HITACHI

ノンストップデータベース

HiRDB Version 10 システム運用ガイド（Windows(R)用）

手引・操作書

3020-6-557-C0

前書き

■ 対象製品

●適用 OS：Windows Server 2019，Windows Server 2022，Windows Server 2025，Windows 10 Pro (x64)，Windows 10 Enterprise (x64)，Windows 11

P-2962-91A4 HiRDB Server Version 10 10-12

P-2962-7PA4 HiRDB Accelerator Version 10 10-00

P-2962-7HA4 HiRDB Non Recover Front End Server Version 10 10-00

P-2962-7JA4 HiRDB Advanced High Availability Version 10 10-00

●適用 OS：Windows Server 2019，Windows Server 2022，Windows Server 2025，Windows 10，Windows 11

P-2662-11A4 HiRDB/Run Time Version 10 10-12

P-2662-12A4 HiRDB/Developer's Kit Version 10 10-12

P-2662-32A4 HiRDB Developer's Suite Version 10 10-12

●適用 OS：Windows Server 2019，Windows Server 2022，Windows Server 2025，Windows 10 Home (x64)，Windows 10 Pro (x64)，Windows 10 Enterprise (x64)，Windows 11

P-2962-11A4 HiRDB/Run Time Version 10(64) 10-12

P-2962-12A4 HiRDB/Developer's Kit Version 10(64) 10-12

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

■ 輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法の規制ならびに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 発行

2026 年 9 月　3020-6-557-C0

■ 著作権

All Rights Reserved. Copyright (C) 2018, 2026, Hitachi, Ltd.

変更内容

変更内容(3020-6-557-C0)　HiRDB Version 10 10-12

追加・変更内容	変更個所
ディクショナリ搬出入ユティリティ（pdexp），及び pddefrev（定義系 SQL の生成）コマンドを使用して，次の定義を移行できるようにしました。 • プラグインが提供する抽象データ型を含む表定義（CREATE TABLE） • インデクス型を指定したインデクス定義（CREATE INDEX） • 手続きのパラメタのデータ型として抽象データ型を指定している手続き（CREATE PROCEDURE） • 手続き定義の中からほかの手続き，関数を呼び出している手続き（CREATE PROCEDURE） また，エクスポート・インポートによる表定義情報と表データの移行で，次の定義を移行できるようにしました。 • プラグインが提供する抽象データ型を含む表定義 • インデクス型を指定したインデクス定義	12.1.2(1)
クライアントマシンが複数の IP アドレスを持つ環境で，接続時の送信元 IP アドレスを明示的に指定する機能をサポートしました。 これによって，IP アドレスによる接続制限機能や，NAT やファイアウォールなど，送信元 IP アドレスを認識する機能を制御できます。	25.12.3

単なる誤字・脱字などはお断りなく訂正しました。

変更内容(3020-6-557-B0)　HiRDB Version 10 10-11

追加・変更内容
次の機能をサポートしたことによって，セキュリティの強化，ロール機能の利便性が向上しました。 • 監査対象イベントの定義で，オブジェクト種別（ROLE），セッション種別（ROLE）を指定できるようにしました。 • CURRENT_ROLE 値関数によって，現在使用しているロール名を検索できるようにしました。 • 埋込み型 UAP，ODBC から SET ROLE 文を発行できるようにしました。

変更内容(3020-6-557-A0)　HiRDB Version 10 10-10

追加・変更内容
表のアクセス権限を一括で管理できるロール機能をサポートしました。これによって，権限の管理や移行が容易になるだけでなく，アクセス制御の一貫性が向上し，セキュリティの強化にもつながります。
HiRDB の適用 OS に次の OS を追加しました。 • Windows Server 2025

変更内容(3020-6-557-90)　HiRDB Version 10 10-09

追加・変更内容
追加，変更個所はありません。

はじめに

このマニュアルは，プログラムプロダクト ノンストップデータベース　HiRDB Version 10 のシステム運用方法について説明したものです。なお，ここに記載されていない前提情報については，マニュアル「HiRDB Version 10 解説」を参照してください。

■ 対象読者

HiRDB Version 10（以降，**HiRDB** と表記します）を使ってリレーショナルデータベースシステムを構築または運用する方々を対象にしています。

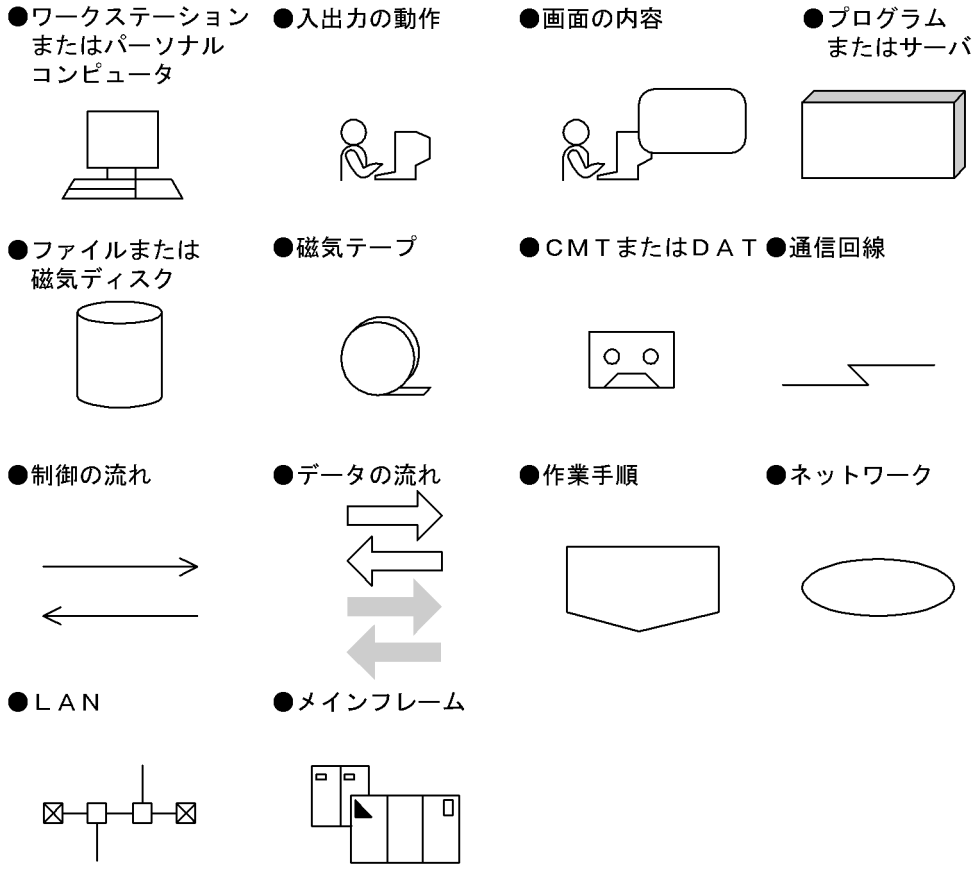
このマニュアルは次に示す知識があることを前提に説明しています。

- Windows のシステム管理の基礎的な知識
- SQL の基礎的な知識

また，このマニュアルは，マニュアル「HiRDB Version 10 解説」を前提としていますので，あらかじめお読みいただくことをお勧めします。

■ 図中で使用している記号

このマニュアルの図中で使用している記号を，次のように定義します。



■ このマニュアルで使用している記号

形式および説明で使用している記号を次に示します。ここで説明する文法記述記号は，説明のための記号なので実際には記述しないでください。

記号	意味
〔 〕	この記号で囲まれている項目は省略できます。 （例）pdbuffer 〔-p〕 　これは，pdbuffer と指定するか，または pdbuffer -p と指定することを示します。
…	この記号の直前の項目を繰り返して指定できます。 （例）pdbuffer -r RD エリア名 〔，RD エリア名〕… 　これは，-r オプションの RD エリア名を繰り返し指定できることを示します。
{{ }}	この記号で囲まれた複数の項目を一つの単位として，繰り返して指定できます。 （例）{{pdbuffer -a バッファ名}} 　これは，pdbuffer -a バッファ名 　　pdbuffer -a バッファ名 と指定できることを示します。
｜	この記号で区切られた項目は選択できます。 （例）pdlogadfg -d sys ｜ spd

記号	意味
	これは，-d オプションに sys か spd のどちらかを指定できることを示します。
{ }	この記号で囲まれている複数の項目のうちから，一つを選択できます。 （例）pdbuffer 〔{-r RD エリア名｜-i 認可識別子．インデクス識別子｜-o}〕 　これは，-r RD エリア名，-i 認可識別子．インデクス識別子，-o の三つのオプションのうち，どれか一つを指定することを示します。
__ （下線）	この記号で示す項目は，省略時の解釈値です。 （例）〔 pd_rpc_trace = Y ｜ N 〕 　これは，オペランドの指定を省略した場合，N が仮定されることを示します。
～	この記号のあとにユーザ指定値の属性を示します。
《 》	ユーザが指定しなかった場合に仮定される値を示します。
〈 〉	ユーザ指定値の構文要素を示します。
(( ))	ユーザ指定値の指定範囲を示します。

目次

前書き　2
変更内容　3
はじめに　5

1　HiRDB の開始及び終了方法　38
1.1　開始方法　39
1.1.1　開始モード　39
1.1.2　pdstart コマンドを実行するサーバマシン　40
1.1.3　自動開始　40
1.1.4　縮退起動（HiRDB/パラレルサーバ限定）　41
1.1.5　例題（HiRDB を正常開始する場合）　41
1.1.6　開始方法のカスタマイズ　43
1.1.7　開始処理完了の確認　44
1.2　終了方法　48
1.2.1　終了モード　48
1.2.2　pdstop コマンドを実行するサーバマシン　49
1.2.3　例題（HiRDB を正常終了する場合）　49
1.2.4　Windows のサービスで HiRDB を終了する方法　51
1.2.5　停止処理完了の確認　51
1.3　特別な開始方法　54
1.3.1　データベースを再度初期設定するときの開始方法（pdstart -i）　54
1.3.2　マスタディレクトリ用 RD エリアに障害が発生したときの開始方法（pdstart -r）　54
1.3.3　データディクショナリ用 RD エリアに障害が発生して，フロントエンドサーバが SUSPEND 状態のときの開始方法（pdstart -a）　56
1.4　ユニット単位の開始及び終了方法（HiRDB/パラレルサーバ限定）　57
1.4.1　ユニット単位の開始方法　57
1.4.2　ユニット単位の終了方法　57
1.4.3　ユニット単位で開始及び終了するときの運用　58
1.5　サーバ単位の開始及び終了方法（HiRDB/パラレルサーバ限定）　60
1.5.1　サーバ単位の開始方法　60
1.5.2　サーバ単位の終了方法　60
1.5.3　サーバ単位で開始及び終了するときの運用　60
1.6　開始時の注意事項　62
1.6.1　HiRDB 開始時の注意事項　62
1.6.2　HiRDB（ユニット）を強制開始するときの注意事項　63

1.6.3 HiRDB の開始処理に失敗したときの注意事項（HiRDB/パラレルサーバ限定） 64
1.7 終了時の注意事項 66
1.7.1 HiRDB 終了時の注意事項 66
1.7.2 計画停止，強制終了，又は異常終了したときの注意事項 67
1.8 HiRDB の開始処理時間を短縮する方法 73
1.8.1 開始処理時間を短縮する方法，及び運用例 73

2 機密保護の設定方法 75
2.1 機密保護とは 76
2.1.1 ユーザ権限の種類 76
2.1.2 改竄防止表との関係 80
2.1.3 監査証跡表との関係 81
2.2 ユーザ権限を設定するには 82
2.2.1 ユーザ権限を管理する人に DBA 権限を与えます 82
2.2.2 表を作成する人に CONNECT 権限，スキーマ定義権限，RD エリア利用権限を与えます 83
2.2.3 表（データベース）をアクセスする人に CONNECT 権限，アクセス権限を与えます 84
2.2.4 PDUSER オペランドに認可識別子とパスワードを指定します 86
2.2.5 他ユーザにスキーマ操作権限を与えます 86
2.3 ユーザ権限を取り消すには 87
2.3.1 DBA 権限を取り消すには 87
2.3.2 CONNECT 権限を取り消すには 87
2.3.3 スキーマ定義権限を取り消すには 87
2.3.4 RD エリア利用権限を取り消すには 88
2.3.5 アクセス権限を取り消すには 88
2.3.6 スキーマ操作権限を取り消すには 88
2.4 ディクショナリ表の参照権限を設定するには 90
2.5 OS ログインユーザの簡易認証機能とは 94
2.5.1 OS ログインユーザの簡易認証機能を使用するための準備 94
2.5.2 OS ログインユーザの簡易認証機能による HiRDB サーバへの接続方法 95
2.5.3 HiRDB のパスワード認証と OS ログインユーザの簡易認証の適用条件 96
2.5.4 簡易認証ユーザに付与できる権限 96
2.5.5 OS ログインユーザの簡易認証機能を使用する際の注意事項 96
2.6 データベースの機密保護での認可識別子 99
2.6.1 ユーザ識別子 99
2.6.2 ロール名 99
2.6.3 ユーザ識別子とロール名の指定できる個所 99
2.7 ロールを使用した権限の設定 101
2.7.1 ロールとは 101
2.7.2 ロールの運用手順 102

2.7.3 ロール使用時の留意事項 107

3 システムログファイルの運用 109

3.1 基本項目 110
3.1.1 システムログは障害の回復又はチューニング情報に使用されます 110
3.1.2 HiRDB はシステムログファイルを各種の状態に分けて管理しています 110
3.1.3 HiRDB 管理者がすることは 111
3.1.4 システムログファイルの状態の変わり方 111
3.1.5 システムログファイルの運用方法を選択しましょう 115
3.1.6 システムログファイルに障害が発生したとき 115
3.1.7 システムログファイルを操作するコマンド 115
3.2 システムログをアンロードする運用 117
3.2.1 HiRDB/シングルサーバの場合 117
3.2.2 HiRDB/パラレルサーバの場合 122
3.3 アンロードレスシステムログ運用 127
3.3.1 HiRDB/シングルサーバの場合 128
3.3.2 HiRDB/パラレルサーバの場合 132
3.4 アンロード状態のチェックを解除する運用 137
3.4.1 スワップ先にできる条件が変わります 137
3.4.2 利点 137
3.4.3 適用基準 137
3.4.4 注意事項 138
3.4.5 環境設定 138
3.4.6 運用方法 139
3.4.7 システムログファイルに障害が発生した場合 139
3.5 システムログファイルの操作方法 141
3.5.1 スワップ先にできる状態のファイルを確認する場合 141
3.5.2 スワップ先にできる状態のファイルがない場合 141
3.5.3 現用ファイルをアンロードする場合 142
3.5.4 アンロード済み状態のファイルをアンロードする場合 142
3.5.5 アンロード待ち状態のファイルに格納されているシステムログが必要ない場合 142
3.5.6 ファイルの状態を変更する場合 143
3.5.7 HiRDB の稼働中にシステムログファイルの容量を大きく（小さく）する場合 143
3.5.8 システムログファイルを新規追加する場合 144
3.5.9 システムログファイルを削除する場合 146
3.6 システムログファイルの状態遷移 147
3.7 システムログファイルのレコード長の変更方法 150
3.7.1 例題１（システムログをアンロードする運用をしている場合） 150
3.7.2 例題２（アンロードレスシステムログ運用をしている場合） 152

3.8 自動ログアンロード機能の運用方法 156
3.8.1 自動ログアンロード機能とは 156
3.8.2 環境設定 158
3.8.3 運用例 1（アンロードログファイル作成ディレクトリが一つの場合） 160
3.8.4 運用例 2（アンロードログファイル作成ディレクトリが二つの場合） 162
3.8.5 障害発生時の対策方法 163
3.8.6 HiRDB 終了時の注意事項 165
3.9 システムログファイルの空き容量監視機能の運用方法 166
3.9.1 システムログファイルの空き容量監視機能とは 166
3.9.2 環境設定 168
3.9.3 空き率が警告値未満になったときの HiRDB の処理 168
3.9.4 空き率が警告値未満になったときの HiRDB 管理者の処置 169
3.9.5 注意事項 171
3.9.6 システムログファイルの状態情報ファイルの出力（空き容量監視機能） 172
3.10 システムログファイルの残量警告機能の運用 175
3.10.1 システムログファイルの残量警告機能とは 175
3.10.2 環境設定 176
3.10.3 空きシステムログファイルの数がしきい値以下になった場合の HiRDB の処理 176
3.10.4 空きシステムログファイルの数がしきい値以下になった場合の HiRDB 管理者の処置 177
3.11 システムログファイルの自動拡張機能の運用方法 178
3.11.1 システムログファイルの自動拡張機能とは 178
3.11.2 環境設定 180
3.11.3 自動拡張時の HiRDB の処理 181
3.11.4 自動拡張できなくなった場合の HiRDB 管理者の処置 182
3.11.5 ほかの機能との関連 184
3.11.6 注意事項 185
3.11.7 システムログファイルの状態情報ファイルの出力（自動拡張機能） 185
3.12 データベース回復時に必要なアンロードログファイルの特定方法 186
3.12.1 最初に指定するアンロードログファイルの情報取得（バックアップ取得時） 187
3.12.2 最後に指定するアンロードログファイルの情報取得（データベース回復時） 191
3.12.3 アンロードログファイルの並び替え（時系列リストの作成） 192

4 シンクポイントダンプファイルの運用 200
4.1 基本項目 201
4.1.1 シンクポイントとは 201
4.1.2 シンクポイントダンプファイルの状態 201
4.1.3 HiRDB 開始時のシンクポイントダンプファイルの状態 201
4.1.4 シンクポイントダンプファイルの状態が変わるときは？ 202
4.1.5 有効保証世代数とファイルの状態との関係 202

4.1.6 シンクポイントダンプファイルを操作するコマンド 204
4.2 シンクポイントダンプの取得間隔の設定方法 206
4.2.1 取得間隔は pd_log_sdinterval オペランドで設定します 206
4.2.2 取得間隔を決める目安 206
4.2.3 シンクポイントダンプの取得間隔の設定例 206
4.3 シンクポイントダンプファイルの操作方法 208
4.3.1 シンクポイントダンプファイルの状態が変わった場合 208
4.3.2 上書きできる状態のファイルがない場合 208
4.3.3 HiRDB の稼働中にシンクポイントダンプファイルの容量を大きく（小さく）する場合 209
4.3.4 ファイルの状態を変更する場合 210
4.3.5 シンクポイントダンプファイルを新規追加する場合 210
4.3.6 シンクポイントダンプファイルを削除する場合 212
4.3.7 ファイル中のシンクポイントダンプに対応するシステムログファイルを知りたい場合 212
4.3.8 シンクポイントダンプファイルの有効保証世代数を増やす場合 212
4.4 シンクポイントダンプファイルの状態遷移 215

5 ステータスファイルの運用 217
5.1 基本項目 218
5.1.1 ステータスファイルの状態 218
5.1.2 HiRDB 開始時のステータスファイルの状態 218
5.1.3 ステータスファイルの状態が変わるときは？ 219
5.1.4 ステータスファイルのスワップとは 219
5.1.5 ステータスファイルを操作するコマンド 219
5.2 ステータスファイルの操作方法 221
5.2.1 ステータスファイルがスワップした場合 221
5.2.2 予備ファイルがない場合 222
5.2.3 HiRDB の稼働中にステータスファイルの容量を大きく（小さく）する場合 223
5.2.4 ファイルの状態を変える場合 224
5.2.5 現用ファイルを変更する場合 224
5.2.6 ステータスファイルを新規追加する場合 224
5.2.7 ステータスファイルを削除する場合 225
5.2.8 ステータスファイルの情報を調べる場合 226
5.3 ステータスファイルの状態遷移 227

6 バックアップの取得方法 229
6.1 バックアップについて 230
6.1.1 基本項目 230
6.1.2 オプション項目 232
6.2 バックアップ取得モード 234

6.2.1 バックアップ取得モードの種類 234
6.2.2 データベースの更新ログ取得方式との関係 235
6.3 同時にバックアップを取得する必要がある RD エリア 236
6.4 バックアップの取得例 244
6.4.1 例題１（HiRDB Control Manager のウィザードでシステム単位にバックアップを取得する場合） 244
6.4.2 例題２（HiRDB Control Manager のウィザードで RD エリア単位にバックアップを取得する場合） 249
6.4.3 例題３（システム単位にバックアップを取得する場合） 254
6.4.4 例題４（システム単位にバックアップを取得する場合） 255
6.4.5 例題５（システム単位にバックアップを取得する場合） 256
6.4.6 例題６（ユニット単位にバックアップを取得する場合） 257
6.4.7 例題７（サーバ単位にバックアップを取得する場合） 258
6.4.8 例題８（RD エリア単位にバックアップを取得する場合） 258
6.5 差分バックアップの取得 261
6.5.1 差分バックアップ機能とは 261
6.5.2 差分バックアップ機能を使用するための準備 263
6.5.3 差分バックアップ機能の運用例 265
6.5.4 累積差分バックアップの作成 268
6.5.5 差分バックアップの履歴情報ファイルの参照 269
6.5.6 差分バックアップ管理ファイルの回復 271
6.6 バックアップ閉塞を使用してバックアップを取得する場合（pdcopy コマンド以外でバックアップを取得する場合） 272
6.6.1 バックアップ閉塞とは 272
6.6.2 例題（ほかの製品のバックアップ機能を使用する場合） 277
6.7 更新凍結コマンド（pddbfrz コマンド）を使用してバックアップを取得する場合 279
6.7.1 更新凍結コマンドの対象となる運用 279
6.7.2 更新凍結コマンド（pddbfrz コマンド）とは 279
6.7.3 運用例 282
6.7.4 満杯データページの調査方法 285
6.7.5 更新凍結コマンドを実行したユーザ LOB 用 RD エリアの操作 286
6.7.6 RD エリアの自動増分との関係 287
6.7.7 注意事項 288

7 データベースの更新ログを取得しないときの運用 289

7.1 データベースの更新ログ取得方式 290
7.1.1 データベースの更新ログ取得方式の概要 290
7.2 更新前ログ取得モードで UAP 又はユティリティを実行するときの運用 293
7.2.1 更新前ログ取得モードで UAP 又はユティリティを実行するときの運用方法 293
7.3 ログレスモードで UAP 又はユティリティを実行するときの運用 295
7.3.1 ログレスモードで UAP 又はユティリティを実行するときの運用方法 295

8 システムの稼働環境を知る方法（システムの状態監視） 298
8.1 メッセージログでシステムの実行状況を確認する 299
8.1.1 メッセージログを参照する（メッセージログの出力先） 299
8.1.2 メッセージログファイルの運用 299
8.1.3 メッセージログの出力方式の選択（HiRDB/パラレルサーバ限定） 300
8.1.4 イベントログへのメッセージ出力抑止 303
8.1.5 イベントログに出力するメッセージの重要度の変更 307
8.1.6 ユティリティ実行時のインフォメーションメッセージ出力抑止 308
8.2 UAP 又はユティリティの実行時間が長い場合 310
8.3 HiRDB の開始又は終了処理が長い場合 313
8.4 RD エリアの状態を知りたい場合 314
8.5 共用メモリの使用状況を知りたい場合 315
8.5.1 共用メモリの使用状況の調査方法 315
8.6 デッドロックが発生した場合 318
8.6.1 基本項目 318
8.6.2 デッドロック情報の出力内容 321
8.6.3 タイムアウト情報の出力内容 325
8.6.4 資源種別と資源情報 330
8.6.5 資源情報の見方 336
8.6.6 デッドロック SQL 情報の出力内容 338
8.6.7 デッドロック情報とデッドロック SQL 情報を突き合わせる方法 338
8.7 排他資源管理テーブル不足が発生した場合 341
8.7.1 排他資源管理テーブル不足が発生した場合の対処方法 341
8.8 UAP の状態監視（シンクポイントダンプ有効化のスキップ回数監視機能） 346
8.8.1 シンクポイントダンプ有効化のスキップ回数監視機能の概要 346
8.9 SQL 実行時間についての警告情報の出力（SQL 実行時間警告出力機能） 352
8.9.1 SQL 実行時間警告出力機能とは 352
8.9.2 SQL 実行時間警告出力機能の使用方法 355
8.9.3 SQL 実行時間警告情報ファイルの出力 358
8.9.4 KFPA20009-W メッセージの出力 364
8.9.5 注意事項 364
8.10 UAP 又はユティリティの実行時間の監視（無応答障害時の影響を抑える方法） 366
8.11 リソースの使用率の監視 367
8.12 サーバプロセスの状態監視（メッセージキュー監視機能） 368
8.12.1 メッセージキュー監視機能の概要 368
8.13 サーバプロセスの異常終了回数の監視（プロセスの異常終了回数監視機能） 371
8.13.1 プロセスの異常終了回数監視機能とは 371
8.13.2 プロセスの異常終了回数監視機能の適用範囲 371
8.13.3 プロセスの異常終了回数監視機能の指定方法 372

8.13.4 備考 373
8.14 ロールバック処理が進行中であることを確認する方法 374
8.14.1 pdls -d trn -a コマンドの実行結果 374
8.14.2 KFPS02235-I メッセージ 374
8.14.3 ロールバック完了の目安 375
8.14.4 注意事項 375

9 システムの稼働環境を変更する方法 377
9.1 HiRDB システム定義を変更する方法 378
9.1.1 変更手順 378
9.1.2 注意事項 379
9.1.3 HiRDB/パラレルサーバ固有の注意事項 379
9.1.4 HiRDB Datareplicator と連携している場合 379
9.2 HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド） 381
9.2.1 変更手順 381
9.2.2 オペランドの指定値を変更するときの注意事項 382
9.2.3 システム構成変更コマンド実行時の注意事項 383
9.2.4 システム構成変更コマンド実行後の HiRDB の状態 384
9.2.5 ほかの機能との関連 385
9.2.6 障害発生時の対処方法 387
9.3 HiRDB の稼働中にグローバルバッファを追加，変更，又は削除する方法（グローバルバッファの動的変更） 390
9.3.1 グローバルバッファの動的変更とは 390
9.3.2 運用例 392
9.4 サーバプロセス数を変更する方法 396
9.4.1 サーバプロセス数の変更方法 396
9.5 ユーザ数を増やすときにすること 400
9.5.1 ユーザ数が増えた場合に見直す項目 400
9.6 HiRDB に接続できないクライアントが発生するときの対処方法（クライアントグループの接続枠保証機能） 402
9.6.1 クライアントグループの接続枠保証機能の概要 402
9.7 通信処理で使用するポート番号を範囲指定する方法 407
9.7.1 ポート番号を範囲指定する方法及び注意事項 407
9.8 ホスト名を変更する方法 409
9.8.1 HiRDB の稼働中にホスト名を変更する手順 409
9.9 コマンドのデッドロックプライオリティ値を変更する方法 411
9.9.1 コマンドのデッドロックプライオリティ値 411
9.9.2 環境設定 412
9.9.3 運用方法 412
9.10 排他制御用プールを分割する方法 414

9.10.1 排他制御用プールの分割の概要 414
9.11 トランザクション回復メッセージキューサイズを変更する方法 419
9.11.1 トランザクション回復メッセージキューサイズの適用基準，及び変更方法 419
9.12 OS の時刻を変更する方法 420
9.12.1 時刻を遅らせる方法 420
9.12.2 方法 1 で時刻を遅らせる場合 421
9.12.3 方法 2 で時刻を遅らせる場合 424
9.12.4 方法 2 で時刻を遅らせた後で障害が発生した場合のデータベースの回復 430
9.12.5 方法 2 で時刻を遅らせる場合の制限事項 432
9.12.6 方法 2 で時刻を遅らせる場合の注意事項 433
9.13 デッドロック監視処理時間を変更する方法 434
9.13.1 デッドロック監視処理時間の変更方法 434

10 HiRDB ファイルシステム領域の運用 437
10.1 HiRDB ファイルシステム領域に関する情報を知る方法 438
10.2 HiRDB ファイルシステム領域を作成（初期設定）する方法 439
10.2.1 作成手順 439
10.3 HiRDB ファイルシステム領域のバックアップを取得する方法 441
10.4 HiRDB ファイルシステム領域を回復する方法 442
10.5 HiRDB ファイルを削除する方法 443
10.5.1 HiRDB ファイルの削除方法 443
10.6 ファイルアクセスエラー検知時ユニットダウン機能 445
10.6.1 ファイルアクセスエラー検知時ユニットダウン機能とは 445
10.6.2 注意事項 445
10.6.3 準備作業 445
10.6.4 ファイルアクセスエラーが発生した場合の対処方法 446
10.7 HDP 機能を使用した HiRDB ファイルシステム領域の仮想化 447
10.7.1 HDP 機能とは 447
10.7.2 HiRDB ファイルシステム領域を効率良く仮想化するためには 448
10.7.3 ストレージを効率良く使用するための運用 448

11 システム構成を変更する方法 456
11.1 ユニットの追加 457
11.1.1 HiRDB の稼働中にユニットを追加する場合 457
11.1.2 HiRDB を終了してユニットを追加する場合 459
11.2 ユニットの削除 462
11.2.1 HiRDB の稼働中にユニットを削除する場合 462
11.2.2 HiRDB を終了してユニットを削除する場合 464
11.3 ユニットの移動 466

11.3.1 HiRDB を終了してユニットを移動する場合（RD エリアを再作成する場合） 466
11.3.2 HiRDB を終了してユニットを移動する場合（RD エリアを再作成しない場合） 469
11.4 サーバの追加 473
11.4.1 HiRDB の稼働中にサーバを追加する場合 473
11.4.2 HiRDB を終了してサーバを追加する場合 475
11.5 サーバの削除 478
11.5.1 HiRDB の稼働中にサーバを削除する場合 478
11.5.2 HiRDB を終了してサーバを削除する場合 480
11.6 サーバの移動 483
11.6.1 HiRDB を終了してサーバを移動する場合(RD エリアを再作成する場合) 483
11.6.2 HiRDB を終了してサーバを移動する場合(RD エリアを再作成しない場合) 486
11.7 HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行方法 490
11.7.1 移行作業をする前に 490
11.7.2 移行手順 492
11.7.3 複数のユーザ用 RD エリアを異なるバックエンドサーバに移行する場合の注意事項 496
11.8 負荷均等化を目的としたバックエンドサーバの移動 498
11.8.1 シナリオによるバックエンドサーバの負荷均等化 498
11.8.2 前提条件及び対象とする業務の条件 500
11.8.3 シナリオ運用方法 500
11.8.4 バックエンドサーバの構成例 504
11.8.5 HiRDB に関する準備 507
11.8.6 ユーザが行うバックエンドサーバの負荷均等化 508

12 システム間で資源を移行する方法 511
12.1 表をほかの HiRDB システムに移行する方法 512
12.1.1 表をほかの HiRDB システムに移行する前に 513
12.1.2 表定義情報と表データを同時に移行する方法 518
12.1.3 表定義情報と表データを別々に移行する方法 519
12.1.4 移行先で表・インデクス定義を行いデータだけ移行する方法 524
12.1.5 例題１（表定義情報と表データを同時に移行する場合） 525
12.1.6 例題２（表単位に定義情報とデータを別々に移行する場合） 529
12.1.7 例題３（スキーマ単位に表を移行する場合） 535
12.1.8 別スキーマに表を移行する場合の制御文ファイルの例 541
12.1.9 例題４（エンディアンが異なるシステムに，バイナリデータを含む表を移行する場合） 542
12.1.10 例題５（固定長文字データの列の定義長を拡張して，表を移行する場合） 547
12.1.11 インポート時に障害が発生した場合の対処方法 553
12.2 ストアドプロシジャをほかの HiRDB システムに移行する方法 555
12.2.1 ストアドプロシジャをほかの HiRDB システムに移行する前に 555
12.2.2 例題 557

13 表の運用 564
13.1 表の格納効率を調べる方法 565
13.1.1 定期的にデータベース状態解析ユティリティを実行します 565
13.1.2 データの格納効率が悪くなるとメッセージが出力されます 569
13.1.3 検索性能などに期待した効果が得られない場合 571
13.2 表の再編成 572
13.2.1 表の再編成とは 572
13.2.2 表の再編成の実行単位 572
13.2.3 データベースの更新ログ取得方式の選択 575
13.2.4 表を再編成する前に 577
13.3 表の再編成（例題） 580
13.3.1 例題１（表単位で再編成する場合：HiRDB/シングルサーバの場合） 580
13.3.2 例題２（表単位で再編成する場合：HiRDB/パラレルサーバの場合） 582
13.3.3 例題３（RD エリア単位で再編成する場合） 584
13.3.4 例題４（スキーマ単位で再編成する場合） 587
13.3.5 例題５（LOB 列が定義されている表を再編成する場合） 590
13.3.6 例題６（ディクショナリ表を再編成する場合） 593
13.3.7 例題７（ログレスモードで再編成する場合） 595
13.3.8 例題８（抽象データ型を定義した表を再編成する場合） 598
13.4 表の再編成時期を予測する方法（再編成時期予測機能） 602
13.4.1 再編成時期を予測するには 602
13.4.2 再編成時期予測機能を使用するための準備 604
13.4.3 運用の流れ 605
13.4.4 再編成時期予測機能使用時の注意事項 610
13.4.5 再編成時期の予測をやめる場合 612
13.4.6 再編成時期予測のカスタマイズ 612
13.5 表のデータを削除する方法 614
13.5.1 表のデータを削除する方法の説明 614
13.6 列を追加する方法 615
13.6.1 列を追加する前に 615
13.6.2 例題１（非 FIX 表に列を追加する場合） 616
13.6.3 例題２（データが格納されている非 FIX 表に列を追加し，追加した列に値を格納する場合） 616
13.6.4 例題３（LOB 列を追加する場合） 617
13.6.5 例題４（抽象データ型の列を追加する場合） 617
13.6.6 例題５（FIX 表に列を追加する場合：予備列を定義している場合） 618
13.6.7 例題６（FIX 表に列を追加する場合：DAT 形式でアンロードする場合） 618
13.6.8 例題７（FIX 表に列を追加する場合：バイナリ形式でアンロードする場合） 620
13.7 列を削除する方法 623
13.7.1 例題（列を削除する場合） 623

13.8 表の定義を変更する方法 626
13.8.1 例題（列のデータ長を変更する場合） 626
13.9 表名又は列名を変更する方法 627
13.9.1 例題１（表名を変更する場合） 627
13.9.2 例題２（列名を変更する場合） 627
13.10 表の横分割数を増やす方法 628
13.10.1 例題１（キーレンジ分割表の横分割数を増やす場合） 628
13.10.2 例題２（フレキシブルハッシュ分割表の横分割数を増やす場合） 630
13.10.3 例題３（FIX ハッシュ分割表の横分割数を増やす場合） 631
13.11 表の横分割数を増やす方法（ハッシュ分割表のリバランス機能を使用している場合） 633
13.11.1 ハッシュ分割表のリバランス機能とは 633
13.11.2 ハッシュ分割表のリバランス機能を使用するための準備 635
13.11.3 例題（リバランス表の横分割数を増やす場合） 636
13.11.4 リバランスユティリティの使い方（表のリバランス処理に時間が掛かる場合） 639
13.11.5 FIX ハッシュ分割表に対する注意事項 641
13.12 表の分割格納条件の変更 642
13.12.1 分割格納条件の変更の目的 642
13.12.2 分割格納条件の変更時に使用する機能 646
13.12.3 前提条件 648
13.12.4 分割格納条件の変更方法（境界値指定の場合） 653
13.12.5 RD エリアの分割（境界値指定の場合） 654
13.12.6 RD エリアの統合（境界値指定の場合） 667
13.12.7 分割格納条件の変更方法（格納条件指定の場合） 675
13.12.8 RD エリアの分割（格納条件指定の場合） 676
13.12.9 RD エリアの統合（格納条件指定の場合） 690
13.12.10 分割格納条件の変更方法（マトリクス分割の場合） 704
13.12.11 RD エリアの分割（マトリクス分割の場合） 706
13.12.12 RD エリアの統合（マトリクス分割の場合） 712
13.12.13 ほかの機能との関連 717
13.13 表の分割格納条件を変更するときの運用 718
13.13.1 例題（境界値指定の場合） 718
13.13.2 例題（格納条件指定の場合） 725
13.13.3 例題（マトリクス分割の場合） 731
13.13.4 データ再登録時の運用 740
13.13.5 RD エリアを再利用する運用 740
13.13.6 データベース再編成ユティリティ及びデータベース作成ユティリティの例 743
13.13.7 非分割キーインデクスがある表の分割・統合 745
13.13.8 インデクスが不完全な状態での分割・統合 745
13.13.9 分割・統合後のデータ件数の確認 745

13.13.10 障害時の運用 747
13.13.11 参照制約及び検査制約を使用している場合の運用 749
13.14 ハッシュ関数を変更する方法 751
13.14.1 例題１（フレキシブルハッシュ分割の場合） 751
13.14.2 例題２（FIX ハッシュ分割の場合） 751
13.15 表の分割定義を変更する方法 754
13.15.1 例題１ キーレンジ分割からハッシュ分割に変更，及び分割キー列を変更する場合 754
13.15.2 例題２ ハッシュ分割からキーレンジ分割に変更する場合 756
13.15.3 例題３ RD エリアを年月単位に循環させて割り当てる場合（マトリクス分割ではない場合） 757
13.15.4 例題４ RD エリアを年月単位に循環させて割り当てる場合（マトリクス分割の場合） 761
13.15.5 例題５ RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割でない場合） 766
13.15.6 例題６ RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割の場合） 770
13.16 別表へのデータの移行方法 777
13.16.1 例題１（別表へデータを移行する場合） 777
13.16.2 例題２（XML 型の列を含むデータを，表単位で移行する場合） 780
13.16.3 列構成情報ファイルの指定例 782
13.17 表を削除する方法 784
13.18 スキーマを削除する方法 785
13.18.1 スキーマを削除すると削除されるリソース 785
13.18.2 他人のリソースにも影響があります 785
13.18.3 スキーマが削除できない場合 786
13.19 抽象データ型を削除する方法 787
13.19.1 抽象データ型を削除する方法と注意事項 787
13.20 既存の表の定義系 SQL を作成する方法 788
13.21 リストの管理（絞込み検索） 789
13.21.1 リストの管理方法 789
13.22 表データ中の空白文字を統一する方法 793
13.22.1 空白変換機能とは 793
13.22.2 空白変換レベルの設定方法 796
13.22.3 表中の空白文字の統一方法 796
13.23 DECIMAL 型の符号部を変換する方法 800
13.23.1 DECIMAL 型の符号正規化機能とは 800
13.23.2 既存のデータを正規化する方法 802

14 インデクスの運用 804

14.1 インデクスの格納効率を上げる方法（インデクスの再編成） 805
14.1.1 インデクスの再編成とは 805
14.1.2 例題１（インデクスの再編成方法） 806
14.1.3 インデクスの再編成中にエラーが発生した場合の処置 807

14.1.4 例題 2（インデクスの再編成中に RD エリアの容量不足が発生した場合：ログレスモード以外で実行した場合） 808
14.1.5 例題 3（インデクスの再編成中に RD エリアの容量不足が発生した場合：ログレスモードで実行した場合） 809
14.2 データを格納している表にインデクスを定義する方法 812
14.2.1 追加方法 812
14.2.2 インデクスの作成時間を短縮する方法（EMPTY オプション） 812
14.3 インデクス名を変更する方法 814
14.4 インデクスを削除する方法 815
14.5 既存のインデクスの定義系 SQL を作成する方法 817
14.6 インデクスページスプリットの発生回数を削減する方法（アンバランスインデクススプリット） 818
14.6.1 インデクスページスプリットとは 818
14.6.2 アンバランスインデクススプリットとは 819
14.7 インデクスの一括作成処理中にエラーが発生したときの対処方法 822
14.7.1 リロード（データロード）をログ取得モード又は更新前ログ取得モードで実行した場合の回復例 823
14.7.2 リロード（データロード）をログレスモードで実行した場合の回復例（インデクス格納 RD エリアにほかの表又はインデクスがない場合） 824
14.7.3 リロード（データロード）をログレスモードで実行した場合の回復例（インデクス格納 RD エリアにほかの表又はインデクスがある場合） 826
14.7.4 インデクス格納 RD エリアがあるディスクに障害が発生したときの回復例 827
14.8 プラグインインデクスの遅延一括作成 830
14.8.1 プラグインインデクスの遅延一括作成とは 830
14.8.2 環境設定方法 832
14.8.3 UAP 実行時の手順 835
14.8.4 注意事項 836
14.8.5 障害発生時の対処方法 838

15 RD エリアの運用 839

15.1 RD エリアの容量が不足してくると 840
15.1.1 RD エリアの容量が不足したときの対処 840
15.2 RD エリアを作成する方法（RD エリアの追加） 842
15.2.1 RD エリアを追加する前に 842
15.2.2 例題 844
15.3 RD エリアの容量を大きくする方法（RD エリアの拡張） 848
15.3.1 RD エリアを拡張する前に 848
15.3.2 例題 849
15.4 RD エリアの容量又は属性を変更する方法（RD エリアの再初期化） 851
15.4.1 RD エリアを再初期化する前に 851
15.4.2 例題 1（インデクスが定義されている場合） 853
15.4.3 例題 2（インデクスが定義されている場合） 857

15.4.4 例題３（LOB 列が定義されている場合） 861
15.4.5 例題４（LOB 列が定義されている場合） 866
15.4.6 例題５（抽象データ型が定義されている場合） 870
15.4.7 例題６（抽象データ型が定義されている場合） 874
15.4.8 例題７（表に関連する全 RD エリアを再初期化してデータの回復をユティリティで行う場合） 879
15.4.9 例題８（表に関連する全 RD エリアを再初期化してデータの回復を UAP で行う場合） 884
15.4.10 例題９（RD エリアのディスク配置を変更する場合） 889
15.5 RD エリア名を変更する方法（RD エリアの属性変更） 895
15.5.1 RD エリア名を変更する前に 895
15.5.2 例題 897
15.5.3 RD エリア名変更時にバックアップを取得していなかった場合の回復方法 899
15.6 RD エリアのオープン契機を変更する方法（RD エリアの属性変更） 901
15.6.1 RD エリアのオープン契機を変更する前に 901
15.6.2 例題 904
15.7 RD エリアを削除する方法 907
15.7.1 RD エリアを削除する前に 907
15.7.2 例題 908
15.8 RD エリアの自動増分 910
15.8.1 RD エリアの自動増分とは 910
15.8.2 RD エリアの自動増分を適用する場合の留意事項 918
15.8.3 HiRDB ファイルシステム領域を自動的に拡張する方式 921
15.8.4 HiRDB ファイルシステム領域内で HiRDB ファイルを拡張する方式 922
15.8.5 例題（HiRDB ファイルシステム領域を自動的に拡張する方式） 925
15.8.6 例題（HiRDB ファイルシステム領域内で HiRDB ファイルを拡張する方式） 928
15.9 RD エリアを移動する方法（RD エリアの移動） 931
15.9.1 RD エリアを移動する前に 931
15.9.2 例題 1（新規サーバマシンのバックエンドサーバに移動する場合） 935
15.9.3 例題 2（別ユニットにあるバックエンドサーバに移動する場合） 938
15.9.4 例題 3（同じユニットにあるバックエンドサーバに移動する場合） 939
15.9.5 例題 4（横分割表を格納した RD エリアを移動する場合） 941
15.9.6 例題 5（抽象データ型を格納した RD エリアを移動する場合） 946
15.9.7 例題 6（オンライン中の RD エリア移動機能で共有ディスク上に HiRDB ファイルがある RD エリアを移動する場合） 950
15.10 RD エリアを格納している HiRDB ファイルシステム領域を移動する方法 953
15.10.1 シンボリックリンクを使用した移動手順 954
15.10.2 ファイルパス変更コマンドを使用した移動手順 955
15.10.3 シンボリックリンクを使用した移動の例題 956
15.10.4 ファイルパス変更コマンドを使用した移動の例題 958
15.11 使用中空きページ及び使用中空きセグメントを再利用する方法 963

15.11.1 ページ及びセグメントの状態 963
15.11.2 使用中空きページの再利用 964
15.11.3 使用中空きセグメントの再利用 969
15.12 物理エラー検知時ユニットダウン機能 972
15.12.1 物理エラー検知時ユニットダウン機能とは 972
15.12.2 注意事項 972
15.12.3 準備作業 973
15.12.4 物理エラーが発生した場合の対処方法 973
15.13 RD エリアアクセス時のページ不正検知時の動作 975

16 ストアドプロシジャ及びストアドファンクションの運用 976
16.1 ストアドプロシジャ及びストアドファンクションを作成（登録）する前に 977
16.2 ストアドプロシジャ及びストアドファンクションを作成（登録）する方法 978
16.2.1 ストアドプロシジャ及びストアドファンクションの作成（登録）方法 978
16.3 ストアドプロシジャ及びストアドファンクションが無効になった場合 980
16.4 ストアドプロシジャ及びストアドファンクションを削除する方法 981
16.4.1 ストアドプロシジャ及びストアドファンクションの削除方法 981
16.5 既存のストアドプロシジャの定義系 SQL を作成する方法 982

17 Java ストアドプロシジャ及び Java ストアドファンクション使用時の運用983
17.1 Java ストアドプロシジャ及び Java ストアドファンクションの概要 984
17.1.1 Java ストアドプロシジャ，Java ストアドファンクションを使用できる環境 984
17.1.2 Java ストアドプロシジャ及び Java ストアドファンクションとは 984
17.1.3 Java ストアドプロシジャ及び Java ストアドファンクションの動作 984
17.2 Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成 986
17.2.1 Java 仮想マシンの位置づけ 986
17.2.2 JRE（Java 実行環境） 987
17.2.3 HiRDB/シングルサーバの場合のシステム構成 988
17.2.4 HiRDB/パラレルサーバの場合のシステム構成 988
17.3 環境設定 990
17.3.1 環境設定手順 990
17.4 JAR ファイルの操作 992
17.4.1 JAR ファイルに障害が発生した場合 992
17.4.2 サーバ構成を変更した場合（HiRDB/パラレルサーバ限定） 992

18 C ストアドプロシジャ及び C ストアドファンクション使用時の運用 993
18.1 C ストアドプロシジャ又は C ストアドファンクション使用時の環境設定 994
18.1.1 C ストアドプロシジャ又は C ストアドファンクションの実行環境を作成するまでの流れ 994
18.1.2 C ライブラリファイルの新規登録 994
18.1.3 C ライブラリファイルの再登録 995

18.1.4 C ライブラリファイルの削除 995
18.2 C ライブラリファイルに障害が発生した場合の対処 997

19 最適化情報の運用 998
19.1 最適化情報とは 999
19.1.1 最適化情報の概要 999
19.1.2 最適化情報の登録方法 999
19.1.3 最適化情報の内容 1000
19.1.4 最適化情報収集レベル 1000
19.2 最適化情報の運用方法 1001
19.2.1 最適化情報の収集，登録 1001
19.2.2 最適化情報の更新 1002
19.2.3 最適化情報の回復 1003
19.2.4 最適化情報の削除 1004
19.3 最適化情報パラメタファイルの運用方法 1005
19.3.1 最適化情報パラメタファイルとは 1005
19.3.2 最適化情報パラメタファイルによる最適化情報の登録 1006
19.3.3 最適化情報パラメタファイルへの最適化情報の退避 1006
19.4 最適化情報を運用するときの注意事項 1008
19.4.1 表及びインデクスの定義変更をした場合の最適化情報パラメタファイルの使用可否 1008
19.4.2 最適化情報退避後の pdgetcst の操作と最適化情報の回復可否 1009
19.4.3 最適化情報の登録，更新，又は削除時にエラーが発生した場合 1010

20 障害が発生したときの対処方法 1011
20.1 障害が発生したときの HiRDB の処理と HiRDB 管理者の処置 1012
20.1.1 障害が発生したときに HiRDB 管理者がすること 1012
20.1.2 障害が発生したときに HiRDB が取得する情報 1015
20.1.3 障害が発生したときの HiRDB の処理 1016
20.1.4 サーバプロセスで障害が発生したときの対処方法 1017
20.1.5 HiRDB が再開始するときに引き継ぐ情報 1018
20.1.6 トランザクションキャンセル時のプロセスダウンメッセージ変更機能 1019
20.1.7 グローバルバッファ制御情報トレース取得機能 1023
20.2 UAP が正しく実行されないときの対処方法 1026
20.2.1 UAP を実行できないときの対処方法 1026
20.2.2 UAP が終了しないときの対処方法 1026
20.2.3 UAP が異常終了したときの対処方法 1027
20.3 運用コマンドが正しく実行されないときの対処方法 1028
20.3.1 運用コマンドを実行できないときの対処方法 1028
20.3.2 運用コマンドが応答待ちタイムアウトになったときの対処方法 1028

20.4 HiRDB が開始できないときの対処方法 1029
20.4.1 HiRDB が正常開始できないときの対処方法 1029
20.4.2 HiRDB が再開始できないときの対処方法 1030
20.4.3 マスタディレクトリ用 RD エリアに障害が発生したときの対処方法 1030
20.5 HiRDB が終了できないときの対処方法 1032
20.6 システムログファイルに障害が発生したときの対処方法 1033
20.6.1 現用ファイルに障害が発生したときの対処方法 1033
20.6.2 HiRDB Datareplicator を使用している場合の対処方法 1036
20.6.3 現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法 1036
20.7 シンクポイントダンプファイルに障害が発生したときの対処方法 1037
20.7.1 障害発生時の対処方法 1037
20.8 ステータスファイルに障害が発生したときの対処方法 1040
20.8.1 現用ファイルに障害が発生したときの対処方法 1040
20.8.2 障害が発生したステータスファイルがある状態で HiRDB（ユニット）を開始するときの手順 1043
20.8.3 現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法 1048
20.9 ファイル障害が発生したとき（システムファイル以外のファイル）の対処方法 1051
20.9.1 HiRDB システム定義ファイルに障害が発生したときの対処方法 1051
20.9.2 メッセージログファイルに障害が発生したときの対処方法 1051
20.9.3 統計ログファイルに障害が発生したときの対処方法 1051
20.9.4 データ連動用連絡ファイルに障害が発生したときの対処方法（HiRDB Datareplicator） 1052
20.10 OS が異常終了したときの対処方法 1053
20.10.1 HiRDB の処理及び HiRDB 管理者の処置 1053
20.11 OLTP システムとの連携中に障害が発生したときの対処方法 1054
20.11.1 OLTP システムとの連携中に通信障害などが発生したときの対処方法 1054
20.11.2 障害によってトランザクションが FORGETTING 状態になったときの対処方法 1055
20.11.3 フロントエンドサーバがあるユニットが稼働していないためトランザクションの居残りが発生したときの対処方法 1057
20.12 通信障害，CPU 障害，又は電源障害が発生したときの対処方法 1058
20.12.1 通信障害が発生したときの対処方法 1058
20.12.2 CPU 障害が発生したときの対処方法 1058
20.12.3 電源障害が発生したときの対処方法 1058
20.13 接続中のユーザが存在するため HiRDB が終了できないときの対処方法 1060
20.13.1 対処手順 1060
20.13.2 接続ユーザ情報ファイルと接続ユーザ詳細ファイル 1064
20.14 未決着状態のトランザクションがあるときの対処方法 1067
20.14.1 未決着状態のトランザクションを決着する方法 1067
20.14.2 未決着状態のトランザクションを手動で決着する方法 1073
20.15 縮退起動をするときの運用方法（HiRDB/パラレルサーバ限定） 1079
20.15.1 縮退起動をするには 1079

20.15.2 正常運転に戻す方法 1079
20.15.3 縮退起動ができない場合があります 1079
20.15.4 縮退起動中は表にアクセスできないことがあります 1080
20.15.5 再開始時に縮退起動をした場合の注意事項 1080
20.15.6 マルチフロントエンドサーバの場合の注意事項 1080
20.15.7 HiRDB Datareplicator と連携している場合の注意事項 1080
20.15.8 次回以降の HiRDB 開始時も縮退起動になる場合の運用 1081
20.15.9 回復不要 FES 使用時の注意事項 1082
20.16 ディスク障害が発生したときの対処方法 1083
20.16.1 対処手順 1083
20.17 システムログファイルの不足によって HiRDB（ユニット）が異常終了したときの対処方法 1086
20.17.1 ユニットの再開始手順（ユニットの再開始を最優先する場合） 1086
20.17.2 ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合） 1091
20.17.3 ユニットの再開始手順（原因を特定してからユニットを再開始する場合） 1096
20.17.4 ユニットの再開始手順（システムログファイルの不足原因が特定されている場合） 1106
20.17.5 システムログファイルをアンロード済み状態に変更する方法 1110
20.17.6 ユニットの再開始時に必要となるレコード数の求め方 1112
20.17.7 システムログファイルの追加方法 1115
20.17.8 追加したシステムログファイルを削除する方法 1118
20.17.9 抽出未完了状態のシステムログファイルがある場合の対処方法 1119
20.17.10 システムログファイルの不足を解消した後に実施すること 1120
20.18 同期点指定の再編成実行中にユティリティが異常終了したときの対処方法 1122
20.18.1 対処方法の概要 1122
20.18.2 例題 1123
20.18.3 アンロードデータファイルを一元化しないときに異常終了した場合の対処方法（HiRDB/パラレルサーバ限定） 1124
20.18.4 注意事項 1126
20.19 RD エリアのページ破壊を検知したときの対処方法 1127
20.19.1 ページ破壊が発生する要因 1127
20.19.2 対処方法 1127
20.20 RD エリアの入出力エラー（パス障害）が発生したときの対処方法 1129
20.20.1 RD エリアの入出力エラーが発生したときの HiRDB の処理 1129
20.20.2 pd_db_io_error_action ＝ unitdown（HiRDB の異常終了）を選択するときは 1129
20.20.3 環境設定 1130
20.20.4 運用方法 1131
20.20.5 注意事項 1131
20.21 コミット処理中に障害が発生したときのトランザクション決着種別の確認方法（HiRDB/パラレルサーバ限定） 1132
20.22 ローカルバッファを使用して共用表を更新した場合に障害が発生したときの対処方法（HiRDB/パラレルサーバ限定） 1136

20.23 システムマネジャユニットに障害が発生したときの対処方法 1137
20.24 ミラー正副不一致状態が発生したときの対処方法 1138
20.24.1 ミラー正副不一致状態を解消する方法 1138
20.25 HiRDB ファイルシステム領域に障害が発生した場合の対処方法 1140
20.25.1 管理できない状態のファイル，及び参照できない領域 1140
20.25.2 領域管理情報の破壊（バージョン 07-02 以前の HiRDB の場合） 1141

21 データベースの回復方法 1143
21.1 データベース回復の概要 1144
21.1.1 データベースをどの状態に回復できるのか？ 1144
21.1.2 バックアップ取得モードとの関連 1147
21.1.3 ログ取得モードとの関連 1148
21.1.4 各種 RD エリア回復時の注意 1148
21.1.5 同一ログの再使用チェック 1151
21.1.6 64 ビットモードの HiRDB を使用している方へ 1154
21.1.7 テープ装置アクセス機能 1154
21.2 データベースをバックアップ取得時点に回復する方法 1155
21.2.1 例題 1（HiRDB Control Manager のウィザードで全 RD エリアを回復する場合） 1155
21.2.2 例題 2（全 RD エリアを回復する場合） 1156
21.2.3 例題 3（RD エリア単位に回復する場合） 1158
21.2.4 例題 4（RD エリア単位に回復する場合：マスタディレクトリ用 RD エリアを含む） 1159
21.3 データベースを最新の同期点に回復する方法 1161
21.3.1 例題 1（全 RD エリアを回復する場合） 1161
21.3.2 例題 2（HiRDB Control Manager のウィザードで RD エリア単位に回復する場合） 1164
21.3.3 例題 3（RD エリア単位に回復する場合） 1166
21.3.4 例題 4（RD エリア単位に回復する場合：マスタディレクトリ用 RD エリアを含む） 1168
21.3.5 例題 5（RD エリア単位に回復する場合：アンロードレスシステムログ運用の場合） 1171
21.4 差分バックアップ機能使用時のデータベースの回復方法 1173
21.4.1 例題 1（最新の差分バックアップ取得時点に回復する場合） 1173
21.4.2 例題 2（最新の同期点に回復する場合） 1174
21.4.3 差分バックアップ管理ファイルがない場合の回復方法 1176
21.5 データベースをほかの HiRDB システムのバックアップから回復する方法 1178
21.5.1 例題（テスト環境の HiRDB システム A で取得したバックアップを本番環境の HiRDB システム B に移行） 1179
21.6 pdcopy コマンド以外でバックアップを取得した場合の回復方法 1181
21.6.1 例題 1（全 RD エリアをバックアップ取得時点に回復する場合） 1181
21.6.2 例題 2（RD エリア単位にバックアップ取得時点に回復する場合） 1182
21.6.3 例題 3（全 RD エリアを最新の同期点に回復する場合） 1182
21.6.4 例題 4（RD エリア単位に回復する場合） 1185
21.6.5 例題 5（マスタディレクトリ用 RD エリアだけを回復する場合） 1187

21.7 1:1 スタンバイレス型系切り替え時のシステム用 RD エリアを含む回復方法 1189
21.7.1 例題 1（全 RD エリアを最新の同期点に回復する場合） 1189

22 チューニング情報の取得方法 1192
22.1 統計ログからチューニング情報を取得する方法 1193
22.1.1 統計ログから取得できるチューニング情報 1193
22.1.2 チューニング情報を取得するための準備 1194
22.1.3 チューニング情報の取得 1196
22.1.4 アプリケーションから HiRDB にアクセスする場合 1198
22.2 システムログからチューニング情報を取得する方法 1203
22.2.1 システムログから取得できるチューニング情報 1203
22.2.2 チューニング情報の取得手順 1203
22.3 データベース状態解析ユティリティからチューニング情報を取得する方法 1205
22.3.1 チューニング情報を取得するための準備 1205
22.3.2 チューニング情報の取得 1205

23 チューニング 1208
23.1 グローバルバッファプールのチューニング 1209
23.1.1 pdbufls コマンドで統計情報を取得した場合 1209
23.1.2 統計解析ユティリティで統計情報を取得した場合 1216
23.2 デファードライト処理のチューニング 1226
23.2.1 デファードライト処理のチューニングで確認すること 1226
23.3 デファードライト処理適用時のシンクポイント処理時間のチューニング 1229
23.3.1 チューニングの手順 1229
23.3.2 デファードライト処理に関する統計情報の見方 1231
23.3.3 シンクポイント処理時間を短縮する方法 1236
23.4 シンクポイントダンプの取得間隔のチューニング 1239
23.5 バッファ長のチューニング 1241
23.5.1 表定義情報用バッファ長のチューニング 1241
23.5.2 ビュー解析情報用バッファ長のチューニング 1241
23.5.3 ユーザ権限情報用バッファ長のチューニング 1242
23.5.4 SQL オブジェクト用バッファ長のチューニング 1242
23.5.5 ユーザ定義型情報用バッファ長のチューニング 1245
23.5.6 ルーチン定義情報用バッファ長のチューニング 1246
23.5.7 レジストリ情報用バッファ長のチューニング 1248
23.6 プロセス数のチューニング 1249
23.6.1 最大起動プロセス数のチューニング 1249
23.6.2 常駐プロセス数のチューニング 1250
23.6.3 非同期 READ プロセス数のチューニング 1251

23.7 インデクスのチューニング 1253
23.7.1 インデクスのチューニングで確認すること 1253
23.8 データベースのチューニング 1255
23.8.1 RD エリア単位の物理的解析情報 1255
23.8.2 RD エリア単位の論理的解析情報 1256
23.8.3 表，インデクス単位の論理的解析情報 1257
23.8.4 クラスタキー及びクラスタリングデータページの格納状態解析 1259
23.9 SQL のチューニング 1261
23.9.1 SQL のチューニング方法 1261
23.10 ロールバック用ログ入力バッファ面数のチューニング 1271

24 セキュリティ監査機能の運用 1272
24.1 セキュリティ監査機能の概要 1273
24.1.1 セキュリティ監査機能とは 1273
24.1.2 監査証跡の取得契機 1274
24.1.3 監査証跡の取得例 1274
24.1.4 監査証跡として取得する情報 1276
24.1.5 監査証跡の参照 1276
24.1.6 前提となるシステム構成（HiRDB/パラレルサーバ限定） 1278
24.1.7 監査対象になるイベント 1278
24.1.8 監査証跡表の自動データロード機能 1283
24.2 監査証跡ファイルに出力される情報 1285
24.3 監査証跡の出力パターン 1287
24.3.1 権限チェック時の出力パターン 1287
24.3.2 イベント終了時の出力パターン 1287
24.3.3 監査証跡の関連づけ 1292
24.4 環境設定方法 1294
24.4.1 セキュリティ監査機能に関するオペランドの指定 1294
24.4.2 監査証跡ファイル用の HiRDB ファイルシステム領域の作成 1296
24.4.3 監査人の登録，監査証跡表を格納する RD エリアの作成，及び監査証跡表の作成 1297
24.4.4 監査対象イベントの定義 1300
24.5 運用方法 1301
24.5.1 HiRDB 管理者が行う操作 1301
24.5.2 監査人が行う操作 1303
24.5.3 監査証跡表の自動データロード機能を適用した場合の運用 1304
24.6 監査証跡ファイルの運用 1314
24.6.1 監査証跡ファイルの作成 1314
24.6.2 監査証跡ファイルの状態 1319
24.6.3 監査証跡ファイルのスワップ 1320

24.6.4 監査証跡ファイルの容量変更 1322
24.7 監査証跡表へのデータ登録 1324
24.7.1 例題 1（特定の監査証跡ファイルをデータロードする場合） 1324
24.7.2 例題 2（HiRDB ファイルシステム領域内の全監査証跡ファイルをデータロードする場合） 1325
24.7.3 データロード中にエラーが発生した場合の対処方法 1326
24.8 監査証跡表の列構成 1328
24.8.1 監査証跡表の列構成の詳細 1328
24.9 監査証跡の絞り込み 1357
24.9.1 選択項目 1357
24.9.2 選択できる単位 1359
24.9.3 選択項目の組み合わせ 1360
24.9.4 セキュリティ監査情報用バッファ 1360
24.9.5 セキュリティ監査情報用バッファに関するエラーが発生した場合の HiRDB の動作 1363
24.9.6 セキュリティ監査情報用バッファの状態遷移 1365
24.10 監査証跡ファイルに障害が発生したときの対処方法 1366
24.10.1 監査証跡ファイルに障害が発生した場合 1366
24.10.2 監査証跡ファイル用の HiRDB ファイルシステム領域に障害が発生した場合（全監査証跡ファイルが閉塞状態になった場合） 1366
24.10.3 スワップ先にできる監査証跡ファイルがないため HiRDB が強制終了した場合 1367
24.10.4 非同期出力用バッファのすべての面がフラッシュ待ちになった場合 1372
24.11 自動データロード機能適用中に障害が発生したときの対処方法 1373
24.11.1 自動データロード実行中に障害が発生した場合 1373
24.11.2 自動データロード機能適用中に HiRDB で障害が発生した場合 1375
24.12 ほかの機能との関連 1378
24.12.1 セキュリティ監査機能とほかの機能との関連 1378
24.13 監査証跡のレコード項目（権限チェック時） 1379
24.14 監査証跡のレコード項目（イベント終了時） 1443
24.15 ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定） 1509
24.16 バージョンアップ時の注意事項 1511
24.16.1 監査証跡ファイルの監査証跡形式が変更になる場合 1511
24.16.2 監査証跡表の列構成が変更になる場合 1511
24.16.3 監査証跡のイベントが増えた場合 1511
24.16.4 バージョンアップ失敗時の注意事項 1512
24.16.5 監査証跡表を格納している RD エリアのバックアップ 1512
24.17 JP1/NETM/Audit との連携 1513
24.17.1 JP1/NETM/Audit との連携の概要 1513
24.17.2 環境設定 1515
24.17.3 運用方法 1519
24.17.4 JP1/NETM/Audit 用監査ログ出力ファイルに関する変更 1523

24.17.5 JP1 側で障害が発生した場合 1524
24.17.6 注意事項 1525

25 CONNECT 関連セキュリティ機能の運用 1526
25.1 CONNECT 関連セキュリティ機能の概要 1527
25.1.1 CONNECT 関連セキュリティ機能とは 1527
25.1.2 パスワードの文字列制限 1527
25.1.3 連続認証失敗回数の制限 1529
25.1.4 IP アドレスによる接続制限 1531
25.1.5 パスワードの有効期間の設定 1533
25.2 パスワードの文字列制限を新規設定する 1534
25.2.1 パスワードの文字列制限の新規設定手順 1534
25.3 パスワードの文字列制限を変更する 1537
25.3.1 パスワードの文字列制限を変更するときの留意事項 1537
25.3.2 パスワードの文字列制限の変更手順 1537
25.4 パスワード無効アカウントロック状態を解除する 1540
25.4.1 ユーザ単位にパスワード無効アカウントロック状態を解除する 1540
25.4.2 全ユーザのパスワード無効アカウントロック状態を解除する 1541
25.5 パスワード無効アカウントロック状態になるユーザを事前調査する 1542
25.5.1 パスワード無効アカウントロック状態になるユーザの事前調査手順 1542
25.6 パスワード無効アカウントロック状態のユーザの権限付与及び削除 1545
25.7 パスワードの文字列制限を解除する 1546
25.8 セキュリティ監査機能使用時の注意事項 1547
25.9 連続認証失敗回数の制限を設定又は解除する 1548
25.9.1 連続認証失敗回数の制限を新規設定する 1548
25.9.2 連続認証失敗回数の制限を解除する 1549
25.9.3 連続認証失敗回数の制限を変更する 1549
25.9.4 連続認証失敗許容回数及びアカウントロック期間を確認する 1549
25.10 連続認証失敗アカウントロック状態のユーザを確認する 1551
25.11 連続認証失敗アカウントロック状態を解除する 1553
25.12 IP アドレスによる接続制限の設定方法 1554
25.12.1 接続制限対象の登録 1554
25.12.2 接続制限対象の削除 1554
25.12.3 登録時の注意事項 1555
25.12.4 接続可否の決定方法 1555
25.13 IP アドレスによる接続制限の運用例 1558
25.13.1 特定の IP アドレスだけ接続許可したい場合 1558
25.13.2 一定範囲内の IP アドレスからの接続を許可するが，一部の IP アドレスは例外的に拒否したい場合1558
25.13.3 特定の IP アドレスだけ接続を拒否したい場合 1558

25.13.4 登録を変更したいとき 1559
25.13.5 FES 又は SDS のあるユニットのマシンからのローカル接続を拒否したいとき 1559
25.13.6 特定のユーザにだけ接続制限を設定したいとき 1559
25.14 パスワード有効期間の設定方法 1561
25.14.1 パスワード有効期間の設定 1561
25.14.2 設定内容の格納場所と参照方法 1561
25.14.3 パスワード有効期間の削除 1561
25.14.4 パスワードの有効期間の設定変更 1562
25.14.5 有効期限日時の確認方法 1562
25.14.6 有効期間が過ぎてしまった場合の対処 1562
25.14.7 パスワード有効期間の運用例 1562
25.14.8 パスワード有効期間運用時の注意事項 1562
25.15 CONNECT 関連セキュリティ機能運用時の注意事項 1563
25.15.1 二重ロックの解除 1563
25.15.2 ディクショナリ用 RD エリアを回復するときの注意事項 1563

26 系切り替え機能の運用 1564
26.1 系切り替え機能の概要 1565
26.1.1 系切り替え機能とは 1565
26.1.2 スタンバイ型系切り替え機能 1568
26.1.3 スタンバイレス型系切り替え機能 1573
26.1.4 系切り替え機能の適用基準 1590
26.1.5 HiRDB がサポートしているクラスタソフトウェア 1591
26.2 システム構成例 1594
26.2.1 HiRDB/シングルサーバのシステム構成例（スタンバイ型系切り替え） 1594
26.2.2 HiRDB/パラレルサーバのシステム構成例（スタンバイ型系切り替え） 1596
26.2.3 1：1 スタンバイレス型系切り替えのシステム構成例 1601
26.2.4 影響分散スタンバイレス型系切り替えのシステム構成例 1604
26.3 IP アドレスの構成例 1611
26.3.1 IP アドレスを引き継ぐ場合 1611
26.3.2 IP アドレスを引き継がない場合 1612
26.4 IP アドレスを引き継ぐかどうかでのホスト名の運用方法の違い 1615
26.4.1 HiRDB/シングルサーバの場合 1615
26.4.2 HiRDB/パラレルサーバの場合 1618
26.5 HiRDB に関する準備 1626
26.5.1 前提条件及び注意事項 1626
26.5.2 共有ディスク装置の準備 1628
26.5.3 HiRDB システム定義の作成 1631
26.5.4 クライアント環境定義の指定 1645

26.5.5 HiRDB システム定義及びクライアント環境定義のホスト名の指定例 1646
26.5.6 RD エリアの作成 1648
26.5.7 グローバルバッファの定義（1：1 スタンバイレス型系切り替え機能限定） 1650
26.5.8 グローバルバッファの定義（影響分散スタンバイレス型系切り替え機能限定） 1653
26.5.9 HiRDB サービスの開始方法の設定の確認（HiRDB/シングルサーバ限定） 1671
26.5.10 HiRDB に関連するサービスの準備 1671
26.5.11 監査証跡ファイルの運用 1672
26.5.12 NetBackup 連携機能を使用する場合の注意事項 1675
26.6 MSFC に関する準備（Windows Server 2012 の場合） 1676
26.6.1 サービスの作成 1676
26.6.2 リソースの登録 1678
26.6.3 共有ディスクの登録 1680
26.6.4 クライアントアクセスポイントの登録（モニタモードの場合だけ） 1681
26.6.5 HiRDB のサービスの登録（モニタモードの場合だけ） 1682
26.6.6 HiRDB のクラスタサービスの登録（モニタモードの場合だけ） 1685
26.7 MSFC に関する準備（Windows Server 2012/R2 以降の場合） 1688
26.7.1 役割の作成 1688
26.7.2 リソースの登録 1690
26.7.3 共有ディスクの登録 1692
26.7.4 クライアントアクセスポイントの登録（モニタモードの場合だけ） 1693
26.7.5 HiRDB のサービスの登録（モニタモードの場合だけ） 1694
26.7.6 HiRDB のクラスタサービスの登録（モニタモードの場合だけ） 1697
26.8 Hitachi HA Toolkit Extension に関する準備（サーバモード限定） 1701
26.8.1 sysdef 定義ファイル 1701
26.8.2 servers 定義ファイル 1701
26.9 CLUSTERPRO に関する準備 1705
26.9.1 グループリソースの作成 1705
26.9.2 モニタリソースの作成 1707
26.9.3 IP アドレスに関連するリソースの作成 1707
26.9.4 ディスクに関連するリソースの作成 1707
26.9.5 スクリプト実行に関連するリソースの作成 1709
26.9.6 HiRDB クライアントに関連する設定項目 1713
26.9.7 Hitachi HA Toolkit Extension に関連するリソースの作成（サーバモードの場合だけ） 1715
26.10 MSFC 及び DataKeeper に関する準備 1716
26.10.1 DataKeeper ジョブ，及び DataKeeper ボリュームリソースの作成 1716
26.10.2 DataKeeper ボリュームリソースの登録 1720
26.11 パブリッククラウド環境に関する準備 1722
26.11.1 クラスタソフトウェアが MSFC の場合 1722
26.11.2 クラスタソフトウェアが CLUSTERPRO の場合 1725

26.12 HiRDB の運用方法の違いは？ 1746
26.12.1 HiRDB の開始方法（サーバモードの場合） 1746
26.12.2 HiRDB の開始方法（モニタモードの場合） 1759
26.12.3 HiRDB の終了方法（サーバモードの場合） 1760
26.12.4 HiRDB の終了方法（モニタモードの場合） 1776
26.12.5 状態の確認 1777
26.12.6 統計ログファイルの運用 1779
26.12.7 運用上の注意事項 1785
26.12.8 スタンバイレス型系切り替え機能使用時の注意事項 1786
26.12.9 パブリッククラウド環境に関する運用時の注意事項 1788
26.13 計画系切り替え 1790
26.13.1 スタンバイ型系切り替え機能の場合 1790
26.13.2 1：1 スタンバイレス型系切り替え機能の場合 1792
26.13.3 影響分散スタンバイレス型系切り替え機能の場合 1793
26.14 連動系切り替え 1798
26.15 障害発生時の HiRDB 管理者の処置 1799
26.15.1 HiRDB 管理者の処置 1799
26.16 系が切り替わった後の運用方法 1801
26.16.1 系が切り替わった後に行うこと 1801
26.17 系の切り替え時間の短縮（ユーザサーバホットスタンバイ，高速系切り替え機能） 1805
26.17.1 ユーザサーバホットスタンバイ 1805
26.17.2 高速系切り替え機能 1805
26.17.3 高速系切り替え機能使用時のシステム構成例 1806
26.17.4 待機系の起動に時間が掛かる場合の確認作業 1809
26.17.5 高速系切り替え機能使用時の注意事項 1810
26.18 トランザクションキューイング機能 1813
26.18.1 トランザクションキューイング機能とは 1813
26.18.2 環境設定方法 1814
26.18.3 キューイングの対象になるトランザクション 1816
26.18.4 注意事項 1816
26.19 サーバ障害以外の障害が発生した場合に系を切り替える方法 1819
26.19.1 サーバプロセスの異常終了が多発した場合に系を切り替える方法 1819
26.19.2 RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える方法 1822
26.20 システム構築例 1824
26.20.1 システム構築例の説明 1824
26.21 停止中のユニットがあるためシステムマネジャユニットの系切り替えに失敗するときの対処方法 1834
26.21.1 縮退起動で対処する方法 1834
26.21.2 pd_ha_mgr_rerun オペランドを指定して対処する方法 1835
26.22 NTFS キャッシュのフラッシュが発生した場合の対処 1838

26.22.1 系が切り替わる原因 1838
26.22.2 対処方法 1839

27 MIB パフォーマンス情報監視機能の運用 1841
27.1 MIB パフォーマンス情報監視機能の概要 1842
27.1.1 MIB パフォーマンス情報監視機能とは 1842
27.1.2 MIB パフォーマンス情報監視機能の目的 1844
27.1.3 MIB 定義ファイル 1844
27.1.4 MIB 環境定義ファイル 1845
27.2 システム構成 1846
27.2.1 HiRDB/シングルサーバの場合 1846
27.2.2 HiRDB/パラレルサーバの場合 1846
27.2.3 マルチ HiRDB の場合 1847
27.2.4 系切り替え機能を使用する場合 1848
27.3 環境設定 1850
27.3.1 環境設定の手順 1850
27.4 MIB 定義ファイル 1852
27.4.1 HiRDB が提供する MIB 定義ファイルの規則 1852
27.4.2 HiRDB が提供する MIB テーブルの一覧 1852
27.5 サーバ稼働状態テーブル（hirServerStatusTable） 1854
27.6 作業表用 HiRDB ファイルシステム領域テーブル（hirFileSystemTable） 1856
27.7 RD エリアテーブル（hirRdareaStatusTable） 1858
27.8 RD エリア詳細テーブル（hirRdareaDetStatusTable） 1860
27.9 グローバルバッファテーブル（hirBufferStatusTable） 1865
27.10 RD エリア–HiRDB ファイルシステム領域テーブル（hirRdareaFileTable） 1868
27.11 SYS 統計情報テーブル（hirStatisInfSysTable） 1871
27.12 ディスク使用量 1893

28 PRF トレース機能 1895
28.1 PRF トレース機能とは 1896
28.1.1 PRF トレース機能の取得対象となるプロセス 1897
28.1.2 PRF トレース情報を取得するための設定 1897
28.1.3 PRF トレース情報ファイルの運用 1898
28.1.4 Cosminexus と連携した PRF トレース情報の取得 1898
28.2 PRF トレース情報の参照手順 1900
28.2.1 PRF トレース情報の取得内容 1900
28.3 注意事項 1902
28.4 PRF トレース情報の詳細と取得ポイント 1903
28.4.1 SQL 実行時の PRF トレース情報の詳細と取得ポイント 1903

28.4.2 ユティリティ実行時の PRF トレース情報の詳細と取得ポイント 1909

29 HiRDB 接続時のパスワード秘匿化機能の運用 1911

29.1 HiRDB 接続時のパスワード秘匿化機能の概要 1912
29.2 HiRDB 接続時のパスワード秘匿化機能の適応範囲 1913
29.3 HiRDB 接続時のパスワード秘匿化機能の設定方法 1914
29.4 HiRDB 接続時のパスワード秘匿化機能の運用例 1916
29.5 HiRDB 接続時のパスワード秘匿化機能の注意事項 1917

付録 1918

付録 A Q & A 1919
付録 A.1 システムログファイルに関する質問 1919
付録 A.2 シンクポイントダンプファイルに関する質問 1921
付録 A.3 ステータスファイルに関する質問 1923
付録 A.4 障害に関する質問 1924
付録 A.5 表，及びインデクスに関する質問 1925
付録 A.6 HiRDB の開始に関する質問 1926
付録 A.7 HiRDB の終了に関する質問 1929
付録 A.8 性能に関する質問 1930
付録 A.9 バックアップに関する質問 1931
付録 A.10 RD エリアの回復に関する質問 1933
付録 A.11 そのほかの質問 1934
付録 B DVD-RAM ライブラリ装置を使用するときの運用 1936
付録 B.1 システム共通定義の設定 1936
付録 B.2 HiRDB ファイルシステム領域の作成 1936
付録 B.3 RD エリアのオープン契機 1936
付録 B.4 同時にアクセスする媒体数 1937
付録 B.5 異常終了時の回復処理 1937
付録 B.6 基本的な運用方針 1938
付録 C 障害調査のために必要な情報 1939
付録 D HiRDB を 24 時間連続稼働するときの注意 1943
付録 D.1 システム構成変更コマンド（pdchgconf コマンド） 1943
付録 D.2 HiRDB システム定義の指定 1944
付録 D.3 バックアップの取得 1945
付録 D.4 データベースの再編成 1945
付録 D.5 使用中空きページとページ内の空き領域の再利用 1946
付録 D.6 RD エリアの拡張 1946
付録 D.7 グローバルバッファの動的変更 1947
付録 D.8 トラブルシュート情報の削除 1948

付録 D.9　系切り替え機能　1949
付録 D.10　プログラムメンテナンス機能（修正版レベルのバージョンアップ）　1950
付録 D.11　回復不要 FES（HiRDB/パラレルサーバ限定）　1950
付録 E　性能向上機能の運用　1951
付録 E.1　バックエンドサーバ接続保持機能（HiRDB/パラレルサーバ限定）　1951
付録 F　HiRDB Control Manager との連携　1958
付録 F.1　HiRDB Control Manager - Console の起動方法　1958
付録 F.2　管理 HiRDB の登録方法　1958
付録 G　論理セクタ長が 4096 バイトのディスク使用時における運用　1960
付録 G.1　ディスクの論理セクタ長の確認方法　1960
付録 G.2　適用範囲　1960
付録 G.3　運用方法　1960
付録 G.4　注意事項　1962

索引　1963

1 HiRDB の開始及び終了方法

この章では，HiRDB の開始及び終了方法について説明します。

なお，HiRDB/パラレルサーバの場合，ユニット単位又はサーバ単位でも，開始及び終了操作ができます。

1.1　開始方法

実行者　HiRDB 管理者

HiRDB は **pdstart** コマンドで開始します。ここでは，HiRDB の開始モード及び開始方法について，次の項目を説明します。

- 開始モード
- pdstart コマンドを実行するサーバマシン
- 自動開始
- 縮退起動（HiRDB/パラレルサーバ限定）
- 例題（HiRDB を正常開始する場合）
- 開始方法のカスタマイズ
- 開始処理完了の確認

1.1.1　開始モード

HiRDB には，次の表に示す開始モードという概念があります。開始モードによって，pdstart コマンドのオプションが異なります。

表 1–1　HiRDB の開始モード

開始モード	実行コマンド	開始モードの説明	前回の終了モード
正常開始	**pdstart**	通常の開始モードです。正常開始では，前回稼働時の情報を引き継ぎません。 ただし，次に示す情報は引き継ぎます。 • 障害閉塞している RD エリアの状態	正常終了
再開始※1		前回の終了モードが右記の場合，HiRDB は（自動的に）再開始となります。再開始では，前回稼働時の情報を引き継ぎます。再開始時に引き継ぐ情報の詳細は，「HiRDB が再開始するときに引き継ぐ情報」を参照してください。	計画停止 強制終了 異常終了
強制開始※2	**pdstart dbdestroy**	通常は使用しないでください。HiRDB を再開始できないときに使用します。HiRDB を強制的に開始します。	なし
–※3	**pdstart -i**	通常は使用しないでください。データベースを再度初期設定するときの開始方法です。	
	pdstart -r	通常は使用しないでください。マスタディレクトリ用 RD エリアに障害が発生したときの開始方法です。	

開始モード	実行コマンド	開始モードの説明	前回の終了モード
	pdstart -R※4	pdstart -r コマンドで HiRDB/パラレルサーバを開始した後，システムマネジャがあるユニットに障害が発生し，異常終了した場合に使用します。このコマンドを実行すると，異常終了したシステムマネジャがあるユニットをいったん強制的に開始し，その後 HiRDB/パラレルサーバを強制終了します。	
	pdstart -a	通常は使用しないでください。フロントエンドサーバが SUSPEND 状態のときの開始方法です。	

注※1

再開始をする前に，必ず「計画停止，強制終了，又は異常終了したときの注意事項」を参照してください。

注※2

強制開始をする前に，必ず「HiRDB（ユニット）を強制開始するときの注意事項」を参照してください。

注※3

これらの開始方法の利用方法については，「特別な開始方法」を参照してください。

注※4

このコマンドの使用については，「マスタディレクトリ用 RD エリアに障害が発生したときの開始方法（pdstart -r）」の「pdstart -r コマンドで開始した場合に障害が発生したときの対処」を参照してください。

1.1.2 pdstart コマンドを実行するサーバマシン

HiRDB/シングルサーバを開始する場合は，シングルサーバを定義したサーバマシンのコマンドプロンプトで pdstart コマンドを実行してください。

HiRDB/パラレルサーバを開始する場合は，システムマネジャを定義したサーバマシンのコマンドプロンプトで pdstart コマンドを実行してください。

1.1.3 自動開始

システム共通定義の **pd_mode_conf** オペランドの指定で，HiRDB を自動開始できます。自動開始とは，OS を起動すると，HiRDB も自動的に開始する方法のことです。また，pdstart コマンドで HiRDB を開始する方法を**手動開始**といいます。

注意事項

- 自動開始する場合は，Windows のサービスで HiRDB のサービス開始を自動にしてください。HiRDB のサービス開始を自動にする方法については，「Windows の［サービス］ダイアログボックスでスタートアップの種類を変更する場合」を参照してください。HiRDB Version 6 以降を新規に導入した場合は，HiRDB のサービス開始は自動になっているため，この作業は必要ありません。
- pd_mode_conf オペランドで HiRDB を自動開始する設定をしておくと，HiRDB（ユニット）が異常終了した場合も，自動的に HiRDB（ユニット）が再開始されます。ただし，再開始時に 3 回連続して異常終了すると，自動的に再開始しなくなります。
- pd_mode_conf オペランドに AUTO や MANUAL1 を指定した場合でも，HiRDB の開始処理中又は終了処理中に HiRDB が異常終了したときは，次回の開始は必ず手動開始となります。異常終了時のメッセージを確認して対処したあと，手動で HiRDB を開始してください。
- HiRDB/パラレルサーバで自動開始（pd_mode_conf=AUTO）をする場合は，最初のユニットを開始してから 20 分以内に全ユニットを開始するようにしてください。20 分以内に全ユニットを開始しないと，HiRDB の開始処理を中止します。この 20 分という制限時間は pd_reduced_check_time オペランドで変更できます。なお，ユニットの異常終了，又は OS の異常終了後の再起動ではこのような制限時間はありません。

1.1.4　縮退起動（HiRDB/パラレルサーバ限定）

障害などで起動できないユニットがある場合，縮退起動機能を使用すると，残りの正常なユニットだけで HiRDB を開始できます。これを**縮退起動**といいます。縮退起動をするには次に示すオペランドを指定します。

- pd_start_level
- pd_start_skip_unit

なお，縮退起動の運用方法については，「縮退起動をするときの運用方法（HiRDB/パラレルサーバ限定）」を参照してください。

1.1.5　例題（HiRDB を正常開始する場合）

HiRDB を正常開始します。

なお，HiRDB/パラレルサーバの場合は(1)の操作は不要です。

(1)　Windows のサービスで HiRDB/Single Server を開始します

この操作は通常必要ありません。HiRDB Version 5.0 からバージョンアップした場合だけこの操作が必要になります。なお，HiRDB Version 5.0 からバージョンアップしたユーザでこの操作を省略したい場合は，「開始方法のカスタマイズ」を参照してください。

〈手順〉

1. **[コントロールパネル]** の **[サービス]** をダブルクリックします。
2. **[サービス]** リストボックスから，HiRDB/SingleServer を選択し，**[開始]** ボタンをクリックします。
3. **[サービスコントロール]** ダイアログが表示されます。



> **注意事項**
>
> この **[サービスコントロール]** ダイアログは，pdstart コマンドが入力されて HiRDB が開始されるまで表示されます。したがって，**[サービスコントロール]** ダイアログの表示が終了するのを待つ必要はありません。2 の操作を終えたら(2)に進んでください。

(2)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(3)　pdstart コマンドの実行結果が表示されます

```
139 14:56:33 unt1 _rdm     KFPS05210-I HiRDB system initialization
 process complete
```

(4)　pdls コマンドで HiRDB の稼働状況を調べます

```
pdls
```

(5)　pdls コマンドの実行結果が表示されます

```
HOSTNAME(145750)     UNITID  SVID      STATUS   STARTTIME
k95x620              unt1    ********  ACTIVE   145632
k95x620              unt1    sds01     ACTIVE   145632
```

〔説明〕

STATUS に ACTIVE が表示されているので，HiRDB が開始しました。

pdls コマンドの実行結果は HiRDB/シングルサーバの例です。

1.1.6　開始方法のカスタマイズ

(1)　HiRDB のサービスの開始操作を省略できます（HiRDB/シングルサーバ限定）

「例題（HiRDB を正常開始する場合）」の(1)の操作（HiRDB のサービスの開始）を省略できます。省略するには，次に示すどちらかの方法を実施してください。

- Windows の［サービス］ダイアログボックスでスタートアップの種類を変更する
- pdntenv コマンドを実行する

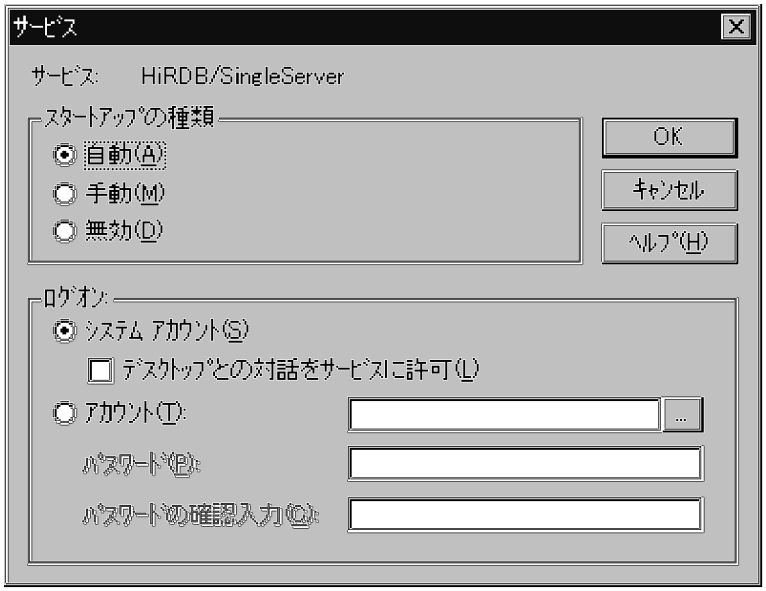
(a)　Windows の［サービス］ダイアログボックスでスタートアップの種類を変更する場合

Windows のサービスで HiRDB のサービス開始を自動にすると，HiRDB のサービスの開始操作を省略できます。HiRDB のサービス開始を自動にする手順を次に示します。

なお，HiRDB/パラレルサーバの場合は，HiRDB のサービス開始を必ず自動にしてください。通常は自動になっています。

〈手順〉

1. **［コントロールパネル］**の**［サービス］**をダブルクリックして HiRDB が開始状態でないことを確認します。
2. **［サービス］**リストボックスから HiRDB/Single Server を選択し，**［スタートアップ］**ボタンをクリックします。
3. **［デスクトップとの対話とサービスに許可］**チェックボックスをオフにします。コマンドプロンプトにメッセージは表示されません。HiRDB が開始状態のとき HiRDB のコマンドプロンプトを閉じた場合，HiRDB のプロセスが強制停止されることがあります。HiRDB の開始状態のときに HiRDB のコマンドプロンプトを使用しないように設定をしてください。
4. **［スタートアップの種類］**で**自動**を選択します。自動の設定は Windows の再起動後に有効となります。



(b)　pdntenv コマンドを実行する場合

pdntenv コマンドを実行すると，HiRDB の終了時に HiRDB のサービスを停止しないようにできます。サービスを停止しないようにすると，次回の HiRDB 開始時に「HiRDB のサービスの開始」操作が不要になります。ただし，HiRDB のサービス開始が手動の場合，Windows 開始後の最初の HiRDB 開始時に，「HiRDB のサービスの開始」操作が必要になります。

pdntenv コマンドの指定例を次に示します。

```
pdntenv -sc manual
```

(2)　pdstart コマンドの入力を省略できます

「例題（HiRDB を正常開始する場合）」の(2)では，pdstart コマンドを入力しています。この pdstart コマンドの入力を省略できます。pdstart コマンドの入力を省略する場合は，システム共通定義で **pd_mode_conf=AUTO** を指定してください。

「例題（HiRDB を正常開始する場合）」の(1)の操作を自動化している場合は，Windows を起動すると，自動的に HiRDB が開始します。「例題（HiRDB を正常開始する場合）」の(1)の操作を自動化していない場合は，Windows のサービスで HiRDB のサービスを開始すると HiRDB を開始できます。

ただし，pd_mode_conf=AUTO をすると，Windows の起動後に pdstart コマンドを入力しないため，pdstart コマンドのオプションを指定できないデメリットがあります（正常開始又は再開始しかできません）。したがって，通常はこの指定をしないことをお勧めします。

1.1.7　開始処理完了の確認

次の手順で HiRDB（又はユニット）の開始処理が完了したかを確認してください。

- HiRDB/シングルサーバの場合

  pdstart コマンドの終了ステータスが 0 であることを確認してください。

- HiRDB/パラレルサーバの場合

  pdstart コマンドを実行して HiRDB システムを開始した場合，pdstart コマンドの終了ステータスが 0 であることを確認してください。

  各 HiRDB ユニットで pdstart -q コマンドを実行して HiRDB ユニットを開始した場合，システムマネジャユニットがあるサーバマシンのイベントログに KFPS05210-I メッセージが出力されていることを確認してください。

  なお，HiRDB/パラレルサーバは，一つでも起動できないユニットがあると開始できません。

上記の方法で開始状態が確認できた場合，AP（SQL）を実行できます。

なお，pdstart コマンドの終了ステータスが 0 以外，又は KFPS05210-I メッセージが出力されていない場合，HiRDB の開始処理でエラーが発生しているおそれがあるため，次に示すどれかの方法で，状態を確認してください。

(1)　HiRDB の開始処理の状態を確認する

(a)　pdstart コマンドの終了ステータスで確認する

pdstart コマンドの終了ステータスが 0 以外の場合，HiRDB の開始処理中に障害が発生しています。イベントログに出力されたメッセージを参照して障害原因を取り除いた後，HiRDB が開始処理中の場合は HiRDB システム，又は開始中の HiRDB ユニットを強制終了（pdstop -f コマンド，又は pdstop -z コマンドで終了）して pdstart コマンドを再度実行してください。

(b)　pdls -d svr コマンドの終了ステータスで確認する

pdstart コマンドの実行後，任意のユニットから pdls -d svr コマンドを実行してください。pdls -d svr コマンドの終了ステータスによって次の表に示すように操作が異なります。

表 1–2　pdls -d svr コマンドの終了ステータスとそのときの操作（HiRDB/シングルサーバの場合）

pdls -d svr コマンドの終了ステータス	HiRDB 管理者の処置
0	HiRDB が稼働状態のため，対応不要です。
4	HiRDB が開始処理中の可能性があるため，5 秒程度の間隔で pd_system_complete_wait_time オペランドに指定した時間又は省略値を目安に，pdls -d svr コマンドを繰り返し実行してください。 pd_system_complete_wait_time オペランドの時間を経過しても状態が変わらない場合は，イベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，HiRDB が開始処理中の場合は HiRDB を強制終了（pdstop -f コマンドで終了）して pdstart コマンドを再度実行してください。
8	HiRDB の開始処理中に障害が発生しています。イベントログに出力されたメッセージを参照して障害原因を取り除いてください。

pdls -d svr コマンドの終了ステータス	HiRDB 管理者の処置
	障害原因を取り除いた後，HiRDB が開始処理中の場合は HiRDB を強制終了（pdstop -f コマンドで終了）して pdstart コマンドを再度実行してください。

表 1–3　pdls -d svr コマンドの終了ステータスとそのときの操作（HiRDB/パラレルサーバの場合）

pdls -d svr コマンドの終了ステータス	HiRDB 管理者の処置
0	HiRDB が稼働状態のため，対応不要です。
4	一部の HiRDB ユニットが開始処理中の可能性があるため，5 秒程度の間隔で pd_system_complete_wait_time オペランドに指定した時間又は省略値を目安に，pdls -d svr コマンドを繰り返し実行してください。 pd_system_complete_wait_time オペランドの時間を経過しても状態が変わらない場合は，システムマネジャユニットがあるサーバマシン，又は非システムマネジャユニットがあるサーバマシンのイベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，HiRDB システム又は HiRDB ユニットが開始中の場合は HiRDB を強制終了（pdstop -f コマンド，又は pdstop -z コマンドで終了）して pdstart コマンドを再度実行してください。
8	HiRDB の開始処理中に障害が発生しています。システムマネジャユニットがあるサーバマシン，又は非システムマネジャユニットがあるサーバマシンのイベントログに出力されたメッセージを参照して障害原因を取り除いてください。障害原因を取り除いた後，HiRDB システム又は HiRDB ユニットが開始中の場合は HiRDB を強制終了（pdstop -f コマンド，又は pdstop -z コマンドで終了）して pdstart コマンドを再度実行してください。

(2)　ユニットの開始処理の状態を確認する

pdstart コマンド，又は pdstart -q コマンドの実行後，各ユニットで pdls -d ust コマンドを実行してください。pdls -d ust コマンドの終了ステータスによって次の表に示すように操作が異なります。

表 1–4　pdls -d ust コマンドの終了ステータスとそのときの操作（HiRDB/シングルサーバの場合）

pdls -d ust コマンドの終了ステータス	HiRDB 管理者の処置
0	HiRDB が稼働状態のため，対応不要です。
1	pdls -d ust コマンド実行中に障害が発生しています。出力されたメッセージを参照して障害原因を取り除いた後に，再度 pdls -d ust コマンドを実行してください。
4	HiRDB が開始処理中の可能性があるため，5 秒程度の間隔で pd_system_complete_wait_time オペランドに指定した時間又は省略値を目安に，pdls -d ust コマンドを繰り返し実行してください。 pd_system_complete_wait_time オペランドの時間を経過しても状態が変わらない場合は，イベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，HiRDB が開始処理中の場合は HiRDB を強制終了（pdstop -f コマンドで終了）して pdstart コマンドを再度実行してください。

pdls -d ust コマンドの終了ステータス	HiRDB 管理者の処置
8 又は 12	HiRDB の開始処理中に障害が発生しています。イベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，HiRDB が開始処理中の場合は HiRDB を強制終了（pdstop -f コマンドで終了）して pdstart コマンドを再度実行してください。
16	実行する HiRDB の HiRDB サービスが停止している可能性があります。HiRDB サービスが停止している場合はサービスを起動した上で，pdls -d ust コマンドを再度実行してください。

表 1–5　pdls -d ust コマンドの終了ステータスとそのときの操作（HiRDB/パラレルサーバの場合）

pdls -d ust コマンドの終了ステータス	HiRDB 管理者の処置
0	ユニットが稼働状態のため，対応不要です。
1	pdls -d ust コマンド実行中に障害が発生しています。出力されたメッセージを参照して障害原因を取り除いた後に，再度 pdls -d ust コマンドを実行してください。
4	HiRDB ユニットが開始処理中の可能性があるため，5 秒程度の間隔で pd_system_complete_wait_time オペランドに指定した時間又は省略値を目安に，pdls -d ust コマンドを繰り返し実行してください。 pd_system_complete_wait_time オペランドの時間を経過しても状態が変わらない場合は，システムマネジャユニットがあるイベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，HiRDB ユニットが開始中の場合は HiRDB ユニットを強制終了（pdstop -z コマンドで終了）して pdstart -q コマンドを再度実行してください。
8 又は 12	HiRDB ユニットの開始処理中に障害が発生しています。システムマネジャユニットがあるサーバマシン，又は非システムマネジャユニットがあるサーバマシンのイベントログに出力されたメッセージを参照して障害原因を取り除いてください。障害原因を取り除いた後，HiRDB ユニットが開始処理中の場合は HiRDB ユニットを強制終了（pdstop -z コマンドで終了）して pdstart -q コマンドを再度実行してください。
16	実行する HiRDB ユニットの HiRDB サービスが停止している可能性があります。HiRDB サービスが停止している場合はサービスを起動した上で，pdls -d ust コマンドを再度実行してください。

1.2　終了方法

実行者　HiRDB 管理者

HiRDB は **pdstop** コマンドで終了します。ここでは，HiRDB の終了モード及び終了方法について，次の項目を説明します。

- 終了モード
- pdstop コマンドを実行するサーバマシン
- 例題（HiRDB を正常終了する場合）
- Windows のサービスで HiRDB を終了する方法
- 停止処理完了の確認

1.2.1　終了モード

HiRDB には，次の表に示す終了モードという概念があります。終了モードによって，pdstop コマンドのオプションが異なります。

表 1–6　HiRDB の終了モード

終了モード	入力コマンド	終了モードの説明
正常終了※1	**pdstop**	通常の終了モードです。CONNECT 要求を禁止し，すべてのユーザの処理が終了した後に HiRDB を終了します。ただし，ユティリティが実行中の場合は HiRDB を終了しないで，KFPS05074-E メッセージを出力します。このとき，pdstop コマンドはリターンコード 8 で終了します。
計画停止※1※2※3	**pdstop -P**	トランザクションの受け付けを禁止し，ユティリティを含むすべてのトランザクションが終了した後に，HiRDB を終了します。
強制終了	**pdstop -f**	処理中のトランザクションの完了を待たずに，HiRDB を直ちに終了します。処理中のトランザクションは，再開始時にロールバックの対象※4となります。
異常終了	なし	何らかの異常によって HiRDB が終了する場合の終了モードです（OS が異常終了する場合も該当します）。処理中のトランザクションの完了を待たずに，HiRDB は直ちに終了します。処理中のトランザクションは，再開始時にロールバックの対象※4となります。

注※1

OLTP システムと連携している場合，OLTP システムを終了した後に HiRDB を正常終了又は計画停止してください。OLTP システムを終了する前に HiRDB を正常終了又は計画停止すると，OLTP システムのトランザクションが処理できなくなるため，HiRDB が正常終了又は計画停止できないことがあります。

注※2

HiRDB を計画停止する場合，シングルサーバ又はフロントエンドサーバで動作中のすべてのトランザクションが終了した後に，HiRDB サーバプロセスが保有していたシステムリソースを解放します。こ

のため，ユニット内のすべてのトランザクションが終了した後に，この解放処理で約 1 分程度費やすことがあります。

注※３

ユティリティ又はトランザクションが実行中のため，pdstop -P コマンド入力後 15 分たっても HiRDB が計画停止しない場合，**KFPS05072-W** メッセージが出力されます。そして，pdstop -P コマンドはリターンコード 4 で終了します。ただし，計画停止処理は続行され，ユティリティ又はトランザクションが終了したら計画停止します。

注※４

処理中のトランザクションは，再開始時にロールバックの対象となります。ただし，次に示す場合のトランザクションはロールバックの対象となりません。

- データベース作成ユティリティ又はデータベース再編成ユティリティをログレスモードで実行している場合
- ログレスモードで UAP を実行している場合

したがって，HiRDB を再開始した後に，HiRDB 管理者が RD エリアをバックアップから回復するか，又はユティリティを再実行する必要があります。このときの RD エリアの回復方法については，「データベースをバックアップ取得時点に回復する方法」を参照してください。

1.2.2　pdstop コマンドを実行するサーバマシン

HiRDB/シングルサーバを終了する場合は，シングルサーバを定義したサーバマシンで pdstop コマンドを実行してください。

HiRDB/パラレルサーバを終了する場合は，システムマネジャを定義したサーバマシンで pdstop コマンドを実行してください。

1.2.3　例題（HiRDB を正常終了する場合）

現在稼働中の HiRDB/シングルサーバを正常終了します。

(1)　pdls コマンドで HiRDB の稼働状況を調べます

```
pdls
```

(2)　pdls コマンドの実行結果が表示されます

```
HOSTNAME(140814)    UNITID  SVID      STATUS   STARTTIME
k95x620             unt1    ********  ACTIVE   140812
k95x620             unt1    sds01     ACTIVE   140812
```

〔説明〕

STATUS に ACTIVE が表示されているので，HiRDB が稼働中です。

(3) pdls コマンドで HiRDB に接続中のユーザがいるかどうかを調べます

```
pdls -d prc
```

(4) pdls コマンドの実行結果が表示されます

```
HOSTNAME : k95x620(141028)
STATUS   PID   UID   GID   SVID   TIME      PROGRAM    C-PID

  L      205   0     0     sds01  999999
  L      201   0     0     sds01  999999
  L      194   0     0     sds01  999999
  L      206   0     0     sds01  999999
  L      198   0     0     sds01  999999
  L      142   0     0     sds01  999999
  L      209   0     0     sds01  999999
  L      212   0     0     sds01  999999
```

〔説明〕

- HiRDB に接続中のユーザがいると，PROGRAM に UAP の識別名称が表示されます。したがって，現在 HiRDB に接続中のユーザはいません。HiRDB に接続中のユーザがいると，HiRDB を終了できません。
- 例えば，HiRDB SQL Executer を使用したままにしていると（HiRDB に接続したままにしていると），HiRDB を終了できません。HiRDB を終了するには，HiRDB SQL Executer を終了してください。
- PROGRAM には，クライアント環境定義の PDCLTAPNAME オペランドに指定した UAP の識別名称が表示されます。PDCLTAPNAME オペランドを省略すると，Unknown が表示されます。

(5) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(6) pdstop コマンドの実行結果が表示されます

```
129 14:15:20 unt1 _rdm     KFPS01841-I HiRDB unit unt1 terminated.
 mode = NORMAL
129 14:15:20 unt1 _rdm     KFPS01850-I HiRDB system terminated.
 mode = NORMAL
```

1.2.4　Windows のサービスで HiRDB を終了する方法

この終了方法は，HiRDB/シングルサーバでだけ使用できます。また，正常終了しかできません。

この方法で終了させた場合，Windows のサービスも含めて終了するため，HiRDB が稼働中でないときに使用できる HiRDB 運用コマンドを含め，すべての HiRDB 運用コマンド，ユティリティが使用できなくなります。

また，HiRDB を前提 PP とした関連製品も使用できなくなります。

Windows のサービスで HiRDB を終了する手順を次に示します。

〈手順〉

1. **［コントロールパネル］**の**［サービス］**をダブルクリックします。
2. **［サービス］**リストボックスから HiRDB/SingleServer をクリックします。
3. **［停止］**ボタンをクリックします。

1.2.5　停止処理完了の確認

次の手順で HiRDB（又はユニット）の停止処理が完了したかを確認してください。

- HiRDB/シングルサーバの場合
  pdstop コマンドの終了ステータスが 0 であることを確認してください。
- HiRDB/パラレルサーバの場合
  pdstop コマンドの終了ステータスが 0 であることを確認してください。又は pdstop コマンドを実行して HiRDB システムを停止した場合，システムマネジャユニットがあるサーバマシンのイベントログに KFPS01850-I メッセージが出力されていることを確認してください。

pdstop コマンド終了後，HiRDB が使用する共有リソース（ディスクや IP アドレス）を無効化する場合は，各ユニットで pdls -d ust コマンドを実行します。終了ステータスが 12 であることを確認した後に，共有リソースを無効化してください。

なお，pdstart コマンドの終了ステータスが 0 以外，又は KFPS05210-I メッセージが出力されていない場合，HiRDB の開始処理でエラーが発生しているおそれがあるため，次に示すどれかの方法で，状態を確認してください。

(1)　HiRDB の停止処理の状態を確認する

(a)　pdstop コマンドの終了ステータスで確認する

- pdstop コマンドの終了ステータスが 4 の場合

ユティリティ又はトランザクションが実行中のため，pdstop コマンドが異常終了しています。pdstop コマンドは異常終了しましたが，計画停止処理は継続しています。ユティリティ又はトランザクションの終了を待ち合わせるか，キャンセルしてください。

- pdstop コマンドの終了ステータスが 8 の場合

  HiRDB の停止処理中に障害が発生しています。イベントログに出力されたメッセージを参照して障害原因を取り除いた後，HiRDB が停止処理中の場合は HiRDB システム又は停止中の HiRDB ユニットを強制終了（pdstop -f コマンド，又は pdstop -z コマンドで終了）してください。

(2) ユニットの停止処理の状態を確認する

pdstop コマンド実行後，各ユニットで pdls -d ust コマンドを実行してください。pdls -d ust コマンドの終了ステータスによって次の表に示すように操作が異なります。

表 1–7 pdls -d ust コマンドの終了ステータスとそのときの操作（HiRDB/シングルサーバの場合）

pdls -d ust コマンドの終了ステータス	HiRDB 管理者の処置
0	HiRDB の停止処理が行われていない可能性があります。イベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，pdstop コマンドを再度実行してください。
1	pdls -d ust コマンド実行中に障害が発生しています。出力されたメッセージを参照して障害原因を取り除いた後に，再度 pdls -d ust コマンドを実行してください。
4	HiRDB が停止処理中の可能性があるため，5 秒程度の間隔で一定期間の間，pdls -d ust コマンドを繰り返し実行してください。一定期間経過しても状態が変わらない場合は，イベントログに出力されたメッセージを参照して障害原因を取り除いてください。障害原因を取り除いた後，HiRDB が停止処理中の場合は HiRDB を強制終了（pdstop -f コマンドで終了）してください。
8	HiRDB の停止処理中に障害が発生して HiRDB が異常終了しました。イベントログに出力されたメッセージを参照して障害原因を取り除いてください。
12	HiRDB は停止状態のため，対応不要です。
16	実行する HiRDB の HiRDB サービスが停止している可能性があります。HiRDB サービスが停止している場合はサービスを起動した上で，pdls -d ust コマンドを再度実行してください。

表 1–8 pdls -d ust コマンドの終了ステータスとそのときの操作（HiRDB/パラレルサーバの場合）

pdls -d ust コマンドの終了ステータス	HiRDB 管理者の処置
0	HiRDB ユニットの停止処理が行われていない可能性があります。イベントログに出力されたメッセージを参照して障害原因を取り除いてください。 障害原因を取り除いた後，pdstop コマンドを再度実行してください。
1	pdls -d ust コマンド実行中に障害が発生しています。出力されたメッセージを参照して障害原因を取り除いた後に，再度 pdls -d ust コマンドを実行してください。

pdls -d ust コマンドの終了ステータス	HiRDB 管理者の処置
4	HiRDB ユニットが停止処理中の可能性があるため，5 秒程度の間隔で一定期間の間，pdls -d ust コマンドを繰り返し実行してください。一定期間経過しても状態が変わらない場合は，イベントログに出力されたメッセージを参照して障害原因を取り除いてください。障害原因を取り除いた後，HiRDB ユニットが停止処理中の場合は HiRDB システム，又は HiRDB ユニットを強制終了（pdstop -f 又は pdstop -z コマンドで終了）してください。
8	HiRDB ユニットの停止処理中に障害が発生して HiRDB ユニットが異常終了しました。イベントログに出力されたメッセージを参照して障害原因を取り除いてください。
12	HiRDB ユニットは停止状態のため，対応不要です。
16	実行する HiRDB ユニットの HiRDB サービスが停止している可能性があります。HiRDB サービスが停止している場合はサービスを起動した上で，pdls -d ust コマンドを再度実行してください。

1.3　特別な開始方法

正常開始，再開始，強制開始のほかに，次に示す特別な開始方法があります。

- データベースを再度初期設定するときの開始方法
- マスタディレクトリ用 RD エリアに障害が発生したときの開始方法
- データディクショナリ用 RD エリアに障害が発生したときの開始方法

1.3.1　データベースを再度初期設定するときの開始方法（pdstart -i）

データベースを再度初期設定する場合（データベース初期設定ユティリティを再実行する場合），**pdstart -i** コマンドで HiRDB を開始してください。

●プラグインを使用している場合

プラグインを使用している場合，pdstart -i コマンドを実行する前にシステム共通定義の pdplugin オペランドを削除してください。削除しないと，レジストリに関する情報が失われます。したがって，データベース初期設定ユティリティの実行後，レジストリに関する環境設定を再度する必要が生じます。

1.3.2　マスタディレクトリ用 RD エリアに障害が発生したときの開始方法（pdstart -r）

マスタディレクトリ用 RD エリアに障害が発生すると，通常の開始モードでは HiRDB を開始できません。この場合，**pdstart -r** コマンドで HiRDB を開始してください。

pdstart -r コマンドで HiRDB を開始した場合，次に示すユティリティ及びコマンドだけを実行できます。そのほかの操作はできません。

- データベース回復ユティリティ（pdrstr）
- データベース複写ユティリティ（pdcopy）
- pdls コマンド
- pdstop コマンド
- HiRDB の稼働状態に関係なく実行できるユティリティ及びコマンド

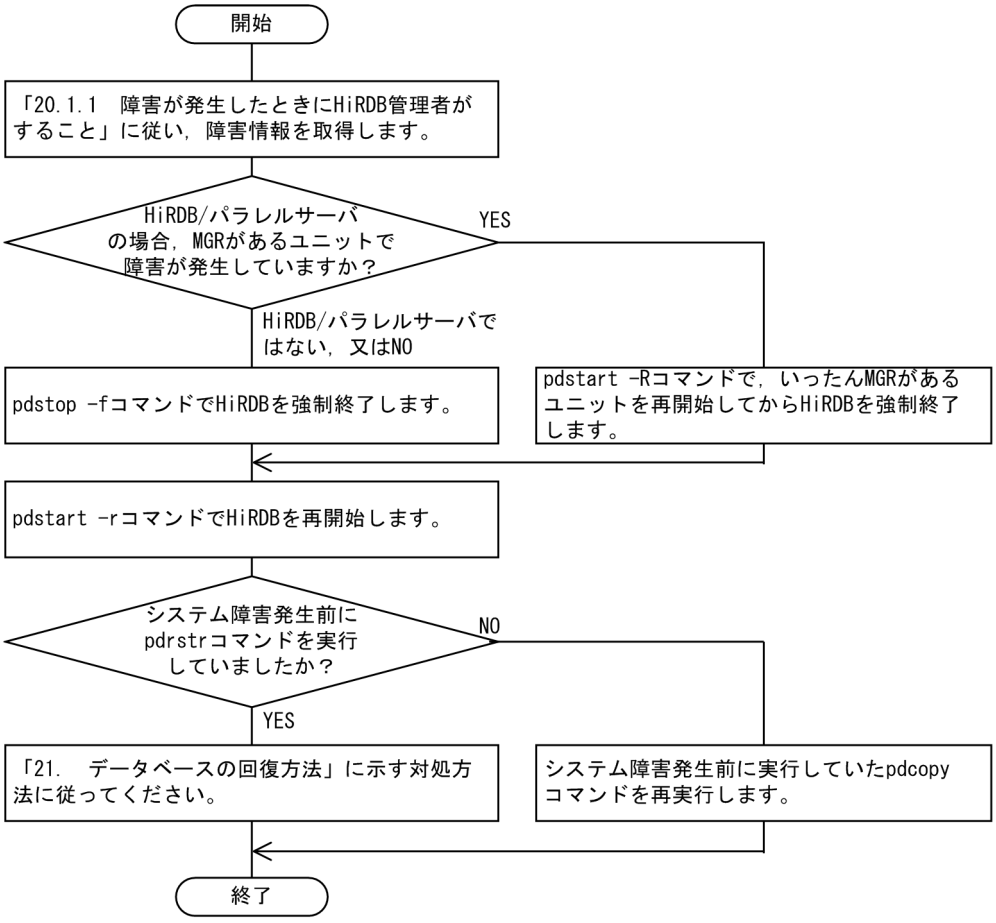
HiRDB 管理者は，これらのユティリティを使用して，マスタディレクトリ用 RD エリアを回復してください。その後，HiRDB を再開始して，業務を続行してください。

マスタディレクトリ用 RD エリアに障害が発生したため，HiRDB を再開始できない場合の対処方法については，「HiRDB が再開始できないときの対処方法」を参照してください。

pdstart -r コマンドで開始した場合に障害が発生したときの対処

pdstart -r コマンドによる HiRDB の開始は，データベースの回復を目的とするため，システム障害，及びシステムマネジャがあるユニットとシステムマネジャがないユニット間の通信障害を認識できません。このため，pdstart -r コマンドで開始した HiRDB で障害が発生した場合，実行中だったデータベース回復ユティリティ（pdrstr），又はデータベース複写ユティリティ（pdcopy）が終了しないことがあります。

この場合，次に示す手順で対処してください。



〔説明〕

- HiRDB を一度強制終了してから再開始します。HiRDB/パラレルサーバの場合，システムマネジャがあるユニットに障害が発生すると，HiRDB を終了できません。そのため，pdstart -R コマンドを実行します。このコマンドを実行すると，システムマネジャがあるユニットをいったん強制的に開始し，その後 HiRDB/パラレルサーバを強制終了します。
- HiRDB の再開始後，ユティリティを再実行してください。データベースの回復を行う場合，データベース回復ユティリティ（pdrstr）の再実行だけでは回復できないことがあります。「データベースの回復方法」を参照し，必要な操作を確認してください。

1.3.3　データディクショナリ用 RD エリアに障害が発生して，フロントエンドサーバが SUSPEND 状態のときの開始方法（pdstart -a）

HiRDB の開始時にデータディクショナリ用 RD エリアに障害が発生すると，フロントエンドサーバが SUSPEND 状態※となります。HiRDB 管理者は，データディクショナリ用 RD エリアを回復した後に，フロントエンドサーバの SUSPEND 状態を解除する必要があります。

このとき，**pdstart -a** コマンドで HiRDB を開始します。そうすれば，フロントエンドサーバの SUSPEND 状態を解除できます。

注※　データディクショナリ用 RD エリアの回復待ちで WAIT している状態のことです。

1.4　ユニット単位の開始及び終了方法（HiRDB/パラレルサーバ限定）

実行者　HiRDB 管理者

HiRDB/パラレルサーバの場合，ユニット単位で開始及び終了操作ができます。

1.4.1　ユニット単位の開始方法

次の表に示すコマンドを実行すると，HiRDB をユニット単位に開始できます。

表 1–9　ユニット単位の開始方法

開始モード	実行コマンド	開始モードの説明	前回の終了モード
正常開始	**pdstart -u** **pdstart -x**	HiRDB の稼働中に，特定のユニットを正常終了させて，再度そのユニットを開始するときの方法です。正常開始では，前回稼働時の情報を引き継ぎません。 ただし，次に示す情報は引き継ぎます。 • 障害閉塞している RD エリアの状態	正常終了
再開始※1		前回の終了モードが右記の場合，HiRDB は（自動的に）再開始となります。再開始では，前回稼働時の情報を引き継ぎます。再開始時に引き継ぐ情報の詳細は，「HiRDB が再開始するときに引き継ぐ情報」を参照してください。	強制終了 異常終了
強制開始※2	**pdstart -u dbdestroy** **pdstart -x dbdestroy**	通常は使用しないでください。ユニットを再開始できないときに使用します。データベースの回復処理をしないで，ユニットを強制的に開始します。	なし

注※1

再開始をする前に，必ず「計画停止，強制終了，又は異常終了したときの注意事項」を参照してください。

注※2

強制開始をする前に，必ず「HiRDB（ユニット）を強制開始するときの注意事項」を参照してください。

1.4.2　ユニット単位の終了方法

次の表に示すコマンドを実行すると，HiRDB をユニット単位に終了できます。

表 1–10　ユニット単位の終了方法

終了モード	入力コマンド	終了モードの説明
正常終了	**pdstop -u** **pdstop -x**	このユニットに対する CONNECT 要求を禁止し，このユニットに接続中のすべての UAP を切り離した後にユニットを終了します。 ユティリティが実行中のため，pdstop -u 又は pdstop -x コマンドを実行してもユニットを終了できない場合は，KFPS05070-E メッセージが出力されます。pdstop -u 又は pdstop -x コマンドはリターンコード 8 で終了します。
強制終了※1	**pdstop -f -u** **pdstop -f -x**	このユニットで処理中のトランザクションの完了を待たずに，ユニットを直ちに終了します。処理中のトランザクションは，再開始時にロールバックの対象※2となります。
異常終了	−	何らかの異常によってユニットが終了する場合の終了モードです。このユニットで処理中のトランザクションの完了を待たずに，ユニットは直ちに終了します。処理中のトランザクションは，再開始時にロールバックの対象※2となります。

（凡例）−：該当しません。

注※1

強制終了する前に，必ず「計画停止，強制終了，又は異常終了したときの注意事項」を参照してください。

注※2

処理中のトランザクションは，再開始時にロールバックの対象となります。ただし，次に示す場合のトランザクションはロールバックの対象となりません。

- **データベース作成ユティリティ又はデータベース再編成ユティリティをログレスモードで実行している場合**
- **ログレスモードで UAP を実行している場合**

したがって，HiRDB を再開始した後に，HiRDB 管理者が RD エリアをバックアップから回復するか，又はユティリティを再実行する必要があります。この場合の RD エリアの回復方法については，「データベースをバックアップ取得時点に回復する方法」を参照してください。

1.4.3　ユニット単位で開始及び終了するときの運用

ユニット単位で開始及び終了するときの運用手順を次に示します。

〈手順〉

1. **pdstart** コマンドで HiRDB を開始して運用します。

   pdstart

2. ユニット内で障害が発生したため，**pdstop -u** コマンドでこのユニットを正常終了します。又はユニットが異常終了しました。

pdstop -u ユニット識別子

3. 障害の原因を取り除いた後，**pdstart -u** コマンドでこのユニットを再開始します。

   pdstart -u ユニット識別子

1.5　サーバ単位の開始及び終了方法（HiRDB/パラレルサーバ限定）

実行者　HiRDB 管理者

HiRDB/パラレルサーバの場合，サーバ単位で開始及び終了操作ができます。ここでは，サーバ単位の開始及び終了方法について説明します。サーバ単位の開始及び終了の対象となるサーバを次に示します。

- フロントエンドサーバ
- ディクショナリサーバ
- バックエンドサーバ

1.5.1　サーバ単位の開始方法

次の表に示すコマンドを実行すると，HiRDB をサーバ単位に開始できます。

表 1–11　サーバ単位の開始方法

開始モード	実行コマンド	開始モードの説明	前回の終了モード
正常開始	**pdstart -s**	HiRDB の稼働中に，特定のサーバを正常終了させて，再度そのサーバを開始するときの方法です。正常開始では，前回稼働時の情報を引き継ぎません。 ただし，次に示す情報は引き継ぎます。 • 障害閉塞している RD エリアの状態	正常終了

1.5.2　サーバ単位の終了方法

次の表に示すコマンドを実行すると，HiRDB をサーバ単位に終了できます。

表 1–12　サーバ単位の終了方法

終了モード	入力コマンド	終了モードの説明
正常終了	**pdstop -s**	このサーバに対するトランザクションの受け付けを禁止し，このサーバで処理中のトランザクションを完了した後に，サーバを終了します。

1.5.3　サーバ単位で開始及び終了するときの運用

サーバ単位で開始及び終了するときの運用手順を次に示します。

〈手順〉

1. **pdstart** コマンドで HiRDB を開始して運用します。

pdstart

2. サーバ内で障害が発生したため，**pdstop -s** コマンドでこのサーバを正常終了します。
   pdstop -s サーバ名
3. 障害の原因を取り除いた後，**pdstart -s** コマンドでこのサーバを正常開始します。
   pdstart -s サーバ名

1.6　開始時の注意事項

ここでは，HiRDB を開始するときの注意事項について説明します。説明する項目は次のとおりです。

- HiRDB 開始時の注意事項
- HiRDB（ユニット）を強制開始するときの注意事項
- HiRDB の開始処理に失敗したときの注意事項（HiRDB/パラレルサーバ限定）

1.6.1　HiRDB 開始時の注意事項

(1)　%PDDIR%¥tmp にアクセスしたままの状態だと HiRDB を開始できないことがあります

エクスプローラなど，ほかのアプリケーションで%PDDIR%¥tmp にアクセスしたままの状態で HiRDB を開始しないでください。HiRDB 開始時に作業ディレクトリである%PDDIR%¥tmp を HiRDB が削除するため，ほかのアプリケーションでアクセスしていると排他エラーとなり，HiRDB が開始できないことがあります。

(2)　HiRDB/パラレルサーバ開始時の注意事項

HiRDB/パラレルサーバのインストール直後は，サービスのアカウントがシステムアカウントに設定されているため，次に示す手順でユーザアカウントに変更してください。システムアカウントの設定で起動した場合，KFPH22011-E（システムコールエラー）で HiRDB が異常終了します。

〈手順〉

1. **［コントロールパネル］**の**［サービス］**をダブルクリックします。
2. **［サービス］**リストボックスから HiRDB/ParallelServer を選択し，**［停止］**をクリックします。
3. HiRDB が開始状態でないことを確認します。
4. **［サービス］**リストボックスから HiRDB/ParallelServer を選択し，**［スタートアップ］**ボタンをクリックします。
5. **［ログオン］**がシステムアカウントに設定されているため，アカウントを選択します。
6. アカウントの右にあるボタンをクリックし，**［ユーザの追加］**を開きます。
7. 名前の一覧から HiRDB 管理者を選択し，**［追加］**をクリックします。
8. **［OK］**をクリックします。
9. **［パスワード］**ボックスに HiRDB 管理者のパスワードを入力します。同じパスワードを**［パスワードの確認入力］**ボックスにも入力します。
10. **［OK］**をクリックします。

11. **[サービス]** リストボックスから HiRDB/ParallelServer を選択し，**[開始]** をクリックします。

(3) HiRDB 開始処理中の注意事項

プロセスの起動とサーバマシンのログオフ処理が重なると，プロセスが初期化エラーになることがあります。この場合，KFPS01820-E メッセージ（end state の値は 8000）が出力されます。プロセスによって，初期化エラーが発生したときの現象は異なります。例えば，ユニットコントローラのプロセスや，後処理プロセスで初期化エラーが発生すると，HiRDB はユニットダウンします。

プロセスは，主に HiRDB の開始処理時に頻繁に生成されますが，稼働中や終了処理時に生成されるものもあります。このため，初期化エラーを回避するには，HiRDB の稼働中はサーバマシンのログオフ操作を控えるのが最適です。これが難しい場合は，プロセスが頻繁に生成される開始処理時などに限り，ログオフ操作を控えてください。

1.6.2 HiRDB（ユニット）を強制開始するときの注意事項

HiRDB（ユニット）を強制開始すると，前回稼働時の情報を引き継ぎません。このため，HiRDB はデータベースを前回稼働時の状態に回復できません。したがって，HiRDB 管理者がデータベースの状態を回復する必要があります。

HiRDB を強制開始すると，前回稼働時に更新したすべての RD エリア（システム用 RD エリアも含みます）が破壊されます。

したがって，強制開始後に，破壊された RD エリアをデータベース回復ユティリティで回復する必要があります。RD エリアを回復しないと，その後の HiRDB の動作を保証できません。破壊された RD エリアを回復する手順を次に示します。

〈手順〉

1. HiRDB の再開始に失敗したときに出力された **KFPS01262-I** メッセージを参照してください。
2. KFPS01262-I メッセージに表示されているシステムログ読み込み開始のファイルグループ名，及びそれ以降に発生したシステムログ※を調べます。
3. 2．で調べたシステムログをデータベース回復ユティリティ（**pdrstr** コマンド）の入力情報にして，RD エリアを回復してください。RD エリアはシステムログだけで回復できます。

注※ 「それ以降に発生したシステムログ」は，**pdlogls -d sys** コマンドで調べられます。

```
pdlogls -d sys

Group    Type Server   Gen No.  Status  Run ID    Block No.
logfg01  sys  sds             1 cnu---u 35108db8        1        2
logfg02  sys  sds             4 cn---cu 3510ecba       10        0
logfg03  sys  sds             1 cn----- 3510ecba        1        3
logfg04  sys  sds             2 cn----- 3510ecba        4        8
logfg05  sys  sds             3 cn----- 3510ecba        9        f
logfg06  sys  sds             0 cn----- 00000000        0        0
```

〔説明〕

例えば，KFPS01262-I メッセージの fg 名が"logfg04"，ブロック番号が"5"だった場合，Run ID，Gen No.と Block No.からシステムログファイルの使用順序が次の順序であることが分かります。

logfg04 → logfg05 → logfg02

つまり，KFPS01262-I メッセージに出力されたファイルグループの Run ID と同じで，かつ Gen No より大きいものを使用すればよいということです。ただし，HiRDB/パラレルサーバでサーバ単位の開始及び終了をしていると，この方法では求められません。

1.6.3 HiRDB の開始処理に失敗したときの注意事項（HiRDB/パラレルサーバ限定）

ここでは，ユニットの開始処理に失敗したときの注意事項について説明します。

(1) システムログファイルの処置

前提条件

2.と 3.の条件については，どちらかを満たしていればよいです。

1. 開始処理に失敗したユニットに複数のサーバを定義しています。
2. 開始モードが，正常開始又は計画停止後の再開始です。
3. 開始モードが強制終了又は異常終了後の再開始で，かつ pd_log_rerun_swap=Y を指定しています。

現象

ユニットを開始するとき，HiRDB はユニット内のサーバを並列に起動しています。このとき，一部のサーバで HiRDB システム定義のオペランド不正，又は何らかの障害が発生すると，そのユニット全体の開始に失敗します。このとき，次に示す現象が起きることがあります。

- **開始処理が完了したサーバでは，現用のシステムログファイルの割り当てが完了しています（KFPS01221-I メッセージが出力されます）。**
- **開始処理が完了していないサーバでは，現用のシステムログファイルが割り当てられていない状態になっています（KFPS01221-I メッセージが出力されません）。**

対策

このとき，HiRDB は次のユニット開始時に，新たな現用ファイルを割り当てます。したがって，前回のユニット開始時に割り当てられた現用ファイルをクローズ状態にします（再使用しません）。このクローズ状態になったシステムログファイルを，再度現用ファイルとして割り当てるために，HiRDB 管理者は(a)，又は(b)の，どちらかの方法で処置をしてください。

(a) システムログファイルを再初期化しない方法

該当するシステムログファイルに **pdlogunld** 又は **pdlogchg** コマンドを実行した後に，HiRDB の稼働状況に応じて次の処置をします。

HiRDB の稼働状況	HiRDB 管理者の処置
HiRDB が稼働中の場合	該当するシステムログファイルに pdlogopen コマンドを実行します。
ユニットを正常開始する前の場合	処置をする必要はありません。ユニット開始時に現用として割り当てできるシステムログファイルとして認識されます。
ユニットを再開始する前の場合	ユニットを再開始した後に，該当するシステムログファイルに pdlogopen コマンドを実行してください。

(b)　システムログファイルを再初期化する方法

〈手順〉

1. 該当するシステムログファイルに，**pdlogunld** 又は **pdlogchg** コマンドを実行します。
   pdlogunld -d sys -s b001 -g syslogfgp03 -o C:¥unld¥unldlog01
2. **pdlogrm** コマンドでシステムログファイルを削除します。
   pdlogrm -d sys -s b001 -f C:¥unt1¥sysfile01¥log01
3. **pdloginit** コマンドで，該当するシステムログファイルを初期化します。
   pdloginit -d sys -s b001 -f C:¥unt1¥sysfile01¥log01 -n 5000

(2)　ディクショナリサーバの開始

現象

次に示す操作後にユニットを開始すると，ディクショナリサーバが RD エリア構成情報を取得します。

- BES の正常停止
- データベース構成変更ユティリティ（pdmod）
  - RD エリアの追加（create rdarea）
  - RD エリアの拡張（expand rdarea）
  - with reconstruction を指定した RD エリアの再初期化（initialize rdarea）
  - RD エリアの属性変更（alter rdarea）
  - RD エリアの移動（move rdarea）

このとき，ディクショナリサーバを開始していないと，最大 60 回 RD エリア構成情報の取得をリトライします。60 回リトライしても情報を取得できないとき，RD エリア構成情報の取得処理を中断し，BES ユニットの開始処理が失敗します。

対策

ディクショナリサーバを開始してから，BES ユニットを開始してください。

1.7　終了時の注意事項

ここでは，HiRDB を終了するときの注意事項について説明します。説明する項目は次のとおりです。

- HiRDB 終了時の注意事項
- 計画停止，強制終了，又は異常終了したときの注意事項

1.7.1　HiRDB 終了時の注意事項

(1)　未決着状態のトランザクションや接続ユーザがいると HiRDB を正常終了できません

現象

HiRDB を正常終了できません。

原因

未決着状態のトランザクションや接続ユーザがいると，HiRDB を正常終了できません。

対策

接続ユーザやトランザクションの状態を確認してから，HiRDB を正常終了させてください。

- pdls -d prc コマンドで接続ユーザを確認できます。
- pdls -d trn コマンドでトランザクションの状態を確認できます。

接続ユーザがいないのに，未決着状態のトランザクションがある場合は，次に示すコマンドで決着してください。

- pdcmt コマンドでトランザクションをコミットできます。
- pdrbk コマンドでトランザクションをロールバックできます。

未決着状態のトランザクションや接続ユーザがいるときに HiRDB を正常終了させる方法については，「未決着状態のトランザクションがあるときの対処方法」を参照してください。

(2)　Windows をシャットダウンするときは HiRDB の状態を確認してください

Windows をシャットダウンする場合は，HiRDB が終了しているかどうかを確認してください。HiRDB を終了する前に Windows をシャットダウンすると HiRDB は強制終了するため，次回 OS ブート後の HiRDB の開始モードは再開始になります。HiRDB は再開始時にシステムの状態を Windows シャットダウン時点に回復するため，再開始に掛かる時間が長くなることがあります。HiRDB の再開始時の注意事項については，「計画停止，強制終了，又は異常終了したときの注意事項」を参照ください。

参考

HiRDB が終了しているかどうかは，pdls -d ust コマンドで確認できます。

1.7.2　計画停止，強制終了，又は異常終了したときの注意事項

HiRDB が計画停止，強制終了，又は異常終了したときは，次に示すことに注意してください。

(1)　HiRDB のバージョンは変更はできません

HiRDB が計画停止，強制終了，又は異常終了したときに，HiRDB のバージョンは変更できません。変更すると HiRDB の再開始に失敗します。HiRDB のバージョンを変更する場合は，必ず HiRDB を正常終了させてください。

(2)　HiRDB システム定義の変更は一部だけできます

HiRDB が計画停止，強制終了，又は異常終了したときに，HiRDB システム定義のオペランドで変更できないものがあります。変更できないオペランドについては，マニュアル「HiRDB システム定義」を参照してください。

●HiRDB/パラレルサーバの場合

HiRDB の正常終了又は計画停止時に一部のユニットが異常終了した場合，次回の HiRDB 開始前に HiRDB システム定義を変更しないでください。変更すると，HiRDB の開始に失敗します。失敗しなくても，開始後に正常に稼働できなくなります。

(3)　サーバ構成の変更はできません（HiRDB/パラレルサーバ限定）

HiRDB が計画停止，強制終了，又は異常終了したときに，HiRDB のサーバ構成を変更しないでください。再開始前にはシステム共通定義の pdstart オペランドと pdunit オペランドは変更できません。変更すると，HiRDB の再開始に失敗します。

HiRDB のサーバ構成を変更する場合は，必ず HiRDB を正常終了させてください。正常終了後であれば，HiRDB のサーバの構成を変更できます。その場合は，次に示すファイルを再初期化してください。再初期化をしないと，HiRDB の開始に失敗することがあります。

- すべてのシステムログファイル
- すべてのシンクポイントダンプファイル
- すべてのユニット用ステータスファイル
- すべてのサーバ用ステータスファイル

(4)　グローバルバッファの追加，削除，変更について

HiRDB が強制終了又は異常終了したときには，次の表に示すグローバルバッファの操作ができません。

表 1–13　強制終了，異常終了したときにできないグローバルバッファの操作

グローバルバッファの操作	正常終了，計画停止	強制終了，異常終了
グローバルバッファの追加 （pdbuffer オペランドの追加）	○	×
グローバルバッファの削除 （pdbuffer オペランドの削除）	○	×
グローバルバッファの変更 （pdbuffer オペランドの変更）	○	×

（凡例）

○：操作できます。

×：操作できません。

(5)　HiRDB 稼働中に追加した RD エリアに割り当てたグローバルバッファについて

HiRDB 稼働中に追加した RD エリアに割り当てたグローバルバッファは，終了モードによって，次回開始時に引き継がれるかどうかが決まります。詳細を次の表に示します。

表 1–14　HiRDB 稼働中に追加した RD エリアに割り当てたグローバルバッファを引き継ぐかどうか

条件	正常終了，計画停止	強制終了，異常終了
HiRDB 稼働中に追加した RD エリアに割り当てたグローバルバッファを引き継ぐかどうか	引き継ぎません	引き継ぎます

(6)　ステータスファイルの削除，変更，初期化はできません

HiRDB が計画停止，強制終了，又は異常終了したときには，次の表に示すステータスファイルの操作ができません。

表 1–15　計画停止，強制終了，異常終了したときにできないステータスファイルの操作

ステータスファイルの操作	正常終了	計画停止，強制終了，異常終了
ステータスファイルの追加	○	○
ステータスファイルの削除	○	×
ステータスファイルの変更	○	×
ステータスファイルの初期化	○	×

（凡例）

○：操作できます。

×：操作できません。実行すると再開始できなくなります。

●計画停止，強制停止，異常終了後には

次に示すオペランドの追加はできます。

- pd_syssts_file_name_1～7
- pd_sts_file_name_1～7

しかし，これらのオペランドの削除，指定値の変更はできません。削除又は変更した場合，該当ユニットの再開始は失敗します。

(7)　シンクポイントダンプファイルの追加，削除，変更，初期化はできません

HiRDB が計画停止，強制終了，異常終了したときには，次の表に示すシンクポイントダンプファイルの操作ができません。

表 1–16　計画停止，強制終了，異常終了したときにできないシンクポイントダンプファイルの操作

シンクポイントダンプファイルの操作	正常終了	計画停止，強制終了，異常終了
シンクポイントダンプファイルの追加	○	×
シンクポイントダンプファイルの削除	○	×
シンクポイントダンプファイルの変更	○	×
シンクポイントダンプファイルの初期化	○	×

（凡例）

○：操作できます。

×：操作できません。実行すると再開始できなくなります。

●計画停止，強制停止，異常終了後には

次に示すオペランドの追加，削除，指定値の変更はできません。

- pdlogadfg -d spd
- pdlogadpf -d spd

誤って追加，削除，指定値を変更した場合，該当ユニットの再開始は失敗します。

なお，前回の HiRDB 稼働中に追加したシンクポイントダンプファイルは再開始後の HiRDB にも引き継がれます。

(8)　システムログファイルの削除，変更はできません

HiRDB が計画停止，強制終了，異常終了したときには，次の表に示すシステムログファイルの操作ができません。

表 1–17　計画停止，強制終了，異常終了したときにできないシステムログファイルの操作

システムログファイルの操作	正常終了	計画停止，強制終了，異常終了
システムログファイルの追加	○	○
システムログファイルの削除	○	×
システムログファイルの変更	○	×
システムログファイルの初期化	○	○※

（凡例）

○：操作できます。

×：操作できません。実行すると再開始できなくなります。

注※

次に示すシステムログファイルは初期化しないでください。誤って初期化すると，該当ユニットの再開始が失敗するか，再開始に成功してもデータベースの内容が不正になります。

- 前回の HiRDB 稼働中，最後に使用したシステムログファイル
- 上書き禁止状態のシステムログファイル

●計画停止，強制停止，異常終了後には

次に示すオペランドの追加はできます。

- pdlogadfg -d sys
- pdlogadpf -d sys

しかし，これらのオペランドの削除，指定値の変更はできません。削除又は変更した場合，該当ユニットの再開始は失敗します。

(9)　現用のシステムログファイルのスワップについて

HiRDB が再開始するときに，現用のシステムログファイルがスワップするかどうかは，pd_log_rerun_swap オペランドの指定によります。現用のシステムログファイルがスワップする条件を次の表に示します。

表 1–18　現用のシステムログファイルがスワップする条件

pd_log_rerun_swap オペランドの指定	正常終了，計画停止	強制終了，異常終了
pd_log_rerun_swap ＝ Y	○	○
pd_log_rerun_swap ＝ N	○	×
pd_log_rerun_swap オペランドを省略	○	×

（凡例）

○：

HiRDB の再開始時に現用のシステムログファイルをスワップします。前回終了時の現用ファイルをスワップさせ，新たな現用ファイルを割り当てます。

×：

HiRDB の再開始時に現用のシステムログファイルをスワップしません。前回終了時の現用ファイルをそのまま現用として使用します。ただし，次に示す場合は，HiRDB の再開始時に現用のシステムログファイルをスワップします。

- 前回終了時の現用ファイルに障害が発生した場合
- HiRDB の停止中にアンロードをした（pdlogunld 又は pdlogchg コマンドを実行した）場合

(10) 再開始するときのシステムログファイルの入力障害について

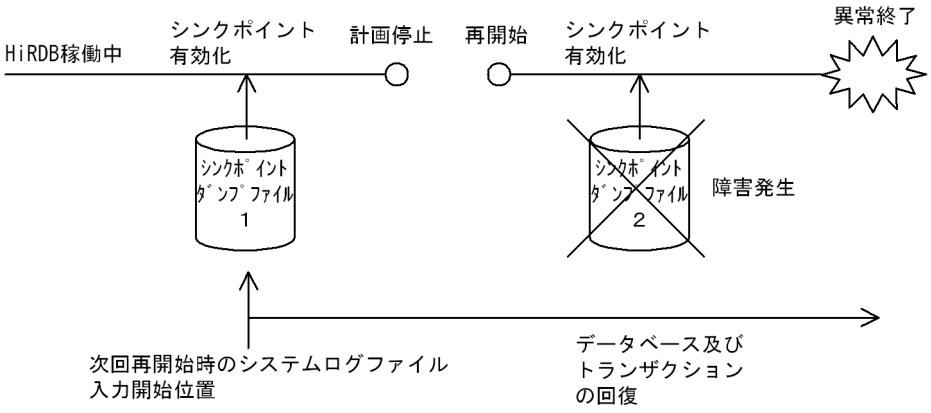
HiRDB を再開始するとき，HiRDB は前回稼働中の最後に有効化したシンクポイント時点をシステムログの入力開始時点とします。そして，最新のシステムログまでを順次入力してデータベース及びトランザクションを回復します。このとき，入力対象となったシステムログファイルのうち，一つ以上のシステムログファイルを障害によって失うと（システムログを二重化している場合は A，B の両系とも失うと），再開始に失敗するか，又は再開始に成功してもデータベースの内容が不正になってしまいます。

この場合，失われたシステムログファイルのアンロードログファイルを取得していれば，バックアップ及びアンロードログファイルを入力情報にしてデータベース回復ユティリティでデータベースを回復できます。しかし，アンロードログファイルがない場合には，システムログを使用した回復はできません。データベースのバックアップ取得時点の状態に回復し，その後の業務を再実行して回復するか，又はデータベースを再作成する必要があります。

(11) 計画停止後の再開始での注意事項

HiRDB を再開始するとき，前回の HiRDB 稼働中の最後に有効化したシンクポイント時点をシステムログの入力開始位置とします。この最後に有効化したシンクポイントダンプファイル，及びステータスファイルに障害が発生すると，HiRDB 再開始時のシステムログの入力開始位置がもう一つ前に有効化したシンクポイント時点まで戻ってしまうことがあります。システムログ入力開始位置が戻ってしまう例を次の図に示します。

図 1–1　システムログ入力開始位置が戻ってしまう例



〔説明〕

このようにして決定したシステムログ入力開始位置から，最新のシステムログファイルまでの間に計画停止をした場合には，次の点に注意してください。

- **計画停止後の再開始前に，HiRDB システム定義を変更すると，再開始に失敗します。この場合，この後に HiRDB システム定義の内容を計画停止前の内容に戻しても，再開始に失敗します。**

このような場合は，ユニットを強制的に正常開始した後に，データベースのバックアップ及びアンロードログファイルからデータベース回復ユティリティを実行して回復する必要があります。

(12)　強制終了するとユニットが異常終了することがあります

HiRDB を強制終了（pdstop -f コマンド）すると，アボートコード Polkcrt で HiRDB のユニットが異常終了することがあります。

これは，強制終了処理でクリティカル状態のプロセスを終了するためです。運用上の問題はありません。この現象を無視してください。強制終了処理では，クリティカル状態のサーバプロセスであっても即時停止します。そのため，この現象が発生することはありますが，次回の pdstart コマンド入力で再開始（システムログからデータベースを回復）するため問題ありません。

1.8　HiRDB の開始処理時間を短縮する方法

1.8.1　開始処理時間を短縮する方法，及び運用例

実行者　HiRDB 管理者

通常，常駐プロセスの起動処理は HiRDB の開始処理中に行われます。このため，常駐プロセス数が多くなると，それに比例して HiRDB の開始処理時間が長くなります。目安として，100MIPS 程度のサーバマシンでプロセスを一つ起動するのに約 1 秒掛かります。

系切り替え機能を使用しているときなど，HiRDB の開始処理時間をなるべく短くしたい場合は，HiRDB の開始処理時間を短縮することを検討してください。

なお，常駐プロセス数は **pd_process_count** オペランドの第 1 パラメタの値によって決定されます。

(1)　開始処理時間を短縮する方法

開始処理時間を短縮するには，pd_process_count オペランドの第 2 パラメタを指定してください。第 2 パラメタの指定有無による処理の違いを次に示します。

(a)　第 2 パラメタを指定しない場合（pd_process_count = 500 と指定した場合）

すべての常駐プロセスの起動処理を HiRDB の開始処理中に行います。この場合，常駐プロセスがすべて起動されないと（この場合は 500 個），HiRDB が開始できません。この場合，100MIPS 程度のサーバマシンで，HiRDB 開始処理時の常駐プロセスの起動に約 500 秒掛かります。

(b)　第 2 パラメタを指定する場合（pd_process_count = 500,50 と指定した場合）

一部の常駐プロセスの起動処理を HiRDB の開始処理中に行い，残りの常駐プロセスの起動処理を HiRDB の開始処理終了後に行います。この場合，一部の常駐プロセスが起動されれば（この場合は 50 個），HiRDB が開始できます。この場合，100MIPS 程度のサーバマシンで，HiRDB 開始処理時の常駐プロセスの起動に約 50 秒掛かります。残りの常駐プロセス（この場合は 450 個の常駐プロセス）は，HiRDB の開始処理終了後に起動されます。

(2)　運用例

運用されているシステムは次のとおりとします。

- 常駐プロセスは 200 個必要とします。
- OpenTP1 の SPP（50 個）については，HiRDB の開始処理終了と同時に処理をします。

●システム定義の指定

pd_process_count = 200,50

〔説明〕

- 第１パラメタには全常駐プロセス数 200 を指定します。
- HiRDB の開始直後に OpenTP1 の SPP 用の常駐プロセスを確保するため，第２パラメタに 50 を指定します。
- HiRDB/パラレルサーバの場合は，フロントエンドサーバ定義の pd_process_count オペランドに指定します。

(3) 注意事項

この機能を使用する場合は，クライアント環境定義の PDCWAITTIME オペランドの値を見直してください。

理由

HiRDB の開始直後に，pd_process_count の第２パラメタの値を超える UAP が実行されると，残りの常駐プロセスが起動した後にトランザクション処理が実行されます。したがって，クライアント環境定義の PDCWAITTIME オペランドの値が小さいと，一部の UAP がタイムアウトで処理できない場合があります。PDCWAITTIME オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

2 機密保護の設定方法

HiRDB ではデータベースの内容破壊，及び機密の漏えいを防ぐために機密保護機能を備えています。この章では，機密保護の設定方法について説明します。

2.1　機密保護とは

データベースを外部の人にアクセスされないように，HiRDB では機密保護機能を用意しています。機密保護機能では**ユーザ権限**という概念を使用していて，必要な権限を持っていないとデータベースにアクセスできないようになっています。

2.1.1　ユーザ権限の種類

ユーザ権限の種類を次の表に示します。

表 2–1　ユーザ権限の種類

ユーザ権限の種類	説明	この権限を持っているとできること	権限付与者		
			H	D	ス
DBA 権限	DBA 権限，CONNECT 権限，及びスキーマ定義権限を与えたり，取り消したりするのに必要な権限です。	• ほかの人に DBA 権限，CONNECT 権限，及びスキーマ定義権限を与えられます。 • 与えた DBA 権限，CONNECT 権限，及びスキーマ定義権限を取り消せます。 • ほかの人のスキーマも定義できます。 • ほかの人のスキーマに対して定義系 SQL 操作ができます。対象となる SQL は表「権限を持つと実行できる定義系 SQL の操作」を参照してください。 • CONNECT 関連セキュリティ機能に関する項目を定義できます。 • HiRDB に接続できます（CONNECT 権限を持っています）。	○	○	×
監査権限	監査人に必要な権限です。セキュリティ監査機能を使用する場合に監査権限を設定します。セキュリティ監査機能については，「セキュリティ監査機能の運用」を参照してください。 監査権限を持つユーザは次に示す権限を持っています。 • CONNECT 権限 • スキーマ定義権限※	表「監査権限を持っていることで操作できること」を参照してください。	○	×	×
CONNECT 権限	HiRDB を利用するために必要な権限です。 CONNECT 権限を持たないユーザが HiRDB を利用しようとするとエラーになります。	データベースに接続（CONNECT）できるようになります。	○	○	×

ユーザ権限の種類	説明	この権限を持っているとできること	権限付与者		
			H	D	ス
スキーマ定義権限	スキーマを定義するために必要な権限です。	自分のスキーマを定義できます。 スキーマを定義すると，自分のスキーマに対する定義系 SQL 操作ができます。対象となる定義系 SQL は表「権限を持つと実行できる定義系 SQL の操作」を参照してください。	○	○	×
スキーマ操作権限	他ユーザが所有しているスキーマに対して，定義系 SQL 操作を行うために必要な権限です。	• スキーマ所有者が，自分のスキーマ操作権限を他ユーザに付与・削除できます。 • スキーマ所有者は，複数のユーザにスキーマ操作権限を付与できます。 • DBA 権限，又は CONNECT 権限のあるユーザにスキーマ操作権限を付与できます。 • ユーザは，1 つのスキーマに対するスキーマ操作権限だけを持つことができます。 • 監査人が所有するスキーマに対するスキーマ操作権限は付与できません。 • 監査人には，スキーマ操作権限を付与できません。 • 対象となる定義系 SQL は表「権限を持つと実行できる定義系 SQL の操作」を参照してください。 留意事項 スキーマ操作権限を使用して他ユーザのスキーマ上に作成したリソースであっても，リソースの所有者は，作成者ではなくスキーマ所有者です。また，アクセス権限もスキーマ所有者にしか与えられません。	×	×	○
RD エリア利用権限	私用 RD エリアを利用するために必要な権限です。 公用 RD エリアに表又はインデクスを作成する場合は，RD エリア利用権限は不要です。	私用 RD エリアに表及びインデクスを作成できます。	○	○	×
アクセス権限	表（実表及びビュー表）をアクセスするために必要な権限です。アクセス権限は表単位に設定し，次に示す４種類があります。また，ロール機能によって必要な権限をロールに集約できます。	ほかの人の表をアクセスできます。	×	×	○
	SELECT 権限	表の検索（SELECT）ができます。	×	×	○
	INSERT 権限	表に行データを追加（INSERT）できます。	×	×	○
	DELETE 権限	表の行データを削除（DELETE）できます。	×	×	○
	UPDATE 権限	表の行データを更新（UPDATE）できます。	×	×	○

ユーザ権限の種類	説明	この権限を持っているとできること	権限付与者		
			H	D	ス
ロール利用権限	ロールを使用するための権限です。ユーザ又はロールに対してロール利用権限を付与できます。	ほかの人の表をアクセスできます。	○	○	×

（凡例）

H：HiRDB 管理者

D：DBA 権限保持者

ス：スキーマ所有者

○：権限を与えられます。

×：権限を与えられません。

注※

副監査人はスキーマ定義権限を持ちません。

表 2–2　監査権限を持っていることで操作できること

操作	主監査人	副監査人
監査証跡表へのアクセス※1	○	○
監査証跡表へのデータロード	○	○
監査証跡表の SELECT 権限の付与及び削除	○	×
監査証跡表の削除	○	×
監査人のパスワード変更※2	○	○
監査対象イベントの定義及び削除	○	○
監査証跡ファイルのスワップ	○	○

（凡例）

○：操作できます。

×：操作できません。

注※1

監査証跡表へのデータの追加及び変更（INSERT 及び UPDATE）はできません。

注※2

監査人が自分のパスワードを変更できます。ほかの監査人のパスワードは変更できません。

表 2–3　権限を持つと実行できる定義系 SQL の操作

定義系 SQL※1	DBA 権限	スキーマ 定義権限 （スキーマ所有者）	スキーマ 操作権限
ALTER INDEX	×	○	○
ALTER PROCEDURE	○	○	×
ALTER ROUTINE	○	○	×
ALTER TABLE	×	○	○
ALTER TRIGGER	○	○	×
COMMENT	×	○	○※2
CREATE 〔PUBLIC〕FUNCTION	×	○	○※2
CREATE INDEX 形式 1～3	×	○	○
CREATE 〔PUBLIC〕 PROCEDURE	×	○	○※2
CREATE ROLE	○	×	×
CREATE SCHEMA	○	○	×
CREATE SEQUENCE	×	○	○
CREATE TABLE	×	○	○
CREATE TRIGGER	×	○	○
CREATE TYPE	×	○	○
CREATE 〔PUBLIC〕VIEW	×	○	○※2
DROP DATA TYPE	○	○	×
DROP 〔PUBLIC〕FUNCTION	○	○	×
DROP INDEX	○	○	×
DROP 〔PUBLIC〕PROCEDURE	○	○	×
DROP ROLE	○	×	×
DROP SCHEMA	○	○	×
DROP SEQUENCE	○	○	×
DROP TABLE	○	○	×
DROP TRIGGER	○	○	×
DROP 〔PUBLIC〕VIEW	○	○	×
GRANT	×	○	×

定義系 SQL※1	DBA 権限	スキーマ 定義権限 （スキーマ所有者）	スキーマ 操作権限
アクセス権限			
REVOKE アクセス権限	×	○	×
GRANT ロール利用権限	○	×	×
REVOKE ロール利用権限	○	×	×

（凡例）

○：操作できます。

×：操作できません。

注※1

スキーマ定義権限の場合は，自スキーマに対する操作，DBA 権限，及びスキーマ操作権限のときは，操作対象スキーマ内に対する操作を示します。

注※2

PUBLIC 指定の SQL 操作はできません。

2.1.2　改竄防止表との関係

表のアクセス権限以外にも，表に対するセキュリティ機能の一つに**改竄防止機能**があります。表を定義するときに**改竄防止オプション**（INSERT ONLY）を指定すると，その表は**改竄防止表**になります。改竄防止表の目的と特徴を次に示します。

目的

- 人為的なミスによるデータの削除又は更新を防げます。
- 不正なデータ更新又は削除からデータを守ることができます。

特徴

- UPDATE 権限がある場合でも表を更新できません。また，表の所有者も表を更新できません。
- DELETE 権限がある場合でも，**行削除禁止期間**を過ぎていないデータは削除できません。表の所有者も削除できません。
- INSERT 権限があれば行を挿入できます。
- SELECT 権限があれば表を検索できます。

改竄防止機能については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

2.1.3　監査証跡表との関係

HiRDB のリソースにアクセスするイベントが発生したとき，セキュリティに関してチェックした内容を監査証跡として監査証跡表に登録する機能（セキュリティ監査機能）を HiRDB ではサポートしています。監査証跡表には，だれが，いつ，どのリソースにアクセスし，セキュリティ上のチェックが成功又は失敗したかが記録されるため，不正なアクセスなどを監査するときの資料として利用できます。

また，監査証跡表を不正に改竄できないように監査証跡表を操作できるユーザが次の表に示すように制限されています。

表 2–4　監査証跡表を操作できるユーザ

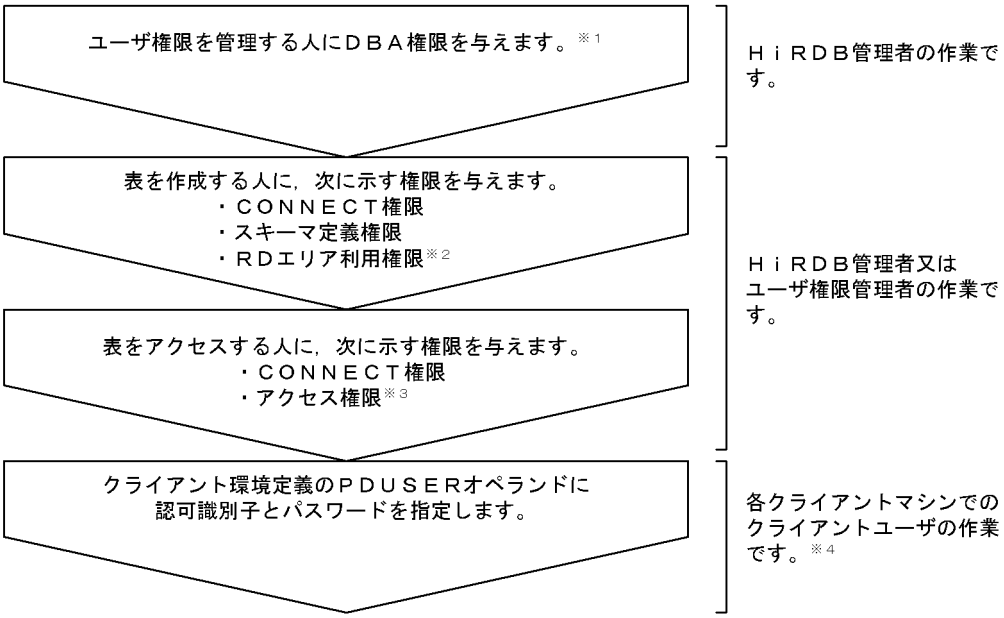
監査証跡表に対する操作	操作できるユーザ
データを参照（SELECT）する	• 監査人 • 監査証跡表に SELECT 権限があるユーザ
データを削除（DELETE 及び PURGE）する	監査人
データを追加又は変更（INSERT 又は UPDATE）する	なし

セキュリティ監査機能については，「セキュリティ監査機能の運用」を参照してください。

2.2　ユーザ権限を設定するには

ユーザ権限を設定する手順を次の図に示します。

図 2–1　ユーザ権限を設定する手順



注※１　HiRDB管理者以外のユーザが，ユーザ権限を管理する場合に必要となる作業です。
注※２　私用ＲＤエリアに表を作成する場合に必要です。
注※３　アクセスする表のアクセス権限が必要です。
注※４　表を作成する人，又は表をアクセスする人の作業です。

監査権限を設定する場合

監査権限の設定方法については，「監査人の登録」を参照してください。

スキーマ操作権限を設定する場合

スキーマ操作権限の設定方法については，「他ユーザにスキーマ操作権限を与えます」を参照してください。

2.2.1　ユーザ権限を管理する人に DBA 権限を与えます

実行者　HiRDB 管理者

ユーザ権限を管理する人に定義系 SQL の **GRANT 文**で DBA 権限を与えます。なお，DBA 権限の付与は，ユーザ権限を管理する人が HiRDB 管理者以外の場合に必要となる作業です。

例題

ユーザ権限を管理する人（認可識別子：USER001，パスワード：HIRDB001）に，DBA 権限を与えます。

```
GRANT DBA TO USER001 IDENTIFIED BY HIRDB001
```

2.2.2　表を作成する人に CONNECT 権限，スキーマ定義権限，RD エリア利用権限を与えます

実行者　HiRDB 管理者又はユーザ権限を管理する人（DBA 権限保持者）

表を作成する人に，定義系 SQL の **GRANT 文**で次に示す権限を与えます。

- CONNECT 権限
- スキーマ定義権限
- RD エリア利用権限※

注※

私用 RD エリアに表を作成する場合に必要です。

公用 RD エリアに表を作成する場合は不要です。

例題

表を作成するユーザ（認可識別子：USER002，パスワード：HIRDB002）に，CONNECT 権限，スキーマ定義権限，RD エリア利用権限（RD エリア名：RDAREA01）を与えます。

```
GRANT CONNECT TO USER002 IDENTIFIED BY HIRDB002
GRANT SCHEMA TO USER002
GRANT RDAREA RDAREA01 TO USER002
```

(1)　GRANT 文以外でスキーマ定義権限を与える方法

スキーマ定義権限を与えたい人のスキーマを定義します。そうすると，スキーマを定義した人に，スキーマ定義権限が与えられます。

(2)　GRANT 文以外で RD エリア利用権限を与える方法

HiRDB 管理者が私用 RD エリアを定義するとき※に，その私用 RD エリアに対して RD エリア利用権限を与えられます。

注※

次に示すユティリティの制御文で指定します。

- データベース初期設定ユティリティの create rdarea 文の USER USED BY オペランド
- データベース構成変更ユティリティの create rdarea 文の USER USED BY オペランド

(3)　私用 RD エリアを公用 RD エリアにする方法

私用 RD エリアの RD エリア利用権限を削除した後に，定義系 SQL の **GRANT 文**で変更します。私用 RD エリアの RD エリア利用権限の削除方法については，「RD エリア利用権限を取り消すには」を参照してください。

例題

RD エリア（RD エリア名：RDAREA02）を公用 RD エリアにします。

```
GRANT RDAREA RDAREA02 TO PUBLIC
```

2.2.3 表（データベース）をアクセスする人に CONNECT 権限，アクセス権限を与えます

(1) CONNECT 権限を与えます

実行者　HiRDB 管理者又はユーザ権限を管理する人（DBA 権限保持者）

表をアクセスする人に，定義系 SQL の **GRANT 文**で CONNECT 権限を与えます。

例題

表をアクセスする人（認可識別子：USER003，パスワード：HIRDB003）に，CONNECT 権限を与えます。

```
GRANT CONNECT TO USER003 IDENTIFIED BY HIRDB003
```

(2) アクセス権限を与えます

実行者　表の所有者

表（自分の表）をアクセスする人に，定義系 SQL の **GRANT 文**でアクセス権限を与えます。

例題１

検索だけを許す人（認可識別子：USER004）には，表（認可識別子．表識別子：USER002.T001）の SELECT 権限だけを与えます。

```
GRANT SELECT ON USER002.T001 TO USER004
```

例題２

検索及び更新を許す人（認可識別子：USER005）には，表（認可識別子．表識別子：USER002.T001）の SELECT 権限及び UPDATE 権限を与えます。

```
GRANT SELECT,UPDATE ON USER002.T001 TO USER005
```

例題３

検索，更新，追加，及び削除を許す人（認可識別子：USER006）には，表（認可識別子．表識別子：USER002.T001）の SELECT 権限，UPDATE 権限，INSERT 権限，及び DELETE 権限を与えます。

```
GRANT ALL ON USER002.T001 TO USER006
```

注意事項

- 表の所有者が持っているアクセス権限だけを，ほかの人に与えられます。表の所有者が持つアクセス権限を表「表の所有者が持つアクセス権限」に示します。
- ほかの人の表から定義したビュー表のアクセス権限は，ほかの人に与えられません。

表 2–5　表の所有者が持つアクセス権限

表の種類	表の所有者が持つアクセス権限	アクセス権限の付与及び取り消しの可否
実表	すべてのアクセス権限	○
自分の実表から定義した読み込み専用のビュー表※1	SELECT 権限	○
自分の実表から定義した更新可能なビュー表※2	すべてのアクセス権限	○
他人の実表から定義した読み込み専用のビュー表※1※3	SELECT 権限	×
他人の実表から定義した更新可能なビュー表※2※3	実表に対して持っているすべてのアクセス権限	×

（凡例）

○：アクセス権限をほかのユーザに与えたり，取り消したりできます。

×：アクセス権限をほかのユーザに与えたり，取り消したりできません。

注※1

読み込み専用のビュー表とは，ビュー定義中に次に示す指定をしているビュー表のことです。

- SELECT 句中に DISTINCT，集合関数，定数，又は四則演算を指定している場合
- SELECT 句に実表の同一列を複数指定している場合
- 表の結合をしている場合
- GROUP BY 句を指定している場合
- HAVING 句を指定している場合
- READ ONLY 句を指定している場合

注※2

更新可能なビュー表とは，読み込み専用でないビュー表のことです。

注※3

ほかの人の表からビュー表を定義するには，その表の SELECT 権限が必要です。

2.2.4　PDUSER オペランドに認可識別子とパスワードを指定します

実行者　クライアントユーザ

クライアント環境定義の PDUSER オペランドに認可識別子とパスワードを指定します。この作業は，クライアントユーザが各自のクライアントマシンで行います。

PDUSER オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

例題

HiRDB 管理者から CONNECT 権限（認可識別子：USER003，パスワード：HIRDB003）を与えられました。その認可識別子とパスワードをクライアント環境定義の PDUSER オペランドに指定します。

クライアント環境定義の内容

```
PDUSER = USER003/HIRDB003
```

2.2.5　他ユーザにスキーマ操作権限を与えます

実行者　スキーマ所有者

スキーマ操作権限は，定義系 SQL の GRANT 文で付与します。スキーマ操作権限の付与は，自分が所有しているリソースの定義や変更を，自分以外がする場合に必要な作業です。

また，スキーマ操作権限対象のスキーマ上のリソースを作成・変更する場合は，定義系 SQL で，リソースの認可識別子にスキーマ操作権限対象のスキーマ名を指定します。

例題

スキーマ操作権限を使用して他人の表及びインデクスを作成します。次の手順で作成してください。

1. 実行者（認可識別子：USER007）が所有しているスキーマに対して，リソースの定義や変更を行うユーザ（認可識別子：USER008，USER009）にスキーマ操作権限を与えます。

   ```
   GRANT SCHEMA OPERATION TO USER008,USER009
   ```

2. ユーザ（認可識別子：USER007）からスキーマ操作権限を付与されたユーザ（認可識別子：USER008）が，スキーマ操作権限対象のスキーマ上に表（T002）を作成します。

   ```
   CREATE TABLE USER007.T002(C001 INT)
   ```

3. ユーザ（認可識別子：USER007）からスキーマ操作権限を付与されたユーザ（認可識別子：USER009）が，スキーマ操作権限対象のスキーマ上の表（T002）にインデクス（I001）を作成します。

   ```
   CREATE INDEX USER007.I001 ON USER007.T002(C001)
   ```

2.3　ユーザ権限を取り消すには

2.3.1　DBA 権限を取り消すには

実行者　HiRDB 管理者

定義系 SQL の **REVOKE 文**で，DBA 権限を取り消します。

例題

ユーザ権限を管理する人（認可識別子：USER001）の DBA 権限を取り消します。

```
REVOKE DBA FROM USER001
```

2.3.2　CONNECT 権限を取り消すには

実行者　HiRDB 管理者又はユーザ権限を管理する人（DBA 権限保持者）

定義系 SQL の **REVOKE 文**で，CONNECT 権限を取り消します。

例題

認可識別子 USER003 の CONNECT 権限を取り消します。

```
REVOKE CONNECT FROM USER003
```

2.3.3　スキーマ定義権限を取り消すには

実行者　HiRDB 管理者又はユーザ権限を管理する人（DBA 権限保持者）

定義系 SQL の **REVOKE 文**で，スキーマ定義権限を取り消します。

例題

認可識別子 USER002 のスキーマ定義権限を取り消します。

```
REVOKE SCHEMA FROM USER002
```

注意事項

スキーマを定義している人のスキーマ定義権限は取り消せません。スキーマ定義権限を取り消す場合は，取り消す人のスキーマが定義されていないかを確認してください。

2.3.4　RD エリア利用権限を取り消すには

実行者　HiRDB 管理者又はユーザ権限を管理する人（DBA 権限保持者）

定義系 SQL の **REVOKE 文**で，RD エリア利用権限を取り消します。

例題

認可識別子 USER002 の RD エリア利用権限（RD エリア名：RDAREA01）を取り消します。

```
REVOKE RDAREA RDAREA01 FROM USER002
```

注意事項

RD エリアに自分の表又はインデクスを定義している場合，その人の RD エリア利用権限は取り消せません。RD エリア利用権限を取り消す場合は，対象ユーザの表又はインデクスが定義されていないかを確認してください。

2.3.5　アクセス権限を取り消すには

実行者　表の所有者

定義系 SQL の **REVOKE 文**で，アクセス権限を取り消します。

例題

認可識別子 USER004 に与えていた，表（認可識別子．表識別子：USER002.T001）の DELETE 権限を取り消します。

```
REVOKE DELETE ON USER002.T001 FROM USER004
```

注意事項

表のアクセス権限（SELECT 権限）を取り消すと，その表を基にして作成したビュー表が削除されます。

2.3.6　スキーマ操作権限を取り消すには

実行者　スキーマ所有者

定義系 SQL の REVOKE 文で削除します。

例題

実行者が他ユーザ（認可識別子：USER008，USER009）に与えたスキーマ操作権限を削除します。

```
REVOKE SCHEMA OPERATION FROM USER008,USER009
```

2.4　ディクショナリ表の参照権限を設定するには

実行者　HiRDB 管理者

システムのセキュリティを強化する場合は，ディクショナリ表の参照権限を設定し，参照できるディクショナリ表を制限してください。次に示すユティリティの制御文で limited を指定すると，ディクショナリ表の参照権限を設定できます。

- データベース初期設定ユティリティの **define system 文**の **dicinf** オペランド
- データベース構成変更ユティリティの **alter system 文**の **dicinf** オペランド

dicinf オペランドの指定値と参照できるディクショナリ表の関係を次の表に示します。

表 2–6　dicinf オペランドの指定値と参照できるディクショナリ表の関係

ディクショナリ表の種類	dicinf オペランドの指定値			
	limited			unlimited
	DBA 権限保持者	監査人	一般ユーザ	
SQL_PHYSICAL_FILES	○	○	RD エリア利用権限がある RD エリアを構成する HiRDB ファイルの情報だけを参照できます。	○
SQL_RDAREAS	○	○	RD エリア利用権限がある RD エリアの情報だけを参照できます。	○
SQL_TABLES	○※1	○※1	アクセス権限がある表の情報だけを参照できます。	○
SQL_COLUMNS	○※1	○※1	アクセス権限がある表の列の情報だけを参照できます。	○
SQL_INDEXES	○※1	○※1	アクセス権限がある表のインデクスの情報だけを参照できます。	○
SQL_USERS	○	○	×	×
SQL_RDAREA_PRIVILEGES	○	○	RD エリア利用権限がある RD エリアの情報だけを参照できます。	○
SQL_TABLE_PRIVILEGES	○※1	○※1	アクセス権限がある表の情報だけを参照できます。	○
SQL_DIV_TABLE	○※1	○※1		○
SQL_INDEX_COLINF	○※1	○※1	アクセス権限がある表のインデクスの情報だけを参照できます。	○
SQL_TABLE_STATISTICS	○	○	アクセス権限がある表の表統計情報だけを参照できます。	○

ディクショナリ表の種類	dicinf オペランドの指定値			
	limited			unlimited
	DBA 権限保持者	監査人	一般ユーザ	
SQL_COLUMN_STATISTICS	○	○	アクセス権限がある表の列統計情報だけを参照できます。	○
SQL_INDEX_STATISTICS	○	○	アクセス権限がある表のインデクス統計情報だけを参照できます。	○
SQL_VIEW_TABLE_USAGE	○	○	アクセス権限があるビュー表の情報だけを参照できます。	○
SQL_VIEWS	○	○		○
SQL_DIV_INDEX	○※1	○※1	アクセス権限がある表のインデクスの情報だけを参照できます。	○
SQL_DIV_COLUMN	○※2	○※2	アクセス権限がある表の情報だけを参照できます。	○
SQL_REFERENTIAL_CONSTRAINTS	○	○	アクセス権限がある表の制約情報だけを参照できます。	○
SQL_ALIASES	○	○	○	○
SQL_ROUTINES	○	○	○	○
SQL_ROUTINE_RESOURCES	○	○	○	○
SQL_ROUTINE_PARAMS	○	○	○	○
SQL_DATATYPES	○	○	○	○
SQL_DATATYPE_DESCRIPTORS	○	○	○	○
SQL_TABLE_RESOURCES	○	○	○	○
SQL_PLUGINS	○	○	○	○
SQL_PLUGIN_ROUTINES	○	○	○	○
SQL_PLUGIN_ROUTINE_PARAMS	○	○	○	○
SQL_REGISTRY	○	○	○	○
SQL_INDEX_TYPES	○	○	○	○
SQL_INDEX_DATATYPE	○	○	○	○
SQL_INDEX_FUNCTION	○	○	○	○
SQL_TYPE_RESOURCES	○	○	○	○
SQL_INDEX_RESOURCES	○	○	○	○
SQL_INDEX_TYPE_FUNCTION	○	○	○	○

ディクショナリ表の種類	dicinf オペランドの指定値			
	limited			unlimited
	DBA 権限保持者	監査人	一般ユーザ	
SQL_EXCEPT	○※1	○※1	アクセス権限がある表のインデクスの情報だけを参照できます。	○
SQL_IOS_GENERATIONS	○	○	○	○
SQL_PARTKEY	○	○	アクセス権限がある表の情報だけを参照できます。	○
SQL_PARTKEY_DIVISION	○	○	アクセス権限がある表の情報だけを参照できます。	○
SQL_TRIGGERS	○	○	自分で定義したトリガの情報だけを参照できます。	○
SQL_TRIGGER_COLUMNS	○	○		○
SQL_TRIGGER_DEF_SOURCE	○	○		○
SQL_TRIGGER_USAGE	○	○		○
SQL_AUDITS	×	○	×	×
SQL_KEYCOLUMN_USAGE	○	○	アクセス権限がある表の情報だけを参照できます。	○
SQL_TABLE_CONSTRAINTS	○	○		○
SQL_CHECKS	○	○		○
SQL_CHECK_COLUMNS	○	○		○
SQL_DIV_TYPE	○	○		○
SQL_SYSPARAMS	○	○	×	×
SQL_INDEX_XMLINF	○	○	アクセス権限がある表の情報だけを参照できます。	○
SQL_SEQUENCES	○	○	自分で定義した順序数生成子の情報だけを参照できます。	○
SQL_ACCESS_SECURITY	○	○	×	×
SQL_ROLES	○	○	×	○
SQL_ROLE_PRIVILEGES	○	○	自分で定義したロール利用権限の情報だけを参照できます。	○

（凡例）

○：すべての列を参照できます。

×：すべての列を参照できません。

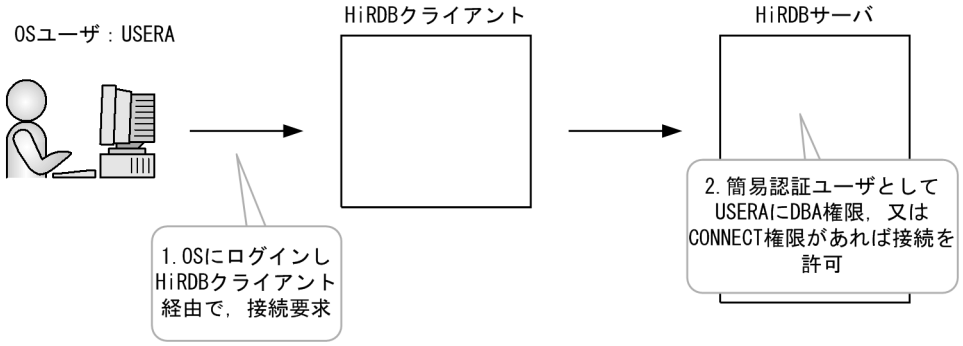
注※1　ディクショナリ表の実表は参照できません。

注※2　ディクショナリ表の実表も参照できます。

2.5　OS ログインユーザの簡易認証機能とは

OS ログインユーザの簡易認証機能とは，HiRDB クライアントから，そのクライアントが動作する OS にログインしているユーザ名で HiRDB サーバに接続する機能です。また，OS ログインユーザの簡易認証機能を使用するユーザを**簡易認証ユーザ**といいます。OS ログインユーザの簡易認証機能は，簡易認証ユーザとして DBA 権限，又は CONNECT 権限を付与されたユーザが利用できます。OS ログインユーザの簡易認証機能を使用した場合の HiRDB サーバへの接続時の流れを次の図に示します。

図 2–2　OS ログインユーザの簡易認証機能を使用した際の HiRDB サーバへの接続時の流れ



なお，簡易認証ユーザに対しては，スキーマ定義権限の付与ができません。また，DBA 権限保持者が簡易認証ユーザのスキーマを作成することもできません。OS ログインユーザの簡易認証機能を使用する場合は，スキーマは簡易認証ユーザ以外のユーザで作成し，これらに対するアクセス権限を簡易認証ユーザに付与してください。簡易認証ユーザはスキーマを所有できないため，与えられたアクセス権限を使用して，簡易認証ユーザ以外が所有する表などのリソースにアクセスする運用となります。

2.5.1　OS ログインユーザの簡易認証機能を使用するための準備

(1)　システム定義による OS ログインユーザの簡易認証機能の有効化

OS ログインユーザの簡易認証機能を有効にするためには，システム共通定義 pd_os_authenticate を指定します。

pd_os_authenticate の指定内容については，マニュアル「HiRDB システム定義」の「オペランドの説明」を参照してください。

(2)　簡易認証ユーザの設定

簡易認証ユーザとして登録するユーザを，SQL の GRANT 文で設定します。

GRANT 文による簡易認証ユーザの指定方法については，マニュアル「HiRDB SQL リファレンス」の「GRANT　形式 1（権限定義）」を参照してください。

2.5.2　OS ログインユーザの簡易認証機能による HiRDB サーバへの接続方法

サーバに接続する際，認可識別子及びパスワードの両方に，この機能を使用して認証することを示す半角ハイフン（-）を指定します。これを簡易認証キーワードといいます。

簡易認証キーワードは次の方法で指定できます。

- **クライアント環境定義 PDUSER**

  環境変数グループ，ユーザ環境変数，デフォルト設定（HiRDB.ini）で指定できます。

  （指定例）

  UNIX シェルの環境変数 PDUSER

```
export PDUSER='"-"/"-"'
```

- **HiRDB が提供するコマンドのオプション**

  運用コマンド，ユティリティなどで指定できます。

- **HiRDB が提供するサーバ接続関数の引数，プロパティ**

  ODBC ドライバ，JDBC ドライバ，HiRDB データプロバイダ for .NET Framework，及び OLE データプロバイダで指定できます。

  （指定例）

  ODBC ドライバのサーバ接続関数

```
SQLConnect(hdbc, "DSN", 3, "-", 1, "-", 1); // 認可識別子 "-", パスワード "-"
```

- **ソースファイルへ埋め込んだ SQL 文の引数**

  C/C++，COBOL，SQLJ の SQL 文埋め込みソースファイルで指定できます。

  （指定例）

  C/C++の埋め込み SQL 文

```
char xuserid[31] = "-";    // 認可識別子 "-"
char xpasswd[31] = "-";    // パスワード "-"
EXEC SQL CONNECT USER :xuserid IDENTOFIED :xpasswd; // Connect実行
```

注意事項

次の場合は，HiRDB サーバへの接続に失敗します。

- OS にログインしているユーザ名の長さが，簡易認証ユーザとして使用できるユーザ名の文字数より長い場合（KFPA11564-E メッセージが出力されます）。
- OS にログインしているユーザ名の取得に失敗した場合（KFPA11731-E メッセージが出力されます）。

2.5.3　HiRDB のパスワード認証と OS ログインユーザの簡易認証の適用条件

システム共通定義 pd_os_authenticate と，認可識別子及びパスワードの組み合わせによって適用される認証方法を次の表に示します。

表 2–7　適用される認証方法

HiRDB クライアントのバージョン，及び認可識別子，パスワードの指定方法		HiRDB サーバのバージョンと pd_os_authenticate の指定方法		
		バージョン 09-50 より前	バージョン 09-50 以降	
			pd_os_authenticate の指定	
			Y	N
バージョン 09-50 より前	認可識別子，パスワードの指定あり	HiRDB のパスワード認証	HiRDB のパスワード認証	HiRDB のパスワード認証
	認可識別子，パスワードの指定が-/-	認証不可（KFPA11564-E メッセージを出力）	認証不可（KFPA11564-E メッセージを出力）	認証不可（KFPA11564-E メッセージを出力）
バージョン 09-50 以降	認可識別子, パスワードの指定あり	HiRDB のパスワード認証	HiRDB のパスワード認証	HiRDB のパスワード認証
	認可識別子, パスワードの指定が-/-	認証不可（KFPA11564-E メッセージを出力）	OS ログインユーザの簡易認証	認証不可（KFPA19632-E メッセージを出力）

2.5.4　簡易認証ユーザに付与できる権限

簡易認証ユーザに付与できる権限を次の表に示します。

表 2–8　簡易認証ユーザに付与できる権限

付与できる権限					
監査権限	DBA 権限	CONNECT 権限	スキーマ定義権限	RD エリア利用権限	アクセス権限
×	○	○	○	○	○

（凡例）

○：権限を付与できます。

×：権限を付与できません。

2.5.5　OS ログインユーザの簡易認証機能を使用する際の注意事項

- 簡易認証ユーザとして使用できるユーザ名の制約（使用文字，文字数の最大）は，認可識別子の制約に従います。

- OS ログインユーザの簡易認証機能を使用する場合，ファイアウォールによって意図しないマシンからの接続を遮断する運用を推奨します。
- OpenTP1 など OLTP 下の UAP をクライアントとして使用する場合は，UAP プロセスが OLTP 下で動作するため，OS ログインユーザの簡易認証機能を使用しない運用を推奨します。
- OS ログインユーザの簡易認証機能は，サーバ接続時のユーザ認証の場合だけ使用できます。その他の用途には使用できません。
  - 認可識別子の指定には使用できません。

    SQL 文での表の所有者，SQL の実行者，操作の対象などを指定する用途で，認可識別子に簡易認証キーワード"-"（半角ハイフン）を指定しても，認可識別子の OS ユーザ名への置き換えは実施しません。
  - ユーザ切り替え（SQL の SET SESSION AUTHORIZATION 文）には使用できません。

    SQL 文でのユーザ切り替え時に簡易認証キーワード"-"を指定しても，HiRDB サーバへの接続に失敗します（KFPA11564-E メッセージを出力）。
- Type4 JDBC ドライバで OS ログインユーザの簡易認証機能を使用する場合は，Type4 JDBC ドライバが OS ログインユーザを JavaVM のシステムプロパティ user.name から取得します。そのため，システムプロパティ user.name の値の取得に失敗した場合，または取得した値が null の場合は，HiRDB サーバへの接続に失敗します（KFPJ80004-E メッセージを出力）。
- OS ログインユーザの簡易認証機能を使用する場合，HiRDB クライアントから LDAP サーバへ接続できない状態，又は LDAP サーバのホスト名解決に失敗する状態が発生していると，HiRDB サーバへの接続に時間が掛かることがあります。
- Windows で OS ログインユーザの簡易認証機能を使用する場合，GRANT 文でユーザ名（認可識別子）をダブルクォートで囲まないで簡易認証ユーザを設定したときは，大文字のユーザ名で Windows にログインしてください。GRANT 文でユーザ名（認可識別子）をダブルクォートで囲んで簡易認証ユーザを設定した場合は，設定したユーザ名と大文字，小文字まで合わせて同じユーザ名で Windows にログインしてください。
- コマンド，ユティリティについての注意事項は，マニュアル「HiRDB コマンドリファレンス」の「OS ログインユーザの簡易認証機能使用時の規則」を参照してください。
- HiRDB の関連製品については，この機能を使用できるのは次の表に示す製品だけです。

表 2–9　OS ログインユーザの簡易認証機能が使用できる HiRDB 関連製品

カテゴリ	製品名
クライアント製品	HiRDB/Developer's Kit
	HiRDB/Run Time
HiRDB の付加 PP	HiRDB Data Convert Type1 Option
	HiRDB Accelerator
	HiRDB Advanced High Availability
	HiRDB Disaster Recovery Light Edition

カテゴリ	製品名
	HiRDB Non Recover Front End Server
	HiRDB Staticizer Option
運用支援製品	HiRDB SQL Executer
	HiRDB SQL Tuning Advisor

2.6　データベースの機密保護での認可識別子

認可識別子とは，権限を識別する上での識別子でありユーザ識別子とロール名のことを指します。ただし，特に断りがない場合には，認可識別子はユーザ識別子のことを指します。

認可識別子がロール名を指す個所については，「ユーザ識別子とロール名の指定できる個所」のロール名を指定できる個所を参照してください。

2.6.1　ユーザ識別子

ユーザ識別子とは，HiRDB に接続するユーザの識別子のことです。

ユーザは，HiRDB への接続時にユーザ識別子とパスワードによってユーザ認証を受けたり，ユーザ識別子単位にスキーマを所有したりすることができます。ユーザ識別子が持つ権限によって，SQL 文，UAP，及びユティリティなどを実行できます。

ユーザ識別子はすべてのユーザ権限を持ちます。ただし，ユーザ識別子が持つ権限を，ほかのユーザ識別子やロール名に使用させることはできません。

2.6.2　ロール名

ロール名とは，権限の集合であるロールの識別子のことです。

ロールはアクセス権限，及びロール利用権限を持ちます。また，ユーザ識別子及びほかのロール名にロール利用権限を付与することで，ロールが持つ権限をユーザ識別子及びほかのロール名に使用させることができます。

ロール名は，HiRDB への接続時のユーザ認証を受けられません。また，ロール名を実行者として，SQL 文，UAP，及びユティリティなどを実行できません。

ロールの詳細や運用については「ロールを使用した権限の設定」を参照してください。

2.6.3　ユーザ識別子とロール名の指定できる個所

ユーザ識別子とロール名では，指定できる個所が異なります。ユーザ識別子とロール名を指定できる個所を次の表に示します

表 2–10　ユーザ識別子とロール名を指定できる個所

指定個所	ユーザ識別子	ロール名
次の SQL 文で TO に続けて指定する認可識別子	○	○

指定個所	ユーザ識別子	ロール名
• GRANT アクセス権限 • GRANT ロール利用権限		
次の SQL 文で FROM に続けて指定する認可識別子 • REVOKE アクセス権限 • REVOKE ロール利用権限	○	○
CREATE ROLE で指定するロール名	×	○
DROP ROLE で指定するロール名	×	○
GRANT でロール利用権限を付与するロール名	×	○
REVOKE でロール利用権限を取り消すロール名	×	○
上記以外の認可識別子 ○ ×	○	×

（凡例）

○：指定できます。

×：指定できません。

2.7　ロールを使用した権限の設定

2.7.1　ロールとは

ロールとは権限の集合のことです。

権限をロールに付与して，必要な権限をロールに集約しておくと，そのロールのロール利用権限を付与，取り消しすることで，ロールが持つ権限をまとめて付与，取り消しできるため，権限の管理や移行が容易になります。

例えば，業務 A に必要なアクセス権限をすべてロール A に付与しておくと，その業務に関するユーザへのアクセス権限の付与，取り消しは，ロール A のロール利用権限の付与，取り消しだけになります。このため，ある業務で使用する多くの表へのアクセス権限の一つ一つをユーザごとに付与，取り消ししなくてもよくなります。また，ロール利用権限はほかのロールに付与できます。

業務 A と業務 B の両方のアクセス権限が必要な場合，業務 A に必要なアクセス権限を集約したロール A のロール利用権限と業務 B に必要なアクセス権限を集約したロール B のロール利用権限を，別のロール AB に付与することで効率よく管理できます。

ロールを使用して設定できる権限を次に示します。

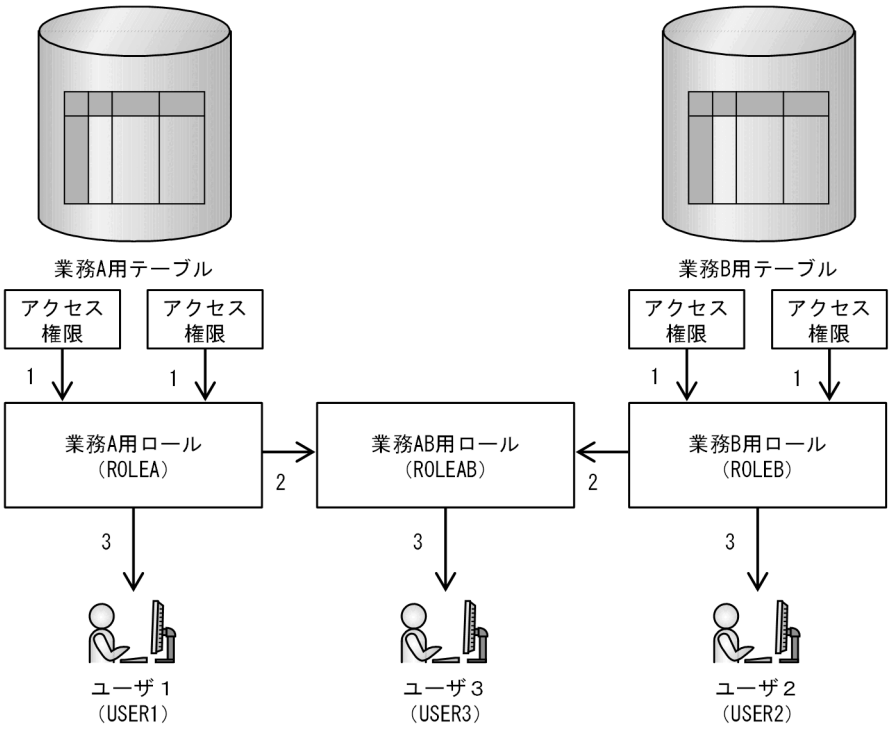
- アクセス権限

ロールを使用する場合には，使用するロールを現在ロールに指定する必要があります。現在ロールとは，現在有効とするロールのことを指します。また，使用するロールのロール利用権限を持っているだけではロールが持つ権限を使用できません。

現在ロールを設定している場合，ユーザが持つ権限と現在ロールが持つ権限を包含する権限を持ちます。

ロールの概要を次の図に示します。図中の番号は説明の番号と対応します。

図 2–3　ロールの概要



〔説明〕

1. 業務用のアクセス権限を各業務用のロールに付与します。
2. 各業務用のロールのロール利用権限を両方の業務用のロールに付与します。
3. 各ロールのロール利用権限をユーザに付与します。

この例では，次の表のようにアクセス権限を持ちます。

表 2–11　ユーザが持つアクセス権限

ユーザ名	使用できるアクセス権限
ユーザ 1（USER1）	現在ロールに ROLEA を指定すると，業務 A 用テーブルにアクセスするために必要なアクセス権限を持ちます。
ユーザ 2（USER2）	現在ロールに ROLEB を指定すると，業務 B 用テーブルにアクセスするために必要なアクセス権限を持ちます。
ユーザ 3（USER3）	現在ロールに ROLEAB を指定すると，業務 A 用，業務 B 用テーブルにアクセスするために必要なアクセス権限を持ちます（両方を同時に使用できます）。

2.7.2　ロールの運用手順

次の 3 つのロールを作成して USER1，USER2，USER3 の権限を設定する場合の，ロールの運用手順を，図「ロールの概要」に基づいて説明します。

- 業務 A 用テーブル（TA1，TA2）にアクセスできる権限を持つロール（ロール名：ROLEA）
- 業務 B 用テーブル（TB1，TB2）にアクセスできる権限を持つロール（ロール名：ROLEB）
- 業務 A 用テーブル，業務 B 用テーブルのどちらもアクセスできる権限を持つロール（ロール名：ROLEAB）

ロールの運用手順（使用する SQL 文，及び操作）を次に示します。

表 2–12　ロールの運用手順

運用手順		使用する SQL 文概要
定義系 SQL によるロールの定義	表・インデクスなどリソース定義	定義系 SQL
	ロールの定義	CREATE ROLE
	ロールへのアクセス権限付与	GRANT アクセス権限
	ユーザへのロール利用権限付与	GRANT ロール名 TO ユーザ名
	ロールへのロール利用権限付与	GRANT ロール名 TO ロール名
ロールの使用	使用するロールの設定，現在ロールの無効化	SET ROLE
ロールが持つアクセス権限の変更	ロールからのアクセス権限取り消し	REVOKE アクセス権限
	ロールへのアクセス権限付与	GRANT アクセス権限
ロール利用権限の取り消し	ユーザからのロール利用権限取り消し	REVOKE ロール名 FROM ユーザ
	ロールからのロール利用権限取り消し	REVOKE ロール名 FROM ロール名
ロールの削除	ロールの削除	DROP ROLE

(1)　表・インデクスなどのリソース定義

ロールでアクセス権限を管理する表・インデクスなどリソースを定義します。

(2)　ロールの定義

実行者　HiRDB 管理者又はユーザ権限を管理する人（DBA 権限保持者）

定義系 SQL の CREATE ROLE 文で，次に示すロールを作成します。

- 表 TA1 及び表 TA2 への SELECT 権限を持ったロール ROLEA
- 表 TB1 及び表 TB2 への SELECT 権限を持ったロール ROLEB
- 表 TA1 及び表 TA2 への SELECT 権限を持ったロール ROLEA，並びに表 TB1 及び表 TB2 への SELECT 権限を持ったロール ROLEB の両方のアクセス権限を持ったロール ROLEAB

例題 1

ロール ROLEA を定義します。

```
CREATE ROLE ROLEA
```

例題 2

ロール ROLEB を定義します。

```
CREATE ROLE ROLEB
```

例題 3

ロール ROLEAB を定義します。

```
CREATE ROLE ROLEAB
```

(3)　ロールへのアクセス権限付与

実行者　表の所有者

ロールを作成した後に，定義系 SQL の **GRANT 文**でロールにアクセス権限を付与します。

例題 1

実行者（認可識別子：USER001）が所有している表（TA1,TA2）に対して，ロール（ROLEA）に SELECT 権限を与えます。

```
GRANT SELECT ON USER001.TA1 TO ROLEA
GRANT SELECT ON USER001.TA2 TO ROLEA
```

例題 2

実行者（認可識別子：USER002）が所有している表（TB1,TB2）に対して，ロール（ROLEB）に UPDATE 権限を与えます。

```
GRANT UPDATE ON USER002.TB1 TO ROLEB
GRANT UPDATE ON USER002.TB2 TO ROLEB
```

(4)　ユーザへのロール利用権限付与

実行者　ロールの作成者

ロールを作成してアクセス権限を付与した後に，定義系 SQL の **GRANT 文**でユーザにロールの利用権限を付与します。

例題 1

ロール（ROLEA）のロール利用権限をユーザ（USER001）に与えます。

```
GRANT ROLEA TO USER001
```

例題 2

ロール（ROLEB）のロール利用権限をユーザ（USER002）に与えます。

```
GRANT ROLEB TO USER002
```

(5)　ロールへのロール利用権限付与

実行者　ロールの作成者

ロールを作成してアクセス権限を付与した後に，ロールに対して定義系 SQL の **GRANT 文**でロールの利用権限を付与します。また，付与したロール（ROLEAB）の利用権限を，ユーザ（USER003）に付与します。

例題

ロール（ROLEA，ROLEB）のロール利用権限をロール（ROLEAB）に与えます。

```
GRANT ROLEA,ROLEB TO ROLEAB
```

ロール（ROLEAB）のロール利用権限をユーザ（USER003）に与えます。

```
GRANT ROLEAB TO USER003
```

(6)　使用するロールの設定

ロールを使用する場合，使用するロールを現在ロールに設定するとロールが持つ権限を使用できます。現在ロールの設定をしないとロール利用権限を持っていてもロールが持つ権限を使用できません。

また，複数のロールのロール利用権限を持っている場合は，SET ROLE 文を実行することで現在ロールを変更できます。なお，トランザクション実行中は，現在ロールを変更できません。

次の方法で，使用する現在ロールを設定します。なお，複数のロールを一度に使用することはできません。

実行者　ロール利用権限保持者

制御系 SQL の **SET ROLE 文**で次に示すロールを設定します。

- 表 TA1 及び表 TA2 への SELECT 権限を持ったロール ROLEA を現在ロールに設定します。

例題

ロール ROLEA を現在ロールに設定します。

```
SET ROLE ROLEA
```

(7)　現在ロールの無効化

実行者　ロール利用権限保持者

制御系 SQL の SET ROLE 文（NONE 指定）で設定されている現在ロールを無効化します。

- SET ROLE 文（NONE 指定）で設定した現在ロール（ROLEA）を無効化します。

例題

現在ロール（ROLEA）を無効化します。

```
SET ROLE NONE
```

(8)　ロールからのアクセス権限取り消し

実行者　表の所有者

定義系 SQL の **REVOKE 文**で，ロールからアクセス権限を取り消します。

例題

表 TA1，TA2 の SELECT 権限をロール（ROLEA）から取り消します。

```
REVOKE SELECT ON USER001.TA1 FROM ROLEA
REVOKE SELECT ON USER001.TA2 FROM ROLEA
```

(9)　ロールへのアクセス権限付与

実行者　表の所有者

定義系 SQL の **GRANT 文**で，ロールにアクセス権限を付与します。

例題 1

実行者（認可識別子：USER001）が所有している表（TA1）に対して，ロール（ROLEA）に DELETE 権限を与えます。

```
GRANT DELETE ON USER001.TA1 TO ROLEA
```

例題 2

実行者（認可識別子：USER001）が所有している表（TA2）に対して，ロール（ROLEA）に全アクセス権限を与えます。

```
GRANT ALL ON USER001.TA2 TO ROLEA
```

(10)　ユーザからのロール利用権限取り消し

実行者　ロールの作成者

定義系 SQL の **REVOKE 文**で，ユーザに付与されたロール利用権限を取り消します。

例題

ユーザ（USER001）に付与された，ロール（ROLEA）のロール利用権限を取り消します。

```
REVOKE ROLEA FROM USER001
```

(11)　ロールからのロール利用権限取り消し

実行者　ロールの作成者

定義系 SQL の **REVOKE 文**で，ロールに付与されたロール利用権限を取り消します。

例題

ロール（ROLEAB）に付与された，ロール（ROLEA，ROLEB）のロール利用権限を取り消します。

```
REVOKE ROLEA,ROLEB FROM ROLEAB
```

(12)　ロールの削除

ロールを定義系 SQL の **DROP ROLE 文**で削除します。

なお，ロールが持つアクセス権限の取り消しやロール利用権限の取り消しを事前に行っていない場合は，関連するリソース（アクセス権限，ロール利用権限）を同時に削除します。

実行者　ロールの作成者

ロールを，定義系 SQL の **DROP ROLE 文**で削除します。

例題

ロール（ROLEA）を削除します。

```
DROP ROLE ROLEA
```

2.7.3　ロール使用時の留意事項

ロール使用時の留意事項を説明します。

(1)　ロールが持つ権限を適用できない処理

ロールが持つ権限は，定義系 SQL，プリプロセス，コンパイル時は適用できません。

例題

ロール（ROLEA）が持つ表 TA1 の SELECT 権限を使用してビュー表 VA1 は定義できません。

```
SET ROLE ROLEA
CREATE VIEW USER001.VA1 AS SELECT * FROM USER001.TA1
```

(2)　処理時間及び使用メモリ量に影響があるロール

大量のロール・ロール利用権限が定義されている環境では，SQL の処理時間及び使用メモリ量に影響があります。

例えば，次のようなケースです。

- GRANT ロール利用権限文でロール利用権限を付与する場合，指定したロール数，及び定義済みのロール・ロール利用権限付与数に比例して処理時間が長くなったり，使用メモリ量が増加したりするおそれがあります。

- SET ROLE 文で現在ロールを設定する場合，ロールに他のロールが含まれるなど複雑なロールを指定すると，処理時間が長くなったり，使用メモリ量が増加したりするおそれがあります。

3 システムログファイルの運用

この章では，システムログファイルの運用方法について説明します。

3.1　基本項目

ここでは，システムログファイルを運用する前に理解して欲しいことについて説明します。

3.1.1　システムログは障害の回復又はチューニング情報に使用されます

システムログファイルにはデータベースの更新履歴情報（システムログ）が格納されます。このシステムログは次に示す目的に使用されます。

- HiRDB 又は UAP が異常終了した場合，HiRDB がデータベース及びトランザクションを回復するのに使用されます。
- データベースに障害が発生した場合，HiRDB 管理者がデータベース回復ユティリティでデータベースを回復するのに使用されます。データベース回復ユティリティの入力情報になります。
- HiRDB 管理者がシステムのチューニングをする場合，チューニング情報の入力情報になります（統計解析ユティリティの入力情報になります）。

3.1.2　HiRDB はシステムログファイルを各種の状態に分けて管理しています

HiRDB はシステムログファイルを次の表に示す状態に分けて管理しています。

表 3–1　システムログファイルの状態

状態	説明	
現用	システムログの出力対象になっているファイルです。この状態のファイルは常に一つです。	
待機（スワップ先にできる状態）	現時点ではシステムログの出力対象ファイルになっていませんが，次に示す場合に現用ファイルとスワップするファイルです。 ・現用ファイルの容量が一杯になった場合 ・現用ファイルに障害が発生した場合 ・pdlogswap コマンドを入力した場合 この状態は次に示す条件をすべて満たしている必要があります。	
	上書きできる状態	ファイル中にシステムの全面回復に必要なシステムログ（有効保証世代数分のシンクポイントダンプに対応するシステムログ）を含んでいない状態です。
	アンロード済み状態	取得したシステムログをアンロードログファイルにアンロードした後の状態です。
	抽出完了状態（HiRDB Datareplicator）	抽出側 HiRDB Datareplicator がシステムログファイル中のシステムログを完全に読み出した状態です。HiRDB Datareplicator を使用する場合にこの状態になります。なお，HiRDB ではシンクポイントダンプを取得するときに抽出状態を確認しています。つまり，抽出側 HiRDB

状態	説明	
		Datareplicator がシステムログを完全に読み出した後に発生するシンクポイント時点で抽出完了状態になります。
待機（スワップ先にできない状態）	この状態のファイルはシステムログの出力対象になりません（現用ファイルになりません）。現用ファイルがスワップした後の状態です。この状態は次に示す条件のどれかを満たしている必要があります。	
	上書きできない状態	ファイル中にシステムの全面回復に必要なシステムログ（有効保証世代数分のシンクポイントダンプに対応するシステムログ）を含んでいる状態です。
	アンロード待ち状態	取得したシステムログをアンロードログファイルにアンロードしていない状態です。
	抽出未完了状態（HiRDB Datareplicator）	抽出側 HiRDB Datareplicator がシステムログファイル中のシステムログを完全に読み出していない（残りがある）状態です。HiRDB Datareplicator を使用する場合にこの状態になります。
予約	この状態のファイルはシステムログの出力対象になりません。	

3.1.3 HiRDB 管理者がすることは

HiRDB を稼働するとシステムログが出力されます。一つのシステムログファイルの容量一杯にシステムログが出力されると，スワップ先にできる状態のファイルにシステムログの出力先が変更されます。このとき，現用ファイルはスワップ先にできない状態に，スワップ先にできる状態のファイルは現用になります。これをシステムログファイルのスワップといいます。このように HiRDB を稼働すると，システムログファイルの状態が変わっていきます。

HiRDB 管理者は，スワップ先にできる状態のファイルが常にあるようにシステムログファイルを運用してください。スワップ先にできる状態のファイルがないときにスワップ処理が発生すると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。

なお，HiRDB では次の機能を用意しています。

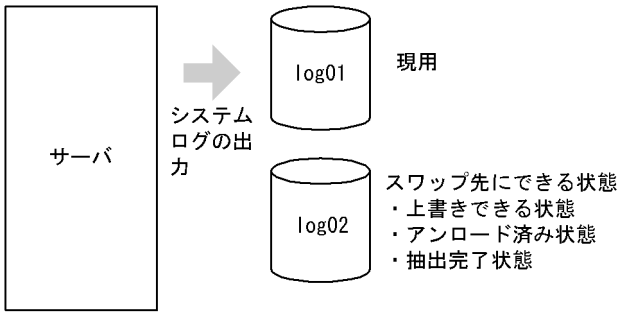
- システムログファイルの空き容量を監視する機能（システムログファイルの空き容量監視機能）
  システムログファイルの空き容量監視機能については，「システムログファイルの空き容量監視機能の運用方法」を参照してください。
- 自動的にシステムログファイルの容量を拡張する機能（システムログファイルの自動拡張機能）
  システムログファイルの自動拡張機能については，「システムログファイルの自動拡張機能の運用方法」を参照してください。

3.1.4 システムログファイルの状態の変わり方

HiRDB の稼働時，次に示すようにシステムログファイルの状態が変わります。

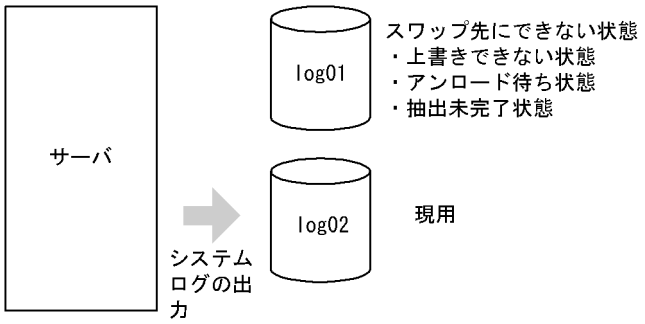
(1)　HiRDB を正常開始します

HiRDB を正常開始すると，pdlogadfg -d sys オペランドで指定したシステムログファイルのうち，**ONL** と指定したファイルがすべてオープンされます。オープンされたファイルのうち，最初に指定したファイルが現用となります。そのほかはスワップ先にできる状態になります。オープン処理に失敗したファイル，及び ONL と指定しなかったファイルは予約の状態になります。なお，HiRDB を再開始した場合は前回稼働時の現用ファイルが引き継がれます。



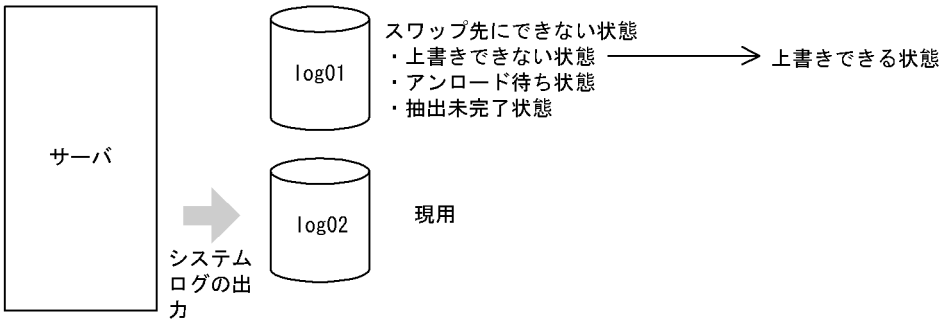
(2)　システムログファイルがスワップするとファイルの状態が変わります

現用ファイルが一杯になると，スワップ先にできる状態のファイルに出力先が変わります（システムログファイルがスワップします）。ファイルの状態は次のように変わります。



(3)　シンクポイントダンプが有効化されるとファイルの状態が変わります

システムログファイルがスワップすると，HiRDB はシンクポイントダンプの有効化処理を実施します。シンクポイントダンプが有効化されると，シンクポイントダンプ有効化以前に取得されたシステムログは，HiRDB の再開始のときに不要になります。ファイルに格納されているすべてのシステムログが不要になると，そのファイルは上書きできない状態から，上書きできる状態に変わります。



なお，実行中のトランザクションがあると，そのトランザクションが終了するまで，シンクポイントダンプは有効化されません。次に示すような長時間掛かるトランザクションの実行中は，そのトランザクションが終了するまでシンクポイントダンプが有効化されないため，ほかのトランザクションとの同時実行を避けてください。

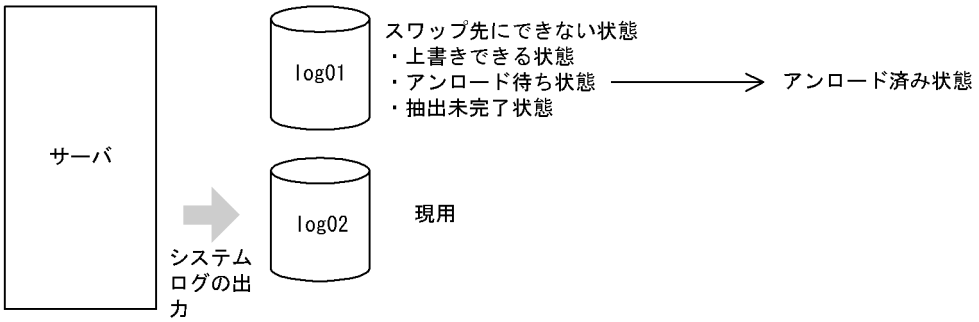
- 表への行挿入（INSERT），データ更新（UPDATE），又は行削除（DELETE）などの更新系 SQL を実行してからコミットまでの時間が長いトランザクション
- 大量のデータを対象としたデータベース作成ユティリティ（pdload），データベース再編成ユティリティ（pdrorg），又は空きページ解放ユティリティ（pdreclaim）を実行するトランザクション
- 大量のデータが格納されている表に対する CREATE INDEX を実行するトランザクション

(4) システムログをアンロードするか，又はシステムログファイルを解放するとファイルの状態が変わります

ここで説明する操作は HiRDB 管理者が行います。

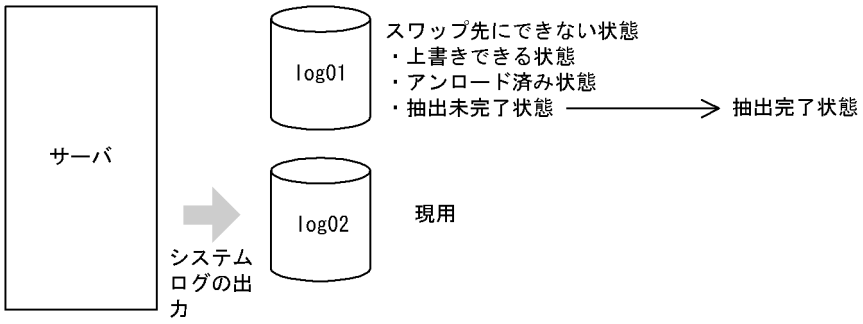
アンロード待ち状態のファイルに格納されているシステムログをアンロードすると，そのファイルはアンロード待ち状態からアンロード済み状態に変わります。ここでアンロードしたシステムログは，データベースを回復するときに使用されます。

アンロードレスシステムログ運用をする場合は，**pdlogchg -z** コマンドでシステムログファイルを解放すると，ファイルの状態がアンロード待ち状態からアンロード済み状態に変わります。



(5)　抽出側 HiRDB Datareplicator がシステムログの抽出を完了するとファイルの状態が変わります

抽出側 HiRDB Datareplicator がシステムログの抽出を完了すると，ファイルの状態が抽出未完了状態から抽出完了状態に変わります。



注意事項

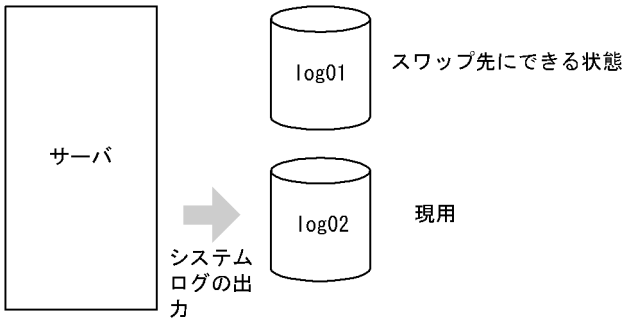
HiRDB ではシンクポイントダンプを取得するときに，抽出状態を確認します。つまり，抽出側 HiRDB Datareplicator がシステムログを完全に読み出した後，次のシンクポイントで抽出完了状態になります。

(6)　これでファイルの状態がスワップ先にできる状態になります

ファイルの状態が次に示す状態になったため，スワップ先にできない状態からスワップ先にできる状態に変わります。

- 上書きできる状態
- アンロード済み状態
- 抽出完了状態

HiRDB 管理者は，このようにスワップ先にできる状態のファイルを常に作るようにシステムログファイルを運用してください。



3.1.5　システムログファイルの運用方法を選択しましょう

システムログファイルの運用方法には，次の表に示す方法があります。HiRDB 管理者はどれかの運用方法を選択してください。

表 3–2　システムログファイルの運用方法

運用の種類	運用方法	データベースの回復方法
システムログをアンロードする運用※	• アンロード待ち状態のファイルをアンロードします。 • 詳細については，「システムログをアンロードする運用」を参照してください。	• バックアップ及びアンロードログファイルを入力情報にしてデータベースを回復します。 • データベースは，バックアップ取得時点，及びバックアップ取得時点以降の任意の同期点に回復できます。
アンロードレスシステムログ運用	• アンロード待ち状態のファイルを解放します（アンロードする必要はありません）。 • サーバ単位にバックアップを取得する必要があります。 • 詳細については，「アンロードレスシステムログ運用」を参照してください。	• バックアップ及びバックアップ取得以降のシステムログを入力情報にしてデータベースを回復します。 • データベースは，バックアップ取得時点，及びバックアップ取得時点以降の任意の同期点に回復できます。
アンロード状態のチェックを解除する運用	• アンロード待ち状態がなくなります。したがって，システムログをアンロードする必要はありません。 • 詳細については，「アンロード状態のチェックを解除する運用」を参照してください。	• バックアップを入力情報にしてデータベースを回復します。 • データベースは，バックアップ取得時点にしか回復できません。

注※

系切り替え機能を使用する場合は，アンロードログファイルを現用系及び予備系で共用できるようにする必要があります。

3.1.6　システムログファイルに障害が発生したとき

システムログファイルに障害が発生したときの対処方法については，「システムログファイルに障害が発生したときの対処方法」を参照してください。

なお，システムログファイルの障害対策で，よくある質問をＱ＆Ａにまとめました。システムログファイルのＱ＆Ａについては，「システムログファイルに関する質問」を参照してください。

3.1.7　システムログファイルを操作するコマンド

システムログファイルを操作するコマンドを次の表に示します。HiRDB 管理者は，これらのコマンドを使用してシステムログファイルを操作します。

表 3–3　システムログファイルを操作するコマンド

コマンド名	説明
pdloginit	システムログファイルを初期設定します。
pdlogls	システムログファイルの情報を表示します。
pdlogunld	• システムログファイルの内容をアンロードログファイルにアンロードします。 • システムログファイルの状態をアンロード待ち状態からアンロード済み状態にします。
pdlogchg	システムログファイルの状態を強制的に，次に示す状態にします。 • アンロード済み状態 • 抽出完了状態
pdlogswap	システムログファイルをスワップさせます。現用ファイルをスワップ先にできない状態にします。
pdlogopen	クローズ状態のシステムログファイルをオープンします。 予約ファイルを上書きできる状態にします。
pdlogcls	オープン状態のシステムログファイルをクローズします。 上書きできる状態のファイルを予約にします。
pdlogrm	システムログファイルを削除します。

3.2　システムログをアンロードする運用

実行者　HiRDB 管理者

データベースの回復に必要な情報（システムログ）をアンロードログファイルにアンロードします。データベースに障害が発生した場合，このアンロードログファイルを入力情報にして，データベース回復ユティリティでデータベースを回復します。したがって，HiRDB 管理者は，アンロード待ち状態のファイルを常にアンロードする必要があります。

システムをログアンロードする運用は，次の二つの方法があります。通常は，自動ログアンロード機能を使用することを推奨します。

自動ログアンロード機能を用いて，自動でアンロードする

システムログファイルを自動でアンロードしてアンロードファイルを作成する機能があります。これを自動ログアンロード機能といいます。自動ログアンロード機能については，「自動ログアンロード機能の運用方法」を参照してください。

pdlogunld コマンドを用いて，手動でアンロードする

pdlogunld コマンドを実行して，システムログファイルをアンロードし，アンロードファイルを作成します。

> **注意事項**
>
> 系切り替え機能を使用する場合は，アンロードログファイルを現用系及び予備系で共用できるようにする必要があります。

3.2.1　HiRDB/シングルサーバの場合

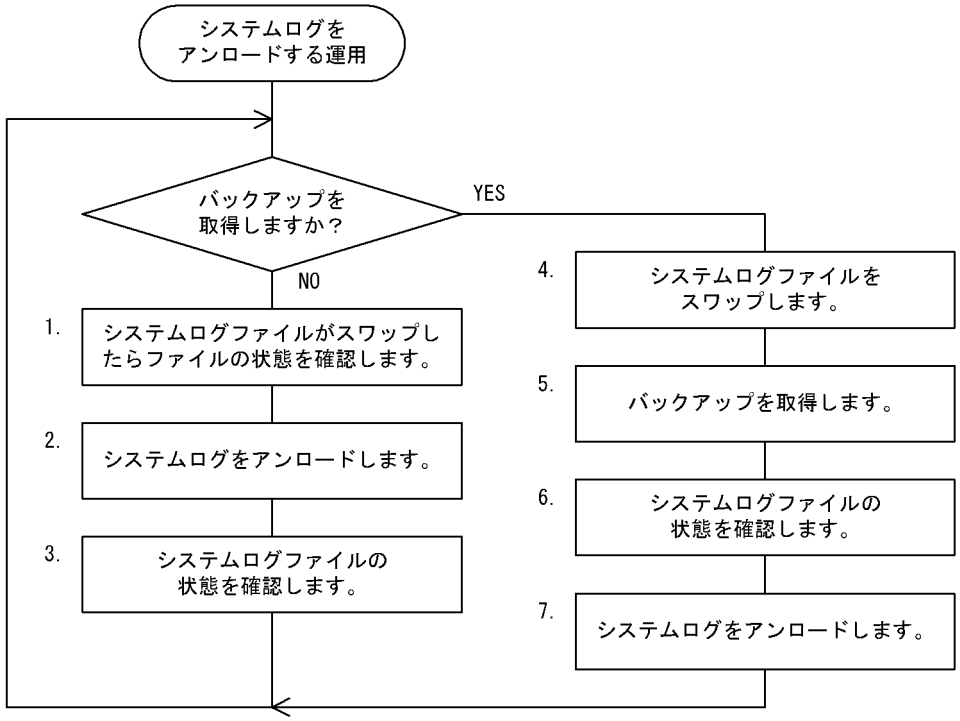
データベースに障害が発生した場合の回復方法

システムログをアンロードする運用中にデータベースに障害が発生した場合，バックアップ及びアンロードログファイル（バックアップ取得以降のシステムログを格納したアンロードログファイル）を入力情報にしてデータベースを回復します。データベースの回復方法については，「データベースの回復方法」を参照してください。

運用例

HiRDB/シングルサーバの場合のシステムログをアンロードする運用の手順を次の図に示します。

図 3–1　システムログをアンロードする運用の手順（HiRDB/シングルサーバの場合）

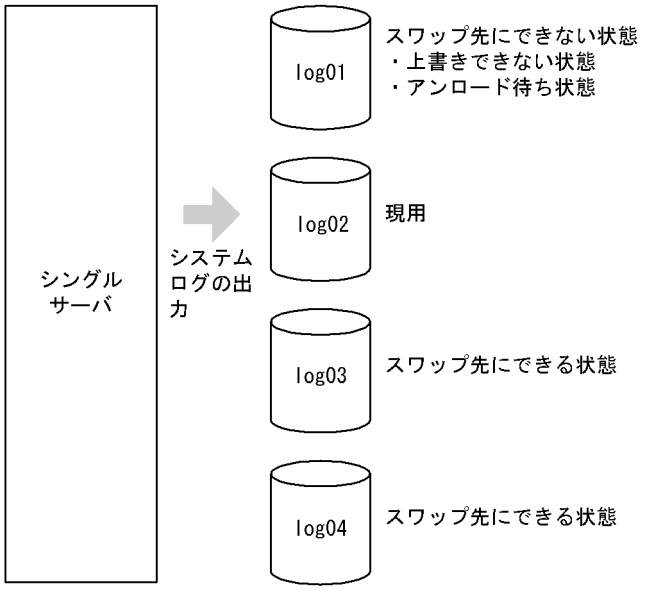


注１　処理ボックスの左にある数字は，この後で説明している（　）レベルに対応しています。
　　例えば，５．の操作は（５）で説明しています。
注２　１～３の操作は，システムログファイルがスワップするたびに繰り返し実行してください。

(1)　システムログファイルがスワップしたらファイルの状態を確認します

システムログファイルにシステムログが一杯に書き込まれると，システムログファイルがスワップします。システムログファイルがスワップすると，KFPS01221-I 及び KFPS01222-I メッセージがメッセージログファイル及びイベントログに出力されます。このとき，**pdlogls** コマンドでシステムログファイルの状態を確認してください。

```
pdlogls -d sys
```



〔説明〕

log01 のファイルにシステムログが一杯に書き込まれたため，システムログの出力先が log01 から log02 に変更されます（システムログファイルがスワップします）。このため，log01 は次に示す状態となります。

- 上書きできない状態
- アンロード待ち状態

(2) システムログをアンロードします

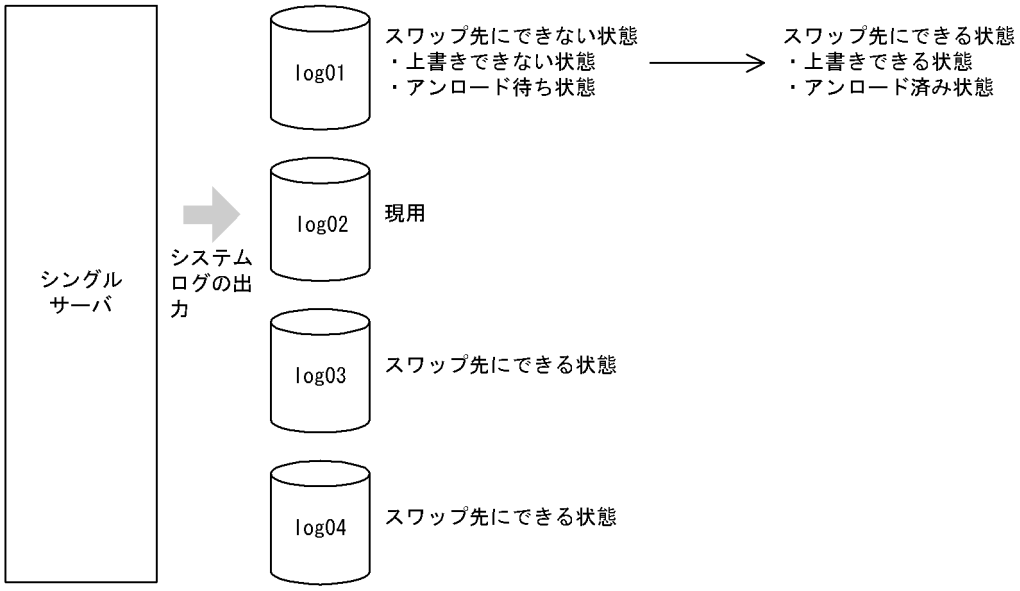
pdlogunld コマンドで，アンロード待ち状態のファイル（log01）をアンロードしてください。

```
pdlogunld -d sys -g log01 -o C:¥unld¥unldlog01
```

(3) システムログファイルの状態を確認します

pdlogls コマンドで，システムログファイル（log01）の状態を確認してください。

```
pdlogls -d sys
```



〔説明〕

- システムログをアンロードしたため，ファイルの状態がアンロード待ち状態からアンロード済み状態に変わります。
- システムログファイルがスワップすると，シンクポイントダンプの有効化処理が実施されます。シンクポイントダンプが有効化されると，ファイルの状態が上書きできない状態から上書きできる状態に変わります。
- したがって，ファイルの状態がスワップ先にできない状態からスワップ先にできる状態に変わります。

重要

スワップ先にできる状態のファイルがないときにシステムログファイルのスワップ処理が発生すると，HiRDB/シングルサーバが異常終了します。したがって，HiRDB 管理者はスワップ先にできる状態のファイルが常にあるように運用してください。なお，スワップ先にできる状態のファイルがなくなると，HiRDB は KFPS01224-I メッセージをメッセージログファイル及びイベントログに出力します。

(4) システムログファイルをスワップします

バックアップを取得する前に，**pdlogswap** コマンドでシステムログファイルをスワップさせてください。データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。データベース回復時に必要なシステムログを格納しているシステムログファイルは，これ以降現用になったシステムログファイルです。

```
pdlogswap -d sys -w
```

(5) バックアップを取得します

pdcopy コマンド（データベース複写ユティリティ）で，全 RD エリアを対象としたバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードには，ｒ又はｓを指定します。

-a：全 RD エリアのバックアップを取得する指定をします。

-b：バックアップファイル名を指定します。

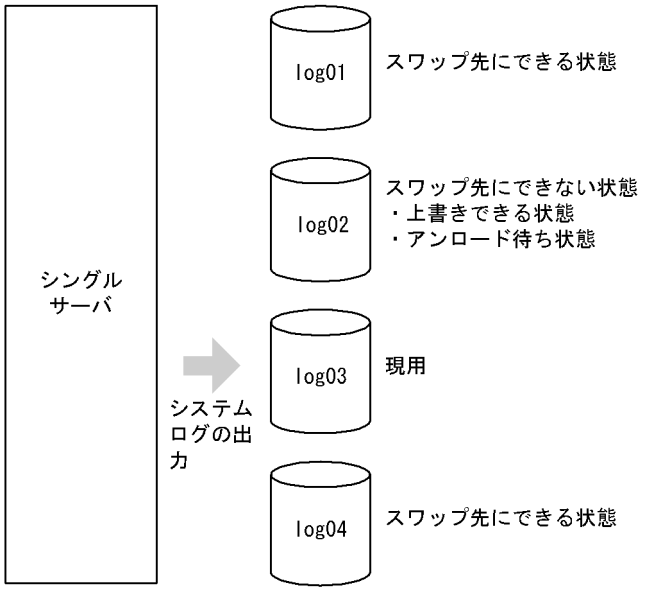
重要

ここで取得したバックアップを使用して RD エリアを回復する場合，データベース回復ユティリティの入力情報となるアンロードログファイルは，現在の現用ファイル以降のシステムログを格納したアンロードログファイルとなります。

(6)　システムログファイルの状態を確認します

pdlogls コマンドで，システムログファイルの状態を確認してください。

```
pdlogls -d sys
```



(7)　システムログをアンロードします

アンロード待ち状態のファイル（log02）を **pdlogunld** コマンドでアンロードしてください。

```
pdlogunld -d sys -g log02 -o C:¥unld¥unldlog02
```

3.2.2　HiRDB/パラレルサーバの場合

データベースに障害が発生した場合の回復方法

システムログをアンロードする運用中にデータベースに障害が発生した場合，バックアップ及びアンロードログファイル（バックアップ取得以降のシステムログを格納したアンロードログファイル）を入力情報にしてデータベースを回復します。データベースの回復方法については，「データベースの回復方法」を参照してください。

フロントエンドサーバのシステムログについて

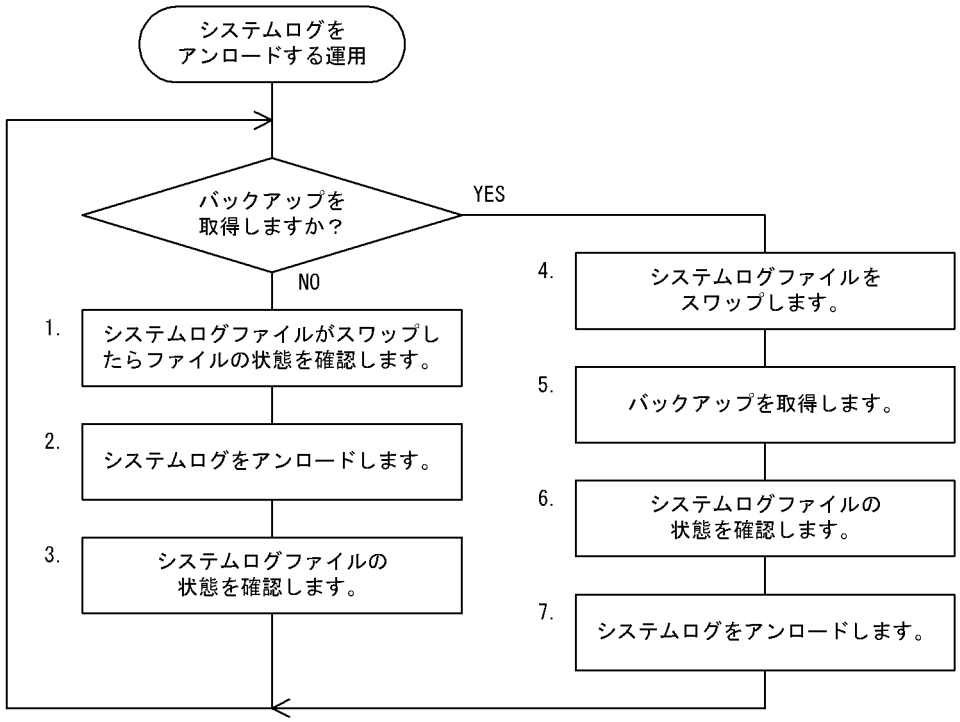
フロントエンドサーバのシステムログは，データベースの回復作業に必要ないため，アンロードする必要はありません。したがって，フロントエンドサーバ定義に **pd_log_unload_check ＝ N** を指定して，フロントエンドサーバのシステムログファイルのアンロード状態のチェックを解除してください。そうすると，フロントエンドサーバのシステムログファイルをアンロードしなくても良くなります。
pd_log_unload_check ＝ N を指定しない場合は，フロントエンドサーバのアンロード待ち状態のファイルに対して，次に示すどちらかの操作が必要になります。

- pdlogunld コマンドで，システムログをアンロードしてファイルの状態をアンロード済み状態にします。
- pdlogchg コマンドで，ファイルの状態を強制的にアンロード済み状態にします。

運用例

HiRDB/パラレルサーバの場合のシステムログをアンロードする運用の手順を次の図に示します。

図 3–2　システムログをアンロードする運用の手順（HiRDB/パラレルサーバの場合）

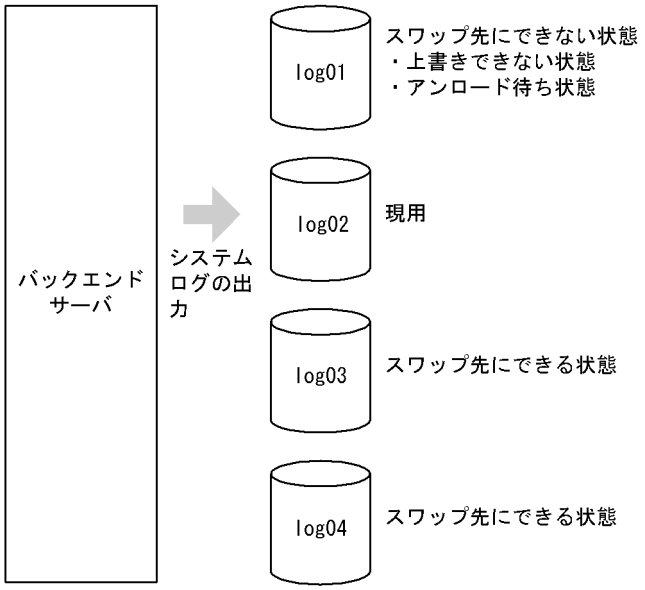


注１　処理ボックスの左にある数字は，この後で説明している（　）レベルに対応しています。
　　例えば，５．の操作は（５）で説明しています。
注２　１～３の操作は，システムログファイルがスワップするたびに繰り返し実行してください。

(1)　システムログファイルがスワップしたらファイルの状態を確認します

システムログファイルにシステムログが一杯に書き込まれると，システムログファイルがスワップします。システムログファイルがスワップすると，KFPS01221-I 及び KFPS01222-I メッセージがメッセージログファイル及びイベントログに出力されます。このとき，**pdlogls** コマンドでシステムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1
```



〔説明〕

log01 のファイルにシステムログが一杯に書き込まれたため，システムログの出力先が log01 から log02 に変更されます（システムログファイルがスワップします）。このため，log01 は次に示す状態となります。

- 上書きできない状態
- アンロード待ち状態

(2)　システムログをアンロードします

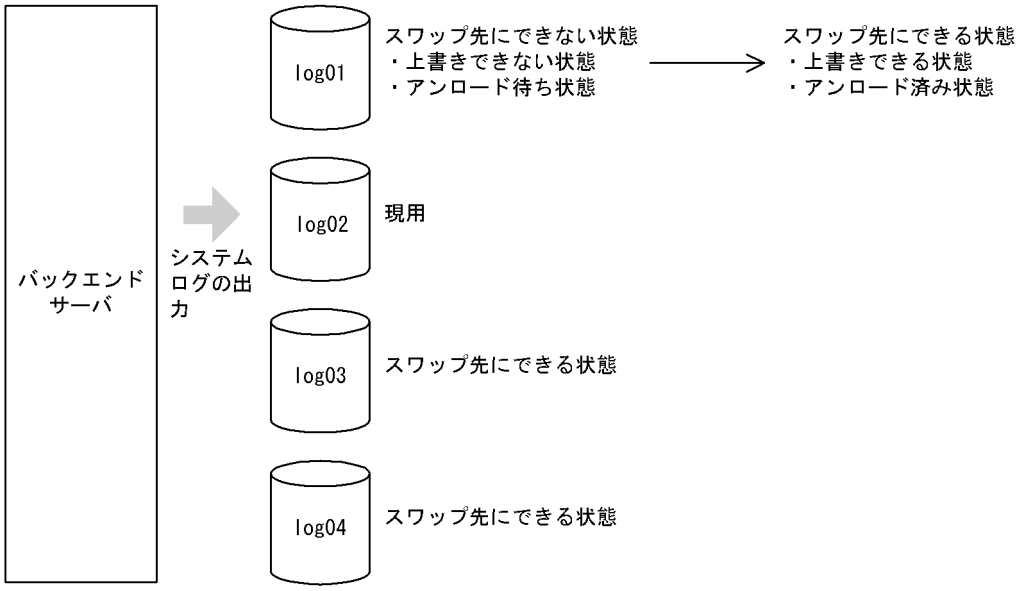
アンロード待ち状態のファイル（log01）を **pdlogunld** コマンドでアンロードしてください。

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog01
```

