ダンプするファイルの番号です。ファイル番号はADAICKを最初に実行するときに必要です。

次の実行でFILEが省略されると、最後にADAICKがアクセスしたファイルが使用されません。

## オプションパラメータ

### NOOPEN：オープン再同期の回避

開始時において、ADAICKは通常、ユーティリティのOPENコールをニュークリアスに対して発行し、関係するファイルのブロックがニュークリアスバッファプール内に残っていないことを確認します。しかし、これはまた他のユーザーに対してそのファイルをロックすることにもなります。NOOPENを指定すると、ADAICKはOPENコールを発行しなくなります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20で終了します。

## 出力に関する考慮事項

最初の未使用 RABN に 1 を加算した数が最終 RABN と等しい場合は、エクステントが一杯であり、おそらく、追加のエクステントが存在します。次の例における最初のエクステント（青色で強調表示）がこれに当てはまります。この場合、最初の未使用 RABN は 00002BFE で、これは、エクステントの最終 RABN (00002BFD) より 1 大きいものです。

```
FI 00018 FCB +1A4      First NI RABN: 00002945
FI 00018 FCB +1A8      Last NI RABN: 00002BFD
FI 00018 FCB +1AC      First unused NI RABN: 00002BFE
FI 00018 FCB +1B0      First NI RABN: 00002EE3
FI 00018 FCB +1B4      Last NI RABN: 00002FCB
FI 00018 FCB +1B8      First unused NI RABN: 00002FBC
```

最初の未使用 RABN は必ずしも次のエクステントにある必要はありません。



# 89 FDTPRINT：フィールド定義テーブルの出力／ダン プ

---

■ 基本パラメータ .....	452
■ オプションパラメータ .....	452

```
ADAICK FDTPRINT FILE = file-number  
[NOUSERABEND]
```

フィールド定義テーブル（FDT）はダンプ／出力されます。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FDTが出力／ダンプされるファイルの番号です。ファイル番号は ADAICK を最初に実行するときが必要です。

次の実行で FILE が省略されると、最後に ADAICK がアクセスしたファイルが使用されません。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 90 GCBPRINT : ジェネラルコントロールブロック (GCB) の出力 / ダンプ

---

- オプションパラメータ ..... 454

```
ADAICK GCBPRINT [NOUSERABEND]
```

ジェネラルコントロールブロック（GCB）はダンプ／出力されます。

## オプションパラメータ

---

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 91 ICHECK：インデックスおよびアドレスコンバータ のチェック

---

- 基本パラメータ ..... 456
- オプションパラメータ ..... 530

```
ADAICK ICHECK FILE = {file-number | file-number-file-number }  
[NOOPEN]  
[NOUSERABEND]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：チェック対象ファイル

チェックする指定ファイル（または単一のファイル範囲）です。FILE は指定する必要があります。

## オプションパラメータ

---

### NOOPEN：オープン再同期の回避

開始時において、ADAICK は通常、ユーティリティの OPEN コールをニュークリアスに対して発行し、関係するファイルのブロックがニュークリアスバッファプール内に残っていないことを確認します。しかし、これはまた他のユーザーに対してそのファイルをロックすることにもなります。NOOPEN を指定すると、ADAICK は OPEN コールを発行しなくなります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 92 INT：解釈済みの出力形式

---

- オプションパラメータ ..... 458

```
ADAICK {INT|NOINT}[NOUSERABEND]
```

この機能により解釈済みの形式でのブロックの出力が要求されます。解釈済みの出力形式での出力の省略についての詳細は、「**NOINT 機能**」を参照してください。この機能は、**ASSOPRINT**、**DATAPRINT**、**FCBPRINT**、**FDTPRINT**、**GCBPRINT**、および **PPTPRINT** 機能と組み合わせて使用する必要があります。

INT 機能と DUMP 機能の両方を指定した場合は、ブロックが両方の形式で出力されます。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 93 NIPRINT : ノーマルインデックスの出力 / ダンプ

---

- 基本パラメータ ..... 460
- オプションパラメータ ..... 460

```
ADAICK NIPRINT FILE = file-number  
[NOUSERABEND]
```

## 基本パラメータ

---

### FILE：ファイル番号

ノーマルインデックスを出力／ダンプするファイルの番号です。ファイル番号は ADAICK を最初に実行するときが必要です。

次の実行で FILE が省略されると、最後に ADAICK がアクセスしたファイルが使用されま  
す。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザー  
アベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベン  
ドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ  
ラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了しま  
す。

# 94 NOBATCH：出力幅を1行あたり80文字に設定

---

- オプションパラメータ ..... 462

```
ADAICK NOBATCH [NOUSERABEND]
```

出力幅を 1 行あたり 80 文字に設定します。出力幅の再設定については、「[BATCH 機能](#)」を参照してください。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 95 NODUMP : ダンプ出力形式の省略

---

- オプションパラメータ ..... 464

**ADAICK { DUMP | NODUMP } [NOUSERABEND]**

この機能では、ADAICKダンプに対して、デフォルトでダンプ出力形式の出力が省略されます。解釈済みの出力形式での出力の省略についての詳細は、「[DUMP 機能](#)」を参照してください。この機能は、[ASSOPRINT](#)、[DATAPRINT](#)、[FCBPRINT](#)、[FDTPRINT](#)、[GCBPRINT](#)、および[PPTPRINT](#) 機能と組み合わせて使用する必要があります。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 96 NOINT：解釈済みの形式の省略

---

- オプションパラメータ ..... 466

**ADAICK {INT | NOINT} [NOUSERABEND]**

この機能では、ADAICKによって、デフォルトで解釈済み出力形式の出力が省略されます。解釈済みの出力形式についての詳細は、「[INT 機能](#)」を参照してください。この機能は、[ASSOPRINT](#)、[DATAPRINT](#)、[FCBPRINT](#)、[FDTPRINT](#)、[GCBPRINT](#)、および [PPTPRINT](#) 機能と組み合わせて使用する必要があります。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 97 PTPRINT : Parallel Participant Table の出力 / ダンプ

---

■ オプションパラメータ .....	468
■ 出力例 .....	468

**ADAICK PPTPRINT [NOUSERABEND]**

Adabas クラスタをダンプ / 出力する Parallel Participant Table (PPT) です。ダンプ / 出力では、'PPH' が PPT ヘッダーのタグで、'PPE' が PPT エントリのタグであることに注意してください。

PPT のために割り当てられた 32 のブロック (RABN) それぞれがクラスタ内のシングルニュークリアスを表し、以下を含みます。

- 固定長の単一ヘッダー
- 可変長の複数エントリ

ダンプ / 出力では、'PPH' が PPT ブロックヘッダーのタグで、'PPE' が PPT ブロックエントリのタグです。

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

### NOUSERABEND : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

## 出力例

```
ADAICK PPTPRINT

MEANING: DUMP ASSO BLOCK 000000BF THRU 000000DE
DB 00072 PPT AT RABN          000000BF
DB 00072 PPT BLOCK NUMBER 01
DB 00072 PPH+000              NUMBER OF ENTRIES: 03
DB 00072 PPH+001              NUCLEUS INDICATOR: C0
DB 00072 PPH+002              EXTERNAL NUCID: 0000
DB 00072 PPH+004              UNUSED: 00000000
```

```

DB 00072 PPE+000                LENGTH OF PPT ENTRY: 0023
DB 00072 PPE+002 HDDATE FROM FIRST PLOG BLK (HIGH): 00000000
DB 00072 PPE+006 HDDATE FROM FIRST PLOG BLK (LOW): 00000000
DB 00072 PPE+00A                PTT STATUS FLAG: 00
DB 00072 PPE+00B                ID OF PPT ENTRY: W
DB 00072 DATASET=ADABAS.GB.UTI.72.WORKR1
DB 00072 PPE+000                LENGTH OF PPT ENTRY: 0023
DB 00072 PPE+002 HDDATE FROM FIRST PLOG BLK (HIGH): 00000000
DB 00072 PPE+006 HDDATE FROM FIRST PLOG BLK (LOW): 00000000
DB 00072 PPE+00A                PTT STATUS FLAG: 00
DB 00072 PPE+00B                ID OF PPT ENTRY: 1
DB 00072 DATASET=ADABAS.GB.UTI.72.PLOGR1
DB 00072 PPE+000                LENGTH OF PPT ENTRY: 0023
DB 00072 PPE+002 HDDATE FROM FIRST PLOG BLK (HIGH): 00000000
DB 00072 PPE+006 HDDATE FROM FIRST PLOG BLK (LOW): 00000000
DB 00072 PPE+00A                PTT STATUS FLAG: 00
DB 00072 PPE+00B                ID OF PPT ENTRY: 2
DB 00072 DATASET=ADABAS.GB.UTI.72.PLOGR2

```

```

ASSO BLOCK 000000BF PPT
0000 03C00000 00000000 00230000 00000000 * . . *
0010 000000E6 7AC1C4C1 7A5BC7C5 C24BE4E3 * WADABAS.GB.UT*
0020 C94BF7F2 4BE6D6D9 D2D9F100 23000000 *I.74.WORKR1 . *
0030 00000000 0000F17A C1C4C17A 5BC7C5C2 * 1ADABAS.GB*
0040 4BE4E3C9 4BF7F24B D7D3D6C7 D9F10023 *.UTI.74.PLOGR1 .*
0050 00000000 00000000 00F27AC1 C4C17A5B * 2ADABAS*
0060 C7C5C24B E4E3C94B F7F24BD7 D3D6C7D9 *.GB.UTI.74.PLOGR*
0070 F2000000 00000000 00000000 00000000 *2 *
0080 00000000 00000000 00000000 00000000 * *
SAME
OFF0 00000000 00000000 00000000 * *

```

```

DB 00072 PPT RABNS 000000C0 - 000000DE (02-32) ARE UNUSED

```

```

A D A I C K TERMINATED NORMALLY

```

```

2000-07-26 09:45:19

```



# 98      UIPRINT : アッパーインデックスの出力 / ダンプ

---

- 基本パラメータ ..... 472
- オプションパラメータ ..... 472

```
ADAICK  UIPRINT  FILE = file-number  
[NOUSERABEND]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

アッパーインデックスを出力／ダンプするファイルの番号です。ファイル番号は ADAICK を最初に実行するときが必要です。

次の実行で FILE が省略されると、最後に ADAICK がアクセスしたファイルが使用されます。

## オプションパラメータ

---

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

# 99 例

---

## 例 1

ファイル18についてインデックスとアドレスコンバータをチェックし、このファイルのFDTを出力／ダンプします。

```
ADAICK ICHECK FILE=18
ADAICK FDTPRINT
```

## 例 2

出力幅を1行あたり120文字に設定します（プリンタ）。ファイル1についてインデックスとアドレスコンバータをチェックし、アソシエータRABNの123から135までを出力／ダンプします。

```
ADAICK BATCH
ADAICK ICHECK FILE=1
ADAICK ASSOPRINT RABN=123-135
```

## 例 3

以下によって、アソシエータブロック（ASSOPRINT）の解釈済み書式（INT）出力が生成されます。NODUMP機能が指定されているため、ブロックのダンプは出力されません。

```
ADAICK INT NODUMP ASSOPRINT RABN=1
```

```
DB 00204 GCB 00000001
DB 00204 GCB+000          DATA BASE ID: 00CC
DB 00204 GCB+002          MAXIMUM NR OF FILES: 07D3
DB 00204 GCB+004          FILE 1 FCB RABN: 0000001F
DB 00204 GCB+008          FILE 1 FDT RABN: 000007F2
DB 00204 GCB+00C          CURRENT SIBA NUMBER: 0003
.....
```

## 例 4

## 例

---

以下によって、アソシエータブロック (ASSOPRINT) のダンプ (DUMP) が出力されます。  
NOINT 機能が指定されているため、ブロックの解釈済み書式出力は生成されません。

```
ADAICK NOINT DUMP ASSOPRINT RABN=1
```

```
ASSO BLOCK 00000001 0000 00CC07D3 0000001F 000007F2 00034000 * ..L . .2 . * 0010  
D4C1E3E3 C8C9C1E2 60C1C4C1 C2C1E240 *MATTHIAS-ADABAS *
```

# 100

## JCL/JCS の必要項目と指定例

---

■ ユーザー出口を使用した照合 .....	476
■ BS2000 .....	476
■ z/OS .....	478
■ z/VM .....	479
■ VSE .....	480

このセクションでは、ADAICK with BS2000、z/OS、z/VM、および VSE システムの実行に必要なジョブ制御情報を説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## ユーザー出口を使用した照合

ADAICK 実行中に照合ユーザー出口を使用する場合、ユーティリティ実行時に ADARUNCDXnn パラメータを指定する必要があります。

UES（ユニバーサルエンコーディングサポート）と合わせて使用する場合の照合ディスクリプタユーザー出口の構文は次のとおりです。

```
ADARUN CDXnn= exit-name
```

ここでは次の内容を表しています。

nn	照合ディスクリプタ出口の番号。01 から 08 までの 2 桁の 10 進数。
exit-name	照合ディスクリプタ出口で制御を持つユーザールーチン名。最大 8 文字。

照合ディスクリプタ出口に指定できるプログラムは 1 つだけです。最大 8 個の照合ディスクリプタ出口を任意の順序で指定できます。詳細は『Adabas DBA リファレンスマニュアル』を参照してください。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAICK パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT DDPRINT		メッセージおよびコード
ADAICK メッセージ	SYSLST DDDRUCK		メッセージおよびコード

**ADAICK JCL 例 (BS2000)**

SDF フォーマットの場合：

```

/.ADAICK LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK *A D A I C K INDEX CHECK
/REMARK *
/REMARK *
/ASS-SYSLST L.ICK.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAhrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAICK,DB=yyyyy,IDTNAME=ADABAS5B
ADAICK ICHECK FILE=27
/LOGOFF SYS-OUTPUT=DEL

```

ISP フォーマットの場合：

```

/.ADAICK LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK *A D A I C K INDEX CHECK
/REMARK *
/REMARK *
/SYSFILE SYSLST=L.ICK.DATA
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyy.DATA ,LINK=DDATAR1,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAICK,DB=yyyyy,IDTNAME=ADABAS5B
ADAICK ICHECK FILE=27
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAICK パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAICK メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADAICK JCL 例 (z/OS)

この例については、MVSJOBS データセットの ADAICK を参照してください。

```
//ADAICK      JOB
//*
//*      ADAICK:
//*          INDEX AND ADDRESS CONVERTER CHECK
//*
//ICK          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
//SYSUDUMP    DD   SYSOUT=X
//DDCARD      DD   *
ADARUN PROG=ADAICK,SVC=xxx,DEVICE=dddd,DBID=yyyyy
//*
//DDKARTE     DD   *
```

```
ADAICK ICHECK FILE=1-3
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAICK パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAICK メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

### ADAICK JCL 例 (z/VM)

```
DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDPRINT,DSN=ADAICK.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAICK.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNICK.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAICK.CONTROL,MODE=A
ADARUN
```

### RUNICK CONTROL A1 の内容

```
ADARUN PROG=ADAICK,DEVICE=dddd,DB=yyyyy
```

### ADAICK CONTROL A1 の内容

ADAICK ICHECK FILE=27

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	
データストレージ	DATARn	ディスク	*	
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADAICK パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	メッセージおよびコード
ADAICK メッセージ		プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADAICK JCS 例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

この例については、ADAICK.X を参照してください。

```
* $$ JOB JNM=ADAICK,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAICK
*      INDEX AND ADDRESS CONVERTER CHECK
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAICK,MODE=SINGLE,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAICK ICHECK FILE=1-3
/*
/&
* $$ EOJ
```

# 101 ADAINVユーティリティ：インバーテッドリスト の管理

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- **COUPLE**：ファイルカップリングディスクリプタの定義
- **INVERT**：ディスクリプタの作成
- **JCL/JCS** の必要項目と指定例



# 102 機能概要

---

INVERT 機能は次のことを行います。

- フィールド定義テーブル (FDT) を変更し、指定したフィールドがディスクリプタであることを示します。
- フィールドのすべての値と対応する ISN リストをインバーテッドリストに追加します。

新規に定義したディスクリプタは、他のディスクリプタと同様に使用できます。この機能は、サブディスクリプタ、スーパーディスクリプタ、フォネティックディスクリプタ、またはハイパーディスクリプタの作成にも使用できます。

COUPLE 機能は、2つのファイルについての共通のディスクリプタを追加します (インバーテッドリストを更新します)。2つのファイル間に同一のフォーマットおよび長さで定義してある共通のディスクリプタが存在する場合、その2つのファイルはカップリングすることができます。1つのファイルは、18個まで他のファイルとカップリングが可能ですが、2つのファイル間には同時に1個のカップリング関係だけが許されます。自分自身とのカップリングは行うことはできません。

 **Note:** ファイル番号が 255 以下のファイルだけがカップリングできます。

カップリングしているファイルのディスクリプタのインバーテッドリストのいずれかに変更が行われると、他のファイルは自動的に変更されます。カップリングのベースとなるディスクリプタを更新する際や、カップリングがされているいずれかのファイルにレコードを追加したり削除したりする際に、カップリングリストの更新に必要なオーバーヘッドについて DBA は十分に考慮すべきです。NU オプションで定義されていないフィールドがカップリングのベースとして使用され、かつ、そのフィールドに多数の空値が含まれている場合は、実行時間が長くなったり、カップリングリストを格納するのに大量のディスクスペースが必要となることがあります。

中断された ADAINV オペレーションを再開させる場合には、その前にファイルをリストアしておく必要はありません。



# 103

## COUPLE：定義ファイルカップリングディスク

### リプタ

---

■ 基本パラメータ .....	486
■ オプションパラメータ .....	487
■ 例 .....	488
■ ファイルカップリング用の一時スペース .....	488
■ アソシエータカップリングリスト .....	489
■ スペースカップリングリスト .....	490
■ スペース割り当て .....	491

```
ADAINV COUPLE FILES = file-number1 , file-number2
DESCRIPTOR = 'fieldname , fieldname'
SORTSIZE = size
TEMPSIZE = size
[LPB = { prefetch-buffer-size | ADARUN-lu } ]
[LWP = { workpool-size | 1048576 } ]
[NOUSERABEND]
[PASSWORD = 'password' ]
[SORTDEV = { device-type | ADARUN-device } ]
[TEMPDEV = { device-type | ADARUN-device } ]
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### DESCRIPTOR：カップリングのベースとして使用するディスクリプタ

DESCRIPTOR パラメータでは、ファイルカップリングのベースを提供するため、各ファイルに1つのディスクリプタを定義します。この場合、サブディスクリプタまたはスーパーディスクリプタも使用できます。これらのディスクリプタは、マルチプルバリューフィールドとして定義したり、マルチプルバリューフィールドから派生させることができます。指定されたディスクリプタは、ピリオディックグループ内に含まれている必要はありません。また、ピリオディックグループから派生したものである必要もありません。ディスクリプタには異なる名前を付けることができますが、長さおよびフォーマット定義は同じである必要があります。

### FILES：カップリングするファイル

FILES パラメータは、カップリングする2つのファイルを指定します。各ファイルの番号は255以下である必要があります。現在互いにカップリングされているファイルを指定することはできません。

### SORTSIZE：ソートサイズ

SORTSIZE は、ソートデータセットまたはR1/2データセット（SORTR2はVSE下ではサポートされません）に対して使用できるスペースを指定するものです。値はシリンダ（数値のみ）か、またはブロック（数値に"B"を伴う）で指定します。ブロックで指定する場合には、それらがシリンダの総数に等しくなければなりません。SORTSIZE パラメータは必須です。ソートスペース見積りに関する詳細情報は『Adabas DBA リファレンスマニュアル』を参照してください。

**TEMPSIZE**：一時ストレージのサイズ

TEMPSIZEは、中間データセットで利用できるスペースを定義します。値はシリンダ（数値のみ）か、またはブロック（数値に"B"を伴う）で指定します。このパラメータは必須です。

## オプションパラメータ

---

**LPB**：プリフェッチバッファサイズ

LPBは、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能な最大値は、32,760 バイトです。デフォルト値は、ADARUN LU パラメータに依存し、この LU パラメータがあまりに小さい場合には ADAINV によって指定された LPB 値もよりも少なくなることがあります。

**LWP**：ワークプールサイズ

LWP は、ディスクリプタ値のソートに使用されるワークプールの大きさを指定します。この値は、数値のみでバイト単位で指定するか、あるいは数値の後に "K" を付けてキロバイト単位で指定します。値が指定されない場合は、デフォルトで1048576バイト（または1024K）が設定されます。ただし、ディスクリプタが非常に長い、またはディスクリプタの数が著しく多いファイルに対して ADAINV の実行時間を短くする場合には、より大きな値を LWP に設定すべきです。ソートデータセットとの問題を避けるため、比較的小規模なファイルにディスクリプタを定義する場合には、LWP 値を小さめに指定してください。

最小ワークプールの大きさは、ソートデータセットのデバイスタイプごとに異なります。

ソートデバイス	最小 LWP	最小 LWP
	バイト	キロバイト
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

カップリングするファイルの片方または両方がセキュリティ保護されている場合は、ファイルの有効なパスワードをこのパラメータで指定する必要があります。両方のファイルがパスワード保護されている場合は、両方のパスワードが同じである必要があります。

**SORTDEV**：ソートデバイスタイプ

ADAINV はディスクリプタ値のソートにソーデータセットを使用します。SORTDEV パラメータは、このソートデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定デバイスと異なる場合にのみ必要です。SORTDEV に関する特定の注意事項については、[z/OS ジョブ制御情報](#)を参照してください。

**TEMPDEV**：一時ストレージデバイスタイプ

ADAINV は中間データセットを使用して中間データを格納します。TEMPDEV パラメータは、このデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定デバイスと異なる場合にのみ必要です。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

---

```
ADAINV COUPLE  
FILES=3,4,DESCRIPTOR='AA,BB'
```

ファイル 3 および 4 をカップリングします。ファイル 3 のディスクリプタ AA およびファイル 4 のディスクリプタ BB を、カップリングのベースとして使用します。

## ファイルカップリング用の一時スペース

---

中間データセットは、カップリングするファイルごとに生成されます。

ファイルに含まれるレコードごとに、エントリがデータセットに書き込まれます。各エントリには、レコードの ISN（ファイルに定義された ISNSIZE に応じて 3 または 4 バイト）、およびカップリングのベースとして使用されるディスクリプタの値（圧縮形式）が含まれます。NU オプションを指定して定義されたディスクリプタの場合、ディスクリプタに空値が含まれるレコードに関してはエントリが書き込まれません。ディスクリプタがマルチプルバリューフィールドの場合は、それぞれの値についてエントリが書き込まれます。

それぞれの中間データセットに必要なスペースは、各 Adabas ファイルに含まれているレコード数、および各レコードのカップリングディスクリプタに存在する異なる値の長さと同数の関数で求められます。

次の式を使用して、中間データセットに必要なスペースを計算します。

$$SP = RECS \times UV \times (ISNSIZE + (AVLEN \times 4))$$

ここでは次の内容を表しています。

SP	必要な中間データセットのスペース（バイト単位）。
RECS	カップリングされたファイルに存在するレコード数。
UV	各レコードに存在するディスクリプタのユニーク値の平均個数。NUオプションを指定しないでディスクリプタを定義した場合、UVは1以下になります。NUオプションを指定してディスクリプタを定義した場合、UVは各レコードに存在する値の平均個数から空値を含むレコードの割合を引いた値になります。例えば、各レコードに存在する値の平均個数が1で、値の20%が空値の場合、UVは $1 - 0.2 = 0.8$ になります。
ISNSIZE	ファイルに存在するISN長（3または4バイト）。
AVLEN	ディスクリプタの各値の（圧縮後の）平均長。

### 例：ファイルカップリングに必要な中間スペースの計算

カップリングするファイルに3バイトのISNが存在し、50,000個のレコードが含まれています。カップリングのベースとして使用するディスクリプタには、レコードあたり1つの値（空値なし）が含まれ、値の平均長は5バイトです。

$$\begin{aligned} SP &= 50,000 \times 1 \times (3 + (5 + 1)) \\ SP &= 50,000 \times 9 \\ SP &= 450,000 \text{ bytes} \end{aligned}$$

## アソシエータカップリングリスト

ADAINVでは、2つのリストのマッチング、結果として得られる各リストのソート、アソシエータカップリングリストへの各リストの書き込みが行われます。

中間データセットには、ファイルごとにマッチング（カップリング）されたISNが格納されます。検出されたマッチングごとに、エントリが中間データセットに書き込まれます。エントリには、マッチング値が存在する各レコードのISNが含まれます。

ADAINVでは、ソートエリアを使用して中間データセットに格納されたエントリがソートされ、ソートされたエントリがファイルAのアソシエータカップリングリストに書き込まれます。ファイルBについても同じ処理が繰り返されます。

中間エリアに必要なサイズは、ファイルのカップリングに使用するディスクリプタに対して、2つのファイルに存在するマッチング値の個数によって異なります。ファイルに定義されたISNSIZEに応じて、それぞれのマッチングに6または8バイト必要です。

一般的に、中間エリアに必要なスペースの2倍の量がソートエリアに必要です。

ファイルカップリングは階層的ではなく双方向です。つまり、もう片方のファイルにカップリングされる ISN が存在するリストごとに、2つのカップリングリストが作成されます。

### 例：カップリングリスト

カップリングする2つのファイルにディスクリプタ AA および BB がそれぞれ含まれているとします。各ファイルの最初の5レコードの値は、次のようになります。

ファイル A		ファイル B	
ISN	フィールド AA の値	ISN	フィールド BB の値
1	20	1	18
2	25	2	40
3	27	3	25
4	30	4	20
5	40	5	20

カップリングのベース AA および BB を使用して2つのファイルをカップリングした結果、カップリングリストは次のようになります。

ファイル A			ファイル B		
ISN in FILE B*	COUNT	COUPLED ISN	ISN in FILE A*	COUNT	COUPLED ISN
2	1	5	1	2	4,5
3	1	2	2	1	3
4	1	1	5	1	2
5	1	1			

\* カップリングされたレコードの個数と ISN を判断するため、内部的にはこのフィールドが Adabas でディスクリプタのように使用されます。

### スペースカップリングリスト

カップリングリストに必要な合計スペースは、カップリングのベースとして使用される2つのディスクリプタ間に存在する共通値の個数によって異なります。

それぞれの共通値のスペース要件は、次のように見積もることができます。

$$SP = 4a + 4b + 6ab$$

ここでは次の内容を表しています。

SP	1つの共通値のスペース要件（バイト単位）
a	ファイル A に存在する共通値を持つレコード数
b	ファイル B に存在する共通値を持つレコード数

カップリングリストに必要な合計スペースは、それぞれの共通値のスペース要件の合計になります。

前に定義したサンプルファイル A および B を使用すると、共通値ごとのスペース要件は次のようになります。

共通値	スペース要件
20	$SP = 4(2) + 4(1) + 6(2 \cdot 1) = 24$ バイト
25	$SP = 4(1) + 4(1) + 6(1 \cdot 1) = 14$ バイト
40	$SP = 4(1) + 4(1) + 6(1 \cdot 1) = 14$ バイト

必要な合計スペース =  $24 + 14 + 14 = 52$  バイト

### 例：カップリングリストのスペース要件

フィールド ID で2つのファイルのカップリングしているとします。ID の値は各ファイルでユニークです。カップリングされたファイルには 5,000 個の共通値があります。

共通値	スペース要件
n	1つの共通値に対して $SP = 4(1) + 4(1) + 6(1) SP = 14$ バイト

共通値は 5,000 個あり、それぞれに 14 バイト必要です。カップリングリストに必要な合計スペースは 70,000 バイトになります。

## スペース割り当て

ADAINV で作成されたカップリングリストは、カップリングされたファイルごとにノーマルインデックス (NI) およびアップパーインデックス (UI) に含まれます。現在ファイルに割り当てられている NI または UI コンポーネントの論理エクステントが ADAINV の実行中にすべて使用されると、追加エクステントをコンポーネントに割り当てようとしています。割り当てられるエクステントのサイズは、コンポーネントに現在割り当てられているすべての論理エクステントの現在の合計サイズの 25 % です。スペースが不足しているか、またはコンポーネントに割り当て済みのエクステントが最大数に達しているか、ADAINV はエラーメッセージを出力して終了します。



# 104 INVERT : ディスクリプタの作成

---

■ 基本パラメータ .....	494
■ オプションのパラメータとサブパラメータ .....	495
■ INVERT 機能のスペース割り当て .....	497
■ 例 .....	498

INVERT 機能はファイルに含まれる既存のフィールドに対して、ディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタまたはファイル内に照合ディスクリプタを作成します。ADAINVINVERTの1回の実行で複数のディスクリプタを作成できますが、対象ファイルは1つのみです。

```
ADAINV INVERT FILES = file-num
SORTSIZE = size
TEMPSIZE = size
[FIELD = 'field-name [ , option]... ' ] ...
[COLDE = 'num, name [ , UQ [ , XI ] ] = parent-field' ]
[HYPDE = 'num, name, length, format [ , option ]... = parent-field, ...' ]
[PHONDE = 'name (field-name)' ]
[SUBDE = 'name [ , UQ [ , XI ] ] = parent-field (begin, end)' ]
[SUPDE = 'name [ , UQ [ , XI ] ] = {parent-field (begin, end)}, ...' ]
[CODE = cipher-code ]
[LPB = { prefetch-buffer-size | ADARUN-lu } ]
[LWP = { workpool-size | 1048576 } ]
[NOUSERABEND]
[PASSWORD = 'password' ]
[SORTDEV = {device-type | ADARUN-device } ]
[TEMPDEV = {device-type | ADARUN-device } ]
[TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILE パラメータは、作成するディスクリプタが含まれているファイルを指定します。

### SORTSIZE：ソートサイズ

SORTSIZE は、ソートデータセットまたはR1/2データセット（SORTR2はVSE下ではサポートされません）に対して使用できるスペースを指定するものです。値はシリンダ（数値のみ）か、またはブロック（数値に"B"を伴う）で指定します。ブロックで指定する場合には、それらがシリンダの総数に等しくなければなりません。SORTSIZEパラメータは必須です。ソートスペース見積りに関する詳細情報は『Adabas DBA リファレンスマニュアル』を参照してください。

### TEMPSIZE：一時ストレージのサイズ

TEMPSIZE は、中間データセットで利用できるスペースを定義します。値はシリンダ（数値のみ）か、またはブロック（数値に"B"を伴う）で指定します。このパラメータは必須です。

## オプションのパラメータとサブパラメータ

### CODE：サイファコード

FILE パラメータで指定したファイルが暗号化されている場合、サイファコードを CODE パラメータで指定しなければなりません。

### FIELD/ COLDE/ HYPDE/ PHONDE/ SUBDE/ SUPDE：ディスクリプタの定義

これらのパラメータは、各種タイプのディスクリプタの定義に使用できます。指定されたファイルに対して、ディスクリプタ定義を少なくとも1つ指定する必要があります。このとき、複数のディスクリプタまたは複数タイプのディスクリプタを指定できます。

1つ以上のフィールドをディスクリプタとして定義するには FIELD パラメータを使用し、照合ディスクリプタには COLDE パラメータ、ハイパーディスクリプタには HYPDE パラメータ、フォネティックディスクリプタには PHONDE パラメータ、サブディスクリプタには SUBDE パラメータ、スーパーディスクリプタには SUPDE パラメータをそれぞれ使用します。

FIELD 指定がある場合は、照合ディスクリプタ、ハイパーディスクリプタ、スーパーディスクリプタ、サブディスクリプタ、フォネティックディスクリプタ指定の前に置く必要があります。

FIELD パラメータは、インバートする既存のフィールドを指定します。フィールドはエレメンタリフィールドでもマルチプルバリュースフィールドでもよく、また（フィールドが FI オプションで定義されていない限り）ピリオディックグループ内のものでも構いません。

ディスクリプタフィールドをユニーク指定したい場合は、フィールド名に続けて "UQ" を指定します。ディスクリプタの一意性を、排除されるインデックス（オカレンス番号）で決定する必要がある場合は、同様に、"XI" も指定してください。



**Note:** Adabas 拡張ファイルの場合、ADAINV は、指定されたコンポーネントファイル内でしかユニークディスクリプタ違反を検出できません。同じ値のユニークディスクリプタが他のいずれかのコンポーネントファイルに存在している場合は、ADAINV で検出できません。したがって、ユニークディスクリプタの値が、全拡張ファイルを通じてユニークであるようにする必要があります。

FIELD パラメータではインバートに複数のフィールドを指定できますが、このパラメータのインスタンスごとに定義されるのは、1つの照合ディスクリプタ、ハイパーディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、またはフォネティックディスクリプタのみに限定されます。ADAINV の実行ごとに、パラメータの複数のインスタンスを指定できます。

サブフィールドまたはスーパーフィールドをインバートする場合は、フィールドの作成時と同じ親フィールドを SUBDE または SUPDE で指定しないとエラーになります。開始および終了の値は元のフィールド定義から取得されます。

親フィールドをNUオプションで指定すると、空値フィールドを含むレコードについてはインバーテッドリストにエントリが作成されません。スーパーディスクリプタとハイパーディスクリプタは、他のディスクリプタエレメントの値に関係なく、エントリが作成されません。

親フィールドが初期化されず、論理的に物理レコードの終了を超えた場合、そのレコードのインバーテッドリストエントリは、パフォーマンスの理由から生成されません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

個々のディスクリプタの構文については、ADACMPユーティリティに関する説明で、「[フィールド定義ステートメント](#)」を参照してください。

**LPB：プリフェッチバッファサイズ**

LPBは、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能最大値は、32,760バイトです。デフォルト値は、ADARUNLUパラメータに依存し、このLUパラメータがあまりに小さい場合にはADAINVによって指定されたLPB値もよりも少なくなることがあります。

**LWP：ワークプールサイズ**

LWPは、ディスクリプタ値のソートに使用されるワークプールの大きさを指定します。この値は、数値のみでバイト単位で指定するか、あるいは数値の後に"K"を付けてキロバイト単位で指定します。値が指定されない場合は、デフォルトで1048576バイト（または1024K）が設定されます。ただし、ディスクリプタが非常に長い、またはディスクリプタの数が著しく多いファイルに対してADAINVの実行時間を短くする場合には、より大きな値をLWPに設定すべきです。ソートデータセットとの問題を避けるため、比較的小規模なファイルにディスクリプタを定義する場合は、LWP値を小さめに指定してください。

最小ワークプールサイズは、SORTデータセットのデバイスタイプに応じて異なります。

ソートデバイス	最小LWP	最小LWP
	バイト	キロバイト
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

**NOUSERABEND：アベンドなし終了**

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ

ラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

FILEパラメータで指定されたファイルがセキュリティ保護されている場合は、ファイルのパスワードをこのパラメータで指定する必要があります。

**SORTDEV**：ソートデバイスタイプ

ADAINV はディスクリプタ値のソートにソーデータセットを使用します。SORTDEVパラメータは、このソートデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定デバイスと異なる場合にのみ必要です。z/OS SORTDEV に関する特定の注意事項については、このセクションの最後にある z/OS ジョブ制御情報を参照してください。

**TEMPDEV**：一時ストレージデバイスタイプ

ADAINV は中間データセットを使用して中間データを格納します。TEMPDEVパラメータは、このデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定デバイスと異なる場合にのみ必要です。

**TEST**：構文テスト

TESTパラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## INVERT 機能のスペース割り当て

インバート中のフィールド値および値を含むレコードの ISN は、インバーテッドリスト（ノーマルインデックスおよびアップパーインデックス）に書き込まれます。

ノーマルインデックスまたはアップパーインデックスのいずれかの論理エクステントが ADAINV の実行中にすべて使用されると、ADAINVによって追加エクステントが割り当てられます。割り当てられるエクステントのサイズは、ファイルに現在割り当てられているすべてのノーマルインデックスエクステントの現在の合計サイズの 25 % です。

新しいエクステントのスペースが不足しているか、割り当て済みのエクステントが最大数に達している場合、ADAINV はエラーメッセージを出力して終了します。

## 例

### 例 1

```
ADAINV INVERT  
FILE=3, FIELD='AR', TEMPSIZE=10, SORTSIZE=5
```

ファイル 3 のフィールド AR をディスクリプタとして指定します。

### 例 2

```
ADAINV INVERT FILE=5, SUBDE='SA=AA(1,4)'  
ADAINV TEMPSIZE=6, SORTSIZE=3
```

ファイル 5 のフィールド AA（位置 1～4）を親フィールドとして使用して、サブディスクリプタ SA を作成します。

### 例 3

```
ADAINV INVERT FILE=6, SUPDE='SB=AA(1,4),AB(1,1)'  
ADAINV TEMPSIZE=5, SORTSIZE=3
```

ファイル 6 のフィールド AA（位置 1～4）および AB（位置 1）を使用して、スーパーディスクリプタ SB を作成します。

### 例 4

```
ADAINV INVERT FILE=1, PHONDE='XX(AA)'  
ADAINV TEMPSIZE=5, SORTSIZE=3
```

フィールド AA をソースフィールドとして使用して、フォネティックディスクリプタ XX を作成します。

### 例 5

```
ADAINV INVERT FILE=6, COLDE='1, Y1=AA'  
ADAINV TEMPSIZE=5, SORTSIZE=4
```

フィールド AA をソースフィールドとして使用して、Y1 という名前の照合ディスクリプタ CDX=01 を作成します。

# 105 JCL/JCS の必要項目と指定例

---

■ ユーザー出口を使用した照合 .....	500
■ BS2000 .....	500
■ z/OS .....	503
■ z/VM .....	505
■ VSE .....	507

ここでは、BS2000、z/OS、z/VM、および VSE システムにおける ADAINV の実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## ユーザー出口を使用した照合

ADAINV の実行中に照合ユーザー出口を使用する場合は、ユーティリティの実行で ADARUN CDXnn パラメータを指定する必要があります。

UES（ユニバーサルエンコーディングサポート）と合わせて使用する場合の照合ディスクリプタユーザー出口の構文は次のとおりです。

```
ADARUN CDXnn= exit-name
```

ここでは次の内容を表しています。

nn	照合ディスクリプタ出口の番号。01 から 08 までの 2 桁の 10 進数。
exit-name	照合ディスクリプタ出口で制御を持つユーザールーチン名。最大 8 文字。

照合ディスクリプタ出口に指定できるプログラムは 1 つだけです。最大 8 個の照合ディスクリプタ出口を任意の順序で指定できます。詳細は『Adabas DBA リファレンスマニュアル』を参照してください。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
中間ストレージ	DDTEMPR1	ディスク	
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大きなファイルを使用する場合、ソートエリアは 2 つのボリュームに分割してください（注を参照）。
リカバリログ（RLOG）	DDRLOGR1	ディスク	リカバリログオプションの使用時に必要
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAINV パラメータ	SYSDTA/DDKARTE		

データセット	リンク名	ストレージ	追加情報
ADARUN メッセージ	SYSOUT/DDPRINT		メッセージおよびコード
ADAINV メッセージ	SYSLST/DDDRUCK		メッセージおよびコード

 **Note:** 大規模なファイルをソートする場合、ソートデータセットが2つのボリュームに分割されているとパフォーマンスが向上することがあります。データセットが2つ指定されている場合は、両方が同じデバイスタイプ（SORTDEV パラメータ）である必要があり、また各々 SORTSIZE パラメータで指定した大きさの正確に半分のサイズにする必要があります。

## ADAINV JCL の例（BS2000）

### カップリングファイル

SDF フォーマットの場合：

```
/.ADAINV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A I N V COUPLE FIELD (REFLECTIVE)
/REMARK *
/ASS-SYSLST L.INV.COUP
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDTEMPR1,ADyyyyyy.TEMP
/SET-FILE-LINK DDSORTR1,ADyyyyyy.SORT
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAINV,DB=yyyyyy,IDTNAME=ADABAS5B
ADAINV COUPLE FILE=1,3,DESCRIPTOR= AA,AA
ADAINV TEMPSIZE=100,SORTSIZE=50
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADAINV LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A I N V COUPLE FIELD (REFLECTIVE)
/REMARK *
/SYSFILE SYSLST=L.INV.COUP
/FILE ADA.MOD,LINK=DDLIB
/FILE ADyyyyyy.ASSOR,LINK=DDASSOR1,SHARUPD=YES
/FILE ADyyyyyy.TEMP,LINK=DDTEMPR1
/FILE ADyyyyyy.SORT,LINK=DDSORTR1
```

```
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAINV,DB=yyyyy,IDTNAME=ADABAS5B
ADAINV COUPLE FILE=1,3,DESCRIPTOR= AA,AA
ADAINV TEMPSIZE=100,SORTSIZE=50
/LOGOFF NOSPOOL
```

### ファイルのインバート

SDF フォーマットの場合：

```
/.ADAINV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A I N V INVERT FIELD (REFLECTIVE)
/REMARK *
/ASS-SYSLST L.INV.INVE
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrS.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDTEMPR1,ADAYyyyyy.TEMP
/SET-FILE-LINK DDSORTR1,ADAYyyyyy.SORT
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAINV,DB=yyyyy,IDTNAME=ADABAS5B
ADAINV INVERT FILE=1
ADAINV TEMPSIZE=100,SORTSIZE=50
ADAINV FIELD= AC
ADAINV SUPDE= S1,UQ=AA(1,3),AD(2,4)
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADAINV LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A I N V INVERT FIELD (REFLECTIVE)
/REMARK *
/SYSFILE SYSLST=L.INV.INVE
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyyy.ASSOR ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyyy.TEMP ,LINK=DDTEMPR1
/FILE ADAYyyyyy.SORT ,LINK=DDSORTR1
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAINV,DB=yyyyy,IDTNAME=ADABAS5B
ADAINV INVERT FILE=1
ADAINV TEMPSIZE=100,SORTSIZE=50
ADAINV FIELD= AC
ADAINV SUPDE= S1,UQ=AA(1,3),AD(2,4)
```

/LOGOFF NOSPOOL

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
中間ストレージ	DDTEMPR1	ディスク	
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大きなファイルを使用する場合、ソートエリアは2つのボリュームに分割してください（注を参照）。
リカバリログ (RLOG)	DDRLOGR1	ディスク	リカバリログオプションの使用時に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAINV パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAINV メッセージ	DDDRUCK	プリンタ	メッセージおよびコード



**Note:** 大規模なファイルをソートする場合、ソートデータセットが2つのボリュームに分割されているとパフォーマンスが向上することがありますが、OSではこの操作は困難です。代わりに2つのソートデータセットを指定できます。どちらも同じデバイスタイプ（SORTDEV パラメータ）で定義しなければならず、また各々 SORTSIZE パラメータで指定した大きさの正確に半分の大きさでなければなりません。

\*

## ADAINV JCL の例 (OS)

## カップリングファイル

この例については、MVSJOBS データセットの ADAINVCO を参照してください。

```
//ADAINVCO JOB
//*
//*   ADAINV:   COUPLE FILES
//*
//INV      EXEC PGM=ADARUN
//STEPLIB DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//*
//DDASSOR1 DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <===== ASSO
//DDDATAR1 DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <===== DATA
```

```
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <===== WORK
//DDTEMPR1 DD DISP=OLD,DSN=EXAMPLE.DByyyyy.TEMPR1 <===== TEMP
//DDSORTR1 DD DISP=OLD,DSN=EXAMPLE.DByyyyy.SORTR1 <===== SORT
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAINV,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAINV COUPLE FILE=2,3,DESCRIPTOR='BB,BB'
ADAINV TEMPSIZE=100,SORTSIZE=100
/*
```

### ファイルのインバート

この例については、MVSJOBS データセットの ADAINV を参照してください。

```
//ADAINVDE JOB
/*
/* ADAINV: INVERT A FIELD TO A DE
/*
//INV EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <===== ASSO
//DDTEMPR1 DD DISP=OLD,DSN=EXAMPLE.DByyyyy.TEMPR1 <===== TEMP
//DDSORTR1 DD DISP=OLD,DSN=EXAMPLE.DByyyyy.SORTR1 <===== SORT
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAINV,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAINV INVERT FILE=1
ADAINV FIELD='AC'
ADAINV SUPDE='S1,UQ=AA(1,3),AD(2,4)'
```

```
ADAINV TEMPSIZE=100 ,SORTSIZE=100
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
中間ストレージ	DDTEMPR1	ディスク	
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大きなファイルを使用する場合は、ソートエリアを2つのボリュームに分割してください。*
リカバリログ (RLOG)	DDRLOGR1	ディスク	リカバリログオプションの使用時に必要
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAINV パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAINV メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

\* 大きいファイルをソートする場合、ソートデータセットが2つのボリュームに分割されているとパフォーマンスが向上することがありますが、CMS ではこの操作は困難です。代わりに2つのソートデータセットを指定できます。どちらも同じデバイスタイプ (SORTDEV パラメータ) で定義しなければならず、また各々 SORTSIZE パラメータで指定した大きさの正確に半分の大きさでなければなりません。

### ADAINV JCL の例 (z/VM)

カップリングファイル

```
DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDTEMPR1,DSN=ADABASV.V.TEMP,VOL=TEMPV1
DATADEF DDSORTR1,DSN=ADABASV.V.SORT,VOL=SORTV1
DATADEF DDPRINT,DSN=ADAINV.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAINV.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNINV.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAINV.CONTROL,MODE=A
ADARUN
```

RUNINV CONTROL A1 の内容

```
ADARUN  PROG=ADAINV,DEVICE=dddd,DB=yyyyyy
```

### ADAINV CONTROL A1 の内容

```
ADAINV  COUPLE  FILE=1,3,DESCRIPTOR='AA,AA'  
ADAINV          TEMPSIZE=100,SORTSIZE=50  
*
```

### ファイルのインバート

```
DATADef  DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1  
DATADef  DDTEMPR1,DSN=ADABASVv.TEMP,VOL=TEMPV1  
DATADef  DDSORTR1,DSN=ADABASVv.SORT,VOL=SORTV1  
DATADef  DDPRINT,DSN=ADAINV.DDPRINT,MODE=A  
DATADef  DUMP,DUMMY  
  
DATADef  DDRUCK,DSN=ADAINV.DDDRUCK,MODE=A  
DATADef  DDCARD,DSN=RUNINV.CONTROL,MODE=A  
DATADef  DDKARTE,DSN=ADAINV.CONTROL,MODE=A  
ADARUN
```

### RUNINV CONTROL A1 の内容

```
ADARUN  PROG=ADAINV,DEVICE=dddd,DB=yyyyyy
```

### ADAINV CONTROL A1 の内容

```
ADAINV  INVERT  FILE=1  
ADAINV          TEMPSIZE=100,SORTSIZE=50  
*  
ADAINV          FIELD='AC'  
ADAINV          SUPDE='S1,UQ=AA(1,3),AD(2,4)'
```

## VSE

ファイル	ファイル名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	
中間ストレージ	TEMPR1	ディスク	*	
ソートエリア	SORTR1	ディスク	*	
リカバリログ (RLOG)	RLOGR1	ディスク	*	リカバリログ (RLOG) オプション が必要
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADAINV パラメータ	-	リーダー	SYSIPT	
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADAINV メッセージ	-	プリンタ	SYS009	メッセージおよびコード

\* あらゆるプログラマ論理ユニットが使用できます。

### ADAINV JCS の例 (VSE)

VSE プロシージャ (PROC) の内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

カップリングファイル

この例については、メンバ ADAINVCO.X を参照してください。

```
* $$ JOB JNM=ADAINVCO,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAINVCO
*      COUPLE FILES
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAINV,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyyy
/*
ADAINV COUPLE FILE=2,3,DESCRIPTOR='BB,BB'
ADAINV      TEMPSIZE=100,SORTSIZE=100
/*
/&
* $$ EOJ
```

ファイルのインバート

この例については、メンバ ADAINV.X を参照してください。

```
* $$ JOB JNM=ADAINV,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAINV
*      INVERT A FIELD TO A DESCRIPTOR
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAINV,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAINV INVERT FILE=1
ADAINV FIELD='AC'
ADAINV SUPDE='S1,UQ=AA(1,3),AD(2,4)'
ADAINV TEMPSIZE=100,ORTSIZE=100
/*
/&
* $$ EOJ
```

# 106

## ADALOD ユーティリティ：ファイルローダー

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- **LOAD**：ファイルのロード
- **UPDATE**：レコードの追加／削除
- ローダーストレージの要件および使用
- 中間データセットスペースの使用状況
- **ADALOD** スペース／統計レポート
- **JCL/JCS** の必要項目と指定例



# 107 機能概要

---

**ADALODLOAD 機能**は、ファイルをデータベースにロードします。ADACMPまたはADAULDユーティリティで生成された圧縮レコードをこのユーティリティの入力に使用できます。

ADALODはデータストレージに圧縮レコードをロードし、ファイルのアドレスコンバータを作成し、ファイルのフィールド定義をフィールド定義テーブル (FDT) に格納します。また、ADALODはファイルに定義されている全ディスクリプタの値と、その値が存在する全レコードのISNを抜き出し、中間データセットに書き出します。その後、このデータセットは値およびISNの順にソートされ、アソシエータインバーテッドリストに書き込まれます。

**ADALOD UPDATE 機能**は、大量のレコードをAdabasファイルに追加したり、Adabasファイルから削除したりする際に使用できます。UPDATE機能は、Adabasのレコード追加/削除コマンドを繰り返し実行するの比べ、処理時間が大幅に短縮されます。追加するレコードは、ADACMPまたはADAULDユーティリティで生成された圧縮レコードを使用します。削除するレコードのISNは、入力データセットに用意するか、またはコントロールステートメントに指定することができます。

1回のADALOD実行中に、レコードの追加と、別のレコードの削除を行うことができます。



**Note:** Adabas 5.1～5.3、6.1、6.2、7.1、7.2、または7.4で動作しているADAULDまたはADACMPで作成された入力データセットからファイルをロードできます。また、ファイルのロードに使用するバージョン以降に組み込まれた新機能を、ロードするファイルで使用していない限り、Adabas 8ファイルをリリース5.3以降のAdabasデータベースにロードすることもできます。



# 108

## LOAD：ファイルのロード

---

■ 基本パラメータ .....	515
■ オプションのパラメータとサブパラメータ .....	517
■ 例 .....	530
■ LOAD データとスペースの要件 .....	531
■ 拡張ファイルのロード .....	535
■ マルチクライアントファイルのロード .....	537

```

ADALOD LOAD FILE = file-number [ , filetype ]
DSSIZE = size
MAXISN = max-number-of-records [MAXISN2 = max-number-of-secondary-spanned-records ]
SORTSIZE = size
TEMPSIZE = size
[ACRABN = starting-rabn ] [AC2RABN = starting-rabn ]
[ADAMFILE ADAMDE = {field | ISN } [ADAMOFLOW = size ] [ADAMPARM = {number | 0 }]]
[ALLOCATION = { FORCE | NOFORCE } ]
[ANCHOR = file-number MINISN = lowest-allocated-isn, NOACEXTENSION ]
[ASSOPFAC = {padding-factor | 10 } ] [ASSOVOLUME = 'Associator-extent-volume' ]
[{BASEFILE | LOBFILE} = file-number ]
[DATAFRM = { YES | NO } ]
[DATAPFAC = { padding-factor | 10 } ] [DATAVOLUME = 'Data-Storage-extent-volume' ]
[DSDEV = device-type ] [DSRABN = start-rabn ] [DSREUSE = { YES | NO } ]
[ETID = owner-id ]
[IGNFDT]
[INDEXCOMPRESSION = { YES | NO } ]
[ISNREUSE = { YES | NO } ] [ISNSIZE = { 3 | 4 } ]
[LIP = { isn-pool-size | 2000 } ]
[LOWNERID = { owner-id-length | 0 } ]
[LWP = { work-pool-size | 1048576 } ]
[MAXDS = {max-DS-secondary-allocation | no-limit } ]
[MAXNI = { max-NI-secondary-allocation | no-limit } ]
[MAXRECL = {max-compressed-record-length | max-possible } ]
[MAXUI = {max-UI-secondary-allocation | no-limit } ]
[MINISN = {lowest-allocated-isn | 1 } ]
[MIXDSDEV]
[NAME = {name | TESTFILE } ]
[NIRABN = start-rabn ] [NISIZE = size ]
[NOACEXTENSION]
[NOUSERABEND]
[NUMREC = {max-number-of-records-to-load | all-records } ]
[PGMREFRESH = { YES | NO } ]
[RESTART]
[SKIPREC = {number | 0 } ]
[SORTDEV = {device-type | ADARUN-device } ]
[TEMPDEV = {device-type | ADARUN-device } ]
[TEST]
[UIRABN = start-rabn ] [UIsize = size ]
[UQDE = descriptor-list ]
[USERISN = { YES | NO } ]
[VERSION = { 4 | 5 | 6 | 7 } ]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### DSSIZE：データストレージのエクステントサイズ

DSSIZE では、ファイルのデータストレージの論理エクステントに対して割り当てるブロック数またはシリンダ数を指定します。この値は指定する必要があります。ブロック値の後に "B" を付加する必要があります ("5000B" など)。

この値として、ADACMP ユーティリティで作成されるスペース要件レポートの値を直接使用することができます。指定されたエクステントサイズが最大空きスペースのサイズを超える場合には、要求を満たすのに必要なファイルエクステント（最大 5 個）が ADALOD によって割り当てられます。

当初ロードするレコード数が少なく、後で大量のレコードを追加する場合は、追加する合計レコード数に比例して ADACMP レポートの値を増やす必要があります。このようにしない場合は、データストレージ（元のデータストレージと 4 つの追加エクステント）のスペース割り当てが追加レコードに対して不足することがあります。この場合、データストレージのスペース割り当てを増やすには、ファイルをアンロードしてからリロード（またはリオーダ）する必要があります。詳細は、「[LOAD データとスペースの要件](#)」の「LOAD ファイルのスペース割り当て」を参照してください。

### FILE：ファイル番号、ファイルタイプ

FILE は、Adabas ファイル番号およびファイルに割り当てるファイルタイプを指定します。

データベース内の別のファイルに現在割り当てられている番号を指定することはできません。このことを行うには、KEEPFDT パラメータ（ADADBSDELETE 機能を参照）を使用して、先にファイルを削除する必要があります。データベースで定義されている最大ファイル番号より大きい番号を指定することはできません。チェックポイントファイル、セキュリティファイル、トリガファイル、またはシステムファイルについては、5000 以下の番号を指定する必要があります。ファイル番号は任意の順序で割り当てることができます。

ファイルタイプはオプションであり、ファイルが Adabas システムファイルまたは Adabas LOB ファイルのどちらであることを示します。次のいずれかのキーワードを指定できます。

CHECKPOINT	Adabas チェックポイントシステムファイル
LOB	Adabas LOB ファイル
SECURITY	Adabas セキュリティシステムファイル
SYSFILE	Adabas システムファイル
TRIGGER	Adabas トリガシステムファイル



#### Notes:

1. ADADEF ユーティリティを使用して作成された既存のチェックポイントファイルを上書きすることはできません。

2. Adabas Security を使用する場合は、セキュリティファイルが必要です。
3. (ATM) データベースでは、SYSFILE 5 および 6 は ATM ニュークリアス用に予約されています。Adabas Transaction Manager Adabas バージョン 7.1 では、このファイル番号を変更することはできません。Adabas の以降のバージョンでは、ファイル番号に柔軟性があります。
4. ATM データベース (ADARUNDTP=TM) で ATM システムファイルをロードするには、パラメータ ADALOD LOAD FILE=5,SYSFILE、ADALOD LOAD FILE=6,SYSFILE を使用します。
5. CHECKPOINT、SECURITY、または TRIGGER を指定すると、//DDEBAND の内容が無視されます。
6. 空の LOB ファイルをロードしている場合、//DDEBAND データセットは不要です。
7. CHECKPOINT、SECURITY、SYSFILE の各ファイルは、唯一の Adabas ユーザーとして稼動している ADADBS DELETE 機能のみが削除できます。システムファイルを削除すると、削除が完了した時点で Adabas は終了します。
8. Adabas では、最大 8 個のシステムファイルを使用できます。
9. LOB ファイルをロードしている場合は、ADALOD LOAD の実行でパラメータ ADAMFILE、ANCHOR、LOWNERID、NUMREC、SKIPREC、および UQDE を指定することはできません。

#### MAXISN：最大 ISN カウント

MAXISN パラメータは必須です。アドレスコンバータ (AC) における ISN マッピングの最大数を指定します。ADALOD は、AC に対してスペースを確保するため、次の計算式を使用して ISN マッピングの数を決定します。

$$(MAXISN - MINISN) + 1$$

デフォルト値はありません。

指定する MAXISN および MINISN 値は、ADALOD の実行中に割り当てられる AC ブロックの初期数を計算するために使用されます。データベースの RABN サイズ (ADADEF DEFINE パラメータ RABNSIZE によって決定される) に応じて、RABN ごとに AC 内に 3 または 4 バイトが必要です。また、各 AC ブロックのブロックサイズはアソシエータのデバイスタイプによって異なります。したがって、割り当てべき AC ブロック数は、有効な ISN マッピング数、RABN サイズ、およびアソシエータデバイスのブロックサイズの影響を受けます。

割り当てを必要とする AC ブロック数を計算するため、ADALOD は次の計算式を使用し、計算結果を最も近い整数に切り上げます。

$$(\#-of-ISN-mappings \times RABN-size) / device-blocksize$$

例えば、データベースの RABN サイズが "3" (ADADEF DEFINE RABNSIZE パラメータ) に設定され、アソシエータが存在するデバイスのブロックサイズが 2544 バイトであるとし、MAXISN=1000 および MINISN=1 が指定されると、マッピングする ISN の実際の個数は  $(1000-1)+1$  (MAXISN MINISN +1) つまり 1000 になります。1000 に 3 (RABN サイズ)

を掛けて、3000 バイトが求められます。最後に、3000 を 2544（デバイスのブロックサイズ）で割り、その結果の値である約 1.18 を切り上げ 2 となります。したがって、この ADALOD の実行には、AC ブロックを 2 つ割り当てる必要がありますこととなります。（対応する ADAREP レポートでは、"MAX-ISN の予期値" が 1000 と表示されず、2 つの AC ブロックに収まる実際の ISN 数、つまりこの場合は約 1694 と表示されます。）

ロードするレコードが  $(MAXISN - MINISN) + 1$  を超えた場合に NOACEXTENSION が指定されていないと、ADALOD は MAXISN 値を増やして追加の AC エクステントを割り当てます。

MAXISN は、ファイルにロードできる最大レコード数を指定しません。Adabas で許可されるファイル内の最大レコード数は、ファイル内の ISN 長が 3 または 4 バイトのどちらであるかを指定する ISNSIZE パラメータによって異なります。（ISNSIZE=3 の場合は 16,777,215 個までのレコードが有効であり、ISNSIZE=4 の場合は 4,294,967,294 個までのレコードが有効です。）

#### **SORTSIZE**：ソートサイズ

SORTSIZE は、ソートデータセットまたは R1/2 データセット（SORTR2 は VSE 下ではサポートされません）に対して使用できるスペースを指定するものです。値はシリンダ（数値のみ）か、またはブロック（数値に "B" を伴う）で指定します。ブロックで指定する場合には、それらがシリンダの総数に等しくなっていなければなりません。SORTSIZE パラメータは必須です。ソートスペース見積りに関する詳細情報は『Adabas DBA リファレンスマニュアル』を参照してください。

#### **TEMPSIZE**：一時ストレージのサイズ

TEMPSIZE はファイルの中間データセットのサイズを指定します。中間サイズは、ファイル内のディスクリプタごとに必要な TEMP スペースの合計です。詳細は、「[LOAD データとスペースの要件](#)」の「LOAD ファイルのスペース割り当て」を参照してください。サイズはシリンダ単位またはブロック単位（後ろに "B" を付加）で指定できます。

## オプションのパラメータとサブパラメータ

#### **ACRABN/ AC2RABN/DSRABN/ NIRABN/ UIRABN**：開始 RABN

アドレスコンバータ（ACRABN）、セカンダリアドレスコンバータ（AC2RABN）、データストレージ（DSRABN）、ノーマルインデックス（NIRABN）、またはアッパーインデックス（UIRABN）のスペース割り当てを、指定された RABN で開始します。

#### **ADAMFILE**：ADAM オプションを使用してロードするファイル

ADAMFILE は、ADAM オプションを使用してファイルをロードすることを指定します。

このパラメータを指定すると、各入力レコードのデータストレージ RABN がランダムマイジングアルゴリズムを使って計算され、各レコードの ADAM ディスクリプタの値に基づいて結果が求められます。ADAM オプションの使用に関する詳細は、ADAMER ユーティリティの説明を参照してください。ADAMFILE を指定すると、ADAMDE の指定も必要です。



**Note:** LOB ファイル (**FILE パラメータ**を参照) をロードしている場合は、ADALOD LOAD の実行で ADAMFILE パラメータを指定できません。

#### ADAMDE：ADAM キー

ADAMDE は、ADAM キーとして使用するフィールドを指定します。

ADAM ディスクリプタをフィールド定義テーブル (FDT) で定義する必要があります。UQ オプションを使用してディスクリプタを定義しておく必要がありますが、その場合、以下は無効です。

- サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、照合ディスクリプタ、またはフォネティックディスクリプタ
- マルチプルバリューフィールド
- ピリオディックグループ内のフィールド
- 可変長
- 空値省略 (NU) オプションの指定

レコードの ISN を ADAM キーとして使用する場合は、ADAMDE = ISN を指定する必要があります。

ADAMFILE パラメータを指定してロードするファイルに対して ADAM オプションを選択した場合は、このパラメータを指定する必要があります。

#### ADAMOFLOW：ADAM ファイルのオーバーフローエリアサイズ

ADAMOFLOW は、ADAM ファイルのオーバーフローに使用するデータストレージエリアのサイズを指定します。ADAMOFLOW 値は、ロードするファイルに対して ADAM オプションを選択した場合にのみ有効です (ADAMFILE パラメータを参照)。

ADALOD は、DSSIZE から ADAMOFLOW (ブロック単位) を引いた値より小さい素数を選択します。この値は、レコードごとのデータストレージ RABN を計算するために使用されます。計算された RABN のブロックにレコードが収まらなると、オーバーフローエリアの次の空き RABN に書き込まれます。

#### ADAMPARM：ADAM ファイルのビット切り捨て

ADAMPARM は、ADAM ランダムマイジングアルゴリズムの入力として使用する前に、ADAM ディスクリプタ値から切り捨てるビット数を指定します。1~255 の範囲の値を指定できます。このパラメータを省略すると、0 ビットの値 (切り捨てなし) が使用されます。

このパラメータは、ほぼ等しい ADAM キーを使用して、ある種のレコードクラスタ化を行います。ADAMPARM は、ADAMFILE パラメータも指定した場合にのみ指定できます。

#### ALLOCATION：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

デフォルト (すなわち ALLOCATION=FORCE) の場合は、RABN 配置パラメータに従ったファイルエクステント割り当てができなると、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとしています。

配置パラメータ DSRABN、NIRABN、または UIRABN に従って十分なスペースを取得できないと、最初のエクステントのみがそこに作成され、残り（5番目のエクステントまで）は他の場所に作成されます。ただし、ALLOCATION=FORCE とともに配置パラメータ ACRABN を使用する場合は、指定された場所で完全なスペースを取得できる必要があります。取得できない場合、ユーティリティはエラーで終了します。

#### ANCHOR：拡張コンポーネント／アンカーファイル

ANCHOR は、既存または新規の拡張ファイルに対して基本（アンカー）ファイルを定義します。ANCHOR で定義されたファイルが FILE パラメータで定義されたファイルと同じである場合、ロードされたファイルは新しい拡張論理ファイルに対する物理基本（アンカー）ファイルになります。そうでない場合は、FILE パラメータで定義されたファイルが、ANCHOR で定義された拡張ファイルの新しいコンポーネントになります。

拡張ファイルの一部でないファイルを ANCHOR で指定すると、このファイルおよび FILE パラメータで指定されたファイルが LOAD 処理によって新しい拡張ファイルとして定義されます。また、ANCHOR で指定したファイルに対して NOACEXTENSION インジケータも設定されます。

既存の拡張ファイルのアンカーファイルを ANCHOR で指定すると、FILE で指定されたファイルが LOAD 処理によって拡張ファイルに追加されます。

 **Note:** 既存の拡張ファイルに新しいファイルをロードする場合は、アンカーファイルおよび追加されるファイルに対して排他更新の使用が必要です。これは、ユーティリティで使用するためにアンカーファイルをロックすることによって達成できます。

ANCHOR で指定されるファイルと FILE で指定されるファイルには、同じフィールド定義テーブル（FDT）構造が必要です。最大レコード長（MAXRECL パラメータ）およびファイルセキュリティ定義も同じである必要があります。

ANCHOR を指定すると、MINISN および NOACEXTENSION パラメータも指定する必要があります。カップリングされたファイルまたはマルチクライアントファイルを拡張ファイルの一部にすることはできません。

 **Note:** LOB ファイル（FILE パラメータを参照）をロードしている場合は、ADALOD LOAD の実行で ANCHOR パラメータを指定できません。

#### ASSOPFAC：アソシエータパディングファクタ

ASSOPFAC は、各アソシエータブロックに使用するパディングファクタを定義します。指定しないと、デフォルトのパディングファクタ 10 が使用されます。

指定値は、ロード処理中に使用しない各アソシエータブロック（パディングエリア）の比率を示します。パディングエリアは、新しいディスクリプタ値に対してブロックにエントリを追加する必要がある場合、または既存の値に対して新しい ISN を追加する必要がある場合に

使用するために確保します。これにより、オーバーフローしたエントリを別のブロックに再配置することから発生するオーバーヘッドを避けることができます。

1 から 90 の値が指定できます。アソシエータブロック内のバイト数からパディング用に確保されたバイト数を引いた値は、ファイル内の最大ディスクリプタ値に 10 バイトを加えた値を超えている必要があります。

ディスクリプタの更新頻度が少ないかまったくない場合には、小さいパディングファクタ (1~10) を指定します。新しいディスクリプタ値 (または新しい ISN) を追加するなど、ディスクリプタ更新を頻繁に行う場合には、大きいパディングファクタ (10~50) を指定します。

### ASSOVOLUME：アソシエータエクステントボリューム



**Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース (AC、NI、UI エクステント) を割り当てるボリュームを指定します。要求したブロック数が指定ボリュームに存在しない場合、ADALOD は、ASSOVOLUME パラメータに関係なく再度割り当てようとします。



**Note:** 指定されたボリュームに未使用の ASSO スペースが 5 ブロック以上存在すると、ADALOD はこのブロックを割り当てます。このスペースが十分でない場合は、ERROR-060 で終了します。指定されたボリュームに空きブロックが残っていない場合は、ADALOD は別のボリュームでスペースを割り当てようとします。

ACRABN、UIRABN、または NIRABN が指定されていると、ADALOD は ASSOVOLUME の値を無視して該当するエクステントタイプを割り当てます。ASSOVOLUME が指定されないと、ファイルのアソシエータスペースは、ADALOD のデフォルト割り当てルールに従って割り当てられます。

### BASEFILE：基本ファイル番号

BASEFILE は、ロードしている LOB ファイルに関連した基本ファイルのファイル番号を指定します。このパラメータは、LOB ファイルをロードするときのみ使用されます。

詳細については、「ラージオブジェクト (LB) フィールドの基本」を参照してください。

### DATAFRM：ADAM データストレージの上書き

DATAFRM は、ロード中の ADAM ファイルのデータストレージの上書きを制御します。DATAFRM=YES (デフォルト) を指定すると、ファイルのロード時にデータストレージエリアが ADALOD によって強制的に再フォーマットされます。DATAFRM=NO を指定すると再フォーマットが行われず、ロード処理速度が大幅に向上するため、レコード数が比較的少ないロード処理には、DATAFRM=NO を指定します。ただし、NO を指定する場合は、FROMRABN を指定した ADAFRM ユーティリティを使用して、データストレージエリアがフォーマット済みであることが前提になります。



**Caution:** DATAFRM=NO の指定には注意してください。プライマリデータストレージが正しくフォーマットされていない場合は、その後のファイル処理でエラーが発生したり、予期しない結果が発生することがあります。

**DATAPFAC**：データストレージパディングファクタ

DATAPFACは、データストレージの各物理ブロックに使用されるパディングファクタです。1～90の範囲のパーセント値を指定できます。指定しないと、デフォルトのパディングファクタ 10 が使用されます。

レコード拡張の頻度が少ないかまったくないことが予期される場合には、小さいパディングファクタ（1～10）を指定します。論理レコードを拡張する大量の更新が計画されている場合には、大きいパディングファクタ（10～50）を指定します。

指定されたパーセント値は、以降のレコード拡張でロード処理中に確保される各データストレージブロック（パディングエリア）の部分を表します。レコード更新の結果、ブロック内の論理レコードで追加スペースが必要になった場合に、パディングエリアが使用されます。これにより、レコードを別のブロックに再配置することから発生するオーバーヘッドを避けることができます。

ファイルにロードされたレコードの長さは異なる場合があるため、パディングファクタは、各ブロックで指定されたパーセントと必ずしも一致しません。Adabasは、少なくとも50バイトがブロックに残るように、異なるレコード長のパディングエリアサイズのバランスを保ちます。

**例**

ブロックサイズは1000バイト、パディングファクタは10%です。したがって、レコードのロードに使用可能なスペース（ブロックサイズからパディングエリアを差し引く）は900バイトになります。

数件のレコードをロードした後、800バイトのブロックが使用されました。次のレコードの長さは170バイトです。レコードがロードされると、ブロックの残りが50バイト未満になるため、このレコードは現在のブロックにロードできません。したがって、レコードは次のブロックにロードされます。

現在のブロックは800バイトが使用されたままの状態になります。800バイトと900バイトの差（-100バイト）はバランスのために使用されます。

次のレコードが170バイトではなく150バイトであり、その時点での累積バランスの値が負のバイト数であると想定します。レコードをロードした後、ブロックに50バイト（1000-950）が残るため、150バイトのレコードはロードされます。

ただし、パディングエリアの50バイトは、バランス用に+50バイトを残して使用されます（900-950）。

ADAM オプションを指定してロードしたファイルの場合、ブロック（パディングエリアを含む）内でスペースが使用可能であれば、計算されたデータストレージブロックに新しいレコードがロードされます。計算されたブロックに格納できないレコードは、別のブロックに格納されます（この場合、パディングエリアは使用されません）。

## DATAVOLUME：データストレージエクステンションボリューム

 **Note:** DATAVOLUME 値は単一引用符でくくらなければなりません。

DATAVOLUME は、ファイルのデータストレージスペース (DS エクステンション) が割り当てられるボリュームを指定します。DSSIZE で要求されるブロック数が指定ボリュームに存在しないと、ADALOD は、DATAVOLUME 値を無視して再度割り当てようとします。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。DATAVOLUME を指定しないと、ADALOD のデフォルト割り当てルールに従って、データストレージスペースが割り当てられます。

## DSDEV：データストレージデバイスタイプ

DSDEV は、ファイルのデータストレージをロードするデバイスタイプを指定します。デフォルト値はありません。DSDEV を指定しない場合、任意のデバイスタイプが使用されます。

## DSREUSE：データストレージの再利用

DSREUSE は、使用可能となったデータストレージスペースを再利用するかどうかを指定します。デフォルトは YES です。

## ETID：マルチクライアントファイルオーナー ID

ETID パラメータは、マルチクライアントファイルにロードされている全レコードに対して新しいオーナー ID を割り当てます。ロードされているレコードのオーナーを識別するユーザー ID を指定します。レコードに割り当てられるオーナー ID は指定したユーザー ID のユーザープロファイルから取得します。

ロードするファイルがマルチクライアントファイル (**LOWNERID** パラメータの説明を参照) で、入力ファイルにオーナー ID が含まれていない場合、つまり入力ファイルがマルチクライアントソースファイルからロードされていない場合には、ETID パラメータを指定する必要があります。

入力ファイルがマルチクライアントソースファイルからロードされた場合には、ETID はオプションです。この場合、ロードされたレコードで元のオーナー ID が保持されます。

非マルチクライアントファイルをロードする場合は、ETID パラメータを指定できません。

 **Note:** ETID パラメータを使用する場合、ADALOD ユーティリティはアクティブニュークリアスを必要とします。ニュークリアスは、ETID 値を内部オーナー ID 値に変換します。

## IGNFDT：古い FDT の無視

KEEPFDT パラメータを指定した ADADBS の DELETE 機能を使用してファイルを削除すると、フィールド定義テーブル (FDT) はアソシエータ内に残ります。そのファイルを再びロードするときに IGNFDT が指定されていないと、ファイルの古い FDT が ADALOD によって新しいものと比較されます (セキュリティ情報は比較されません)。両方の FDT が同一であれば、ADALOD はファイルをロードし、古い FDT は新しい FDT に置換されます。FDT が同一でなければ、古い FDT が保持され、ADALOD はエラーメッセージを出力して処理を終了します。

IGNFDT パラメータを指定すると、ファイルの既存の（古い）FDT は ADALOD で無視され、比較は行われません。古い FDT は新しい FDT に置換され、ファイルがロードされます。

#### **INDEXCOMPRESSION**：ファイルインデックスの圧縮

INDEXCOMPRESSION には、ファイルインデックスを圧縮形式または非圧縮形式のどちらでロードするかを指定します。通常、インデックスを圧縮すると、インデックススペースが少なくなり、Adabas ニュークリアスのインデックス操作の効率が向上します。

INDEXCOMPRESSION を指定しないと、ADALOD はシーケンシャル入力ファイルからデフォルト値を取得します。入力ファイルの作成方法によって、次のようになります。

- ADACMP を使用した場合は、デフォルト値は NO になります。
- ADAULD を使用した場合は、アンロード時のファイルの値がデフォルト値になります。

#### **ISNREUSE**：ISN の再利用

ISNREUSE は、レコード削除の結果解放された ISN を新しいレコードに再割り当てできるかどうかを指定します。デフォルトは NO です。

#### **ISNSIZE**：3 または 4 バイト ISN

ISNSIZE は、ファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを示します。デフォルトは 3 バイトです。

#### **LIP**：ISN バッファプールサイズ

ISN と割り当てられたデータストレージ RABN を入れるための ISN プールサイズを指定します。値はバイト単位で指定し、数値のみ ("2048" など) を指定するか、あるいは数値の後に "K" を付けて ("2K" など) キロバイト単位で指定します。LIP のデフォルトは、2000 バイトです。

LIP は、USERISN=YES が指定され、ユーザー指定の ISN がソートされていない場合のロード中にアドレスコンバータの入出力回数を削減するために使用できます。処理対象となるすべての ISN を保持できるだけのバッファサイズを LIP で指定すると、最適なパフォーマンスを得ることができます。

1 件の入力レコード長は、 $ISNSIZE + RABNSIZE + 1$  です。したがって、エントリ長は、最小 7 バイト（ファイルの ISNSIZE が 3 でデータベースの RABNSIZE が 3）、最大 9 バイト（ISNSIZE が 4 で RABNSIZE が 4）になります。

#### **LOBFILE**：LOB ファイル番号

LOBFILE は、ロードしている基本ファイルに関連した LOB ファイルのファイル番号を指定します。このパラメータは、基本ファイルのロード時にのみ使用されます。

詳細については、「ラージオブジェクト (LB) フィールドの基本」を参照してください。

#### **LOWNERID**：マルチクライアントファイルの内部オーナー ID 長

LOWNERID パラメータは、マルチクライアントファイルの各レコードに割り当てられる内部オーナー ID 値の長さを指定します。有効な長さは 0~8 です。LOWNERID パラメータを指定しないと、入力ファイルにあるオーナー ID の長さがデフォルト値になります。

LOWNERID パラメータの指定値またはデフォルト値によって、ファイルをマルチクライアントファイルとしてロードするのか、非マルチクライアントファイルとしてロードするのかが決まります。有効なLOWNERIDの値が0の場合はファイルが標準の非マルチクライアントファイルとしてロードされ、0でない場合はマルチクライアントファイルとしてロードされます。

LOWNERID パラメータを ETID パラメータと組み合わせると、次の操作を行うことができます。

- 非マルチクライアントファイルをマルチクライアントファイルとして再ロードする
- ファイルのオーナー ID 長を増加／減少する
- ファイルのレコードからオーナー ID を削除する

次の表に、LOWNERID パラメータと入力ファイルのオーナー ID 長の有効な組み合わせを示します。

LOWNERID パラメータ設定	入力ファイルのオーナー ID 長の値	
	0	2
0	非マルチクライアントファイルとして保持	非マルチクライアントファイルに変換
1	マルチクライアントファイルの設定 (ETID)	オーナー ID 長の縮小
2	マルチクライアントファイルの設定 (ETID)	オーナー ID 長の保持
3	マルチクライアントファイルの設定 (ETID)	オーナー ID 長の拡張
(指定なし)	非マルチクライアントファイルとして保持	マルチクライアントファイルとして保持

マルチクライアントファイル (LOWNERID の指定値またはデフォルト値が 0 以外) をロードするとき、ETID パラメータを指定して、ロードするすべてのレコードに新しいオーナー ID を割り当てることができます。入力ファイルにオーナー ID がすでに含まれており、ETID を省略すると、すべてのレコードで元のオーナー ID が保持されます。

上の表で "入力ファイルのオーナー ID 長の値が 0" となっている列に ETID パラメータが示されている箇所は、入力ファイルにオーナー ID が存在しないため、ETID パラメータが必須であることを示しています。



**Note:** LOB ファイル (**FILE パラメータ**を参照) をロードしている場合は、ADALOD LOAD の実行で LOWNERID パラメータを指定できません。

### LWP：ワークプールサイズ

LWP は、ディスクリプタ値のソートに使用されるワークプールの大きさを指定します。この値は、数値のみでバイト単位で指定するか、あるいは数値の後に "K" を付けてキロバイト単位で指定します。値が指定されない場合は、デフォルトで 1048576 バイト (または 1024K) が設定されます。ただし、ディスクリプタが非常に長い、またはディスクリプタの数が著しく多いファイルに対して ADALOD の実行時間を短くする場合には、より大きな値を LWP

に設定すべきです。ソートデータセットとの問題を避けるため、比較的小規模なファイルをロードする場合には、LWP 値を小さめに指定してください。

最小ワークプールの大きさは、ソートデータセットのデバイスタイプごとに異なります。

ソートデバイス	最小 LWP	最小 LWP
	バイト	キロバイト
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

#### MAXDS/MAXNI/MAXUI：最大 2 次割り当て

データストレージ (MAXDS)、ノーマルインデックス (MAXNI)、またはアップーインデックス (MAXUI) の 2 次エクステンントごとに割り当てる最大ブロック数を指定します。値はブロック単位で指定し (MAXNI=8000B など)、65535B を超えることはできません。この値を指定しない場合は、無制限であると見なされます (デフォルト)。

#### MAXISN2：最大セカンダリ ISN に相当するセカンダリアドレスコンバータ RABN の割り当て

ADALOD 入力ファイル内にスパンドレコードが存在するかどうかに関係なく、MAXISN2 の指定は任意です。このパラメータを使用して、ISN におけるセカンダリアドレスコンバータ (AC2) のサイズが決定されます。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

ADALOD は、次の方式に従って、セカンダリ AC 内のスペースを使用できるように ISN マッピングの数を判断します。

1. セカンダリ AC がまだ割り当てられていない場合は、セカンダリ AC ブロック 1 つが割り当てられます。
2. 現在の MAXISN2 設定が MAXISN 値 (予期される現在の最大プライマリ ISN 数) の 4 倍以上である場合は、追加のプライマリ ISN を割り当てるときと同じアルゴリズムを使用してセカンダリ ISN の数を決定します。
3. 上記のいずれの条件も満たさない場合、セカンダリ AC のスペースは、次の 2 つの計算のうち小さい値として割り当てられます。
  - 旧 MAXISN2 設定の 10 倍 ( $10 \times \text{oldMAXISN2}$ )
  - 旧 MAXISN2 設定と MAXISN 設定の合計 ( $\text{oldMAXISN2} + \text{MAXISN}$ )

デフォルト値はありません。

#### MAXRECL：最大圧縮レコード長

MAXRECL は、ファイルに有効な最大圧縮レコード長を指定します。デフォルト値は、使用デバイスタイプでサポートされている最大長です。

**MINISN：最小 ISN 件数**

このパラメータは、ファイルに割り当てることができる最小 ISN 件数を指定します。デフォルトは 1 です。

MINISN の主な目的は、Adabas 拡張ファイルのコンポーネントファイルに対して ISN 範囲の最小値を割り当てることです。拡張ファイルに対して ANCHOR を指定する場合は、MINISN が必要です。

MINISN は、1 を大幅に超えた値がすべてのレコードに割り当てられているファイルで、アソシエータスペースの浪費を避けるために使用します。例えば、銀行の口座番号に ISN 番号を使用しており、最小の口座番号が 1,000,001 であるとしします。この場合、MINISN = 1000001 と指定すると、1~999999 の ISN に対してアドレスコンバータのスペース割り当てが停止され、未使用になります。詳細は、MAXISN パラメータの説明を参照してください。

**MIXDSDEV：混合デバイスタイプデータストレージ**

MIXDSDEV では、異なるデバイスで 2 次データストレージエクステントを割り当てることができます。異なるブロック長が有効になります。MIXDSDEV を指定しない場合（デフォルト）は、指定ファイルに対するデータストレージエクステントがすべて同じデバイスタイプに存在している必要があります。

**NAME：ファイル名**

NAME は、ファイルに割り当てる名前です。この名前は、ADAREP ユーティリティで作成されるデータベースステータスレポートに、このファイルに関するデータとともに出力されます。許可される最大文字数は 16 です。デフォルト名は TESTFILE です。

ファイル名に特殊文字または埋め込み空白が含まれている場合は、名前を単一引用符で囲う（'...'）必要があります。単一引用符自体が名前に含まれる場合は、'JAN'S FILE' のように 2 回連続して指定する必要があります。

**NISIZE：ノーマルインデックスサイズ**

NISIZE は、ノーマルインデックスに対して割り当てられるブロックまたはシリンダ数を指定します。ブロック値の場合には値の後に "B" を指定する必要があります（"5500B" など）。

指定されたエクステントサイズが最大空きスペースのサイズを超える場合には、要求を満たすのに必要なファイルエクステント（最大 5 個）が ADALOD によって割り当てられます。

NISIZE パラメータを省略すると、次のようになります。

- ADALOD は、入力データセットからのレコードのサンプリングに基づいてノーマルインデックスに対するスペースを割り当てます。この見積り処理は、CPU タイムの増加と入出力処理を要求します。したがって、サイズが分かっている場合には見積り処理が実行されないようにパラメータの設定を行うべきです。
- このパラメータが省略され、なおかつ INDEXCOMPRESSION=YES が設定された場合、ADALOD は予想される圧縮率が想定できないため、インデックス圧縮を考慮してインデックスサイズを計算することができません。したがって、ADALOD は、必要以上に大きなインデックスを割り当てる可能性があります。

ロードするレコード数が少なく、後で大量のレコードを追加する場合は、レコードのデータ量の合計に合わせて、ノーマルインデックスを増やすことができるように NISIZE パラメータを設定する必要があります。詳細は、「**LOAD データとスペースの要件**」の「LOAD ファイルのスペース割り当て」を参照してください。

**NOACEXTENSION**：アドレスコンバータエクステントの制限

NOACEXTENSION を指定すると、このファイルに定義された MAXISN を今後増やすことができなくなります。追加のアドレスコンバータ (AC) エクステントは作成されません。NOACEXTENSION は、Adabas 拡張ファイルを構成しているコンポーネントファイルに主に適用されます。ANCHOR を指定すると、NOACEXTENSION も指定する必要があります。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**NUMREC**：ロードするレコード数の制限

NUMREC は、ロードするレコード数の限度を指定します。NUMREC を指定すると、(入力データセットのエンドオブファイル条件によって ADALOD がすでに終了していない限り) 指定数のレコードを処理した後で ADALOD は終了します。このオプションは、テスト用にファイルのサブセットを作成する場合によく使用されます。このパラメータを省略すると、入力レコードがすべて処理されます。

入力データセット内のレコード数が NUMREC の指定値より多い場合、ADALOD は NUMREC で指定されたレコード数を処理した後、コンディションコード 4 で終了します。

 **Note:** LOB ファイル (**FILE** パラメータを参照) をロードしている場合は、ADALOD LOAD の実行で NUMREC パラメータを指定できません。

**PGMREFRESH**：プログラムで生成されたファイルのリフレッシュ

PGMREFRESH は、ロード中のファイルに対してユーザープログラムでリフレッシュ処理を実行できるかどうかを指定します。PGMREFRESH を指定すると、E1 コマンドまたはニュークリアスに対する同等のコールを使用して、リフレッシュを行うことができます。

**RESTART**：中断 ADALOD 処理の再開

RESTART は、中断された ADALOD 実行を強制的に再開させるもので、中断までに達していた最終再開位置から再スタートが行われます。再開位置とは、中間データセットから復元可能な実行処理の最終位置を意味します。

エラー条件定義によって ADALOD が中断された場合、再開が可能であるかどうかを示すメッセージが ADALOD によって発行されます。

ADALOD オペレーションを再開するときに、次の各パラメータを変更できます。

- TEMPSIZEを増やして中間データセットをより大きくすることができます。ただし、中間データセットには再開操作に必要な情報が含まれているため、この内容は変更できません。
- SORTSIZE パラメータ、SORTDEV パラメータ、およびソートデータセットを変更できます。

その他のパラメータを変更することはできません。DDEBAND/EBAND および DDFILEA/FILEA データセットは変更できません。

### SKIPREC：スキップするレコード数

SKIPRECは、ロード処理を開始する前にスキップする入力レコード数を指定します。デフォルトは0です（レコードがスキップされていない場合）。



**Note:** LOB ファイル (**FILE** パラメータを参照) をロードしている場合は、ADALOD LOAD の実行で SKIPREC パラメータを指定できません。

### SORTDEV：ソートデバイスタイプ

ADALOD はディスクリプタ値のソートにソートデータセットを使用します。SORTDEV パラメータは、このデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

### TEMPDEV：一時ストレージデバイスタイプ

ADALOD は、中間データセットを使用して中間データを格納します。TEMPDEV パラメータは、このデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### UISIZE：アッパーインデックスサイズ

UISIZE は、アッパーインデックスに対して割り当てられるブロック数またはシリンダ数を指定します。ブロック値の場合には値の後に "B" を指定する必要があります ("5500B" など)。

指定されたエクステンツサイズが最大空きスペースのサイズを超える場合には、要求を満たすのに必要なファイルエクステンツ（最大5個）が ADALOD によって割り当てられます。

UISIZE パラメータを省略すると、次のようになります。

- ADALOD は、入力データセットからのレコードのサンプリングに基づいてアッパーインデックスに対するスペースを割り当てます。この見積り処理は、CPU タイムの増加と入出力処理を要求します。したがって、サイズが分かっている場合には見積り処理が実行されないようにパラメータの設定を行うべきです。
- このパラメータが省略され、なおかつ INDEXCOMPRESSION=YES が設定された場合、ADALOD は予想される圧縮率が想定できないため、インデックス圧縮を考慮してインデッ

クスサイズを計算することができません。したがって、ADALOD は、必要以上に大きなインデックスを割り当てる可能性があります。

ロードするレコード数が少なく後から大量のレコードを追加する場合は、総合計のレコード数に合わせてアッパーインデックスを増やすことができるように UISIZE パラメータを設定する必要があります。詳細は、「[LOAD データとスペースの要件](#)」の「LOAD ファイルのスペース割り当て」を参照してください。

#### UQDE：ユニークディスクリプタ

UQDE は、1つ以上のディスクリプタをユニークとして定義します。指定された各ディスクリプタは、それぞれの入力レコードで異なる値を持っている必要があります。ADALOD 処理中にユニークでない値が検出されると、ADALOD はエラーメッセージを出力して終了します。

ADACMP ユーティリティでユニークディスクリプタ (UQ) オプションを指定した場合は、UQDE パラメータをここで指定する必要はありません。

更新コマンドによってディスクリプタの値が重複する可能性がある場合、ユニークディスクリプタ (UQ) オプションを使用して定義されたディスクリプタについては、追加または更新コマンドによる更新が Adabas によって防止されます。

 **Note:** Adabas 拡張ファイルの場合、ADALOD は、指定されたコンポーネントファイル内でしかユニークディスクリプタ違反を検出できません。同じ値のユニークディスクリプタが他のいずれかのコンポーネントファイルに存在していても、ADALOD では検出できません。したがって、ユニークディスクリプタの値が、全拡張ファイルを通じてユニークであるようにする必要があります。

 **Note:** LOB ファイル ([FILE パラメータ](#)を参照) をロードしている場合は、ADALOD LOAD の実行で UQDE パラメータを指定できません。

#### USERISN：ユーザー ISN 割り当て

USERISN=YES は、ロードされたファイルに対して USERISN オプションが有効になり、新規の各レコードに対してユーザーが入力データで ISN を指定することを示します。

USERISN=NO の場合、Adabas は、それぞれの新規レコードに ISN を割り当てます。

USERISN を指定しないと、入力ファイルに応じたデフォルト設定を使用すると見なされます。USERISN オプションを指定した ADACMP で入力ファイルを作成した場合、または USERISN オプションを指定したファイルから ADAULD で入力ファイルを作成した場合には、ADALOD のデフォルトは USERISN=YES になります。それ以外の場合は、USERISN=NO がデフォルトになります。ここで USERISN を指定すると、既存のデフォルト値が上書きされます。

 **Note:** USERISN オプションを指定して Adabas 5.2 ファイルを最初にロードした場合は、ファイルを再ロードするときに USERISN=YES を再度指定する必要はありません。ADALOD は上記のようにデフォルトを想定します。ただし、USERISN オプションを指定して Adabas 5.1 ファイルを最初にロードした場合は、再ロードするたびに USERISN=YES を指定する必要があります。

### VERSION：入力データ形式

このパラメータは本来、Adabasにロードする出力（ADACMP）データセットのバージョンを指定していました。

シーケンシャル入力データセットのバージョンは ADALOD で決定されるため、このパラメータは無視されます。旧バージョンの ADALOD ジョブとの互換性のためにのみ使用できます。

## 例

---

### 例 1

```
ADALOD LOAD FILE=6,MAXISN=20000,DSSIZE=20,ASSOPFAC=15,  
ADALOD DATAPFAC=15,TEMPSIZE=20,SORTSIZE=10
```

ファイル 6 をロードします。ファイルに対して最初に有効なレコード数は 20,000 です。データストレージに 20 シリンダを割り当てます。アソシエータおよびデータストレージのブロックパディングファクタは、どちらも 15 % です。中間データセットおよびソートデータセットは、それぞれ 20 および 10 シリンダです。

### 例 2

```
ADALOD LOAD FILE=7,MAXISN=350000,ASSOPFAC=5,MINISN=100001  
ADALOD DATAPFAC=15,DSSIZE=100,USERISN=YES  
ADALOD TEMPSIZE=200,SORTSIZE=100
```

ファイル 7 をロードします。ファイルに対して最初に割り当てられたレコード数は 250,000、最小レコード数は 100,001 です。アソシエータパディングファクタは 5 % です。データストレージパディングファクタは 15 % です。データストレージに 100 シリンダを割り当てます。ISN は入力に含まれます。中間データセットおよびソートデータセットは、それぞれ 200 および 100 シリンダです。

### 例 3

```
ADALOD LOAD FILE=8,ADAMFILE,ADAMDE='AK'  
ADALOD ADAMPARM=4,ADAMOFLOW=5,UQDE='AK',MINISN=1  
ADALOD MAXISN=10000,DSSIZE=20,ASSOPFAC=5,DATAPFAC=5  
ADALOD TEMPSIZE=10,SORTSIZE=5
```

ファイル 8 を ADAM ファイルとしてロードします。フィールド AK は ADAM キーです。AK 値を使用してレコードに対するデータストレージ RABN を計算する際に、AK の各値から 4 ビット切り捨てます。ADAM オーバーフローエリアのサイズは 5 シリンダです。フィールド AK はユニークディスクリプタとして定義されています。ファイルに対して最初に割り当てられた最大レコード数は 10,000 です。データストレージに 20 シリンダを割り当てます。このうち、ADAM オーバーフローが 5 シリンダを占めています。アソシエータおよびデータストレージのパディングファクタは、どちらも 5 % です。中間データセットおよびソートデータセットのサイズは、それぞれ 10 および 5 シリンダです。

## 例 4

```
ADALOD LOAD FILE=9,NAME=INVENTORY,MAXISN=5000
ADALOD      DSSIZE=2000B,DSRABN=30629,NISIZE=300B,UISIZE=50B
ADALOD      MAXDS=1000B,MAXNI=50B,MAXUI=1B
ADALOD      INDEXCOMPRESSION=YES
ADALOD      ASSOPFAC=20,DATAPFAC=10
ADALOD      TEMPSIZE=10,SORTSIZE=5,UQDE='U1,U2'
```

ファイル 9 をロードします。ファイルのテキスト名は INVENTORY です。ファイルの初期スペース割り当ては 5,000 レコードです。データストレージに、RABN 30,629 から開始して 2,000 ブロックを割り当てます。ノーマルインデックスに 300 ブロックを割り当てます。アップパーインデックスに 50 ブロックを割り当てます。データストレージ、ノーマルインデックス、およびアップパーインデックスの 2 次エクステンントごとの最大割り当ては、それぞれ 1000 ブロック、50 ブロック、および 1 ブロックです。インデックスを圧縮します。アソシエータパディングファクタは 20 % です。データストレージパディングファクタは 10 % です。中間データセットおよびソートデータセットは、それぞれ 10 および 5 シリンダです。ディスクリプタ U1 および U2 はユニークディスクリプタとして定義されています。

## 例 5

```
ADALOD LOAD FILE=2,SECURITY
ADALOD      DSSIZE=20B,MAXISN=2000,NISIZE=20B,UISIZE=5B
ADALOD      TEMPSIZE=10,SORTSIZE=5
```

ファイル 2 を Adabas セキュリティファイルとしてロードします。DDEBAND の内容は無視されます。スペース割り当ては、データストレージに 20 ブロック、アドレスコンバータに 2000 ISN、ノーマルインデックスに 20 ブロック、アップパーインデックスに 5 ブロックです。中間サイズは 10 シリンダ、ソートエリアサイズは 5 シリンダです。

## LOAD データとスペースの要件

次の一般情報は、LOAD 操作に関するデータ要件および ADALOD LOAD によるスペース割り当ての方法について説明します。スペース割り当ての詳細については、『*Adabas DBA リファレンスマニュアル*』を参照してください。

### LOAD 操作の入力データ

ADACMP または ADAULD ユーティリティで生成された圧縮データレコードは、ADALOD の入力として使用できます。MODE=SHORT オプションを指定して実行された ADAULD ユーティリティの出力を ADALOD の入力として使用すると、ディスクリプタ情報はすべて FDT から削除され、ファイルに対するインデックスは存在しません。

### LOAD のファイルスペース割り当て

ADALOD は、ロードするファイルに対してノーマルインデックス (NI)、アップパーインデックス (UI)、アドレスコンバータ (AC)、およびデータストレージのスペースを割り当てます。

### インデックススペースの割り当て

NISIZEやUISIZEパラメータを指定すると、ユーザーが指定した値に基づいて割り当てが行われます。このパラメータを指定しないと、ADALODは各ディスクリプタに存在する値のサンプリングに基づいてこれらのインデックスに対するスペースを割り当てます。

ディスクリプタ値は、次のようにサンプリングされます。

1. ADALODは圧縮された入力レコードを読み込み、データストレージにレコードを格納し、各ディスクリプタのそれぞれの値を抽出してそれを中間データセットに書き込みます。各中間ブロックには1つのディスクリプタの値のみが格納されます。この処理が完了すると、次の情報が取得されます。

- 各ディスクリプタの値の個数
- 各ディスクリプタに必要なバイト数
- 各ディスクリプタに使用される中間RABN

ユニークディスクリプタの場合、NIのスペース要件は使用される中間サイズと同じになります。非ユニークディスクリプタの場合は、重複する値の個数を求める必要があります。重複するそれぞれの値のスペース要件を見積り、必要なバイト数からその値を差し引く必要があります。この結果が、重複するディスクリプタに対するNIサイズになります。

重複する値の個数は、1つのディスクリプタの値を含む16個までの中間ブロックを読み込んで求められます。この値がソートされ、重複する値の個数が求められます。この個数に、次の計算式で得られる係数が掛けられます。

$$\frac{\text{値の総数}}{\text{サンプル内の値の数}}$$

その結果が、このディスクリプタに対してファイル全体に存在する同一ディスクリプタ値の予測個数になります。このスペース要件が中間サイズの見積りから差し引かれます。

2. アップパーインデックス (UI) サイズは、ノーマルインデックス (NI) および中間サイズがすべて使用可能になった後で計算されます。

3. NI および UI のサイズには、次の計算式の結果がそれぞれ掛けられます。

$$\frac{(\text{MAXISN} - \text{MINISN}) + 1}{\text{ロード対象のレコード数}}$$

例えば、MINISN=1（デフォルト）を指定したときに 10000 個のレコードに対して 10 ブロックの UI スペースと 500 ブロックの NI スペースを必要とする場合、MAXISN=60000 と指定すると、60 ブロックの UI と 3000 ブロックの NI が割り当てられることとなります。

$$10 \times \frac{60000}{10000} = 60 \text{ UI ブロック}$$

$$500 \times \frac{60000}{10000} = 3000 \text{ NI ブロック}$$

ただし、USERISN=YES の指定が有効な場合には、この計算は行われません。

MAXISN を適切に設定することで、ロードするレコード数が少なく後から大量のレコードを追加するようなファイルに対して、割り当てサイズを増やすことが可能になります。

NISIZE および UISIZE パラメータを指定すると、未割り当てのアソシエータ RABN を使用してスペースが割り当てられます。NIRABN および／または UIRABN パラメータを指定すると、ユーザーが指定した RABN を使用してスペースが割り当てられます。

アドレスコンバータのスペース割り当て

アドレスコンバータの割り当ては、ファイルに対する MAXISN および MINISN の値を基にしています。ADALOD は、次の式から計算されるバイト数を格納するのに必要なブロック数を割り当てます。

$$\text{RABNSIZE} \times ((\text{MAXISN} - \text{MINISN}) + 1)$$

ACRABN パラメータを指定すると、ユーザーが指定したブロック番号から開始してアドレスコンバータが割り当てられます。このパラメータを指定しないと、未割り当てのアソシエータ RABN が使用されます。

データストレージのスペース割り当て

データストレージの割り当ては、DSSIZE パラメータで指定された値を基にしています。DSRABN パラメータを指定すると、ユーザーが指定したブロック番号から開始して割り当

てが行われます。このパラメータを指定しないと、未割り当てのデータストレージRABNが使用されます。

データベースに異なるデバイスタイプが存在する場合、DSDEVを指定することによって、データストレージの割り当てを指定のデバイスタイプで強制的に行うことができます。MIXDSDEVパラメータを指定すると、MAXRECLで指定された長さのレコードをデバイスタイプで格納できるという前提のもとに、異なるデバイスタイプでデータストレージを割り当てることができます。

中間エリアのスペース割り当て

ADALODはディスクリプタごとに、値およびその値を含むレコードのISNのリストを生成し、この情報を中間データセットに書き込みます。ディスクリプタ情報に必要なスペースは、各ディスクリプタに必要なスペースの合計になります。各ディスクリプタに必要なスペースは、次の式で計算されます。

$$SP = N * NPE * NMU * (L + 4)$$

ここでは次の内容を表しています。

SP	ディスクリプタに必要なスペース (バイト単位)
N	ロードするレコードの数
NPE	ディスクリプタがピリオディックグループ内に含まれている場合の平均オカレンス数。ピリオディックグループに含まれない場合のNPEは1。
NMU	ディスクリプタがマルチプルバリューフィールドの場合の平均オカレンス数。マルチプルバリューフィールドでない場合のNMUは1。
L	ディスクリプタの各値の (圧縮後の) 平均長

例

20,000レコードを含むファイルをロードします。ファイルには2つのディスクリプタ (AAおよびCC) があります。ディスクリプタAAにはレコードごとに1個の値が存在し、圧縮後の値の平均長は3バイトです。ディスクリプタCCにはレコードごとに平均10個の値が存在し、圧縮後の値の平均長は4バイトです。

フィールド定義：

```
01,AA,5,U,DE
01,CC,12,A,DE,MU
```

■ AAのスペース要件

$$SP = 20,000 \quad 1 \quad (3 + 4)$$

$$SP = 140,000 \text{ bytes}$$

■ CCのスペース要件

```
SP = 20,000      10      (4 + 4)
SP = 1,600,000 bytes
```

■ 合計スペース要件 = 1,740,000 バイト

必要なシリンダ数を計算するには、必要なブロック数をシリンダ当たりのブロック数で除算します。

モデル 3380 デバイスタイプの場合：

$$\text{必要なブロック数} = \frac{1,740,000}{7476 \text{バイト/ブロック}} = 232 \sim 233 \text{ブロック}$$

$$\text{必要なシリンダ数} = \frac{233 \text{ブロック}}{90 \text{ブロック/シリンダ}} = 2 \sim 3 \text{シリンダ}$$

### LOAD によるアソシエータの更新

ADALOD は、入力時に収集されたディスクリプタ値をソートした後、その値をノーマルインデックスおよびアップパーインデックス内に格納します。割り当てられたインデックススペースがノーマルインデックスまたはアップパーインデックスに対して不十分であると、ADALOD によって 4 個までの追加エクステントが割り当てられます。

割り当てられた各追加エクステントは、インデックスに対して現在割り当てられているスペース合計の約 25 % です。追加エクステントのスペースが不足しているか、割り当て済みのエクステントが最大数に達しているか、ADALOD はエラーメッセージを出力して終了します。

## 拡張ファイルのロード

拡張ファイルは、一連の標準 Adabas 物理ファイルで構成されます。拡張ファイル内でのファイル順序は任意です。最初のファイルが 53、2 番目ファイルが 127、3 番目のファイルが 13 などでもかまいません。各コンポーネントファイルに割り当てられる ISN はユニークである必要があり、2 つのファイルに同じ ISN が存在することはできません。全ファイルの ISN 範囲は昇順でなければなりません、順序間にギャップがあってもかまいません。

拡張ファイルチェーン内のレコード総数は、4,294,967,294 を超えることはできません。

拡張論理ファイルを構成する物理コンポーネントファイルの順序は、順序内での最初のコンポーネントファイル（アンカー）を定義する ANCHOR パラメータで定義されます。アンカーファイルは他の Adabas ファイルと同様にロードされます。追加の各コンポーネントファイルは、アンカーファイルを参照する ANCHOR パラメータを使用してロードする必要があります。ADALOD は、追加されたファイルに割り当てられた ISN 範囲に従って、新しい物理ファイルを既存の拡張ファイルチェーンに挿入します。追加された各コンポーネントファイルのロード時

には、NOACEXTENSIONパラメータを指定して、Adabasで新しいISNがコンポーネントファイルに割り当てられるのを防ぐ必要があります。

ADALODは、アンカーファイルおよび拡張ファイルを構成する1つの物理（コンポーネント）ファイルのみを処理します。拡張ファイル全体を処理するものではありません。

### 拡張ファイルへのデータのロード

データ（数百万のレコードなど）を異なる物理ファイルにロードするには、まず入力データをいくつかのDDEBAND/EBAND入力ファイルに分割する必要があります。DDEBAND/EBANDファイルデータは、SKIPRECおよびNUMRECパラメータを使用してコンポーネントファイルにマッピングできますが、スキップまたは限度なしの1対1のマッピングを推奨します。これにより、後で使用されないレコードは読み込む必要がなくなるため、パフォーマンスが向上します。

#### 例

拡張ファイルをロードするジョブの一部を抜粋した次の例で、DDEBAND/EBANDファイルからコンポーネントファイルへのマッピングを示します。

```
//DDEBAND      DD DSN=LOAD.DATA.FILE1,...
//DDKARTE      DD *
ADALOD LOAD   FILE=40,NAME='XXX_Part1'
ADALOD        MINISN=1,MAXISN=10000000,NOACEXTENSION
ADALOD        NUMREC=10000000
ADALOD        DSSIZE=...,NISIZE=...,UISIZE...
ADALOD        SORTSIZE=...,TEMPSIZE=...
.
.

//DDEBAND      DD DSN=LOAD.DATA.FILE1,...
//DDKARTE      DD *
ADALOD LOAD   FILE=41,NAME='XXX_Part2',ANCHOR=40
ADALOD        MINISN=10000001,MAXISN=20000000,NOACEXTENSION
ADALOD        NUMREC=10000000,SKIPREC=10000000
ADALOD        DSSIZE=...,NISIZE=...,UISIZE...
ADALOD        SORTSIZE=...,TEMPSIZE=...
.
.

//DDEBAND      DD DSN=LOAD.DATA.FILE2,...
//DDKARTE      DD *
ADALOD LOAD   FILE=35,NAME='XXX_Part2',ANCHOR=40
ADALOD        MINISN=20000001,MAXISN=30000000,NOACEXTENSION
ADALOD        NUMREC=10000000
ADALOD        DSSIZE=...,NISIZE=...,UISIZE...
ADALOD        SORTSIZE=...,TEMPSIZE=...
```

## マルチクライアントファイルのロード



**Note:** マルチクライアントファイルを拡張ファイルの一部にすることはできません。また、拡張ファイルをマルチクライアントファイルに変換することもできません。

マルチクライアントファイルには、複数のユーザーまたはユーザーグループのレコードが格納されます。マルチクライアント機能では、各レコードにオーナー ID をつけることで、物理ファイルを複数の論理ファイルに分割します。各ユーザーは、ユーザーのオーナー ID に関連付けられたレコードのサブセットだけにアクセスできます。

RACF や CA-Top Secret などの外部セキュリティパッケージがインストールされている場合でも、ユーザーは変わらず Natural ETID またはログオン ID で識別されます。オーナー ID は、ユーザー ID に対して割り当てられます。1つのユーザー ID は、1つのオーナー ID しか持つことができませんが、同じオーナー ID が複数のユーザーに所属していてもかまいません。

ADALODLOAD 機能では、標準環境からマルチクライアント環境へのアプリケーションの移行をサポートするため、LOWNERID および ETID パラメータを使用します。これらのパラメータが組み合わせられて、オーナー ID の定義と、およびファイルがマルチクライアントファイルかどうかの判断が行われます。

LOWNERID は、マルチクライアントファイルの各レコードに割り当てる内部オーナー ID 値の長さを指定します。ETID パラメータと組み合わせると、LOWNERID パラメータは、標準ファイルをマルチクライアントファイルとして再ロードしたり、ファイルのオーナー ID 長を変更したり、ファイルのレコードからオーナー ID を削除するために使用できます。

LOWNERID パラメータを指定しなければ、入力ファイル用のオーナー ID の長さは、引き続き同じ長さになります。

ETID は、マルチクライアントファイルの中にロードされる全レコードに新しいオーナー ID を割り当てます。入力ファイルにオーナー ID が含まれていない場合、つまり、マルチクライアントソースファイルからアンロードした入力ファイルでない場合、ETID パラメータを必ず指定しなければなりません。

## マルチクライアントファイルのロード／更新例

```
ADALOD LOAD FILE=20,LOWNERID=2,NUMREC=0
```

ファイル 20 をマルチクライアントファイルとして作成します。内部オーナー ID の長さは 2 バイトですが、実際のオーナー ID (ETID) は指定されていません。実際にファイルにはレコードが 1 つもロードされていません (NUMREC=0)。

```
ADALOD LOAD FILE=20,LOWNERID=2,ETID=USER1
```

ファイル 20 をマルチクライアントファイルとして作成し、指定されたレコードをすべてロードし、それにユーザー USER1 を割り当てます。内部オーナー ID の長さは 2 バイトです。

```
ADALOD UPDATE FILE=20,ETID=USER2
```

一括更新を実行して、マルチクライアントファイルであるファイル 20 にレコードを追加します。新しいレコードをすべてロードし、それに USER2 を割り当てます。

# 109

## UPDATE：レコードの追加／削除

---

■ 基本パラメータ .....	660
■ オプションのパラメータとサブパラメータ .....	542
■ 例 .....	547
■ ISN の指定形式 .....	548
■ UPDATE データとスペースの要件 .....	550
■ 拡張ファイルの一括更新 .....	551



**Caution:** ADALOD UPDATE が（スペース不足などの原因で）異常終了した場合、異常終了の前に行われたファイルの更新はバックアウトできません。このため、ADALOD UPDATE を実行する前に、ファイルで ADASAV SAVE を実行することを推奨します。

UPDATE 機能は、既存のファイルに対して、大量のレコード（ISN）を追加または削除します。1 回の UPDATE 操作で、ISN の追加と削除をどちらも行うことができます。

追加するレコードは圧縮形式（ADACMP または ADAULD の出力）でなければならず、DDEBAND/EBAND 入力データセット内に存在する必要があります。

削除する ISN は、DDISN または DELISN パラメータあるいは両方のパラメータで指定する必要があります。



### Notes:

1. Adabas ニュークリアスがアクティブの場合は、UPDATE 機能を Adabas システムファイルに対して使用できず、チェックポイントファイルまたはセキュリティファイルの更新にも使用できません。
2. マルチクライアントファイルを拡張ファイルの一部にすることはできません。また、拡張ファイルをマルチクライアントファイルに変換することもできません。

```

ADALOD UPDATE  FILE = file-number
                   SORTSIZE = size
                   TEMPSIZE = size
                   [DDISN]
                   [DELISN = isn-list ]
                   [DSREUSE = { YES | NO } ]
                   [ETID = multiclient-file-owner-id ]
                   [ISNREUSE = { YES | NO } ]
                   [LIP = { isn-pool-size | 2000 } ]
                   [LWP = { work-pool-size | 1048576 } ]
                   [MAXISN = number
                   [ACRABN = starting-rabn ]
                   [ASSOVOLUME = ' Associator-extent-volume ' ] ]
                   [MAXISN2 = number
                   [AC2RABN = starting-rabr ]
                   [ASSOVOLUME = ' Associator-extent-volume ' ] ]
                   [NOUSERABEND]
                   [NUMREC = number ]
                   [PASSWORD = ' password ' ]
                   [RESTART]
                   [SKIPREC = { number | 0 } ]
                   [SORTDEV = { device-type | ADARUN-device } ]
                   [TEMPDEV = { device-type | ADARUN-device } ]
                   [TEST]
                   [USERISN = { YES | NO } ]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FILEは、更新ファイルの番号を指定します。ADABAS拡張ファイルのコンポーネントファイルを指定すると、そのコンポーネントファイルのみが更新され、他のコンポーネントファイルは別個のUPDATE操作で更新する必要があります。

### SORTSIZE：ソートサイズ

SORTSIZEは、ソートデータセットに使用できるブロック数またはシリンダ数を指定します。

**TEMPSIZE**：一時ストレージのサイズ

TEMPSIZEは、中間データセットに使用できるブロック数またはシリンダ数を指定します。

## オプションのパラメータとサブパラメータ

**ACRABN**：アドレスコンバータの開始 RABN

ACRABNは、アドレスコンバータのスペース割り当てを、指定されたRABNで開始します。ACRABNは、ファイルのアドレスコンバータに対する増分がMAXISNで指定されている場合にのみ有効です。

**AC2RABN**：セカンダリアドレスコンバータの開始 RABN

AC2RABNを指定すると、指定されたRABNからセカンダリアドレスコンバータのスペース割り当てが開始されます。AC2RABNは、ファイルのセカンダリアドレスコンバータに対する増分がMAXISN2で指定されている場合にのみ有効です。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリISNを、セカンダリレコードが格納されているデータストレージブロックのRABNにマッピングするために使用されます。デフォルト値はありません。

**ASSOVOLUME**：アソシエータエクステントボリューム

**Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUMEは、ファイルのアドレスコンバータに対する増分がMAXISNまたはMAXISN2で指定されている場合にのみ有効です。

ASSOVOLUMEは、ファイルのアドレスコンバータエクステントを割り当てるボリュームを指定します。要求したブロック数が指定ボリュームに存在しない場合、ADALODは、ASSOVOLUMEパラメータに関係なく再度割り当てようとします。

ACRABNまたはAC2RABNを指定すると、ADALODはアドレスコンバータエクステントタイプの割り当て時にASSOVOLUMEの値を無視します。ASSOVOLUMEが指定されないと、ファイルのアソシエータスペースは、ADALODのデフォルト割り当てルールに従って割り当てられます。

**DDISN**：削除する ISN のシーケンシャルデータセットからの読み込み

DDISNを指定すると、削除するISNがDDISN/ISNシーケンシャルデータセットから読み込まれます。DDISNおよびDELISNパラメータをどちらも指定すると、2つのリストのISNが結合されます。DDISN/ISNデータセットには、可変または可変ブロック形式のレコードが存在する必要があります。詳細は、「[ISNの指定形式](#)」を参照してください。

UPDATE機能を実行すると、すべてのISNが最初に読み込まれ、ISNプールに発生順に格納されます。ISNプールサイズ（LIPで指定される）には、DDISN/ISNから読み込んだすべてのデータを格納できる大きさが必要です。次に、レコードが昇順にソートされます。範囲の重複や同じISNを二重に指定することは許可されません。処理中に検出されなかったISNは無視されます。

Adabas 拡張ファイルから ISN を削除する場合は、すべてのコンポーネントファイルの完全な ISN リストを指定できます。UPDATE 機能は、処理対象のコンポーネントファイルに適した ISN のみを自動的に選択します。

 **Note:** このパラメータは、LOB ファイルの ADALOD UPDATE 操作で指定できません。

#### DELISN：削除する ISN

DELISN は、削除するレコードの ISN のリストを指定します。DDISN および DELISN パラメータをどちらも指定すると、2つのリストの ISN が結合されます。範囲リストは次のように指定できます。

```
DELISN=10-80,90,100-110
```

範囲の重複や同じ ISN を二重に指定することは許可されません。単一 ISN または ISN 範囲を 32 個まで指定できます。Adabas 拡張ファイルから ISN を削除する場合は、すべてのコンポーネントファイルの完全なリストを指定できます。UPDATE 機能は、リストから適切な ISN を選択し、それをコンポーネントファイルから削除します。

 **Note:** このパラメータは、LOB ファイルの ADALOD UPDATE 操作で指定できません。

#### DSREUSE：データストレージの再利用

DSREUSE は、レコード削除の結果、使用可能になったデータストレージのスペースを再利用するかどうかを指定します。

このパラメータは、UPDATE 機能の実行に対してのみ有効です。DSREUSE の永続設定は変更されません。この値を指定しないと、永続設定がデフォルトになります。

#### ETID：マルチクライアントファイルオーナー ID

ETID パラメータは、既存のマルチクライアントファイルに追加されたすべてレコードに新しいオーナー ID を割り当てます。オーナー ID は、マルチクライアントファイルが最後にロードされたときに LOWNERID で指定されたオーナー ID の長さに自動的に調節されます。ETID を指定しないと、入力ソースで指定されたオーナー ID が、ロードされたすべてのレコードで保持されます。

既存のファイルがマルチクライアントファイルであり、入力ファイルがマルチクライアントファイルからアンロードされなかった場合は、ETID パラメータを指定する必要があります。既存のファイルがマルチクライアントファイルでない場合は、ETID を指定できません。

 **Note:** ETID パラメータを使用する場合、ADALOD ユーティリティはアクティブニュークリアスを必要とします。ニュークリアスは、ETID 値を内部オーナー ID 値に変換します。

#### ISNREUSE：ISN の再利用

ISNREUSE は、削除されたレコードの ISN を新しいレコードに再割り当てできるかどうかを指定します。

この ISNREUSE 設定は、UPDATE 機能の実行中にのみ有効です。ISNREUSE の永続設定は変更されません。この値を指定しないと、永続設定がデフォルトになります。

#### LIP：ISN ワークプールサイズ

LIP は、削除する ISN を収めるワークプールのサイズを指定します。このプールには、ISN ごとに 4 バイト、ISN 範囲ごとに 8 バイトが必要です。値はバイト単位で指定し、数値のみ ("2048" など) を指定するか、あるいは数値の後に "K" を付けて ("2K" など) キロバイト単位で指定します。LIP のデフォルトは、2000 バイトです。

#### LWP：ワークプールサイズ

LWP は、ディスクリプタ値のソートに使用されるワークプールの大きさを指定します。この値は、数値のみでバイト単位で指定するか、あるいは数値の後に "K" を付けてキロバイト単位で指定します。値が指定されない場合は、デフォルトで 1048576 バイト (または 1024K) が設定されます。ただし、ディスクリプタが非常に長い、またはディスクリプタの数が著しく多いファイルに対して ADALOD の実行時間を短くする場合には、より大きな値を LWP に設定すべきです。ソートデータセットとの問題を避けるため、比較的小規模なファイルを更新する場合には、LWP 値を小さめに指定してください。

最小ワークプールの大きさは、ソートデータセットのデバイスタイプごとに異なります。

ソートデバイス	最小 LWP	最小 LWP
	バイト	キロバイト
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

#### MAXISN：ファイルに割り当てる最大 ISN

MAXISN パラメータを使用し、ファイルに対して新しい ISN 設定を指定できます。現在のレコード件数と追加する ISN (レコード) 件数の合計が現在の MAXISN 設定を超える場合は、このパラメータを指定する必要があります。現在の設定を超えた値を指定することによって、アドレスコンバータに必要な追加スペースが決まり、ADALOD で新しいエクステン트가割り当てられます。MAXISN が現在の設定より小さい場合には、アドレスコンバータのスペースに変更はありません。



**Note:** NOACEXTENSION がアクティブな状態で最後にファイルをロードした場合は、ファイルの MAXISN 設定を増やすことはできません。

MAXISN 設定は、予期されるすべての拡張に適した量を増やす必要があります。これにより、アドレスコンバータエクステン트가早く消費されることを回避でき、割り当て済みのアドレスコンバータエクステン트가最大数に達したためにファイルをアンロード／リロードしたり、ADAORD REORFASSO ユーティリティを実行する必要性が発生しにくくなります。

オプションの ACRABN パラメータを使用すると、新しいアドレスコンバータエクステンツを特定の RABN 番号から開始できます。詳細は、ACRABN パラメータの説明を参照してください。

MAXISN パラメータを省略すると、古い MAXISN 値を超えた場合にのみ、新しいアドレスコンバータエクステンツが割り当てられます。

#### **MAXISN2**：ファイルのセカンダリアドレスコンバータに割り当てる最大 ISN

ADALOD 入力ファイル内にスパンドレコードが存在するかどうかに関係なく、MAXISN2 の指定は任意です。このパラメータを使用して、ISN におけるセカンダリアドレスコンバータ (AC2) のサイズが決定されます。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

ADALOD は、次の方式に従って、セカンダリ AC 内のスペースを使用できるように ISN マッピングの数を判断します。

1. セカンダリ AC がまだ割り当てられていない場合は、セカンダリ AC ブロック 1 つが割り当てられます。
2. 現在の MAXISN2 設定が MAXISN 値（予期される現在の最大プライマリ ISN 数）の 4 倍以上である場合は、追加のプライマリ ISN を割り当てるときと同じアルゴリズムを使用してセカンダリ ISN の数を決定します。
3. 上記のいずれの条件も満たさない場合、セカンダリ AC のスペースは、次の 2 つの計算のうち小さい値として割り当てられます。
  - 旧 MAXISN2 設定の 10 倍 ( $10 \times \text{oldMAXISN2}$ )
  - 旧 MAXISN2 設定と MAXISN 設定の合計 ( $\text{oldMAXISN2} + \text{MAXISN}$ )

デフォルト値はありません。

#### **NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

#### **NUMREC**：追加する最大レコード件数

NUMREC は、追加するレコード件数を制限します。NUMREC を指定すると、（入力データセットのエンドオブファイル条件によって ADALOD がすでに終了していない限り）指定数のレコードを追加した後で ADALOD は終了します。このパラメータを省略すると、すべての入力レコードが追加されます。

入力データセット内のレコード数が NUMREC の指定値より多い場合、ADALOD は NUMREC で指定されたレコード数を追加した後、コンディションコード 4 で終了します。



**Note:** このパラメータは、LOB ファイルの ADALOD UPDATE 操作で指定できません。

#### **PASSWORD：**ファイルパスワード

更新するファイルがパスワード保護されている場合は、このパラメータを使用して有効なパスワードを指定する必要があります。PASSWORD にデフォルトはありません。

#### **RESTART：**中断 ADALOD 処理の再開

RESTART は、中断された ADALOD 実行を強制的に再開させるもので、中断までに達していた最終再開位置から再スタートが行われます。再開位置とは、中間データセットから復元可能な実行処理の最終位置を意味します。

エラー条件定義によって ADALOD が中断された場合、再開が可能であるかどうかを示すメッセージが ADALOD によって発行されます。

ADALOD オペレーションを再開するときに、次の各パラメータを変更できます。

- TEMPSIZE を増やして中間データセットをより大きくすることができます。ただし、中間データセットには再開操作に必要な情報が含まれているため、この内容は変更できません。
- SORTSIZE パラメータ、SORTDEV パラメータ、およびソートデータセットを変更できます。

その他のパラメータを変更することはできません。DDEBAND/EBAND、DDFILEA/FILEA、および DDISN/ISN データセットは変更できません。

#### **SAVEDREC：**シーケンシャルファイルへの削除レコードの保存

SAVEDREC は、削除したレコードをシーケンシャルデータセットに書き込むことを指定します。データセットのフォーマットは、MODE=SHORT オプションを使用した ADAULD ユーティリティで作成されるものと同じです。



**Note:** このパラメータは、LOB ファイルの ADALOD UPDATE 操作で指定できません。

#### **SKIPREC：**スキップするレコード数

SKIPREC は、更新処理を開始する前にスキップする入力レコード数を指定します。デフォルトは 0 です（レコードがスキップされていない場合）。



**Note:** このパラメータは、LOB ファイルの ADALOD UPDATE 操作で指定できません。

#### **SORTDEV：**ソートデバイスタイプ

ADALOD はディスクリタ値のソートにソートデータセットを使用します。SORTDEV パラメータは、このデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

**TEMPDEV**：一時ストレージデバイスタイプ

ADALOD は、中間データセットを使用して中間データを格納します。TEMPDEV パラメータは、このデータセットに使用するデバイスタイプを示します。このパラメータは、使用するデバイスタイプが、ADARUN DEVICE パラメータで TEMP に割り当てられた標準のデバイスタイプと異なる場合にのみ必要です。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**USERISN**：ユーザー ISN 割り当て

USERISN=YES は、ファイルに対して USERISN オプションが有効になり、新規の各レコードに対してユーザーが入力データで ISN を指定することを示します。USERISN=NO の場合、Adabas は、それぞれの新規レコードに ISN を割り当てます。

指定された USERISN 設定は、UPDATE 機能の実行中にのみ有効です。USERISN の永続設定は変更されず、このパラメータが指定されない場合のデフォルトとなります。

ディスクリプタ値の ISN がハイパー出口によって変更されたハイパーディスクリプタを持つファイルで ADALOD UPDATE 機能を実行する場合、追加／ロード操作に対して USERISN=YES は不要になります。

USERISN=YES が指定されていないファイルからレコードを追加する場合は、ADALOD パラメータ USERISN=NO を指定し、更新するファイルに対して USERISN オプションを指定する必要があります。この機能は Adabas Text Retrieval (TRS) に役立ちます。

## 例

### 例 1

```
ADALOD UPDATE FILE=6,MAXISN=18000
ADALOD          TEMPSIZE=10,SORTSIZE=5
```

ファイル 6 にレコードを追加します。ファイルの MAXISN を 18,000 に増やします。

### 例 2

```
ADALOD UPDATE FILE=7,TEMPSIZE=10,
ADALOD          ETID=USER3, SORTSIZE=5
```

ユーザーのオーナー ID が USER3 であるレコードをマルチクライアントファイル 7 に追加します。

### 例 3

```
ADALOD UPDATE FILE=8,DELISN=1000-1999,5000-5999
ADALOD TEMPSIZE=10,SORTSIZE=5
```

ISN が 1,000～1,999 および 5,000～5,999 のレコードをファイル 8 から削除します。入力データセットが指定されている場合は、レコードを追加します。

### 例 4

```
ADALOD UPDATE FILE=6
ADALOD DDISN,SAVEDREC
ADALOD TEMPSIZE=10,SORTSIZE=5
```

ファイル 6 からレコードを削除します。削除するレコードの ISN は入力データセットに含まれています。削除したレコードを出力データセットに保存します。

### 例 5

```
ADALOD UPDATE FILE=6,DDISN,LIP=20K,SKIPREC=500
ADALOD TEMPSIZE=5,SORTSIZE=10
```

ファイル 6 のレコードを追加および削除します。削除するレコードを識別する ISN は入力データセット (DDISN) に含まれています。ISN プールサイズを 20K に設定します。入力データセットの最初の 500 レコードをスキップします。

## ISN の指定形式

---

DDISN または ISN データセットでは、ISN を 2 つの形式で指定できます。1 つ目の形式は、31 ビットの ISN が指定されているすべての場合に使用できます。レコードには、単一 ISN および ISN 範囲が混在できます。

2 つ目の形式では、32 ビットの ISN がサポートされ、バージョン 6 以上の Adabas でのみ使用できます。各レコードでは、単一 ISN (最初のフルワードの X'00000000' で示される) または ISN 範囲 (最初のフルワードの X'FFFFFFFF' で示される) のどちらかを指定できます。

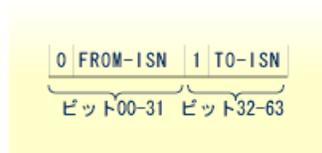
レコードの最初のフルワードに X'00000000' または X'FFFFFFFF' 以外の値が含まれている場合は、31 ビット形式であると見なされます。DDISN/ISN データセットには、両方の形式のレコードが含まれていてもかまいません。

## 形式1：31ビット形式

単一 ISN には 4 バイト必要です。上位ビットを 0 に設定し、ビット 01～31 に ISN を指定します。



ISN 範囲には 8 バイト必要です。最初の 4 バイトで、範囲の最初の ISN を単一 ISN として指定します。次の 4 バイトで、上位ビットを 1 に設定し、最後の ISN を指定します。

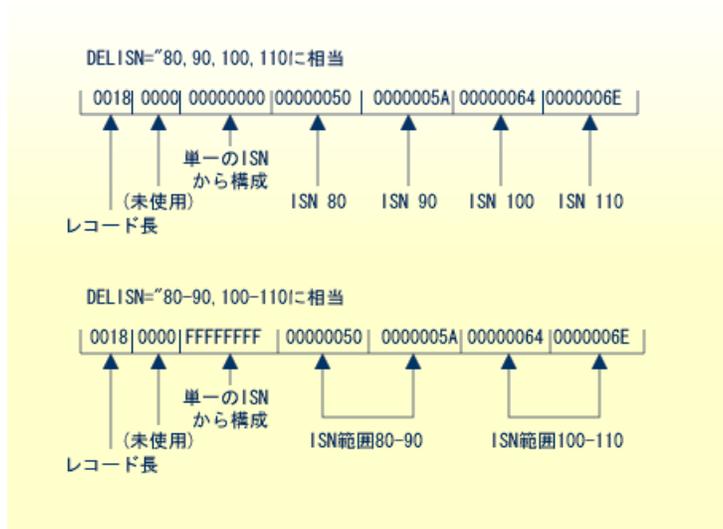


次の例に、DELISN=10～80、90、100～110 に相当する値を含む可変長レコードを示します。



## 形式2：32ビット形式

32ビット形式の場合、各レコードの最初のフルワードは、レコードに単一 ISN または ISN 範囲のどちらが含まれているかを示します。単一 ISN を示すには、最初のフルワードにゼロ (X'00000000') を指定し、ISN 範囲を示すには 1 (X'FFFFFFFF') を指定します。次の例では、最初のレコードに単一 ISN が含まれ、2 番目のレコードに ISN 範囲が含まれています。2 つのレコードは、最初のフルワードを除き同一です。



## UPDATE データとスペースの要件

次の一般情報は、UPDATE 操作に関するデータ要件および ADALOD UPDATE によるスペース割り当ての方法について説明します。スペース割り当ての詳細については、『Adabas DBA リファレンスマニュアル』を参照してください。

### UPDATE 操作の入力データ

追加するレコードは、ADACMP または ADAULD ユーティリティで作成された圧縮形式のデータレコードである必要があります。ADACMP の実行に使用するデータ定義は、フィールド定義テーブル (FDT) に含まれているように、レコードを追加するファイルの定義と一致している必要があります。

 **Note:** 暗号化ファイルに追加するレコードは、ファイル内にすでに存在するレコードに対して使用されたサイファコードと同じものを使用して、すでに暗号化されている必要があります。

削除するレコードの ISN は、DELISN パラメータまたは入力データセットのどちらで指定してもかまいません。入力データセットで指定する場合には、各 ISN を 4 バイトの 2 進数として指定する必要があります。データセットのレコード形式は、VARIABLE BLOCKED である必要があります。必要に応じて、Adabas 拡張ファイルに対して追加または削除するすべての ISN を指定できます。UPDATE 機能は、処理中のコンポーネントファイルに適した ISN を選択します。

### UPDATE のスペース割り当て

レコードを追加する場合に、MAXISN に大きい値が指定されていると、ADALOD は追加のアドレスコンバータエクステントを割り当てます。新しいエクステントのサイズは、新しい MAXISN と古い MAXISN の設定値の差に基づいています。新しいエクステントのスペース

が不足しているか、割り当て済みのエクステントがすでに最大数に達しているか、エラーメッセージを出力して処理が終了します。

データストレージエクステントの追加が必要な場合、ADALOD はファイルに対して現在割り当てられているデータストレージエクステントの合計サイズの約 25 % の大きさの追加エクステントを割り当てます。アドレスコンバータの場合、追加されたエクステントのスペースが不足しているか、割り当て済みのエクステントが最大数に達しているか、エラーメッセージを出力して処理が終了します。

#### UPDATE ディスクリプタ情報の生成

レコードを追加する場合、ADALOD UPDATE は、すべてのディスクリプタ値および対応する新しいレコードの ISN のリストを生成し、この情報を中間データセットに書き込みます。

#### UPDATE を使用したアソシエータの更新

入力処理の前に、ADALOD UPDATE は、ファイルの既存のノーマルインデックスを中間データセットにコピーしますが、削除する ISN のディスクリプタ値は削除します。

ADALOD は、入力フェーズで中間データセットに書き込まれた情報をソートし、ソートされた値を現在のノーマルインデックスと結合します。この処理中にノーマルインデックスがリオーダされ、アソシエータブロックパディングファクタがブロックごとに再設定されます。その時点で、新しいアッパーインデックスが作成されます。

ディスクリプタの更新の結果発生した、部分的に占有されたブロックの空スペースは再利用されます。このため、インデックスの最後に存在する空ブロック数が増える可能性があります。リオーダ処理の結果、ノーマルインデックスやアッパーインデックスの1つ以上のエクステントが空になることがあります。ADALOD はこれらのエクステントサイズの圧縮、削除、変更を行いません。

ノーマルインデックスまたはアッパーインデックスに新しい空スペースが必要な場合は、1つ以上の追加エクステントが割り当てられます。割り当てられた各追加エクステントのサイズは、インデックスに対して現在割り当てられている合計スペースの約 25 % です。追加エクステントのスペースが不足しているか、割り当て済みのエクステントが最大数に達しているか、ADALOD はエラーメッセージを出力して終了します。

## 拡張ファイルの一括更新

ADALOD UPDATE を使用して拡張ファイルを一括更新する場合、個々のコンポーネントファイルに対してレコードの追加または削除を行う必要があります。ただし、各コンポーネントファイルの処理は、同じ ADALOD コマンドを使用して行うことができます。

DELISN または DDISN を指定してレコードを削除する場合は、すべてのコンポーネントファイルから削除するすべての ISN のリストを指定できます。ADALOD は、現在処理中のコンポーネントファイルに対して指定された適切な範囲の ISN 値のみを自動的に選択します。

USERISN=YES で新規レコードを追加するときも同じです。

USERISN=NOを指定して拡張ファイルに新しいレコードを追加する場合に、解放されたISNが存在しないと、ISN範囲を広げることができない（すべてのコンポーネントファイルに対してNOACEXTENSIONが有効）ため、新しいアドレスコンバータエクステンツを割り当てることができません。その代り、ADALODは、エンドオブファイルになったときと同じようにインデックスを作成します。ロードされなかった残りのレコードは後からSKIPRECパラメータを使用して別のコンポーネントファイルに追加することができます。

ADALODは、各コンポーネントファイル間に渡ってのユニークディスクリプタ値のチェックは行いません。

#### 例

次の例に、拡張ファイルに対する一括更新を示します（完全なジョブの関連部分のみを示しています）。

```
      .
      .
//DDEBAND DD DSN=MOREDATA.LOAD.PART1-2,...
//DDKARTE DD *
ADALOD UPDATE FILE=40,USERISN=YES
ADALOD      DELISN=9000001-9500000,12000001-14000000
ADALOD      SORTSIZE=...,TEMPSIZE=...
      .
      .
//DDEBAND DD DSN=MOREDATA.LOAD.PART1-2,...
//DDKARTE DD *
ADALOD LOAD FILE=41,USERISN=YES
ADALOD      DELISN=9000001-9500000,12000001-14000000
ADALOD      SORTSIZE=...,TEMPSIZE=...
      .
      -
```

# 110 ローダーストレージの要件および使用

## 静的ストレージ

静的	タイプ*	サイズ
ADARUN、ADALOD モジュール	A	約 180 KB

## 動的ストレージ

動的	タイプ*	サイズ
ソートワークプール	A	LWP
汎用ワークプール	A	6* (アソシエータブロックサイズ)
アソシエータの入力/出力バッファ	A	アソシエータブロックサイズ
ISN プール	A	LIP
データストレージの入力/出力バッファ	A	データストレージブロックサイズ
AC ビットマップ	A	4 KB
中間の入力/出力バッファ	A	中間ブロックサイズ
DVT 分割	A	中間ブロックサイズ 窠「 ディスクリプタ数
内部ディスクリプタテーブル	A	ディスクリプタ数 窠「 74
DDEBAND/EBAND の入力/出力バッファ	O	DDEBAND/EBAND ブロックサイズ
DDOLD/OLD の入力/出力バッファ	O	DDOLD/OLD ブロックサイズ
DDISN/ISN データの入力/出力バッファ	O	DDISN/ISN ブロックサイズ
中間オーバーフローに書き込む 必要がある場合の入力/出力バッファ	O	DDFILEA/FILEA ブロックサイズ

\* タイプ A を常に使用し、必要に応じてタイプ O を使用します。



# 111 TEMP データセットスペースの使用状況

---

- シーケンシャル TEMP データセット ..... 556

ADALOD は、次の情報を格納するために TEMP データセットを使用します。

- 再スタート情報
- 削除する各レコードのデータストレージ RABN/ISN (UPDATE のみ)
- 処理開始時のノーマルインデックスの内容 (UPDATE のみ)
- 入力データセットから取得されたディスクリプタ値
- ADAM オーバーフローエリア (ADAM ファイルのみ)

## シーケンシャル TEMP データセット

---

入力データセットからディスクリプタ値を収集しているときに TEMP データセットがいっぱいになると、ADALOD は、残りのディスクリプタをシーケンシャル中間ファイル DD/FILEA に書き込みます (JCL で指定されている場合)。ディスクリプタは、新しいインデックスが作成された時点で読み戻されます。

DD/FILEA コールが実際に行われると、すべてのディスクリプタ値を保持できるだけの十分な大きさの TEMP データセットを指定した場合に比べ、ADALOD の処理速度が著しく低下します。DD/FILEA TEMP データセットは、ADALOD 処理中にすべてのディスクリプタに対して十分なスペースを確保するための安全策としてのみ使用するのが一般的です。したがって、DD/FILEA TEMP ファイルを指定することにより、中間エリアのオーバーランが原因となる ADALOD のアベンドを避けることができます。



### Notes:

1. ADALOD は、DD/EBAND 入力ファイルから DD/FILEA にディスクリプタ値のみを書き込みます。
2. 通常の TEMP データセットは、ディスクリプタごとにすべての値を保持できるだけの十分な大きさが必要です。

ISN を削除するためにのみ ADALOD UPDATE を実行している場合は、次の 3 手順の合計として TEMP データセットのサイズを計算する必要があります。

1. TEMP データセットのパート 1 には、以下が含まれます。
  - RABN 1：再スタート情報
  - RABN 2：ファイルの FCB
  - RABN 3 - n：DETAB (DE の件数 \* 116) 例えば、1つのエレメントは 116 バイトです。DE の件数は、FDT に含まれる DE の個数になります。

- TEMP データセットのパート 2 には、削除する ISN のリストおよびそれに属する DS-RABN (エレメントごとに 6 バイト) が含まれます。計算式は次のとおりです。

$$(6 * \text{to-be-deleted-ISNs}) / (\text{TEMPBLKSIZE} - 16) = \text{needed-blocks}$$

- パート 3 には、完全な NI が含まれます。

$$(\text{count NI-BLKS} * \text{ASSOBLKSIZE}) / (\text{TEMPBLKSIZE} - 16) = \text{needed blocks}$$



# 112

## ADALOD スペース／統計レポート

LOAD または UPDATE 操作中に、ADALOD はメッセージ出力データセット (z/OS および z/VM システムでは DDDRUCK、VSE システムでは SYS009、BS2000 では SYSOUT) にレポートを出力します。レポートには次の情報が表示されます。

- 実行する ADALOD 機能 (LOAD または UPDATE)、および影響を受けるデータベース／ファイル
- NI/UI サイズの見積り (NI/UISIZE パラメータが指定されていない場合に、LOAD 機能に対してのみ出力される)
- Adabas コンポーネント別の使用可能および使用済みのファイルスペース (LOAD 機能に対してのみ出力される)
- ファイルに割り当てられている現在の RABN (LOAD 機能に対してのみ出力される)
- ファイル処理統計 (処理済みレコードおよび使用済みシステムストレージ)

**ADALOD LOAD** レポートの例：

```
PARAMETERS:ADALOD  LOAD FILE...
.
.