(3)　システムログファイルの状態を確認します

pdlogls コマンドで，システムログファイル（log01）の状態を確認してください。

```
pdlogls -d sys -s bes1
```



〔説明〕

- システムログをアンロードしたため，ファイルの状態がアンロード待ち状態からアンロード済み状態に変わります。
- システムログファイルがスワップすると，シンクポイントダンプの有効化処理が実施されます。シンクポイントダンプが有効化されると，ファイルの状態が上書きできない状態から上書きできる状態に変わります。
- したがって，ファイルの状態がスワップ先にできない状態からスワップ先にできる状態に変わります。

重要

スワップ先にできる状態のファイルがない状態でシステムログファイルがスワップすると，ユニットが異常終了します。したがって，HiRDB 管理者はスワップ先にできる状態のファイルが常にあるように運用してください。なお，スワップ先にできる状態のファイルがなくなると，HiRDB は KFPS01224-I メッセージをメッセージログファイル及びイベントログに出力します。

(4) システムログファイルをスワップします

バックアップを取得する前に，**pdlogswap** コマンドでシステムログファイルをスワップさせてください。この例ではバックエンドサーバ（bes1）下の RD エリアのバックアップを取得するので，bes1 のシステムログファイルをスワップさせます。

データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。データベース回復時に必要なシステムログを格納しているシステムログファイルは，これ以降現用になったシステムログファイルです。

```
pdlogswap -d sys -s bes1 -w
```

(5)　バックアップを取得します

pdcopy コマンド（データベース複写ユティリティ）で，サーバ単位のバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -s bes1 -b C:¥pdcopy¥backup01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードには，ｒ又はｓを指定します。

-s：バックエンドサーバ（bes1）下の全 RD エリアのバックアップを取得する指定をします。

-b：バックアップファイル名を指定します。

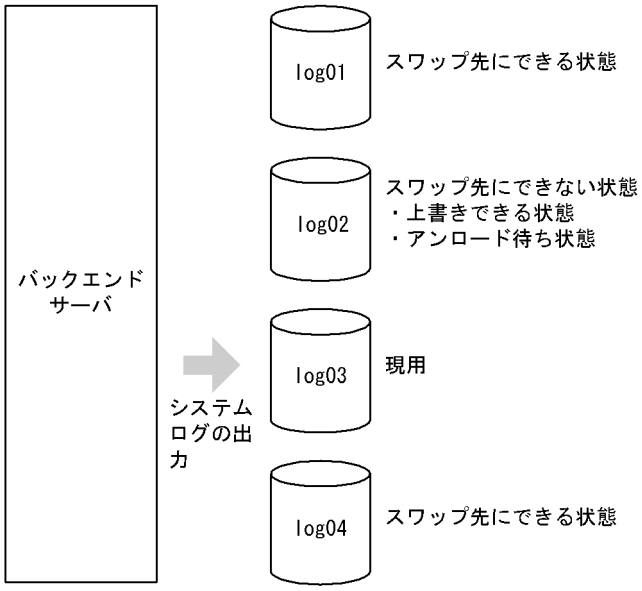
重要

ここで取得したバックアップを使用して RD エリアを回復する場合，データベース回復ユティリティの入力情報となるアンロードログファイルは，現在の現用ファイル以降のシステムログを格納したアンロードログファイルとなります。

(6)　システムログファイルの状態を確認します

pdlogls コマンドで，バックエンドサーバ（bes1）のシステムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1
```



(7)　システムログをアンロードします

アンロード待ち状態のファイル（log02）を **pdlogunld** コマンドでアンロードしてください。

```
pdlogunld -d sys -s bes1 -g log02 -o C:¥unld¥unldlog02
```

3.3　アンロードレスシステムログ運用

実行者　HiRDB 管理者

システムログのアンロード運用では，データベースの障害発生に備えてシステムログを常にアンロードする必要があります。しかし，この運用は CPU 及び入出力の負荷が大きく，HiRDB 管理者の運用も煩雑となります。システムログをアンロードしなくても，データベースの障害発生時に，直接システムログをデータベース回復ユティリティの入力情報としてデータベースを回復できます。この運用方法をアンロードレスシステムログ運用といいます。アンロードレスシステムログ運用には，次に示す利点があります。

利点

- システムログのアンロード操作がなくなるため，CPU 及び入出力の負荷を削減でき，HiRDB 管理者の運用を簡易化できます。
- アンロードログファイルを保管するためのディスク容量が必要なくなります。
- HiRDB 管理者はアンロード待ち状態のファイルをアンロードする必要はありません。ただし，アンロードする代わりに定期的にデータベースのバックアップを取得する必要があります。そして，そこで取得したログポイント情報ファイルでシステムログファイルを解放する必要があります。

ログポイントという概念があります

アンロードレスシステムログ運用には，ログポイントという概念があります。

データベースに障害が発生してデータベースを回復する場合，バックアップ取得時点よりも前のシステムログは必要ありません。このデータベースの回復に必要なシステムログと必要のないシステムログを区別する位置を**ログポイント**といいます。ログポイントは，データベース複写ユティリティでバックアップを取得すると設定されます。

データベース複写ユティリティでバックアップを取得してログポイントを設定するときに，ログポイント情報ファイルに次の図に示すログポイント情報が出力されます。この情報は，システムログファイルを解放するときに使用します。

図 3–3　ログポイント情報

```
 # The Log Point Information
system_id                 = aaaa                ･･･  1.
unit_id                   = bbbb                ･･･  2.
type                      = ccc                 ･･･  3.
server_name               = dddddddd            ･･･  4.
server_runit              = 0xeeeeeeee          ･･･  5.
log server_runit          = 0xffffffff          ･･･  6.
filegroup                 = gggggggg            ･･･  7.
generation                = 0xhhhhhhhh          ･･･  8.
block                     = 0xiiiiiiii          ･･･  9.
header_write_counter      = 0xjjjjjjjj          ･･･ 10.
header_write_time         = 0xkkkkkkkk          ･･･ 11.
```

〔説明〕

1. HiRDB 識別子

2. ユニット識別子
3. ログ種別
4. ログ取得対象サーバ名
5. ログ取得対象サーバの LAN-ID
6. ログポイント情報を設定したログサーバの LAN-ID
7. ログポイント以降のログが格納されている先頭ファイルグループ名
8. ログポイント以降のログが格納されている先頭ファイルグループ世代番号
9. ログポイント以降のログが格納されている先頭ブロック番号
10. 現用割り当て時のヘッダ更新回数
11. 使用開始時刻

3.3.1 HiRDB/シングルサーバの場合

バックアップの取得単位について

バックアップの取得は，システム単位に実施してください（全 RD エリアを対象としてバックアップを取得します）。そこで取得したログポイント情報ファイルで，システムログファイルを解放してください。

バックアップの取得間隔について

バックアップは毎日決まった時間に取得することをお勧めします。

なお，バックアップの取得間隔はシステムログファイルの容量に依存します。システムログファイルの容量が多いほど，バックアップの取得間隔を長くできます。スワップ先にできる状態のファイルがなくならないように，バックアップの取得間隔を決めてください。

システムログファイルの個数について

全システムログファイルの個数は，１日に使用するシステムログファイルの個数の倍以上にすることをお勧めします。

例えば，１日に使用するシステムログファイルの個数が２個の場合，全システムログファイルの個数は４個以上にしてください。

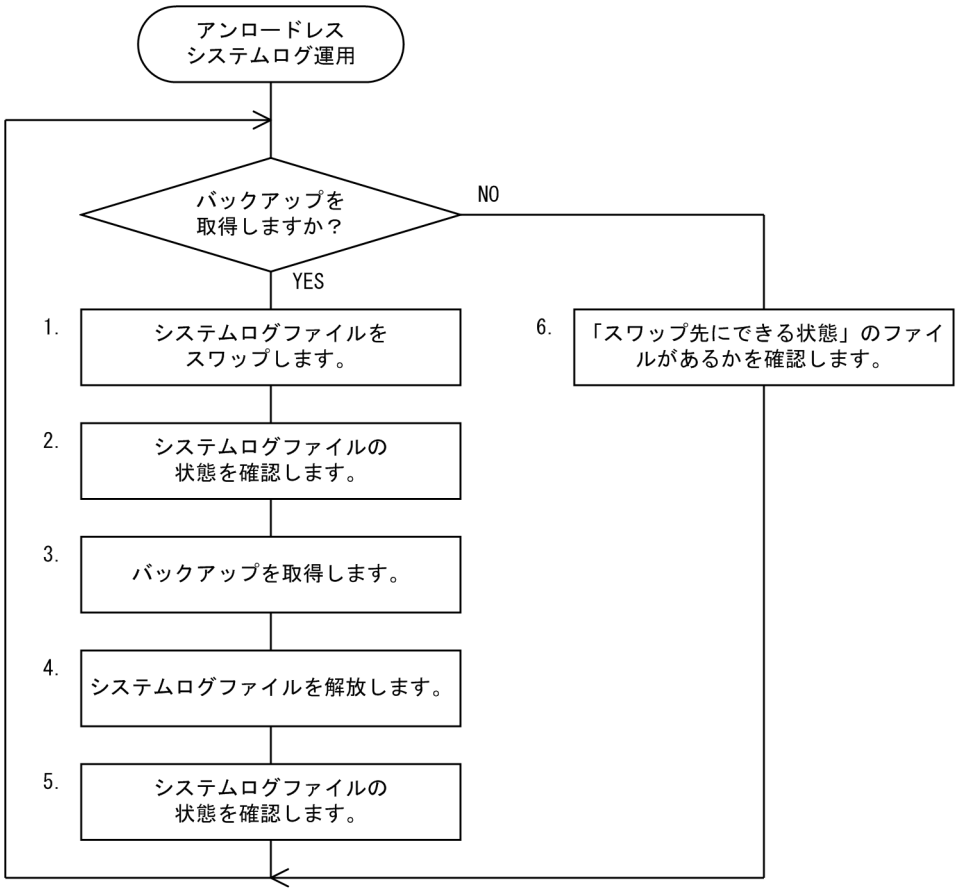
データベースに障害が発生した場合の回復方法

アンロードレスシステムログ運用中にデータベースに障害が発生した場合，バックアップ及びシステムログ（バックアップ取得以降のシステムログ）を入力情報にしてデータベースを回復します。データベースの回復方法については，「データベースの回復方法」を参照してください。

運用例

HiRDB/シングルサーバの場合のアンロードレスシステムログ運用の手順を次の図に示します。

図 3–4　アンロードレスシステムログ運用の手順（HiRDB/シングルサーバの場合）



注　処理ボックスの左にある数字は，この後で説明している（　）レベルに対応しています。
例えば，5.の操作は（5）で説明しています。

(1)　システムログファイルをスワップします

バックアップを取得する前に，**pdlogswap** コマンドでシステムログファイルをスワップさせてください。

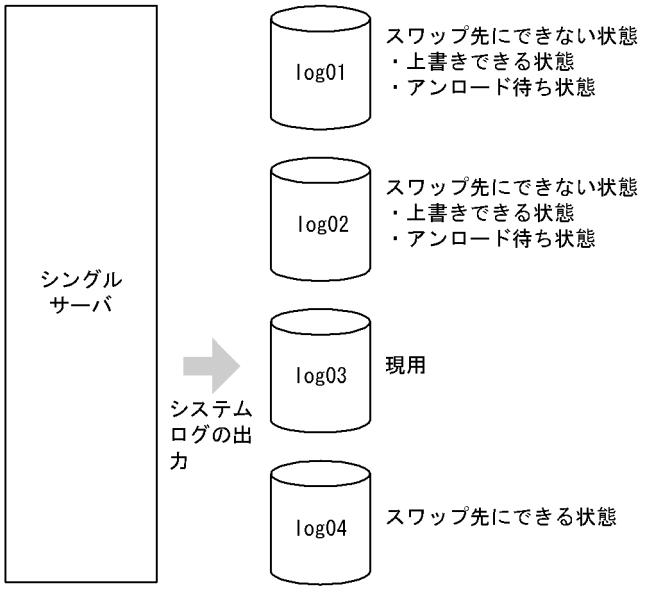
データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。データベース回復時に必要なシステムログを格納しているシステムログファイルは，これ以降現用になったシステムログファイルです。

```
pdlogswap -d sys -w
```

(2)　システムログファイルの状態を確認します

pdlogls コマンドで，システムログファイルの状態を確認してください。

```
pdlogls -d sys
```



〔説明〕

- 今日 1 日で log01 及び log02 のファイルを使用しました。したがって，両ファイルともアンロード待ち状態になっています。
- pdlogswap -w コマンドでシステムログファイルをスワップしたので，シンクポイントダンプが有効化されています。このため，log01 及び log02 のファイルは上書きできる状態になっています。

(3)　バックアップを取得します

pdcopy コマンド（データベース複写ユティリティ）で，全 RD エリアを対象としたバックアップを取得します。このとき，-z オプションを指定してログポイント情報ファイルも同時に取得してください。バックアップの取得については，「バックアップの取得方法」を参照してください。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01
-z C:¥pdcopy¥logpoint01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードには r を指定します。

-a：全 RD エリアのバックアップを取得する指定をします。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。

ログポイント情報ファイルに障害が発生した場合

ログポイント情報ファイルに障害が発生した場合，ここで取得したバックアップファイルからログポイント情報ファイルを再作成してください。**pdrstr -z** コマンド（データベース回復ユティリティ）で，ログポイント情報ファイルを再作成できます。

```
pdrstr -b C:¥pdcopy¥backup01 -z C:¥pdcopy¥logpoint01
```

(4)　システムログファイルを解放します

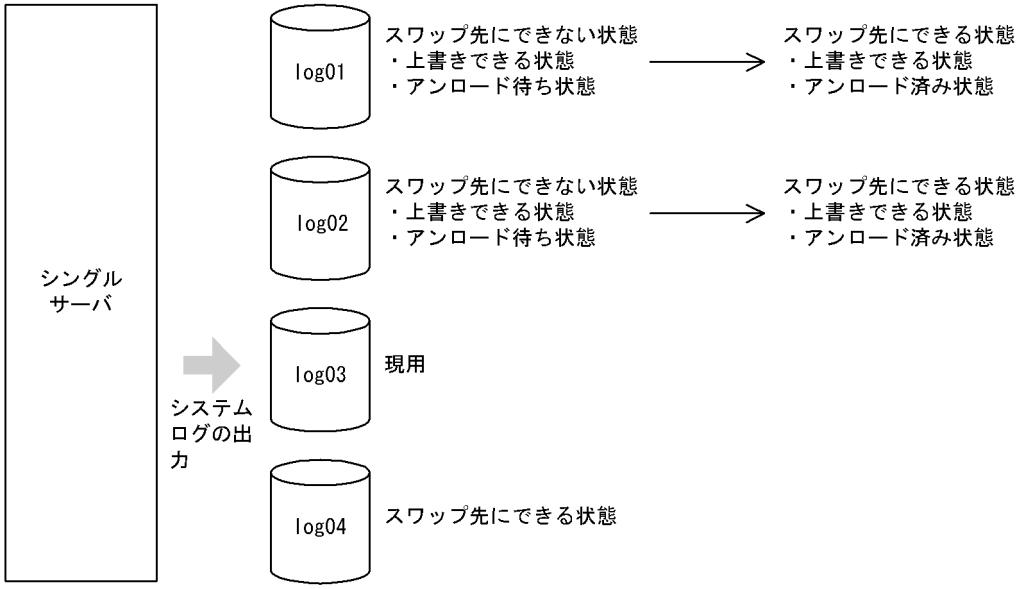
pdlogchg コマンドで，ログポイント以前のシステムログファイル（log01，log02）を解放してください。-z オプションには，(3)で取得したログポイント情報ファイルを指定します。

```
pdlogchg -z C:¥pdcopy¥logpoint01
```

(5)　システムログファイルの状態を確認します

pdlogls コマンドで，システムログファイルの状態を確認してください。

```
pdlogls -d sys
```



〔説明〕

- システムログファイルを解放したため，ファイルの状態がアンロード待ち状態からアンロード済み状態に変わります。
- したがって，ファイルの状態が上書きできる状態かつ，アンロード済み状態になったため，スワップ先にできない状態からスワップ先にできる状態に変わります。

(6)　スワップ先にできる状態のファイルがあるかを確認します

pdlogls コマンドで，スワップ先にできる状態のファイルがあるかを確認してください。

```
pdlogls -d sys
```

重要

スワップ先にできる状態のファイルがない状態でシステムログファイルがスワップすると，HiRDB/シングルサーバが異常終了します。したがって，HiRDB 管理者はスワップ先にできる状態のファイルが

常にあるように運用してください。なお，スワップ先にできる状態のファイルがなくなると，HiRDB は KFPS01224-I メッセージをメッセージログファイル及びイベントログに出力します。

3.3.2 HiRDB/パラレルサーバの場合

バックアップの取得単位について

バックアップの取得は，必ずサーバ単位（バックエンドサーバ，ディクショナリサーバごと）に実施してください。サーバごとにログポイント情報ファイルを取得して，サーバごとにシステムログファイルを解放してください。

バックアップの取得間隔について

バックアップは毎日決まった時間に取得することをお勧めします。

なお，バックアップの取得間隔はシステムログファイルの容量に依存します。システムログファイルの容量が多いほど，バックアップの取得間隔を長くできます。スワップ先にできる状態のファイルがなくならないように，バックアップの取得間隔を決めてください。

システムログファイルの個数について

各サーバのシステムログファイルの個数は，各サーバで１日に使用するシステムログファイルの個数の倍以上にすることをお勧めします。

例えば，あるバックエンドサーバで１日に使用するシステムログファイルの個数が２個の場合，そのバックエンドサーバのシステムログファイルの個数は４個以上にしてください。

フロントエンドサーバのシステムログファイルについて

フロントエンドサーバのシステムログは，データベースの回復作業に必要ありません。そこで，フロントエンドサーバ定義に **pd_log_unload_check ＝ N** を指定することをお勧めします。そうすると，フロントエンドサーバのシステムログファイルのアンロード状態のチェックが解除されます。このため，フロントエンドサーバのシステムログファイルを解放する必要がなくなります。

pd_log_unload_check ＝ N を指定しない場合は，フロントエンドサーバのアンロード待ち状態のファイルに対して，pdlogchg コマンドでファイルの状態を強制的にアンロード済み状態にする必要があります。

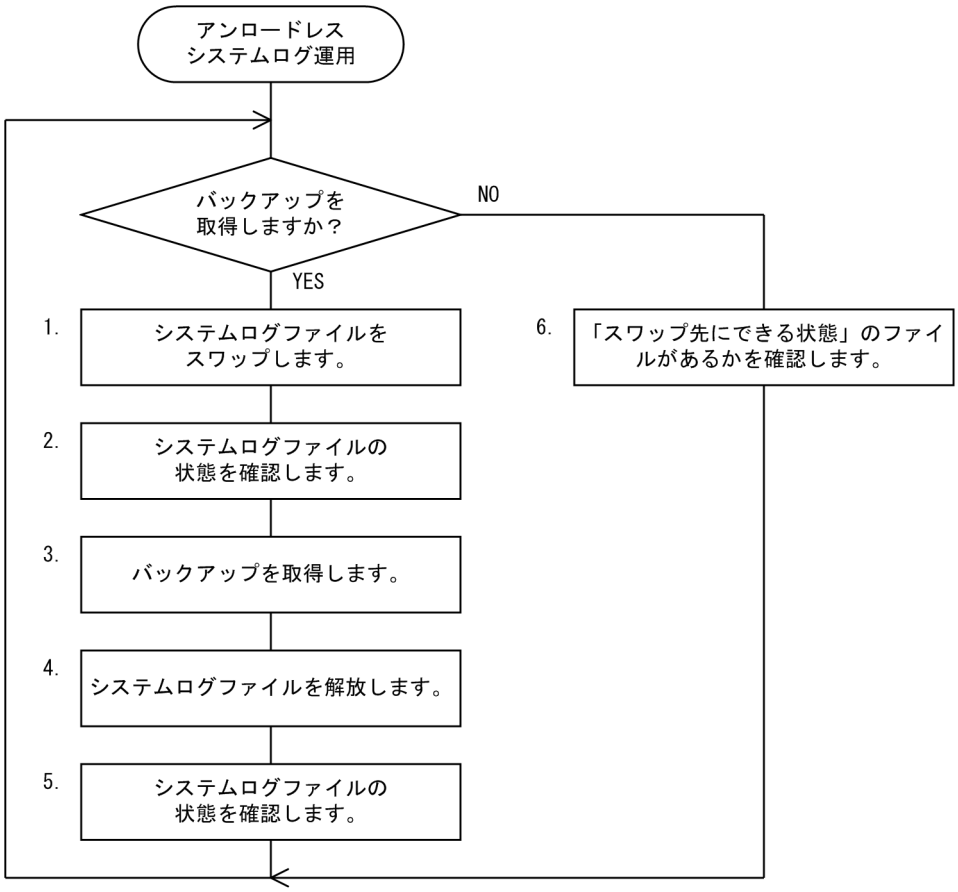
データベースに障害が発生した場合の回復方法

アンロードレスシステムログ運用中にデータベースに障害が発生した場合，バックアップ及びシステムログ（バックアップ取得以降のシステムログ）を入力情報にしてデータベースを回復します。データベースの回復方法については，「データベースの回復方法」を参照してください。

運用例

HiRDB/パラレルサーバの場合のアンロードレスシステムログ運用の手順を次の図に示します。

図 3–5　アンロードレスシステムログ運用の手順（HiRDB/パラレルサーバの場合）



注　処理ボックスの左にある数字は，この後で説明している（　）レベルに対応しています。
例えば，5.の操作は（5）で説明しています。

(1)　システムログファイルをスワップします

バックアップを取得する前に，**pdlogswap** コマンドで，バックアップを取得するサーバのシステムログファイルをスワップさせてください。

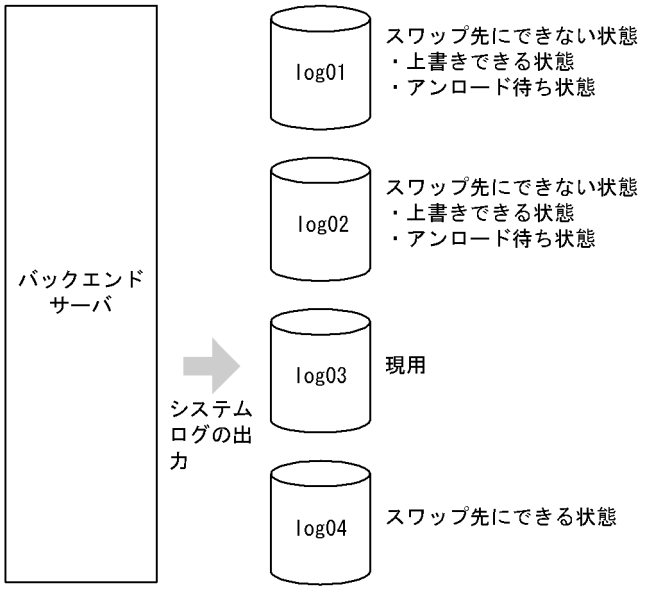
データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。データベース回復時に必要なシステムログを格納しているシステムログファイルは，これ以降現用になったシステムログファイルです。

```
pdlogswap -d sys -s bes1 -w
```

(2)　システムログファイルの状態を確認します

pdlogls コマンドで，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1
```



〔説明〕

- 今日 1 日で log01 及び log02 のファイルを使用しました。したがって，両ファイルともアンロード待ち状態になっています。
- pdlogswap -w コマンドでシステムログファイルをスワップしたので，シンクポイントダンプが有効化されています。このため，log01 及び log02 のファイルは上書きできる状態になっています。

(3) バックアップを取得します

pdcopy コマンド（データベース複写ユティリティ）でバックエンドサーバ（bes1）下の RD エリアのバックアップを取得します。このとき，-z オプションを指定してログポイント情報ファイルも同時に取得してください。バックアップの取得については，「バックアップの取得方法」を参照してください。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -s bes1 -b C:¥pdcopy¥bes1bkup01
-z C:¥pdcopy¥bes1logp01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードには r を指定します。

-s：バックエンドサーバ（bes1）下の RD エリアのバックアップを取得する指定をします。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。

ログポイント情報ファイルに障害が発生した場合

ログポイント情報ファイルに障害が発生した場合，ここで取得したバックアップファイルからログポイント情報ファイルを再作成してください。**pdrstr -z** コマンド（データベース回復ユティリティ）で，ログポイント情報ファイルを再作成できます。

```
pdrstr -b C:¥pdcopy¥bes1bkup01 -z C:¥pdcopy¥bes1logp01
```

(4)　システムログファイルを解放します

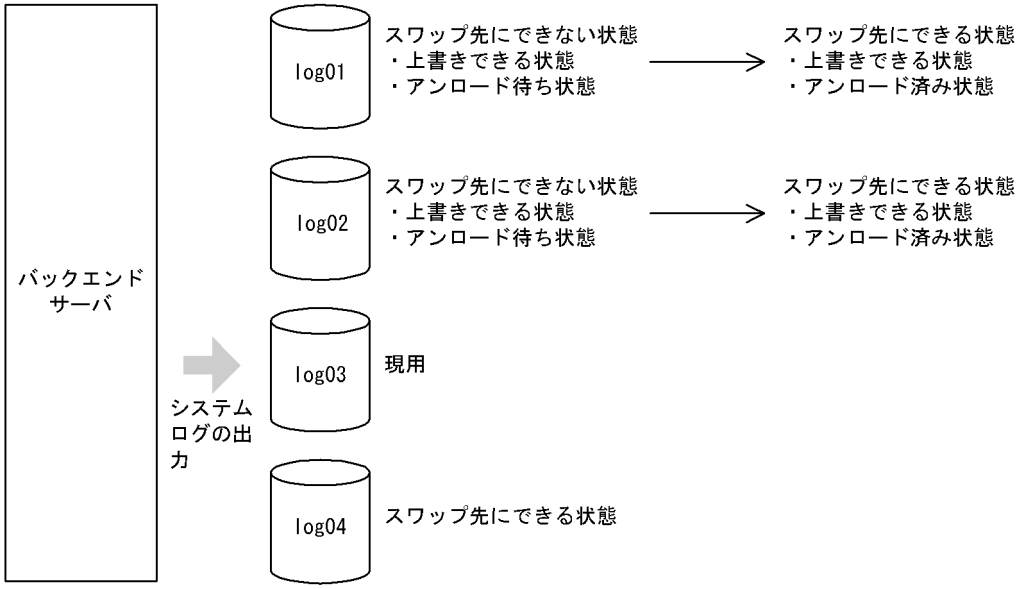
pdlogchg コマンドで，ログポイント以前のシステムログファイル（log01，log02）を解放してください。-z オプションには，(3)で取得したログポイント情報ファイルを指定します。

```
pdlogchg -z C:¥pdcopy¥bes1logp01 -x host01
```

(5)　システムログファイルの状態を確認します

pdlogls コマンドで，バックエンドサーバ（bes1）のシステムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1
```



〔説明〕

- システムログファイルを解放したため，ファイルの状態がアンロード待ち状態からアンロード済み状態に変わります。
- したがって，ファイルの状態が上書きできる状態かつ，アンロード済み状態になったため，スワップ先にできない状態からスワップ先にできる状態に変わります。

(6)　スワップ先にできる状態のファイルがあるかを確認します

pdlogls コマンドで，スワップ先にできる状態のファイルがあるかを確認してください。

```
pdlogls -d sys -s bes1
pdlogls -d sys -s bes2
pdlogls -d sys -s bes3
pdlogls -d sys -s bes4
pdlogls -d sys -s dic
```

重要

スワップ先にできる状態のファイルがない状態でシステムログファイルがスワップすると，そのユニットが異常終了します。したがって，HiRDB 管理者はスワップ先にできる状態のファイルが常にあるように運用してください。なお，スワップ先にできる状態のファイルがなくなると，HiRDB は KFPS01224-I メッセージをメッセージログファイル及びイベントログに出力します。

3.4　アンロード状態のチェックを解除する運用

実行者　HiRDB 管理者

HiRDB は，システムログファイルのアンロード状態を常にチェックしています。アンロード待ち状態のファイルは上書きできないようにしています。HiRDB 管理者は，このアンロード待ち状態のファイルをアンロードするか，又はこのファイルを解放するかして，ファイルの状態をアンロード済み状態にする必要があります。

しかし，データベースの回復にシステムログ（アンロードログ）を使用しない場合は，このシステムログのアンロード操作又は解放操作（pdlogunld コマンド又は pdlogchg コマンドの実行）はむだな作業になります。このような場合は，HiRDB が行っているアンロード状態のチェックを解除することをお勧めします。そうすれば，HiRDB 管理者のシステムログファイルのアンロード操作又は解放操作が不要になります。

3.4.1　スワップ先にできる条件が変わります

アンロード状態のチェックを解除すると，スワップ先にできる条件は次の二つだけになります。

- 上書きできる状態
- 抽出完了状態（HiRDB Datareplicator）

アンロードの状態は，システムログファイルをスワップ先にできるかどうかの条件に関係がなくなります。

3.4.2　利点

- システムログファイルのアンロード操作又は解放操作がなくなるため，運用方法が簡単になります。
- アンロードログファイルを保管するためのファイル容量が必要なくなります。

3.4.3　適用基準

主に，参照系のデータベースの場合に適用します。

例えば，次に示すように，システムログを使用しなくてもデータベースを回復できる場合にこの運用をお勧めします。

- データを再ロードするだけでデータベースを回復できる場合
- バックアップだけでデータベースを回復できる場合
- データベースのバックアップを取得した時点以降の更新については，実行した UAP，ユティリティを再実行すればデータベースを回復できる場合

HiRDB/パラレルサーバの場合

- この運用は，サーバ単位に実行できます。したがって，適用基準を満たしているバックエンドサーバごとにこの運用を適用してもかまいません。
- フロントエンドサーバのシステムログは，データベースの回復作業に必要ないのでアンロードする必要はありません。このため，フロントエンドサーバについてはこの運用を適用することをお勧めします。

3.4.4　注意事項

(1)　データベースの回復について

データベースの回復にシステムログが必要なのにこの運用をした場合は，データベースの回復手段がなくなります。

(2)　システムログファイルの容量について

更新量が多いトランザクションを実行すると，トランザクションの開始から終了までの間にすべてのシステムログファイルを使用することが考えられます。この場合，HiRDB はすべてのシステムログファイルの上書きを禁止します（上書きできない状態にします）。そうすると，現用として割り当てられるシステムログファイルがなくなるため注意してください。

(3)　pdlogunld 及び pdlogchg コマンドに制限事項が発生します

1. HiRDB の稼働中は次に示すコマンドが使用できなくなります。
   - pdlogunld（-f オプションを指定する場合だけは使用できます）
   - pdlogchg（-R オプションを指定する場合だけは使用できます）

   HiRDB の停止中はこれらのコマンドを通常通り使用できます（全オプションを指定できます）。
2. pdlogunld 又は pdlogchg コマンドの実行処理中に HiRDB を開始しないでください。開始すると，pdlogunld 又は pdlogchg コマンドがエラーになります。なお，pdlogunld コマンドがエラーになっても，アンロードログファイルが作成されることがあります。しかし，このアンロードログファイルはデータベースを回復するときに使用できません。

3.4.5　環境設定

この運用をする場合は，HiRDB システム定義の各サーバ定義に次に示す指定をしてください。

- **pd_log_unload_check ＝ N**

HiRDB/シングルサーバの場合は，シングルサーバ定義でこのオペランドを指定します。

HiRDB/パラレルサーバの場合は，サーバ共通定義で一括して指定するか，又は次に示す各サーバ定義でそれぞれ指定してください。

- フロントエンドサーバ定義
- バックエンドサーバ定義
- ディクショナリサーバ定義

3.4.6　運用方法

スワップ先にできる状態のファイルがあるかどうかを，pdlogls コマンドで確認してください。システムログファイルのアンロード操作又は解放操作は不要です。

●データベースを更新する UAP を実行するときの運用方法

データベースを更新する UAP を実行するときは，次に示す手順に従ってください。

〈手順〉

1. **pdlogswap** コマンドで，システムログファイルをスワップします。

   pdlogswap -d sys -s b001

2. **pdcopy** コマンドで RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

   pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01

3. UAP を実行します。

障害発生時，表にデータを再度格納し，UAP とその後の処理が再実行できるような場合は，上記の〈手順〉は必要ありません。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

3.4.7　システムログファイルに障害が発生した場合

システムログファイルに障害が発生すると，障害の発生したシステムログファイルは予備となり，以降の HiRDB 稼働時に現用として割り当てられません。ここでいう障害の発生したシステムログファイルとは，物理的な障害が発生したシステムログファイルのほかに，次に示すようなシステムログファイルも含まれます。

- HiRDB の異常終了後に強制正常開始（pdstart dbdestroy 又は pdstart -i）をしたときに，前回の HiRDB 稼働時に現用として使用したシステムログファイル
- 正常開始中に何らかの要因でユニットが開始できなかったときに，前回開始時に現用として割り当てられたシステムログファイル（開始完了までに現用ファイルを割り当てていたサーバに限ります）

上記のように，障害の発生したシステムログファイルは現用として割り当てられません。

なお，前回の HiRDB 稼働時に現用として割り当てていたシステムログファイルは，pdlogchg 又は pdlogunld コマンドを実行すると，再度現用ファイルとして割り当てられます。

ただし，物理的に障害の発生したシステムログファイルであっても，pdlogchg 又は pdlogunld コマンドを実行すると，次回の HiRDB 稼働時に現用ファイルの割り当て対象になります。障害の発生したシステムログファイルが現用になると，HiRDB の異常終了の原因となります。したがって，障害要因を取り除いた後に pdlogchg 又は pdlogunld コマンドを実行してください。

3.5　システムログファイルの操作方法

実行者　HiRDB 管理者

ここでは，システムログファイルの操作方法について説明します。説明する項目は次のとおりです。

1. スワップ先にできる状態のファイルを確認する場合
2. スワップ先にできる状態のファイルがない場合
3. 現用ファイルをアンロードする場合
4. アンロード済み状態のファイルをアンロードする場合
5. アンロード待ち状態のファイルに格納されているシステムログが必要ない場合
6. ファイルの状態を変更する場合
7. HiRDB の稼働中にシステムログファイルの容量を大きく（小さく）する場合
8. システムログファイルを新規追加する場合
9. システムログファイルを削除する場合

3.5.1　スワップ先にできる状態のファイルを確認する場合

pdlogls コマンドで，スワップ先にできる状態のファイルがあるかどうかを確認できます。スワップ先にできる状態のファイルがない場合はスワップ先にできる状態のファイルを追加してください。追加方法については，「スワップ先にできる状態のファイルがない場合」を参照してください。

スワップ先にできる状態のファイルがない状態でシステムログファイルのスワップが発生すると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。なお，スワップ先にできる状態のファイルがなくなると，HiRDB はその旨のメッセージを出力します。

3.5.2　スワップ先にできる状態のファイルがない場合

pdlogls コマンドで，上書きできる状態のファイルがあるかどうかを確認してください。

(1)　上書きできる状態のファイルがある場合

上書きできる状態のファイルがある場合は，そのファイルをアンロード済み状態にしてスワップ先にできる状態のファイルを作成してください。アンロード待ち状態のファイルをアンロード済み状態にするには，次に示す二つの方法があります。

- **pdlogunld** コマンドでシステムログファイルをアンロードします。
- **pdlogchg** コマンドでシステムログを破棄します。

ただし，システム及びデータベースの回復に必要なシステムログを格納しているファイルに対して，pdlogchg コマンドを実行するときは注意してください。そのファイルが上書きされると，システム及びデータベースの回復に必要なシステムログがなくなり，障害発生時にデータベースが回復できなくなることがあります。

なお，HiRDB Datareplicator を使用している場合は抽出未完了状態を解除する必要があります。この状態を解除しないとスワップ先にできる状態になりません。

(2)　上書きできる状態のファイルがない場合

システムログファイルを追加してください。システムログファイルの追加方法については，「システムログファイルを新規追加する場合」を参照してください。

3.5.3　現用ファイルをアンロードする場合

データベースの回復処理などで現用ファイルをアンロードする場合は，**pdlogunld -f** コマンドを実行してください。ただし，pdlogunld -f コマンドを実行しても，ファイルの状態は変わりません。

3.5.4　アンロード済み状態のファイルをアンロードする場合

アンロード済み状態のファイルをアンロードする場合は，次に示す手順でアンロードを実行してください。

〈手順〉

1. **pdlogcls** コマンドでアンロード対象のファイルをクローズします（予約状態にします）。
   pdlogcls -d sys -s b001 -g syslog01
2. **pdlogunld -f** コマンドでアンロードを実行します。なお，ファイルの状態は変わりません。
   pdlogunld -d sys -s b001 -g syslog01 -f

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

3.5.5　アンロード待ち状態のファイルに格納されているシステムログが必要ない場合

アンロード待ち状態のファイルに格納されているシステムログが必要ない場合は，**pdlogchg** コマンドでファイルの状態をアンロード待ち状態からアンロード済み状態に変えてください。

3.5.6　ファイルの状態を変更する場合

(1)　現用ファイルを変更する場合

現用ファイルを変更する場合は，**pdlogswap** コマンドでシステムログファイルをスワップさせます。現用ファイルはスワップ先にできない状態になり，スワップ先にできる状態のファイルの一つが現用ファイルになります。現用ファイル中のシステムログが必要な場合にこのコマンドを実行します。

なお，スワップ先にできる状態のファイルがない場合は pdlogswap コマンドを実行できません。

(2)　予約ファイルを上書きできる状態にする場合

pdlogopen コマンドで予約ファイルを上書きできる状態にできます。ただし，HiRDB を正常開始した場合は上書きできない状態になります。

(3)　上書きできる状態のファイルを予約にする場合

pdlogcls コマンドで上書きできる状態のファイルを予約にできます。ただし，上書きできる状態のファイルが一つしかない場合は，そのファイルを予約にできません。また，上書きできない状態のファイルは，ファイル中にシステムの回復に必要なシステムログを含んでいるため，予約にできません。

(4)　抽出未完了状態のファイルを抽出完了状態にする場合

pdlogchg -R コマンドで抽出未完了状態のファイルを抽出完了状態にできます。強制的にファイルの状態を変更すると，HiRDB Datareplicator が必要なシステムログを抽出できなくなり，反映側データベースとの同期が取れない，又はデータ連動ができないことがあります。障害発生によってシステムログファイルの再作成が必要なときなどに pdlogchg -R コマンドを実行し，それ以外の場合は pdlogchg -R コマンドを実行しないでください。

3.5.7　HiRDB の稼働中にシステムログファイルの容量を大きく（小さく）する場合

次に示す手順でシステムログファイルの容量を大きくしてください。

〈手順〉

1. **pdlogls** コマンドでシステムログファイルの状態を確認します。
   pdlogls -d sys -s b001
2. **pdlogcls** コマンドで，スワップ先にできる状態のファイルを予約にします。
   pdlogcls -d sys -s b001 -g syslog01
3. **pdlogrm** コマンドで予約ファイルを削除します。
   pdlogrm -d sys -s b001 -f C:¥sysfile01¥syslog1a

pdlogrm -d sys -s b001 -f C:¥sysfile01¥syslog1b

4. **pdloginit** コマンドで，3 で削除したシステムログファイルを再作成します。

   このとき，レコード数を変更前のシステムログファイルより大きくしてください。ファイル容量を小さくする場合はレコード数を小さくしてください。

   pdloginit -d sys -s b001 -f C:¥sysfile01¥syslog1a -n 5000

   pdloginit -d sys -s b001 -f C:¥sysfile01¥syslog1b -n 5000

5. **pdlogopen** コマンドで，4 で作成したファイルをスワップ先にできる状態にします。

   pdlogopen -d sys -s b001 -g syslog01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

注意事項

スワップ先にできる状態のファイルを予約に変更するとき，すべてのスワップ先にできる状態のファイルを予約に変更しないでください。スワップ先にできる状態のファイルがないときにシステムログファイルのスワップが発生すると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。したがって，スワップ先にできる状態のファイルを一つ以上残した状態でシステムログファイルの容量を変更してください。

3.5.8　システムログファイルを新規追加する場合

システムログファイルを新規追加する手順を次に示します。なお，ファイルを追加する前に，マニュアル「HiRDB システム導入・設計ガイド」で，システムログファイルの設計について読むことをお勧めします。

(1)　HiRDB を正常終了できる場合

〈手順〉

1. **pdfstatfs** コマンドで，システムログファイルを作成する HiRDB ファイルシステム領域に空きがあるかを確認します。

   pdfstatfs C:¥sysfile01

   空きがない場合は HiRDB ファイルシステム領域を新規作成してください。HiRDB ファイルシステム領域の作成方法については，「HiRDB ファイルシステム領域を作成（初期設定）する方法」を参照してください。

2. **pdstop** コマンドで HiRDB を正常終了します。

3. HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 4 で追加するシステムログファイルを指定します。

   ・pdlogadfg -d sys オペランド

   ・pdlogadpf -d sys オペランド

4. **pdloginit** コマンドでシステムログファイルを追加（初期設定）します。

   pdloginit -d sys -s b001 -f C:¥sysfile01¥syslog1a -n 5000

   pdloginit -d sys -s b001 -f C:¥sysfile01¥syslog1b -n 5000

5. **pdconfchk** コマンドで HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

6. **pdstart** コマンドで HiRDB を正常開始します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) HiRDB を正常終了できない場合

HiRDB システム定義を変更するときにシステム構成変更コマンド（pdchgconf コマンド）を使用します。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

〈手順〉

1. **pdfstatfs** コマンドで，システムログファイルを作成する HiRDB ファイルシステム領域に空きがあるかを確認します。

   pdfstatfs C:¥sysfile01

   空きがない場合は HiRDB ファイルシステム領域を新規作成してください。HiRDB ファイルシステム領域の作成方法については，「HiRDB ファイルシステム領域を作成（初期設定）する方法」を参照してください。

2. %PDDIR%¥conf¥chgconf ディレクトリを作成します。

3. 使用中の HiRDB システム定義ファイルを 2 で作成したディレクトリ下にコピーします。

4. %PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 5 で追加するシステムログファイルを指定します。

   ・pdlogadfg -d sys オペランド

   ・pdlogadpf -d sys オペランド

5. **pdloginit** コマンドでシステムログファイルを追加（初期設定）します。

   pdloginit -d sys -s b001 -f C:¥sysfile01¥syslog1a -n 5000

   pdloginit -d sys -s b001 -f C:¥sysfile01¥syslog1b -n 5000

6. **pdconfchk** コマンドで，%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

7. **pdchgconf** コマンドで，HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

備考

ここで説明する方法でシステムログファイルを追加する場合，pdchgconf コマンドの実行中にシステムログファイルがスワップします。スワップ先にできる状態のファイルがない場合は，追加したシステムログファイルにスワップします。

3.5.9　システムログファイルを削除する場合

pdlogrm コマンドで予約ファイルを削除できます。予約以外の状態のシステムログファイルは削除できません。

必要に応じて次に示すオペランドを削除してください。pdlogrm コマンドで削除したシステムログファイルに対応するオペランドを削除します。

- pdlogadfg -d sys オペランド
- pdlogadpf -d sys オペランド

これらのオペランドを削除しないと，削除したシステムログファイルは実体のないファイルになります。

3.6　システムログファイルの状態遷移

HiRDB 稼働中のシステムログファイルの状態遷移を次の表に示します。

- 表「HiRDB 稼働中のシステムログファイルの状態遷移（システムログファイルの状態が 1～9）」
- 表「HiRDB 稼働中のシステムログファイルの状態遷移（システムログファイルの状態が 10～17）」

備考

- 表中の状態遷移はイベントが正常に処理されることを前提としています。
- HiRDB は稼働中のときだけシステムログファイルの状態を管理しています。
- 予約ファイルに対して pdlogunld 及び pdlogchg コマンドを実行しても，pdlogls コマンドで表示されるシステムログファイルの状態は変化しません。システムログファイルの状態を知りたい場合は，pdlogopen コマンドを実行した後に pdlogls コマンドを実行してください。

表 3–4　HiRDB 稼働中のシステムログファイルの状態遷移（システムログファイルの状態が 1～9）

イベント			システムログファイルの状態							
			現用	待機（アンロード待ち状態）				待機（アンロード済み状態）			
				上書きできる状態		上書きできない状態		上書きできる状態		上書きできない状態	
				完了	未完	完了	未完	完了	未完	完了	未完
			1	2	3	4	5	6	7	8	9
システムログファイルの容量満杯でスワップ発生			→5	－	－	－	－	→1	－	－	－
現用ファイルの障害でスワップ発生	非二重化で障害発生，又は二重化でＡ系とＢ系共に障害発生		→13	－	－	－	－	→1	－	－	－
	二重化で片系に障害発生	片系運転可	→5	－	－	－	－	→1	－	－	－
		片系運転不可	→13	－	－	－	－	→1	－	－	－
pdlogswap コマンドの実行			→5	－	－	－	－	→1	－	－	－
シンクポイントダンプの有効化			－	－	－	－	－	－	－	→6	→7
自動ログアンロード機能によるアンロード完了			－	→6	→7	→8	→9	－	－	－	－
HiRDB Datareplicator による全データ抽出完了			－	－	→2	－	→4	－	→6	－	→8
pdlogunld コマンドの実行			－	→6	→7	→8	→9	－	－	－	－

イベント	システムログファイルの状態								
	現用	待機（アンロード待ち状態）				待機（アンロード済み状態）			
		上書きできる状態		上書きできない状態		上書きできる状態		上書きできない状態	
		完了	未完	完了	未完	完了	未完	完了	未完
	1	2	3	4	5	6	7	8	9
pdlogchg コマンドの実行	−	→6	→7	→8	→9	−	−	−	−
pdlogchg -R の実行	−	−	→2	−	→4	−	→6	−	→8
pdlogopen コマンドの実行※	−	−	−	−	−	−	−	−	−
pdlogcls コマンドの実行※	−	→10	→11	→12	→13	→14	→15	→16	→17

（凡例）

完了：抽出完了状態

未完：抽出未完了状態

−：該当しない，又は状態が遷移しないことを示しています。

→n：イベント後のシステムログファイルの状態を示しています。

例えば，→1 の場合はイベント後にシステムログファイルの状態は現用になります。

注※　-a 及び-b オプションの指定を省略した場合に適用されます。

表 3–5　HiRDB 稼働中のシステムログファイルの状態遷移（システムログファイルの状態が 10〜17）

イベント			システムログファイルの状態							
			予約かつアンロード待ち状態				予約かつアンロード済み状態			
			上書きできる状態		上書きできない状態		上書きできる状態		上書きできない状態	
			完了	未完	完了	未完	完了	未完	完了	未完
			10	11	12	13	14	15	16	17
システムログファイルの容量満杯でスワップ発生			−	−	−	−	−	−	−	−
現用ファイルの障害でスワップ発生	非二重化で障害発生，又は二重化でＡ系とＢ系共に障害発生		−	−	−	−	−	−	−	−
	二重化で片系に障害発生	片系運転可	−	−	−	−	−	−	−	−
		片系運転不可	−	−	−	−	−	−	−	−

イベント	システムログファイルの状態							
	予約かつ アンロード待ち状態				予約かつ アンロード済み状態			
	上書きできる状態		上書きできない状態		上書きできる状態		上書きできない状態	
	完了	未完	完了	未完	完了	未完	完了	未完
	10	11	12	13	14	15	16	17
pdlogswap コマンドの実行	−	−	−	−	−	−	−	−
シンクポイントダンプの有効化	−	−	→10	→11	−	−	→14	→15
自動ログアンロード機能によるアンロード完了	→14	→15	→16	→17	−	−	−	−
HiRDB Datareplicator による全データ抽出完了	−	→10	−	→12	−	→14	−	→16
pdlogunld コマンドの実行	→14	→15	→16	→17	−	−	−	−
pdlogchg コマンドの実行	→14	→15	→16	→17	−	−	−	−
pdlogchg -R の実行	−	→10	−	→12	−	→14	−	→16
pdlogopen コマンドの実行※	→2	→3	→4	→5	→6	→7	→8	→9
pdlogcls コマンドの実行※	−	−	−	−	−	−	−	−

（凡例）

完了：抽出完了状態

未完：抽出未完了状態

−：該当しない，又は状態が遷移しないことを示しています。

→n：イベント後のシステムログファイルの状態を示しています。

例えば，→1 の場合はイベント後にシステムログファイルの状態は現用になります。

注※　-a 及び-b オプションの指定を省略した場合に適用されます。

3.7　システムログファイルのレコード長の変更方法

実行者　HiRDB 管理者

システムログファイルのレコード長を変更できます。選択できるレコード長は 1024，2048，4096 バイトのどれかを選択できます。例えば，レコード長を短くすると，システムログの容量を減らせます。

なお，現在のレコード長は，pdlogls コマンドで調べられます。

注意事項

- システムログファイルのレコード長は統一してください。
- システムログファイルのレコード長に 2048 又は 4096 を選択した場合は，pd_log_rec_leng オペランドに選択したレコード長を指定してください。pd_log_rec_leng の指定と異なるレコード長のシステムログファイルはオープンされません。
- システムログファイルのレコード長を変更すると，次の点も変更になります。

  ・全システムログファイルの総容量

  ・ユニットコントローラが使用する共用メモリサイズ

  これらの見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

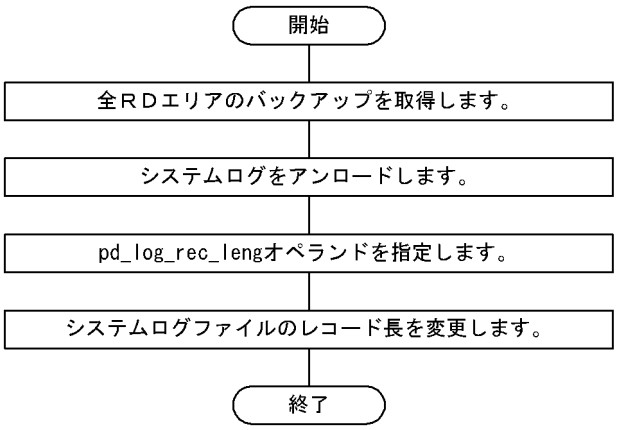
3.7.1　例題１（システムログをアンロードする運用をしている場合）

システムログファイルのレコード長を 4096 バイトから 1024 バイトに変更します。

前提条件

- HiRDB は正常終了した状態です。
- HiRDB Datareplicator とデータ連携している場合は，最新のシステムログファイルの抽出が完了した後に，HiRDB Datareplicator を終了させておいてください。

〈手順〉



(1)　pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(2)　pdcopy コマンドで全 RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -a -b C:¥pdcopy¥backup01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。pdstart -r で HiRDB を開始しているため，x を指定する場合でも，RD エリアを閉塞かつクローズ状態にする必要がありません。

-a：全 RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(4)　アンロード待ち状態のシステムログファイルの内容を pdlogunld コマンドでアンロードします

```
pdlogunld -d sys -g log1 -o C:¥unld¥unldlog1
```

(5)　pd_log_rec_leng オペランドを削除します

サーバ定義に pd_log_rec_leng オペランドを指定していれば削除します。

(6) pdlogrm コマンドでシステムログファイルを削除します

```
pdlogrm -d sys -f C:¥sysfile_a¥log1a
pdlogrm -d sys -f C:¥sysfile_b¥log1b
pdlogrm -d sys -f C:¥sysfile_a¥log2a
pdlogrm -d sys -f C:¥sysfile_b¥log2b
pdlogrm -d sys -f C:¥sysfile_a¥log3a
pdlogrm -d sys -f C:¥sysfile_b¥log3b
pdlogrm -d sys -f C:¥sysfile_a¥log4a
pdlogrm -d sys -f C:¥sysfile_b¥log4b
```

〔説明〕

システムログファイルを二重化している場合，B系のシステムログファイルも忘れずに削除してください。

(7) pdloginit コマンドでシステムログファイルを再作成します

```
pdloginit -d sys -f C:¥sysfile_a¥log1a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log1b -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log2a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log2b -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log3a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log3b -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log4a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log4b -n 2000
```

(8) pdstart コマンドで HiRDB を開始します

```
pdstart
```

HiRDB Datareplicator とデータ連携している場合は，HiRDB Datareplicator を開始してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

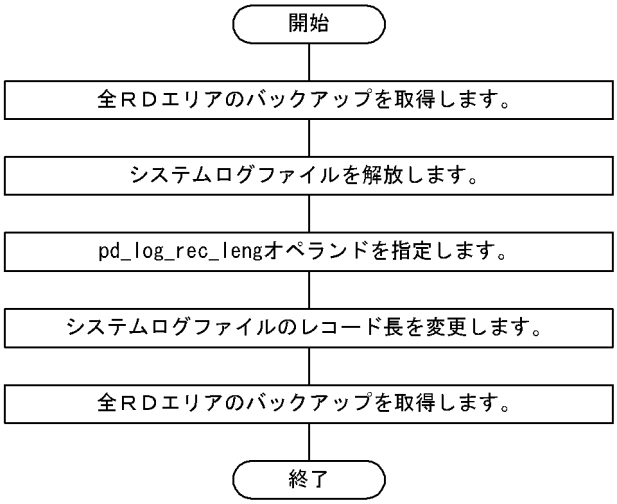
3.7.2 例題２（アンロードレスシステムログ運用をしている場合）

システムログファイルのレコード長を 4096 バイトから 1024 バイトに変更します。

前提条件

- HiRDB は正常終了した状態です。
- HiRDB Datareplicator とデータ連携している場合は，最新のシステムログファイルの抽出が完了した後に，HiRDB Datareplicator を終了させておいてください。

〈手順〉



(1)　pdstart コマンドで HiRDB を開始します

```
pdstart
```

(2)　pdcopy コマンドで全 RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01
-z C:¥pdcopy¥logpoint01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。HiRDB/パラレルサーバの場合は，-a 指定でバックアップを取得できません。-s 指定でサーバ単位に全 RD エリアのバックアップを取得してください。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(4)　アンロード待ち状態のシステムログファイルを pdlogchg コマンドで解放します

```
pdlogchg -d sys -g log1 -z C:¥pdcopy¥logpoint01
```

(5) pd_log_rec_leng オペランドを削除します

サーバ定義に pd_log_rec_leng オペランドを指定していれば削除します。

(6) pdlogrm コマンドでシステムログファイルを削除します

```
pdlogrm -d sys -f C:¥sysfile_a¥log1a -u
pdlogrm -d sys -f C:¥sysfile_b¥log1b -u
pdlogrm -d sys -f C:¥sysfile_a¥log2a -u
pdlogrm -d sys -f C:¥sysfile_b¥log2b -u
pdlogrm -d sys -f C:¥sysfile_a¥log3a -u
pdlogrm -d sys -f C:¥sysfile_b¥log3b -u
pdlogrm -d sys -f C:¥sysfile_a¥log4a -u
pdlogrm -d sys -f C:¥sysfile_b¥log4b -u
```

〔説明〕

- 強制削除のオプション-u を指定してください。
- システムログファイルを二重化している場合，B系のシステムログファイルも忘れずに削除してください。

(7) pdloginit コマンドでシステムログファイルを再作成します

```
pdloginit -d sys -f C:¥sysfile_a¥log1a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log1b -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log2a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log2b -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log3a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log3b -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log4a -n 2000
pdloginit -d sys -f C:¥sysfile_b¥log4b -n 2000
```

(8) pdstart コマンドで HiRDB を開始します

```
pdstart
```

HiRDB Datareplicator とデータ連携している場合は，HiRDB Datareplicator を開始してください。

(9) pdcopy コマンドで全 RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02
-z C:¥pdcopy¥logpoint02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。HiRDB/パラレルサーバの場合は，-a 指定でバックアップを取得できません。-s 指定でサーバ単位に全 RD エリアのバックアップを取得してください。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

3.8　自動ログアンロード機能の運用方法

実行者　HiRDB 管理者

システムログをアンロードする運用では，pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードする必要があります。この作業を忘れるとスワップ先にできる状態のシステムログファイルがなくなり，HiRDB が異常終了します。

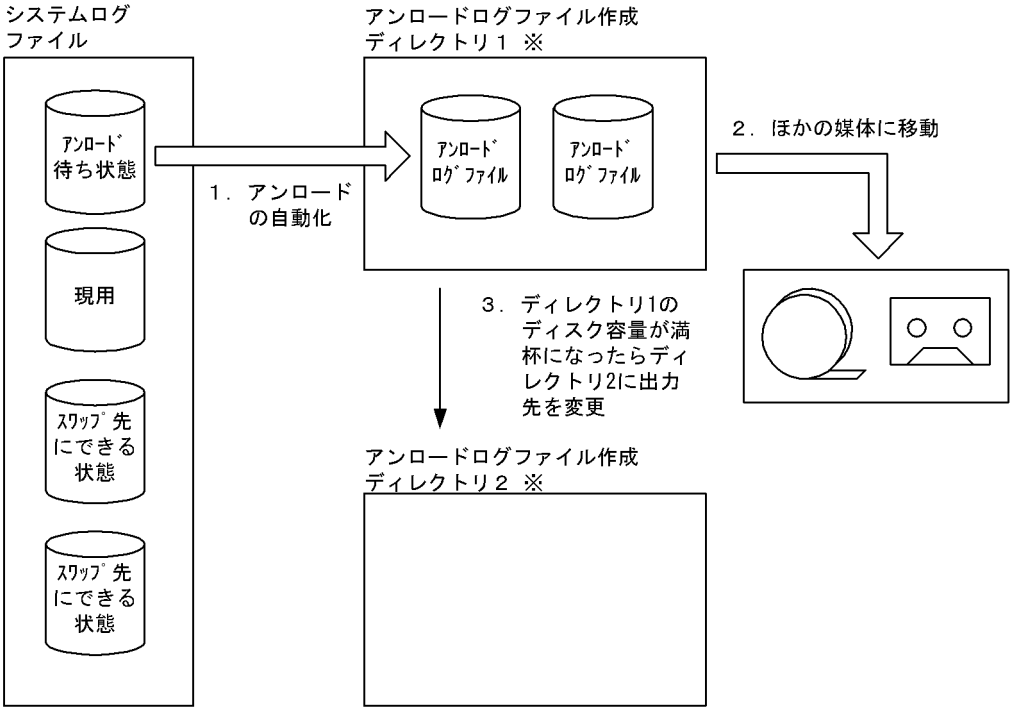
HiRDB ではシステムログファイルのアンロード作業を自動化する機能（自動ログアンロード機能）を提供しています。

なお，ここでの説明はシステムログをアンロードする運用を理解していることを前提にしています。

3.8.1　自動ログアンロード機能とは

アンロード待ち状態のシステムログファイルを HiRDB が自動的にアンロードする機能をシステムログファイルの**自動ログアンロード機能**といいます。HiRDB 管理者が事前に作成したディレクトリ（これを**アンロードログファイル作成ディレクトリ**といいます）下にアンロードログファイルが作成されます。自動ログアンロード機能を次の図に示します。

図 3–6　自動ログアンロード機能



注※ アンロードログファイルをHiRDBファイルシステム領域に作成することもできます。

〔説明〕

1. アンロード待ち状態のシステムログファイルができると，自動的にシステムログのアンロードが行われます。アンロードログファイルはアンロードログファイル作成ディレクトリ下に作成されます。
2. HiRDB 管理者は定期的にアンロードログファイルを別の媒体に移動してください。
3. アンロードログファイル作成ディレクトリを二つ以上作成した場合，ディレクトリのディスク容量が満杯になったら，ほかのディレクトリにアンロードログファイルの出力先を切り替えます。

(1) HiRDB 管理者がすること

ディスク容量満杯又はディスク障害などで，アンロードログファイル作成ディレクトリにアンロードログファイルを作成できない場合，自動ログアンロード機能を停止します。したがって，**HiRDB 管理者は定期的にアンロードログファイル作成ディレクトリ内のアンロードログファイルを別の媒体に移動してください。また，不要なアンロードログファイルを削除してください。**

なお，アンロードログファイル作成ディレクトリを二つ以上作成した場合は，全ディレクトリでアンロードログファイルを作成できないときに自動ログアンロード機能を停止します。

(2) ほかの機能との関係

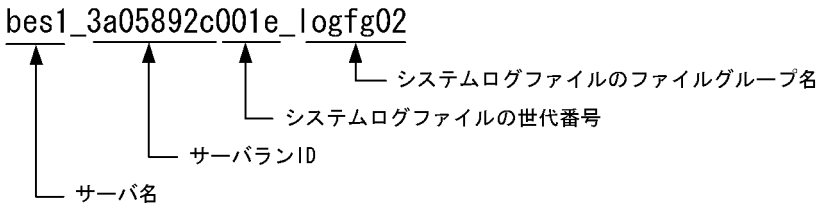
システムログファイルのアンロード状態を HiRDB がチェックしない場合（pd_log_unload_check = N を指定している場合）は，自動ログアンロード機能は使用できません。

(3) 作成されるアンロードログファイルの名称

自動ログアンロード機能で作成されるアンロードログファイルの名称は，次に示す形式になります。なお，作成されたアンロードログファイルの名称は，KFPS01212-I メッセージで表示されます。

サーバ名_サーバラン ID と世代番号_ファイルグループ名称

（例）



自動ログアンロード機能で作成されるアンロードログファイルの名称は，pdlogatul コマンドで確認できます。

(4)　アンロードログファイル作成ディレクトリを二つ以上作成した場合

ディスク容量満杯又はディスク障害などで，アンロードログファイル作成ディレクトリ内にアンロードログファイルを作成できない場合，ほかのディレクトリ下にアンロードログファイルを作成します。このとき，HiRDB は pd_log_auto_unload_path オペランドに指定した順番にディレクトリを使用します。

例えば，pd_log_auto_unload_path ＝"ディレクトリ 1"，"ディレクトリ 2"，"ディレクトリ 3"，...と指定した場合，HiRDB はディレクトリ 1，ディレクトリ 2，ディレクトリ 3，...の順番にディレクトリを使用します。ただし，ディレクトリ 2 がディスク障害などで使用できない場合，ディレクトリ 1 の次にディレクトリ 3 を使用します。

アンロードログファイルの作成失敗でディレクトリが切り替わる場合は，KFPS01155-W（reason=1667）メッセージを出力します。

アンロードログファイルの作成が失敗してもディレクトリが切り替わらない場合は，KFPS01155-W（reason=1667）メッセージを出力しないで，自動ログアンロード機能を停止します。

なお，HiRDB の正常開始時はディレクトリ 1 から使用します。ただし，これはディレクトリ 1 にアンロードログファイルがない場合です。ディレクトリ 1 にアンロードログファイルがある場合はディレクトリ 2 を使用します。

HiRDB の再開始時（システムログファイルのスワップも含む）は HiRDB 終了時に使用していたディレクトリを使用します。

注意事項

アンロードログファイル作成ディレクトリを二つ以上作成している場合の注意事項を次に示します。

- HiRDB を正常開始したときに空のディレクトリがないと自動ログアンロード機能を停止します。

3.8.2　環境設定

自動ログアンロード機能を使用するには，次に示す環境設定が必要です。

- アンロードログファイル作成ディレクトリの作成
- pd_log_auto_unload_path オペランドの指定
- pd_log_auto_unload_restart オペランドの指定（障害時にリトライさせる場合）

(1)　アンロードログファイル作成ディレクトリの作成

HiRDB 管理者がアンロードログファイル作成ディレクトリを作成してください。ディレクトリを作成する前に，空きディスク容量を確認してください。空きディスク容量が少ないと，アンロードログファイルの作成によってディスク容量不足が発生することがあります。

また，作成するディレクトリには，HiRDB 管理者に対して読み込み権限，書き込み権限，及び実行権限を与えてください。

アンロードログファイルを HiRDB ファイルシステム領域に作成する場合は，ユティリティ用の HiRDB ファイルシステム領域を作成してください。

なお，HiRDB/パラレルサーバの場合は，サーバごとに異なるディレクトリを作成してください。

(a) 必要な空きディスク容量

必要な空きディスク容量を次に示します。

```
必要なディスク容量（単位：バイト）
  ＝（アンロードするシステムログファイルの総レコード数※1
    ×システムログファイルのレコード長）※2
    ×ディレクトリ内に作成するアンロードログファイル数×1.2
```

注

- ここで計算する空きディスク容量は 1 ディレクトリの容量ではありません。ディレクトリを二つ以上作成する場合は，全ディレクトリの空きディスク容量です。
- 一つのディレクトリではディスク容量が不足する場合は，ディレクトリを二つ以上作成してください。

注※1

システムログファイルの自動拡張機能を適用している場合，pd_log_auto_expand_size オペランドの拡張上限サイズに指定した値で計算してください。

注※2

システムログファイルの概算値です。

(b) アンロードログファイル作成ディレクトリを幾つ作成するか

アンロードログファイル作成ディレクトリは最大 128 個作成できますが，多く作成することはお勧めしません。管理が大変にならないよう多くても 4 個ぐらいにすることをお勧めします。1 個だと管理が楽になりますが，ディスク容量満杯又はディスク障害などが発生した場合，自動ログアンロード機能が停止します。

ディレクトリを二つ以上作成しておけば，すべてのディレクトリでアンロードログファイルの作成が失敗しないかぎり自動ログアンロード機能が停止しません。このとき，異なるパーティションにディレクトリを作成してください。

(c) 注意事項（重要）

- マルチ HiRDB の場合は，HiRDB ごとに異なるディレクトリを作成してください。同じディレクトリを指定すると，どのアンロードログファイルがどの HiRDB に対応しているか分からなくなる可能性があります。
- ディレクトリ内にアンロードログファイル以外のファイルを作成しないでください。アンロードログファイル以外のファイルを作成した場合，そのファイルを HiRDB が削除することがあります。

(d) アンロードログファイルを HiRDB ファイルシステム領域に作成する場合

アンロードログファイルを HiRDB ファイルシステム領域に作成する場合は，pdfmkfs コマンドで HiRDB ファイルシステム領域を作成してください。pdfmkfs コマンドに指定するオプションの目安を次の表に示します。

表 3–6 pdfmkfs コマンドに指定するオプションの目安（アンロードログファイルを HiRDB ファイルシステム領域に作成する場合）

pdfmkfs コマンドのオプション	指定値の目安
-k	使用目的には UTL を指定します。
-n	HiRDB ファイルシステム領域長には次に示す計算式の値を指定します。 アンロードするシステムログファイルの総レコード数×システムログファイルのレコード長×作成するアンロードログファイル数×1.2÷1048576
-l	最大ファイル数には，作成するアンロードログファイル数を指定します。
-e	最大増分回数には，作成するアンロードログファイル数×23 を指定します。

(2) pd_log_auto_unload_path オペランドの指定

作成したアンロードログファイル作成ディレクトリを **pd_log_auto_unload_path** オペランドに指定してください。

アンロードログファイルを HiRDB ファイルシステム領域に作成する場合は，その HiRDB ファイルシステム領域名を指定してください。

(3) pd_log_auto_unload_restart オペランドの指定（障害時にリトライさせる場合）

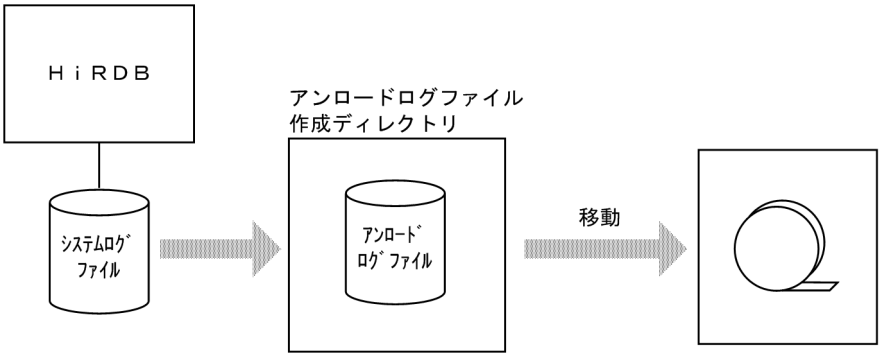
pd_log_auto_unload_restart オペランドに Y を指定，又は指定を省略すると，一時的な障害でアンロードが失敗した場合，次回のシステムログファイルのスワップを契機にアンロードをリトライします。

3.8.3 運用例 1（アンロードログファイル作成ディレクトリが一つの場合）

アンロードログファイル作成ディレクトリが一つの場合は，次に示すような運用が考えられます。

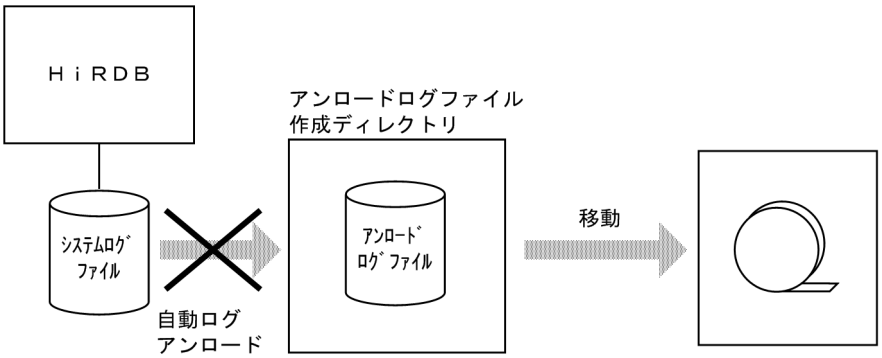
(1) 1 週間ごとにアンロードログファイルを別媒体に移動する運用

アンロードログファイル作成ディレクトリに 1 週間分のアンロードログファイルが格納できる場合，1 週間ごとにアンロードログファイルを別媒体に移動します。このように定期的にアンロードログファイルを別媒体に移動します。



(2)　自動ログアンロード機能を停止して，アンロードログファイルを別媒体に移動する運用

自動ログアンロード機能を停止して，自動作成したアンロードログファイルをディレクトリ単位に移動することで HiRDB を連続稼働させます。



この運用をする場合の手順を次に示します。

＜手順＞

1. pdlogatul -t -w コマンドを実行して，自動ログアンロード機能を停止します。
2. ディレクトリ内の全アンロードログファイルを別の媒体に移動します。
3. 別の媒体へのデータ移動が正常に完了したかどうか，次の 2 点を確認します。

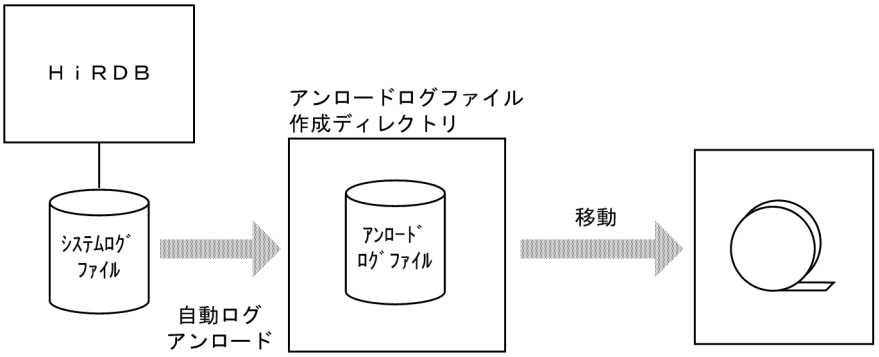
   ・表「自動ログアンロード機能の使用時に障害が発生したときの対策方法」にあるメッセージが出力されていないこと

   ・pdlogucat コマンドを実行し，対象のアンロードログファイルのシステムログのアンロードの完了状態が「completed」であること
4. データ移動が正常に完了した後，アンロードログファイル作成先ディレクトリ内にファイルが残っている場合は全ファイルを削除します。
5. pdlogatul -b コマンドを実行し，自動ログアンロード機能を再開始します。

バックアップファイルとアンロードログファイルからデータベースを回復する場合，別の媒体に移動したアンロードログファイルをいったん磁気ディスク上に回復した後，アンロードログファイルの指定順を確認し，pdrstr コマンドを実行してください。

(3)　自動ログアンロード機能を停止しないで，アンロードログファイルを別媒体に移動する運用

自動ログアンロード機能を停止しないで，自動作成したアンロードログファイルを別の媒体に移動することで，HiRDB を連続稼働させます。



この運用をする場合の手順を次に示します。

＜手順＞

1. 次のどちらかの方法で操作可能なアンロードログファイルの名称を確認してください。

   ・KFPS01212-I メッセージのアンロードログファイルの名称

   ・pdlogatul -i コマンドの実行結果に表示される現用世代のシステムログ世代番号より若い世代番号のアンロードログファイルの名称

   アンロードログファイルの名称については，「作成されるアンロードログファイルの名称」を参照してください。なお，世代番号が ffff 以上の場合は，pdlogucat コマンドで確認します。

2. 1.で取得した名称のアンロードログファイルを別の媒体に移動します。

3. 別の媒体へのデータ移動が正常に完了したかどうか，次の 2 点を確認します。

   ・表「自動ログアンロード機能の使用時に障害が発生したときの対策方法」にあるメッセージが出力されていないこと

   ・pdlogucat コマンドを実行し，対象のアンロードログファイルのシステムログのアンロードの完了状態が「completed」であること

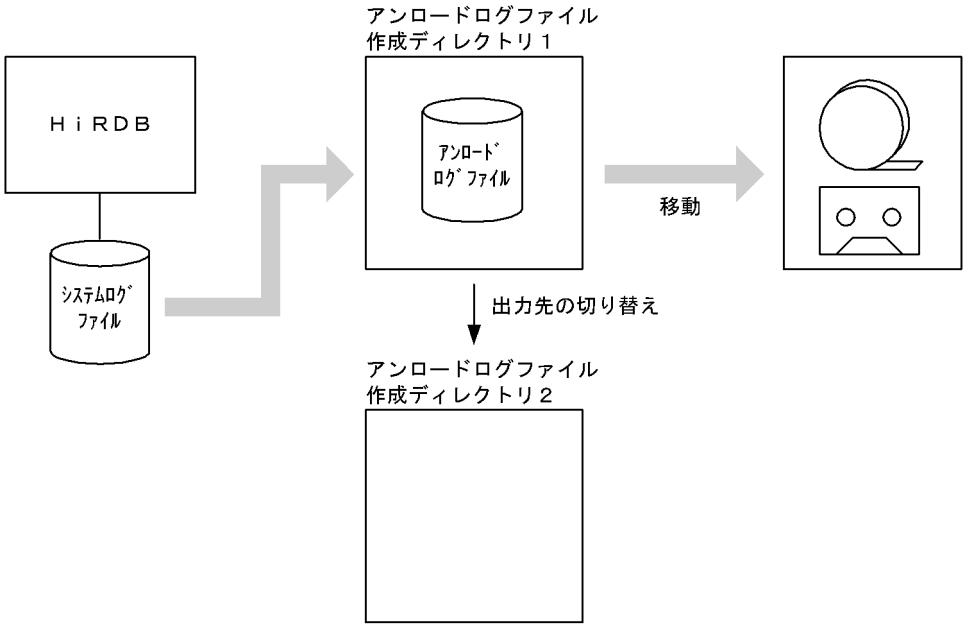
4. データ移動が正常に完了した後，アンロードログファイル作成先ディレクトリ内に 2.で移動したアンロードログファイルがある場合は削除します。

バックアップファイルとアンロードログファイルからデータベースを回復する場合，別の媒体に移動したアンロードログファイルをいったん磁気ディスク上に回復した後，アンロードログファイルの指定順を確認し，pdrstr コマンドを実行してください。

3.8.4　運用例 2（アンロードログファイル作成ディレクトリが二つの場合）

ここでは，次に示す条件下での自動ログアンロード機能の運用例を説明します。

- アンロードログファイル作成ディレクトリを二つ作成します。
- 片方のディレクトリの容量が満杯になってディレクトリが切り替わったら，そのディレクトリ内のアンロードログファイルをほかの媒体に移動します。



(1)　ディスクが満杯になるとアンロードログファイル作成ディレクトリが切り替わります

ディスクが満杯になるとアンロードログファイル作成ディレクトリ 2 に切り替わります。このとき，KFPS01151-I メッセージが出力されます。

```
KFPS01151-I bes1 changed auto log unload directory from /unlddir1/bes1
to /unlddir2/bes1.reason=1665
```

〔説明〕

アンロードログファイル作成ディレクトリが，C:¥unlddir1¥bes1 から C:¥unlddir2¥bes1 に切り替わりました。

(2)　ディレクトリ内の全アンロードログファイルをほかの媒体に移動します

C:¥unlddir1¥bes1 内の全アンロードログファイルをほかの媒体に移動してください。

(1)と(2)の作業を繰り返して，アンロードログファイルを別の媒体に格納していきます。

3.8.5　障害発生時の対策方法

自動ログアンロード機能の使用時に障害が発生したときの対策方法を次の表に示します。

表 3–7　自動ログアンロード機能の使用時に障害が発生したときの対策方法

障害内容	対策方法
KFPS01157-W (reason=1662)メッセージが出力された場合	pd_log_auto_unload_path オペランドに指定したディレクトリパス長，又はディレクトリの指定数に誤りがあります。 pd_log_auto_unload_path オペランドの指定値を見直し，HiRDB を再開始してください。
KFPS01155-W (reason=1667)メッセージが出力された場合	アンロードログファイル作成ディレクトリがあるディスクに障害が発生しました。 アンロードログファイルがなくなる可能性があるため，いったん運用を停止してバックアップを取得してください。ディスク障害が発生したアンロードログファイル作成ディレクトリを使用している全サーバ下の RD エリアのバックアップを取得してください。 なお，障害が発生したディスクを回復すれば，自動ログアンロード機能を再実行できます。
KFPS01151-I (reason=1665)メッセージが出力された場合	アンロードログファイル作成ディレクトリがあるディスクの容量が満杯です。 ディレクトリが一つの場合は自動ログアンロード機能を停止します。ディレクトリ内のアンロードログファイルを別の媒体に移動するか，不要なファイルを削除してください。その後，pdlogatul コマンドで自動ログアンロード機能を開始してください。 ディレクトリが二つ以上の場合はディレクトリを切り替えます。全ディレクトリのディスク容量が満杯になると，自動ログアンロード機能を停止します。この場合，ディレクトリ内のアンロードログファイルを別の媒体に移動してください。その後，pdlogatul コマンドで自動ログアンロード機能を開始してください。
KFPS01150-E メッセージが出力された場合	自動ログアンロード機能が停止しました。 • **pd_log_auto_unload_restart オペランドに Y を指定しているか，指定を省略した場合** 障害原因を取り除いてください。システムログファイルのスワップを契機に HiRDB は自動的にアンロードをリトライします。 • **pd_log_auto_unload_restart オペランドに N を指定している場合** 障害原因を取り除いた後に pdlogatul -b コマンドで自動ログアンロード機能を再実行してください。 障害の原因としては，すべてのアンロードログファイル作成ディレクトリが次に示す状態になったため，アンロードログファイル作成ディレクトリが使用できなくなったことが考えられます。 • ディスク満杯※ • ディスク障害 • パーミッション不正 • ディレクトリが存在しない • 前回の HiRDB 稼働時に作成されたアンロードログファイルがある 上記の要因に当てはまらない場合は，次に示す資料を取得して保守員に連絡してください。 • イベントログ • %PDDIR%¥spool 下の全ファイル • pdlogls コマンドの実行結果

障害内容	対策方法
	• HiRDB システム定義ファイルのバックアップ
KFPS01280-E メッセージが出力された場合	自動ログアンロード処理に失敗しました。 KFPS01280-E メッセージの直後に KFPS01271-I メッセージが出力されているかを確認してください。 • **KFPS01271-I メッセージが出力されている場合** アンロードログファイルは正常に作成されているため，対策は必要ありません。 • **KFPS01271-I メッセージが出力されていない場合** 次の 2 点を確認してください。次のようになっていれば，対策する必要はありません。 ・アンロードログファイル作成ディレクトリに，該当する世代のアンロードログファイルがある ・pdlogucat コマンドの実行結果から，該当するアンロードログファイルのアンロードの完了状態が「completed」となっている 上記のようになっていない場合，一部のアンロードログファイルが消失しているため，アンロードログファイルを使用したデータベース回復を行うことができません。この場合，一度運用を停止して HiRDB のバックアップを取得してください。
KFPS01158-W メッセージが出力された場合	自動ログアンロードの再開始に失敗しました。 システムマネジャユニットのイベントログに出力された KFPS01150-E メッセージを確認し，メッセージの理由コードから原因を特定して対策してください。 また，自動ログアンロード機能が停止しているため，アンロード待ち状態のシステムログファイルグループが残っています。次のどちらかのコマンドを実行し，アンロード済み状態にしてください。 • pdlogunld コマンド • pdstart コマンド（次回の HiRDB 開始時に行われる自動ログアンロード機能によるアンロード）

注※

ディスク容量不足などでアンロード処理に失敗した場合，HiRDB は未完成のアンロードログファイルを削除します。アンロードログファイルの作成中にいったんディスク容量が満杯になった後にそのファイルを削除するため，ディスク容量に空きがあるように見えることがあります。

3.8.6 HiRDB 終了時の注意事項

アンロード処理中に pdstop コマンドを実行した場合，アンロード処理が終了した後に HiRDB の正常終了又は計画停止処理を行います。アンロード処理を待ち合わせないで早急に HiRDB を正常終了又は計画停止する場合は，pdlogatul -t コマンドで自動ログアンロード機能を停止し，その後 pdstop コマンドを実行してください。この場合，アンロード待ち状態のシステムログファイルは次回の HiRDB 開始時に自動的にアンロードされます。

なお，HiRDB を強制終了した場合は，アンロード処理中でもアンロード処理を中断します。

3.9　システムログファイルの空き容量監視機能の運用方法

実行者　HiRDB 管理者

ここでは，システムログファイルの空き容量監視機能の運用方法について説明します。説明する項目は次のとおりです。

- システムログファイルの空き容量監視機能とは
- 環境設定
- 空き率が警告値未満になったときの HiRDB の処理
- 空き率が警告値未満になったときの HiRDB 管理者の処置
- 注意事項
- システムログファイルの状態情報ファイルの出力（空き容量監視機能）

3.9.1　システムログファイルの空き容量監視機能とは

(1)　機能概要

HiRDB の運用を続けると，データベースの更新ログがシステムログファイルに蓄積されていきます。すべてのシステムログファイルが満杯になると，データベースの更新ログが出力できなくなり，HiRDB の運用が続行できなくなるため，HiRDB は異常終了します。この異常終了によるサービス停止を回避するために，HiRDB 管理者はシステムログファイルの使用状況を常に監視する必要があります。

HiRDB ではシステムログファイルの空き容量を監視する機能として，**システムログファイルの空き容量監視機能**を提供しています。この機能を使用すると，システムログファイルの空き率を HiRDB が監視し，HiRDB 管理者が指定したレベルに従い，状況に応じて警告メッセージを出力したり，又はデータベースの利用を制限したりします。システムログファイルの空き容量監視機能は次に示す二つのレベルのどちらかを選択できます。

レベル 1：

システムログファイルの空き率が警告値未満になった場合，警告メッセージ KFPS01162-W を出力します。

レベル 2：

システムログファイルの空き率が警告値未満になった場合，新規トランザクションのスケジューリングを抑止して，サーバ内の全トランザクションを強制終了します。このとき，KFPS01160-E メッセージを出力します。これによって，システムログの出力量を抑えます。

(2)　レベル 1 とレベル 2 の機能差

レベル 1 とレベル 2 の機能差を次の表に示します。

表 3–8　レベル 1 とレベル 2 の機能差

機能の項目	レベル 1	レベル 2
システムログファイルの空き率の監視	○	○
エラー又は警告メッセージの出力	○	○
新規トランザクションのスケジューリング抑止	×	○
サーバ内の全トランザクションの強制終了	×	○
シンクポイントの取得	×	○

〔凡例〕

○：処理を行います。

×：処理を行いません。

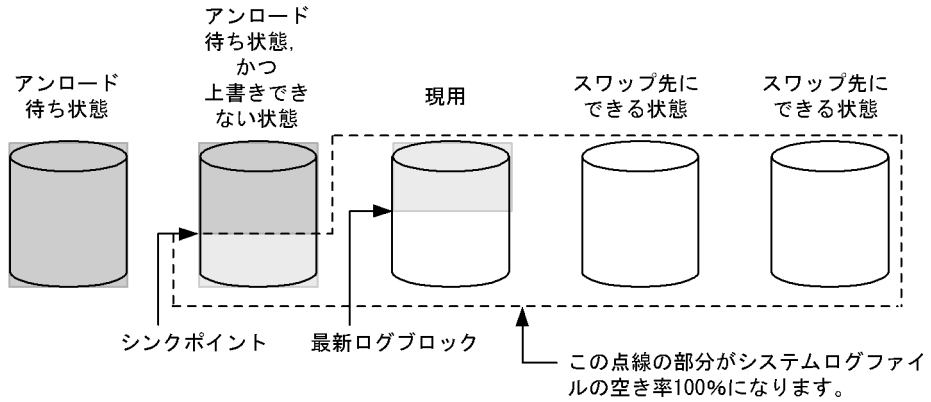
(3)　システムログファイルの空き率について

システムログファイルの空き率は，次に示す計算式を基にして HiRDB が算出します。

システムログファイルの空き率（単位：%）＝
**　システムログファイルの空き容量÷（システムログファイルの空き容量＋更新ログ量）×100%**

システムログファイルの空き容量とは，現用ファイルのシステムログ出力可能領域とスワップ先にできる状態のファイルの全領域を加算したものです。更新ログ量とは，シンクポイントから最新ログブロックまでのシステムログ量のことです。システムログファイルの空き率の概念を次の図に示します。

図 3–7　システムログファイルの空き率の概念



〔説明〕

- ファイルの薄い網掛け部分が更新ログ量となります。
- ファイルの白い部分がシステムログファイルの空き容量となります。
- ファイルの濃い網掛け部分は上書きできない状態のシステムログで，ロールバック処理の対象になりません。よって，更新ログ量，又はシステムログファイルの空き容量のどちらにも含まれません。

(4)　監視機能が警告するシステムログファイルの空き率（警告値）は

システムログファイルの空き容量監視監視機能が警告するシステムログファイルの空き率（警告値）は，システム障害が発生したときにデータベースの回復を安全に行える値です（データベースの回復時にシステムログファイルの容量不足が発生しない値です）。警告値はサーバの種類によって異なります。システムログファイルの空き容量監視機能が警告するシステムログファイルの空き率（警告値）を次の表に示します。

表 3–9　システムログファイルの空き容量監視機能が警告するシステムログファイルの空き率（警告値）

サーバの種類		システムログファイルの空き容量監視機能が警告するシステムログファイルの空き率（警告値）
HiRDB/シングルサーバの場合		67%
HiRDB/パラレルサーバの場合	フロントエンドサーバ	30%
	ディクショナリサーバ	67%
	バックエンドサーバ	

3.9.2　環境設定

(1)　どちらのレベルを選択するか

システムログファイルの空き容量不足による HiRDB の異常終了の可能性を低くするため，レベル 2 の設定を推奨します。ただし，レベル 2 の場合，システムログファイルの空き容量が不足したときに，サーバ内の全トランザクションが強制終了されます。このため，システムログファイルの設計をより正確に行う必要があります。システムログファイルの設計については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　レベルの設定

どちらのレベルを使用するかを **pd_log_remain_space_check** オペランドで指定します。このオペランドに warn（省略値）を指定するとレベル 1 が，safe を指定するとレベル 2 が設定されます。

3.9.3　空き率が警告値未満になったときの HiRDB の処理

(1)　レベル 1 の場合

システムログファイルの空き率が警告値未満になると，警告メッセージ KFPS01162-W を出力します。

(2)　レベル 2 の場合

(a)　新規トランザクションのスケジューリング抑止

KFPS01160-E メッセージを出力して，該当するサーバの新規トランザクションのスケジューリングを抑止します。運用コマンド及びユティリティの内部で発生するトランザクションも抑止対象になります。以後，このサーバでトランザクション処理を行う UAP には，KFPA19703-E（reason=TRNPAUSE）メッセージを出力してエラーリターンします。運用コマンド及びユティリティは異常終了します。このサーバ以外ではトランザクションを処理できますが，ほかのサーバからこのサーバにトランザクションがブランチした場合，UAP には KFPA19703-E（reason=TRNPAUSE）メッセージを出力してエラーリターンします。

(b)　サーバ内の全トランザクションの強制終了

該当するサーバで処理中のトランザクションを強制終了します。強制終了したトランザクションには KFPA11722-E（reason=SERVER PROCESS DOWN）メッセージを出力してエラーリターンします。なお，次に示すトランザクションは強制終了対象外になります。この場合，KFPS01163-W メッセージを出力してトランザクションの処理が終了するまで待ち合わせます。

- コミット又はロールバック中のトランザクション
- コミット 2 相目決着指示待ちのトランザクション
- ログレスモードのトランザクション

また，上記以外のトランザクション以外でも，システムログファイルに障害が発生している場合は，トランザクションを強制終了しないことがあります。

(c)　シンクポイントの取得

シンクポイントを「有効保証世代数＋ 1」回取得します。シンクポイントがすぐに有効化できない場合は，シンクポイントの取得をスキップしないでシンクポイントの有効化が完了するまで待ち合わせます。シンクポイントを取得するとシステムログファイルの空き率が増えます。空き率が警告値以上になると，新規トランザクションのスケジューリングを開始します。このとき，KFPS01161-I メッセージを出力します。

3.9.4　空き率が警告値未満になったときの HiRDB 管理者の処置

HiRDB 管理者は次に示すことをしてください。

(1)　空き率が不足したサーバを特定する

KFPS01160-E 又は KFPS01162-W メッセージを参照して空き率が不足したサーバを特定してください。pdlogcls コマンドのエラーの場合は，pdlogcls コマンドに指定したサーバが対象サーバになります。

レベル 2 限定の処置

このサーバは新規トランザクションのスケジューリング抑止状態になっています。コマンドで新規トランザクションのスケジューリング抑止状態のサーバを確認する場合は，pdls -d svr コマンドを実行してください。スケジューリング抑止状態のサーバは，コマンドの実行結果の STATUS に TRNPAUSE と表示されます。

(2) システムログファイルの状態を確認する

該当するサーバのシステムログファイルに対して pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

(3) システムログファイルをスワップ先にできる状態にする

スワップ先にできる状態のシステムログファイルを増やすために次に示す処置をしてください。

- 予約ファイルがある場合は，そのファイルをスワップ先にできる状態にしてください。
- アンロード待ち状態のファイルがある場合は，そのファイルをアンロード済み状態にしてください。
- 抽出未完了状態のファイルがある場合は，そのファイルを抽出完了状態にしてください。

(4) 強制終了されたトランザクションを確認する（レベル 2 限定）

強制終了されたトランザクションは KFPS00993-I メッセージで確認できます。このメッセージ中の REQUEST に log_remain_check と表示されたトランザクションが強制終了の対象になっています。KFPS01161-I メッセージが出力されると，新規トランザクションがスケジューリングされます。このメッセージが出力された後に，強制終了されたトランザクションを再実行してください。

(5) KFPS01163-W メッセージが出力された場合（レベル 2 限定）

強制終了対象外のトランザクションがある場合は，KFPS01163-W メッセージを出力してトランザクションの処理が終了するまで待ち合わせます。このトランザクションをコマンドなどで強制終了すると RD エリアが障害閉塞になる可能性があるため，pdcancel コマンドなどで強制終了しないでください。

なお，このトランザクションを発生させた UAP，運用コマンド，又はユティリティは次に示す手順で特定できます。

〈手順〉

1. KFPS01160-E 又は KFPS01163-W メッセージを参照して対象サーバ名を特定します。
2. **pdls -d trn -a -s サーバ名**コマンドでトランザクション情報を取得します。-s サーバ名には 1 で特定したサーバ名を指定します。
3. コマンドの実行結果中の PROGRAM 又は C-PID を参照して，このトランザクションを発生させた UAP，運用コマンド，又はユティリティを特定します。

(6)　システムログファイルの容量不足の原因を特定する

システムログファイルの容量不足の原因をシステムログファイルの状態情報ファイルを利用して特定してください。特定方法については，「システムログファイルの状態情報ファイルの出力（空き容量監視機能）」を参照してください。

(7)　システムログファイルの設計を見直す

システムログファイルの空き容量監視機能によって何度も警告が発生する場合，システムログファイルの設計（個数，ファイル容量など）に不備があると考えられます。したがって，システムログファイルの設計をし直してください。システムログファイルの設計については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(8)　pdlogswap コマンドを実行しないようにする

空き率が警告値未満になったときに pdlogswap コマンドでシステムログファイルをスワップすると，スワップ先にできる状態のファイルがないため，HiRDB が異常終了する可能性が高くなります。したがって，pdlogswap コマンドを実行しないようにしてください。

3.9.5　注意事項

(1)　レベル 1 及びレベル 2 共通の注意事項

- pdlogswap コマンドでシステムログファイルをスワップすると，システムログファイルの空き率が減ります。システムログファイルの個数が少ない場合は空き率が極端に減るため注意してください。
- pdlogswap コマンドを連続実行すると，システムログファイルの空き率が減るため注意してください。
- スワップ先にできる状態のファイルに障害が発生すると，システムログファイルの空き率が減ります。システムログファイルの個数が少ない場合は空き率が極端に減るため注意してください。
- HiRDB の再開始中に HiRDB が異常終了した場合，再び HiRDB の再開始処理を行います。このとき，スワップ先にできる状態のシステムログファイルを更に必要とするため，システムログファイルの容量不足によって HiRDB が異常終了することがあります。これについては，システムログファイルの空き容量監視機能を使用しても防げないため，このような異常事態も想定してシステムログファイルを設計してください。

(2)　レベル 2 限定の注意事項

pdlogcls コマンドでスワップ先にできる状態のファイルをクローズすると，システムログファイルの空き率が減ります。この結果，空き率が警告値未満となる場合はそのファイルをクローズしません。このとき，KFPS01280-E（理由コード 712）メッセージを出力してコマンドを異常終了します。

(3)　システムログファイルの自動拡張機能との関連

システムログファイルの空き容量監視機能とシステムログファイルの自動拡張機能を同時に使用している場合，スワップ先にできる状態のシステムログファイルがあれば，システムログファイルの自動拡張機能が適用されます。スワップ先にできる状態のシステムログファイルがないときは，システムログファイルの空き容量監視機能が適用されます。システムログファイルの自動拡張機能については，「システムログファイルの自動拡張機能の運用方法」を参照してください。

3.9.6　システムログファイルの状態情報ファイルの出力（空き容量監視機能）

ここでは，システムログファイルの容量不足の原因をシステムログファイルの状態情報ファイルを利用して特定する方法について説明します。

(1)　システムログファイルの状態情報ファイルの内容

システムログファイルの空き率が警告値未満になった場合，システムログファイルの状態情報ファイルが出力されます。このファイルにはシステムログファイル，シンクポイント，及びトランザクションの状態が取得されます。システムログファイルの状態情報ファイルは%PDDIR%¥spool¥pdjnlinf 下に作成されます。ファイル名称は次に示す規則に従って付けられます。

pdsnap．サーバ名．発生時刻

備考

- 日付及び時刻情報の取得に失敗した場合は，発生時刻が 999999999999 になります。
- 同じ名称のファイルがある場合は，そのファイルは上書きされます。
- ファイル名を変更している場合は，pdcspool コマンドでファイルを削除できないことがあります。

システムログファイルの状態情報ファイルの出力内容

システムログファイルの状態情報ファイルの出力内容を次に示します。バックエンドサーバ bes1 で空き率が警告値未満になった例です。

```
pdsnap.bes1.030415124350      ←ファイル名※1
Date：2003/04/15              ←発生日※1
Time：12:43:50                ←発生時刻※1
pdlogls -d sys -e -s bes1     ←HiRDBが内部発行したコマンド
  pdloglsコマンドの実行結果
            :

pdlogls -d spd -e -s bes1
  pdloglsコマンドの実行結果
            :

pdls -d trn -a -s bes1
  pdlsコマンドの実行結果
            :
```

```
pdls -d svr
  pdlsコマンドの実行結果※2
            :
```

注※1

日付及び時刻情報の取得に失敗した場合，99...99 が表示されます。必要に応じてファイル名を変更してください。

注※2

HiRDB/パラレルサーバの場合，該当するサーバのユニットについての情報だけを表示します。

(2) システムログファイルの容量不足の原因を特定する方法

システムログファイルの状態情報ファイルを利用して，システムログファイルの空き容量不足の原因を特定します。特定方法を次に説明します。

(a) システムログファイルの状態情報ファイルの特定

KFPS01160-E 又は KFPS01162-W メッセージからシステムログファイルの状態情報ファイルを特定します。

（例）KFPS01160-E が出力されていた場合

KFPS01160-E Insufficient system log space. Transaction service stopped. Transactions terminate by force, server = bes1, output file name = pdsnap.bes1.030401084500

この場合のシステムログファイルの状態情報ファイル名は次のとおりになります。

%PDDIR%¥spool¥pdjnlinf¥pdsnap.bes1.030401084500

(b) システムログファイルの状態の取得

システムログファイルの状態情報ファイルの内容を参照して，システムログファイルの状態を確認してください。システムログファイルの状態は pdlogls -d sys コマンドの出力結果の Status に表示されています。

(c) システムログファイルの空き容量不足の発生原因

システムログファイルの状態から発生原因を推測できます。システムログファイルの空き容量不足の発生原因を次の表に示します。

表 3–10 システムログファイルの空き容量不足の発生原因

システムログファイルの状態	システムログファイルの空き容量不足の発生原因
アンロード待ち状態のファイルが多い場合	システムログファイルをアンロードしていなかったと考えられます。定期的にシステムログファイルをアンロードしてください。
上書きできない状態のファイルが多い場合	次に示す原因が考えられます。 • 1 トランザクションで大量更新を行うと，更新ログ量が増えてシステムログファイルの空き率が減ります。このようなトランザクションは複数のトランザクションに分割してください。できない場合はシステムログファイルの容量を大きくしてください。

システムログファイルの状態	システムログファイルの空き容量不足の発生原因
	• 長時間決着しないトランザクションがあるとシンクポイントを取得できないため，その間の更新ログ量が増えてシステムログファイルの空き率が減ります。このようなトランザクションがある場合，サーバの最大待ち時間（クライアント環境定義の PDSWAITTIME オペランドの値又は pd_watch_pc_client_time オペランドの値）を見直すか，又はシンクポイントダンプ有効化のスキップ回数監視機能を使用してください。
抽出未完了状態のファイルが多い場合	システムログファイルの抽出処理が追い付かなかったと考えられます。システムログファイルの設計を見直すか，又は HiRDB Datareplicator を使用したデータ連動の運用方法を見直してください。

3.10　システムログファイルの残量警告機能の運用

実行者　HiRDB 管理者

ここでは，システムログファイルの残量警告機能の運用方法について説明します。説明する項目は次のとおりです。

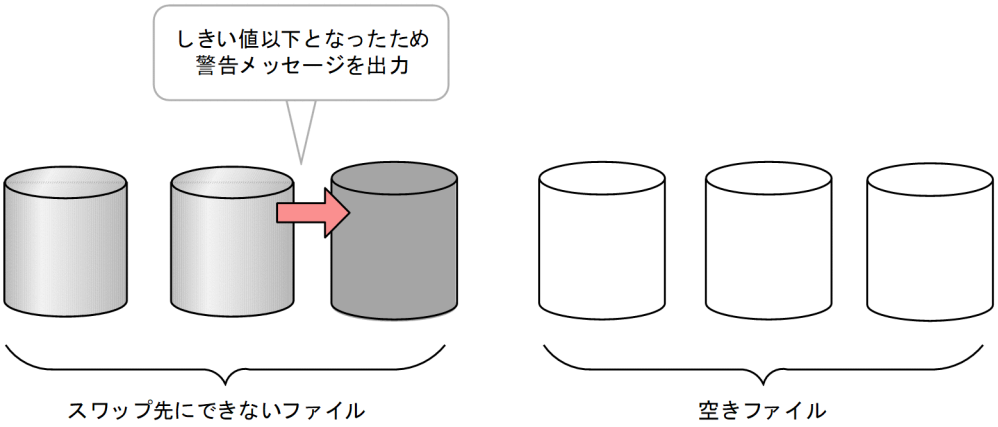
- システムログファイルの残量警告機能とは
- 環境設定
- 空きシステムログファイルの数がしきい値未満になったときの HiRDB の処理
- 空きシステムログファイルの数がしきい値未満になったときの HiRDB 管理者の処置

3.10.1　システムログファイルの残量警告機能とは

(1)　機能概要

HiRDB では空きシステムログファイルの数を監視する機能として，**システムログファイルの残量警告機能**を提供しています。この機能を使用すると，空きシステムログファイル数を監視し，ユーザが指定したしきい値以下になった場合に使用状況をユーザに通知することができます。

図 3–8　システムログファイルの残量警告機能の概要



(2)　空きシステムログファイルについて

スワップ先にできるシステムログファイルグループを空きシステムログファイルといいます。システムログファイルの状態が次のすべての条件を満たす場合，空きシステムログファイルとなります。

- オープン状態である。
- 現用のシステムログファイルではない。
- 上書き禁止状態ではない。

- pd_log_unload_check オペランドに'Y'を指定している場合で，かつ未アンロード状態ではない。
- HiRDB Datareplicator と連携している場合で，かつ抽出未完了状態ではない。

3.10.2　環境設定

pd_log_fg_warning_point オペランドに警告メッセージを出力したい空きシステムログファイルの数を指定してください。pd_log_fg_warning_point オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

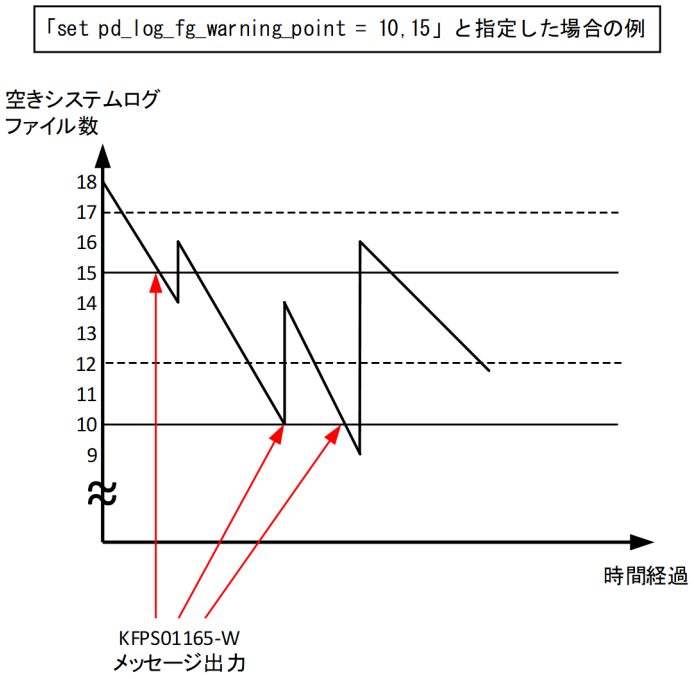
3.10.3　空きシステムログファイルの数がしきい値以下になった場合の HiRDB の処理

数秒間隔で空きシステムログファイル数の判定を行い，その数がしきい値以下になると，警告メッセージ KFPS01165-W を出力します。

警告メッセージは，空きシステムログファイル数がしきい値以下になったときに 1 回出力します。その後，空きシステムログファイル数が警告メッセージ出力のしきい値＋ 2 以上まで回復し，再び空きシステムログファイル数がしきい値以下になった場合は，再度警告メッセージを出力します。

再度警告メッセージ出力する場合の例を次の図に示します。

図 3–9　再度警告メッセージ出力する場合の例



また，運用しているシステムログファイルグループ数−2 以上の値を指定した場合，一度警告メッセージを出力すると，そのあとは警告メッセージを出力しなくなります。

なお，1 回の判定契機で複数のしきい値以下になった場合は，条件を満たした数分の警告メッセージを出力します。

3.10.4 空きシステムログファイルの数がしきい値以下になった場合の HiRDB 管理者の処置

KFPS01166-I メッセージの情報からスワップ先にできない要因を調査して，「システムログファイルの空き容量不足の発生原因」を参照して対策を検討してください。

3.11　システムログファイルの自動拡張機能の運用方法

実行者　HiRDB 管理者

ここでは，システムログファイルの自動拡張機能の運用方法について説明します。説明する項目は次のとおりです。

- システムログファイルの自動拡張機能とは
- 環境設定
- 自動拡張時の HiRDB の処理
- 自動拡張できなくなった場合の HiRDB 管理者の処置
- ほかの機能との関連
- 注意事項
- システムログファイルの状態情報ファイルの出力（自動拡張機能）

3.11.1　システムログファイルの自動拡張機能とは

システムログファイルの容量不足が発生すると，HiRDB システム（又はユニット）が異常終了します。これを回避するため，自動的にシステムログファイルの容量を拡張する機能（**システムログファイルの自動拡張機能**）を提供しています。この機能を適用することで，システムログファイルの容量不足による HiRDB システム（又はユニット）の異常終了の頻度を低減できます。

(1)　自動拡張する契機

システムログファイルを自動拡張する契機には，次の二つがあります。

契機 1：

システムログファイルの空き容量監視機能によって，システムログファイルの空き率が警告値未満になったことを HiRDB が検知したとき

契機 2：

現用のシステムログファイルが満杯になってスワップする時に，スワップ先にできる状態の待機ファイルがないとき

システムログファイルを自動拡張する契機になった場合，KFPS01391-I メッセージとシステムログファイルの状態情報ファイルが出力されます。システムログファイルの状態情報ファイルについては，「システムログファイルの状態情報ファイルの出力（自動拡張機能）」を参照してください。

なお，契機 1 でシステムログファイルを自動拡張する場合，警告メッセージの出力や，新規トランザクションのスケジューリング抑止などは行いません。

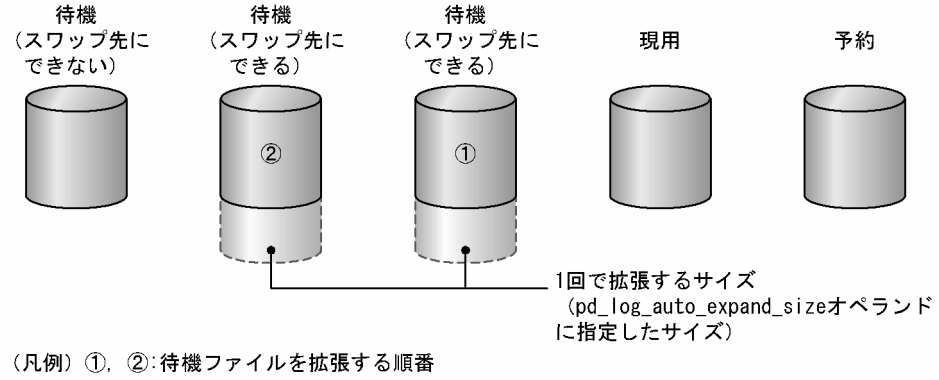
(2)　自動拡張する方法

システムログファイルを拡張する契機ごとに自動拡張する方法を説明します。

契機 1 の場合：

HiRDB はスワップ先にできる状態の待機ファイルを一つずつ順番に拡張します。拡張契機になってから，拡張対象のシステムログファイルを 1 回ずつ拡張したら，1 回目の自動拡張を終了します。契機 1 の場合の自動拡張の概要を次の図に示します。

図 3–10　システムログファイルの自動拡張の概要（契機 1 の場合）



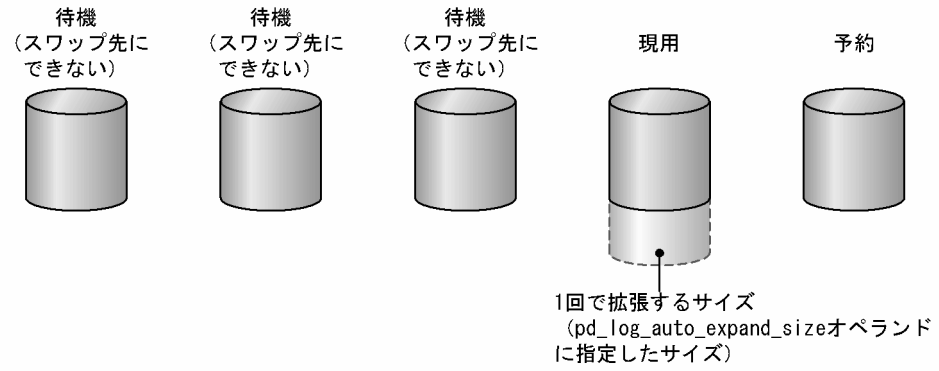
〔説明〕

契機 1 の場合，HiRDB は待機ファイルを順番に拡張します。1 回で拡張するサイズは，pd_log_auto_expand_size オペランドで指定します。現用ファイル，スワップ先にできない待機ファイル，及び予約ファイル（閉塞も含む）は拡張対象外となります。

契機 2 の場合：

HiRDB は現用ファイルを拡張します。契機 2 の場合の自動拡張の概要を次の図に示します。

図 3–11　システムログファイルの自動拡張の概要（契機 2 の場合）



〔説明〕

契機 2 の場合，HiRDB は現用ファイルを拡張します。1 回で拡張するサイズは，pd_log_auto_expand_size オペランドで指定します。待機ファイル，及び予約ファイル（閉塞も含む）は拡張対象外となります。

なお，システムログファイルを二重化している場合，ファイルグループごとに両系のシステムログファイルを拡張します。

自動拡張時の HiRDB の処理の詳細については，「自動拡張時の HiRDB の処理」を参照してください。

(3)　自動拡張できなくなる契機

システムログファイルの自動拡張機能を適用している場合でも，次に示す契機でシステムログファイルは自動拡張できなくなります。

1. システムログファイルを作成している HiRDB ファイルシステム領域があるディスクが満杯になった
2. システムログファイルの容量が，システムログファイルの上限（100 ギガバイト）に達した
3. システムログファイルの容量が，pd_log_auto_expand_size オペランドで指定した拡張上限に達した
4. 入出力エラーが発生し，システムログファイルの拡張に失敗した
5. メモリ不足でシステムログファイルの拡張に失敗した

1.以外の場合，システムログファイルは拡張対象外となります。自動拡張できなくなった場合の HiRDB 管理者の処置については，「自動拡張できなくなった場合の HiRDB 管理者の処置」を参照してください。

3.11.2　環境設定

システムログファイルの自動拡張機能を適用するための環境設定について説明します。

(1)　HiRDB システム定義

HiRDB システム定義で次のように指定してください。

- **pd_log_auto_expand_size** オペランドで，1 回当たりに拡張するサイズに 1 以上の値を指定
  0 を指定すると，システムログファイルの自動拡張機能は適用されません。

pd_log_auto_expand_size オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

(2)　HiRDB ファイルシステム領域

自動拡張機能を適用するシステムログファイルを作成する HiRDB ファイルシステム領域を作成する場合，**-a** オプション指定で **pdfmkfs** コマンドを実行してください。pdfmkfs コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(3)　システムログファイルの配置

pdfmkfs -a コマンドで作成した一つの HiRDB ファイルシステム領域に，システムログファイルを一つだけ配置してください。ファイルを複数作成しようとすると，エラーになります。

3.11.3　自動拡張時の HiRDB の処理

(1)　契機 1 で自動拡張する場合

システムログファイルの空き容量監視機能で，システムログファイルの空き率が警告値未満になったことを HiRDB が検知した場合に自動拡張するときの HiRDB の処理の詳細を次に示します。

- スワップ先にできる状態の待機ファイルが複数ある場合，現用ファイルに割り当てる順番が最後のシステムログファイルから順に拡張します。
- システムログファイルの拡張中に，そのファイルにスワップした場合，拡張をやめて現用ファイルになるため，拡張対象外となります。このとき，スワップに時間が掛かることがあります。
- システムログファイルの拡張中に，待機ファイルがスワップ先にできる状態になった場合，そのファイルも拡張対象となります。
- システムログファイルの拡張中に，現用状態，スワップ先にできない状態，又は予約状態になったシステムログファイルは拡張対象外となります。
- すべてのスワップ先にできる状態の待機ファイルを 1 回ずつ拡張したら，拡張を終了します。

(2)　契機 2 で自動拡張する場合

現用のシステムログファイル満杯によるスワップ時に，スワップ先にできる状態の待機ファイルがない場合に自動拡張するときの HiRDB の処理の詳細を次に示します。

- 現用ファイルを拡張します。このとき，システムログの入出力に時間が掛かることがあります。
- 現用ファイルの拡張中に，拡張対象のファイルが増えた場合（待機ファイルがスワップ先にできる状態になったなど），現用ファイルの拡張をやめ，拡張対象となったシステムログファイルを拡張します。なお，現用ファイルの拡張をやめるまでに時間が掛かることがあります。
- 現用ファイルの拡張中に，現用状態，スワップ先にできない状態，又は予約状態になったシステムログファイルは拡張対象外となります。
- すべてのスワップ先にできる状態の待機ファイルを 1 回ずつ拡張したら，拡張を終了します。

(3)　拡張中にユニットが異常終了した場合

システムログファイルの自動拡張中にユニットが異常終了した場合，ユニット再開始後に自動拡張可否の状態を引き継ぎます。

なお，ユニットが異常終了したタイミングによっては，拡張中だったシステムログファイルがある HiRDB ファイルシステム領域に不整合が発生するおそれがあります。その場合，pdffsck コマンドで HiRDB ファイルシステム領域の整合性の検証及び修復を行う必要があります。

3.11.4　自動拡張できなくなった場合の HiRDB 管理者の処置

HiRDB 管理者は，システムログファイルの状態を確認し，原因に応じて対処します。

(1)　自動拡張できなくなったシステムログファイルを特定する

KFPS01390-I メッセージを参照し，自動拡張できなくなったシステムログファイルを特定してください。KFPS01390-I メッセージには，自動拡張できなくなったシステムログファイルがあるサーバ名，ファイルグループ名，二重化している場合は系種別が表示されます。

(2)　自動拡張できなくなった原因を特定する

KFPS01390-I メッセージに表示される理由コードを参照し，自動拡張できなくなった原因を特定してください。また，該当するサーバのシステムログファイルに対して pdlogls -d sys -e -E コマンドを実行して，システムログファイルの状態を確認してください。pdlogls -d sys -e -E コマンドを実行すると，次の点を確認できます。

- システムログファイルの自動拡張機能を適用しているかどうか
- 該当するシステムログファイルグループが拡張対象かどうか
- 該当するシステムログファイルグループが拡張中かどうか
- 該当するシステムログファイルグループ，又は要素ファイルが拡張できるかどうか
- 該当する要素ファイルの使用できない領域の容量

(3)　特定した原因に応じてシステムログファイルを見直す

自動拡張できなくなった原因に応じて，対処してください。

システムログファイルを作成している HiRDB ファイルシステム領域があるディスクが満杯になった場合

自動拡張できなくなった原因が，システムログファイルを作成している HiRDB ファイルシステム領域があるディスクが満杯になったという場合（KFPS01390-I メッセージの理由コードが 205）の対処を次に示します。

1. OS のコマンドで，該当するシステムログファイルを作成しているディスクが満杯になったことを確認してください。
2. 不要なファイルを削除するなどして，ディスクの空き容量を増やしてください。

システムログファイルの容量が，システムログファイルの上限に達した場合

自動拡張できなくなった原因が，システムログファイルの上限に達したという場合（KFPS01390-I メッセージの理由コードが 1692）の対処を次に示します。

1. pdfls コマンドで，該当するシステムログファイルがシステムログファイルの上限（100 ギガバイト）に達したことを確認してください。

2. システムログファイルの総容量及び世代数を見直してください。システムログファイルの総容量及び世代数については，マニュアル「HiRDB システム導入・設計ガイド」の「システムログファイルの設計」を参照してください。
3. システムログファイルグループを追加してください。

システムログファイルの容量が，pd_log_auto_expand_size オペランドで指定した拡張上限に達した場合

自動拡張できなくなった原因が，pd_log_auto_expand_size オペランドで指定した拡張上限に達したという場合（KFPS01390-I メッセージの理由コードが 1694）の対処を次に示します。

1. pdfls コマンドで，該当するシステムログファイルが pd_log_auto_expand_size オペランドで指定した拡張上限に達したことを確認してください。
2. pd_log_auto_expand_size オペランドに指定する拡張上限サイズを見直すか，指定を省略してください。

   HiRDB システム定義を変更する場合，次のどちらかの方法で変更してください。

   ・pdchgconf コマンドを使用して変更する

   ・HiRDB を正常終了し，HiRDB システム定義を変更してから HiRDB を再開始する

入出力エラーが発生し，システムログファイルの拡張に失敗した場合

自動拡張できなくなった原因が，入出力エラーが発生し，システムログファイルの拡張に失敗したという場合（KFPS01390-I メッセージの理由コードが 209）の対処を次に示します。

1. pdlogls -d sys コマンドで，該当するシステムログファイルが障害閉塞していることを確認してください。
2. KFPS01390-I メッセージの前後に出力されたメッセージを参照し，入出力エラーの発生原因を特定し，対策してください。
3. 再度拡張できる状態にするには，該当するシステムログファイルを再作成してください。

> **参考**
>
> システムログファイルは，次の手順で再作成します。
>
> 1. pdlogcls コマンドで該当するシステムログファイルをクローズします。
> 2. pdlogrm コマンドで該当するシステムログファイルを削除します。
> 3. pdloginit コマンドでシステムログファイルを再作成します。
> 4. pdlogopen コマンドでシステムログファイルをオープンします。

メモリ不足でシステムログファイルの拡張に失敗した場合

自動拡張できなくなった原因が，メモリ不足が発生し，システムログファイルの拡張に失敗したという場合（KFPS01390-I メッセージの理由コードが 101）の対処を次に示します。

1. OS のコマンドで，メモリ不足が発生したことを確認してください。
2. メモリ所要量を見直してください。不要なプログラムを起動している場合は停止し，メモリ消費量を減らしてください。

3. 再度拡張できる状態にするには，該当するシステムログファイルをいったんクローズして，再度オープンしてください。

3.11.5　ほかの機能との関連

(1)　システムログファイルを二重化している場合

システムログファイルを二重化している場合，どちらかの系のシステムログファイルが自動拡張できなくなったとき，該当するファイルグループを拡張できない状態にして，拡張対象外とします。このとき，該当するファイルグループが使用できるシステムログファイルのサイズは，サイズが小さい方の系のサイズとし，差異となる領域はシステムログファイルが使用できない領域とします。

なお，システムログファイルの片系運転を適用している場合，システムログファイルに障害が発生しても，HiRDB は正常な系の自動拡張を続行します。ただし，自動拡張で拡張できない状態になった場合は，HiRDB はシステムログファイルに障害が発生したと判断しません。そのため，この場合は片系運転は行いません。自動拡張できなくなったファイルグループを拡張できない状態にした後で，さらにシステムログファイルに障害が発生した場合は片系運転を行いますが，障害が発生したのが拡張できない状態になった系のときは，ファイルグループを拡張できる状態に戻し，障害が発生しなかった正常な系を自動拡張します。このように，システムログファイルの片系運転を続けると，障害が発生した系と正常な系とでファイルサイズが異なっていきます。そのため，障害の要因を取り除いた後で，障害が発生した系を回復するには，正常な系のファイルサイズ以上の大きさで，障害が発生した系のファイルを再作成することをお勧めします。

(2)　システムログファイルの自動オープンをしている場合

システムログファイルの自動オープンをしていて，HiRDB システム（又はユニット）の再開始時にスワップ先にできる状態の待機ファイルがない場合，予約ファイルがあれば，HiRDB が予約ファイルをオープンしてスワップ先にできる状態にして，処理を続行します。予約ファイルがない場合，又はオープンしたファイルもすべて満杯になった場合は，現用ファイルを拡張します。

(3)　自動ログアンロード機能を使用している場合

システムログファイルの自動ログアンロード機能を使用していて，HiRDB システム（又はユニット）の再開始時にシステムログファイルが次の状態の場合，HiRDB は 1 世代のログアンロードが完了するまで待ちます。

- スワップ先にできる状態の待機ファイル及び予約ファイルがない
- アンロード待ち状態のシステムログファイルがある

アンロードが完了したら，そのシステムログファイルが現用状態になります。そのファイルが満杯になり，スワップ先にできる状態の待機ファイルがない場合は，現用ファイルを拡張します。

3.11.6　注意事項

- raw I/O 機能を適用した HiRDB ファイルシステム領域に自動拡張するシステムログファイルを配置する場合，その HiRDB ファイルシステム領域を作成するディスクには，十分な空き容量を確保してください。
- システムログファイルの自動拡張に伴って，HiRDB ファイルシステム領域が拡張される場合，HiRDB は拡張した領域を初期化するため，ディスク入出力の負荷が上昇します。
- 拡張中のシステムログファイルにスワップする場合，スワップに時間が掛かることがあります。この場合，pd_log_swap_timeout オペランドの指定値，又は pdlogswap コマンドの-t オプションの指定値を見直してください。
- 現用ファイルを拡張する場合，拡張中は HiRDB ファイルシステム領域に排他が掛かるため，システムログの入出力に時間が掛かることがあります。

3.11.7　システムログファイルの状態情報ファイルの出力（自動拡張機能）

システムログファイルを自動拡張する契機になった場合，システムログファイルの状態情報ファイルが出力されます。このファイルにはシステムログファイル，シンクポイント，及びトランザクションの状態が取得されます。システムログファイルの状態情報ファイルは%PDDIR%¥spool¥pdjnlinf 下に作成されます。ファイル名称は次に示す規則に従って付けられます。

pdsnap．サーバ名．世代番号

備考

- 世代番号は，サーバ開始後の最初の自動拡張契機では，1 になります。それ以外では 2 になります。
- 同じ名称のファイルがある場合は，そのファイルは上書きされます。
- ファイル名を変更している場合は，pdcspool コマンドでファイルを削除できないことがあります。

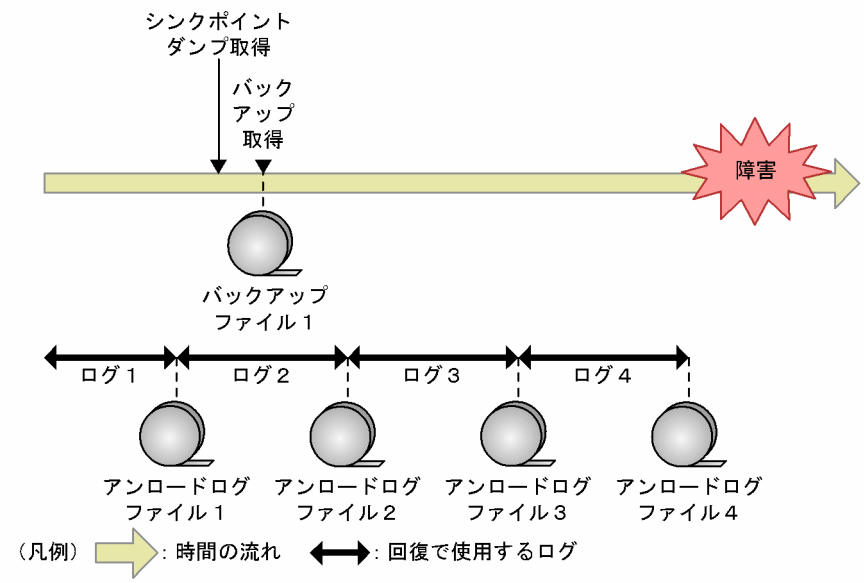
システムログファイルの状態情報ファイルの出力内容などについては，「システムログファイルの状態情報ファイルの出力（空き容量監視機能）」を参照してください。

3.12　データベース回復時に必要なアンロードログファイルの特定方法

データベース回復ユティリティ（pdrstr）でデータベースを最新の同期点に回復する場合，-d オプションでディレクトリを指定して pdrstr コマンドを実行するときは，アンロードログファイルの順序を意識する必要はありません。しかし，-l オプションでファイル名を指定して pdrstr コマンドを実行するときは，アンロードログファイルを時系列（世代順）に指定する必要があります。ここでは，データベース回復時の入力情報となるアンロードログファイルの特定方法について説明します。

次の図に示す例の場合，データベースを最新の同期点まで回復するには，バックアップファイル 1 とアンロードログファイル 2～4 を使用します。このとき，pdrstr コマンドの-l オプションに，アンロードログファイル 2，3，4 の順序で指定する必要があります。

図 3–12　障害が発生した場合の例



pdrstr コマンドの-l オプションに最初に指定するアンロードログファイル（例ではアンロードログファイル 2）を特定するには，バックアップ取得時に情報を取得します。また，-l オプションの最後に指定するアンロードログファイル（例ではアンロードログファイル 4）は，データベース回復時に作成したアンロードログファイルになります。最初と最後のファイルが特定できたら，時系列になるよう並べ替え，pdrstr コマンドの-l オプションに指定します。データベース回復時に使用するアンロードログファイルを特定して，データベースを回復する手順を次の図に示します。

図 3–13　データベース回復時に使用するアンロードログファイルの特定手順



注※1

手順は，「最初に指定するアンロードログファイルの情報取得（バックアップ取得時）」を参照してください。

注※2

手順は，「最後に指定するアンロードログファイルの情報取得（データベース回復時）」を参照してください。

注※3

手順は，「アンロードログファイルの並び替え（時系列リストの作成）」を参照してください。

3.12.1　最初に指定するアンロードログファイルの情報取得（バックアップ取得時）

pdrstr コマンドの-l オプションに最初に指定するアンロードログファイルを特定するために，バックアップ取得時に情報を取得します。

(1)　HiRDB の稼働中にバックアップを取得する場合

HiRDB の稼働中にバックアップを取得する場合に，最初に指定するアンロードログファイルの情報を取得する手順を次に示します。

〈手順〉

1. pdlogswap -d sys -w コマンドを実行し，システムログファイルをスワップし，シンクポイントダンプを取得します。HiRDB/パラレルサーバの場合は，バックアップ取得対象の全サーバのシステムログファイルをスワップしてください。
2. pdlogls -d sys，及び pdlogatul -d sys -i コマンドを実行し，実行結果から，現用のシステムログファイルの次の情報を記録します。
   ・サーバ名

・ファイルグループ名

・サーバラン ID

・世代番号

ここで確認したシステムログファイルが，最初に指定するアンロードログファイルになります。

3. データベース複写ユティリティ（pdcopy），又はほかの製品のバックアップ機能でバックアップを取得します。

参考

手順 3.で pdcopy コマンドを実行する時に-z オプション及び-p オプションを指定すると，pdcopy コマンドの処理結果リストにデータベース回復時に必要なシステムログファイルの情報を出力します。pdcopy コマンド実行時に取得する情報は，それ以前に pdlogls -d sys 及び pdlogatul -d sys -i コマンドから得られる情報より正確なため，指定できる場合は pdcopy コマンド実行時に-z オプション及び-p オプションを指定することをお勧めします。

(2) HiRDB の停止中，又は pdstart -r コマンドで HiRDB を開始している場合にバックアップを取得するとき

HiRDB の停止中，又は pdstart -r コマンドで HiRDB を開始している場合にバックアップを取得するときに，最初に指定するアンロードログファイルの情報を取得する手順を次に示します。

〈手順〉

1. データベース複写ユティリティ（pdcopy），又はほかの製品のバックアップ機能でバックアップを取得します。
2. HiRDB を正常開始します（pdstart -r コマンドで開始している場合はいったん正常終了してから，正常開始します）。
3. HiRDB の正常開始直後に pdlogls -d sys，及び pdlogatul -d sys -i コマンドを実行し，実行結果から，現用のシステムログファイルの次の情報を記録します。

   ・サーバ名

   ・ファイルグループ名

   ・サーバラン ID

   ・世代番号

   ここで確認したシステムログファイルが，最初に指定するアンロードログファイルになります。

(3) 運用例 1

自動ログアンロード機能を使用している場合，HiRDB の稼働中にバックエンドサーバ（bes1）のバックアップを取得するときの例を示します。

1. pdlogswap -d sys -w コマンドを実行し，システムログファイルをスワップし，シンクポイントダンプを取得します。

```
pdlogswap -d sys -s bes1 -w
```

pdlogswap コマンドの戻り値が 4 の場合，シンクポイントダンプが有効化されていないため，KFPS02183-I メッセージが出力され，シンクポイントダンプが有効化されるまで待ってください。

2. pdlogls -d sys，及び pdlogatul -d sys -i コマンドを実行します。

```
pdlogls -d sys -s bes1

HOSTNAME : dbhost1(153027)
Group    Type Server  Gen No.  Status  Run ID   Block No.       Ex-Status
logfg01  sys  bes1          13 os----u 3a765d6d     eff    fc8  --------
logfg02  sys  bes1          14 os----u 3a765d6d     fc9   1092  --------
logfg03  sys  bes1          15 os----u 3a765d6d    1093   115b  --------
logfg04  sys  bes1          16 osu---u 3a765d6d    115c   1226  --------
logfg05  sys  bes1          17 oc-d--u 3a765d6d    1227   1227  --------
logfg06  sys  bes1          12 os----u 3a765d6d     e35    efe  --------
```

```
pdlogatul -d sys -s bes1 -i

HOSTNAME : dbhost1(153153)
SERVER_NAME:bes1
AUTO_LOG_UNLOAD  NOW_UNLOAD_LOG_GROUP  CREATE_DIR
         ACTIVE                  ****  C:¥unload_dir
CURRENT LOG GENERATION INFO.
LOG_GROUP GEN_NO.  SERVER_RUN_ID RUN_ID   UNLOAD_FILE_NAME
logfg05          17 3a765d82      3a765d6d bes1_3a765d820017_logfg05
```

上記の結果から，現用のシステムログファイルの次の情報を記録します（下線部分）。

- サーバ名：bes1
- ファイルグループ名：logfg05
- サーバラン ID：3a765d82
- 世代番号：17

なお，ここで確認するのはサーバラン ID（SERVER_RUN_ID）です。ラン ID（RUN_ID）と間違えないように注意してください。

3. データベース複写ユティリティ（pdcopy）でバックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast_rdarea¥mast_rdarea01 -M r -s bes1 -b C:¥pdcopy¥backup¥bes1bkup01

-z C:¥pdcopy¥logpoint¥bes1logp01 -p C:¥pdcopy¥result¥bes1result01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-s：バックエンドサーバ（bes1）の RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。

-p：pdcopy コマンドの処理結果リストのファイル名を指定します。

参考

pdcopy コマンドの処理結果リストから最初に指定するアンロードログファイルを特定するためには，《LOG FILE INFORMATION》の情報を記録してください。出力例を次に示します。記録する情報は下線部分です。

```
*** DB BACKUP INFORMATION LIST ***       BACKUPMODE : EXCLUSIVE
<<LOG FILE INFORMATION>>
  SYSTEM ID     : HRD1
  UNIT ID       : UNT1
  TYPE          : sys
  SERVER NAME   : bes1
  SERVER RUN ID : 3a765d82
  LOG SERVER RUN ID : 3a765d6d
  FILE NAME     : logfg05
  GENERATION NO : 17
  BLOCK NO      : 1227
  HEADER WRITE COUNTER : 3
  HEADER WRITE TIME : 3a765d7a
```

- サーバ名：bes1
- ファイルグループ名：logfg05
- サーバラン ID：3a765d82
- 世代番号：17

(4)　運用例 2

自動ログアンロード機能を使用している場合，pdstart -r コマンドで HiRDB を開始しているときにディクショナリサーバ（dic）のバックアップを取得するときの例を示します。

1. データベース複写ユティリティ（pdcopy）でバックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast_rdarea¥mast_rdarea01 -M x -s dic -b C:¥pdcopy¥backup¥dicbkup01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照・更新不可能モードを指定します。

-s：ディクショナリサーバ（dic）の RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

2. HiRDB をいったん正常終了してから，正常開始します。

```
pdstop

pdstart
```

3. HiRDB の正常開始直後に pdlogls -d sys，及び pdlogatul -d sys -i コマンドを実行します。

```
pdlogls -d sys -s dic

HOSTNAME : dbhost1(152953)
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
logfg01  sys  dic             1 oc-d--u 3a7a6090         1         1 --------
logfg02  sys  dic             5 os----u 3a765d6d        85        99 --------
logfg03  sys  dic             2 os----u 3a765d6d       100       109 --------
logfg04  sys  dic             3 os----u 3a765d6d       110       121 --------
logfg05  sys  dic             3 os----u 3a765d6d       122       130 --------
```

```
pdlogatul -d sys -s dic -i

HOSTNAME : dbhost1(152953)
SERVER_NAME:dic
AUTO_LOG_UNLOAD  NOW_UNLOAD_LOG_GROUP  CREATE_DIR
         ACTIVE                  ****  C:¥unload_dir
CURRENT LOG GENERATION INFO.
LOG_GROUP GEN_NO.  SERVER_RUN_ID RUN_ID   UNLOAD_FILE_NAME
logfg01           1 3a7a60ae      3a7a6090 dic_3a7a60ae0007_logfg01
```

上記の結果から，現用のシステムログファイルの次の情報を記録します（下線部分）。

- サーバ名：dic
- ファイルグループ名：logfg01
- サーバラン ID：3a7a60ae
- 世代番号：1

なお，ここで確認するのはサーバラン ID（SERVER_RUN_ID）です。ラン ID（RUN_ID）と間違えないように注意してください。

3.12.2 最後に指定するアンロードログファイルの情報取得（データベース回復時）

pdrstr コマンドの-l オプションに最後に指定するアンロードログファイルを特定するために，データベース回復時に行う手順を次に示します。

〈手順〉

1. pdlogswap -d sys コマンドを実行して，現用のシステムログファイルをスワップします。
2. 自動ログアンロード機能を使用している場合はシステムログのアンロード処理が終了したことを確認します。自動ログアンロード機能を使用していない場合や停止している場合は手動でアンロードします。

ここで作成するアンロードログファイルが，pdrstr に入力する最後のアンロードログファイルとなります。

3.12.3　アンロードログファイルの並び替え（時系列リストの作成）

時系列リストを作成し，「最初に指定するアンロードログファイルの情報取得（バックアップ取得時）」で記録したアンロードログファイルから，「最後に指定するアンロードログファイルの情報取得（データベース回復時）」で作成したアンロードログファイルまでを時系列に並び替えます。データベースを回復するときにこのリストを参照して，pdrstr コマンドの-l オプションにアンロードログファイルを時系列に指定してください。

(1)　自動ログアンロード機能を使用している場合

自動ログアンロード機能を使用している場合の並び替え手順を次に示します。

〈手順〉

1. pdlogatul -d sys -i コマンドを実行して，自動ログアンロード機能が動作していることを確認します。HiRDB/パラレルサーバの場合は，回復対象の全サーバについて確認してください。動作していない場合，pdlogatul -b コマンドで自動ログアンロード機能を再開始してください。
2. pdlogls -d sys コマンドでシステムログファイルの状態を確認します。アンロード待ち状態のシステムログファイルがあれば，アンロード待ち状態のシステムログファイルがなくなるまで待ってください。
3. 「最初に指定するアンロードログファイルの情報取得（バックアップ取得時）」で記録した「サーバ名」，「サーバラン ID」，「ログファイルグループ名」，及び「世代番号」を確認します。
4. 回復対象となる HiRDB サーバごとにすべてのアンロードログファイルの昇順の時系列リストを作成します。

   アンロードログファイル名は次に示す形式に従って決められているため，アンロードログファイル作成ディレクトリ内のアンロードログファイルは時系列に並んでいます。

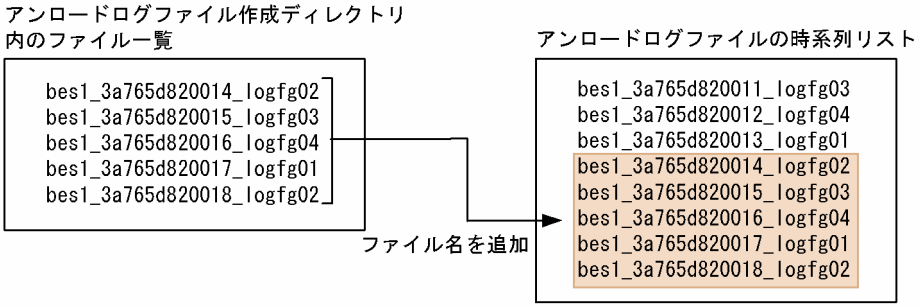
   サーバ名_サーバラン ID と世代番号※_ファイルグループ名

   注※

   世代番号は下 4 けただけであり，4 けたに満たない場合は，世代番号の値が 4 けたになるように左側に 0 が入った値になります。

   アンロードログファイルの時系列リストの作成方法を次の図に示します。

図 3–14　アンロードログファイルの時系列リストの作成方法



〔説明〕

アンロードログファイル作成ディレクトリ内のファイル一覧をそのままアンロードログファイルの時系列リストに追加してください。アンロードログファイル作成ディレクトリが二つ以上の場合，ファイル名を比べて時系列に並び替えてください。

注意事項

4.で作成した時系列リストで，世代番号が 0000 から ffff になると，アンロードログファイルが時系列に並びません。この場合，サーバラン ID が同じで世代番号が 0000 から通番になるファイルをサーバラン ID が同じで世代番号が ffff のファイルの下に移動してください。アンロードログファイルが時系列に並ばない例を次に示します。

●アンロードログファイル作成ディレクトリ内のファイル一覧

```
bes1_3a765d820000_logfg04    ← 0000がある
bes1_3a765d820001_logfg01
bes1_3a765d820002_logfg02
bes1_3a765d82fffe_logfg02
bes1_3a765d82ffff_logfg03
```

この場合，アンロードログファイルの作成順序は次のとおりになります。この順にアンロードログファイルの時系列リストを作成してください。

1. bes1_3a765d82fffe_logfg02
2. bes1_3a765d82ffff_logfg03
3. bes1_3a765d820000_logfg04
4. bes1_3a765d820001_logfg01
5. bes1_3a765d820002_logfg02

5. 昇順に並べた時系列リストから，3.で確認したアンロードログファイルを先頭とする昇順リストを抽出します。ここで抽出した昇順リストがデータベース回復時に指定するアンロードログファイルの指定順序となります。

(2)　自動ログアンロード機能を使用していない場合

自動ログアンロード機能を使用していない場合の並び替え手順を次に示します。

〈手順〉

1. アンロード待ち状態のシステムログファイルがないことを確認します。アンロード待ち状態のシステムログファイルがある場合，pdlogunld コマンドでアンロードします。
2. 「最初に指定するアンロードログファイルの情報取得（バックアップ取得時）」で記録した「サーバ名」，「サーバラン ID」，「ログファイルグループ名」，及び「世代番号」を確認します。
3. すべてのアンロードログファイルに対して pdlogucat コマンドを実行し，各アンロードログファイルの「サーバ名」，「サーバラン ID」，「ログファイルグループ名」，及び「世代番号」を確認します。
4. 3.で確認した「サーバ名」，「サーバラン ID」，「世代番号」，「ログファイルグループ名」をキーにして昇順に並び替え，時系列リストを作成します。
5. 4.で作成した昇順の時系列リストから，2.で確認したアンロードログファイルを先頭とする昇順リストを抽出します。ここで抽出した昇順リストがデータベース回復時に指定するアンロードログファイルの指定順序となります。

(3) 注意事項

アンロードログファイルの時系列リストでアンロードログファイルの抜けに注意してください。サーバラン ID が同じ場合は世代番号が 1 ずつ加算されるか，又は同一世代番号で異なるファイルグループ名称になっています。

これに該当しない場合はアンロードログファイルが抜けている可能性があります。該当するアンロードログファイルがない場合，抜けている分のシステムログファイルを再度アンロードする必要があります。システムログファイルが既に上書きされていてアンロードログファイルを作成できない場合は，最新の同期点にデータベースを回復できません。この場合，データベースをバックアップ取得時点にしか回復できません。

(4) 運用例 1

自動ログアンロード機能を使用している場合，HiRDB の稼働中にバックエンドサーバ（bes1）を回復するときの例を示します。なお，ここでは「最後に指定するアンロードログファイルの情報取得（データベース回復時）」の手順も合わせて説明します。

1. pdlogswap -d sys コマンドを実行して，現用のシステムログファイルをスワップします。

```
pdlogswap -d sys -s bes1 -w
```

2. pdlogatul -d sys -i コマンドを実行して，自動ログアンロード機能が動作していることを確認します。

```
pdlogatul -d sys -s bes1 -i

HOSTNAME : dbhost1(101653)
SERVER_NAME:bes1
AUTO_LOG_UNLOAD  NOW_UNLOAD_LOG_GROUP  CREATE_DIR
          ACTIVE               logfg05  C:¥unload_dir2
CURRENT LOG GENERATION INFO.
```

```
LOG_GROUP  GEN_NO.  USER_RUN_ID RUN_ID   UNLOAD_FILE_NAME
logfg01           7 3a7a60ae    3a7a6090 bes1_3a7a60ae0007_logfg01
```

AUTO_LOG_UNLOAD が ACTIVE と表示されていることから，自動ログアンロード機能が動作中であると確認できます。

3. pdlogls -d sys コマンドでシステムログファイルの状態を確認します。

```
pdlogls -d sys -s bes1

HOSTNAME : dbhost1(101724)
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
logfg01  sys  bes1            7 oc-d--u 3a7a0690      43f      43f  --------
logfg02  sys  bes1            2 os----u 3a7a0690       cb      194  --------
logfg03  sys  bes1            3 os----u 3a7a0690      195      25e  --------
logfg04  sys  bes1            4 os----u 3a7a0690      25f      328  --------
logfg05  sys  bes1            5 osu-b-u 3a7a0690      329      3f2  --------
logfg06  sys  bes1            6 osu---u 3a7a0690      3f3      43e  --------
```

上記結果から，logfg05 及び logfg06 がアンロード待ち状態で，logfg05 のアンロード中であることが確認できます。一定時間待ち，再度 pdlogls -d sys コマンドを実行します。

```
pdlogls -d sys -s bes1

HOSTNAME : dbhost1(103519)
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
logfg01  sys  bes1            7 oc-d--u 3a7a0690      43f      43f  --------
logfg02  sys  bes1            2 os----u 3a7a0690       cb      194  --------
logfg03  sys  bes1            3 os----u 3a7a0690      195      25e  --------
logfg04  sys  bes1            4 os----u 3a7a0690      25f      328  --------
logfg05  sys  bes1            5 os----u 3a7a0690      329      3f2  --------
logfg06  sys  bes1            6 os----u 3a7a0690      3f3      43e  --------
```

上記結果から，アンロードログファイルの作成が完了したことが確認できます。

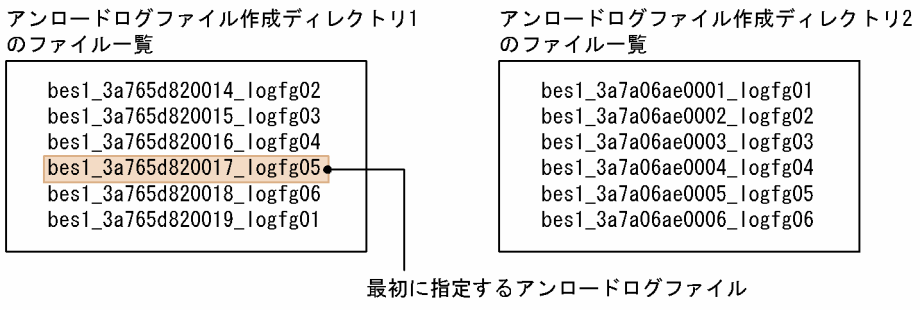
4. バックアップ時に取得した最初に指定するアンロードログファイルの情報から，「サーバ名」，「サーバラン ID」，「ログファイルグループ名」，及び「世代番号」を確認します。この例での情報を次に示します。

   - サーバ名：bes1
   - ファイルグループ名：logfg05
   - サーバラン ID：3a765d82
   - 世代番号：17

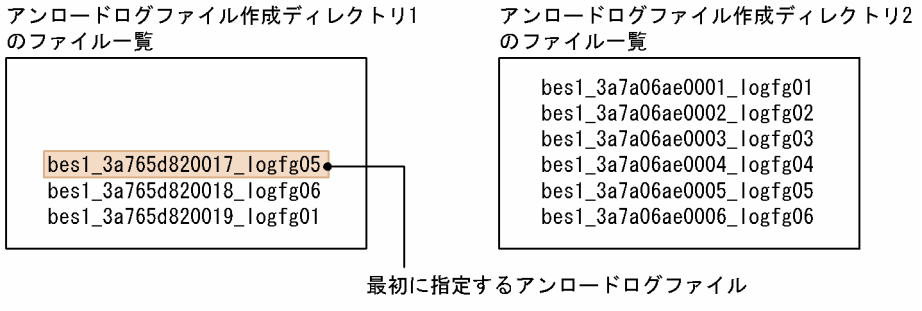
   このことから，最初に指定するアンロードログファイルは bes1_3a765d820017_logfg05 です。

5. 昇順の時系列リストを作成します。

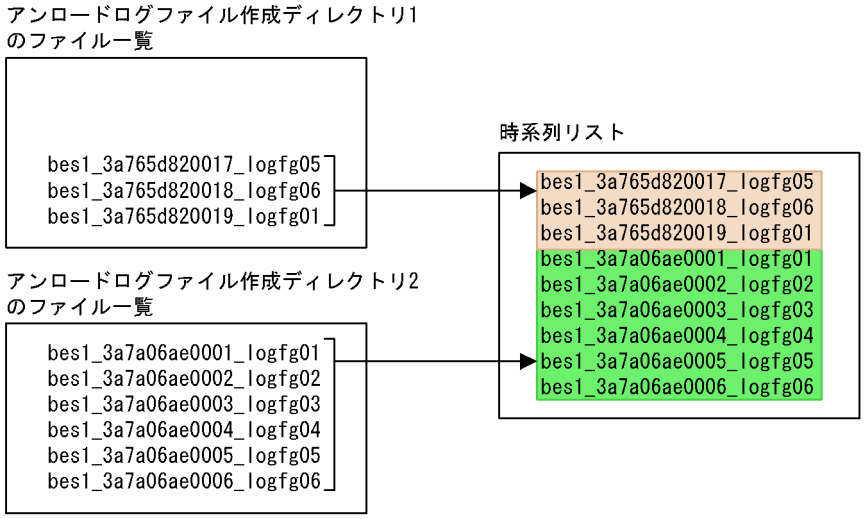
   まず，アンロードログファイル作成ディレクトリ内のファイル一覧を作成します。ここでは，アンロードログファイル作成ディレクトリが二つあるとします。



最初に指定するアンロードログファイルが bes1_3a765d820017_logfg05 なので，bes1_3a765d820017_logfg05 があるアンロードログファイル作成ディレクトリ 1 のファイル一覧から bes1_3a765d820017_logfg05 より前の世代番号のファイル名を削除します。



二つのアンロードログファイル作成ディレクトリのファイル一覧を昇順に並び替えてから，一つの時系列リストを作成します。



6. 5.で作成した時系列リストの順でアンロードログファイルを指定して，pdrstr コマンドを実行します。

```
pdrstr -m C:¥rdarea¥mast_rdarea¥mast_rdarea01 -s bes1 -b C:¥pdcopy¥backup¥bes1bkup01
-l C:¥unload_dir1¥bes1_3a765d820017_logfg05,C:¥unload_dir1¥bes1_3a765d820018_logfg06,
C:¥unload_dir1¥bes1_3a765d820019_logfg01,C:¥unload_dir2¥bes1_3a7a06ae0001_logfg01,
C:¥unload_dir2¥bes1_3a7a06ae0002_logfg02,C:¥unload_dir2¥bes1_3a7a06ae0003_logfg03,
C:¥unload_dir2¥bes1_3a7a06ae0004_logfg04,C:¥unload_dir2¥bes1_3a7a06ae0005_logfg05,
C:¥unload_dir2¥bes1_3a7a06ae0006_logfg06
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-s：回復対象となるサーバ（bes1）を指定します。

-b：回復に使用するバックアップファイルを指定します。

-l：回復に使用するアンロードログファイルを時系列に指定します。

(5) 運用例 2

自動ログアンロード機能を使用している場合，マスタ用 RD エリアに障害が発生し，pdstart -r コマンドで HiRDB を開始してディクショナリサーバ（dic）を回復するときの例を示します。

1. HiRDB を pdstart -r コマンドで開始する前に，pdlogls -d sys コマンドを実行して，現用のシステムログファイルを確認します。

```
pdlogls -d sys -s dic

HOSTNAME : dbhost1(103519)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
logfg01  sys  dic             1 cn----u 3a7a6090         1         b --------
logfg02  sys  dic             2 cn----u 3a7a6090         c        18 --------
logfg03  sys  dic             3 cn----u 3a7a6090        19        24 --------
logfg04  sys  dic             4 cn---cu 3a7a6090        25        2c --------
logfg05  sys  dic            11 cn----u 3a765d82       122       130 --------
```

   上記結果から，logfg04 以外はアンロードが完了していることが確認できます。

2. HiRDB が再開始していない場合，自動ログアンロード機能でアンロードできないため，-f オプション指定で pdlogunld -d sys コマンドを実行して，手動で logfg04 をアンロードします。

```
pdlogunld -d sys -s dic -g logfg04 -o C:¥unload_dir2¥dic_unload_logfg04 -f
```

3. pdstart -r コマンドで HiRDB を開始します。

```
pdstart -r
```

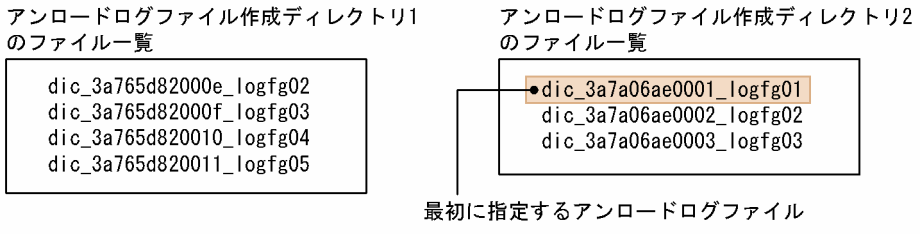
4. バックアップ時に取得した最初に指定するアンロードログファイルの情報から，「サーバ名」，「サーバラン ID」，「ログファイルグループ名」，及び「世代番号」を確認します。この例での情報を次に示します。

   - サーバ名：dic
   - ファイルグループ名：logfg01
   - サーバラン ID：3a7a60ae
   - 世代番号：1

   このことから，最初に指定するアンロードログファイルは dic_3a7a60ae0001_logfg01 です。

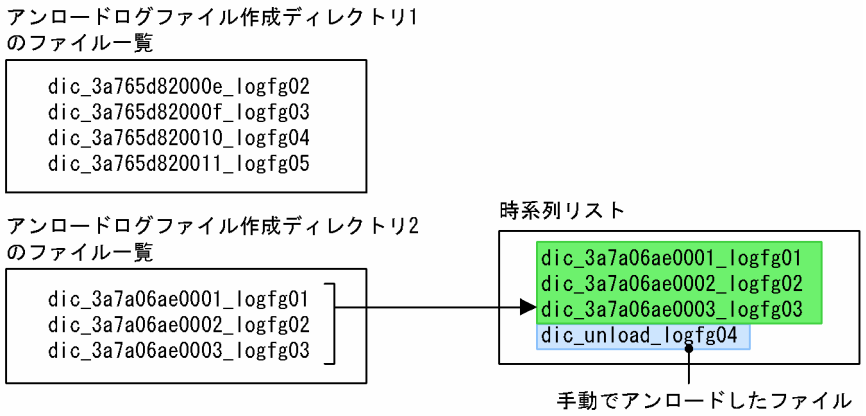
5. 時系列リストを作成し，昇順に並び替えます。

   まず，アンロードログファイル作成ディレクトリ内のファイル一覧を作成します。ここでは，アンロードログファイル作成ディレクトリが二つあるとします。



それぞれのアンロードログファイル作成ディレクトリのファイル一覧の最初のファイルを比べると，アンロードログファイル作成ディレクトリ 1 のファイルはアンロードログファイル作成ディレクトリ 2 のファイルより古いため，アンロードログファイル作成ディレクトリ 2 のファイルだけを使用します。

さらに，2.で手動でアンロードしたファイルを最後に追加して，時系列リストを作成します。



6. 5.で作成した時系列リストの順でアンロードログファイルを指定して，pdrstr コマンドを実行します。

```
pdrstr -m C:¥rdarea¥mast_rdarea¥mast_rdarea01 -r RDMASTER -b C:¥pdcopy¥backup¥dicbkup01
-l C:¥unload_dir2¥dic_3a7a06ae0001_logfg01,C:¥unload_dir2¥dic_3a7a06ae0002_logfg02,
C:¥unload_dir2¥dic_3a7a06ae0003_logfg03,C:¥unload_dir2¥dic_unload_logfg04
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

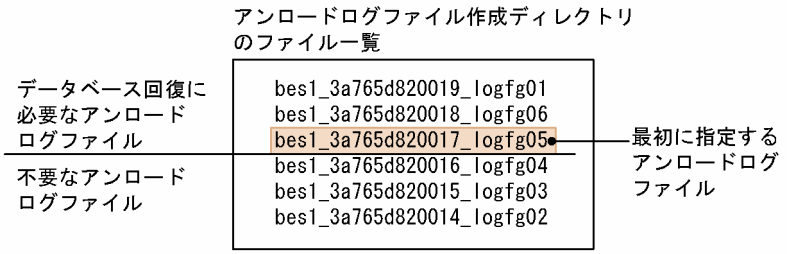
-r：マスタ用 RD エリア名を指定します。

-b：回復に使用するバックアップファイルを指定します。

-l：回復に使用するアンロードログファイルを時系列に指定します。

(6)　不要なアンロードログファイルの削除

時系列リストは，不要なアンロードログファイルを特定する場合も使用できます。**降順**の時系列リストを作成し，最初に指定するアンロードログファイルより古いファイルが不要なアンロードログファイルです。時系列リストの例を次に示します。



〔説明〕

この例では，最初に指定するアンロードログファイルが bes1_3a765d820017_logfg05 なので，不要なアンロードログファイルは bes1_3a765d820016_logfg04，bes1_3a765d820015_logfg03，及び bes1_3a765d820014_logfg02 です。

4 シンクポイントダンプファイルの運用

この章では，シンクポイントダンプファイルの運用方法について説明します。

4.1　基本項目

ここでは，シンクポイントダンプファイルを運用する前に理解して欲しいことについて説明します。

4.1.1　シンクポイントとは

HiRDB が異常終了した場合，システムログだけで回復処理をすると，HiRDB 開始からのすべてのシステムログが必要となり，システムの回復に多大な時間が掛かります。そこで，HiRDB 稼働中に一定の間隔でポイントを設け，そのポイントで回復する必要のある HiRDB 管理情報を保存することで，ポイント以前のシステムログは不要になり，システムの回復時間を短縮できます。このポイントを**シンクポイント**，シンクポイントで取得する HiRDB 管理情報を**シンクポイントダンプ**，シンクポイントダンプを格納するファイルを**シンクポイントダンプファイル**といいます。

HiRDB は前回のシンクポイント以降又は HiRDB 開始以降のデータベース更新内容をシンクポイント時にデータベースに反映します。HiRDB 管理者は，障害発生に備えてシンクポイントダンプファイルを作成してください。

4.1.2　シンクポイントダンプファイルの状態

HiRDB はシンクポイントダンプファイルを次の表に示す状態に分けて管理しています。

表 4–1　シンクポイントダンプファイルの状態

状態	説明
書き込み中	シンクポイントダンプが出力されているファイルです。
上書きできる状態	ファイル中にシステムの回復に必要なシンクポイントダンプを含まないため，上書きできる状態のファイルです。有効保証世代の対象外となるファイルの状態です。
上書きできない状態	ファイル中にシステムの回復に必要なシンクポイントダンプを含んでいるため，上書きできない状態のファイルです。有効保証世代の対象となるファイルの状態です。
予約	この状態のファイルは，シンクポイントダンプの出力対象になりません。ファイルはクローズ中の状態です。 また，HiRDB システム定義にシンクポイントダンプファイル名称を指定しただけで，その名称に対応するシンクポイントダンプファイルを作成していない場合もこの状態となります。

4.1.3　HiRDB 開始時のシンクポイントダンプファイルの状態

HiRDB を開始すると，pdlogadfg -d spd オペランドで指定したシンクポイントダンプファイルのうち，ONL と指定したファイルがすべてオープンされて，上書きできる状態になります。

なお，オープン処理に失敗したファイル，及び ONL と指定しなかったファイルは予約の状態になります。

4.1.4　シンクポイントダンプファイルの状態が変わるときは？

シンクポイントダンプが取得されて有効化※されると，シンクポイントダンプファイルの状態が変わります。シンクポイントダンプは，次に示すときに取得されます。

1. サーバの開始，又は再開始処理が完了したとき
2. サーバの終了準備処理が完了したとき
3. システムログファイルがスワップしたとき
4. 前回のシンクポイントダンプを取得してから，pd_log_sdinterval オペランドで指定したログブロック数のシステムログを取得したとき
5. 前回のシンクポイントダンプを取得してから，pd_log_sdinterval オペランドで指定した時間が経過したとき
6. pdlogsync コマンドを実行したとき

注※

シンクポイントダンプが有効化されると，メッセージ **KFPS02183-I** が出力されます。ただし，pd_spd_assurance_msg オペランドで N を指定すると，メッセージ KFPS02183-I は出力されません。

4.1.5　有効保証世代数とファイルの状態との関係

HiRDB は，1 世代のシンクポイントダンプを一つのシンクポイントダンプファイルに格納します。用意したファイルすべてにシンクポイントダンプを取得すると，最初のファイルに戻ってデータを上書きします。

HiRDB は，１世代前のシンクポイントダンプファイルを上書きできない状態にします。ただし，有効保証世代数を 2 世代にした場合，過去 2 世代のファイルを上書きできない状態にします。シンクポイントダンプの出力によるファイルの状態変化を次の図に示します。

図 4–1　シンクポイントダンプの出力によるファイルの状態変化（有効保証世代数を 1 世代にした場合）

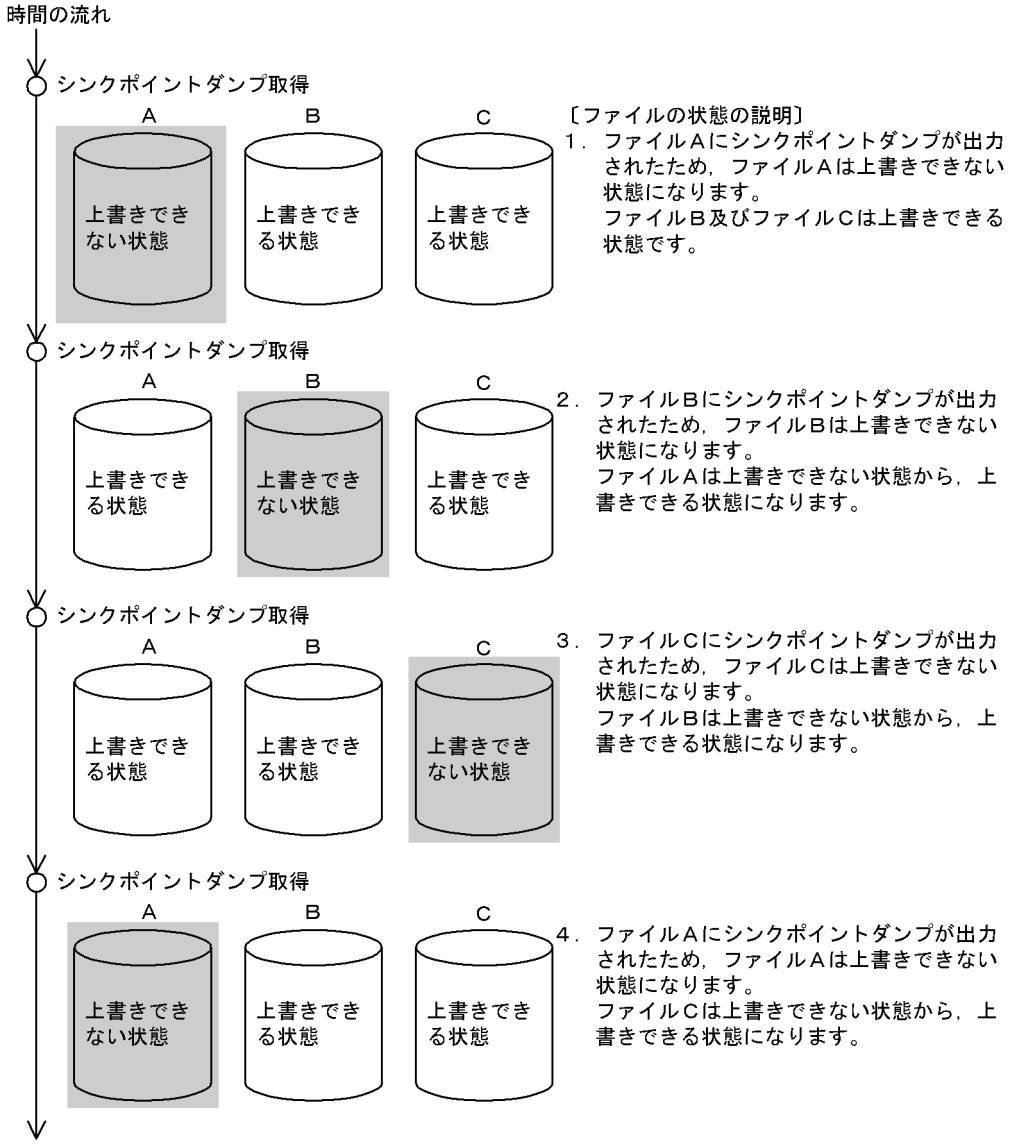
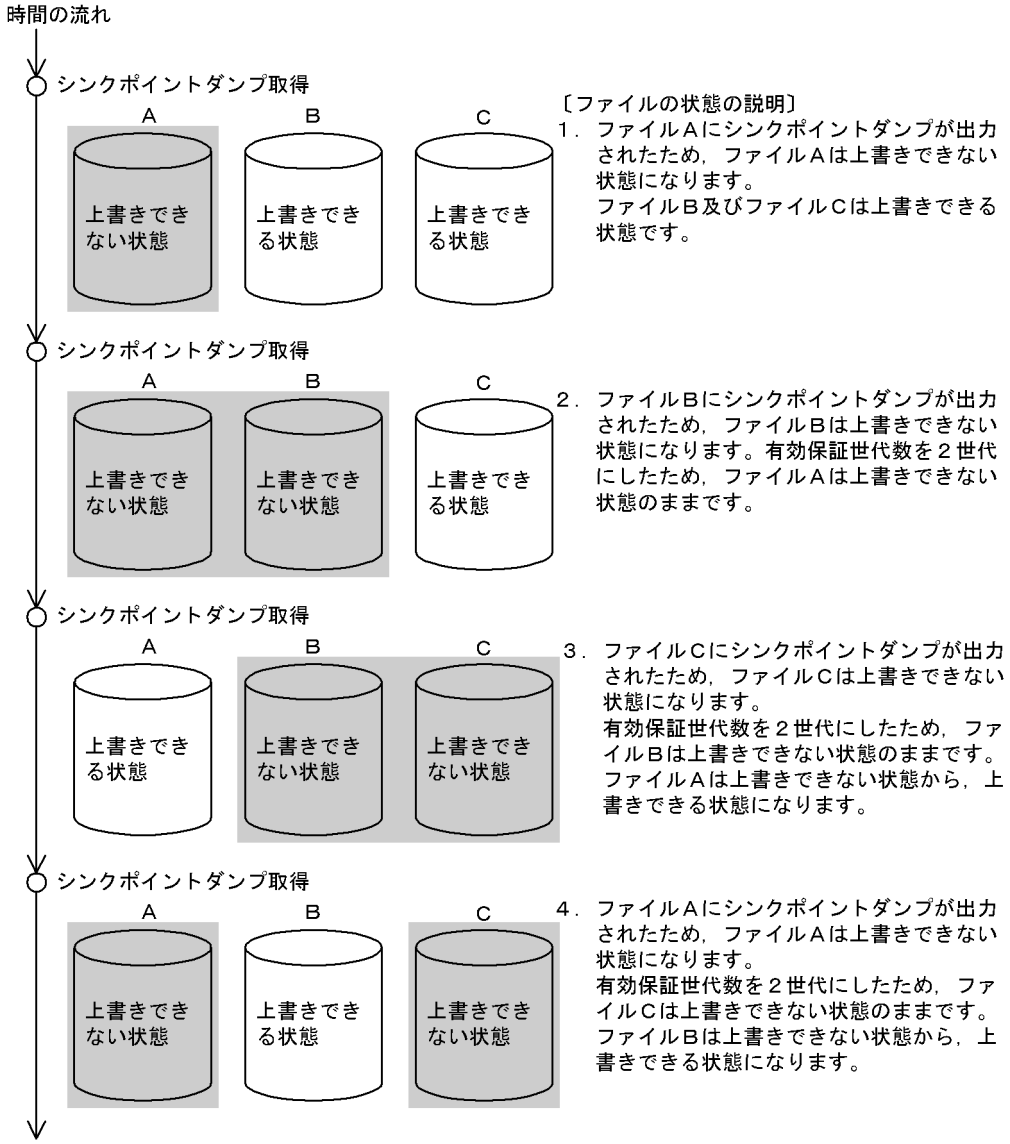


図 4–2　シンクポイントダンプの出力によるファイルの状態変化（有効保証世代数を 2 世代にした場合）



4.1.6　シンクポイントダンプファイルを操作するコマンド

シンクポイントダンプファイルを操作するコマンドを次の表に示します。

表 4–2　シンクポイントダンプファイルを操作するコマンド

コマンド名	説明
pdloginit	シンクポイントダンプファイルを初期設定します。
pdlogadpf	シンクポイントダンプファイルを，HiRDB システム定義で指定したファイルグループに割り当てます。

コマンド名	説明
pdlogopen	クローズ状態のシンクポイントダンプファイルをオープンします。上書きできる状態のファイルを予約にします。
pdlogcls	オープン状態のシンクポイントダンプファイルをクローズします。予約ファイルを上書きできる状態にします。
pdlogls	シンクポイントダンプファイルの情報を表示します。
pdlogrm	シンクポイントダンプファイルを削除します。
pdlogsync	シンクポイントダンプを取得します。

4.2　シンクポイントダンプの取得間隔の設定方法

実行者　HiRDB 管理者

4.2.1　取得間隔は pd_log_sdinterval オペランドで設定します

シンクポイントダンプの取得間隔を **pd_log_sdinterval** オペランドで設定してください。シンクポイントダンプの取得間隔は，次に示す条件で設定できます。

・システムログの出力量による設定

pd_log_sdinterval オペランドで指定したログブロック数分のシステムログが出力されるたびに，シンクポイントダンプを取得します。

・経過時間による設定

前回シンクポイントダンプを取得してから，pd_log_sdinterval オペランドで指定した経過時間が過ぎるとシンクポイントダンプを取得します。

4.2.2　取得間隔を決める目安

シンクポイントダンプの取得間隔を短くすれば，HiRDB を再開始するときのデータベース回復に掛かる所要時間が短くなります。ただし，シンクポイントダンプの取得回数が増えるため，オンライン中の性能が低下する場合があります。

逆に，シンクポイントダンプの取得間隔を長くすれば，HiRDB を再開始するときのデータベース回復に掛かる所要時間が長くなります。ただし，シンクポイントダンプの取得回数が減るため，オンライン中の性能が向上する場合があります。

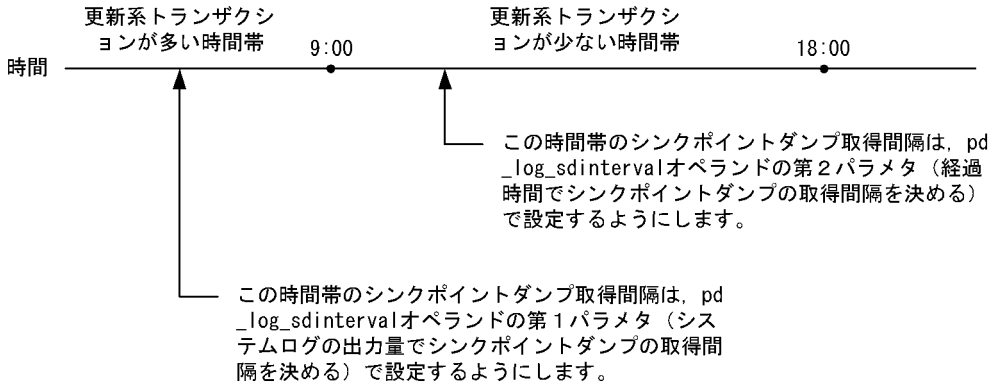
シンクポイントダンプがどの程度の間隔で取得されているかは，統計解析ユティリティのシステムの稼働に関する統計情報の「シンクポイントダンプ取得間隔時間（SYNC POINT GET INTERVAL）」で確認できます。

シンクポイントダンプの取得間隔のチューニング方法については，「シンクポイントダンプの取得間隔のチューニング」を参照してください。

4.2.3　シンクポイントダンプの取得間隔の設定例

昼間（9：00～18：00）は更新系トランザクションが少なく，夜間（18:00～9:00）は更新系トランザクションが多い場合のシステムを例にします。

（例）



〔説明〕

- 更新系トランザクションが少ない時間帯（9：00～18：00）は，経過時間を目安にシンクポイントダンプを取得します。
- 更新系トランザクションが多い時間帯（18：00～9：00）は，システムログの出力量を目安にシンクポイントダンプを取得します。
- そのほかに，必要ならば，pdlogsync コマンドでシンクポイントダンプを取得します。

4.3　シンクポイントダンプファイルの操作方法

実行者　HiRDB 管理者

ここでは，シンクポイントダンプファイルの操作方法について説明します。説明する項目は次のとおりです。

1. シンクポイントダンプファイルの状態が変わった場合
2. 上書きできる状態のファイルがない場合
3. HiRDB の稼働中にシンクポイントダンプファイルの容量を大きく（小さく）する場合
4. ファイルの状態を変更する場合
5. シンクポイントダンプファイルを新規追加する場合
6. シンクポイントダンプファイルを削除する場合
7. ファイル中のシンクポイントダンプに対応するシステムログファイルを知りたい場合
8. シンクポイントダンプファイルの有効保証世代数を増やす場合

なお，シンクポイントダンプファイルの運用で，よくある質問をＱ＆Ａにまとめました。シンクポイントダンプファイルのＱ＆Ａについては，「シンクポイントダンプファイルに関する質問」を参照してください。

4.3.1　シンクポイントダンプファイルの状態が変わった場合

HiRDB 管理者は，上書きできる状態のファイルがあるかどうかを **pdlogls** コマンドで確認してください。上書きできる状態のファイルがない場合は予約ファイルを上書きできる状態にしてください。**上書きできる状態のファイルがない状態でシンクポイントダンプが出力されると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。**なお，上書きできる状態のファイルがなくなると，HiRDB はその旨のメッセージを出力します。

4.3.2　上書きできる状態のファイルがない場合

次に示すどちらかの方法で予約ファイルを上書きできる状態にしてください。予約ファイルがない場合はシンクポイントダンプファイルを追加してください。追加方法については，「シンクポイントダンプファイルを新規追加する場合」を参照してください。

(1)　シンクポイントダンプファイルの実体がある場合

シンクポイントダンプファイルの実体があるとは，pdloginit コマンドでシンクポイントダンプファイルを作成しているという意味です。

〈手順〉

1. **pdllogls** コマンドで予約ファイルを確認してください。

pdlogls -d spd -s b001

2. **pdlogopen** コマンドで，実体がある予約ファイルを上書きできる状態にしてください。

   pdlogopen -d spd -s b001 -g syncfg01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) シンクポイントダンプファイルの実体がない場合

シンクポイントダンプファイルの実体がないとは，pdloginit コマンドでシンクポイントダンプファイルを作成しないで，pdlogadfg オペランドにファイルグループ名を指定しただけの状態のことです。

〈手順〉

1. **pdllogls** コマンドで予約ファイルを確認してください。

   pdlogls -d spd -s b001

2. **pdloginit** コマンドでシンクポイントダンプファイルを作成します。作成するシンクポイントダンプファイルの名称は，HiRDB システム定義で指定したシンクポイントダンプファイル名称と同じにしてください。

   pdloginit -d spd -s b001 -f C:¥sysfile01¥sync01 -n 5000

3. **pdlogadpf** コマンドで，1 で作成したシンクポイントダンプファイルと **pdlogadfg** オペランドで指定したファイルグループを対応付けます。

   pdlogadpf -d spd -s b001 -g syncfg01 -a C:¥sysfile01¥sync01

4. **pdlogopen** コマンドで，1 で作成したシンクポイントダンプファイルを上書きできる状態にします。

   pdlogopen -d spd -s b001 -g syncfg01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

4.3.3 HiRDB の稼働中にシンクポイントダンプファイルの容量を大きく（小さく）する場合

シンクポイントダンプファイルの容量不足が発生した場合，次に示す手順でシンクポイントダンプファイルの容量を大きくしてください。

〈手順〉

1. **pdlogls** コマンドでシンクポイントダンプファイルの状態を確認します。

   pdlogls -d spd -s b001

2. **pdlogcls** コマンドで上書きできる状態のファイルを予約にします。

   pdlogcls -d spd -s b001 -g syncfg01

3. **pdlogrm** コマンドで予約ファイルを削除します。

   pdlogrm -d spd -s b001 -f C:¥sysfile01¥sync01

4. **pdloginit** コマンドで，3 で削除したシンクポイントダンプファイルを再作成します。

   このとき，レコード数を変更前のシンクポイントダンプファイルより大きくしてください。ファイル容量を小さくする場合はレコード数を小さくしてください。

   pdloginit -d spd -s b001 -f C:¥sysfile01¥sync01 -n 5000

5. **pdlogopen** コマンドで，4 で作成したシンクポイントダンプファイルを上書きできる状態にします。

   pdlogopen -d spd -s b001 -g syncfg01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

注意事項

上書きできる状態のファイルを予約に変更するとき，すべての上書きできる状態のファイルを予約に変更しないでください。上書きできる状態のファイルがないときにシンクポイントダンプが出力されると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。したがって，上書きできる状態のファイルを一つ以上残した状態でシンクポイントダンプファイルの容量を変更してください。

4.3.4　ファイルの状態を変更する場合

pdlogopen コマンドで予約ファイルを上書きできる状態にできます。**pdlogcls** コマンドで上書きできる状態のファイルを予約にできます。

4.3.5　シンクポイントダンプファイルを新規追加する場合

シンクポイントダンプファイルを新規追加する手順を次に示します。なお，シンクポイントダンプファイルを追加する前に，マニュアル「HiRDB システム導入・設計ガイド」で，シンクポイントダンプファイルの設計について読むことをお勧めします。

(1)　HiRDB を正常終了できる場合

〈手順〉

1. **pdfstatfs** コマンドで，シンクポイントダンプを作成する HiRDB ファイルシステム領域に空きがあるかを確認します。

   pdfstatfs C:¥sysfile01

空きがない場合は HiRDB ファイルシステム領域を新規作成してください。HiRDB ファイルシステム領域の作成方法については，「HiRDB ファイルシステム領域を作成（初期設定）する方法」を参照してください。

2. **pdstop** コマンドで HiRDB を正常終了します。
3. HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 4 で追加するシンクポイントダンプファイルを指定します。
   - pdlogadfg -d spd オペランド
   - pdlogadpf -d spd オペランド
4. **pdloginit** コマンドでシンクポイントダンプファイルを追加（初期設定）します。

   pdloginit -d spd -s b001 -f C:¥sysfile01¥sync01 -n 5000
5. **pdconfchk** コマンドで HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
6. **pdstart** コマンドで HiRDB を正常開始します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) HiRDB を正常終了できない場合

HiRDB システム定義を変更するときにシステム構成変更コマンド（pdchgconf コマンド）を使用します。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

〈手順〉

1. **pdfstatfs** コマンドで，シンクポイントダンプを作成する HiRDB ファイルシステム領域に空きがあるかを確認します。

   pdfstatfs C:¥sysfile01

   空きがない場合は HiRDB ファイルシステム領域を新規作成してください。HiRDB ファイルシステム領域の作成方法については，「HiRDB ファイルシステム領域を作成（初期設定）する方法」を参照してください。
2. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
3. 使用中の HiRDB システム定義ファイルを 2 で作成したディレクトリ下にコピーします。
4. %PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 5 で追加するシンクポイントダンプファイルを指定します。
   - pdlogadfg -d spd オペランド
   - pdlogadpf -d spd オペランド
5. **pdloginit** コマンドでシンクポイントダンプファイルを追加（初期設定）します。

   pdloginit -d spd -s b001 -f C:¥sysfile01¥sync01 -n 5000

6. **pdconfchk** コマンドで，%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
7. **pdchgconf** コマンドで，HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

4.3.6 シンクポイントダンプファイルを削除する場合

pdlogrm コマンドで予約ファイルを削除できます。なお，予約以外の状態のシンクポイントダンプファイルは削除できません。

必要に応じて次に示すオペランドを削除してください。pdlogrm コマンドで削除したシンクポイントダンプファイルに対応するオペランドを削除します。

- pdlogadfg -d spd オペランド
- pdlogadpf -d spd オペランド

これらのオペランドを削除しないと，削除したシンクポイントダンプファイルは実体のないファイルになります。

4.3.7 ファイル中のシンクポイントダンプに対応するシステムログファイルを知りたい場合

pdlogls コマンドで，シンクポイントダンプファイル中のシンクポイントダンプに対応するシステムログファイルが分かります。

4.3.8 シンクポイントダンプファイルの有効保証世代数を増やす場合

シンクポイントダンプファイルの有効保証世代数を増やす手順を次に示します。

(1) HiRDB を正常終了できる場合

〈手順〉

1. **pdlogls** コマンドで，シンクポイントダンプファイルの数を確認してください。ファイル数が「変更後の有効保証世代数＋１」以上あるかを確認します。ない場合はシンクポイントダンプファイルを新規追加します。

   pdlogls -d spd -s b001

2. **pdstop** コマンドで HiRDB を正常終了させます。
3. HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 4 で追加するシンクポイントダンプファイルを指定します。
   ・pdlogadfg -d spd オペランド
   ・pdlogadpf -d spd オペランド
   また，pd_spd_assurance_count オペランドに有効保証世代数を指定します。
4. **pdloginit** コマンドでシンクポイントダンプファイルを追加（初期設定）します。
   pdloginit -d spd -s b001 -f C:¥sysfile01¥sync01 -n 5000
5. **pdconfchk** コマンドで HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
6. **pdstart** コマンドで HiRDB を正常開始します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) HiRDB を正常終了できない場合

HiRDB システム定義を変更するときにシステム構成変更コマンド（pdchgconf コマンド）を使用します。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

〈手順〉

1. **pdlogls** コマンドで，シンクポイントダンプファイルの数を確認してください。ファイル数が「変更後の有効保証世代数＋１」以上あるかを確認します。ない場合はシンクポイントダンプファイルを新規追加します。
   pdlogls -d spd -s b001
2. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
3. 使用中の HiRDB システム定義ファイルを 2 で作成したディレクトリ下にコピーします。
4. %PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 5 で追加するシンクポイントダンプファイルを指定します。
   ・pdlogadfg -d spd オペランド
   ・pdlogadpf -d spd オペランド
   また，pd_spd_assurance_count オペランドに有効保証世代数を指定します。
5. **pdloginit** コマンドでシンクポイントダンプファイルを追加（初期設定）します。
   pdloginit -d spd -s b001 -f C:¥sysfile01¥sync01 -n 5000
6. **pdconfchk** コマンドで，%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
7. **pdchgconf** コマンドで，HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

4.4　シンクポイントダンプファイルの状態遷移

HiRDB 稼働中のシンクポイントダンプファイルの状態遷移を次の表に示します。

表 4–3　HiRDB 稼働中のシンクポイントダンプファイルの状態遷移（シンクポイントダンプファイルを二重化していない場合）

イベント	シンクポイントダンプファイルの状態			
	オープン			クローズ
	上書きできる状態	書き込み中	上書きできない状態	予約
	1	2	3	4
シンクポイントダンプの出力中	→2	−	−	−
シンクポイントダンプの有効化	−	→3	→1※	−
シンクポイントダンプの出力障害	−	→4	−	−
pdlogcls コマンドの実行※	→4	−	−	−
pdlogopen コマンドの実行※	−	−	−	→1

（凡例）

−：該当しない，又は状態が遷移しないことを示しています。

→n：イベント後のシンクポイントダンプファイルの状態を示しています。

例えば，→1 の場合はイベント後にシンクポイントダンプファイルの状態は上書きできる状態になります。

注　表中の状態遷移はイベントが正常に処理されることを前提としています。

注※　上書きできない状態のファイルのうち，一番古い世代のファイルの状態遷移です。

表 4–4　HiRDB 稼働中のシンクポイントダンプファイルの状態遷移（シンクポイントダンプファイルを二重化している場合）

イベント		シンクポイントダンプファイルの状態									
		上書きできる状態			書き込み中			上書きできない状態			予約
		両系オープン	A 系オープン B 系クローズ	A 系クローズ B 系オープン	両系オープン	A 系オープン B 系クローズ	A 系クローズ B 系オープン	両系オープン	A 系オープン B 系クローズ	A 系クローズ B 系オープン	両系クローズ
		1	2	3	4	5	6	7	8	9	10
シンクポイントダンプの取得直前		→4	→5	→6	－	－	－	－	－	－	－
シンクポイントダンプの有効化		－	－	－	→7	→8	→9	→1※	→2※	→3※	－
A 系ファイルの出力障害		－	－	－	→6	→10	－	－	－	－	－
B 系ファイルの出力障害		－	－	－	→5	－	→10	－	－	－	－
pdlogcls コマンドの実行	-a 及び-b オプション省略	→10	→10	→10	－	－	－	－	－	－	－
	-a オプション指定	→3	→10	－	－	－	－	－	－	－	－
	-b オプション指定	→2	－	→10	－	－	－	－	－	－	－
pdlogopen コマンドの実行	-a 及び-b オプション省略	－	→1	→1	－	－	－	－	→7	→7	→1
	-a オプション指定	－	－	→1	－	－	－	－	－	→7	→2
	-b オプション指定	－	→1	－	－	－	－	－	→7	－	→3

（凡例）

－：該当しない，又は状態が遷移しないことを示しています。

→n：イベント後のシンクポイントダンプファイルの状態を示しています。

例えば，→1 の場合はイベント後にシンクポイントダンプファイルの状態は上書きできる状態（両系オープン）になります。

注　表中の状態遷移はイベントが正常に処理されることを前提としています。

注※　上書きできない状態のファイルのうち，一番古い世代のファイルの状態遷移です。

5 ステータスファイルの運用

この章では，ステータスファイルの運用方法について説明します。

5.1　基本項目

HiRDB を再開始するときに必要な情報をステータスファイルに格納します。ここでは，ステータスファイルを運用する前に理解してほしいことについて説明します。

5.1.1　ステータスファイルの状態

HiRDB はステータスファイルを次の表に示す状態に分けて管理しています。

表 5–1　ステータスファイルの状態

状態	説明
現用	システムステータス情報が出力されているファイルです。
予備	現時点では，システムステータス情報の出力対象になっていませんが，現用ファイルが入出力障害などで使用できなくなった場合，現用ファイルにスワップするファイルです。ファイルはオープン中の状態です。
予約	この状態のステータスファイルは，システムステータス情報の出力対象になりません。ファイルはクローズ中の状態です。 また，HiRDB システム定義にステータスファイル名称を指定しただけで，その名称に対応するステータスファイルを作成していない場合もこの状態となります。
閉塞	ステータスファイルに障害が発生して閉塞している状態です。

5.1.2　HiRDB 開始時のステータスファイルの状態

HiRDB を正常開始すると，システム定義の次に示すオペランドで最初に指定したステータスファイルが現用となります。残りのステータスファイルは予備になります。

- pd_syssts_file_name_1〜7（ユニット用ステータスファイル）
- pd_sts_file_name_1〜7（サーバ用ステータスファイル）

参考

- オープン処理に失敗したファイルは予約となります。また，障害のためオープンできなかったファイルは閉塞状態になります。
- HiRDB を再開始した場合は，前回稼働時の現用ファイルが引き継がれます。

5.1.3　ステータスファイルの状態が変わるときは？

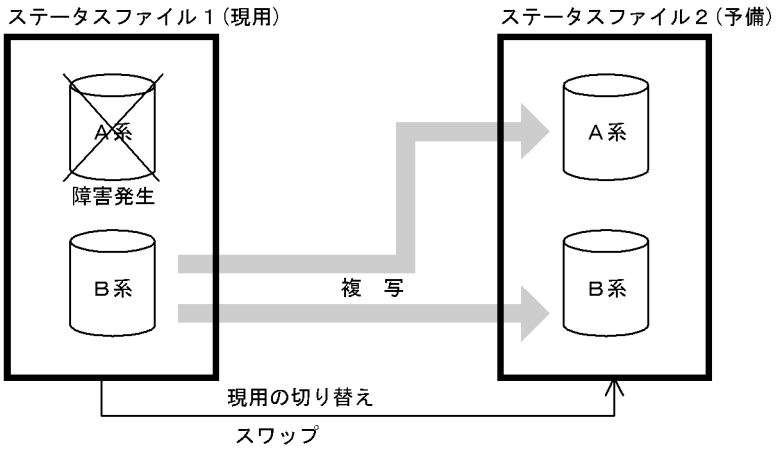
ステータスファイルがスワップすると，ステータスファイルの状態が変わります。次に示すときに，ステータスファイルはスワップします。

- ステータスファイルに障害が発生したとき
- ステータスファイルの容量が不足したとき
- pdstsswap コマンドを実行したとき

5.1.4　ステータスファイルのスワップとは

HiRDB は，A 系，B 系のどちらかの現用ファイルに障害が発生した場合，障害が発生しなかった系のファイルの内容を予備ファイルに複写します（現用ファイルには，A 系と B 系の両方に同じ内容が書き込まれています）。その後，予備ファイルを現用に切り替えます。現用だったファイルは閉塞になります。これを**ステータスファイルのスワップ**といいます。このとき，予備ファイルがないとユニットが異常終了します。ステータスファイルのスワップを次の図に示します。

図 5–1　ステータスファイルのスワップ



5.1.5　ステータスファイルを操作するコマンド

ステータスファイルを操作するコマンドを次の表に示します。

表 5–2　ステータスファイルを操作するコマンド

コマンド名	説明
pdstsinit	ステータスファイルを初期設定します。
pdstsopen	• クローズ状態のステータスファイルをオープンします。予約ファイルを予備にします。 • 初期設定したステータスファイルをオープンします。

コマンド名	説明
pdstscls	オープン状態のステータスファイルをクローズします。予備ファイルを予約にします。
pdstsswap	ステータスファイルをスワップさせます。現用ファイルを予備にします。
pdstsrm	ステータスファイルを削除します。

5.2　ステータスファイルの操作方法

実行者　HiRDB 管理者

ここでは，ステータスファイルの運用方法について説明します。説明する項目は次のとおりです。

1. ステータスファイルがスワップした場合
2. 予備ファイルがない場合
3. HiRDB の稼働中にステータスファイルの容量を大きく（小さく）する場合
4. ファイルの状態を変える場合
5. 現用ファイルを変更する場合
6. ステータスファイルを新規追加する場合
7. ステータスファイルを削除する場合
8. ステータスファイルの情報を調べる場合

なお，ステータスファイルの運用で，よくある質問をＱ＆Ａにまとめました。ステータスファイルのＱ＆Ａについては，「ステータスファイルに関する質問」を参照してください。

5.2.1　ステータスファイルがスワップした場合

HiRDB 管理者は，ステータスファイルがスワップしたときに，**pdls -d sts** コマンドで次に示すことを確認してください。

- 予備ファイルがあるかを確認してください。
- ファイル内のレコード使用率を確認してください。

(1)　予備ファイルがあるかを確認してください

HiRDB 管理者は予備ファイルがあるかどうかを確認してください。予備ファイルがない場合，予約ファイルを予備の状態にしてください。**予備ファイルがない状態でステータスファイルに障害が発生すると，ステータスファイルが片系運転になるか，又はユニットが異常終了します。**

(2)　ファイル内のレコード使用率を確認してください

ステータスファイルの容量が不足すると，HiRDB が予備ファイルの中から，現用ファイルより容量の大きいファイルを選択して，自動的にスワップします。しかし，現用ファイルより容量の大きい予備ファイルがないと，ユニットが異常終了します。したがって，容量が最も大きいファイルを現用として使用している場合，ステータスファイル内のレコード使用率を調べてください。

5.2.2　予備ファイルがない場合

次に示すどちらかの方法で予約ファイルを予備にしてください。予約ファイルがない場合はステータスファイルを追加してください。追加方法については，「ステータスファイルを新規追加する場合」を参照してください。

(1)　ステータスファイルの実体がある場合

ステータスファイルの実体があるとは，pdstsinit コマンドでステータスファイルを作成しているという意味です。

〈手順〉

1. **pdls** コマンドで実体がある予約ファイルを確認してください。ファイルの状態が CLOSE と表示されます。

   pdls -d sts -s b001

2. **pdstsopen** コマンドで予約ファイルを予備にしてください。

   pdstsopen -s b001 -n sstsfg01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2)　ステータスファイルの実体がない場合

ステータスファイルの実体がないとは，pdstsinit コマンドでステータスファイルを作成しないで，HiRDB システム定義にステータスファイルの定義をしただけの状態のことです。

〈手順〉

1. **pdls** コマンドで実体がない予約ファイルを確認してください。ファイルの状態が NONE と表示されます。

   pdls -d sts -s b001

2. **pdstsinit** コマンドでステータスファイルを作成します。作成するステータスファイルの名称は，HiRDB システム定義で定義したステータスファイル名称と同じにしてください。

   pdstsinit -s b001 -f C:¥sysfile01¥ssts1a -l 4096 -c 1000

   pdstsinit -s b001 -f C:¥sysfile01¥ssts1b -l 4096 -c 1000

3. **pdstsopen** コマンドで，2 で作成したステータスファイルをオープンします。

   pdstsopen -s b001 -n sstsfg01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

5.2.3 HiRDB の稼働中にステータスファイルの容量を大きく（小さく）する場合

レコード使用率が高くなった場合，次に示す手順でステータスファイルの容量を大きくしてください。

〈手順〉

1. **pdls** コマンドで次に示すことを確認してください。
   ・予備のステータスファイル（ファイルの状態が STANDBY と表示されます）
   ・ステータスファイルのレコード数
   pdls -d sts -s b001
2. **pdstscls** コマンドで予備ファイルを予約にします。
   pdstscls -s b001 -n sstsfg01
3. **pdstsrm** コマンドで予約ファイルを削除します。
   pdstsrm -s b001 -f C:¥sysfile01¥ssts1a
   pdstsrm -s b001 -f C:¥sysfile01¥ssts1b
4. **pdstsinit** コマンドで，3 で削除したステータスファイルを再作成します。
   このとき，レコード数を変更前のステータスファイルより大きくしてください。ファイル容量を小さくする場合はレコード数を小さくしてください。
   pdstsinit -s b001 -f C:¥sysfile01¥ssts1a -l 4096 -c 1000
   pdstsinit -s b001 -f C:¥sysfile01¥ssts1b -l 4096 -c 1000
5. **pdstsopen** コマンドで，4 で作成したステータスファイルを予備にします。
   pdstsopen -s b001 -n sstsfg01
6. **pdstsswap** コマンドで現用ファイルをスワップします。スワップ後，1〜5 の手順で現用ファイルの容量を大きくしてください。
   pdstsswap -s b001

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

注意事項

- 予備ファイルを予約に変更するとき，すべての予備ファイルを予約に変更しないでください。予備ファイルがないときにステータスファイルのスワップが発生すると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。したがって，最低一つの予備ファイルを残した状態でステータスファイルの容量を変更してください。
- pd_syssts_last_active_file 又は pd_sts_last_active_file オペランドを指定している場合は注意が必要です。手順 6 で現用ファイルを変更しているため，これらのオペランドの指定値を次回の HiRDB 開始時までに変更してください。変更しないと，HiRDB（HiRDB/パラレルサーバの場合はユニット）を開始できません。

5.2.4　ファイルの状態を変える場合

pdstsopen コマンドで予約ファイルを予備にできます。**pdstscls** コマンドで予備ファイルを予約にできます。

5.2.5　現用ファイルを変更する場合

現用ファイルを変更したい場合は，**pdstsswap** コマンドでステータスファイルをスワップさせます。現用ファイルは予備になります。

5.2.6　ステータスファイルを新規追加する場合

ステータスファイルを新規に追加する手順を次に示します。なお，ステータスファイルを追加する前に，マニュアル「HiRDB システム導入・設計ガイド」で，ステータスファイルの設計について読むことをお勧めします。

(1)　HiRDB を正常終了できる場合

〈手順〉

1. **pdfstatfs** コマンドで，ステータスファイルを作成する HiRDB ファイルシステム領域に空きがあるかを確認します。

   pdfstatfs C:¥sysfile01

   空きがない場合は HiRDB ファイルシステム領域を新規作成してください。HiRDB ファイルシステム領域の作成方法については，「HiRDB ファイルシステム領域を作成（初期設定）する方法」を参照してください。
2. **pdstop** コマンドで HiRDB を正常終了します。
3. HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 4 で追加するステータスファイルを指定します。
   - pd_syssts_file_name オペランド（ユニット用ステータスファイルの場合）
   - pd_sts_file_name オペランド（サーバ用ステータスファイルの場合）
4. **pdstsinit** コマンドでステータスファイルを追加（初期設定）します。

   pdstsinit -s b001 -f C:¥sysfile01¥ssts1a -l 4096 -c 1000

   pdstsinit -s b001 -f C:¥sysfile01¥ssts1b -l 4096 -c 1000
5. **pdconfchk** コマンドで HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
6. **pdstart** コマンドで，HiRDB を正常開始します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2)　HiRDB を正常終了できない場合

HiRDB システム定義を変更するときにシステム構成変更コマンド（pdchgconf コマンド）を使用します。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

〈手順〉

1. **pdfstatfs** コマンドで，ステータスファイルを作成する HiRDB ファイルシステム領域に空きがあるかを確認します。

   pdfstatfs C:¥sysfile01

   空きがない場合は HiRDB ファイルシステム領域を新規作成してください。HiRDB ファイルシステム領域の作成方法については，「HiRDB ファイルシステム領域を作成（初期設定）する方法」を参照してください。
2. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
3. 使用中の HiRDB システム定義ファイルを 2 で作成したディレクトリ下にコピーします。
4. %PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義に次に示すオペランドを追加します。このオペランドには 5 で追加するステータスファイルを指定します。
   - pd_syssts_file_name オペランド（ユニット用ステータスファイルの場合）
   - pd_sts_file_name オペランド（サーバ用ステータスファイルの場合）
5. **pdstsinit** コマンドでステータスファイルを追加（初期設定）します。

   pdstsinit -s b001 -f C:¥sysfile01¥ssts1a -l 4096 -c 1000

   pdstsinit -s b001 -f C:¥sysfile01¥ssts1b -l 4096 -c 1000
6. **pdconfchk** コマンドで，%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
7. **pdchgconf** コマンドで，HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

5.2.7　ステータスファイルを削除する場合

pdstsrm コマンドで予約及び閉塞ファイルを削除できます。現用及び予備ファイルは削除できません。

必要に応じて次に示すオペランドを削除してください。pdstsrm コマンドで削除したステータスファイルに対応するオペランドを削除します。

- pd_syssts_file_name オペランド（ユニット用ステータスファイルの場合）

- pd_sts_file_name オペランド（サーバ用ステータスファイルの場合）

これらのオペランドを削除しないと，削除したステータスファイルは実体のないファイルになります。このため，次に示すオペランドに stop を指定していると，HiRDB（HiRDB/パラレルサーバの場合はユニット）を開始できなくなります。

- pd_syssts_initial_error
- pd_sts_initial_error

5.2.8　ステータスファイルの情報を調べる場合

(1)　論理ファイルの情報を調べる場合

pdls -d sts コマンドを実行すると，次に示す情報が表示されます。

- 論理ファイル名
- 論理ファイルの状態
- ファイル内のレコード使用率
- ファイル内の連続空きレコード数
- ファイル内の管理レコード数
- 物理ファイルの稼働系
- 物理ファイルの状態
- レコード長
- レコード数
- 物理ファイル名

(2)　物理ファイルの情報を調べる場合

pdcat -d sts コマンドを実行すると，次に示す情報が表示されます。

- 初期設定年月日と時刻
- 現用決定年月日と時刻
- レコード長
- レコード数
- ファイル内のレコード使用率
- ファイル内の連続空きレコード数
- ファイル内の管理レコード数

5.3　ステータスファイルの状態遷移

HiRDB 稼働中のステータスファイルの状態遷移を次の図，及び次の表に示します。

図 5–2　HiRDB 稼働中のステータスファイルの状態遷移

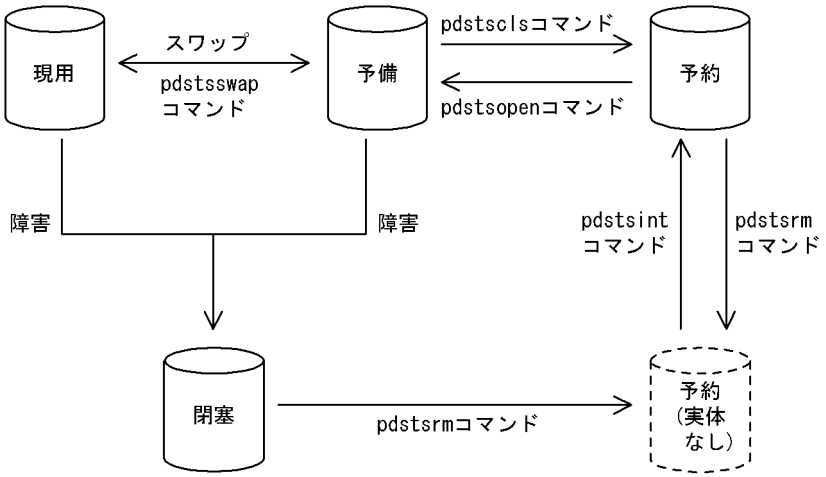


表 5–3　HiRDB 稼働中のステータスファイルの状態遷移

イベント				ステータスファイルの状態				
				オープン		クローズ		
				現用	予備	予約		閉塞
				ACTIVE※	STAND BY※	CLOSE※	NONE※	BLOCK ADE※
				1	2	3	4	5
ステータスファイルの容量不足				→2	→1	－	－	－
pdstsswap コマンドの実行				→2	→1	－	－	－
pdstscls コマンドの実行				－	→3	－	－	－
pdstsopen コマンドの実行				－	－	→2	－	－
出力障害	両系障害			→5	－	－	－	－
	片系障害	予備ファイルあり		→5	－	－	－	－
		予備ファイルなし	片系運転可 両系運転中	－	－	－	－	－
			片系運転可 片系運転中	→5	－	－	－	－
			片系運転不可	→5	－	－	－	－

（凡例）

－：該当しない，又は状態が遷移しないことを示しています。

→n：イベント後のステータスファイルの状態を示しています。

例えば，→1 の場合はイベント後にステータスファイルの状態は現用になります。

注

- 表中の状態遷移はイベントが正常に処理されることを前提としています。
- 予約又は閉塞ファイルに対して pdstsinit 又は pdstsrm コマンドを実行しても，pdls コマンドで表示されるステータスファイルの状態は変わりません。

注※

pdls -d sts コマンドの実行結果（ステータスファイルの状態）です。

6 バックアップの取得方法

障害の発生に備えて，HiRDB 管理者はデータベースのバックアップを取得しておいてください。
この章では，バックアップの取得方法について説明します。

6.1　バックアップについて

HiRDB 管理者は，データベースの障害に備えてバックアップを取得しておく必要があります。バックアップは，データベース複写ユティリティ（pdcopy コマンド）で取得します。ここでは，バックアップを取得するときに知っておいて欲しい基本的なことについて説明します。

また，バックアップを取得するときのオプション機能についても「オプション項目」で説明します。

6.1.1　基本項目

(1)　バックアップの取得単位

バックアップの取得単位は，データベース複写ユティリティのオプションで指定します。バックアップは次に示す単位で取得できます。

- システム単位（すべての RD エリア）
- ユニット単位（ユニット下のすべての RD エリア）※
- サーバ単位（サーバ下のすべての RD エリア）※
- RD エリア単位（個々の RD エリア）

注※　HiRDB/パラレルサーバ限定の機能です。

(2)　バックアップの取得時期

バックアップは定期的に取得してください。ただし，次に示す作業をする場合は必ずバックアップを取得してください。このタイミングで取得していないと，障害が発生したときに最新の同期点までデータベースを回復できなくなります。

- ログレスモードの UAP 又はユティリティを実行する場合，その実行前後
- 更新前ログ取得モードの UAP 又はユティリティを実行した場合，その実行後
- RD エリアに対してデータベース構成変更ユティリティを実行した場合，その実行後
  なお，RD エリアの移動を実行する場合，その実行前にも取得してください。
- システムログを使用した回復を実行した後
- アンロードログファイルだけを使用して回復を実行する前
- HiRDB のバージョンアップをする場合，その実行前後
- プラグインの登録後（pdplgrgst コマンドの実行後）
- プラグインの使用を中止した後（pdplugin オペランドを削除した後）

- RD エリアをインメモリ化した場合，又はインメモリデータバッファとインメモリ RD エリアの同期を取った場合，その実行後

また，次に示す作業をする場合にバックアップを取得しておくと，障害が発生したときに回復に掛かる時間を短縮できます。

- システム用 RD エリアのバックアップを取得する必要がある処理の実行後

  システム用 RD エリアのバックアップを取得する必要がある処理とは，表「同時にバックアップを取得する必要がある RD エリア」で，マスタディレクトリ用 RD エリア，データディレクトリ用 RD エリア，又はデータディクショナリ用 RD エリアのバックアップ取得が必要と記載されている処理です。
- 表のデータを大量更新した後

(3) バックアップファイルを格納するサーバマシン

バックアップファイルは，HiRDB が稼働するサーバマシンであればどこにでも作成できます。バックアップを取得した RD エリアと同じサーバマシンに作成する必要はありません。CMT 又は DAT などのデバイスが，ほかのサーバマシンにある場合に適用してください。

バックアップファイルを格納するサーバマシンは，データベース複写ユティリティ(pdcopy コマンド)のオプションで指定できます。

(4) データベース複写ユティリティ以外で RD エリアのバックアップを取得する場合の注意事項

データベース複写ユティリティ以外で RD エリアのバックアップを取得する場合は，次に示すどれかの操作をしてからバックアップを取得してください。

- pdhold -c コマンドで RD エリアを閉塞かつクローズ状態にしてください。
- pdhold -b コマンドで RD エリアをバックアップ閉塞状態にしてください。
- HiRDB を正常終了させてください。

(5) リスト用 RD エリアについて

リスト用 RD エリアのバックアップは取得できません。リストの基になった表があればリストを再作成できるため，バックアップを取得する必要はありません。

(6) バックアップファイルのサイズについて（重要）

ディスクの残容量に余裕があるのに，データベース複写ユティリティ実行時にディスク容量不足を示す旨のメッセージが出力された場合は，バックアップファイルの大きさが上限を超えている可能性があります。バックアップファイルの最大値は次のとおりです。

条件	バックアップファイルの最大値
HiRDB ファイルシステム領域にバックアップファイルを作成する場合	1,048,575 メガバイト
バックアップファイルを Windows のファイルで作成する場合	2^{63}–1 バイト

(7)　バックアップファイルの内容確認

データベース複写ユティリティで取得したバックアップの情報を **pdbkupls** コマンドで確認できます。例えば，次に示す情報を確認できます。確認できる情報の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

- バックアップの取得日時
- バックアップ取得対象 RD エリアの名称
- バックアップ取得モード（-M オプション）の指定値

6.1.2　オプション項目

(1)　差分バックアップ機能

差分バックアップ機能とは，前回のバックアップ取得時点からの差分情報だけをバックアップとして取得する機能です。このため，バックアップの取得処理時間を短縮できます。データベースの容量が多くてデータ更新量が少ない場合に，差分バックアップ機能の使用を検討してください。差分バックアップ機能の運用方法については，「差分バックアップの取得」を参照してください。

(2)　バックアップ閉塞

次に示す場合は，バックアップ取得対象 RD エリアをバックアップ閉塞する必要があります。

- pdcopy コマンド以外でバックアップを取得する場合（ほかの製品の機能でバックアップを取得する場合）

バックアップ閉塞については，「バックアップ閉塞を使用してバックアップを取得する場合（pdcopy コマンド以外でバックアップを取得する場合）」を参照してください。

(3)　更新凍結コマンド（pddbfrz コマンド）

更新凍結コマンドとは，ユーザ LOB 用 RD エリア中の満杯データページ（すべて割り当て済み）の HiRDB ファイルを更新凍結状態にする機能です。更新凍結状態の HiRDB ファイルはバックアップを再度取得する必要がなくなります。このため，ユーザ LOB 用 RD エリアのバックアップ取得時間を短縮できることがあります。更新凍結コマンドを使用したバックアップの取得方法については，「更新凍結コマンド（pddbfrz コマンド）を使用してバックアップを取得する場合」を参照してください。

(4)　NetBackup 連携機能

NetBackup 連携機能を使用すると，データベース複写ユティリティ（pdcopy）又はデータベース回復ユティリティ（pdrstr）で使用するバックアップファイルを NetBackup サーバが管理する媒体上に作成できます。NetBackup 連携機能を使用する場合は JP1/VERITAS NetBackup Agent for HiRDB License が必要になります。NetBackup 連携機能については，マニュアル「JP1/VERITAS NetBackup v4.5 Agent for HiRDB License」を参照してください。

なお，NetBackup 連携機能を使用していて，HiRDB 稼働中に標準ホスト名（コンピュータ名）を変更する場合は，「NetBackup 連携機能使用時に標準ホスト名（コンピュータ名）を変更する場合」を，系切り替え構成で NetBackup 連携機能を使用する場合は，「NetBackup 連携機能を使用する場合の注意事項」を参照して設定してください。

(5)　テープ装置アクセス機能

DAT，DLT，又は LTO 上にあるファイルを，テープ装置アクセス機能を使用してアクセスできます。次に示すファイルに対してテープ装置アクセス機能が適用できます。

- バックアップファイル（pdcopy コマンド又は pdrstr コマンドの-b オプションに指定するバックアップファイル）

テープ装置アクセス機能については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.2　バックアップ取得モード

6.2.1　バックアップ取得モードの種類

データベース複写ユティリティ(**pdcopy** コマンド)の**-M**オプションでバックアップ取得モードを選択できます。バックアップ取得モードを次の表に示します。

表 6–1　バックアップ取得モード

バックアップ取得モード（-Mオプションの指定値）	モードの説明	バックアップ取得時点への RD エリアの回復方法の違い
参照・更新不可能モード（ｘ）	バックアップ取得中に，バックアップ対象 RD エリアを参照及び更新できません。バックアップを取得する前に，対象 RD エリアを pdhold -c コマンドで閉塞かつクローズ状態にする必要があります。	ここで取得したバックアップを使用して，データベースをバックアップ取得時点に回復できます。 また，システムログを使用すれば，バックアップ取得時点以降の任意の同期点に回復できます。
参照可能モード（ｒ）	バックアップ取得中に，バックアップ対象 RD エリアは参照だけできます。更新はできません。	
更新可能モード（ｓ）※1	バックアップ取得中に，バックアップ対象 RD エリアを参照及び更新できます。	データベースをバックアップ取得時点には回復できません。バックアップ取得時点以降の任意の同期点への回復だけとなります。したがって，データベースを回復するには，バックアップ及びバックアップ取得直前のシンクポイントからのシステムログ※2が必要になります。

注※1

- ログレスモード又は更新前ログ取得モードの UAP（ユティリティを含む）の実行中に，更新可能モードでバックアップを取得しないでください。
- pdcopy コマンドを更新可能モードで実行する場合は，対象 RD エリアをバックアップ閉塞する必要があります。

注※2

データベース複写ユティリティの処理結果出力ファイルに，RD エリアを回復するときに必要なシステムログファイルのラン ID 及び世代番号が出力されます。

ポイント

RD エリアの構成変更後に取得するバックアップは，次に示すバックアップ取得モードで取得してください。

- 参照・更新不可能モード（ｘ）

- 参照可能モード（ｒ）

6.2.2　データベースの更新ログ取得方式との関係

データベースの更新ログ取得方式によって，バックアップ取得時に指定できるバックアップ取得モードが異なります。更新ログ取得方式によって指定できるバックアップ取得モードを次の表に示します。データベースの更新ログ取得方式については，「データベースの更新ログを取得しないときの運用」を参照してください。

表 6–2　更新ログ取得方式によって指定できるバックアップ取得モード

データベースの 更新ログ取得方式	指定するバックアップ取得モード （-Mオプションの指定値）
ログ取得モード	参照・更新不可能モード（ｘ） 参照可能モード（ｒ） 更新可能モード（ｓ）
更新前ログ取得モード	参照・更新不可能モード（ｘ） 参照可能モード（ｒ）
ログレスモード	

注意事項

1. ログレスモード又は更新前ログ取得モードの UAP（又はユティリティ）の実行中に，更新可能モード（-M s 指定）でバックアップを取得しないでください。
2. ログレスモード又は更新前ログ取得モードの UAP（又はユティリティ）の実行後は，次に示すモードでバックアップを取得してください。
   ・参照・更新不可能モード（-M ｘ指定）
   ・参照可能モード（-M ｒ指定）

6.3　同時にバックアップを取得する必要がある RD エリア

バックアップを取得する RD エリアは，処理対象の RD エリアだけではなく，その処理によって更新された RD エリアについても必要です。同時にバックアップを取得する必要がある RD エリアを次の表に示します。

表 6–3　同時にバックアップを取得する必要がある RD エリア

前回のバックアップ取得以降に実行した処理	バックアップを取得する RD エリアの種類									
	MST	DIR	DIC	DIC LOB	USR	USR LOB	LOB indx	LOB data	REG	REG LOB
ALTER INDEX	○	○	○		○		○ ※３７			
ALTER PROCEDURE			○	○						
ALTER ROUTINE			○	○						
ALTER TABLE	○ ※１	○ ※１	○		○					
ALTER TRIGGER			○	○						
CREATE CONNECTION SECURITY			○							
CREATE FUNCTION			○	○						
CREATE INDEX	○	○	○		○		○ ※１５			
CREATE PROCEDURE			○	○						
CREATE SCHEMA			○							
CREATE SEQUENCE	○		○		○					
CREATE TABLE	○	○ ※２	○	○ ※３４	○	○ ※９		○ ※１６		
CREATE TRIGGER			○	○						
CREATE TYPE			○	○ ※２４						
CREATE VIEW	○		○							
DROP CONNECTION SECURITY			○							
DROP DATA TYPE			○	○ ※２４						
DROP FUNCTION			○	○						

前回のバックアップ取得以降に実行した処理	バックアップを取得する RD エリアの種類									
	MST	DIR	DIC	DIC LOB	USR	USR LOB	LOB indx	LOB data	REG	REG LOB
DROP INDEX	○	○	○		○		○ ※17			
DROP PROCEDURE			○	○						
DROP SCHEMA	○ ※3	○ ※3	○	○ ※14	○ ※38	○ ※10	○ ※18	○ ※19		
DROP SEQUENCE	○		○		○					
DROP TABLE	○	○ ※8	○	○ ※34	○ ※38	○ ※9	○ ※20	○ ※21		
DROP TRIGGER			○	○						
DROP VIEW	○		○							
上記以外の定義系 SQL	○ ※7		○							
INSERT					○ ※13，38，39	○ ※13	○ ※22	○ ※23		
PURGE TABLE			○ ※35		○ ※38	○ ※9	○ ※20	○ ※21		
UPDATE					○ ※13，38，39	○ ※13	○ ※22	○ ※23		
上記以外の操作系 SQL					○ ※13，38	○ ※13	○ ※22	○ ※23		
DBPARTNER による表の定義及び削除	○	○ ※2，※8	○		○	○ ※9				
DBPARTNER による表データの抽出及び更新					○ ※13，38	○ ※13				
データベース作成ユティリティ			○ ※35		○ ※39	○ ※9	○ ※20	○ ※21		

前回のバックアップ取得以降に実行した処理		バックアップを取得する RD エリアの種類									
		MST	DIR	DIC	DIC LOB	USR	USR LOB	LOB indx	LOB data	REG	REG LOB
データベース構成変更ユティリティ	RD エリアの追加	○		○	○ ※30	○ ※5	○ ※11	○ ※11	○ ※11		
	RD エリアの拡張	○		○	○ ※30	○ ※5	○ ※11	○ ※11	○ ※11	○ ※25	○ ※26
	RD エリアの削除	○		○							
	RD エリアの再初期化	○	○ ※4	○	○ ※30	○ ※5	○ ※11，※12			○ ※25	○ ※26
	RD エリアの属性変更	○		○	○ ※40	○ ※40	○ ※40	○ ※40	○ ※40	○ ※40	○ ※40
データベース再編成ユティリティ	リロード			○ ※6，※35，※36	○ ※6	○ ※5，※36	○ ※20	○ ※21	○ ※31		
	再編成			○ ※6，※35，※36	○ ※6	○ ※5，※36	○ ※20	○ ※21	○ ※31		
	インデクス再作成					○ ※32	○ ※33	○ ※33			
	インデクス再編成					○					
	ディクショナリ表のメンテナンス	○	○	○	○	○	○	○	○	○	○
ディクショナリ搬出入ユティリティ	表定義情報の搬入	○	○	○	○ ※34	○			○ ※9		
	ストアドプロシジャの搬入			○	○						
整合性チェックユティリティ				○ ※35		○ ※35					
最適化情報収集ユティリティ				○							

前回のバックアップ取得以降に実行した処理	バックアップを取得する RD エリアの種類									
	MST	DIR	DIC	DIC LOB	USR	USR LOB	LOB indx	LOB data	REG	REG LOB
レジストリ機能初期設定ユティリティ									○ ※25	○ ※26
リバランスユティリティ			○	○	○	○ ※9	○ ※20	○ ※21		
データベース定義ユティリティ	定義系 SQL と同じ									
pdparaload コマンド			○ ※35		○ ※39	○ ※9	○ ※20	○ ※21		
pdplgrgst コマンド			○	○						
プラグインの削除	○	○	○	○	○	○ ※27	○ ※28	○ ※29	○	○
HiRDB のバージョンアップ	○	○	○	○	○	○	○	○	○	○

（凡例）

○：同時にバックアップを取得する必要がある RD エリア（空白の欄は，バックアップを取得する必要がないことを示します）

MST：マスタディレクトリ用 RD エリア

DIR：データディレクトリ用 RD エリア

DIC：データディクショナリ用 RD エリア

DIC LOB：データディクショナリ LOB 用 RD エリア

USR：ユーザ用 RD エリア（一時表用 RD エリアは除きます）

USR LOB：ユーザ LOB 用 RD エリア

REG：レジストリ用 RD エリア

REG LOB：レジストリ LOB 用 RD エリア

LOB indx：ユーザ LOB 用 RD エリア（プラグインインデクスを格納している場合）

LOB data：ユーザ LOB 用 RD エリア（抽象データ型を格納している場合）

注

オブジェクト格納用のデータディクショナリ LOB 用 RD エリアのバックアップを取得する場合は，データベース複写ユティリティの-M オプションを省略するか，-M オプションに x 又は r を指定してください。

注※1

インデクスを構成する列を削除した場合，又はインデクスを構成する列より列 ID が小さい列を削除した場合に必要です。列 ID は，ディクショナリ表（SQL_COLUMNS 表）の COLUMN_ID 列を検索すれば分かります。

注※2

横分割表又はクラスタキーを定義した場合に必要です。

注※3

削除対象のスキーマに，表又はインデクスが定義されていた場合に必要です。

注※4

対象 RD エリアにインデクスを格納している場合に必要です。

注※5

ユーザ用 RD エリアを対象とする場合に必要です。

注※6

ディクショナリ表を再編成した場合に必要です。

注※7

基表に対する REVOKE アクセス権限によって，アクセス権限を失ったビュー表が削除された場合に必要です。

注※8

次に示す場合に必要です。

- 横分割表を削除した場合
- 削除対象の表にインデクスが定義されている場合
- 削除対象の表にクラスタキーが定義されている場合

注※9

対象表が LOB 列を定義している場合に必要です。

注※10

削除対象のスキーマに，LOB 列を定義した表がある場合に必要です。

注※11

ユーザ LOB 用 RD エリアを対象とする場合に必要です。

注※12

LOB 列を定義している表を格納しているユーザ用 RD エリアを再初期化した場合に必要です。

注※13

更新した表に LOB 列が定義されていない場合は，更新した表を格納するユーザ用 RD エリアのバックアップを取得します。更新した表に LOB 列が定義されている場合は，表「更新した表に LOB 列が定義されている場合にバックアップを取得する RD エリア」に示す RD エリアのバックアップを取得します。

注※14

削除対象のスキーマに，ルーチン，参照制約動作が CASCADE で定義されている表，又はトリガが定義されている場合に必要です。

注※15

プラグインインデクスを作成する場合に必要です。

注※16

表に定義した抽象データ型に LOB 属性がある場合に必要です。

注※17

プラグインインデクスを削除する場合に必要です。

注※18

削除するスキーマにプラグインインデクスを定義した表がある場合に必要です。

注※19

削除するスキーマに LOB 属性を定義した表がある場合に必要です。

注※20

処理対象の表がプラグインインデクスを定義している場合に必要です。

注※21

処理対象の表が LOB 属性を定義している場合に必要です。

注※22

プラグインインデクスを定義した属性を更新した場合に必要です。

注※23

LOB 属性を更新した場合に必要です。

注※24

SQL 手続きで記述された関数定義がある場合に必要です。

注※25

レジストリ用 RD エリアが処理対象の場合に必要です。

注※26

レジストリ LOB 用 RD エリアが処理対象の場合に必要です。

注※27

プラグインが提供する抽象データ型を定義した表に，LOB 属性を定義した場合に必要です。

注※28

プラグインインデクスを定義している場合に必要です。

注※29

プラグインが提供する抽象データ型に LOB 属性を定義した場合に必要です。

注※30

データディクショナリ LOB 用 RD エリアを対象とする場合に必要です。

注※31

対象表が LOB 列を定義している場合に必要です。なお，LOB 列構成基表だけを再編成する場合（-j オプション指定なし)もユーザ用 RD エリアと一緒に取得する必要があります。

注※32

表格納 RD エリア及びインデクス格納 RD エリアを対で取得する必要があります。

注※33

プラグインインデクスを処理対象とする場合も，ログレス閉塞に備えて表格納 RD エリア（ユーザ用 RD エリア）を取得する必要があります。

注※34

参照制約動作が CASCADE の場合に必要です。

注※35

参照表，被参照表，又は検査制約を定義した表が処理対象にある場合に必要です。

注※36

検査保留状態を変更した場合に必要です。

注※37

プラグインインデクスを変更する場合に必要です。

注※38

処理対象となる表にインデクスを定義している場合，表を格納している RD エリアだけでなく，インデクスを格納している RD エリアも必要です。

注※39

順序数生成子を使用してデータの更新をした場合，表やインデクスを格納している RD エリアだけなく，順序数生成子格納 RD エリアも必要です。

注※40

構成変更対象の RD エリアの場合に必要です。

表 6–4　更新した表に LOB 列が定義されている場合にバックアップを取得する RD エリア

更新種別及び更新条件			ユーザ用	ユーザ LOB 用
INSERT 文	LOB 列以外の列にデータあり	LOB 列が null	○	–
		LOB 列にデータあり	○	○
DELETE 文	LOB 列以外の列にデータあり	LOB 列が null	○	–
		LOB 列にデータあり	○	○
UPDATE 文	LOB 列以外の列に更新あり	LOB 列に更新なし	○	–

更新種別及び更新条件				ユーザ用	ユーザ LOB 用
		LOB 列に更新あり		○	○
	LOB 列以外の列に更新なし	LOB 列に更新あり	null→データありに更新	○	○
			データあり→null に更新	○	○
			データ A →データ B に更新	−	○

（凡例）

○：バックアップを取得する必要があります。

−：バックアップを取得する必要がありません。

6.4　バックアップの取得例

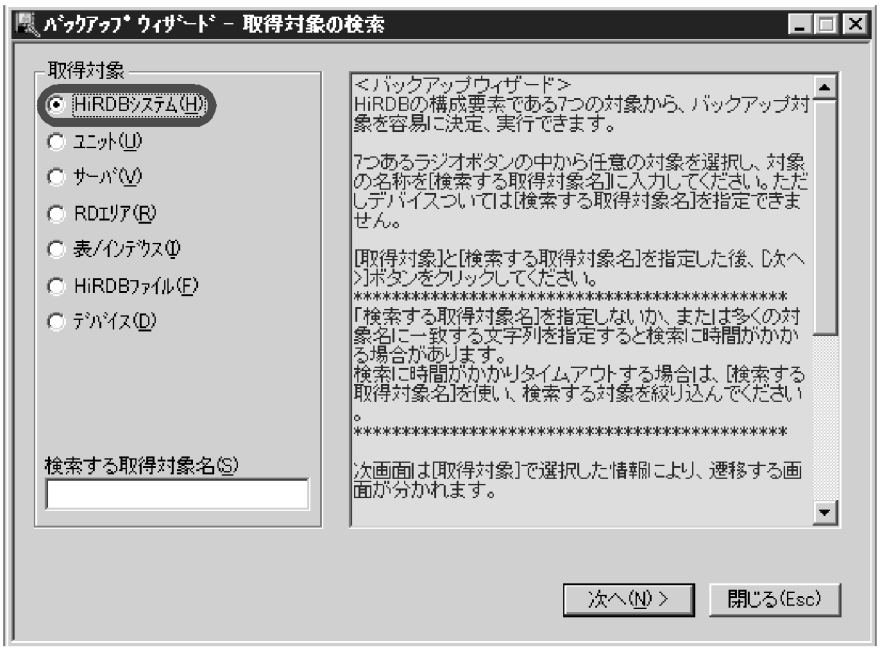
実行者　HiRDB 管理者

6.4.1　例題１（HiRDB Control Manager のウィザードでシステム単位にバックアップを取得する場合）

HiRDB Control Manager のバックアップウィザードを使用して，システム単位のバックアップを取得します。

〈手順〉

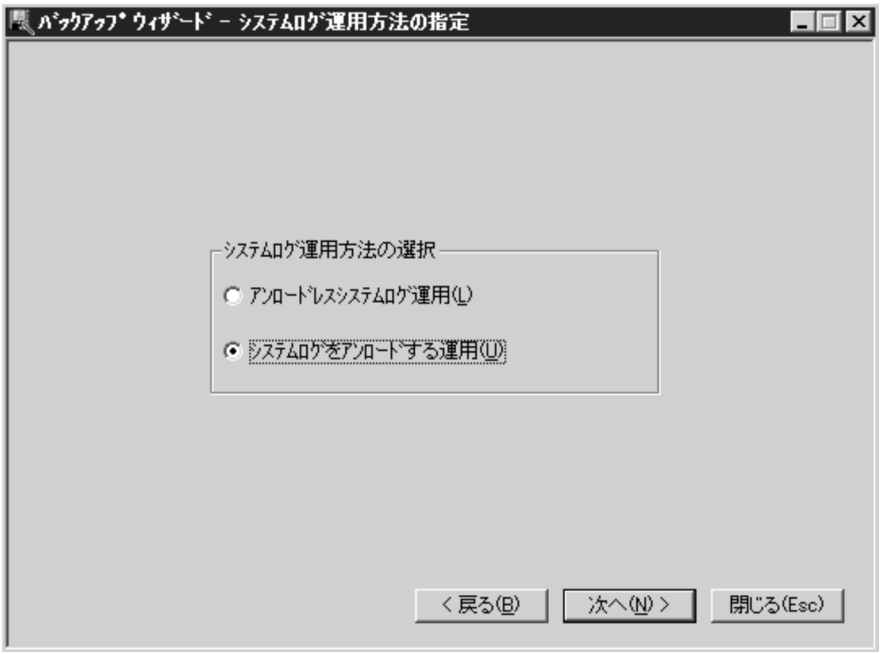
1. HiRDB Control Manager - Console を起動します。
   起動方法については，「HiRDB Control Manager - Console の起動方法」を参照してください。
2. 操作対象の HiRDB サーバを登録します。
   既に登録されている場合は，この手順は必要ありません。登録方法については，「管理 HiRDB の登録方法」を参照してください。
3. タブメニューの［**バックアップ**］－［**ウィザード**］を選択して，［バックアップウィザード - 取得対象の検索］画面を表示します。
   取得対象として，「HiRDB システム」を選択します。



4. **［次へ］**をクリックして，［バックアップウィザード - HiRDB システムの表示］画面を表示します。
   ［HiRDB システム］欄から，バックアップ対象の HiRDB システム名を選択します。



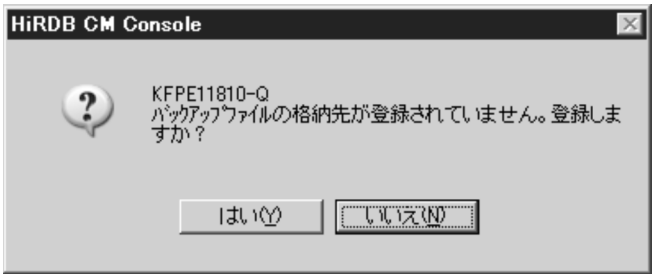
5. **[次へ]** をクリックして，［バックアップウィザード - システムログ運用方法の指定］画面を表示します。
   システムログ運用方法の選択については，システムが自動的に正しい値を設定するため，変更する必要はありません。



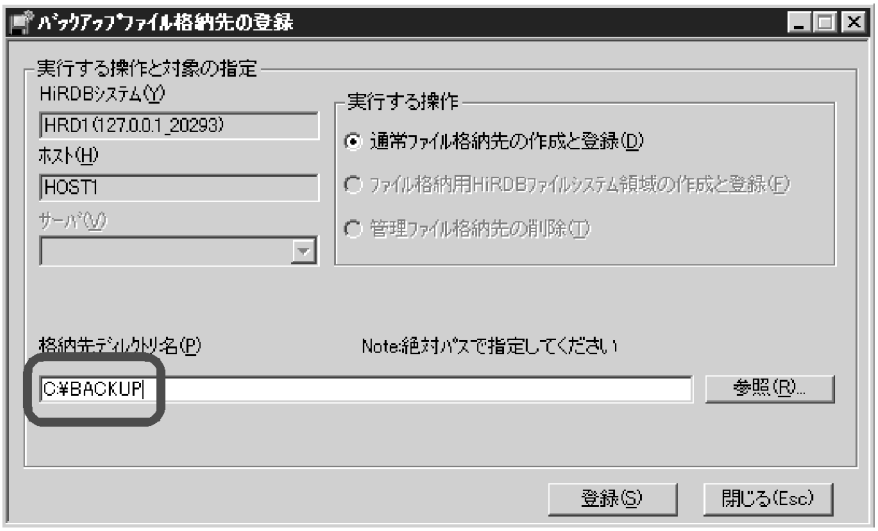
6. **[次へ]** をクリックして，［バックアップウィザード - バックアップ取得先の指定］画面を表示します。
   ここでは，次の内容を指定します。
   - バックアップファイルのファイル種別
   - 格納先のホスト名
   - バックアップファイル名
     バックアップファイル名は，**[バックアップ取得先]** をクリックして指定します。



7. バックアップファイルの格納先が登録されていない場合は，次のポップアップが表示されます。
   バックアップファイルの格納先が登録済みの場合は，この画面は表示されません。



8. **[はい]** をクリックして，［バックアップファイル格納先の登録］画面を表示します。
   ［格納先ディレクトリ名］欄に，バックアップファイルの格納先ディレクトリを指定します。
   バックアップファイルの格納先が登録済みの場合は，この画面は表示されません。



9. **[登録]** をクリックして，［バックアップファイル格納先の指定］画面を表示します。

［バックアップファイル名］欄に，バックアップファイル名を入力します。



10. **[追加]** をクリックします。

［バックアップ取得先］欄に，追加したバックアップファイル名が表示されます。

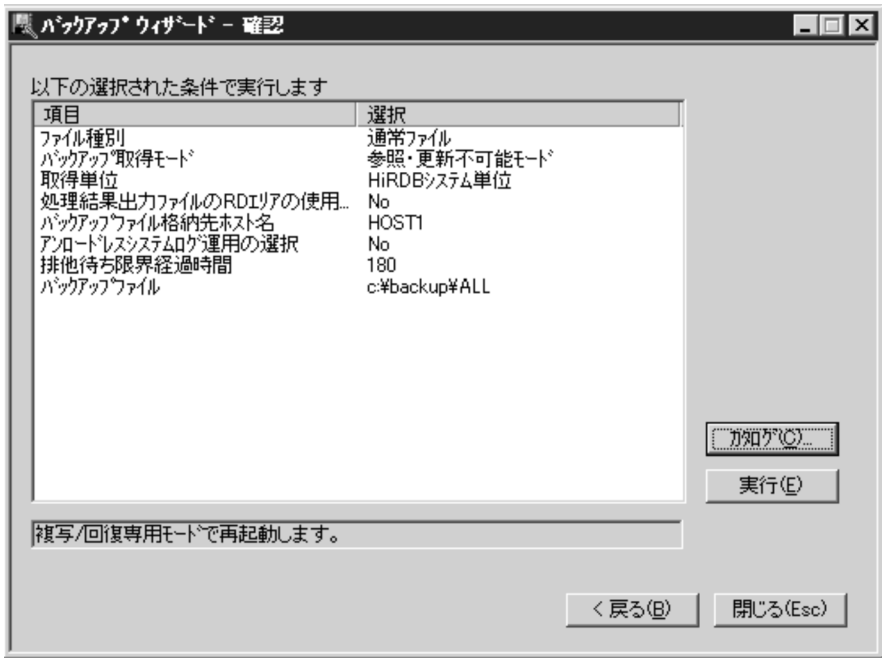


11. **[OK]** をクリックして，［バックアップウィザード - バックアップ取得先の指定］画面に戻ります。



12. **[次へ]** をクリックして，［バックアップウィザード - 確認］画面を表示します。
    設定した条件を確認します。問題があれば **[戻る]** をクリックして，該当する箇所を修正します。

    - このバックアップを繰り返し実行する場合は，**[カタログ]** をクリックしてカタログ登録をすると，次回以降バックアップを実行するときのウィザードの入力が省略できます。カタログ登録したバックアップは，日時指定でスケジュールすることもできます。詳細については，HiRDB Control Manager のヘルプを参照してください。



13. **[実行]** をクリックすると，バックアップ処理が始まります。

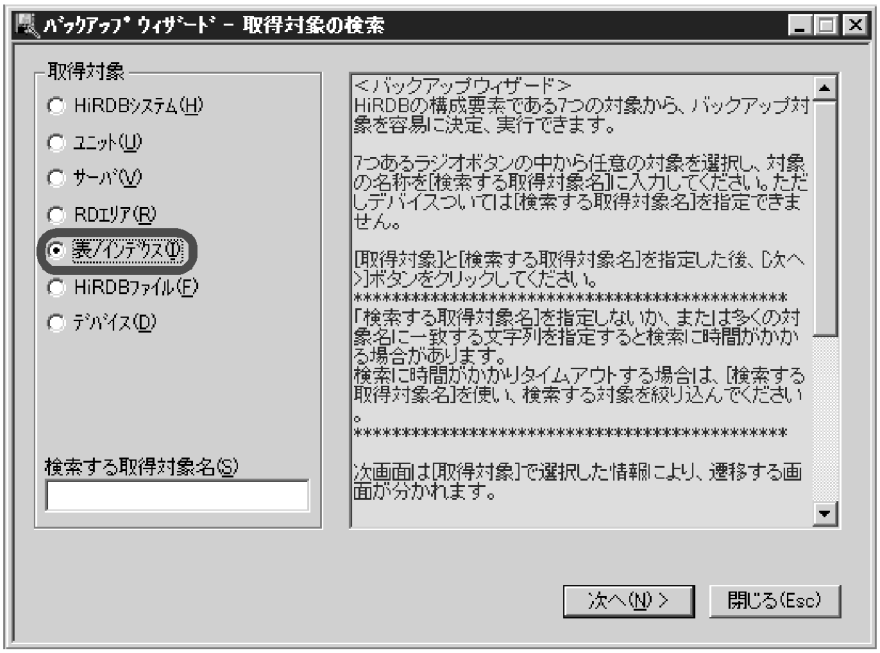
- バックアップ取得モードを変更する場合は，タブメニューの **[バックアップ]** − **[ウィザード環境設定]** で変更してください。

6.4.2　例題２（HiRDB Control Manager のウィザードで RD エリア単位にバックアップを取得する場合）

HiRDB Control Manager のバックアップウィザードで，任意の表を選択してバックアップを取得します。バックアップウィザードでバックアップを取得する表を選択すると，その表に関連する RD エリア（同時にバックアップを取得する必要がある RD エリア）がリストアップされ，それらの RD エリアも同時にバックアップを取得できます。

〈手順〉

1. HiRDB Control Manager - Console を起動します。
   起動方法については，「HiRDB Control Manager - Console の起動方法」を参照してください。
2. 操作対象の HiRDB サーバを登録します。
   既に登録されている場合は，この手順は必要ありません。登録方法については，「管理 HiRDB の登録方法」を参照してください。
3. タブメニューの［**バックアップ**］－［**ウィザード**］を選択して，［バックアップウィザード - 取得対象の検索］画面を表示します。
   取得対象として，「表/インデクス」を選択します。



4. ［**次へ**］をクリックして，［バックアップウィザード - 表/インデクスの表示］画面を表示します。
   ［表/インデクス］欄から，バックアップを取得する表を選択します。



5. **[次へ]** をクリックして，［バックアップウィザード - 関連する RD エリアの追加］画面を表示します。



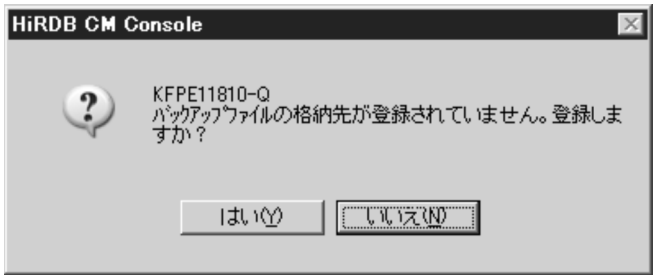
［関連する RD エリア］欄には，表に定義されているインデクスの格納 RD エリアなど，表を回復するときに同時に回復する必要がある RD エリアが表示されます。これらの RD エリアも同時にバックアップを取得する場合は，［関連する RD エリア］欄から該当する RD エリアを選択し，［追加］をクリックしてください。そうすると，［バックアップ対象 RD エリア］欄に，関連する RD エリアが追加されます。



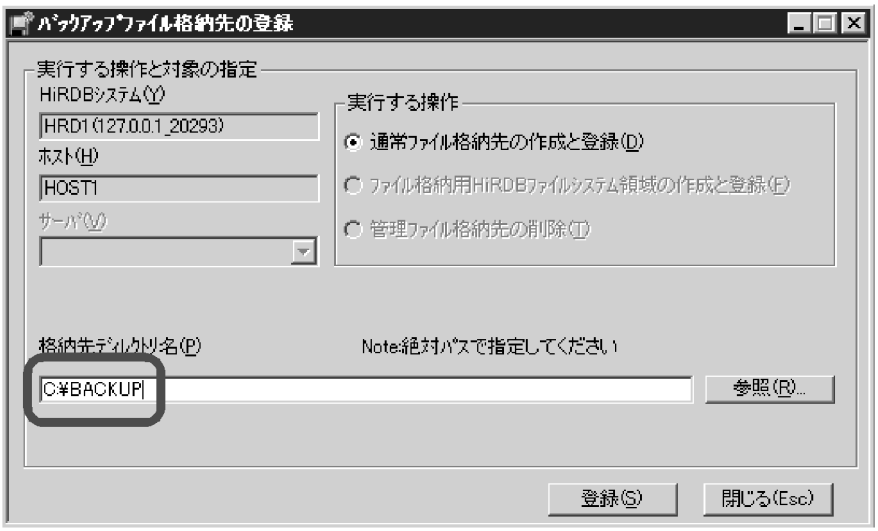
6. **[次へ]** をクリックして，［バックアップウィザード - バックアップ取得先の指定］画面を表示します。
   ここでは，次の内容を指定します。
   - バックアップファイルのファイル種別
   - 格納先のホスト名
   - バックアップファイル名
     バックアップファイル名は，**[バックアップ取得先]** をクリックして指定します。



7. バックアップファイルの格納先が登録されていない場合は，次のポップアップが表示されます。
   バックアップファイルの格納先が登録済みの場合は，この画面は表示されません。



8. **[はい]** をクリックして，［バックアップファイル格納先の登録］画面を表示します。
   ［格納先ディレクトリ名］欄に，バックアップファイルの格納先ディレクトリを指定します。
   バックアップファイルの格納先が登録済みの場合は，この画面は表示されません。



9. **[登録]** をクリックして，［バックアップファイル格納先の指定］画面を表示します。
   ［バックアップファイル名］欄に，バックアップファイル名を入力します。



10. **[追加]** をクリックします。
    ［バックアップ取得先］欄に，追加したバックアップファイル名が表示されます。



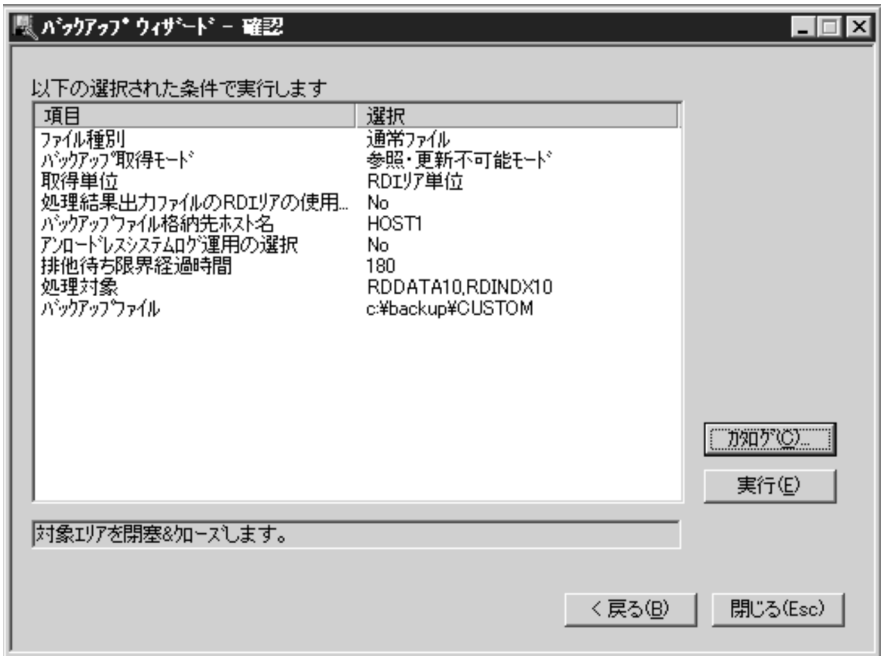
11. **[OK]** をクリックして，［バックアップウィザード - バックアップ取得先の指定］画面に戻ります。



12. **[次へ]** をクリックして，［バックアップウィザード - 確認］画面を表示します。

    設定した条件を確認します。問題があれば **[戻る]** をクリックして，該当する箇所を修正します。

    - このバックアップを繰り返し実行する場合は，**[カタログ]** をクリックしてカタログ登録をすると，次回以降バックアップを実行するときのウィザードの入力が省略できます。カタログ登録したバックアップは，日時指定でスケジュールすることもできます。詳細については，HiRDB Control Manager のヘルプを参照してください。



13. **[実行]** をクリックすると，バックアップ処理が始まります。

- バックアップ取得モードを変更する場合は，タブメニューの［**バックアップ**］－［**ウィザード環境設定**］で変更してください。

6.4.3　例題３（システム単位にバックアップを取得する場合）

HiRDB/シングルサーバの稼働中に，全 RD エリアを対象としたバックアップを取得します。

(1)　pdlogswap コマンドでシステムログファイルをスワップさせます

データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。(2)で取得するバックアップを使用して RD エリアを回復する場合は，これ以降に取得したシステムログ（これ以降，現用になったファイルのシステムログ）が入力情報になります。

```
pdlogswap -d sys -w
```

(2)　pdcopy コマンドで全 RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01
-z C:¥pdcopy¥logpoint01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。アンロードレスシステムログ運用又は自動ログアンロード機能を使用している場合に指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.4.4 例題４（システム単位にバックアップを取得する場合）

HiRDB/パラレルサーバの稼働中に，全 RD エリアを対象としたバックアップを取得します。

(1) 全バックエンドサーバ及びディクショナリサーバのシステムログファイルを pdlogswap コマンドでスワップさせます

データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。(2)で取得するバックアップを使用して RD エリアを回復する場合は，これ以降に取得したシステムログ（これ以降，現用になったファイルのシステムログ）が入力情報になります。

```
pdlogswap -d sys -s bes1 -w
pdlogswap -d sys -s bes2 -w
pdlogswap -d sys -s dic -w
```

(2) pdcopy コマンドで全 RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

備考

自動ログアンロード機能を使用している場合は，pdlogatul コマンドを実行してバックアップ取得時点の現用システムログファイルに対するアンロードログファイル名を記録しておいてください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.4.5　例題５（システム単位にバックアップを取得する場合）

HiRDB を一度終了した後に pdstart -r コマンドで HiRDB を開始し，全 RD エリアを対象としたバックアップを取得します。

> **注意事項**
>
> この方法でバックアップを取得する場合は，ログポイント情報ファイルを作成できません。したがって，アンロードレスシステムログ運用をしている場合は，この方法をとらないでください。

(1)　pdstop コマンドで HiRDB を正常終了させます

```
pdstop
```

(2)　pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(3)　pdcopy コマンドで全 RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照・更新不可能モードを指定します。pdstart -r で HiRDB を開始しているため，x（参照・更新不可能モード）を指定する場合でも，RD エリアを閉塞かつクローズ状態にする必要がありません。

-a：全 RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

備考

自動ログアンロード機能を使用している場合は，pdlogatul コマンドを実行してバックアップ取得時点の現用システムログファイルに対するアンロードログファイル名を記録しておいてください。

(4)　pdstop コマンドで HiRDB を正常終了させます

```
pdstop
```

(5) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.4.6 例題６（ユニット単位にバックアップを取得する場合）

HiRDB/パラレルサーバの稼働中に，ユニット下の RD エリアを対象としたバックアップを取得します。

(1) pdlogswap コマンドでユニット下の全システムログファイルをスワップさせます

データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。(2)で取得するバックアップを使用して RD エリアを回復する場合は，これ以降に取得したシステムログ（これ以降，現用になったファイルのシステムログ）が入力情報になります。

```
pdlogswap -d sys -s bes1 -w
pdlogswap -d sys -s bes2 -w
```

(2) pdcopy コマンドでユニット単位のバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -u UNT1 -b C:¥pdcopy¥backup01
-p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-u：ユニット（UNT1）下の全 RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

備考

自動ログアンロード機能を使用している場合は，pdlogatul コマンドを実行してバックアップ取得時点の現用システムログファイルに対するアンロードログファイル名を記録しておいてください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.4.7　例題７（サーバ単位にバックアップを取得する場合）

HiRDB/パラレルサーバの稼働中に，バックエンドサーバ（bes1）下の RD エリアを対象としたバックアップを取得します。アンロードレスシステムログ運用をしているため，同時にログポイント情報ファイルも取得します。

(1)　バックアップ取得対象サーバのシステムログファイルを pdlogswap コマンドでスワップさせます

データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。(2)で取得するバックアップを使用して RD エリアを回復する場合は，これ以降に取得したシステムログ（これ以降，現用になったファイルのシステムログ）が入力情報になります。

```
pdlogswap -d sys -s bes1 -w
```

(2)　pdcopy コマンドでサーバ単位のバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -s bes1 -b C:¥pdcopy¥backup01
-z C:¥pdcopy¥logpoint01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-s：バックエンドサーバ（bes1）下の RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。アンロードレスシステムログ運用及び自動ログアンロード機能を使用している場合に指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.4.8　例題８（RD エリア単位にバックアップを取得する場合）

HiRDB の稼働中に，RD エリア（rdarea01，rdarea02）のバックアップを取得します。なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時にバックアップを取得する必要があります。バックアップ対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1) バックアップ対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

バックアップ対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2) バックアップ取得対象の RD エリアを pdhold -c コマンドで閉塞かつクローズ状態にします

この操作は，バックアップ取得モードに x（参照・更新不可能モード）を指定する場合に必要です。バックアップ取得モードに r（参照可能モード）又は s（更新可能モード）を指定する場合は，この操作は必要ありません。

```
pdhold -r rdarea01,rdarea02 -c
```

(3) バックアップ取得対象 RD エリアがあるサーバのシステムログファイルを pdlogswap コマンドでスワップさせます

データベース回復時に必要なシステムログを物理的に分けるために，システムログファイルをスワップさせます。(4)で取得するバックアップを使用して RD エリアを回復する場合は，これ以降に取得したシステムログ（これ以降，現用になったファイルのシステムログ）が入力情報になります。

```
pdlogswap -d sys -s bes1 -w
```

(4) pdcopy コマンドで RD エリア単位のバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -r rdarea01,rdarea02
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照・更新不可能モードを指定します。

-r：RD エリア（rdarea01，rdarea02）のバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

備考

自動ログアンロード機能を使用している場合は，pdlogatul コマンドを実行してバックアップ取得時点の現用システムログファイルに対するアンロードログファイル名を記録しておいてください。

(5)　pdrels -o コマンドで RD エリアの閉塞を解除してオープンします

この操作は，バックアップ取得モードにｘ（参照・更新不可能モード）を指定する場合に必要です。バックアップ取得モードにｒ（参照可能モード）又はｓ（更新可能モード）を指定する場合は，この操作は必要ありません。

```
pdrels -r rdarea01,rdarea02 -o
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.5　差分バックアップの取得

実行者　HiRDB 管理者

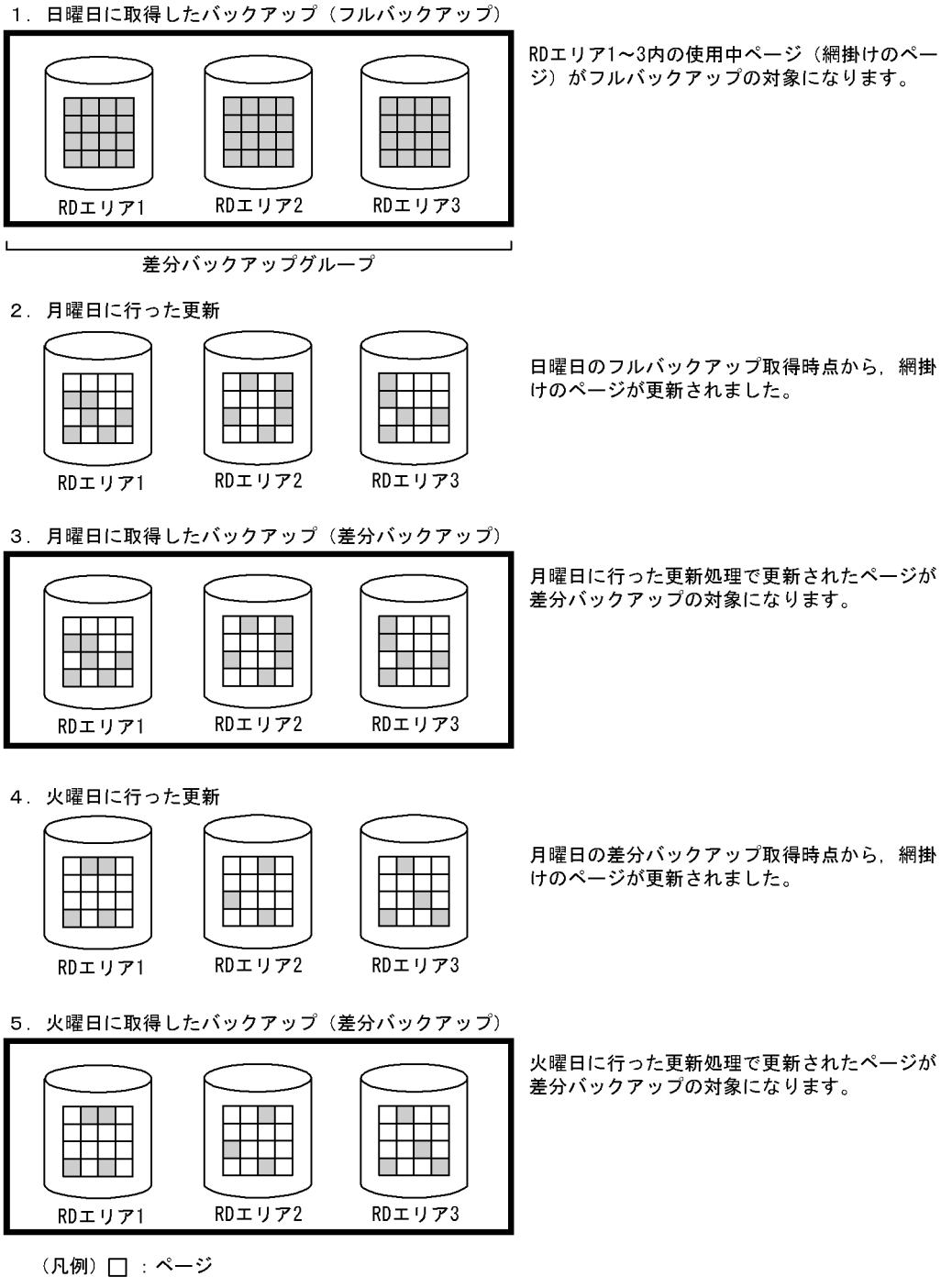
ここでは，差分バックアップ機能を使用したバックアップの取得方法について説明します。ここで説明する項目は次のとおりです。

- 差分バックアップ機能とは
- 差分バックアップ機能を使用するための準備
- 差分バックアップ機能の運用例
- 累積差分バックアップの作成
- 差分バックアップの履歴情報ファイルの参照
- 差分バックアップ管理ファイルの回復

6.5.1　差分バックアップ機能とは

通常，バックアップは RD エリア単位に取得するため，更新したページも更新しないページもバックアップの取得対象になります。差分バックアップ機能を使用すると，前回のバックアップ取得時点から現在までに更新したページだけがバックアップの取得対象になります。このように，前回のバックアップ取得時点からの差分だけをバックアップとして取得するため，バックアップの取得処理時間を短縮できます。データベースの容量が多くてデータ更新量が少ない場合に，差分バックアップ機能の使用を検討してください。差分バックアップ機能の概要を次の図に示します。

図 6–1　差分バックアップ機能の概要



〔説明〕

1. 日曜日に RD エリア 1～3 のバックアップを取得します。このとき，RD エリア 1～3 内の使用中ページがバックアップの対象になります。このバックアップを**フルバックアップ**といい，グループ化した RD エリア群を**差分バックアップグループ**といいます。
2. 月曜日の業務で更新処理を行います。
3. 月曜日の業務終了後に RD エリア 1～3 のバックアップを取得します。このとき，RD エリア 1～3 内の更新ページがバックアップの対象になります。このバックアップを**差分バックアップ**といいます。

4. 火曜日の業務で更新処理を行います。
5. 火曜日の業務終了後に RD エリア 1〜3 のバックアップを取得します。このとき，RD エリア 1〜3 内の更新ページがバックアップの対象になります。

データベースの回復方法

差分バックアップ機能使用時のデータベースの回復方法については，「差分バックアップ機能使用時のデータベースの回復方法」を参照してください。

参考

アンロードレスシステムログ運用の場合でも，差分バックアップ機能を使用できます。

注意事項

- 同じ差分バックアップグループのバックアップファイル（フルバックアップファイル，差分バックアップファイル，及び累積差分バックアップファイル）は，同じサーバマシンに格納してください。累積差分バックアップファイルについては，「累積差分バックアップの作成」を参照してください。
- LOB 用 RD エリアに対しては差分バックアップを取得できません。毎回，全使用中のページがバックアップの対象になります。
- RD エリアの構成変更をした場合は，RD エリアのフルバックアップを取得し直す必要があります。RD エリアの構成変更後に RD エリアを回復する場合は，RD エリアの構成変更前のバックアップを使用できません。
- テープ装置にバックアップを取得する場合は，NetBackup 連携機能を使用してバックアップを取得してください。NetBackup 連携機能を使用しないとテープ装置にバックアップを取得できません。

6.5.2 差分バックアップ機能を使用するための準備

(1) 差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域の作成

バックアップを取得したときに**差分バックアップ管理ファイル**が出力されます。この差分バックアップファイルには差分バックアップ取得時の情報が格納されていて，バックアップの取得時及びバックアップを使用したデータベースの回復時に HiRDB が使用します。

この差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域を **pdfmkfs** コマンドで作成してください。HiRDB/パラレルサーバの場合，システムマネジャを定義したユニットに HiRDB ファイルシステム領域を作成してください。

```
pdfmkfs -n 10 -l 4096 -e 60000 -k UTL C:¥pdcopy¥admfile
```

〔説明〕

-n：HiRDB ファイルシステム領域長をメガバイト単位で指定します。

-l：最大ファイル数を指定します。

-e：増分回数を指定します。

-k：HiRDB ファイルシステム領域の使用目的（ユティリティ用）を指定します。

C:¥pdcopy¥admfile：

HiRDB ファイルシステム領域名を指定します。この HiRDB ファイルシステム領域内に差分バックアップ管理ファイルが作成されます。差分バックアップ管理ファイルの名称は，差分バックアップグループ名と同じになります。

注意事項

- 差分バックアップ管理ファイルは，差分バックアップ取得時の情報が格納されている重要なファイルです。このファイルがないと差分バックアップ機能が使用できなくなり，データベースを回復できなくなります。したがって，この HiRDB ファイルシステム領域及びこのファイルを誤って削除しないようにしてください。バックアップの取得後，pdfbkup コマンドで差分バックアップ管理ファイルのバックアップを取得してください。

参考

同じ HiRDB ファイルシステム領域内にバックアップファイルと差分バックアップ管理ファイルを格納できます。

(2)　差分バックアップグループ化する RD エリアについて

差分バックアップ機能を適用する RD エリア群（RD エリアのグループ）を**差分バックアップグループ**といいます。ここでは，どのように RD エリアをグループ化するかを説明します。

(a)　最新の差分バックアップ取得時点に回復する場合，又はアンロードログファイル（システムログファイル）を使用した回復をする場合

この場合，差分バックアップグループ内の特定の RD エリアを回復できます。データベースのデータ量が多い場合に全 RD エリアのバックアップを一つのグループにすると，特定の RD エリアを回復するときに時間が掛かります。これは，バックアップファイルをシーケンシャルに読み込むための時間を必要とするからです。

このため，適当な数の RD エリアをグループ化してバックアップを取得することをお勧めします。例えば，ディスク障害になった場合はディスク単位に RD エリアを回復するため，同じディスクに格納されている RD エリアを一つのグループにします。

(b)　最新でない差分バックアップ取得時点に回復する場合（バックアップファイルだけを使用して回復する場合）

この場合，差分バックアップグループ内の特定の RD エリアを回復できません。回復する場合はグループ内の全 RD エリアが回復の対象になります。

このため，関連のある RD エリアをグループ化することをお勧めします。例えば，同一表のデータを格納している RD エリアとその表のインデクスを格納している RD エリアをグループ化してください。そうすれば，RD エリアを回復する場合に pdrstr コマンドを 1 回実行すれば，表格納 RD エリア及びインデクス格納 RD エリアを同時に回復できます。

注意事項

バックアップファイルだけを使用して，最新でない差分バックアップ取得時点に RD エリアを回復した場合，差分バックアップ管理ファイル中の回復時点以降の差分バックアップ情報が無効になります。このため，最新の差分バックアップ取得時点に RD エリアを回復できなくなります。なお，回復中にエラーが発生した場合は，差分バックアップ情報を無効にしません。

6.5.3　差分バックアップ機能の運用例

差分バックアップ機能の運用の流れを次に示します。

〈手順〉

1. 差分バックアップ機能を適用する RD エリアを選びます（差分バックアップグループを決めます）。
2. バックアップ対象 RD エリアのフルバックアップを取得します。
3. バックアップ対象 RD エリアの差分バックアップを取得します。例えば，1 日に 1 回取得してください。
4. バックアップ対象 RD エリアのフルバックアップを定期的に取得します。例えば，1 週間に 1 回取得してください。

次に示す条件下での差分バックアップ機能の運用例を説明します。

〈条件〉

- ユーザ用 RD エリア（rdarea01，rdarea02）に対して差分バックアップ機能を適用します。
- 1 週間に一度（日曜日），フルバックアップを取得します。
- 毎日（月曜日～土曜日）の更新情報を差分バックアップとして取得します。

(1)　フルバックアップの取得（日曜日に行うこと）

pdlogswap コマンドでシステムログファイルをスワップしてから，pdcopy コマンドでフルバックアップを取得します。

```
pdlogswap -d sys -w

pdcopy -m C:¥rdarea¥mast¥mast01 -M r -r rdarea01,rdarea02 -g backupg1(S)
-b C:¥pdcopy¥backup01 -d a -K C:¥pdcopy¥admfile -L 5 -o C:¥pdcopy¥rfile
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-r：バックアップ取得対象 RD エリアを指定します。

ここで指定した RD エリア群が差分バックアップグループになります。バックアップ取得対象 RD エリアは途中で変更できません。

-g：差分バックアップグループ名を指定します。

最初のフルバックアップ取得時には，必ず差分バックアップグループ名に(S)を指定してください。次回以降の差分バックアップ取得時に，ここで指定した差分バックアップグループ名を指定します。

-b：バックアップファイル名（フルバックアップファイル名）を指定します。

-d：バックアップ種別を指定します。

a：フルバックアップを取得します。

b：最新のフルバックアップからの累積差分バックアップを取得します。

c：前回取得した累積差分バックアップ，又は前回取得したフルバックアップのどちらか最新のものから，累積差分バックアップを取得します。

d：差分バックアップを取得します。

累積差分バックアップについては，「累積差分バックアップの作成」を参照してください。

-K：差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域名を指定します。

-L：差分バックアップ管理ファイルの容量をメガバイト単位で指定します。

-o：差分バックアップの履歴情報ファイル名を指定します。

差分バックアップの履歴情報ファイルについては，「差分バックアップの履歴情報ファイルの参照」を参照してください。

(2)　差分バックアップ管理ファイルのバックアップの取得（日曜日に行うこと）

作成された差分バックアップ管理ファイルのバックアップを pdfbkup コマンドで取得してください。

```
pdfbkup C:¥pdcopy¥admfile¥backupg1 D:¥backup¥pdcopy¥admfile¥backupg1
```

〔説明〕

C:¥pdcopy¥admfile¥backupg1：

差分バックアップ管理ファイル名を指定します。差分バックアップ管理ファイル名は差分バックアップグループ名と同じになります。

D:¥backup¥pdcopy¥admfile¥backupg1：

差分バックアップ管理ファイルのバックアップファイル名を指定します。

(3) 差分バックアップの取得（月曜日～土曜日に行うこと）

pdlogswap コマンドでシステムログファイルをスワップしてから，pdcopy コマンドで差分バックアップを取得します。

```
pdlogswap -d sys -w

pdcopy -m C:¥rdarea¥mast¥mast01 -M r -g backupg1 -b C:¥pdcopy¥backup02
-d d -K C:¥pdcopy¥admfile -o C:¥pdcopy¥rfile
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-g：差分バックアップグループ名を指定します。(S)の指定は必要ありません。

-b：バックアップファイル名（差分バックアップファイル名）を指定します。

-d：バックアップ種別を指定します。

a：フルバックアップを取得します。

b：最新のフルバックアップからの累積差分バックアップを取得します。

c：前回取得した累積差分バックアップ，又は前回取得したフルバックアップのどちらか最新のものから，累積差分バックアップを取得します。

d：差分バックアップを取得します。

累積差分バックアップについては，「累積差分バックアップの作成」を参照してください。

-K：差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域名を指定します。

-o：差分バックアップの履歴情報ファイル名を指定します。

差分バックアップの履歴情報ファイルについては，「差分バックアップの履歴情報ファイルの参照」を参照してください。

(4) 差分バックアップ管理ファイルのバックアップの取得（月曜日～土曜日に行うこと）

作成された差分バックアップ管理ファイルのバックアップを pdfbkup コマンドで取得してください。

```
pdfbkup -r C:¥pdcopy¥admfile¥backupg1 D:¥backup¥pdcopy¥admfile¥backupg1
```

〔説明〕

-r：差分バックアップ管理ファイルのバックアップを上書きする指定をします。

C:¥pdcopy¥admfile¥backupg1：

差分バックアップ管理ファイル名を指定します。差分バックアップ管理ファイル名は差分バックアップグループ名と同じになります。

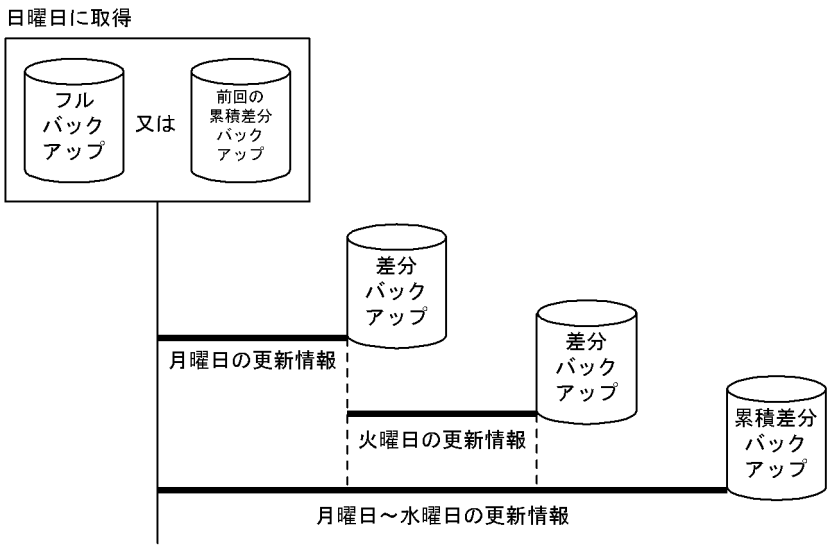
D:¥backup¥pdcopy¥admfile¥backupg1：

差分バックアップ管理ファイルのバックアップファイル名を指定します。

6.5.4　累積差分バックアップの作成

pdcopy コマンド（-d オプションに c を指定）を実行して**累積差分バックアップ**を作成できます。累積差分バックアップを作成すると，回復時に使用するバックアップファイルの容量を削減できることがあります。例えば，毎日同一ページを更新している場合，同一ページの更新情報をマージするため，その分，回復時に使用するバックアップファイルの容量を削減できます。累積差分バックアップの概念を次の図に示します。

図 6–2　累積差分バックアップの概念



〔説明〕

- 水曜日に累積差分バックアップを取得すると，データベースの回復時には月曜日と火曜日に取得した差分バックアップを使用しません。データベースの回復時に必要なバックアップは，フルバックアップと水曜日に取得した累積差分バックアップになります。
- フルバックアップからの累積差分バックアップを取得するか，又は前回取得した累積差分バックアップからの累積差分バックアップを取得するかは，pdcopy コマンドの-d オプションで指定します。

(1)　注意事項

累積差分バックアップの作成後，pdfbkup コマンドで差分バックアップ管理ファイルのバックアップを取得してください。

(2)　累積差分バックアップの運用例

累積差分バックアップを使用すると，例えば，次に示すような運用ができます。

- 日曜日にフルバックアップを取得します。
- 月曜日～火曜日に差分バックアップを取得します。
- 水曜日に，月曜日～水曜日の分の**累積差分バックアップ**を取得します。※
- 木曜日～金曜日に差分バックアップを取得します。
- 土曜日に，木曜日～土曜日の分の**累積差分バックアップ**を取得します。※

注※

このときの pdcopy コマンドの指定例を次に示します。なお，pdcopy コマンドを実行する前に pdlogswap コマンドでシステムログファイルをスワップしてください。

```
pdlogswap -d sys -w
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -g backupg1 -b C:¥pdcopy¥backup03
-d c -K C:¥pdcopy¥admfile -o C:¥pdcopy¥rfile
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-g：差分バックアップグループ名を指定します。(S)の指定は必要ありません。

-b：バックアップファイル名（累積差分バックアップファイル名）を指定します。

-d：バックアップ種別を指定します。

a：フルバックアップを取得します。

b：最新のフルバックアップからの累積差分バックアップを取得します。

c：前回取得した累積差分バックアップ，又は前回取得したフルバックアップのどちらか最新のものから，累積差分バックアップを取得します。

d：差分バックアップを取得します。

-K：差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域名を指定します。

-o：差分バックアップの履歴情報ファイル名を指定します。

6.5.5　差分バックアップの履歴情報ファイルの参照

差分バックアップの履歴情報ファイルには，差分バックアップの取得情報が時系列に格納されています。pdcopy コマンドで差分バックアップの履歴情報ファイルを参照できます。履歴情報ファイルはメモ帳などで参照してください。差分バックアップの履歴情報ファイルの出力例を次に示します。

```
pdcopy (06-00) ***** DB COPY *****  2000-10-26 21:30:20[1] utl3[2]

/hirdb/pdcopy/admfile[3]
backupg1[4]
2000-10-19 21:00:20[5]
2000-10-26 21:18:20[6]
RDAREA NAME :rdarea01,rdarea02[7]
a[8] 2000-10-19 21:08:20[9],2000-10-19 21:09:25[10] r[11]u[12] host01:/hirdb/backup01[13]
d[8] 2000-10-20 21:01:20[9],2000-10-20 21:01:38[10] r[11]u[12] host01:/hirdb/backup02[13]
c[8] 2000-10-21 21:05:20[9],2000-10-21 21:06:10[10] r[11]u[12] host01:/hirdb/backup03[13]
b[8] 2000-10-22 21:01:20[9],2000-10-22 21:01:45[10] r[11]u[12] host01:/hirdb/backup04[13]
```

〔説明〕

1. pdcopy コマンドの実行日時
2. HiRDB 識別子
3. 差分バックアップ管理ファイルを格納している HiRDB ファイルシステム領域名
4. バックアップグループ名
5. バックアップグループの作成日時
6. バックアップグループの最終更新日時
7. バックアップグループに属する RD エリア名
8. バックアップ種別

   a：フルバックアップ

   b：最新のフルバックアップからの累積差分バックアップ

   c：前回取得した累積差分バックアップ又は前回取得したフルバックアップのどちらか最新のものからの累積差分バックアップ

   d：差分バックアップ
9. バックアップ取得開始日時
10. バックアップ取得終了日時
11. バックアップ取得モード

    x：参照・更新不可能モード

    r：参照可能モード

    s：更新可能モード
12. バックアップファイルの種別

    u：Windows のファイル

    i：バックアップファイル用の HiRDB ファイルシステム領域
13. バックアップファイル名

    ホスト名：バックアップファイル名の形式で表示されます。

6.5.6　差分バックアップ管理ファイルの回復

差分バックアップ管理ファイルに障害が発生すると，差分バックアップ機能が使用できなくなります。したがって，差分バックアップ管理ファイルに障害が発生した場合は，次に示すどちらかの処置をしてください。

- 差分バックアップ管理ファイルのバックアップから差分バックアップ管理ファイルを回復してください。
- 差分バックアップグループ名に(S)を指定して，フルバックアップを取得し直してください。

6.6　バックアップ閉塞を使用してバックアップを取得する場合（pdcopy コマンド以外でバックアップを取得する場合）

ここでは，主に pdcopy コマンド以外でバックアップを取得する方法（ほかの製品の機能でバックアップを取得する方法）について説明します。pdcopy コマンド以外でバックアップを取得する場合は，RD エリアを**バックアップ閉塞**する必要があります。

pdcopy コマンド以外でバックアップを取得したときのデータベースの回復方法については，「pdcopy コマンド以外でバックアップを取得した場合の回復方法」を参照してください。

6.6.1　バックアップ閉塞とは

次に示す場合に，バックアップ対象 RD エリアをバックアップ閉塞してください。

- pdcopy コマンド以外（ほかの製品の機能）でバックアップを取得する場合
- pdcopy コマンドを更新可能モードで実行する場合

RD エリアをバックアップ閉塞すると，オンライン中でもバックアップを取得できます。RD エリアをバックアップ閉塞するには，**pdhold** コマンドで**-b** オプションを指定します。

(1)　バックアップ閉塞の種類

バックアップ閉塞には次の表に示す四つの種類があります。

表 6–5　バックアップ閉塞の種類

バックアップ閉塞の種類	説明
参照可能バックアップ閉塞	バックアップ閉塞中，バックアップ閉塞 RD エリアの参照はできるが，更新は SQL エラー（–920）になります。 **特徴及び注意事項** • ここで取得したバックアップを使用して，データベースをバックアップ取得時点に回復できます。 • 更新トランザクションとのデッドロックは発生しません。 • 参照可能バックアップ閉塞中の RD エリアに対してログレスモードの更新トランザクションを実行すると，更新トランザクションは RD エリアで排他待ち状態になり，排他解除後にエラーになるため，RD エリアが障害閉塞状態になります。
参照可能バックアップ閉塞（更新 WAIT モード）	バックアップ閉塞中，バックアップ閉塞 RD エリアを参照できます。更新はバックアップ閉塞が解除されるまで排他待ち状態になります。 **特徴及び注意事項** • ここで取得したバックアップを使用して，データベースをバックアップ取得時点に回復できます。 • pd_lck_wait_timeout オペランド又はクライアント環境定義の PDCWAITTIME オペランドには，バックアップ閉塞中の時間以上を指定してください。更新トランザクションがタイムアウトになると，SQL エラー（–770）になります。

バックアップ閉塞の種類	説明
	• 更新トランザクションとの間でデッドロックが発生する場合があります。更新トランザクションをエラーにしないようにするには，pd_deadlock_priority_use = Y を指定してください。デッドロックが発生した場合は，pdhold コマンドが取得したバックアップ閉塞の排他をすべて解除し，再度同じバックアップ閉塞処理が繰り返されます。このリトライ処理は，最大 5 回繰り返されます。5 回目のリトライ処理でもデッドロックが発生する場合は，pdhold コマンドが取得したバックアップ閉塞の排他をすべて解除して，pdhold コマンドをエラーにします。 • ログレスモードの更新トランザクションがタイムアウト又はデッドロックでエラーになった場合，更新した RD エリアが障害閉塞状態になります。
更新可能バックアップ閉塞	バックアップ閉塞中，バックアップ取得対象 RD エリアの参照及び更新ができます。更新トランザクションが実行中でも，pdhold コマンドを待ち状態にしないで，RD エリアをすぐに更新可能バックアップ閉塞状態にします。 **特徴及び注意事項** • データベースをバックアップ取得時点には回復できません。バックアップ取得時点以降の任意の同期点への回復だけとなります。したがって，データベースを回復するには，バックアップ及びバックアップ取得直前のシンクポイントからのシステムログが必要になります。 • 前回のシンクポイント時点から現時点の間に，更新前ログ取得モード又はログレスモードの更新トランザクションを実行した場合は，更新可能バックアップ閉塞を適用しないでください。このバックアップを使用してデータベースを回復できません。
更新可能バックアップ閉塞（WAIT モード）	バックアップ閉塞中，バックアップ取得対象 RD エリアの参照及び更新ができます。更新トランザクションが実行中の場合は，更新トランザクションが終了するまで pdhold コマンドを待ち状態にします。 **特徴及び注意事項** • バックアップ閉塞中に発生した更新トランザクションによって更新されたバッファの内容が RD エリアに反映された場合は，バックアップ閉塞を解除したときに **KFPH00157-W** メッセージを出力します。このメッセージが出力された場合は，データベースをバックアップ取得時点に回復できません。バックアップ取得時点以降の任意の同期点への回復だけとなります。したがって，データベースを回復するには，バックアップ閉塞コマンド実行開始時点からのシステムログが必要になります。 • **KFPH00157-W** メッセージが出力されなかった場合は，ここで取得したバックアップを使用して，データベースをバックアップ取得時点に回復できます。 ただし，pdcopy コマンドの更新可能モード（-M s 指定）でバックアップを取得した場合は，データベースをバックアップ取得時点に回復できません。バックアップ取得時点以降の任意の同期点への回復だけになります。したがって，データベースを回復するには，バックアップ及びバックアップ取得直前のシンクポイントからのシステムログが必要になります。 • 更新前ログ取得モード又はログレスモードの更新トランザクションは，バックアップ閉塞 RD エリアで排他待ちになります。したがって，pd_lck_wait_timeout オペランド又はクライアント環境定義の PDCWAITTIME オペランドには，バックアップ閉塞中の時間以上を指定してください。更新トランザクションがタイムアウトになった場合，更新トランザクションがエラーになります。 • 更新前ログ取得モード又はログレスモードの更新トランザクションとの間でデッドロックが発生する場合があります。更新トランザクションをエラーにしないようにするには，pd_deadlock_priority_use = Y を指定してください。デッドロックが発生した場合は，pdhold コマンドが取得したバックアップ閉塞の排他をすべて解除し，再度同じバックアップ閉塞処理が繰り返されます。このリトライ処理は，最大 5 回繰り返されます。5 回目のリトライ処理でもデッドロックが発生する場合は，pdhold コマンドが取得したバックアップ閉塞の排他をすべて解除して，pdhold コマンドをエラーにします。 • 更新前ログ取得モード又はログレスモードの更新トランザクションがタイムアウト又はデッドロックでエラーになった場合，更新した RD エリアが障害閉塞状態になります。

参考

参照可能バックアップ閉塞，及び参照可能バックアップ閉塞（更新 WAIT モード）は**データベースの静止化**ともいいます。

(2)　前提条件（RD エリアの構成について）

バックアップ取得対象 RD エリアの全構成ファイルが一回のバックアップで取得できるように，RD エリアの構成ファイルをディスクに配置してください。

(3)　注意事項

1. HiRDB が使用しているディスクのバックアップを取得している場合，HiRDB が異常終了又は強制終了するとそのバックアップは無効になります。この場合，再度バックアップを取得してください。
2. RD エリアの状態が，更新可能バックアップ閉塞又は更新可能バックアップ閉塞（WAIT モード）の場合，RD エリアの自動増分が抑止されます。そのため，これらの閉塞状態中に新規ページの確保が発生する大量データの追加又は更新業務の実行は避けてください。RD エリアの自動増分の抑止を解除するには，pdrels コマンドで RD エリアの閉塞を解除してください。
3. pdcopy コマンド以外（ほかの製品の機能）でバックアップを取得する場合，バックアップ取得中にデータベース構成変更ユティリティ（pdmod コマンド）を実行しないでください。
4. バックアップ閉塞中に HiRDB が異常終了又は強制終了した場合，バックアップ閉塞の種類によっては HiRDB の再開始時にバックアップ閉塞を引き継ぎません。この場合，再度バックアップ閉塞を実施してバックアップを取得してください。HiRDB 再開始時のバックアップ閉塞の引き継ぎ状態を次の表に示します。

表 6–6　HiRDB 再開始時のバックアップ閉塞の引き継ぎ状態

バックアップ閉塞の種類	再開始時に状態を引き継ぐかどうか
参照可能バックアップ閉塞	○
参照可能バックアップ閉塞（更新 WAIT モード）	×
更新可能バックアップ閉塞	×
更新可能バックアップ閉塞（WAIT モード）	×

（凡例）

○：状態を引き継ぎます。

×：状態を引き継ぎません。

(4) 排他との関係

(a) バックアップ閉塞中の排他待ちタイムアウト

次に示すモードでバックアップ閉塞をする場合，排他待ちタイムアウトに注意してください。

- 参照可能バックアップ閉塞（更新 WAIT モード）
- 更新可能バックアップ閉塞
- 更新可能バックアップ閉塞（WAIT モード）

参照可能バックアップ閉塞（更新 WAIT モード）ではバックアップ閉塞を解除するまで，更新トランザクションがバックアップ閉塞 RD エリア（資源種別 0602）で排他待ちになります。更新可能バックアップ閉塞及び更新可能バックアップ閉塞（WAIT モード）ではバックアップ閉塞を解除するまで，ログレスモード又は更新前ログ取得モードの更新トランザクションがバックアップ閉塞 RD エリア（資源種別 0602）で排他待ちになります。

このため，pd_lck_wait_timeout オペランド又はクライアント環境定義の PDCWAITTIME オペランドの指定値がバックアップ閉塞から閉塞解除までの時間より短いと更新トランザクションがエラーになります。エラーになる場合はこれらのオペランドの指定値を変更してください。

これらのモードでバックアップを取得する場合は，閉塞から閉塞解除までの時間が短かい（論理ボリュームを分割する間だけでよい）ミラーディスクの機能を利用してバックアップを取得するときにお勧めします。

(b) バックアップ閉塞時のデッドロック

複数の RD エリアを次に示すモードでバックアップ閉塞する場合，デッドロックに注意してください。

- 参照可能バックアップ閉塞（更新 WAIT モード）
- 更新可能バックアップ閉塞
- 更新可能バックアップ閉塞（WAIT モード）

参照可能バックアップ閉塞（更新 WAIT モード）では，更新トランザクションとの間でデッドロックになることがあります。更新可能バックアップ閉塞及び更新可能バックアップ閉塞（WAIT モード）では，ログレスモード又は更新前ログ取得モードの更新トランザクションとの間でデッドロックになることがあります。更新トランザクションをエラーにしたくない場合は，pd_deadlock_priority_use = Y を指定してください。この場合，デッドロックが発生すると pdhold コマンドがエラーになるため，一定時間後に pdhold コマンドを再度実行してください。なお，pdhold コマンドに複数の RD エリアを指定した場合，全 RD エリアのバックアップ閉塞処理を無効にします。

また，複数の RD エリアを複数回の pdhold コマンドでバックアップ閉塞すると，バックアップ閉塞 RD エリアでデッドロックになることがあります。この場合，バックアップ閉塞した RD エリアを閉塞解除した後に pdhold コマンドを再度実行してください。このため，複数の RD エリアをバックアップ閉塞する場合は，1 回の pdhold コマンドで複数の RD エリアを指定してください（成功するまで繰り返し実行してください）。

HiRDB/パラレルサーバの場合，異なるサーバにある RD エリアを指定すると，グローバルデッドロックになることがあります。HiRDB ではグローバルデッドロックを検知できないため，トランザクションがタイムアウトエラーになります。グローバルデッドロックを避けるには，複数の RD エリアを指定するとき，サーバ単位に pdhold コマンドを実行してください。

(5)　更新可能バックアップ閉塞中に出力されるシステムログ量

更新可能バックアップ閉塞中にデータベースを更新すると，追加のシステムログが出力されます。追加で出力されるシステムログ量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「更新可能バックアップ閉塞中に出力されるシステムログ量」を参照してください。ただし，バックアップ閉塞中に HiRDB が異常終了又は強制終了した場合，HiRDB の再開始時にバックアップ閉塞を引き継ぎません（参照可能バックアップ閉塞を除く）。このため，システムログが追加で出力されることはありません。

(6)　データベースの回復に必要なシステムログの求め方

データベースを最新の同期点に回復する場合（範囲指定の回復も含む），バックアップ取得後のシステムログが必要になります。データベースの回復に必要なシステムログの求め方を次の表に示します。

表 6–7　データベースの回復に必要なシステムログの求め方（バックアップ閉塞使用時）

条件	データベースの回復に必要なシステムログの求め方
• 参照可能バックアップ閉塞の場合 • 参照可能バックアップ閉塞（更新 WAIT モード）の場合	バックアップ取得以降のシステムログが必要になります。バックアップの取得前に pdlogls コマンドを実行して，現用のシステムログについて次に示す情報を記録しておいてください。 • ログサーバプロセスのラン ID（Run ID） • システムログファイルの世代番号（Gen No.） • システムログファイルのファイルグループ名称（Group） また，バックアップの取得日時を記録しておいてください。pdrstr コマンドに-T オプションを指定する場合はこの時刻を指定します。
更新可能バックアップ閉塞（WAIT モード）の場合	バックアップ閉塞コマンド実行開始時点からのシステムログが必要になります。バックアップ閉塞の実行前に pdlogls コマンドを実行して，次に示す情報を記録しておいてください。 • ログサーバプロセスのラン ID（Run ID） • システムログファイルの世代番号（Gen No.） • システムログファイルのファイルグループ名称（Group） また，バックアップ閉塞実行開始日時を記録しておいてください。pdrstr コマンドに-T オプションを指定する場合はこの時刻を指定します。
更新可能バックアップ閉塞の場合	データベースの回復に必要なシステムログの開始位置を表示する KFPS02183-I メッセージを参照してください。※ 更新可能バックアップ閉塞前の KFPS02183-I メッセージに表示されたシステムログファイルのファイルグループ名，世代番号，シンクポイントダンプ取得開始時刻がデータベースの回復に必要なシステムログの開始位置を示しています。ログサーバプロセスのラン ID は pdlogls コマンドで確認してください。

条件	データベースの回復に必要なシステムログの求め方
	pdrstr コマンドに-T オプションを指定する場合は，シンクポイントダンプ取得開始時刻を指定します。

注※　pd_spd_assurance_msg=N を指定すると，KFPS02183-I メッセージは出力されません。

6.6.2　例題（ほかの製品のバックアップ機能を使用する場合）

HiRDB の稼働中に RD エリア（rdarea01，rdarea02）のバックアップを取得します。ほかの製品のバックアップ機能を使用して，RD エリア（rdarea01，rdarea02）を構成する HiRDB ファイルシステム領域のバックアップを取得します。

(1)　pdhold コマンドでバックアップ取得対象 RD エリアをバックアップ閉塞します

```
pdhold -r rdarea01,rdarea02 -b -w
```

〔説明〕

-r：バックアップ取得対象 RD エリア名を指定します。

-b -w：参照可能バックアップ閉塞（更新 WAIT モード）を指定します。

(2)　システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

(3)　バックアップを取得します

ほかの製品のバックアップ機能を使用して，RD エリア（rdarea01，rdarea02）を構成する HiRDB ファイルシステム領域のバックアップを取得します。

備考

自動ログアンロード機能を使用している場合は，pdlogatul コマンドを実行してバックアップ取得時点の現用システムログファイルに対するアンロードログファイル名を記録しておいてください。

(4)　pdrels コマンドで RD エリアのバックアップ閉塞を解除します

```
pdrels -r rdarea01,rdarea02
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

6.7　更新凍結コマンド（pddbfrz コマンド）を使用してバックアップを取得する場合

ユーザ LOB 用 RD エリアのバックアップ取得時間の短縮を支援する機能に更新凍結コマンド（**pddbfrz** コマンド）があります。ここでは，更新凍結コマンドを使用してバックアップを取得する方法について説明します。

6.7.1　更新凍結コマンドの対象となる運用

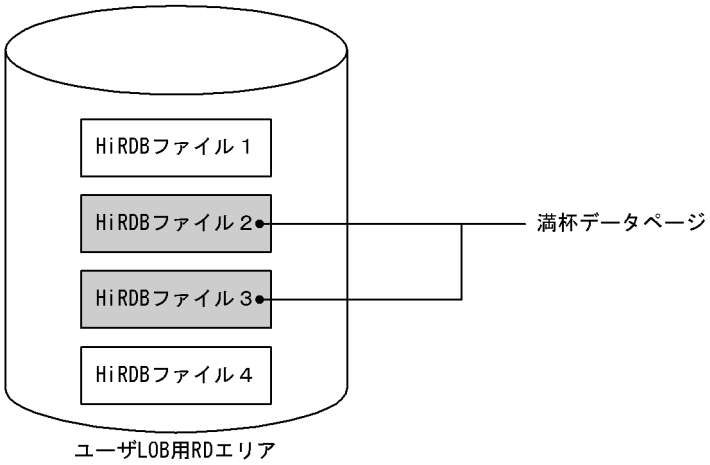
次に示す運用をする場合に更新凍結コマンド（pddbfrz コマンド）を使用して，ユーザ LOB 用 RD エリアのバックアップ取得時間の短縮を図ってください。

- データベース複写ユティリティ（pdcopy コマンド）以外の方法で，ユーザ LOB 用 RD エリアのバックアップを取得している
- バックアップの取得単位は HiRDB ファイル単位である
- 登録済みの LOB データの更新又は削除処理は通常発生しない運用である（LOB データの追加処理が通常の運用である）
- バックアップ取得対象のユーザ LOB 用 RD エリアは複数の HiRDB ファイルから構成されている

6.7.2　更新凍結コマンド（pddbfrz コマンド）とは

更新凍結コマンドを実行すると，ユーザ LOB 用 RD エリア中の満杯データページ（すべて割り当て済み）の HiRDB ファイルを**更新凍結状態**にします。更新凍結状態の HiRDB ファイル中のデータを更新及び削除できません。更新凍結コマンドの処理概要を次の図に示します。

図 6–3　更新凍結コマンドの処理概要



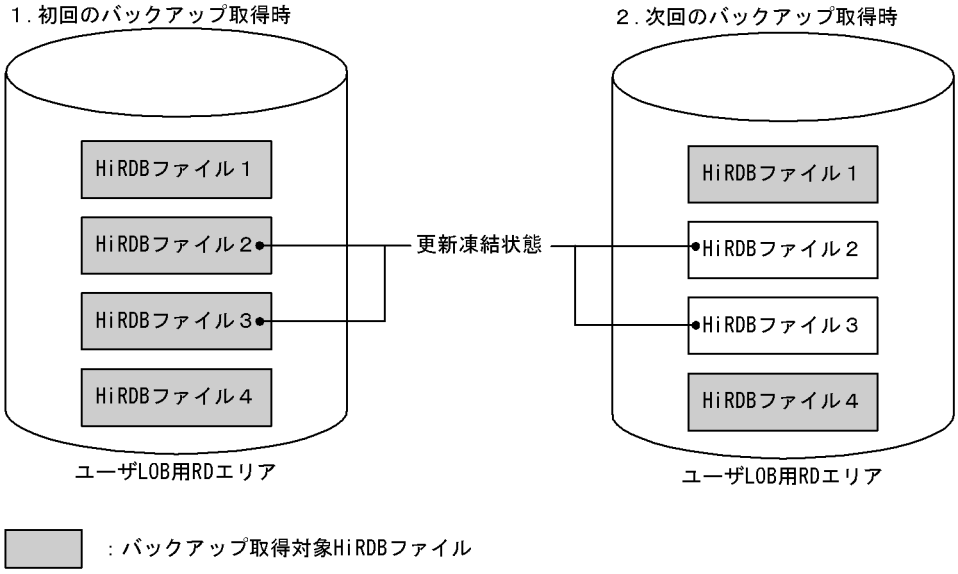
〔説明〕

- HiRDB ファイル 1～4 で構成されるユーザ LOB 用 RD エリアがあります。HiRDB ファイル 2，3 が満杯データページです。満杯データページの調査方法については，「満杯データページの調査方法」を参照してください。
- このユーザ LOB 用 RD エリアに更新凍結コマンドを実行すると，HiRDB ファイル 2，3 が**更新凍結状態**になります。更新凍結状態になった HiRDB ファイルは KFPH27024-I メッセージに表示されます。
- HiRDB ファイル 1 及び HiRDB ファイル 4 は**更新可能状態**になります。

(1)　更新凍結コマンドのバックアップ取得への利用方法

更新凍結コマンドのバックアップ取得への利用方法を次の図に示します。

図 6–4　更新凍結コマンドのバックアップ取得への利用方法



〔説明〕

バックアップ取得前に更新凍結コマンドを実行します。その結果，HiRDB ファイル 2，3 が更新凍結状態になりました。

1. 初回のバックアップ取得時には全 HiRDB ファイル（HiRDB ファイル 1～4）のバックアップを取得します。
2. HiRDB ファイル 2～3 は更新凍結状態のため，初回のバックアップ取得時と内容が変わっていません。したがって，次回のバックアップ取得時には HiRDB ファイル 2，3 のバックアップを取得する必要はありません。HiRDB ファイル 1 及び HiRDB ファイル 4 のバックアップだけを取得します。

備考

先頭 HiRDB ファイル（図「更新凍結コマンドのバックアップ取得への利用方法」の場合は HiRDB ファイル 1）には管理レコードがあるため，データ部が満杯になっても常に書き込みが発生します。このため，更新凍結状態になることはありません。したがって，先頭 HiRDB ファイルのバックアップは常に取得することになります。

(2)　バックアップ取得時の手順

バックアップ取得時の手順を次に示します。バックアップ取得時の運用例については，「運用例」を参照してください。

〈手順〉

1. 更新凍結コマンドで，満杯データページの HiRDB ファイルを更新凍結状態にします。
2. KFPH27024-I メッセージ又はデータベース状態解析ユティリティ（pddbst コマンド）で，更新凍結状態の HiRDB ファイルを確認します。
3. バックアップを取得します。1 回目のバックアップ取得時には全 HiRDB ファイルのバックアップを取得します。2 回目以降のバックアップ取得時には，更新可能状態の HiRDB ファイルのバックアップだけを取得します。更新凍結状態の HiRDB ファイルはバックアップを取得する必要はありません。

(3)　更新凍結状態を解除するには

HiRDB ファイルの更新凍結状態を解除するには，**pddbfrz -d** コマンドを実行します。

また，RD エリアを再初期化すると，更新凍結状態は解除されます。

(4)　更新凍結状態の HiRDB ファイルを調べるには

データベース状態解析ユティリティ（pddbst コマンド）を実行すると，HiRDB ファイルが更新凍結状態かどうかが分かります。

(5)　更新凍結状態にできない HiRDB ファイル

次に示す HiRDB ファイルは更新凍結状態にできません。

- ユーザ LOB 用 RD エリア以外の HiRDB ファイル
- ユーザ LOB 用 RD エリアの先頭 HiRDB ファイル
- プラグインインデクス格納用のユーザ LOB 用 RD エリアの HiRDB ファイル
- 未使用（表定義がない）のユーザ LOB 用 RD エリアの HiRDB ファイル
- データ型プラグインが作成する論理ファイルがある HiRDB ファイル※

注※

データ型プラグインが使用するユーザ LOB 用 RD エリアの場合，プラグインが使用する論理ファイルがある HiRDB ファイルは更新凍結状態になりません。論理ファイルは通常先頭 HiRDB ファイルにありますが，論理ファイルの拡張が発生すると，先頭以外の HiRDB ファイルに作成されことがあります。そのため，データ型プラグインを使用する場合，CREATE TABLE で作成されるプラグインの論理ファイルが先頭 HiRDB ファイルに格納できるように先頭 HiRDB ファイルの容量を見積もってください。

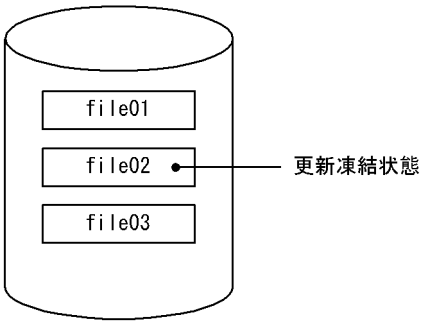
6.7.3　運用例

次に示す条件で運用されているユーザ LOB 用 RD エリア（LOB001）のバックアップ取得例を説明します。

- ユーザ LOB 用 RD エリア（LOB001）は三つの HiRDB ファイルから構成されている
- データの追加に伴い HiRDB ファイルを追加してユーザ LOB 用 RD エリア（LOB001）を拡張する

(1)　ユーザ LOB 用 RD エリア（LOB001）に対して更新凍結コマンドを実行します

```
pddbfrz -r LOB001
```

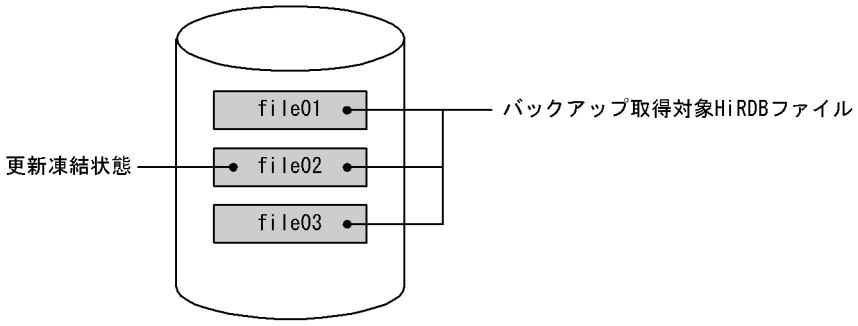


〔説明〕

満杯データページの HiRDB ファイル（file02）が更新凍結状態になりました。更新凍結状態になった HiRDB ファイルを KFPH27024-I メッセージ又はデータベース状態解析ユティリティ（pddbst コマンド）で確認してください。

(2)　バックアップを取得します

データベース複写ユティリティ（pdcopy コマンド）以外の方法で，ユーザ LOB 用 RD エリア（LOB001）のバックアップを取得してください。ユーザ LOB 用 RD エリア（LOB001）を構成する HiRDB ファイル単位にバックアップを取得します。



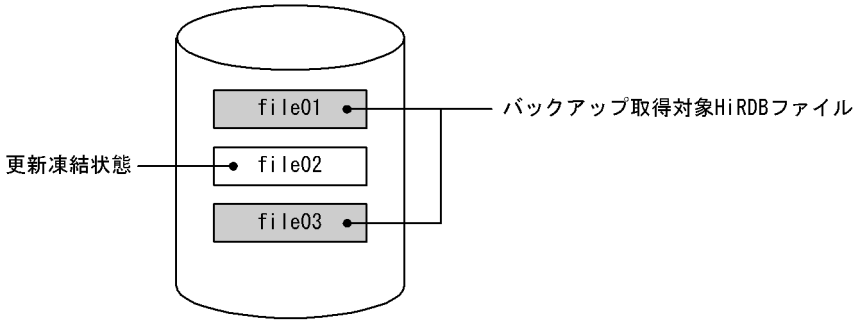
〔説明〕

全 HiRDB ファイル（file01～file03）のバックアップを取得します。

データベース複写ユティリティ（pdcopy コマンド）以外の方法でバックアップを取得する方法については，「バックアップ閉塞を使用してバックアップを取得する場合（pdcopy コマンド以外でバックアップを取得する場合）」を参照してください。

(3)　定期的にバックアップを取得します

1 週間に一度など定期的にバックアップを取得します。

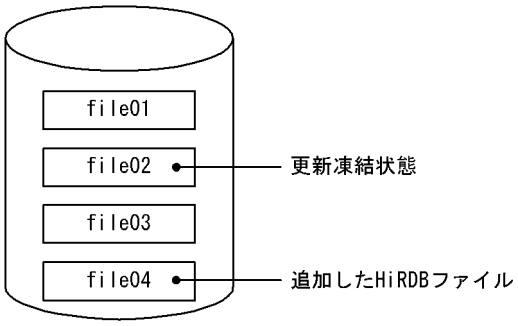


〔説明〕

更新可能状態の HiRDB ファイル（file01，file03）のバックアップを取得します。更新凍結状態の HiRDB ファイル（file02）はバックアップを取得する必要はありません。

(4)　HiRDB ファイルを追加してユーザ LOB 用 RD エリア（LOB001）を拡張します

ユーザ LOB 用 RD エリア（LOB001）の拡張方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

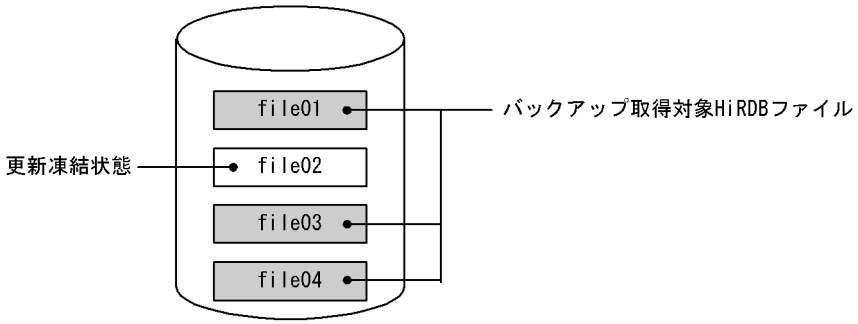


〔説明〕

HiRDB ファイル（file04）を追加してユーザ LOB 用 RD エリア（LOB001）を拡張しました。

(5)　バックアップを取得します

ユーザ LOB 用 RD エリア（LOB001）の拡張後，ユーザ LOB 用 RD エリアのバックアップを取得します。ファイルの使用中セグメント数が 0 の場合も，必ずバックアップを取得してください。理由については，「注意事項」の(1)を参照してください。



〔説明〕

更新可能状態の HiRDB ファイル（file01，file03，file04）のバックアップを取得します。更新凍結状態の HiRDB ファイル（file02）はバックアップを取得する必要はありません。

(6)　データベース状態解析ユティリティ（pddbst コマンド）で満杯データページが発生したかどうかを確認します

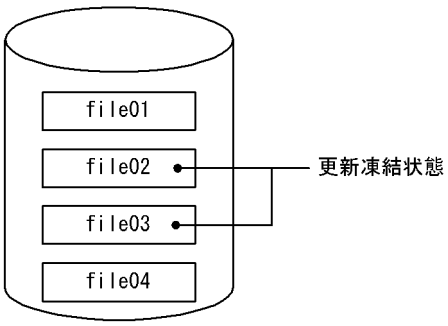
LOB データを追加していくと，満杯データページの HiRDB ファイルが新たに発生します。データベース状態解析ユティリティ（pddbst コマンド）で満杯データページが発生したかどうかを確認してください。満杯データページの調査方法については，「満杯データページの調査方法」を参照してください。

```
pddbst -r LOB001
```

(7)　満杯データページが新たに発生していたら更新凍結コマンドを実行します

(6)の結果，満杯データページが新たに発生していたら凍結コマンドを実行してください。

```
pddbfrz -r LOB001
```

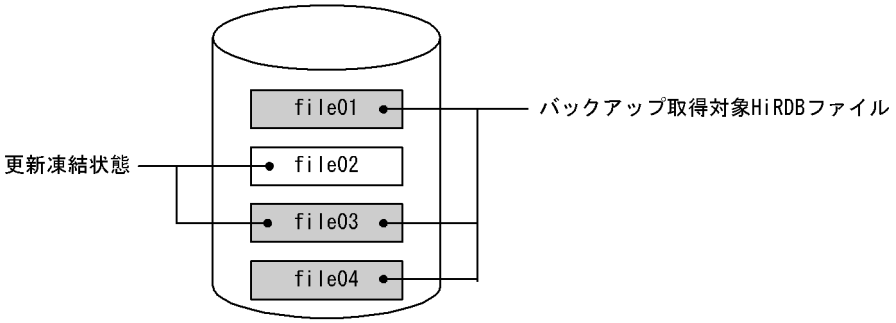


〔説明〕

満杯データページの HiRDB ファイル（file02，file03）が更新凍結状態になります。更新凍結状態になった HiRDB ファイルを KFPH27024-I メッセージ又はデータベース状態解析ユティリティ（pddbst コマンド）で確認してください。

(8)　バックアップを取得します

ユーザ LOB 用 RD エリアのバックアップを取得します。

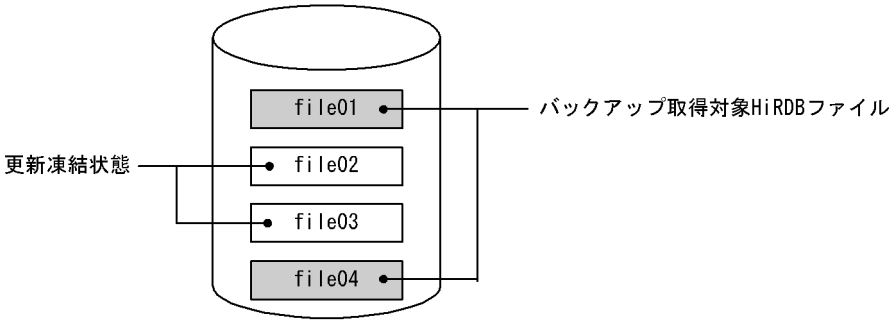


〔説明〕

HiRDB ファイル（file02）以外の HiRDB ファイル（file01，file03，file04）のバックアップを取得します。

(9)　定期的にバックアップを取得します

1 週間に一度など定期的にバックアップを取得します。



〔説明〕

更新可能状態の HiRDB ファイル（file01，file04）のバックアップを取得します。更新凍結状態の HiRDB ファイル（file02，file03）はバックアップを取得する必要はありません。

6.7.4　満杯データページの調査方法

HiRDB ファイルが満杯状態かどうかは，データベース状態解析ユティリティ（pddbst コマンド）の RD エリアの物理的解析を実行すると分かります。

●データベース状態解析ユティリティの出力例

```
RD Area Name   : LOB001
    :
    :
 Freeze Specified : Y
```

```
HiRDB File Name : C:¥rdarea¥lob001¥file01
 File Size      :       8192 segments   Extent Count :   1/ 24
 Segment        :       8190/      8192
 Freeze Status  : U
HiRDB File Name : C:¥rdarea¥lob001¥file02
 File Size      :       8192 segments   Extent Count :   1/ 24
 Segment        :       8000/      8192
 Freeze Status  : F    2001/12/06    20:37:05
HiRDB File Name : C:¥rdarea¥lob001¥file03
 File Size      :       8192 segments   Extent Count :   1/ 24
 Segment        :       1000/      8192
 Freeze Status  : U
```

〔説明〕

下線部分の情報から現在使用中の（データを格納する）HiRDB ファイルが C:¥rdarea¥lob001¥file03 ということが分かります。したがって，その一つ前の HiRDB ファイル（C:¥rdarea¥lob001¥file02）が満杯データページの HiRDB ファイルということになります。

備考

- LOB データはできるかぎり連続ページに確保されるため，データが 1 ページに収まらない場合は，この例のように HiRDB ファイルの最後の数ページが使用されないことがあります。
- LOB データを削除すると，使用中ページが 100%になりません。そのため，この例のように 8000 / 8192 となります。
- 最終 HiRDB ファイル（C:¥rdarea¥lob001¥file03）の使用中ページが最終ページになると，最終 HiRDB ファイルも満杯データページの HiRDB ファイルとみなします。

そのほか更新凍結状態に関係する項目を説明します。

Freeze Specified：更新凍結コマンドを実行した RD エリアかどうかを示します

Y：更新凍結コマンド実行済み

N：更新凍結コマンド未実行

Freeze Status：HiRDB ファイルが更新凍結状態かどうかを示しています。

F：更新凍結状態

U：更新可能状態

6.7.5　更新凍結コマンドを実行したユーザ LOB 用 RD エリアの操作

更新凍結コマンドを実行したユーザ LOB 用 RD エリアに対してできない操作があります。更新凍結コマンドを実行したユーザ LOB 用 RD エリアに対する操作の実行可否を次の表に示します。次の表で実行できない操作が必要な場合は，更新凍結コマンド（-d オプション指定）で，ユーザ LOB 用 RD エリアの更新凍結状態を解除してから操作をしてください。なお，その操作後，ユーザ LOB 用 RD エリアの全 HiRDB ファイルのバックアップを取得し直す必要があります。

表 6–8　更新凍結コマンドを実行したユーザ LOB 用 RD エリアに対する操作の実行可否

操作		HiRDB ファイルの状態	
		更新凍結状態	更新可能状態
SQL（データ操作）	INSERT	−	○
	UPDATE	×	○
	DELETE	×	○
SQL（表操作）	PURGE TABLE	×	×
	DROP TABLE	×	×
データベース作成ユティリティ	追加モード	−	○
	作成モード	×	×
データベース再編成ユティリティ（表の再編成，リロード）		×	×
リバランスユティリティ		×	×
データベース構成変更ユティリティ		○	○
データベース回復ユティリティ		○	○
運用コマンド		○	○

（凡例）

○：実行できます。

×：実行できません。

−：処理が発生しません。

6.7.6　RD エリアの自動増分との関係

自動増分を設定しているユーザ LOB 用 RD エリアの最終 HiRDB ファイルが更新凍結状態の場合は，RD エリアの自動増分が動作しません。この場合，HiRDB ファイルを追加してユーザ LOB 用 RD エリアを拡張すると自動増分が動作します。

また，最終 HiRDB ファイルが更新凍結状態のユーザ LOB 用 RD エリアに対しては，RD エリアの自動増分を設定できません。この場合，HiRDB ファイルを追加してユーザ LOB 用 RD エリアを拡張します。このユーザ LOB 用 RD エリアの拡張時に RD エリアの自動増分を設定してください。

6.7.7　注意事項

(1)　バックアップ取得対象の HiRDB ファイルについて

更新凍結状態の HiRDB ファイルは最低１回バックアップを取得してください。更新凍結状態でない HiRDB ファイルの管理領域はデータページの使用の有無と関係なく使用されます。このため，データベース状態解析ユティリティ（pddbst コマンド）の結果，HiRDB ファイルの使用中セグメント数が 0 であっても必ずバックアップを取得してください。

(2)　更新凍結状態へ変わるタイミングについて

HiRDB ファイルの状態が変わるのは更新凍結コマンドの実行時だけです。更新凍結コマンドの実行時に満杯データページでない HiRDB ファイルが，その後のデータ追加で満杯データページになっても HiRDB ファイルの状態は変わりません（更新凍結状態にはなりません）。満杯データページ後に更新凍結コマンドを実行しないと，HiRDB ファイルの状態は変わりません（更新凍結状態にはなりません）。

7 データベースの更新ログを取得しないときの運用

この章では，次に示すモードで UAP 又はユティリティを実行するときの運用方法について説明します。

・更新前ログ取得モード

・ログレスモード

7.1　データベースの更新ログ取得方式

7.1.1　データベースの更新ログ取得方式の概要

HiRDB は，UAP 又はユティリティ※によって更新されたデータベースの履歴情報（システムログ中のデータベースの更新ログ）をシステムログファイルに取得しています。ただし，ユーザの指定によっては，このデータベースの更新ログを取得しないこともできます。データベースの更新ログを取得しないと，その分の処理時間が短縮されます。したがって，UAP 又はユティリティの実行時間を短縮できます。

注※　ここでいうユティリティとは，次に示すユティリティのことです。

- データベース作成ユティリティ
- データベース再編成ユティリティ
- リバランスユティリティ

(1)　データベースの更新ログ取得方式

UAP 又はユティリティを実行するときのデータベースの更新ログ取得方式には，次に示す三つのモードがあります。

ログ取得モード

ロールバック及びロールフォワードに必要なデータベース更新ログを取得します。

更新前ログ取得モード

ロールバックに必要なデータベース更新ログだけを取得します。

ログレスモード

データベース更新ログを取得しません。

(2)　データベースの更新ログ取得方式の指定方法

データベースの更新ログ取得方式の指定方法を次の表に示します。

表 7–1　データベースの更新ログ取得方式の指定方法

条件	データベースの更新ログ取得方式の指定方法
UAP の場合	クライアント環境定義の **PDDBLOG** オペランドで，データベースの更新ログ取得方式を指定します。PDDBLOG オペランドでは，ログ取得モード又はログレスモードを指定できます。更新前ログ取得モードは指定できません。
データベース作成ユティリティの場合	データベース作成ユティリティの **-l** オプションで，データベースの更新ログ取得方式を指定します。
データベース再編成ユティリティの場合	データベース再編成ユティリティの **-l** オプションで，データベースの更新ログ取得方式を指定します。

条件	データベースの更新ログ取得方式の指定方法
リバランスユティリティ	リバランスユティリティの **-l** オプションで，データベースの更新ログ取得方式を指定します。リバランスユティリティの-l オプションでは，ログ取得モード又はログレスモードを指定できます。更新前ログ取得モードは指定できません。
ユーザ LOB 用 RD エリアを使用している場合	ユーザ LOB 用 RD エリアに格納されているデータについては，CREATE TABLE の **RECOVERY** オペランドで，データベースの更新ログ取得方式を指定します。

(3) RECOVERY オペランドに関する注意事項

RECOVERY オペランドで指定したデータベースの更新ログ取得方式は，PDDBLOG オペランド又は-l オプションの指定によって変更される場合があります。RECOVERY オペランドと PDDBLOG オペランド又は-l オプションの関係を次の表に示します。

表 7–2　RECOVERY オペランドと PDDBLOG オペランド又は-l オプションの関係

PDDBLOG オペランド 又は-l オプションの指定	RECOVERY オペランドの指定	UAP（又はユティリティ） 実行時に仮定される値
ALL a ログ取得モード	ALL	ALL
	PARTIAL	PARTIAL
	NO	NO
p 更新前ログ取得モード※	ALL	PARTIAL
	PARTIAL	PARTIAL
	NO	NO
NO n ログレスモード	ALL	NO
	PARTIAL	NO
	NO	NO

（凡例）

ALL 又は a ：ログ取得モード

PARTIAL 又は p ：更新前ログ取得モード

NO 又は n ：ログレスモード

注※　PDDBLOG オペランドでは，更新前ログ取得モードを指定できません。

(4) データベースの更新ログ取得方式による運用方法の違い

データベースの更新ログ取得方式によって次に示す運用方法が異なります。

- UAP 又はユティリティが異常終了したときの HiRDB の処理と HiRDB 管理者の処置
- データベースを回復できる時点

(a) UAP 又はユティリティが異常終了したときの HiRDB の処理と HiRDB 管理者の処置

UAP 又はユティリティが異常終了したときの HiRDB の処理と HiRDB 管理者の処置を次の表に示します。

表 7–3 UAP 又はユティリティが異常終了したときの HiRDB の処理と HiRDB 管理者の処置

データベースの更新ログ取得方式	HiRDB の処理	HiRDB 管理者の処置
ログ取得モード	更新した RD エリアの状態を UAP 実行前の状態又は異常終了直前の同期点にロールバックします。	UAP 実行前の状態にロールバックされた場合は，UAP を再実行してください。異常終了直前の同期点にロールバックされた場合は，同期点以降の処理を実行してください。
更新前ログ取得モード		
ログレスモード	ロールバックしません。更新した RD エリアを**障害閉塞**（**ログレス閉塞**）します。RD エリアの内容は破壊されます。	UAP 実行前に取得したバックアップを入力情報として，データベース回復ユティリティで RD エリアを回復してください。その後，UAP 又はユティリティを再実行してください。

(b) データベースを回復できる時点

データベース回復ユティリティを使用してデータベースを回復できる時点を次の表に示します。

表 7–4 データベースを回復できる時点

データベースの更新ログ取得方式	データベースを回復できる時点
ログ取得モード	バックアップ取得時点又はバックアップ取得時点以降の任意の同期点
更新前ログ取得モード	バックアップ取得時点
ログレスモード	

(5) バックアップについての注意（重要）

1. ログレスモード又は更新前ログ取得モードの UAP 又はユティリティの実行中に，更新可能モード（-M s 指定）でバックアップを取得しないでください。
2. ログレスモード又は更新前ログ取得モードの UAP 又はユティリティの実行後は，次に示すモードでバックアップを取得してください。
   - 参照・更新不可能モード（-M x 指定）
   - 参照可能モード（-M r 指定）

7.2　更新前ログ取得モードで UAP 又はユティリティを実行するときの運用

7.2.1　更新前ログ取得モードで UAP 又はユティリティを実行するときの運用方法

実行者　HiRDB 管理者

更新前ログ取得モードで UAP 又はユティリティを実行するときの運用方法について説明します。なお，UAP の場合は，ユーザ LOB 用 RD エリアに対してだけ（CREATE TABLE の RECOVERY オペランド），更新前ログ取得モードを実行できます。

ここでは，主にユティリティを更新前ログ取得モードで実行するときの運用方法について説明します。

(1)　利点

更新後のデータベース更新ログを取得しないため，その分の処理時間が短縮されます。したがって，ログ取得モードを適用したときに比べて，UAP 又はユティリティの実行時間を短縮できます。

(2)　適用基準

データベース作成ユティリティ又はデータベース再編成ユティリティを実行するときの推奨値（省略値）です。

(3)　注意事項（重要）

(a)　UAP 又はユティリティの実行後に，更新した RD エリアのバックアップを取得してください

pdrstr コマンドで RD エリアを回復するとき，入力情報のシステムログ中に更新前ログ取得モードで取得したログが入っていると pdrstr コマンドがエラーになります。したがって，UAP 又はユティリティの実行後にバックアップを取得しないと，pdrstr コマンドで RD エリアを回復する必要が生じた場合，RD エリアを最新の状態に回復できません（UAP 又はユティリティ実行後の反映処理を回復できません）。RD エリアは UAP 又はユティリティ実行前の状態にしか回復できません。

(b)　ほかのユーザが更新対象 RD エリアを更新しないようにしてください

RD エリア 1 に表 T1，T2 を格納しているとします。T1 を更新前ログ取得モードで更新し，T2 をログ取得モードで更新した場合，pdrstr コマンドで RD エリアを回復する必要が生じた場合，RD エリアを最新の状態に回復できません。入力情報のシステムログ中に更新前ログ取得モードで取得したログが入っていると pdrstr コマンドがエラーになるため，T1 更新前の同期点にしか回復できません。

したがって，ユティリティの場合はユティリティを実行する前に pdhold コマンドで更新対象 RD エリアを閉塞してください。UAP の場合はほかのユーザが更新対象 RD エリアを更新しないようにしてください。

(4) 運用方法

(a) ユティリティ（データベース再編成ユティリティ）を更新前ログ取得モードで実行する場合の運用方法

ユティリティ（データベース再編成ユティリティ）を更新前ログ取得モードで実行する場合の運用方法については，「データベースの更新ログ取得方式の選択」及び「表の再編成（例題）」を参照してください。

(b) 更新前ログ取得モードでユーザ LOB 用 RD エリアを更新する UAP を実行する場合の運用方法

更新前ログ取得モードでユーザ LOB 用 RD エリアを更新する UAP を実行する場合の運用方法を次に示します。

〈手順〉

1. 更新前ログ取得モードでユーザ LOB 用 RD エリアを更新する UAP を実行します。
2. pdlogswap -d sys -w コマンドで，システムログファイルをスワップします。
3. pdcopy コマンドで，更新対象 RD エリアのバックアップを取得します。
   バックアップの取得方法については，「例題 8（RD エリア単位にバックアップを取得する場合）」を参照してください。

7.3　ログレスモードで UAP 又はユティリティを実行するときの運用

7.3.1　ログレスモードで UAP 又はユティリティを実行するときの運用方法

実行者　HiRDB 管理者

ログレスモードで UAP 又はユティリティを実行するときの運用方法について説明します。

(1)　利点

データベース更新ログを取得しないため，その分の処理時間が短縮されます。したがって，UAP 又はユティリティの実行時間を短縮できます。

(2)　適用基準

1. 大量のデータを追加，更新，又は削除する UAP を実行する場合
2. 大量のデータをデータロードする場合
3. 表データが多い表を再編成（リロード）する場合
4. ログレスモードの UAP が更新する RD エリアを占有できる場合

(3)　注意事項（重要）

1. クライアント環境定義に PDDBLOG ＝ NO を指定している場合，この UAP で更新する表は排他モード（EX）で排他制御します。このため，同じ表をほかの UAP で同時に更新しないでください。
2. クライアント環境定義に PDDBLOG ＝ ALL を指定（又は省略）していて，かつその UAP で更新する RD エリアにユーザ LOB 用 RD エリアが含まれている場合，表定義で RECOVERY オペランドに NO を指定したユーザ LOB 用 RD エリアを更新するときは，ほかの UAP がその RD エリアを更新しないようにしてください。ほかの UAP が更新すると，ログレスモードの UAP が異常終了したときに，ほかの UAP による RD エリアの更新内容が無効になります。
3. ログレスモードの UAP 又はユティリティが異常終了すると，UAP 又はユティリティが更新した表を格納する RD エリアが**障害閉塞**（ログレス閉塞）します。HiRDB はその RD エリアを回復しません。HiRDB 管理者が RD エリアを回復する必要があります。したがって，ログレスモードの UAP 又はユティリティが異常終了すると，障害閉塞した RD エリアを回復するまで，その RD エリアはほかの UAP 又はユティリティからアクセスできなくなります。なお，RD エリアはバックアップ取得時点にしか回復できません。
4. ログレスモードの UAP 又はユティリティの実行中は，シンクポイントダンプを取得しません。このため，ログレスモードの UAP 又はユティリティと同時にほかの UAP 又はユティリティを実行しているときに，システム障害が発生すると，再開始に必要な時間が増大します。したがって，ログレスモードで UAP 又はユティリティを実行する場合は，そのほかの UAP 又はユティリティを実行しないようにしてください。

(4)　ほかの機能との関連

1. データベース定義ユティリティで CREATE INDEX を実行する場合，インデクスを作成する表に行データが格納されていると，CREATE INDEX を実行するときにインデクスを一括作成します。このとき，クライアント環境定義に PDDBLOG ＝ NO と指定していると，データベースの更新ログを取得しないでインデクスを一括作成します。したがって，このとき HiRDB 管理者はログレスモードの運用を実行してください。
2. RD エリアの障害閉塞は，HiRDB の次回正常開始時にも引き継がれます。オープン状態又はクローズ状態も引き継がれます。

(5)　運用方法

(a)　ユティリティ（データベース再編成ユティリティ）をログレスモードで実行する場合の運用方法

ユティリティ（データベース再編成ユティリティ）をログレスモードで実行する場合の運用方法については，「データベースの更新ログ取得方式の選択」及び「例題７（ログレスモードで再編成する場合）」を参照してください。

(b)　ログレスモードで UAP を実行する場合の運用方法

ログレスモードで UAP を実行する場合の運用方法を次に示します。

〈手順〉

1. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
2. **pdcopy** コマンドで，更新対象 RD エリアのバックアップを取得します。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。
3. ログレスモードの UAP を実行します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドで，更新対象 RD エリアのバックアップを取得します。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

注

バックアップとシステムログを使用して UAP 実行前の状態に RD エリアを回復できる場合は，手順 2.のバックアップを取得する必要はありません。ただし，プラグインインデクスの一括作成をする場合は，ユーザ LOB 用 RD エリアのバックアップを必ず取得してください。

(6)　UAP 又はユティリティが異常終了した場合

UAP 又はユティリティが異常終了すると，その UAP 又はユティリティが更新した RD エリアは**障害閉塞**（ログレス閉塞）となります。HiRDB 管理者は，障害閉塞をした RD エリアを回復してください。RD エリアの回復方法については，「データベースをバックアップ取得時点に回復する方法」を参照してください。

(a)　障害閉塞となった RD エリアを知るには

pddbls コマンドを実行すると，どの RD エリアが障害閉塞になっているかが分かります。

(b)　UAP 又はユティリティを実行する前にバックアップを取得しなかった場合

UAP 又はユティリティを実行する前にバックアップを取得しなかった場合は，**範囲指定の回復**で RD エリアを回復してください。範囲指定の回復については，「データベースを最新の同期点に回復する方法」を参照してください。

- このときの入力情報は，障害閉塞となった RD エリアの最新のバックアップとそのバックアップ以降のシステムログを格納したアンロードログファイル（システムログファイル）となります。
- データベース回復ユティリティの-T オプションの回復終了時刻には，UAP を開始した時刻を指定してください。
- 上記の方法でデータベースを回復した後に再度 UAP 又はユティリティを実行してください。

(c)　初期データから RD エリアを回復する場合（UAP が異常終了した場合）

データベース回復ユティリティを使用しないで，初期データから RD エリアを回復する場合の手順を次に示します。なお，ユティリティが異常終了した場合は，この手順で RD エリアを回復できません。

〈手順〉

1. **PURGE TABLE** 文で，UAP が更新したすべての表の行データを削除します。※
2. **pdrels** コマンドで，障害閉塞となった RD エリアの閉塞状態を解除します。
3. **pdload** コマンドで初期データを表に格納し直します。

注※

一時表用 RD エリアが障害閉塞になった場合，一時表に対して PURGE TABLE 文は実行できません。そのため，一時表用 RD エリアを回復する場合は，pdmod コマンドで一時表用 RD エリアを再初期化してください。

(d)　初期データから RD エリアを回復する場合（ユティリティが異常終了した場合）

データベース回復ユティリティを使用しないで，初期データから RD エリアを回復する場合の手順を次に示します。

〈手順〉

1. **pdclose** コマンドで回復対象の RD エリアをクローズします。
2. **pdmod** コマンドで RD エリアを再初期化します。
3. **pdopen** コマンドで RD エリアをオープンします。
4. **pdload** コマンドで初期データを表に格納し直します。
5. **pdrels** コマンドで，障害閉塞となった RD エリアの閉塞状態を解除します。

8 システムの稼働環境を知る方法（システムの状態監視）

この章では，システムの稼働環境を知る方法（システムの状態監視）について説明します。

8.1　メッセージログでシステムの実行状況を確認する

実行者　HiRDB 管理者

システムの実行状況をメッセージログで確認してください。メッセージログは，HiRDB が出力するメッセージの履歴情報として，次に示す場合に利用します。

- HiRDB が出力したメッセージを確認する場合
- HiRDB に異常が発生したときに異常箇所及び原因を調べる場合

8.1.1　メッセージログを参照する（メッセージログの出力先）

HiRDB が出力したメッセージは，メッセージログとしてメッセージログファイル（ファイル名は%PDDIR%¥spool¥pdlog1 又は pdlog2）とイベントログに出力されます。

メッセージログファイルに出力されたメッセージは，**pdcat** コマンドを実行すると参照できます。イベントログに出力されたメッセージは，各 OS で決められた手順に従って参照してください。

参考

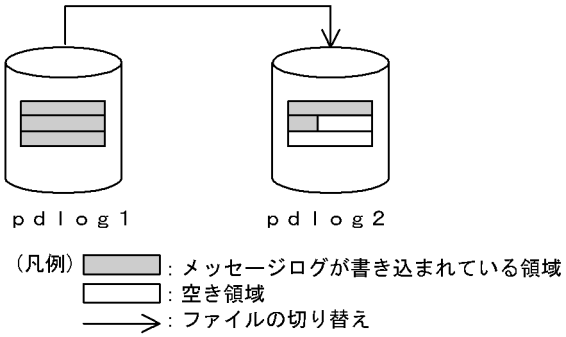
- pdcat コマンドを実行した場合，メッセージログファイル中のメッセージログをマージし，古い方のメッセージログから順に表示します。
- HiRDB/パラレルサーバの場合，メッセージログの出力方式（システムマネジャのサーバマシンに出力するか，各サーバマシンに出力するか）を選択できます。詳細については，「メッセージログの出力方式の選択（HiRDB/パラレルサーバ限定）」を参照してください。

8.1.2　メッセージログファイルの運用

HiRDB は二つのメッセージログファイル（pdlog1 と pdlog2）を交互に使用します。片方のメッセージログファイルの容量が一杯になると，HiRDB はそのメッセージログファイルへのメッセージログの出力をやめて，もう片方のメッセージログファイルに出力します。このとき，切り替え先のメッセージログファイルに格納されている古いメッセージログは消去されます。

メッセージログファイルの切り替えを次の図に示します。

図 8–1　メッセージログファイルの切り替え



〔説明〕

pdlog1 が一杯になると，メッセージログの出力先ファイルが pdlog2 に切り替わります。そのとき，pdlog2 に格納されていた古いメッセージログは消去されます。

参考

- メッセージログが出力されているメッセージログファイルを知りたい場合は，エクスプローラなどでファイルの最終更新日時を確認してください。最終更新日時が新しい方が，現在使用しているメッセージログファイルです。
- HiRDB の開始時（すべての開始モード）には，メッセージログファイルの切り替えは行われません。HiRDB の終了時に出力先となっていたメッセージログファイルに続けて出力されます。

- **メッセージログファイルが切り替わった場合に HiRDB 管理者がすること**

  メッセージログファイルが切り替わった場合，KFPS01910-I メッセージが切り替え前のメッセージログファイルに出力されます。切り替え前のメッセージログファイル中のメッセージログを保存する場合は，メッセージログファイルのバックアップを取得してください。

- **メッセージログファイルの容量を変更する場合**

  pd_mlg_file_size オペランドでメッセージログファイルの容量を設定します。必要に応じて容量を変更してください。

8.1.3　メッセージログの出力方式の選択（HiRDB/パラレルサーバ限定）

システムマネジャのユニットが停止したり，システムマネジャのユニットとほかのユニット間で通信障害が発生したりすると，メッセージログが正しく出力されないで，次に示す不具合が発生することがあります。

- メッセージが出力されない
- メッセージの出力順序が変わる
- メッセージ発行元のユニットのイベントログにメッセージログが出力されるが，その出力が遅延する

これらの不具合を防ぐために**メッセージログの出力分散化**を検討してください。

> **参考**
>
> HiRDB/パラレルサーバの場合，通常はシステムマネジャがあるサーバマシンにメッセージログが出力されます。

(1)　メッセージログの出力分散化とは

HiRDB/パラレルサーバの場合，メッセージログの出力方式（メッセージログの出力先）を次に示すどちらかに選択できます。

1. メッセージログの出力先は，システムマネジャがあるサーバマシンのメッセージログファイル及びイベントログになります。
2. メッセージログの出力先は，各サーバマシンのメッセージログファイル及びイベントログになります。メッセージ出力元のサーバマシンに出力されます。

通常は 1 の方式でメッセージログが出力されます。必要に応じて 2 の方式に変更してください。2 の方式を**メッセージログの出力分散化**といいます。

pd_mlg_msg_log_unit オペランドで方式を選択します。このオペランドを省略した場合は，1 の方式が適用されます。

1 の方式の場合と 2 の方式の場合のメッセージログの出力方式をそれぞれ次の図に示します。

図 8–2　通常のメッセージログの出力方式（1 の方式の場合）

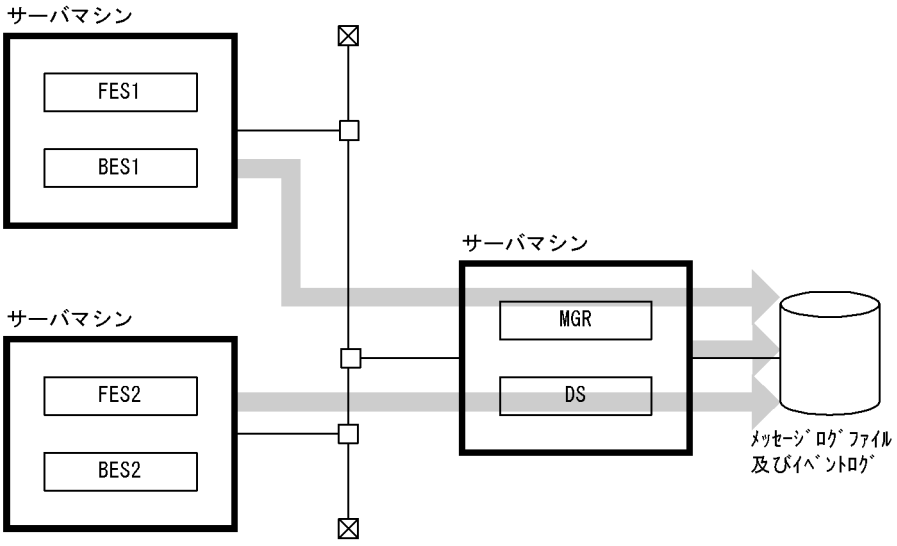
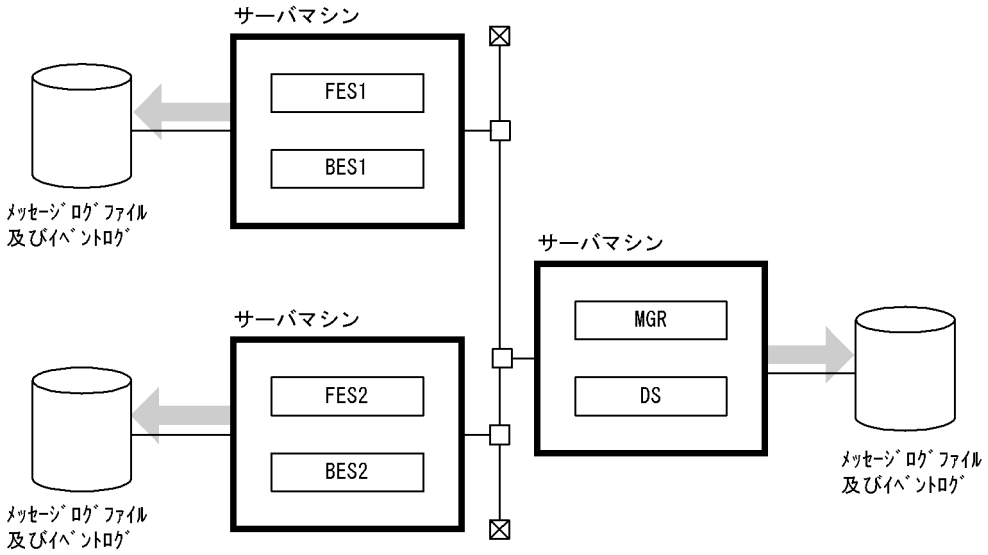


図 8–3　出力分散化したときのメッセージログの出力方式（2 の方式の場合）



参考

pdcat コマンドでメッセージログを参照する場合，pdcat コマンドを実行したユニットのメッセージログだけが参照できます。このため，メッセージログの出力分散化をした場合，参照したいメッセージログがあるユニットで pdcat コマンドを実行してください。

(2)　メッセージログの出力方式の選択基準

メッセージログの出力方式とメッセージログの出力方式のメリット及びデメリットを次の表に示します。

表 8–1　メッセージログの出力方式

選択する方式※	条件	メッセージログの出力先
1 の方式（manager）	通常時	システムマネジャがあるサーバマシンのメッセージログファイル及びイベントログに出力されます。
	MGR ユニットの障害時又は通信障害時	各サーバマシンのイベントログに出力されます。出力されるメッセージログは一部のメッセージログだけです。また，メッセージログが正しく出力されないことがあります。
2 の方式（local）	通常時	各サーバマシンのメッセージログファイル及びイベントログに出力されます。
	MGR ユニットの障害時又は通信障害時	

注※　（　）内は pd_mlg_msg_log_unit オペランドの指定値です。

表 8–2　メッセージログの出力方式のメリット及びデメリット

選択する方式※	メリット	デメリット
1 の方式（manager）	メッセージログをシステムマネジャのサーバマシンで一元管理できるため，メッセージの監視作業が local 指定時より簡単になります。	システムマネジャのユニットが停止したり，システムマネジャのユニットとほかのユニット間で通信障害が発生したりした場合，メッセージログが正しく出力されないことがあります。
2 の方式（local）	システムマネジャのユニットが停止したり，システムマネジャのユニットとほかのユニット間で通信障害が発生したりした場合でも，メッセージログが正しく出力されます。	メッセージログが各サーバマシンに出力されるため，メッセージの監視作業が manager 指定時より煩雑になります。

注※　（　）内は pd_mlg_msg_log_unit オペランドの指定値です。

(3)　環境設定

メッセージログの出力分散化をする場合の環境設定手順を次に示します。

〈手順〉

1. 各サーバマシンにメッセージログファイルが作成されるため，メッセージログファイルの容量を見積もり直してください。容量見積もりの結果を pd_mlg_file_size オペランドに指定します。
2. pd_mlg_msg_log_unit オペランドに local を指定します。
3. 各サーバマシンの時刻を同じに設定します。

> **参考**
>
> 各サーバマシンの時刻が異なる場合，各サーバマシンに出力されるメッセージログの付加情報の時刻が異なります。このため，メッセージログを参照する場合は，サーバマシン間の時刻のずれを考慮する必要があります。

(4)　メッセージログの出力分散化を選択するときの注意事項

- JP1/BASE を使用してメッセージの監視をする場合，全サーバマシンのイベントログを監視対象にする必要があります。
- JP1/Performance Management - Agent Option for HiRDB で収集するレコードに PD_MLOG を指定する場合，システムマネジャがあるサーバマシンに出力されるメッセージログだけが収集対象になります。

8.1.4　イベントログへのメッセージ出力抑止

HiRDB はシステムの稼働状況を知らせるメッセージとして，障害情報やトランザクション情報，システムファイルの情報などをイベントログに出力します。これらのメッセージは，HiRDB の稼働状況を知るための重要な情報ですが，稼働環境によっては大量のメッセージが出力されるため，必要なメッセージを見付

けるのに時間が掛かります。この場合，イベントログに出力されるメッセージのうち，必要のないメッセージの出力を抑止して，メッセージ検索時の運用性向上を図ることができます。

(1)　環境設定

出力抑止するメッセージを **pdmlgput** オペランドで指定します。このオペランドでは次に示すような指定もできます。

- 全メッセージを出力抑止する
- 出力対象とするメッセージを指定する
- 指定したメッセージの重要度表示を変更して出力する（「イベントログに出力するメッセージの重要度の変更」を参照）

(2)　メッセージの出力抑止時の処理

メッセージの出力抑止を行った場合，イベントログに出力されるメッセージが出力抑止されます。出力抑止されたメッセージは，メッセージログファイルに出力されます。メッセージの出力抑止の使用有無によるメッセージ出力処理の違いを次の表に示します。

表 8–3　メッセージの出力抑止の使用有無によるメッセージ出力処理の違い

出力抑止の指定	システムマネジャが あるサーバマシン		システムマネジャが ないサーバマシン	
	イベントログ	メッセージ ログファイル	イベントログ	メッセージ ログファイル
出力抑止を指定しない場合	○	○	×※1	×
出力抑止を指定する場合	×	○※2	×	○※1※3

（凡例）

○：メッセージを出力します。

×：メッセージを出力しません。

注

HiRDB/シングルサーバの場合はシステムマネジャがあるサーバマシンの欄が該当します。

注※1

HiRDB/パラレルサーバの場合，システムマネジャがあるサーバマシンのイベントログ及びメッセージログファイルにメッセージが出力されます。ただし，次に示す場合は，メッセージは出力されないか，又はメッセージ出力元サーバマシンのイベントログに出力されることがあります。

- システムマネジャユニットの開始完了前，又は障害が発生している場合
- メッセージログサーバの異常終了後，又は再起動前

- 待機系システム

注※2

イベントログ及びメッセージログファイルの両方に出力されているメッセージについては，メッセージログファイルにだけ出力されます。

注※3

HiRDB/パラレルサーバの場合，システムマネジャがないサーバマシンの%PDDIR%¥spool 下に pdlog1 及び pdlog2 ファイル（メッセージログファイル）を作成し，そこにイベントログに出力していたメッセージが出力されます。

(3) メッセージの出力抑止の例外について

次に示す場合はメッセージの出力を抑止しません。イベントログに出力します。

(a) メッセージログファイルに障害が発生した場合

メッセージログファイルに障害が発生した場合，障害発生を示すメッセージをイベントログに出力します。また，障害発生のため，メッセージログファイルに出力できなかったメッセージもイベントログに出力します。

(b) HiRDB システム定義の定義エラーがある場合

HiRDB システム定義ファイルの内容に誤りがあるときに出力されるメッセージは出力抑止されません。この場合，イベントログにエラーメッセージが出力されます。

(4) 使用例

(a) 例 1

すべてのメッセージをイベントログに出力しないで，メッセージログファイルに出力します。この場合の pdmlgput オペランドの指定例を次に示します。

```
pdmlgput -s N -c ALL
```

(b) 例 2

KFPS01820 及び KFPS00105 メッセージをイベントログに出力しないで，メッセージログファイルに出力します。この場合の pdmlgput オペランドの指定例を次に示します。

```
pdmlgput -s N -m KFPS01820,KFPS00105
```

(c) 例 3

KFPH00211 及び KFPH00212 メッセージだけをイベントログに出力し，そのほかのメッセージはメッセージログファイルに出力します。この場合の pdmlgput オペランドの指定例を次に示します。

```
pdmlgput -s N -c ALL                          ...1
pdmlgput -s Y -m KFPH00211,KFPH00212         ...2
```

〔説明〕

1. すべてのメッセージをイベントログに出力しないで，メッセージログファイルに出力します。
2. KFPH00211 及び KFPH00212 メッセージは，イベントログ及びメッセージログファイルに出力します。

(5) 注意事項

(a) メッセージログファイルの容量が一杯になったときの処理について

メッセージログファイルの容量（pd_mlg_file_size オペランドの値）が一杯になった場合，メッセージログの出力先をもう一方のメッセージログファイルに切り替えます。切り替えを行う場合，KFPS01910-I メッセージがイベントログに出力されます。KFPS01910-I メッセージの出力を抑止している場合，KFPS01910-I メッセージは切り替え前のメッセージログファイルに出力されます。メッセージログファイルを監視している場合，KFPS01910-I メッセージが出力された後に，監視対象のメッセージログファイルをもう一方のメッセージログファイルに変えてください。

(b) メッセージログファイルの容量について

メッセージの出力抑止を適用する場合，メッセージログファイルの容量を十分に確保してください。メッセージログファイルの容量は，pd_mlg_file_size オペランドで指定します。

メッセージの出力抑止を適用すると，稼働環境によってはメッセージログファイルに大量のメッセージが出力されることがあります。このため，メッセージログファイルの容量を十分に確保しないと，メッセージログファイルが短い間隔で上書きされて，メッセージログファイルだけに出力されているシステムの稼働状況や，トラブルシュート情報などの重要なメッセージが消去されてしまう可能性があります。

(6) ほかの機能との関連（HiRDB/パラレルサーバ限定）

メッセージの出力抑止とメッセージログの出力分散化を組み合わせて使用するときの HiRDB の処理を次の表に示します。

表 8–4 メッセージの出力抑止とメッセージログの出力分散化を組み合わせて使用するときの HiRDB の処理

条件		メッセージの出力先	
		メッセージの出力抑止を適用しない場合	メッセージの出力抑止を適用する場合
メッセージログの出力分散化	適用しない場合	• システムマネジャがあるサーバマシンのメッセージログファイル • システムマネジャがあるサーバマシンのイベントログ	• システムマネジャがあるサーバマシンのメッセージログファイル

条件		メッセージの出力先	
		メッセージの出力抑止 を適用しない場合	メッセージの出力抑止 を適用する場合
	適用する場合	• 各サーバマシンのメッセージログファイル • 各サーバマシンのイベントログ	• 各サーバマシンのメッセージログファイル

8.1.5　イベントログに出力するメッセージの重要度の変更

HiRDB をイベントログに出力されるメッセージの重要度で監視している場合，監視対象の重要度ではないメッセージも監視するために，ユーザが監視が必要と判断するメッセージの重要度表示を変更して，そのメッセージをイベントログに出力できます。例えば，HiRDB のメッセージの重要度が W（警告）のメッセージだけを監視している場合，システムの状況を表す，重要度が I（インフォメーション）のメッセージを監視できないことがあります。このようなとき，メッセージの重要度表示を W に変更して，イベントログに出力し，監視漏れを防ぎます。

(1)　環境設定

重要度表示を変更するメッセージ ID と，変更後の重要度を **pdmlgput** オペランドに指定します。なお，このオペランドでは，メッセージの出力抑止も指定できるため，-s オプションには Y を指定します。

注意事項

重要度を変更できるのは，イベントログに出力されるメッセージだけです。そのため，同じメッセージがメッセージログファイルに出力されている場合，メッセージログファイルのメッセージの重要度は変更されません。

参考

HiRDB のメッセージの重要度と，OS がイベントログに出力するメッセージの優先度との対応を次の表に示します。

HiRDB のメッセージの重要度	イベントログに出力されるメッセージの優先度
E	EVENTLOG_ERROR_TYPE
W	EVENTLOG_WARNING_TYPE
Q	EVENTLOG_INFORMATION_TYPE
I	

イベントログに出力されるメッセージの優先度の詳細については，OS のマニュアルを参照してください。

(2) 例題

インフォメーションメッセージの KFPH00211-I 及び KFPH00212-I メッセージの重要度表示を警告メッセージに変更し，KFPH00211-W 及び KFPH00212-W メッセージとしてイベントログに出力します（メッセージログファイルに出力されるメッセージの重要度はインフォメーションのままです）。この場合の pdmlgput オペランドの指定例を次に示します。

```
pdmlgput -s Y -l W -m KFPH00211,KFPH00212
```

〔説明〕

-s：イベントログにメッセージを出力するため，Y を指定します。

-l：変更後の重要度 W（警告メッセージ）を指定します。

-m：重要度表示を変更したいメッセージ ID を指定します。

8.1.6 ユティリティ実行時のインフォメーションメッセージ出力抑止

ユティリティを実行した場合，ユティリティの処理状況を知らせるインフォメーションメッセージが，端末の標準出力，イベントログ，及びメッセージログファイルに大量に出力されることがあります。これによって，次に示す不具合が発生するおそれがあります。

- 標準出力の画面がスクロールしたり，ファイルがラップアラウンドしたりして，重要なメッセージが消えてしまう
- 短時間に大量のメッセージがバックエンドサーバからシステムマネジャに送信され，それが原因で通信エラーになる（HiRDB/パラレルサーバの場合）

このような不具合が発生する場合，ユティリティが出力するインフォメーションメッセージの出力を抑止してください。なお，インフォメーションメッセージの出力を抑止できるのは，次に示すユティリティです。

- データベース作成ユティリティ（pdload）
- データベース再編成ユティリティ（pdrorg）
- リバランスユティリティ（pdrbal）

(1) インフォメーションメッセージ出力抑止レベルの指定

ユティリティが出力するメッセージの出力を抑止するには，ユティリティ実行時，次に示すオプションにインフォメーションメッセージ出力抑止レベル（lvl0，lvl1，又は lvl2）を指定します。

- pdload の-m オプション

- pdrorg の-m オプション
- pdrbal の-m オプション

オプションの指定値と，メッセージの出力先を次の表に示します。

表 8–5　オプションの指定値とメッセージの出力先

オプションの指定値	意味	出力先		
		標準出力	イベントログ及びメッセージログファイル	ワークファイル※
lvl0	出力を抑止しない	○	○	×
lvl1	標準出力への出力を抑止する	×	○	×
lvl2	標準出力，イベントログ及びメッセージログファイルへの出力を抑止する	×	×	○

（凡例）

○：メッセージを出力します。

×：メッセージを出力しません。

注※

ワークファイル出力先ディレクトリ（pd_tmp_directory オペランドで指定したディレクトリ）直下に作成されるファイルです。pd_tmp_directory オペランドを省略している場合，HiRDB がワークファイル出力先を決定します。詳細については，マニュアル「HiRDB システム定義」を参照してください。

ワークファイル名については，マニュアル「HiRDB コマンドリファレンス」で，それぞれのユティリティの-m オプションの説明を参照してください。

ワークファイルにメッセージを出力するのはトラブルシュート目的（ユティリティの動作を確認するため）であり，通常，メッセージを確認する必要がない場合は lvl1 を指定することをお勧めします。

(2)　出力を抑止するメッセージ

ユティリティは，次に示す基準に従って出力を抑止するメッセージを決定します。

1. 大量に出力されるインフォメーションメッセージ（メッセージ ID が「-I」）
   - トランザクション単位に処理の開始又は終了を示すメッセージ
   - 生成したワークファイルの名称を示すメッセージ
   - 生成したワークファイルの削除を示すメッセージ
   - RD エリア数に比例して増加するメッセージ
   - インデクス数に比例して増加するメッセージ
   - サーバ数に比例して増加するメッセージ
2. 1.で出力を抑止したメッセージと同意のメッセージ（出力が少ない場合でも出力抑止の対象とする）

8.2　UAP 又はユティリティの実行時間が長い場合

実行者　HiRDB 管理者

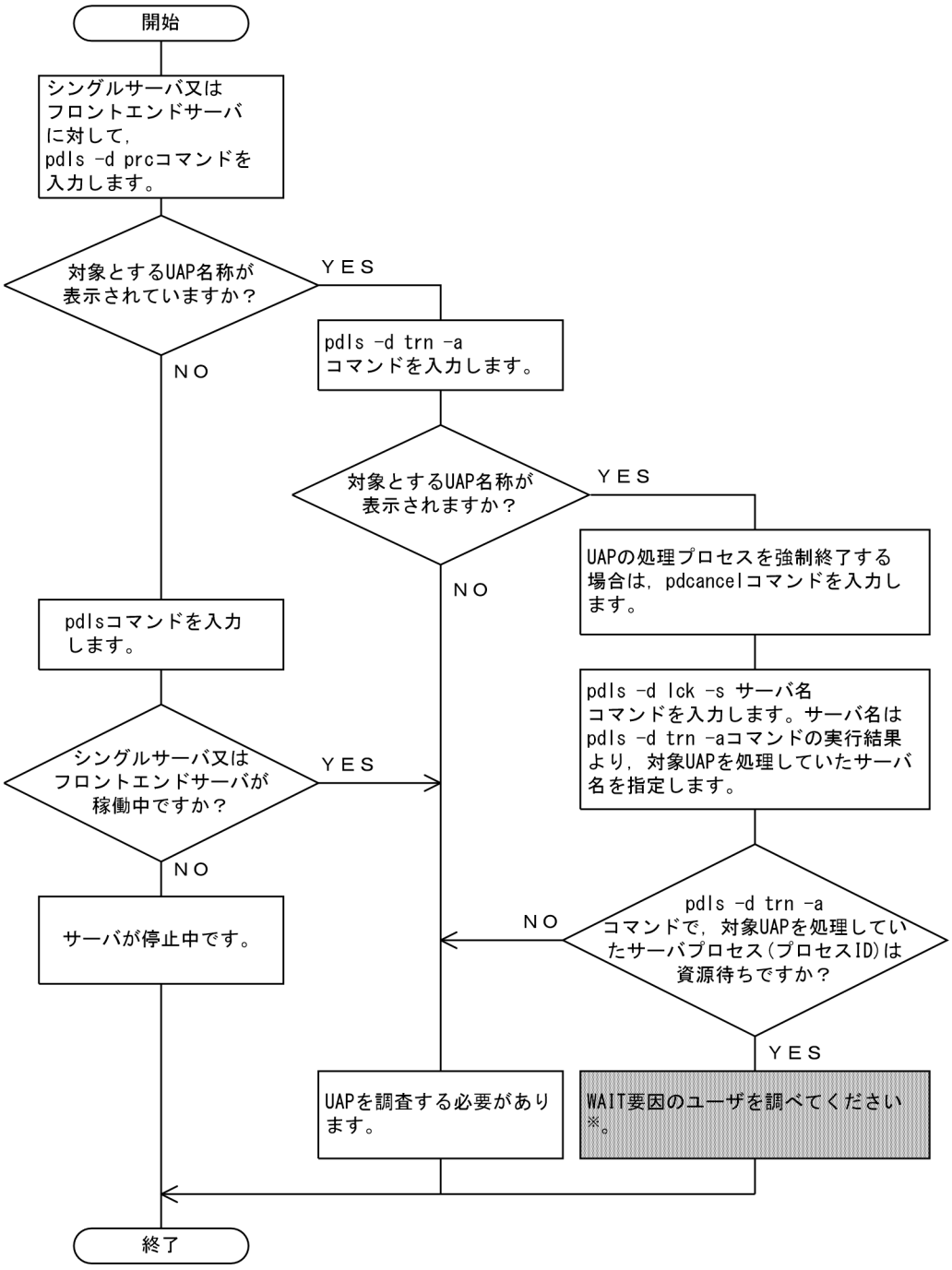
UAP 又はユティリティの処理時間が予想より長い場合，UAP 又はユティリティの実行状況を運用コマンドで調べてください。UAP 又はユティリティの実行状況を調べる方法を次の図に示します。

なお，次に示すユティリティは，次の図に示す方法では調査できません。

- データベース初期設定ユティリティ
- データベース複写ユティリティ
- データベース回復ユティリティ

これらのユティリティは，**pdls -d prc -a** コマンドでユティリティの処理が実行中かどうかが分かるだけです。

図 8–4　UAP 又はユティリティの実行状況を調べる方法



注

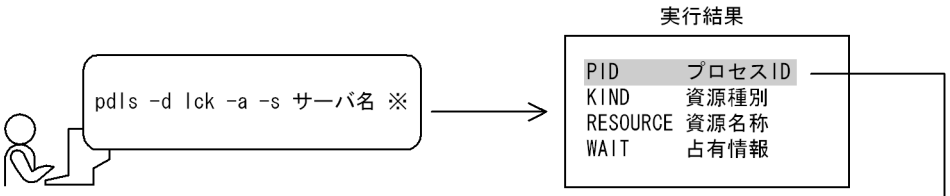
説明文中の UAP とはユティリティを含んでいます。

注※

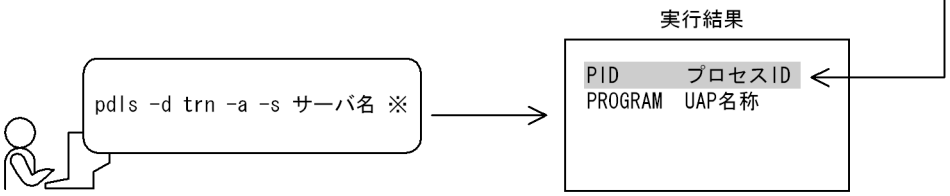
手順は，次の図を参照してください。

図 8–5　WAIT 要因のユーザを調べる方法

1．資源の占有状態を知るために″pdls -d lck -a -s サーバ名″コマンドを入力します。これで，待ち状態のサーバプロセスが待っている資源種別及び資源名称と同じ資源種別及び資源名称を占有しているサーバプロセスのプロセスIDが分かります。



2．資源を占有しているUAPを知るには，″pdls -d trn -a -s サーバ名″コマンドを入力します。資源を占有していたプロセスIDが示すUAPが，ＷＡＩＴ要因となっているUAPです。



注※　図８－４で指定したサーバ名を指定します。

8.3　HiRDB の開始又は終了処理が長い場合

実行者　HiRDB 管理者

HiRDB の開始又は終了処理が長い場合，**pdls -d svr** コマンドで HiRDB の開始又は終了処理の実行状況が分かります。このコマンドの実行結果の見方については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

8.4　RD エリアの状態を知りたい場合

実行者　HiRDB 管理者

UAP 又はユティリティを実行する前に，対象となる RD エリアが，UAP 又はユティリティを実行できる状態かどうかを調べる必要があります。RD エリアの状態を調べるには，**pddbls** コマンドを実行します。

注意事項

- 定義している RD エリア数が多い場合，pddbls コマンドのオプションに ALL を指定すると，コマンド処理時間が長くなります。この場合は，オプションに RD エリア名を指定するか，又はサーバ名を指定してください。
- 閉塞している RD エリアだけを知りたい場合は，オプションに-b を指定すると，閉塞している RD エリアが表示されます。

8.5　共用メモリの使用状況を知りたい場合

8.5.1　共用メモリの使用状況の調査方法

実行者　HiRDB 管理者

(1)　共用メモリの使用状況を調べる方法

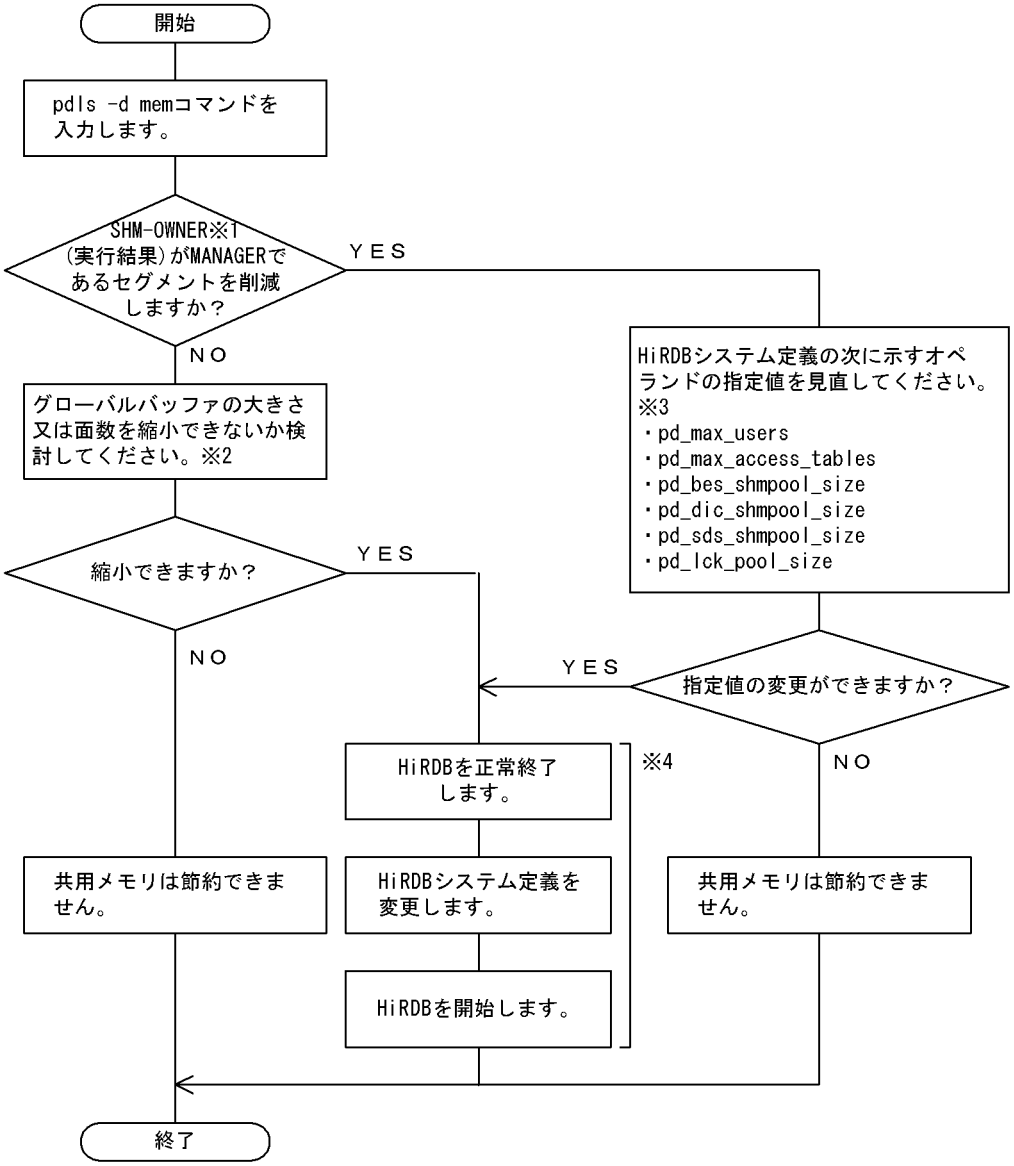
pdls -d mem コマンドで，共用メモリの使用状況が分かります。

(2)　共用メモリを節約できるかどうかを調べる方法

共用メモリが不足しているため，プロセスが実行できない場合，共用メモリの使用状況を調べてください。OS の問題であれば，OS の機能で対策できないかを検討してください。

OS の機能で対策できない場合は，HiRDB が使用する共用メモリを節約できるかどうかを次の図に示す方法で検討してください。

図 8–6　HiRDB が使用する共用メモリを節約できるかどうかを調べる方法



注※１

pdls -d mem コマンドの実行結果です。

注※２

グローバルバッファの設計については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

注※３

これらのオペランドの指定値の目安については，マニュアル「HiRDB システム定義」を参照してください。

注※４

システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できます。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必

要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

8.6　デッドロックが発生した場合

実行者　HiRDB 管理者

ここでは，デッドロック又はタイムアウトが発生したときに出力される情報について説明します。ここで説明する項目は次のとおりです。

- 基本項目
- デッドロック情報の出力内容
- タイムアウト情報の出力内容
- 資源種別と資源情報
- 資源情報の見方

8.6.1　基本項目

デッドロック又はタイムアウトが発生すると，HiRDB は次の表に示す情報を出力します。

表 8–6　デッドロック又はタイムアウトが発生したときに出力される情報

出力する条件	出力される情報	内容
pd_lck_deadlcok_check オペランドに Y を指定し，かつ pd_lck_deadlock_info オペランドに Y を指定する	デッドロック情報	• サーバ内の複数のトランザクション間に発生したデッドロックの情報です。 • HiRDB/パラレルサーバの場合，サーバ間でデッドロックが発生したときは，タイムアウト情報として出力されます。 • デッドロックについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。
pd_lck_deadlock_info オペランドに Y を指定する	タイムアウト情報	排他待ち限界経過時間を超えたことを示す情報です。
pd_lck_deadlock_check オペランドに Y を指定し，かつ pd_lck_deadlock_info オペランドに Y を指定し，かつ pd_lck_deadlock_sql_info オペランドに Y を指定する	デッドロック SQL 情報	• サーバ内で発生したデッドロックに関連するトランザクションが実行した SQL の情報です。 • デッドロックについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(1) デッドロック情報，タイムアウト情報，及びデッドロック SQL 情報の出力先は？

(a) デッドロック情報及びタイムアウト情報

デッドロック情報及びタイムアウト情報は，次のファイルに出力されます。このファイルを，デッドロック・タイムアウト情報ファイルといいます。

- **%PDDIR%¥spool¥pdlckinf¥ファイル名**

「ファイル名」は，デッドロック又はタイムアウトが発生した日時を基に，HiRDB が次のように決定します。

```
（例）
・デッドロック又はタイムアウトが発生した日時が10月3日9時29分56秒のとき
  ファイル名：0ct3092956
                └┘└┘└┘
          10月3日  9時29分56秒
・デッドロック又はタイムアウトが発生した日時が10月10日18時6分0秒のとき
  ファイル名：0ct10180600
                └┘└┘└┘
          10月10日 18時6分0秒
```

(b) デッドロック SQL 情報

デッドロック SQL 情報は，次のファイルに出力されます。このファイルを，デッドロック SQL 情報ファイルといいます。

- **%PDDIR%¥spool¥pdlckinf¥ファイル名**

「ファイル名」は，デッドロックが発生した日時，サーバ，デッドロックエラーになるトランザクションのトランザクション識別子，デッドロックエラーになるトランザクションのスレッド ID を基に，HiRDB が次のように決定します。

```
（例）
・デッドロックが発生した日時が10月3日9時29分59秒，
  発生サーバがbes1，トランザクション識別子がSYS1UNT100020000，スレッドIDが1のとき
    ファイル名：0ct3092959_bes1_SYS1UNT100020000_1.sql
                  └┘└┘└┘
               10月3日 9時29分59秒

・デッドロックが発生した日時が10月10日18時6分0秒，
  発生サーバがbes1，トランザクション識別子がSYS1UNT100020000，スレッドIDが1のとき
    ファイル名：0ct10180600_bes1_SYS1UNT100020000_1.sql
                  └┘└┘└┘
               10月10日18時6分0秒
```

(2) HiRDB 管理者がすることは？

これらの情報を出力するには，システム共通定義で次の指定が必要です。

1. **pd_lck_deadlock_check** オペランドでデッドロックの発生を監視することを指定します。

2. **pd_lck_deadlock_info** オペランドでデッドロック情報及びタイムアウト情報を出力することを指定します。
3. **pd_lck_deadlock_sql_info** オペランドでデッドロック SQL 情報を出力することを指定します。
4. **pd_lck_wait_timeout** オペランドで排他待ち限界経過時間を指定します。

(3) デッドロック情報，タイムアウト情報，及びデッドロック SQL 情報の参照方法は？

次のメッセージが出力された場合に，デッドロック情報，タイムアウト情報，又はデッドロック SQL 情報を参照してください。

- KFPA11911-E（デッドロックの発生を示すメッセージ）
- KFPS00441-I（デッドロック情報の出力を示すメッセージ）
- KFPS00451-I（タイムアウト情報の出力を示すメッセージ）
- KFPS00455-I（デッドロック SQL 情報の出力を示すメッセージ）

デッドロック情報，タイムアウト情報，及びデッドロック SQL 情報はメモ帳などで参照できます。ファイル名は上記の KFPS00441-I メッセージ，KFPS00451-I メッセージ，及び KFPS00455-I メッセージに出力されます。

デッドロック情報の出力内容については，「デッドロック情報の出力内容」を参照してください。

タイムアウト情報の出力内容については，「タイムアウト情報の出力内容」を参照してください。

デッドロック SQL 情報の出力内容については，「デッドロック SQL 情報の出力内容」を参照してください。

(4) デッドロック情報，タイムアウト情報，及びデッドロック SQL 情報への対処方法は？

デッドロック又はタイムアウトの発生頻度は，UAP のアクセス順序を変更することや，UAP で排他の範囲を広くしてアクセスすることなどで低減できます。このため，参照したデッドロック情報，タイムアウト情報，又はデッドロック SQL 情報を基に，デッドロック又はタイムアウトの要因となった資源及び SQL を見直すことが重要です。デッドロック又はタイムアウトの要因となった資源ごとの対処方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。デッドロック情報とデッドロック SQL 情報を突き合わせる方法については，「デッドロック情報とデッドロック SQL 情報を突き合わせる方法」を参照してください。

(5) 不要なデッドロック・タイムアウト情報ファイル及びデッドロック SQL 情報ファイルを削除しましょう

HiRDB 管理者は不要となったデッドロック・タイムアウト情報ファイル及びデッドロック SQL 情報ファイルを次に示す(a)，(b)又は(c)の方法で削除してください。

(a)　HiRDB のコマンドで削除する方法

pdcspool コマンドで削除します。pdcspool コマンドを実行すると，%PDDIR%¥spool 下にあるトラブルシュート情報ファイルが削除されます。したがって，デッドロック・タイムアウト情報ファイル又はデッドロック SQL 情報ファイルだけを削除する場合は，「OS の機能で削除する方法」を参照してください。

(b)　HiRDB の機能で削除する方法

pd_spool_cleanup_interval_level オペランドを省略するか all を指定すると，定期的に HiRDB がデッドロック・タイムアウト情報ファイル及びデッドロック SQL 情報ファイルを削除します。通常は 24 時間ごとに削除します。この削除間隔を pd_spool_cleanup_interval オペランドで変更できます。

また，pd_spool_cleanup_level オペランドを省略するか all を指定すると，HiRDB の開始時に HiRDB がデッドロック・タイムアウト情報ファイル及びデッドロック SQL 情報ファイルを削除します。

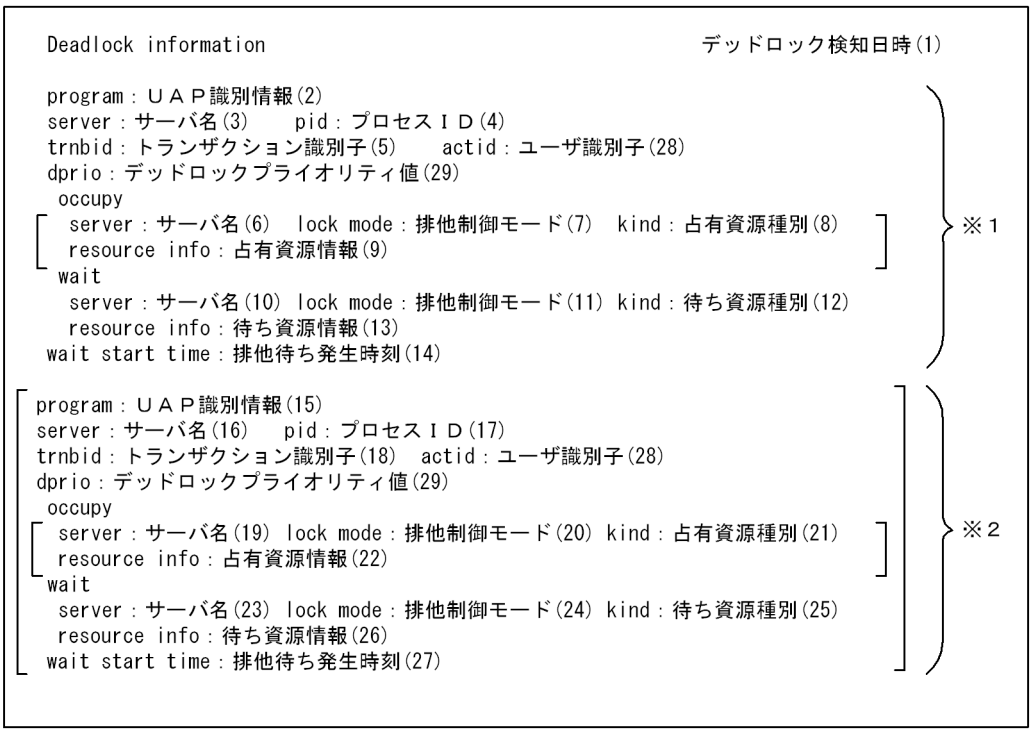
(c)　OS の機能で削除する方法

OS の機能（OS の del コマンドなど）で削除してください。del コマンドについては，OS のマニュアルを参照してください。

8.6.2　デッドロック情報の出力内容

デッドロック情報の出力内容を次の図に示します。

図 8–7　デッドロック情報の出力内容



注　［　］内は複数出力される場合があります。
注※１　デッドロックエラーとなったトランザクションを送信したサーバの情報
注※２　デッドロックを構成したトランザクションを送信したサーバの情報

デッドロック情報の出力内容を以下に説明します。

（1）デッドロック検知日時

HiRDB がデッドロックを検知した日付と時刻（月　日　時:分:秒 西暦）を示します。

●デッドロックとなったトランザクションを送信したサーバの情報

（2）～（14）の情報は，デッドロックとなったトランザクションが最初に資源を占有したときの情報です。

（2）UAP 識別情報

デッドロックとなったトランザクションを送信したサーバに接続している UAP の識別情報を示します。

ここで表示する情報は pdls -d prc 又は pdls -d trn コマンドで表示される PROGRAM に該当します。ただし，一部のユティリティでは表示できません。表示できない場合は"********"を表示します。また，再開始処理で回復中のトランザクションの場合は"Rerun"が表示されます。

（3）サーバ名

デッドロックとなったトランザクションを送信したサーバのサーバ名を示します。

（4）プロセス ID

デッドロックとなったトランザクションを送信したサーバプロセスのプロセス ID を示します。

（5）トランザクション識別子

デッドロックとなったトランザクションの識別子を示します。

先頭が_cmd の場合は pdhold コマンドで確保した排他です。

先頭が_utl の場合は pdcopy コマンドで確保した排他です。

●デッドロックとなったトランザクションが占有したすべての資源の情報

（6）～（9）の情報は，デッドロックの原因となったすべての資源の情報です。このトランザクションが資源を占有していなかった場合，各情報には空白が出力されます。

（6）サーバ名

デッドロックとなったトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（7）排他制御モード

デッドロックとなったトランザクションが占有した資源に掛けた排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（8）占有資源種別

デッドロックとなったトランザクションが占有した資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（9）占有資源情報

デッドロックとなったトランザクションが占有した資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

●デッドロックとなったトランザクションが排他待ちとなった資源の情報

（10）～（14）の情報は，デッドロックの原因となった資源の情報です。

（10）サーバ名

デッドロックとなったトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（11）排他制御モード

デッドロックとなったトランザクションが排他待ちとなっている資源に掛けようとした排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（12）待ち資源種別

デッドロックとなったトランザクションが排他待ちとなっている資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（13）待ち資源情報

デッドロックとなったトランザクションが排他待ちとなっている資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

（14）排他待ち発生時刻

デッドロックとなったトランザクションが排他待ちとなった時刻（時：分：秒）を示します。

●デッドロックを構成したトランザクションを送信したサーバの情報

（15）～（29）の情報は，デッドロックを構成したトランザクションが最初に資源を占有したときの情報です。複数出力される場合があります。

（15）UAP 識別情報

デッドロックを構成したトランザクションを送信したサーバに接続している UAP の識別情報を示します。

ここで表示する情報は pdls -d prc 又は pdls -d trn コマンドで表示される PROGRAM に該当します。ただし，一部のユティリティでは表示できません。表示できない場合は"********"を表示します。また，再開始処理で回復中のトランザクションの場合は"Rerun"が表示されます。

（16）サーバ名

デッドロックを構成したトランザクションを送信したサーバの識別子を示します。

（17）プロセス ID

デッドロックを構成したトランザクションを送信したサーバプロセスのプロセス ID を示します。

（18）トランザクション識別子

デッドロックを構成したトランザクションの識別子を示します。

●デッドロックを構成したトランザクションが占有したすべての資源の情報

（19）～（22）の情報は，デッドロックの原因となったすべての資源の情報です。このトランザクションが資源を占有していなかった場合，各情報には空白が出力されます。

（19）サーバ名

デッドロックを構成したトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（20）排他制御モード

デッドロックを構成したトランザクションが占有した資源に掛けた排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（21）占有資源種別

デッドロックを構成したトランザクションが占有した資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（22）占有資源情報

デッドロックを構成したトランザクションが占有した資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

●デッドロックを構成したトランザクションが排他待ちとなった資源の情報

（23）～（29）の情報は，デッドロックの原因となった資源の情報です。

（23）サーバ名

デッドロックを構成したトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（24）排他制御モード

デッドロックを構成したトランザクションが排他待ちとなっている資源に掛けようとした排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（25）待ち資源種別

デッドロックを構成したトランザクションが排他待ちとなっている資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（26）待ち資源情報

デッドロックを構成したトランザクションが排他待ちとなっている資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

（27）排他待ち発生時刻

デッドロックを構成したトランザクションが排他待ちとなった時刻（時：分：秒）を示します。

（28）ユーザ識別子

HiRDB に接続しているユーザを一意に識別するために，HiRDB が動的に割り当てる通番を示します。なお，ユーザ識別子を使って SQL 文を特定できる場合があります。マニュアル「HiRDB コマンドリファレンス」の「pdobils（SQL オブジェクト用バッファの統計情報表示）」を参照してください。

（29）デッドロックプライオリティ値

デッドロックとなったトランザクションのデッドロックプライオリティ値を示します。

デッドロック情報の出力例を次の図に示します。

図 8–8　デッドロック情報の出力例

```
Deadlock information                                    Jun  2 06:12:43: 1995

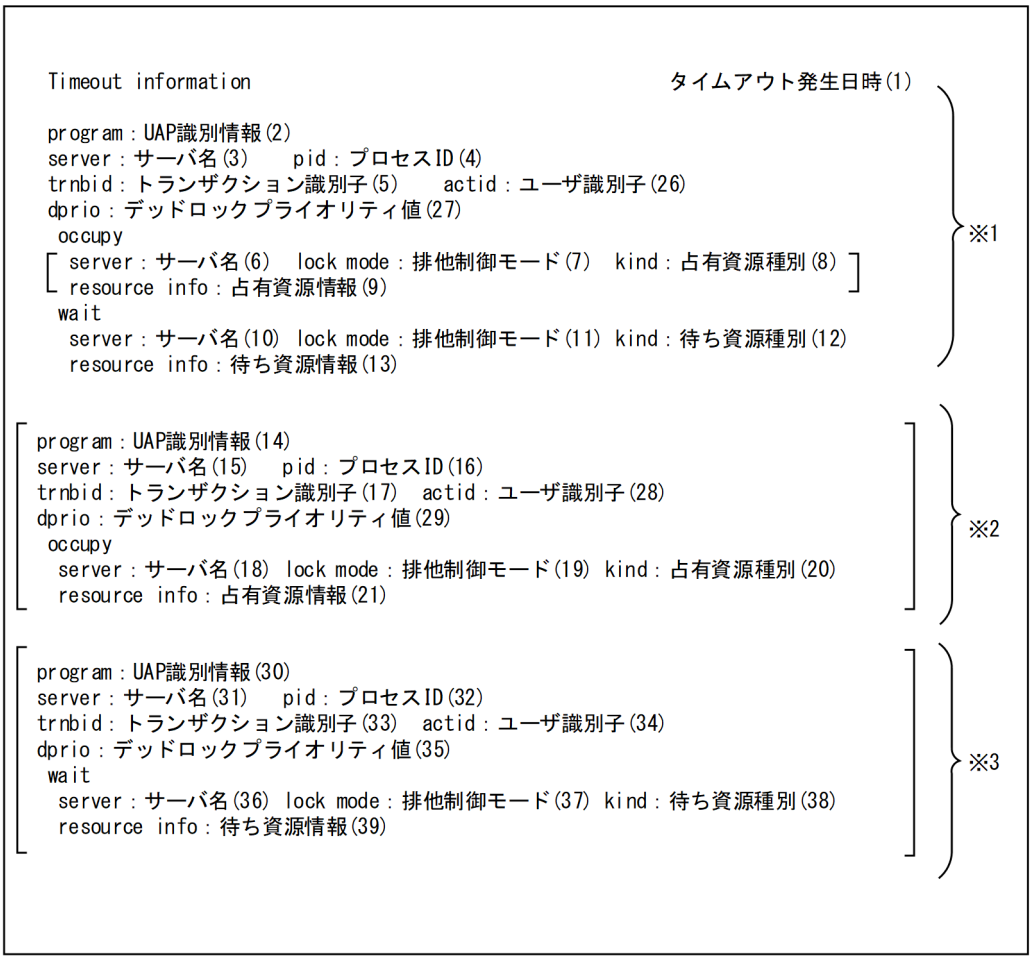
program : SPPY415
server : SDS     pid : 5251
trnbid : q192u19200000000     actid : 1-1-4     dprio : 64
 occupy
  server : SDS   lock mode : PR    kind : 0007
  resource info : 006000000190000017d0002000000
 wait
  server : SDS   lock mode : EX    kind : 0007
  resource info : 00600000019000001790002000000
wait start time : 06:12:43

program : SPPE201
server : SDS     pid : 5249
trnbid : q192u19200000003     actid : 1-1-6     dprio : 64
 occupy
  server : SDS   lock mode : PR    kind : 0007
  resource info : 00600000019000001790002000000
 wait
  server : SDS   lock mode : EX    kind : 0007
  resource info : 006000000190000017d0002000000
wait start time : 06:12:43
```

8.6.3　タイムアウト情報の出力内容

タイムアウト情報の出力内容を次の図に示します。

図 8-9　タイムアウト情報の出力内容



注　［　］内は複数出力される場合があります。
注※1　タイムアウトとなったトランザクションを送信したサーバの情報
注※2　タイムアウトの要因となったトランザクションを送信したサーバの情報
注※3　タイムアウトとなったトランザクションのほかに，(13)で排他待ちになっている
　　トランザクションを送信したサーバの情報

タイムアウト情報の出力内容を以下に説明します。

（1）タイムアウト発生日時

HiRDB がタイムアウトを検知した日付と時刻（月　日　時：分：秒　西暦）を示します。

●タイムアウトとなったトランザクションを送信したサーバの情報

（2）～（13），（26）～（27）の情報は，タイムアウトとなったトランザクションが占有した資源の情報です。

（2）UAP 識別情報

タイムアウトとなったトランザクションを送信したサーバに接続している UAP の識別情報を示します。トランザクションを送信したのがユティリティの場合，ユティリティ名を示します。

ここで表示する情報は pdls -d prc 又は pdls -d trn コマンドで表示される PROGRAM に該当します。ただし，一部のユティリティでは表示できません。表示できない場合は"********"を表示します。また，再開始処理で回復中のトランザクションの場合は"Rerun"が表示されます。

（3）サーバ名

タイムアウトとなったトランザクションを送信したサーバのサーバ名を示します。

（4）プロセス ID

タイムアウトとなったトランザクションを送信したサーバプロセスのプロセス ID を示します。

（5）トランザクション識別子

タイムアウトとなったトランザクションの識別子を示します。

先頭が_cmd の場合は pdhold コマンドで確保した排他です。

先頭が_utl の場合は pdcopy コマンドで確保した排他です。

（26）ユーザ識別子

HiRDB に接続しているユーザを一意に識別するために，HiRDB が動的に割り当てる通番を示します。

（27）デッドロックプライオリティ値

トランザクションのデッドロックプライオリティ値を示します。

●タイムアウトとなったトランザクションが占有したすべての資源の情報

（6）～（9）の情報は，タイムアウトとなったトランザクションが占有したすべての資源の情報です。このトランザクションが資源を占有していなかった場合，各情報には空白が出力されます。

（6）サーバ名

タイムアウトとなったトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（7）排他制御モード

タイムアウトとなったトランザクションが占有した資源に掛けた排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（8）占有資源種別

タイムアウトとなったトランザクションが占有した資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（9）占有資源情報

タイムアウトとなったトランザクションが占有した資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

●タイムアウトとなったトランザクションが排他待ちとなった資源の情報

（10）～（13）の情報は，タイムアウトとなったトランザクションが排他待ちとなった資源の情報です。

（10）サーバ名

タイムアウトとなったトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（11）排他制御モード

タイムアウトとなったトランザクションが排他待ちとなっている資源に掛けようとした排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（12）待ち資源種別

タイムアウトとなったトランザクションが排他待ちとなっている資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（13）待ち資源情報

タイムアウトとなったトランザクションが排他待ちとなっている資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

●タイムアウトの要因となったトランザクションを送信したサーバの情報

（14）～（21），（28）～（29）の情報は，タイムアウトの要因となったトランザクションが占有した資源の情報です。複数出力される場合があります。

（14）UAP 識別情報

タイムアウトの要因となったトランザクションを送信したサーバに接続している UAP の識別情報を示します。トランザクションを送信したのがユティリティの場合，ユティリティ名を示します。

ここで表示する情報は pdls -d prc 又は pdls -d trn コマンドで表示される PROGRAM に該当します。ただし，一部のユティリティでは表示できません。表示できない場合は"********"を表示します。また，再開始処理で回復中のトランザクションの場合は"Rerun"が表示されます。

（15）サーバ名

タイムアウトの要因となったトランザクションを送信したサーバのサーバ名を示します。

（16）プロセス ID

タイムアウトの要因となったトランザクションを送信したサーバプロセスのプロセス ID を示します。

（17）トランザクション識別子

タイムアウトの要因となったトランザクションの識別子を示します。

（28）ユーザ識別子

HiRDB に接続しているユーザを一意に識別するために，HiRDB が動的に割り当てる通番を示します。なお，ユーザ識別子を使って SQL 文を特定できる場合があります。マニュアル「HiRDB コマンドリファレンス」の「pdobils（SQL オブジェクト用バッファの統計情報表示）」を参照してください。

（29）デッドロックプライオリティ値

トランザクションのデッドロックプライオリティ値を示します。

●タイムアウトの要因となったトランザクションが占有したすべての資源の情報

（18）～（21）の情報は，タイムアウトの要因となったすべての資源の情報です。このトランザクションが資源を占有していなかった場合，各情報には空白が出力されます。

（18）サーバ名

タイムアウトの要因となったトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（19）排他制御モード

タイムアウトの要因となったトランザクションが占有した資源に掛けた排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（20）占有資源種別

タイムアウトの要因となったトランザクションが占有した資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（21）占有資源情報

タイムアウトの要因となったトランザクションが占有した資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

●タイムアウトとなったトランザクションと同じ資源で排他待ちになっているトランザクションを送信したサーバの情報

（30）～（39）の情報は，タイムアウトとなったトランザクションと同じ資源で排他待ちになっているトランザクションの情報です。複数出力される場合があります。

（30）UAP 識別情報

排他待ちになっているトランザクションを送信したサーバに接続している UAP の識別情報を示します。トランザクションを送信したのがユティリティの場合，ユティリティ名を示します。

ここで表示する情報は pdls -d prc 又は pdls -d trn コマンドで表示される PROGRAM に該当します。ただし，一部のユティリティでは表示できません。表示できない場合は"********"を表示します。また，再開始処理で回復中のトランザクションの場合は"Rerun"が表示されます。

（31）サーバ名

排他待ちになっているトランザクションを送信したサーバのサーバ名を示します。

（32）プロセス ID

排他待ちになっているトランザクションを送信したサーバプロセスのプロセス ID を示します。

（33）トランザクション識別子

排他待ちになっているトランザクションの識別子を示します。

（34）ユーザ識別子

HiRDB に接続しているユーザを一意に識別するために，HiRDB が動的に割り当てる通番を示します。

（35）デッドロックプライオリティ値

トランザクションのデッドロックプライオリティ値を示します。

●タイムアウトとなったトランザクションと同じ資源で排他待ちになっているトランザクションが排他待ちとなった資源の情報

（36）～（39）の情報は，タイムアウトとなったトランザクションと同じ資源で排他待ちになっているトランザクションの資源の情報です。

（36）サーバ名

排他待ちになっているトランザクションが資源の占有を要求したサーバのサーバ名を示します。

（37）排他制御モード

排他待ちになっているトランザクションが排他待ちとなっている資源に掛けようとした排他制御モードを示します。排他制御モードについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

（38）待ち資源種別

排他待ちになっているトランザクションが排他待ちとなっている資源の種別を示します。資源種別については，「資源種別と資源情報」を参照してください。

（39）待ち資源情報

排他待ちになっているトランザクションが排他待ちとなっている資源の情報を示します。資源情報については，「資源種別と資源情報」を参照してください。

タイムアウト情報の出力例を次の図に示します。

図 8–10　タイムアウト情報の出力例

```
Timeout information                                    Jun  2 06:12:43: 2000

program : SPPY415
server : SDS     pid : 4510
trnbid : q192u19200000001     actid : 1-1-5     dprio : 64
 occupy
  server : SDS   lock mode : PR    kind : 3001
  resource info : 7c000200000000000000000000
  server : SDS   lock mode : SR    kind : 0001
  resource info : 030000000000000000000000000
  server : SDS   lock mode : SR    kind : 0002
  resource info : 080002000000000000000000000
  server : SDS   lock mode : SU    kind : 0001
  resource info : 050000000000000000000000000
  server : SDS   lock mode : SU    kind : 0002
  resource info : 7c000200000000000000000000
  server : SDS   lock mode : EX    kind : 0007
  resource info : 050000001b000002790002000000
 wait
  server : SDS   lock mode : EX    kind : 0007
  resource info : 050000001b0000027c0002000000

program : SPPE201
server : SDS     pid : 4481
trnbid : q192u19200000003     actid : 1-1-4     dprio : 64
 occupy
  server : SDS   lock mode : EX    kind : 0007
  resource info : 050000001b0000027c0002000000
```

8.6.4　資源種別と資源情報

デッドロック情報又はタイムアウト情報に出力される資源種別と資源情報を次の表に示します。

表 8–7　資源種別と資源情報

資源種別	種別名	資源名称 1※1	資源名称 2※1	資源情報	内容
0001	RDAR	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	RD エリア

資源種別	種別名	資源名称 1※1	資源名称 2※1	資源情報	内容
0002	TABL	表名又は順序数生成子名	−	1〜8 けた：表番号又は順序数生成子番号 9〜28 けた：00 固定	表又は順序数生成子
0003	INDX	インデクス名	RD エリア名	1〜8 けた：インデクス番号 9〜16 けた：RD エリア番号 17〜28 けた：00 固定	インデクス
0004	PAGE	表名又はインデクス名	RD エリア名	1〜2 けた：ファイル番号 3〜8 けた：RD エリア番号 9〜16 けた：ページ番号 17〜24 けた：表番号又はインデクス番号 25〜28 けた：00 固定	ページ
0007	ROW	表名	RD エリア名	1〜2 けた：ファイル番号 3〜8 けた：RD エリア番号 9〜10 けた：スロット番号 11〜16 けた：ページ番号 17〜24 けた：表番号 25〜28 けた：00 固定	行
0008	NWROW	表名	RD エリア名	1〜2 けた：ファイル番号 3〜8 けた：RD エリア番号 9〜10 けた：スロット番号 11〜16 けた：ページ番号 17〜24 けた：表番号 25〜28 けた：00 固定	行（WITHOUT LOCK NOWAIT 検索制御用）
000B	TABN	表名	−	1〜8 けた：表番号 9〜28 けた：00 固定	表（NOWAIT 検索時）
000C	NKEY	インデクス名	−	1〜8 けた：インデクス番号 9〜28 けた：00 固定	キー値（NULL）
000D	DKEY	インデクス名	−	1〜8 けた：インデクス番号 9〜28 けた：キー値又はエンコードされたキー値	キー値（非 NULL）
000E	LFID	RD エリア名	論理ファイル番号	1〜8 けた：RD エリア番号 9〜16 けた：論理ファイル番号 17〜28 けた：00 固定	論理ファイル
000F	IXIF	インデクス名	RD エリア名	1〜8 けた：インデクス番号 9〜16 けた：RD エリア番号 17〜28 けた：00 固定	インデクス情報ファイル

資源種別	種別名	資源名称 1※1	資源名称 2※1	資源情報	内容
0010	TRWT	表名	インデクス格納 RD エリア名	1〜8 けた：表番号 9〜16 けた：インデクス格納 RD エリア番号 17〜28 けた：00 固定	トランザクション決着待ち
0011	SHWT	表名 （RD エリア更新完了待ちの場合は，"****"を出力します）	RD エリア名	1〜2 けた：ファイル番号 3〜8 けた：RD エリア番号 9〜10 けた：スロット番号 11〜16 けた：ページ番号 17〜24 けた：表番号 25〜28 けた：00 固定 RD エリア更新完了待ちの場合は，RD エリア番号以外は 0 となります。	共用 RD エリアトランザクション決着待ち
0012	UCWT	−	−	1〜16 けた：トランザクション情報 17〜28 けた：00 固定	ユニークチェックのトランザクション待ち
0102	RRAMB	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	RD エリア管理情報
0111	MFCB	−	−	1〜8 けた：ページ番号 9〜12 けた：ファイル番号 13〜28 けた：00 固定	マスタディレクトリセグメント情報
0112	MTCB	表名	−	1〜8 けた：表番号 9〜28 けた：00 固定	マスタディレクトリ表情報
0113	MICB	インデクス名	−	1〜8 けた：インデクス番号 9〜28 けた：00 固定	マスタディレクトリインデクス情報
0121	RATM	表名	RD エリア名	1〜8 けた：RD エリア番号 9〜16 けた：表番号又は表管理番号 17〜28 けた：00 固定	ディクショナリ表情報又はユーザディレクトリ表情報
0122	RAIM	RD エリア名	−	1〜8 けた：RD エリア番号 9〜16 けた：インデクス番号又はインデクス管理番号 17〜28 けた：00 固定	ディクショナリインデクス情報又はユーザディレクトリインデクス情報
0132	SBMB	RD エリア名	−	1〜8 けた：RD エリア番号 9〜16 けた：セグメント番号 17〜20 けた：ファイル番号 21〜28 けた：00 固定	ディクショナリセグメント情報又はユーザディレクトリセグメント情報
0143	RDLF	RD エリア名	最終ファイル番号	1〜8 けた：RD エリア番号 9〜12 けた：最終ファイル番号 13〜28 けた：00 固定	RD エリア増分

資源種別	種別名	資源名称 1※1	資源名称 2※1	資源情報	内容
0152	SGMB	RD エリア名	−	1〜8 けた：RD エリア番号 9〜16 けた：ページ番号 17〜20 けた：ファイル番号 21〜28 けた：00 固定	データベース複写ユティリティ
0300	MENT	RD エリア名	−	1〜8 けた：RD エリア番号 9〜16 けた：ページ番号 17〜20 けた：ファイル番号 21〜24 けた：エントリ番号 25〜28 けた：00 固定	ユーザ LOB 用 RD エリアの管理情報
0301	LOBID	RD エリア名	−	1〜8 けた：RD エリア番号 9〜16 けた：LOB 番号 17〜28 けた：00 固定	ユーザ LOB 用 RD エリアの管理情報
0502	TEMP	サーバ名	−	先頭から 8 バイトがサーバ名を示しています。サーバ名が 8 バイトに満たない場合は，NULL を埋めます。	一時表実体化管理情報
0503	TPID	−	−	1〜8 けた：ユーザ固有番号 9〜28 けた：00 固定	ユーザ固有 ID 管理（一時表）
0601	RDAS	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	データベース状態解析ユティリティ
0602	HOLD	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	バックアップ閉塞 RD エリア
0605	MDBS	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	インメモリデータ処理の同期情報
0606	RDAH	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	pdhold コマンド受付状態
0900	PLGR	−	−	1〜8 けた：プラグイン ID 9〜28 けた：プラグインの独自資源番号	プラグイン資源番号
0D00 〜 0FFF	UNCH	資源種別	−	1〜8 けた：内部情報 9〜16 けた：インデクス ID 17〜28 けた：分割したインデクスキー値	ユニークチェック
2002	ROMB	−	−	1〜14 けた：SQL オブジェクト番号 15〜28 けた：00 固定	SQL オブジェクト管理情報
2003	SPCH	−	−	'SPCH'固定 余った末尾には 0 を埋めます。	SQL オブジェクトキャッシュ
2004	TRAL	"****"	"****"	'TMPRDAREA'固定 余った末尾には 0 を埋めます。	一時表用 RD エリア管理情報

資源種別	種別名	資源名称 1※1	資源名称 2※1	資源情報	内容
2005	TRAI	"****"	"****"	1〜8 けた：一時表用 RD エリア管理ブロックの番号 9〜28 けた：00 固定	一時表用 RD エリア情報
3001	PTBL	"****"	"****"	1〜8 けた：表番号 9〜28 けた：00 固定	前処理表
3005	DICT	"****"	"****"	'DICT'固定 余った末尾には 0 を埋めます。	ディクショナリ表
3006	VIEW	"****"	"****"	1〜8 けた：ビュー表番号 9〜28 けた：00 固定	ビュー表
3008	TBPL	"****"	"****"	先頭から 5 バイト※2が認可識別子を示しています。 6〜14 バイト※3が表識別子を示しています。	表定義情報バッファ
3010	AUTH	"****"	"****"	先頭から 14 バイトが認可識別子を示しています。	ユーザ権限情報バッファ
3011	OBJI	"****"	"****"	ルーチンオブジェクト番号	ルーチンオブジェクト
3012	DTYP	"****"	"****"	'DTYP'固定 余った末尾には 0 を埋めます。	データ型定義情報
3013	RTPL	"****"	"****"	先頭から 12 バイトがルーチン識別子を示しています。 13〜14 バイトがパラメタ数を示しています。	ルーチン定義情報用バッファ
3014	TPPL	"****"	"****"	先頭から 5 バイト※2が認可識別子を示しています。 6〜14 バイト※3がデータ型識別子を示しています。	ユーザ定義型情報バッファ
3015	DICR	"****"	"****"	'DICR'固定	ディクショナリ用 RD エリア変更
3016	CONS	"****"	"****"	'CONSEC'固定	CONNECT 関連セキュリティ定義情報
3017	SQID	"****"	"****"	1〜8 けた：順序数生成子番号※4 9〜28 けた：00 固定	順序数生成子
3018	SQPL	"****"	"****"	先頭から 5 バイト※2が認可識別子を示しています。 6〜14 バイト※3が順序数生成子識別子を示しています。	順序数生成子定義情報

資源種別	種別名	資源名称 1※1	資源名称 2※1	資源情報	内容
5001	DICU	−	−	'DICTMODUTL'固定 余った末尾には 0 を埋めます。	データベース構成変更ユティリティ
5002	LCBF	RD エリア名	−	1〜8 けた：RD エリア番号 9〜28 けた：00 固定	−
5003	EXP1	−	−	'EXPIMPMDL'固定 余った末尾には 0 を埋めます。	データベース搬出入ユティリティ
5004	RBAL	表名	−	1〜8 けた：表番号 9〜28 けた：00 固定	リバランスユティリティ
5005	RRAMB	サーバ名	−	1〜16 けた：サーバ名 17〜28 けた：00 固定	ZRRAMB 更新
5006	RCLM	RD エリア名	表名又はインデクス名	1〜8 けた：RD エリア番号 9〜16 けた：表番号又はインデクス番号 17〜28 けた：00 固定	空きページ解放ユティリティ

（凡例）

−：該当しません。

注

- 資源種別は 16 進形式（4 けた）で出力されます。
- 資源情報は 16 進形式（28 けた）で出力されます。資源情報の見方については，「資源情報の見方」を参照してください。

注※1

資源名称 1 及び資源名称 2 の表名，順序数生成子名，又は RD エリア名を表示できない場合，"****"を表示します。その場合，資源情報の表番号，順序数生成子番号，又は RD エリア番号から対象資源を特定してください。

注※2

６バイト以上の認可識別子は次の形式で出力されます。

認可識別子の先頭３バイト＋認可識別子の後ろ２バイト

例えば，認可識別子が k87m341 の場合は，"k8741"となります。

なお，情報は ASCII コードで，１バイトが２けたで出力されます。

注※3

10 バイト以上の表識別子，データ型識別子，及び順序数生成子識別子は次の形式で出力されます。

表識別子，データ型識別子，又は順序数生成子識別子の先頭 5 バイト＋表識別子，データ型識別子，又は順序数生成子識別子の後ろ 4 バイト

例えば，表識別子が TABLE002498 の場合は，"TABLE2498"となります。

なお，情報は ASCII コードで，１バイトが２けたで出力されます。

注※4

順序数生成子番号は，ディクショナリ表 SQL_SEQUENCES の SEQUENCE_ID 列を参照してください。

8.6.5　資源情報の見方

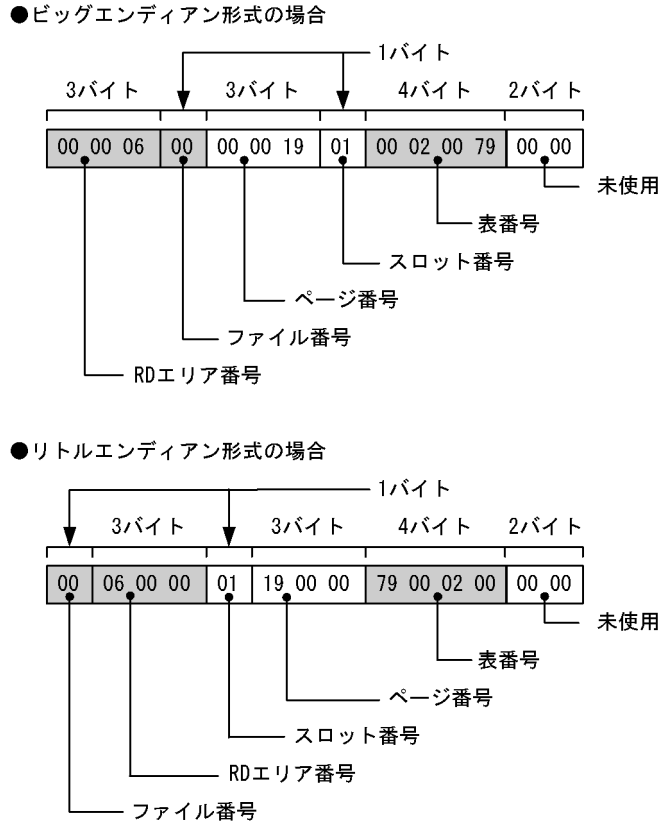
資源情報は 16 進形式（28 けた）で出力されます。複数記載されている資源情報は，記載順に連続して出力されます。資源情報が 28 けたに満たない場合，余った末尾には 0 を埋めます。キャラクタ文字はすべて ASCII コードで，1 バイトが 2 けたで出力されます。

注意事項

資源情報はリトルエンディアン形式で出力されます。

資源情報の出力例（資源種別 0007 の場合）を次の図に示します。

図 8–11　資源情報の出力例（資源種別 0007 の場合）

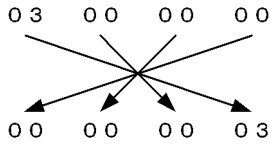


(1)　RD エリア番号から RD エリア名を調べる方法

資源情報に出力された RD エリア番号から RD エリア名を調べる手順を次に示します。

〈手順〉

1. リトルエンディアン形式をビッグエンディアン形式に変換します。RD エリア番号に **03000000** が出力されたとします。この場合，次のようにビッグエンディアン形式に変換します。



   RD エリア番号は **3** になります。

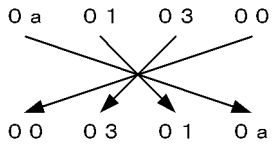
2. 資源情報に出力された RD エリア番号は 16 進形式のため，10 進形式に変換してください。
3. pddbls コマンドで，RD エリア番号に対応する RD エリア名を調べてください。

(2) インデクス番号からインデクス名を調べる方法

資源情報に出力されたインデクス番号からインデクス名を調べる手順を次に示します。

〈手順〉

1. リトルエンディアン形式をビッグエンディアン形式に変換します。インデクス番号に **0a010300** が出力されたとします。この場合，次のようにビッグエンディアン形式に変換します。



   インデクス番号は **3010a** になります。

2. 資源情報に出力されたインデクス番号は 16 進形式のため，10 進形式に変換してください。
3. ディクショナリ表 SQL_INDEXES の INDEX_ID 列を検索して，インデクス番号に対応するインデクス名を調べてください。

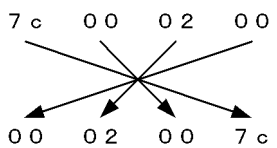
   ディクショナリ表 SQL_INDEXES の検索方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(3) 表番号（ビュー表番号）から表名（ビュー表名）を調べる方法

資源情報に出力された表番号（ビュー表番号）から表名（ビュー表名）を調べる手順を次に示します。

〈手順〉

1. リトルエンディアン形式をビッグエンディアン形式に変換します。表番号に **7c000200** が出力されたとします。この場合，次のようにビッグエンディアン形式に変換します。



表番号は **2007c** になります。

2. 資源情報に出力された表番号（ビュー表番号）は 16 進形式のため，10 進形式に変換してください。
3. ディクショナリ表 SQL_TABLES の TABLE_ID 列を検索して，表番号（ビュー表番号）に対応する表名（ビュー表名）を調べてください。

   ディクショナリ表 SQL_TABLES の検索方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

8.6.6　デッドロック SQL 情報の出力内容

デッドロック SQL 情報は pdobils コマンドの使用中情報出力形式で出力します。出力形式の詳細はマニュアル「HiRDB コマンドリファレンス」を参照してください。

注

- デッドロックに関連するトランザクションの SQL 情報を出力します。
- 最大 64 トランザクションの情報を出力します。

  デッドロックに関連するトランザクションの数が 64 を超える場合は，デッドロック SQL 情報の末尾に下記に示す情報を出力します。

  ```
  (Number of deadlock transactions = トランザクション数(1) exceed output limit = 64)
  ```

  (1)デッドロックに関連するトランザクション数

- PDCWAITTIME，PDLCKWAITTIME に指定した時間を超過するなどでエラーになったトランザクションの情報は出力しません。
- HiRDB パラレルサーバで，サーバ間でデッドロックが発生した場合は出力しません。
- クライアント環境定義 PDVWOPTMODE 又は PDUAPREPLVL で，SQL オブジェクトの作成を行う指定をした場合など，SQL 情報が出力されないときがあります。詳細については，マニュアル「HiRDB コマンドリファレンス」の「pdobils（SQL オブジェクト用バッファの統計情報表示）」の「オペランド」の「-A」を参照してください。

8.6.7　デッドロック情報とデッドロック SQL 情報を突き合わせる方法

デッドロックが発生した場合は，デッドロック情報に出力されている排他資源の情報とデッドロック SQL 情報に出力されている SQL の情報からデッドロックの原因を特定して対処してください。

対応するデッドロック情報とデッドロック SQL 情報を付き合わせる方法を以下に示します。

(1)　対応するファイルを特定します

デッドロックが発生した日時を基に，対応するデッドロック・タイムアウト情報ファイルとデッドロック SQL 情報ファイルを特定します。

（例）
デッドロックが発生した日時が10月3日6時29分56秒のとき
　デッドロック情報ファイル　　：**Oct3062956**
　デッドロックSQL情報ファイル：**Oct3062956**_bes1_SYS1UNT100020000_1.sql
　　日時が同じファイルを参照する

(2)　対応するトランザクションを特定します

ユーザ識別子を基に，対応するデッドロック情報のトランザクションとデッドロック SQL 情報のトランザクションを特定します。

ユーザ識別子は，デッドロック情報では actid，デッドロック SQL 情報では ACTID で示しています。

```
 （デッドロック情報）

Deadlock information                                    Oct 18 15:01:07 2023

 program:SPPY415
 server:bes1     pid:29754
 trnbid:SYS1UNT100020000 actid:3-1-10 dprio:64     ・・・ユーザ識別子①
  occupy
   server:bes1     lock mode:EX    kind:0007
   resource info:0004000003e100007a0002000000
  wait
   server:bes1     lock mode:EX    kind:0007
   resource info:000400006719000079000200000
  wait start time 15:01:07
 program:SPPE201
 server:bes1     pid:29755
 trnbid:SYS1UNT100020001 actid:3-1-9 dprio:64     ・・・・ユーザ識別子②
  occupy
   server:bes1     lock mode:EX    kind:0007
   resource info:000400006719000079000200000
  wait
   server:bes1     lock mode:EX    kind:0007
   resource info:0004000003e100007a0002000000
  wait start time 15:00:44
```

```
 （デッドロックSQL情報）

pdobils 10-08
<< SQL OBJECT CACHE INFORMATION >> DATE: 2023/10/18 TIME: 15:00:51
HOST NAME                  : node01
SERVER NAME                : fes
CACHE SIZE(KB)             : 520
SQL OBJECT TOTAL SIZE(B)   : 10560

<< UAP INFORMATION >>
 PROCESS ID                : 29730
 TRANSACTION ID(GID/BID)   : SYS1UNT102000017/SYS1UNT100020016
 SERVICE NAME              :
 UAP NAME                  : SPPE201
 CONNECT NO                : 10
 ACTID                     : 3-1-10     ・・・ユーザ識別子①

<< USING SQL OBJECT LIST >>     ・・・・ユーザ識別子①のトランザクションの情報
*SQL OBJECT NO             : 5
 STATUS                    : ACTIVE
 TYPE                      : STATIC SQL
 SIZE(B)                   : 3520
 EXECUTE COUNT             : 0
 PREPROCESSOR USER         : hiruser
```

```
 PREPROCESSOR SOURCE        : SPPE201-30668-10
 SECTION NO                 : 1
 ISOLATION LEVEL            : 2
 OPTIMIZE LEVEL             : 2756512
 ADDITIONAL OPTIMIZE LEVEL  : 12385
 ACCESS TYPE SSCT# 2        : hiruser.T2 (T:0x0002007a) KEY SCAN (I:0x000300f2)
 SQL                        : update t2 set c2=2002 where c2=1002

<< UAP INFORMATION >>
 PROCESS ID                 : 29729
 TRANSACTION ID(GID/BID)    : SYS1UNT102000016/SYS1UNT100020015
 SERVICE NAME               :
 UAP NAME                   : SPPY415
 CONNECT NO                 : 9
 ACTID                      : 3-1-9     ・・・・ユーザ識別子②

<< USING SQL OBJECT LIST >>     ・・・・ユーザ識別子②のトランザクションの情報
*SQL OBJECT NO              : 2
 STATUS                     : ACTIVE
 TYPE                       : STATIC SQL
 SIZE(B)                    : 3520
 EXECUTE COUNT              : 0
 PREPROCESSOR USER          : hiruser
 PREPROCESSOR SOURCE        : SPPY415-30663-9
 SECTION NO                 : 1
 ISOLATION LEVEL            : 2
 OPTIMIZE LEVEL             : 2756512
 ADDITIONAL OPTIMIZE LEVEL  : 12385
 ACCESS TYPE SSCT# 2        : hiruser.T1 (T:0x00020079) KEY SCAN (I:0x000300f1)
 SQL                        : update t1 set c1=2001 where c1=1001

*SQL OBJECT NO              : 4
 STATUS                     : ACTIVE
 TYPE                       : STATIC SQL
 SIZE(B)                    : 3520
 EXECUTE COUNT              : 0
 PREPROCESSOR USER          : hiruser
 PREPROCESSOR SOURCE        : SPPY415-30663-9
 SECTION NO                 : 2
 ISOLATION LEVEL            : 2
 OPTIMIZE LEVEL             : 2756512
 ADDITIONAL OPTIMIZE LEVEL  : 12385
 ACCESS TYPE SSCT# 2        : hiruser.T2 (T:0x0002007a) KEY SCAN (I:0x000300f2)
 SQL                        : update t2 set c2=3002 where c2=2002

<< SQL OBJECT STATUS COUNT >>
STATUS   | TYPE    STATIC SQL     DYNAMIC SQL    ROUTINE        TOTAL
ACTIVE           : 3              0              0              3
LRU              : 0              0              0              0
TEMPORARY        : 0              0              0              0
COMPILE/TRANSFER : 0              0              0              0
PROCESS          : -              -              -              0
RELEASE          : -              -              -              0
TOTAL            : 3              0              0              3
```

8.7　排他資源管理テーブル不足が発生した場合

8.7.1　排他資源管理テーブル不足が発生した場合の対処方法

実行者　HiRDB 管理者

排他資源管理テーブル不足が発生すると，HiRDB は **KFPS00443-I** メッセージと，排他資源管理テーブル情報を出力します。排他資源管理テーブル情報は，排他資源管理テーブル不足が発生したユニットのディレクトリ（%PDDIR%¥spool¥pdlckinf）に出力されます。HiRDB 管理者はその情報を参照して，排他資源管理テーブル不足解消のための対策を立ててください。

注　排他制御処理の分散を適用している場合は，「排他資源管理テーブル」を「排他制御用プールパーティション」に読み替えてください。

(1)　排他資源管理テーブル情報のファイル名は

排他資源管理テーブル情報は，排他資源管理テーブル不足が発生するたびにファイルに出力されます。出力ファイル名は，「出力日時.mem」となります。mem はファイル記述子です。例えば，10 月 3 日 9 時 16 分 2 秒に，排他資源管理テーブル不足が発生すると，その出力ファイル名は Oct3091602.mem となります。ファイル名は，**KFPS00447-I** メッセージに表示されます。

(2)　不要な排他資源管理テーブル情報ファイルを削除しましょう

HiRDB 管理者は不要となった排他資源管理テーブル情報ファイルを次に示す(a)，(b)又は(c)の方法で削除してください。

(a)　HiRDB のコマンドで削除する方法

pdcspool コマンドで削除します。pdcspool コマンドを実行すると，%PDDIR%¥spool 下にあるトラブルシュート情報ファイルが削除されます。したがって，排他資源管理テーブル情報ファイルだけを削除する場合は，「OS の機能で削除する方法」を参照してください。

(b)　HiRDB の機能で削除する方法

pd_spool_cleanup_interval_level オペランドを省略するか all を指定すると，定期的に HiRDB が排他資源管理テーブル情報ファイルを削除します。通常は 24 時間ごとに削除します。この削除間隔を pd_spool_cleanup_interval オペランドで変更できます。

また，pd_spool_cleanup_level オペランドを省略するか all を指定すると，HiRDB の開始時に HiRDB が排他資源管理テーブル情報ファイルを削除します。

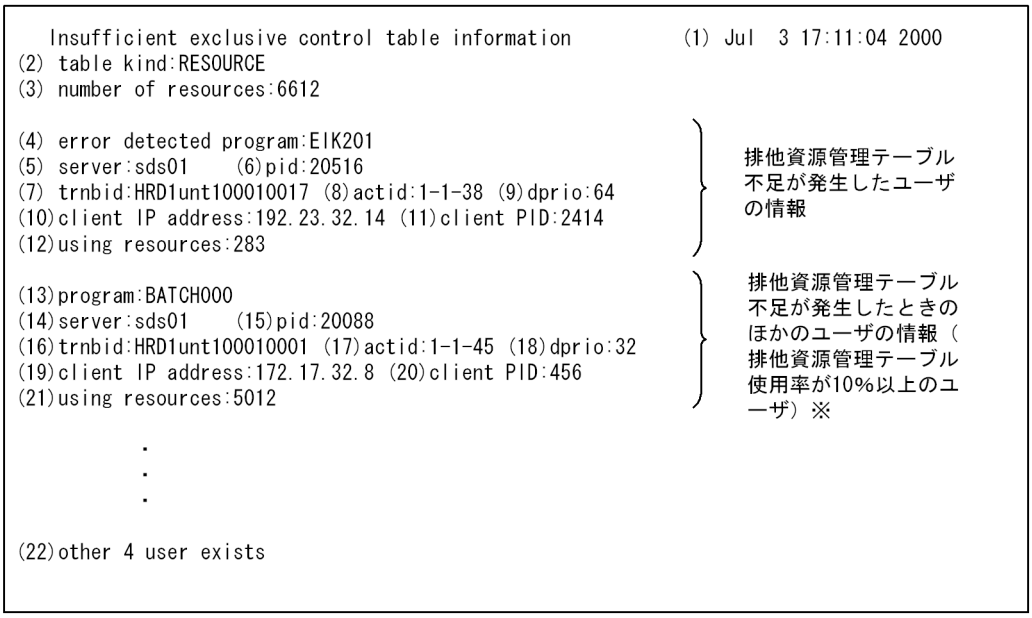
(c)　OS の機能で削除する方法

OS の機能（OS の del コマンドなど）で削除してください。del コマンドについては，OS のマニュアルを参照してください。

(3)　排他資源管理テーブル情報の出力内容

排他資源管理テーブル情報の出力内容を次の図に示します。

図 8–12　排他資源管理テーブル情報の出力内容



注※　対象ユーザが複数ある場合は，その数の分だけ情報が表示されます。

〔説明〕

（1）排他資源管理テーブル不足の検知日時（月　日　時:分:秒 西暦）

（2）不足した排他資源管理テーブルの種別（次のどちらかを出力します）

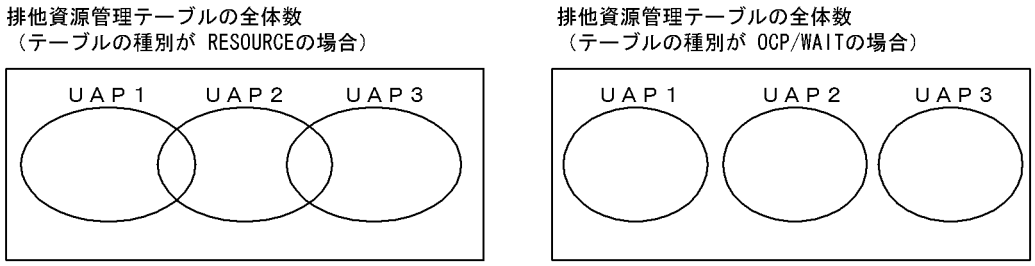
- **RESOURCE**

  使用する資源名称を管理するときに使用するテーブルです。複数ユーザで共有します。そのため，全ユーザの使用しているテーブルの合計数がテーブルの総数を超えることがあります。

- **OCP/WAIT**

  共有，待ちの状態を管理するときに使用するテーブルです。複数ユーザで共有しません。

RESOURCE と OCP/WAIT の違いを次に示します。



（3）使用できる排他資源管理テーブルの総数

排他資源管理テーブル不足が発生したユーザの情報

（4）UAP 識別情報 ※1

（5）サーバ名

（6）プロセス ID

（7）トランザクション識別子

（8）ユーザ識別通番

（9）デッドロックプライオリティ値

（10）クライアントの IP アドレス ※3

（11）クライアントのプロセス ID ※2，※3

（12）現在使用しているテーブル数

排他資源管理テーブル不足が発生したときのほかのユーザの情報（排他資源管理テーブル使用率が 10%以上のユーザ）

（13）UAP 識別情報 ※1

（14）サーバ名

（15）プロセス ID

（16）トランザクション識別子

（17）ユーザ識別通番

（18）デッドロックプライオリティ値

（19）クライアントの IP アドレス ※3

（20）クライアントのプロセス ID ※2，※3

（21）現在使用しているテーブル数

排他資源管理テーブル不足が発生したときのほかのユーザの情報（排他資源管理テーブル使用率が 10%未満のユーザ）

（22）排他資源管理テーブルの使用率が 10%に満たないユーザの数

注※1

CONNECT していたクライアント UAP の名称を 30 バイトで表示します。

ここで表示する情報は pdls -d prc や pdls -d trn で出力される PROGRAM に該当します。ただし，一部のユティリティは表示できません。この場合，「********」が表示されます。また，再開始処理でトランザクションを回復中のときは”Rerun”を表示します。

注※2

CONNECT していたクライアント UAP がリンケージしているクライアントライブラリのバージョンが HiRDB Version 4.0 04-00 より前の場合，及び Type4 JDBC ドライバから接続している場合は，プロセス ID に 0 が表示されます。

注※3

バックエンドサーバ及びディクショナリサーバでは，表示できないことがあります。表示できない場合，クライアントの IP アドレスは「*.*.*.*」，プロセス ID は 0 を表示します。

(4) 調査方法

排他資源管理テーブル情報の次の情報を参照します。

- （3）使用できる排他資源管理テーブルの総数
- （21）現在使用しているテーブル数

この情報から，どの UAP がどれだけの排他資源管理テーブルを使用しているかが分かります。排他資源管理テーブルの使用数は，UAP が行う排他要求数と一致します。したがって，「（21）現在使用しているテーブル数」の値が大きい UAP は，不当に多くの排他要求をしている可能性があります。UAP が行う排他要求数は SQL によって異なります。各 SQL の排他要求数（排他資源数の見積もり）については，マニュアル「HiRDB システム定義」を参照してください。

(5) 対策方法

(a) 不当に多くの排他要求を行っている UAP がある場合

多くの排他要求をしないように UAP を修正してください。

また，クライアント環境定義の PDLOCKLIMIT オペランドを指定すると，1 UAP の排他要求数の上限を設定できます。

(b) 不当に多くの排他要求を行っている UAP がない場合

排他資源管理テーブル数が少な過ぎると思われます。次に示すどちらかの対策をとってください。

●ユニットで使用する共用メモリに余裕がない場合（共用メモリを大きくできない場合）

排他資源管理テーブル数をたくさん使用する UAP を同時に実行しないでください。

●ユニットで使用する共用メモリに余裕がある場合（共用メモリを大きくできる場合）

HiRDB システム定義を修正します。修正するオペランドはサーバの種類によって異なります。排他資源管理テーブルが不足したサーバは，排他資源管理テーブル情報の「（5）サーバ名」を参照すれば分かります。

排他資源管理テーブルが不足したサーバがフロントエンドサーバの場合は，エラーが発生したフロントエンドサーバ定義の pd_fes_lck_pool_size オペランドの値を大きくしてください。

排他資源管理テーブルが不足したサーバがフロントエンドサーバ以外の場合は，エラーが発生したサーバ定義の pd_lck_pool_size オペランドの値を大きくしてください。

注意事項

HiRDB システム定義を変更する場合は，HiRDB 全体又は変更する定義があるユニットを正常終了させてください。その後，HiRDB システム定義を変更してください。pdstop -s で該当するサーバだけを正常終了させても変更した定義は有効になりません。

8.8　UAP の状態監視（シンクポイントダンプ有効化のスキップ回数監視機能）

8.8.1　シンクポイントダンプ有効化のスキップ回数監視機能の概要

UAP が無限ループしてデータベースを更新し続けるとシンクポイントが有効化できないため，上書きできない状態のシステムログファイルが増えてしまいます。上書きできない状態のシステムログファイルが増えて全システムログファイルが上書きできない状態になると，HiRDB が異常終了します。

また，上書きできない状態のシステムログファイルが，全システムログファイルの 1/3 以上になったときに HiRDB が異常終了又は強制終了すると，HiRDB を再開始するときのロールバック処理でシステムログファイルが不足する場合があります。この場合，システムログファイルを新規追加しないと，HiRDB を再開始できません。そして，この再開始処理に要する時間も長くなります。

このようなことを防ぐために，HiRDB では**シンクポイントダンプ有効化のスキップ回数監視機能**を設けています。

(1)　シンクポイントダンプ有効化のスキップ回数監視機能とは

UAP の無限ループなどが発生すると，シンクポイントダンプの有効化処理が連続してできないことがあります（シンクポイントダンプの有効化処理が連続してスキップされることがあります）。このスキップ回数がある一定の回数に達した場合，対象トランザクションを強制的に中断してロールバック処理を行います。これをシンクポイントダンプ有効化のスキップ回数監視機能といいます。この機能を使用する場合は **pd_spd_syncpoint_skip_limit** オペランドを指定します。

(2)　pd_spd_syncpoint_skip_limit オペランドに指定する値

通常は pd_spd_syncpoint_skip_limit オペランドに 0 を指定してください。0 を指定すると，スキップ回数の上限値を HiRDB が自動計算します。0 を指定して不都合が発生した場合は，(3)に示すどちらかの方法でこのオペランドの値を計算してください。

次に示す条件を一つでも満たす場合は，自動計算よりも(3)の方法で計算した方が，精度が高くなります。

- 現用ファイルとして使用可能なシステムログファイルが 5 世代以下である
- 処理時間が掛かるトランザクションを並列実行する
- 反映側 HiRDB のデータ反映トランザクション処理に時間が掛かる（HiRDB Datareplicator との連携時）

KFPS02101-I メッセージが出力される場合は，自動計算をやめて(3)の方法で計算してください。

次に示す場合は，自動計算で算出されたスキップ回数の上限値で運用を続行するかどうか検討してください。自動計算をやめる場合は，「出力されるシステムログ量から計算する方法」に示す方法で計算してください。

- 上書きできない状態のシステムログファイルのシステムログ量（HiRDB を再開始するときに読み込むシステムログ量）が，スキップしている間に増加して，HiRDB の再開始処理に要する時間が長くなる場合
  なお，この場合，HiRDB の再開始処理時間を短縮するために，補正してから自動計算します（KFPS02103-I メッセージが出力されます）。
- 自動計算で算出されたスキップ回数の上限値が 100,000 回を超える場合
  なお，この場合，100,000 回を仮定します（KFPS02103-I メッセージが出力されます）。

(3) 計算方法

計算方法には次に示す二つの方法があります。どちらかの方法で計算してください。

- 出力されるシステムログ量から計算する方法
- 全システムログファイルの容量から計算する方法

ここで求めた値より多少小さい値を pd_spd_syncpoint_skip_limit オペランドに指定してください。

(a) 出力されるシステムログ量から計算する方法

次に示す計算式から求めます。

〈計算式〉

{(↑a÷b↑÷c）÷d}－1

a：

データベースの更新量が最も多いトランザクション処理で出力されるシステムログ量と，このトランザクションが稼働中に同時実行するトランザクション処理で出力されるシステムログ量の合計です（単位：バイト）。システムログ量の求め方については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

●HiRDB Datareplicator を使用している場合

反映側 HiRDB のデータ反映トランザクション処理に時間が掛かる場合，この機能によって反映側 HiRDB がデータ反映トランザクションをロールバックすることがあります。したがって，データ反映トランザクション処理で出力されるシステムログ量を加算する必要があります。次に示す計算式で求めた値を加算してください。

データ反映トランザクション処理で出力されるシステムログ量

＝Σ（データベースの更新量が最も多いトランザクション処理で出力されるシステムログ量）

Σ は HiRDB Datareplicator の反映環境定義の cmtintvl（trncmtintvl，tblcmtintvl）オペランドに指定したトランザクション数分の合計を意味しています。

b：

システムログファイルのレコード長です。レコード長は pdlogls コマンドで調べられます。

c：

システムログブロックの平均レコード数です。通常，3×4096÷b 程度を目安にしてください。なお，次に示す計算式から正確に求められます。

↑システムログ出力ブロック長の平均値÷ｂ↑

システムログ出力ブロック長の平均値は，統計解析ユティリティのシステムの稼働に関する統計情報（OUTPUT BLOCK LENGTH）から分かります。

ｄ：

pd_log_sdinterval オペランドの第１パラメタ（シンクポイントダンプの取得間隔をシステムログの出力量で指定）の値です。

(b)　全システムログファイルの容量から計算する方法

次に示す計算式から求めます。

〈計算式〉

{（ａ×ｂ×ｃ）÷ｄ}　÷ｅ

ａ：

HiRDB の稼働中に，スワップ先にできる状態にできるシステムログファイル数です。

ｂ：

システムログファイルのレコード数です。レコード数が各ファイルで異なる場合は平均値を求めてください。

ｃ：

シンクポイントダンプの有効化をスキップしてよい割合です。全システムログファイルに対して，上書きできない状態にしてよいファイルの割合を代入します。

- HiRDB/シングルサーバの場合は 0.333 以下の値にしてください。有効保証世代数が 2 の場合は 0.167 以下の値にしてください。
- バックエンドサーバの場合は 0.333 以下の値にしてください。有効保証世代数が 2 の場合は 0.167 以下の値にしてください。
- ディクショナリサーバの場合は 0.5 程度にしてください。
- フロントエンドサーバの場合は 0.7 程度にしてください。

ｄ：

システムログブロックの平均レコード数です。通常，3×4096÷f 程度を目安にしてください。なお，次に示す計算式から正確に求められます。

↑システムログ出力ブロック長の平均値÷ｆ↑

システムログ出力ブロック長の平均値は，統計解析ユティリティのシステムの稼働に関する統計情報（OUTPUT BLOCK LENGTH）から分かります。

ｅ：

pd_log_sdinterval オペランドの第１パラメタ（シンクポイントダンプの取得間隔をシステムログの出力量で指定）の値です。

ｆ：

システムログファイルのレコード長です。レコード長は pdlogls コマンドで調べられます。

(4) pd_spd_syncpoint_skip_limit オペランドの値を確認する方法

稼働中の HiRDB が適用しているシンクポイントダンプ有効化のスキップ回数の上限値は，pdlogls -d spd コマンドで確認できます。pdlogls -d spd コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(5) pd_spd_syncpoint_skip_limit オペランドに指定する値が適切でないと

指定値が大き過ぎると，全システムログファイルが上書きできない状態になる可能性があります。この場合，HiRDB は異常終了します。このとき，新規のシステムログファイルを追加しないと，HiRDB を再開始できません。

指定値が小さ過ぎると，強制的にロールバックされるトランザクションが増える可能性があります。

(6) シンクポイントダンプ有効化のスキップ回数監視機能を使用してはいけない場合

次に示す場合は，シンクポイントダンプ有効化のスキップ回数監視機能を使用しないでください。

- 大量更新のバッチ処理で，commit 文を発行するまでに出力されるシステムログ量が，全システムログファイルの合計容量の 1/3 以上となる場合
- データベースの更新量が最も多いトランザクションで出力するシステムログ量と，このトランザクションが稼働中に同時実行するトランザクションで出力するシステムログ量の合計が，全システムログファイルの合計容量の 1/3 以上となる場合
- 100,000 回を超えるシンクポイントダンプの有効化スキップ回数を監視する場合

なお，システムログ量の求め方については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(7) ロールバックの対象にならないトランザクション

シンクポイントの連続スキップ回数が pd_spd_syncpoint_skip_limit オペランドに指定した値を超えても，次に示すトランザクションはロールバックの対象とはなりません。

- 既にロールバック中のトランザクション
- OpenTP1 からの commit ２相目決着指示待ちのトランザクション
- ユティリティが生成したトランザクション

(8) スキップ回数にカウントされないケース

次に示す場合に実施されるシンクポイントダンプの有効化処理がスキップされたときは，そのスキップは連続スキップ回数としてカウントされません。

- 前回のシンクポイントダンプ有効化から，pd_log_sdinterval オペランドで指定した時間が経過した場合

- pdlogswap コマンドを実行してシステムログファイルをスワップした場合
- pdlogsync コマンドを実行した場合

> **参考**
>
> これらの場合，連続スキップ回数としてはカウントされませんが，KFPS02179-I メッセージに表示される「シンクポイントダンプ取得契機を無視した回数の累計」にはカウントされます。

(9)　注意事項

HiRDB の稼働中に pdlogswap コマンドを連続して実行すると，上書きできない状態のシステムログファイルが増えてしまうため，現用として使用可能なシステムログファイルが少なくなります。このため，システムログファイル不足によるユニットの異常終了が起こり易くなります。

(10)　HiRDB Datareplicator を使用している場合の注意事項

反映側 HiRDB のデータ反映トランザクション処理に時間が掛かる場合，シンクポイントダンプ有効化のスキップ回数監視機能によってデータ反映トランザクションを強制的にロールバックすることがあります。このとき，反映側 HiRDB では KFPS00993-I（REQUEST= abnormal_tran_end）メッセージを出力し，反映側 HiRDB Datareplicator では KFRB03007-W メッセージ及び KFRB03013-I メッセージを出力します。このときの対処方法を次に示します。

〈手順〉

1. **pdstop** コマンドで反映側 HiRDB を正常終了します。※
2. **pd_spd_syncpoint_skip_limit** オペランドの値を変更します。指定値については，「出力されるシステムログ量から計算する方法」を参照してください。
3. システムログファイルの世代数が次に示す条件式を満たすかどうかを確認してください。満たさない場合は，この条件式を満たすようにシステムログファイルを追加してください。

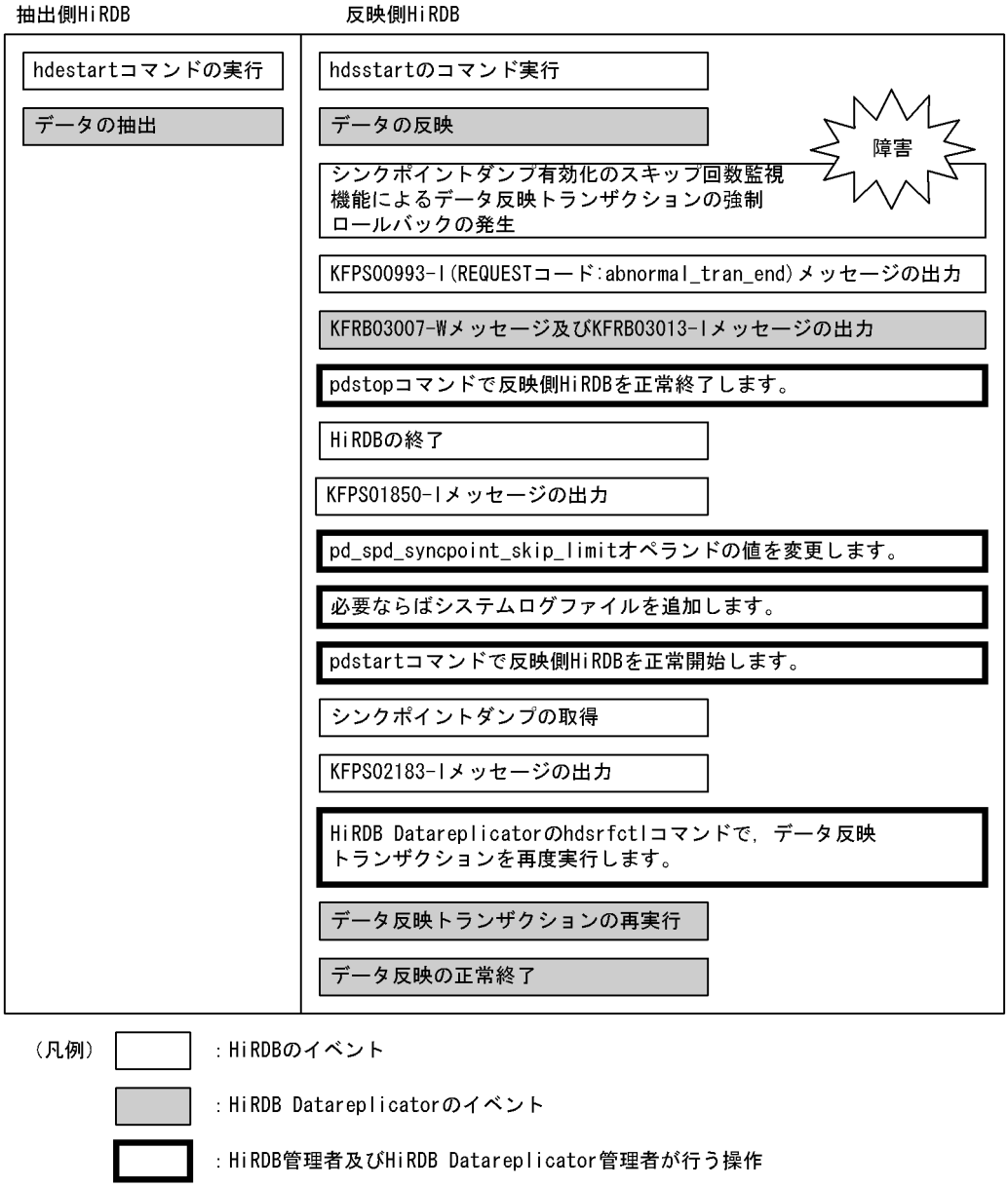
   変更後の pd_spd_syncpoint_skip_limit オペランドの値≦システムログファイルの世代数÷3
4. **pdstart** コマンドで反映側 HiRDB を正常開始します。
5. 反映側 HiRDB Datareplicator の **hdsrfctl** コマンドで，データ反映トランザクションを再度実行します。

注※

システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できるため，HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

シンクポイントダンプ有効化のスキップ回数監視機能によってデータ反映トランザクションが強制的にロールバックされるときの流れを次の図に示します。

図 8–13　シンクポイントダンプ有効化のスキップ回数監視機能によってデータ反映トランザクションが強制的にロールバックされるときの流れ



8.9　SQL 実行時間についての警告情報の出力（SQL 実行時間警告出力機能）

SQL の実行時間が次に示す，ある一定時間以上になった場合，その SQL に対して警告情報を出力できます。

- クライアントの最大待ち時間（PDCWAITTIME オペランドの値）に対するある比率以上の時間
- 警告をファイルに出力する契機となる時間

なお，ここではクライアントの最大待ち時間を PDCWAITTIME と表記し，クライアントの最大待ち時間を超えることを PDCWAITTIME オーバと表記します。

8.9.1　SQL 実行時間警告出力機能とは

SQL の実行後に HiRDB が SQL の実行時間を調べます。その結果，SQL の実行時間が設定した警告時間以上であった場合，その SQL に対して次に示す警告情報を出力します。これを **SQL 実行時間警告出力機能**といいます。

- SQL 実行時間警告情報ファイル
- 警告メッセージ（KFPA20009-W）

(1)　SQL 実行時間警告出力機能の使用目的

SQL 実行時間警告出力機能は次に示す目的などに使用します。

- データ量の増加などで HiRDB のサーバプロセスからの応答時間が長くなる UAP について，PDCWAITTIME オーバが発生する可能性があることを事前に検知する
- SQL 応答待ち時間が一定時間以上の SQL に関する情報を取得してチューニングの資料にする

(2)　警告時間の設定方法

警告情報を出力する時間を警告時間と呼びます。SQL 実行時間が警告時間以上になった場合，警告情報を出力します。警告時間は，次の値で決定されます。

- PDCWAITTIME に対する比率
- 警告をファイルに出力する契機となる時間：出力の契機となる時間を短い時間（ミリ秒単位など）で厳密に指定できます。

(3)　SQL 実行時間警告出力機能の対象となる SQL

CONNECT 以外の SQL を SQL 実行時間警告出力機能の対象とします。

(4)　警告情報が出力されたときの対処方法

警告情報が出力されて PDCWAITTIME オーバになる可能性がある場合は，出力された情報を参考にして次に示す対処をしてください。

1. 排他競合が発生しているかを見直す
2. ネットワーク障害が発生していないかを見直す
3. SQL をチューニングする
4. PDCWAITTIME オペランドの指定値を大きくする
5. データ件数の増加によって SQL の実行時間が長くなっていないかを確認する

(5)　警告情報の出力条件

SQL 実行時間警告出力機能を使用する場合，SQL の実行時間が設定した時間以上のとき以外にも警告情報を出力します。また，SQL 実行時間警告出力機能を使用しない場合でも警告情報を出力（メッセージだけ出力）するときがあります。SQL 実行時間警告出力機能の警告情報の出力条件を次の表に示します。

表 8–8　SQL 実行時間警告出力機能の警告情報の出力条件

条件		警告情報の出力可否	
		SQL 実行時間警告情報ファイル	KFPA20009-W メッセージ
SQL 実行時間警告出力機能を使用する場合	SQL の実行時間が設定した時間以上である	○	○
	PDCWAITTIME オーバによってサーバプロセスが強制終了された	△	△
	そのほかの要因によってサーバプロセスが強制終了された	△	△
SQL 実行時間警告出力機能を使用しない場合	SQL の実行時間が設定した時間以上である	×	×
	PDCWAITTIME オーバによってサーバプロセスが強制終了された	×	△
	そのほかの要因によってサーバプロセスが強制終了された	×	△

（凡例）

○：出力します。

△：出力しますが，一部の情報を出力しません。

また，強制終了のタイミングによっては，SQL 実行時間警告情報ファイル又は KFPA20009-W メッセージを出力しません。

なお，SQL 実行時間警告情報ファイル及び KFPA20009-W メッセージの再出力を pd_dump_suppress_watch_time オペランドで抑止できます。

×：出力しません。

注

SQL の実行時間が PDCWAITTIME オペランドの値を超えた場合は，SQL 実行時間警告出力機能の使用に関係なくサーバプロセスを強制終了します。

(6) PDCWAITTIME と SQL 実行時間警告出力機能の関係

PDCWAITTIME と SQL 実行時間警告出力機能の関係を次の図に示します。

図 8–14 PDCWAITTIME と SQL 実行時間警告出力機能の関係（1/2）

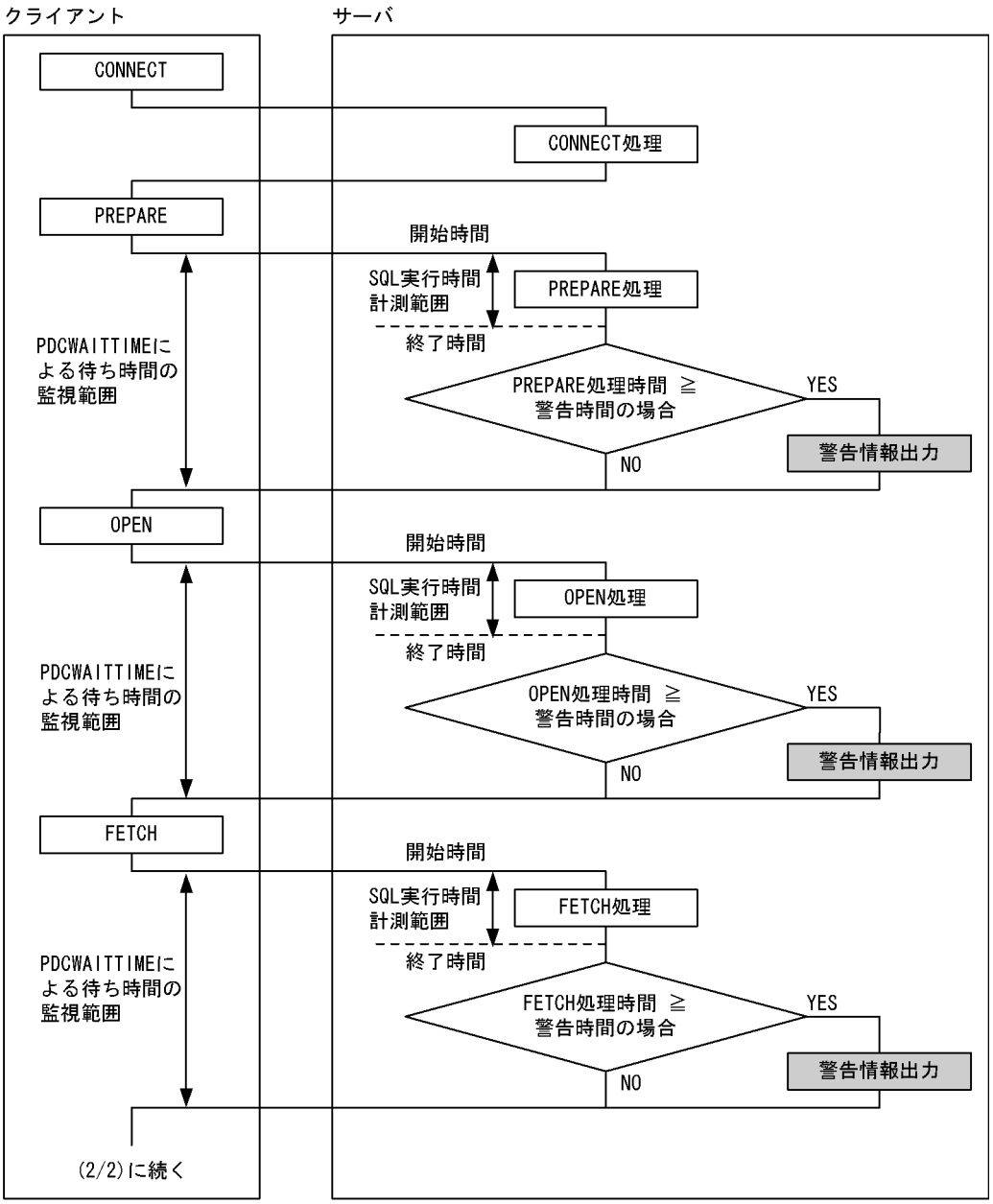
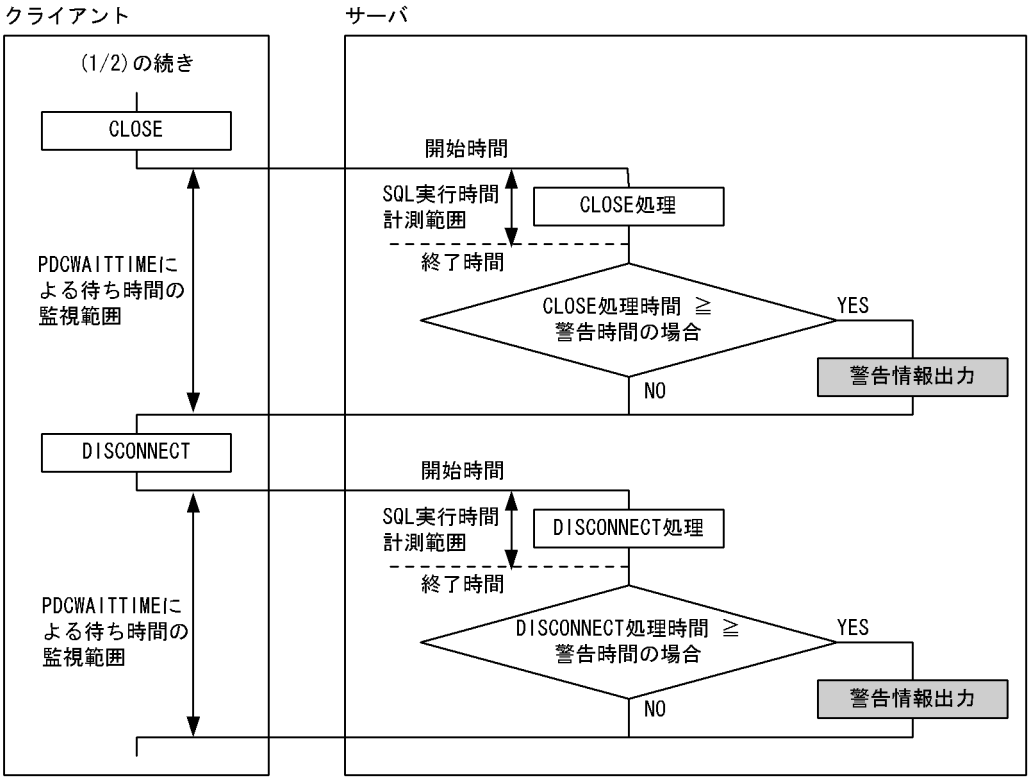


図 8–15　PDCWAITTIME と SQL 実行時間警告出力機能の関係（2/2）



〔説明〕

- クライアントからサーバに要求を受け渡したときから，要求に対する実行結果を返すまでを SQL 実行時間の計測範囲としています。
- SQL の実行結果をクライアントに返すときに，SQL の実行時間を HiRDB が調べます。SQL の実行時間が設定した警告時間以上の場合は警告情報を出力します。

8.9.2　SQL 実行時間警告出力機能の使用方法

SQL 実行時間警告出力機能を使用する場合は次に示すことをしてください。

- PDCWAITTIME オペランドの指定
- PDCWAITTIME に対する比率の指定，又は警告をファイルに出力する契機となる時間の指定
- SQL 実行時間警告情報ファイルの出力先ディレクトリの指定
- SQL 実行時間警告情報ファイルの最大容量の指定

(1)　PDCWAITTIME の指定

クライアント環境定義の PDCWAITTIME にクライアントの最大待ち時間を指定してください。
PDCWAITTIME に 0 を指定するか又は省略した場合，その HiRDB クライアントから実行される SQL については SQL 実行時間警告出力機能が適用されません。

ただし，PDCWAITTIMEWRNPNT，又は pd_cwaittime_wrn_pnt オペランドに「秒数 sec,auto」を指定した場合は，PDCWAITTIME オペランドを省略したときでも SQL 実行時間警告出力機能が適用されます。

PDCWAITTIME については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(2) PDCWAITTIME に対する比率の指定，又は警告をファイルに出力する契機となる時間の指定

警告情報を出力する条件を次に示すオペランドで指定します。

- PDCWAITTIMEWRNPNT（クライアント環境定義）
- pd_cwaittime_wrn_pnt オペランド（HiRDB システム定義）

(a) PDCWAITTIME に対する比率の指定

PDCWAITTIMEWRNPNT 及び pd_cwaittime_wrn_pnt オペランドで PDCWAITTIME に対する比率（0%～100%未満）を指定します。例えば，PDCWAITTIME に 100（秒）を指定し，PDCWAITTIMEWRNPNT 又は pd_cwaittime_wrn_pnt オペランドに 90（%）を指定すると，SQL の実行後に HiRDB が SQL の実行時間を調べます。その結果，SQL の実行時間が 90 秒以上 100 秒未満の場合に警告情報が出力されます。

符号なし整数，又は符号なし 10 進数（小数点を含む数，小数第 6 位まで）で指定できます。

(b) 警告をファイルに出力する契機となる時間の指定

上記のオペランドで SQL 実行時間の警告を出力する契機となる時間を，0 から PDCWAITTIME の設定値未満で指定します。

なお，PDCWAITTIMEWRNPNT の設定値に PDCWAITTIME の指定値より大きな時間を設定した場合には，CONNECT 要求時にエラーになります。

注意事項

- pd_cwaittime_wrn_pnt オペランドの指定よりも PDCWAITTIMEWRNPNT の指定の方が優先されます。
- **PDCWAITTIMEWRNPNT に 0%を指定した場合，その HiRDB クライアントから実行される SQL については SQL 実行時間警告出力機能が適用されません。**

参考

警告時間は，前記に示すオペランドの値から次に示す計算式を使用して HiRDB が決定します。

- 符号なし整数で PDCWAITTIME に対する比率を設定した場合
  警告時間＝ MAX（↓（ａ×ｂ）÷100↓，1）
  ａ：PDCWAITTIME の値

ｂ：PDCWAITTIMEWRNPNT 又は pd_cwaittime_wrn_pnt オペランドの値（これらのオペランドの優先度に従い適用された値が代入されます）

- 符号なし 10 進数で PDCWAITTIME に対する比率を設定した場合

  警告時間＝（ａ×ｂ）÷100

  ａ：PDCWAITTIME の値

  ｂ：PDCWAITTIMEWRNPNT 又は pd_cwaittime_wrn_pnt オペランドの値（これらのオペランドの優先度に従い適用された値が代入されます）

  ただし，警告時間は小数第 6 位まで有効です。小数第 7 位以下は切り捨てられます。

- 警告をファイルに出力する契機となる時間を設定した場合

  警告時間＝警告をファイルに出力する契機となる時間の絶対値

なお，HiRDB サーバ内部で使用するタイマの精度がプラットフォームに依存するため，タイマの精度が低い場合，実際の SQL の実行時間が警告時間を超えても警告情報が出力されないことがあります。

(3) SQL 実行時間警告情報ファイルの出力先ディレクトリの指定

SQL 実行時間警告情報ファイルの出力先ディレクトリを pd_cwaittime_report_dir オペランドに指定します。このディレクトリ下に二つのファイル（pdcwwrn1 及び pdcwwrn2）を作成します。

なお，SQL 実行時間警告情報ファイルへの出力時に，ファイルシステム障害，ディレクトリ又はファイルへの書き込み権限がないなどのエラーを OS が検知すると，SQL 実行時間警告情報ファイルへの出力を行いません。この場合でも SQL の実行処理は継続されます。

(4) SQL 実行時間警告情報ファイルの最大容量の指定

SQL 実行時間警告情報ファイルの最大容量を変更する場合に pd_cwaittime_report_size オペランドを指定します。このオペランドに指定した値は SQL 実行時間警告情報ファイル一つ分の容量です。SQL 実行時間警告情報ファイルは二つ作成されるため，指定値に注意してください。例えば，10000 を指定すると，ディレクトリ下に最大 10000 バイトのファイルが二つ作成されます。

備考

- ファイルへの出力容量がこのオペランドの値を超えた場合，出力先をもう片方のファイルに切り替えます。これを繰り返して二つのファイルを交互に使用します。このとき，切り替え先の古い情報は削除されます。
- 1 回で出力する SQL 実行時間警告情報の大きさがファイル容量を超えた場合，SQL 実行時間警告情報をすべて出力しません。ファイル容量までの SQL 実行時間警告情報を出力します。この場合，SQL 実行時間警告情報の最後に#を付加します。

8.9.3　SQL 実行時間警告情報ファイルの出力

(1)　SQL 実行時間警告情報ファイルの参照方法

SQL 実行時間警告情報は，テキストエディタなどテキストファイルを参照できるソフトウェアで参照できます。

なお，HiRDB/パラレルサーバの場合は，警告対象となった SQL を発行する UAP が接続するフロントエンドサーバがあるサーバマシンに出力されます。

備考

- 使用中の SQL 実行時間警告情報ファイルを知りたい場合，OS のコマンドなど（dir コマンドなど）でファイルの最終更新日時を調べてください。最終更新日時の新しいファイルが使用中の SQL 実行時間警告情報ファイルです。
- HiRDB を開始した後の出力先ファイルは，最終更新日時が新しいファイルになります。
- SQL 実行時間警告情報はファイルの最終位置から追加書きされるため，SQL 実行時間警告情報はファイル内で時系列順に表示されます。
- SQL の実行後，SQL 実行時間警告情報ファイルはクローズされているため，SQL が実行されていないときに OS のコマンドを利用してバックアップを取得したり，ファイルを参照したりしても，書き込み中のファイルを使用することはありません。また，SQL の実行中であっても出力先でないファイルであれば，書き込み中のファイルを使用することはありません。

(2)　SQL 実行時間警告情報の出力形式

SQL 実行時間警告情報の出力形式（符号なし整数で PDCWAITTIME に対する比率を設定した場合，又は警告をファイルに出力する契機となる時間に auto を指定した場合）を次に示します。

```
** SQL CWAITTIME WARNING INFORMATION 07-01    2002/07/04 14:32:22 **
  REASON(01)
  CWAITIME(600) CWAITTIME_WRN_PNT(70) CWAITTIME_WRN_TIME(420)

* UAP INFORMATION *
  UAP_NAME(userprog1) CLTPID(408)
  IP_ADDR(196.12.42.146) SERVICE_NAME(service1)
  USERID(hiuser01) START_TIME(2002/07/03 20:24:42)

* SERVER INFORMATION *
  HOST(host03) PORT(1146)
  SVRNAME(fes1) SVRPID(905)

* SQL INFORMATION *
  OPTIMIZE_LEVEL(132768) ADDITIONAL_OPTIMIZE_LEVEL(3)
  ISOLATION_LEVEL(2)
  SQLOBJ_SIZE(2608) SQLCOUNT(1)

CNCTNO     SQL-       OP   SEC  SQL   SQL   START-TIME          END-TIME EXEC-
           COUNTER    CODE NO   CODE  WARN                             TIME
```

```
---------- ---------- ---- ---- ----- ----- ------------------ -------- -----
        10         10 AUI2 1890     0 -0000 2002/07/04 14:32:22 14:39:30   428

* SQL MESSAGE *
  "*" [*]

* SQL STATEMENT *
  DELETE FROM ZAIKO WHERE ZNO=1
```

SQL 実行時間警告情報の出力形式（符号なし 10 進数で PDCWAITTIME に対する比率を設定した場合，又は警告をファイルに出力する契機となる時間を設定した場合）を次に示します。

```
** SQL CWAITTIME WARNING INFORMATION 07-01    2002/07/04 14:32:22.100000 **
  REASON(01)
  CWAITIME(600) CWAITTIME_WRN_PNT(70.001000) CWAITTIME_WRN_TIME(420.006000)

* UAP INFORMATION *
  UAP_NAME(userprog1) CLTPID(408)
  IP_ADDR(196.12.42.146) SERVICE_NAME(service1)
  USERID(hiuser01) START_TIME(2002/07/03 20:24:42)

* SERVER INFORMATION *
  HOST(host03) PORT(1146)
  SVRNAME(fes1) SVRPID(905)

* SQL INFORMATION *
  OPTIMIZE_LEVEL(132768) ADDITIONAL_OPTIMIZE_LEVEL(3)
  ISOLATION_LEVEL(2)
  SQLOBJ_SIZE(2608) SQLCOUNT(1)

CNCTNO     SQL-       OP   SEC  SQL   SQL
           COUNTER    CODE NO   CODE  WARN
---------- ---------- ---- ---- ----- -----
        10         10 AUI2 1890     0 -0000

START-TIME                 END-TIME        EXEC-TIME

-------------------------- --------------- ------------
2002/07/04 14:32:22.122222 14:39:30.822223   428.700001

* SQL MESSAGE *
  "*" [*]

* SQL STATEMENT *
  DELETE FROM ZAIKO WHERE ZNO=1
```

SQL 実行時間警告情報の出力内容を次の表に示します。

表 8–9　SQL 実行時間警告情報の出力内容

出力項目	ヘッダ名	説明	最大出力文字数(バイト)	出力可否	
				条件1	条件2
HiRDB のバージョン	SQL CWAITTIME WARNING INFORMATION	HiRDB のバージョンを VV-RR-ZZ の形式で表示します。ZZ がない場合は ZZ を表示しません。	8	○	○
出力日時		SQL 実行時間警告情報ファイルの出力日時を次のどちらかの形式で表示します。 YYYY/MM/DD hh:mm:ss （YYYY：年，MM：月，DD：日，hh：時，mm：分，ss：秒） YYYY/MM/DD hh:mm:ss.uuuuuu （YYYY：年，MM：月，DD：日，hh：時，mm：分，ss：秒，uuuuuu：マイクロ秒）	19 又は 26	○	○
理由コード	REASON	SQL 実行時間警告情報ファイルの出力理由を表示します。 • **00**：UAP の強制終了などでサーバプロセスが強制終了しました。 • **01**：SQL の実行時間が設定した時間以上になりました。	2	○	○
PDCWAITTIME オペランドの値	CWAITIME	クライアント環境定義の PDCWAITTIME オペランドの値を表示します（単位：秒）。	5	○	○
オペランドに指定した比率，又は時間	CWAITTIME_ WRN_PNT	PDCWAITTIMEWRNPNT オペランド又は pd_cwaittime_wrn_pnt オペランドに指定した比率（単位：%），又は時間（単位：秒）を表示します。これらのオペランドの優先度に従い適用された値を表示します。	2 又は 12	○	○
警告時間	CWAITTIME_ WRN_TIME	警告情報の出力条件となる時間※1を表示します（単位：秒）。	5 又は 12	○	○
UAP 名称	UAP_NAME	クライアント環境定義の PDCLTAPNAME オペランドに指定した UAP 名称を表示します。	30	○	○
プロセス番号	CLTPID	クライアントのプロセス番号を表示します。Type4 JDBC ドライバから接続している場合は，0 が表示されます。	10	○	○
IP アドレス	IP_ADDR	UAP を実行したクライアントの IP アドレスを表示します。	15	○	○

出力項目	ヘッダ名	説明	最大出力文字数(バイト)	出力可否	
				条件1	条件2
サービス名称	SERVICE_NAME	UAP の種類によって，表示されるサービス名称が次のようになります。 • **OpenTP1 の UAP の場合** OpenTP1 の SUP（サービス利用プログラム）が SPP（サービス提供プログラム）に要求したサービスの場合，又は TP1/Message Control が MHP（メッセージ処理プログラム）に要求したサービスの場合は，該当するサービス名称を表示します。 上記以外の場合は，＊を表示します。 • **OpenTP1 の UAP 以外の場合** ＊を表示します。	31	△	△
認可識別子	USERID	コネクトユーザ名を表示します。	30	○	○
UAP 開始時間	START_TIME	UAP の実行開始時刻を YYYY/MM/DD hh:mm:ss の形式で表示します（YYYY：年，MM：月，DD：日，hh：時，mm：分，ss：秒）。	19	○	○
ホスト名	HOST	サーバプロセスが稼働しているホスト名を表示します。	30	○	○
ポート番号	PORT	サーバプロセスの通信ポート番号を表示します。	5	○	△
サーバ名	SVRNAME	サーバ名を表示します。HiRDB/シングルサーバの場合はシングルサーバ名を，HiRDB/パラレルサーバの場合はフロントエンドサーバ名を表示します。	8	○	○
プロセス番号	SVRPID	サーバプロセスのプロセス番号を表示します。	10	○	○
SQL 最適化オプション	OPTIMIZE_LEVEL	SQL 最適化オプションの値を 10 進形式で表示します。取得できない場合は＊を表示します。	10	○	○
SQL 拡張最適化オプション	ADDITIONAL_OPTIMIZE_LEVEL	SQL 拡張最適化オプションの値を 10 進形式で表示します。取得できない場合は＊を表示します。	10	○	○
データ保証レベル	ISOLATION_LEVEL	データ保証レベルの値を表示します。取得できない場合は＊を表示します。	10	○	○
SQL オブジェクトのサイズ	SQLOBJ_SIZE	SQL オブジェクトのサイズ（単位：バイト）を表示します。取得できない場合は＊を表示します。	10	△	×

出力項目	ヘッダ名	説明	最大出力文字数(バイト)	出力可否	
				条件1	条件2
SQL の処理行数	SQLCOUNT	SQL の処理行数（SELECT 文で取り出した行数など）を表示します。取得できない場合は＊を表示します。 出力内容については，マニュアル「HiRDB SQL リファレンス」の SQLERRD を参照してください。	10	△	×
コネクト通番	CNCTNO	サーバが CONNECT を受け付けるごとにカウントしている通番を表示します。	10	○	○
SQL カウンタ	SQL-COUNTER	SQL 文を受け付けるごとにカウントしている通番を表示します。取得できない場合は＊を表示します。	10	○	×
オペレーションコード	OP CODE	SQL に対応するオペレーションコードを表示します。取得できない場合は＊を表示します。	4	○	×
セクション番号	SEC NO	SQL に対応するセクション番号を表示します。取得できない場合は＊を表示します。	4	△	×
SQLCODE	SQL CODE	SQL を実行した結果の SQLCODE を表示します。取得できない場合は＊を表示します。	5	○	×
警告情報	SQL WARN	警告情報を 16 進※2 で表示します。取得できない場合は＊を表示します。	5	○	×
SQL 開始時間	START-TIME	クライアントからの SQL の実行要求を受け付けた日時を次のどちらかの形式で表示します。 YYYY/MM/DD hh:mm:ss （YYYY：年，MM：月，DD：日，hh：時，mm：分，ss：秒） YYYY/MM/DD hh:mm:ss.uuuuuu （YYYY：年，MM：月，DD：日，hh：時，mm：分，ss：秒，uuuuuu：マイクロ秒） 取得できない場合は＊を表示します。	19 又は 26	○	△
SQL 終了時間	END-TIME	クライアントからの要求に対する処理が終了した時刻を次のどちらかの形式で表示します。 hh:mm:ss（hh：時，mm：分，ss：秒） hh:mm:ss.uuuuuu（hh：時，mm：分，ss：秒，uuuuuu：マイクロ秒）	8 又は 15	○	○

出力項目	ヘッダ名	説明	最大出力文字数(バイト)	出力可否	
				条件1	条件2
		サーバプロセスが強制終了した場合は，処理を中断した時刻を表示します。			
SQL 実行時間	EXEC-TIME	クライアントからの要求に対する処理時間を表示します（単位：秒）。サーバプロセスが強制終了した場合は，処理を中断するまでの時間を表示します。取得できない場合は＊を表示します。	5 又は 12	○	△
SQL メッセージ	SQL MESSAGE	SQL の実行中に出力されたメッセージを表示します。取得できない場合は＊を表示します。	254	△	×
		［ ］内の情報はシステムの保守情報です。取得できない場合は＊を表示します。	21	○	△
SQL 文	SQL STATEMENT	SQL 文を表示します。SQL 文中に注釈（コメント）や，SQL 最適化指定をしている場合は，それらも含めて表示します。取得できない場合は＊を表示します。	2,000,000	△	△

（凡例）

条件 1：SQL の実行時間が設定した時間以上になった場合

条件 2：SQL の実行時間が設定した時間以上になる前にサーバプロセスが強制終了した場合

○：常に出力されます。

△：出力されないことがあります。

×：出力されません。

注※1

警告時間は次に示す計算式から求められます。

- 符号なし整数で PDCWAITTIME に対する比率を設定した場合

警告時間＝ MAX（↓（ａ×ｂ）÷100↓，1）

ａ：PDCWAITTIME オペランドの値

ｂ：PDCWAITTIMEWRNPNT オペランド又は pd_cwaittime_wrn_pnt オペランドの値（これらのオペランドの優先度に従い適用された値が代入されます）

- 符号なし 10 進数で PDCWAITTIME に対する比率を設定した場合

警告時間＝（ａ×ｂ）÷100

ａ：PDCWAITTIME オペランドの値

ｂ：PDCWAITTIMEWRNPNT オペランド又は pd_cwaittime_wrn_pnt オペランドの値（これらのオペランドの優先度に従い適用された値が代入されます）

ただし，警告時間は小数第 6 位まで有効です。小数第 7 位以下は切り捨てられます。

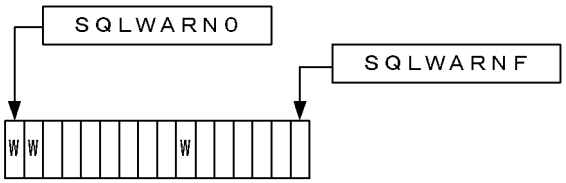
- 警告をファイルに出力する契機となる時間を設定した場合

警告時間＝警告をファイルに出力する契機となる時間の絶対値

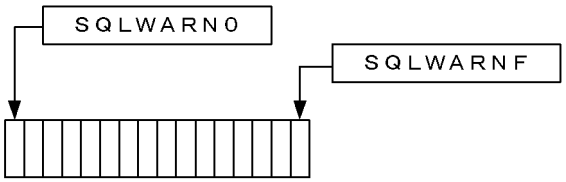
注※２

SQLWARN0～SQLWARNF の各警告情報の左から１ビットを割り当て，警告フラグが設定されているものは１，設定されていないものは０として 16 ビットの数値を求めます。この値を 4 けたの 16 進数値として表示します。一つ以上の警告フラグが設定されている場合は先頭に W を，警告フラグが設定されていない場合は–を表示します。

（例１）警告情報の内容が次に示す場合は，**WC040** が表示されます。



（例２）警告情報の内容が次に示す場合は，**-0000** が表示されます。



8.9.4　KFPA20009-W メッセージの出力

KFPA20009-W メッセージはメッセージログファイル及びイベントログに出力されます。

なお，ユニット内で過去 5 分以内に KFPA20009-W メッセージ（理由コード＝ 01）が出力された場合は，KFPA20009-W メッセージ（理由コード＝ 01）を再度出力しません。

ただし，複数のサーバプロセスで同時に SQL 実行時間警告情報の出力条件を満たした場合，5 分間に複数回 KFPA20009-W メッセージを出力することがあります。