FUNCTION TO BE EXECUTED:

LOAD FILE NUMBER      7  (MYOWNFILE)
INTO DATABASE         0013 (MYBESTDB)

AVAILABLE SPACE:

(LOAD function only)

      I FILE I  DEV  I  NUMBER OF  I      FROM      TO      I
      I LAY- I  TYPE I  BLOCKS    I      RABN    RABN  I
      I OUT  I      I              I              I
```

```

I-----I-----I-----I-----I
I ASSO I 3380 I 2695 I 137 2831 I
I DATA I 3380 I 1339 I 3 1341 I
-----
    
```

ESTIMATED NORMAL INDEX SIZE = 37 BLOCKS

ESTIMATED UPPER INDEX SIZE = 8 BLOCKS

TOP ISN = 773, MAX ISN EXPECTED = 1335

I FILE	I DEV	I LIST	I ALLOC	I FROM	TO	I UNUSED
I LAY-	I TYPE	I TYPE	I SPACE	I RABN	RABN	I SPACE
I OUT	I	I	I (BLOCKS)	I		I (BLOCKS)
I ASSO	I 3380	I AC	I 2	I 137	138	I 0
I ASSO	I 3380	I UI	I 8	I 139	146	I 0
I ASSO	I 3380	I NI	I 37	I 147	183	I 15
I DATA	I 3380	I DS	I 60	I 3	62	I 48

PROCESSING STATISTICS

```

773 INPUT RECORDS PROCESSED
14 BLOCKS USED ON TEMP-DATASET (0%)
0 BLOCKS USED ON SORT PART 1 (0%)
0 BLOCKS USED ON SORT PART 2 (0%)
51824 BYTES OF STORAGE USED TO STORE RECORDS
    
```

# 113 JCL/JCS の必要項目と指定例

---

■ ユーザー出口を使用した照合 .....	562
■ BS2000 .....	562
■ z/OS .....	566
■ z/VM .....	568
■ VSE .....	571

ここでは、BS2000、z/OS、z/VM、およびVSEシステムにおけるADALODの実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

 **Note:** オプションのRecovery Aid (RLOG) で実行している場合は、すべての中間データセットがジョブ制御でもカタログされる必要があります。

このchapterでは、次のトピックについて説明します。

## ユーザー出口を使用した照合

ADALOD 実行中に照合ユーザー出口を使用する場合は、ユーティリティの実行時に ADARUN CDXnn パラメータを指定する必要があります。

UES (ユニバーサルエンコーディングサポート) と合わせて使用する場合の照合ディスクリプタユーザー出口の構文は次のとおりです。

```
ADARUN CDXnn= exit-name
```

ここでは次の内容を表しています。

nn	照合ディスクリプタ出口の番号。01 から 08 までの 2 桁の 10 進数。
exit-name	照合ディスクリプタ出口で制御を持つユーザールーチン名。最大 8 文字。

照合ディスクリプタ出口に指定できるプログラムは 1 つだけです。最大 8 個の照合ディスクリプタ出口を任意の順序で指定できます。詳細は『Adabas DBA リファレンスマニュアル』を参照してください。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	
一時エリア	DDTEMPR1	ディスク	
TEMP オーバーフロー (任意)	DDFILEA	ディスク/テープ	中間データセットが小さすぎる場合にディスクリプタ値を格納

データセット	リンク名	ストレージ	追加情報
ソートエリア	DDSORTR1	ディスク	大規模なファイルの場合は、ソートエリアを2つのボリュームに分割 <sup>1</sup>
ソートエリア	DDSORTR2	ディスク	
リカバリログ (RLOG)	DDRLOGR1	ディスク	リカバリログオプションの使用時に必要
圧縮データ	DDEBAND	ディスク/テープ	ADACMP または ADAULD ユーティリティの出力
削除する ISN	DDISN	ディスク/テープ	削除する ISN <sup>2</sup>
削除レコード	DDOLD	ディスク/テープ	削除レコード (存在する場合) <sup>3</sup>
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADALOD パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/ DDPRINT		ADALOD レポート (「メッセージ およびコード」参照)
ADALOD メッセージ	SYSLST/ DDDRUCK		メッセージおよびコード



#### Notes:

1. 大規模ファイルをソートする場合、ソートデータセットが2つのボリュームを占めているか、2つのソートデータセットが指定されている場合は、パフォーマンスが向上します。両方のデータセットが同一のデバイスタイプ上に存在し (SORTDEV パラメータ)、各データセットは SORTSIZE パラメータで指定したサイズの正確に半分の大きさでなければなりません。
2. ISN ごとに4バイト、REC-FORM=VB、BUFF-LEN はシーケンシャルファイルの説明どおり、REC-SIZE の最大値は BUFF-LEN - 4 になります。(ISP 形式の場合、REC-FORM は RECFM、BUFF-LEN は BLKSIZE、REC-SIZE は LRECL になります)
3. REC-FORM=VB、BUFF-LEN はシーケンシャルファイルの説明どおり、REC-SIZE の最大値は BUFF-LEN - 4 になります。(ISP 形式の場合、REC-FORM は RECFM、BUFF-LEN は BLKSIZE、REC-SIZE は LRECL になります)

## ADALOD JCL の例 (BS2000)

### ファイルのロード

SDF フォーマットの場合：

```

/.ADALOD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A L O D LOAD FILE
/REMARK *
/ASS-SYSLST L.LOD.LOAD
/ASS-SYSDTA *SYSCMD

```

```
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAYyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDTEMPR1,ADAYyyyy.TEMP
/SET-FILE-LINK DDSORTR1,ADAYyyyy.SORT
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADALOD,DB=yyyyyy,IDTNAME=ADABAS5B
ADALOD LOAD FILE=1
ADALOD NAME= TESTFILE-1
ADALOD MAXISN=10000,DSSIZE=10
ADALOD TEMPSIZE=100,ORTSIZE=50
/LOGOFF SYS-OUTPUT=DEL
```

**ISP** フォーマットの場合：

```
/.ADALOD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A L O D LOAD FILE
/REMARK *
/SYSFILE SYSLST=L.LOD.LOAD
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyy.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAYyyyy.WORK,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAYyyyy.TEMP,LINK=DDTEMPR1
/FILE ADAYyyyy.SORT,LINK=DDSORTR1
/FILE CMP.AUS,LINK=DDEBAND

/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADALOD,DB=yyyyyy,IDTNAME=ADABAS5B
ADALOD LOAD FILE=1
ADALOD NAME= TESTFILE-1
ADALOD MAXISN=10000,DSSIZE=10
ADALOD TEMPSIZE=100,ORTSIZE=50
/LOGOFF NOSPOOL
```

**更新**

**SDF** フォーマットの場合：

```

/.ADALOD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A L O D LOAD FILE
/REMARK *
/DELETE-FILE LOD.ISN
/SET-JOB-STEP
/CREATE-FILE LOD.ISN,PUB(SPACE=(48,48))
/SET-JOB-STEP
/DELETE-FILE LOD.OLD
/SET-JOB-STEP
/CREATE-FILE LOD.OLD,PUB(SPACE=(480,48))
/SET-JOB-STEP
/ASS-SYSLST L.LOD.LOAD
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAYyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDTEMPR1,ADAYyyyy.TEMP
/SET-FILE-LINK DDSORTR1,ADAYyyyy.SORT
/SET-FILE-LINK DDEBAND,CMP.AUS
/SET-FILE-LINK DDISN,LOD.ISN
/SET-FILE-LINK DDOLD,LOD.OLD
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADALOD,DB=yyyyy,IDTNAME=ADABAS5B
ADALOD UPDATE FILE=1,DDISN,SAVEDREC
ADALOD TEMPSIZE=100,SORTSIZE=50
ADALOD DELISN=100 199,230,301 399
/LOGOFF SYS-OUTPUT=DEL

```

### ISP フォーマットの場合：

```

/.ADALOD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A L O D MASS UPDATE
/REMARK *
/SYSFILE SYSLST=L.LOD.UPDA
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAYyyyy.WORK ,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAYyyyy.TEMP ,LINK=DDTEMPR1
/FILE ADAYyyyy.SORT ,LINK=DDSORTR1
/FILE CMP.AUS,LINK=DDEBAND
/FILE LOD.ISN,LINK=DDISN ,SPACE=(48,48)
/FILE LOD.OLD,LINK=DDOLD ,SPACE=(480,48)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADALOD,DB=yyyyy,IDTNAME=ADABAS5B

```

```
ADALOD UPDATE FILE=1,DDISN,SAVEDREC
ADALOD TEMPSIZE=100,SORTSIZE=50
ADALOD DELISN=100 199,230,301 399
/LOGOFF NOSPOOL
```

## z/OS

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	Adabas ニュークリアスがアクティブでない場合にのみ必要
一時エリア	DDTEMPR1	ディスク	
TEMP オーバーフロー (任意)	DDFILEA	ディスク/テープ	中間データセットが小さすぎる場合にディスクリプタ値を格納
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大規模なファイルを使用している場合は、ソートエリアを2つのボリュームに分割 <sup>1</sup>
リカバリログ (RLOG)	DDRLOGR1	ディスク	リカバリログオプションの使用時に必要
圧縮データ	DDEBAND	ディスク/テープ	ADACMPまたはADAULDユーティリティの出力
削除する ISN	DDISN	ディスク/テープ	削除する ISN <sup>2</sup>
削除レコード	DDOLD	ディスク/テープ	削除レコード (存在する場合) <sup>3</sup>
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADALOD パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	ADALOD レポート (「メッセージおよびコード」参照)
ADALOD メッセージ	DDDRUCK	プリンタ	メッセージおよびコード



### Notes:

1. 大規模ファイルをソートする場合、ソートデータセットが2つのボリュームを占めているか、2つのソートデータセットが指定されている場合は、パフォーマンスが向上します。2つのボリュームを使用する場合、各ボリュームは、SORTSIZE パラメータで指定した大きさの正確に半分の大きさでなければなりません。2つのデータセットを使用する場合、どちらも同じデバイスタイプ (SORTDEV パラメータ) で定義しなければなりません。
2. ISN 当たり 4 バイト、RECFM=VB、BLKSIZE はシーケンシャルファイルの説明のとおり、LRECL の最大値は BLKSIZE - 4 に等しくなります。
3. RECFM=VB、BLKSIZE はシーケンシャルファイルの説明のとおり、LRECL の最大値は BLKSIZE - 4 に等しくなります。

## ADALOD JCL の例 (z/OS)

ADAM ファイルまたは Adabas デモファイルをロードする他の ADALOD 例については、MVSJOBS データセットの ADALODE、ADALODA、ADALODM、および ADALODV も参照してください。

ファイルのロード

この例については、MVSJOBS データセットの ADALOD を参照してください。

```
//ADALOD      JOB
//*
//*      ADALOD: LOAD FILE
//*
//LOD         EXEC PGM=ADARUN
//STEPLIB     DD      DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//*
//DDASSOR1    DD      DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1    DD      DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1    DD      DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDTEMPR1    DD      DISP=OLD,DSN=EXAMPLE.DByyyyy.TEMPR1 <=== TEMP
//DDSORTR1    DD      DISP=OLD,DSN=EXAMPLE.DByyyyy.SORTR1 <=== SORT
//DDEBAND     DD      DISP=OLD,DSN=EXAMPLE.DByyyyy.DDEBAND <=== INPUT
//DDDRUCK     DD      SYSOUT=X
//DDPRINT     DD      SYSOUT=X
//SYSUDUMP    DD      SYSOUT=X
//DDCARD      DD      *
ADARUN PROG=ADALOD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE     DD      *
ADALOD LOAD FILE=1
ADALOD NAME='TESTFILE-1'
ADALOD MAXISN=10000,DSSIZE=10
ADALOD TEMPSIZE=100,SORTSIZE=100
/*
```

更新

この例については、MVSJOBS データセットの ADALODMU を参照してください。

```
//ADALODMU JOB
//*
//*      ADALOD: MASS UPDATE
//*
//LOD      EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1  <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1  <=== DATA
//DDTEMPR1 DD DISP=OLD,DSN=EXAMPLE.DByyyyyy.TEMPR1  <=== TEMP
//DDSORTR1 DD DISP=OLD,DSN=EXAMPLE.DByyyyyy.SORTR1  <=== SORT
//DDEBAND DD DISP=OLD,DSN=EXAMPLE.DByyyyyy.DDEBAND  <=== INPUT
//DDISN DD DISP=OLD,DSN=EXAMPLE.DByyyyyy.DDISN      <=== ISNS TO DEL
//DDOLD DD DISP=(NEW,CATLG),DSN=EXAMPLE.DByyyyyy.DDOLD, <=== DEL REC
//          SPACE=(TRK,(100,20),RLSE),UNIT=DISK,VOL=SER=VOLvvv
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADALOD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADALOD UPDATE FILE=1,LWP=400K,SAVEDREC
ADALOD TEMPSIZE=100,ORTSIZE=100
ADALOD DELISN=100-199,230,301-399
/*
```

## z/VM

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	
一時エリア	DDTEMPR1	ディスク	
TEMP オーバーフロー (任意)	DDFILEA	ディスク/テープ	中間データセットが小さすぎる場合にディスクリプタ値を格納
ソートエリア	DDSORTR1	ディスク	大きなファイルを使用する場合、ソートエリアは2つのボリュームに分割 <sup>1</sup>
ソートエリア	DDSORTR2	ディスク	
リカバリログ (RLOG)	DRLOGR1	ディスク	リカバリログオプションの使用時に必要

データセット	DD名	ストレージ	追加情報
圧縮データ	DDEBAND	ディスク/テープ	ADACMP または ADAULD ユーティリティの出力
削除する ISN	DDISN	ディスク/テープ	削除する ISN <sup>2</sup>
削除レコード	DDOLD	ディスク/テープ	削除レコード (存在する場合) <sup>3</sup>
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADALOD パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	ADALOD レポート (「メッセージおよびコード」を参照)
ADALOD メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード



#### Notes:

1. 大規模ファイルをソートする場合、ソートデータセットが2つのボリュームを占めているか、2つのソートデータセットが指定されている場合は、パフォーマンスが向上します。両方のデータセットが同一のデバイスタイプ上に存在し (SORTDEV パラメータ)、各データセットは SORTSIZE パラメータで指定したサイズの正確に半分の大きさでなければなりません。
2. ISN 当たり 4 バイト、RECFM=VB、BLKSIZE はシーケンシャルファイルの説明のとおり、LRECL の最大値は BLKSIZE - 4 に等しくなります。
3. RECFM=VB、BLKSIZE はシーケンシャルファイルの説明のとおり、LRECL の最大値は BLKSIZE - 4 に等しくなります。

## ADALOD JCL の例 (z/VM)

### ファイルのロード

```

DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDWORKR1,DSN=ADABASV.V.WORK,VOL=WORKV1
DATADEF DDTEMPR1,DSN=ADABASV.V.TEMP,VOL=TEMPV1
DATADEF DDSORTR1,DSN=ADABASV.V.SORT,VOL=SORTV1
DATADEF DDEBAND,DSN=FILE001.LODD001,MODE=A
DATADEF DDPRINT,DSN=ADALOD.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADALOD.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNLOD.CONTROL,MODE=A
DATADEF DDKARTE,DSN=FILE001.LODC001,MODE=A
ADARUN

```

### RUNLOD CONTROL A1 の内容

```
ADARUN  PROG=ADALOD,DEVICE=dddd,DB=yyyyyy
```

### FILE015 LODC001 A1 の内容

```
ADALOD  LOAD  FILE=1
ADALOD      NAME= 'TESTFILE-1 '
ADALOD      MAXISN=50000 ,DSSIZE=10
ADALOD      TEMPSIZE=100 ,SORTSIZE=50
```

### 更新

```
DATADEF  DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF  DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1
DATADEF  DDWORKR1,DSN=ADABASVv.WORK,VOL=WORKV1
DATADEF  DDTEMPR1,DSN=ADABASVv.TEMP,VOL=TEMPV1
DATADEF  DDSORTR1,DSN=ADABASVv.SORT,VOL=SORTV1
DATADEF  DDEBAND,DSN=ADALOD.LODD015,MODE=A
DATADEF  DDISN,DSN=ADALOD.ISN,MODE=A
DATADEF  DDOLD,DSN=ADABASVv.OLDISN,MODE=A
DATADEF  DDPRINT,DSN=ADALOD.DDPRINT,MODE=A
DATADEF  DUMP,DUMMY
DATADEF  DDDRUCK,DSN=ADALOD.DDDRUCK,MODE=A
DATADEF  DDCARD,DSN=RUNLOD.CONTROL,MODE=A
DATADEF  DDKARTE,DSN=UPDATE.CONTROL,MODE=A
ADARUN
```

### RUNLOD CONTROL A1 の内容

```
ADARUN  PROG=ADALOD,DEVICE=dddd,DB=yyyyyy
```

### UPDATE CONTROL A1 の内容

```
ADALOD  UPDATE  FILE=1,DDISN,SAVEDREC
```

ADALOD            TEMPSIZE=100 , SORTSIZE=50  
 ADALOD            DELISN=100-199 , 230 , 301-399

## VSE

データセット	記号	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク		1
データストレージ	DATARn	ディスク		1
ワーク	WORKR1	ディスク	1	ニュークリアスがアクティブでない場合に必要
圧縮データ	EBAND	テープ ディスク	SYS010 1	
リカバリログ (RLOG)	RLOGR1	ディスク		リカバリログオプションの使用時に必要
一時エリア	TEMPR1	ディスク	1	
TEMP オーバーフロー (任意)	FILEA	テープ ディスク	SYS012 1	中間データセットが小さすぎる場合にディスクリプタ値を格納
ソートエリア	SORTR1	ディスク		大きなファイルを使用する場合、ソートエリアは2つのボリュームに分割 <sup>2</sup>
削除する ISN	ISN	テープ ディスク	SYS016 1	削除する ISN
削除レコード	OLD	テープ ディスク	SYS014 1	削除済み ISN
ADALOD メッセージ	--	プリンタ	SYS009	ADALOD レポート (「メッセージおよびコード」参照)
ADARUN メッセージ	--	プリンタ	SYSLST	メッセージおよびコード
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 1	
ADALOD パラメータ	-	リーダー	SYSIPT	



### Notes:

- すべてのプログラマ論理ユニットを使用できます。
- 大規模ファイルをソートする場合、ソートデータセットが2つのボリュームに分割されているとパフォーマンスが向上することがあります。2つのボリュームを使用する場合、各ボリュームは、SORTSIZE パラメータで指定した大きさの正確に半分の大きさでなければなりません。2つのデータセットを使用する場合、どちらも同じデバイスタイプ (SORTDEV パラメータ) で定義しなければなりません。

## ADALOD JCS の例 (VSE)

VSE プロシージャ (PROC) の内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

ファイルのロード

この例については、メンバ ADALOD.X を参照してください。

```
* $$ JOB JNM=ADALOD,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADALOD
*          SAMPLE FILE LOAD
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYSTEM,TAPE
// PAUSE MOUNT LOAD INPUT FILE ON TAPE cuu
// TLBL EBAND,'DEMO.FILE'
// MTC REW,SYS010
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADALOD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADALOD LOAD FILE=1
ADALOD NAME='TESTFILE-1'
ADALOD MAXISN=10000,DSSIZE=10
ADALOD TEMPSIZE=100,SORTSIZE=100
/*
/&
* $$ EOJ
```

更新

この例については、メンバ ADALODMU.X を参照してください。

```
* $$ JOB JNM=ADALODMU,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADALODMU
*          MASS UPDATE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,DISK,VOL=DISK01,SHR
// ASSGN SYS014,DISK,VOL=DISK02,SHR
// ASSGN SYS016,DISK,VOL=DISK03,SHR
// DLBL EBAND,'FILE.INPUT',,SD
// EXTENT SYS010,DISK01,1,0,sssss,nnnnn
// DLBL OLD,'FILE.OLD',,SD
// EXTENT SYS014,DISK02,1,0,sssss,nnnnn
// DLBL ISN,'FILE.ISN',,SD
```

```
// EXTENT SYS016,DISK03,1,0,sssss,nnnnn
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADALOD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADALOD UPDATE FILE=1,LWP=400K,SAVEDREC
ADALOD TEMPSIZE=100,SORTSIZE=100
ADALOD DELISN=100-199,230,301-399
/*
/&
* $$ EOJ
```



# 114 ADAMER ユーティリティ：ADAM 見積り

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ADAM アクセス回数の見積り
- ADAMER 出力レポートの内容
- JCL/JCS の必要項目と指定例



# 115 機能概要

---

ADAMER ユーティリティは、ADAM ディスクリプタを使ってレコードを検索し、読み込むのに必要なデータストレージのアクセス回数を示す統計表を作成します。この情報は次のことを判断するのに使用します。

- 標準の Adabas アクセス方法に対して、ADAM ディスクリプタを使用したレコード検索に必要なアクセス回数が、ADAM オプションを使用することによって削減できるかどうか判断します。
- ADAM ディスクリプタをランダムマイジングして、最適レコード分布を行うのに必要なデータストレージスペースを決定します。

ADAMER の入力データは、ADACMP または ADAULD ユーティリティによって生成されたファイルの圧縮レコードを含むデータセットです。

ADAM ディスクリプタとして使用するフィールドは、ADAMDE パラメータで指定します。マルチプルバリューフィールドやピリオディックグループ内のフィールドは使用できません。ランダムマイジングのベースとして、レコードに割り当てられる ISN をディスクリプタの代わりに使用できます (ADAMDE=ISN パラメータ)。

ADAM ディスクリプタは各レコードでユニークな値をもたなければなりません。これは、ADAM ディスクリプタに重複した値が存在すると、ADALOD ユーティリティの ADAM オプションでファイルを正常にロードできないからです。ADAMER ユーティリティには、ユニーク (UQ) に定義されたディスクリプタフィールドが必要ですが、このユーティリティは値がユニークかどうかのチェックを行いません。ディスクリプタ値がユニークかどうかのチェックは、ADALOD ユーティリティが ADAM ファイルとしてファイルをロードするときに行います。

BITRANGE パラメータを用いて、ランダムマイジングアルゴリズムの入力として使用する前に、各 ADAM ディスクリプタ値からビット数を切り捨てるように指定できます。こうすれば、同じ値で始まる ADAM ディスクリプタ値 (例えば、40643210、40643220、40643344) をもつレコードをデータストレージ内の同一物理ブロックにロードできます。この手法により、ADAM ディスクリプタをレコードの読み込み順の制御に使用すると、ファイルの順次読み込みを最適化でき、チェック桁のような意味のない情報を除くこともできます。



# 116 ADAM アクセス回数の見積り

---

■ 基本パラメータ .....	580
■ オプションパラメータ .....	580
■ 例 .....	582

```

ADAMER ADAMDE = { descriptor | ISN }
    MAXISN = maximum-number-of-records
    [BITRANGE = { minimum | 0 } { maximum | 18 } { increment | 2 } ]
    [DATADEV = { device-type | ADARUN-device } ]
    [DATAPFAC = padding-factor ]
    [DATASIZE = minimum, maximum [, increment ] ]
    [NOUSERABEND]
    [NUMREC = { number-of-records | all-records } ]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### ADAMDE：ADAM キー

ADAM キーとして使用するディスクリプタを指定します。ISN を指定すると、ADAMER は各入力レコードの ISN をランダムマイジングアルゴリズムの入力として使用します。

ADAM ディスクリプタはフィールド定義テーブル (FDT) に存在する必要があり、ユニークディスクリプタ (UQ) として定義する必要があります。サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、照合ディスクリプタ、またはフォネティックディスクリプタは指定できません。また、ディスクリプタには NU オプションも指定できず、MU フィールド、ピリオディックグループ内のフィールド、可変長フィールドも無効です。

### MAXISN：ファイルに割り当てる最大 ISN

ファイルに含まれると予期されるレコードの総数です。

MAXISN には、最初からロードされるレコード数だけでなく、ファイルに追加することが予期されるレコード数も含めておく必要があります。

## オプションパラメータ

### BITRANGE：ADAM キーのビット切り捨て

ADAM ランダムマイジングアルゴリズムへの入力として ADAM ディスクリプタ値を使用する前に、ADAM ディスクリプタ値から切り捨てるビット数の最小値、最大値、および増分値です。ビットは、常に圧縮値の右端から切り捨てられます。

ADAMER の実行ごとに、最大 20 の異なるビット切り捨てが有効です。

## 例

次のように指定すると、統計表を生成する各データストレージサイズに対して0ビット、2ビット、4ビットの切り捨てが行われます。

```
BITRANGE=0,4,2
```

このパラメータを省略すると、デフォルトのBITRANGEとして0、18、2が使用されます。

**DATADEV**：データストレージデバイスタイプ

データストレージに使用するデバイスタイプです。DATADEVパラメータを省略すると、ADARUN DEVICEパラメータで指定されたデバイスタイプがデフォルトになります。

**DATAPFAC**：データストレージパディングファクタ

ファイルに使用するデータストレージパディングファクタです。指定する数値は、ファイルの初期ロード中に使用しない各データストレージ物理ブロックを%で表します。1から90の値が指定できます。

このパラメータを省略すると、ADAMERの実行中に10%のパディングファクタが使用されます。

**DATASIZE**：ADAM 見積りのデータストレージサイズ

ADAM統計表を作成するデータストレージのサイズをシリンダ単位で指定します。ADAMの実行ごとに、最大4つのデータストレージサイズが計算されます。最小値および最大値は、増分なしで指定できます。4つのすべての値に基づいたレポートを作成するために、ADAMERで2つの増分が計算されます。

## 例

次のように指定すると、データストレージサイズが100、125、150、および175シリンダの統計表が作成されます。

```
DATASIZE=100,175,25
```

DATASIZEパラメータの指定を省略すると、次のような4つのデータストレージサイズの統計表がADAMERで作成されます。

<b>サイズ1：</b>	最初の100個の入力レコードが読み込まれ、そこに存在するADAMディスクリプタ値とMAXISNで指定された値に基づいて、データストレージサイズの要件が決まります。結果として得られるデータストレージサイズがデータストレージサイズ1として使用されます。
<b>サイズ2：</b>	データストレージサイズ1の1.33倍。
<b>サイズ3：</b>	データストレージサイズ2の1.33倍。
<b>サイズ4：</b>	データストレージサイズ3の1.33倍。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド34（ダンプあり）またはユーザーアベンド35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベン  
ドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ  
ラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了しま  
す。

### NUMREC：読み込みレコードの最大数

入力ファイルから読み込まれる最大レコード数です。NUMRECパラメータの指定がない場  
合は、すべてのレコードが読み込まれます。

## 例

---

### 例 1

```
ADAMER ADAMDE=CC,  
ADAMER DATADEV=3350,DATASIZE=50,110,20,  
ADAMER DATAPFAC=10,MAXISN=225000,BITRANGE=2,6,1
```

ADAM ディスクリプタは CC です。モデル 3350 デバイスタイプをデータストレージとして使  
用します。データストレージサイズが 50、70、90、および 110 シリンダの統計表を作成します。  
データストレージパディングファクタには 10 %を使用します。ファイルの予定レコード数は  
225,000 です。データストレージサイズごとに、ビット切り捨て 2、3、4、5、および 6 ビットに  
ついて統計表を作成します。

### 例 2

```
ADAMER ADAMDE=CD,DATADEV=3380,DATAPFAC=5,MAXISN=80000
```

ADAM ディスクリプタは CD です。モデル 3380 デバイスタイプをデータストレージとして使  
用します。データストレージパディングファクタには 5 %を使用します。ファイルの予定レコー  
ド数は 80,000 です。他のパラメータにはすべてデフォルト値を使用します。

# 117

## ADAMER 出力レポートの内容

ADAMER で作成されるレポートには、次の項目が含まれます。

フィールド	説明
LOADISNS	入力データセットに含まれるレコード数。
MAXISN	ファイルのレコード総数 (MAXISN パラメータの説明を参照)。
DATA DEVICE	データストレージデバイスタイプ (DATADEV パラメータの説明を参照)。
DATAPFAC	データストレージパディングファクタ (DATAPFAC パラメータの説明を参照)。

次の項目は、"AVERAGE NUMBER OF EXCP" の下に表示されます。

フィールド	説明
Data Storage SIZE	DATASIZE パラメータの説明を参照してください。シリンダ数は最も近い整数値に切り上げられます。
BIT-PARM	BITRANGE パラメータの説明を参照してください。
FOR LOADISNS	ADAM ディスクリプタの使用時に、レコードの検索および読み込みに必要な平均 I/O 回数。ファイル内のレコード数が入力データセットに含まれるレコード数に等しいということを前提にして結果が得られます。
DISK USAGE	ファイルの最初のロード後に占められるデータストレージスペースの比率。ロードされるレコード数が入力データセットに含まれるレコード数に等しいということを前提にして結果が得られます。
FOR MAXISN	ADAM ディスクリプタの使用時に、レコードの検索および読み込みに必要な平均 I/O 回数。ファイル内のレコード数が MAXISN パラメータで指定された値に等しいということを前提にして結果が得られます。
DISK USAGE	ファイルの最初のロード後に占められるデータストレージスペースの比率。ロードされるレコード数が MAXISN パラメータで指定された値に等しいということを前提にして結果が得られます。

ADAMER レポートの情報を使用して、次のことを判断できます。

- アクセス回数とデータストレージスペース要件との間の最適なバランス。
- 開始値が似ているレコードが同じ物理ブロックにロードされるようにするため、各 ADAM ディスクリプタ値から切り捨てる最適なビット数。これは、順次読み込み処理を最適化する場合にのみ必要です。

# 118

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	586
▪ z/OS .....	587
▪ z/VM .....	588
▪ VSE .....	742

ここでは、BS2000、z/OS、z/VM、およびVSEシステムにおけるADAMERの実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
入力データ	DDEBAND	テープ/ディスク	ADACMPまたはADAULDユーティリティの出力
ADARUNパラメータ	DDCARD		オペレーションマニュアル
ADAMERパラメータ	SYSDTA/DDKARTE		
ADARUNメッセージ	SYSOUT/DDPRINT		メッセージおよびコード
ADAMERメッセージ/レポート	SYSLST/DDDRUCK		メッセージおよびコード

### ADAMER JCL の例 (BS2000)

SDF フォーマットの場合：

```

/.ADALOD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A M E R ALL FUNCTIONS
/REMARK *
/ASS-SYSLST L.MER
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDEBAND,CMP.AUS
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAMER,DB=yyyyy,IDTNAME=ADABAS5B
ADAMER ADAMDE=AA,DATASIZE=5200,BITRANGE=8,10,1
ADAMER MAXISN=10000
/LOGOFF SYS-OUTPUT=DEL
    
```

ISP フォーマットの場合：

```

/.ADAMER LOGON
/OPTION MSG=FM,DUMP=YES
/REMARK *
/REMARK * A D A M E R ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.MER
/FILE ADA.MOD,LINK=DDLIB
/FILE CMP.AUS,LINK=DDEBAND
/EXEC (ADARUN,ADA.MOD)

ADARUN PROG=ADAMER,DB=yyyyy,IDTNAME=ADABAS5B
ADAMER ADAMDE=AA,DATASIZE=5200,BITRANGE=8,10,1
ADAMER MAXISN=10000
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD名	ストレージ	追加情報
入力データ	DDEBAND	テープ/ディスク	ADACMPまたはADAULDユーティリティの出力
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAMER パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAMER メッセージ/レポート	DDDRUCK	プリンタ	メッセージおよびコード

### ADAMER JCL の例 (z/OS)

この例については、MVSJOBS データセットの ADAMER を参照してください。

```

//ADAMER JOB
//*
//* ADAMER:
//* ADAM ESTIMATION
//*
//MER EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDEBAND DD DISP=OLD,DSN=EXAMPLE.DByyyyy.COMPR1 <=== COMPRESS
DATA
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *

```

## JCL/JCS の必要項目と指定例

```
ADARUN  PROG=ADAMER,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE  DD  *
ADAMER  MAXISN=1000,ADAMDE=AA,BITRANGE=0,2,4
ADAMER  DATADEV=eeee,DATAPFAC=10,DATASIZE=100,175,25
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
入力データ	DDEBAND	テープ/ディスク	ADACMP または ADAULD ユーティリティの出力
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAMER パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAMER メッセージ/レポート	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

## ADAMER JCL の例 (z/VM)

```
DATADEF  DDEBAND,DSN=ADABASV.V.BAND,MODE=A
DATADEF  DDPRINT,DSN=ADAMER.DDPRINT,MODE=A
DATADEF  DUMP,DUMMY
DATADEF  DDDRUCK,DSN=ADAMER.DDDRUCK,MODE=A
DATADEF  DDCARD,DSN=RUNMER.CONTROL,MODE=A
DATADEF  DDKARTE,DSN=ADAMER.CONTROL,MODE=A
ADARUN
```

## RUNMER CONTROL A1 の内容

```
ADARUN  PROG=ADAMER,DEVICE=dddd,DB=yyyyy
```

## ADAMER CONTROL A1 の内容

```
ADAMER ADAMDE=AA,DATASIZE=5200,BITRANGE=8,10,1 ADAMERMAXISN=10000
```

## VSE

ファイル	シンボル名前	ストレージ	論理ユニット	追加情報
入力データ	EBAND	テープ ディスク	SYS010 *	ADACMP または ADAULD ユーティリティの出力
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	オペレーションマニュアル
ADAMER パラメータ	-	リーダー	SYSIPT	
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADAMER メッセージ/レポート	-	プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADAMER JCS の例 (VSE)

VSE プロシージャ (PROC) の内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。この例については、メンバADAMER.Xを参照してください。

```
* $$ JOB JNM=ADAMER,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAMER
// OPTION LOG,PARTDUMP
*      ADAM ESTIMATION
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// DLBL EBAND,'EXAMPLE.DByyyyy.COMPR1',0,SD
// EXTENT SYS004
// ASSGN SYS004,DISK,VOL=DISK01,SHR
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAMER,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAMER MAXISN=1000,ADAMDE=AA,BITRANGE=0,2,4
ADAMER DATADEV=eeee,DATAPFAC=10,DATASIZE=100,175,25
/*
/&
* $$ EOJ
```



# 119

## ADAORD ユーティリティ：リオーダ機能

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- REORASSO：アソシエータのリオーダ
- REORDATA：データストレージのリオーダ
- REORDB：データベースのリオーダ
- REORFASSO：1つのファイルに対するアソシエータのリオーダ
- REORFDATA：1つのファイルに対するデータストレージのリオーダ
- REORFILE：ファイルのリオーダ
- RESTRUCTUREDB：データベースの再構築
- RESTRUCTUREF：1つのファイルの再構築
- STORE：ファイルの格納
- JCL/JCS の必要項目と指定例



# 120 機能概要

---

■ リオーダ機能 .....	594
■ 再構築機能 .....	595
■ 格納機能 .....	595
■ スペース割り当て .....	596
■ Adabas 8 の考慮事項 .....	596

ADAORD ユーティリティは次のことに使用できます。

- データベースまたはデータベース内の1つのファイルに対して、アソシエータまたはデータストレージをリオーダします (REORASSO、REORFASSO、REORDATA、REORFDATA、REORDB、および REORFILE 機能)。
- データベースまたはデータベース内の1つのファイルを再構築し、結果として得られる出力ファイルを既存のデータベースに格納します (RESTRUCTUREDDB、RESTRUCTUREF、および STORE 機能)。

ADAORD の1回の実行中には、1つの機能のみを実行できます。

ADAORDの実行中に、データベースの一部が上書きされます。したがって、ADAORD 機能を実行する前にデータベース (またはファイル) を ADASAV ユーティリティでバックアップしておいてください。

また、RESTRUCTUREF (ファイル) 以外のすべての ADAORD 機能では、操作に関連するデータベースファイルに対して EXF 排他制御を必要とします。RESTRUCTUREF では EXU 制御が必要であり、他ユーザーに対して対象となっているデータベースファイルの読み込みだけのアクセスを認めます。ただし、チェックポイントファイル、セキュリティファイル、または ADALOD の SYSFILE オプションを使用してロードされたファイルが関係する操作では、データベースに対して排他制御を必要とすることに注意してください。

 **Note:** アソシエータエクステンツの開始RABNを指定するときは、FCB、FDT および DSST に必要なスペースも考慮する必要があります。

このchapterでは、次のトピックについて説明します。

## リオーダ機能

---

REORASSO 機能は、すべてのファイルに対してすべてのアソシエータブロックを物理的にリオーダします。REORFASSO 機能は、1つのファイルに対してアソシエータをリオーダします。この機能は、アソシエータスペース内のフラグメントを削除し、複数のアドレスコンバータ、ノーマルインデックスとアップパーインデックス、およびデータストレージスペーステーブル (DSST) コンポーネントのエクステンツを、コンポーネントごとに1つの論理エクステンツに結合します。

REORDATA 機能は、データベース内のすべてのファイルに対してデータストレージをリオーダします。REORFDATA 機能は、1つのファイルに対してデータストレージをリオーダします。この機能は、空ブロックしか持たないエクステンツを圧縮し、またファイル削除によって発生するデータストレージ内のフラグメントを削除します。

REORDB 機能は、ADAORD の1回の実行で REORASSO と REORDATA 機能の両方を行います。

REORFILE 機能は、ADAORD の 1 回の実行で REORFASSO と REORFDATA 機能の両方を行います。レコードはディスクリプタ、ISN による論理的な順番、またはレコードが現在格納されている順にリオーダできます。

REORDATA、REORDB、REORFDATA、および STORE 機能は、ADAM ファイルをリオーダしません。ただし、これらの機能を使用して ADAM ファイルを異なる RABN に再配置することはできます。

## 再構築機能

RESTRUCTUREDB 機能は、データベース全体を 1 つのシーケンシャルデータセットにアンロードします。RESTRUCTUREF 機能は、1 つ以上のファイルを 1 つのシーケンシャルデータセットにアンロードします。アンロードされたデータが含まれるこのデータセットは、STORE 機能の入力として使用できます。

RESTRUCTURE 機能は、異なる物理デバイスにデータベースを再配置したり、1 つ以上のファイルを別のデバイスに再配置するために使用します。

RESTRUCTURE 機能で作成されたシーケンシャルデータセットのフォーマットは、データベースのデバイスタイプとは無関係であり、ADALOD や ADASAV ユーティリティに必要なフォーマットと互換性はありません。したがって、ターゲットデータベースは、ソースデータベースと異なるデバイスタイプにあってもかまいません。

アソシエータおよびデータストレージは RESTRUCTURE/STORE 処理の一部としてリオーダされます。

RESTRUCTUREDB/F 機能でオーバーフローエリアを使用する ADAM ファイルが再構築された後、STORE によって DATA ブロックサイズの小さいデータベースにそのファイルが格納されると、ADAORDERROR-103 が発生することがあります。この場合、ADAULD および ADALOD ユーティリティを使用して ADAM ファイルを移動します。

## 格納機能

STORE 機能は、RESTRUCTUREDB、RESTRUCTUREF、または REORDB 機能で作成された DDFILEA 出力を使用して、1 つ以上のファイルを既存のデータベースにロードします。

アソシエータおよびデータストレージは RESTRUCTURE/STORE 処理の一部としてリオーダされます。

STORE 機能は、ADAM ファイルをリオーダしません。ただし、他の ADAORD 機能と組み合わせて、ADAM ファイルを異なる RABN に再配置することはできます。RESTRUCTUREDB または RESTRUCTUREF 機能でオーバーフローエリアを使用する ADAM ファイルが再構築された後、STORE によって DATA ブロックサイズの小さいデータベースにそのファイルが格納され

ると、ADAORDER-103が発生することがあります。この場合、ADAULD および ADALOD ユーティリティを使用して ADAM ファイルを移動します。

## スペース割り当て

---

ADAORD は、`xxSIZE` または `MAXISN` および `MAXISN2` パラメータ（指定された場合）に必要なスペースの量を割り当てます。指定がない場合は、ファイルの現在のサイズに基づいてスペースが割り当てられます。`xxRELEASE` パラメータは、必要なスペースの量に影響を与えることに注意してください。

可能であれば、`xxxxVOLUME` パラメータで指定されたボリュームでスペースが割り当てられます。解放されたスペースが指定ボリュームで不足していると、ADAORD はデフォルトの割り当てルールに従って、他ボリュームの残りの必要スペースを割り当てます。

`xxRABN` パラメータは、関連する `xxxxVOLUME` パラメータより優先されます。

## Adabas 8 の考慮事項

---

ADAORD STORE を使用すると、Adabas 8 以前のバージョンのデータベースやファイルを再構築して、Adabas 8 データベースに格納できます。ただし、再構築した Adabas 8 データベース出力、または Adabas 7 などの以前のバージョンで実行中のデータベースに含まれるファイルは、格納できません。

# 121

## REORASSO : アソシエータのリオーダ

---

■ オプションのパラメータとサブパラメータ .....	763
■ 例 .....	603

REORASSO機能は、アソシエータ全体をリオーダします。ファイルが明示的に指定されていないと、既存の定義に従って関連するアソシエータ情報がリオーダされます。特定のファイルのアソシエータ情報をリオーダするには、REORASSO機能を使用します。

この機能は、オペレーション時に使用されるデータベースファイルに対してEXF 排他制御を必要とします。また、ADAORDの実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能に指定されたファイルがもともと ISNREUSE=YES がアクティブな状態でロードされていた場合、このリオーダ機能は、そのファイルのファイルコントロールブロック（FCB）内の最初の未使用 ISN 値をアドレスコンバータ内で最初に検出された実際の未使用 ISN にリセットします。

次に、ADAORD REORASSO 機能の構文を示します。

```

ADAORD REORASSO [DBINDEXCOMPRESSION = { YES | NO } ]
                [FILE = file-number ]
                [ACRABN = starting-rabn ]
                [AC2RABN = starting-rabn ]
                [ALLOCATION = { FORCE | NOFORCE } ]
                [ASSOPFAC = padding-factor ]
                [ASSOVOLUME = 'Associator-extent-volume' ]
                [INDEXCOMPRESSION = { YES | NO } ]
                [ISNSIZE = { 3 | 4 } ]
                [MAXISN = highest-isn ]
                [MAXISN2 = highest-isn ]
                [NIRABN = starting-rabn ]
                [NIRELEASE]
                [NISIZE = size ]
                [UIRABN = starting-rabn ]
                [UIRELEASE]
                [UISIZE = size ]
                [LPB = { prefetch-buffer-size | ADARUN-lu } ]
                [MAXFILES = maximum-number-files ]
                [NEWDBID = database-identifier ]
                [NEWDBNAME = database-name]
                [NOUSERABEND]
                [RAID]
                [TEST]

```

 **Note:** MAXFILES または NEWDBID パラメータを指定すると、アクティブなニュークリアスは REORASSO 機能の完了後に自動的に終了します。

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

### ACRABN：アドレスコンバータの開始 RABN

ファイルのアドレスコンバータエクステントの開始 RABN です。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。

アソシエータエクステントの開始 RABN を指定するときは、FCB、FDT および DSST に必要なスペースも考慮する必要があります。

### AC2RABN：セカンダリアドレスコンバータの開始 RABN

ファイルのセカンダリアドレスコンバータエクステントの開始 RABN です。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

アソシエータエクステントの開始 RABN を指定するときは、FCB、FDT および DSST に必要なスペースも考慮する必要があります。

### ALLOCATION：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRABN、NIRABN または UIRABN に従ってファイルエクステント割り当てが得られない場合に取るべき動作を指定します。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### ASSOPFAC：アソシエータパディングファクタ

新しいアソシエータブロックパディングファクタです。指定値は、リオーダ処理時に使用しない各アソシエータブロックのパーセントを表します。1 から 90 の値が指定できます。パディング後に残っているバイト数は、最大ディスクリプタ値に 10 を加えたものより大きくなければなりません。

このパラメータ指定がなければ、現在ファイルに有効なアソシエータパディングファクタが使用されます。

**ASSOVOLUME**：アソシエータエクステントボリューム

 **Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース (AC、NI、UI エクステント) を割り当てるボリュームを指定します。指定ボリュームに必要なブロック数が存在しないと、ADAORD はデフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

ACRABN、UIRABN、または NIRABN が指定されていると、ADAORD は、該当するエクステントタイプを割り当てるときに、ASSOVOLUME を無視します。

ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADAORD のデフォルト割り当てルールに従って割り当てられます。

**DBINDEXCOMPRESSION**：データベースインデックスの圧縮

DBINDEXCOMPRESSION は、ファイルのインデックスが圧縮形式または非圧縮形式で再構築されるかどうかを示します。INDEXCOMPRESSION パラメータが未指定の全ファイルに適用されます。

DBINDEXCOMPRESSION は、圧縮インデックスまたは非圧縮インデックスの再構成をデータベース内の全ファイルに対して使用できるので、ファイルごとにインデックス圧縮を指定する必要がなくなります。

**FILE**：ファイル番号

後続のパラメータセットが適用されるファイル番号です。指定された各ファイルおよびそのパラメータは、ADAORD REORASSO 機能ステートメントに続く別個の ADAORD ステートメントでそれぞれ指定する必要があります。

番号が指定されないファイルについては、現在のアソシエータブロックパディングファクタと MAXISN 値が保持され、アソシエータのスペース割り当てはすべて同じ状態で残ります。

**INDEXCOMPRESSION**：ファイルインデックスの圧縮

INDEXCOMPRESSION は、ファイルが圧縮形式または非圧縮形式で再構成されるかどうかを示します。通常、インデックスを圧縮すると、インデックススペースが少なくなり、Adabas ニュークリアスのインデックス操作の効率が向上します。

INDEXCOMPRESSION を指定せず、

- DBINDEXCOMPRESSION パラメータがデータベース全体に対して指定された場合は、データベース値がデフォルトになります。
- DBINDEXCOMPRESSION が指定されない場合は、ファイルの現在の圧縮形式がデフォルトになります。

**ISNSIZE**：3 または 4 バイト ISN

ISNSIZE はファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック (FCB) 内に格納されます。

 **Note:** 物理的にカップリングされているファイルの ISNSIZE を ADAORD で変更することはできません。

**LPB：プリフェッチバッファサイズ**

内部プリフェッチバッファのサイズをバイト単位で指定します。指定可能な最大値は、32,760 バイトです。デフォルトは、ADARUN LU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなる場合があります。

**MAXFILES：最大ファイル数**

MAXFILES パラメータは、データベースの中にロードできる最大ファイル数を指定します。このパラメータの最小値は 3 です。許可された最大値は 5000 または ASSOR1 ブロックサイズより 1 小さい値のうち、どちらか小さい方の値です。例えば、2003 は、ASSOR1 が 3380 DASD 上に格納されたデータベースの最大 MAXFILES 値です。

このパラメータを省略すると、現在の MAXFILES 値が保持されます。

MAXFILES を指定すると、ニュークリアスは ADAORD REORASSO 機能の完了後に終了します。

**MAXISN：ファイルに対する可能最大 ISN**

MAXISN は、ファイルに対して割り当て可能な最大 ISN を指定します。この値は、ADAREP データベースレポートに出力される現在の TOPISN より大きい必要があります。

指定された値は、ADAORD によりアドレスコンバータスペース必要量の計算に使用されます。このパラメータが省略されると、ファイルに対する現在の MAXISN 値が保持されます。

**MAXISN2：ファイルに対する可能最大セカンダリ ISN**

MAXISN は、ISN 内のセカンダリアドレスコンバータ (AC2) のサイズを指定します。この値は、ADAREP データベースレポートに出力される現在の TOP AC2 ISN より大きい必要があります。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

ADAORD は、ファイルのセカンダリアドレスコンバータで必要なスペースを計算するために指定された値を使用します。このパラメータが省略されると、ファイルに対する現在の MAXISN2 値が保持されます。ファイルにセカンダリアドレスコンバータエクステンが含まれていない場合、このパラメータは無視されます。

**NEWDBID：データベース ID**

NEWDBID はデータベースに割り当てる ID です。1~65535 の範囲の値を指定できます。Online System Security を使用するシステムの場合、値 999 は予約済みです。このパラメータを省略すると、現在のデータベース ID が保持されます。

NEWDBID を指定すると、ニュークリアスは ADAORD REORASSO 機能の完了後に終了します。

**NEWDBNAME：データベース名**

データベースに割り当てる名前です。1 から 16 文字までで名前を割り当てるができます。このパラメータ指定がなければ、現在のデータベース名が保持されます。

データベース名に特殊文字または埋め込み空白が含まれている場合は、名前を単一引用符で囲う ('...') 必要があります。単一引用符自体が名前に含まれる場合は、'JAN'SDB' のように 2 回連続して指定する必要があります。

**NIRABN**：ノーマルインデックスの開始 RABN

NIRABN は、ノーマルインデックスエクステントの開始 RABN 番号を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**NIRELEASE**：未使用ノーマルインデックスブロックの解放

NIRELEASE が指定されると、指定ファイルに属する未使用のノーマルインデックス (NI) ブロックが解放されます。NIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた NI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**NISIZE**：ノーマルインデックスサイズ

NISIZE は、ノーマルインデックスに対して割り当てるブロック数またはシリンダ数を指定します。値をブロック単位で指定する場合、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**RAID**：ファイルが存在しないと判断した場合の動作

RAID は、ADAORD に、データベースに存在しないファイルを参照するあらゆる FILE パラメータを無視するように指示します。

RAID が指定されない場合 (デフォルト)、ADAORD は、データベースに存在しないファイルを参照する FILE パラメータを検出すると、エラーメッセージを出力して終了します。

RAID パラメータは、Adabas Recovery Aid (ADARAI) で構築されたリカバリジョブで使用する場合に指定します。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定パラメータの構文だけがテストされ、値や変数の正当性はチェックされません。

**UIRABN** : アッパーインデックスの開始 RABN

UIRABN は、ファイルのアッパーインデックスエクステントの開始 RABN 番号を指定します。このパラメータを省略すると、ADAORD によって各エクステントの開始 RABN が割り当てられます。

**UIRELEASE** : 未使用アッパーインデックスブロックの解放

UIRELEASE が指定されると、指定ファイルに属する未使用のアッパーインデックス (UI) ブロックが解放されます。UIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた UI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**UISIZE** : アッパーインデックスサイズ

UISIZE は、アッパーインデックスに対して割り当てるブロック数またはシリンダ数を指定します。値をブロック単位で指定する場合、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

## 例

---

### 例 1

```
ADAORD REORASSO
```

アソシエータをリオーダします。

### 例 2

```
ADAORD REORASSO
ADAORD MAXFILES=200
ADAORD NEWDBID=6,NEWDBNAME=DATABASE-6
```

アソシエータをリオーダします。データベースは最大 200 ファイルまで許容できます。新しいデータベース ID および名前は各々 6 および DATABASE-6 です。

### 例 3

```
ADAORD REORASSO
ADAORD FILE=1,ACRABN=1000,NIRABN=2200,
ADAORD FILE=2,MAXISN=500000,
ADAORD FILE=4,ASSOPFAC=5
```

アソシエータをリオーダします。ファイル 1 のアドレスコンバータの割り当ては RABN 1,000 から開始します。ファイル 1 のノーマルインデックスは RABN 2,200 から開始します。ファイ

ル2の MAXISN は 500,000 にセットします。ファイル4のアソシエータブロックパディングファクタは5%にセットします。各ファイルの現在の定義に従って、他のすべてのデータベースファイルのアソシエータ情報をリオーダします。

# 122

## REORDATA：データストレージのリオーダ

---

■ オプションのパラメータとサブパラメータ .....	606
■ 例 .....	609

REORDATA 機能は、すべてのファイルに対してデータストレージをリオーダします。指定のないファイルについては、既存の定義に従ってリオーダします。REORDATA 機能は、ADAM ファイルをリオーダしません。しかし、これは、異なる RABN に ADAM ファイルの再配置に使用できます。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORD の実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

次に、ADAORD REORDATA 機能の構文を示します。

```
ADAORD REORDATA [FILE = file-number ]
                 [ALLOCATION = { FORCE | NOFORCE } ]
                 [DATAPFAC = padding-factor ]
                 [DATAVOLUME = 'Data-Storage-extent-volume' ]
                 [DSDEV = device-type ]
                 [DSRABN = starting-rabn ]
                 [DSRELEASE]
                 [DSSIZE =size ]
                 [MAXRECL = record-length ]
                 [SORTSEQ = { descriptor | ISN | physical-sequence } ]
                 [LIP = { isn-pool-size | 16384 } ]
                 [LPB = { prefetch-buffer-size | ADARUN-lu } ]
                 [NOUSERABEND]
                 [RAID]
                 [TEST]
```

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

---

**ALLOCATION**：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ DSRABN に従ってファイルエクステント割り当てが得られない場合に取りべき動作を指定します。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、割り当てができないと、配置パラメータに関係なく再度割り当てようとします。

**DATAPFAC**：データストレージパディングファクタ

DATAPFAC は、新データストレージパディングファクタを指定します。指定値は、リオーダ処理時レコード拡張のために確保される各データストレージブロックのパーセントを表します。1 から 90 までの範囲の値が指定できます（パディングファクタの設定の詳細については ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。このパラメータ指定がなければ、ファイルに対して現在有効なパディングファクタが使用されます。

**DATAVOLUME**：データストレージエクステントボリューム

 **Note:** DATAVOLUME パラメータは単一引用符でくくられなければなりません。

ファイルのデータストレージスペースエクステントを割り当てるときに使用するボリューム番号を指定します。DSSIZE で要求されたブロックの数が指定ボリュームに存在しないと、ADAORD はデフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。

DATAVOLUME を指定しないと、ファイルの現在のサイズに基づいてデータストレージスペースが割り当てられます。DSRELEASE パラメータは、必要なスペースの量にも影響を与えます。

**DSDEV**：データストレージデバイスタイプ

DSDEV は、そのファイルのデータストレージのデバイスタイプです。指定デバイスタイプは、通常、そのデータベースの作成時か、あるいは ADADBS ユーティリティの ADD 機能によって、すでに Adabas に対して定義されているものでなければなりません。

DSDEV が指定されない場合、ADAORD はリオーダ前に使用されていたデバイスタイプでファイルを割り当てようとしています。

**DSRABN**：データストレージ開始 RABN

指定されたファイルのデータストレージエクステントの開始 RABN です。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**DSRELEASE**：未使用データストレージブロックの解放

DSRELEASE を指定することで、指定ファイルに属する未使用のデータストレージ (DS) ブロックが解放されます。DSRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた DS ブロック数が ADAORD によって割り当てられます。

 **Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**DSSIZE**：データストレージサイズ

DSSIZE は、ファイルのデータストレージ (DS) の論理エクステントに対して割り当てられるブロック数またはシリンダ数を指定します。値をブロック単位で指定する場合、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは DATAPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**FILE：**ファイル番号

FILE は、後続のパラメータセットが適用されるファイルを指定します。指定された各ファイルおよびそのパラメータは、ADAORDREORDATA ステートメントに続く別個の ADAORD ステートメントでそれぞれ指定する必要があります。

番号が指定されないファイルについては、現在の物理順でリオーダされ、現在のデータストレージパディングファクタおよびスペース割り当ては保持されます。

**LIP：ISN バッファプールサイズ**

LIP パラメータは、アドレスコンバータの再作成時のアソシエータの入出力処理の回数を減少させるために使用できます。処理対象となる中で一番規模が大きいファイルの全 ISN を受け入れるだけの大きさを指定したときに最適のパフォーマンスとなります。

ISN と割り当てられたデータストレージ RABN を入れるための ISN プールサイズを指定します。値はバイト単位で指定し、数値のみ ("2048" など) を指定するか、あるいは数値の後に "K" を付けて ("2K" など) キロバイト単位で指定します。LIP のデフォルトは 16384 バイト (または 16K) です。

1 件の入力レコード長は、ISNSIZE + RABNSIZE です。したがって、エントリ長は、最小 6 バイト (ファイルの ISNSIZE が 3 でデータベースの RABNSIZE が 3 の場合)、最大 8 バイト (ISNSIZE が 4 で RABNSIZE が 4 の場合) になります。

**LPB：**プリフェッチバッファサイズ

LPB は、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能な最大値は、32,760 バイトです。デフォルトは、ADARUN LU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなる場合があります。

**MAXRECL：**最大圧縮レコード長

ファイルの最大圧縮レコード長の制限を変更するには、MAXRECL パラメータを使用します。MAXRECL を指定すると、次の 2 つの効果があります。

ファイルの DATA データセットは、指定された長さをサポートするデバイスに対してのみ割り当てられます。

指定された長さを超えるデータストレージレコードがファイル内に含まれている場合、ADAORD は異常終了し、ERROR-126 メッセージ (データストレージレコードが長すぎる) を出力します。

MAXRECL を指定しなければ、最大圧縮レコード長は変更されません。

**NOUSERABEND：**アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エ

ラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

#### RAID：ファイルが存在しないと判断した場合の動作

RAID は、ADAORD に、データベースに存在しないファイルを参照するあらゆる FILE パラメータを無視するように指示します。

RAID が指定されない場合（デフォルト）、ADAORD は、データベースに存在しないファイルを参照する FILE パラメータを検出すると、エラーメッセージを出力して終了します。

RAID パラメータは、Adabas Recovery Aid (ADARAI) で構築されたリカバリジョブで使用する場合に指定します。

#### SORTSEQ：レコードの処理順序

SORTSEQ は、ファイルを処理する順番を指定します。このパラメータを省略すると、レコードが物理順で処理されます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。空値省略されたディスクリプタフィールド、ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。



**Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない（値を持たない）場合は、空値省略されたディスクリプタフィールドでなくても、そのレコードはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

ISN を指定すると、ファイルが ISN 昇順に処理されます。また、Adabas チェックポイントまたはセキュリティファイルについては、SORTSEQ=ISN しか許可されません。

#### TEST：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

### 例 1

```
ADAORD REORDATA
```

データベース全体のデータストレージをリオーダします。

### 例 2

```
ADAORD REORDATA
```

```
ADAORD FILE=1,DSRABN=1000,DSSIZE=200B
```

## REORDATA : データストレージのリオーダ

---

ADAORD	FILE=4 ,SORTSEQ=AA
ADAORD	FILE=5 ,DATAPFAC=15

データストレージをリオーダします。ファイル1のデータストレージはRABN 1,000から開始し、200ブロックを割り当てます。ディスクリプタAAを順序制御に使用して、ファイル4をリオーダします。ファイル5のデータストレージパディングファクタは15%にセットします。既存の定義に従って、他のすべてのデータベースをリオーダします。

# 123

## REORDB：データベースのリオーダ

---

■ オプションのパラメータとサブパラメータ .....	614
■ 例 .....	620

REORDB 機能は、データベースに対するすべてのアソシエータおよびデータストレージをリオーダします。指定のないファイルについては、既存の定義に従ってリオーダします。

REORDB 機能は、ADAM ファイルをリオーダしません。しかし、これは、異なる RABN に ADAM ファイルの再配置に使用できます。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORD の実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能に指定されたファイルがもともと ISNREUSE=YES がアクティブな状態でロードされていた場合、このリオーダ機能は、そのファイルのファイルコントロールブロック (FCB) 内の最初の未使用 ISN 値をアドレスコンバータ内で最初に検出された実際の未使用 ISN にリセットします。

次に、ADAORD REORDB 機能の構文を示します。

```

ADAORD REORDB  [DBINDEXCOMPRESSION = { YES | NO } ]
                  [FILE = file-number ]
                  [ACRABN = starting-rabn ]
                  [AC2RABN = starting-rabn ]
                  [ALLOCATION = { FORCE | NOFORCE } ]
                  [ASSOPFAC = padding-factor ]
                  [ASSOVOLUME = 'Associator-extent-volume' ]
                  [DATAPFAC = padding-factor ]
                  [DATAVOLUME = 'Data-Storage-extent-volume' ]
                  [DSDEV = device-type ]
                  [DSRABN = starting-rabn ]
                  [DSRELEASE]
                  [DSSIZE = size ]
                  [INDEXCOMPRESSION = { YES | NO } ]
                  [ISNSIZE = { 3 | 4 } ]
                  [MAXISN = highest-isn ]
                  [MAXISN2 = highest-isn ]
                  [MAXRECL = record-length ]
                  [NIRABN = starting-rabn ]
                  [NIRELEASE]
                  [NISIZE = size ]
                  [SORTSEQ = { descriptor | ISN | physical-sequence } ]
                  [UIRABN = starting-rabn ]
                  [UIRELEASE]
                  [UIsize = size ]
                  [LIP = { isn-pool-size | 16384 } ]
                  [LPB = { prefetch-buffer-size | ADARUN-lu } ]
                  [MAXFILES = maximum-number-files ]
                  [NEWDBID = database-identifier ]
                  [NEWDBNAME = database-name ]
                  [NOUSERABEND]
                  [RAID]
                  [TEST]

```

 **Note:** MAXFILES または NEWDBID パラメータを指定すると、アクティブなニュークリアスは REORDB 機能の完了後に自動的に終了します。

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

### ACRABN：アドレスコンバータの開始 RABN

ファイルのアドレスコンバータエクステンツの開始 RABN です。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。要求されたスペースは、1つのエクステンツで使用できる必要があります。

アソシエータエクステンツの開始 RABN を指定するときは、FCB、FDT および DSST に必要なスペースも考慮する必要があります。

### AC2RABN：セカンダリアドレスコンバータの開始 RABN

ファイルのセカンダリアドレスコンバータエクステンツの開始 RABN です。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。要求されたスペースは、1つのエクステンツで使用できる必要があります。ファイルにセカンダリアドレスコンバータエクステンツが含まれていない場合、このパラメータは無視されます。

### ALLOCATION：ファイルエクステンツ割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステンツ割り当てが得られない場合取るべき動作を指定します。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステンツ割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### ASSOPFAC：アソシエータパディングファクタ

新しいアソシエータブロックパディングファクタです。指定値は、リオーダ処理時に使用しない各アソシエータブロックのパーセントを表します。1 から 90 の値が指定できます。パディング後に残っているバイト数は、最大ディスクリプタ値に 10 を加えたものより大きくなければなりません。

このパラメータ指定がなければ、現在ファイルに有効なアソシエータパディングファクタが使用されます。

### ASSOVOLUME：アソシエータエクステンツボリューム



**Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UI エクステンツ）を割り当てるボリュームを指定します。要求スペースが指定ボリュームに存在しないと、ADAORD は、デフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

ACRABN、UIRABN、または NIRABN が指定されていると、ADAORD は、該当するエクステントタイプを割り当てるときに、ASSOVOLUME を無視します。

ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADAORD のデフォルト割り当てルールに従って割り当てられます。

#### **DATAPFAC**：データストレージパディングファクタ

DATAPFAC は、新データストレージパディングファクタを指定します。指定値は、リオーダ処理時レコード拡張のために確保される各データストレージブロックのパーセントを表します。1 から 90 までの範囲の値が指定できます（パディングファクタの設定の詳細については ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。このパラメータ指定がなければ、ファイルに対して現在有効なパディングファクタが使用されます。

#### **DATAVOLUME**：データストレージエクステントボリューム

 **Note:** DATAVOLUME パラメータは単一引用符でくくらなければなりません。

ファイルのデータストレージスペースエクステントを割り当てるときに使用するボリューム番号を指定します。DSSIZE で要求されたブロック数が指定ボリュームに存在しないと、ADAORD はデフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。

DATAVOLUME が指定されない場合、データストレージスペースは、ADAORD のデフォルト割り当てルールに従って、割り当てられます。

#### **DBINDEXCOMPRESSION**：データベースインデックスの圧縮

DBINDEXCOMPRESSION は、ファイルのインデックスが圧縮形式または非圧縮形式で再構築されるかどうかを示します。INDEXCOMPRESSION パラメータが未指定の全ファイルに適用されます。

DBINDEXCOMPRESSION は、圧縮インデックスまたは非圧縮インデックスの再構成をデータベース内の全ファイルに対して使用できるので、ファイルごとにインデックス圧縮を指定する必要がなくなります。

#### **DSDEV**：データストレージデバイスタイプ

DSDEV は、そのファイルのデータストレージのデバイスタイプです。指定デバイスタイプは、通常、そのデータベースの作成時か、あるいは ADADBS ユーティリティの ADD 機能によって、すでに Adabas に対して定義されているものでなければなりません。

DSDEV が指定されない場合、ADAORD はリオーダ前に使用されていたデバイスタイプでファイルを割り当てようとしています。

#### **DSRABN**：データストレージ開始 RABN

ファイルのデータストレージエクステントの開始 RABN です。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**DSRELEASE：未使用データストレージブロックの解放**

DSRELEASEを指定することで、指定ファイルに属する未使用のデータストレージ（DS）ブロックが解放されます。DSRELEASEが指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた DS ブロック数が ADAORD によって割り当てられます。



**Note:** Adabasは、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**DSSIZE：データストレージサイズ**

DSSIZEは、ファイルのデータストレージ（DS）の論理エクステントに対して割り当てられるブロック数またはシリンダ数を指定します。値をブロック単位で指定する場合、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは DATAPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**FILE：ファイル番号**

後続のパラメータセットが適用されるファイルです。指定された各ファイルは、ADAORD REORDB 機能ステートメントに続く別個の ADAORD ステートメントでそれぞれ指定する必要があります、ファイルに適用されるパラメータをその直後に指定する必要があります。

番号が指定されないファイルについては、現在のアソシエータブロックパディングファクタと MAXISN 値が保持され、アソシエータおよびデータストレージのスペース割り当てはすべて同じ状態で残ります。

**INDEXCOMPRESSION：ファイルインデックスの圧縮**

INDEXCOMPRESSION は、ファイルが圧縮形式または非圧縮形式で再構成されるかどうかを示します。通常、インデックスを圧縮すると、インデックススペースが少なくなり、Adabas ニュークリアスのインデックス操作の効率が向上します。

INDEXCOMPRESSION を指定せず、

- DBINDEXCOMPRESSION パラメータがデータベース全体に対して指定された場合は、データベース値がデフォルトになります。
- DBINDEXCOMPRESSION が指定されない場合は、ファイルの現在の圧縮形式がデフォルトになります。

**ISNSIZE：3 または 4 バイト ISN**

ISNSIZE はファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック（FCB）内に格納されます。



**Note:** 物理的にカップリングされているファイルの ISNSIZE を ADAORD で変更することはできません。

**LIP：ISN バッファプールサイズ**

LIP パラメータは、アドレスコンバータの再作成時のアソシエータの入出力処理の回数を減少させるために使用できます。処理対象となる中で一番規模が大きいファイルの全 ISN を受け入れるだけの大きさを指定したときに最適のパフォーマンスとなります。

ISN と割り当てられたデータストレージ RABN を入れるための ISN プールサイズを指定します。値はバイト単位で指定し、数値のみ ("2048" など) を指定するか、あるいは数値の後に "K" を付けて ("2K" など) キロバイト単位で指定します。LIP のデフォルトは 16384 バイト (または 16K) です。

1 件の入力レコード長は、ISNSIZE + RABNSIZE です。したがって、エントリ長は、最小 6 バイト (ファイルの ISNSIZE が 3 でデータベースの RABNSIZE が 3 の場合)、最大 8 バイト (ISNSIZE が 4 で RABNSIZE が 4 の場合) になります。

**LPB：プリフェッチバッファサイズ**

LPB は、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能最大値は、32,760 バイトです。デフォルトは、ADARUNLU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなる場合があります。

**MAXFILES：最大ファイル数**

MAXFILES パラメータは、データベースの中にロードできる最大ファイル数を指定します。このパラメータの最小値は 3 です。許可された最大値は 5000 または ASSOR1 ブロックサイズより 1 小さい値のうち、どちらか小さい方の値です。例えば、2003 は、ASSOR1 が 3380 DASD 上に格納されたデータベースの最大 MAXFILES 値です。

このパラメータを省略すると、現在の MAXFILES 値が保持されます。

MAXFILES を指定すると、ニュークリアスは ADAORD REORDB 機能の完了後に終了します。

**MAXISN：ファイルに有効な最大 ISN**

ファイルに対して割り当て可能な最大 ISN です。この値は、ADAREP データベースレポートに出力される現在の TOPISN より大きい必要があります。

指定された値は、ADAORD によりアドレスコンバータスペース必要量の計算に使用されます。このパラメータが省略されると、ファイルに対する現在の MAXISN 値が保持されます。

**MAXISN2：ファイルに対する可能最大セカンダリ ISN**

MAXISN は、ISN 内のセカンダリアドレスコンバータ (AC2) のサイズを指定します。この値は、ADAREP データベースレポートに出力される現在の TOP AC2 ISN より大きい必要があります。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

ADAORD は、ファイルのセカンダリアドレスコンバータで必要なスペースを計算するために指定された値を使用します。このパラメータが省略されると、ファイルに対する現在の MAXISN2 値が保持されます。ファイルにセカンダリアドレスコンバータエクステンが含まれていない場合、このパラメータは無視されます。

**MAXRECL：最大圧縮レコード長**

ファイルの最大圧縮レコード長の制限を変更するには、MAXRECL パラメータを使用します。MAXRECL を指定すると、次の2つの効果があります。

- ファイルの DATA データセットは、指定された長さをサポートするデバイスに対してのみ割り当てられます。
- 指定された長さを超えるデータストレージレコードがファイル内に含まれている場合、ADAORD は異常終了し、ERROR-126 メッセージ（データストレージレコードが長すぎる）を出力します。

MAXRECL を指定しなければ、最大圧縮レコード長は変更されません。

**NEWDBID：データベース ID**

NEWDBID はデータベースに割り当てる ID です。1~65,535 の範囲の値を指定できます。Adabas Online System Security を使用するシステムの場合、値 999 は予約済みです。このパラメータを省略すると、現在のデータベース ID が保持されます。

NEWDBID を指定すると、ニュークリアスは ADAORD REORDB 機能の完了後に終了します。

**NEWDBNAME：データベース名**

NEWDBNAME は、データベースに割り当てる名前を指定します。16 文字までで名前を割り当てることができます。名前に特殊文字または埋め込み空白が含まれている場合は、'JAN'S DB' のように名前を単一引用符で囲む ('...') 必要があります。このパラメータ指定がなければ、現在のデータベース名が保持されます。

**NIRABN：ノーマルインデックスの開始 RABN**

NIRABN は、ファイルのノーマルインデックスエクステントの開始 RABN を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**NIRELEASE：未使用ノーマルインデックスブロックの解放**

NIRELEASE が指定されると、指定ファイルに属する未使用のノーマルインデックス (NI) ブロックが解放されます。NIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた NI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**NISIZE：ノーマルインデックスサイズ**

NISIZE は、ファイルのノーマルインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**NOUSERABEND：アベンドなし終了**

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベン  
ドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION" (エ  
ラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了しま  
す。

#### **RAID**：ファイルが存在しないと判断した場合の動作

RAIDは、ADAORDに、データベースに存在しないファイルを参照するあらゆるFILEパラ  
メータを無視するように指示します。

RAIDが指定されない場合（デフォルト）、ADAORDは、データベースに存在しないファ  
イルを参照するFILEパラメータを検出すると、エラーメッセージを出力して終了します。

RAIDパラメータは、Adabas Recovery Aid (ADARAI) で構築されたリカバリジョブで使  
用する場合に指定します。

#### **SORTSEQ**：ファイルの処理順序

SORTSEQは、ファイルを処理する順番を指定します。このパラメータを省略すると、レ  
コードが物理順で処理されます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。空値  
省略されたディスクリプタフィールド、ハイパーディスクリプタ、フォネティックディスク  
リプタ、マルチプルバリュースフィールド、またはピリオディックグループ内のディスクリ  
プタは指定してはなりません。



**Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない（値  
を持たない）場合は、空値省略されたディスクリプタフィールドでなくても、そのレ  
コードはインバーテッドリストに含まれません。したがって、そのレコードはユー  
ティリティの実行時に除外されます。

ISNを指定すると、ファイルがISN昇順に処理されます。また、Adabasチェックポイント  
またはセキュリティファイルについては、SORTSEQ=ISNしか許可されません。

#### **TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文  
だけがテストされ、値や変数の正当性はテストされません。

#### **UIRABN**：アッパーインデックスの開始 RABN

UIRABNは、ファイルに対するアソシエータのアッパーインデックスエクステンツの開始  
RABNを定義します。このパラメータ指定がなければ、ADAORDが開始RABNを割り当  
てます。

#### **UIRELEASE**：未使用アッパーインデックスブロックの解放

UIRELEASEは、ファイルに属する未使用のアッパーインデックス (UI) ブロックを解放し  
ます。UIRELEASEが指定されない場合には、少なくともそのファイルがリオーダされる以  
前に割り当てられたUIブロックがADAORDにより割り当てられます。



**Note:** Adabasは、あらゆる変更されたパディングファクタまたはファイルがリオーダ  
される前のブロックサイズ値を用いてファイルのエクステンツサイズを計算します。

**UISIZE**：アッパーインデックスサイズ

UISIZE は、アッパーインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。UISIZE が省略されると、ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によってスペースが割り当てられます。

**例****例 1**

```
ADAORD REORDB
```

アソシエータおよびデータストレージをリオーダします。現在のデータベースパラメータへの変更はありません。

**例 2**

```
ADAORD REORDB
ADAORD MAXFILES=200
ADAORD NEWDBID=6,NEWDBNAME=DATABASE-6
```

アソシエータおよびデータストレージをリオーダします。データベースは最大 200 ファイルまで許容できます。新しいデータベース ID および名前は各々 6 および DATABASE-6 です。REORDB の実行中にニュークリアスがアクティブになっている場合は、実行後に停止されます (NEWDBID が指定されました)。

**例 3**

```
ADAORD REORDB
ADAORD FILE=1,ACRABN=1000,NIRABN=2200,SORTSEQ=ISN
ADAORD FILE=2,MAXISN=500000
ADAORD
FILE=4,ASSOPFAC=5,DATAPFAC=20,DSSIZE=5,DSRABN=1
```

アソシエータおよびデータストレージをリオーダします。ファイル 1 のアドレスコンバータの割り当ては RABN 1,000 から開始します。ファイル 1 のノーマルインデックスの割り当ては RABN 2,200 から開始します。

ファイル 1 のデータストレージ部分を、ISN で昇順にリオーダします。ファイル 2 の MAXISN は 500,000 にセットします。ファイル 4 に対して次の指定を行います。アソシエータブロックパディングファクタを 5% に変更し、データストレージブロックパディングファクタは 20% にセットし、RABN1 から開始して 5 つのシリンダに新しい DSSIZE を割り当てます。既存の定義に従って、他のすべてのファイルをリオーダします。

**例 4**

```
ADAORD REORDB
ADAORD FILE=66
ADAORD DSRELEASE
ADAORD NIRELEASE
ADAORD UIRELEASE
ADAORD RAID
```

1. データベース全体をリオーダします。
2. ファイル 66 のデータストレージ、ノーマルインデックス、およびアップパーインデックスから未使用のストレージをすべて解放します。
3. ただし、ファイル 66 がデータベースに存在しない場合、ADAORD はエラーメッセージを出力して終了するのではなく、この状態を無視して処理を続行します。

#### 例 5

```
ADAORD REORDB
ADAORD DBINDEXCOMPRESSION=YES
ADAORD FILE=1
ADAORD FILE=2, INDEXCOMPRESSION=NO
ADAORD FILE=3
```

非圧縮形式のインデックスで再構築されているファイル2を除き、圧縮形式のインデックスを使用してすべてのファイルをリオーダおよび再構築します。



# 124 REORFASSO : 1つのファイルに対するアソシ

## エータのリオーダ

---

■ 基本パラメータ .....	625
■ オプションパラメータ .....	625
■ 例 .....	628

REORFASSO 機能は、1つのファイルに対してアソシエータをリオーダします。指定されないファイルのアソシエータ情報はリオーダされません。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORD の実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能に指定されたファイルがもともと ISNREUSE=YES がアクティブな状態でロードされていた場合、このリオーダ機能は、そのファイルのファイルコントロールブロック (FCB) 内の最初の未使用 ISN 値をアドレスコンバータ内で最初に検出された実際の未使用 ISN にリセットします。

次に、ADAORD REORFASSO 機能の構文を示します。

```
ADAORD REORFASSO [FILE = file-number ]  
    [ACRABN = starting-rabn ]  
    [AC2RABN = starting-rabn ]  
    [ALLOCATION = { FORCE | NOFORCE } ]  
    [ASSOPFAC = padding-factor ]  
    [ASSOVOLUME = 'Associator-extent-volume' ]  
    [INDEXCOMPRESSION = { YES | NO } ]  
    [ISNSIZE = { 3 | 4 } ]  
    [MAXISN = highest-isn ]  
    [MAXISN2 = highest-isn ]  
    [NIRABN = starting-rabn ]  
    [NIRELEASE]  
    [NISIZE = size ]  
    [PASSWORD = password ]  
    [UIRABN = starting-rabn ]  
    [UIRELEASE]  
    [UIsize = size ]  
    [EXCLUDE = file-list ]  
    [LPB = { prefetch-buffer-size | ADARUN-lu } ]  
    [NOUSERABEND]  
    [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FILEは、処理対象となり、後続ステートメント内のパラメータが適用されるファイルを指定します。複数のファイルおよび関連するパラメータを1回のREORFASSO操作で指定できます。例については、このセクションの最後を参照してください。Adabas拡張ファイルのコンポーネントファイルを指定した場合は、指定したコンポーネントファイルのアソシエータだけがリオーダされます。他のコンポーネントファイルはまったく影響を受けません。

## オプションパラメータ

### ACRABN：アドレスコンバータの開始RABN

ACRABNは、ファイルのアドレスコンバータの開始RABNを指定します。このパラメータ指定がなければ、ADAORDが開始RABNを割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。

アソシエータエクステントの開始RABNを指定するときは、FCB、FDTおよびDSSTに必要なスペースも考慮する必要があります。

### AC2RABN：セカンダリアドレスコンバータの開始RABN

ファイルのセカンダリアドレスコンバータエクステントの開始RABNです。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリISNを、セカンダリレコードが格納されているデータストレージブロックのRABNにマッピングするために使用されます。

このパラメータ指定がなければ、ADAORDが開始RABNを割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

### ALLOCATION：ファイルエクステント割り当てができない場合の動作

ALLOCATIONは、配置パラメータACRABN、NIRABNまたはUIRABNに従ってファイルエクステント割り当てが得られない場合に取りべき動作を指定します。

デフォルト（すなわちALLOCATION=FORCE）の場合は、RABN配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCEの場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### ASSOPFAC：アソシエータパディングファクタ

ASSOPFACは、リオーダ処理中に使用されないアソシエータブロック内のパーセントを表すアソシエータブロックパディングファクタを定義します。1から90までの範囲の値が指定できます。パディング後の空きバイト数は、最も大きいディスクリプタ値+10より大きくなければなりません。

このパラメータを省略すると、ファイルの現在のパディングファクタが使用されます。

### ASSOVOLUME：アソシエータエクステントボリューム

 **Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UIエクステント）を割り当てるボリュームを指定します。要求スペースが指定ボリュームに存在しないと、ADAORD は、デフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

ACRABN、UIRABN、または NIRABN が指定されていると、ADAORD は、該当するエクステントタイプを割り当てるときに、ASSOVOLUME を無視します。

ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADAORD のデフォルト割り当てルールに従って割り当てられます。

### EXCLUDE：リオーダから指定されたファイルの除外

EXCLUDE は、REORDER 処理から除外されるファイル（つまりリオーダされないファイル）をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

EXCLUDE パラメータで指定されるファイルは、FILE パラメータでも指定する必要があります。

EXCLUDE パラメータは、Adabas Recovery Aid（ADARAI）によって構築されたりカバリジョブで使用するためのパラメータです。

### INDEXCOMPRESSION：ファイルインデックスの圧縮

INDEXCOMPRESSION は、ファイルが圧縮形式または非圧縮形式で再構成されるかどうかを示します。通常、インデックスを圧縮すると、インデックススペースが少なくなり、Adabas ニュークリアスのインデックス操作の効率が向上します。

INDEXCOMPRESSION を指定しない場合、ファイルの現在の形式がデフォルトになります。

### ISNSIZE：3 または 4 バイト ISN

ISNSIZE はファイル内の ISN 長が3バイトであるか4バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック（FCB）内に格納されます。

 **Note:** 物理的にカップリングされているファイルの ISNSIZE を ADAORD で変更することはできません。

### LPB：プリフェッチバッファサイズ

LPB は、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能最大値は、32,760 バイトです。デフォルトは、ADARUNLU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなる場合があります。

**MAXISN：割り当てる最大 ISN**

MAXISN は、ファイルに対して割り当てる最大 ISN を指定します。この値は、ADAREP データベースレポートに出力される現在の TOPISN より大きい必要があります。

指定された値は、ADAORD によりアドレスコンバータスペース必要量の計算に使用されます。このパラメータを省略すると、ファイルの現在の MAXISN 値が保持されます。

**MAXISN2：ファイルに対する可能最大セカンダリ ISN**

MAXISN は、ISN 内のセカンダリアドレスコンバータ (AC2) のサイズを指定します。この値は、ADAREP データベースレポートに出力される現在の TOP AC2 ISN より大きい必要があります。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

ADAORD は、ファイルのセカンダリアドレスコンバータに必要なスペースを計算するために指定された値を使用します。このパラメータが省略されると、ファイルに対する現在の MAXISN2 値が保持されます。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

**NIRABN：ノーマルインデックスの開始 RABN**

NIRABN は、ノーマルインデックスの開始 RABN を指定します。NIRABN を省略すると、開始 RABN が割り当てられます。

**NIRELEASE：未使用ノーマルインデックスブロックの解放**

NIRELEASE が指定されると、指定ファイルに属する未使用のノーマルインデックス (NI) ブロックが解放されます。NIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた NI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**NISIZE：ノーマルインデックスサイズ**

NISIZE は、ファイルのノーマルインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**NOUSERABEND：アベンドなし終了**

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**UIRABN**：アッパーインデックスの開始 RABN

UIRABN は、アッパーインデックスの開始 RABN を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**UIRELEASE**：未使用アッパーインデックスブロックの解放

UIRELEASE が指定されると、指定ファイルに属する未使用のアッパーインデックス (UI) ブロックが解放されます。UIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた UI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**UISIZE**：アッパーインデックスサイズ

UISIZE は、ファイルのアッパーインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

## 例

---

### 例 1

```
ADAORD REORFASSO FILE=9 ADAORD ASSOPFAC=5 ADAORD FILE=23 ADAORD  
UIRABN=3151,UISIZE=50B ADAORD NIRABN=3201
```

ファイル 9 および 23 のアソシエータをリオーダします。他のファイルのアソシエータデータは変更されません。

ファイル 9 のアソシエータパディングファクタは 5 % にセットします。ファイル 23 のアッパーインデックスの新しい開始 RABN を 3151、そのサイズを 50 ブロックに変更します。ノーマルインデックスの新しい開始 RABN は 3201、そのサイズは変更前と同じです。

### 例 2

```
ADAORD REORFASSO FILE=104 ADAORD ASSOPFAC=5,NISIZE=5B,UISIZE=2B ADAORD  
ACRABN=10000,NIRABN=10510,UIRABN=10515 ADAORD FILE=105
```

ファイル104および105のアソシエータをリオーダします。他のファイルのアソシエータ情報は変更されません。

ファイル104のアソシエータパディングファクタは5にセットします。ノーマルインデックスおよびアッパーインデックスのサイズは、それぞれ5ブロックおよび2ブロックになります。アドレスコンバータの開始RABNは10000になります。ノーマルインデックスの開始RABNは10510になります。アッパーインデックスの開始RABNは10515になります。既存の定義に従って、ファイル105の情報をリオーダします。



# 125 REORFDATA : 1つのファイルに対するデータストレージのリオーダ

---

- 基本パラメータ ..... 632
- オプションパラメータ ..... 633
- 例 ..... 636

REORFDATA 機能は、1つのファイルに対してデータストレージをリオーダします。指定されないファイルのデータストレージはリオーダされません。

REORFDATA 機能は、ADAM ファイルをリオーダしません。しかし、これは、異なる RABN に ADAM ファイルの再配置に使用できます。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORD の実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

次に、ADAORD REORFDATA 機能の構文を示します。

```
ADAORD REORFDATA [FILE = file-number ]
                [ALLOCATION = { FORCE | NOFORCE } ]
                [DATAPFAC = padding-factor ]
                [DATAVOLUME = 'Data-Storage-extent-volume' ]
                [DSDEV = device-type ]
                [DSRABN = starting-rabn ]
                [DSRELEASE]
                [DSSIZE = size ]
                [MAXRECL = record-length ]
                [PASSWORD = password ]
                [SORTSEQ = { descriptor | ISN | physical-sequence } ]
                [EXCLUDE = file-list ]
                [LIP = { isn-pool-size | 16384 } ]
                [LPB = { prefetch-buffer-size | ADARUN-lu } ]
                [NOUSERABEND]
                [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### FILE：ファイル番号

FILEは、処理対象となり、後続ステートメント内のパラメータが適用されるファイルを指定します。複数のファイルおよび関連するパラメータを1回のREORFDATA操作で指定できます。例については、このセクションの最後を参照してください。

## オプションパラメータ

**ALLOCATION**：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ DSRABN に従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

デフォルト（ALLOCATION=FORCE）では、RABN 配置パラメータに従ったファイルエクステントの割り当てができない場合、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、割り当てができないと、配置パラメータに関係なく再度割り当てようとします。

**DATAPFAC**：データストレージパディングファクタ

DATAPFAC は、新データストレージパディングファクタを指定します。指定値は、リオーダ処理時レコード拡張のために確保される各データストレージブロックのパーセントを表します。1 から 90 までの範囲の値が指定できます（パディングファクタの設定の詳細については ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。このパラメータ指定がなければ、ファイルに対して現在有効なパディングファクタが使用されません。

**DATAVOLUME**：データストレージエクステントボリューム

 **Note:** DATAVOLUME パラメータは単一引用符でくくらなければなりません。

ファイルのデータストレージスペースエクステントを割り当てるときに使用するボリューム番号を指定します。DSSIZE で要求されたブロックの数が指定ボリュームに存在しないと、ADAORD はデフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。

DATAVOLUME が指定されない場合、データストレージスペースは、ADAORD のデフォルト割り当てルールに従って、割り当てられます。

**DSDEV**：データストレージデバイスタイプ

DSDEV は、そのファイルのデータストレージのデバイスタイプです。指定デバイスタイプは、通常、そのデータベースの作成時か、あるいは ADADBS ユーティリティの ADD 機能によって、すでに Adabas に対して定義されているものでなければなりません。

DSDEV が指定されない場合、ADAORD はリオーダ前に使用されていたデバイスタイプでファイルを割り当てようとします。

**DSRABN**：データストレージ開始 RABN

DSRABN は、ファイルのデータストレージエクステントの開始 RABN を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**DSRELEASE**：未使用データストレージブロックの解放

DSRELEASEを指定することで、ファイルに属する未使用のデータストレージ（DS）ブロックが解放されます。DSRELEASEが指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられたDSブロック数がADAORDによって割り当てられます。



**Note:** Adabasは、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**DSSIZE**：データストレージサイズ

DSSIZEは、データストレージに対して割り当てられるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に"B"を指定する必要があります（"2000B"など）。

このパラメータが省略されると、そのファイルのエクステントサイズはDATAPFACパディングファクタ内での全増減に比例してADAORDによって計算されます。

**EXCLUDE**：リオーダから指定されたファイルの除外

EXCLUDEは、REORDER処理から除外されるファイル（つまりリオーダされないファイル）をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

EXCLUDEパラメータで指定されるファイルは、FILEパラメータでも指定する必要があります。

EXCLUDEパラメータは、Adabas Recovery Aid（ADARAI）によって構築されたりカバリジョブで使用するためのパラメータです。

**LIP**：ISNバッファプールサイズ

LIPパラメータは、アドレスコンバータの再作成時のアソシエータの入出力処理の回数を減少させるために使用できます。処理対象となる中で一番規模が大きいファイルの全ISNを受け入れるだけの大きさを指定したときに最適のパフォーマンスとなります。

ISNと割り当てられたデータストレージRABNを入れるためのISNプールサイズを指定します。値はバイト単位で指定し、数値のみ（"2048"など）を指定するか、あるいは数値の後に"K"を付けて（"2K"など）キロバイト単位で指定します。LIPのデフォルトは16384バイト（または16K）です。

1件の入力レコード長は、ISNSIZE + RABNSIZEです。したがって、エントリ長は、最小6バイト（ファイルのISNSIZEが3でデータベースのRABNSIZEが3の場合）、最大8バイト（ISNSIZEが4でRABNSIZEが4の場合）になります。

**LPB**：プリフェッチバッファサイズ

LPBは、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能最大値は、32,760バイトです。デフォルトは、ADARUNLUパラメータの指定に依存します。LU値が小さすぎる場合には、LPB値が指定値よりも小さくなる場合があります。

**MAXRECL**：最大圧縮レコード長

ファイルの最大圧縮レコード長の制限を変更するには、MAXRECL パラメータを使用します。MAXRECL を指定すると、次の2つの効果があります。

- ファイルの DATA データセットは、指定された長さをサポートするデバイスに対してのみ割り当てられます。
- 指定された長さを超えるデータストレージレコードがファイル内に含まれている場合、ADAORD は異常終了し、ERROR-126 メッセージ（データストレージレコードが長すぎる）を出力します。

MAXRECL を指定しなければ、最大圧縮レコード長は変更されません。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

**SORTSEQ**：ファイルリオーダ順序

SORTSEQ は、ファイルを処理する順番を指定します。このパラメータを省略すると、レコードが物理順で処理されます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。空値省略されたディスクリプタフィールド、ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリュースフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。

 **Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない（値を持たない）場合は、空値省略されたディスクリプタフィールドでなくても、そのレコードはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

ISN を指定すると、ファイルが ISN 昇順に処理されます。また、Adabas チェックポイントまたはセキュリティファイルについては、SORTSEQ=ISN しか許可されません。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定パラメータの構文だけがテストされ、値や変数の正当性はチェックされません。

## 例

---

### 例 1

```
ADAORD REORFDATA FILE=16
```

ファイル 16 のデータストレージをリオーダします。他のファイルは影響を受けません。

### 例 2

```
ADAORD REORFDATA FILE=246  
ADAORD DATAPFAC=5,DSSIZE=10,SORTSEQ=MZ  
ADAORD FILE=247
```

ファイル 246 および 247 のデータストレージをリオーダします。他のファイルのデータストレージはリオーダされません。

ファイル 246 のデータストレージパディングファクタは 5 にセットします。ファイルの既存の定義に従って、ファイル 247 のデータストレージをリオーダします。

# 126 REORFILE：ファイルのリオーダ

---

■ 基本パラメータ .....	640
■ オプションパラメータ .....	640
■ 例 .....	814

REORFILE機能は、1つのファイルに対してアソシエータおよびデータストレージをリオーダします。他のファイルのアソシエータおよびデータストレージは影響を受けません。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORDの実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能に指定されたファイルがもともと ISNREUSE=YES がアクティブな状態でロードされていた場合、このリオーダ機能は、そのファイルのファイルコントロールブロック (FCB) 内の最初の未使用 ISN 値をアドレスコンバータ内で最初に検出された実際の未使用 ISN にリセットします。

次に、ADAORD REORFILE 機能の構文を示します。

```
ADAORD REORFILE [FILE = file-number ]
    [ACRABN = starting-rabn ]
    [AC2RABN = starting-rabn ]
    [ALLOCATION = { FORCE | NOFORCE } ]
    [ASSOPFAC = padding-factor ]
    [ASSOVOLUME = 'Associator-extent-volume' ]
    [DATAPFAC = padding-factor ]
    [DATAVOLUME = 'Data-Storage-extent-volume' ]
    [DSDEV = device-type ]
    [DSRABN = starting-rabn ]
    [DSRELEASE]
    [DSSIZE = size ]
    [INDEXCOMPRESSION = { YES | NO } ]
    [ISNSIZE = { 3 | 4 } ]
    [MAXISN = highest-isn ]
    [MAXISN2 = highest-isn ]
    [MAXRECL = record-length ]
    [NIRABN = starting-rabn ]
    [NIRELEASE]
    [NISIZE = size ]
    [PASSWORD = password ]
    [SORTSEQ = { descriptor | ISN | physical-sequence } ]
    [UIRABN = starting-rabn ]
    [UIRELEASE]
    [UIsize = size ]
    [EXCLUDE = file-list ]
    [LIP = { isn-pool-size | 16384 } ]
    [LPB = { prefetch-buffer-size | ADARUN-lu } ]
    [NOUSERABEND]
    [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FILEは、処理対象となり、後続ステートメント内のパラメータが適用されるファイルを指定します。複数のファイルおよび関連するパラメータを1回のREORFILE操作で指定できます。例については、このセクションの最後を参照してください。ADABAS拡張ファイルのコンポーネントファイルを指定した場合、そのファイルのアソシエータおよびデータストレージのみがリオーダーされ、他のコンポーネントファイルには悪影響を与えません。

## オプションパラメータ

### ACRABN：アドレスコンバータの開始RABN

ACRABNは、ファイルのアドレスコンバータエクステントの開始RABNを指定します。このパラメータ指定がなければ、ADAORDが開始RABNを割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。

アソシエータエクステントの開始RABNを指定するときは、FCB、FDTおよびDSSTに必要なスペースも考慮する必要があります。

### AC2RABN：セカンダリアドレスコンバータの開始RABN

ファイルのセカンダリアドレスコンバータエクステントの開始RABNです。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリISNを、セカンダリレコードが格納されているデータストレージブロックのRABNにマッピングするために使用されます。

このパラメータ指定がなければ、ADAORDが開始RABNを割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

### ALLOCATION：ファイルエクステント割り当てができない場合の動作

ALLOCATIONは、配置パラメータACRABN、DSRABN、NIRABNまたはUIRABNに従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

デフォルト（すなわちALLOCATION=FORCE）の場合は、RABN配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCEの場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### ASSOPFAC：アソシエータパディングファクタ

ASSOPFACは、新アソシエータブロックパディングファクタを指定します。指定値は、リオーダー処理時に使用しない各アソシエータブロックのパーセントを表します。1から90の値が指定できます。パディング後に残っているバイト数は、最大ディスクリプタ値に10を加えたものより大きくなければなりません。

このパラメータ指定がなければ、現在ファイルに有効なアソシエータパディングファクタが使用されます。

#### ASSOVOLUME：アソシエータエクステントボリューム

 **Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UIエクステント）を割り当てるボリュームを指定します。要求スペースが指定ボリュームに存在しないと、ADAORD は、デフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

ACRABN、UIRABN、または NIRABN が指定されていると、ADAORD は、該当するエクステントタイプを割り当てるときに、ASSOVOLUME を無視します。

ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADAORD のデフォルト割り当てルールに従って割り当てられます。

#### DATAPFAC：データストレージパディングファクタ

DATAPFAC は、新データストレージパディングファクタを指定します。指定値は、リオーダ処理時レコード拡張のために確保される各データストレージブロックのパーセントを表します。1 から 90 までの範囲の値が指定できます（パディングファクタの設定の詳細については ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。このパラメータ指定がなければ、ファイルに対して現在有効なパディングファクタが使用されます。

#### DATAVOLUME：データストレージエクステントボリューム

 **Note:** DATAVOLUME パラメータは単一引用符でくくらなければなりません。

ファイルのデータストレージスペースエクステントを割り当てるときに使用するボリューム番号を指定します。DSSIZE で要求されたブロックの数が指定ボリュームに存在しないと、ADAORD はデフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。

DATAVOLUME が指定されない場合、データストレージスペースは、ADAORD のデフォルト割り当てルールに従って、割り当てられます。

#### DSDEV：データストレージデバイスタイプ

DSDEV は、ファイルのデータストレージに対して使用されるデバイスタイプを指定します。指定デバイスタイプは、通常、そのデータベースの作成時か、あるいは ADADBS ユーティリティの ADD 機能によって、すでに Adabas に対して定義されているものでなければなりません。

このパラメータを指定しないと、ADAORD はリオーダ前に使用されていたデバイスタイプでファイルを割り当てようとしています。

**DSRABN**：データストレージ開始 RABN

DSRABN は、ファイルのデータストレージエクステントの開始 RABN を指定します。  
DSRABN を省略すると、開始 RABN が割り当てられます。

**DSRELEASE**：未使用データストレージブロックの解放

DSRELEASE を指定することで、ファイルに属する未使用のデータストレージ (DS) ブロックが解放されます。DSRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた DS ブロック数が ADAORD によって割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**DSSIZE**：データストレージサイズ

DSSIZE は、データストレージに対して割り当てられるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは DATAPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**EXCLUDE**：リオーダから指定されたファイルの除外

EXCLUDE は、REORDER 処理から除外されるファイル (つまりリオーダされないファイル) をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

EXCLUDE パラメータで指定されるファイルは、FILE パラメータでも指定する必要があります。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

**INDEXCOMPRESSION**：ファイルインデックスの圧縮

INDEXCOMPRESSION は、ファイルが圧縮形式または非圧縮形式で再構成されるかどうかを示します。通常、インデックスを圧縮すると、インデックススペースが少なくなり、Adabas ニュークリアスのインデックス操作の効率が向上します。

INDEXCOMPRESSION を指定しない場合、ファイルの現在の形式がデフォルトになります。

**ISNSIZE**：3 または 4 バイト ISN

ISNSIZE はファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック (FCB) 内に格納されます。



**Note:** 物理的にカップリングされているファイルの ISNSIZE を ADAORD で変更することはできません。

**LIP：ISN バッファプールサイズ**

LIP パラメータは、アドレスコンバータの再作成時のアソシエータの入出力処理の回数を減少させるために使用できます。処理対象となる中で一番規模が大きいファイルの全 ISN を受け入れるだけの大きさを指定したときに最適のパフォーマンスとなります。

ISN と割り当てられたデータストレージ RABN を入れるための ISN プールサイズを指定します。値はバイト単位で指定し、数値のみ ("2048" など) を指定するか、あるいは数値の後に "K" を付けて ("2K" など) キロバイト単位で指定します。LIP のデフォルトは 16384 バイト (または 16K) です。

1 件の入力レコード長は、ISNSIZE + RABNSIZE です。したがって、エントリ長は、最小 6 バイト (ファイルの ISNSIZE が 3 でデータベースの RABNSIZE が 3 の場合)、最大 8 バイト (ISNSIZE が 4 で RABNSIZE が 4 の場合) になります。

**LPB：プリフェッチバッファサイズ**

LPB は、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能な最大値は、32,760 バイトです。デフォルトは、ADARUN LU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなる場合があります。

**MAXISN：ファイルに対する可能最大 ISN**

MAXISN は、ファイルに対して割り当てる最大 ISN を指定します。この値は、ADAREP データベースレポートに出力される現在の TOPISN より大きい必要があります。

指定された値は、ADAORD によりアドレスコンバータスペース必要量の計算に使用されます。このパラメータが省略されると、ファイルに対する現在の MAXISN 値が保持されます。

**MAXISN2：ファイルに対する可能最大セカンダリ ISN**

MAXISN は、ISN 内のセカンダリアドレスコンバータ (AC2) のサイズを指定します。この値は、ADAREP データベースレポートに出力される現在の TOP AC2 ISN より大きい必要があります。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

ADAORD は、ファイルのセカンダリアドレスコンバータで必要なスペースを計算するために指定された値を使用します。このパラメータが省略されると、ファイルに対する現在の MAXISN2 値が保持されます。ファイルにセカンダリアドレスコンバータエクステンが含まれていない場合、このパラメータは無視されます。

**MAXRECL：最大圧縮レコード長**

ファイルの最大圧縮レコード長の制限を変更するには、MAXRECL パラメータを使用します。MAXRECL を指定すると、次の 2 つの効果があります。

- ファイルの DATA データセットは、指定された長さをサポートするデバイスに対してのみ割り当てられます。
- 指定された長さを超えるデータストレージレコードがファイル内に含まれている場合、ADAORD は異常終了し、ERROR-126 メッセージ (データストレージレコードが長すぎる) を出力します。

MAXRECL を指定しなければ、最大圧縮レコード長は変更されません。

**NIRABN**：ノーマルインデックスの開始 RABN

NIRABN は、ノーマルインデックスエクステントの開始 RABN を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**NIRELEASE**：未使用ノーマルインデックスブロックの解放

NIRELEASE が指定されると、ファイルに属する未使用のノーマルインデックス (NI) が解放されます。NIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた NI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**NISIZE**：ノーマルインデックスサイズ

NISIZE は、ファイルのノーマルインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**PASSWORD**：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

**SORTSEQ**：リオーダ順序

SORTSEQ は、ファイルを処理する順番を指定します。このパラメータを省略すると、レコードが物理順で処理されます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。空値省略されたディスクリプタフィールド、ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。



**Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない (値を持たない) 場合は、空値省略されたディスクリプタフィールドでなくても、そのレコードはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

ISN を指定すると、ファイルが ISN 昇順に処理されます。また、Adabas チェックポイントまたはセキュリティファイルについては、SORTSEQ=ISN しか許可されません。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**UIRABN**：アッパーインデックスの開始 RABN

UIRABN は、ファイルのアッパーインデックスエクステントの開始 RABN を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

**UIRELEASE**：未使用アッパーインデックスブロックの解放

UIRELEASE が指定されると、ファイルに属する未使用のアッパーインデックス (UI) が解放されます。UIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた UI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

**UISIZE**：アッパーインデックスのインデックスサイズ

UISIZE は、アッパーインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステントサイズは ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

## 例

---

### 例 1

```
ADAORD REORFILE FILE=16
```

ファイル 16 のアソシエータおよびデータストレージをリオーダします。ファイルの現在のエクステントサイズおよびパディングファクタはすべて保持されます。他のファイルはリオーダされません。

### 例 2

```
ADAORD REORFILE FILE=246
ADAORD DATAPFAC=5,DSSIZE=10B,SORTSEQ=MZ
ADAORD ASSOPFAC=20,MAXISN=5000
ADAORD FILE=20
```

ファイル 246 をリオーダします。他のファイルのアソシエータおよびデータストレージは変更されません。データストレージパディングファクタは 5 にセットします。10 個のシリンダの新しいデータストレージサイズを使用します。データストレージをディスクリプタ MZ の論理順にリオーダします。新しいアソシエータパディングファクタは 20 です。指定可能な最大 ISN は 5000 です。現在のパラメータに変更を加えずに、ファイル 20 をリオーダします。

## 例 3

```
ADAORD  REORFILE  FILE=9
ADAORD          ASSOPFAC=5, DATAPFAC=15
ADAORD          FILE=23
ADAORD          DSRABN=24032
ADAORD          UIRABN=3151, UISIZE=50B
ADAORD          NIRABN=3201
```

ファイル 9 および 23 のアソシエータをリオーダします。他のファイルは変更されません。

ファイル 9、10、および 12 のアソシエータパディングファクタは 5 % にセットし、データストレージパディングファクタは 15 % にセットします。ファイル 23 の新しいデータストレージ開始 RABN は 24032 です。また、ファイル 23 のアッパーインデックスの新しい開始 RABN を 3151、そのサイズを 50 ブロックに変更します。ノーマルインデックスの新しい開始 RABN は 3201、そのサイズは変更前と同じです。

# 127

## RESTRUCTUREDB: データベースの再構築

---

- オプションのパラメータとサブパラメータ ..... 649
- 例 ..... 652

RESTRUCTUREDB 機能は、データベース全体をシーケンシャルデータセットにアンロードします。このデータセットを STORE 機能の入力として使用し、データを新しいデータベースにロードすることができます。ターゲットデータベースは、ソースデータベースと異なる物理デバイスタイプにあってもかまいません。アソシエータおよびデータストレージは RESTRUCTURE/STORE 処理の一部としてリオーダされます。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORD の実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能に指定されたファイルがもともと ISNREUSE=YES がアクティブな状態でロードされていた場合、このリオーダ機能は、そのファイルのファイルコントロールブロック (FCB) 内の最初の未使用 ISN 値をアドレスコンバータ内で最初に検出された実際の未使用 ISN にリセットします。

RESTRUCTUREDB 機能でオーバーフローエリアを使用する ADAM ファイルを再構築した後、STORE 機能を使用して、DATA の小さいデータベースに再構築されたファイルを格納すると、ADAORDERROR-103 が発生することがあります。代わりに、ADAULD および ADALOD ユーティリティを使用して ADAM ファイルを移動します。



**Note:** ADAORDSTORE を使用すると、Adabas 8 以前のバージョンのデータベースやファイルを再構築して、Adabas 8 データベースに格納できます。ただし、再構築した Adabas 8 データベース出力、または Adabas 7 などの以前のバージョンで実行中のデータベースに含まれるファイルは、格納できません。

次に、ADAORD RESTRUCTUREDB 機能の構文を示します。

```

ADAORD { RESTRUCTUREDDB | REDB }
  [DBASSODEV = { device-type | ADARUN-device }]
  [DBDATADEV = { device-type | ADARUN-device }]
  [DBINDEXCOMPRESSION = { YES | NO }]
  [FILE = file-number ]
  [ASSOPFAC = padding-factor ]
  [ASSODEV = { device-type | DBASSODEV-value }]
  [DATADEV = { device-type | DBDATADEV-value }]
  [DATAPFAC = padding-factor ]
  [INDEXCOMPRESSION = { YES | NO }]
  [ISNSIZE = { 3 | 4 }]
  [SORTSEQ = { descriptor | ISN | physical-sequence }]
  [LPB = { prefetch-buffer-size | ADARUN-lu }]
  [NOUSERABEND]
  [TEST]

```

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

### ASSODEV：アソシエータデバイスタイプ

ASSODEVには、ファイルの ASSO データセット用に新しいデータベースが使用するデバイスタイプを指定します。このパラメータは、使用するデバイスタイプがデフォルトのデバイスタイプと異なる場合にのみ必要となります。デフォルトのデバイスタイプは、DBASSODEV パラメータ値となります。DBASSODEVを指定しなければ、ADARUN DEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。これらのパラメータは、DDFILEA/FILEA データセットに書かれるデータには影響を及ぼしません。

### ASSOPFAC：アソシエータパディングファクタ

ADAORD は、ASSOPFAC 値を使用して、指定ファイルの STORE 機能の実行に必要なスペースを計算します。有効な値の範囲は 1~90 です。ASSODEV によって指定されたデバイスタイプおよび ASSOPFAC によって指定されたパディングファクタにより AC、NI、UI ブロック数を計算します。これらのパラメータは、DDFILEA に書かれたデータには影響を及ぼしません。ASSOPFAC を指定しなければ、ファイルの現在のパディングファクタが使用されます。

### DATADEV：データストレージデバイスタイプ

DATADEV は指定ファイルの新 DATA データセットに使用するデバイスタイプを指定します。このパラメータは、使用するデバイスタイプがデフォルトのデバイスタイプと異なる場合にのみ必要となります。デフォルトのデバイスタイプは、DBDATADEV パラメータ値と

なります。DBDATADEV を指定しなければ、ADARUN DEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。これらのパラメータは、DDFILEA に書かれたデータには影響を及ぼしません。

**DATAPFAC**：データストレージパディングファクタ

ADAORD は指定ファイルの STORE 機能の実行に必要なスペースを計算するために DATAPFAC を使用します。1~90 の範囲の値が指定できます（パディングファクタの設定についての詳細は、ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。DATADEV によって指定されたデバイスタイプおよび DATAPFAC によって指定されたパディングファクタによりデータストレージブロック数を計算します。DATAPFAC を指定しなければ、ファイルの現在のパディングファクタが使用されます。これらのパラメータは、DDFILEA に書かれたデータには影響を及ぼしません。

**DBASSODEV**：アソシエータに対するデフォルトデバイスタイプ

DBASSODEV は新 ASSO データセットのデフォルトデバイスタイプを指定します。ADAORD はリストラクチャされた各ファイルの ASSO スペース必要量を計算するときに、このパラメータに指定されたデバイスタイプを使用します。DBASSODEV を指定しなければ、ADARUNDEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。

ファイルのデフォルトデバイスタイプを変更するときは、FILE および ASSODEV パラメータを使用します。

**DBDATADEV**：データストレージに対するデフォルトデバイスタイプ

DBDATADEV は新 DATA データセットのデフォルトデバイスタイプを指定します。ADAORD はリストラクチャされた各ファイルの DATA スペース必要量を計算するときに、このパラメータに指定されたデバイスタイプを使用します。DBDATADEV を指定しなければ、ADARUNDEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。

ファイルのデフォルトデバイスタイプを変更するときは、FILE および DATADEV パラメータを使用します。

**DBINDEXCOMPRESSION**：データベースのインデックスサイズの計算

DBINDEXCOMPRESSION は、すべてのファイルに対して ADAORD で計算および表示されるインデックススペースが、圧縮形式または非圧縮形式のどちらのインデックスに基づいているかを示します。INDEXCOMPRESSION パラメータが未指定の全ファイルに適用されます。

DBINDEXCOMPRESSION は、データベース内のすべてのファイルについて圧縮／非圧縮インデックスのサイズを計算するために使用でき、ファイルごとにサイズを計算する必要がなくなります。

**FILE**：ファイル番号

FILE は、後続のパラメータセットが適用されるファイルを指定します。このパラメータで指定されないファイルのレコードは、ファイルの物理順にアンロードされます。

Adabas チェックポイントファイルまたはセキュリティファイルが指定されている場合は、SORTSEQ パラメータを指定しません。

**INDEXCOMPRESSION** : ファイルのインデックスサイズの計算

INDEXCOMPRESSION は、指定ファイルに対するインデックススペースの計算と表示を圧縮形式または非圧縮形式のいずれで行うかを指定します。

INDEXCOMPRESSION を指定せず、

- DBINDEXCOMPRESSION パラメータがデータベース全体に対して指定された場合は、データベース値がデフォルトになります。
- DBINDEXCOMPRESSION が指定されない場合は、ファイルの現在の圧縮形式がデフォルトになります。

**ISNSIZE** : 3 または 4 バイト ISN

ISNSIZE はファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック (FCB) 内に格納されます。



**Note:** 物理的にカップリングされているファイルの ISNSIZE を ADAORD で変更することはできません。

**LPB** : プリフェッチバッファサイズ

LPB は、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能な最大値は、32,760 バイトです。デフォルトは、ADARUN LU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなることがあります。

**NOUSERABEND** : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

**SORTSEQ** : ファイルの処理順序

SORTSEQ は、ファイルを処理する順番を指定します。このパラメータを省略すると、レコードが物理順で処理されます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。空値省略されたディスクリプタフィールド、ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。



**Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない (値を持たない) 場合は、空値省略されたディスクリプタフィールドでなくても、そのレコードはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

ISN を指定すると、ファイルが ISN 昇順に処理されます。また、Adabas チェックポイントまたはセキュリティファイルについては、SORTSEQ=ISN しか許可されません。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

---

### 例 1

```
ADAORD RESTRUCTUREDB
```

RESTRUCTUREDB 機能を実行します。すべてのファイルを物理順にアンロードします。

### 例 2

```
ADAORD RESTRUCTUREDB FILE=146 ,SORTSEQ=MZ  
ADAORD FILE=151 ,SORTSEQ=TF
```

RESTRUCTUREDB 機能を実行します。ファイル 146 をディスクリプタ MZ の順にアンロードします。ファイル 151 をディスクリプタ TF の順にアンロードします。他のすべてのファイルは物理順にアンロードされます。

# 128

## RESTRUCTUREF：個別のファイルの再構築

---

■ 基本パラメータ .....	655
■ オプションパラメータ .....	655
■ 例 .....	658

RESTRUCTUREF 機能は、1つ以上のファイルをシーケンシャルデータセットにアンロードします。このデータセットを STORE 機能の入力として使用し、ファイルを別のデータベースにロードすることができます。ターゲットデータベースは、元のデータベースと異なる物理デバイスタイプにあってもかまいません。アソシエータおよびデータストレージは RESTRUCTURE/STORE 処理の一部としてリオーダされます。

RESTRUCTUREF では EXU 制御が必要であり、他ユーザーに対して対象となっているデータベースファイルの読み込みだけのアクセスを認めます。ただし、チェックポイントファイルまたはセキュリティファイルに関する操作では、データベースに対して排他制御を必要とすることに注意してください。また、ADAORD の実行中に、データベースの一部が上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能に指定されたファイルがもともと ISNREUSE=YES がアクティブな状態でロードされていた場合、このリオーダ機能は、そのファイルのファイルコントロールブロック (FCB) 内の最初の未使用 ISN 値をアドレスコンバータ内で最初に検出された実際の未使用 ISN にリセットします。

RESTRUCTUREF 機能 でオーバーフローエリアを使用する ADAM ファイルを再構築した後、STORE 機能を使用して、DATA の小さいデータベースに再構築されたファイルを格納すると、ADAORDERROR-103が発生することがあります。代わりに、ADAULD および ADALOD ユーティリティを使用して ADAM ファイルを移動します。



**Note:** ADAORDSTORE を使用すると、Adabas 8 以前のバージョンのデータベースやファイルを再構築して、Adabas 8 データベースに格納できます。ただし、再構築した Adabas 8 データベース出力、または Adabas 7 などの以前のバージョンで実行中のデータベースに含まれるファイルは、格納できません。

次に、ADAORD RESTRUCTUREF 機能の構文を示します。

```

ADAORD { RESTRUCTUREF | REF }
  FILE = file-number
    [ASSODEV = { device-type | DBASSODEV-value }]
    [ASSOPFAC = padding-factor ]
    [DATADEV = { device-type | DBDATADEV-value }]
    [DATAPFAC = padding-factor ]
    [DBASSODEV = { device-type | ADARUN-device }]
    [DBDATADEV = { device-type | ADARUN-device }]
    [INDEXCOMPRESSION = { YES | NO }]
    [ISNSIZE = { 3 | 4 }]
    [PASSWORD = password ]
    [SORTSEQ = { descriptor | ISN | physical-sequence }]
[LPB = { prefetch-buffer-size | ADARUN-lu }]
[NOUSERABEND]
[TEST]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：ファイル番号

FILEは、再構築するファイルを指定します。処理するファイルごとに別個のADAORD FILE ステートメントを指定し、そのファイルの関連パラメータを含む ADAORD ステートメントを続けて指定する必要があります。

カップリングされているファイルまたは拡張ファイルの一部であるファイルを指定すると、関連ファイルが自動的にファイルリストに追加されます。ファイルが追加されたことを示すメッセージは DDPRINT に出力されます。

## オプションパラメータ

### ASSODEV：アソシエータデバイスタイプ

ASSODEVは、指定ファイルの新しい ASSO データセットに使用するデバイスタイプを指定します。このパラメータは、使用するデバイスタイプがデフォルトのデバイスタイプと異なる場合にのみ必要となります。デフォルトのデバイスタイプは、DBASSODEV パラメータ値となります。DBASSODEVを指定しなければ、ADARUNDEVICE パラメータによって指

定されたデバイスタイプがデフォルトとなります。これらのパラメータは、DDFILEA/FILEA データセットに書かれるデータには影響を及ぼしません。

**ASSOPFAC：アソシエータパディングファクタ**

ADAORD は、ASSOPFAC 値を使用して、指定ファイルの STORE 機能の実行に必要なスペースを計算します。有効な値の範囲は 1～90 です。ASSODEV によって指定されたデバイスタイプおよび ASSOPFAC によって指定されたパディングファクタにより AC、NI、UI ブロック数を計算します。ASSOPFAC を指定しなければ、ファイルの現在のパディングファクタが使用されます。これらのパラメータは、DDFILEA に書かれたデータには影響を及ぼしません。

**DATADEV：データストレージデバイスタイプ**

DATADEV は指定ファイルの新 DATA データセットに使用するデバイスタイプを指定します。このパラメータは、使用するデバイスタイプがデフォルトのデバイスタイプと異なる場合にのみ必要となります。デフォルトのデバイスタイプは、DBDATADEV パラメータ値となります。DBDATADEV を指定しなければ、ADARUN DEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。これらのパラメータは、DDFILEA に書かれたデータには影響を及ぼしません。

**DATAPFAC：データストレージパディングファクタ**

ADAORD は指定ファイルの STORE 機能の実行に必要なスペースを計算するために DATAPFAC を使用します。1～90 の範囲の値が指定できます（パディングファクタの設定についての詳細は、ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。DATADEV によって指定されたデバイスタイプおよび DATAPFAC によって指定されたパディングファクタによりデータストレージブロック数を計算します。DATAPFAC を指定しなければ、ファイルの現在のパディングファクタが使用されます。これらのパラメータは、DDFILEA に書かれたデータには影響を及ぼしません。

**DBASSODEV：アソシエータに対するデフォルトデバイスタイプ**

DBASSODEV は新 ASSO データセットのデフォルトデバイスタイプを指定します。ADAORD はリストラクチャされた各ファイルの ASSO スペース必要量を計算するときに、このパラメータに指定されたデバイスタイプを使用します。DBASSODEV を指定しなければ、ADARUN DEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。

ファイルのデフォルトデバイスタイプを変更するときは、FILE および ASSODEV パラメータを使用します。

**DBDATADEV：データストレージに対するデフォルトデバイスタイプ**

DBDATADEV は新 DATA データセットのデフォルトデバイスタイプを指定します。ADAORD はリストラクチャされた各ファイルの DATA スペース必要量を計算するときに、このパラメータに指定されたデバイスタイプを使用します。DBDATADEV を指定しなければ、ADARUN DEVICE パラメータによって指定されたデバイスタイプがデフォルトとなります。

ファイルのデフォルトデバイスタイプを変更するときは、FILE および DATADEV パラメータを使用します。

**INDEXCOMPRESSION：ファイルのインデックスサイズの計算**

INDEXCOMPRESSION は、指定ファイルに対するインデックススペースの計算と表示を圧縮形式または非圧縮形式のいずれで行うかを指定します。

INDEXCOMPRESSION を指定せず、

- データベース全体に対して圧縮値が指定されている場合は、データベース値がデフォルトになります。
- データベースに対して圧縮値が指定されていない場合は、ファイルの現在の圧縮形式がデフォルトになります。

#### ISNSIZE：3 または 4 バイト ISN

ISNSIZE はファイル内の ISN 長が 3 バイトであるか 4 バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック (FCB) 内に格納されます。



**Note:** 物理的にカップリングされているファイルの ISNSIZE を ADAORD で変更することはできません。

#### LPB：プリフェッチバッファサイズ

LPB は、内部プリフェッチバッファの大きさをバイト単位で指定するものです。最大サイズは 32,760 バイトです。デフォルトは、ADARUN LU パラメータの指定に依存します。LU 値が小さすぎる場合には、LPB 値が指定値よりも小さくなることがあります。

#### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ v が表示され、ユーティリティはコンディションコード 20 で終了します。

#### PASSWORD：ファイルパスワード

ファイルがパスワード保護されている場合、このパラメータを使用してパスワードを指定します。

#### SORTSEQ：ファイルの処理順序

SORTSEQ は、ファイルを処理する順番を指定します。このパラメータを省略すると、レコードが物理順で処理されます。

ディスクリプタを指定すると、ファイルがディスクリプタ値の論理順で処理されます。空値省略されたディスクリプタフィールド、ハイパーディスクリプタ、フォネティックディスクリプタ、マルチプルバリューフィールド、またはピリオディックグループ内のディスクリプタは指定してはなりません。



**Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない (値を持たない) 場合は、空値省略されたディスクリプタフィールドでなくても、そのレコードはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

ISN を指定すると、ファイルが ISN 昇順に処理されます。また、Adabas チェックポイントまたはセキュリティファイルについては、SORTSEQ=ISN しか許可されません。

### TEST：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定されたパラメータのみをテストでき、値や変数の妥当性はテストされません。

## 例

---

### 例 1

```
ADAORD RESTRUCTUREF FILE=15
```

RESTRUCTUREF 機能を実行します。ファイル 15 を物理順にアンロードします。他ファイルはアンロードされません。

### 例 2

```
ADAORD RESTRUCTUREF FILE=25, SORTSEQ=KL  
ADAORD FILE=40, SORTSEQ=J3
```

RESTRUCTUREF 機能を実行します。ファイル 25 および 40 をアンロードします。ファイル 25 をディスクリプタ KL の順にアンロードします。ファイル 40 をディスクリプタ J3 の順にアンロードします。他ファイルはアンロードされません。

# 129 STORE：ファイルの格納

---

■ オプションのパラメータとサブパラメータ .....	662
■ 例 .....	668

STORE 機能は、RESTRUCTURE 機能で作成された出力を使用して、1つ以上のファイルを既存のデータベースにロードします。アソシエータおよびデータストレージはRESTRUCTURE/STORE 処理の一部としてリオーダされます。

ALLFILES パラメータを指定すると、入力データセットに含まれるすべてのファイルが格納されます。ALLFILES パラメータを指定しないと、FILE パラメータで指定されたファイルのみが格納されます。

ALLFILES を指定した場合でも、FILE パラメータで1つ以上のファイルを指定できます。STORE 機能では、FILE ステートメントで指定されたファイルが、そのファイルのFILE ステートメントの直後にあるサブパラメータで定義された内容に従ってロードされます。他のファイルはすべて、既存の定義に従ってロードされます。

データベース内の既存のファイルを上書きするには、OVERWRITE パラメータを指定する必要があります。

この機能は、オペレーション時に使用されるデータベースファイルに対して EXF 排他制御を必要とします。また、ADAORD の実行中に、データベースは部分的に上書きされます。したがって、ADAORD 機能を実行する前にデータベース（またはファイル）を ADASAV ユーティリティでバックアップしておいてください。

この機能で指定されたファイルが ISNREUSE=YES がアクティブな状態でロードされていた場合、そのファイルのコントロールブロック（FCB）にあるの最初の未使用の ISN 値は、アドレスコンバータで最初に検出された実際の未使用の ISN にリセットされます。



### Notes:

1. STORE 機能は、ADAM ファイルをリオーダしません。ただし、他の ADAORD 機能を併用して、ADAM ファイルを異なる RABN に再配置することはできます。
2. ADAM ファイルで以前にオーバーフローエリアを使用していた場合、再構築された ADAM ファイルを DATA ブロッキングファクタが前より小さいデバイスに格納すると、ユーティリティ ERROR 103 が発生する可能性があります。ADAM ファイルを異なるデバイスに再配置するには、ADAULD および ADALOD ユーティリティを使用します。
3. Adabas バージョン 5.1 または 5.2 のチェックポイントファイルおよびセキュリティファイルは、内部構造の変更のため、バージョン 5.3 のファイルに格納できません。

次に、ADAORD STORE 機能の構文を示します。

```
ADAORD STORE [ALLFILES]
[CHECKPOINT]
[FILE = file-number ]
  [ACRABN = starting-rabn ]
  [AC2RABN = starting-rabn ]
  [ALLOCATION = { FORCE | NOFORCE } ]
  [ASSOPFAC = padding-factor ]
  [ASSOVOLUME = 'Associator-extent-volume' ]
  [DATAPFAC = padding-factor ]
  [DATAVOLUME = 'Data-Storage-extent-volume' ]
  [DSDEV = device-type ]
  [DSRABN = starting-rabn ]
  [DSRELEASE]
  [DSSIZE = size ]
  [INDEXCOMPRESSION = { YES | NO } ]
  [INSIZE = { 3 | 4 } ]
  [MAXISN = highest-isn ]
  [MAXISN2 = highest-isn ]
  [MAXRECL = record-length ]
  [NIRABN = starting-rabn ]
  [NIRELEASE]
  [NISIZE = size ]
  [UIRABN = starting-rabn ]
  [UIRELEASE]
  [UIsize = size ]
[EXCLUDE = file-list ]
[LIP = { isn-pool-size | 16384 } ]
[NOUSERABEND]
[OVERWRITE]
[SECURITY]
[TEST]
[TRIGGER]
```

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

### ACRABN：アドレスコンバータの開始 RABN

ACRABNは、ファイルのアドレスコンバータエクステントの開始RABNを指定します。このパラメータ指定がなければ、ADAORDが開始RABNを割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。

アソシエータエクステントの開始RABNを指定するときは、FCB、FDTおよびDSSTに必要なスペースも考慮する必要があります。

### AC2RABN：セカンダリアドレスコンバータの開始 RABN

ファイルのセカンダリアドレスコンバータエクステントの開始 RABN です。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリISNを、セカンダリレコードが格納されているデータストレージブロックのRABNにマッピングするために使用されます。

このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。要求されたスペースは、1つのエクステントで使用できる必要があります。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

### ALLFILES：全ファイルを選択して格納

ALLFILES を指定すると、入力データセットに含まれるすべてのファイルがデータベースに格納されます。ALLFILES を指定しないと、FILE パラメータで指定されたファイルのみが格納されます。

カップリングされているファイルまたは拡張ファイルの一部であるファイルが入力データセットに含まれ、関連ファイルがデータセットに存在しないと、ファイルリストの不整合を示す ERROR-138 が返されます。STORE 機能を正常に実行する前に関連ファイルを追加しなければなりません。

### ALLOCATION：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### ASSOPFAC：アソシエータパディングファクタ

ASSOPFAC は、新アソシエータブロックパディングファクタを指定します。指定値は、リオーダ処理時に使用しない各アソシエータブロックのパーセントを表します。1 から 90 の値が指定できます。パディング後に残っているバイト数は、最大ディスクリプタ値に 10 を加えたものより大きくなければなりません。

このパラメータ指定がなければ、現在ファイルに有効なアソシエータパディングファクタが使用されます。

#### ASSOVOLUME：アソシエータエクステントボリューム

 **Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UIエクステント）を割り当てるボリュームを指定します。要求スペースが指定ボリュームに存在しないと、ADAORD は、デフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

ACRABN、UIRABN、または NIRABN が指定されていると、ADAORD は、該当するエクステントタイプを割り当てるときに、ASSOVOLUME を無視します。

ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADAORD のデフォルト割り当てルールに従って割り当てられます。

#### CHECKPOINT：チェックポイントファイルの格納

ALLFILES が指定されているか、FILE パラメータでチェックポイントファイルが指定されていると、DDFILEA/FILEA テープにあるチェックポイントファイルがデータベースに格納され、それが新しいチェックポイントファイルになります。新しいチェックポイントファイルのファイル番号は、古いチェックポイントファイルのものと同等である必要があります。

CHECKPOINT パラメータが指定されないと、FILE パラメータまたは ALLFILES パラメータでチェックポイントファイルが指定されていても、DDFILEA/FILEA テープのチェックポイントファイルはデータベースに格納されません。

#### DATAPFAC：データストレージパディングファクタ

DATAPFAC は、新データストレージパディングファクタを指定します。指定値は、リオーダー処理時レコード拡張のために確保される各データストレージブロックのパーセントを表します。1 から 90 までの範囲の値が指定できます（パディングファクタの設定の詳細については ADALOD LOAD DATAPFAC パラメータに関する説明を参照してください）。このパラメータ指定がなければ、ファイルに対して現在有効なパディングファクタが使用されます。

#### DATAVOLUME：データストレージエクステントボリューム

 **Note:** DATAVOLUME パラメータの値は、単一引用符で囲う必要があります。

ファイルのデータストレージスペースエクステントを割り当てるときに使用するボリューム番号を指定します。DSSIZE で要求されたブロックの数が指定ボリュームに存在しないと、ADAORD はデフォルト割り当てルールに従って、他ボリュームの残りのブロックを割り当てます。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。

DATAVOLUME が指定されない場合、データストレージスペースは、ADAORD のデフォルト割り当てルールに従って、割り当てられます。

### DSDEV：データストレージデバイスタイプ

DSDEVは、ファイルのデータストレージに対して使用されるデバイスタイプを指定します。指定デバイスタイプは、通常、そのデータベースの作成時か、あるいは ADADBS ユーティリティの ADD 機能によって、すでに Adabas に対して定義されているものでなければなりません。

このパラメータを指定しないと、ADAORD は再構築前に使用されていたデバイスタイプでファイルを割り当てようとします。

### DSRABN：データストレージ開始 RABN

指定されたファイルのデータストレージエクステンツの開始 RABN を指定します。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

### DSRELEASE：未使用データストレージブロックの解放

DSRELEASE を指定することで、指定ファイルに属する未使用のデータストレージ (DS) ブロックが解放されます。DSRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた DS ブロック数が ADAORD によって割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステンツサイズを計算します。

### DSSIZE：データストレージサイズ

DSSIZE は、データストレージに対して割り当てられるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、そのファイルのエクステンツサイズは DATAPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

### EXCLUDE：指定されたファイルを格納から除外

EXCLUDE は、STORE 処理から除外するファイル (格納されないファイル) の番号をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

EXCLUDE パラメータは、ALLFILES も指定されている場合にのみ指定できます。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

### FILE：ファイル番号

FILE は、格納するファイルを指定します。処理するファイルごとに別個のステートメントを指定し、そのファイルの関連パラメータを含む ADAORD ステートメントを続けて指定する必要があります。

カップリングされているファイルまたは拡張ファイルの一部であるファイルを指定し、関連ファイルを指定しないと、ファイルリストの不整合を示すERROR-138が返されます。STORE機能を正常に実行する前に関連ファイルを追加しなければなりません。

#### **INDEXCOMPRESSION**：ファイルインデックスの圧縮

INDEXCOMPRESSIONは、ファイルが圧縮形式または非圧縮形式で再構成されるかどうかを示します。通常、インデックスを圧縮すると、インデックススペースが少なくなり、Adabas ニュークリアスのインデックス操作の効率が向上します。

INDEXCOMPRESSIONを指定しないと、対応する再構築操作におけるファイルの形式がデフォルトになります。

#### **ISNSIZE**：3 または 4 バイト ISN

ISNSIZEはファイル内のISN長が3バイトであるか4バイトであるかを指定します。デフォルトはファイルに使用された現在の値です。この値はファイルコントロールブロック (FCB) 内に格納されます。



**Note:** 物理的にカップリングされているファイルのISNSIZEをADAORDで変更することはできません。

#### **LIP**：ISN バッファプールサイズ

LIPパラメータは、アドレスコンバータの再作成時のアソシエータの入出力処理の回数を減少させるために使用できます。処理対象となる中で一番規模が大きいファイルの全ISNを受け入れるだけの大きさを指定したときに最適のパフォーマンスとなります。

ISNと割り当てられたデータストレージRABNを入れるためのISNプールサイズを指定します。値はバイト単位で指定し、数値のみ("2048"など)を指定するか、あるいは数値の後に"K"を付けて("2K"など)キロバイト単位で指定します。LIPのデフォルトは16384バイト(または16K)です。

1件の入力レコード長は、ISNSIZE + RABNSIZEです。したがって、エントリ長は、最小6バイト(ファイルのISNSIZEが3でデータベースのRABNSIZEが3の場合)、最大8バイト(ISNSIZEが4でRABNSIZEが4の場合)になります。

#### **MAXISN**：ファイルに対する可能最大ISN

MAXISNは、ファイルに対して割り当て可能な最大ISNを指定します。この値は、ADAREPデータベースレポートに出力される現在のTOPISNより大きい必要があります。

指定された値は、ADAORDによりアドレスコンバータスペース必要量の計算に使用されます。このパラメータが省略されると、ファイルに対する現在のMAXISN値が保持されます。

#### **MAXISN2**：ファイルに対する可能最大セカンダリISN

MAXISNは、ISN内のセカンダリアドレスコンバータ(AC2)のサイズを指定します。この値は、ADAREPデータベースレポートに出力される現在のTOP AC2 ISNより大きい必要があります。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリISNを、セカンダリレコードが格納されているデータストレージブロックのRABNにマッピングするために使用されます。

ADAORD は、ファイルのセカンダリアドレスコンバータで必要なスペースを計算するために指定された値を使用します。このパラメータが省略されると、ファイルに対する現在の MAXISN2 値が保持されます。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

### MAXRECL：最大圧縮レコード長

ファイルの最大圧縮レコード長の制限を変更するには、MAXRECL パラメータを使用します。MAXRECL を指定すると、次の2つの効果があります。

- ファイルの DATA データセットは、指定された長さをサポートするデバイスに対してのみ割り当てられます。
- 指定された長さを超えるデータストレージレコードがファイル内に含まれている場合、ADAORD は異常終了し、ERROR-126 メッセージ（データストレージレコードが長すぎる）を出力します。

MAXRECL を指定しないと、次のいずれかがデフォルト値になります。

- ファイルが再構築される前の最大圧縮レコード長がデフォルトの ADALOD MAXRECL 値であった場合、DATA は任意のデバイスに割り当てられ、そのデバイスタイプから新しい最大レコード長が取得されます。
- それ以外の場合、最大圧縮レコード長は変更しません。

### NIRABN：ノーマルインデックスの開始 RABN

ファイルのノーマルインデックスエクステントの開始 RABN です。このパラメータ指定がなければ、ADAORD が開始 RABN を割り当てます。

### NIRELEASE：未使用ノーマルインデックスブロックの解放

NIRELEASE が指定されると、ファイルに属する未使用のノーマルインデックス (NI) が解放されます。NIRELEASE が指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた NI ブロックが ADAORD により割り当てられます。



**Note:** Adabas は、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステントサイズを計算します。

### NISIZE：ノーマルインデックスサイズ

ノーマルインデックスに対して割り当てるブロック数またはシリンダ数を指定します。指定可能な連続する最大の単一 RABN エリアを超えた値を指定することはできません。したがって、ブロック（ブロック数の後に "B" が続く）の指定は推奨されていません。

このパラメータが省略されると、ファイルのエクステントサイズ（ブロック数）は ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

このパラメータが省略され、INDEXCOMPRESSION パラメータが指定されると、ADAORD では、インデックスの圧縮率が分からないため、インデックスサイズ計算時にサイズ変更が考慮されません。そのため、必要量より大きくなったり、小さくて2次インデックスエクステントの割り当てが発生したりすることがあります。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーが発生すると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 034（ダンプ付き）またはユーザーアベンド 035（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**OVERWRITE**：既存ファイルの上書き

格納するファイルがデータベースにすでに存在すると、OVERWRITEパラメータが指定されていない限り、ADAORDはエラーメッセージを出力して終了します。

**SECURITY**：セキュリティファイルの格納

SECURITYは、DDFILEA/FILEA テープにあるセキュリティファイルを格納し、それをデータベースの新しいセキュリティファイルにします。新しいセキュリティファイルのファイル番号は、古いセキュリティファイルのものと同一である必要があります。

SECURITYを省略すると、FILEパラメータまたはALLFILESパラメータでセキュリティファイルが指定されていても、DDFILEA/FILEA テープのセキュリティファイルはデータベースに格納されません。

**TEST**：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**TRIGGER**：トリガファイルの格納

DDFILEA/FILEA テープにあるトリガファイルをデータベースの新しいトリガファイルとして格納するには、TRIGGERパラメータを指定します。新しいトリガファイルのファイル番号は、古いトリガファイルのものと同一である必要があります。

TRIGGERパラメータを指定しないと、FILEパラメータまたはALLFILESパラメータでトリガファイルが指定されていても、DDFILEA/FILEA テープのトリガファイルはデータベースに格納されません。

**UIRABN**：アッパーインデックスの開始 RABN

UIRABNは、ファイルのアッパーインデックスエクステンツの開始 RABN を指定します。このパラメータ指定がなければ、ADAORDが開始 RABN を割り当てます。

**UIRELEASE**：未使用アッパーインデックスブロックの解放

UIRELEASEが指定されると、ファイルに属する未使用のアッパーインデックス（UI）が解放されます。UIRELEASEが指定されない場合には、少なくともそのファイルがリオーダされる以前に割り当てられた UI ブロックが ADAORD により割り当てられます。



**Note:** Adabasは、あらゆる変更されたパディングファクタまたはファイルがリオーダされる前のブロックサイズ値を用いてファイルのエクステンツサイズを計算します。

### UISIZE：アッパーインデックスサイズ

UISIZE は、アッパーインデックスに対して割り当てるブロック数またはシリンダ数を指定します。ブロック単位で指定する場合には、値の後に "B" を指定する必要があります ("2000B" など)。

このパラメータが省略されると、ファイルのエクステントサイズ (ブロック数) は ASSOPFAC パディングファクタ内での全増減に比例して ADAORD によって計算されます。

このパラメータが省略され、INDEXCOMPRESSION パラメータが指定されると、ADAORD では、インデックスの圧縮率が分からないため、インデックスサイズ計算時にサイズ変更が考慮されません。そのため、必要量より大きくなったり、小さくて2次インデックスエクステントの割り当てが発生したりすることがあります。

## 例

---

### 例 1

```
ADAORD STORE FILE=14,OVERWRITE
```

RESTRUCTURE 機能または REORDB 機能のいずれかでアンロードされたファイル 14 を、既存のデータベースに格納します。ファイルがすでに存在する場合は、そのファイルを削除してから格納します。

### 例 2

```
ADAORD STORE FILE=1,OVERWRITE
ADAORD FILE=2,OVERWRITE
ADAORD FILE=3,OVERWRITE
```

ファイル 1、2、および 3 を既存のデータベースに書き込みます。古いファイル 1、2、および 3 は削除されます。

### 例 3

```
ADAORD STORE OVERWRITE,ALLFILES
ADAORD FILE=1,ACRABN=1000,NIRABN=2200
ADAORD FILE=2,MAXISN=500000
ADAORD
FILE=4,ASSOPFAC=5,DATAPFAC=20,DSSIZE=5B,DSRABN=1
```

RESTRUCTURE 機能でアンロードされたすべてのファイルを既存のデータベースに格納します。ファイル 1 のアドレスコンバータは RABN 1000 から開始します。ファイル 1 のノーマルインデックスは RABN 2200 から開始します。ファイル 2 の MAXISN は 500,000 にセットします。ファイル 4 に対して次の指定を行います。アソシエータブロックパディングファクタは 5% にセットし、データストレージブロックパディングファクタは 20% にセットし、RABN1 から開始して 5 つのシリンダに新しい DSSIZE を割り当てます。

入力データセットに含まれる他のすべてのファイルは、それぞれのデフォルト値でリストアされます。ファイルがすでに存在する場合は、そのファイルを削除してから格納します。

#### 例 4

```
ADAORD STORE ALLFILES,CHECKPOINT
ADAORD EXCLUDE=20,10
```

入力データセットにあるすべてのファイル（チェックポイントファイルを含む）を格納します。ただし、ファイル 10 および 20 は除外し、格納しません。

#### 例 5

```
ADAORD STORE ALLOCATION=NOFORCE
ADAORD FILE=10
ADAORD DSRABN=12345
```

ファイル 10 をデータベースに格納します。RABN 12345 から開始してデータストレージを割り当てます。別のファイルで占有されているためにこの割り当てができない場合、ADAORD はデータベースの他のデータストレージで割り当てようとしています。



# 130

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	672
▪ z/OS .....	677
▪ z/VM .....	680
▪ VSE .....	683

ここでは、BS2000、z/OS、z/VM、およびVSEシステムにおけるADAORDの実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

 **Note:** RESTRUCTURExxまたはSTORE機能用のオプションのRecovery Aid (ADARAI)を使用して実行している場合は、中間データセットもすべてジョブ制御にカタログする必要があります。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
中間ストレージ	DDFILEA	テープ/ディスク	
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI オプションの使用時に必要
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAORD パラメータ	SYS/DTA/DDKARTE		
ADARUN メッセージ	SYSOUT/DDPRINT		メッセージおよびコード
ADAORD メッセージ	SYSLST/DDDRUCK		メッセージおよびコード

### ADAORD JCL の例 (BS2000)

ファイルデータストレージのリオーダ、ファイルのリオーダ、データのリオーダ、データベースのリオーダ

SDF フォーマットの場合：

```

/.ADAORD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A O R D REORDER FILE DATA, REORDER FILE, REORDER DATA
/REMARK * REORDER DATABASE
/REMARK *
/DELETE-FILE ADAyyyyy.FILEA
/SET-JOB-STEP
/CREATE-FILE ADAyyyyy.FILEA,PUB(SPACE=(4800,480))
/SET-JOB-STEP
/ASS-SYSLST L.ORD.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
    
```

```

/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAYyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDFILEA,ADAYyyyy.FILEA
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAORD,DB=yyyyyy,IDENTNAME=ADABAS5B
ADAORD REORDATA FILE=1,DSSIZE=80,DATAPFAC=30
/LOGOFF SYS-OUTPUT=DEL

```

### ISP フォーマットの場合：

```

/.ADAORD LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A O R D REORDER FILE DATA, REORDER FILE, REORDER DATA
/REMARK * REORDER DATABASE
/REMARK *
/SYSFILE SYSLST=L.ORD.DATA
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyy.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAYyyyy.WORK,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAYyyyy.FILEA,LINK=DDFILEA,SPACE=(4800,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAORD,DB=yyyyyy,IDENTNAME=ADABAS5B
ADAORD REORDATA FILE=1,DSSIZE=80,DATAPFAC=30
/LOGOFF NOSPOOL

```

### アソシエータのリオーダ

#### SDF フォーマットの場合：

```

/.ADAORD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A O R D REORDER FILE ASSO, REORDER ASSO
/REMARK *
/DELETE-FILE ADAYyyyy.FILEA
/SET-JOB-STEP
/CREATE-FILE ADAYyyyy.FILEA,PUB(SPACE=(4800,480))
/SET-JOB-STEP
/ASS-SYSLST L.ORD.REOR
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDFILEA,ADAYyyyy.FILEA
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAORD,DB=yyyyyy,IDENTNAME=ADABAS5B

```

```
ADAORD REORFASSO
ADAORD FILE=1,MAXISN=20000,NISIZE=300B
ADAORD FILE=3,NISIZE=400B,ASSOPFAC=2
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADAORD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A O R D REORDER FILE ASSO, REORDER ASSO
/REMARK *
/SYSFILE SYSLST=L.ORD.REOR
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyyy.FILEA ,LINK=DDFILEA ,SPACE=(4800,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAORD,DB=yyyyyy, IDTNAME=ADABAS5B
ADAORD REORFASSO
ADAORD FILE=1,MAXISN=20000,NISIZE=300B
ADAORD FILE=3,NISIZE=400B,ASSOPFAC=2
/LOGOFF NOSPOOL
```

### 再構築

#### SDF フォーマットの場合：

```
/.ADAORD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A O R D RESTRUCTURE
/REMARK *
/DELETE-FILE ADAyyyyyy.FILEA
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.FILEA,PUB(SPACE=(4800,480))
/SET-JOB-STEP
/ASS-SYSLST L.ORD.REST
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDRLOGR1,ADAyyyyyy.RLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDFILEA,ADAyyyyyy.FILEA
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAORD,DB=yyyyyy, IDTNAME=ADABAS5B
ADAORD RESTRUCTUREF
```

```
ADAORD FILE=1,DATADEV=dddd
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADAORD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A O R D RESTRUCTURE
/REMARK *
/SYSFILE SYSLST=L.ORD.REST
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.RLOGR1 ,LINK=DDRLOGR1,SHARUPD=YES
/FILE ADAyyyyyy.FILEA ,LINK=DDFILEA ,SPACE=(4800,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAORD,DB=yyyyyy, IDTNAME=ADABAS5B
ADAORD RESTRUCTUREF
ADAORD FILE=1,DATADEV=dddd
/LOGOFF NOSPOOL
```

### 格納

#### SDF フォーマットのテープの例：

```
/.ADAORD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A O R D STORE
/REMARK *
/DELETE-FILE ADAyyyyyy.FILEA
/SET-JOB-STEP
/REMARK Here, a tape has already been initialized in TSOS by
/REMARK /START-INIT
/REMARK INIT T-C4,VSN=ADA001,UNIT=M0 or M1
/REMARK END
/CREATE-FILE ADAyyyyyy.FILEA,SUP=TAPE (DEVICE=TAPE-C4,VOL=ADA001)
/SET-JOB-STEP
/ASS-SYSLST L.ORD.STOR
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAyyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDRLOGR1,ADAyyyyyy.RLOGR1,SHARE-UPD=YES
/REMARK Do not set the BUFFER-SIZE option for this file
/SET-FILE-LINK DDFILEA,ADAyyyyyy.FILEA
/REMARK
/REMARK Here, using TAPEREL=NO, the tape will remain
/REMARK mounted This must be done from the console.
/REMARK if the tape is to be dismounted, omit this
```

```

/REMARK parameter
/REMARK
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAORD,DB=yyyyy,IDTNAME=ADABAS5B,TAPEREL=NO
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAFAC=30,DSRABN=1234
ADAORD MAXISN=200000
/LOGOFF SYS-OUTPUT=DEL

```

**ISP フォーマットのテープの例：**

```

/.ADAORD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A O R D STORE
/REMARK *
/SYSFILE SYSLST=L.ORD.STOR
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyy.RLOGR1 ,LINK=DDRLOGR1,SHARUPD=YES
/REMARK Here, a tape has already been initialized in TSOS by
/REMARK /START-INIT
/REMARK INIT T-C4,VSN=ADA001,UNIT=M0 or M1
/REMARK END
/REMARK Do not set the BLKSIZE option for this file
/FILE ADAyyyyy.FILEA ,LINK=DDFILEA ,DEVICE=TAPE-C4,VOLUME=ADA001,LABEL=STD
/REMARK
/REMARK Here, using TAPEREL=NO, the tape will remain
/REMARK mounted This must be done from the console.
/REMARK if the tape is to be dismounted, omit this
/REMARK parameter
/REMARK
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAORD,DB=yyyyy,IDTNAME=ADABAS5B,TAPEREL=NO
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAFAC=30,DSRABN=1234
ADAORD MAXISN=200000
/LOGOFF NOSPOOL

```

**SDF フォーマットのテープ以外の例：**

```

/.ADAORD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A O R D STORE
/REMARK *
/DELETE-FILE ADAyyyyy.FILEA
/SET-JOB-STEP
/CREATE-FILE ADAyyyyy.FILEA,PUB(SPACE=(4800,480))

```

```

/SET-JOB-STEP
/ASS-SYSLST L.ORD.STOR
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADA vrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDRLOGR1,ADAYyyyy.RLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDFILEA,ADAYyyyy.FILEA
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAORD,DB=yyyyyy,IDENTNAME=ADABAS5B
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30,DSRABN=1234
ADAORD MAXISN=200000
/LOGOFF SYS-OUTPUT=DEL

```

ISP フォーマットのテープ以外の例：

```

/.ADAORD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A O R D STORE
/REMARK *
/SYSFILE SYSLST=L.ORD.STOR
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyy.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAYyyyy.RLOGR1,LINK=DDRLOGR1,SHARUPD=YES
/FILE ADAYyyyy.FILEA,LINK=DDFILEA,SPACE=(4800,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAORD,DB=yyyyyy,IDENTNAME=ADABAS5B
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30,DSRABN=1234
ADAORD MAXISN=200000
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	REORASSO または REORFASSO には使用しない
中間ストレージ	DDFILEA	テープ/ディスク	
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI オプションの使用時に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAORD パラメータ	DDKARTE	リーダー	

データセット	DD 名	ストレージ	追加情報
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAORD メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADAORD JCL の例 (z/OS)

ファイルアソシエータのリオーダ

```
//ADAORDA JOB
//*
//* ADAORD: REORDER FILE ASSO,
//* REORDER ASSO
//*
//ORD EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDFILEA DD DSN=&&DDFILEA,DISP=(,PASS), <=== INTERMEDIATE
// UNIT=SYSDA,VOL=SER=vvvvvvv,SPACE=(CYL,NN) STORAGE
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAORD REORFASSO
ADAORD FILE=1,ASSOPFAC=15,MAXISN=10000
/*
```

この例については、MVSJOBS データセットの ADAORDA を参照してください。

ファイルデータストレージのリオーダ、ファイルのリオーダ、データのリオーダ、データベースのリオーダ

```
//ADAORDD JOB
//*
//* ADAORD: REORDER DATA STORAGE
//*
//ORD EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDFILEA DD DSN=DDFILEA, <=== INTERMEDIATE
// UNIT=TAPE,VOL=SER=vvvvvvv,DISP=(,PASS) FILE
//DDDRUCK DD SYSOUT=X
```

```
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X

//DDCARD DD *
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAORD REORDATA
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30
/*
```

この例については、MVSJOBS データセットの ADAORDD を参照してください。

## 再構築

```
//ADAORDR JOB
/*
/* ADAORD: RESTRUCTURE
/*
//ORD EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <== DATA
//DDFILEA DD DSN=FILEA, <== INTERMEDIATE
// UNIT=TAPE,VOL=SER=vvvvvv,DISP=(,KEEP) <== FILE
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAORD RESTRUCTUREF
ADAORD FILE=1,DATADEV=eeee
/*
```

この例については、MVSJOBS データセットの ADAORDR を参照してください。

## 格納

```
//ADAORDS JOB
/*
/* ADAORD: STORE INTO A DIFFERENT DATABASE
/* AFTER ADAORD RESTRUCTURE
/*
//ORD EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
```

```

/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDFILEA DD DSN=FILE1, <=== INTERMEDIATE
// UNIT=TAPE,VOL=SER=vvvvvvv,DISP=OLD STORAGE
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy <=== DBID
/*
//DDKARTE DD *
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30,DSRABN=1234,MAXISN=200000
/*

```

この例については、MVSJOBS データセットの ADAORDS を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
中間ストレージ	DDFILEA	テープ/ディスク	
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI オプションの使用時に必要
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAORD パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAORD メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

### ADAORD JCL の例 (z/VM)

#### アソシエータのリオーダ

```

DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDFILEA,DSN=ADAORD.FILEA,MODE=A
DATADEF DDPRINT,DSN=ADAORD.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAORD.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNORD.CONTROL,MODE=A

```

```
DATADEF DDKARTE , DSN=REORASSO . CONTROL , MODE=A  
ADARUN
```

### RUNORD CONTROL A1 の内容

```
ADARUN PROG=ADAORD , DEVICE=dddd , DB=yyyyyy
```

### REORASSO CONTROL A1 の内容

```
ADAORD REORFASSO  
ADAORD FILE=1 , MAXISN=20000 , ASSOPFAC=20  
ADAORD FILE=3 , NISIZE=400B , ASSOPFAC=20
```

### データストレージのリオーダ

```
DATADEF DDASSOR1 , DSN=ADABASVv . ASSO , VOL=ASSOV1  
DATADEF DDDATAR1 , DSN=ADABASVv . DATA , VOL=DATAV1  
DATADEF DDFILEA , DSN=ADAORD . FILEA , MODE=A  
DATADEF DDPRINT , DSN=ADAORD . DDPRINT , MODE=A  
DATADEF DUMP , DUMMY  
DATADEF DDDRUCK , DSN=ADAORD . DDDRUCK , MODE=A  
DATADEF DDCARD , DSN=RUNORD . CONTROL , MODE=A  
DATADEF DDKARTE , DSN=REORDATA . CONTROL , MODE=A  
ADARUN
```

### RUNORD CONTROL A1 の内容

```
ADARUN PROG=ADAORD , DEVICE=dddd , DB=yyyyyy
```

### REORDATA CONTROL A1 の内容

```
ADAORD REORDATA
```

```
ADAORD      FILE=1 ,DSSIZE=80 ,DATAPFAC=30
```

### 再構築

```
DATADDEF DDASSOR1 ,DSN=ADABASVv . ASSO ,VOL=ASSOV1
DATADDEF DDFILEA ,DSN=ADAORD . FILEA ,MODE=A
DATADDEF DDRLOGR1 ,DSN=ADABVv . RLOGR1 ,VOL=RLOGV1
DATADDEF DDPRINT ,DSN=ADAORD . DDPRINT ,MODE=A
DATADDEF DUMP ,DUMMY
DATADDEF DDDRUCK ,DSN=ADAORD . DDDRUCK ,MODE=A
DATADDEF DDCARD ,DSN=RUNORD . CONTROL ,MODE=A
DATADDEF DDKARTE ,DSN=RESTRUCT . CONTROL ,MODE=A
ADARUN
```

### RUNORD CONTROL A1 の内容

```
ADARUN PROG=ADAORD ,DEVICE=dddd ,DB=yyyyy
```

### RESTRUCT CONTROL A1 の内容

```
ADAORD RESTRUCTREF
ADAORD      FILE=1 ,DATADDEV=eeee
```

### 格納

```
DATADDEF DDASSOR1 ,DSN=ADABASVv . ASSO ,VOL=ASSOV1
DATADDEF DDDATAR1 ,DSN=ADABASVv . DATA ,VOL=DATAV1
DATADDEF DDRLOGR1 ,DSN=ADABVv . RLOGR1 ,VOL=RLOGV1
DATADDEF DDFILEA ,DSN=ADAORD . FILEA ,MODE=A
DATADDEF DDPRINT ,DSN=ADAORD . DDPRINT ,MODE=A
DATADDEF DUMP ,DUMMY
DATADDEF DDDRUCK ,DSN=ADAORD . DDDRUCK ,MODE=A
DATADDEF DDCARD ,DSN=RUNORD . CONTROL ,MODE=A
DATADDEF DDKARTE ,DSN=ORDSTORE . CONTROL ,MODE=A
ADARUN
```

### RUNORD CONTROL A1 の内容

```
ADARUN PROG=ADAORD,DEVICE=dddd,DB=yyyyyy
```

## ORDSTORE CONTROL A1 の内容

```
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30,DSRABN=1234
ADAORD MAXISN=200000
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	
データストレージ	DATARn	ディスク	*	
中間ストレージ	FILEA	テープ ディスク	SYS010 *	
リカバリログ (RLOG)	RLOGR1	ディスク		ADARAI オプションの使用時に必要
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADAORD パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	メッセージおよびコード
ADAORD メッセージ		プリンタ	SYS009	メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

### ADAORD JCS の例 (VSE)

VSE プロシージャ (PROC) の内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

以下に例とメンバの対応表を示します。

例	メンバ
ファイルアソシエータのリオーダ	ADAORDA.X
ファイルデータストレージのリオーダ	ADAORDD.X
再構築	ADAORDR.X
ファイルの格納	ADAORDS.X

ファイルアソシエータのリオーダ

```

* $$ JOB JNM=ADAORDA,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAORDA
*      REORDER THE ASSOCIATOR.
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,DISK,VOL=DISK01,SHR
// DLBL FILEA,'ADABAS.Vvr.TEMP'
// EXTENT SYS010,DISK01,1,0,sssss,nnnnn
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAORD REORFASSO
ADAORD FILE=1,ASSOPFAC=15,MAXISN=10000
/*
/&
* $$ EOJ

```

ファイルデータストレージのリオーダ、ファイルのリオーダ、データのリオーダ、データベースのリオーダ

```

* $$ JOB JNM=ADAORDD,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAORDD
*      REORDER DATA STORAGE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,TAPE,D0
// PAUSE MOUNT SCRATCH TAPE ON TAPE cuu
// MTC REW,SYS010
// MTC WTM,SYS010,5
// MTC REW,SYS010
// TLBL FILEA,'ADABAS.Vvr.TEMP'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAORD REORDATA
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30
/*
/&
* $$ EOJ

```

再構築

```
* $$ JOB JNM=ADAORDR,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAORDR
*      RESTRUCTURE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,TAPE,D0
// PAUSE MOUNT SCRATCH TAPE ON TAPE cuu
// MTC REW,SYS010
// MTC WTM,SYS010,5
// MTC REW,SYS010
// TLBL FILEA,'ADABAS.Vvr.TEMP'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAORD RESTRUCTUREF
ADAORD FILE=1,DATADEV=eeee
/*
/&
* $$ EOJ
```

## ファイルの格納

```
* $$ JOB JNM=ADAORDS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAORDS
*      STORE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS010,TAPE,D0
// PAUSE MOUNT SCRATCH TAPE ON TAPE cuu
// MTC REW,SYS010
// MTC WTM,SYS010,5
// MTC REW,SYS010
// TLBL FILEA,'ADABAS.Vvr.TEMP'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAORD,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAORD STORE
ADAORD FILE=1,DSSIZE=80,DATAPFAC=30,DSRABN=1234,MAXISN=200000
/*
/&
* $$ EOJ
```



# 131 ADAPLP ユーティリティ：プロテクションログ ／ワークの出力

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ADAPLP の構文と例
- JCL/JCS の必要項目と指定例



# 132

## 機能概要

---

ADAPLP ユーティリティは、Adabas WORK データセットまたは Adabas データプロテクションログ内のデータプロテクションレコードを出力します。



# 133 ADAPLP の構文と例

---

■ オプションのパラメータとサブパラメータ .....	693
■ 例 .....	696

次の図に、次の内容を指定するための ADAPLP 構文を示します。シーケンシャル中間 (IPLOGPRI)、マルチ (PLOG..) または シーケンシャル (SPLOG..) プロテクションログ、または WORK データセットの出力

```

ADAPLP { IPLOGPRI | PLOGPRI | SPLOGPRI | WORKPRI }
    [
      TYPE = type
      FILE = file-number [ ISN = isn ]
      RABN = data-storage-rabn
    ]
    [ DEVICE = { multiple-PLOG-device-type | ADARUN-device } ]
    [ LAYOUT = { 1 | 2 | 3 } ]
    [ NOUSERABEND ]
    [ NUMBER = { 1 | n } ]
    [ PRINT ]
    [ SKIPRABN = { block-count | 0 } ]
    [ STOPRABN = block-count ]

```

ここで、*type* は次のいずれかです。

```

    [
      [
        ALL
        ASSO
        DATA
      ] [ FILE = file-number ] [ ISN = isn ]

      [
        ALL
        DATA
      ] [ RABN = data-storage-rabn ]

      [
        C1
        C5
        ET
        EEKZ
        SAVO
        VEKZ
      ]

      REPR [ FILE = file-number ]
    ]

```

IPLOGPRI 機能は PLOG マージプロセスから作られたシーケンシャル中間データセットを出力するために使用されます。ADAPLP/IPLOGPRI への入力、ADARESユーティリティで作成されたDDPLOG DDJCL ステートメントで指定されている MERGINT1/MERGINT2 データセットである必要があります。操作は SPLOGPRI 機能に似ています。

このchapterでは、次のトピックについて説明します。

## オプションのパラメータとサブパラメータ

### DEVICE：デバイスタイプ

DEVICE パラメータは、マルチプロテクションデータセットの出力が設定されたデバイスタイプを指定します。このパラメータは、デバイスタイプが標準の ADARUN デバイスと異なる場合にのみ必要です。

### FILE：データの出力元となるファイル

FILE パラメータは、指定された Adabas ファイルの情報を含むプロテクションレコードに限定して出力する場合に使用できます。

RABN パラメータまたは TYPE = C1、C5、ET、EEKZ、SAVO、VEKZ を指定した場合には、FILE パラメータを指定できません。RABN パラメータを使用する場合は、ISN を指定しません。

### ISN：データを出力する ISN

このパラメータは、指定された ISN で識別されるプロテクションレコードに限定して出力する場合に使用できます。RABN パラメータまたは TYPE = C1、C5、ET、EEKZ、SAVO、VEKZ を指定した場合には、ISN パラメータを指定できません。

### LAYOUT：出力形式

PRINT パラメータで要求されたプロテクションログレコードの出力形式を制御します。1 (デフォルト)、2、または 3 を指定します。

LAYOUT=1 (デフォルト)

EBCDIC		16進	オフセット	
↓		↓	↓	
*	I 1	SYN*	00250000000BC9059C4DF1CD05E2E8D5	0000
*5A	ADA003	*	F5C10204020307C1C4C1F0F0F3020206	0010
*GAOS1	/	*	C7C1F0E2F10062006100000003000440	0020

## LAYOUT=2/3

オフセット16進	EBCDIC
0000	0025000000BC9059C4DF1CD05E2E8D5 * I 1 SYN*
0010	F5C10204020307C1C4C1F0F0F3020206 *5A ADA003 *
0020	C7C1F0E2F10062006100000003000440 *GAOS1 / *

LAYOUT=3 は LAYOUT=2 と同じ形式ですが、PLOG レコードタイプごとの説明も含まれます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**NUMBER**：マルチプロテクションログデータセット番号

NUMBER パラメータは、出力対象となるいずれかのマルチ（2～8）プロテクションログ（PLOG）データセットを指定します。"2"を指定すると DD/PLOGR2 データセットが選択され、"3"を指定すると DD/PLOGR3 データセットが選択されます。デフォルトの"1"を指定すると、DD/PLOGR1 が選択されます。

**PRINT**：データプロテクションレコード全体の出力

PRINT パラメータは、データプロテクションログレコード全体を出力する場合に指定します。このパラメータを省略すると、プロテクションログレコードのヘッダーのみが出力されます。

**RABN**：指定されたデータストレージブロックの更新のみの出力

RABN パラメータは、エラーが発生した可能性のある特定のデータストレージブロックに対して行われたすべての更新を追跡するために使用できます。

指定のデータストレージブロックで削除、更新、追加が行われたデータストレージレコードの処理前後のイメージを記述するプロテクションレコードに出力が制限されます。

RABN パラメータは、TYPE=ALL（デフォルト）または TYPE=DATA 機能に対して指定できます。つまり、データストレージプロテクションレコードを選択する機能に対して指定できます。

**SKIPRABN**：スキップするブロック数

SKIPRABN パラメータは、出力の開始前にスキップするブロック数を指定します。出力するブロック数のカウント（STOPRABN パラメータを参照）は、このパラメータで指定されたブロック数がスキップされた後で開始します。

**STOPRABN**：出力するブロック数

STOPRABN パラメータは、出力するブロック数を制限します。このパラメータを省略すると、プロテクションログの終わりまでの全ブロックが出力されます。STOPRABN パラメータで指定された RABN カウントに加え、RABN1 も出力されます。したがって、出力される RABN の合計数は、STOPRABN で指定した値より常に 1 つ多くなります。

**TYPE**：出力されるレコードのタイプ

TYPE パラメータは、出力用に選択するプロテクションレコードのタイプを指定します。次の値を指定できます。

ALL	全プロテクションレコード（デフォルト）
ASSO	アソシエータプロテクションレコード
DATA	データストレージプロテクションレコード
C1	Adabas C1 コマンドによる発生レコード
C5	Adabas C5 コマンドによる発生レコード
EEKZ	ニュークリアスバッファフラッシュの終了時に書かれたレコード
ET	Adabas ET コマンドによる発生レコード
REPR	自動再スタートがインデックスの修復に使用する WORK データセットレコード
SAVO	オンライン SAVE データベース/ファイルレコード
VEKZ	更新コマンドの終了時に書かれたレコード



**Note:** FILE、ISN、または RABN パラメータを指定すると、プロテクションレコード数がさらに削減されます。

**例****例 1**

```
ADAPLP WORKPRI PRINT,TYPE=ALL,STOPRABN=40
```

Adabas ワークから 41 件のデータプロテクションレコードを出力します。

**例 2**

```
ADAPLP WORKPRI PRINT,TYPE=ASSO,STOPRABN=10
```

Adabas ワークから 11 件のアソシエータデータプロテクションブロックを出力します。

**例 3**

```
ADAPLP PLOGPRI PRINT
```

---

いずれかのマルチプロテクションログデータセットに含まれるすべてのデータプロテクションブロックを出力します。



# 134 JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	700
▪ z/OS .....	704
▪ z/VM .....	707
▪ VSE .....	709

ここでは、BS2000、z/OS、z/VM、および VSE システムにおける ADAPLP の実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	WORKPRI に必要。
ワーク	DDWORKR1 DDWORKR4	ディスク	WORKPRI に必要。
シーケンシャルプロテクションログ	DDPLOG	テープ/ディスク	SPLOGPRI または IPLOGPRI に必要。
マルチプロテクションログ	DDPLOGR1	ディスク	NUMBER=1 のとき PLOGPRI に必要 (デフォルト)。
マルチプロテクションログ	DDPLOGRn	ディスク	NUMBER=n のとき PLOGPRI に必要。
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAPLP パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT DDPRINT		メッセージおよびコード
ADAPLP メッセージ	SYSLST DDDRUCK		メッセージおよびコード

### ADAPLP JCL の例 (BS2000)

#### シーケンシャルプロテクションログの出力

SDF フォーマットの場合：

```

/.ADAPLP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT SEQUENTIAL PROTECTION LOG
/REMARK *
/ASS-SYSLST L.PL.P.SPLO
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDPLOG,ADAYyyyy.PLOG
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAPLP,DB=yyyyy,IDTNAME=ADABAS5B
    
```

```
ADAPLP SPLOGPRI PRINT
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADAPLP LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT SEQUENTIAL PROTECTION LOG
/REMARK *
/SYSFILE SYSLST=L.PLP.SPLO
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.PLOG,LINK=DDPLOG
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAPLP,DB=yyyyy, IDTNAME=ADABAS5B
ADAPLP SPLOGPRI PRINT
/LOGOFF NOSPOOL
```

シーケンシャル中間プロテクションログの出力

SDF フォーマットの場合：

```
/.ADAPLP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT SEQUENTIAL PROTECTION LOG
/REMARK *
/ASS-SYSLST L.PLP.SPLO
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDPLOG,ADAYyyyy.PLOG
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAPLP,DB=yyyyy, IDTNAME=ADABAS5B
ADAPLP IPLOGPRI PRINT
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADAPLP LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT SEQUENTIAL PROTECTION LOG
/REMARK *
/SYSFILE SYSLST=L.PLP.SPLO
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.PLOG,LINK=DDPLOG
```

```
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAPLP,DB=yyyyyy,IDTNAME=ADABAS5B
ADAPLP IPLOGPRI PRINT
/LOGOFF NOSPOOL
```

## 1つのマルチプロテクションログデータセットの出力

SDF フォーマットの場合：

```
/.ADAPLP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT MULTIPLE PROTECTION LOG
/REMARK *
/ASS-SYSLST L.PLP.PLOG
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDPLOGR1,ADAYyyyyy.PLOG
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAPLP,DB=yyyyyy,IDTNAME=ADABAS5B
ADAPLP PLOGPRI PRINT
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADAPLP LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT MULTIPLE PROTECTION LOG
/REMARK *
/SYSFILE SYSLST=L.PLP.PLOG
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyyy.PLOG,LINK=DDPLOGR1
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAPLP,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADAPLP PLOGPRI PRINT
/LOGOFF NOSPOOL
```

## ワークの出力

**SDF** フォーマットの場合：

```
/.ADAPLP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT ADABAS WORK
/REMARK *
/ASS-SYSLST L.PLP.WORK
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADyyyyyy.WORK,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAPLP,DB=yyyyyy,IDTNAME=ADABAS5B
ADAPLP WORKPRI PRINT,TYPE=ASSO
/LOGOFF SYS-OUTPUT=DEL
```

**ISP** フォーマットの場合：

```
/.ADAPLP LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A P L P PRINT ADABAS WORK
/REMARK *
/SYSFILE SYSLST=L.PLP.WORK
/FILE ADA.MOD,LINK=DDLIB
/FILE ADyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADyyyyyy.WORK,LINK=DDWORKR1,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAPLP,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADAPLP WORKPRI PRINT,TYPE=ASSO
/LOGOFF NOSPOOL
```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	WORKPRI に必要。
ワーク	DDWORKR1 DDWORKR4	ディスク	WORKPRI に必要。
シーケンシャルプロテクションログ	DDPLOG	テープ/ディスク	SPLOGPRI または IPLOGPRI に必要。
マルチプロテクションログ	DDPLOGR1	ディスク	NUMBER=1 のとき PLOGPRI に必要 (デフォルト)。
マルチプロテクションログ	DDPLOGRn	ディスク	NUMBER=n のとき PLOGPRI に必要。
ADAPLP メッセージ	DDDRUCK	プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAPLP パラメータ	DDKARTE	リーダー	

### ADAPLP JCL の例 (z/OS)

次のジョブ例については、MVSJOBS データセットを参照してください。

ジョブメンバ	説明
ADAPLP	プロテクションログの出力 (マルチデータセット PLOG から)
ADAPLPS	プロテクションログの出力 (シーケンシャル PLOG から)
ADAPLPW	Adabas ワークの出力

次に、これらのジョブを示します。

#### 1つのマルチプロテクションログデータセットの出力

```

//ADAPLP      JOB
//*
//*    ADAPLP: PROTECTION LOG PRINT (FROM MULTIPLE PLOG)
//*
//PLP         EXEC PGM=ADARUN
//STEPLIB     DD    DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDPLOGR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1 <=== PLOG
DATASET
//DDDRUCK    DD    SYSOUT=X
//DDPRINT    DD    SYSOUT=X
//SYSUDUMP   DD    SYSOUT=X
//DDCARD     DD    *
ADARUN PROG=ADAPLP,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE    DD    *
ADAPLP PLOGPRI
/*

```

### シーケンシャルプロテクションログの出力

```

//ADAPLPS     JOB
//*
//*    ADAPLP: PROTECTION LOG PRINT (FROM SEQUENTIAL PLOG)
//*
//PLP         EXEC PGM=ADARUN
//STEPLIB     DD    DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1   DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDPLOG     DD    DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOG, <=== PLOG DATASET
//
//                                     UNIT=TAPE,VOL=SER=PLOG5
//DDDRUCK    DD    SYSOUT=X
//DDPRINT    DD    SYSOUT=X
//SYSUDUMP   DD    SYSOUT=X

//DDCARD     DD    *
ADARUN PROG=ADAPLP,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE    DD    *
ADAPLP SPLOGPRI
/*

```

### シーケンシャル中間プロテクションログの出力

```
//ADAPLPS      JOB
//*
//*      ADAPLP: PROTECTION LOG PRINT (FROM SEQUENTIAL PLOG)
//*
//PLP          EXEC PGM=ADARUN
//STEPLIB      DD      DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1     DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1     DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1     DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDPLOG       DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOG,    <=== PLOG DATASET
//
//              UNIT=TAPE,VOL=SER=PLOG5
//DDDRUCK      DD      SYSOUT=X
//DDPRINT      DD      SYSOUT=X
//SYSUDUMP     DD      SYSOUT=X

//DDCARD       DD      *
ADARUN PROG=ADAPLP,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE      DD      *
ADAPLP IPLOGPRI
/*
```

## Adabas ワークの出力

```
//ADAPLPW      JOB
//*
//*      ADAPLP: PRINT ADABAS WORK
//*
//PLP          EXEC PGM=ADARUN
//STEPLIB      DD      DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1     DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1     DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1     DD      DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDDRUCK      DD      SYSOUT=X
//DDPRINT      DD      SYSOUT=X
//SYSUDUMP     DD      SYSOUT=X
//DDCARD       DD      *
ADARUN PROG=ADAPLP,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE      DD      *
```

```
ADAPLP WORKPRI PRINT
/*
```

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	WORKPRI に必要。
ワーク	DDWORKR1 DDWORKR4	ディスク	WORKPRI に必要。
シーケンシャルプロテクションログ	DDPLOG	テープ/ディスク	SPLOGPRI または IPLOGPRI に必要。
マルチプロテクションログ	DDPLOGR1	ディスク	NUMBER=1 のとき PLOGPRI に必要 (デフォルト)。
マルチプロテクションログ	DDPLOGRn	ディスク	NUMBER=n のとき PLOGPRI に必要。
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAPLP パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAPLP メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

### ADAPLP の例 (z/VM)

#### シーケンシャルプロテクションログの出力

```
DATADEF DDPLOG,DSN=ADABASV.V.SIBA,MODE=A
DATADEF DDPRINT,DSN=ADAPLP.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAPLP.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNPLP.CONTROL,MODE=A
DATADEF DDKARTE,DSN=PLPSPLOG.CONTROL,MODE=A
ADARUN
```

#### RUNPLP CONTROL A1 の内容

```
ADARUN  PROG=ADAPLP ,DEVICE=dddd ,DB=yyyyy
```

### PLPSPLOG CONTROL A1 の内容

```
ADAPLP  SPLOGPRI  PRINT
```

### シーケンシャル中間プロテクションログの出力

```
DATADEF  DDPLOG ,DSN=ADABASV . SIBA ,MODE=A
DATADEF  DDPRINT ,DSN=ADAPLP . DDPRINT ,MODE=A
DATADEF  DUMP ,DUMMY
DATADEF  DDDRUCK ,DSN=ADAPLP . DDDRUCK ,MODE=A
DATADEF  DDCARD ,DSN=RUNPLP . CONTROL ,MODE=A
DATADEF  DDKARTE ,DSN=PLPSPLOG . CONTROL ,MODE=A
ADARUN
```

### RUNPLP CONTROL A1 の内容

```
ADARUN  PROG=ADAPLP ,DEVICE=dddd ,DB=yyyyy
```

### PLPSPLOG CONTROL A1 の内容

```
ADAPLP  IPLOGPRI  PRINT
```

### 1つのマルチプロテクションログデータセットの出力

```
DATADEF  DDPLOG ,DSN=ADABASV . PLOG2 ,VOL=PLOGV2
DATADEF  DDPRINT ,DSN=ADAPLP . DDPRINT ,MODE=A
DATADEF  DUMP ,DUMMY
DATADEF  DDPLPLOGR1 ,DSN=ADAPLP . DDPLOG ,MODE=A
DATADEF  DDDRUCK ,DSN=ADAPLP . DDDRUCK ,MODE=A
DATADEF  DDCARD ,DSN=RUNPLP . CONTROL ,MODE=A
DATADEF  DDKARTE ,DSN=PLPDPLOG . CONTROL ,MODE=A
ADARUN
```

### RUNPLP CONTROL A1 の内容

```
ADARUN  PROG=ADAPLP,DEVICE=dddd,DB=yyyyy
```

## PLPDPLOG CONTROL A1 の内容

```
ADAPLP  PLOGPRI  PRINT
```

## ワークの出力

```
DATADEF  DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF  DDWORKR1,DSN=ADABASVv.WORK,VOL=WORKV1
DATADEF  DDPRINT,DSN=ADAPLP.DDPRINT,MODE=A
DATADEF  DUMP,DUMMY
DATADEF  DDDRUCK,DSN=ADAPLP.DDDRUCK,MODE=A
DATADEF  DDCARD,DSN=RUNPLP.CONTROL,MODE=A
DATADEF  DDKARTE,DSN=PLPWORK.CONTROL,MODE=A
ADARUN
```

## RUNPLP CONTROL A1 の内容

```
ADARUN  PROG=ADAPLP,DEVICE=dddd,DB=yyyyy
```

## PLPWORK CONTROL A1 の内容

```
ADAPLP  WORKPRI  PRINT,TYPE=ASSO
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	*	WORKPRI に必要。
ワーク	WORKR1	ディスク	*	WORKPRI に必要。
シーケンシャルプロテクションログ	PLOG	テープ ディスク	SYS014 *	SPLOGPRI または IPLOGPRI に必要。
マルチプロテクションログ	PLOGR1	ディスク	*	NUMBER=1 のとき PLOGPRI に必要（デフォルト）。
マルチプロテクションログ	PLOGRn	ディスク	*	NUMBER=n のとき PLOGPRI に必要。
ADAPLP レポート		プリンタ	SYS009	

ファイル	記号名	ストレージ	論理ユニット	追加情報
ADARUN メッセージ		プリンタ	SYSLST	
ADARUN パラメータ	SYSRDR CARD	リーダー/テープ/ ディスク		オペレーションマニュアル
ADAPLP パラメータ	SYSIPT	リーダー		

\* すべてのプログラマ論理ユニットを使用できます。

### ADAPLP JCS の例 (VSE)

VSE プロシージャ (PROC) の内容については、「[VSE用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

以下に例とメンバの対応表を示します。

例	メンバ
シーケンシャルプロテクションログの出力	ADAPLPS.X
マルチプロテクションログの出力	ADAPLP.X
Adabas ワークの出力	ADAPLPW.X

#### シーケンシャルプロテクションログの出力

```
* $$ JOB JNM=ADAPLPS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAPLPS
*      PROTECTION LOG PRINT (FROM SEQUENTIAL PLOG)
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS014,cuu
// PAUSE MOUNT LOAD INPUT FILE ON TAPE cuu
// TLBL PLOG,'EXAMPLE.DByyyyy.PLOG'
// MTC REW,SYS014
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAPLP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAPLP SPLOGPRI
/*
/&
* $$ EOJ
```

#### シーケンシャル中間プロテクションログの出力

```

* $$ JOB JNM=ADAPLPS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAPLPS
*     PROTECTION LOG PRINT (FROM SEQUENTIAL PLOG)
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS014, cuu
// PAUSE MOUNT LOAD INPUT FILE ON TAPE cuu
// TLBL PLOG, 'EXAMPLE.DByyyyy.PLOG'
// MTC REW, SYS014
// EXEC ADARUN, SIZE=ADARUN
ADARUN PROG=ADAPLP, MODE=MULTI, SVC=xxx, DEVICE=dddd, DBID=yyyyy
/*
ADAPLP IPLOGPRI
/*
/&
* $$ EOJ

```

### 1つのマルチプロテクションログデータセットの出力

```

* $$ JOB JNM=ADAPLP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAPLP
*     PROTECTION LOG PRINT (FROM MULTIPLE PLOG)
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN, SIZE=ADARUN
ADARUN PROG=ADAPLP, MODE=MULTI, SVC=xxx, DEVICE=dddd, DBID=yyyyy
/*
ADAPLP PLOGPRI
/*
/&
* $$ EOJ

```

### Adabas ワークの出力

```

* $$ JOB JNM=ADAPLPW,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAPLPW
*     PRINT ADABAS WORK
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN, SIZE=ADARUN
ADARUN PROG=ADAPLP, MODE=MULTI, SVC=xxx, DEVICE=dddd, DBID=yyyyy
/*
ADAPLP WORKPRI PRINT
/*

```

/&  
\* \$\$ EOJ

# 135 ADAPRIユーティリティ：選択したAdabasブロックの出力

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ADAPRI の構文と例
- JCL/JCS の必要項目と指定例



# 136

## 機能概要

---

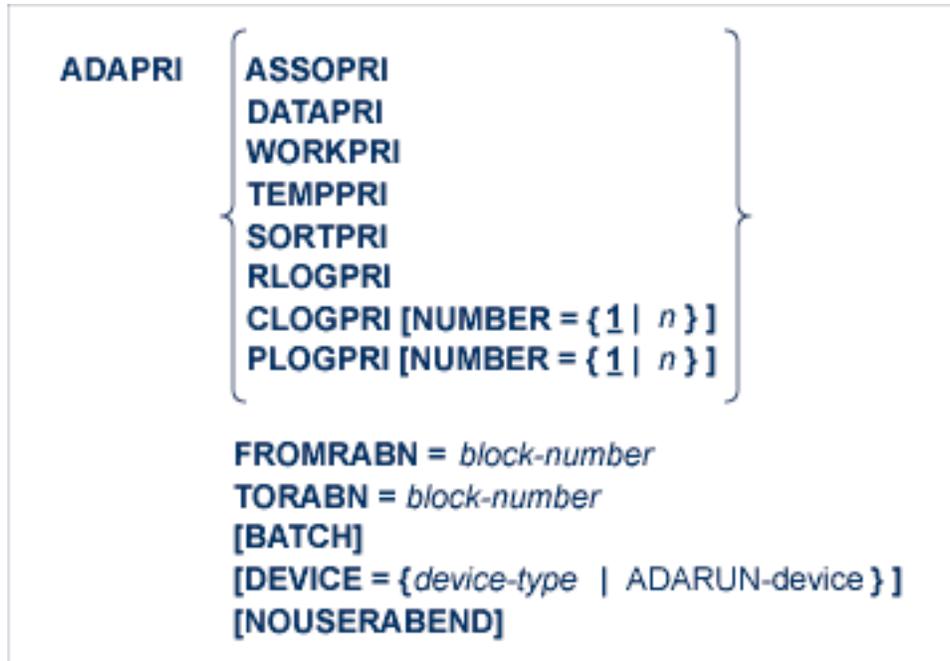
ADAPRI ユーティリティは、アソシエータ (ASSO..)、データストレージ (DATA..)、ワーク (WORK..)、中間 (TEMP..)、ソート (SORT..)、マルチデータセットコマンドログ (CLOG)、マルチデータセットプロテクションログ (PLOG)、またはリカバリログ (RLOG) データセットに含まれているブロック (ブロック範囲) の内容を実出力します。1 つ以上のデータセットを、ADAPRI の 1 回の実行で出力できます。



# 137 ADAPRI の構文と例

---

■ 基本パラメータ .....	718
■ オプションパラメータ .....	718
■ 例 .....	719



このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FROMRABN/TORABN：出力するブロック範囲

出力する RABN の開始番号および終了番号です。両方を指定する必要があり、デフォルトはありません。

出力は、FROMRABN パラメータで指定されたブロック番号で開始し、TORABN パラメータで指定されたブロック番号で終了します。範囲内の各ブロックは 16 進形式で出力されます。

## オプションパラメータ

### BATCH：出力形式

出力の行の長さを制御します。BATCH の指定がない場合、デフォルトの行サイズ 80 文字が使用されます。BATCH を指定した場合、出力行サイズは 120 文字になります。

### DEVICE：デバイスタイプ

出力するデータセットが含まれるデバイスタイプです。このパラメータは、デバイスタイプが ADARUN DEVICE パラメータで指定された標準のデバイスタイプと異なる場合に必要です。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**NUMBER**：コマンド/プロテクションログデータセット番号

出力するブロックが含まれる、マルチ（2～8）コマンドログ（CLOG）またはプロテクションログ（PLOG）データセットの番号です。NUMBERパラメータは、CLOGPRIまたはPLOGPRIを指定した場合にのみ指定できます。NUMBER=2を指定すると、DD/CLOGR2のブロックが出力されます。NUMBERを指定せずにCLOGPRIまたはPLOGPRI機能を指定すると、DD/CLOGR1（デフォルト）のブロックが出力されます。

## 例

**例 1**

```
ADAPRI ASSOPRI FROMRABN=1,TORABN=2
```

アソシエータ（ジェネラルコントロールブロックが含まれる）のブロック1および2を出力します。

**例 2**

```
ADAPRI DATAPRI FROMRABN=8000,TORABN=8120
```

データストレージのブロック 8000 から 8120 までを出力します。

**例 3**

```
ADAPRI WORKPRI FROMRABN=1,TORABN=100,BATCH
```

Adabas ワークのブロック 1 から 100 までを出力します。出力行サイズには 120 文字を使用します。

**例 4**

```
ADAPRI CLOGPRI FROMRABN=1,TORABN=80,BATCH,NUMBER=2
```

コマンドログデータセット DD/CLOGR2 のブロック 1 から 100 までを行幅 120 文字の形式で出力します。

**例 5**

```
ADAPRI DSIMPRI FROMRABN=1,TORABN=1
```

DSIM データセットのブロック 1 のみを出力します。Adabas Delta Save Facility がインストールされている場合は、DSIM データセットのみを使用します。

### 例 6

```
ADAPRI ASSOPRI FROMRABN=X'19D619',TORABN=X'19D619'
```

アソシエータのブロック 1693209 を出力します。FROMRABN および TORABN パラメータには 16 進値が使用されることに注意してください。

# 138

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	722
▪ z/OS .....	723
▪ z/VM .....	724
▪ VSE .....	725

ここでは、BS2000、z/OS、z/VM、および VSE システムにおける ADAPRI の実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
出力するブロックが含まれるデータセット	DDASSORn DDDATARn DDWORKR1 DDWORKR4 DDTEMPR1 DDSORTRn DDCLOGRn DDPLOGRn DDDSIMR1	ディスク ディスク ディスク ディスク ディスク ディスク ディスク ディスク ディスク	アソシエータ データストレージ* WORK* TEMP SORT マルチコマンド ログマルチプロテクションログDSIM データセット
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI 使用時に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAPRI パラメータ	SYSIPT/DDKARTE	リーダー	
ADARUN メッセージ	SYSOUT	プリンタ	メッセージおよびコード
ADAPRI メッセージ	SYSLST	プリンタ	メッセージおよびコード

\* データストレージまたはワークからブロックを出力する場合は、アソシエータのリンク名も存在する必要があります。

### ADAPRI JCL の例 (BS2000)

SDF フォーマットの場合：

```

/.ADAPRI LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A P R I ALL FUNCTIONS
/REMARK *
/ASS-SYSLST L.PRI
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAPRI,DB=yyyyy,IDENTNAME=ADABAS5B
    
```

```
ADAPRI DATAPRI FROMRABN=27,TORABN=34
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADAPRI LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A P R I ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.PRI
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAPRI,DB=yyyyy, IDTNAME=ADABAS5B
ADAPRI DATAPRI FROMRABN=27,TORABN=34
/LOGOFF NOSPOOL
```

## z/OS

データセット	DD 名	ストレージ	追加情報
出力するブロックが含まれるデータセット	DDASSORn DDDATARn DDWORKR1 DDWORKR4 DDTEMPR1 DDSORTR1 DDCLOGRn DDPLOGRn DDDSIMR1	ディスク ディスク ディスク ディスク ディスク ディスク ディスク ディスク ディスク	アソシエータ データストレージ* WORK* TEMP SORT マルチコマンドログ マルチ プロテクションログ DSIM データセット
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI 使用時に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAPRI パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAPRI メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

\* データストレージまたはワークからのブロックを出力するときは、アソシエータの DD ステートメントも存在する必要があります。

## ADAPRI JCL の例 (z/OS)

```
//ADAPRI      JOB
//*
//*      ADAPRI:
//*      MAINTENANCE PRINT
//*
//PRI          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDTEMPR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.TEMPR1 <=== TEMP
//DDSORTR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.SORTR1 <=== SORT
//DDPLOGR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1 <=== PLOG 1
//DDPLOGR2    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2 <=== PLOG 2
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
//SYSUDUMP    DD   SYSOUT=X
//DDCARD      DD   *
ADARUN PROG=ADAPRI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE     DD   *
ADAPRI ASSOPRI DEVICE=eeee,FROMRABN=1,TORABN=1
/*
```

この例については、MVSJOBS データセットの ADAPRI を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
出力するブロックが含まれるデータセット	DDASSORn DDDATARn DDWORKR1 DDWORKR4 DDTEMPR1 DDSORTR1 DDCLOGRn DDPLOGRn DDDSIMR1	ディスク	アソシエータデータストレージ* WORK* TEMP SORT マルチコマンドログ マルチプロテクションログ DSIM データセット
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI 使用時に必要
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAPRI パラメータ	DDKARTE	ディスク/端末/リーダー	

データセット	DD名	ストレージ	追加情報
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAPRI メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

\* データストレージまたはワークからのブロックを出力するときは、アソシエータのDDステートメントも存在する必要があります。

### ADAPRI の例 (z/VM)

```
DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1
DATADEF DDPRINT,DSN=ADAPRI.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADAPRI.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNPRI.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAPRI.CONTROL,MODE=A
ADARUN
```

### RUNPRI CONTROL A1 の内容

```
ADARUN PROG=ADAPRI,DEVICE=dddd,DB=yyyyyy
```

### ADAPRI CONTROL A1 の内容

```
ADAPRI DATAPRI FROMRABN=27,TORABN=34
```

## VSE

ファイル	記号名	ストレージ	追加情報
出力するブロックが含まれるファイル	ASSORn DATARn WORKR1 TEMPR1 SORTR1 CLOGRn PLOGRn SIMR1	ディスク ディスク ディスク ディスク ディスク ディスク ディスク ディスク	アソシエータ データストレージ * WORK* TEMP SORT マルチコマンドログ マルチプロテクションログ DSIM データセット
リカバリログ (RLOG)	RLOGR1	ディスク	ADARAI の使用時に必要
ADARUN メッセージ	SYSLST	プリンタ	メッセージおよびコード
ADAPRI メッセージ	SYS009	プリンタ	メッセージおよびコード

ファイル	記号名	ストレージ	追加情報
ADARUN パラメータ	SYSRDR CARD	リーダー/テープ/ディスク	オペレーションマニュアル
ADAPRI パラメータ	SYSIPT	リーダー	

\* データストレージまたはワークからブロックを出力する場合は、アソシエータのJCS ステートメントも存在する必要があります。

### ADAPRI JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

```
* $$ JOB JNM=ADAPRI,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAPRI
*      MAINTENANCE PRINT
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAPRI,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAPRI ASSOPRI DEVICE=eeee,FROMRABN=1,TORABN=1
/*
/&
* $$ EOJ
```

この例については、メンバ ADAPRI.X を参照してください。

# 139

## ADARAI ユーティリティ : Adabas Recovery Aid

---

ADARAI では、次の機能を使用できます。

- 機能概要
- **CHKDB** : データベースステータスのチェック
- **DISABLE** : リカバリロギングの非アクティブ化
- **LIST** : 現在の **RLOG** 世代の表示
- **PREPARE** : **RLOG** の初期化および開始
- **RECOVER** : リカバリジョブストリームの構築
- **REMOVE** : *Recovery Aid* の削除
- **JCL/JCS** の必要項目と指定例



# 140 機能概要

---

- 概念およびコンポーネント ..... 730

ADARAIユーティリティは、リカバリログ (RLOG) の準備、RLOGに含まれる情報のリスト、データベースを回復するためのジョブ制御ステートメントの作成、および ADARAI ロギングの無効化を行います。

Adabas セッションが異常終了したときのために、トランザクションリカバリが用意されています。Adabas 自動バックアウトルーチンは、どの Adabas セッションでも開始時に常に自動的に起動され、中断されたすべてのトランザクションの処理内容をデータベースから削除します。

『*Adabas オペレーションマニュアル*』にある再スタート/リカバリに関する情報を参照してください。

ただし、データベースのデータセット (ASSO、DATA、または WORK) が破壊された場合は、データベースをリストアおよび再生成して、失われたデータをリカバリする必要があります。

Adabas Recovery Aid ユーティリティ ADARAI を使用すると、データベースのリカバリを自動化し、最適化できます。Adabas 自動バックアウトルーチンは、データベースのリカバリに必要なすべての情報を記録および報告し、リカバリジョブストリーム (JCL/JCS) を作成します。このストリームに基づいて、最後に SAVE を実行した時点から障害およびエラーの時点までに実行したジョブが再実行されます。



**Note:** ジョブストリーム生成機能は、VSE または z/VM の環境では使用できません。

このchapterでは、次のトピックについて説明します。

## 概念およびコンポーネント

---

Adabas Recovery Aid は次の 2 つのコンポーネントで構成されています。

- データベースに対して関連イベントが発生したときに情報収集するインタフェース (ADARAC)
- 収集された情報の列挙、データベースまたはデータベース上のファイルを回復するジョブの生成、またはリカバリロギングの非アクティブ化を行うユーティリティ (ADARAI)

このsectionでは、次のトピックについて説明します。

- 収集インタフェース
- リカバリログ (RLOG)
- 世代：リカバリの単位

## ■ 非現行世代の保持

### 収集インターフェース

収集インターフェースは、ニュークリアスの停止／開始、ユーティリティの実行、Adabas Online Systemで生成されたイベントなど、発生した各イベントに関する情報を記録するために、ニュークリアスおよびすべてのユーティリティからコールされます。

### リカバリログ (RLOG)

イベント情報は、ユーティリティコンポーネントで使用するため、インターフェースによってリカバリログファイル (RLOG) にすべて記録されます。RLOGには、リカバリジョブコントロールの構築に必要なデータセット、ユーティリティパラメータ、およびプロテクションログに関する情報が格納されます。RLOG データセットは DD/RLOGR1 です。

ニュークリアスクラスタ環境では、すべてのニュークリアスで同じRLOGを使用します。RLOGの同時更新はロックによって制御されます。



#### Notes:

1. 実行が RLOG 上に記録されるユーティリティによって使用されるシーケンシャルデータセットは、ADALOD LOAD 操作に対する DD/EBAND 入力などのリカバリ操作に備えて、保持し、使用できる状態にしておく必要があります。
2. ADADBS ファイルの変更が、RLOG データセットに記録されるようになりました。
3. RLOG に記録される情報には、通常、リカバリに必要なもの以外も含まれています。また、これらの情報は、一定の期間にデータベースで発生したイベントの記録としても使用できます。

### 世代：リカバリの単位

情報は世代別に RLOG に格納されます。世代とは、リカバリに使用される論理単位のことです。

世代には、次の一連のオペレーション間のすべてのアクティビティが含まれます。

- ADASAV SAVE/RESTORE (データベース)
- RESTORE GCB
- SAVE DELTA/RESTORE DELTA (データベース)

最初の世代には最初のオペレーションが含まれ、次の世代にまで拡張されます (ただし次の世代は除外されます)。新しい世代は、前のオペレーションの後、データベースを完全に回復できる状態になった時点で開始します。

世代には、正常、制限付き、およびエラーの 3 種類があります。

- 世代の開始時にフルセーブを使用でき、かつ、アクティビティの記録中に異常なイベントが発生しなかった場合は、"正常"であることを示すラベルが世代に付けられます。
- ADARAI でデータベースを再構築するにはユーザーが介入する必要があるイベントがロギングサイクル中に発生すると、その世代には、"制限付き"であることを示すラベルが付けられます。ジョブは ADARAI によって生成されますが、実行するには、ユーザーの介入が必要です。例えば、WORK データセットサイズが減少した場合、減少した時点までのリカバリジョブを正しく実行できるようにするには、ユーザーが元のサイズで WORK データセットを作成する必要があります。
- 何らかの理由でロギングサイクル中にエラーが発生すると、"エラー"であることを示すラベルが世代に付けられます。ジョブは ADARAI によって生成されますが、実行するには、変更を加える必要があります。



**Note:** 世代が制限付きまたはエラーになった場合は、データベースのオンライン/オフラインセーブを実行して、なるべく早く新しい世代を開始することをお勧めします。Adabas Delta Save がインストールされている場合は、SAVE DELTA を実行すると新しい世代が開始されます。

### 非現行世代の保持

非現行世代には、データベースに影響を与えた操作履歴が含まれており、問題の解決または監査に使用できます。

リカバリジョブの RESTORE ステップが実行された後、データベースの回復に失敗した場合は、非現行世代へアクセスできることが重要になります。この時点では、回復対象の世代は現行世代になります。このとき、リカバリジョブを再構築することが必要になった場合、回復対象の世代は旧世代になります。

RLOG には、ADARAI PREPARE ステップの実行中に MINGENS パラメータで指定された世代数が保持されます。RLOG に保存された数値が MINGENS パラメータの指定値に達すると、ADARAI によって世代が再生されます。

新しい世代および保存済みの世代が使用可能な RLOG スペースを超えると、次のいずれかのイベントが発生します。

- MINGENS で指定された最小世代数を保持する場合は、最も古い世代が上書きされます。
- それ以外の場合は、RLOG コントロールブロックにフラグが設定され、RLOG はサービス休止になります。この場合、データは記録されなくなります。

# 141

## CHKDB : データベースステータスのチェック

---

```
ADARAI CHKDB [{ACTIVE | INACTIVE}]
```

ADARAI CHKDB 機能は、ニュークリアスに対してコマンドを発行しニュークリアスレスポンスコードをテストすることによって、指定されたニュークリアスのリカバリステータスをチェックします。

予期されるレスポンスコードがコマンドから表示されない場合、10 秒後に CHKDB によって別のコマンドが再発行されます。最大 10 個のコマンドが発行されます。10 回試行しても目的のニュークリアスステータス（アクティブ／非アクティブ）にならない場合、ADARAI はエラー 158 で終了します。

例

```
ADARAI CHKDB
```

リカバリニュークリアスのアクティブステータスをテストします。



# 142

## DISABLE：リカバリロギングの非アクティブ化

---

### ADARAI DISABLE

ADARAIDISABLE 機能は、RLOG テーブル（コントロールブロック）を非アクティブステータスに設定することによって、リカバリロギングを無効にします。



**Note:** ADARAI DISABLE は、データベースが非アクティブの状態で行う必要があります。

DISABLE を実行すると、情報が RLOG に記録されなくなり、現行世代が終了します。DISABLE 実行前の RLOG の内容は保持され、表示やリカバリに使用することができます。

新しい世代を開始すると、リカバリロギングを再び開始できます。「[世代：リカバリの単位](#)」を参照してください。

例

```
ADARAI DISABLE
```

Recovery Aid ロギングをすべて非アクティブ化します。



# 143 LIST：現在の RLOG 世代の表示

---

■ BS2000 における LIST の追加情報 .....	739
■ 構文 .....	934
■ オプションパラメータ .....	740
■ 例 .....	741



**Note:** Adabas バージョン 6 の RLOG は表示できません。バージョン 7 以上の RLOG のみがサポートされています。

ADARAI LIST 機能は、RLOG の内容をテーブル形式で表示するために使用します。

- 世代は番号順に表示されます。
- RLOG ブロック範囲が世代ごとに表示されます。
- 停止／開始日時が世代ごとに表示されます。

ユーティリティの実行およびニュークリアセッションの開始／停止エントリを含む次の情報が、RLOG のエントリごとに表示されます。

- RLOG エントリが書き込まれたイベント名
- 情報が RLOG に書き込まれた日時
- イベントに関連する PLOG 番号（存在する場合）
- 関連するチェックポイントが存在する PLOG ブロック（存在する場合）
- 記録されたイベントに関して DD/CARD および DD/KARTE ステートメントで指定されたパラメータ
- 記録されたイベント中に書き込みまたは読み込みが行われたファイルの詳細

ニュークリアクラスタ環境では、ニュークリアセッションの開始エントリに PLOG データセットも表示されます。クラスタニュークリア ID (NUCID) も表示されます。

例

```
*** 2001-08-21 11:37:08 NUCLEUS PLOG NUMBER=4
*** START NUCLEUS SESSION NUCID 40002

SYNC PLOG BLOCK NUMBER = 1
ACTIVE PLOG DATA SET NAMES: EXAMPLE.DBdddd.PLOGR21
                             EXAMPLE.DBdddd.PLOGR22
```

このchapterでは、次のトピックについて説明します。

## BS2000 における LIST の追加情報

BS2000 システムでは、LIST によって次の情報も表示されます。

- ファイルまたはファイル世代グループ (FGG) の特性および物理的な場所 (テープ、ディスクなど)
- ADARAI RECOVER 機能に必要な各データセットの有無および状態 (書き込み済みおよび消去済み、上書き済み、一時ファイルとして書き込み済み)
- 互換性のないコード化ファイル ID (CFID) の比較に関するエラーメッセージまたは警告メッセージ

一致しないコード化ファイル ID (CFID) は、BS2000 でデータセットが上書きされたことを示す場合が多いため、LIST は CFID を比較して、データセットが上書きされたかどうかを判断します。データセットが失われたか上書きされた場合、LIST は次の状態に対してエラーまたは警告を示します。

- データセットで次のことが行われると、"エラー" が示されます。
  - ■ ディスクに書き込まれた後、消去された
    - ディスクに書き込まれた後、上書きされた (CFID は一致しない) オリジナルとカタログの両方のエントリが報告されます。
    - 中間ディスクデータセットに書き込まれた
  - データセットに対して次のことが行われると、"警告" が示されます。
    - ■ テープに書き込まれ、カタログエントリが消去された
      - テープに書き込まれた後、上書きされた (CFID は一致しない) オリジナルとカタログの両方のエントリが報告されます。
      - 中間テープデータセットに書き込まれた

オペレーティングシステムによって異なる LIST 機能の結果例については、「出力例」のセクションを参照してください。

## 構文

```
ADARAI LIST [GENS = { NO | YES }]  
            [RELGEN = { gen-number | gen-number - gen-number }]  
            [RLOGDEV = { device | ADARUN-device }]
```

## オプションパラメータ

### GENS：世代の出力制御

GENS には、世代情報を出力するかどうかを指定します。GENS=NO は RLOG 制御情報のみを出力します。GENS=YES（デフォルト）は世代情報も出力します。

### RELGEN：相対リカバリ世代番号

RELGEN は、出力する相対世代番号（または世代番号範囲）を指定します。現行世代は、常に、相対世代 "0"（ゼロ）とカップリングされます。最後に完了した世代は相対世代 "1"、つまり 2 世代前となり、その前の世代は相対世代 "2" として指定されます。

#### 例

3 世代前から最後に完了した世代までの世代範囲を出力するには、RELGEN=3-1 と指定します。

指定された最初の世代番号が 2 番目の世代番号より小さいと、最初の世代番号と一致するように 2 番目の世代番号が減少されます。

#### 例

RELGEN=2-3 と指定すると、RELGEN=2-2 に変更されます。

RELGEN を指定しないと、すべての世代が出力されます。

指定された世代は、現在 RLOG に存在している必要があります。ただし、表示される各世代には、相対番号の代わりに、1（RLOG 操作開始後の最初の世代）から始まる昇順のオーダー番号が付くことに注意してください。

#### 例

RELGEN=0 は世代番号 690、RELGEN=3-2 は世代番号 687 および 688 にそれぞれ相当します。

I	GEN-	I	BLOCK	I	DATE /TIME	I	
I	NUMBER	I	FROM	TO I	FROM	TO	I
I	-----I	I	-----I	I	-----I	I	I
I	690	I	715	715	I 2001-08-20 02:07:13	2001-08-20 08:51:19	I
I	689	I	714	714	I 2001-08-17 18:24:49	2001-08-20 02:03:21	I
I	688	I	713	713	I 2001-08-16 18:24:26	2001-08-17 16:48:16	I
I	687	I	712	712	I 2001-08-15 18:29:09	2001-08-16 12:54:28	I
I	686	I	711	711	I 2001-08-14 18:24:30	2001-08-15 17:45:44	I
I	685	I	710	710	I 2001-08-13 18:32:07	2001-08-14 15:46:25	I
I	684	I	709	709	I 2001-08-13 02:07:15	2001-08-13 18:00:18	I
I	683	I	708	708	I 2001-08-10 18:25:59	2001-08-13 02:03:23	I
I	682	I	707	707	I 2001-08-09 18:36:39	2001-08-10 10:24:14	I

### RLOGDEV：RLOG 代替デバイス

RLOGDEV は、RLOG ファイルが含まれるデバイスタイプを指定します。RLOG ファイルが ADARUN DEVICE パラメータで指定されたデバイスタイプ（デフォルトのデバイスタイプ）上に存在する場合は、RLOGDEV を指定する必要はありません。

## 例

### 入力例

```
ADARAI LIST
```

この例では、RLOG に存在するすべての世代が出力されます。

```
ADARAI LIST RELGEN=15-1
```

LIST によって最後から 15 世代（RLOG に存在する場合）が表示されます。現行世代（0）は含まれません。

### 出力例

BS2000

## LIST：現在の RLOG 世代の表示

---

次の例は、1つの BS2000 ディスクデータセットの LIST 出力を示します。

```
LINK=DDSAVE1 PATHNAME=:A:$GEB.RAI.vv.SAVE.012
SIZE=6387 SEC-ALLO=96 LPP=6336
FCBTYPE=SAM RECFORM=V BLKSIZE=(STD,16) RECSIZE=32748 BLKCNTRL=PAMKEY
VSN/DEV PUBA00/D3480 /AC PUBA01/D3480 /AC
VSN/DEV PUBA02/D3480 /AC
```

次の例は、BS2000 ファイル世代グループ (FGG) の LIST 出力を示します。

```
FGG INDEX BASE=10 CURRENT=10 FIRST=1 MAX=255 DISP=DEL
LINK=DDSAVE1 PATHNAME=:A:$GEB.RAI.vv.SAVE.TAPE.01(*0010)
TAPE DEVICE=TAPE=C1 (B5) FSEQ=1 BLKCOUNT=4000
FCBTYPE=SAM RECFORM=V BLKSIZE=32760 RECSIZE=32756 BLKCNTRL=NO
VOLUMES GEBR11 GEBR12 GEBR13 GEBR17 GEBR19
```

次の例は、ADARAILIST エラー／警告出力を示します。1つ目は失われたディスクデータセットの例で、2つ目は上書きされたテープデータセットの例です。

```
LINK=DDSAVE1 DISC DATASET NOT PRESENT - E R R O R
ORIGINAL CATALOG ENTRY:
LINK=DDSAVE1 PATHNAME=:A:$GEB.RAI.vv.SAVE.012
SIZE=6387 SEC-ALLO=96 LPP=6336
FCBTYPE=SAM RECFORM=V BLKSIZE=(STD,16) RECSIZE=32748 BLKCNTRL=PAMKEY
VSN/DEV PUBA00/D3480 /AC PUBA01/D3480 /AC
VSN/DEV PUBA02/D3480 /AC

LINK=DDSAVE1 CFID MISMATCH - W A R N I N G
ORIGINAL 379949EE NOW: 379972F0
ORIGINAL CATALOG ENTRY:
LINK=DDSAVE1 PATHNAME=:A:$GEB.RAI.vv.SAVE.012
TAPE DEVICE=TAPE=C1 (B5) FSEQ=1 BLKCOUNT=4000
FCBTYPE=SAM RECFORM=V BLKSIZE=32760 RECSIZE=32756 BLKCNTRL=NO
CREATION DATE yyyy-mm-dd 11:44:35
VOLUMES GEBR11 GEBR12 GEBR13 GEBR17 GEBR19
ACTUAL CATALOG ENTRY:
LINK=DDSAVE1 PATHNAME=:A:$GEB.RAI.vv.SAVE.012
TAPE DEVICE=TAPE=C1 (B5) FSEQ=1 BLKCOUNT=3900
FCBTYPE=SAM RECFORM=V BLKSIZE=32760 RECSIZE=32756 BLKCNTRL=NO
CREATION DATE yyyy-mm-dd 12:34:56
VOLUMES GEBR23 GEBR65 GEBR66 GEBR67 GEBR68
```

**z/OS**

A D A R A I Vv.v SMv DBID = 00203 STARTED yyyy-mm-dd hh:mm:ss

PARAMETERS:

-----

ADARAI LIST RELGEN=0  
RECOVERY LOG FILE FOR DATABASE 203

START RABN FOR LOG DATA AREA IS 21  
HIGHEST LOG AREA RABN IS 633  
CURRENT VALUE FOR ROTATING RABN IS 23

GEN-	I	I	BLOCK	I	DATE /TIME	I
I NUMBER	I S I	I	FROM	TO I	FROM	TO I
I	3 I N I	I	23	23 I	yyyy-01-13 16:06:28	yyyy-01-13 16:11:35 I
I	2 I N I	I	22	22 I	yyyy-01-09 16:07:10	yyyy-01-13 16:04:13 I
I	1 I N I	I	21	21 I	yyyy-01-09 16:04:41	yyyy-01-09 16:06:16 I
I	0 I R I	I	20	20 I	yyyy-01-09 16:04:07	yyyy-01-09 16:04:30 I

\*\*\* yyyy-01-13 16:06:28  
\*\*\* SAVE DATABASE OFFLINE

DELTA SAVE ID IS AS FOLLOWS:  
FULL SAVE.....2  
LOW DELTA SAVE NUMBER...0  
HIGH DELTA SAVE NUMBER..0  
DATE WRITTEN.....yyyy-01-13  
TIME WRITTEN.....16:12:03

FILES = 1,2,3,19

ADARUN DBID=203,SVC=249,DEVICE=3390,PLOGRQ=YES  
ADARUN NCLOG=2,CLOGDEV=3390,CLOGSIZE=150  
ADARUN NPLOG=2,PLOGSIZE=1350  
ADARUN PLOGDEV=3390  
ADARUN DSF=YES  
ADARUN UEX2=USEREX2M  
ADARUN PROG=ADASAV

ADASAV SAVE

//DDSAVE1 DD DSN=EXAMPLE.ADASAV.FULL.G0058V00,  
// UNIT=3390,SPACE=(TRK,(5,5)),DISP=NEW,  
// DCB=(RECFM=VB,BLKSIZE=27998,LRECL=27994),

LIST : 現在の RLOG 世代の表示

---

```
// VOL=SER=(SMS018)
```

DDSAVE1 VOLSER=SMS018 FROM BLOCK=1 (ASSO) TO BLOCK =1598 VOLUME IS ASSOCIATED WITH PLOG NO. 6

```
DDSAVE1 VOLSER=SMS018 FROM BLOCK=1 (DATA)
          TO BLOCK =750
          VOLUME IS ASSOCIATED WITH PLOG NO. 6
```

```
*** yyyy-01-13 16:07:09 NUCLEUS PLOG NUMBER=7
*** START NUCLEUS SESSION [NUCID=nnnnn]
```

```
SYNC PLOG BLOCK NUMBER = 5
[ACTIVE PLOG DATASET NAMES: EXAMPLE.DBdddd.PLOGR21
                             EXAMPLE.DBdddd.PLOGR22]
```

```
ADARUN DBID=203,SVC=249,DEVICE=3390,PLOGRQ=YES
ADARUN NCLOG=2,CLOGSIZE=150,CLOGDEV=3390
ADARUN NPLOG=2,PLOGSIZE=1350
ADARUN PLOGDEV=3390
ADARUN DSF=YES
ADARUN UEX2=USEREX2M
ADARUN PROG=ADANUC
ADARUN MODE=MULTI
ADARUN LOCAL=YES
ADARUN SPT=NO
ADARUN LWP=480000
ADARUN LP=200
ADARUN TT=1800
ADARUN TNAE=1800
ADARUN LBP=80000
ADARUN NH=500
ADARUN LFP=60000
ADARUN LU=65525
ADARUN NAB=45
ADARUN LQ=12000
ADARUN LI=20000
ADARUN NT=10
ADARUN NC=300
ADARUN NU=300
ADARUN LS=20000
ADARUN TNAX=1800
ADARUN CT=300
ADARUN OPENRQ=NO
ADARUN LOGGING=NO
ADARUN LOGCB=NO
ADARUN LOGSB=NO
ADARUN LOGFB=NO
```

```
ADARUN IGNDIB=NO
ADARUN FORCE=NO

*** yyyy-01-13 16:07:18 NUCLEUS PLOG NUMBER=7
*** END NUCLEUS SESSION

HIGHEST PLOG BLOCK WRITTEN = 7

*** yyyy-01-13 16:07:22
*** COPY MULTIPLE PROTECTION LOG DATASET FOR PLOG 7

ADARUN DBID=203,SVC=249,DEVICE=3390,PLOGRQ=YES
ADARUN NCLOG=2,CLOGSIZE=150,CLOGDEV=3390
ADARUN NPLOG=2,PLOGSIZE=1350
ADARUN PLOGDEV=3390
ADARUN DSF=YES
ADARUN UEX2=USEREX2M
ADARUN PROG=ADARES,MODE=MULTI

ADARES PLCOPY OPENOUT
ADARES DSIMSIZE=5

//DDSIAUS1 DD DSN=EXAMPLE.PLOG.G0243V00,UNIT=3390,
//          SPACE=(TRK,(10,1)),DISP=NEW,DCB=(RECFM=VB,
//          BLKSIZE=27998,LRECL=27994),
//          VOL=SER=(SMS018)

DDSIAUS1  VOLSER=SMS018      FROM BLOCK=1
                                TO BLOCK  =7
                                FROM DATE =yyyy-01-13 17:07:09
                                TO  DATE =yyyy-01-13 17:07:18
                                VOLUME IS ASSOCIATED WITH PLOG NO. 7

*** yyyy-01-13 16:07:39 NUCLEUS PLOG NUMBER=8
*** START NUCLEUS SESSION [NUCID=nnnnn]

SYNC PLOG BLOCK NUMBER = 3
[ACTIVE PLOG DATASET NAMES: EXAMPLE.DBddddd.PLOGR21
                             EXAMPLE.DBddddd.PLOGR22]

ADARUN DBID=203,SVC=249,DEVICE=3390,PLOGRQ=YES
ADARUN NCLOG=2,CLOGSIZE=150,CLOGDEV=3390
ADARUN NPLOG=2,PLOGSIZE=1350
ADARUN PLOGDEV=3390
ADARUN DSF=YES

ADARUN UEX2=USEREX2M
ADARUN PROG=ADANUC
ADARUN MODE=MULTI
ADARUN LOCAL=YES
```

```
ADARUN SPT=NO
ADARUN LWP=480000
ADARUN LP=200
ADARUN TT=1800
ADARUN TNAE=1800
ADARUN LBP=80000
ADARUN NH=500
ADARUN LFP=60000
ADARUN LU=65525
ADARUN NAB=45
ADARUN LQ=12000
ADARUN LI=20000
ADARUN NT=10
ADARUN NC=300
ADARUN NU=300
ADARUN LS=20000
ADARUN TNAX=1800
ADARUN CT=300
ADARUN OPENRQ=NO
ADARUN LOGGING=NO
ADARUN LOGCB=NO
ADARUN LOGSB=NO
ADARUN LOGFB=NO
ADARUN IGNDIB=NO
ADARUN FORCE=NO
```

```
*** yyyy-01-13 16:09:16 NUCLEUS CHECKPOINT
ENCOUNTERED
```

```
CHECKPOINT IS ON PLOG NUMBER 8 BLOCK NUMBER 4
SYNS-CHECKPOINT IS 'DELETE FILE'
FILES = 1
```

```
*** yyyy-01-13 16:09:16 NUCLEUS CHECKPOINT ENCOUNTERED
```

```
CHECKPOINT IS ON PLOG NUMBER 8 BLOCK NUMBER 5
SYNS-CHECKPOINT IS 'DELETE FILE'
FILES = 2
```

```
*** yyyy-01-13 16:10:27 NUCLEUS PLOG NUMBER=8
```

```
*** ADABAS UTILITY RUN
```

```
SYNP-CHECKPOINT ID IS 'ADALOD - LOAD'
SYNP-CHECKPOINT IS FOUND ON PLOG 8 IN BLOCK NO. 6
FILES = 1
```

```
ADARUN DBID=203,SVC=249,DEVICE=3390,PLOGRQ=YES
ADARUN NCLOG=2,CLOGSIZE=150,CLOGDEV=3390
ADARUN NPLOG=2,PLOGSIZE=1350
ADARUN PLOGDEV=3390
ADARUN DSF=YES
ADARUN UEX2=USEREX2M
```

```
ADARUN PROG=ADALOD,MODE=MULTI

ADALOD LOAD FILE=1
ADALOD NAME='EMPLOYEES'
ADALOD MAXISN=1500,DSSIZE=1
ADALOD TEMPSIZE=15,SORTSIZE=15

//DDEBAND DD
DSN=ADABAS.Vvrs.EMPL,UNIT=3390,DISP=OLD,
//          VOL=SER=(ADA001)

*** yyyy-01-13 16:11:21 NUCLEUS PLOG NUMBER=8
*** ADABAS UTILITY RUN

SYNP-CHECKPOINT ID IS 'ADALOD - LOAD'
SYNP-CHECKPOINT IS FOUND ON PLOG 8 IN BLOCK NO. 7
FILES = 2

ADARUN
PROG=ADALOD,MODE=SINGLE,SVC=249,DEVICE=3390,DBID=203

ADALOD LOAD FILE=2
ADALOD NAME='VEHICLES'
ADALOD MAXISN=1000,DSSIZE=1
ADALOD TEMPSIZE=15,SORTSIZE=15

//DDEBAND DD
DSN=ADABAS.Vvrs.VEHI,UNIT=3390,DISP=OLD,
//          VOL=SER=(ADA001)

*** yyyy-01-13 16:11:31 NUCLEUS PLOG NUMBER=8
*** END NUCLEUS SESSION

HIGHEST PLOG BLOCK WRITTEN = 9

*** yyyy-01-13 16:11:35
*** COPY MULTIPLE PROTECTION LOG DATASET FOR PLOG 8

ADARUN DBID=203,SVC=249,DEVICE=3390,PLOGRQ=YES
ADARUN NCLOG=2,CLOGSIZE=150,CLOGDEV=3390
ADARUN NPLOG=2,PLOGSIZE=1350
ADARUN PLOGDEV=3390
ADARUN DSF=YES
ADARUN UEX2=USEREX2M
ADARUN PROG=ADARES,MODE=MULTI

ADARES PLCOPY OPENOUT
ADARES DSIMSIZE=5

//DDSIAUS1 DD DSN=EXAMPLE.PLOG.G0244V00,UNIT=3390,
```

LIST : 現在の RLOG 世代の表示

```
//          SPACE=(TRK,(10,1)),DISP=NEW,DCB=(RECFM=VB,
//          BLKSIZE=27998,LRECL=27994),
//          VOL=SER=(SMS018)

DDSIAUS1  VOLSER=SMS018    FROM BLOCK=1
                                TO BLOCK  =9
                                FROM DATE =yyyy-01-13  17:07:39
                                TO   DATE =yyyy-01-13  17:11:30
                                VOLUME IS ASSOCIATED WITH PLOG NO. 8

A D A R A I  TERMINATED NORMALLY                yyyy-01-13  16:12:03
```

VSE

```
A D A R A I  Vv.v  SMv  DBID = 00059  STARTED  yyyy-mm-dd  hh:mm:ss

PARAMETERS:

ADARAI LIST

18:45:04 ADAI64  FILE RLOGR1  HAS BEEN OPENED IN ECKD MODE
18:45:04 ADAI64  FILE RLOGM1  HAS BEEN OPENED IN ECKD MODE

RECOVERY LOG FILE FOR DATABASE 59

START RABN FOR LOG DATA AREA IS      26
HIGHEST LOG AREA RABN IS              633
CURRENT VALUE FOR ROTATING RABN IS    26

I GEN-   I   I      BLOCK      I          DATE /TIME          I
I NUMBER I S I  FROM      TO I          FROM          TO          I
I-----I---I-----I-----I-----I-----I-----I-----I
I      1 I N I    26      26 I  yyyy-08-30  17:06:51  yyyy-08-30  18:44:35 I
I      0 I R I    25      25 I  yyyy-08-30  17:01:02  yyyy-08-30  17:05:05 I
I-----I---I-----I-----I-----I-----I-----I-----I

*** yyyy-08-30  17:06:51
*** SAVE DATABASE OFFLINE NON INCREMENTAL

SAVE DATASET PLOG NUMBER      = 1966
ADASAV SAVE
// TLBL SAVE1,'PMIG.ADAvrs.SAVE1',0,ADES01

DDSAVE1  VOLSER=XXXXXX    FROM BLOCK=1
                                TO BLOCK  =6192
                                VOLUME IS ASSOCIATED TO PLOG NO. 1966
FILE=001,002,003,004,005,006,008,009,010,011,012,013,014,015
FILE=016,017,019,021,022,023,025,027
```

```
DDSAVE1      VOLSER=XXXXXX      FROM BLOCK=1
              TO BLOCK   =31961
              VOLUME IS ASSOCIATED TO PLOG NO. 1966
FILE=001,002,003,004,005,006,008,009,010,011,012,013,014,015
FILE=016,017,019,021,022,023,025,027

*** yyyy-08-30 17:08:12 NUCLEUS PLOG NUMBER=1967
*** START NUCLEUS SESSION

*** yyyy-08-30 17:10:15 NUCLEUS PLOG NUMBER=1967
*** ADABAS UTILITY RUNSYNP-CHECKPOINT ID IS 35 (UNLOAD FILE)
SYNP-CHECKPOINT IS FOUND ON PLOG 1967 IN BLOCK NO. 5
FILE=001
ADAULD FILE=1 NUMRECS=100
// DLBL OUT1,'VSESP.SAPLB.ULD2',7,SD
// EXTENT SYS034,SYSWK1,1,0,16365,30

*** yyyy-08-30 17:14:28 NUCLEUS PLOG NUMBER=1967
*** ADABAS UTILITY RUN

SYNP-CHECKPOINT ID IS 35 (UNLOAD FILE)
SYNP-CHECKPOINT IS FOUND ON PLOG 1967 IN BLOCK NO. 8
FILE=001
ADAULD FILE=1 NUMRECS=100
// DLBL OUT1,'VSESP.SAPLB.ULD2',7,SD
// EXTENT SYS034,SYSWK1,1,0,16365,30

*** yyyy-08-30 18:44:35 NUCLEUS PLOG NUMBER=1967
*** ADABAS UTILITY RUN

SYNP-CHECKPOINT ID IS 30 (LOAD FILE)
SYNP-CHECKPOINT IS FOUND ON PLOG 1967 IN BLOCK NO. 12
FILE=004
ADALOD LOAD FILE=4,ISNREUSE=YES,ORTSIZE=5,TEMPSIZE=5,DSSIZE=50B
ADALOD
MAXISN=10,NAME='TESTFILE',DSREUSE=YES,LWP=1000000,LIP=500
ADALOD NUMREC=5,NISIZE=5B,UISIZE=5B
// DLBL EBAND,'VSESP.SAPLB.ULD2',7,SD
// EXTENT SYS011,SYSWK1,1,0,16365,30

ADAI03 RLOGR1          3 READS          0 WRITES
ADAI03 RLOGM1          1 READS          0 WRITES

A D A R A I  TERMINATED NORMALLY                yyyy-08-30 18:45:03
```



# 144

## PREPARE : RLOG の初期化および開始

---

▪ 構文 .....	752
▪ 基本パラメータ .....	753
▪ オプションパラメータ .....	753
▪ 例 .....	753

リカバリログ (RLOG) は、それを使用する前に、準備する必要があります。RLOG ファイルを開始するには、次のステップが必要です。

**ステップ 1 : ADAFRM RLOGFRM** 機能を使用して、**RLOG** ファイルをフォーマットします。

ADARAI PREPARE を実行する前に、ADAFRM ユーティリティの RLOGFRM 機能を使用して、RLOG データセットをフォーマットする必要があります。この操作を行わないと、エラー 159 が返されます。

**ステップ 2 : ADARAI PREPARE** 機能を実行して **RLOG** を準備します。

ADARAI PREPARE は、データベースが非アクティブの状態で行う必要があります。

ADARAI PREPARE 機能は、次のことを定義するために使用します。

- RLOG サイズ (ADAFRM RLOGFRM 機能の SIZE パラメータの値と同じサイズが必要)
- 保持する最小世代数 (デフォルトは 4)
- デバイスタイプ (デフォルトは ADARUNDEVICE パラメータで指定されたデバイスタイプ)

**ステップ 3 : ADASAV SAVE** (データベース) 機能を実行して最初のログ世代を開始します。

PREPARE 機能を実行すると、最初の世代に対してロギングが開始しますが、この世代はフルデータベースセーブ/リストアで開始されていないため、**制限付きステータス**になります。

世代のステータスの詳細については、『*Adabas オペレーションマニュアル*』を参照してください。

このchapterでは、次のトピックについて説明します。

## 構文

---

```
ADARAI PREPARE RLOGSIZE = size  
                [RLOGDEV = { device | ADARUN-device }]  
                [MINGENS = { count | 4}]
```

## 基本パラメータ

### RLOGSIZE : RLOG エリアサイズ

RLOGSIZE は、RLOG ファイルのサイズをシリンダまたはブロック単位で定義します。この値は、ADAFRM RLOGFRM 機能の SIZE パラメータで定義された値と同じである必要があります。RLOGSIZE は指定する必要があります。デフォルトはありません。

 **Note:** RLOG データセットは、16,777,215 (x'FFFFFF') ブロック/RABN 以内に制限されています。

## オプションパラメータ

### MINGENS : RLOG 世代数

MINGENS は、RLOG に保持するロギング世代数を指定します。RLOG では、“0” から開始する昇順の番号が世代に付けられます。最小値は4世代（デフォルト）、最大値は32です。

### RLOGDEV : RLOG デバイスタイプ

RLOGDEV は、RLOG ファイルが含まれるデバイスタイプを指定します。RLOG ファイルが ADARUN DEVICE パラメータで指定されたデバイスタイプ（デフォルトのデバイスタイプ）上に存在する場合は、RLOGDEV を指定する必要はありません。

 **Important:** デフォルトとは異なるデバイスタイプを RLOG データセットに対して選択すると、すべての ADARES PLCOPY および COPY の実行で RLOGDEV パラメータの指定が必要になります。

## 例

### 例 1

```
ADARAI PREPARE MINGENS=4 ,RLOGSIZE=5
```

この ADARAI PREPARE 機能では、ログサイズが5シリンダで少なくとも4世代を保持する RLOG を定義および初期化します。ADARUN DEVICE パラメータで指定されるデバイスは、デフォルトで RLOG デバイスになります。

### 例 2

```
ADARAI PREPARE
RLOGSIZE=20 ,MINGENS=20 ,RLOGDEV=3390
```

この例では、3390 デバイスタイプで20世代までを保持する大規模な RLOG サイズ（20シリンダ）を定義します。



# 145 RECOVER：リカバリジョブストリームの構築

---

▪ リカバリ処理 .....	756
▪ 最適化されたリカバリ処理 .....	758
▪ 要件 .....	758
▪ 制約 .....	759
▪ リカバリに必要な入力 .....	760
▪ リカバリ操作の出力 .....	760
▪ RECOVER 機能の実行 .....	761
▪ ファイルレベルのリカバリ .....	762
▪ 構文 .....	961
▪ オプションのパラメータとサブパラメータ .....	764
▪ 例 .....	968
▪ スケルトンジョブ制御 .....	767
▪ JCL を変更するユーザー出口 .....	770
▪ リカバリ前のチェック .....	770
▪ RECOVER 機能またはリカバリジョブストリームの再スタート .....	771

 **Note:** 現在、RECOVER 機能は BS2000 および z/OS システムでのみ使用できます。z/VM および VSE システムでは、今後サポートされる予定です。

ADARAI RECOVER 機能は、Adabas データベースまたは選択したデータベースファイルを回復するジョブ制御情報（リカバリジョブストリーム）を構築します。RECOVER 機能は次のことを行います。

- PLOG 情報を読み込み、PLCOPY が必要かどうかを判断します。
- RLOG を読み込み、スケルトンジョブ制御からリカバリジョブストリームを構築します。

ADARAI RECOVER は、データベースまたはファイルを RECOVER 機能の実行前の状態にリストアするために必要なジョブストリームを構築します。完全なジョブストリームは、DD/JCLOUT データセットに送られます。

ADARAI は、生成されたジョブストリームにエラーメッセージまたは情報メッセージも適宜含めます。ジョブをサブミットする前に、エラーを手動で修正する必要があります。ジョブストリームにメッセージが存在することは、ADARAI RECOVER から返されるゼロ以外のリターンコードによって示されます。

BS2000 システムの場合、RECOVER 機能では次の処理も実行します。

- ジョブ制御を生成するとき、BS2000 の LIST 機能によって行われるものと同じチェックを実行します。
- エラーが発生したチェックに対して、作成されたジョブ制御に BS2000 の /REMARK ステートメントを含めます。

 **Note:** このようなエラーが発生した場合は、ジョブ制御を手動で修正する必要があります。

このchapterでは、次のトピックについて説明します。

## リカバリ処理

---

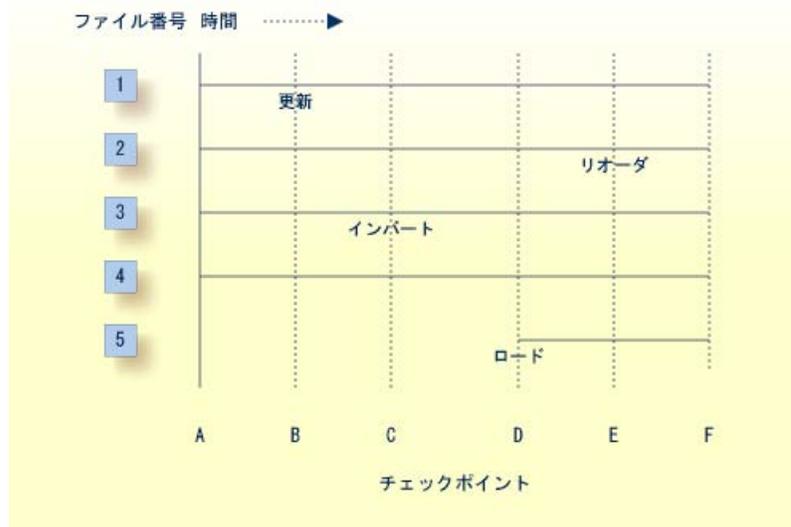
ADARAI RECOVER 機能は、リストア対象の世代として認識した正確なシーケンスに基づいてジョブを構築します。

- 世代を開始した操作で作成されたデータセットからデータベースをリストアします。
- 検出された次のユーティリティチェックポイントまで PLOG を再生成します。
- ユーティリティを再実行するジョブステップを生成し、そのチェックポイントの後で再生成を開始します。

すべてのユーティリティが再生され、世代に存在する最後の PLOG ブロックが再生成されるまで、このシーケンスが続行します。

次の図に、ADARAI の動作を示します。

- 新しい世代を A から開始するためにデータベースが保存されます。
- 世代中のさまざまな時点でデータベースが実行されます。
- ■ ファイル 1 に対する更新の実行
  - ファイル 2 に対するリオーダの実行
  - ファイル 3 に対するインバートの実行
  - ファイル 5 に対するロードの実行



上記の場合、リカバリ順序は次のようになります。

1. データベースを A の状態に戻すため、フルセーブまたはフルセーブと差分セーブがリストアされます。
2. データベース再生成は、A の最初のチェックポイントから B の更新チェックポイントまで実行されます。その後、再生成ジョブは終了します。
3. ファイル 1 の更新ユーティリティが実行され、データベース再生成は B のチェックポイントおよび C のインバートチェックポイント間で実行されます。
4. ファイル 3 のインバートユーティリティが実行され、データベース再生成はチェックポイント C および D のロードチェックポイント間で実行されます。
5. ファイル 5 のロードユーティリティが実行され、データベース再生成はチェックポイント D および E のリオーダチェックポイント間で実行されます。
6. ファイル 2 のリオーダが実行され、データベース再生成はチェックポイント E および F の最新データベースレベル間で実行されます。

## 最適化されたリカバリ処理

ADARAI RECOVER 機能の OPT パラメータを使用すると、大規模なデータベースのリカバリ所要時間を最小限に抑える操作またはシーケンスを識別できます。

例えば、10,000 個の更新を含むファイルを削除または再ロードする場合、初めからファイルをリストアし、10,000 個の更新を再生した後、削除またはロード操作が発生したときにすべてを破棄するような操作は、避ける必要があります。

最適化を選択すると、ADARAI は、例に示すようなファイルのリストアをジョブのメインのリストアでは行いません。ファイルの再生成は、ファイルが削除された後、またはロードによって作成された後にのみ行われます。

- 削除されたファイルには更新が含まれなくなります。
- ロードによって作成されたファイルでは、ロード以降に行われた更新のみが重要です。

最適化されたジョブストリームを使用すると、回復されたデータベースは元の構築とは異なる方法で再構築されます。最適化されたリカバリジョブは元のジョブとまったく同じ方法で再生するわけではないため、リカバリを行うとき、データベースでスペースが不足するなどの問題が発生する場合があります。ただし、ほとんどの場合、データベースリカバリを最適化することによって得ることができる利点に比べると、この問題を重要視する必要はありません。問題が発生する可能性については、状況に応じて判断する必要があります。

## 要件

正常に実行するリカバリジョブを生成するため、ADARAI には次の要件があります。

- デュアルプロテクションロギングまたはマルチプロテクションロギングがアクティブな状態でデータベースを実行する必要があります。
- データベースでファイルを更新するユーティリティ機能に対するシーケンシャルデータセット入力を保持する必要があります。
- SAVE または MERGE 機能で作成されるシーケンシャル出力データセットは、保持する必要があります。SAVE FILE 機能には、RESTFILE=YES を指定して ADARAI RECOVER を使用する場合にのみ、このことが当てはまります。
- 保持されたデータセットの名前は、変更しないで元のままにしておく必要があります。これは、ADARAI では名前の異なるコピーを探知できないためです。

保持されたデータセットをカタログすることをお勧めします。

## 制約

### シャドーデータベース

元のデータベースのセーブに対して差分セーブ出力およびDSIM データセットをリストアすることによってシャドーデータベース（通常の本番データベースのコピー）を構築すると、元のデータベースの差分セーブ中に発生した PLOG アクティビティは ADARAI で認識されません。このため、シャドーデータベースでリストア操作が必要になった場合に、ADARAI は DSIM データセットを再構築できません。

ただし、DSIM データセットと差分セーブデータセットを結合してオフラインの新しい差分セーブデータセットを作成し、結合された新しいデータセットをシャドーデータベースに対してリストアすると、PLOG が不要になるため、シャドーデータベースのリカバリに必要なすべての情報が ADARAI に提供されます。

### DSIM データセットを使用した差分セーブのリストア

一般的に、ADARAI では RESTORE DELTA 処理が問題なく行われます。ただし、RESTORE DELTA で DSIM データセット（本質的には作業データセット）を使用している場合は、ADARAI RECOVER が必要になったとき、DSIM データセットが元の状態に保たれていないことがあります。このため、RECOVER 処理中にこのような RESTORE DELTA を再生する前に、ADARAI では DSIM データセットの作成に使用された COPY または PLCOPY 要求を記録し、データセットを再構築するためのジョブステップを発行します。

ADARAI は、RLOG 全体から適切なエントリを検索します。エントリが検出されないと、RESTORE ステップの前に DSIM データセットを再構築できないため、RESTORE DELTA を再生できません。

### DD/FILEA ファイル

生成されたりカバリジョブで、ADARAI は ADAORD ユーティリティの DD/FILEA ファイルを書き込みます。この処理は、回避することができません。これは、REORDER 機能を再生する必要があり、この機能によって、DD/FILEA ファイルの書き込みが要求されるためです。

この場合、次の制限があります。

- ADAORDSTORE 処理では、リカバリを行っている世代の一部としてユーティリティが実行されたときに、同じ DD/FILEA が単に読み込まれます。
- ステップの実行後に削除される一時ファイル（DISP=NEW、DELETE）を DD/FILEA に対して使用できます。これは、リカバリジョブの実行時にファイルの作成および削除が再度行われるためです。
- 既存のファイル（DISP=OLD）を DD/FILEA に対して使用できます。リカバリジョブが実行されるとき、このファイルがまだ存在すると、ADARAI ではこのファイルに対して、単に元のジョブが実行されたときのファイル属性を割り当てます。
- 新しいファイル（DISP=NEW、CATLG）が DD/FILEA に割り当てられ元の ADAORD REORDER ステップで保持されており、リカバリジョブが REORDER ステップ段階になっ

たときにまだファイルが存在する（正常な動作）場合は、ADARAIでは同じファイルを再度作成しようとし、その結果ジョブが失敗します。

- GDG を使用していると、世代によって作成された実際のデータセットの名前のみが ADARAI リカバリジョブで認識されます。データセットがすでに存在する（正常な動作）と、ADARAI は同じファイルを再度作成しようとし、その結果ジョブが失敗します。

## リカバリに必要な入力

---

次のデータセットは、ADARAI RECOVER 機能に対する入力です。

- DD/RLOGR1：リカバリログ
- DD/PLOGR1 および DD/PLOGRn：マルチプロテクションログ。ADARAI RECOVER パラメータ FEOFPL=YES（デフォルト）を使用した場合に必要です。
- DD/JCLIN：サイトによって異なるスケルトンジョブ制御ステートメントを指定します。RECOVER 操作では、完全なデータベースリカバリジョブストリームを作成するため、このステートメントが RLOG 情報と結合されます。

BS2000 システムの場合、DDJCLIN は可変レコード形式の SAM データセットです。EDT を使用すると、このデータセットを作成および編集できます。詳細は、「[スケルトンジョブ制御](#)」を参照してください。

z/OS システムの場合、RECFM=FB、LRECL=80、および 80 バイトの倍数である BLKSIZE で DDJCLIN データセットを定義する必要があります。

## リカバリ操作の出力

---

ADARAI RECOVER の出力は、データベースリカバリを行う実行準備済みのジョブストリームです。このリカバリジョブストリームは、DD/JCLOUT ファイルに書き込まれます。RECOVER 操作中にエラーの可能性が検出されると、ADARAI は警告メッセージを出力しコンディションコード 4 で終了します。「[リカバリ前のチェック](#)」を参照してください。

BS2000 システムの場合、DDJCLOUT および DDJCLCON は可変レコード形式の SAM データセットです。これは、BS2000 のジョブ制御規則に準拠しています。

z/OS システムの場合、RECFM=FB、LRECL=80、および 80 バイトの倍数である BLKSIZE として定義されたデータセットを DDJCLOUT DD ステートメントでポイントしている必要があります。

リカバリジョブストリームには、次の時点でニュークリアスを開始するジョブステップが含まれています。

- 最初の再生成ジョブステップの前および

## ■ ニュークリアスを自動終了するユーティリティ操作の後

ユーティリティが最初にシングルユーザーモードで実行されたかどうかにかかわらず、ADARAI RECOVER ジョブでは、データベースがアクティブな状態で、すべてのユーティリティを再生します。最初にシングルユーザーモードで実行されたユーティリティは、マルチユーザーモードで再生されます。これらのジョブステップについては、「[リカバリジョブストリームの構築](#)」および「[スケルトンジョブ制御](#)」を参照してください。

## RECOVER 機能の実行

RECOVER 機能は、RELGEN パラメータの制御の下に一度に 1 世代ずつ実行されます。RELGEN を指定しない場合、デフォルトは現行世代です。

RECOVER は、ニュークリアスがアクティブまたは非アクティブのどちらの状態でも実行できます。この機能は世代の RLOG 情報を変更しないため、同じ世代に対して複数回実行できます。

ただし、RECOVER が失敗した後、DD/JCLOUT リカバリジョブストリームの実行中に RECOVER を再実行する場合は、作成される新しいリカバリジョブストリームが元のリカバリジョブストリームと異なることがあります。これは、RLOG を更新するデータベースに対して、元のリカバリジョブストリームでユーティリティを実行する場合があります。新しい RECOVER 操作ではその後、失敗したリカバリジョブストリームの一部として実行されるユーティリティのリカバリジョブストリームが構築されます。

また、ADASAV RESTORE の実行後にリカバリジョブストリームが失敗すると、新しい世代が作成されます。この場合は、RELGEN=1 のパラメータ設定を使用して RECOVER を実行し、元の世代を取得してください。

### PLOG の処理

ADARAI RECOVER FEOFPL=YES と指定すると、RECOVER は実行の開始時点で PLOG を読み込み、コピーする必要がある情報を検索します。必要に応じて、ニュークリアスをコールし、PLOG 切り替えを強制的に行います。ニュークリアスが非アクティブの場合は、ユーザー出口 2 が起動されます。

### リカバリログの読み込み

次に、RECOVER はスケルトンジョブ制御をストレージに読み込み、RELGEN パラメータで指定された世代の先頭から開始して発生順に RLOG を読み込みます。世代の定義については、「[世代：リカバリの単位](#)」を参照してください。

データベース全体をリカバリする場合、RECOVER は ADASAV SAVE または RESTORE 情報を使用して新しい RESTORE/RESTONL データベース操作を作成します。ファイルレベルのリカバリの場合は、SAVE/RESTORE データベース情報を使用して RESTORE FILE=... 機能を作成します。

### リカバリジョブストリームの構築

データベースまたはファイルをリストアするジョブストリームを作成した後、RECOVERでは %JCL-STARTNUC ステートメントを使用してニュークリアスを開始するジョブステップを作成します。

次に、最初の再生成ジョブステップを作成します。世代の開始がオンライン SAVE（または DELTA SAVE）に基づいていない限り、FROM チェックポイント（FROMCP）はこのジョブステップに含まれません。その場合、再生成はオンラインセーブの最後のチェックポイント（SYN2）で開始します。

ユーティリティの次のチェックポイント（REGENERATE が停止すべきポイント）までのすべての PLOG が取り込まれ、適切なパラメータが ADARES REGENERATE 機能に対して指定されます。再生成する PLOG が 99 個を超えると、ADARAI は複数の REGENERATE ジョブステップを生成し、各ジョブステップで 99 個までの入力 PLOG データセットが処理されます。

次のユーティリティ実行までの PLOG が再生成されると、ユーティリティジョブステップが出力リカバリジョブに生成されます。次に ADARAI は、ユーティリティの次のチェックポイントまでの PLOG がすべて含まれている別の REGENERATE ジョブステップを挿入します。

リカバリジョブは、RELGEN パラメータで指定された世代の終わりが検出されるまで、REGENERATE ステップおよびユーティリティステップの挿入を続けます。この時点で、完全なジョブストリームが DD/JCLOUT ファイルに送られます。

## ファイルレベルのリカバリ

---

RECOVER 機能の FILE パラメータを指定すると、ファイルレベルでリカバリを行うことができます。ファイルレベルのリカバリ処理は、基本的にはデータベースレベルのリカバリ処理と同じですが、FILE パラメータで指定されたファイルに処理が限定されます。

ADARAI は、パラメータをユーティリティ実行ステートメントに追加することによって、ファイル固有の結果を DD/JCLOUT に出力します。例えば、次のステートメントが元の ADASAV RESTORE ステートメントに含まれているとします。

```
ADASAV RESTORE FMOVE=2,3,NIRABN=100,1000,DSSIZE=550B,20
```

この場合、RECOVER FILE=3 から次の DD/JCLOUT ステートメントが作成されます。

```
ADASAV RESTORE FMOVE=2,3,NIRABN=100,1000,DSSIZE=550B,20
ADASAV EXCLUDE=2
```

 **Note:** リカバリ対象ファイルが拡張ファイルチェーンの一部であるかカップリングされていると、チェーンまたはカップリングリスト内のすべてのファイルをまとめてリカバリする必要があります。カップリングファイルまたは拡張ファイルチェーンがすべてまとめてリカバリされないと、ADARAI でこれが検出され、ADARAI RECOVER 機能は失敗します。

ファイルレベルのリカバリジョブを実行する前に、Adabas ニュークリアスはアクティブになっている必要があります。データベース自体を開始するデータベースレベルのリカバリジョブとは、この点が異なります。

ファイルレベルの RECOVER 操作では、データベース全体で実行されたユーティリティに対するジョブ制御 (ADADEF NEWWORK など) は作成されません。ただし、データベース全体だけでなく個々のファイルに対しても再実行できるユーティリティは例外です。このようなユーティリティの例として、ADASAV RESTORE FILE=... ジョブ制御の作成に使用できる DD/SAVE 入力データセットを指定する ADASAV RESTORE (データベース) を挙げるすることができます。

## 構文

```
ADARAI RECOVER [AUTOBACKOUT]
[DRIVES = {number | 1}]
[DSIMSIZE = {size, DSIMDEV = {device | ADARUN-device}}]
[ FEOFPL = { NO
              YES [, PLOGDEV = {device | ADARUN-device} ] } ]
[FILE = { file-list [, AUTOBACKOUT]}]
[JCLLOG = { YES | NO}]
[OPT = { YES | NO}]
[RELGEN = {number | 0}]
[RESTFILE = { YES | NO}]
[RLOGDEV = { device | ADARUN-device}]
```

## オプションのパラメータとサブパラメータ

### AUTOBACKOUT：未完了トランザクションのバックアウト

AUTOBACKOUT は、ファイルレベルのリカバリでのみ指定できます。

AUTOBACKOUT を指定すると、リカバリジョブの最後の REGENERATE 機能が終了した時点で、未完了のトランザクションがバックアウトされます。完了したトランザクションのみがデータベースに残ります。

AUTOBACKOUT を指定しないと、未完了のトランザクションがデータベースに残ります。

データベースレベルのリカバリの場合、最後の REGENERATE 機能が終了した時点で、未完了のトランザクションは常にバックアウトされます。

### DRIVES：ADASAV リストアの入力ドライブボリューム

DRIVES は、生成中のリカバリジョブの RESTORE ステップに対する入力として使用する入力データセット数を指定します。

DRIVES パラメータには、世代を開始したジョブの DRIVES パラメータ以下の値を指定する必要があります。例えば、DRIVE=4 のデータベースセーブで世代を開始した場合、RECOVER DRIVES パラメータには 1、2、3、または 4 のみを指定できます。

RESTORE ステップの DRIVES に低い値を指定すると、ADARAI RECOVER は必要な DD/RESTn DD/DLBL のみを割り当て、各 DD/RESTn DD/DLBL に同じ個数の入力データセットを割り当てます。

### DSIMDEV：DSIM データセットのデバイスタイプ

DSIMDEV は、ADARUN DEVICE パラメータで指定されたもの（デフォルト）と異なる場合に、DSIM データセットのデバイスタイプを指定します。

### DSIMSIZE：DSIM データセットサイズ

サイズをシリンダ単位で指定します。

リカバリ中のデータベースで Adabas Delta Save Facility がアクティブになっている場合は、ADARAI で生成しなければならない可能性のある ADARES COPY 操作に対して DSIMSIZE を指定可能にするため、このパラメータの指定が必要です。

### FEOFPL：複数の PLOG の同期

FEOFPL=YES（デフォルト）と指定すると、すべての複数の PLOG データセットからプロテクションログ（PLOG）データが確実にコピーされます。

- ニュークリアスがアクティブの場合、ADARAI はプロテクションログを強制的に切り替えます。次に、ニュークリアスはログデータをコピーするユーザー出口 12 をコールし、ADARAI はコピーが完了するまで待機します。したがって、FEOFPL=YES を指定する場合は、ADARUN パラメータ UEX12 も指定する必要があることに注意してください。
- ニュークリアスがアクティブでない場合は、ADARAI からユーザー出口 12 をコールし、ログデータがコピーされます。

ニュークリアスクラス環境では、FEOFPL=YES の動作が異なります。

- 少なくとも1つの Adabas ニュークリアスが使用可能な場合、ADARAI はニュークリアスをコールして PLOG を切り替えます。
- Adabas ニュークリアスが使用可能でない場合、ADARAI は手動実行を必要とするジョブを生成します。

どちらでもない場合は、FEOFPL=NO を指定して ADARAI を再スタートする必要があります。

#### FILE：ファイル番号

FILE は、リカバリジョブストリームの構築時に取り込む1つ以上のデータベースファイルを指定します。FILE を指定すると、データベースレベルのリカバリではなくファイルレベルのリカバリが行われ、指定されたファイルのみが RECOVER 操作の対象になります。FILE を指定しないと、すべてのデータベースファイルが取り込まれます（デフォルト）。

#### JCLLOG：ユーザー指定のジョブ制御

JCLLOG は、ユーザー指定の入力ジョブ制御（DDJCLIN 内の JCL または JCLIN 内の JCS）のリストを制御します。JCLLOG=YES と指定すると、ユーザー指定の入力ジョブ制御エレメントがユーティリティログに出力されます。デフォルトは、入力ジョブ制御ステートメントのリストなし（NO）です。

#### OPT：世代のリカバリジョブの最適化

OPT=YES と指定すると、ADARAI は特定の世代に対して作成するリカバリジョブを最適化しようとします。つまり、データベースまたはファイルを元の論理状態に戻すために不要なステップを除外しようとします。

OPT=NO と指定すると、リカバリジョブは最適化されません。

 **Note:** データベース上のスペースが限定されていると、データベースの構築方法が元の方法と異なるため、最適化されたリカバリジョブが失敗することがあります。この状態が発生した場合、最適化なしで生成されたリカバリジョブを使用するか、リカバリを行う前にデータベースのサイズを増やす必要があります。

#### PLOGDEV：マルチプロテクションログのデバイスタイプ

PLOGDEV 値は、FEOFPL=YES を指定した場合にのみ使用されます。

PLOGDEV は、ADARUN DEVICE パラメータで指定されたもの（デフォルト）と異なる場合に、PLOG デバイスタイプを指定します。

#### RELGEN：相対リカバリ世代番号

RELGEN は、リカバリに使用する相対世代番号を指定します。現行世代は常に、デフォルトである相対世代 "0"（ゼロ）でカップリングされます。2 世代前、つまり最後に完了した世代の前の世代は、相対世代 "2" として指定されます。

指定された世代は、現在 RLOG に存在している必要があります。使用可能な現在の RLOG 世代を確認するには、ADARAI LIST 機能を使用します。ただし、表示される世代には、世代 "1"（RLOG 操作開始後の最初の世代）から始まる昇順の番号が付くことに注意してください。

### RESTFILE：リストアファイルのジョブステップの作成

RESTFILE=NO（デフォルト）と指定すると、記録された ADASAV SAVE FILE= 実行の ADASAV RESTORE FILE=... ジョブステップが DDJCLOUT リカバリジョブストリームに含まれなくなります。ADARES REGENERATE は ADASAV SAVE FILE=... チェックポイントで停止しないため、このようなジョブステップは含まれません。

RESTFILE=YES と指定すると、記録された ADASAV SAVE FILE=... ユーティリティのすべての実行に対して、リカバリジョブストリームで ADASAV RESTORE FILE=... ジョブステップが作成されます。



**Note:** RESTFILE=YESを使用する場合は、世代で作成されたファイルセーブデータセットを保持する必要があります。

RESTFILE=YES および OPT=YES の両方を指定すると、RESTORE ステップまでのリストア済みファイルが無視されるため、RESTORE FILE= によってリカバリ処理時間が短縮されます。

RESTFILE=YES および OPT=NO を指定すると、不要な RESTORE ステップがリカバリジョブに含まれます。リカバリジョブをこのように生成してから、目的のファイルに関して RESTORE ステップの前にあるすべてのステップを手動で削除してもかまいません。

### RLOGDEV：RLOG 代替デバイス

RLOGDEV は、RLOG ファイルが含まれるデバイスタイプを指定します。RLOG ファイルが ADARUN DEVICE パラメータで指定されたデバイスタイプ（デフォルトのデバイスタイプ）上に存在する場合は、RLOGDEV を指定する必要はありません。

## 例

---

### 例 1