8.9.5　注意事項

1. PDCWAITTIMEWRNPNT，pd_cwaittime_wrn_pnt オペランドともに「秒数 sec,auto」を指定しないで SQL 実行時間警告出力機能を使用すると，警告情報を出力しない場合でも SQL の開始時間と実行時間を取得するため，システムコールを発行する処理が余計に発生します。性能への影響を最小限にしたい場合は，「秒数 sec,auto」を使用してください。
2. SQL 実行時間警告情報を出力するタイミングによっては，「警告情報の出力条件」で説明した SQL 実行時間警告情報ファイルの出力条件の両方を満たすことがあります。そのため，同じ SQL の SQL 実行時間警告情報を 2 回出力することがあります。

3. クライアントが処理要求をしてから SQL 実行時間の計測を開始するまでの間，及び SQL 実行時間の計測を終えてから実行結果がクライアントに届くまでの間に，この機能の出力処理や通信処理が発生します。そのため，ネットワーク負荷又は入出力負荷によってこれらの処理時間が長くなると，次に示す現象が発生することがあります。
   - クライアント側の待ち時間が SQL 実行時間警告情報ファイルの出力条件を満たしているのに，SQL 実行時間警告情報ファイルが出力されない
   - PDCWAITTIME オーバでサーバプロセスが強制終了されたが，SQL 実行時間警告情報ファイルに出力されている SQL 実行時間は SQL 実行時間警告情報ファイルの出力条件を満たしていない

8.10 UAP 又はユティリティの実行時間の監視（無応答障害時の影響を抑える方法）

ここでは，UAP，ユティリティ，又は運用コマンドに無応答障害が発生したときにその影響を最小限に抑える方法について説明します。

夜間に実行するバッチジョブなどで障害（通信障害，瞬断を含む一時的な障害，ディスク障害など）が発生して UAP，ユティリティ，又は運用コマンドが無応答状態になると，そのほかの UAP，ユティリティ，又は運用コマンドもその影響を受けて停滞してしまい，最悪，翌日のオンライン業務に支障をきたすことがあります。UAP，ユティリティ，又は運用コマンドの無応答障害の影響を最小限に抑えるために，次に示すオペランド又はオプションを指定してください。

- クライアント環境定義の **PDCWAITTIME** オペランド
- システム共通定義の運用コマンド及びユティリティの実行時間監視オペランド（pd_utl_exec_time，pd_cmd_exec_time）
- 各運用コマンド及びユティリティの実行時間監視オプション
- 各ユティリティの exec_time オペランド

これらのオペランド又はオプションで指定した時間を過ぎても UAP，ユティリティ，又は運用コマンドの実行が終了しない場合，その UAP，ユティリティ，又は運用コマンドを強制終了します。これによって，そのほかの業務に与える影響を最小限に抑えられます。これらのオペランド又はオプションには無応答障害が発生した可能性が高いと考えられる時間を指定してください。

クライアント環境定義の PDCWAITTIME オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

システム共通定義の pd_utl_exec_time と pd_cmd_exec_time は，実行時間を監視する対象の運用コマンド及びユティリティが異なるため，必要に応じてどちらかのオペランドを指定してください。なお，監視できる運用コマンド及びユティリティについては，マニュアル「HiRDB システム定義」の各オペランドの説明を参照してください。

各運用コマンド及びユティリティのオプション，各ユティリティのオペランドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

8.11　リソースの使用率の監視

次の表に示すリソースの使用率がある一定の値に達した場合に，警告メッセージを出力できます。警告メッセージを出力するかどうかは，システム共通定義のオペランドで指定します。また，これらのオペランドで警告メッセージの出力契機を指定できます。例えば，リソースの使用率が 90%に達したときに警告メッセージを出力できます。

表 8–10　使用率を監視できるリソース

指定するオペランド※	使用率を監視できるリソース	出力する警告メッセージ
pd_max_users_wrn_pnt	pd_max_users オペランドで指定した最大同時接続数	KFPS05123-W
pd_max_access_tables_wrn_pnt	pd_max_access_tables オペランドで指定した同時アクセス可能実表数	
pd_max_rdarea_no_wrn_pnt	pd_max_rdarea_no オペランドで指定した RD エリアの最大数	
pd_max_file_no_wrn_pnt	pd_max_file_no オペランドで指定した RD エリアを構成する HiRDB ファイルの最大数	
pdwork_wrn_pnt	pdwork オペランドで指定した作業表用ファイル用の HiRDB ファイルシステム領域	
pd_max_list_users_wrn_pnt	pd_max_list_users で指定したリスト作成ユーザ数	
pd_max_list_count_wrn_pnt	pd_max_list_count で指定した 1 ユーザ当たりのリスト作成数	
pd_aud_file_wrn_pnt	スワップ先にできない監査証跡ファイル数	
pd_rdarea_list_no_wrn_pnt	サーバ内のリスト作成数	KFPH22023-W

注※

pd_watch_resource = AUTO を指定すると，表「使用率を監視できるリソース」に示すリソースの使用率が 80%を超えるとメッセージが出力されます。この場合，表「使用率を監視できるリソース」に示す各オペランドを指定する必要はありません。メッセージの出力契機を変更したい場合（80%以外にしたい場合）に，各オペランドで出力契機を指定してください。

8.12　サーバプロセスの状態監視（メッセージキュー監視機能）

8.12.1　メッセージキュー監視機能の概要

サーバプロセスの沈み込みが発生したサーバでは，UAP のレスポンスの低下やシステムのハングアップなどが起こることがあります。ここでは，メッセージキュー監視機能を使用してサーバプロセスの沈み込みを監視する方法について説明します。

なお，サーバプロセスの沈み込みとは，CPU 負荷による処理性能低下，又は入出力障害による入出力遅延によって，サーバプロセスが処理されない状態のままになることです。

(1)　メッセージキュー監視機能とは

HiRDB では，サーバプロセスの割り当て処理でメッセージキューを使用しています。サーバプロセスの沈み込みが発生すると，メッセージキューからメッセージを取り出せなくなります。HiRDB では，ある一定時間（これを**メッセージキュー監視時間**といいます）を超えてもメッセージキューからメッセージを取り出せない場合，警告メッセージ及びエラーメッセージ（KFPS00888-W 及び KFPS00889-E）を出力します。これを**メッセージキュー監視機能**といいます。このメッセージが出力されると，サーバプロセスが沈み込んでいる可能性があります。

メッセージキュー監視時間は，通常 600 秒間（10 分間）です。この監視時間を **pd_queue_watch_time** オペランドで変更できます。

(2)　警告メッセージが出力された場合の対処方法

警告メッセージが出力された場合，サーバプロセスが沈み込んでいる可能性があります。この場合，次に示すどちらかの方法で対処してください。

- ユニットを再開始する
- トランザクションをキャンセルする

(a)　ユニットを再開始する場合

サーバプロセスの沈み込みが発生したユニットを再開始すると，サーバプロセスの沈み込みを解決できることがあります。通常，メッセージキュー監視時間を過ぎると，HiRDB は沈み込みが発生したユニットを異常終了させます。

なお，ユニットを異常終了させたくない場合は，**pd_queue_watch_timeover_action** オペランドに continue を指定してください。

(b)　トランザクションをキャンセルする場合

(a)の方法で対処しない場合（できない場合も含む），サーバプロセスの沈み込みが発生したサーバで実行中のトランザクションを **pdcancel** コマンドなどで終了させてください。トランザクションが発生してい

ない場合は，サーバプロセスの沈み込みが発生したサーバを pdkill コマンドなどで終了させた後，沈み込み要因を特定して対策してください。

(3)　サーバプロセスの沈み込みの対策方法

メッセージキュー監視対象サーバの，メッセージキュー滞留要因と対策を次の表に示します。

表 8–11　メッセージキュー滞留要因と対策

要因	監視対象サーバプロセス				対策
	FES	BES	DS	SDS	
CPU 負荷が高いため，メッセージキューからメッセージを取り出せません。	○	○	○	○	CPU 負荷になっている要因を調査して，対策してください。
入出力障害によって入出力が遅延しているため，メッセージキューからメッセージを取り出せません。	○	○	○	○	入出力障害になっている要因を調査して，対策してください。
pd_max_users オペランドの値を超えて同時に接続要求が発生した場合に，メッセージを取り出すプロセス数が不足しています（高速接続機能を使用した場合に発生しやすい）。	○	－	－	○	同時接続数を減らすか，又は高速接続機能を使用しないで通常接続を行ってください。若しくは，pd_max_users オペランドの値を大きくしてください。
フロントエンドサーバ，又はユティリティサーバの起動数に比べて，バックエンドサーバ，及びディクショナリサーバの起動数が少ない場合に，メッセージを取り出すプロセス数が不足しています（マルチフロントエンドサーバ環境で発生しやすい）。	－	○	○	－	pd_max_bes_process，pd_max_dic_process，及び pd_max_users オペランドの値が正しく指定されているか確認してください。又は，接続数を減らしてください。

（凡例）

○：該当します。

－：該当しません。

参考

HiRDB サーバプロセス数は次に示すオペランドで制限されています。

- pd_max_server_process
  ユニット内で稼働するサーバ数が多い場合は，このオペランドの指定値を詳細に見積もってください。また，スタンバイレス型系切り替え機能を使用する場合は，系切り替えが発生したときを想定した見積もりが必要になります。
- pd_max_bes_process
  マルチフロントエンドサーバ，又は 1：1 スタンバイレス型系切り替え機能を使用する場合は，このオペランドの指定値を詳細に見積もってください。

- pd_max_dic_process
  マルチフロントエンドサーバを使用する場合は，このオペランドの指定値を詳細に見積もってください。
- pd_ha_max_server_process
  影響分散スタンバイレス型系切り替え機能を使用する場合は，このオペランドの指定値を詳細に見積もってください。
- pd_max_users
  同時接続数が多い場合は，適切な値を指定する必要があります。

(4) 備考

メッセージキューから最後に取り出したメッセージの取り出し時間を **pdls -d scd** コマンドで調べられます。

8.13　サーバプロセスの異常終了回数の監視（プロセスの異常終了回数監視機能）

サーバプロセスの異常終了が多発すると，新たなサービスを受け付けられないことがあります。しかし，サーバプロセスの異常終了では HiRDB を異常終了しないため，実質オンライン停止状態になります。この実質オンライン停止状態を防ぐためにプロセスの異常終了回数監視機能があります。

8.13.1　プロセスの異常終了回数監視機能とは

サーバプロセスの異常終了回数が一定時間内に **pd_down_watch_proc** オペランドの値を超えた場合，HiRDB（HiRDB/パラレルサーバの場合は該当するユニット）を異常終了します。これを**プロセスの異常終了回数監視機能**といいます。

この機能は系切り替え機能を使用する場合に使用することをお勧めします。サーバプロセスの異常終了が多発した場合に HiRDB を異常終了するため，すぐに系を切り替えられます。この機能を使用しないと HiRDB が異常終了しないため，系が切り替わりません。

また，系切り替え機能を使用しない場合でも，HiRDB の再開始によってメモリ及びリソース状態がリフレッシュされて処理効率が向上します。

なお，プロセスの異常終了回数監視機能によって HiRDB が異常終了する場合，KFPS01821-E 及び KFPS00729-E メッセージが出力されます。

8.13.2　プロセスの異常終了回数監視機能の適用範囲

この機能では，PDCWAITTIME オーバ，又はアボートによって異常終了するサーバプロセスを監視対象とします。HiRDB/シングルサーバの場合は，シングルサーバプロセスの異常終了回数がカウントされます。HiRDB/パラレルサーバの場合は，ユニット内のフロントエンドサーバ，バックエンドサーバ，及びディクショナリサーバプロセスの異常終了回数の合計値がカウントされます。サーバプロセスが異常終了する要因と異常終了回数のカウント対象を次の表に示します。

表 8–12　サーバプロセスが異常終了する要因と異常終了回数のカウント対象

サーバプロセスが異常終了する要因	異常終了回数のカウント対象			
	シングルサーバプロセス	フロントエンドサーバプロセス	ディクショナリサーバプロセス	バックエンドサーバプロセス
クライアント環境定義の PDCWAITTIME オペランドの値を超えた	○	○	×※1	×※1
pdcancel コマンド	×	×※2	×	×

サーバプロセスが異常終了する要因	異常終了回数のカウント対象			
	シングルサーバプロセス	フロントエンドサーバプロセス	ディクショナリサーバプロセス	バックエンドサーバプロセス
内部強制終了（HiRDB が内部的に SIGKILL を発行してサーバプロセスを終了させる場合）	○※3	○※3	×※1	×※1
アボート	○	○	○	○
OLTP システムのトランザクション回復処理によるサーバプロセスの異常終了	○	○	×	×
上記以外のサーバプロセス異常終了	○	○	○	○

（凡例）

○：サーバプロセスの異常終了回数にカウントされます。

×：サーバプロセスの異常終了回数にカウントされません。

注※1

トランザクションブランチが発生した状態で障害を検知した場合，同一トランザクションブランチから発生したフロントエンドサーバプロセスの異常終了がカウントの対象になります。

注※2

pdcancel コマンドでバックエンドサーバプロセス又はディクショナリサーバプロセスを強制終了すると，フロントエンドサーバプロセスを内部的に強制終了します。この場合，フロントエンドサーバプロセスの異常終了がカウントされることがあります。

注※3

OLTP システムによるグローバルトランザクションが発生した状態で障害を検知した場合，同一グローバルトランザクションから発生したシングルサーバプロセス又はフロントエンドサーバプロセスの異常終了がカウントの対象になります。

8.13.3　プロセスの異常終了回数監視機能の指定方法

pd_down_watch_proc オペランドで，サーバプロセスの異常終了回数を監視する時間と異常終了回数の上限を指定します。

（例）pd_down_watch_proc = 1000，60

この場合，60 秒間隔でサーバプロセスの異常終了回数を監視します。60 秒以内にサーバプロセスの異常終了回数が 1000 を超えると，HiRDB を異常終了します。

8.13.4　備考

- サーバプロセスが異常終了すると，KFPS01820-E メッセージが出力されます。pdcancel コマンドでサーバプロセスが異常終了したときにも KFPS01820-E メッセージが出力されますが，これは異常終了回数のカウントに含まれません。
- 相互系切り替え構成の場合，系を切り替えると同一サーバマシンで複数の HiRDB を稼働するため，逆にトラフィックが上昇して効果が得られないことがあります。プロセスの異常終了回数監視機能を使用する場合は，HiRDB が異常終了した系で再開始することをお勧めします。

8.14　ロールバック処理が進行中であることを確認する方法

ロールバックしているトランザクションがある場合，次の情報を基にロールバック処理が進行中であることを確認できます。

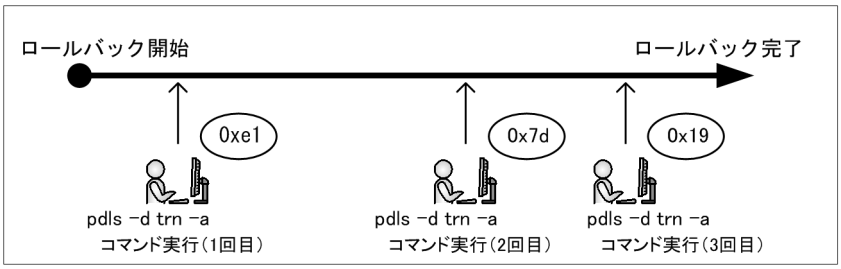
- pdls -d trn -a コマンドの実行結果
- KFPS02235-I メッセージ

8.14.1　pdls -d trn -a コマンドの実行結果

pdls -d trn -a コマンドの実行結果に表示される入力中のシステムログのブロック番号を確認します。pdls -d trn -a コマンドを時間をおいて複数回実行し，実行結果に表示されている入力中のシステムログのブロック番号が前回の番号と異なっていればロールバック処理は進行中です。

ロールバック処理が進行中の例を次の図に示します。

図 8–16　ロールバック処理が進行中の例（pdls -d trn -a コマンドの場合）



〔説明〕

内の番号は，入力中のシステムログのブロック番号を示しています。

実行結果に表示される入力中のシステムログのブロック番号が変わっているため，ロールバック処理は進行中です。

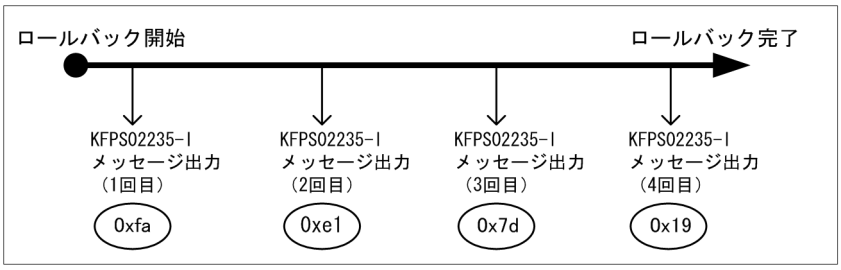
ロールバック処理が進行中であることを確認できない状態が続く場合は，何らかの障害によってロールバック処理が止まっている可能性があります。「障害調査のために必要な情報」を参照し，無応答の障害時に必要な情報を取得後，保守員に連絡してください。

8.14.2　KFPS02235-I メッセージ

KFPS02235-I メッセージに表示される入力中のシステムログのブロック番号を確認します。KFPS02235-I メッセージは，pd_trn_rollback_msg_interval オペランドに指定した値の間隔で出力されます。メッセージに表示されている入力中のシステムログのブロック番号が，前回の番号と異なっていればロールバック処理は進行中です。

ロールバック処理が進行中の例を次の図に示します。

図 8–17　ロールバック処理が進行中の例（KFPS02235-I メッセージの場合）



〔説明〕

○ 内の番号は，入力中のシステムログのブロック番号を示しています。

メッセージに表示される入力中のシステムログのブロック番号が変わっているため，ロールバック処理は進行中です。

ロールバック処理が進行中であることを確認できない状態が続く場合は，何らかの障害によってロールバック処理が止まっている可能性があります。「障害調査のために必要な情報」を参照し，無応答の障害時に必要な情報を取得後，保守員に連絡してください。

8.14.3　ロールバック完了の目安

ロールバック処理は，最大で前回のシンクポイントダンプ取得位置までログブロックを読み込むことがあります。そのため，前回のシンクポイントダンプ取得時点のシステムログのブロック番号を，ロールバック完了の目安としてください。ロールバック完了の目安の確認方法を次の表に示します。

項番	ロールバックの契機	ロールバック処理で最大に読み込むシステムログのブロック番号	確認方法
1	HiRDB の再開始時のロールバック以外	前回のシンクポイントダンプ取得時点のシステムログのブロック番号	シンクポイントダンプの取得時に出力される，KFPS02183-I メッセージで確認できます。
2	HiRDB の再開始時のロールバック	ログ入力開始時点のシステムログのブロック番号	ログの読み込み開始時に出力される，KFPS01262-I メッセージで確認できます。

8.14.4　注意事項

データベース回復ユティリティ（pdrstr）を実行すると，回復の対象になるトランザクションの中で無効にするトランザクションもロールバックされます。このロールバック処理の場合，pdls -d trn -a コマンドを実行してもロールバック処理の進行状況は確認できません。また，KFPS02235-I メッセージも出力されません。

このロールバック処理が進行しているかどうかは，データベース回復ユティリティが出力する次のメッセージで確認できます。

- KFPR26281-I メッセージ
  処理の内容が"Roll back"のとき，ロールバックが開始，又は終了したことを表します。

- KFPR26282-I メッセージ

  アンロードログファイル，又はシステムログファイルを 5 ギガバイト読み込むたびにメッセージを出力し，現在までに読み込んだファイルサイズを表示します。

9 システムの稼働環境を変更する方法

この章では，システムの稼働環境を変更する方法について説明します。

9.1　HiRDB システム定義を変更する方法

実行者　HiRDB 管理者

ここでは，HiRDB システム定義（UAP 環境定義を除く）の変更方法を説明します。UAP 環境定義の変更方法については，マニュアル「HiRDB システム定義」を参照してください。

9.1.1　変更手順

HiRDB の実行環境を変更するには，HiRDB システム定義を変更します。HiRDB システム定義を変更するときの手順を次に示します。なお，%PDDIR%¥conf はユニット制御情報定義ファイルを格納しているディレクトリを意味しています。%PDCONFPATH%はそれ以外の HiRDB システム定義ファイルを格納しているディレクトリを意味しています。

〈手順〉

1. %PDDIR%¥conf 及び%PDCONFPATH%下にサブディレクトリを作成します。この例ではサブディレクトリとして work を作成します。
2. ユニット制御情報定義ファイルを%PDDIR%¥conf¥work 下にコピーします。そのほかの HiRDB システム定義ファイルを%PDCONFPATH%¥work 下にコピーします。
3. %PDDIR%¥conf¥work 及び%PDCONFPATH%¥work 下にコピーした HiRDB システム定義を変更します。
4. **pdconfchk -d work** コマンドで，%PDDIR%¥conf¥work 及び%PDCONFPATH%¥work 下の HiRDB システム定義の内容をチェックします。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
5. **pdstop** コマンドで HiRDB を正常終了します。
6. **pdlogunld** コマンドで，アンロード待ち状態のシステムログファイルをアンロードします。
7. 3 で変更した HiRDB システム定義ファイルを%PDDIR%¥conf 又は%PDCONFPATH%下にコピーして，HiRDB システム定義ファイルを置き換えます。
8. 次に示すオペランドの指定値を変更した場合は，**pdloginit** コマンドでシステムログファイルを初期化します。
   - pd_log_dual
   - pdstart
9. **pdstart** コマンドで HiRDB を正常開始します。

9.1.2　注意事項

1. HiRDB/パラレルサーバの場合はユニットごとに%PDDIR%¥conf 及び%PDCONFPATH%下にサブディレクトリを作成して，HiRDB システム定義の内容をチェックしてください。
2. 稼働中の HiRDB が使用している HiRDB システム定義は変更又は削除しないでください。変更又は削除した場合はその HiRDB の動作を保証できません。
3. HiRDB が計画停止，強制終了，又は異常終了した場合，HiRDB システム定義のオペランドで変更できるものと変更できないものがあります。変更できないオペランドについては，マニュアル「HiRDB システム定義」を参照してください。
4. HiRDB システム定義を変更した後に，%PDDIR%¥conf 下のファイルのバックアップを取得してください。HiRDB 運用ディレクトリがあるディスクの障害などに備えて，HiRDB 運用ディレクトリ下のファイル（%PDDIR%¥conf 下のファイル）のバックアップを取得します。HiRDB 運用ディレクトリを回復するには，%PDDIR%¥conf 下のファイルのバックアップが必要になります。また，%PDCONFPATH%が HiRDB 運用ディレクトリ下にある場合は，同様にバックアップを取得してください。
5. システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できるため，HiRDB を正常終了する必要はありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。
6. スタンバイレス型系切り替え機能使用時の注意事項です。正規 BES ユニットの HiRDB システム定義を変更する場合は，正規 BES ユニット及び代替 BES ユニットを pdstop -u コマンドで正常終了してから変更してください。HiRDB システム定義の変更後，正規 BES ユニットのユニット制御情報定義ファイル及びバックエンドサーバ定義ファイルを代替 BES ユニットにコピーしてください。詳細については，「1：1 スタンバイレス型系切り替え機能の場合」を参照してください。

9.1.3　HiRDB/パラレルサーバ固有の注意事項

1. システム共通定義を修正する場合はすべてのサーバマシンのシステム共通定義を同じように修正してください。
2. 正常終了又は計画停止したときに一部のユニットが異常終了した場合，次回開始前に HiRDB システム定義を変更しないでください。変更すると，HiRDB の開始に失敗します。失敗しなくても，開始後に正常に稼働できなくなります。

9.1.4　HiRDB Datareplicator と連携している場合

次に示すオペランドを追加，変更，又は削除する場合は HiRDB Datareplicator を終了してください。そして，オペランドの追加，変更，又は削除後に HiRDB Datareplicator を開始してください。

- pd_log_dual

- pd_log_max_data_size
- pdlogadfg -d sys
- pdlogadpf -d sys

HiRDB Datareplicator を稼働したまま，これらのオペランドを追加，変更，又は削除した場合，HiRDB Datareplicator による抽出が失敗することがあります。

9.2 HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）

実行者　HiRDB 管理者

ここでは，システム構成変更コマンド（pdchgconf コマンド）を使用した HiRDB システム定義の変更方法について説明します。システム構成変更コマンドを使用すると，HiRDB の稼働中に HiRDB システム定義を変更できるため，HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

9.2.1 変更手順

システム構成変更コマンドを使用した HiRDB システム定義の変更方法を次に示します。

〈手順〉

1. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
2. 使用中の HiRDB システム定義ファイルを 1 で作成したディレクトリ下にコピーします。
3. %PDDIR%¥conf¥chgconf 下の HiRDB システム定義を変更します。
4. **pdconfchk -d chgconf** コマンドで，%PDDIR%¥conf¥chgconf 下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。
5. **pdchgconf** コマンドで，HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。
   pdchgconf コマンドを実行すると，使用中（変更前）の HiRDB システム定義ファイルが%PDDIR%¥conf¥backconf 下に退避されます。そして，%PDDIR%¥conf¥chgconf 下の変更後の HiRDB システム定義ファイルが%PDDIR%¥conf 下にコピーされます。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

注意事項

- %PDDIR%¥conf，%PDDIR%¥conf¥chgconf，%PDDIR%¥conf¥backconf ディレクトリ，及びこれらディレクトリ下のすべてのファイルに HiRDB 管理者の書き込み権限と参照権限を与えてください。
- クライアント環境定義の PDAUTORECONNECT に YES を指定し，自動再接続機能を使用するようにしてください。自動再接続機能を使用すると，HiRDB サーバとの接続が切断しても UAP にエラーを返却しないで処理を続行できます。
- pdchgconf コマンドの入力後，15 分以上トランザクション又はユティリティが動き続けた場合，pdchgconf コマンドが異常終了します。

9.2.2　オペランドの指定値を変更するときの注意事項

(1)　指定値を変更できないオペランド

次に示すオペランドの指定値は変更できません。

- pd_system_id
- pd_master_file_name

これらのオペランドの指定値を変更するとデータベースを初期化する必要があるため，システム構成変更コマンドで HiRDB システム定義を変更できません。

(2)　削除できないオペランド

次に示すオペランドは削除できません。

- pdlogadfg -d sys 及び pdlogadpf -d sys（現用及び上書きできない状態のシステムログファイルに対応するオペランドだけが削除できません）
- pdlogadfg -d spd 及び pdlogadpf -d spd（書き込み中及び上書きできない状態のシンクポイントダンプファイルに対応するオペランドだけが削除できません）
- pd_syssts_file_name_1～7（現用のユニット用ステータスファイルに対応するオペランドだけが削除できません）
- pd_sts_file_name_1～7（現用のサーバ用ステータスファイルに対応するオペランドだけが削除できません）

(3)　無効になるオペランド

縮退起動に関するオペランドの指定は無効になります。

(4)　pd_rpl_init_start オペランド変更時の注意事項

pdrplstart 又は pdrplstop コマンドで HiRDB Datareplicator のデータ抽出機能の実行状況を変更した場合は注意が必要です。pd_rpl_init_start オペランドの値によってはシステム構成変更コマンドが実行できません。pd_rpl_init_start オペランドの値とシステム構成変更コマンドの実行可否を次の表に示します。

表 9–1　pd_rpl_init_start オペランドの値とシステム構成変更コマンドの実行可否

変更前の pd_rpl_init_start オペランドの値	HiRDB Datareplicator のデータ抽出機能の実行状況	変更後の pd_rpl_init_start オペランドの値	システム構成変更コマンドの実行可否
Y	実行中	Y	○
		N（省略値）	×

変更前の pd_rpl_init_start オペランドの値	HiRDB Datareplicator のデータ抽出機能 の実行状況	変更後の pd_rpl_init_start オペランドの値	システム構成変更 コマンドの 実行可否
	pdrplstop コマンドによって停止中	Y	×
		N（省略値）	○
N（省略値）	停止中	Y	×
		N（省略値）	○
	pdrplstart コマンドによって実行中	Y	○
		N（省略値）	×

（凡例）

○：pd_rpl_init_start オペランドの値を変更できます。

×：pd_rpl_init_start オペランドの値を変更できません。

(5)　前提製品が必要なオペランド

付加 PP 又はプラグインのセットアップや削除をする場合は HiRDB を終了する必要があります。このため，付加 PP 又はプラグインなどの前提製品が必要なオペランドを削除したり，指定値を変更したりできないことがあります。

9.2.3　システム構成変更コマンド実行時の注意事項

システムログファイルについて（重要）

システム構成変更コマンドを実行すると，システムログファイルがスワップします。このため，スワップ先にできる状態のファイルがあることを確認してからシステム構成変更コマンドを実行してください。スワップ先にできる状態のファイルがないときにシステム構成変更コマンドを実行すると，HiRDB が異常終了します。この場合，スワップ先にできる状態のファイルを準備した後に pdstart コマンドで HiRDB を開始してください。

システム構成変更コマンドを実行できない環境

次に示す条件を一つでも満たす場合はシステム構成変更コマンドを実行できません。

- pdstart -r コマンドで HiRDB を開始した場合
- 一部のユニット又はサーバが停止中の場合（縮退起動も含む）※
- ユニット間のネットワークに通信障害が発生中の場合※
- pdrplstop コマンドが実行中の場合
- インメモリ RD エリアがある場合
- SQL セッション固有一時表が実体化している場合

注※　HiRDB/パラレルサーバ限定の条件です。

参考

回復不要 FES を使用している場合，システム構成変更コマンドを実行できない条件が変わります。詳細については，「回復不要 FES との関連（HiRDB/パラレルサーバ限定）」を参照してください。

システム構成変更コマンド実行時の制限事項

- システム構成変更コマンドの実行中は，コマンド及びユティリティを実行しないでください。実行した場合，HiRDB が停止している旨のエラーメッセージが出力されることがあります。
- システム構成変更コマンドを実行するとカーソルを保持できなくなるため，ホールダブルカーソルを使用する UAP の実行中にシステム構成変更コマンドを実行しないでください。
- システム構成変更コマンドを実行すると UNTIL DISCONNECT 指定の排他を保持できなくなるため，UNTIL DISCONNECT 指定の LOCK 文を使用する UAP の実行中にシステム構成変更コマンドを実行しないでください。
- システム構成変更コマンドの実行中は UAP のレスポンスが通常より遅くなります。
- 07-00 より前の HiRDB クライアントを使用している場合，システム構成変更コマンドを実行すると HiRDB サーバと HiRDB クライアントの接続が切断されます。このため，07-00 より前の HiRDB クライアントを使用している場合は，業務に支障がないように配慮してください。

HiRDB Datareplicator 連携時の注意事項

HiRDB Datareplicator との連携中に，ユニット又はサーバの追加，削除をする場合，HiRDB Datareplicator の抽出環境を再構築する必要があります。また，HiRDB Datareplicator の抽出環境の再構築中に HiRDB のトランザクションが動作して抽出側 DB の追加，更新，又は削除を行うと，反映側 DB との不整合が発生します。このため，HiRDB Datareplicator の抽出環境の再構築中にシステム構成変更コマンドを実行しないでください。

HiRDB/シングルサーバ限定の注意事項

次に示す場合は，システム構成変更コマンドの実行がエラーになり HiRDB が停止することがあります。

- HiRDB Version 6 より古いバージョンで HiRDB を稼働したことがある場合
- pdstop コマンドで HiRDB を終了するときに Windows のサービスも停止する場合（pdntenv コマンドの-sc auto オプションを指定した場合）

システム構成変更コマンドを実行する場合は，pdntenv コマンド（-sc manual オプション指定）で pdstop コマンド実行時に Windows のサービスが終了しないように設定してください。

9.2.4　システム構成変更コマンド実行後の HiRDB の状態

システム構成変更コマンドを実行すると，HiRDB は正常開始したときと同じ状態になるため，次に示すことが発生します。

1. システムログファイルがスワップします。
2. メッセージログファイルがスワップします。メッセージログファイル中のメッセージを保存する場合は，システム構成変更コマンドの実行前にメッセージログファイルのバックアップを取得してください。
3. データベース構成変更ユティリティで割り当てたグローバルバッファが無効になります。
4. pdbufmod コマンドで動的変更したグローバルバッファが無効になります。
5. pd_spool_cleanup_interval オペランドで指定するトラブルシュート情報の削除処理間隔の時間カウントが 0 に戻ります。
6. pd_spool_cleanup オペランドに normal 又は force（省略値）を指定している場合，出力済みのトラブルシュート情報が削除されます。
7. pdstbegin コマンド又は pdstend コマンドで変更した統計情報の取得が無効になり，pd_statistics 及び pdstbegin オペランドの指定値が有効になります。
8. pdchprc コマンドで変更した常駐プロセス数が無効になり，HiRDB システム定義の指定が有効になります。
9. HiRDB（HiRDB/パラレルサーバの場合はユニット）の異常終了後に再度入出力エラーが発生したため，pd_db_io_error_action ＝ unitdown の指定が無効になっている場合，unitdown の指定が有効になります。
10. 絞込み検索で使用しているリストがなくなります。

9.2.5　ほかの機能との関連

(1)　HiRDB Datareplicator との関連（HiRDB/パラレルサーバ限定）

ユニット構成又はバックエンドサーバ構成を変更すると，HiRDB Datareplicator の定義などを修正するために HiRDB Datareplicator を終了する必要があります。ユニット構成又はバックエンドサーバ構成を変更するときの手順を次に示します。

〈手順〉

1. **pdrplstop** コマンドで HiRDB Datareplicator との連携を停止します。
2. HiRDB Datareplicator の **hdestop** コマンドでデータ抽出機能を停止します。
3. pd_rpl_init_start オペランドに Y を指定している場合は N に変更します。
4. ユニット構成又はバックエンドサーバ構成を変更し，システム構成変更コマンドを実行します。ユニット構成又はバックエンドサーバ構成の変更方法については，「システム構成を変更する方法」を参照してください。
5. HiRDB Datareplicator の抽出環境を再構築します。
6. HiRDB Datareplicator の **hdestart** コマンドでデータ抽出機能を開始します。
7. **pdrplstart** コマンドで HiRDB Datareplicator との連携を開始します。

8. 3 で pd_rpl_init_start ＝ N に変更した場合は，N から Y に再変更してシステム構成変更コマンドを実行します。

(2) 系切り替え機能との関連

(a) スタンバイ型系切り替え機能の場合

現用系と実行系が同じときだけシステム構成変更コマンドを実行できます。また，システム構成変更コマンドは現用系（実行系）の HiRDB システム定義だけを対象にします。予備系の HiRDB システム定義は現用系から予備系へ HiRDB システム定義ファイルをコピーして変更してください。スタンバイ型系切り替え機能使用時の HiRDB システム定義の変更手順を次に示します。

〈手順 1〉サーバモードの場合

1. 予備系 HiRDB を終了します。
2. 現用系でシステム構成変更コマンドを実行します。
3. システム構成変更コマンドの処理終了後，現用系から予備系に HiRDB システム定義ファイルをコピーします。
4. 予備系で **pdstart** コマンド（HiRDB/パラレルサーバの場合は **pdstart -q** コマンド）を実行して予備系 HiRDB を開始します。

〈手順 2〉モニタモードの場合

1. 現用系でシステム構成変更コマンドを実行します。
2. システム構成変更コマンドの処理終了後，現用系から予備系に HiRDB システム定義ファイルをコピーします。

(b) スタンバイレス型系切り替え機能の場合

スタンバイレス型系切り替え機能使用時の HiRDB システム定義の変更手順を次に示します。なお，**代替中はシステム構成変更コマンドを使用できません。**

〈手順〉

1. 次に示す HiRDB システム定義を代替 BES ユニットの%PDDIR%¥conf¥chgconf 下にコピーします。
   ・正規 BES のバックエンドサーバ定義
   ・正規 BES ユニットのユニット制御情報定義
2. **pdstop -q -c** コマンドで代替部の待機状態を解除します。
3. システム構成変更コマンドを実行します。なお，システム構成変更コマンドを実行すると，2 で待機状態を解除した代替部は自動的に待機状態になります。

(c) 注意事項（スタンバイ型及びスタンバイレス型共通）

1 はスタンバイ型系切り替え機能使用時の注意事項で，2 はスタンバイレス型系切り替え機能使用時の注意事項です。

1. システム構成変更コマンドの実行中は，現用系と予備系で HiRDB システム定義の内容が不一致になるため，計画系切り替え及びサーバ障害による系切り替えができません。
2. システム構成変更コマンドの実行中は，正規 BES ユニットと代替 BES ユニットで，正規 BES のバックエンドサーバ定義と正規 BES ユニットのユニット制御情報定義の内容が不一致になるため，計画系切り替え及びサーバ障害による系切り替えができません。

(3)　回復不要 FES との関連（HiRDB/パラレルサーバ限定）

(a)　システム構成変更コマンドを実行できる条件

回復不要 FES を使用する場合，システム構成変更コマンドを実行できる条件が変わります。次に示す条件を一つでも満たす場合はシステム構成変更コマンドを実行できません。

- pdstart -r コマンドで HiRDB を開始した場合
- 一部のユニット又はサーバが停止中の場合（縮退起動も含む）※
- ユニット間のネットワークに通信障害が発生中の場合
- pdrplstop コマンドが実行中の場合
- pd_mode_conf オペランドの値が AUTO の場合
- システム定義の変更後，停止中のユニット内のフロントエンドサーバが回復不要 FES でなくなる場合
- インメモリ RD エリアがある場合

注※　回復不要 FES ユニットは対象外になります。

ポイント

回復不要 FES ユニットが稼働していなくてもシステム構成変更コマンドを実行できます。回復不要 FES ユニット以外のユニットはすべて稼働している必要があります。また，稼働中のユニット内のサーバがすべて稼働している必要があります。

(b)　注意事項

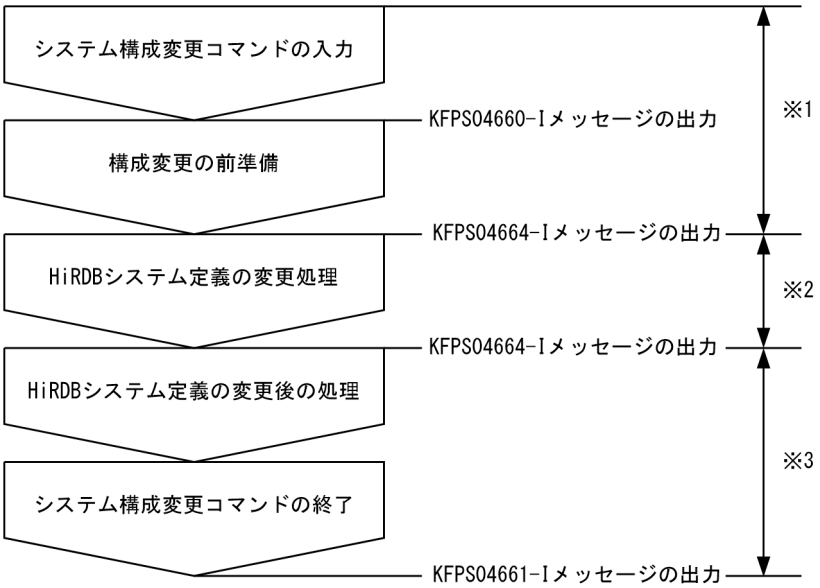
システム構成変更コマンドを実行した場合，停止中のユニット（回復不要 FES ユニット）の HiRDB システム定義も変更されます。このとき，HiRDB システム定義の変更を行うため，停止中のユニットが開始されます。

9.2.6　障害発生時の対処方法

(1)　システム構成変更コマンド実行時にエラーが発生した場合

システム構成変更コマンド実行時にエラーが発生した場合の対処方法を次の図に示します。

図 9–1　システム構成変更コマンド実行時にエラーが発生した場合の対処方法



注※1

この区間でエラーが発生した場合は「構成変更の前準備でエラーが発生した場合」を参照してください。

注※2

この区間でエラーが発生した場合は「HiRDB システム定義の変更処理でエラーが発生した場合」を参照してください。

注※3

この区間でエラーが発生した場合は「HiRDB システム定義の変更後の処理でエラーが発生した場合」を参照してください。

(a)　構成変更の前準備でエラーが発生した場合

この区間でエラーが発生した場合は，出力されたメッセージからエラー原因を調査して対処してください。また，pdls コマンドで HiRDB が停止していないかを確認してください。停止している場合は pdstart コマンドで HiRDB を開始してください。この場合，変更前の HiRDB システム定義で HiRDB が開始します。

なお，HiRDB の終了処理失敗に伴うメッセージ及びアボートコードが出力されることがあります。この場合，メッセージ及びアボートコードの対処方法に従ってください。

(b)　HiRDB システム定義の変更処理でエラーが発生した場合

この区間でエラーが発生した場合は，HiRDB 運用ディレクトリがあるディスクの使用状況や，HiRDB システム定義ファイルのアクセス権を確認してください。また，pdls コマンドで HiRDB が停止していないかを確認してください。停止している場合は，%PDDIR%¥conf¥backconf 下の変更前の HiRDB システム定義ファイルを%PDDIR%¥conf 下にコピーして回復してください。その後，pdstart コマンドで HiRDB を開始してください。

(c)　HiRDB システム定義の変更後の処理でエラーが発生した場合

この区間でエラーが発生した場合はシステム構成変更コマンドが自動的に変更前の HiRDB システム定義に戻します。出力されたメッセージからエラー原因を調査して対処してください。その後，%PDDIR%¥conf 下の HiRDB システム定義を確認してください。

また，pdls コマンドで HiRDB が停止していないかを確認してください。停止している場合は pdstart コマンドで HiRDB を開始してください。変更後の HiRDB システム定義で HiRDB を開始する場合は，%PDDIR%¥conf¥chgconf 下の HiRDB システム定義ファイルを%PDDIR%¥conf 下にコピーした後に HiRDB を開始してください。

なお，HiRDB の開始処理失敗に伴うメッセージ及びアボートコードを表示することがあります。この場合，変更後の HiRDB システム定義及び環境に問題があると考えられるため，次に示す対処をしてください。

- pdconfchk コマンドで変更後の HiRDB システム定義を確認してください。
- メッセージに従って，変更後のシステム構成で必要なリソースを確保してください。

(2)　変更後の HiRDB システム定義で HiRDB が稼働できない場合

変更後の HiRDB システム定義で HiRDB が稼働できない状態が発生した場合，自動的に変更前の定義に戻して HiRDB を稼働しようとします。変更前の HiRDB システム定義でも HiRDB が稼働しない場合，エラーメッセージに従って対処してください。

(3)　KFPS04665-W メッセージが出力された場合

KFPS04665-W メッセージが出力された場合，次に示す手順で対処してください。

〈手順〉

1. KFPS04665-W メッセージに表示されたユニットを確認します。
2. **pdls -d svr** コマンドを実行してユニットの状態を確認します。実行結果の STATUS を確認して，STOP(A)のユニットがあるか確認します。
3. STOP(A)のユニットがある場合，そのユニットを **pdstop -z** コマンドで強制終了してください。
4. KFPS04665-W メッセージに表示されたユニットの%PDDIR%¥conf 及び%PDCONFPATH%下の HiRDB システム定義ファイルを，変更後の HiRDB システム定義ファイルに置き換えます。
5. **pdstart -u** コマンドでユニットを再開始します。

9.3　HiRDB の稼働中にグローバルバッファを追加，変更，又は削除する方法（グローバルバッファの動的変更）

実行者　HiRDB 管理者

ここでは，HiRDB の稼働中に pdbufmod コマンドでグローバルバッファを追加，変更，又は削除する方法について説明します。

9.3.1　グローバルバッファの動的変更とは

HiRDB の稼働中に pdbufmod コマンドでグローバルバッファを追加，変更，又は削除できます。これを**グローバルバッファの動的変更**といいます。例えば，次に示す場合にグローバルバッファを動的変更してください。

- 追加した RD エリアにグローバルバッファを割り当てる場合
- RD エリアの割り当て先グローバルバッファを変更する場合
- グローバルバッファのチューニングの結果，グローバルバッファの定義を変更する場合

なお，グローバルバッファを動的変更するには **HiRDB Advanced High Availability** が必要になります。

(1)　動的変更したグローバルバッファの有効期限

動的変更したグローバルバッファは HiRDB を正常終了又は計画停止すると無効になります。このため，HiRDB を終了したときに，動的変更したグローバルバッファを pdbuffer オペランドで定義してください。なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定値を変更できます。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

また，HiRDB を正常終了又は計画停止する前に，**pdbufls -k def** コマンドの実行結果（グローバルバッファの定義情報）を保存してください。それを参考にして pdbuffer オペランドを修正してください。

(2)　準備作業

グローバルバッファを動的変更するには次に示す準備が必要です。

(a)　HiRDB システム定義の指定

1. pd_dbbuff_modify オペランドに Y を指定してください。
2. pd_dbbuff_attribute オペランドに fixed を指定した場合，この機能で追加又は変更したグローバルバッファが使用する共用メモリも実メモリ上に固定されます。そのため，実メモリサイズを十分考慮してグ

ローバルバッファを追加又は変更してください。実メモリサイズに余裕がない場合は pd_dbbuff_attribute オペランドを省略するか，又は free を指定してください。

3. SHMMAX オペランドの指定値を見積もり直してください。

(b)　メモリに関する準備作業

1. 次に示す共用メモリが増加するため，共用メモリの必要量を見積もり直してください。共用メモリの見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
   - グローバルバッファが使用する共用メモリ
   - シングルサーバが使用する共用メモリ
   - ディクショナリサーバが使用する共用メモリ
   - バックエンドサーバが使用する共用メモリ
2. ステータスファイルの必要量が増加するため，ステータスファイルの容量を見積もり直してください。ステータスファイルの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(3)　注意事項

(a)　ほかの機能との関連

高速系切り替え機能又はスタンバイレス型系切り替え機能を使用する場合は，グローバルバッファの動的変更機能を使用できません。これらの機能を使用する場合は，システム構成変更コマンド（pdchgconf コマンド）で pdbuffer オペランドの指定値を変更してグローバルバッファを割り当ててください。pdchgconf コマンドを使用すると，HiRDB の稼働中に HiRDB システム定義を変更できます。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(b)　排他制御

グローバルバッファの動的変更時（pdbufmod コマンドの実行時）は，関連する RD エリアに対して排他モード（EX）で排他が掛かります。このため，関連する RD エリアにアクセスするトランザクションは待ち状態になります。

(c)　グローバルバッファ数の上限

グローバルバッファ数の上限は，pd_max_add_dbbuff_no オペランド，及び pd_max_add_dbbuff_shm_no オペランドで設定します。

pdbufls -k def コマンドの実行結果で表示されるグローバルバッファを数えることで，グローバルバッファ数を確認できます。HiRDB/パラレルサーバの場合はサーバ名（ヘッダ名：SVID）が一致するグローバルバッファを数えてください。

(d) グローバルバッファ用の共用メモリセグメント数の上限

グローバルバッファを動的変更（追加及びメモリ所要量が増加する変更）すると，新規の共用メモリセグメントを確保し，動的変更したグローバルバッファをその共用メモリセグメントに割り当てます。この共用メモリセグメントは，pd_max_add_dbbuff_no オペランド，及び pd_max_add_dbbuff_shm_no オペランドで設定します。なお，HiRDB/パラレルサーバの場合は 1 サーバ当たりで設定します。pdls -d mem コマンドの実行結果で表示される共用メモリセグメントを数えることで，共用メモリセグメント数を確認できます。HiRDB/パラレルサーバの場合はサーバ名（ヘッダ名：SHM-OWNER）が一致するグローバルバッファを数えてください。

共用メモリセグメント数が上限を超えた場合は，動的変更したグローバルバッファの定義情報を pdbuffer オペランドに反映してください。その後，HiRDB を正常開始すると共用メモリセグメント数を減らせます。

なお，動的変更したグローバルバッファが使用する共用メモリサイズが SHMMAX オペランドの値より大きい場合，グローバルバッファが使用する共用メモリが複数の共用メモリセグメントに確保されます。確保される共用メモリセグメント数は次に示す計算式から求められます。

↑動的変更したグローバルバッファが使用する共用メモリサイズ÷SHMMAX オペランドの値↑

グローバルバッファが使用する共用メモリについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

9.3.2 運用例

pdbufmod コマンドでグローバルバッファを追加，変更，又は削除します。

(1) 例題 1（グローバルバッファの追加）

グローバルバッファ（gbuf01）を追加して，そのグローバルバッファに追加した RD エリア（RDAREA1）を割り当てます。

```
pdbufmod -k add -a gbuf01 -r RDAREA1 -n 1000
```

〔説明〕

-k add：グローバルバッファを追加する場合に指定します。

-a：追加するグローバルバッファの名称を指定します。

-r：割り当てる RD エリアを指定します。

-n：グローバルバッファのバッファ面数を指定します。

(2) 例題 2（グローバルバッファの追加，及び RD エリアの割り当て先グローバルバッファの変更）

グローバルバッファ（gbuf02）を追加します。同時に，RD エリア（RDAREA1）の割り当て先グローバルバッファを gbuf01 から gbuf02 に変更します。

```
pdbufmod -k add -a gbuf02 -r RDAREA1 -n 1000
```

〔説明〕

-k add：グローバルバッファを追加する場合に指定します。

-a：追加するグローバルバッファの名称を指定します。

-r：割り当てる RD エリアを指定します。

-n：グローバルバッファのバッファ面数を指定します。

(3) 例題 3（RD エリアの割り当て先グローバルバッファの設定）

既存のグローバルバッファ（gbuf01）に，追加した RD エリア（RDAREA1）を割り当てます。

```
pdbufmod -k add -a gbuf01 -r RDAREA1
```

〔説明〕

-k add：既存のグローバルバッファに RD エリアを割り当てる場合に指定します。

-a：割り当て先のグローバルバッファの名称を指定します。

-r：割り当てる RD エリアを指定します。

(4) 例題 4（RD エリアの割り当て先グローバルバッファの変更）

RD エリア（RDAREA1）の割り当て先グローバルバッファを gbuf01 から gbuf02 に変更します。gbuf01 及び gbuf02 は既存のグローバルバッファとします。

```
pdbufmod -k add -a gbuf02 -r RDAREA1
```

〔説明〕

-k add：既存のグローバルバッファに RD エリアを割り当てる場合に指定します。

-a：割り当て先のグローバルバッファの名称を指定します。

-r：割り当てる RD エリアを指定します。

(5) 例題 5（グローバルバッファの削除）

RD エリアを割り当てていないグローバルバッファ（gbuf01）を削除します。

```
pdbufmod -k del -a gbuf01
```

〔説明〕

-k del：グローバルバッファを削除する場合に指定します。

-a：削除するグローバルバッファの名称を指定します。

(6)　例題 6（RD エリアを割り当てているグローバルバッファの削除）

RD エリア（RDAREA1）を割り当てているグローバルバッファ（gbuf01）を削除します。

```
pdhold -r RDAREA1 -c                                1
pdbufmod -k del -a gbuf01                           2
```

〔説明〕

1. RD エリアを閉塞クローズ状態にします。ただし，gbuf01 がインデクス用グローバルバッファの場合はインデクス格納 RD エリアを閉塞クローズ状態にします。
2. グローバルバッファを削除します。

(7)　例題 7（RD エリアをグローバルバッファから切り離す）

RD エリア（RDAREA1）をグローバルバッファ（gbuf01）から切り離します。

```
pdhold -r RDAREA1 -c                                1
pdbufmod -k del -r RDAREA1                          2
```

〔説明〕

1. RD エリアを閉塞クローズ状態にします。
2. RD エリアをグローバルバッファから切り離します。

(8)　例題 8（グローバルバッファの定義変更）

グローバルバッファ（gbuf01）のバッファ面数を 1000 から 2000 に変更します。

```
pdbufmod -k upd -a gbuf01 -n 2000
```

〔説明〕

-k upd：グローバルバッファの定義を変更する場合に指定します。

-a：定義を変更するグローバルバッファの名称を指定します。

-n：変更後のバッファ面数を指定します。

なお，pdbufmod コマンドで次に示す項目を変更できます。

- バッファ面数
- バッファサイズ

- 同時実行最大プリフェッチ数
- 一括入力最大ページ数
- デファードライトトリガ時の更新ページ出力比率

9.4　サーバプロセス数を変更する方法

9.4.1　サーバプロセス数の変更方法

ここでは，サーバプロセス数の変更方法について説明します。説明する項目は次のとおりです。

- サーバプロセス数を指定するオペランド
- サーバプロセス数を変更する方法
- サーバプロセス数を変更するときの運用例
- 注意事項（HiRDB/パラレルサーバ限定）

(1)　サーバプロセス数を指定するオペランド

(a)　最大起動プロセス数を指定するオペランド

HiRDB が起動するサーバプロセスの最大数（最大起動プロセス数）は，次の表に示す HiRDB システム定義のオペランドによって決まります。

表 9–2　最大起動プロセス数を指定するオペランド

サーバの種類	指定する定義	オペランド名
シングルサーバ	システム共通定義	pd_max_users
フロントエンドサーバ	システム共通定義	pd_max_users
ディクショナリサーバ	ディクショナリサーバ定義，又はサーバ共通定義	pd_max_dic_process※1
バックエンドサーバ	バックエンドサーバ定義，又はサーバ共通定義	pd_max_bes_process※2

注1

マルチフロントエンドサーバの場合，ディクショナリサーバやバックエンドサーバには複数のフロントエンドサーバから処理要求が送られます。したがって，システム共通定義の pd_max_users オペランドの指定値を超える処理が集中することがあります（最大で pd_max_users オペランドの指定値×フロントエンドサーバ数の処理が集中します）。マルチフロントエンドサーバの場合は，処理の集中度に見合った最大プロセス数を算出し，pd_max_dic_process オペランド及び pd_max_bes_process オペランドに指定してください。

注2

マルチフロントエンドサーバでない場合も，RD エリアやグローバルバッファに関連する運用コマンド（pdbufls，pddbls，pdopen，pdclose，pdhold，又は pdrels）を pd_max_users の指定値以上同時に実行すると，ディクショナリサーバに pd_max_users の指定値を超える処理が集中することがあります。この場合も，処理の集中度に見合った最大プロセス数を pd_max_dic_process オペランドに指定してください。

注※1

pd_max_dic_process オペランドを省略した場合は，システム共通定義の pd_max_users オペランドの値を仮定します。

注※2

pd_max_bes_process オペランドを省略した場合は，システム共通定義の pd_max_users オペランドの値を仮定します。

(b)　常駐プロセス数を指定するオペランド

HiRDB を開始するときに生成するサーバプロセス数（常駐プロセス数）は，**pd_process_count** オペランドの値によって決まります。

処理要求の集中度が高いシステムでは，常駐プロセス数を大きめの値に設定すれば，処理をすぐに始められます。常駐プロセス数を少ない値で設定すると，常駐プロセス数よりも多い処理要求が集中した場合に，HiRDB は最大起動プロセス数を上限としてサーバプロセスを起動し要求に応じますが，プロセス起動の分だけ処理開始が遅くなります。ただし，常駐プロセス数を大きく設定すると，常にサーバプロセスが起動しているため，数に応じてメモリ占有量が大きくなります。そのため，常駐プロセス数の設定は，処理の集中度とメモリ占有量を考慮してするようにしてください。

(2)　サーバプロセス数を変更する方法

サーバプロセス数を変更するには次に示す方法があります。

- pdchprc コマンドで変更する
- HiRDB システム定義で変更する

(a)　pdchprc コマンドで変更する方法

pdchprc コマンドで，最大起動プロセス数及び常駐プロセス数を変更できます。ただし，pdchprc コマンドでのプロセス数の変更は一時的なものです。HiRDB が終了するか，又は pdchprc コマンドが実行されるまでの間だけ有効です。

(b)　HiRDB システム定義で変更する方法

表「最大起動プロセス数を指定するオペランド」に示すオペランドと pd_process_count オペランドの値を変更してください。なお，表「最大起動プロセス数を指定するオペランド」に示すオペランドの値を変更する場合は，正常終了又は計画停止させてください。HiRDB を強制終了，又は異常終了した後に表「最大起動プロセス数を指定するオペランド」に示すオペランドの値を変更すると，HiRDB の再開始時にエラーとなり，HiRDB が再開始できないことがあります。ただし，pd_process_count オペランドの値だけを変更する場合，終了モードは関係ありません。

システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できます。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

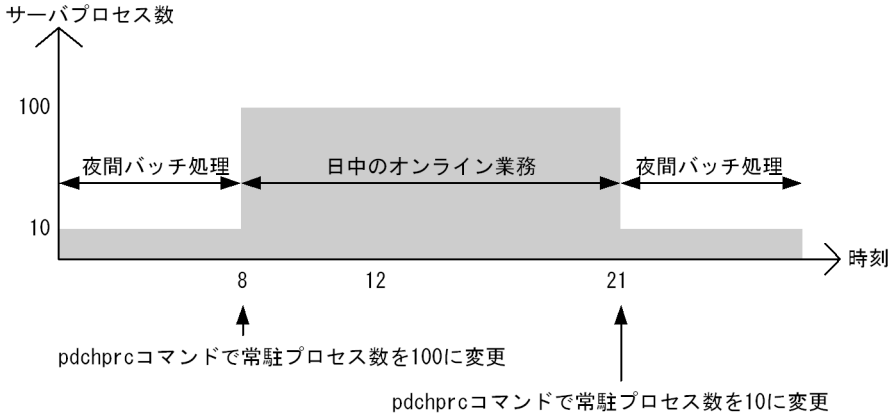
(3)　サーバプロセス数を変更するときの運用例

pdchprc コマンドで，HiRDB 稼働中に最大起動プロセス数及び常駐プロセス数を一時的に変更できます。運用例を次に示します。

(a)　バッチ処理でサーバプロセス数を減少させる運用

オンライン業務終了後，夜間バッチ処理をするときに，サーバプロセスの常駐プロセス数を一時的に減少させます。夜間バッチ処理では，オンライン業務のように同時に処理要求が発生することは少ないと考えられます。そのため，オンライン業務中のようにサーバプロセスを多数起動しておく必要はなく，夜間バッチ処理が必要とする分だけ，サーバプロセスを起動しておけばよいことになります。バッチ処理でサーバプロセス数を減少させる運用を次の図に示します。

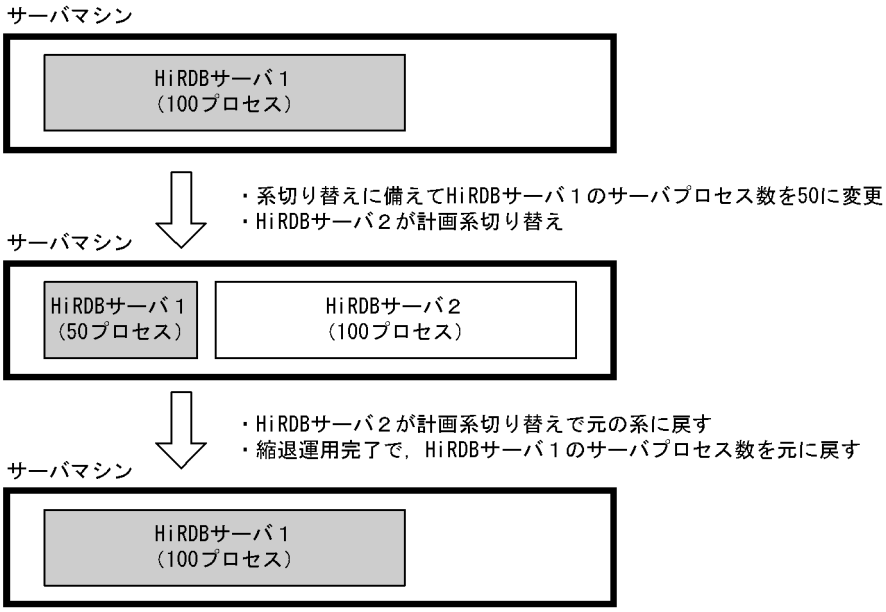
図 9–2　バッチ処理でサーバプロセス数を減少させる運用



(b)　相互系切り替え環境でのシステム縮退運用

相互系切り替え環境では，系切り替えが発生すると一つのサーバマシンで複数の HiRDB を運用する事態が発生し，一時的にメモリ占有量が拡大します。この場合，pdchprc コマンドを使用して HiRDB のサーバプロセス数を少なくするように設定すると，メモリ占有量の拡大を最小限に抑え込めます。相互系切り替え環境でのシステム縮退運用を次の図に示します。

図 9–3　相互系切り替え環境でのシステム縮退運用



(4)　注意事項（HiRDB/パラレルサーバ限定）

pdchprc コマンドで，サーバプロセスの最大数を０にするときは注意してください。システムマネジャと同じユニットに配置したフロントエンドサーバのサーバプロセスの最大数を０にすると，次に示すユティリティが実行できなくなります。

- データベース構成変更ユティリティ
- データベース作成ユティリティ
- データベース再編成ユティリティ
- データベース状態解析ユティリティ
- 最適化情報収集ユティリティ

9.5　ユーザ数を増やすときにすること

9.5.1　ユーザ数が増えた場合に見直す項目

ここでは，ユーザ数が増えた場合に見直す項目について説明します。

(1)　HiRDB システム定義のオペランド

ユーザ数が増えた場合，次に示すオペランドの指定値を見直す必要があります。

- pd_max_users
- pd_max_server_process※
- pd_utl_exec_mode
- pd_system_complete_wait_time
- pd_process_terminator_max
- pd_queue_watch_time
- pd_process_count
- pd_lck_pool_size
- pd_fes_lck_pool_size
- pd_sql_object_cache_size
- pd_table_def_cache_size
- pd_view_def_cache_size

注※

pd_max_server_process オペランドを省略すると，このオペランドの値を HiRDB が自動計算します。

(2)　HiRDB が使用する共用メモリ

HiRDB が使用する共用メモリ（mapped file）が増加します。そのため，%PDDIR%下のファイルシステムを圧迫します。その場合，KFPO00113-E メッセージでエラーが通知されるのでメッセージを基に対策してください。

- errno=12 の場合は，メモリを増設するか又はほかのプログラムを停止する必要があります。

共用メモリ及びセマフォに関する数値が実際にどのくらいになるかは，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(3)　HiRDB が使用するポート数

ユーザ数が増えた場合，HiRDB が使用するポート数が増加します。ポート数が不足すると，処理が中断したり，ほかのプログラムの通信処理に影響を与えたりします。OS が自動的に割り当てるポート数は OS によって異なるため，各サーバマシンのネットワークに関する設定を確認してください。HiRDB が使用するポート数については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

OS が自動的に割り当てるポート数で足りない場合は，pd_registered_port オペランドで HiRDB が使用するポート番号を範囲指定してください。詳細については，「通信処理で使用するポート番号を範囲指定する方法」を参照してください。

(4)　排他制御用プールサイズ

ユーザ数が増えた場合，排他要求数が増加します。その結果，排他制御用プールサイズの不足によるエラーが発生することがあります（KFPS00443-I メッセージが出力されます）。この場合，pd_lck_pool_size オペランドの指定値を見直してください。

なお，排他制御用プールサイズ不足がフロントエンドサーバで発生した場合は，pd_fes_lck_pool_size オペランドの指定値を見直してください。排他制御用プールサイズ不足がどのサーバで発生したかは，KFPS00443-I メッセージに表示されます。

(5)　ステータスファイルの容量

ステータスファイルの容量が増加します。ステータスファイルの容量見積もりは，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

容量見積もりの後に，ステータスファイルを再作成してください。

ステータスファイルを再作成する場合，pdstsinit コマンドの-c オプションの指定値（レコード数）を大きくしてください。

なお，ユニット用ステータスファイル，及びサーバ用ステータスファイルの両方とも再作成してください。

(6)　HiRDB 購入時の契約ユーザ数を超える場合

HiRDB 購入時の契約ユーザ数を超える場合は，新たに HiRDB アップグレードが必要となります。

9.6 HiRDB に接続できないクライアントが発生するときの対処方法（クライアントグループの接続枠保証機能）

9.6.1 クライアントグループの接続枠保証機能の概要

pd_max_users で指定した数（最大同時接続数）のクライアントが HiRDB に接続していると，新たなクライアントは HiRDB に接続できなくなります。このため，重要な業務を実施するクライアントが HiRDB に接続できないことがあります。このような場合，**クライアントグループの接続枠保証機能**を使用してください。

(1) クライアントグループの接続枠保証機能とは

HiRDB に接続するクライアントを次に示すようにグループ化して，各クライアントグループの HiRDB への接続枠を保証できます。

X/Open XA インタフェースのクライアントグループ（XA）

X/Open XA インタフェースで HiRDB にアクセスするクライアントのグループです。クライアントが PC 又は WS であっても，X/Open XA インタフェースで HiRDB にアクセスすると，X/Open XA インタフェースのクライアントグループとなります。

PC クライアントのクライアントグループ（PC）

Windows 系及び Linux 系クライアントのグループです。

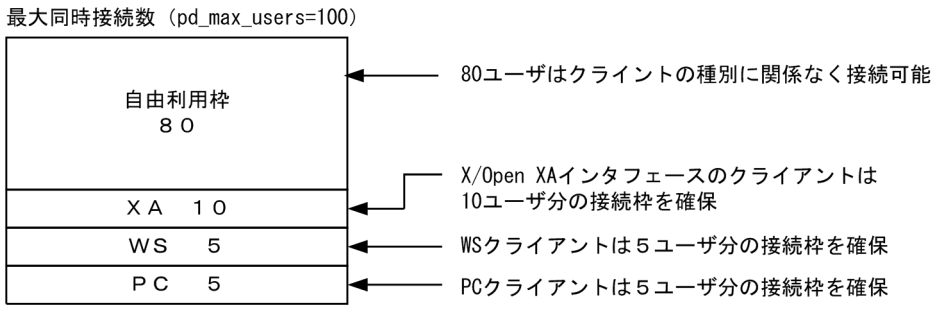
WS クライアントのクライアントグループ（WS）

UNIX 系クライアントのグループです。

注 （ ）内の文字列はクライアントグループの名称です。

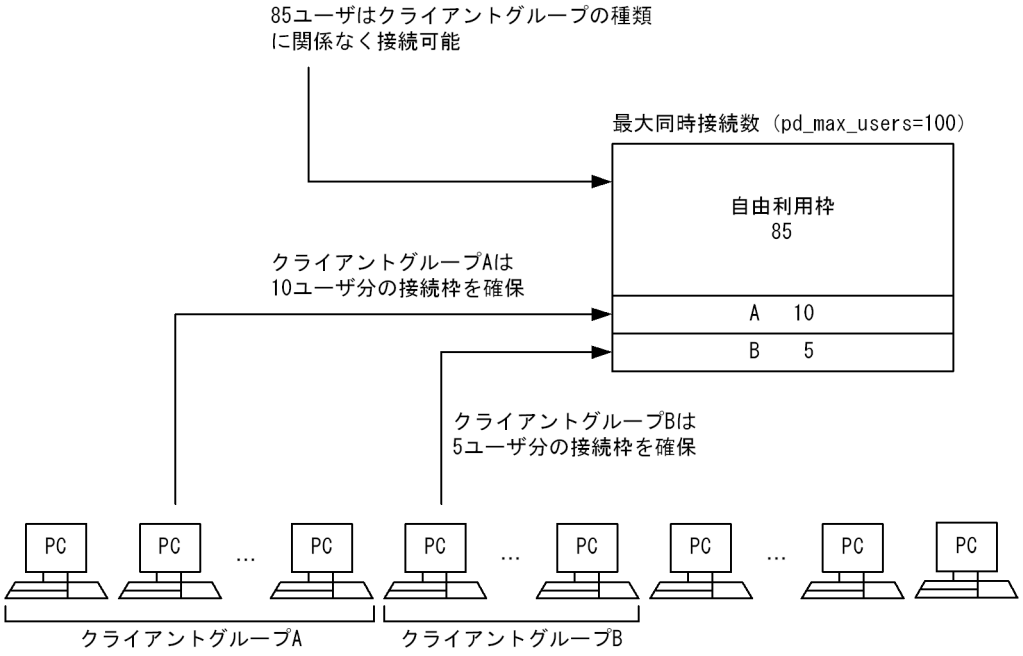
接続枠を指定したクライアントグループに所属するクライアントは，ほかのクライアントグループのクライアントによる HiRDB へのアクセスが集中したときでも，接続最低保証数までは HiRDB に確実に接続できます。クライアントグループの接続枠保証機能を次の図に示します。

図 9–4 クライアントグループの接続枠保証機能



(2)　ユーザが任意にクライアントグループを決められます

（1）で説明したクライアントグループは，HiRDB が決めているクライアントグループです。このほかにユーザが任意にクライアントグループを決められます。ユーザが任意に決めたクライアントグループは，10 グループまで指定できます。



(3)　環境設定方法

クライアントグループの接続枠保証機能を使用する場合は，次に示すオペランドを指定します。

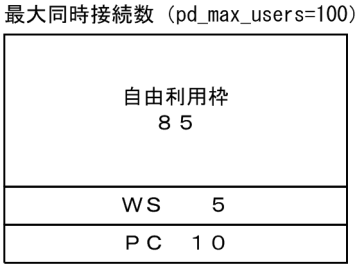
- システム共通定義の **pdcltgrp** オペランド
- クライアント環境定義の **PDCLTGRP** オペランド※

注※　ユーザ任意のクライアントグループを定義する場合に指定します。

環境設定方法を例題形式で説明します。

例題１

WS クライアント５ユーザ分，PC クライアント 10 ユーザ分の接続枠を保証するようにします。
なお，pd_max_users=100 を指定しているとします。



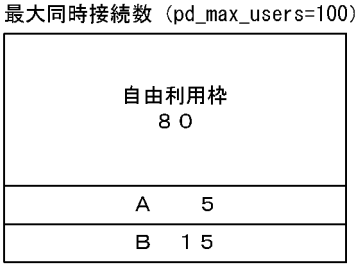
●システム共通定義の指定

```
pdcltgrp -g WS -u 5
pdcltgrp -g PC -u 10
```

例題２

ユーザ任意のクライアントグループを設定し，グループＡは５ユーザ分，グループＢは 15 ユーザ分の接続枠を保証するようにします。

なお，pd_max_users=100 を指定しているとします。



●システム共通定義の指定

```
pdcltgrp -g A -u 5
pdcltgrp -g B -u 15
```

●グループＡのクライアント環境定義の指定

```
PDCLTGRP = A
```

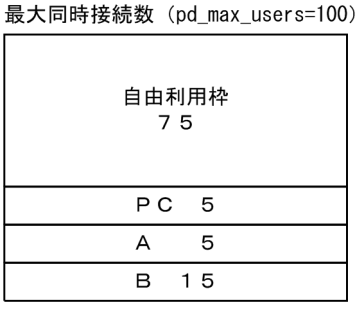
●グループ B のクライアント環境定義の指定

```
PDCLTGRP = B
```

例題３

PC クライアント 5 ユーザ分の接続枠を保証するようにします。また，ユーザ任意のクライアントグループを設定し，グループＡは５ユーザ分，グループＢは 15 ユーザ分の接続枠を保証するようにします。

なお，pd_max_users=100 を指定しているとします。



●システム共通定義の指定

```
pdcltgrp -g PC -u 5
pdcltgrp -g A -u 5
pdcltgrp -g B -u 15
```

●グループＡのクライアント環境定義の指定

```
PDCLTGRP = A
```

●グループ B のクライアント環境定義の指定

```
PDCLTGRP = B
```

参考

pdcltgrp オペランドと PDCLTGRP オペランドの指定値が合っていない場合，ユーザ任意のクライアントグループは無視されます。

(4) 注意事項

1. 自由利用枠を小さく設定しないでください。小さく設定すると，ユティリティ又は UAP が接続ユーザ数オーバになる確率が高くなります。
2. pdcltgrp オペランドの-u オプションで指定した接続保障ユーザ数の合計が pd_max_users オペランドの値以上にならないようにしてください。pd_max_users オペランドの値以上になった場合，HiRDB 開始処理が中止します。
3. ユティリティ（データベース定義ユティリティを除く）は，クライアントグループ化できません。
4. X/Open XA インタフェースのクライアントグループ（XA）を指定するには，HiRDB クライアントのバージョンが次に示すどちらかの条件を満たす必要があります。
   - 04-05（05-00 及び 05-01 では指定できません）
   - 05-02 以降
5. PC クライアントのクライアントグループ（PC）又は WS クライアントのクライアントグループ（WS）を指定するには，HiRDB クライアントのバージョンが次に示す条件を満たす必要があります。
   - 04-00 以降

(5) リソースの使用率の監視との関連

pd_max_users の使用率の監視をしている場合※，クライアントグループの接続枠保証機能を使用すると，pd_max_users オペランドの使用率のほかに自由利用枠の使用率についても監視を適用します。ただし，自由利用枠が 10 ユーザ以上あることが条件となります。

注※

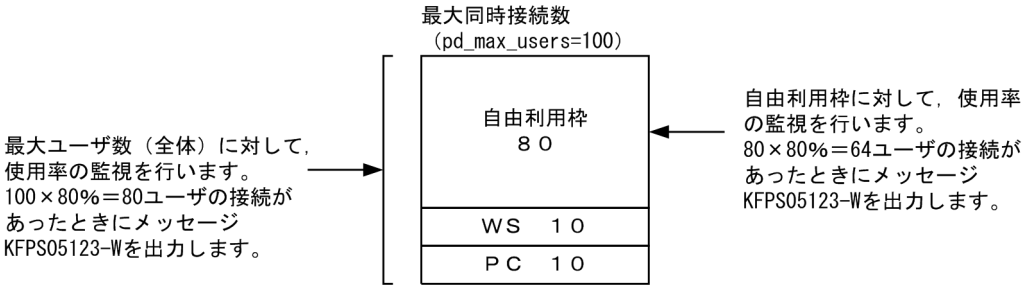
次に示すどちらかのオペランドを指定している場合です。

- pd_watch_resource = AUTO
- pd_max_users_wrn_pnt

次に，pd_max_users の使用率の監視をしている場合に，エラーメッセージが出力されるタイミングの例を示します。

（例）

pd_max_users の使用率が 80%になったときにメッセージを出力する指定をしています。



9.7　通信処理で使用するポート番号を範囲指定する方法

9.7.1　ポート番号を範囲指定する方法及び注意事項

HiRDB サーバと HiRDB クライアント間，又は HiRDB サーバ間の通信処理で使用するポート番号は，OS が自動的に割り当てています。通信処理が大量に発生すると，ポート番号の不足によって処理が中断したり，又はほかのプログラムの通信処理に影響を及ぼしたりします。これを防ぐために，HiRDB が通信処理で使用するポート番号を範囲指定できます。

なお，通信処理で使用するポート番号が少ない場合には，この機能を使用する必要はありません。また，この機能はサーバ間通信に該当します。HiRDB 下のすべてのプロセスに有効となるものではありません。コマンドなどの通信処理では OS が割り当てるポート番号を使用します。

(1)　HiRDB が使用するポート数

HiRDB が使用するポート数については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　ポート番号を範囲指定するには

pd_registered_port オペランドでポート番号を範囲指定します。

(3)　注意事項

pd_registered_port オペランドで範囲指定するポート番号は，次に示すポート番号と重複しないようにしてください。重複すると，HiRDB を開始できないなどの不具合が発生することがあります。

1. pd_name_port オペランドで指定したポート番号
2. pd_service_port オペランドで指定したポート番号
3. pdunit オペランドの-p オプションで指定したポート番号
4. pd_crypto_service_port オペランドで指定したポート番号
5. **%windir%¥system32¥drivers¥etc¥services** に登録されているポート番号
6. クライアント環境定義の PDCLTRCVPORT オペランドで指定したポート番号（HiRDB クライアントが HiRDB サーバと同一サーバマシン上に存在する場合）
7. ほかのプログラムで使用しているポート番号

pd_registered_port_check オペランドに Y，C，又は W を指定すると，5 のポート番号の重複チェックを HiRDB が行います。このオペランドに N を指定した場合は，ポート番号の重複チェックを HiRDB は行いません。この場合，HiRDB 管理者がポート番号の重複チェックを行う必要があります。

ポイント

- マルチ HiRDB 環境の場合，各 HiRDB のポート番号の予約範囲（pd_registered_port オペランドの指定範囲）が重複しないようにしてください。
- 系切り替え機能を使用していて，一つのサーバマシンに複数のユニットが存在する場合は，システム共通定義ではなく，ユニット制御情報定義の pd_registered_port オペランドを指定してください。そのとき，各ユニットのポート番号の予約範囲が重複しないようにしてください。
- pd_registered_port_level オペランドで HiRDB 予約ポート機能の対象範囲を指定できます。詳細については，マニュアル「HiRDB システム定義」を参照してください。

9.8　ホスト名を変更する方法

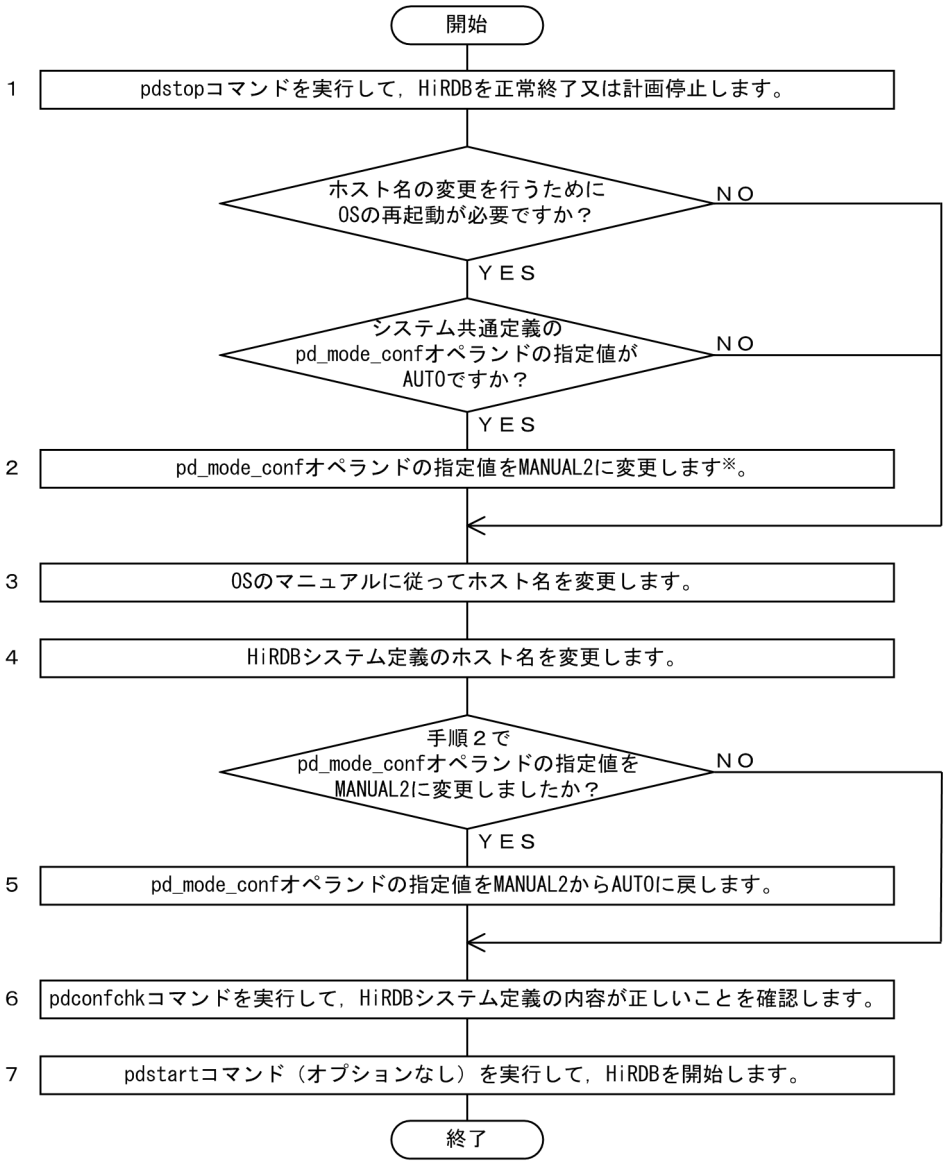
9.8.1　HiRDB の稼働中にホスト名を変更する手順

実行者　HiRDB 管理者

ここでは，HiRDB の稼働中にホスト名（コンピュータ名）を変更する方法について説明します。

HiRDB の稼働中にホスト名を変更する場合は，次の手順に従ってください。

〈手順〉



注※

指定値が AUTO のままの場合，OS の再起動が行われると HiRDB も自動的に開始しますが，HiRDB システムが起動に失敗します。

NetBackup 連携機能使用時に標準ホスト名（コンピュータ名）を変更する場合

NetBackup 連携機能を使用していて，HiRDB の稼働中に標準ホスト名（コンピュータ名）を変更する場合は，NetBackup に添付されているドキュメントを参照して，NetBackup サーバ，及び NetBackup クライアントの設定を行ってください。また，HiRDB では，次のどちらかを実施する必要があります。

1. 標準ホスト名（コンピュータ名）を変更した後，初回の HiRDB 開始直後にバックアップを取得し直します。
2. 1.が実施できない場合で，標準ホスト名（コンピュータ名）を変更する前のバックアップを使用して回復を行うときは，「NetBackup 連携機能を使用する場合の注意事項」の環境設定，及び運用を実施します。

9.9　コマンドのデッドロックプライオリティ値を変更する方法

ここでは，次に示すコマンドのデッドロックプライオリティ値を変更する方法について説明します。

- pdhold -b（バックアップ閉塞）

9.9.1　コマンドのデッドロックプライオリティ値

コマンドのデッドロックプライオリティ値は通常 UAP より低いため，デッドロックが発生した場合，コマンドがエラーになります。コマンドをエラーにしたくない場合は，UAP よりコマンドのデッドロックプライオリティ値を高くしてください。なお，コマンドのデッドロックプライオリティ値を高くした場合，デッドロックが発生するとトランザクションがエラーになります。

(1)　デッドロックプライオリティ値を変更できるコマンド

次に示すコマンドのデッドロックプライオリティ値を変更できます。

- pdhold -b（バックアップ閉塞）

(2)　デッドロックプライオリティ値の一覧

デッドロックプライオリティ値の一覧を次の表に示します。

表 9–3　デッドロックプライオリティ値の一覧

<table>
<tr><th colspan="4">種別及び条件</th><th>デッドロックプライオリティ値</th></tr>
<tr><td rowspan="5">UAP</td><td rowspan="5">クライアント環境定義の PDDLKPRIO の値</td><td colspan="2">96</td><td>96</td></tr>
<tr><td colspan="2">64</td><td>64</td></tr>
<tr><td colspan="2">32</td><td>32</td></tr>
<tr><td rowspan="2">省略</td><td>X/Open XA インタフェースを使用する場合</td><td>96</td></tr>
<tr><td>X/Open XA インタフェースを使用しない場合</td><td>64</td></tr>
<tr><td colspan="4">ユティリティ</td><td>64</td></tr>
<tr><td rowspan="2">コマンド</td><td colspan="3">pdhold -b</td><td>pd_command_deadlock_priority の値※</td></tr>
<tr><td colspan="3">上記以外のコマンド</td><td>64</td></tr>
</table>

注※

このオペランドにデッドロックプライオリティ値として 32，64，96，120 のどれかを指定できます。このオペランドを省略した場合，排他資源の種別によってデッドロックプライオリティ値が次のとおりに設定されます。

- 64 に設定される排他資源
  ディクショナリ表（DICT）
- 120 に設定される排他資源
  RD エリア（RDAR）
  RD エリア増分（RDLF）
  RD エリア状態（RDAS）
  バックアップ閉塞（HOLD）

参考

- デッドロックプライオリティ値が小さい方が処理の優先度が高くなります。デッドロックプライオリティ値が大きくなるほど優先度が低くなります。
- デッドロックプライオリティ値が同じ場合は，先に実行したトランザクションが優先されます。

9.9.2　環境設定

コマンドのデッドロックプライオリティ値を変更する場合，次に示すオペランドを指定してください。

- **pd_deadlock_priority_use**
  コマンドのデッドロックプライオリティ値を変更するかどうかを指定します。変更する場合は Y を指定します。
- **pd_command_deadlock_priority**
  コマンドのデッドロックプライオリティ値を指定します。指定できる値は 32，64，96，120 のどれかです。

9.9.3　運用方法

(1)　コマンドのデッドロックプライオリティ値を変更するときの運用手順

コマンドのデッドロックプライオリティ値を変更するときの運用手順を次に示します。

〈手順〉

1. デッドロックを発生する可能性があるトランザクションのデッドロックプライオリティ値（クライアント環境定義の PDDLKPRIO オペランドの値）を確認します。
2. pd_deadlock_priority_use オペランドに Y を指定します。
3. pd_command_deadlock_priority オペランドにコマンドのデッドロックプライオリティ値を指定します。

ポイント

pd_command_deadlock_priority オペランドの指定値の目安を次に示します。

- デッドロック発生時にトランザクションをエラーにする場合は，オペランドの指定値を次に示す条件を満たすようにしてください。

  pd_command_deadlock_priority の値＜ PDDLKPRIO の値

- デッドロック発生時にコマンドをエラーにする場合は，オペランドの指定値を次に示す条件を満たすようにしてください。

  pd_command_deadlock_priority の値＞ PDDLKPRIO の値

(2)　オペランドの指定例

次に示す運用をするときのオペランドの指定例を説明します。

- UAP1 のトランザクションとコマンドでデッドロックが発生した場合，コマンドをエラーにする
- UAP2 のトランザクションとコマンドでデッドロックが発生した場合，UAP2 をエラーにする

このときのオペランドの指定例を次に示します。

UAP1 を実行するクライアント環境定義の指定

```
PDDLKPRIO 32
```

UAP2 を実行するクライアント環境定義の指定

```
PDDLKPRIO 96
```

システム共通定義の指定

```
pd_deadlock_priority_use = Y
pd_command_deadlock_priority = 64
```

9.10　排他制御用プールを分割する方法

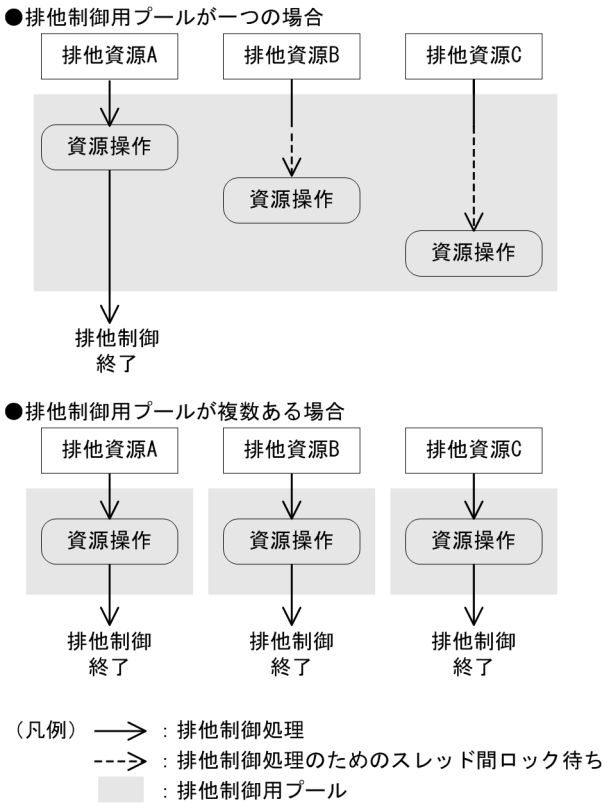
9.10.1　排他制御用プールの分割の概要

通常，排他制御の処理はサーバごとに一元管理されており，排他制御用プールは一つです。そのため，トランザクションの多重実行数と稼働率が高いシステムでは，排他制御処理のためのスレッド間ロック待ち数がトランザクションの常時多重実行数と同じになり，排他制御の操作に時間が掛かってしまいます。

この状態は，排他制御用プールを複数に分割して，排他制御の処理を分散させることで回避できます。排他制御用プールを分割したものを排他制御用プールパーティションといいます。

それぞれの排他資源が，異なる排他制御用プールパーティションを使用すると，排他制御処理が分散され，排他制御処理のためのスレッド間ロック待ち数が少なくなる場合があります。これによって，排他制御に掛かる時間が減るため，SQL の実行時間が短縮できます。排他制御処理の分散の例を次の図に示します。

図 9–5　排他制御処理の分散



[説明]

排他制御用プールが一つの場合は，排他資源 A の排他操作が終わるまで，排他資源 B 及び C の排他操作は待たされます。一方，排他制御用プールが複数ある場合は，排他資源 A，排他資源 B，排他資源 C それぞれの排他操作が同時に行われます。

ただし，特定の排他資源に対して処理が集中した場合は，排他制御処理は分散されません。この場合，排他制御用プールパーティションのスレッド間ロック待ち数は変わらないため，SQL の実行時間は短縮できません。

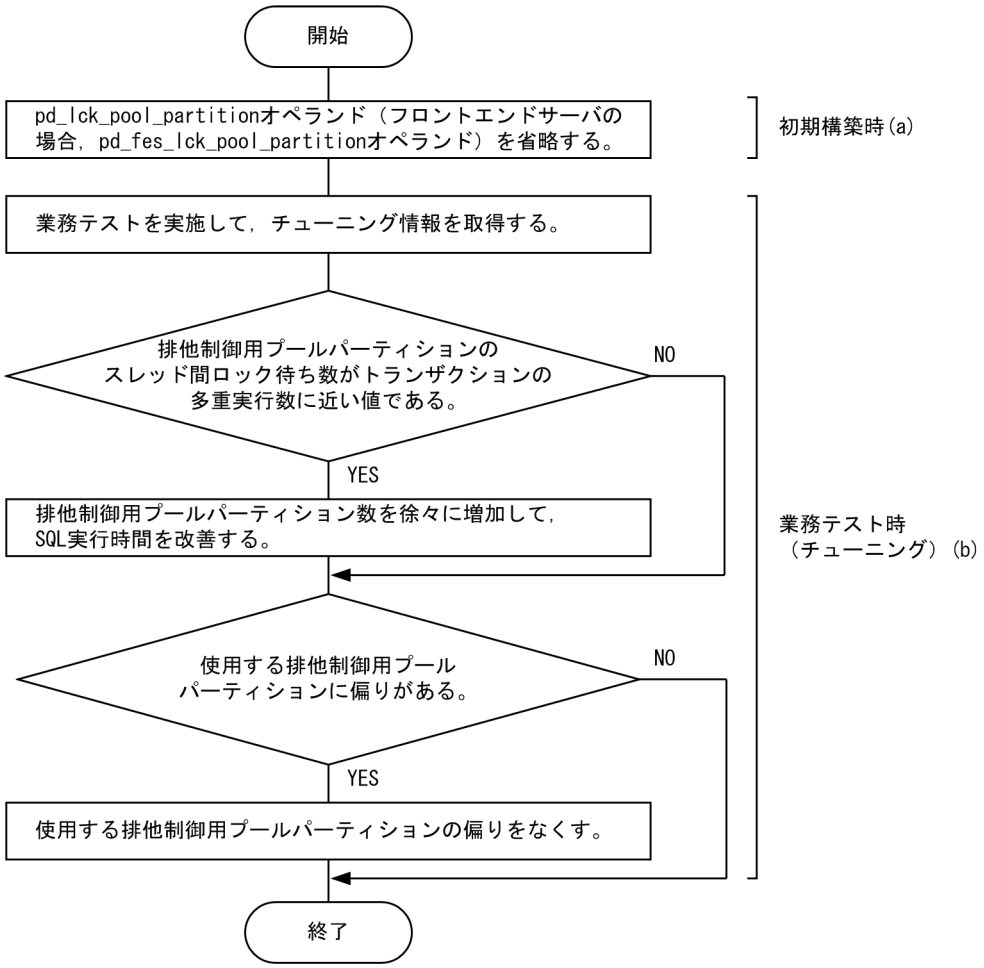
(1) 指定方法

pd_lck_pool_partition オペランド（フロントエンドサーバの場合は pd_fes_lck_pool_partition オペランド）に排他制御用プールの分割数を指定します。これによって，排他制御処理を分散させることができます。

pd_lck_pool_partition オペランド及び pd_fes_lck_pool_partition オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

指定値を決定する手順を次の図に示します。

図 9–6 pd_lck_pool_partition オペランドの指定値を決定する手順



(a) 初期構築時

pd_lck_pool_partition オペランド（フロントエンドサーバの場合，pd_fes_lck_pool_partition オペランド）を省略します。

また，初期構築以降にチューニングした場合，共用メモリの使用量が変わります。そのため，チューニングする可能性がある場合は，事前に共用メモリの使用量を計算して必要な物理メモリの確保をすることを推奨します。

(b) チューニング

排他制御用プールパーティション数をチューニングする場合は，次の情報が必要になります。

- SQL 実行時間（SQL トレースに出力されます）
- サーバごとの排他制御用プールパーティションのスレッド間ロック待ち数（システムの稼働に関する統計情報に出力されます）
- 排他制御用プールパーティションごとの排他制御用プールパーティションのスレッド間ロック待ち数（pdls -d lck -T コマンドで表示されます）

実際に業務内容を想定した多重実行数と稼働率でアプリケーションを実行し，チューニング情報を取得してください。取得したチューニング情報を基に排他制御用プールパーティション数を徐々に増加しながら，業務アプリケーション全体で SQL 実行時間の合計が最も短くなる排他制御用プールパーティション数を設定してください。

ただし，次に示す場合は排他制御用プールパーティション数をチューニング前の値に戻して，ほかの方法で性能向上を図ってください。

- 排他制御用プールパーティション数を増やしたが，排他制御用プールパーティションのスレッド間ロック待ち数が減少しない場合
- 排他制御用プールパーティション数を増やしたことによって排他制御用プールパーティションのスレッド間ロック待ち数が減少したが，SQL の実行時間が短くならない場合

なお，排他制御処理を分散させた場合，ハッシュ分割によって排他資源ごとに使用する排他制御用プールパーティションを決定します。そのため，対象となる排他資源によっては，使用する排他制御用プールパーティションに偏りが発生することがあります。使用する排他制御用プールパーティションが偏ると，次の現象が発生するおそれがあります。

- 排他制御の性能が改善しない

  pdls -d lck -T コマンドを実行し，排他制御用プールパーティションごとのスレッド間ロック待ち数の平均値に偏りがあるかどうか確認してください。偏りがある場合は，pd_lck_pool_partition オペランド（フロントエンドサーバの場合は pd_fes_lck_pool_partition オペランド）の値を，現在の指定値より大きい素数に変更してください。これによって，ハッシュ分割の偏りを解消します。
- 排他制御用プールパーティションのサイズ不足

  pdls -d lck -p -e コマンドを実行し，排他制御用プールパーティションごとに使用した排他資源の使用率の最高値が，パーティションごとに大きく異なるかどうか確認してください。大きく異なっている場合は，pd_lck_pool_partition オペランド（フロントエンドサーバの場合は pd_fes_lck_pool_partition オペランド）の値を，現在の指定値より大きい素数に変更してください。これによって，ハッシュ分割の偏りを解消します。又は，pd_lck_pool_size オペランド（フロントエンドサーバの場合は

pd_fes_lck_pool_size オペランド）の値を増やしてください。これによって，排他資源管理テーブルの容量を増やします。

> **参考**
>
> 排他制御用プールのサイズは，pd_lck_pool_size オペランドで変更します。排他制御用プールは HiRDB によって排他制御用プールパーティションに割り当てられるため，排他制御用プールパーティションのサイズは排他制御用プールのサイズに比例します。排他制御用プールパーティションの利用率は，システムの稼働に関する統計情報に出力されます。利用率が高い場合は，より多くの排他資源を必要とするトランザクションを実行すると排他制御用プールパーティションのサイズ不足が発生し，トランザクションが実行できなくなります。その場合は，pd_lck_pool_size オペランドの指定値を増加することを検討してください。なお，システムの稼働に関する統計情報については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) 排他制御処理を分散させた場合のメリット及びデメリット

排他制御処理を分散させた場合の SQL ごとのメリット及びデメリットを次の表に示します。

表 9–4 排他制御処理を分散させた場合のメリット及びデメリット

SQL	排他制御用プールパーティション数を増やした場合		排他制御用プールパーティション数を減らした場合	
	メリット	デメリット	メリット	デメリット
処理完了時に COMMIT 及び ROLLBACK をしない SQL（ほとんどの操作系 SQL 及び制御系 SQL の一部など）	多重実行時の SQL 性能が向上します。	SQL 単体の性能が劣化します。	SQL 単体の性能が向上します。	多重実行時の SQL の性能が劣化します。
処理完了と同時に COMMIT 又は ROLLBACK をする SQL（COMMIT，ROLLBACK，定義系 SQL，PURGE TABLE など）	−	多重実行時の SQL の性能が劣化します。また，SQL 単体の性能も劣化します。	多重実行時の SQL の性能が向上します。また，SQL 単体での性能も向上します。	−

（凡例） −：該当しません。

(3) 注意事項

- 排他制御処理を分散させると，排他制御が使用する管理情報を排他制御用プールパーティション数分追加するため，ユニットコントローラ用共用メモリのサイズが増加します。ユニットコントローラが使用する共用メモリの計算式については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。排他制御用プールパーティション数を増やす場合は，ユニットコントローラ用共用メモリのサイズの増加によって，物理メモリが不足しないよう，必要に応じてシステム構成を変更してください。

- 排他制御用プールパーティションのスレッド間ロック待ち数が 0，又は 0 に近い値の場合でも，スレッド間ロック待ちが発生していることがあります。その場合は，「チューニング」を行っても性能が向上しないことがあります。性能が向上しないときは，次を実施してください。
  - 同時実行する UAP 又はユティリティの数を減らす
  - 業務又は HiRDB システム定義を見直して，排他要求数を減らす

9.11　トランザクション回復メッセージキューサイズを変更する方法

9.11.1　トランザクション回復メッセージキューサイズの適用基準，及び変更方法

トランザクション回復メッセージキューとは，回復対象となるトランザクションを一時的に格納しておくための領域です。HiRDB は，トランザクションの回復処理を行うとき，回復対象となるトランザクションをトランザクション回復メッセージキューに登録します。登録されたトランザクションは，トランザクション回復メッセージキューから取り出され，回復処理が行われます。

トランザクションがトランザクション回復メッセージキューに登録されるときに，多数の回復待ちトランザクションがあると，トランザクション回復メッセージキューへの登録待ちが発生し，トランザクションの回復に時間が掛かります。これを防ぐために，トランザクション回復メッセージキューのサイズを大きくすることができます。

(1)　適用基準

HiRDB は，トランザクション回復メッセージキューが不足すると，次のように KFPS00854-W メッセージを出力します。このメッセージが出力された場合は，トランザクション回復メッセージキューサイズの変更を検討してください。

```
KFPS00854-W  Insufficient memory in message buffer pool. server=_trnrcv
```

(2)　トランザクション回復メッセージキューサイズの変更方法

トランザクション回復メッセージキューサイズは，システム定義の pd_trn_rcvmsg_store_buflen オペランドで指定します。このオペランドの指定値の目安については，マニュアル「HiRDB システム定義」を参照してください。

9.12　OS の時刻を変更する方法

HiRDB ではシステムログファイル，ステータスファイル，及びディクショナリの中に日付・時間を格納しており，HiRDB の再開始時などにその情報を利用します。また，ユティリティの作業用ファイル中にも日付・時刻情報を持ち，チェック処理などに使用しています。そのため，OS の時刻を変更する場合は，HiRDB が持つ日付・時刻情報に影響がないようにする必要があります。OS の時刻を変更する場合の影響や注意について，次に示します。

時刻を進める場合

統計情報が正しく表示されない，メッセージ中の時刻表示が一定時間更新されないなどの影響があります。

なお，HiRDB 開始処理中に日時を変更すると，結果を保証できません。また，各 OS による制約事項もあるため，注意してください。

時刻を遅らせる場合

再開始の失敗，データベース回復不正，不当なシステムウェイトなど様々な問題を引き起こすおそれがあります。時刻を遅らせる場合，手順を誤るとこのような問題を引き起こすおそれがあるため，この節で示す手順に従ってください。

この節では，特に注意を必要とする，時刻を**遅らせる**場合について説明します。

9.12.1　時刻を遅らせる方法

時刻を遅らせる場合，次の二つの方法があります。システムの運用に応じてどちらかを選択してください。

方法 1：

時刻を遅らせた時間（例えば，1 時間遅らせた場合は，1 時間）が経過するのを待ってから HiRDB を開始する方法です。次のような場合にこの方法を選択してください。

- 時刻を遅らせた時間が経過するのを待つ間，サービスを停止できる
- 時刻を遅らせた時間が短時間である
- NetBackup 連携機能を使用してバックアップ取得及び回復をしている

方法 2：

時刻を遅らせた時間が経過するのを待たない方法です。ただし，HiRDB が持つ日付・時刻情報に影響がないように対処しながら時刻を遅らせる必要があります。サービス停止時間を短くする必要がある場合にこの方法を選択してください。

なお，これらの方法で時刻を遅らせる場合，HiRDB の運用に支障をきたさないように対処する必要があります。対処が必要な条件，対処，及び対処を誤った場合の影響を次の表に示します。

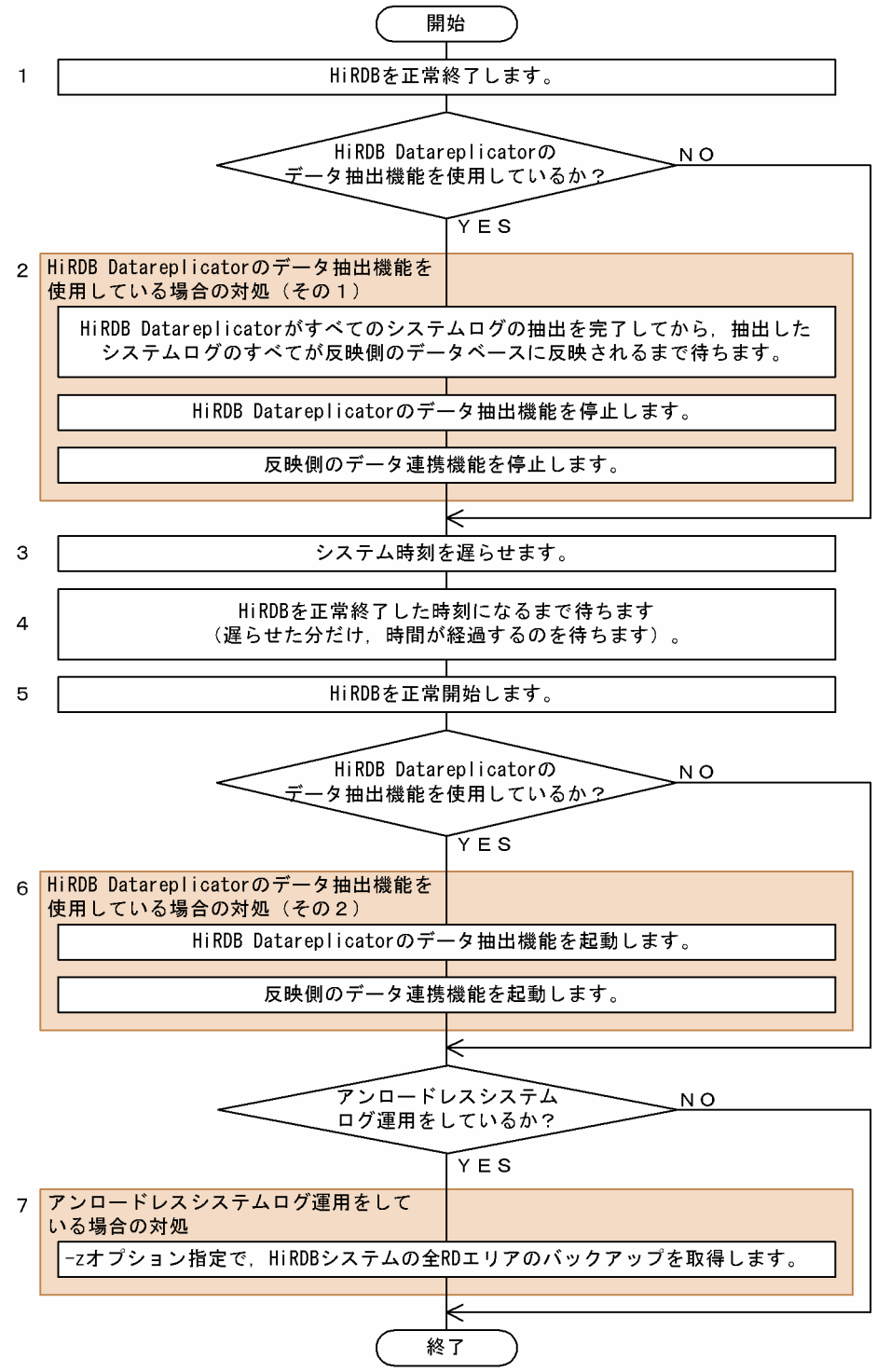
表 9–5　時刻を遅らせる場合に対処が必要な条件，対処，及び対処を誤ったときの影響

対処が必要な条件	時刻を遅らせる方法	対処	対処を誤ったときの影響
HiRDB Datareplicator のデータ抽出機能を使用している場合	方法 1 及び方法 2	HiRDB Datareplicator は，システムログ中の時刻情報を参照しています。そのため，時刻を遅らせることで HiRDB Datareplicator が誤動作しないよう対処します。	反映側のデータベースが破壊されるなどの障害が発生するおそれがあります。
アンロードレスシステムログ運用をしている場合	方法 1 及び方法 2	ログポイントを取得するため，時刻を遅らせた後で-z オプション指定でデータベース複写ユティリティ（pdcopy）を実行することで対処します。	ログポイントを取得していないと，障害が発生したときに，最新の状態まで回復できません。
pdcat コマンドでメッセージログファイルを表示している場合	方法 2	メッセージログファイルは，HiRDB が出力したメッセージを時系列に記録するファイルです。pdcat コマンドはメッセージログファイルを読み込み，時刻順に表示します。時刻を遅らせることで，時刻が不正になり，pdcat コマンドの表示結果が不正にならないよう対処します。	時刻を遅らせる以前のメッセージを HiRDB で参照できなくなります。
統計情報を取得している場合	方法 2	統計ログファイルに出力される統計情報が，時刻を遅らせることで時刻順に並ばなくなります。統計解析ユティリティ（pdstedit）による統計情報の集計結果が不正にならないよう対処します。	時刻を遅らせる以前の統計情報を解析できなくなります。
方法 2 で時刻を遅らせた場合	方法 2	時刻を遅らせる前にシステムログファイルをアンロードしたり，アンロードしたファイルを退避し，削除する必要があります。	障害発生時の回復ができなくなります。
		時刻を遅らせた後で，HiRDB の障害に備えてバックアップを取得し，システムログファイル及びシンクポイントダンプファイルを初期化する必要があります。	時刻を遅らせた後で障害が発生した場合，時刻を遅らせた後の状態に回復できなくなります。

9.12.2　方法 1 で時刻を遅らせる場合

方法 1 で時刻を遅らせる場合，次の手順に従ってください。

〈手順〉



注

処理ボックスの左にある数字はこの後で説明している（　）レベルに対応しています。例えば，3の操作は(3)で説明しています。

(1)　HiRDB を正常終了します

pdstop コマンドで HiRDB を正常終了します。

(2) HiRDB Datareplicator のデータ抽出機能を使用している場合の対処（その 1）

次の対処をしてください。なお，(b)と(c)の順序はどちらが先でもかまいません。HiRDB Datareplicator のコマンドについては，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

(a) HiRDB Datareplicator がすべてのシステムログの抽出を完了してから，抽出したシステムログのすべてが反映側のデータベースに反映されるまで待ちます

すべてのシステムログの抽出が完了したかどうかの確認方法，及びすべてのシステムログが反映されたかどうかの確認方法については，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」の「抽出側システムの構成変更」を参照してください。

(b) HiRDB Datareplicator のデータ抽出機能を停止します

hdestop コマンドで，HiRDB Datareplicator のデータ抽出機能を停止します。

(c) 反映側のデータ連携機能を停止します

反映側のデータ連携機能を停止します。反映側のデータ連携機能が HiRDB Datareplicator の場合は，hdsstop コマンドで停止します。

(3) システム時刻を遅らせます

OS の設定で時刻を遅らせます。HiRDB/パラレルサーバの場合，各サーバマシンの時刻が異なると，メッセージの突き合わせなどが煩雑になるため，各サーバマシンの時刻を同じに設定します。

(4) HiRDB を正常終了した時刻になるまで待ちます

遅らせた分だけ（例えば，1 時間遅らせた場合は，1 時間），時間が経過するのを待ちます。

(5) HiRDB を正常開始します

pdstart コマンドで HiRDB を正常開始します。

(6) HiRDB Datareplicator のデータ抽出機能を使用している場合の対処（その 2）

次の対処をしてください。なお，(a)と(b)の順序はどちらが先でもかまいません。

(a) HiRDB Datareplicator のデータ抽出機能を起動します

hdestart コマンドで，(2)(b)で停止した HiRDB Datareplicator のデータ抽出機能を起動します。

(b)　反映側のデータ連携機能を起動します

(2)(c)で停止した反映側データ連携機能を起動します。反映側のデータ連携機能が HiRDB Datareplicator の場合は，hdsstart コマンドで起動します。

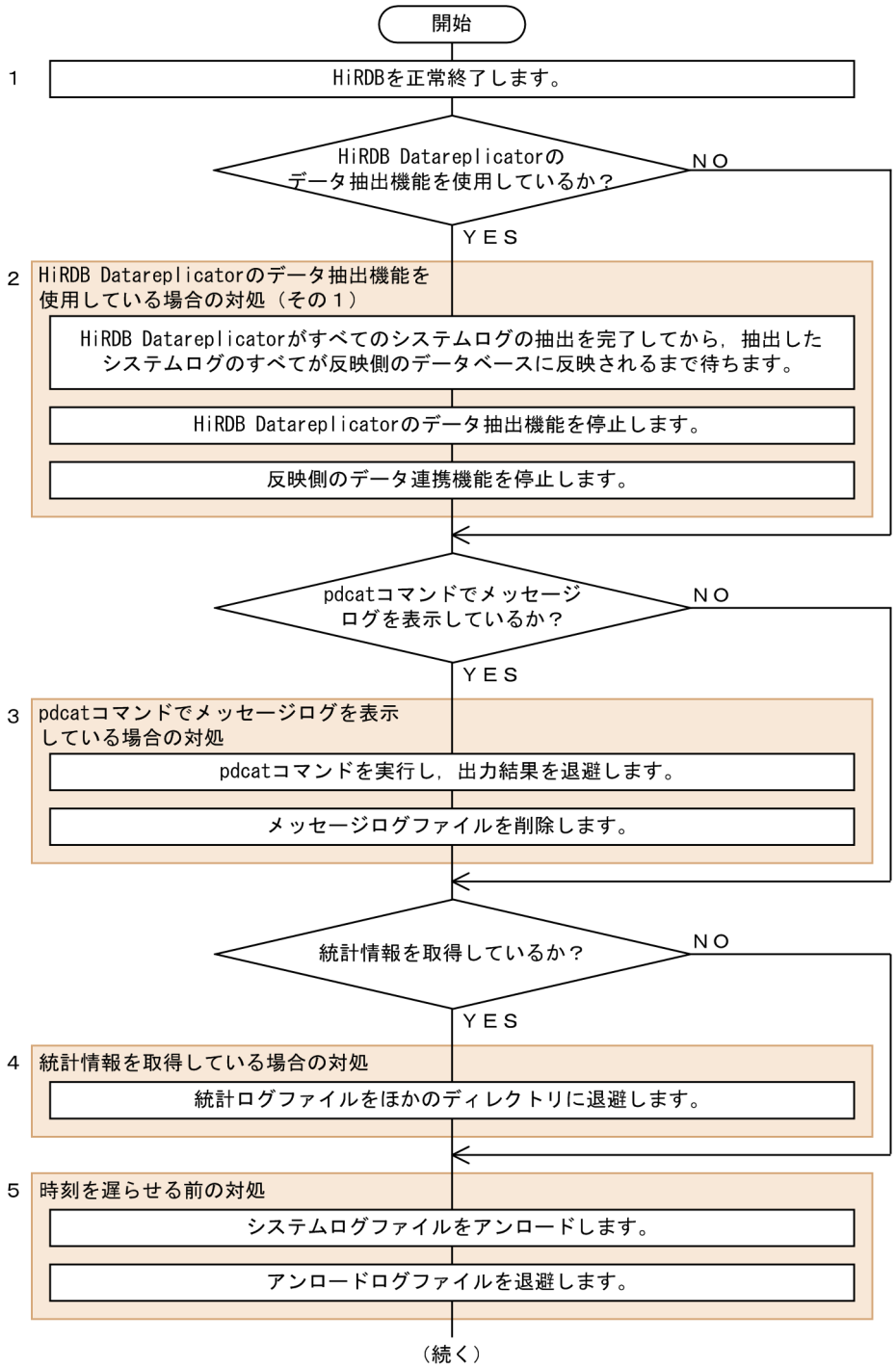
(7)　アンロードレスシステムログ運用をしている場合の対処

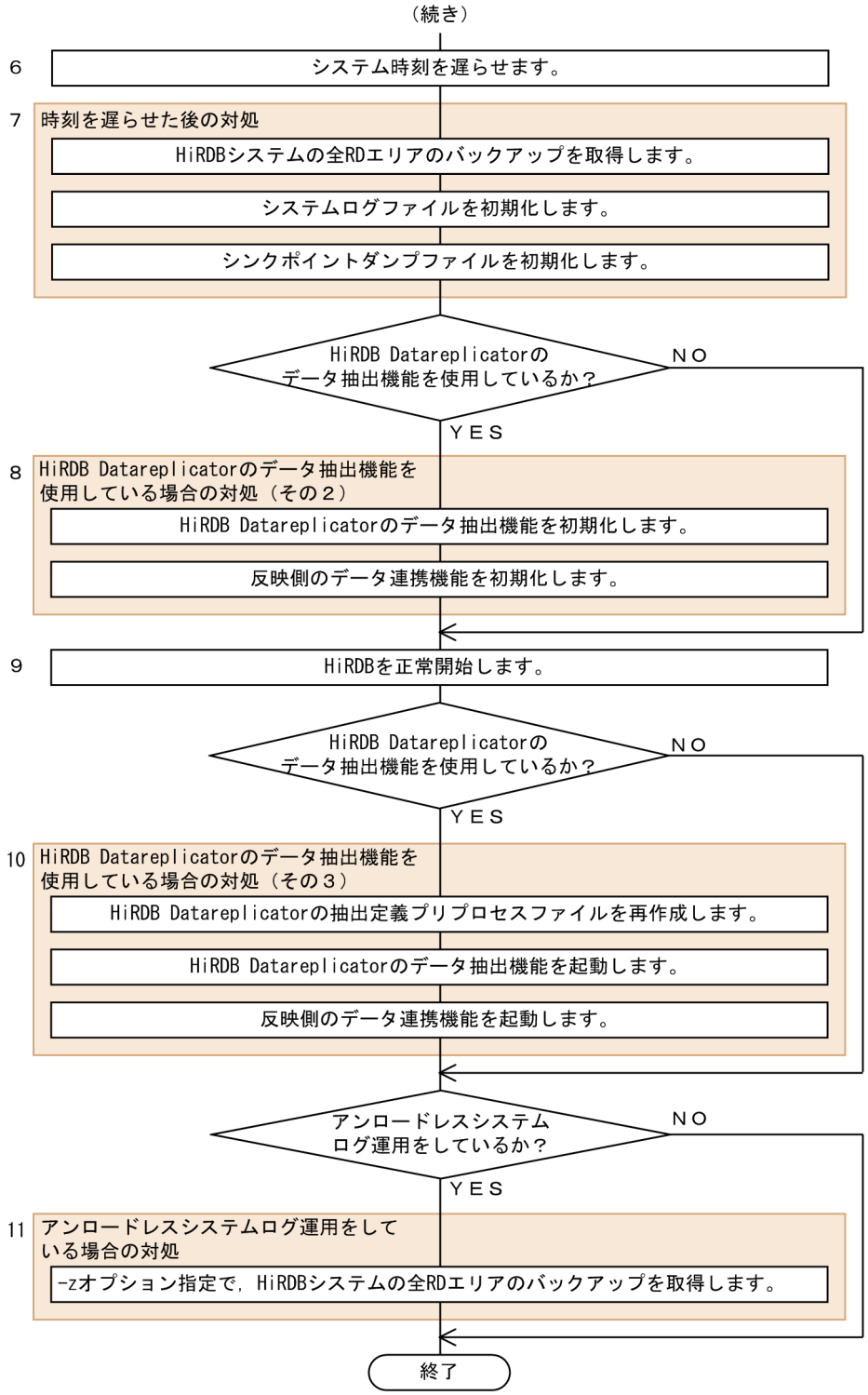
ログポイントを取得するために，-z オプション指定でデータベース複写ユティリティ（pdcopy）を実行して，全 RD エリアのバックアップを取得します。なお，HiRDB/パラレルサーバで全 RD エリアのバックアップを取得するには，サーバごとに-z オプション指定の pdcopy コマンドを実行する必要があります。

9.12.3　方法 2 で時刻を遅らせる場合

方法 2 で時刻を遅らせる場合，次の手順に従ってください。

〈手順〉


開始
1
HiRDBを正常終了します。
HiRDB Datareplicatorの
データ抽出機能を使用しているか？
ＮＯ
ＹＥＳ
2
HiRDB Datareplicatorのデータ抽出機能を
使用している場合の対処（その１）
HiRDB Datareplicatorがすべてのシステムログの抽出を完了してから，抽出した
システムログのすべてが反映側のデータベースに反映されるまで待ちます。
HiRDB Datareplicatorのデータ抽出機能を停止します。
反映側のデータ連携機能を停止します。
pdcatコマンドでメッセージ
ログを表示しているか？
ＮＯ
ＹＥＳ
3
pdcatコマンドでメッセージログを表示
している場合の対処
pdcatコマンドを実行し，出力結果を退避します。
メッセージログファイルを削除します。
統計情報を取得しているか？
ＮＯ
ＹＥＳ
4
統計情報を取得している場合の対処
統計ログファイルをほかのディレクトリに退避します。
5
時刻を遅らせる前の対処
システムログファイルをアンロードします。
アンロードログファイルを退避します。
（続く）



注

処理ボックスの左にある数字はこの後で説明している（　）レベルに対応しています。例えば，3の操作は(3)で説明しています。

(1)　HiRDB を正常終了します

pdstop コマンドで HiRDB を正常終了します。

(2) HiRDB Datareplicator のデータ抽出機能を使用している場合の対処（その 1）

次の対処をしてください。なお，(b)と(c)の順序はどちらが先でもかまいません。HiRDB Datareplicator のコマンドについては，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

(a) HiRDB Datareplicator がすべてのシステムログの抽出を完了してから，抽出したシステムログのすべてが反映側のデータベースに反映されるまで待ちます

すべてのシステムログの抽出が完了したかどうかの確認方法，及びすべてのシステムログが反映されたかどうかの確認方法については，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」の「抽出側システムの構成変更」を参照してください。

(b) HiRDB Datareplicator のデータ抽出機能を停止します

hdestop コマンドで，HiRDB Datareplicator のデータ抽出機能を停止します。

(c) 反映側のデータ連携機能を停止します

反映側のデータ連携機能を停止します。反映側のデータ連携機能が HiRDB Datareplicator の場合は，hdsstop コマンドで停止します。

(3) pdcat コマンドでメッセージログファイルを表示している場合の対処

(a) pdcat コマンドを実行し，出力結果を退避します

pdcat コマンドを実行し，出力結果をリダイレクトしてファイルに取得します。これによって，時刻を遅らせる前にメッセージログファイル（%PDDIR%¥spool¥pdlog1，pdlog2）に出力したメッセージを退避します。HiRDB/パラレルサーバで，pd_mlg_msg_log_unit オペランドに local を指定している場合，すべてのユニットに対して pdcat コマンドを実行してください。

なお，時刻を遅らせる前のメッセージを参照する必要がない場合は，この手順は省略してかまいません。

(b) メッセージログファイルを削除します

メッセージログファイル（%PDDIR%¥spool¥pdlog1，pdlog2）を OS のコマンドで削除します。HiRDB/パラレルサーバで，pd_mlg_msg_log_unit オペランドに local を指定している場合，すべてのユニットでメッセージログファイルを削除してください。

(4) 統計情報を取得している場合の対処

統計ログファイル（%PDDIR%¥spool¥pdstj1，pdstj2）を OS のコマンドでほかのディレクトリにコピーし，コピー元の統計ログファイルを OS のコマンドで削除します。これによって，時刻を遅らせる前の統計ログファイルを退避します。HiRDB/パラレルサーバの場合，すべてのユニットで統計ログファイルを退避してください。

なお，時刻を遅らせる以前の統計情報が不要の場合は，この手順は省略してかまいません。

(5)　時刻を遅らせる前の対処

(a)　システムログファイルをアンロードします

すべてのアンロード待ち状態のシステムログファイルを pdlogunld コマンドでアンロードします。ここでアンロードしたシステムログファイルは，障害が発生してデータベースを回復する場合に，時刻を遅らせる前の状態にするときに使用します。

(b)　アンロードログファイルを退避します

時刻を遅らせる前後のアンロードログファイルを異なるディレクトリで管理するため，(a)でアンロードしたアンロードログファイルを OS のコマンドでほかのディレクトリにコピーし，コピー元のすべてのアンロードログファイルを OS のコマンドで削除します。

注意事項

時刻を遅らせる前後のアンロードログファイルが同じディレクトリに混在していると，次に示す現象が発生するおそれがあります。

- アンロードログファイル格納ディレクトリを指定してデータベースを回復する場合，データベースが破壊される
- 自動ログアンロード機能を使用している場合，作成されるアンロードログファイルのパス名に時刻情報が含まれるため，上書きされる

(6)　システム時刻を遅らせます

OS の設定で時刻を遅らせます。HiRDB/パラレルサーバの場合，各サーバマシンの時刻が異なると，メッセージの突き合わせなどが煩雑になるため，各サーバマシンの時刻を同じに設定します。

(7)　時刻を遅らせた後の対処

(a)　HiRDB システムの全 RD エリアのバックアップを取得します

HiRDB システムの全 RD エリアのバックアップを取得します。ここで取得したバックアップファイルは，障害が発生してデータベースを回復する場合に，時刻を遅らせた後の状態にするときに使用します。バックアップの取得手順を次に示します。

データベース複写ユティリティ（pdcopy）でバックアップを取得する場合

1. pdstart -r コマンドで HiRDB を開始します。
2. pdcopy コマンドで全 RD エリアのバックアップを取得します。差分バックアップ機能を使用している場合，-g オプションの差分バックアップグループ名の最後に(S)を付けてください。

3. pdstop コマンドで HiRDB を正常終了します。

データベース複写ユティリティ（pdcopy）以外でバックアップを取得する場合

pdcopy コマンド以外（ほかの製品の機能）で，HiRDB システムの全 RD エリアのバックアップを取得します。

(b) システムログファイルを初期化します

時刻を遅らせた後で，時刻を遅らせる前のシステムログファイルを使用すると，HiRDB が誤動作するおそれがあるため，システムログファイルを初期化します。手順を次に示します。

1. pdlogrm -d sys コマンドでシステムログファイルを削除します。
2. pdloginit -d sys コマンドでシステムログファイルを再作成します。

(c) シンクポイントダンプファイルを初期化します

次に示す理由から，HiRDB が誤動作するおそれがあるため，シンクポイントダンプファイルを初期化します。

- 時刻を遅らせた後で，時刻を遅らせる前のシンクポイントダンプファイルを使用すると，前回のシンクポイントダンプ取得時刻が未来の時刻になる
- シンクポイントダンプに対応したシステムログファイルが(b)で初期化されている

シンクポイントダンプファイルを初期化する手順を次に示します。

1. pdlogrm -d spd コマンドでシンクポイントダンプファイルを削除します。
2. pdloginit -d spd コマンドでシンクポイントダンプファイルを再作成します。

(8) HiRDB Datareplicator のデータ抽出機能を使用している場合の対処（その 2）

次の対処をしてください。なお，(a)と(b)の順序はどちらが先でもかまいません。

(a) HiRDB Datareplicator のデータ抽出機能を初期化します

時刻を遅らせた後も HiRDB Datareplicator のデータ抽出機能を正しく動作させるために，hdestart -i コマンドで HiRDB Datareplicator のデータ抽出機能を初期化します。

(b) 反映側のデータ連携機能を初期化します

(a)で HiRDB Datareplicator のデータ抽出機能を初期化したため，反映側のデータ連携機能も初期化します。反映側のデータ連携機能が HiRDB Datareplicator の場合は，hdsstart -i -q コマンドで初期化します。

(9) HiRDB を正常開始します

pdstart コマンドで HiRDB を正常開始します。

(10)　HiRDB Datareplicator のデータ抽出機能を使用している場合の対処（その 3）

次の対処をしてください。なお，(b)と(c)の順序はどちらが先でもかまいません。

(a)　HiRDB Datareplicator の抽出定義プリプロセスファイルを再作成します

(8)で HiRDB Datareplicator のデータ抽出機能を初期化したため，hdeprep コマンドで抽出定義プリプロセスファイルを再作成します。

(b)　HiRDB Datareplicator のデータ抽出機能を起動します

hdestart コマンドで，(2)(b)で停止した HiRDB Datareplicator のデータ抽出機能を起動します。

(c)　反映側のデータ連携機能を起動します

(2)(c)で停止した反映側データ連携機能を起動します。反映側のデータ連携機能が HiRDB Datareplicator の場合は，hdsstart コマンドで起動します。

(11)　アンロードレスシステムログ運用をしている場合の対処

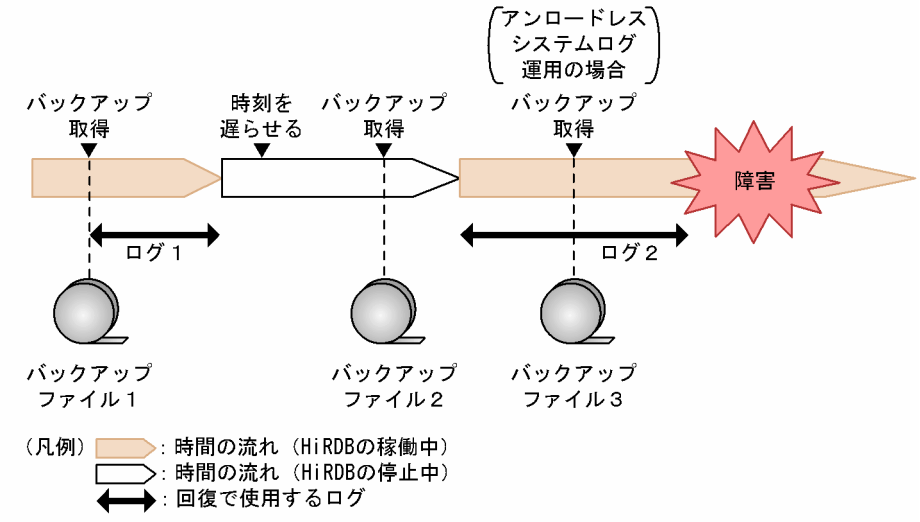
ログポイントを取得するために，-z オプション指定でデータベース複写ユティリティ（pdcopy）を実行して，全 RD エリアのバックアップを取得します。なお，HiRDB/パラレルサーバで全 RD エリアのバックアップを取得するには，サーバごとに-z オプション指定の pdcopy コマンドを実行する必要があります。

9.12.4　方法 2 で時刻を遅らせた後で障害が発生した場合のデータベースの回復

「方法 2 で時刻を遅らせる場合」の手順に従って時刻を遅らせた後で障害が発生した場合の回復の方法について説明します。

時刻を遅らせた後で障害が発生した場合の例を次の図に示します。この例を使用して説明します。

図 9–7　時刻を遅らせた後で障害が発生した場合の例



(1)　最新の同期点までデータベースを回復する場合の対処

障害発生直前の最新の同期点までデータベースを回復するには，次のように対処します。

アンロードレスシステムログ運用をしていないとき

時刻を遅らせた後に取得したバックアップファイル 2，及びバックアップ取得後のログ 2 のアンロードログファイルを使用して，pdrstr コマンドを実行します。

アンロードレスシステムログ運用をしているとき

時刻を遅らせ，HiRDB を開始した後で取得したバックアップファイル 3，及びログ 2 のシステムログファイルを使用して，pdrstr コマンドを実行します。

(2)　バックアップファイルを使用できない場合の対処

時刻を遅らせた後でバックアップの取得を忘れたり，取得したバックアップファイルが破損したりして，バックアップを使用できない場合，次の手順で回復できます。

アンロードレスシステムログ運用をしていないとき

1. 時刻を遅らせる前に取得したバックアップファイル 1，及びログ 1 のアンロードログファイルを使用して，pdrstr コマンドで時刻を遅らせる状態までデータベースを回復します。
2. 時刻を遅らせた後のログ 2 のアンロードログファイルを使用して，pdrstr コマンドで障害発生直前の最新の同期点までデータベースを回復します。

> **注意事項**
>
> 基本的には，ログだけを使用したデータベースの回復は，バックアップだけを使用した回復の後で実行する必要があります。上記に示したバックアップを使用できない場合の回復手順は，この節に記載している手順どおりに時刻を変更した場合にだけ実行してください。

アンロードレスシステムログ運用をしているとき

バックアップファイル 3 が使用できないときは，バックアップファイル 2 を使用してバックアップファイル 2 取得時点までデータベースを回復します。バックアップファイル 2 も使用できないときは，バックアップファイル 1 を使用してバックアップファイル 1 取得時点までデータベースを回復します。

9.12.5　方法 2 で時刻を遅らせる場合の制限事項

方法 2 で時刻を遅らせる場合の制限事項と対策を次の表に示します。

表 9–6　方法 2 で時刻を遅らせる場合の制限事項と対策

制限事項	対策
NetBackup 連携機能を使用したバックアップ取得及び回復はできません。	方法 1 で時刻を遅らせてください。
pdfrstr コマンド実行時，-t オプションを指定していても，タイムスタンプを比較できません。	次に示すどちらかの対策をしてください。 • -t オプションを使用しないで，バックアップファイルをユーザが管理するようにしてください。 • 時刻を遅らせた後で，pdfbkup コマンドを再度実行し，取得したバックアップファイルを使用して，pdfrstr コマンドを実行してください。
pdcspool コマンドで一部のファイルが削除できません。	次に示すどちらかの対策をしてください。 • 時刻を遅らせる前に HiRDB を終了した時刻まで待ってから，pdcspool コマンドを再度実行してください。 • -d オプションに 0 を指定して pdcspool コマンドを実行してください。すべてのファイルを削除できます。
HiRDB Datareplicator のデータ抽出機能を使用している場合，抽出対象の表を生成した時刻より遅い時刻に遅らせることはできません。	OS の時刻を遅らせた後で HiRDB Datareplicator の抽出機能を起動するまでに，抽出対象の表に対して次に示す手順を実行してください。 1. データベース再編成ユティリティ（pdrorg -k unld）で表のデータをアンロードします。 2. DROP TABLE で表を削除します。 3. CREATE TABLE で表を再定義します。 4. データベース再編成ユティリティ（pdrorg -l reld）で表のデータをリロードします。 5. hdeprep コマンドで HiRDB Datareplicator の抽出プリプロセス定義ファイルを再生成します。
JP1/PFM-Agent for HiRDB を使用している場合，OS の時刻を遅らせてから，前回 HiRDB を終了した時刻になるまでパフォーマンスデータを取得できません。	次に示す手順を実行してください。 1. OS の時刻を進めて，運用又はテストをする前に，JP1/PFM-Agent for HiRDB を停止してください。 2. 運用又はテスト終了後，OS の時刻を遅らせてから JP1/PFM-Agent for HiRDB を再起動してください。
HiRDB CM を使用している場合，HiRDB CM のバックアップファイル及びアンロードログファイル管理機能を使用できません。	次に示すどちらかの対策をしてください。 **対策 1** 1. OS の時刻を進めて，運用又はテストをする前に，HiRDB CM を停止してください。

制限事項	対策
	2. 運用又はテスト終了後，OS の時刻を遅らせてから HiRDB CM を再起動してください。 **対策 2** 1. OS の時刻を遅らせる前に HiRDB CM が記憶しているバックアップファイル及びアンロードログファイルの情報を削除してください。 2. OS の時刻を遅らせた後で，HiRDB CM が記憶しているバックアップファイル及びアンロードログファイルの情報を再取得してください。

9.12.6　方法 2 で時刻を遅らせる場合の注意事項

方法 2 で時刻を遅らせる場合の注意事項を次に示します。

(1)　コマンド又はユティリティの出力結果

時刻を遅らせた後で，時刻を遅らせる前に HiRDB のファイル中に記録された時刻を表示した場合，未来の時刻が表示されることがあります。

(2)　データベース再編成ユティリティ（pdrorg）によるインデクスの一括作成

HiRDB は RD エリア中の表管理情報とインデクス情報ファイルに，インデクス情報ファイルの作成時刻を保持します。インデクスの一括作成実行時，インデクス情報ファイルのデータと表のデータの整合性を確保するために，HiRDB は保持した時刻が一致しているか確認し，一致しない場合はエラー終了します。

システムの時刻を遅らせた後でインデクス情報ファイルを作成し，時刻を遅らせる前に作成したインデクス情報ファイルを指定した場合，RD エリア中の表管理情報の時刻が一致すると，インデクスの一括作成を実行してもエラーにならないで，表とインデクスが不整合な状態になります。不整合な状態になった場合，回復するには，インデクスの再作成（-k ixrc）が必要になります。

(3)　再編成時期予測機能による DB メンテナンス予定日

時刻を遅らせた後で，データベース状態解析ユティリティ（pddbst）による再編成時期予測機能を実行すると，DB メンテナンス予定日が時刻を遅らせる前より早まったり，不要なメンテナンスを指示したりすることがあります。

(4)　時刻を遅らせる前後のアンロードログファイルの管理

時刻を遅らせた後で，-T，-d，及び-l オプション指定でデータベース回復ユティリティ（pdrstr）を実行する場合，ログの入力順序が正しくないとデータベースが破壊されるため，時刻を遅らせる前後のアンロードログファイルは別々に管理する必要があります。

9.13　デッドロック監視処理時間を変更する方法

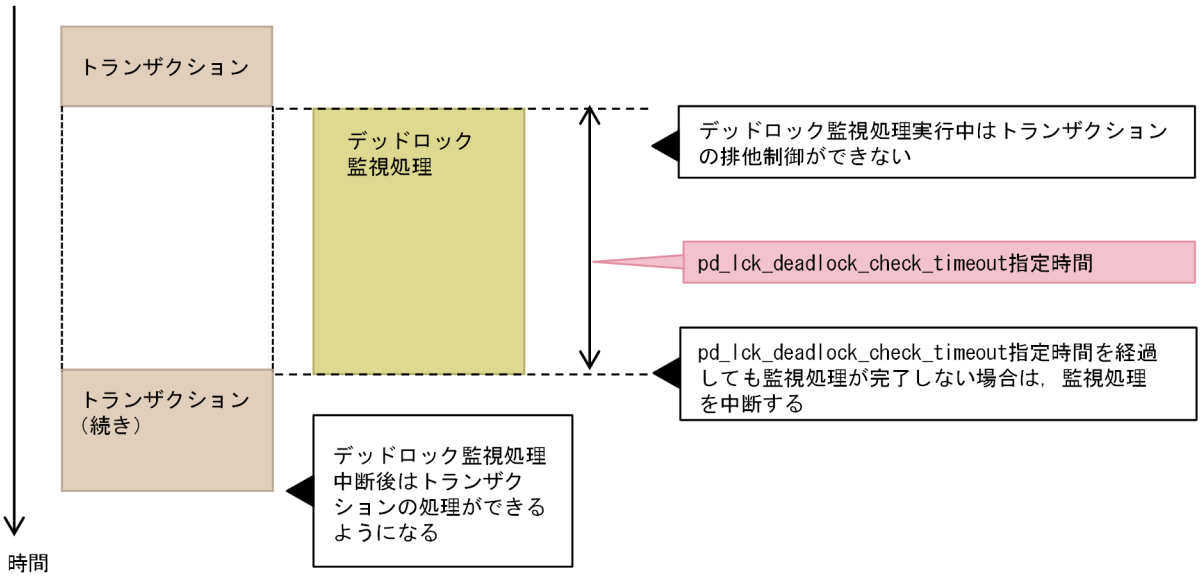
9.13.1　デッドロック監視処理時間の変更方法

デッドロックの検出方法がインターバル監視方式の場合（pd_lck_pool_partition 又は pd_fes_lck_pool_partition オペランドに 2 以上を指定した場合），デッドロックの監視間隔の間にデッドロックが多発するおそれがあり，デッドロックが多発すると，デッドロック監視処理にて検出するまでに長時間掛かることがあります。デッドロック監視処理中は，排他取得や排他解除を行えないため，トランザクションが長時間無応答になります。

pd_lck_deadlock_check_timeout オペランドに 0 以外の値を指定することで，デッドロック監視処理に時間が掛かる場合に監視処理を中断し，排他取得や排他解除ができるようになり，トランザクション処理を続行できます。

トランザクションとデッドロック監視処理の関係について，次の図に示します。

図 9–8　トランザクションとデッドロック監視処理の関係



(1)　デッドロック監視中断後の監視処理正常化までの流れ

デッドロック監視処理を中断した場合，デッドロックに関連しているトランザクションは，排他待ち限界経過時間（PDLCKWAITTIME や pd_lck_wait_timeout など）や HiRDB クライアントの最大待ち時間（PDCWAITTIME）経過後にロールバックすることで，デッドロックが解消されます。

また，デッドロックに関連していないトランザクションもデッドロック監視処理中は排他取得や排他解除ができないが，デッドロック監視中断後には排他取得や排他解除ができ，トランザクションを続行できます。

上記により，デッドロックの数が少なくなることでデッドロック監視処理が監視時間内に正常に終了するようになります。

(2) 関連する設定

各 UAP やユティリティは排他待ち限界経過時間や HiRDB クライアントの最大待ち時間などのトランザクションのタイムアウト時間を指定してください。タイムアウトすることでデッドロックが解消し，この機能によって中断したデッドロック監視処理を正常化します。関連する各オペランドの指定の目安については，マニュアル「HiRDB システム定義」を参照してください。

- pd_lck_deadlock_check_timeout
- pd_lck_deadlock_check_pause

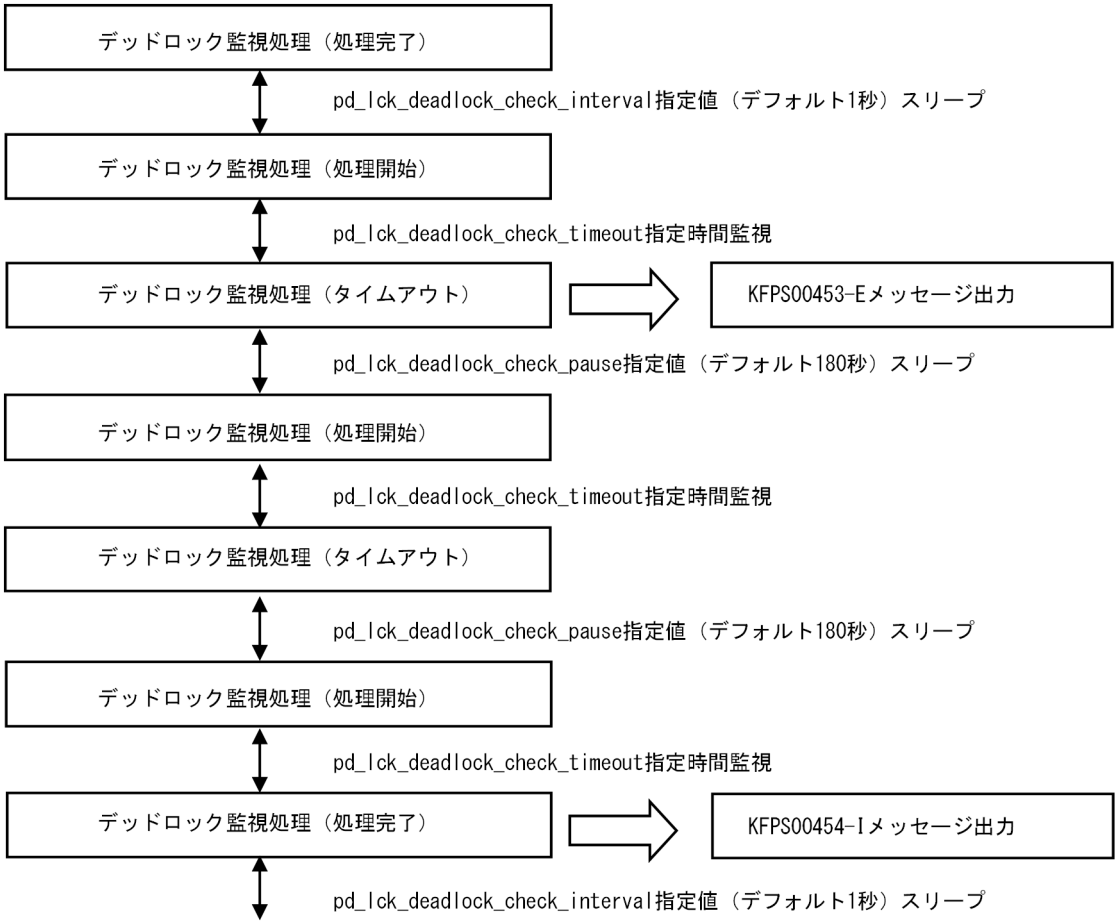
(3) デッドロック監視処理再開までの流れ

デッドロック監視処理が中断する際，KFPS00453-E メッセージを出力します。その後，pd_lck_deadlock_check_pause オペランド指定値時間スリープした後デッドロック監視処理を実行します。連続してデッドロック監視処理が中断した場合，KFPS00453-E メッセージは最初の１回だけ出力します。

デッドロック監視処理が中断した後にデッドロック監視処理が完了した場合，KFPS00454-I メッセージを出力します。その後，pd_lck_deadlock_check_interval 指定値ごとに監視する処理に戻ります。

デッドロック監視処理を中断してから再開するまでの動作を次の図に示します。

図 9-9　デッドロック監視処理を中断してから再開するまでの動作



10 HiRDB ファイルシステム領域の運用

この章では，HiRDB ファイルシステム領域，及び HiRDB ファイルの運用方法について説明します。

10.1　HiRDB ファイルシステム領域に関する情報を知る方法

実行者　HiRDB 管理者

HiRDB ファイルシステム領域に関する情報を知るには，次の表に示すコマンドを利用します。

表 10–1　HiRDB ファイルシステム領域に関する情報を表示するコマンド

コマンド名	利用するときの状況
pdfstatfs	• HiRDB ファイルシステム領域の領域長を知りたい場合 • HiRDB ファイルシステム領域の未使用領域長を知りたい場合 • HiRDB ファイルシステム領域内に作成した HiRDB ファイル数を知りたい場合 • HiRDB ファイルシステム領域内にこれから作成できる HiRDB ファイル数を知りたい場合
pdfls	• HiRDB ファイルシステム領域内にどのような HiRDB ファイルがあるか知りたい場合 • HiRDB ファイルの所有者，アクセス権，レコード長，レコード数，又は更新クローズ日時を知りたい場合 • HiRDB ファイルの排他制御の状態を知りたい場合

pdfstatfs 及び pdfls コマンドを実行して表示される内容の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

10.2　HiRDB ファイルシステム領域を作成（初期設定）する方法

ここでは，HiRDB ファイルシステム領域を作成（初期設定）する方法について説明します。

10.2.1　作成手順

HiRDB ファイルシステム領域を作成するときの手順を次に示します。

〈手順〉

1. HiRDB ファイルシステム領域を初期設定します。
2. OS 又はデバイスドライバの機能で，物理ボリューム及び論理ボリュームの入出力エラーを検知するまでの時間を指定します。

(1)　HiRDB ファイルシステム領域を初期設定します

実行者　HiRDB 管理者

pdfmkfs コマンドで HiRDB ファイルシステム領域を作成します。HiRDB ファイルシステム領域の設計方法については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。コマンド実行例を次に示します。

（例）

```
pdfmkfs -n 25 -l 5 -k DB -e 5 C:¥rdarea¥rdarea01
```

〔説明〕

-n：HiRDB ファイルシステム領域の領域長をメガバイト単位で指定します。

ここで指定する領域長は，パーティションの領域長と同じにするか，又は小さくしてください。パーティションの領域長より大きくすると，そのパーティションに物理的に続くパーティションを破壊する場合があります。

-l：HiRDB ファイルシステム領域に作成する HiRDB ファイルの最大数を指定します。

-k：HiRDB ファイルシステム領域の使用目的を指定します。

DB：RD エリア用の HiRDB ファイルシステム領域とします。

SYS：システムファイル用の HiRDB ファイルシステム領域とします。

WORK：作業表用ファイル用の HiRDB ファイルシステム領域とします。

UTL：ユティリティ用の HiRDB ファイルシステム領域とします。

SVR：すべての目的で使用できる HiRDB ファイルシステム領域とします（ユティリティ用を除く）。

-e：HiRDB ファイルの増分回数の最大値を指定します。

C:¥rdarea¥rdarea01：作成する HiRDB ファイルシステム領域の名称を指定します。

pdfmkfs コマンドの実行後，pdfstatfs コマンドで実行結果を確認することをお勧めします。pdfstatfs コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) OS 又はデバイスドライバの機能で，物理ボリューム及び論理ボリュームの入出力エラーを検知するまでの時間を指定します。

実行者　Administrator

詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB ファイルシステム領域を作成する準備」を参照してください。

10.3　HiRDB ファイルシステム領域のバックアップを取得する方法

実行者　HiRDB 管理者

pdfbkup コマンドで，HiRDB ファイルシステム領域のバックアップを取得できます。次に示す場合に pdfbkup コマンドを使用します。

- HiRDB ファイルシステム領域内の断片化した空き領域をなくす場合
- HiRDB ファイルシステム領域の容量やファイル数などの属性を変更する場合
- 障害発生に備えて，HiRDB ファイルシステムの内容を保存しておく場合(データベースのバックアップを RD エリア単位ではなく，HiRDB ファイルシステム単位に取得したい場合)

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

10.4　HiRDB ファイルシステム領域を回復する方法

実行者　HiRDB 管理者

pdfbkup コマンドで取得したバックアップから HiRDB ファイルシステム領域をリストア（回復）できます。**pdfrstr** コマンドで，HiRDB ファイルシステム領域をリストアします。次に示す場合に pdfrstr コマンドを使用します。

- HiRDB ファイルシステム領域内の断片化した空き領域をなくす場合
- HiRDB ファイルシステム領域の容量やファイル数などの属性を変更する場合
- 障害発生時に，以前取得したバックアップから HiRDB ファイルシステム領域の内容を回復する場合※

注※

障害発生直前の同期点にデータベースを回復する場合，HiRDB ファイルシステム領域をリストアした後，データベース回復ユティリティでアンロードログだけを使用した回復をする必要があります。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

10.5　HiRDB ファイルを削除する方法

10.5.1　HiRDB ファイルの削除方法

実行者　HiRDB 管理者

(1)　RD エリアを構成する HiRDB ファイルを削除する方法

RD エリアを構成する HiRDB ファイルの削除及び HiRDB ファイルの容量変更は，データベース構成変更ユティリティ（**pdmod** コマンド）で実行します。

なお，削除した HiRDB ファイルは，HiRDB を終了しなくてもデータベース構成変更ユティリティで再作成できます。

注意事項

- pdfrm コマンドで HiRDB ファイルを削除する場合は，RD エリアの構成に関係なく強制的に削除できるため，その実行には十分注意してください。
- pdfrm コマンドで HiRDB ファイルを削除する場合，HiRDB 稼働中は HiRDB が使用している HiRDB ファイルシステム領域中にある HiRDB ファイルは削除できません。HiRDB を終了した後に，pdfrm コマンドを実行してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2)　システムファイルを削除する方法

システムファイルは，次の表に示す運用コマンドで削除してください。

表 10–2　システムファイルを削除するコマンド

使用するコマンド名	システムファイルの種類
pdlogrm -d sys	システムログファイルの場合
pdlogrm -d spd	シンクポイントダンプファイルの場合
pdstsrm	ステータスファイルの場合

注意事項

- これらの運用コマンドでシステムファイルを削除できない場合は，削除できない原因を調査し，削除することに問題がなければ pdfrm コマンドで削除してください。

- 削除したシステムファイルは，HiRDB を終了しなくても運用コマンドで再作成できます。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

10.6　ファイルアクセスエラー検知時ユニットダウン機能

ここでは，ファイルアクセスエラー検知時ユニットダウン機能を使用する方法について説明します。なお，ここでいうファイルアクセスエラーとは，HiRDB がファイルを操作したときにアクセス権がないためファイル操作に失敗したエラーを意味し，HiRDB ファイルシステムに対するアクセス要求から返されるエラーコードに-1540 が出力されるエラーのことです。詳細については，マニュアル「HiRDB メッセージ」の「HiRDB ファイルシステムのエラーコード一覧」を参照してください。

10.6.1　ファイルアクセスエラー検知時ユニットダウン機能とは

RD エリア用の HiRDB ファイルシステム領域のアクセス権を誤って設定すると，HiRDB は RD エリアに対するファイルアクセスエラーを検知し，その RD エリアを障害閉塞します。ファイルアクセスエラー検知時ユニットダウン機能を使用すると，RD エリアに対するファイルアクセスエラーを検知した場合，RD エリアを閉塞しないでユニットダウンさせることができます（ただし，マスタディレクトリ用 RD エリアの場合はファイルアクセスエラー検知時ユニットダウン機能を使用していなくても，障害閉塞しないでユニットダウンします）。これによって，HiRDB を再開始する前にファイルアクセスエラーの原因を取り除くことができ，RD エリアが障害閉塞することによる回復作業を回避できます。

10.6.2　注意事項

次に示す運用で UAP 又はユティリティを実行中に，ファイルアクセスエラーを検知してユニットダウンした場合，処理対象の RD エリアが閉塞することがあります。

- 更新前ログ取得モード又はログレスモードで実行している場合
- CREATE TABLE の RECOVERY オペランドで NO（ログレスモード）を指定しているユーザ LOB 用 RD エリアに対して，UAP 又はユティリティを実行している場合

ファイルアクセスエラー検知時ユニットダウン機能を使用する場合は，できるだけこれらの運用は避けてください。もし，これらの運用が必要な場合は，RD エリアが閉塞しても最新の状態に回復できるよう，UAP 又はユティリティの実行前にバックアップを取得してください。

10.6.3　準備作業

この機能を使用するには，システム共通定義（pdsys）を次のように指定してください。

1. pd_db_access_error_action オペランドに unitdown を指定します。
   この値を指定すると，ファイルアクセスエラーを検知した場合，KFPH23040-I メッセージを出力し，ユニットダウンします。この場合，RD エリアの障害閉塞は発生しないため，HiRDB の再開始だけで回復します。

2. pd_mode_conf オペランドに MANUAL2 を指定します。

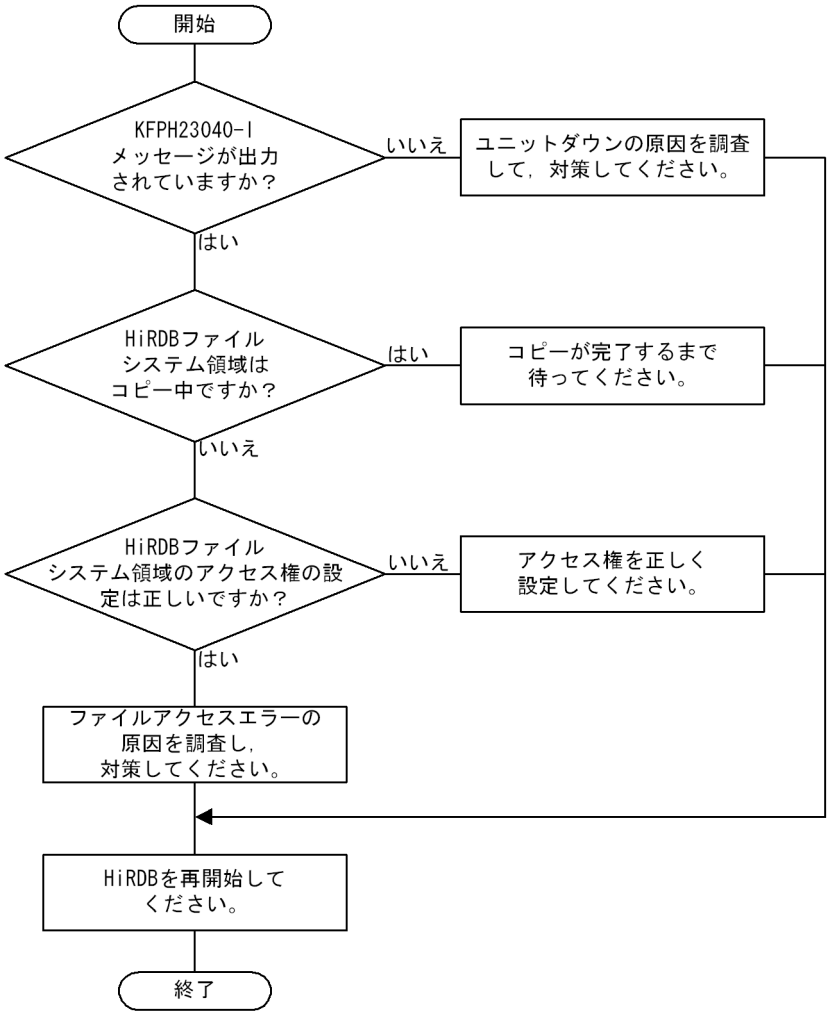
   この値を指定すると，ユニットダウン後に HiRDB を自動開始しないようにできます。

   なお，AUTO 又は MANUAL1 を指定すると，ファイルアクセスエラーを検知してユニットダウンしても，原因を取り除く前に HiRDB が再開始してしまうおそれがあります。この場合，再びファイルアクセスエラーを検知してしまい，ユニットダウンと再開始が繰り返されますので注意してください。

10.6.4　ファイルアクセスエラーが発生した場合の対処方法

ファイルアクセスエラー検知時ユニットダウン機能を使用していてユニットダウンが発生した場合，HiRDB 管理者は次の図に示す手順に従って対応してください。

図 10–1　ファイルアクセスエラーが発生した場合の対処方法



10.7　HDP 機能を使用した HiRDB ファイルシステム領域の仮想化

10.7.1　HDP 機能とは

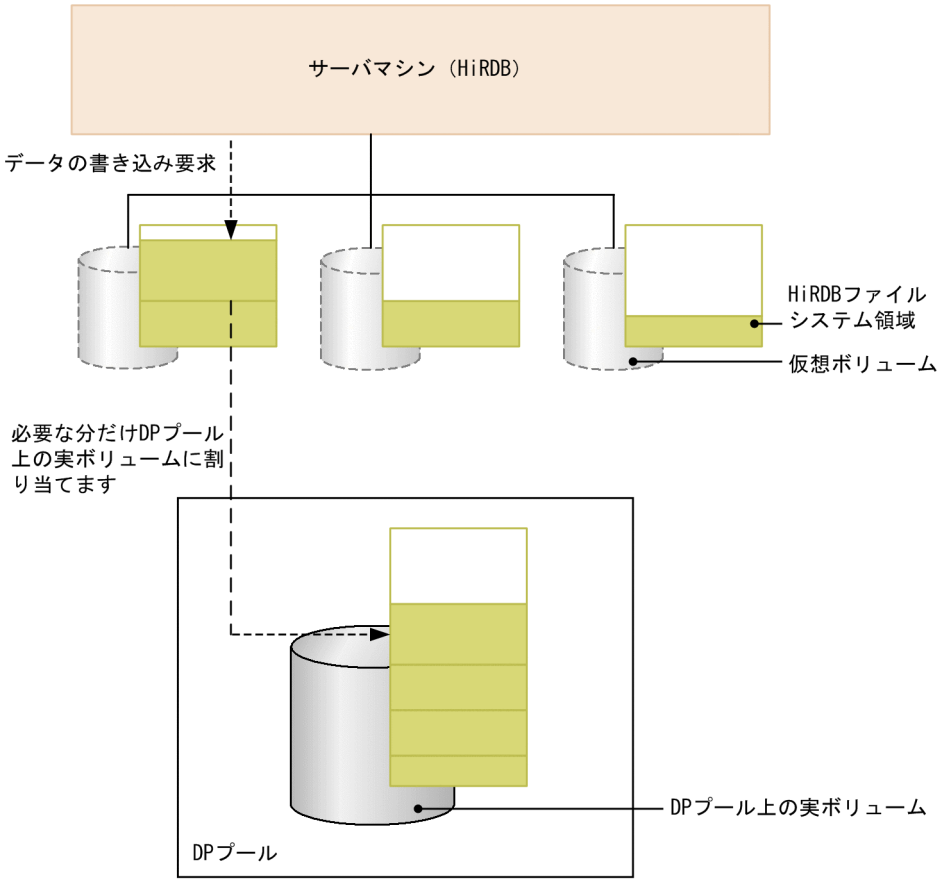
HDP 機能とは，サーバマシンに仮想ボリュームを提供し，サーバマシンからのデータの書き込み要求に対して，必要な分だけの実容量を割り当てる機能のことをいいます。

仮想ボリュームに HiRDB ファイルシステム領域を作成することで，実ボリュームに HiRDB ファイルシステム領域を作成する場合に比べて，容易にファイルサイズを拡張できるようになります。

HDP 機能の詳細については，HDP 機能のドキュメントを参照してください。

HDP 機能の概要を次の図に示します。

図 10–2　HDP 機能の概要



注意事項

DP プールが容量不足の状態でデータの書き込み要求を行うと，入出力エラーが発生します。このため，DP プールが容量不足にならないように容量を監視する必要があります。DP プールの容量を確認する方法については，HDP 機能のドキュメントを参照してください。

10.7.2　HiRDB ファイルシステム領域を効率良く仮想化するためには

HiRDB ファイルシステム領域を効率良く仮想化する（HDP 機能を有効にする）ためには，次の条件で HiRDB ファイルシステム領域，及び RD エリアを作成することをお勧めします。

(1)　HiRDB ファイルシステム領域

次の条件をすべて満たすように，pdfmkfs コマンドで HiRDB ファイルシステム領域を作成してください。

- -i 及び-a オプションを指定しない
- -n オプションには，今後使用する可能性がある最大サイズを指定する
- ダイレクトディスクアクセスで作成する

> **参考**
>
> -i オプションを指定すると，HiRDB ファイルシステム領域全体を初期化するため，-n オプションに指定したサイズ分の実ディスクが割り当てられます。そのため，仮想化していないことになります。

(2)　RD エリア

RD エリアの自動増分の適用有無によって，RD エリアの初期サイズを決めてください。RD エリアの自動増分については，「RD エリアの自動増分」を参照してください。

- RD エリアの自動増分を適用する場合

  初期サイズを必要最低限の大きさにしてください。また，自動増分の方式は「HiRDB ファイルシステム領域内で HiRDB ファイルを増分する方式」を適用してください。この場合，RD エリアを自動増分するタイミングで，増分したサイズだけ DP プールを使用します。

- RD エリアの自動増分を適用しない場合

  初期サイズを，今後予想される最大使用サイズにしてください。この場合，RD エリアを作成した時点で RD エリアサイズ分の DP プールを使用します。

  RD エリア使用量が最大使用サイズに達した場合，RD エリアの拡張を行ってください。RD エリアの拡張については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

10.7.3　ストレージを効率良く使用するための運用

HiRDB ファイルシステム領域内の HiRDB ファイルを削除すると，HiRDB ファイルが割り当てられていた領域は使用済み領域となります。この使用済み領域は，DP プール上のディスク割り当てが解除されないかぎり，DP プール上の実ボリュームが使われないまま確保され続けます。ここでは，使用済み領域のディスク割り当てを解除する手順と，解除をお勧めする操作について説明します。

(1)　使用済み領域のディスク割り当てを解除する手順

使用済み領域のディスク割り当てを解除する手順を次に示します。

〈手順〉

1. pdfstatfs -d コマンドで，使用済み領域のディスク割り当てを解除できるかどうか確認する。※
2. pdfzeroinit コマンドで使用済み領域にゼロデータを書き込む。
3. HDP 機能のゼロデータ破棄を行う。

注※

pdfstatfs -d コマンドの実行結果の例を次に示します。

```
user area capacity                262129[kB]
remain user area capacity         262129[kB]
peak capacity                     131072[kB] ......1
available file size               262129[kB]
available file count              20
current file count                 0
remain file count                 20
free area count                    0
available expand count            20
current expand count               0 ..............2
initialize area kind        WORK
initialize user id          hirman
initialize time             Wed Nov 11 15:41:26 2009
area auto expand            NOUSE
```

〔説明〕

1. 現時点でのユーザ最大使用量が割り当て単位より大きいことを確認してください。例えば，割り当て単位が 32MB と仮定すると，現時点でのユーザ最大使用量は 128MB であるため，ディスク割り当てを解除できることになります。
2. HiRDB ファイル数が 0 であることを確認してください。

注意事項

1. pdfzeroinit コマンドを実行すると，現時点で使用している HiRDB ファイルシステム領域全体にゼロデータを書き込みます。そのため，オンライン業務への影響を考慮して実行するようにしてください。
2. ユーザ最大使用量と DP プール上のディスク割り当て量には差異があります。ユーザ最大使用量より DP プール上のディスク割り当て量が少ない場合に pdfzeroinit コマンドを実行すると，初期化によって DP プール上のディスク割り当て量が，ユーザ最大使用量に DP プールの割り当て単位を加算した値まで増加します。このため，DP プールが容量不足にならないように容量を監視し，DP プールの残容量がユーザ最大使用量と DP プール上のディスク割り当て量の差より十分に多い場合だけ，pdfzeroinit コマンドを実行してください。

3. pdfzeroinit コマンドは，初期化するサイズによって書き込み処理に時間が掛かります。なお，業務でアクセス中の HiRDB ファイルシステム領域に対しては，pdfzeroinit コマンドを実行しないでください。実行した場合，業務に影響を与えるおそれがあります。
4. HiRDB ファイルシステム領域の種別が WORK の HiRDB ファイルシステム領域に対して pdfzeroinit コマンドを実行する場合は，HiRDB が停止した状態で実行してください。HiRDB 稼働中に実行すると，作業表用ファイルを必要とする SQL 及びユティリティに影響を与えるおそれがあります。
5. 初期化対象の領域に対して，pdfstatfs-c コマンドを実行しないでください。実行した場合，pdfzeroinit コマンドで使用済みの領域を初期化できなくなります。

(2)　使用済み領域のディスク割り当て解除をお勧めする操作

一時的に大きなサイズの HiRDB ファイルシステム領域を使用する業務で，使用済み領域のディスク割り当てを解除すると，ストレージを有効に利用できるため，効果があります。なお，効果的に使用済み領域のディスク割り当てを解除するためには，次の条件を満たしている必要があります。

RD エリア，共用 RD エリア，システムファイル，及びリスト用 RD エリアの場合：

一つの HiRDB ファイルシステム領域に一つの HiRDB ファイルを作成している。

RD エリア，及び共用 RD エリアの場合：

一つの RD エリアに一つの表又は一つのインデクスだけを定義している。

使用済み領域のディスク割り当て解除をお勧めする操作を次の表に示します。

表 10–3　使用済み領域のディスク割り当て解除をお勧めする操作

pdfmkfs の-k オプションの指定値	操作の対象となる HiRDB ファイルシステム領域の種類	操作
DB 及び SDB	RD エリア及び共用 RD エリア	RD エリアの削除
		RD エリアの再初期化
		表の全行削除
		表，インデクスの削除
		表の再編成
SYS	システムファイル	システムログファイルの削除
WORK	作業表用ファイル	大きな作業表を使用する結合処理などを伴うバッチ処理
	リスト用 RD エリア	RD エリアの削除
		RD エリアの再初期化

(a)　RD エリア削除時の使用済み領域のディスク割り当て解除

RD エリアを削除するときに，使用済み領域のディスク割り当てを解除する手順を次の図に示します。

図 10–3　使用済み領域のディスク割り当てを解除する手順（RD エリアの削除時）



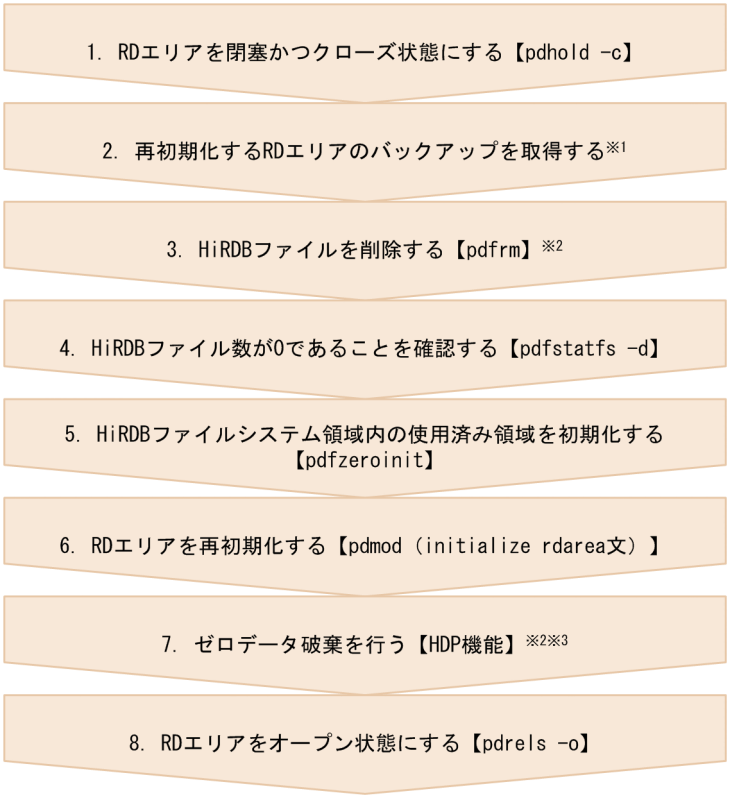
注※

HiRDB が稼働中の場合は，ほかのプロセスから HiRDB ファイルシステム領域をアクセスしていない状態で実行する必要があります。そのため，事前にサーバプロセスのリフレッシュ（pdpfresh）を行っておくことをお勧めします。

(b)　RD エリア再初期化時の使用済み領域のディスク割り当て解除

RD エリアを再初期化するときに，使用済み領域のディスク割り当てを解除する手順を次の図に示します。

図 10–4　使用済み領域のディスク割り当てを解除する手順（RD エリアの再初期化時）



注※1

HiRDB ファイルの削除から RD エリアの再初期化までの間に障害が発生した場合，回復用として使用するバックアップです。なお，このバックアップの取得は任意です。

注※2

HiRDB が稼働中の場合は，ほかのプロセスから HiRDB ファイルシステム領域をアクセスしていない状態で実行する必要があります。そのため，事前にサーバプロセスのリフレッシュ（pdpfresh）を行っておくことをお勧めします。

注※3

ゼロデータ破棄は，次の二つの条件を満たす場合に実行できます。

- 該当するディスクの領域にアクセスしていない
- HiRDB が停止状態，又は RD エリアがクローズ状態である

(c)　表の全行削除時の使用済み領域のディスク割り当て解除

表の全行削除をするときに，使用済み領域のディスク割り当てを解除するには，表の全行削除（PURGE TABLE 文）を実行した後に，「RD エリア再初期化時の使用済み領域のディスク割り当て解除」を実施してください。

(d)　表，インデクスの削除時の使用済み領域のディスク割り当て解除

表，インデクスを削除するときに，使用済み領域のディスク割り当てを解除する手順を次に示します。

〈手順〉

1.「RD エリア再初期化時の使用済み領域のディスク割り当て解除」を実施する

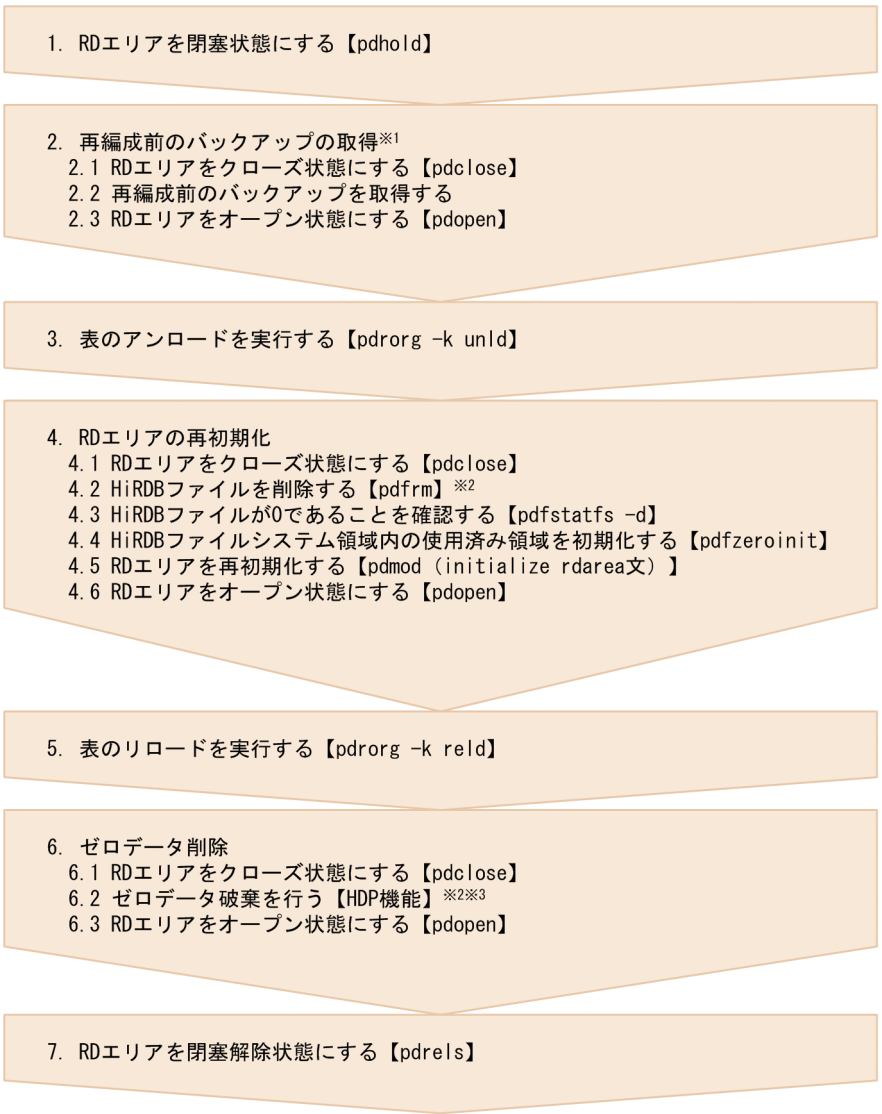
2.表又はインデクスを削除する

> **参考**
>
> 表又はインデクスの削除を先に実施してもエラーにはなりませんが，RD エリアの再初期化を先に実施した方が，表又はインデクスを削除するために HiRDB が使用するリソース（排他資源，システムログ量）を少なくできます。

(e)　表の再編成時の使用済み領域のディスク割り当て解除

表を再編成するときに，使用済み領域のディスク割り当てを解除する手順を次の図に示します。

図 10–5　使用済み領域のディスク割り当てを解除する手順（表の再編成時）



注※1

HiRDB ファイルの削除から RD エリアの再初期化までの間に障害が発生した場合，回復用として使用するバックアップです。なお，このバックアップの取得は任意です。

注※2

HiRDB が稼働中の場合は，ほかのプロセスから HiRDB ファイルシステム領域をアクセスしていない状態で実行する必要があります。そのため，事前にサーバプロセスのリフレッシュ（pdpfresh）を行っておくことをお勧めします。

注※3

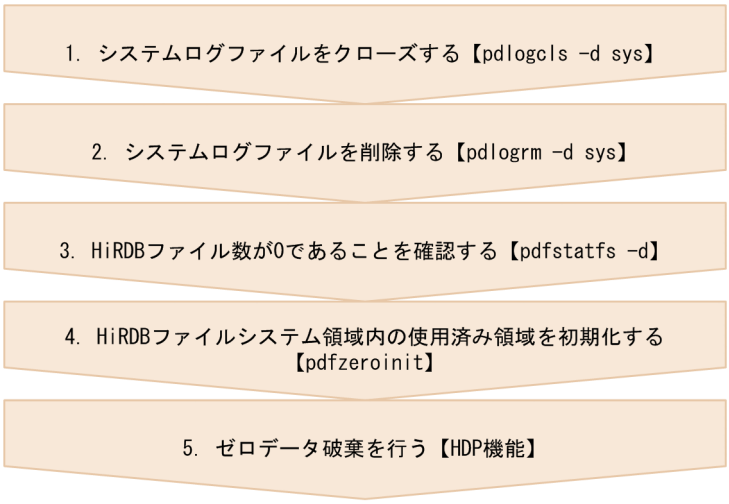
ゼロデータ破棄は，次の二つの条件を満たす場合に実行できます。

- 該当するディスクの領域にアクセスしていない
- HiRDB が停止状態，又は RD エリアがクローズ状態である

(f)　システムログファイルの削除時の使用済み領域のディスク割り当て解除

システムログファイルを削除するときに，使用済み領域のディスク割り当てを解除する手順を次の図に示します。

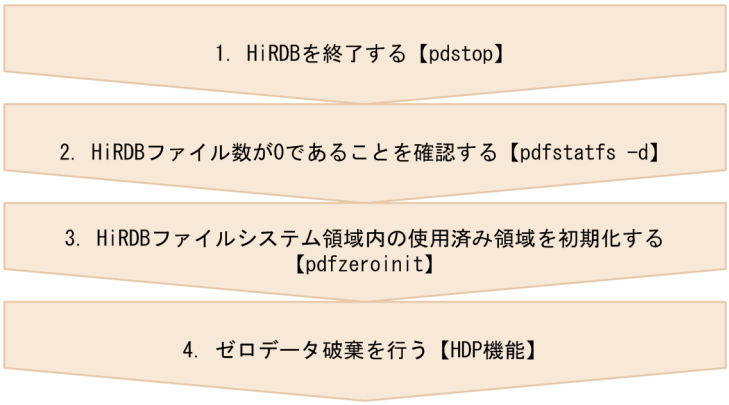
図 10–6　使用済み領域のディスク割り当てを解除する手順（システムログファイルの削除時）



(g)　作業表用ファイルを作成する SQL，ユティリティ実行後の使用済み領域のディスク割り当て解除

作業表用ファイルを作成する SQL，ユティリティの実行後に，使用済み領域のディスク割り当てを解除する手順を次の図に示します。

図 10–7　使用済み領域のディスク割り当てを解除する手順（作業表用ファイルを作成する SQL，ユティリティの実行後）



注

系切り替え機能を適用している場合で，現用系，予備系のそれぞれのサーバマシンに作業表用ファイルの HiRDB ファイルシステム領域を作成しているときは，系を切り替えた後，待機系で手順 2.～4.を実施してください。

11 システム構成を変更する方法

この章では，HiRDB/パラレルサーバのユニット構成又はサーバ構成を変更する方法について説明します。また，HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行方法について説明します。

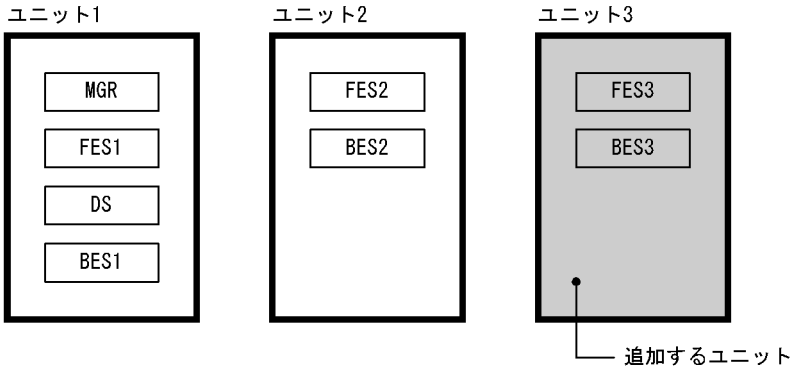
11.1　ユニットの追加

実行者　HiRDB 管理者

ここでは，HiRDB/パラレルサーバのユニットを追加する方法を説明します。

11.1.1　HiRDB の稼働中にユニットを追加する場合

HiRDB/パラレルサーバにユニット 3 を追加して，そのユニット内にフロントエンドサーバ（FES3）とバックエンドサーバ（BES3）を定義します。このときの手順を説明します。



システム構成変更コマンド（pdchgconf コマンド）を使用するため，HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

(1)　新規サーバマシンを準備します

新規サーバマシンを準備して，HiRDB をインストールしてセットアップしてください。HiRDB のインストール及びセットアップについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

なお，次に示すことを既存のユニットに合わせてください。

- HiRDB のバージョン
- アドレシングモード（32 ビットモード又は 64 ビットモード）
- 文字コード種別
- 付加 PP 及び関連製品のインストール状況

(2)　変更後の HiRDB システム定義を作成します

次に示す手順で，ユニット構成の変更を反映した HiRDB システム定義を作成してください。また，(6)で RD エリアを追加又は移動する場合はグローバルバッファの定義も修正してください。

〈手順〉

1. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
2. 使用中の HiRDB システム定義ファイルを 1 で作成したディレクトリ下にコピーします。
3. %PDDIR%¥conf¥chgconf 下の HiRDB システム定義を変更します。
4. %PDDIR%¥conf¥chgconf 下の HiRDB システム定義ファイルを%PDDIR%¥conf 下にコピーします。
   4 の操作は追加するユニットのサーバマシンに対してだけ行ってください。

注意事項

既存ユニット（ユニット 1 及びユニット 2）内の各サーバに対応している pdstart オペランドの指定値を変更しないでください。変更すると，システムファイルの初期化などを行う必要があります。pdstart オペランドの指定値を変更したときに行う操作については，マニュアル「HiRDB システム定義」の pdstart オペランドの説明を参照してください。

(3) pdloginit 及び pdstsinit コマンドでユニット 3 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes3 -f C:¥sysarea¥log01 -n 5000 -D       1
   :
pdloginit -d spd -s fes3 -f C:¥sysarea¥sync01 -n 5000 -D      2
   :
pdstsinit -s fes3 -f C:¥sysarea¥ssts01 -c 3000 -D             3
   :
pdstsinit -u UNT3 -f C:¥sysarea¥usts01 -c 3000 -D             4
   :
```

〔説明〕

1. FES3 及び BES3 のシステムログファイルを作成します。
2. FES3 及び BES3 のシンクポイントダンプファイルを作成します。
3. FES3 及び BES3 のサーバ用ステータスファイルを作成します。
4. ユニット 3 のユニット用ステータスファイルを作成します。

これらのコマンドはユニット 1（システムマネジャのユニット）からではなくユニット 3 から実行するため，pdloginit 及び pdstsinit コマンドに-D オプションを指定してください。

(4) pdconfchk コマンドで変更後の HiRDB システム定義をチェックします

```
pdconfchk -d chgconf
```

%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(5)　pdchgconf コマンドで HiRDB システム定義を変更します

```
pdchgconf
```

HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

(6)　pdmod コマンドで RD エリアを追加又は移動します

必要に応じて BES3 に RD エリアを追加又は移動してください。RD エリアの追加については，「RD エリアを作成する方法（RD エリアの追加）」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(2)でグローバルバッファの定義を変更しなかった場合は，pdbufmod コマンドでグローバルバッファを動的変更してください。

(7)　クライアント環境定義を変更します

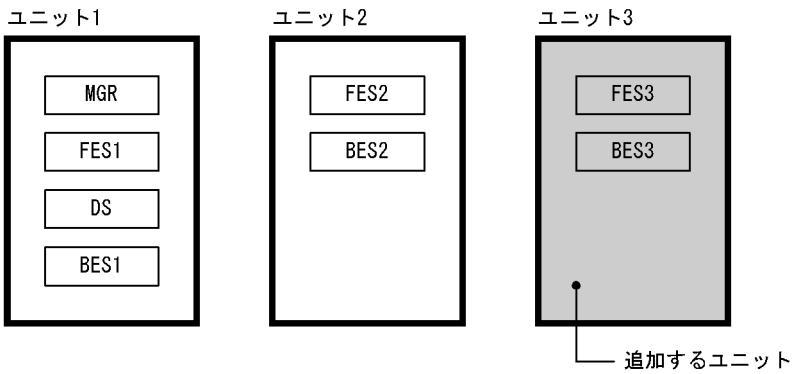
クライアント環境定義の次に示すオペランドに，追加したフロントエンドサーバ（FES3）を必要に応じて指定してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.1.2　HiRDB を終了してユニットを追加する場合

HiRDB/パラレルサーバにユニット 3 を追加して，そのユニット内にフロントエンドサーバ（FES3）とバックエンドサーバ（BES3）を定義します。このときの手順を説明します。



(1)　新規サーバマシンを準備します

新規サーバマシンを準備して，HiRDB をインストールしてセットアップしてください。HiRDB のインストール及びセットアップについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

なお，次に示すことを既存のユニットに合わせてください。

- HiRDB のバージョン
- アドレシングモード（32 ビットモード又は 64 ビットモード）
- 文字コード種別
- 付加 PP 及び関連製品のインストール状況

(2)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(3)　HiRDB システム定義を変更します

ユニット構成の変更を反映した HiRDB システム定義を作成してください。また，(7)で RD エリアを追加又は移動する場合はグローバルバッファの定義も修正してください。

注意事項

既存ユニット（ユニット 1 及びユニット 2）内の各サーバに対応している pdstart オペランドの指定値を変更しないでください。変更すると，システムファイルの初期化などを行う必要があります。pdstart オペランドの指定値を変更したときに行う操作については，マニュアル「HiRDB システム定義」の pdstart オペランドの説明を参照してください。

(4)　pdloginit 及び pdstsinit コマンドでユニット 3 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes3 -f C:¥sysarea¥log01 -n 5000        1
   :
pdloginit -d spd -s fes3 -f C:¥sysarea¥sync01 -n 5000       2
   :
pdstsinit -s fes3 -f C:¥sysarea¥ssts01 -c 3000              3
   :
pdstsinit -u UNT3 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. FES3 及び BES3 のシステムログファイルを作成します。
2. FES3 及び BES3 のシンクポイントダンプファイルを作成します。
3. FES3 及び BES3 のサーバ用ステータスファイルを作成します。
4. ユニット 3 のユニット用ステータスファイルを作成します。

(5)　pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(6)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(7)　pdmod コマンドで RD エリアを追加又は移動します

必要に応じて BES3 に RD エリアを追加又は移動してください。RD エリアの追加については，「RD エリアを作成する方法（RD エリアの追加）」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(8)　クライアント環境定義を変更します

クライアント環境定義の次に示すオペランドに，追加したフロントエンドサーバ（FES3）を必要に応じて指定してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

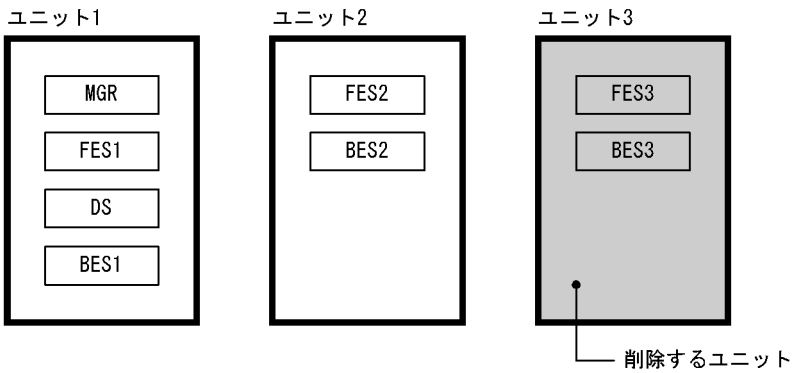
11.2　ユニットの削除

実行者　HiRDB 管理者

ここでは，HiRDB/パラレルサーバのユニットを削除する方法を説明します。

11.2.1　HiRDB の稼働中にユニットを削除する場合

HiRDB/パラレルサーバのユニット 3 を削除します。このときの手順を説明します。



システム構成変更コマンド（pdchgconf コマンド）を使用するため，HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

ポイント

- システムマネジャ又はディクショナリサーバのユニットは削除できません。
- フロントエンドサーバが 0 個になるようなユニットの削除はできません。
- バックエンドサーバが 0 個になるようなユニットの削除はできません。

(1)　pdmod コマンドで BES3 の RD エリアを削除又は移動します

RD エリアの削除については，「RD エリアを削除する方法」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(2)　クライアント環境定義を変更します

クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに削除対象のフロントエンドサーバ（FES3）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP

- PDSERVICEPORT

(3) 変更後の HiRDB システム定義を作成します

次に示す手順で，ユニット構成の変更を反映した HiRDB システム定義を作成してください。

〈手順〉

1. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
2. 使用中の HiRDB システム定義ファイルを 1 で作成したディレクトリ下にコピーします。
3. %PDDIR%¥conf¥chgconf 下の HiRDB システム定義を変更します。

(4) pdconfchk コマンドで変更後の HiRDB システム定義をチェックします

```
pdconfchk -d chgconf
```

%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(5) pdchgconf コマンドで HiRDB システム定義を変更します

```
pdchgconf
```

HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

(6) pdlogrm 及び pdstsrm コマンドでユニット 3 のシステムファイルを削除します

```
pdlogrm -d sys -s fes3 -f C:¥sysarea¥log01 -D          1
   :
pdlogrm -d spd -s fes3 -f C:¥sysarea¥sync01 -D         2
   :
pdstsrm -s fes3 -f C:¥sysarea¥ssts01 -D                3
   :
pdstsrm -u UNT3 -f C:¥sysarea¥usts01  -D               4
   :
```

〔説明〕

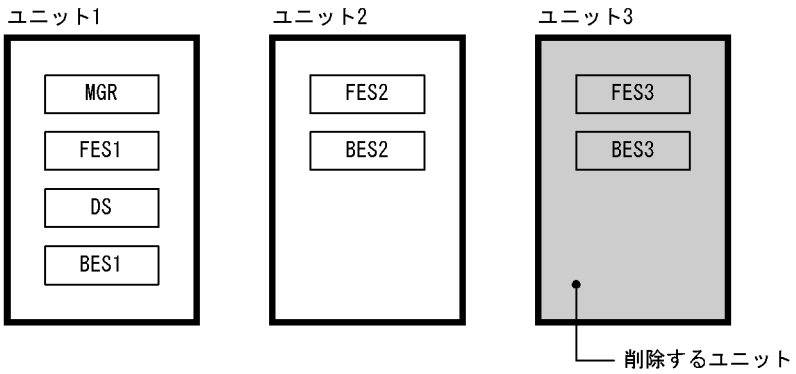
1. FES3 及び BES3 のシステムログファイルを削除します。
2. FES3 及び BES3 のシンクポイントダンプファイルを削除します。
3. FES3 及び BES3 のサーバ用ステータスファイルを削除します。
4. ユニット 3 のユニット用ステータスファイルを削除します。

これらのコマンドはユニット 1（システムマネジャのユニット）からではなくユニット 3 から実行するため，pdlogrm 及び pdstsrm コマンドに-D オプションを指定してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.2.2　HiRDB を終了してユニットを削除する場合

HiRDB/パラレルサーバのユニット 3 を削除します。このときの手順を説明します。



ポイント

- システムマネジャ又はディクショナリサーバのユニットは削除しないでください。
- フロントエンドサーバが 0 個になるようなユニットの削除はしないでください。
- バックエンドサーバが 0 個になるようなユニットの削除はしないでください。

(1)　pdmod コマンドで BES3 の RD エリアを削除又は移動します

RD エリアの削除については，「RD エリアを削除する方法」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(2)　クライアント環境定義を変更します

クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに削除対象のフロントエンドサーバ（FES3）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

(3) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(4) pdlogrm 及び pdstsrm コマンドでユニット 3 のシステムファイルを削除します

```
pdlogrm -d sys -s fes3 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s fes3 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s fes3 -f C:¥sysarea¥ssts01                3
   :
pdstsrm -u UNT3 -f C:¥sysarea¥usts01                4
   :
```

〔説明〕

1. FES3 及び BES3 のシステムログファイルを削除します。
2. FES3 及び BES3 のシンクポイントダンプファイルを削除します。
3. FES3 及び BES3 のサーバ用ステータスファイルを削除します。
4. ユニット 3 のユニット用ステータスファイルを削除します。

(5) HiRDB システム定義を変更します

ユニット構成の変更を反映した HiRDB システム定義を作成してください。

(6) pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(7) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

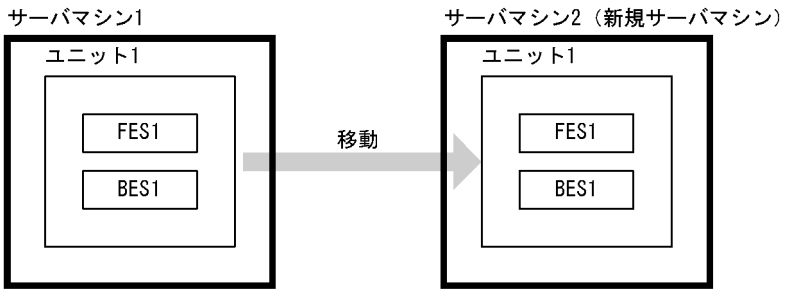
11.3　ユニットの移動

実行者　HiRDB 管理者

ここでは，HiRDB/パラレルサーバのユニットを移動する方法を説明します。

11.3.1　HiRDB を終了してユニットを移動する場合（RD エリアを再作成する場合）

HiRDB/パラレルサーバのユニットを移動します。このときの手順を説明します。



参考

システムマネジャのユニットを移動すると影響が大きいため，移動しないことをお勧めします。

(1)　新規サーバマシンを準備します

新規サーバマシンを準備して，HiRDB をインストールしてセットアップしてください。HiRDB のインストール及びセットアップについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

なお，次に示すことを既存のユニットに合わせてください。

- HiRDB のバージョン
- アドレシングモード（32 ビットモード又は 64 ビットモード）
- 文字コード種別
- 付加 PP 及び関連製品のインストール状況

(2)　pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(3) pdrorg コマンドで BES1 の表データを RD エリア単位にアンロードします

RD エリア単位のアンロードについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(4) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(5) pdlogls コマンドでユニット 1 のシステムログファイルの状態をチェックします

```
pdlogls -d sys -s bes1
```

(6) pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードします

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog01
```

(7) pdlogrm 及び pdstsrm コマンドで移動元ユニットのシステムファイルを削除します

```
pdlogrm -d sys -s fes1 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s fes1 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s fes1 -f C:¥sysarea¥ssts01                3
   :
pdstsrm -u UNT1 -f C:¥sysarea¥usts01                4
   :
```

〔説明〕

1. 移動元ユニット（FES1 及び BES1）のシステムログファイルを削除します。
2. 移動元ユニット（FES1 及び BES1）のシンクポイントダンプファイルを削除します。
3. 移動元ユニット（FES1 及び BES1）のサーバ用ステータスファイルを削除します。
4. 移動元ユニットのユニット用ステータスファイルを削除します。

(8) HiRDB システム定義を変更します

ユニット構成の変更を反映した HiRDB システム定義を作成してください。

(9) pdloginit 及び pdstsinit コマンドでユニット 1 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes1 -f C:¥sysarea¥log01 -n 5000        1
   :
pdloginit -d spd -s fes1 -f C:¥sysarea¥sync01 -n 5000       2
   :
pdstsinit -s fes1 -f C:¥sysarea¥ssts01 -c 3000              3
   :
pdstsinit -u UNT1 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. FES1 及び BES1 のシステムログファイルを作成します。
2. FES1 及び BES1 のシンクポイントダンプファイルを作成します。
3. FES1 及び BES1 のサーバ用ステータスファイルを作成します。
4. ユニット 1 のユニット用ステータスファイルを作成します。

(10) pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(11) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(12) pdmod コマンドでユニット 1 の RD エリアを再初期化します

RD エリアの再初期化については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください。

(13) pdrorg コマンドでユニット 1 の表データを RD エリア単位にリロードします

(2)で作成したアンロードデータファイルを入力データにします。RD エリア単位のリロードについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(14) pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(15)　クライアント環境定義を変更します

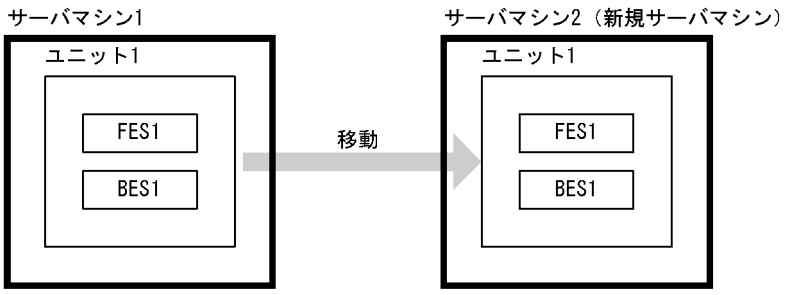
クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに移動対象のフロントエンドサーバ（FES1）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.3.2　HiRDB を終了してユニットを移動する場合（RD エリアを再作成しない場合）

HiRDB/パラレルサーバのユニットを移動します。このときの手順を説明します。



参考

システムマネジャのユニットを移動すると影響が大きいため，移動しないことをお勧めします。

(1)　新規サーバマシンを準備します

新規サーバマシンを準備して，HiRDB をインストールしてセットアップしてください。HiRDB のインストール及びセットアップについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

なお，次に示すことを既存のユニットに合わせてください。

- HiRDB のバージョン
- アドレシングモード（32 ビットモード又は 64 ビットモード）
- 文字コード種別
- 付加 PP 及び関連製品のインストール状況

(2) pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(3) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(4) pdlogls コマンドでユニット 1 のシステムログファイルの状態をチェックします

```
pdlogls -d sys -s bes1
```

(5) pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードします

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog01
```

(6) pdlogrm 及び pdstsrm コマンドで移動元ユニットのシステムファイルを削除します

```
pdlogrm -d sys -s fes1 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s fes1 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s fes1 -f C:¥sysarea¥ssts01               3
   :
pdstsrm -u UNT1 -f C:¥sysarea¥usts01               4
   :
```

〔説明〕

1. 移動元ユニット（FES1 及び BES1）のシステムログファイルを削除します。
2. 移動元ユニット（FES1 及び BES1）のシンクポイントダンプファイルを削除します。
3. 移動元ユニット（FES1 及び BES1）のサーバ用ステータスファイルを削除します。
4. 移動元ユニットのユニット用ステータスファイルを削除します。

(7) HiRDB システム定義を変更します

ユニット構成の変更を反映した HiRDB システム定義を作成してください。

(8) pdloginit 及び pdstsinit コマンドでユニット 1 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes1 -f C:¥sysarea¥log01 -n 5000        1
   :
pdloginit -d spd -s fes1 -f C:¥sysarea¥sync01 -n 5000       2
   :
pdstsinit -s fes1 -f C:¥sysarea¥ssts01 -c 3000              3
   :
pdstsinit -u UNT1 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. FES1 及び BES1 のシステムログファイルを作成します。
2. FES1 及び BES1 のシンクポイントダンプファイルを作成します。
3. FES1 及び BES1 のサーバ用ステータスファイルを作成します。
4. ユニット 1 のユニット用ステータスファイルを作成します。

(9) pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(10) RD エリアの HiRDB ファイルシステム領域を移動元のユニットから移動します

HiRDB ファイルシステム領域のファイルパス名称を移動元ユニットと異なるパス名にする場合，pdchpathf コマンドを使用して変更してください。pdchpathf コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(11) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(12) (10)で pdchpathf コマンドを実行した場合，pdchpathn コマンドを実行します

pdchpathn コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(13) pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(14)　クライアント環境定義を変更します

クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに移動対象のフロントエンドサーバ（FES1）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

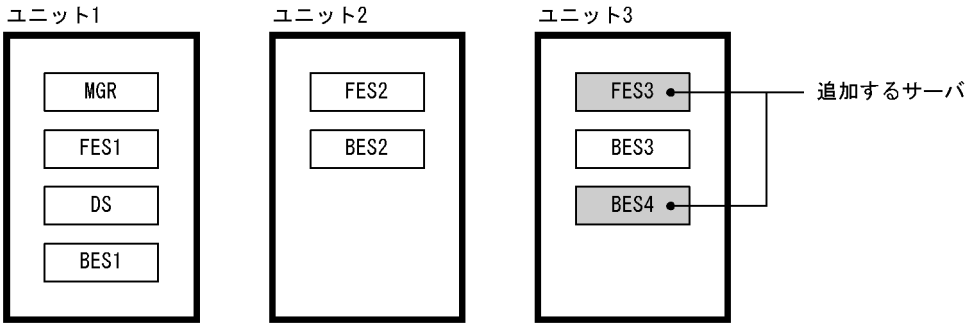
11.4　サーバの追加

実行者　HiRDB 管理者

ここでは，HiRDB/パラレルサーバのサーバを追加する方法を説明します。

11.4.1　HiRDB の稼働中にサーバを追加する場合

HiRDB/パラレルサーバのユニット 3 にフロントエンドサーバ（FES3）とバックエンドサーバ（BES4）を追加します。このときの手順を説明します。



システム構成変更コマンド（pdchgconf コマンド）を使用するため，HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

(1)　ユニット 3 のメモリ所要量を確認します

サーバを追加するため，ユニット 3 のメモリ所要量を見積もり直してください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　変更後の HiRDB システム定義を作成します

次に示す手順で，サーバ構成の変更を反映した HiRDB システム定義を作成してください。また，(6)で RD エリアを追加又は移動する場合はグローバルバッファの定義も修正してください。

〈手順〉

1. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
2. 使用中の HiRDB システム定義ファイルを 1 で作成したディレクトリ下にコピーします。
3. %PDDIR%¥conf¥chgconf 下の HiRDB システム定義を変更します。

(3) pdloginit 及び pdstsinit コマンドで FES3 及び BES4 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes3 -f C:¥sysarea¥log01 -n 5000 -D       1
   :
pdloginit -d spd -s fes3 -f C:¥sysarea¥sync01 -n 5000 -D      2
   :
pdstsinit -s fes3 -f C:¥sysarea¥ssts01 -c 3000 -D             3
   :
```

〔説明〕

1. FES3 及び BES4 のシステムログファイルを作成します。
2. FES3 及び BES4 のシンクポイントダンプファイルを作成します。
3. FES3 及び BES4 のサーバ用ステータスファイルを作成します。

これらのコマンドはユニット 1（システムマネジャのユニット）からではなくユニット 3 から実行するため，pdloginit 及び pdstsinit コマンドに-D オプションを指定してください。

(4) pdconfchk コマンドで変更後の HiRDB システム定義をチェックします

```
pdconfchk -d chgconf
```

%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(5) pdchgconf コマンドで HiRDB システム定義を変更します

```
pdchgconf
```

HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

(6) pdmod コマンドで RD エリアを追加又は移動します

必要に応じて BES4 に RD エリアを追加又は移動してください。RD エリアの追加については，「RD エリアを作成する方法（RD エリアの追加）」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(2)でグローバルバッファの定義を変更しなかった場合は，pdbufmod コマンドでグローバルバッファを動的変更してください。

(7) クライアント環境定義を変更します

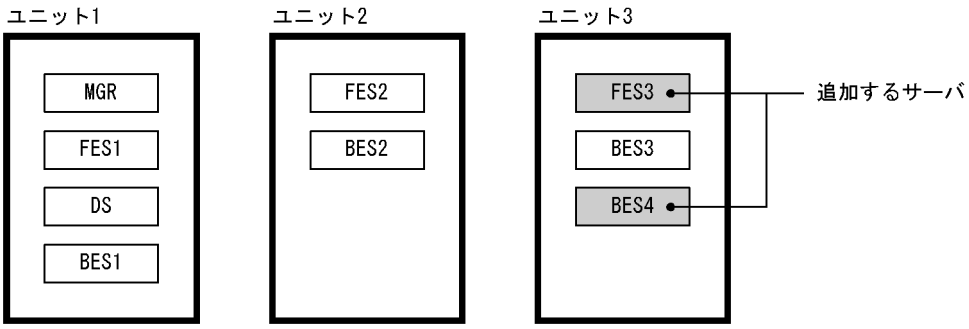
クライアント環境定義の次に示すオペランドに，追加したフロントエンドサーバ（FES3）を必要に応じて指定してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.4.2　HiRDB を終了してサーバを追加する場合

HiRDB/パラレルサーバのユニット 3 にフロントエンドサーバ（FES3）とバックエンドサーバ（BES4）を追加します。このときの手順を説明します。



(1)　ユニット 3 のメモリ所要量を確認します

サーバを追加するため，ユニット 3 のメモリ所要量を見積もり直してください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(3)　pdlogls コマンドでユニット 3 のシステムログファイルの状態をチェックします

```
pdlogls -d sys -s bes3
```

(4)　pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードします

```
pdlogunld -d sys -s bes3 -g log01 -o C:¥unld¥unldlog01
```

システムログをアンロードする運用をしていない場合は，pdcopy コマンドで全 RD エリアのバックアップを取得してください。バックアップの取得については，「バックアップの取得例」を参照してください。

(5) pdlogrm 及び pdstsrm コマンドでユニット 3 のシステムファイルを削除します

```
pdlogrm -d sys -s bes3 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s bes3 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s bes3 -f C:¥sysarea¥ssts01                3
   :
pdstsrm -u UNT3 -f C:¥sysarea¥usts01                4
   :
```

〔説明〕

1. BES3 のシステムログファイルを削除します。
2. BES3 のシンクポイントダンプファイルを削除します。
3. BES3 のサーバ用ステータスファイルを削除します。
4. ユニット 3 のユニット用ステータスファイルを削除します。

(6) HiRDB システム定義を変更します

サーバ構成の変更を反映した HiRDB システム定義を作成してください。また，(10)で RD エリアを追加又は移動する場合はグローバルバッファの定義も修正してください。

(7) pdloginit 及び pdstsinit コマンドでユニット 3 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes3 -f C:¥sysarea¥log01 -n 5000        1
   :
pdloginit -d spd -s fes3 -f C:¥sysarea¥sync01 -n 5000       2
   :
pdstsinit -s fes3 -f C:¥sysarea¥ssts01 -c 3000              3
   :
pdstsinit -u UNT3 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. FES3，BES3，及び BES4 のシステムログファイルを作成します。
2. FES3，BES3，及び BES4 のシンクポイントダンプファイルを作成します。
3. FES3，BES3，及び BES4 のサーバ用ステータスファイルを作成します。
4. ユニット 3 のユニット用ステータスファイルを作成します。

ステータスファイルを再作成したため，RD エリアの障害閉塞状態は引き継がれません。したがって，HiRDB の正常開始後，必要に応じて pdhold コマンドで RD エリアを再度閉塞してください。

(8)　pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(9)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(10)　pdmod コマンドで RD エリアを追加又は移動します

必要に応じて BES4 に RD エリアを追加又は移動してください。RD エリアの追加については，「RD エリアを作成する方法（RD エリアの追加）」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(11)　クライアント環境定義を変更します

クライアント環境定義の次に示すオペランドに，追加したフロントエンドサーバ（FES3）を必要に応じて指定してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

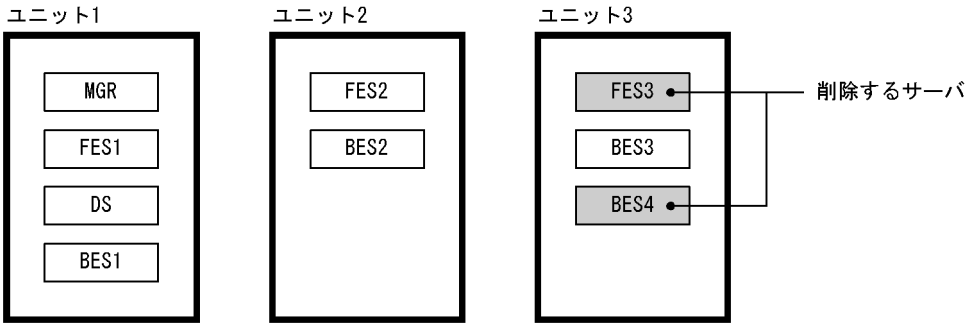
11.5　サーバの削除

実行者　HiRDB 管理者

ここでは，HiRDB/パラレルサーバのサーバを削除する方法を説明します。

11.5.1　HiRDB の稼働中にサーバを削除する場合

HiRDB/パラレルサーバのフロントエンドサーバ（FES3）とバックエンドサーバ（BES4）を削除します。このときの手順を説明します。



システム構成変更コマンド（pdchgconf コマンド）を使用するため，HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

ポイント

- システムマネジャ又はディクショナリサーバは削除できません。
- フロントエンドサーバが 0 個になるようなサーバの削除はできません。
- バックエンドサーバが 0 個になるようなサーバの削除はできません。

(1)　pdmod コマンドで BES4 の RD エリアを削除又は移動します

RD エリアの削除については，「RD エリアを削除する方法」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(2)　クライアント環境定義を変更します

クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに削除対象のフロントエンドサーバ（FES3）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP

- PDSERVICEPORT

(3) 変更後の HiRDB システム定義を作成します

次に示す手順で，サーバ構成の変更を反映した HiRDB システム定義を作成してください。

〈手順〉

1. %PDDIR%¥conf¥chgconf ディレクトリを作成します。
2. 使用中の HiRDB システム定義ファイルを 1 で作成したディレクトリ下にコピーします。
3. %PDDIR%¥conf¥chgconf 下の HiRDB システム定義を変更します。

(4) pdconfchk コマンドで変更後の HiRDB システム定義をチェックします

```
pdconfchk -d chgconf
```

%PDDIR%¥conf¥chgconf ディレクトリ下の HiRDB システム定義のチェックを行います。エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(5) pdchgconf コマンドで HiRDB システム定義を変更します

```
pdchgconf
```

HiRDB システム定義を変更後の HiRDB システム定義に置き換えます。

(6) pdlogrm 及び pdstsrm コマンドで FES3 及び BES4 のシステムファイルを削除します

```
pdlogrm -d sys -s fes3 -f C:¥sysarea¥log01 -D          1
   :
pdlogrm -d spd -s fes3 -f C:¥sysarea¥sync01 -D         2
   :
pdstsrm -s fes3 -f C:¥sysarea¥ssts01 -D                3
   :
```

〔説明〕

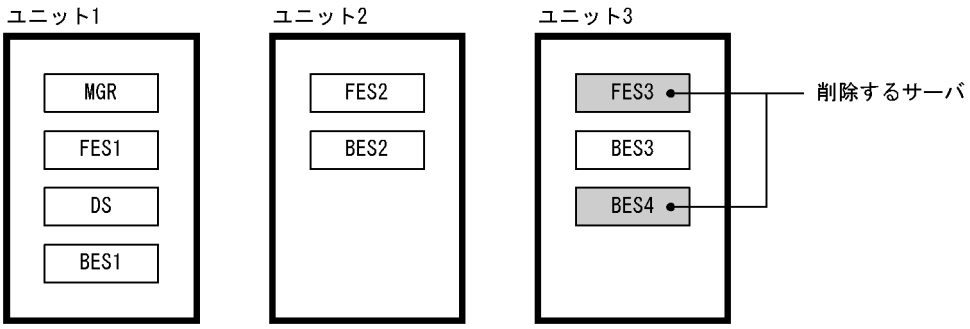
1. FES3 及び BES4 のシステムログファイルを削除します。
2. FES3 及び BES4 のシンクポイントダンプファイルを削除します。
3. FES3 及び BES4 のサーバ用ステータスファイルを削除します。

これらのコマンドはユニット 1（システムマネジャのユニット）からではなくユニット 3 から実行するため，pdlogrm 及び pdstsrm コマンドに-D オプションを指定してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.5.2　HiRDB を終了してサーバを削除する場合

HiRDB/パラレルサーバのフロントエンドサーバ（FES3）とバックエンドサーバ（BES4）を削除します。このときの手順を説明します。



ポイント

- システムマネジャ又はディクショナリサーバは削除しないでください。
- フロントエンドサーバが 0 個になるようなサーバの削除はしないでください。
- バックエンドサーバが 0 個になるようなサーバの削除はしないでください。

(1)　pdmod コマンドで BES4 の RD エリアを削除又は移動します

RD エリアの削除については，「RD エリアを削除する方法」を参照してください。RD エリアの移動については，「RD エリアを移動する方法（RD エリアの移動）」を参照してください。

(2)　クライアント環境定義を変更します

クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに削除対象のフロントエンドサーバ（FES3）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

(3)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(4) pdlogls コマンドでユニット 3 のシステムログファイルの状態をチェックします

```
pdlogls -d sys -s bes3
```

(5) pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードします

```
pdlogunld -d sys -s bes3 -g log01 -o C:¥unld¥unldlog01
```

(6) pdlogrm 及び pdstsrm コマンドでユニット 3 のシステムファイルを削除します

```
pdlogrm -d sys -s fes3 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s fes3 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s fes3 -f C:¥sysarea¥ssts01                3
   :
pdstsrm -u UNT3 -f C:¥sysarea¥usts01                4
   :
```

〔説明〕

1. FES3，BES3，及び BES4 のシステムログファイルを削除します。
2. FES3，BES3，及び BES4 のシンクポイントダンプファイルを削除します。
3. FES3，BES3，及び BES4 のサーバ用ステータスファイルを削除します。
4. ユニット 3 のユニット用ステータスファイルを削除します。

(7) HiRDB システム定義を変更します

サーバ構成の変更を反映した HiRDB システム定義を作成してください。

(8) pdloginit 及び pdstsinit コマンドでユニット 3 に必要なシステムファイルを作成します

```
pdloginit -d sys -s bes3 -f C:¥sysarea¥log01 -n 5000        1
   :
pdloginit -d spd -s bes3 -f C:¥sysarea¥sync01 -n 5000       2
   :
pdstsinit -s bes3 -f C:¥sysarea¥ssts01 -c 3000              3
   :
```

```
pdstsinit -u UNT3 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. BES3 のシステムログファイルを作成します。
2. BES3 のシンクポイントダンプファイルを作成します。
3. BES3 のサーバ用ステータスファイルを作成します。
4. ユニット 3 のユニット用ステータスファイルを作成します。

ステータスファイルを再作成したため，RD エリアの障害閉塞状態は引き継がれません。したがって，HiRDB の正常開始後，必要に応じて pdhold コマンドで RD エリアを再度閉塞してください。

(9) pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(10) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

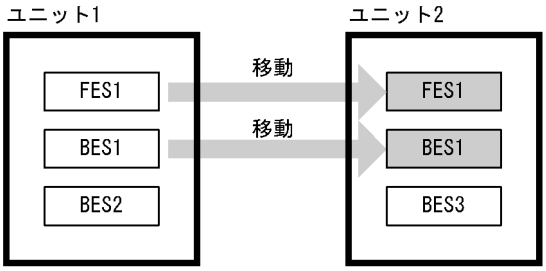
11.6　サーバの移動

実行者　HiRDB 管理者

ここでは，HiRDB/パラレルサーバのサーバを移動する方法を説明します。

11.6.1　HiRDB を終了してサーバを移動する場合(RD エリアを再作成する場合)

HiRDB/パラレルサーバのフロントエンドサーバ（FES1）及びバックエンドサーバ（BES1）を移動します。このときの手順を説明します。



参考

システムマネジャを移動すると影響が大きいため，移動しないことをお勧めします。

(1)　ユニット 2 のメモリ所要量を確認します

サーバを移動するため，ユニット 2 のメモリ所要量を見積もり直してください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(3)　pdrorg コマンドで BES1 の表データを RD エリア単位にアンロードします

RD エリア単位のアンロードについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(4)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(5) pdlogls コマンドで，移動元ユニット内のシステムログファイルの状態をチェックします

```
pdlogls -d sys -s bes1
pdlogls -d sys -s bes2
```

(6) pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードします

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog01
pdlogunld -d sys -s bes2 -g log02 -o C:¥unld¥unldlog02
```

システムログをアンロードする運用をしていない場合は，pdcopy コマンドで全 RD エリアのバックアップを取得してください。バックアップの取得については，「バックアップの取得例」を参照してください。

(7) pdlogrm 及び pdstsrm コマンドでユニット 1 及びユニット 2 の全システムファイルを削除します

```
pdlogrm -d sys -s fes1 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s fes1 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s fes1 -f C:¥sysarea¥ssts01               3
   :
pdstsrm -u UNT1 -f C:¥sysarea¥usts01               4
   :
```

〔説明〕

1. FES1，BES1～BES3 のシステムログファイルを削除します。
2. FES1，BES1～BES3 のシンクポイントダンプファイルを削除します。
3. FES1，BES1～BES3 のサーバ用ステータスファイルを削除します。
4. ユニット 1 及びユニット 2 のユニット用ステータスファイルを削除します。

(8) HiRDB システム定義を変更します

サーバ構成の変更を反映した HiRDB システム定義を作成してください。

(9) pdloginit 及び pdstsinit コマンドでユニット 1 及びユニット 2 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes1 -f C:¥sysarea¥log01 -n 5000       1
   :
pdloginit -d spd -s fes1 -f C:¥sysarea¥sync01 -n 5000      2
```

```
   :
pdstsinit -s fes1 -f C:¥sysarea¥ssts01 -c 3000              3
   :
pdstsinit -u UNT1 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. FES1，BES1～BES3 のシステムログファイルを作成します。
2. FES1，BES1～BES3 のシンクポイントダンプファイルを作成します。
3. FES1，BES1～BES3 のサーバ用ステータスファイルを作成します。
4. ユニット 1 及びユニット 2 のユニット用ステータスファイルを作成します。

ステータスファイルを再作成したため，RD エリアの障害閉塞状態は引き継がれません。したがって，HiRDB の正常開始後，必要に応じて pdhold コマンドで RD エリアを再度閉塞してください。

(10) pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(11) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(12) pdmod コマンドで BES1 の RD エリアを再初期化します

RD エリアの再初期化については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください。

(13) pdrorg コマンドで，移動したユニット内の表データを RD エリア単位にリロードします

(2)で作成したアンロードデータファイルを入力データにします。RD エリア単位のリロードについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(14) pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(15)　クライアント環境定義を変更します

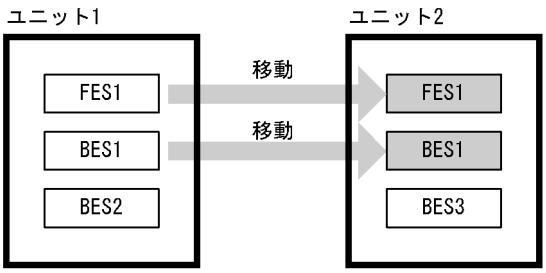
クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに移動対象のフロントエンドサーバ（FES1）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.6.2　HiRDB を終了してサーバを移動する場合(RD エリアを再作成しない場合)

HiRDB/パラレルサーバのフロントエンドサーバ（FES1）及びバックエンドサーバ（BES1）を移動します。このときの手順を説明します。



参考

システムマネジャを移動すると影響が大きいため，移動しないことをお勧めします。

(1)　ユニット 2 のメモリ所要量を確認します

サーバを移動するため，ユニット 2 のメモリ所要量を見積もり直してください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(3)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

必ず正常終了させてください。

(4) pdlogls コマンドで，移動元ユニット内のシステムログファイルの状態をチェックします

```
pdlogls -d sys -s bes1
pdlogls -d sys -s bes2
```

(5) pdlogunld コマンドでアンロード待ち状態のシステムログファイルをアンロードします

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog01
pdlogunld -d sys -s bes2 -g log02 -o C:¥unld¥unldlog02
```

システムログをアンロードする運用をしていない場合は，pdcopy コマンドで全 RD エリアのバックアップを取得してください。バックアップの取得については，「バックアップの取得例」を参照してください。

(6) pdlogrm 及び pdstsrm コマンドで，ユニット 1 及びユニット 2 の全システムファイルを削除します

```
pdlogrm -d sys -s fes1 -f C:¥sysarea¥log01          1
   :
pdlogrm -d spd -s fes1 -f C:¥sysarea¥sync01         2
   :
pdstsrm -s fes1 -f C:¥sysarea¥ssts01               3
   :
pdstsrm -u UNT1 -f C:¥sysarea¥usts01               4
   :
```

〔説明〕

1. FES1，BES1～BES3 のシステムログファイルを削除します。
2. FES1，BES1～BES3 のシンクポイントダンプファイルを削除します。
3. FES1，BES1～BES3 のサーバ用ステータスファイルを削除します。
4. ユニット 1 及びユニット 2 のユニット用ステータスファイルを削除します。

(7) HiRDB システム定義を変更します

サーバ構成の変更を反映した HiRDB システム定義を作成してください。

(8) pdloginit 及び pdstsinit コマンドで，ユニット 1 及びユニット 2 に必要なシステムファイルを作成します

```
pdloginit -d sys -s fes1 -f C:¥sysarea¥log01 -n 5000        1
   :
pdloginit -d spd -s fes1 -f C:¥sysarea¥sync01 -n 5000       2
   :
pdstsinit -s fes1 -f C:¥sysarea¥ssts01 -c 3000              3
   :
pdstsinit -u UNT1 -f C:¥sysarea¥usts01 -c 3000              4
   :
```

〔説明〕

1. FES1，BES1～BES3 のシステムログファイルを作成します。
2. FES1，BES1～BES3 のシンクポイントダンプファイルを作成します。
3. FES1，BES1～BES3 のサーバ用ステータスファイルを作成します。
4. ユニット 1 及びユニット 2 のユニット用ステータスファイルを作成します。

ステータスファイルを再作成したため，RD エリアの障害閉塞状態は引き継がれません。したがって，HiRDB の正常開始後，必要に応じて pdhold コマンドで RD エリアを再度閉塞してください。

(9) pdconfchk コマンドで HiRDB システム定義をチェックします