```
ADARAI RECOVER,DRIVES=3
```

RECOVER 機能では、現行世代（0、デフォルト）に基づいて、リカバリジョブストリームを構築します。ジョブストリームの SAVE RESTORE 部分に、3つの入力データセット DDREST1、DDREST2、および DDREST3 のステートメントを含めます。

### 例 2

```
ADARAI RECOVER FILE=3,4,7,8,11
ADARAI RELGEN=2, JCLLOG=YES
```

リカバリジョブストリームは 3 番目に古い世代に基づいており、データベースファイル 3、4、7、8、および 11 のみのアクティビティが存在し、ファイルレベルのジョブ制御を作成します。RECOVER では、データセット DDJCLIN からユーティリティログにユーザー指定のジョブ制御も追加します。

## 例 3

```
ADARAI RECOVER,RELGEN=1,OPT=Y
```

RECOVER機能は、最後の世代（現行世代の前にある世代）に基づいてリカバリジョブストリームを構築します。ADARAIでは、リカバリジョブの処理時間を短縮するために不要な処理を削除します。

## スケルトンジョブ制御

スケルトンジョブ制御は、DD/JCLINファイルに含まれており、RECOVER機能の入力として読み込まれます。RECOVERは、スケルトンジョブ制御とRLOG情報を結合してリカバリジョブストリームを作成します。スケルトンジョブ制御は、通常、安定しており、操作環境に固有です。

スケルトンジョブ制御内の各機能は、次の形式のステートメントで識別されます。

```
%%name
```

%%JCL-ADASAV または %%JCL-STARTNUC などのように、名前は機能に固有です。ジョブ制御ステートメントは %% 名前ステートメントに続き、次の %%JCL ステートメントで終了します。各スケルトンセクションには、コメントやプログラム実行など、有効な任意のジョブ制御ステートメントを含めることができます。これにより、リカバリ処理を柔軟に行うことができます。

ADARAIでは、スケルトンジョブ制御内のステートメントの妥当性をチェックしません。無効なステートメントは、リカバリジョブストリームの実行中にジョブ制御でエラーが発生して初めて判明します。

### ジョブヘッダー：%%JCL-JOB-HEADER

ジョブヘッダーステートメントは、リカバリジョブストリームの先頭で他のジョブ制御ステートメントの前に配置します。

このジョブ制御は完全なリカバリジョブに関連しており、OS用のJOBとJOBLIBステートメントおよびVSE用のPOWERJCLとJOBステートメントが含まれています。

### ジョブトレーラ：%%JCL-JOB-TRAILER

ジョブトレーラストatementは、リカバリジョブストリームの最後に配置します。

%%JCL-STARTNUC セクションで推奨されるように ADARUN UTIONLY=YES パラメータを指定してニュークリアスを開始した場合は、このセクションで ADADBS OPERCOM UTIONLY=NO 機能を実行するステートメントを指定して、リカバリ操作の後でデータベースを使用可能にすることがあります（このドキュメントの後にあるスケルトンジョブ制御の例を参照）。

**ステップトレーラ：%%JCL-STEP-TRAILER**

ステップトレーラステートメントは、リカバリジョブストリームの各ステップの後に配置します。

**DD/KARTE ジョブ制御：%%JCL-DDKARTE**

オペレーティングシステムによって異なる DD/KARTE ステートメントは、ADARAI で RLOG から生成される DD/KARTE パラメータの前にある各ジョブステップに含まれます。

z/OS および VSE では、ジョブストリームに DD/KARTE パラメータが含まれていることを、このステートメントで示す必要があります。

**DD/FILEA ジョブ制御：%%JCL-DDFILEA**

この（オプションの）JCL カードは、ADAORD REORDER 処理で問題を回避するために指定します。プレースホルダとして、ジョブ内の元の DD/FILEA ステートメントに異なる DD/DLBL ステートメントを指定することができます。指定した場合は、ADAORD REORDER が後で発生したときに、元の DD/FILEA ステートメントの代わりに挿入されます。

**ユーティリティジョブ制御：%%JCL-utility**

このスケルトンセクションは、リカバリジョブストリームでユーティリティジョブステップを作成するために使用します。DD/JCLIN では、次のユーティリティジョブが使用可能です。

%%JCL-ADADEF	%%JCL-ADAORD
%%JCL-ADAINV	%%JCL-ADARES
%%JCL-ADALOD	%%JCL-ADASAV

各セクションには、次のものが含まれています。

- ユーティリティ実行の必要に応じて、DD/ASSOR1、DD/DATAR1、DD/WORKR1、DD/SORTR1、DD/TEMPR1 などのデータベースファイル
- DD/TEMPR1 オーバーフローファイルとして使用する場合の ADALOD 用の DD/FILEA
- DD/PRINT および DD/DRUCK ステートメントまたは割り当て
- DD/CARD ステートメントまたは割り当ておよび DBID、DEVICE、PROG、SVC などのすべての ADARUN 必須パラメータ
- Adabas ライブラリまたは他のライブラリに関する必要情報

DD/CARD パラメータ、データベースファイル、またはライブラリに、プロシージャまたは区分データセット（PDS）メンバを使用できます。

**ニュークリアスを開始するジョブ制御：%%JCL-STARTNUC**

このジョブ制御は、Adabas ニュークリアスの開始に必要なすべてのステートメントで構成されます。RECOVER 機能はこのジョブ制御を使用して、最初の再生ジョブステップの前に（ニュークリアスがすでにアクティブになっていない場合は、アクティブなニュークリアスを必要とする各ユーティリティコールの前に）ニュークリアスを開始するジョブステップを作成します。

このジョブ制御には、以下を含むニュークリアスジョブ全体が取り込まれている必要があります。

- ジョブステートメント
- プログラム実行ステートメント
- ライブラリ定義
- データベースファイル定義
- ADARUN パラメータを含む DD/CARD 情報

このセクションでは、適切なジョブ入力システム（BS2000 のプロシージャモードの EDT および z/OS の IEBGENER など）にニュークリアスジョブ制御をサブミットする方法も必要になります。このジョブ制御の例については、このドキュメントの後半にあるスケルトンジョブ制御例の %%JCL-STARTNUC セクションを参照してください。

ニュークリアスが実際にアクティブになるまでリカバリジョブストリームの実行を停止する方法を、このジョブ制御に含めることも重要です。例えば、データベースに対して CL（クローズ）コマンドを発行するプログラムを作成できます。レスポンスコード 148 によってデータベースがアクティブでないことが示された場合は、プログラムは指定時間待機し、その後、CL コマンドを再発行できます。プログラムの処理はレスポンスコード 0 が発生するまで続行し、この後、次のリカバリステップが実行可能になります。この操作は、ADARAI CHKDB ACTIVE 機能を使用して実行できます。

#### ニュークリアスを停止するジョブ制御：%%JCL-ENDNUC

非アクティブなニュークリアスを必要とするユーティリティが検出されるたびに、Adabas ニュークリアスに対して ADAEND を実行するため、RECOVER は %%JCL-ENDNUC ジョブ制御をジョブストリームに挿入します。ADADBS OPERCOM ADAEND 機能は、ニュークリアスの停止にも使用できます。ADADBS OPERCOM を使用する場合は、ADADBS OPERCOM 機能の実行に必要なすべてのステートメントがジョブ制御ステートメントに含まれている必要があります。ニュークリアス開始スケルトンジョブ制御の場合と同様に、ニュークリアスが非アクティブになるまでリカバリジョブストリームの実行を停止する方法も必要になります。この操作は、ADARAI CHKDB INACTIVE 機能を使用して実行できます。

#### 特殊なジョブ制御ステートメント

次の特殊なキーワード／ステートメントは、DD/JCLOUT リカバリジョブストリームの世代を制御するため、DD/JCLIN スケルトンジョブ制御で使用されます。

<b>%STEP</b>	(オプションの) %STEP キーワードがプログラム実行ステートメントに含まれていると、%STEP キーワードが含まれているジョブステップごとに、ジョブストリームにステップ番号が生成されます。ステップ番号は 1 で始まる昇順の番号です。
<b>%SEQUENTIAL</b>	シーケンシャルファイルのジョブ制御ステートメントを生成する各 %% スケルトンセクションに含まれている必要があります。ADARAI は、%SEQUENTIAL ステートメントの代わりに、必要なシーケンシャルジョブ制御ステートメントを作成します。このステートメントが含まれない場合は、処理中にエラーが発生します。

%KARTE	Adabas DD/KARTE パラメータが生成される各 %% スケルトンセクションに含まれている必要があります。ADARAI は、%KARTE ステートメントの代わりに、必要な DD/KARTE パラメータを作成します。このステートメントが含まれない場合は、処理中にエラーが発生します。
%DBID	(オプションの) %DBID キーワードがプログラム実行ステートメントに含まれていると、5桁のデータベース ID 番号が生成されます。データベース番号が5桁未満の場合は、先行ゼロが埋め込まれます。

## JCL を変更するユーザー出口

ADARAI には、自動生成されるリカバリジョブをサブミットする前にユーザーが変更できるようにするため、ユーザー出口 UEXRAI が用意されています。必要な変更としては、デバイスタイプやボリューム名などがあります。

UEXRAI は、DDJCLOUT に JCL レコードが書き込まれる直前に JCL レコードを取得します。

ユーザー出口をコールするには、レジスタを次のように設定してください。

R1	DDJCLOUT にまもなく書き込まれる JCL レコード行
R13	72 バイトの標準レジスタセーブエリア
R14	リターンアドレス
R15	エントリポイント

## リカバリ前のチェック

リカバリジョブストリームを開始する前に、リカバリデータベースとリカバリジョブストリームのステータスをチェックします。

データベースレベルのリカバリの場合は、次のことがチェックされます。

- 既存のニュークリアスセッションが終了したこと
- セッションエントリが ID テーブルから削除されたこと

 **Note:** 残りの DIB エントリまたは保留中のニュークリアスセッションの自動再スタートは、初期 RESTORE ステップで自動処理されるため無視できます。

- すべての必須のデータベースコンポーネント (ASSO、DATA など) が、少なくとも 1 回はフォーマットされていること
- ■ リカバリする世代中で変更されたすべてのコンポーネントを、世代の開始時点で有効なサイズおよびデバイスタイプに割り当て、フォーマットします。

- サイズが変更されたすべてのコンポーネントを、リカバリする世代中に使用された最大サイズに割り当て、フォーマットします。

ファイルレベルのリカバリの場合は、次のことがチェックされます。

- ニュークリアスがアクティブになっていること ADARAI によって作成されたリカバリジョブは、ニュークリアスを自動的に開始しません。

## RECOVER 機能またはリカバリジョブストリームの再スタート

RECOVER 機能では RLOG を読み込むのみであり、変更を加えないため、ADARAI RECOVER 機能が中断された場合には、最初から再スタートできます。

通常のリストア／再生成処理と同じように、RECOVER 機能で作成された DD/JCLOUT リカバリジョブストリームを再スタートできます。ただし、ジョブストリームを編集し、正常に完了したユーティリティ操作のステップを削除する必要がある場合があります。この操作の後、（中断の原因が取り除かれた場合は）失敗したユーティリティ操作から開始してリカバリ処理を続行できます。

中断されたリカバリジョブは、常に最初から再スタートできます。また、エラー原因およびエラーを含むジョブステップによっては、失敗したジョブステップまたは少し前のステップからリカバリジョブを再スタートすることもできます。



# 146

## REMOVE : Recovery Aid の削除

---

- 例 ..... 774

ADARAI REMOVE は機能的には以前の ADARAI NORAI 機能と同じであり、REMOVE または NORAI のどちらかを指定できます。



**Note:** ADARAI REMOVE/NORAI は、データベースが非アクティブの状態で行う必要があります。

### ADARAI REMOVE

ADARAI REMOVE 機能は、リカバリロギング（つまり Recovery Aid）がデータベースで使用できずその情報が RLOG に記録されなくなることを示すように、アソシエータ GCB を更新することによってリカバリロギングを無効にします。

既存の RLOG 情報は保持され、次の PREPARE 操作が実行されるまで REMOVE 実行後の表示またはリカバリ操作に使用できます。ADARAI PREPARE 機能を実行すると、既存の RLOG データがすべて失われます。

REMOVE 機能の使用後にリカバリロギングを再スタートするには、ADARAI PREPARE 機能を実行してから、ADASAV SAVE/RESTORE データベース、RESTORE GCB、および（または）SAVE DELTA/RESTORE DELTA（データベース）機能を実行して新しい世代を開始します。RLOG の準備については、[ADARAI PREPARE](#) の説明を参照してください。

## 例

---

ADARAI REMOVE

Recovery Aid ロギングをすべて停止します。

# 147 JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	776
▪ z/OS .....	792
▪ z/VM .....	798
▪ VSE .....	800

ここでは、BS2000、z/OS、z/VM、および VSE システムにおける ADARAI の実行に必要なジョブ制御情報について説明し、各ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

ここでは、BS2000 システムで ADARAI を使用する場合の追加考慮事項および要件について説明します。

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	RECOVER に必要
ワーク	DDWORKR1 DDWORKR4	ディスク	RECOVER に必要
データプロテクションログ	DDPLOGRn	テープ/ディスク	RECOVER に必要
リカバリログ (RLOG)	DDRLOGR1	ディスク	
ジョブストリーム入力	DDJCLIN	ディスク	RECOVER に必要
リカバリジョブ出力	DDJCLOUT	ディスク	RECOVER に必要
リカバリジョブ出力 (コンソールサブタスクの JCL)	DDJCLCON	ディスク	オプションであり、RECOVER のみに使用
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADARAI パラメータ	SYSDTA/ DDKARTE		オペレーションマニュアル
ADARUN メッセージ	SYSOUT DDPRINT		メッセージおよびコード
ADARAI メッセージ	SYSLST DDDRUCK		メッセージおよびコード

### BS2000 のデータセット

シーケンシャルディスクデータセットの操作

共用ディスクに存在するシーケンシャルデータセットに関する制限はありません。

専用ディスクに存在するシーケンシャルデータセットはエクスポートできません。このことは、カタログエントリは消去できず、専用ディスク上のデータセットに対する /IMPORT-FILE コマンドをリカバリジョブ制御に含めることができないことを意味します。

シーケンシャルテープデータセットの操作

テープに存在するシーケンシャルデータセットのカタログエントリは、システムのカタログから削除しないことをお勧めします。RECOVERによってシーケンシャルデータセットのカタログエントリがテープまたはカートリッジで検出されないと、ジョブストリームに次のステートメントが取り込まれます。

```
/IMPORT-FILE FILE-NAME=tempfile ,...
```

ここで *tempfile* は次の構造になっています。

```
#ADARAI.RECOVER.TAPE.nnnnn
```

### ファイル世代グループ (FGG) の使用

ディスクまたはテープにかかわらず、シーケンシャルデータセットに対して FGG を使用する場合には、制限はありません。

### 入力データセット

ADARAI は、元のカタログエントリを使用して BS2000 のシーケンシャルデータセットにアクセスしようとしています。シーケンシャルデータセットがファイル世代グループ (FGG) に存在すると、ADARAI では絶対メンバを割り当てますが、ベースポインタを使用または変更することはありません。

シーケンシャルデータセットがパスワードによって読み込み保護されている場合は、スケルトンジョブ制御の %%JCL-JOB-HEADER セクションにパスワードが含まれている必要があります。

ADARAI はデータセットのエクスポートおよび消去を行いません。テープデータセットのカタログエントリが存在しないと、ADARAI は #ADARAI.RECOVER.TAPE.nnnnn という名前で中間データセットを作成します。ここで、*nnnnn* は "00001"、"00002" などです。

## ADARAI JCL の例 (BS2000)

### リカバリロギングの開始 (ADARAI PREPARE)

SDF フォーマットの場合：

```
/.ADARAI LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK *A D A R A I   START RECOVERY LOGGING
/REMARK *
/ASS-SYSLST L.RAI.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,DB.yyyyyy.ASSO
/SET-FILE-LINK DDRLOGR1,DB.yyyyyy.RLOGR1,OPEN-MODE=OUTIN,BUFF-LEN=STD(2)
```

```
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN  PROG=ADARAI,DBID=yyyyy,MODE=MULTI
ADARAI  PREPARE RLOGSIZE=5,RLOGDEV=dddd,MINGENS=5
/LOGOFF SYS-OUTPUT=DEL
```

**ISP** フォーマットの場合：

```
/.ADARAI LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK *A D A R A I   START RECOVERY LOGGING
/REMARK *
/REMARK *
/SYSFILE SYSLST=L.RAI.DATA
/FILE ADAvrS.MOD,LINK=DDLIB
/FILE DB.yyyyy.ASSO ,LINK=DDASSOR1
/FILE DB.yyyyy.RLOGR1,LINK=DDRLOGR1,OPEN=OUTIN,BLKSIZE=(STD,2)
/EXEC (ADARUN,ADAvrS.MOD)
ADARUN  PROG=ADARAI,DBID=yyyyy,MODE=MULTI
ADARAI  PREPARE RLOGSIZE=5,RLOGDEV=dddd,MINGENS=5
/LOGOFF NOSPOOL
```

**RLOG** のリスト (ADARAI LIST)

**SDF** フォーマットの場合：

```
/.ADARAI LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK *A D A R A I   LIST RECOVERY LOGS
/REMARK *
/ASS-SYSLST L.RAI.DATA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrS.MOD
/SET-FILE-LINK DDASSOR1,DB.yyyyy.ASSO
/SET-FILE-LINK DDRLOGR1,DB.yyyyy.RLOGR1
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN  PROG=ADARAI,DBID=yyyyy,MODE=MULTI
ADARAI  LIST GENS=NO,RLOGDEV=dddd,RELGEN=1
/LOGOFF SYS-OUTPUT=DEL
```

**ISP** フォーマットの場合：

```

/.ADARAI LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK *A D A R A I LIST RECOVERY LOGS
/REMARK *
/REMARK *
/SYSFILE SYSLST=L.RAI.DATA
/FILE ADAvrs.MOD,LINK=DDLIB
/FILE DB.yyyyyy.ASSO ,LINK=DDASSOR1
/FILE DB.yyyyyy.RLOGR1,LINK=DDRLOGR1
/EXEC (ADARUN,ADAvrs.MOD)
ADARUN PROG=ADARAI,DBID=yyyyyy,MODE=MULTI
ADARAI LIST GENS=NO,RLOGDEV=dddd,RELGEN=1
/LOGOFF NOSPOOL

```

## リカバリ JCL の作成 (ADARAI RECOVER)

SDF フォーマットの場合：

```

/.ADARAI LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK *A D A R A I BUILD RECOVERY JCL STREAM
/REMARK *
/DELETE-FILE DB.yyyyyy.JCLOUT
/SET-JOB-STEP
/CREATE-FILE DB.yyyyyy.JCLOUT,PUB(SPACE=(48,48))
/SET-JOB-STEP
/DELETE-FILE DB.yyyyyy.JCLCON
/SET-JOB-STEP
/CREATE-FILE DB.yyyyyy.JCLCON,PUB(SPACE=(48,48))
/SET-JOB-STEP
/ASS-SYSLST L.RAI.DATA
/ASS-SYSDTA *SYSCMD

/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,DB.yyyyyy.ASSO
/SET-FILE-LINK DDRLOGR1,DB.yyyyyy.RLOGR1
/SET-FILE-LINK DDPLOGR1,DB.yyyyyy.PLOGR1
/SET-FILE-LINK DDPLOGR2,DB.yyyyyy.PLOGR2
/SET-FILE-LINK DDJCLIN,DB.yyyyyy.JCLIN
/SET-FILE-LINK DDJCLOUT,DB.yyyyyy.JCLOUT
/SET-FILE-LINK DDJCLCON,DB.yyyyyy.JCLCON
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARAI,DBID=yyyyyy,MODE=MULTI
ADARUN UEX2=EXITR2

```

```
ADARAI RECOVER PLOGDEV=2201, FEOFPL=YES, RELGEN=1  
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADARAI LOGON  
/OPTION MSG=FH, DUMP=YES  
/REMARK *  
/REMARK *A D A R A I    BUILD RECOVERY JCL STREAM  
/REMARK *  
/REMARK *  
  
/SYSFILE SYSLST=L.RAI.DATA  
/FILE ADAvrs.MOD, LINK=DDLIB  
/FILE DB.yyyyyy.ASSO, LINK=DDASSOR1  
/FILE DB.yyyyyy.RLOGR1, LINK=DDRLOGR1  
/FILE DB.yyyyyy.PLOGR1, LINK=DDPLOGR1  
/FILE DB.yyyyyy.PLOGR2, LINK=DDPLOGR2  
/FILE DB.yyyyyy.JCLIN, LINK=DDJCLIN  
/FILE DB.yyyyyy.JCLOUT, LINK=DDJCLOUT  
/FILE DB.yyyyyy.JCLCON, LINK=DDJCLCON  
/EXEC (ADARUN, ADAvrs.MOD)  
ADARUN PROG=ADARAI, DBID=yyyyyy, MODE=MULTI  
ADARUN UEX2=EXITR2  
ADARAI RECOVER PLOGDEV=2201, FEOFPL=YES, RELGEN=1  
/LOGOFF NOSPOOL
```

### スケルトンジョブ制御

BS2000 システムの場合、スケルトンジョブ制御には次の特性があります。

- 最初の 2 つの位置は空白にする必要があります。
- ステートメントの最大長は 256 文字であり、最初の 2 つの位置にある空白も含まれます。
- ステートメントは継続でき、継続マーク (-) は任意の位置に配置できます（位置 74 には限定されません）。
- バージョン 10 の ADARAI では、次のコマンドが自動的に追加されます。

```
/MODIFY-SDF-OPTIONS CONTINUATION=NEW-MODE
```

- BS2000 Executive のジョブ制御ステートメントは、必要に応じて 72 バイトのセグメントに自動的に分割されます。通常のプログラムコントロールステートメントはセグメント化されません。
- 文字列置換（下の説明を参照）の場合を除き、ADARAI はジョブ制御ステートメントの構文をチェックまたは変更しません。

上記の一般的な JCL 特性に加え、BS2000 システムではスケルトン JCL に次のものを含めることができます。

- 頻繁に使用されるジョブ制御入力を繰り返すための置換文字列
- DDFILEA データセットの上書き、テープボリュームの割り当て、およびディスクスペースクレジット
- RECOVER で生成されるコンソールメッセージのサブタスク処理

以上のオプションは、スケルトン JCL の先頭で %%JCL-JOB-HEADER ステートメントの直前に指定する必要があります。

#### 置換文字列の指定

データセット名などの頻繁に使用する文字列は、置換セクションで定義できます。定義された文字列は、ユーザプログラムのコントロールステートメントだけでなく BS2000 の JCL にも挿入されます。

文字列置換を使用するには、次の JCL ステートメントを %%JCL-JOB-HEADER の前に配置します。

```
%%JCL-BS2-SUBSTITUTION
```

このステートメントの後に、次の形式の置換定義が続きます。

```
%%argname=substring
```

ここで、*argname* は置換される 1~8 文字の JCL または ユーザプログラムの制御引数であり、*substring* は 128 文字以内の置換文字列です。1 行あたり 1 つの置換ステートメントが有効であり、次の %%JCL ステートメントが出現すると、置換ステートメントは終了します。

#### DDFILEA JCL オプション

ここで説明している JCL ステートメントは、スケルトン JCL の %%JCL-JOB-HEADER の前に配置する必要があります。

ADAORD は、BS2000 リカバリジョブの途中で DDFILEA 出力データセットをオープンする唯一のユーティリティです。ADARAI は、データセットが RECOVER 実行中に存在したかどうかにかかわらず、データセットをディスクまたはテープに割り当てることを支援します。

既存のテープデータセットが上書きされることはありません。既存の DDFILEA ディスクデータセットを上書きするには、スケルトン JCL で次のステートメントを指定します。

```
%%JCL-BS2-WORK-DATASET-OVERWRITE=YES
```

OVERWRITE のデフォルトは NO です。DDFILEA 割り当てが検出されず上書きが禁止されている（デフォルト）と、ADARAI では DDFILEA データセットを TAPE-C1 カートリッジに書き込もうとします。

DDFILEA をテープではなくディスクに割り当てするには、次の JCL ステートメントを指定します。

```
%%JCL-BS2-WORK-DISK-SPACE
```

次に、1 つ以上のディスクスペース割り当てステートメントを次のように指定します。

```
:catid :=pam-pages
```

ADARAI では、指定された PAM ページ数をパブセットに割り当てるために、パブセットの可用性および ADARAI タスクのログオンユーザー ID をチェックします。チェックに失敗した場合は、関連するパブセットの結合ファイルにユーザーのログオン ID を追加し、生成されたりカバリジョブを開始する前に ADARAIRECOVER ジョブステップを繰り返す必要があります。この操作を行わないと、ADAORD ジョブステップがアベンドする可能性があります。

DDFILEA 出力テープデバイスおよびボリュームを割り当てるには、次のステートメントを指定します。

```
%%JCL-BS2-WORK-TAPE-VOLUMES=device-type
```

出力テープデータセットを作成する必要がある場合、ADARAI は #ADARAI.RECOVER.TAPE nnnnn という名前の中間データセットを使用します。ここで、nnnnn は "00001"、"00002" などです。次のセクションで説明するコンソールサブタスクオプションが使用可能になっている場合は、各行に 1 つ以上の volser 番号を空白またはコンマで区切って指定できます。各行の最初の 2 つの位置は空白にする必要があり、行の最大長は 256 文字です。

次に例を示します。

```
%%JCL-BS2-WORK-TAPE-VOLUMES=TAPE-C1
  A00001,A00002,A00003,A00004
  A00005,A00006,A00007,A00008
```

512 ボリュームまで指定できます。

**BS2000** のコンソールサブタスク

ADARAI RECOVER で作成されたりリカバリジョブをサブミットすると、オペレータの介入を必要とする次のようなコンソールメッセージが表示される場合があります。

- テープデータセットのカatalogエントリが使用できず、中間データセットに対して ADARAI が /IMPORT-FILE ステートメントを挿入したため、DMS0DA5 コンソールメッセージが表示されました。
- DDFILEA データセットにテープ出力が必要であり、ADARAI が JCL を取り込んだため、スクラッチテープのマウントを要求するメッセージが表示されました。

これらのメッセージには、リカバリジョブでサブタスクとして並列実行される UCON プログラムで自動的に対応できます。UCON プログラム (ADAR2C) は、関係するコンソールメッセージをすべて受信し、可能な限り応答を送信します。

コンソールサブタスクを実行するには、%%JCL-JOB-HEADER の前に次のセクションを指定します。

```
%%JCL-BS2-CONSOLE-SUBTASK-SPECIFICATION
CONSOLE-NAME=name,C'password'          (this statement is required)
DCAM-APPL=dcamappl                      (this statement is optional)
```

ここでは次の内容を表しています。

<i>name</i>	/LOGON ユーザー ID
<i>password</i>	ユーザー ID のログオンパスワード
<i>dcamappl</i>	DCAM アプリケーション名 (デフォルトは "RAIRUCON")

詳細は、『BS2000 Systems Administration』の「*Authorized User Tasks*」を参照してください。

次のオプションのステートメントをスケルトン JCL に含め、その後に *volser* 定義を指定することができます。コンソールサブタスクがコールされない場合、*volser* 定義は影響を及ぼしません。

```
%%JCL-BS2-WORK-TAPE-VOLUMES=device-type
```

次のステートメントを ADARAI RECOVER 機能の JCL (スケルトン JCL ではない) に含めません。

```
/SET-FILE-LINK  
FILE-NAME=console-job, LINK-NAME=DDJCLCON
```

次の RECOVER 機能の JCL 例には、コンソールサブタスクが含まれています。

```
/ LOGON  
/ SET-FILE-LINK FILE-NAME=ADA.JCLIN, LINK-NAME=DDJCLIN  
/ SET-FILE-LINK FILE-NAME=ADA.JCLOUT, LINK-NAME=DDJCLOUT  
/ SET-FILE-LINK FILE-NAME=ADA.JCLCON, LINK-NAME=DDJCLCON  
.  
.  
  
/ ASSIGN-SYSDTA TO-FILE=*SYSCMD  
/ SET-FILE-LINK ADAvrs.MOD, LINK-NAME=DDLIB  
/ START-PROG *M(E=ADARUN, L=ADAvrs.MOD)  
ADARUN PROG=ADARAI, DB=47  
ADARAI RECOVER  
/ LOGOFF NOSPOOL
```

コンソールサブタスクは、次の BS2000 コンソールメッセージを評価します。ここで、*tsn* はタスクのシリアル番号、*mn* は簡略テクニカルデバイス名です。

### NKVT010 VOLUME volser IS MOUNTED ON DEVICE mn

volser およびデバイスが「[DDFILEA JCL オプション](#)」の説明に従って指定されていると、サブタスクのオンライン volser テーブルに登録されます。この時点でメッセージ NKVT013 も未解決の場合は、サブタスクからレスポンス "tsn.mn" が返されます。

### NKVT011/97 VOLUME volser IS DISMOUNTED FROM DEVICE mn...

volser はオンライン volser テーブルから削除されます。

### NKVT013 MOUNT TAPE '\*SCRAT' ON DEVICE ...

このメッセージが RECOVER タスクに関係しており、volser がオンライン volser テーブルで使用可能な場合は、サブタスクからレスポンス "tsn.mn" が送られます。

### DMS0DFB ACKNOWLEDGE VSN volser ON DEVICE mn...

このメッセージが RECOVER タスクに関係している場合は、サブタスクからレスポンス "tsn" が送られます。

**DMS0DA5 INVALID FILE SPECIFICATION: VSN volser FOR FILE file...**

関連する "tsn" が RECOVER タスクの "tsn" で、*file* が #ADARAI.RECOVER.TAPE.nnnnn という名前の中間データセットの場合は、メッセージがレスポンス "tsn.I" で応答されます。

サブタスクから応答するには、誤ったファイル ID を持つテープをログオン ID が無視できる必要があります。このことを可能にするには、システム管理者のログオン ID (TSOS) を使用して、次のステートメントを発行します。

```
/MOD-USER userid,PROTECTION-ATTRIBUTE=(TAPE-ACCESS=READ)
```

**スケルトンジョブ制御の例 (BS2000)**

```
%%JCL-BS2-WORK-DATASET-OVERWRITE = NO
%%JCL-BS2-WORK-DISK-SPACE
    :A: = 500000
    :B: = 1000000
%%JCL-BS2-CONSOLE-SUBTASK-SPECIFICATION
    CONSOLE-NAME=CON1,C'PASSWORD'
%%JCL-BS2-WORK-TAPE-VOLUMES=TAPE-C1
    A00001,A00002,A00003,A00004,A00005
    A00006,A00007,A00008,A00009,A00010
%%JCL-BS2-SUBSTITUTION
%%USERID = ADAvr
%%AC %%ASSOR1 = $ADAvr.DByyyyy.ASSOR1
%%ASSOR2 = $ADAvr.DByyyyy.ASSOR2
%%DATAR1 = $ADAvr.DByyyyy.DATAR1
%%DATAR2 = $ADAvr.DByyyyy.DATAR2
%%WORK = $ADAvr.DByyyyy.WORK
%%SORT = $ADAvr.DByyyyy.SORT%
%%TEMP = $ADAvr.DByyyyy.TEMP
%%RLOGR1 = $ADAvr.DByyyyy.RLOGR1
%%PLOGR1 = $ADAvr.DByyyyy.PLOGR1
%%PLOGR2 = $ADAvr.DByyyyy.PLOGR2
%%DBID = yyyyy
%%IDTNAME = ADABASvB
%%USEREX = ADAvr.s.USEREXITS.MOD
%%DDLIB = ADAvr.s.MOD
%%DEVICE = dddd
%%JCL-JOB-HEADER
/.RECOVER LOGON %%USERID,%%ACCOUNT
/ SET-FILE-LINK FILE-NAME=%%USEREX,LINK-NAME=BLSLIB01
/ SET-FILE-LINK FILE-NAME=%%DDLIB,LINK-NAME=DDLIB
/ ASSIGN-SYSOUT TO-FILE=$ADAvv.RAI.OUT
/ ASSIGN-SYSLST TO-FILE=$ADAvv.RAI.LST%%JCL-JOB-TRAILER
/REMARK -----*
/REMARK * RECOVERY JOB SUCCESSFULLY TERMINATED *
/REMARK -----*
/LOGOFF NOSPOOL
```

```

/ .JOBERROR REMARK
/REMARK ++++++*
/REMARK *          RECOVERY JOB TERMINATED WITH ERROR          *
/REMARK ++++++*
/LOGOFF NOSPOOL
%%JCL-ADARES
/REMARK
/REMARK REGENERATE/ BACKOUT
/REMARK
/      SET-FILE-LINK FILE-NAME=%RLOGR1, LINK-NAME=DDRLOGR1,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%ASSOR1, LINK-NAME=DDASSOR1,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%ASSOR2, LINK-NAME=DDASSOR2,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%DATAR1, LINK-NAME=DDDATAR1,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%DATAR2, LINK-NAME=DDDATAR2,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%WORK, LINK-NAME=DDWORKR1,        -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADARES, LINK-NAME=DDDRUCK
/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADARES, LINK-NAME=DDPRINT
%SEQUENTIAL
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE ( ELEMENT=ADARUN, LIBRARY=%DDLIB )
ADARUN MODE=MULTI, PROG=ADARES
ADARUN DBID=%DBID, DE=%DEVICE, IDTNAME=%IDTNAME
%KARTE
%%JCL-STEP-TRAILER
/SET-JOB-STEP%%JCL-ADASAV
/REMARK
/REMARK RESTORE FILE(S)/DATABASE
/REMARK
/ .%STEP REMARK
/      SET-FILE-LINK FILE-NAME=%RLOGR1, LINK-NAME=DDRLOGR1,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%ASSOR1, LINK-NAME=DDASSOR1,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%ASSOR2, LINK-NAME=DDASSOR2,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%DATAR1, LINK-NAME=DDDATAR1,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%DATAR2, LINK-NAME=DDDATAR2,      -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%WORK, LINK-NAME=DDWORKR1,        -
/                          SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK
FILE-NAME=RAI.DRU.JCLO.RESTORE, LINK-NAME=DDDRUCK
/      SET-FILE-LINK
FILE-NAME=RAI.PRI.JCLO.RESTORE, LINK-NAME=DDPRINT

```

```

%SEQUENTIAL
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE(ELEMENT=ADARUN,LIBRARY=%DDLIB)
ADARUN MODE=MULTI,PROG=ADASAV
ADARUN DBID=%DBID,DE=%DEVICE,IDTNAME=%IDTNAME
%KARTE
%%JCL-ENDNUC
/REMARK
/REMARK ADADBS END NUCLEUS

/REMARK
/      SET-FILE-LINK FILE-NAME=%RLOGR1,LINK-NAME=DDRLOGR1,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%ASSOR1,LINK-NAME=DDASSOR1,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%ASSOR2,LINK-NAME=DDASSOR2,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%DATAR1,LINK-NAME=DDDATAR1,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%DATAR2,LINK-NAME=DDDATAR2,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%WORK,LINK-NAME=DDWORKR1,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%SORT,LINK-NAME=DDSORTR1
/      SET-FILE-LINK FILE-NAME=%TEMP,LINK-NAME=DDTEMPR1
/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADADBS,LINK-NAME=DDDRUCK
/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADADBS,LINK-NAME=DDPRINT
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE(ELEMENT=ADARUN,LIBRARY=%DDLIB)
ADARUN MODE=MULTI,PROG=ADADBS
ADARUN DBID=%DBID,DE=%DEVICE,IDTNAME=%IDTNAME
ADADBS OPERCOM ADAEND
/REMARK
/REMARK CHECK INACTIVE DATABASE
/REMARK
/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADARAI,LINK-NAME=DDDRUCK
/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADARAI,LINK-NAME=DDPRINT
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE(ELEMENT=ADARUN,LIBRARY=%DDLIB)
ADARUN
MODE=MULTI,PROG=ADARAI,DBID=%DBID,DE=%DEVICE,IDTNAME=%IDTNAME
ADARAI CHKDB INACTIVE
%%JCL-ADADEF
/REMARK
/REMARK DEFINE NEWWORK
/REMARK
/      SET-FILE-LINK FILE-NAME=%RLOGR1,LINK-NAME=DDRLOGR1,      -
/                          SUP=DISK(SHARE-UPD=YES)
/      SET-FILE-LINK FILE-NAME=%ASSOR1,LINK-NAME=DDASSOR1,      -

```

```

/
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%ASSOR2 , LINK-NAME=DDASSOR2 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%DATAR1 , LINK-NAME=DDDATAR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%DATAR2 , LINK-NAME=DDDATAR2 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%WORK , LINK-NAME=DDWORKR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%SORT , LINK-NAME=DDSORTR1
/      SET-FILE-LINK FILE-NAME=%%TEMP , LINK-NAME=DDTEMPR1
/      SET-FILE-LINK FILE-NAME=RAI . DRU . JCLO . ADADEF , LINK-NAME=DDDRUCK
/      SET-FILE-LINK FILE-NAME=RAI . PRI . JCLO . ADADEF , LINK-NAME=DDPRINT
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE ( ELEMENT=ADARUN , LIBRARY=%%DDLIB )
ADARUN MODE=MULTI , PROG=ADAEF
ADARUN DBID=%%DBID , DE=%%DEVICE , IDTNAME=%%IDTNAME
%KARTE
%%JCL-ADALOD
/REMARK
/REMARK LOAD A FILE/ MASS UPDATE
/REMARK
/      SET-FILE-LINK FILE-NAME=%%RLOGR1 , LINK-NAME=DDRLOGR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%ASSOR1 , LINK-NAME=DDASSOR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%ASSOR2 , LINK-NAME=DDASSOR2 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%DATAR1 , LINK-NAME=DDDATAR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%DATAR2 , LINK-NAME=DDDATAR2 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%WORK , LINK-NAME=DDWORKR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )
/      SET-FILE-LINK FILE-NAME=%%SORT , LINK-NAME=DDSORTR1
/      SET-FILE-LINK FILE-NAME=%%TEMP , LINK-NAME=DDTEMPR1
/      SET-FILE-LINK FILE-NAME=RAI . DRU . JCLO . ADALOD , LINK-NAME=DDDRUCK
/      SET-FILE-LINK FILE-NAME=RAI . PRI . JCLO . ADALOD , LINK-NAME=DDPRINT
%SEQUENTIAL
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE ( ELEMENT=ADARUN , LIBRARY=%%DDLIB )
ADARUN MODE=MULTI , PROG=ADALOD
ADARUN DBID=%%DBID , DE=%%DEVICE , IDTNAME=%%IDTNAME
%KARTE
%%JCL-ADAORD
/REMARK
/REMARK REORDER
/REMARK
/      SET-FILE-LINK FILE-NAME=%%RLOGR1 , LINK-NAME=DDRLOGR1 ,      -
/      SUP=DISK ( SHARE-UPD=YES )

```

```

/      SET-FILE-LINK FILE-NAME=%%ASSOR1, LINK-NAME=DDASSOR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%ASSOR2, LINK-NAME=DDASSOR2,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%DATAR1, LINK-NAME=DDDATAR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%DATAR2, LINK-NAME=DDDATAR2,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%WORK, LINK-NAME=DDWORKR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%SORT, LINK-NAME=DDSORTR1
/
/      SET-FILE-LINK FILE-NAME=%%TEMP, LINK-NAME=DDTEMPR1
/
/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADAORD, LINK-NAME=DDDRUCK
/
/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADAORD, LINK-NAME=DDPRINT
%%SEQUENTIAL
/
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/
/      START-PROGRAM
FROM-FILE=*MODULE ( ELEMENT=ADARUN, LIBRARY=%%DDLIB )
ADARUN MODE=MULTI, PROG=ADAORD
ADARUN DBID=%%DBID, DE=%%DEVICE, IDTNAME=%%IDTNAME
%KARTE
%%JCL-ADAINV
/REMARK
/REMARK INVERT/ COUPLE
/REMARK
/
/      SET-FILE-LINK FILE-NAME=%%RLOGR1, LINK-NAME=DDRLOGR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%ASSOR1, LINK-NAME=DDASSOR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%ASSOR2, LINK-NAME=DDASSOR2,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%DATAR1, LINK-NAME=DDDATAR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%DATAR2, LINK-NAME=DDDATAR2,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%WORK, LINK-NAME=DDWORKR1,      -
/
/      SUP=DISK ( SHARE-UPD=YES )
/
/      SET-FILE-LINK FILE-NAME=%%SORT, LINK-NAME=DDSORTR1
/
/      SET-FILE-LINK FILE-NAME=%%TEMP, LINK-NAME=DDTEMPR1
/
/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADADBS, LINK-NAME=DDDRUCK
/
/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADADBS, LINK-NAME=DDPRINT
/
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/
/      START-PROGRAM
FROM-FILE=*MODULE ( ELEMENT=ADARUN, LIBRARY=%%DDLIB )
ADARUN MODE=MULTI, PROG=ADAINV
ADARUN DBID=%%DBID, DE=%%DEVICE, IDTNAME=%%IDTNAME
%KARTE
%%JCL-STARTNUC
/REMARK
/REMARK START NUCLEUS
/REMARK
/
/      MODIFY-JOB-SWITCHES ON=( 4, 5 )

```

```

/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM FROM-FILE=EDT
:/.ADANUC LOGON %%USERID,%%ACCOUNT
:/      SET-FILE-LINK FILE-NAME=%%RLOGR1, LINK-NAME=DDRLOGR1, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%PLOGR1, LINK-NAME=DDPLOGR1, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%PLOGR2, LINK-NAME=DDPLOGR2, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%ASSOR1, LINK-NAME=DDASSOR1, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%ASSOR2, LINK-NAME=DDASSOR2, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%DATAR1, LINK-NAME=DDDATAR1, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%DATAR2, LINK-NAME=DDDATAR2, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=%%WORK, LINK-NAME=DDWORKR1, -
:/      SUP=DISK (SHARE-UPD=YES)
:/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADANUC, LINK-NAME=DDDRUCK
:/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADANUC, LINK-NAME=DDPRINT
:/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
:/      SET-FILE-LINK FILE-NAME=%USEREX, LINK-NAME=BLSLIB01
:/      SET-FILE-LINK FILE-NAME=%DDLIB, LINK-NAME=DDLIB
:/      START-PROGRAM FROM-FILE=*MODULE (ELEMENT=ADARUN, LIBRARY=%%DDLIB)
:ADARUN PROG=ADANUC
:ADARUN DATABASE=%%DBID          DATA BASE ID
:ADARUN IDTNAME=%%IDTNAME        NAME OF IDT
:ADARUN DEVICE=%%DEVICE
:ADARUN IDTNAME=%%IDTNAME
:ADARUN LFIOP=1000000, LWP=800000, LBP=4000000, LU=64000
:ADARUN LP=200, LS=20000, TT=900, TNAE=900, OPENRQ=NO, PLOGRQ=YES
:ADARUN NAB=20, NH=400, NU=200, NISNHQ=50, NC=20, NPLOG=2, PLOGDEV=2000
:ADARUN PLOGSIZE=1000, NCLOG=2, CLOGDEV=2000, CLOGSIZE=500
:ADARUN UEX2=USEREX2
:/LOGOFF SYSOUT=DEL
@D &:1-1:
@W '#NUCENT' O
@H
/STEP
/ENTER-JOB #NUCENT, PROC-ADMIS=PAR (USER-ID=%%USERID,
/                                     ACCOUNT=%%ACCOUNT)
/STEP
/      SET-FILE-LINK FILE-NAME= %%DDLIB, LINK-NAME=DDLIB
/      SET-FILE-LINK FILE-NAME=RAI.DRU.JCLO.ADARES, LINK-NAME=DDDRUCK
/      SET-FILE-LINK FILE-NAME=RAI.PRI.JCLO.ADARES, LINK-NAME=DDPRINT
/      ASSIGN-SYSDTA TO-FILE=*SYSCMD
/      START-PROGRAM
FROM-FILE=*MODULE (ELEMENT=ADARUN, LIBRARY=%%DDLIB)
ADARUN MODE=MULTI, PROG=ADARAI

```

```
ADARUN DBID=%DBID,DE=%DEVICE, IDTNAME=%IDTNAME
ADARAI CHKDB ACTIVE
```

## ADAR2E ユーティリティ

ヘッドクラッシュなどの回復不可能なハードウェアエラーなどが原因でパブセットメンバが失われた場合は、新しいメンバを初期化してパブセットに取り込み、パブセット内のすべてのデータセットをリストアします。ほとんどの場合、ASSO、DATA、WORK、およびPLOGデータセットを、ハードウェア障害が発生する前と同じディスク位置に再割り当てする必要があります。

ADAR2E ユーティリティは、割り当てられたデータセットの TSOSCAT エントリを読み込み、Adabas ファイルを前と同じディスク位置に再割り当てするプロシージャを作成します。ADAR2E ユーティリティの動作は、ADARAI および他の Adabas ユーティリティから完全に独立しています。Adabas の主要コンポーネント (ASSO、DATA、WORK、PLOG、CLOG、TEMP、または SORT) の割り当てが変更するたびに、ADAR2E を実行する必要があります。

次の JCL 例は、ADAR2E ユーティリティを実行するためのものです。

```
/ LOGON
/ SET-FILE-LINK FILE-NAME=DB.yyyyyy.ASSO-01, LINK-NAME=DDASSOR1
/ SET-FILE-LINK FILE-NAME=DB.yyyyyy.DATA-01, LINK-NAME=DDDATAR1
/ SET-FILE-LINK FILE-NAME=DB.yyyyyy.WORK-01, LINK-NAME=DDWORKR1
/ SET-FILE-LINK FILE-NAME=DB.yyyyyy.TEMP-01, LINK-NAME=DDTEMPR1
/ SET-FILE-LINK FILE-NAME=DB.yyyyyy.PLOGR1, LINK-NAME=DDPLOGR1
/ SET-FILE-LINK FILE-NAME=DB.yyyyyy.PLOGR2, LINK-NAME=DDPLOGR2
/ SET-FILE-LINK FILE-NAME=ALLOCATE.JOB, LINK-NAME=DDJCLOUT
/ START-PROGRAM *MOD(LIB=ADAvrs, ELEM=ADAR2E)
/ LOGOFF SYS-OUTPUT=DEL
```

次に、使用可能なリンク名のリストを示します。

```
DDASSOR1, DDASSOR2, DDASSOR3, DDASSOR4, DDASSOR5
DDDATAR1, DDDATAR2, DDDATAR3, DDDATAR4, DDDATAR5
DDWORKR1, DDWORKR2
DDSORTR1, DDSORTR2
```

```
DDTEMPR1, DDTEMPR2
DDPLOGR1, DDPLOGR2, DDPLOGR3, DDPLOGR4, DDPLOGR5, DDPLOGR6, DDPLOGR7, DDPLOGR8
```

次に、ADAR2E ユーティリティで作成されるジョブ制御の例を示します。

```
/ BEGIN-PROCEDURE
/ MODIFY-SDF-OPTIONS CONTINUATION=NEW-MODE
/ CREATE-FILE FILE-NAME=:A:$ADABAS.DB.yyyyyy.ASSO-01,-
/   SUPPORT=PUBLIC-DISK(VOLUME=PUBA00,-
/   DEVICE-TYPE=Ddddd,SPACE=ABSOLUTE-
/   (FIRST-PAGE=32833,SIZE=5001))
/ MODI-FILE-ATTR FILE-NAME=:A:$ADABAS.DB.yyyyyy.ASSO-01,-
/   SUPPORT=PUBLIC-DISK(VOLUME=PUBA00,-
/   DEVICE-TYPE=Ddddd,SPACE=ABSOLUTE-
/   (FIRST-PAGE=29761,SIZE=5001))
/   .
/   .
/ CREATE-FILE FILE-NAME=:A:$ADABAS.DB.yyyyyy.DATA-01,-
/   SUPPORT=PUBLIC-DISK(VOLUME=PUBA02,-
/   DEVICE-TYPE=Ddddd,SPACE=ABSOLUTE-
/   (FIRST-PAGE=119809,SIZE=8001))
/ MODI-FILE-ATTR FILE-NAME=:A:$ADABAS.DB.yyyyyy.DATA-01,-
/   SUPPORT=PUBLIC-DISK(VOLUME=PUBA03,-
/   DEVICE-TYPE=Ddddd,SPACE=ABSOLUTE-
/   (FIRST-PAGE=75841,SIZE=8001))
```

ADARAIRECOVER 機能を使用する前に、システム管理者のログオン ID (TSOS) を使用して、ADAR2E で生成された再割り当てプロシージャを実行します。

BS2000 バージョン 11 以上では、TSOS を使用して次のコマンドを発行することにより、特定のサブセットに対してディスクスペースの割り当て権限をすべてのユーザーに付与することができます。

```
/MODIFY-MASTER-CATALOG CATID= catid,PHYSICAL-ALLO=USER-ALLOWED
```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データプロテクションログ	DDPLOGRn	テープ/ディスク	FEOFPL=YES と指定された場合にのみ RECOVER に必要
リカバリログ (RLOG)	DDRLOGR1	ディスク	
ジョブストリーム入力	DDJCLIN	ディスク	RECOVER のみに必要

データセット	DD名	ストレージ	追加情報
リカバリジョブ出力	DDJCLOUT	ディスク	RECOVER のみに必要
ADARAI メッセージ	DDDRUCK	プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADARAI パラメータ	DDKARTE	リーダー	

## JCL の例 (z/OS)

### リカバリロギングの準備 (ADARAI PREPARE)

```
//RAIPREP EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOADLIB
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.ADAyyyyy.ASSOR1
//DDRLOGR1 DD DISP=SHR,DSN=EXAMPLE.ADAyyyyy.RLOGR1
//DDDRUCK DD SYSOUT=A
//DDPRINT DD SYSOUT=A
//DDCARD DD *
ADARUN PROGRAM=ADARAI,SVC=xxx,DEVICE=dddd,DBID=yyyyy,MODE=MULTI
//DDKARTE DD *
ADARAI PREPARE RLOGSIZE=5,MINGENS=5
```

### RLOG のリスト (ADARAI LIST)

```
//RAILIST EXEC PGM=ADARUN
//STEPLIB DD DSN=ADABAS.Vvrs.LOADLIB,DISP=SHR
//DDASSOR1 DD DSN=EXAMPLE.ADAyyyyy.ASSOR1,DISP=SHR
//DDRLOGR1 DD DSN=EXAMPLE.ADAyyyyy.RLOGR1,DISP=SHR
//DDDRUCK DD SYSOUT=A
//DDPRINT DD SYSOUT=A
//DDCARD DD *
ADARUN PROGRAM=ADARAI,SVC=xxx,DEVICE=dddd,DBID=yyyyy,MODE=MULTI
//DDKARTE DD *
ADARAI LIST
/*
```

### リカバリ JCL の作成 (ADARAI RECOVER)

```
//ADARAI EXEC PGM=ADARUN
//STEPLIB DD DSN=ADABAS.Vvrs.LOADLIB,DISP=SHR
//*
//DDASSOR1 DD DSN=EXAMPLE.ADAyyyyy.ASSOR1,DISP=SHR
//*
//DDRLOGR1 DD DSN=EXAMPLE.ADAyyyyy.RLOGR1,DISP=SHR
//DDJCLIN DD DSN=EXAMPLE.ADAyyyyy.RAIJCL(JCLIN),DISP=SHR
//DDJCLOUT DD SYSOUT=A
//*
//DDPLOGR1 DD DSN=EXAMPLE.ADAyyyyy.PLOGR1,DISP=SHR
//DDPLOGR2 DD DSN=EXAMPLE.ADAyyyyy.PLOGR2,DISP=SHR
//*
//DDDRUCK DD SYSOUT=A
//DDPRINT DD SYSOUT=A
//DDCARD DD *
ADARUN PROGRAM=ADARAI,SVC=xxx,DEVICE=dddd,DBID=yyyyy,MODE=MULTI
ADARUN UEX2=UEX2
//DDKARTE DD *
ADARAI RECOVER JCLLOG=YES,RELGEN=0,DRIVES=2
```

## スケルトンジョブ制御の例 (z/OS)

この例は、MVSJOBS データセットのメンバ ADARAIIN に存在します。

```
%%JCL-JOB-HEADER
//ADARECOV JOB 5,'ADA-USER',MSGCLASS=X,CLASS=A,REGION=4096K
//*
//JOBLIB DD DSN=ADABAS.Vvvv.LOAD,DISP=SHR
//*
%%JCL-JOB-TRAILER
//*
/* END OF RECOVERY
/*
//
%%JCL-STEP-TRAILER
//*
/* END OF STEP
/*
%%JCL-DDKARTE
//DDKARTE DD *
%%JCL-ADADEF
//*
/* DEFINE NEWWORK
/*
//%STEP EXEC PGM=ADARUN ADADEF JOB STEP
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDDATAR1 DD DSN=ADABAS.Vvvv.DATAR1,DISP=SHR
//DDWORKR1 DD DSN=ADABAS.Vvvv.WORKR1,DISP=SHR
//DDTEMPR1 DD DSN=ADABAS.Vvvv.TEMPR1,DISP=OLD
//DDSORTR1 DD DSN=ADABAS.Vvvv.SORTR1,DISP=OLD
```

```

//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADADEF,DBID=xxxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *
%KARTE
%%JCL-ADAINV
//*
//* INVERT / COUPLE
//*
//%STEP EXEC PGM=ADARUN ADAINV JOB STEP
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//DDSORTR1 DD DSN=ADABAS.Vvvv.SORTR1,DISP=OLD
//DDTEMPR1 DD DSN=ADABAS.Vvvv.TEMPR1,DISP=OLD
//SYSUDUMP DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADAINV,DBID=xxxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *
%KARTE
%%JCL-ADALOD
//*
//* LOAD A FILE / MASS UPDATE
//*
//%STEP EXEC PGM=ADARUN ADALOD JOB STEP
%SEQUENTIAL
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDDATAR1 DD DSN=ADABAS.Vvvv.DATAR1,DISP=SHR
//DDWORKR1 DD DSN=ADABAS.Vvvv.WORKR1,DISP=SHR
//DDTEMPR1 DD DSN=ADABAS.Vvvv.TEMPR1,DISP=OLD
//DDSORTR1 DD DSN=ADABAS.Vvvv.SORTR1,DISP=OLD
//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADALOD,DBID=xxxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *
%KARTE
%%JCL-ADAORD
//*
//* REORDER
//*
//%STEP EXEC PGM=ADARUN ADAORD JOB STEP
%SEQUENTIAL
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDDATAR1 DD DSN=ADABAS.Vvvv.DATAR1,DISP=SHR
//DDWORKR1 DD DSN=ADABAS.Vvvv.WORKR1,DISP=SHR

```

```

//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADAORD,DBID=xxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *
%KARTE
%%JCL-ADARES
//*
//* PLCOPY / REGENERATE / BACKOUT
//*
//%STEP EXEC PGM=ADARUN
%SEQUENTIAL
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//* Omit DDPLGRx for Parallel and Cluster Services
//DDPLOGR1 DD DSN=ADABAS.Vvvv.PLOGR1,DISP=SHR
//DDPLOGR2 DD DSN=ADABAS.Vvvv.PLOGR2,DISP=SHR
//* Include MERGIN1 and MERGIN2 for Parallel and Cluster Services
//MARGIN1 DD DSN=ADABAS.Vvvv.INTERI,DISP=SHR
//MARGIN2 DD DSN=ADABAS.Vvvv.INTERO,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADARES,DBID=xxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *
%KARTE
%%JCL-DDSIAUS1
//DDSIAUS1 DD DSN=ADABAS.Vvvv.PLCOPY(+1),
// DISP=(NEW,CATLG),UNIT=TAPE
%%JCL-DDSIAUS2
//DDSIAUS2 DD DSN=ADABAS.Vvvv.PLCOPY2(+1), Optional second copy
// DISP=(NEW,CATLG),UNIT=TAPE
%%JCL-ADASAV
//*
//* RESTORE FILE(S)/DATABASE
//*
//%STEP EXEC PGM=ADARUN ADASAV JOB STEP
%SEQUENTIAL
//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDDATAR1 DD DSN=ADABAS.Vvvv.DATAR1,DISP=SHR
//DDWORKR1 DD DSN=ADABAS.Vvvv.WORKR1,DISP=SHR
//DDTEMPR1 DD DSN=ADABAS.Vvvv.TEMPR1,DISP=OLD
//SYSUDUMP DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADASAV,DBID=xxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *

```

```

%KARTE
%%JCL-STARTNUC
//*
//* START NUCLEUS
//*
//%STEP EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=(*,INTRDR)
//SYSUT1 DD DATA,DLM='$$'
//ADANUC JOB 5,'ADANUC',CLASS=A,MSGCLASS=X,REGION=6M,TIME=1440
//NUCxxxx EXEC PGM=ADARUN
//STEPLIB DD DSN=ADABAS.Vvvv.LOAD,DISP=SHR
//*
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR
//DDDATAR1 DD DSN=ADABAS.Vvvv.DATAR1,DISP=SHR
//DDWORKR1 DD DSN=ADABAS.Vvvv.WORKR1,DISP=SHR
//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//DDPLOGR1 DD DSN=ADABAS.Vvvv.PLOGR1,DISP=SHR
//DDPLOGR2 DD DSN=ADABAS.Vvvv.PLOGR2,DISP=SHR
//INTRDR2 DD SYSOUT=(*,INTRDR)
//SYSUDUMP DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN PROG=ADANUC
ADARUN UEX2=UEX2
ADARUN DUALPLS=nnn dual PLOG size
ADARUN DUALPLD=mmmm dual PLOG device type
ADARUN MODE=MULTI
ADARUN DATABASE=xxxxx DATA BASE ID
ADARUN DEVICE=yyyy ASSOCIATOR DEVICE TYPE
ADARUN SVC=zzz SVC NUMBER
$$
//*
//* End of input for internal reader. Check whether nucleus is active.
//*
//ADARAI EXEC PGM=ADARUN
//STEPLIB DD DSN=ADABAS.Vvvv.LOAD,DISP=SHR
//SYSOUT DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROGRAM=ADARAI,DBID=xxxxx,SVC=zzz,DE=yyyy
//DDKARTE DD *
ADARAI CHKDB ACTIVE
//*
%%JCL-ENDNUC
//*
//* ADADBS END NUCLEUS
//*
//%STEP EXEC PGM=ADARUN
//DDASSOR1 DD DSN=ADABAS.Vvvv.ASSOR1,DISP=SHR

```

```

//DDDATAR1 DD DSN=ADABAS.Vvvv.DATAR1,DISP=SHR
//DDWORKR1 DD DSN=ADABAS.Vvvv.WORKR1,DISP=SHR
//DDTEMPR1 DD DSN=ADABAS.Vvvv.TEMPR1,DISP=OLD
//DDSORTR1 DD DSN=ADABAS.Vvvv.SORTR1,DISP=OLD
//DDRLOGR1 DD DSN=ADABAS.Vvvv.RLOGR1,DISP=SHR
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROG=ADADBS,DBID=xxxxx,DE=yyyy,SVC=zzz
//DDKARTE DD *
ADADBS OPERCOM ADAEND
//*
/* Check whether nucleus is inactive.
/*
//ADARAI EXEC PGM=ADARUN
//STEPLIB DD DSN=ADABAS.Vvvv.LOAD,DISP=SHR
//SYSOUT DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//DDDRUCK DD SYSOUT=*
//DDCARD DD *
ADARUN MODE=MULTI,PROGRAM=ADARAI,DBID=xxxxx,SVC=zzz,DE=yyyy
//DDKARTE DD *
ADARAI CHKDB INACTIVE
/*

```

## z/VM

ADARAI は、現在、ADAMAINT EXEC（データベース保守）および INSTADA EXEC（データベースインストール）のどちらでもサポートされていません。ADARAI の実行に必要な情報は、EXEC、CONTROL ファイルなどに手動で追加する必要があります。

### ▶手順 147.1. 必要な情報を次の手順で追加します。

- 1 正しいデータベース ID（デフォルト ID は "00001"）が含まれるように ADARAI PARM を更新します。
- 2 DBA 仮想マシンの PROFILE EXEC ファイルにある ADARAI ミニディスクに、複数の書き込み LINK コマンドを追加します。DDRLOGR1 などの ADARAI ミニディスクを Adabas ニュークリアス仮想マシンの CP ディレクトリで定義する必要があります。
- 3 DDRLOGR1 に対する DATADEF ステートメントで ADFdbid EXEC を完成させます。すべてのユーティリティおよびニュークリアスに対して、このステートメントを常に定義する必要があります。データセットは CMS でフォーマット化され、定義済みのファイル名およびファイルタイプで確保する必要があります。

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	RECOVER に必要
ワーク	DDWORKR1 DDWORKR4	ディスク	RECOVER に必要
データプロテクションログ	DDPLOGRn	テープ/ディスク	RECOVER に必要
リカバリログ (RLOG)	DDRLOGR1	ディスク	
ジョブストリーム入力	DDJCLIN	ディスク	RECOVER に必要
リカバリジョブ出力	DDJCLOUT	ディスク	RECOVER に必要
ADARAI メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADARAI パラメータ	DDKARTE	ディスク/端末/リーダー	

## JCL の例 (z/VM)

### リカバリロギングの開始 (ADARAI PREPARE)

```

DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDRLOGR1,DSN=ADABASV.V.RLOG1,VOL=RLOGV1
DATADEF DDDRUCK,DSN=ADARAI.DDDRUCK,MODE=A
DATADEF DDPRINT,DSN=ADAFRM.DDPRINT,MODE=A
DATADEF DDCARD,DSN=RUNRAI.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADARAI.CONTROL,MODE=A
ADARUN

```

### RUNRAI CONTROL A1 の内容

```

ADARUN PROG=ADARAI,DEVICE=dddd,DB=99,MODE=MULTI

```

### ADARAI CONTROL A1 の内容

```
ADARUN PREPARE RLOGSIZE=5,RLOGDEV=eeee,MINGENS=5
```

## RLOG のリスト (ADARAI LIST)

```
DATADef ASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADef RLOGR1,DSN=ADABASVv.RLOG1,VOL=RLOGV1
DATADef DDDRUCK,DSN=ADARAI.DDDRUCK,MODE=A
DATADef DDPRINT,DSN=ADARAI.DDPRINT,MODE=A
DATADef DDCARD,DSN=RUNRAI.CONTROL,MODE=A
DATADef DDKARTE,DSN=ADARAI.CONTROL,MODE=A
```

## RUNRAI CONTROL A1 の内容

```
ADARUN
PROG=ADARAI,DEVICE=dddd,DB=yyyyy,MODE=MULTI
```

## ADARAI CONTROL A1 の内容

```
ADARUN LIST
```

## VSE

---

次の機能は、VSE システムで ADARAI を使用する場合に使用できます。

機能	対処
CHKDB	データベースステータスのチェック
DISABLE	リカバリログ (RLOG) の無効化
LIST	RLOG コンテンツのリスト
PREPARE	RLOG の開始
REMOVE	RLOG の削除

ジョブストリームの再構築に使用する ADARAI RECOVER 機能は、現在、VSE システムではサポートされていません。

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク	注 1 参照	
データストレージ	DATARn	ディスク	注 1 参照	RECOVER に必要
ワーク	WORKR1	ディスク		RECOVER に必要
リカバリログ (RLOG)	RLOGR1	ディスク		
データプロテクションログ	PLOGRn	テープ ディスク		RECOVER に必要
ジョブストリーム入力	JCLIN	SYSIPT		RECOVER に必要
リカバリジョブ出力	JCLOUT	SYSPCH		RECOVER に必要
ADARAI メッセージ		プリンタ	SYS009	メッセージおよびコード
ADARUN メッセージ		プリンタ	SYSLST	メッセージおよびコード
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 注 1 参照	
ADARAI パラメータ		リーダー	SYSIPT	



**Note:** すべてのプログラマ論理ユニットを使用できます。

## JCS の例 (VSE)

### リカバリロギングの開始 (ADARAI PREPARE)

```
// ASSGN SYS009,00F
// EXEC PROC=ADAVvFIL
// EXEC PROC=ADAVvLIB
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARAI,SVC=xxx,DEVICE=dddd,DB=yyyyy,MODE=MULTI
/*
ADARAI PREPARE RLOGSIZE=5,MINGENS=5
/*
```

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

### RLOG のリスト (ADARAI LIST)

```
// ASSGN SYS009,00F
// EXEC PROC=ADAVvFIL
// EXEC PROC=ADAVvLIB
// ASSGN SYS000,SYSIPT
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARAI,SVC=xxx,DEVICE=dddd,DB=yyyyy,MODE=MULTI
/*
ADARAI LIST
/*
```

# 148 ADAREP ユーティリティ：データベースステータスレポート

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- レポート構文
- セーブテープ入力の処理
- レポート内容
- JCL/JCS の必要項目と指定例



# 149 機能概要

---

ADAREPユーティリティはデータベースステータスレポートを作成します。このレポートにはデータベースの現在の物理レイアウトおよび論理的内容に関する情報が表示されます。SAVETAPE パラメータが指定されている場合は、セーブテープに関する情報も表示されます。

データベースステータスに出力される情報には次のものがあります。

- データベースの名前、番号、作成日付／時刻、ファイルステータス、現在のログ番号。
- シリンダおよびブロックで、現在使用されているものと割り当てられているが未使用のアソシエータ、データ、ワークのスペースの量と位置。
- デバイスタイプ、VOLSER 番号、ファイル番号（該当する場合）を含むアソシエータおよびデータストレージの RABN 情報と使用状況（AC、NI/UI、データストレージ、DSST、代替（セーブテープからのみ）、未使用）。
- 代替 RABN ブロックの量と位置（セーブテープからのみ）。
- ISN、エクステント、パディングファクタ、アソシエータとデータストレージの使用／未使用のスペース、ファイルオプションのファイルごとの概要。
- すべての概要情報に加えて、MINISN/MAXISN 設定、MUPLEX 設定、詳細スペース情報、作成および最終使用の日付／時刻、フィールド定義テーブル（FDT）の内容、一般および拡張チェックポイントファイルの情報などが含まれる詳細情報（ファイルごとの表示も可能）。



# 150 レポート構文

---

■ オプションパラメータ .....	808
■ 例 .....	811

```

ADAREP [REPORT]
      [ {CPEXLIST [, OFFSET = {column-number | 63}] | CPLIST }
        [FROMDATE = yyyyymmdd] [TODATE = yyyyymmdd]
        [FROMSESSION = session] [TOSESSION = session]]
      [ {FILE = {file-list | all-files} [, LAYOUT = 1] | NOFILE } ]
      [LIMCOUNT | NOCOUNT]
      [NOFDT]
      [ {NOLGLIST | NOPHLIST | NOSTD } ]
      [NOUSERABEND] }
      [SAVETAPE
        [PLOGNUM = protection-log-number [ {SYN1 | SYN4} = PLOG-block-number ]]]
    
```

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

ADAREPは、単独で指定してデータベースステータスレポートを取得することができます。パラメータ値を追加して、レポートをカスタマイズすることもできます。

**CPEXLIST**：チェックポイントリストを拡張形式で出力

**CPLIST**：チェックポイントリストを通常形式で出力

これらのパラメータを使用して、チェックポイントリストを通常形式（CPLIST）または拡張形式（CPEXLIST）で出力します。チェックポイント情報を表示するには、CPEXLISTまたはCPLISTのどちらかを指定する必要があります。CPEXLISTを指定すると、通常のCPLIST情報に次の情報が追加されますが、内容は元のチェックポイントごとに異なります。

- （ユーティリティまたは Adabas Online System／基本サービスの）機能名
- チェックポイント固有データ

CPEXLIST レポートを表示する場合は、OFFSET を指定して表示を読みやすくすることができます。

FROMDATE、TODATE、FROMSESSION、TOSESSION の各パラメータを使用して、出力するチェックポイントの範囲を指定できます。

**FILE | NOFILE**：ファイル情報の表示または非表示

FILE は、ステータス情報を出力または表示するファイルのリストを定義します。このパラメータを省略すると、すべてのファイルのステータス情報が出力対象になります。

NOFILE を指定すると、すべてのファイルおよびフィールドの説明情報の出力が禁止されます。

**FROMDATE | TODATE**：レポートの開始／終了チェックポイント日付

CPLISTまたはCPEXLISTを指定した場合に、チェックポイント情報の開始日または終了日、あるいはその両方を特定できます。yyyymmdd形式の有効な日付指定方法は次のとおりです。

```
ADAREP FROMDATE=19960101,TODATE=19960228      January 1-February 28, 1996
ADAREP FROMDATE=19951111                      November 11, 1995 to checkpoint file end
ADAREP TODATE=19951223                        From checkpoint file begin to (and including)
                                              December 23, 1995
```

FROMDATEを指定しない場合、レポートはシステムで最初のチェックポイント情報

(FROMSESSIONでそれよりも後が指定された場合はFROMSESSIONセッションで最初の情報)から開始されます。TODATEを指定しない場合、レポートは最後のチェックポイント情報(TOSESSIONでそれよりも前が指定された場合はTOSESSIONのセッションで最後の情報)まで継続されます。

**FROMSESSION | TOSESSION**：レポートの開始／終了セッション

開始または終了、あるいはその両方のセッション番号を指定します。FROMSESSIONのセッション番号よりも前のセッションとTOSESSIONのセッション番号よりも後のセッションは、レポート情報に含まれません。FROMSESSIONを指定しない場合、レポートはシステムで最初のチェックポイント情報(FROMDATEでそれよりも後が指定された場合はFROMDATE日付の最初の情報)から開始されます。TOSESSIONを指定しない場合、レポートは最後のチェックポイント情報(TODATEでそれよりも前が指定された場合はTODATE日付の最後の情報)まで継続されます。

**LAYOUT**：出力形式

LAYOUT=1を指定すると、"Contents of Database"の表が1つの120文字(120列)形式で出力されます。通常、"Contents of Database"と"File Space Allocations"の情報は、レポートでは2つの別のセクションに表示されます。ただし、LAYOUT=1を指定すると、この2つが"Contents of Database"セクションにマージされます。さらに、パディングファクタも追加されます。

以下に、LAYOUT=1を指定したときにレポートの"Contents of Database"セクションがどのように表示されるかを示します。このサンプルを、の「[データベースの内容：全般的なファイルステータス](#)」および「[ファイルスペース割り当て](#)」で説明されている"Contents of Database"セクションや"File Space Allocations"セクションと比較してください。

```
*****
* Contents of Database      68 (EXAMP68          ) *          yyyy-mm-dd  hh:mm:ss
*****

File      Name              Loaded      TOP-ISN      MAX-ISN      PADD
Blocks Allocated          (Blocks Unused)
                                     A% D%          NI
UI          AC              DATA/CYL
```

```

1 EMPLOYEES          2001-12-28          1107          1695  10 10          200
15          2          150/1
1  Extents: NI      1  UI      1  AC      1  DS      1          138
2          104/0
2 MISCELLANEOUS     2001-12-28          1779          2543  10 10          200
15          3          150/1
2  Extents: NI      1  UI      1  AC      1  DS      1          174
10          97/0
3 VEHICLES          2001-12-28          773          1695  10 10          200
15          2          150/1
3  Extents: NI      1  UI      1  AC      1  DS      1          183
8          138/0

```

### LIMCOUNT | NOCOUNT : ロードされたレコード数のカウント

ADAREP はアドレスコンバータを読み取ることによって、ファイルの RECORDS LOADED の値を決定します。そのため、ファイルが非常に大きい場合は、大量の I/O アクティビティが発生する可能性があります。LIMCOUNT を指定すると、ADAREP はファイルの TOPISN の値をチェックします。TOPISN が 1000 を超える場合、RECORDS LOADED の下に "NOT COUNTED" と表示されます。

NOCOUNT を指定すると、どのファイルの RECORDS LOADED についても値は出力されません。LIMCOUNT および NOCOUNT のどちらも指定しない場合、ADAREP では各ファイルの RECORDS LOADED の正確な値がコンパイルされます。

### NOFDT : フィールド定義の出力の省略

各ファイルのフィールド定義テーブル (FDT) の出力を省略します。ADAREP は Adabas チェックポイントファイルおよびセキュリティファイルの FDT を出力しません。

### NOLGLIST | NOPHLIST | NOSTD : データベースレイアウトの出力の省略

NOLGLIST を指定すると、論理データベースレイアウト情報が省略されます。NOPHLIST を指定すると、物理データベースレイアウト情報が省略されます。NOSTD を指定すると、すべてのデータベースレイアウト情報が省略されます。これは、NOLGLIST と NOPHLIST を指定した場合と同じです。

### NOUSERABEND : アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### OFFSET : 拡張レポートの行揃えの定義

OFFSET パラメータを指定すると、拡張情報の最初の部分が通常のチェックポイント情報の最後に揃えて 1 行に出力されます。デフォルトは 63 です。拡張情報を表示するには OFFSET

を小さくする必要があります。このようにしない場合、拡張部分が 80 列の中に表示されなくなることがあります。

**PLOGNUM**：プロテクションログ番号

PLOGNUMは、ADASAVのセーブ操作中に使用されたニュークリアスプロテクションログの番号を指定します。Adabasバージョン5.1のオンラインセーブで作成したセーブテープを指定する場合は、このパラメータが必須です。Adabasバージョン5.2以上を使用して作成したテープでは、PLOGNUMがテープに含まれています。このようなテープに対してPLOGNUMを指定すると、テープの情報が書き換えられます。

**SAVETAPE**：セーブテープレポートの出力

SAVETAPEを指定すると、レポートがセーブテープから出力されます。セーブテープレポートは、セーブテープの内容を調べるのに使用します。詳細については、「[セーブテープ入力の処理](#)」セクションを参照してください。

CPLISTまたはCPEXLISTをSAVETAPEとともに指定することはできません。NOCOUNTはSAVETAPEとともに指定する必要があります。これは、ADAREPではセーブテープ上のファイルのレコード数がカウントされないためです。CPLISTまたはCPEXLISTを指定した場合や、NOCOUNTを指定しなかった場合は、ADAREPによって警告メッセージが出力され、これらのオプションは内部で変更され、処理が続行されます。

**SYN1 | SYN4**：開始ブロック番号

SYN1/SYN4は、対応するセーブ操作が開始したSYN1/SYN4チェックポイントが含まれるブロック番号を指定します。Adabasバージョン5.1のオンラインセーブで作成したセーブテープを指定する場合は、このパラメータが必須です。ADASAVバージョン5.2以上を使用して作成したテープでは、SYN1/SYN4チェックポイント情報がテープに含まれています。このようなテープに対してSYN1/SYN4を指定すると、テープの情報が書き換えられます。

## 例

**例 1**

```
ADAREP REPORT
```

データベース、ファイル、フィールドすべての情報が出力されます。チェックポイントデータは出力されません。

**例 2**

```
ADAREP REPORT NOFDT
```

データベースおよびファイルの情報が出力されます。チェックポイントデータおよび各ファイルのフィールド定義は出力されません。

**例 3**

## レポート構文

---

```
ADAREP CPEXLIST, FROMDATE=19980701, TODATE=19980715  
ADAREP NOSTD, NOFILE
```

1998年7月1日から1998年7月15日の間のすべてのチェックポイントのチェックポイントリスト（拡張形式）が出力されます。その他の情報は出力されません。

### 例 4

```
ADAREP REPORT SAVETAPE, NOCOUNT
```

データベース、ファイル、フィールドすべての情報がセーブテープから出力されます。ADAREPではセーブテープ上のファイルのレコード数がカウントされないため、NOCOUNTを指定する必要があります。

# 151      セーブテープ入力の処理

---

■ プロテクションログの入力 .....	814
■ 入力テープのチェック .....	815
■ 同時パラメータ .....	815
■ デルタセーブテープのレポート .....	816
■ レポートレイアウト .....	816

SAVETAPE キーワードを指定すると、レポートはセーブテープから出力されます。

セーブテープの使用に関して次の点に注意してください。

- セーブテープは ADASAV バージョン 5.1 以上で作成されたものでなければなりません。
- オンラインまたはオフラインのどちらで作成されたものでもかまいません。
- データベースセーブ、ファイルセーブ、およびデルタセーブによるテープを使用できます。
- セーブテープは DD/SAVE シーケンシャル入力ファイルとして提供される必要があります。

ADAREP ではセーブテープ全体はスキャンされません。通常は最初のカセット／テープリールを指定するのみで十分です。

ADAREP は、ジェネラルコントロールブロック (GCB)、読み取り不可ブロック (BUB)、ミラーテーブル、ミラーBUB、フリースペーステーブル (FST)、およびすべてのファイルコントロールブロック (FCB) を取得するためにセーブテープを読み取ります。これらの情報がメインメモリに格納されると、ADAREP は通常のデータベースレポートについて読み取りを続けます。ファイル定義テーブル (FDT) はメインメモリバッファに保存されないため、必要に応じてセーブテープから読み取られます。



### Notes:

1. Adabas バージョン 7.2 以上ではサポートされていないため、BUB やミラー情報は保存されません。しかし、バージョン 7.2 よりも前の Adabas バージョンからバージョン 5.1 に戻されたセーブテープ上の BUB やミラー情報は、引き続きレポートされます。
2. ADASAV バージョン 5.1 を使用して作成したセーブテープには、ミラーテーブルやミラー BUB、FST は含まれません。レポートの該当する場所には、これらのブロックが使用不可であることを示すメッセージが表示されます。

このchapterでは、次のトピックについて説明します。

## プロテクションログの入力

---

オンラインセーブテープを使用する場合は、対応するプロテクションログを DD/PLOG シーケンシャル入力ファイルとして指定することもできます。

- DD/PLOG を指定すると、ADAREP は FCB ブロックおよび FST ブロックのプロテクションログをスキャンして、それぞれのバージョンが最新であることを確認します。
- DD/PLOG を指定しない場合、ADAREP は警告メッセージを出力して処理を続行します。オンラインセーブ操作の開始時点 (SYN1/SYN4 チェックポイントの時点) のデータベースステータスが表示されます。オンラインセーブ操作中に割り当てられたセカンダリエクステンションは、プロテクションログのみで表示されるため、レポートには反映されません。さらに、セーブテープ上の FCB ブロックと FST ブロックの中の矛盾によるファントムエラーが物理レ

アウトセクションでレポートされる場合があります。これは、ADASAV が FCB ブロックと FST ブロックを保存した短い期間にセカンダリエクス Tent が割り当てられた場合にのみ発生します。

ADASAV バージョン 5.1 で作成したセーブテープを使用している場合は、パラメータ PLOGNUM と SYN1/SYN4 に、プロテクションログ番号および SYN1 または SYN4 チェックポイントのブロック番号を指定する必要があります。ADASAV バージョン 5.2 以上を使用して作成したテープでは、テープにこれらの情報が含まれています。このようなテープに対して PLOGNUM や SYN1/SYN4 を指定すると、テープの情報が書き換えられます。

DD/PLOG を指定するときは、2 つのテープユニットを並行して使用する必要があります。ADASAV RESTONL ではセーブテープとプロテクションログを連結できないためです。

## 入力テープのチェック

DD/SAVE および DD/PLOG 入力データセットをオープンすると、ADAREP は入力テープが正しいことを確認するために照合を行います。

- 無効なセーブテープが指定された場合、ADAREP は ERROR-128 (invalid save tape supplied) を表示して終了します。
- 無効なプロテクションログが指定された場合、ADAREP は適切な警告メッセージを表示し、コンディションコードを 4 に設定して、処理を続行します。

## 同時パラメータ

ファイルの CPLIST/CPEXLIST 情報およびロードされたレコード数は、セーブテープから出力できません。SAVETAPE とともに CPLIST/CPEXLIST パラメータが指定されるか、または NOCOUNT パラメータが指定されない場合、ADAREP は警告メッセージを出力し、これらのオプションを内部で変更して処理を続行します。

ADASAV バージョン 5.3.2 以上で作成されたセーブテープの場合は、レポートに VOLSER 番号が出力されます。これよりも前のバージョンの ADASAV を使用して作成されたセーブテープの場合は、アスタリスクが VOLSER 番号として出力されます。

## デルタセーブテープのレポート

デルタセーブテープは、情報のほとんどがアクセスできないか、または再構成する必要があります。

- デルタセーブステータスは、常に有効です。DLOG エリアの使用はヘッダーの 1 ブロックのみで、"n%" と表示されます。
- 最後の完全なセーブの番号、最後のデルタセーブの番号および日時は DSID から取得されません。
- 変更ブロックの見積り数は、MAXFILES の 5 倍に 30 を加えた値を次の 100 の倍数に切り上げた数です。
- DLOG エリアの位置は GCB から取得されます。
- 最後の完全なセーブの日時は再構成できないので、常に "unknown" と表示されます。

## レポートレイアウト

セーブテープレポートは、セーブテープの内容を調べるのに使用します。

セーブテープレポートの先頭には短いヘッダーがあります。このヘッダーには、使用されたセーブテープの種類、セーブテープがオンラインまたはオフラインのどちらで作成されたか、作成日時、作成に使用された ADASAV のバージョン、セーブテープ上のデータベース ID および、場合によっては、セーブテープのデルタセーブ ID が示されます。オンラインセーブテープの場合は、対応するプロテクションログのセッション番号と SYN1/SYN4 チェックポイントのブロック番号（指定されたものまたはテープから抽出されたもの）が表示されます。

```
A D A R E P   Vv.r   SMs   DBID = nnnnnn   STARTED           yyyy-mm-dd   hh:mm:ss
```

```
PARAMETERS:
```

```
-----
```

```
ADAREP REPORT SAVETAPE
```

```
*****
```

```
*
* REPORT FROM           ONLINE DATABASE SAVE           *
* CREATED AT           yyyy-mm-dd   hh:mm:ss           *
* BY ADASAV VERSION    V vr                               *
* DBIB                  nnnnnn                           *
* DSID                  1 / 0 /   yyyy-mm-dd   hh:mm:ss   *
```

```
* PLOG SESSION NR          17          *
* SYN1 BLOCK NR           137          *
*                          *
*****
*****
*                          *
* DATA BASE REPORT *          yyyy-mm-dd   hh:mm:ss
*                          *
*****
```

レポートの1行目に出力されるデータベースIDは、ADARUN DBID パラメータから取得されます。このDBIDには、セーブテープ上のデータベースIDとの照合は行われません。その代わりに、セーブテープがオープンされてGCBが読み取られた後は、セーブテープのDBIDがレポート全体を通して使用されます。

ファイルセーブレポートの物理レイアウトは、データベース内の各RABNの使用状況を示すRABN範囲の表になります。ファイルセーブテープにはセーブされたファイルのFCBのみが含まれるため、物理レイアウト表は連続しませんが、これはエラーではなく"unknown"範囲としてレポートされます。



# 152 レポート内容

---

■ データベースの全般的な情報 .....	820
■ ファイル情報 .....	831
■ チェックポイント情報 .....	843

ADAREP データベースステータスレポートには、全般的なデータベース情報に加えて、データベース内の各ファイルのステータス、割り当て、および定義に関する情報が含まれます。レポートは SYSLST (BS2000)、DDDRUCK (z/OS または z/VM)、SYS009 (VSE) の各データセットから出力するように設計されていますが、次の図は、端末に表示されるレポート出力の例を示しています。以降の各ページでは、レポートに表示される順序で各セクションを示し、各部分の説明は同じページまたは見開きのページに記載されています。

 **Note:** 個々の Adabas アドオン製品を使用して、ADAREP レポートに表示される情報を補足することができます。例えば、データベースでレプリケーションがサポートされる (Event Replicator for Adabas を使用) 場合は、このレポートの各所に追加の統計値が表示されます。このレポートにアドオン製品が与える影響に関する完全な情報については、それぞれの Adabas アドオン製品のドキュメントを参照してください。

このchapterでは、次のトピックについて説明します。

## データベースの全般的な情報

---

最初のセクションには、データベースに関する全般的な情報とその物理レイアウトが表示されます。

```
*****
* Data Base Report *                               yyyy-mm-dd7  hh:mm:ss
*****

Data Base Name           = EXAMPLE-DB
Data Base Number        =          68
Date Loaded              = yyyy-mm-ss
Time Loaded              = hh:mm:ss
System Files             =          19
Maximum number of files =         1000
Number of files loaded   =          20
Current Log Tape Number =          0
```

```
RABNSIZE           =          3
Recovery Aid       = No
```

ユニバーサルエンコーディングサポート (UES) が有効 (UES=YES) な場合は、次の情報も表示されます。

```
Universal Encoding Sup. = Yes
ALPHA FILE ENCODING     =          37
WIDE FILE ENCODING      =         4095
ALPHA ASCII ENCODING    =          437
WIDE USER ENCODING      =         4095
Replication             = No
```

UES=NO の場合、この情報は表示されません。

フィールド	説明
DATABASE NAME	データベースに割り当てられた名前。 ADADEF ユーティリティの DBNAME パラメータを参照。
DATABASE NUMBER	データベースに割り当てられた番号 (ID)。 ADADEF ユーティリティの DBIDENT パラメータを参照。
DATE LOADED	データベースが最初に定義された日付。
TIME LOADED	データベースが最初に定義された時刻。
SYSTEM FILES	Adabas システムファイルのファイル番号。
TRIGGER FILE	データベース内にトリガファイルが存在する場合は、そのファイル番号を表示します。データベース内にトリガファイルが存在しない場合には、この行は出力されません。
MAXIMUM NUMBER OF FILES	データベースに許可される最大ファイル数。 ADADEF ユーティリティの MAXFILES パラメータを参照。
NUMBER OF FILES LOADED	データベース内に現在存在するファイル数。
CURRENT LOG TAPE NUMBER	データベースの最新のデータプロテクションログの番号。
RABNSIZE	データベース内のブロックの長さ。 RABNSIZE=3 は 24 ビットブロックを示し、RABNSIZE=4 は 31 ビットブロックを示します。
RECOVERY AID	Adabas Recovery Aid (ADARAI) がデータベースに対してアクティブであるかどうかを示します。
UNIVERSAL ENCODING SUPPORT	ユニバーサルエンコーディングサポート (UES) がデータベースに対してアクティブであるかどうかを示します。
ALPHA FILE ENCODING	データベース内の英数字 (A) 形式フィールドに対する現在のファイルエンコードセット。 EBCDIC に対応する必要があります。
WIDE FILE ENCODING	データベース内のワイド文字 (W) 形式フィールドに対する現在のファイルエンコードセット。

## レポート内容

フィールド	説明
ALPHAASCII ENCODING	データベース内の英数字 (A) 形式フィールドに対する現在のユーザーエンコードセット。ASCII に対応する必要があります。
WIDE USER ENCODING	データベース内のワイド文字 (W) 形式フィールドに対する現在のユーザーエンコードセット。
REPLICATION	データベースのレプリケーション (Event Replicator for Adabas を使用) がアクティブであるかどうかを示します。

## データベースコンポーネントのスペース割り当て

"PHYSICAL LAYOUT" の表には、データベースの主なコンポーネント (アソシエータ、データストレージ、ワーク) のスペース割り当てがリストされます。

"UNUSED STORAGE" の表には、アソシエータエリアおよびデータストレージエリアの未使用スペースがリストされます。このスペースはデータベースのどのファイルにも割り当てられていません。

### PHYSICAL LAYOUT

DD- NAMES	I I	DEV TYPE	I I	NMBR OF CYLS	I I	NMBR OF BLOCKS	EXTENTS IN FROM TO	BLK. I	BLOCK LNNGTH	I I	NMBR OF M-BYTE	I I
ASSOR1	I	3380	I	100	I	28481	1	28481	I	2004	I	54
DATAR1	I	3380	I	200	I	26991	1	26991	I	4820	I	124
WORKR1	I	3380	I	40	I	5391	1	5391	I	4820	I	24

### UNUSED STORAGE

DD- NAMES	I I	DEV TYPE	I I	NMBR OF CYLS	I I	NMBR OF BLOCKS	EXTENTS IN FROM TO	BLK. I	BLOCK LNNGTH	I I	NMBR OF M-BYTE	I I
ASSOR1	I	3380	I	98	I	28134	328	28461	I	2004	I	54

DATAR1	I	3380	I	198	I	26811	131	26941	I	4820	I	124	I
--------	---	------	---	-----	---	-------	-----	-------	---	------	---	-----	---

これらの表の各列は次の情報を示します。

列	説明
DDNAMES	データベースのアソシエータ、データストレージ、ワークの各コンポーネントを定義するジョブ/タスク制御名 (DD 接頭辞なし)。
DEV TYPE	アソシエータ、データストレージ、ワークの各コンポーネントが含まれる物理デバイス。
NMBR OF CYLS	アソシエータ、データストレージ、ワークの各コンポーネントに割り当てられた DASD シリンダの数。1つの完全なシリンダに満たないシリンダが割り当てられている場合は、この列に "0" が表示されます。
NMBR OF BLOCKS	アソシエータ、データストレージ、ワークの各コンポーネントに割り当てられたブロックの総数。
EXTENTS IN BLK	エクステントがブロック範囲ごとにリストされます。
BLOCK LENGTH	ブロックサイズ。ブロックサイズはコンポーネントおよびデバイスタイプに応じて異なります。
NUMBER OF M-BYTES	コンポーネントのストレージサイズをメガバイト単位で示します。

### データベースの内容：一般的なファイルステータス

次のセクションには、データベース内の各ファイルのステータスに関する情報が表示されます。以下に例を示します。

```

*****
* Contents of Database      99 (EXAMPLE-DB)          *          yyyy-mm-dd  hh:mm:ss
*****

```

File	Name	Loaded	TOP-ISN	MAX-ISN	EXTENTS			
					N	U	A	D
1	EMPLOYEES	2001-12-28	1107	1695	1	1	1	1

## レポート内容

2	MISCELLANEOUS	2001-12-28	1779	2543	1	1	1	1
3	VEHICLES	2001-12-28	773	1695	1	1	1	1

LOB ファイルを使用するデータベースを表示する別の例です。

```

*****
* Contents of Database      99 (EXAMPLE-DB)          *          yyyy-mm-dd  hh:mm:ss
*****
File      Name                Loaded      TOP-ISN    MAX-ISN
          N      U      A      D
1  SQLNC                2006-02-17      0      1377      1      1      1      301 *
2  BASEFILE             2006-09-01      45      1377      1      1      1      1
3  LOBFILE              2006-09-01      3      1377      1      1      1      1
6  abcdefghijklmnop    2006-07-17      0      4133      1      1      1      300

Warning: * indicates less than 10 file extents remain for use

```

ADAREP ユーティリティの実行に LAYOUT=1 を指定すると、レポートのファイルスペース割り当てセクションに表示されるすべての情報にパディングファクタ情報がマージされてこのセクションに表示されます。

この表の各列は次の情報を提供します。

列	説明
FILE	Adabas ファイル番号。
NAME	ファイル名。ADALOD ユーティリティの NAME パラメータを参照。このファイルでさらに 10 個以上のエクステントを構築できない場合は、名前の右にアスタリスク (*) マークが付きます。
LOADED	ファイルがロードされた日付。
TOP-ISN	現在ファイルに割り当てられている最大 ISN。
MAX-ISN	ファイル内のレコードに割り当て可能な最大 ISN。ADALOD ユーティリティの MAXISN パラメータを参照。
EXTENTS	ノーマルインデックス (N)、アップパーインデックス (U)、アドレスコンバータ (A)、およびデータストレージ (D) に現在割り当てられている論理エクステントの数。定義可能になった論理ファイルエクステントの最大数は、アソシエータの第 1 データセット (DDASSOR1) のブロックサイズから決定されます。エクステントの情報は、FCB の可変セクション内に格納されます。使用された FCB サイズがアソシエータデータセットのブロックサイズに到達するまで、新しいエクステントを追加できるようになりました。エクステント制限に到達した場合は、最後のエクステントが一杯になる前に、ADAORD REORFILE またはユーティリティシーケンスの ADAULD、ADADBS DELETE、ADALOD LOAD を使用してファイルをリオーダしてください。このようにしない場合は、Adabas によってファイルがロックされます。

列	説明
PADD	<p>アソシエータ (A%) およびデータストレージ (D%) に定義されたブロックパディングファクタ。詳細については、ADALOD LOAD ユーティリティの <b>ASSOFFAC</b> パラメータおよび <b>DATAPFAC</b> パラメータを参照。</p> <p><b>注意:</b> この列は LAYOUT=1 が指定されている場合にのみ表示されます。</p>

## ファイルオプション

次のセクションには、データベース内の各ファイルに対してアクティブなファイルオプションがリストされます。以下に例を示します。

```

*****
* File Options *
*****

          ADAM File
          . Coupled File
          . . ISNREUSE
          . . . DSREUSE
          . . . . CIPHERED File
          . . . . . Expanded File
          . . . . . . USERISN
          . . . . . . . NOACEXTENSION
          . . . . . . . . MIXDSDEV
          . . . . . . . . . PGMREFRESH
          . . . . . . . . . . Multi-Client File
          . . . . . . . . . . . Index Compressed
          . . . . . . . . . . . . 2-Byte MU/PE Index
          . . . . . . . . . . . . . Spanned Record
          . . . . . . . . . . . . . . Replicated
          . . . . . . . . . . . . . . . Priv Use
File Name
-----
1 EMPLOYEES      . . I D . . . . .
2 64BIT-1       . . . D . . . . .

```

## レポート内容

```

3 MISC . . . I D . . . . .
19 CHECKPOINT-FILE . . . D . . . . .

```

LOB ファイルを使用するデータベースを表示する別の例です。

```

*****
* File Options *
*****

          ADAM File
          . Coupled File
          . . ISNREUSE
          . . . DSREUSE
          . . . . CIPHERED File
          . . . . . Expanded File
          . . . . . USERISN
          . . . . . NOACEXTENSION
          . . . . . MIXDSDEV
          . . . . . PGMREFRESH
          . . . . . Multi-Client File
          . . . . . Index Compressed
          . . . . . 2-Byte MU/PE Index
          . . . . . Spanned Record
          . . . . . Replicated
          . . . . . Contains LOB Fields
          . . . . . Privileged Use
          . . . . . LOB File

File Name
-----
2 BASEFILE . . . I D . . . . . C . . . L . .
3 LOBFILE . . . I D . . . . . C . . . . . L
11 UES-EMPLOYEEES . . . D . . . . .
13 UES-TEST . . . D . . . . .
14 DBCS3035 . . . D . . . . .
15 COLLATION1 . . . I D . . . . . P . . . . .
18 SECURITY . . . D . . . . .
19 CHECKPOINT . . . D . . . . .
23 EMPL23-EXT . . . D . . U . . P . . . . .
24 SAGABS_MYFILEEXX . . . I D . . . . . M . . C . . . . .
50 EMPL50-EXT . . . D . . U . . . . .
88 BASEF-LBLA . . . D . . . . . L . .
89 LOBF-LBLA . . . D . . . . . C . . . . . L
101 file101 . . . I D . . . . . M . . C . . . . .

```

```

266 UES-EMPLOYEES . . . D . . . . .
267 UES-EMPLOYEES . . . D . . . . .

```

ファイルに対してどのオプションがアクティブであるかは、各ファイル名が表示されている行で、次のようなコードを使用して示されます。

コード	説明
A	ADAM ファイル。ADAM オプションでロードされたファイルです。
C	カップリング、暗号化、またはインデックス圧縮。ファイルが1つ以上のファイルにカップリングされていること、ファイルデータが暗号化されていること、またはファイルインデックスが圧縮されていることのうち、1つ以上が該当することを示します。
D	スペースの再利用。レコード削除の結果として発生したブロック内の解放されたスペースは、新しいレコードに利用できます。
I	ISN の再利用。削除されたレコードの ISN は、新しいレコードに再割り当てすることができます。
L	このコードが [Contains LOB Fields] 列に表示されている場合は、そのファイルに LB フィールドが含まれています (このファイルは基本ファイルです)。このコードが [LOB File] 列に表示されている場合は、そのファイルは LOB ファイルであり、基本ファイルではありません。 <b>注意:</b> これら 2 つの列は相互に排他的です。"L" はどちらかの列にのみ表示されるか、またはどちらにも表示されません。
M	MIXDSDEV がアクティブ (複数のデータストレージデバイスタイプ) であるか、マルチクライアントファイルであるか、あるいはその両方。
N	NOACEXTENSION オプションで定義されたファイル。
P	PGMREFRESH がアクティブです。"P" は Priv Use 列でも使用されます。この場合は、ファイルがニュークリアスによって特権使用のためにロックされていることを示します。特権使用情報は、ADAREP レポートの「 <a href="#">ファイル情報</a> 」セクションでも確認できます。
R	ファイルに対してレプリケーション (Event Replicator for Adabas 処理) がアクティブです。
S	ファイルに対してスパンドレコードサポートがアクティブです。
T	ファイルに対して 2 バイトの MU/PE インデックス (MU/PE の発生が 191 を超えた場合) がアクティブです。
U	USERISN オプションでロードされたファイルです。
X	ファイルは拡張ファイルのコンポーネントです。

## ファイルスペース割り当て

次のセクションには、データベース内の各ファイルに割り当てられたスペースが示されます。以下に例を示します。

```

*****
* FILE SPACE ALLOCATIONS *
*****

FILE      NAME      ALLOC.: NI      UI      AC      DATA/CYL
      UNUSED:

1      EMPLOYEES      100      30      03      80/0
1              24      17              31/0
2      VEHICLES      10      20      03      30/0
2              03      02              12/0
10     CHECKPOINT    10      01      01      20/0
10              05      0              11/0
    
```

LOB ファイルを使用するデータベースを表示する別の例です。

```

*****
* File Space Allocations *
*****

File      Name      Alloc.: NI      UI      AC      Data/Cyl
      Unused:

2  BASEFILE      10              5              1              10/0
2              10              5              9/0
3  LOBFILE      10              5              1              100/1
3              9              3              99/1
11 UES-EMPLOYEES  47              20             1              75/1
11              0              4              53/0
13 UES-TEST      30              20             1              10/0
13              25              14              9/0
14 DBCS3035      60              22             1              30/0
14              22              9              8/0
15 COLLATION1    6              3              1              1/0
15              4              0              0/0
18 SECURITY      1              1              1              1/0
18              1              0              1/0
19 CHECKPOINT    30              3              4              90/1
19              30              2              53/0
23 EMPL23-EXT    60              24             2              30/0
23              52              19             27/0
    
```

24	SAGABS_MYFILEXX	24	4	1	5/0
24		24	0		5/0
50	EMPL50-EXT	80	49	2	100/1
50		25	28		75/1
88	BASEF-LBLA	10	5	1	10/0
88		10	4		10/0
89	LOBF-LBLA	10	5	8	411/5
89		10	4		411/5
101	file101	7	14	2	72/0
101		6	12		71/0
266	UES-EMPLOYEEES	30	20	1	75/1
266		30	18		75/1
267	UES-EMPLOYEEES	63	20	1	75/1
267		16	4		53/0

LAYOUT=1を指定すると、このセクションはレポートのContents of Database セクションにマージされます。

ファイルスペース割り当て表では、各ファイルが2行にわたって表示されます。最初の行は、割り当てられたブロック数とシリンダ数を示します。2行目は、現在使用されていないブロック数とシリンダ数を示しています。

最初の2列はファイルの番号と論理名を示しています。残りの列は次の情報を示しています。

列	示している数
NI	ノーマルインデックスのブロック数
UI	アップパーインデックスのブロック数
AC	アドレスコンバータのブロック数
DATA/CYL	データストレージのブロック数およびシリンダ数

## LOB ファイル

データベースにLOBファイルが存在する場合は、LOBファイルを説明する追加セクションがレポートに表示されます。以下に例を示します。

```

*****
*                               *
* LOB Files *                               yyyy-mm-dd  hh:mm:ss
*                               *
*****

File with  I Associated I
LOB fields I LOB file   I
-----
                2 I           3 I
    
```

## レポート内容

```
-----I-----I
      88 I          89 I
-----I-----I
```

各列は次の情報を示しています。

列	示す情報
File with LOB fields	LOB フィールドが含まれるファイルのファイル番号
Associated LOB file	実際の LOB フィールド値が保存されている LOB ファイルの番号

## データベースの物理レイアウト

次のセクションには、RABN シーケンスにおけるデータベースのすべてのスペース割り当てがリストされます。アソシエータに割り当てられている RABN が最初にリストされ、その後にデータストレージに割り当てられている RABN がリストされます。

```
*****
*
* Physical Layout of the Database *          yyyy-mm-dd  hh:mm:ss
*
*****

      From      To      Number  Dev   Table File VOLSER
      Blk       Blk    of Blks Type   Type      Number
-----
      131      -      162         32 3390    PPT      0 ADA001
      163      -      163          1 3390   DSST      0 ADA001
      164      -      164          1 3390    AC       19 ADA001
      165      -      174         10 3390    UI       19 ADA001
      175      -      224         50 3390    NI       19 ADA001
      225      -      242         18 3390    AC        2 ADA001
      243      -      243          1 3390   AC2        2 ADA001
      244      -      254         11 3390  UNUSED      0 ADA001
      255      -      259          5 3390    AC        1 ADA001
```



**Note:** 通常、物理レイアウト表が連続していない場合は、その箇所を指すエラーメッセージが表示されます。ただし、ファイルセーブの物理レイアウトの場合は表示されません。ファイルセーブではセーブされたファイルの FCB のみが含まれるため、物理レイアウト表は連続せず、このことは unknown (不明) 範囲としてレポートされます。

この表の各列は次の情報を提供します。

列	説明														
FROM BLK	論理エクステント内の最初のブロックの RABN。														
TO BLK	論理エクステント内の最後のブロックの RABN。														
NUMBER OF BLKS	エクステント内に含まれるブロック数。														
DEV TYPE	物理デバイスのタイプ。														
TABLE TYPE	割り当てが行われた要素。 <table border="1"> <tbody> <tr> <td>AC</td> <td>アドレスコンバータ</td> </tr> <tr> <td>NI</td> <td>ノーマルインデックス</td> </tr> <tr> <td>UI</td> <td>アップパーインデックス</td> </tr> <tr> <td>DS</td> <td>データストレージ</td> </tr> <tr> <td>DSF</td> <td>DSF ロギングエリア</td> </tr> <tr> <td>DSST</td> <td>データストレージスペーステーブル</td> </tr> <tr> <td>UNUSED</td> <td>使用可能なスペース</td> </tr> </tbody> </table>	AC	アドレスコンバータ	NI	ノーマルインデックス	UI	アップパーインデックス	DS	データストレージ	DSF	DSF ロギングエリア	DSST	データストレージスペーステーブル	UNUSED	使用可能なスペース
AC	アドレスコンバータ														
NI	ノーマルインデックス														
UI	アップパーインデックス														
DS	データストレージ														
DSF	DSF ロギングエリア														
DSST	データストレージスペーステーブル														
UNUSED	使用可能なスペース														
FILE	割り当てが行われたファイル。0は特定のファイルに関連しないエクステントを示します。														
VOLSER NUMBER	エクステントが含まれるボリュームのボリュームシリアル番号。JCLにデータストレージデータセットが存在する場合にのみ、データストレージに対して表示されます。														

## ファイル情報

### 一般特性

データベース内の各ファイルに関する詳細情報は、データベース情報の後に表示されます。この情報は、特定のファイルに限定したり、すべて省略することができます。このセクションの最初の部分には、ファイルの特性に関する情報が表示されます。スパンドレコードが含まれるファイルの例を次に示します。

```
*****
*
* File      2 (64BIT-1          ) *                yyyy-mm-dd  hh:mm:ss
*
*****

TOP-ISN      =                25          Highest Index Level =  3
```

## レポート内容

```
MAX-ISN Expected = 15,263 Padding Factor ASSO = 10%
Records Loaded = 25 Padding Factor DATA = 0%
MIN-ISN = 1 Length of Client NR = 0
TOP AC2 ISN = 100
MAX AC2 ISN Exp. = 848
MIN AC2 ISN = 1
Number of Updates = 0 ISNSIZE = 3

MAX COMP REC LEN = N/A Date Loaded = 2006-03-25
BLK/ADD DS EXT = 0 Time Loaded = 16:41:21
BLK/ADD UI EXT = 0
BLK/ADD NI EXT = 0
```

基本ファイルと LOB ファイルの組み合わせにおける基本ファイルの一般特性の例を次に示します。

```
*****
*
* File 2 (BASEFILE ) *
*
*
*****

TOP-ISN = 45 Highest Index Level = 0
MAX-ISN Expected = 1,377 Padding Factor ASSO = 10%
Records Loaded = 44 Padding Factor DATA = 10%
MIN-ISN = 1 Length of Client NR = 0
Number of Updates = 55,937 ISNSIZE = 3

MAX COMP REC LEN = 10,792 Date Loaded = 2006-09-01
BLK/ADD DS EXT = 0 Time Loaded = 14:16:57
BLK/ADD UI EXT = 0 Date of last update = 2007-01-23
BLK/ADD NI EXT = 0 Time of last update = 07:00:52

FILE ALPHA CODE = 37
FILE WIDE CODE = 4,095
```

```
USER WIDE CODE      =          4,095
```

基本ファイルとLOBファイルの組み合わせにおけるLOBファイルの一般特性の例を次に示します。

```
*****
*                               *
* File      3 (LOBFILE          ) *                               yyyy-mm-dd  hh:mm:ss
*                               *
*****

TOP-ISN          =          3      Highest Index Level =  3
MAX-ISN Expected =      1,377      Padding Factor ASSO = 10%
Records Loaded   =          0      Padding Factor DATA = 10%
MIN-ISN          =          1      Length of Client NR  =  0
Number of Updates =          0      ISNSIZE                =  3

MAX COMP REC LEN =      10,792      Date Loaded          = 2006-09-01
BLK/ADD DS EXT   =          0      Time Loaded          = 14:16:58
BLK/ADD UI EXT   =          0
BLK/ADD NI EXT   =          0

FILE ALPHA CODE  =      DB DEFAULT
FILE WIDE CODE   =      DB DEFAULT
USER WIDE CODE   =      DB DEFAULT
```

このレポートには次の情報が表示されます（上記のサンプルでは、これらすべてのフィールドが表示されない可能性があります）。

フィールド	説明
TOP-ISN	現在ファイルに割り当てられている最大 ISN。
MAX-ISN EXPECTED	ファイルに対して予想される最大 ISN。ADALOD ユーティリティの MAXISN パラメータを参照。
Records Loaded	ファイル内に現在含まれているレコード数。
MIN-ISN	ファイル内のレコードに対して割り当て可能な最小 ISN。ADALOD ユーティリティの MINISN パラメータを参照。
TOP AC2 ISN	ファイル内で使用中の最大セカンダリ ISN。スパンドレコードがファイルに対してアクティブな場合のみ、この統計値が与えられます。
MAX AC2 ISN EXP	ファイル内で予想される最大セカンダリ ISN。スパンドレコードがファイルに対してアクティブな場合のみ、この統計値が与えられます。
MIN AC2 ISN	ファイル内の最小セカンダリ ISN。スパンドレコードがファイルに対してアクティブな場合のみ、この統計値が与えられます。
Number of Updates	ファイルのロード後にファイルに適用された更新の回数。

## レポート内容

フィールド	説明
MAX COMP REC LEN	ファイルに対して許可される最大圧縮レコード長。ADALODユーティリティのMAXRECLパラメータを参照。このファイルに対してスパンドレコードサポートが有効な場合は、このフィールドの値は"N/A"と表示されます。
BLK/ADD DSEXT	各セカンダリデータストレージエクステントに割り当て可能な最大ブロック数。ADALODユーティリティのMAXDSパラメータを参照。
BLK/ADD UI EXT	各セカンダリアップパーインデックスエクステントに割り当て可能な最大ブロック数。ADALODユーティリティのMAXUIパラメータを参照。
BLK/ADD NI EXT	各セカンダリノーマルインデックスエクステントに割り当て可能な最大ブロック数。ADALODユーティリティのMAXNIパラメータを参照。
FILE ALPHA CODE	ファイル内の英数字フィールドに対する現在のファイルエンコードセット。UES=NOの場合には、この情報は表示されません。
FILE WIDE CODE	ファイル内のワイド文字フィールドに対する現在のファイルエンコードセット。UES=NOの場合には、この情報は表示されません。
USER WIDE CODE	ファイル内のワイド文字フィールドに対する現在のユーザーエンコードセット。UES=NOの場合には、この情報は表示されません。
HIGHEST INDEX LEVEL	ファイルに対して現在アクティブな最大インデックスレベル。
PADDING FACTOR ASSO	アソシエータパディングファクタ。詳細については、ADALOD LOAD ASSOPFACパラメータまたはADAORD REORASSO機能およびREORFASSO機能のセクションを参照してください。
PADDING FACTOR DATA	データストレージパディングファクタ。詳細については、ADALOD LOAD DATAPFACパラメータまたはADAORD REORDATA機能およびREORFDATA機能のセクションを参照してください。  <b>注意:</b> スパンドレコードが使用される場合は、ブロック全体が使用されるため、パディングファクタは無視されます。したがって、このレポートでは0と表示されることがよくあります。パディングファクタは、スパンドレコードの最後の短いセグメントのみに使用されます。
LENGTH OF CLIENT NR	マルチクライアントファイルのオーナーIDの長さ。
ISNSIZE	ファイルに含まれるISNが3バイトであるか4バイトであるか。
DATE LOADED	ファイルがロードされた日付。
TIME LOADED	ファイルがロードされた時刻。
DATE OF LAST UPDATE	ファイルが最後に変更された日付。
TIME OF LAST UPDATE	ファイルが最後に変更された時刻。

## オプション

ファイルのファイルオプション設定が次に表示されます。スパンドレコードが使用される場合の例を次に示します。

```

ADAM File           No
Ciphered File       No
ISN Reusage         No
Space Reusage       Yes
Coupled Files       None
Expanded File       No
USERISN             No
NOACEXTENSION       No
MIXDSDEV            No
PGMREFRESH          No
Multi Client File   No
Privileged usage    No
Online INVERT       None
Index Compressed    No
Spanned Rec Supp    Yes
Two Byte MU/PE      No

```

ADABAS version needed for this file: V71 or later

基本ファイルと LOB ファイルの組み合わせにおける基本ファイルの例を次に示します。ファイルに LB フィールドが含まれていることが示されています。

```

ADAM File           No
Ciphered File       No
ISN Reusage         Yes
Space Reusage       Yes
Coupled Files       None
Expanded File       No
USERISN             No
NOACEXTENSION       No
MIXDSDEV            No
PGMREFRESH          No
Multi Client File   No
Privileged usage    No
Online INVERT       None
Index Compressed    Yes
Spanned Rec Supp    No
Two Byte MU/PE      No

```

## レポート内容

```
LOB file                No
Contain LOB fields Yes
```

基本ファイルとLOBファイルの組み合わせにおけるLOBファイルの例を次に示します。ファイル自体がLOBファイルであることが示されています。

```
ADAM File                No
Ciphpered File          No
ISN Reusage             Yes
Space Reusage           Yes
Coupled Files           None
Expanded File           No
USERISN                 No
NOACEXTENSION           No
MIXDSDEV                No
PGMREFRESH              No
Multi Client File      No
Privileged usage       No
Online INVERT           None
Index Compressed       Yes
Spanned Rec Supp       No
Two Byte MU/PE         No
LOB file                Yes
Contain LOB fields No
```

フィールド	説明
ADAM File	ファイルが ADAM オプションでロードされたかどうかを示します。
Ciphpered File	ファイルが暗号化オプションでロードされたかどうかを示します。
ISN Reusage	ファイル ISN を再利用できるかどうかを示します。
Space Reusage	ファイルのデータストレージスペースを再利用できるかどうかを示します。
カップリングファイル	このファイルが物理的にカップリングされている対象のファイルを示します。
Expanded File	ファイルが拡張ファイルの一部であるかどうか、および一部である場合はその拡張ファイルのファイル番号が表示されます。
USERISN	ファイルが USERISN オプションでロードされたかどうかを示します。
NOACEXTENSION	ファイルで MAXISN 設定の増加が許可されるかどうかを示します。
MIXDSDEV	ファイルのデータストレージエクステンツが異なるデバイスタイプで許可されるかどうかを示します。
PGMREFRESH	ファイルを E1 コマンドでリフレッシュできるかどうかを示します。
Multiclient File	複数のオーナー/オーナー ID に属するレコードをファイル内に含めることができるかどうかを示します。
Privileged usage	ファイルが特権使用のためにニュークリアスによってロックされているかどうかを示します。ロックされている場合は、Adabas ユーティリティのみがファイルへのアクセスを許可されます。
Online INVERT	オンラインでインバートされるディスクリプタを示します。

フィールド	説明
Index Compressed	ファイルインデックスが圧縮されているかどうかを示します。
Spanned Rec Supp	ファイルのスパンドレコードサポートがアクティブであるかどうかを示します。
Two Byte MU/PE	ファイルの2バイトのMU/PE インデックス（MU/PE の発生が 191 を超えた場合）がアクティブであるかどうかを示します。
LOB ファイル	ファイルが LOB ファイルであるかどうかを示します。
Contain LOB fields	ファイルに1つ以上の LB フィールドが含まれる（このファイルが基本ファイルである）かどうかを示します。

## DELTA SAVE 変更フラグ

Delta Save Facility がデータベースにインストールされており、DSF ロギングが有効である場合、ADAREP は各ファイルに対して DELTA SAVE 変更フラグを表示します。

```

DELTA SAVE CHANGE FLAGS:
  SAVE ENTIRE INDEX      = [YES | NO]
  SAVE ENTIRE ADDR CONV = [YES | NO]
  SAVE ENTIRE DATA STOR = [YES | NO]
  TOTAL CHANGES BY UTILITIES = nnn BLOCKS

```

各フラグは、ファイルのすべてのインデックス、アドレスコンバータ、またはデータストレージがそれぞれユーティリティによって変更されたかどうか、および次のデルタセーブ処理で全体的にセーブされるかどうかを示します。

"TOTAL CHANGE BY UTILITIES" には、ADALOD UPDATE の実行で変更されたブロックだけでなく、全体的にセーブされるエクステンツ内のブロックが含まれます。

## スペース割り当て

次のセクションには、ファイルのスペース割り当てがリストされます。スパンドレコードが使用される場合のスペース割り当てを示す例を次に示します。

List	I	Dev	Block	I	Space	Alloc.	I	From	To	I	Unused	Space	I
Type	I	Type	Lngh	I	Blocks	Cyl	I	RABN	RABN	I	Blocks	Cyl	I
	I		I				I			I			I
-----	I	-----	I	-----	I	-----	I	-----	I	-----	I	-----	I
	I		I				I			I			I
AC	I	3390	2544	I	18	0I		225	242I				I
AC2	I	3390	2544	I	1	0I		243	243I				I
NI	I	3390	2544	I	500	1I		524	1023I		499		1I
UI	I	3390	2544	I	100	0I		1024	1123I		98		0I

## レポート内容

DSST	I	3390	2544	I	1	0	I	163	163	I		
	I		I				I		I			
DS	I	3390	5064	I	1500	10	I	473	1972	I	1397	9
	I		I				I		I			

基本ファイルと LOB ファイルの組み合わせにおける基本ファイルのスペース割り当ての例を次に示します。

List	Type	Dev	Block	Length	Space	Alloc.	From	To	Unused	Space		
					Blocks	Cyl	RABN	RABN	Blocks	Cyl		
AC	I	8391	4136	I	1	0	I	1717	1717	I		
NI	I	8391	4136	I	10	0	I	1988	1997	I	10	0
UI	I	8391	4136	I	5	0	I	1718	1722	I	5	0
FDT	I	8391	4136	I	4	0	I	335	338	I		
DSST	I	8391	4136	I	1	0	I	1563	1563	I		
DS	I	8391	10796	I	10	0	I	176	185	I	9	0

基本ファイルと LOB ファイルの組み合わせにおける LOB ファイルのスペース割り当ての例を次に示します。

List	Type	Dev	Block	Length	Space	Alloc.	From	To	Unused	Space		
					Blocks	Cyl	RABN	RABN	Blocks	Cyl		
AC	I	8391	4136	I	1	0	I	1998	1998	I		
NI	I	8391	4136	I	10	0	I	1732	1741	I	9	0
UI	I	8391	4136	I	5	0	I	1999	2003	I	3	0
FDT	I	8391	4136	I	4	0	I	339	342	I		
DSST	I	8391	4136	I	1	0	I	1563	1563	I		
DS	I	8391	10796	I	100	1	I	403	502	I	99	1

I	I	I	I	I
---	---	---	---	---

スペース割り当て表には、次の情報が表示されます。

列	説明
LIST TYPE	データベースコンポーネント。各コードは以下を表します。
AC	アドレスコンバータ
AC2	セカンダリアドレスコンバータエクステンツ (スパンドレコード用)
NI	ノーマルインデックス
UI	アップパーインデックス
DS	データストレージ
DSF	ファイル固有の DSF ロギングエリア
DSST	データストレージスペーステーブル
UNUSED	使用可能なスペース
DEV TYPE	コンポーネントが含まれる物理デバイス。
BLOCK LENGTH	ブロック長はコンポーネントとデバイスタイプに応じて異なります。
SPACE ALLOC.	コンポーネントに割り当てられたブロックおよびシリンダの総数。"0" は 1 つの完全なシリンダに満たないことを示します。
FROM RABN	論理エクステンツ内の最初のブロックの RABN。
TO RABN	論理エクステンツ内の最後のブロックの RABN。
UNUSED SPACE	現在使用されていないブロック数およびシリンダ数。"0" は 1 つの完全なシリンダに満たないことを示します。

### フィールド定義テーブル

フィールド定義テーブル (FDT) が次に表示されます。この情報は省略可能です。レポートの FDT セクションの一般的な例を次に示します。

FIELD DESCRIPTION TABLE									
LEVEL	I	I	I	I	I	I	I	I	I
	NAME	LENGTH	FORMAT	OPTIONS	PARENT	OF			
1	AA	8	A	DE,UQ					
1	AB								
2	AC	20	A	NU					
2	AE	20	A	DE				SUPERDE	PHONDE

## レポート内容

I	2	I	AD	I	20	I	A	I	NU	I		
	1	I	AF	I	1	I	A	I	FI	I		I
	1	I	AG	I	1	I	A	I	FI	I		I
	1	I	AH	I	6	I	U	I	DE	I		I
	1	I	A2	I		I		I		I		I
	1	I	AO	I	6	I	A	I	DE	I	SUBDE , SUPERDE	I
	1	I	AQ	I		I		I	PE	I		I
	2	I	AR	I	3	I	A	I	NU	I	SUPERDE	I
	2	I	AS	I	5	I	P	I	NU	I	SUPERDE	I
	1	I	A3	I		I		I		I		I
	2	I	AU	I	2	I	U	I		I	SUPERDE	I
	2	I	AV	I	2	I	U	I	NU	I	SUPERDE	I

基本ファイルとLOBファイルの組み合わせにおける基本ファイルに関連するFDTの一部の例を次に示します。基本ファイル内のLBフィールドが示されています。

Field Description Table

Level	I	I	I	I	I	I	I	I	I
	Name	Length	Format	Options	Parent of				
1	AA	6	A	NU					
1	AP	2	P	NU					
1	A1	0	A	NU					
1	A2	0	A	NU, NV					
1	A3	0	A	MU, NU					
1	A4	0	A	MU, NU, NV					
1	A5	0	A	NC					
1	A6	0	A	NC, NV					
1	B1	0	A	LA, NU					
1	B2	0	A	LA, NB, NU					
1	B3	0	A	LA, NU, NV					
1	B4	0	A	LA, NB, NU, NV					
1	B5	0	A	LA, MU, NU					
1	B6	0	A	LA, NB, MU, NU					
1	B7	0	A	LA, MU, NU, NV					
1	B8	0	A	LA, NB, MU, NU, NV					
1	C1	0	A	LB, NU					
1	C2	0	A	LB, NB, NU					
1	C3	0	A	LB, NU, NV					
1	C4	0	A	LB, NB, NU, NV					
1	C5	0	A	LB, MU, NU					
1	C6	0	A	LB, NB, MU, NU					
1	C7	0	A	LB, MU, NU, NV					
1	C8	0	A	LB, NB, MU, NU, NV					
1	D1	0	W	NU					

1	I	D2	I	0	I	W	I	NU,NV	I	I
1	I	D3	I	0	I	W	I	MU,NU	I	I
1	I	D4	I	0	I	W	I	MU,NU,NV	I	I
1	I	D5	I	0	I	W	I	NC	I	I
1	I	D6	I	0	I	W	I	NC,NV	I	I
1	I									

FDT セクションは LOB ファイルに対しては出力されません。

フィールド	説明	
LEVEL	フィールドレベル。	
NAME	フィールド名。	
LENGTH	フィールド長 (バイト数)。	
FORMAT	フィールドのデータタイプ。各コードは以下を表します。	
	A	英数字
	B	バイナリ
	F	固定小数点
	P	パック 10 進数
	G	浮動小数点
	U	アンパック 10 進数
	W	ワイド文字
OPTIONS	DE	ディスクリプタ
	FI	固定ストレージ
	LA	ロング英数字
	LB	ラージオブジェクトフィールド
	MU	マルチプルバリューフィールド
	NB	空白圧縮なし
	NC	空値/カウントなし
	NN	空値の指定不可
	NU	空値省略
	NV	変換なし (英字フィールドとワイド文字フィールド)
	PE	ピリオディックグループ。ピリオディックグループを構成するフィールドがこの後に続き、より大きなレベル番号を持ちます。
	UQ	ユニークディスクリプタ
XI	PE の UQ から除外されるインデックス (オカレンス) 番号	
PARENT OF	このフィールドが照合ディスクリプタ、サブ/スーパーフィールド、サブ/スーパーディスクリプタ、ハイパーディスクリプタ、またはフォネディックディスクリプタの親フィールドであるかどうかを示します。	

## 特殊ディスクリプタ

次のセクションには、ファイル内の特殊ディスクリプタ（照合ディスクリプタ、サブディスクリプタ、サブフィールド、スーパーディスクリプタ、スーパーフィールド、フォネティックディスクリプタ、およびハイパーディスクリプタ）に関する情報が表示されます。

```

SPECIAL DESCRIPTOR TABLE
      I      I      I      I      I      I      I      I
TYPE I NAME I LENGTH I FORMAT I OPTIONS I STRUCTURE I
      I      I      I      I      I      I      I      I
-----I-----I-----I-----I-----I-----I-----I
      I      I      I      I      I      I      I      I
SUPER I  H1 I    4 I    B I    DE,NU I  AU ( 1 - 2 ) I
      I      I      I      I      I      I      I      I
SUB I   S1 I    4 I    A I    DE I   AO ( 1 - 4 ) I
SUPER I  S2 I   26 I    A I    DE I   AO ( 1 - 6 ) I
      I      I      I      I      I      I      I      I
SUPER I  S3 I   12 I    A I    DE,NU,PE I AR ( 1 - 3 ) I
      I      I      I      I      I      I      I      I
      I      I      I      I      I      I      I      I
PHON I   PH I    I      I      I      I      I      I
      I      I      I      I      I      I      I      I
      I      I      I      I      I      I      I      I
COL I   Y1 I   20 I    W I    DE I   CDX 8,PA I
COL I   Y2 I   12 I    A I    DE,NU,PE I CDX 1,AR I
      I      I      I      I      I      I      I      I
      I      I      I      I      I      I      I      I
-----I-----I-----I-----I-----I-----I-----I
    
```

各特殊ディスクリプタの名前、長さ、形式とともに、この表には次の情報が表示されます。

列	説明	
TYPE	SUB	サブフィールド／サブディスクリプタ
	SUPER	スーパーフィールド／スーパーディスクリプタ
	PHON	フォネティックディスクリプタ
	HYPHER	ハイパーディスクリプタ
	COL	照合ディスクリプタ
OPTIONS	DE	ディスクリプタフィールド
	FI	固定小数点
	LA	ロング英数字
	MU	マルチプルバリューフィールド
	NC	空値のカウントなし（SQL 空値表現）
	NN	空値の指定不可

列	説明										
	<table border="1"> <tr> <td>NU</td> <td>空値省略</td> </tr> <tr> <td>NV</td> <td>変換なし (英字フィールドとワイド文字フィールド)</td> </tr> <tr> <td>PE</td> <td>ピリオディックグループ</td> </tr> <tr> <td>UQ</td> <td>ユニークディスクリプタ</td> </tr> <tr> <td>XI</td> <td>PE の UQ から除外されるインデックス (オカレンス) 番号</td> </tr> </table>	NU	空値省略	NV	変換なし (英字フィールドとワイド文字フィールド)	PE	ピリオディックグループ	UQ	ユニークディスクリプタ	XI	PE の UQ から除外されるインデックス (オカレンス) 番号
NU	空値省略										
NV	変換なし (英字フィールドとワイド文字フィールド)										
PE	ピリオディックグループ										
UQ	ユニークディスクリプタ										
XI	PE の UQ から除外されるインデックス (オカレンス) 番号										
STRUCTURE	サブ、スーパー、またはハイパーディスクリプタの構成フィールドおよびフィールドバイト。フォネティックディスクリプタは同等の英数字エレメンタリフィールドを示します。照合ディスクリプタは関連付けられた照合ディスクリプタユーザー出口と親フィールドの名前を示します。										

## チェックポイント情報

CPLIST パラメータまたは CPEXLIST パラメータを指定すると、チェックポイント情報も表示されます。

```

*****
* CHECK-POINT-LIST *                               yyyy-mm-dd hh:mm:ss
*****

CP          CP          DATE          TIME          PLOG          BLOCK          JOBNAME
NAME        TYPE
           USER TYPE
           VOLSER NR....