```
pdconfchk
```

エラーがある場合は HiRDB システム定義を修正して，再度 pdconfchk コマンドを実行してください。

(10) RD エリアの HiRDB ファイルシステム領域を移動元のユニットから移動します

HiRDB ファイルシステム領域のファイルパス名称を移動元ユニットと異なるパス名にする場合，pdchpathf コマンドを使用して変更してください。pdchpathf コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(11) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(12) (10)で pdchpathf コマンドを実行している場合は，pdchpathn コマンドを実行します

pdchpathn コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(13)　pdcopy コマンドで全 RD エリアのバックアップを取得します

バックアップの取得については，「バックアップの取得例」を参照してください。

(14)　クライアント環境定義を変更します

クライアント環境定義に次に示すオペランドを指定している場合は注意が必要です。これらのオペランドに移動対象のフロントエンドサーバ（FES1）を指定している場合は指定値を変更してください。

- PDFESHOST
- PDSERVICEGRP
- PDSERVICEPORT

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

11.7　HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行方法

ここでは，HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行方法について説明します。説明する項目は次のとおりです。

- 移行作業をする前に
- 移行手順
- 複数のユーザ用 RD エリアを異なるバックエンドサーバに移行する場合の注意事項

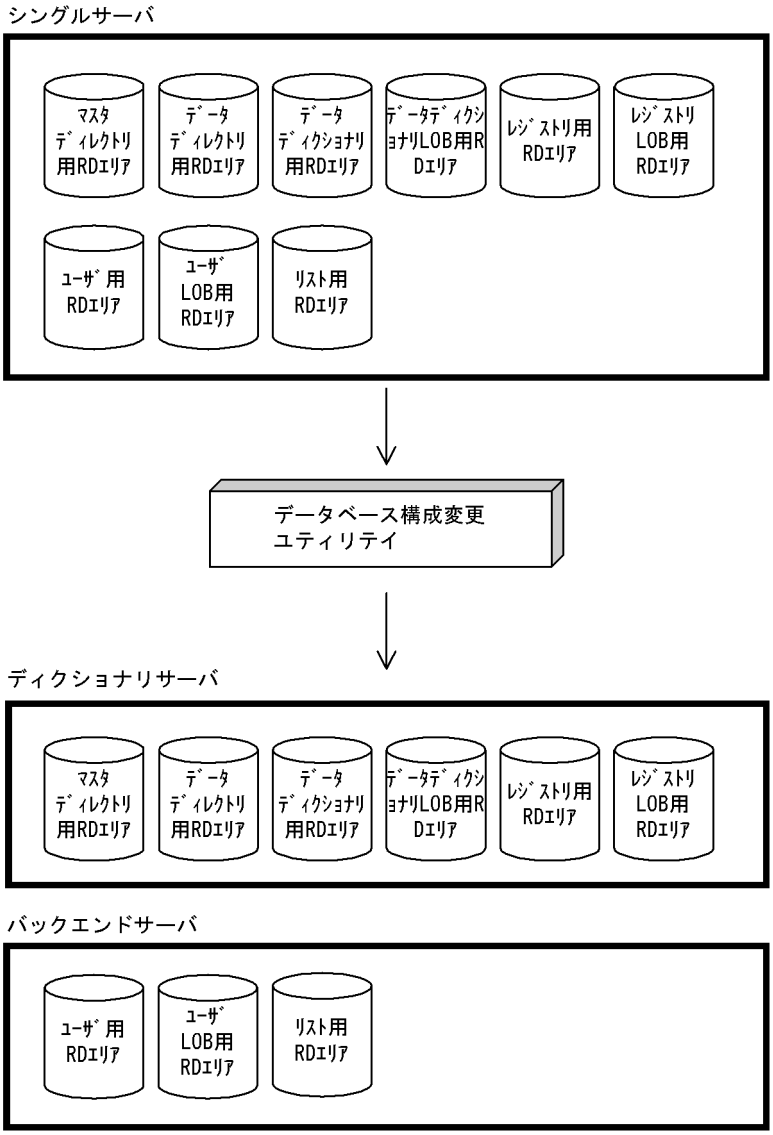
11.7.1　移行作業をする前に

(1)　移行はデータベース構成変更ユティリティ（pdmod コマンド）の機能を使用します

HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行は，データベース構成変更ユティリティ（**pdmod** コマンド）の機能を使用します。データベース構成変更ユティリティの **alter HiRDB mode to parallel** 文を実行して，シングルサーバに設置している RD エリアをディクショナリサーバ及びバックエンドサーバに移行します。

データベース構成変更ユティリティによる RD エリアの移行を次の図に示します。

図 11–1　データベース構成変更ユティリティによる RD エリアの移行



(2)　注意事項

(a)　移行をする前にバックアップを取得します

移行をする前に，次に示す RD エリアを構成する HiRDB ファイル又は HiRDB ファイルシステム領域のバックアップを **pdfbkup** コマンドで取得してください。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア

(b)　ストアドプロシジャ及びストアドファンクションがある場合

移行をすると，HiRDB/シングルサーバのときに作成したストアドプロシジャ及びストアドファンクション（ユーザ定義関数のストアドファンクション）が無効になります。したがって，移行後に，ストアドプロシジャ及びストアドファンクションを再作成してください。

ストアドプロシジャ及びストアドファンクションを再作成するには，HiRDB/パラレルサーバに移行後，**ALTER ROUTINE** を実行してください。

(c)　ユーザ LOB 用 RD エリアがある場合

LOB データを格納しているユーザ LOB 用 RD エリアと，その LOB データの LOB 列構成基表が格納されているユーザ用 RD エリアを同じバックエンドサーバに配置してください。同じバックエンドサーバに配置しないと，RD エリアをアクセスしたときにバックエンドサーバが異常終了することがあります。

したがって，移行後に，ディクショナリ表（SQL_RDAREA，SQL_TABLES，SQL_DIV_COLUMN）を参照して，同じバックエンドサーバに RD エリアが配置されているかどうかを確認してください。

(d)　リスト用 RD エリアがある場合

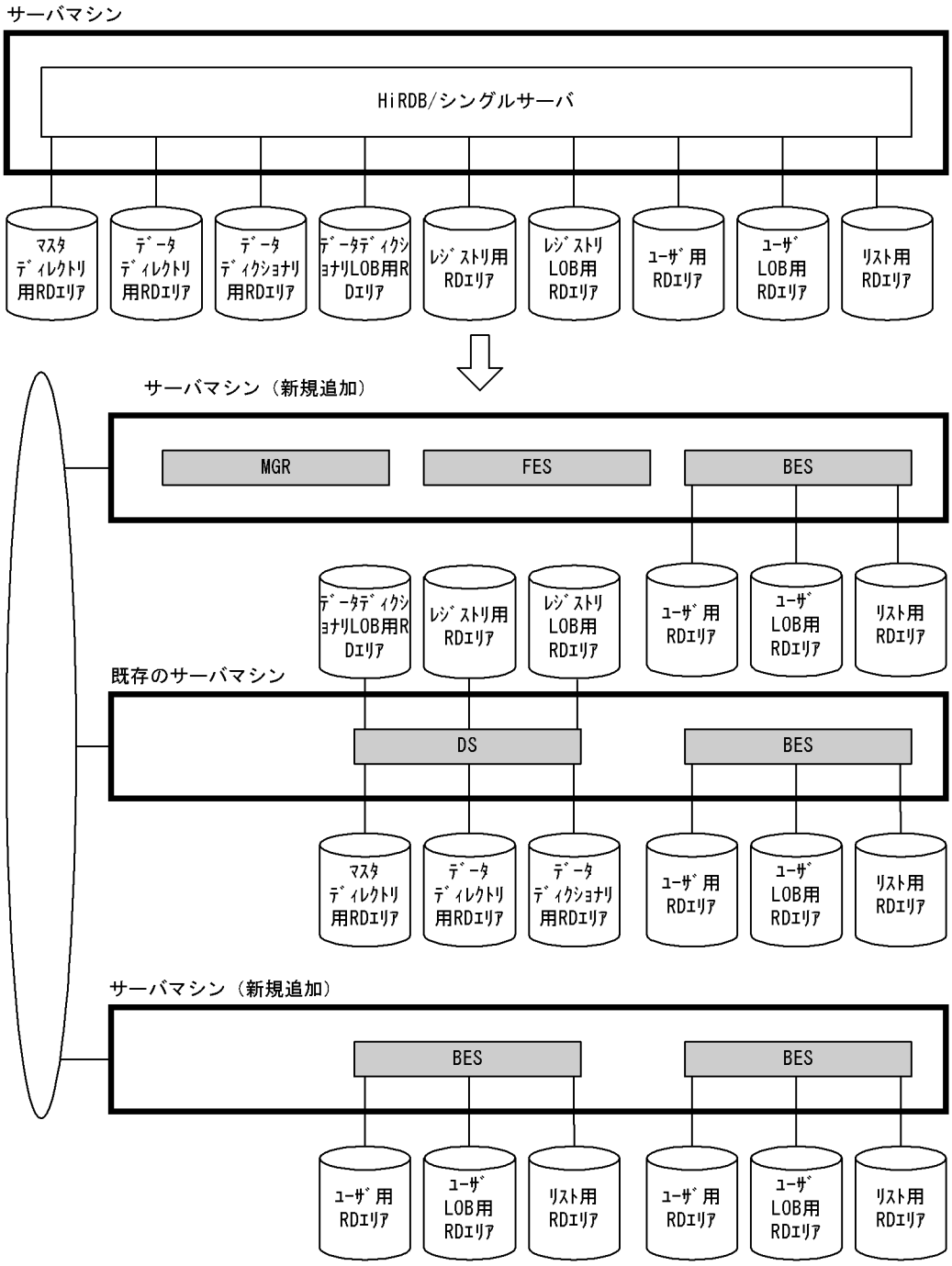
リスト用 RD エリアと，リストの基表が格納されているユーザ用 RD エリアを同じバックエンドサーバに配置してください。

11.7.2　移行手順

実行者　HiRDB 管理者

HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行モデルを次の図に示します。

図 11–2　HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行モデル



HiRDB/シングルサーバから HiRDB/パラレルサーバへの移行手順を次に示します。

(1)　データベース構成変更ユティリティを実行します

データベース構成変更ユティリティ（**pdmod** コマンド）の **alter HiRDB mode to parallel 文**で，RD エリアを HiRDB/パラレルサーバ用に構成変更します。このときのユティリティ使用例については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) HiRDB を正常終了させます

pdstop コマンドで，HiRDB/シングルサーバを正常終了させます。必ず正常終了させてください。

(3) HiRDB/パラレルサーバの環境設定をします

HiRDB/パラレルサーバの環境設定をします。HiRDB/パラレルサーバの環境設定については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

また，環境設定するときの注意事項を次に示します。

- HiRDB 運用ディレクトリは，HiRDB/シングルサーバと異なるディレクトリにしてください。
- HiRDB/シングルサーバで使用していたシステムファイルを使用しないで，HiRDB/パラレルサーバ用にシステムファイルを新規作成してください。
- ディクショナリサーバは，HiRDB/シングルサーバが使用したサーバマシン（既存のサーバマシン）に定義してください。

(4) ユーザ用 RD エリア，ユーザ LOB 用 RD エリア，及びリスト用 RD エリアを別サーバマシンに移行します

この時点では，ユーザ用 RD エリア，ユーザ LOB 用 RD エリア，及びリスト用 RD エリアは，HiRDB/シングルサーバとして使用したサーバマシン（既存のサーバマシン）にあります。別のサーバマシンにユーザ用 RD エリア，ユーザ LOB 用 RD エリア，及びリスト用 RD エリアを移行する方法には，次に示す方法があります。

- HiRDB の運用コマンドを使用する方法

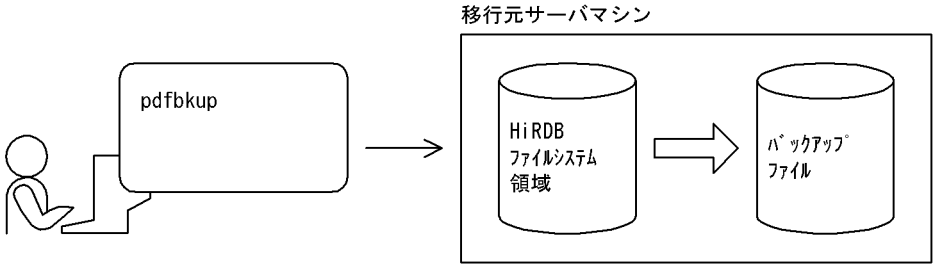
(a) HiRDB の運用コマンドを使用する方法

HiRDB の運用コマンドを使用すると，HiRDB ファイルシステム領域単位又は HiRDB ファイル単位に移行できます。

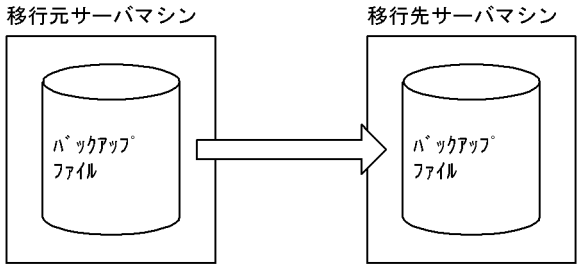
HiRDB の運用コマンドを使用して，RD エリアの内容を別サーバマシンに移行する方法を次の図に示します。

図 11–3　RD エリアの内容を別サーバマシンに移行する方法（HiRDB の運用コマンドを使用する場合）

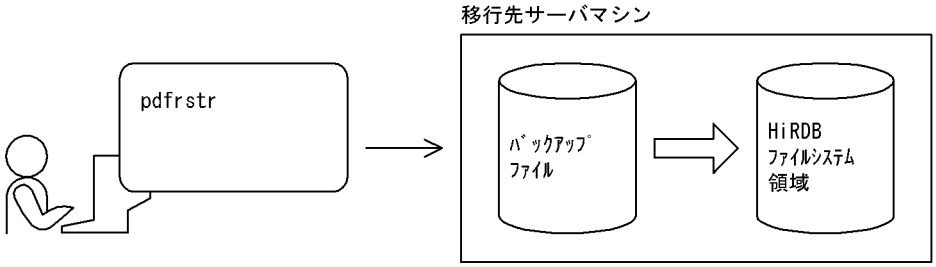
１．pdfbkupのコマンドで,移行するHiRDBファイルシステム領域のバックアップを作成します。



２．バックアップファイルを移行先のサーバマシンに作成します。



３．pdfrstrコマンドで,移行先のサーバマシンにHiRDBファイルシステム領域の内容をリストアします。
このとき，HiRDBファイル単位でもリストアできます。



注　３．の操作をする前に，移行先のサーバマシンにHiRDBファイルシステム領域が作成されているかを確認してください。作成されていない場合は，HiRDBファイルシステム領域を作成した後，３．の操作をしてください。

●移行のときの規則及び留意事項

1. HiRDB 識別子は変更できません（システム共通定義の pd_system_id オペランドの指定値は変更できません）。
2. 移行先のサーバマシンに作成する HiRDB ファイルシステム領域名称は，移行元の HiRDB ファイルシステム領域名称と同じにしてください。
3. 移行先のサーバマシンに作成する HiRDB ファイルシステム領域長は，移行元の HiRDB ファイルシステム領域長と同じか，又は大きくしてください。
4. HiRDB ファイル単位でリストアする場合，移行先のサーバマシンに作成する HiRDB ファイルシステム領域長には，その HiRDB ファイルを格納できるだけの大きさが必要です。

5. 移行が完了しても，移行元の HiRDB ファイルシステム領域をすぐに削除しないでください。HiRDB/パラレルサーバの開始及び動作確認後に，移行元の HiRDB ファイルシステム領域を削除してください。

(5) HiRDB を正常開始します

pdstart コマンドで，HiRDB/パラレルサーバを正常開始します。

(6) ストアドプロシジャ及びストアドファンクションを再作成します

ストアドプロシジャ及びストアドファンクション（ユーザ定義関数のストアドファンクション）を作成していた場合，**ALTER ROUTINE** でストアドプロシジャ及びストアドファンクションを再作成します。

HiRDB/シングルサーバから HiRDB/パラレルサーバに移行すると，すべてのストアドプロシジャ及びストアドファンクションが無効になるため，この操作が必要になります。

11.7.3 複数のユーザ用 RD エリアを異なるバックエンドサーバに移行する場合の注意事項

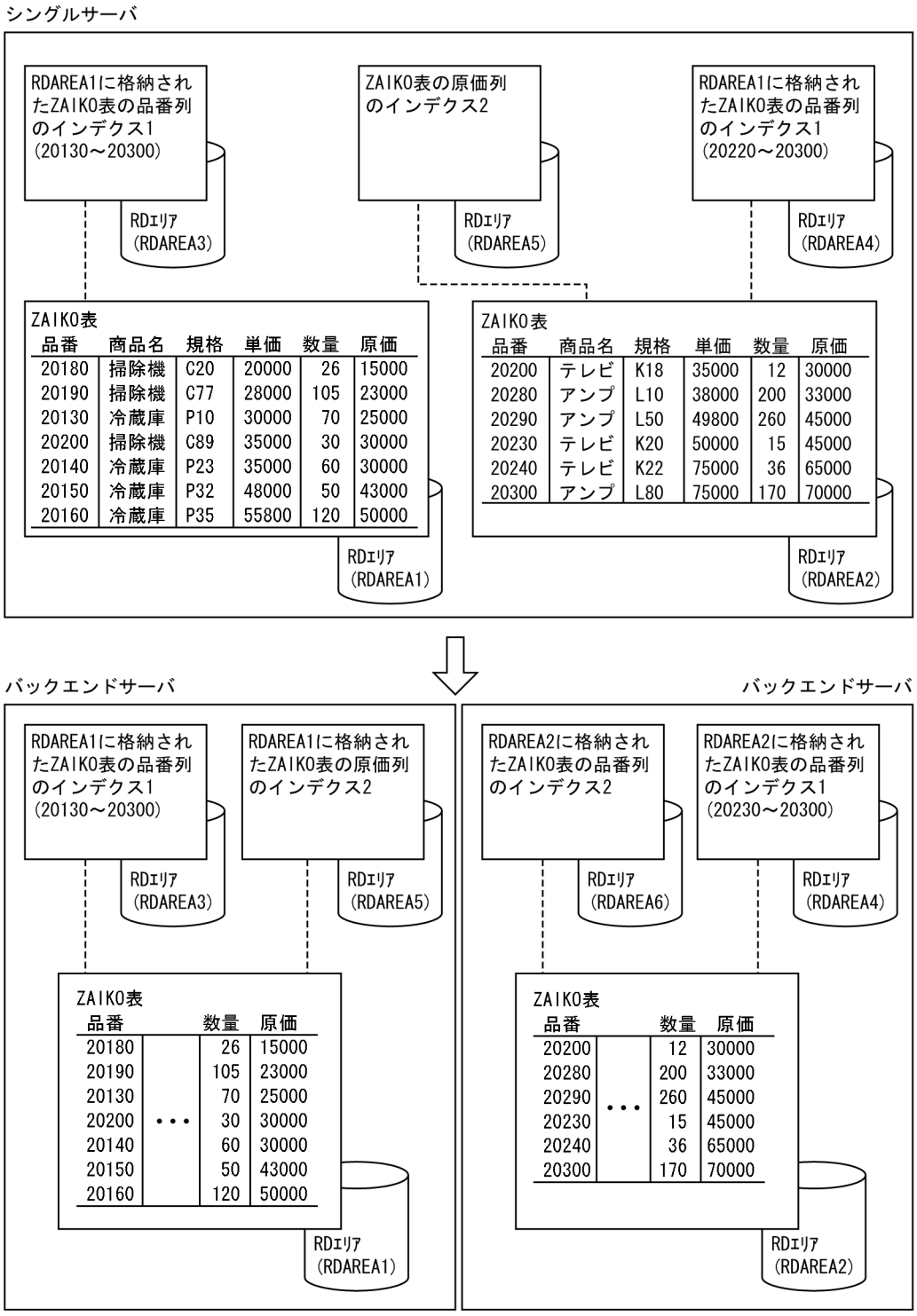
(1) 非横分割表の場合

表とインデクスを格納する RD エリアが異なる場合，それらの RD エリアを同じバックエンドサーバに設置してください。データベース構成変更ユティリティの実行後，ディクショナリ表（SQL_RDAREAS 表，SQL_TABLES 表，及び SQL_INDEXES 表）を検索して，同じバックエンドサーバに RD エリアが移行されていることを確認してください。

(2) 横分割表の場合

1. 表とインデクスを格納する RD エリアが異なる場合，それらの RD エリアを同じバックエンドサーバに設置してください。データベース構成変更ユティリティの実行後，ディクショナリ表（SQL_RDAREAS 表，SQL_TABLES 表，SQL_DIV_TABLE 表，SQL_INDEXES 表，及び SQL_DIV_INDEX 表）を検索して，同じバックエンドサーバに RD エリアが移行されていることを確認してください。
2. 表に非横分割インデクスを定義している場合，その表は複数のバックエンドサーバに移行できません。したがって，複数のバックエンドサーバに移行する場合は，移行する前に非横分割インデクスを削除し，移行した後に非横分割インデクスを定義し直してください。表に非横分割インデクスを定義している場合の移行方法を次の図に示します。

図 11–4　表に非横分割インデクスを定義している場合の移行方法



〔説明〕

ZAIKO 表は RDAREA1 と RDAREA2 に横分割されています。インデクス 1 は横分割されています。インデクス 2 は横分割されていません。ZAIKO 表を複数のバックエンドサーバに移行する場合は，インデクス 2 を削除した後に移行します。その後，インデクス 2 を定義します。

11.8　負荷均等化を目的としたバックエンドサーバの移動

ここでは，バックエンドサーバの負荷均等化を目的としたバックエンドサーバの移動方法について説明します。

11.8.1　シナリオによるバックエンドサーバの負荷均等化

月末などにピークを迎える業務がある場合，特定のバックエンドサーバに負荷が集中し，業務の処理速度が遅くなることがあります。この場合，バックエンドサーバの負荷変動に応じてバックエンドサーバの配置を動的に変更し，各ユニットに掛かる負荷を均等化できます。

なお，ここで説明するバックエンドサーバの負荷均等化は，次に示す方法で行うことを想定しています。

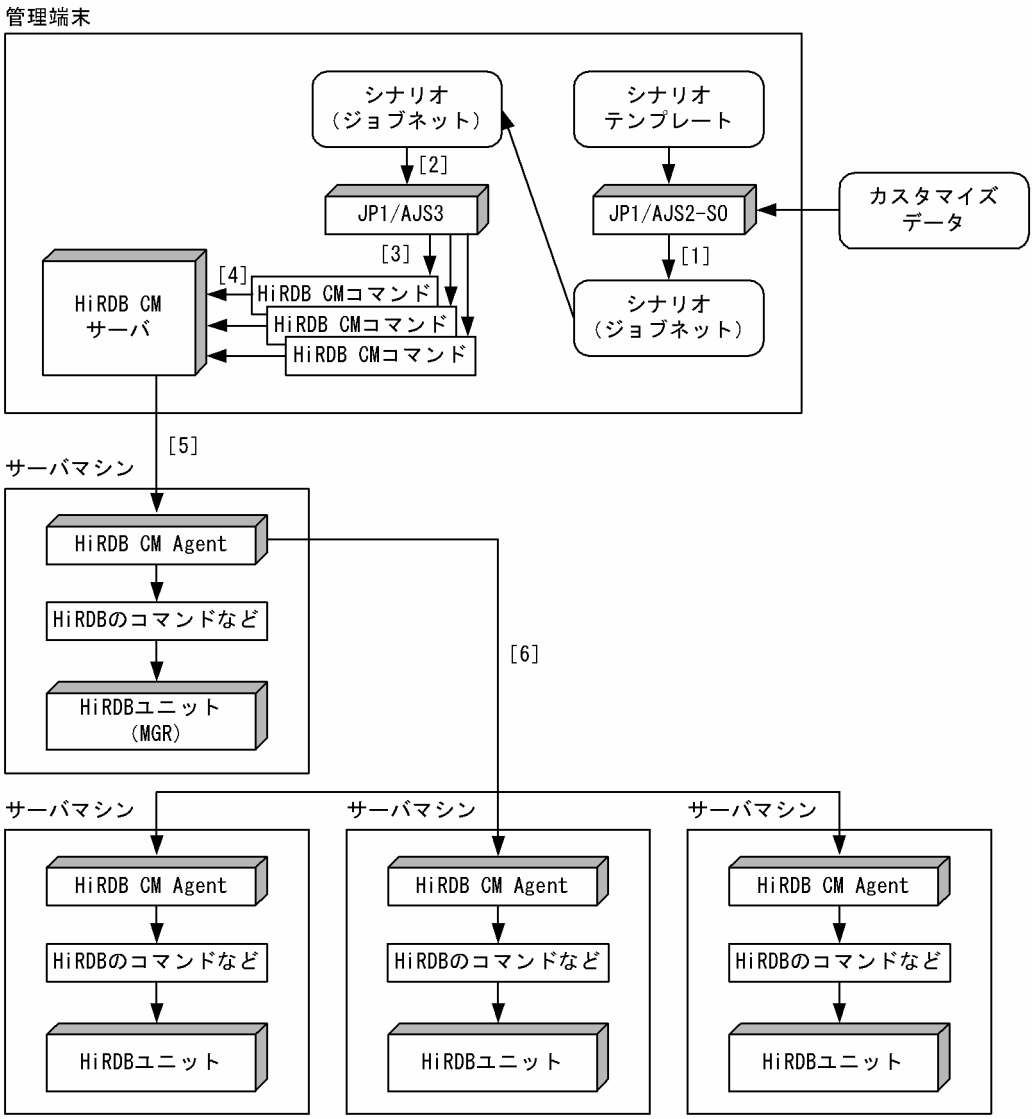
- **影響分散スタンバイレス型系切り替え機能**を使用してバックエンドサーバを移動する
- **JP1/AJS2-SO** で生成したシナリオを **JP1/AJS3**（JP1/AJS2 も含みます）で実行する
- **HiRDB CM** を使用したシナリオを実行する

HiRDB のポリシー運用の場合，シナリオテンプレートは次の図に示すようにシナリオ化されて実行されます。

用語の説明

- **シナリオ**：シナリオジョブに実行順序を付けてネットワーク化したオブジェクトです。シナリオにシナリオ登録を実行すると JP1/AJS3 に登録され，実行できます。
- **シナリオテンプレート**：シナリオの実行に必要な環境情報（パラメータ）をシナリオ変数として定義した，手順だけで構成されるシナリオのひな形です。シナリオを作成するときの部品となります。
- **シナリオ変数の値**：シナリオテンプレートからシナリオを生成するときに補完するユーザ（システム）固有の情報です（HiRDB 識別子，ホスト名，サーバ名，移動先のユニット識別子など）。
- **ジョブネット**：JP1/AJS3 で実行順序を関連づけたジョブの集まりです。ジョブネットを実行すると，ジョブネット中のジョブが実行順序に従って自動的に実行されます。

図 11–5　バックエンドサーバの負荷均等化のシナリオ運用形態



〔説明〕

1. JP1/AJS2-SO にシナリオテンプレートとシナリオ変数の値を入力してシナリオ（JP1/AJS3 のジョブネット）を生成します。シナリオを最初に実行するときに 1 回だけ行います。
2. 生成したシナリオ（ジョブネット）を JP1/AJS3 に登録して実行します。
3. ジョブネットは HiRDB CM のコマンドを実行します。
4. HiRDB CM のコマンドは HiRDB CM サーバに処理を要求します。
5. HiRDB CM サーバは，システムマネジャがあるサーバマシンの HiRDB CM Agent に処理要求をします。
6. システムマネジャがあるサーバマシンの HiRDB CM Agent は，HiRDB のコマンドなどを実行したり，ほかのユニットの HiRDB CM Agent に処理要求をしたりします。

HiRDB CM については，HiRDB CM のヘルプを参照してください。

11.8.2　前提条件及び対象とする業務の条件

シナリオによるバックエンドサーバの負荷均等化を行う場合の前提条件，及び対象とする業務の条件を次に示します。

前提条件

- 複数ユニット構成の HiRDB/パラレルサーバである
- 影響分散スタンバイレス型系切り替え機能を適用している（又は適用可能である）

対象とする業務の条件

- 一つの HiRDB で二つ以上の異なる業務を実行する
- 各業務で主に使用するバックエンドサーバ（負荷が掛かるバックエンドサーバ）が全バックエンドサーバのうちの一部である
- 各業務で主に使用するバックエンドサーバ（負荷が掛かるバックエンドサーバ）が業務ごとに互いに排他的である※
- 各業務の負荷のピーク期間（時間）が重ならない
- 各業務の負荷変動が予測又は想定できる
- 移動対象のバックエンドサーバを使用する業務のトランザクションは短時間で終了する（少なくともバックエンドサーバ移動時に長時間決着しないトランザクションがない）
- バックエンドサーバの移動中にそのバックエンドサーバを使用するトランザクションが一時的に滞留したり，又はキャンセルされたりすることを許容できる

注※

バックエンドサーバの移動による各ユニットの負荷均等化が目的のため，各業務が完全に排他的である必要はありません。

参考

- バックエンドサーバの移動をするときはバックエンドサーバの負荷が高い状態です。バックエンドサーバの負荷が高いときにバックエンドサーバを移動するため，トランザクションが一時保留されたり，又はキャンセルされたりして業務に影響が出ます。
- バックエンドサーバの移動先ユニットも想定以上に負荷が高い可能性があります。このため，バックエンドサーバを移動しても，バックエンドサーバの負荷が均等化されないことがあります。

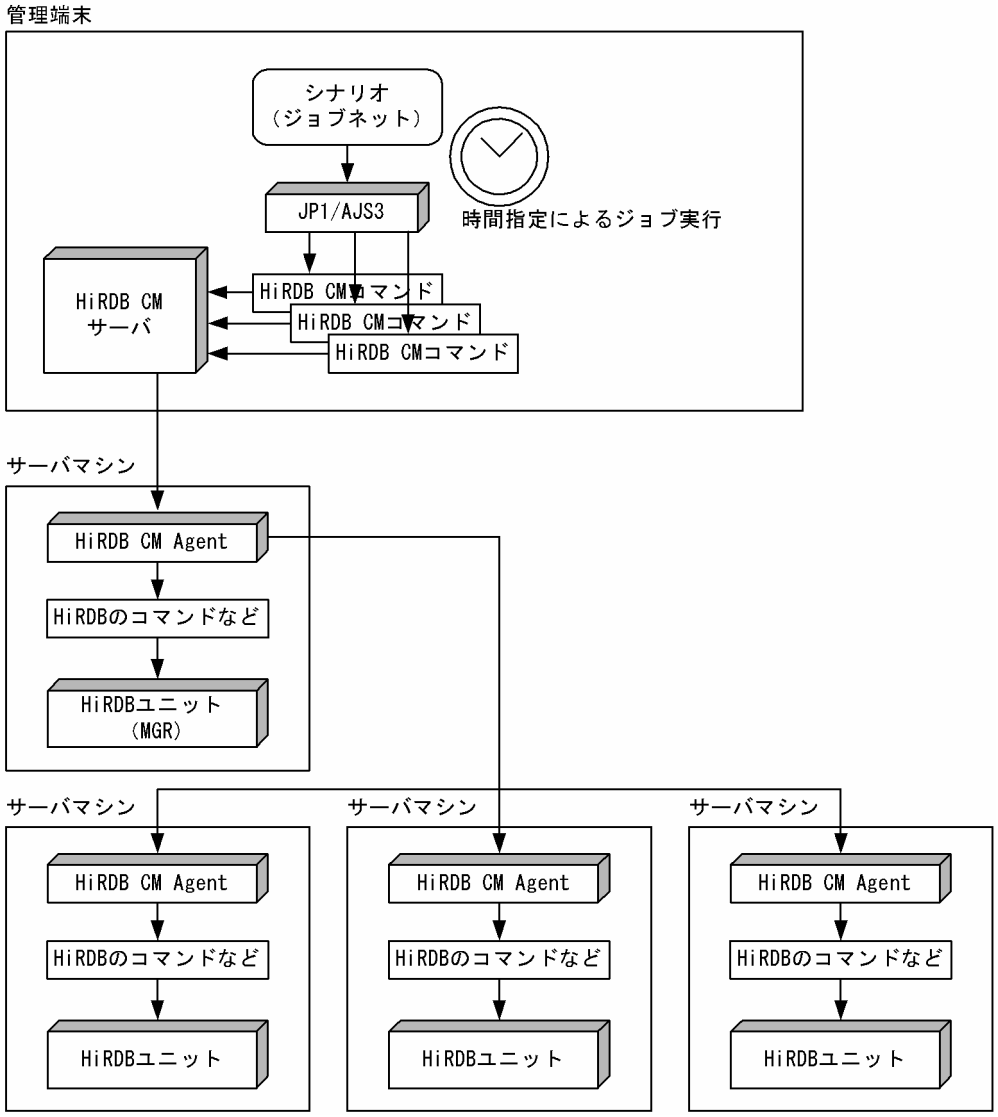
11.8.3　シナリオ運用方法

バックエンドサーバの負荷均等化のシナリオについて説明します。

(1)　JP1/AJS3 によるシナリオ定期実行

JP1/AJS3 を使用し，バックエンドサーバの負荷均等化を行う時期や，時間帯を設定してシナリオを実行します。JP1/AJS3 によるシナリオ定期実行を次の図に示します。

図 11–6　JP1/AJS3 によるシナリオ定期実行



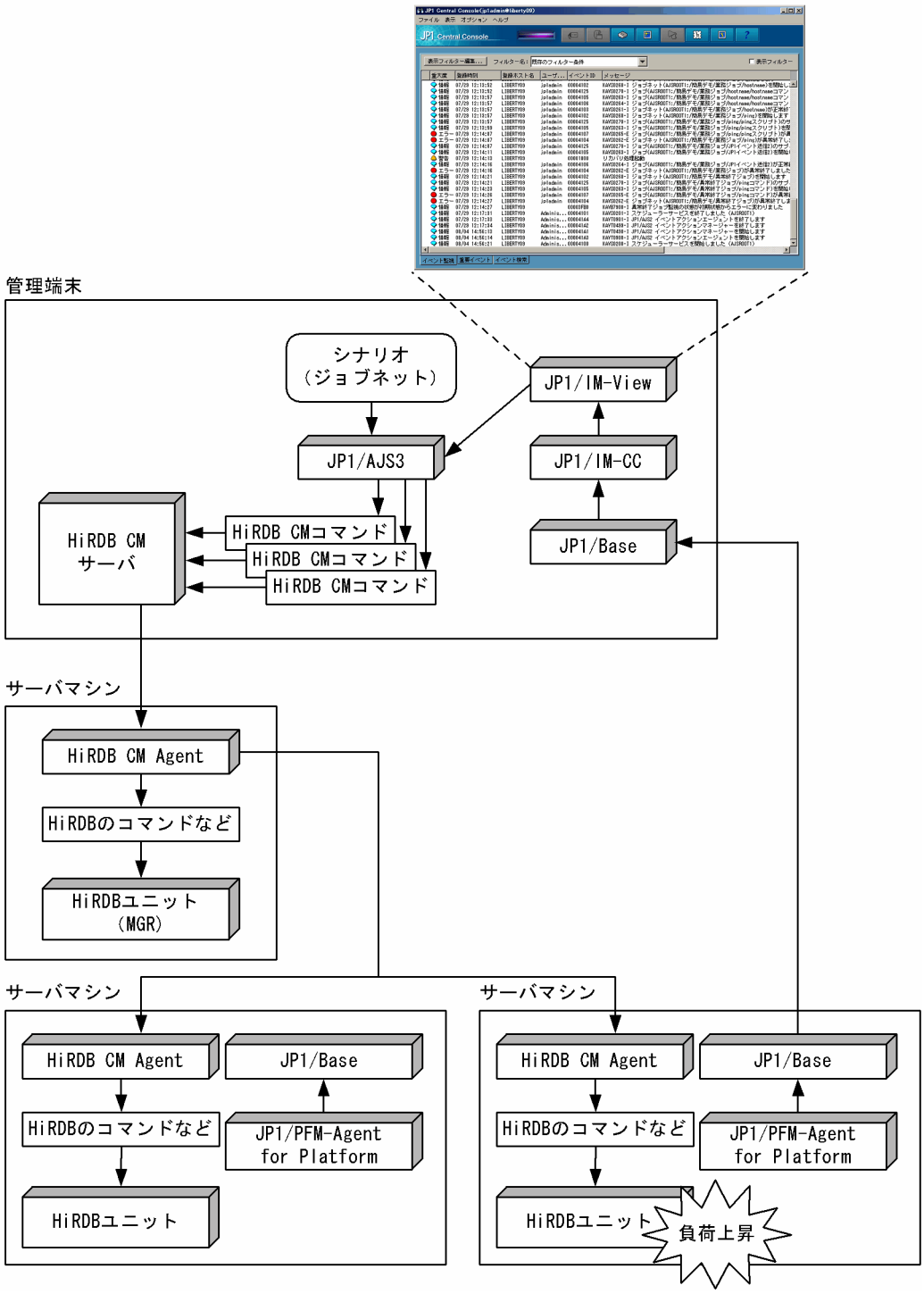
特徴

- バックエンドサーバの負荷に偏りが生じる時期や，時間帯が予測できる場合に有効です。
- バックエンドサーバを移動する時期や，時間帯を設定できるため，運用計画が立てやすいです。
- ポリシー運用の最小製品構成でこの運用が行えます。

(2) JP1/PFM によるバックエンドサーバの負荷監視とユーザ操作によるシナリオ実行

JP1/PFM を使用してバックエンドサーバの負荷を監視します。負荷が設定値を超えた場合，それをイベントとして JP1/IM-CC に通知して JP1/IM-View に表示します。これをオペレータが監視して GUI 操作でシナリオを実行します。JP1/PFM によるバックエンドサーバの負荷監視とユーザ操作によるシナリオ実行を次の図に示します。

図 11–7　JP1/PFM によるバックエンドサーバの負荷監視とユーザ操作によるシナリオ実行



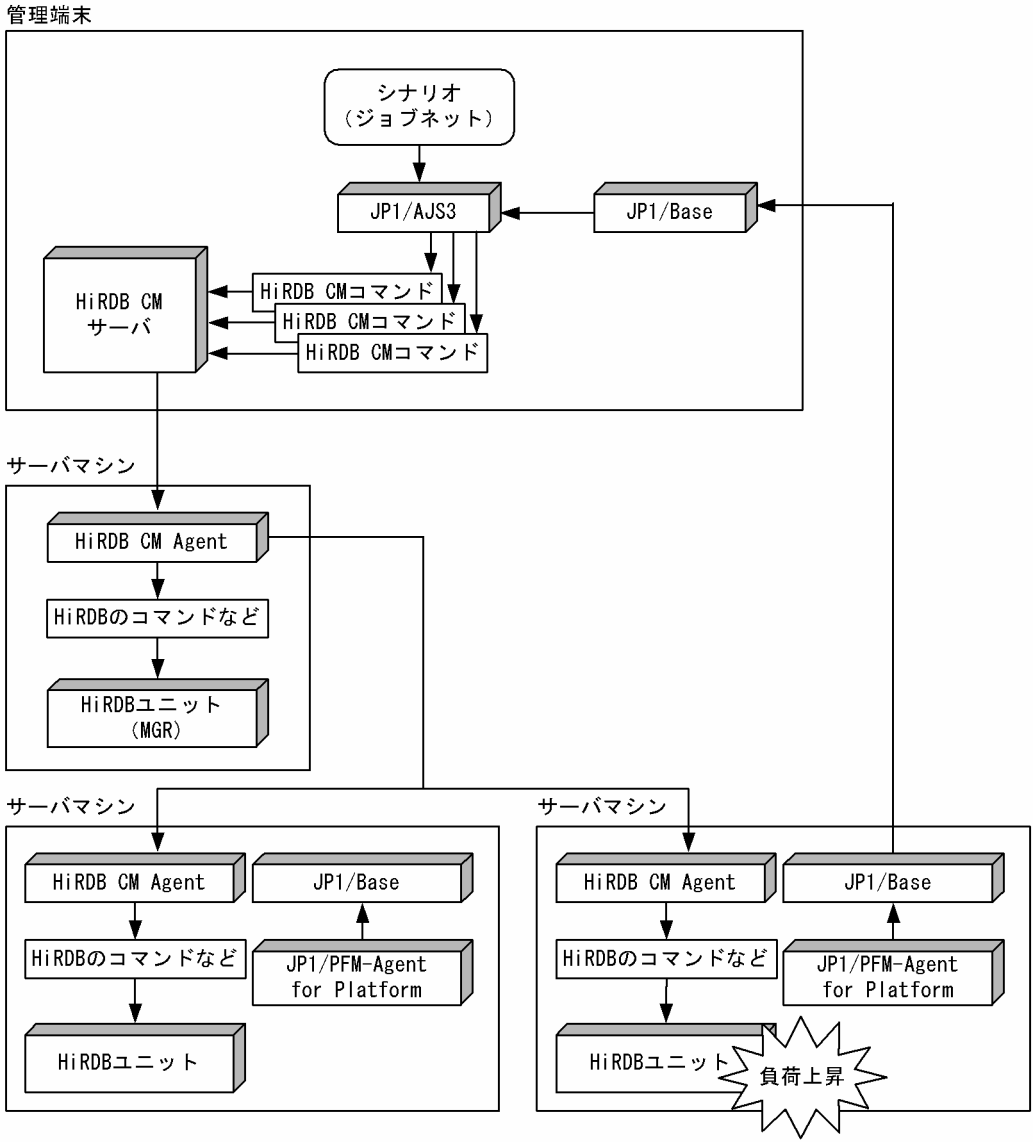
特徴

- バックエンドサーバの負荷均等化を動的に行えます。
- バックエンドサーバの負荷を半自動的に変更できます。
- バックエンドサーバの移動要否をオペレータが最終確認した後に，バックエンドサーバの移動をするため，柔軟かつ安全な運用ができます。

(3) JP1/PFM によるバックエンドサーバの負荷監視と自動シナリオ実行

JP1/PFM を使用してバックエンドサーバの負荷を監視します。負荷が設定値を超えた場合，自動的に JP1/AJS3 でシナリオを実行します。JP1/PFM によるバックエンドサーバの負荷監視と自動シナリオ実行を次の図に示します。

図 11–8 JP1/PFM によるバックエンドサーバの負荷監視と自動シナリオ実行



特徴

- バックエンドサーバの負荷均等化を動的に行えます。
- オペレータの操作を必要としません。バックエンドサーバの負荷を自動で変更できます。

11.8.4　バックエンドサーバの構成例

負荷均等化を行うときのバックエンドサーバの構成例について説明します。

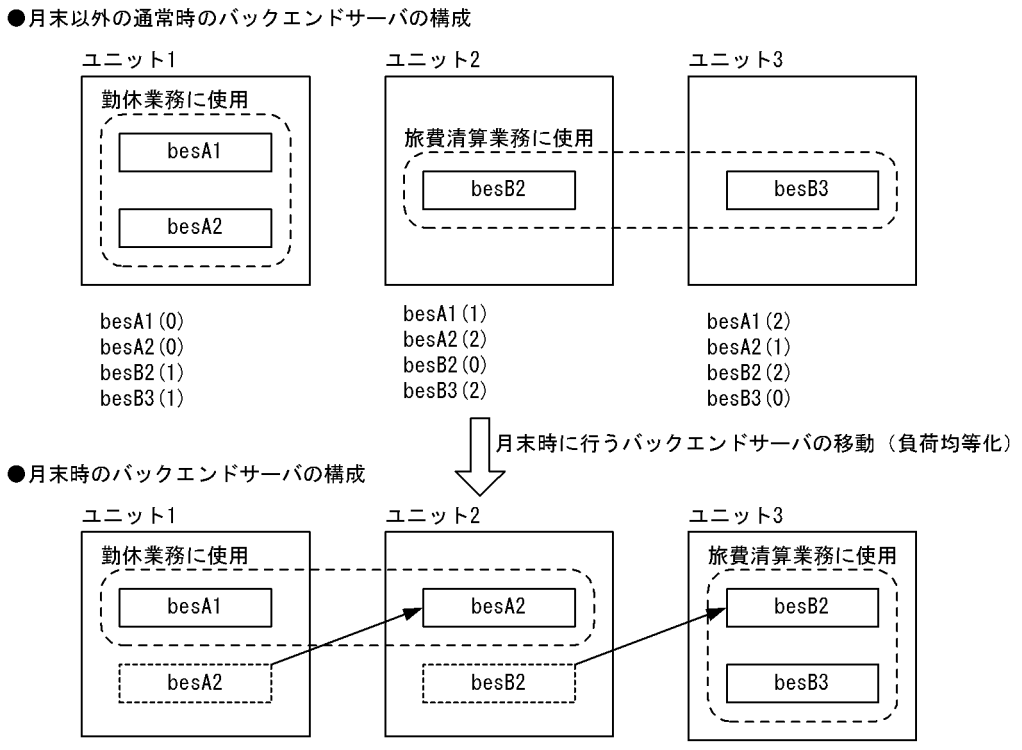
(1)　構成例 1（2 業務，3 ユニット，4 バックエンドサーバで構成する）

想定する業務

- 勤休業務と旅費清算業務の 2 業務が存在する
- 勤休業務と旅費清算業務にそれぞれバックエンドサーバを二つずつ割り当てて，業務間で排他的に利用する
- 通常時は旅費清算業務の性能が求められる
- 月末は勤休業務の負荷が高くなる
- 月末は旅費清算業務より勤休業務を優先する

負荷均等化を行うときのバックエンドサーバの構成例を次の図に示します。

図 11–9　負荷均等化を行うときのバックエンドサーバの構成例 1（2 業務，3 ユニット，4 バックエンドサーバの場合）



注

（ ）内の数字は各バックエンドサーバのユニット内の切り替え優先順位を示しています。MSFC の各バックエンドサーバに対応するグループのプロパティの優先所有者に，昇順になるようにノード名を指定し，優先順位を決めます。

なお，(0) は現用系を意味しています。この場合，各グループの優先所有者の先頭に該当するノードを指定します。

〔説明〕

- 勤休業務の負荷が高くなる月末には，バックエンドサーバを移動して負荷均等化を行います。
- バックエンドサーバ移動の基本シナリオを 2 個組み合わせてこのシナリオを実現します。
- 図中に示す優先順位を指定すると，バックエンドサーバの移動後に障害が発生した場合，バックエンドサーバの負荷アンバランスを避けられます。

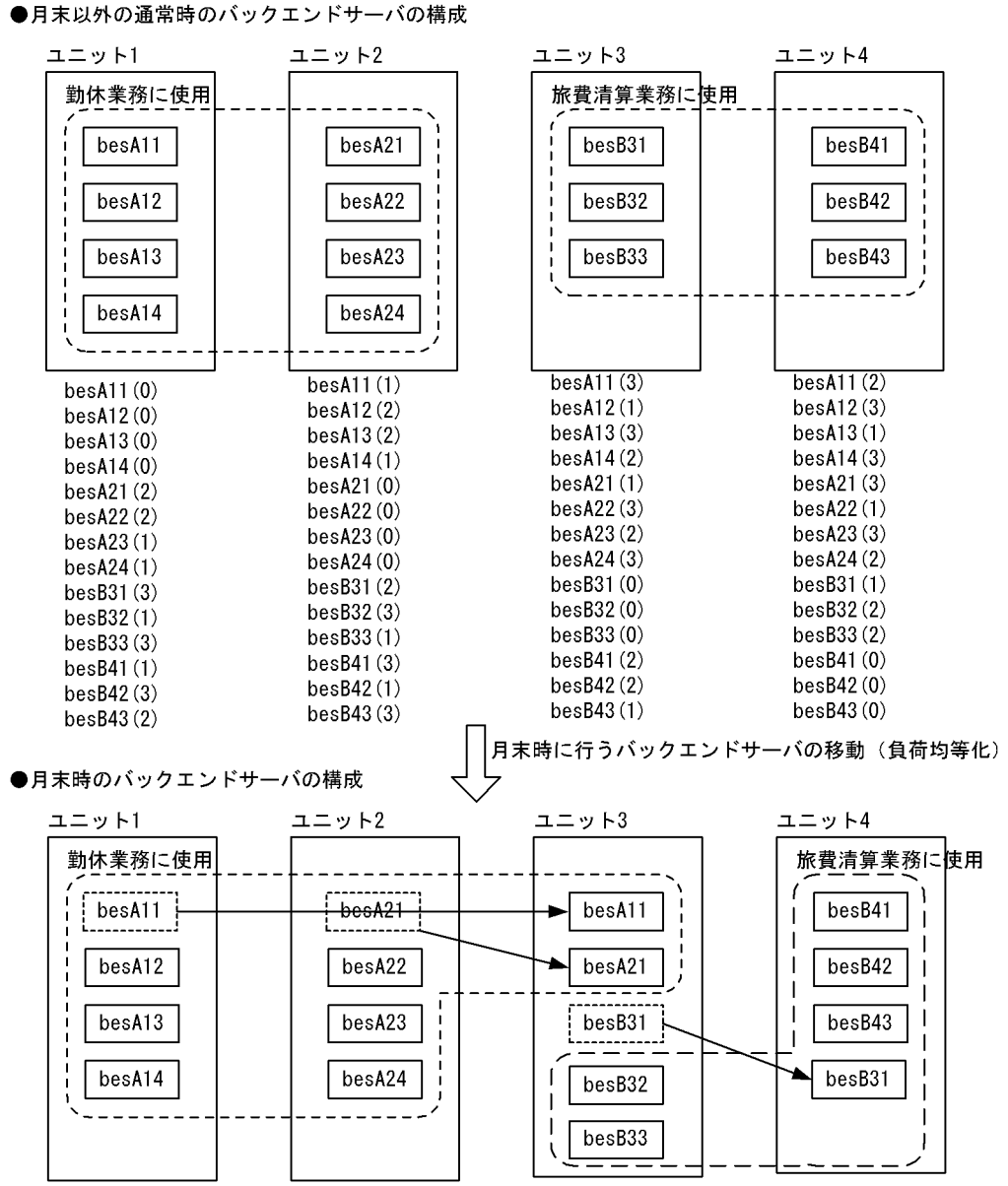
(2) 構成例 2（2 業務，4 ユニット，14 バックエンドサーバで構成する）

想定する業務

- 勤休業務と旅費清算業務の 2 業務が存在する
- 勤休業務にバックエンドサーバを八つ，旅費清算業務にバックエンドサーバを六つ割り当てて，業務間で排他的に利用する
- 通常時，勤休業務と旅費清算業務は同等の性能が求められる
- 月末は勤休業務の負荷が高くなる
- 月末も旅費清算業務の性能低下は極力避ける

負荷均等化を行うときのバックエンドサーバの構成例を次の図に示します。

図 11–10　負荷均等化を行うときのバックエンドサーバの構成例 2（2 業務，4 ユニット，14 バックエンドサーバの場合）



注

（　）内の数字は各バックエンドサーバのユニット内の切り替え優先順位を示しています。MSFC の各バックエンドサーバに対応するグループのプロパティの優先所有者に，昇順になるようにノード名を指定し，優先順位を決めます。

なお，(0) は現用系を意味しています。この場合，各グループの優先所有者の先頭に該当するノードを指定します。

〔説明〕

- 勤休業務の負荷が高くなる月末には，バックエンドサーバを移動して負荷均等化を行います。
- バックエンドサーバ移動の基本シナリオを 3 個組み合わせてこのシナリオを実現します。

11.8.5　HiRDB に関する準備

HiRDB に関する準備作業を次に示します。

(1)　影響分散スタンバイレス型系切り替え機能を適用する

影響分散スタンバイレス型系切り替え機能を適用してください。影響分散スタンバイレス型系切り替え機能については，「系切り替え機能の運用」を参照してください。

(2)　トランザクションキューイング機能の適用要否を選択する

トランザクションキューイング機能を適用するかどうかを選択してください。適用要否の目安を次に示します。トランザクションキューイング機能については，「トランザクションキューイング機能」を参照してください。

(a)　トランザクションキューイング機能を適用する場合の処理方式

トランザクションキューイング機能を適用する場合の処理方式を次に示します。

処理方式

1. バックエンドサーバの移動時，移動対象のバックエンドサーバに対する新規トランザクションの開始を保留します。
2. 移動対象のバックエンドサーバで処理中のトランザクションの終了を待ち合わせます。移動対象のバックエンドサーバで処理中のトランザクションがなくなった時点でバックエンドサーバを移動します。
3. 1 で保留中のトランザクションを開始します。

なお，トランザクションのキューイング待ち時間を pd_ha_trn_queuing_wait_time オペランドで指定できます。長時間実行するトランザクションがあるため，バックエンドサーバの終了に時間が掛かる場合，この時間を超えるとトランザクションのキューイングが解除されて，バックエンドサーバの移動を中止します。この場合，1 で保留中のトランザクションは再開されます。

(b)　トランザクションキューイング機能を適用しない場合の処理方式

新規トランザクションの開始保留，及び実行中トランザクションの終了待ち合わせをしないで，バックエンドサーバを移動します。

(c)　トランザクションキューイング機能のメリット及びデメリット

トランザクションキューイング機能のメリット及びデメリットを次の表に示します。

表 11–1　トランザクションキューイング機能のメリット及びデメリット

トランザクションキューイング機能の適用可否	メリット	デメリット
適用する場合	バックエンドサーバ移動時のトランザクションエラーを回避できます（バックエンドサーバの開始に時間が掛かる場合を除く）。	• バックエンドサーバの移動中にトランザクションが滞留します。 • バックエンドサーバの移動に掛かる時間が適用しない場合に比べて長くなります。 • 長時間実行するトランザクションがある場合，バックエンドサーバの移動を中止します。
適用しない場合	• バックエンドサーバの移動中にトランザクションが滞留しません。 • 長時間実行するトランザクションがある場合でも，バックエンドサーバを移動できます。 • バックエンドサーバの移動に掛かる時間が適用する場合に比べて短くなります。	• バックエンドサーバの移動時に実行中のトランザクションがエラーになります。 • バックエンドサーバの移動時に開始したトランザクションがエラーになります。

参考

トランザクションキューイング機能は，障害発生時の系切り替えにも適用されるため，系切り替え機能の運用も考慮して検討してください。

(d)　トランザクションキューイング機能の設定方法

トランザクションキューイング機能を適用する場合は，pd_ha_transaction オペランドに queuing を指定してください。

また，次に示す時間の合計をトランザクションのキューイング待ち時間として pd_ha_trn_queuing_wait_time オペランドに指定してください。

- バックエンドサーバの移動時に発生するトランザクションの最大滞留時間
- バックエンドサーバの移動シナリオのサーバ強制終了（pdstop -s -f コマンド入力時：KFPS01843-I メッセージ出力とほぼ同時と考えてよい）から，サーバ開始完了（KFPS01813-I メッセージ出力）までの時間

トランザクションキューイング機能を適用しない場合は，pd_ha_transaction オペランドに error を指定するか，又はこのオペランドを省略してください。

11.8.6　ユーザが行うバックエンドサーバの負荷均等化

ユーザが，コマンドを実行してバックエンドサーバの負荷を均等化できます。

(1)　操作手順

Hitachi HA Toolkit Extension 使用時の操作手順を次に示します。なお，複数のバックエンドサーバを移動する場合は，「バックエンドサーバの移動の自動化」を参照してください。

1.移動するサーバの状態を確認する

システムマネジャがあるユニットで，pdls -d ha -s サーバ名 -a コマンドを実行し，その表示内容（標準出力）から次の項目を確認してください。

- 移動対象となるバックエンドサーバが，移動元ユニットで稼働状態（ONL）である
- 移動対象となるバックエンドサーバが，移動先ユニットで待機状態（SBY）又は停止状態（STP）である

2.移動するバックエンドサーバのトランザクションキューイング機能を開始する

バックエンドサーバの移動元ユニットで，pdtrnqing -s サーバ名コマンドを実行し，トランザクションキューイング機能を開始してください。pd_ha_transaction オペランドに queuing を指定していない場合は，この操作を行う必要はありません。

コマンドがリターンコード 0 で正常終了，又はリターンコード 4 で警告終了したことを確認してください。リターンコードが 0 の場合は，トランザクションキューイング機能が正常に開始されたことを意味します。リターンコードが 4 の場合は，pd_ha_transaction オペランドの指定値が error であることを意味します。リターンコードが 8 及び 12 の場合は，障害が発生しているため手順を終了してください。

3.クラスタソフトウェアの系切り替えコマンドでバックエンドサーバを移動する

クラスタソフトウェアの系切り替えコマンドを使用して，移動先ユニットに系を切り替えてください。

4.バックエンドサーバの状態を確認する

システムマネジャがあるユニットで，pdls -d ha -s サーバ名 -a コマンドを実行し，その表示内容（標準出力）及びリターンコードから次の項目を確認してください。

- バックエンドサーバが，移動先ユニットで稼働状態（ONL）である

5.移動するバックエンドサーバのトランザクションキューイング機能を解除する

バックエンドサーバの移動先ユニットで，pdtrnqing -d -s サーバ名コマンドを実行し，トランザクションキューイング機能を解除してください。pd_ha_transaction オペランドに queuing を指定していない場合は，この操作をする必要はありません。

コマンドのリターンコードが 0 であることを確認してください。リターンコードが 0 でない場合，障害が発生しているため手順を終了してください。

(2)　障害発生時の対処

バックエンドサーバの移動中にエラーが発生した場合，イベントログ，及び標準出力・標準エラー出力のメッセージを参照し，原因を取り除いてください。

また，トランザクションキューイング機能開始以降に障害が発生した場合は，トランザクションキューイング機能を中断するため，システムマネジャがあるユニットで次のコマンドを実行してください。

```
pdtrnqing -d -f -s サーバ名
```

上記コマンドを実行すると，トランザクションキューイング機能は解除されます。ただし，障害によって不当にトランザクションキューイング機能が動作することを防ぐための対処であるため，開始されていないバックエンドサーバに対して処理要求が発生し，SQL エラーとなります。

(3) 注意事項

(a) バックエンドサーバの移動の自動化

トランザクションキューイング機能を使用して，複数のバックエンドサーバを移動する場合は，次の手順で自動化してください。

1. トランザクションキューイング機能の開始
2. バックエンドサーバの移動
3. トランザクションキューイング機能の解除

ユーザが手作業で行うと，バックエンドサーバの移動処理に時間が掛かり，トランザクションキューイング機能のタイムアウトが発生し，トランザクションがエラーとなることがあります。

(b) 複数のバックエンドサーバを移動する場合

トランザクションキューイング機能を使用して，複数のバックエンドサーバを移動する場合，1 サーバごとに「操作手順」を実行してください。複数同時には実行しないでください。複数のバックエンドサーバを同時に移動させると，それらのバックエンドサーバにアクセスするトランザクションがキューイングされ，トランザクションが終了できなくなることがあります。この場合，pdtrnqing コマンドがエラーとなることがあります。

12 システム間で資源を移行する方法

この章では，ほかの HiRDB システムへの表，及びストアドプロシジャの移行方法について説明します。

12.1　表をほかの HiRDB システムに移行する方法

ディクショナリ搬出入ユティリティ，データベース再編成ユティリティ，及びデータベース作成ユティリティを使用して，現在使用している表をほかの HiRDB システムに移行できます。この機能を使用すると，次に示すことができます。

- テスト系で使用していたデータベースを本番系に移行できます。
- ある HiRDB システムで作成したデータベースを，ほかの HiRDB システムに移行できます。
- システムを再構築するときに既存のデータベース情報を保存でき，システムを再構築した後にそのデータベース情報を回復できます。

次の表に示すように，表をほかの HiRDB に移行する方法は二つあります。

表 12–1　表の移行方法

条件		移行方法
移行元システムと移行先システムの RD エリア構成	プラグインが提供する抽象データ型の列の有無	
同じ	なし	表定義情報と表データを同時に移行する。
	あり	移行先で表・インデクス定義をしてデータだけ移行する。
異なる	なし	表定義情報と表データを別々に移行する。
	あり	移行先で表・インデクス定義をしてデータだけ移行する。

それぞれの移行方法について，説明します。

表定義情報と表データを同時に移行する方法

実行者　HiRDB 管理者，及び表の所有者

データベース再編成ユティリティ及びデータベース作成ユティリティを使用して，表定義情報と表データを同時に移行する方法です。移行できる情報や詳細については，「表定義情報と表データを同時に移行する方法」を参照してください。

表定義情報と表データを別々に移行する方法

実行者　HiRDB 管理者

ディクショナリ搬出入ユティリティ，データベース再編成ユティリティ，及びデータベース作成ユティリティを使用して，表定義情報と表データを別々に移行する方法です。

移行できる情報や詳細については，「表定義情報と表データを別々に移行する方法」を参照してください。

12.1.1　表をほかの HiRDB システムに移行する前に

(1)　制限事項

- ディクショナリ搬出入ユティリティでは，抽象データ型を定義している表の定義情報をほかの HiRDB システムに移行できません。移行できる情報や詳細については，「表定義情報と表データを別々に移行する方法」を参照してください。
- 表定義情報と表データを同時に移行する方法では，スキーマ単位で移行できません（表単位の移行だけができます）。表定義情報と表データを別々に移行する方法では，スキーマ単位又は表単位に移行できます。
- 移行元システムと移行先システムの文字コードが異なる場合で，文字コード変換が必要な列を持つ表を移行するときは，まず文字コード変換をしてから，移行先へデータロードしてください。

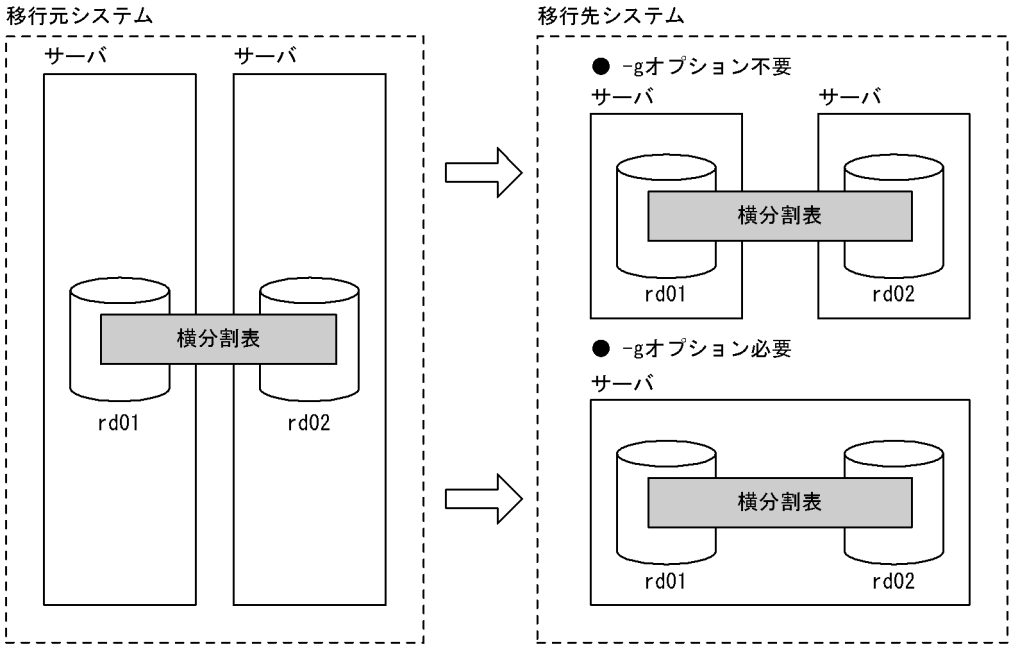
  文字コード変換が不要な列（数値型の列など）だけの表を移行するときは，移行元システムでデータベース再編成ユティリティを使用してデータベース作成ユティリティ用の入力データファイルを作成し，移行先システムでデータベース作成ユティリティを使用してデータロードしてください。

  データベース再編成ユティリティは，移行元システムのアンロードデータファイルの文字コードと，移行先システムの文字コードが異なる時点でエラーとなりますが，この方法で移行すると，エラーになりません。
- 改竄防止表はほかの HiRDB システムに移行できません。また，移行先システムで表データをリロードできません。
- データベース複写ユティリティとデータベース回復ユティリティを使用したデータの移行はできません。移行元システムで取得したバックアップを移行先システムでリストア（回復）しないでください。

(2)　RD エリアに関する注意事項（重要）

1. 表定義情報を移行する場合，移行先システムに移行元システムと同じ名称の RD エリアを作成しておく必要があります。
2. 表定義情報と表データを同時に移行する方法で，非分割表を移行する場合，移行元システムと移行先システムで RD エリア名が異なるとき，移行先システムでデータベース作成ユティリティ実行時に-w rdaomit オプションを指定すれば，移行ができます。ただし，表に BLOB 列が定義されている場合は，表定義エラーとなります。
3. 表定義情報と表データを別々に移行する方法で，移行対象の表が格納されている RD エリア名と RD エリアを格納するサーバ構成が，移行元システムと移行先システムで異なる場合，データベース再編成ユティリティ実行時に-g オプションを指定する必要があります。-g オプションを指定する必要があるシステム構成例を次の図に示します。

図 12–1　-g オプションを指定する必要があるシステム構成例



(3)　表定義（データ型及び列の定義長）を変更する場合の注意事項

データ型又は列の定義長を変更できるのは，データベース再編成ユティリティで移行元システムの表データを DAT 形式又はバイナリ形式でアンロードしたファイルを，移行先システムでデータロードするときです。DAT 形式及びバイナリ形式の入力データファイルの制限事項については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(a)　表定義を変更して表データを移行できない場合

表定義を変更して表データを移行できないのは，次の場合です。

- 移行元システムと移行先システムのデータ型の組み合わせによって，移行に使用する入力データファイルの形式（DAT 形式かバイナリ形式か）が異なります。このため，一つの表内に DAT 形式だけで移行できるデータ型とバイナリ形式だけで移行できるデータ型が混在していると，移行できません。この場合は UAP などで移行用のデータを作成する必要があります。
- DAT 形式のファイルで移行する場合，アンロードするときに文字列に変換するため，文字列に変換できないデータが含まれている表は移行できません。

  移行先システムで表データをデータロードする場合，移行元システムのデータが移行先システムのデータ型に対応した形式でないと，エラーになったり，データが切り捨てられるおそれがあります。このため，移行元システムのデータが，移行先システムのデータ型に対応している必要があります。詳細は，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(b)　移行できるデータ型の組み合わせと入力データファイルの形式

移行元システムと移行先システムで，移行できるデータ型の組み合わせと，その場合に作成する入力データファイルの形式を次の表に示します。

表 12–2　移行できるデータ型の組み合わせと入力データファイルの形式

移行できるデータ型の組み合わせ		入力データファイルの形式	
移行元システム	移行先の列	DAT 形式	バイナリ形式
数データ	数データ	△	△
	固定長文字データ	○	×
	可変長文字データ		
	長大データ		
	バイナリデータ		
固定長文字データ	数データ	△	×
	固定長文字データ	○	○
	可変長文字データ		×
	日付データ	△	
	時刻データ		
	日間隔データ		
	時間隔データ		
	時刻印データ		
	長大データ	○	
	バイナリデータ		
可変長文字データ	数データ	△	×
	固定長文字データ	○	
	可変長文字データ		○
	日付データ	△	×
	時刻データ		
	日間隔データ		
	時間隔データ		
	時刻印データ		
	長大データ	○	
	バイナリデータ		
日付データ	固定長文字データ	○	×
	可変長文字データ		
	日付データ		○
	長大データ		×

移行できるデータ型の組み合わせ		入力データファイルの形式	
移行元システム	移行先の列	DAT 形式	バイナリ形式
	バイナリデータ		
時刻データ	固定長文字データ	○	×
	可変長文字データ		
	日付データ	×	
	時刻データ	○	○
	長大データ		×
	バイナリデータ		
日間隔データ	固定長文字データ	○	×
	可変長文字データ		
	日間隔データ		○
	長大データ		×
	バイナリデータ		
時間隔データ	固定長文字データ	○	×
	可変長文字データ		
	時間隔データ		○
	長大データ		×
	バイナリデータ		
時刻印データ	固定長文字データ	○	×
	可変長文字データ		
	時刻印データ		○
	長大データ		×
	バイナリデータ		
長大データ	長大データ	×	○
	バイナリデータ		
バイナリデータ	数データ	△	×
	固定長文字データ	○	
	可変長文字データ		
	日付データ	△	
	時刻データ		
	日間隔データ		

移行できるデータ型の組み合わせ		入力データファイルの形式	
移行元システム	移行先の列	DAT 形式	バイナリ形式
	時間隔データ		
	時刻印データ		
	長大データ		
	バイナリデータ	○	○

（凡例）

○：移行できます。

△：データ値が移行先システムのデータ型に対応した値であれば，移行できます。

×：移行できません。

(c)　列の定義長を変更する場合の入力データファイルの形式

移行先システムで定義長をどのように変更するかと，その場合に作成する入力データファイルの形式を次の表に示します。

移行元システム		移行先システム	入力データファイル	
定義長を変更する列のデータ型		変更後の定義長 （変更前と比べて）	DAT 形式	バイナリ形式
数データ	DECIMAL 型	長い	△	×
		短い	△	×
	DECIMAL 型以外	–※1	–	–
固定長文字データ		長い	○※2	○※2，※3
		短い	△	×
可変長文字データ		長い	○	○
		短い	△	×
時刻印データ		長い	△	×
		短い	△	×
長大データ		長い	×	○
		短い	×	×
バイナリデータ		長い	△	○
		短い	△	×

（凡例）

○：移行できます。

△：データ値が移行先システムのデータ型に対応した値であれば，移行できます。

×：移行できません。

−：該当しません。

注※1

DECIMAL 型以外の数データは，定義長が固定のため，対象外です。

注※2

左詰めで入力され，余りは空白で埋められます。

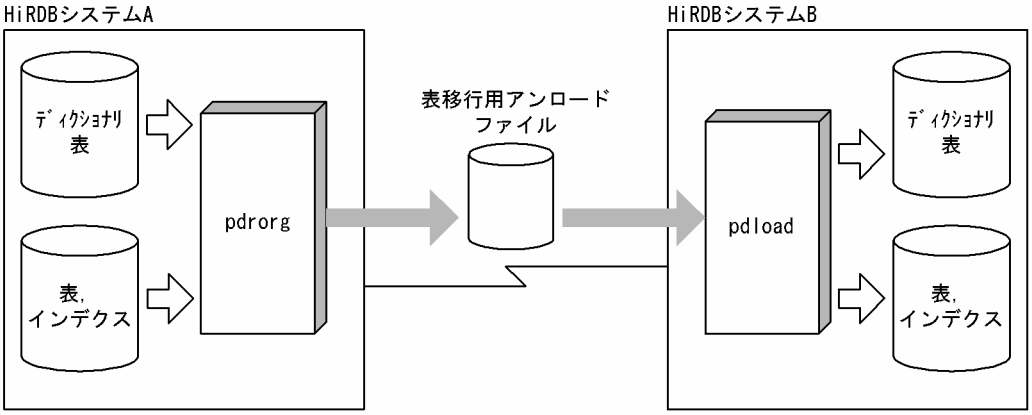
注※3

HiRDB のバージョンが 08-04 より前の場合は，移行できません。

12.1.2　表定義情報と表データを同時に移行する方法

データベース再編成ユティリティ（**pdrorg** コマンド）及びデータベース作成ユティリティ（**pdload** コマンド）で，現在使用している表の表定義情報と表データをほかの HiRDB システムに移行します。データベース再編成ユティリティ及びデータベース作成ユティリティを使用した表定義情報及び表データの移行を次の図に示します。

図 12–2　データベース再編成ユティリティ及びデータベース作成ユティリティを使用した表定義情報及び表データの移行



移行元システムで，データベース再編成ユティリティを使用してディクショナリ表から逆生成した定義情報と表データを**表移行用アンロードファイル**に出力します。このことを，**エクスポート**といいます。移行先システムで，データベース作成ユティリティを使用して，表移行用アンロードファイルを入力として表を定義し，表データを格納することを**インポート**といいます。エンディアンが異なるシステムに表を移行する場合，インポート時にエンディアンを変換します。

(1)　移行する情報

- 表定義情報

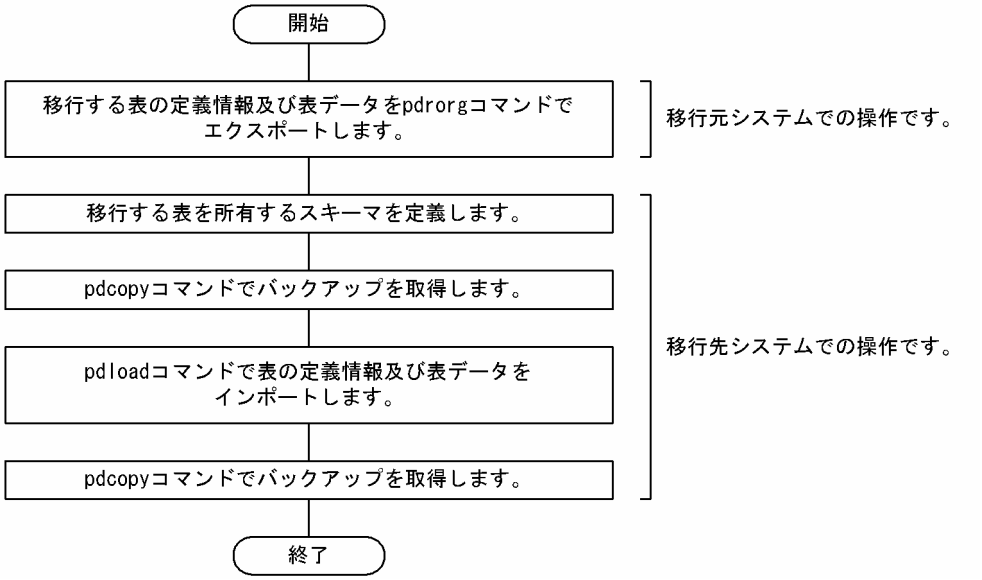
- 実表の定義（ユーザが定義した抽象データ型を含まない表に限ります）
- インデクスの定義（部分構造インデクスは除きます）

- 表データ

(2)　移行手順

表定義情報と表データを同時にほかの HiRDB システムに移行する手順の概略を次の図に示します。

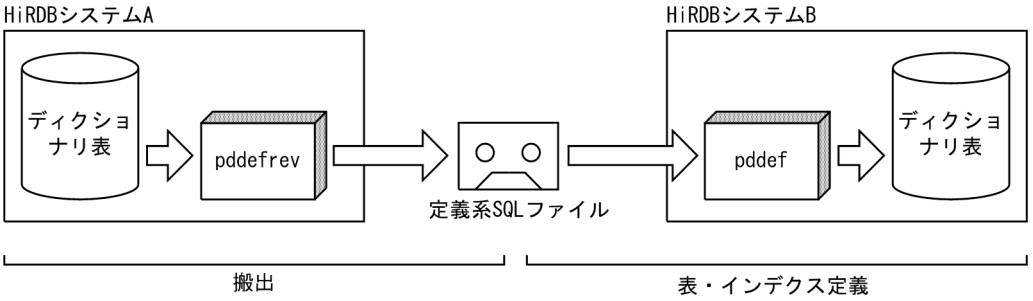
図 12–3　表定義情報と表データを同時にほかの HiRDB システムに移行する手順



12.1.3　表定義情報と表データを別々に移行する方法

まず，pddefrev（定義系 SQL の生成）を使用して定義系 SQL を生成します。移行先でデータベース定義ユティリティ（pddef）を使用して，生成した定義系 SQL によって表・インデクスを定義します。pddefrev（定義系 SQL の生成）を使用した表定義情報の移行を次の図に示します。

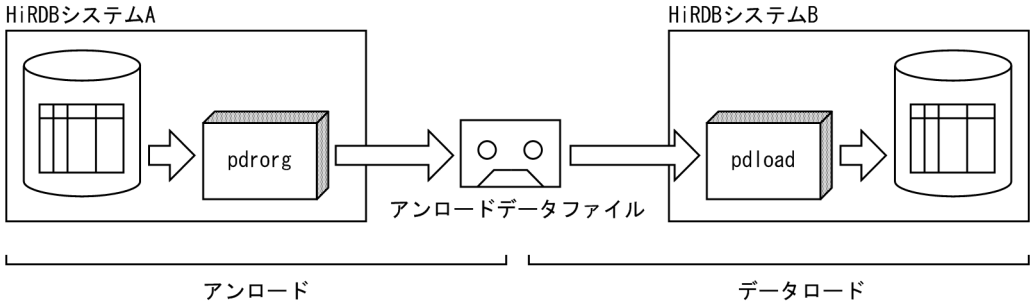
図 12–4　pddefrev（定義系 SQL の生成）を使用した表定義情報の移行



注　HiRDBシステムAの表定義情報はそのまま残ります。

次に，データベース再編成ユティリティ及びデータベース作成ユティリティで，現在使用している表データをほかの HiRDB システムに移行します。データベース再編成ユティリティ及びデータベース作成ユティリティを使用した表データの移行を次の図に示します。

図 12–5　データベース再編成ユティリティ及びデータベース作成ユティリティを使用した表データの移行



注　HiRDBシステムAの表データはそのまま残ります。

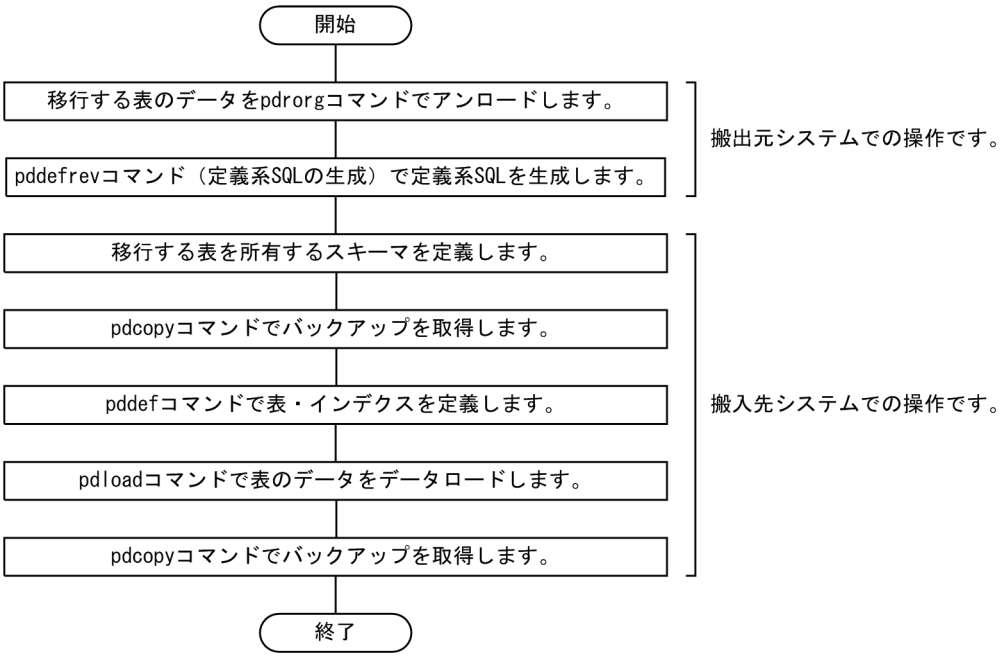
(1)　移行する情報

- 表定義情報
  - 実表の定義（抽象データ型を含まない表に限ります）
  - ビュー表の定義
  - インデクスの定義
  - コメント情報
- 表データ

(2)　移行手順

表定義情報と表データを別々にほかの HiRDB システムに移行する手順の概略を次の図に示します。

図 12–6　表定義情報と表データを別々にほかの HiRDB システムに移行する手順

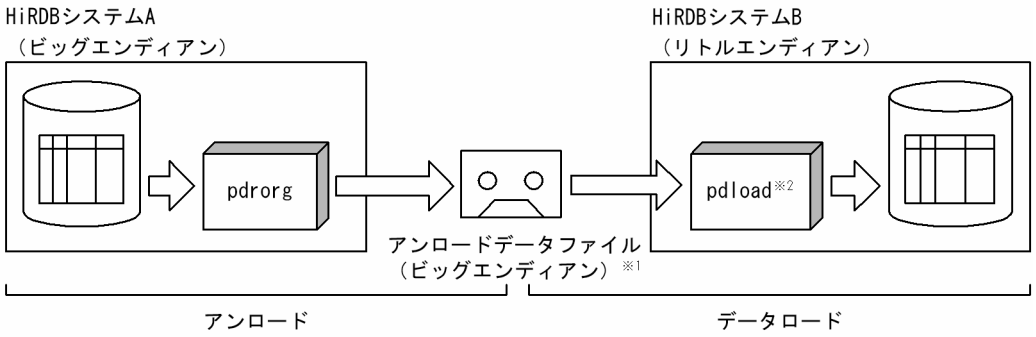


(3)　エンディアンが異なるシステムに表データを移行する場合

移行先システムでデータロードするときにエンディアン変換を行います。

ビッグエンディアンのシステムからリトルエンディアンのシステムに，バイナリデータを含む表を移行した場合の例を次の図に示します。

図 12–7　エンディアンが異なるシステム間の表データの移行



注1　HiRDBシステムAの表データはそのまま残ります。
注2　次に示す場合はpdrorgコマンド指定時に-jオプションを指定する必要があります。
　　・移行対象の表にLOB列が定義されている場合
　　・移行対象の表にLOB属性の抽象データ型が定義されている場合

注※1　移行対象の表にバイナリデータを含まない場合，アンロードファイルはDAT形式となります。
　　DAT形式はテキストデータのため，エンディアンの違いを意識する必要はありません。
注※2　アンロードデータファイルがバイナリ形式の場合だけ，エンディアン変換を行います。

注意事項

移行対象の表データに含まれるバイナリデータ（LOB 型，BLOB 型又は BINARY 型）の有無に応じて，移行元のデータを抽出する pdrorg コマンド，及び移行先でデータをロードする pdload コマンドのオプションを次のように指定してください。

表データのバイナリデータの有無	pdrorg コマンドのオプション指定	pdload コマンド	
		オプション指定	制御文ファイルでの指定
なし	-k unld -W dat	－	－
あり	-k unld -W bin	-W -b	option 文の srcendian オペランドに，移行元システムのエンディアン形式を指定します。 big：ビッグエンディアン little：リトルエンディアン

（凡例）

－：指定する必要はありません。

コマンドの実行例については，「例題４（エンディアンが異なるシステムに，バイナリデータを含む表を移行する場合）」を参照してください。pdrorg コマンド及び pdload コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(a)　移行元の表データに BLOB 型のデータが含まれる場合

移行元のデータを抽出する pdrorg コマンド，及び移行先でデータをロードする pdload コマンドのオプションを，移行元の HiRDB のバージョンに応じて次のように指定してください。

HiRDB のバージョン	BLOB 列の有無	pdrorg コマンド		pdload コマンドのオプション指定
		オプション指定	アンロードファイルの種類	
02-00～02-04	－	-k unld	●	-U
03-00～04-02	なし	-k unld -W bin	○	-W -b
	あり	×	×	×
04-03～05-04	なし	-k unld -W bin	○	-W -b
	あり	-k unld -j	●	-U
05-05 以降	なし	-k unld -W bin	○	-W -b
	あり	-k unld -W bin -j	○	-W -b

（凡例）

－：該当しません。

×：移行できません。

●：pdrorg 用アンロードファイルが作成されます。

○：pdload 用アンロードファイルが作成されます。

(b)　移行元の表データに抽象データ型が含まれる場合

移行元のデータを抽出する pdrorg コマンドのオプションを，移行元の HiRDB のバージョンに応じて次のように指定してください。

移行先でデータをロードするときは，移行元で作成されたアンロードファイルの種類に応じて，pdrorg 又は pdload コマンドを使用してください。

HiRDB のバージョン	抽象データ型の種別	pdrorg コマンド	
		オプション指定	アンロードファイルの種類
05-00～05-03	BLOB 属性	×	×
	BLOB 属性以外	×	×
05-04	BLOB 属性	×	×
	BLOB 属性以外	-k unld	●
05-05～05-06	BLOB 属性	-k unld -j	●
	BLOB 属性以外	-k unld	●
06-00 以降	BLOB 属性	-k unld -W bin -j	○
	BLOB 属性以外	-k unld -W bin	○

（凡例）

×：移行できません。

●：pdrorg 用アンロードファイルが作成されます。
移行先では pdrorg コマンドでデータをロードしてください。

○：pdload 用アンロードファイルが作成されます。
移行先では pdload コマンドでデータをロードしてください。

(c)　移行元の表データに繰返し列又は BINARY 列が含まれる場合

移行元のデータを抽出する pdrorg コマンド，及び移行先でデータをロードする pdload コマンドのオプションを，次のように指定してください。

pdrorg コマンド		pdload コマンドのオプション指定
オプション指定	アンロードファイルの種類	
-k unld -W bin	○	-W -b

（凡例）

○：pdload 用アンロードファイルが作成されます。

注

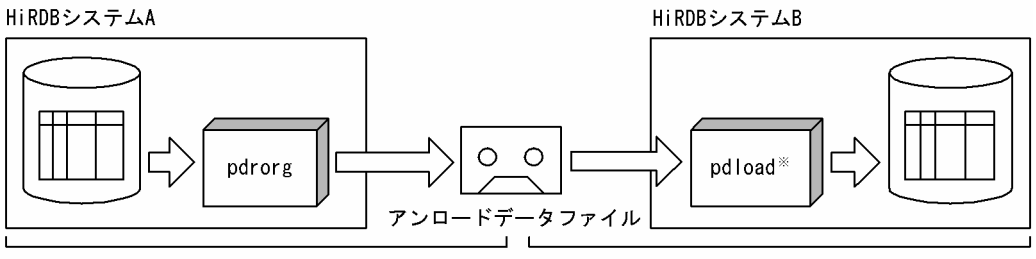
繰返し列はバージョン 05-03 以降，BINARY 列はバージョン 07-00 以降の HiRDB で使用できるデータ型です。

(4)　固定長文字データの定義長を拡張して表データを移行する場合

移行先システムの表定義（データ型又は列の定義長）が，移行元システムと異なっていても，移行先システムでデータロードできます。

固定長文字データの列の定義長を拡張して，移行先システムに表を移行する場合の例を次の図に示します。

図 12–8　固定長文字データの列の定義長を拡張する表データの移行



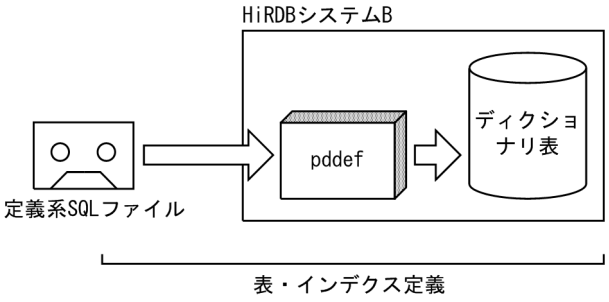
注1　HiRDBシステムAの表データはそのまま残ります。
注2　次に示す場合はpdrorgコマンド指定時に-jオプションを指定する必要があります。
・移行対象の表にLOB列が定義されている場合
・移行対象の表にLOB属性の抽象データ型が定義されている場合
注3　表データをデータロードする前に定義長を変更した表をHiRDBシステムBで定義しておく必要があります。

注※　固定長文字列データ型の列定義長が拡張されていた場合，表データは左詰めで入力され，定義長を拡張した部分は空白で埋められます。

12.1.4　移行先で表・インデクス定義を行いデータだけ移行する方法

移行先でデータベース定義ユティリティ（pddef）を使用して，定義系 SQL によって表・インデクスを定義します。データベース定義ユティリティを使用した表・インデクス定義を次の図に示します。

図 12–9　データベース定義ユティリティを使用した表・インデクス定義



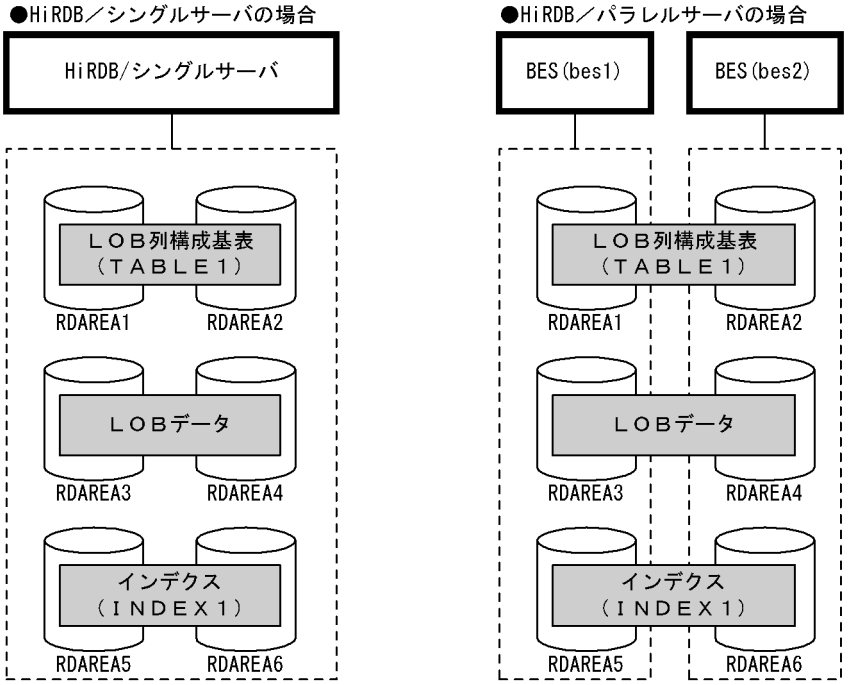
次に，データベース再編成ユティリティ及びデータベース作成ユティリティで，現在使用している表データをほかの HiRDB システムに移行します。移行方法については，「表定義情報と表データを別々に移行する方法」を参照してください。

12.1.5　例題１（表定義情報と表データを同時に移行する場合）

HiRDB システムＡの表（TABLE1）を HiRDB システムＢに移行します。

- TABLE1 には LOB 列が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。

なお，移行先システムには，同じ名称の RD エリア（RDAREA1～RDAREA6）が作成されているとします。



(1)　pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥file¥export_file
```

〔説明〕

表移行用アンロードファイルの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥file¥export_file
```

〔説明〕

表移行用アンロードファイル（bes1 用）の名称を指定します。-g オプションが仮定されるため，表移行用アンロードファイルを 1 か所に作成します。

(3) pdrorg コマンドで TABLE1 をエクスポートします

```
pdrorg -k unld -W bin -w -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

-k：エクスポート（アンロード）をするため unld を指定します。

-W：表移行用アンロードファイルを作成するため bin -w を指定します。

-t：エクスポートする表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4) pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(5) 表移行用アンロードファイルを CMT などの媒体に格納します

(3)で作成した表移行用アンロードファイルを CMT などの媒体に格納します。

移行元システム（HiRDB システムＡ）での作業はこれで終わりです。

(6) 移行先システムに表移行用アンロードファイルを格納します

これ以降は移行先システム（HiRDB システムＢ）での作業になります。

CMT などの媒体に格納した表移行用アンロードファイルを移行先システムに格納します。

(7) インポートを実行するユーザのスキーマを pddef コマンドで定義します

```
pddef
   CREATE SCHEMA AUTHORIZATION USR01;
```

(8) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(9)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

データベース移行中の障害発生に備えてバックアップを取得します。バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 表をインポートするユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(10)　pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
source C:¥file¥export_file                    1
idxwork C:¥idxwork                            2
sort C:¥sortwork                              3
```

〔説明〕

1. 表移行用アンロードファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
source bes1:C:¥file¥export_file          1
idxwork bes1 C:¥idxwork                  2
sort bes1 C:¥sortwork                    3
idxwork bes2 C:¥idxwork                  4
sort bes2 C:¥sortwork                    5
```

〔説明〕

1. 表移行用アンロードファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(11)　pdload コマンドで TABLE1 をインポートします

```
pdload -b -W -w all TABLE1 C:¥pdload¥load01
```

〔説明〕

インデクス（INDEX1）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-b：表移行用アンロードファイルを入力するため，-W -w を指定します。また，表定義情報と表データを同時にインポートするため，all を指定します。

C:¥pdload¥load01：(11)で作成した pdload コマンドの制御文ファイル名を指定します。

このとき，移行元システムと移行先システムでエンディアンが異なる場合，インポート時にエンディアン変換をします。なお，インポート時に障害が発生した場合の対処については，「インポート時に障害が発生した場合の対処方法」を参照してください。

(12)　システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(13)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 表をインポートするユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

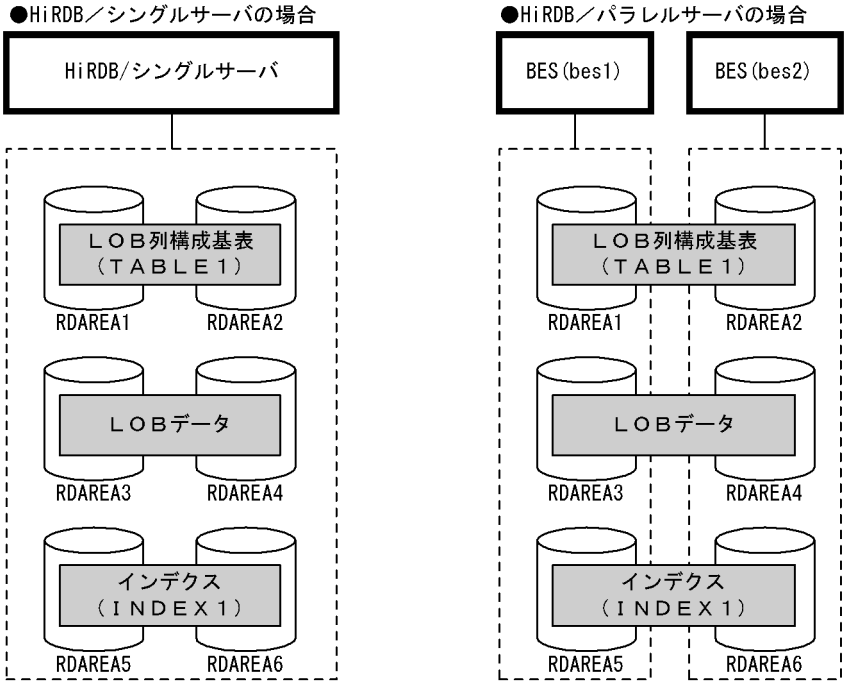
バックアップの取得については，「バックアップの取得方法」を参照してください。

12.1.6　例題２（表単位に定義情報とデータを別々に移行する場合）

HiRDB システムＡの表（TABLE1）を HiRDB システムＢに移行します。

- TABLE1 には LOB 列が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。

なお，データの格納先や表定義変更がある場合は，pddefrev で生成した定義系 SQL を変更します。



(1)　pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイルの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイル（bes1 用）の名称を指定します。

(3)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -W bin -g -t TABLE1 -j C:¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-g：HiRDB/パラレルサーバで，TABLE1 をサーバ間横分割している場合に指定してください。アンロードデータファイルが一つになります。

-t：アンロードする表の名称を指定します。

-j：LOB 列のデータも同時にアンロードします。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4) pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(5) pddefrev コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pddefrev¥exp01）の内容を次に示します。

```
-t USR01.TABLE1
```

〔説明〕

USR01：TABLE1 を所有するスキーマの名称です。

TABLE1：搬出する表の名称です。

(6) pddefrev コマンドで TABLE1 の表定義情報を搬出します

```
pddefrev -o C:¥pddefrev¥sql1 -f C:¥pddefrev¥exp01
```

〔説明〕

-o：定義系 SQL を出力するファイルの名称を指定します。

-f：(5)で作成した pddefrev コマンドの制御文ファイル名を指定します。

(7) アンロードデータファイル及び搬出ファイルを CMT などの媒体に格納します

(3)で作成したアンロードデータファイル，及び(6)で作成した搬出ファイルを CMT などの媒体に格納します。

移行元システム（HiRDB システム A）での作業はこれで終わりです。

(8)　移行先システムにアンロードデータファイル及び搬出ファイルを格納します

これ以降は移行先システム（HiRDB システムＢ）での作業になります。

CMT などの媒体に格納したアンロードデータファイル及び搬出ファイルを移行先システムに格納します。

(9)　TABLE1 を所有するユーザのスキーマを pddef コマンドで定義します

```
pddef
   CREATE SCHEMA AUTHORIZATION USR01;
```

(10)　システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(11)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

データベース移行中の障害発生に備えてバックアップを取得します。バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(12)　pddef コマンドで TABLE1 の表定義をします

```
pddef C:¥pddefrev¥sql1
```

〔説明〕

定義系 SQL のファイルの名称を指定します。

(13)　pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(14)　pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
source C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
source bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
idxwork bes2 C:¥pdrorg¥idxwork              4
sort bes2 C:¥sortwork                       5
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。

5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(15) pdload コマンドで TABLE1 のデータをデータロードします

```
pdload -b -W TABLE1 C:¥pdload¥load01
```

〔説明〕

インデクス（INDEX1）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE1：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(16) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(17) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

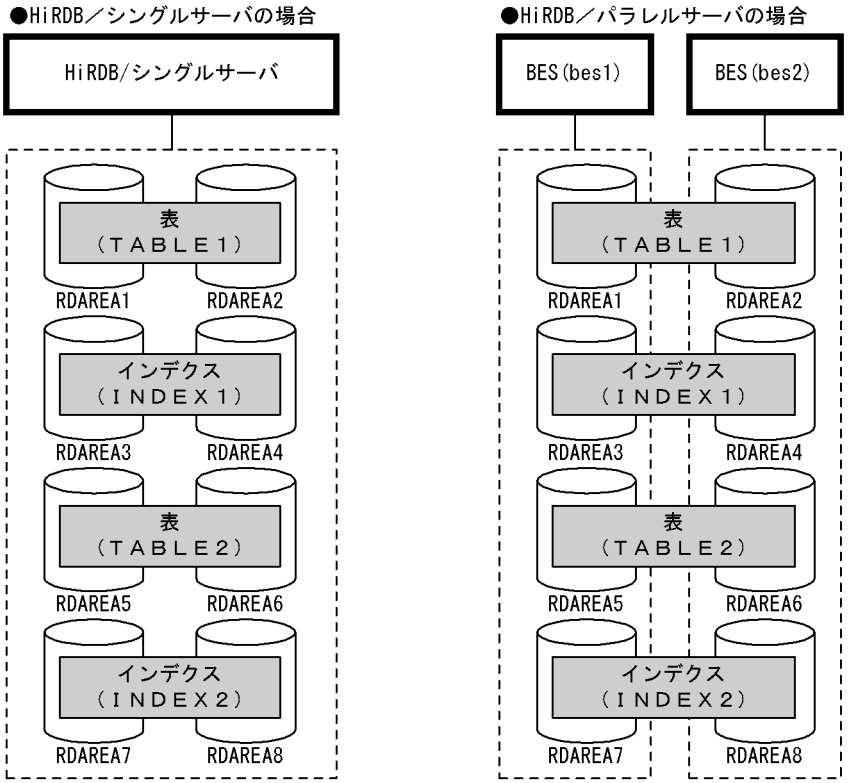
(18)　pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

12.1.7　例題３（スキーマ単位に表を移行する場合）

HiRDB システムＡの認可識別子 USR01 が所有するすべての表を HiRDB システムＢに移行します。なお，移行先システムには，同じ名称の RD エリアが作成されているとします。また，表のアンロード及びリロードをスキーマ単位で行います。



(1)　pdhold コマンドで移行対象のデータを格納している RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,…
```

(2) pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a) HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイルの名称を指定します。

(b) HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイルの名称を指定します。-g オプションが仮定されるため，アンロードデータファイルを 1 か所に作成します。

(3) pdrorg コマンドでスキーマ単位のアンロードします

```
pdrorg -k unld -t USR01.all C:¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-t：アンロード対象のスキーマの認可識別子を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

備考

スキーマ単位のアンロードでは-j オプション（LOB データがあるときのアンロード）が仮定されています。また，HiRDB/パラレルサーバの場合は-g オプション（アンロードデータファイルの一元化）も仮定されています。

(4) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,…
```

(5) pdexp コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdexp¥exp01）の内容を次に示します。

```
-t USR01.TABLE1
-t USR01.TABLE2
```

〔説明〕

USR01：TABLE1，TABLE2 を所有するスキーマの名称です。

TABLE1，TABLE2：搬出する表の名称です。

(6) pdexp コマンドですべての表の表定義情報を搬出します

```
pdexp -e C:¥pdexp¥expfile1 -f C:¥pdexp¥exp01
```

〔説明〕

-e：搬出ファイルの名称を指定します。

-f：(5)で作成した pdexp コマンドの制御文ファイル名を指定します。

(7) アンロードデータファイル及び搬出ファイルを CMT などの媒体に格納します

(3)で作成したアンロードデータファイル，及び(6)で作成した搬出ファイルを CMT などの媒体に格納します。

移行元システム（HiRDB システムＡ）での作業はこれで終わりです。

(8) 移行先システムにアンロードデータファイル及び搬出ファイルを格納します

これ以降は移行先システム（HiRDB システムＢ）での作業になります。

CMT などの媒体に格納したアンロードデータファイル及び搬出ファイルを移行先システムに格納します。

(9) スキーマ（認可識別子：USR01）を pddef コマンドで定義します

別スキーマ（USR01 以外）に表を移行する場合は，この操作は必要ありません。

```
pddef
   CREATE SCHEMA AUTHORIZATION USR01;
```

(10) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(11)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

データベース移行中の障害発生に備えてバックアップを取得します。バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(12)　pdexp コマンドで表定義情報を搬入します

```
pdexp -i C:¥pdexp¥expfile1
```

〔説明〕

-i：搬出ファイルの名称を指定します。

(13)　pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,…
```

(14)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

別スキーマ（USR01 以外）に表を移行する場合は，制御文ファイルの内容が変わります。別スキーマに表を移行する場合の制御文ファイルの例については，「別スキーマに表を移行する場合の制御文ファイルの例」を参照してください。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
idxwork bes2 C:¥pdrorg¥idxwork              4
sort bes2 C:¥sortwork                       5
```

〔説明〕

1. アンロードデータファイルの名称を指定します。-g オプションが仮定されるため，アンロードデータファイルを１か所に作成します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(15)　pdrorg コマンドですべての表のデータをリロードします

```
pdrorg -k reld -t USR01.all C:¥pdrorg¥rorg01
```

〔説明〕

インデクス（INDEX1）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：リロードをするため reld を指定します。

-t：アンロード対象のスキーマの認可識別子を指定します。別スキーマ（USR01 以外）に表を移行する場合は，移行先のスキーマの認可識別子を指定してください。認可識別子 USR02 に移行する場合は**-t USR02.all** と指定します。

C:¥pdrorg¥rorg01：(14)で作成した pdrorg コマンドの制御文ファイル名を指定します。

備考

スキーマ単位のアンロードでは-j オプション（LOB データがあるときのアンロード）が仮定されています。また，HiRDB/パラレルサーバの場合は-g オプション（アンロードデータファイルの一元化）も仮定されています。

(16) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(17) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(18) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

12.1.8　別スキーマに表を移行する場合の制御文ファイルの例

別スキーマに表を移行する場合のデータベース再編成ユティリティの制御文ファイルの例を次に示します。移行元のスキーマの認可識別子を USR01 とし，移行先のスキーマの認可識別子を USR02 とします。

(1)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
tblname USR01                               4
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。
4. 移行元のスキーマの認可識別子を指定します。

(2)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
idxwork bes2 C:¥pdrorg¥idxwork              4
sort bes2 C:¥sortwork                       5
tblname USR01                               6
```

〔説明〕

1. アンロードデータファイルの名称を指定します。-g オプションが仮定されるため，アンロードデータファイルを１か所に作成します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。
6. 移行元のスキーマの認可識別子を指定します。

(3) pdrorg コマンドの例

```
pdrorg -k reld -t USR02.all C:¥pdrorg¥rorg01
```

〔説明〕

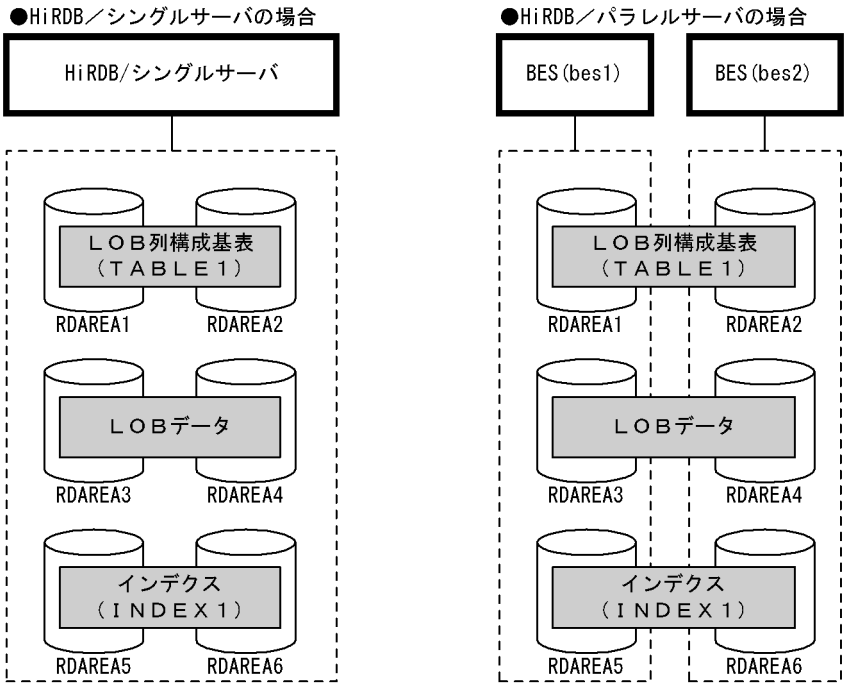
-t：アンロード対象のスキーマの認可識別子を指定します。

12.1.9 例題４（エンディアンが異なるシステムに，バイナリデータを含む表を移行する場合）

HiRDB システムＡ（ビッグエンディアン）の表（TABLE1）を HiRDB システムＢ（リトルエンディアン）に移行します。

- TABLE1 には LOB 列が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。

なお，移行先システムには，同じ名称の RD エリア（RDAREA1～RDAREA6）が作成されているとします。



(1) pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(2) pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイルの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイル（bes1 用）の名称を指定します。

(3)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -g -W bin -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-g：HiRDB/パラレルサーバで，TABLE1 をサーバ間横分割している場合に指定してください。アンロードデータファイルが一つになります。

-W：pdload 用の入力データファイルを作成する場合に指定します。この場合，TABLE1 にはバイナリ形式の列を含むため，バイナリ形式の bin を指定します。

-t：アンロードする表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4)　pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(5)　pddefrev コマンドで定義系 SQL を生成します

```
pddefrev -f ctrl1 -o C:¥pddefrev¥expfile1
```

〔説明〕

-f：定義系 SQL を生成したい表やストアドプロシジャを指定した制御文ファイルを指定します。

-o：生成した定義系 SQL を出力するファイルの名称を指定します。

(6) アンロードデータファイル及び搬出ファイルを CMT などの媒体に格納します

(3)で作成したアンロードデータファイル，及び(5)で生成したファイルを CMT などの媒体に格納します。

移行元システム（HiRDB システムＡ）での作業はこれで終わりです。

(7) 移行先システムにアンロードデータファイル及び搬出ファイルを格納します

これ以降は移行先システム（HiRDB システムＢ）での作業になります。

CMT などの媒体に格納したアンロードデータファイル及び搬出ファイルを移行先システムに格納します。

(8) TABLE1 を所有するユーザのスキーマを pddef コマンドで定義します

```
pddef
   CREATE SCHEMA AUTHORIZATION USR01;
```

(9) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(10) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

データベース移行中の障害発生に備えてバックアップを取得します。バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(11)　pddef コマンドで定義系 SQL を実行します

```
pddef < expfile1
```

〔説明〕

(5)で生成した定義系 SQL を記述した入力ファイルを指定します。

(12)　pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(13)　pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
source C:¥pdrorg¥unfile1                    1
option srcendian=big                        2
idxwork C:¥pdrorg¥idxwork                   3
sort C:¥sortwork                            4
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. 1.で指定した，アンロードデータファイルのエンディアン形式を指定します。
3. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
4. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
option srcendian=big                        2
idxwork bes1 C:¥pdrorg¥idxwork              3
sort bes1 C:¥sortwork                       4
```

```
idxwork bes2 C:¥pdrorg¥idxwork                  5
sort bes2 C:¥sortwork                           6
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. 1.で指定した，アンロードデータファイルのエンディアン形式を指定します。
3. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
4. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
5. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
6. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(14) pdload コマンドで TABLE1 のデータをロードします

```
pdload -b -W TABLE1 C:¥pdload¥load01
```

〔説明〕

インデクス（INDEX1）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-b：pdrorg コマンドで-W オプションを指定して，バイナリ形式で出力したファイルを入力データファイルとするため，-W を指定します。

C:¥pdload¥load01：(13)で作成した pdload コマンドの制御文ファイル名を指定します。

(15) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(16) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
-b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(17)　pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

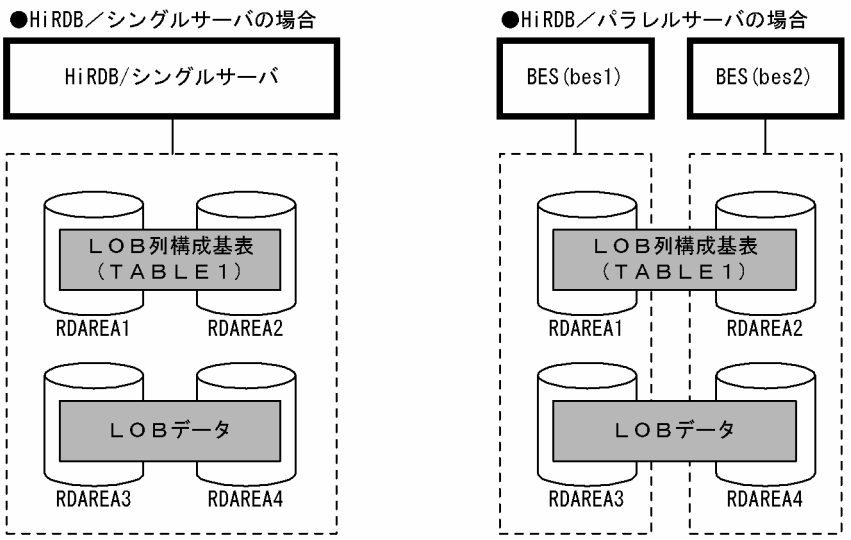
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

12.1.10　例題５（固定長文字データの列の定義長を拡張して，表を移行する場合）

HiRDB システムＡの表（TABLE1）を HiRDB システムＢに移行します。

- TABLE1 には LOB 列が定義されています。

なお，移行先システムには，同じ名称の RD エリア（RDAREA1～RDAREA4）が作成されているとします。



(1)　pdhold コマンドで RDAREA1～RDAREA4 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイルの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイル（bes1 用）の名称を指定します。

(3)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -g -W bin -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-g：HiRDB/パラレルサーバで，TABLE1 をサーバ間横分割している場合に指定してください。アンロードデータファイルが一つになります。

-W：pdload 用の入力データファイルを作成する場合に指定します。この場合，バイナリ形式の入力データファイルを作成するため，bin を指定します。

-t：アンロードする表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4) pdrels コマンドで RDAREA1～RDAREA4 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

(5) アンロードデータファイル及び搬出ファイルを CMT などの媒体に格納します

(3)で作成したアンロードデータファイルを CMT などの媒体に格納します。

移行元システム（HiRDB システムＡ）での作業はこれで終わりです。

(6) 移行先システムにアンロードデータファイル及び搬出ファイルを格納します

これ以降は移行先システム（HiRDB システムＢ）での作業になります。

CMT などの媒体に格納したアンロードデータファイル及び搬出ファイルを移行先システムに格納します。

(7) TABLE1 を所有するユーザのスキーマを pddef コマンドで定義します

```
pddef
   CREATE SCHEMA AUTHORIZATION USR01;
```

(8) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(9)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4
-b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

データベース移行中の障害発生に備えてバックアップを取得します。バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 移行する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA4）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(10)　列の定義長を拡張した TABLE1 を pddef コマンドで定義します

```
pddef
   CREATE TABLE TABLE1
     (商品コード CHAR(10) NOT NULL,
      商品イメージ BLOB(10K) IN ((RDAREA3),(RDAREA4))
     ) IN ((RDAREA1) 商品コード = ’ 00000001’ ,(RDAREA2));
```

〔説明〕

移行元システムでの表定義を次に示します。

```
CREATE TABLE TABLE1
  (商品コード CHAR(8) NOT NULL,
   商品イメージ BLOB(10K) IN ((RDAREA3),(RDAREA4))
  ) IN ((RDAREA1) 商品コード = ’ 00000001’ ,(RDAREA2))
```

移行先システムでは，商品コードの列の定義長を拡張しています。

なお，HiRDB システム B に既に TABLE1 が定義されている場合は，DROP TABLE で表を削除してから定義してください。

(11)　pdhold コマンドで RDAREA1～RDAREA4 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

(12)　pdload コマンドの列構成情報ファイルを作成します

列構成情報ファイル（C:¥pdload¥column01）の内容を次に示します。

```
商品コード,type=char(8)                    1.
商品イメージ                               2.
```

〔説明〕

1. 移行元システムでの列名及びデータ型を指定します。
2. 移行元システムでの列名を指定します。

(13)　pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
source C:¥pdrorg¥unfile1                   1
idxwork C:¥pdrorg¥idxwork                  2
sort C:¥sortwork                           3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1              1
idxwork bes1 C:¥pdrorg¥idxwork             2
sort bes1 C:¥sortwork                      3
idxwork bes2 C:¥pdrorg¥idxwork             4
sort bes2 C:¥sortwork                      5
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。

3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(14) pdload コマンドで TABLE1 のデータをロードします

```
pdload -b -W -c C:¥pdload¥column01 TABLE1 C:¥pdload¥load01
```

〔説明〕

-b：pdrorg コマンドで-W オプションを指定して，バイナリ形式で出力したファイルを入力データファイルとするため，-W を指定します。

-c：列の定義長を変更するため，列構成情報ファイルを指定します。

C:¥pdload¥load01：(13)で作成した pdload コマンドの制御文ファイル名を指定します。

(15) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(16) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,RDAREA1,RDAREA2,RDAREA3,RDAREA4
-b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 移行する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA4）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(17)　pdrels コマンドで RDAREA1～RDAREA4 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

12.1.11　インポート時に障害が発生した場合の対処方法

移行先システムで，pdload コマンドで表をインポートする場合に障害が発生したときに考えられる要因と対処を次の表に示します。

表 12–3　インポートする場合に障害が発生したときに考えられる要因と対処

障害	考えられる要因	対処	再度実行する pdload のオプション
定義エラー	移行先システムでインポートするユーザのスキーマが定義されていません。	インポートするユーザのスキーマを定義して，インポートを再度実行してください。	-b -W -w all
	移行先システムで，インポートしようとする表が既に定義されています。	移行元システムと同じ定義の表をインポートする場合： 移行先システムで既に定義されている表定義を削除してから，インポートを再度実行してください。 移行元システムと異なる定義の表をインポートする場合： 表定義情報と表データを別々に移行する方法で移行してください。	
	移行先システムに，インポートする表を格納する RD エリアがありません。	次のどちらかの対処をしてから，表データだけインポートしてください。 • SQL 定義ファイル※中の格納 RD エリアを修正して，pddef で再度定義 • 移行先システムに格納 RD エリアを追加	-b -W -w data
	表定義が 2MB を超えています。	SQL 定義ファイル※中の SQL を修正して 2MB 以内になるようにして，pddef で再度定義してください。その後，表データだけインポート（-b -W -w data オプション指定で pdload を実行）してください。	

障害	考えられる要因	対処	再度実行する pdload のオプション
		SQL 定義ファイルの修正方法については，「SQL 定義ファイルの修正例」を参照してください。	
データロードのエラー	マニュアル「HiRDB コマンドリファレンス」の「エラー時のデータベースの状態とその回復方法」を参照して，対処してください。		

注※

SQL 定義ファイルとは，pdload コマンドでインポートした表定義情報を基に表やインデクスを定義した SQL 文を出力するファイルです。インポート時に障害が発生した場合に出力されます。ファイル名は次のとおりです。

```
出力先ディレクトリ/SQLFILE-表名-xxxxxxx
```

出力先ディレクトリ：pd_tmp_directory に指定したディレクトリ

環境変数 TMP の指定があれば TMP に指定したディレクトリ，指定がなければ%PDDIR%¥tmp ディレクトリになります。詳細については，マニュアル「HiRDB コマンドリファレンス」の「データベース作成ユティリティ（pdload）」の「制御文で指定しない場合のファイル出力先ディレクトリ」を参照してください。

表名：インポートする表名

xxxxxxx：ファイル作成時刻とプロセス ID を文字列に変換した値

(1)　SQL 定義ファイルの修正例

インポート時に作成する SQL には，省略時仮定値も作成されるため，表定義が 2MB を超えることがあります。この場合，HiRDB は 2MB を超えないように列 ID の大きいものから列定義部分を削除して CREATE TABLE を作成し，その後で ALTER TABLE で列定義部分を追加します。このとき，SQL エラーが発生するおそれがあります。

SQL 定義ファイルの SQL を次のように修正してください。

1. CREATE TABLE の省略時仮定値の部分を削除します。
2. ALTER TABLE で追加している列定義を CREATE TABLE に移動します。

具体的な SQL の例は，マニュアル「HiRDB コマンドリファレンス」の「搬入時に作成される定義系 SQL が 2,000,000 バイトを超えた場合の修正例」を参照してください。

12.2　ストアドプロシジャをほかの HiRDB システムに移行する方法

実行者　HiRDB 管理者

ディクショナリ搬出入ユティリティを使用すると，現在使用しているストアドプロシジャをほかの HiRDB システムに移行できます。この機能を使用すると，次に示すことができます。

- テスト用システムで使用していたストアドプロシジャを本番用システムに移行できます。
- ある HiRDB システムで作成したストアドプロシジャを，ほかの HiRDB システムに移行できます。
- システムを再構築するときに既存のストアドプロシジャを保存でき，システムを再構築した後にそのストアドプロシジャを回復できます。

12.2.1　ストアドプロシジャをほかの HiRDB システムに移行する前に

(1)　制限事項

CREATE TYPE 中で指定する手続き（ストアドプロシジャ）は，ほかの HiRDB システムに移行できません。

(2)　移行する情報

移行する情報は次のとおりです。

- 表定義情報及び表データ
- ストアドプロシジャ

(a)　表定義情報及び表データの移行方法

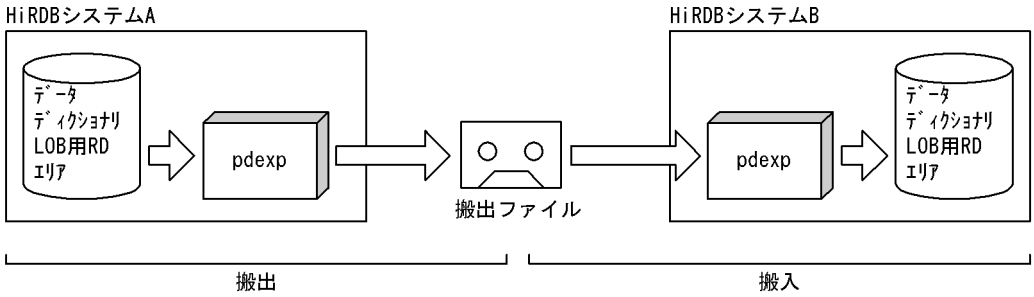
ストアドプロシジャが使用する表の表定義情報及び表データを移行先システムに移行します。移行方法については，「表をほかの HiRDB システムに移行する方法」を参照してください。

ただし，移行先システムにストアドプロシジャが使用する表の表定義情報及び表データがある場合は，この操作は不要です。

(b)　ストアドプロシジャの移行方法

ディクショナリ搬出入ユティリティ（**pdexp** コマンド）で，データディクショナリ LOB 用 RD エリア中のストアドプロシジャをほかの HiRDB システムに移行します。ディクショナリ搬出入ユティリティを使用したストアドプロシジャの移行を次の図に示します。

図 12–10　ディクショナリ搬出入ユティリティを使用したストアドプロシジャの移行



注　HiRDBシステムAのストアドプロシジャはそのまま残ります。

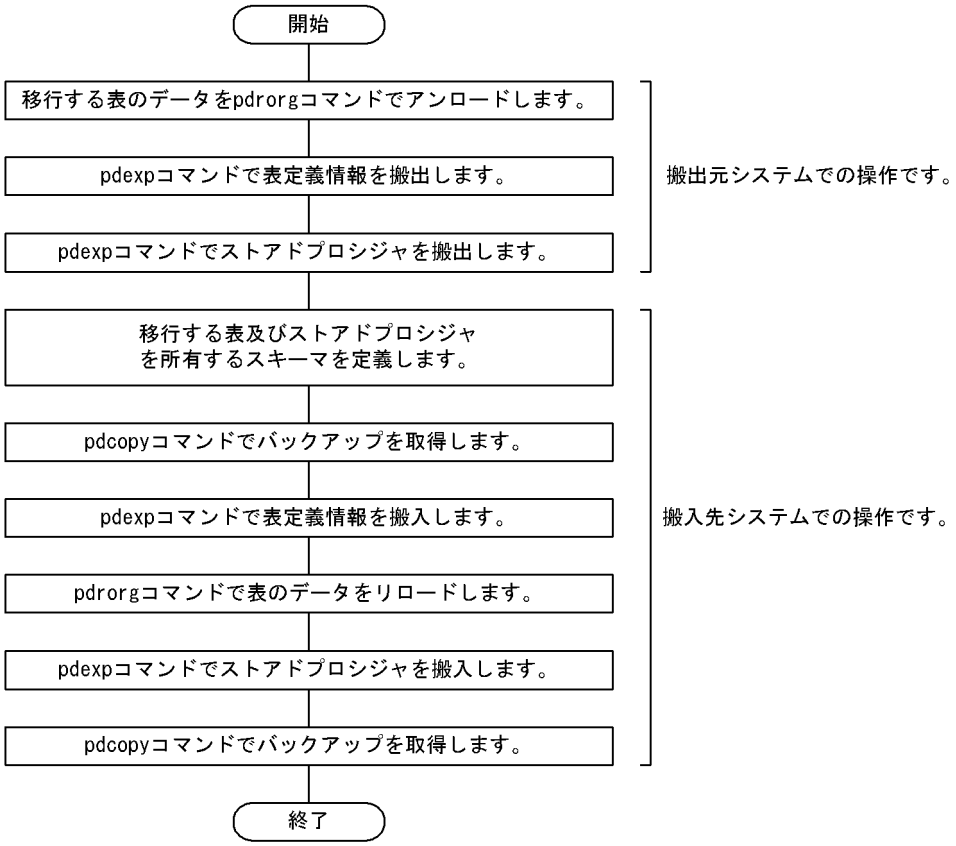
(3)　移行するときの規則

1. ストアドプロシジャと表定義情報を同時に移行できません。表定義情報を移行した後にストアドプロシジャを移行してください。
2. 移行先システムに，移行する表及びストアドプロシジャのスキーマを定義しておいてください。
3. 移行先システムに，データディクショナリ LOB 用 RD エリアを作成しておいてください。

(4)　移行手順

ストアドプロシジャの移行手順の概略を次の図に示します。

図 12–11　ストアドプロシジャの移行手順

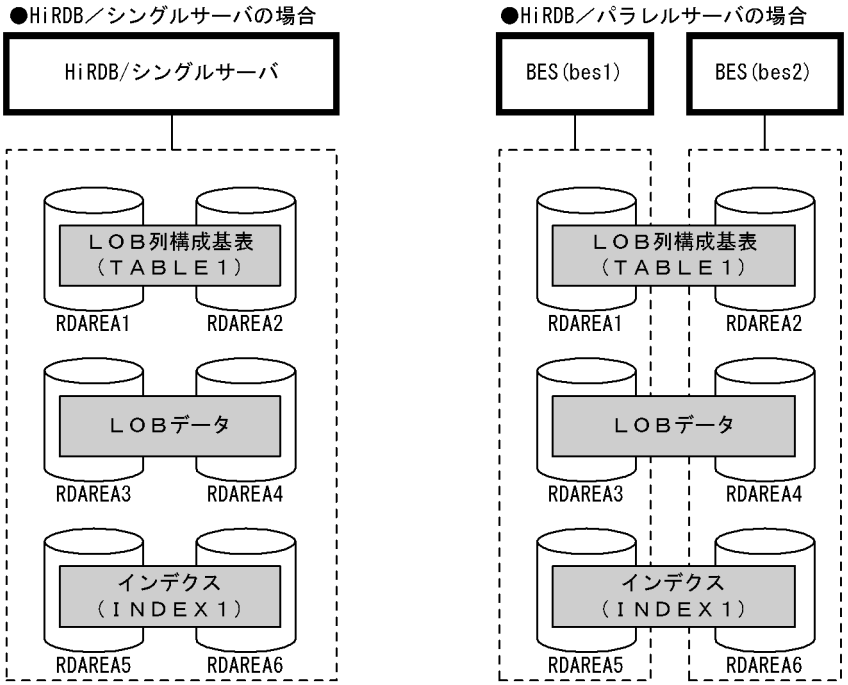


12.2.2　例題

ストアドプロシジャ（ルーチン識別子：PROC1）及びストアドプロシジャが使用している表（TABLE1）を HiRDB システムＡから HiRDB システムＢに移行します。

- TABLE1 には LOB 列が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。

なお，移行先システムには，同じ名称の RD エリア（RDAREA1～RDAREA6）が作成されているとします。



(1)　pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1
```

〔説明〕

アンロードデータファイルの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1            1
unload bes2:C:¥pdrorg¥unfile2            2
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. アンロードデータファイル（bes1 用）の名称を指定します。

(3)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -W bin -g -t TABLE1 C;¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-g：HiRDB/パラレルサーバで，TABLE1 をサーバ間横分割している場合に指定してください。アンロードデータファイルが一つになります。

-t：アンロードする表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4)　pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(5)　pdexp コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdexp¥exp01）の内容を次に示します。

```
-t USR01.TABLE1
```

〔説明〕

USR01：TABLE1 を所有するスキーマの名称です。

TABLE1：搬出する表の名称です。

(6)　pdexp コマンドで TABLE1 の表定義情報を搬出します

```
pdexp -e C:¥pdexp¥expfile1 -f C:¥pdexp¥exp01
```

〔説明〕

-e：搬出ファイルの名称を指定します。

-f：(5)で作成した pdexp コマンドの制御文ファイル名を指定します。

(7) pdexp コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdexp¥exp02）の内容を次に示します。

```
-p USR01.PROC1
```

〔説明〕

USR01：PROC1 を所有するスキーマの名称です。

PROC1：搬出するストアドプロシジャの名称です。

(8) pdexp コマンドでストアドプロシジャを搬出します

```
pdexp -e C:¥pdexp¥expfile2 -f C:¥pdexp¥exp02
```

〔説明〕

-e：搬出ファイルの名称を指定します。

-f：(7)で作成した pdexp コマンドの制御文ファイル名を指定します。

(9) アンロードデータファイル及び搬出ファイルを CMT などの媒体に格納します

(3)で作成したアンロードデータファイル，及び(6)，(8)で作成した搬出ファイルを CMT などの媒体に格納します。移行元システム（HiRDB システムＡ）での作業はこれで終わりです。

(10) 移行先システムにアンロードデータファイル及び搬出ファイルを格納します

これ以降は移行先システム（HiRDB システムＢ）での作業になります。CMT などの媒体に格納したアンロードデータファイル及び搬出ファイルを移行先システムに格納します。

(11) PROC1 及び TABLE1 を所有するユーザのスキーマを pddef コマンドで定義します

```
pddef
   CREATE SCHEMA AUTHORIZATION USR01;
```

(12) システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(13) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,DICLOB,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,
RDAREA6 -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

データベース移行中の障害発生に備えてバックアップを取得します。バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- データディクショナリ LOB 用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(14) pdexp コマンドで TABLE1 の表定義情報を搬入します

```
pdexp -i C:¥pdexp¥expfile1
```

〔説明〕

-i：搬出ファイルの名称を指定します。

(15) pdhold コマンドで RDAREA1～RDAREA6 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(16)　pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
source C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
source bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
idxwork bes2 C:¥pdrorg¥idxwork              4
sort bes2 C:¥sortwork                       5
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(17)　pdload コマンドで TABLE1 のデータをデータロードします

```
pdload -b -W TABLE1 C:¥pdload¥load01
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE1：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(18)　システムログファイルをスワップします

バックアップを取得する前に，pdlogswap コマンドでシステムログファイルをスワップします。

```
pdlogswap -d sys -w
```

〔説明〕

HiRDB/パラレルサーバの場合，-s オプションを指定して，バックアップ対象の全サーバ分 pdlogswap コマンドを実行します。

(19)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r
-r RDMAST,RDDIR,RDDIC,DICLOB,RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,
RDAREA6 -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

バックアップを取得する RD エリアを次に示します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- データディクショナリ LOB 用 RD エリア
- 搬入する表を格納するユーザ用 RD エリア及びユーザ LOB 用 RD エリア（RDAREA1～RDAREA6）

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(20)　pdrels コマンドで RDAREA1～RDAREA6 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(21)　pdexp コマンドでストアドプロシジャを搬入します

```
pdexp -i C:¥pdexp¥expfile2
```

〔説明〕

-i：搬出ファイルの名称を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13 表の運用

この章では，表の運用方法について説明します。

13.1　表の格納効率を調べる方法

実行者　HiRDB 管理者又は DBA 権限保持者

データの追加又は削除を繰り返すと，行の配置が乱れてデータの検索効率や格納効率が低下します。したがって，HiRDB 管理者はデータの格納効率が低下していないかどうかを確認する必要があります。データの格納効率が低下している場合は次に示すどれかの処置をしてください。

- 表を再編成する
- RD エリアの容量を大きくする
- 使用中空きページ及び使用中空きセグメントを解放する

> **参考**
>
> これらの作業を行う必要がある表，インデクス，又は RD エリアを HiRDB が特定してくれます。詳細については，「表の再編成時期を予測する方法（再編成時期予測機能）」を参照してください。

13.1.1　定期的にデータベース状態解析ユティリティを実行します

データベース状態解析ユティリティ（pddbst コマンド）を定期的に実行して，データの格納効率を確認してください。

(1)　表の格納効率を確認するには

表の格納効率を確認するには，次のどちらかの方法を実施してください。

- HiRDB Control Manager のデータベース状態解析
- データベース状態解析ユティリティの「表単位の状態解析」

上記の方法で確認した情報から，次に示す作業が必要かどうかを判断してください。

- 表の再編成
- 使用中空きページ及び使用中空きセグメントの解放
- 不要な行の削除
- 表の分割格納条件の変更
- ハッシュ関数の変更
- インデクスの再作成又は再編成
- RD エリアの拡張，再初期化，追加，又は削除

抽象データ型を定義した表については，表単位の状態解析はできません。抽象データ型を定義した表については，RD エリア単位の状態解析（物理的解析）で表データの格納効率を確認してください。

(2) インデクスの格納効率を確認するには

インデクスの格納効率を確認するには，次のどちらかの方法を実施してください。

- HiRDB Control Manager のデータベース状態解析
- データベース状態解析ユティリティの「インデクス単位の状態解析」

上記の方法で確認した情報から，次に示す作業が必要かどうかを判断してください。

- インデクスの再作成又は再編成
- 使用中空きページ及び使用中空きセグメントの解放
- インデクスページスプリット発生回数の削減

なお，インデクス単位の状態解析はプラグインインデクスに関する情報を取得しません。

(3) クラスタキー及びクラスタリングデータページの格納効率を確認するには

クラスタキー及びクラスタリングデータページの格納効率を確認するには，データベース状態解析ユティリティの「クラスタキー及びクラスタリングデータページの格納状態解析」を実施してください。ここで確認した情報から，次に示す作業が必要かどうかを判断してください。

- 表の再編成
- インデクス構成列中のデータ重複度の高い列を除いて，クラスタキーを指定した表の再定義
- インデクス定義の列構成の見直し

(4) RD エリア内のデータ格納効率を確認するには

RD エリア内のデータ格納効率を確認するには，次のどちらかの方法を実施してください。

- HiRDB Control Manager のデータベース状態解析
- データベース状態解析ユティリティの「RD エリア単位の状態解析」

上記の方法で確認した情報から，次に示す作業が必要かどうかを判断してください。

- 表の再編成
- インデクスの再作成又は再編成
- 使用中空きページ及び使用中空きセグメントの解放
- 表の分割格納条件の変更
- ハッシュ関数の変更

- RD エリアの拡張，再初期化，追加，又は削除

●そのほかの方法でも RD エリアの使用状況が確認できます

- pddbls -a コマンドでも RD エリアの使用状況が確認できます。
- データベース複写ユティリティ（pdcopy コマンド）でバックアップを取得したときに出力される RD エリアの使用状況でも，RD エリアの格納効率が確認できます。

(5)　HiRDB Control Manager を使用してデータベースの状態解析をする場合

HiRDB Control Manager のデータベース状態解析機能を使用すると，データベースの格納状態を視覚的に確認できます。機能の使用手順を次に示します。

1. HiRDB Control Manager - Console を起動します。
   起動方法については，「HiRDB Control Manager - Console の起動方法」を参照してください。
2. 操作対象の HiRDB サーバを登録します。
   既に登録されている場合は，この手順は必要ありません。登録方法については，「管理 HiRDB の登録方法」を参照してください。
3. タブメニューの［**データメンテナンス**］－［**データベース状態解析**］を選択して，［データベース状態解析ユティリティ］画面を表示します。
4. 確認する対象を「情報種別」で選択します。
   - 表の格納効率を確認する場合
     「ユーザ表」を選択してください。
   - インデクスの格納効率を確認する場合
     「ユーザインデクス」を選択してください。
   - RD エリアの格納効率を確認する場合
     「RD エリア」を選択してください。

   「情報種別」を選択すると，対象資源の一覧が表示されます。この一覧から，確認したい資源（表，インデクス又は RD エリア）を選択します。



5. **[状態解析]**（  ）をクリックします。

   「ページ使用状態」に，状態解析結果が表示されます。

   画面上部の各ボタンから，表の再編成や空きページの解放などが実行できます。ページ使用状態の傾向と対策については，HiRDB Control Manager のヘルプを参照してください。



13.1.2　データの格納効率が悪くなるとメッセージが出力されます

データの格納効率が悪くなると，HiRDB は次に示すメッセージを出力します。

- KFPA12300-I 又は KFPH00211-I
- KFPH00212-I
- KFPH22017-I

(1)　KFPA12300-I 又は KFPH00211-I が出力された場合

RD エリアの容量が不足気味になると，KFPA12300-I 又は KFPH00211-I メッセージ（RD エリアの最終ファイル，又は RD エリア全体の容量の使用率を示すメッセージ）が出力されます。

HiRDB 管理者の処置

pddbls -a コマンド又はデータベース状態解析ユティリティの「RD エリア単位の状態解析」を実施して，RD エリアの構成を変更するかどうかを決めてください。RD エリアの構成を変更する場合は，「RD エリアの運用」を参照してください。

(2) KFPH00212-I が出力された場合

表の検索効率や格納効率が低下すると，KFPH00212-I メッセージが出力されます。

HiRDB 管理者の処置

データベース状態解析ユティリティの「表単位の状態解析」を実施して，次に示すどちらかの処置をしてください。

- 表を再編成する
- 使用中空きページ及び使用中空きセグメントを解放する

表を再編成する場合は，「表の再編成」及び「表の再編成（例題）」を参照してください。使用中空きページ及び使用中空きセグメントを解放する場合は，「使用中空きページ及び使用中空きセグメントを再利用する方法」を参照してください。

ただし，次の場合は，データベース構成変更ユティリティ（pdmod コマンド）で RD エリアの構成を変更する必要があります。RD エリアの構成を変更する場合は，「RD エリアの運用」を参照してください。

- 同じ RD エリア内の表に対してこのメッセージが頻繁に出力される場合
- 表の再編成中又は表の再編成の直後にこのメッセージが出力される場合
- インデクスの再編成中又はインデクスの再編成の直後にこのメッセージが出力される場合

(3) KFPH22017-I が出力された場合

LOB データの検索効率や格納効率が低下すると，KFPH22017-I メッセージが出力されます。

HiRDB 管理者の処置

データベース状態解析ユティリティの「表単位の状態解析」を実施して，表を再編成するかどうかを決めてください。表を再編成する場合は，「表の再編成」及び「表の再編成（例題）」を参照してください。

ただし，KFPH22017-I メッセージに表示された RD エリアの種類によって，再編成する RD エリアが異なります。

- **ユーザ LOB 用 RD エリアの場合**

  ユーザ LOB 用 RD エリアの LOB データを再編成してください。

- **データディクショナリ LOB 用 RD エリアの場合**

  次に示すストアドプロシジャ及びストアドファンクションに関するディクショナリ表を再編成してください。

  ・SQL_ROUTINES

  ・SQL_ROUTINE_RESOURCES

・SQL_ROUTINE_PARAMS

13.1.3　検索性能などに期待した効果が得られない場合

検索性能などに期待した効果が得られない場合は，データベース状態解析ユティリティを実行して，データの格納状態を確認することをお勧めします。これによって，データの格納状態を改善するかどうかを決めます。

なお，システムの性能を向上させるには，次に示すコマンドを実行して取得した統計情報に基づいた，システムのチューニングが必要な場合があります。

- 統計解析ユティリティ（pdstedit コマンド）
- pdbufls コマンド

システムのチューニング方法については，「チューニング情報の取得方法」及び「チューニング」を参照してください。

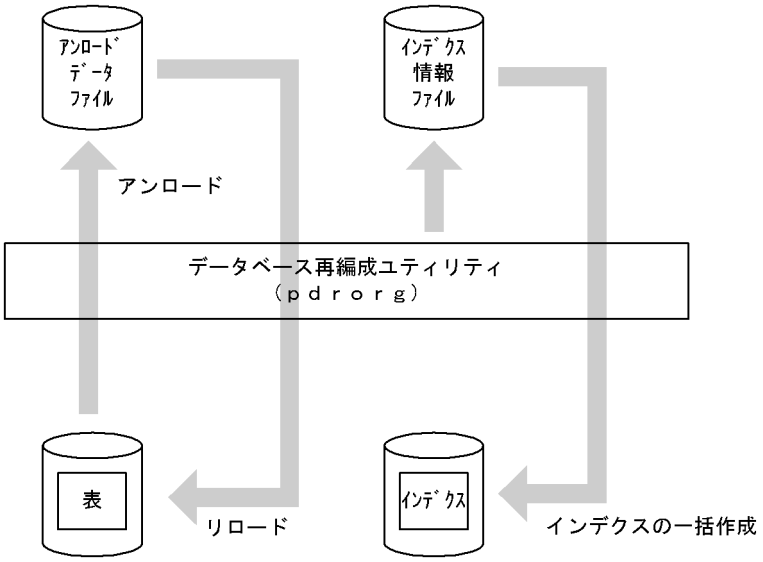
13.2　表の再編成

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

13.2.1　表の再編成とは

データの追加及び削除を繰り返すと，データの格納効率が悪くなり，データを検索するときの性能が低下します。これを防ぐために，定期的に **pdrorg** コマンド（データベース再編成ユティリティ）で表の再編成を実施してください。表の再編成の処理概要を次の図に示します。

図 13–1　表の再編成の処理概要



〔説明〕

- 表データをアンロードデータファイルに吸い上げます。これを表データの**アンロード**といいます。その後，表にデータを格納し直します。これを表データの**リロード**といいます。これら全体の処理を**表の再編成**といいます。
- 表にインデクスが定義されていると，データをリロードするときにインデクス情報ファイルにインデクス情報が出力されます。その情報を基にして HiRDB がインデクスを一括作成します。これによって，インデクスも再編成されます。

13.2.2　表の再編成の実行単位

表の再編成は次に示す単位で実行できます。

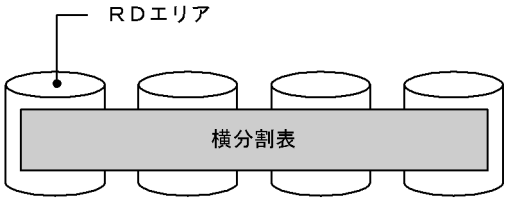
- 表単位の再編成
- RD エリア単位の再編成

- スキーマ単位の再編成

(1)　表単位の再編成

再編成処理を表単位に行います。通常はこの方法を実施してください。データベース状態解析ユティリティの結果から，表全体を再編成する必要がある場合に表単位の再編成を実行します。表単位の再編成を次の図に示します。

図 13–2　表単位の再編成



注　網掛け部分のデータが再編成対象となります。

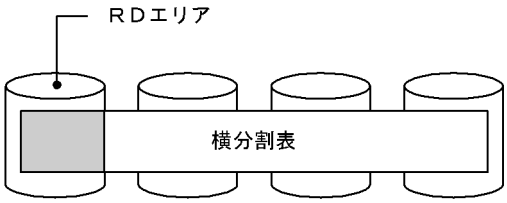
〔説明〕

データベース再編成ユティリティの-t オプションで再編成対象の表を指定します。

(2)　RD エリア単位の再編成

表を格納する RD エリア単位に再編成処理を行います。この方法は表を横分割しているときだけ有効です。データベース状態解析ユティリティの結果から，横分割表のある部分だけを再編成すればよい場合に RD エリア単位の再編成を実行します。この場合，表単位の再編成に比べて，処理時間を短縮できます。RD エリア単位の再編成を次の図に示します。

図 13–3　RD エリア単位の再編成



注　網掛け部分のデータが再編成対象となります。

〔説明〕

データベース再編成ユティリティの-t オプションで再編成対象の表を指定して，かつ-r オプションで再編成対象の RD エリアを指定します。

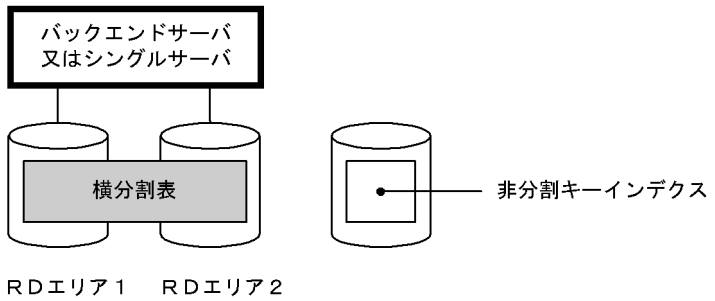
参考

次に示す条件をすべて満たす表を RD エリア単位で再編成すると，非分割キーインデクスが作成されません。非分割キーインデクスを作成するには，もう一度データベース再編成ユティリティ（インデクスの再作成）を実行する必要があります。

- 再編成対象の表がサーバ内横分割されている
- 再編成対象の表に非分割キーインデクスが定義されている
- 非分割キーインデクスを横分割していない

RD エリア単位の再編成時に非分割キーインデクスが作成されない例を次の図に示します。

図 13–4　RD エリア単位の再編成時に非分割キーインデクスが作成されない例



〔説明〕

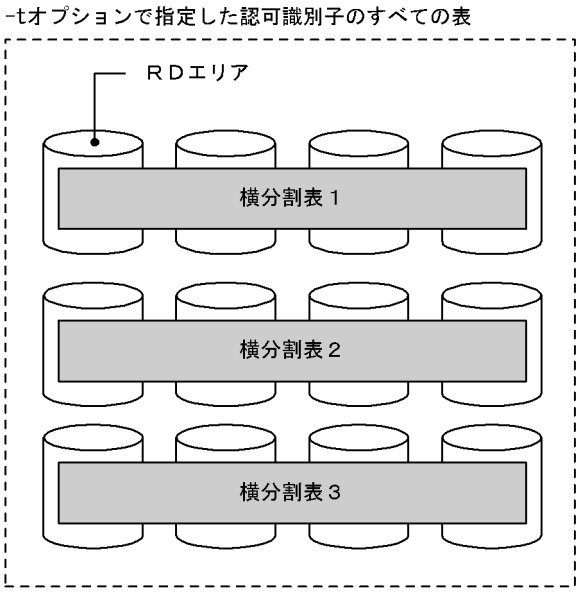
RD エリア１に対して RD エリア単位の再編成をしても，非分割キーインデクスは作成されません。インデクス情報ファイルにインデクス情報が出力されるだけです。非分割キーインデクスを作成するには，データベース再編成ユティリティでインデクスを再作成する必要があります。

したがって，この場合は RD エリア単位ではなく，表単位で再編成することをお勧めします。

(3)　スキーマ単位の再編成

スキーマ内のすべての表を一括して再編成します。所有する表の再編成を一括して行う場合にスキーマ単位の再編成を実行します。スキーマ単位の再編成を次の図に示します。

図 13–5　スキーマ単位の再編成



注　網掛け部分のデータが再編成対象となります。

〔説明〕

データベース再編成ユティリティの-t オプションで再編成対象のスキーマの認可識別子を指定します。指定形式は**-t 認可識別子.all** です。

13.2.3　データベースの更新ログ取得方式の選択

データベース再編成ユティリティを実行するときのデータベースの更新ログ取得方式には，次に示す三つのモードがあります。

- ログ取得モード
- 更新前ログ取得モード（省略値）
- ログレスモード

これらのモードの機能については，「データベースの更新ログ取得方式」を参照してください。

(1)　モードの選択基準

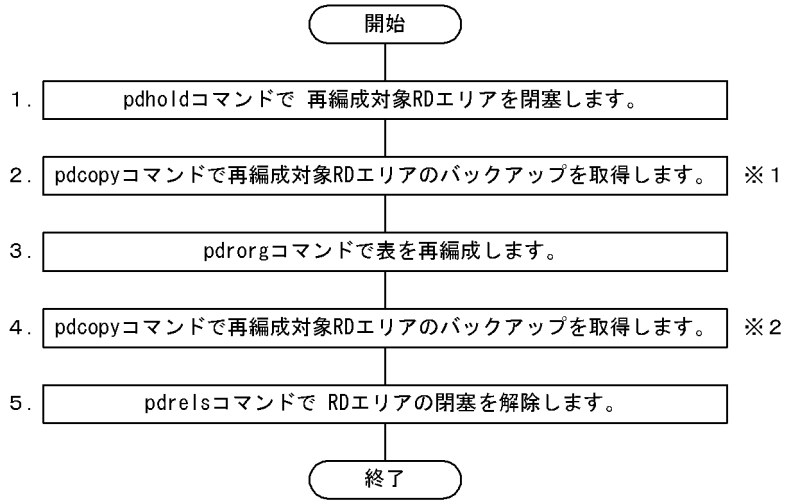
基本的には省略値である更新前ログ取得モードを選択してください。ただし，次に示す条件を満たすような場合はほかのモードの選択を検討してください。

条件	選択するモード
再編成対象の表データが多く，再編成に時間が掛かる	ログレスモード
再編成対象の表データが少ない	ログ取得モード

(2)　運用方法の違い

選択したモードによってデータベースを再編成するときの運用が異なります。運用方法の違いを次の図に示します。

図 13–6　データベースの更新ログ取得方式による運用方法の違い（表の再編成）



注※1

ログレスモードを選択したときに必要な操作です。ログレスモードの pdrorg コマンドが異常終了した場合，このバックアップを使用して RD エリアを回復します。ただし，「再編成の前にバックアップを取得しないでよい場合」で説明している条件を満たすときはバックアップを取得する必要はありません。

注※2

更新前ログ取得モード又はログレスモードを選択したときに必要な操作です。ここでバックアップを取得しないと，pdrstr コマンドで RD エリアを回復する必要が生じた場合，RD エリアを最新の状態に回復できません（pdrorg コマンド実行後の反映処理を回復できません）。RD エリアは pdrorg コマンドの実行前の状態にしか回復できません。

補足事項

更新前ログ取得モード又はログレスモードを選択した場合，図「データベースの更新ログ取得方式による運用方法の違い（表の再編成）」の手順 1～4 の間は再編成対象 RD エリアを閉塞したままにしてください。手順 4 でバックアップを取得する前に RD エリアの内容が更新された場合，pdrstr コマンドで RD エリアを回復する必要が生じたときにその更新内容を回復できません。RD エリアは pdrorg コマンドの実行前の状態にしか回復できません。pdrstr コマンドで RD エリアを回復するとき，入力情報のシステムログ中に更新前ログ取得モード又はログレスモードで取得したログが入っていると pdrstr コマンドがエラーになります。

(3)　再編成の前にバックアップを取得しないでよい場合

ログレスモードで pdrorg コマンドを実行する場合は，pdrorg コマンドの実行前にバックアップを取得する必要があります。ただし，次に示す 1，2 の条件のうちどちらかを満たす場合は，pdrorg コマンドが異常終了したときに RD エリアの状態を pdrorg コマンドの実行前の状態に戻せるため，バックアップの取得

を省略できます。ただし，バックアップを取得した方が RD エリアを回復するときの運用が簡単なため，基本的にはバックアップを取得することをお勧めします。

項番	条件		障害発生時の RD エリア回復方法
1	アンロードデータを使用して pdrorg コマンド実行前の状態に RD エリアを回復できる場合	RD エリア内に再編成対象の表，及びその表のインデクスだけを格納している場合	再編成対象の RD エリアをデータベース構成変更ユティリティ（pdmod コマンド）で再初期化した後に，再度リロードすると回復できます。
		次に示す条件をすべて満たす場合 • 再編成対象の表に LOB データがある • RD エリア内に再編成対象の LOB 列構成基表だけを格納している • - j オプションを指定してアンロードデータを取得している	再編成対象の RD エリアをデータベース構成変更ユティリティ（pdmod コマンド）で再初期化した後に，- j オプションを指定したリロードをすると回復できます。
2	バックアップとシステムログを使用して pdrorg コマンド実行前の状態に RD エリアを回復できる場合		pdclose コマンドで再編成対象の RD エリアをクローズした後に，pdlogswap コマンドでシステムログファイルをスワップして，データベース回復ユティリティ（pdrstr コマンド）にここまでのシステムログを入力すれば回復できます。

13.2.4　表を再編成する前に

(1)　LOB 列が定義されている表を再編成する場合

LOB 列構成基表と LOB データを同時に再編成できます。また，LOB 列構成基表だけを再編成したり，又は LOB データだけを再編成したりできます。なお，LOB 列構成基表と LOB データを同時に再編成する場合は，pdrorg コマンド実行時に**-j** オプションを指定することをお勧めします。

(2)　抽象データ型が定義されている表を再編成する場合

抽象データ型の種類によって，再編成できる場合とできない場合があります。再編成可否を次の表に示します。

表 13–1　抽象データ型が定義されている表の再編成可否

条件		再編成可否
プラグインが提供する抽象データ型	LOB 属性なし	再編成できます。
	LOB 属性あり	抽象データ型列構成基表だけを再編成できます。※
ユーザが定義した抽象データ型		再編成できません。

注※

プラグインに UNLOAD 機能又はコンストラクタパラメタ逆生成機能がある場合，-j オプションを指定すれば表全体を再編成できます。

(3)　大量データを格納した表を再編成する場合

大量データを格納した表を再編成する場合，**同期点指定の再編成**を実施するかどうかを検討してください。

通常，表の再編成処理では全データの格納処理を完了するまでトランザクションを決着できません。このため，データベース再編成ユティリティ実行中はシンクポイントダンプを有効化できません。したがって，大量データの再編成処理中に HiRDB が異常終了すると，HiRDB の再開始処理に長い時間を必要とします。これを防ぐために，データ格納時（リロード処理時）に任意の件数で同期点を設定してトランザクションを決着できます。これを同期点指定の再編成といいます。

また，データベースの更新ログ取得方式がログ取得モード又は更新前ログ取得モードの場合，同期点指定の再編成中にユティリティが異常終了すると，異常終了直前にコミットした次のデータからデータロードを再開始するため，処理時間が短縮できます。

同期点指定の再編成をするには，データベース再編成ユティリティの option 文で**同期点行数**（何件データを格納したら同期点を取得するか）を指定してください。

注意事項

1. この機能を適用しない場合よりも同期点処理が実行される分，処理性能が低下します。
2. ユティリティが異常終了したとき，そのタイミングによって対処方法が異なります。異常終了時の対処方法については，「同期点指定の再編成実行中にユティリティが異常終了したときの対処方法」を参照してください。
3. 表を複数のバックエンドサーバに横分割している場合，アンロードデータファイルを一元化してください（データベース再編成ユティリティで-g オプションを指定してください）。アンロードデータファイルを一元化しないと，ユティリティが異常終了した場合に対処方法が複雑になります。このときの対処方法については，「アンロードデータファイルを一元化しないときに異常終了した場合の対処方法（HiRDB/パラレルサーバ限定）」を参照してください。
4. 同期点ごとに新規ページからデータの格納を開始するため，この機能を使用しないときに比べてデータを格納するために必要なページ数が多くなります。したがって，満杯状態の表を再編成する場合はこの機能を使用しないようにしてください。使用した場合，容量不足でデータベース再編成ユティリティがエラーになることがあります。
5. 改竄防止表に対して同期点指定の再編成はできません。

(4)　満杯状態の RD エリア内の表を再編成する場合

表の再編成時，ページ内の未使用領域の比率は CREATE TABLE の PCTFREE オペランドの指定が適用されます。したがって，満杯状態の RD エリア内の表を再編成すると，表の再編成時に RD エリアの容量不足が発生することがあります。これを防ぐには，データベース再編成ユティリティ（pdrorg コマンド）の option 文で **tblfree** 及び **idxfree** オペランドを指定して，CREATE TABLE の PCTFREE オペランドで指定したページ内の未使用領域の比率を変更してください。

tblfree オペランドには表の未使用領域の比率を指定します。idxfree オペランドにはインデクスの未使用領域の比率を指定します。

なお，これはあくまで再編成前に RD エリアの拡張がすぐにできないときの暫定的な処置です。データの更新に備えて，CREATE TABLE の PCTFREE オペランドの値を考慮した再編成ができるように，データベース構成変更ユティリティ（pdmod コマンド）で RD エリアを拡張してください。

(5) HiRDB Datareplicator によるデータ連携をしている場合

抽出側データベースに対してデータベース再編成ユティリティを実行する場合，-l オプションには n 又は p を指定してください（ログレスモード又は更新前ログ取得モードでデータベース再編成ユティリティを実行してください）。

ログ取得モードでデータベース再編成ユティリティを実行した場合，抽出側データベースに対して実行された一部の更新情報だけが反映側データベースに転送されるため，抽出側と反映側のデータベースの内容が不整合になります。

(6) テープ装置アクセス機能

DAT，DLT，又は LTO 上にあるファイルを，テープ装置アクセス機能を使用してアクセスできます。次に示すファイルに対してテープ装置アクセス機能が適用できます。

- アンロードデータファイル（pdrorg コマンドの unload 文に指定するアンロードデータファイル）
- アンロードデータファイル（pdrorg コマンドの lobunld 文に指定する LOB データのアンロードデータファイル）

テープ装置アクセス機能については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(7) 再編成した後にすること

再編成した後に，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3　表の再編成（例題）

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

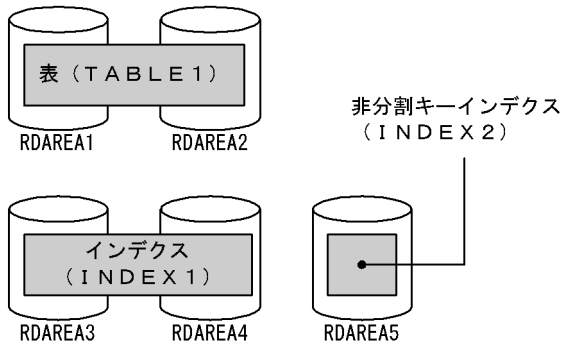
ここでは，表を再編成するときの運用例を説明します。説明する項目は次のとおりです。

- 例題１（表単位で再編成する場合：HiRDB/シングルサーバの場合）
- 例題２（表単位で再編成する場合：HiRDB/パラレルサーバの場合）
- 例題３（RD エリア単位で再編成する場合）
- 例題４（スキーマ単位で再編成する場合）
- 例題５（LOB 列が定義されている表を再編成する場合）
- 例題６（ディクショナリ表を再編成する場合）
- 例題７（ログレスモードで再編成する場合）
- 例題８（抽象データ型を定義した表を再編成する場合）

13.3.1　例題１（表単位で再編成する場合：HiRDB/シングルサーバの場合）

横分割表（TABLE1）を表単位で再編成します。再編成するときの条件は次のとおりです。

- TABLE1 はユーザ用 RD エリア（RDAREA1～RDAREA2）に横分割されています。
- TABLE1 には，横分割インデクス（INDEX1）及び非分割キーインデクス（INDEX2）が定義されています。
- INDEX1 はユーザ用 RD エリア（RDAREA3～RDAREA4）に横分割されています。
- INDEX2 はユーザ用 RD エリア（RDAREA5）に格納されています。
- 表を再編成するときにインデクスを一括作成（省略値）します。
- 更新前ログ取得モード（省略値）で表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。

2. **pdrorg** コマンドの制御文ファイルを作成します。
3. **pdrorg** コマンドで表を再編成します。
4. 再編成対象 RD エリアのバックアップを取得します。
5. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- 更新前ログ取得モードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 4）にバックアップを取得する必要があります。
- 更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 1～4 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(3)　pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

インデクス（INDEX1 及び INDEX2）も同時に一括作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：再編成をするため rorg を指定します。

-t：再編成する表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA5）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

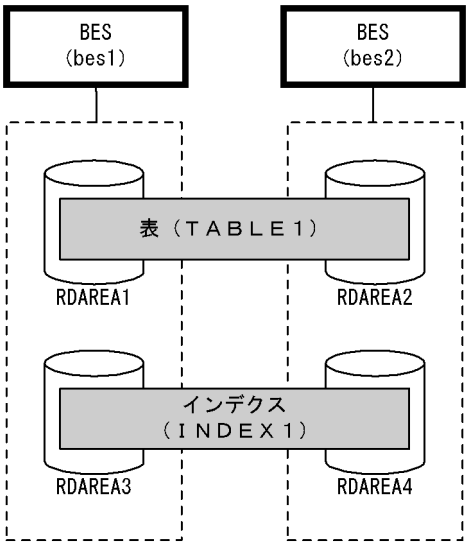
```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3.2　例題２（表単位で再編成する場合：HiRDB/パラレルサーバの場合）

横分割表（TABLE1）を表単位で再編成します。再編成するときの条件は次のとおりです。

- TABLE1 はユーザ用 RD エリア（RDAREA1～RDAREA2）に横分割されています。
- TABLE1 には，横分割インデクス（INDEX1）が定義されています。INDEX1 はユーザ用 RD エリア（RDAREA3～RDAREA4）に横分割されています。
- 表を再編成するときにインデクスを一括作成（省略値）します。
- 更新前ログ取得モード（省略値）で表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。

2. **pdrorg** コマンドの制御文ファイルを作成します。
3. **pdrorg** コマンドで表を再編成します。
4. 再編成対象 RD エリアのバックアップを取得します。
5. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- 更新前ログ取得モードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 4）にバックアップを取得する必要があります。
- 更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 1～4 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

```
unload bes1:C:¥pdrorg¥unfile1                1
idxwork bes1 C:¥pdrorg¥idxwork               2
sort bes1 C:¥sortwork                        3
unload bes2:C:¥pdrorg¥unfile2                4
idxwork bes2 C:¥pdrorg¥idxwork               5
sort bes2 C:¥sortwork                        6
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. アンロードデータファイル（bes2 用）の名称を指定します。
5. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
6. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(3)　pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

インデクス（INDEX1 及び INDEX2）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：再編成をするため rorg を指定します。

-t：再編成する表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA4）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

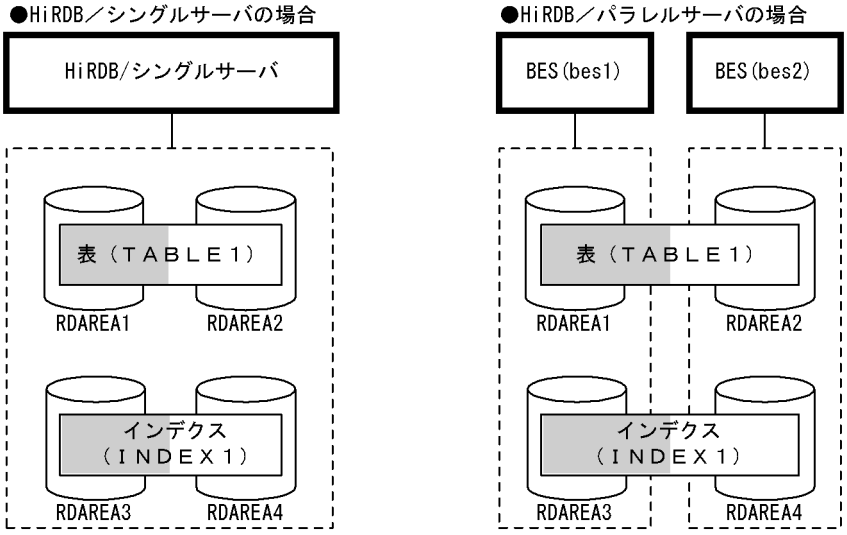
```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3.3　例題３（RD エリア単位で再編成する場合）

横分割表（TABLE1）を RD エリア単位で再編成します。再編成するときの条件は次のとおりです。

- TABLE1 はユーザ用 RD エリア（RDAREA1～RDAREA2）に横分割されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。INDEX1 はユーザ用 RD エリア（RDAREA3～RDAREA4）に横分割されています。
- 再編成するデータは RDAREA1 の表データ，及び RDAREA3 のインデクスデータです。
- 更新前ログ取得モード（省略値）で表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。
2. **pdrorg** コマンドの制御文ファイルを作成します。
3. **pdrorg** コマンドで表を再編成します。
4. 再編成対象 RD エリアのバックアップを取得します。
5. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- 更新前ログ取得モードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 4）にバックアップを取得する必要があります。
- 更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 1～4 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA3
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。

(3)　pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -t TABLE1 -r RDAREA1 C:¥pdrorg¥rorg01
```

〔説明〕

RDAREA3 に格納されている INDEX1 のインデクスデータも同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：再編成をするため rorg を指定します。

-t：再編成する表の名称を指定します。

-r：再編成対象の表（TABLE1）が格納されている RD エリアの名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1，RDAREA3）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

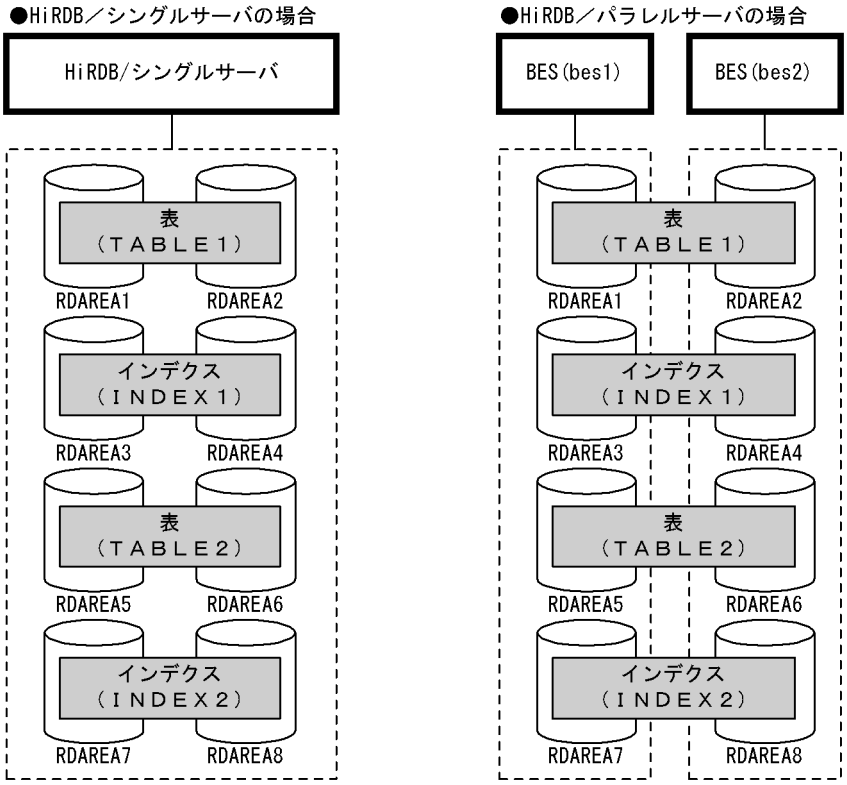
```
pdrels -r RDAREA1,RDAREA3
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3.4　例題４（スキーマ単位で再編成する場合）

認可識別子 USR01 のユーザが所有するすべての表を再編成します。再編成するときの条件は次のとおりです。

- 表を再編成するときにインデクスを一括作成（省略値）します。
- 更新前ログ取得モード（省略値）で表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. SQL で再編成対象 RD エリアを調べます。
2. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。
3. **pdrorg** コマンドの制御文ファイルを作成します。
4. **pdrorg** コマンドで表を再編成します。

5. 再編成対象 RD エリアのバックアップを取得します。
6. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- 更新前ログ取得モードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 5）にバックアップを取得する必要があります。
- 更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 2～5 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　SQL で再編成対象 RD エリアを調べます

```
SELECT DISTINCT(RDAREA_NAME) FROM MASTER.SQL_TABLES            1
     WHERE TABLE_SCHEMA=USR01 AND RDAREA_NAME IS NOT NULL;
SELECT DISTINCT(RDAREA_NAME) FROM MASTER.SQL_DIV_TABLE         2
     WHERE TABLE_SCHEMA=USR01;
SELECT DISTINCT(RDAREA_NAME) FROM MASTER.SQL_INDEXES           3
     WHERE TABLE_SCHEMA=USR01 AND RDAREA_NAME IS NOT NULL;
SELECT DISTINCT(RDAREA_NAME) FROM MASTER.SQL_DIV_INDEX         4
     WHERE TABLE_SCHEMA=USR01;
```

〔説明〕

1. 非横分割表を格納する RD エリアを検索します。
2. 横分割表を格納する RD エリアを検索します。
3. 非横分割インデクスを格納する RD エリアを検索します。
4. 横分割インデクスを格納する RD エリアを検索します。

(2)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,…
```

(3)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b) HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1                1
idxwork bes1 C:¥pdrorg¥idxwork               2
sort bes1 C:¥sortwork                        3
idxwork bes2 C:¥pdrorg¥idxwork               4
sort bes2 C:¥sortwork                        5
```

〔説明〕

1. アンロードデータファイルの名称を指定します。-g オプションが仮定されるため，アンロードデータファイルを１か所に作成します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
5. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(4) pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -t USR01.all C:¥pdrorg¥rorg01
```

〔説明〕

インデクス（INDEX1 及び INDEX2）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：再編成をするため rorg を指定します。

-t：再編成対象のスキーマの認可識別子を指定します。

C:¥pdrorg¥rorg01：(3)で作成した pdrorg コマンドの制御文ファイル名を指定します。

備考

スキーマ単位の再編成では-j オプション（LOB データがあるときの再編成）及び-g オプション（アンロードデータファイルの一元化）が仮定されています。

(5)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA8）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(6)　pdrels コマンドで RD エリアの閉塞を解除します

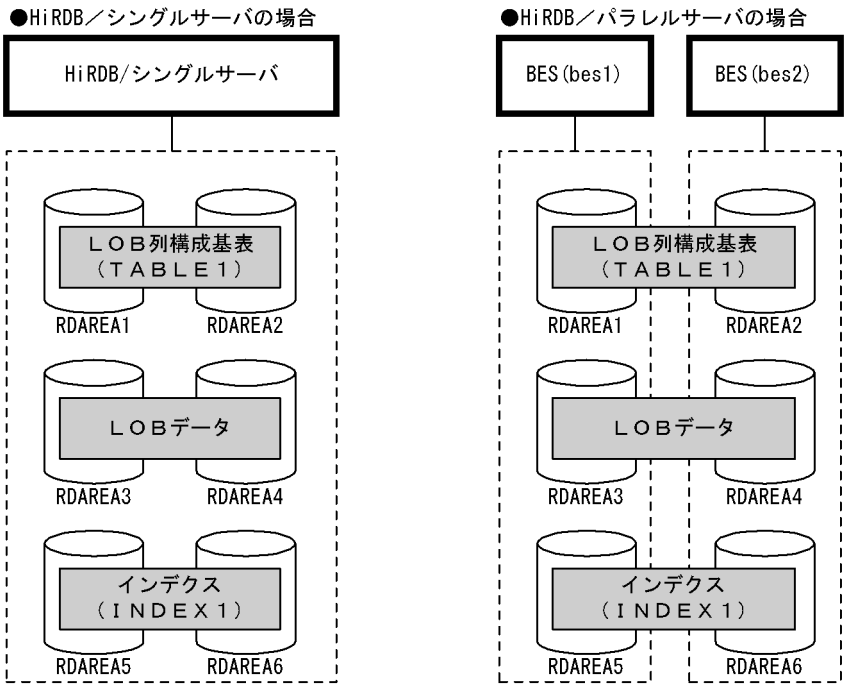
```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3.5　例題５（LOB 列が定義されている表を再編成する場合）

LOB 列が定義されている表（TABLE1）を表単位で再編成します。このとき，LOB データも一緒に再編成します。再編成するときの条件は次のとおりです。

- LOB 列構成基表は，ユーザ用 RD エリア（RDAREA1～RDAREA2）に横分割されています。
- LOB データは，ユーザ LOB 用 RD エリア（RDAREA3～RDAREA4）に横分割されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。INDEX1 は，ユーザ用 RD エリア（RDAREA5～RDAREA6）に横分割されています。
- 表を再編成するときにインデクスを一括作成（省略値）します。
- 更新前ログ取得モード（省略値）で表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。
2. **pdrorg** コマンドの制御文ファイルを作成します。
3. **pdrorg** コマンドで表を再編成します。
4. 再編成対象 RD エリアのバックアップを取得します。
5. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- 更新前ログ取得モードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 4）にバックアップを取得する必要があります。
- 更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 1～4 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(2)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
unload bes2:C:¥pdrorg¥unfile2               4
```

```
idxwork bes2 C:¥pdrorg¥idxwork                5
sort bes2 C:¥sortwork                         6
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. アンロードデータファイル（bes2 用）の名称を指定します。
5. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
6. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(3) pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -j -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

インデクス（INDEX1）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：再編成をするため rorg を指定します。

-j：LOB 列がある表を再編成する場合に指定します。

-t：再編成する表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4) 再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA6）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3.6　例題６（ディクショナリ表を再編成する場合）

データディクショナリ用 RD エリア（DATADIC）及びデータディクショナリ LOB 用 RD エリア（DATALOB）に格納されているディクショナリ表を再編成します。更新前ログ取得モード（省略値）でディクショナリ表を再編成します。

〈手順〉

1. **pdhold** コマンドでデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアを閉塞します。
2. データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアのバックアップを取得します。
3. **pdrorg** コマンドの制御文ファイルを作成します。
4. **pdrorg** コマンドでディクショナリ表を再編成します。
5. データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアのバックアップを取得します。
6. **pdrels** コマンドでデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 1～5 の間はデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドでデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアを閉塞します

RD エリアを閉塞した状態でバックアップを取得してください。バックアップを取得した後に，ほかのユーザが RD エリアの内容を更新しないように RD エリアの閉塞を解除しないでください。

```
pdhold -r DATADIC,DATALOB
```

(2)　データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアのバックアップを取得します

再編成中の異常終了に備えて，データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアのバックアップを取得してください。RD エリアの拡張に必要となるディクショナリ表の再編成中にデータディクショナリ用 RD エリア又はデータディクショナリ LOB 用 RD エリアに容量不足が発生すると，

RD エリアを拡張する対処が取れないため，RD エリアをバックアップから回復する必要があります。また，ディクショナリ表の再編成中にユニットがダウンすると，HiRDB 起動時にディクショナリ表を参照できないで，フロントエンドサーバが SUSPEND 状態となるため，RD エリアをバックアップから回復する必要があります。バックアップからの回復が必要かどうかの判断は，「注意事項」の条件を参照してください。

(3) pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

```
unload C:¥pdrorg¥unfile1                    1
lobunld C:¥pdrorg¥unfile2                   2
```

〔説明〕

1. データディクショナリ用 RD エリア用のアンロードデータファイルの名称を指定します。
2. データディクショナリ LOB 用 RD エリア用のアンロードデータファイルの名称を指定します。

(4) pdrorg コマンドでディクショナリ表を再編成します

```
pdrorg -k rorg -c dic C:¥pdrorg¥rorg01
```

〔説明〕

-k：再編成をするため rorg を指定します。

-c：ディクショナリ表を再編成する場合に指定します。

C:¥pdrorg¥rorg01：(3)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(5) データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアのバックアップを取得します

データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリア（DATADIC 及び DATALOB）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(6) pdrels コマンドでデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアの閉塞を解除します

```
pdrels -r DATADIC,DATALOB
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(7)　注意事項

ディクショナリ表の再編成時，次に示すエラーが発生した場合は注意が必要です。

- **アンロードは完了しましたが，リロード中にデータディクショナリ用 RD エリアの容量不足が発生しました。**

この場合，次に示す条件に応じて対処してください。

条件１：

データベース再編成ユティリティを更新ログ取得モード，又は更新前ログ取得モードで実行していて，SQL_PHYSICAL_FILES 表及び SQL_RDAREAS 表のリロードが完了しているとき

1. データベース構成変更ユティリティでデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアを拡張又は追加します。
2. オプションや制御文の指定を変更しないで，データベース再編成ユティリティを再度実行します。

条件２：

データベース再編成ユティリティを更新ログ取得モード，又は更新前ログ取得モードで実行していて，SQL_PHYSICAL_FILES 表及び SQL_RDAREAS 表のリロードが完了していないとき，又は，データベース再編成ユティリティをログレスモードで実行しているとき

1. 再編成前に取得したバックアップを使用して，データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアをデータベース回復ユティリティで回復します。
2. データベース構成変更ユティリティでデータディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアを拡張又は追加します。
3. ディクショナリ表の再編成を再度実施します。

   エラー発生時（RD エリアの拡張又は追加前）に作成したアンロードデータファイルを使用してリロード処理だけをしないでください。

- **リロード中にユニットがダウンし，HiRDB を再開始しましたが，ディクショナリ用 RD エリアの障害と認識してフロントエンドサーバが SUSPEND 状態となりました（HiRDB/パラレルサーバの場合）。**

この場合，次のように対処してください。

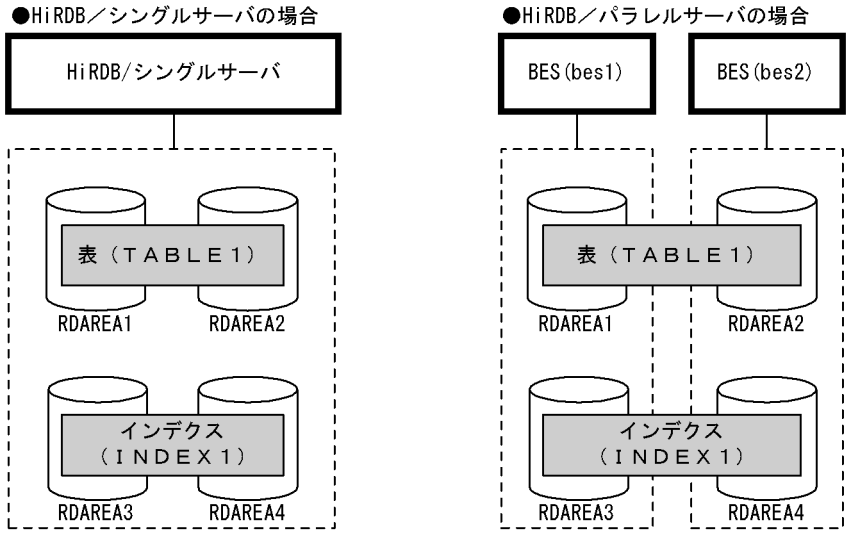
1. 再編成前に取得したバックアップを使用して，データディクショナリ用 RD エリア及びデータディクショナリ LOB 用 RD エリアをデータベース回復ユティリティで回復します。
2. pdstart -a コマンドで HiRDB を開始します。

   詳細については，「データディクショナリ用 RD エリアに障害が発生して，フロントエンドサーバが SUSPEND 状態のときの開始方法（pdstart -a）」を参照してください。

13.3.7　例題７（ログレスモードで再編成する場合）

横分割表（TABLE1）を表単位で再編成します。再編成するときの条件は次のとおりです。

- TABLE1 はユーザ用 RD エリア（RDAREA1～RDAREA2）に横分割されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。INDEX1 はユーザ用 RD エリア（RDAREA3～RDAREA4）に横分割されています。
- 表を再編成するときにインデクスを一括作成（省略値）します。
- ログレスモードで表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。
2. 再編成対象 RD エリアのバックアップを取得します。
3. **pdrorg** コマンドの制御文ファイルを作成します。
4. **pdrorg** コマンドで表を再編成します。
5. 再編成対象 RD エリアのバックアップを取得します。
6. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- ログレスモードで pdrorg コマンドを実行するため，pdrorg コマンドの実行前にバックアップを取得する必要があります。pdrorg コマンドが異常終了した場合，このバックアップを使用して RD エリアを回復します。ただし，ある条件下ではバックアップを取得する必要がありません。その条件については，「再編成の前にバックアップを取得しないでよい場合」を参照してください。
- ログレスモードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 5）にバックアップを取得する必要があります。

- ログレスモードで pdrorg コマンドを実行するため，前記の手順 1～5 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

(2)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA4）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(3)　pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
unload bes2:C:¥pdrorg¥unfile2               4
idxwork bes2 C:¥pdrorg¥idxwork              5
sort bes2 C:¥sortwork                       6
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。

4. アンロードデータファイル（bes2 用）の名称を指定します。
5. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
6. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(4)　pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -t TABLE1 -l n C:¥pdrorg¥rorg01
```

〔説明〕

インデクス（INDEX1）も同時に再作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：再編成をするため rorg を指定します。

-t：再編成する表の名称を指定します。

-l：ログ取得方式に n（ログレスモード）を指定します。

C:¥pdrorg¥rorg01：(3)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(5)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA4）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(6)　pdrels コマンドで RD エリアの閉塞を解除します

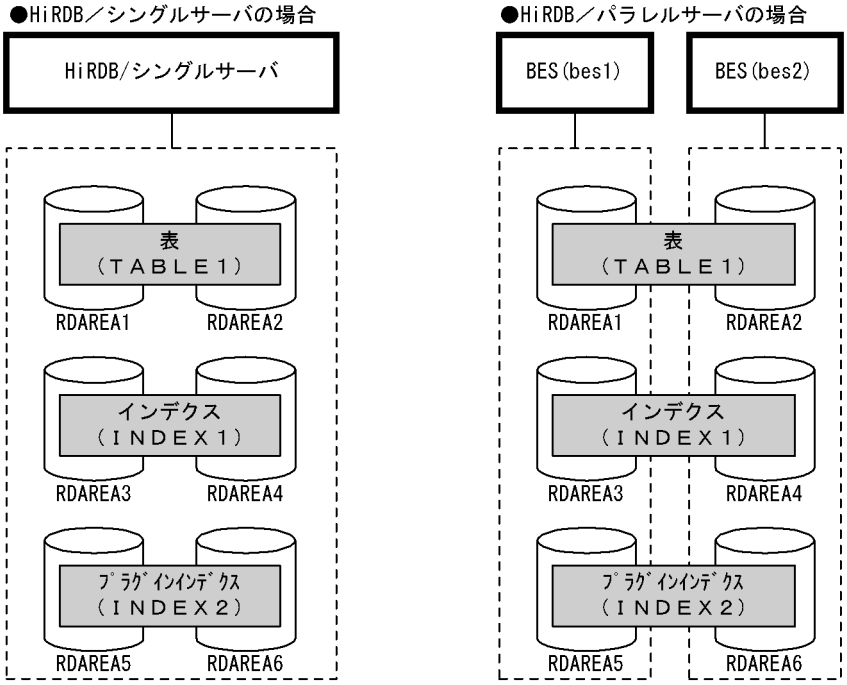
```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.3.8　例題８（抽象データ型を定義した表を再編成する場合）

LOB 属性がない抽象データ型（GEOMETRY 型，又は XML 型）を定義した表（TABLE1）を再編成します。LOB 属性がある抽象データ型（SGMLTEXT 型）の再編成方法については，「例題５（LOB 列が定義されている表を再編成する場合）」を参照してください。再編成するときの条件は次のとおりです。

- TABLE1 はユーザ用 RD エリア（RDAREA1～RDAREA2）に横分割されています。
- 更新前ログ取得モード（省略値）で表を再編成します。



注　網掛け部分のデータが再編成対象となります。

〈手順〉

1. **pdhold** コマンドで再編成対象 RD エリアを閉塞します。
2. **pdrorg** コマンドの制御文ファイルを作成します。
3. **pdrorg** コマンドで表を再編成します。
4. 再編成対象 RD エリアのバックアップを取得します。
5. **pdrels** コマンドで RD エリアの閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

ポイント

- 更新前ログ取得モードで pdrorg コマンドを実行するため，pdrorg コマンドの実行後（手順 4）にバックアップを取得する必要があります。
- 更新前ログ取得モードで pdrorg コマンドを実行するため，前記の手順 1～4 の間は再編成対象 RD エリアを閉塞したままにしてください。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

(2) pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a) HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                    1
idxwork C:¥pdrorg¥idxwork                   2
sort C:¥sortwork                            3
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. インデクス情報ファイル作成用のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリの名称を指定します。

(b) HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1               1
idxwork bes1 C:¥pdrorg¥idxwork              2
sort bes1 C:¥sortwork                       3
unload bes2:C:¥pdrorg¥unfile2               4
idxwork bes2 C:¥pdrorg¥idxwork              5
sort bes2 C:¥sortwork                       6
```

〔説明〕

1. アンロードデータファイル（bes1 用）の名称を指定します。
2. インデクス情報ファイル作成用（bes1 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
3. ソート用ワークディレクトリ（bes1 用）の名称を指定します。
4. アンロードデータファイル（bes2 用）の名称を指定します。
5. インデクス情報ファイル作成用（bes2 用）のディレクトリ名を指定します。このディレクトリ下にインデクス情報ファイルが作成されます。
6. ソート用ワークディレクトリ（bes2 用）の名称を指定します。

(3) pdrorg コマンドで表を再編成します

```
pdrorg -k rorg -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

-k：再編成をするため rorg を指定します。

-t：再編成する表の名称を指定します。

C:¥pdrorg¥rorg01：(2)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(4)　再編成対象 RD エリアのバックアップを取得します

再編成対象 RD エリア（RDAREA1～RDAREA6）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5,RDAREA6
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.4　表の再編成時期を予測する方法（再編成時期予測機能）

実行者　DBA 権限保持者

ここでは，表（インデクスを含む）の再編成時期，又は RD エリアの拡張時期を予測する方法について説明します。説明する項目は次のとおりです。

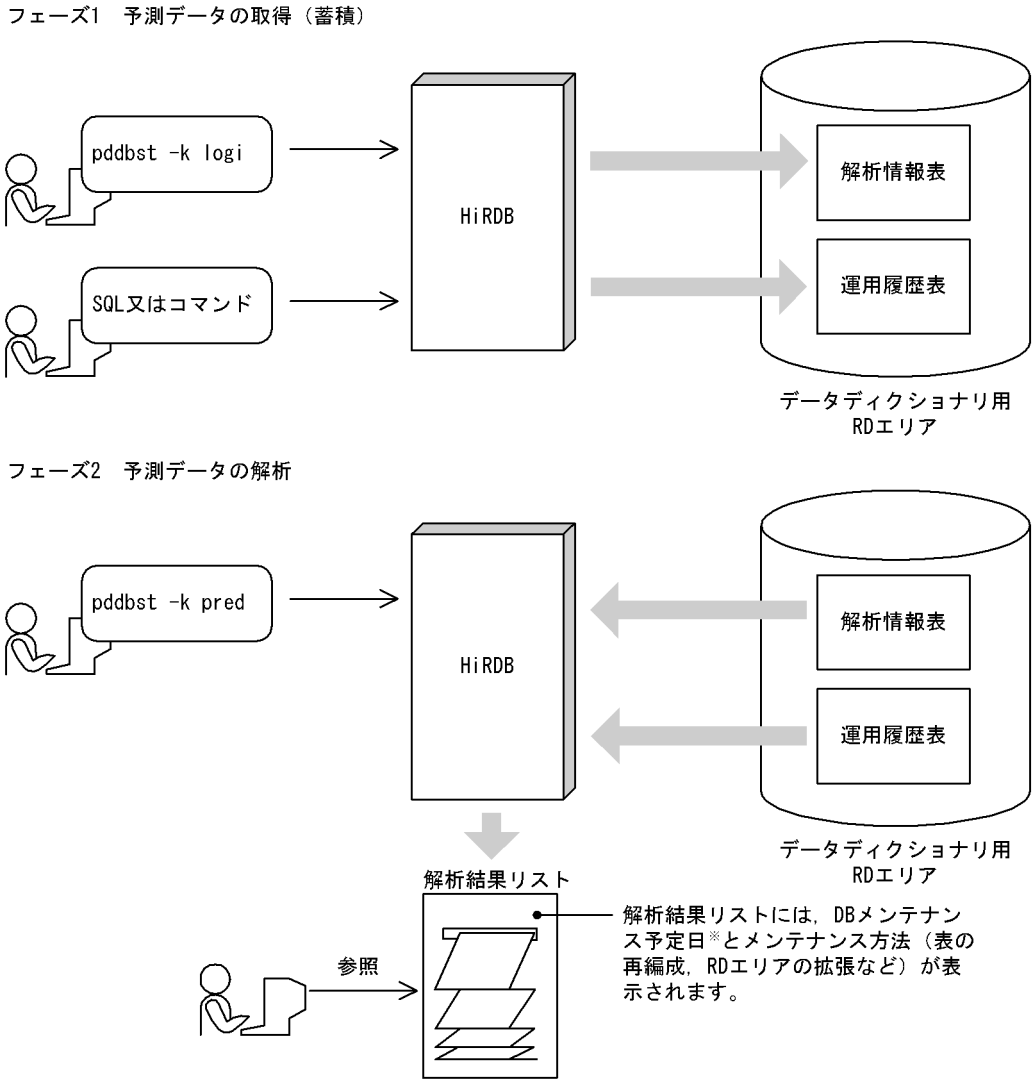
- 再編成時期を予測するには
- 再編成時期予測機能を使用するための準備
- 運用の流れ
- 再編成時期予測機能使用時の注意事項
- 再編成時期の予測をやめる場合
- 再編成時期予測のカスタマイズ

13.4.1　再編成時期を予測するには

表やインデクスの再編成をするかどうか，又は RD エリアを拡張するかどうかは，出力されたメッセージや，pddbst コマンドの実行結果から，再編成する表はどれか，いつ再編成する必要があるかなど，ユーザが総合的に判断する必要がありました。そのため，再編成する必要がない表を再編成したり，出力されたメッセージを見逃したため，再編成する必要がある表を再編成しなかったりするおそれがありました。

これらの運用を簡単にするために HiRDB が再編成時期の予測を行うようにしました。これを**再編成時期予測機能**といいます。再編成時期予測機能の概要を次の図に示します。

図 13–7　再編成時期予測機能の概要



注※

RD エリアのメンテナンスが必要になる予定日を **DB メンテナンス予定日**といいます。HiRDB が行う予測データの解析方法の概要については，「HiRDB が行う予測データの解析方法」を参照してください。

再編成時期の予測は，次に示す二つのフェーズに分かれています。

- **フェーズ 1　再編成時期の予測データの取得**
  - pddbst コマンドを定期的に実行し，**解析情報表**にデータベースの解析結果を蓄積していきます。
  - 次の SQL 又はコマンドが実行されると，**運用履歴表**にデータベースの運用履歴情報が出力されます。
    - 定義系 SQL（DROP SCHEAM，DROP TABLE，DROP INDEX，ALTER TABLE）
    - 操作系 SQL（PURGE TABLE）
    - pdrorg コマンド
    - pdreclaim コマンド
    - pdload コマンド

・pdmod コマンド

- **フェーズ 2　再編成時期の予測データの解析**

  解析情報表及び運用履歴表を入力情報にして，pddbst コマンドで再編成時期の予測データを解析します。ユーザは pddbst コマンドの実行結果を参照し，必要に応じて次に示すどれかの操作を行います。なお，RD エリアの自動増分は，RD エリア作成時に自動増分を適用していれば，HiRDB が自動的に行います（ユーザが対処する必要はありません）。

  - pdrorg コマンドによる表又はインデクスの再編成
  - pdreclaim コマンドによる使用中空きページの解放，及び使用中空きセグメントの解放
  - pdmod コマンドによる RD エリアの拡張
  - RD エリアの自動増分
  - pdmod コマンドによる RD エリアの再初期化

参考

- 再編成時期予測機能には，予測レベル 1 と予測レベル 2 があります。予測レベル 1 は，主に RD エリアの容量不足を監視する場合に使用します。予測レベル 2 は，RD エリアの容量不足のほかに，データ格納効率の劣化によるオンライン性能への影響を監視する場合に使用します。
- 再編成時期予測機能は RD エリアの容量不足を発生させないことを一番の目的としているため，再編成後のデータ格納状態が最大となることを前提にして予測しています。例えば，分岐行（複数ページに分割して格納されているデータ）は再編成後も分岐行になると判断します。したがって，表を再編成するか，又は RD エリアを拡張するかを判断する場合，RD エリアの拡張を予測しやすい傾向があります。

13.4.2　再編成時期予測機能を使用するための準備

再編成時期予測機能を使用するには次に示す準備が必要です。

(1)　解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアを作成する

解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアの容量見積もりをした後に，データベース構成変更ユティリティ（pdmod コマンド）でデータディクショナリ用 RD エリアを作成してください。

解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

参考

解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアの容量が不足した場合，HiRDB が再編成時期を予測できなくなります（容量不足が原因で SQL やコマンドがエラーとなることはありません）。

なお，表を追加したため，当初の見積もりで不足する場合は，解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアを拡張してください。

(2) pd_rorg_predict オペランドを指定する

pd_rorg_predict オペランドに Y を指定してください。

(3) システムログファイルの容量見積もりをする

解析情報表又は運用履歴表が更新されるとシステムログが出力されるため，システムログファイルの容量見積もりを再度行ってください。システムログファイルの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

参考

HiRDB/パラレルサーバの場合は，ディクショナリサーバのシステムログファイルに出力されます。

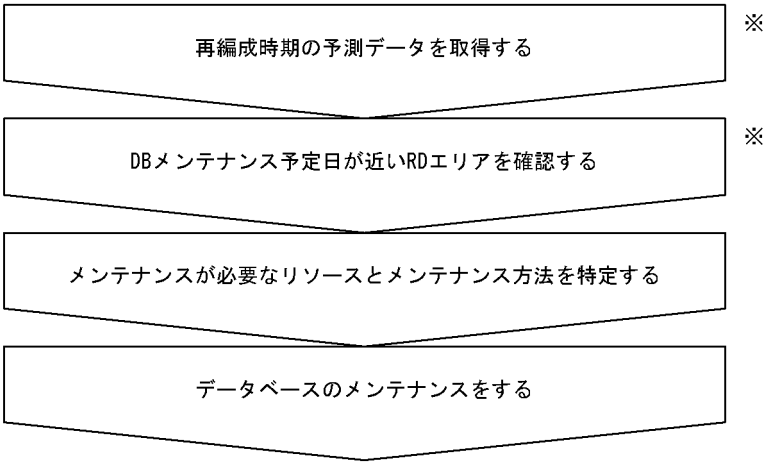
13.4.3 運用の流れ

ここでは，再編成時期を予測する運用の流れについて説明します。予測レベル 1 の場合は(1)を，予測レベル 2 の場合は(1)及び(2)を参照してください。

(1) 予測レベル 1 の場合

表の再編成時期を予測するときの運用の流れを次の図に示します。

図 13–8　表の再編成時期を予測するときの運用の流れ



注※　この作業は毎日行うことをお勧めします。

(a)　再編成時期の予測データを取得する

次に示すコマンドを実行して再編成時期の予測データを解析情報表に取得します。このコマンドを毎日実行して再編成時期の予測データを蓄積してください。

```
pddbst -k logi -r ALL -e 1
```

注意事項

(a)の作業を 4 回以上行ってください。4 件以上の予測データが蓄積されないと正しい予測ができません（KFPK10312-I，又は KFPK10313-I メッセージが出力されます）。なお，RD エリアの状態が変わらないと，何回実行しても 1 回実行したとみなされます。詳細については，「予測データを正しく取得していない場合」を参照してください。

参考

データの格納状態が大きく変化した後は，再編成が必要な状態になる可能性が高いため，大量更新をした後は(a)と(b)の作業をすることをお勧めします。

(b)　DB メンテナンス予定日が近い RD エリアを確認する

次に示すコマンドを実行して，DB メンテナンス予定日が近い RD エリアを確認します。このコマンドを毎日実行して確認してください。

```
pddbst -k pred -r ALL -e 1
```

DB メンテナンス予定日が近い RD エリアの一覧が表示されます。出力結果の例を次に示します。

出力結果の例

```
 pddbst 07-02       ***** Rdarea resource forecast *****     2005/05/15 11:07:10
No         Date        RdareaName                          ResKind
---------- ----------- ------------------------------- ----------
         1 2005/05/31  "RDUSER01"                         Segment
         2 2005/06/01  "RDUSER02"                         Segment
---------- ----------- ------------------------------- ----------
```

〔説明〕

Date：DB メンテナンス予定日が表示されます。

RdareaName：メンテナンスが必要な RD エリアが表示されます。

注意事項

DB メンテナンス予定日はあくまで予測であって，運用方法によっては DB メンテナンス予定日より前に RD エリアの容量不足が発生することがあります。

参考

- (c)の作業は pddbst コマンドの実行時間が長くなることがあるため，オンライン業務に影響を与えたくない場合は，(b)の作業でメンテナンスの必要がある RD エリアを特定しておいて，オンライン業務に影響を及ぼさない時間帯に(c)の作業をしてください。
- (c)の作業の実行時間が問題にならない場合，(b)の作業を省略し，(c)の作業だけ実行してもかまいません。

(c)　メンテナンスが必要なリソースとメンテナンス方法を特定する

(b)の結果，DB メンテナンス予定日が近い RD エリアがある場合，次に示すコマンドを実行し，メンテナンスが必要な表，インデクス，及び RD エリアを調査します。

```
pddbst -k pred -r ALL -e 1 -m
```

メンテナンスをする表，インデクス，及び RD エリアの一覧とメンテナンス方法が表示されます。出力結果の例を次に示します。

出力結果の例

```
 pddbst 07-02       ***** Rdarea resource forecast *****     2005/05/15 11:07:10
No         Date        RdareaName                          ResKind
---------- ----------- ------------------------------- ----------
         1 2005/05/31  "RDUSER01"                         Segment
         2 2005/06/01  "RDUSER02"                         Segment
---------- ----------- ------------------------------- ----------

 pddbst 07-02       ***** Maintenance Information *****      2005/05/15 11:07:10
No          : 1
Rdarea Name : "RDUSER01"
Method      :        ...[1]
```

```
Segment     : 42560
--------------------------------------------------------------------------------
 Reclaim      Reorganize   Type Name                                      Date
------------ ------------ ---- ----------------------------------------- ----------
      28089 *       9363    T   "k1234567"."table01"   ...[2]             2005/05/31
          0        12342 *  T   "k1234567"."table02"   ...[3]             2005/05/31
        851         7235 *  I   "k1234567"."index01"   ...[4]             2005/05/31
          3          -10    T   "k1234567"."table10"   ...[5]             2005/05/31
================================================================================
No          : 2
Rdarea Name : "RDUSER02"
Method      : Expand   ...[6]
Segment     : 1523
--------------------------------------------------------------------------------
 Reclaim      Reorganize   Type Name                                      Date
------------ ------------ ---- ----------------------------------------- ----------
        228         1035    T   "k1234567"."table01"                     2005/06/03
          0          121    T   "k1234567"."table03"                     2005/06/03
         30           60    I   "k1234567"."index07"                     2005/06/03
================================================================================
```

〔説明〕

1. Method の欄に何も表示されていないため，RD エリアのメンテナンスは必要ありません。
2. RD エリア（RDUSER01）に格納されている表（table01）は，Reclaim の欄に*が表示されているため，pdreclaim コマンドで使用中空きページの解放，及び使用中空きセグメントの解放をしてください。
3. RD エリア（RDUSER01）に格納されている表（table02）は，Reorganize の欄に*が表示されているため，pdrorg コマンドで表の再編成をしてください。
4. RD エリア（RDUSER01）に格納されているインデクス（index01）は，Reorganize の欄に*が表示されているため，pdrorg コマンドでインデクスの再編成をしてください。
5. RD エリア（RDUSER01）に格納されている表（table10）は，*が表示されていないため，メンテナンスの必要はありません。
6. Method の欄に RD エリアのメンテナンス方法が表示されています。

   Expand：RD エリアを拡張してください。拡張する必要があるセグメント数は，Segment で表示されている 1523 セグメントです。

   Extend：HiRDB が RD エリアの自動増分をします。

   Reinit：DB の再作成（RD エリア内のデータをアンロードし，RD エリアを再初期化した後，アンロードデータをリロード）をしてください。

   表示なし：RD エリアのメンテナンスは必要ありません。

参考

- そのほかのヘッダの意味を次に示します。

  Type：リソースの種別が表示されます。

  T：表

I：インデクス

L：LOB 用 RD エリア

Name：表名又はインデクス名が表示されます。

Date：DB メンテナンス予定日が表示されます。

- 表示しているセグメント数は，メンテナンス予定日までに拡張が必要な最低限のセグメント数です。表示された値以上の拡張を実施してください。なお，RD エリアの拡張回数には上限があるため，注意してください。

(d)　データベースのメンテナンスをする

HiRDB が出した指示に従い，次に示すどれかの処置をしてください。

- 表を再編成する
- インデクスを再編成する
- RD エリアを拡張する
- RD エリアを再初期化する

注意事項

(d)の作業中に(a)の作業は行わないでください。(d)の作業中に(a)の作業を行うと，正しい予測ができなくなります。

(2)　予測レベル 2 の場合

予測レベル 1 の再編成時期の予測データの取得では，RD エリア単位の情報だけ取得するため，ディレクトリページだけ解析対象となります。しかし，予測レベル 2 では，更に表単位，及びインデクス単位の情報も取得するため，データページ，及びインデクスページの解析も必要となります。そのため，予測レベル 2 は，予測レベル 1 に比べて予測データを取得する pddbst の実行時間が長くなります。

オンライン業務中に，予測レベル 2 の再編成時期の予測データを取得する場合，業務 AP へ影響を与える可能性もあります。オンライン中の業務 AP への影響を最小限にしたい場合は，次の実行方法を選択できます（二つ同時にも指定できます）。

- **インターバル解析**
- **マージ解析**

インターバル解析，及びマージ解析については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.4.4　再編成時期予測機能使用時の注意事項

(1)　予測対象外のリソース

次に示すユーザ LOB 用 RD エリアは再編成時期の予測の対象外になります。

- プラグインインデクスを格納しているユーザ LOB 用 RD エリア

(2)　正しい予測ができないケース

次に示す場合は，再編成時期の正しい予測ができません。

(a)　予測データを正しく取得していない場合

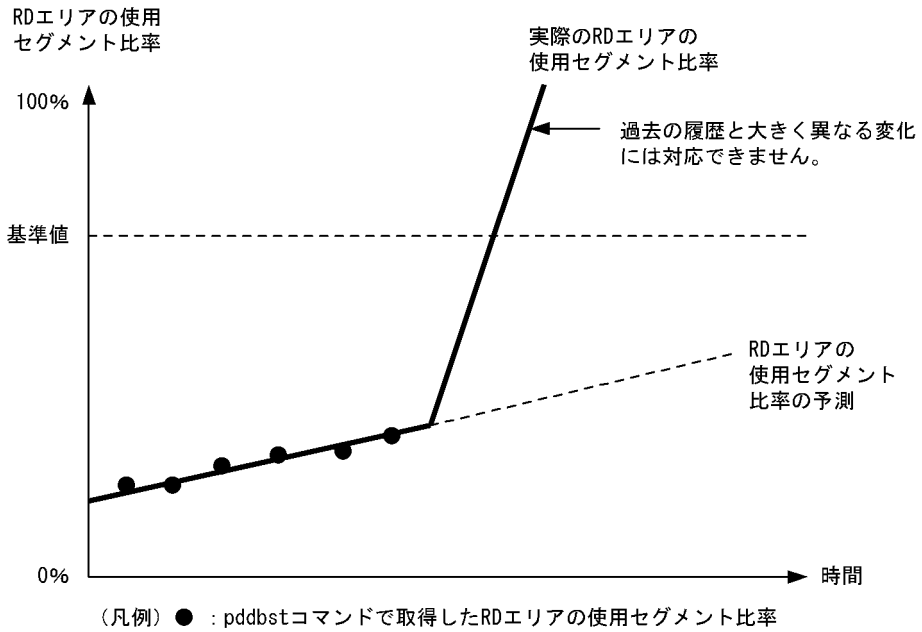
データの格納状態に変化がある予測データが 4 件以上蓄積されないと，再編成時期予測機能は実行できません。また，予測データを定期的に取得しないと，正しい予測ができません。例えば，週末に更新を行うシステムで，週の始めの 4 日間だけ予測データを取得しても，正しい予測ができません。

> **参考**
>
> 予測データを定期的に取得するほど，予測の精度が上がります。

(b)　データの格納状態が急激に変化するような運用をする場合

RD エリアの使用セグメント比率の変移（履歴）から今後の使用状況を予測するため，データの格納状態が急激に変化するような運用をする場合は正しい予測ができません。例を次の図に示します。

図 13–9　データの格納状態が急激に変化するため正しい予測ができない例



注　基準値については，「再編成時期予測のカスタマイズ」を参照してください。

(c)　最適なページサイズの RD エリアに表を格納していない場合

表の行長に合わせた最適なページサイズの RD エリアに表を格納していない場合，再編成時期の正しい予測ができません。例えば，再編成しても効果のない表に対して再編成を指示したり，再編成の直後に再編成の指示を出したりすることがあります。

(d)　キー値に偏りがあるリバランス表の場合

リバランス表を定義する場合は，データが均等に配置されるハッシュ関数や分割キーを選択してください。キー値に偏りがある場合，特定のハッシュグループ（セグメント）にデータが格納されるため，正しい予測ができません。

(3)　予測の傾向

次に示す条件をすべて満たす表は，RD エリアの拡張を行う予測が出やすくなります。

- 表に可変長文字列（VARCHAR，NVARCHAR，又は MVARCHAR）が定義されている
- 上記の列にノースプリットオプションが定義されていない
- pdload コマンドを使用したデータロード以外の方法でデータを格納した

(4)　ユーザが定義した抽象データ型がある場合

ユーザが定義した抽象データ型（プラグインが提供する抽象データ型でない場合）が定義されている表は，pdrorg コマンドによる再編成及び pdreclaim コマンドによる使用中空きページの解放ができないため，再編成の指示は行いません。RD エリア内に該当表しかない場合，必ず RD エリアの拡張又は再初期化の指示になります。

(5)　予測時の制限

実長が 255 バイト以下の可変長文字列（VARCHAR，NVARCHAR，又は MVARCHAR）が，別のページに分岐した状態の場合（256 バイト以上のデータが 255 バイト以下に更新されたケースが該当する），再編成すると行分岐を解消し，格納効率及びアクセス効率を上げることができますが，データ長の情報までは取得しないため，これを予測に反映できません。

また，NO SPLIT を指定していない可変長文字列のデータ型がある表の場合，ページ長よりも実長が短いデータが多量にあると，再編成直後でもページ使用率が低い状態のページが多量に存在する状態となります。そのため，基準値定義ファイルによるカスタマイズが必要となります。

(6)　解析情報表及び運用履歴表の再編成

ディクショナリ表の再編成をする場合，解析情報表及び運用履歴表も再編成の対象になります。ほかのディクショナリ表とまとめて再編成してください。解析情報表及び運用履歴表だけを再編成する必要はありません。

参考

表やインデクスを大量に削除すると，両表のアクセス効率を向上するために定義しているインデクスに無効領域が発生します。この場合，表やインデクスが定義されたときに再利用されるため，データディクショナリ用 RD エリアの容量が十分なときは再編成をする必要はありません。

(7)　予測精度

予測レベル 1 の場合，RD エリア内にリソースが一つだけあるときは（例えば，1RD エリアに 1 表だけ格納されているなど），解析情報表及び運用履歴表の情報から予測します。RD エリア内にリソースが複数ある場合，解析情報表の情報から予測します。したがって，後者よりは前者の方が，予測精度は上がります。

(8)　データベースの回復と再編成時期予測との関係

再編成時期の予測は，過去のデータベースの履歴を基に解析します。そのため，障害発生時にデータベースを最新の状態に回復しない運用（例えば，バックアップ取得時点に回復するなど）を行っている場合は，正しい予測ができません。このような運用の場合，予測の基になる情報をリセットすれば正しい予測ができるようになります。これを，**状態解析結果蓄積情報のリセット**といいます。状態解析結果蓄積情報のリセットについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.4.5　再編成時期の予測をやめる場合

再編成時期の予測をやめる場合は，pd_rorg_predict オペランドの指定値を N に変更してください。再度，再編成時期の予測を開始する場合はこのオペランドの指定値を Y に戻してください。

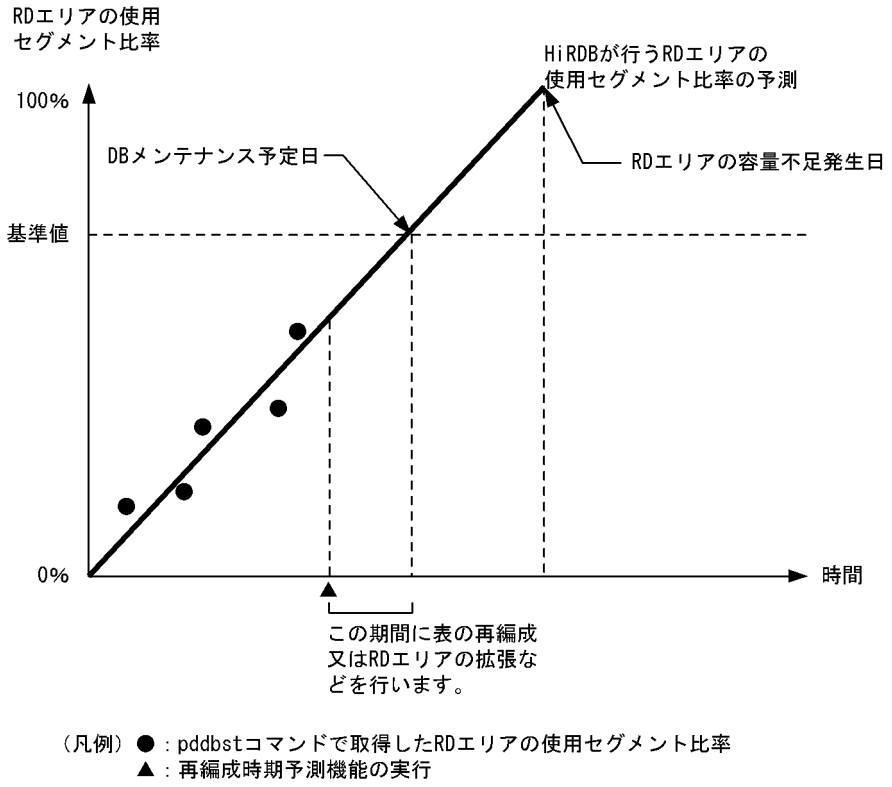
なお，再編成時期の予測を完全にやめて，解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアをユーザ用 RD エリアとして使用する場合は，解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリアを pdmod コマンドで削除した後に（pddbst コマンド実行中には削除しないでください），pd_rorg_predict オペランドの指定値を N に変更するか，又はこのオペランドを削除してください。

13.4.6　再編成時期予測のカスタマイズ

(1)　HiRDB が行う予測データの解析方法

RD エリアの使用セグメント比率の履歴から，HiRDB は DB メンテナンス予定日を予測しています。HiRDB が行う予測データの解析方法の概要を次の図に示します。

図 13–10　HiRDB が行う予測データの解析方法の概要



〔説明〕

- RD エリアの使用セグメント比率が**基準値**（省略値は 80%）を超えると予測される日を DB メンテナンス予定日としています。
- 基準値は基準値定義ファイルで変更できます。基準値定義ファイルについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2)　基準値のカスタマイズ

例えば，定期的に初期データロードを行う運用のため，DB メンテナンス予測日前に格納状態が変化するなど，基準値が現在の運用方法に合わない場合，基準値を変更して HiRDB が行う再編成時期の予測をカスタマイズする必要があります。

13.5　表のデータを削除する方法

13.5.1　表のデータを削除する方法の説明

実行者　表の所有者又は DELETE 権限があるユーザ

表から不要になった行を削除するには，**DELETE 文**又は **PURGE TABLE 文**を実行します。

ポイント

- ビュー表の行を削除すると，ビュー表に対応する実表の行が削除されます。
- 改竄防止表の場合，行削除禁止期間を過ぎている行だけを削除できます。

(1)　行を選択して削除する場合

行を選択して削除する場合は，DELETE 文に **WHERE** オペランドを指定して実行します。

(2)　すべての行を削除する場合

表にあるすべての行を削除する場合は，次に示すどちらかの SQL を実行します。

- WHERE オペランドを省略した DELETE 文
- PURGE TABLE 文

考慮点

- 行数が多い表からすべての行を削除する場合は，PURGE TABLE 文を実行した方が，DELETE 文を実行するよりも速く削除できます。
- DELETE 文で大量の行を削除する場合は，**LOCK 文**で表に排他モードの **EXCLUSIVE** を指定してから実行すると，排他制御のオーバヘッドが少なくなります。
- DELETE 文で大量に行を削除しても，セグメントは確保されたままになるため，ユーザ用 RD エリア内の空き領域は変わりません。空き領域を作りたい場合は，PURGE TABLE 文を使用してください。

13.6　列を追加する方法

実行者　HiRDB 管理者及び表の所有者

表に列を追加する場合は，**ALTER TABLE** に **ADD** オペランドを指定して実行します。

13.6.1　列を追加する前に

(1)　FIX 表に列を追加する場合の留意事項

データが格納されている FIX 表には列を追加できません。したがって，データが格納されている FIX 表に列を追加する場合は，次に示す手順で列を追加します。

〈手順〉

1. **pdrorg** コマンドで表データをアンロードします。
2. **PURGE TABLE** で表データを削除します。
3. **ALTER TABLE** で列を追加します。
4. **pdhold** コマンドで表格納 RD エリアを閉塞します。
5. **pdload** コマンドで表データをデータロードします。
6. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
7. **pdcopy** コマンドでバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。
8. **pdrels** コマンドで RD エリアの閉塞を解除します。

留意事項

1. 手順 1 で表データをアンロードするときに，DAT 形式でアンロードする方法とバイナリ形式でアンロードする方法があります。DAT 形式でアンロードした方が操作が簡単なため，基本的には DAT 形式でアンロードしてください。
2. DAT 形式ファイルに変換できない文字データ（0x00，0x0a）が表中にある場合，その表データを DAT 形式でアンロードできません。この場合は，バイナリ形式でアンロードしてください。
3. DAT 形式でアンロードしたときの方法については，「例題 6（FIX 表に列を追加する場合：DAT 形式でアンロードする場合）」を参照してください。
4. バイナリ形式でアンロードしたときの方法については，「例題 7（FIX 表に列を追加する場合：バイナリ形式でアンロードする場合）」を参照してください。

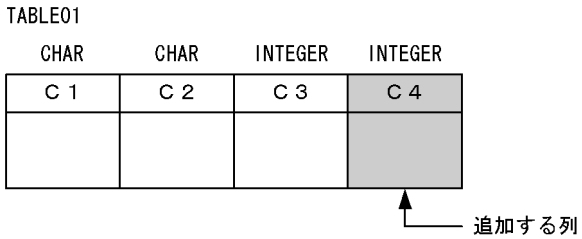
(2)　注意事項

- 改竄防止表には列を追加できません。

- 表に列を追加すると，その表を使用しているストアドルーチンが無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。
- トリガ SQL 文中に指定している表に NOT NULL 列を追加すると，トリガが無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
- 表に列を追加した後，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.6.2 例題 1（非 FIX 表に列を追加する場合）

非 FIX 表（TABLE01）に列（C4）を追加します。



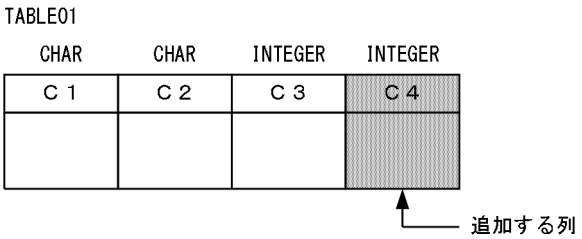
(1) ALTER TABLE で C4 列を追加します

```
ALTER TABLE TABLE01 ADD C4 INTEGER;
```

この場合，ALTER TABLE による列追加前に格納済みの行の C4 列にはナル値が格納されます。

13.6.3 例題 2（データが格納されている非 FIX 表に列を追加し，追加した列に値を格納する場合）

非 FIX 表（TABLE01）に列（C4）を追加します。



(1) ALTER TABLE で C4 列を追加します

- 列にナル値を許す場合

```
ALTER TABLE TABLE01 ADD C4 INTEGER DEFAULT 10 ON ROW EXISTS;
```

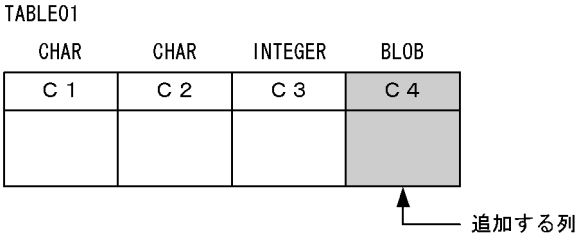
- 列にナル値を許さない場合

```
ALTER TABLE TABLE01 ADD C4 INTEGER DEFAULT 10 ON ROW EXISTS NOT NULL;
```

この場合，ALTER TABLE による列追加前に格納済みの行の C4 列には 10 が格納されます。

13.6.4　例題 3（LOB 列を追加する場合）

表（TABLE01）に BLOB 型の列（C4）を追加します。C4 列のデータはユーザ LOB 用 RD エリア（ULOB01，ULOB02）に格納します。



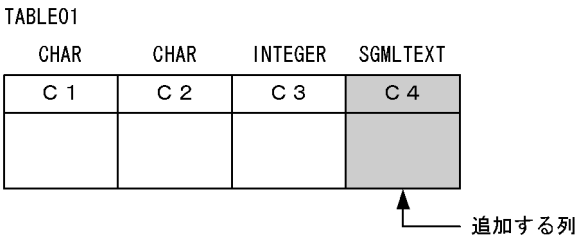
(1)　ALTER TABLE で C4 列を追加します

```
ALTER TABLE TABLE01 ADD C4 BLOB(1M) IN ((ULOB1),(ULOB2));
```

この場合，ALTER TABLE による列追加前に格納済みの行の C4 列にはナル値が格納されます。

13.6.5　例題 4（抽象データ型の列を追加する場合）

表（TABLE01）に SGMLTEXT 型の列（C4）を追加します。C4 列のデータはユーザ LOB 用 RD エリア（ULOB01，ULOB02）に格納します。



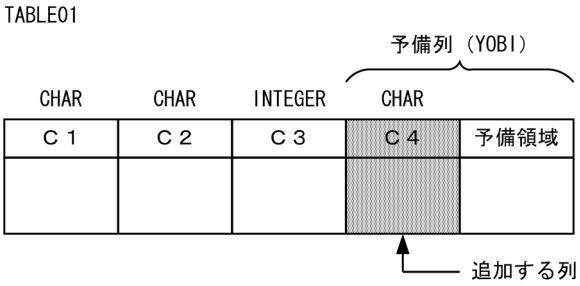
(1)　ALTER TABLE で C4 列を追加します

```
ALTER TABLE TABLE01 ADD C4 SGMLTEXT
    ALLOCATE(SGMLTEXT IN ((ULOB1),(ULOB2)))
    PLUGIN プラグインオプション;
```

この場合，ALTER TABLE による列追加前に格納済みの行の C4 列にはナル値が格納されます。

13.6.6　例題 5（FIX 表に列を追加する場合：予備列を定義している場合）

予備列を定義している FIX 表（TABLE01）に列（C4）を追加します。



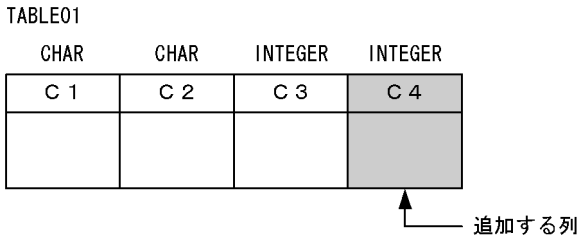
(1)　ALTER TABLE で C4 列を追加します

```
ALTER TABLE TABLE01 ADD C4 CHAR(3) INTO YOBI;
```

この場合，ALTER TABLE による列追加前に格納済みの行の C4 列には定義長（3 バイト）分のナル文字（0x00）が格納されます。

13.6.7　例題 6（FIX 表に列を追加する場合：DAT 形式でアンロードする場合）

FIX 表（TABLE01）に列（C4）を追加します。



(1)　pdhold コマンドで TABLE01 の格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,…
```

(2) pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W dat -t TABLE01 -g C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W dat：アンロードデータファイルを pdload コマンドの入力ファイル（DAT 形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld01：pdrorg コマンドの制御文ファイル名を指定します。

(3) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,…
```

(4) PURGE TABLE で TABLE01 のデータを削除します

```
PURGE TABLE TABLE01;
```

(5) ALTER TABLE で C4 列を追加します

```
ALTER TABLE TABLE01 ADD C4 INTEGER WITH DEFAULT;
```

このとき，**WITH DEFAULT** オペランドを必ず指定してください。この場合，C4 列には 0 が格納されます。

(6) 列構成情報ファイル（C:¥pdload¥column01）を作成します

```
C1
C2
C3
```

(2)で作成した入力データファイルの列構成の形式と TABLE01 の列構成が異なるため，データロードするときに列構成情報ファイルが必要になります。

●列構成情報ファイル指定のポイント

列を追加した場合の列構成情報ファイル指定のポイントは，追加した列を指定しないことです。こうすることで，HiRDB は入力データ中にその列のデータがないことを検知し，追加した列に省略値又はナル値を格納します。FIX 表の場合はナル値を格納できないため，ALTER TABLE で列を追加するときに WITH DEFAULT オペランドを指定して省略値を格納するようにします。

(7)　pdhold コマンドで TABLE01 の格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,…
```

(8)　TABLE01 にデータロードをします

```
pdload -c C:¥pdload¥column01 TABLE01 C:¥pdload¥load01
```

〔説明〕

-c C:¥pdload¥column01：(6)で作成した列構成情報ファイル名を指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(9)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

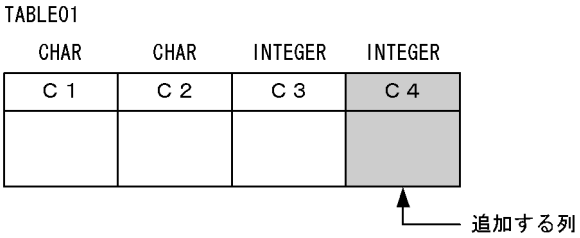
(10)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.6.8　例題 7（FIX 表に列を追加する場合：バイナリ形式でアンロードする場合）

FIX 表（TABLE01）に列（C4）を追加します。TABLE01 中には，DAT 形式ファイルに変換できない文字データ（0x00，0x0a）があるとします。



(1) pdhold コマンドで TABLE01 の格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,…
```

(2) pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。TABLE01 中には，DAT 形式ファイルに変換できない文字データ（0x00，0x0a）があるため，バイナリ形式にしてアンロードします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld01：pdrorg コマンドの制御文ファイル名を指定します。

(3) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,…
```

(4) PURGE TABLE で TABLE01 のデータを削除します

```
PURGE TABLE TABLE01;
```

(5) ALTER TABLE で C4 列を追加します

```
ALTER TABLE TABLE01 ADD C4 INTEGER WITH DEFAULT;
```

このとき，**WITH DEFAULT** オペランドを必ず指定してください。この場合，C4 列には 0 が格納されます。

(6) 列構成情報ファイル（C:¥pdload¥column01）を作成します

```
C1,type=char(4)
C2,type=char(10)
C3,type=integer
```

TABLE01 のデータ型を指定した列構成情報ファイルを作成します

(2)でバイナリ形式ファイルを作成しましたが，FIX 表なので固定長データ形式ファイルとして扱います。したがって，固定長データ形式ファイルのデータロードに必要な列構成情報ファイルを作成します。これ

によって，入力データファイルの列構成の形式と列構成が異なる TABLE01 へのデータロードができるようになります。

●列構成情報ファイル指定のポイント

列を追加した場合の列構成情報ファイル指定のポイントは，追加した列を指定しないことです。こうすることで，HiRDB は入力データ中にその列のデータがないことを検知し，追加した列に省略値又はナル値を格納します。FIX 表の場合はナル値を格納できないため，ALTER TABLE で列を追加するときに WITH DEFAULT オペランドを指定して省略値を格納するようにします。

(7) pdhold コマンドで TABLE01 の格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,…
```

(8) TABLE01 にデータロードをします

```
pdload -a -c C:¥pdload¥column01 TABLE01 C:¥pdload¥load01
```

〔説明〕

-a：入力ファイルが固定長データ形式ファイルであることを指定します。

-c C:¥pdload¥column01：(6)で作成した列構成情報ファイル名を指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(9) データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(10) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.7　列を削除する方法

実行者　HiRDB 管理者及び表の所有者

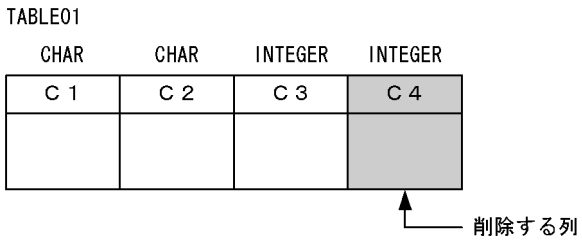
表の列を削除する場合は，**ALTER TABLE** に **DROP** オペランドを指定して実行します。

注意事項

1. データが格納されている表の列は削除できません。
2. クラスタキーを定義した列は削除できません。
3. LOB 列は削除できません。
4. 表に抽象データ型が定義されている場合は，その表のすべての列を削除できません。
5. 改竄防止表の列は削除できません。
6. 表の列を削除すると，その表を使用しているストアドルーチンが無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。
7. トリガ SQL 文中に指定している表の列を削除すると，トリガが無効になります。また，トリガ動作条件又はトリガ SQL 文中で，新旧値相関名を使用して参照している列を削除すると，トリガが無効になります。トリガが無効になった場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
8. 表の列を削除した後，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.7.1　例題（列を削除する場合）

表（TABLE01）の列（C4）を削除します。



(1)　pdhold コマンドで TABLE01 の格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,…
```

(2)　pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W dat -t TABLE01 -g C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W dat：

アンロードデータファイルを pdload コマンドの入力ファイル（DAT 形式）として使用できるようにします。

DAT 形式ファイルに変換できない文字データ（0x00，0x0a）が表中にある場合，その表データを DAT 形式でアンロードできません。この場合は，バイナリ形式（-W bin 指定）でアンロードしてください。ただし，この場合，(6)以降の操作方法（列構成情報ファイルの作成及びデータロード）が異なります。「例題 7（FIX 表に列を追加する場合：バイナリ形式でアンロードする場合）」を参考にして操作をしてください。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld01：pdrorg コマンドの制御文ファイル名を指定します。

(3)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,…
```

(4)　PURGE TABLE で TABLE01 のデータを削除します

```
PURGE TABLE TABLE01;
```

(5)　ALTER TABLE で C4 列を削除します

```
ALTER TABLE TABLE01 DROP C4;
```

(6)　列構成情報ファイル（C:¥pdload¥column01）を作成します

```
C1
C2
C3
*skipdata*
```

(2)で作成した入力データファイルの列構成の形式と TABLE01 の列構成が異なるため，データロードするときに列構成情報ファイルが必要になります。

(7)　pdhold コマンドで TABLE01 の格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2,…
```

(8)　TABLE01 にデータロードをします

```
pdload -c C:¥pdload¥column01 TABLE01 C:¥pdload¥load01
```

〔説明〕

-c C:¥pdload¥column01：(6)で作成した列構成情報ファイル名を指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(9)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(10)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.8　表の定義を変更する方法

実行者　表の所有者

表の定義を変更する場合は，**ALTER TABLE** に **CHANGE** オペランドを指定して実行します。変更できる定義を次に示します。

- 可変長の文字列のデータ長
- CHAR から MCHAR への変更
- 繰返し列の最大要素数
- 既定値のない非ナル値制約を既定値のある非ナル値制約に変更
- 一意性制約のないクラスタキーを一意性制約のあるクラスタキーに変更
- 一意性制約のあるクラスタキーを一意性制約のないクラスタキーに変更
- 表の最小排他資源単位
- ハッシュ関数

注意事項

1. LOB 列の定義は変更できません。
2. 抽象データ型の列の定義は変更できません。
3. 表の定義を変更すると，その表を使用しているストアドルーチンが無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。
4. トリガ SQL 文中に指定している表の定義を変更すると，トリガが無効になります。またトリガ動作条件又はトリガ SQL 文中で，新旧値相関名を使用して参照している列の表定義を変更すると，トリガが無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
5. 表の定義を変更した後，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.8.1　例題（列のデータ長を変更する場合）

表（TABLE01）の DATA01 列の定義長を VARCHAR(150)から VARCHAR(200)に変更します。

(1)　ALTER TABLE で DATA01 列の定義を変更します

```
ALTER TABLE TABLE01 CHANGE DATA01 VARCHAR(200);
```

13.9　表名又は列名を変更する方法

実行者　表の所有者

表名又は列名を変更する場合は，**ALTER TABLE** に **RENAME** オペランドを指定して実行します。

注意事項

1. 改竄防止表の表名又は列名は変更できません。
2. 表名又は列名を変更すると，その表を使用しているストアドルーチンが無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。
3. トリガ SQL 文中に指定している表の表名又は列名を変更すると，トリガが無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
4. トリガを定義した表の表名は変更できません。
5. トリガを定義した表の列のうち，次に示す列の列名は変更できません。
   - トリガ契機列
   - トリガ動作条件中で新旧値相関名を使用して参照している列
   - トリガ SQL 文中で新旧値相関名を使用して参照している列

13.9.1　例題１（表名を変更する場合）

表の名称を TABLE01 から TABLE02 に変更します。

(1)　ALTER TABLE で表名を変更します

```
ALTER TABLE TABLE01 RENAME TABLE TO TABLE02;
```

13.9.2　例題２（列名を変更する場合）

表（TABLE01）の C1 列の名称を C2 に変更します。

(1)　ALTER TABLE で列名を変更します

```
ALTER TABLE TABLE01 RENAME COLUMN FROM C1 TO C2;
```

13.10　表の横分割数を増やす方法

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

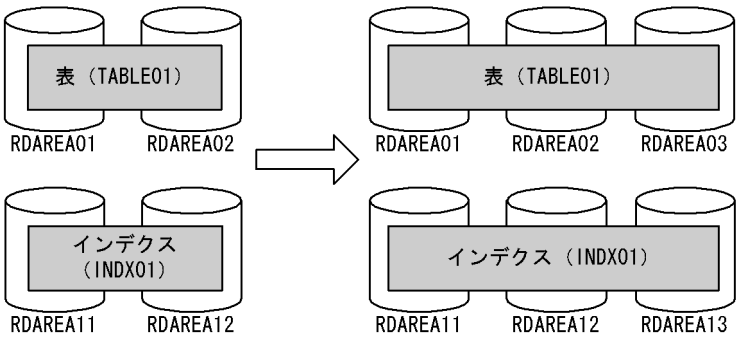
表に対してデータの追加を繰り返した結果，一つの表のデータ量が大きくなると，この表に対する処理性能が悪くなる場合があります。このような場合は，表の横分割数を増やしてください。

注意事項

1. キーレンジ分割表の場合，DROP TABLE で表を削除して CREATE TABLE で表を再定義してください。表の再定義時に表の横分割数を増やしてください。
2. ハッシュ分割表の場合，ALTER TABLE で表の横分割数を増やせます。
3. 表の横分割数を増やすと，その表を使用しているストアドルーチンが無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。
4. トリガ SQL 文中に指定している表の横分割数を増やすと，トリガが無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
5. 表の横分割数を増やした後，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。ただし，抽象データ型は，最適化情報収集ユティリティの対象外となります。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.10.1　例題１（キーレンジ分割表の横分割数を増やす場合）

表（TABLE01）の横分割数を二つから三つにします。インデクス（INDX01）も同様に分割します。TABLE01 はキーレンジ分割しています。



(1)　pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2) pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld01：pdrorg コマンドの制御文ファイル名を指定します。

(3) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

(4) DROP TABLE で TABLE01 を削除します

```
DROP TABLE TABLE01;
```

(5) CREATE TABLE 及び CREATE INDEX で，TABLE01 と INDX01 を再定義します

```
CREATE TABLE TABLE01 …
    IN ((RDAREA01),(RDAREA02),(RDAREA03)) … ;
CREATE INDEX INDX01 …
    IN ((RDAREA11),(RDAREA12),(RDAREA13));
```

(6) pdhold コマンドでデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(7) pdload コマンドで TABLE01 にデータをデータロードします

```
pdload -b -W TABLE01 C:¥pdload¥load01
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(8)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(9)　pdrels コマンドで RD エリアの閉塞を解除します

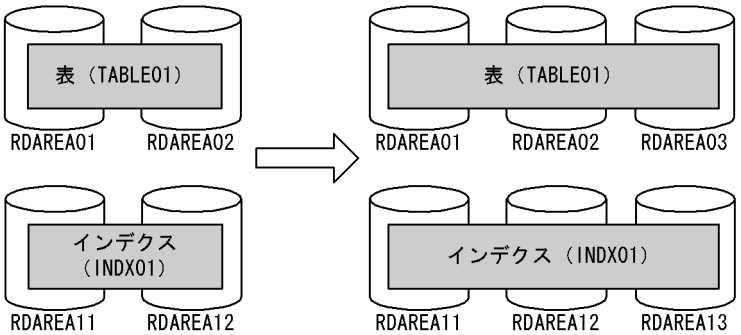
```
pdrels -r RDAREA01,RDAREA02,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.10.2　例題２（フレキシブルハッシュ分割表の横分割数を増やす場合）

表（TABLE01）の横分割数を二つから三つにします。インデクス（INDX01）も同様に分割します。TABLE01 はフレキシブルハッシュ分割しています。

なお，この表はハッシュ分割表のリバランス機能を使用していないとします。



(1)　ALTER TABLE で TABLE01 に RDAREA03 を追加します

```
ALTER TABLE TABLE01 ADD RDAREA RDAREA03;
```

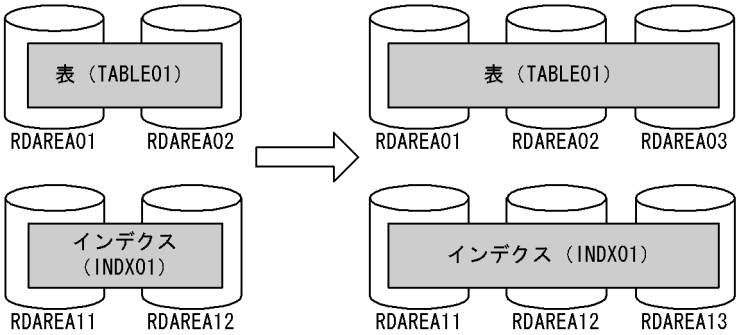
このとき，RDAREA03 にデータは格納されません。この後に追加されるデータがハッシュ関数を使用して格納されます。

ハッシュ分割表のリバランス機能を使用すると，RDAREA03 にデータが格納されるため，既存の RD エリアと新規追加した RD エリアとの間でデータ量の偏りが生じません。ハッシュ分割表のリバランス機能については，「表の横分割数を増やす方法（ハッシュ分割表のリバランス機能を使用している場合）」を参照してください。

13.10.3　例題３（FIX ハッシュ分割表の横分割数を増やす場合）

表（TABLE01）の横分割数を二つから三つにします。インデクス（INDX01）も同様に分割します。TABLE01 は FIX ハッシュ分割しています。

なお，この表はハッシュ分割表のリバランス機能を使用していないとします。ハッシュ分割表のリバランス機能を使用すると，表データがあっても ALTER TABLE で RD エリアを追加できます。このため，表データのアンロード及びデータロードが不要になります。ハッシュ分割表のリバランス機能については，「表の横分割数を増やす方法（ハッシュ分割表のリバランス機能を使用している場合）」を参照してください。



(1)　pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2)　pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld02
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld02：pdrorg コマンドの制御文ファイル名を指定します。

(3)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

(4)　PURGE TABLE 文で TABLE01 のデータを削除します

```
PURGE TABLE TABLE01;
```

FIX ハッシュ分割した表の横分割数を増やす場合は，表のデータを削除する必要があります。

(5)　ALTER TABLE で TABLE01 に RDAREA03 を追加します

```
ALTER TABLE TABLE01 ADD RDAREA RDAREA03;
```

(6)　pdhold コマンドでデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(7)　pdload コマンドで TABLE01 にデータをデータロードします

```
pdload -b -W TABLE01 C:¥pdload¥load02
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load02：pdload コマンドの制御文ファイル名を指定します。

(8)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(9)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.11　表の横分割数を増やす方法（ハッシュ分割表のリバランス機能を使用している場合）

実行者　HiRDB 管理者及び表の所有者

ここでは，ハッシュ分割表のリバランス機能を使用して表の横分割数を増やす方法について説明します。ここで説明する項目は次のとおりです。

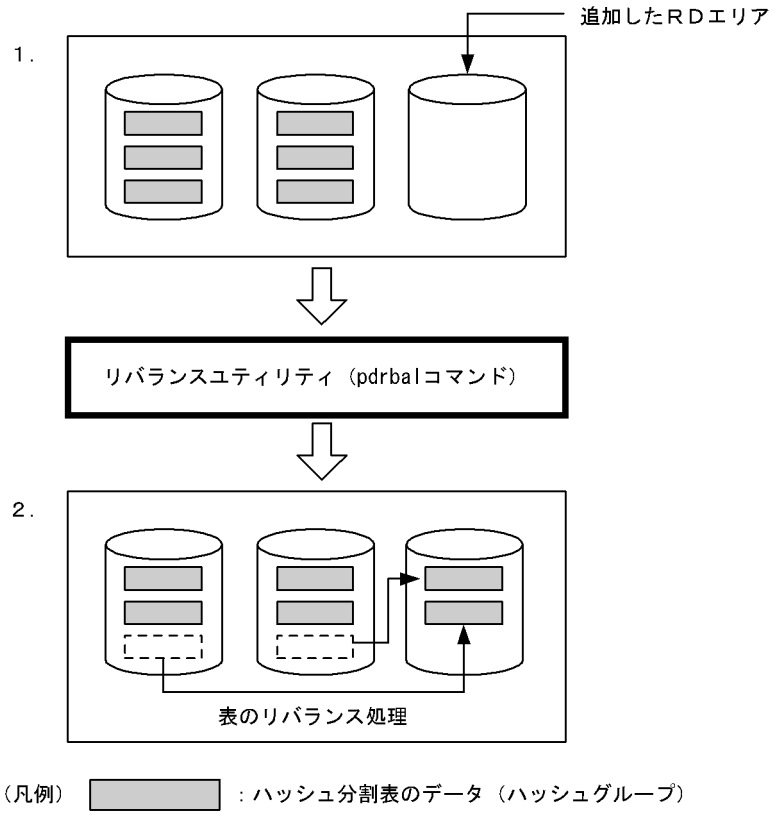
- ハッシュ分割表のリバランス機能とは
- ハッシュ分割表のリバランス機能を使用するための準備
- 例題（リバランス表の横分割数を増やす場合）
- リバランスユティリティの使い方（表のリバランス処理に時間が掛かる場合）
- FIX ハッシュ分割表に対する注意事項

13.11.1　ハッシュ分割表のリバランス機能とは

ハッシュ分割表のデータ量が増加したため RD エリアを追加すると（表の横分割数を増やすと），既存の RD エリアと新規追加した RD エリアとの間でデータ量の偏りが生じます。**ハッシュ分割表のリバランス機能**を使用すると，表の横分割数を増やすときにデータ量の偏りを修正できます。ハッシュ分割表のリバランス機能を次の図に示します。

なお，ハッシュ分割表のリバランス機能は，FIX ハッシュ及びフレキシブルハッシュのどちらにも適用できます。

図 13–11　ハッシュ分割表のリバランス機能



〔説明〕

1. ハッシュ分割表のデータが一杯になったため，ハッシュ分割表を格納する RD エリアを追加しました（表の横分割数を増やしました）。追加した RD エリアにはデータが配置されず，データ量の偏りが生じます。
2. **リバランスユティリティ**（pdrbal コマンド）を実行してデータ量の偏りを修正します。リバランスユティリティを実行すると，ハッシュグループ単位にデータが移動して再配置されます。これを**表のリバランス**といいます。

   ハッシュ分割表のリバランス機能を使用すると，分割キーをハッシュした結果を基にして HiRDB がデータを 1024 のグループ（これを**ハッシュグループ**といいます）に分けます。このグループごとに RD エリアのセグメントを割り当ててデータを格納します。データの再配置もこのハッシュグループ単位に行います。

(1)　適用基準

- データの増加が見込まれていて，将来 RD エリアを追加する可能性がある場合※
- データ容量が大きいため，表を再作成するのが難しい場合

注※

データが格納されている FIX ハッシュ分割表（ハッシュ関数 HASH1～HASH6 を使用した FIX ハッシュ分割表）には RD エリアを追加できませんが，リバランス機能を持つハッシュ関数 HASHA～HASHF を使用した FIX ハッシュ分割表には RD エリアを追加できるようになります。

(2)　注意事項

各ハッシュグループのデータ量はハッシュ関数のハッシング結果に依存します。したがって，分割キー値が偏っていると，ハッシュグループのデータ量も偏ってしまい，データを均等に分割できないことがあります。

(3)　運用方法

ハッシュ分割表のリバランス機能の運用手順の概略を次に示します。

〈手順〉

1. ハッシュ分割表を定義するときに，その表を**リバランス表**として定義します。
2. 表の横分割数を増やすため，表格納 RD エリアを追加します。
3. リバランスユティリティ※を実行して表のリバランスを行います。

注※　リバランスユティリティには次の表に示す実行モードがあります。

表 13–2　リバランスユティリティの実行モード

実行モードの種類	説明
共有モード	表のリバランス処理中に表をアクセスできます。 大規模データベースの場合，表のリバランス処理に数日掛かることがあります。その間，リバランス表へのアクセスを実行しながらリバランス処理を実行できます。ただし，表へのアクセス処理と表のリバランス処理を同時に実行するため，両方の処理性能が低下します。なお，処理性能の低下をなるべく防ぐためにトラフィックの少ない時間帯にリバランス処理を行うように処理を複数回に分けられます。
占有モード	表のリバランス処理中に表をアクセスできません。 リバランス処理に専念できるため，性能的に共有モードより優れています。表へのアクセスを中断できるならば，占有モードでリバランスユティリティを実行することをお勧めします。 なお，占有モードの場合も，リバランス処理を複数回に分けられます。

13.11.2　ハッシュ分割表のリバランス機能を使用するための準備

(1)　リバランス表の定義

CREATE TABLE で指定するハッシュ関数に HASHA～HASHF を指定します。HASHA～HASHF を指定した表を**リバランス表**といいます。ハッシュ関数（HASHA～HASHF）については，マニュアル「HiRDB SQL リファレンス」を参照してください。

```
CREATE TABLE TABLE01
    (SCODE DEC(5) NOT NULL,SNAME NCHAR(15), … ) HASH HASHF BY SCODE
     IN (RDAREA01,RDAREA02);
CREATE INDEX INDX01 …
     IN ((RDAREA11),(RDAREA12));
```

(2)　リバランス表を格納する RD エリア

1. リバランス表を格納する RD エリアには，そのリバランス表だけしか格納できません。ほかの表及びインデクスを格納できません。
2. リバランス表を格納する RD エリアには，**↑1024÷分割 RD エリア数↑**分のセグメント数が必要です。RD エリアを作成するときにはこの条件を満たすようにしてください。
3. RD エリアの容量を見積もる場合，**↑1024÷分割 RD エリア数↑**分のセグメント数の余裕値をとってください。
4. リバランス表にデータを格納するには，未使用セグメントが必要になります。このため，空きページがあっても RD エリア満杯エラーとなることがあります。したがって，RD エリアの残容量は空きページ数ではなく，未使用セグメント数で管理してください。

(3)　リバランス表にインデクスを定義する場合

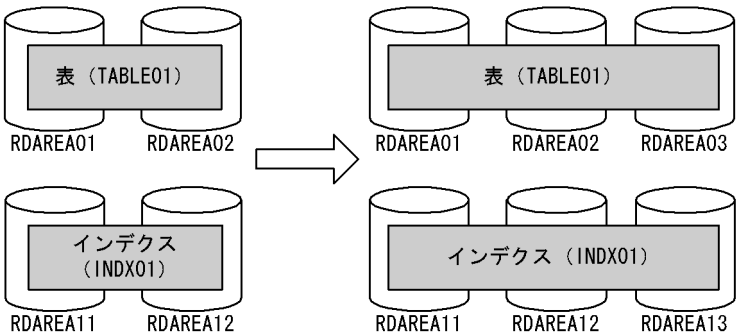
- リバランス表を格納する RD エリアには，そのリバランス表だけしか格納できません。ほかの表及びインデクスを格納できません。したがって，リバランス表に対して CREATE INDEX を指定する場合は，インデクス格納 RD エリア名を省略できません。省略するとリバランス表を格納した RD エリアが仮定されて，エラーになります。必ず，リバランス表を格納する RD エリアと別の RD エリアを指定してください。
- クラスタキーインデクスは定義できません。

(4)　グローバルバッファ

表格納 RD エリアごとにグローバルバッファの面数を**↑1024÷分割 RD エリア数↑×2** 以上にしてください。そうしないと，バッファリング効果が得られなくなります。

13.11.3　例題（リバランス表の横分割数を増やす場合）

リバランス表（TABLE01）の横分割数を二つから三つにします。インデクス（INDX01）も同様に分割します。このように，**インデクス格納 RD エリアと表格納 RD エリアの数が一致しないとリバランスユティリティを実行できません。**



(1) pdfmkfs コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します

```
pdfmkfs -n 50 -l 10 -k DB -i C:¥rdarea¥area03
pdfmkfs -n 50 -l 10 -k DB -i C:¥rdarea¥area13
```

〔説明〕

-n：HiRDB ファイルシステム領域の領域長をメガバイト単位で指定します。データベース状態解析ユティリティ(pddbst コマンド)で，現状の RD エリアの状態を解析し，移動対象となるデータ量から追加する RD エリアの容量見積もりをしてください。

-l：HiRDB ファイルシステム領域内に作成する HiRDB ファイル数の上限値を指定します。

-k：HiRDB ファイルシステム領域の用途を指定します。RD エリア用の HiRDB ファイルシステム領域を作成するため，DB を指定します。

-i：HiRDB ファイルシステム領域の全領域を初期化する場合に指定します。-i オプションを指定すると，領域全体を確保します。-i オプションを省略すると，HiRDB ファイルシステム領域の管理情報だけを作成します。

C:¥rdarea¥area03：作成する HiRDB ファイルシステム領域（表格納 RD エリア用）の名称を指定します。

C:¥rdarea¥area13：作成する HiRDB ファイルシステム領域（インデクス格納 RD エリア用）の名称を指定します。

(2) pdmod コマンドで RD エリアを追加します

```
pdmod -a C:¥pdmod¥create01
```

〔説明〕

-a：pdmod コマンドの制御文ファイル名を指定します。制御文ファイルの例を次に示します。各オペランドの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

```
create rdarea RDAREA03
   globalbuffer gbuf03
   for user used by PUBLIC
   server name bes1
   page 4096 characters
   storage control segment 10 pages
   file name ”C:¥rdarea¥area03¥file01”
      initial 1000 segments;
create rdarea RDAREA13
   globalbuffer gbuf13
   for user used by PUBLIC
   server name bes1
   page 4096 characters
   storage control segment 10 pages
   file name ”C:¥rdarea¥area13¥file01”
      initial 1000 segments;
```

(3) ALTER TABLE で表の横分割数を増やします

```
ALTER TABLE TABLE01
    ADD RDAREA RDAREA03
      FOR INDEX INDX01 IN RDAREA13;
```

表データ格納用 RD エリア（RDAREA03）とインデクス格納用 RD エリア（RDAREA13）を追加しました。これで，TABLE01 の横分割数が二つから三つに増えました。

(4) pdrbal コマンドで表のリバランスを行います

(a) 共有モードで実行する場合

```
pdrbal -k share -t TABLE01 -c 100 C:¥pdrbal¥cfile01
```

〔説明〕

-k：共有モードで実行するため share を指定します。

-t：リバランス表の名称を指定します。

-c：コミット単位を指定します。

C:¥pdrbal¥cfile01：pdrbal コマンドの制御文ファイルの名称を指定します。制御文ファイルの内容を次に示します。

```
report C:¥tmp¥output
execstop time,9:00
```

〔説明〕

report：pdrbal コマンドの実行結果の出力先を指定します。

execstop：pdrbal コマンドの実行時間を指定します。

(b) 占有モードで実行する場合

```
pdrbal -k exclusive -t TABLE01 -l a C:¥pdrbal¥cfile01
```

〔説明〕

-k：占有モードで実行するため exclusive を指定します。

-t：リバランス表の名称を指定します。

-l：データベースの更新ログ取得方式を指定します。この例では a（ログ取得モード）を指定します。

C:¥pdrbal¥cfile01：pdrbal コマンドの制御文ファイルの名称を指定します。制御文ファイルの内容を次に示します。

```
report C:¥tmp¥output
idxwork C:¥index¥work
sort C:¥sort¥work
```

〔説明〕

report：pdrbal コマンドの実行結果の出力先を指定します。

idxwork：インデクス情報ファイル出力用のディレクトリ名を指定します。

sort：ソート用ワークディレクトリ名を指定します。

(5) pdgetcst コマンドで表の最適化情報を取得します

必要であれば最適化情報収集ユティリティ（pdgetcst コマンド）で表の最適化情報を取得してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

```
pdgetcst -t TABLE01 -l C:¥pdgetcst¥output
```

〔説明〕

-t：最適化情報を取得する表名を指定します。

-l：pdgetcst コマンドの実行結果の出力先を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.11.4 リバランスユティリティの使い方（表のリバランス処理に時間が掛かる場合）

ここでは，表のリバランス処理（リバランスユティリティの処理）に時間が掛かる場合のリバランスユティリティの使い方を説明します。

(1) 共有モードでの使い方

共有モードの場合は，リバランス表へのアクセスと同時にリバランス処理を行えます。よって，24 時間運転などオンラインを停止できない場合に有効です。ここでは，システムの 24 時間運転を前提とした説明を行います。

●使い方の指針

- 比較的トラフィックが空いている時間帯にリバランスユティリティを実行します。
- 1 回のリバランスユティリティで処理が終了しない場合は，複数回に分けてリバランスユティリティを実行します。

●運用例

21：00 から 9：00 までの間，比較的トラフィックが空いています。この場合，21：00 にリバランスユティリティを実行し，制御文ファイルの execstop time オペランドで 12 時間ユティリティを実行する指定をします。

●コマンド指定例

```
pdrbal -k share -t TABLE01 -c 100 C:¥pdrbal¥cfile01
```

●制御文ファイルの内容

```
execstop time,12:00
```

●リターンコードについて

12 時間経過しても（9：00 になっても）リバランス処理が終了しない場合，リバランスユティリティがリターンコード 4 で終了し，リバランス処理をいったん中断します。21：00 になったら再度リバランスユティリティを実行してください。これを毎日繰り返してください。リバランス処理が終了すると，リターンコードが 0 で返ってきます。

(2)　占有モードでの使い方

占有モードの場合，リバランスユティリティ実行中はリバランス表に対してアクセスできません。

●使い方の指針

- リバランス表のアクセスを停止できる時間帯にリバランスユティリティを実行します。
- 1 回のリバランスユティリティで処理が終了しない場合は，複数回に分けてリバランスユティリティを実行します。

●運用例

23：00 から 5：00 までの間，リバランス表のアクセスを停止できます。この場合，23：00 にリバランスユティリティを実行し，制御文ファイルの execstop time オペランドで 6 時間ユティリティを実行する指定をします。

●コマンド指定例

```
pdrbal -k exclusive -t TABLE01 C:¥pdrbal¥cfile01
```

●制御文ファイルの内容

```
execstop time,6:00
```

●リターンコードについて

6 時間経過しても（5：00 になっても）リバランス処理が終了しない場合，リバランスユティリティがリターンコード 4 で終了し，リバランス処理をいったん中断します。23：00 になったら再度リバランスユティリティを実行してください。これを毎日繰り返してください。リバランス処理が終了すると，リターンコードが 0 で返ってきます。

●インデクスを定義している場合の注意事項

インデクスを定義した表をインデクス一括作成モードで処理する場合，execstop time オペランドで指定した時間でリバランスユティリティが終了しないことがあります。これはインデクスの作成が完了しないと，リバランスユティリティを終了できないためです。したがって，インデクス一括作成モードで処理する場合は，execstop time オペランドの指定値に余裕を持たせてください。

13.11.5　FIX ハッシュ分割表に対する注意事項

FIX ハッシュ分割表にハッシュ分割表のリバランス機能を適用すると，表のリバランス処理中は次に示す検索処理の性能が低下します。

- 分割キーに条件を指定した検索
- GROUP BY 句に指定する列がすべての分割キー含む場合の検索

また，UNIQUE 指定のインデクスを定義している場合，リバランス処理中は UNIQUE 指定のインデクスを更新できません。

13.12　表の分割格納条件の変更

実行者　HiRDB 管理者及び表の所有者

キーレンジ分割※で横分割した表の分割格納条件を，ALTER TABLE で変更できます。表の分割格納条件を変更することで，古いデータを格納していた RD エリアを再利用でき，作業時間を短くできます。ALTER TABLE を使用すると，分割格納条件を変更する表を削除し，再作成する必要がありません。

注※

次に示す分割方法の場合に，表の分割格納条件を ALTER TABLE で変更できます。

- 境界値指定
- 格納条件指定（格納条件の比較演算子に＝だけを使用している場合）
- マトリクス分割（境界値指定のキーレンジ分割の次元だけ）

なお，表の分割格納条件を変更するには，**HiRDB Advanced High Availability** が必要です。

13.12.1　分割格納条件の変更の目的

横分割表に格納されたデータは，時間経過とともに増加していきます。この結果，過去の古いデータを削除する必要が出てきます。しかし，データの登録日時などを分割キーに指定している場合，古いデータを削除するだけでは，古いデータを格納していた RD エリアを再利用できません。古いデータを格納していた RD エリアを再利用するためには，次の運用が必要となります。

1. 変更する表のデータをすべてアンロードします。
2. 変更する表を削除します。
3. 新たな分割格納条件を指定して表を再作成します。
4. 再作成した表に対してデータをデータロードします。

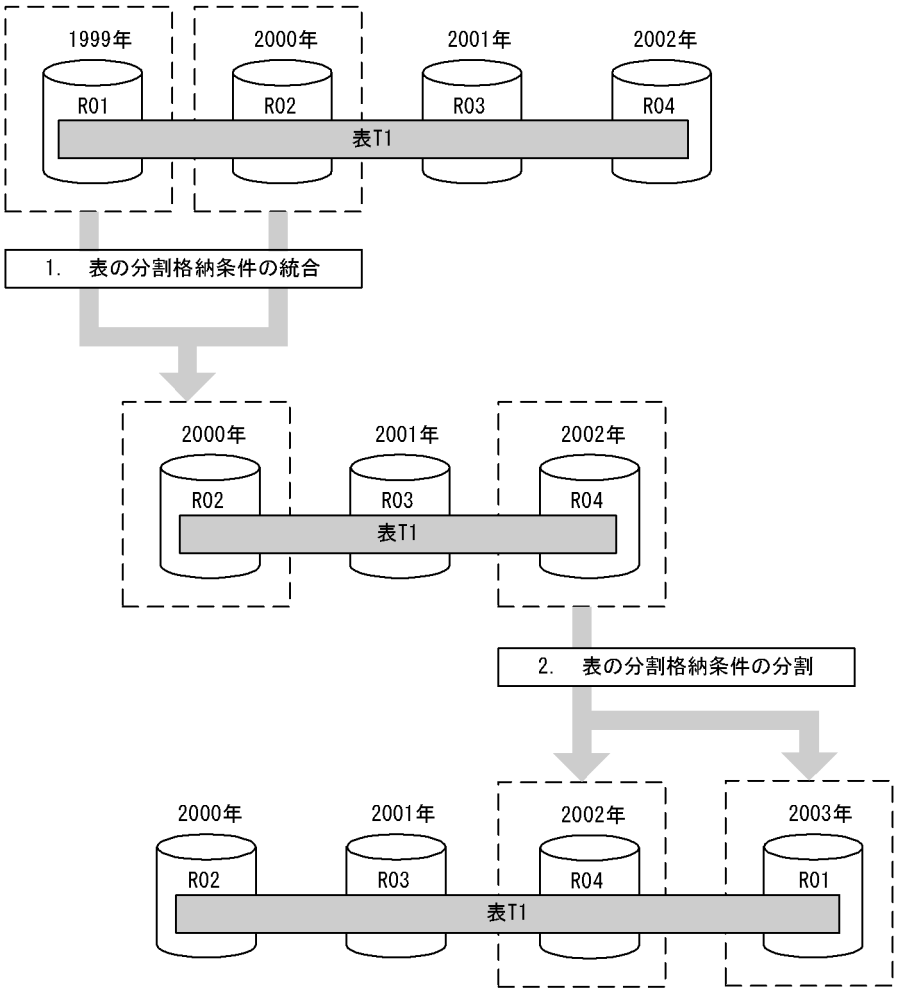
この変更作業をする間，変更する表に対する業務は一時停止しなくてはなりません。変更する表の全データをアンロード後，データロードする必要があるため，作業時間が長くなり，その間停止している業務への影響は大きくなります。

横分割した表の分割格納条件を ALTER TABLE で変更することで，RD エリアを再利用するための作業時間を短くできます。

(1)　分割格納条件の変更の概要（境界値指定の場合）

分割格納条件の変更の概要（境界値指定の場合）を次の図に示します。

図 13–12　分割格納条件の変更の概要（境界値指定の場合）



〔説明〕

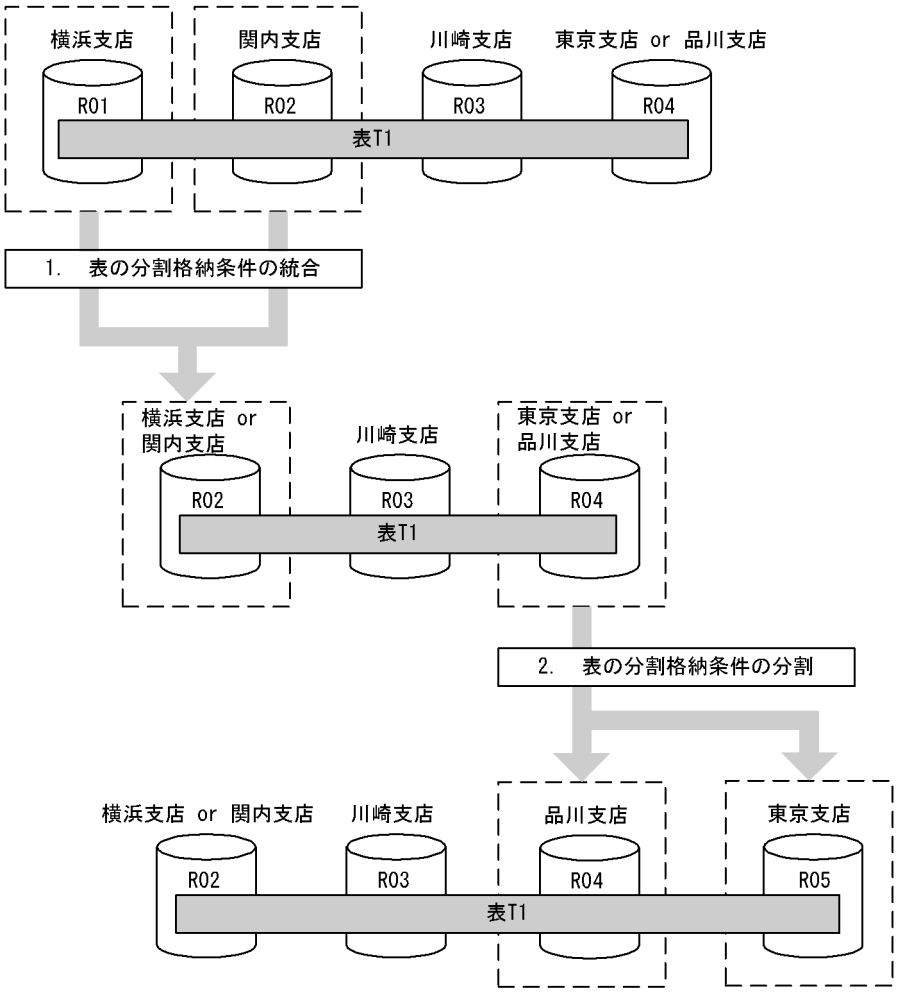
1. 分割格納条件の統合によって，RD エリア（R01，R02）に格納していた，1999 年以前と 2000 年の格納条件（定義情報）を一つに統合して，RD エリア R02 に格納できます。同時に RD エリア R01 中のデータを削除します。
2. 分割格納条件の分割によって，2002 年の格納条件を 2002 年の格納条件と 2003 年の格納条件に分割します。この結果，2002 年のデータを RD エリア R04 に，2003 年以降のデータを 1999 年のデータを格納していた RD エリア R01 を再利用して格納できます。

この結果，1.のときは R01 と R02 のデータだけをアンロードし，分割格納条件の変更後，R02 へロードするだけでよくなります。また，2.のときは R04 だけをアンロードし，分割格納条件の変更後，R04 と R01 にロードすればよいことになります。また，1.で R01 の内容を削除してもよく，2.で 2003 年以降のデータが R04 になければ，アンロードとデータロードの必要がないため，短い時間で RD エリアの構成を変更できます。このため，上昇していくキーの値とともに増加するデータを格納する RD エリアを循環して再利用する運用ができるようになります。

(2)　分割格納条件の変更の概要（格納条件指定の場合）

分割格納条件の変更の概要（格納条件指定の場合）を次の図に示します。

図 13–13　分割格納条件の変更の概要（格納条件指定の場合）



〔説明〕

支店名をキーにして表を格納条件指定で横分割しています。支店の統廃合によって，支店の削除又は追加がある場合，支店のデータを複数の支店に分割したり，複数の支店のデータを一つの支店に統合したりできます。

1. 横浜支店と関内支店が統合されるため，両支店のデータを RD エリア（R02）に統合し，RD エリア（R01）中のデータを削除します。
2. 品川支店が東京支店から分離して追加されるため，RD エリア（R04）中のデータを分割します。品川支店のデータを RD エリア（R04）に格納し，東京支店のデータを RD エリア（R05）に格納します。

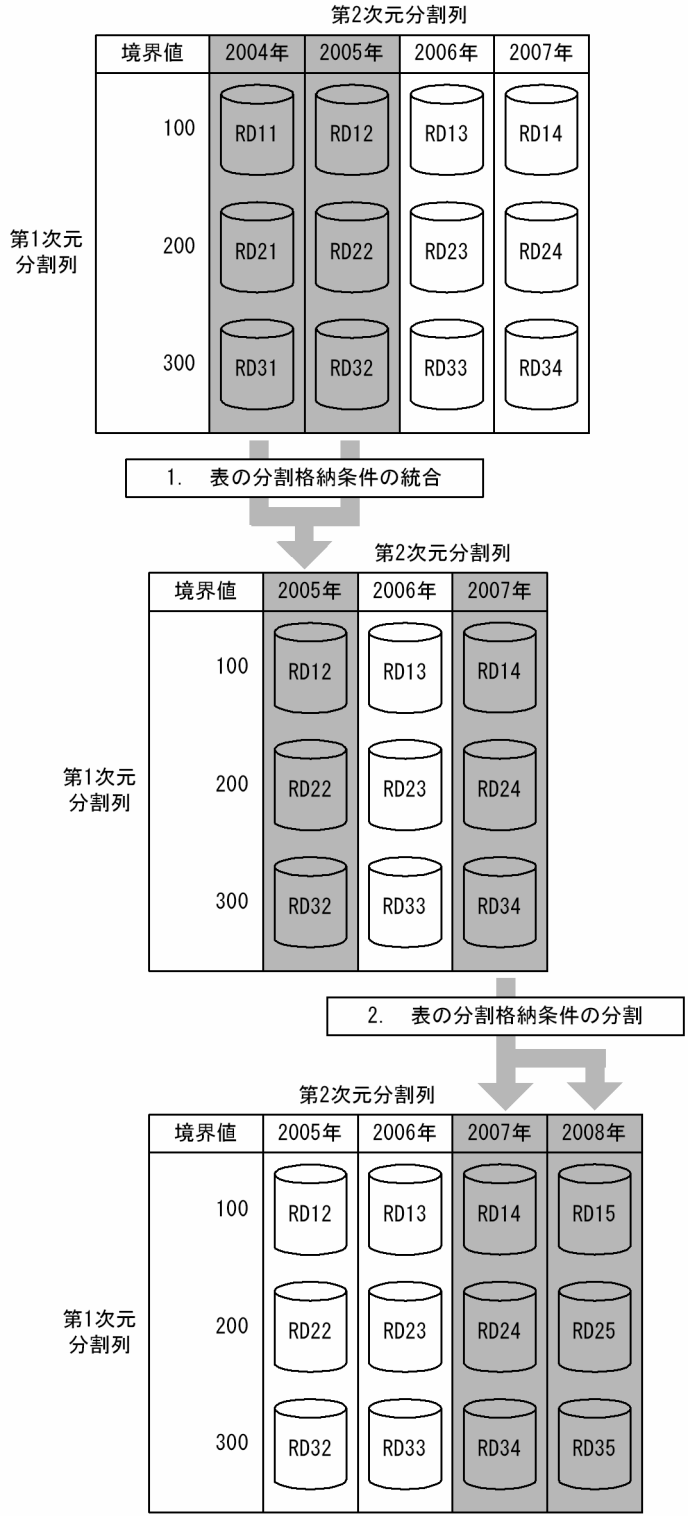
ポイント

- 「1．表の分割格納条件の統合」のときは，アンロードする必要があるデータは R01 と R02 のデータだけです。R03 と R04 のデータはアンロードする必要はありません。
- 「2．表の分割格納条件の分割」のときは，アンロードする必要があるデータは R04 のデータだけです。R02 と R03 のデータはアンロードする必要はありません。

(3)　分割格納条件の変更の概要（マトリクス分割の場合）

分割格納条件の変更の概要（マトリクス分割の場合）を次の図に示します。

図 13–14　分割格納条件の変更の概要（マトリクス分割の場合）



〔説明〕

1. 第 2 次元の分割格納条件の統合によって，RD エリア（RD11，RD12，RD21，RD22，RD31，及び RD32）に格納していた，2004 年以前と 2005 年の格納条件（定義情報）を一つに統合して，RD エリア RD12，RD22，及び RD32 に格納します。同時に RD エリア RD11，RD21，及び RD31 のデータを削除します。
2. 分割格納条件の分割によって，2007 年の格納条件を 2007 年の格納条件と 2008 年の格納条件に分割します。2007 年のデータを RD エリア RD14，RD24，及び RD34 に，2008 年以降のデータを RD エリア RD15，RD25，及び RD35 に格納します。

13.12.2　分割格納条件の変更時に使用する機能

表の分割格納条件を変更するときに使用する機能について説明します。

(1)　分割機能

表の分割格納条件を変更（分割）し，1RD エリア中のデータを複数の RD エリアに分割して格納します。分割機能の概要を次の図に示します。

図 13–15　分割機能の概要（境界値指定の場合）

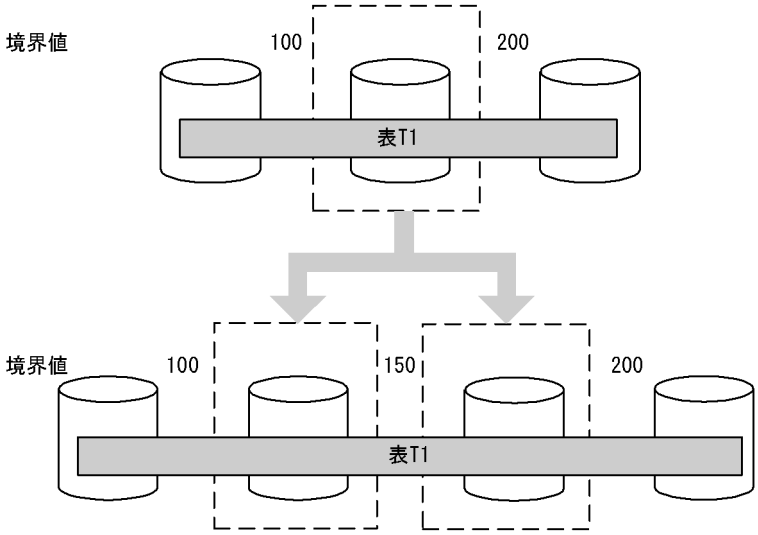
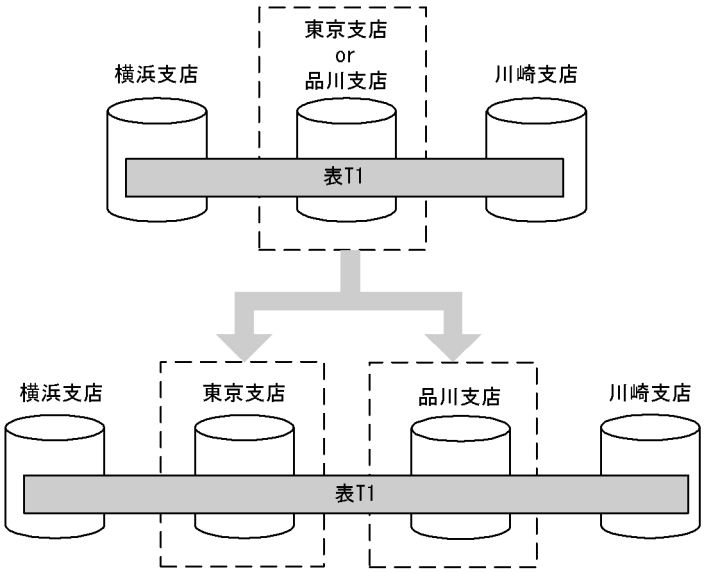


図 13–16　分割機能の概要（格納条件指定の場合）



分割する場合，分割する RD エリアに格納されているデータは削除されます。このため，分割前の格納条件を格納している RD エリアのデータをアンロードしておき，分割後に分割された格納条件を格納する RD エリアにデータロードする必要があります。ただし，分割前の RD エリアをそのまま分割後の RD エリアとして使用する場合だけ分割前に格納されていたデータをアンロード，及びデータロードしないで使用することもできます。

分割前の RD エリアを分割後の RD エリアとして使用しない場合，分割前の RD エリアに格納されていたデータを使用することはできません。

(2)　統合機能

表の分割格納条件を変更（統合）し，複数の RD エリアのデータを一つの RD エリアに格納します。統合機能の概要を次の図に示します。

図 13–17　統合機能の概要（境界値指定の場合）

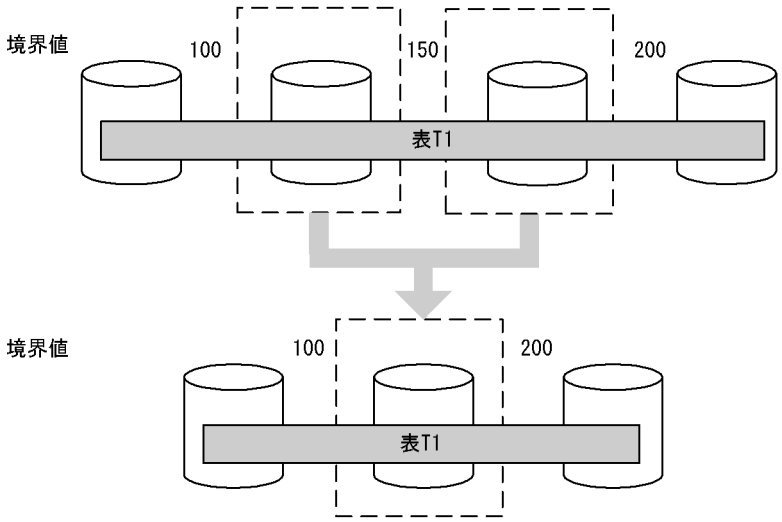
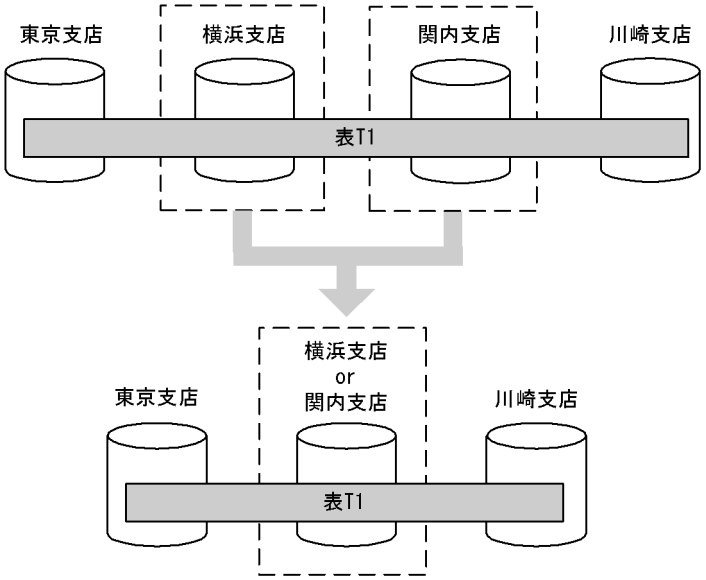


図 13–18　統合機能の概要（格納条件指定の場合）



統合する場合，統合する RD エリアに格納されているデータは削除されます。このため，統合前の格納条件を格納しているすべての RD エリアのデータをアンロードしておき，統合後に統合された格納条件を格納する RD エリアにデータロードする必要があります。ただし，統合前の一つの RD エリアを統合後の RD エリアとして使用する場合だけ統合前に格納されていたデータを使用することもできます。

統合前の RD エリアを統合後の RD エリアとして使用しない場合，統合前の RD エリアに格納されていたデータを使用することはできません。統合前の RD エリアに格納されているデータを新たに追加した RD エリアに格納したい場合は，統合前に統合するすべての RD エリアのデータをアンロードしておき，統合後にデータロードする必要があります。

13.12.3　前提条件

(1)　表の分割方法

キーレンジ分割が前提となります。表の分割方法による分割格納条件変更の適用範囲を次の表に示します。

表 13–3　表の分割方法による分割格納条件変更の適用範囲

表の分割方法			適用可否
キーレンジ分割	格納条件指定	格納条件の比較演算子に＝だけを使用している場合	○
		格納条件の比較演算子に＝以外の条件を使用している場合	×
	境界値指定		○
ハッシュ分割（リバランス含む）	FIX ハッシュ指定なし		×※1
	FIX ハッシュ指定あり		
マトリクス分割	境界値指定のキーレンジ分割との組み合わせ		○※2

表の分割方法		適用可否
	境界値指定のキーレンジ分割とハッシュ分割の組み合わせ	○※3

（凡例）

○：適用できます。

×：適用できません。

注※1

定義系 SQL の ALTER TABLE で ADD RDAREA を指定して RD エリアの追加だけができます。

注※2

1 回の実行で，一つの次元だけ変更できます。

注※3

境界値指定のキーレンジ分割の次元だけ変更できます。

(2)　表の種類による適用範囲

次の表については，分割格納条件の変更はできません。

- 非分割表：非分割表には分割格納条件がないため，分割格納条件は変更できません。
- 改竄防止表：分割格納条件の変更の指定によっては，RD エリア内の対象表のデータが削除されます。データの削除はデータの改竄となるため，改竄防止表の分割格納条件は変更できません。
- 抽象データ型を含む表：抽象データ型を含む表の分割格納条件は変更できません。

(3)　前提製品

分割格納条件を変更するためには，HiRDB Advanced High Availability が必要です。この製品をセットアップしないで分割格納条件を変更（ALTER TABLE の CHANGE RDAREA）しようとした場合，KFPA11948-E メッセージを出力して ALTER TABLE の実行をエラーとします。

(4)　そのほかの適用範囲

(a)　インデクス

分割格納条件を変更する表にインデクスが定義されている場合，インデクスの定義条件によっては分割格納条件を変更できないケースがあります。インデクスの種類と分割格納条件の変更の適用可否を次の表に示します。

表 13–4　インデクスの種類と分割格納条件の変更の適用可否

インデクスの種類	インデクスの 分割／非分割	適用可否
クラスタキーインデクス	分割キーインデクス	○※1

インデクスの種類	インデクスの 分割／非分割	適用可否
プライマリキーインデクス （プライマリキー及びクラスタキーを定義したインデクスを含みます）	分割キーインデクス	○※1
B-tree インデクス	分割キーインデクス	○※1
	非分割キーインデクス	○※1※2
プラグインインデクス	該当しません。	×

（凡例）

○：適用できます。

×：適用できません。

注※1

インデクス格納用 RD エリアの分割条件によっては，分割格納条件が変更できないことがあります。インデクス格納用 RD エリアの分割条件による分割格納条件の変更可否を次の表に示します。

注※2

表格納用 RD エリアと 1 対 1 に分割されている場合は，分割格納条件を変更できます。非分割キーインデクスの場合の分割格納条件の変更の適用可否を表「非分割キーインデクスの場合の分割格納条件の変更の適用可否」に示します。

表 13–5　インデクス格納用 RD エリアの分割条件による分割格納条件の変更可否

インデクス格納用 RD エリアの分割条件			分割格納条件の変更可否	
			分割	統合
分割キーインデクス	インデクス格納用 RD エリアを分割している		○	○※1
	インデクス格納用 RD エリアを分割していない		×※2	×※3
非分割キーインデクス	表格納用 RD エリアとインデクス格納用 RD エリアが 1 対 1 に対応するように分割されている	インデクス格納用 RD エリアを分割している	○	○※1
		インデクス格納用 RD エリアを分割していない	×※2	×※3
	上記以外		×	×

（凡例）

○：分割格納条件を変更できます。

×：分割格納条件を変更できません。

注※1

統合後にインデクス格納用 RD エリアの総数が 1 個になる場合は，横分割インデクスから非横分割インデクスへの変更を伴うため，分割格納条件を変更できません。

注※2

非横分割インデクスから横分割インデクスへの変更を伴うため，分割格納条件を変更できません。

注※3

横分割インデクスを非分割インデクスにできるケースは，対象表を格納する RD エリアが 1 個のときだけです。そのため，分割格納条件を変更できません。

インデクス格納用 RD エリアと表格納用 RD エリアとの関係によって，分割格納条件を変更できない場合は，該当するインデクスを削除してから ALTER TABLE を実行し，その後にインデクスを定義し直してください。

表格納用 RD エリアを分割できないケースを次の図に示します。

図 13–19　表格納用 RD エリアを分割できないケース（その 1）

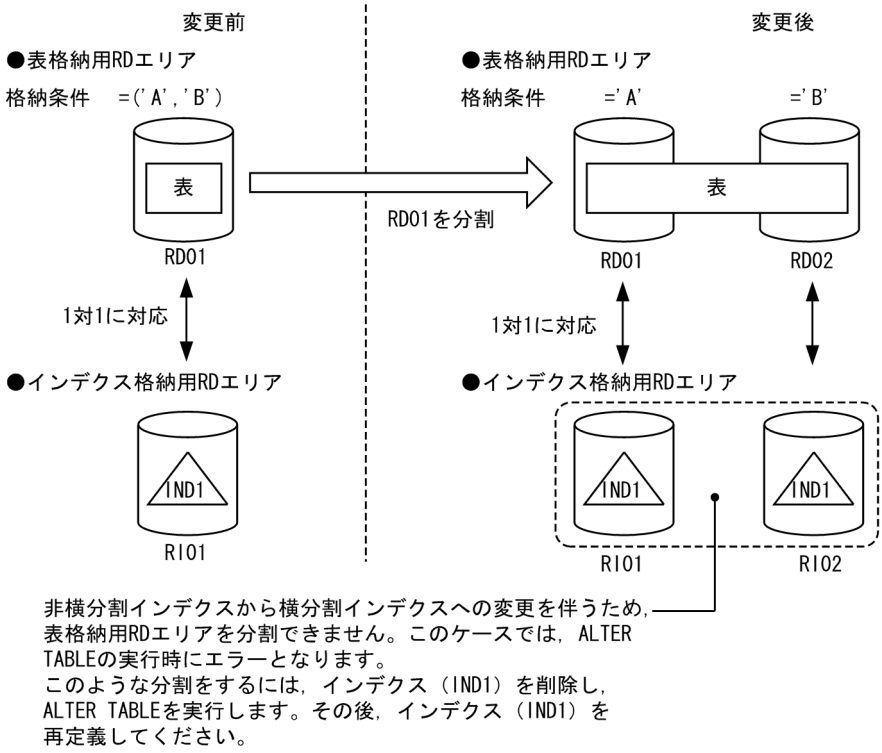


図 13–20　表格納用 RD エリアを分割できないケース（その 2）

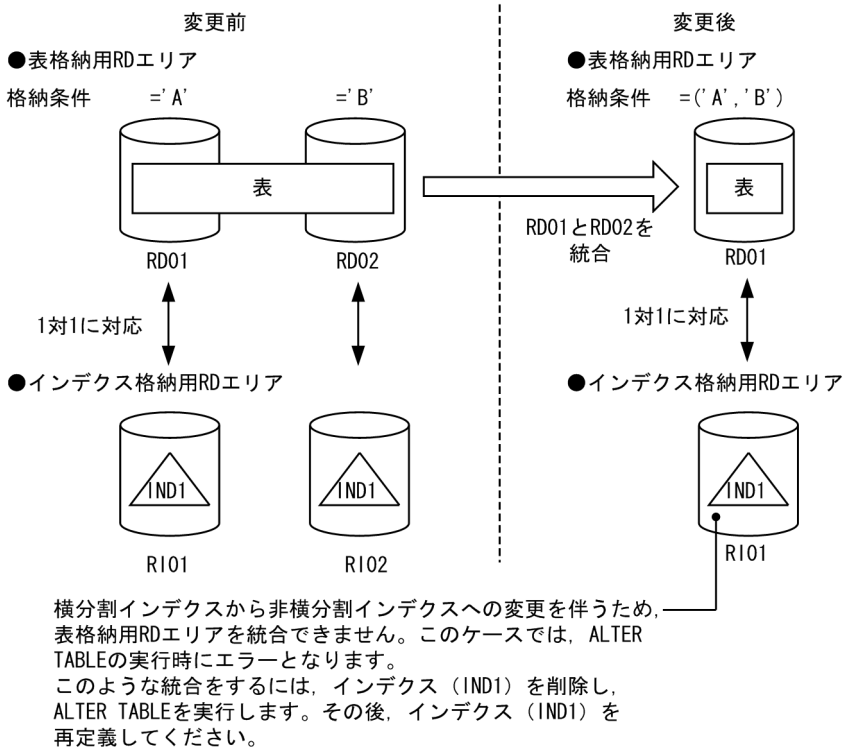


表 13–6　非分割キーインデクスの場合の分割格納条件の変更の適用可否

表格納用 RD エリア		インデクス格納用 RD エリア	
サーバ間横分割	各サーバでのサーバ内横分割	サーバ間横分割あり	サーバ間横分割なし
あり	あり	○	×
	なし	−	○※
なし	あり	○	×
	なし	−	−

（凡例）

○：適用できます。

×：適用できません。

−：該当しません。

注※

複数の分割格納条件を同じ RD エリアに格納している場合，変更できません。

(b)　LOB やインデクスなどの RD エリア

分割格納条件を変更する表に LOB やインデクスなどを格納する RD エリアが定義されている場合，表以外の RD エリアも同時に表と同じように分割，又は統合する必要があります。

(c)　変更後の RD エリア

変更後の RD エリアは，次の用途では使用できません。

- 表及びインデクス格納用 RD エリアとするユーザ用 RD エリア以外の RD エリア
- BLOB 列格納用 RD エリアとするユーザ LOB 用 RD エリア以外の RD エリア
- 共用 RD エリア
- リバランス表，及びリバランス表に関連するリソースを格納している RD エリア

13.12.4　分割格納条件の変更方法（境界値指定の場合）

ALTER TABLE の CHANGE RDAREA で分割格納条件を変更します。なお，分割と統合を同時に実行することはできません。分割と統合を実行する場合は ALTER TABLE を 2 回実行してください。

(1)　分割格納条件の統合前

分割格納条件を統合，分割する前の状態です。

```
CREATE FIX TABLE ”T1”(”C1” INT,”C2” CHAR(10)) PARTITIONED BY ”C1”
 IN((”TA1”)100,(”TA2”)200,(”TA3”)400,(”TA4”)500,(”TA5”)600,(”TA6”))
CREATE INDEX ”I1” ON ”T1”(”C1”)
 IN((”IA1”),(”IA2”),(”IA3”),(”IA4”),(”IA5”),(”IA6”))
```

境界値	100	200	400	500	600	
RDエリア	TA1	TA2	TA3	TA4	TA5	TA6
	IA1	IA2	IA3	IA4	IA5	IA6

(2)　分割格納条件の統合

100 と 200 の境界値を 200 に統合します。

```
ALTER TABLE ”T1” CHANGE RDAREA
 ((100),(200)) INTO ”TA2”
 FOR INDEX ”I1” INTO ”IA2”
```

境界値	200	400	500	600	
RDエリア	TA2	TA3	TA4	TA5	TA6
	IA2	IA3	IA4	IA5	IA6

(3)　分割格納条件の分割

600 より大きい格納範囲を 600 と 700 に分割します。

```
ALTER TABLE ”T1” CHANGE RDAREA
 ((MAX)) INTO ((”TA6”)700,(”TA1”))
 FOR INDEX ”I1” INTO ((”IA6”),(”IA1”))
```

境界値	200	400	500	600	700
RDエリア	TA2	TA3	TA4	TA5	TA6	TA1
	IA2	IA3	IA4	IA5	IA6	IA1

13.12.5　RD エリアの分割（境界値指定の場合）

境界値で分割した表の，特定の格納範囲のデータを複数の RD エリアに分割します。機能の詳細を次に示します。

(1)　上限値

分割機能での上限値を次の表に示します。

表 13–7　分割機能での上限値（境界値指定の場合）

項目	上限値	上限値を超えた場合の動作
分割の対象にできる RD エリア数	1	ALTER TABLE をエラーとします。
1 回の操作で分割できる数	16	
分割した結果の総分割 RD エリア数（重複を含む）	4,096	
分割した結果の総境界値上限数（other を含む）	4,096	

(2)　分割対象 RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された分割対象の境界値によって，その格納条件に合致する RD エリアを分割対象の RD エリアとします。境界値が指定されれば，対象となる RD エリアが特定できるため，RD エリアの指定は必要ありません。ALTER TABLE の指定と分割対象 RD エリアの決定方法を次の表に示します。

表 13–8　ALTER TABLE の指定と分割対象 RD エリアの決定方法（境界値指定の場合）

指定内容	条件	指定可否	分割対象 RD エリアの決定方法
境界値	表定義中に指定された境界値あり	○	指定された境界値を格納している RD エリアを分割対象とします。
	表定義中に指定された境界値なし	×	−
'MAX'	−	○	最大境界値より大きい分割キー値のデータを格納している RD エリアを分割対象とします。

（凡例）

○：指定できます。

×：指定できません。

−：該当しません。

(3)　分割後の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された，変更後の境界値と RD エリアの指定によって，格納する RD エリアを決定します。このとき，分割後の境界値の指定は，昇順になっている必要があり，指定した境界値の最大値は，分割前の境界値と等しくなければなりません。また，分割したすべての境界値が，分割前の格納条件の範囲内でなければなりません。ALTER TABLE の指定値と分割後の RD エリアの決定方法を次の表に示します。表で参照される，分割前後の境界値の条件を図「分割前後の境界値の条件（境界値指定の場合）」に示します。

表 13–9　ALTER TABLE の指定値と分割後の RD エリアの決定方法（境界値指定の場合）

分割前の境界値	分割後の境界値の指定	条件	指定可否	分割対象 RD エリアの決定方法
最小境界値（図の 1.）	境界値	分割後の境界値が最小境界値より小さい値（図の 4.）	○	指定された格納条件と RD エリアを分割後の RD エリアとします。
		分割後の境界値が最小境界値以上の値（図の 1.，2.，3.，5.）	×	ALTER TABLE をエラーとします。
	省略	なし	○	分割された格納条件の中の，最大値の範囲の格納対象を指定された RD エリアとします。
中間境界値又は最大境界値（図の 2.）	境界値	分割後の境界値が分割前の境界値の一つ前の境界値以下（分割前の境界値が 200 の場合：図の 1.，4.）	×	ALTER TABLE をエラーとします。
		分割後の境界値が分割前の境界値より小さく，一つ前の境界値より大きい値（分割前の境界値が 200 の場合：図の 5.）	○	指定された格納条件と RD エリアを分割後の RD エリアとします。
		分割後の境界値が分割前の境界値以上の値（分割前の境界値が 200 の場合：200 以上）	×	ALTER TABLE をエラーとします。
	省略	なし	○	分割された格納条件の中の，最大値の範囲の格納対象を指定された RD エリアとします。
'MAX'（図の 3.）	境界値	分割後の境界値が最大境界値より小さい値（図の 1.，2.，4.，5.）	×	ALTER TABLE をエラーとします。
		分割後の境界値が最大境界値より大きい値（図の 3.）	○	指定された格納条件と RD エリアを分割後の RD エリアとします。

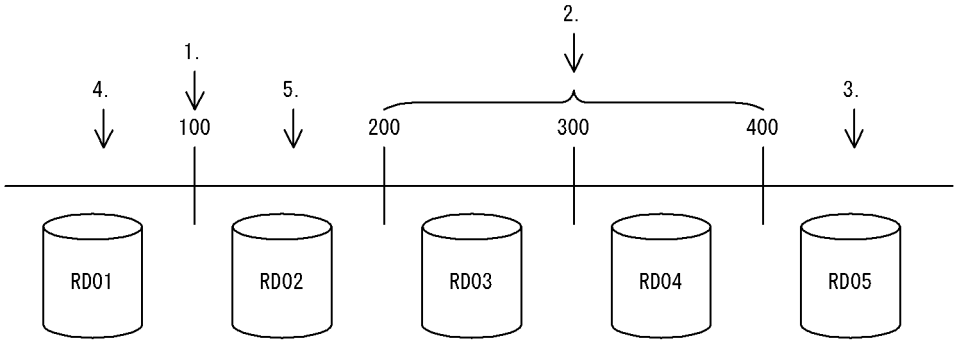
分割前の境界値	分割後の境界値の指定	条件	指定可否	分割対象 RD エリアの決定方法
	省略	なし	○	最大境界値より大きいデータの格納対象を指定された RD エリアとします。

（凡例）

○：指定できます。

×：指定できません。

図 13–21　分割前後の境界値の条件（境界値指定の場合）



分割後の RD エリアは，分割前の RD エリアを再利用しても，新たに用意したものであってもかまいません。また，変更後の RD エリアは複数の格納範囲を同じ RD エリアに格納できます。このとき，指定によっては，システムが自動的に境界値を統合したり，ALTER TABLE をエラーにしたりできます。システムの処置のパターンを次に示します。

1. 指定されたとおりに分割します

   分割した結果，連続した格納範囲を同じ RD エリアに格納しない指定となっている場合，指定されたとおりに分割します。

2. 自動的に格納範囲を統合します

   分割対象の格納範囲の，前後の格納範囲を格納する RD エリアを分割後の RD エリアに指定した場合，次のときはシステムで自動的に連続した境界値を統合します。

   - 分割対象の格納範囲の，直前の格納範囲の RD エリアと分割後の先頭の格納範囲の RD エリアが同じ
   - 分割対象の格納範囲の，直後の格納範囲の RD エリアと分割後の最後の格納範囲の RD エリアが同じ

3. ALTER TABLE をエラーとします

   指定した分割後の RD エリアの中で，連続した複数の境界値を同じ RD エリアに格納するような分割はできません。このとき，ALTER TABLE をエラーとします。このような場合は，ユーザが境界値の範囲を統合して一つの RD エリアに格納するように ALTER TABLE を修正して再実行します。

複数の格納範囲を同じ RD エリアに格納するときのシステムの処置を次の表に示します。

表 13–10　複数の格納範囲を同じ RD エリアに格納するときのシステムの処置

変更後 RD エリアの指定			指定された RD エリアの中で分割対象の格納範囲以外で使用されている RD エリア	分割統合後の結果	システムの処置
指定した分割後の RD エリアだけに着目すると，連続した複数の境界値を同じ RD エリアに格納していない	分割前の表で使われていない RD エリアを指定	−	−	−	指定されたとおりに分割します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 1)」の例 1−1 が該当します。
	分割前の表で使われている RD エリアを指定	分割対象の格納範囲でしか使用されていない	−	−	指定されたとおりに分割します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 1)」の例 1−2 が該当します。
		分割対象の格納範囲以外で使用されている	分割後の先頭に指定	分割対象の格納範囲の直前の RD エリアと同じ	直前の境界値と格納範囲を統合します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 2」の例 2−1 が該当します。
				分割対象の格納範囲の直前の RD エリアと異なる	指定されたとおりに分割します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 1)」の例 1−3 が該当します。
			分割の中間に指定	−	指定されたとおりに分割します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 2)」の例 1−4 が該当します。

変更後 RD エリアの指定			指定された RD エリアの中で分割対象の格納範囲以外で使用されている RD エリア	分割統合後の結果	システムの処置
			分割の最後に指定	分割対象の格納範囲の直後の RD エリアと同じ	直後の境界値と格納範囲を統合します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 2」の例 2−2 が該当します。
				分割対象の格納範囲の直後の RD エリアと異なる	指定されたとおりに分割します。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 2)」の例 1−5 が該当します。
指定した分割後の RD エリアだけに着目すると，連続した複数の境界値を同じ RD エリアに格納している	−	−	−	−	ALTER TABLE をエラーとします。 図「複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 3」の例 3−1 が該当します。

（凡例）−：該当しません。

図 13–22　複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 1）

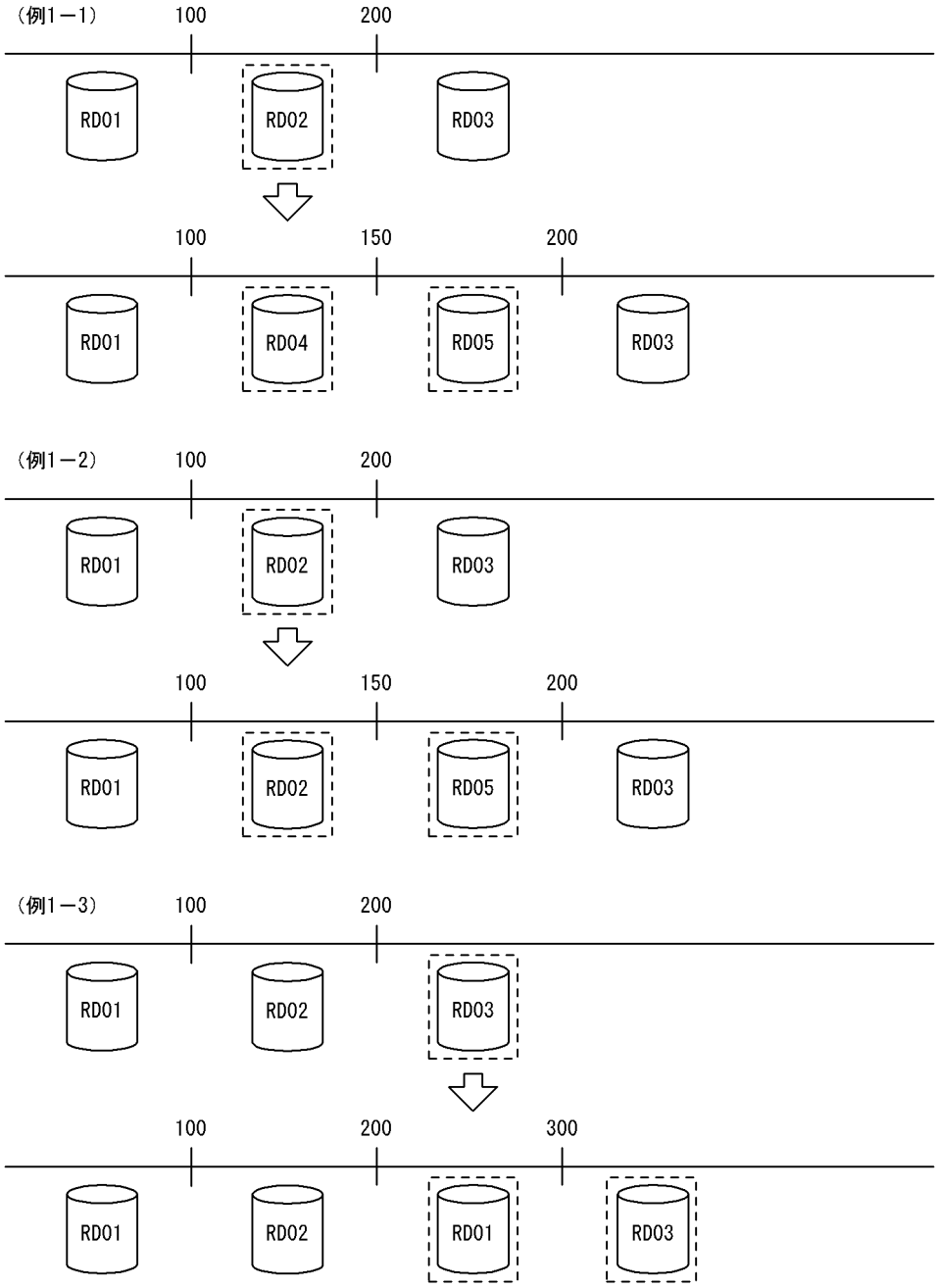


図 13–23　複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 1（その 2）

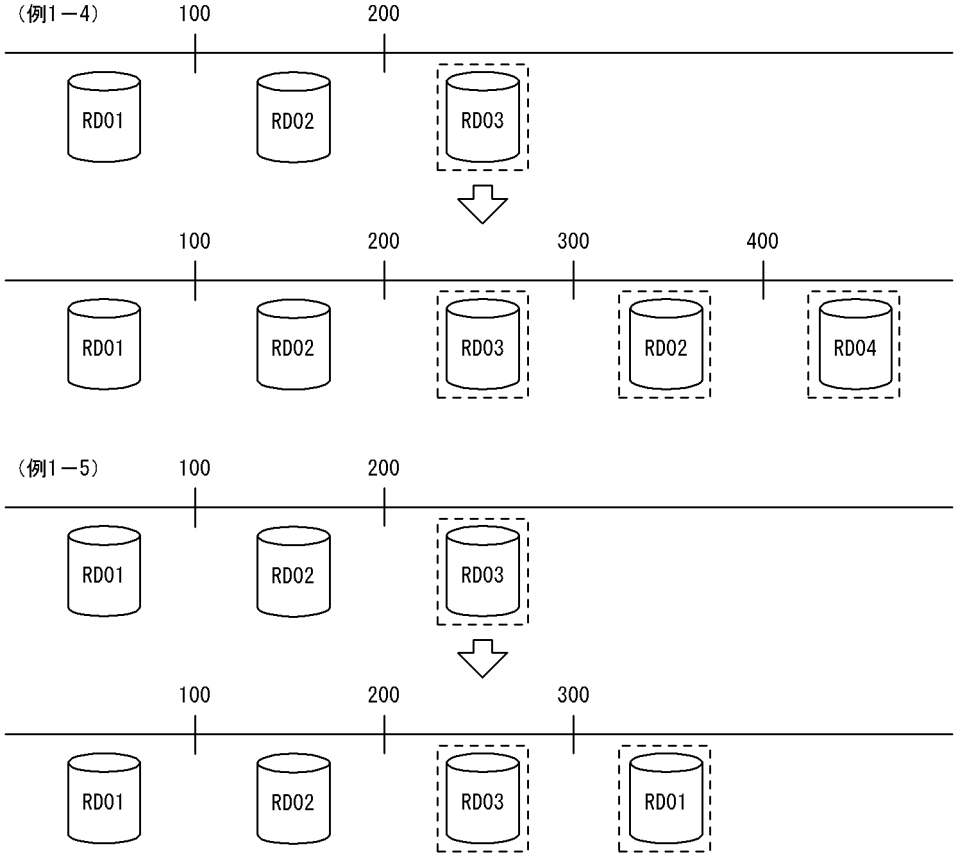


図 13–24　複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 2

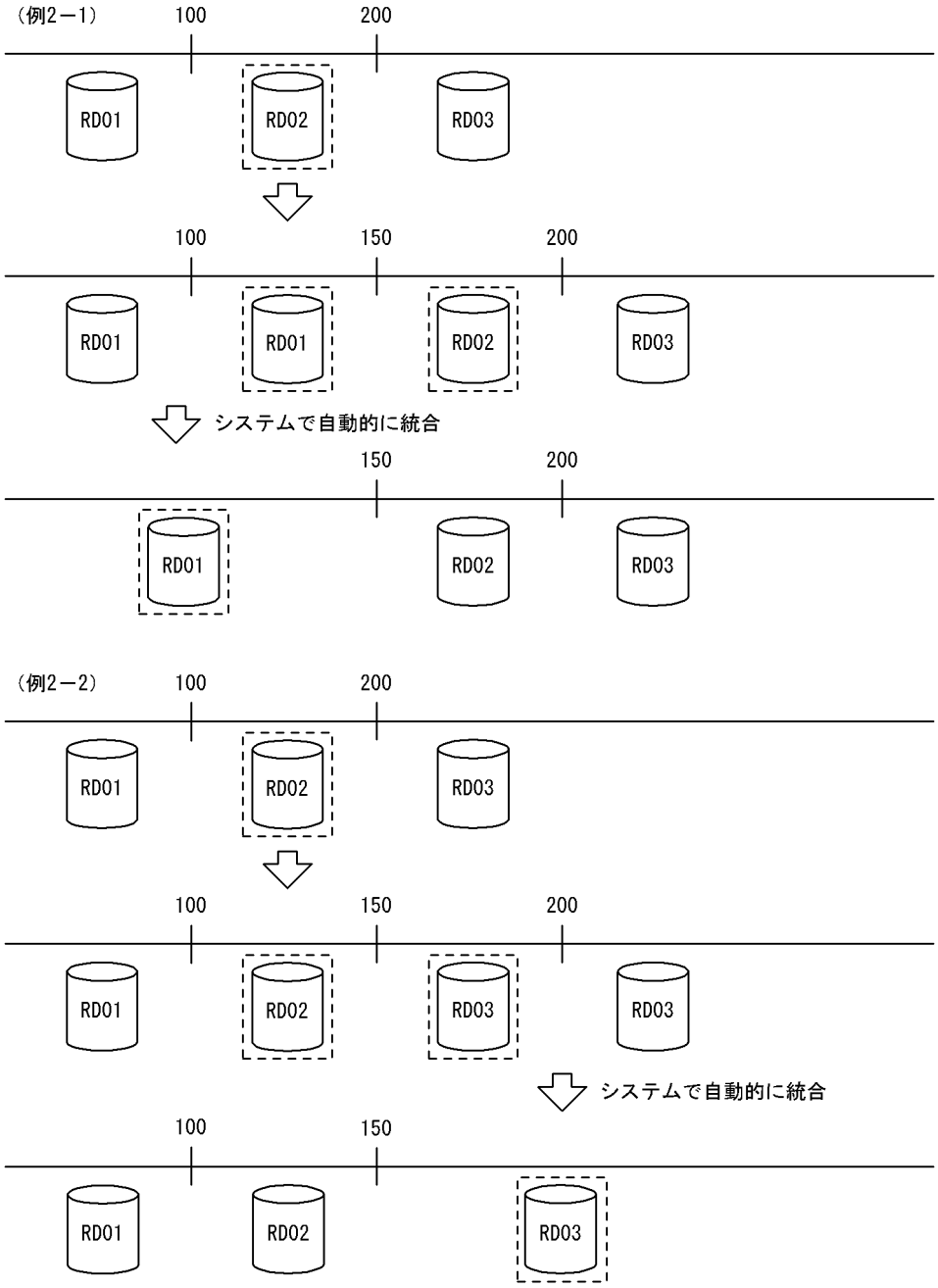
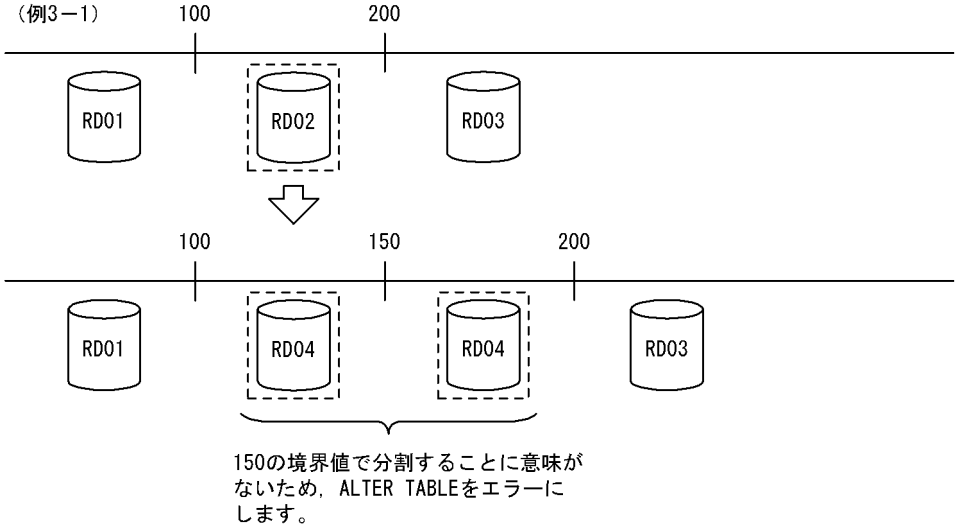


図 13–25　複数の格納範囲を同じ RD エリアに格納するときのシステムの処置の例 3



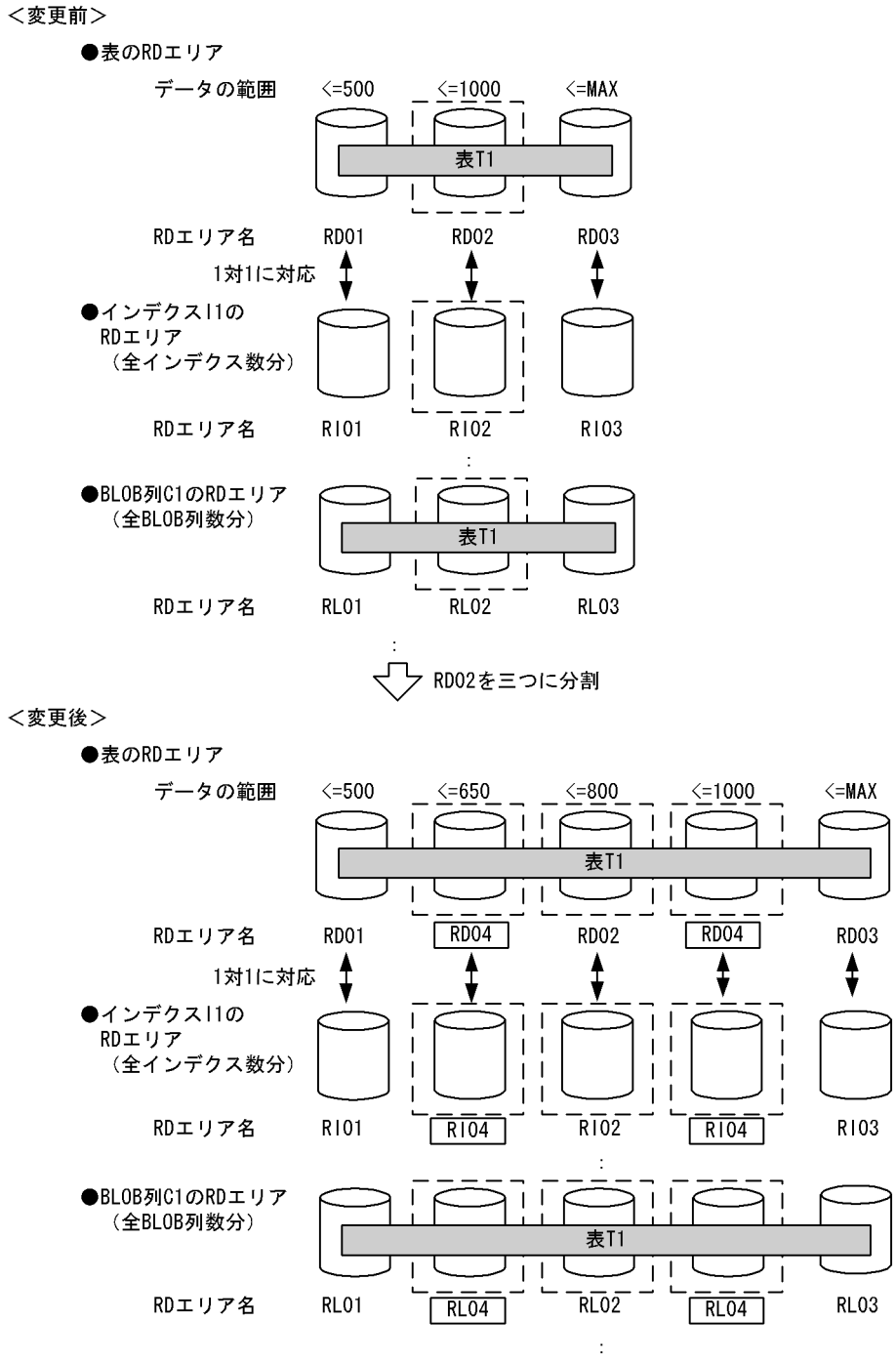
(4)　表と表以外の RD エリアの対応

分割格納条件を変更する表に分割キーインデクスなどが定義されている場合，インデクスなどのデータを RD エリアに格納する必要があるため，表格納用 RD エリアと同様に分割しなければなりません。表と表以外の RD エリアの指定方法（境界値分割での格納条件の分割）を次の表に示します。表に示すリソースが複数定義されている場合はすべてが対象となります。正しく指定されていない場合，ALTER TABLE をエラーとします。表と表以外の RD エリアの対応の例を次の図に示します。この例では，表と表以外の RD エリアを三つに分割しています。

表 13–11　表と表以外の RD エリアの指定方法（境界値分割での格納条件の分割）

リソース名		指定方法
列	BLOB 列	表格納用 RD エリアと 1 対 1 で指定します。 表格納用 RD エリアが重複指定されている場合は，表に対応するよう重複指定します。また，既存の表格納用 RD エリアを変更後に使用する場合は，同じ境界値に対応するように既存のインデクス，LOB 格納用 RD エリアを指定しなければなりません。
インデクス	クラスタキーインデクス プライマリキーインデクス（プライマリキー及びクラスタキーを定義したインデクスを含む） B-tree インデクス	

図 13–26　表と表以外の RD エリアの対応の例



(5)　分割する RD エリア内のデータの扱い

境界値による格納範囲を分割する場合，基本的にはシステムが自動的に RD エリアの該当する表のデータを削除します。ただし，条件によってはデータを削除しないで残すこともできます。

1. データの削除

   境界値による格納範囲を分割する場合，分割する RD エリア内のデータの一部は，分割後はその RD エリアに格納される格納範囲外のデータとなってしまう可能性があるため，分割対象とした格納範囲の

RD エリア内のデータは削除します。ただし，削除するのは分割格納条件を変更する表のデータだけです（同じ RD エリアに含まれる別の表のデータは削除されません）。RD エリア内のデータを削除する場合，次のどちらかの方法でデータを削除します。

- すべての定義情報の削除

  分割前の RD エリアが，分割した結果，分割対象の表で使用されなくなった場合，ディクショナリ中の，その表が格納条件ごとに使用する RD エリアの情報（SQL_DIV_TABLE 表）から，分割前の RD エリアの情報を削除します。また，RD エリア内で管理している，表の情報も削除します。この結果，その RD エリア内にあった，分割対象の表のデータはすべて削除されます。イメージ的には特定 RD エリアに対して DROP TABLE を実行するのと同じです。

- データだけの削除

  分割前の RD エリアを分割後も分割対象の表で使用する場合，ディクショナリの情報や RD エリア内で管理している情報は削除しないで，RD エリア内の分割対象とした表のデータだけを削除します。表のデータを削除する場合，分割対象の格納範囲の RD エリアがほかの格納範囲で使用されていると，その格納範囲のデータも削除します。イメージ的には特定 RD エリアに対して PURGE TABLE を実行するのと同じです。

なお，RD エリア内のデータを削除する場合，対応する次の RD エリア内のデータもすべて削除します。

- インデクス格納用 RD エリア内のインデクスキー
- BLOB 列格納用 RD エリア内のデータ

2. データの保存

「1.データの削除」で示したとおり，境界値による格納条件を分割する場合，基本的には分割前の RD エリア内のデータは削除します。しかし，次の条件をすべて満たす場合は，RD エリア内のデータをそのまま使用できるため，データを削除しないようにすることもできます。

- 分割前の RD エリアを分割後の RD エリアの一部としてそのまま使用する場合
- 分割前の RD エリア内にある境界値以下のデータしかない場合
- 分割した結果，分割前の RD エリアのすべてのデータが分割後の格納範囲と一致している場合

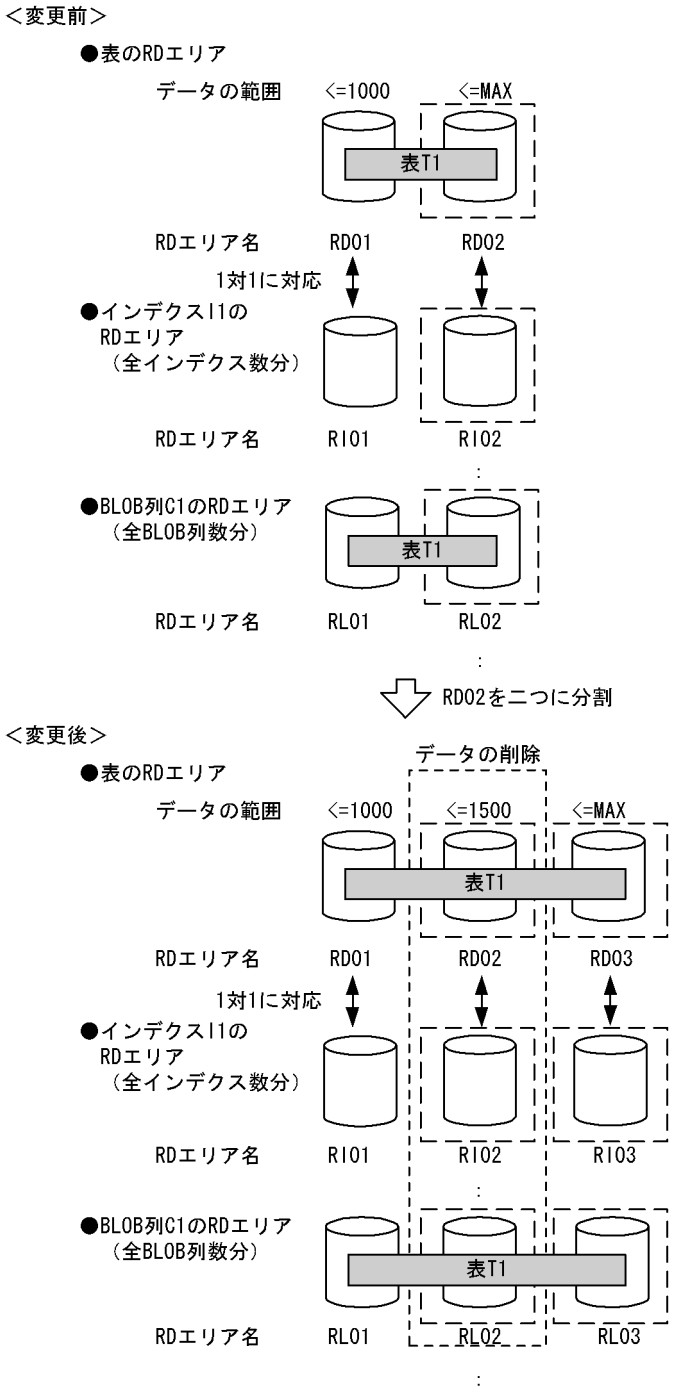
RD エリア内のデータを削除するかどうかは，ALTER TABLE の WITHOUT PURGE 句の指定によって選択できます。WITHOUT PURGE 句の指定とデータの扱いを次の表に示します。ただし，システムでは，分割した結果，RD エリア内のすべてのデータが分割後の格納範囲と一致しているかどうかをチェックしません。

表 13–12　WITHOUT PURGE 句の指定とデータの扱い（境界値指定の場合）

変更前後の RD エリアの関係	WITHOUT PURGE 句の指定可否	WITHOUT PURGE 句の指定あり	WITHOUT PURGE 句の指定なし
分割前の RD エリアを分割後の RD エリアに含みます。	指定できます。	分割前の RD エリアのデータを削除しません。	分割前の RD エリアのデータを削除します。
分割前の RD エリアを分割後の RD エリアに含みません。	指定できません。	該当しません。	

WITHOUT PURGE 指定なしの場合のデータの削除対象 RD エリアの例を次の図に示します。この例では，WITHOUT PURGE を指定していないため，分割前の RD エリアのデータは削除されます。

図 13–27　WITHOUT PURGE 指定なしの場合のデータの削除対象 RD エリアの例



データを削除しない場合の注意事項

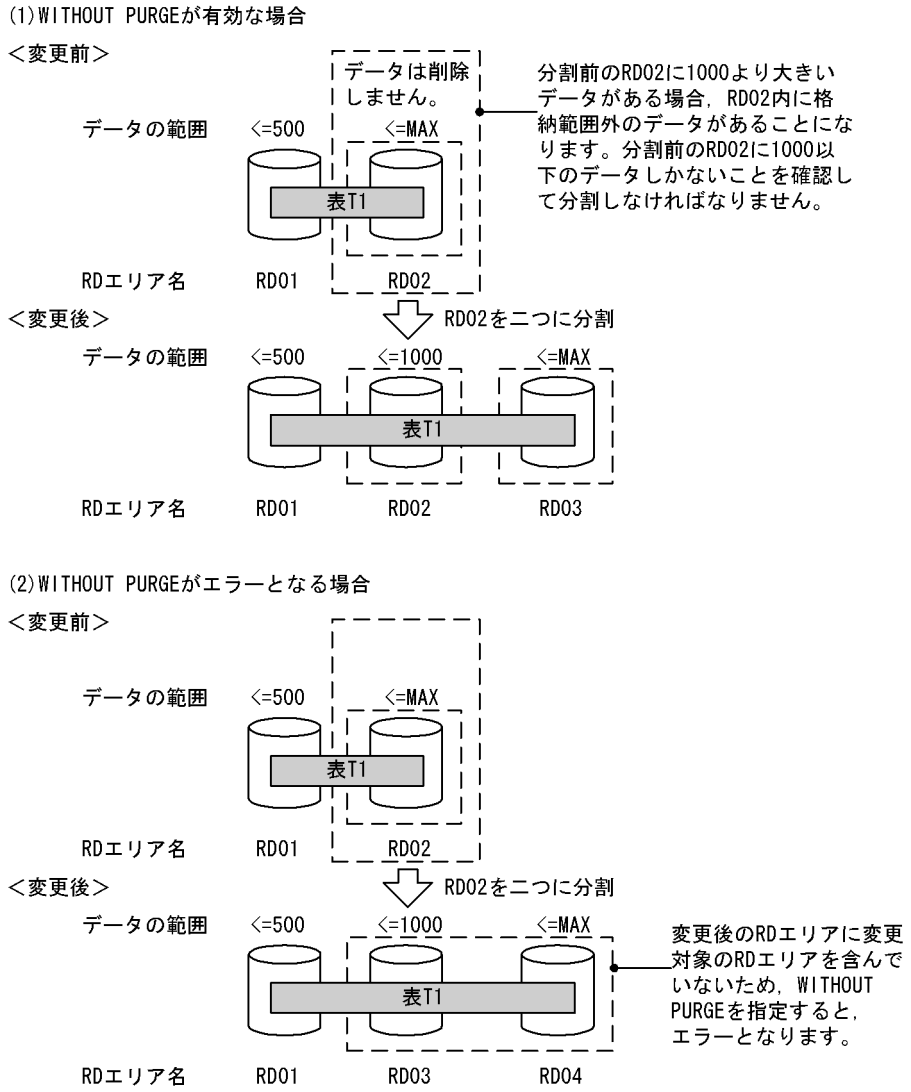
分割前の RD エリアのデータが，分割格納条件を変更した結果，格納範囲以外となった場合，ALTER TABLE 実行時はデータの妥当性をチェックしません。したがって，格納範囲外のデータが RD エリアにあることになり，SQL 実行時など HiRDB が正しく動作しません。

このため，WITHOUT PURGE を指定して分割格納条件を変更する場合は，十分に注意する必要があります。分割前の RD エリアのデータが分割後の格納範囲となることが保証できない場合は，分割前の

RD エリアのデータをアンロードし，WITHOUT PURGE 指定なしで分割格納条件を変更し，その後アンロードしていたデータを分割した RD エリアにロードする必要があります。誤って分割した場合の回復方法については，「分割後の格納条件に合わないデータが残った場合の回復手順」を参照してください。

変更対象 RD エリアのデータの扱い（WITHOUT PURGE 指定がある場合）の例を次の図に示します。この例では，WITHOUT PURGE が有効な場合，分割前の RD エリアは削除されないで，再利用されます。

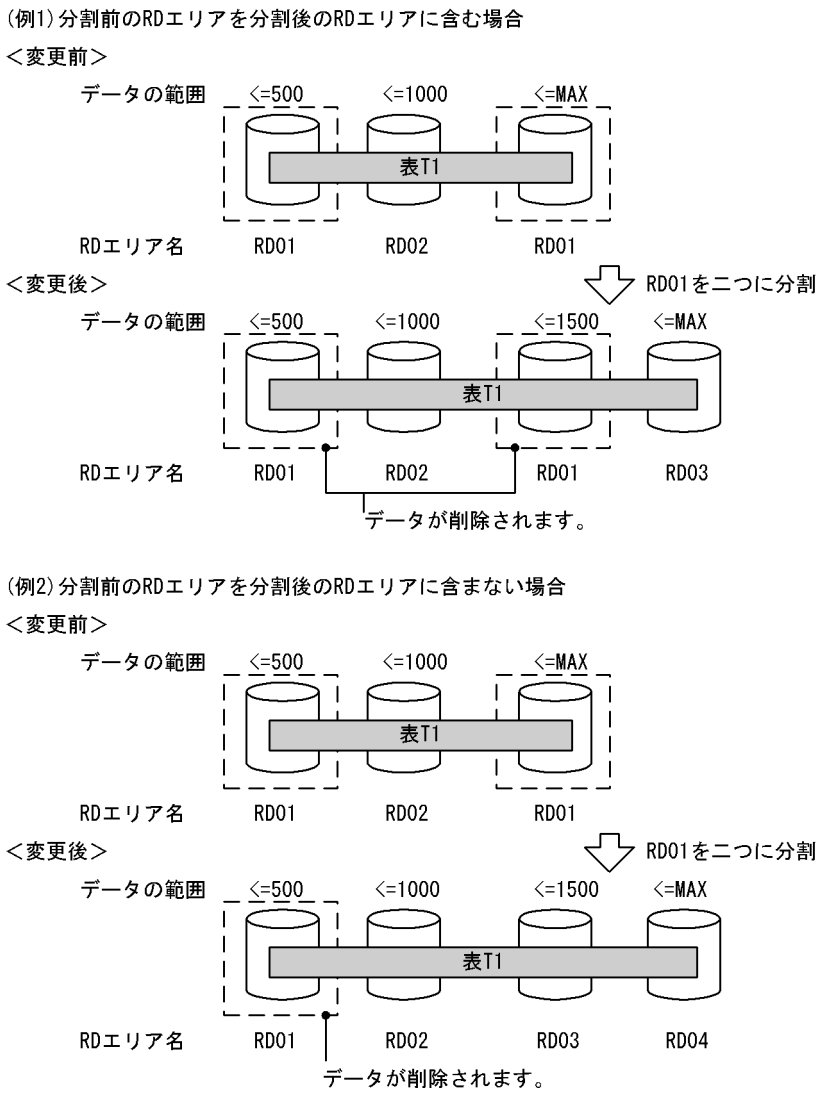
図 13–28　変更対象 RD エリアのデータの扱い（WITHOUT PURGE 指定がある場合）の例



データを削除する場合の注意事項

分割前の RD エリアが，ほかの格納範囲でも格納されるような指定の場合，ほかの格納範囲のデータも削除します。分割前の RD エリアを分割後の RD エリアに含む場合も，含まない場合も同じです。RD エリアのデータを削除する場合の例を次の図に示します。

図 13–29　RD エリアのデータを削除する場合の例



13.12.6　RD エリアの統合（境界値指定の場合）

(1)　上限値と下限値

統合機能での上限値と下限値を次の表に示します。

表 13–13　統合機能での上限値と下限値（境界値指定の場合）

項目	上限値と下限値	上限値と下限値を超えた場合の動作
統合の対象にできる RD エリア数	2～16（下限値～上限値）	ALTER TABLE がエラーとなります。
1 回の操作での統合後の RD エリア数	1（固定値）	
統合後の表の RD エリアの総数	2（下限値）	

(2)　統合対象の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された統合対象の複数の境界値によって，その格納条件に合致するすべての RD エリアを統合対象の RD エリアとします。境界値が指定されれば，対象となる RD エリアが特定できるため，RD エリアの指定は必要ありません。このとき，複数の境界値の指定は，昇順である必要があります。また，表に定義されている格納条件の順番に指定されている必要があります。例えば，表定義に境界値が 10，20，30，40 と定義されている場合，10，30，40 など間を飛ばして指定すると ALTER TABLE をエラーとします。

ALTER TABLE の指定と統合対象 RD エリアの決定方法を次の表に示します。

表 13–14　ALTER TABLE の指定と統合対象 RD エリアの決定方法（境界値指定の場合）

指定内容	条件 1	条件 2	動作
境界値	表定義中に指定された境界値あり	定義されている境界値の順に指定されている	指定された格納条件に合う RD エリアを統合対象とします。
		定義されている境界値の順に指定されていない	ALTER TABLE をエラーとします。
	表定義中に指定された境界値なし	なし	
'MAX'	直前に指定された境界値が表定義中の最大境界値である	なし	最大境界値より大きい分割キー値のデータを格納している RD エリアを統合対象とします。
	直前に指定された境界値が表定義中の最大境界値でない	なし	ALTER TABLE をエラーとします。

(3)　統合後の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された，統合後の RD エリアが統合前の複数の格納条件をすべて格納する RD エリアとなります。統合された後の境界値は，統合されるすべての境界値をマージしたものとなるため，指定は必要ありません。例えば，10，20，30，40 という境界値があった場合，20 と 30 を統合すると，境界値は，10，30，40 となり，10 より大きく，30 以下のすべてのデータが統合後の RD エリアの格納条件値となります。

統合後の RD エリアは，統合前の RD エリアの中の一つであっても，新たに用意したものであってもかまいません。統合後の RD エリアの指定可否を次の表に示します。

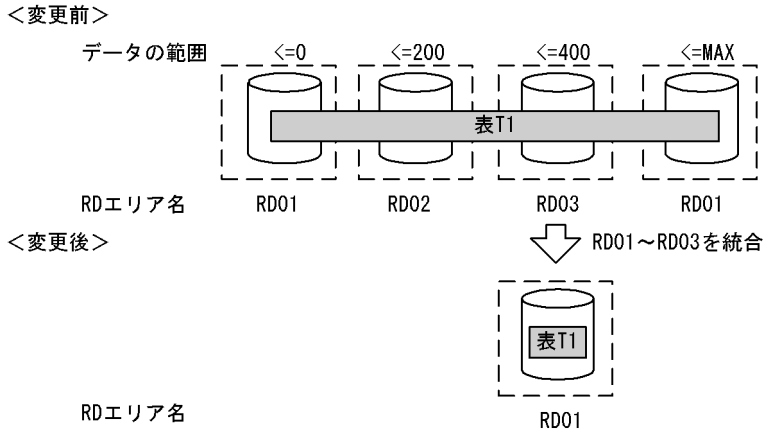
表 13–15　統合後の RD エリアの指定可否（境界値指定の場合）

統合後の RD エリア		指定可否
統合前の RD エリアの中の一つ	−	指定できます。
統合前の RD エリアの中にない	統合対象の境界値以外で使用されている RD エリア	指定できます。
	統合対象の表では使用されていなかった RD エリア（新たに用意された RD エリア）	指定できます。

（凡例） －：該当しません。

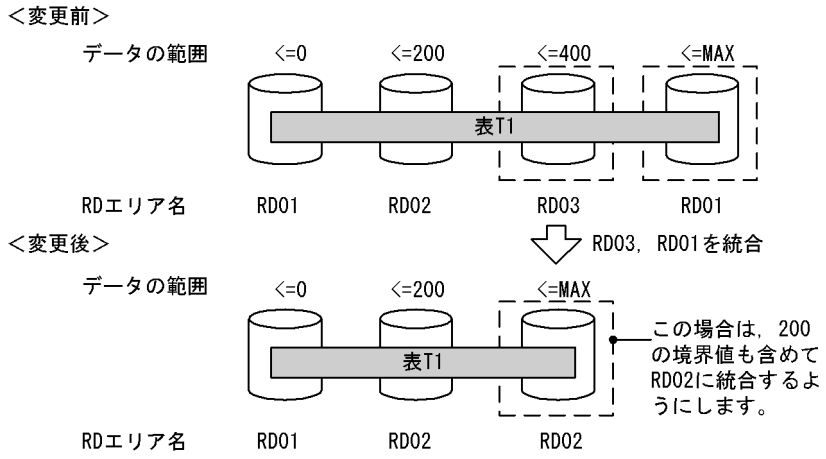
統合後の RD エリアは複数の格納条件に対して，同じ RD エリアに格納するように統合できます。ただし，統合した結果，一つの RD エリアにすべてのデータが格納されるような統合はできません。統合できない例（一つの RD エリアにすべてのデータを統合）を次の図に示します。

図 13–30　統合できない例（一つの RD エリアにすべてのデータを統合）



また，統合後の RD エリアが，前後の格納範囲の RD エリアと同一となるような統合はできません。この場合は，前後の格納範囲の RD エリアも含めて統合します。統合できない例（前後の格納範囲の RD エリアと同一となるような統合）を次の図に示します。

図 13–31　統合できない例（前後の格納範囲の RD エリアと同一となるような統合）



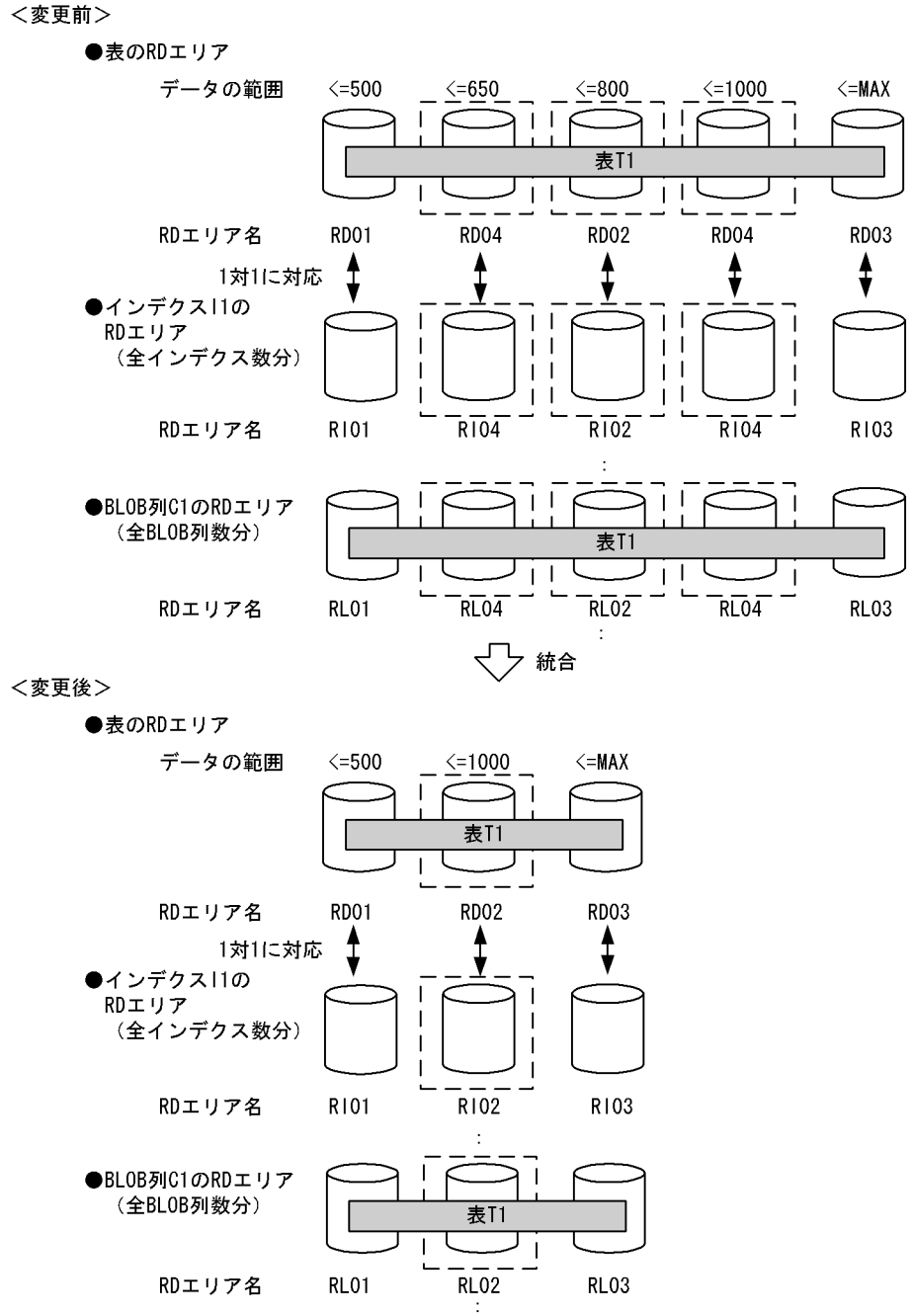
(4)　表と表以外の RD エリアの対応

分割格納条件を変更する表に分割キーインデクスなどが定義されている場合，表格納用 RD エリアと対になる RD エリアにインデクスなどのデータを格納する必要があります。表と表以外の RD エリアの指定方法（境界値分割での格納条件の統合）を次の表に示します。この表に示すリソースが複数定義されている場合は，すべてが対象となります。正しく指定されていない場合，ALTER TABLE をエラーとします。表と表以外の RD エリアの対応を次の図に示します。

表 13–16　表と表以外の RD エリアの指定方法（境界値分割での格納条件の統合）

リソース名		指定方法
列	BLOB 列	表格納用 RD エリアと 1 対 1 で指定します。 表格納用 RD エリアが重複指定されている場合は，表に対応するよう重複指定します。また，既存の表格納用 RD エリアを変更後に使用する場合は，同じ境界値に対応するように既存のインデクス，LOB 格納用 RD エリアを指定しなければなりません。
インデクス	クラスタインデクス プライマリインデクス（プライマリクラスタインデクスを含む） B-tree インデクス	

図 13–32　表と表以外の RD エリアの対応



(5)　統合する RD エリアのデータの扱い

境界値による格納範囲を統合する場合，基本的には，システムが自動的に RD エリアのデータを削除します。ただし，条件によっては，データを削除しないで残すこともできます。

1. データの削除

   境界値による格納範囲を統合する場合，統合前の RD エリアは，統合後はその表が使用する RD エリアではなくなる可能性があるため，RD エリア内のデータはシステムが自動的に削除します。ただし，削除するのは分割格納条件を変更する表のデータだけです（同じ RD エリアに含まれる別の表のデータは削除されません）。RD エリア内のデータを削除する場合，次のどちらかの方法でデータを削除します。

   - すべての定義情報の削除

     統合前の RD エリアが，統合した結果，統合対象の表で使用されなくなった場合，ディクショナリ中の，その表が格納条件ごとに使用する RD エリアの情報（SQL_DIV_TABLE 表）から，統合前の RD エリアの情報を削除します。また，RD エリア内で管理している，表の情報も削除します。この結果，その RD エリア内にあった，統合対象の表のデータはすべて削除されます。イメージ的には特定の RD エリアに DROP TABLE を実行するのと同じです。

   - データだけの削除

     統合前の RD エリアを統合後も統合対象の表で使用する場合，ディクショナリの情報や RD エリア内で管理している情報は削除しないで，RD エリア内の統合対象とした表のデータだけを削除します。表のデータを削除する場合，統合対象の格納範囲の RD エリアがほかの格納範囲で使用されていると，その格納範囲のデータも削除します。イメージ的には特定 RD エリアに対して PURGE TABLE を実行するのと同じです。

   なお，RD エリア内のデータを削除する場合，対応する次の RD エリア内のデータもすべて削除します。

   - インデクス格納用 RD エリア内のインデクスキー
   - BLOB 列格納用 RD エリア内のデータ

2. データの保存

   「1.データの削除」で示したとおり，境界値による格納条件を統合する場合，基本的には統合前の RD エリア内のデータは削除します。しかし，次の条件を満たす場合は，RD エリア内のデータをそのまま使用できるため，データを削除しないようにすることもできます。

   - 統合前の RD エリアを統合後の RD エリアの一部として，そのまま使用する場合

   RD エリア内のデータを削除するかどうかは，ALTER TABLE の WITHOUT PURGE 句の指定によって選択できます。WITHOUT PURGE 句の指定とデータの扱いを次の表に示します。

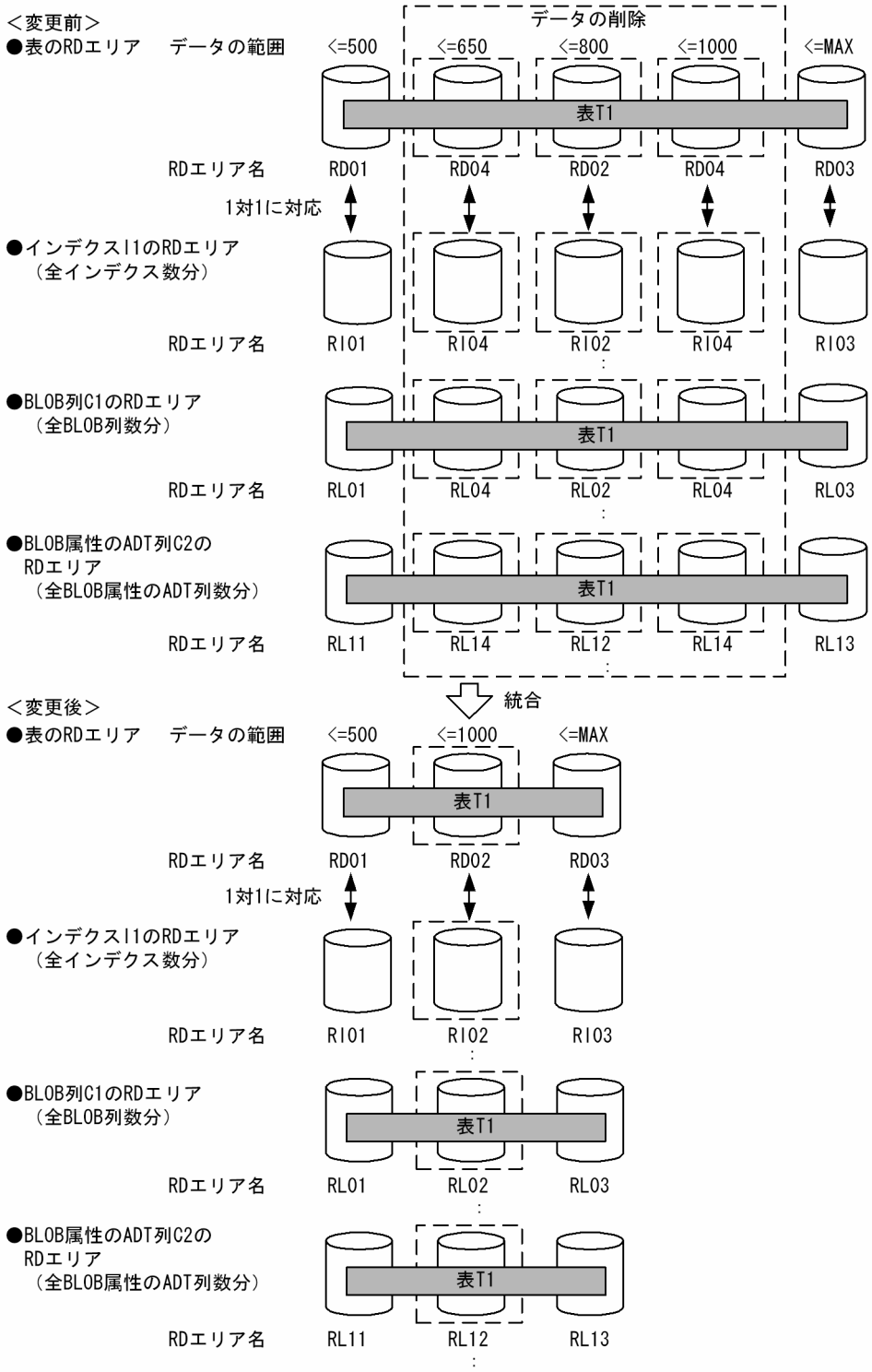
表 13–17　WITHOUT PURGE 句の指定とデータの扱い（境界値指定の場合）

変更前後の RD エリアの関係	WITHOUT PURGE 句の指定可否	WITHOUT PURGE 句の指定あり	WITHOUT PURGE 句の指定なし
統合前の RD エリアの一つを統合後の RD エリアとします。	指定できます。	統合後の RD エリアのデータを削除しません。統合前のそのほかの RD エリアはそのすべてのデータを削除します。	統合前のすべての RD エリアのデータを削除します。

変更前後の RD エリアの関係	WITHOUT PURGE 句の指定可否	WITHOUT PURGE 句の指定あり	WITHOUT PURGE 句の指定なし
統合前の RD エリアを統合後の RD エリアにしません（統合前の RD エリアとは別の RD エリアを統合後の RD エリアとします）。	指定できません。	ALTER TABLE をエラーとします。	

統合時のデータの削除対象 RD エリアを次の図に示します。

図 13–33　統合時のデータの削除対象 RD エリア



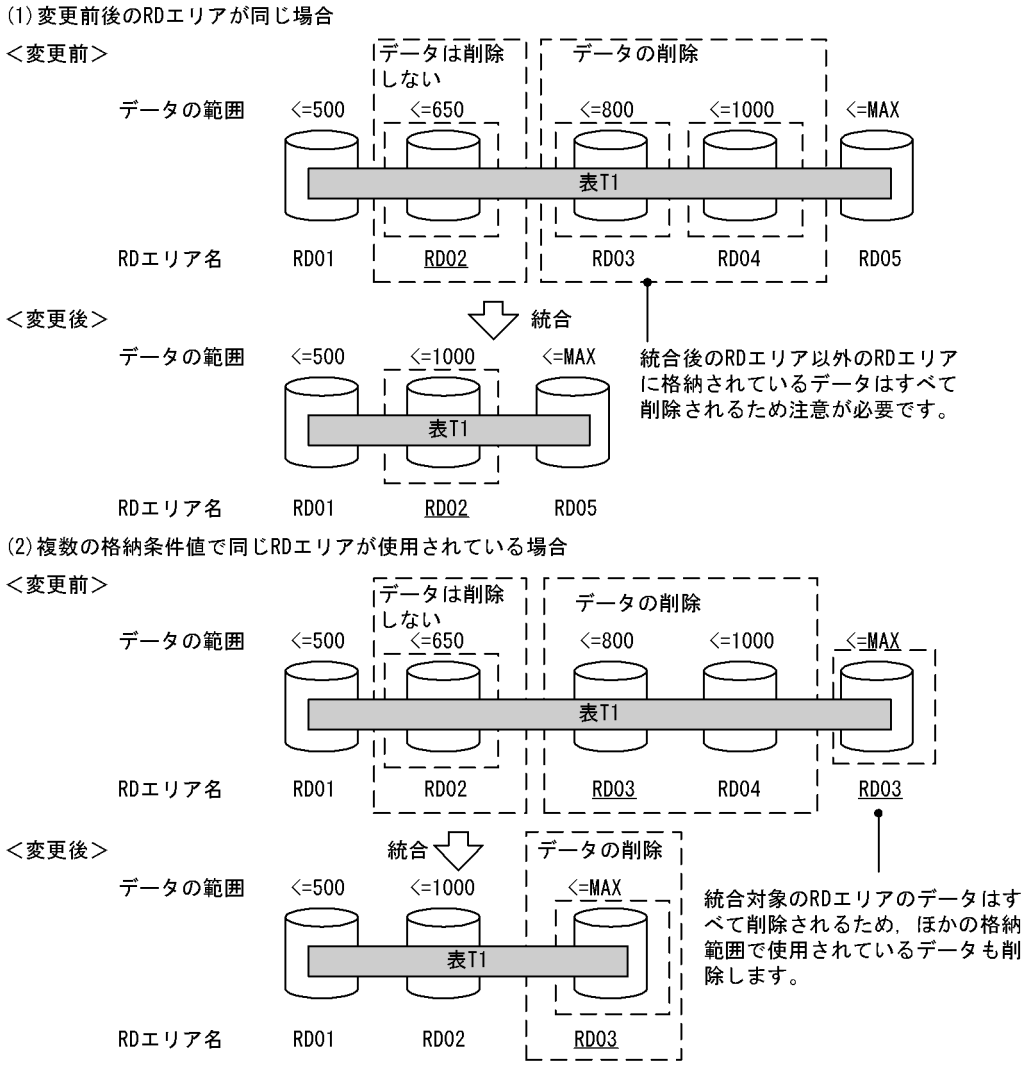
3. データを削除しない場合の注意事項

WITHOUT PURGE を指定してデータを保存した場合，統合前のすべてのデータをアンロードし，統合後の RD エリアにロードすると，保存したデータが二重に登録されてしまいます。このため，WITHOUT PURGE を指定してデータを保存する RD エリアのデータはアンロード，及び統合後のロードを実行してはなりません。

WITHOUT PURGE を指定しても統合前の RD エリアの中で統合後の RD エリアとしなかった RD エリアのデータはすべて削除します。また，統合する格納条件以外でも使用されている RD エリアが統合後の RD エリアではなかった場合，ほかの格納条件値のデータも削除します。このため，複数の格納範囲で使用されている RD エリアの中で特定の格納範囲を統合する場合は，WITHOUT PURGE を指定しないで分割格納条件を変更し，その後アンロードしていたデータを統合した RD エリアにロードする必要があります。

変更対象 RD エリアのデータの扱い（WITHOUT PURGE 指定がある場合）を次の図に示します。

図 13–34　変更対象 RD エリアのデータの扱い（WITHOUT PURGE 指定がある場合）



4. データを削除する場合の注意事項

データを削除しない場合の注意事項と同じように，統合する格納条件以外でも使用されている RD エリアが，統合後の RD エリアではなかった場合，ほかの格納範囲のデータも削除します。このため，統合する格納範囲以外でも統合対象の RD エリアを使用している場合は，統合対象の RD エリアのデータをすべてアンロードし，WITHOUT PURGE を指定しないで分割格納条件を変更し，その後アンロードしていたデータを統合した RD エリアにデータロードすることを推奨します。

13.12.7　分割格納条件の変更方法（格納条件指定の場合）

ALTER TABLE の CHANGE RDAREA で分割格納条件を変更します。なお，分割と統合を同時に実行することはできません。分割と統合を実行する場合は ALTER TABLE を 2 回実行してください。

(1)　分割格納条件の統合前

分割格納条件を統合，分割する前の状態です。

```
CREATE FIX TABLE "T1"("C1" CHAR(10),"C2"...
 IN(("TA1")"C1"='横浜支店',("TA2")"C1"='関内支店',("TA3")"C1"='川崎支店',
    ("TA4")"C1"=('東京支店','品川支店'),("TA5"))
CREATE INDEX "I1" ON "T1"("C1")
 IN(("IA1"),("IA2"),("IA3"),("IA4"),("IA5"))
```

キー条件	C1='横浜支店'	C1='関内支店'	C1='川崎支店'	C1='東京支店' 又は C1='品川支店'	その他
RDエリア	TA1	TA2	TA3	TA4	TA5
	IA1	IA2	IA3	IA4	IA5

(2)　分割格納条件の統合

横浜支店のデータと関内支店のデータを同じ RD エリアに統合します。

```
ALTER TABLE "T1" CHANGE RDAREA PARTITIONED CONDITION (("TA1"),("TA2")) INTO ("TA1")
 FOR INDEX "I1" INTO "IA1"
```

キー条件	C1='横浜支店' 又は C1='関内支店'	C1='川崎支店'	C1='東京支店' 又は C1='品川支店'	その他
RDエリア	TA1	TA3	TA4	TA5
	IA1	IA3	IA4	IA5

(3)　分割格納条件の分割

東京支店のデータと品川支店のデータを別々の RD エリアに分割します。

```
ALTER TABLE "T1" CHANGE RDAREA PARTITIONED CONDITION
 (("TA4")) INTO (("TA4")"C1"=('東京支店'),("TA6")"C1"=('品川支店'))
 FOR INDEX "I1" INTO (("IA4"),("IA6"))
```

キー条件	C1='横浜支店' 又は C1='関内支店'	C1='川崎支店'	C1='東京支店'	C1='品川支店'	その他
RDエリア	TA1	TA3	TA4	TA6	TA5
	IA1	IA3	IA4	IA6	IA5

13.12.8　RD エリアの分割（格納条件指定の場合）

(1)　格納条件の変更規則

格納条件の変更規則について説明します。

(a)　上限値

格納条件を変更するときは，次の表に示す上限値の規則を守ってください。

表 13–18　分割機能での上限値（格納条件指定の場合）

項目	上限値	上限値を超えた場合の動作
分割の対象にできる RD エリア数	1	ALTER TABLE をエラーとします。
1 回の操作で分割できる数	16	
分割した結果の総分割 RD エリア数（重複を含む）	4,096	
分割後の総格納条件数（格納条件を指定していない RD エリアも含む）	15,000	

(b)　分割時の規則

分割対象となる RD エリアは一定の条件を満たしている必要があります。RD エリアの分割可否を次の表に示します。

表 13–19　RD エリアの分割可否

分割対象 RD エリア※1	表に定義した格納条件	分割可否	説明
格納条件を指定した RD エリアの場合	分割対象 RD エリアに格納条件を一つだけ指定している	×	ALTER TABLE がエラーになります。
	分割対象 RD エリアに格納条件を二つ以上指定している	○	格納条件を指定した RD エリアを分割する例を図「格納条件を指定した RD エリアを分割する例」に示します。
格納条件を指定していない RD エリアの場合	−	○	格納条件を指定していない RD エリアを分割する例を図「格納条件を指定していない RD エリアを分割する例」に示します。
OTHERS 指定※2の RD エリアの場合	格納条件が全 RD エリアに指定されている	○	OTHERS 指定の RD エリアを分割する例を図「OTHERS 指定の RD エリアを分割する例」に示します。
	格納条件なしの RD エリアがある	×	ALTER TABLE がエラーになります。 OTHERS 指定の RD エリアを分割できない例を図「OTHERS 指定の RD エリアを分割できない例」に示します。

（凡例）

○：分割できます。

×：分割できません。

−：該当しません。

注※1

ALTER TABLE の CHANGE RDAREA の変更前 RD エリア情報リストに指定された RD エリアが分割対象 RD エリアになります。

注※2

OTHERS 指定の RD エリアについては，「OTHERS 指定の RD エリア」を参照してください。

図 13–35　格納条件を指定した RD エリアを分割する例

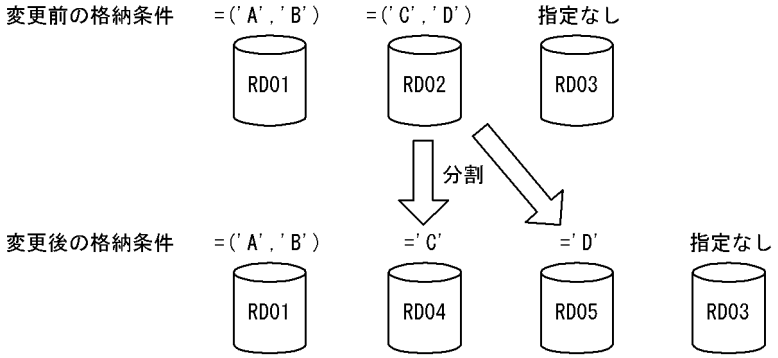


図 13–36　格納条件を指定していない RD エリアを分割する例

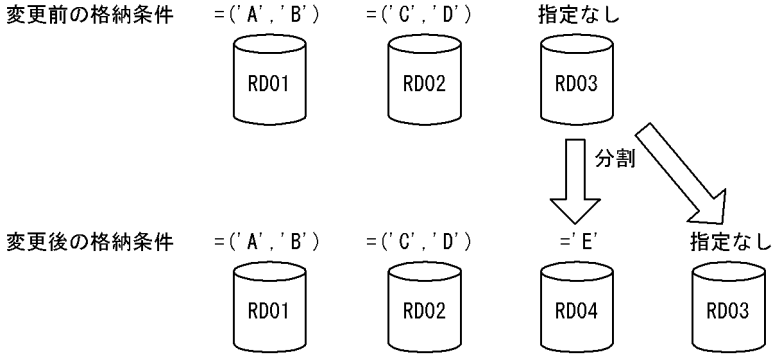


図 13–37　OTHERS 指定の RD エリアを分割する例

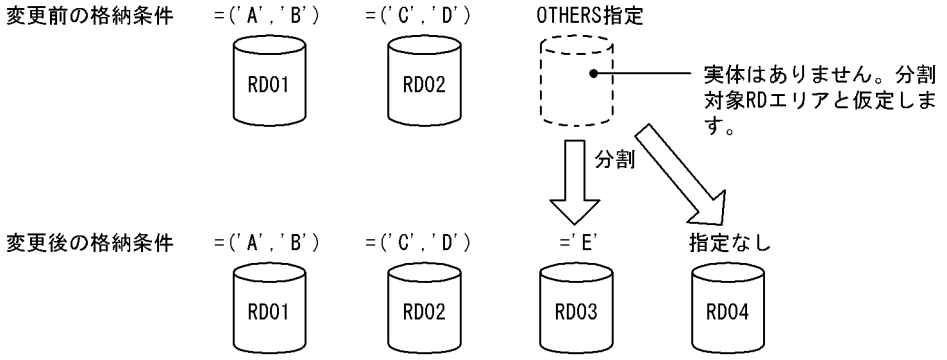
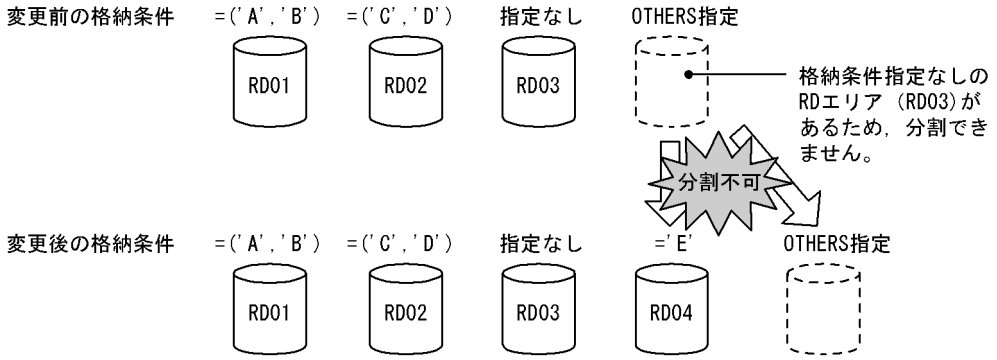


図 13–38　OTHERS 指定の RD エリアを分割できない例



(2)　OTHERS 指定の RD エリア

格納条件指定の分割時，実体のないダミーの RD エリアを分割できます。また，分割後に実体のある RD エリアとダミーの RD エリアに分割することもできます。この実体のないダミーの RD エリアを **OTHERS 指定の RD エリア**といいます。OTHERS 指定の RD エリアは，格納条件なしの RD エリアと仮定されて分割又は統合処理が行われます。OTHERS 指定の RD エリアの利用方法を次に示します。

- **新たな格納条件を指定した RD エリアや，格納条件なしの RD エリアを追加したりできる**

  OTHERS 指定の RD エリアを格納条件なしの RD エリアと仮定し，分割を行います。これによって，新たな格納条件を指定した RD エリアや，格納条件なしの RD エリアを追加できます。具体例については，図「OTHERS 指定の RD エリアを分割する例」を参照してください。

- **格納条件なしの RD エリアを削除したり，新たな格納条件を追加したりできる**

  格納条件なしの RD エリア又は OTHERS 指定の RD エリアを分割した場合，分割後の RD エリアの一つに OTHERS 指定の RD エリアを指定できます。OTHERS 指定の RD エリアは格納条件なしの RD エリアと仮定されるため，格納条件ありの RD エリアと一緒に指定すると，格納条件なしの RD エリアを削除したり，新たな格納条件を追加したりできます。具体例については，「分割後の RD エリアの決定方法」を参照してください。

(3)　分割後の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定した分割後の格納条件と RD エリアに従ってデータを格納する RD エリアが決まります。分割後の RD エリアは，分割対象 RD エリアでもよいし，新たに用意した RD エリアでもかまいません。ただし，分割後の RD エリア名に重複した名称を指定できません。

分割対象 RD エリアと分割後 RD エリアの組み合わせ可否について以下に説明します。

(a)　複数の格納条件を指定した RD エリアを分割対象とする場合

分割対象 RD エリア（複数の格納条件を指定した RD エリア）と分割後 RD エリアの組み合わせ可否を次の表に示します。

表 13–20　分割対象 RD エリア（複数の格納条件を指定した RD エリア）と分割後 RD エリアとの組み合わせ可否

ALTER TABLE の CHANGE RDAREA に指定する分割後 RD エリアの条件		分割後 RD エリアとしての指定可否	
		格納条件なしの RD エリアがある場合	全 RD エリアに格納条件が指定されている場合
分割対象 RD エリアを指定しない場合	新規の RD エリアだけを指定している	○（図「分割できるケース（その 1）」を参照）	
	格納条件指定ありの RD エリアを含んでいる	×（図「分割できないケース（その 1）」を参照）	
	格納条件指定なしの RD エリアを含んでいる	×（図「分割できないケース（その 1）」を参照）	–
	OTHERS 指定の RD エリアを指定	×（図「分割できないケース（その 2）」を参照）	
分割対象 RD エリアを指定する場合	分割対象 RD エリア以外は新規の RD エリアである	○（図「分割できるケース（その 2）」を参照）	
	格納条件指定ありの RD エリアを含んでいる	×（図「分割できないケース（その 1）」を参照）	
	格納条件指定なしの RD エリアを含んでいる	×（図「分割できないケース（その 1）」を参照）	–
	OTHERS 指定の RD エリアを指定	×（図「分割できないケース（その 2）」を参照）	

（凡例）

○：分割できます。

×：分割できません。

–：該当しません。

図 13–39　分割できるケース（その 1）

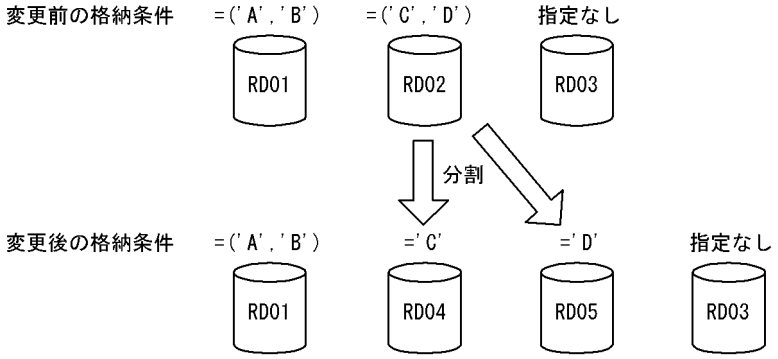


図 13–40　分割できるケース（その 2）

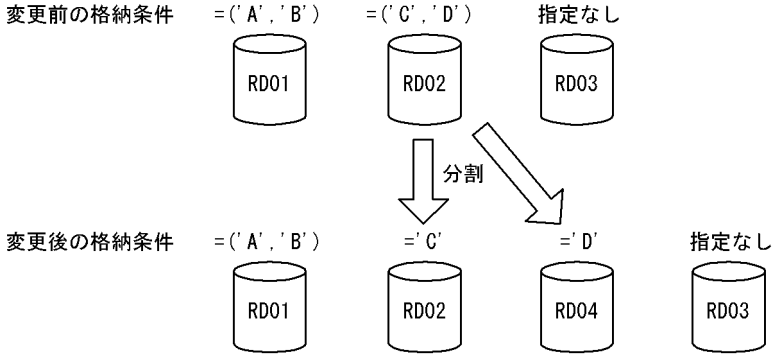
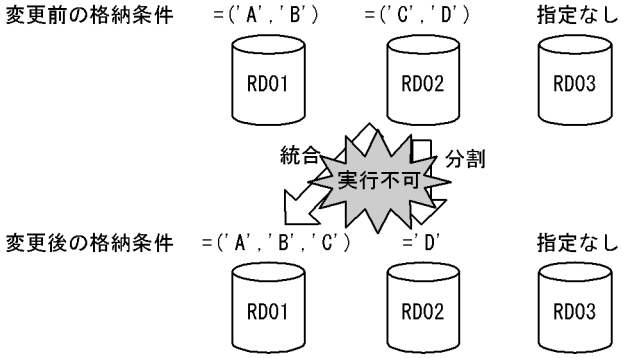
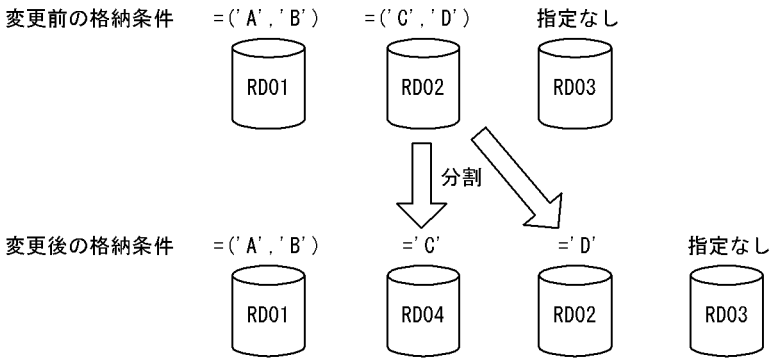


図 13–41　分割できないケース（その 1）



分割と統合を同時に行おうとしているため，ALTER TABLE実行時にエラーとなります。
この場合，分割と統合を２段階に分けて行ってください。

①１回目のALTER TABLEでRD02の格納条件を分割します。



②２回目のALTER TABLEでRD01とRD04を統合します。

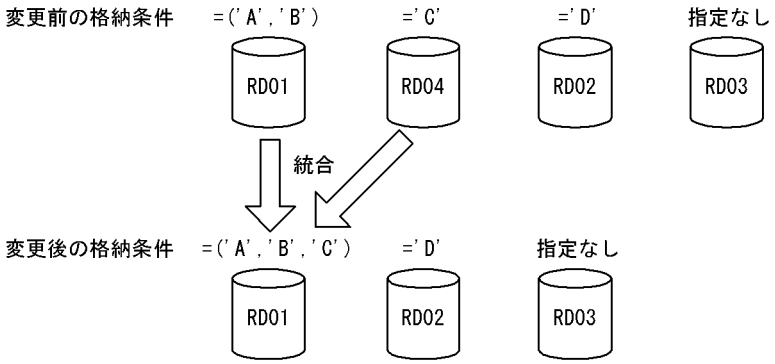
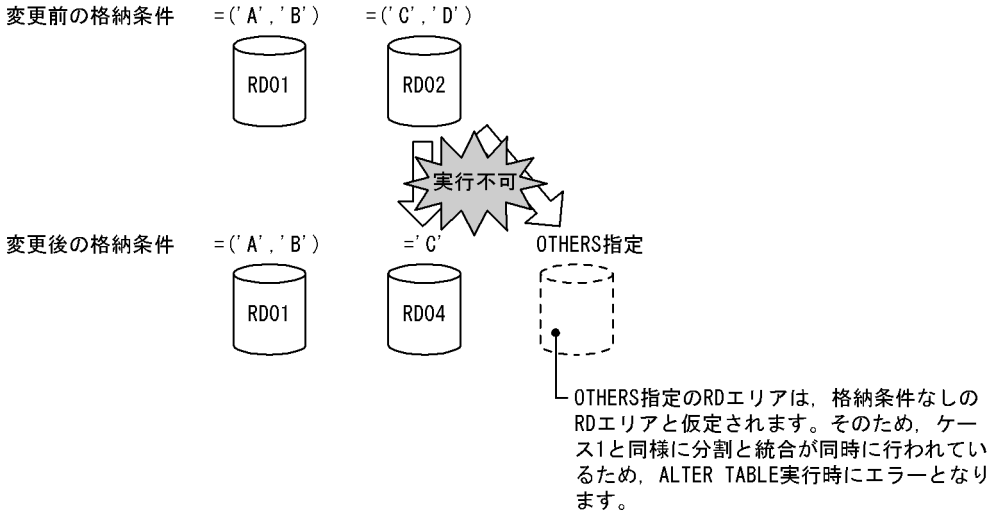


図 13–42　分割できないケース（その 2）



(b)　格納条件なしの RD エリアを分割対象とする場合

分割対象 RD エリア（格納条件なしの RD エリア）と分割後 RD エリアの組み合わせ可否を次の表に示します。

表 13–21　分割対象 RD エリア（格納条件なしの RD エリア）と分割後 RD エリアの組み合わせ可否

ALTER TABLE の CHANGE RDAREA に指定する分割後 RD エリアの条件			分割後 RD エリアとしての指定可否	
			格納条件なしの RD エリアがある場合	全 RD エリアに格納条件が指定されている場合
分割対象 RD エリアを指定しない場合	新規の RD エリアだけを指定している	OTHERS 指定の RD エリアを含まない	○（図「分割できるケース（その 3）」を参照）	−
		OTHERS 指定の RD エリアを含む	○（図「分割できるケース（その 4）」を参照）	−
	格納条件指定ありの RD エリアを含んでいる		×（図「分割できないケース（その 3）」を参照）	−
分割対象 RD エリアを指定する場合	分割対象 RD エリア以外は新規の RD エリアである	OTHERS 指定の RD エリアを含まない	○（図「分割できるケース（その 5）」を参照）	−
		OTHERS 指定の RD エリアを含む	○（図「分割できるケース（その 6）」を参照）	−
	格納条件指定ありの RD エリアを含んでいる		×（図「分割できないケース（その 3）」を参照）	−

（凡例）

○：分割できます。

×：分割できません。

−：該当しません。

図 13–43　分割できるケース（その 3）

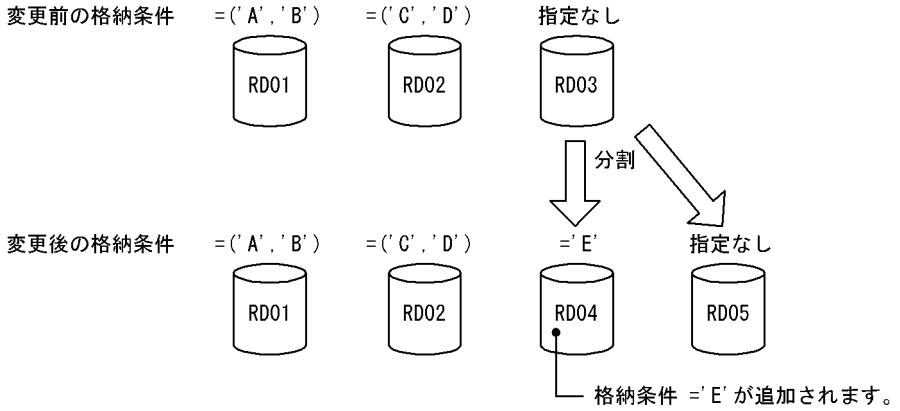
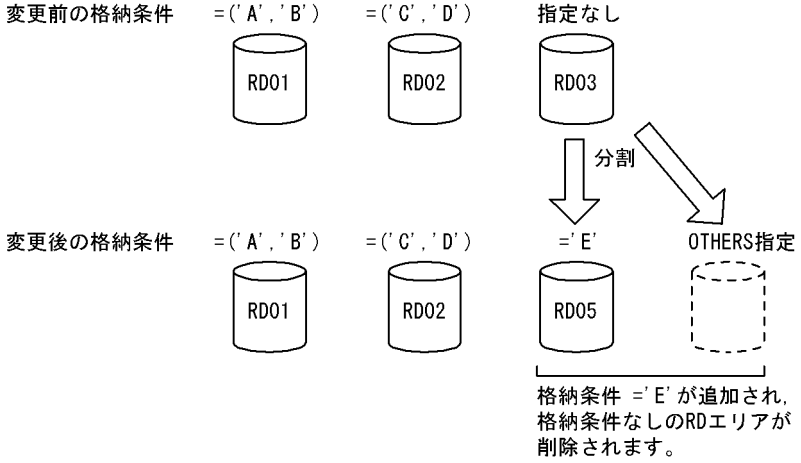


図 13–44　分割できるケース（その 4）



注意事項

この例の場合，分割後に'A'，'B'，'C'，'D'，'E'以外のデータが参照できなくなります。また，分割後に'A'，'B'，'C'，'D'，'E'以外のデータを挿入できなくなります。よって，分割後の RD エリアに OTHERS を指定する場合は注意が必要です。

図 13–45　分割できるケース（その 5）

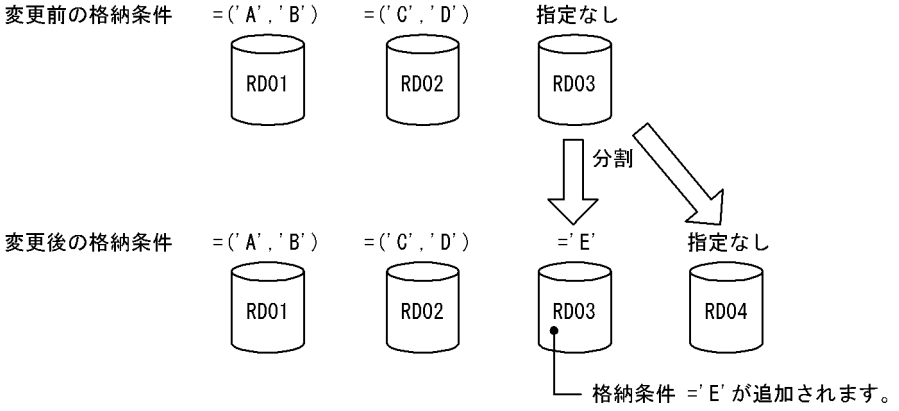
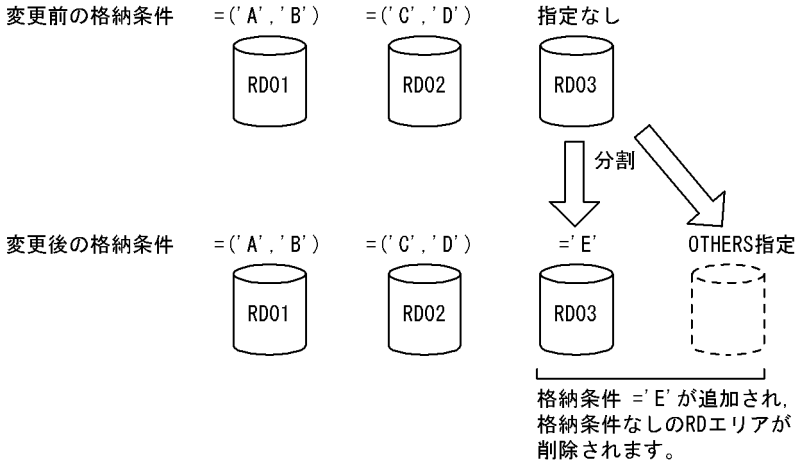


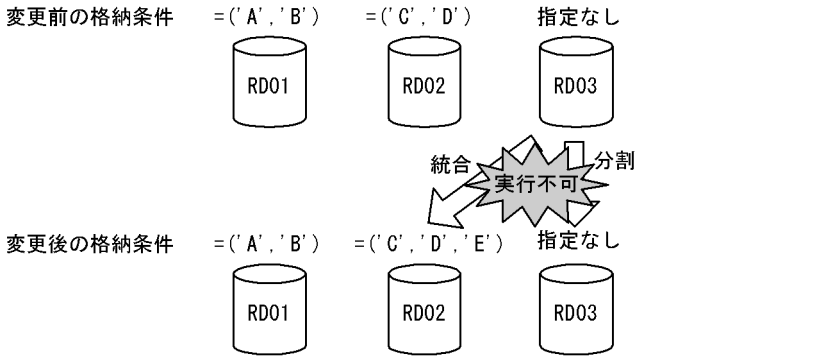
図 13–46　分割できるケース（その 6）



注意事項

この例の場合，分割後に'A'，'B'，'C'，'D'，'E'以外のデータが参照できなくなります。また，分割後に'A'，'B'，'C'，'D'，'E'以外のデータを挿入できなくなります。よって，分割後の RD エリアに OTHERS を指定する場合は注意が必要です。

図 13–47　分割できないケース（その 3）



分割と統合を同時に行おうとしているため，ALTER TABLE実行時にエラーとなります。
この場合，分割と統合を２段階に分けて行ってください。

(c)　OTHERS 指定の RD エリアを分割対象とする場合

分割対象 RD エリア（OTHERS 指定の RD エリアを分割対象とする場合）と分割後 RD エリアの組み合わせ可否を次の表に示します。

表 13–22　分割対象 RD エリア（OTHERS 指定の RD エリアを分割対象とする場合）と分割後 RD エリアの組み合わせ可否

ALTER TABLE の CHANGE RDAREA に指定する分割後 RD エリアの条件		分割後 RD エリアとしての指定可否	
		格納条件なしの RD エリアがある場合	全 RD エリアに格納条件が指定されている場合
分割後の RD エリアに OTHERS 指定の RD エリアがある場合	新規の RD エリアだけを指定している	×	○（図「分割できるケース（その 7）」を参照）
	格納条件指定ありの RD エリアを含んでいる	×	×（図「分割できないケース（その 4）」を参照）
分割後の RD エリアに OTHERS 指定の RD エリアがない場合	新規の RD エリアだけを指定している	×	○（図「分割できるケース（その 8）」を参照）
	格納条件指定ありの RD エリアを含んでいる	×	×（図「分割できないケース（その 4）」を参照）

（凡例）

○：分割できます。

×：分割できません。

図 13–48　分割できるケース（その 7）

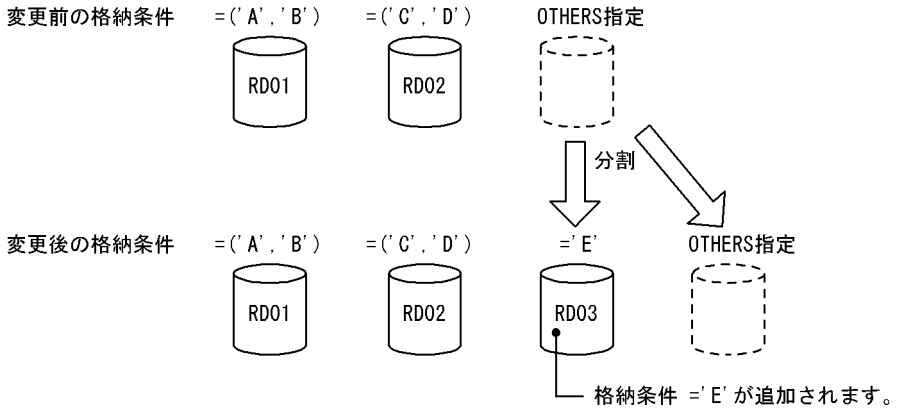


図 13–49　分割できるケース（その 8）

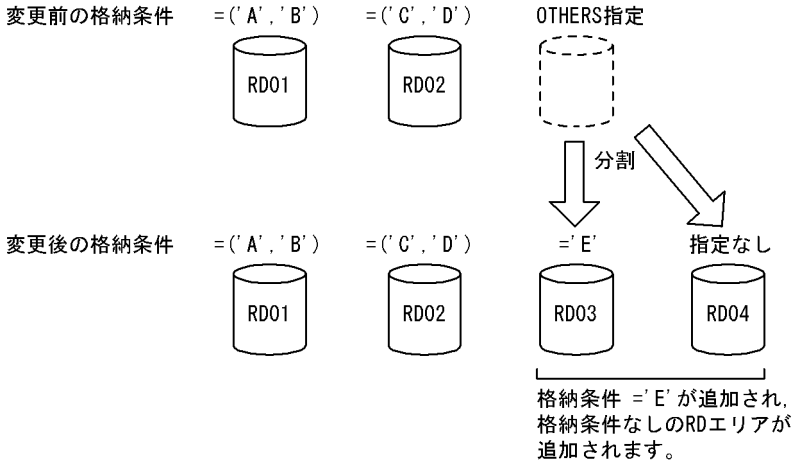
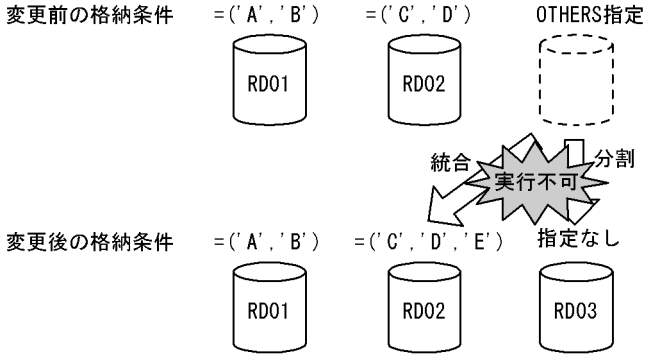


図 13–50　分割できないケース（その 4）



分割と統合を同時に行おうとしているため，ALTER TABLE実行時にエラーとなります。
この場合，分割と統合を２段階に分けて行ってください。

(4)　分割後の格納条件の指定方法

分割対象 RD エリアに対して分割後の格納条件を指定します。このときの格納条件は次に示す条件をすべて満たす必要があります。

- 指定した分割キー値が表定義に存在する
- 一つの格納条件に指定した列名のデータ型と定数のデータ型が比較可能である
- 分割後の格納条件が重複していない

分割後の格納条件の指定可否を次の表に示します。

表 13–23　分割後の格納条件の指定可否

分割対象 RD エリア	分割後の格納条件の指定内容		指定可否
格納条件を指定した RD エリアの場合	分割対象 RD エリアに含まれている格納条件を指定する	すべての格納条件を指定する	○
		一部の格納条件を指定しない	×
	分割対象 RD エリアに含まれない格納条件を指定する		×

分割対象 RD エリア	分割後の格納条件の指定内容		指定可否
	格納条件を指定しない		×
格納条件を指定していない RD エリアの場合	表定義で指定済みの格納条件を含んでいる		×
	表定義で指定済みの格納条件を含んでいない	格納条件を指定していない RD エリアがある	○
		格納条件を指定していない RD エリアがない	×
OTHERS 指定の RD エリアの場合	表定義で指定済みの格納条件を含んでいる		×
	表定義で指定済みの格納条件を含んでいない		○

（凡例）

○：指定できます。

×：指定できません。

なお，分割後の格納条件には，分割後の 1RD エリアに対して，一つ以上の格納条件を指定できます。

(5)　表に分割キーインデクスなどが定義されている場合

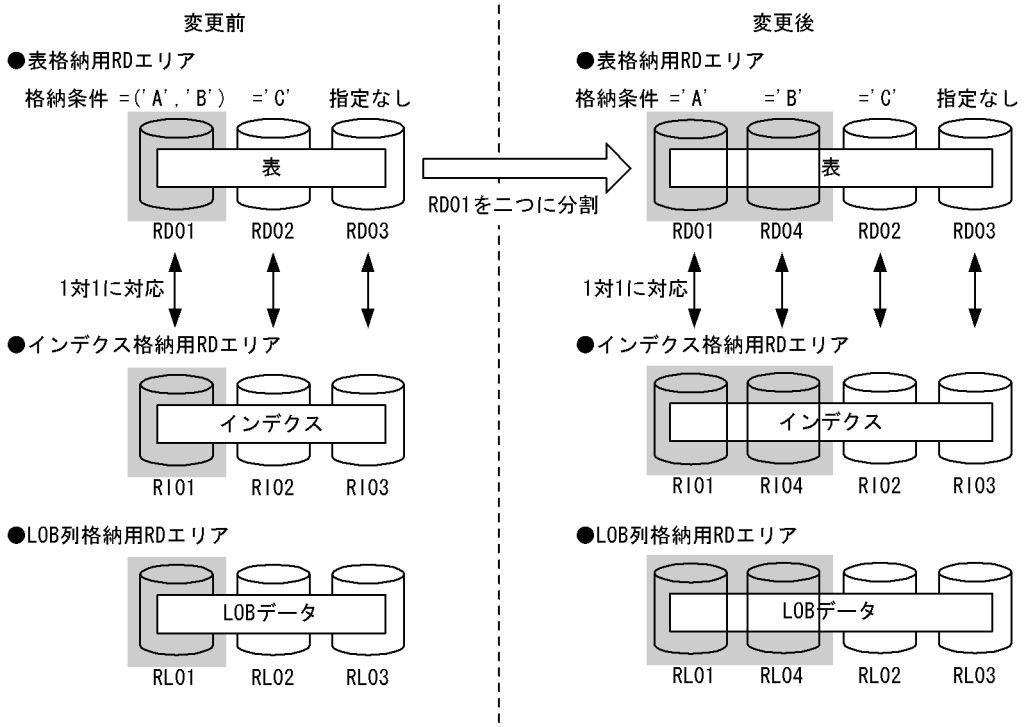
格納条件を変更する表に分割キーインデクスなどが定義されている場合，表格納用 RD エリアを分割したときに，インデクス格納用 RD エリアも同様に分割する必要があります。格納条件を変更するときに分割対象となるリソースを次の表に示します。

表に分割キーインデクスなどが定義されている場合の分割例を次の図に示します。

表 13–24　格納条件を変更するときに分割対象となるリソース

表に定義されているリソース	分割方法
インデクス	表格納用 RD エリアと 1 対 1 になるように分割してください。
クラスタキー	
プライマリキー	
LOB 列	

図 13–51　表に分割キーインデクスなどが定義されている場合の分割例



また，分割時には次に示す規則があります。

- リソースが複数定義されている場合は，すべてのリソースが分割対象になります。
- 分割指定が正しくない場合，ALTER TABLE の実行時にエラーとなります。
- 格納条件を変更した後のインデクス格納用 RD エリア名，LOB 列格納用 RD エリア名に重複した名称を指定できません。

(6)　分割対象 RD エリア内のデータの扱い

ALTER TABLE の実行時，分割対象 RD エリア内のデータは通常削除されます。また，対応する次の RD エリア内のデータも削除されます。

- インデクス格納用 RD エリア内のインデクスデータ
- LOB 列格納用 RD エリア内の LOB データ

参考

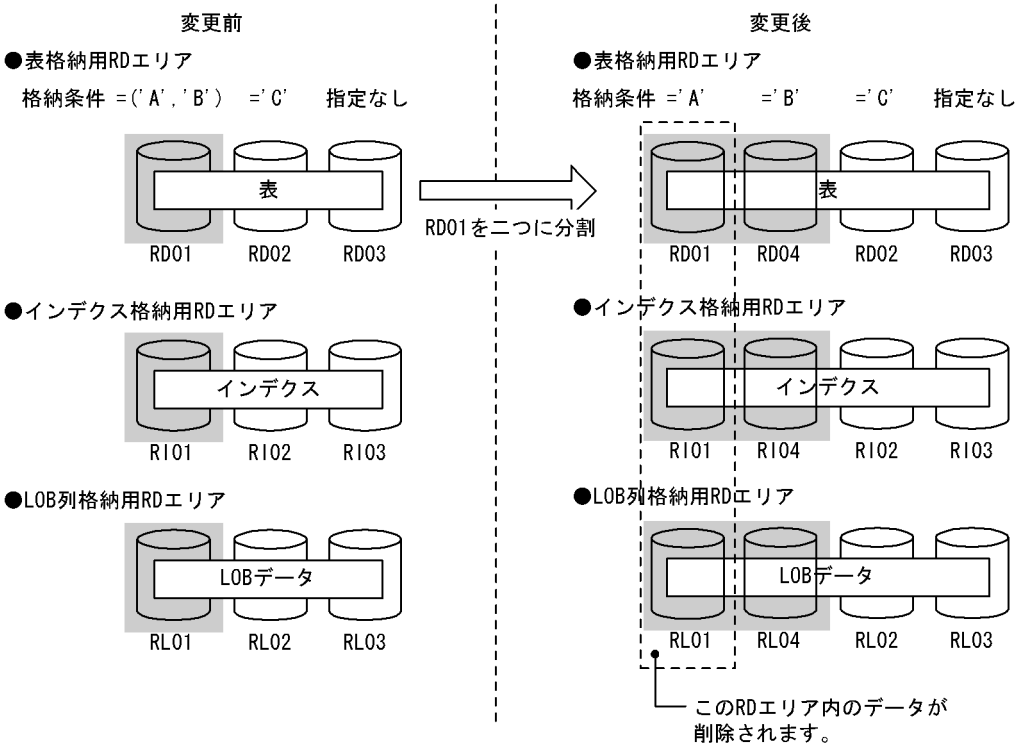
分割対象 RD エリア内のデータが削除される理由を次に示します。

- 分割対象 RD エリア内のデータの一部が，分割後はその RD エリアに格納される格納条件対象外のデータとなる可能性があるため

なお，分割対象 RD エリア以外の RD エリア内のデータは削除されません。

分割時に削除対象となるデータを次の図に示します。

図 13–52　分割時に削除対象となるデータ



(a)　データが削除されないケース

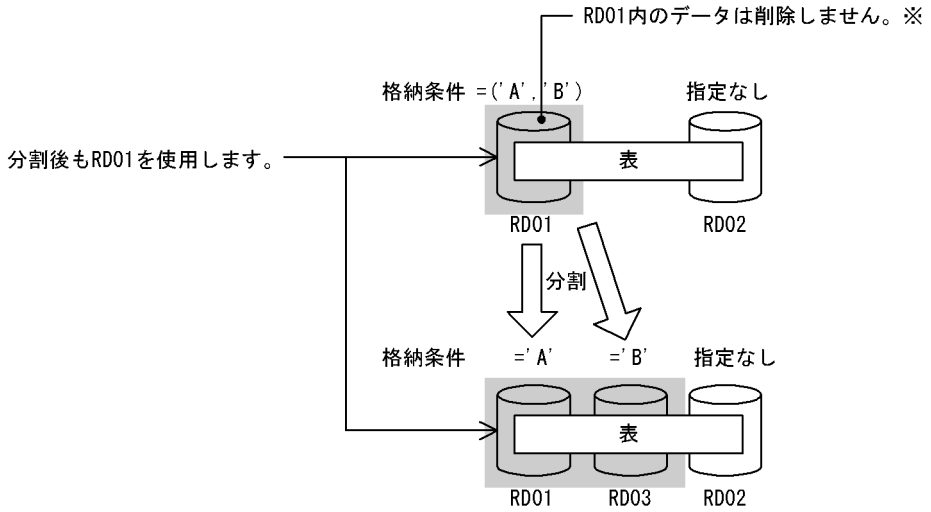
次に示す条件をすべて満たす場合は，分割対象 RD エリア内のデータをそのまま使用できる可能性があるため，データを削除しないようにできます。

1. 分割対象 RD エリアを分割後にも使用する
2. 分割対象 RD エリア内には，分割後の格納条件を満たすデータだけが存在する
3. 分割対象となるインデクス格納用 RD エリア又は LOB 列格納用 RD エリアに対しても 1 及び 2 の条件が満たされている

データを削除しない場合は，ALTER TABLE で WITHOUT PURGE を指定してください。なお，分割対象 RD エリアを分割後にも使用する指定をしていない場合に WITHOUT PURGE を指定すると，ALTER TABLE の実行時にエラーとなります。

WITHOUT PURGE が有効なケースと WITHOUT PURGE がエラーとなるケースを次の図に示します。

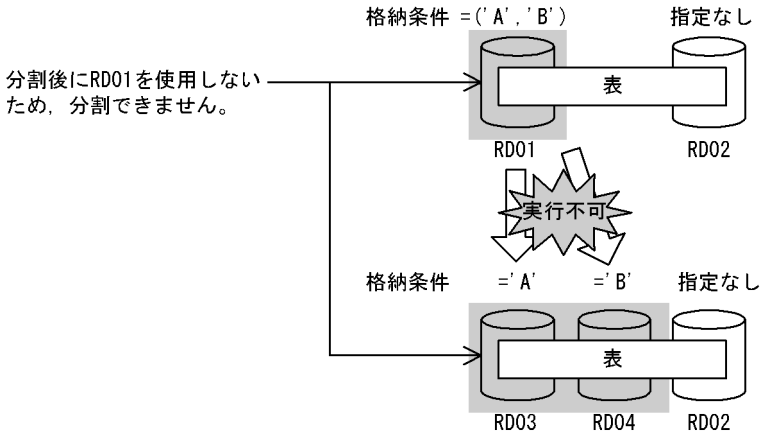
図 13–53　WITHOUT PURGE が有効なケース



注※

分割前の RD01 内に='B'のデータがある場合，分割後の RD01 内に格納条件対象外のデータが格納されてしまいます。このため，分割前の RD01 内に='A'以外のデータがないことを確認する必要があります。

図 13–54　WITHOUT PURGE がエラーとなるケース



(b)　WITHOUT PURGE を指定する場合の注意事項

WITHOUT PURGE を指定して分割した場合，HiRDB は RD エリア内の全データが分割後の格納条件と一致しているかどうかのチェックを行いません。格納条件対象外のデータが RD エリア中にあると，SQL を実行するときに HiRDB が正しく動作しません。このため，WITHOUT PURGE 指定をして格納条件を変更する場合は，分割対象 RD エリア内のデータを確認するようにしてください。

ポイント

通常は WITHOUT PURGE を指定しないで格納条件を変更してください。格納条件の変更手順については，「表の分割格納条件を変更するときの運用」を参照してください。

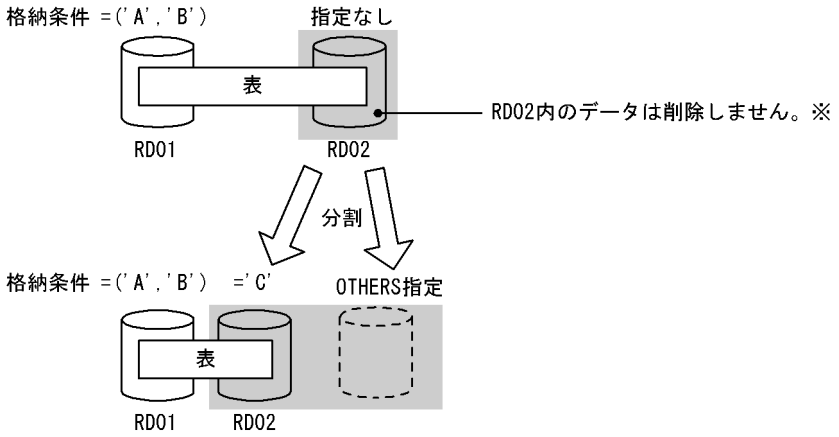
(c)　分割後の RD エリアに OTHERS 指定をする場合の注意事項

次に示す条件をすべて満たす場合，分割後の RD エリアはすべて格納条件を指定した RD エリアになります（分割後，格納条件なしの RD エリアがなくなります）。

- 格納条件なしの RD エリアを分割対象としている
- 分割後に OTHERS 指定の RD エリアを指定している

このため，分割後の格納条件に合わないデータがあると，そのデータは格納先 RD エリアがなくなります。分割後に格納先 RD エリアがなくなる例を次の図に示します。

図 13–55　分割後に格納先 RD エリアがなくなる例



注※

格納条件が='C'以外のデータは，分割後に格納先 RD エリアがなくなります。

13.12.9　RD エリアの統合（格納条件指定の場合）

(1)　格納条件の変更規則

格納条件の変更規則について説明します。

(a)　上限値及び下限値

格納条件を変更するときは，次の表に示す上限値及び下限値の規則を守ってください。

表 13–25　統合機能での上限値及び下限値

項目	上限値	下限値	上限値又は下限値を超えた場合の動作
一度に統合の対象にできる RD エリア数	16	2	ALTER TABLE をエラーとします。
ALTER TABLE の実行後，統合される RD エリア数	1（固定）		

項目	上限値	下限値	上限値又は下限値を超えた場合の動作
統合した結果の RD エリアの総数	−	1	

（凡例）−該当しません。

(b)　統合時の規則

統合対象となる RD エリアは一定の条件を満たしている必要があります。RD エリアの統合可否を次の表に示します。

表 13–26　RD エリアの統合可否

統合対象 RD エリア		統合可否	説明
統合対象 RD エリアは，表定義に指定された RD エリアである		○	RD エリアを統合できます。
統合対象 RD エリアに，表定義に指定されていない RD エリアがある		×	ALTER TABLE がエラーになります。
OTHERS 指定の RD エリア※	すべての RD エリアに格納条件が指定されている	○	OTHERS 指定の RD エリアを統合する例を図「OTHERS 指定の RD エリアを統合する例」に示します。
	格納条件なしの RD エリアがある	×	ALTER TABLE がエラーになります。 OTHERS 指定の RD エリアを統合できない例を図「OTHERS 指定の RD エリアを統合できない例」に示します。

（凡例）

○：統合できます。

×：統合できません。

注※

OTHERS 指定の RD エリアについては，「OTHERS 指定の RD エリア」を参照してください。

図 13–56　OTHERS 指定の RD エリアを統合する例

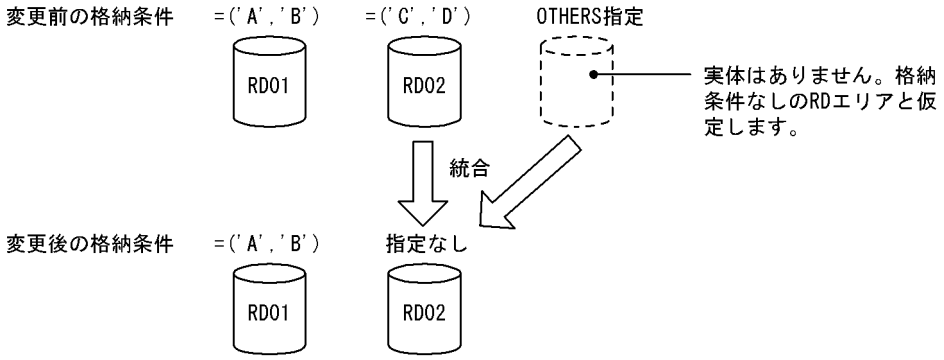
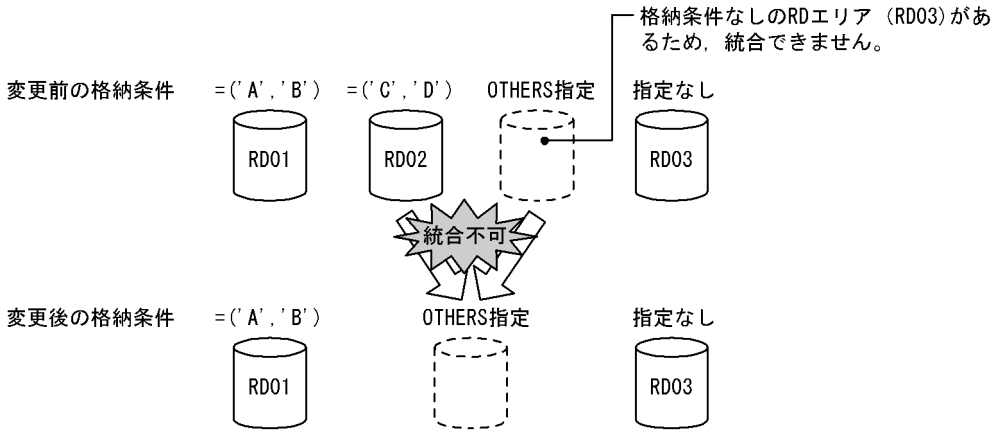


図 13–57　OTHERS 指定の RD エリアを統合できない例



(2)　OTHERS 指定の RD エリア

格納条件指定の統合時，実体のないダミーの RD エリアを統合対象にできます。また，ダミーの RD エリアに統合することもできます。この実体のないダミーの RD エリアを **OTHERS 指定の RD エリア**といいます。OTHERS 指定の RD エリアは，格納条件なしの RD エリアと仮定されて分割又は統合処理が行われます。OTHERS 指定の RD エリアの利用方法を次に示します。

- **既存の格納条件を削除したり，格納条件なしの RD エリアを作成したりできる**

  OTHERS 指定の RD エリアを統合対象 RD エリアに含めることで，既存の格納条件を削除したり，格納条件なしの RD エリアを作成したりできます。具体例については，図「OTHERS 指定の RD エリアを統合する例」を参照してください。

- **格納条件ありの RD エリアを削除できる**

  統合対象 RD エリアに，格納条件なしの RD エリア又は OTHERS 指定の RD エリアがある場合，統合後の RD エリアに OTHERS 指定の RD エリアを指定できます。OTHERS 指定の RD エリアは格納条件なしの RD エリアと仮定されるため，格納条件ありの RD エリアと一緒に統合すると，格納条件ありの RD エリアを削除できます。具体例については，「統合後の RD エリアの決定方法」を参照してください。

(3)　統合後の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に統合後の RD エリアを指定します。この RD エリアには，統合対象 RD エリアの全格納条件を満たすデータが格納されます。統合後の RD エリアは，統合対象 RD エリアの中の一つであってもよいし，新たに用意した RD エリアでもかまいません。

ただし，統合時の条件によっては統合できないこともあります。統合対象 RD エリアと統合後 RD エリアの組み合わせ可否について以下に説明します。

(a)　統合後の RD エリア総数の条件

統合後の RD エリアの総数について条件があります。統合時の条件と統合後の RD エリア総数の関係を次の表に示します。

表 13–27　統合時の条件と統合後の RD エリア総数の関係

統合時の条件			統合後の RD エリアの総数	
統合対象表の条件	統合対象 RD エリアの条件		1 個	2 個以上
格納条件なしの RD エリアがある	OTHERS 指定の RD エリアがある		×※1	
	OTHERS 指定の RD エリアがない	格納条件なしの RD エリアがある	×（図「統合できないケース（その 1）」を参照）	○（図「統合できるケース（その 1）」を参照）
		格納条件なしの RD エリアがない	−	○（図「統合できるケース（その 2）」を参照）
格納条件なしの RD エリアがない	OTHERS 指定の RD エリアがある		×（図「統合できないケース（その 2）」を参照）	○（図「統合できるケース（その 3）」を参照）
	OTHERS 指定の RD エリアがない		○※2（図「統合できるケース（その 4）」を参照）	○（図「統合できるケース（その 5）」を参照）

（凡例）

○：統合できます。

×：統合できません。

−：該当しません。

注※1

統合対象表に格納条件なしの RD エリアがある場合は，統合対象 RD エリアに OTHERS 指定の RD エリアを指定することはできません。

注※2

統合対象表にインデクスが定義されている場合は統合できません。

図 13–58　統合できるケース（その 1）

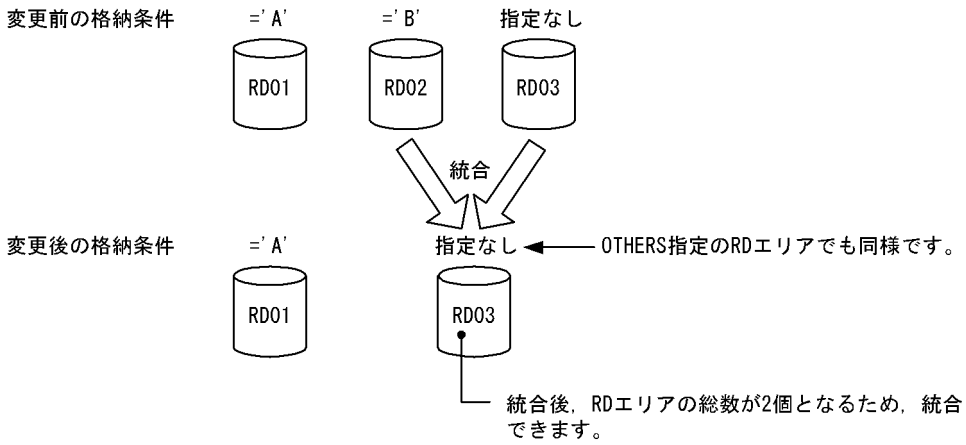


図 13–59　統合できるケース（その 2）

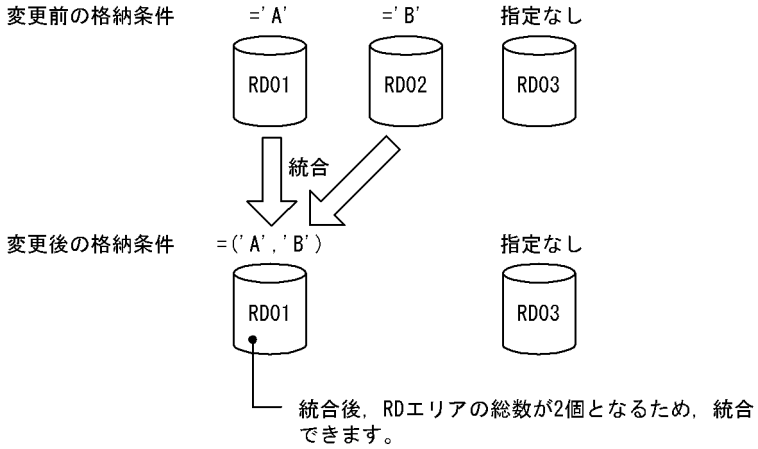


図 13–60　統合できるケース（その 3）

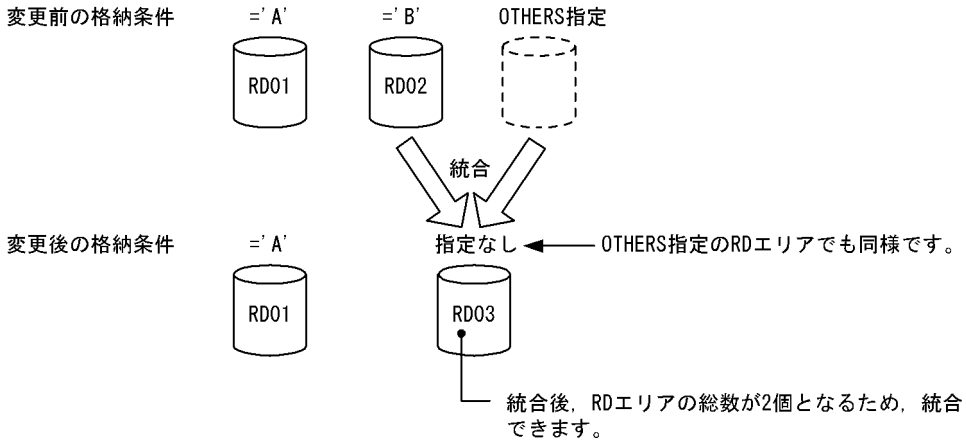


図 13–61　統合できるケース（その 4）

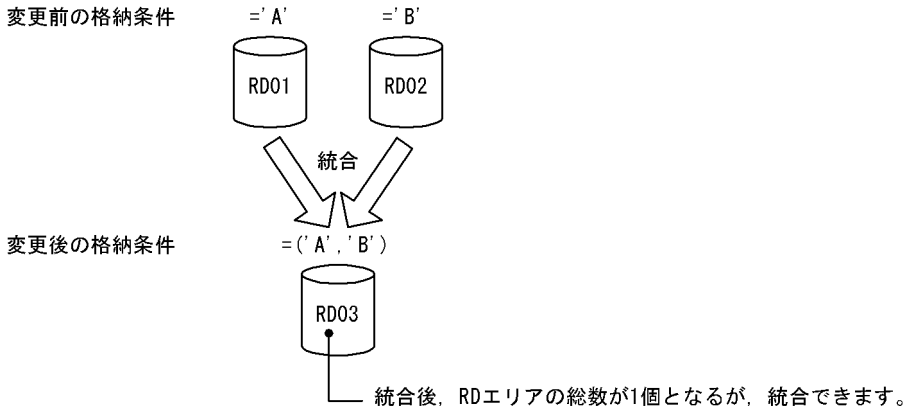


図 13–62　統合できるケース（その 5）

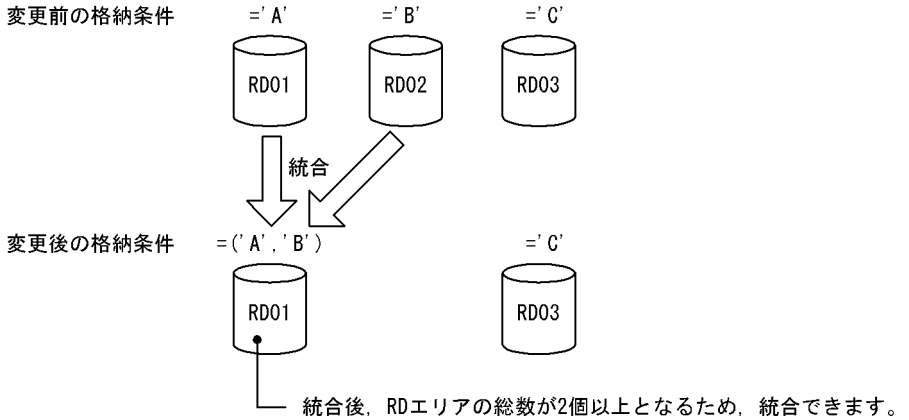


図 13–63　統合できないケース（その 1）

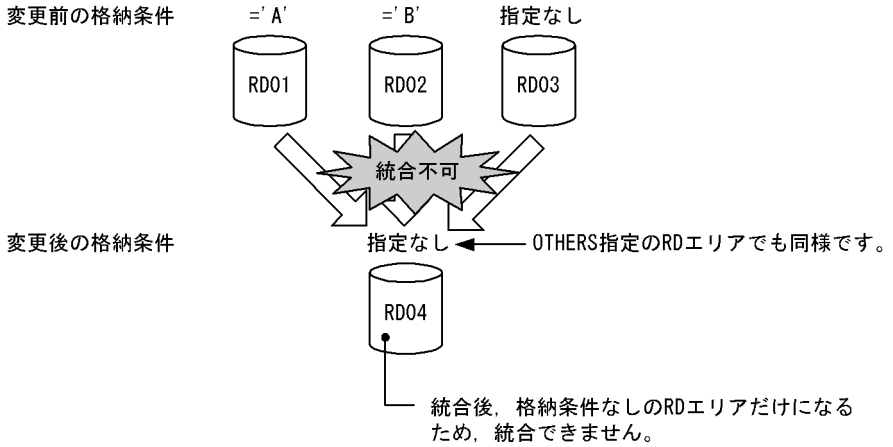
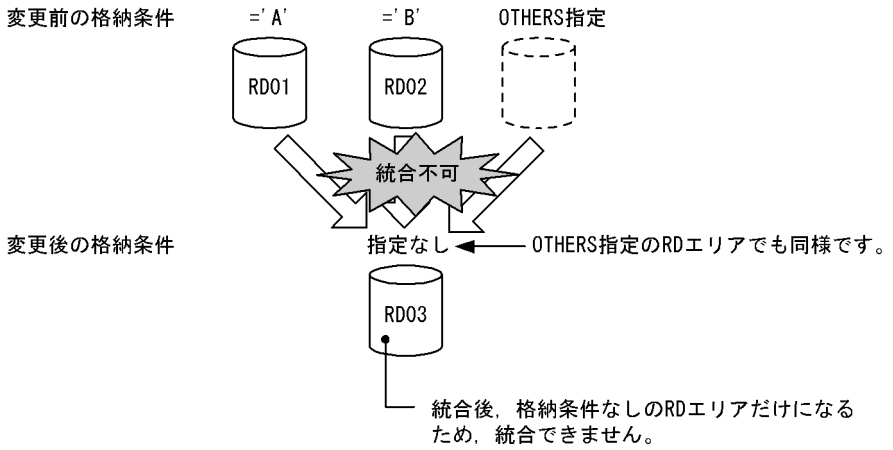


図 13–64　統合できないケース（その 2）



(b)　格納条件ありの RD エリアだけを統合対象とする場合

統合後の RD エリアの指定条件と統合可否（格納条件ありの RD エリアだけを統合対象とする場合）を次の表に示します。

表 13–28　統合後の RD エリアの指定条件と統合可否（格納条件ありの RD エリアだけを統合対象とする場合）

統合後の RD エリアの指定条件		統合可否
統合対象 RD エリアを統合後に使用しない	表格納 RD エリアのどれかを統合後の RD エリアに指定する	×（RD エリアの指定誤り）
	表格納 RD エリア以外の RD エリアを統合後の RD エリアに指定する	○（図「統合できるケース（その 6）」を参照）
	OTHERS 指定の RD エリアを統合後の RD エリアに指定する	×（図「統合できないケース（その 3）」を参照）
統合対象 RD エリアを統合後に使用する		○（図「統合できるケース（その 7）」を参照）

（凡例）

○：統合できます。

×：統合できません。

図 13–65　統合できるケース（その 6）

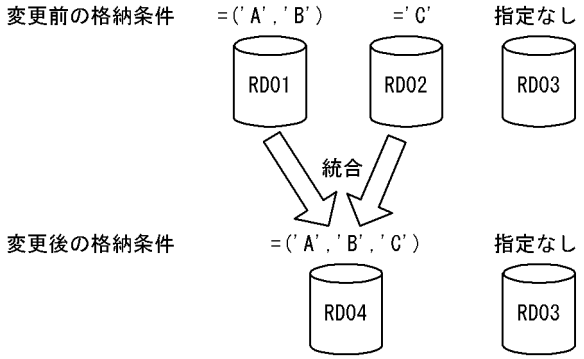


図 13–66　統合できるケース（その 7）

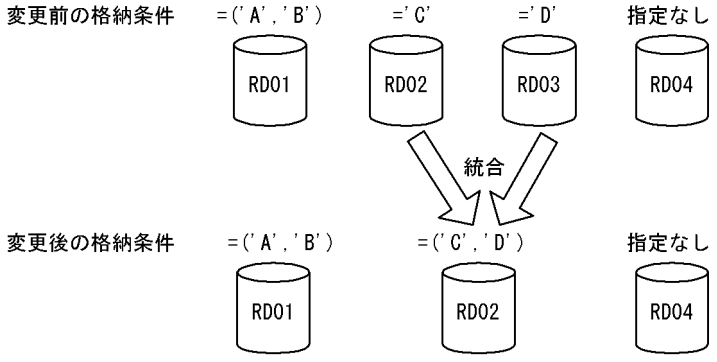
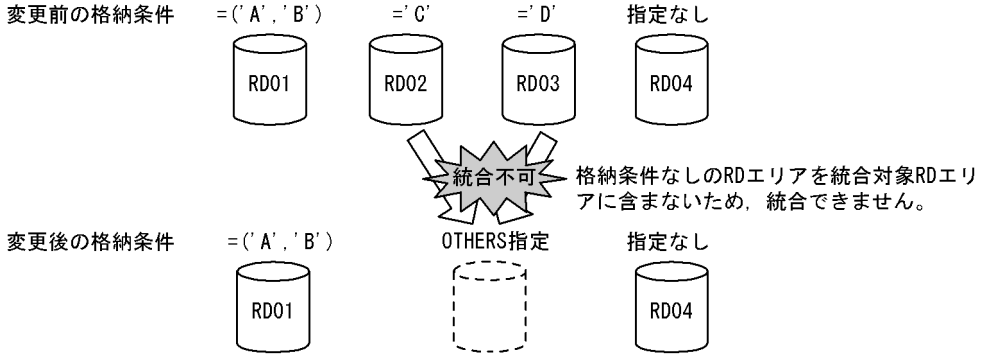


図 13–67　統合できないケース（その 3）



(c)　格納条件なしの RD エリアを統合対象とする場合

統合後の RD エリアの指定条件と統合可否（格納条件なしの RD エリアを統合対象とする場合）を次の表に示します。

表 13–29　統合後の RD エリアの指定条件と統合可否（格納条件なしの RD エリアを統合対象とする場合）

統合後の RD エリアの指定条件		統合後の表格納 RD エリアの条件	
		格納条件なしの RD エリアがある場合	格納条件なしの RD エリアがない場合
統合対象 RD エリアを統合後に使用しない	表格納 RD エリアのどれかを統合後の RD エリアに指定する	×（RD エリアの指定誤り）	−
	表格納 RD エリア以外の RD エリアを統合後の RD エリアに指定する	○（図「統合できるケース（その 8）を参照」）	−
	OTHERS 指定の RD エリアを統合後の RD エリアに指定する	○（図「統合できるケース（その 9）を参照」）	−
統合対象 RD エリアを統合後に使用する		○（図「統合できるケース（その 10）を参照」）	−

（凡例）

○：統合できます。

×：統合できません。

−：該当しません。

図 13–68　統合できるケース（その 8）

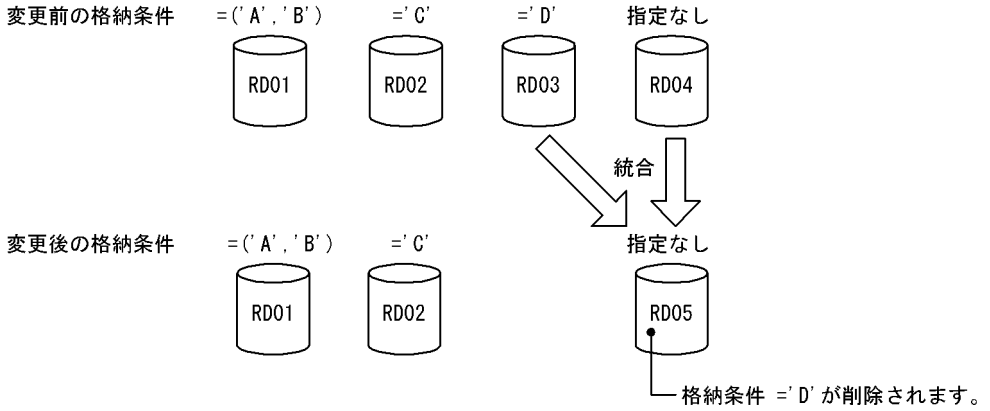


図 13–69　統合できるケース（その 9）

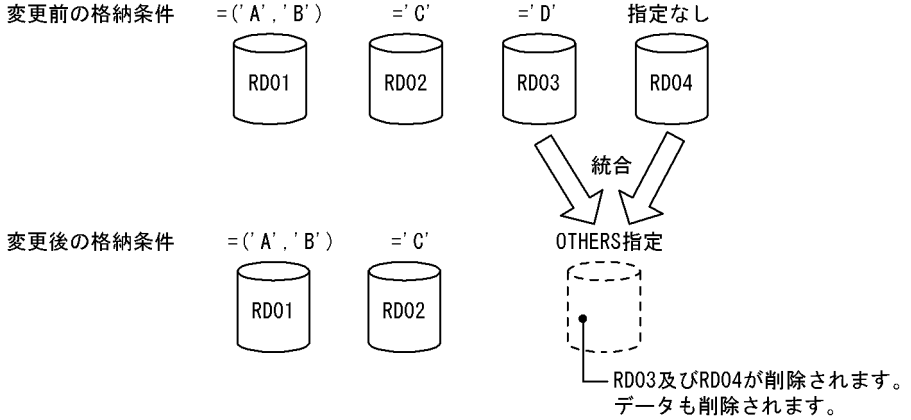
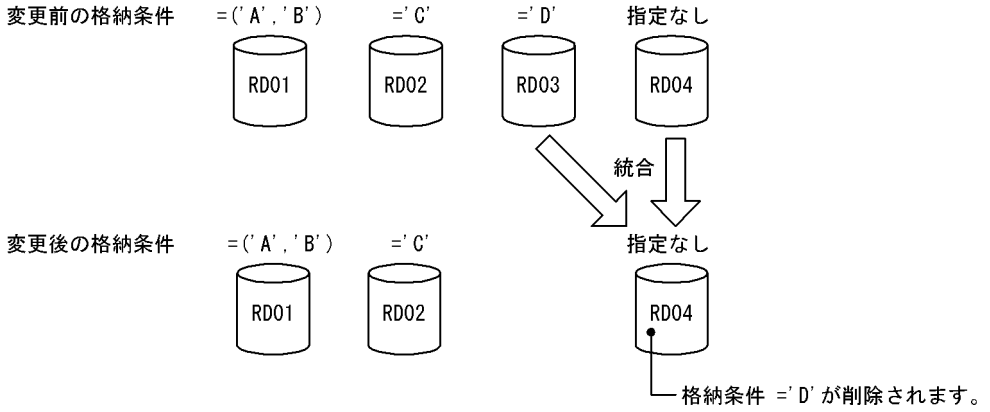


図 13–70　統合できるケース（その 10）



(d)　OTHERS 指定の RD エリアを統合対象とする場合

統合後の RD エリアの指定条件と統合可否（OTHERS 指定の RD エリアを統合対象とする場合）を次の表に示します。

表 13–30　統合後の RD エリアの指定条件と統合可否（OTHERS 指定の RD エリアを統合対象とする場合）

統合後の RD エリアの指定条件		統合後の表格納 RD エリアの条件	
		格納条件なしの RD エリアがある場合	格納条件なしの RD エリアがない場合
統合対象 RD エリアを統合後に使用しない	格納条件ありの RD エリアを統合後の RD エリアに指定する	×（OTHERS 指定不可）	×（統合対象に存在しない RD エリアを指定している）
	表格納 RD エリア以外の RD エリアを統合後の RD エリアに指定する	×（OTHERS 指定不可）	○（図「統合できるケース（その 11）」を参照）
	OTHERS 指定の RD エリアを統合後の RD エリアに指定する	×（OTHERS 指定不可）	○（図「統合できるケース（その 12）」を参照）
統合対象 RD エリアを統合後に使用する	格納条件ありの RD エリアを統合後の RD エリアに指定する	×（OTHERS 指定不可）	○（図「統合できるケース（その 11）」を参照）
	OTHERS 指定の RD エリアを統合後の RD エリアに指定する	×（OTHERS 指定不可）	○（図「統合できるケース（その 12）」を参照）

（凡例）

○：統合できます。

×：統合できません。

図 13–71　統合できるケース（その 11）

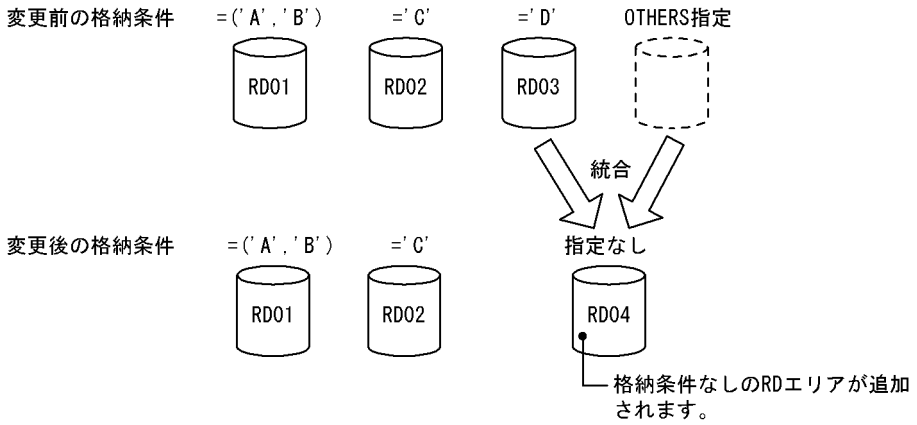
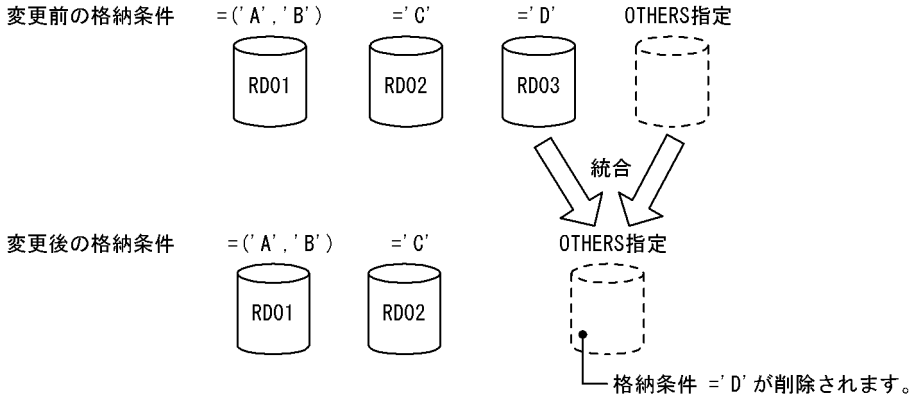


図 13–72　統合できるケース（その 12）



(4)　表に分割キーインデクスなどが定義されている場合

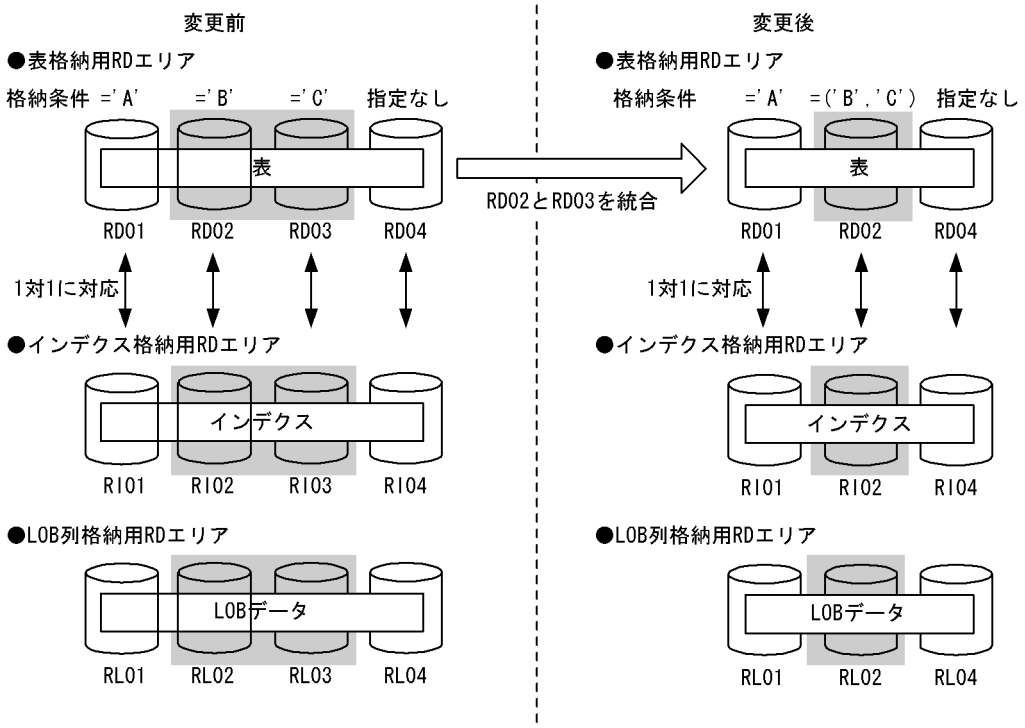
統合対象表に分割キーインデクスなどが定義されている場合，表格納用 RD エリアだけでなく，インデクス格納用 RD エリアも同様に統合する必要があります。同時に統合する必要があるリソースを次の表に示します。

また，表に分割キーインデクスなどが定義されている場合の統合例を次の図に示します。

表 13–31　格納条件を変更するときに分割対象となるリソース

表に定義されているリソース	統合方法
インデクス	表格納用 RD エリアと 1 対 1 になるように統合してください。
クラスタキー	
プライマリキー	
LOB 列	

図 13–73　表に分割キーインデクスなどが定義されている場合の統合例



また，統合時には次に示す規則があります。

- リソースが複数定義されている場合は，すべてのリソースが統合対象になります。
- 統合指定が正しくない場合，ALTER TABLE の実行時にエラーとなります。

(5)　統合対象 RD エリア内のデータの扱い

ALTER TABLE の実行時，統合対象 RD エリア内のデータは通常削除されます。また，対応する次の RD エリア内のデータも削除されます。

- インデクス格納用 RD エリア内のインデクスデータ
- LOB 列格納用 RD エリア内の LOB データ

参考

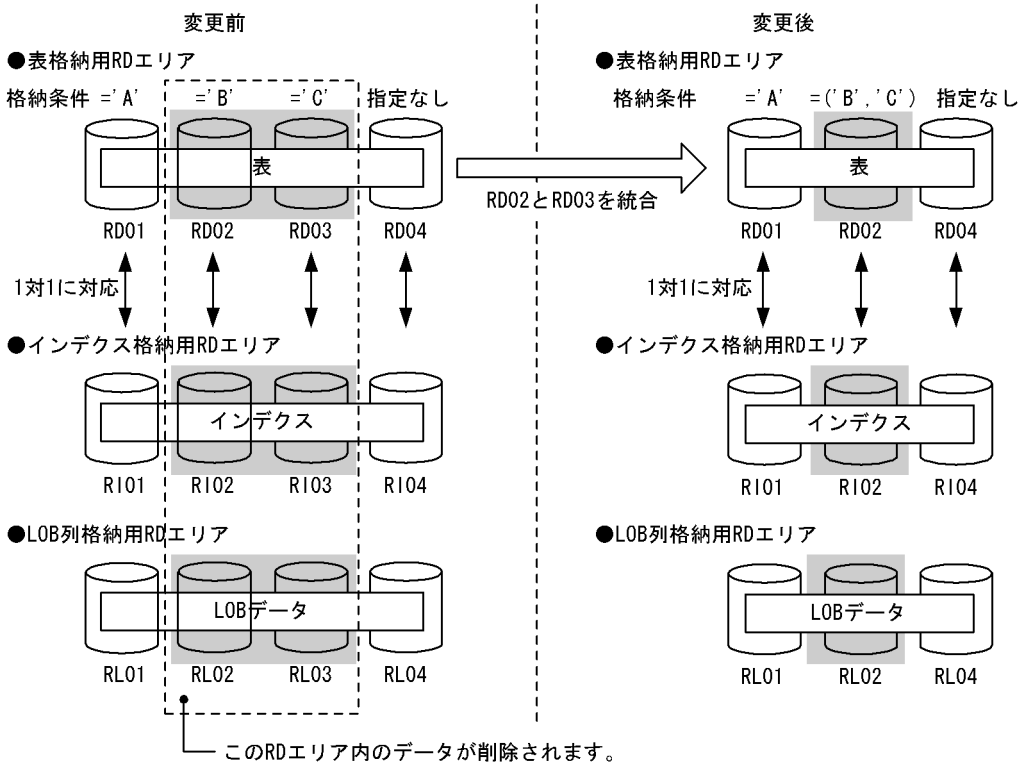
統合対象 RD エリア内のデータが削除される理由を次に示します。

- 統合対象 RD エリアは統合後に使用されなくなる可能性があるため

なお，統合対象 RD エリア以外の RD エリア内のデータは削除されません。

統合時に削除対象となるデータを次の図に示します。

図 13–74　統合時に削除対象となるデータ



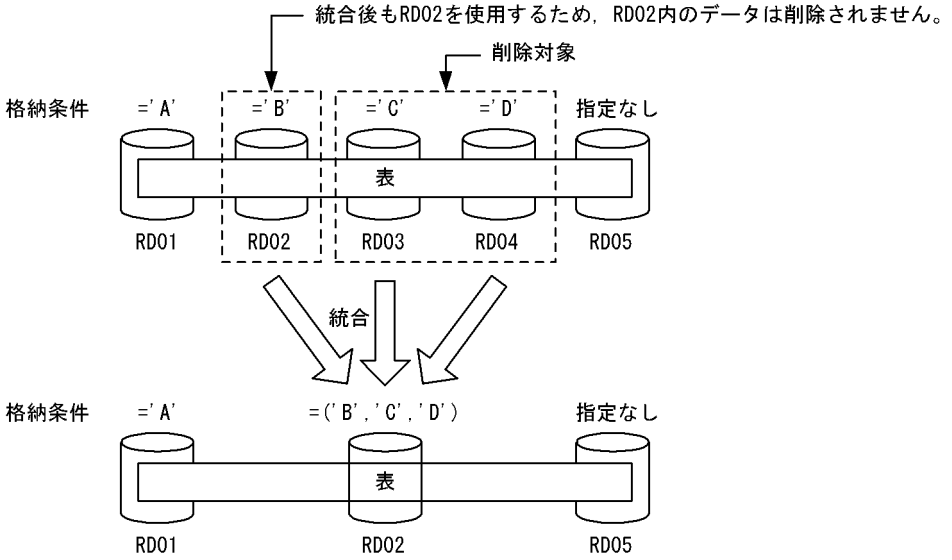
(a)　データが削除されないケース

次に示す条件をすべて満たす場合は，統合対象 RD エリア内のデータをそのまま使用できる可能性があるため，データを削除しないようにできます。

1. 統合対象 RD エリアの一つを統合後にも使用する
2. 統合対象となるインデクス格納用 RD エリア又は LOB 列格納用 RD エリアに対しても 1 の条件が満たされている

データを削除しない場合は，ALTER TABLE で WITHOUT PURGE を指定してください。なお，統合対象 RD エリアの一つを統合後にも使用する指定をしていない場合に WITHOUT PURGE を指定すると，ALTER TABLE の実行時にエラーとなります。WITHOUT PURGE を指定したときの処理を次の図に示します。

図 13–75　WITHOUT PURGE を指定したときの処理



〔説明〕

RD02 に統合したため，RD02 内のデータ（格納条件が='B'のデータ）は削除されません。RD03 及び RD04 内のデータ（格納条件が='C'及び'D'のデータ）が削除されます。

注意事項

WITHOUT PURGE でデータを保存した場合，統合前のデータをアンロードして統合後の RD エリアにデータロードすると，データが二重登録されてしまいます。このため，WITHOUT PURGE でデータを保存する場合は，データのアンロード及びデータロードを行わないでください。

図「WITHOUT PURGE を指定したときの処理」を例にすると，RD02 内のデータ（格納条件が='B'のデータ）をアンロードして統合後の RD エリアにデータロードすると，データが二重登録されてしまいます。

(b)　OTHERS 指定の RD エリアに統合する場合の注意事項

次に示す条件をすべて満たす場合に統合を行うと，統合前のデータを格納する RD エリアがなくなります。

- 格納条件なし又は OTHERS 指定の RD エリアを統合対象としている
- OTHERS 指定の RD エリアに統合している

統合後にデータを格納する RD エリアがなくなるケースを次の図に示します。

図 13–76　統合後にデータを格納する RD エリアがなくなるケース（その 1）

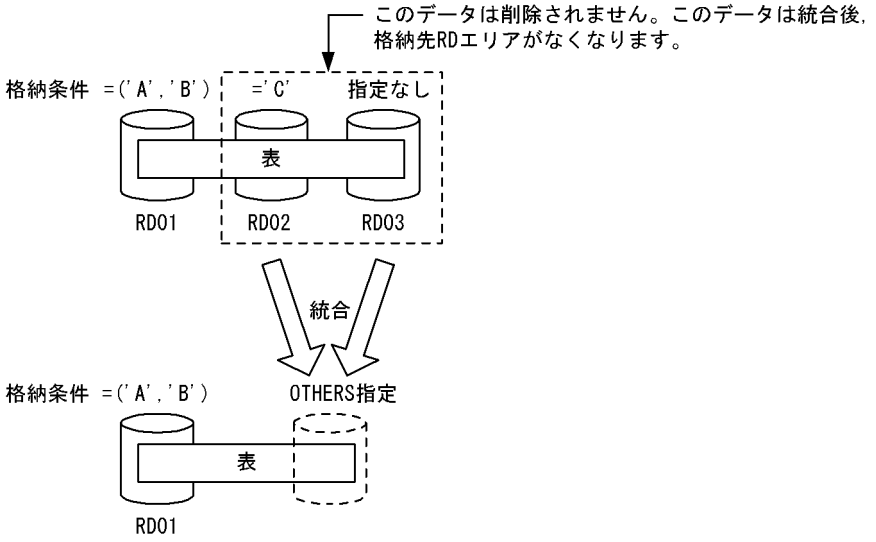
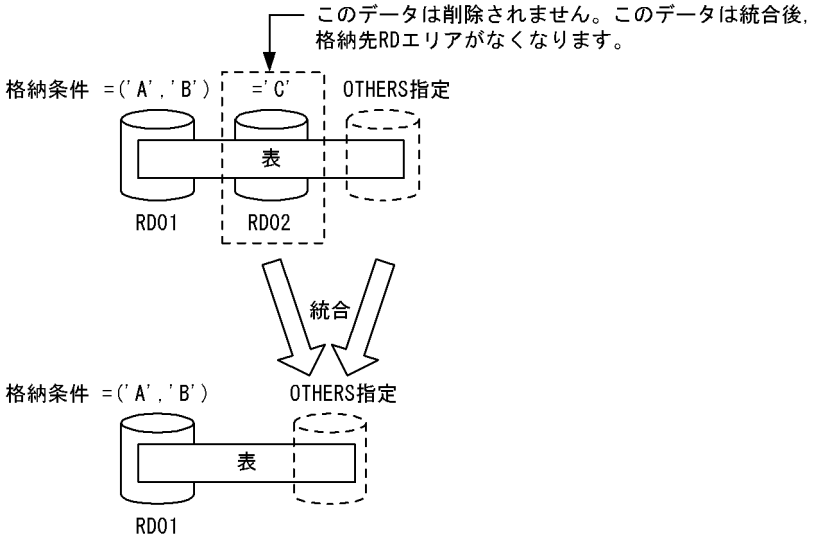


図 13–77　統合後にデータを格納する RD エリアがなくなるケース（その 2）



13.12.10　分割格納条件の変更方法（マトリクス分割の場合）

ALTER TABLE の CHANGE RDAREA で分割格納条件を変更します。なお，分割と統合を同時に実行することはできません。分割と統合を実行する場合は ALTER TABLE を 2 回実行してください。また，マトリクス分割の場合，変更できるのは境界値指定のキーレンジ分割の次元だけです。

(1)　分割格納条件の統合前

分割格納条件を統合，分割する前の状態です。

```
CREATE FIX TABLE ”USERA”.”ZAIKO”
(”GNO” CHAR(5),”GNAME” CHAR(8),”KIKAKU” CHAR(3),”TANKA” INTEGER,”NYUUKOBI” CHAR(10))
```

```
  PARTITIONED BY MULTIDIM
    ("GNO"(('20000')),"NYUUKOBI" (('2000-12-31'),('2001-12-31') ,('2002-12-31')))
    IN (("RDAT01","RDAT02","RDAT03","RDAT04"),("RDAT11","RDAT12","RDAT13","RDAT14"))
  CLUSTER KEY ("GNO","NYUUKOBI")
    IN (("RDAI01","RDAI02","RDAI03","RDAI04"),("RDAI11","RDAI12","RDAI13","RDAI14"))
```

		第2次元分割列（NYUKOBI）			
		～2000	～2001	～2002	その他
第1次元分割列（GNO）	20000以下	RDAT01 RDAI01	RDAT02 RDAI02	RDAT03 RDAI03	RDAT04 RDAI04
	その他	RDAT11 RDAI11	RDAT12 RDAI12	RDAT13 RDAI13	RDAT14 RDAI14

(2)　分割格納条件の統合

第 2 次元分割列 NYUKOBI の「～2000」と「～2001」を「～2001」に統合します。

```
ALTER TABLE "USERA"."ZAIKO" CHANGE RDAREA
MULTIDIM ("NYUUKOBI" (('2000-12-31'),('2001-12-31'))) AT (('2001-12-31'))
  INTO (("RDAT02"),("RDAT12"))
FOR CLUSTER KEY
  INTO (("RDAI02"),("RDAI12"))
```

		第2次元分割列（NYUKOBI）		
		～2001	～2002	その他
第1次元分割列（GNO）	20000以下	RDAT02 RDAI02	RDAT03 RDAI03	RDAT04 RDAI04
	その他	RDAT12 RDAI12	RDAT13 RDAI13	RDAT14 RDAI14

注　RDAT01，RDAT02，RDAT11，RDAT12のデータ，
　　及びRDAI01，RDAI02，RDAI11，RDAI12のキー値は削除されます。

(3)　分割格納条件の分割

第 2 次元分割列 NYUKOBI の「その他」を「～2003」と「その他」に分割します。

```
ALTER TABLE "USERA"."ZAIKO" CHANGE RDAREA
MULTIDIM ("NYUUKOBI" ((MAX))) AT (('2003-12-31'),(MAX))
  INTO (("RDAT04","RDAT01"),("RDAT14","RDAT11"))
FOR CLUSTER KEY
  INTO (("RDAI04","RDAI01"),("RDAI14","RDAI11"))
```

第1次元分割列（GNO）		第2次元分割列（NYUKOBI）			
		～2001	～2002	～2003	その他
	20000以下	RDAT02 RDAI02	RDAT03 RDAI03	RDAT04 RDAI04	RDAT01 RDAI01
	その他	RDAT12 RDAI12	RDAT13 RDAI13	RDAT14 RDAI14	RDAT11 RDAI11

注　RDAT04，RDAT14のデータ，及びRDAI04，RDAI14のキー値は削除されます。

13.12.11　RD エリアの分割（マトリクス分割の場合）

マトリクス分割表で分割キーとなるどちらかの次元で分割されたデータを複数の RD エリアに分割します。機能の詳細を次に示します。

(1)　上限値と下限値

分割機能での上限値と下限値を次の表に示します。

表 13–32　分割機能での上限値と下限値（マトリクス分割の場合）

項目	上限値	下限値	上限値又は下限値を超えた場合の動作
変更できる次元数	1（固定）		指定できません。
分割の対象にできる特定境界値格納 RD エリア群数※	1（固定）		
1 回の操作で分割できる数	16	2	ALTER TABLE をエラーとします。
分割した結果の総分割 RD エリア数（重複を含む）	4,096	4	

注※

特定境界値格納 RD エリア群数とは，ある次元の，特定の格納条件のデータを格納する RD エリア群の数です。例えば，次に示す例では，第 2 次元分割列の境界値 2000 に対する特定境界値格納 RD エリア群は RD12，RD22，RD32，及び RD42 となります。

第1次元分割列	第2次元分割列		
	1000	2000	3000
100	RD11	RD12	RD13
200	RD21	RD22	RD23
300	RD31	RD32	RD33
MAX	RD41	RD42	RD43

(2)　分割対象 RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された変更対象列名と境界値によって，その格納条件に合致する RD エリアを分割対象の RD エリアとします。変更対象列名と格納条件が指定されれば，対象となる RD エリアが特定できるため，RD エリアの指定は必要ありません。分割対象の次元は，ALTER TABLE の CHANGE

RDAREA に指定された変更対象列名で決定します。境界値についての，ALTER TABLE の指定と分割対象 RD エリアの決定方法を次の表に示します。

表 13–33　ALTER TABLE の指定と分割対象 RD エリアの決定方法（マトリクス分割の場合）

指定内容	条件	指定可否	分割対象 RD エリアの決定方法
境界値	表定義中に，変更対象列名に対応する境界値あり	○	格納条件に合致する特定境界値格納 RD エリア群の RD エリアを分割対象とします。
	表定義中に，変更対象列名に対応する境界値なし	×	–
'MAX'	–	○	最大境界値より大きい分割キー値のデータを格納している RD エリアを分割対象とします。

（凡例）

○：指定できます。

×：指定できません。

–：該当しません。

(3)　分割後の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された次元の，変更後の境界値リスト（ある次元に指定された境界値の組み合わせ）と RD エリアを対にして，格納する RD エリアを決定します。このとき，分割後の境界値の指定は，次の条件を満たす必要があります。

- 境界値の指定は昇順
- 指定した境界値の最大値は，分割前の境界値と等しい
- 分割したすべての境界値が，分割前の格納条件の範囲内である

ALTER TABLE の指定値と分割後の RD エリアの決定方法を次の表に示します。表で参照される，分割前後の境界値の条件を次の図に示します。

表 13–34　ALTER TABLE の指定値と分割後の RD エリアの決定方法（マトリクス分割の場合）

分割前の境界値	分割後の境界値の指定	条件	指定可否	分割対象 RD エリアの決定方法
最小境界値（図の 1.）	境界値	分割後の境界値が最小境界値以下の値（図の 1.，4.）	○	指定された格納条件と RD エリアを分割後の RD エリアとします。
		分割後の境界値が最小境界値より大きい値（図の 2.，3.，5.）	×	ALTER TABLE をエラーとします。
	'MAX'	–	×	
中間境界値又は最大境界値（図の 2.）	境界値	分割後の境界値が分割前の境界値の一つ前の境界値より小さい値（分割前の境界値が 200 の場合：図の 1.，4.）	×	ALTER TABLE をエラーとします。

分割前の境界値	分割後の境界値の指定	条件	指定可否	分割対象 RD エリアの決定方法
		分割後の境界値が分割前の境界値以下で，一つ前の境界値より大きい値（分割前の境界値が 200 の場合：200，図の 5.）	○	指定された格納条件と RD エリアを分割後の RD エリアとします。
		分割後の境界値が分割前の境界値より大きい値（分割前の境界値が 200 の場合：200 より大きい値）	×	ALTER TABLE をエラーとします。
	'MAX'	−	×	
'MAX'（図の 3.）	境界値	分割後の境界値が最大境界値以下の値（図の 1.，2.，4.，5.）	×	ALTER TABLE をエラーとします。
		分割後の境界値が最大境界値より大きい値（図の 3.）	○	指定された格納条件と RD エリアを分割後の RD エリアとします。
	'MAX'	−	×	分割後の最大境界値より大きいデータの格納対象を指定された RD エリアとします。

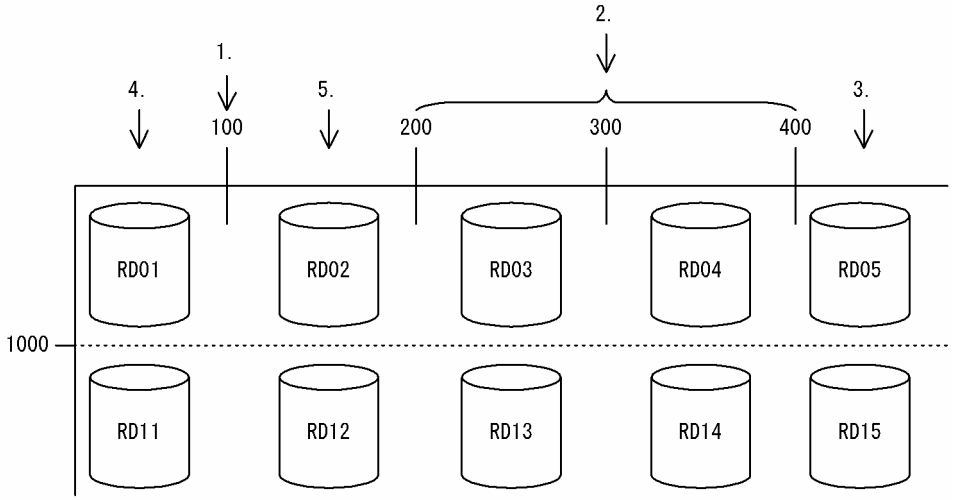
（凡例）

○：指定できます。

×：指定できません。

−：該当しません。

図 13–78　分割前後の境界値の条件（マトリクス分割の場合）



分割後の RD エリアは，分割前の RD エリアを再利用しても，新たに用意したものであってもかまいません。また，変更後の RD エリアは複数の格納範囲を同じ RD エリアに格納できます。

(4)　RD エリアの指定方法

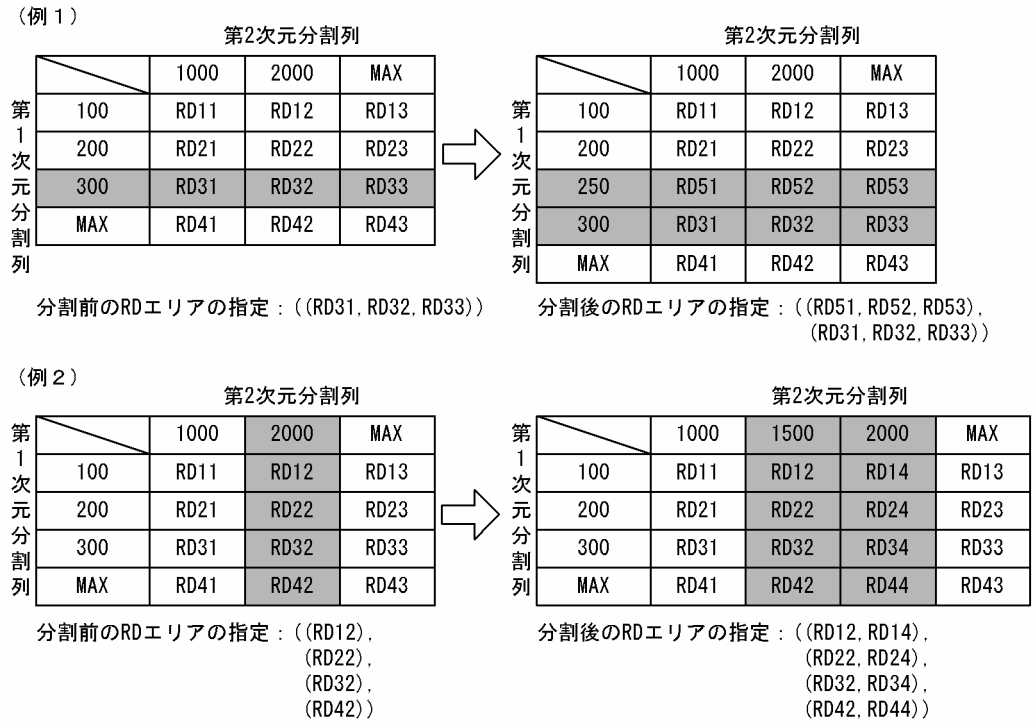
マトリクス分割表の場合，第 1 次元の RD エリアを分割するか，第 2 次元の RD エリアを分割するかで，ALTER TABLE の CHANGE RDAREA に指定する RD エリアの指定方法が異なります。分割対象となる特定境界値格納 RD エリア群数，及び一つの特定境界値格納 RD エリア群の RD エリア数は，分割対象ではない次

元の分割数に依存します。分割対象となる次元及び指定する RD エリア数の関係を次の表に，表で使用している例を次の図に示します。

表 13–35　分割対象となる次元及び指定する RD エリア数の関係

分割対象次元	分割対象の特定境界値格納 RD エリア群数	分割対象の特定境界値格納 RD エリア群ごとの RD エリア数
第 1 次元	変更後の境界値リストで指定した境界値の数 （図中の（例 1）の場合，2）	第 2 次元の分割数 （図中の（例 1）の場合，3）
第 2 次元	第 1 次元の分割数 （図中の（例 2）の場合，4）	変更後の境界値リストで指定した境界値の数 （図中の（例 2）の場合，2）

図 13–79　RD エリアの分割後の RD エリアの指定（マトリクス分割の場合）



(5)　表と表以外の RD エリアの対応

分割格納条件を変更する表に分割キーインデクスなどが定義されている場合，インデクスなどのデータを RD エリアに格納する必要があるため，表格納用 RD エリアと同様に分割しなければなりません。表と表以外の RD エリアの指定方法（マトリクス分割）を次の表に示します。表に示すリソースが複数定義されている場合はすべてが対象となります。正しく指定されていない場合，ALTER TABLE をエラーとします。

表 13–36　表と表以外の RD エリアの指定方法（マトリクス分割）

リソース名		指定方法
列	BLOB 列	表格納用 RD エリアと 1 対 1 で指定します。

リソース名		指定方法
インデクス	クラスタキーインデクス プライマリキーインデクス（プライマリキー及びクラスタキーを定義したインデクスを含む） B-tree インデクス	表格納用 RD エリアが重複指定されている場合は，表に対応するよう重複指定します。また，既存の表格納用 RD エリアを変更後に使用する場合は，同じ境界値に対応するように既存のインデクス，LOB 列格納用 RD エリアを指定しなければなりません。

(6) 分割する RD エリア内のデータの扱い

境界値による格納範囲を分割する場合，基本的にはシステムが自動的に RD エリアの該当する表のデータを削除します。ただし，条件によってはデータを削除しないで残すこともできます。

1. データの削除

   境界値による格納範囲を分割する場合，分割する特定境界値格納 RD エリア群のデータの一部は，分割後はその RD エリアに格納される格納範囲外のデータとなってしまう可能性があるため，分割対象とした特定境界値格納 RD エリア群の RD エリアのデータは削除します。ただし，削除するのは分割格納条件を変更する表のデータだけです（同じ RD エリアに含まれる別の表のデータは削除されません）。RD エリア内のデータを削除する場合，次のどちらかの方法でデータを削除します。

   - すべての定義情報の削除

     分割前の RD エリアが，分割した結果，分割対象の表で使用されなくなった場合，ディクショナリ中の，その表が格納条件ごとに使用する RD エリアの情報（SQL_DIV_TABLE 表）から，分割前の RD エリアの情報を削除します。また，RD エリア内で管理している表の情報も削除します。この結果，その RD エリア内にあった，分割対象の表のデータはすべて削除されます。イメージ的には特定 RD エリアに対して DROP TABLE を実行するのと同じです。

   - データだけの削除

     分割前の RD エリアを分割後も分割対象の表で使用する場合，ディクショナリの情報や RD エリア内で管理している情報は削除しないで，RD エリア内の分割対象とした表のデータだけを削除します。表のデータを削除する場合，分割対象の格納範囲の RD エリアがほかの格納範囲で使用されていると，その格納範囲のデータも削除します。イメージ的には特定 RD エリアに対して PURGE TABLE 文を実行するのと同じです。

   なお，RD エリア内のデータを削除する場合，対応する次の RD エリア内のデータもすべて削除します。

   - インデクス格納用 RD エリア内のインデクスキー
   - BLOB 列格納用 RD エリア内のデータ

2. データの保存

   「1.データの削除」で示したとおり，境界値による格納範囲を分割する場合，基本的には分割前の RD エリア内のデータは削除します。しかし，次の条件をすべて満たす場合は，RD エリア内のデータをそのまま使用できるため，データを削除しないようにすることもできます。

   - 分割前の特定境界値格納 RD エリア群を，分割後の特定境界値格納 RD エリア群の一部としてそのまま使用する場合

- 分割前の RD エリア内にある境界値以下のデータしかない場合
- 分割した結果，分割前の特定境界値格納 RD エリア群のすべてのデータが分割後の格納範囲と一致している場合

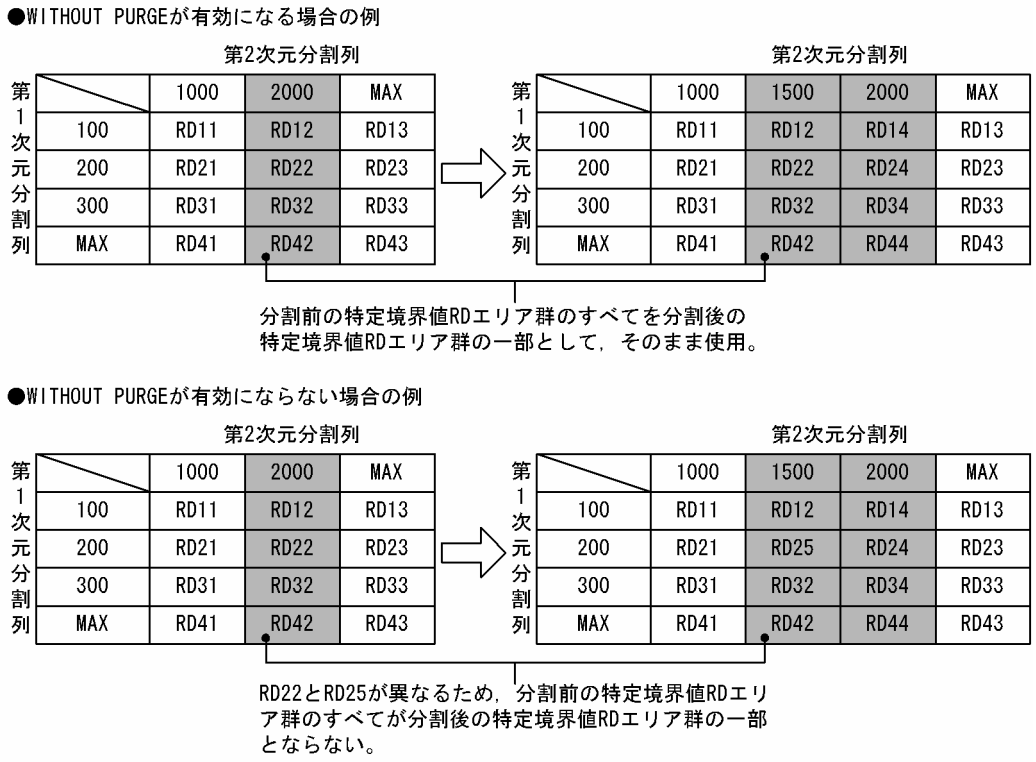
RD エリア内のデータを削除するかどうかは，ALTER TABLE の WITHOUT PURGE 句の指定によって選択できます。WITHOUT PURGE 句の指定とデータの扱いを次の表に示します。ただし，システムでは，分割した結果，RD エリア内のすべてのデータが分割後の格納範囲と一致しているかどうかをチェックしません。

表 13–37　WITHOUT PURGE 句の指定とデータの扱い（マトリクス分割の場合）

変更前後の RD エリアの関係	WITHOUT PURGE 句の指定可否	WITHOUT PURGE 句の指定あり	WITHOUT PURGE 句の指定なし
分割前の特定境界値格納 RD エリア群を，分割後の特定境界値格納 RD エリア群として使用します。	指定できます。	分割前の特定境界値格納 RD エリア群のデータを削除しません。	分割前の特定境界値格納 RD エリア群のデータを削除します。
分割前の特定境界値格納 RD エリア群を，分割後の特定境界値格納 RD エリア群として使用しません。	指定できません。	ALTER TABLE をエラーとします。	

WITHOUT PURGE 句の指定が有効になる場合と，有効にならない場合の例を次の図に示します。

図 13–80　WITHOUT PURGE 句の指定が有効になる場合と有効にならない場合の例



データを削除しない場合の注意事項

分割前の RD エリアのデータが，分割格納条件を変更した結果，格納範囲以外となった場合，ALTER TABLE 実行時はデータの妥当性をチェックしません。したがって，格納範囲外のデータが RD エリアにあることになり，SQL 実行時など HiRDB が正しく動作しません。

データを削除する場合の注意事項

分割前の RD エリアが，ほかの格納範囲でも格納されるような指定の場合，ほかの格納範囲のデータも削除します。分割前の RD エリアを分割後の RD エリアに含む場合も，含まない場合も同じです。

13.12.12　RD エリアの統合（マトリクス分割の場合）

(1)　上限値と下限値

統合機能での上限値と下限値を次の表に示します。

表 13–38　統合機能での上限値と下限値（マトリクスの場合）

項目	上限値	下限値	上限値又は下限値を超えた場合の動作
統合の対象にできる特定境界値格納 RD エリア群の数	16	2	ALTER TABLE がエラーとなります。
1 回の操作で統合した後の特定境界値格納 RD エリア群	1（固定）		
統合した結果の総分割 RD エリア数（重複を含む）	4,096	4	

(2)　統合対象の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された複数の変更対象列名と境界値によって，その格納条件に合致する RD エリアを統合対象の RD エリアとします。変更対象列名と格納条件が指定されれば，対象となる RD エリアが特定できるため，RD エリアの指定は必要ありません。このとき，複数の境界値の指定は，昇順である必要があります。また，表に定義されている格納条件の順番に指定されている必要があります。例えば，表定義に境界値が 10，20，30，40 と定義されている場合，10，30，40 など間を飛ばして指定すると ALTER TABLE をエラーとします。

ALTER TABLE の指定と統合対象 RD エリアの決定方法を次の表に示します。

表 13–39　ALTER TABLE の指定と統合対象 RD エリアの決定方法（マトリクス分割の場合）

指定内容	条件 1	条件 2	動作
境界値	表定義中に指定された境界値あり	定義されている境界値の順に指定されている	指定された格納条件に合うすべての RD エリアを統合対象とします。
		定義されている境界値の順に指定されていない	ALTER TABLE をエラーとします。
	表定義中に指定された境界値なし	なし	

指定内容	条件 1	条件 2	動作
'MAX'	なし	なし	最大境界値より大きい分割キー値のデータを格納している特定境界値格納 RD エリア群の RD エリアを統合対象とします。

(3) 統合後の RD エリアの決定方法

ALTER TABLE の CHANGE RDAREA に指定された次元の，変更後の境界値リスト（ある次元に指定された境界値の組み合わせ）と RD エリアを対にして，統合後の格納条件をすべて格納する RD エリアを決定します。このとき，統合後の境界値は統合前の境界値の最大値と同じである必要があります。

統合後の RD エリアは，統合前の RD エリアの中の一つであっても，新たに用意したものであってもかまいません。統合後の RD エリアの指定可否を次の表に示します。

表 13–40 統合後の RD エリアの指定可否（マトリクス分割の場合）

統合後の RD エリア		指定可否
統合前の RD エリアと同じ		指定できます。
統合前の RD エリアではない	統合対象の境界値以外で使用されている RD エリア	指定できます。
	統合対象の表では使用されていなかった RD エリア（新たに用意された RD エリア）	指定できます。

統合後の RD エリアは複数の格納条件に対して，同じ RD エリアに格納するように統合できます。

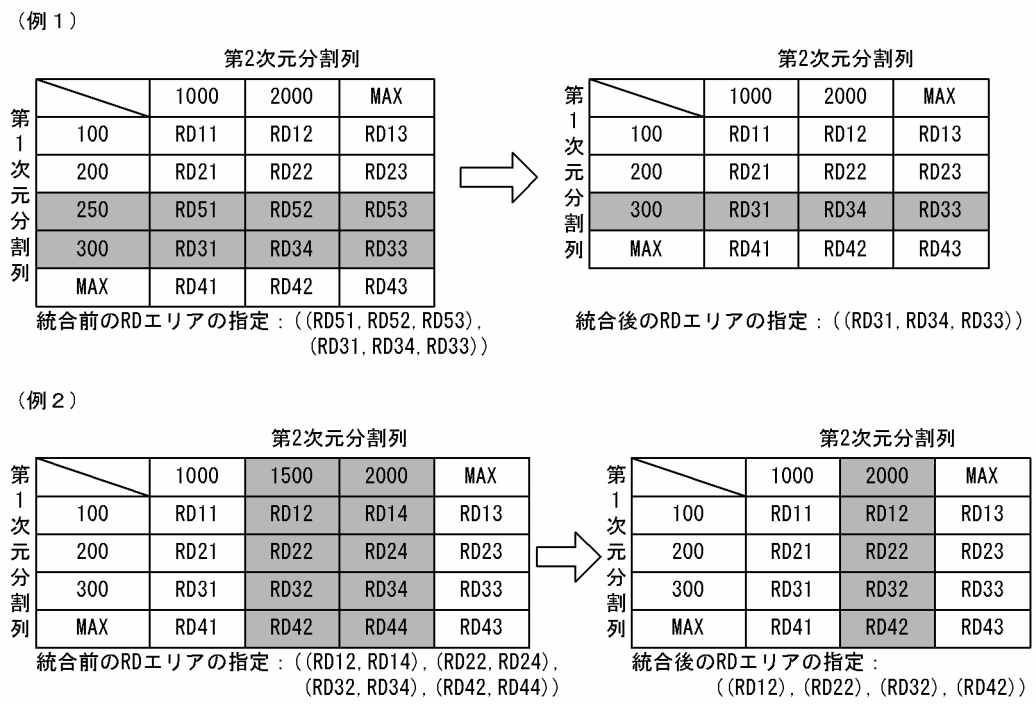
(4) RD エリアの指定方法

マトリクス分割表の場合，第 1 次元の RD エリアを統合するか，第 2 次元の RD エリアを統合するかで，ALTER TABLE の CHANGE RDAREA に指定する RD エリアの指定方法が異なります。統合対象となる特定境界値格納 RD エリア群数，及び一つの特定境界値格納 RD エリア群の RD エリア数は，統合対象ではない次元の分割数に依存します。統合対象となる次元及び指定する RD エリア数の関係を次の表に，表で使用している例を次の図に示します。

表 13–41 統合対象となる次元及び指定する RD エリア数の関係

統合対象次元	統合対象の特定境界値格納 RD エリア群数	統合対象の特定境界値格納 RD エリア群ごとの RD エリア数
第 1 次元	変更後の境界値リストで指定した境界値の数 （図中の（例 1）の場合，2）	第 2 次元の分割数 （図中の（例 1）の場合，3）
第 2 次元	第 1 次元の分割数 （図中の（例 2）の場合，4）	変更後の境界値リストで指定した境界値の数 （図中の（例 2）の場合，2）

図 13–81　RD エリアの統合後の RD エリアの指定（マトリクス分割の場合）



(5)　表と表以外の RD エリアの対応

分割格納条件を変更する表に分割キーインデクスなどが定義されている場合，表格納用 RD エリアと対になる RD エリアにインデクスなどのデータを格納する必要があります。表と表以外の RD エリアの指定方法（マトリクス分割）を次の表に示します。この表に示すリソースが複数定義されている場合は，すべてが対象となります。正しく指定されていない場合，ALTER TABLE をエラーとします。

表 13–42　表と表以外の RD エリアの指定方法（マトリクス分割）

リソース名		指定方法
列	BLOB 列	表格納用 RD エリアと 1 対 1 で指定します。 表格納用 RD エリアが重複指定されている場合は，表に対応するよう重複指定します。また，既存の表格納用 RD エリアを変更後に使用する場合は，同じ境界値に対応するように既存のインデクス，LOB 格納用 RD エリアを指定しなければなりません。
インデクス	クラスタインデクス プライマリインデクス（プライマリクラスタインデクスを含む） B-tree インデクス	

(6)　統合する RD エリアのデータの扱い

境界値による格納範囲を統合する場合，基本的には，システムが自動的に RD エリアのデータを削除します。ただし，条件によっては，データを削除しないで残すこともできます。

1. データの削除

   境界値による格納範囲を統合する場合，統合前の特定境界値格納 RD エリア群は，統合後はその表が使用する RD エリアではなくなる可能性があるため，RD エリア内のデータはシステムが自動的に削除し

ます。ただし，削除するのは分割格納条件を変更する表のデータだけです（同じ RD エリアに含まれる別の表のデータは削除されません）。RD エリア内のデータを削除する場合，次のどちらかの方法でデータを削除します。

- すべての定義情報の削除

  統合前の RD エリアが，統合した結果，統合対象の表で使用されなくなった場合，ディクショナリ中の，その表が格納条件ごとに使用する RD エリアの情報（SQL_DIV_TABLE 表）から，統合前の RD エリアの情報を削除します。また，RD エリア内で管理している，表の情報も削除します。この結果，その RD エリア内にあった，統合対象の表のデータはすべて削除されます。イメージ的には特定の RD エリアに DROP TABLE を実行するのと同じです。

- データだけの削除

  統合前の RD エリアを統合後も統合対象の表で使用する場合，ディクショナリの情報や RD エリア内で管理している情報は削除しないで，RD エリア内の統合対象とした表のデータだけを削除します。表のデータを削除する場合，統合対象の格納範囲の RD エリアがほかの格納範囲で使用されていると，その格納範囲のデータも削除します。イメージ的には特定 RD エリアに対して PURGE TABLE を実行するのと同じです。

なお，RD エリア内のデータを削除する場合，対応する次の RD エリア内のデータもすべて削除します。

- インデクス格納用 RD エリア内のインデクスキー
- BLOB 列格納用 RD エリア内のデータ

2. データの保存

「1.データの削除」で示したとおり，境界値による格納条件を統合する場合，基本的には統合前の RD エリア内のデータは削除します。しかし，次の条件を満たす場合は，RD エリア内のデータをそのまま使用できるため，データを削除しないようにすることもできます。

- 統合前の特定境界値格納 RD エリア群を，統合後の特定境界値格納 RD エリア群の一部としてそのまま使用する場合

RD エリア内のデータを削除するかどうかは，ALTER TABLE の WITHOUT PURGE 句の指定によって選択できます。WITHOUT PURGE 句の指定とデータの扱いを次の表に示します。ただし，システムでは，統合した結果，RD エリア内のすべてのデータが統合後の格納範囲と一致しているかどうかをチェックしません。

表 13–43　WITHOUT PURGE 句の指定とデータの扱い（マトリクス分割の場合）

変更前後の RD エリアの関係	WITHOUT PURGE 句の指定可否	WITHOUT PURGE 句の指定あり	WITHOUT PURGE 句の指定なし
統合前の特定境界値格納 RD エリア群を，統合後の特定境界値格納 RD エリア群に含みます。	指定できます。	統合後の特定境界値格納 RD エリア群の RD エリアのデータを削除しません。	統合前の特定境界値格納 RD エリア群のすべてのデータを削除します。
統合前の特定境界値格納 RD エリア群を，統合後の特定境	指定できません。	統合前の特定境界値格納 RD エリア群のすべてのデータを削除します。	

変更前後の RD エリアの関係	WITHOUT PURGE 句の指定可否	WITHOUT PURGE 句の指定あり	WITHOUT PURGE 句の指定なし
界値格納 RD エリア群に含みません。			

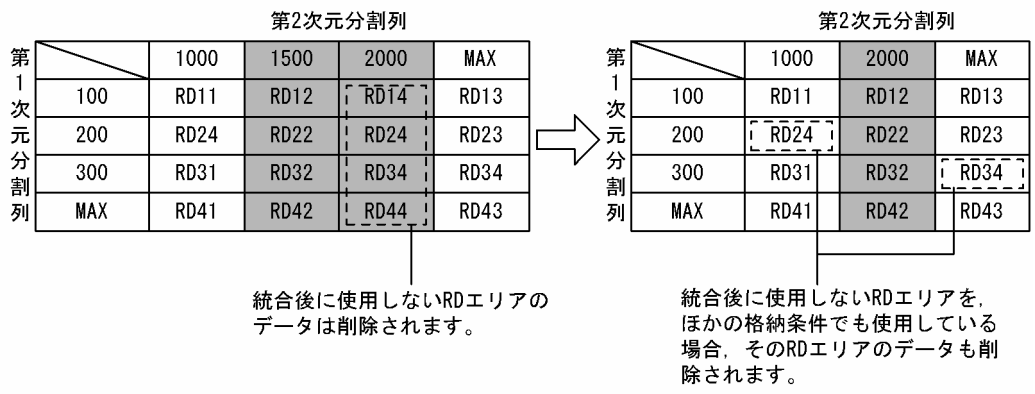
3. データを削除しない場合の注意事項

WITHOUT PURGE を指定してデータを保存した場合，統合前のすべてのデータをアンロードし，統合後の RD エリアにロードすると，保存したデータが二重に登録されてしまいます。このため，WITHOUT PURGE を指定してデータを保存する RD エリアのデータはアンロード，及び統合後のロードを実行してはなりません。

WITHOUT PURGE を指定しても，統合前の特定境界値格納 RD エリア群の中で，統合後の特定境界値格納 RD エリア群にしなかった RD エリアのデータはすべて削除します。また，統合する格納条件以外でも使用されている RD エリアがある場合，統合後にその RD エリアを使用しないときは，ほかの格納条件値のデータも削除します。このため，複数の格納範囲で使用されている RD エリアの中で特定の格納範囲を統合する場合は，WITHOUT PURGE を指定しないで分割格納条件を変更し，その後アンロードしていたデータを統合した RD エリアにロードする必要があります。

変更対象 RD エリアのデータの扱い（WITHOUT PURGE 指定がある場合）を次の図に示します。

図 13–82　変更対象 RD エリアのデータの扱い（WITHOUT PURGE 指定がある場合）



4. データを削除する場合の注意事項

データを削除しない場合の注意事項と同じように，統合する格納条件以外でも使用されている RD エリアが，統合後の RD エリアではなかった場合，ほかの格納範囲のデータも削除します。このため，統合する格納範囲以外でも統合対象の RD エリアを使用している場合は，統合対象の RD エリアのデータをすべてアンロードし，WITHOUT PURGE を指定しないで分割格納条件を変更し，その後アンロードしていたデータを統合した RD エリアにデータロードすることを推奨します。

13.12.13　ほかの機能との関連

(1)　関連リソースの無効化

分割格納条件を変更する表にルーチンが定義されている場合，ルーチンを無効化します。ルーチンにはトリガ定義で作成されたルーチンも含みます（参照制約動作で HiRDB が内部的に作成したトリガのルーチンも含みます）。

無効になったルーチンは，ALTER ROUTINE で再作成する必要があります。

(2)　参照制約及び検査制約

被参照表，参照表，及び検査制約表に対する分割格納条件変更時の運用については，「参照制約及び検査制約を使用している場合の運用」を参照してください。

(3)　ユティリティとの同時実行

データベース作成ユティリティ，データベース再編成ユティリティなどのデータベースを更新するユティリティを実行中に，分割格納条件を変更することはできません。変更した場合，分割格納条件に合わないデータが，データベースに登録されてしまうことがあります。

13.13　表の分割格納条件を変更するときの運用

表の分割格納条件を変更するときの運用方法について説明します。

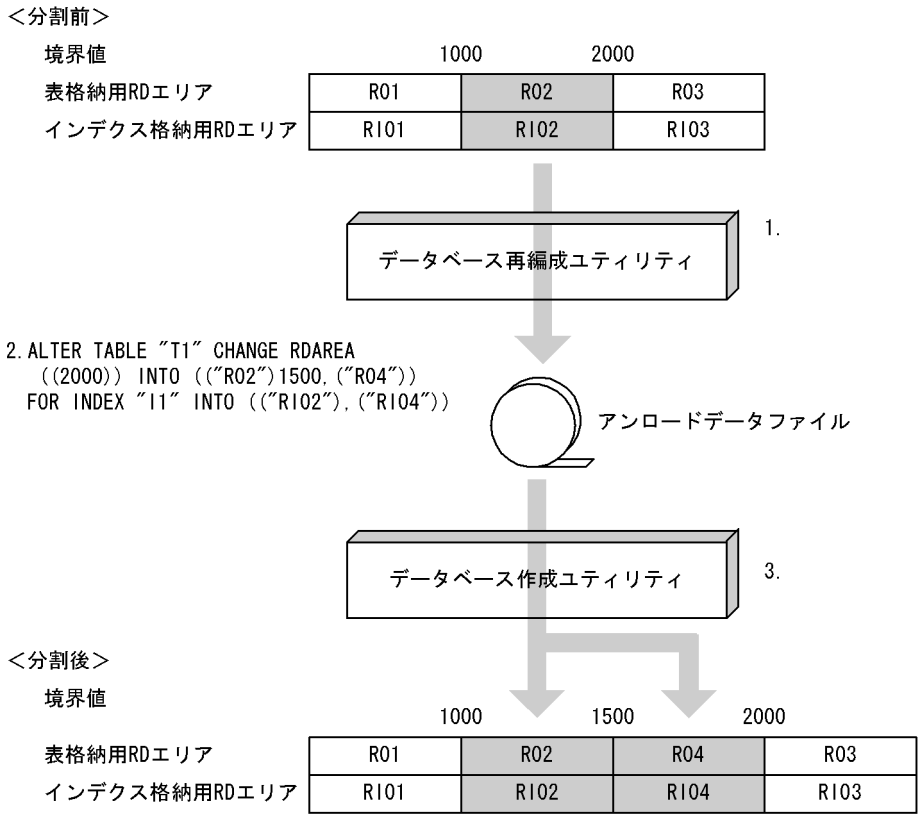
13.13.1　例題（境界値指定の場合）

表の分割格納条件を変更する手順（境界値指定の場合）を説明します。

なお，表の分割格納条件を変更する前にバックアップを取得してください。

(1)　例題 1（分割時の基本的な運用）

ある格納範囲の RD エリアの空き容量が不足したため，新たな RD エリアを用意してデータの格納範囲を分割する場合の運用について説明します。分割後の RD エリアは，元々使用していたものを使用しても，すべて新たな RD エリアを用意してもかまいません。横分割表の分割時の基本的な運用の例（元々使用していたものを使用）を説明します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で格納条件を変更する RD エリア内の表データ（R02）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。

2. ALTER TABLE で分割したい境界値を分割します。このとき，分割対象の RD エリア（R02，RI02）のデータは削除されます。

3. 分割した境界値を格納する RD エリアに対して，1.で作成したアンロードデータファイルを入力ファイルとして，分割後のすべての RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。

4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。

5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(2) 例題 2（分割前の RD エリアを分割後に使用してデータを削除しないで分割する運用）

時系列に増加する RD エリア内のデータに対して，今後のために新たな RD エリアを用意しておく運用について説明します。この場合，分割対象の RD エリアには，分割後の境界値より大きいデータが格納されていないことが前提となります。

分割前の RD エリアを分割後に使用してデータを削除しないで分割する運用の例を説明します。この例は，登録日を分割キーとしている表であり，R03 には 2005 年以降のデータが格納されていないことが前提です。この表に対して 2005 年以降のデータを格納する新たな RD エリアを用意する場合の運用例です。

＜分割前＞

1. SELECT MAX("C1") FROM "T1"
   WHERE "C1">'2004-12-31'
   でデータが1件もヒットしない（2005年以降のデータがない）ことを確認します。

境界値(分割キー="C1")	'2002-12-31'	'2003-12-31'	
表格納用RDエリア	R01	R02	R03
インデクス格納用RDエリア	RI01	RI02	RI03

2. ALTER TABLE "T1" CHANGE RDAREA
   ((MAX)) INTO (("R03")'2004-12-31',("R04"))
   FOR INDEX "I1" INTO (("RI03"),("RI04"))
   WITHOUT PURGE

＜分割後＞

境界値	'2002-12-31'	'2003-12-31'	'2004-12-31'	
表格納用RDエリア	R01	R02	R03	R04
インデクス格納用RDエリア	RI01	RI02	RI03	RI04

注　1.～2.は，手順の項番に対応しています。

〈手順〉

1. 分割対象の境界値を含む RD エリアが分割後の格納条件のデータしかないことを確認します（SELECT MAX(分割キー列名) FROM 表名 WHERE 分割キー列名>分割後の境界値）。分割後の格納条件以外の