SYNP        30          1995-06-03  14:07:38  47           1              DUAL GA0TB1
           LOAD
           VOLSER = WRK001
SYNC        01  ET  1995-06-03  14:08:16  48           2              DUAL GANUC70A
           SESSION OPEN IGNDIB=N FORCE=N
SYNP        1C  UTI  1995-06-03  14:08:36  48           3              DUAL GA0TB1
           RESTRUCT
    
```

この表の各列は次の情報を提供します。

列	説明	
CP-NAME	チェックポイントID。ユーザー非同期チェックポイントの場合は、ユーザープログラムによって指定されるチェックポイント ID になります。"SYN" で始まるチェックポイント名は、Adabas ニュークリアスおよびユーティリティ用に予約されています。	
	SYNC	ニュークリアスの初期化中に作成された同期チェックポイント。ADARUN IGNDIB パラメータおよび FORCE パラメータのステータスが含まれます。
	SYNF	1 つまたは複数のファイルの排他 (EXF) 制御を必要とするユーザープログラムまたはユーティリティによって指定されたチェックポイント。
	SYNP	特権制御を必要とするユーティリティによって指定されたチェックポイント。このようなユーティリティでは、Adabas ニュークリアスを使用しなくても更新を実行できます。
	SYNS	Adabas Online System (SYSAOS) または ADADBS によって取得されたチェックポイントであり、ニュークリアスによる 3 つの例外が含まれます。このチェックポイントで識別される機能は、再生成時にユーザーの介在なしに実装されます。  例外には、ニュークリアスセッションの最後で記録される 2 つ目の SYNS 5B、ADARUN INTNAS パラメータによって指定された間隔で記録される SYNS 60、およびファイルにさらにスペースが割り当てられるときに記録される SYNS 61 があります。
	SYNV	データセットへの順次書き込み中に変更されたボリューム ID がクローズされたことを示します。
	SYNX	1 つ以上のファイルの排他制御 (EXU) を必要とするユーティリティから与えられたチェックポイント。
	SYN1	オンラインの ADASAV 実行 (SAVE データベース 機能) の開始時に作成されたチェックポイント。
	SYN2	オンラインの ADASAV 実行 (SAVE データベース 機能) の終了時に作成されたチェックポイント。
	SYN4	オンラインの ADASAV 実行 (SAVE ファイル操作) の開始時に作成されたチェックポイント。
	SYN5	オンラインの ADASAV 実行 (SAVE ファイル操作) の終了時に作成されたチェックポイント。
CP TYPE	チェックポイント番号。使用可能なチェックポイント番号については、次のチェックポイントの表を参照。	
USER TYPE	チェックポイントを設定した Adabas ユーザーのタイプ。ユーザータイプは次のとおりです。	
	ET	ET ユーザー

列	説明	
	EXF	排他的ファイル制御ユーザーまたはユーティリティ（特権ユーザー）
	EXU	排他的ファイル更新ユーザーまたはユーティリティ
	UTI	ユーティリティ更新制御ユーティリティ（特権ユーザー）
	UTS	オンライン ADASAV SAVE ファイル（特権ユーザー）
DATE TIME	チェックポイントが指定された日付および時刻。	
PLOG NR.	チェックポイントがチェックポイントファイルに書き込まれたときに使用されていたデータプロテクションログの番号。	
BLOCK NR.	チェックポイントが書き込まれたデータプロテクションログのブロック番号。	
VOLSER-NUMBER	シーケンシャルプロテクション（DD/SIBA）ログのボリュームシリアル番号。ボリュームシリアル番号は、デュアルログを使用する場合は "DUAL"、マルチログを使用する場合は "MULTI" になります。	
JOBNAME	チェックポイントを作成したジョブの名前。	

次の表に、Adabas ニュークリアスまたはユーティリティによって書き込まれるチェックポイントについての説明を示します。

タイプ	名前	発行元	説明
01	SYNC	ADANUC	ニュークリアスセッションの開始時にニュークリアスによって書き込まれます。
01	SYNF	ユーザー/ユーティリティ	EXF（排他使用）モードで使用されるファイルによるユーザー/ユーティリティセッションの OPEN。
01	SYNX	EXU ユーザー	EXU ユーザーのオープン。
02	SYNV	ADANUC	VOLSER エントリ。シーケンシャルログが使用される場合に、DD/SIBA 上でのボリューム切り換え時およびセッションの終了時に書き込まれます。
03	SYNF	ユーザー/ユーティリティ	EXF ユーザーのチェックポイントのクローズ。
03	SYNX	EXU	EXU ユーザーのチェックポイントのクローズ。
05	SYNP	ADASAV	ファイルの SAVE（オペレーションの開始）
06	SYNP	ADASAV	データベースの SAVE（オペレーションの開始）
07	SYNP	ADASAV	ファイルの RESTORE（オペレーションの終了）
08	SYNP	ADASAV	RESTPLOG（オペレーションの終了）
09	SYNV	ADASAV	ファイルの SAVE、VOLSER エントリ。DD/SAVE 上でのボリューム変更時および SAVE 操作の終了時に書き込まれます。
0A	SYNV	ADASAV	データベースの SAVE、VOLSER エントリ。DD/SAVE 上でのボリューム切り換え時および SAVE 操作の終了時に書き込まれます。

## レポート内容

タイプ	名前	発行元	説明
0B	SYNP	ADASAV	SAVE DELTA (オペレーションの終了)
0C	SYNP	ADASAV	RESTORE DELTA (オペレーションの終了)
0D	SYNP	ADASAV	MERGE (オペレーションの終了)
0E	SYNV	ADASAV	SAVE DELTA、VOLSER エントリ
0F	SYNV	ADASAV	MERGE、VOLSER エントリ
10	SYNP	ADAINV	ファイルの COUPLE
11	SYNP	ADAINV	フィールドの INVERT
15	SYNP	ADAORD	アソシエータデータベースの REORDER
16	SYNP	ADAORD	データストレージデータベースの REORDER
17	SYNP	ADAORD	データベースの REORDER
18	SYNP	ADAORD	アソシエータファイルの REORDER
19	SYNP	ADAORD	データストレージファイルの REORDER
1A	SYNP	ADAORD	ファイルの REORDER
1B	SYNP	ADAORD	STORE
1C	SYNP	ADAORD	RESTRUCTURE
1D	SYNP	ADADEF	DEFINE NEWWORK
1E	SYNP	ADADEF	デフォルト文字エンコードの MODIFY
22	SYNX	ADARES	ファイルの REGENERATE
23	SYNX	ADARES	ファイルの BACKOUT
24	SYNX	ADARES	全ファイルの REGENERATE。CPEXLIST に除外ファイルがリストされます。
25	SYNX	ADARES	全ファイルの BACKOUT。CPEXLIST に除外ファイルがリストされます。
26	SYNP	ADARES	データストレージの REPAIR
27	SYNV	ADARES	シーケンシャルプロテクションログの COPY
28	SYNP	ADARES	PLCOPY 機能の正常終了
28	SYNV	ADARES	デュアル/マルチプロテクションログの PLCOPY
29	SYNV	ADARES	デュアル/マルチコマンドログの CLCOPY
2A	SYNP	ADARES	PLCOPY MERGE 機能の正常終了
2A	SYNV	ADARES	デュアル/マルチプロテクションログの PLCOPY MERGE
2B	SYNP	ADARES	CLOG MERGE 機能の正常終了
2B	SYNV	ADARES	デュアル/マルチコマンドログの CLOG MERGE
30	SYNP	ADALOD	ファイルの LOAD
31	SYNP	ADALOD	大量更新
35	SYNX	ADAULD	ファイルのアンロード
3F	SYNP	ADAZAP	正常な VERIFY - REPLACE
40	SYNS	SYSAOS	エクステンツの追加

タイプ	名前	発行元	説明
41	SYNS	SYSAOS	デフォルトフィールド長の CHANGE
42	SYNS	SYSAOS	データベースサイズの DECREASE
44	SYNS	SYSAOS	ファイルの削除
45	SYNS	SYSAOS	データベースサイズの INCREASE
47	SYNS	SYSAOS	スペースの RECOVER
48	SYNS	SYSAOS	ファイルのリフレッシュ
49	SYNS	SYSAOS	コンポーネントファイルを拡張ファイルチェーンから削除
4A	SYNS	SYSAOS	ディスクリプタの解放
4B	SYNS	SYSAOS	ファイルの RENAME
4C	SYNS	SYSAOS	ファイルの RENUMBER
4D	SYNS	SYSAOS	RESET DIB
4E	SYNS	SYSAOS	ISN の再利用
4F	SYNS	SYSAOS	データストレージの再利用
50	SYNS	SYSAOS	ファイルの UNCOUPLE
51	SYNS	SYSAOS	ファイルエクステンットの ALLOCATE
52	SYNS	SYSAOS	ファイルエクステンットの DEALLOCATE
53	SYNS	SYSAOS	チェックポイントの削除
54	SYNS	SYSAOS	ユーザープライオリティの設定
55	SYNS	SYSAOS	FCB の修正
57	SYNS	SYSAOS	ファイルの DEFINE
58	SYNS	SYSAOS	FDT の書き込み
59	SYNS	SYSAOS	新フィールドの DEFINE
5B	SYNS	ADADBS	リフレッシュされた統計の書き込み（ユーザー要求ごとに一部またはすべて）
5B	SYNS	ADANUC	ニュークリアスセッション終了時に（全）統計の書き込み
5B	SYNS	ADARES	リフレッシュされた統計の書き込み（コマンド、ファイル、およびスレッド使用、DRES および DSTAT）
5C	SYNS	SYSAOS	デフォルトフィールドフォーマットの CHANGE
5D	SYNS	SYSAOS	ファイルエンコードの変更
5E	SYNS	ADADBS	ADADBS REPTOR 機能（Event Replicator for Adabas のドキュメントを参照）
60	SYNS	ADANUC	ニュークリアス統計チェックポイント
61	SYNS	ADANUC	ファイルスペースの割り当て
64	SYNS	ADASCR	ファイルのプロテクト
65	SYNS	ADASCR	フィールドのプロテクト
66	SYNS	SYSAOS	コンポーネントファイルを拡張ファイルチェーンにリンク
68	SYNS	SYSAOS	USERISN のオン/オフ設定

## レポート内容

タイプ	名前	発行元	説明
69	SYNS	SYSAOS	MIXDSDEV のオン/オフ設定
6A	SYNS	SYSAOS	DELTA SAVE DLOG エリアのインストール
6B	SYNS	SYSAOS	DELTA SAVE DLOG エリアの変更
6C	SYNS	SYSAOS	DELTA SAVE DLOG エリアの削除
6E	SYNS	ADADBS	ADADBS REPLICATION 機能 (Event Replicator for Adabas のドキュメントを参照)
6F	SYNS	SYSAOS	オンライン処理の開始
70	SYNS	SYSAOS	オンラインインバート処理
71	SYNS	SYSAOS	オンラインリオーダー処理
73	SYNC	ADANUC	ニュークリアスの静止完了
74	SYNC	ADANUC	ニュークリアスの通常処理の再開
75	SYNS	ADANUC	ニュークリアスの開始時に検出されたレスポンス72よりも後の HEURI ユーザーエントリを削除
76	SYNS	ADANUC	ニュークリアスセッション中に検出されたレスポンス72よりも後の HEURI ユーザーエントリを削除
77	SYNS	ADADBS	スパンドレコードサポートの有効化
78	SYNS	ADADBS	拡張 MU/PE フィールドの有効化または無効化

# 153

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	850
▪ z/OS .....	851
▪ z/VM .....	852
▪ VSE .....	853

このセクションでは、BS2000、z/OS、z/VM、および VSE の各システムで ADAREP を実行するために必要なジョブ制御情報について説明し、それぞれのジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
セーブテープ	DDSAVE	テープ/ディスク	SAVETAPE 指定の場合のみ
プロテクションログ	DDPLOG	テープ/ディスク	オンラインセーブテープのオプション
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAREP パラメータ	SYSDTA/DDKARTE		
ADARUN メッセージ	SYSOUT DDPRINT		メッセージおよびコード
ADAREP メッセージ	SYSLST DDDRUCK		メッセージおよびコード

### ADAREP JCL の例 (BS2000)

SDF フォーマットの場合：

```

/.ADAREP LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E P ALL FUNCTIONS
/REMARK *
/ASS-SYSLST L.REP
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAREP,DB=yyyyyy,IDTNAME=ADABAS5B
ADAREP CPLIST
/LOGOFF SYS-OUTPUT=DEL
    
```

ISP フォーマットの場合：

```

/.ADAREP LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E P ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.REP
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAREP,DB=yyyyy,IDTNAME=ADABAS5B
ADAREP CPLIST
/LOGOFF NOSPOOL

```

## z/OS

データセット	DD名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
セーブテープ	DDSAVE	テープ/ディスク	SAVETAPE 指定の場合のみ
プロテクションログ	DDPLOG	テープ/ディスク	オンラインセーブテープのオプション
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAREP パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAREP メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

### ADAREP JCL の例 (z/OS)

すべての機能

```

//ADAREP JOB
//*
//* ADAREP: ALL FUNCTIONS
//*
//REP EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X

```

```
//DDCARD DD *
ADARUN PROG=ADAREP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAREP CPEXLIST
/*
```

この例については、MVSJOBS データセットの ADAREP を参照してください。

セーブテープからのレポート

```
//ADAREPS JOB
/*
/* ADAREP: REPORT FROM A SAVE TAPE
/*
//REP EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDSAVE DD DISP=SHR,DSN=EXAMPLE.DByyyyy.SAVE <=== SAVE DATASET
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADAREP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADAREP REPORT SAVETAPE,NOCOUNT
/*
```

この例については、MVSJOBS データセットの ADAREPS を参照してください。

## z/VM

データセット	DD 名	媒体	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
セーブテープ	DDSAVE	テープ/ディスク	SAVETAPE 指定の場合のみ
プロテクションログ	DDPLOG	テープ/ディスク	オンラインセーブテープのオプション
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAREP パラメータ	DDKARTE	ディスク/端末/リーダー	

データセット	DD名	媒体	追加情報
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAREP メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

## ADAREP の例 (z/VM)

```

DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1
DATADEF DDPRINT,DSN=ADAREP.DDPRINT,MODE=A
DATADEF DUMP,DUMMY

DATADEF DDDRUCK,DSN=ADAREP.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNREP.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAREP.CONTROL,MODE=A
ADARUN

```

## RUNREP CONTROL A1 の内容

```
ADARUN PROG=ADAREP,DEVICE=dddd,DB=yyyyyy
```

## ADAREP CONTROL A1 の内容

```
ADAREP CPLIST
```

## VSE

データセット	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSORn	ディスク		
データストレージ	DATARn	ディスク		
セーブテープ	SAVE	テープ ディスク	SYS010 注参照	SAVETAPE 指定の場合のみ
プロテクションログ	PLOG	テープ ディスク	SYS011 注参照	オンラインセーブテープのオプション
ADARUN パラメータ	SYSRDR CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 注参照	オペレーションマニュアル
ADAREP パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	メッセージおよびコード

データセット	記号名	ストレージ	論理ユニット	追加情報
ADAREP レポート		プリンタ	SYS009	



**Note:** すべてのプログラマ論理ユニットを使用できます。

## ADAREP JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

すべての機能

```
* $$ JOB JNM=ADAREP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAREP
*     ALL FUNCTIONS
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAREP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAREP    CPEXLIST
/*
/&
* $$ EOJ
```

この例については、メンバ ADAREPX を参照してください。

セーブテープからのレポート

```
* $$ JOB JNM=ADAREPS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAREPS
*     REPORT FROM A SAVE TAPE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYSTEM,TAPE
// PAUSE MOUNT LOAD SAVE FILE ON TAPE cuu
// TLBL SAVE,'EXAMPLE.DByyyyy.SAVE'
// MTC REW,SYS010
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAREP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAREP    REPORT SAVETAPE,NOCOUNT
/*
```

```
/&  
* $$ EOJ
```

この例については、メンバ ADAREPS.X を参照してください。



# 154 ADARES ユーティリティ：データベースリカバリ

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- BACKOUT 機能
- CLCOPY：デュアルコマンドログのコピー
- COPY：シーケンシャルプロテクションログまたはセーブテープのコピー
- MERGE CLOG：ニュークリアスクラスタコマンドログのマージ
- PLCOPY：プロテクションログのシーケンシャルデータセットへのコピー
- REGENERATE：更新の再生成
- REPAIR：データストレージブロックの修復
- マルチスレッド処理統計値
- JCL/JCS の必要項目と指定例



# 155 機能概要

---

■ ADARES の使用について .....	860
------------------------	-----

ADARES コーティリティは、データベースリカバリに関連する機能を実行します。

## ADARES の使用について

---

1. BACKOUT (BACKOUT DPLOG/MPLOG を除く)、REGENERATE、REPAIR の各機能では、プロテクションログデータを含むシーケンシャルデータセットが入力として必要です。デュアル/マルチプロテクションログを直接使用することはできません。デュアル/マルチプロテクションログをシーケンシャルログに変換するには、ADARESPLCOPY 機能を使用します。
2. REGENERATE、BACKOUT、COPY の各機能は、次の条件を満たしている場合に限り、1つのジョブ内で複数のセッションを処理できます。
  - DDSIIN/SIIN 入力ファイル内のセッション群はセッション番号で昇順に配列され、セッション番号に番号抜けがある場合は、それらのセッションがセーブデータセット生成を表すセッションのみであること
  - ファイルに対する各セッションはブロック番号 1 から始まり、ブロック番号の番号抜けがないこと
3. 1つのセッションのみを選択するには、PLOGNUM を使用するか、TOPLOG なしで FROMPLOG を使用してセッション番号を指定します。セッションの範囲を指定するには、FROMPLOG と TOPLOG を指定します。
4. ADARES COPY 機能は、ADASAV 出力セーブ (DD/SAVE) ファイルを対象としても実行できます。セーブファイルを指示するパラメータは指定できません。ADARES では、セーブファイルの構造によってセーブファイルを認識します。ADARES COPY の 1 回の実行では、1つのセーブファイルのみコピーできます。セーブファイルをコピーするには、PLOGNUM でそのセッション番号を指定します。
5. Adabas 拡張ファイルについて：(ファイルの) BACKOUT 機能および REGENERATE 機能では、拡張ファイル全体の処理のみが行われます。指定されたファイルが拡張ファイルのアンカーファイルまたはコンポーネントファイルである場合は、拡張ファイルのその他すべてのコンポーネントファイルも指定する必要があります。
6. ADARES の追加 Delta Save Facility 処理を実行するには、DD/CARD 入力に ADARUN パラメータ DSF=YES を指定する必要があります。
7. マルチスレッドの BACKOUT、BACKOUT DPLOG/MPLOG、および REGENERATE では、メインメモリを追加する必要があります。メモリの量は、対応するニュークリアス ADARUN パラメータ NT および NU の値を使用して、次の式で計算できます。

$$(NT \times 70,000) + (NU \times 72)$$

例えば NT = 28 で NU = 1000 の場合は、約 2 MB のメインメモリが必要になります。

8. マルチスレッドのバックアウト／再生機能の使用時に最適な処理を行うために、ニュークリアス ADARUN パラメータ NAB を次の式で求められる値よりも大きく設定することをお勧めします。

$$NAB = NT \times (32K + 108) / 4096$$

## Adabas ニュークリアスクラスタ環境での ADARES の使用

Adabas ニュークリアスクラスタ環境では、クラスタ内のすべての個々のニュークリアスのプロテクションログ（およびオプションでコマンドログ）は、全体としてすべてのニュークリアスによって共有されるクラスタデータベースの年代順でそれぞれ1つのログファイルにマージされます。年代順は、すべての個々のニュークリアスログレコードのタイムスタンプに基づいて決定されます。これらのタイムスタンプは、オペレーティングシステムによってクラスタ全体で同期が取られます。

### ログのマージ

リカバリ処理のために、すべてのプロテクションログデータセット（PLOG）は、クラスタデータベースごとに1つのログストリームにマージしなければなりません。PLOG は、ADARES PLCOPY が実行されるときに自動的にマージされます。PLCOPY プロセスは、Parallel Participant Table（PPT）にアクセスしてコピー対象のPLOGを決定し、ダイナミックアロケーションを使用して適切なデータセットにアクセスします。

既存の PLCOPY ジョブはクラスタ環境で実行するために修正しなければなりません。ユーザー出口2の修正も必要になる場合があります。必要な中間データセットの追加を示すサンプルの PLCOPY ジョブ ADARESMP およびサンプルのユーザー出口2（USEREX2P）が提供されています。「[ニュークリアスクラスタプロテクションログの自動的なコピー／マージ](#)」を参照してください。ただし、PLOGDD ステートメントを削除する必要はありません。これらのステートメントが残っている場合は、無視されます。

デフォルトでは、デュアル／マルチコマンドログデータセット（CLOG）は、ADARES CLCOPY 機能を使用して各ニュークリアスのシーケンシャルデータセットにコピーできますが、結果として得られたデータセットが、その後、クラスタ全体でクラスタデータベースの1つの CLOG データセットに自動的にマージされることはありません。ADARES MERGE CLOG 機能を使用すると、各ニュークリアスからの CLCOPY 出力を手動でマージできます。デフォルトでは、CLOG データセットをユーザー出口2 JCL に指定する必要があります。これらのデータセットは動的には割り当てられません。

ただし、会計処理をはじめとする追跡の目的で、PLOG マージプロセスを自動化する場合と同じ方法で CLOG マージプロセスを自動化すると便利です。ADARUN CLOGMRG=YES を指定すると、ADARES CLCOPY ジョブが UEX2 からサブミットされて実行されるときに、CLOG マージプロセスが自動的に起動されます。ADARUN LOGGING=YES も指定する必要があります。PLCOPY プロセスと同様に、CLCOPY プロセスはその後 Parallel Participant Table（PPT）にアクセスしてコピー対象の CLOG を決定し、ダイナミックアロケーションを使用して適切なデータセットにアクセスします。

既存の CLCOPY ジョブを修正して、中間データセットを含める必要があります。必要な中間データセットの追加を示すサンプルの CLCOPY ジョブ ADARESMC が提供されています。「[ニュークリアス クラスタ コマンド ログの自動的なコピー/マージ](#)」を参照してください。サンプルユーザー出口 2 (USEREX2P) には、マージのために CLCOPY と PLCOPY の両方の機能が含まれています。

自動化された PLCOPY ジョブおよび CLCOPY ジョブでは、可能な限り多くのデータがコピー/マージされます。ニュークリアスがまだログデータセットに書き込み中である場合は、データセットが "部分的に" マージされます。

### 中間データセット

マージは、マージされるすべての PLOG と CLOG の最小タイムスタンプで始まり、すべてのデータセットの中で最小の終了タイムスタンプで終わります。このポイントを超えるレコードは中間データセットに書き込まれます。それは後続のマージへの入力として供給しなければなりません。照合が実行され、正しい中間データセットが供給されたことを保証します。

ADARES では、マージされる PLOG または CLOG の少なくとも 1 つは "完了" ステータスであることを想定しています。これに該当しない場合、ADARES は、コピーするデータがないことをレポートします。

サンプルユーザー出口 2 (PLOG および CLOG 用の USEREX2P) は、中間データセットに必要な JCL を示すために提供されています。中間データセットが CLCOPY と PLCOPY の両方のジョブで使用される場合は、上書きされないように、それぞれのデータセット名をユニークな名前にしなければなりません。

#### ■ PLCOPY の例

```
//MERGIN1 DD DISP=SHR,DSN=EXAMPLE.PINTERI  
//MERGIN2 DD DISP=SHR,DSN=EXAMPLE.PINTERO
```

#### ■ CLCOPY の例

```
//MERGIN1 DD DISP=SHR,DSN=EXAMPLE.CINTERI  
//MERGIN2 DD DISP=SHR,DSN=EXAMPLE.CINTERO
```

PLCOPY であるか CLCOPY であるかに応じて、ユーザー出口 2 によってサブミットされるジョブは、適切なステートメントセットを参照する必要があります。

PLOG データセットの DD ステートメントは、一度セッション起動 JCL に指定すると、ダイナミックアロケーションでオープンされるようになるため、ADARES に対して再び指定する必要はありません。DD ステートメントが供給された場合、それらは無視されます。

実行されるたびに JCL を手動で変更する必要はありません。ADARES では、どの中間データセットが入力として予期されているかを判断するために、Parallel Participant Table (PPT)

で制御情報を保守します。ADARESでは、入力および出力にどちらを使用するかを決定するために、両方のデータセットのヘッダー情報をチェックします。

次のチェックは中間データセットが正しく供給されたことを確認するために行われます。

1. DBIDは中間データセットヘッダーに保存され、ログのDBIDと一致しなければなりません。
2. ログ番号は中間データセットヘッダーに保存されています。この番号は、ログデータセットの現在の番号か、またはログデータセットの現在の番号から1を減算した番号と一致する必要があります。
3. 中間データセットヘッダーのSTCKは、PPTに保存されているSTCKと一致する必要があります。

チェックのいずれかが失敗すると、ADARES ERROR 157が返されます。

ADARESでは、予期されたレコード数が中間データセットに含まれていることも確認します。含まれていない場合は、ADARES ERROR 164が返されます。

#### チェックポイントのユニークな識別

プロテクションログ (PLOG) マージプロセスの後、ブロック番号は必ずしも同じにはなりません。この状況で、チェックポイントをユニークに識別するために、TOBLK/FROMBLKパラメータを指定できるすべての機能 (つまり BACKOUT と REGENERATE) に NUCID も指定する必要があります。

マージプロセスは、ブロックにつき多くても1つのチェックポイントがあることを保証します。マージが実行される前の (古い) ブロック番号とチェックポイント記述時のNUCIDが記録されます。ADAREPでレポートされるようにブロック番号とNUCIDを指定すると、ADARESはユニークにブロックを識別することができます。

 **Note:** Adabas ニュークリアスクラスタ環境で、ADAREPは、すべてのチェックポイント情報を出力するとき、NUCIDを含んでいます。

Adabasニュークリアスクラスタ環境で必要とされる追加パラメータは、NUCID、TONUCID、およびFROMNUCIDです。NUCIDが開始および終了チェックポイントのものと同等であれば、NUCIDだけを指定する必要があります。

 **Note:** ADASAVでは、RESTONL機能およびRESTPLOG機能のためにブロックをユニークに識別できるように、この情報をヘッダーに保存します。



# 156 BACKOUT 機能

---

すべての更新レコードの更新前と更新後のイメージのデータプロテクション情報は、各 Adabas セッション中にプロテクションログに書き込まれます。この情報は、更新の取り消しや再適用に必要です。

プロテクションログは、ディスク上のシーケンシャルデータセットまたはデュアル/マルチプロテクションログデータセット（ダイレクトアクセス）に割り当てることができます。デュアル/マルチプロテクションログを使用する場合は、ADARES PLCOPY 機能を使用してシーケンシャルデータセットにコピーする必要があります。このデータセットは、ADARES BACKOUT または REGENERATE への入力として使用できます。

プロテクションログファイルに対して 3480/3490 テープカートリッジ圧縮（IDRC）を使用することはお勧めしません。ADARES ユーティリティの BACKOUT 機能は、圧縮データを処理するときに、z/OS では少なくとも 2 回実行されます。また、BACKOUT 機能では VSE システムおよび z/VM システムでの圧縮データがサポートされていません。

ADARES BACKOUT {DPLOG | MPLOG} 機能はクラスタデータベースに有効ではありません。マージされた PLOG が BACKOUT を実行するために必要とされるので、これは禁止されます。

**BACKOUT** シーケンシャルプロテクションログ（SIBA）を使用した更新のバックアウト

**BACKOUT DPLOG/MPLOG** デュアル/マルチプロテクションログを使用した更新のバックアウト



# 157 BACKOUT：シーケンシャルプロテクションログ (SIBA) を使用した更新のバックアウト

---

■ 基本パラメータ .....	869
■ オプションのパラメータとサブパラメータ .....	869
■ 例 .....	873

## BACKOUT：シーケンシャルプロテクションログ（SIBA）を使用した更新のバックアウト

---

BACKOUT 機能は、指定された2つのチェックポイント間に適用されたすべての更新を取り消します。両方のチェックポイントがシーケンシャルプロテクションログの入力データセットに存在する必要があります。

BACKOUT 機能では、シーケンシャル入力に使用するテープドライブで逆順読み込みがサポートされていることを前提としています。



**Note:** 中断した BACKOUT 実行は、最初から再実行しなければなりません。

BACKOUT 処理の開始点として、プロテクションログのログ番号（PLOGNUM）またはセッション番号（FROMPLOG）のどちらかを指定することもできます。セッション番号を指定する場合は、TOPLOG パラメータを使用すると、処理するセッションの範囲も指定できます。

デフォルトでは、ADARES は ADARUN DBID パラメータで指定されたデータベースを処理します。別のデータベースに対する BACKOUT 処理が必要な場合は、PLOGDBID パラメータを使用してデータベースを指定してください。

デフォルトでは、BACKOUT 処理は入力データセットの最後に到達するまで続きます。TOCP パラメータを使用することによって、BACKOUT 処理の範囲を制限できます。

デフォルトでは、指定した入力データセットのすべてのファイルが BACKOUT 処理の対象になります。特定のファイルを処理の対象に指定することができます。

BACKOUT 処理終了時、データベース全体に BACKOUT が指定されると、ADARES は、すべての不完全な論理トランザクションを自動的にバックアウトし、入力データセットが終了するまで続きます。このことは、次に該当する場合にも発生します。

- FILE パラメータおよび CONTINUE パラメータの2つが指定された場合
- TOCP パラメータが指定されなかった場合

NOAUTOBACKOUT パラメータを指定してこの処理を無効にすることができます。

```
ADARES BACKOUT { PLOGNUM = protection-log-number  
                  FROMPLOG = start-session [, TOPLOG = stop-session ] }  
[EXCLUDE = file-list ]  
[FILE = file-list [CONTINUE] ]  
[FROMMCP = checkpoint-name [, FROMBLK = checkpoint-block ]]  
[IGNORECOUPLE]  
[IGNOREEXP]  
[MTR = { YES | NO } [NPCALLS = maximum-number-of-parallel-calls ]]  
[NOAUTOBACKOUT]  
[NOUSERABEND]  
[PARALLELREAD]  
[PLOGDBID = { alternate-log-dbid | ADARUN-dbid }]  
[TEST]  
[TOCP = checkpoint-name [, TOBLK = checkpoint-block ]]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### PLOGNUM：プロテクションログ番号

PLOGNUMは、BACKOUT処理の入力として使用するシーケンシャルプロテクションログのログ番号を指定します。ログ番号は、データベースステータスレポートから得ることができます。

### FROMPLOG：BACKOUTの開始セッション

FROMPLOGは、BACKOUT処理を開始するセッション番号を指定します。ADARESはシーケンシャルPLOG入力（DD/SIIN）ファイル内で正しい開始セッションを検索します。開始点をさらに明確に定義したい場合には、FROMMCP および FROMBLK パラメータを指定します。

## オプションのパラメータとサブパラメータ

### CONTINUE：オートバックアウトオプション付きのファイルリカバリ

CONTINUEは、BACKOUT機能による排他使用のためにデータベース全体をロックします。

CONTINUEは、不完全なトランザクションによってファイルの内容が変更されていた場合、それらをファイルバックアウトの際に自動バックアウトします。CONTINUEを指定すれば、FILEパラメータによって指定されたデータセットの不完全トランザクションによるすべての変更がバックアウトされます。

## BACKOUT：シーケンシャルプロテクションログ（SIBA）を使用した更新のバックアウト

---

ファイルリスト中にカップリングファイルまたは拡張ファイルのコンポーネントファイルが存在する場合、通常ではデフォルトチェックとして行われるカップルファイルのペアチェックおよび拡張ファイルの全コンポーネントファイルのチェックは発生しません。したがって、IGNORECOUPLE または IGNOREEXP を指定してチェックを中止する必要はありません。

### EXCLUDE：指定したファイルをバックアップから除外

EXCLUDE は、BACKOUT 処理から除外するファイル（バックアウトされないファイル）をリストします。これらのファイルに関するプロテクションレコードはすべて無視されます。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

FILE パラメータが指定されている場合、EXCLUDE パラメータに指定されたすべてのファイルが FILE パラメータにも指定されている必要があります。

EXCLUDE パラメータは、BACKOUT 実行時にトランザクションロジックが使用されるかどうかには関係ありません。

EXCLUDE パラメータは、Adabas Recovery Aid（ADARAI）によって構築されたりカバリジョブで使用するためのパラメータです。

除外されたファイルは ADAREP レポートの拡張 CPLIST にリストされます。

### FILE：対象ファイル

BACKOUT 処理ですべてのファイルを対象とする場合は、このパラメータを指定しないでください。

指定したファイルが Adabas 拡張ファイルのコンポーネントファイルである場合は、その拡張ファイルの他のすべてのコンポーネントファイルも指定する必要があります。指定したファイルが他のファイルとカップリングされている場合は、そのファイルも指定しなければなりません。

### FROMBLK：BACKOUT の開始ブロック

FROMBLK は、FROMMCP チェックポイントエントリが含まれるブロック番号を指定します。このブロック番号は、データベースステータスレポートから取得でき、PLOGNUM または FROMPLOG のいずれかを参照します。FROMBLK は、FROMMCP が指定された場合にのみ指定できます。

### FROMMCP：BACKOUT の開始チェックポイント

FROMMCP は、バックアウト処理の開始対象チェックポイントを指定します。チェックポイント ID（名前）は、データベースステータスレポートから取得でき、PLOGNUM または FROMPLOG のいずれかを参照します。

バックアウト処理をログの最後から開始する場合、このパラメータを省略しなければなりません。

### IGNORECOUPLE：指定されていないカップリングファイルは無視

IGNORECOUPLE（または CONTINUE）を指定すると、FILE リストの中で、カップルファイルのペアがすべて揃っているかどうかのチェックが BACKOUT 機能で行われなくなりま

す。CONTINUE パラメータおよび IGNORECOUPLE パラメータの双方が指定されず、またその FILE リスト内でカップルファイルの一方が指定されていない場合、ADARES はエラーメッセージを発行して終了します。

**IGNOREEXP**：拡張構成要素ファイルリストのチェックの中止

FILE リストに Adabas 拡張コンポーネントファイルが含まれている場合、ADARESBACKOUT では、通常、リストのコンポーネントファイルに関連するすべての追加コンポーネントファイルもリストに存在していることをチェックします。存在していない場合、ADARES は BACKOUT 処理を終了してエラーメッセージを発行します。IGNOREEXP（または CONTINUE）を指定すると、関連するコンポーネントファイルに対するチェックが行われなくなります。

**MTR**：マルチスレッド再生成の有効／無効の切り替え

MTR が YES の場合はマルチスレッド再生成機能をアクティブにし、NO の場合は使用不可にします。

マルチスレッド再生成機能がアクティブであるときは、性能を向上するために、PLOG 情報を含む複数のバッファが同時に Adabas ニュークリアスに送られます。マルチスレッド再生成機能が使用されないときは、一度に Adabas に送られるバッファは 1 つだけです。

ニュークリアス ADARUN パラメータ MODE=SINGLE が指定されると、MTR は自動的に NO に設定されます。Adabas がシングルユーザーモードで実行しているときはマルチスレッドは使用できません。

FILE パラメータを指定しないか、あるいは CONTINUE パラメータ付きで指定すると、MTR のデフォルト値は YES になります。この場合、マルチスレッド再生成はデータベース全体を排他制御し、通常は有効になります。

上記以外のデフォルト値は NO です。CONTINUE なしで FILE が指定されたときなど、マルチスレッド再生成が一部のファイルだけを排他制御する場合は、異なるファイルにアクセスする通常のアプリケーションを同時に実行できますが、運用中のアプリケーションの性能に悪い影響を与える可能性があります。

**NOAUTOBACKOUT**：不完全トランザクションのバックアウトの禁止

1 つの Adabas セッション中にマルチプロテクションログが作成されており、連続していくつかの BACKOUT 機能を実行しなければならない場合、自動バックアウト処理は最後の入力ログに対してのみ実行すべきです。したがって、最後の入力ログを使用する BACKOUT 実行を除くすべての BACKOUT 実行に対して NOAUTOBACKOUT パラメータを指定しなければなりません。

 **Note:** NOAUTOBACKOUT パラメータはシングルユーザーモードでは指定できません。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

## BACKOUT：シーケンシャルプロテクションログ（SIBA）を使用した更新のバックアウト

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベン  
ドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エ  
ラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了しま  
す。

### NPCALLS：同時コール最大数

MTR=YES のとき、Adabas ニュークリアスに送られる同時コール数を制限するために  
NPCALLS パラメータを指定できます。

NPCALLSのデフォルト値は、FILEパラメータが指定されていないか、あるいはCOUTINUE  
付きで指定されている場合は、ニュークリアス ADARUN パラメータ NT に 1 を加えた値か  
NC のどちらか小さい方になります。

FILEパラメータがCONTINUEなしで指定されている場合は、デフォルト値はニュークリア  
ス ADARUN パラメータ NT に 1 を加えた値か、NC を 2 で割った値のどちらか小さい方  
になります。

NPCALLS は、主として、デフォルト値で許可される同時コール数を減らすために使いま  
す。同時コール数が少ないと、ADARESがニュークリアスにかかる負荷も少なくなります。  
このことは、BACKOUT FILE と同時に実行されるアプリケーションプログラムが利用できる  
リソースを増やす場合は、特に役立ちます。

### PARALLELREAD：他のユーザーによる読み取り専用ファイル利用の有効化

PARALLELREADパラメータを使用すると、データベース全体およびファイル指向の両方の  
機能で、ADARES BACKOUT で処理されているファイルに同時読み取り専用アクセスがで  
きるようになります。

- ファイル指向機能では、PARALLELREAD を指定することにより、ADARES はレコード  
バッファに "EXU=file-list" を含む OPEN コールを発行します。これは、ADARES がアク  
ティブな間、他のユーザーのためにファイルへの読み取り専用アクセスを可能にします。
- FILE が指定されていないとき、または CONTINUE が指定されているときは、  
PARALLELREAD パラメータはデータベース全体でのセッションバックアウトに有効で  
す。このパラメータによって、データベースセッションがバックアウトされているときで  
も読み取り専用ユーザーがデータベースにアクセスできるようになります。

更新コマンドは拒否されます。

バックアウトが行われているデータベースセッションで更新されたレコードを、同時アクセ  
スしているユーザーが読み取った場合は、アプリケーションの観点から論理的に間違っ  
たレコードイメージ、または矛盾を示すレスポンスコード（113 など）が表示されることが  
あります。



**Note:** PARALLELREAD での ADARES オペレーション中に、アソシエータとデータ  
ストレージが位置的に異なると、ニュークリアスレスポンスが 113 または 199 になる  
可能性があります。

**PLOGDBID：代替プロテクションログ ID**

ADARUN ステートメントの DBID パラメータで指定されたデータベース以外のデータベースからのプロテクションログを使用してバックアウト処理を行う場合、PLOGDBID は、代替プロテクションログのデータベース ID を指定します。デフォルトでは、ADARUN 指定のデータベースのデータベース ID です。

**TEST：構文テスト**

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**TOBLK：TOCP ブロックの終了**

TOBLK は、TOCP チェックポイントエントリをもつブロック番号を指定します。このブロック番号は、データベースステータスレポートから取得でき、PLOGNUM または FROMPLOG のいずれか、または TOPLOG（指定されている場合）を参照します。TOBLK は、TOCP が指定された場合のみ指定できます。

**TOCP：バックアウトの終了チェックポイントブロック**

TOCP は、バックアウト処理を終了させるチェックポイントを指定します。ログの先頭までバックアウト処理を続ける場合には、TOCP を指定しないでください。チェックポイント ID（名前）は、データベースステータスレポートから取得でき、TOPLOG（指定されている場合）あるいは PLOGNUM または FROMPLOG のいずれかを参照します。

**TOPLOG：バックアウトに対する終了 PLOG セッション**

TOPLOG は、指定された ADARES 機能によって処理される最終セッションを指定します。FROMPLOG/TOPLOG で定義された対象範囲外のセッション番号を持つ PLOG 入力（DD/SIIN）ファイル上のセッションが ADARES で見つかった場合、このセッションは ADARES 処理から除外されます。TOPLOG は、FROMPLOG が同時に指定されている場合にのみ指定できます。また TOPLOG が指定されない場合には、FROMPLOG セッションがデフォルトとなります。終了点をさらに明確に定義するには、TOCP パラメータおよび TOBLK パラメータを指定します。

## 例

### 例 1

```
ADARES BACKOUT PLOGNUM=3
```

全ファイルをバックアウト処理の対象とします。プロテクションログ番号は 3 です。バックアウト処理はログの末尾から開始し、ログの先頭で終了します。バックアウト処理の終了時には、不完全なトランザクションの自動バックアウト（ただし方向は順方向）が実行されます。

### 例 2

## BACKOUT：シーケンシャルプロテクションログ（SIBA）を使用した更新のバックアウト

---

```
ADARES BACKOUT
FILE=4,7,PLOGNUM=11,FROMCP=CH18,FROMBLK=1864,
ADARES TOCP=CH01,TOBLK=1
```

バックアウトはファイル 4 と 7 に限定されます。チェックポイント CH01 と CH18 の間にファイル 4 と 7 に適用されたすべての更新が取り消されます。CH01 はデータプロテクションログ 11 のブロック 1 にあります。CH18 はデータプロテクションログ 11 のブロック 1864 にあります。不完全なトランザクションの自動バックアウトは実行されません。

# 158 BACKOUT DPLOG/MPLOG：デュアル／マルチプ

## ロテクションログを使用した更新のバックアウト

---

▪ 機能の実行 .....	876
▪ 構文 .....	877
▪ 基本パラメータ .....	878
▪ オプションパラメータ .....	878
▪ 例 .....	882

## BACKOUT DPLOG/MPLOG：デュアル／マルチプロテクションログを使用した更新のバックアウト

---

BACKOUT {DPLOG | MPLOG} 機能では、それぞれ同じ Adabas デュアルプロテクションログデータセットまたはマルチプロテクションログデータセットにある2つのチェックポイント間に適用された更新をすべて取り消します。

BACKOUT {DPLOG | MPLOG} 機能は、クラスタデータベースには有効ではありません。マージされた PLOG が BACKOUT を実行するために必要とされるので、これは禁止されます。

このchapterでは、次のトピックについて説明します。

## 機能の実行

---

BACKOUT DPLOG/MPLOG 機能は、次の手順で実行することをお勧めします。

### 1. オペレータコマンドまたは Online System コマンド FEOFPL の発行

現在のプロテクションログデータセットで強制的に EOF を実行し、新しいデータセットに切り換えます。新しいプロテクションログデータセットには、BACKOUT DPLOG/MPLOG に必要なすべての情報が含まれています。

### 2. ユーザーアプリケーションの実行

このアプリケーションのニュークリアスによって書き込まれたすべてのプロテクションデータは、1つのプロテクションログデータセットに収まる必要があります。アプリケーションの実行中は、プロテクションログの切り換えは発生しません。この時点でアプリケーションプログラムが失敗し、バックアウトが必要になったと想定します。

### 3. オペレータコマンドまたは Online System コマンド FEOFPL の再発行

プロテクションログデータセットをクローズします。クローズしたデータセットには、BACKOUT DPLOG/MPLOG に必要なすべての情報が含まれています。

### 4. ADARES PLCOPY の実行

プロテクションログデータセットの内容をシーケンシャルデータセットにコピーします。この処理は、ADARES PLCOPY を実行するか、DPLOG の場合はユーザー出口 2 を、MPLOG の場合はユーザー出口 12 を使用することによって実行できます。

### 5. ADARES BACKOUT DPLOG/MPLOG の実行

これによってセッションは手順 1 の状態にバックアウトされます。

中断した BACKOUT DPLOG/MPLOG は、最初から再度実行しなければなりません。プロテクションログデータセットの使用対象データが使用できない場合（このデータプロテクションログがニュークリアスによって再び使用される場合）は、シーケンシャルコピーからの BACKOUT を実行する必要があります。

## 構文

---

通常は、パラメータFROMMCP/TOCP/FROMBLK/TOBLKを指定しないでください。バックアウトには、1つのプロテクションログデータセットの内容全体を使用することをお勧めします。

ニュークリアスは、バックアウト処理中に、現在使用可能なプロテクションログデータセットに新しいプロテクションログ情報を書き込みます。これはニュークリアスが使用できる唯一のデータセットです。BACKOUT DPLOG/MPLOGの途中でプロテクションログの切り換えが発生した場合、ニュークリアスはADARESPLCOPYによってデータセット全体がコピーされるまで待機し、その後バックアウト処理を再開します。

デフォルトでは、指定した入力データセットのすべてのファイルがBACKOUT処理の対象になります。特定のファイルを処理の対象に指定することができます。

CONTINUEは、ファイルのバックアウト中に不完全なトランザクションによる変更が行われた場合に、その自動バックアウトを許可します。CONTINUEを指定すれば、FILEパラメータによって指定されたデータセットの不完全トランザクションによるすべての変更がバックアウトされます。ファイルリスト中にカップリングファイルまたは拡張ファイルのコンポーネントファイルが存在する場合、通常ではデフォルトチェックとして行われるカップルファイルのペアチェックおよび拡張ファイルの全コンポーネントファイルのチェックは発生しません。したがって、IGNORECOUPLEまたはIGNOREEXPを指定してチェックを中止する必要はありません。

CONTINUEを指定すると、ファイルのバックアウト中はデータベース全体がBACKOUT機能による排他使用のためにロックされます。

BACKOUT処理終了時、データベース全体にBACKOUTが指定されると、ADARESは、すべての不完全な論理トランザクションを自動的にバックアウトし、入力データセットが終了するまで継続します。このことは、次に該当する場合にも発生します。

- FILEパラメータおよびCONTINUEパラメータの2つが指定された場合
- TOCPパラメータが指定されなかった場合

NOAUTOBACKOUTパラメータを指定してこの処理を無効にすることができます。

```
ADARES BACKOUT { DPLOG [DUALPLD = {device-type | ADARUN-device}]
                  MPLOG [PLOGDEV = {device-type | ADARUN-device}] }
[EXCLUDE = file-list]
[FILE = file-list [CONTINUE]]
[FROMCP = checkpoint-name [, FROMBLK = checkpoint-block]]
[IGNORECOUPLE]
[IGNOREEXP]
[MTR = { YES | NO } [NPCALLS = maximum-number-of-parallel-calls]]
[NOAUTOBACKOUT]
[NOUSERABEND]
[PARALLELREAD]
[PLOGDBID = {alternate-log-dbid | ADARUN-dbid}]
[TEST]
[TOCP = checkpoint-name [, TOBLK = checkpoint-block]]
```

## 基本パラメータ

---

### DPLOG | MPLOG：デュアル／マルチ PLOG ソース

DPLOG は、デュアルプロテクションログデータセットを入力として使用することを示します。MPLOGは、マルチプロテクションログデータセットを入力として使用することを示します。

## オプションパラメータ

---

### CONTINUE：オートバックアウトオプション付きのファイルリカバリ

CONTINUE は、BACKOUT 機能による排他使用のためにデータベース全体をロックします。

CONTINUEは、不完全なトランザクションによってファイルの内容が変更されていた場合、それらをファイルバックアウトの際に自動バックアウトします。CONTINUEを指定すれば、FILEパラメータによって指定されたデータセットの不完全トランザクションによるすべての変更がバックアウトされます。

ファイルリスト中にカップリングファイルまたは拡張ファイルのコンポーネントファイルが存在する場合、通常ではデフォルトチェックとして行われるカップルファイルのペアチェックおよび拡張ファイルの全コンポーネントファイルのチェックは発生しません。したがって、IGNORECOUPLE または IGNOREEXP を指定してチェックを中止する必要はありません。

#### DUALPLD | PLOGDEV：PLOG デバイスタイプ

DUALPLDはデュアルプロテクションログデータセットに使用するデバイスタイプを指定します。PLOGDEVはマルチプロテクションログデータセットに使用するデバイスタイプを指定します。デフォルトは、ADARUN DEVICE パラメータで指定されているデバイスタイプです。

#### EXCLUDE：指定したファイルをバックアップから除外

EXCLUDEは、BACKOUT 処理から除外するファイル（バックアウトされないファイル）をリストします。これらのファイルに関するプロテクションレコードはすべて無視されます。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

FILEパラメータが指定されている場合、EXCLUDEパラメータに指定されたすべてのファイルがFILEパラメータにも指定されている必要があります。

EXCLUDEパラメータは、BACKOUT 実行時にトランザクションロジックが使用されるかどうかには関係ありません。

EXCLUDEパラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

除外されたファイルは ADAREP レポートの拡張 CPLIST にリストされます。

#### FILE：対象ファイル

FILEはバックアウト処理の対象とするファイルを指定します。すべてのファイルを対象とする場合は、このパラメータを省略しなければなりません。指定ファイルがAdabas 拡張ファイルのコンポーネントファイルである場合、拡張ファイルの全コンポーネントファイルも指定しなければなりません。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。



**Note:** 開始前に、ADARES は 指定されたすべてのファイルを BACKOUT の実行期間中ロックします。したがって、FILE パラメータの指定がなければ、データベース全体をロックします。他のユーザーは、UTYPE=EXUパラメータが指定されている場合は指定されたファイルに、PARALLELREADパラメータが指定されている場合はデータベースに対して、読み取り専用アクセスが可能です。

#### FROMMCP：バックアウトの開始チェックポイント

FROMMCPは、バックアウト処理の開始対象チェックポイントを指定します。チェックポイント ID は、データベースステータスレポートから取得できます。バックアウト処理をログの最後から開始する場合は、FROMMCP パラメータを指定しないでください。

#### FROMBLK：開始 FROMMCP ブロック番号

FROMBLKはFROMMCP チェックポイントエントリが含まれるブロック番号です。このブロック番号は、データベースステータスレポートから取得できます。FORMBLKは、FROMMCP が指定された場合のみ指定できます。

**IGNORECOUPLE**：指定のないカップルファイルの無視

IGNORECOUPLE（またはCONTINUE）を指定すると、FILEリストの中で、カップルファイルのペアがすべて揃っているかどうかのチェックがBACKOUT機能で行われなくなります。CONTINUEパラメータおよびIGNORECOUPLEパラメータの双方が指定されず、またそのFILEリスト内でカップルファイルの一方が指定されていない場合、ADARESはエラーメッセージを発行して終了します。

**IGNOREEXP**：拡張構成要素ファイルリストのチェックの中止

FILEリストにAdabas拡張コンポーネントファイルが含まれている場合、ADARESBACKOUTでは通常、リストのコンポーネントファイルに関連するすべての追加コンポーネントファイルもリストに存在していることをチェックします。存在していない場合、ADARESはBACKOUT処理を終了してエラーメッセージを発行します。IGNOREEXP（またはCONTINUE）を指定すると、関連するコンポーネントファイルに対するチェックが行われなくなります。

**MTR**：マルチスレッドバックアウトの切り換え

MTR=YESを指定するとマルチスレッドバックアウト機能がアクティブになり、MTR=NOを指定すると使用不可になります。

マルチスレッドバックアウト機能をアクティブにすると、性能を向上させるために、PLOG情報を含む複数のバッファが同時にAdabasニュークリアスに送られます。マルチスレッド再生機能を使用されないときは、一度にAdabasに送られるバッファは1つだけです。

ニュークリアスADARUNパラメータMODE=SINGLEが指定されると、MTRは自動的にNOに設定されます。Adabasがシングルユーザーモードで実行しているときはマルチスレッドは使用できません。

FILEパラメータを指定しないか、あるいはCONTINUEパラメータ付きで指定すると、MTRのデフォルト値はYESになります。このような場合、マルチスレッドバックアウトはデータベース全体を排他制御するため、通常は効果があります。

上記以外のデフォルト値はNOです。CONTINUEなしでFILEが指定されたときなど、マルチスレッドバックアウトが一部のファイルのみを排他制御する場合は、異なるファイルにアクセスする通常のアプリケーションを同時に実行できますが、運用中のアプリケーションの性能に悪い影響を与える可能性があります。

**NOAUTOBACKOUT**：不完全トランザクションのバックアウトの禁止

1つのAdabasセッション中にマルチプロテクションログが作成されており、連続していくつかのBACKOUT機能を実行しなければならない場合、自動バックアウト処理は最後の入力ログに対してのみ実行すべきです。したがって、最後の入力ログを使用するBACKOUT実行を除くすべてのBACKOUT実行に対してNOAUTOBACKOUTパラメータを指定しなければなりません。



**Note:** NOAUTOBACKOUTパラメータはシングルユーザーモードでは指定できません。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**NPCALLS**：同時コール最大数

MTR=YES のとき、Adabas ニュークリアスに送られる同時コール数を制限するために NPCALLS パラメータを指定できます。

NPCALLSのデフォルト値は、FILEパラメータが指定されていないか、あるいはCONTINUE付きで指定されている場合は、ニュークリアス ADARUN パラメータ NT に 1 を加えた値か NC のどちらか小さい方になります。

FILEパラメータがCONTINUEなしで指定されている場合は、デフォルト値はニュークリアス ADARUN パラメータ NT に 1 を加えた値か、NC を 2 で割った値のどちらか小さい方になります。

NPCALLSは、主として、デフォルト値で許可される同時コール数を減らすために使います。同時コール数が少ないと、ADARESがニュークリアスにかかる負荷も少なくなります。このことは、BACKOUT DPLOG FILE と同時に実行されるアプリケーションプログラムが利用できるリソースを増やす場合に特に役立ちます。

**PARALLELREAD**：他のユーザーによる読み取り専用ファイル利用の有効化

PARALLELREADパラメータは、データベース全体およびファイル指向の両方の機能に対して、ADARES BACKOUT DPLOG で処理されているファイルに同時読み取り専用アクセスを提供します。

- ファイル指向機能では、PARALLELREADを指定することにより、ADARESはレコードバッファに"EXU=file-list"を含むOPENコールを発行します。これは、ADARESがアクティブな間、他のユーザーのためにファイルへの読み取り専用アクセスを可能にします。
- FILEが指定されていないとき、またはCONTINUEが指定されているときは、PARALLELREADパラメータはデータベース全体でのDPLOGバックアウトに有効です。このパラメータによって、データベースDPLOGがバックアウトされているときでも読み取り専用ユーザーがデータベースにアクセスできるようになります。

更新コマンドは拒否されます。

バックアウトが行われているデータベースDPLOGで更新されたレコードを、同時アクセスしているユーザーが読み取った場合は、アプリケーションの観点から論理的に間違ったレコードイメージ、または矛盾を示すレスポンスコード（113など）が表示されることがあります。

## BACKOUT DPLOG/MPLOG：デュアル／マルチプロテクションログを使用した更新のバックアウト

---



**Note:** PARALLELREAD での ADARES オペレーション中に、アソシエータとデータストレージが位置的に異なると、ニュークリアスレスポンスが 113 または 199 になる可能性があります。

### PLOGDBID：代替プロテクションログ ID

ADARUN ステートメントの DBID パラメータで指定されたデータベース以外のデータベースからのプロテクションログを使用してバックアウト処理を行う場合、PLOGDBID は、代替プロテクションログのデータベース ID を指定します。デフォルトでは、ADARUN 指定のデータベースのデータベース ID です。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### TOBLK：終了 TOCP ブロック番号

TOBLK は、TOCP チェックポイントエントリをもつブロック番号を指定します。TOBLK は、TOCP が指定された場合のみ指定できます。

### TOCP：バックアウトの終了チェックポイント

TOCP は、バックアウト処理を終了するチェックポイントを指定します。バックアウト処理は、指定されたチェックポイントまで継続します。バックアウト処理をログの先頭まで継続する場合は、TOCP や NOAUTOBACKOUT を指定しないでください。

## 例

---

1. ADADBS OPERCOM FEOFPL
2. ファイル 20 および 21 でユーザーアプリケーションが失敗
3. ADADBS OPERCOM FEOFPL
4. ADARES BACKOUT DPLOG,FILE=20,21

この例は、PLCOPY 機能がユーザー出口 2 で実行されることを想定しています。プロテクションログの切り換えが発生するたびに、デュアルプロテクションログの内容をシーケンシャルデータセットにコピーするためのジョブがこのユーザー出口からサブミットされます。

1. 新しい PLOG に切り換えます。
2. 新しい PLOG データセット上に PLOG データを作成するユーザーセッションを実行します。
3. PLOG データセットをクローズします。ユーザー出口 2 が、クローズしたばかりの PLOG データセットの内容をコピーするジョブをサブミットします。
4. ファイル 20 および 21 について、この PLOG からの BACKOUT を PLOG の先頭まで実行します。

# 159 CLCOPY : デュアルコマンドログのコピー

---

■ オプションパラメータ .....	884
■ 例 .....	885

CLCOPY 機能は、Adabasセッションにコマンド情報のデュアルログが指定された場合に限り使用します。この機能は、タイムスタンプの早い方のデータセットをシーケンシャルデータセットにコピーします。いったんCLCOPY機能が正常に完了すると、コピーされたデータセットは空としてマークされます。したがって、この機能は1つのデータセットに対して1回だけ使用できます。

CLCOPY 機能はシングルユーザーモードでは使用できません。

```
ADARES CLCOPY [DUALCLD = {device-type | ADARUN-device}]  
[NOUSERABEND]  
[OPENOUT]  
[TEST]  
[TWOOCOPIES]
```

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

---

ADARES CLCOPY はパラメータなしで指定できます。

### DUALCLD：デュアルコマンドログのデバイスタイプ

DUALCLDは、デュアルコマンドログデータセットに使用されるデバイスタイプを指定します。このパラメータは、コマンドログデータセットに使用されるデバイスタイプが、ADARUN DEVICE パラメータで指定されているものと異なる場合に必要です。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### OPENOUT：DDSIAUS1/2 または SIAUS1/2 出力データセットのオープン

OPENOUT パラメータは、実際にデータがコピーされなくても ADARES がオープンする DD/SIAUS1/2 出力データセットを指定します。OPENOUT を使用しない場合、ADARES が最初の入力レコードを読み込もうとしたときにエンドオブファイル条件を検出するとシーケンシャル出力データセットはオープンされません。これにより一部のオペレーティングシステム環境では問題が発生することがあります。OPENOUT を使用すると、出力データセットは最初の入力レコードを読み込む前にオープンされます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**TWOCOPIES**：2つの出力コピーの作成

TWOCOPIES は、同じ出力を2つ作成します。

## 例

### 例 1

```
ADARES CLCOPY
```

デュアルコマンドログをコピーします。

### 例 2

```
ADARES CLCOPY TWOCOPIES
```

デュアルコマンドログをコピーします。コピーの出力は2つ作成されます。



# 160 COPY : シーケンシャルプロテクションログまたはセーブテープのコピー

---

▪ オプションパラメータ .....	889
▪ 例 .....	890

COPY機能は、Adabasシーケンシャルプロテクションログデータセットをコピーします。シーケンシャルプロテクションログデータセットを作成したAdabasセッションが異常終了した場合は、まずCOPY機能を実行しなければ、このデータセットを他のADARES機能の入力として使用することはできません。

### ADARES COPY について

- データプロテクションログデータセットをADARESBACKOUT機能の入力として使用できるようにするには、事前にこの機能を使用してデータプロテクションログデータセットをディスクからテープデータセットにコピーする必要があります。
- この機能は、後続のAdabasセッションで他のデータプロテクションログデータセットが作成されている場合でも使用できます。
- また、ADASAV SAVE 出力 (DD/SAVE<sub>n</sub>) を入力とすることもできます。1回のADARES COPYの実行では、1つのADASAV SAVE 入力ボリュームのみがコピー可能です。SAVE出力テープはDD/SIIN ジョブ制御ファイルに割り当てる必要があります。
- この機能は、任意の入力データセットに対して何回でも実行できます。

#### ADARES COPY

```
[ PLOGNUM = protection-log-number  
  FROMPLOG = start-session [, TOPLOG = stop-session ]  
[NONUC]  
[NOUSERABEND]  
[OPENOUT]  
[RLOGDEV = { device-type | ADARUN-device }]  
[TEST]  
[TWO_COPIES]  
[UTICPLIST]
```

Adabas Delta Save Facility を使用している場合、COPY 機能には特別な用途があります。詳細は『*Adabas Delta Save Facility* マニュアル』を参照してください。

ADARES COPY はパラメータなしで指定できます。PLOGNUM と FROMPLOG のどちらも指定しないで ADARES COPY を指定すると、入力プロテクションログ全体がコピーされます。

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

### FROMPLOG：COPYの開始セッション

FROMPLOGは、指定されたADARES機能の開始時のセッション番号を指定します。ADARESはPLOG入力（DD/SIIN）ファイル内で正しい開始セッションを検索します。

### NONUC：ニュークリアスレスポンスの無視

Adabasニュークリアスが自動再起動を実行できない場合は、NONUCパラメータを使用してニュークリアスレスポンスコードを無視できます。この場合、ADARESでは、できる限り、入力テープを物理的にコピーしようとします。

このパラメータは、他のすべての対処が失敗した場合を除き、指定しないでください。ただし、コピーする複数のニュークリアスセッションがPLOGシーケンシャル入力（DD/SIIN）ファイルに含まれている場合は、NONUCを指定する必要があります。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### OPENOUT：DDSIAUS1/2 または SIAUS1/2 出力データセットのオープン

OPENOUTパラメータは、実際にデータがコピーされなくてもADARESがオープンするDD/SIAUS1/2出力データセットを指定します。OPENOUTを使用しない場合、ADARESが最初の入力レコードを読み込もうとしたときにエンドオブファイル条件を検出するとシーケンシャル出力データセットはオープンされません。これにより一部のオペレーティングシステム環境では問題が発生することがあります。OPENOUTを使用すると、出力データセットは最初の入力レコードを読み込む前にオープンされます。

### PLOGNUM：プロテクションログ番号

コピーするデータセットのAdabasプロテクションログ番号を指定します。この番号は、ADAREPユーティリティによって作成されたデータベースステータスレポートから取得できます。COPY機能の出力にも同じログ番号が割り当てられます。

### RLOGDEV：RLOGデータセットのデバイスタイプ

RLOGDEVパラメータは、Adabas Recovery Aid (ADARAI) がアクティブな場合に、リカバリログ (RLOG) データセットのデバイスタイプを指定するために使用します。

RLOGDEVを指定しなかった場合（デフォルト）、リカバリログのデバイスタイプはADARUN DEVICEパラメータと同じであると想定されます。

RLOGDEVに指定された値またはデフォルト値が正しくない場合、ADARESCOPYはERROR 149 "missing or mismatching RLOGDEV parameter" を表示して終了します。

RLOGDEV パラメータを使用すると、データベースの GCB が破壊されても ADARES は ADARAI 用の機能を記録することができます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**TOPLOG**：バックアウトに対する終了 **PLOG** セッション

TOPLOG は、指定された ADARES 機能によって処理される最終セッションを指定します。指定された TOPLOG よりも番号の大きいセッションが、ADARES によって PLOG 入力 (DD/SIIN) ファイルで検出された場合、そのセッションは ADARES 処理の対象から除外されます。

**TWOCOPIES**：出力コピーを 2 つ作成

TWOCOPIES は、同じ出力を 2 つ作成します。TWOCOPIES を指定しなければ、デフォルトによって 1 つの出力が作成されます。

**UTICPLIST**：すべてのユーティリティチェックポイントの出力

UTICPLIST パラメータを指定すると、COPY 機能の実行中にデータプロテクションログ上で検出された SYNPN、SYNV、および SYNS のチェックポイントすべてが、ADARES によって選択されて出力されます。

## 例

---

### 例 1

```
ADARES COPY PLOGNUM=6
```

データプロテクションログ 6 がコピーされます。

### 例 2

```
ADARES COPY PLOGNUM=8 ,TWOCOPIES
```

データプロテクションログ 8 がコピーされます。コピーの出力は 2 つ作成されます。

# 161 MERGE CLOG : ニュークリアスクラスタコマン ドログのマージ

---

- 基本パラメータ ..... 892

Adabas クラスタ環境では、次の2つの方法のいずれかを使用してクラスタ内のコマンドログ（CLOG）をマージできます。

- システムが適切に設定されている場合（CLOGMRG=YES およびユーザー出口2）、CLOGは自動的にマージされます。
- それ以外の場合は、ADARES MERGE CLOG ユーティリティ機能を使用してCLOGを手動でマージできます。

**ADARES MERGE CLOG, NUMLOG = nn**

MERGE CLOG 機能への入力としてシーケンシャルデータセットが想定されています。そのため、ADARES CLCOPY 機能は ADARES MERGE 機能よりも先に実行する必要があります。

CLOGLAYOUT=5 に含まれているタイムスタンプは、コマンドログレコードを適切にマージするために必要です。

## 基本パラメータ

---

**NUMLOG**：コマンドログデータセットの数

NUMLOG パラメータは必須です。マージプロセスの対象とするコマンドログデータセットの数を指定します。最大数は32です。

# 162      PLCOPY：プロテクションログのシーケンシャルデータセットへのコピー

---

- オプションパラメータ ..... 894
- 例 ..... 896

PLCOPY 機能は、プロテクション情報のデュアル／マルチログが Adabas セッションに指定された場合にのみ使用します。この機能は、タイムスタンプの早い方のデータセットをシーケンシャルデータセットにコピーします。いったん PLCOPY 機能が正常に完了すると、コピーされたデータセットは空としてマークされます。したがって、この機能は 1 回の Adabas セッションの 1 つのデータセットに対して 1 回のみ使用できます。

プロテクションログファイルに対するハードウェア圧縮 (IDRC) の使用はお勧めしません。ADARES BACKOUT 機能では、VSE システムおよび z/VM システムでのハードウェア圧縮データはサポートされません。z/OS システムでは、BACKOUT 機能で圧縮データを処理すると 2 倍以上の実行時間がかかることがあります。

PLCOPY 機能は、シングルユーザーモードでは使用できません。

```
ADARES PLCOPY [PLOGDEV = { device-type | ADARUN-device }]  
[NOPPT]  
[NOUSERABEND]  
[OPENOUT]  
[RLOGDEV = { device-type | ADARUN-device }]  
[SBLKNUM = starting-block-num ]  
[TEST]  
[TWOCOPIES]  
[UTICPLIST]
```

Adabas Delta Save Facility を使用している場合、PLCOPY 機能には特殊な用途があります。詳細は『*Adabas Delta Save Facility* マニュアル』を参照してください。

ADARES PLCOPY はパラメータなしで指定できます。

このchapterでは、次のトピックについて説明します。

## オプションパラメータ

---

### PLOGDEV：PLOG デバイスタイプ

PLOGDEV は、デュアル／マルチプロテクションログデータセットに使用するデバイスタイプを指定します。このパラメータは、デュアル／マルチプロテクションログデータセットに使用されるデバイスタイプが、ADARUN DEVICE パラメータで指定されているものと異なる場合に必要です。

**NOPPT**（クラスタニュークリアス環境のみ）

Parallel Participant Table (PPT) は、コピーするデータセットを ADARES PLCOPY に通知します。PPT が破壊された場合、DBA は、コピーおよびマージされる PLOG データセットを ADARES NOPPT 機能を使用して指定できます。

ADARAI を使用している場合は、ニュークリアスの初期化時に PLOG データセットが RLOG に書き込まれます。エラーが発生し、最終的な PLCOPY が必要な場合は、RLOG に書き込まれた PLOG データセットに基づいて ADARAI が PLCOPY NOPPT JCL を構築できます。

NOPPT は、PPT が上書きされた緊急時にのみ使用してください。PPT を無視し、DD/PLOG データセットに JCL が供給されるように指定します。

 **Caution:** このパラメータを使用すると、PPT および一般に PPT によって提供されるすべての制御タイプ情報が無視されます。このため、このパラメータは慎重に使用する必要があります。

このパラメータを使用するときは以下を供給しなければなりません。

- 正しい中間データセット
- DD/PLOG01～nn の形式の全ニュークリアスからの正しい入力プロテクションログ。

 **Caution:** PPT なしでは、ADARES は入力データセット上で広範囲な検証を実行することができません。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**OPENOUT**：DDSIAUS1/2 または SIAUS1/2 出力データセットのオープン

OPENOUT パラメータは、実際にデータがコピーされなくても ADARES がオープンする DD/SIAUS1/2 出力データセットを指定します。OPENOUT を使用しない場合、ADARES が最初の入力レコードを読み込もうとしたときにエンドオブファイル条件を検出するとシーケンシャル出力データセットはオープンされません。これにより一部のオペレーティングシステム環境では問題が発生することがあります。OPENOUT を使用すると、出力データセットは最初の入力レコードを読み込む前にオープンされます。

**RLOGDEV**：RLOG データセットのデバイスタイプ

RLOGDEV パラメータは、Adabas Recovery Aid (ADARAI) がアクティブな場合に、リカバリログ (RLOG) データセットのデバイスタイプを指定するために使用します。

RLOGDEV を指定しなかった場合（デフォルト）、リカバリログのデバイスタイプは ADARUN DEVICE パラメータと同じであると想定されます。

RLOGDEV に指定された値またはデフォルト値が正しくない場合、ADARES PLCOPY は ERROR 149 "missing or mismatching RLOGDEV parameter" を表示して終了します。

RLOGDEV パラメータを使用すると、データベースの GCB が破壊されても ADARES は ADARAI 用の機能を記録することができます。

### SBLKNUM

SBLKNUM パラメータは、NOPPT パラメータと組み合わせてのみ、かつ、PLCOPY 機能に対してのみ指定できます。

ユーザーは SBLKNUM を使用して、シーケンシャルマージ出力の開始ブロック番号を指定できます。このパラメータを省略した場合は、PPT を読み取り、そこからブロック番号を取得する試みが行われます。この読み取りが失敗すると、出力はブロック 1 から開始されません。

このパラメータの値を決定するには、ユーザーは前回の PLCOPY の出力を確認して、連続する次のブロック番号を使用する必要があります。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### TWOCOPIES：2つの出力コピーの作成

TWOCOPIES は、同じ出力を 2 つ作成します。TWOCOPIES を指定しなければ、デフォルトによって 1 つの出力が作成されます。

### UTICPLIST：すべてのユーティリティチェックポイントの出力

UTICPLIST パラメータを指定すると、PLCOPY 機能の実行中にデータプロテクションログで検出された SYNPN、SYNV、および SYNS のチェックポイントすべてが ADARES によって選択され、出力されます。

## 例

---

### 例 1

デュアル/マルチプロテクションログをコピーします。

```
ADARES PLCOPY
```

### 例 2

デュアル/マルチプロテクションログのコピーを 2 つ作成します。

```
ADARES PLCOPY TWOCOPIES
```

### 例 3

デュアル／マルチプロテクションログをコピーします。Adabas Recovery Aid (ADARAI) はアクティブです。リカバリログ (RLOG) のデバイスタイプは 8390 です。

```
ADARES PLCOPY RLOGDEV=8390
```



# 163

## REGENERATE：更新の再生成

---

■ 構文 .....	900
■ 基本パラメータ .....	901
■ オプションのパラメータとサブパラメータ .....	901
■ 例 .....	906

REGENERATE 機能は、2つのチェックポイント間に実行されたすべての更新を再適用します。

通常の更新がリストアされるのに加えて、ADARES REGENERATE では選択したファイルに指定されたチェックポイント間で実行された次のADADBSユーティリティ（またはAdabas Online System）機能の更新もすべてリストアされます。

ALLOCATE	DELETE	NEWFIELD	RELEASE
CHANGE	DSREUSE	PRIORITY	RENAME
DEALLOCATE	ISNREUSE	RECOVER	RENUMBER
DELCP	MODFCB	REFRESH	UNCOUPLE

データベースについては、ファイルに関連する上記のすべての処理以外に、データベースに関連する次のADADBS（またはAdabas Online System）機能がすべて実行されます。

ADD	INCREASE（データセットのサイズ）
DECREASE（データセットのサイズ）	RECOVER

このchapterでは、次のトピックについて説明します。

## 構文

```

ADARES REGENERATE {
    PLOGNUM = protection-log-number
    FROMPLOG = start-session [, TOPLOG = stop-session ]
    [ALLOCATION = { FORCE | NOFORCE } ]
    [EXCLUDE = file-list ]
    [FILE = file-list [, CONTINUE] ]
    [FROMCP = { checkpoint-name | 'SYNS, INCLUDE' }
    [FROMBLK = checkpoint-block ] ]
    [IGNORECOUPLE]
    [IGNOREEXP]
    [MTR = { YES | NO } ] [NPCALLS = maximum-number-of-parallel-calls ] ]
    [NOAUTOBACKOUT]
    [NOUSERABEND]
    [PARALLELREAD]
    [PLOGDBID = { alternate-log-dbid | ADARUN-dbid } ]
    [RAID]
    [TEST]
    [TOCP = { checkpoint-name | 'SYNS, INCLUDE' }
    [AUTOBACKOUT] [TOBLK = checkpoint-block ] ]
}
    
```

## 基本パラメータ

プロテクションログのログ番号（PLOGNUM）またはセッション番号（FROMPLOG）のどちらかを、REGENERATE 処理の開始ポイントとして指定できます。セッション番号を指定する場合は、TOPLOG パラメータを使用すると、処理するセッションの範囲も指定できます。

### FROMPLOG：再生成の開始セッション

FROMPLOG は、指定された ADARES 機能の開始時のセッション番号を指定します。

ADARES は PLOG 入力ファイル内で正しい開始セッションを検索します。開始点をさらに明確に定義したい場合には、FROMMCP および FROMBLK パラメータを指定します。

### PLOGNUM：プロテクションログ番号

PLOGNUM は、再生成処理の入力として使用されるデータプロテクションログのログ番号です。ログ番号は、データベースステータスレポートから得ることができます。

## オプションのパラメータとサブパラメータ

### ALLOCATION：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

ALLOCATION は、再生成の一部として再生される次の処理に関係します。

- ADADBS ALLOCATE
- Adabas Online System の "ファイル定義"
- Adabas Online System の "DLOG AREA のインストール/変更"

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### AUTOBACKOUT：TOCP チェックポイントからのトランザクションのバックアウト

TOCP パラメータを指定すると、通常は処理の最後に不完全トランザクションのバックアウトは行われません。したがって、この TOCP チェックポイントに対応するユーティリティ機能を再実行し、開始チェックポイントを指定する FROMMCP を使用して別の ADARES 処理を続けることができます。

TOCP チェックポイントで REGENERATE/BACKOUT を終了する必要がある状況では、AUTOBACKOUT パラメータを使用して不完全トランザクションをバックアウトすることによって、データベースに論理的な不整合が発生しないようにすることができます。

AUTOBACKOUTは、TOCPが指定されている場合にのみ使用できることに注意してください。

**CONTINUE**：オートバックアウトオプション付きのファイルリカバリ

CONTINUEは、ファイルの再生成中に不完全なトランザクションによって行われた変更のAUTOBACKOUTを許可します。CONTINUEを指定すれば、FILEパラメータによって指定されたデータセットの不完全トランザクションによるすべての変更がバックアウトされます。

ファイルリストにカップリングされたコンポーネントファイルまたは拡張コンポーネントファイルが存在する場合にCONTINUEを指定すると、カップルファイルのペアまたはコンポーネントファイルセットがすべて揃っているかどうかに関する通常のリストのチェックは実行されません。したがって、IGNORECOUPLEまたはIGNOREEXPを指定してそれぞれのファイルリストのチェックを中止する必要はありません。

CONTINUEを指定すると、REGENERATE機能のみで使用するためにデータベース全体がロックされます。

**EXCLUDE**：指定されたファイルの再生成からの除外

EXCLUDEでは、REGENERATE処理から除外する、つまり再生成しないファイルの番号をリストします。これらのファイルに関するプロテクションレコードはすべて無視されます。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

FILEパラメータが指定されている場合、EXCLUDEパラメータに指定されたすべてのファイルがFILEパラメータにも指定されている必要があります。

EXCLUDEパラメータは、REGENERATE実行時にトランザクションロジックが使用されるかどうかには関係ありません。

EXCLUDEパラメータは、Adabas Recovery Aid (ADARAI)によって構築されたりカバリジョブで使用するためのパラメータです。

除外されたファイルはADAREPレポートの拡張CPLISTにリストされます。

**FILE**：対象ファイル

FILEは再生成処理の対象とするファイルを指定します。すべてのファイルを対象とする場合は、FILEパラメータを指定しないでください。指定ファイルがAdabas拡張ファイルのコンポーネントファイルである場合、拡張ファイルの全コンポーネントファイルも指定しなければなりません。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。



**Note:** 開始前に、ADARESは指定されたすべてのファイルをREGENERATEの実行期間中ロックします。したがって、FILEパラメータの指定がなければ、データベース全体をロックします。

**FROMBLK**：再生成の開始ブロック

FROMBLKは、FROMMCPチェックポイントエントリが含まれるブロック番号を指定します。このブロック番号は、前回の ADASAV リストア出力またはデータベースステータスレポートから取得できます。PLOGNUM または FROMPLOG を参照します。FORMBLK は、FROMMCP が指定された場合のみ指定できます。

**FROMMCP**：再生成の開始チェックポイント

FROMMCP は、REGENERATE 処理を開始するチェックポイントを定義します。処理は、指定されたチェックポイントの直後の情報から開始されます。このチェックポイント名は、前回の ADASAV リストアの出力 (SYN2/5)、データベースステータスレポート、あるいは UTICPLIST の指定による ADARES COPY/PLCOPY 出力から取得できます。処理をログの最初から開始する場合は、FROMMCP パラメータを指定しないでください。FROMMCP は PLOGNUM または FROMPLOG によって指定されるプロテクションログを参照します。

SYNS,INCLUDE オプションの詳細については、「INCLUDE (再生成へのチェックポイントのインクルード)」を参照してください。

**IGNORECOUPLE**：指定のないカップルファイルの無視

IGNORECOUPLE (または CONTINUE) を指定した場合、REGENERATE 機能では、FILE リストに対して、カップルファイルのペアがすべて揃っているかどうかのチェックが行われません。CONTINUE および IGNORECOUPLE の双方が指定されず、またその FILE リスト内でカップルファイルの一方が指定されていない場合、ADARES はエラーメッセージを発行して終了します。

**IGNOREEXP**：拡張構成要素ファイルリストのチェックの中止

FILE リスト内に Adabas 拡張コンポーネントファイルが含まれている場合、ADARES BACKOUT では通常、関連するすべてのコンポーネントファイルもリストに存在していることをチェックします。存在していない場合、ADARES は REGENERATE 処理を終了し、エラーメッセージを発行します。IGNOREEXP (または CONTINUE) を指定すると、関連するコンポーネントファイルに対するチェックが行われなくなります。

**INCLUDE**：再生成へのチェックポイントのインクルード

オプションキーワードである INCLUDE を FROMMCP または TOCP に指定すると、処理で再生成が開始または停止するチェックポイントがインクルードされます。つまり、そのチェックポイントに関連する機能が再実行されます。チェックポイント名は SYNS にする必要があります。これは、ADARES では SYNS チェックポイントに関連する機能のみが再実行可能であるためです。チェックポイント名とパラメータを組み合わせた 'SYNS,INCLUDE' はアポストロフィで囲む必要があります。

INCLUDE を指定しない場合 (デフォルト)、REGENERATE 処理は FROMMCP で指定したチェックポイントの直後に開始され、TOCP で指定したチェックポイントの直前で停止します。

INCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

**MTR**：マルチスレッド再生成の有効/無効の切り替え

MTR が YES の場合はマルチスレッド再生成機能をアクティブにし、NO の場合は使用不可にします。

マルチスレッド再生成機能がアクティブであるときは、性能を向上するために、PLOG 情報を含む複数のバッファが同時に Adabas ニュークリアスに送られます。マルチスレッド再生成機能が使用されないときは、一度に Adabas に送られるバッファは 1 つだけです。

ニュークリアス ADARUN パラメータ MODE=SINGLE が指定されると、MTR は自動的に NO に設定されます。Adabas がシングルユーザーモードで実行しているときはマルチスレッドは使用できません。

FILE パラメータを指定しないか、あるいは CONTINUE パラメータ付きで指定すると、MTR のデフォルト値は YES になります。この場合、マルチスレッド再生成はデータベース全体を排他制御し、通常は有効になります。

上記以外のデフォルト値は NO です。CONTINUE なしで FILE が指定されたときなど、マルチスレッド再生成が一部のファイルだけを排他制御する場合は、異なるファイルにアクセスする通常のアプリケーションを同時に実行できますが、運用中のアプリケーションの性能に悪い影響を与える可能性があります。

### NOAUTOBACKOUT：不完全トランザクションのバックアウトの禁止

NOAUTOBACKOUT を指定すると、REGENERATE 処理の最後で通常行われる不完全なトランザクションのバックアウトが停止されます。通常、ADARES では、下記の 2 つの条件が両方満たされる場合に、この機能の最後で、不完全なすべての論理トランザクションの自動バックアウトを実行します。

- REGENERATE がデータベース全体を対象にしている場合（FILE パラメータの省略）、または CONTINUE パラメータが指定されている場合
- TOCP パラメータの指定がなく、入力データセットの最後まで処理が実行される場合

1 つの Adabas セッションで作成されたマルチプロテクションログを処理するために、REGENERATE を連続して複数回実行する必要がある場合は、最後の入力ログに対してのみ自動バックアウトを実行する必要があります。したがって、最後の入力ログが使用される実行を除くすべての REGENERATE 実行に対して NOAUTOBACKOUT パラメータを指定しなければなりません。



**Note:** NOAUTOBACKOUT パラメータはシングルユーザーモードでは指定できません。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### NPCALLS：同時コール最大数

MTR=YES のとき、Adabas ニュークリアスに送られる同時コール数を制限するために NPCALLS パラメータを指定できます。

NPCALLSのデフォルト値は、FILEパラメータが指定されていないか、あるいはCONTINUE付きで指定されている場合は、ニュークリアス ADARUN パラメータ NT に 1 を加えた値か NC のどちらか小さい方になります。

FILEパラメータがCONTINUEなしで指定されている場合は、デフォルト値はニュークリアス ADARUN パラメータ NT に 1 を加えた値か、NC を 2 で割った値のどちらか小さい方になります。

NPCALLS は、主として、デフォルト値で許可される同時コール数を減らすために使います。同時コール数が少ないと、ADARESがニュークリアスにかかる負荷も少なくなります。これは、REGENERATE FILE と同時に実行されるアプリケーションプログラムが利用できるリソースを増やすときに特に役立ちます。

#### **PARALLELREAD**：他のユーザーによる読み取り専用ファイル利用の有効化

PARALLELREADパラメータを使用すると、データベース全体およびファイル指向の両方の機能で、ADARESREGENERATEで処理されているファイルに同時読み取り専用アクセスができるようになります。

- ファイル指向機能では、PARALLELREADを指定することにより、ADARESはレコードバッファに"EXU=file-list"を含むOPENコールを発行します。これは、ADARESがアクティブな間、他のユーザーのためにファイルへの読み取り専用アクセスを可能にします。
- FILEが指定されていないとき、またはCONTINUEが指定されているときは、PARALLELREADパラメータはデータベース全体でのセッション再生成に有効です。このパラメータによって、データベースセッションが再生成されているときでも読み取り専用ユーザーがデータベースにアクセスできるようになります。

更新コマンドは拒否されます。

再生成が行われているデータベースセッションで更新されたレコードを、同時アクセスしているユーザーが読み取った場合は、アプリケーションから見ると論理的に間違っただレコードイメージ、または矛盾を示すレスポンスコード（113など）が表示されることがあります。

 **Note:** PARALLELREADでのADARESオペレーション中に、アソシエータとデータストレージが位置的に異なると、ニュークリアスレスポンスが113または199になる可能性があります。

#### **PLOGDBID**：代替プロテクションログ ID

PLOGDBIDは、PLOGの取得元となった代替DBIDを指定します。ADARUNステートメントのDBIDパラメータで指定されたデータベース以外のデータベースのプロテクションログを使用して再生成を実行する場合は、PLOGDBIDを使用して代替プロテクションログのデータベースIDを指定します。デフォルトは、ADARUNで指定されたデータベースID (DBID) です。

**RAID**：ニュークリアスレスポンスコードの受け取りまたはユーティリティチェックポイントに続くアクション

RAID パラメータは、ユーティリティチェックポイント（ADADBS または Adabas Online System チェックポイントを除く）に到達したか、ファイルのニュークリアスレスポンスコードを受け取ったためにファイルが除外されるたびに、エラー 146 で再生成を終了します。

RAID を指定しない場合（デフォルト）、ADARES は 1 つのファイルが REGENERATE 処理から除外された後もその他のファイルの処理を続行します。

RAID は、Adabas Recovery Aid (ADARAI) によって構築されたリカバリジョブで使用するためのパラメータです。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**TOBLK**：TOCP ブロックの終了

TOBLK は、TOCP チェックポイントエントリが含まれるブロック番号を指定します。TOBLK は、TOCP が同時に指定されていないと指定できず、TOPLOG（指定されている場合）、あるいは PLOGNUM または FROMPLOG によって指定されたプロテクションログを参照します。

**TOCP**：再生成の終了チェックポイントブロック

TOCP は、REGENERATE 処理を直前で終了させるチェックポイントを指定します。処理は指定したチェックポイントまで継続されますが、このチェックポイントは処理の対象になりません。REGENERATE 処理をログの最後まで続ける場合は、TOCP を指定しないでください。TOCP は TOPLOG（指定されている場合）、あるいは PLOGNUM または FROMPLOG によって指定されたプロテクションログを参照します。

SYNS,INCLUDE オプションの詳細については、「INCLUDE（再生成へのチェックポイントのインクルード）」を参照してください。

**TOPLOG**：再生成の終了 PLOG セッション

TOPLOG は、指定された ADARES 機能によって処理される最終セッションを指定します。指定された TOPLOG セッションよりも番号が大きいセッションが、ADARES によって PLOG 入力ファイルで検出された場合、そのセッションは ADARES 処理の対象から除外されます。また TOPLOG が指定されない場合には、FROMPLOG セッションがデフォルトとなります。

## 例

---

### 例 1

```
ADARES REGENERATE PLOGNUM=4
```

すべてのファイルが再生成処理の対象になります。プロテクションログ番号は 4 です。再生成処理は、ログの先頭で始まり、ログの末尾で終了します。REGENERATE 処理の最後に、不完全なトランザクションが自動的にバックアウトされます。

## 例 2

```
ADARES REGENERATE
FILE=4,7, FROMPLOG=11, FROMCP=CH01, FROMBLK=106,
ADARES TOPLOG=12, TOCP=CH05, TOBLK=2031
```

再生成処理は、ファイル 4 と 7 に限定されます。チェックポイント CH01 と CH05 の間でファイル 4 と 7 に適用された更新のすべてが再適用されます。CH01 はデータプロテクションログ 11 のブロック 106 にあります。CH05 はデータプロテクションログ 12 のブロック 2031 にあります。上記の例と同様に、REGENERATE 処理の後に不完全なトランザクションの自動バックアウトは行われません。

## 例 3

```
ADARES REGENERATE EXCLUDE=10,11,12
```

ファイル 10～12 が REGENERATE データベース機能から除外されます。これらのファイルに対する変更は再生されません。

## 例 4

```
ADARES REGENERATE
ADARES FROMCP= 'SYNS, INCLUDE', FROMBLK=123
ADARES TOCP=SYNP, TOBLK=234
```

1. ADARES がデータベースを再生成します。
2. REGENERATE は PLOG ブロック 123 の SYNS チェックポイントで開始され、ADARES は関連する ADADBS/Adabas Online System 機能を再実行します。
3. REGENERATE はブロック 234 の SYNP チェックポイントの直前で終了し、ADARES は関連するユーティリティ機能を再生しません。

## 例 5

```
ADARES REGENERATE FILE=10
ADARES FROMCP= 'SYNS, INCLUDE', FROMBLK=345
ADARES TOCP= 'SYNS, INCLUDE', TOBLK=456
```

1. ADARES がファイル 10 を再生成します。
2. REGENERATE は PLOG ブロック 345 の SYNS チェックポイントで開始され、ADARES は関連する ADADBS/Adabas Online System 機能がファイル 10 に関するものである場合のみ再実行します。
3. REGENERATE はブロック 456 の SYNS チェックポイントで終了し、ADARES は関連する ADADBS/Adabas Online System 機能がファイル 10 に関するものである場合にのみ再生します。

## 例 6

ADARES REGENERATE

ADARES RAID

1. ADARES がデータベースを再生成します。
2. ADARES は、入力 PLOG で見つかったすべてのデータベース更新を再実行します。
3. ADARES は、ニュークリアスレスポンスコードを受け取るか、または ADADBS または Adabas Online System 以外のユーティリティチェックポイントに到達すると、ただちにエラー 146 で終了します。

# 164 REPAIR：データストレージブロックの修復

---

■ 構文 .....	910
■ 基本パラメータ .....	910
■ オプションパラメータ .....	911
■ 例 .....	911

 **Caution:** REPAIR 機能は、正しく使用しないとデータが失われることがあります。必ず Software AG 技術サポートと相談のうえ使用してください。

REPAIR 機能は、プロテクションログと ADASAV ユーティリティの出力を使用して1つまたは複数のデータストレージブロックを修復するために使用します。

### Notes:

1. 中断した REPAIR 機能は最初から再実行しなければなりません。
2. 最後の ADASAV SAVE 操作以降に ADAORD、ADALOD、ADADBS DEALLOCATE、ADASAV RESTORE FMOVE のいずれかのユーティリティ機能によって RABN 範囲が変更されている場合は、REPAIR 機能を実行しないでください。
3. DDSIIN/SIIN 入力は、"ADASAV SAVE (DD/SAVE*n*) ;, " の順序で連結する必要があります。

このchapterでは、次のトピックについて説明します。

## 構文

---

```
ADARES REPAIR DSRABN = { rabn | rabn - rabn }  
[FILE = { locked-file-list | entire-database } ]  
[NOUSERABEND]  
[TEST]
```

## 基本パラメータ

---

**DSRABN**：修復するデータストレージ **RABN**

DSRABN は、修復する1つ以上のデータストレージ RABN を指定します。1つの RABN を入力するか、または RABN の範囲（1000-1234 など）を指定できます。

## オプションパラメータ

### FILE：ロック対象ファイルリスト

FILEは1つ以上のファイルをロックして、REPAIR実行中に他のユーザーによって読み取りや更新が行われないようにします。指定したファイルのみがADARES REPAIRの排他使用のためにロックされます。リストに含まれていないファイルは、そのデータベースの他のユーザーが引き続き使用できます。FILEが指定されない場合は、データベース全体がロックされます。ユーザーキューは空でなければなりません。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TESTパラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

### 例 1

```
ADARES REPAIR DSRABN=1434,FILE=20
```

データストレージブロック 1434 を修復します。ファイルの処理中は、ファイル 20 のみがロックされます。

### 例 2

```
ADARES REPAIR DSRABN=1462-2543
```

データストレージブロック 1462～2543 を修復します。



# 165 マルチスレッド処理統計値

ADARES BACKOUT、BACKOUT DPLOG、または REGENERATE を MTR=YES を指定して実行すると、ユーティリティが正常終了した後で処理統計値の表が DDDRUCK に出力されます。例としては、次のようなものがあります。

## MULTI - THREADING PROCESSING STATISTIC

```
PLOG BLOCKS READ FROM INPUT          20472
PLOG RECORDS SENT TO ADABAS          764554
COMMANDS PROCESSED                   302273
TRANSACTION PROCESSED                55045
NUMBER OF ADABAS CALLS              56450
MAXIMUM CALLS IN PARALLEL            71
AVERAGE RECORD BUFFER SIZE          1403
```

フィールド	説明
PLOG BLOCKS READ FROM INPUT	入力プロテクションログから読み込まれた PLOG ブロック数
PLOG RECORDS SENT TO ADABAS	バックアウト処理または再生成処理に選択された PLOG レコード数
COMMANDS PROCESSED	処理された更新コマンド数 (N1、E1 など)
TRANSACTIONS PROCESSED	バックアウトまたは再生成されたトランザクション数
NUMBER OF ADABAS CALLS	バックアウトまたは再生成を実行するために発行された Adabas コール数
MAXIMUM CALLS IN PARALLEL	ニュークリアスによって同時処理されたバックアウトコールまたは再生成コールの最大数
AVERAGE CALLS IN PARALLEL	ADARES 実行中にニュークリアスによって同時処理されたバックアウトコールまたは再生成コールの平均数
AVERAGE RECORD BUFFER SIZE	バックアウトコールまたは再生成コールで使用されたレコードバッファの平均サイズ



# 166

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	916
▪ z/OS .....	924
▪ z/VM .....	931
▪ VSE .....	935

このセクションでは、ADARES を BS2000、z/OS、VSE、および z/VM の各システムで実行するために必要なジョブ制御情報を説明し、各ジョブストリームの例を示します。

 **Notes:**

1. プロテクションログをコピーするために使用される DD/SIAUS1/2 デバイスタイプが IDRC (ハードウェア圧縮) デバイスである場合は、BACKOUT 機能がサポートされない可能性があります。詳細については、このマニュアルの PLCOPY 機能の説明を参照してください。
2. オプションの Recovery Aid (ADARAI) で実行している場合は、すべての中間データセットをジョブ制御にカタログする必要もあります。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
シーケンシャルプロテクションログまたは ADASAV DDSAVE <sub>n</sub> 出力	DDSIIN	テープ/ディスク	COPY、REGENERATE、REPAIR 機能の入力ログ
マルチプロテクションログ	DDPLOG <sub>Rn</sub>	ディスク	PLCOPY 機能の入力ログおよび BACKOUT DPLOG
マルチコマンドログ	DDCLOG <sub>Rn</sub>	ディスク	CLCOPY 機能の入力ログ
シーケンシャルプロテクションログ	DDBACK	テープ	BACKOUT 機能 (BACKOUT DPLOG ではない) の入力ログ
コピーしたログ	DDSIAUS1	テープ/ディスク	COPY、CLCOPY、PLCOPY 機能の出力
デュアルコピーログ	DDSIAUS2	テープ/ディスク	COPY 機能 (TWO COPIES を指定) で 2 部作成する場合のみ必須
データストレージ	DDDATAR <sub>n</sub>	ディスク	REGENERATE で FROMMCP=SYN1 または SYN4 が指定された場合のみ必要
アソシエータ	DDASSOR <sub>n</sub>	ディスク	
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI 使用時に必要
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADARES パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/ DDPRINT		メッセージおよびコード
ADARES メッセージ	SYSLST DDDRUCK		メッセージおよびコード

**ADARES JCL の例 (BS2000)**

## デュアル/マルチコマンドログのコピー

SDF フォーマットの場合：

```

/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S COPY DUAL/MULTIPLE COMMAND LOG
/REMARK *
/DELETE-FILE ADAyyyyyy.AUS1
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.AUS1,PUB(SPACE=(960,480))
/SET-JOB-STEP
/ASS-SYSLST L.RES.CLCO
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDCLOGR1,ADAyyyyyy.CLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDCLOGR2,ADAyyyyyy.CLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDSIAUS1,ADAyyyyyy.AUS1
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES CLCOPY
/LOGOFF SYS-OUTPUT=DEL

```

ISP フォーマットの場合：

```

/.ADARES LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E S COPY DUAL/MULTIPLE COMMAND LOG
/REMARK *
/SYSFILE SYSLST=L.RES.CLCO
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.CLOGR1,LINK=DDCLOGR1,SHARUPD=YES
/FILE ADAyyyyyy.CLOGR2,LINK=DDCLOGR2,SHARUPD=YES
/FILE ADAyyyyyy.AUS1,LINK=DDSIAUS1,SPACE=(960,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B

```

```
ADARES CLCOPY
/LOGOFF NOSPOOL
```

## シーケンシャルプロテクションログのコピー

SDF フォーマットの場合：

```
/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S COPY SEQUENTIAL PLOG
/REMARK *
/DELETE-FILE ADAyyyyyy.SIBA.COP1
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.SIBA.COP1,PUB(SPACE=(960,480))
/SET-JOB-STEP
/ASS-SYSLST L.RES.COPY
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDSIIN,ADAYyyyyy.SIBA
/SET-FILE-LINK DDSIAUS1,ADAYyyyyy.SIBA.COP1
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES COPY PLOGNUM=ppp
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADARES LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A R E S COPY SEQUENTIAL PLOG
/REMARK *
/SYSFILE SYSLST=L.RES.COPY
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyyy.SIBA,LINK=DDSIIN
/FILE ADAYyyyyy.SIBA.COP1,LINK=DDSIAUS1,SPACE=(960,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADARES COPY PLOGNUM=ppp
/LOGOFF NOSPOOL
```

## デュアル/マルチプロテクションログのコピー

SDF フォーマットの場合：

```
/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S COPY DUAL/MULTIPLE PROTECTION LOG
/REMARK *
/DELETE-FILE ADAyyyyyy.AUS1
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.AUS1,PUB(SPACE=(960,480))
/SET-JOB-STEP
/DELETE-FILE ADAyyyyyy.AUS2
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.AUS2,PUB(SPACE=(960,480))
/SET-JOB-STEP
/ASS-SYSLST L.RES.PLCO
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAhrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR1,ADAyyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADAyyyyyy.PLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDSIAUS1,ADAyyyyyy.AUS1
/SET-FILE-LINK DDSIAUS2,ADAyyyyyy.AUS2
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES PLCOPY TWOCOPIES
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADARES LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E S COPY DUAL/MULTIPLE PROTECTION LOG
/REMARK *
/SYSFILE SYSLST=L.RES.PLCO
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.PLOGR1,LINK=DDPLOGR1,SHARUPD=YES
/FILE ADAyyyyyy.PLOGR2,LINK=DDPLOGR2,SHARUPD=YES
/FILE ADAyyyyyy.AUS1,LINK=DDSIAUS1,SPACE=(960,480)
/FILE ADAyyyyyy.AUS2,LINK=DDSIAUS2,SPACE=(960,480)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADARES PLCOPY TWOCOPIES
/LOGOFF NOSPOOL
```

## シーケンシャルプロテクションログを使用したバックアウト

**SDF** フォーマットの場合：

```
/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S BACKOUT FROM SEQUENTIAL PLOG
/REMARK *
/ASS-SYSLST L.RES.BACK
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDBACK,ADyyyyyy.BACK
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES BACKOUT
/LOGOFF SYS-OUTPUT=DEL
```

**ISP** フォーマットの場合：

```
/.ADARES LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E S BACKOUT FROM SEQUENTIAL PLOG
/REMARK *
/SYSFILE SYSLST=L.RES.BACK
/FILE ADA.MOD,LINK=DDLIB
/FILE ADyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADyyyyyy.BACK,LINK=DDBACK
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADARES BACKOUT
/LOGOFF NOSPOOL
```

## デュアル/マルチプロテクションログのバックアウト

SDF フォーマットの場合：

```
/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S BACKOUT FROM DUAL/MULTIPLE PLOG
/REMARK *
/ASS-SYSLST L.RES.BADP
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR1,ADyyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADyyyyyy.PLOGR2,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES BACKOUT DPLOG
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADARES LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E S BACKOUT FROM DUAL/MULTIPLE PLOG
/REMARK *
/SYSFILE SYSLST=L.RES.BADP
/FILE ADA.MOD,LINK=DDLIB
/FILE ADyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADyyyyyy.PLOGR1,LINK=DDPLOGR1,SHARUPD=YES
/FILE ADyyyyyy.PLOGR2,LINK=DDPLOGR2,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADARES BACKOUT DPLOG
/LOGOFF NOSPOOL
```

### 再生成機能

SDF フォーマットの場合：

```
/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S REGENERATE
/REMARK *
/ASS-SYSLST L.RES.REGE
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDSIIN,ADyyyyyy.SIBA
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES REGENERATE FILE=1,CONTINUE,PLOGNUM=ppp
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADARES LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E S REGENERATE
/REMARK *
/SYSFILE SYSLST=L.RES.REGE
/FILE ADA.MOD,LINK=DDLIB
/FILE ADyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADyyyyyy.SIBA,LINK=DDSIIN
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADARES REGENERATE FILE=1,CONTINUE,PLOGNUM=ppp
/LOGOFF NOSPOOL
```

## データストレージの修復

SDF フォーマットの場合：

```
/.ADARES LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A R E S REPAIR DATASTORAGE
/REMARK *
/ASS-SYSLST L.RES.REPA
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDSIIN,ADyyyyyy.SAVE
/SET-FILE-LINK DDSIIN01,ADyyyyyy.PLOG5
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
ADARES REPAIR DSRABN=3456 3490
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADARES LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A R E S REPAIR DATASTORAGE
/REMARK *
/SYSFILE SYSLST=L.RES.REPA
/FILE ADA.MOD,LINK=DDLIB
/FILE ADyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADyyyyyy.SAVE,LINK=DDSIIN
/FILE ADyyyyyy.PLOG5,LINK=DDSIIN01
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADARES,DB=yyyyyy,IDTNAME=ADABAS5B
```

```
ADARES REPAIR DSRABN=3456 3490
/LOGOFF NOSPOOL
```

## z/OS

データセット	DD 名	ストレージ	追加情報
シーケンシャルプロテクションログまたは ADASAVDDSAVEN 出力	DDSIIN	テープ/ディスク	COPY、REGENERATE、REPAIR 機能の入力ログ
マルチプロテクションログ	DDPLOGRn	ディスク	PLCOPY 機能の入力ログおよび BACKOUT DPLOG/MPLOG
マルチコマンドログ	DDCLOGRn	ディスク	CLCOPY 機能の入力ログ
シーケンシャルプロテクションログ	DDBACK	テープ	BACKOUT 機能 (BACKOUTDPLOG ではない) の入力ログ
コピーしたログ	DDSIAUS1	テープ/ディスク	COPY、CLCOPY、PLCOPY 機能の出力
デュアルコピーログ	DDSIAUS2	テープ/ディスク	COPY 機能 (TWOCPY を指定) で 2 部作成する場合のみ必須
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI 使用時に必要
データストレージ	DDDATARn	ディスク	REGENERATE で FROMMCP= SYN1 または SYN4 が指定された場合のみ必要
アソシエータ	DDASSORn	ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADARES パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADARES メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

### ADARES JCL の例 (z/OS)

シーケンシャルプロテクションログのコピー

```
//ADARESCP JOB
//*
//* ADARES: COPY SEQUENTIAL PROTECTION LOG
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
```

```
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDSIIN DD DSN=EXAMPLE.DByyyyy.SIBA, <=== PLOG
// VOL=SER=vvvvvv,DISP=OLD,UNIT=TAPE
//DDSIAUS1 DD DSN=EXAMPLE.DByyyyy.PLOG(+1), <=== PLOG COPY
// VOL=SER=vvvvvv,UNIT=TAPE,DISP=(NEW,CATLG)
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADARES COPY
/*
```

この例については、MVSJOBS データセットの ADARESCP を参照してください。

デュアル/マルチプロテクションログのコピー

```
//ADARESCD JOB
/*
/* ADARES: COPY DUAL/MULTIPLE PROTECTION LOG
/* TWO COPIES OF OUTPUT ARE TO BE CREATED
/*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.VVRS.LOAD <=== ADABAS LOAD
/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDPLOGR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1 <=== PLOG1
//DDPLOGR2 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2 <=== PLOG2
//DDSIAUS1 DD DSN=EXAMPLE.DByyyyy.PLOG1(+1), <=== PLOG COPY 1
// VOL=SER=vvvvvv,UNIT=TAPE,DISP=(NEW,CATLG)
//DDSIAUS2 DD DSN=EXAMPLE.DByyyyy.PLOG2(+1), <=== PLOG COPY 2
// VOL=SER=vvvvvv,UNIT=TAPE,DISP=(NEW,CATLG)
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X

//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADARES PLCOPY TWOCOPIES
/*
```

この例については、MVSJOBS データセットの ADARESCD を参照してください。

ニュークリアスクラスタの自動コピー／マージ プロテクションログ



**Note:** 中間データセットが CLCOPY と PLCOPY の両方に使用されるとき、それらが上書きされないように、データセット名をユニークにしなければならないことに注意してください。

必要な中間データセット MERGINT1 と MERGINT2 を割り当てるためのサンプル JCL は次のとおりです。

```
//ALLOC JOB
//*
//* Example to allocate the MERGINT1 and the MERGINT2 data sets
//*
//ALLOC EXEC PGM=IEFBR14
//MERGINT1 DD DISP=(NEW,CATLG,DELETE),DSN=EXAMPLE.PINTERI,
//          SPACE=(CYL,(1,10,0)),UNIT=3390,VOL=SER=volser,
//          RECFM=VB,BLKSIZE=27998,LRECL=27994
//MERGINT2 DD DISP=(NEW,CATLG,DELETE),DSN=EXAMPLE.PINTERO,
//          SPACE=(CYL,(1,10,0)),UNIT=3390,VOL=SER=volser,
//          RECFM=VB,BLKSIZE=27998,LRECL=27994
```

この例については、MVSJOBS データセットの ADARESMP を参照してください。

PPT を無視したニュークリアスクラスタプロテクションログの自動的なコピー／マージ

```
//ADARESIP JOB
//*
//* ADARES: COPY/MERGE DUAL/MULTIPLE PROTECTION LOGS FROM ALL
//* NUCLEI IN AN ADABAS CLUSTER
//* PPT IS TO BE IGNORED
//* THIS IS ONLY FOR EMERGENCY USE WHEN THE PPT HAS BEEN
//* OVER-WRITTEN - USE CAUTION WHEN SUBMITTING
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.VVRS.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.ASSOR1 <=== ASSO
//DDATAR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.DATAR1 <=== DATA
//DDPLOG01 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.PLOGR1 <=== PLOG1 NUC1
//DDPLOG02 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.PLOGR2 <=== PLOG2 NUC1
//DDPLOG03 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.PLOGR1A <=== PLOG1 NUC2
//DDPLOG04 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.PLOGR2A <=== PLOG2 NUC2
//DDPLOG05 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.PLOGR1B <=== PLOG1 NUC3
//DDPLOG06 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.PLOGR2B <=== PLOG2 NUC3
//MERGINT2 DD DISP=SHR,DSN=EXAMPLE.INTERO <=== INTER
```

```
//MERGINT1 DD DISP=SHR,DSN=EXAMPLE.INTERI <=== INTER
//DDSIAUS1 DD DSN=EXAMPLE.DBYYYYY.PLOG1(+1), <=== PLOG COPY
//
//          VOL=SER=ADAXXX,UNIT=TAPE,DISP=(NEW,CATLG)
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=XXX,DEVICE=3390,DBID=YYYYY
/*
//DDKARTE DD *
ADARES PLCOPY NOPPT
/*
//
```

MVSJOBS データセットの ADARESIP を参照してください。

デュアル/マルチコマンドログのコピー

```
//ADARESCC JOB
//*
//* ADARES: COPY DUAL/MULTIPLE COMMAND LOG
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.WORKR1 <=== WORK
//DDCLOGR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.CLOGR1 <=== CLOG1
//DDCLOGR2 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.CLOGR2 <=== CLOG2
//DDSIAUS1 DD DSN=EXAMPLE.DBYYYYY.CLOG, <=== OUTPUT OF
//          VOL=SER=vvvvvv,UNIT=TAPE,DISP=(NEW,CATLG) CLCOPY

//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADARES CLCOPY
/*
```

この例については、MVSJOBS データセットの ADARESCC を参照してください。

ニュークリアスクラスタの自動コピー/マージ コマンドログ



**Note:** 中間データセットが CLCOPY と PLCOPY の両方に使用されるとき、それらが上書きされないように、データセット名をユニークにしなければならないことに注意してください。

必要な中間データセット MERGINT1 と MERGINT2 を割り当てるためのサンプル JCL は次のとおりです。

```
//ALLOC JOB
//*
//* Example to allocate the MERGINT1 and the MERGINT2 data sets
//*
//ALLOC EXEC PGM=IEFBR14
//MERGINT1 DD DISP=(NEW,CATLG,DELETE),DSN=EXAMPLE.CINTERI,
//          SPACE=(CYL,(1,10,0)),UNIT=3390,VOL=SER=volser,
//          RECFM=VB,BLKSIZE=27998,LRECL=27994
//MERGINT2 DD DISP=(NEW,CATLG,DELETE),DSN=EXAMPLE.CINTERO,
//          SPACE=(CYL,(1,10,0)),UNIT=3390,VOL=SER=volser,
//          RECFM=VB,BLKSIZE=27998,LRECL=27994
```

この例については、MVSJOBS データセットの ADARESMC を参照してください。

ニュークリアスタ環境でのシーケンシャルコマンドログの手動マージ

```
//ADARESCM JOB
//*
//* ADARES: MERGE SEQUENTIAL COMMAND LOGS
//* FOR USE WITH AN ADABAS NUCLEUS CLUSTER
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.VVRS.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.WORKR1 <=== WORK
//DDCLOG01 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.CLOGR1A <=== CLOG1
//DDCLOG02 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.CLOGR1B <=== CLOG2
//DDCLOG03 DD DISP=SHR,DSN=EXAMPLE.DBYYYYY.CLOGR1C <=== CLOG3
//DDSIAUS1 DD DSN=EXAMPLE.DBYYYYY.CLOGM, <=== OUTPUT OF
//          VOL=SER=ADAXXX,UNIT=TAPE,DISP=(NEW,CATLG) CLOG MERGE
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=XXX,DEVICE=3390,DBID=YYYYY
/*
//DDKARTE DD *
ADARES MERGE CLOG,NUMLOG=3
/*
//
```

この例については、MVSJOBS データセットの ADARESCM を参照してください。

## シーケンシャルプロテクションログのバックアウト

```
//ADARESP JOB
//*
//* ADARES: BACKOUT FROM A SEQUENTIAL PLOG
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDBACK DD DSN=EXAMPLE.DByyyyyy.PLOG(-5), <=== PLOG TAPE
// UNIT=TAPE,DISP=OLD
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADARES BACKOUT PLOGNUM=nnn
/*
```

この例については、MVSJOBS データセットの ADARESP を参照してください。

## デュアル/マルチプロテクションログのバックアウト

```
//ADARESB JOB
//*
//* ADARES: BACKOUT FROM DUAL/MULTIPLE PLOG
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDPLOGR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1 <=== PLOG1
//DDPLOGR2 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2 <=== PLOG2
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
```

```
ADARES BACKOUT DPLOG
/*
```

この例については、MVSJOBS データセットの ADARESB を参照してください。

### 再生成機能

```
//ADARESR JOB
/*
/* ADARES: REGENERATE
/*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
/*

//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDSIIN DD DSN=EXAMPLE.DByyyyy.PLOG(-5), <=== PLOG TAPE
// UNIT=TAPE,DISP=OLD
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADARES REGENERATE FILE=1
/*
```

この例については、MVSJOBS データセットの ADARESR を参照してください。

### データストレージの修復

```
//ADARESRP JOB
/*
/* ADARES: REPAIR DATASTORAGE
/*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
/*

//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDSIIN DD DSN=EXAMPLE.DByyyyy.SAVE, <=== SAVE
OUTPUT
// DISP=OLD,UNIT=TAPE
// DD DSN=EXAMPLE.DByyyyy.PLOG(-5), <=== PLOG TAPE
// DISP=OLD,UNIT=TAPE
```

```
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxxx,DEVICE=dddd,DBID=yyyyyy
/*
//DDKARTE DD *
ADARES REPAIR DSRABN=3456-3490
```

この例については、MVSJOBS データセットの ADARESRP を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
シーケンシャルプロテクションログまたは ADASAV DDSAVE 出力	DDSIIN	テープ/ディスク	COPY、REGENERATE、REPAIR 機能の入力ログ
マルチプロテクションログ	DDPLOGRn	ディスク	PLCOPY および BACKOUT DPLOG の入力ログ
マルチコマンドログ	DDCLOGRn	ディスク	CLCOPY の入力ログ
シーケンシャルプロテクションログ	DDBACK	テープ	BACKOUT (BACKOUT DPLOG ではない) の入力ログ
コピーしたログ	DDSIAUS1	テープ/ディスク	CLCOPY および PLCOPY の出力
デュアルコピーログ	DDSIAUS2	テープ/ディスク	TWOCOPIES に必須
データストレージ	DDDATARn	ディスク	REGENERATE で FROMCP=SYN1 または SYN4 が指定された場合のみ必要
アソシエータ	DDASSORn	ディスク	
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI 使用時に必要
ADARUN パラメータ	DDCARD	ディスク/ 端末/リーダー	オペレーションマニュアル
ADARES パラメータ	DDKARTE	ディスク/ 端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/ 端末/プリンタ	メッセージおよびコード
ADARES メッセージ	DDDRUCK	ディスク/ 端末/プリンタ	メッセージおよびコード

## ADARES JCL の例 (z/VM)

### シーケンシャルプロテクションログのコピー

```
DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF DDSIIN,DSN=ADABASVv.SIBA,MODE=A
DATADEF DDSIAUS1,DSN=ADABASVv.SIAUS1,UNIT=181,VOL=SIBA01
DATADEF DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADARES.CONTROL,MODE=A
ADARUN
```

### RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

### ADARES CONTROL A1 の内容

```
ADARES COPY PLOGNUM=ppp
```

### デュアル/マルチプロテクションログのコピー

```
DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF DDPLOGR1,DSN=ADABASVv.PLOG1,VOL=PLOGV1
DATADEF DDPLOGR2,DSN=ADABASVv.PLOG2,VOL=PLOGV2
DATADEF DDSIAUS1,DSN=ADABASVv.SIAUS1,UNIT=181,VOL=SIBA01
DATADEF DDSIAUS2,DSN=ADABASVv.SIAUS2,UNIT=182,VOL=SIBA02
DATADEF DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADARES.CONTROL,MODE=A
ADARUN
```

### RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

### ADARES CONTROL A1 の内容

```
ADARES PLCOPY TWOCOPIES
```

### デュアル/マルチコマンドログのコピー

```
DATADef DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADef DDCLOGR1,DSN=ADABASVv.CLOG1,VOL=CLOGV1
DATADef DDCLOGR1,DSN=ADABASVv.CLOG1,VOL=CLOGV2
DATADef DDSIAUS1,DSN=ADABASVv.CLOG1,UNIT=181,VOL=CLOG01
DATADef DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADef DUMP,DUMMY
DATADef DDDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADef DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADef DDKARTE,DSN=ADARES.CONTROL,MODE=A
LOAD ADARUN ( START
```

### RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

### ADARES CONTROL A1 の内容

```
ADARES CLCOPY
```

### シーケンシャルプロテクションログを使用したバックアウト

```
DATADef DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADef DDBACK,DSN=ADABASVv.SIBA,MODE=A
DATADef DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADef DUMP,DUMMY
DATADef DDDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADef DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADef DDKARTE,DSN=BACKOUT.CONTROL,MODE=A
LOAD ADARUN ( START
```

### RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

## BACKOUT CONTROL A1 の内容

```
ADARES BACKOUT
```

### デュアル／マルチプロテクションログのバックアウト

```
DATADef DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADef DDPLOGR1,DSN=ADABASVv.PLOG1,VOL=PLOGV1
DATADef DDPLOGR2,DSN=ADABASVv.PLOG2,VOL=PLOGV2
DATADef DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADef DUMP,DUMMY
DATADef DDDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADef DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADef DDKARTE,DSN=BACKOUT.CONTROL,MODE=A
LOAD ADARUN ( START
```

## RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

## BACKOUT CONTROL A1 の内容

```
ADARES BACKOUT DPLOG
```

### 再生成機能

```
DATADef DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADef DDSIIN,DSN=ADABASVv.SIBA,MODE=A
DATADef DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADef DUMP,DUMMY
DATADef DDDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADef DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADef DDKARTE,DSN=REGENER.CONTROL,MODE=A
LOAD ADARUN ( START
```

## RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

## REGENER CONTROL A1 の内容

```
ADARES REGENERATE FILE=1
```

## データストレージの修復

```
DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDSIIN,DSN=ADABASV.V.SAVE,MODE=A
DATADEF DDSIIN,DSN=ADABASV.V.SIBA,MODE=A,CONCAT=1
DATADEF DDPLOGR1,DSN=ADABASV.V.PLOG1,VOL=PLOGV1
DATADEF DDPLOGR2,DSN=ADABASV.V.PLOG2,VOL=PLOGV2
DATADEF DDPRINT,DSN=ADARES.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDRUCK,DSN=ADARES.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNRES.CONTROL,MODE=A
DATADEF DDKARTE,DSN=BACKOUT.CONTROL,MODE=A
LOAD ADARUN ( START
```

## RUNRES CONTROL A1 の内容

```
ADARUN PROG=ADARES,DEVICE=dddd,DB=yyyyyy
```

## BACKOUT CONTROL A1 の内容

```
ADARES REPAIR DSRABN=3456-3490
```

## VSE

データセット	記号	ストレージ	論理ユニット	追加情報
シーケンシャルプロテクションログまたは ADASAV SAVE <sub>n</sub> 出力	SIIN	テープ ディスク	SYS020 注参照	COPY、REGENERATE、REPAIR の入力ログ
マルチプロテクションログ	PLOGR <sub>n</sub>	ディスク	注参照	PLCOPY および BACKOUT DPLOG の入力ログ
マルチコマンドログ	CLOGR <sub>n</sub>	ディスク	注参照	CLCOPY の入力ログ
シーケンシャルプロテクションログ	BACK	テープ	SYS020	BACKOUT (BACKOUT DPLOG ではない) の入力ログ

データセット	記号	ストレージ	論理ユニット	追加情報
コピーしたログ	SIAUS1	テープ ディスク	SYS021 注参照	COPY、CLCOPY、PLCOPY の出力
デュアルコピーログ	SIAUS2	テープ ディスク	SYS022 注参照	TWOCOPIES に必須
データストレージ	DATARn	ディスク		REGENERATE で FROMCP= SYN1 または SYN4 が指定され た場合のみ必要
アソシエータ	ASSORn		注参照	
リカバリログ (RLOG)	RLOGR1	ディスク		ADARAI 使用時に必要
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 注参照	
ADARES パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	
ADARES メッセージ		プリンタ	SYS009	

 **Note:** すべてのプログラマ論理ユニットを使用できます。

### ADARES JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

以下に例とメンバの対応表を示します。

例	メンバ
シーケンシャルプロテクションログのコピー	ADARESCP.X
デュアル/マルチプロテクションログのコピー	ADARESCD.X
デュアル/マルチコマンドログのコピー	ADARESCC.X
再生成	ADARESR.X
シーケンシャルプロテクションログのバックアウト	ADARESP.X
デュアルプロテクションログのバックアウト	ADARESB.X
データストレージの修復	ADARESRP.X

シーケンシャルプロテクションログのコピー

```

* $$ JOB JNM=ADARESCP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESCP
*       COPY SEQUENTIAL PLOG(TAPE)
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS020,TAPE,D0
// PAUSE MOUNT INPUT TAPE ON TAPE cuu
// MTC REW,SYS020
// TLBL S1IN,'ADABAS.Vvr.SIBA'
// ASSGN SYS022,TAPE,D0
// PAUSE MOUNT SCRATCH TAPE ON TAPE cuu
// MTC REW,SYS022
// MTC WTM,SYS022,5
// MTC REW,SYS022
// TLBL SIAUS1,'ADABAS.Vvr.SIAUS1'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADARES COPY
/*
/&
* $$ EOJ

```

#### デュアル/マルチプロテクションログのコピー

```

* $$ JOB JNM=ADARESCD,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESCD
*       CLCOPY WITH OPTION TWOCOPIES(TAPE)
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS021,TAPE,D0
// ASSGN SYS022,TAPE,D0
// PAUSE MOUNT SCRATCH TAPE ON TAPES cu1 AND cu2
// MTC REW,SYS022
// MTC WTM,SYS022,5
// MTC REW,SYS022
// MTC REW,SYS021
// MTC WTM,SYS021,5
// MTC REW,SYS021
// TLBL SIAUS1,'ADABAS.Vvr.PLOGC1'
// TLBL SIAUS2,'ADABAS.Vvr.PLOGC2'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADARES PLCOPY TWOCOPIES
/*

```

```
/&
* $$ EOJ
```

### デュアル/マルチコマンドログのコピー

```
* $$ JOB JNM=ADARESCC,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESCC
*       COPY DUAL/MULTIPLE COMMAND LOG
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS022,DISK,VOL=vvvvvv,SHR
// DLBL SIAUS1,'ADABAS.Vvr.CLOG',0,SD
// EXTENT SYS022,vvvvvv,1,0,sssss,nnnnn
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADARES CLCOPY
/*
/&
* $$ EOJ
```

### 再生成

```
* $$ JOB JNM=ADARESR,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESR
*       REGENERATE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS020,DISK,VOL=vvvvvv,SHR
// DLBL SIIN,'EXAMPLE.DByyyyy.PLOG'
// EXTENT SYS020,vvvvvv
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADARES REGENERATE FILE=1
/*
/&
* $$ EOJ
```

### シーケンシャルプロテクションログのバックアウト

```

* $$ JOB JNM=ADARESSP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESSP
*      BACKOUT FROM A SEQUENTIAL PLOG
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS020,TAPE
// PAUSE *** PLEASE MOUNT TAPE ***
// MTC REW,SYS020
// TLBL BACK,'DByyyyy.PLCOPY.TAPE'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADARES BACKOUT PLOGNUM=ppp
/*
/&
* $$ EOJ

```

### デュアルプロテクションログのバックアウト

```

* $$ JOB JNM=ADARESB,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESB
*      BACKOUT FROM DUAL PLOG
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADARES BACKOUT DPLOG
/*
/&
* $$ EOJ

```

### データストレージの修復

```

* $$ JOB JNM=ADARESRP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADARESRP
*      REPAIR DATASTORAGE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS020,TAPE
// PAUSE MOUNT INPUT TAPE ON cuu
// MTC REW,SYS020
// TLBL SIIN,'EXAMPLE.ADAyyyyy.SAVE1'
// TLBL SIIN01,'ADABAS.ADAyyyyy.PLOG5'          (*)
// EXEC ADARUN,SIZE=ADARUN

```

```
ADARUN  PROG=ADARES ,MODE=MULTI ,SVC=xxx ,DEVICE=dddd ,DBID=yyyyy  
/*  
ADARES  REPAIR  DSRABN=3456-3490  
/*  
/&  
* $$ EOI
```

(\*) シーケンシャル入力ファイル、**VSE** 連結のセクションを参照。

# 167 ADASAV ユーティリティ：データベースまたはファイルの保存／リストア

---

ADASAV ユーティリティを使用すると、データベース、指定した1つ以上のファイルの内容をシーケンシャルデータセットへセーブしたり、シーケンシャルデータセットからリストアすることができます。このchapterでは、次のトピックについて説明します。

- 機能概要
- **RESTONL**：オンラインソースからのデータベースのリストア
- **RESTONL GCB**：オンラインソースからのデータベース増分のリストア
- **RESTONL FILES**：オンラインソースから元の **RABN** へのファイルのリストア
- **RESTONL FMOVE** 機能：オンラインソースから別の **RABN** へのファイルのリストア
- **RESTORE**：オフラインソースからのデータベースのリストア
- **RESTORE GCB**：オフラインソースからのデータベース増分のリストア
- **RESTORE FILES** 機能：オフラインソースから元の **RABN** へのファイルのリストア
- **RESTORE FMOVE** 機能：オフラインソースから別の **RABN** へのファイルのリストア
- **RESTPLOG**：プロテクションログのみのリストア
- **SAVE**：データベースのセーブ
- **SAVE FILES**：指定ファイルのセーブ
- **JCL/JCS** の必要項目と指定例



# 168 機能概要

---

■ RESTONL 機能とRESTORE 機能 .....	944
■ Adabas リリースのサポート .....	945

ADASAV ユーティリティは、データベースあるいは指定した1つ以上のファイルの内容をシーケンシャルデータセットへセーブ、またはシーケンシャルデータセットからリストアします。

ADASAV の実行が必要な頻度は、データベース内のファイルの数やサイズ、および更新の量とタイプによって決まります。

データベースの規模が大きい場合、ADASAV 機能はデータベースが構成されているいくつかのディスクパックに並行して実行することができます。

Adabas Delta SaveFacility とともに使用するための特別な ADASAV 機能が用意されています。詳細については、『*Adabas Delta Save Facility* マニュアル』を参照してください。

このchapterでは、次のトピックについて説明します。

## RESTONL 機能とRESTORE 機能

---

RESTORE 機能または RESTONL 機能で処理する場合は、最初にアソシエータおよびデータストレージのデータセットをフォーマットする必要があります。処理が中断された場合、これらの機能が正常に再実行されるまでデータベース更新を行わないようにしてください。

RESTONL 機能は、Adabas ニュークリアスがアクティブな状態（つまりオンライン）で作成された SAVE データセットからリストアします。RESTORE 機能は、Adabas ニュークリアスがアクティブでない状態（つまりオフライン）で作成された SAVE データセットからリストアします。

RESTONL および RESTORE には副機能（GCB、FILES、FMOVE）があります。

- 副機能を指定しないと、RESTONL および RESTORE はデータベース全体をリストアします。
- GCB 副機能を指定すると、RESTONL および RESTORE は、ジェネラルコントロールブロック（GCB）、データベースのアソシエータ RABN3~30、および指定ファイルをリストアします。
- FILES 副機能を指定すると、RESTONL および RESTORE は、1つ以上のファイルを既存のデータベースの元の RABN にリストアします。
- FMOVE 副機能を指定すると、RESTONL および RESTORE は、既存データベースのエクステンションサイズを変更できるフリースペースに1つ以上のファイルをリストアします。

このsectionでは、次のトピックについて説明します。

- [RESTPLOG 機能とRESTONL 機能](#)

## ■ オンラインの SAVE とオフラインのSAVE

### RESTPLOG 機能とRESTONL 機能

オンライン SAVE 中に変更が生じると、RESTONL 機能に続いて RESTPLOG 機能が自動的に実行されます。RESTPLOG は、オンライン SAVE 処理中に生じたためにオンライン SAVE に含まれていない更新を適用します。

プロテクションログ (PLOG) のリストアによる更新が完了する前に RESTONL 機能や RESTONL FILES 機能が終了したときも、RESTPLOG が実行されます。RESTPLOG は、RESTONL 機能が正常に実行されなかったために適用されなかったデータベース更新を適用します。

### オンラインの SAVE とオフラインのSAVE

データベースまたは1つ以上のファイルをセーブするための SAVE 機能は、Adabas ニュークリアスがアクティブであっても (オンライン) アクティブでなくても (オフライン) 実行できます。リカバリエイドのオプションがアクティブな場合は、SAVE データベース処理が新しく RLOG 生成を開始します。

## Adabas リリースのサポート

Adabas データベース全体のリストアは、セーブデータセットの作成に使用したのと同じ Adabas リリースを使用することによってのみ可能です。ただし、個々のファイルのリストアは、Adabas の以前のバージョン (バージョン5.1まで) で作成されたセーブデータセットからでも、RESTORE FILES、RESTORE FMOVE、RESTONL FILES、または RESTONL FMOVE の機能を使用して行うことができます。

### Adabas バージョン 5 からファイルをリストアするときの考慮事項

Adabas 5 のセーブデータセットからリストアすると、ノーマルインデックスまたはアップパーインデックスに使用するために存在している可能性がある未使用の RABN チェーンが RESTORE FILE 機能によって破棄されます。これにより、このようなチェーンのすべてのブロックが "アクセス不可能なインデックスブロック" になり、ADAICK ICHECK 機能により WARNING-163 として報告されます。これらのブロックは、ADAORDREORFASSO などのリオーダ機能によってリオーダされるまでは、再使用されません。RESTORE FMOVE 機能は、未使用の RABN チェーンを破棄しませんが、その代わりに、新しい Adabas バージョン構造に変換します。

データベースに含まれるデバイスタイプがデータストレージと WORK で異なり、かつデータストレージのブロックサイズが WORK ブロックサイズよりも大きい場合は、Adabas 5 のセーブデータセットからのリストアが難しいことがあります。ファイルの最大圧縮レコード長が WORK ブロックサイズで許容される長さを超えている場合は、ADASAV によってリストアが拒否されることがあります。これは、以降の Adabas バージョンでプロテクションレコードヘッダーの長さが増加したことによるものです。このような場合にファイルをリストアするには、ADADEF

NEWWORK 機能を使用して、ブロックサイズを大きくした新しい WORK デバイスタイプをインストールする必要があります。

# 169 RESTONL : オンラインソースからのデータベースのリストア

---

▪ 条件 .....	948
▪ 結果 .....	949
▪ 構文 .....	949
▪ オプションのパラメータとサブパラメータ .....	949
▪ 例 .....	951

RESTONL 機能は、Adabas ニュークリアスがアクティブな状態で作成されたデータベースの SAVE データセットからデータベースをリストアします。



### Notes:

1. RESTONL (データベース) 処理が途中で中断した場合、その再実行は最初から行う必要があります。RESTONL (データベース) で PLOG をリストアしていたときに中断が発生した場合には、RESTPLOG 機能を用いてリストア処理を完了できます。処理を正常に終了するかまたはリストア処理を再実行するまで、データベースはアクセスできません。
2. ADASAV RESTONL (データベース) ジョブ制御内に DD/WORKnn の DD 名、記号名、またはリンク名が含まれていると、これらのデータセットはリセットされます。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶**手順 169.1. RESTONL (データベース) 機能を使用する場合は、次の条件が満たされている必要があります。**

- 1 正しい SAVE データセットを指定しなければなりません。RESTONL に使用したものと同じバージョンの Adabas を使用して、オンラインのデータベース SAVE 操作によって作成されている必要があります。
- 2 出力データベースは元のデータベースと同じ物理レイアウト (デバイスタイプ、エクステンツサイズ) でなければなりません。アソシエータおよびデータストレージのデータセットは事前にフォーマットされたものが存在していなければなりません。リストアする SAVE データセットはこのデータベースまたは異なるデータベースから作成されたものでもかまいません。
- 3 出力データベース、または出力データベースの DBID と一致するデータベースの Adabas ニュークリアスがアクティブであってははいけません。
- 4 SAVE 処理 (SAVE 実行の出力参照) 実行時のニュークリアスセッション中に作成された情報を含むプロテクションログ (PLOG) データセットを指定しなければなりません。他のセッションで作成された PLOG データセットも組み込むことができます。
- 5 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる 1 つの SAVE データセットに連結することができます。