データ（2005 年以降のデータ）がある場合は，「例題 1（分割時の基本的な運用）」で実施してください。

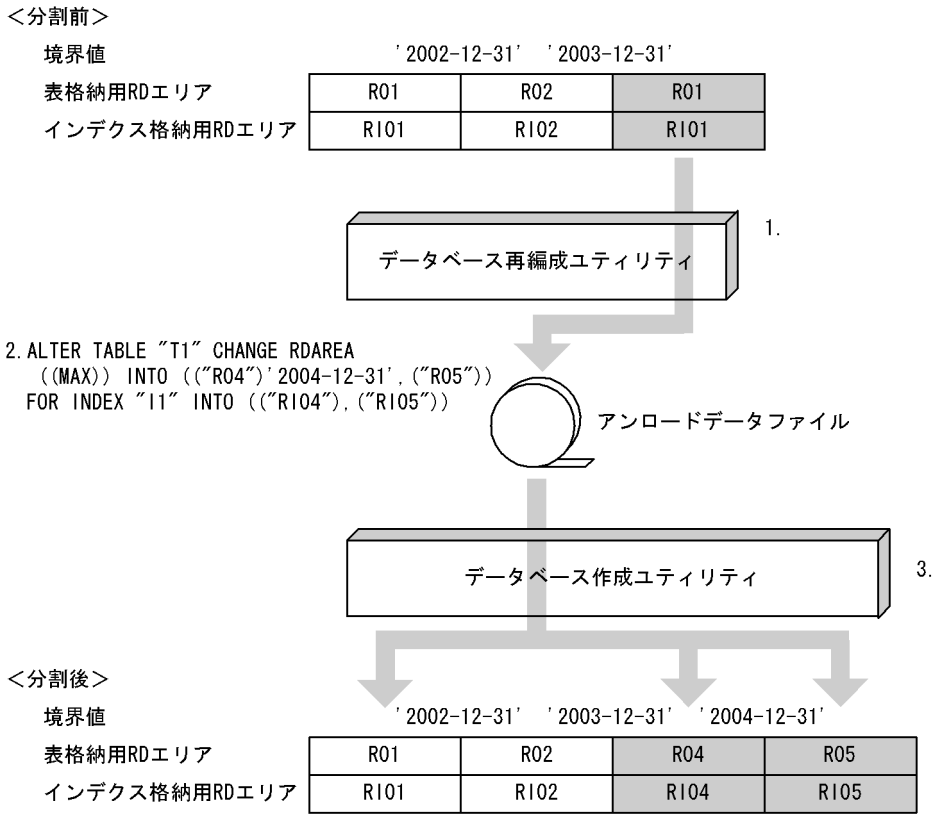
この手順を実施しなかった結果，分割後の格納条件に合わないデータが残った場合の回復手順については，「分割後の格納条件に合わないデータが残った場合の回復手順」を参照してください。

2. ALTER TABLE で分割したい境界値を分割します。このとき，WITHOUT PURGE を指定します。
3. 2.で無効になった，ルーチン，トリガを ALTER ROUTINE で再作成します。

(3) 例題 3（分割対象の RD エリアがほかの境界値で使用されている場合の運用）

分割対象の RD エリアがほかの境界値でも使用されている場合，ほかの境界値のデータも削除されます。このため，分割対象の RD エリアがほかの境界値でも使用されている場合，分割前にデータベース再編成ユティリティ（pdrorg）でアンロードしたデータをデータベース作成ユティリティ（pdload）でデータロードするときに注意が必要です。

分割対象の RD エリアがほかの境界値で使用されている場合の運用例を説明します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

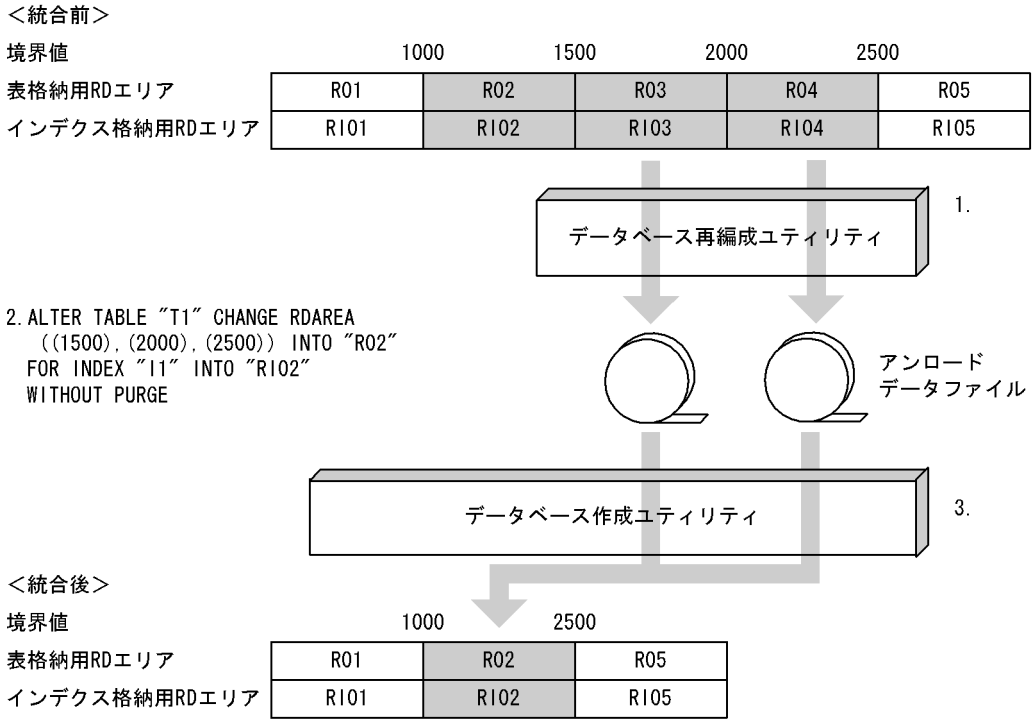
1. データベース再編成ユティリティ（pdrorg）で格納条件を変更する RD エリア内の表データをデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。このとき，'2002-12-31'以前のデータもアンロードされます。
2. ALTER TABLE で分割したい境界値を分割します。このとき，分割対象の RD エリア（R01，RI01）のデータは（'2002-12-31'以前のデータも）削除されます。
3. 分割した境界値を格納する RD エリアに対して，1.で作成したアンロードデータファイルを入力として，分割後のすべての RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。このとき，R01 のデータも 2.の時点でデータが削除されているため，R01 に対してもデータロードする必要があります。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(4) 例題 4（統合前の一つの RD エリアを統合後の RD エリアとして使用する場合の運用）

ある連続した格納範囲の RD エリアに空き容量が多いため，一つの RD エリアにデータの格納範囲を統合する場合の運用について説明します。統合後の RD エリアは，統合前の一つの RD エリアを使用しても使用しなくてもかまいません。

統合前の一つの RD エリアを統合後の RD エリアとして使用する場合の運用例を説明します。



注　1.～3.は，手順の項番に対応しています。

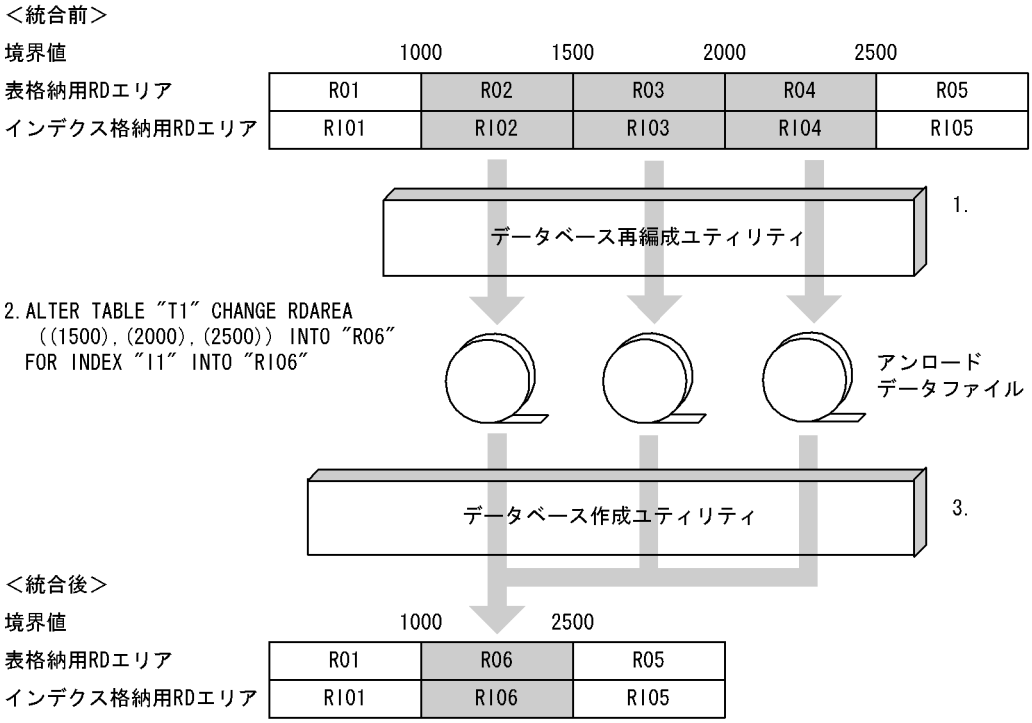
〈手順〉

1. データベース再編成ユティリティ（pdrorg）で統合対象の RD エリアのうち，統合後に使用しないすべての RD エリア内の表データ（R03，R04）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で統合したい境界値を統合します。このとき，R02 は統合後もそのまま使用するため，WITHOUT PURGE を指定します。
3. 統合した境界値を格納する RD エリアに対して，1.で作成したすべてのアンロードデータファイルを入力として，統合後の RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。このとき，作成モードでデータロードすると，統合前の R02 のデータが削除されてしまうため，注意が必要です。誤って作成モードでデータロードした場合は，ALTER TABLE 実行前の状態にバックアップから回復して，最初からし直します。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(5) 例題 5（統合前の RD エリアを統合後の RD エリアとして使用しない場合の運用）

ある連続した格納範囲の RD エリアに空き容量が多いため，一つの RD エリアにデータの格納範囲を統合する場合の運用について説明します。統合後の RD エリアは，統合前の一つの RD エリアを使用しても使用しなくてもかまいません。

統合前の RD エリアを統合後の RD エリアとして使用しない場合の運用例を説明します。



注　1.～3.は，手順の項番に対応しています。

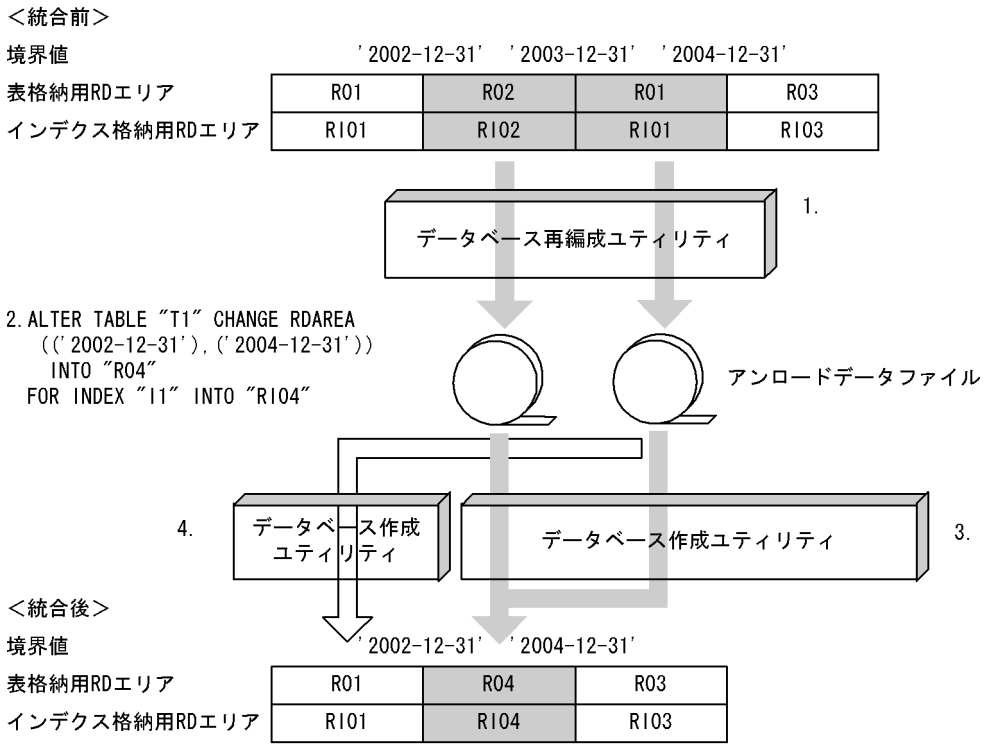
〈手順〉

1. データベース再編成ユティリティ（pdrorg）で統合対象のすべての RD エリア内の表データ（R02，R03，R04）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で統合したい境界値を統合します。
3. 統合した境界値を格納する RD エリアに対して，1.で作成したすべてのアンロードデータファイルを入力として，統合後の RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(6) 例題 6（統合対象の RD エリアがほかの境界値で使用されている場合の運用）

統合対象の RD エリアがほかの境界値でも使用されている場合，ほかの境界値のデータも削除されます。このため，統合後にデータが削除される RD エリアに対しては，統合前にデータベース再編成ユティリティ（pdrorg）でアンロードしたデータをデータベース作成ユティリティ（pdload）でデータロードするときに注意が必要です。

統合対象の RD エリアがほかの境界値で使用されている場合の運用例を説明します。



注　1.～4.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で統合対象のすべての RD エリア内の表データをデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。このとき，'2002-12-31'以前のデータもアンロードされます。
2. ALTER TABLE で統合したい境界値を統合します。このとき，ほかの境界値で使用されている RD エリア（R01）のデータも削除されます。
3. 統合した境界値を格納する RD エリアに対して，1.で作成したすべてのアンロードデータファイルを入力として，統合後の RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を実行します。
4. ほかの境界値で使用されている R01 のデータは削除されているため，R01 をアンロードしたアンロードデータファイルを入力として，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。

5. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
6. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

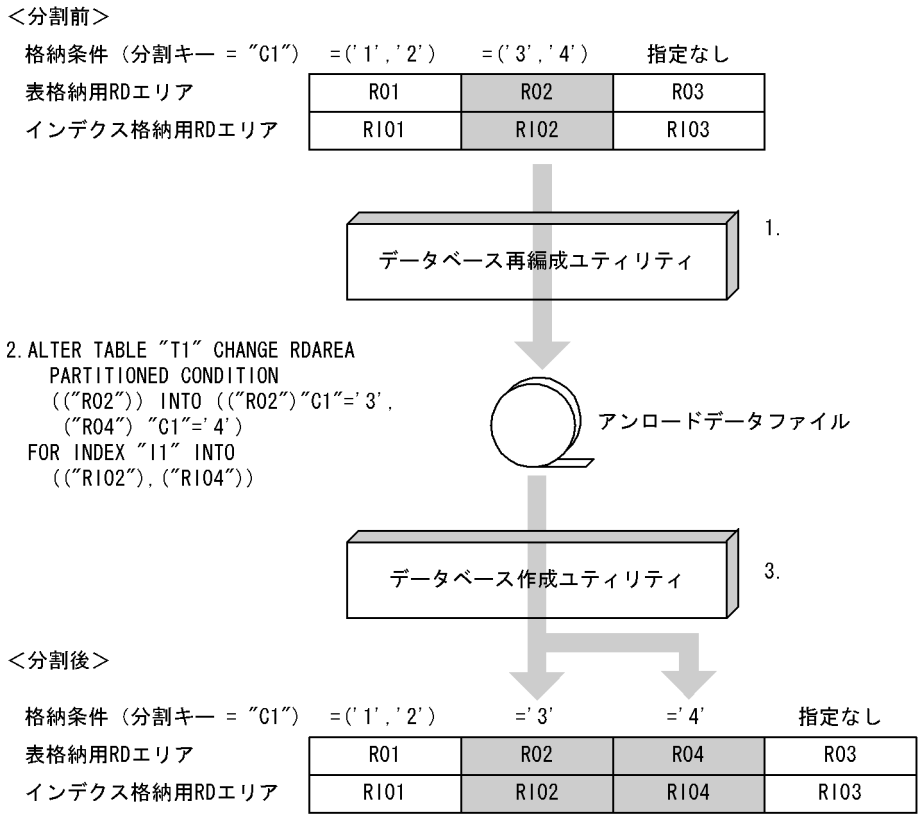
13.13.2　例題（格納条件指定の場合）

表の分割格納条件を変更する手順（格納条件指定の場合）を説明します。

なお，表の分割格納条件を変更する前にバックアップを取得してください。

(1)　例題 1（複数の値を持つ格納条件を指定している RD エリアを分割する）

RD エリア（R02）の容量が不足してきたため，分割格納条件を変更して RD エリアを分割します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で格納条件を変更する RD エリア内の表データ（R02）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で RD エリアを分割します。このとき，分割対象の RD エリア（R02，RI02）のデータは削除されます。

3. データベース作成ユティリティ（pdload）の追加モードで，RD エリア（R02）にデータロードします。1 の操作で作成したアンロードデータファイルを入力ファイルとし，RD エリア単位にデータロードしてください。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。

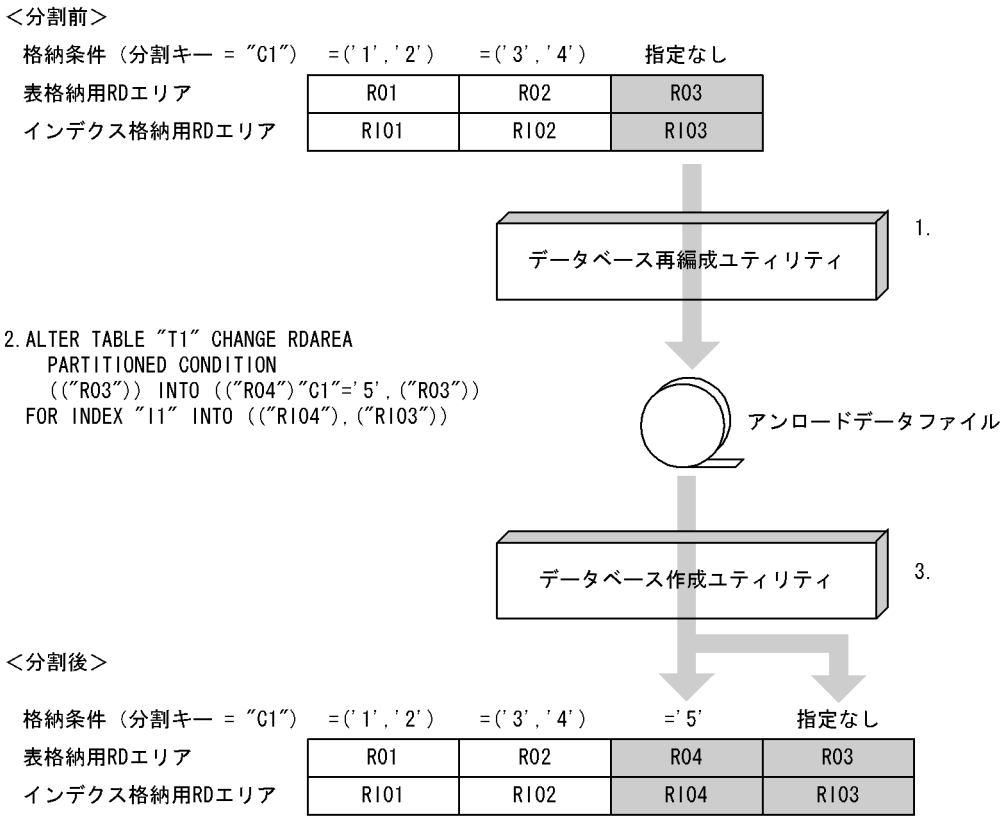
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。

5. 2 の操作で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(2)　例題 2（分割格納条件を追加する）

支店コードを分割キーとしている表の分割格納条件（支店コードが '5'の分割格納条件）を追加します。次のように分割格納条件を追加します。

- 支店コードが '5'のデータを格納するための新たな RD エリア（R04，RI04）を用意する
- 格納条件なしの RD エリア（R03）から支店コードが'5'のデータを抽出し，RD エリア（R04）に格納する
- 分割前に使用していた RD エリアを分割後にも使用する



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で格納条件を変更する RD エリア内の表データ（R03）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で RD エリアを分割します。このとき，分割対象の RD エリア（R03，RI03）のデータは削除されます。
3. データベース作成ユティリティ（pdload）の追加モードで，RD エリア（R03，R04）にデータロードします。1 の操作で作成したアンロードデータファイルを入力ファイルとし，RD エリア単位にデータロードしてください。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2 の操作で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(3) 例題 3（分割前の RD エリアを分割後に使用してデータを削除しないで分割する）

支店コードを分割キーとしている表の分割格納条件を変更します。次のように分割を行います。

- データを削除しないで分割する
- 支店コードが '5'のデータを格納するための新たな RD エリア（R04，RI04）を用意する
- 分割前に使用していた RD エリアを分割後にも使用する

なお，この例は，RD エリア（R03）に支店コードが '5'のデータが格納されていないことを前提としています。

＜分割前＞

```
1.SELECT "C1" FROM "T1"
    WHERE "C1"='5'
  でデータが1件もヒットしないことを確認します。
```

格納条件（分割キー = "C1"）	=('1','2')	=('3','4')	指定なし
表格納用RDエリア	R01	R02	R03
インデクス格納用RDエリア	RI01	RI02	RI03

```
2.ALTER TABLE "T1" CHANGE RDAREA PARTITIONED CONDITION
   (("R03")) INTO (("R04")"C1"='5',("R03"))
  FOR INDEX "I1" INTO (("RI04"),("RI03"))
  WITHOUT PURGE
```

＜分割後＞

格納条件（分割キー = "C1"）	=('1','2')	=('3','4')	='5'	指定なし
表格納用RDエリア	R01	R02	R04	R03
インデクス格納用RDエリア	RI01	RI02	RI04	RI03

注　1.～2.は，手順の項番に対応しています。

〈手順〉

1. 追加する格納条件に一致するデータ（格納条件 ='5'のデータ）が分割対象表に格納されていないことを確認します。

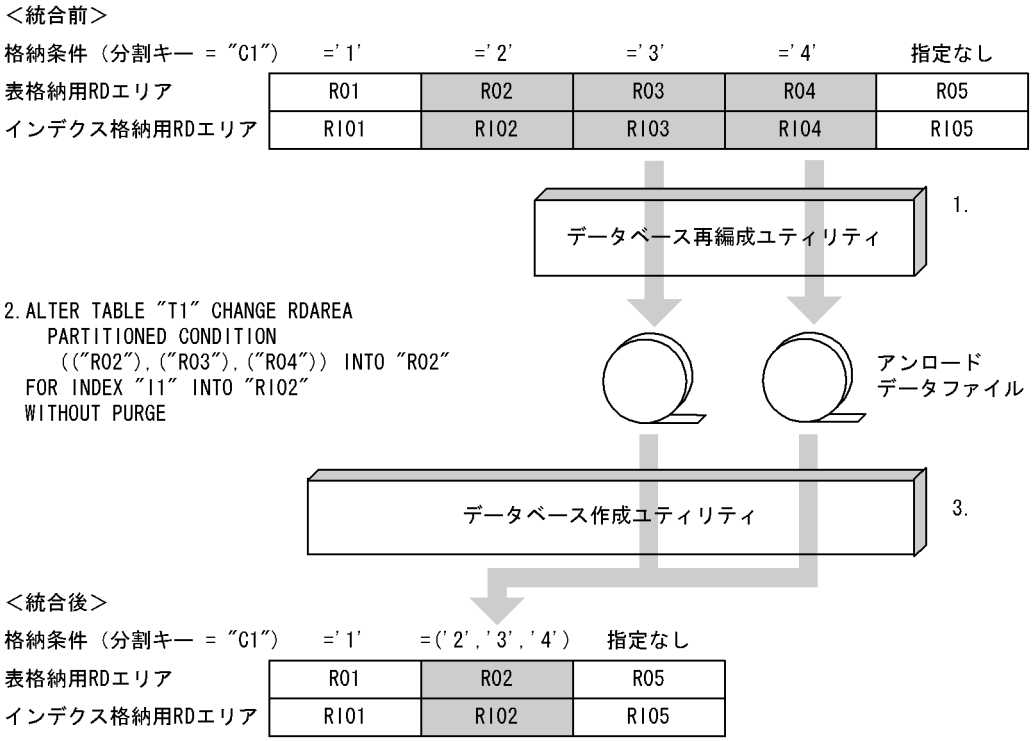
   格納条件 ='5'のデータがある場合は，「例題 2（分割格納条件を追加する）」の方法で分割してください。この方法で分割しなかった結果，分割後に格納条件と合わないデータが残ってしまった場合は，「分割後の格納条件に合わないデータが残った場合の回復手順」に示す方法でデータを回復してください。

2. ALTER TABLE で RD エリアを分割します。このとき，WITHOUT PURGE を指定します。

3. 2 の操作で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(4)　例題 4（分割格納条件を変更して RD エリアを統合する）

RD エリア（R02，R03，R04）に分割していた格納条件を R02 に統合します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で，RD エリア（R03，R04）内のデータをアンロードします。データベース作成ユティリティ（pdload）で入力できる形式でアンロードしてください。

2. ALTER TABLE で RD エリアを統合します。このとき，R02 は統合後もそのまま使用するため，WITHOUT PURGE を指定します。

3. データベース作成ユティリティ（pdload）の追加モードで，RD エリア（R02）にデータロードします。1 の操作で作成したアンロードデータファイルを入力ファイルとし，RD エリア単位にデータロードしてください。

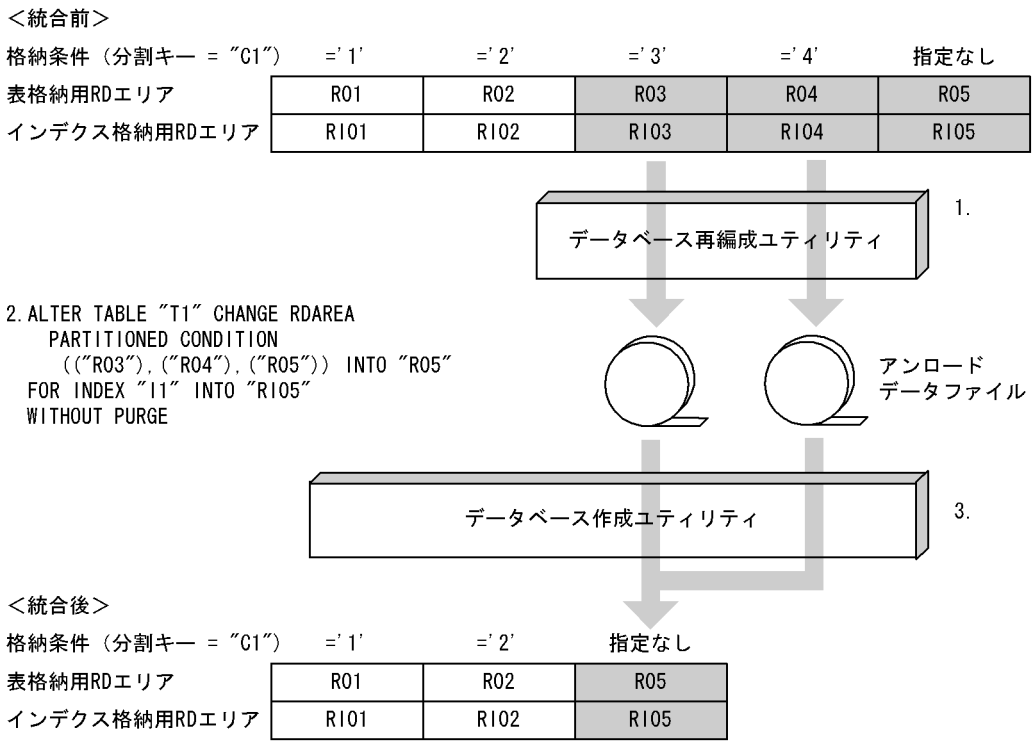
   なお，作成モードでデータロードをすると，統合前の R02 のデータが削除されてしまいます。誤って作成モードで実行してしまった場合は，ALTER TABLE 実行前の状態にバックアップから回復し，最初から操作をし直してください。

4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。

5. 2 の操作で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(5)　例題 5（分割格納条件を削除する）

RD エリア（R03，R04，R05）に分割していた格納条件を R05 に統合し，R03，R04 の格納条件を削除します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

格納条件を削除する RD エリア（R03，R04）内のデータが必要ない場合，1 と 3 の操作（アンロードとデータロード）は必要ありません。

1. データベース再編成ユティリティ（pdrorg）で，RD エリア（R03，R04）内のデータをアンロードします。データベース作成ユティリティ（pdload）で入力できる形式でアンロードしてください。

2. ALTER TABLE で RD エリアを統合します。

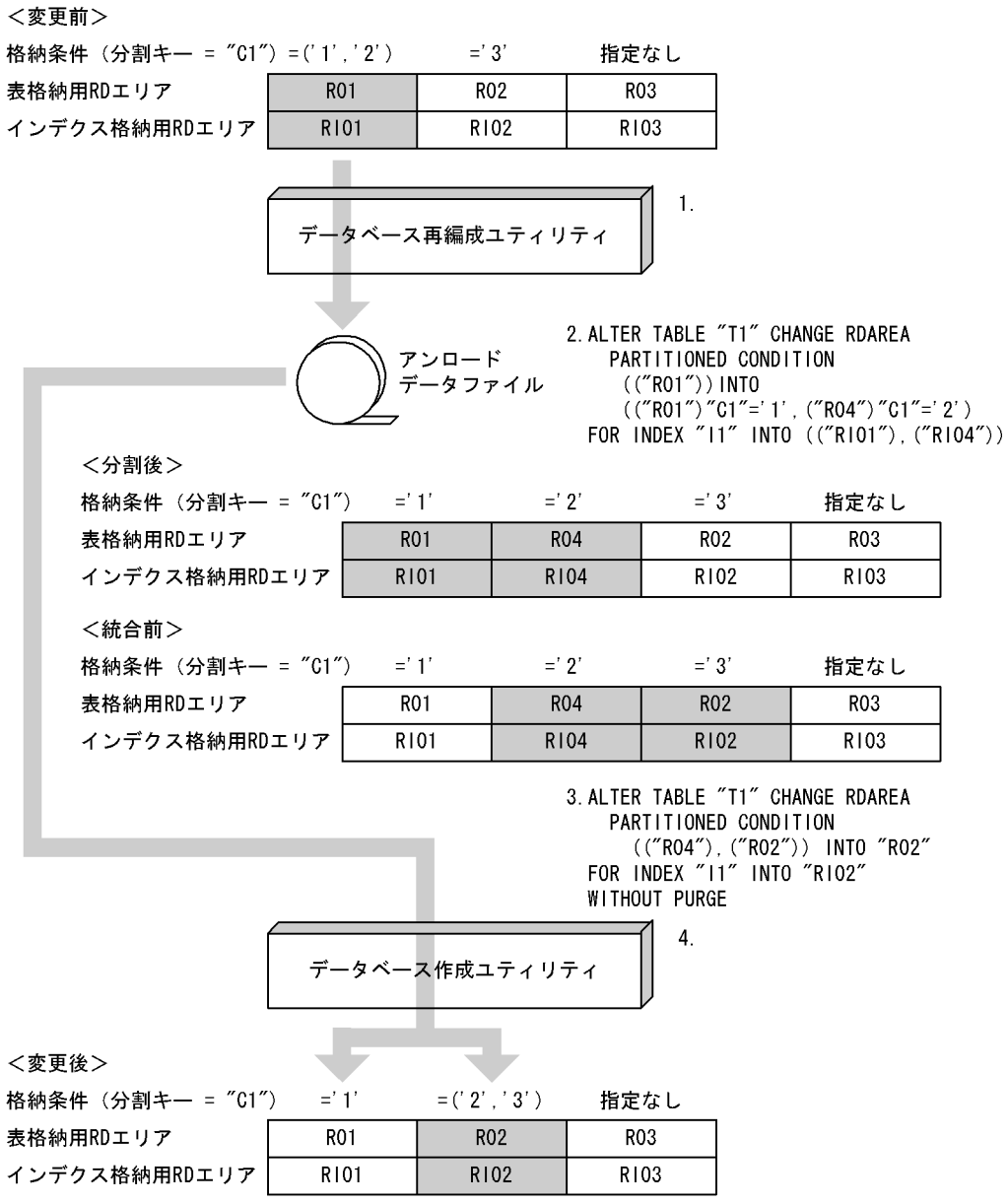
3. データベース作成ユティリティ（pdload）の追加モードで，RD エリア（R05）にデータロードします。1 の操作で作成したアンロードデータファイルを入力ファイルとし，RD エリア単位にデータロードしてください。

4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。

5. 2 の操作で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(6) 例題 6（分割と統合を連続して行う）

RD エリア（R01）を R01 と R04 に分割した後に，RD エリア（R02 と R04）を R02 に統合します。



注　1.～4.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で，RD エリア（R01）内のデータをアンロードします。データベース作成ユティリティ（pdload）で入力できる形式でアンロードしてください。
2. ALTER TABLE で RD エリアを分割します。
3. ALTER TABLE で RD エリアを統合します。
4. データベース作成ユティリティ（pdload）の追加モードで，RD エリア（R01，R02）にデータロードします。1 の操作で作成したアンロードデータファイルを入力ファイルとし，RD エリア単位にデータロードしてください。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。
5. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
6. 2 及び 3 の操作で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

> **参考**
>
> 通常，3 の操作の前に R02 及び R04 内のデータのアンロードが必要になります。しかし，この例のように 2 回の SQL で分割と統合を連続して実行する場合は，1 の操作で作成したアンロードデータファイルを 4 のデータロードの入力情報に使用できるため，3 の操作の前に必要なアンロードが不要になります。

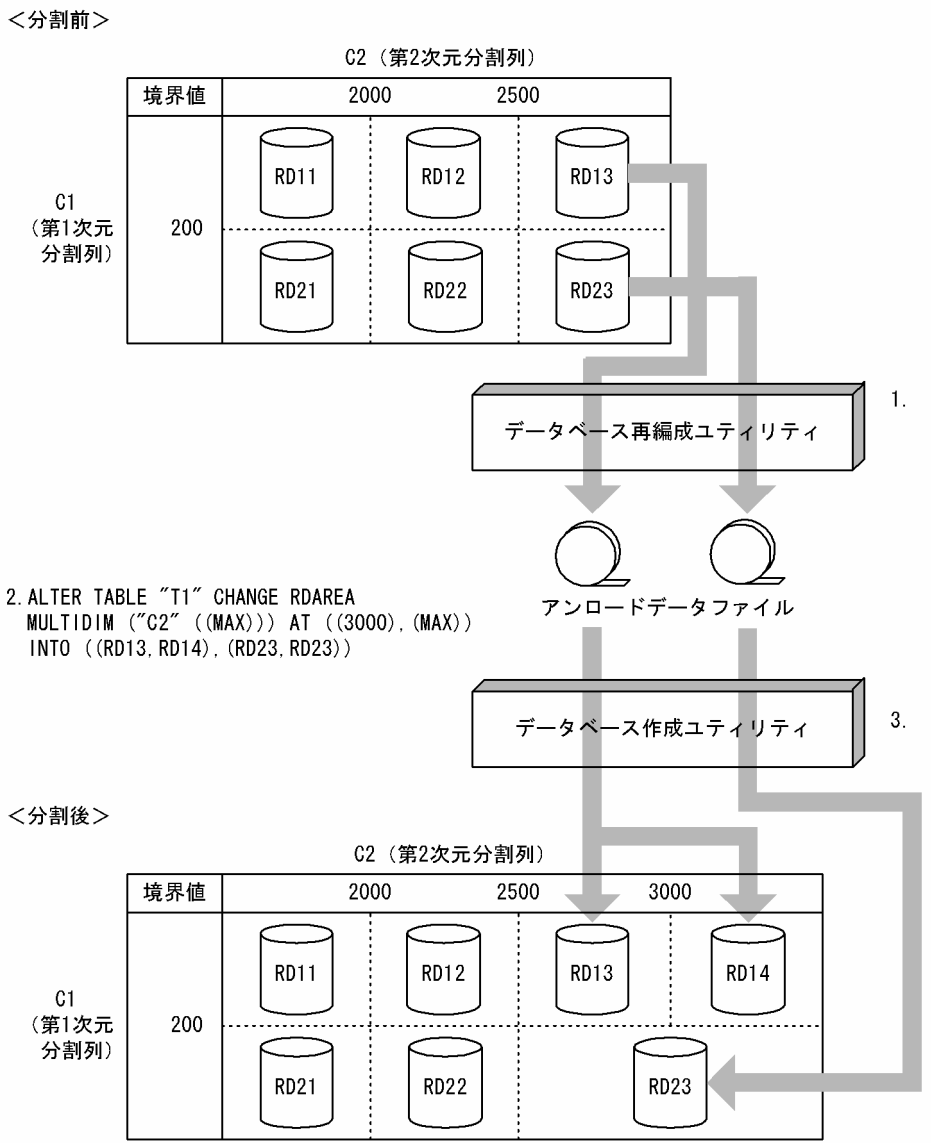
13.13.3　例題（マトリクス分割の場合）

表の分割格納条件を変更する手順（マトリクス分割の場合）を説明します。

なお，表の分割格納条件を変更する前にバックアップを取得してください。

(1)　例題 1（分割時の基本的な運用）

ある格納範囲の RD エリアの空き容量が不足したため，新たな RD エリアを用意してデータの格納範囲を分割する場合の運用について説明します。分割後の RD エリアは，元々使用していたものを使用しても，すべて新たな RD エリアを用意してもかまいません。マトリクス分割表の分割時の基本的な運用の例（元々使用していたものを使用）を説明します。この例では，RD13 の空き容量が不足したため，第 2 次元の RD13 に対する格納範囲を分割し，分割後の格納範囲を RD13 と，新たに用意した RD エリア（RD14）に格納します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で格納条件を変更する RD エリア内の表データ（RD13，RD23）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で分割したい境界値を分割します。このとき，分割対象の RD エリア（RD13，RD23）のデータは削除されます。
3. 分割した境界値を格納する RD エリアに対して，1.で作成したアンロードデータファイルを入力ファイルとして，分割後のすべての RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。

4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。

5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

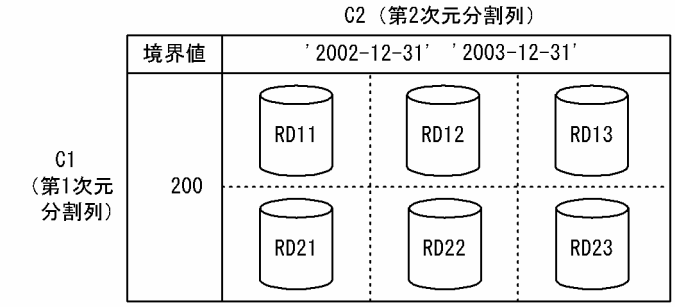
(2) 例題 2（分割前の RD エリアを分割後に使用してデータを削除しないで分割する運用）

時系列に増加する RD エリア内のデータに対して，今後のために新たな RD エリアを用意しておく運用について説明します。この場合，分割対象の RD エリアには，分割後の境界値より大きいデータが格納されていないことが前提となります。

分割前の RD エリアを分割後に使用してデータを削除しないで分割する運用の例を説明します。この例は，登録日を分割キーとしている表であり，RD13 及び RD23 には 2005 年以降のデータが格納されていないことが前提です。第 2 次元の RD13 に対する格納範囲を分割し，新たに用意した RD エリア（RD14 及び RD24）に 2005 年以降のデータを格納します。

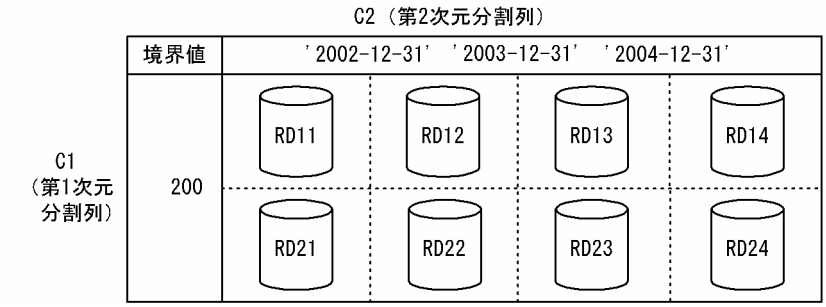
＜分割前＞

1. SELECT MAX("C1") FROM "T1"
   WHERE "C2">'2004-12-31'
   でデータが1件もヒットしない(2005年以降のデータが存在しない)ことを確認します。



2. ALTER TABLE "T1" CHANGE RDAREA
   MULTIDIM ("C2" ((MAX))) AT (('2004-12-31'),(MAX))
   INTO ((RD13,RD14),(RD23,RD24))
   WIHTOUT PURGE

＜分割後＞



注　1.～2.は，手順の項番に対応しています。

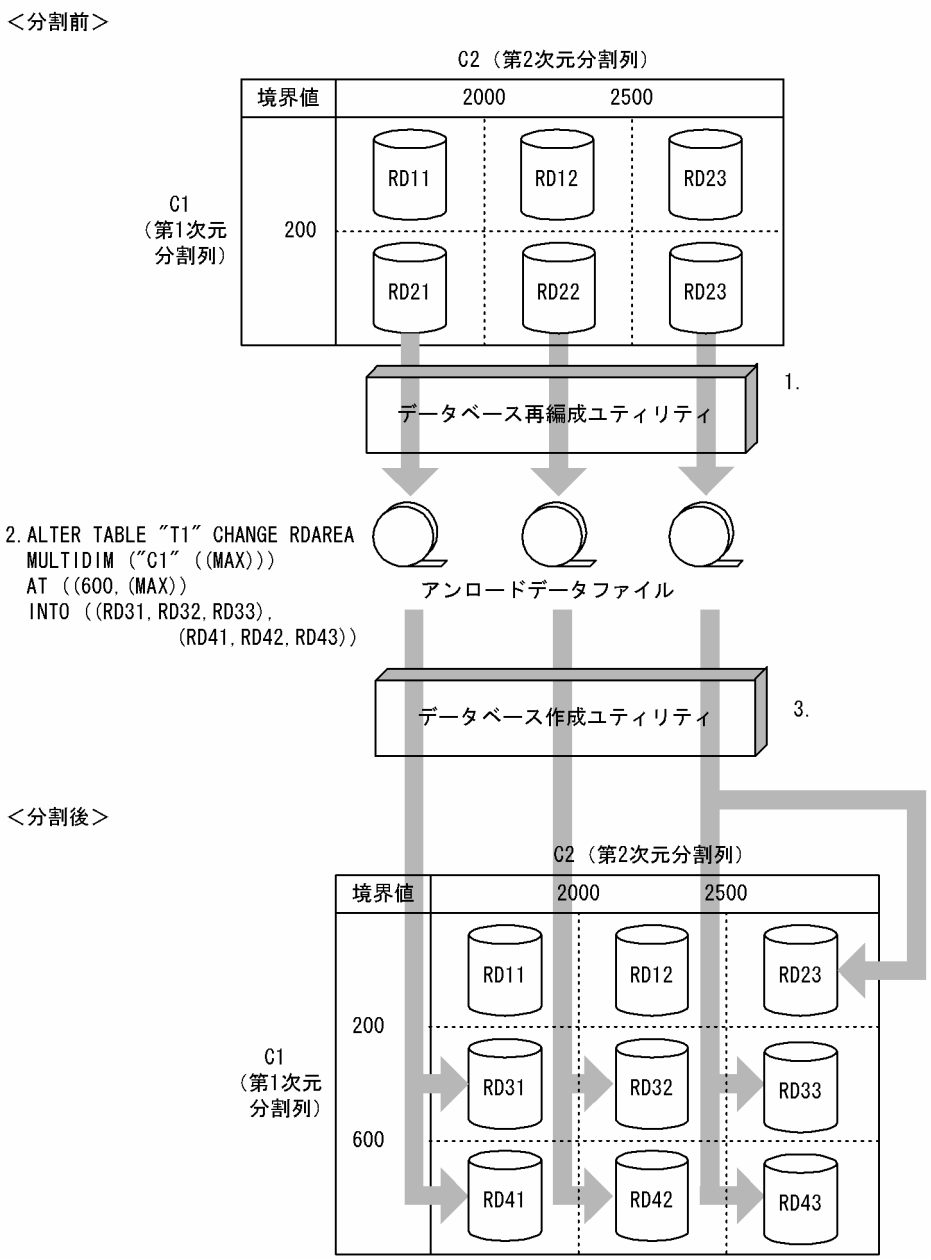
〈手順〉

1. 分割対象の境界値を含む RD エリアが分割後の格納条件のデータしかないことを確認します（SELECT MAX(分割キー列名) FROM 表名 WHERE 分割キー列名>分割後の境界値）。分割後の格納条件以外のデータ（2005 年以降のデータ）がある場合は，「例題 1（分割時の基本的な運用）」で実施してください。

   この手順を実施しなかった結果，分割後の格納条件に合わないデータが残った場合の回復手順については，「分割後の格納条件に合わないデータが残った場合の回復手順」を参照してください。
2. ALTER TABLE で分割したい境界値を分割します。このとき，WITHOUT PURGE を指定します。
3. 2.で無効になった，ルーチン，トリガを ALTER ROUTINE で再作成します。

(3) 例題 3（分割対象の RD エリアがほかの境界値で使用されている場合の運用）

分割対象の RD エリアがほかの境界値でも使用されている場合，ほかの境界値のデータも削除されます。このため，分割対象の RD エリアがほかの境界値でも使用されている場合，分割前にデータベース再編成ユティリティ（pdrorg）でアンロードしたデータをデータベース作成ユティリティ（pdload）でデータロードするときに注意が必要です。この例では，ほかの境界値で使用されている RD エリア（RD23）の第 1 次元の境界値を分割します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で格納条件を変更する RD エリア内の表データ（RD21，RD22，RD23）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。このとき，2000 以下のデータもアンロードされます。
2. ALTER TABLE で分割したい境界値を分割します。このとき，分割対象の RD エリア（RD21，RD22，RD23）のデータは（RD23 のほかの格納範囲のデータも）削除されます。
3. 分割した境界値を格納する RD エリアに対して，1.で作成したアンロードデータファイルを入力として，分割後のすべての RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。このとき，RD23 のデータも 2.の時点でデータが削除されているため，RD23 に対しても追加モードでデータロードする必要があります。

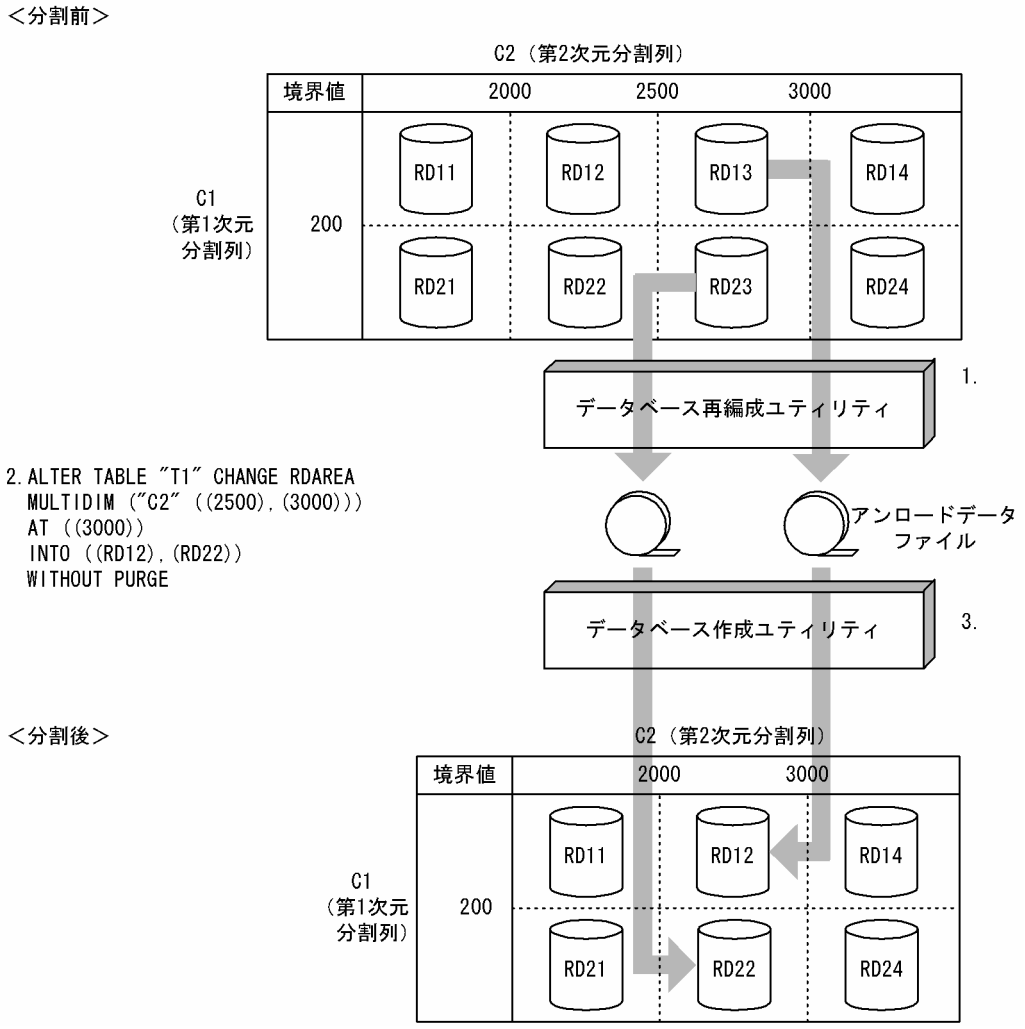
アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。

4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。

5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(4) 例題 4（統合前の一つの特定境界値格納 RD エリア群を統合後の特定境界値格納 RD エリア群として使用する場合の運用）

ある連続した格納範囲の特定境界値格納 RD エリア群に空き容量が多いため，一つの特定境界値格納 RD エリア群にデータの格納範囲を統合する場合の運用について説明します。統合後の RD エリアは，統合前の一つの RD エリアを使用しても使用しなくてもかまいません。この例では，RD12，RD13，RD22，及び RD23 に分割していた第 2 次元の格納条件を，RD12 及び RD22 に統合します。



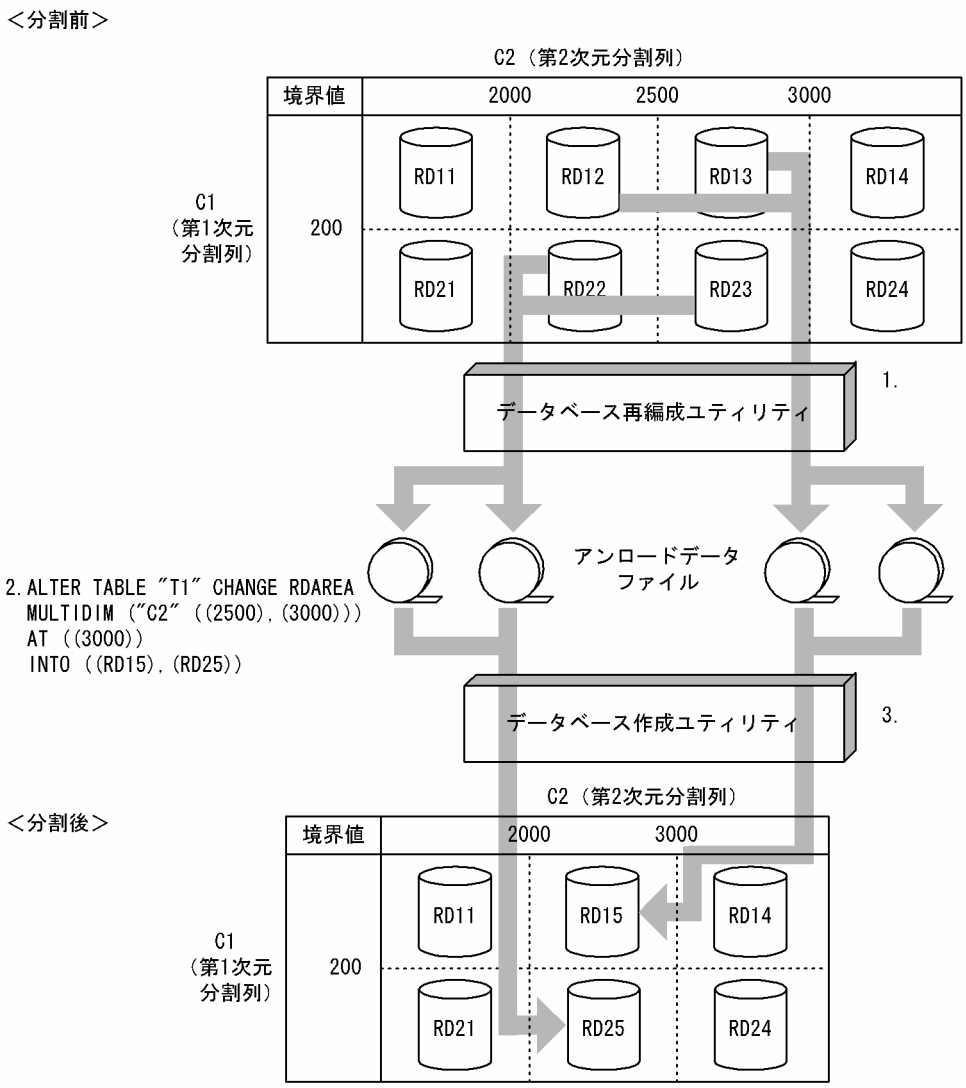
注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で統合対象の RD エリアのうち，統合後に使用しない RD エリア内の表データ（RD13，RD23）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で統合したい境界値を統合します。このとき，RD12 及び RD22 は統合後もそのまま使用するため，WITHOUT PURGE を指定します。
3. 統合した境界値を格納する RD エリアに対して，1.で作成したすべてのアンロードデータファイルを入力として，統合後の RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。このとき，作成モードでデータロードすると，統合前の RD12 及び RD22 のデータが削除されてしまうため，注意が必要です。誤って作成モードでデータロードした場合は，ALTER TABLE 実行前の状態にバックアップから回復して，手順 1.からやり直します。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(5) 例題 5（統合前の一つの特定境界値格納 RD エリア群を統合後の特定境界値格納 RD エリア群として使用しない場合の運用）

この例では，RD12，RD13，RD22，及び RD23 に分割していた第 2 次元の格納条件を，新たに用意した RD15 及び RD25 に統合します。



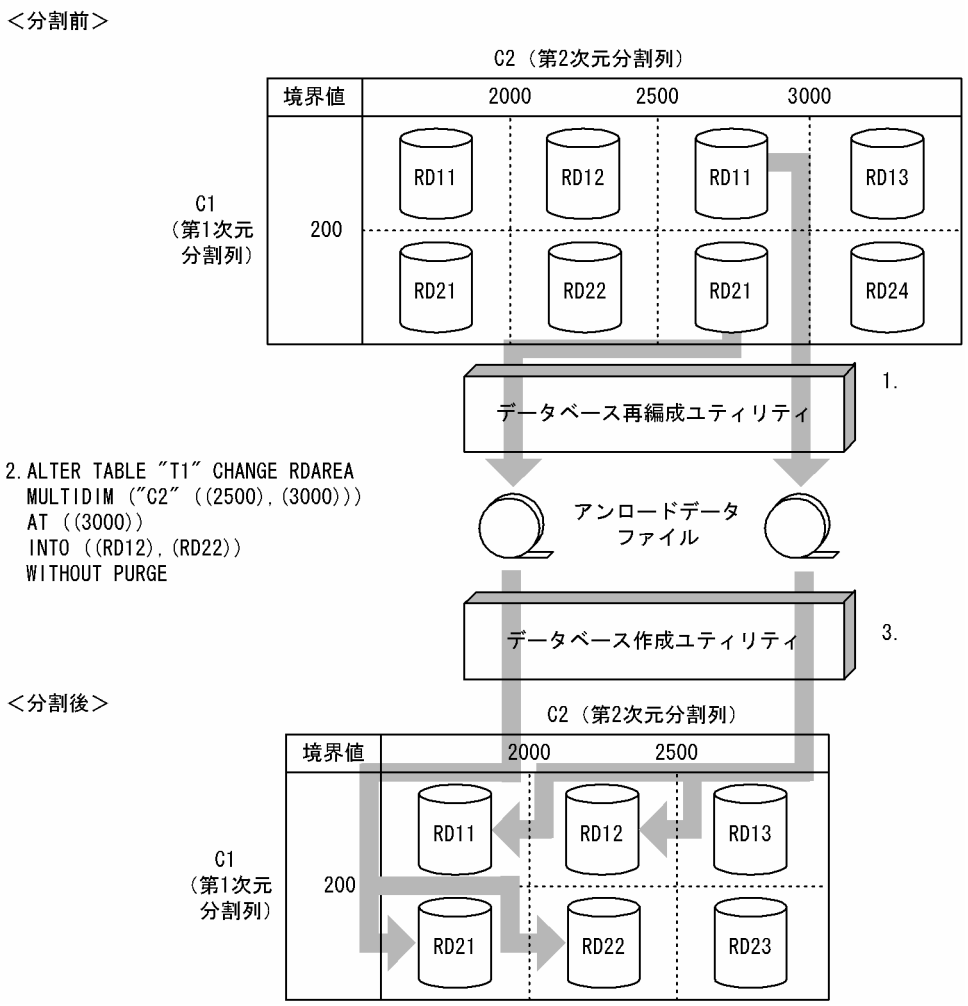
注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で統合対象のすべての RD エリア内の表データ（RD12，RD13，RD22，RD23）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。
2. ALTER TABLE で統合したい境界値を統合します。
3. 統合した境界値を格納する RD エリアに対して，1.で作成したすべてのアンロードデータファイルを入力として，統合後の RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を使用して，追加モードでデータロードします。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

(6) 例題 6（統合対象の RD エリアがほかの境界値で使用されている場合の運用）

統合対象の RD エリアがほかの境界値でも使用されている場合，ほかの境界値のデータも削除されます。このため，統合後にデータが削除される RD エリアに対しては，統合前にデータベース再編成ユティリティ（pdrorg）でアンロードしたデータをデータベース作成ユティリティ（pdload）でデータロードするときに注意が必要です。この例では，ほかの境界値で使用されている RD エリア（RD11 及び RD21）の第 2 次元の境界値を統合します。



注　1.～3.は，手順の項番に対応しています。

〈手順〉

1. データベース再編成ユティリティ（pdrorg）で統合対象のすべての RD エリア内の表データ（RD11，RD12，RD22，RD23）をデータベース作成ユティリティ（pdload）で入力できる形式でアンロードします。このとき，2000 以下のデータもアンロードされます。
2. ALTER TABLE で統合したい境界値を統合します。このとき，統合対象の RD エリアのデータは（RD11 及び RD21 のほかの格納範囲のデータも）削除されます。

3. 統合した境界値を格納する RD エリアに対して，1.で作成したすべてのアンロードデータファイルを入力として，統合後の RD エリアに対して，RD エリア単位にデータベース作成ユティリティ（pdload）を実行します。このとき，RD11 及び RD21 のデータも 2.の時点でデータが削除されているため，RD11 及び RD21 に対しても追加モードでデータロードする必要があります。
4. データロードした後のデータ件数を確認します。詳細は，「分割・統合後のデータ件数の確認」を参照してください。
5. 2.で無効になったルーチン，トリガを ALTER ROUTINE で再作成します。

13.13.4　データ再登録時の運用

「例題（境界値指定の場合）」，「例題（格納条件指定の場合）」，又は「例題（マトリクス分割の場合）」では，分割・登録後の事前にアンロードしたデータのロードは，RD エリア単位にデータベース作成ユティリティ（pdload）を実行することを説明しました。この運用では，複数の RD エリアにデータロードする場合にデータベース作成ユティリティ（pdload）を並行して実行できるため，業務停止による影響を少なくできます。また，データベース作成ユティリティ（pdload）実行時の排他制御がデータロードの対象となる RD エリアだけに限定できるため，そのほかの RD エリアをアクセスする業務には，影響を与えないという長所があります。

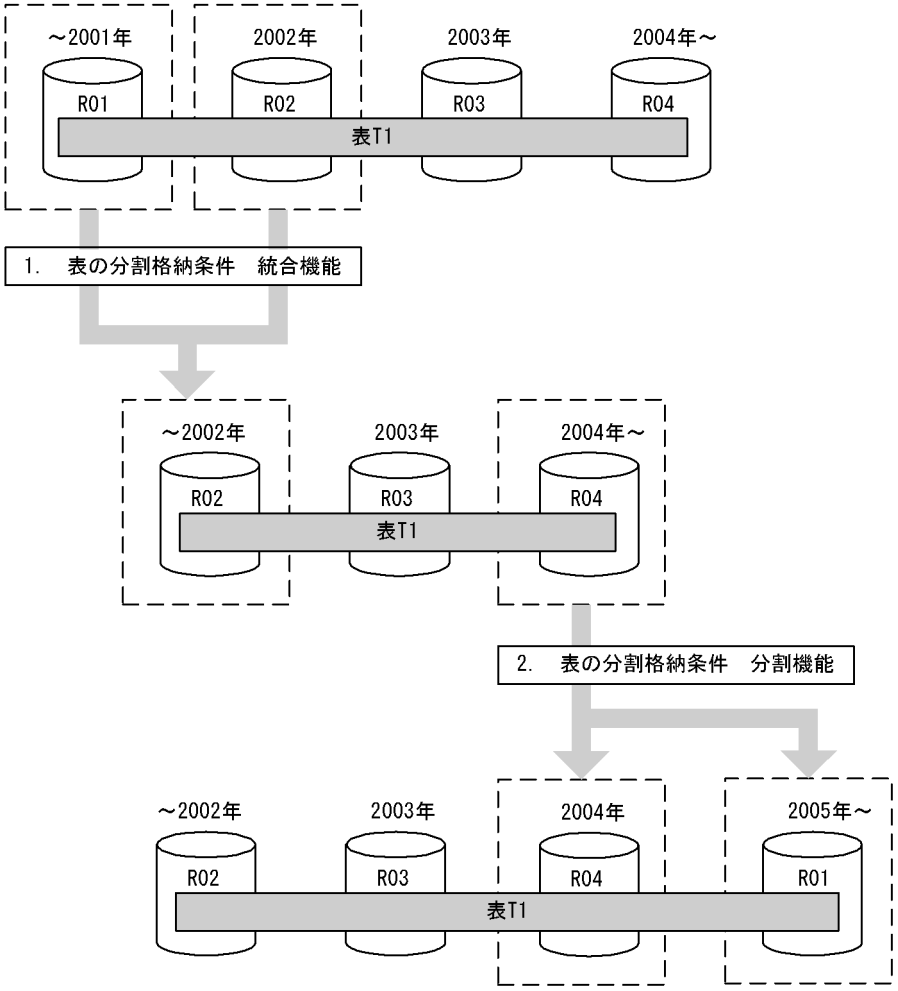
しかし，ユーザがデータを再登録する対象の RD エリアを正しく認識する必要があり，運用ミスを起こしやすいことと，複数の RD エリアにデータを登録する場合，データベース作成ユティリティ（pdload）を複数回実行しなければならないという短所もあります。

このため，業務停止時間を十分に確保でき，その間，分割・統合する表に対するアクセスを中断できる場合は，RD エリア単位ではなく表単位のデータベース作成ユティリティ（pdload）でデータロードすることをお勧めします。

13.13.5　RD エリアを再利用する運用

RD エリアを再利用する運用とは，時系列にデータが増加していくようなデータベースで，過去の古いデータを格納している RD エリアのデータを削除し，新しいデータを格納するための RD エリアとして再利用するような運用をいいます。RD エリアを再利用する運用の例を次の図に示します。

図 13–83　RD エリアを再利用する運用の例



注　1.～2.は，手順の項番に対応しています。

〈手順〉

1. 分割格納条件　統合機能によって，R01，R02 に格納していた 2002 年以前のデータを R02 の RD エリアに統合します。このとき，システムが R01 の中のデータをすべて削除します。
2. 分割格納条件　分割機能によって，2004 年以降のデータで 2004 年のデータを R04 に，2005 年以降のデータを 2001 年以前のデータを格納していた R01 を再利用して格納できます。

RD エリアを再利用するための運用手順を次の図に示します。

図 13–84　RD エリアを再利用するための運用手順

境界値	'2001-12-31'	'2002-12-31'	'2003-12-31'	
表格納用RDエリア	R01	R02	R03	R04
インデクス格納用RDエリア	RI01	RI02	RI03	RI04

1. 統合

```
ALTER TABLE "T1" CHANGE RDAREA
 (('2001-12-31'),('2002-12-31')) INTO "R02"
FOR INDEX "I1" INTO "RI02" WITHOUT PURGE
```

境界値	'2002-12-31'	'2003-12-31'	
表格納用RDエリア	R02	R03	R04
インデクス格納用RDエリア	RI02	RI03	RI04

2. 分割

```
ALTER TABLE "T1" CHANGE RDAREA
 ((MAX)) INTO (("R04")'2004-12-31',("R01"))
FOR INDEX "I1" INTO (("RI04"),("RI01")) WITHOUT PURGE
```

境界値	'2002-12-31'	'2003-12-31'	'2004-12-31'	
表格納用RDエリア	R02	R03	R04	R01
インデクス格納用RDエリア	RI02	RI03	RI04	RI01

注　1.～2.は，手順の項番に対応しています。

〈手順〉

1. R01 を該当する表から切り離すために，ALTER TABLE で最小境界値（'2001-12-31'）を最小境界値より一つ大きい境界値（'2002-12-31'）に統合します。統合後の RD エリアには，最小境界値より一つ大きい境界値を格納している RD エリア（R02，RI02）を指定します。このとき，R02 内のデータは残す必要があるため，WITHOUT PURGE を指定します。この結果，R01 が定義上，該当する表から切り離された状態となり，R01 内の該当する表のデータが削除されます。
2. ALTER TABLE で最大境界値より大きい値を格納する RD エリア（R04）をその RD エリアに格納されている分割キー値の最大値以上の値（'2004-12-31'）で分割します。分割後の RD エリアには，最大境界値より大きい値を格納していた RD エリア（R04，RI04）と最小境界値を格納していた RD エリア（R01，RI01）を指定します。このとき，R04 のデータは残す必要があるため，WITHOUT PURGE を指定します。

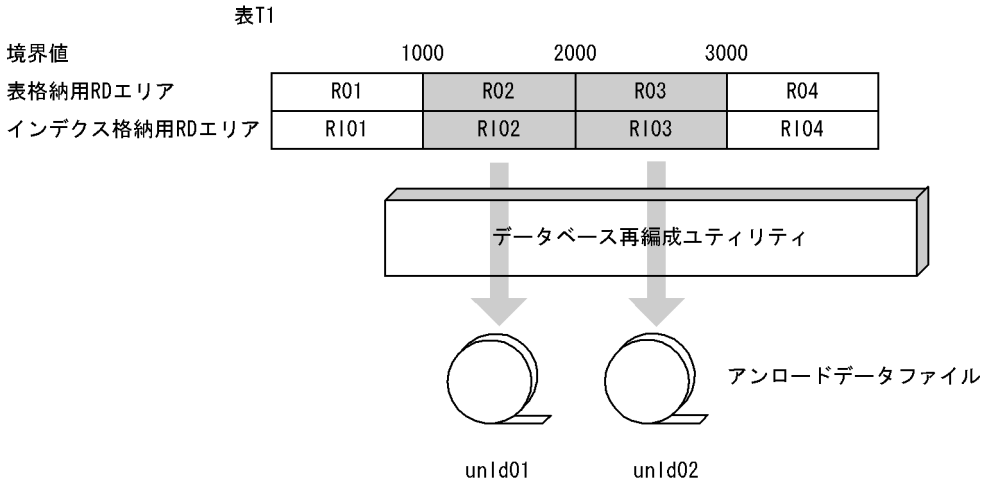
   R04 内に'2005-1-1'以降のデータがあると，格納条件に合わないデータが R04 内に残る可能性があるため，注意が必要です。詳細については，「例題 2（分割前の RD エリアを分割後に使用してデータを削除しないで分割する運用）」を参照してください。

13.13.6　データベース再編成ユティリティ及びデータベース作成ユティリティの例

(1)　データベース再編成ユティリティ（pdrorg）の例

データベース再編成ユティリティ（pdrorg）は，RD エリア単位に，データベース作成ユティリティ（pdload）が入力できる形式でアンロードする必要があります。データベース再編成ユティリティの例を次の図に示します。この例では，表の分割格納条件を変更する前に必要なアンロードデータファイルを作成しています。

図 13–85　データベース再編成ユティリティの例



(a)　R02 をアンロードします

R02 をアンロードする場合の制御文とコマンドラインの例を次に示します。

```
●制御文（control_file1）
unload C:¥pdrorg¥unld01

●コマンドライン
pdrorg -k unld -r R02 -W bin -t T1 control_file1
```

〔説明〕

-k：アンロードをするため unld を指定します。

-r：アンロード対象の RD エリア名を指定します。

-W bin：データベース作成ユティリティが使用する形式でアンロードします。必ず指定します。

-t：アンロードする表の表識別子を指定します。

control_file1：制御文ファイル名を指定します。

(b)　R03 をアンロードします

R03 をアンロードする場合の制御文とコマンドラインの例を次に示します。

```
●制御文（control_file2）
unload C:¥pdrorg¥unld02

●コマンドライン
pdrorg -k unld -r R03 -W bin -t T1 control_file2
```

〔説明〕

-k：アンロードをするため unld を指定します。

-r：アンロード対象の RD エリア名を指定します。

-W bin：データベース作成ユティリティが使用する形式でアンロードします。必ず指定します。

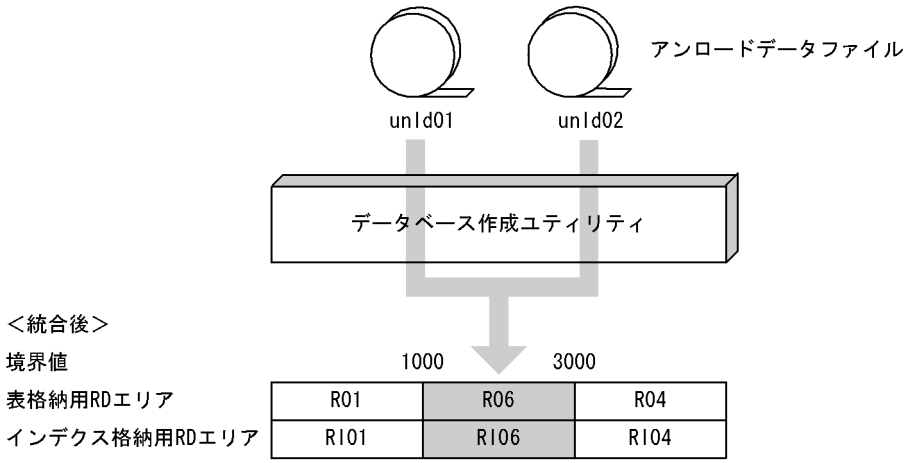
-t：アンロードする表の表識別子を指定します。

control_file2：制御文ファイル名を指定します。

(2)　データベース作成ユティリティ（pdload）の例

データベース作成ユティリティ（pdload）は，データベース再編成ユティリティ（pdrorg）で作成したアンロードデータファイルから RD エリア単位に追加モードで実行する必要があります。データベース作成ユティリティの例を次の図に示します。この例では，表の分割格納条件を変更した表からデータベースを作成しています。

図 13–86　データベース作成ユティリティの例



(a)　R01 と R02 を R06 に統合します

R06 に統合する場合の制御文とコマンドラインの例を次に示します。

```
●制御文（control_file）
option divermsg=off
source R06 C:¥pdrorg¥unld01,C:¥pdrorg¥unld02

●コマンドライン
pdload -W -b T1 control_file
```

〔説明〕

-W：データベース再編成ユティリティで出力したファイルを入力ファイルとします。必ず指定します。

-b：入力ファイルがバイナリ形式です。必ず指定します。

T1：対象の表識別子を指定します。

control_file：制御文ファイル名を指定します。

13.13.7　非分割キーインデクスがある表の分割・統合

分割・統合する対象の表に表格納用 RD エリアと 1 対 1 になっていない非分割キーインデクスが定義してあると，分割・統合できません。

〈手順〉

1. 非分割キーインデクスを削除します。
2. 分割・統合対象の RD エリアの表データをアンロードします。
3. ALTER TABLE で分割したい境界値又は RD エリアを分割します。
4. 分割・統合した RD エリアに 2.で作成したアンロードデータファイルからデータベース作成ユティリティでデータロードします。
5. 非分割キーインデクスを再定義します。

13.13.8　インデクスが不完全な状態での分割・統合

データベース作成ユティリティ実行後，インデクスを再作成していない状態で分割格納条件を変更してはなりません。

もし，上記のインデクスがある状態で分割格納条件を変更してしまった場合は，そのインデクスに対してデータベース再編成ユティリティでインデクスを再作成する必要があります。

13.13.9　分割・統合後のデータ件数の確認

分割格納条件を変更する場合，事前にデータをアンロードしておいて，分割格納条件を変更してからデータをロードする必要があります。このため，変更前のデータ件数とデータ再登録後のデータ件数が一致することを確認してください。確認方法には，次の二つがあります。より正確にデータ件数を確認したい場合は，後者の方法をお勧めします。

- **ユティリティが出力するデータ件数で確認する方法**

  1. データベース再編成ユティリティ実行時に出力される KFPL00723-I メッセージでアンロード件数を控えておきます。

2. 分割格納条件を変更後，データロード時に出力される KFPL00723-I メッセージでデータロードした件数を控えます。複数の RD エリアにデータロードした場合は，その総和を控えます。

3. 1.の件数と 2.の件数が一致していることを確認します。

- **分割格納条件の変更前後で，データベース状態解析ユティリティを実行する方法**

  RD エリア内に格納されているデータ件数は，データベース状態解析ユティリティ（pddbst）の表単位の状態解析を実行すれば確認できます。

  1. データベース状態解析ユティリティ（pddbst）の表単位の状態解析を実行し，次の例の対象 RD エリアの下線（Row Count：RD エリアの格納行数）部分の件数の総和を控えます。

```
pddbst -t 表名 -s
```

  実行結果を次に示します。

```
RD Area Name   : RD002
  Server        : bes1
  Status        :
  Original RD Area Name : RD002
  Generation Number :  0   Replica RD Area Count :  0
  History1 Hold Status :      Hold Code :     0  Hold Time :
  History2 Hold Status :      Hold Code :     0  Hold Time :
  Job Name      :         Line Count :               Index Method :
  Unused Segment:       3999
  Search Mode : INS   Segment Reuse :          - segments
  Reuse Search Failure :          0/         0
          Used(Full)       Used(      Full)        Sum
  Segment 100%(  0%)          1(         0)          1
  Page     30%(  0%)          3(         0)         10
  Row Count     :                  200

RD Area Name   : RD004
  Server        : bes2
  Status        :
  Original RD Area Name : RD004
  Generation Number :  0   Replica RD Area Count :  0
  History1 Hold Status :      Hold Code :     0  Hold Time :
  History2 Hold Status :      Hold Code :     0  Hold Time :
  Job Name      :         Line Count :               Index Method :
  Unused Segment:       3999
  Search Mode : INS   Segment Reuse :          - segments
  Reuse Search Failure :          0/         0
          Used(Full)       Used(      Full)        Sum
  Segment 100%(  0%)          1(         0)          1
  Page     20%(  0%)          2(         0)         10
  Row Count     :                  100
```

  2. 分割格納条件を変更し，データベース作成ユティリティでデータロード後，1.と同じ手順で件数の総和を控えます。

  3. 1.と 2.の件数が一致することを確認します。

データ件数を確認した結果，件数が一致しなかった場合に考えられる要因と対策を次に示します。

考えられる要因	対策
データベース作成ユティリティでデータをロードする場合，追加モードでデータロードする必要がありますが，作成モードでデータロードしていました。	この場合，事前に取得したバックアップから回復し，再実行する必要があります。
データロードする RD エリアに対して，データロードを実行していません。	この場合，アンロードデータファイルからデータロードする RD エリアに対してデータロードします。

13.13.10　障害時の運用

(1)　障害時の戻し運用

ALTER TABLE 実行中に異常終了した場合，表の変更処理はロールバックするため，回復は必要ありません。ALTER TABLE を再度実行します。

しかし，ALTER TABLE 実行後は，RD エリアのデータが削除されたりするために，バックアップを取得していないと，ALTER TABLE 実行前の状態には回復できません。

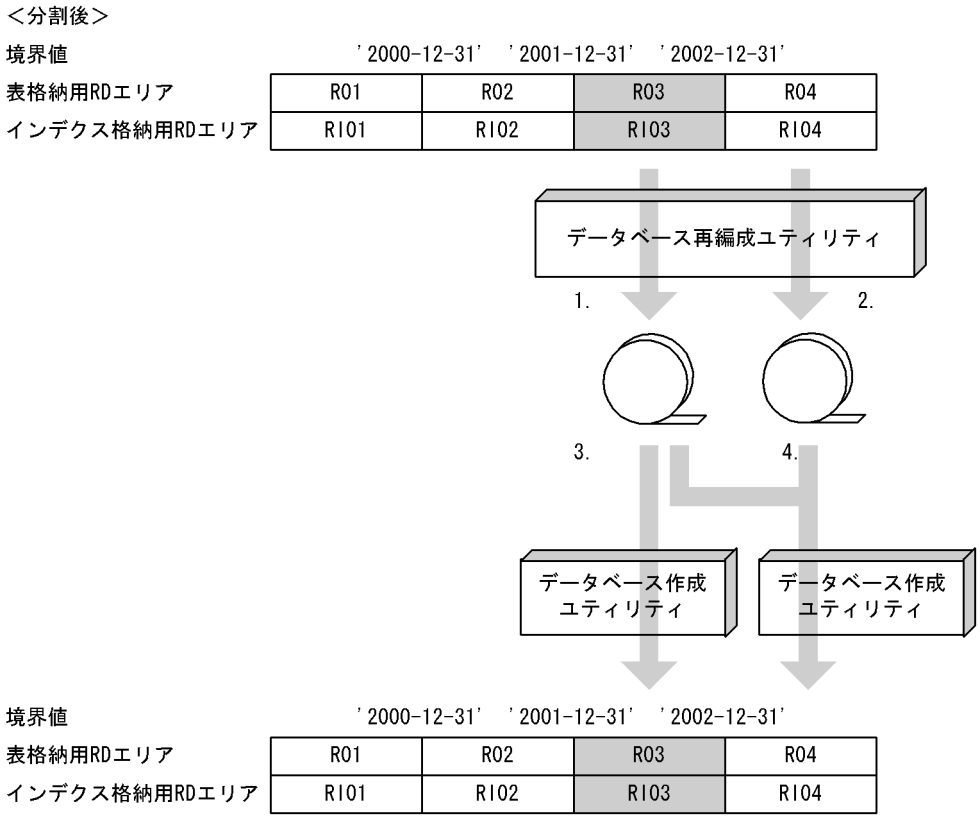
このため，すべての運用パターンについて，事前にバックアップし，回復時はそのバックアップから回復する必要があります。バックアップの取得対象を次に示します。

- マスタディレクトリ
- データディレクトリ
- ディクショナリ用 RD エリア（ディクショナリ LOB 用 RD エリアを含みます）
- 分割条件の変更対象の RD エリア（インデクス用，LOB 用 RD エリアを含みます）

(2)　分割後の格納条件に合わないデータが残った場合の回復手順

不当に WITHOUT PURGE を指定した結果，分割後の格納条件に合わないデータが残った場合，分割キーで検索すると，データが検索できないなどの不具合が発生します。分割後の格納条件に合わないデータが残った場合の回復手順の例を次の図に示します。この例は，本来は R04 に格納するはずの不正なデータが R03 に残っている場合の回復手順です。

図 13–87　分割後の格納条件に合わないデータが残った場合の回復手順の例



注　1.～4.は，手順の項番に対応しています。

〈手順〉

1. 不正なデータが格納されている RD エリアの表データをデータベース作成ユティリティで入力できる形式でアンロードします。
2. 本来格納されるはずの RD エリア内の表データをデータベース作成ユティリティで入力できる形式でアンロードします。
3. 不正なデータが格納されている RD エリアに対して，1.で作成したアンロードデータファイルを入力として，RD エリア単位にデータベース作成ユティリティを作成モードで実行します。

   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。
4. 本来格納されるはずの RD エリアに対して，1.，及び 2.で作成したアンロードデータファイルを入力として，RD エリア単位にデータベース作成ユティリティを作成モードで実行します。

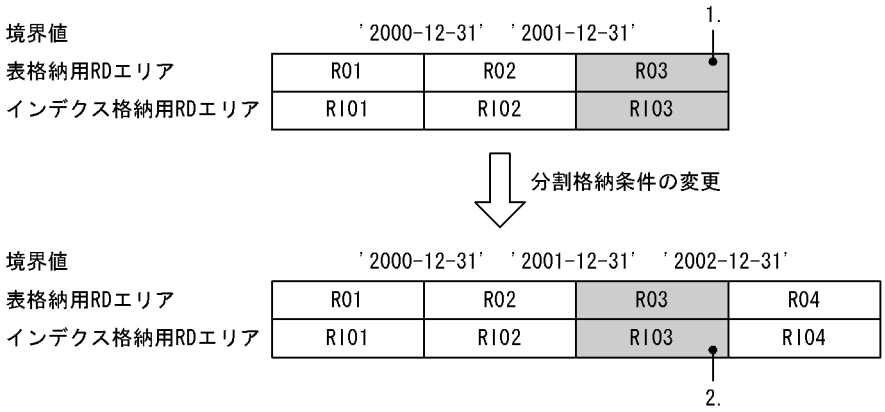
   アンロードデータファイル中には，分割後の格納条件に合わないデータが含まれているため，エラーデータ情報が出力されます。エラーデータ情報が不要な場合は，option 文の divermsg=off を指定して，エラーデータ情報の出力を抑止します。

13.13.11　参照制約及び検査制約を使用している場合の運用

(1)　被参照表に対して分割格納条件を変更する場合

分割格納条件の変更時は，表のデータが削除されることがあります。このため，被参照表の分割格納条件を変更した場合は，被参照表と参照表の整合性が保証されません。

分割時の WITHOUT PURGE が適用できる場合に限り，表のデータは削除されません。しかし，「分割後の格納条件に合わないデータが残った場合の回復手順」で示したとおり，分割格納条件の変更後に格納条件に合わないデータがあった場合，そのデータは検索できなくなり，参照表のデータと不整合が発生します。例を次に示します。



〔説明〕

1. 被参照表の R03 に「2004-12-30」のデータがあり，参照表にも「2004-12-30」のデータがあるとします。この場合，参照制約の整合性はとれています。
2. WITHOUT PURGE を指定した場合，「2001-12-31～2002-12-31」のデータを格納する R03 に「2004-12-30」のデータが残った状態となります。この場合，被参照表の「2004-12-30」のデータは検索できなくなり，参照制約の整合性はとれなくなります。

システム定義の pd_check_pending=USE 指定時に，被参照表に対して分割格納条件の変更をした場合，WITHOUT PURGE の指定に関係なく，参照表はすべて検査保留状態となります。また，pd_check_pending=NOUSE 指定時には，参照表は非検査保留状態となります。そのため，分割格納条件の変更後は，表の整合性を確認し，検査保留状態を解除する必要があります。詳細については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　参照表に対して分割格納条件を変更する場合

参照表に対して分割格納条件を変更する場合の運用について説明します。

- **システム定義で pd_check_pending=USE を指定している場合**

  参照表に対して分割格納条件を変更した場合，表のデータが削除された RD エリアは非検査保留状態となります。これに伴って，該当表を格納している全 RD エリアが非検査保留状態となる場合，該当表の

検査保留状態は解除されます（SQL_TABLES 表の CHECK_PEND 列，及び SQL_REFERENTIAL_CONSTRAINTS 表の CHECK_PEND 列の値がナル値になります）。

- **システム定義で pd_check_pending=NOUSE を指定している場合**

  参照表に対して分割格納条件を変更した場合，表のデータが削除された RD エリアは非検査保留状態となります。

(3) 検査制約を定義した表に対して分割格納条件を変更する場合

検査制約を定義した表に対して分割格納条件を変更する場合の運用について説明します。

- **システム定義で pd_check_pending=USE を指定している場合**

  検査制約を定義した表に対して分割格納条件を変更した場合，表のデータが削除された RD エリアは非検査保留状態となります。これに伴って，該当表を格納している全 RD エリアが非検査保留状態となる場合，該当表の検査保留状態は解除されます（SQL_TABLES 表の CHECK_PEND2 列，及び SQL_CHECKS 表の CHECK_PEND2 列の値がナル値になります）。

- **システム定義で pd_check_pending=NOUSE を指定している場合**

  検査制約を定義した表に対して分割格納条件を変更した場合，表のデータが削除された RD エリアは非検査保留状態となります。

13.14　ハッシュ関数を変更する方法

実行者　HiRDB 管理者及び表の所有者

RD エリアごとの格納行数に偏りがあると，表に対する処理性能が悪くなることがあります。このような場合は，ハッシュ関数を変更して RD エリアごとの格納行数を均等にし，表に対する処理性能を向上してください。

注意事項

1. ハッシュ関数を変更すると，その表を使用しているストアドルーチンが無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。
2. トリガ SQL 文中に指定している表のハッシュ関数を変更すると，トリガが無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
3. ハッシュ関数の変更後，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
4. コンストラクタ逆生成関数を持つプラグインの抽象データ型を定義した表については，ハッシュ関数を変更できます。

13.14.1　例題１（フレキシブルハッシュ分割の場合）

表（TABLE01）のハッシュ関数を HASH6 に変更します。TABLE01 はフレキシブルハッシュ分割しています。

(1)　ALTER TABLE でハッシュ関数を変更します

```
ALTER TABLE TABLE01 CHANGE HASH HASH6;
```

13.14.2　例題２（FIX ハッシュ分割の場合）

表（TABLE01）のハッシュ関数を HASH6 に変更します。TABLE01 は FIX ハッシュ分割しています。

(1)　pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2)　pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld02
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld02：pdrorg コマンドの制御文ファイル名を指定します。

(3) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

(4) PURGE TABLE 文で TABLE01 のデータを削除します

```
PURGE TABLE TABLE01;
```

FIX ハッシュ分割した表のハッシュ関数を変更する場合は，表のデータを削除する必要があります。

(5) ALTER TABLE でハッシュ関数を変更します

```
ALTER TABLE TABLE01 CHANGE HASH HASH6;
```

(6) pdhold コマンドでデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(7) pdload コマンドで TABLE01 にデータをデータロードします

```
pdload -b -W TABLE01 C:¥pdload¥load02
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load02：pdload コマンドの制御文ファイル名を指定します。

(8)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(9)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.15　表の分割定義を変更する方法

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

ここでは，表の分割定義を変更する方法について説明します。

> **参考**
>
> 表の分割方法を変更した後，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.15.1　例題１　キーレンジ分割からハッシュ分割に変更，及び分割キー列を変更する場合

- 表（TABLE01）の横分割方法をキーレンジ分割からフレキシブルハッシュ分割に変更します。
- 表（TABLE01）の分割キー列を変更します。

(1)　pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2)　pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld01：pdrorg コマンドの制御情報ファイル名を指定します。

(3)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

(4)　DROP TABLE で TABLE01 を削除します

```
DROP TABLE TABLE01;
```

(5)　CREATE TABLE で TABLE01 の分割方法を再定義します

```
CREATE TABLE TABLE01 … ;
```

(6)　CREATE INDEX でインデクスを再定義します

```
CREATE INDEX INDX01 … ;
```

(7)　pdhold コマンドでデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(8)　pdload コマンドで TABLE01 にデータロードします

```
pdload -b -W TABLE01 C:¥pdload¥load01
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御情報ファイル名を指定します。

(9)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(10)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.15.2　例題２　ハッシュ分割からキーレンジ分割に変更する場合

分割キーを含む列構成に変更がない場合にかぎり，次に示す方法で，表の分割方法をキーレンジ分割からハッシュ分割に変更できます。

(1)　pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2)　pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld02
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld02：pdrorg コマンドの制御情報ファイル名を指定します。

(3)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

(4)　DROP TABLE で TABLE01 を削除します

```
DROP TABLE TABLE01;
```

(5)　CREATE TABLE で TABLE01 の分割方法を再定義します

```
CREATE TABLE TABLE01 … ;
```

(6)　CREATE INDEX でインデクスを再定義します

```
CREATE INDEX INDX01 … ;
```

(7)　pdhold コマンドでデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(8)　pdload コマンドで TABLE01 にデータをデータロードします

```
pdload -b -W TABLE01 C:¥pdload¥load02
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load02：pdload コマンドの制御文ファイル名を指定します。

(9)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(10)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

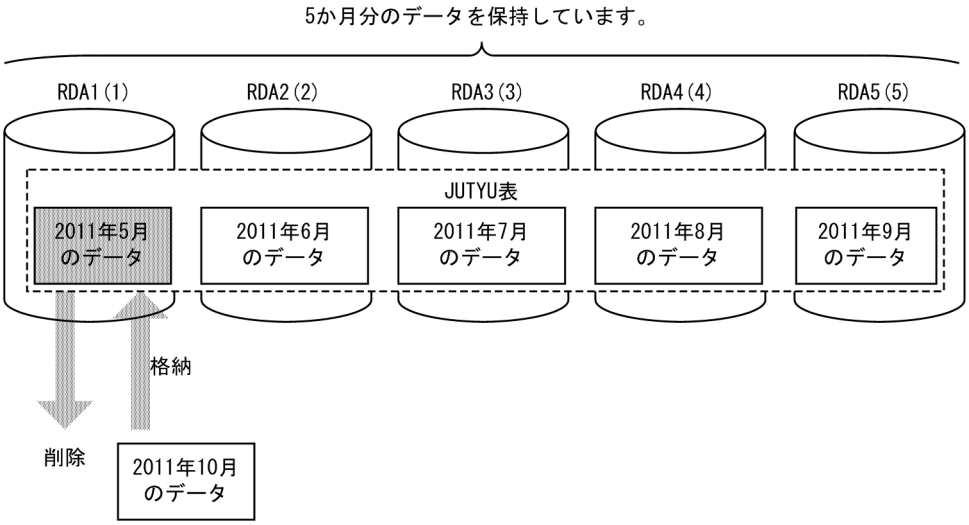
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.15.3　例題３　RD エリアを年月単位に循環させて割り当てる場合（マトリクス分割ではない場合）

ハッシュ関数 HASH0 を使用して，JUTYU 表に 2011 年 5 月～9 月の最新 5 か月のデータを保持している状態で，最も古い 2011 年 5 月の 1 か月分のデータを RD エリア単位に削除し，新しい 2011 年 10 月のデータを格納する例を説明します。

RD エリアを年月単位に循環させて割り当てる方法（マトリクス分割ではない場合）の例を次の図に示します。

図 13–88　RD エリアを年月単位に循環させて割り当てる方法（マトリクス分割ではない場合）



注 1　RD エリア名の(m)は，分割条件指定順序を示しています。

注 2　JUTYU 表は，次のように定義されているものとします。

```
CREATE TABLE JUTYU
  (DNO CHAR(6), TCODE CHAR(5), SCODE CHAR(4), JDATE CHAR(8) NOT NULL)
  FIX HASH HASH0 BY JDATE IN (RDA1,RDA2,RDA3,RDA4,RDA5)
```

(1)　削除対象の 2011 年 5 月のデータが格納されている RD エリア名を特定します

次のどちらかの方法で RD エリア名を特定します。

(a)　表分割ハッシュ関数 p_rdb_dbhash を使用する方法

(i) 表分割ハッシュ関数の実行

HiRDB が提供する表分割ハッシュ関数を使用して，削除対象の 2011 年 5 月のデータが格納されている RD エリア名を特定します。表分割ハッシュ関数の引数を次に示します。なお，表分割ハッシュ関数については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

項番	引数	指定値
1	hashcode（ハッシュ関数コード）	p_rdb_HASH0(100)
2	ncol（分割キーの列数）	1
3	collst（分割キーの指定順序とデータ型コード，データ長コード）	データ型コード：PDSQL_CHAR データ長コード：8
4	dadlst（分割キーに格納するデータ）	項番 3 のデータ型の値
5	ndiv（表の分割数）	5

項番	引数	指定値
6	ncspace（HiRDB サーバで使用する各国文字の全角空白文字）	ncspace[0]：0x81 ncspace[1]：0x40
7	flags（空白変換レベル，DECIMAL 型の符号正規化機能の使用有無）	0
8	rdno（分割条件指定順序，又は分割キー内通番）	なし

注

文字コード種別，空白変換レベル，及び DECIMAL 型の符号正規化機能は，すべて省略しているものとします。

上記の表分割ハッシュ関数を実行すると，分割条件指定順序として **rdno=1** が返ってきます。

(ii) SQL の実行

表分割ハッシュ関数の結果を基に，次の SQL を実行し，RD エリア名を特定します。

```
SELECT RDAREA_NAME
    FROM MASTER.SQL_DIV_TABLE
      WHERE TABLE_SCHEMA='USER1'  /* ユーザ名 */
        AND TABLE_NAME='JUTYU'    /* ハッシュ分割表名 */
        AND DIV_NO=1              /* 分割条件指定順序 */
```

上記の SQL を実行すると，次に示すように削除対象の RD エリア名として **RDA1** が返ってきます。

```
 RDAREA_NAME
 ------------------------------
 RDA1
```

(b)　CREATE PUBLIC FUNCTION を定義しておく方法

(i) CREATE PUBLIC FUNCTION の定義

ハッシュ関数 HASH0 の場合，次の SQL をあらかじめ定義しておくことで RD エリアを特定できます。

```
CREATE PUBLIC FUNCTION HASH0
 (DIVKEY VARCHAR(8)※,          /*  'YYYYMMDD'  又は  'YYYYMM'  */
  NDIV INT)                    /* 分割列数 */
  RETURNS INT
    BEGIN
      DECLARE DIVDATE DATE;
      SET DIVDATE = CASE LENGTH(DIVKEY)
             WHEN 6 THEN DATE(SUBSTR(DIVKEY,1,6)||'01','YYYYMMDD')
             ELSE DATE(DIVKEY,'YYYYMMDD')
             END;
  RETURN MOD(YEAR(DIVDATE)*12+MONTH(DIVDATE), NDIV)+1;    /* 戻り値 */
  END
END_FUNC;
```

注※

列のデータは CHAR(8)ですが，パラメタは VARCHAR でなければ文字列定数を指定した呼び出しができません。

(ii) SQL の実行

次の SQL で，2011 年 5 月のデータがどこに入っているのかを検索します。

```
SELECT RDAREA_NAME
  FROM MASTER.SQL_DIV_TABLE
    WHERE TABLE_SCHEMA='USER1'              /* ユーザ名 */
      AND TABLE_NAME='JUTYU'                /* ハッシュ分割表名 */
      AND DIV_NO=HASH0('20110501',5);       /* RDエリア定義順序 */
```

上記の SQL を実行すると，次に示すように削除対象の RD エリア名として **RDA1** が返ってきます。

```
RDAREA_NAME
------------------------------
RDA1
```

(2) pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDA1
```

(3) pdrorg コマンドで JUTYU 表の RDA1 のデータをアンロードします

```
pdrorg -k unld -t JUTYU -r RDA1 unload_control_file
```

〔説明〕

-k：アンロードをするため unld を指定します。

-t：アンロードする表の名称を指定します。

-r：特定の RD エリアだけアンロードするため，対象となる RD エリア名を指定します。

unload_control_file：pdrorg コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

```
unload bes1:C:¥pdrorg¥unload_file

/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file：アンロードデータファイルの名称 */
```

(4) pdload コマンドで RDA1 に対して 0 件のデータロードをします

最も古い 2011 年 5 月のデータ（RDA1 のデータ）を削除するため，0 件のデータロードを実行します。

```
pdload -d JUTYU load_control_file
```

〔説明〕

-d：既存のデータを削除してからデータロードをするために指定します。

JUTYU：データロードする表の名称を指定します。

load_control_file：pdload コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

```
source RDA1 bes1:C:¥pdload¥load_file

/* RDA1：0件のデータをデータロードするRDエリアの名称 */
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file：ロードする0件データファイルの名称 */
```

分割表に，分割していない非分割インデクスを定義している場合，pdload コマンド実行後に非分割インデクスを一括作成してください。

(5) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDA1
```

コマンドの実行後，UAP などで 2011 年 10 月のデータを挿入すると，RDA1 にデータが格納されます。

なお，コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

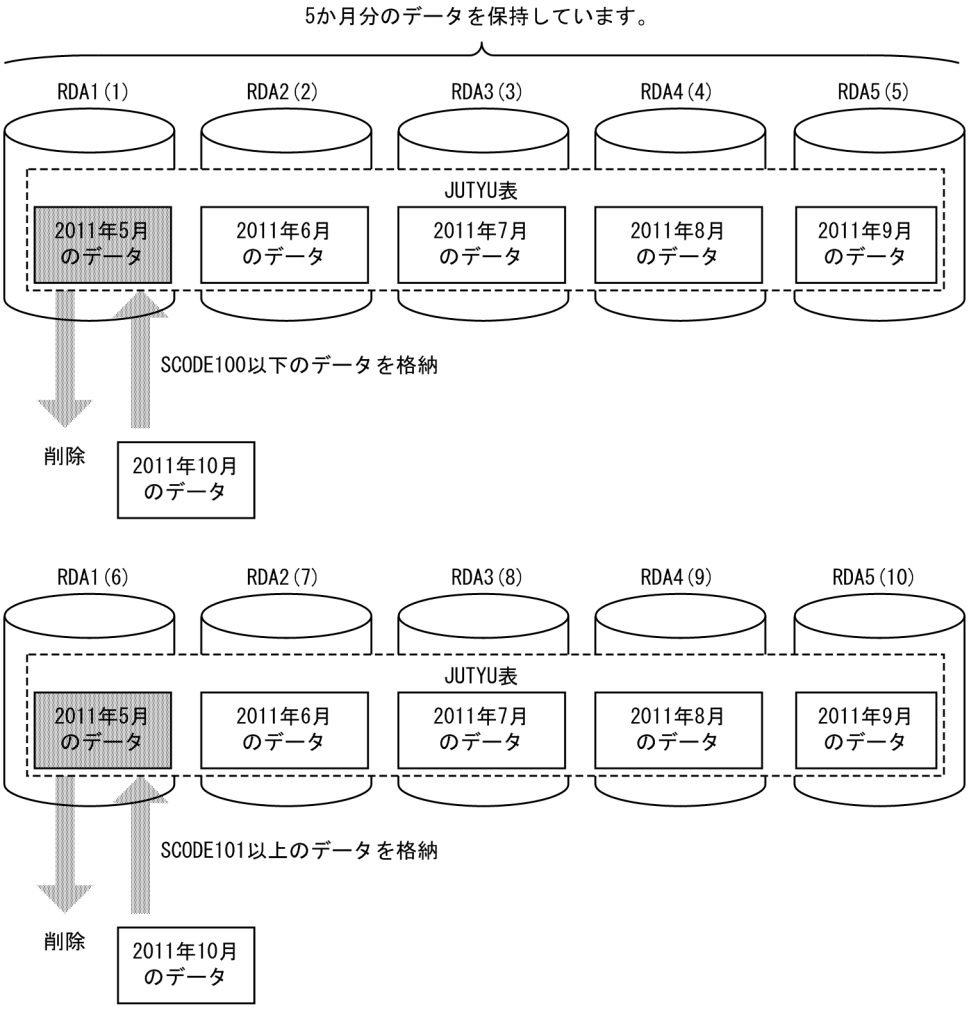
13.15.4 例題４ RD エリアを年月単位に循環させて割り当てる場合（マトリクス分割の場合）

ハッシュ関数 HASH0 を使用したマトリクス分割表を定義して，JUTYU 表に 2011 年 5 月～9 月の最新 5 か月のデータを保持している状態で，最も古い 2011 年 5 月の 1 か月分のデータを RD エリア単位に削除し，新しい 2011 年 10 月のデータを格納する例を説明します。なお，ここでのマトリクス分割表は，次のように分割しているものとします。

- 第 1 次元は，SCODE 列でキーレンジ分割する。第 1 次元の分割数は 2 とし，境界値として 100 を指定する。
- 第 2 次元は，JDATE 列でハッシュ分割する。第 2 次元の分割数は 5 とし，ハッシュ関数として HASH0 を指定する。
- JDATE 列のデータ型は CHAR(8)で，年月日を‘YYYYMMDD’の形式で格納する。

RD エリアを年月単位に循環させて割り当てる場合（マトリクス分割の場合）の例を次の図に示します。

図 13–89　RD エリアを年月単位に循環させて割り当てる場合（マトリクス分割の場合）



注 1　RD エリア名の(m)は，分割条件指定順序を示しています。

注 2　JUTYU 表は，次のように定義されているものとします。

```
CREATE TABLE JUTYU
  (DNO CHAR(6), TCODE CHAR(5), SCODE CHAR(4) NOT NULL, JDATE CHAR(8) NOT NULL)
    PARTITIONED BY MULTIDIM
    (SCODE(('100')),
    FIX HASH HASHZ BY JDATE) IN
    ((RDA1,RDA2,RDA3,RDA4,RDA5), (RDA6,RDA7,RDA8,RDA9,RDA10));
```

(1)　削除対象の 2011 年 5 月のデータが格納されている RD エリア名を特定します

次のどちらかの方法で RD エリア名を特定します。

(a)　表分割ハッシュ関数 p_rdb_dbhash を使用する方法

(i) 表分割ハッシュ関数の実行

HiRDB が提供する表分割ハッシュ関数を使用して，削除対象の 2011 年 5 月のデータが格納されている RD エリア名を特定します。表分割ハッシュ関数の引数を次に示します。なお，表分割ハッシュ関数については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

項番	引数	指定値
1	hashcode（ハッシュ関数コード）	p_rdb_HASH0(100)
2	ncol（分割キーの列数）	1
3	collst（分割キーの指定順序とデータ型コード，データ長コード）	データ型コード：PDSQL_CHAR データ長コード：8
4	dadlst（分割キーに格納するデータ）	項番 3 のデータ型の値
5	ndiv（表の分割数）	5
6	ncspace（HiRDB サーバで使用する各国文字の全角空白文字）	ncspace[0]：0x81 ncspace[1]：0x40
7	flags（空白変換レベル，DECIMAL 型の符号正規化機能の使用有無）	0
8	rdno（分割条件指定順序，又は分割キー内通番）	なし

注

文字コード種別，空白変換レベル，及び DECIMAL 型の符号正規化機能は，すべて省略しているものとします。

上記の表分割ハッシュ関数を実行すると，分割条件指定順序として **rdno=1** が返ってきます。

(ii) SQL の実行

表分割ハッシュ関数の結果を基に，次の SQL を実行し，RD エリア名を特定します。

```
SELECT RDAREA_NAME,DCVALUES
    FROM MASTER.SQL_DIV_TABLE
      WHERE TABLE_SCHEMA='USER1'      /* ユーザ名 */
        AND TABLE_NAME='JUTYU'        /* ハッシュ分割表名 */
          AND MOD(DIV_NO-1,5)+1 = 1   /* RDエリア定義順序 */
```

上記の SQL を実行すると，次のように削除対象の RD エリア名として **RDA1 及び RDA6** が返ってきます。

```
RDAREA_NAME                DCVALUES※
-------------------------  ----------------------
RDA1                       100
RDA6                       **********************
```

注※

DCVALUES 列も同時に検索することによって，RD エリアが第 1 次元のどの分割に対応するかの情報も得られます。

(b)　CREATE PUBLIC FUNCTION を定義しておく方法

(i) CREATE PUBLIC FUNCTION の定義

ハッシュ関数 HASH0 の場合，次の SQL をあらかじめ定義しておくことで RD エリアを特定できます。

```
CREATE PUBLIC FUNCTION HASH0
 (DIVKEY VARCHAR(8)※,          /*  'YYYYMMDD'  又は  'YYYYMM'  */
  NDIV INT)                    /* 分割列数 */
  RETURNS INT
    BEGIN
      DECLARE DIVDATE DATE;
      SET DIVDATE = CASE LENGTH(DIVKEY)
             WHEN 6 THEN DATE(SUBSTR(DIVKEY,1,6)||'01','YYYYMMDD')
             ELSE DATE(DIVKEY,'YYYYMMDD')
             END;
  RETURN MOD(YEAR(DIVDATE)*12+MONTH(DIVDATE), NDIV)+1;    /* 戻り値 */
  END
END_FUNC;
```

注※

列のデータは CHAR(8)ですが，パラメタは VARCHAR でなければ文字列定数を指定した呼び出しができません。

(ii) SQL の実行

次の SQL で，2011 年 5 月のデータがどこに入っているのかを検索します。

```
SELECT RDAREA_NAME,DCVALUES
  FROM MASTER.SQL_DIV_TABLE
    WHERE TABLE_SCHEMA='USER1'                    /* ユーザ名 */
      AND TABLE_NAME='JUTYU'                      /* ハッシュ分割表名 */
      AND MOD(DIV_NO-1,5)+1=HASH0('20110531',5)  /* RDエリア定義順序 */
```

上記の SQL を実行すると，次のように削除対象の RD エリア名として **RDA1 及び RDA6** が返ってきます。

```
 RDAREA_NAME                DCVALUES※
 -------------------------  ----------------------
 RDA1                       100
 RDA6                       ********************
```

注※

DCVALUES 列も同時に検索することによって，RD エリアが第 1 次元のどの分割に対応するかの情報も得られます。

(2)　pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDA1
pdhold -r RDA6
```

(3) pdrorg コマンドで JUTYU 表の RDA1 及び RDA6 のデータをアンロードします

```
pdrorg -k unld -t JUTYU -r RDA1 unload_control_file1
```

```
pdrorg -k unld -t JUTYU -r RDA6 unload_control_file6
```

〔説明〕

-k：アンロードをするため unld を指定します。

-t：アンロードする表の名称を指定します。

-r：特定の RD エリアだけアンロードするため，対象となる RD エリア名を指定します。

unload_control_file：pdrorg コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

＜ RDA1 の場合＞

```
unload bes1:C:¥pdrorg¥unload_file1

/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file1：アンロードデータファイルの名称 */
```

＜ RDA6 の場合＞

```
unload bes1:C:¥pdrorg¥unload_file6

/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file6：アンロードデータファイルの名称 */
```

(4) pdload コマンドで RDA1 及び RDA6 に対して 0 件のデータロードをします

最も古い 2011 年 5 月のデータ（RDA1 及び RDA6 のデータ）を削除するため，0 件のデータロードを実行します。

```
pdload -d JUTYU load_control_file1
```

```
pdload -d JUTYU load_control_file6
```

〔説明〕

-d：既存のデータを削除してからデータロードをするために指定します。

JUTYU：データロードする表の名称を指定します。

load_control_file：pdload コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

＜RDA1 の場合＞

```
source RDA1 bes1:C:¥pdload¥load_file1

/* RDA1：0件のデータをデータロードするRDエリアの名称 */
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file1：ロードする0件データファイルの名称 */
```

＜RDA6 の場合＞

```
source RDA6 bes1:C:¥pdload¥load_file6

/* RDA6：0件のデータをデータロードするRDエリアの名称 */
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file6：ロードする0件データファイルの名称 */
```

分割表に，分割していない非分割インデクスを定義している場合，pdload コマンド実行後に非分割インデクスを一括作成してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDA1
pdrels -r RDA6
```

コマンドの実行後，UAP などで 2011 年 10 月のデータを挿入すると，RDA1 及び RDA6 にデータが格納されます。

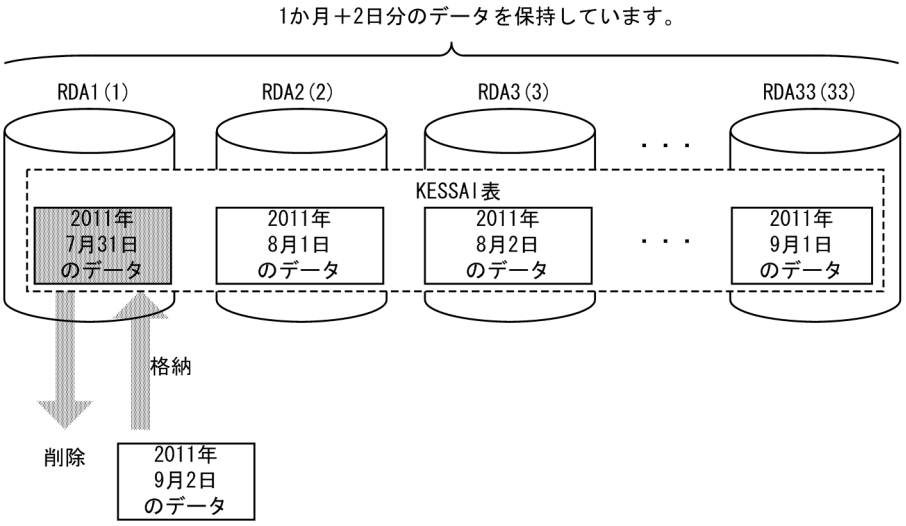
なお，コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.15.5　例題５　RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割でない場合）

ハッシュ関数 HASHZ を使用して，KESSAI 表に 2011 年 7 月 31 日～9 月 1 日の 1 か月＋ 2 日分のデータを保持している状態で，最も古い 2011 年 7 月 31 日のデータを RD エリア単位に削除し，新しい 2011 年 9 月 2 日のデータを格納する例を説明します。

RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割でない場合）の例を次の図に示します。

図 13–90　RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割でない場合）



注 1　RD エリア名の(m)は，分割条件指定順序を示しています。

注 2　KESSAI 表は，次のように定義されているものとします。

```
CREATE TABLE KESSAI
  (DNO CHAR(6), TCODE CHAR(5), SCODE CHAR(4), JDATE CHAR(8) NOT NULL)
  FIX HASH HASHZ BY JDATE IN
  (RDA1,RDA2,RDA3,RDA4,RDA5,RDA6,RDA7,RDA8,RDA9,RDA10
  ,RDA11,RDA12,RDA13,RDA14,RDA15,RDA16,RDA17,RDA18,RDA19,RDA20
  ,RDA21,RDA22,RDA23,RDA24,RDA25,RDA26,RDA27,RDA28,RDA29,RDA30
  ,RDA31,RDA32,RDA33);
```

(1)　削除対象の 2011 年 7 月 31 日のデータが格納されている RD エリア名を特定します

次のどちらかの方法で RD エリア名を特定します。

(a)　表分割ハッシュ関数 p_rdb_dbhash を使用する方法

(i) 表分割ハッシュ関数の実行

HiRDB が提供する表分割ハッシュ関数を使用して，削除対象の 2011 年 7 月 31 日のデータが格納されている RD エリア名を特定します。表分割ハッシュ関数の引数を次に示します。なお，表分割ハッシュ関数については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

項番	引数	指定値
1	hashcode（ハッシュ関数コード）	p_rdb_HASHZ(99)
2	ncol（分割キーの列数）	1
3	collst（分割キーの指定順序とデータ型コード，データ長コード）	データ型コード：PDSQL_CHAR データ長コード：8

項番	引数	指定値
4	dadlst（分割キーに格納するデータ）	項番 3 のデータ型の値
5	ndiv（表の分割数）	33
6	ncspace（HiRDB サーバで使用する各国文字の全角空白文字）	ncspace[0]：0x81 ncspace[1]：0x40
7	flags（空白変換レベル，DECIMAL 型の符号正規化機能の使用有無）	0
8	rdno（分割条件指定順序，又は分割キー内通番）	なし

注

文字コード種別，空白変換レベル，及び DECIMAL 型の符号正規化機能は，すべて省略しているものとします。

上記の表分割ハッシュ関数を実行すると，分割条件指定順序として **rdno=1** が返ってきます。

(ii) SQL の実行

表分割ハッシュ関数の結果を基に，次の SQL を実行し，RD エリア名を特定します。

```
SELECT RDAREA_NAME
    FROM MASTER.SQL_DIV_TABLE
      WHERE TABLE_SCHEMA='USER1'    /* ユーザ名 */
        AND TABLE_NAME='KESSAI'     /* ハッシュ分割表名 */
        AND DIV_NO=1                /* 分割条件指定順序 */
```

上記の SQL を実行すると，次に示すように削除対象の RD エリア名として **RDA1** が返ってきます。

```
 RDAREA_NAME
 ------------------------------
 RDA1
```

(b)　CREATE PUBLIC FUNCTION を定義しておく方法

(i) CREATE PUBLIC FUNCTION の定義

ハッシュ関数 HASHZ の場合，次の SQL をあらかじめ定義しておくことで RD エリアを特定できます。

```
CREATE PUBLIC FUNCTION HASHZ
 (IDATE VARCHAR(8)※,        /*  'YYYYMMDD'  */
  NDIV INT)                 /* 分割列数 */
RETURNS INT
  BEGIN
    RETURN MOD(DAYS(DATE(IDATE,'yyyymmdd'),NDIV)+1;  /* HASHZによる戻り値 */
  END
END_FUNC;
```

注※

列のデータは CHAR(8)ですが，パラメタは VARCHAR でなければ文字列定数を指定した呼び出しができません。

(ii) SQL の実行

次の SQL で，2011 年 7 月 31 日のデータがどこに入っているのかを検索します。

```
SELECT RDAREA_NAME
  FROM MASTER.SQL_DIV_TABLE
    WHERE TABLE_SCHEMA='USER1'                /* ユーザ名 */
      AND TABLE_NAME='KESSAI'                 /* ハッシュ分割表名 */
      AND DIV_NO=HASHZ('20110731',33);        /* RDエリア定義順序 */
```

上記の SQL を実行すると，次に示すように削除対象の RD エリア名として **RDA1** が返ってきます。

```
RDAREA_NAME
------------------------------
RDA1
```

(2) pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDA1
```

(3) pdrorg コマンドで KESSAI 表の RDA1 のデータをアンロードします

```
pdrorg -k unld -t KESSAI -r RDA1 unload_control_file
```

〔説明〕

-k：アンロードをするため unld を指定します。

-t：アンロードする表の名称を指定します。

-r：特定の RD エリアだけアンロードするため，対象となる RD エリア名を指定します。

unload_control_file：pdrorg コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

```
unload bes1:C:¥pdrorg¥unload_file

/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file：アンロードデータファイルの名称 */
```

(4) pdload コマンドで RDA1 に対して 0 件のデータロードをします

最も古い 2011 年 7 月 31 日のデータ（RDA1 のデータ）を削除するため，0 件のデータロードを実行します。

```
pdload -d KESSAI load_control_file
```

〔説明〕

-d：既存のデータを削除してからデータロードをするために指定します。

KESSAI：データロードする表の名称を指定します。

load_control_file：pdload コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

```
source RDA1 bes1:C:¥pdload¥load_file

/* RDA1：0件のデータをデータロードするRDエリアの名称 */
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file：ロードする0件データファイルの名称 */
```

分割表に，分割していない非分割インデクスを定義している場合，pdload コマンド実行後に非分割インデクスを一括作成してください。

(5) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDA1
```

コマンドの実行後，UAP などで 2011 年 9 月 2 日のデータを挿入すると，RDA1 にデータが格納されます。

なお，コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

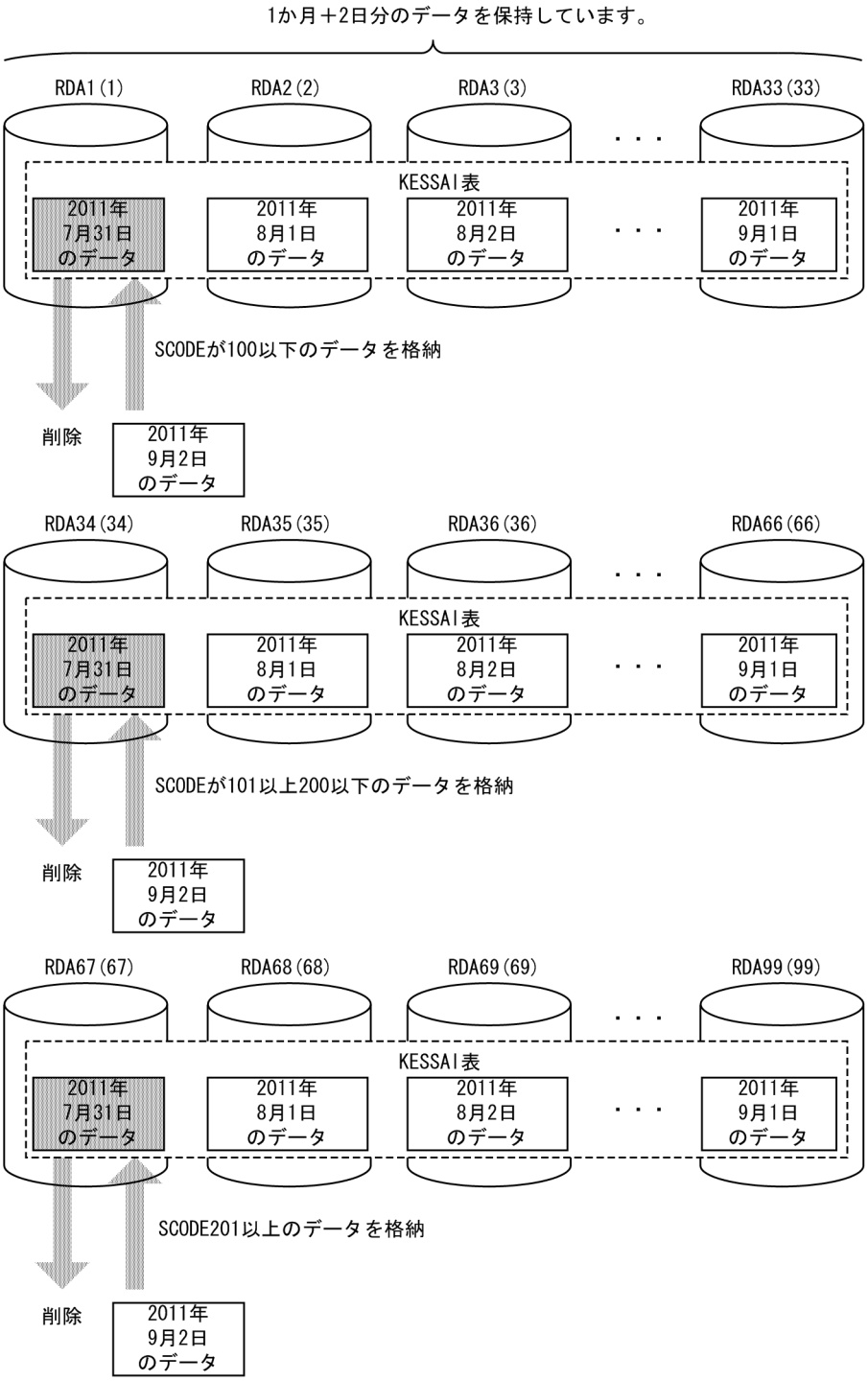
13.15.6 例題６ RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割の場合）

ハッシュ関数 HASHZ を使用したマトリクス分割表を定義して，KESSAI 表に 2011 年 7 月 31 日～9 月 1 日の 1 か月＋ 2 日分のデータを保持している状態で，最も古い 2011 年 7 月 31 日のデータを RD エリア単位に削除し，新しい 2011 年 9 月 2 日のデータを格納する例を説明します。なお，ここでのマトリクス分割表は，次のように分割しているものとします。

- 第 1 次元は，SCODE 列でキーレンジ分割する。第 1 次元の分割数は 3 とし，境界値として 100 及び 200 を指定する。
- 第 2 次元は，JDATE 列でハッシュ分割する。第 2 次元の分割数は 33 とし，ハッシュ関数として HASHZ を指定する。
- JDATE 列のデータ型は CHAR(8)で，年月日を‘YYYYMMDD’の形式で格納する。

RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割の場合）の例を次の図に示します。

図 13–91　RD エリアを年月日単位に循環させて割り当てる場合（マトリクス分割の場合）



注 1　RD エリア名の(m)は，分割条件指定順序を示しています。

注 2　KESSAI 表は，次のように定義されているものとします。

```
CREATE TABLE KESSAI
  (DNO CHAR(6), TCODE CHAR(5), SCODE CHAR(4) NOT NULL, JDATE CHAR(8) NOT NULL)
  PARTITIONED BY MULTIDIM
  (SCODE(('100'),('200')),
  FIX HASH HASHZ BY JDATE) IN
```

```
((RDA1,RDA2,RDA3,RDA4,RDA5,RDA6,RDA7,RDA8,RDA9,RDA10
 ,RDA11,RDA12,RDA13,RDA14,RDA15,RDA16,RDA17,RDA18,RDA19,RDA20
 ,RDA21,RDA22,RDA23,RDA24,RDA25,RDA26,RDA27,RDA28,RDA29,RDA30
 ,RDA31,RDA32,RDA33),(RDA34,RDA35,RDA36,RDA37,RDA38,RDA39,RDA40
 ,RDA41,RDA42,RDA43,RDA44,RDA45,RDA46,RDA47,RDA48,RDA49,RDA50
 ,RDA51,RDA52,RDA53,RDA54,RDA55,RDA56,RDA57,RDA58,RDA59,RDA60
 ,RDA61,RDA62,RDA63,RDA64,RDA65,RDA66),(RDA67,RDA68,RDA69,RDA70
 ,RDA71,RDA72,RDA73,RDA74,RDA75,RDA76,RDA77,RDA78,RDA79,RDA80
 ,RDA81,RDA82,RDA83,RDA84,RDA85,RDA86,RDA87,RDA88,RDA89,RDA90
 ,RDA91,RDA92,RDA93,RDA94,RDA95,RDA96,RDA97,RDA98,RDA99));
```

(1) 削除対象の 2011 年 7 月 31 日のデータが格納されている RD エリア名を特定します

次のどちらかの方法で RD エリア名を特定します。

(a) 表分割ハッシュ関数 p_rdb_dbhash を使用する方法

(i) 表分割ハッシュ関数の実行

HiRDB が提供する表分割ハッシュ関数を使用して，削除対象の 2011 年 7 月 31 日のデータが格納されている RD エリア名を特定します。表分割ハッシュ関数の引数を次に示します。なお，表分割ハッシュ関数については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

項番	引数	指定値
1	hashcode（ハッシュ関数コード）	p_rdb_HASHZ(99)
2	ncol（分割キーの列数）	1
3	collst（分割キーの指定順序とデータ型コード，データ長コード）	データ型コード：PDSQL_CHAR データ長コード：8
4	dadlst（分割キーに格納するデータ）	項番 3 のデータ型の値
5	ndiv（表の分割数）	33
6	ncspace（HiRDB サーバで使用する各国文字の全角空白文字）	ncspace[0]：0x81 ncspace[1]：0x40
7	flags（空白変換レベル，DECIMAL 型の符号正規化機能の使用有無）	0
8	rdno（分割条件指定順序，又は分割キー内通番）	なし

注

文字コード種別，空白変換レベル，及び DECIMAL 型の符号正規化機能は，すべて省略しているものとします。

上記の表分割ハッシュ関数を実行すると，分割条件指定順序として **rdno=1** が返ってきます。

(ii) SQL の実行

表分割ハッシュ関数の結果を基に，次の SQL を実行し，RD エリア名を特定します。

```
SELECT RDAREA_NAME,DCVALUES
    FROM MASTER.SQL_DIV_TABLE
      WHERE TABLE_SCHEMA='USER1'      /* ユーザ名 */
        AND TABLE_NAME='KESSAI'       /* ハッシュ分割表名 */
        AND MOD(DIV_NO-1,33)+1 = 1    /* RDエリア定義順序 */
```

上記の SQL を実行すると，次に示すように削除対象の RD エリア名として **RDA1，RDA34 及び RDA67** が返ってきます。

```
 RDAREA_NAME                    DCVALUES※
 ------------------------------ ----------------------
 RDA1                           100
 RDA34                          200
 RDA67                          ********************
```

注※

DCVALUES 列も同時に検索することによって，RD エリアが第 1 次元のどの分割に対応するかの情報も得られます。

(b)　CREATE PUBLIC FUNCTION を定義しておく方法

(i) CREATE PUBLIC FUNCTION の定義

ハッシュ関数 HASHZ の場合，次の SQL をあらかじめ定義しておくことで RD エリアを特定できます。

```
CREATE PUBLIC FUNCTION HASHZ
 (IDATE VARCHAR(8)※,         /*  'YYYYMMDD'  */
  NDIV INT)                  /* 分割列数 */
RETURNS INT
  BEGIN
    RETURN MOD(DAYS(DATE(IDATE,'yyyymmdd'),NDIV)+1;  /* HASHZによる戻り値 */
  END
END_FUNC;
```

注※

列のデータは CHAR(8)ですが，パラメタは VARCHAR でなければ文字列定数を指定した呼び出しができません。

(ii) SQL の実行

次の SQL で，2011 年 7 月 31 日のデータがどこに入っているのかを検索します。

```
SELECT RDAREA_NAME,DCVALUES
  FROM MASTER.SQL_DIV_TABLE
    WHERE TABLE_SCHEMA='USER1'          /* ユーザ名 */
      AND TABLE_NAME='KESSAI'           /* ハッシュ分割表名 */
      AND MOD(DIV_NO-1,33)+1=HASHZ('20110731',33)    /* RDエリア定義順序 */
```

上記の SQL を実行すると，次に示すように削除対象の RD エリア名として **RDA1，RDA34 及び RDA67** が返ってきます。

```
RDAREA_NAME                    DCVALUES※
------------------------------ --------------------
RDA1                           100
RDA34                          200
RDA67                          ********************
```

注※

DCVALUES 列も同時に検索することによって，RD エリアが第 1 次元のどの分割に対応するかの情報も得られます。

(2) pdhold コマンドでアンロード対象 RD エリアを閉塞します

```
pdhold -r RDA1
pdhold -r RDA34
pdhold -r RDA67
```

(3) pdrorg コマンドで KESSAI 表の RDA1，RDA34 及び RDA67 のデータをアンロードします

```
pdrorg -k unld -t KESSAI -r RDA1 unload_control_file1
```

```
pdrorg -k unld -t KESSAI -r RDA34 unload_control_file34
```

```
pdrorg -k unld -t KESSAI -r RDA67 unload_control_file67
```

〔説明〕

-k：アンロードをするため unld を指定します。

-t：アンロードする表の名称を指定します。

-r：特定の RD エリアだけアンロードするため，対象となる RD エリア名を指定します。

unload_control_file：pdrorg コマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

＜ RDA1 の場合＞

```
unload bes1:C:¥pdrorg¥unload_file1

/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file1：アンロードデータファイルの名称 */
```

＜ RDA34 の場合＞

```
unload bes1:C:¥pdrorg¥unload_file34
```

```
/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file34：アンロードデータファイルの名称 */
```

＜RDA67の場合＞

```
unload bes1:C:¥pdrorg¥unload_file67

/* bes1：アンロードデータファイルがあるサーバの名称 */
/* C:¥pdrorg¥unload_file67：アンロードデータファイルの名称 */
```

(4) pdloadコマンドでRDA1，RDA34及びRDA67に対して0件のデータロードをします

最も古い2011年7月31日のデータ（RDA1，RDA34及びRDA67のデータ）を削除するため，0件のデータロードを実行します。

```
pdload -d KESSAI load_control_file1
```

```
pdload -d KESSAI load_control_file34
```

```
pdload -d KESSAI load_control_file67
```

〔説明〕

-d：既存のデータを削除してからデータロードをするために指定します。

KESSAI：データロードする表の名称を指定します。

load_control_file：pdloadコマンドの制御情報ファイル名を指定します。制御情報ファイルの内容を次に示します。

＜RDA1の場合＞

```
source RDA1 bes1:C:¥pdload¥load_file1

/* RDA1：0件のデータをデータロードするRDエリアの名称 */
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file1：ロードする0件データファイルの名称 */
```

＜RDA34の場合＞

```
source RDA34 bes1:C:¥pdload¥load_file34

/* RDA34：0件のデータをデータロードするRDエリアの名称 */
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file34：ロードする0件データファイルの名称 */
```

＜RDA67の場合＞

```
source RDA67 bes1:C:¥pdload¥load_file67

/* RDA67：0件のデータをデータロードするRDエリアの名称 */
```

```
/* bes1：ロードする0件データファイルがあるサーバの名称 */
/* C:¥pdload¥load_file67：ロードする0件データファイルの名称 */
```

分割表に，分割していない非分割インデクスを定義している場合，pdload コマンド実行後に非分割インデクスを一括作成してください。

(5) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDA1
pdrels -r RDA34
pdrels -r RDA67
```

コマンドの実行後，UAP などで 2011 年 9 月 2 日のデータを挿入すると，RDA1，RDA34，RDA67 にデータが格納されます。

なお，コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

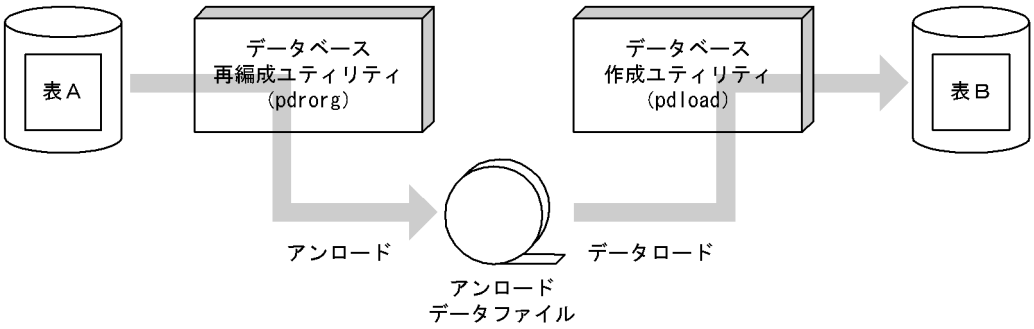
13.16　別表へのデータの移行方法

実行者　HiRDB 管理者，DBA 権限保持者，及び表の所有者

別表へデータを移行できます。データを移行するには，**pdrorg** 及び **pdload** コマンドを実行します。

別表へのデータの移行方法を次の図に示します。

図 13–92　別表へのデータの移行方法

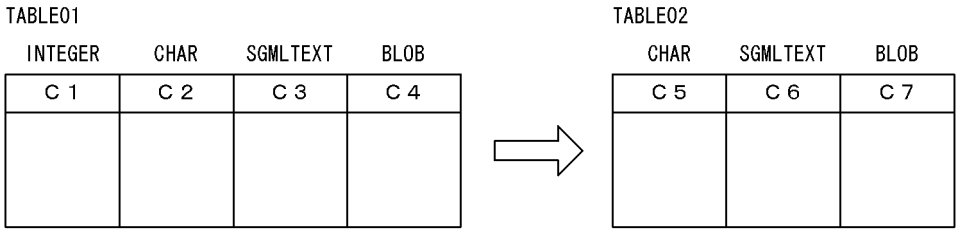


留意事項

- データの移行後，移行先の表に対して，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
- 別の HiRDB システムに表を移行する場合，表に抽象データ型の列が定義されている場合，データベース再編成ユティリティ（pdrorg コマンド）で抽象データ型のデータをアンロードするときにコンストラクタパラメタ逆生成関数を使う必要があります。

13.16.1　例題１（別表へデータを移行する場合）

TABLE01 のデータを TABLE02 に移行（コピー）します。TABLE01 と TABLE02 の表定義は次のように異なっているとします。



(1) pdhold コマンドでアンロード及びデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2) pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a) HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                                  1
unld_func type=sgmltext,func=unsgmltext(sgmltext)     2
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. 表に SGMLTEXT 型が定義されているため，コンストラクタパラメタ逆生成関数の情報を指定します。

(b) HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1                            1
unld_func type=sgmltext,func=unsgmltext(sgmltext)    2
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. 表に SGMLTEXT 型が定義されているため，コンストラクタパラメタ逆生成関数の情報を指定します。

(3) pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥rorg01：pdrorg コマンドの制御文ファイル名を指定します。

(4)　列構成情報ファイル（C:¥pdload¥column01）を作成します

```
*skipdata*,type=integer
C5
C6,func=(sgmltext,param=blob)
C7
```

(3)で作成した入力データファイルの列構成の形式と TABLE02 の列構成が異なるため，データロードするときに列構成情報ファイルが必要になります。列構成情報ファイルの指定例については，「列構成情報ファイルの指定例」を参照してください。

(5)　pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
source C:¥pdrorg¥unfile1
```

〔説明〕

(3)で作成したアンロードデータファイルの名称を指定します。

(b)　HiRDB/パラレルサーバの場合

```
source bes1:C:¥pdrorg¥unfile1
```

〔説明〕

(3)で作成したアンロードデータファイルの名称を指定します。

(6)　pdload コマンドで TABLE02 にデータをロードします

```
pdload -b -W -c C:¥pdload¥column01 TABLE02 C:¥pdload¥load01
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

-c C:¥pdload¥column01：(4)で作成した列構成情報ファイル名を指定します。

TABLE02：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(7)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(8)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA01,RDAREA02,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.16.2　例題２（XML 型の列を含むデータを，表単位で移行する場合）

表の列に XML 型を含むデータを，表単位で移行する場合の方法を説明します。

TABLE01 と TABLE02 は表定義が同じとします。TABLE01 のデータを TABLE02 に移行（コピー）します。

(1)　pdhold コマンドでアンロード及びデータロード対象 RD エリアを閉塞します

```
pdhold -r RDAREA01,RDAREA02,…
```

(2)　アンロード時に実行する pdrorg コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdrorg¥rorg01）の内容を次に示します。

(a)　HiRDB/シングルサーバの場合

```
unload C:¥pdrorg¥unfile1                          1
unld_func type=XML,func=extracts(XML)             2
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. コンストラクタパラメタ逆生成関数（extracts）を指定します。

(b)　HiRDB/パラレルサーバの場合

```
unload bes1:C:¥pdrorg¥unfile1                     1
unld_func type=XML,func=extracts(XML)             2
```

〔説明〕

1. アンロードデータファイルの名称を指定します。
2. コンストラクタパラメタ逆生成関数（extracts）を指定します。

(3) pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥rorg01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥rorg01：pdrorg コマンドの制御文ファイル名を指定します。

(4) データロード時に実行する pdload コマンドの制御文ファイルを作成します

制御文ファイル（C:¥pdload¥load01）の内容を次に示します。

(a) HiRDB/シングルサーバの場合

```
source C:¥pdrorg¥unfile1                                    1
```

〔説明〕

1. アンロードデータファイルの名称を指定します。

(b) HiRDB/パラレルサーバの場合

```
source bes1:C:¥pdrorg¥unfile1                               1
```

〔説明〕

1. アンロードデータファイルの名称を指定します。

(5) pdload コマンドで TABLE02 にデータをデータロードします

```
pdload -b -W TABLE02 C:¥pdload¥load01
```

〔説明〕

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE02：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

(6)　データロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でデータロードを実行したため，データロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(7)　pdrels コマンドで RD エリアの閉塞を解除します

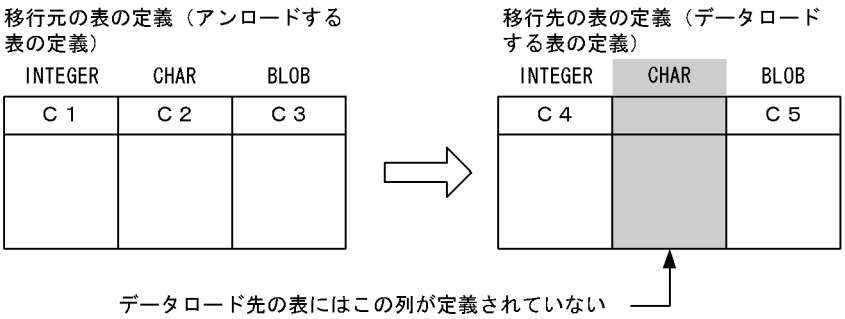
```
pdrels -r RDAREA01,RDAREA02,…
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.16.3　列構成情報ファイルの指定例

列構成情報ファイルの指定例を次に示します。

(1)　例１（列が削除されている場合）



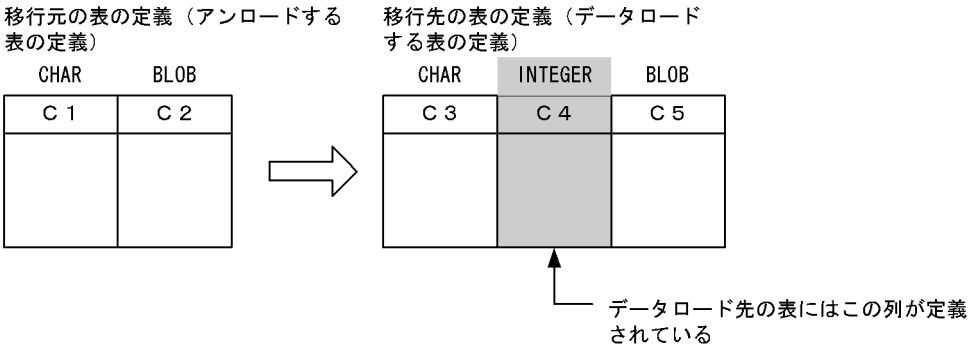
●列構成情報ファイルの指定例

```
C4
*skipdata*,type=char(4)
C5
```

〔説明〕

移行先の表には CHAR 型の列がないため，skipdata 文を指定します。

(2) 例２（列が追加されている場合）



●列構成情報ファイルの指定例

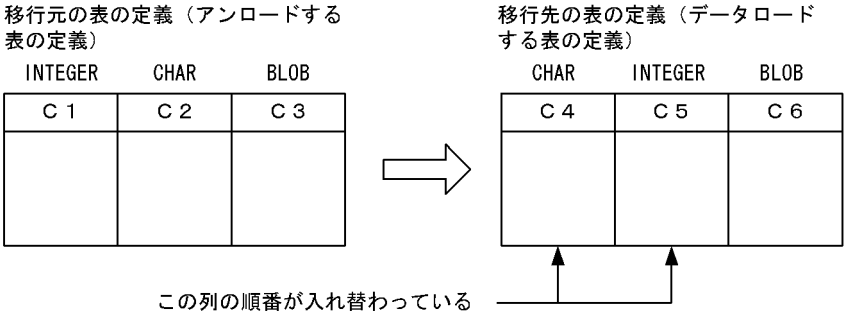
```
C3
C5
```

〔説明〕

移行先の表には C4 列が追加されています。列構成情報ファイルには追加した C4 列を記述しません。

列が追加されている場合の列構成情報ファイル指定のポイントは，追加された列を指定しないことです。こうすることで，HiRDB は入力データ中にその列のデータがないことを検知し，追加された列に省略値又はナル値を格納します。FIX 表の場合はナル値を格納できないため，ALTER TABLE で列を追加するときに WITH DEFAULT オペランドを指定して省略値を格納するようにします。

(3) 例３（列の順番が変更されている場合）



●列構成情報ファイルの指定例

```
C5
C4
C6
```

〔説明〕

C4 と C5 列の順番が入れ替わっているため，C5，C4 の順番で指定します。

13.17　表を削除する方法

実行者　表の所有者又は DBA 権限保持者

不要になった表を削除する場合は **DROP TABLE** を実行します。表に格納されているデータは削除されるため，ユーザ用 RD エリアに空き領域ができます。このため，ユーザ用 RD エリアを有効に利用できます。

注意事項

1. 表を削除すると，次に示すリソースも削除されます。
   - 表のインデクス
   - その表を基にして作成したビュー表
   - その表に対するアクセス権限
   - その表に定義したトリガ
2. 表を削除すると，その表を使用しているストアドルーチンが無効になります。
3. トリガ SQL 文中に指定している表を削除すると，トリガが無効になります。
4. データを格納している改竄防止表は削除できません。

13.18　スキーマを削除する方法

実行者　スキーマの所有者又は DBA 権限保持者

スキーマを削除するには **DROP SCHEMA** を使用します。

13.18.1　スキーマを削除すると削除されるリソース

スキーマを削除すると，そのスキーマ所有者のリソースも削除されます。削除されるリソースを次に示します。

- 表
- インデクス
- ビュー表
- コメント
- アクセス権限
- 抽象データ型
- インデクス型
- ストアドプロシジャ
- ストアドファンクション
- トリガ

13.18.2　他人のリソースにも影響があります

ビュー表が削除されます

スキーマ所有者の表及びビュー表が削除されるため，そのリソースを基にして作成したビュー表が削除されます。

アクセス権限が削除されます

スキーマ所有者の表及びビュー表が削除されるため，そのリソースに対するアクセス権限が削除されます。

ストアドルーチン及びトリガが無効になります

スキーマ所有者の表，ビュー表，インデクス，抽象データ型，ストアドルーチン，及びトリガが削除されるため，そのリソースを使用しているストアドルーチン及びトリガが無効になります。

13.18.3　スキーマが削除できない場合

- 自分の表に定義した抽象データ型を使用しているユーザがいると，スキーマを削除できません。
- データを格納している改竄防止表を所有しているユーザのスキーマは削除できません。

13.19　抽象データ型を削除する方法

13.19.1　抽象データ型を削除する方法と注意事項

実行者　表の所有者又は DBA 権限保持者

抽象データ型を削除するには，**DROP DATA TYPE** を使用します。

注意事項

削除対象の抽象データ型を使用するリソース（表，インデクス，及び抽象データ型）がある場合，その抽象データ型は削除できません。

(1)　抽象データ型を削除できない場合があります

削除するデータ型又は削除するデータ型の上位のデータ型が次に示す条件を満たす場合，DROP DATA TYPE で **WITH PROGRAM** を指定してください。指定しないと削除できません。

- 手続き又は関数中の SQL パラメタに指定しているデータ型
- 関数の戻り値に指定しているデータ型
- 手続き又は関数から呼び出している関数の引数及び戻り値で指定しているデータ型
- 手続き，関数，又はトリガ中に指定しているデータ型（コンポネント指定で抽象データ型をアクセスする場合，その途中のデータ型も含みます）

抽象データ型を削除すると，前記の条件に該当するストアドルーチン及びトリガが無効になります。この場合，ALTER PROCEDURE，ALTER ROUTINE，又は ALTER TRIGGER でストアドルーチン及びトリガを再作成してください。

(2)　サブタイプは削除できません

- 削除対象のデータ型の上位のデータ型が表定義に使用されていると，そのデータ型は削除できません。
- 削除対象のデータ型，及び削除対象のデータ型の上位のデータ型がほかの抽象データ型に使用されていると，そのデータ型は削除できません。

13.20　既存の表の定義系 SQL を作成する方法

実行者　HiRDB 管理者又は DBA 権限保持者

pddefrev コマンドで，現在使用している表の定義系 SQL を作成できます。

現在使用している表と同じ定義の表を作成するときに使用すると便利です。pddefrev コマンドで作成された定義系 SQL は，データベース定義ユティリティ（pddef コマンド）の入力情報となります。

> **参考**
>
> 抽象データ型を定義した表は，pddefrev コマンドの対象外となります（定義系 SQL を作成できません）。

13.21　リストの管理（絞込み検索）

13.21.1　リストの管理方法

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

ここでは，絞込み検索をするときに使用するリストの管理方法について説明します。リストを使用している場合，次に示すことに注意してください。

(1)　HiRDB を終了するとすべてのリストが削除されます

HiRDB を終了（異常終了を含む）するとすべてのリストが削除されるため，リストを使用した検索ができなくなります。リストが必要な場合は，**ASSIGN LIST 文**で再度リストを作成してください。

HiRDB/パラレルサーバの場合

1. ユニット単位の終了をすると，そのユニットにあるすべてのリストが削除されます。
2. サーバ単位の終了をすると，そのサーバにあるすべてのリストが削除されます。

(2)　リストの基表をデータベース回復ユティリティで回復した場合

リストの基表を最新の同期点の状態まで回復した場合，リストはそのまま使用できます。しかし，最新の同期点の状態まで回復しなかった場合（例えば，バックアップ取得時点に回復した場合），基表とリストの整合性が合わなくなる可能性があるため，**ASSIGN LIST 文**でリストを再作成してください。

(3)　リストの基表を格納している RD エリアを再初期化した場合

リストの基表を格納している RD エリアを再初期化した場合，そのリストに対して次に示すどちらかの処置をしてください。

- **ASSIGN LIST 文**でリストを再作成します。
- **DROP LIST 文**でリストを削除します。

(4)　リストが無効になる操作

リストの基表に対して次に示す操作をした場合，リストの検索結果が不正になります。この場合，**ASSIGN LIST 文**でリストを再作成してください。

- 表の再編成
- 作成モードでの表へのデータロード
- PURGE TABLE 文の実行

(5)　リスト及びリスト用 RD エリアに対して使用できないコマンドがあります

リスト及びリスト用 RD エリアに対して，次に示すコマンドが使用できません。

- データベース作成ユティリティ
- データベース再編成ユティリティ
- データベース複写ユティリティ
- データベース回復ユティリティ
- データベース状態解析ユティリティ（RD エリア単位の物理的解析だけはできます）

(6)　ディクショナリサーバが終了したときの留意事項

リストの作成及び削除のオーバヘッド低減のため，HiRDB ではリストの管理情報をディクショナリサーバのメモリに保持しています。このため，ディクショナリサーバ又はディクショナリサーバがあるユニットを終了すると，リストの管理情報が失われます。その結果，これまで作成した全リストが無効になります。この場合，**ASSIGN LIST 文**で再度リストを作成してください。

なお，ディクショナリサーバの再開始後，終了以前に実行していた他ユーザのリストを使用したトランザクションの回復が完了するまでの間，リストを使用した処理が KFPA11998-E エラー（トランザクション未決着状態でのリスト操作）となることがあります。

(7)　リストの情報を調べるコマンド（pdlistls コマンド）

pdlistls コマンドでリストの情報を調べられます。表示される情報は次のとおりです。

- リスト名
- リストの所有者
- リストの基表名
- リストの基表の所有者
- pd_max_list_users の値（リスト所有者数の最大値）
- pd_max_list_count の値（1 ユーザ当たりの最大リスト作成数）

(8)　リストの初期化（削除）時期の変更

(1)で説明したように，HiRDB を終了するとリストが削除されます。このリストの削除処理（初期化処理）は HiRDB の開始処理時に行われるため，作成したリスト数が多いと HiRDB の開始処理に時間が掛かります。次に示すような場合に，リストの初期化時期の変更を検討してください。

- HiRDB の開始時間を短縮したい
- 系の切り替え時間を短縮したい（ユーザサーバホットスタンバイ又は高速系切り替え機能を使用している）

リストの初期化時期は **pd_list_initialize_timing** オペランドで次のように変更できます。

- ASSIGN LIST 文の実行時にリストを初期化する（DEFER 指定）
- 待機系の HiRDB を開始するときにリストを初期化する（STANDBY 指定）

(a) ASSIGN LIST 文の実行時にリストを初期化する場合

HiRDB を開始するときにリストを初期化しないで，ASSIGN LIST 文を実行するときにリストを初期化します。そのため，ASSIGN LIST 文を実行するときに初期化する分のオーバヘッドが掛かります。この初期化処理のオーバヘッドを小さくするには，リスト用 RD エリアの容量及び最大リスト数をなるべく小さく設定してください。リスト作成数が多い場合は，リスト用 RD エリアの数を増やして対応してください。これで，ASSIGN LIST 文実行時の初期化処理のオーバヘッドを分散できます。

(b) 待機系の HiRDB を開始するときにリストを初期化する場合（高速系切り替え機能限定）

待機系の HiRDB を開始するときにリストを初期化します。系の切り替え時にはリストを初期化しません。また，系切り替え後の ASSIGN LIST 文の実行時にもリストを初期化しません。そのため，リスト用 RD エリアの容量及び最大リスト数を大きく設定し，作成するリスト用 RD エリア数を少なくできます。ただし，次に説明する準備作業をする必要があります。

準備作業

実行系と待機系の両方にリストが必要になります。それぞれのローカルディスクに作成してください。次の二つの方法があります。

実行系の HiRDB で作成したリストを待機系の HiRDB にコピーする方法

1. 実行系の HiRDB のリスト用 RD エリアを格納している全 HiRDB ファイルシステム領域を待機系のローカルディスクにコピーします。
2. リスト用 RD エリアに定義されているすべての HiRDB ファイル名が，1 で作成した HiRDB ファイルシステム領域を指すようにリンクを設定します。

データベース構成変更ユティリティ（pdmod コマンド）の再初期化を使用する方法

1. **pdstop** コマンドで実行系の HiRDB を正常終了します。
2. **pdfmkfs** コマンドで，リスト用 RD エリアを作成する HiRDB ファイルシステム領域を作成します。実行系と同じ設定，同じ数だけ作成してください。
3. リスト用 RD エリアに定義されているすべての HiRDB ファイル名が，2 で作成した HiRDB ファイルシステム領域を指すようにリンクを設定します。
4. **pdstart** コマンドで待機系の HiRDB を正常開始します。このとき，待機系の HiRDB には HiRDB ファイルシステム領域内に HiRDB ファイルがないため，リスト用 RD エリアがオープン，障害閉塞状態になります。
5. **pdmod** コマンドですべてのリスト用 RD エリアを再初期化します。このとき，再初期化する RD エリア名だけを指定してください。

6. **pdrels** コマンドですべてのリスト用 RD エリアの閉塞を解除します。
7. **pdstop** コマンドで待機系の HiRDB を正常終了します。

(9)　リスト用 RD エリアの構成変更

(8)(b)に示した実行系と待機系のローカルディスクに別々にリスト用 RD エリアを作成している場合，(8)の再初期化以外の RD エリアの構成変更は，次の手順で行います。

1. 実行系でリスト用 RD エリアの構成を変更します。
2. **pdstop** コマンドで実行系の HiRDB を正常終了します。
3. 実行系の HiRDB のリスト用 RD エリアを格納しているすべての HiRDB ファイルシステム領域を待機系のローカルディスクにコピーします。
4. リスト用 RD エリアに定義されているすべての HiRDB ファイル名が，3 で作成した HiRDB ファイルシステム領域を指すようにリンクを設定します。

13.22　表データ中の空白文字を統一する方法

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

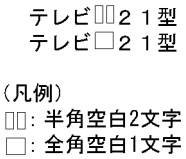
ここでは，全角の空白と半角の空白が表データに混在している場合の空白文字の統一方法について説明します。空白データを統一するには，**空白変換機能**を使用します。

13.22.1　空白変換機能とは

データを比較するとき，全角の空白 1 文字と半角の空白 2 文字は異なるデータと認識されます。したがって，全角の空白 1 文字と半角の空白 2 文字が表データ中に混在していると，検索結果が不正になることがあります。

（例）

次に示すデータは異なるデータと認識されます。



なお，ここでいう全角空白とは次に示すコードのことです。半角空白 2 文字とは，X'2020'のことです。

- シフト JIS 漢字の場合：X'8140'
- EUC 中国語漢字の場合：X'A1A1'
- 中国語漢字コード（GB18030）の場合※：X'A1A1'
- Unicode（UTF-8）又は Unicode（IVS 対応 UTF-8）の場合※：X'E38080'

注※　文字コードが Unicode（UTF-8），Unicode（IVS 対応 UTF-8）又は中国語漢字コード（GB18030）の場合は，NCHAR 及び NVARCHAR を使用できません。

(1)　空白変換レベル

空白変換機能には三つのレベルがあり，次の表に示すように空白を変換します。

表 13–44　空白変換レベル

レベル	説明
レベル 0	空白変換をしません。
レベル 1	操作系 SQL での定数，埋込み変数，又は？パラメタのデータの空白，及びユティリティで格納するデータの空白を次のように変換します。

レベル	説明
	• 文字列定数を各国文字列定数とみなした場合，半角空白 2 文字を全角空白 1 文字に変換します。なお，半角空白が 1 文字単独で現れる場合は変換しません。 • 文字列定数を混在文字列定数とみなした場合，全角空白 1 文字を半角空白 2 文字に変換します。 • 各国文字列型の列へのデータの格納時，及び各国文字列型の値式との比較時は，埋込み変数，又は？パラメタの半角空白 2 文字を全角空白 1 文字に変換します。なお，半角空白が 1 文字単独で現れる場合は変換しません。 • 混在文字列型の列へのデータの格納時，及び混在文字列型の値式との比較時は，埋込み変数，又は？パラメタの全角空白 1 文字を半角空白 2 文字に変換します。
レベル 3	空白変換レベル 1 の処理に加えて次の処理が加わります。 • 各国文字列型の値式のデータを検索するときに，全角空白 1 文字を半角空白 2 文字に変換します。

レベル 1 の処理方式とレベル 3 の処理方式を次の図に示します。

図 13–93　レベル 1 の処理方式

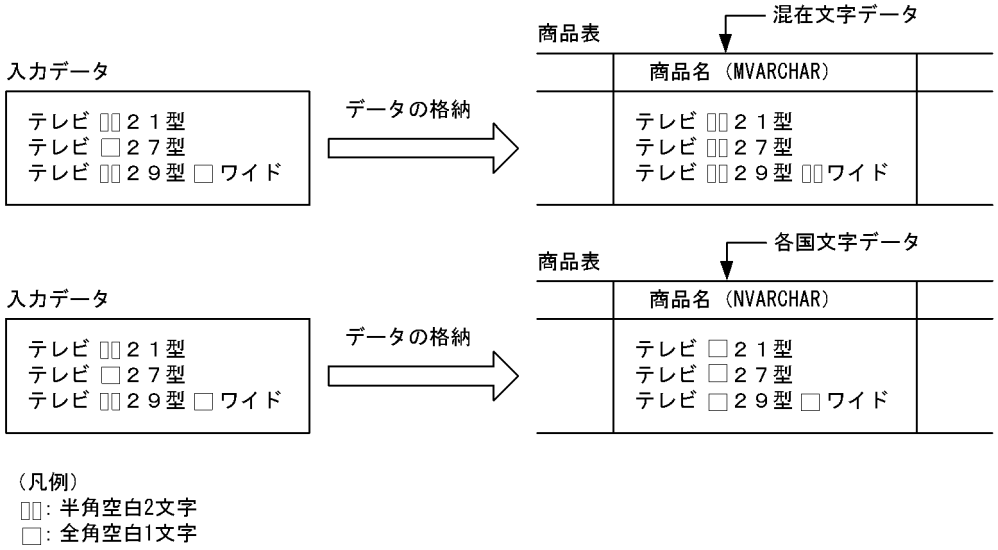
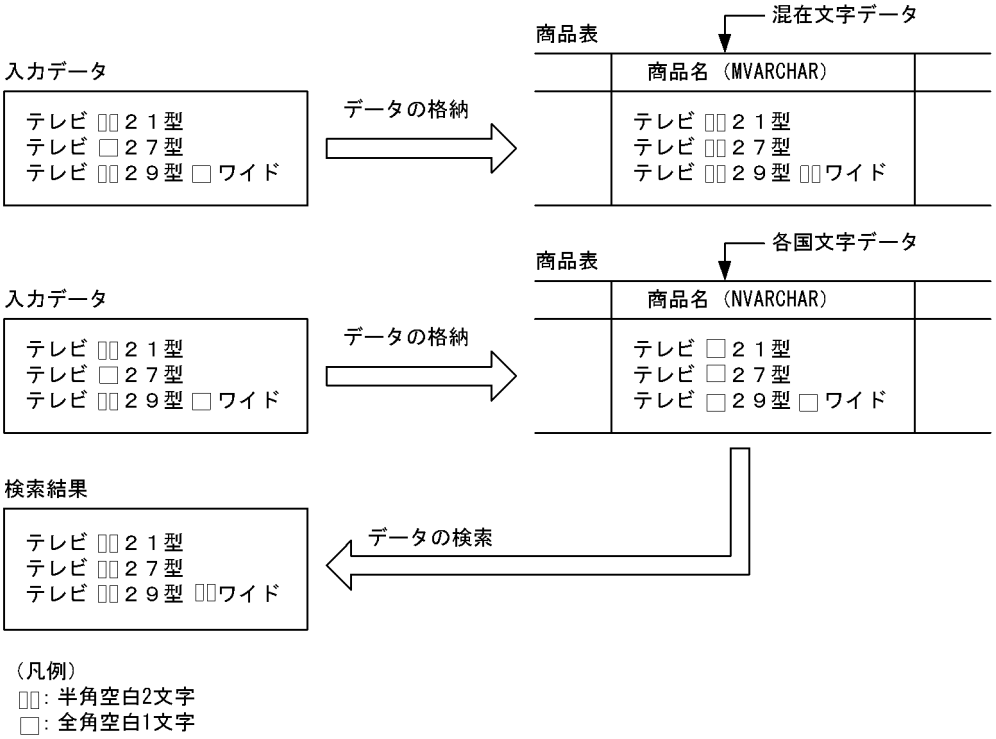


図 13–94　レベル 3 の処理方式



(2)　注意事項

1. 空白変換レベルを変更すると，変更の前後で UAP の検索結果が変わることがあります。UAP の検索結果を変えたくない場合は，空白変換レベルを変更しないでください。
2. レベル 3 を指定してソートをした場合，HiRDB はソートの結果に対して空白変換をするため，期待した結果を得られないことがあります。
3. レベル 3 を指定して各国文字データを検索した場合，レベル 0 又は 1 のときに比べて性能が劣化することがあります。例えば，NVARCHAR（2000）などのデータ長が長いデータを検索するときに性能が劣化します。これは，全角空白を半角空白に変換するオーバヘッドが発生するためです。
4. クラスタキーの列へデータを格納する場合，空白変換によってユニークエラーとなることがあります。この場合，空白変換をしないでデータを格納するか，又は既存データの空白文字を統一してください。既存データの空白文字の統一方法については，「表中の空白文字の統一方法」を参照してください。
5. 各国文字列の空白変換は，先頭から 2 文字単位で変換します。
6. 空白変換レベルに 1 又は 3 を指定したときの注意事項を次に示します。
   - ハッシュ分割した表に対して，表分割ハッシュ関数を使用した UAP で格納先 RD エリアを求める場合は，表分割ハッシュ関数の引数に空白変換レベルを指定してください。指定しないと表分割ハッシュ関数の結果が不正になることがあります。
   - キーレンジ分割（分割キーが各国文字データ又は混在文字データの場合）した表に対して，UAP でキーレンジ分割処理をしている場合は，分割キー値を空白変換関数で変換してください。変換しないとキーレンジ分割の結果が不正になることがあります。

表分割ハッシュ関数，及び空白変換関数については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

13.22.2　空白変換レベルの設定方法

空白変換レベルは次に示すオペランドで指定できます。

- システム共通定義の **pd_space_level** オペランド
- クライアント環境定義の **PDSPACELVL** オペランド
- データベース作成ユティリティの option 文の **spacelvl** オペランド
- データベース再編成ユティリティの option 文の **spacelvl** オペランド

13.22.3　表中の空白文字の統一方法

空白文字を統一するときの運用方法を次に示します。

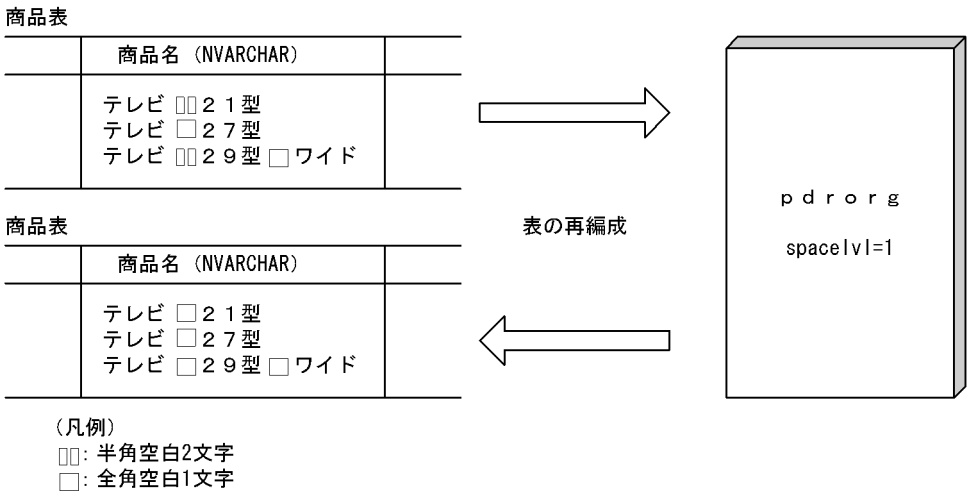
運用方法

1. 既存データの空白文字を統一します。「既存データの空白文字を統一する場合」を参照してください。
2. システム共通定義で pd_space_level=1 を指定します。
   これで新規に格納（データロード又は INSERT）するデータに関しては空白文字を統一できます。
3. 必要に応じてクライアント定義で PDSPACELVL=3 を指定してデータを検索してください。

(1)　既存データの空白文字を統一する場合

既存データの空白文字を統一する方法を次の図に示します。

図 13–95　既存データの空白文字を統一する方法



〔説明〕

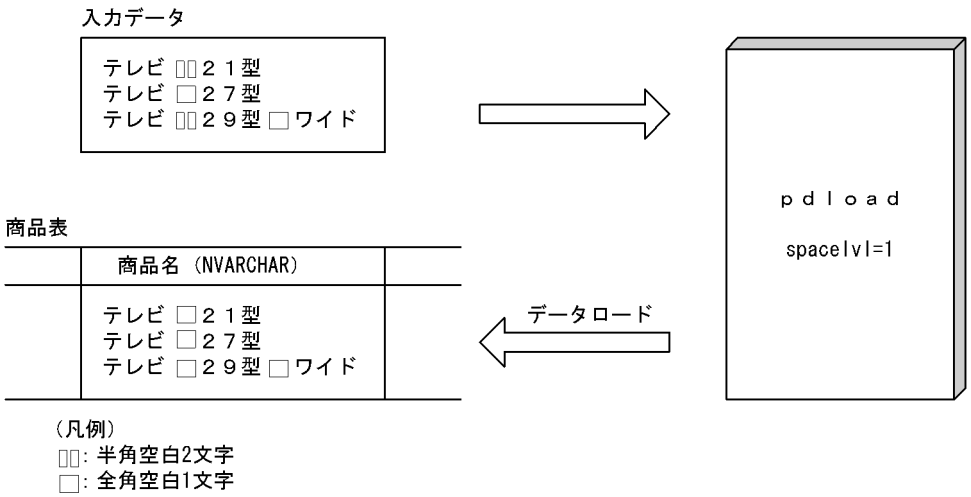
既存データの空白文字を統一するには，データベース再編成ユティリティ（pdrorg コマンド）で表を再編成してください。表を再編成するとき，pdrorg コマンドの option 文で **spacelvl=1** を指定してください。

表の再編成方法については，「表の再編成」及び「表の再編成（例題）」を参照してください。

(2)　新規データの空白文字を統一する場合

新規データの空白文字を統一する方法を次の図に示します。

図 13–96　新規データの空白文字を統一する方法



〔説明〕

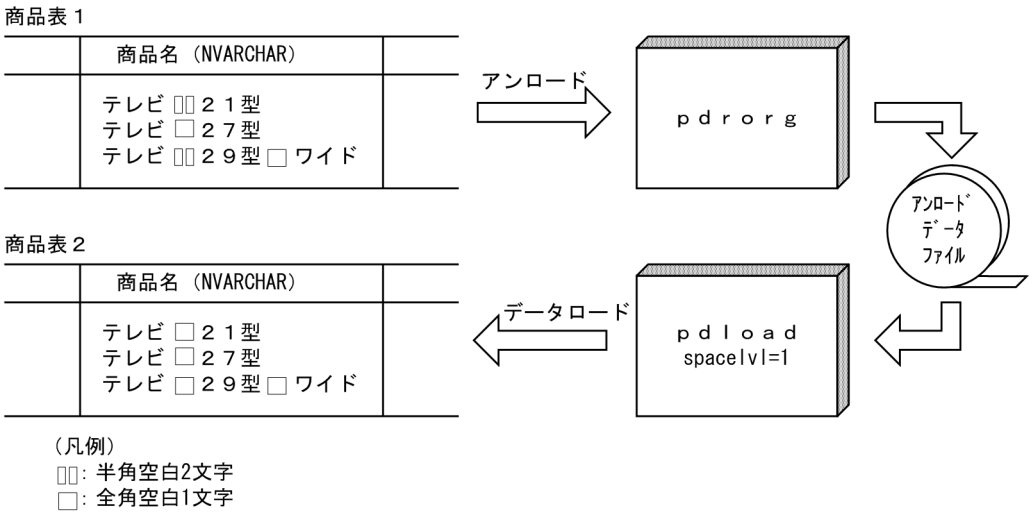
新規データの空白文字を統一する場合は，データベース作成ユティリティ（pdload コマンド）でデータロードをするときに，option 文で **spacelvl=1** を指定してください。なお，option 文の spacelvl オペランドを省略すると，システム共通定義の pd_space_level オペランドの指定が仮定されます。

データロードの方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(3)　別システムに表データを移行する場合

別システムに表データを移行する場合の空白文字の統一方法を次の図に示します。

図 13–97　別システムに表データを移行する場合の空白文字の統一方法



〔説明〕

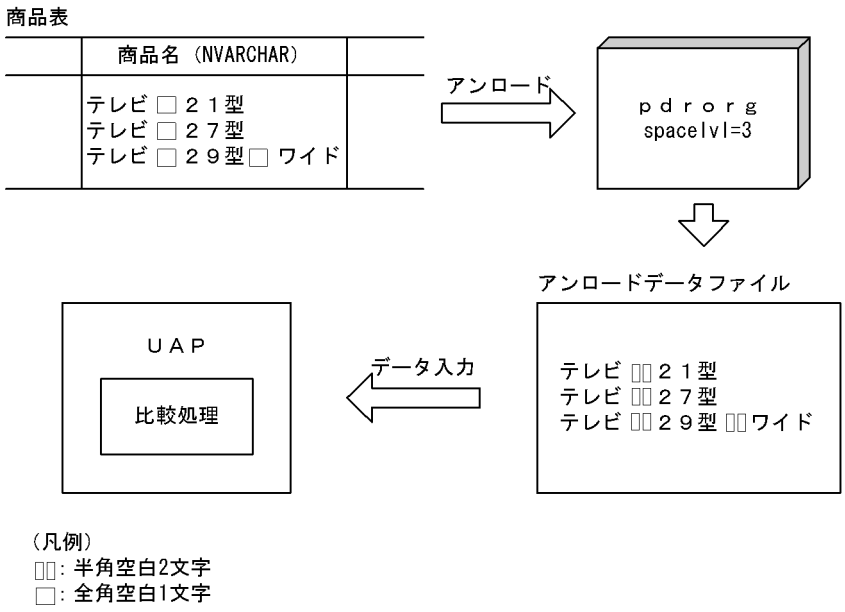
表データを別システムに移行する場合は，移行先システムでデータをデータロードするときに，データベース作成ユティリティ（pdload コマンド）の option 文で **spacelvl=1** を指定してください。

表データを別システムに移行する方法については，「表をほかの HiRDB システムに移行する方法」を参照してください。

(4)　-W オプション指定で出力したアンロードデータを UAP で使用する場合

-W オプション指定で出力したアンロードデータを UAP で使用する場合（この UAP で比較処理をする場合）の空白変換方法を次の図に示します。なお，この場合，表中の空白文字が統一されている必要があります。

図 13–98　-W オプション指定で出力したアンロードデータを UAP で使用する場合



〔説明〕

- データをアンロードするときにデータベース再編成ユティリティ（pdrorg コマンド）の option 文で **spacelvl=3** を指定してください。各国文字データの全角空白 1 文字が半角空白 2 文字に変換されます。
- UAP で実施する各国文字データの比較処理は半角空白で行われます。

13.23　DECIMAL 型の符号部を変換する方法

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

DECIMAL 型，日間隔型，及び時間隔型のデータ形式は，値の整数部分と符号部分で構成される符号付きパック形式です。通常，HiRDB では符号付きパック形式データの符号部分として X'C'（正），X'D'（負），X'F'（正）を有効な値として，UAP 又はユティリティから入力される符号をそのままデータベースに格納※しています。また，＋ 0（符号部 X'C'又は X'F'）と−0（符号部 X'D'）は異なる値としています。

DECIMAL 型の符号正規化機能を使用すると，DECIMAL 型，日間隔型，及び時間隔型の符号付きパック形式の符号部を変換できます。ここでは，DECIMAL 型の符号正規化機能を使用して符号付きパック形式の符号部を変換する方法について説明します。

注※

SQL 実行中の型変換又は演算などで符号が変換される場合があります。また，複数列インデクスを使用した場合に符号が変換される場合があります。

13.23.1　DECIMAL 型の符号正規化機能とは

(1)　符号付きパック形式の符号部の仕様

HiRDB では符号付きパック形式の符号部の仕様が次の表に示すようになっています。

表 13–45　符号付きパック形式の符号部の仕様

符号部	意味
X'C'	正の値を示しています。
X'D'	負の値を示しています。
X'F'	正の値を示しています。

(2)　符号付きパック形式の符号部の変換規則

DECIMAL 型の符号正規化機能を使用すると，データを入力したときに符号付きパック形式の符号部を次の表に示す規則に従って変換します。この符号部を変換することを**符号部を正規化する**といいます。符号部を正規化すると＋ 0 と−0 を同じ値として処理できます。

表 13–46　符号付きパック形式の符号部の変換規則（0 データ以外の場合）

埋込み変数のデータの符号部	正規化しない場合	正規化する場合
X'A'	エラー	X'C'に変換
X'B'	エラー	X'D'に変換

埋込み変数のデータの符号部	正規化しない場合	正規化する場合
X'C'	無変換	無変換
X'D'	無変換	無変換
X'E'	エラー	X'C'に変換
X'F'	無変換	X'C'に変換
X'0'～X'9'	エラー	エラー

表 13–47　符号付きパック形式の符号部の変換規則（0 データの場合）

0 データの符号部	正規化しない場合	正規化する場合
X'A'	エラー	X'C'に変換
X'B'	エラー	
X'C'	無変換	
X'D'	無変換	
X'E'	エラー	
X'F'	無変換	

(3)　適用基準

符号部に仕様差がある UAP を使用する場合に，DECIMAL 型の符号正規化機能を使用するとよいケースがあります。この場合，符号変換規則をよく確認してから DECIMAL 型の符号正規化機能を使用してください。

例えば，XDM/RD 又は XDM/RD E2 の UAP を HiRDB に移行した場合に，DECIMAL 型の符号正規化機能を使用するとよいケースがあります。XDM/RD と HiRDB は DECIMAL 型の符号部に仕様差があります。

(4)　環境設定

DECIMAL 型の符号正規化機能を使用するには，システム共通定義で **pd_dec_sign_normalize=Y** を指定します。

なるべく HiRDB の新規導入時に指定してください。既に HiRDB を運用しているときに符号部を正規化するには，DECIMAL 型を定義した表のデータをデータロードし直す必要があります。

(5)　注意事項

1. 正規化されたデータと正規化されていないデータが混在した状態で SQL を実行すると，不当なデータベース破壊を検知したり，実行結果が不正になることがあります。このため，pd_dec_sign_normalize オペランドの指定値を変更した場合，データベース再編成ユティリティ（pdrorg コマンド）及びデータベース作成ユティリティ（pdload コマンド）を使用し，データロードをし直して正規化状態を統一する必要があります。方法については，「既存のデータを正規化する方法」を参照してください。

2. DECIMAL 型の符号正規化機能を使用する場合の注意事項を次に示します。

   - ハッシュ分割した表に対して，表分割ハッシュ関数を使用した UAP で格納先 RD エリアを求める場合は，表分割ハッシュ関数の引数に「DECIMAL 型の符号正規化機能あり」を指定してください。指定しないと表分割ハッシュ関数の結果が不正になることがあります。
   - キーレンジ分割（分割キーが DECIMAL 型のデータの場合）した表に対して，UAP でキーレンジ分割処理をしている場合は，分割キー値を DECIMAL 型符号正規化関数で変換してください。変換しないとキーレンジ分割の結果が不正になることがあります。

   表分割ハッシュ関数，及び DECIMAL 型符号正規化関数については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

13.23.2　既存のデータを正規化する方法

表（TABLE01）には DECIMAL 型の列があります。この DECIMAL 型の符号部を正規化します。

(1)　pd_dec_sign_normalize=Y を指定します

pdstop コマンドで HiRDB を正常終了します。そして，**pd_dec_sign_normalize** オペランドに Y を指定します。その後，**pdstart** コマンドで HiRDB を正常開始してください。

システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できます。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(2)　pdrorg コマンドで TABLE01 のデータをアンロードします

```
pdrorg -k unld -W bin -t TABLE01 -g C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-W bin：アンロードデータファイルを pdload コマンドの入力ファイル（バイナリ形式）として使用できるようにします。

-t：アンロードする表の名称を指定します。

-g：HiRDB/パラレルサーバで，TABLE01 をサーバ間横分割している場合に指定します。-g オプションを指定すると，アンロードデータファイルが一元化されます（一つになります）。

C:¥pdrorg¥unld01：pdrorg コマンドの制御文ファイル名を指定します。

(3)　pdload コマンドで TABLE01 にデータロードします

```
pdload -d -b -W TABLE01 C:¥pdload¥load01
```

〔説明〕

-d：作成モードを指定します。

-b：バイナリ形式のデータをロードする場合に指定します。

-W：pdrorg コマンドで作成したバイナリ形式の入力データファイルを使用する場合に指定します。

TABLE01：データロードする表の名称を指定します。

C:¥pdload¥load01：pdload コマンドの制御文ファイル名を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

補足事項

前記の方法ではバイナリ形式の入力ファイル（-W bin）を使用していますが，DAT 形式の入力ファイル（-W dat）を使用した場合，アンロード時にいったん文字に変換されます。そして，データロード時の変換処理で自動的に符号部の正規化が行われます。したがって，pd_dec_sign_normalize オペランドの指定値に関係なく符号部の正規化が行われます。

14 インデクスの運用

この章では，インデクスの運用方法について説明します。

14.1　インデクスの格納効率を上げる方法（インデクスの再編成）

ここでは，インデクスの格納効率を上げる方法（インデクスの再編成）について説明します。

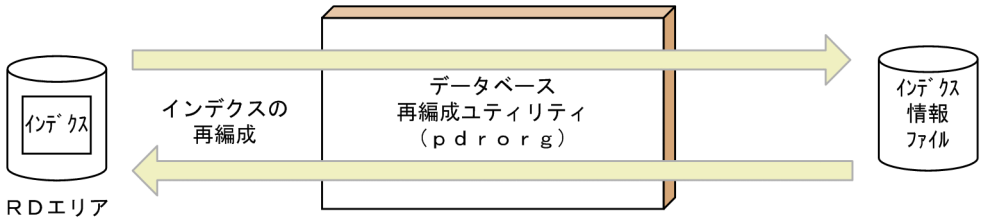
14.1.1　インデクスの再編成とは

データの削除（DELETE）及び更新（UPDATE）を繰り返すと，インデクスの格納効率が悪くなり，インデクスを使用した検索をするときの性能が低下します。これを防ぐためには，データベース再編成ユティリティで次に示すどれかの処置をとってください。

- インデクスの再編成
- インデクスの再作成
- 表の再編成

ここでは，性能的に優れているインデクスの再編成について説明します。インデクスの再編成の処理概要を次の図に示します。

図 14–1　インデクスの再編成の処理概要



〔説明〕

インデクスのキー情報を検索してインデクス情報ファイルを作成し，その情報を基にインデクスを再配置します。これをインデクスの再編成といいます。インデクスの再編成は，インデクス単位又はインデクス格納 RD エリア単位に実行できます。

(1)　適用基準

大量データの追加，削除，更新によって生じるインデクスの格納ページの無効領域を解放する場合に適用します。

(2)　表の再編成との使い分け

- データの更新（UPDATE）が多い場合は，インデクスだけを再編成してください。
- データの削除（DELETE）及び追加（INSERT）が多い場合は，表を再編成してください。
- 表を再編成する時間の余裕がない場合に，インデクスだけを再編成すれば，インデクス検索時間を短縮できます。

(3)　インデクスの再作成との違い

インデクス再作成の処理方式では表データを検索しますが，インデクス再編成の処理方式では表データを検索しません。したがって，インデクスの再作成に比べて処理時間が短くなり※，ソート処理も不要なため性能的にも優れています。ただし，プラグインインデクスは再編成できないため，再作成してください。

注※　次に示す条件を満たす場合に処理時間が短くなります。

表格納 RD エリアの使用中ページ数 ＞ インデクス格納 RD エリアの使用中ページ数

(4)　インデクスの再編成時の注意事項

- 同一 RD エリア内にある複数のインデクスを同時に再編成する場合は，インデクスを再編成する前に **pdhold** コマンドで RD エリアを閉塞状態にしてください。そして，インデクスの再編成終了後に **pdrels** コマンドで RD エリアの閉塞状態を解除してください。
- プラグインインデクスは再編成できないため，再作成してください。

(5)　再編成の実行時間を短縮する方法

インデクスの再編成をするとき，データベースの更新ログを取得しなければ（ログレスモード又は更新前ログ取得モード），その分の処理時間が短縮されます。データベースの更新ログ取得方式は，pdrorg コマンドの **-l** オプションで指定します。省略値は更新前ログ取得モードです。

(6)　満杯状態の RD エリア内のインデクスを再編成する場合

インデクスの再編成時，ページ内の未使用領域の比率は CREATE TABLE 又は CREATE INDEX の PCTFREE オペランドの指定が適用されます。したがって，満杯状態の RD エリア内のインデクスを再編成すると，インデクスの再編成時に RD エリアの容量不足が発生することがあります。これを防ぐには，pdrorg コマンドの **option 文**で **idxfree** オペランドを指定して，CREATE TABLE 又は CREATE INDEX の PCTFREE オペランドで指定したページ内の未使用領域の比率を変更してください。

なお，これはあくまで再編成前に RD エリアの拡張がすぐにできないときの暫定的な処置です。データの更新に備えて，CREATE TABLE の PCTFREE オペランドの値を考慮した再編成ができるように，データベース構成変更ユティリティ（pdmod コマンド）で RD エリアを拡張してください。

14.1.2　例題 1（インデクスの再編成方法）

表（TABLE01）に定義したインデクス（INDEX01）を再編成します。再編成するときの条件は次のとおりです。

- インデクス格納 RD エリアは RDAREA1 とします。
- RDAREA1 には INDEX01 だけが格納されています。
- 更新前ログ取得モードで再編成を実行します。

(1)　pdhold コマンドで再編成対象 RD エリアを閉塞します

```
pdhold -r RDAREA1
```

(2)　pdrorg コマンドでインデクスを再編成します

```
pdrorg -k ixor -t TABLE01 C:¥pdrorg¥rorg01
```

〔説明〕

-k：インデクスを再編成するため ixor を指定します。

-t：インデクスを定義した表の名称を指定します。

C:¥pdrorg¥rorg01：pdrorg コマンドの制御文情報ファイル名を指定します。制御文情報ファイルの指定例を次に示します。

```
idxname name=INDEX01
idxwork bes1 C:¥pdrorg
```

INDEX01：インデクス識別子

bes1：インデクス情報ファイルを作成するサーバ名（HiRDB/パラレルサーバの場合に指定します）

C:¥pdrorg：インデクス情報ファイルを作成するディレクトリ名

(3)　再編成対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でインデクスを再編成したため，再編成対象 RD エリア（RDAREA1）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(4)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.1.3　インデクスの再編成中にエラーが発生した場合の処置

インデクスの再編成中にエラーが発生して，pdrorg コマンドが異常終了した場合の処置について説明します。pdrorg コマンドが異常終了した後の操作手順を次に示します。

〈手順〉

1. メッセージログファイル又はイベントログ中に，**KFPL00715-I** メッセージが出力されているかどうかを確認してください。KFPL00715-I メッセージは，「インデクスロードの処理開始」を告げるメッセージです。このメッセージが出力されている場合は，インデクス情報ファイルが作成されています。
2. KFPL00715-I メッセージが出力されていない場合は，再度 pdrorg コマンド（インデクスの再編成）を実行してください。ただし，pdcancel などによる異常終了の場合は，インデクスキー情報がなくなっている可能性があるため，インデクスの再編成を再実行できないことがあります。この場合は，インデクスの再作成(-k ixrc 指定)を実行してください。
3. KFPL00715-I メッセージが出力されている場合は，作成されたインデクス情報ファイルを入力ファイルとして次に示す操作をしてください。

- **インデクスの再編成をログレスモード以外で実行した場合**

  pdrorg コマンドでインデクスを一括作成（-k ixmk 指定）してください。

- **インデクスの再編成をログレスモードで実行した場合**

  インデクス格納 RD エリアは障害閉塞しています。したがって，pdmod コマンドでインデクス格納 RD エリアを再初期化して，障害閉塞状態を解除してください。その後，pdrorg コマンドでインデクスを再作成（-k ixrc 指定）してください。

留意事項

インデクスの再編成時に起こる一般的なエラーとして，インデクス格納 RD エリアの容量不足があります。この場合の対処方法を「例題 2（インデクスの再編成中に RD エリアの容量不足が発生した場合：ログレスモード以外で実行した場合）」及び「例題 3（インデクスの再編成中に RD エリアの容量不足が発生した場合：ログレスモードで実行した場合）」で説明します。

なお，インデクスの定義時に PCTFREE オペランドに 0 以外を指定している場合は，pdrorg コマンドの option 文の idxfree オペランドを指定すれば，インデクス格納 RD エリアを拡張しなくても済む場合があります。

14.1.4 例題 2（インデクスの再編成中に RD エリアの容量不足が発生した場合：ログレスモード以外で実行した場合）

インデクスの再編成中にインデクス格納 RD エリアの容量不足で pdrorg コマンドが異常終了しました。

(1) インデクス格納 RD エリアの容量を拡張します

RD エリアの容量を拡張する方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

(2) pdhold コマンドでインデクス格納 RD エリアを閉塞します

```
pdhold -r RDAREA1
```

(3)　pdrorg コマンドでインデクスを一括作成します

インデクスの再編成時に出力したインデクス情報ファイルを入力情報にして，インデクスの一括作成（-k ixmk 指定）を実行してください。

```
pdrorg -k ixmk -t TABLE01 C:¥pdrorg¥rorg01
```

〔説明〕

-k：インデクスを一括作成するため ixmk を指定します。

-t：インデクスを定義した表の名称を指定します。

C:¥pdrorg¥rorg01：pdrorg コマンドの制御文情報ファイル名を指定します。制御文情報ファイルの指定例を次に示します。

```
index INDEX01 RDAREA1 C:¥pdrorg¥index_inf01
```

INDEX01：インデクス識別子

RDAREA1：インデクスを格納している RD エリア名

C:¥pdrorg¥index_inf01：インデクス情報ファイル名

(4)　インデクス格納 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でインデクスを一括作成したため，インデクス格納 RD エリア（RDAREA1）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA1
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.1.5　例題 3（インデクスの再編成中に RD エリアの容量不足が発生した場合：ログレスモードで実行した場合）

インデクスの再編成中にインデクス格納 RD エリアの容量不足で pdrorg コマンドが異常終了しました。

(1)　インデクス格納 RD エリアを再初期化します

インデクス格納 RD エリアは障害閉塞状態になっています。したがって，pdmod コマンドでインデクス格納 RD エリアを再初期化して，障害閉塞状態を解除します。このとき，一緒にインデクス格納 RD エリアの

容量を大きくします。RD エリアの再初期化方法については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください。

(2) pdhold コマンドでインデクス格納 RD エリア及び表格納 RD エリアを閉塞します

```
pdhold -r RDAREA1,RDAREA2
```

(3) pdrorg コマンドでインデクスを再作成します

再初期化した RD エリア内のすべてのインデクスを再作成する必要があります。

```
pdrorg -k ixrc -t TABLE01 C:¥pdrorg¥rorg01
```

〔説明〕

-k：インデクスを再作成するため ixrc を指定します。

-t：インデクスを定義した表の名称を指定します。

C:¥pdrorg¥rorg01：pdrorg コマンドの制御文情報ファイル名を指定します。制御文情報ファイルの指定例を次に示します。

```
idxname name=INDEX01
idxwork bes1 C:¥pdrorg
sort bes1 C:¥tmp¥sortwork
```

INDEX01：インデクス識別子

bes1：インデクス情報ファイルを作成するサーバ名（HiRDB/パラレルサーバの場合に指定します）

C:¥pdrorg：インデクス情報ファイルを作成するディレクトリ名

bes1：ソート用ワークファイルを作成するサーバ名（HiRDB/パラレルサーバの場合に指定します）

C:¥tmp¥sortwork：ソート用ワークディレクトリ名

(4) インデクス格納 RD エリア及び表格納 RD エリアのバックアップを取得します

インデクス格納 RD エリア及び表格納 RD エリア（RDAREA1 及び RDAREA2）のバックアップを取得します。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5) pdrels コマンドでインデクス格納 RD エリア及び表格納 RD エリアの閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.2　データを格納している表にインデクスを定義する方法

実行者　表の所有者

データを格納している表に対する処理を変更する場合や，この表に対する検索性能の向上を目的とする場合などは，表にインデクスを追加できます。

注意事項

抽象データ型（プラグインが提供する抽象データ型を除く）の列にインデクス（B-tree インデクス）は定義できません。

14.2.1　追加方法

データを格納している表にインデクスを定義する手順を次に示します。

〈手順〉

1. CREATE INDEX でインデクスを定義します。インデクスを定義するとき，インデクスを定義する表をアクセスする UAP は排他待ちの状態になります。
2. インデクスを定義した後に，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。ただし，プラグインインデクスは，最適化情報収集ユティリティの対象外となります。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
3. インデクスを定義すると，インデクスを定義した表を使用しているストアドルーチンのインデクス情報が無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。

   また，トリガ SQL 文中に指定している表にインデクスを定義すると，トリガのインデクス情報が無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。

14.2.2　インデクスの作成時間を短縮する方法（EMPTY オプション）

データが大量にある表に対してインデクスを定義する場合，インデクスの実体の作成（CREATE INDEX の実行）に時間が掛かります。その間，ほかの定義系 SQL は実行できません。

CREATE INDEX に **EMPTY** オプションを指定すると，インデクスの実体を作成しないで，定義上のインデクスを作成します。これを**未完状態のインデクス**と呼びます。インデクスの実体を作成しないため，CREATE INDEX の実行は即時終了し，ほかの定義系 SQL を実行できるようになります。

なお，プラグインインデクスに対しても EMPTY オプションを指定できます。

参考

1. インデクスの実体が未作成であるため，未完状態のインデクスを使った検索や，未完状態のインデクスを定義している表の列の更新はできません(SQL エラーとなります)。
2. インデクスが未完状態であるかどうかは，データベース状態解析ユティリティ(pddbst コマンド)で確認できます。インデクス単位の状態解析，又は RD エリア単位の状態解析(論理的解析)では，status に状態を表示し，クラスタキー及びクラスタリングデータページの格納状態解析では，警告メッセージで未完状態であることを通知します。
3. インデクスの実体は，データベース再編成ユティリティ(pdrorg コマンド)のインデクス再作成機能(-k ixrc)を使用して作成します。インデクスの実体を作成すると，実体を作成したインデクスの未完状態は解除されます。また，PURGE TABLE 文で表を全件削除した場合，その表のすべてのインデクスの未完状態が解除されます。
4. 分割表の場合は，インデクスも複数の RD エリアに分割格納しますが，未完状態は，個々の分割格納されたインデクスごとに管理します。データベース再編成ユティリティ(pdrorg コマンド)では，このインデクスを格納する RD エリア単位にインデクスの実体を作成できます。このため，分割格納したインデクスの一部を作成した時点では，インデクスの一部は作成済みで一部は未完状態となるため，SQL に指定した条件などによって，実行できたりエラーになったりする場合があります。

14.3　インデクス名を変更する方法

実行者　表の所有者

ALTER INDEX でインデクス名を変更できます。なお，未完状態のインデクスもインデクス名を変更できます。名称を変更できるインデクスを次に示します。

- CREATE INDEX で定義した B-tree インデクス
- インデクス型を指定したインデクス
- 部分構造インデクス

インデクス名を変更する手順を次に示します。

〈手順〉

1. ALTER INDEX の RENAME オペランドに変更後のインデクス名を指定して実行します。インデクス名の変更中，変更対象のインデクスを定義した表をアクセスする UAP は排他待ち状態になります。
2. インデクス名を変更すると，変更したインデクスを使用しているストアドプロシジャが無効になります。また，名称を変更したインデクスが定義してある表を使用しているストアドプロシジャのインデクス情報が無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドプロシジャを再作成してください。

   また，トリガで使用しているインデクスの名称を変更すると，トリガが無効になります。トリガ SQL 文中に指定している表のインデクス名を変更すると，トリガのインデクス情報が無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
3. インデクス名を変更したインデクスにインデクス専用のグローバルバッファを割り当てている場合，そのグローバルバッファに指定しているインデクス名も変更してください。変更方法を次に示します。

グローバルバッファの変更方法

次に示すどちらかの方法でグローバルバッファを変更してください。

- HiRDB を正常終了して pdbuffer オペランドの指定を変更してください（インデクス名を変更したインデクスに対応する pdbuffer オペランドの指定を変更してください）。
- システム構成変更コマンド（pdchgconf コマンド）で pdbuffer オペランドの指定を変更してください（インデクス名を変更したインデクスに対応する pdbuffer オペランドの指定を変更してください）。この場合，HiRDB を正常終了する必要はありません。ただし，この方法でグローバルバッファを変更する場合は HiRDB Advanced High Availability が必要になります。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

14.4　インデクスを削除する方法

実行者　HiRDB 管理者及び表の所有者（又は DBA 権限保持者）

データを格納している表に対する処理を変更すると，表の探索条件に合わない不要なインデクスが発生します。不要なインデクスをそのまま残しておくと，行や列の追加又は更新処理のときに，このインデクスに対する余分な処理時間が必要になります。このため，不要になったインデクスは直ちに削除するようにします。

未使用インデクスは，HiRDB SQL Tuning Advisor で調べることができます。未使用インデクスの調査方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

インデクスを削除する手順を次に示します。

〈手順〉

1. DROP INDEX でインデクスを削除します。インデクスの削除中，削除対象のインデクスを定義した表をアクセスする UAP は排他待ち状態になります。
2. インデクスを削除した後に，必要があれば最適化情報収集ユティリティ（pdgetcst コマンド）を実行してください。ただし，プラグインインデクスは最適化情報収集ユティリティの対象外となります。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
3. インデクスを削除すると，削除したインデクスを使用しているストアドルーチンが無効になり，インデクスを削除した表を使用しているストアドルーチンのインデクス情報が無効になります。この場合，ALTER PROCEDURE 又は ALTER ROUTINE でストアドルーチンを再作成してください。

   また，トリガで使用しているインデクスを削除すると，トリガが無効になります。トリガ SQL 文中に指定している表のインデクスを削除すると，トリガのインデクス情報が無効になります。この場合，ALTER TRIGGER 又は ALTER ROUTINE でトリガを再作成してください。
4. 削除したインデクスにインデクス専用のグローバルバッファを割り当てている場合，そのグローバルバッファを削除してください。削除方法を次に示します。

グローバルバッファの削除方法

次に示すどれかの方法でグローバルバッファを削除してください。

- HiRDB を正常終了して pdbuffer オペランドの指定を変更してください（削除したインデクスに対応する pdbuffer オペランドを削除してください）。
- システム構成変更コマンド（pdchgconf コマンド）で pdbuffer オペランドの指定を変更してください（削除したインデクスに対応する pdbuffer オペランドを削除してください）。この場合，HiRDB を正常終了する必要はありません。ただし，この方法でグローバルバッファを削除する場合は HiRDB Advanced High Availability が必要になります。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

- pdbufmod コマンドでグローバルバッファを削除してください。pdbufmod コマンドでグローバルバッファを削除する方法については，「HiRDB の稼働中にグローバルバッファを追加，変更，又は削除する方法（グローバルバッファの動的変更）」を参照してください。ただし，この方法でグローバルバッファを削除する場合は次に示す条件をすべて満たす必要があります。

  ・HiRDB Advanced High Availability を導入している

  ・pd_dbbuff_modify オペランドに Y を指定している

  なお，この方法で削除したグローバルバッファの情報は HiRDB を正常終了又は計画停止すると無効になるため，HiRDB を終了したときに pdbuffer オペランドの指定を変更してください（削除したインデクスに対応する pdbuffer オペランドを削除してください）。pdbuffer オペランドの指定を変更しないと，このグローバルバッファはどのインデクスにも割り当てられないため，HiRDB を開始したときにエラーになります。

14.5　既存のインデクスの定義系 SQL を作成する方法

実行者　HiRDB 管理者又は DBA 権限保持者

pddefrev コマンドで，現在使用しているインデクスの定義系 SQL を作成できます。

現在使用しているインデクスと同じ定義のインデクスを作成するときに使用すると便利です。pddefrev コマンドで作成された定義系 SQL は，データベース定義ユティリティ（pddef コマンド）の入力情報となります。

> **参考**
>
> プラグインインデクスは，pddefrev コマンドの対象外となります（定義系 SQL を作成できません）。

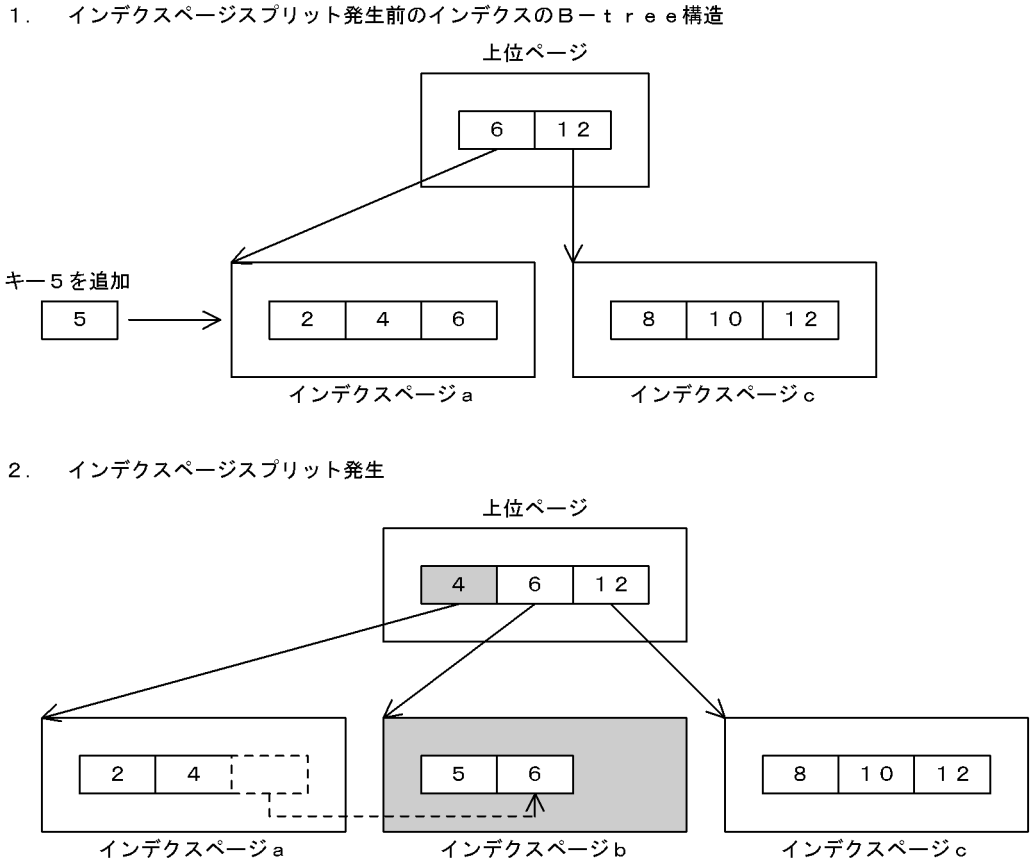
14.6　インデクスページスプリットの発生回数を削減する方法（アンバランスインデクススプリット）

実行者　表の所有者

14.6.1　インデクスページスプリットとは

HiRDB のインデクスは B-tree 構造をしています。このため，インデクスページにキーを追加しようとした場合，追加するインデクスページに空き領域がないとき，インデクスページスプリットが発生します。**インデクスページスプリット**とは，キーを追加するインデクスページに空き領域がないとき，HiRDB が空き領域を確保するために，このインデクスページのインデクス情報を均等に２分割して，後ろの半分を新しいページに移すことです。通常のインデクスページスプリットの例を次の図に示します。

図 14–2　インデクスページスプリットの例



(1)　インデクスページスプリットの発生に伴うログ量の増加

表にインデクスがある場合，表に対してデータを追加すると，これに伴ってインデクスページにキーが追加されます。このとき，インデクスページ内の未使用領域の比率が小さいと，インデクスページスプリットが頻繁に発生することがあります。インデクスページスプリットが頻繁に発生すると，インデクスの構造の変更に伴うログ量が増加します。

(2)　インデクスページスプリット発生回数の削減方法

インデクスページスプリットの発生回数を削減するには，統計解析ユティリティ（pdstedit コマンド）で取得したインデクスページスプリットの情報を基に，インデクスページ内の未使用領域の比率を変更します。インデクスページ内の未使用領域の比率を変更するには，定義系 SQL の **CREATE INDEX** の **PCTFREE** オペランドの値を変更して，インデクスを再度定義します。

統計解析ユティリティで取得したインデクスページスプリットの情報への対処の仕方については，「インデクスのチューニング」を参照してください。

14.6.2　アンバランスインデクススプリットとは

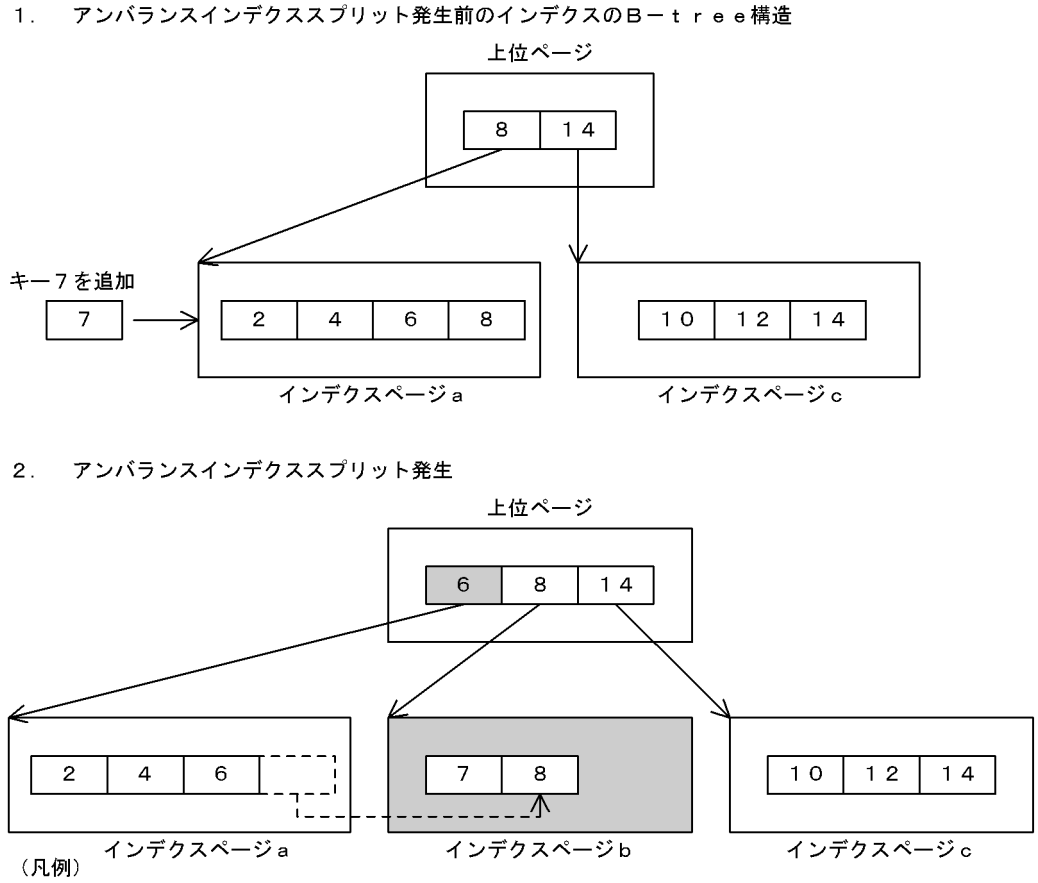
通常のインデクスページスプリットでは，連続したキー値を追加すると，インデクスページのデータ格納効率が下がります。このような場合，アンバランスインデクススプリットを使用します。**アンバランスインデクススプリット**とは，インデクスページのインデクス情報を均等に２分割しないで，不均等に分割することです。これによって，連続したキー値を追加したときに，データの格納効率を向上できます。

アンバランスインデクススプリットが指定された場合，インデクスページのインデクス情報を分割する比率は，キー値の追加位置で決定します。

追加位置がインデクスページから見て前半部分であれば，以降，前半部分にキーが追加されると予測します。このため，追加するキー値より一つ大きいキー値を分割位置として，前半部分を左側ページに格納します。

追加位置が後半部分であれば，以降，後半部分にキーが追加されると予測します。このため，追加位置を分割位置として，後半部分を右側ページに格納します。インデクスページの後半部分にキーを追加した場合のアンバランスインデクススプリットの例を次の図に示します。

図 14–3　アンバランスインデクススプリットの例



(1)　アンバランスインデクススプリットの適用範囲

アンバランスインデクススプリットは，最終ページでないリーフページだけに適用します。また，ロールバック時は，アンバランスインデクススプリットを適用しないで，通常のインデクスページスプリットを実施します。この場合，50:50 の割合でページ分割します。

また，リーフページの最終ページは，定義系 SQL の **CREATE INDEX** の **PCTFREE** オペランドで指定した値に従ってデータを分割します。これは，データロードなどによるキー値の昇順追加に対応するためです。

(2)　アンバランスインデクススプリットの適用基準

次に示す条件を満たすインデクスに，アンバランスインデクススプリットを適用すると効果があります。

- キーの重複度に偏りがないインデクス
- キー長がほぼ均一であるインデクス
- 連続した中間キー値を頻繁に追加するインデクス

(3) 指定方法

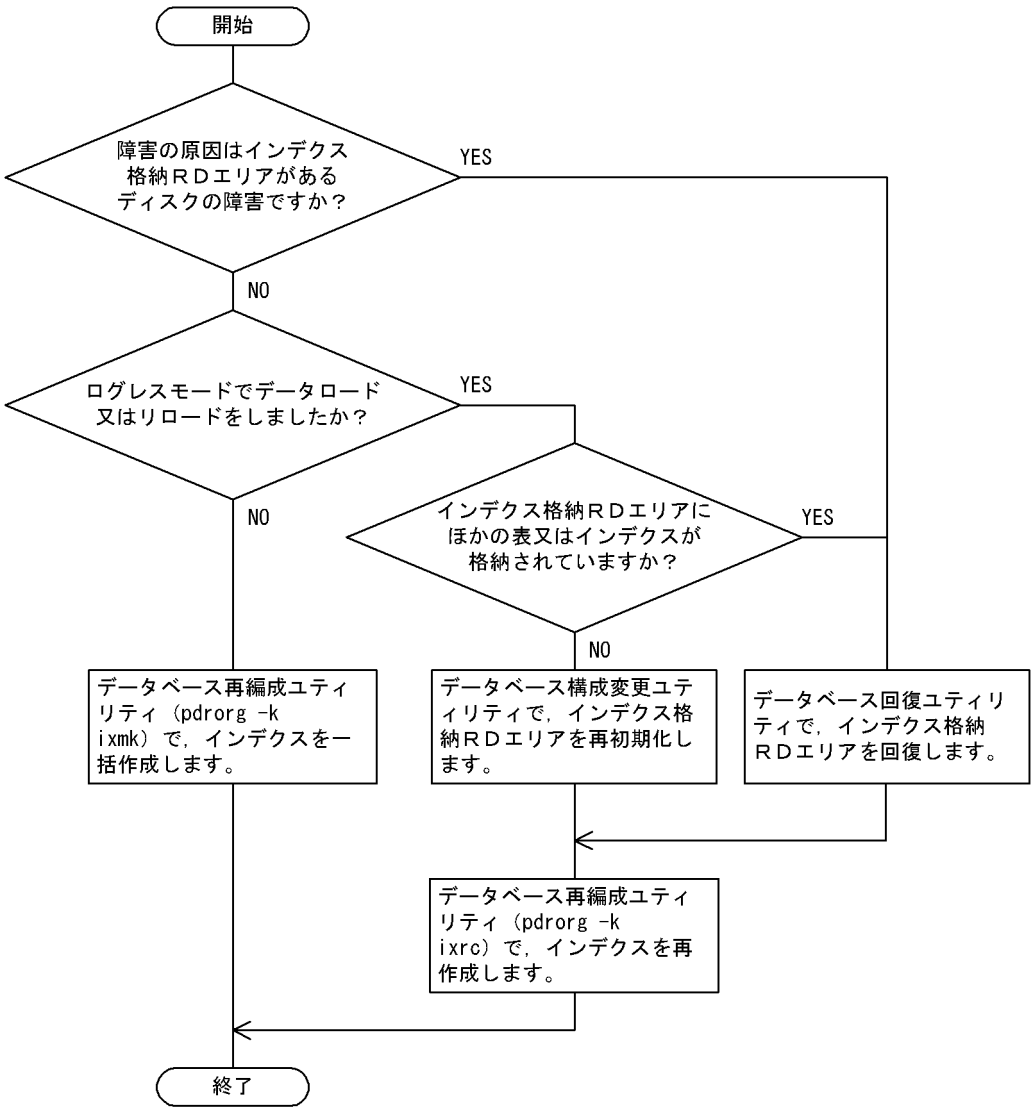
アンバランスインデクススプリットを使用する場合，定義系 SQL の **CREATE INDEX**，**CREATE TABLE**，又は **ALTER TABLE** のインデクスオプションに **UNBALANCED SPLIT** を指定します。

14.7　インデクスの一括作成処理中にエラーが発生したときの対処方法

実行者　HiRDB 管理者

データロード又はリロード時（データベース作成ユティリティ又はデータベース再編成ユティリティ実行時）に，インデクスを一括作成する指定をすると，データロード又はリロード処理が終了した後にインデクスの一括作成処理が実行されます。このインデクスの一括作成処理中にエラーが発生した場合（KFPL00716-I メッセージが出力されます），表データは完成状態になっていますが，インデクスデータは未完成状態のままです。このときのインデクスデータの回復方法の流れを次の図に示します。

図 14–4　インデクスデータの回復方法の流れ

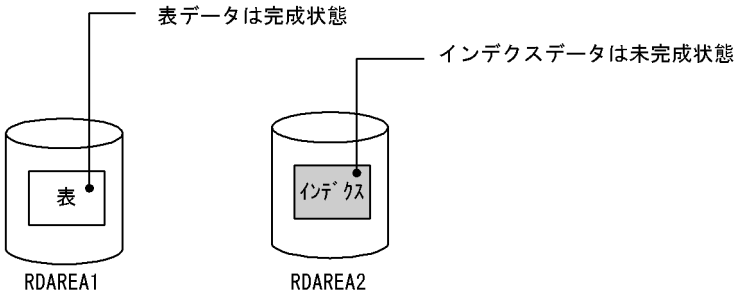


14.7.1　リロード（データロード）をログ取得モード又は更新前ログ取得モードで実行した場合の回復例

例題

- インデクスが定義されている表のリロードをしていました。このとき，インデクス格納 RD エリアの容量不足のため（又はソート処理エラーのため），インデクスの一括作成処理中にエラーが発生しました。なお，ソート処理エラーが発生した場合は，KFPL15062-E メッセージが出力されます。
- プラグインインデクスが定義されている表に，追加モードでデータロードをしていました。このとき，インデクス格納 RD エリアの容量不足のため，プラグインインデクスの一括作成処理中にエラーが発生しました。

このとき，データは次に示すようになっています。



(1)　pdmod コマンドでインデクス格納 RD エリアの容量を拡張します

RD エリアの容量不足でエラーとなった場合は，pdmod コマンドでインデクス格納 RD エリア（RDAREA2）の容量を拡張してください。RD エリアの容量拡張方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

なお，ソート処理エラーの場合は，この操作は必ずしも必要ありません。出力されたエラーメッセージを参照して対策してください。

(2)　pdhold コマンドでインデクス格納 RD エリアを閉塞します

```
pdhold -r RDAREA2
```

既に閉塞している場合はこの操作は必要ありません。

(3)　pdrorg コマンドでインデクスを一括作成します

```
pdrorg -k ixmk -t TABLE1 C:¥pdrorg¥rorg01
```

リロード（データロード）時に作成されたインデクス情報ファイルを入力情報にしてインデクスを一括作成してください。

(4)　インデクス格納 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でインデクスを一括作成したため，インデクス格納 RD エリア（RDAREA2）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA2
```

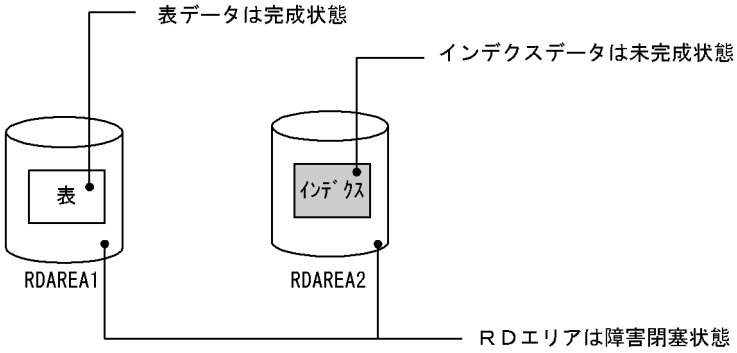
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.7.2　リロード（データロード）をログレスモードで実行した場合の回復例（インデクス格納 RD エリアにほかの表又はインデクスがない場合）

例題

- インデクスが定義されている表のリロードをしていました。このとき，インデクス格納 RD エリアの容量不足のため（又はソート処理エラーのため），インデクスの一括作成処理中にエラーが発生しました。なお，ソート処理エラーが発生した場合は，KFPL15062-E メッセージが出力されます。
- プラグインインデクスが定義されている表に，追加モードでデータロードをしていました。このとき，インデクス格納 RD エリアの容量不足のため，プラグインインデクスの一括作成処理中にエラーが発生しました。

このとき，データは次に示すようになっています。



(1)　メッセージを確認します

データロードの場合は KFPL00703-I メッセージが出力されているかを確認してください。リロードの場合は KFPL00714-I 又は KFPL00734-I メッセージが出力されているかを確認してください。出力されていれば表データは完成状態になっているため，pdrels コマンドで表データ格納 RD エリア（RDAREA1）の閉塞状態を解除してください。

```
pdrels -r RDAREA1
```

(2)　pdmod コマンドでインデクス格納 RD エリアを再初期化します

pdmod コマンドでインデクス格納 RD エリア（RDAREA2）を再初期化してください。RD エリアの再初期化方法については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください。

(3)　pdmod コマンドでインデクス格納 RD エリアの容量を拡張します

RD エリアの容量不足でエラーとなった場合は，pdmod コマンドでインデクス格納 RD エリア（RDAREA2）の容量を拡張してください。RD エリアの容量拡張方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

なお，ソート処理エラーの場合は，この操作は必ずしも必要ありません。出力されたエラーメッセージを参照して対策してください。

(4)　pdhold コマンドでインデクス格納 RD エリアを閉塞します

```
pdhold -r RDAREA2
```

既に閉塞している場合はこの操作は必要ありません。

(5)　pdrorg コマンドでインデクスを再作成します

```
pdrorg -k ixrc -t TABLE1 C:¥pdrorg¥rorg02
```

(6)　インデクス格納 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でインデクスを再作成したため，インデクス格納 RD エリア（RDAREA2）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(7)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA2
```

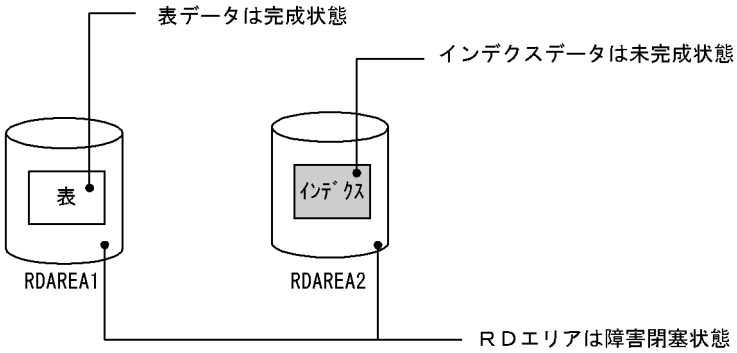
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.7.3　リロード（データロード）をログレスモードで実行した場合の回復例（インデクス格納 RD エリアにほかの表又はインデクスがある場合）

例題

- インデクスが定義されている表のリロードをしていました。このとき，インデクス格納 RD エリアの容量不足のため（又はソート処理エラーのため），インデクスの一括作成処理中にエラーが発生しました。なお，ソート処理エラーが発生した場合は，KFPL15062-E メッセージが出力されます。
- プラグインインデクスが定義されている表に，追加モードでデータロードをしていました。このとき，インデクス格納 RD エリアの容量不足のため，プラグインインデクスの一括作成処理中にエラーが発生しました。

このとき，データは次に示すようになっています。



(1)　メッセージを確認します

データロードの場合は KFPL00703-I メッセージが出力されているかを確認してください。リロードの場合は KFPL00714-I 又は KFPL00734-I メッセージが出力されているかを確認してください。出力されていれば表データは完成状態になっているため，pdrels コマンドで表データ格納 RD エリア（RDAREA1）の閉塞状態を解除してください。

```
pdrels -r RDAREA1
```

(2)　pdrstr コマンドでインデクス格納 RD エリアを回復します

インデクス格納 RD エリア（RDAREA2）を回復してください。RD エリアの回復方法については，「データベースの回復方法」を参照してください。

(3)　pdmod コマンドでインデクス格納 RD エリアの容量を拡張します

RD エリアの容量不足でエラーとなった場合は，pdmod コマンドでインデクス格納 RD エリア（RDAREA2）の容量を拡張してください。RD エリアの容量拡張方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

なお，ソート処理エラーの場合は，この操作は必ずしも必要ありません。出力されたエラーメッセージを参照して対策してください。

(4) pdhold コマンドでインデクス格納 RD エリアを閉塞します

```
pdhold -r RDAREA2
```

既に閉塞している場合はこの操作は必要ありません。

(5) pdrorg コマンドでインデクスを再作成します

```
pdrorg -k ixrc -t TABLE1 C:¥pdrorg¥rorg03
```

(6) インデクス格納 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でインデクスを再作成したため，インデクス格納 RD エリア（RDAREA2）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(7) pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA2
```

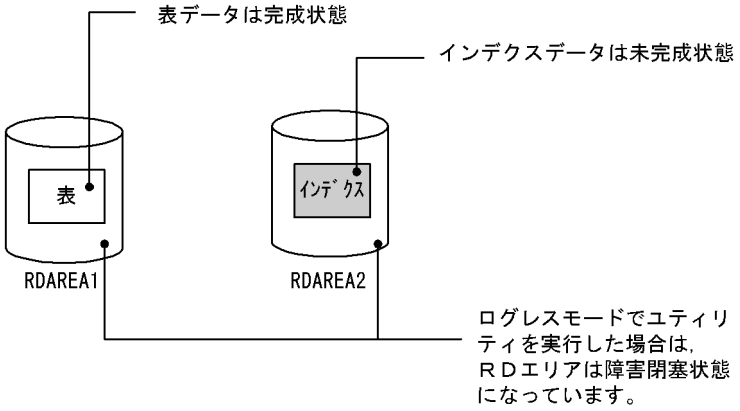
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.7.4 インデクス格納 RD エリアがあるディスクに障害が発生したときの回復例

例題

- インデクスが定義されている表のリロードをしていました。このとき，インデクス格納 RD エリアがあるディスクに障害が発生したため，インデクスの一括作成処理中にエラーが発生しました。
- プラグインインデクスが定義されている表に，追加モードでデータロードをしていました。このとき，インデクス格納 RD エリアがあるディスクに障害が発生したため，プラグインインデクスの一括作成処理中にエラーが発生しました。

このとき，データは次に示すようになっています。



(1)　メッセージを確認します

データロードの場合は KFPL00703-I メッセージが出力されているかを確認してください。リロードの場合は KFPL00714-I 又は KFPL00734-I メッセージが出力されているかを確認してください。出力されていれば表データは完成状態になっているため，pdrels コマンドで表データ格納 RD エリア（RDAREA1）の閉塞状態を解除してください。この操作は，ログレスモードでリロード（データロード）をしたときだけ必要になります。

```
pdrels -r RDAREA1
```

(2)　ディスク障害を対策します

ディスク障害の対策を実施してください。

(3)　pdhold コマンドでインデクス格納 RD エリアを閉塞します

```
pdhold -r RDAREA2
```

この操作は，ログ取得モード又は更新前ログ取得モードでリロード（データロード）をしたときだけ必要になります。

(4)　pdrstr コマンドでインデクス格納 RD エリアを回復します

インデクス格納 RD エリア（RDAREA2）を回復してください。RD エリアの回復方法については，「データベースの回復方法」を参照してください。

なお，RD エリアを回復した後も RD エリアの閉塞を解除しないでください。

(5)　pdrorg コマンドでインデクスを再作成します

```
pdrorg -k ixrc -t TABLE1 C:¥pdrorg¥rorg05
```

(6)　インデクス格納 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でインデクスを再作成したため，インデクス格納 RD エリア（RDAREA2）のバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(7)　pdrels コマンドで RD エリアの閉塞を解除します

```
pdrels -r RDAREA2
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

14.8　プラグインインデクスの遅延一括作成

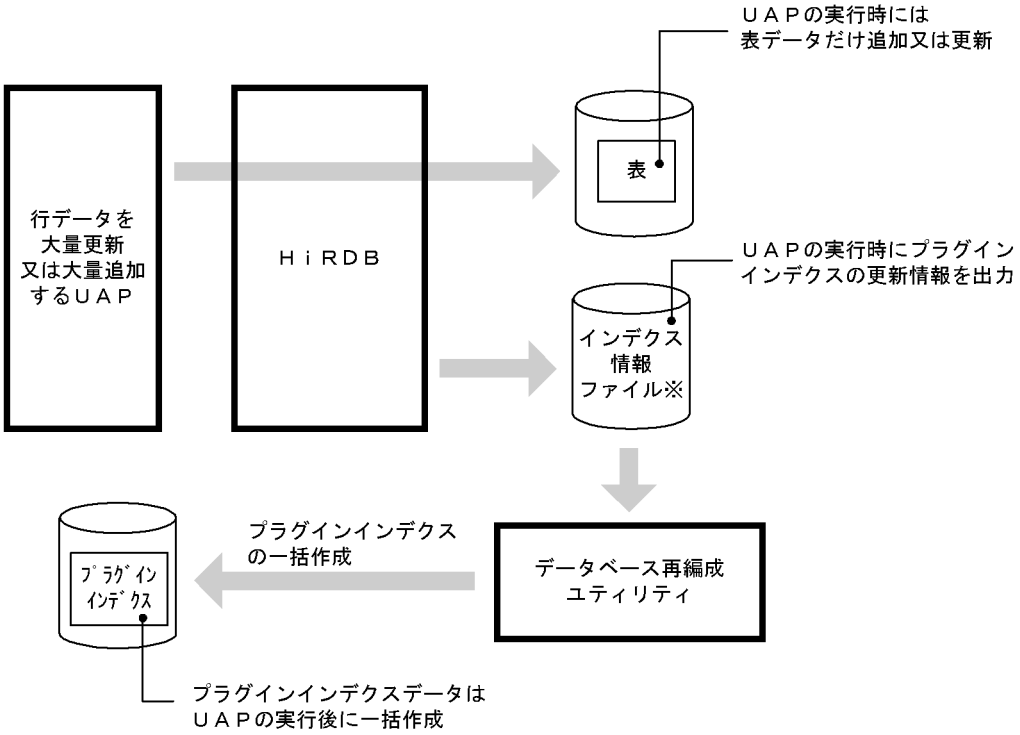
実行者　HiRDB 管理者

14.8.1　プラグインインデクスの遅延一括作成とは

プラグインインデクスを定義した表の行データを大量追加（又は大量更新）すると，プラグインインデクスへのキー追加処理が行われます。このため，大量の行データを追加すると，プラグインインデクスへのキー追加処理が性能を悪くする原因になることがあります。

行データを追加したときに，プラグインインデクスのデータ追加処理をしないで，データベース再編成ユティリティを使用して後で一括してプラグインインデクスのデータ追加処理ができます。これを**プラグインインデクスの遅延一括作成**といいます。プラグインインデクスの遅延一括作成を次の図に示します。

図 14–5　プラグインインデクスの遅延一括作成



注※

系切り替え機能を使用する場合は，インデクス情報ファイルを共有ディスク上に作成してください。

(1)　前提条件

使用しているプラグインが，プラグインインデクスの遅延一括作成をサポートしているかどうかを確認してください。この機能をサポートしていないプラグインに対しては，この機能を使用できません。

このマニュアルの発行時点でこの機能をサポートしているプラグインは次のとおりです。

- HiRDB Text Search Plug-in

(2) 利点

プラグインインデクスのデータ追加処理をしない分だけ，大量の行データを追加又は更新する UAP の実行時間（表データの作成時間）を短縮できます。

(3) 機能の適用範囲

この機能が適用されるのは，行の追加及び更新時だけです。すなわち，次に示す SQL に対して適用されます。

- INSERT 文
- UPDATE 文

行の削除（DELETE 文）には適用されません。

(4) 適用基準

- プラグインインデクスを定義した表に対して，大量追加又は大量更新をするバッチ処理に適用します。
- HiRDB Text Search Plug-in を使用する場合，プラグインインデクスに大量追加又は大量更新をするバッチ処理にはこの機能が有効です。しかし，オンライン処理でプラグインインデクスの更新を行い，その内容を参照する場合は，HiRDB Text Search Plug-in の差分インデクス機能が有効です。

(5) 制限事項

プラグインインデクスの遅延一括作成をするときは，次に示す制限事項があります。

(a) プラグインインデクスを一括作成するまで，追加又は更新した行に対してプラグインインデクスを使用した検索をしないでください

UAP の実行後からプラグインインデクスを一括作成するまでの間は，表データとプラグインインデクスデータが不整合の状態になっています。※

このときに，追加又は更新した行に対してプラグインインデクスを使用した検索を実施すると，不正な処理結果を返すことがあります。したがって，プラグインインデクスを一括作成するまでは，追加又は更新した行に対してプラグインインデクスを使用した検索をしないでください。ただし，プラグインインデクスを使用しなければ，正しい処理結果が返ります。

なお，追加又は更新した行以外の行に対しては，プラグインインデクスを使用した検索ができます。

注※

データを追加した場合，プラグインインデクスデータは UAP 実行前の状態になっています。データを更新した場合，更新前のプラグインインデクスデータは削除された状態になっています。

このように，追加又は更新した行については，表データとプラグインインデクスデータが不整合の状態になっています。

(b) プラグインインデクスを一括作成するまで，通常の UAP ではプラグインインデクスを更新できません

プラグインインデクスを一括作成するまで，通常の UAP（プラグインインデクスの遅延一括作成を指定していない UAP）ではプラグインインデクスを更新できません。ただし，PURGE TABLE 文は実行できます。

(c) 同一表に対して UAP は同時に実行できません

インデクス情報ファイルには，排他モード（EX）で排他が掛かります。このため，同一表に対してこの機能を使用した UAP は同時に実行できません。

(d) プラグインインデクスを一括作成するまで追加データロードはできません

プラグインインデクスを一括作成するまで追加データロードはできません。ただし，表を横分割している場合は，遅延一括作成機能を使用してデータを格納していない RD エリアに対して，UAP による処理と追加データロードができます。

(e) プラグインインデクスを一括作成するまでプラグインインデクス名を変更しないでください

プラグインインデクスを一括作成する前に，ALTER INDEX でプラグインインデクス名を変更すると，UAP 実行時に出力したインデクス情報ファイルではプラグインインデクスの一括作成ができなくなります。一括作成前にプラグインインデクス名を変更した場合，プラグインインデクス名を元に戻してプラグインインデクスを一括作成した後で，再度プラグインインデクス名を変更してください。

(f) プラグインインデクスを一括作成するまでプラグインインデクスが格納されている RD エリア名を変更しないでください

プラグインインデクスを一括作成する前に，データベース構成変更ユティリティ（pdmod）でプラグインインデクスが格納されている RD エリア名を変更すると，プラグインインデクスの一括作成ができなくなります。一括作成前に RD エリア名を変更した場合，RD エリア名を元に戻してプラグインインデクスを一括作成した後で，再度 RD エリア名を変更してください。

14.8.2 環境設定方法

インデクス情報ファイルを Windows のディレクトリ下に作成するか，又は HiRDB ファイルシステム領域に作成するかによって環境設定方法が異なります。通常は Windows のディレクトリ下にインデクス情報ファイルを作成してください。

(1)　インデクス情報ファイルを Windows のディレクトリ下に作成する場合（通常の方法）

(a)　インデクス情報ファイルを作成するディレクトリを作成します

インデクス情報ファイルを作成するディレクトリを作成してください。このディレクトリ下にインデクス情報ファイルが作成されます。

ポイント

- インデクス情報ファイルは大容量になります。ディスク容量に余裕があるディスクにディレクトリを作成してください。ファイル容量については，「インデクス情報ファイルについて」を参照してください。
- ファイル名の重複を避けるため，このディレクトリ下にはインデクス情報ファイル以外のファイルを置かないようにしてください。
- ここで指定したディレクトリ下のファイルの作成及び削除は，HiRDB が行います。したがって，このディレクトリには HiRDB 管理者の書き込み権限が必要になります。

(b)　pd_plugin_ixmk_dir オペランドを指定します

HiRDB システム定義を修正します。いったん HiRDB を正常終了してください。

HiRDB/シングルサーバの場合

シングルサーバ定義の pd_plugin_ixmk_dir オペランドに，(a)で作成したディレクトリ名称を指定します。

HiRDB/パラレルサーバの場合

処理対象の表があるバックエンドサーバのバックエンドサーバ定義に，pd_plugin_ixmk_dir オペランドを指定してください。(a)で作成したディレクトリ名称を指定してください。表がサーバ間横分割されている場合は，各バックエンドサーバのバックエンドサーバ定義に pd_plugin_ixmk_dir オペランドを指定してください。

(c)　クライアント環境定義に PDPLGIXMK ＝ YES を指定します

クライアント環境定義に PDPLGIXMK ＝ YES を指定してください。

PDPLGIXMK オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(2)　インデクス情報ファイルを HiRDB ファイルシステム領域に作成する場合

(a)　インデクス情報ファイル用の HiRDB ファイルシステム領域を作成します

```
pdfmkfs -n 50 -l 256 -k UTL -e 60000 D:¥hd001¥ixdir
```

〔説明〕

- -l オプションには，この機能を使用して更新するプラグインインデクスを格納する RD エリア数（サーバ内の RD エリア数）を指定してください。
- -k オプションには UTL を指定してください。
- -e オプションには 60000 を指定してください。
- ここで作成する HiRDB ファイルシステム領域は，プラグインインデクスの遅延一括作成で使用するインデクス情報ファイル専用の領域としてください。
- 共有ディスクに HiRDB ファイルシステム領域を作成してください。
- インデクス情報ファイルは大容量になります。ディスク容量に余裕があるディスクにディレクトリを作成してください。ファイル容量については，「インデクス情報ファイルについて」を参照してください。
- ファイル名の重複を避けるため，この HiRDB ファイルシステム領域下にはインデクス情報ファイル以外のファイルを置かないようにしてください。
- ここで指定した HiRDB ファイルシステム領域下のファイルの作成及び削除は，HiRDB が行います。したがって，この HiRDB ファイルシステム領域には HiRDB 管理者の書き込み権限が必要になります。

(b)　pd_plugin_ixmk_dir オペランドを指定します

HiRDB システム定義を修正します。いったん HiRDB を正常終了してください。

HiRDB/シングルサーバの場合

シングルサーバ定義の pd_plugin_ixmk_dir オペランドに，(a)で作成した HiRDB ファイルシステム領域名を絶対パス名で指定してください。

HiRDB/パラレルサーバの場合

処理対象の表があるバックエンドサーバのバックエンドサーバ定義に，pd_plugin_ixmk_dir オペランドを指定してください。(a)で作成した HiRDB ファイルシステム領域名を絶対パス名で指定してください。表がサーバ間横分割されている場合は，各バックエンドサーバのバックエンドサーバ定義に pd_plugin_ixmk_dir オペランドを指定してください。

(c)　クライアント環境定義にオペランドを追加します

クライアント環境定義に次に示すオペランドを指定します。これらのオペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

オペランド名	説明
PDPLGIXMK	プラグインインデクスの遅延一括作成をするかどうかを指定します。YES を指定します。
PDPLGPFSZ	遅延一括作成用のインデクス情報ファイルの初期容量を指定します。
PDPLGPFSZEXP	遅延一括作成用のインデクス情報ファイルの増分値を指定します。ファイル容量が満杯になった場合，ここで指定した値の分だけファイル容量を増分します。

●プラグインインデクスの更新可能件数の目安

PDPLGPFSZ 及び PDPLGPFSZEXP オペランドを省略すると，省略値としてそれぞれ 8192 キロバイトが仮定されます。このときのプラグインインデクスの更新可能件数は次のとおりとなります。

使用しているプラグイン	増分なしの場合	増分ありの場合
HiRDB Text Search Plug-in	215,000 件	5,160,000 件

●PDPLGPFSZ 及び PDPLGPFSZEXP オペランドの指定例

複数のプラグインインデクスを更新する場合の上記オペランドの指定例を次に示します。

（例）

- プラグインインデクス 1 のファイル容量：10240 キロバイト
- プラグインインデクス 2 のファイル容量：81920 キロバイト

この場合の PDPLGPFSZ 及び PDPLGPFSZEXP オペランドの指定例を次に示します。