## 結果

この機能を実行すると、データベースの物理ステータスが ADASAV SAVE 処理終了時と同じになります。

## 構文

```
ADASAV RESTONL [BUFNO = {number-of-buffers | 1}]
                [CLOGDEV = {CLOG1-device-type | ADARUN-device}]
                [DRIVES = {count | 1}]
                [EXCLUDE = file-list ]
                [NEWDBID = new-database-id ]
                [NEWDBNAME = new-database-name ]
                [NOUSERABEND]
                [OVERWRITE]
                [PLOGDEV = {PLOG-device-type | ADARUN-device}]
                [PLOGNUM = PLOG-number [, SYN1 = PLOG-block-number ]]
                [TEST]
```

## オプションのパラメータとサブパラメータ

### BUFNO：ドライブごとのバッファカウント

BUFNO 値に DRIVES パラメータ値を掛けたものが RESTONL 機能に対する固定割り当てバッファとなります。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはない、大量のスペースを割り当てることとなります。デフォルトは 1（1 ドライブ当たり 1 バッファ）です。

### CLOGDEV：コマンドログデバイスタイプ

デュアル/マルチコマンドログ（CLOG）のデバイスタイプを指定します。このパラメータは、CLOG のデバイスタイプが ADARUN DEVICE パラメータで指定されたもの（デフォルト）と異なる場合にのみ必要です。

### DRIVES：並行リストア用のテープドライブ

DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

### **EXCLUDE**：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル（リストアされないファイル）の番号をリストします。データベースリストアでは、

- EXCLUDE パラメータに指定されたファイルはリストアデータベースには含まれません。
- EXCLUDE パラメータに指定されたすべてのファイルがセーブデータセット上に存在している必要があります。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたリカバリジョブで使用するためのパラメータです。

### **NEWDBID**：リストアするデータベースの新規 ID

NEWDBID は、リストアするデータベースに対して新たに異なるデータベース ID を割り当てる場合に使用します。ID には 1~65,535 の範囲の数値を使用できます。Adabas Online System Security を導入した場合、DBID 999 は予約されます。

NEWDBID を指定する場合には ADARUN DBID パラメータで SAVE データセット上のデータベース ID を指定しなければなりません。

NEWDBID で指定された DBID の Adabas ニュークリアスは、アクティブであってはけません。

### **NEWDBNAME**：リストアするデータベースの新規名

NEWDBNAME は、リストア対象データベースに対して新たな名前を割り当てる場合に使用します。NEWDBNAME が省略された場合には、リストア対象データベースに元の名前がそのまま使用されます。

### **NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### **OVERWRITE**：既存データベースの上書き

リストア処理で既存のデータベースを上書きする場合、OVERWRITE パラメータを指定する必要があります。上書きされるデータベースの Adabas ニュークリアスは、アクティブであってはけません。

### **PLOGDEV**：プロテクションログのデバイスタイプ

デュアル/マルチプロテクションログ (PLOG) に割り当てるデバイスタイプを指定します。このパラメータは、PLOG のデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

**PLOGNUM**：プロテクションログ番号

PLOGNUM は、ADASAV SAVE のアクティブ時に使用したニュークリアスプロテクションログの番号を指定します（オンラインSAVE機能の出力リスト参照）。複数のニュークリアスセッションのシーケンシャルプロテクション（SIBA）ログは、連結できます。ADASAV は、PLOGNUM 値よりも小さい番号をもったプロテクションログをスキップします。PLOGNUM の指定は任意です。

PLOGNUM を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

**SYN1**：開始ブロック番号

SYN1 は、対応する SAVE 処理を開始する SYN1 チェックポイントが含まれるブロック番号を指定します。オンラインSAVE機能の出力リストを参照してください。このパラメータは任意指定です。

SYN1 を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

**例 1**

```
ADASAV RESTONL
```

ニュークリアスがアクティブな状態（オンライン）でセーブされたデータベースをリストアします。リストア処理に必要なプロテクションログ番号および SYN1 ブロック番号は、ADASAV によって自動的に決定されます。

**例 2**

```
ADASAV RESTONL
ADASAV EXCLUDE=255
ADASAV EXCLUDE=400
```

オンラインセーブデータセットから行うデータベースのリストアからファイル 255 と 400 を除外します。



# 170 RESTONL GCB : オンラインソースからのデータ ベース増分のリストア

---

▪ 条件 .....	954
▪ 結果 .....	955
▪ 構文 .....	955
▪ オプションのパラメータとサブパラメータ .....	956
▪ 例 .....	958

RESTONL GCB 機能では、Adabas ニュークリアスがアクティブだったときに作成されたデータベースの SAVE データセットから、次のものをリストアします。

- ジェネラルコントロールブロック (GCB)
- データベースのアソシエータ RABN 3~30
- チェックポイントファイル
- セキュリティファイル (存在する場合)
- FILES パラメータで指定された全ファイル



### Notes:

1. RESTONL GCB 処理が途中で中断した場合、その再実行は最初から行う必要があります。RESTONL GCB で PLOG をリストアしていたときに中断が発生した場合には、RESTPLOG 機能を用いてリストア処理を完了させることができます。処理を正常に終了するかまたはリストア処理を再実行するまで、データベースはアクセスできません。
2. ADASAV RESTONL GCB ジョブ制御内に DD/WORKnn の DD 名、記号名、またはリンク名が存在していると、これらのデータセットはリセットされます。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶ **手順 170.1. RESTONL GCB 機能を使用する場合は、次の条件が満たされている必要があります。**

- 1 正しいSAVEデータセットを指定しなければなりません。このデータセットは、RESTONL に使用したものと同一バージョンの Adabas を使用して、オンラインのデータベース SAVE 操作によって作成されたものである必要があります。また、リストアするファイルが含まれている必要もあります。
- 2 出力データベースは元のデータベースと同じ物理レイアウト（デバイスタイプ、エクステントサイズ）でなければなりません。アソシエータおよびデータストレージのデータセットは事前にフォーマットされたものが存在していなければなりません。リストアする SAVE データセットはこのデータベースから作成されたものでも、別のデータベースから作成されたものでもかまいません。
- 3 出力データベース、または出力データベースの DBID と一致するデータベースの Adabas ニュークリアスがアクティブであってはけません。
- 4 SAVE 処理（SAVE 実行の出力参照）実行時のニュークリアスセッション中に作成された情報を含むプロテクションログ（PLOG）データセットを指定しなければなりません。他のセッションで作成された PLOG データセットも組み込むことができます。

- SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる 1 つの SAVE データセットに連結することができます。
- マルチボリュームデータベースの SAVE データセットからほんのわずかのファイルをリストアする場合、リストアするファイルのデータを実際を含むテープボリュームだけを ADASAV ジョブ制御に指定する必要があります。対応する SYN1 チェックポイントと同様に SAVE 処理のジョブプロトコルは各ボリュームに含まれるファイルまたはファイル群の一部を示します。

## 結果

この機能を実行すると、データベースには、指定されたファイル、チェックポイントファイル、およびセキュリティファイルが格納されます。これらのファイルの物理ステータスは ADASAV SAVE 処理終了時と同じになります。

この処理は、RESTONL（データベース）と同等です。ただし、FILES パラメータに指定されていないファイルは除外されます。

 **Important:** ターゲットアソシエータおよびデータストレージデータセット内の既存のデータベースは完全に上書きされ、そのデータベース内のファイルはすべて失われます。

## 構文

```
ADASAV RESTONL GCB [BUFNO = {number-of-buffers | 1}]
                   [CLOGDEV = {CLOG1-device-type | ADARUN-device}]
                   [DRIVES = {count | 1}]
                   [EXCLUDE = file-list]
                   [FILES = file-list]
                   [NEWDBID = new-database-id]
                   [NEWDBNAME = new-database-name]
                   [NOUSERABEND]
                   [OVERWRITE]
                   [PLOGDEV = {PLOG-device-type | ADARUN-device}]
                   [PLOGNUM = PLOG-number [, SYN1 = PLOG-block-number]]
                   [TEST]
```

## オプションのパラメータとサブパラメータ

### BUFNO：ドライブごとのバッファカウント

BUFNO 値に DRIVES パラメータ値を掛けたものが RESTONL 機能に対する固定割り当てバッファとなります。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはなく、大量のスペースを割り当てることになります。デフォルトは 1 (1 ドライブ当たり 1 バッファ) です。

### CLOGDEV：コマンドログデバイスタイプ

コマンドログ (CLOG) のデバイスタイプを指定します。このパラメータは、CLOG のデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

### DRIVES：並行リストア用のテープドライブ

DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

### EXCLUDE：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル (リストアされないファイル) の番号をリストします。

EXCLUDE パラメータに指定されたファイルはリストアデータベースには含まれません。

EXCLUDE パラメータに指定されたすべてのファイルがセーブデータセット上に存在している必要があります。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

### FILES：リストア対象ファイル

FILES は、データベースリストア処理に含まれるファイル (群) を指定します。チェックポイントファイルとセキュリティファイル (存在する場合) は常にリストアされます。指定ファイルが Adabas 拡張ファイルのコンポーネントファイルである場合、拡張ファイルの全コンポーネントファイルも指定しなければなりません。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。

### NEWDBID：リストアするデータベースの新規 ID

NEWDBID は、リストアするデータベースに対して新たに異なるデータベース ID を割り当てる場合に使用します。ID には 1~65,535 の範囲の数値を使用できます。Adabas Online System Security を導入した場合、DBID 999 は予約されます。

NEWDBID を指定する場合には ADARUN DBID パラメータで SAVE データセット上のデータベース ID を指定しなければなりません。

NEWDBID で指定された DBID の Adabas ニュークリアスは、アクティブであってはけません。

**NEWDBNAME**：新データベース名

NEWDBNAME は、リストア対象データベースに対して新たな名前を割り当てる場合に使用します。NEWDBNAME が省略された場合には、リストア対象データベースに元の名前がそのまま使用されます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**OVERWRITE**：既存データベースの上書き

リストア処理で既存のデータベースを上書きする場合、OVERWRITE パラメータを指定する必要があります。

上書きされるデータベースの Adabas ニュークリアスは、アクティブであってはけません。

**PLOGDEV**：プロテクションログのデバイスタイプ

デュアル/マルチプロテクションログ（PLOG）のデバイスタイプを指定します。このパラメータは、PLOG のデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

**PLOGNUM**：プロテクションログ番号

PLOGNUM は、ADASAV SAVE のアクティブ時に使用したニュークリアスプロテクションログの番号を指定します（オンラインSAVE機能の出力リスト参照）。複数のニュークリアスセッションのシーケンシャルプロテクション（SIBA）ログは、連結できます。ADASAV は、PLOGNUM 値よりも小さい番号をもったプロテクションログをスキップします。PLOGNUM パラメータの指定は任意です。

PLOGNUM を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

**SYN1**：開始ブロック番号

SYN1 は、対応する SAVE 処理を開始する SYN1 チェックポイントが含まれるプロテクションログのブロック番号を指定します。オンラインSAVE機能の出力リストを参照してください。このパラメータは任意指定です。

SYN1 を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

---

### 例 1

```
ADASAV RESTONL GCB
```

ニュークリアスがアクティブだった（オンライン）ときにセーブされたデータベース GCB がリストアされます。チェックポイントファイルとセキュリティファイルもリストアされます。リストア処理を自動的に行うために必要なプロテクションログ番号とSYN1ブロック番号がADASAVによって決定されます。

### 例 2

```
ADASAV RESTONL GCB,FILES=3,4,5,OVERWRITE  
ADASAV PLOGNUM=15,SYN1=20
```

ファイル3、4、5が、チェックポイントファイルやセキュリティファイルとともにリストアされます。プロテクションログ番号は15、SYN1チェックポイントが含まれるブロックは20です。元のデータベースは上書きされます。

# 171      RESTONLFILES : オンラインソースから元のRABN へのファイルのリストア

---

▪ 条件 .....	960
▪ 結果 .....	961
▪ 構文 .....	962
▪ オプションのパラメータとサブパラメータ .....	962
▪ 例 .....	965

RESTONL FILES 機能では、Adabas ニュークリアスがアクティブ（オンライン）だったときに作成されたファイルまたはデータベースの SAVE データセットから、ファイルをリストアします。1 つ以上のファイルをリストアできます。ファイルは、既存のデータベースに元の RABN でリストアされます。



### Notes:

1. 中断した RESTONL FILES 処理は、最初から再実行する必要があります。RESTONL FILES で PLOG のリストア時に中断が発生した場合には、RESTPLOG 機能を使用して、リストア処理を完了することができます。処理を正常に終了するかまたはリストア処理を再実行するまで、リストア対象のファイルにはアクセスできません。
2. Adabas バージョン 5 からのチェックポイントファイルおよびセキュリティファイルはリストアできません。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶手順171.1.RESTONL FILES機能を使用する場合は、次の条件が満たされている必要があります。

- 1 正しい SAVE データセットを指定しなければなりません。指定するデータセットは、データベースまたはファイルの SAVE データセットで、リストアするファイルが含まれていなければなりません。バージョン 5.1 以上で作成した SAVE データセットを使用できます。
- 2 ファイルは、異なるデータベースから作成された SAVE テープを使用してリストアすることもできます。ただし、デバイスタイプは同じである必要があります。
- 3 データベースはすでに存在している必要があります。リストアするファイルは、そのデータベースからでも、または異なるデータベースから作成されたものでもかまいません。
- 4 リストアするファイルに使用された元のすべての RABN は、フリー（フリースペーステーブルに従って使用できる）であるか、または上書きされるファイルに占有されている必要があります。
- 5 Adabas ニュークリアスは出力データベースでアクティブであっても、アクティブでなくてもかまいません。

Adabas ニュークリアスがチェックポイントまたはセキュリティファイルをリストアするためにアクティブになっている場合、ADASAV ユーティリティに排他的なデータベース制御が必要になります。つまり、データベース上にアクティブなユーザーが存在しない必要があります。

- 6 SAVE 処理（SAVE 実行の出力参照）実行時のニュークリアスセッション中に作成された情報を含むプロテクションログ（PLOG）データセットを指定しなければなりません。他のセッションで作成された PLOG データセットも組み込むことができます。リストアするファ

イルがオンライン SAVE 処理中に変更されなかった場合、プロテクションログデータセットは省略できます。

- 7 SAVE テープが Adabas バージョン 5.1 で作成されている場合、オンライン SAVE 処理の始めに Adabas ニュークリアスによって書かれた SYN1/SYN4 チェックポイントの位置を指定しなければなりません。
- 8 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる 1 つの SAVE データセットに連結することができます。
- 9 マルチボリュームデータベースの SAVE データセットからほんのわずかのファイルをリストアする場合、リストアするファイルのデータを実際に含むテープボリュームだけを ADASAV ジョブ制御に指定する必要があります。対応する SYN4 チェックポイントと同様に SAVE 処理のジョブプロトコルは各ボリュームに含まれるファイルまたはファイル群の一部を示します。
- 10 拡張ファイルおよびカップリングされたファイルのリストアまたは上書きは全体でしかできません。つまり、拡張ファイルの 1 ファイルを指定するときは、拡張ファイルの他のファイルもすべて指定しなければならないということです。カップリングの関係にある一方のファイルを指定する場合は、他方のファイルもすべて指定しなければなりません。
- 11 チェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルの上書きは、それぞれ、別のチェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルのみの場合に可能です。このようなファイルがすでに異なるファイル番号で存在している場合、チェックポイント、セキュリティ、トリガの各ファイルはリストアできません。
- 12 NEWFILES パラメータでリストアするファイルに新規ファイル番号を割り当てることができます。

## 結果

この機能を実行すると、指定したファイルの物理ステータスが ADASAV SAVE 処理の終了時と同じになります。

## 構文

```

ADASAV RESTONL FILES = file-list [ALLOCATION = { FORCE | NOFORCE }]
                                     [BUFNO = { number-of-buffers | 1 }]
                                     [DRIVES = { count | 1 }]
                                     [EXCLUDE = file-list ]
                                     [NEWFILES = file-list ]
                                     [NOUSERABEND]
                                     [OVERWRITE]
                                     [PASSWORD = 'password-list' ]
                                     [PLOGNUM = protection-log-number ]
                                     [ { SYN1 | SYN4 } = PLOG-block-number ]
                                     [TEST]

```

FILESのファイルリストは、リストアするファイル（群）を指定します。指定ファイルがAdabas拡張ファイルのコンポーネントファイルである場合、拡張ファイルの全コンポーネントファイルも指定しなければなりません。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。

指定したファイルリストが、対応するSAVE機能に使用したファイルリストと一致する必要はありません。ファイルリストが対応するSAVE機能に使用されなかった場合でも、ファイルリストを指定できます。

ファイルは、異なるデータベースから作成されたSAVEテープを使用してリストアすることもできます。ただし、デバイスタイプは同じである必要があります。

## オプションのパラメータとサブパラメータ

**ALLOCATION**：ファイルエクステント割り当てができない場合の動作

ALLOCATIONは、配置パラメータACRABN、DSRABN、NIRABNまたはUIRABNに従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

ALLOCATIONは、セーブデータセットのファイルから取得したRABN指定に基づいて行われます。

デフォルト（すなわちALLOCATION=FORCE）の場合は、RABN配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCEの場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

 **Note:** ALLOCATION=NOFORCE が指定された ADASAV RESTONL FILES 処理では、セーブデータセットからのリストアの完了後、プロテクションログからのリストアの完了までにエラーが発生すると、RESTPLOG を完了できません。

**BUFNO**：ドライブごとのバッファカウント

BUFNO 値に DRIVES パラメータ値を掛けたものが RESTONL 機能に対する固定割り当てバッファとなります。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはない、大量のスペースを割り当てることとなります。デフォルトは 1 (1 ドライブ当たり 1 バッファ) です。

**DRIVES**：並行リストア用のテープドライブ

DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

**EXCLUDE**：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル (リストアされないファイル) の番号をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

**NEWFILES** パラメータが

- 指定されていない場合、EXCLUDE パラメータに指定されたすべてのファイルが FILES パラメータにも指定されている必要があります。
- 指定されている場合、EXCLUDE パラメータに指定されたすべてのファイルが NEWFILES パラメータにも指定されている必要があります。この場合 EXCLUDE パラメータに指定されたファイル番号は、FILES パラメータに指定された古いファイル番号ではなく、NEWFILES の新しいファイル番号を参照します。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

**NEWFILES**：新規ファイル番号

NEWFILES パラメータは、FILES で指定された各ファイルに割り当てる新規ファイル番号を指定します。このパラメータはオプションです。新しいファイル番号が割り当てられない場合は、元のファイル番号が保持されます。NEWFILES は、拡張ファイル、物理的にカップリングされているファイル、またはレプリケートされたファイルには指定できません。

NEWFILES で指定したファイル番号がすでにデータベースに存在している場合、対応するファイルは、OVERWRITE パラメータも指定しないと、リストアされません。上書きされるファイルがパスワードで保護されている場合、対応する PASSWORD パラメータも指定する必要があります。

複数のファイルをリストアする場合、NEWFILES パラメータのファイル番号リストは、FILES パラメータのファイルリストに対応させなければなりません。あるファイルに新規ファイル

番号を割り当てない場合、NEWFILESのファイル番号リストの該当ファイル番号のエントリには、0を指定する必要があります。例を参照してください。

NEWFILESを使用して基本ファイルまたはLOBファイルのファイル番号を再割り当てできるのは、LOBファイルグループのファイルが両方ともリストアされる場合のみです。この場合、ADASAVはNEWFILESパラメータによって指定された新しいファイル番号を両ファイルに割り当て、2つのファイル間のリンクを適宜調整します。ただし、LOBファイルグループの1ファイルのみをリストアした場合、NEWFILESパラメータを使用して新しいファイル番号を割り当ててはできません。代わりにADADBSまたはAOSRENUMBER機能を使用してください。

#### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

#### OVERWRITE：既存ファイルの上書き

このパラメータが指定されると、既存ファイルの削除後にリストアが行われます。リストア対象のファイルがすでにデータベース内に存在している場合には、このパラメータが指定されていないとADASAVはそのファイルをスキップします。



**Note:** 不用意にデータベースを上書きしないために、OVERWRITEパラメータは、必ずFILESファイルリストの前ではなく、後ろに指定してください。

#### PASSWORD：ファイルパスワード/パスワード群

FILESファイルリストの1つ以上のファイルがパスワードで保護されている場合、パスワードまたはパスワードリストを指定します。これは、すでにデータベース内に存在していて上書きされるファイルに適用されます。NEWFILESパラメータが指定されている場合、PASSWORDパラメータは、新規ファイル番号に関連するパスワードを指定する必要があります。

複数のパスワード保護ファイルをリストアする場合、FILESリストの保護されたファイル番号の位置に対応する位置決め値として正しいパスワードを指定しなければなりません。

PASSWORDパラメータの詳細については、例を参照してください。パスワード保護ファイルを上書きするときは、Adabasニュークリアスがアクティブである必要があります。

#### PLOGNUM：プロテクションログ番号

PLOGNUMは、ADASAV SAVE処理がアクティブだったときに使用したニュークリアスプロテクションログ（PLOG）の番号を指定します（オンラインSAVE機能の出力リスト参照）。ADASAVバージョン5.2以上で作成したSAVEテープをリストアするとき、またはリストアするファイルがSAVE処理中に変更されなかったとき、このパラメータはオプションです。複数のニュークリアスセッションのシーケンシャルプロテクション（SIBA）ログは、連結できます。ADASAVは、PLOGNUM値よりも小さい番号をもったPLOGをスキップします。

PLOGNUM を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

 **Note:** バージョン 5.1 の SAVE データセットからリストアすることはできません。

#### SYN1|SYN4：開始ブロック番号

リストア処理を開始する SYN1/SYN4 チェックポイントを持つブロック番号を指定します（ブロック番号に対するオンライン SAVE 機能の出力リストを参照してください）。ADASAV バージョン 5.2 以上で作成された SAVE テープをリストアするとき、このパラメータは任意指定です。

SYN1/SYN4 を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

 **Note:** バージョン 5.1 の SAVE データセットからリストアすることはできません。

#### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

### 例 1

```
ADASAV RESTONL FILES=3,4,5, OVERWRITE
ADASAV      PASSWORD='PWDFIL3,,PWDFIL5'
ADASAV      PLOGNUM=15,SYN1=20
```

ファイル 3、4、5 をリストアします。ファイル 3 と 5 はパスワード保護されており、そのパスワードはそれぞれ PWDFIL3 と PWDFIL5 です。PLOG 番号は 15、SYN1 チェックポイントが含まれるブロックは 20 です。旧ファイルは上書きされます。

### 例 2

```
ADASAV RESTONL FILES=11,12,13,14,OVERWRITE
ADASAV      NEWFILES=16,0,17
```

ファイル 11、12、13、14 をリストアします。ファイル 11 と 13 はファイル 16、17 としてリストアされます。ファイル 12 と 14 は、対応する NEWFILES パラメータ値が 0 または指定されていないので、変更されません。ファイル 12、14、16、17 は（すでにデータベースに存在している場合）上書きされます。



# 172 RESTONLFMOVE 機能：オンラインソースから別の RABN へのファイルのリストア

---

▪ 条件 .....	968
▪ 結果 .....	969
▪ 構文 .....	970
▪ オプションパラメータ .....	971
▪ 例 .....	977

## RESTONL FMOVE 機能：オンラインソースから別の RABN へのファイルのリストア

---

RESTONL FMOVE 機能では、Adabas ニュークリアスがアクティブだったときに作成されたファイルまたはデータベースの SAVE データセットから、ファイルをリストアします。1 つ以上のファイルをリストアできます。ファイルは既存のデータベースのフリースペースにリストアされます。エクステントサイズを変更できます。



### Notes:

1. 中断した RESTONL FMOVE 処理は最初から再実行する必要があります。PLOG のリストア時に中断した RESTONL FMOVE 処理は、RESTPLOG 機能を使用してリカバリすることはできません。処理を正常に終了するかまたはリストア処理を再実行するまで、リストア対象のファイルにはアクセスできません。
2. Adabas バージョン 5 からのチェックポイントファイルおよびセキュリティファイルはリストアできません。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶ **手順 172.1. RESTONL FMOVE 機能を使用する場合は、次の条件が満たされている必要があります。**

- 1 正しい SAVE データセットを指定しなければなりません。指定するデータセットは、データベースまたはファイルの SAVE データセットで、リストアするファイルが含まれていなければなりません。Adabas バージョン 5.1 以上の SAVE データセットを使用できます。
- 2 データベースはすでに存在している必要があります。リストアするファイルは、そのデータベースからでも、または異なるデータベースから作成されたものでもかまいません。
- 3 FMOVE のファイルリストは、新規 RABN を使用してリストアするファイルを指定します。RABN は、使用された元の RABN と同じデバイスタイプに割り当てなければなりません。
- 4 リストアするファイルに対して、フリースペース（フリースペーステーブルに従って使用できる）または上書きされるファイルが使用するスペースは、データベースで十分に利用できる大きさを必要とします。
- 5 Adabas ニュークリアスは出力データベースでアクティブであっても、アクティブでなくてもかまいません。

Adabas ニュークリアスがチェックポイントまたはセキュリティファイルをリストアするためにアクティブになっている場合、ADASAV ユーティリティに排他的なデータベース制御が必要になります。つまり、データベース上にアクティブなユーザーが存在しない必要があります。

- 6 SAVE 処理（SAVE 実行の出力参照）実行時のニュークリアスセッション中に作成された情報を含むプロテクションログ（PLOG）データセットを指定しなければなりません。他のセッションで作成された PLOG データセットも組み込むことができます。リストアするファ

イルがオンライン SAVE 処理中に変更されなかった場合、プロテクションログデータセットは省略できます。

- 7 SAVE テープが Adabas バージョン 5.1 で作成されている場合、オンライン SAVE 処理の始めに Adabas ニュークリアスによって書かれた SYN1/SYN4 チェックポイントの位置を指定しなければなりません。
- 8 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる 1 つの SAVE データセットに連結することができます。
- 9 マルチボリュームデータベースの SAVE データセットからほんのわずかのファイルをリストアする場合、リストアするファイルのデータを実際に含むテープボリュームだけを ADASAV ジョブ制御に指定する必要があります。対応する SYN4 チェックポイントと同様に SAVE 処理のジョブプロトコルは各ボリュームに含まれるファイルまたはファイル群の一部を示します。
- 10 拡張ファイルおよびカップリングされたファイルのリストアまたは上書きは全体でしかできません。つまり、拡張ファイルの 1 ファイルを指定するときは、拡張ファイルの他のファイルもすべて指定しなければならないということです。カップリングの関係にある一方のファイルを指定する場合は、他方のファイルもすべて指定しなければなりません。
- 11 チェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルの上書きは、それぞれ、別のチェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルのみの場合に可能です。このようなファイルがすでに異なるファイル番号で存在している場合、チェックポイント、セキュリティ、トリガの各ファイルはリストアできません。
- 12 NEWFILES パラメータでリストアするファイルに新規ファイル番号を割り当てることができます。

## 結果

---

この機能を実行すると、指定されたファイルの内容が ADASAV SAVE 処理の終了時と同じになります。ただし、同じデータベースブロック内である必要はありません。

## 構文

---

```
ADASAV RESTONL FMOVE = file-list [ACRABN = AC-start-rabn-list ]  
[AC2RABN = AC2-start-rabn-list ]  
[ALLOCATION = { FORCE | NOFORCE } ]  
[ASSOVOLUME = 'Associator-extent-volume' ]  
[BUFNO = { number-of-buffers | 1 } ]  
[DATAVOLUME = 'Data-Storage-extent-volume' ]  
[DRIVES = { count | 1 } ]  
[DSRABN = DS-start-rabn-list ]  
[DSSIZE = DS-size-list ]  
[EXCLUDE = file-list ]  
[MAXISN = isn-count-list ]  
[MAXISN2 = isn-count-list ]  
[NEWFILES = file-list ]  
[NIRABN = Ni-start-rabn-list ]  
[NISIZE = Ni-size-list ]  
[NOUSERABEND]  
[OVERWRITE]  
[PASSWORD = 'password-list' ]  
[PLOGNUM = protection-log-number ]  
  [ { SYN1 | SYN4 } = PLOG-block-number ]  
[TEST]  
[UIRABN = Ui-start-rabn-list ]  
[UISIZE = Ui-size-list ]
```

FMOVE のファイルリストは、新規 RABN を使用してリストアするファイルを指定します。  
RABN は、使用された元の RABN と同じデバイスタイプに割り当てなければなりません。

指定されたファイルが Adabas 拡張ファイルのコンポーネントファイルの場合、その拡張ファイルの他のコンポーネントファイルもすべて指定する必要があります。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。

## オプションパラメータ

---

### ACRABN：アドレスコンバータ開始 RABN/RABN リスト

ACRABN は、FMOVE によって指定された各ファイルに対するアドレスコンバータの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して ACRABN を省略すると、アドレスコンバータの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、ACRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに ACRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

### AC2RABN：セカンダリアドレスコンバータ開始 RABN/RABN リスト

AC2RABN は、FMOVE によって指定された各ファイルに対するセカンダリアドレスコンバータの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して AC2RABN を省略すると、セカンダリアドレスコンバータの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。ファイルにセカンダリアドレスコンバータエクステン트가含まれていない場合、このパラメータは無視されます。

複数のファイルをリストアするとき、AC2RABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに AC2RABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。

### ALLOCATION：ファイルエクステンタ割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステンタ割り当てが得られない場合取るべき動作を指定します。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステンタ割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

### ASSOVOLUME：アソシエータエクステンタボリューム



**Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UI エクステンタ）を割り当てるボリュームを指定します。リクエストしたブロック数が指定ボリュームになかった場合、ADASAV は ASSOVOLUME パラメータに関係なく割り当てようとします。

## RESTONL FMOVE 機能：オンラインソースから別の RABN へのファイルのリストア

ACRABN、UIRABN、NIRABNが指定された場合、ADASAVは、対応するエクステントタイプを割り当てるとき、ASSOVOLUME の値を無視します。ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADASAV のデフォルト割り当てルールに従って割り当てられます。

複数ファイルをリストアする場合、ASSOVOLUME パラメータのボリュームリストは、FMOVE パラメータのファイルリストに対応させなければなりません。あるファイルにボリュームを指定しない場合、ボリュームリスト内の該当エントリは空のままにする必要があります。例を参照してください。

### BUFNO：ドライブごとのバッファカウント

BUFNO 値は、RESTONL 処理の固定バッファを割り当てます。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはなく、大量のスペースを割り当てることとなります。デフォルトは 1 (1 ドライブ当たり 1 バッファ) です。

### DATAVOLUME：データストレージエクステントボリューム



**Note:** DATAVOLUME 値は単一引用符でくくらなければなりません。

DATAVOLUME は、ファイルのデータストレージスペース (DS エクステント) が割り当てられるボリュームを指定します。DSSIZE でリクエストしたブロック数が指定ボリュームになかった場合、ADASAV は DATAVOLUME パラメータに関係なく割り当てようとします。

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。DATAVOLUME を指定しないと、データストレージスペースは、ADASAV のデフォルト割り当てルールに従って割り当てられます。

複数ファイルをリストアする場合、DATAVOLUME パラメータのボリュームリストは、FMOVE パラメータのファイルリストに対応させなければなりません。あるファイルにボリュームを指定しない場合、ボリュームリスト内の該当エントリは空のままにする必要があります。例を参照してください。

### DRIVES：並行リストア用のテープドライブ

ADASAV では、複数のセーブデータセットボリュームのファイルを、データベースの元の RABN とは異なる RABN に並行してリストアできます。DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

### DSRABN：データストレージ開始 RABN/RABN リスト

DSRABN は、FMOVE で指定された各ファイルに対するデータストレージの開始 RABN を指定します。DSRABN は、FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して DSRABN を省略すると、ファイルのデータストレージの位置は、使用された元の RABN と同じデバイスタイプのデータストレージの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、DSRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに DSRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

**DSSIZE**：新規データストレージサイズ

DSSIZE は、FMOVE で指定された各ファイルのデータストレージに割り当てられる新規サイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

サイズは、シリンダ数またはブロック数（数値に "B" を付加）で指定できます。最低でも、元のデータストレージと同じ大きさにする必要があります。

DSSIZE を省略すると、元のデータストレージサイズが使用されます。

複数ファイルをリストアする場合、DSSIZE パラメータのサイズリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルにサイズを指定しない場合、サイズリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

**EXCLUDE**：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル（リストアされないファイル）の番号をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

NEWFILES パラメータが

- 指定されていない場合、EXCLUDE パラメータに指定されたすべてのファイルが FMOVE パラメータにも指定されている必要があります。
- 指定されている場合、EXCLUDE パラメータに指定されたすべてのファイルが NEWFILES パラメータにも指定されている必要があります。この場合 EXCLUDE パラメータに指定されたファイル番号は、FMOVE パラメータに指定された古いファイル番号ではなく、NEWFILES の新しいファイル番号を参照します。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

**MAXISN**：新規最大 ISN

FMOVE で指定された各ファイルに割り当てる新規 ISN 数を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

値は、最低でも元の最大割り当て ISN (MAXISN) と同じ大きさにする必要があります。

MAXISN を省略すると、元の ISN 数が使用されます。

複数ファイルをリストアする場合、MAXISN パラメータの ISN 数のリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに ISN 数を指定しな

## RESTONL FMOVE 機能：オンラインソースから別の RABN へのファイルのリストア

い場合、ISN 数のリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

デバイスタイプが異なる複数のアソシエータエクステントでデータベースが構成されている場合、MAXISN が指定されており、かつオンラインセーブ処理中にニュークリアスが追加のアドレスコンバータエクステントを割り当てていると、ERROR-171 が発生することがあります。この場合は、エラーメッセージで示されているファイルの MAXISN パラメータを削除してから RESTONL FMOVE を再実行してください。

### MAXISN2：新規最大セカンダリ ISN

MAXISN は、ISN 内のセカンダリアドレスコンバータ (AC2) のサイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

値は、最低でも元の最大割り当て ISN (MAXISN2) と同じ大きさにする必要があります。

MAXISN2 を省略すると、元の ISN 数が使用されます。ファイルにセカンダリアドレスコンバータエクステントが含まれていない場合、このパラメータは無視されます。

複数のファイルをリストアする場合、MAXISN2 パラメータの ISN 数のリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに ISN 数を指定しない場合、ISN 数のリスト内の該当エントリには、0 を指定しなければなりません。

デバイスタイプが異なる複数のアソシエータエクステントでデータベースが構成されているときに、MAXISN が指定され、オンラインセーブ処理中にニュークリアスが追加のアドレスコンバータエクステントを割り当てていた場合は、ERROR-171 が発生することがあります。この場合は、エラーメッセージで示されているファイルの MAXISN2 パラメータを削除してから RESTONL FMOVE を再実行してください。

### NEWFILES：新規ファイル番号

NEWFILES パラメータは、FMOVE で指定された各ファイルに割り当てる新規ファイル番号を指定します。このパラメータは、オプションです。ファイル番号を新しく割り当てる必要がない場合は、元のファイル番号が保持されます。NEWFILES は、拡張ファイル、物理的にカップリングされているファイル、またはレプリケートされたファイルには指定できません。

NEWFILES で指定したファイル番号がすでにデータベースに存在している場合、対応するファイルは、OVERWRITE パラメータも指定しないと、リストアされません。上書きされるファイルがパスワードで保護されている場合、対応する PASSWORD パラメータも指定する必要があります。

複数のファイルをリストアする場合、NEWFILES パラメータのファイル番号リストは、FMOVE パラメータのファイルリストに対応させなければなりません。あるファイルに新規ファイル番号を割り当てない場合、NEWFILES のファイル番号リストの該当ファイル番号のエントリには、0 を指定する必要があります。例を参照してください。

NEWFILES を使用して基本ファイルまたは LOB ファイルのファイル番号を再割り当てできるのは、LOB ファイルグループのファイルが両方ともリストアされる場合のみです。この場合、ADASAV は NEWFILES パラメータによって指定された新しいファイル番号を両ファイルに割り当て、2つのファイル間のリンクを適宜調整します。ただし、LOB ファイルグループの1ファイルのみをリストアした場合、NEWFILES パラメータを使用して新しいファイル番号を割り当ててはできません。代わりに ADADBS または AOS RENUMBER 機能を使用してください。

**NIRABN**：ノーマルインデックスの開始 RABN/RABN リスト

NIRABN は、FMOVE で指定された各ファイルに対するノーマルインデックスの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して NIRABN を省略すると、ノーマルインデックスの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、NIRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに NIRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

**NISIZE**：新規ノーマルインデックスサイズ

NISIZE は、FMOVE で指定された各ファイルのノーマルインデックスに割り当てた新規サイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

サイズは、シリンダ数またはブロック数（数値に "B" を付加）で指定できます。最低でも、元のノーマルインデックスと同じ大きさにする必要があります。

NISIZE を省略すると、元のノーマルインデックスサイズが使用されます。

複数ファイルをリストアする場合、NISIZE パラメータのサイズリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルにサイズを指定しない場合、サイズリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**OVERWRITE**：既存ファイルの上書き

このパラメータが指定されると、既存ファイルの削除後にリストアが行われます。リストア対象のファイルがすでにデータベース内に存在している場合には、このパラメータが指定されていないと ADASAV はそのファイルをスキップします。

## RESTONL FMOVE 機能：オンラインソースから別の RABN へのファイルのリストア

 **Note:** 不用意にデータベースを上書きしないために、OVERWRITE パラメータは、必ず FMOVE ファイルリストの前ではなく後ろに指定してください。

### PASSWORD：Adabas Security のファイルパスワード

PASSWORD は、FILES または FMOVE ファイルリスト内のファイルがパスワード保護されている場合のパスワードまたはパスワードリストを指定します。これは、すでにデータベース内に存在していて上書きされるファイルに適用されます。NEWFILES パラメータが指定されている場合、PASSWORD パラメータは、新規ファイル番号に関連するパスワードを指定する必要があります。

複数のパスワード保護ファイルをリストアする場合は、正しいパスワードを指定する必要があります。このパスワードは、FILES リストまたは FMOVE リストの保護ファイル番号の位置に対応する、位置が固定されている値になります。PASSWORD パラメータの詳細については、[例](#)を参照してください。パスワード保護ファイルを上書きするときは、Adabas ニュークリアスをアクティブにしなければなりません。

### PLOGNUM：プロテクションログ番号

PLOGNUM は、ADASAV SAVE 処理がアクティブだったときに使用したニュークリアスプロテクションログ (PLOG) の番号を指定します (オンライン SAVE 機能の出力リスト参照)。ADASAV バージョン 5.2 以上で作成した SAVE テープをリストアするとき、またはリストアするファイルが SAVE 処理中に変更されなかったとき、このパラメータはオプションです。複数のニュークリアスセッションのシーケンシャルプロテクション (SIBA) ログは、連結できます。ADASAV は、PLOGNUM 値よりも小さい番号をもった PLOG をスキップします。

PLOGNUM を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

 **Note:** バージョン 5.1 の SAVE データセットからリストアすることはできません。

### SYN1|SYN4：開始ブロック番号

リストア処理を開始する SYN1/SYN4 チェックポイントを持つブロック番号を指定します (ブロック番号に対するオンライン SAVE 機能の出力リストを参照してください)。ADASAV バージョン 5.2 以上で作成された SAVE テープをリストアするとき、このパラメータは任意指定です。

SYN1/SYN4 を指定しない場合、ADASAV は SAVE データセットに格納された情報から正しい値を自動的に決定します。

 **Note:** バージョン 5.1 の SAVE データセットからリストアすることはできません。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定されたパラメータの構文のみをテストでき、値や変数の整合性はチェックできません。

**UIRABN**：アップパーインデックスの開始 RABN/RABN リスト

UIRABN は、FMOVE で指定された各ファイルに対するアップパーインデックスの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して UIRABN を省略すると、アップパーインデックスの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、UIRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに UIRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

**UISIZE**：新規アップパーインデックスサイズ

FMOVE で指定された各ファイルのアップパーインデックスに割り当てる新規サイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

サイズは、シリンダ数またはブロック数（数値に "B" を付加）で指定できます。最低でも、元のアップパーインデックスと同じ大きさにする必要があります。

UISIZE を省略すると、元のアップパーインデックスサイズが使用されます。

複数ファイルをリストアする場合、UISIZE パラメータのサイズリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルにサイズを指定しない場合、サイズリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

## 例

### 例 1

```
ADASAV RESTONL PLOGNUM=25,SYN1=160,OVERWRITE
ADASAV          FMOVE=1,2
ADASAV          ACRABN=2100,2300
ADASAV          DSRABN=1500,2000
```

## RESTONL FMOVE 機能：オンラインソースから別の RABN へのファイルのリストア

```
ADASAV          NIRABN=0,2380
ADASAV          UIRABN=2190
```

プロテクションログ25を使用します。SYN1チェックポイントが含まれるブロックは160です。ファイル1およびファイル2が削除されリストアされます。ファイル1は次の開始RABNを使用してリストアされます。

アドレスコンバータ	2100
データストレージ	1500
ノーマルインデックス	(ADASAVによる選択)
アップパーインデックス	2190

ファイル2は次の開始RABNを使用してリストアされます。

アドレスコンバータ	2300
データストレージ	2000
ノーマルインデックス	2380
アップパーインデックス	(ADASAVによる選択)

### 例2

```
ADASAV RESTONL
PLOGNUM=4711,SYN4=99,FMOVE=3,4,5,OVERWRITE
ADASAV          PASSWORD='PWD3, ,PWD5'
```

FMOVE ファイルリストで指定されたファイルを、以前と異なる RABN にリストアできます。ファイル3と5はパスワード保護されており、パスワードはそれぞれ PWD3 と PWD5 です。

### 例3

```
ADASAV RESTONL FMOVE=11,12,13,14,OVERWRITE
ADASAV          NEWFILES=16,0,17
```

ファイル11、12、13、14をリストアします。ファイル11と13はファイル16、17としてリストアされます。ファイル12と14は、対応するNEWFILESパラメータ値が0または指定されていないので、変更されません。ファイル12、14、16、17は（すでにデータベースに存在している場合）上書きされます。

# 173 RESTORE：オフラインソースからのデータベースのリストア

---

▪ 条件 .....	980
▪ 結果 .....	981
▪ 構文 .....	981
▪ オプションパラメータ .....	981
▪ 例 .....	983

RESTORE 機能では、Adabas ニュークリアスがアクティブでないときに作成されたデータベースの SAVE データセットから、データベースをリストアします。



### Notes:

1. RESTORE（データベース）処理が途中で中断した場合、その再実行は最初から行う必要があります。処理を正常に終了するかまたはリストア処理を再実行するまで、データベースはアクセスできません。
2. ADASAV RESTORE（データベース）のジョブ制御内に DDWORKnn/WORKnn の DD 名、記号名、またはリンク名が存在している場合、これらのデータセットはリセットされます。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶手順 173.1. RESTORE（データベース）機能を使用する場合は、次の条件が満たされている必要があります。

- 1 正しい SAVE データセットを指定しなければなりません。RESTORE に使用したものと同じバージョンの Adabas を使用して、オフラインのデータベース SAVE 操作によって作成されたものである必要があります。
- 2 出力データベースは元のデータベースと同じ物理レイアウト（デバイスタイプ、エクステンションサイズ）でなければなりません。アソシエータおよびデータストレージのデータセットは事前にフォーマットされたものが存在していなければなりません。リストアする SAVE データセットはこのデータベースまたは異なるデータベースから作成されたものでもかまいません。
- 3 出力データベース、または出力データベースの DBID と一致するデータベースの Adabas ニュークリアスがアクティブであってははいけません。
- 4 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる1つの SAVE データセットに連結することができます。

## 結果

この機能を実行すると、データベースの物理ステータスが ADASAV SAVE 処理時と同じになります。

## 構文

```
ADASAV RESTORE [BUFNO = {number-of-buffers | 1}]
                [CLOGDEV = {CLOG1-device-type | ADARUN-device}]
                [DRIVES = {count | 1}]
                [EXCLUDE = file-list ]
                [NEWDBID = new-database-id ]
                [NEWDBNAME = new-database-name ]
                [NOUSERABEND]
                [OVERWRITE]
                [PLOGDEV = {PLOG-device-type | ADARUN-device}]
                [TEST]
```

## オプションパラメータ

### BUFNO：ドライブごとのバッファカウント

BUFNO 値に DRIVES パラメータ値を掛けたものが RESTORE 機能に対する固定割り当てバッファとなります。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはなく、大量のスペースを割り当てることとなります。デフォルトは 1（1 ドライブ当たり 1 バッファ）です。

### CLOGDEV：コマンドログデバイスタイプ

デュアル/マルチコマンドログ（CLOG）に割り当てるデバイスタイプを指定します。このパラメータは、CLOG に使用されるデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

### DRIVES：並行リストア用のテープドライブ

DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

### **EXCLUDE**：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル（リストアされないファイル）の番号をリストします。データベースリストアでは、

- EXCLUDE パラメータに指定されたファイルはリストアデータベースには含まれません。
- EXCLUDE パラメータに指定されたすべてのファイルがセーブデータセット上に存在している必要があります。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は1度だけリスト可能です。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたリカバリジョブで使用するためのパラメータです。

### **NEWDBID**：リストアするデータベースの新規 ID

NEWDBID は、リストアするデータベースに対して新たに異なるデータベース ID を割り当てる場合に使用します。ID には 1～65,535 の範囲の数値を使用できます。Adabas Online System Security を導入した場合、DBID 999 は予約されます。

NEWDBID を指定する場合には ADARUN DBID パラメータで SAVE データセット上のデータベース ID を指定しなければなりません。

NEWDBID で指定された DBID の Adabas ニュークリアスは、アクティブであってはけません。

### **NEWDBNAME**：新データベース名

NEWDBNAME は、リストア対象データベースに対して新たな名前を割り当てる場合に使用します。NEWDBNAME が省略された場合には、リストア対象データベースに元の名前がそのまま使用されます。

### **NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### **OVERWRITE**：既存データベースの上書き

リストア処理で既存のデータベースを上書きする場合、OVERWRITE パラメータを指定する必要があります。

上書きされるデータベースの Adabas ニュークリアスは、アクティブであってはけません。

### **PLOGDEV**：プロテクションログのデバイスタイプ

デュアル/マルチプロテクションログ (PLOG) のデバイスタイプを指定します。このパラメータは、PLOG のデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

### 例 1

```
ADASAV RESTORE OVERWRITE
```

データベースがリストアされます。既存のデータベースが上書きされる場合があります。

### 例 2

```
ADASAV RESTORE EXCLUDE=10,11,12
```

オフラインのセーブデータセットからのデータベースのリストアでファイル 10～12 が除外されます。



# 174 RESTORE GCB : オフラインソースからのデータ ベース増分のリストア

---

▪ 条件 .....	986
▪ 結果 .....	987
▪ 構文 .....	987
▪ オプションパラメータ .....	988
▪ 例 .....	989

RESTORE GCB 機能では、Adabas ニュークリアスがアクティブでないときに作成されたデータベースの SAVE データセットから、次のものをリストアします。

- ジェネラルコントロールブロック (GCB)
- データベースのアソシエータ RABN 3~30
- チェックポイントファイル
- セキュリティファイル (存在する場合)
- FILES パラメータで指定された全ファイル



### Notes:

1. RESTORE GCB 処理が途中で中断した場合、その再実行は最初から行う必要があります。処理を正常に終了するかまたはリストア処理を再実行するまで、データベースはアクセスできません。
2. ADASAV RESTORE GCB ジョブ制御内に DDWORKnn/WORKnn の DD 名、記号名、またはリンク名が存在していると、これらのデータセットはリセットされます。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶手順 174.1. RESTORE GCB 機能を使用する場合は、次の条件が満たされている必要があります。

- 1 正しい SAVE データセットを指定しなければなりません。RESTORE に使用したものと同じバージョンの Adabas を使用して、オフラインのデータベース SAVE 操作によって作成されたものである必要があります。また、リストアするファイルが含まれている必要もあります。
- 2 出力データベースは元のデータベースと同じ物理レイアウト（デバイスタイプ、エクステンションサイズ）でなければなりません。アソシエータおよびデータストレージのデータセットは事前にフォーマットされたものが存在していなければなりません。リストアする SAVE データセットはこのデータベースまたは異なるデータベースから作成されたものでもかまいません。
- 3 出力データベース、または出力データベースの DBID と一致するデータベースの Adabas ニュークリアスがアクティブであってははいけません。
- 4 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる 1 つの SAVE データセットに連結することができます。

- マルチボリュームデータベースの SAVE データセットからほんのわずかのファイルをリストアする場合、リストアするファイルのデータを実際を含むテープボリュームだけを ADASAV ジョブ制御に指定する必要があります。対応する SYNВ チェックポイントと同様に SAVE 処理のジョブプロトコルは各ボリュームに含まれるファイルまたはファイル群の一部を示します。

## 結果

この機能を実行すると、データベースには、指定されたファイル、チェックポイントファイル、およびセキュリティファイルが格納されます。これらのファイルの物理ステータスは ADASAV SAVE 処理時と同じになります。

この処理は、RESTORE（データベース）と同等です。ただし、FILES パラメータに指定されていないファイルは除外されます。

 **Important:** ターゲットアソシエータおよびデータストレージデータセット内の既存のデータベースは完全に上書きされ、そのデータベース内のファイルはすべて失われます。

## 構文

```
ADASAV RESTORE GCB [BUFNO = {number-of-buffers | 1}]
                   [CLOGDEV = {CLOG1-device-type | ADARUN-device}]
                   [DRIVES = {count | 1}]
                   [EXCLUDE = file-list]
                   [FILES = file-list]
                   [NEWDBID = new-database-id]
                   [NEWDBNAME = new-database-name]
                   [NOUSERABEND]
                   [OVERWRITE]
                   [PLOGDEV = {PLOG-device-type | ADARUN-device}]
                   [TEST]
```

## オプションパラメータ

### BUFNO：ドライブごとのバッファアカウント

BUFNO 値に DRIVES パラメータ値を掛けたものが RESTORE 機能に対する固定割り当てバッファとなります。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはなく、大量のスペースを割り当てることになります。デフォルトは 1 (1 ドライブ当たり 1 バッファ) です。

### CLOGDEV：コマンドログデバイスタイプ

コマンドログ (CLOG) のデバイスタイプを指定します。このパラメータは、CLOG のデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

### DRIVES：並行リストア用のテープドライブ

DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

### EXCLUDE：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル (リストアされないファイル) の番号をリストします。

EXCLUDE パラメータに指定されたファイルはリストアデータベースには含まれません。

EXCLUDE パラメータに指定されたすべてのファイルがセーブデータセット上に存在している必要があります。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

### FILES：リストア対象ファイル

FILES は、データベースリストア処理に含まれるファイル (群) を指定します。指定ファイルが Adabas 拡張ファイルのコンポーネントファイルである場合、拡張ファイルの全コンポーネントファイルも指定しなければなりません。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。チェックポイントファイルとセキュリティファイルは常にリストアされます。

### NEWDBID：リストアするデータベースの新規 ID

NEWDBID は、リストアするデータベースに対して新たに異なるデータベース ID を割り当てる場合に使用します。ID には 1~65,535 の範囲の数値を使用できます。Adabas Online System Security を導入した場合、DBID 999 は予約されます。

NEWDBID を指定する場合には ADARUN DBID パラメータで SAVE データセット上のデータベース ID を指定しなければなりません。

NEWDBID で指定された DBID の Adabas ニュークリアスは、アクティブであってはけません。

**NEWDBNAME**：新データベース名

NEWDBNAME は、リストア対象データベースに対して新たな名前を割り当てる場合に使用します。NEWDBNAME が省略された場合には、リストア対象データベースに元の名前がそのまま使用されます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**OVERWRITE**：既存データベースの上書き

リストア処理で既存のデータベースを上書きする場合、OVERWRITE パラメータを指定する必要があります。

上書きされるデータベースの Adabas ニュークリアスは、アクティブであってはけません。

**PLOGDEV**：プロテクションログのデバイスタイプ

デュアル/マルチプロテクションログ（PLOG）のデバイスタイプを指定します。このパラメータは、PLOG のデバイスタイプが ADARUN DEVICE パラメータの指定と異なる場合にのみ必要です。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

### 例 1

```
ADASAV RESTORE GCB,FILES=2,4,6,8
```

データベースアソシエータの GCB、RABN 3~30、チェックポイントファイル、セキュリティファイル、およびファイル 2、4、6、8 がリストアされます。

### 例 2

```
ADASAV RESTORE GCB,FILES=3,4,5,OVERWRITE
```

アソシエータの GCB およびファイル 3、4、5 がリストアされます。既存のデータベースは上書きされます。



# 175 RESTORE FILES 機能：オフラインソースから元の RABN へのファイルのリストア

---

▪ 条件 .....	992
▪ 結果 .....	993
▪ 構文 .....	993
▪ オプションパラメータ .....	994
▪ 例 .....	996

## RESTORE FILES 機能：オフラインソースから元の RABN へのファイルのリストア

---

RESTORE FILES 機能では、Adabas ニュークリアスがアクティブでないときに作成されたファイルまたはデータベースの SAVE データセット、または UTYPE=EXU で作成されたファイルの SAVE データセットから、ファイルをリストアします。1 つ以上のファイルをリストアできます。ファイルは、既存のデータベースに元の RABN でリストアされます。



### Notes:

1. 中断した RESTORE FILES 処理は最初から再実行する必要があります。処理を正常に終了するかまたはリストア処理を再実行するまで、リストア対象のファイルにはアクセスできません。
2. Adabas バージョン 5 からのチェックポイントファイルおよびセキュリティファイルはリストアできません。

この chapter では、次のトピックについて説明します。

## 条件

---

▶ **手順 175.1. RESTORE FILES 機能を使用する場合は、次の条件が満たされている必要があります。**

- 1 正しい SAVE データセットを指定しなければなりません。指定するデータセットは、データベースまたはファイルの SAVE データセットで、リストアするファイルが含まれていなければなりません。
- 2 ファイルは、異なるデータベースを使用して作成した SAVE データセットを使用してリストアすることができます。ただし、使用するデバイスタイプは同じである必要があります。
- 3 データベースはすでに存在している必要があります。リストアするファイルは、そのデータベースからでも、または異なるデータベースから作成されたものでもかまいません。Adabas バージョン 5.1 以上の SAVE データセットを使用できます。
- 4 リストアするファイルに使用された元のすべての RABN は、フリー（フリースペーステーブルに従って使用できる）であるか、または上書きされるファイルに占有されている必要があります。
- 5 Adabas ニュークリアスは出力データベースでアクティブであっても、アクティブでなくてもかまいません。

Adabas ニュークリアスがチェックポイントまたはセキュリティファイルをリストアするためにアクティブになっている場合、ADASAV ユーティリティに排他的なデータベース制御が必要になります。つまり、データベース上にアクティブなユーザーが存在しない必要があります。

- 6 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、

DRIVES パラメータなしでリストアされる1つのSAVE データセットに連結することができます。

- 7 マルチボリュームデータベースのSAVE データセットからほんのわずかのファイルをリストアする場合、リストアするファイルのデータを実際を含むテープボリュームだけを ADASAV ジョブ制御に指定する必要があります。対応する SYNВ チェックポイントと同様にSAVE 処理のジョブプロトコルは各ボリュームに含まれるファイルまたはファイル群の一部を示します。
- 8 拡張ファイルおよびカップリングされたファイルのリストアまたは上書きは全体でしかできません。つまり、拡張ファイルの1ファイルを指定するときは、拡張ファイルの他のファイルもすべて指定しなければならないということです。カップリングの関係にある一方のファイルを指定する場合は、他方のファイルもすべて指定しなければなりません。
- 9 チェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルの上書きは、それぞれ、別のチェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルのみの場合に可能です。このようなファイルがすでに異なるファイル番号で存在している場合、チェックポイント、セキュリティ、トリガの各ファイルはリストアできません。
- 10 NEWFILES パラメータでリストアするファイルに新規ファイル番号を割り当てることができます。

## 結果

この機能を実行すると、指定したファイルの物理ステータスが ADASAV SAVE 処理時と同じになります。

## 構文

```
ADASAV RESTORE FILES = file-list [ALLOCATION = {FORCE | NOFORCE}]  
[BUFNO = {number-of-buffers | 1}]  
[DRIVES = {count | 1}]  
[EXCLUDE = file-list ]  
[NEWFILES = file-list ]  
[NOUSERABEND]  
[OVERWRITE]  
[PASSWORD = 'password-list' ]  
[TEST]
```

FILES のファイルリストは、リストアするファイル（群）を指定します。

## RESTORE FILES 機能：オフラインソースから元のRABNへのファイルのリストア

---

Adabas 拡張ファイルの場合、アンカーファイルを含む拡張ファイルの全コンポーネントファイルを指定する必要があります。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。

指定したファイルリストが、対応する SAVE 機能に使用したファイルリストと一致する必要はありません。ファイルリストが対応する SAVE 機能に使用されなかった場合でも、ファイルリストを指定できます。

ファイルは、異なるデータベースを使用して作成した SAVE データセットを使用してリストアすることもできます。ただし、使用するデバイスタイプは同じである必要があります。

## オプションパラメータ

---

**ALLOCATION**：ファイルエクステント割り当てができない場合の動作

ALLOCATION は、配置パラメータ ACRRABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステント割り当てが得られない場合取るべき動作を指定します。

ALLOCATION は、セーブデータセットのファイルから取得した RABN 指定に基づいて行われます。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステント割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

**BUFNO**：ドライブごとのバッファカウント

BUFNO 値に DRIVES パラメータ値を掛けたものが RESTORE 機能に対する固定割り当てバッファとなります。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはない、大量のスペースを割り当てることとなります。デフォルトは 1（1 ドライブあたり 1 バッファ）です。

**DRIVES**：並行リストア用のテープドライブ

DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

**EXCLUDE**：指定したファイルをリストアから除外

EXCLUDE は、リストアオペレーションから除外するファイル（リストアされないファイル）の番号をリストします。

このパラメータは省略可能です。指定しなかった場合、ファイルは一切除外されません。ファイル番号は 1 度だけリスト可能です。

NEWFILES パラメータが

- 指定されていない場合、EXCLUDEパラメータに指定されたすべてのファイルがFILESパラメータにも指定されている必要があります。
- 指定されている場合、EXCLUDEパラメータに指定されたすべてのファイルがNEWFILESパラメータにも指定されている必要があります。この場合EXCLUDEパラメータに指定されたファイル番号は、FILESパラメータに指定された古いファイル番号ではなく、NEWFILESの新しいファイル番号を参照します。

EXCLUDEパラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

#### NEWFILES：新規ファイル番号

NEWFILESパラメータは、FILESで指定された各ファイルに割り当てる新規ファイル番号を指定します。このパラメータは、オプションです。ファイル番号を新しく割り当てる必要がない場合は、元のファイル番号が保持されます。NEWFILESは、拡張ファイル、物理的にカップリングされているファイル、またはレプリケートされたファイルには指定できません。

NEWFILESで指定したファイル番号がすでにデータベースに存在している場合、対応するファイルは、OVERWRITEパラメータも指定しないと、リストアされません。上書きされるファイルがパスワードで保護されている場合、対応するPASSWORDパラメータも指定する必要があります。

複数のファイルをリストアする場合、NEWFILESパラメータのファイル番号リストは、FILESパラメータのファイルリストに対応させなければなりません。あるファイルに新規ファイル番号を割り当てない場合、NEWFILESのファイル番号リストの該当ファイル番号のエントリには、0を指定する必要があります。例を参照してください。

NEWFILESを使用して基本ファイルまたはLOBファイルのファイル番号を再割り当てできるのは、LOBファイルグループのファイルが両方ともリストアされる場合のみです。この場合、ADASAVはNEWFILESパラメータによって指定された新しいファイル番号を両ファイルに割り当て、2つのファイル間のリンクを適宜調整します。ただし、LOBファイルグループの1ファイルのみをリストアした場合、NEWFILESパラメータを使用して新しいファイル番号を割り当てることができません。代わりにADADBSまたはAOSRENUMBER機能を使用してください。

#### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

## RESTORE FILES 機能：オフラインソースから元のRABNへのファイルのリストア

---

### OVERWRITE：既存ファイルの上書き

このパラメータが指定されると、既存ファイルの削除後にリストアが行われます。リストア対象のファイルがすでにデータベース内に存在している場合には、このパラメータが指定されていないと ADASAV はそのファイルをスキップします。



**Note:** 不用意にデータベースを上書きしないために、OVERWRITE パラメータは、必ず FILES ファイルリストの前ではなく、後ろに指定してください。

### PASSWORD

FILES に指定された 1 つ以上のファイルがパスワードで保護されている場合は、パスワードまたはパスワードリストを指定します。このことは、すでにデータベース内に存在し、上書きされるファイルにのみ当てはまります。NEWFILES パラメータが指定されている場合、PASSWORD パラメータは、新規ファイル番号に関連するパスワードを指定する必要があります。

複数のパスワード保護ファイルをリストアする場合、FILES リストの保護されたファイル番号の位置に対応する位置決め値として正しいパスワードを指定しなければなりません。PASSWORD パラメータの詳細については、例を参照してください。パスワード保護ファイルを上書きするときは、Adabas ニュークリアスをアクティブにしなければなりません。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

---

### 例 1

```
ADASAV RESTORE FILES=3,4,5,OVERWRITE,  
ADASAV PASSWORD='PWD3,,PWD5'
```

ファイル 3、4、5 をリストアします。既存のファイル 3、4、5 は、リストアされたファイルによって上書きされます。ファイル 3 と 5 にはパスワード PWD3 と PWD5 が設定されています。

### 例 2

```
ADASAV RESTORE FILES=11,12,13,14,OVERWRITE  
ADASAV NEWFILES=16,0,17
```

ファイル 11、12、13、14 をリストアします。ファイル 11 と 13 はファイル 16、17 としてリストアされます。ファイル 12 と 14 は、対応する NEWFILES パラメータ値が 0 または指定されていないので、変更されません。ファイル 12、14、16、17 は（すでにデータベースに存在している場合）上書きされます。

# 176 RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリストア

---

▪ 条件 .....	998
▪ 結果 .....	999
▪ 構文 .....	999
▪ オプションパラメータ .....	1000
▪ 例 .....	1006

## RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリストア

---

RESTORE FMOVE 機能では、Adabas ニュークリアスがアクティブでないときに作成されたファイルまたはデータベースの SAVE データセット、または UTYPE=EXU で作成されたファイルの SAVE データセットから、ファイルをリストアします。1 つ以上のファイルをリストアできます。ファイルは既存のデータベースのフリースペースにリストアされます。エクステントサイズを変更できます。



### Notes:

1. 中断した RESTORE FMOVE 処理は最初から再実行する必要があります。処理を正常に終了するかまたはリストア処理を再実行するまで、リストア対象のファイルにはアクセスできません。
2. Adabas バージョン 5 からのチェックポイントファイルおよびセキュリティファイルはリストアできません。

このchapterでは、次のトピックについて説明します。

## 条件

---

▶ **手順 176.1. RESTORE FMOVE 機能を使用する場合は、次の条件が満たされている必要があります。**

- 1 正しい SAVE データセットを指定しなければなりません。指定するデータセットは、データベースまたはファイルの SAVE データセットで、リストアするファイルが含まれていなければなりません。Adabas バージョン 5.1 以上の SAVE データセットを使用できます。
- 2 データベースはすでに存在している必要があります。リストアするファイルは、そのデータベースからでも、または異なるデータベースから作成されたものでもかまいません。
- 3 FMOVE のファイルリストは、新規 RABN（およびサイズ）を使用してリストアするファイルを指定します。RABN は、使用された元の RABN と同じデバイスタイプに割り当てなければなりません。デバイスタイプが同じであれば、元のデータベースと異なるデータベースにファイルをリストアできます。
- 4 リストアするファイルに対して、フリースペース（フリースペーステーブルに従って使用できる）または上書きされるファイルが使用するスペースは、データベースで十分に利用できる大きさを必要とします。
- 5 Adabas ニュークリアスは出力データベースでアクティブであっても、アクティブでなくてもかまいません。

Adabas ニュークリアスがチェックポイントまたはセキュリティファイルをリストアするためにアクティブになっている場合、ADASAV ユーティリティに排他的なデータベース制御が必要になります。つまり、データベース上にアクティブなユーザーが存在しない必要があります。

- 6 SAVE 処理が DRIVES パラメータを指定して実行された場合、作成された SAVE データセットのリストアでも DRIVES パラメータを指定できます。そのとき、リストア処理は異なる SAVE データセットと並行して実行されます。あるいは、その SAVE データセットを、DRIVES パラメータなしでリストアされる 1 つの SAVE データセットに連結することができます。
- 7 マルチボリュームデータベースの SAVE データセットからほんのわずかのファイルをリストアする場合、リストアするファイルのデータを実際を含むテープボリュームだけを ADASAV ジョブ制御に指定する必要があります。対応する SYNВ チェックポイントと同様に SAVE 処理のジョブプロトコルは各ボリュームに含まれるファイルまたはファイル群の一部を示します。
- 8 拡張ファイルおよびカップリングされたファイルのリストアまたは上書きは全体でしかできません。つまり、拡張ファイルの 1 ファイルを指定するときは、拡張ファイルの他のファイルもすべて指定しなければならないということです。カップリングの関係にある一方のファイルを指定する場合は、他方のファイルもすべて指定しなければなりません。
- 9 チェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルの上書きは、それぞれ、別のチェックポイント、セキュリティ、トリガ、またはユーザー定義のシステムファイルのみの場合に可能です。このようなファイルがすでに異なるファイル番号で存在している場合、チェックポイント、セキュリティ、トリガの各ファイルはリストアできません。
- 10 NEWFILES パラメータでリストアするファイルに新規ファイル番号を割り当てることができます。

## 結果

---

この機能を実行すると、指定したファイルの内容が ADASAV SAVE 処理時と同じになります。ただし、同じデータベースブロック内である必要はありません。

## 構文

---

FMOVE のファイルリストは、新規 RABN（およびサイズ）を使用してリストアするファイルを指定します。

Adabas 拡張ファイルの場合、アンカーファイルを含む拡張ファイルの全コンポーネントファイルを指定する必要があります。指定されたファイルが他のファイルとカップリングされている場合、カップリングされている他方のファイルも指定しなければなりません。

RABN は、使用された元の RABN と同じデバイスタイプに割り当てなければなりません。デバイスタイプが同じであれば、元のデータベースと異なるデータベースにファイルをリストアできます。

```
ADASAV RESTORE FMOVE = file-list [ACRABN = AC-start-rabn-list ]
[AC2RABN = AC2-start-rabn-list ]
[ALLOCATION = { FORCE | NOFORCE } ]
[ASSOVOLUME = 'Associator-extent-volume' ]
[BUFNO = { number-of-buffers | 1 } ]
[DATAVOLUME = 'Data-Storage-extent-volume' ]
[DRIVES = { count | 1 } ]
[DSRABN = DS-start-rabn-list ]
[DSSIZE = DS-size-list ]
[EXCLUDE = file-list ]
[MAXISN = isn-count-list ]
[MAXISN2 = isn-count-list ]
[NEWFILES = file-list ]
[NIRABN = Ni-start-rabn-list ]
[NISIZE = Ni-size-list ]
[NOUSERABEND]
[OVERWRITE]
[PASSWORD = 'password-list' ]
[TEST]
[UIRABN = Ui-start-rabn-list ]
[UISIZE = Ui-size-list ]
```

## オプションパラメータ

### ACRABN：アドレスコンバータ開始 RABN/RABN リスト

ACRABN は、FMOVE によって指定された各ファイルに対するアドレスコンバータの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して ACRABN を省略すると、アドレスコンバータの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、ACRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに ACRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

### AC2RABN：セカンダリアドレスコンバータ開始 RABN/RABN リスト

AC2RABN は、FMOVE によって指定された各ファイルに対するセカンダリアドレスコンバータの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して AC2RABN を省略すると、セカンダリアドレスコンバータの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選

扱します。ファイルにセカンダリアドレスコンバータエクステン트가含まれていない場合、このパラメータは無視されます。

複数のファイルをリストアするとき、AC2RABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに AC2RABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。

**ALLOCATION**：ファイルエクステンツ割り当てができない場合の動作

ALLOCATION は、配置パラメータ AC2RABN、DSRABN、NIRABN または UIRABN に従ってファイルエクステンツ割り当てが得られない場合取るべき動作を指定します。

デフォルト（すなわち ALLOCATION=FORCE）の場合は、RABN 配置パラメータに従ったファイルエクステンツ割り当てができないと、このユーティリティはエラーで終了します。

ALLOCATION=NOFORCE の場合は、配置パラメータによる割り当てができないと、配置パラメータなしで再度割り当てようとします。

**ASSOVOLUME**：アソシエータエクステンツボリューム

 **Note:** ASSOVOLUME 値は単一引用符でくくらなければなりません。

ASSOVOLUME は、ファイルのアソシエータスペース（AC、NI、UI エクステンツ）を割り当てるボリュームを指定します。リクエストしたブロック数が指定ボリュームになかった場合、ADASAV は ASSOVOLUME パラメータに関係なく割り当てようとします。

AC2RABN、UIRABN、NIRABN が指定された場合、ADASAV は、対応するエクステンツタイプを割り当てるとき、ASSOVOLUME の値を無視します。ASSOVOLUME を指定しないと、ファイルのアソシエータスペースは、ADASAV のデフォルト割り当てルールに従って割り当てられます。

複数ファイルをリストアする場合、ASSOVOLUME パラメータのボリュームリストは、FMOVE パラメータのファイルリストに対応させなければなりません。あるファイルにボリュームを指定しない場合、ボリュームリスト内の該当エントリは空のままにする必要があります。59 ページの例を参照してください。

**BUFNO**：バッファカウント

BUFNO 値は、RESTORE 処理の固定バッファを割り当てます。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはなく、大量のスペースを割り当てることになります。デフォルトは 1（1 ドライブ当たり 1 バッファ）です。

**DATAVOLUME**：データストレージエクステンツボリューム

 **Note:** DATAVOLUME 値は単一引用符でくくらなければなりません。

DATAVOLUME は、ファイルのデータストレージスペース（DS エクステンツ）が割り当てられるボリュームを指定します。DSSIZE でリクエストしたブロック数が指定ボリュームになかった場合、ADASAV は DATAVOLUME パラメータに関係なく割り当てようとします。

## RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリストア

---

DSRABN を指定した場合、関連ファイルに対する DATAVOLUME は無視されます。DATAVOLUME を指定しないと、データストレージスペースは、ADASAV のデフォルト割り当てルールに従って割り当てられます。

複数ファイルをリストアする場合、DATAVOLUME パラメータのボリュームリストは、FMOVE パラメータのファイルリストに対応させなければなりません。あるファイルにボリュームを指定しない場合、ボリュームリスト内の該当エントリは空のままにする必要があります。例を参照してください。

### DRIVES：並行リストア用のテープドライブ

ADASAV では、複数のセーブデータセットボリュームのファイルを、データベースの元の RABN とは異なる RABN に並行してリストアできます。DRIVES は並行するリストア処理で使用するテープドライブの数を指定します。1 から 8 までの数が指定可能であり、デフォルトは 1 です。

### DSRABN：データストレージ開始 RABN/RABN リスト

DSRABN は、FMOVE で指定された各ファイルに対するデータストレージの開始 RABN を指定します。DSRABN は、FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して DSRABN を省略すると、ファイルのデータストレージの位置は、使用された元の RABN と同じデバイスタイプのデータストレージの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、DSRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに DSRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

### DSSIZE：新規データストレージサイズ

DSSIZE は、FMOVE で指定された各ファイルのデータストレージに割り当てられる新規サイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

サイズは、シリンダ数またはブロック数（数値に "B" を付加）で指定できます。最低でも、元のデータストレージと同じ大きさにする必要があります。

DSSIZE を省略すると、元のデータストレージサイズが使用されます。

複数ファイルをリストアする場合、DSSIZE パラメータのサイズリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルにサイズを指定しない場合、サイズリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

### EXCLUDE：指定したファイルをリストアから除外

EXCLUDE を指定する場合は、リストア処理から除外するファイル、つまりリストアされないファイルの番号をリストします。ファイル番号は 1 度だけリスト可能です。

番号を指定しない場合は、どのファイルも除外されません。

NEWFILES パラメータが

- 指定されていない場合、EXCLUDE パラメータに指定されたすべてのファイルが FMOVE パラメータにも指定されている必要があります。
- 指定されている場合、EXCLUDE パラメータに指定されたすべてのファイルが NEWFILES パラメータにも指定されている必要があります。この場合 EXCLUDE パラメータに指定されたファイル番号は、FMOVE パラメータに指定された古いファイル番号ではなく、NEWFILES の新しいファイル番号を参照します。

EXCLUDE パラメータは、Adabas Recovery Aid (ADARAI) によって構築されたりカバリジョブで使用するためのパラメータです。

#### MAXISN：新規最大 ISN

FMOVE で指定された各ファイルに割り当てる新規 ISN 数を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。値は、最低でも元の最大割り当て ISN (MAXISN) と同じ大きさにする必要があります。

MAXISN を省略すると、元の ISN 数が使用されます。

複数ファイルをリストアする場合、MAXISN パラメータの ISN 数のリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに ISN 数を指定しない場合、ISN 数のリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

#### MAXISN2：新規最大セカンダリ ISN

MAXISN は、ISN 内のセカンダリアドレスコンバータ (AC2) のサイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。セカンダリアドレスコンバータは、セカンダリスパンドレコードのセカンダリ ISN を、セカンダリレコードが格納されているデータストレージブロックの RABN にマッピングするために使用されます。

値は、最低でも元の最大割り当て ISN (MAXISN2) と同じ大きさにする必要があります。

MAXISN2 を省略すると、元の ISN 数が使用されます。ファイルにセカンダリアドレスコンバータエクステン트가含まれていない場合、このパラメータは無視されます。

複数のファイルをリストアする場合、MAXISN2 パラメータの ISN 数のリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに ISN 数を指定しない場合、ISN 数のリスト内の該当エントリには、0 を指定しなければなりません。

デバイスタイプが異なる複数のアソシエータエクステンツでデータベースが構成されている場合、MAXISN2 が指定されており、かつオンラインセーブ処理中にニュークリアスが追加のアドレスコンバータエクステンツを割り当てていると、ERROR-171 が発生することがあります。この場合は、エラーメッセージで示されているファイルの MAXISN2 パラメータを削除してから RESTONL FMOVE を再実行してください。

#### NEWFILES：新規ファイル番号

NEWFILES パラメータは、FMOVE で指定された各ファイルに割り当てる新規ファイル番号を指定します。このパラメータは、オプションです。ファイル番号を新しく割り当てる必要がない場合は、元のファイル番号が保持されます。NEWFILES は、拡張ファイル、物理的

## RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリストア

---

にカップリングされているファイル、またはレプリケートされたファイルには指定できません。

NEWFILES で指定したファイル番号がすでにデータベースに存在している場合、対応するファイルは、OVERWRITE パラメータも指定しないと、リストアされません。上書きされるファイルがパスワードで保護されている場合、対応する PASSWORD パラメータも指定する必要があります。

複数のファイルをリストアする場合、NEWFILES パラメータのファイル番号リストは、FMOVE パラメータのファイルリストに対応させなければなりません。あるファイルに新規ファイル番号を割り当てない場合、NEWFILES のファイル番号リストの該当ファイル番号のエントリには、0 を指定する必要があります。例を参照してください。

NEWFILES を使用して基本ファイルまたは LOB ファイルのファイル番号を再割り当てできるのは、LOB ファイルグループのファイルが両方ともリストアされる場合のみです。この場合、ADASAV は NEWFILES パラメータによって指定された新しいファイル番号を両ファイルに割り当て、2つのファイル間のリンクを適宜調整します。ただし、LOB ファイルグループの1ファイルのみをリストアした場合、NEWFILES パラメータを使用して新しいファイル番号を割り当ててはできません。代わりに ADADBS または AOS RENUMBER 機能を使用してください。

### NIRABN：ノーマルインデックスの開始 RABN/RABN リスト

NIRABN は、FMOVE で指定された各ファイルに対するノーマルインデックスの開始 RABN を指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

FMOVE を指定して NIRABN を省略すると、ノーマルインデックスの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、NIRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに NIRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

### NISIZE：新規ノーマルインデックスサイズ

NISIZE は、FMOVE で指定された各ファイルのノーマルインデックスに割り当てた新規サイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

サイズは、シリンダ数またはブロック数（数値に "B" を付加）で指定できます。最低でも、元のノーマルインデックスと同じ大きさにする必要があります。

NISIZE を省略すると、元のノーマルインデックスサイズが使用されます。

複数ファイルをリストアする場合、NISIZE パラメータのサイズリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルにサイズを指定しない場合、サイズリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**OVERWRITE**：既存ファイルの上書き

このパラメータが指定されると、既存ファイルの削除後にリストアが行われます。リストア対象のファイルがすでにデータベース内に存在している場合には、このパラメータが指定されていないと ADASAV はそのファイルをスキップします。

 **Note:** 不用意にデータベースを上書きしないために、OVERWRITE パラメータは、必ず FMOVE ファイルリストの前ではなく後ろに指定してください。

**PASSWORD**：Adabas Security のファイルパスワード

PASSWORD は、FILES または FMOVE ファイルリスト内のファイルがパスワード保護されている場合のパスワードまたはパスワードリストを指定します。これは、すでにデータベース内に存在していて上書きされるファイルに適用されます。NEWFILES パラメータが指定されている場合、PASSWORD パラメータは、新規ファイル番号に関連するパスワードを指定する必要があります。

複数のパスワード保護ファイルをリストアする場合は、正しいパスワードを指定する必要があります。このパスワードは、FILES リストまたは FMOVE リストの保護ファイル番号の位置に対応する、位置が固定されている値になります。PASSWORD パラメータの詳細については、[例](#)を参照してください。パスワード保護ファイルを上書きするときは、Adabas ニュークリアスをアクティブにしなければなりません。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**UIRABN**：アッパーインデックスの開始 RABN/RABN リスト

UIRABN は、FMOVE で指定された各ファイルに対するアッパーインデックスの開始 RABN を指定します。FMOVE パラメータと組み合わせただけ使用できます。

FMOVE を指定して UIRABN を省略すると、アッパーインデックスの位置は、使用された元の RABN と同じデバイスタイプのアソシエータの空きエリアから ADASAV が選択します。

複数ファイルをリストアするとき、UIRABN パラメータの RABN リストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルに UIRABN の値を指定しない場合、RABN リスト内の該当エントリには、0 を指定しなければなりません。[例](#)を参照してください。

## RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリストア

---

### UISIZE：新規アッパーインデックスサイズ

FMOVE で指定された各ファイルのアッパーインデックスに割り当てる新規サイズを指定します。FMOVE パラメータと組み合わせた場合にだけ使用できます。

サイズは、シリンダ数またはブロック数（数値に "B" を付加）で指定できます。最低でも、元のアッパーインデックスと同じ大きさにする必要があります。

UISIZE を省略すると、元のアッパーインデックスサイズが使用されます。

複数ファイルをリストアする場合、UISIZE パラメータのサイズリストは FMOVE パラメータのファイルリストに対応させる必要があります。あるファイルにサイズを指定しない場合、サイズリスト内の該当エントリには、0 を指定しなければなりません。例を参照してください。

## 例

---

### 例 1

```
ADASAV RESTORE FMOVE=4,6, ACRABN=0,3820,  
MAXISN=0,2000000
```

並行 RESTORE 処理に 3 つのテープドライブを使用します。ファイル 4 と 6 は新規 RABN でリストアされます。ファイル 4 のスペース割り当ては元のサイズで行われます。

ファイル 6 のアドレスコンバータはアソシエータ RABN 3820 で始まり、ファイルの MAXISN の値は 2,000,000 まで増やされます。

### 例 2

```
ADASAV RESTORE FMOVE=3,4,5, OVERWRITE  
ADASAV          PASSWORD='PWD3,,PWD5'
```

ファイル 3、4、5 をリストアします。これらのファイルがすでにデータベースに存在している場合は、上書きされます。ファイル 3 と 5 を上書きできるようにパスワードを指定しています。サイズにはすべて元の値が使用されます。ファイルが元の RABN 以外にリストアされる場合もあります。

### 例 3

```
ADASAV RESTORE  FMOVE=1,2  
ADASAV          FMOVE=3,4
```

ファイル 1~4 をリストアします。

### 例 4

## RESTORE FMOVE 機能：オフラインソースから別の RABN へのファイルのリストア

```
ADASAV RESTORE FMOVE=11,12,13,14,OVERWRITE
ADASAV          NEWFILES=16,0,17
```

ファイル 11、12、13、14 をリストアします。ファイル 11 と 13 はファイル 16、17 としてリストアされます。ファイル 12 と 14 は、対応する NEWFILES パラメータ値が 0 または指定されていないので、変更されません。ファイル 12、14、16、17 は（すでにデータベースに存在している場合）上書きされます。



# 177 RESTPLOG : プロテクションログのみのリスト ア

---

■ 基本パラメータ .....	1010
■ オプションパラメータ .....	1011
■ 例 .....	1011

RESTPLOGでは、すでにリストアされたデータベースまたはファイル（指定した場合）のPLOG内の変更をリストアします。RESTPLOGでは、関連するオンラインのSAVE（データベースまたはFILES）処理中に記録されたPLOGの変更のみをリストアします。

RESTPLOG機能は次のような場合に使用します。

1. SAVEデータセットがオンライン、つまりAdabasニュークリアスがアクティブな状態であるときに作成されている場合
2. オンラインSAVE時に作成された出力を使用してRESTONL機能を実行し、データベースまたはファイルをリストアし、SAVEテープからのデータベースまたはファイルのリストアを完了したが、PLOGに記録された更新が完了する前にエラーが発生したため終了した場合
3. RESTPLOG機能を実行して、PLOG内に記録されているリストア済みのデータベースまたはファイルにすべての更新を再度適用する場合これにより、再びRESTONLを使用してデータベースやファイル全体をリストアする必要がなくなります。

ALLOCATION=NOFORCEを指定したADASAV RESTONL FMOVEまたはADASAV RESTONL FILESを完了させる目的でRESTPLOGを使用することはできません。これらの処理は再起動する必要があります。

```
ADASAV RESTPLOG PLOGNUM = protection-log-number
                 { SYN1 | SYN4 } = starting-block-number
                 [FILES = file-list ]
                   [NEWFILES = file-list ]
                 [NOUSERABEND]
                 [TEST]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

**PLOGNUM**：プロテクションログ番号

PLOGNUMはリストアするプロテクションログの番号を指定します。

**SYN1|SYN4**：開始ブロック番号

SYN1またはSYN4は、リストア処理が開始されるSYN1またはSYN4チェックポイントがそれぞれ含まれるブロック番号を指定します。

## オプションパラメータ

### FILES：リストアするファイルのリスト

FILES パラメータは、中断された RESTONL FILES または RESTONL GCB でリストア中だったファイルを指定します。RESTPLOG では、中断された機能で指定されていたファイルと同じファイルを指定する必要があります。

RESTONL（データベース）実行が中断された場合は、FILES パラメータは省略する必要があります。この場合、RESTPLOG 機能はデータベースのすべてのファイルに対して実行されます。

### NEWFILES：新規ファイル番号

NEWFILES パラメータは、FILES パラメータにリストされた各ファイルに割り当てる新規ファイル番号を指定します。RESTPLOG で完了することになっている、中断された RESTONL FILE 機能に指定されていたものと同じ新規ファイル番号割り当てを指定する必要があります。

NEWFILES を指定しない場合、リストアされるファイルは元の番号を保持します。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

## 例

```
ADASAV RESTPLOG PLOGNUM=30,SYN1=150
```

プロテクションログ 30 のすべての更新内容を、その影響を受けた全ファイルに適用します。SYN1 チェックポイントが含まれるブロックは 150 です。



# 178

## SAVE : データベースのセーブ

---

▪ 構文 .....	1015
▪ オプションパラメータ .....	1015
▪ 例 .....	1017

ADASAV SAVE（データベース）機能は、データベースの内容をシーケンシャルデータセットにセーブします。データベースで使用されている全ブロックをセーブします。

SAVE（データベース）機能は、Adabas ニュークリアスがアクティブであっても、アクティブでなくても実行できます。実行しているときに、Adabas ニュークリアスの状態によって、次のようになります。

- ニュークリアスがアクティブな状態で実行した場合、後でデータベースをリストアするには、RESTONL 機能を使用する必要があります。
- ニュークリアスがアクティブでない状態で実行した場合、後でデータベースをリストアするには、RESTORE 機能を使用する必要があります。

どちらの場合も、SAVE データセットにセーブされたデータベースから1つだけまたはいくつかのファイルをリストアすることができます。

Adabas ニュークリアスがアクティブでない場合、SAVE 機能の実行中にニュークリアスを起動することはできません。また、セーブ中にセーブされているデータベースを変更するユーティリティ（ADALOD など）を実行することもできません。ニュークリアスセッションの自動再スタートがペンディングの場合、または別のオフラインユーティリティ（ADALOD や ADASAV など）を現在実行している場合は、SAVE 機能をオフラインで実行することはできません。

セーブ処理の実行中に Adabas ニュークリアスがアクティブな場合、ユーザーはセーブされているデータベースにフルアクセスできます。ユーザーは、読み込み、検索、更新、挿入、削除コマンドを発行できます。ただし、セーブするデータベースに変更を行うユーティリティ（ADALOD、ADAINV、ADADBSREFRESH など）をセーブ機能の実行中に実行してはいけません。また、開始することもできません。ニュークリアスがプロテクションログなしで稼働している場合、オンラインセーブ処理も実行できません。

オンラインセーブ処理では、ADASAV がセーブ処理を実行している間にセーブするデータベースを変更できます。したがって、Adabas ニュークリアスは、変更されたすべてのブロックもプロテクションログに書き込みます。このプロテクションログは後続のリストア処理（RESTONL 機能）に指定しなければなりません。

データベースのオンラインセーブ開始は SYN1 チェックポイントによってマークされます。オンラインセーブの終了時、ニュークリアスは現在アクティブなすべてのトランザクションの同期を取ります。これは、Adabas は、ET 状態のユーザーに余分な更新コマンドを発行しませんが、他のアクティブなユーザーを ET 状態になるまで継続させることを意味します。そのとき、この状態は SYN2 チェックポイントでマークされます。したがって、SYN2 チェックポイントは、処理中のトランザクションがない、データベースの整合性が保たれている状態をマークします。この状態は、後でデータベースまたはファイルを SAVE データセットからリストアするときに再現されます。

トランザクションの同期に必要な最大時間は、TTSYN パラメータで制限できます。

複数のディスクボリュームに存在するデータベースは、DRIVES パラメータが指定されると、複数の SAVE データセットに並行してセーブされます。この処理モードは、セーブの経過時間を

大きく縮小します。ドライブ番号の昇順に連結するとき、結果のSAVEデータセットは、DRIVESパラメータを指定しないで作成したSAVEデータセットに等しくなります。

SAVE（データベース）機能は、インバート、ロード、リフレッシュ、リオーダ、リストア状態のファイルをセーブしません。実際には、ファイルリストからこのようなファイルを削除し、メッセージADAU15を出力して、残りのファイルのセーブ処理を実行します。最後に、ADASAVはリターンコード4で終了します。

リカバリエイド（RLOG）オプションがアクティブな場合は、SAVE（データベース）機能が新しいRLOG生成を開始します。

このchapterでは、次のトピックについて説明します。

## 構文

```
ADASAV SAVE [BUFNO = {number-of-buffers | 1}]
            [DRIVES = {count | 1}]
            [INCREMENTAL]
            [NOUSERABEND]
            [PERDRIVE = disk-drive-per-tape-drive , ...]
            [TTSYN = {seconds | ADARUN-tt}]
            [TWO COPIES]
            [TEST]
```

## オプションパラメータ

### BUFNO：バッファカウント

BUFNO 値は、SAVE 処理用の固定バッファを割り当てます。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することなく、大量のスペースを割り当てることとなります。デフォルトは 1（1 ドライブ当たり 1 バッファ）です。

### DRIVES：SAVE と並行処理するテープドライブ

DRIVES は並行する SAVE 処理で使用されるシーケンシャル出力データセット（通常はテープドライブ上）の数です。最大 8 までの数を指定できます。デフォルトは 1 です。

**INCREMENTAL**：変更ファイルのみ保存

INCREMENTAL は、最後の ADASAV SAVE 処理以降に変更されたファイルのみをセーブします。INCREMENTAL が指定されない場合、SAVE 機能によってすべてのデータベースファイルがセーブされます。

**NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

**PERDRIVE**：テープドライブ当たりのディスクドライブ

PERDRIVE は、単一の DRIVES テープドライブに割り当てられるディスクドライブの数を指定します。例えば、データベースが7つのディスクドライブ上に存在し、SAVE 処理に3つのテープドライブを使用できる場合は、PERDRIVE=3,2,2 と指定すると、最初の3つのディスクドライブがテープドライブ 1 に書き込まれ、次の2つのディスクドライブがテープドライブ 2 に書き込まれ、次の2つのディスクドライブがテープドライブ 3 に書き込まれます。ドライブの順序は、このマニュアルの最後に説明しているように、DD/SAVE<sub>n</sub> および DD/DUAL<sub>n</sub> のジョブ制御指定に一致します。

PERDRIVE で指定されたドライブの合計数は、すべてのアソシエータ (ASSO) ディスクと DATA ディスクの合計数と等しくする必要があります。ASSO と DATA が同じ1つのディスク上にある場合は、2つの別個のディスクとして計算されます。DRIVES パラメータが使用され、PERDRIVE パラメータが省略された場合には、ADASAV によって最も効率的なテープドライブ使用が決定されます。

**TEST**：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

**TTSYN**：SYN2 チェックポイント制御

このパラメータを使用すると、現在の ADASAV 処理の同期チェックポイント処理時におけるニュークリアスの ADARUNTT（最大トランザクションタイム）を減少させることができます。指定値は、秒単位でのおおよその時間（TT \* 1.05 秒）であり、ニュークリアスの ADARUN TT 値未満でなければなりません。TTSYN が省略されるか、ニュークリアスの TT 値よりも大きい場合には、この TT 値がデフォルトとなります。

ADASAV SAVE 機能実行時に Adabas ニュークリアスがアクティブな場合、同期チェックポイント SYN2 は、SAVE 処理の終了時に取得されます。これはすべてのユーザーが確実に ET 状態になる時間です。ユーザーが1人でも ET 状態になっていない場合は、それ以外のユーザーに対する新たなトランザクションを開始できず、他のユーザーはこの SYN2 チェックポイントを取得できるようになるまで待機する必要があります。

ADARUN TT 値は、論理トランザクションに対する許容最大経過時間を制御します。SYN2 チェックポイントの処理が可能になるまでの最大待機時間です。ADASAV SAVE TTSYN パ

ラメータを使用した場合、同期チェックポイント処理の間だけ TT 値を減少させることができます。本来の TT 値は、ADASAV による SAVE 処理が終了すると、再び効力を持つようになります。

**TWOCOPIES**：2つの出力コピーの作成

TWOCOPIES は、ADASAV 出力の物理コピーを 2 つ作成します。

## 例

```
ADASAV SAVE DRIVES=4
```

4つのテープドライブを並行使用して SAVE 機能を実行します。



# 179

## SAVE FILES：指定ファイルのセーブ

---

▪ 構文 .....	1021
▪ オプションパラメータ .....	1022
▪ 例 .....	1024

ADASAV SAVE FILES 機能は、1つ以上のファイルの内容をシーケンシャルデータセットにセーブします。ファイルに使用されている全ブロックがセーブされます。

SAVE FILES 機能は、Adabas ニュークリアスがアクティブであっても、アクティブでなくても実行できます。実行しているときに、Adabas ニュークリアスの状態によって、次のようになります。

- ニュークリアスがアクティブな状態で実行した場合、後でファイルをリストアするには、RESTONL 機能を使用する必要があります。
- ニュークリアスがアクティブでない状態で実行した場合、後でファイルをリストアするには、RESTORE 機能を使用する必要があります。

どちらの場合も、SAVE データセットにセーブされたすべてのファイルから1つのみ、またはいくつかのファイルをリストアすることができます。

Adabas ニュークリアスがアクティブでない場合、SAVE 機能の実行中にニュークリアスを起動することはできません。また、セーブ中にセーブされているファイルを変更するユーティリティ (ADALOD など) を実行することもできません。ニュークリアスセッションの自動再スタートがペンディングの場合、またはセーブされるファイルで別のオフラインユーティリティ (ADALOD や ADASAV など) を現在実行している場合は、SAVE 機能をオフラインで実行することはできません。

セーブ処理の実行中に Adabas ニュークリアスがアクティブな状態にいる場合、ユーザーはセーブされているファイルにフルアクセスできます。ユーザーは、読み込み、検索、更新、挿入、削除コマンドを発行できます。ただし、セーブするファイルを変更するユーティリティ (ADALOD、ADAINV、ADADBSREFRESH など) をセーブ機能の実行中に実行しないでください。また、これらのユーティリティを開始することもできません。ニュークリアスがプロテクションログなしで稼動している場合、オンラインセーブ処理も実行できません。

オンラインセーブ処理では、ADASAV がセーブ処理を実行している間にセーブするファイルを変更できます。したがって、Adabas ニュークリアスは、ファイルの変更されたすべてのブロックをプロテクションログに書き込みます。このプロテクションログは後続のリストア処理 (RESTONL 機能) に指定しなければなりません。

ファイルのオンラインセーブ開始はSYN4チェックポイントでマークされます。オンラインセーブの終了時、ニュークリアスは現在アクティブなすべてのトランザクションの同期を取ります。これは、Adabas は、ET 状態のユーザーに余分な更新コマンドを発行しませんが、他のアクティブなユーザーを ET 状態になるまで継続させることを意味します。このとき、この状態は SYN5 チェックポイントでマークされます。したがって、SYN5 チェックポイントは、処理中のトランザクションがない、データベースの整合性が保たれている状態をマークします。この状態は、後でファイルを SAVE データセットからリストアするときに再現されます。

トランザクションの同期に必要な最大時間は、TTSYN パラメータで制限できます。

パラメータ UTYPE=EXU が指定され、Adabas ニュークリアスがアクティブな場合、セーブ処理はオフラインセーブと同様に実行されます。ADASAV は、同時更新に対して EXU オープンで

セーブされるすべてのファイルをロックします。このファイルを後でリストアするには、RESTONL 機能ではなく RESTORE 機能を使用する必要があります。

複数のオフラインセーブ（ファイル）処理、または UTYPE=EXU でのファイルのセーブは異なるファイルに並行して実行することができます。一度にアクティブにできるオンラインセーブ（ファイル）処理は 1 つのみです。

複数のディスクボリュームに存在するデータベースのファイルは、DRIVES パラメータが指定されたとき、複数の SAVE データセットに並行してセーブされます。この処理モードは、セーブの経過時間を大きく縮小します。ドライブ番号の昇順に連結するとき、結果の SAVE データセットは、DRIVES パラメータを指定しないで作成した SAVE データセットに等しくなります。

SAVE FILES 機能では、インバート、ロード、リフレッシュ、リオーダ、リストア状態のファイルをセーブしません。実際には、ファイルリストからこのようなファイルを削除し、メッセージ ADAU15 を出力して、残りのファイルのセーブ処理を実行します。最後に、ADASAV はリターンコード 4 で終了します。

このchapterでは、次のトピックについて説明します。

## 構文

```
ADASAV SAVE FILES = file-list [BUFNO = {number-of-buffers | 1}]
[DRIVES = {count | 1}]
[INCREMENTAL]
[NOUSERABEND]
[PASSWORD = 'password-list']
[PERDRIVE = disk-drive-per-tape-drive , ...]
[TEST]
[TTSYN = {seconds | ADARUN-tt}]
[TWOCOPIES]
[UTYPE = EXU]
```

FILES ファイルリストには、セーブするファイルを指定します。指定されたファイルが他のファイルにカップリングされているか、または拡張ファイルのコンポーネントである場合、ADASAV SAVE FILES 機能は、条件に応じて、次のように実行されます。

- ニュークリアスがアクティブである場合は、拡張ファイルのすべてのコンポーネントファイルまたはファイルにカップリングされているすべてのファイルが FILES ファイルリストに指定されているときにのみ、ADASAV は SAVE FILES 処理を受け入れます。

- ニュークリアスがアクティブでない場合は、拡張ファイルのコンポーネントファイルの1つまたはカップリングされているファイルの1つが指定されているとき、SAVE FILES 処理が可能になります。この後、ADASAV はファイルリストを自動的に拡張します。

## オプションパラメータ

---

### BUFNO：バッファカウント

BUFNO 値は、SAVE 処理用の固定バッファを割り当てます。通常、2 または 3 の値で最適なパフォーマンスが得られ、255 まで指定可能です。ただし、5 を超える値を指定してもパフォーマンスが大きく向上することはない、大量のスペースを割り当てることになり、デフォルトは 1 (1 ドライブあたり 1 バッファ) です。

### DRIVES：SAVE と並行処理するテープドライブ

DRIVES は、並行する SAVE 処理で使用するテープドライブの数を指定します。最大 8 つのドライブを指定できます。デフォルトは 1 です。

### INCREMENTAL：変更ファイルのみ保存

INCREMENTAL は、最後の ADASAV SAVE 処理以降に変更された FILES リスト内のファイルのみをセーブします。INCREMENTAL が指定されない場合、SAVE 機能によって FILES リストのすべてのファイルがセーブされます。

### NOUSERABEND：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34 (ダンプあり) またはユーザーアベンド 35 (ダンプなし) で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION" (エラーのためユーティリティを終了します) を表示してコンディションコード 20 で終了します。

### PASSWORD：ファイルパスワード

PASSWORD は、FILES ファイルリスト内の 1 つ以上のファイルがパスワード保護されている場合のパスワードまたはパスワードリストを指定します。パスワード保護ファイルが複数ある場合は、正しいパスワードを指定する必要があります。このパスワードは、FILES リストの保護ファイル番号の位置に対応する、位置が固定されている値になります。PASSWORD パラメータの詳細については、このセクションの最後にある例を参照してください。パスワード保護ファイルをセーブするときは、Adabas ニュークリアスをアクティブにする必要があります。

### PERDRIVE：テープドライブ当たりのディスクドライブ

PERDRIVE は、1 つのテープドライブに対して割り当てるディスクドライブの数を指定します。例えば、データベースが 7 つのディスクドライブ上に存在し、SAVE 処理に 3 つのテープドライブを使用できる場合は、PERDRIVE=3,2,2 と指定すると、最初の 3 つのディスクドライブがテープドライブ 1 に書き込まれ、次の 2 つのディスクドライブがテープドライブ 2 に書き込まれ、次の 2 つのディスクドライブがテープドライブ 3 に書き込まれます。ドライ

ブの順序は、このマニュアルの最後で説明しているように、DDSAVEN/DDDUALn または SAVEN/DUALn のジョブ制御指定に一致します。

PERDRIVE で指定されたドライブの合計数は、すべてのアソシエータ (ASSO) ディスクおよび DATA ディスクの合計数と等しくなければならず、ASSO および DATA が同じ 1 つのディスク上にある場合には、2 つの別個のディスクとして計算されます。DRIVES パラメータが使用され、PERDRIVE パラメータが省略された場合には、ADASAV によって最も効率的なテープドライブ使用が決定されます。

#### TEST：構文テスト

TEST パラメータは、処理を実際には行わずに構文のみをテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

#### TTSYN：SYN5 チェックポイント制御

このパラメータを使用すると、現在の ADASAV 処理の同期チェックポイント処理時におけるニュークリアスの ADARUNTT (最大トランザクションタイム) を減少させることができます。指定値は、秒単位でのおおよその時間 (TT \* 1.05 秒) であり、ニュークリアスの ADARUN TT 値未満でなければなりません。TTSYN が省略されるか、ニュークリアスの TT 値よりも大きい場合には、ニュークリアスの TT 値がデフォルトとなります。



**Note:** UTYPE=EXU が指定されると TTSYN は効力を失います。

ADASAV SAVE 実行時に Adabas ニュークリアスがアクティブ状態の場合、同期チェックポイント SYN5 は、この SAVE 処理の終了時に取得されます。これはすべてのユーザーが確実に ET 状態になる時間です。ユーザーが 1 人でも ET 状態になっていない場合は、それ以外のユーザーに対する新たなトランザクションを開始できず、他のユーザーはこの SYN5 チェックポイントを取得できるようになるまで待機する必要があります。

ADARUN TT 値は、論理トランザクションに対する許容最大経過時間を制御します。SYN5 チェックポイントの処理が可能になるまでの最大待機時間です。ADASAV SAVE TTSYN パラメータを使用した場合、同期チェックポイント処理の間だけ TT 値を減少させることができます。本来の TT 値は、ADASAV による SAVE 処理が終了すると、再び効力を持つようになります。

#### TWOCOPIES：2 つの出力コピーの作成

TWOCOPIES は、ADASAV 出力の物理コピーを 2 つ作成します。

#### UTYPE=EXU：オープンするユーザータイプ

ADASAV では、レコードバッファ "EXU=file-list" を指定した Adabas オープンコマンドが発行されます。これにより、Adabas ニュークリアスがアクティブでかつプロテクションログを使用していない場合でもファイルセーブが可能となります。SAVE 機能の処理中は、セーブされているファイルに対する更新は許可されません。対応する RESTORE ファイル処理ではプロテクションログを必要としません。

## 例

---

### 例 1

```
ADASAV SAVE FILES=10,15
```

ファイル 10 およびファイル 15 をセーブします。

### 例 2

```
ADASAV SAVE FILES=3,4,5,  
ADASAV          PASSWORD='PWD3,,PWD5'
```

ファイル 3、4、5 をセーブします。ファイル 3 およびファイル 5 はパスワード保護されており、そのパスワードはそれぞれ PWD3 および PWD5 です。

# 180

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	1026
▪ z/OS .....	1032
▪ z/VM .....	1037
▪ VSE .....	1040

このセクションでは、ADASAVをBS2000、z/OS、z/VM、およびVSEの各システムで実行するために必要なジョブ制御情報について説明し、システムごとのジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	Adabasニュークリアスがアクティブでないときのみ必要
マルチプロテクションログ	DDPLOGRn	ディスク	RESTORE および RESTONL データベース/GCBではオプション
マルチコマンドログ	DDCLOGRn	ディスク	RESTORE および RESTONL データベース/GCBではオプション
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAIの場合に必要
バックアップコピー	DDSAVE1~8	テープ/ディスク	SAVE 機能にのみ必要
デュアルコピー	DDDUAL1~8	テープ/ディスク	SAVE 機能でバックアップを2部 作成する場合のみ必要
バックアップコピー (RESTORE 機能の入力)	DDREST1~8	テープ/ディスク	RESTORE および RESTONL 機能に必要
シーケンシャルプロテクションログ	DDPLOG	テープ/ディスク	RESTONL および RESTPLOG 機能に必要
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADASAV パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/DDPRINT		メッセージおよびコード
ADASAV メッセージ	SYSLST/ DDDRUCK		メッセージおよびコード

 **Note:** RESTONL では、入力 SAVE テープとシーケンシャルプロテクションログは、DDREST1 を使用して連結できます。

## ADASAV JCL の例 (BS2000)

### ファイルのセーブ、データベースのセーブ

SDF フォーマットの場合：

```

/.ADASAV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A S A V SAVE FILES, SAVE DATABASE
/REMARK *
/DELETE-FILE ADAyyyyyy.SAVE1
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.SAVE1,SUP=TAPE(DEVICE=TAPE-C1,VOL=SAV101),-
/  PROT=(USER-ACCESS=ALL-USERS)
/SET-JOB-STEP
/DELETE-FILE ADAyyyyyy.DUAL1
/SET-JOB-STEP
/CREATE-FILE ADAyyyyyy.DUAL1,SUP=TAPE(DEVICE=TAPE-C1,VOL=SAV101),-
/  PROT=(USER-ACCESS=ALL-USERS)
/SET-JOB-STEP
/ASS-SYSLST L.SAV.SAVE
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAYyyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDSSAVE1,ADAYyyyyy.SAVE1,TAPE(FILE-SEQ=1)
/SET-FILE-LINK DDSDUAL1,ADAYyyyyy.DUAL1,TAPE(FILE-SEQ=1)
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV SAVE TWOCOPIES
/LOGOFF SYS-OUTPUT=DEL

```

### ファイルのセーブ、データベースのセーブ (続き)

ISP フォーマットの場合：

```

/.ADASAV LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A S A V SAVE FILES, SAVE DATABASE
/REMARK *

/SYSFILE SYSLST=L.SAV.SAVE
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAYyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAYyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES

```

```
/FILE ADAyyyyyy.WORK ,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyyy.SAVE1 ,LINK=DDSAVE1 ,DEVICE=TAPE-C1,VOLUME=SAV101
/FILE ADAyyyyyy.DUAL1 ,LINK=DDDUAL1 ,DEVICE=TAPE-C1,VOLUME=SAV201
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV SAVE TWOCOPIES
/LOGOFF NOSPOOL
```

### オンラインで作成された SAVE データセットからのファイルのリストア

SDF フォーマットの場合：

```
/.ADASAV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A S A V RESTORE FILES, RESTORE DATABASE
/REMARK * FROM ONLINE CREATED SAVE DATASETS
/REMARK *
/ASS-SYSLST L.SAV.REON
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrS.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAyyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAyyyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR1,ADAyyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADAyyyyyy.PLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDREST1,ADAyyyyyy.SAVE1
/SET-FILE-LINK DDPLOG,ADAyyyyyy.PLOG
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV RESTONL FILES=2,PLOGNUM=ppp,SYN1=43
/LOGOFF SYS-OUTPUT=DEL
```

### オンラインで作成された SAVE データセットからのファイルのリストア (続き)

ISP フォーマットの場合：

```
/.ADASAV LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A S A V RESTORE FILES, RESTORE DATABASE
/REMARK * FROM ONLINE CREATED SAVE DATASETS
/REMARK *
/SYSFILE SYSLST=L.SAV.REON
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
```

```

/FILE ADAyyyyyy.WORK ,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyyy.SAVE1 ,LINK=DDREST1
/FILE ADAyyyyyy.PLOG ,LINK=DDPLOG
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV RESTONL FILES=2,PLOGNUM=ppp,SYN1=43
/LOGOFF NOSPOOL

```

## データベースのリストア

SDF フォーマットの場合：

```

/.ADASAV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * ADASAV:
/REMARK * EXAMPLE HOW TO USE ADASAV TO RESTORE THE
/REMARK * ENTIRE DATABASE /REMARK *
/REMARK *
/DELETE-FILE ADAyyyyyy.SAVE1
/SET-JOB-STEP
/IMPORT-FILE
SUP=TAPE(F-NAME=ADAyyyyyy.SAVE1,DEV-TYPE=TAPE-C1,VOL=SAV101)
/SET-JOB-STEP
/ASS-SYSLST L.SAV.REST
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAyyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAyyyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR1,ADAyyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADAyyyyyy.PLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDREST1,ADAyyyyyy.SAVE1,TAPE(FILE-SEQ=1),ACC-METH=SAM,-
/ BUFF-LEN=32768,REC-FORM=V
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B

```

```
ADASAV RESTORE OVERWRITE
/LOGOFF SYS-OUTPUT=DEL
```

### データベースのリストア（続き）

ISP フォーマットの場合：

```
/.ADASAV LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * ADASAV:
/REMARK * EXAMPLE HOW TO USE ADASAV TO RESTORE THE
/REMARK * ENTIRE DATABASE
/REMARK *
/SYSFILE SYSLST=L.SAV.REST
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyy.WORK ,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyy.SAVE1 ,LINK=DDREST1,DEVICE=TAPE-C1,VOLUME=ADA001,-
/ STATE=FOREIGN.-
/ FCBTYPE=SAM,RECFORM=V,RECSIZE=,BLKSIZE=32768,LABEL=STD
/FILE ADAyyyyy.PLOGR1,LINK=DDPLOGR1,SHARUPD=YES
/FILE ADAyyyyy.PLOGR2,LINK=DDPLOGR2,SHARUPD=YES
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADASAV,DB=yyyyy, IDTNAME=ADABAS5B
ADASAV RESTORE OVERWRITE
/LOGOFF NOSPOOL
```

### RESTONL 機能中断後のプロテクションログのリストア

SDF フォーマットの場合：

```
/.ADASAV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A S A V RESTORE PROTECTION LOG
/REMARK *
/ASS-SYSLST L.SAV.REPL
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAYyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDPLOG,ADAYyyyy.PLOG
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADASAV,DB=yyyyy, IDTNAME=ADABAS5B
```

```
ADASAV RESTPLOG FILES=2,PLOGNUM=ppp,SYN1=43
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADASAV LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A S A V RESTORE PROTECTION LOG
/REMARK *
/SYSFILE SYSLST=L.SAV.REPL
/FILE ADA.MOD,LINK=DDLIB
/FILE ADAyyyyyy.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyyy.WORK,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyyy.PLOG,LINK=DDPLOG
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV RESTPLOG FILES=2,PLOGNUM=ppp,SYN1=43
/LOGOFF NOSPOOL
```

### オフラインで作成された SAVE データセットからのファイルのリストア

#### SDF フォーマットの場合：

```
/.ADASAV LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A S A V RESTORE FILES, RESTORE DATABASE
/REMARK * FROM OFFLINE CREATED SAVE DATASETS
/REMARK *
/ASS-SYSLST L.SAV.REFM
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyyy.DATA,SHARE-UPD=YES
/SET-FILE-LINK DDWORKR1,ADAYyyyyy.WORK,SHARE-UPD=YES
/SET-FILE-LINK DDREST1,ADAYyyyyy.SAVE1
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV RESTORE FMOVE=2
/LOGOFF SYS-OUTPUT=DEL
```

#### ISP フォーマットの場合：

```

/.ADASAV LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A S A V RESTORE FILES, RESTORE DATABASE
/REMARK * FROM OFFLINE CREATED SAVE DATASETS
/REMARK *
/SYSFILE SYSLST=L.SAV.REFM
/FILE ADA.MOD ,LINK=DDLIB
/FILE ADAyyyyyy.ASSO ,LINK=DDASSOR1,SHARUPD=YES
/FILE ADAyyyyyy.DATA ,LINK=DDDATAR1,SHARUPD=YES
/FILE ADAyyyyyy.WORK ,LINK=DDWORKR1,SHARUPD=YES
/FILE ADAyyyyyy.SAVE1 ,LINK=DDREST1
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADASAV,DB=yyyyyy,IDTNAME=ADABAS5B
ADASAV RESTORE FMOVE=2
/LOGOFF NOSPOOL
    
```

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	Adabas ニュークリアスがアクティブでないときのみ必要
マルチプロテクションログ	DDPLOGRn	ディスク	RESTORE および RESTONL データベース/GCB ではオプション
マルチコマンドログ	DDCLOGRn	ディスク	RESTORE および RESTONL データベース/GCB ではオプション
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI の場合に必要
バックアップコピー	DDSAVE1~8	テープ/ディスク	SAVE 機能にのみ必要
デュアルコピー	DDDUAL1~8	テープ/ディスク	SAVE 機能でバックアップを2部作成する場合のみ必要
バックアップコピー (RESTORE 機能の入力)	DDREST1~8	テープ/ディスク	RESTORE および RESTONL 機能に必要
シーケンシャルプロテクションログ	DDPLOG	テープ/ディスク	RESTONL および RESTPLOG 機能に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADASAV パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード

データセット	DD 名	ストレージ	追加情報
ADASAV メッセージ	DDDRUCK	プリンタ	メッセージおよびコード



**Note:** RESTONL では、入力 SAVE テープとシーケンシャルプロテクションログは、DDREST1 を使用して連結できます。

## ADASAV JCL の例 (z/OS)

### データベースのセーブ

```
//ADASAV JOB
//*
//* ADASAV:
//* EXAMPLE HOW TO USE ADASAV TO SAVE THE
//* ENTIRE DATABASE
//*
//SAVE EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDSAVE1 DD DSN=EXAMPLE.DByyyyy.DDSAVE1,UNIT=TAPE, <=== OUTPUT
// DISP=(,CATLG),VOL=SER=ADABCK
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADASAV,SVC=xxx,DE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADASAV SAVE
/*
```

この例については、MVSJOBS データセットの ADASAV を参照してください。

### データベースのセーブ（出力コピーを 2 つ作成）

```
//ADASAVT JOB
//*
//* ADASAV:
//* EXAMPLE HOW TO USE ADASAV TO SAVE THE
//* ENTIRE DATABASE CREATING TWO COPIES OF THE OUTPUT
//*
//SAVE EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
```

```
// *
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDSAVE1 DD DSN=EXAMPLE.DByyyyyy.DDSAVE1,UNIT=TAPE, <=== OUTPUT
//
// DISP=(,CATLG),VOL=SER=ADABCK
//DDDUAL1 DD DSN=EXAMPLE.DByyyyyy.DDSAVD1,UNIT=TAPE, <=== OUTPUT
//
// DISP=(,CATLG),VOL=SER=ADABCK1
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADASAV,SVC=xxx,DE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADASAV SAVE TWOCOPIES
/*
```

この例については、MVSJOBS データセットの ADASAVT を参照してください。

データベースのリストア

```
//ADASAVR JOB
// *
// * ADASAV:
// * EXAMPLE HOW TO USE ADASAV TO RESTORE THE
// * ENTIRE DATABASE
// *
//RESTORE EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
// *
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDPLOGR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1 <=== PLOG1
//DDPLOGR2 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2 <=== PLOG2
//DDREST1 DD DSN=EXAMPLE.DByyyyyy.SAVE, <=== SAVE OUTPUT
//
// DISP=OLD,UNIT=TAPE,VOL=SER=ADABCK
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X

//DDCARD DD *
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADASAV RESTORE OVERWRITE
```

/\*

この例については、MVSJOBS データセットの ADASAVR を参照してください。

オフラインで作成された **SAVE** データセットからのファイルのリストア

```
//ADASAVRF JOB
//*
//*   ADASAV:
//*   EXAMPLE HOW TO USE ADASAV TO RESTORE A FILE
//*   TO ANY RABNS FROM AN OFFLINE SAVE
//*
//RESTORE EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABASLOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDPLOGR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1 <=== PLOG1
//DDPLOGR2 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2 <=== PLOG2
//DDREST1 DD DSN=EXAMPLE.DByyyyyy.SAVE, <=== SAVE OUTPUT
// DISP=OLD,UNIT=TAPE,VOL=SER=ADABCK
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADASAV RESTORE FMOVE=2
/*
```

この例については、MVSJOBS データセットの ADASAVRF を参照してください。

オンラインで作成された **SAVE** データセットからのファイルのリストア

```
//ADASAVRO JOB
//*
//*   ADASAV:
//*   EXAMPLE HOW TO USE ADASAV TO RESTORE FILES
//*   FROM SAVE DATA SETS CREATED ONLINE
//*
//RESTORE EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
```

```
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDPLOGR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1 <=== PLOG1
//DDPLOGR2 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2 <=== PLOG2
//DDREST1 DD DSN=EXAMPLE.DByyyyyy.SAVE, <=== SAVE OUTPUT
// DISP=OLD,UNIT=TAPE,VOL=SER=ADABCK
//DDPLOG DD DSN=EXAMPLE.DByyyyyy.PLOG, <=== PLOG OUTPUT
// DISP=OLD,UNIT=TAPE,VOL=SER=PLOGD1
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADASAV RESTONL FILES=2,PLOGNUM=ppp,SYN1=1
/*
```

この例については、MVSJOBS データセットの ADASAVRO を参照してください。

### RESTONL 機能中断後のプロテクションログのリストア

```
//ADASAVRP JOB
/*
/* ADASAV:
/* EXAMPLE HOW TO USE ADASAV TO RESTORE THE
/* PROTECTION LOG AFTER AN INTERRUPTED RESTONL
/*
//RESTORE EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDPLOGR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1 <=== PLOG1
//DDPLOGR2 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2 <=== PLOG2
//DDPLOG DD DSN=EXAMPLE.DByyyyyy.PLOG, <=== PLOG INPUT
// DISP=OLD,UNIT=TAPE,VOL=SER=PLOGD1
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE DD *
ADASAV RESTPLOG PLOGNUM=ppp,SYN1=1
```

/\*

この例については、MVSJOBS データセットの ADASAVRP を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	
データストレージ	DDDATARn	ディスク	
ワーク	DDWORKR1 DDWORKR4	ディスク	Adabas ニュークリアスがアクティブでないときのみ必要
マルチプロテクションログ	DDPLOGRn	ディスク	RESTORE および RESTONL データベース/GCB ではオプション
マルチコマンドログ	DDCLOGRn	ディスク	RESTORE および RESTONL データベース/GCB ではオプション
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI の場合に必要
バックアップコピー	DDSAVE1~8	テープ/ディスク	SAVE 機能にのみ必要
デュアルコピー	DDDUAL1~8	テープ/ディスク	SAVE 機能でバックアップを2部作成する場合のみ必要
バックアップコピー (RESTORE 機能の入力)	DDREST1~8	テープ/ディスク	RESTORE および RESTONL 機能に必要
シーケンシャルプロテクションログ	DDPLOG	テープ/ディスク	RESTONL および RESTPLOG 機能に必要
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADASAV パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADASAV メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード



**Note:** RESTONL では、入力 SAVE テープとシーケンシャルプロテクションログは、DDREST1 を使用して連結できます。

## ADASAV ジョブ制御の例 (z/VM)

### データベースのセーブ

```

DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1
DATADEF DDWORKR1,DSN=ADABASVv.WORK,VOL=WORKV1
DATADEF DDSAVE1,DSN=ADABASVv.COPY1,MODE=A
DATADEF DDDUAL1,DSN=ADABASVv.COPY2,MODE=A
DATADEF DDPRINT,DSN=ADASAV.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADASAV.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNSAV.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADASAV.CONTROL,MODE=A
ADARUN
    
```

### RUNSAV CONTROL A1 の内容

```

ADARUN PROG=ADASAV,DEVICE=dddd,DB=yyyyyy
    
```

### ADASAV CONTROL A1 の内容

```

ADASAV SAVE TWOCOPIES
    
```

### オンラインで作成された **SAVE** データセットからのファイルのリストア

```

DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1
DATADEF DDWORKR1,DSN=ADABASVv.WORK,VOL=WORKV1
DATADEF DDREST1,DSN=ADABASVv.SAVE,MODE=A
DATADEF DDPL0G,DSN=ADABASVv.SIBA,MODE=A
DATADEF DDPRINT,DSN=ADASAV.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADASAV.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNSAV.CONTROL,MODE=A
DATADEF DDKARTE,DSN=RESTONL.CONTROL,MODE=A
    
```

```
ADARUN
```

### *RUNSAV CONTROL A1 の内容*

```
ADARUN  PROG=ADASAV,DEVICE=dddd,DB=yyyyyy
```

### *RESTONL CONTROL A1 の内容*

```
ADASAV  RESTONL  FILES=2,PLOGNUM=ppp,SYN1=1
```

### **RESTONL** 機能中断後のプロテクションログのリストア

```
DATADef  DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1  
DATADef  DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1  
DATADef  DDWORKR1,DSN=ADABASVv.WORK,VOL=WORKV1  
DATADef  DDPLOG,DSN=ADABASVv.PLOG,MODE=A  
DATADef  DDPRINT,DSN=ADASAV.DDPRINT,MODE=A  
DATADef  DUMP,DUMMY  
DATADef  DDDRUCK,DSN=ADASAV.DDDRUCK,MODE=A  
DATADef  DDCARD,DSN=RUNSAV.CONTROL,MODE=A  
DATADef  DDKARTE,DSN=RESTFILE.CONTROL,MODE=A  
ADARUN
```

### *RUNSAV CONTROL A1 の内容*

```
ADARUN  PROG=ADASAV,DEVICE=dddd,DB=yyyyyy
```

### *RESTFILE CONTROL A1 の内容*

```
ADASAV  RESTPLOG  FILES=2,PLOGNUM=ppp,SYN1=1
```

オフラインで作成された **SAVE** データセットからのファイルのリストア

```

DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDDATAR1,DSN=ADABASV.V.DATA,VOL=DATAV1
DATADEF DDWORKR1,DSN=ADABASV.V.WORK,VOL=WORKV1
DATADEF DDREST1,DSN=ADABASV.V.SAVE,MODE=A
DATADEF DDPRINT,DSN=ADASAV.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADASAV.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNSAV.CONTROL,MODE=A
DATADEF DDKARTE,DSN=RESTORE.CONTROL,MODE=A
ADARUN
    
```

*RUNSAV CONTROL A1 の内容*

```
ADARUN PROG=ADASAV,DEVICE=dddd,DB=yyyyy
```

*RESTORE CONTROL A1 の内容*

```
ADASAV RESTORE FMOVE=2
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	情報
アソシエータ	ASSORn	ディスク	注 1 参照	
データストレージ	DATARn	ディスク	注 1 参照	
ワーク	WORKR1	ディスク	注 1 参照	Adabas ニュークリアスがアクティブでないときのみ必要
マルチプロテクションログ	PLOGRn	ディスク		RESTORE および RESTONL データベース/GCB に対して任意に指定します。
マルチコマンドログ	CLOGRn	ディスク		RESTORE および RESTONL データベース/GCB に対して任意に指定します。
リカバリログ (RLOG)	RLOGR1	ディスク		ADARAI の場合に必要
バックアップコピー	SAVE1-8	テープ ディスク	SYS011- SYS018 注 1 参照	SAVE 機能にのみ必要

ファイル	記号名	ストレージ	論理ユニット	情報
デュアルコピー	DUAL1-8	テープ ディスク	SYS021- SYS028 注 1 参照	SAVE 機能でバックアップを 2 部作成する場合のみ必要
バックアップコピー (RESTORE 機能の入力)	REST1-8	テープ ディスク	SYS011- SYS018 注 1 参照	RESTORE および RESTONL 機 能にのみ必要
シーケンシャルプロテクショ ンログ	PLOG	テープ ディスク	SYS010 注 1 参照	RESTONL および RESTPLOG 機能に必要
ADARUN パラメータ	SYSRDR CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 注 1 参照	
ADASAV パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	
ADASAV メッセージ		プリンタ	SYS009	

**Notes:**

1. すべてのプログラマ論理ユニットを使用できます。
2. RESTONL では、入力 SAVE テープとシーケンシャルプロテクションログは、名前 REST1 を使用して連結できます。

**ADASAV JCS の例 (VSE)**

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

以下に例とメンバの対応表を示します。

例	メンバ
データベースのセーブ	ADASAV.X
データベースのセーブ (出力コピーを 2 つ作成)	ADASAVT.X
データベースのリストア	ADASAVR.X
オンラインで作成された SAVE データセットからのファイルのリストア	ADASAVRO.X
RESTONL 機能中断後のプロテクションログの リストア	ADASAVRP.X
オフラインで作成された SAVE データセットからのファイルのリストア	ADASAVRE.X

データベースのセーブ

```

* $$ JOB JNM=ADASAV,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASAV
*         SAVE THE ENTIRE DATABASE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS011,TAPE
// PAUSE MOUNT LOAD SAVE FILE ON TAPE cuu
// TLBL SAVE1, 'EXAMPLE.DByyyyy.SAVE'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADASAV SAVE
/*
/&
* $$ EOJ

```

データベースのセーブ（出力コピーを2つ作成）

```

* $$ JOB JNM=ADASAVT,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASAVT
*         SAVE THE ENTIRE DATABASE CREATING TWO COPIES OF THE OUTPUT
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// PAUSE MOUNT LOAD SAVE TAPES ON cu1 AND cu2
// ASSGN SYS011,TAPE
// TLBL SAVE1, 'EXAMPLE.DByyyyy.SAVE'
// ASSGN SYS021,TAPE
// TLBL DUAL1, 'EXAMPLE.DByyyyy.SAVE.COPY'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADASAV SAVE TWOCOPIES
/*
/&
* $$ EOJ

```

データベースのリストア

```

* $$ JOB JNM=ADASAVR,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASAVR
*       RESTORE THE ENTIRE DATABASE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS011,TAPE
// PAUSE MOUNT LOAD SAVE FILE ON TAPE cuu
// TLBL REST1, 'EXAMPLE.DByyyyyy.SAVE'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADASAV RESTORE OVERWRITE
/*
/&
* $$ EOJ

```

オンラインで作成された **SAVE** データセットからのファイルのリストア

```

* $$ JOB JNM=ADASAVRO,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASAVRO
*       RESTORE FILES FROM SAVE DATA SETS CREATED ONLINE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// PAUSE MOUNT LOAD SAVE FILE ON TAPE cu1 AND PLOG ON TAPE cu2
// ASSGN SYS011,TAPE
// TLBL REST1, 'EXAMPLE.DByyyyyy.SAVE'
// ASSGN SYS010,TAPE
// TLBL PLOG, 'EXAMPLE.DByyyyyy.PLOG5'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADASAV RESTONL FILES=2,PLOGNUM=ppp,SYN1=1
/*
/&
* $$ EOJ

```

**RESTONL** 機能中断後のプロテクションログのリストア

```

* $$ JOB JNM=ADASAVRP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASAVRP
*       RESTORE THE PROTECTION LOG AFTER AN INTERRUPTED RESTONL
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// PAUSE MOUNT LOAD PLOG FILE ON TAPE cuu
// ASSGN SYS010,TAPE
// TLBL PLOG, 'EXAMPLE.DByyyyyy.PLOG5'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADASAV RESTPLOG PLOGNUM=ppp,SYN1=1
/*
/&
* $$ EOJ

```

オフラインで作成された **SAVE** データセットからのファイルのリストア

```

* $$ JOB JNM=ADASAVRF,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASAVRF
*       RESTORE A FILE TO ANY RABNS FROM AN OFFLINE SAVE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS011,TAPE
// PAUSE MOUNT LOAD SAVE FILE ON TAPE cuu
// TLBL REST1, 'EXAMPLE.DByyyyyy.SAVE'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASAV,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADASAV RESTORE FMOVE=2
/*
/&
* $$ EOJ

```

# 181 ADASEL ユーティリティ：プロテクションデータの選択

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ADASEL 構文
- グローバルパラメータによる ADASEL デフォルトの変更方法
- JCL/JCS の必要項目と指定例



# 182 機能概要

---

ADASELユーティリティでは、Adabasシーケンシャル（SIBA）プロテクションログまたはデュアル／マルチ（PLOG）プロテクションログ内の情報を選択します。ADASELでは情報を圧縮解除して、出力データセット（DDDRUCK/DRUCK）またはユーザー指定の出力データセットに書き込みます。

プロテクションログの中には、任意のAdabasセッション中にデータベースに適用されたすべての更新の情報が入っています。ADASELで選択した情報は、監査のために使用したり、NaturalプログラムやAdabas以外のプログラムへの入力として使用することができます。

新規のレコード、更新されたレコード、および削除されたレコードについて、ビフォーイメージ、アフターイメージ、または両方のイメージを選択できます。また、Adabas C5 コマンドを使用すると、プロテクションログに書き込まれたデータを選択することもできます。



**Note:** Adabasセッションがデュアル／マルチプロテクションロギングを使用している場合は、ADARES PLCOPY 機能でプロテクションログをコピーしてから ADASEL への入力として使用する必要があります。

Adabasセッションがシーケンシャルプロテクションログを使用している場合、およびセッションが異常終了した場合は、ADARESCOPY機能でプロテクションログをコピーしてからADASELへの入力として使用する必要があります。



# 183 ADASEL 構文

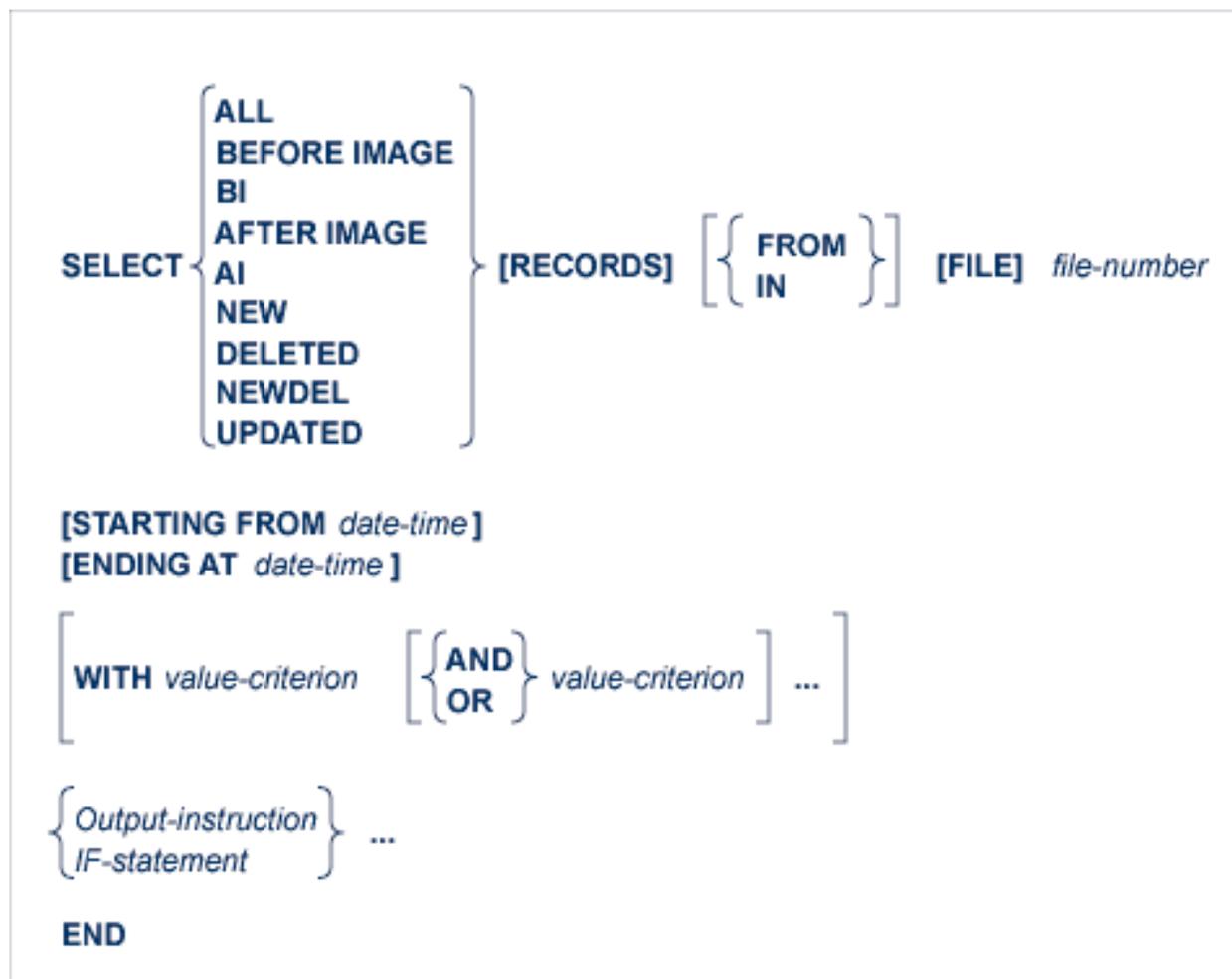
---

▪ SELECT ステートメント .....	1051
▪ 追加の選択条件 .....	1052
▪ date-time .....	1053
▪ WITH 節 .....	1053
▪ IF ステートメント .....	1054
▪ value-criterion .....	1056
▪ output-instruction .....	1061

他の Adabas ユーティリティと異なり、ADASEL では各パラメータ行の最初にユーティリティ名を付ける必要はありません。選択要求は次の要素で構成されます。

- キーワード *SELECT*。選択オプションとファイル番号が後に続きます。
- 追加の選択条件を指定する節およびステートメント（オプション）
- 1 つ以上の出力命令
- キーワード *END*

ADASEL 構文の概要を以下に示します。



複数の選択要求をコーディングできます。それぞれの要求は **SELECT** で始まり、**END** で終わります。

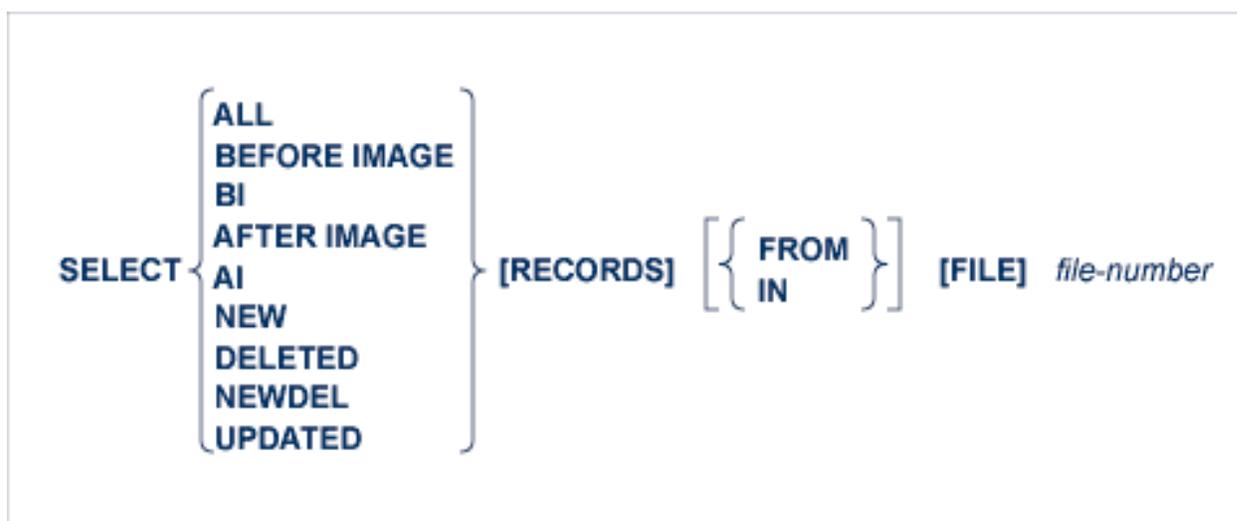
例

```
SELECT ALL RECORDS FROM FILE 1
      DISPLAY AA BB CC
END
```

```
SELECT BEFORE IMAGE FILE 2
      OUTPUT TO EXPA1
END
```

このchapterでは、次のトピックについて説明します。

## SELECT ステートメント



キーワード SELECT の後に、次の選択オプションのうちの 1 つを指定してください。

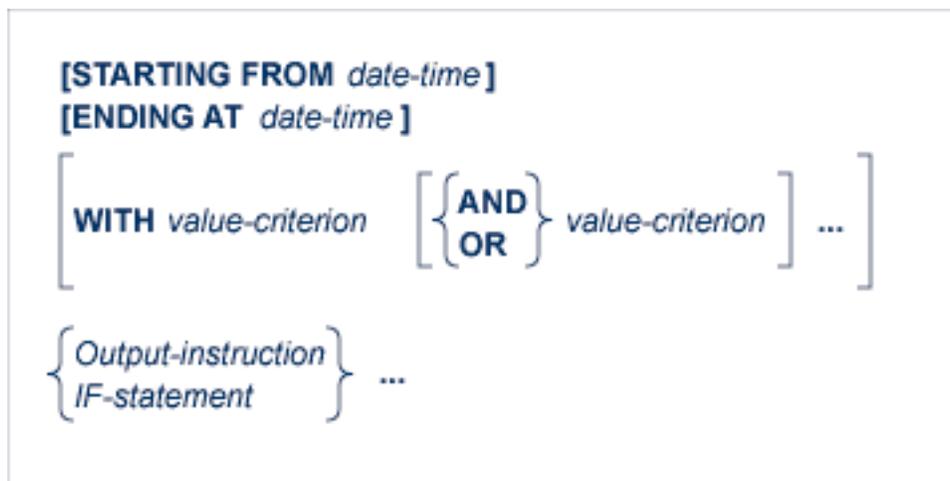
オプション	選択レコード
ALL	A1 (更新) コマンドおよび E1 (削除) コマンドのビフォーイメージ、A1 コマンドおよび N1 (追加) コマンドのアフターイメージ
BEFORE IMAGE   BI	A1 コマンドおよび E1 コマンドのビフォーイメージ
AFTER IMAGE   AI	A1 コマンドおよび N1 コマンドのアフターイメージ
NEW	N1 コマンドのアフターイメージ
DELETED	E1 コマンドのビフォーイメージ
NEWDEL	N1 コマンドのアフターイメージおよび E1 コマンドのビフォーイメージ
UPDATED	A1 コマンドのビフォーイメージとアフターイメージ

## file-number

選択するプロテクションログデータの Adabas ファイルを指定します。指定可能なファイル番号は、0～5000 または 0～(ASSO ブロックサイズ-1) のうち、どちらか小さい方の数字です。C5 コマンドによって書かれたユーザーデータを選択するには、チェックポイントファイルのファイル番号を指定します。

## 追加の選択条件

追加の選択条件を指定するときは、STARTING FROM 節、ENDING AT 節、WITH 節、および IF ステートメントが使用できます。これらのオプションの節およびステートメントは、前述の選択オプションよりも優先されます。



STARTING FROM 節および ENDING AT 節は、一定の時間内に追加、更新または削除されたレコードだけに選択を制限します。date-time 変数については、次の「[date-time](#)」セクションで説明します。

指定した値の条件を満たすレコードを選択するには、WITH 節を使用します。AND または OR 論理演算子を使用することで複数の条件を指定できます。WITH 節については、「[WITH 節](#)」のセクションで説明します。

一定の条件に基づいてレコードの選択および出力命令を実行するには、IF ステートメントを使用します。IF ステートメントについては、「[IF ステートメント](#)」のセクションで説明します。

WITH 節と IF ステートメントの両方で使用する value-criterion 変数の構文については、「[value-criterion](#)」のセクションで説明します。

出力命令については「[output-instruction](#)」を参照してください。

## date-time

*date-time* 変数に使用できるフォーマットを以下に示します。

フォーマット	説明
yyyymmdd/hhmmss	日付／時刻
J(yyyyddd/hhmmss)	ユリウス暦の日付／時刻
X'xxxxxxx'	STCK の表示

 **Note:** *yyyy* に指定可能な最小値は "1980" です。

例

1996年5月12日（ユリウス暦の日付132日）の午前0時以前（午前0時を含む）に追加、削除または更新されたすべてのレコードをファイル1から選択します。

```
SELECT ALL RECORDS FROM FILE 1
  ENDING AT J(1996132/240000)
  DISPLAY AA BB CC
END
```

1996年の1月1日と12月31日の間（1月1日と12月31日を含む）に追加、削除または更新されたすべてのレコードをファイル112から選択します。

```
SELECT ALL 112
  STARTING FROM 19960101/000000
  ENDING AT 19961231/240000
  OUTPUT TO EXPAL
END
```

## WITH 節

指定した値の条件を満たすレコードを選択するには、WITH 節を使用します。AND または OR 論理演算子を使用することで複数の条件を指定できます。

- *value-criteria* が AND 演算子で連結されている場合、レコードが選択されるためにはすべての条件を満たされる必要があります。
- *value-criteria* が OR 演算子で連結されている場合は、いずれかの条件を満たすレコードが選択されます。

*value-criterion* 変数の構文については「*value-criterion*」のセクションで説明します。

例

プロテクションログには、2つの更新されたレコードのビフォーイメージとアフターイメージが含まれています。レコードのフィールド BB の内容は、次のとおりです。

ビフォーイメージ	アフターイメージ
BB=SMITH	BB=ZINN
BB=SMITH	BB=JONES

SELECT ステートメントには、選択対象をさらに制限する WITH 節が含まれています。

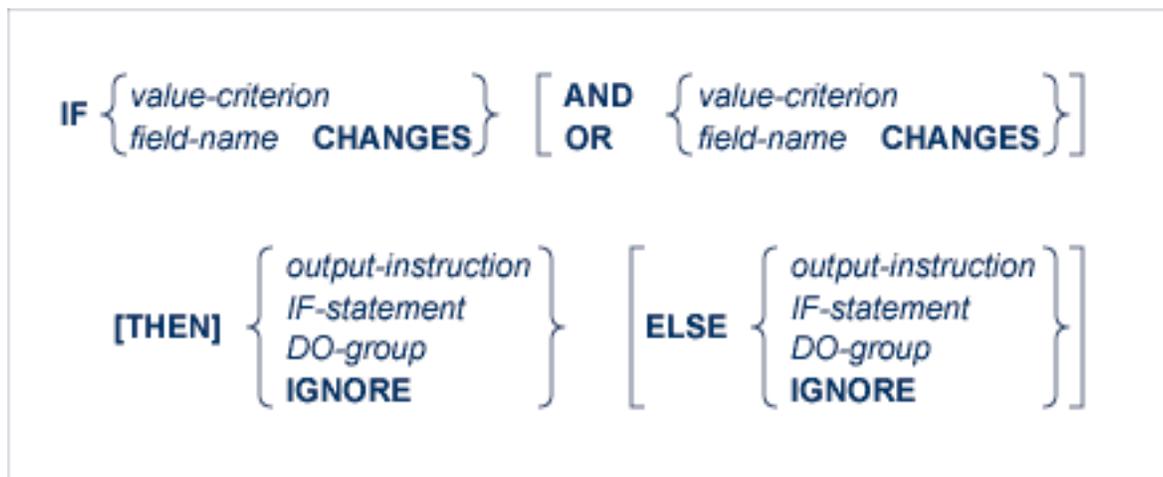
```
SELECT ALL RECORDS FROM FILE 1
  WITH BB = 'SMITH'
  DISPLAY AA BB CC
END
```

この例では、ALL オプションが使用されているにもかかわらず、2つのビフォーイメージのみが選択されます。これは、BB フィールドのビフォーイメージに "SMITH" が含まれているためです。ADASEL では、BB フィールドに "SMITH" 以外の値があるレコード（この場合は2つのアフターイメージ）をすべて無視します。AFTER IMAGE オプションが指定されている場合、レコードは1つも選択されません。

## IF ステートメント

一定の条件に基づいてレコードの選択および出力命令を実行するには、IF ステートメントを使用します。

デフォルト設定により、ADASEL では IF ステートメントのネストが 20 まで許可されます。このデフォルトを変更する詳細については、「[グローバルパラメータによる ADASEL デフォルトの変更方法](#)」セクションを参照してください



`value-criterion` 変数の構文については「[value-criterion](#)」のセクションで説明します。出力命令については「[output-instruction](#)」を参照してください。

"`field-name`CHANGES" 条件は、更新時に指定フィールドの値が変更されたレコードを選択します。ADASELでは、ビフォーイメージとアフターイメージ間の変更内容を検出します。したがって、この条件は、プロテクションログにビフォーイメージとアフターイメージの両方を書き込む A1 (UPDATE) コマンドに対してのみ有効です。`field-name` は、FDT のエレメンタリフィールドの 2 文字の Adabas 名である必要があります。グループ、ピリオディックグループ (PE)、スーパーディスクリプタ、サブディスクリプタ、フォネティックディスクリプタまたはハイパーディスクリプタは参照できません。ただし、マルチプルバリューフィールド (MU) やピリオディックグループ (PE) のメンバフィールドは参照できます。「[value-criterion](#)」の「[MU および PE メンバフィールドのインデックス](#)」のセクションを参照してください。



**Note:** "IF `field-name` CHANGES" 条件に対しては、アフターイメージのみが報告されます。変更されたフィールドのビフォーイメージとアフターイメージの両方を ADASEL を使用して報告するには、実行の OUTPUT 命令に LOGINFO オプションまたは EXTENDED オプションを指定してください。詳細については、「[OUTPUT 命令](#)」を参照してください。

DO グループの構文は次のようになります。

```
DO  output-instruction...  DOEND
```

DO グループとは、一連の出力命令 (NEWPAGE、SKIP、DISPLAY、OUTPUT) のことです。このグループはキーワード DO で始まり、キーワード DOEND で終わる必要があります。DO グループにはネストされた IF ステートメントを含めることはできません。また、DO グループは別の DO グループ内にネストすることもできません。

IGNORE を指定すると、ADASEL によって項目の表示または出力が行われません。

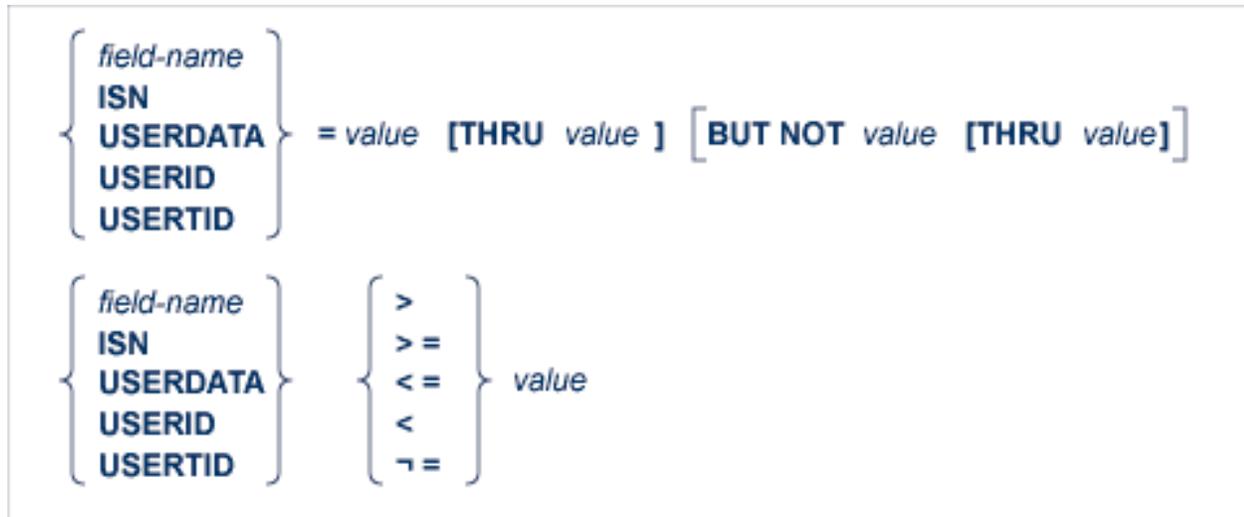
例

```
SELECT ALL FROM FILE 77
  IF AA = 'SMITH' THEN
    IF BB CHANGES THEN DO
      DISPLAY 'Field BB changed:' BB AA CC
      SKIP 1 LINE
    DOEND
  ELSE DISPLAY AA BB CC
```

```
ELSE IGNORE
END
```

## value-criterion

value-criterion は、1 つ以上の値に基づいてレコードを選択するために、WITH 節または IF ステートメントで使用します。構文は次のとおりです。



BUT NOT 節は、等式 (=) で指定された範囲から値または値範囲を除外します。

### 比較の対象

ADASEL は、値または値範囲を次の項目に対して比較します。

- 指定したフィールドの内容。 *field-name* は、FDT のエレメンタリフィールドの 2 文字の Adabas 名でなければなりません。グループ、ピリオディックグループ (PE)、スーパーディスクリプタ、サブディスクリプタ、フォネティックディスクリプタまたはハイパーディスクリプタは参照できません。ただし、マルチプルバリューフィールド (MU) やピリオディックグループ (PE) のメンバフィールドは参照できます。「[MU および PE メンバフィールドのインデックス](#)」を参照してください。
- ISN。レコードの Adabas 内部シーケンス番号です。
- USERDATA。C5 コマンドによって書かれたユーザーデータです。
- USERID。レコードを追加、削除または更新したユーザーのユーザー ID (ETID) です。
- USERTID。レコードを追加、削除または更新したユーザーの端末 ID です。

## 論理演算子

等式または不等式の論理演算子は、語句、省略形または記号で表現できます。これらの表現方法を次の表に示します。

比較	語句	省略形	記号
等しい	EQUAL	EQ	=
より大きい	GREATER THAN	GT	>
より大きいまたは等しい	GREATER EQUAL	GE	>=
より小さいまたは等しい	LESS EQUAL	LE	<=
より小さい	LESS THAN	LT	<
等しくない	NOTEQUAL	NE	≠

 **Note:** 記号 ≠ の 16 進表示は X'5F7E' です。

## 値のフォーマット

条件値のフォーマットは、比較の対象となる項目のデフォルトフォーマットに応じて異なります。

Adabas フィールド (*field-name*) のデフォルトフォーマットは、FDT に指定されたフォーマットです。下表は、条件値を表示するときの最大長 (バイト数) および有効なフォーマットを示します。

条件値		最大バイト数	最大桁数
FDT のフィールドフォーマット	有効なフォーマット		
英数字	英数字	253	
	16 進	253	506
10 進 (パックまたはアンパック)	10 進数 (0-9)	29 *	
2 進	10 進	4 *	10
	16 進	126	252
浮動小数点	16 進	8	16
固定小数点	16 進	4	8
ワイド文字	16 進	253	506

\* マイナス符号を除きます。

その他の項目のデフォルトフォーマットおよび最大長（バイト数）は、次のとおりです。

項目	デフォルトフォーマット	条件値	
		有効なフォーマット	最大長さ
ISN	2 進	10 進、16 進	4
USERDATA	英数字	英数字、16 進	30
USERID	バイナリ、英数字	10 進、16 進	8
USERTID	バイナリ、英数字	10 進、16 進	8

値のフォーマットの例 1：

デフォルトフォーマットが英数字の場合、値は英数字または 16 進で表示できます。

```
BA EQ 'SMITH' or BA EQ X'E2D4C9E3C8'
```

値のフォーマットの例 2：

デフォルトフォーマットがパックまたはアンパック 10 進の場合は、10 進数（0～9）で値を表示します。先頭のマイナス符号は、値が負であることを示します。29 桁まで（マイナス符号は除く）が指定可能です。他の特殊文字（\$、小数点、コンマなど）は許可されません。

```
NU = 123456789
NU = -987654321
```

値のフォーマットの例 3：

デフォルトフォーマットが 2 進の場合、値は 16 進または数字フォーマットで表示できます。

2 進の Adabas フィールドに対し、252 桁の 16 進数字（126 バイト）まで許可されます。

数字フォーマットでは、10 桁の 10 進数字（4 バイナリバイト）まで許可されます。したがって、-2,147,483,648～2,147,483,647 の範囲の 10 進数字で表示できます。

```
BB = 2147483647 or BB = X'80000000'
BB = -2147483648 or BB = X'7FFFFFFF'
```

英数字値

英数字の値は、アポストロフィで囲む必要があります。

```
AA = 'SMITH'
```

英数字文字列内にアポストロフィが必要な場合は、2 つの連続したアポストロフィを使用します。間にスペースや文字は入れません。

```
JJ = 'Smith''s Market'
```

16 進値

16 進値は、先頭に "X" を付け、値をアポストロフィで囲まなければなりません。

```
AA = X'E2D4C9E3C8'
```

16 進値は、偶数個の 16 進文字から成ります。

```
JJ = X'04D2'
```

### 継続行

ADASEL は 1~72 桁を入力行として扱います。英数字または 16 進値を複数行にわたって継続させるには、全文字列の一番最後にのみ閉じるアポストロフィを指定します。値は閉じるアポストロフィが見つかるまで連結されます。

英数字文字列では、アポストロフィに囲まれた先頭および最後のスペースは文字列の一部とみなされます。16 進文字列では、これらのスペースは無視されます。

### 例 1：英数字文字列

```

1.....7
2
AA = 'THIS IS AN EXAMPLE OF HOW TO CONTINUE AN ALPHANUMERIC VALU
E. KEY THROUGH COLUMN 72 AND CONTINUE IN COLUMN 1 OF THE NEXT
LINE.'
```

```

1.....7
2
AA = 'DO NOT CONTINUE AN ALPHA VALUE THIS WAY. LEADING AND
TRAILING SPACES IN COLUMNS 1-72 ARE INCLUDED.'
```

上記の 2 番目の値は、次のように扱われます。

```
'DO NOT CONTINUE AN ALPHA VALUE THIS WAY. LEADING AND TRAILING BLANKS
```

```
IN COLUMNS 1-72 ARE INCLUDED.'
```

例 2：16 進英数字文字列

```

1.....7
2.....2

XX = X'C1C2C3C4C5C6C7C8C9
      D1D2D3D4D5D6D7D8D9'
```

上記の 16 進値は、次のように扱われます。x'c1c2c3c4c5c6c7c8c9d1d2d3d4d5d6d7d8d9'

**MU** および **PE** メンバフィールドのインデックス  
**MU** フィールドまたは **PE** のメンバフィールド

*field-name* がマルチプルバリューフィールド (MU) またはピリオディックグループ (PE) のメンバフィールドを参照する場合は、フィールド名の直後にインデックス (オカレンス番号) を付ける必要があります。

AAi	"AA" は MU のフィールド名で、i はインデックスです。
BBk	"BB" は PE のメンバフィールドで、k は PE のインデックスです。

*i* および *k* に有効な値は、Adabas 8 以上がインストールされ、拡張 MU および拡張 PE のカウントが要求されている場合は "1"~"65,534" の範囲、それ以外の場合は "1"~"191" の範囲になります。

 **Note:** ファイル中で 191 個を超える MU フィールドまたは PE グループを使用することは、そのファイルが明示的に許可されている必要があります (デフォルトでは許可されません)。この場合には、ADADBS MUPEX 機能または ADACMP COMPRESS MUPEX および MUPECOUNT パラメータを使用します。

例

ファイル 12 で、フィールド JT は MU です。次のステートメントは、JT の第 2 オカレンスが "Programmer" である全レコードのビフォーイメージを選択します。

```
SELECT BI FROM FILE 12
      WITH JT2 = 'Programmer'
      DISPLAY NA
END
```

フィールド SA は、PE のメンバです。次のステートメントは、ピリオディックグループの第 3 オカレンスの SA が 35000 以上である全レコードを選択します。

```
SELECT ALL FROM 12
      WITH SA3 >= 35000
```

```
DISPLAY NA SA3
END
```

## PE 内の MU

PE 内に MU が含まれている場合は、**両方**（PE および MU）のインデックスを指定する必要があります。

ABk(i)	"AB" は MU の名前、 <i>i</i> は AB のオカレンス、 <i>k</i> は AB が属する PE のオカレンスです。
--------	--

### 例

ファイル 211 で、マルチプルバリューフィールド ST は PE のメンバです。次のステートメントによって、ピリオディックグループの第 2 オカレンスの ST の第 3 オカレンスが "PAST DUE" である全レコードが選択されます。

```
SELECT ALL FROM FILE 211
  WITH ST2(3) = 'PAST DUE'
  DISPLAY AA BB ST2(3)
END
```

## output-instruction

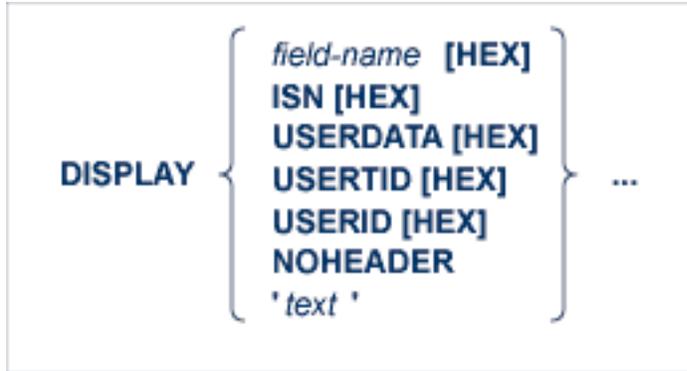
出力命令には DISPLAY、OUTPUT、SKIP、NEWPAGE などがあります。少なくとも 1 つの出力命令が必要です。出力命令は複数指定することができ、また、IF ステートメントの一部に出力命令を含めることもできます。構文は次のとおりです。

```
[DISPLAY item ... ]
[OUTPUT [ { [WITH] LOGINFO | EXTENDED } ] [TO] EXPANn ]
[SKIP n { LINE | LINES } ]
[NEWPAGE]
```

**DISPLAY** 命令については次のセクション、OUTPUT 命令については「**OUTPUT 命令**」のセクションで説明します。**SKIP** および **NEWPAGE** に関する説明も参照してください。

### DISPLAY 命令

DISPLAY は、出力レポートを DDDRUCK/DRUCK に書き込みます。構文では 1 つ以上の出力タイプを指定します。複数の出力タイプを指定するときは、それぞれを 1 つ以上のスペースで区切ります。



ここでは次の内容を表しています。

<i>field-name</i>	フィールドの内容を表示します。 <i>field-name</i> は、FDTのエレメンタリフィールドの2文字のAdabas名である必要があります。 <i>field-name</i> は、グループ、ピリオディックグループ (PE)、スーパーディスクリプタ、サブディスクリプタ、フォネティックディスクリプタまたはハイパーディスクリプタは参照できません。しかし、マルチプルバリューフールド (MU) や PE のメンバフィールドは参照できます。MUおよびPEメンバフィールドのインデックスについては、「 <b>MUまたはPE フィールド</b> 」を参照してください。
HEX	出力のタイプに対応する16進数の値を表示します。HEXは、出力不能な文字が出力に含まれる場合に特に便利です。出力タイプとその後に続くHEXキーワードの間には1つ以上のスペースを空けてください。
ISN	選択した各レコードのISNを表示します。
USERDATA	C5コマンドによってプロテクションログに書き込まれたレコードを表示します。チェックポイントファイルのファイル番号は、SELECTステートメントに指定する必要があります。
USERID	レコードを追加、削除または更新したユーザーのIDを表示します。
USERTID	レコードを追加、削除または更新したユーザーのTID (端末ID) を表示します。
NOHEADER	ヘッダーを省略します。
'text'	テキスト文字列を表示します。

### 例

修正されたレコードを選択し、テキスト文字列 "The following records were modified:" を表示します。次に、フィールド AA と CC を16進フォーマットで、フィールド BB をFDTに定義されたフォーマットで表示します。

```
SELECT UPDATED RECORDS FROM FILE 117
  DISPLAY 'The following records were modified:'
  DISPLAY AA HEX BB CC HEX
END
```

新しいレコードごとにフィールド AA を表示します。また、そのレコードを追加したユーザーのユーザー ID と端末 ID も一緒に表示します。ヘッダーは省略します。

```
SELECT NEW RECORDS FROM FILE 211
  DISPLAY AA USERID USERTID NOHEADER
END
```

### デフォルトフォーマット

フィールドはデフォルトフォーマットに従って表示されます。

英数字	入力したとおりに表示され、出力不能文字は空白に変換されて表示されます。
2進	X'80000000'より小さい値は、符号なしの10進数(0~9)で表示されます。それ以外の値は16進表記で表示されます。
パック/アンパック	10進数字(0~9)で表示されます。負の値は、先頭にマイナス符号が付加されます。

### MU または PE フィールド

*field-name* が MU または PE のメンバを参照する場合は、インデックスをフィールド名の一部として指定することによって、1つまたは一定範囲のオカレンスを表示できます。

```
DISPLAY AA5
```

Adabas 8以降がインストールされ、ファイルに対する拡張 MU および拡張 PE のカウントがオンになっている場合、有効なインデックス値は "1"~"65534" の範囲になります。それ以外の場合の有効なインデックス値は "1"~"191" の範囲です。また、インデックス範囲の上限として "N" を指定できます。ADASEL は範囲内の最初のオカレンスから開始して全オカレンスを表示します。

 **Note:** ファイル中で 191 個を超える MU フィールドまたは PE グループを使用することは、そのファイルが明示的に許可されている必要があります (デフォルトでは許可されません)。この場合には、ADADBS MUPEX 機能または ADACMP COMPRESS MUPEX および MUPECOUNT パラメータを使用します。

DISPLAY ステートメントに PE 名を指定することはできません。ピリオディックグループ全体を表示したいときは、グループ内の各フィールド名を指定しなければなりません。

PE 内に MU が含まれている場合は、両方 (PE および MU) のインデックスを指定しなければなりません。以下に示すインデックスフォーマットでは、*i* と *j* が MU インデックス、*k* と *l* が PE インデックスです。AB は PE のメンバフィールドを表し、MB は PE のメンバフィールドである MU を表します。

インデックス	表示される値
MU <sub>i</sub>	MU のオカレンス <i>i</i>
MU <sub>i-j</sub>	MU のオカレンス <i>i</i> ~ <i>j</i>
MU <sub>i-N</sub>	オカレンス <i>i</i> から始まる MU の全オカレンスの値
AB <sub>k</sub>	PE のオカレンス <i>k</i> のフィールド AB。フィールド AB は PE に属します。
AB <sub>k-1</sub>	PE のオカレンス <i>k</i> ~ <i>1</i> のフィールド AB
AB <sub>k-N</sub>	オカレンス <i>k</i> から始まる PE の全オカレンスのフィールド AB
MB <sub>k(i)</sub>	PE のオカレンス <i>k</i> にある MB のオカレンス <i>i</i> 。MB は PE に属します。
MB <sub>k - 1(i)</sub>	PE のオカレンス <i>k</i> ~ <i>1</i> にある MB のオカレンス <i>i</i>
MB <sub>k - 1(i-j)</sub>	PE のオカレンス <i>k</i> ~ <i>1</i> にある MB のオカレンス <i>i</i> ~ <i>j</i>
MB <sub>k - 1(i-N)</sub>	PE のオカレンス <i>k</i> ~ <i>1</i> にある MB の全オカレンス (オカレンス <i>i</i> 以降)
MB <sub>k-N(i - j)</sub>	PE の全オカレンス (オカレンス <i>k</i> 以降) にある MB のオカレンス <i>i</i> ~ <i>j</i>
MB <sub>k-N(i-N)</sub>	PE の全オカレンス (オカレンス <i>k</i> 以降) にある MB の全オカレンス (オカレンス <i>i</i> 以降)

例

ファイル 12 には次の PE が含まれています。

レベル	名前	説明的な名前	フォーマット	長さ	オプション	オカレンス
1	JT	JOB TITLE	A	16	DE,MU	12
1	PA	INCOME			PE	12
2	SA	SALARY	P	6	DE,MU	7
2	BO	BONUS	P	5		

下記は、ファイル 12 に対する正しい DISPLAY ステートメントを表しています。

```
SELECT NEW FROM FILE 12
  DISPLAY JT1
END
```

```
SELECT ALL FROM FILE 12
  DISPLAY JT1-5 SA1-5(1-N) BO1-5
END
```

```
SELECT ALL FROM FILE 12
  WITH JT3 = 'Programmer' THRU 'Systems Analyst'
```

```
DISPLAY JT3 SA3(1-N) BO3
END
```

```
SELECT UPDATED FROM FILE 12
  DISPLAY JT2-N SA2-N(1-N)
END
```

## OUTPUT 命令

OUTPUT 命令によって、プロテクションログの非圧縮レコードが出力データセットに書き込まれます。

OUTPUT [ [WITH] LOGINFO  
EXTENDED ] [TO] EXPAn

最高 20 個の出力データセットが使用できます。出力データセットは EXPAn パラメータおよび DDEXPAn/EXPAn ジョブ制御ステートメントに指定します。

### 例

更新または削除された全レコードのビフォーイメージをデータセット DDEXPA1/EXPA1 に書き込みます。

```
SELECT BEFORE IMAGE FILE 2
  OUTPUT TO EXPA1
END
```

### 出力レコードのフォーマット

出力レコードのフォーマットは、LOGINFO パラメータか EXTENDED パラメータのどちらが指定されているかに応じて異なります。LOGINFO および EXTENDED は、追加情報の表示に使用されます。

下の表に、すべての出力レコードに共通のフィールドを示します。カッコ内の値は、LOGINFO (バイト 32~38) または EXTENDED (バイト 64~70) が指定されている場合のフィールド位置を示しています。

バイト	説明	
0~1	プロテクションログレコード長 (バイナリ形式)	
2~3	ゼロに設定 (X'0000')	
4~5	レコードのイメージタイプ	
	C'BI'	ビフォーイメージ
	C'AI'	アフターイメージ

バイト	説明
	C'C5'   ユーザーデータ
6~7	Adabas のファイル番号 (バイナリ形式)
8~9 (32~33, 64~65)	非圧縮レコードの長さ (このフィールドおよび ISN を含む)
10~13 (34~37, 66~69)	ISN (バイナリ形式) または C5 コマンドから得られたユーザーデータ
14 (38, 70)	圧縮解除プロテクションログデータの始まり

 **Note:** 各ブロックの最初のレコードの先頭には、ブロック長を表す 2 バイトおよび空値または空白から成る 2 バイトが付加されます。

プロテクションログレコードのフィールドは、ファイルの FDT に定義された順番、長さおよびフォーマットに従って提供されます。FDT に定義された長さよりも長い英数字フィールドは切り捨てられます。FDT に定義された長さよりも長い数字フィールドがある場合は、ADASEL は異常終了します。

MU および PE には、オカレンス数を示す 1 バイトのバイナリフィールドがあります。

可変長フィールドのデフォルト長は 0 であり、先頭に値の長さ (このフィールドを含む) を示す 1 バイトのフィールドが付加されます。

NC 省略オプション付きで定義されたフィールドに空値が含まれていた場合、その空値は ADASEL によって空の値 (フィールドのフォーマットに応じて空白またはゼロ) に圧縮解除されます。このような NC フィールドの空値処理は、ADASEL にのみ当てはまります。

## LOGINFO

LOGINFO を指定すると、各レコードに次の情報が追加されます。

バイト	説明
8~15	レコードを追加、削除または更新したユーザーの ID
16~19	レコードを追加、削除または更新したユーザーの TID の下位 4 バイト (コミュニケーション ID から取得、TP モニタのユーザーのみ)
20~23	レコードの存在するデータストレージ RABN (バイナリ形式)
24~27	レコードのデータプロテクションブロック番号 (バイナリ形式)
28~31	更新タイムスタンプ (STCK の上位 4 バイトがバイナリ形式で格納される)

## EXTENDED

EXTENDED を指定すると、各レコードに次の情報が追加されます。

バイト	説明				
8～15	レコードを追加、削除または更新したユーザーの ID (ETID)				
16～23	レコードを追加、削除または更新したユーザーの端末 ID の下位 8 バイト (コミュニケーション ID から取得、TP モニタのユーザーのみ)				
24～27	レコードの存在するデータストレージ RABN (バイナリ形式)				
28～31	レコードのデータプロテクションブロック番号 (バイナリ形式)				
32～35	更新タイムスタンプ (STCK の上位 4 バイトがバイナリ形式で格納される)				
36	バックアウトインジケータ： <table border="1" data-bbox="391 625 1068 724"> <tr> <td>C'B'</td> <td>レコードはバックアウトの結果です。</td> </tr> <tr> <td>C'</td> <td>通常のレコードです。</td> </tr> </table>	C'B'	レコードはバックアウトの結果です。	C'	通常のレコードです。
C'B'	レコードはバックアウトの結果です。				
C'	通常のレコードです。				
37	予約				
38～41	トランザクション番号				
42～63	予約				

出力データセットの指示

EXPA<sub>n</sub> パラメータは出力データセットを識別します。n の値は、DDEXPA<sub>n</sub>/EXPA<sub>n</sub>JCL ステートメントの値と一致する必要があります。有効な出力データセット番号は 1～20 であり、先行ゼロは付加しません。

有効なステートメント：	OUTPUT TO EXPA3
無効なステートメント：	OUTPUT TO EXPA03

DD/EXPA<sub>n</sub>JCL ステートメントにも上記の規則が適用されます。

例

ファイル 1 の全レコードを選択します。BA フィールドの内容が "SMITH" または "SMYTH" である非圧縮レコードを DDEXPA1/EXPA1 に書き込みます。その他のレコードはすべて DDEXPA2/EXPA2 に書き込みます。

```
SELECT ALL RECORDS FROM FILE 1
IF BA = 'SMITH' OR BA = 'SMYTH'
  THEN OUTPUT TO EXPA1
ELSE
  OUTPUT TO EXPA2
END
```

**NEWPAGE 命令と SKIP 命令**

NEWPAGE 命令と SKIP 命令はページのフォーマットを制御します。

- NEWPAGE は、次の行を表示する前にページイジェクトを強制します。
- SKIP を指定すると、次の行を表示する前に、指定された行数の空白行が出力されます。

例

```
SELECT ALL RECORDS FROM FILE 1
  WITH BA EQUAL 'SMITH' THRU 'SMYTH'
IF BA CHANGES THEN DO
  NEWPAGE
  DISPLAY 'NEW NAME' BA BB BC
DOEND
ELSE DO
  SKIP 2 LINES
  DISPLAY BA BB BC
DOEND
END
```

# 184 グローバルパラメータによる ADASEL デフォルトの変更方法

---

ADASEL グローバルパラメータは、デフォルトのテーブルおよびバッファサイズを変更します。これらの変更は、SET GLOBALS ステートメントが指定された ADASEL 実行に対してのみ有効となります。

SET GLOBALS を使用する場合、最初の ADASEL 入力ステートメントとして SET GLOBALS を指定する必要があります。ただし、その前にコメントを指定できます。構文は次のとおりです。

```
SET GLOBALS { global-parameter = value }...
```

パラメータ名、等号、値の間にはスペースを入れません。ただし、各パラメータは1つ以上のスペースで区切る必要があります。特殊文字を区切り記号として使用することはできません。

複数行にまたがる場合は、SET GLOBALS キーワードを各行で繰り返す必要があります。

例

```
SET GLOBALS LST=15000 NF=15
SET GLOBALS LS=132
```

先頭がパラメータ名でない空白以外の文字列が出現した時点で、SET GLOBALS ステートメントは終了します。したがって、末尾にはコメントを指定できません。

## グローバルパラメータによる ADASEL デフォルトの変更方法

ADASEL は次のグローバルパラメータを提供します。下線部分はデフォルト値を示しています。

パラメータ	説明
LST={len   12000 }	ステートメントテーブルの長さを指定します。このテーブルは、変換された ADASEL ステートメントの格納に使用されます。各ステートメントはその複雑さに応じて1つ以上のセグメントに変換されます。各セグメントの長さは、44バイトに値の長さを加えた長さです。例えば、IF BA EQ 'SMITH'... は、49バイトを必要とします。すなわち、44バイトに"SMITH"の5バイトを加えた長さです。デフォルトのテーブルサイズ(12,000バイト)は、約200セグメントまで扱えます。テーブルサイズを超過すると、SEL003エラーが発生します。
NCFLD={n   10} NCUPD={n   10}	選択問い合わせに許可されている "field-name CHANGES" ステートメントの最大数、および元のセッションでの並行更新の最大数。ステートメントに CHANGES 条件が含まれている場合、ADASEL は NCFLD * NCUPD を指定したチェンジプールを使用して変更されたフィールド値を追跡します。このプールが小さすぎると、エラー SEL060 が発生します。その場合は、パラメータの一方または両方を増やしてから ADASEL を再実行してください。
NF={n   20}	1回の ADASEL 実行で処理できる最大ファイル数を指定します。NF は、処理する各ファイルの FDT にスペースを割り当てるために使用します。NF 値を超過すると、SEL014エラーが発生します。この値は、最大出力ファイル数 (DDEXPAN/EXPAN) とは無関係です。1回の ADASEL 実行で20ファイル以上を処理できますが、書き込み可能な出力ファイル数は20までです。
NIF={n   20}	ネストされた IF の許容レベル数。
NOUSERABEND	このパラメータを指定すると、エラーが発生した後、ADASEL はユーザーアバンド 034 の代わりにコンディションコード 20 で終了します。
NV={n   100}	フィールド値の数。この値は、フィールド値を評価するためのテーブル割り当てに使用します。ステートメントに指定された各フィールドごとに1つのエントリが必要です(重複を含みます)。例えば、次のステートメントでは同じ Adabas フィールドが使用されていますが、2つのエントリを必要とします。 <pre>IF BA ='SMITH'     THEN OUTPUT TO EXPA1 ELSE IF BA ='SMYTH'     THEN OUTPUT TO EXPA2</pre>
LPV={n   0}	PE のフィールド値の評価に使用する PE 値テーブルの長さ。通常、ADASEL は PE のオカレンスとして推測した数を使用してテーブルサイズを計算します。テーブルサイズが不十分な場合は SEL047 エラーが発生します。面上の指示に従って、グローバル LPV パラメータを使用してテーブルサイズを増加できます。
PS={n   60}	ページサイズパラメータは、新しいページを開始するまでの出力行数を変更するときに使用します。最小のページサイズは2であり、最大は999です。
LS={n   80}	行サイズパラメータを使用すると、出力する列数を変更できます。出力行が行サイズよりも長い場合、その行は最も近くにある空白の箇所まで出力されます。その行の残りの部分は、次の出力行の1列目から続けて出力されます。最小の行サイズは1であり、最大は132です。

# 185

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	1072
▪ z/OS .....	1073
▪ z/VM .....	1074
▪ VSE .....	1075

このセクションでは、ADASEL を BS2000、z/OS、z/VM、および VSE の各システムで実行するために必要なジョブ制御情報について説明し、システムごとのジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
プロテクションログ	DDSIIN	テープ/ディスク	シーケンシャルログ
選択データ	DDEXPA <sub>n</sub>	テープ/ディスク	ADASEL の出力
アソシエータ	DDASSOR <sub>n</sub>	ディスク	
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADASEL パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/ DDPRI <sub>n</sub> T		メッセージおよびコード
ADASEL メッセージ	SYSLS <sub>T</sub> / DDDRUCK		メッセージおよびコード

### ADASEL JCL の例 (BS2000)

SDF フォーマットの場合：

```

/.ADASEL LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A S E L ALL FUNCTIONS
/REMARK *
/DELETE-FILE SEL.AUS
/SET-JOB-STEP
/CREATE-FILE SEL.AUS,PUB(SPACE=(48,48))
/SET-JOB-STEP
/ASS-SYSLST L.SEL
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAyyyyy.ASSO
/SET-FILE-LINK DDSIIN,ADAyyyyy.SIBA
/SET-FILE-LINK DDEXPA1,SEL.AUS
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADASEL,DB=yyyyy,IDTNAME=ADABAS5B
SELECT ALL FROM FILE 11
DISPLAY AA BB BA BC CA CC
    
```

```
END
/LOGOFF SYS-OUTPUT=DEL
```

### ISP フォーマットの場合：

```
/.ADASEL LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A S E L ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.SEL
/FILE ADA.ASSO ,LINK=DDASSOR1
/FILE ADA.MOD ,LINK=DDLIB
/FILE ADAyyyyy.SIBA ,LINK=DDSIIN
/FILE SEL.AUS ,LINK=DDEXPA1 ,SPACE=(48,48)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADASEL,DB=yyyyy,IDTNAME=ADABAS5B
SELECT ALL FROM FILE 11
DISPLAY AA BB BA BC CA CC
END
/LOGOFF NOSPOOL
```

## z/OS

データセット	DD名	ストレージ	追加情報
プロテクションログ	DDSIIN	テープ/ディスク	シーケンシャルログ
選択データ	DDEXPA <sub>n</sub>	テープ/ディスク	ADASEL の出力
アソシエータ	DDASSOR <sub>n</sub>	ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADASEL パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADASEL メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## ADASEL JCL の例 (z/OS)

```
//ADASEL      JOB
//*
//*      ADASEL:
//*      SELECT PROTECTION DATA
//*
//SEL          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1   <=== ASSO
//DDDATAR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1   <=== DATA
//DDWORKR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1   <=== WORK
//DDSIIN      DD   DISP=OLD,DSN=EXAMPLE.DByyyyyy.PLOG5     <=== OUTPUT ADARES
//*                                                    PLCOPY
//DDEXPA1     DD   DISP=(,CATLG),DSN=EXAMPLE.DByyyyyy.EXPA1, <= OUTPUT ADASEL
//                                                    SPACE=(TRK,(10,5),RLSE),VOL=SER=vvvvvv,UNIT=uuuu
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
//SYSUDUMP    DD   SYSOUT=X
//DDCARD      DD   *
ADARUN PROG=ADASEL,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE     DD   *

*
*      SELECT UPDATES FOR FILE NUMBER 1
*
SELECT ALL FROM FILE 1
      OUTPUT TO EXPA1
END
/*
```

この例については、MVSJOBS データセットの ADASEL を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
プロテクションログ	DDSIIN	テープ/ディスク	シーケンシャルログ
選択データ	DDEXPA <sub>n</sub>	テープ/ディスク	
アソシエータ	DDASSOR <sub>n</sub>	ディスク	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADASEL パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード

データセット	DD名	ストレージ	追加情報
ADASEL メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード

## 例 (z/VM)

```

DATADEF DDASSOR1,DSN=ADABASV.V.ASSO,VOL=ASSOV1
DATADEF DDSIIN,DSN=ADABASV.V.SIBA,UNIT=181,VOL=SIBAXX
DATADEF DDEXPA1,DSN=ADABASV.V.EXPA1,MODE=A
DATADEF DDPRINT,DSN=ADASEL.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDDRUCK,DSN=ADASEL.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNSEL.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADASEL.CONTROL,MODE=A
LOAD ADARUN ( START

```

## RUNSEL CONTROL A1 の内容

```
ADARUN PROG=ADASEL,DEVICE=dddd,DB=yyyyy
```

## ADASEL CONTROL A1 の内容

```

SELECT ALL FROM FILE 11
  DISPLAY ISN
END

```

## VSE

ファイル	ファイル名	ストレージ	論理ユニット	追加情報
プロテクションログ	SIIN	テープ ディスク	SYS010 *	シーケンシャルログ
選択データ	EXPA1-20	テープ ディスク	SYS011-SYS030 *	ADASEL の出力
アソシエータ	ASSORn	ディスク	*	
ADARUN パラメータ	SYSRDRCARD	リーダー/テープ/ ディスク		オペレーションマニュアル
ADASEL パラメータ	SYSIPT	リーダー		
ADARUN メッセージ	SYSLST	プリンタ		メッセージおよびコード
ADASEL メッセージ	SYS009	プリンタ		メッセージおよびコード

\* すべてのプログラマ論理ユニットを使用できます。

## ADASEL JCS の例 (VSE)

```
* $$ JOB JNM=ADASEL,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASEL
*         SELECT PROTECTION DATA
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// PAUSE MOUNT LOAD INPUT PLOG FILE ON TAPE cuu
// ASSGN SYS010,TAPE
// TLBL SIIN,'EXAMPLE.DByyyyy.PLOG5'
// DLBL EXPA1,'EXAMPLE.ADAyyyyy.EXPA1'
// EXTENT SYS015,,,,ssss,nnnn
// ASSGN SYS015,DISK,VOL=vvvvvv,SHR
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADASEL,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
*         SELECT UPDATES FOR FILE NUMBER 1
*
SELECT ALL FROM FILE 1
        OUTPUT TO EXPA1

END
/*
/&
* $$ EOJ
```

この例については、メンバ ADASEL.X を参照してください。

# 186 ADAULD ユーティリティ：ファイルのアンロード

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- UNLOAD FILE：指定したファイルのアンロード
- ADAULD 入力処理
- ADAULD 出力処理
- ADAULD ユーザー出口9
- JCL/JCS の必要項目と指定例



# 187 機能概要

ADAULD ユーティリティでは、Adabas ファイルをアンロードします。Adabas ファイルをアンロードする主な理由を下記に示します。

- 非 Adabas プログラムによるデータの処理を可能にするため。この場合、ファイルをアンロードした後、ADACMP ユーティリティの DECOMPRESS 機能を用いてファイルを圧縮解除しなければなりません。
- 同じデータをもつ1つまたは複数のテストファイル作成のため。この手順では、ファイルをアンロードした後、別のファイル番号を持つテストファイルとして再ロードする必要があります。
- フィールド定義テーブル (FDT) を変更するため。これにはファイルをアンロードし、圧縮解除し、変更したフィールド定義を使用して圧縮し、さらに再ロードしなければなりません。ADADBS ユーティリティを使ってフィールド定義をファイルに追加する場合は、ファイルを最初にアンロードする必要はありません。

レコードをアンロードする順序は、次のいずれかが可能です。

物理順	レコードをデータストレージ内に物理的に位置している順序でアンロードします。
論理順	レコードをユーザーが指定したディスクリプタの値の順序でアンロードします。
ISN	レコードを ISN の昇順にアンロードします。

アンロードするレコードのサブセットを示すために、選択条件 (SELCRIT/SELVAL) を任意で使用します。

- 選択条件が指定されていない場合、全レコードが物理順にアンロードされます。
- 選択条件が指定され、ソート順 (SORTSEQ) は指定されていない場合、レコードは ISN 順にアンロードされます。
- 選択条件とソート順の両方が指定された場合は、選択されたレコードがニュークリアスのワークプールエリアでソートされ、指定されたソート順にアンロードされます。

- 選択条件に一致するレコードが1つも見つからなかった場合、ADAULD は FDT のみが含まれるファイルを作成し、レジスタ 15 のコンディションコード 4 を発行します。

アンロードされたレコードは圧縮形式で出力されます。出力レコードの形式は、ADACMP ユーティリティで作成されたレコードと同じです。

MODE=SHORT オプションを使用すると、ファイルのノーマルインデックスとアップパーインデックスを作成するために必要となるディスクリプタエントリがアンロード処理で省略されます。これにより、アンロードに必要な処理時間を減少することができます。ただし、MODE=SHORT を使用して作成された出力は、ディスクリプタ情報がすべて排除されているため、MODE=SHORT を使用しないでアンロードした同じファイルとは FDT が異なります。



**Note:** 中断した ADAULD UNLOAD FILE は、最初から再実行する必要があります。

# 188

## UNLOAD FILE：指定したファイルのアンロード

---

- 基本パラメータ ..... 1082
- オプションのパラメータとサブパラメータ ..... 1082
- 例 ..... 1088

```

ADAULD [UNLOAD] FILE= file-number
[CODE = cipher-key]
[DDISN]
[ERRLIM = {error-threshold-count | 1}]
[ETID = multiclient-file-owner-id]
[LPB = { prefetch-buffer-size | based-on-ADARUN-lu}]
[LRECL = { maximum-compressed-length | 4000 }]
[MODE = SHORT]
[NOUSERABEND]
[NUMOUT = { 1 | 2 }]
[NUMREC = number ]
[PASSWORD = file-password ]
[SAVETAPE]
[PLOGNUM = plog-number, { SYN1 | SYN4 } = plog-block-number ]
[TEMPDEV = { device-type | ADARUN-device }]
[TEMPSIZE = size ]
[SELCRIT = 'selection-criteria', SELVAL = 'values-for-selection-criteria' ]
[STARTISN = value ]
[
  SORTSEQ = {
    descriptor [,MU] [,NU]
    ISN [, STARTISN = value]
    physical-sequence
  } ]
[TEST]
[UTYPE = { EXU | EXF }]

```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE

アンロードするファイルの番号を指定します。チェックポイントファイルおよびセキュリティファイルはアンロードできません。

## オプションのパラメータとサブパラメータ

### CODE：サイファコード

アンロードするファイルが暗号化されている場合、CODEで適切なサイファコードを指定する必要があります。

**DDISN**：アンロードした ISN の DD/ISN 出力ファイルの作成

DDISN パラメータを指定すると、ADAULD ではアンロードした ISN のリストをシーケンシャル出力ファイル DD/ISN に書き込みます。DD/ISN は、アンロードされたレコードを削除するために ADALOD UPDATE の入力として使用できるように、構造化されています。

DDISN キーワードを指定すると、次のように処理されます。

- 指定した DD/ISN ファイルが JCL で見つからない場合は、ADAULD がエラー 081 で終了します。
- マルチプルバリューフィールドを参照するハイパーディスクリプタまたはディスクリプタが SORTSEQ によって指定された場合は、ADAULD がエラー 133 で終了します。これは、重複した ISN が DD/ISN に含まれている可能性があるためです。

**ERRLIM**：エラーの限界値

ERRLIM は、処理が終了する前に ADAULD によって受け付けられるニュークリアスレスポンスコードの最大数を設定します。デフォルト設定は 1 です。つまり最初のエラーが発生すると、ADAULD はエラー 124 で終了します。

ERRLIM の値を 1 よりも大きく設定すると、レスポンスコード 255（すべてのアタッチドバッファが割り当て済み）のような、周期的に発生する状態に対応することができます。この場合、ユーティリティはリターンコード 8 で終了し、ユーザーアベンドは発生しません。ADAULD の出力ファイルを使用できます。ただし、返されるニュークリアスレスポンスコードによってはレコードが存在しない可能性があります。

**ETID**：マルチクライアントファイル所有者 ID

マルチクライアントファイルをアンロードする場合は、ETID パラメータを使用すると、UNLOAD 処理の対象を指定されたユーザーの所有するレコードのみに制限できます。ETID パラメータを指定しなければ、全レコードをアンロードします。

マルチクライアントファイルに対して SELCRIT/SELVAL パラメータが指定されている場合は、必ず ETID パラメータも指定しなければなりません。

**LPB**：プリフェッチバッファサイズ

LPB は、内部プリフェッチバッファの大きさを指定します。最大値は 32767 バイトです。

デフォルトでは、ADAULD はプリフェッチバッファをできるだけ大きくして、最高のパフォーマンスを得ようとしています。LPB パラメータを使用すると、ユーザーはプリフェッチバッファを小さくすることを選択できます。このことは、ADAULD でプリフェッチが頻繁に使用されることによって、他のユーザーに関連するニュークリアスリソースが大量に消費される場合などに役立ちます。

デフォルト値は、ADARUN LU パラメータによって設定された中間ユーザーバッファの長さに依存します。ADAULD は、Adabas 制御情報を格納するために必要なスペース（108 バイト）および指定した最大圧縮レコード長（LRECL）を LU の値から引いて、デフォルトの LPB 値を決定します。この結果は、LPB に許された最大値（32767 バイト）以下である必要があります。

ニュークリアスを必要とする ADAULD のようなユーティリティのレコードバッファを格納するために、LU のデフォルト値は 65535（最大値）に設定されます。LU の値が小さすぎる場合、ADAULD は LPB パラメータに指定された値を縮小することがあります。

### **LRECL**：最大圧縮レコード長

LRECL は、返される最大圧縮レコード長（DVT を含む）をバイト単位で指定します。

この長さは、Adabas レコードバッファ長として使用されます。この値が小さすぎる場合はレスポンスコード 53 が発生します。デフォルト値は 4000 バイトで、許容最大値は 32760 バイトです。

### **MODE=SHORT**：ディスクリプタ情報の排除

このパラメータは、ノーマルインデックスおよびアップパーインデックスの構築に使用されるディスクリプタ情報を出力に含めるかどうかを指定します。

MODE=SHORT を指定すると、ディスクリプタ情報はアンロードされません。フィールド定義テーブル（FDT）が出力データセットに書き込まれるとき、FDT からすべてのディスクリプタ情報が削除されます。

出力が ADALOD ユーティリティへの直接入力として使用される場合、ファイルにディスクリプタは含まれません。

スーパーディスクリプタの場合は、MODE=SHORT を指定するとスーパーフィールドとしてアンロードされます。出力が ADALOD への直接入力として使用される場合、ロードされたファイルにはスーパーフィールドが含まれます。

### **NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABEND を指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ "utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

### **NUMOUT**：出力ファイルの数

NUMOUT は、作成される出力ファイルの数を指定します。1 よりも大きな数を指定する場合は、ユーザー出口 9 を使用して、出力ファイルの選択（DD/OUT1 または DD/OUT2）を制御する必要があります。詳細については、『Adabas DBA リファレンスマニュアル』を参照してください。使用できる値は 1（デフォルト）および 2 です。

### **NUMREC**：アンロードするレコード数

NUMREC は、アンロードするレコードの数を制限します。このパラメータを省略すると、すべての制限が無効になります。

### **PASSWORD**：ファイルパスワード

アンロードするファイルがセキュリティ保護されている場合は、PASSWORD パラメータを指定する必要があります。

**PLOGNUM**：プロテクションログ番号

SAVETAPE を指定し、かつ ADAULD の入力としてオンラインセーブテープを使用する場合、DD/PLOG シーケンシャル入力データセットとして対応するプロテクションログが必要です。

ADASAV バージョン 5.1 で作成したオンラインセーブテープを使用する場合は、追加パラメータ PLOGNUM と SYN1 または SYN4 を指定しなければなりません。

- PLOGNUM は、ADASAV SAVE オペレーションがアクティブの間に使用したニュークリアsproテクションログ番号です。
- SYN1 または SYN4 は、対応する ADASAV SAVE を開始した SYN1 または SYN4 チェックポイント含むブロックの番号です。

ADASAV バージョン 5.2 以上で作成したオンラインセーブテープは、この情報を含んでいます。PLOGNUM SYN1 または SYN4 を指定すると、テープ情報は書き換えられます。

**SAVETAPE**

セーブテープからファイルをアンロードします。このオプションは、セーブテープのファイルをブロックサイズの異なるデータベースに移動する場合、およびテスト環境の異なるセーブテープのファイルを使用する場合に役立ちます。

オンラインセーブテープを使用する場合は、TEMPDEV パラメータも指定する必要があります。オンラインセーブテープが ADASAV バージョン 5.1 で作成されている場合は、PLOGNUM と SYN1 または SYN4 も指定する必要があります。PLOGNUM と SYN1 または SYN4 を ADASAV バージョン 5.2 以降で作成されたオンラインセーブテープに指定すると、テープに含まれている情報が書き換えられます。

詳細については、「[入力セーブテープの処理](#)」のセクションを参照してください。

SORTSEQ パラメータと SELCRIT パラメータを SAVETAPE と同時に使用することはできません。

ETID パラメータを SAVETAPE と同時に使用することはできません。マルチクライアントファイルの特定クライアントのレコードを選択するにはユーザー出口 9 を使用する必要があります。詳細については、「[ADAULD ユーザー出口 9](#)」を参照してください。

セーブテープからアンロードするファイルが暗号化されている場合は、通常どおり CODE パラメータを指定する必要があります。

 **Note:** Adabas Delta Save Facility で使用するための特殊な SAVETAPE 機能が用意されています。詳細については、『*Adabas Delta Save Facility* マニュアル』を参照してください。

**SELCRIT**：選択条件

SELCRIT パラメータを使用すると、指定した選択条件を満たすレコードのみがアンロードされるように制限できます。選択条件は、『*Adabas* コマンドリファレンスマニュアル』の説明に従い、サーチバッファ構文を使用して指定する必要があります。

複数の条件を指定する場合は、次のように、それぞれの条件に個別の ADAULDSELCRIT ステートメントを付加します。

```
ADAULD SELCRIT = 'AA, 20, A, D, '  
ADAULD SELCRIT = 'AB, 10, A.'
```

ADAULD によって、これらの条件が次のように連結されます。

```
'AA, 20, A, D, AB, 10, A.'
```

選択条件に対応する値は、SELVAL パラメータを使用して指定する必要があります。

### SELVAL：選択条件に対する値

SELVAL は、SELCRIT パラメータで指定した選択条件に対応する値を指定します。値のフォーマットはバリュースタンプに使用するものと同じです。詳細は『*Adabas コマンドリファレンスマニュアル*』を参照してください。

値は複数の行に指定できます。パックされた 10 進数またはバイナリ値は、次の例に示すように 16 進形式で指定することができます。

```
SELVAL='PARIS '  
SELVAL=X'00149C'  
SELVAL='AB100'
```

### SORTSEQ：アンロードの順序

SORTSEQ は、アンロードする ISN のソート順を指定します。SORTSEQ を指定しない場合、ISN は物理順でアンロードされます。

ディスクリプタ名を指定すると、そのディスクリプタ値の論理順の昇順にレコードがアンロードされます。ディスクリプタ、サブディスクリプタ、ハイパーディスクリプタまたはスーパーディスクリプタの名前を指定できます。ピリオディックグループ内のフィールドは参照しないでください。

- ディスクリプタ名がマルチプルバリュースタンプを参照する場合は、MU を指定する必要があります。この場合は、レコード内のディスクリプタのそれぞれの値について同じレコードが値の昇順で 1 回アンロードされます。MU が指定されない場合（デフォルト）、ADAULD は MU ディスクリプタを拒否してエラーメッセージを出力します。
- 空値省略 (NU) オプションで定義されているフィールドがディスクリプタ名によって参照される場合は、NU を指定する必要があります。この場合、空値が含まれるディスクリプタのレコードはアンロードされません。NU が指定されない場合（デフォルト）、ADAULD は NU ディスクリプタを拒否します。

 **Note:** ディスクリプタフィールドまたは後続のフィールドが初期化されていない（値を持たない）場合は、空値省略されたディスクリプタフィールドでなくても、そのレコードはインバーテッドリストに含まれません。したがって、そのレコードはユーティリティの実行時に除外されます。

SORTSEQ=ISN が指定されると、レコードは ISN の順序で昇順にアンロードされます。

SELCRIT/SELVAL と SORTSEQ の両方を指定すると、レコードはニュークリアスのワークプールエリアでソートされます。したがって、ADARUN LS および LWP セッションパラメータで十分なスペースを指定する必要があります。LS および LWP パラメータの詳細については、『Adabas オペレーションマニュアル』を参照してください。

**STARTISN：開始 ISN**

STARTISN は SELCRIT/SELVAL パラメータおよび SORTSEQ パラメータとともに使用します。STARTISN を使用すると、アンロード対象レコードを ISN に従って制限できます。STARTISN は単独で指定できません。

- STARTISN を SELCRIT/SELVAL とともに指定すると、STARTISN で指定した値以上の ISN を持ち、さらにそのフィールド内容が SELCRIT/SELVAL 条件を満たすレコードが、ディスクリプタ名別に ISN の昇順でアンロードされます。
- STARTISN を SORTSEQ=ISN とともに指定すると、STARTISN で指定されたレコードで始まるすべてのレコードが ISN 順でアンロードされます。

**SYN1|SYN4：開始ブロック番号**

SAVETAPE を指定し、かつ ADAULD の入力としてオンラインセーブテープを使用する場合、DD/PLOG シーケンシャル入力データセットとして対応するプロテクションログが必要です。

ADASAV バージョン 5.1 で作成したオンラインセーブテープを使用する場合は、追加パラメータ PLOGNUM と SYN1 または SYN4 を指定しなければなりません。

- PLOGNUM は、ADASAV SAVE オペレーションがアクティブの間に使用したニュークリアスプロテクションログ番号です。
- SYN1 または SYN4 は、対応する ADASAV SAVE を開始した SYN1 または SYN4 チェックポイント含むブロックの番号です。

ADASAV バージョン 5.2 以上で作成したオンラインセーブテープは、この情報を含んでいます。PLOGNUM SYN1 または SYN4 を指定すると、テープ情報は書き換えられます。

**TEMPDEV：一時ストレージデバイスタイプ**

SAVETAPE を指定し、ADAULD の入力としてオンラインセーブテープを使用する場合、処理中は中間データセットを使用して中間データが格納されます。TEMPDEV パラメータは、この中間データセットに使用されるデバイスタイプを示します。このパラメータは、使用するデバイスタイプが、ADARUN DEVICE パラメータで TEMP に割り当てられた標準のデバイスタイプと異なる場合にのみ必要です。

中間データセットのブロックサイズは、少なくとも、アンロードするファイルのデータストレージブロックサイズに 16 バイトを加えた値である必要があります。

**TEMPSIZE：一時ストレージのサイズ**

TEMPSIZE はファイルの中間データセットのサイズを指定します。サイズはシリンダ単位またはブロック単位（後ろに "B" を付加）で指定できます。

中間データセットは、プロテクションログのすべてのデータストレージブロックを格納するのに十分な大きさを必要とします。最悪の場合は、ファイルのデータストレージブロック数と同じ数のブロックが必要になります。ただし、PLOG データセットより大きくする必要は

ありません。中間データセットが小さすぎると、ADAULDはエラーコード136（temp data set too small）を返します。

### TEST：構文テスト

このパラメータは、実際に処理を行わずに構文をテストします。指定したパラメータの構文だけがテストされ、値や変数の正当性はテストされません。

### UTYPE：ユーザータイプ

アンロード処理で有効とするユーザータイプを指定します。

- EXU（デフォルト）が指定されると、ファイルの更新はできませんが、他のユーザーがファイルを読み取ることはできます。
- EXFが指定されると、ファイルはADAULDでのみ使用可能となり、他のユーザーはファイルの読み取りも書き込みもできなくなります。

## 例

---

### 例 1

```
ADAULD FILE=6
```

ファイル6をアンロードします。レコードはデータストレージ内に物理的に位置している順番でアンロードされます。

### 例 2

```
ADAULD FILE=6, SORTSEQ=AA
```

ファイル6をアンロードします。レコードをアンロードする順序を制御するために、ディスクリプタ AA の値が使用されます。

### 例 3

```
ADAULD FILE=6, SORTSEQ=ISN
```

ファイル6をアンロードします。レコードはISNの昇順にアンロードされます。

### 例 4

```
ADAULD FILE=6, SORTSEQ=ISN, STARTISN=10000
```

ファイル6をアンロードします。レコードはISNの昇順にアンロードされます。ISNが10000以上のレコードだけがアンロードされます。

### 例 5

```
ADAULD FILE=6, SORTSEQ=AB, MODE=SHORT
```

ファイル 6 をアンロードします。レコードをアンロードする順序を制御するために、ディスクリプタ AB の値が使用されます。ノーマルインデックスおよびアップパーインデックスの作成に使用されたエントリはアンロードされません。出力されるフィールド定義テーブル (FDT) からすべてのディスクリプタ情報が消去されます。

#### 例 6

```
ADAULD FILE=6,SELCRIT='AA,1,S,AA,2.',SELVAL='AMM'
```

ファイル 6 をアンロードします。AA=A から MM のレコードだけがアンロードされます。レコードは ISN 順に返されます。

#### 例 7

```
ADAULD FILE=6,UTYPE=EXF
```

ファイル 6 をアンロードします。ユーザータイプは EXF が指定されているため、アンロード処理中にファイルがロックされ、他のユーザーによるファイルの読み取りおよび更新を防ぎます。



# 189 ADAULD 入力処理

---

- 入力セーブテープの処理 ..... 1092

ADAULDは、Adabas ファイルを次のいずれかの対象からアンロードするために使用されます。

- データベース
- セーブテープ (SAVETAPE キーワードを指定した場合)

## 入力セーブテープの処理

---

セーブテープを入力として使用する場合は、DD/SAVE シーケンシャルファイルが対象になります。オンラインおよびオフラインで作成したデータベースセーブテープおよびファイルセーブテープを使用できます。セーブテープは ADASAV バージョン 5.1 以上で作成したものである必要があります。

セーブテープ上の DBID が、ADAULD の実行に指定した ADARUN DBID と一致する必要があります。

ファイルにハイパーディスクリプタが定義されている場合は、対応するハイパー出口が ADARUN パラメータで ADAULD に対して指定されている必要があります。ハイパー出口ルーチンを使用できないときは、MODE=SHORT を指定してファイルをアンロードする必要があります。ハイパー出口の詳細については、『*Adabas DBA リファレンスマニュアル*』を参照してください。

オンラインセーブテープの場合は、次の条件があります。

- 対応するプロテクションログが DD/PLOG シーケンシャル入力データセットとして必要です。
- このプロテクションログには、データストレージブロックの中間ストレージとして中間 (DD/TEMPR1) データセットが必要です。TEMPSIZE および TEMPDEV を指定する必要があります。

ADASAV バージョン 5.1 で作成したオンラインセーブテープを使用する場合は、追加パラメータ PLOGNUM と SYN1 または SYN4 を指定しなければなりません。

- PLOGNUM は、ADASAV SAVE オペレーションがアクティブの間に使用したニュークリアスプロテクションログ番号です。
- SYN1 または SYN4 は、対応する ADASAV SAVE を開始した SYN1 または SYN4 チェックポイント含むブロックの番号です。

ADASAV バージョン 5.2 以上で作成したオンラインセーブテープは、この情報を含んでいます。PLOGNUM SYN1 または SYN4 を指定すると、テープ情報は書き換えられます。

ADAULD ユーティリティプロトコルでは、見つかったセーブテープの種類を示す短いヘッダーを DD/DRUCK に表示します。このヘッダーには作成日時、作成に使用した ADASAV のバージョン、セーブテープで見つかったデータベース ID が含まれます。オンラインセーブテープの場合は、対応するプロテクションログのセッション番号と SYN1/SYN4 チェックポイントのブロック番号も含まれます。

```
A D A R E P      Vv.r   SMs      DBID = nnnnn  STARTED                yyyy-mm-dd   hh:mm:ss
```

```
PARAMETERS:
```

```
-----
```

```
ADAULD UNLOAD FILE=3, SAVETAPE
```

```
*****
*
* UNLOAD FROM                ONLINE DATABASE SAVE          *
* CREATED AT                 yyyy-mm-dd  hh:mm:ss          *
* BY ADASAV VERSION         V vr                          *
* DBIB                       nnnnn                        *
* DSID                       1 / 0 /   yyyy-mm-dd  hh:mm:ss *
* PLOG SESSION NR           17                            *
* SYN1 BLOCK NR             137                            *
*
*****
```

ADAULD は、最初にファイルコントロールブロック (FCB) とファイル定義テーブル (FDT) をセーブテープから読み取ります。次に、以下の処理を行います。

- オフラインセーブテープに対しては、テープをスキャンしてファイルのデータストレージ RABN を検索し、データストレージレコードを抽出します。次に、各データストレージレコードについて、FDT に従ってディスクリプタ値を生成します。
- オフラインセーブテープに対しては、プロテクションログをスキャンして、関連ファイルの各データストレージブロックの最新バージョンを中間データセットにコピーします。中間データセット上のデータストレージブロックの位置は、メインメモリのディレクトリに記憶されます。その後、セーブテープをスキャンして、ファイルのデータストレージブロックを検索します。より新しいバージョンのデータストレージブロックが中間データセットに存在する場合は、DD/OUT1 または DD/OUT2 に実際にアンロードされます。この場合は、並行使用できるテープユニットが 2 つ必要です。ADASAV RESTONL でセーブテープとプロテクションログを連結することはできません。

ADAULD は DD/SAVE および DD/PLOG 入力データセットをオープンした後、照合を行って入力テープが正しいことを確認します。セーブテープが正しくない場合、ADAULD はエラー 134

(invalid save tape supplied) を表示して終了します。プロテクションログテープが正しくない場合は、エラー 135 (invalid protection log supplied) を表示して終了します。

# 190 ADAULD 出力処理

ADAULDは指定された順序でレコードをアンロードします。アンロードされたレコードはシーケンシャルデータセット DD/OUT1 と DD/OUT2 のいずれかまたは両方に書き込まれます。これらの出力データセットへの書き込みは、ユーザー出口9によって制御されます。

出力されるレコードは、MODE=SHORTオプションが使用されない場合は、ADACMPユーティリティによって作成される出力と同じフォーマットです。MODE=SHORTが指定されると、ノーマルインデックスおよびアップパーインデックスに必要なディスクリプタエントリが省略され、ディスクリプタ情報が Adabas FDT から削除されます。レコードの ISN は圧縮データレコードの直前に付きます。ISN は4バイトの2進数として与えられます。

DDISN パラメータを指定すると、ADAULD はアンロードした ISN のリストをシーケンシャル出力ファイル DD/ISN に書き込みます。DD/OUT1 または DD/OUT2 の片方または両方に書き込まれる ISN のスーパーセットを含む DD/ISN ファイルが1つだけ作成されます。ユーザ出口9に拒否された ISN は DD/ISN に書き込まれません。

DD/ISN は、アンロードされたレコードを削除するために ADALOD UPDATE の入力として使用できるように、構造化されています。

DD/ISN に書き込まれた ISN の数は、DD/DRUCK ユーティリティプロトコルの ADAULD 統計に表示されます。

## A D A U L D Statistics

-----

Number of Output Data Sets	=	1
Number of Requested Records	=	16777215
STARTISN	=	0
Options	=	DVT
Unload Sequence	=	PHYS SEQ
Number of Records Read	=	307
Number of Records Written	=	307

Number of Record Segments Read	=	777
Number of Record Segments Written	=	777
Records Written to DDOUT1	=	307
Records Written to DDOUT2	=	0
Records Rejected by USEREXIT-9	=	0
Number of ISNS Written to DDISN	=	307

DD/ISN に書き込まれる ISN の数は常に、読み取られたレコードの数からユーザー出口 9 に拒否されたレコードの数を引いた値になります。

DD/ISN ファイルの ISN はアプリケーションによって認識される ISN、つまり、データストレージレコードに格納された内部 ISN に MINISN-1 を加えた値になります。

# 191 ADAULD ユーザー出口 9

---

ユーザー出口9が存在する場合は、レコードが出力データセットに書き込まれる直前に、選択したレコードごとに呼び出されます。ユーザー出口にはレコードアドレスが提供されており、次のようなアクションコードを返します。

1	レコードを DD/OUT1 に書き込む。
2	レコードを DD/OUT2 に書き込む。
3	レコードを DD/OUT1 と DD/OUT2 に書き込む。
I	このレコードを無視する。

上記のデータセットは同じブロックサイズである必要があります。ユーザー出口の詳細については、『*Adabas DBA リファレンスマニュアル*』を参照してください。

---

# 192

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	1100
▪ z/OS .....	1102
▪ z/VM .....	1104
▪ VSE .....	1106

このセクションでは、ADAULD を BS2000、z/OS、z/VM、および VSE の各システムで実行するために必要なジョブ制御情報について説明し、システムごとのジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アンロードデータ	DDOUT1	テープ/ディスク	ADAULD によって出力 (注参照)
アンロードデータ	DDOUT2	テープ/ディスク	ADAULD によって出力 (注参照)
アンロード ISN	DDISN	テープ/ディスク	DDISN の場合に必要
セーブテープ	DDSAVE	テープ/ディスク	SAVETAPE の場合に必要
シーケンシャル PLOG	DDPLOG	テープ/ディスク	オンラインセーブテープの場合に必要
一時エリア	DDTEMPR1	ディスク	オンラインセーブテープの場合に必要
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI の場合に必要
ADARUN パラメータ	DDCARD		オペレーションマニュアル
ADAULD パラメータ	SYSDTA/ DDKARTE		
ADARUN メッセージ	SYSOUT/ DDPRINT		メッセージおよびコード
ADAULD メッセージ	SYSLST/ DDDRUCK		メッセージおよびコード

 **Note:** DDOUT1 および DDOUT2 のデータセットが同じブロックサイズでない場合は、ADAULD エラーが発生します。DDOUT2 は、NUMOUT=2 が指定された場合にのみ必要とされます。

### ADAULD JCL の例 (BS2000)

#### データベースからのアンロード

SDF フォーマットの場合：

```

/.ADAULD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A U L D NON-SAVETAPE FUNCTIONS
/REMARK *
/DELETE-FILE ADAyyyyyy.OUT1
/SET-JOB-STEP
    
```

```

/CREATE-FILE ADAyyyyy.OUT1,PUB(SPACE=(480,48))
/SET-JOB-STEP
/ASS-SYSLST L.ULD
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDOUT1,ADAyyyyy.OUT1
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAULD,DB=yyyyy,IDTNAME=ADABAS5B
ADAULD FILE=1,SORTSEQ=AA
/LOGOFF SYS-OUTPUT=DEL

```

ISP フォーマットの場合：

```

/.ADAULD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A U L D NON-SAVETAPE FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.ULD
/FILE ADA.MOD ,LINK=DDLIB
/FILE ADAyyyyy.OUT1 ,LINK=DDOUT1 ,SPACE=(480,48)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAULD,DB=yyyyy,IDTNAME=ADABAS5B
ADAULD FILE=1,SORTSEQ=AA
/LOGOFF NOSPOOL

```

オフラインセーブテープからのアンロード

SDF フォーマットの場合：

```

/.ADAULD LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * A D A U L D SAVETAPE FUNCTION
/REMARK *
/DELETE-FILE ADAyyyyy.OUT1
/SET-JOB-STEP
/CREATE-FILE ADAyyyyy.OUT1,PUB(SPACE=(480,48))
/SET-JOB-STEP
/ASS-SYSLST L.ULD
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDOUT1,ADAyyyyy.OUT1
/SET-FILE-LINK DDSAVE,ADAyyyyy.SAVE
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAULD,DB=yyyyy,IDTNAME=ADABAS5B

```

```
ADAULD FILE=1,SAVETAPE
/LOGOFF SYS-OUTPUT=DEL
```

ISP フォーマットの場合：

```
/.ADAULD LOGON
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * A D A U L D SAVETAPE FUNCTION
/REMARK *
/SYSFILE SYSLST=L.ULD
/FILE ADA.MOD ,LINK=DDLIB
/FILE ADAyyyyy. OUT1 ,LINK=DDOUT1 ,SPACE=(480,48)
/FILE ADAyyyyy.SAVE ,LINK=DDSAVE
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAULD,DB=yyyyy, IDTNAME=ADABAS5B
ADAULD FILE=1,SAVETAPE
/LOGOFF NOSPOOL
```

z/OS

データセット	DD名	ストレージ	追加情報
アンロードデータ	DDOUT1	テープ/ディスク	ADAULD によって出力 (注参照)
アンロードデータ	DDOUT2	テープ/ディスク	ADAULD によって出力 (注参照)
アンロード ISN	DDISN	テープ/ディスク	DDISN の場合に必要
セーブテープ	DDSAVE	テープ/ディスク	SAVETAPE の場合に必要
シーケンシャル PLOG	DDPLOG	テープ/ディスク	オンラインセーブテープの場合に必要
一時エリア	DDTEMPR1	ディスク	オンラインセーブテープの場合に必要
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI の場合に必要
ADAULD メッセージ	DDDRUCK	プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAULD パラメータ	DDKARTE	リーダー	



**Note:** DDOUT1 および DDOUT2 のデータセットが同じブロックサイズでない場合は、ADAULD エラーが発生します。DDOUT2 は、NUMOUT=2 が指定された場合にのみ必要とされます。

## ADAULD JCL の例 (z/OS)

### ファイルのアンロード

```
//ADAULD      JOB
//*
//*      ADAULD:
//*      UNLOAD A FILE
//*
//ULD          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
//SYSUDUMP    DD   SYSOUT=X

//DDOUT1      DD
DISP=( ,CATLG),DSN=EXAMPLE.DByyyyyy.OUT1,UNIT=DISK, <===
//              VOL=SER=DISK01,SPACE=(TRK,(200,10),RLSE)
//DDCARD      DD   *
ADARUN PROG=ADAULD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
//DDKARTE     DD   *
ADAULD FILE=1,SORTSEQ=AA

00000100
/*
```

この例については、MVSJOBS データセットの ADAULD を参照してください。

### オフラインで作成したセーブテープからのファイルのアンロード

```
//ADAULDS     JOB
//*
//*      ADAULD:
//*      UNLOAD A FILE FROM AN OFFLINE SAVE TAPE
//*
//ULD          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD           <=== ADABAS LOAD
//*
//DDASSOR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASSOR1 <=== ASSO
//DDDATAR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDWORKR1    DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.WORKR1 <=== WORK
//DDSAVE      DD   DISP=SHR,DSN=EXAMPLE.DByyyyyy.SAVE     <=== SAVE DATASET
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
```

```
//SYSUDUMP DD SYSOUT=X
//DDOUT1 DD DISP=(,CATLG),DSN=EXAMPLE.DByyyyy.OUT1,UNIT=DISK, <===
// VOL=SER=DISK01,SPACE=(TRK,(200,10),RLSE)
//DDCARD DD *
ADARUN PROG=ADAULD,MODE=MULTI,SVC=xxxx,DEVICE=dddd,DBID=yyyyyy
/*
//DDKARTE DD *
ADAULD FILE=1,SAVETAPE

00000100
/*
```

この例については、MVSJOBS データセットの ADAULDS を参照してください。

## z/VM

データセット	DD名	ストレージ	追加情報
アンロードデータ	DDOUT1	テープ/ディスク	ADAULD によって出力 (注参照)
アンロードデータ	DDOUT2	テープ/ディスク	ADAULD によって出力 (注参照)
アンロード ISN	DDISN	テープ/ディスク	DDISN の場合に必要
セーブテープ	DDSAVE	テープ/ディスク	SAVETAPE の場合に必要
シーケンシャル PLOG	DDPLOG	テープ/ディスク	オンラインセーブテープの場合に必要
一時エリア	DDTEMPR1	ディスク	オンラインセーブテープの場合に必要
リカバリログ (RLOG)	DDRLOGR1	ディスク	ADARAI の場合に必要
ADAULD メッセージ	DDDRUCK	ディスク/端末/プリンタ	メッセージおよびコード
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAULD パラメータ	SYSIPT	ディスク/端末/リーダー	



**Note:** DDOUT1 および DDOUT2 のデータセットが同じブロックサイズでない場合は、ADAULD エラーが発生します。DDOUT2 は、NUMOUT=2 が指定された場合にのみ必要とされます。

## 例 (z/VM)

### データベースからのアンロード

```
DATADEF DDOUT1,DSN=ADABASV.ULD1,UNIT=181,VOL=ULDF1
DATADEF DDPRINT,DSN=ADAULD.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDRUCK,DSN=ADAULD.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNULD.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAULD.CONTROL,MODE=A
ADARUN
```

### RUNULD CONTROL A1 の内容

```
ADARUN PROG=ADAULD,DEVICE=dddd,DB=yyyyy
```

### ADAULD CONTROL A1 の内容

```
ADAULD FILE=1, SORTSEQ=AA
```

### オフラインセーブテープからのアンロード

```
DATADEF DDOUT1,DSN=ADABASV.ULD1,UNIT=181,VOL=ULDF1
DATADEF DDSAVE,DSN=ADABASV.ULD1,UNIT=181,VOL=ULDF1
DATADEF DDPRINT,DSN=ADAULD.DDPRINT,MODE=A
DATADEF DUMP,DUMMY
DATADEF DDRUCK,DSN=ADAULD.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNULD.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAULD.CONTROL,MODE=A
ADARUN
```

### RUNULD CONTROL A1 の内容

```
ADARUN PROG=ADAULD,DEVICE=dddd,DB=yyyyy
```

### ADAULD CONTROL A1 の内容

```
ADAULD FILE=1, SAVETAPE
```

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アンロードデータ	OUT1	テープ ディスク	SYS010 注 1 参照	ADAULD によって出力 (注 2 参照)
アンロードデータ	OUT2	テープ ディスク	SYS011 注 1 参照	ADAULD によって出力 (注 2 参照)
アンロード ISN	ISN	テープ ディスク	SYS012 注 1 参照	DDISN の場合に必要
セーブテープ	SAVE	テープ ディスク	SYS013 注 1 参照	SAVETAPE の場合に必要
シーケンシャル PLOG	PLOG	テープ ディスク	SYS014 注 1 参照	オンラインセーブテープの場合 に必要
一時エリア	TEMPR1	ディスク	注 1 参照	オンラインセーブテープの場合 に必要
リカバリログ (RLOG)	RLOGR1	ディスク	注 1 参照	ADARAI の場合に必要
メッセージ	SYSLST	プリンタ		メッセージおよびコード
ADARUN パラメータ	SYSRDR CRD	リーダー/ テープ/ディスク		オペレーションマニュアル
ADAULD パラメータ	SYSIPT	リーダー		



### Notes:

1. すべてのプログラマ論理ユニットを使用できます。
2. OUT1 および OUT2 のデータセットが同じブロックサイズでない場合は、ADAULD エラーが発生します。OUT2 は、NUMOUT=2 が指定された場合にのみ必要です。

### ADAULD JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

データベースからのファイルのアンロード

```
* $$ JOB JNM=ADAULD,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D

// JOB ADAULD
*      UNLOAD A FILE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS004,DISK,VOL=vvvvvv,SHR
// DLBL OUT1,'EXAMPLE.ADA99.OUT1'
// EXTENT SYS004,,,,ssss,nnnn
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAULD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAULD FILE=1, SORTSEQ=AA
/*
/&
* $$ EOJ
```

この例については、メンバ ADAULD.X を参照してください。

オフラインで作成したセーブテープからのファイルのアンロード

```
* $$ JOB JNM=ADAULDS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAULDS
*      UNLOAD A FILE FROM AN OFFLINE SAVE TAPE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// ASSGN SYS004,DISK,VOL=vvvvvv,SHR
// DLBL OUT1,'EXAMPLE.ADA99.OUT1'
// EXTENT SYS004,,,,ssss,nnnn
// ASSGN SYS013,TAPE
// PAUSE MOUNT LOAD SAVE FILE ON TAPE cuu
// TLBL SAVE,'EXAMPLE.DByyyyy.SAVE'
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAULD,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAULD FILE=1, SAVETAPE
/*
/&
* $$ EOJ
```

この例については、メンバ ADAULDS.X を参照してください。



# 193 ADAVAL ユーティリティ：データベースの整合性チェック

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- `VALIDATE`：データストレージとアソシエータの整合性チェック
- ADAVAL 出力例
- JCL/JCS の必要項目と指定例



# 194 機能概要

---

ADAVAL（整合性チェック）ユーティリティは、Adabas データベース内の任意のファイルまたは全ファイルの整合性チェックを行います。チェックポイントとセキュリティファイルはチェック対象ではありません。

ADAVAL はデータストレージのレコードのディスクリプタ値とアソシエータに格納された対応する値との整合性をチェックします。したがって、アソシエータとデータストレージの同期がとれていることと、アソシエータ内に値の不足がないことが保証されます。

ADAVAL を実行する前に、ADAICK ユーティリティを使用して、インバーテッドリストの整合性をチェックする必要があります。



**Note:** ADAICK を実行したときにエラーが発生した場合は、ADAICK エラーの原因が修正されるまで、ADAVAL を実行しないでください。ADAVAL のこの実行制限は、ADAICK WARNING-163 を除くすべての ADAICK エラーに当てはまります。

ADAVAL を実行するときは、Adabas ニュークリアスがアクティブである必要があります。ADAVAL は、整合性チェックの対象となるすべてのファイルに対して EXF（排他使用）ステータスを割り当て、他のユーティリティおよび他のユーザーが使用できないようにします。ADAVAL で指定されたファイルが現在使用中の場合には、エラーメッセージが発行され、処理が停止します。ADAVAL でエラーが発生した場合には、コンディションコード 4 が返されます。

ADAVAL は、比較された全フィールドおよび整合性チェックで拒否された ISN のリストを SYSOUT (DD/DRUCK) に出力します。通常の ADAVAL 出力を「[ADAVAL 出力例](#)」に示します。

## 機能概要

必要に応じて、拒否された ISN をシーケンシャルデータセット (DD/FEHL) に出力することもできます。DD/FEHL 内の最初のレコードは、常に次のようになります。

バイト	説明
0~1	バイナリ形式でのレコード長 (X'0012' など)
2~3	常に 0 が入る (X'0000' など)
4~9	プログラム ID (C'ADAVAl' など)
10~13	4 バイトのパック形式で表したユリウス暦による日付。フォーマットは YYYYYDDDF ("F" = B'1111') です。
14~17	4 バイトのパック形式で表した時刻。フォーマットは hhmmsssthh ( $t=1/10$ 秒、 $h=1/100$ 秒) です。

残りすべての DD/FEHL レコードは次のフォーマットになります。アスタリスク (\*) が付いた項目は、通常の SYSOUT および DD/DRUCK 出力にも含まれます。

バイト	説明				
0~1	バイナリ形式でのレコード長 (X'0012' など)				
2~3	常に 0 が入る (X'0000' など)				
4~5*	Adabas ファイル番号 (バイナリ形式)				
6*	フラグバイト： <table border="1" style="margin-left: 20px;"> <tr> <td>C'-</td> <td>値がありません。</td> </tr> <tr> <td>C'+</td> <td>値が正しくありません。</td> </tr> </table>	C'-	値がありません。	C'+	値が正しくありません。
C'-	値がありません。				
C'+	値が正しくありません。				
7	ゼロに設定				
8~11	ISN (バイナリ形式)				
12~13*	フィールド定義テーブル (FDT) に格納されているディスクリプタ名				
14*	ディスクリプタ値の長さ (バイナリ形式)				
15以降*	ディスクリプタ値				

# 195 VALIDATE：整合性チェックデータストレージと アソシエータ

---

- 基本パラメータ ..... 1114
- オプションパラメータ ..... 1115

```

ADAVAL VALIDATE FILE=file-list
                    SORTSIZE=size
                    TEMPSIZE=size
                    [CODE = cipher-key]
                    [DESCRIPTOR = 'descriptor-list' ]
                    [LPB = { prefetch-buffer-length | based-on-ADARUN-lu } ]
                    [LRECL = { record-buffer-length | 4000 } ]
                    [LWP = { work-pool-size | 1048576 } ]
                    [NOUSERABEND]
                    [PASSWORD = password ]
                    [SORTDEV = { device-type | ADARUN-device } ]
                    [TEMPDEV = { device-type | ADARUN-device } ]

```

VALIDATE 機能は、データストレージの内容をアソシエータの値に対してチェックします。この処理は、インデックス内の対応する各値に対してチェックされる DVT を作成するコマンドを発行して行います。



**Note:** ADAVAL VALIDATE はチェックポイントファイルやセキュリティファイルには実行できません。

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

### FILE：整合性チェックを行うファイル

FILEは、1つ以上のAdabas ファイル番号またはその範囲を指定します。最大1000個のファイルを指定可能です。

ファイルリストの継続は下記のように行われます。

```

ADAVAL VALIDATE FILE=1-10,15
ADAVAL FILE=13,31-35

```

ADAVAL では FILE パラメータが指定されるたびにファイルのリストが連結されます。

### SORTSIZE：ソートエリアのサイズ

SORTSIZE は、ソートデータセットに使用できるブロックまたはシリンダの数を指定します。ブロックでの指定の場合、値の後に "B" を伴う必要があります ("2000B" など)。ブロック値は、自動的に次のフルシリンダへの切り上げが行われます。SORTSIZE 値の見積りの詳細については、『Adabas DBA リファレンスマニュアル』を参照してください。

**TEMPSIZE**：一時ストレージエリアサイズ

TEMPSIZEは、中間データセットに使用できるブロックまたはシリンダの数を指定します。ブロックでの指定の場合、値の後に "B" を伴う必要があります ("2000B" など)。ブロック値は、自動的に次のフルシリンダへの切り上げが行われます。TEMPSIZE 値の見積りの詳細については、「ADALOD」の説明の「LOAD ファイルのスペース割り当て」のセクションを参照してください。

## オプションパラメータ

**CODE**：サイファコード

整合性チェックを行うファイルが暗号化されている場合は、CODEパラメータが必要です。

**DESCRIPTOR**：整合性チェックを行うディスクリプタ

DESCRIPTOR パラメータは、整合性チェック処理を1つのディスクリプタフィールドに限定します。これにより、アソシエータが非常に大きい場合や、特定のディスクリプタを評価する必要がある場合に、整合性チェックの実行を制限できるようになります。DESCRIPTOR が指定されない場合、ADAVAL では該当するすべてのディスクリプタフィールドがチェックされます。

DESCRIPTOR の使用例を次に示します。

```
ADAVAL VALIDATE  
FILE=5,DESCRIPTOR='AA,CC,BB'
```

**LPB**：プリフェッチバッファサイズ

LPBは、内部プリフェッチバッファの大きさをバイト単位で指定するものです。指定可能な最大値は、32,760 バイトです。デフォルト値は、ADARUN LU の現在の値によって異なります。LU の値が小さすぎる場合は、ADAVAL VALIDATE によって LPB 値が減らされることがあります。

**LRECL**：最大ディスクリプタ値

LRECLは、整合性チェック対象ファイルのレコード内の全ディスクリプタ値の最大長さを指定します。ADAVALではこの長さを使用して一時レコードバッファを作成します。LRECL 値が小さすぎる場合は、サイズを超えるレコードが検出されると、レスポンスコード 53 が発生します。LRECL のデフォルトは 4000 バイトで、指定可能な最大値は 32760 バイトです。

**LWP**：ワークプールサイズ

LWPは、ディスクリプタ値のソートに使用されるワークプールの大きさを指定します。この値は、数値のみでバイト単位で指定するか、あるいは数値の後に "K" を付けてキロバイト単位で指定します。値が指定されない場合は、デフォルトで1048576バイト（または1024K）が設定されます。ただし、ディスクリプタが非常に長い、またはディスクリプタの数が著しく多いファイルに対してADAVALの実行時間を短くする場合には、より大きな値をLWPに設定します。ソートデータセットで問題が発生するのを防ぐために、比較的小さなファイルをチェックするときは、LWPの値を小さく設定する必要があります。

最小ワークプールサイズは、SORT データセットのデバイスタイプに応じて異なります。

ソートデバイス	最小 LWP	最小 LWP
	バイト	キロバイト
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

#### **NOUSERABEND**：アベンドなし終了

機能の実行中にエラーになると、ユーティリティは、エラーメッセージを出力してユーザーアベンド 34（ダンプあり）またはユーザーアベンド 35（ダンプなし）で終了します。

NOUSERABENDを指定すると、ユーティリティは、エラーメッセージを出力してもアベンドしません。代わりに、メッセージ"utility TERMINATED DUE TO ERROR CONDITION"（エラーのためユーティリティを終了します）を表示してコンディションコード 20 で終了します。

#### **PASSWORD**：ファイルパスワード

整合性チェックを行うファイルがパスワード保護されている場合はこのパラメータが必要です。

#### **SORTDEV**：ソートデバイスタイプ

SORTDEV パラメータは、ADAVAL がディスクリプタ値をソートするために使用するソートデータセットに使われるデバイスタイプを示します。ソートデータセットのサイズは SORTSIZE で指定します。このパラメータは、使用するデバイスタイプが ADARUNDEVICE パラメータの指定と異なる場合にのみ必要です。

#### **TEMPDEV**：一時ストレージデバイスタイプ

TEMPDEV パラメータは、ADAVAL が中間データを格納するために使用する中間データセットに使われるデバイスタイプを示します。データセットのサイズは TEMPSIZE パラメータで指定します。このパラメータは、使用するデバイスタイプが ADARUNDEVICE パラメータの指定と異なる場合にのみ必要です。

# 196

## ADAVAL 出力例

ADAVAL 出力では、すべてのデータストレージおよびアソシエータのエントリとそのステータスをファイルおよびディスクリプタごとに示したSYSOUT (DD/DRUCK) テーブルが提供されます。以下に ADAVAL VALIDATE の出力例を示します。

FILE	DE	F	ISN	DE-VALUE	
1	AA	***	NO	INCONSISTENCIES	***
1	BA	-	35	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	173	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	471	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	534	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	597	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	622	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	658	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	717	07C6D935 C5D4C1D5	*.FREEMAN*
1	BA	-	152	05D4C5E8 C5D9	*.MEYER*
1	BA	+	153	05D4C5E8 C5D9	*.MEYER*
1	BB	***	NO	INCONSISTENCIES	***
1	CA	***	NO	INCONSISTENCIES	***
1	CB	***	NO	INCONSISTENCIES	***
1	CC	***	NO	INCONSISTENCIES	***
1	CD	***	NO	INCONSISTENCIES	***
1	PA	***	NO	INCONSISTENCIES	***

ここでは次の内容を表示しています。

- F (フラグ) 列のダッシュ (-) は指定されたデータストレージディスクリプタにインバーテッドリストエントリが存在しないことを示し、プラス符号 (+) はアソシエータのインバーテッドリストエントリが正しくないことを示します。
- DE-VALUE 列には、圧縮されたディスクリプタ値が最初に 16 進数、次に英数字で表示されます。



**Note:** ディスクリプタの整合性チェックが成功するたびに "\*\*\* NO INCONSISTENCIES \*\*\*" エントリが表示されます。

# 197 JCL/JCS の必要項目と指定例

---

▪ ASSO、DATA、WORK データセット .....	1120
▪ ユーザー出口を使用した照合 .....	1120
▪ 大きなファイルのソート .....	1121
▪ BS2000 .....	1121
▪ z/OS .....	1122
▪ z/VM .....	1123
▪ VSE .....	1124

このセクションでは、ADAVAL を BS2000、z/OS、z/VM、および VSE の各システムで実行するために必要なジョブ制御情報について説明し、システムごとのジョブストリームを示します。

このchapterでは、次のトピックについて説明します。

## ASSO、DATA、WORK データセット

ASSO、DATA、WORK の各データセットは、Adabas がマルチユーザーモード (ADARUN MODE=MULTI) で実行される場合は指定する必要はありません。これは、ADAVAL ではこれらのワークセットをオープンしないためです。データベースがアクティブなときは、ADAVAL は特別な Adabas コマンドを使用してデータベースに関する情報を受け取ります。ただし、データベースがアクティブでないときには ADAVAL で問題が発生します。

Adabas がシングルユーザーモード (ADARUN MODE=SINGLE) で実行される場合は、ASSO、DATA、WORK の各データセットを指定する必要があります。

## ユーザー出口を使用した照合

ADAVAL 実行中に照合ユーザー出口を使用するには、ユーティリティの実行時に ADARUN CDXnn パラメータを指定する必要があります。

ユニバーサルエンコーディングサブシステム (UES) と合わせて使用する場合の照合ディスクリプタユーザー出口パラメータのフォーマットは、次のようになります。

```
ADARUN CDXnn= exit-name
```

ここでは次の内容を表しています。

nn	照合ディスクリプタ出口の番号。01 から 08 までの 2 桁の 10 進数。
exit-name	照合ディスクリプタ出口で制御を持つユーザールーチン名。最大 8 文字。

照合ディスクリプタ出口に指定できるプログラムは 1 つだけです。最大 8 個の照合ディスクリプタ出口を任意の順序で指定できます。詳細は『Adabas DBA リファレンスマニュアル』を参照してください。

## 大きなファイルのソート

大きなファイルをソートする場合は、ソートデータセットが2つのボリュームを占有しているか、ソートデータセットが2つ指定されている場合に、パフォーマンスを向上させることができます。両方のデータセットが同一のデバイスタイプ上に存在し（SORTDEV パラメータ）、各データセットは SORTSIZE パラメータで指定したサイズの正確に半分の大きさでなければなりません。

### BS2000

データセット	リンク名	ストレージ	追加情報
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大きいファイルを使用し、ソートエリアを2つのボリュームに分割する場合（注参照）
一時エリア	DDTEMPR1	ディスク	
ADARUN パラメータ	SYSDTA/DDCARD		オペレーションマニュアル
ADAVAL パラメータ	SYSDTA/DDKARTE		
ADARUN メッセージ	SYSOUT/DDPRINT		メッセージおよびコード
ADAVAL メッセージ	SYSLST/DDDRUCK		メッセージおよびコード
拒否データ	DDFEHL	テープ/ディスク	

### ADAVAL JCL の例 (BS2000)

SDF フォーマットの場合：

```

/. ADAVAL LOGON
/ MODIFY-TEST-OPTIONS DUMP=YES
/ REMARK *
/ REMARK * A D A V A L ALL FUNCTIONS
/ REMARK *
/ DELETE-FILE VAL.FEHL
/ SET-JOB-STEP
/ CREATE-FILE VAL.FEHL,PUB(SPACE=(48,48))
/ SET-JOB-STEP
/ ASS-SYSLST L.VAL
/ ASS-SYSDTA *SYSCMD
/ SET-FILE-LINK DDLIB,ADAvrs.MOD
/ SET-FILE-LINK DDSORTR1,ADAvyyyyy.SORT
/ SET-FILE-LINK DDTEMPR1,ADAvyyyyy.TEMP

```

```

/SET-FILE-LINK DDFEHL1,VAL.FEHL
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADAVL,DB=yyyyy,IDTNAME=ADABAS5B
ADAVL VALIDATE FILE=30,SORTSIZE=3,TEMPSIZE=5
/LOGOFF SYS-OUTPUT=DEL
    
```

ISP フォーマットの場合：

```

/.ADAVL LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * A D A V A L ALL FUNCTIONS
/REMARK *
/SYSFILE SYSLST=L.VAL
/FILE ADAyyyyy.TEMP ,LINK=DDTEMPR1
/FILE ADAyyyyy.SORT ,LINK=DDSORTR1
/FILE ADA.MOD,LINK=DDLIB
/FILE VAL.FEHL,LINK=DDFEHL,SPACE=(48,48)
/EXEC (ADARUN,ADA.MOD)
ADARUN PROG=ADAVL,DB=yyyyy,IDTNAME=ADABAS5B
ADAVL VALIDATE FILE=30,SORTSIZE=3,TEMPSIZE=5
/LOGOFF NOSPOOL
    
```

## z/OS

データセット	DD 名	ストレージ	追加情報
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大きいファイルを使用し、ソートエリアを2つのボリュームに分割する場合（注参照）
一時エリア	DDTEMPR1	ディスク	
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAVL パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAVL メッセージ	DDDRUCK	プリンタ	メッセージおよびコード
拒否データ	DDFEHL	テープ/ディスク	

## ADAVAL JCL の例 (z/OS)

```
//ADAVAL      JOB
//*
//*      ADAVAL: VALIDATE DATA BASE
//*
//VAL          EXEC PGM=ADARUN
//STEPLIB     DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDSORTR1    DD   DISP=OLD,DSN=EXAMPLE.DByyyyy.SORTR1 <=== SORT
//DDTEMPR1    DD   DISP=OLD,DSN=EXAMPLE.DByyyyy.TEMPR1 <=== TEMP
//DDDRUCK     DD   SYSOUT=X
//DDPRINT     DD   SYSOUT=X
//SYSUDUMP    DD   SYSOUT=X
//DDCARD      DD   *
ADARUN PROG=ADAVAL,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*

//DDKARTE     DD   *
ADAVAL VALIDATE FILE=1,TEMPSIZE=ttt,SORTSIZE=sss
/*
```

この例については、MVSJOBS データセットの ADAVAL を参照してください。

## z/VM

データセット	DD 名	ストレージ	追加情報
ソートエリア	DDSORTR1	ディスク	
ソートエリア	DDSORTR2	ディスク	大きいファイルを使用し、ソートエリアを2つのボリュームに分割する場合（注参照）
一時エリア	DDTEMPR1	ディスク	
ADARUN パラメータ	DDCARD	ディスク/端末/リーダー	オペレーションマニュアル
ADAVAL パラメータ	DDKARTE	ディスク/端末/リーダー	
ADARUN メッセージ	DDPRINT	ディスク/端末/プリンタ	メッセージおよびコード
ADAVAL メッセージ	DDDRUCK	ディスク/端末/プリンタ	
拒否データ	DDFEHL	テープ/ディスク	

例 (z/VM)

```
DATADEF DDPRINT , DSN=ADAVAL , DDPRINT , MODE=A
DATADEF DUMP , DUMMY
DATADEF DDDRUCK , DSN=ADAVAL . DDDRUCK , MODE=A
DATADEF DDTEMPR1 , DSN=ADABASVv . TEMP , VOL=TEMPV1
DATADEF DDSORTR1 , DSN=ADABASVv . SORT , VOL=SORTV1
DATADEF DDCARD , DSN=RUNVAL . CONTROL , MODE=A
DATADEF DDKARTE , DSN=ADAVAL . CONTROL , MODE=A
DATADEF DDFEHL , DSN=FILE030 . VALERR , MODE=A
ADARUN
```

RUNVAL CONTROL A1 の内容

```
ADARUN PROG=ADAVAL , DEVICE=dddd , DB=yyyyy
```

ADAVAL CONTROL A1 の内容

```
ADAVAL VALIDATE FILE=30 , SORTSIZE=3 , TEMPSIZE=5
```

VSE

ファイル	ファイル名	ストレージ	論理ユニット	追加情報
ソートエリア	SORTR1	ディスク		
ソートエリア	SORTR2	ディスク		大きなファイルを使用する場合は、ソートエリアを2つのボリュームに分割してください（注を参照）。
一時エリア	TEMPR1	ディスク*		
ADARUN パラメータ	- CARD CARD	リーダー テープ ディスク	SYSRDR SYS000 *	
ADAVAL パラメータ	-	リーダー	SYSIPT	ユーティリティ
ADARUN メッセージ	-	プリンタ	SYSLST	メッセージおよびコード
ADAVAL メッセージ	-	プリンタ	SYS009	メッセージおよびコード
拒否データ	FEHL	テープ ディスク	SYS014 *	

\* あらゆるプログラマ論理ユニットが使用できます。

## ADAVAL JCS の例 (VSE)

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

```
* $$ JOB JNM=ADAVAL,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAVAL
*      VALIDATE DATABASE
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAVAL,MODE=SINGLE,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAVAL VALIDATE FILE=1,TEMPSIZE=ttt,SORTSIZE=sss
/*
/&
* $$ EOJ
```

この例については、メンバ ADAVAL.X を参照してください。



# 198 ADAWRK ユーティリティ：ワークエリアリカバリレポート

---

ADAWRK ユーティリティを使用すると、WORK パート 1 の自動再スタートエリア内のレコードからレポートを生成できます。この情報は、データベースの自動再スタートが失敗し、データベースが起動しない場合に使用することができます。ADAWRK レポートのデータによって、次のことが可能になります。

- 時間のかかるデータベースのリストア／再生成（ADARES ユーティリティ、GENERATE 機能）を実行する必要があるかどうか判断できます。
- AREXCLUDE を使用して特定のファイルを自動再スタートから除外し、これらの単一ファイルのみをリストア／再生成することに利点があるかどうか判断できます。
- データベースを簡単に修復して、以前よりすばやく起動して機能するようできるかどうかを判断できます。

このchapterでは、次のトピックについて説明します。

- 機能概要
- ユーティリティ構文
- レポートの解説
- JCL/JCS の必要項目と指定例



# 199 機能概要

---

データベースの自動再スタートが失敗し、データベースが起動しない場合は、データベースを元の実行状態に戻すために何ができるかを把握する必要があります。このとき、損失データを最小限にし、失われた更新内容を回復するために十分な情報を得ることが必要です。ADAWRKユーティリティはこの判断に役立ちます。次のレポートを生成することができます。

- サマリレポートはデフォルトで生成され、WORK パート 1 の自動再スタートエリア内のデータの概要を示します。
- ファイルレポートはオプションのレポートで、WORK パート 1 の自動再スタートエリア内のデータの内訳をファイルごとに示します。
- トランザクションレポートはオプションのレポートで、WORK パート 1 の自動再スタートエリア内のデータの内訳をトランザクションごとに示します。

WORK パート 1 の自動再スタートエリアで見つかったチェックポイントレコードおよび関連データも出力することができます。このchapterには、すべてのレポートおよびチェックポイントレコードのサンプルおよび詳細な説明が記載されています。

ADAWRK ユーティリティでは、自動再スタート処理ロジックの一部として修正が必要になる可能性のあるトランザクションについてのみ、レポートします。

WORK パート 1 の自動再スタートエリアのすべてのレコード、および ADAWRK の実行で生成されたすべてのレポートは、コミュニケーション ID、ETID、ユーザー ID、およびファイル番号でフィルタリングすることができます。

Adabas データベースの自動再スタート処理の詳細については、「リカバリ／再スタート設計」を参照してください。



## 200 ユーティリティ構文

---

ADAWRK ユーティリティの構文を以下に示します。このchapterには、この構文を使用したサンプルのJCLが記載されています。

```
ADAWRK [ABEND34]  
[CHECKPOINT = {YES | NO }]  
[CMID = id1 [, id2 ] ... [, id24 ]]  
[ETID = etid1 [, etid2 ] ... [, etid32 ]]  
[FILES = fn1 [, fn2 ] ... [, fn64 ]]  
[FORCE = {YES | NO }]  
[LWP = nnnK | 1024K ]  
[NOPPT]  
[NOUSERABEND]  
[REPORTFILE = {YES | NO }]  
[SUMMARY = {NO | YES }]  
[TEST]  
[TIMEZONE = {MACHINE | LOCAL | {+ | -} nn }]  
[TRANSACTIONS = { NO | YES | DETAIL | FULL}]  
[USERID = id1 [, id2 ] ... [, id24 ]]
```

ADAWRK をパラメータなしで単独で指定すると、サマリレポートを生成することができます（デフォルトは SUMMARY=YES）。他のパラメータ値を任意で追加することによって、ADAWRK で生成されるレポートをカスタマイズすることができます。それぞれのパラメータを以下に示します。

### ABEND34

ABEND34パラメータを使用すると、ADAWRKユーティリティにエラーが発生したときに、ユーザーアベンド 35 をユーザーアベンド 34 に変更することができます。これにより、このユーティリティが異常終了したときにダンプが確実に出力されます。

### CHECKPOINT

CHECKPOINT パラメータを使用すると、WORK パート 1 の自動再スタートエリアで見つかったチェックポイントをADAWRKレポート出力に出力する必要があるかどうかを示すことができます。有効な値は "YES" または "NO" です。"YES" の値は見つかったチェックポイントレコードを出力することを示し、"NO" の値は出力しないことを示します。デフォルトは "NO" です。

### CMID

CMID パラメータを使用すると、32 バイトのコミュニケーション ID を 16 進形式で 24 個まで指定することができます。WORK パート 1 の自動再スタートエリアレコードの中で、CMID パラメータで指定された値に等しいコミュニケーション ID を持つもののみが、ADAWRK ユーティリティによって処理され、レポートに出力されます。

### ETID

ETID パラメータを使用すると、32 個以下の ETID を文字形式で指定することができます。ETID には 1~8 バイトの長さが必要です。ETID を指定すると、その ETID に対応する WORK パート 1 の自動再スタートエリアレコードのみが ADAWRK ユーティリティによって処理され、レポートに出力されます。

### FILES

FILES パラメータを使用すると、レポートに含める必要があるファイル番号を 64 個まで指定できます。FILES パラメータに示されたファイルに対応する WORK パート 1 の自動再スタートエリアレコードのみが、ADAWRK ユーティリティによって処理され、レポートに出力されます。ただし、FILES パラメータを指定しない場合は、デフォルトでデータベース内のすべてのファイルが処理されます。

必要に応じて、このパラメータにファイル番号の範囲を指定することができます。例えば、FILES=2-20 は、2~20 のファイル番号を持つすべてのファイルをユーティリティで処理する必要があることを示します。

### FORCE

FORCE パラメータを使用すると、WORK パート 1 の自動再スタートエリア内で矛盾が発生したときに、ADAWRK 処理をどのように実行する必要があるかを示すことができます。有効な値は "YES" または "NO" です。"YES" の値は ADAWRK ユーティリティでは、アベンドを発生させることなく、データの解釈を続行する必要があることを示し、"NO" の値は WORK パート 1 の自動再スタートエリア内の矛盾が発生すると、適切なメッセージを出力し、アベンドが発生してユーティリティが終了することを示します。

### LWP

LWP パラメータを使用すると、ADAWRK ユーティリティによって内部で使用されるワークプールのサイズを指定することができます。有効な値は 100K~1048576K (つまり 1GB) です。デフォルトは 1024K (1MB) です。LWP はキロバイト単位で指定する必要があります。"K" を指定しなかった場合はエラーが発生します。例えば、LWP=500K という指定は有効ですが、LWP=500 は有効ではありません。

**NOPPT**

NOPPT パラメータを使用すると、データベースのアソシエータ (ASSO) データセットがユーティリティによって開かれないようにすることができます。このパラメータを指定すると、ASSO データセットは開かれませんが、このパラメータを指定しない場合に、ユーティリティに ASSO データセットが与えられると、ユーティリティは PPT を使用します。

このパラメータによって、WORK データセットを利用できてもアソシエータを利用できない場合の対処法が提供されます。

**NOUSERABEND**

NOUSERABEND パラメータを使用すると、ユーザーアベンドが発生したときに、ユーティリティがリターンコード 20 で終了する必要があることを示すことができます。このパラメータを指定すると、アベンドが発生する代わりに、リターンコード 20 が出力されます。このパラメータを指定しない場合は、必要に応じてユーザーアベンドが発生します。

**REPORTFILE**

REPORTFILE パラメータを使用すると、ファイル統計レポートを出力する必要があるかどうかを示すことができます。有効な値は "YES" または "NO" です。"NO" はファイル統計レポートを出力しないことを示し、"YES" はファイル統計レポートを出力することを示します。

デフォルトは "NO" です。

REPORTFILE パラメータとともに CMID、FILES、USERID、または ETID のパラメータを指定すると、すべてのパラメータによって指定される基準をすべて満たす更新のみがファイル統計レポートに含まれます。

**SUMMARY**

SUMMARY パラメータを使用すると、サマリレポートを出力する必要があるかどうかを示すことができます。有効な値は "YES" または "NO" です。"NO" はサマリレポートを出力しないことを示し、"YES" はサマリレポートを出力することを示します。

デフォルトは "YES" です。

SUMMARY パラメータとともに CMID、FILES、USERID、または ETID のパラメータを指定すると、すべてのパラメータによって指定される基準をすべて満たす更新のみがサマリレポートに含まれます。

**TEST**

TEST パラメータを使用すると、ADAWRK ユーティリティに指定したパラメータが正しいかどうかをテストすることができます。TEST パラメータを指定した場合、レポートは出力されません。

**TIMEZONE**

TIMEZONE パラメータを使用すると、ADAWRK レポートを出力する前に時間の値を特定のタイムゾーンに変換することができます。TIMEZONE パラメータに有効な値は LOCAL、MACHINE、および +|- nn 形式 (nn は 0~23 の有効な値) による時差指定です。

パラメータ値	説明
LOCAL	時間はマシンで見つかったローカルの調整値によって調整されます。これがデフォルトです。
MACHINE	時間は見つかったとおりに出力されます。
-23~+23	時間は指定された時間数だけ戻して (マイナス記号が指定された場合) または進めて (プラス記号が指定された場合) 調整されます。

**TRANSACTIONS**

TRANSACTIONS パラメータを使用すると、トランザクションレポートを出力する必要があるかどうかを示すことができます。有効な値は "YES"、"NO"、"DETAIL"、および "FULL" です。

パラメータ値	説明
NO	トランザクションレポートを出力しません。これがデフォルトです。
YES	トランザクションレポートを出力します。
DETAIL	WORK パート 1 の自動再スタートエリアで見つかったトランザクションデータに関する追加の詳細を出力します。詳細には各トランザクションで更新されたファイルおよび ISN 番号などが含まれます。
FULL	トランザクションの各更新に関連するすべてのデータストレージおよびインバーテッドリスト (ディスクリプタバリュートーブル DVT) が出力されます。

TRANSACTIONS パラメータとともに CMID、FILES、USERID、または ETID のパラメータを指定すると、すべてのパラメータによって指定される基準をすべて満たす更新のみがトランザクションレポートに含まれます。

**USERID**

USERID パラメータを使用すると、ユーザー ID 値を 24 個まで指定できます。このユーザー ID 値を使用して、ユーティリティによって処理された WORK パート 1 の自動再スタートエリアレコードをフィルタリングできます。ユーザー ID は、レコードのコミュニケーション ID の最後の 8 バイトに格納されています。USERID パラメータに示されているユーザー ID に最後の 8 バイトが一致するコミュニケーション ID を持つレコードのみが ADAWRK ユーティリティによって処理され、レポートに出力されます。

ユーザー ID は長さ 1~8 バイトの文字形式または 16 進形式で指定する必要があります。ユーザー ID を文字形式で指定する場合は、アスタリスク (\*) をワイルドカード文字として使用できます。ユーザー ID を 16 進形式で指定する場合は、ワイルドカードは使用できません

(16進形式のユーザー ID でアスタリスクを指定すると、ADAWRK 処理で X'5C' に変換されます)。

文字形式のユーザー ID に含まれるアスタリスクのワイルドカード文字は、ユーザー ID の最後に指定する必要があります。例えば、USERID=ABC\* と指定すると、WORK パート 1 の自動再スタートエリアレコードの中で "ABC" という文字で始まるユーザー ID の値を持つすべてのレコードが ADAWRK ユーティリティによって処理されます。ただし、USERID=\*ABC と指定すると、アスタリスクのワイルドカードは無視され、WORK パート 1 の自動再スタートエリアレコードの中で "ABC" というユーザー ID の値を持つすべてのレコードが ADAWRK ユーティリティによって処理されます。



## 201 レポートの解説

---

■ レポートの1ページ目 .....	1138
■ サマリレポート .....	1139
■ ファイルレポート .....	1142
■ トランザクションレポート .....	1146
■ チェックポイントレコードのレポート .....	1184

このchapterでは、ADAWRK ユーティリティによって生成される可能性があるレポートについて説明します。

すべてのレポートは、フィルタパラメータである CMID、ETID、FILES、および USERID の設定の影響を受けます。これらのパラメータを ADAWRK の実行で指定すると、WORK パート 1 の自動再スタートエリアレコードの中でフィルタパラメータに指定された基準を満たすレコードのみが ADAWRK によって処理され、レポートが生成されます。

## レポートの 1 ページ目

---

レポートの 1 ページ目のサンプルを以下に示します。このページは、ADAWRK パラメータの指定にかかわらず、常に出力されます。この 1 ページ目では、レポートを生成するために使用された ADAWRK パラメータだけでなく、レポートに使用された WORK データセットも示されます。

```
A D A W R K   V8.1  SM8   DBID = 15002   Started           2007-05-24  18:57:14
```

```
Parameters:
```

```
-----
```

```
ADAWRK NOPPT
```

```
ADAWRK FORCE=YES
```

```
ADAWRK FILES=1-4,5,10
```

```
ADAWRK TRANSACTIONS=FULL
```

```
ADAWRK REPORTFILE=YES
```

```
Work datasets provided in JCL.
```

```
Database ID on Work is 15,002.
```

```
The following Work datasets were used:
```

```
-      ADABAS.ADAWRK.TC1.WORK          NUCID:      0
```

## サマリレポート

サマリレポートは、ADAWRKSUMMARYパラメータによってトリガされます。SUMMARY=YES（デフォルト）を指定すると、WORKパート1の自動再スタートエリア内のデータの概要を示すサマリレポートが生成されます。このレポートは、次のような多数のセクションで構成されています。

- 何が処理されたかの概要および指定された入力から収集できる環境情報
- 自動再スタートの一部として変更されるファイルの概要およびそのファイルのどの部分が影響を受けるかについての詳細
- WORKパート1の自動再スタートエリア内のシステムチェックポイントレコード（"SYN" という文字で始まる名前を持つチェックポイント）の合計
- WORKパート1の自動再スタートエリア内のユーザーチェックポイントレコードの合計
- 再実行が必要なETトランザクションの合計（バックアウトされたトランザクションの小計を含む）
- 再実行が必要なETトランザクションに対して、ファイルに実行された挿入、更新、削除オペレーションの回数
- 再実行が必要なETトランザクションに対して、ファイルへのデータ修正回数の合計（挿入、更新、および削除の合計）、データレコード更新回数の合計、関連するディスクリプタバリュートーブル（インバーテッドリスト）レコードを使用した更新コマンド実行回数の合計、およびディスクリプタバリュートーブル（インバーテッドリスト）更新回数の合計
- ファイルに対する不完全なトランザクションの合計
- 不完全なトランザクションに対して、ファイルに実行された挿入、更新、および削除オペレーションの合計
- 不完全なトランザクションに対して、ファイルへのデータ修正回数の合計（挿入、更新、および削除の合計）、データレコード更新回数の合計、関連するディスクリプタバリュートーブル（インバーテッドリスト）レコードを使用した更新コマンド実行回数の合計、およびディスクリプタバリュートーブル（インバーテッドリスト）更新回数の合計

ADAWRK によって生成されるサマリレポートのサンプルを以下に示します。

```

*****
*
*          A D A W R K  Summary Report
*
*****

Work Dataset: ADABAS.ADAWRK.TC1.WORK          NUCID:      0

I Item                I Work RABN  I      Blocks I Date/Time          I
I-----I-----I-----I-----I
I LP value            I          I      1,000 I
I Last written block  I          6 I          I 2007-05-08 12:30:54 I
I Autobackout endpoint I          4 I          2 I 2007-05-08 12:30:53 I
I Backward repair endpoint I          5 I          1 I
I-----I-----I-----I-----I

Summary of potential inconsistencies:

-----

"Index Structure"....: Index structures may be inconsistent.

"Index Values".....: Index values may be incorrect.

"Address Converter"..: Address converter may be incorrect.

"Data Contents".....: Data contents may be incorrect according to application.

I File I Index I Index I Address I Data I
I Number I Structure I Values I Converter I Contents I
I-----I-----I-----I-----I
    
```

```

I      1 I          I          I          I          I
I      2 I          I          I          I          I
I      3 I          I          I          I          I
I      4 I          I          I          I          I
I      5 I          I          I          I          I
I-----I-----I-----I-----I-----I
I     10 I          I          I          I          I
I-----I-----I-----I-----I-----I

```

The following checkpoints were encountered in the area of interest:

Total number of system checkpoints: 0

Completed transactions that will be re-done during autorestart processing:

```

Total transactions.....: 0
    Backed out.....: 0
Total modification commands.....: 0
    Inserts (N1).....: 0
    Updates (A1).....: 0
    Deletes (E1).....: 0
Total data records modified.....: 0
Total modifications with descriptor updates..: 0
Total descriptor updates.....: 0

```

Incomplete transactions that must be backed out during autorestart processing:

Total transactions.....:	0
Total modification commands.....:	0
Inserts (N1).....:	0
Updates (A1).....:	0
Deletes (E1).....:	0
Total data records modified.....:	0
Total modifications with descriptor updates..:	0
Total descriptor updates.....:	0

## ファイルレポート

ファイルレポートは、ADAWRK REPORTFILE パラメータによってトリガされます。REPORTFILE=YESを指定すると、WORKパート1の自動再スタートエリア内のデータの内訳をファイルごとに示したファイルレポートが生成されます。各ファイルについて、ファイルレポートに次の情報を記載することができます。

- サマリレポートで示されるものと同様の情報
- 再実行が必要なETトランザクションの合計（バックアウトされたトランザクションの小計を含む）
- 再実行が必要なETトランザクションに対して、ファイルに実行された挿入、更新、削除オペレーションの回数
- 再実行が必要なETトランザクションに対して、ファイルへのデータ修正回数の合計（挿入、更新、および削除の合計）、データレコード更新回数の合計、関連するディスクリプタバリュートーブル（インバーテッドリスト）レコードを使用した更新コマンド実行回数の合計、およびディスクリプタバリュートーブル（インバーテッドリスト）更新回数の合計
- ファイルに対する不完全なトランザクションの合計
- 不完全なトランザクションに対して、ファイルに実行された挿入、更新、および削除オペレーションの合計
- 不完全なトランザクションに対して、ファイルへのデータ修正回数の合計（挿入、更新、および削除の合計）、データレコード更新回数の合計、関連するディスクリプタバリュートーブル（インバーテッドリスト）レコードを使用した更新コマンド実行回数の合計、およびディスクリプタバリュートーブル（インバーテッドリスト）更新回数の合計

ファイルレポートのサンプルを以下に示します。

```

*****
*
*                      A D A W R K  File Report
*
*****

I-----I
I                      File      1                      I
I-----I

Oldest activity in Work RABN          54 at 2007-06-28 00:54:38 by NUCID 0

Latest activity in Work RABN          54 at 2007-06-28 00:54:38 by NUCID 0

I File I  Index  I Index  I Address I  Data  I
I Number I Structure I Values  I Converter I Contents I
I-----I-----I-----I-----I-----I
I      1 I          I          I      *  I      *  I
I-----I-----I-----I-----I-----I

Completed transactions that will be re-done during autorestart processing:

    Total transactions.....:          1

        Backed out.....:          0
    
```

```

Total modification commands.....:      1
    Inserts (N1).....:      1
    Updates (A1).....:      0
    Deletes (E1).....:      0
Total data records modified.....:      1
Total modifications with descriptor updates..:      0
Total descriptor updates.....:      0
    
```

Incomplete transactions that must be backed out during autorestart processing:

```

Total transactions.....:      0
Total modification commands.....:      0
    Inserts (N1).....:      0
    Updates (A1).....:      0
    Deletes (E1).....:      0
Total data records modified.....:      0
Total modifications with descriptor updates..:      0
Total descriptor updates.....:      0
    
```

```

I-----I
I           File      2           I
I-----I
    
```

```

Latest activity in Work RABN      0 at           by NUCID 0
Oldest activity in Work RABN      0 at           by NUCID 0
    
```

```

I File I Index I Index I Address I Data I
I Number I Structure I Values I Converter I Contents I
I-----I-----I-----I-----I-----I
I 2 I I I I I
I-----I-----I-----I-----I-----I

```

Completed transactions that will be re-done during autorestart processing:

```

Total transactions.....: 0
    Backed out.....: 0
Total modification commands.....: 0
    Inserts (N1).....: 0
    Updates (A1).....: 0
    Deletes (E1).....: 0
Total data records modified.....: 1
Total modifications with descriptor updates..: 0
Total descriptor updates.....: 0

```

Incomplete transactions that must be backed out during autorestart processing:

```

Total transactions.....: 0
Total modification commands.....: 0
    Inserts (N1).....: 0
    Updates (A1).....: 0
    Deletes (E1).....: 0
Total data records modified.....: 0

```

```
Total modifications with descriptor updates...:      0
Total descriptor updates.....:                      0
```

## トランザクションレポート

トランザクションレポートは、ADAWRK TRANSACTIONS パラメータによってトリガされます。TRANSACTIONS=YES、TRANSACTIONS=DETAIL、または TRANSACTIONS=FULL を指定すると、WORK パート 1 の自動再スタートエリア内のデータの内訳をトランザクションごとに示したトランザクションレポートが生成されます。各トランザクションについて、トランザクションレポートに次の情報を記載することができます。

- トランザクションのコミュニケーション ID
- トランザクションが現在もオープンであるか、コミットされているか、またはバックアウトされているか
- トランザクションによって実行された挿入、更新、削除の数
- トランザクションによって実行された修正の合計（挿入、更新、削除の合計）
- 関連するディスクリプタバリューテーブル（インバーテッドリスト）レコードを伴う更新コマンドの合計
- ET データがトランザクションに関連付けられているかどうか、およびトランザクションに関連付けられている ETID があるかどうか
- TRANSACTIONS=DETAIL を指定した場合は、トランザクションによって更新された各ファイル、およびトランザクションによって更新された各ファイルの ISN
- TRANSACTIONS=FULL を指定した場合は、トランザクションの各更新に関連付けられたデータストレージイメージ、およびトランザクションの各更新に関連付けられたディスクリプタバリューテーブル（インバーテッドリスト）データ

トランザクションレポートの一部を以下に示します。

```
*****
*
*                               A D A W R K   Transaction Report                               *
*
*****
```

```

I-----I
I          New Communication ID          I
I-----I

```

Communication ID: ' .~..o 8 U001U001'

x'0004A10E209600004040404040404000F8EE80E4F0F0F1E4F0F0F1'

This transaction was committed (ET).

ETID: None

ET data was provided.

First protection data in Work RABN 4 written at 2007-05-08 12:30:53.

Last protection data in Work RABN 0 written at .

```

Total modification commands.....:      1
      Inserts (N1).....:      1
      Updates (A1).....:      0
      Deletes (E1).....:      0
Total data records modified.....:      1
Total modifications with descriptor updates..:      0
Total descriptor updates.....:      0

```

File 1 ISN 11 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 000B09C1 F8C1F800 00000103 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

```
0020 0A                                '.'

Related descriptor value updates in Work RABN          4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:
0000 C1F8C1F8 00000001                    'A8A8  .'

Field name A9 image: AI data:
0000 C1F9C1F9 C1000000 0A                  'A9A9A  .'

File      1  ISN                12 (internal) Inserted

After Image Data Storage in Work RABN          4 written at 12:30:53
0000 00210000 000C09C1 F8C1F800 00000203  ' .  ..A8A8  ..'
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000  'A2.A4A4.A9A9A  '
0020 09                                '.'

Related descriptor value updates in Work RABN          4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:
0000 C1F8C1F8 00000002                    'A8A8  .'

```

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 09 'A9A9A .'

File 1 ISN 13 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 000D09C1 F8C1F800 00000303 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 08 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 00000003 'A8A8 .'

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 08 'A9A9A .'

File 1 ISN 14 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 000E09C1 F8C1F800 00000403 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 07 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 00000004 'A8A8 .'

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 07 'A9A9A .'

File 1 ISN 15 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 000F09C1 F8C1F800 00000503 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 06 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 00000005 'A8A8 .'

Field name A9 image: AI data:

```
0000 C1F9C1F9 C1000000 06          'A9A9A  .'
```

```
File      1  ISN                16 (internal) Inserted
```

```
After Image Data Storage in Work RABN          4 written at 12:30:53
```

```
0000 00210000 001009C1 F8C1F800 00000603 ' .  ..A8A8  ..'
```

```
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A  '
```

```
0020 05                                '.'
```

```
Related descriptor value updates in Work RABN          4 written at 12:30:53
```

This record contains 2 descriptor updates.

Field name A8 image: AI data:

```
0000 C1F8C1F8 00000006          'A8A8  .'
```

Field name A9 image: AI data:

```
0000 C1F9C1F9 C1000000 05          'A9A9A  .'
```

```
File      1  ISN                17 (internal) Inserted
```

```
After Image Data Storage in Work RABN          4 written at 12:30:53
```

```
0000 00210000 001109C1 F8C1F800 00000703 ' .  ..A8A8  ..'
```

```
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A  '
```

```
0020 04                                '.'
```

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 00000007 'A8A8 .'

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 04 'A9A9A .'

File 1 ISN 18 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 001209C1 F8C1F800 00000803 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 03 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 00000008 'A8A8 .'

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 03 'A9A9A .'

File 1 ISN 19 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 001309C1 F8C1F800 00000903 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 02 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 00000009 'A8A8 .'

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 02 'A9A9A .'

File 1 ISN 20 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 001409C1 F8C1F800 00000A03 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 01 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

0000 C1F8C1F8 0000000A 'A8A8 .'

Field name A9 image: AI data:

0000 C1F9C1F9 C1000000 01 'A9A9A .'

File 1 ISN 10 (internal) Updated

Related descriptor value updates in Work RABN 4 written at 12:30:53

0000 00210000 000A09C1 F8C1F800 00000A03 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 01 '.'

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 000A09E4 F8E4F800 00000103 ' . ..U8U8 ..'

0010 E4F205E4 F4E4F40A E4F9E4F9 C1000000 'U2.U4U4.U9U9A '

0020 06 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: BI data:

```
0000 C1F8C1F8 0000000A          'A8A8  .'
```

Field name A9 image: BI data:

```
0000 C1F9C1F9 C1000000 01          'A9A9A  .'
```

```
File      1  ISN                11 (internal) Updated
```

```
Related descriptor value updates in Work RABN          4 written at 12:30:53
```

```
0000 00210000 000B09C1 F8C1F800 00000103  ' .  ..A8A8  ..'
```

```
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000  'A2.A4A4.A9A9A  '
```

```
0020 0A          ' . '
```

```
After Image Data Storage in Work RABN          4 written at 12:30:53
```

```
0000 00210000 000B09E4 F8E4F800 00000203  ' .  ..U8U8  ..'
```

```
0010 E4F205E4 F4E4F40A E4F9E4F9 C1000000  'U2.U4U4.U9U9A  '
```

```
0020 05          ' . '
```

```
Related descriptor value updates in Work RABN          4 written at 12:30:53
```

This record contains 2 descriptor updates.

Field name A8 image: BI data:

```
0000 C1F8C1F8 00000001          'A8A8  .'
```

Field name A9 image: BI data:

```
0000 C1F9C1F9 C1000000 0A          'A9A9A  .'
```

File 1 ISN 12 (internal) Updated

Related descriptor value updates in Work RABN 4 written at 12:30:53

0000 00210000 000C09C1 F8C1F800 00000203 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 09 '.'

After Image Data Storage in Work RABN 4 written at 12:30:53

0000 00210000 000C09E4 F8E4F800 00000303 ' . ..U8U8 ..'

0010 E4F205E4 F4E4F40A E4F9E4F9 C1000000 'U2.U4U4.U9U9A '

0020 04 '.'

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: BI data:

0000 C1F8C1F8 00000002 'A8A8 .'

Field name A9 image: BI data:

0000 C1F9C1F9 C1000000 09 'A9A9A .'

File 1 ISN 13 (internal) Updated

Related descriptor value updates in Work RABN 4 written at 12:30:53

```
0000 00210000 000D09C1 F8C1F800 00000303 ' . ..A8A8 ..'
```

```
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A  '
```

```
0020 08 '.'
```

```
After Image Data Storage in Work RABN 4 written at 12:30:53
```

```
0000 00210000 000D09E4 F8E4F800 00000403 ' . ..U8U8 ..'
```

```
0010 E4F205E4 F4E4F40A E4F9E4F9 C1000000 'U2.U4U4.U9U9A  '
```

```
0020 03 '.'
```

```
Related descriptor value updates in Work RABN 4 written at 12:30:53
```

This record contains 2 descriptor updates.

Field name A8 image: BI data:

```
0000 C1F8C1F8 00000003 'A8A8  .'
```

Field name A9 image: BI data:

```
0000 C1F9C1F9 C1000000 08 'A9A9A  .'
```

```
File 1 ISN 14 (internal) Updated
```

```
Related descriptor value updates in Work RABN 4 written at 12:30:53
```

```
0000 00210000 000E09C1 F8C1F800 00000403 ' . ..A8A8 ..'
```

```
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A  '
```

```
0020 07 '.'
```

After Image Data Storage in Work RABN 4 written at 12:30:53

```
0000 00210000 000E09E4 F8E4F800 00000503 ' . ..U8U8 ..'
0010 E4F205E4 F4E4F40A E4F9E4F9 C1000000 'U2.U4U4.U9U9A '
0020 02 '.'
```

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: BI data:

```
0000 C1F8C1F8 00000004 'A8A8 .'
```

Field name A9 image: BI data:

```
0000 C1F9C1F9 C1000000 07 'A9A9A .'
```

File 1 ISN 15 (internal) Updated

Related descriptor value updates in Work RABN 5 written at 12:30:54

```
0000 00210000 000F09C1 F8C1F800 00000503 ' . ..A8A8 ..'
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '
0020 06 '.'
```

After Image Data Storage in Work RABN 5 written at 12:30:54

```
0000 00210000 000F09E4 F8E4F800 00000603 ' . ..U8U8 ..'
0010 E4F205E4 F4E4F40A E4F9E4F9 C1000000 'U2.U4U4.U9U9A '
0020 01 '.'
```

Related descriptor value updates in Work RABN 5 written at 12:30:54

This record contains 2 descriptor updates.

Field name A8 image: BI data:

0000 C1F8C1F8 00000005 'A8A8 .'

Field name A9 image: BI data:

0000 C1F9C1F9 C1000000 06 'A9A9A .'

File 1 ISN 16 (internal) Deleted

Related descriptor value updates in Work RABN 5 written at 12:30:54

0000 00210000 001009C1 F8C1F800 00000603 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 05 '.'

Related descriptor value updates in Work RABN 5 written at 12:30:54

This record contains 2 descriptor updates.

Field name A8 image: BI data:

0000 C1F8C1F8 00000006 'A8A8 .'

Field name A9 image: BI data:

0000 C1F9C1F9 C1000000 05 'A9A9A .'

File 1 ISN 17 (internal) Deleted

Related descriptor value updates in Work RABN 5 written at 12:30:54

0000 00210000 001109C1 F8C1F800 00000703 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 04 '.'

Related descriptor value updates in Work RABN 5 written at 12:30:54

This record contains 2 descriptor updates.

Field name A8 image: BI data:

0000 C1F8C1F8 00000007 'A8A8 .'

Field name A9 image: BI data:

0000 C1F9C1F9 C1000000 04 'A9A9A .'

File 1 ISN 18 (internal) Deleted

Related descriptor value updates in Work RABN 5 written at 12:30:54

0000 00210000 001209C1 F8C1F800 00000803 ' . ..A8A8 ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '

0020 03 '.'

Related descriptor value updates in Work RABN 5 written at 12:30:54

This record contains 2 descriptor updates.

Field name A8 image: BI data:

0000 C1F8C1F8 00000008 'A8A8 .'

Field name A9 image: BI data:

0000 C1F9C1F9 C1000000 03 'A9A9A .'

Transaction was committed (ET)

I-----I

I New Communication ID I

I-----I

Communication ID: ' .~..o 8 U001U001'

x'0004A10E209600004040404040404000F8EE80E4F0F0F1E4F0F0F1'

This transaction was committed (ET).

ETID: None

ET data was provided.

First protection data in Work RABN 4 written at 2007-05-08 12:30:53.

Last protection data in Work RABN 0 written at .

```

Total modification commands.....:      0
    Inserts (N1).....:      0
    Updates (A1).....:      0
    Deletes (E1).....:      0

Total data records modified.....:      0
Total modifications with descriptor updates..:      0
Total descriptor updates.....:      0
    
```

File 1 ISN 11 (internal) Inserted

After Image Data Storage in Work RABN 4 written at 12:30:53

```

0000 00210000 000B09C1 F8C1F800 00000103 ' . ..A8A8 ..'
0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000 'A2.A4A4.A9A9A '
0020 0A ' .'
    
```

Related descriptor value updates in Work RABN 4 written at 12:30:53

This record contains 2 descriptor updates.

Field name A8 image: AI data:

```

0000 C1F8C1F8 00000001 'A8A8 .'
    
```

Field name A9 image: AI data:

```

0000 C1F9C1F9 C1000000 0A 'A9A9A .'
    
```

File 1 ISN 12 (internal) Inserted

```

After Image Data Storage in Work RABN          4 written at 12:30:53

0000 00210000 000C09C1 F8C1F800 00000203  ' .  ..A8A8  ..'

0010 C1F205C1 F4C1F40A C1F9C1F9 C1000000  'A2.A4A4.A9A9A  '

.

. (etc)

.

.I-----I
I          New Communication ID          I
I-----I

Communication ID:  ' .~..o          .T.A001  '

                x'0005A10E209600004040404040404000FAE380C1F0F0F100000000'

This transaction was committed (ET).

ETID: None

ET data was not provided.

First protection data in Work RABN          2 written at 2007-07-13 19:19:23

Last  protection data in Work RABN          5 written at 2007-07-13 19:19:23

Total modification commands.....:          1
    Inserts (N1).....:          1
    Updates (A1).....:          0
    Deletes (E1).....:          0

Total data records modified.....:          4

Total modifications with descriptor updates..:          0

Total descriptor updates.....:          0

```

```

--- File      34  ISN                1 (internal) Inserted
.. This segment ISN =                1 (primary)
. Next segment ISN =                1 (secondary)

#1: After Image Data Storage in Work RABN          2 written <19:19:23
0000 13090000 00000000 00010081 00000001  '..      . a  .'
0010 00000001 6480FCD6 C3C3E4D9 C5D5C3C5  '    ....OCCURENCE'
0020 F0F0F100 00000A40 40404040 40404040  '001      .      .'
.
.
.
12F0 40404040 4040C5D5 C4D6C3C3 E4D9C5D5  '          ENDOCCUREN'
1300 C3C5F0F3 F1000000 0A                'CE031      .'

--- File      34  ISN                1 (internal) Inserted
.. This segment ISN =                1 (secondary)
. Next segment ISN =                2 (secondary)

#2: After Image Data Storage in Work RABN          3 written <19:19:23
0000 138F0000 00000000 00010041 00000002  '..      . .  .'
0010 00000001 80FC4040 40404040 40C5D5C4  '    ...          END'
0020 D6C3C3E4 D9C5D5C3 C5F0F3F2 0000000A  'OCCURENCE032  .'
.
.
.

```

```
1370 40404040 40404040 404040C5 D5C411D6 '          END.O'
1380 C3C3E4D9 C5D5C3C5 F0F6F200 00000A 'OCCURENCE062 .'

--- File      34  ISN                1 (internal) Inserted
.. This segment ISN =                2 (secondary)
. Next segment ISN =                3 (secondary)

#3: After Image Data Storage in Work RABN          4 written <19:19:23
0000 12F70000 00000000 00020041 00000003 '.7      . . .'
0010 00000001 75404040 40404040 40404040 ' ..      '
0020 40404040 40404040 40404040 40404040 '          '
.
.
.
12E0 40404040 40404040 40404040 40404040 '          '
12F0 40404040 C5D5C4                '      END'

--- File      34  ISN                1 (internal) Inserted
.. This segment ISN =                3 (secondary)
. Last segment

#4: After Image Data Storage in Work RABN          5 written <19:19:23
0000 05100000 00000000 00030041 00000000 '..      . . .'
0010 00000001 80FCD6C3 C3E4D9C5 D5C3C5F0 ' ...OCCURENCE0'
0020 F9F30000 000A4040 40404040 40404040 '93      .'
.
```

## レポートの解説

---

```
.  
.
04F0 40404040 40404040 40404040 40404040 ' '
0500 40404040 40404040 40404040 40C5D5C4 ' END'

Transaction was committed (ET).

.
.
.

(etc.)
```

次のトランザクションレポートのサンプルは、レポートが EXU ユーザーにはどのように表示されるかを示しています（レポートの U002）。

```
*****
*                               A D A W R K Transaction Report                               *
*****

I-----I
I                               New Communication ID                               I
I-----I

Communication ID: '????hhhhA.Qb2.?? U001 '

                x'7777777788888888C11FD882F238574600000000E4F0F0F100000000'

This transaction was committed (ET).

ETID: None
```



```
Communication ID: ' .~..o          9??U001      '
                x'0005A10E209600004040404040404000F9B880E4F0F0F100000000'
```

This transaction was committed (ET).

ETID: 'U001 '

ET data was not provided.

First protection data in Work RABN 2 written at 2007-08-30 12:32:13

Last protection data in Work RABN 2 written at 2007-08-30 12:32:13

```

Total modification commands.....:      10
    Inserts (N1).....:      10
    Updates (A1).....:      0
    Deletes (E1).....:      0
Total data records modified.....:      10
Total modifications with descriptor updates..:      10
Total descriptor updates.....:      10
```

--- File 4 ISN 11 (internal) Inserted

#2: After Image Data Storage in Work RABN 2 written <12:32:13

```
0000 000F0000 000B09E4 D3F0F100 000000      ' .      ..UL01      '
```

#3: Descriptor value updates in Work RABN 2 written <12:32:13

Value for descriptor A8 inserted:

```
0000 E4D3F0F1 00000000          'UL01      '
```

```
--- File      4  ISN                12 (internal) Inserted

#4: After Image Data Storage in Work RABN          2 written <12:32:13
0000 000F0000 000C09E4 D3F0F100 000001  ' .  ..UL01  .'

#5: Descriptor value updates in Work RABN          2 written <12:32:13
Value for descriptor A8 inserted:
0000 E4D3F0F1 00000001                      'UL01  .'

--- File      4  ISN                13 (internal) Inserted

#6: After Image Data Storage in Work RABN          2 written <12:32:13
0000 000F0000 000D09E4 D3F0F100 000002  ' .  ..UL01  .'

#7: Descriptor value updates in Work RABN          2 written <12:32:13
Value for descriptor A8 inserted:
0000 E4D3F0F1 00000002                      'UL01  .'

--- File      4  ISN                14 (internal) Inserted

#8: After Image Data Storage in Work RABN          2 written <12:32:13
0000 000F0000 000E09E4 D3F0F100 000003  ' .  ..UL01  .'

#9: Descriptor value updates in Work RABN          2 written <12:32:13
Value for descriptor A8 inserted:
0000 E4D3F0F1 00000003                      'UL01  .'
```

--- File 4 ISN 15 (internal) Inserted

#10: After Image Data Storage in Work RABN 2 written <12:32:13

0000 000F0000 000F09E4 D3F0F100 000004 ' . ..UL01 .'

#11: Descriptor value updates in Work RABN 2 written <12:32:13

Value for descriptor A8 inserted:

0000 E4D3F0F1 00000004 'UL01 .'

--- File 4 ISN 16 (internal) Inserted

#12: After Image Data Storage in Work RABN 2 written <12:32:13

0000 000F0000 001009E4 D3F0F100 000005 ' . ..UL01 .'

#13: Descriptor value updates in Work RABN 2 written <12:32:13

Value for descriptor A8 inserted:

0000 E4D3F0F1 00000005 'UL01 .'

--- File 4 ISN 17 (internal) Inserted

#14: After Image Data Storage in Work RABN 2 written <12:32:13

0000 000F0000 001109E4 D3F0F100 000006 ' . ..UL01 .'

#15: Descriptor value updates in Work RABN 2 written <12:32:13

Value for descriptor A8 inserted:

```
0000 E4D3F0F1 00000006          'UL01  .'

--- File      4  ISN              18 (internal) Inserted

#16: After Image Data Storage in Work RABN          2 written <12:32:13
0000 000F0000 001209E4 D3F0F100 000007  ' .  ..UL01  .'

#17: Descriptor value updates in Work RABN          2 written <12:32:13
      Value for descriptor A8 inserted:
0000 E4D3F0F1 00000007          'UL01  .'

--- File      4  ISN              19 (internal) Inserted

#18: After Image Data Storage in Work RABN          2 written <12:32:13
0000 000F0000 001309E4 D3F0F100 000008  ' .  ..UL01  .'

#19: Descriptor value updates in Work RABN          2 written <12:32:13
      Value for descriptor A8 inserted:
0000 E4D3F0F1 00000008          'UL01  .'

--- File      4  ISN              20 (internal) Inserted

#20: After Image Data Storage in Work RABN          2 written <12:32:13
0000 000F0000 001409E4 D3F0F100 000009  ' .  ..UL01  .'

#21: Descriptor value updates in Work RABN          2 written <12:32:13
```

Value for descriptor A8 inserted:

0000 E4D3F0F1 00000009 'UL01 .'

Transaction was committed (ET).

```

I-----I
I          New Communication ID          I
I-----I
    
```

Communication ID: ' .~..o 9??U002 '

x'0005A10E209600004040404040404000F9B880E4F0F0F200000000'

These updates were performed by an EXU user.

First protection data in Work RABN 1 written at 2007-08-30 12:32:08

Last protection data in Work RABN 1 written at 2007-08-30 12:32:08

```

Total modification commands.....:      10
    Inserts (N1).....:      10
    Updates (A1).....:      0
    Deletes (E1).....:      0

Total data records modified.....:      10
Total modifications with descriptor updates..:      10
Total descriptor updates.....:      10
    
```

--- File 5 ISN 1 (internal) Inserted

```
#22: After Image Data Storage in Work RABN          1 written <12:32:08
0000 000F0000 000109E4 F0F0F200 000000          ' . ..U002  '
```

```
#23: Descriptor value updates in Work RABN          1 written <12:32:08
Value for descriptor A8 inserted:
0000 E4F0F0F2 00000000                          'U002  '
```

```
--- File      5  ISN                               2 (internal) Inserted
```

```
#24: After Image Data Storage in Work RABN          1 written <12:32:08
0000 000F0000 000209E4 F0F0F200 000001          ' . ..U002  .'
```

```
#25: Descriptor value updates in Work RABN          1 written <12:32:08
Value for descriptor A8 inserted:
0000 E4F0F0F2 00000001                          'U002  .'
```

```
--- File      5  ISN                               3 (internal) Inserted
```

```
#26: After Image Data Storage in Work RABN          1 written <12:32:08
0000 000F0000 000309E4 F0F0F200 000002          ' . ..U002  .'
```

```
#27: Descriptor value updates in Work RABN          1 written <12:32:08
Value for descriptor A8 inserted:
0000 E4F0F0F2 00000002                          'U002  .'
```

--- File 5 ISN 4 (internal) Inserted

#28: After Image Data Storage in Work RABN 1 written <12:32:08

0000 000F0000 000409E4 F0F0F200 000003 ' . ..U002 .'

#29: Descriptor value updates in Work RABN 1 written <12:32:08

Value for descriptor A8 inserted:

0000 E4F0F0F2 00000003 'U002 .'

--- File 5 ISN 5 (internal) Inserted

#30: After Image Data Storage in Work RABN 1 written <12:32:08

0000 000F0000 000509E4 F0F0F200 000004 ' . ..U002 .'

#31: Descriptor value updates in Work RABN 1 written <12:32:08

Value for descriptor A8 inserted:

0000 E4F0F0F2 00000004 'U002 .'

--- File 5 ISN 6 (internal) Inserted

#32: After Image Data Storage in Work RABN 1 written <12:32:08

0000 000F0000 000609E4 F0F0F200 000005 ' . ..U002 .'

#33: Descriptor value updates in Work RABN 1 written <12:32:08

Value for descriptor A8 inserted:

0000 E4F0F0F2 00000005 'U002 .'

```
--- File      5  ISN                7 (internal) Inserted

#34: After Image Data Storage in Work RABN          1 written <12:32:08
0000 000F0000 000709E4 F0F0F200 000006      ' .  ..U002  .'
```

```
#35: Descriptor value updates in Work RABN          1 written <12:32:08
      Value for descriptor A8 inserted:
0000 E4F0F0F2 00000006                          'U002  .'
```

```
--- File      5  ISN                8 (internal) Inserted

#36: After Image Data Storage in Work RABN          1 written <12:32:08
0000 000F0000 000809E4 F0F0F200 000007      ' .  ..U002  .'
```

```
#37: Descriptor value updates in Work RABN          1 written <12:32:08
      Value for descriptor A8 inserted:
0000 E4F0F0F2 00000007                          'U002  .'
```

```
--- File      5  ISN                9 (internal) Inserted

#38: After Image Data Storage in Work RABN          1 written <12:32:08
0000 000F0000 000909E4 F0F0F200 000008      ' .  ..U002  .'
```

```
#39: Descriptor value updates in Work RABN          1 written <12:32:08
      Value for descriptor A8 inserted:
```

0000 E4F0F0F2 00000008 'U002 .'

--- File 5 ISN 10 (internal) Inserted

#40: After Image Data Storage in Work RABN 1 written <12:32:08

0000 000F0000 000A09E4 F0F0F200 000009 ' . ..U002 .'

#41: Descriptor value updates in Work RABN 1 written <12:32:08

Value for descriptor A8 inserted:

0000 E4F0F0F2 00000009 'U002 .'

1

Summary of communication IDs with recovery data:

-----

Communication ID: '????hA.Qb2.?? U001 '

x'7777777788888888C11FD882F238574600000000E4F0F0F100000000'

Communication ID: ' .~..o 9??U002 '

x'0005A10E209600004040404040404000F9B880E4F0F0F200000000'

Communication ID: ' .~..o 9??U001 '

x'0005A10E209600004040404040404000F9B880E4F0F0F100000000'

1

\*\*\*\*\*

\* A D A W R K Summary Report \*

\*\*\*\*\*

Work Dataset: RD.DB035.WORKR1

I Item	I Work RABN	I Blocks	I Date/Time
I LP value	I	I 800	I
I Last written block	I 2	I	I 2007-08-30 12:32:13
I Autobackout endpoint	I 1	I 2	I 2007-08-30 12:32:08
I Backward repair endpoint	I 1	I 2	I 2007-08-30 12:32:08

Summary of potential inconsistencies:

-----

"Index Structure"....: Index structures may be inconsistent.

"Index Values".....: Index values may be incorrect.

"Address Converter"..: Address converter may be incorrect.

"Data Contents".....: Data contents may be incorrect according to application.

I File	I Index	I Index	I Address	I Data
I Number	I Structure	I Values	I Converter	I Contents
I 4	I	I	I	I *
I 5	I *	I *	I *	I *

```

I      19 I          I          I          I      *      I
I-----I-----I-----I-----I-----I

```

The following checkpoints were encountered in the area of interest:

Total number of system checkpoints: 5

Completed transactions that will be re-done during autorestart processing:

```

Total transactions.....: 3
    Backed out.....: 0
Total modification commands.....: 21
    Inserts (N1).....: 21
    Updates (A1).....: 0
    Deletes (E1).....: 0
Total data records modified.....: 21
Total modifications with descriptor updates..: 20
Total descriptor updates.....: 20

```

Incomplete transactions that must be backed out during autorestart processing:

```

Total transactions.....: 0
Total modification commands.....: 0
    Inserts (N1).....: 0
    Updates (A1).....: 0
    Deletes (E1).....: 0

```

```

Total data records modified.....:      0
Total modifications with descriptor updates..:  0
Total descriptor updates.....:      0

```

1

```

*****
*
*                      A D A W R K   File Report
*
*****

```

```

I-----I
I                      File      4                      I
I-----I

```

```

Oldest activity in Work RABN      2 at 2007-08-30 12:32:13
Latest activity in Work RABN      2 at 2007-08-30 12:32:13

```

```

I File I   Index I Index I Address I Data I
I Number I Structure I Values I Converter I Contents I
I-----I-----I-----I-----I-----I
I      4 I           I           I           I      * I
I-----I-----I-----I-----I-----I

```

Completed transactions that will be re-done during autorestart processing:

```

Total transactions.....:      1

```

```

        Backed out.....:          0
Total modification commands.....:    10
        Inserts (N1).....:         10
        Updates (A1).....:          0
        Deletes (E1).....:          0
Total data records modified.....:    10
Total modifications with descriptor updates..:  10
Total descriptor updates.....:      10
    
```

Incomplete transactions that must be backed out during autorestart processing:

```

Total transactions.....:          0
Total modification commands.....:    0
        Inserts (N1).....:          0
        Updates (A1).....:          0
        Deletes (E1).....:          0
Total data records modified.....:    0
Total modifications with descriptor updates..:  0
Total descriptor updates.....:      0
    
```

```

I-----I
I          File      5          I
I-----I
    
```

```

Oldest activity in Work RABN          1 at 2007-08-30 12:32:08
Latest activity in Work RABN          1 at 2007-08-30 12:32:08
    
```

```

I File I Index I Index I Address I Data I
I Number I Structure I Values I Converter I Contents I
I-----I-----I-----I-----I-----I
I 5 I * I * I * I * I
I-----I-----I-----I-----I-----I

```

Completed transactions that will be re-done during autorestart processing:

```

Total transactions.....: 1
    Backed out.....: 0
Total modification commands.....: 10
    Inserts (N1).....: 10
    Updates (A1).....: 0
    Deletes (E1).....: 0
Total data records modified.....: 10
Total modifications with descriptor updates..: 10
Total descriptor updates.....: 10

```

Incomplete transactions that must be backed out during autorestart processing:

```

Total transactions.....: 0
Total modification commands.....: 0
    Inserts (N1).....: 0
    Updates (A1).....: 0
    Deletes (E1).....: 0

```

```

Total data records modified.....:      0
Total modifications with descriptor updates...:      0
Total descriptor updates.....:      0
    
```

```

I-----I
I           File      19           I
I-----I
    
```

```

Oldest activity in Work RABN          2 at 2007-08-30 12:32:13
Latest activity in Work RABN          2 at 2007-08-30 12:32:13
    
```

```

I File I   Index   I Index   I Address I   Data   I
I Number I Structure I Values   I Converter I Contents I
I-----I-----I-----I-----I-----I
I   19 I           I           I           I   *   I
I-----I-----I-----I-----I-----I
    
```

Completed transactions that will be re-done during autorestart processing:

```

Total transactions.....:      1
    Backed out.....:      0
Total modification commands.....:      1
    Inserts (N1).....:      1
    Updates (A1).....:      0
    
```

```
Deletes (E1).....: 0
Total data records modified.....: 1
Total modifications with descriptor updates..: 0
Total descriptor updates.....: 0
```

Incomplete transactions that must be backed out during autorestart processing:

```
Total transactions.....: 0
Total modification commands.....: 0
    Inserts (N1).....: 0
    Updates (A1).....: 0
    Deletes (E1).....: 0
Total data records modified.....: 0
Total modifications with descriptor updates..: 0
Total descriptor updates.....: 0
```

A D A W R K Terminated normally

2007-08-30 12:33:39

## チェックポイントレコードのレポート

WORKパート1の自動再スタートエリアで見つかったチェックポイントレコードおよび関連データは ADAWRK ユーティリティで出力することができ、CHECKPOINT パラメータでトリガされます。CHECKPOINT=YES を指定すると、次のようなチェックポイントレコードが出力に表示されます。同時にトランザクションレポートを要求すると、これらのチェックポイントレコードはトランザクションレポートの各所に記載されます。

```
A D A W R K   V8.1  SM8   DBID = 15003  Started           2007-01-10  18:34:01
```

```
Work RABN           1 written at hh:mm:ss  System Checkpoint
```

```
written by jobname='cccccccc'
```

```
Checkpoint Name=nnnn Type=tt xxxxxxxxxxx
```

```
Work RABN           1 written at hh:mm:ss  System Checkpoint
```

```
written by jobname='cccccccc'
```

```
Checkpoint Name=nnnn Type=tt xxxxxxxxxxx
```

```
Work RABN           1 written at hh:mm:ss  System Checkpoint
```

```
written by jobname='cccccccc'
```

```
Checkpoint Name=nnnn Type=tt xxxxxxxxxxx
```

# 202

## JCL/JCS の必要項目と指定例

---

■ z/OS .....	1186
■ VSE .....	1189
■ z/VM .....	1191

このセクションでは、ADAWRK を z/OS システムで実行するために必要なジョブ制御情報について説明し、z/OS ジョブストリームの例を示します。

このchapterでは、次のトピックについて説明します。

## z/OS

データセット	DD 名	ストレージ	追加情報
アソシエータ	DDASSOR <sub>n</sub>	ディスク	NOPPT パラメータが指定されている場合は不要
ワーク	DDWORKR <sub>n</sub>	ディスク	クラスタニュークリアスに対して複数のWORK データセットの指定が必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAWRK パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAWRK メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

### ADAWRK JCL の例 (z/OS)

次の例はサマリレポートを生成します。この例については、MVSJOBS データセットのメンバ ADAWRKJ を参照してください。

```
//ADAWRK      JOB
//*
//*      ADAWRK: ALL FUNCTIONS
//*
//WRK        EXEC  PGM=ADARUN
//STEPLIB    DD   DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//*
//DDASSOR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDWORKR1   DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDDRUCK    DD   SYSOUT=X
//DDPRINT    DD   SYSOUT=X
//SYSUDUMP   DD   SYSOUT=X
```

```
//DDCARD      DD      *

ADARUN  PROG=ADAWRK,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy

/*

//DDKARTE     DD      *

ADAWRK    SUMMARY=YES

/*
```

次の例では、サマリレポートとトランザクションレポートが生成されます。トランザクションレポートでは、指定したWORK データセット上のすべてのトランザクションの完全レポートが生成されます。この例については、MVSJOBS データセットのメンバ ADAWRKT を参照してください。

```
//ADAWRK      JOB

//*

//*   ADAWRK: PRINT ALL RELEVANT TRANSACTIONS ON WORK

//*

//WRK        EXEC   PGM=ADARUN

//STEPLIB    DD     DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD

//*

//DDASSOR1   DD     DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO

//DDWORKR1   DD     DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK

//DDDRUCK    DD     SYSOUT=X

//DDPRINT    DD     SYSOUT=X

//SYSUDUMP   DD     SYSOUT=X

//DDCARD     DD     *

ADARUN  PROG=ADAWRK,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy

/*

//DDKARTE     DD      *
```

```
ADAWRK      TRANSACTIONS=FULL
```

```
/*
```

次の例では、サマリレポート、トランザクションレポート、ファイルレポートが生成されます。WORK データセット上のすべてのトランザクションの完全なレポートが含まれます。ただし、ユーティリティによって処理およびレポートされるのはファイル 1、2、3、4、5、10 の WORK データセットのみです。また、サンプル JCL にアソシエータデータセットがないことに注意してください。これは、実行に PPT を使用しないことを指定する NOPPT パラメータが指定されており、このため、アソシエータデータセットが不要であるためです。

```
//ADAWRK      JOB
//*
//*      ADAWRK: PRINT ALL RELEVANT TRANSACTIONS ON WORK
//*
//WRK        EXEC   PGM=ADARUN
//STEPLIB    DD     DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//*
//DDWORKR1   DD     DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDDRUCK    DD     SYSOUT=X
//DDPRINT    DD     SYSOUT=X
//SYSUDUMP   DD     SYSOUT=X
//DDCARD     DD     *
ADARUN      PROG=ADAWRK,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
//*
//DDKARTE    DD     *
ADAWRK      NOPPT
ADAWRK      FORCE=YES
ADAWRK      FILES=1-4,5,10
ADAWRK      TRANSACTIONS=FULL
ADAWRK      REPORTFILE=YES
```

/\*

## VSE

ファイル	記号名	ストレージ	論理ユニット	追加情報
アソシエータ	ASSOR <sub>n</sub>	ディスク	*	NOPPTパラメータが指定されている場合は不要
ワーク	WORKR <sub>n</sub>	ディスク	*	クラスタニュークリアスに対して複数のWORKデータセットの指定が必要
ADARUN パラメータ		リーダー テープ ディスク	SYSRDR SYS000 *	オペレーションマニュアル
ADAWRK パラメータ		リーダー	SYSIPT	
ADARUN メッセージ		プリンタ	SYSLST	メッセージおよびコード
ADAWRK メッセージ		プリンタ	SYS009	メッセージおよびコード

\*

## ADAWRK JCL の例 (VSE)

次の例はサマリレポートを生成します。この例については、メンバ ADAWRKJ.X を参照してください。

```

* $$ JOB JNM=ADAWRKJ,CLASS=A,DISP=D

* $$ LST CLASS=A,DISP=D

// JOB ADAWRKJ      EXECUTE THE ADABAS UTILITY ***WRK***

// OPTION LOG,PARTDUMP

*

* *****

*      SAMPLE JOB STREAM TO USE THE ADABAS UTILITY ADAWRK

*      PRINT SUMMARY REPORT

* *****

// EXEC PROC=ADAVVLIB                                     <=====

```

```
// EXEC PROC=ADAVVFIL <=====
*
* *****
*
*   DON'T FORGET TO CUSTOMIZE PARAMETERS OF ADABAS UTILITY
*
* *****
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAWRK,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy <=====
/*
ADAWRK SUMMARY=YES <=====
/*
/&
* $$ EOJ
```

次の例では、サマリレポートとトランザクションレポートが生成されます。トランザクションレポートでは、指定した WORK データセット上のすべてのトランザクションの完全レポートが生成されます。この例については、メンバ ADAWRKT.X を参照してください。

```
* $$ JOB JNM=ADAWRKT,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAWRKT EXECUTE THE ADABAS UTILITY ***WRK***
// OPTION LOG,PARTDUMP
*
* *****
*
*   SAMPLE JOB STREAM TO USE THE ADABAS UTILITY ADAWRK
*
*   PRINT ALL RELEVANT TRANSACTIONS ON WORK
*
* *****
// EXEC PROC=ADAVVLIB <=====
// EXEC PROC=ADAVVFIL <=====
```

```

*
* *****
*      DON'T FORGET TO CUSTOMIZE PARAMETERS OF ADABAS UTILITY
* *****

// EXEC ADARUN,SIZE=ADARUN

ADARUN  PROG=ADAWRK,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy  <=====

/*

ADAWRK  TRANSACTIONS=FULL                                     <=====

/*

/&

* $$ EOJ

```

## z/VM

ファイル	記号名	ストレージ	追加情報
アソシエータ	DDASSOR <sub>n</sub>	ディスク	NOPPT パラメータが指定されている場合は不要
ワーク	DDWORKR <sub>n</sub>	ディスク	
ADARUN パラメータ	DDCARD	ディスク/ 端末/ リーダー	オペレーションマニュアル
ADAWRK パラメータ	DDKARTE	ディスク/ 端末/ リーダー	
ADARUN メッセージ	DDPRINT	ディスク/ 端末/ プリンタ	メッセージおよびコード
ADAWRK メッセージ	DDDRUCK	ディスク/ 端末/ プリンタ	メッセージおよびコード

**ADAWRK JCL の例 (z/VM)**

次の例では、ADAWRK サマリレポートが生成されます。この例については、配布テープで提供されているメンバ ADAWRK を参照してください。

```
DATADEF DDASSOR1,DSN=ADABASVV.ASSO,VOL=ASSOV1
DATADEF DDWORKR1,DSN=ADABASVV.WORK,VOL=WORKV1
DATADEF DDPRINT,DSN=ADAWRK.DDPRINT,MODE=A
DATADEF DUMP,DUMMY

DATADEF DDDRUCK,DSN=ADAWRK.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNWRK.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAWRK.CONTROL,MODE=A

ADARUN
```

このサンプル JCL に基づき、サマリレポートに使用された ADARUN パラメータが RUNWRK.CONTROL データセット (DDCARD 記号名) に保存されます。例として、配布テープに提供されているメンバ RUNWRK を参照してください。

```
ADARUN PROG=ADAWRK,DEVICE=dddd,DBID=yyyyy
```

このサンプル JCL に基づき、サマリレポートに使用された ADAWRK パラメータが ADAWRK.CONTROL データセット (DDKARTE 記号名) に保存されます。例として、配布テープのメンバ WRKSUMM を参照してください。

```
ADAWRK SUMMARY=YES
```

トランザクションレポートに使用される ADAWRK パラメータの例として、配布テープのメンバ WRKTRNS を参照してください。

# 203 ADAZAP ユーティリティ：ASSO、DATA、WORK データセットの表示または修正

---

このchapterでは、次のトピックについて説明します。

- 機能概要
- ADAZAP 構文
- JCL/JCS の必要項目と指定例



# 204 機能概要

---

ADAZAP は、Adabas ニュークリアスが非アクティブであるときのみ実行できます。

ADAZAP は、ASSO データセット、DATA データセット、または WORK データセットの内容を 16 進数形式で表示し、変更（任意）するために使用します。

物理データベースブロックを修正することは大きな危険を伴うため、このユーティリティの使用は制限されています。Software AG では、書面で要求があった場合にのみ、受領が認可されている個々のお客様のサイトのユーザーに対し、ADAZAP ユーティリティを実行するために必要なマスタコードを提供します。

さらに、Software AG では、他の ZAP プログラムを保護するのと同様に、外部セキュリティシステムを使用して ADAZAP を保護することをお勧めします。

また、ADAZAP を実行する前に、現在のセーブテープを利用可能にしておくこともお勧めします。ADAZAP の実行中にエラーが発生した場合は、影響を受けたファイルまたはデータベースをリストアップすることが必要になることがあります。

データの修正が成功すると、REP データおよび VER データが含まれる SYNPF 3F チェックポイントレコードが書き込まれ、加えられた変更に対する監査証跡が提供されます。

Adabas の以前のリリースでは、異なる構文で実行される ADAZAP のバージョンが非公式に配布されていました。このような以前のバージョンについては、これまでも現在もドキュメントは提供されておらず、サポートもされていません。



# 205 ADAZAP 構文

---

■ 基本パラメータ .....	1198
■ オプションパラメータ .....	1198
■ 例 .....	1199

```
ADAZAP MCODE = master-code
      { ASSO | DATA | WORK }
      [LENGTH = length-of-data]
      [OFFSET = { offset-from-RABN-start | 0}]
      [RABN = { rabn-number | 1}]
      [REP = replace-data]
      [VER = verify-data]
```

このchapterでは、次のトピックについて説明します。

## 基本パラメータ

---

### MCODE

セキュリティの目的で、ADAZAPユーティリティを実行するにはマスタコードが必要です。Software AG では、書面で要求があった場合に、認可されたユーザーに対して8バイトのマスタコードを提供しています。

### ASSO | DATA | WORK

表示または修正する物理データセットを指定する必要があります。

## オプションパラメータ

---

### LENGTH

表示するデータの長さです。VERを指定した場合はLENGTHを指定できず、その逆も同様です。

表示される最小バイト数は16です。これよりも下位のアドレスは16バイト境界に切り下げられ、これよりも上位のアドレスは16バイト境界に切り上げられます。

### OFFSET

ブロックの先頭からのオフセットです。値はブロックの長さに等しいか、またはそれよりも小さい、つまりブロック内に収まる必要があります。デフォルト値は0です。

### RABN

表示または修正する相対 Adabas ブロック番号 (RABN) です。デフォルトは1です。

### REP

置き換えデータです。VERパラメータに指定した確認データに等しいか、またはそれよりも短くする必要があります。最大128バイトの16進数データを指定できます。

**VER**

確認データです。置き換えデータ以上の長さにする必要があります。最大 128 バイトの 16 進数データを指定できます。

**例****例 1**

```
ADAZAP MCODE=master-code  
ADAZAP ASSO OFFSET=X'10',LENGTH=16
```

RABN はデフォルト値の 1 が使用されます。ADAZAP はデータベース名称を表示します。

**例 2**

```
ADAZAP MCODE=master-code  
ADAZAP WORK OFFSET=X'10'  
ADAZAP VER=X'C1C2'  
ADAZAP REP=X'C2C1'
```

RABN はデフォルト値の 1 が使用されます。ADAZAP は WORK データセット内のデータを修正します。



# 206

## JCL/JCS の必要項目と指定例

---

▪ BS2000 .....	1202
▪ z/OS .....	1203
▪ z/VM .....	1204
▪ VSE .....	1205

ADAZAPユーティリティを使用するためのサンプルジョブを以下に示します。DATA、ASSO、またはWORKにある特定の Adabas RABN の内容を変更するために使用することができます。

- RABN、オフセット、および置き換える値は 16 進数で指定してください。
- マスタパスワードを取得するには、お近くのサポート部門に連絡してください。

 **Important:** このユーティリティは慎重に使用する必要があります。誤って使用すると深刻な問題を引き起こす可能性があります。

このchapterでは、次のトピックについて説明します。

## BS2000

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSOR <sub>n</sub>	ディスク	ASSO が ZAP されている場合に必要
データストレージ	DDDATAR <sub>n</sub>	ディスク	DATA が ZAP されている場合に必要
ワーク	DDWORKR <sub>n</sub>	ディスク	WORK が ZAP されている場合に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAZAP パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAZAP メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

### ADAZAP JCL の例 (BS2000)

SDF フォーマットの場合：

```

/BEGIN-PROC A
/REMA
/REMA SAMPLE JCL FOR ADAZAP
/REMA
/ASS-SYSOUT L.ADAZAP.OUT
/ASS-SYSLST L.ADAZAP.LST
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADABAS.Vvrs.MOD
/SET-FILE-LINK DDASSOR1,DByyyyyy.ASSOR1,SUP=DISK(SHARE-UPD=YES)
/SET-FILE-LINK DDDATAR1,DByyyyyy.DATAR1,SUP=DISK(SHARE-UPD=YES)
/SET-FILE-LINK DDWORKR1,DByyyyyy.WORKR1,SUP=DISK(SHARE-UPD=YES)
/STA-PROG *M(ADABAS.Vvrs.MOD,ADARUN),RUN-MODE=*ADV(ALT=Y)
ADARUN PROG=ADAZAP,DB=yyyyyy
ADAZAP MCODE=xxxxxxxxx <<--- MASTER PASSWORD
ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16 <=== DISPLAY ASSO RABN 1
ADAZAP ASSO
    
```

```

RABN=1,OFFSET=X'10',VER=X'C1C2',REP=X'C2C1'
ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16 <=== DISPLAY ASSO RABN 1
/SET-JOB-STEP
/ASS-SYSDTA *PRIM
/ASS-SYSLST *PRIM
/ASS-SYSOUT *PRIM
/END-PROC

```

### ISP フォーマットの場合：

```

/.ADAZAP PROC
/OPTION MSG=FH,DUMP=YES
/REMARK *
/REMARK * SAMPLE JCL FOR ADAZAP
/REMARK *
/SYSFILE SYSLST=L.ZAP
/SYSFILE SYSDTA=(SYSCMD)
/FILE ADAyyyyy.TEMP ,LINK=DDTEMPR1
/FILE ADAyyyyy.SORT ,LINK=DDSORTR1
/FILE ADAyyyyy.ASSO ,LINK=DDASSOR1
/EXEC (ADARUN,ADA.MOD)
ADAZAP MCODE=xxxxxxxxxxx << MASTER PASSWORD
ADAZAP ASSO RABN=1,OFFSET=X 10 ,LENGTH=16 <===DISPLAY ASSO RABN 1
ADAZAP ASSO RABN=1,OFFSET=X 10 ,VER=X C1C2 ,REP=X C2C1
ADAZAP ASSO RABN=1,OFFSET=X 10 ,LENGTH=16 <===DISPLAY ASSO RABN 1
/STEP
/SYSFILE SYSDTA=( )
/SYSFILE SYSLST=( )
/ENDP

```

## z/OS

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	ASSO が ZAP されている場合に必要
データストレージ	DDDATARn	ディスク	DATA が ZAP されている場合に必要
ワーク	DDWORKRn	ディスク	WORK が ZAP されている場合に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAZAP パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAZAP メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

例 (z/OS)

```
//ADAZAP      JOB
//*
//ZAP         EXEC PGM=ADARUN
//STEPLIB    DD  DISP=SHR,DSN=ADABAS.Vvrs.LOAD          <=== ADABAS LOAD
//*
//DDASSOR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1   DD  DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDDRUCK    DD  SYSOUT=X
//DDPRINT    DD  SYSOUT=X
//SYSUDUMP   DD  SYSOUT=X
//DDCARD     DD  *
ADARUN PROG=ADAZAP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
//DDKARTE    DD  *
    ADAZAP MCODE=mmmmmmmmmm                <<--- MASTER PASSWORD

    ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16          <=== DISPLAY ASSO RABN 1
    ADAZAP ASSO
RABN=1,OFFSET=X'10',VER='C1C2',REP=X'C2C1'
    ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16          <=== DISPLAY ASSO RABN 1
/*
```

この例については、MVSJOBS データセットの ADAZAP を参照してください。

z/VM

データセット	リンク名	ストレージ	追加情報
アソシエータ	DDASSORn	ディスク	ASSO が ZAP されている場合に必要
データストレージ	DDDATARn	ディスク	DATA が ZAP されている場合に必要
ワーク	DDWORKRn	ディスク	WORK が ZAP されている場合に必要
ADARUN パラメータ	DDCARD	リーダー	オペレーションマニュアル
ADAZAP パラメータ	DDKARTE	リーダー	
ADARUN メッセージ	DDPRINT	プリンタ	メッセージおよびコード
ADAZAP メッセージ	DDDRUCK	プリンタ	メッセージおよびコード

## 例 (z/VM)

```

DATADEF DDASSOR1,DSN=ADABASVv.ASSO,VOL=ASSOV1 <=== ASSO
DATADEF DDDATAR1,DSN=ADABASVv.DATA,VOL=DATAV1 <=== DATA
DATADEF DDWORKR1,DSN=ADABASVv.WORK,VOL=WORKV1 <=== WORK
DATADEF DDPRINT,DSN=ADAZAP.DDPRINT,MODE=A
DATADEF DUMP,DUMMY

DATADEF DDRUCK,DSN=ADAZAP.DDDRUCK,MODE=A
DATADEF DDCARD,DSN=RUNZAP.CONTROL,MODE=A
DATADEF DDKARTE,DSN=ADAZAP.CONTROL,MODE=A
ADARUN

```

## RUNREP CONTROL A1 の内容

```
ADARUN PROG=ADAZAP,DEVICE=dddd:
```

## ADAREP CONTROL A1 の内容

```

ADAZAP MCODE=mmmmmmmmmm <<--- MASTER PASSWORD
ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16 <===DISPLAY ASSO RABN 1
ADAZAP ASSO RABN=1,OFFSET=X'10',VER='C1C2',REP=X'C2C1'
ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16 <===DISPLAY ASSO RABN 1

```

## VSE

データセット	リンク名	ストレージ	追加情報
アソシエータ	ASSOR <sub>n</sub>	ディスク	ASSO が ZAP されている場合に必要
データストレージ	DATAR <sub>n</sub>	ディスク	DATA が ZAP されている場合に必要
ワーク	WORKR <sub>n</sub>	ディスク	WORK が ZAP されている場合に必要
ADARUN パラメータ	CARD	リーダー	オペレーションマニュアル
ADAZAP パラメータ	KARTE	リーダー	
ADARUN メッセージ	PRINT	プリンタ	メッセージおよびコード
ADAZAP メッセージ	DRUCK	プリンタ	メッセージおよびコード

## 例 (VSE)

```
* $$ JOB JNM=ADAZAP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADAZAP
*
// EXEC PROC=ADAVvLIB
// EXEC PROC=ADAVvFIL
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADAZAP,MODE=MULTI,SVC=xxx,DEVICE=dddd,DBID=yyyyy
/*
ADAZAP MCODE=MMMMMMMMM <<--- MASTER PASSWORD
ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16 <===DISPLAY ASSO RABN 1
ADAZAP ASSO
RABN=1,OFFSET=X'10',VER='C1C2',REP=X'C2C1'
ADAZAP ASSO RABN=1,OFFSET=X'10',LENGTH=16 <===DISPLAY ASSO RABN 1
/*
/&
* $$ EOJ
```

この例については、メンバ ADAZAP.X を参照してください。

VSE プロシージャ (PROC) については「[VSE 用のライブラリおよびファイルプロシージャの例](#)」を参照してください。

## A Adabas シーケンシャルファイル

---

- シーケンシャルファイル表 ..... 1208
- オペレーティングシステムに応じた考慮事項 ..... 1210

このdocumentでは、次のトピックについて説明します。

## シーケンシャルファイル表

このセクションでは、Adabas ユーティリティが使用するシーケンシャルファイルをまとめます。表の見出しと内容については、表の後の説明を参照してください。

ユーティリティ	ファイル名	VSE テープ SYS	出力	入力	デバイス別 BLKSIZE	連結
ADACDC	DD/SIIN	10		x		○
ADACMP	DD/AUSBA	12	x	x		○
	DD/EBAND	10	x			
	DD/FEHL	14				
ADACNV	DD/FILEA	10	x			
ADALOD	DD/EBAND	10	x	x	○	○
	DD/FILEA	12	x	x		○
	DD/ISN	16		x		
	DD/OLD	14				
ADAMER	DD/EBAND	10		x		
ADAORD	DD/FILEA	10	x	x	○	
ADAPLP	DD/PLOG	14		x		○
ADARAI	DD/OUT	10	x			
ADAREP	DD/SAVE	10		x		○
	DD/PLOG	11		x		○
ADARES	DD/BACK	20	x	x		○
	DD/SIAUS1	21	x	x		○
	DD/SIAUS2	22				
	DD/SIIN	20				
ADASAV	DD/DEL1	31		x		○
	DD/DEL2	32		x		○
	DD/DEL3	33		x		○
	DD/DEL4	34		x		○
	DD/DEL5	35		x		○
	DD/DEL6	36		x		○
	DD/DEL7	37		x		○
	DD/DEL8	38		x		○
	DD/DUAL1	21	x			
	DD/DUAL2	22	x			
	DD/DUAL3	23	x			
	DD/DUAL4	24	x			

ユーティリティ	ファイル名	VSE テープ SYS	出力	入力	デバイス別 BLKSIZE	連結
	DD/DUAL5	25	x			
	DD/DUAL6	26	x			
	DD/DUAL7	27	x			
	DD/DUAL8	28	x			
	DD/FULL	30		x		○
	DD/PLOG	10		x		○
	DD/REST1	11		x		○
	DD/REST2	12		x		
	DD/REST3	13		x		
	DD/REST4	14		x		
	DD/REST5	15		x		
	DD/REST6	16		x		
	DD/REST7	17		x		
	DD/REST8	18		x		
	DD/SAVE1	11	x			
	DD/SAVE2	12	x			
	DD/SAVE3	13	x			
	DD/SAVE4	14	x			
	DD/SAVE5	15	x			
	DD/SAVE6	16	x			
	DD/SAVE7	17	x			
	DD/SAVE8	18	x			
ADASEL	DD/EXPA1	11	x			
	DD/EXPA2	12	x			
	DD/EXPA3	13	x			
	DD/EXPA4	14	x			
	DD/EXPA5	15	x			
	DD/EXPA6	16	x			
	DD/EXPA7	17	x			
	DD/EXPA8	18	x			
	DD/EXPA9	19	x			
	DD/EXPA10	20	x			
	DD/EXPA11	21	x			
	DD/EXPA12	22	x			
	DD/EXPA13	23	x			
	DD/EXPA14	24	x			
	DD/EXPA15	25	x			
	DD/EXPA16	26	x			
	DD/EXPA17	27	x			
	DD/EXPA18	28	x			
	DD/EXPA19	29	x			
	DD/EXPA20	30	x			

ユーティリティ	ファイル名	VSE テープ SYS	出力	入力	デバイス別 BLKSIZE	連結
	DD/SIIN	10		x		○
ADAULD	DD/OUT1	10	x	x	○	○
	DD/OUT2	11	x	x	○	○
	DD/ISN	12	x	x	○	○
	DD/SAVE	13		x		○
	DD/PLOG	14				
	DD/FULL	30				
	DD/DEL1-8	31~38				
ADAVAL	DD/FEHL	14	x		○	

出力と入力の両方がマークされているファイルは、指定されたプログラムによって書かれた後に読まれるものです。BS2000、z/VM、z/OS、および OS 互換のファイルの名前は "DD" で始まります (DDSIIN、DDFEHL など)。VSE ファイルの名前には "DD" は付けません。

## オペレーティングシステムに応じた考慮事項

次のセクションでは、オペレーティングシステム別のファイルとデバイス定義の特性について説明します。

### BS2000 システム



**Note:** この解説では SPF フォーマットを使用します。ISP フォーマットでは次のようになります。

SPF フォーマット	ISP フォーマット
BUFF-LEN	BLKSIZE=(STD,16) で定義される BLKSIZE
REC-FORM	RECFM
REC-SIZE	RECSIZE
SET-FILE-LINK	FILE

ファイルの参照に使用される LINK 名は次のように決定されます。

- LINK 名を構成するため、ファイル名に接頭辞 DD が付けられます。
- "連結" 列に "○" と示されている複数のファイルがテープにある場合、これらのファイルを連結することができます。連結を指定した場合は、指定されたリンク名を使用して最初のファイルが読み込まれ、最初のエンドオブファイル時に LINK 名に 01 が付けられ、そのリンク名に /SET-FILE-LINK (ISP フォーマットの場合は /FILE) ステートメントがなくなるまで読み取りが継続するという処理が行われます。

- その後のエンドオブファイルごとに LINK 名に 1 が追加され、その LINK 名の /SET-FILE-LINK (ISP フォーマットでは /FILE) ステートメントが存在する限り、最大で 99 まで読み取りが継続されます。LINK 名が 6 文字より長い場合、超過した分の文字はファイルの連番で上書きされます (例えば DDEBAND は DDEBAN01 になります)。
- BS2000 では、マルチボリュームファイルの逆順読み取りはサポートされていないため、ADARES DDBACK ファイルのすべてのボリュームは、LINK 名 DDBACK、DDBACK01、DDBACK02 などを使用して /SET-FILE-LINK (ISP フォーマットでは /FILE) ステートメントに書かれた順序と逆順に指定する必要があります。

シーケンシャルファイルの BUFF-LEN は次のように決定されます。

1. BUFF-LEN は /SET-FILE-LINK ステートメントまたはデータセットのカタログエントリから取得されます (存在する場合)。
2. BUFF-LEN を /SET-FILE-LINK ステートメントまたはカタログから取得できなかった場合、ADARUN QBLKSIZE パラメータの値が使用されます (指定されている場合)。
3. この方法でも取得できなかった場合、BUFF-LEN はデバイスタイプに依存し、次のようになります。

テープ :	32760
ディスク :	32768 (BUFF-LEN=(STD,16))

REC-SIZE と REC-FORM は次のようになります。

テープ :	REC-SIZE = BUFF-LEN - 4; REC-FORM = V;
ディスク :	REC-SIZE = BUFF-LEN - 20; RECFORM = V;
入力 :	/SET-FILE-LINK ステートメントまたはデータセットのカタログエントリから取得されます。

 **Note:** TAPE データセットの HDR2 に REC-FORM、REC-SIZE、または BUFF-LEN が一切含まれていない場合を除いて REC-FORM、REC-SIZE、BUFF-LEN を入力データセットに指定しないでください。

プライマリおよびセカンダリ割り当ての SPACE パラメータには、BUFF-LEN パラメータに指定した PAM ブロック数の 3 倍の数を指定します。それ以外の値では I/O エラーが発生します。デフォルトでは /CREATE-FILE ...,PUB(SPACE(48,48)) および /SET-FILE-LINK ...,BUFF-LEN=STD(16) (ISP フォーマットの場合は BLKSIZE=(STD,16)、SPACE=(48,48)) が有効な最小値です。

DDDRUCK および DDPRINT データセットのうちディスクに書き込み済みの部分は、通常のニュークリアスまたはユーティリティの読み取りセッションでアクセス可能です。これには次の BS2000 読み取りアクセスが含まれます。

■ SHOW-FILE

- @READ データセット
- /COPY-FILE (ISP フォーマットの場合は /COPY)

### BS2000 のシーケンシャル入力ファイルの連結

複数のデータセットを ADABAS ユーティリティへの入力媒体として使用するとき、一部のオペレーティングシステム (z/OS など) では連結機能を使用できます。

BS2000 では、次のように、元の LINK 名と 2 桁の連番 (01~99) から作成した LINK 名を指定した /SET-FILE-LINK (ISP フォーマットの場合は /FILE) ステートメントを追加することで、同様の機能を実行できます。

```
/SET-FILE-LINK DDTEST,firstfile
/SET-FILE-LINK DDTEST01,secondfile
/SET-FILE-LINK DDTEST02,thirdfile
...
/SET-FILE-LINK DDTEST99,lastfile
```

ISP フォーマットでは次のようになります。

```
/FILE firstfile ,LINK=DDTEST
/FILE secondfile,LINK=DDTEST01
/FILE thirdfile ,LINK=DDTEST02
...
/FILE lastfile ,LINK=DDTEST99
```

元の LINK 名の長さが 7 文字または 8 文字の場合、7~8 番目の位置は連番で上書きされます。例としては、次のようなものがあります。

```
/SET-FILE-LINK DDEBAND,firstfile
/SET-FILE-LINK DDEBAND01,lastfile
```

ISP フォーマットでは次のようになります。

```
/FILE firstfile ,LINK=DDEBAND
/FILE secondfile,LINK=DDEBAN01
```

連結オプションのある入力ファイル进行处理する場合、1つの入力ファイルのエンドオブファイルで、次のデータセットに対する /SET-FILE-LINK (ISP フォーマットの場合は /FILE) ステートメントが存在するかどうかのチェックが行われます。存在しない場合、シーケンシャル GET コールは EOF を返します。存在すれば、現在オープンしているデータセットがクローズされて次のファイルに対してオープンが試みられます。

このようにして連結するファイルは、ファイル特性（ブロックサイズ、レコードフォーマット、およびレコードサイズ）が同じである必要があります。

この連結の仕様は、逆順処理されるファイルに対しても適用されます。LINK 名の順序は作成順序の逆になります。例えば、ADARES で DDBACK を行う場合、次のようになります。

```
/SET-FILE-LINK DDBACK,lastfile
/SET-FILE-LINK DDBACK01,filebeforelast
/...
/SET-FILE-LINK DDBACKnn,firstfile
```

ISP フォーマットでは次のようになります。

```
/FILE lastfile ,LINK=DDBACK
/FILE filebeforelast,LINK=DDBACK01
/....
/FILE firstfile ,LINK=DDBACKnn
```

各ボリュームを別々の /SET-FILE-LINK（ISP フォーマットの場合は /FILE）ステートメントで指定すれば、この機能を用いてマルチボリュームファイルを逆順に処理できます。

次のリストは、連結オプションを使用する場合の LINK 名とユーティリティの一覧です。

DDDELn (n = 1~8)	ADASAV
DDEBAND	ADACMP ADALOD ADAMER
DDFULL	ADASAV
DDISN	ADALOD
DDPLOG	ADAPLP ADASAV
DDBACK	ADARES
DDSIIN	ADARES ADASEL
DDREST1	ADASAV (使用する LINK 名は DDREST1、DDREST01、DDREST02 など)

## ADARES 付きで連結機能を使用する例

ニュークリアスセッションにおいて、F1、F2、F3 という 3つのプロテクションログファイルが ADARES PLCOPY で作成されています。

セッションを特定のポイントにバックアウトする際は、ADARES BACKOUT 機能に次の /SET-FILE-LINK (ISP フォーマットの場合は /FILE) ステートメントを使用します。

```
/SET-FILE-LINK DDBACK,F3  
/SET-FILE-LINK DDBACK01,F2  
/SET-FILE-LINK DDBACK02,F1
```

ISP フォーマットでは次のようになります。

```
/FILE F3,LINK=DDBACK  
/FILE F2,LINK=DDBACK01  
/FILE F1,LINK=DDBACK02
```

セッション中に作成されたプロテクションログからデータベースを再生成するには、ADARES REGENERATE 機能に次の /SET-FILE-LINK (ISP フォーマットの場合は /FILE) ステートメントを使用します。

```
/SET-FILE-LINK DDSIIN,F1  
/SET-FILE-LINK DDSIIN01,F2  
/SET-FILE-LINK DDSIIN02,F3
```

ISP フォーマットでは次のようになります。

```
/FILE F1,LINK=DDSIIN  
/FILE F2,LINK=DDSIIN01  
/FILE F3,LINK=DDSIIN02
```

## バージョン 11.2 (OSD 2.0) におけるコントロールステートメント読み込み手順

BS2000バージョン11.2 (OSD 2.0) では、SYSIPTシステムファイルが利用できなくなりました。バージョン 5.3.3 以降、Adabas ではすべてのコントロールステートメントを SYSDTA システムファイルから読み込むことができます。

BS2000バージョン10.0または11.0で実行している場合、SYSIPT割り当ては従来どおり使用できますが、Software AGでは、BS2000バージョン11.2 (OSD 2.0) に移行する前に、すべてのAdabasユーティリティおよびEntire Net-WorkジョブコントロールでSYSDTAシステムファイルを指定するよう変更することをお勧めします。

## ADARUN TAPEREL：テープ解放オプション

ADARUN パラメータ TAPEREL は、テープ上のファイルにアクセスするテープ処理制御を実行するのに必要です。詳細は『ADABAS オペレーションマニュアル』を参照してください。

## z/OS システム

DDNAME はファイル名の先頭に文字 DD を付けて指定します。

ユーティリティがクローズ後にデータセット情報にアクセスできるようにするには、ユーティリティで使用するシーケンシャルデータセットの DD ステートメントに FREE=CLOSE が含まれないようにする必要があります。

シーケンシャルファイルの BLKSIZE は次のように決定します。

- ファイルの "デバイス別 BLKSIZE" 列に "○" と示されている場合、デフォルトの BLKSIZE はデバイスタイプに応じて次のようになります。

テープ：	32760
3330 ディスク：	13030
3340 ディスク：	8368
3350 ディスク：	19069
3375 ディスク：	17600
3380 ディスク：	23476
3390 ディスク：	27998

- ファイルの "デバイス別 BLKSIZE" 列で "○" が指定されていない場合、そのファイルの BLKSIZE は DD ステートメントまたはデータセットラベルから取得されます（存在する場合）。これはすべての入力ファイルで存在するはずです。
- ファイルの "デバイス別 BLKSIZE" 列で "○" が指定されておらず、かつ DD ステートメントまたはデータセットラベルから BLKSIZE を取得できない場合、ADARUN QBLKSIZE パラメータの値が使用されます（指定されている場合）。

ADACMP EBAND 以外のすべてのシーケンシャルファイルの RECFM および LRECL は、それぞれ VB および BLKSIZE-4 です。ADACMP EBAND については、RECFM および LRECL が DD ステートメントとデータセットラベルのいずれか、または両方から得られるようにしなければなりません。

DD ステートメントで DCB BUFNO パラメータの指定がない場合、オペレーティングシステムのデフォルトが使用されます。

## z/VM システム

DATADEF 名はファイル名の先頭に文字 DD を付けて指定します。

シーケンシャルファイルの BLKSIZE は次のように決定します。

- ファイルの "デバイス別 BLKSIZE" 列に "○" と示されている場合、BLKSIZE はデバイスタイプに応じて次のようになります。

テープ：	32760
FBA ディスク：	32760
3330 ディスク：	13030
3340 ディスク：	8368
3350 ディスク：	19069
3375 ディスク：	17600
3380 ディスク：	23476
3390 ディスク：	27998

- ファイルの "デバイス別 BLKSIZE" 列で "○" が指定されていない場合、そのファイルの BLKSIZE は DD ステートメントまたはデータセットラベルから取得されます（存在する場合）。これはすべての入力ファイルで存在するはずです。
- ファイルの "デバイス別 BLKSIZE" 列で "○" が指定されておらず、かつ DD ステートメントまたはデータセットラベルから BLKSIZE を取得できない場合、ADARUN QBLKSIZE パラメータの値が使用されます（指定されている場合）。

ADACMP EBAND を除くすべてのシーケンシャルファイルで、RECFM は VB、LRECL は BLKSIZE - 4 になります。ADACMP EBAND については、RECFM および LRECL が DATADEF ステートメントとデータセットラベルのいずれか、または両方から得られるようにしなければなりません。

## VSE システム

次の項目は、VSE 配下で実行されているユーティリティによって、ファイルがどのように参照されるかを決定します。

- ファイル名は、DLBL または TLBL ステートメントのファイル名として使用されます。
- "連結" 列に "○" と示されているファイルがテープに複数ある場合、これらのファイルを次のように連結することができます。
  - ファイルは指示されたファイル名で最初に読み込まれます。
  - 最初のエンドオブファイル時にファイル名に "01" が付けられ、またそのファイル名に対して TLBL ステートメントが存在する場合、引き続き読み込みが行われます。

- 以降の各エンドオブファイル時にファイル名に1が加算されていき、そのファイル名に対する TLBL ステートメントがある限り最大 99 になるまで読み込みは継続されます。
- VSE では複数ボリュームのテープファイルの逆順読み込みをサポートしないので、ADARES BACK ファイルの各ボリュームは、ファイル名 BACK、BACK01、BACK02 のように TLBL ステートメントに書かれた順序とは逆の順序に指定しなければなりません。

ディスク上のシーケンシャルファイルには、すべてのプログラマ論理ユニットを使用できます。テープ上のシーケンシャルファイルには、VSE テープの SYS 番号を用いなければなりません。またこれらの番号はすべて『Adabas インストールマニュアル』に定義された手順を用いて変更できます。

シーケンシャルファイルの BLKSIZE は次のように決定します。

- ファイルの "デバイス別 BLKSIZE" 列に "○" と示されている場合、BLKSIZE はデバイスタイプに応じて次のようになります。

テープ：	32760
FBA ディスク：	32760
3330 ディスク：	13030
3340 ディスク：	8368
3350 ディスク：	19069
3375 ディスク：	17600
3380 ディスク：	23476
3390 ディスク：	27998

- ファイルの "デバイス別 BLKSIZE" 列に "○" と示されていない場合は、ADARUN QBLKSIZE パラメータの値が指定されていればその値が使用されます。

ADACMP EBAND については、BLKSIZE は下記のように ADACMP コントロールカードに指定された RECFM および LRECL パラメータに応じて、チェックされた後に実際の BLKSIZE に変更されることがあります。

RECFM の値	実際の BLKSIZE
F	LRECL。
FB	BLKSIZE / LRECL * LRECL で、除算の余りは乗算の前に切り捨てられます。
U	LRECL。この値は BLKSIZE より大きくてはなりません。
V	LRECL+4。この値は、BLKSIZE より大きくてはなりません。
VB	BLKSIZE。この値は、LRECL+4 より小さくてはなりません。

ADACMP EBAND を除くすべてのシーケンシャルファイルの RECFORM は VARBLK です。ADACMP EBAND については、コントロールステートメントの RECFM パラメータで与えられます。

VSE メッセージ 4140D が、複数ボリュームのテープファイルの最初のボリュームを参照しているのか、それとも以降のボリュームなのかを見分けるために、テープファイルのオープン時にオペレータに対してメッセージ ADAI31 が出力されますが、ボリューム終了時には出力されません。

### VSE のシーケンシャル入力ファイルの連結

Adabas ユーティリティに対して複数データセットを入力媒体として使用したい場合、いくつかの OS (z/OS など) では連結機能を使用できます。

VSE ではこの機能に類似したものとして、元の LINK 名に 2 桁の連番 (01~99) を加えた LINK 名を指定した FILE ステートメントを加えることができます。

```
// DLBL TEST  , 'firstfile'  
// EXTENT ...  
// DLBL TEST01, 'secondfile'  
// EXTENT ...  
...  
// DLBL TEST99, 'lastfile'  
// EXTENT ...
```

連結オプションのある入力ファイル进行处理する場合、1つの入力ファイルのエンドオブファイルで、次のデータセットに対する FILE ステートメントが存在するかどうかのチェックが行われます。存在しない場合、シーケンシャル GET コールは EOF を返します。存在すれば、現在オープンしているデータセットがクローズされて次のファイルに対してオープンが試みられます。

このようにして連結するファイルは、ファイル特性 (ブロックサイズ、レコードフォーマット、およびレコードサイズ) が同じである必要があります。

この連結の仕様は、逆順処理されるファイルに対しても適用されます。LINK 名の順番は作成時と逆の順番になります。例えば、ADARES で BACK を使用する場合は次のようになります。

```
// DLBL BACK  , 'lastfile'  
// EXTENT ...  
// DLBL BACK01, 'filebeforelast'  
// EXTENT ...  
...  
// DLBL BACKnn, 'firstfile'  
// EXTENT ...
```

各ボリュームを別々の FILE ステートメントで指定すれば、この機能を用いて複数ボリュームファイルの逆処理もできることにご注目ください。

以下は、連結オプションを使用する場合の LINK 名とユーティリティの一覧です。

DELn (n=1~8)	ADASAV
EBAND	ADACMP ADALOD ADAMER
FULL	ADASAV
ISN	ADALOD
PLOG	ADAPLP ADASAV
BACK	ADARES
SIIN	ADARES ADASEL
REST1	ADASAV (使用する LINK 名は REST1、REST101、REST102 など)

#### ADARES 付きで連結機能を使用する例

ニュークリアスセッションにおいて、F1、F2、F3 という 3つのプロテクションログファイルが ADARES PLCOPY で作成されています。

セッションをある時点までバックアウトしたい場合、ADARES BACKOUT 機能に次のような FILE ステートメントを使用します。

```
// DLBL BACK  , 'F3'
// EXTENT ...
// DLBL BACK01, 'F2'
// EXTENT ...
// DLBL BACK02, 'F1'
// EXTENT ...
```

セッション中に作成したプロテクションログからデータベースを再生成するには、次のような FILE ステートメントを ADARES REGENERATE 機能に使用します。

```
// DLBL SIIN  , 'F1'
// EXTENT ...
// DLBL SIIN01, 'F2'
// EXTENT ...
// DLBL SIIN02, 'F3'
// EXTENT ...
```

---

# B VSE用のライブラリおよびファイルプロシージャ の例

---

- Adabas のライブラリ (ADAVvLIB) ..... 1222
- Adabas のファイル (ADAVvFIL) ..... 1223

VSE の例では、Adabas ライブラリ (ADAVvLIB) および Adabas ファイル (ADAVvFIL) を定義するプロシージャが、アクセス可能なプロシージャライブラリにカタログ化されていると仮定しています。

プロシージャのカタログ化については、『Adabas インストールマニュアル』の VSE セクションの「ライブラリ/データベース定義用のカタログプロシージャ」を参照してください。

Delta Save Facility を使用したカタログ化プロシージャについては『Adabas Delta Save Facility マニュアル』を参照してください。

この document では、次のトピックについて説明します。

## Adabas のライブラリ (ADAVvLIB)

---

```
// PROC
* ***** *
* LIBRARY DEFINITIONS AND CHAINING FOR ADABAS *
* ***** *
// SETPARM VERS=vrs          <- CURRENT VERSION
// SETPARM ADALIB=SAGLIB     <- SAG PRODUCT LIBRARY
// SETPARM ADASUB=ADA&VERS   <- ADABAS SUBLIBRARY
// DLBL SAGLIB, 'SAG.PRODUCT.LIBRARY'
// EXTENT ,vvvvvv
// LIBDEF *,SEARCH=&ADALIB..&ADASUB,TEMP
// LIBDEF PHASE,CATALOG=&ADALIB..&ADASUB,TEMP
// ASSGN SYS009,PRINTER
```

ここでは次の内容を表しています。

vrs	Adabas バージョン、リビジョン、およびシステムメンテナンス (SM) レベル
vvvvvv	割り当てられたプログラマ論理ユニット

## Adabas のファイル (ADAVvFIL)

```
// ASSGN SYS031,dddd,VOL=ADA001,SHR
// ASSGN SYS032,dddd,VOL=ADA002,SHR
// ASSGN SYS033,dddd,VOL=ADA003,SHR
// ASSGN SYS034,dddd,VOL=ADA004,SHR
// DLBL ASSOR1,'EXAMPLE.ADAyyyyy.ASSOR1',99/365,DA
// EXTENT SYS031,ADA001,,15,1500
// DLBL DATAR1,'EXAMPLE.ADAyyyyy.DATAR1',99/365,DA
// EXTENT SYS032,ADA002,,15,3000
// DLBL WORKR1,'EXAMPLE.ADAyyyyy.WORKR1',99/365,DA
// EXTENT SYS033,ADA003,,15,600
// DLBL PLOGR1,'EXAMPLE.ADAyyyyy.PLOGR1',99/365,DA
// EXTENT SYS034,ADA004,,15,600
// DLBL PLOGR2,'EXAMPLE.ADAyyyyy.PLOGR2',99/365,DA
// EXTENT SYS034,ADA004,,615,600
// DLBL CLOGR1,'EXAMPLE.ADAyyyyy.CLOGR1',99/365,DA
// EXTENT SYS034,ADA004,,1215,750
// DLBL CLOGR2,'EXAMPLE.ADAyyyyy.CLOGR2',99/365,DA
// EXTENT SYS034,ADA004,,1965,750
// DLBL TEMPR1,'EXAMPLE.ADAyyyyy.TEMPR1',99/365,DA
// EXTENT SYS032,ADA002,,3015,1500
// DLBL SORTR1,'EXAMPLE.ADAyyyyy.SORTR1',99/365,DA
// EXTENT SYS033,ADA003,,615,375
// EXTENT SYS034,ADA004,,2715,375
// DLBL RLOGR1,'EXAMPLE.ADAyyyyy.RLOGR1',99/365,DA
// EXTENT SYS033,ADA003,,990,150
```



# 索引

## A

- ABEND34 パラメータ, 1132
- ACTIVATE パラメータ, 337
- ADAACK ユーティリティ, 11
  - ACCHECK 機能, 15
  - BS2000 JCL, 20
  - VSE JCS, 23
  - z/OS JCL, 21
  - z/VM JCL, 22
  - 機能概要, 13
- Adabas
  - ADASAV リリースのサポート, 945
- Adabas 8
  - ADACNV CONVERT ユーティリティサポート, 189
  - ADACNV REVERT ユーティリティサポート, 193
  - ADAORD の考慮事項, 596
  - ADASAV のサポート, 945
  - ADASEL IF ステートメントの有効なインデックス範囲, 1060, 1063
- Adabas Delta Save Facility
  - ステータスの表示
    - ユーティリティの使用, 293
  - データベースステータスレポート, 837
  - 変更フラグ, 837
- Adabas Review
  - ハブ ID の設定または修正, 301
  - 非アクティブ化
    - ユーティリティの使用, 301
  - ローカルモード
    - ユーティリティ使用への切り替え, 301
- Adabas Statistics Facility
  - ADADBS REFRESHSTATS を使用, 320
- Adabas コントロールブロック
  - ロギングの開始
    - ユーティリティの使用, 298
  - ロギングの停止
    - ユーティリティの使用, 299
- Adabas ファイル
  - ADACMP COMPRESS を使用した圧縮, 87
  - ユーティリティ別シーケンシャルリスト, 1207
- ADACDC ユーティリティ, 25
  - ADADBS または ADALOD ISNREUSE でエラー, 38
  - JCL の要件と例, 53
    - BS2000, 54
    - VSE, 58
    - z/OS, 55
    - z/VM, 56
  - オペレーティングシステムの要因, 41
  - BS2000, 43
  - VSE, 43
  - z/OS, 42
  - 機能概要, 27
  - 構文, 36
  - 処理, 28
  - 処理フェーズ, 28
  - 実行, 35
  - 抽出ファイル, 29
  - トランザクションファイル, 33
  - 入力データ, 31
  - パラメータ, 36
  - プライマリ出力ファイル, 30, 32
    - チェックポイント, 30
  - ユーザー出口, 45
    - インストール, 46
    - インターフェイスの説明, 46
    - 呼び出し, 48
    - レコードの更新または追加するために使用, 49
  - 例, 51
- ADACMP ユーティリティ, 61
  - BS2000 JCL の要件と例, 153
  - COMPRESS 機能, 64, 87
  - COMPRESS 機能で圧縮されたデータレコードの出力, 166
  - COMPRESS 機能で拒否されたデータレコードの出力, 166
  - COMPRESS 機能の出力, 165
  - COMPRESS 機能のストレージ要件レポート, 173
  - COMPRESS 例, 97
  - DECOMPRESS 機能, 65, 99
  - DECOMPRESS 機能で拒否されたデータレコードの出力, 176
  - DECOMPRESS 機能でのマルチクライアントファイルの処理, 105
  - DECOMPRESS 機能の出力, 175
  - DECOMPRESS 例, 106
  - JCL の要件と例, 151
  - VSE JCL の要件と例, 162
  - z/OS JCL の要件と例, 156
  - z/VM JCL の要件と例, 160
  - 可変長フィールドのサイズ, 73
  - 再起動に関する考慮事項, 85
  - 処理, 75
    - データ圧縮, 82
    - データの検証, 82
    - 入力データの構造, 68
    - 入力データの必要条件, 67
    - ピリオディックグループカウントの入力条件, 70
    - マルチプルバリュースフィールドカウント, 68
  - ユーザー出口 6, 85
  - ユーザー出口 6 を使用した圧縮, 152

- ユーザー出口を使用した照合, 153
- ADACNV ユーティリティ, 181
  - Adabas 8 逆変換サポート, 193
  - Adabas 8 変換サポート, 189
  - CONVERT 機能, 187
  - JCL の要件と例, 195
    - BS2000, 196
    - VSE, 201
    - z/OS, 199
    - z/VM, 200
  - REVERT 機能, 191
  - 機能概要, 183
- ADADBS ISNREUSE
  - ADACDC ユーティリティでエラー, 38
- ADADBS REPLICATION 機能
  - ACTIVATE パラメータ, 337
  - DEACTIVATE パラメータ, 337
  - DSBI パラメータ, 337
  - FILE パラメータ, 336
  - KEY パラメータ, 337
  - MODIFY パラメータ, 336
  - NOKEY パラメータ, 337
  - OFF パラメータ, 336
  - ON パラメータ, 336
  - TARGET パラメータ, 338
- ADADBS ユーティリティ, 203
  - ADD データセット機能, 207
  - ALLOCATE 機能, 211
  - CHANGE 機能, 215
  - CVOLSER 機能, 219
  - DEALLOCATE 機能, 223
  - DECREASE 機能, 227
  - DELCP 機能, 231
  - DELETE 機能, 235
  - DSREUSE 機能, 239
  - ENCODEF 機能, 243
  - INCREASE 機能, 247
    - BS2000 プロシージャ, 254
    - VSE プロシージャ, 250
    - z/OS 手順, 249
    - z/VM プロシージャ, 252
    - 一般手順, 249
  - ISNREUSE 機能, 255
  - JCL の要件と例
    - BS2000, 360
    - VSE, 363
    - z/OS, 361
    - z/VM, 362
  - MODFCB 機能, 259
  - MUPEX 機能, 263
  - NEWFIELD 機能, 267
  - ONLINVERT 機能, 271
  - ONLREORFASSO 機能, 275
  - ONLREORFDATA 機能, 279
  - ONLREORFILE 機能, 283
  - OPERCOM 機能, 287
  - PRIORITY 機能, 305
  - RECORDSPANNING 機能, 309
  - RECOVER 機能, 313
  - REFRESH 機能, 315
  - REFRESHSTATS 機能, 319
  - RELEASE 機能, 323
  - RENAME 機能, 327
  - RENUMBER 機能, 331
  - REPLICATION 機能, 335
  - RESETDIB 機能, 339
  - RESETPPT 機能, 343
  - SPANCOUNT 機能, 347
  - TRANSACTIONS 機能, 351
  - UNCOUPLE 機能, 355
  - 機能概要, 205
  - 構文チェック, 206
- ADADCK ユーティリティ, 365
  - DSCHECK 機能, 369
  - JCL の要件と例
    - BS2000, 374
    - VSE, 377
    - z/OS, 375
    - z/VM, 376
- ADADEF ユーティリティ, 379
  - DEFINE 機能, 383
  - JCL の要件と例
    - BS2000, 402
    - VSE, 407
    - z/OS, 403
    - z/VM, 405
  - MODIFY 機能, 393
  - NEWWORK 機能, 397
  - 機能概要, 381
    - チェックポイントファイル, 382
    - データベースコンポーネント, 382
- ADAEND オペレータコマンド
  - ユーティリティの使用, 290
- ADAFRM ユーティリティ, 409
  - JCL の要件と例
    - BS2000, 420
    - VSE, 425
    - z/OS, 422
    - z/VM, 423
  - 新しい RABN のフォーマット, 412
  - 機能概要, 411
  - すべての機能, 414, 415
  - データセットブロック/シリンダを 0 にリセット, 412
  - データベースコンポーネントのフォーマット, 412
- ADAICK ユーティリティ, 427
  - ACCHECK 機能, 431
  - ASSOPRINT 機能, 435
  - BATCH 機能, 437
  - DATAPRINT 機能, 439
  - DSCHECK 機能, 441
  - DUMP 機能, 445
  - FCBPRINT 機能, 447
  - FDTPRINT 機能, 451
  - GCBPRINT 機能, 453
  - ICHECK 機能, 455
  - INT 機能, 457
  - JCL の要件と例
    - BS2000, 476
    - VSE, 480
    - z/OS, 478
    - z/VM, 479
  - NIPRINT 機能, 459
  - NOBATCH 機能, 461
  - NODUMP 機能, 463
  - NOINT 機能, 465
  - PPTPRINT 機能, 467
  - UIPRINT 機能, 471
  - 機能概要, 429

- ユーザー出口照合, 476
- 例, 473
- ADAINV ユーティリティ, 481
- COUPLE 機能, 485
- INVERT 機能, 493
- JCL の要件と例
  - BS2000, 500
  - VSE, 507
  - z/OS, 503
  - z/VM, 505
- 機能概要, 483
- 実行中のスペース割り当て, 497
- ユーザー出口
  - 照合, 360, 500
  - 例, 498
- ADALOD ISNREUSE
  - ADACDC ユーティリティでエラー, 38
- ADALOD ユーティリティ, 509
- JCL の要件と例
  - BS2000, 562
  - VSE, 571
  - z/OS, 566
  - z/VM, 568
- LOAD 機能, 513
  - 入力データ, 531
  - ファイルに対するスペース割り当て, 531
  - 例, 530
- LOAD 機能を使用したアソシエータの更新, 535
- UPDATE 機能, 539
  - アソシエータの更新, 551
  - 拡張ファイルの一括更新, 551
  - スペース割り当て, 550
  - ディスクリプタ情報の生成, 551
  - 入力要件, 550
  - 例, 547
  - ストレージの要件および使用, 553
  - スペース/統計レポート, 559
  - 中間データセットの要件, 555
  - ユーザー出口
    - 照合, 562
- ADAM
  - ADAMER ユーティリティからの検索統計, 575
  - ファイルのロード
    - ユーティリティの使用, 517
- ADAMER ユーティリティ, 575
- JCL の要件と例
  - BS2000, 586
  - VSE, 589
  - z/OS, 587
  - z/VM, 588
- 構文, 579
- 出力レポート, 583
- 例, 582
- ADAORD ユーティリティ, 591
- Adabas 8 の考慮事項, 596
- JCL の要件と例
  - BS2000, 672
  - VSE, 683
  - z/OS, 677
  - z/VM, 680
- REORASSO 機能, 597
- REORDATA 機能, 605
- REORDB 機能, 611
- REORFASSO 機能, 623
- REORFDATA 機能, 631
- REORFILE 機能, 637
- RESTRUCTUREDDB 機能, 647
- RESTRUCTUREF 機能, 653
- STORE 機能, 659
- ADAPLP ユーティリティ, 687
- JCL の要件と例
  - BS2000, 700
  - VSE, 709
  - z/OS, 704
  - z/VM, 707
- PLOGPRI 機能, 691
- SPLOGPRI 機能, 691
- WORKPRI 機能, 691
- 例, 696
- ADAPRI ユーティリティ, 713
- ASSOPRI 機能, 717
- CLOGPRI 機能, 717
- DATAPRI 機能, 717
- JCL の要件と例
  - BS2000, 722
  - VSE, 725
  - z/OS, 723
  - z/VM, 724
- PLOGPRI 機能, 717
- RLOGPRI 機能, 717
- SORTPRI 機能, 717
- TEMPPRI 機能, 717
- WORKPRI 機能, 717
- 例, 719
- ADAR2E ユーティリティ
  - BS2000 パブセットメンバのリカバリ, 791
- ADARAI ユーティリティ, 727
- BS2000 RECOVER スケルトンジョブ制御の入力, 780
- BS2000 での LIST 機能の出力例, 741
- BS2000 における特別な考慮事項, 776
- CHKDB 機能, 733
- DISABLE 機能, 735
- JCL の要件と例
  - BS2000, 776
- LIST 機能, 737
- PREPARE 機能, 751
- RECOVER 機能, 755
- RECOVER 機能の BS2000 スケルトンジョブ制御入力, 785
- RECOVER 機能の z/OS スケルトンジョブ制御入力, 794
- RECOVER 機能の出力ジョブストリーム, 760
- RECOVER 機能の実行, 761
- RECOVER 機能のスケルトンジョブ制御入力, 767
- RECOVER 機能の入力データセット, 760
- REMOVE 機能, 773
- VSE JCL の要件と例, 800
- VSE での LIST 機能の出力例, 748
- VSE における特別な考慮事項, 800
- z/OS JCL の要件と例, 792
- z/OS での LIST 機能の出力例, 742
- z/VM JCL の要件と例, 798
- z/VM における特別な考慮事項, 798
- 概念, 730
- 機能ディレクトリ, 727
- 中断後の RECOVER 機能の再スタート, 771
- ファイルレベルのリカバリ, 762
- ADAREP ユーティリティ, 803
- BS2000 JCL の要件と例, 850
- REPORT 機能, 803

- VSE JCL の要件と例, 853
  - z/OS JCL の要件と例, 851
  - z/VM JCL の要件と例, 852
  - 例, 811
  - ADARES ユーティリティ, 857
  - BACKOUT DPLOG 機能, 875
  - BACKOUT MPLOG 機能, 875
  - BACKOUT 機能, 867
  - BS2000 JCL の要件と例, 916
  - CLCOPY 機能, 883
  - COPY 機能, 887
  - MERGE CLOG 機能, 891
  - PLCOPY 機能, 893
  - REGENERATE 機能, 899
  - REPAIR 機能, 909
  - VSE JCL の要件と例, 935
  - z/OS JCL の要件と例, 924
  - z/VM JCL の要件と例, 931
  - 機能ディレクトリ, 860
  - ADASAV ユーティリティ, 941
  - Adabas リリースのサポート, 945
  - BS2000 JCL の要件と例, 1026
  - RESTONL FILES 機能, 959
  - RESTONL FMOVE 機能, 967
  - RESTONL GCB 機能, 953
  - RESTONL 機能, 947
  - 概要, 944
  - RESTONL 機能の概要, 945
  - RESTORE FILES 機能, 991
  - RESTORE FMOVE 機能, 997
  - RESTORE GCB 機能, 985
  - RESTORE 機能, 979
  - RESTORE 機能の概要, 944
  - RESTPLOG 機能, 1009
  - RESTPLOG 機能の概要, 945
  - SAVE FILES 機能, 1019
  - SAVE 機能, 1013
  - SAVE 機能の概要, 945
  - VSE JCL の要件と例, 1040
  - z/OS JCL の要件と例, 1032
  - z/VM JCL の要件と例, 1037
  - ADASEL ユーティリティ, 1045
  - Adabas 8 IF ステートメントの有効なインデックス範囲, 1060, 1063
  - BS2000 JCL の要件と例, 1072
  - DISPLAY 命令の構文, 1061
  - IF ステートメントの構文, 1054
  - NEWPAGE 命令の構文, 1067
  - OUTPUT 命令の構文, 1065
  - SKIP 命令の構文, 1067
  - VSE JCL の要件と例, 1075
  - z/OS JCL の要件と例, 1073
  - z/VM JCL の要件と例, 1074
  - グローバルパラメータ, 1069
  - 構文, 1049
  - SELECT ステートメント, 1051
  - value criterion, 1056
  - WITH 節, 1053
  - 出力命令, 1061
  - 追加の選択条件, 1052
  - 日付/時刻, 1053
  - ADAULD ユーティリティ, 1077
  - BS2000 JCL の要件と例, 1100
  - JCL の要件と例
  - VSE, 1106
  - LOAD FILE 機能の DDISN パラメータ, 1083
  - UNLOAD FILE 機能, 1081
  - z/OS JCL の要件と例, 1102
  - z/VM JCL の要件と例, 1104
  - 出力処理, 1095
  - 入力処理, 1091
  - ユーザー出口 9 の処理, 1097
  - ADAVAL ユーティリティ, 1109
  - BS2000 JCL の要件と例, 1121
  - VALIDATE 機能, 1113
  - VSE JCL の要件と例, 1124
  - z/OS JCL の要件と例, 1122
  - z/VM JCL の要件と例, 1123
  - 拒否された ISN, 1111
  - 出力例, 1117
  - ADAWRK ユーティリティ, 1127
  - VSE JCL の要件と例, 1189
  - z/OS JCL の要件と例, 1186
  - z/VM JCL の要件と例, 1191
  - 概要, 1129
  - 構文, 1131
  - レポート, 1137
  - ADAZAP ユーティリティ, 1193
  - BS2000 JCL の要件と例, 1202
  - VSE JCL の要件と例, 1205
  - z/OS JCL の要件と例, 1203
  - z/VM JCL の要件と例, 1204
  - ALOCKF オペレータコマンド
  - ユーティリティの使用, 290
  - ATM
  - システムファイルのロード, 516
- ## B
- BS2000
  - ADARAI 使用時の特別な考慮事項, 776
  - 失われたバブセットメンバをリカバリする ADAR2E ユーティリティ, 791
  - リカバリログの追加情報, 739
- ## C
- CANCEL オペレータコマンド
  - ユーティリティの使用, 291
  - CHECKPOINT パラメータ, 1132
  - CMID パラメータ, 1132
  - CT
  - ADARUN パラメータ
  - 設定を上書きするコマンド, 292
  - CT オペレータコマンド
  - ユーティリティの使用, 292
- ## D
- DAUQ オペレータコマンド
  - ユーティリティの使用, 293
  - DCQ オペレータコマンド
  - ユーティリティの使用, 293
  - DD/JCLIN
  - ADARAI の要件, 760
  - DDIB オペレータコマンド
  - ユーティリティの使用, 293
  - DDSF オペレータコマンド

ユーティリティの使用, 293  
 DEACTIVATE パラメータ, 337  
 DFILES オペレータコマンド  
   ユーティリティの使用, 294  
 DFILUSE オペレータコマンド  
   ユーティリティの使用, 294  
 DHQA オペレータコマンド  
   ユーティリティの使用, 294  
 DLOCKF オペレータコマンド  
   ユーティリティの使用, 294  
 DNC オペレータコマンド  
   ユーティリティの使用, 294  
 DNH オペレータコマンド  
   ユーティリティの使用, 295  
 DNU オペレータコマンド  
   ユーティリティの使用, 295  
 DONLSTAT オペレータコマンド  
   ユーティリティの使用, 295  
 DPARM オペレータコマンド  
   ユーティリティの使用, 295  
 DRES オペレータコマンド  
   ユーティリティの使用, 295  
 DSBI パラメータ, 337  
 DSTAT オペレータコマンド  
   ユーティリティの使用, 296  
 DTH オペレータコマンド  
   ユーティリティの使用, 296  
 DUQ オペレータコマンド  
   ユーティリティの使用, 296  
 DUQA オペレータコマンド  
   ユーティリティの使用, 296  
 DUQE オペレータコマンド  
   ユーティリティ, 296  
 DUUQE オペレータコマンド  
   ユーティリティの使用, 296

## E

ETID パラメータ, 1132

## F

FEOFCL オペレータコマンド  
   ユーティリティの使用, 297  
 FEOFPL オペレータコマンド  
   ユーティリティの使用, 297  
 FILE パラメータ, 336  
 FILES パラメータ, 1132  
 FORCE パラメータ, 1132

## H

HALT オペレータコマンド  
   ユーティリティの使用, 297

## I

I/O アクティビティ  
   ログインの開始  
     ユーティリティの使用, 298  
   ログインの停止  
     ユーティリティの使用, 299  
 IF ステートメント  
   有効なインデックス範囲, 1060, 1063

## ISN

アンロードするファイル, 1083  
 再使用のために設定  
   ユーティリティの使用, 255  
 指定形式, 548  
 ISN バッファ  
   ログインの開始  
     ユーティリティの使用, 298  
   ログインの停止  
     ユーティリティの使用, 299  
 ISN プールのテーブル  
   使用方法を表示するコマンド, 295

## K

KEY パラメータ, 337

## L

LOCKF オペレータコマンド  
   ユーティリティの使用, 298  
 LOCKU オペレータコマンド  
   ユーティリティの使用, 298  
 LOCKX オペレータコマンド  
   ユーティリティの使用, 298  
 LOGGING  
   オペレータコマンド  
     ユーティリティの使用, 298  
 LOGxx  
   オペレータコマンド  
     ユーティリティの使用, 298  
 LWP パラメータ, 1132

## M

MODIFY パラメータ, 336  
 MU フィールド  
   ファイルに最大数の設定, 263

## N

NOKEY パラメータ, 337  
 NOLOGGING  
   オペレータコマンド  
     ユーティリティの使用, 299  
 NOLOGxx  
   オペレータコマンド  
     ユーティリティの使用, 299  
 NOPPT パラメータ, 1133  
 NOUSERABEND パラメータ, 1133

## O

OFF パラメータ, 336  
 ON パラメータ, 336  
 ONLRESUME  
   オペレータコマンド  
     ユーティリティの使用, 299  
 ONLSTOP  
   オペレータコマンド  
     ユーティリティの使用, 300  
 ONLSUSPEND  
   オペレータコマンド

ユーティリティの使用, 300

## P

Parallel Participant Table (PPT)

出力/ダンプ

ユーティリティの使用, 467

PE グループ

ファイルに最大数の設定, 263

PPT

アソシエータデータセットでのリセット, 343

## R

RALOCKF オペレータコマンド

ユーティリティの使用, 300

RALOCKFA オペレータコマンド

ユーティリティの使用, 301

RDUMPST オペレータコマンド

ユーティリティの使用, 301

READONLY

オペレータコマンド

ユーティリティの使用, 301

Recovery Aid (参照 ADARAI ユーティリティ)

REPORTFILE パラメータ, 1133

RESTONL 機能, 945

RESTPLOG 機能, 945

REVIEW

オペレータコマンド

ユーティリティの使用, 301

RLOG (参照 リカバリログ)

## S

SQL

空値表現のサポート, 121

STOPF

オペレータコマンド

ユーティリティの使用, 301

STOPI

オペレータコマンド

ユーティリティの使用, 302

STOPU

オペレータコマンド

ユーティリティの使用, 302

SUMMARY パラメータ, 1133

SYNCC

オペレータコマンド

ユーティリティの使用, 303

## T

TARGET パラメータ, 338

TEST パラメータ, 1133

TIMEZONE パラメータ, 1134

TNAA

オペレータコマンド

ユーティリティの使用, 303

TNAE

オペレータコマンド

ユーティリティの使用, 303

TNAX

オペレータコマンド

ユーティリティの使用, 303

TRANSACTIONS パラメータ, 1134

TSOSCAT (BS2000)

ADAR2E ユーティリティを使用したメンバの読み込み, 791

TT

オペレータコマンド

ユーティリティの使用, 303

## U

UNLOCKF

オペレータコマンド

ユーティリティの使用, 303

UNLOCKU

オペレータコマンド

ユーティリティの使用, 304

UNLOCKX

オペレータコマンド

ユーティリティの使用, 304

USERID パラメータ, 1134

UTIONLY

オペレータコマンド

ユーティリティの使用, 304

## V

VSE

ADARAI 使用時の特別な考慮事項, 800

ライブラリおよびファイルプロシージャの例, 1221

## W

WORK データセット

ADAFRM を使用したフォーマット, 415

ADAFRM を使用してブロック数/シリンダ数を0にリセット, 415

ADAPLP を使用してデータプロテクションレコードを出力します, 692

ADAPRI を使用したブロックの出力, 715

## X

XID プール

使用方法を表示するコマンド, 296

## Z

z/VM

ADARAI 使用時の特別な考慮事項, 798

## あ

アソシエータ

ADAFRM を使用したフォーマット, 414

ADAICK を使用して物理構造をチェックします。 , 429

ADALOD LOAD を使用した更新, 535

ADAORD REORASSO を使用したデータベースのリオーダ, 597

ADAORD REORFASSO を使用したファイルのリオーダ, 623

ADAPRI を使用したブロックの出力, 715

ADAVAL VALIDATE を使用したデータストレージ内容に対する値のチェック, 1113

PPT のリセット, 343

カップリングリスト  
   作成, 489  
 更新  
   ユーティリティの使用, 551  
 データセットのサイズ減少  
   ユーティリティの使用, 227  
 データセットのサイズ増加  
   ユーティリティの使用, 247  
 データセットの追加  
   ユーティリティの使用, 208  
 ブロックの出力/ダンプ  
   ユーティリティの使用, 435  
 ブロック/シリンダを 0 にリセット  
   ユーティリティの使用, 415  
 リオーダー  
   ユーティリティの使用, 594  
 アソシエータのリオーダー  
   オンライン処理の開始  
   ユーティリティの使用, 275  
 アタッチドバッファ  
   使用方法を表示するコマンド, 295  
 アPPERインデックス  
   エクステントの割り当て  
   ユーティリティの使用, 212  
   エクステントの割り当て解除  
   ユーティリティの使用, 224  
   出力/ダンプ  
   ユーティリティの使用, 471  
 アドレスコンバータ  
   ADAICK によるインデックスのチェック, 455  
   ADAICK を使用した特定ファイルの検証, 427  
   エクステントの割り当て  
   ユーティリティの使用, 212  
   エクステントの割り当て解除  
   ユーティリティの使用, 224  
   スペース割り当て  
   ユーティリティの使用, 533, 550  
   データストレージに対するチェック  
   ユーティリティの使用, 15

## い

インデックス  
   ADAICK によるアドレスコンバータのチェック, 455  
   ADAICK を使用した特定ファイルの検証, 427  
   カップリングリストのスペース割り当て, 491  
   スペース割り当て  
   ADAINV による, 497  
   ユーティリティの使用, 532  
 インバート  
   オンライン処理の開始  
   ユーティリティの使用, 271

## え

英数字フィールド  
   無変換オプション (NV) , 120  
 エンコード  
   変更  
   ユーティリティの使用, 393

## お

大文字, 5

大文字と小文字が混在, 5  
 オペレータコマンド  
   ADADBS OPERCOM 機能, 287  
 オンラインインバート  
   開始  
   ユーティリティの使用, 271  
 オンライン処理  
   クリーンに停止する  
   ユーティリティの使用, 300  
   ステータスを表示  
   ユーティリティの使用, 295  
   中断された処理を再開する  
   ユーティリティの使用, 299  
   中断する  
   ユーティリティの使用, 300  
 オンラインリオーダー  
   アソシエータ, 275  
   データストレージ, 279  
   ファイル, 283

## か

解釈済みの出力形式  
   アクティブ化, 457  
   省略, 465  
 角カッコ ([ ]), 6  
 拡張ファイル  
   アンカーファイルのロード, 519  
   一括更新, 551  
   ロード, 535  
 拡張ファイルチェーン  
   ディスクリプタがユニークであるかどうかのチェック, 120  
 下線, 6

## <

空値  
   SQL サポート, 121  
   指定不可 (NN) フィールドオプション, 124  
   省略 (NU) フィールドオプション, 119  
   非カウント (NC) フィールドオプション, 122  
 空値インジケータ  
   レコードバッファに指定, 123  
 空白圧縮  
   NB フィールドオプション, 118  
 繰り返しの構文要素, 7  
 グループ定義, 108

## こ

構文形式  
   大文字, 5  
   大文字と小文字が混在, 5  
   角カッコ ([ ]), 6  
   下線, 6  
   区切り文字と記号, 7  
   繰り返しの要素, 7  
   小文字, 5  
   サブパラメータ, 6  
   斜体, 5  
   省略記号 (...), 7  
   字下げ, 6  
   選択型の要素, 6  
   相互排他的な選択肢, 6

縦線 (|) , 6  
 短縮キーワード, 6  
 中カッコ ({}), 6  
 デフォルト, 6  
 必須の要素, 6  
 標準フォント, 5  
 太字, 5  
 構文中での選択肢, 6  
 構文中の区切り文字と記号, 7  
 固定ストレージ (FI)  
   ADACMP での使用, 114  
 コマンドキュー  
   使用方法を表示するコマンド, 295  
 コマンドキューエレメント  
   表示  
     ユーティリティの使用, 294  
     ポストされたコマンドキューエレメントの数を表示  
       ユーティリティの使用, 293  
 コマンドログ  
   ADAFRM を使用したフォーマット, 415  
   ADAPRI を使用した複数 CLOG からのブロックの出力, 715  
   ADARESCLCOPY を使用したデュアルデータセットのシー  
   ケンシャルデータセットへのコピー, 883  
   デュアルのクローズ/切り替え  
     ユーティリティの使用, 297  
   データセットの追加  
     ユーティリティの使用, 208  
   複数をマージする, 891, 927  
   ロギングの開始  
     ユーティリティの使用, 298  
   ロギングの停止  
     ユーティリティの使用, 299  
 小文字, 5

## さ

再実行プール  
 使用方法を表示するコマンド, 296  
 サブディスクリプタ  
   ADACMP COMPRESS を使用した定義, 136  
   定義  
     ADAINV の使用, 494  
 サブパラメータ  
   指定, 9  
 サブパラメータの構文, 6  
 サブフィールド  
   ADACMP COMPRESS を使用した定義, 138  
 サマリレポート, 1139  
 サーチャッファ  
   ロギングの開始  
     ユーティリティの使用, 298  
   ロギングの停止  
     ユーティリティの使用, 299

## し

斜体, 5  
 出力  
   ADAICK を使用して幅を 132 文字に設定, 437  
   解釈済みの形式, 457  
   解釈済みの形式の省略, 465  
   出力形式ダンプ, 445  
   ダンプ出力形式の省略, 463  
   幅を 80 文字に設定

ユーティリティの使用, 461  
 出力形式ダンプ  
   アクティブ化, 445  
   省略, 463  
 照合ディスクリプタ  
   ADACMP COMPRESS での定義, 128  
   定義  
     ADAINV の使用, 494  
 省略記号 (...), 7  
 シーケンシャルコマンドプールのテーブル  
   使用方法を表示するコマンド, 295  
 ジェネラルコントロールブロック (GCB)  
   出力/ダンプ  
     ユーティリティの使用, 453  
 字下げ, 6

## す

ストレージ  
   固定 (FI) , 114  
 スパンドレコード  
   カウント, 347  
   ファイルの有効化/無効化, 309  
 スペース  
   ADAINV を使用したファイルカップリングリストの計算,  
   490  
   一時的なファイルカップリング, 488  
   回復  
     ユーティリティの使用, 313  
   見積もりレポート (ADACMP) , 173  
 スレッド  
   ステータスの表示  
     ユーティリティの使用, 296  
 スーパーディスクリプタ  
   ADACMP COMPRESS を使用した定義, 139  
   定義  
     ADAINV の使用, 494  
 スーパーフィールド  
   ADACMP COMPRESS を使用した定義, 148

## せ

セキュリティプール  
 使用方法を表示するコマンド, 295  
 世代, 731  
 セッション  
   現パラメータの表示  
     ユーティリティの使用, 295  
   すぐにキャンセル  
     ユーティリティの使用, 291  
   正常終了  
     ユーティリティの使用, 290  
   停止  
     ユーティリティの使用, 297  
   統計値のリセット  
     ユーティリティの使用, 319  
 選択型の構文要素, 6  
 セーブテープ  
   ADARES COPY ユーティリティを使用したコピー, 887  
 全般的なファイルステータス, 823

## そ

ソートデータセット

ADAFRM を使用したフォーマット、415  
ADAPRI を使用したブロックの出力、715

## た

対応パラメータ  
指定、9  
タイムアウトコントロール  
ET ロジックユーザーに対する非アクティビティリミットの設定、303  
アクセスオンリーユーザーに対する非アクティビティリミットの設定、303  
排他制御ユーザーに対する非アクティビティリミットの設定、303  
リージョン間コミュニケーション制限  
設定を上書きするコマンド、292

縦線 (|)、6

短縮キーワード、6

ダンプ

オンラインステータスの終了  
ユーティリティの使用、301

## ち

チェックポイント

Adabas ニュークリアス/ユーティリティによる書き込み、845  
削除

ユーティリティの使用、231

データベースステータスレポート、843

チェックポイントファイル

定義、383

チェックポイントレコードのレポート、1184

中カッコ ({}), 6

中間データセット

ADAFRM を使用したフォーマット、415

ADALOD 要件、555

ADAPRI を使用したブロックの出力、715

スペース割り当て

ユーティリティの使用、534

## つ

通常処理の中断

ADADBS 機能、351

通常処理を再開する

ADADBS 機能、351

## て

ディスクボリューム

指定箇所にある Adabas エクステンツの出力

ユーティリティの使用、219

ディスクリプタ

サブディスクリプタ、136

照合、128

スーパーディスクリプタ、139

定義

ADAINV ユーティリティの使用、481, 493

ディスクリプタステータスからの解除

ユーティリティの使用、323

ハイパーディスクリプタ、131

ファイルカップリングの指定

ユーティリティの使用、485

フィールドオプション (DE)、113

フォネティック、135

ユニーク、120

デフォルトのパラメータ値、6

データ圧縮

ADACMP ユーティリティ、64, 82

NC オプションありのフィールド、122

データストレージ

ADAFRM を使用したフォーマット、414

ADAFRM を使用してブロック数/シリンダ数を0にリセット、415

ADAICK によるブロックの出力/ダンプ、439

ADAICK によるレコードの出力/ダンプ、441

ADAORD REORDATA を使用したデータベースのリオーダ、605

ADAORD REORFDATA を使用したファイルのリオーダ、631

ADAPRI を使用したブロックの出力、715

ADAVAL VALIDATE を使用したアソシエータ値に対するチェック、1113

アドレスコンバータのチェック

ユーティリティの使用、15

エクステンツの割り当て

ユーティリティの使用、212

エクステンツの割り当て解除

ユーティリティの使用、224

指定されたファイルのチェック

ユーティリティの使用、365

スペース割り当て

ユーティリティの使用、533

データセットのサイズ減少

ユーティリティの使用、227

データセットのサイズ増加

ユーティリティの使用、247

データセットの追加

ユーティリティの使用、208

ブロックの再使用

ユーティリティの使用、239

ブロックを修復する

ユーティリティの使用、909

リオーダ

ユーティリティの使用、594

データストレージのリオーダ

オンライン処理の開始

ユーティリティの使用、279

データセット

ADAPRI を使用したブロックの出力、715

中間カップリングストレージ

ユーティリティを使用した計算、488

追加/更新

ユーティリティの使用、509

フォーマット

ユーティリティの使用、409

データ定義

ADACMP COMPRESS の COLDE ステートメント、128

ADACMP COMPRESS の FNDEF ステートメント、108

ADACMP COMPRESS の HYPDE ステートメント、131

ADACMP COMPRESS の PHONDE ステートメント、135

ADACMP COMPRESS の SUBDE ステートメント、136

ADACMP COMPRESS の SUPDE ステートメント、139

ADACMP を使用する構文形式、107

COLDE ステートメント

ADAINV INVERT、495

DE - ディスクリプタフィールドオプション、113

- FI - 固定ストレージフィールドオプション, 114
- FIELD ステートメント
  - ADAINV INVERT, 495
- HYPDE ステートメント
  - ADAINV INVERT, 495
- LA - ロング英数字フィールドオプション, 114
- LB - ラージオブジェクトフィールドオプション, 115
- MU - マルチプルバリュースフィールドオプション, 116
- NB - 空白圧縮フィールドオプション, 118
- NC - 圧縮/圧縮解除でカウントされない空値, 122
- NC - 空値非カウント SQL フィールドオプション, 122
- NN - 非空値 SQL フィールドオプション, 124
- NV - 無変換フィールドオプション, 120
- PE - ビリオディックグループフィールドオプション, 126
- PHONDE ステートメント
  - ADAINV INVERT, 495
- SUBDE ステートメント
  - ADAINV INVERT, 495
- SUBFN ステートメント, 138
- SUPDE ステートメント
  - ADAINV INVERT, 495
- SUPFN ステートメント, 148
- U - 空値省略フィールドオプション, 119
- UQ - ユニークディスクリプタフィールドオプション, 120
- XI - UQ フィールドオプションから PE インスタンスを除外, 121
- フィールドオプション, 112
- データの圧縮解除
  - ADACMP ユーティリティ, 65, 99
  - NC オプションありのフィールド, 122
- データベース
  - ADAORD REORASSO を使用したアソシエータのリオーダ, 597
  - ADAORD REORDATA を使用したデータストレージのリオーダ, 605
  - ADAORD REORDB を使用したデータストレージおよびアソシエータのリオーダ, 611
  - ADAORD RESTRUCTUREDB を使用したシーケンシャルデータセットへのアンロード, 647
  - ADAORD RESTRUTCTUREDB を使用したシーケンシャルデータセットへのアンロード, 595
  - ADAORD STORE を使用したファイルのロード, 595
  - ADARAI RECOVER を使用したリカバリ, 755
  - ADARAI を使用したリカバリの自動化および最適化, 730
  - ADAREP REPORT を使用したステータスレポートの作成, 805
  - ADARES REGENERATE を使用して 2 つのチェックポイント間で実行された更新のリストア, 899
  - ADASAV RESTONL GCB を使用したオンラインソースからの増分のリストア, 953
  - ADASAV RESTONL を使用したオンラインソースからのリストア, 947
  - ADASAV RESTORE GCB を使用したオフラインソースからの増分のリストア, 985
  - ADASAV RESTORE を使用したオフラインソースからのリストア, 979
  - ADASAV SAVE を使用したセーブ, 1013
  - ADASAV を使用したセーブ/リストア, 941
  - ADAVAL を使用した整合性チェック, 1109
  - ADAZAP ユーティリティを使用した物理ブロックの修正, 1195
  - アクティブ/非アクティブステータスのチェック
    - ユーティリティの使用, 733
  - コンポーネントデータセット
    - フォーマット, 409
  - 新規定義
    - ユーティリティの使用, 379
  - 静止, 351
  - 定義, 383
  - ファイル削除開始
    - ユーティリティの使用, 235
  - 変換, 181
  - リカバリ
    - ユーティリティ, 857
  - リストア用ジョブストリームの構築, 755
  - 割り当てられた名前の変更
    - ユーティリティの使用, 327
  - データベースコンポーネントのスペース割り当て, 822
  - データベースの逆変換
    - 下位バージョン, 191
  - データベースの静止
    - ADADBS 機能, 351
  - データベースの全般的な情報, 820
  - データベースの変換
    - 下位バージョン, 191
    - 上位バージョンへ, 187
  - データ保全ブロック
    - エントリの表示
      - ユーティリティの使用, 293
    - エントリのリセット
      - ユーティリティの使用, 339
- と
- 特殊ディスクリプタ, 842
- トランザクション
  - ET ロジックユーザーに対するタイムリミットの設定
    - ユーティリティの使用, 303
  - トランザクション ID (XID) プール
    - 使用方法を表示するコマンド, 296
  - トランザクション処理
    - 中断/再開, 351
  - トランザクションレポート, 1146
- に
- ニュークリアス
  - 現在のオペレーティングステータスの表示
    - ユーティリティの使用, 296
- の
- ノーマルインデックス
  - エクステントの割り当て
    - ユーティリティの使用, 212
  - エクステントの割り当て解除
    - ユーティリティの使用, 224
  - 出力/ダンプ
    - ユーティリティの使用, 459
- は
- ハイパーディスクリプタ
  - ADACMP COMPRESS での定義, 131
  - 定義
    - ADAINV の使用, 494
- バリュースバッファ
  - ロギングの開始

ユーティリティの使用, 299  
 ログインの停止  
 ユーティリティの使用, 299  
 パラメータ  
 位置の値  
 指定, 9

## ひ

必須の構文要素, 6  
 標準フォント, 5  
 ピリオディックグループ  
 データ定義オプション (PE) , 126  
 ピリオディックグループ定義, 125  
 ADACMP COMPRESS の FNDEF ステートメント, 125

## ふ

ファイル  
 Adabas Delta Save 変更フラグ, 837  
 Adabas シーケンシャルファイルの BS2000 コントロールステートメント読み込み手順, 1214  
 Adabas シーケンシャルファイルの BS2000 テープ解放オプション, 1215  
 Adabas シーケンシャルファイルの BS2000 レコードフォーマット, 1210  
 Adabas シーケンシャルファイルの BS2000 連結入力, 1212  
 Adabas シーケンシャルファイルの VSE レコードフォーマット, 1216  
 Adabas シーケンシャルファイルの z/OS レコードフォーマット, 1215  
 Adabas シーケンシャルファイルの z/VM レコードフォーマット, 1216  
 Adabas シーケンシャルファイルの名前およびブロックの決定, 1208  
 ADACMP COMPRESS を使用した圧縮, 87  
 ADAM オプションを使用したロード, 517  
 ADAORD REORFASSO を使用したアソシエータのリオーダ, 623  
 ADAORD REORFDATA を使用したデータストレージのリオーダ, 631  
 ADAORD REORFILE を使用したアソシエータおよびデータストレージのリオーダ, 637  
 ADAORD RESTRUCTUREDB を使用したシーケンシャルデータセットへのアンロード, 595  
 ADAORD RESTRUCTUREF を使用したシーケンシャルデータセットへのアンロード, 653  
 ADAORD STORE を使用した既存のデータベースへのロード, 595, 659  
 ADARAI RECOVER を使用したリカバリ, 762  
 ADARES REGENERATE を使用して 2 つのチェックポイント間で実行された更新のリストア, 899  
 ADASAV FILES を使用したセーブ, 1019  
 ADASAV RESTONL FILES を使用したオンラインソースから元の RABN へのリストア, 959  
 ADASAV RESTONL FMOVE を使用したオンラインソースから別の RABN へのリストア, 967  
 ADASAV RESTORE FILES を使用したオフラインソースから元の RABN へのリストア, 991  
 ADASAV RESTORE FMOVE を使用したオフラインソースから別の RABN へのリストア, 997  
 ADASAV を使用したセーブ/リストア, 941  
 EXU または EXF ユーザーを除くすべてのユーザーに対してロック, 298

MU または PE の制限指定, 263  
 オプション設定レポート, 835  
 削除  
 ユーティリティの使用, 235  
 指定ファイルのロック解除  
 ユーティリティの使用, 303, 304  
 指定ユーザーの停止  
 ユーティリティの使用, 301  
 処理されたコマンドの合計表示, 294  
 シーケンシャル入力ファイルの VSE 連結, 1218  
 事前ロック, 290  
 事前ロックの解除  
 指定ファイルで, 300  
 全ファイルで, 301  
 情報, 831  
 スパンドレコードのカウント, 347  
 スペース割り当て, 531  
 スペース割り当てレポート, 837  
 全セキュリティレベルでのロック, 298  
 空のステータスにリセット  
 ユーティリティの使用, 315  
 タイプ, 515  
 特殊ディスクリプタレポート, 842  
 パラメータ修正  
 ユーティリティの使用, 259  
 ファイル番号の変更  
 ユーティリティの使用, 331  
 フィールド定義テーブルレポート, 839  
 マルチクライアントファイルの圧縮解除, 105  
 ユーザータイプ数の表示, 294  
 ユーティリティ以外のすべての使用に対してロック, 298  
 ユーティリティ使用のロック解除, 304  
 ユーティリティ別 Adabas シーケンシャルリスト, 1207  
 リストア用ジョブストリームの構築, 755  
 レコードのスパニングの有効化/無効化, 309  
 レプリケーションのアクティブ化または非アクティブ化, 335  
 ロックファイルの表示, 294  
 ロード  
 ユーティリティの使用, 513  
 割り当てられた名前の変更  
 ユーティリティの使用, 327  
 ファイルエクステンツ  
 指定されたディスクボリュームに関する出力  
 ユーティリティの使用, 219  
 割り当て  
 ユーティリティの使用, 211  
 割り当て解除  
 ユーティリティの使用, 223  
 ファイルエンコード  
 変更, 243  
 ファイルオプション, 825, 835  
 ファイルカップリング  
 アンカップル  
 ユーティリティの使用, 355  
 一時スペース  
 ユーティリティを使用した計算, 488  
 カップリングリスト, 490  
 ユーティリティの使用, 481, 485  
 リスト  
 ユーティリティを使用して作成, 489  
 ファイルコントロールブロック (FCB)  
 ダンプ/出力  
 ユーティリティの使用, 447

ファイル情報, 831  
 ファイルスペース割り当て, 828  
 ファイルの圧縮, 87  
 ファイルのアンロード, 1077  
 ファイルのリオーダー  
   オンライン処理の開始  
   ユーティリティの使用, 283  
 ファイルレポート, 1142  
 ファイルロック  
   事前, 290  
 フィールド  
   サブフィールド, 138  
   スーパーフィールド, 148  
   追加  
     ユーティリティの使用, 267  
   標準長の変更  
     ユーティリティの使用, 215  
 フィールド定義, 108, 125  
   ADACMP を使用する構文形式, 107  
 フィールド定義テーブル  
   データベースステータスレポート, 839  
 フィールド定義テーブル (FDT)  
   出力/ダンプ  
     ユーティリティの使用, 451  
   フィールドの追加  
     ユーティリティの使用, 267  
 フォネティックディスクリプタ  
   ADACMP COMPRESS を使用した定義, 135  
   定義  
     ADAINV の使用, 494  
 フォーマットバッファ  
   ロギングの開始  
     ユーティリティの使用, 298  
   ロギングの停止  
     ユーティリティの使用, 299  
 フォーマットプール  
   使用方法を表示するコマンド, 295  
 太字, 5  
 物理データベースレイアウト, 830  
 プロシージャ  
   VSE ライブラリおよびファイルの定義例, 1221  
 プロテクションログ  
   ADAFRM を使用したフォーマット, 415  
   ADAPLP を使用してシーケンシャル PLOG からレコード  
   を出力します。、692  
   ADAPLP を使用してシーケンシャル中間PLOGからレコー  
   ドを出力します。、692  
   ADAPLP を使用して複数の PLOG からレコードを出力し  
   ます。、692  
   ADAPRI を使用した複数 PLOG からのブロックの出力, 715  
   ADARES BACKOUT DPLOG を使用した 2 つのチェック  
   ポイント間の更新のデュアルバックアウト, 875  
   ADARES BACKOUT MPLOG を使用した 2 つのチェック  
   ポイント間の更新のマルチバックアウト, 875  
   ADARES BACKOUT を使用した 2 つのチェックポイント  
   間の更新のシーケンシャルバックアウト, 867  
   ADARES COPY を使用したシーケンシャルコピー  
   ユーティリティを使用したコピー, 887  
   ADARES PLCOPY を使用したシーケンシャルデータセッ  
   トへのデュアルコピー, 893  
   ADASAV RESTPLOG を使用したリストア, 1009  
   ADASEL を使用した情報選択, 1045  
   圧縮データの使用, 865  
   説明される情報, 865

デュアルのクローズ/切り替え  
   ユーティリティの使用, 297  
 データセットタイプ, 865  
 複数をマージする, 926

## ほ

ホールドキュー  
   ISN のカウント表示  
     ユーティリティの使用, 295  
   使用方法を表示するコマンド, 295  
 ホールドキューエレメント  
   表示  
     ユーティリティの使用, 294

## ま

マルチクライアントファイル  
   ADALOD LOAD を使用した新しいオーナー ID の割り当  
   て, 522  
   ADALOD LOAD を使用したオーナー ID 長の指定, 523  
   ADAULD を使用したアンロード, 1083  
   圧縮解除, 105  
   ロード, 537  
 マルチプルバリュフィールド  
   フィールドオプション (MU) , 116

## ゆ

ユニバーサルエンコーディングサポート (UES)  
   無変換フィールドオプション (NV) , 120  
 ユニークディスクリプタ  
   ADACMP での使用, 120  
   PE インスタンスの除外, 121  
   定義  
     ADALOD の使用, 529  
 ユニークディスクリプタプール  
   使用方法を表示するコマンド, 295  
 ユーザー  
   カウント表示  
     ユーティリティの使用, 295  
   指定 ID を持つユーザーの停止  
     ユーティリティの使用, 302  
   指定されたジョブ名を持つユーザーの停止  
     ユーティリティの使用, 302  
   指定ファイルを使用しているユーザーの停止  
     ユーティリティの使用, 301  
   全 ET ロジックユーザーの再同期, 303  
   タイムアウトしたユーザーの停止  
     ユーティリティの使用, 302  
   非アクティブティタイムリミット  
     ユーティリティの使用, 303  
   プライオリティ変更  
     ユーティリティの使用, 305  
 ユーザーキュー  
   使用方法を表示するコマンド, 295  
 ユーザーキューエレメント  
   5 つまで表示  
     オペレータコマンドの使用, 296  
   指定したユーザーを対象とした表示  
     ユーティリティの使用, 296  
   すべて表示  
     ユーティリティの使用, 296  
   中止されたユーザーキューエレメントの削除

- ユーティリティの使用, 301, 302
- 表示
  - ユーティリティの使用, 293
- ユーティリティの表示
  - ユーティリティの使用, 296
- ユーザーキューファイルリストプール
  - 使用方法を表示するコマンド, 295
- ユーザー出口
  - 6 (ADACMP ユーティリティで使用), 85
  - 9、ADAULD で使用, 1097
- ADACDU, 45
- ハイパーディスクリプタ, 131
- ユーザーデータ
  - ロギングの開始
    - ユーティリティの使用, 298
  - ロギングの停止
    - ユーティリティの使用, 299
- ユーティリティ
  - ファイルのシーケンシャルリスト, 1207
- ユーティリティコントロールステートメント
  - 規則, 7
  - 構文, 4
    - parameter list, 4
  - 構文記号, 5
  - パラメータ値, 8
    - 値, 8
    - 値範囲, 9
    - 値リスト, 9
    - デフォルト, 6
- ユーティリティ専用ステータス
  - スイッチオン/オフ
    - ユーティリティの使用, 304

## よ

- 読み取り専用ステータス
  - スイッチオン/オフ
    - ユーティリティの使用, 291, 301

## ら

- ラジオオブジェクト (LB)
  - フィールドオプション, 115

## り

- リカバリ
  - ADARAI RECOVER の使用, 755
  - BS2000 のジョブストリームスケルトン, 780
  - 失われた BS2000 パブセットメンバの, 791
  - 開始前のチェック, 770
  - ジョブストリームの構築, 762
  - ジョブストリームの再スタート, 771
  - スケルトンジョブストリーム, 767
  - 中断後の再スタート, 771
- リカバリログ
  - ADAFRM を使用したフォーマット, 415
  - ADAPRI を使用したブロックの出力, 715
  - ADARAI DISABLE を使用した非アクティブ化, 735
  - ADARAI LIST を使用した内容のチェック, 738
  - ADARAI LIST を使用した内容の表示, 737
  - ADARAI PREPARE を使用した初期化および開始, 751
  - ADARAI PREPARE を使用した準備, 752
  - ADARAI REMOVE を使用した削除, 773

- ADARAI を使用した準備, 730
- BS2000 の追加情報, 739
- 説明, 731
- リカバリの単位, 731
- リソース
  - 現使用方法の表示
    - ユーティリティの使用, 295
  - 統計
    - 表示用のコマンド, 295

## れ

- レコード
  - スバンドレコードのカウント, 347
  - スバンド (レコード) の有効化/無効化, 309
  - 追加/削除
    - ユーティリティの使用, 539
- レコードバッファ
  - 空値インジケータ値, 123
  - ロギングの開始
    - ユーティリティの使用, 298
  - ロギングの停止
    - ユーティリティの使用, 299
- レプリケーション
  - アクティブ化または非アクティブ化, 335
- レポート, 1127
  - ADAREPREPORT を使用して作成したデータベースステータス, 803
  - ADAWRK ユーティリティ, 1137
  - ADAWRK を使用して生成したワークエリアリカバリレポート, 1127
  - WORK パート 1 リカバリレポートの解説, 1137
  - サマリ, 1139
  - チェックポイントレコード, 1184
  - データベースステータスチェックポイント情報, 843
  - データベースステータスの説明, 819
  - データベースステータスファイル情報, 831
  - トランザクション, 1146
  - ファイル, 1142

## ろ

- ロング英数字 (LA)
  - フィールドオプション, 114

## わ

- ワイド文字フィールド
  - 無変換オプション (NV), 120
- ワークファイル
  - 定義
    - ユーティリティの使用, 397
- ワークプール
  - 使用方法を表示するコマンド, 295