```
PDPLGPFSZ=10240
PDPLGPFSZEXP=10240
```

このように指定すると，プラグインインデクス 2 のファイル容量は 7 回増分し，領域をむだに使用することがなくなります。なお，1 ファイルは 23 回まで増分できます。

14.8.3　UAP 実行時の手順

UAP を実行してから，プラグインインデクスを遅延一括作成するまでの手順を次に示します。

〈手順〉

1. UAP を実行します。

   UAP を実行すると，プラグインインデクスの情報がインデクス情報ファイルに出力されます。作成されたインデクス情報ファイル名称を **KFPH25100-I** メッセージで確認してください。

2. **pdhold** コマンドで，プラグインインデクス格納 RD エリアを閉塞してください。

   pdhold -r RDAREA1

3. **pdrorg** コマンドで，プラグインインデクスを一括作成してください。このとき，1 で作成したインデクス情報ファイルを入力情報にしてください。

   pdrorg -k ixmk -t TABLE1 C:¥pdrorg¥rorg06

4. **pdrels** コマンドで，プラグインインデクス格納 RD エリアの閉塞を解除してください。

   pdrels -r RDAREA1

14.8.4　注意事項

(1)　インデクス情報ファイルについて

(a)　作成されるファイル数

インデクス情報ファイルは，プラグインインデクスを格納している RD エリアごとに作成されます。したがって，作成されるインデクス情報ファイルは，UAP で更新したプラグインインデクスの格納 RD エリア数になります。

(b)　作成されるファイルの名称

インデクス情報ファイルに付けられる名称は，インデクス識別子及び RD エリア名によって決まります（次に示す条件に従って決まります）。このインデクス情報ファイル名をデータベース再編成ユティリティ（pdrorg コマンド）の制御情報ファイルに指定します。

条件	インデクス情報ファイル名
次に示す条件をすべて満たす場合 • インデクス識別子長＋ RD エリア名称長 ＜ 30 バイト • インデクス識別子及び RD エリア名が注で説明する文字列である	インデクス識別子．RD エリア名
次に示す条件をすべて満たす場合 • インデクス識別子長＋ RD エリア名称長 ≧ 30 バイト • インデクス識別子及び RD エリア名が注で説明する文字列である	インデクス ID．RD エリア ID.インデクス識別子の一部※
インデクス識別子及び RD エリア名に注で説明する文字列以外の文字列が含まれている場合	インデクス ID．RD エリア ID

注

Ａ～Ｚ，ａ～ｚ，０～９，ピリオド（.），下線（＿），及び@で構成される文字列

この規則に従ってインデクス識別子及び RD エリア名を付けることをお勧めします。

注※

インデクス識別子の先頭 12 バイトが表示されます。

(c)　作成されるファイルの容量

インデクス情報ファイルの容量は大きくなるため注意してください。１ファイルの容量の概算は，次に示す計算式から求められます。ただし，pd_plugin_ixmk_dir オペランドに HiRDB ファイルシステム領域名を指定する場合は，1 ファイルのサイズを 1,024 バイトの境界になるよう切り上げる必要があります。

（12＋Ｅ）×（Ａ＋Ｂ＋Ｃ×Ｄ）＋1024　　（単位：バイト）

Ａ：行データの追加件数（INSERT 文での更新件数）

Ｂ：行データの更新件数（UPDATE 文での更新件数）

C：プラグインインデクスの遅延一括作成を指定した UAP の数

D：UAP 中の COMMIT 発行回数

E：プラグインの種類によって異なります。HiRDB Text Search Plug-in の場合は 27 となります。

(d) ファイルの削除

インデクス情報ファイルは不要になった時点で，HiRDB が自動的に削除します。削除するタイミングは次のとおりです。

- プラグインインデクスの一括作成が正常終了したとき
- プラグインインデクスの再作成を実行したとき

上記の操作をしない場合は，ファイルは削除されません。自分でファイルを削除してください。

(2) バックアップについて

UAP の実行中からプラグインインデクスを一括作成するまでの間は，表データとプラグインインデクスデータが不整合の状態になっています。したがって，この間はバックアップを取得しないでください。

(3) インデクス情報ファイル間のデッドロック

同じ横分割表に対して，プラグインインデクスの遅延一括作成を指定した UAP を同時に実行すると，インデクス情報ファイルの排他処理でデッドロックが発生することがあります。

(4) ストアドプロシジャ及びストアドファンクションについて

HiRDB Version 5.0 05-02 より前のバージョンで作成したストアドプロシジャ及びストアドファンクションについては，プラグインインデクスの遅延一括作成ができません。PDPLGIXMK ＝ YES を指定してプラグインインデクスの遅延一括作成をすると，KFPA11537-E メッセージが出力されてエラーになります。この場合，次に示すどちらかの処置をとってください。

- **プラグインインデクスの遅延一括作成をする場合**
  プラグインインデクスを更新するストアドプロシジャ及びストアドファンクションに対して，ALTER ROUTINE 又は ALTER PROCEDURE を実行してください。
- **プラグインインデクスの遅延一括作成をしない場合**
  PDPLGIXMK=NO を指定して UAP を実行してください。

14.8.5　障害発生時の対処方法

(1)　インデクス情報ファイルに障害が発生した場合

UAP の実行後からプラグインインデクスを一括作成するまでの間に，インデクス情報ファイルに障害が発生した場合は，プラグインインデクスを遅延一括作成できません。この場合は，次に示すどちらかの方法でプラグインインデクスを作成してください。

- データベース再編成ユティリティでプラグインインデクスを再作成してください。
- DROP INDEX を実行し，その後 CREATE INDEX を実行してプラグインインデクスを作成してください。

上記のどちらの方法も，「UAP で追加したデータ件数＋表に元からあるデータ件数」分のプラグインインデクスデータを作成するため，データ件数が大量の場合は大変時間が掛かります。

したがって，インデクス情報ファイルの容量不足などにならないように，ディスク容量に十分余裕があるディレクトリを準備してください。

(2)　UAP の実行時に障害が発生した場合

ロールバックされた場合

障害原因を対策した後に，UAP を再度実行してください。

ロールバックされない場合

ログレスモードで UAP を実行した場合などは，ロールバックされません。この場合は，データベース回復ユティリティでデータベースを回復した後に，UAP を再度実行してください。

(3)　プラグインインデクスの一括作成時に障害が発生した場合

障害原因を対策した後に，データベース再編成ユティリティを再度実行して，プラグインインデクスを一括作成してください。

15 RD エリアの運用

この章では，RD エリアの運用方法と使用中空きページ及び使用中空きセグメントを再利用する方法について説明します。

15.1　RD エリアの容量が不足してくると

15.1.1　RD エリアの容量が不足したときの対処

実行者　HiRDB 管理者

表に対してデータの追加や削除を繰り返すと，最初に見積もった RD エリアの容量では対応できなくなる場合があります。HiRDB は，データの検索効率や格納効率が低下した表に対して，次に示すメッセージを出力します。

- KFPA12300-I
- KFPH00211-I
- KFPH00212-I

(1)　次に示す場合は RD エリアの容量を大きくしてください

特に次に示す場合は RD エリアの容量を大きくするか，又は RD エリアを作成する必要があります。

- 同じ RD エリア内の表に対してこのメッセージが頻繁に出力される場合
- 表の再編成中又は表の再編成の直後にこのメッセージが出力される場合
- インデクスの再編成中又はインデクスの再編成の直後にこのメッセージが出力される場合

このような場合はデータベース構成変更ユティリティ（**pdmod** コマンド）で RD エリアの容量を大きくするか，又は RD エリアを作成してください。

(2)　pddbls コマンドで RD エリアの未使用セグメント数が分かります

pddbls コマンドで，RD エリアの残容量（未使用セグメント数）が分かります。

例題

RD エリア（RDAREA1）の未使用セグメント数を pddbls コマンドで調べます。

```
pddbls -r RDAREA1 -a

STATE OF RDAREA
    RDAREA        ID       STATUS                TYPE
                           OPNMODE
    RDAREA1       4        OPEN                  USER
                           INITIAL
    SEGMENT   75 / 700
```

〔説明〕

未使用セグメント数／全セグメント数が SEGMENT に表示されます。未使用セグメント数が RD エリアの残容量となります。この例では，未使用セグメント数が 75 で，全セグメント数が 700 です。未使用セグメント数が残り少なくなったら，RD エリアを拡張します。

(3)　リスト用 RD エリアの容量が不足した場合

ASSIGN LIST 文を実行すると，RD エリアのリスト作成数オーバー（**KFPA11812-E** メッセージ）となる場合は，データベース構成変更ユティリティでリスト用 RD エリアを追加してください。ASSIGN LIST 文を実行すると，RD エリアのページ数不足（**KFPA11756-E** メッセージ）となる場合は，次に示すどれかの方法で，RD エリアの容量を大きくしてください。

- RD エリアの追加
- RD エリアの拡張
- RD エリアの再初期化

ただし，RD エリアの再初期化でリスト用 RD エリアの容量を拡張すると，リスト用 RD エリア内のリストが削除されます。この場合，ASSIGN LIST 文でリストを再作成する必要があります。

(4)　RD エリアの自動増分機能を利用してください

RD エリアが容量不足になったとき，HiRDB ファイルシステム領域内に空き領域があれば，自動的に RD エリアを拡張します。これを **RD エリアの自動増分**といいます。RD エリアの自動増分については，「RD エリアの自動増分」を参照してください。

15.2　RD エリアを作成する方法（RD エリアの追加）

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **create rdarea 文**で，新規の RD エリアを追加できます。

15.2.1　RD エリアを追加する前に

(1)　追加できる RD エリア

追加できる RD エリアを次に示します。

- データディクショナリ用 RD エリア※
- データディクショナリ LOB 用 RD エリア
- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア
- リスト用 RD エリア

注※

ストアドプロシジャ，ストアドファンクション，及びトリガに関するディクショナリ表，及び再編成時期予測機能で使用する解析情報表と運用履歴表を格納するデータディクショナリ用 RD エリアのことです。

(2)　RD エリアを追加するときは？

次の場合に RD エリアを新規に追加します。

1. 新しい表やインデクスを作成する場合，既にある RD エリアには定義したくないとき
2. 新しく追加する RD エリアに一部の表やインデクスを移すことで，既にある RD エリアの空き領域を大きくしたい場合
3. 絞込み検索をする場合，又はリストの作成数がオーバーした場合（リスト用 RD エリアを追加します）

(3)　注意事項

RD エリアを追加する場合は次に示すオペランドの値に注意してください。これらのオペランドの値を超えて RD エリアを追加できません。

- システム定義の pd_max_rdarea_no オペランドで指定した RD エリアの最大数
- システム定義の pd_max_file_no オペランドで指定した RD エリアを構成する HiRDB ファイルの最大数

(4)　グローバルバッファの割り当て

追加した RD エリアをすぐに使用する場合は，(a)～(c)に示すどれかの方法でグローバルバッファを割り当てる必要があります。

(a)　pdmod コマンドの実行時にグローバルバッファを割り当てる方法

pdmod コマンドの create rdarea 文の **globalbuffer** オペランドでグローバルバッファを割り当てます。この場合，既存のグローバルバッファ（システム定義の pdbuffer オペランドの-r 又は-o オプション指定のグローバルバッファ）にだけ割り当てられます。なお，次に示す場合はこの方法を使用できないため，(b)又は(c)の方法を使用してください。

- 追加する RD エリアのページ長がグローバルバッファのバッファサイズ以上の場合
- インデクス用又は LOB 用のグローバルバッファに割り当てる場合
- 新規追加したグローバルバッファに割り当てる場合

この方法を使用して割り当てたグローバルバッファは，HiRDB を正常終了又は計画停止したときに無効になります。そのため，HiRDB を終了したときに pdbuffer オペランドの指定値を変更してください。なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定値を変更できます。

参考

- pdbuffer オペランドの指定値を変更しないと，次回の HiRDB 開始以降，追加した RD エリアは-o オプション指定のグローバルバッファに割り当てられます。
- HiRDB を再開始した場合は globalbuffer オペランドで指定したグローバルバッファに割り当てられます。

(b)　pdbufmod コマンドでグローバルバッファを割り当てる方法

pdbufmod コマンドでグローバルバッファを割り当てます。この場合，既存のグローバルバッファだけでなく，新規追加したグローバルバッファにも割り当てられます。ただし，この方法でグローバルバッファを割り当てる場合は次に示す条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入している
- pd_dbbuff_modify オペランドに Y を指定している

pdbufmod コマンドでグローバルバッファを割り当てる方法については，「HiRDB の稼働中にグローバルバッファを追加，変更，又は削除する方法（グローバルバッファの動的変更）」を参照してください。

(c)　システム構成変更コマンド（pdchgconf コマンド）でグローバルバッファを割り当てる方法

システム構成変更コマンドで pdbuffer オペランドの指定値を変更してグローバルバッファを割り当てます。この場合，既存のグローバルバッファだけでなく，新規追加したグローバルバッファにも割り当てられます。ただし，この方法でグローバルバッファを割り当てる場合は HiRDB Advanced High Availability が必要になります。

システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(5)　ストアドプロシジャ又はストアドファンクションを使用するための規則

1. データディクショナリ LOB 用 RD エリアを追加すると，同時にルーチン管理用のディクショナリ表が作成されます。そのため，データディクショナリ用 RD エリアに空きがない場合には，あらかじめデータディクショナリ用 RD エリアを拡張しておいてください。
2. データディクショナリ LOB 用 RD エリアの追加と同時に，ルーチン管理用のディクショナリ表を格納するデータディクショナリ用 RD エリアを追加できます。
3. データディクショナリ LOB 用 RD エリアを追加する場合には，一度に二つの RD エリアを追加してください。

15.2.2　例題

ユーザ用 RD エリア（RDAREA1）を追加します。

〈手順〉

1. **pdfmkfs** コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します。ただし，既存の HiRDB ファイルシステム領域に RD エリアを追加する場合は，この操作は必要ありません。
2. **pdmod** コマンドの制御文ファイルを作成します。
3. **pdmod** コマンドで RD エリアを追加します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdbufmod** コマンドでグローバルバッファを割り当てます。
7. **pdbuffer** オペランドを修正します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) pdfmkfs コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area01
```

〔説明〕

RD エリア用の HiRDB ファイルシステム領域（C:¥rdarea¥area01）を作成します。

-n：HiRDB ファイルシステム領域の大きさをメガバイト単位で指定します。

-l：HiRDB ファイルシステム領域に作成する HiRDB ファイル数の最大値を指定します。

-k：RD エリア用の HiRDB ファイルシステム領域を作成するため，DB を指定します。

-i：HiRDB ファイルシステム領域を初期化することを指定します。

C:¥rdarea¥area01：作成する HiRDB ファイルシステム領域の名称を指定します。

(2) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **create rdarea 文**を記述した制御文ファイル（C:¥pdmod¥create01）を作成します。制御文ファイルの内容を次に示します。

```
create rdarea RDAREA1                              1
   globalbuffer gbuf01                             2
   for user used by PUBLIC                         3
   server name bes1                                4
   page 4096 characters                            5
   storage control segment 10 pages                6
   file name ”C:¥rdarea¥area01¥file01”              7
      initial 1000 segments;                       8
```

〔説明〕

1. 追加する RD エリア（RDAREA1）を指定します。
2. RDAREA1 に割り当てるグローバルバッファ（gbuf01）を指定します。次回の HiRDB 開始以降，ここで指定したグローバルバッファに割り当てられないため，pdbuffer オペランドの指定値を変更する必要があります。なお，(6)でグローバルバッファを割り当てる場合はこのオペランドを指定する必要はありません。
3. RDAREA1 を公用 RD エリアにします。
4. このオプションは HiRDB/パラレルサーバのときだけ指定してください。ここには RD エリアを追加するサーバの名称を指定します。
5. ページ長を指定します。
6. セグメントサイズを指定します。
7. RD エリアを構成する HiRDB ファイルを指定します。

   C:¥rdarea¥rdarea01 は，(1)で作成した HiRDB ファイルシステム領域です。
8. HiRDB ファイルのセグメント数を指定します。

(3) pdmod コマンドで RD エリアを追加します

```
pdmod -a C:¥pdmod¥create01
```

〔説明〕

-a：(2)で作成した pdmod コマンドの制御文ファイル名を指定します。

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを追加すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアの内容が更新されます。そのため，ここでは全 RD エリアのバックアップを取得しています。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) pdbufmod コマンドでグローバルバッファを割り当てます

新規のグローバルバッファ（gbuf01）を追加して RDAREA1 に割り当てます。

```
pdbufmod -k add -a gbuf01 -r RDAREA1 -n 1000
```

〔説明〕

-k add：グローバルバッファを追加する場合に指定します。

-a：追加するグローバルバッファの名称を指定します。

-r：割り当てる RD エリアを指定します。

-n：グローバルバッファのバッファ面数を指定します。

なお，pdbufmod コマンドを実行する場合は，次に示す条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入している

- システム定義の pd_dbbuff_modify オペランドに Y を指定している

(7) pdbuffer オペランドを修正します

割り当てたグローバルバッファは HiRDB を正常終了又は計画停止したときに無効になります。そのため，HiRDB を終了したときに pdbuffer オペランドの指定を変更してください。pdbuffer オペランドの指定例を次に示します。

```
                          :
pdbuffer -a gbuf01 -r RDAREA1,RDAREA2,RDAREA3 -n 1000
                          :
```

〔説明〕

グローバルバッファ（gbuf01）に，追加した RD エリア（RDAREA1）を割り当てます。

なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定を変更できます。ただし，システム構成変更コマンドを実行する場合は HiRDB Advanced High Availability が必要になります。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.3　RD エリアの容量を大きくする方法（RD エリアの拡張）

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **expand rdarea 文**で，RD エリアの容量を大きく（拡張）できます。RD エリアの拡張とは，格納されているデータを残した状態で，RD エリアの容量を増やすことです。RD エリアを拡張するときは，既にある RD エリアに HiRDB ファイルを追加します。

15.3.1　RD エリアを拡張する前に

(1)　容量を拡張できる RD エリア

容量を拡張できる RD エリアを次に示します。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア
- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- データディクショナリ LOB 用 RD エリア
- レジストリ用 RD エリア
- レジストリ LOB 用 RD エリア
- リスト用 RD エリア

(2)　RD エリアを拡張するときは？

次の場合に RD エリアを拡張します。

1. 表の行数が増加してデータが格納できなくなった場合
2. RD エリアの再初期化の方法では RD エリアの容量を増やせない場合（RD エリアの再初期化については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください）
3. 一つの RD エリアを複数の HiRDB ファイルで構成してもよい場合

(3)　注意事項

1. HiRDB ファイル数を増やす場合は pd_max_file_no オペランドの値に注意してください。RD エリアを構成する全 HiRDB ファイル数がこのオペランドの値を超える場合は RD エリアを追加できません。
2. 拡張する RD エリアを次に示す状態にする必要があります。

- RD エリアが障害閉塞状態の場合は，障害閉塞となった要因を取り除いた後に pdrels コマンドで閉塞状態を解除してください。
- RD エリアがクローズ状態の場合は，pdopen コマンドでオープン状態にしてください。

15.3.2　例題

ユーザ用 RD エリア（RDAREA1）の容量を拡張します。

〈手順〉

1. **pdfmkfs** コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します。拡張する RD エリアの HiRDB ファイルシステム領域に余裕がない場合にこの操作をしてください。
2. **pdmod** コマンドの制御文ファイルを作成します。
3. **pdmod** コマンドで RD エリアを拡張します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　pdfmkfs コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area11
```

〔説明〕

RD エリア用の HiRDB ファイルシステム領域（C:¥rdarea¥area11）を作成します。

-n：HiRDB ファイルシステム領域の大きさをメガバイト単位で指定します。

-l：HiRDB ファイルシステム領域に作成する HiRDB ファイル数の最大値を指定します。

-k：RD エリア用の HiRDB ファイルシステム領域を作成するため，DB を指定します。

-i：HiRDB ファイルシステム領域を初期化することを指定します。

C:¥rdarea¥area11：作成する HiRDB ファイルシステム領域の名称を指定します。

(2)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **expand rdarea 文**を記述した制御文ファイル（C:¥pdmod¥expand01）を作成します。制御文ファイルの内容を次に示します。

```
expand rdarea RDAREA1                              1
   file name ”C:¥rdarea¥area11¥file01”             2
      initial 1000 segments;                       3
```

〔説明〕

1. 拡張する RD エリア（RDAREA1）を指定します。
2. 追加する HiRDB ファイルを指定します。
   C:¥rdarea¥rdarea11 は，(1)で作成した HiRDB ファイルシステム領域です。
3. HiRDB ファイルのセグメント数を指定します。

(3) pdmod コマンドで RD エリアを拡張します

```
pdmod -a C:¥pdmod¥expand01
```

〔説明〕

-a：(2)で作成した pdmod コマンドの制御文ファイル名を指定します。

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02
-p C:¥pdcopy¥list02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを拡張すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアの内容が更新されます。そのため，ここでは全 RD エリアのバックアップを取得しています。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4　RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **initialize rdarea 文**で，RD エリアの容量又は属性を変更（再初期化）できます。RD エリアの再初期化とは，既にあるデータを消去して，RD エリアを構成する HiRDB ファイルを大きく又は小さくしたり，セグメントサイズを変更したりして，RD エリアの容量又は属性を変更することです。

なお，RD エリアを再初期化することで，RD エリアを構成する HiRDB ファイルを変更し，RD エリアのディスク配置を変更できます。手順については，「例題９（RD エリアのディスク配置を変更する場合）」を参照してください。

ただし，マスタディレクトリ用 RD エリア，データディクショナリ用 RD エリア，及びデータディレクトリ用 RD エリアは再初期化できないため，これらの RD エリアのディスク配置を変更する場合は，データベース初期設定ユティリティでデータベースを再度初期設定する必要があります。この場合のデータベースの移行方法については，「システム間で資源を移行する方法」を参照してください。

15.4.1　RD エリアを再初期化する前に

(1)　再初期化できる RD エリア

再初期化できる RD エリアを次に示します。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア
- データディクショナリ LOB 用 RD エリア（オブジェクト格納用だけ）
- レジストリ用 RD エリア
- レジストリ LOB 用 RD エリア
- リスト用 RD エリア

(2)　RD エリアを再初期化するときは？

次の場合に RD エリアを再初期化します。

1. 表の行数が増加してデータが格納できなくなった場合
2. HiRDB ファイルの構成（個数，及び一つの HiRDB ファイルの容量）を変更する場合
3. HiRDB ファイルの構成（HiRDB ファイルの名称）を変更する場合

(3) 注意事項

1. 再初期化の前後に再初期化対象 RD エリアのバックアップを取得してください。再初期化後に RD エリアの障害が発生すると，再初期化以前に取得したバックアップからは回復できません。
2. RD エリアを再初期化すると RD エリア中のデータが消去されます。このため，データベース再編成ユティリティ（pdrorg コマンド）を使用してデータをアンロードしておいてください。
3. 再初期化する RD エリアは pdhold コマンドで閉塞かつクローズ状態にしてください。
4. HiRDB ファイル数を増やす場合は pd_max_file_no オペランドの値に注意してください。RD エリアを構成する全 HiRDB ファイル数がこのオペランドの値を超える場合は RD エリアを追加できません。
5. ユーザ LOB 用 RD エリアを再初期化した場合，LOB 列構成基表のデータは消去されないでそのまま残ります。LOB 列は長さが 0 のデータとして扱われます。
6. ストアドプロシジャのオブジェクト格納用のデータディクショナリ LOB 用 RD エリアを再初期化した場合は，ALTER PROCEDURE ALL ですべての SQL オブジェクトを再作成してください。
7. リスト用 RD エリアを再初期化すると，リスト用 RD エリア内のリストはすべて削除されます。したがって，絞込み検索をする場合は ASSIGN LIST 文で再度リストを作成してください。
8. リスト用 RD エリアに対してはデータベース再編成ユティリティ（pdrorg コマンド）を実行できません。
9. 改竄防止表を格納した RD エリアは再初期化できません。
10. データディクショナリ LOB 用 RD エリア（オブジェクト格納用）を再初期化する場合は，ディクショナリ表の SQL_DIV_COLUMN 表を検索し，データディクショナリ LOB 用 RD エリア名を確認する必要があります。検索例を次に示します。

    例題

    ```
    SELECT RDAREA_NAME FROM MASTER.SQL_DIV_COLUMN
       WHERE TABLE_SCHEMA='HiRDB'
         AND TABLE_NAME='SQL_ROUTINES'
         AND COLUMN_NAME='ROUTINE_OBJECT'
    ```

11. サーバ内分割表に対して RD エリア単位にデータベース再編成ユティリティ（pdrorg）を実行する場合，処理対象の RD エリアだけではなく，表及びインデクスにも排他が掛かります（詳細については，マニュアル「HiRDB コマンドリファレンス」の「ユティリティの排他制御モード」を参照してください）。また，pdmod コマンドで再初期化（initialize rdarea 文）を実行する場合，処理対象の RD エリア及び表に排他モード（EX）の排他が掛かります。そのため，再編成中でない RD エリアに対して再初期化を実行しても排他エラーとなることがあります。例えば，RD エリア R1，R2，及び R3 に横分割表 T1 が格納されていたとします。R1 に対して再編成を実行しているときに，R2 に対して再初期化を実行すると排他エラーとなります。

    このような場合は initialize rdarea 文に without lock table オペランドを指定して再初期化を実行してください。without lock table オペランドを指定すると pdmod コマンドは処理対象の表に排他を掛けなくなるため，再編成と再初期化を同時に実行できます。without lock table オペランドの詳細は，マニュアル「HiRDB コマンドリファレンス」の「RD エリアの再初期化」を参照してください。

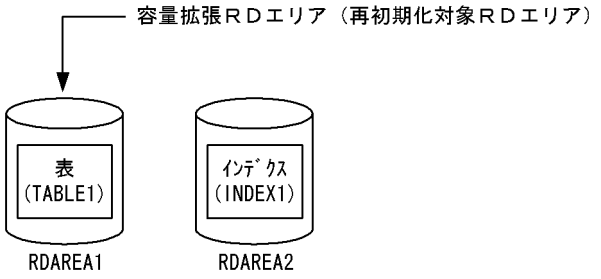
(4) レジストリ用 RD エリア又はレジストリ LOB 用 RD エリアを再初期化する場合

- レジストリ用 RD エリア又はレジストリ LOB 用 RD エリアを再初期化した場合は，プラグインが使用するレジストリ情報を再登録してください。
- レジストリ LOB 用 RD エリアを再初期化した場合は，レジストリ用 RD エリアも再初期化してください。

15.4.2 例題１（インデクスが定義されている場合）

ユーザ用 RD エリア（RDAREA1）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）が格納されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。インデクスはユーザ用 RD エリア（RDAREA2）に格納されています。



注　RDAREA1を再初期化すると，RDAREA2に格納されているINDEX1のデータは削除されます。

〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAREA1 を閉塞します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdrorg** コマンドの制御文ファイルを作成します。
7. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
8. **pdclose** コマンドで RDAREA1 をクローズします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAREA1 を再初期化（容量拡張）します。
11. **pdopen** コマンドで RDAREA1 をオープンします。
12. **pdrorg** コマンドの制御文ファイルを作成します。

13. **pdrorg** コマンドで TABLE1 のデータをリロードします。
14. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
15. **pdcopy** コマンドでバックアップを取得します。
16. **pdrels** コマンドで RDAREA1 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
```

(2)　HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area02
```

(3)　pdhold コマンドで RDAREA1 を閉塞します

```
pdhold -r RDAREA1
```

(4)　pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile
```

〔説明〕

アンロードファイルの名称を指定します。

(7) pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-t：アンロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(8) pdclose コマンドで RDAREA1 をクローズします

```
pdclose -r RDAREA1
```

(9) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA1                              1
   with reconstruction                                 2
   file name ”C:¥rdarea¥area02¥file01”                 3
      initial 3000 segments;                           4
```

〔説明〕

RDAREA1 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA1）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAREA1 を再初期化（容量拡張）します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

(11) pdopen コマンドで RDAREA1 をオープンします

```
pdopen -r RDAREA1
```

(12) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdrorg¥reld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile                          1
index INDEX1 C:¥unld¥index_inf                   2
sort C:¥tmp¥sortwork                             3
```

〔説明〕

1. アンロードファイルの名称を指定します。
2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥unld¥index_inf）を指定します。
3. ソート用ワークディレクトリの名称を指定します。

(13) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -t TABLE1 C:¥pdrorg¥reld01
```

〔説明〕

インデクス（INDEX1）も同時に作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：リロードをするため reld を指定します。

-t：リロードをする表の名称を指定します。

C:¥pdrorg¥reld01：(12)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(14)　pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(15)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(16)　pdrels コマンドで RDAREA1 の閉塞を解除します

```
pdrels -r RDAREA1
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.3　例題２（インデクスが定義されている場合）

ユーザ用 RD エリア（RDAREA2）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）が格納されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。インデクスはユーザ用 RD エリア（RDAREA2）に格納されています。



注　RDAREA2を再初期化しても，RDAREA1に格納されているTABLE1のデータは削除されません。

〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAREA2 を閉塞かつクローズ状態にします。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdmod** コマンドの制御文ファイルを作成します。
7. **pdmod** コマンドで RDAREA2 を再初期化（容量拡張）します。
8. **pdopen** コマンドで RDAREA2 をオープンします。
9. **pdload** コマンドの制御文ファイルを作成します。
10. **pdload** コマンドでインデクスを一括作成します（0 件データロードでインデクスを一括作成します）。
11. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
12. **pdcopy** コマンドでバックアップを取得します。
13. **pdrels** コマンドで RDAREA2 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
```

(2) HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area02
```

(3) pdhold コマンドで RDAREA2 を閉塞かつクローズ状態にします

```
pdhold -r RDAREA2 -c
```

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA2                                  1
   with reconstruction                                     2
   file name ”C:¥rdarea¥area02¥file01”                     3
      initial 3000 segments;                               4
```

〔説明〕

RDAREA2 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA2）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(7) pdmod コマンドで RDAREA2 を再初期化（容量拡張）します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(6)で作成した pdmod コマンドの制御文ファイル名を指定します。

(8) pdopen コマンドで RDAREA2 をオープンします

```
pdopen -r RDAREA2
```

(9) pdload コマンドの制御文ファイルを作成します

pdload コマンドの **source 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdload¥load01）を作成します。制御文ファイルの内容を次に示します。

```
source C:¥load¥loadfile error=¥tmp¥err        1
index INDEX1 C:¥load¥index_inf                2
sort C:¥tmp¥sortwork                          3
```

〔説明〕

1. 入力データファイル及びエラー情報ファイルを指定します。
2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥load¥index_inf）を指定します。
3. ソート用ワークディレクトリの名称を指定します。

(10) pdload コマンドでインデクスを一括作成します（0 件データロードでインデクスを一括作成します）

```
pdload TABLE1 C:¥pdload¥load01
```

〔説明〕

0 件データロードでインデクスを一括作成します。-i オプションを省略してインデクス一括作成モードでインデクス（INDEX1）を一括作成します。

TABLE1：データロードをする表の名称を指定します。

C:¥pdrorg¥load01：(9)で作成した pdload コマンドの制御文ファイル名を指定します。

(11) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(12)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(13)　pdrels コマンドで RDAREA2 の閉塞を解除します

```
pdrels -r RDAREA2
```

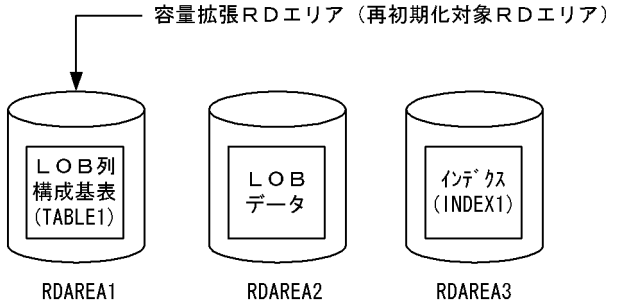
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，プラグインインデクスを格納した RD エリアも同様の手順で再初期化して，容量を増やせます。

15.4.4　例題３（LOB 列が定義されている場合）

ユーザ用 RD エリア（RDAREA1）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）の LOB 列構成基表が格納されています。
- LOB データはユーザ LOB 用 RD エリア（RDAREA2）に格納されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。インデクスはユーザ用 RD エリア（RDAREA3）に格納されています。



注　RDAREA1を再初期化すると，RDAREA2に格納されているLOBデータ，
及びRDAREA3に格納されているインデクスデータは削除されます。

〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAREA1 及び RDAREA2 を閉塞します。

4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdrorg** コマンドの制御文ファイルを作成します。
7. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
8. **pdclose** コマンドで RDAREA1 をクローズします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAREA1 を再初期化（容量拡張）します。
11. **pdopen** コマンドで RDAREA1 をオープンします。
12. **pdrorg** コマンドの制御文ファイルを作成します。
13. **pdrorg** コマンドで TABLE1 のデータをリロードします。
14. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
15. **pdcopy** コマンドでバックアップを取得します。
16. **pdrels** コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
```

(2)　HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area02
```

(3)　pdhold コマンドで RDAREA1 及び RDAREA2 を閉塞します

```
pdhold -r RDAREA1,RDAREA2
```

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile1
```

〔説明〕

アンロードファイルの名称を指定します。

(7) pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-j：LOB 列がある表をアンロードする場合に指定します。

-t：アンロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(8) pdclose コマンドで RDAREA1 をクローズします

```
pdclose -r RDAREA1
```

(9) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA1                                    1
   with reconstruction                                       2
   file name ”C:¥rdarea¥area02¥file01”                       3
      initial 3000 segments;                                 4
```

〔説明〕

RDAREA1 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA1）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAREA1 を再初期化（容量拡張）します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

(11) pdopen コマンドで RDAREA1 をオープンします

```
pdopen -r RDAREA1
```

(12) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdrorg¥reld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile                    1
index INDEX1 C:¥unld¥index_inf             2
sort C:¥tmp¥sortwork                       3
```

〔説明〕

1. アンロードファイルの名称を指定します。
2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥unld¥index_inf）を指定します。
3. ソート用ワークディレクトリの名称を指定します。

(13) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -j -t TABLE1 C:¥pdrorg¥reld01
```

〔説明〕

インデクス（INDEX1）も同時に作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：リロードをするため reld を指定します。

-j：LOB 列がある表をリロードする場合に指定します。

-t：リロードをする表の名称を指定します。

C:¥pdrorg¥reld01：(12)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(14) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(15) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(16) pdrels コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します

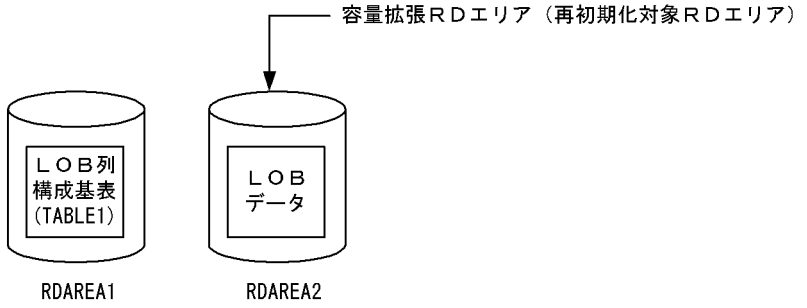
```
pdrels -r RDAREA1,RDAREA2
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.5　例題４（LOB 列が定義されている場合）

ユーザ用 LOB 用 RD エリア（RDAREA2）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）の LOB 列構成基表が格納されています。
- LOB データはユーザ LOB 用 RD エリア（RDAREA2）に格納されています。



注　RDAREA2を再初期化しても，RDAREA1に格納されているLOB列構成基表のデータは削除されません。

〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAREA1 及び RDAREA2 を閉塞します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdrorg** コマンドの制御文ファイルを作成します。
7. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
8. **pdclose** コマンドで RDAREA2 をクローズします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAREA2 を再初期化（容量拡張）します。
11. **pdopen** コマンドで RDAREA2 をオープンします。
12. **pdrorg** コマンドで TABLE1 のデータをリロードします。
13. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
14. **pdcopy** コマンドでバックアップを取得します。
15. **pdrels** コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
```

(2) HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area02
```

(3) pdhold コマンドで RDAREA1 及び RDAREA2 を閉塞します

```
pdhold -r RDAREA1,RDAREA2
```

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6)　pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile
```

〔説明〕

アンロードファイルの名称を指定します。

(7)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-j：LOB 列がある表をアンロードする場合に指定します。

-t：アンロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(8)　pdclose コマンドで RDAREA2 をクローズします

```
pdclose -r RDAREA2
```

(9)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA2                              1
   with reconstruction                                 2
   file name ”C:¥rdarea¥area02¥file01”                 3
      initial 3000 segments;                           4
```

〔説明〕

RDAREA2 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA2）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAREA2 を再初期化（容量拡張）します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

(11) pdopen コマンドで RDAREA2 をオープンします

```
pdopen -r RDAREA2
```

(12) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：リロードをするため reld を指定します。

-j：LOB 列がある表をリロードする場合に指定します。

-t：リロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(13) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(14) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(15) pdrels コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します

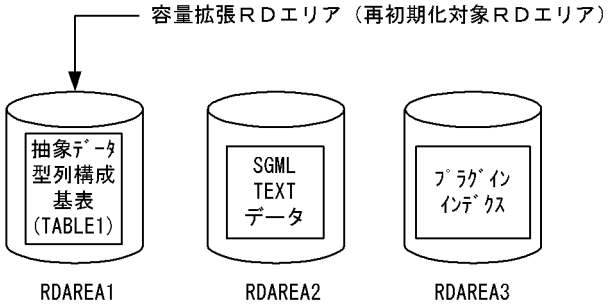
```
pdrels -r RDAREA1,RDAREA2
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.6　例題５（抽象データ型が定義されている場合）

ユーザ用 RD エリア（RDAREA1）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）の抽象データ型列構成基表が格納されています。TABLE1 には，プラグインが提供する抽象データ型（SGMLTEXT 型）が定義されています。
- SGMLTEXT データは，ユーザ LOB 用 RD エリア（RDAREA2）に格納されています。
- プラグインインデクスは，ユーザ LOB 用 RD エリア（RDAREA3）に格納されています。



注　RDAREA1を再初期化すると，RDAREA2に格納されているSGMLTEXTデータ，
及びRDAREA3に格納されているプラグインインデクスのデータは削除されます。

〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAREA1 及び RDAREA2 を閉塞します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdrorg** コマンドの制御文ファイルを作成します。
7. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
8. **pdclose** コマンドで RDAREA1 をクローズします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAREA1 を再初期化（容量拡張）します。
11. **pdopen** コマンドで RDAREA1 をオープンします。
12. **pdrorg** コマンドの制御文ファイルを作成します。
13. **pdrorg** コマンドで TABLE1 のデータをリロードします。
14. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
15. **pdcopy** コマンドでバックアップを取得します。
16. **pdrels** コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
```

(2)　HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area02
```

(3)　pdhold コマンドで RDAREA1 及び RDAREA2 を閉塞します

```
pdhold -r RDAREA1,RDAREA2
```

(4)　pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6)　pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile1
```

〔説明〕

アンロードファイルの名称を指定します。

(7)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-j：次に示す表をアンロードする場合に指定します。

・LOB 列がある表

・LOB 属性の抽象データ型を定義した表

-t：アンロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(8)　pdclose コマンドで RDAREA1 をクローズします

```
pdclose -r RDAREA1
```

(9)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA1                              1
   with reconstruction                                 2
   file name ”C:¥rdarea¥area02¥file01”                 3
      initial 3000 segments;                           4
```

〔説明〕

RDAREA1 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA1）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAREA1 を再初期化（容量拡張）します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

(11) pdopen コマンドで RDAREA1 をオープンします

```
pdopen -r RDAREA1
```

(12) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdrorg¥reld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile                    1
index INDEX1 C:¥unld¥index_inf             2
sort C:¥tmp¥sortwork                       3
```

〔説明〕

1. アンロードファイルの名称を指定します。
2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥unld¥index_inf）を指定します。
3. ソート用ワークディレクトリの名称を指定します。

(13) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -j -t TABLE1 C:¥pdrorg¥reld01
```

〔説明〕

インデクス（INDEX1）も同時に作成するため，-i オプションを省略してインデクス一括作成モードでインデクスを一括作成します。

-k：リロードをするため reld を指定します。

-j：次に示す表をリロードする場合に指定します。

・LOB 列がある表

・LOB 属性の抽象データ型を定義した表

-t：リロードをする表の名称を指定します。

C:¥pdrorg¥reld01：(12)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(14)　pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(15)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(16)　pdrels コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します

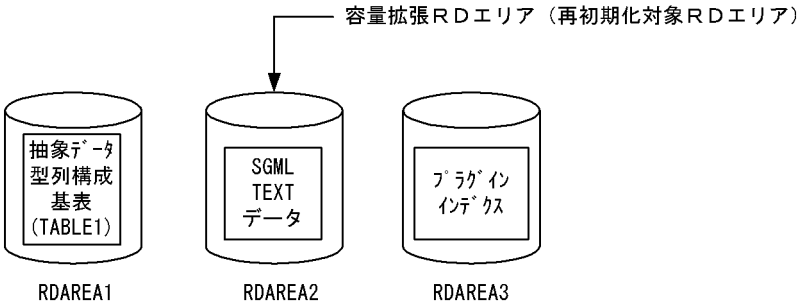
```
pdrels -r RDAREA1,RDAREA2
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.7　例題６（抽象データ型が定義されている場合）

ユーザ LOB 用 RD エリア（RDAREA2）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）の抽象データ型列構成基表が格納されています。TABLE1 には，プラグインが提供する抽象データ型（SGMLTEXT 型）が定義されています。
- SGML TEXT データは，ユーザ LOB 用 RD エリア（RDAREA2）に格納されています。
- プラグインインデクスは，ユーザ LOB 用 RD エリア（RDAREA3）に格納されています。



注　RDAREA2を再初期化しても，RDAREA1及びRDAREA3に格納されているデータは削除されません。

〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAREA1 及び RDAREA2 を閉塞します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdrorg** コマンドの制御文ファイルを作成します。
7. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
8. **pdclose** コマンドで RDAREA2 をクローズします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAREA2 を再初期化（容量拡張）します。
11. **pdopen** コマンドで RDAREA2 をオープンします。
12. **pdrorg** コマンドの制御文ファイルを作成します。
13. **pdrorg** コマンドで TABLE1 のデータをリロードします。
14. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
15. **pdcopy** コマンドでバックアップを取得します。
16. **pdrels** コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
```

(2) HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area02
```

(3) pdhold コマンドで RDAREA1 及び RDAREA2 を閉塞します

```
pdhold -r RDAREA1,RDAREA2
```

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile
```

〔説明〕

アンロードファイルの名称を指定します。

(7) pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-j：次に示す表をアンロードする場合に指定します。

・LOB 列がある表

・LOB 属性の抽象データ型を定義した表

-t：アンロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(8) pdclose コマンドで RDAREA2 をクローズします

```
pdclose -r RDAREA2
```

(9) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA2                                    1
   with reconstruction                                       2
   file name ”C:¥rdarea¥area02¥file01”                       3
      initial 3000 segments;                                 4
```

〔説明〕

RDAREA2 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA2）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAREA2 を再初期化（容量拡張）します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

(11) pdopen コマンドで RDAREA2 をオープンします

```
pdopen -r RDAREA2
```

(12) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdrorg¥reld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile                    1
index INDEX1 C:¥unld¥index_inf             2
sort C:¥tmp¥sortwork                       3
```

〔説明〕

1. アンロードファイルの名称を指定します。
2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥unld¥index_inf）を指定します。
3. ソート用ワークディレクトリの名称を指定します。

(13) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：リロードをするため reld を指定します。

-j：次に示す表をリロードする場合に指定します。

- LOB 列がある表
- LOB 属性の抽象データ型を定義した表

-t：リロードをする表の名称を指定します。

C:¥pdrorg¥unld01：(12)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(14) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(15)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(16)　pdrels コマンドで RDAREA1 及び RDAREA2 の閉塞を解除します

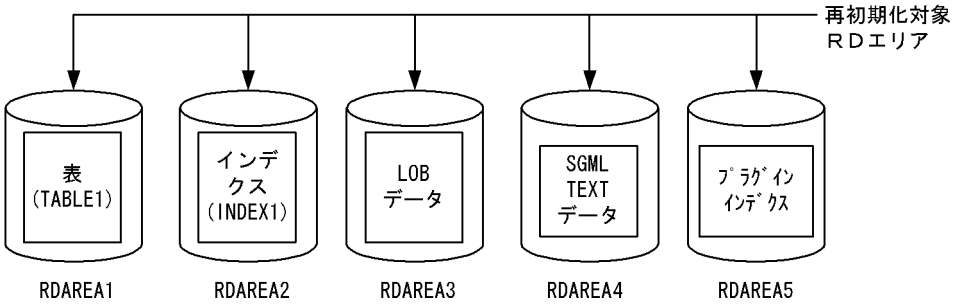
```
pdrels -r RDAREA1,RDAREA2
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.8　例題 7（表に関連する全 RD エリアを再初期化してデータの回復をユティリティで行う場合）

ユーザ用 RD エリア（RDAREA1，RDAREA2）とユーザ LOB 用 RD エリア（RDAREA3，RDAREA4，RDAREA5）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には表（TABLE1）の LOB 列構成基表及び抽象データ型列構成基表が格納されています。TABLE1 には，プラグインが提供する抽象データ型（SGMLTEXT 型）が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。インデクスはユーザ用 RD エリア（RDAREA2）に格納されています。
- LOB データは，ユーザ LOB 用 RD エリア（RDAREA3）に格納されています。
- SGMLTEXT データは，ユーザ LOB 用 RD エリア（RDAREA4）に格納されています。
- プラグインインデクスは，ユーザ LOB 用 RD エリア（RDAREA5）に格納されています。



〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。

2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAERA1～RDAREA5 を閉塞します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdrorg** コマンドの制御文ファイルを作成します。
7. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
8. **pdclose** コマンドで RDAERA1～RDAREA5 をクローズします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAERA1～RDAREA5 を再初期化します。
11. **pdopen** コマンドで RDAERA1～RDAREA5 をオープンします。
12. **pdrorg** コマンドの制御文ファイルを作成します。
13. **pdrorg** コマンドで TABLE1 のデータをリロードします。
14. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
15. **pdcopy** コマンドでバックアップを取得します。
16. **pdrels** コマンドで RDAERA1～RDAREA5 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
pdfstatfs C:¥rdarea¥area02
pdfstatfs C:¥rdarea¥area03
pdfstatfs C:¥rdarea¥area04
pdfstatfs C:¥rdarea¥area05
```

再初期化対象 RD エリアのすべての HiRDB ファイルシステム領域を確認してください。

(2) HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area11
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area12
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area13
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area14
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area15
```

(3) pdhold コマンドで RDAERA1～RDAREA5 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの unload 文を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile
```

〔説明〕

アンロードファイルの名称を指定します。

(7)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-j：次に示す表をアンロードする場合に指定します。

・LOB 列がある表

・LOB 属性の抽象データ型を定義した表

-t：アンロードする表の名称を指定します。

C:¥pdrorg¥unld01：(6)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(8)　pdclose コマンドで RDAERA1～RDAREA5 をクローズします

```
pdclose -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(9)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA1                                    1
   with reconstruction                                       2
   file name ”C:¥rdarea¥area11¥file01”                       3
      initial 3000 segments;                                 4
initialize rdarea RDAREA2
   with reconstruction
   file name ”C:¥rdarea¥area12¥file01”
      initial 3000 segments;
initialize rdarea RDAREA3
   with reconstruction
   file name ”C:¥rdarea¥area13¥file01”
      initial 3000 segments;
initialize rdarea RDAREA4
   with reconstruction
   file name ”C:¥rdarea¥area14¥file01”
       initial 3000 segments;
initialize rdarea RDAREA5
   with reconstruction
   file name ”C:¥rdarea¥area15¥file01”
       initial 3000 segments;
```

〔説明〕

再初期化する RD エリアには新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリアを指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。

3. RD エリアを構成する HiRDB ファイルを指定します。

4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAERA1～RDAREA5 を再初期化します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

> **参考**
>
> RDAREA1 の再初期化時，KFPX14255-W，KFPX24231-W，及び KFPX24242-W の RD エリア状態不正メッセージが出力されます。これは，RDAREA2～RDAREA5 を同時に再初期化しているためで問題はありません。また，RDAREA2，RDAREA4，RDAREA5 の再初期化時，インデクスが未完状態に，LOB 属性の抽象データ型がアクセス禁止状態になりますが，(13)のデータのリロード時にそれぞれ解除されるため問題ありません。

(11) pdopen コマンドで RDAERA1～RDAREA5 をオープンします

```
pdopen -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(12) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdrorg¥reld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile                    1
index INDEX1 C:¥unld¥index_inf             2
sort C:¥tmp¥sortwork                       3
```

〔説明〕

1. アンロードファイルの名称を指定します。

2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥unld¥index_inf）を指定します。

3. ソート用ワークディレクトリの名称を指定します。

(13) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -j -t TABLE1 C:¥pdrorg¥reld01
```

〔説明〕

-k：リロードをするために reld を指定します。

-j：次に示す表をリロードする場合に指定します。

・LOB 列がある表

・LOB 属性の抽象データ型を定義した表

-t：リロードする表の名称を指定します。

C:¥pdrorg¥reld01：(12)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(14) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(15) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(16) pdrels コマンドで RDAERA1～RDAREA5 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

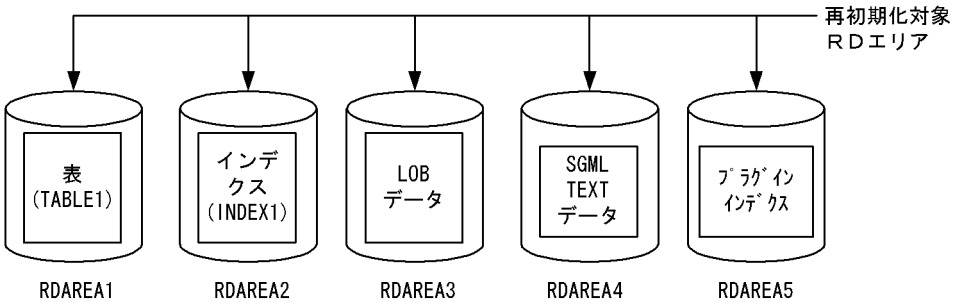
コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.9 例題８（表に関連する全 RD エリアを再初期化してデータの回復を UAP で行う場合）

ユーザ用 RD エリア（RDAREA1，RDAREA2）とユーザ LOB 用 RD エリア（RDAREA3，RDAREA4，RDAREA5）を再初期化します。このとき，RD エリアの容量を大きくします。

- RDAREA1 には，表（TABLE1）の LOB 列構成基表及び抽象データ型列構成基表が格納されています。TABLE1 には，プラグインが提供する抽象データ型（SGMLTEXT 型）が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。インデクスはユーザ用 RD エリア（RDAREA2）に格納されています。
- LOB データは，ユーザ LOB 用 RD エリア（RDAREA3）に格納されています。
- SGMLTEXT データは，ユーザ LOB 用 RD エリア（RDAREA4）に格納されています。

- プラグインインデクスは，ユーザ LOB 用 RD エリア（RDAREA5）に格納されています。



〈手順〉

1. **pdfstatfs** コマンドで HiRDB ファイルシステム領域に空きがあるか確認します。
2. HiRDB ファイルシステム領域を準備します。
3. **pdhold** コマンドで RDAERA1～RDAREA5 を閉塞及びクローズ状態にします。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. **pdmod** コマンドの制御文ファイルを作成します。
7. **pdmod** コマンドで RDAERA2～RDAREA5 を再初期化します。
8. **pdrels** コマンドで RDAREA2～RDAREA5 を閉塞解除してオープン状態にします。
9. **pdmod** コマンドの制御文ファイルを作成します。
10. **pdmod** コマンドで RDAREA1 を再初期化します。
11. **pdrels** コマンドで RDAREA1 を閉塞解除してオープン状態にします。
12. UAP を実行します。
13. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
14. **pdcopy** コマンドでバックアップを取得します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) pdfstatfs コマンドで HiRDB ファイルシステム領域に空きがあるか確認します

```
pdfstatfs C:¥rdarea¥area01
pdfstatfs C:¥rdarea¥area02
pdfstatfs C:¥rdarea¥area03
pdfstatfs C:¥rdarea¥area04
pdfstatfs C:¥rdarea¥area05
```

再初期化対象 RD エリアのすべての HiRDB ファイルシステム領域を確認してください。

(2) HiRDB ファイルシステム領域を準備します

(1)の結果，HiRDB ファイルシステム領域に空きがないとします。再初期化するときに RD エリアの容量を大きくするため，次に示すどれかの方法で HiRDB ファイルシステム領域を準備してください。

1. 既存の HiRDB ファイルシステム領域より大きい HiRDB ファイルシステム領域を新規に割り当てる
2. 既存の HiRDB ファイルシステム領域に加えて新規の HiRDB ファイルシステム領域を割り当てる
3. 既存の HiRDB ファイルシステム領域を大きくする

ここでは，1 の方法で HiRDB ファイルシステム領域を準備します。

```
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area11
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area12
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area13
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area14
pdfmkfs -n 100 -l 10 -k DB -i C:¥rdarea¥area15
```

(3) pdhold コマンドで RDAERA1～RDAREA5 を閉塞及びクローズ状態にします

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5 -c
```

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA2                                    1
   with reconstruction                                       2
   file name ”C:¥rdarea¥area12¥files01”                      3
      initial 3000 segments;                                 4
initialize rdarea RDAREA3
   with reconstruction
   file name ”C:¥rdarea¥area13¥files01”
      initial 3000 segments;
initialize rdarea RDAREA4
   with reconstruction
   file name ”C:¥rdarea¥area14¥files01”
       initial 3000 segments;
initialize rdarea RDAREA5
   with reconstruction
   file name ”C:¥rdarea¥area15¥files01”
       initial 3000 segments;
```

〔説明〕

再初期化する RD エリアには新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリアを指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(7)　pdmod コマンドで RDAERA2～RDAREA5 を再初期化します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(6)で作成した pdmod コマンドの制御文ファイル名を指定します。

> **参考**
>
> RDAREA2，RDAREA4，RDAREA5 の再初期化時，インデクスが未完状態に，LOB 属性の抽象データ型がアクセス禁止状態になりますが，(10)の RDAREA1 の再初期化時にそれぞれ解除されるため問題ありません。

(8) pdrels コマンドで RDAREA2～RDAREA5 を閉塞解除してオープン状態にします

```
pdrels -r RDAREA2,RDAREA3,RDAREA4,RDAREA5 -o
```

(9) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init02）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA1                                    1
   with reconstruction                                       2
   file name ”C:¥rdarea¥area11¥file01”                       3
      initial 3000 segments;                                 4
```

〔説明〕

RDAREA1 には新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリア（RDAREA1）を指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。
4. HiRDB ファイルのセグメント数を指定します。

(10) pdmod コマンドで RDAREA1 を再初期化します

```
pdmod -a C:¥pdmod¥init02
```

〔説明〕

-a：(9)で作成した pdmod コマンドの制御文ファイル名を指定します。

(11) pdrels コマンドで RDAREA1 を閉塞解除してオープン状態にします

```
pdrels -r RDAREA1 -o
```

(12) UAP を実行します

TABLE1 にデータを挿入する UAP を実行してデータを回復します。

(13) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(14)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

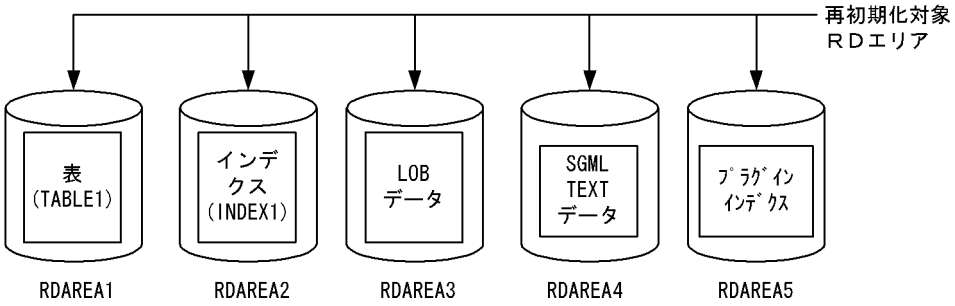
RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.4.10　例題９（RD エリアのディスク配置を変更する場合）

ユーザ用 RD エリア（RDAREA1，RDAREA2）とユーザ LOB 用 RD エリア（RDAREA3，RDAREA4，RDAREA5）を再初期化します。このとき，RD エリアのディスク配置を C:¥rdarea から D:¥rdarea2 に変更します。

- RDAREA1 には表（TABLE1）の LOB 列構成基表及び抽象データ型列構成基表が格納されています。TABLE1 には，プラグインが提供する抽象データ型（SGMLTEXT 型）が定義されています。
- TABLE1 にはインデクス（INDEX1）が定義されています。インデクスはユーザ用 RD エリア（RDAREA2）に格納されています。
- LOB データは，ユーザ LOB 用 RD エリア（RDAREA3）に格納されています。
- SGMLTEXT データは，ユーザ LOB 用 RD エリア（RDAREA4）に格納されています。
- プラグインインデクスは，ユーザ LOB 用 RD エリア（RDAREA5）に格納されています。



〈手順〉

1. HiRDB ファイルシステム領域を準備します。
2. **pdhold** コマンドで RDAERA1～RDAREA5 を閉塞します。
3. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
4. **pdcopy** コマンドでバックアップを取得します。

5. **pdrorg** コマンドの制御文ファイルを作成します。
6. **pdrorg** コマンドで TABLE1 のデータをアンロードします。
7. **pdclose** コマンドで RDAERA1～RDAREA5 をクローズします。
8. **pdmod** コマンドの制御文ファイルを作成します。
9. **pdmod** コマンドで RDAERA1～RDAREA5 を再初期化します。
10. **pdopen** コマンドで RDAERA1～RDAREA5 をオープンします。
11. **pdrorg** コマンドの制御文ファイルを作成します。
12. **pdrorg** コマンドで TABLE1 のデータをリロードします。
13. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
14. **pdcopy** コマンドでバックアップを取得します。
15. **pdrels** コマンドで RDAERA1～RDAREA5 の閉塞を解除します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　HiRDB ファイルシステム領域を準備します

新しく RD エリアを配置するディスク（D:¥rdarea2）に HiRDB ファイルシステム領域を準備してください。

```
pdfmkfs -n 100 -l 10 -k DB -i D:¥rdarea2¥area11
pdfmkfs -n 100 -l 10 -k DB -i D:¥rdarea2¥area12
pdfmkfs -n 100 -l 10 -k DB -i D:¥rdarea2¥area13
pdfmkfs -n 100 -l 10 -k DB -i D:¥rdarea2¥area14
pdfmkfs -n 100 -l 10 -k DB -i D:¥rdarea2¥area15
```

-n 及び-l オプションには，変更前のディスク（C:¥rdarea）に作成していた HiRDB ファイルシステム領域と同じか，又は大きい値を指定してください。

(2)　pdhold コマンドで RDAERA1～RDAREA5 を閉塞します

```
pdhold -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(3)　pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(4)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：このオプションを指定すると，全 RD エリアのバックアップを取得します。RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここでは全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(5)　pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの unload 文を記述した制御文ファイル（C:¥pdrorg¥unld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile
```

〔説明〕

アンロードファイルの名称を指定します。

(6)　pdrorg コマンドで TABLE1 のデータをアンロードします

```
pdrorg -k unld -j -t TABLE1 C:¥pdrorg¥unld01
```

〔説明〕

-k：アンロードをするため unld を指定します。

-j：次に示す表をアンロードする場合に指定します。

- LOB 列がある表
- LOB 属性の抽象データ型を定義した表

-t：アンロードする表の名称を指定します。

C:¥pdrorg¥unld01：(5)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(7)　pdclose コマンドで RDAERA1～RDAREA5 をクローズします

```
pdclose -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(8)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **initialize rdarea 文**を記述した制御文ファイル（C:¥pdmod¥init01）を作成します。制御文ファイルの内容を次に示します。

```
initialize rdarea RDAREA1                                    1
   with reconstruction                                       2
   file name ”D:¥rdarea2¥area11¥file01”                      3
      initial 3000 segments;                                 4
initialize rdarea RDAREA2
   with reconstruction
   file name ”D:¥rdarea2¥area12¥file01”
      initial 3000 segments;
initialize rdarea RDAREA3
   with reconstruction
   file name ”D:¥rdarea2¥area13¥file01”
      initial 3000 segments;
initialize rdarea RDAREA4
   with reconstruction
   file name ”D:¥rdarea2¥area14¥file01”
       initial 3000 segments;
initialize rdarea RDAREA5
   with reconstruction
   file name ”D:¥rdarea2¥area15¥file01”
       initial 3000 segments;
```

〔説明〕

再初期化する RD エリアには新規追加した HiRDB ファイルシステム領域を割り当てます。

1. 再初期化する RD エリアを指定します。
2. 再初期化前とファイル構成が変わるため，with reconstruction を指定します。
3. RD エリアを構成する HiRDB ファイルを指定します。新しく RD エリアを配置するディスク（D:¥rdarea2）を指定します。
4. HiRDB ファイルのセグメント数を指定します。変更前のディスク（C:¥rdarea）に作成していた HiRDB ファイルと同じか，又は大きい値を指定してください。

(9)　pdmod コマンドで RDAERA1～RDAREA5 を再初期化します

```
pdmod -a C:¥pdmod¥init01
```

〔説明〕

-a：(8)で作成した pdmod コマンドの制御文ファイル名を指定します。

> **参考**
>
> RDAREA1 の再初期化時，KFPX14255-W，KFPX24231-W，及び KFPX24242-W の RD エリア状態不正メッセージが出力されます。これは，RDAREA2～RDAREA5 を同時に再初期化しているためで問題はありません。また，RDAREA2，RDAREA4，RDAREA5 の再初期化時，インデクスが未完状態に，LOB 属性の抽象データ型がアクセス禁止状態になりますが，(12)のデータのリロード時にそれぞれ解除されるため問題ありません。

(10) pdopen コマンドで RDAERA1～RDAREA5 をオープンします

```
pdopen -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

(11) pdrorg コマンドの制御文ファイルを作成します

pdrorg コマンドの **unload 文**，**index 文**，及び **sort 文**を記述した制御文ファイル（C:¥pdrorg¥reld01）を作成します。制御文ファイルの内容を次に示します。

```
unload C:¥unld¥unldfile                    1
index INDEX1 C:¥unld¥index_inf             2
sort C:¥tmp¥sortwork                       3
```

〔説明〕

1. アンロードファイルの名称を指定します。
2. インデクス識別子（INDEX1），及びインデクス情報ファイル名（C:¥unld¥index_inf）を指定します。
3. ソート用ワークディレクトリの名称を指定します。

(12) pdrorg コマンドで TABLE1 のデータをリロードします

```
pdrorg -k reld -j -t TABLE1 C:¥pdrorg¥reld01
```

〔説明〕

-k：リロードをするために reld を指定します。

-j：次に示す表をリロードする場合に指定します。

- LOB 列がある表
- LOB 属性の抽象データ型を定義した表

-t：リロードする表の名称を指定します。

C:¥pdrorg¥reld01：(11)で作成した pdrorg コマンドの制御文ファイル名を指定します。

(13) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(14) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

RD エリアを再初期化すると，「同時にバックアップを取得する必要がある RD エリア」で説明している RD エリアが更新されるため，6.3 で説明している RD エリアのバックアップを取得する必要があります。ここ

では全 RD エリアのバックアップを取得します。バックアップの取得については，「バックアップの取得方法」を参照してください。

(15)　pdrels コマンドで RDAERA1～RDAREA5 の閉塞を解除します

```
pdrels -r RDAREA1,RDAREA2,RDAREA3,RDAREA4,RDAREA5
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.5　RD エリア名を変更する方法（RD エリアの属性変更）

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **alter rdarea 文**で，RD エリア名を変更できます。

15.5.1　RD エリア名を変更する前に

(1)　名称を変更できる RD エリア

マスタディレクトリ用 RD エリア以外のすべての RD エリアの名称を変更できます。

(2)　RD エリア名の変更に伴う作業

RD エリア名を変更すると，定義情報や UAP などに指定している RD エリア名もすべて変更する必要があります。そのため，変更する RD エリアの名称を指定している箇所を調査しておいてください。

RD エリア名を変更した場合に必要な作業を次の表に示します。この表の項目以外にも RD エリア名を指定している箇所があれば，変更が必要です。

表 15–1　RD エリア名を変更した場合に必要な作業

項目	必要な作業	必要な作業をしなかった場合の問題
システム共通定義の pdbuffer オペランド	HiRDB を正常終了したときに，pdbuffer オペランドに指定している RD エリア名を，変更後の RD エリア名に変更してください。	HiRDB の正常終了後に RD エリア名を変更しないで正常開始すると，-o オプション指定のグローバルバッファがある場合は，そのグローバルバッファが名称変更後の RD エリアに割り当てられます。アクセスはできますが，処理性能が劣化するおそれがあります。-o オプションを指定しているグローバルバッファがない場合は，名称変更後の RD エリアにグローバルバッファが割り当てられないため，変更後の RD エリアにアクセスしようとすると，エラーになります。
UAP 環境定義の pdlbuffer オペランド	pdlbuffer オペランドに指定している RD エリア名を，変更後の RD エリア名に変更してください。	ほかの UAP によるグローバルバッファの占有やバッファの排他処理による待ち状態が発生し，性能が劣化するおそれがあります。
SQL 手続き文	データベース構成変更ユティリティ（pdmod）を実行した場合に警告メッセージ（KFPX24238-W）が出力されたときは，ルーチン又はトリガを再作成してください※。	ルーチン又はトリガの実行時に，SQL オブジェクトが無効のため，エラーになります。
UAP	変更前の RD エリア名を指定している UAP を再作成してください。	UAP が正常に動作しないおそれがあります。

項目	必要な作業	必要な作業をしなかった場合の問題
コマンド若しくはユティリティのオプション，又はユティリティの制御文	指定している RD エリア名を，変更後の RD エリア名に変更してください。	コマンド又はユティリティが正常に動作しないおそれがあります。
関連製品	指定している RD エリア名を，変更後の RD エリア名に変更してください。 **（例）** • HiRDB Datareplicator の反映定義 • HiRDB Dataextractor のコマンド引数 • JP1/Performance Management - Agent Option for HiRDB の RD エリアの稼働情報，データベース操作に関する HiRDB ファイルの統計情報，再編成時期予測機能の情報 • JP1/HiCommand Protection Manager for HiRDB の RD エリア名とディスクのマッピング	関連製品が正常に動作しないおそれがあります。
運用スクリプト	運用スクリプトを使用している場合，スクリプト中に指定している RD エリア名を，変更後の RD エリア名に変更してください。	運用スクリプトが正常に動作しないおそれがあります。

注※

ルーチンを再作成する手順を次に示します。

1. ALTER ROUTINE を実行し，ルーチン又はトリガを再作成します。

   SQL 手続き文中に変更前の RD エリア名を指定しているルーチン又はトリガは，ALTER ROUTINE 実行時にエラーとなります。

2. GET DIAGNOSTICS を実行し，ALTER ROUTINE の診断情報でエラーとなったルーチン又はトリガを確認します。

3. DROP PROCEDURE 又は DROP TRIGGER でエラーとなったルーチン又はトリガを削除します。

4. SQL 手続き文中の RD エリア名を，変更後の RD エリア名に変更します。

5. CREATE PROCEDURE 又は CREATE TRIGGER でルーチン又はトリガを再作成します。

(3) 注意事項

1. 名称を変更する RD エリアは，pdhold -c コマンドで閉塞かつクローズ状態にしておきます。ただし，データディクショナリ用 RD エリア及びデータディレクトリ用 RD エリアは，閉塞かつオープン状態にしておきます。

2. 名称を変更した RD エリアを使用するには，pdrels -o コマンドで閉塞状態を解除してオープン状態にする必要があります。

3. データベース再編成ユティリティ（pdrorg）で LOB 列を含む表を再編成する場合，データをアンロードしてからリロードするまでに，LOB データが格納されている RD エリアの名称を変更しないでください。リロードする前に RD エリア名を変更すると，リロード時にエラーになります。

4. インデクス情報出力モード（-i n オプション）でデータベース作成ユティリティ（pdload）又はデータベース再編成ユティリティ（pdrorg）を実行して，後でインデクスを作成する場合，インデクス情報ファイルを出力してからインデクスを作成するまでに，インデクスを格納している RD エリアの名称を変更しないでください。インデクス作成前に RD エリア名を変更すると，インデクス作成時にエラーとなります。
5. プラグインインデクスの遅延一括作成をする場合，プラグインインデクスを格納している RD エリア名を変更しないでください。プラグインインデクスの一括作成前に RD エリア名を変更すると，プラグインインデクスの一括作成ができなくなります。

15.5.2 例題

ユーザ用 RD エリア RDAREA01 の名称を RDAREA10 に変更します。

〈手順〉

1. 名称を変更する RD エリアを **pdhold** コマンドで閉塞かつクローズ状態にします。
2. **pdmod** コマンドの制御文ファイルを作成します。
3. **pdmod** コマンドで RD エリア名を変更します。
4. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
5. **pdcopy** コマンドでバックアップを取得します。
6. 名称を変更した RD エリアを **pdrels** コマンドで閉塞を解除しオープン状態にします。

手順の数字はこの後で説明している（ ）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) 名称を変更する RD エリアを pdhold コマンドで閉塞かつクローズ状態にします

```
pdhold -r RDAREA01 -c
```

(2) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの alter rdarea 文を記述した制御文ファイル（C:¥pdmod¥alter01）を作成します。制御文ファイルの内容を次に示します。

```
alter rdarea RDAREA01 rename rdarea to RDAREA10;
```

〔説明〕

RDAREA01 の名称を RDAREA10 に変更します。

(3) pdmod コマンドで RD エリア名を変更します

```
pdmod -a C:¥pdmod¥alter01
```

〔説明〕

-a：(2)で作成した pdmod コマンドの制御文ファイル名を指定します。

(4) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(5) pdcopy コマンドでバックアップを取得します

RD エリア名を変更すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されます。RD エリア名変更後に障害が発生した場合，最新の状態に回復するために，バックアップを取得します。ここでバックアップを取得していないと，回復する手順が煩雑になります。バックアップの取得については，「バックアップの取得方法」を，バックアップを取得していない場合の回復方法については，「RD エリア名変更時にバックアップを取得していなかった場合の回復方法」を参照してください。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -r RDMAST,RDDIC,RDAREA10 -b C:¥pdcopy¥backup01 -p C:¥pd
copy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-r：バックアップを取得する RD エリアの名称を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

(6) 名称を変更した RD エリアを pdrels コマンドで閉塞を解除しオープン状態にします

```
pdrels -r RDAREA10 -o
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.5.3 RD エリア名変更時にバックアップを取得していなかった場合の回復方法

RD エリア名変更時に次に示す RD エリアのバックアップを取得していなかった場合，名称を変更した RD エリアを最新の状態に回復する方法について説明します。

- 変更した RD エリア
- マスタディレクトリ用 RD エリア
- データディクショナリ用 RD エリア

この場合，次の 2 段階に分けてデータベースを回復する必要があります。

1. RD エリア名を変更する（pdmod コマンド実行）直前の状態まで回復する
2. RD エリア名を変更し，それ以降の更新を手動で反映する

それぞれについて説明します。

(1) RD エリア名を変更する（pdmod コマンド実行）直前の状態まで回復する

マスタディレクトリ用 RD エリア，データディクショナリ用 RD エリア，名称を変更した RD エリアの順に，それまでのバックアップファイルとアンロードログファイルを使用して，名称変更直前の状態まで回復します。データディクショナリ用 RD エリアを回復しないで，先に名称を変更した RD エリアを回復しようとすると，RD エリアを閉塞状態にできません。次に示す手順で回復してください。

1. pdstop コマンドで HiRDB を正常終了します。
2. pdstart -r コマンドで HiRDB を開始します。
3. マスタディレクトリ用 RD エリアを，名称変更直前の状態まで回復します。
   -T オプションを指定して，データベース回復ユティリティ（pdrstr）を実行します。-T オプションには，RD エリア名を変更した時刻の 1 秒前を指定します。
4. pdstop コマンドで HiRDB を正常終了します。
5. pdstart コマンドで HiRDB を正常開始します。
6. pdhold 及び pdclose コマンドでデータディクショナリ用 RD エリアを閉塞かつクローズ状態にします。
7. データディクショナリ用 RD エリアを，名称変更直前の状態まで回復します。
   -T オプションを指定して，データベース回復ユティリティ（pdrstr）を実行します。-T オプションには，RD エリア名を変更した時刻の 1 秒前を指定します。
8. pdhold 及び pdclose コマンドで名称を変更した RD エリアを閉塞かつクローズ状態にします。
9. 名称を変更した RD エリアを名称変更直前の状態まで回復します。
   -T オプションを指定して，データベース回復ユティリティ（pdrstr）を実行します。-T オプションには，RD エリア名を変更した時刻の 1 秒前を指定します。

10. pdrels 及び pdopen コマンドで RD エリアの閉塞を解除し，オープン状態にします。

(2)　RD エリア名を変更し，それ以降の更新を手動で反映する

pdmod コマンドで RD エリア名を変更し，障害発生時点までの更新を手動で反映します。

15.6　RD エリアのオープン契機を変更する方法（RD エリアの属性変更）

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **alter rdarea 文**で，RD エリアのオープン契機を変更できます。

15.6.1　RD エリアのオープン契機を変更する前に

(1)　RD エリアのオープン契機とは

RD エリアのオープン契機とは，HiRDB が行う RD エリアのオープン処理の実施時期のことです。通常，RD エリアのオープン処理は HiRDB の開始時に行われます。このため，RD エリアの数に比例して HiRDB の開始時間が長くなります。この RD エリアのオープン処理を HiRDB 開始時に行わないで，RD エリアへのアクセス処理が発生したときに行うことができます。このように RD エリアのオープン契機を変更できます。RD エリアのオープン契機には，次の表に示す三つの属性があります。

表 15–2　RD エリアのオープン契機

属性	初期状態	オープン契機	クローズ契機	長所	短所
INITIAL	オープン状態	• HiRDB 開始時 • pdopen コマンド実行時	pdclose コマンド実行時	初回 SQL から高速実行	システムの開始に時間が掛かる
DEFER	クローズ状態	• RD エリアの初回アクセス時 • pdopen コマンド実行時	pdclose コマンド実行時	• システムの開始が高速 • 初回アクセス以後は通常 SQL も高速	各 RD エリアの初回アクセスに時間が掛かる
SCHEDULE	クローズ状態	• トランザクション内での RD エリア初回アクセス時 • pdopen コマンド実行時	• トランザクション終了時 • pdclose コマンド実行時	• システムの開始が高速 • ファイルオープンの集中を回避（**DVD-RAM ライブラリ装置**向き）	各トランザクションごとに RD エリアの初回アクセスが高負荷となる

(2)　適用基準

RD エリアのオープン契機は，最初は INITIAL 属性です。次に示す場合に RD エリアのオープン契機の変更を考えてください。

- RD エリアが多いため HiRDB の開始処理時間が長い場合
- DVD-RAM ライブラリ装置を使用する場合

各属性の適する運用形態を次の表に示します。

表 15–3　各属性の適する運用形態

属性	適する運用形態
INITIAL	システム開始時に HiRDB ファイルシステム領域をオープンして，RD エリア情報をメモリ上に常駐します。RD エリアの初回アクセス時にも，そのサーバプロセス上でオープンします。ただし，この場合には RD エリア情報の再作成をしないため，初回 SQL から高速な運用ができます。 システム開始時の RD エリア初期状態はオープン状態であり，以降は障害閉塞への遷移を除いて，運用コマンドの入力がないかぎり RD エリアの状態は遷移しません。 **特殊な運用形態を用いない場合は，この属性を推奨します。** この属性のとき，クローズ状態の RD エリアに対してはアクセスできません。
DEFER	システム開始時に HiRDB ファイルシステム領域のオープンをしないで，RD エリアに対する初回アクセス時にオープンし，RD エリア情報をメモリ上に常駐します。2 回目以降のアクセスでは，HiRDB ファイルシステム領域のオープン以降の処理をしないため，高速な運用ができます。 システム開始時の RD エリア初期状態はクローズ状態であり，各 RD エリアに対する初回アクセス時に該当する RD エリアをオープン状態にします。以降は障害閉塞への遷移を除いて，運用コマンドの入力がないかぎり RD エリアの状態は遷移しません。 **多数の HiRDB ファイルシステム領域に対するオープンが重なるケースを回避したい場合や，HiRDB の開始時間を短縮したい場合に，この属性を指定します。** HiRDB を再開始する場合は，回復処理時に回復対象 RD エリアをオープンします。 この属性を指定した場合，クローズ状態の RD エリアに対してもアクセスできます。
SCHEDULE	システム開始時に HiRDB ファイルシステム領域のオープンをしないで，システム開始後，各トランザクション内での RD エリアに対する初回アクセス時にオープンし，RD エリア情報をメモリ上に常駐します。トランザクションの終了時に，そのトランザクション内でオープンした HiRDB ファイルシステム領域をクローズします。以降もトランザクションが変わると RD エリアに対する初回アクセス時にオープン以降の処理をするため，トランザクションに掛かる負荷は増加します。 システム開始時の RD エリアの初期状態はクローズ状態であり，アクセスのあった RD エリアのトランザクション処理中だけオープン状態とします。トランザクション終了時に，トランザクション内でオープン状態としたすべての RD エリアをクローズ状態にします。 pdopen コマンドを入力すると，次回閉塞クローズ状態になるまでの間オープン状態を継続できます。そのほかの運用コマンドを用いて，RD エリアのステータスを任意に遷移させることもできます。障害を検知したときは障害閉塞となります。 **DVD-RAM ライブラリ装置を使用するなど，多数の HiRDB ファイルシステム領域のオープンが重なることを回避したい場合や，システムの開始時間を短縮したい場合に，この属性を指定します。** HiRDB を再開始する場合は，回復処理時に回復対象 RD エリアをオープンして，回復処理の終了後にクローズします。 この属性を指定した場合，クローズ状態の RD エリアに対してもアクセスできます。

(3)　オープン契機を変更できる RD エリア

オープン契機を変更できる RD エリアを次に示します。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア
- リスト用 RD エリア

(4) 注意事項

1. オープン契機を変更する RD エリアは，pdhold -c コマンドで閉塞かつクローズ状態にしておきます。
2. オープン契機を変更した RD エリアを使用するには，pdrels -o コマンドで閉塞状態を解除してオープン状態にする必要があります。
3. 同じ HiRDB ファイルシステム領域中にある RD エリアのオープン契機は統一してください。
4. 高速系切り替え機能の対象になる待機系ユニットは，待機状態のときに RD エリアをオープンしていません。また，系の切り替え時間を最小限に抑えるため，系切り替えの発生時に全面回復で必要な RD エリアだけをオープンして，そのほかの RD エリアはオープンしません。したがって，待機系の RD エリアのオープン契機は INITIAL 属性になりません。INITIAL 属性の RD エリアは DEFER 属性になります。
5. スタンバイレス型系切り替え機能では系の切り替え時間を最小限に抑えるため，系切り替えの発生時に全面回復で必要な RD エリアだけをオープンして，そのほかの RD エリアはオープンしません。したがって，正規 BES 又は代替部の RD エリアのオープン契機は次のようになります。

   ・系切り替えが発生した場合，代替部の RD エリアのオープン契機は SCHEDULE 属性になります。

   ・障害が回復して正規 BES に系を切り戻した場合，正規 BES 下の INITIAL 又は DEFER 属性の RD エリアのオープン契機は DEFER 属性になります。SCHEDULE 属性の RD エリアは SCHEDULE 属性のままです。
6. オープン属性による UAP の RD エリアへのアクセス可否を次の表に示します。

表 15–4　オープン属性による UAP の RD エリアへのアクセス可否

RD エリアのオープン契機	RD エリアの状態		RD エリアのアクセス可否
INITIAL	閉塞なし	オープン	○
		クローズ	×
	コマンド閉塞	オープン	×
		クローズ	×
	参照可能閉塞	オープン	△※2
		クローズ	×
	参照可能バックアップ閉塞※1	オープン	△※2
	更新可能バックアップ閉塞※1	オープン	○
	障害閉塞	オープン	×
		クローズ	×
	ログレス閉塞	オープン	△※3
		クローズ	×
DEFER 又は SCHEDULE	閉塞なし	オープン	○

RD エリアの オープン契機	RD エリアの状態		RD エリアの アクセス可否
		クローズ	○
	コマンド閉塞	オープン	×
		クローズ	×
	参照可能閉塞	オープン	△※2
		クローズ	△※2
	参照可能バックアップ閉塞※1	オープン	△※2
		クローズ	×
	更新可能バックアップ閉塞※1	オープン	×
		クローズ	○
	障害閉塞	オープン	×
		クローズ	×
	ログレス閉塞	オープン	△※3
		クローズ	△※3

（凡例）

○：アクセスできます。

△：一部の SQL だけアクセスできます。

×：アクセスできません。

注※1

一時表用 RD エリアは，この状態には遷移しません。

注※2

参照系 SQL だけアクセスできます。

注※3

永続実表の PURGE TABLE だけ実行できます。

15.6.2 例題

ユーザ用 RD エリア（RDAREA1～RDAREA3）のオープン契機を INITIAL から SCHEDULE に変更します。RDAREA1～RDAREA3 を除いた全ユーザ用 RD エリアのオープン契機を INITIAL から DEFER に変更します。なお，RDAREA1～RDAREA3 は同じ HiRDB ファイルシステム領域にあるとします。

〈手順〉

1. HiRDB を正常終了します。

2. システム共通定義を変更します。
3. HiRDB を正常開始します。
4. オープン契機を変更する RD エリアを **pdhold** コマンドで閉塞かつクローズ状態にします。
5. **pdmod** コマンドの制御文ファイルを作成します。
6. **pdmod** コマンドで RD エリアのオープン契機を変更します。
7. オープン契機を変更した RD エリアを **pdrels** コマンドで閉塞を解除しオープン状態にします。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　HiRDB を正常終了します

```
pdstop
```

システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できるため，ここで HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(2)　システム共通定義を変更します

システム共通定義で次に示すオペランドを指定します。システム共通定義の内容を次に示します。

```
                :
set pd_rdarea_open_attribute_use = Y                    1
set pd_rdarea_open_attribute = DEFER                    2
                :
```

〔説明〕

1. RD エリアのオープン契機に DEFER 又は SCHEDULE を使用することを指定します。
2. 全 RD エリアのオープン契機の標準値を DEFER にします。

(3)　HiRDB を正常開始します

```
pdstart
```

(4)　オープン契機を変更する RD エリアを pdhold コマンドで閉塞かつクローズ状態にします

```
pdhold -r RDAREA1,RDAREA2,RDAREA3 -c
```

(5) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **alter rdarea 文**を記述した制御文ファイル（C:¥pdmod¥alter01）を作成します。制御文ファイルの内容を次に示します。

```
alter rdarea RDAREA1 open attribute SCHEDULE;
alter rdarea RDAREA2 open attribute SCHEDULE;
alter rdarea RDAREA3 open attribute SCHEDULE;
```

〔説明〕

RDAREA1～RDAREA3 のオープン契機を DEFER から SCHEDULE に変更します。

(6) pdmod コマンドで RD エリアのオープン契機を変更します

```
pdmod -a C:¥pdmod¥alter01
```

〔説明〕

-a：(5)で作成した pdmod コマンドの制御文ファイル名を指定します。

> **参考**
>
> データベース構成変更ユティリティで RD エリアを追加した直後はオープン契機の指定が有効になりません。追加した直後は INITIAL 属性になっています。オープン契機の指定を有効にするには HiRDB を一度終了させた後に再度開始してください。開始モードに関係なく有効になります。

(7) オープン契機を変更した RD エリアを pdrels コマンドで閉塞を解除しオープン状態にします

```
pdrels -r RDAREA1,RDAREA2,RDAREA3 -o
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.7　RD エリアを削除する方法

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **remove rdarea 文**で，RD エリアを削除できます。

15.7.1　RD エリアを削除する前に

(1)　削除できる RD エリア

削除できる RD エリアを次に示します。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア
- リスト用 RD エリア

(2)　RD エリアを削除するときは？

次の場合に RD エリアを削除します。

- 既にある RD エリアが不要になった場合
- RD エリアを作成し直す場合

(3)　注意事項

- 削除する RD エリア内に表又はインデクスがある場合，表又はインデクスを削除してから RD エリアを削除してください。
- 削除する RD エリアは，pdhold コマンドによる閉塞又は障害閉塞の状態で，かつ pdclose コマンドでクローズ状態にしておきます。

(4)　グローバルバッファの定義変更

RD エリアを削除した場合は pdbuffer オペランドの指定値を変更してください。グローバルバッファの定義からその RD エリアを削除してください。pdbuffer オペランドの指定値を変更しないと，メモリの使用効率が悪くなることがあります。例えば，次に示すような場合が考えられます。

- 削除した RD エリアが pdbuffer オペランドの-o オプション指定のグローバルバッファに割り当てられていた場合，ほかの RD エリアに必要以上に大きなグローバルバッファが割り当てられることがあります。

- 削除した RD エリアだけが pdbuffer オペランドの-o オプション指定のグローバルバッファに割り当てられていた場合，使用されないグローバルバッファがメモリ上に確保されたままになります。

15.7.2 例題

ユーザ用 RD エリア（RDAREA1）を削除します。

〈手順〉

1. 削除する RD エリアに格納されている表及びインデクスを削除します。
2. 削除する RD エリアを **pdhold** コマンドで閉塞かつクローズ状態にします。
3. **pdmod** コマンドの制御文ファイルを作成します。
4. **pdmod** コマンドで RD エリアを削除します。
5. **pdbufmod** コマンドでグローバルバッファを削除します。
6. **pdbuffer** オペランドを修正します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) 削除する RD エリアに格納されている表及びインデクスを削除します

```
DROP TABLE TABLE1;
DROP INDEX INDEX1;
```

(2) 削除する RD エリアを pdhold コマンドで閉塞かつクローズ状態にします

```
pdhold -r RDAREA1 -c
```

(3) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **remove rdarea 文**を記述した制御文ファイル（C:¥pdmod¥remove01）を作成します。制御文ファイルの内容を次に示します。

```
remove rdarea RDAREA1;
```

〔説明〕

削除する RD エリア（RDAREA1）を指定します。

(4) pdmod コマンドで RD エリアを削除します

```
pdmod -a C:¥pdmod¥remove01
```

〔説明〕

-a：(3)で作成した pdmod コマンドの制御文ファイル名を指定します。

(5)　pdbufmod コマンドグローバルバッファを削除します

削除した RD エリアに割り当てていたグローバルバッファが不要になった場合（ほかの RD エリアが割り当てられていない場合），そのグローバルバッファを削除します。

```
pdbufmod -k del -a gbuf01
```

〔説明〕

-k del：グローバルバッファを削除する場合に指定します。

-a：削除するグローバルバッファの名称を指定します。

なお，pdbufmod コマンドを実行する場合は，次に示す条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入している
- pd_dbbuff_modify オペランドに Y を指定している

(6)　pdbuffer オペランドを修正します

グローバルバッファの削除情報は HiRDB を正常終了又は計画停止したときに無効になります。そのため，HiRDB を終了したときに，削除した RD エリアの名称を pdbuffer オペランドから削除してください。

なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定を変更できます。ただし，システム構成変更コマンドを実行する場合は HiRDB Advanced High Availability が必要になります。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.8　RD エリアの自動増分

実行者　HiRDB 管理者

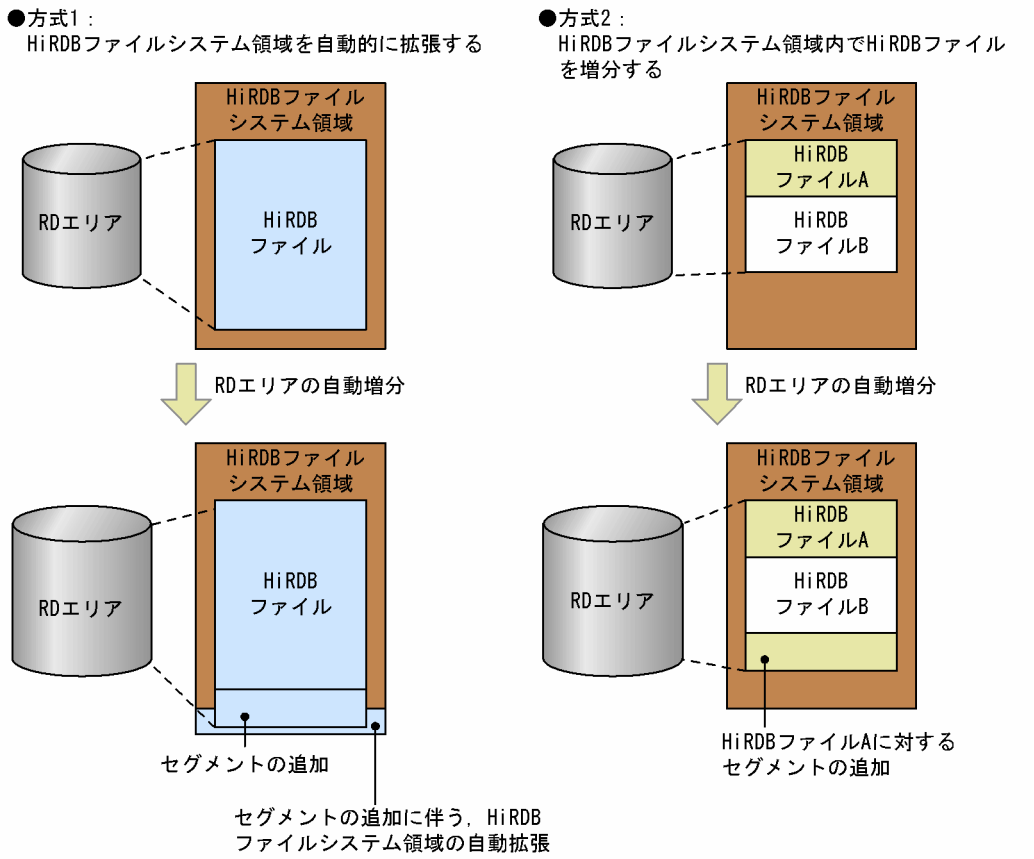
15.8.1　RD エリアの自動増分とは

RD エリアの容量不足が近付いたときに，RD エリアを構成する HiRDB ファイルに未使用セグメントを追加して，RD エリアの容量を自動的に拡張します。これを **RD エリアの自動増分**といいます。RD エリアの自動増分には，次の二つの方式があります。

1. HiRDB ファイルシステム領域を自動的に拡張する方式
2. HiRDB ファイルシステム領域内で HiRDB ファイルを増分する方式

RD エリアの自動増分を次の図に示します。

図 15–1　RD エリアの自動増分



〔説明〕

方式 1 では，セグメントの追加によって HiRDB ファイルシステム領域サイズの上限を超える場合，HiRDB ファイルシステム領域を必要な分だけ自動的に拡張します。この方式では，一つの HiRDB ファイルシステム領域に，HiRDB ファイルを一つだけ作成できます。

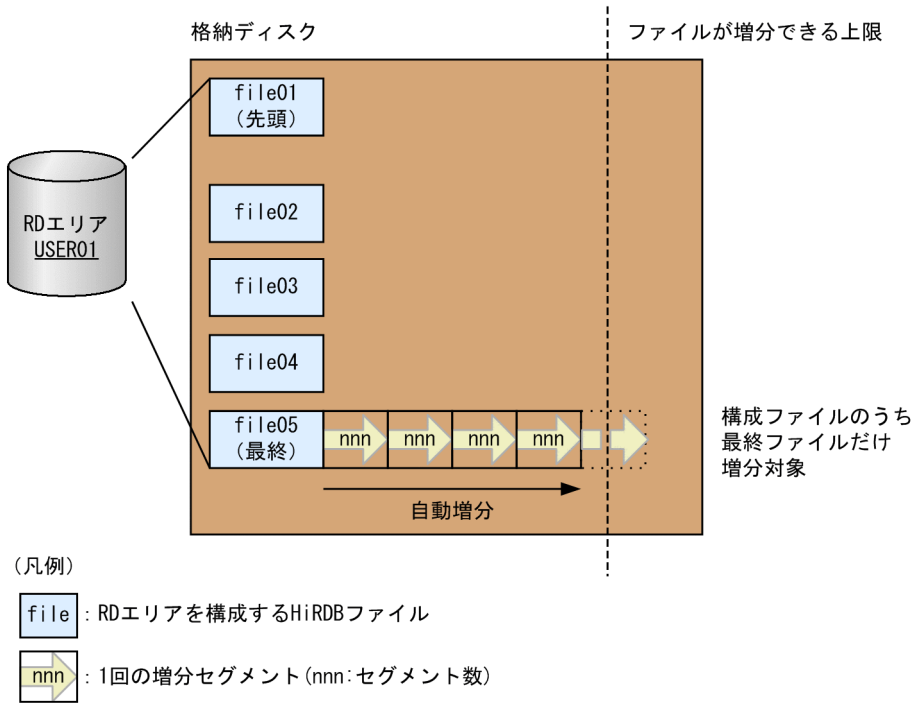
方式 2 では，HiRDB ファイルシステム領域サイズ内で HiRDB ファイルを拡張又は追加し，HiRDB ファイルシステム領域の上限まで自動増分します。

RD エリア構成ファイルが複数ある場合，自動増分の対象となるファイルをシステム共通定義 pd_rdarea_extension_file で選択できます。**pd_rdarea_extension_file** オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

1. pd_rdarea_extension_file=last の場合

   自動増分の対象ファイルは，RD エリアを構成する HiRDB ファイルのうち，最終ファイルが対象となります。

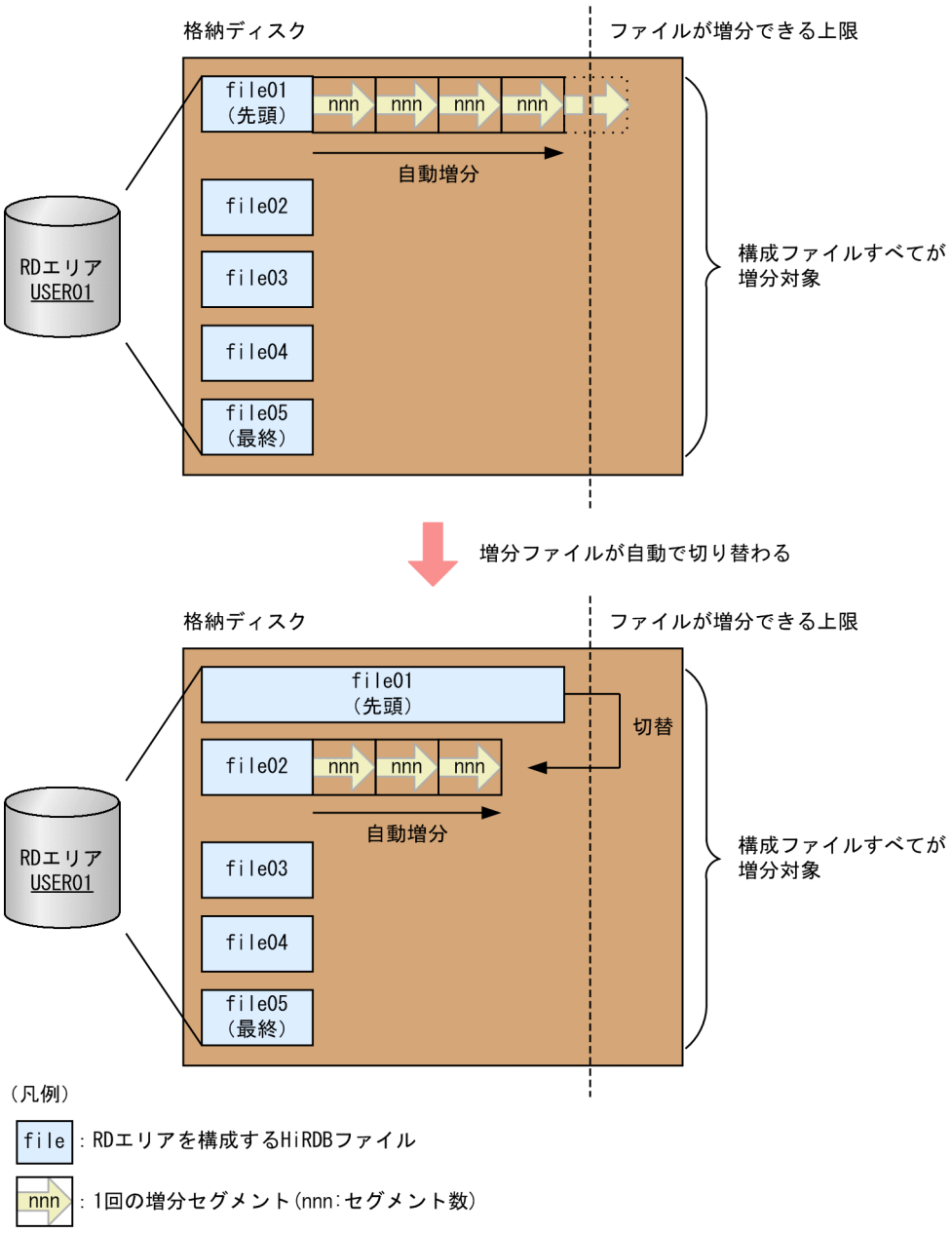
   図 15–2　pd_rdarea_extension_file=last の方式



2. pd_rdarea_extension_file=all の場合

   自動増分の対象ファイルは，RD エリアを構成する HiRDB ファイルすべてが対象となります。

図 15–3　pd_rdarea_extension_file=all の方式



(1)　自動増分を適用できる RD エリア

RD エリアの自動増分を適用できる RD エリアを次の表に示します。

表 15–5　自動増分機能を使用できる RD エリア種別と自動増分の対象 HiRDB ファイル

#	RD エリア種別		自動増分適用可否	自動増分の対象 HiRDB ファイル	
				all※	last※
1	ユーザ用 RD エリア	一時表用 RD エリア以外	○	構成ファイルすべて	最終ファイル
2		一時表用 RD エリア		最終ファイル	
3	データディクショナリ用 RD エリア				

#	RD エリア種別	自動増分適用可否	自動増分の対象 HiRDB ファイル	
			all※	last※
4	レジストリ用 RD エリア			
5	ユーザ LOB 用 RD エリア			
6	データディクショナリ LOB 用 RD エリア			
7	レジストリ LOB 用 RD エリア			
8	マスタディレクトリ用 RD エリア	×	−	−
9	データディレクトリ用 RD エリア			
10	リスト用 RD エリア			

（凡例）

○：自動増分を指定できる

×：自動増分を指定できない

−：自動増分の対象ファイルなし

注※

システム共通定義 pd_rdarea_extension_file オペランドの指定値です。

なお，プラグインが提供する抽象データ型を格納している場合，その RD エリアに対しては自動増分が適用されません。ただし，HiRDB Text Search Plug-in の SGMLTEXT 型の場合は，インデクス管理用領域の一部だけが自動増分の対象となります。

(2) 自動増分の契機

HiRDB は，次に示す RD エリアの自動増分契機になると，RD エリアを自動増分します。

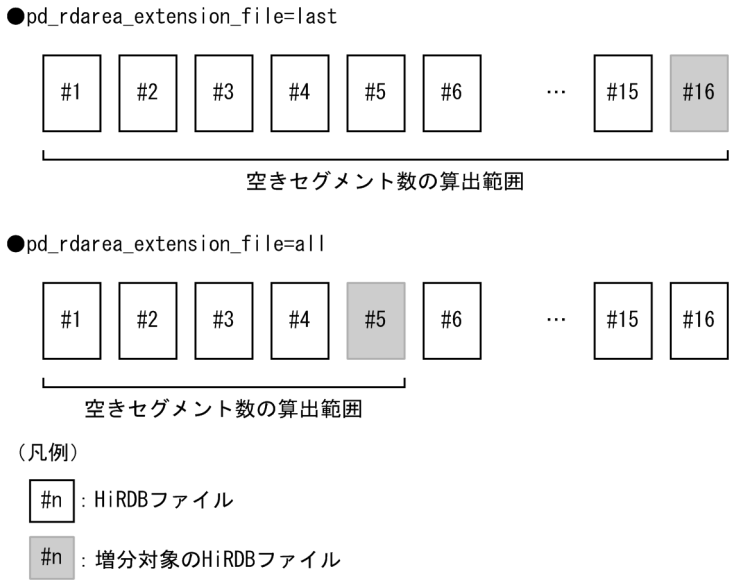
- RD エリア内の空きセグメント数が 1 回の自動増分で増分するセグメント数未満になったとき

  例えば，1 回の自動増分セグメント数を 50 セグメントとしている場合，空きセグメント数が 50 セグメント未満になったときに自動増分します。
- RD エリア内に空きセグメントがなく，新しいセグメントを確保できないとき

自動増分契機は，**pd_rdarea_extension_timing** オペランドで指定できます。詳細については，マニュアル「HiRDB システム定義」を参照してください。

自動増分契機を判断するために空きセグメント数を算出する範囲を次の図に示します。

図 15–4　空きセグメント数を算出する範囲



(3)　RD エリアの自動増分方式の選択基準

RD エリアの自動増分を適用する場合，基本的には方式 1 をお勧めします。方式 1 では，HiRDB ファイルの最大サイズ（64 ギガバイト）まで自動的に HiRDB ファイルシステム領域を拡張するため，領域の見積もりやデータベースの拡張が容易になります。

ただし，方式 1 には次の制限事項があるため，適用できないシステムの場合は，方式 2 を選択してください。

- 一つの HiRDB ファイルシステム領域に，一つの HiRDB ファイルしか格納できないため，RD エリア構成ファイル数と同じ数の HiRDB ファイルシステム領域を作成する必要がある。
- ディスクに空きがあれば，HiRDB ファイルシステム領域が 64 ギガバイトまで自動的に拡張するため，ディスク使用量の上限を設定できない。

(4)　自動増分の対象ファイルの選択基準

次に示す運用をする場合，pd_rdarea_extension_file に all を指定してください。

- RD エリア作成時に複数の HiRDB ファイル構成とし，すべての構成ファイルを自動増分させたい。

all を指定した場合，最大 1 テラバイト（64 ギガバイト［構成ファイルの上限サイズ］×16［最大構成ファイル数］）まで RD エリアを自動的に拡張できます。

ただし，all 指定には次の制限事項があるため，適用できないシステムの場合は，last を指定してください。

- すべての構成ファイル配置先に，構成ファイルが自動増分によって拡張できるディスク容量をあらかじめ用意する必要がある。

(5)　RD エリアの自動増分を適用する場合の運用

(a)　容量監視

RD エリアの自動増分を適用する場合，次に示す容量を監視するようにしてください。

- HiRDB ファイルシステム領域の容量
- HiRDB ファイルシステム領域を作成したディスクの空き容量

特に方式 1 の場合，ディスク使用量の上限を設定できないため，空き容量の監視が必要です。

また，RD エリア及び HiRDB ファイルシステム領域のサイズがむだに大きくならないように，必要に応じて RD エリア内の表及びインデクスを再編成するようにしてください。なお，pd_rdarea_warning_point オペランドの指定で，HiRDB ファイルの使用率やエクステント数を監視して，警告メッセージを出力できます。これによって，自動増分できなくなる前に対処できます。pd_rdarea_warning_point オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

容量を監視する運用ができない場合

容量を監視する運用にできない場合，HiRDB ファイルシステム領域やディスクの容量不足などによって，RD エリアの自動増分時に容量不足が発生すると障害閉塞になるおそれがあります。そのため，RD エリアの自動増分を適用しないか，次に示す条件に該当する場合は，自動拡張時に拡張する領域を初期化する設定をしてください。

- HiRDB ファイルシステム領域を自動的に拡張する方式を適用する
- raw I/O 機能を適用しない HiRDB ファイルシステム領域に RD エリアを割り当てる

自動拡張時の領域の初期化については，「自動増分時に拡張する HiRDB ファイルシステム領域の初期化」を参照してください。

(b)　自動増分対象ファイルの運用

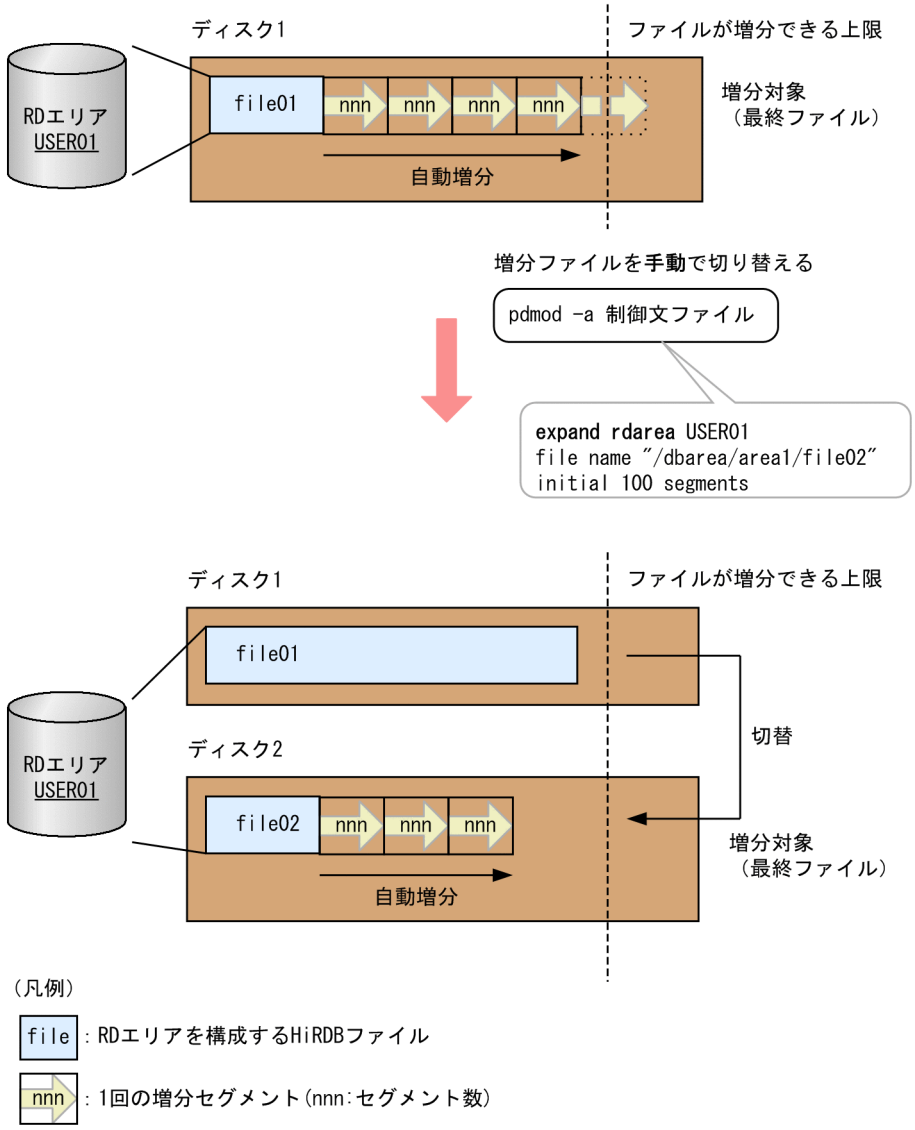
自動増分対象の RD エリアの構成ファイルは，pd_rdarea_extension_file の指定値によって次のように運用してください。

1. pd_rdarea_extension_file=last の場合

   1 つの HiRDB ファイルの構成で RD エリアを作成してください。HiRDB ファイルの容量を監視し，格納ディスクの使用できる容量がなくなる前に，RD エリア拡張（expand rdarea）で HiRDB ファイルを追加します。以降，追加したファイルが増分対象（最終ファイル）となるため，該当ファイルに対し同じ運用をします。

   この機能を適用した自動増分の流れを次の図に示します。

図 15–5　pd_rdarea_extension_file=last を選択した運用例



2. pd_rdarea_extension_file=all の場合

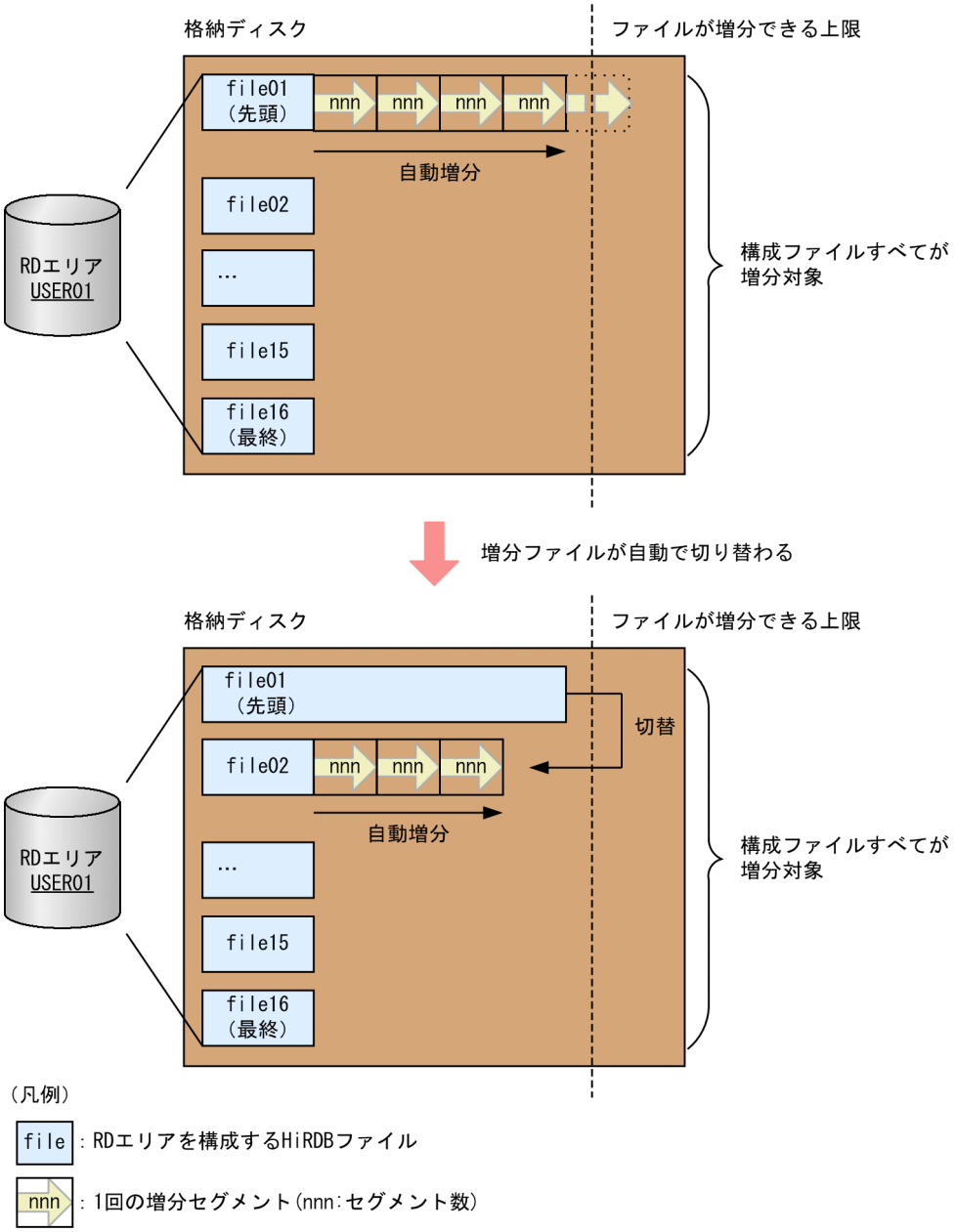
構成ファイルすべてを最小ファイルサイズで構成した RD エリアを作成します。なお，構成ファイル数を 16 未満にした場合，RD エリア拡張（expand rdarea）によって HiRDB ファイルを追加することで，後から RD エリアの最大サイズを拡張できます。

データ追加によって自動増分契機に達すると先頭の HiRDB ファイルから増分を開始し，ファイルが増分できる上限になると，次の構成ファイルの増分を開始します。

このため，HiRDB ファイルの容量監視，RD エリア拡張（expand rdarea）を行わなくても先頭ファイルから最終ファイルまで自動増分します。

この機能を適用した自動増分の流れを次の図に示します。

図 15–6　pd_rdarea_extension_file=all を選択した運用例



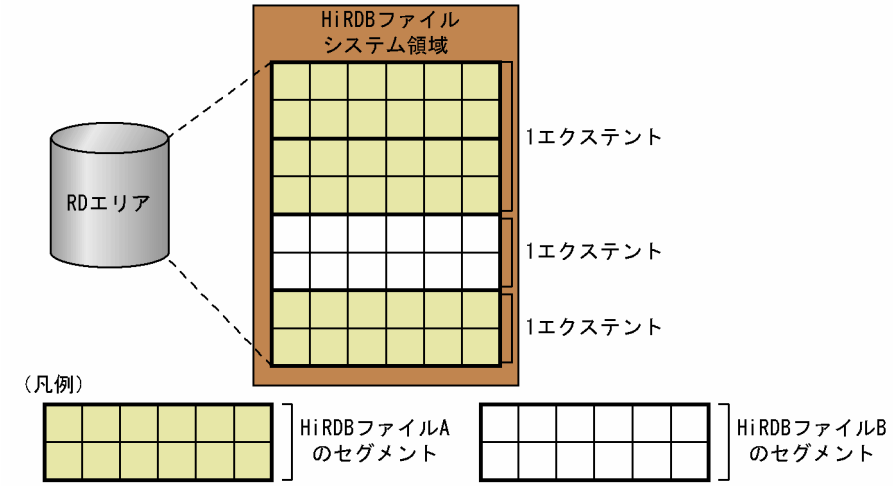
<注意事項>

pd_rdarea_warning_point オペランドを指定している場合は，自動増分する HiRDB ファイルの領域使用率通知メッセージ（KFPH22037-W），および使用エクステント数の通知メッセージ（KFPH22038-W）が，RD エリアを構成する各 HiRDB ファイルがしきい値を超えるごとに出力されます。対象ファイルが最終ファイルの場合，このメッセージの対処を行ってください。

参考

RD エリアとエクステントの関係

エクステントとは，HiRDB ファイルシステム領域内の連続した領域の固まりのことです。エクステントを次の図に示します。



１ HiRDB ファイルが保持できるエクステント数の上限は 24 です。自動増分の際に，対象となる HiRDB ファイルの最終割り当てエクステントに連続する空きが確保できた場合にはエクステント数の増加はありませんが，不連続な空きが割り当てられた場合にはエクステント数が増加します。例えば，上図では，HiRDB ファイル A のエクステント数は 2 になります。エクステントの情報については，pdfls コマンドで調べられます。

なお，RD エリアの削除・再初期化（割り当てサイズ縮小又は with reconstruction 付き）・統合では，割り当て済みエクステントの削除やサイズ縮小が発生するため，HiRDB ファイルシステム領域内に断片化した空きが発生します。この状態で RD エリアの追加・拡張・再初期化を行うと，自動増分をしていなくても 1HiRDB ファイルに，複数のエクステントが割り当てられることがあるので注意してください。

15.8.2　RD エリアの自動増分を適用する場合の留意事項

RD エリアの自動増分を適用する場合の留意事項について説明します。

(1)　増分セグメント数

一つの業務中に頻繁に自動増分が発生しないように増分セグメント数を決定してください。

(2)　自動増分の抑止

RD エリアの自動増分を適用していても，自動増分が抑止される場合があります。その要因と主な対処方法を次の表に示します。

表 15–6　RD エリアの自動増分が抑止される要因と主な対処方法

要因	主な対処方法
HiRDB ファイルシステム領域を作成したディスクの容量不足が発生しました。 pd_rdarea_extension_file=all の場合，RD エリアの構成ファイルすべてが抑止状態となるまでは RD エリアの自動増分は抑止されません。最終ファイルが抑止状態になると，RD エリアの自動増分が抑止されます。ただし，構成ファイルをすべて同じディスクに配置している場合は，自動増分できないまま最終ファイルまで切り替わるため，早急に対処を行ってください。	対処手順を次に示します。 1. ディスクの容量を確認し，ディスクの増設などを行います。 2. データベース再編成ユティリティ（pdrorg コマンド）で，表及びインデクスを再編成します。 3. データベース構成変更ユティリティ（pdmod コマンド）の expand rdarea 文で RD エリアを拡張します。 4. raw I/O 機能を適用しない HiRDB ファイルシステム領域に RD エリアを割り当てる場合，HiRDB 管理者での OS のシステム資源の制限値を，HiRDB ファイルシステム領域のサイズより大きな値又は無制限にしてください。
HiRDB ファイルシステム領域の最大増分回数（エクステント数の上限）を超えました。 pd_rdarea_extension_file=all の場合，RD エリアの構成ファイルすべてが抑止状態となるまでは RD エリアの自動増分は抑止されません。最終ファイルが抑止状態になると，RD エリアの自動増分が抑止されます。ただし，構成ファイルをすべて同じ HiRDB ファイルシステム領域に配置している場合は，自動増分できないまま最終ファイルまで切り替わるため，早急に対処を行ってください。	対処手順は「HiRDB ファイルシステム領域の増分回数が上限に達したときの対処方法」を参照してください。
HiRDB ファイルの最大増分回数（エクステント数の上限）を超えました。 pd_rdarea_extension_file=all の場合，RD エリアの構成ファイルすべてが抑止状態となるまでは RD エリアの自動増分は抑止されません。最終ファイルが抑止状態になると，RD エリアの自動増分が抑止されます。	対処手順は「HiRDB ファイルの増分回数が上限に達したときの対処方法」を参照してください。
RD エリアの状態が，更新可能バックアップ閉塞状態又は更新可能バックアップ閉塞状態（WAIT モード）になっています。※	pdrels コマンドで RD エリアの閉塞を解除してください。
RD エリアがインメモリ化されています。	インメモリデータバッファと RD エリアの同期を取ってからインメモリ化を解除してください。詳細については，マニュアル「HiRDB バッチ高速化機能」を参照してください。
ユーザ LOB 用 RD エリアの最終 HiRDB ファイルが更新凍結状態です。	対処手順を次に示します。 1. データベース構成変更ユティリティ（pdmod コマンド）の expand rdarea 文で RD エリアを拡張します。 2. pddbfrz -d コマンドを実行して，HiRDB ファイルの更新凍結状態を解除します。

注※

RD エリアの状態が，更新可能バックアップ閉塞状態又は更新可能バックアップ閉塞状態（WAIT モード）の場合，RD エリアの自動増分が抑止されます。そのため，これらの閉塞状態中に新規ページの確保が発生する大量データの追加又は更新業務の実行は避けてください。ただし，pd_rdarea_extension_timing オペランドに use を指定している場合，更新可能バックアップ閉塞状態

のため，RD エリアの自動増分が抑止されても，データ追加によるページ不足エラーになるまで，空きセグメント（増分セグメント数分）を利用できます。このため，pd_rdarea_extension_timing オペランドに nouse を指定している場合よりも業務への影響を軽減できます。

(3)　更新トランザクションの実行時間

RD エリアの自動増分中は，その RD エリアの最終 HiRDB ファイルに対して排他が掛かります。そのため，自動増分中の RD エリアにある表やインデクスを更新（データ追加も含む）するトランザクションは待ち状態になります。また，自動増分時に HiRDB ファイルシステム領域を初期化する設定（pd_rdarea_expand_format オペランドに Y を指定）をしている場合，初期化によるディスクへの入出力が発生するため，その分トランザクションの実行時間が長くなります。これらの原因によって，次に示す理由でトランザクションがキャンセルされることがあります。

- 排他待ち限界経過時間を超えた
- HiRDB クライアントの最大待ち時間やユティリティの最大実行時間を超過した

そのため，次に示す項目を監視している場合は，RD エリアの構成情報（ページサイズ，セグメントサイズ，増分セグメント数），及び HiRDB ファイルシステム領域を配置しているディスクの入出力性能から自動増分に掛かる処理時間を考慮して値を設定してください。

- 排他待ち限界経過時間（システム共通定義の pd_lck_wait_timeout オペランド）
- 「UAP 又はユティリティの実行時間の監視（無応答障害時の影響を抑える方法）」に示すオペランド及びオプション

自動増分に掛かる処理時間の目安の見積もり式（単位：秒）を次に示します。

該当する RD エリアのページサイズ×（A×増分セグメント数＋ B）÷C

A：自動増分時に HiRDB ファイルシステム領域を初期化する場合はセグメントサイズ，初期化しない場合は 0

B：自動増分時に作成するディレクトリページ数

- データディクショナリ用 RD エリア，ユーザ用 RD エリア，及びレジストリ用 RD エリアの場合
  ↑増分セグメント数÷b↑＋↑増分セグメント数÷f↑
  b：↓ページサイズ−20÷（↑セグメントサイズ÷32↑×8 ＋ 56）↓
  f：↓（125×ページサイズ）÷（16×b）↓×b
- LOB 用 RD エリアの場合
  ↑セグメントサイズ÷64000↑×96

C：HiRDB ファイルシステム領域を作成したディスクの入出力性能（1 秒当たりの入出力バイト数）

15.8.3　HiRDB ファイルシステム領域を自動的に拡張する方式

この方式の場合，RD エリアの自動増分で HiRDB ファイルシステム領域サイズの上限を超える場合，HiRDB ファイルの最大サイズ（64 ギガバイト）まで自動的に HiRDB ファイルシステム領域を拡張します。そのため，この方式の RD エリアがあるディスクの見積もりは，RD エリアの最大サイズが次のようになる点を考慮してください。

この方式の場合の RD エリアの最大サイズ

- 一つの RD エリアに一つの HiRDB ファイルの場合：64 ギガバイト
- 一つの RD エリアに複数の HiRDB ファイルの場合：

  自動増分対象ファイル以外の HiRDB ファイルの総容量＋（自動増分対象ファイル数※×64 ギガバイト）

注※

pd_rdarea_extension_file に last を指定した場合，「自動増分対象ファイル数」は 1 です。all を指定した場合，「自動増分対象ファイル以外のファイルの総容量」は 0，「自動増分対象ファイル数」は RD エリア構成ファイル数です。

(1)　自動増分の設定方法

自動増分の設定手順を次に示します。

〈手順〉

1. 必要があれば，HiRDB システム定義を変更します。

   pd_rdarea_extension_timing オペランドで自動増分契機を指定します。自動増分の契機については，「自動増分の契機」を参照してください。

   pd_rdarea_expand_format オペランドで自動増分時に拡張する HiRDB ファイルシステム領域を初期化するかどうかを指定します。初期化については，「自動増分時に拡張する HiRDB ファイルシステム領域の初期化」を参照してください。

   pd_rdarea_extension_file オペランドで自動増分の対象ファイルを指定します。自動増分の対象ファイルについては，「自動増分の対象ファイルの選択基準」を参照してください。

   HiRDB システム定義の変更方法については，「HiRDB システム定義を変更する方法」又は「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

2. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成するときに，-a オプションを指定します。このとき，-i オプションを指定して，HiRDB ファイルシステム領域を初期化することをお勧めします。
3. RD エリアを作成するときにユティリティの制御文※で増分セグメント数を指定します。

注※

データベース初期設定ユティリティ，データベース構成変更ユティリティ，又はレジストリ機能初期設定ユティリティの CREATE RDAREA，EXPAND RDAREA，INITIALIZE RDAREA，ALTER RDAREA 文で指定できます。

(2)　自動増分時に拡張する HiRDB ファイルシステム領域の初期化

raw I/O 機能を適用しない HiRDB ファイルシステム領域に RD エリアを割り当てる場合，その HiRDB ファイルシステム領域がスパースファイルになることがあります。スパースファイルは，データ・ブロックに対して割り当てられていない領域を持つファイルのことで，ファイルに書き込まれた部分の領域だけが割り当てられます。このように HiRDB ファイルシステム領域が事前に確保されないため，デファードライト処理でディスクに書き込むときに，ディスクの容量不足が発生するおそれがあります。RD エリアの自動増分時にディスクの容量不足が発生すると，その RD エリアが障害閉塞します。このとき，自動増分時に拡張する HiRDB ファイルシステム領域を初期化する設定（システム共通定義の pd_rdarea_expand_format オペランドに Y を指定）にしておくと，RD エリアは障害閉塞しません。ただし，初期化処理時にディスクへの入出力が発生するため，その分，次に示すトランザクションの処理に時間が掛かります。

- 初期化しているセグメントを更新するトランザクション
- 自動増分で新規セグメントを割り当てるトランザクション

このため，初期化しない（省略値）ことをお勧めします。初期化することをお勧めするのは，「RD エリアの自動増分を適用する場合の運用」の「容量を監視する運用ができない場合」です。

なお，自動増分時に HiRDB ファイルシステム領域を初期化する場合，遅延が発生することを考慮して増分セグメント数を決定してください。

15.8.4　HiRDB ファイルシステム領域内で HiRDB ファイルを拡張する方式

この方式の場合，HiRDB ファイルシステム領域の上限まで自動増分します。自動増分で，HiRDB ファイルシステム領域に空きがないと，自動増分できません。この場合，RD エリアを拡張するか，RD エリア内の表及びインデクスを再編成してください。また，HiRDB ファイルのエクステント数が上限値である 24 を超えた場合，HiRDB ファイルシステム領域のエクステントを統合して一つにするか，又は RD エリアを拡張してください。HiRDB ファイルシステム領域のエクステント数が上限値を超えた場合（増分回数が pdfmkfs コマンド又は pdfchfs コマンドの-e オプションで指定した増分回数の上限値を超えた場合），HiRDB ファイルシステム領域のエクステントを統合して一つにするか，pdfchfs コマンドの-e オプション指定値を変更して増分回数の上限値を大きくしてください。ただし，pdfchfs コマンドで変更できる上限値は，pdfmkfs コマンドの-e オプション指定値のため，この上限を超えるときは，HiRDB ファイルシステム領域のエクステントを統合して一つにしてください。HiRDB ファイルシステム領域を統合する手順については，「HiRDB ファイルシステム領域のエクステントを統合する」を参照してください。

(1)　自動増分の設定方法

自動増分の設定手順を次に示します。

〈手順〉

1. 必要があれば，HiRDB システム定義を変更します。

pd_rdarea_extension_timing オペランドで自動増分契機を指定します。自動増分の契機については，「自動増分の契機」を参照してください。

pd_rdarea_extension_file オペランドで自動増分の対象ファイルを指定します。自動増分の対象ファイルについては，「自動増分の対象ファイルの選択基準」を参照してください。

HiRDB システム定義の変更方法については，「HiRDB システム定義を変更する方法」又は「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

2. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成するときに，最大増分回数（-e オプション）を指定します。raw I/O 機能を使用しない場合，-i オプションを指定して，HiRDB ファイルシステム領域を初期化することをお勧めします。※1
3. RD エリアを作成するときにユティリティの制御文※2 で増分セグメント数を指定します。

注※1

-i オプションを指定して HiRDB ファイルシステム領域を初期化していない場合，データベース複写ユティリティ（pdcopy コマンド）で取得したバックアップファイルから RD エリアを回復できないことがあります。これは，HiRDB ファイルの追加やほかの RD エリアの自動増分などで HiRDB ファイルシステム領域のエクステント数が大きくなり，エクステント数の上限を超えているためです。この場合，HiRDB ファイルシステム領域のエクステントを統合し，pdfmkfs コマンドの-e オプションの指定値を見直してください。エクステントの統合については，「HiRDB ファイルシステム領域のエクステントを統合する」を参照してください。

注※2

データベース初期設定ユティリティ，データベース構成変更ユティリティ，又はレジストリ機能初期設定ユティリティの CREATE RDAREA，EXPAND RDAREA，INITIALIZE RDAREA，ALTER RDAREA 文で指定できます。

(2) HiRDB ファイルシステム領域が容量不足になったときの対処方法

HiRDB ファイルシステム領域の容量不足によって自動増分ができないときの対処方法を次に示します。

〈手順〉

1. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成します。raw I/O 機能を使用しない場合，-i オプションを指定して，HiRDB ファイルシステム領域を初期化することをお勧めします。
2. **pdrorg** コマンドで表データをアンロードします。これは推奨手順であり，必ず実行する必要はありません。実行した方が性能が向上します。アンロード方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
3. **pdmod** コマンド（expand rdarea 文）で RD エリアを拡張します。1 で作成した HiRDB ファイルシステム領域を追加割り当てしてください。RD エリアの拡張方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。
4. **pdrorg** コマンドで表データをリロードします。2 でアンロードした場合はリロードしてください。リロード方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
5. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。

6. **pdcopy** コマンドでバックアップを取得します。バックアップの取得方法については，「バックアップの取得方法」を参照してください。

(3)　HiRDB ファイルの増分回数が上限に達したときの対処方法

HiRDB ファイルのエクステントの数が上限（24 個）に達すると，それ以上自動増分ができなくなります。この場合，次に示すどれかの処置をしてください。

1. HiRDB ファイルシステム領域のエクステントを統合する
2. RD エリアを再初期化する
3. 「HiRDB ファイルシステム領域が容量不足になったときの対処方法」の方法で RD エリアを拡張する

なお，一つの HiRDB ファイルシステム領域に複数の RD エリアを割り当てている場合は，1 又は 3 の方法を選択してください。

(a)　HiRDB ファイルシステム領域のエクステントを統合する

エクステントを統合する（エクステントを一つにする）手順を次に示します。

〈手順〉

1. **pdstop** コマンドで HiRDB を正常終了します。
2. **pdfbkup** コマンドで HiRDB ファイルシステム領域のバックアップを取得します。
3. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を初期設定し直します。
4. **pdfrstr** コマンドで HiRDB ファイルシステム領域を回復します。1 で取得したバックアップから回復します。
5. **pdstart** コマンドで HiRDB を正常開始します。

(b)　RD エリアを再初期化する

RD エリアを再初期化してください。RD エリアの再初期化については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください。pdmod コマンドの initialize rdarea 文を指定するときの注意事項を次に示します。

- with reconstruction オペランドを指定してください。
- initial オペランドには必要セグメント数を指定してください。

これらのオペランドを指定しないと，RD エリアを定義したときのサイズで RD エリアが再作成されてしまいます。

(4)　HiRDB ファイルシステム領域の増分回数が上限に達したときの対処方法

HiRDB ファイルシステム領域のエクステントの数が上限に達すると，それ以上自動増分ができなくなります。この場合，次に示すどちらかの処置をしてください。

1. HiRDB ファイルシステム領域のエクステントを統合する
2. HiRDB ファイルシステム領域の増分回数の上限値を大きくする（pdfchfs コマンドで増分回数の上限値を小さくしているときだけ）

(a)　HiRDB ファイルシステム領域のエクステントを統合する

詳細は，「HiRDB ファイルシステム領域のエクステントを統合する」を参照してください。

(b)　HiRDB ファイルシステム領域の増分回数の上限値を大きくする

HiRDB ファイルシステム領域の増分回数の上限値を HiRDB 停止中に大きくする手順を次に示します。HiRDB 稼働中でも実施できます。詳細は，マニュアル「HiRDB コマンドリファレンス」の「pdfchfs（HiRDB ファイルシステム領域の管理情報変更）」を参照してください。なお，大きくできる増分回数の上限値は pdfmkfs コマンドの-e オプション指定値のため，この上限を超えて大きくする必要がある場合は，HiRDB ファイルシステム領域のエクステントを統合してください。

〈手順〉

1. **pdstop** コマンドで HiRDB を正常終了します。
2. **pdfchfs** コマンドで HiRDB ファイルシステム領域の増分回数の上限値を設定し直します。
3. **pdstart** コマンドで HiRDB を正常開始します。

15.8.5　例題（HiRDB ファイルシステム領域を自動的に拡張する方式）

自動増分機能を適用したユーザ用 RD エリア（RDAREA1）を追加します。

〈手順〉

1. HiRDB システム定義を変更します。
2. **pdfmkfs** コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します。
3. **pdmod** コマンドの制御文ファイルを作成します。
4. **pdmod** コマンドで RD エリアを追加します。
5. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
6. **pdcopy** コマンドでバックアップを取得します。
7. **pdbufmod** コマンドでグローバルバッファを割り当てます。
8. **pdbuffer** オペランドを修正します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) HiRDB システム定義を変更します

pd_rdarea_extension_timing オペランドの指定値を変更します。この例では，自動増分の契機に use を指定します。また，自動増分時に自動拡張する領域を初期化するため，pd_rdarea_expand_format オペランドを指定します。

```
                          :
set pd_rdarea_extension_timing=use
set pd_rdarea_expand_format=Y
                          :
```

HiRDB システム定義の変更方法については，「HiRDB システム定義を変更する方法」又は「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(2) pdfmkfs コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します

```
pdfmkfs -n 100 -k DB -a -i C:¥rdarea¥area01
```

〔説明〕

-a オプションを指定して，100 メガバイトの RD エリア用 HiRDB ファイルシステム領域（C:¥rdarea¥area01）を作成します。-i オプションを指定して，HiRDB ファイルシステム領域を初期化します。

(3) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **create rdarea 文**を記述した制御文ファイル（C:¥pdmod¥create01）を作成します。制御文ファイルの内容を次に示します。

```
create rdarea RDAREA1
   for user used by PUBLIC
   server name bes1
   page 4096 characters
   storage control segment 10 pages
   extension use 500 segments
   file name ”C:¥rdarea¥area01¥file01”
      initial 1000 segments;
```

〔説明〕

RD エリアの自動増分を適用するために，extension オペランドに増分セグメント数を指定します。

この例では，RD エリア（RDAREA1）に割り当てられる最終 HiRDB ファイル（C:¥rdarea¥area01¥file01）は，自動増分契機になると，500 セグメントずつ自動的に増分します。

(4) pdmod コマンドで RD エリアを追加します

```
pdmod -a C:¥pdmod¥create01
```

〔説明〕

(3)で作成した pdmod コマンドの制御文ファイル名を指定して，実行します。

(5) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(6) pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

pdmod コマンドはログを無効化するため，実行後は必ずバックアップを取得します。ここでバックアップを取得していない場合，RD エリアを pdmod コマンド実行前の状態にしか回復できません。バックアップの取得については，「バックアップの取得方法」を参照してください。

(7) 追加した RD エリアにグローバルバッファを割り当てます

```
pdbufmod -k add -a gbuf01 -r RDAREA1 -n 1000
```

〔説明〕

新規のグローバルバッファ（gbuf01）を追加して RDAREA1 に割り当てます。

この例では，pdbufmod コマンドでグローバルバッファを追加して割り当てています。

なお，pdbufmod コマンドを実行する場合は，次に示す条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入している
- pd_dbbuff_modify オペランドに Y を指定している

(8) pdbuffer オペランドを修正します

割り当てたグローバルバッファは，HiRDB を正常終了又は計画停止したときに無効になります。そのため，HiRDB を終了したときに pdbuffer オペランドの指定値を変更してください。pdbuffer オペランドの指定例を次に示します。

```
                    :
pdbuffer -a gbuf01 -r RDAREA1,RDAREA2,RDAREA3 -n 1000
                    :
```

〔説明〕

グローバルバッファ（gbuf01）に，追加した RD エリア（RDAREA1）を割り当てます。

なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定を変更できます。ただし，システム構成変更コマンドを実行する場合は HiRDB Advanced High Availability が必要になります。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.8.6 例題（HiRDB ファイルシステム領域内で HiRDB ファイルを拡張する方式）

自動増分機能を適用したユーザ用 RD エリア（RDAREA2）を追加します。

〈手順〉

1. HiRDB システム定義を変更します。
2. **pdfmkfs** コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します。
3. **pdmod** コマンドの制御文ファイルを作成します。
4. **pdmod** コマンドで RD エリアを追加します。
5. **pdlogswap -d sys -w** コマンドで，システムログファイルをスワップします。
6. **pdcopy** コマンドでバックアップを取得します。
7. **pdbufmod** コマンドでグローバルバッファを割り当てます。
8. **pdbuffer** オペランドを修正します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1) HiRDB システム定義を変更します

pd_rdarea_extension_timing オペランドの指定値を変更します。この例では，自動増分の契機に nouse を指定します。

```
                          :
set pd_rdarea_extension_timing=nouse
                          :
```

HiRDB システム定義の変更方法については，「HiRDB システム定義を変更する方法」，又は「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(2) pdfmkfs コマンドで RD エリア用の HiRDB ファイルシステム領域を作成します

```
pdfmkfs -n 100 -l 10 -e 230 -k DB -i C:¥rdarea¥area02
```

〔説明〕

-e オプションを指定して，100 メガバイトの RD エリア用 HiRDB ファイルシステム領域（C:¥rdarea¥area02）を作成します。-i オプションを指定して，HiRDB ファイルシステム領域を初期化します。

(3) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **create rdarea 文**を記述した制御文ファイル（C:¥pdmod¥create02）を作成します。制御文ファイルの内容を次に示します。

```
create rdarea RDAREA2
   for user used by PUBLIC
   server name bes1
   page 4096 characters
   storage control segment 10 pages
   extension use 300 segments
   file name ”C:¥rdarea¥area02¥file01”
      initial 500 segments
   file name ”C:¥rdarea¥area02¥file02”
      initial 500 segments;
```

〔説明〕

RD エリアの自動増分を適用するために，extension オペランドに増分セグメント数を指定します。

この例では，RD エリア（RDAREA2）に割り当てられる最終 HiRDB ファイル（C:¥rdarea¥area02¥file02）は，自動増分契機になると，300 セグメントずつ自動的に増分します。

(4) pdmod コマンドで RD エリアを追加します

```
pdmod -a C:¥pdmod¥create02
```

〔説明〕

(2)で作成した pdmod コマンドの制御文ファイル名を指定して，実行します。

(5) pdlogswap -d sys -w コマンドで，システムログファイルをスワップします

```
pdlogswap -d sys -w
```

(6)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

pdmod コマンドはログを無効化するため，実行後は必ずバックアップを取得します。ここでバックアップを取得していない場合，RD エリアを pdmod コマンド実行前の状態にしか回復できません。バックアップの取得については，「バックアップの取得方法」を参照してください。

(7)　追加した RD エリアにグローバルバッファを割り当てます

```
pdbufmod -k add -a gbuf02 -r RDAREA2 -n 1000
```

〔説明〕

新規のグローバルバッファ（gbuf02）を追加して RDAREA2 に割り当てます。

この例では，pdbufmod コマンドでグローバルバッファを追加して割り当てています。

なお，pdbufmod コマンドを実行する場合は，次に示す条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入している
- pd_dbbuff_modify オペランドに Y を指定している

(8)　pdbuffer オペランドを修正します

割り当てたグローバルバッファは HiRDB を正常終了又は計画停止したときに無効になります。そのため，HiRDB を終了したときに pdbuffer オペランドの指定値を変更してください。pdbuffer オペランドの指定例を次に示します。

```
                          :
pdbuffer -a gbuf02 -r RDAREA1,RDAREA2,RDAREA3 -n 1000
                          :
```

〔説明〕

グローバルバッファ（gbuf02）に，追加した RD エリア（RDAREA2）を割り当てます。

なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定を変更できます。ただし，システム構成変更コマンドを実行する場合は HiRDB Advanced High Availability が必要になります。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

15.9　RD エリアを移動する方法（RD エリアの移動）

実行者　HiRDB 管理者

データベース構成変更ユティリティ（**pdmod** コマンド）の **move rdarea 文**で，RD エリアをほかのバックエンドサーバに移動できます。RD エリアの移動機能は HiRDB/パラレルサーバ限定の機能です。

15.9.1　RD エリアを移動する前に

(1)　移動できる RD エリア

移動できる RD エリアを次に示します。

- ユーザ用 RD エリア
- ユーザ LOB 用 RD エリア

(2)　移動手順

RD エリアの移動手順を次に示します。なお，ここで説明する移動手順は基本的な操作手順です。システムの構成によっては若干手順が異なります。詳細については，「例題 1（新規サーバマシンのバックエンドサーバに移動する場合）」以降の例題で説明しています。

(a)　オンライン中の RD エリアの移動機能を使用しない場合(pdmod コマンドで norestart オペランドを指定しない場合)

オンライン中の RD エリアの移動機能を使用しない場合(pdmod コマンドで norestart オペランドを指定しない場合)の手順を次に示します。

〈手順〉

1. 移動先のバックエンドサーバがあるサーバマシンのメモリ所要量を見積もり直します。
2. **pdcopy** コマンドでバックアップを取得します。移動対象 RD エリア，マスタディレクトリ用 RD エリア，及びデータディクショナリ用の RD エリアのバックアップを取得する必要があります。バックアップの取得については，「バックアップの取得方法」を参照してください。
3. **pdhold** コマンドで移動対象 RD エリアを閉塞クローズ状態にします。
4. **pdmod** コマンドで RD エリアを移動します。
5. **pdstop** コマンドで HiRDB を正常終了します。RD エリアを移動した後に HiRDB を一度終了してください。この操作をしないと，RD エリアの移動後の処理を保証しません。
6. 移動対象 RD エリアの HiRDB ファイルを移動します。※
7. **pdstart** コマンドで HiRDB を正常開始します。

8. **pdcopy** コマンドでバックアップを取得します。移動対象 RD エリア，マスタディレクトリ用 RD エリア，及びデータディクショナリ用の RD エリアのバックアップを取得する必要があります。バックアップの取得については，「バックアップの取得方法」を参照してください。

注

RD エリアの移動開始から HiRDB の開始までの間（手順 4〜7 までの間）に，UAP 及びそのほかのユティリティを実行しないでください。

注※

別のサーバマシンに RD エリアを移動する場合に必要な操作です。移動対象 RD エリアの HiRDB ファイルを移動先のサーバマシンに移動する必要があります。このとき，HiRDB ファイルのパス名は移動元と同じパス名にしてください。同じパス名にできない場合は，別のディレクトリに格納してこのパスへのシンボリックリンクをしてください。

また，移動した後に移動元の HiRDB ファイルを削除する必要があります。方法については，「例題 1（新規サーバマシンのバックエンドサーバに移動する場合）」以降の例題で説明しています。

(b) オンライン中の RD エリアの移動機能を使用する場合(pdmod コマンドで norestart オペランドを指定する場合)

オンライン中の RD エリアの移動機能を使用する場合(pdmod コマンドで norestart オペランドを指定する場合)の手順を次に示します。

〈手順〉

1. 移動先のバックエンドサーバがあるサーバマシンのメモリ所要量を見積もり直します。
2. pdcopy コマンドでバックアップを取得します。移動対象 RD エリア，マスタディレクトリ用 RD エリア，及びデータディクショナリ用の RD エリアのバックアップを取得する必要があります。バックアップの取得については，「バックアップの取得方法」を参照してください。
3. pdhold コマンドで移動対象 RD エリアを閉塞クローズ状態にします。
4. pdmod コマンドで RD エリアを移動します。
5. 移動対象 RD エリアの HiRDB ファイルを移動します※。ただし，移動対象 RD エリアの HiRDB ファイルを共有ディスク上に作成してあり，移動先からも同一パス名で参照可能としてあれば，この手順は不要です。
6. pdrels コマンドで移動対象 RD エリアを閉塞なしオープン状態にします。
7. pdcopy コマンドでバックアップを取得します。移動対象 RD エリア，マスタディレクトリ用 RD エリア，及びデータディクショナリ用の RD エリアのバックアップを取得する必要があります。バックアップの取得については，「バックアップの取得方法」を参照してください。

注※

別のサーバマシンに RD エリアを移動する場合に必要な操作です。移動対象 RD エリアの HiRDB ファイルを移動先のサーバマシンに移動する必要があります。このとき，HiRDB ファイルのパス名は移動元と同じパス名にしてください。同じパス名にできない場合は，別のディレクトリに格納してこのパスへのシンボリックリンクをしてください。

また，移動した後に移動元の HiRDB ファイルを削除する必要があります。方法については，「例題 1（新規サーバマシンのバックエンドサーバに移動する場合）」以降の例題で説明しています。

(3)　注意事項

(a)　関連するすべての RD エリアを移動してください

RD エリアを移動する場合，関連するすべての RD エリアを移動する必要があります。関連するすべての RD エリアを移動しないと，データベース構成変更ユティリティ（pdmod コマンド）の実行時にエラーになります。

- 移動対象 RD エリアに格納されている表にインデクスを定義している場合は，そのインデクスを格納している RD エリアも移動する必要があります。
- 移動対象 RD エリアに格納されている表に LOB 列を定義している場合は，その LOB データを格納しているユーザ LOB 用 RD エリアも移動する必要があります。
- 移動対象 RD エリアに格納されている表を横分割している場合は，分割条件単位に表，インデクス，及び LOB 列を格納している RD エリアを移動する必要があります。

(b)　表に非分割キーインデクスが定義されている場合

移動対象 RD エリアに格納されている表に非分割キーインデクスを定義している場合は，そのインデクスの分割数が表の分割数と一致しているときだけ，RD エリアを移動できます。詳細は，マニュアル「HiRDB コマンドリファレンス」の「RD エリアの移動」を参照してください。一致していない場合，次に示す手順で RD エリアを移動します。

〈手順〉

1. **DROP INDEX** で非分割キーインデクスを削除します。
2. RD エリアを移動します。
3. **CREATE INDEX** で非分割キーインデクスを再作成します。

ただし，UNIQUE を指定した非分割キーインデクスは再作成できない場合があります。再作成できない場合については，マニュアル「HiRDB SQL リファレンス」の CREATE INDEX を参照してください。

なお，移動対象 RD エリアに定義しているインデクスが非分割キーインデクスであるかどうかは，ディクショナリ表 SQL_INDEXES の設定値によって判別できます。ディクショナリ表を検索する SQL 文を次に示します。

```
SELECT DIV_INDEX FROM MASTER.SQL_INDEXES
     WHERE TABLE_SCHEMA = 'インデクスを定義した表の所有者の認可識別子'
       AND INDEX_NAME = 'インデクス識別子'
```

検索結果が Y の場合：分割キーインデクス

検索結果が N の場合：非分割キーインデクス

(c)　表の分割キー以外に主キーを定義している場合

移動対象 RD エリアに格納されている表の分割キー以外に主キーを定義している場合は，RD エリアを移動できません。この場合，次に示す手順で RD エリアを移動します。

〈手順〉

1. **pdrorg** コマンドで表データをアンロードします。
2. **DROP TABLE** で表の定義を削除します。
3. RD エリアを移動します。
4. **CREATE TABLE** で表を定義します。
5. **pdrorg** コマンドで表データをリロードします。

(d)　表を基表とするルーチンがある場合

RD エリアを移動すると，移動対象 RD エリアに格納されている表を基表とするルーチンが無効になるため，ALTER ROUTINE でルーチンを再作成してください。

(e)　オンライン中の RD エリアの移動で，表を基表とするリストがある場合

移動対象 RD エリアに格納されている表を基表とするリストがある場合は，オンライン中に RD エリアを移動できません。この場合，次に示す手順で RD エリアを移動します。

〈手順〉

1. DROP LIST 文でリストを削除します。
2. RD エリアを移動します。
3. ASSIGN LIST 文でリストを再作成します。

(f)　系切り替えを適用した環境で RD エリアの移動を行う場合

系切り替えを適用した環境で RD エリアを移動する場合，移動した RD エリアの共有ディスクの設定を変更する必要があります。この場合，次に示す手順で RD エリアを移動します。

なお，系切り替え環境で RD エリアを移動する場合の前提条件については，マニュアル「HiRDB コマンドリファレンス」の「RD エリアの移動」を参照してください。

〈手順〉

1. RD エリアを移動します。
2. 移動した RD エリアの共有ディスクが移動先サーバの共有リソースとなるように，クラスタソフトウェアの設定を変更します。
3. OS のコマンドで移動元での共有ディスクの非活性化，及び移動先での共有ディスクの活性化をします。

15.9.2　例題 1（新規サーバマシンのバックエンドサーバに移動する場合）

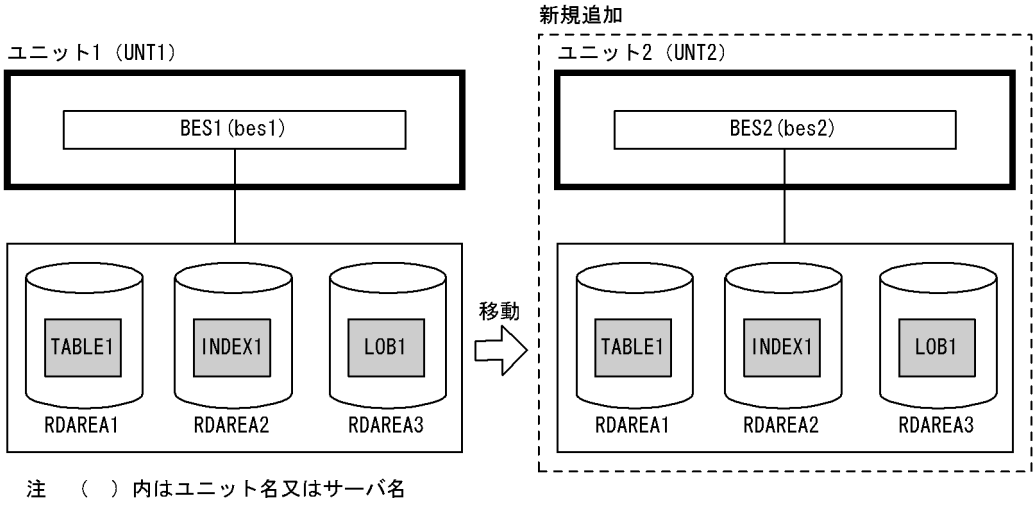
新規サーバマシンに作成したバックエンドサーバに RD エリアを移動します。

- RDAREA1 には非横分割表（TABLE1）が格納されています。
- TABLE1 にはインデクス（INDEX1）と LOB 列（LOB1）が定義されています。インデクスは RDAREA2 に，LOB データは RDAREA3 に格納されています。
- 各 RD エリアを構成する HiRDB ファイルは次のとおりです。

  RDAREA1：C:¥area1¥rdarea1

  RDAREA2：C:¥area2¥rdarea2

  RDAREA3：C:¥area2¥rdarea2



(1)　ユニット 2 を追加します

ユニットの追加については，「ユニットの追加」を参照してください。

(2)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3) pdhold コマンドで移動対象 RD エリアを閉塞クローズします

```
pdhold -r RDAREA1,RDAREA2,RDAREA3 -c
```

(4) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **move rdarea 文**を記述した制御文ファイル（C:¥pdmod¥move01）を作成します。制御文ファイルの内容を次に示します。

```
move rdarea RDAREA1,RDAREA2,RDAREA3                1
   to bes2;                                        2
```

〔説明〕

1. 移動対象の RD エリア名を指定します。
2. 移動先のサーバ名を指定します。

(5) pdmod コマンドで RD エリアを移動します

```
pdmod -a C:¥pdmod¥move01
```

〔説明〕

-a：pdmod コマンドの制御文ファイル名を指定します。

(6) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(7) 移動対象 RD エリアの HiRDB ファイルを移動します

移動対象 RD エリアの HiRDB ファイルを，新規追加したサーバマシンに次に示すどちらかの方法で移動します。

(a) HiRDB ファイルシステム領域単位で移動する場合

この方法は次に示す前提条件があります。

- 移動対象の HiRDB ファイルだけが HiRDB ファイルシステム領域内にある

OS の rcp 又は ftp コマンドで HiRDB ファイルシステム領域を移動します。

```
rcp C:¥area1 host2:C:¥area1
rcp C:¥area2 host2:C:¥area2
rcp C:¥area3 host2:C:¥area3
```

〔説明〕

RDAREA1～RDAREA3 の HiRDB ファイルシステム領域を移動します。

(b)　HiRDB ファイル単位で移動する場合

(a)の方法ができない場合は，次に示す手順で HiRDB ファイルを移動します。

●移動元での操作手順

1. **pdfbkup** コマンドで移動対象 HiRDB ファイルのバックアップを取得します。
2. **rcp** 又は **ftp** コマンドで HiRDB ファイルのバックアップを移動します。
3. **pdfrm** コマンドで移動対象 HiRDB ファイルを削除します。

（例）

```
pdfbkup C:¥area1¥rdarea1 C:¥tmp¥bk_rdarea1      1
pdfbkup C:¥area2¥rdarea2 C:¥tmp¥bk_rdarea2      1
pdfbkup C:¥area3¥rdarea3 C:¥tmp¥bk_rdarea3      1
rcp C:¥tmp¥"bk_*" host2:C:¥tmp¥                 2
pdfrm C:¥area1¥rdarea1                          3
pdfrm C:¥area2¥rdarea2                          3
pdfrm C:¥area3¥rdarea3                          3
```

〔説明〕

1. RDAREA1～RDAREA3 の HiRDB ファイルのバックアップを取得します。
2. HiRDB ファイルのバックアップを移動します。
3. RDAREA1～RDAREA3 の HiRDB ファイルを削除します。

●移動先での操作手順

1. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成します。
2. **pdfrstr** コマンドで移動対象 HiRDB ファイルをリストアします。

（例）

```
pdfmkfs -n 30 -l 10 -k DB C:¥area1              1
pdfmkfs -n 30 -l 10 -k DB C:¥area2              1
pdfmkfs -n 30 -l 10 -k DB C:¥area3              1
pdfrstr C:¥tmp¥bk_rdarea1 C:¥area1              2
pdfrstr C:¥tmp¥bk_rdarea2 C:¥area2              2
pdfrstr C:¥tmp¥bk_rdarea3 C:¥area3              2
```

〔説明〕

1. RDAREA1～RDAREA3 の HiRDB ファイルシステム領域を作成します。移動元と同じパス名にしてください。同じパス名にできない場合は，別のディレクトリに作成してこのパスへのシンボリックリンクをしてください。
2. RDAREA1～RDAREA3 の HiRDB ファイルをリストアします。

(8)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(9)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

15.9.3　例題 2（別ユニットにあるバックエンドサーバに移動する場合）

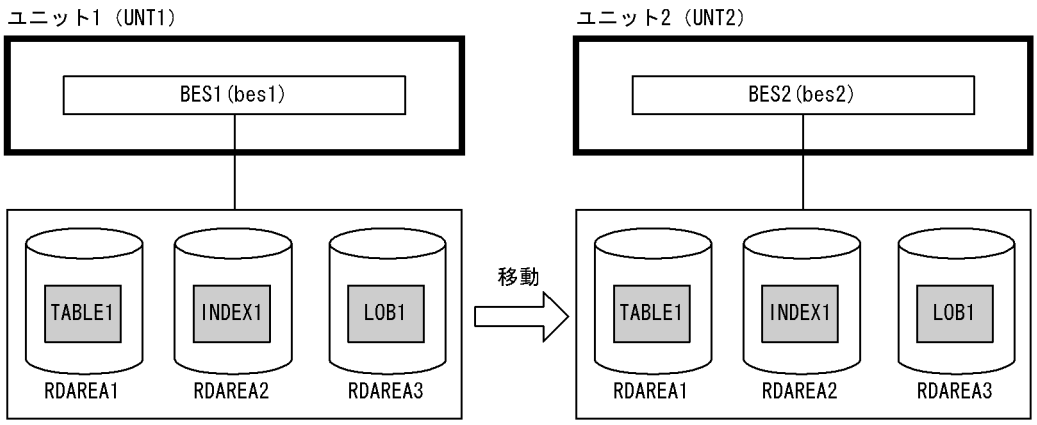
別ユニットにあるバックエンドサーバに RD エリアを移動します。

- RDAREA1 には非横分割表（TABLE1）が格納されています。
- TABLE1 にはインデクス（INDEX1）と LOB 列（LOB1）が定義されています。インデクスは RDAREA2 に，LOB データは RDAREA3 に格納されています。
- 各 RD エリアを構成する HiRDB ファイルは次のとおりです。

  RDAREA1：C:¥area1¥rdarea1

  RDAREA2：C:¥area2¥rdarea2

  RDAREA3：C:¥area3¥rdarea3



注　（　）内はユニット名又はサーバ名

(1)　メモリ所要量の見積もり

移動先のユニットがあるサーバマシンで必要なメモリ所要量を見積もってください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

これ以降の操作は「pdcopy コマンドでバックアップを取得します」以降の操作と同じです。

15.9.4　例題 3（同じユニットにあるバックエンドサーバに移動する場合）

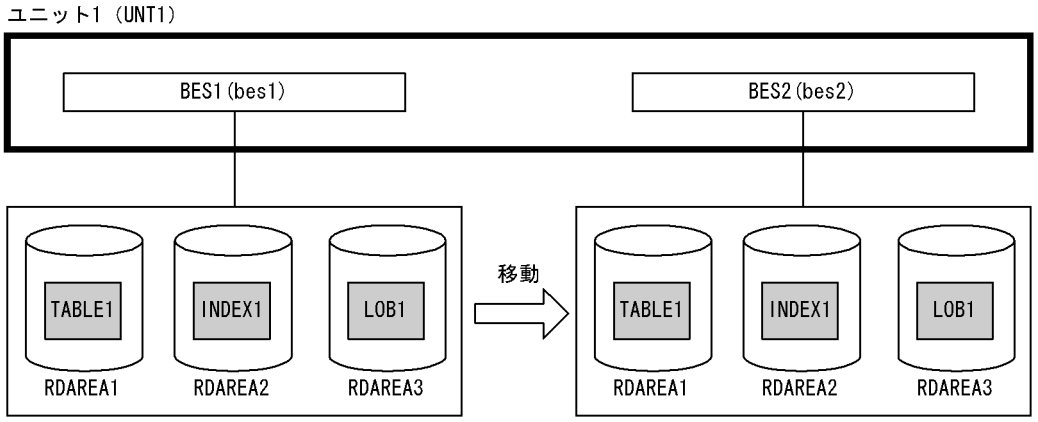
同じユニットにあるバックエンドサーバに RD エリアを移動します。

- RDAREA1 には非横分割表（TABLE1）が格納されています。
- TABLE1 にはインデクス（INDEX1）と LOB 列（LOB1）が定義されています。インデクスは RDAREA2 に，LOB データは RDAREA3 に格納されています。
- 各 RD エリアを構成する HiRDB ファイルは次のとおりです。

  RDAREA1：C:¥area1¥rdarea1

  RDAREA2：C:¥area2¥rdarea2

  RDAREA3：C:¥area3¥rdarea3



注　（　）内はユニット名又はサーバ名

(1)　メモリ所要量の見積もり

移動先のバックエンドサーバで必要なメモリ所要量を見積もってください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3)　pdhold コマンドで移動対象 RD エリアを閉塞クローズします

```
pdhold -r RDAREA1,RDAREA2,RDAREA3 -c
```

(4)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **move rdarea 文**を記述した制御文ファイル（C:¥pdmod¥move01）を作成します。制御文ファイルの内容を次に示します。

```
move rdarea RDAREA1,RDAREA2,RDAREA3                1
   to bes2;                                        2
```

〔説明〕

1. 移動対象の RD エリア名を指定します。
2. 移動先のサーバ名を指定します。

(5)　pdmod コマンドで RD エリアを移動します

```
pdmod -a C:¥pdmod¥move01
```

〔説明〕

-a：pdmod コマンドの制御文ファイル名を指定します。

(6)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(7)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(8)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

15.9.5　例題 4（横分割表を格納した RD エリアを移動する場合）

新規サーバマシンに作成したバックエンドサーバに RD エリアを移動します。サーバ内横分割していた表（TABLE1）をサーバ間横分割します。

- RDAREA1 及び RDAREA2 には横分割表（TABLE1）が格納されています。

- TABLE1 には分割キーインデクス（INDEX1）と非分割キーインデクス（INDEX2）が定義されています。また，LOB 列（LOB1）が定義されています。INDEX1 は RDAREA3 及び RDAREA4 に，INDEX2 は RDAREA3 に，LOB データは RDAREA5 及び RDAREA6 に格納されています。
- 各 RD エリアを構成する HiRDB ファイルは次のとおりです。

  RDAREA1：C:¥area1¥rdarea1

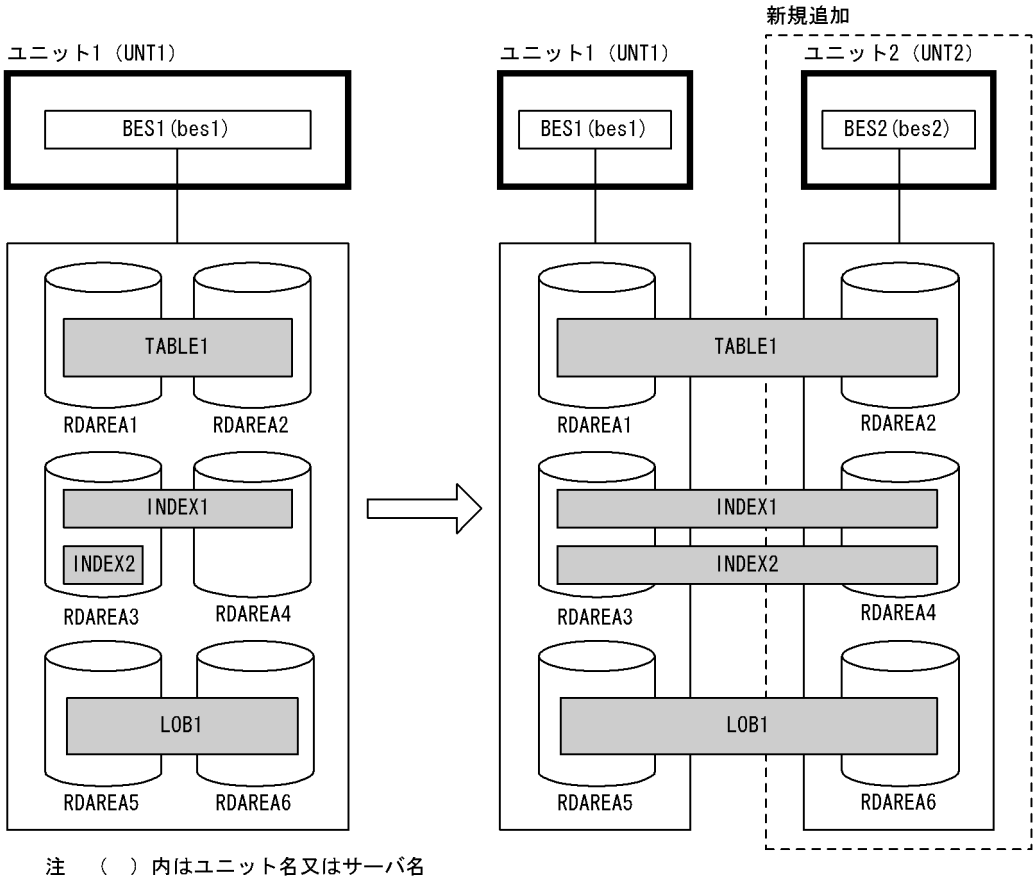
  RDAREA2：C:¥area2¥rdarea2

  RDAREA3：C:¥area3¥rdarea3

  RDAREA4：C:¥area4¥rdarea4

  RDAREA5：C:¥area5¥rdarea5

  RDAREA6：C:¥area6¥rdarea6



注　（　）内はユニット名又はサーバ名

(1)　ユニット 2 を追加します

ユニットの追加については，「ユニットの追加」を参照してください。

(2)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3) DROP INDEX で非分割キーインデクスを削除します

```
DROP INDEX INDEX2;
```

インデクスの分割数が表の分割数と一致していない非分割キーインデクスを格納している RD エリアは移動できないため，DROP INDEX で非分割キーインデクス（INDEX2）を削除します。

(4) pdhold コマンドで移動対象 RD エリアを閉塞クローズします

```
pdhold -r RDAREA2,RDAREA4,RDAREA6 -c
```

(5) pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **move rdarea 文**を記述した制御文ファイル（C:¥pdmod¥move01）を作成します。制御文ファイルの内容を次に示します。

```
move rdarea RDAREA2,RDAREA4,RDAREA6                    1
   to bes2;                                            2
```

〔説明〕

1. 移動対象の RD エリア名を指定します。
2. 移動先のサーバ名を指定します。

(6) pdmod コマンドで RD エリアを移動します

```
pdmod -a C:¥pdmod¥move01
```

〔説明〕

-a：pdmod コマンドの制御文ファイル名を指定します。

(7)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(8)　移動対象 RD エリアの HiRDB ファイルを移動します

移動対象 RD エリアの HiRDB ファイルを，新規追加したサーバマシンに次に示すどちらかの方法で移動します。

(a)　HiRDB ファイルシステム領域単位で移動する場合

この方法は次に示す前提条件があります。

- 移動対象の HiRDB ファイルだけが HiRDB ファイルシステム領域内にある

OS の rcp 又は ftp コマンドで HiRDB ファイルシステム領域を移動します。

```
rcp C:¥area2 host2:C:¥area2
rcp C:¥area4 host2:C:¥area4
rcp C:¥area6 host2:C:¥area6
```

〔説明〕

RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルシステム領域を移動します。

(b)　HiRDB ファイル単位で移動する場合

(a)の方法ができない場合は，次に示す手順で HiRDB ファイルを移動します。

●移動元での操作手順

1. **pdfbkup** コマンドで移動対象 HiRDB ファイルのバックアップを取得します。
2. **rcp** 又は **ftp** コマンドで HiRDB ファイルのバックアップを移動します。
3. **pdfrm** コマンドで移動対象 HiRDB ファイルを削除します。

（例）

```
pdfbkup C:¥area2¥rdarea2 C:¥tmp¥bk_rdarea2      1
pdfbkup C:¥area4¥rdarea4 C:¥tmp¥bk_rdarea4      1
pdfbkup C:¥area6¥rdarea6 C:¥tmp¥bk_rdarea6      1
rcp C:¥tmp¥"bk_*" host2:C:¥tmp¥                 2
pdfrm C:¥area2¥rdarea2                          3
pdfrm C:¥area4¥rdarea4                          3
pdfrm C:¥area6¥rdarea6                          3
```

〔説明〕

1. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルのバックアップを取得します。
2. HiRDB ファイルのバックアップを移動します。

3. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルを削除します。

●移動先での操作手順

1. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成します。
2. **pdfrstr** コマンドで移動対象 HiRDB ファイルをリストアします。

（例）

```
pdfmkfs -n 30 -l 10 -k DB C:¥area2            1
pdfmkfs -n 30 -l 10 -k DB C:¥area4            1
pdfmkfs -n 30 -l 10 -k DB C:¥area6            1
pdfrstr C:¥tmp¥bk_rdarea2 C:¥area2            2
pdfrstr C:¥tmp¥bk_rdarea4 C:¥area4            2
pdfrstr C:¥tmp¥bk_rdarea6 C:¥area6            2
```

〔説明〕

1. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルシステム領域を作成します。移動元と同じパス名にしてください。同じパス名にできない場合は，別のディレクトリに作成してこのパスへのシンボリックリンクをしてください。
2. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルをリストアします。

(9)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(10)　CREATE INDEX で非分割キーインデクスを再作成します

```
CREATE INDEX INDEX2 on TABLE1(C2) IN ((RDAREA3),(RDAREA4));
```

(3)で削除した非分割キーインデクスを再作成してください。

(11)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

15.9.6　例題 5（抽象データ型を格納した RD エリアを移動する場合）

新規サーバマシンに作成したバックエンドサーバに抽象データ型（SGMLTEXT 型）を格納した RD エリアを移動します。サーバ内横分割していた表（TABLE1）をサーバ間横分割します。

- RDAREA1 及び RDAREA2 には横分割表（TABLE1）が格納されています。
- TABLE1 には抽象データ型（SGMLTEXT 型）が定義されています。また，プラグインインデクスが定義されています。SGMLTEXT データは RDAREA3 及び RDAREA4 に，プラグインインデクスは RDAREA5 及び RDAREA6 に格納されています。
- 各 RD エリアを構成する HiRDB ファイルは次のとおりです。

  RDAREA1：C:¥area1¥rdarea1

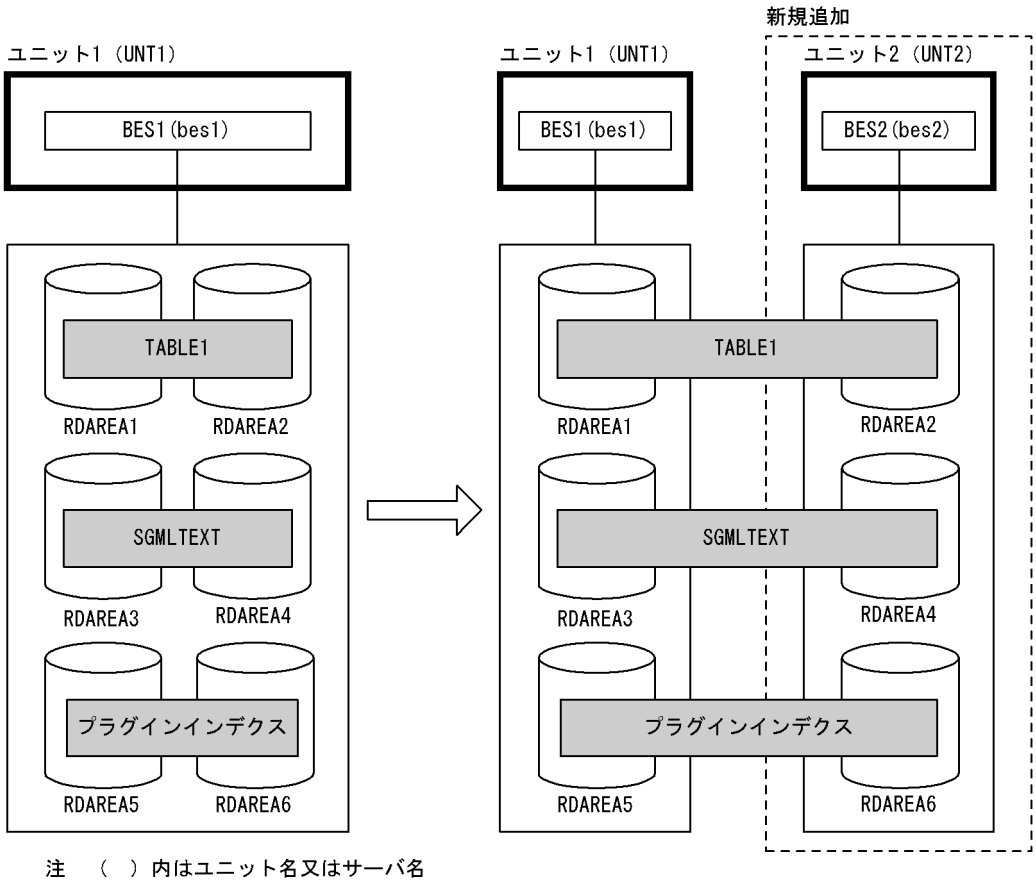
  RDAREA2：C:¥area2¥rdarea2

  RDAREA3：C:¥area3¥rdarea3

  RDAREA4：C:¥area4¥rdarea4

  RDAREA5：C:¥area5¥rdarea5

  RDAREA6：C:¥area6¥rdarea6



注　（　）内はユニット名又はサーバ名

(1)　ユニット 2 を追加します

ユニットの追加については，「ユニットの追加」を参照してください。

(2)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01 -p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3)　pdhold コマンドで移動対象 RD エリアを閉塞クローズします

```
pdhold -r RDAREA2,RDAREA4,RDAREA6 -c
```

(4)　pdmod コマンドの制御文ファイルを作成します

pdmod コマンドの **move rdarea 文**を記述した制御文ファイル（C:¥pdmod¥move01）を作成します。制御文ファイルの内容を次に示します。

```
move rdarea RDAREA2,RDAREA4,RDAREA6                    1
   to bes2;                                            2
```

〔説明〕

1. 移動対象の RD エリア名を指定します。
2. 移動先のサーバ名を指定します。

(5)　pdmod コマンドで RD エリアを移動します

```
pdmod -a C:¥pdmod¥move01
```

〔説明〕

-a：pdmod コマンドの制御文ファイル名を指定します。

(6)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(7)　移動対象 RD エリアの HiRDB ファイルを移動します

移動対象 RD エリアの HiRDB ファイルを，新規追加したサーバマシンに次に示すどちらかの方法で移動します。

(a)　HiRDB ファイルシステム領域単位で移動する場合

この方法は次に示す前提条件があります。

- 移動対象の HiRDB ファイルだけが HiRDB ファイルシステム領域内にある

OS の rcp 又は ftp コマンドで HiRDB ファイルシステム領域を移動します。

```
rcp C:¥area2 host2:C:¥area2
rcp C:¥area4 host2:C:¥area4
rcp C:¥area6 host2:C:¥area6
```

〔説明〕

RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルシステム領域を移動します。

(b)　HiRDB ファイル単位で移動する場合

(a)の方法ができない場合は，次に示す手順で HiRDB ファイルを移動します。

●移動元での操作手順

1. **pdfbkup** コマンドで移動対象 HiRDB ファイルのバックアップを取得します。
2. **rcp** 又は **ftp** コマンドで HiRDB ファイルのバックアップを移動します。
3. **pdfrm** コマンドで移動対象 HiRDB ファイルを削除します。

（例）

```
pdfbkup C:¥area2¥rdarea2 C:¥tmp¥bk_rdarea2      1
pdfbkup C:¥area4¥rdarea4 C:¥tmp¥bk_rdarea4      1
pdfbkup C:¥area6¥rdarea6 C:¥tmp¥bk_rdarea6      1
rcp C:¥tmp¥"bk_*" host2:C:¥tmp¥                 2
pdfrm C:¥area2¥rdarea2                          3
pdfrm C:¥area4¥rdarea4                          3
pdfrm C:¥area6¥rdarea6                          3
```

〔説明〕

1. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルのバックアップを取得します。
2. HiRDB ファイルのバックアップを移動します。

3. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルを削除します。

●移動先での操作手順

1. **pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成します。
2. **pdfrstr** コマンドで移動対象 HiRDB ファイルをリストアします。

（例）

```
pdfmkfs -n 30 -l 10 -k DB C:¥area2            1
pdfmkfs -n 30 -l 10 -k DB C:¥area4            1
pdfmkfs -n 30 -l 10 -k DB C:¥area6            1
pdfrstr C:¥tmp¥bk_rdarea2 C:¥area2            2
pdfrstr C:¥tmp¥bk_rdarea4 C:¥area4            2
pdfrstr C:¥tmp¥bk_rdarea6 C:¥area6            2
```

〔説明〕

1. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルシステム領域を作成します。移動元と同じパス名にしてください。同じパス名にできない場合は，別のディレクトリに作成してこのパスへのシンボリックリンクをしてください。
2. RDAREA2，RDAREA4，RDAREA6 の HiRDB ファイルをリストアします。

(8)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(9)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup02 -p C:¥pdcopy¥list02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

15.9.7　例題 6（オンライン中の RD エリア移動機能で共有ディスク上に HiRDB ファイルがある RD エリアを移動する場合）

別ユニットにあるバックエンドサーバに RD エリアを移動します。

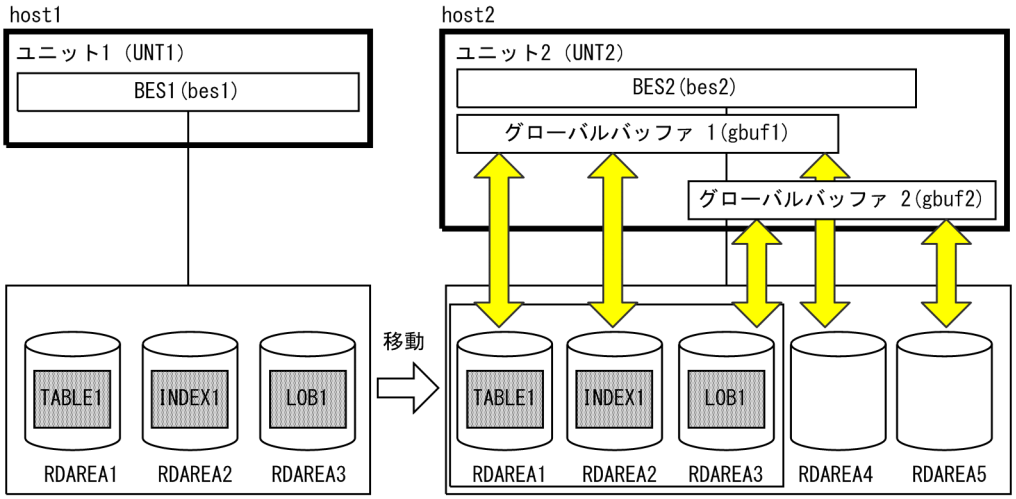
- RDAREA1 には非横分割表（TABLE1）が格納されています。
- TABLE1 にはインデクス（INDEX1）と LOB 列（LOB1）が定義されています。インデクスは RDAREA2 に，LOB データは RDAREA3 に格納されています。
- BES2 には RDAREA1，RDAREA2 に割り当て可能なグローバルバッファ（gbuf1），及び RDAREA3 に割り当て可能なグローバルバッファ（gbuf2)があります。
- 各 RD エリアを構成する HiRDB ファイルは次のとおりです。

  RDAREA1：L:¥area1¥rdarea1_1

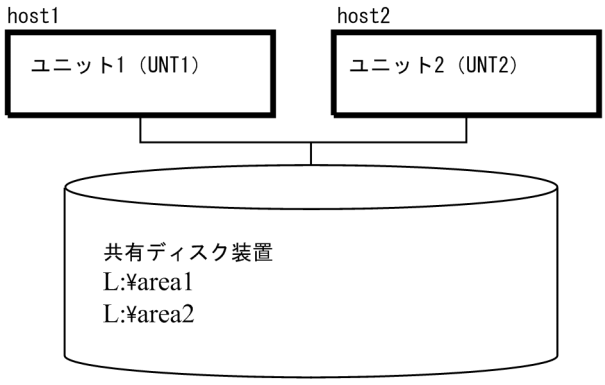
  RDAREA2：L:¥area1¥rdarea1_2

  RDAREA3：L:¥area2¥rdarea2_3

【RDエリア構成図（論理）】



【ファイル構成図（物理）】



注　（　）内はユニット名又はサーバ名

(1)　メモリ所要量の見積もり

移動先のバックエンドサーバで必要なメモリ所要量を見積もってください。メモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m D:¥rdarea¥mast¥mast01 -M r -a -b D:¥pdcopy¥backup01 -p D:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(3)　pdhold コマンドで移動対象 RD エリアを閉塞クローズします

```
pdhold -r RDAREA1,RDAREA2,RDAREA3 -c
```

(4)　pdmod コマンドで RD エリアを移動します

```
pdmod -a D:¥pdmod¥move01
```

〔説明〕

-a：pdmod コマンドの制御文ファイル名を指定します。

制御文ファイル（D:¥pdmod¥move01）の内容を次に示します。

```
move rdarea norestart                              1
   RDAREA1 globalbuffer gbuf1,                     2
   RDAREA2 globalbuffer gbuf1,
   RDAREA3 globalbuffer gbuf2
   to bes2;                                        3
```

〔説明〕

1. オンライン中の RD エリア移動機能を指定します。
2. 移動対象の RD エリア名，及び RD エリアに割り当てるグローバルバッファを指定します。
3. 移動先のサーバ名を指定します。

(5)　pdrels コマンドで移動対象 RD エリアを閉塞解除オープンします

```
pdrels -r RDAREA1,RDAREA2,RDAREA3 -o
```

(6)　pdcopy コマンドでバックアップを取得します

```
pdcopy -m D:¥rdarea¥mast¥mast01 -M r -a -b D:¥pdcopy¥backup02 -p D:¥pdcopy¥list02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。RD エリアを移動すると，マスタディレクトリ用 RD エリア及びデータディクショナリ用 RD エリアが更新されるため，全 RD エリアのバックアップを取得します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

15.10　RD エリアを格納している HiRDB ファイルシステム領域を移動する方法

実行者　HiRDB 管理者

データベースに格納するデータの増加に伴い，データベース容量が満杯になり，RD エリアを配置するディスクの変更が必要になる場合があります。ディスクの変更に当たり，RD エリアを別のディスクに移動する必要がありますが，通常，RD エリアを移動する場合は対象の RD エリアをすべて再作成する必要があります。

再作成しないで HiRDB ファイルシステム領域を別のディスクに移動するには，次の方法があります。

1. パス名を変更しないで移動する方法

   パス名を変更しないで別ディスクに移動するには，シンボリックリンクを使用します。移動元には移動先の HiRDB ファイルシステム領域を参照するシンボリックリンクを作成することによって，RD エリアの再作成を回避できます。また，HiRDB で指定するファイルシステム領域名（定義，制御文，コマンドの引数，及びバッチファイルでの指定など）の変更は不要になります。

   HiRDB ファイルシステム領域に対してシンボリックリンクを作成する方法，及び適用条件については，マニュアル「HiRDB コマンドリファレンス」の「pdfmkfs（HiRDB ファイルシステム領域の初期設定）」を参照してください。

2. パス名を変更して移動する方法

   別ディスクに移動して，HiRDB が管理しているパス名称も変更するには，ファイルパス変更コマンドを使用します。HiRDB で指定するファイルシステム領域名の変更が必要ですが，複数のマシンで動作していた HiRDB を一つのマシンに集約する場合などで，ファイルパス名称が重なるときに使用します。

   格納している RD エリアの種類によって，それぞれの方法で HiRDB ファイルシステム領域を移動できるかどうか異なります。移動できる HiRDB ファイルシステム領域を次の表に示します。

   表 15–7　移動できる HiRDB ファイルシステム領域

格納している RD エリアの種類	HiRDB の稼働状態			
	稼働中		停止中	
	シンボリックリンク	ファイルパス変更コマンド	シンボリックリンク	ファイルパス変更コマンド
マスタディレクトリ用 RD エリア	×	×	○	○
データディレクトリ用 RD エリア	×	×	○	○
データディクショナリ用 RD エリア	×	×	○	○
データディクショナリ LOB 用 RD エリア	×	×	○	○
ユーザ用 RD エリア	○	×	○	○
ユーザ LOB 用 RD エリア	○	×	○	○
レジストリ用 RD エリア	○	×	○	○

格納している RD エリアの種類	HiRDB の稼働状態			
	稼働中		停止中	
	シンボリックリンク	ファイルパス変更コマンド	シンボリックリンク	ファイルパス変更コマンド
レジストリ LOB 用 RD エリア	○	×	○	○
リスト用 RD エリア	○	×	○	○

（凡例）

○：HiRDB ファイルシステム領域を移動できます。

×：HiRDB ファイルシステム領域を移動できません。

15.10.1　シンボリックリンクを使用した移動手順

シンボリックリンクを使用して HiRDB ファイルシステム領域の移動手順を示します。

(1)　移動手順

HiRDB が稼働中の場合の移動手順を次に示します。HiRDB が停止中の場合，手順 1，手順 2，及び手順 6 の操作は不要です。

〈手順〉

1. 移動する HiRDB ファイルシステム領域に格納している RD エリアをすべて閉塞，クローズ状態にします。
2. 移動する HiRDB ファイルシステム領域を HiRDB の常駐プロセスから切り離すため，ユーザサーバプロセスをリフレッシュします。
3. 移動元の RD エリアを構成する HiRDB ファイルシステム領域を移動先のディスクにコピーします。コピーした HiRDB ファイルシステム領域名の長さ（フルパス名）は，HiRDB ファイルシステム領域名としての最大長を超えないようにしてください（詳細は，マニュアル「HiRDB コマンドリファレンス」の「pdfmkfs（HiRDB ファイルシステム領域の初期設定）」を参照してください）。超えた場合，手順 4 で行う移動先の実体パス名を指定した pdfstatfs コマンドは KFPI21574-E を出力しエラー終了します。
4. 移動元の RD エリアを構成する HiRDB ファイルシステム領域を削除します。
5. コピー先の HiRDB ファイルシステム領域に対して，移動元の HiRDB ファイルシステム領域と同一の名前でシンボリックリンクを作成します。
6. 閉塞，クローズ状態にした RD エリアを閉塞解除し，オープン状態にします。

(2)　注意事項

1. シンボリックリンクの作成，削除，及び管理（リンク先の実体存在の確認や，アクセス確認等）は，適切に行ってください。HiRDB がリンク先の HiRDB ファイルシステム領域へのアクセスに失敗すると，RD エリアは閉塞します。
2. シンボリックリンク先の実体（HiRDB ファイルシステム領域）を削除する場合は，実体のパス名を指定してください。シンボリックリンクを指定した場合は，実体ではなくシンボリックリンクが削除されます。
3. RD エリアの移動によってシンボリックリンクを作成した RD エリアを，再度別ディスクに配置する場合，「移動手順」の手順 3 でコピー時に指定する移動元には実体のパス名を指定してください。シンボリックリンクを指定した場合は，実体ではなくシンボリックリンクをコピーします。

15.10.2　ファイルパス変更コマンドを使用した移動手順

ファイルパス変更コマンドを使用した移動の手順を次に示します。

(1)　ユニットの変更のない移動

1. 移動する RD エリア及びシステム用 RD エリアのバックアップを取得します。
2. HiRDB に接続するシステムを停止します。
3. HiRDB を正常停止します。HiRDB/パラレルサーバの場合，ディクショナリサーバのユニット，及び移動する HiRDB のユニットが正常停止したことを確認します。
4. 移動元の RD エリアを構成する HiRDB ファイルシステム領域を，移動先のディスクに移動します。
5. マスタディレクトリ用 RD エリアの先頭 HiRDB ファイルを移動する場合は，システム定義 pd_master_file_name オペランドを，移動後の HiRDB ファイル名称に変更します。
6. pdchpathf コマンドを実行します。
7. HiRDB を開始します。
8. HiRDB の開始完了後，pdchpathn コマンドを実行します。
9. HiRDB の全 RD エリアを対象としたバックアップを取得します。
10. HiRDB に接続するシステムを開始します。

(2)　ユニットを変更する移動

1. 移動前の環境でバックアップを取得します。
2. 新しい環境で HiRDB をセットアップし，システムファイルの作成，及びシステム定義を新環境用に変更します。マスタディレクトリ用 RD エリアの先頭 HiRDB ファイルのパスが変わる場合は，システム定義 pd_master_file_name オペランドも，移動後の HiRDB ファイル名称に変更します。

3. 旧環境の，全ユニットが正常停止した状態の RD エリアを構成する HiRDB ファイルシステム領域のディスクを，新環境に移動します。
4. pdchpathf コマンドを実行します。
5. HiRDB を開始します。
6. HiRDB の開始完了後，pdchpathn コマンドを実行します。
7. HiRDB の全 RD エリアを対象としたバックアップを取得します。

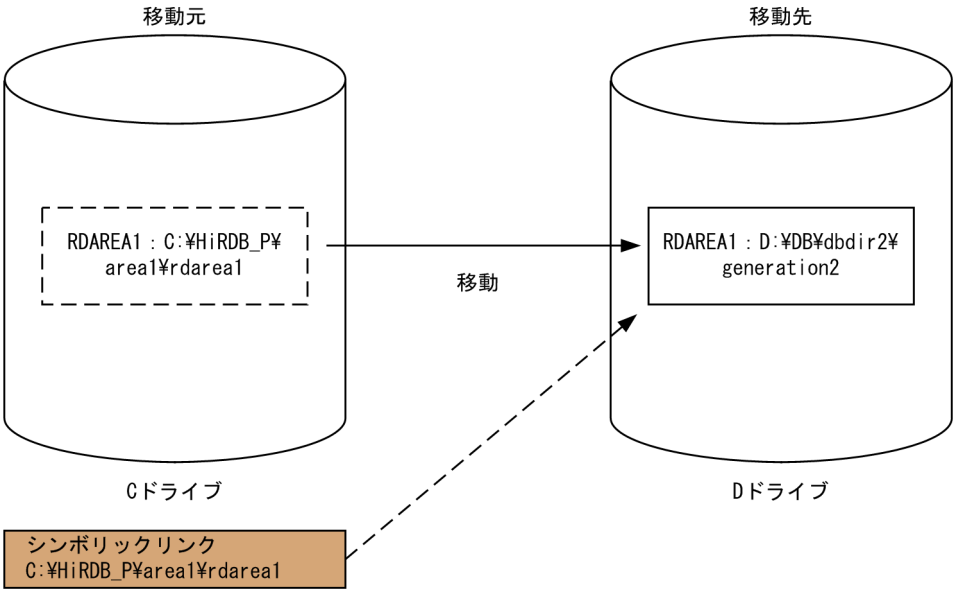
(3) 注意事項

1. 移動した HiRDB ファイルシステム領域名の長さ（フルパス）は，HiRDB ファイルシステム領域に作成している最大の HiRDB ファイル名称長と合わせて，最大長（167 文字）を超えないようにしてください。
2. 系切り替え環境の場合，変更するユニットや，ディスクにアクセスするために，IP アドレスの設定及びディスクの有効化をしておく必要があります。
3. pdchpathf コマンドが I/O エラーなどで更新中にエラーになった場合，(1)又は(2)のバックアップから回復し，やり直す必要があります。(1)又は(2)のバックアップの回復は，移動前のパスに対して行われるため，HiRDB ファイルシステム領域を移動前のパスに戻してから回復してください。制御文の指定内容に不備があり，整合性チェックでエラーになった場合は，エラー要因を取り除くと再実行できます。
4. pdchpathn コマンドの実行が完了する前に，業務 AP や pddbls コマンド以外の運用コマンド・ユティリティなどを実行しないでください。
5. 新しく HiRDB をセットアップする場合，運用ディレクトリのパスも変更できます。
6. OS 又は HiRDB のバージョンが異なる HiRDB ファイルシステム領域の移動はできません。

15.10.3 シンボリックリンクを使用した移動の例題

C ドライブに格納している HiRDB ファイルシステム領域（C:¥HiRDB_P¥area1¥rdarea1）を D ドライブに移動し，C ドライブに C:¥HiRDB_P¥area1¥rdarea1 という名前のシンボリックリンクを作成する手順を次に示します。

図 15–7　シンボリックリンクを使用した HiRDB ファイルシステム領域の移動



(1)　移動元の RD エリアを閉塞，クローズ状態にします

移動する HiRDB ファイルシステム領域（C:¥HiRDB_P¥area1¥rdarea1）に構成ファイルを格納しているすべての RD エリアを閉塞，クローズ状態にします。

```
pdhold -r RDAREA1 -c
```

〔説明〕

-r：閉塞する RD エリア名を指定します。

-c：閉塞状態にした後，RD エリアをクローズします。

(2)　ユーザサーバプロセスのリフレッシュ及びユーザサーバプロセスの終了確認を行います

(a)　ユーザサーバプロセスをリフレッシュします

```
pdpfresh
```

(b)　ユーザサーバプロセスの終了を確認します

pdpfresh コマンドを-c オプション指定で実行し，リターンコードが 1 の場合はリターンコードが 0 になるまで繰り返し実行します。

```
pdpfresh -c
```

〔説明〕

-c：pdpfresh コマンドでプロセス終了要求を受けたユーザサーバプロセスが，すべて終了しているかどうかを調べる場合に指定します。

(3) 移動元の RD エリアを構成する HiRDB ファイルシステム領域を移動先のディスクにコピーします

C ドライブの HiRDB ファイルシステム領域（C:¥HiRDB_P¥area1¥rdarea1）を D ドライブ（D:¥DB¥dbdir2¥generation2）へコピーします。

(4) 移動元の HiRDB ファイルシステム領域を削除します

移動元の HiRDB ファイルシステム領域（C:¥HiRDB_P¥area1¥rdarea1）を削除します。

(5) シンボリックリンクを作成します

Administrators 権限で mklink コマンドを実行し，シンボリックリンクを作成します。

シンボリックリンク名は移動元の HiRDB ファイルシステム領域と同一にします。

```
mklink C:¥HiRDB_P¥area1¥rdarea1 D:¥DB¥dbdir2¥generation2
                （移動元）                （移動先）
```

〔説明〕

mklink コマンドの形式については，Windows のマニュアルを参照してください。

(6) (1)で閉塞，クローズ状態にした RD エリアを閉塞解除し，オープン状態にします

```
pdrels -r RDAREA1 -o
```

〔説明〕

-r：閉塞状態を解除する RD エリア名を指定します。

-o：閉塞状態を解除した後，RD エリアをオープンします。

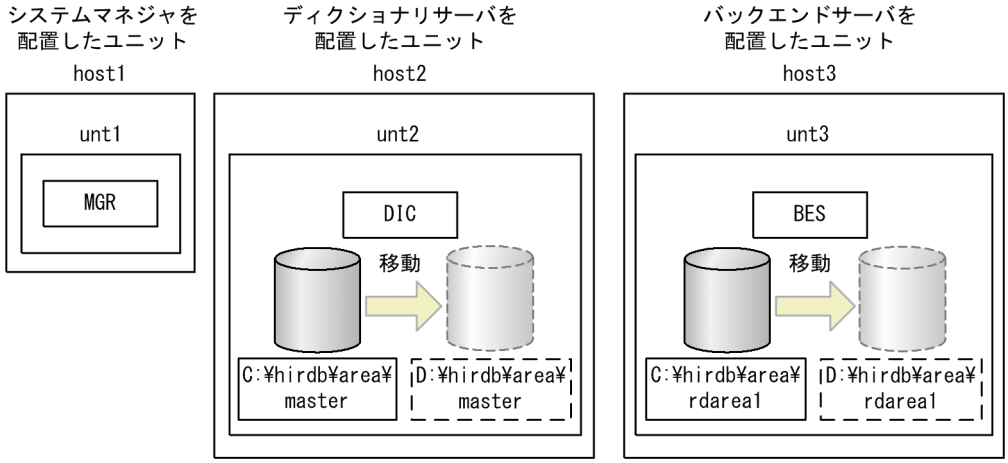
15.10.4 ファイルパス変更コマンドを使用した移動の例題

ファイルパス変更コマンドを使用して，HiRDB ファイルシステム領域を移動する例題を次に示します。

(1)　ユニットの変更のない移動の例題

ファイルパス変更コマンドを使用してマスタディレクトリ用 RD エリアを格納している C:¥hirdb¥area¥master を D:¥hirdb¥area¥master に移動し，ユーザ用 RD エリアを格納している C:¥hirdb¥area¥rdarea1 を D:¥hirdb¥area¥rdarea1 に移動する手順を示します。

図 15–8　ユニットの変更のない移動



(a)　RD エリアのバックアップを取得します

```
pdcopy -m C:¥hirdb¥area¥master -Mr -a -b バックアップファイル名
```

(b)　HiRDB に接続するシステムを停止します

(h)の HiRDB 開始後に HiRDB へアクセスしないように HiRDB に接続するシステムを停止してください。停止方法については，各製品のマニュアルを確認してください。

(c)　HiRDB を正常停止します

```
pdstop
KFPS01841-I HiRDB unit unt2 terminated. mode = NORMAL
KFPS01841-I HiRDB unit unt3 terminated. mode = NORMAL
KFPS01841-I HiRDB unit unt1 terminated. mode = NORMAL
KFPS01850-I HiRDB system terminated. mode = NORMAL
```

pdstop コマンド実行時，KFPS01850-I メッセージが出力され，終了モードに NORMAL が出力されていることを確認してください。

(d)　HiRDB ファイルシステム領域を移動します

移動元の RD エリアを構成する HiRDB ファイルシステム領域を移動先のディスクに移動します。

(e) マスタディレクトリ用 RD エリアの先頭 HiRDB ファイルを移動する場合は，システム定義 pd_master_file_name オペランドを移動後の HiRDB ファイル名称に変更します

```
変更前: set pd_master_file_name = ”C:¥hirdb¥area¥master”
変更後: set pd_master_file_name = ”D:¥hirdb¥area¥master”
```

(f) pdchpathf コマンドを実行します

```
pdchpathf 制御文ファイル パス変更情報ファイル
```

制御文

```
change path host host2
area ”D:¥hirdb¥area¥master”;
change path host host3
area ”D:¥hirdb¥area¥rdarea1”;
```

このコマンドは，ディクショナリサーバを配置したユニット（host2）で実行します。

制御文は，マスタディレクトリ用 RD エリアのファイルを移動する場合は，必ず先頭の制御文に指定してください。先頭に指定していない場合，コマンドがエラーになります。パス変更情報ファイルは，pdchpathf コマンドが情報を出力するファイルで，pdchpathn コマンドの入力データになります。

パス変更情報ファイルがない，又はパス変更情報ファイルが不正な内容である場合，pdchpathn コマンドは実行できません。また，系切り替えが発生し，ファイルが参照できなくなった場合は，バックアップから回復し，やり直してください。

(g) HiRDB を開始します

```
pdstart
……
KFPS05210-I HiRDB system initialization process complete
```

KFPS05210-I メッセージが出力されていることを確認してください。

(h) HiRDB の開始完了後，pdchpathn コマンドを実行します

```
pdchpathn パス変更情報ファイル
```

このコマンドは，システムマネジャを配置したユニット（host1）で実行します。

ディクショナリサーバを配置したユニットの，(g)で指定したパス変更情報ファイルを指定してください。また，系切り替え構成で別のマシンでディクショナリサーバを起動した場合は，該当マシンにファイルをコピーしてください。

(i)　HiRDB のバックアップを再度取得します

```
pdcopy -m D:¥hirdb¥area¥master -Mr -a -b バックアップファイル名
```

変更前のバックアップからは回復はできません。

(j)　HiRDB に接続するシステムを開始します

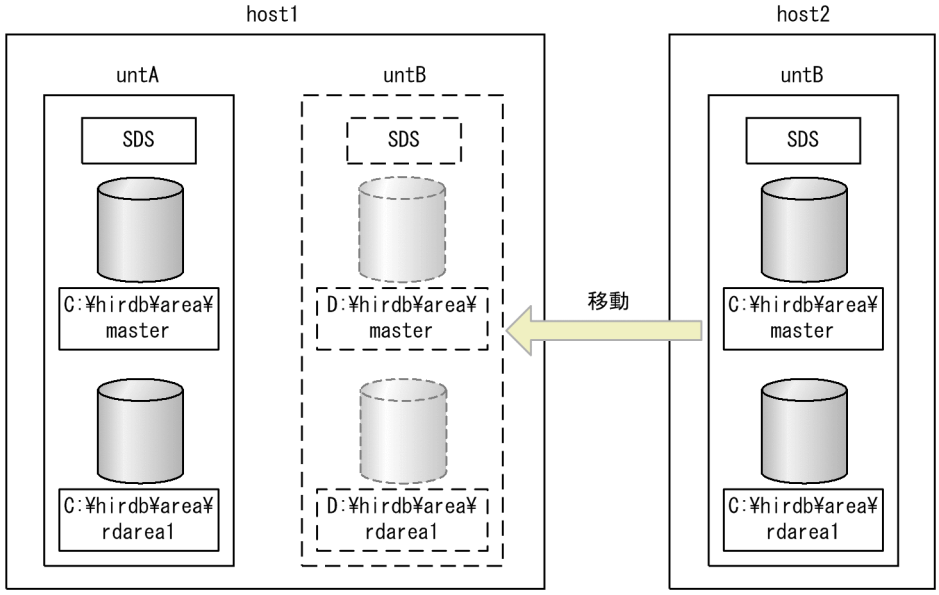
(b)で停止したシステムを必要に応じて開始してください。開始方法については各製品のマニュアルを確認してください。

(2)　ユニットを変更する移動の例題

host2 の untB の HiRDB を host1 に移動します。このとき，host1 にはすでに別の untA の HiRDB があり，同じ構成のため，HiRDB のパス名も重なっています。

host1 で新しく untB の HiRDB をセットアップし，ファイルパス変更コマンドを使用してマスタディレクトリ用 RD エリアを格納している C:¥hirdb¥area¥master を D:¥hirdb¥area¥master に移動し，ユーザ用 RD エリアを格納している C:¥hirdb¥area¥rdarea1 を D:¥hirdb¥area¥rdarea1 に移動する手順を示します。

図 15–9　ユニットを変更する移動



(a)　host2 で untB のバックアップを取得します

```
pdcopy -m C:¥hirdb¥area¥master -Mr -a -b バックアップファイル名
```

(b)　システムファイルの作成，及びシステム定義を新環境用へ変更します

host1 で untB の HiRDB をセットアップし，システムファイルの作成，及びシステム定義を新環境用へ変更します。pd_master_file_name を移動後の名称に変更します。

(c) HiRDB を正常停止します

host2 の HiRDB を正常停止します。その後，host2 のデータを host1 の新規パスに移動します。

(d) pdchpathf コマンドを実行します

```
pdchpathf 制御文ファイル パス変更情報ファイル
```

制御文

```
change path
area ”D:¥hirdb¥area¥master”
area ”D:¥hirdb¥area¥rdarea1”;
```

制御文は，マスタディレクトリ用 RD エリアのファイルを移動する場合は，必ず先頭の制御文に指定してください。先頭に指定していない場合，コマンドがエラーになります。パス変更情報ファイルは，pdchpathf コマンドが情報を出力するファイルで，pdchpathn コマンドの入力データになります。

パス変更情報ファイルがない，又はパス変更情報ファイルが不正な内容である場合，pdchpathn コマンドは実行できません。また，系切り替えが発生し，ファイルが参照できなくなった場合は，バックアップから回復し，やり直してください。

(e) HiRDB を開始します

```
pdstart
……
KFPS05210-I HiRDB system initialization process complete
```

KFPS05210-I メッセージが出力されていることを確認してください。

(f) HiRDB の開始完了後，pdchpathn コマンドを実行します

パス変更情報ファイルは，(e)で指定したパス変更情報ファイルを指定してください。

```
pdchpathn パス変更情報ファイル
```

(g) HiRDB のバックアップを再度取得します

```
pdcopy -m D:¥hirdb¥area¥master -Mr -a -b バックアップファイル名
```

変更前のバックアップからは回復はできません。

15.11　使用中空きページ及び使用中空きセグメントを再利用する方法

表及びインデクスの使用中空きページを未使用ページ化して再利用できます。同様に使用中空きセグメントを未使用セグメント化して再利用できます。ここでは，その方法について説明します。説明する項目は次のとおりです。

- ページ及びセグメントの状態
- 使用中空きページの再利用
- 使用中空きセグメントの再利用

15.11.1　ページ及びセグメントの状態

この節の説明を読む前にページ及びセグメントの状態について理解しておく必要があります。ページとセグメントには次の表に示す状態があります。

表 15–8　ページの状態

ページの状態	説明
未使用ページ	まだ割り当てられていないページです。
使用中空きページ	データの削除※によって，データが格納されていないページです。
使用中ページ	データが格納されていて，データを追加できる空き領域があるページです。 空き領域の再利用機能を使用している表の場合は，データの削除※によってページ内の空き領域が使用できないために，データを追加できなかったページを含みます。
使用中満杯ページ	データが格納されていて，データを追加できる空き領域がないページです。 空き領域の再利用機能を使用していない表及びインデクスの場合は，データの削除※によってページ内の空き領域が使用できないために，データを追加できなかったページを含みます。

注※

データの削除を実行したトランザクションが COMMIT するまで，データの削除によって発生した空き領域は使用できません。

表 15–9　セグメントの状態

セグメントの状態	説明
使用中セグメント※	表又はインデクスのデータを格納しているセグメントです。 データが満杯でセグメント内にデータを追加できないセグメントを**満杯セグメント**，それ以外の使用中セグメントを**空きありセグメント**といいます。 空きありセグメントのうち，データの削除でセグメント内の全ページが空きページ（使用中空きページ又は未使用ページ）のセグメントを**使用中空きセグメント**といいます。
未使用セグメント	使用されたことがないセグメントです。このセグメントは RD エリア内のすべての表（又はインデクス）が使用できます。

セグメントの状態	説明
空きセグメント	データを格納していないセグメントです。使用中空きセグメントと未使用セグメントは空きセグメントになります。

注※

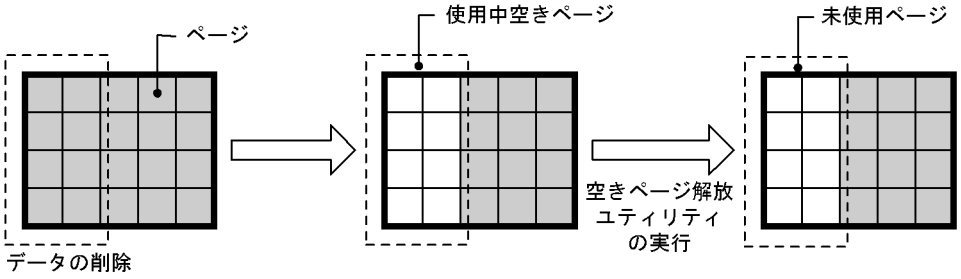
使用中セグメントを使用できるのは，このセグメントにデータを格納した表又はインデクスだけです。ほかの表又はインデクスはこのセグメントを使用できません。

15.11.2　使用中空きページの再利用

(1)　使用中空きページの解放

バッチジョブなどで表データを大量に削除すると，その表データを格納しているページ（データページ）の一部が使用中空きページになることがあります。また，インデクスを定義している場合は，インデクスのキー値を格納しているページ（インデクスページ）の一部が使用中空きページになります。**空きページ解放ユティリティ**（**pdreclaim** コマンド）を実行すると，この使用中空きページを未使用ページ化して再利用できます。これを**使用中空きページの解放**といいます。使用中空きページの解放を次の図に示します。

図 15–10　使用中空きページの解放



ポイント

- LOB 用 RD エリアに格納されているデータの使用中空きページは解放できません。
- プラグインインデクスの使用中空きページは解放できません。

(2)　使用中空きページを解放したときの効果

(a)　表の使用中空きページを解放したときの効果

表の使用中空きページを解放したときの効果を次の表に示します。

表 15–10　表の使用中空きページを解放したときの効果

効果がある項目	説明	効果の度合い
表を再編成するサイクルを長くできる	使用中空きページを再利用できるためデータの格納効率が良くなります。このため，表を再編成するサイクルを長くできます。	○
大量データ検索時の性能が向上する	使用中空きページは使用中ページのため検索処理時のサーチ対象になりますが，未使用ページはサーチ対象になりません（サーチ処理がスキップされます）。その分，検索処理の性能が向上します。特に，大量データを検索するときに効果が出ます。	△
INSERT 及び UPDATE 時の性能が向上する	使用中ページにデータを格納するとき，データの格納に必要な連続した空き領域を確保できないと，HiRDB は**ページコンパクション**という処理を行います。ページコンパクションとは，データの格納に必要な連続した空き領域を確保するために行われるページ内のデータ詰め替え処理のことです。 使用中空きページを解放するときにページコンパクションも同時に行います。このため，INSERT 及び UPDATE 処理の延長で行われるページコンパクションが不要になり，その分処理性能が向上します。なお，ページコンパクションの処理対象ページは，満杯ページ及び使用中空きページを除いた使用中ページになります。	△
分岐行の INSERT 及び UPDATE 時のエラー発生を抑えられる	未使用ページがない状態で分岐行の INSERT 及び UPDATE を実行すると，エラー（KFPA11756-E メッセージ）になります。使用中空きページの解放で未使用ページが増えるため，このエラーの発生を抑えられます。	△

（凡例）

○：効果があります。

△：条件によって効果の度合いが変わります。

特に，次の表に示す条件をすべて満たす表に対してこの機能を適用すると効果があります。

表 15–11　使用中空きページの解放の効果がある表

条件	理由
可変長文字列の更新がない	分岐行が発生しないため，使用中空きページが解放できる可能性が高くなります。 分岐行は基本行と別のページにあり，そのページには別のデータの基本行が格納されます。このため，分岐していると，ある一定の値（登録期間など）の行を削除してもページに格納されているデータが 0 件にならない可能性があります。したがって，分岐していなければ，ある期間に INSERT したデータを削除すると，連続した領域（ページ）が空くため，使用中空きページを解放できる可能性が高くなります。
繰返し列の要素数変更がない	
NULL 値を実データに，又は実データを NULL 値に変更しない	
クラスタキーインデクスを定義していない	クラスタキーインデクスを定義している場合はデータの格納位置がキー値で決定するため，使用中空きページを解放してもそのページを使用しない可能性が高く，効果が余りないと考えられます。
LOB データがない	LOB 用 RD エリアの格納データはこの機能を適用できません。

条件	理由
同一ページ中のデータを一度に削除する	使用中空きページができやすいため，使用中空きページを解放できる可能性が高くなります。
REUSE 表（空き領域の再利用機能を使用した表）である	REUSE 表の場合，空きページ解放ユティリティで解放した空きページを空きページ再利用モード時にすぐに再利用できます。REUSE 表でない場合，空きページ解放ユティリティで解放した空きページを RD エリア内に新規セグメントが確保できなくなるまで使用しません。

上記の表に示す条件から，**クラスタキーインデクスを定義していない FIX 属性の REUSE 表がこの機能を適用するのに最適な表になります。**

(b)　インデクスの使用中空きページを解放したときの効果

インデクスの使用中空きページを解放したときの効果を次の表に示します。

表 15–12　インデクスの使用中空きページを解放したときの効果

効果がある項目	説明	効果の度合い
インデクス格納 RD エリアの容量不足の発生を抑えられる	空きページ（使用中空きページ）があるのに領域不足になる場合は使用中空きページを解放してください。なお，キー値の更新又は削除が多い場合でもインデクス格納 RD エリアの容量不足の発生を抑えられます。	◎
インデクスを再編成するサイクルを長くできる	使用中空きページを再利用できるためデータの格納効率が良くなります。このため，インデクスを再編成するサイクルを長くできます。	○
インデクスを使用した大量データ検索時の性能が向上する	使用中空きページは使用中ページのため検索処理時のサーチ対象になりますが，未使用ページはサーチ対象になりません（サーチ処理がスキップされます）。その分，検索処理の性能が向上します。特に，大量データを検索するときに効果が出ます。	△

（凡例）

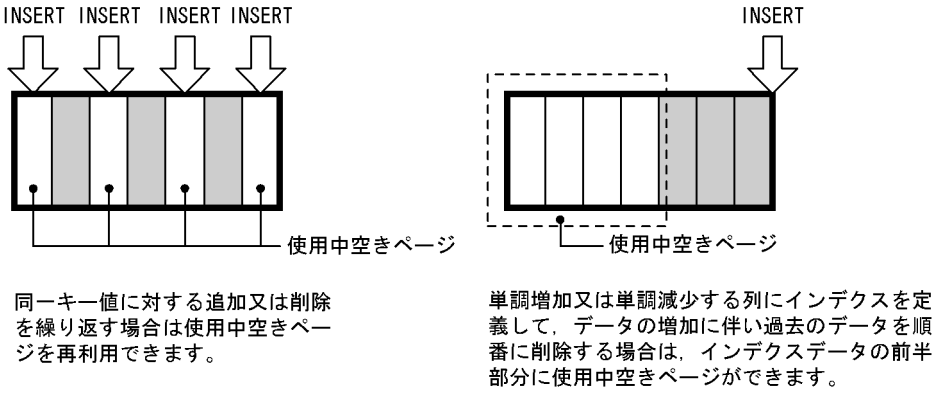
◎：特に効果があります。

○：効果があります。

△：条件によって効果の度合いが変わります。

特に，削除したキー値を再度登録しない場合にこの機能を適用すると効果があります。同一キー値の追加又は削除を繰り返す場合は使用中空きページを再利用するため，使用中空きページが大量に発生することはありません。しかし，単調増加又は単調減少する列（日付，通番など）にインデクスを定義してデータの増加に伴い過去のデータを順番に削除する場合は，インデクスページの前半部分に再利用されない使用中空きページが大量に発生します。インデクスページに使用中空きページが作成される処理を次の図に示します。

図 15–11　インデクスページに使用中空きページが作成される処理



なお，使用中空きページの解放後は解放したページにキー値を格納していくため，データの格納効率が良くなります。

参考

使用中空きページだけが処理の対象になります。格納効率が低下したページのページコンパクションはありません。格納効率が低下したページ（使用中空きページを除く）が大量にある場合に適用しても効果はありません。

(3)　表又はインデクスの再編成との違い

性能面及びデータの格納効率という点から見ると，使用中空きページの解放より表又はインデクスの再編成の方が優れています。しかし，使用中空きページの解放の場合は，ユティリティの実行中に処理対象表又はインデクスをアクセスできます。再編成の場合は，ユティリティの実行中に処理対象表又はインデクスをアクセスできません。このため，使用中空きページの解放の場合は業務を中断する必要がありません。

再編成をするか，使用中空きページを解放するかはデータベース状態解析ユティリティの実行結果から判断してください。判断基準を次に示します。

- 使用中空きページが大量にある場合は使用中空きページを解放してください。
- セグメント内の空きページ比率（CREATE TABLE の PCTFREE オペランドの値）と掛け離れたページ使用率の使用中ページが大量にある場合は再編成をしてください。

(4)　運用方法

(a)　システムログファイルの容量を見積もる

使用中空きページを解放するときにデータベースの更新ログを取得するため，システムログファイルの容量を見積もり直してください。システムログファイルの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(b)　ページの状態を確認する

データベース状態解析ユティリティ（pddbst コマンド）で，ページの格納効率，使用中空きページ数などを定期的に確認してください。ページの使用率が 0%のページが使用中空きページです。ページの使用率及び使用中空きページ数は「RD エリア単位の状態解析（論理的解析）」又は「表又はインデクス単位の状態解析」の **Used Page Ratio** に表示されます。

使用中空きページ数が増加してきた場合や，データの格納効率が低下してきた場合に使用中空きページの解放を検討してください。

(c)　空きページ解放ユティリティを実行する

空きページ解放ユティリティで使用中空きページを解放します。

表の使用中空きページを解放する場合は，ページコンパクションを行うかどうかを検討してください。次に示す場合はページコンパクションの効果がないため，ページコンパクションをしないで空きページ解放ユティリティを実行してください。

- 使用中ページの大部分が満杯ページか又は使用中空きページの場合
- すべてのページが空きページの場合

(d)　結果を確認する

空きページ解放ユティリティの実行結果を参照して，予定どおり使用中空きページが解放されているかを確認してください。また，使用中空きセグメントが多い場合は使用中空きセグメントの解放を検討してください。使用中空きセグメントの解放については，「使用中空きセグメントの再利用」を参照してください。

(e)　空きページ解放ユティリティが異常終了した場合

空きページ解放ユティリティが異常終了した場合，利用できない使用中ページが残るため，すぐに空きページ解放ユティリティを再実行してください。空きページ解放ユティリティを再実行すると，利用できない使用中ページが解放されます。

(5)　注意事項

- インデクスに対して空きページ解放ユティリティを実行する場合，ユティリティの実行中に UAP が長時間待たされることはありません。ただし，処理対象のインデクスに UAP がアクセスした場合，空きページ解放ユティリティは UAP のトランザクション決着処理を待つことがあります。
- 待ち時間の上限を，空きページ解放ユティリティの-w オプションで指定できます。指定した待ち時間内に待ち状態が解除されないと，KFPH25002-E メッセージを出力して空きページ解放ユティリティを異常終了します。-w オプションを省略すると UAP のトランザクションが決着するまで待ち続けます。
- 未使用ページが少ない状態で空きページ解放ユティリティを実行中に，ロールバックが発生すると，ロールバック中にページ不足が発生し，RD エリアが障害閉塞します。このため，多くの DELETE や UPDATE を伴う業務の実行中は，空きページ解放ユティリティを実行しないでください。

- インデクスに対して空きページ解放ユティリティを実行する場合，解放できる使用中空きページを，いったんすべて解放予約状態にします。解放予約状態の使用中空きページは，トランザクションの決着処理待ち後に解放され，未使用ページ化します。このため，空きページ解放ユティリティ実行中は，一時的に利用できるページが少なくなり，ページ不足が発生しやすくなります。
- RD エリアにインデクスだけを格納している場合，障害閉塞したとき，再初期化後に格納しているすべてのインデクスを再作成して回復してください。RD エリアに表とインデクスの両方を格納している場合は，バックアップから回復してください。
- インデクスに対して空きページ解放ユティリティを実行中にエラーが発生した場合，解放予約状態の使用中空きページが残ります。早期に空きページ解放ユティリティを再実行してください。

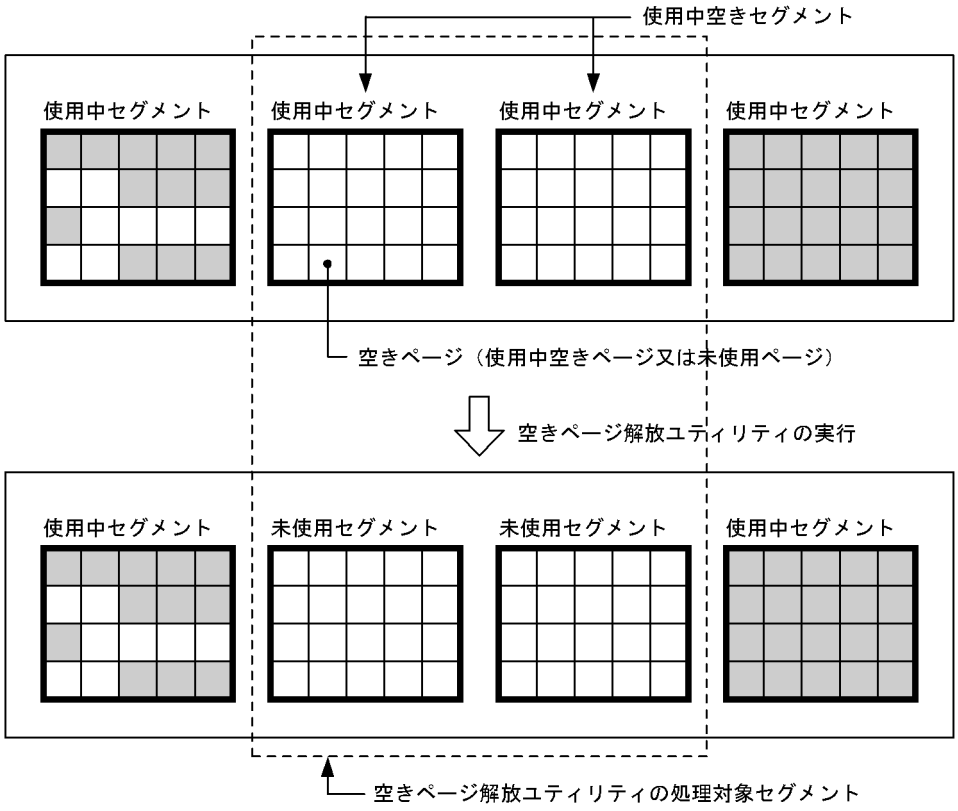
  なお，HiRDB が異常終了した場合も，エラーが発生した場合と同様に，解放予約状態の使用中空きページが残ります。HiRDB を再開始した後，空きページ解放ユティリティを再実行してください。

15.11.3　使用中空きセグメントの再利用

(1)　使用中空きセグメントの解放

空きページ解放ユティリティを実行すると，使用中空きセグメントを未使用セグメント化して再利用できます。これを**使用中空きセグメントの解放**といいます。使用中空きセグメントの解放を次の図に示します。

図 15–12　使用中空きセグメントの解放



(2)　使用中空きセグメントを解放したときの効果

一度使用されたセグメントは使用している表（又はインデクス）だけが使用でき，ほかの表は使用できません。使用中空きセグメントを解放して使用中空きセグメントを未使用セグメント化すると，その未使用セグメントをほかの表が使用できるようになります。

(3)　運用方法

空きページ解放ユティリティ又はデータベース状態解析ユティリティの実行結果から使用中空きセグメントを解放するかどうかを決めてください。大量の使用中空きセグメントがある場合に実行してください。

(a)　システムログファイルの容量を見積もる

使用中空きセグメントを解放するときにデータベースの更新ログを取得するため，システムログファイルの容量を見積もり直してください。システムログファイルの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(b)　空きページ解放ユティリティを実行する

空きページ解放ユティリティで使用中空きセグメントを解放します。

空きページ解放ユティリティ実行時のオプション指定値によるメリットとデメリットを次に示します。

オプションの指定値	メリット	デメリット
-j	RD エリアを占有するため，高速に処理でき，-a オプションと比べて実行時間が短くなります。	RD エリアを占有するため，ほかの UAP が処理対象の RD エリアにアクセスできなくなり，待ち状態になります※。
-a	ほかの UAP と同時実行できます。	内部的に複数フェーズに分けて処理するため，-j オプションと比べて実行時間が長く，出力するシステムログの量も多くなります。

注※

UAP を待ち状態にしないでエラーリターンする場合は，pdhold コマンドで RD エリアを閉塞してからユティリティを実行してください。

基本的に-j オプションをお勧めします。-j オプション指定で空きページ解放ユティリティを実行する間オンラインサービスを停止できない場合などは，比較的トラフィックが空いている時間帯に-a オプション指定で空きページ解放ユティリティを実行してください。

(c)　実行結果を確認する

空きページ解放ユティリティの実行結果を参照して，予定どおりの使用中空きセグメントが解放されているかを確認してください。

なお，-a オプション指定で空きページ解放ユティリティを実行中に，ユティリティ又は HiRDB が異常終了した場合は，データベース状態解析ユティリティ（pddbst）で該当する表又はインデクスを解析し，**解放途中セグメント**（解放処理が途中のため，再利用できない状態のセグメント）がないことを確認してください。

(d)　空きページ解放ユティリティが異常終了した場合

-j オプション指定時

空きページ解放ユティリティが異常終了すると，ユティリティ実行前の状態（何もしていない状態）になるため，空きページ解放ユティリティを再実行してください。なお，空きページ解放ユティリティを再実行しなくても問題はありませんが，使用中空きセグメントは解放されません。ユティリティ実行中に HiRDB が異常終了した場合も同様です。

-a オプション指定時

空きページ解放ユティリティが異常終了すると，解放途中セグメントが残ることがあります。解放途中セグメントが残っていた場合，次に示すどちらかの対処をしてください。

- 空きページ解放ユティリティを再実行する
- データベース再編成ユティリティ（pdrorg）で表，又はインデクスの再編成をする

ユティリティ実行中に HiRDB が異常終了した場合も同様です。

15.12　物理エラー検知時ユニットダウン機能

ここでは，物理エラー検知時ユニットダウン機能を使用する方法について説明します。なお，ここでいう物理エラーとは，RD エリアに対するアクセスで入出力エラー，ファイルオープンエラーが発生し，KFPH00306-E メッセージの閉塞理由に i/o error occurred，又は open error occurred が出力されるエラーのことです。

15.12.1　物理エラー検知時ユニットダウン機能とは

HiRDB は，稼働中に RD エリアに障害が発生すると，障害が発生した RD エリアを障害閉塞させることで影響範囲を局所化して，処理を続行します。しかし，RD エリアが少ないシステムの場合，障害が発生した RD エリアだけを障害閉塞させても業務全体が停止することがあります。この場合，業務を再開するには障害閉塞した RD エリアを回復する必要があります。

物理エラー検知時ユニットダウン機能を使用すると，障害が発生した RD エリアを障害閉塞しないで，ユニットダウンさせることができます（ただし，マスタディレクトリ用 RD エリアの場合は，物理エラー検知時ユニットダウン機能を使用していなくても障害閉塞しないでユニットダウンします）。ユニットダウンすることで，ユニットを再開始する前に物理エラーによる障害閉塞の要因を取り除くことができ，RD エリアが障害閉塞することによる回復作業を回避できます。

15.12.2　注意事項

この機能を使用するときは，次の点に注意してください。

- RD エリアに対するアクセスで物理エラーが発生し，KFPH00307-E メッセージを出力して RD エリアがコマンド閉塞した場合は，pd_db_hold_action オペランドに unitdown を指定していてもユニットダウンしません。
- 物理エラー検知時ユニットダウン機能を使用した場合に物理エラーが発生すると，次のケースでは処理対象の RD エリアが障害閉塞することがあります。

  ・更新前ログ取得モード又はログレスモードで UAP やユティリティを実行している

  ・CREATE TABLE の RECOVERY オペランドで NO を指定してログレスモードにしたユーザ LOB 用 RD エリアに対して，UAP 又はユティリティを実行している

  物理エラー検知時ユニットダウン機能を使用する場合は，できるだけこれらの運用は避けてください。もし，これらの運用が必要な場合は，RD エリアが閉塞しても最新の状態に回復できるよう，UAP 又はユティリティの実行前にバックアップを取得してください。
- 物理エラー検知時ユニットダウン機能を使用する場合は，KFPH23047-I メッセージを監視してください。KFPH23047-I メッセージを出力してユニットダウンした場合は，「物理エラーが発生した場合の対処方法」に従って対処してください。ユニットダウンの原因を取り除く前に HiRDB を再開始すると，再び物理エラーを検知してしまい，ユニットダウンと再開始が繰り返されるため注意してください。

15.12.3　準備作業

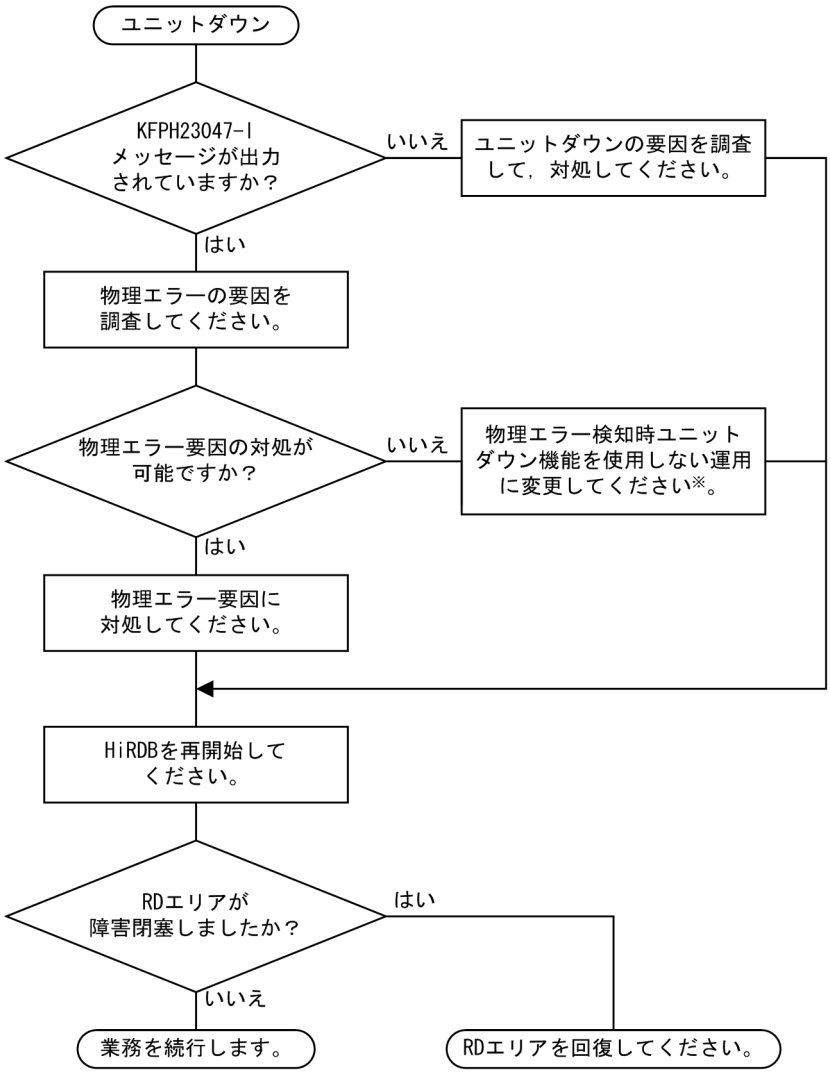
この機能を使用するには，システム共通定義（pdsys）を次のように指定してください。

1. pd_db_hold_action オペランドに unitdown を指定します。
   この値を指定すると，物理エラーを検知した場合，KFPH23047-I メッセージを出力し，ユニットダウンします。このとき，RD エリアの障害閉塞は発生しません。
2. pd_mode_conf オペランドに MANUAL2 を指定します。
   この値を指定すると，ユニットダウン後に HiRDB（ユニット）を自動開始しないようにできます。
   なお，AUTO 又は MANUAL1 を指定した場合，物理エラー検知時ユニットダウン機能でユニットダウンしても，要因を取り除く前に HiRDB（ユニット）が自動的に再開始してしまうおそれがあります。この場合，再び物理エラーを検知してしまい，ユニットダウンと再開始が繰り返されますので注意してください。

15.12.4　物理エラーが発生した場合の対処方法

物理エラー検知時ユニットダウン機能を使用していてユニットダウンが発生した場合，HiRDB 管理者は次の図に示す手順に従って対処してください。

図 15–13　物理エラーが発生した場合の対処方法



注※

物理エラーの要因に対処できない場合，ユニットを再開始しても再度同じ要因でユニットダウンを繰り返すことになります。そのときは，システム共通定義の pd_db_hold_action オペランドに dbhold を指定して，物理エラー検知時ユニットダウン機能を使用しないように変更してください。

15.13　RD エリアアクセス時のページ不正検知時の動作

RD エリアにアクセスした場合に，OS やハードの障害などによってページの読み込み内容に不整合を検出したとき，ページの読み込みをリトライします。リトライしても正常に読み込みできなかった場合，RD エリアを閉塞し，KFPH00306-E メッセージ又は KFPH00307-E メッセージに閉塞した理由（time stamp invalid）を出力します。

ページの読み込みリトライと，RD エリアの閉塞の動作は変更できます。通常は，この動作を変更する必要はありません。OS やハードが一時的に不安定な状況でページの不整合を検知する場合で，かつ不安定な状況の発生をすぐに対処できないときに，ページ不正検知時の動作の変更を検討してください。

変更方法を次に示します。

- ページの読み込みリトライ

  ページの読み込みリトライ方法は，システム共通定義に pd_dbbuff_read_retry_count オペランド及び pd_dbbuff_read_retry_intvl オペランドを指定することで，それぞれ読み込み回数及び読み込み間隔を変更できます。詳細はマニュアル「HiRDB システム定義」を参照してください。

- RD エリアの閉塞

  RD エリアの閉塞動作は，システム共通定義の pd_db_timestamp_invalid_action オペランドに unitdown を指定することで，RD エリアを閉塞させないでユニットダウンさせることができます。詳細はマニュアル「HiRDB システム定義」を参照してください。

16 ストアドプロシジャ及びストアドファンクションの運用

この章では，ストアドプロシジャ及びストアドファンクションの運用方法について説明します。

なお，ストアドプロシジャとは，SQL ストアドプロシジャ，Java ストアドプロシジャ，及び C ストアドプロシジャの総称です。ストアドファンクションとは，SQL ストアドファンクション，Java ストアドファンクション，及び C ストアドファンクションの総称です。

16.1 ストアドプロシジャ及びストアドファンクションを作成（登録）する前に

実行者　HiRDB 管理者

ストアドプロシジャ及びストアドファンクションを作成する前に，データベース構成変更ユティリティ（pdmod コマンド）で，ストアドプロシジャ及びストアドファンクションを格納する RD エリアを作成してください。作成する RD エリアを次に示します。

- データディクショナリ LOB 用 RD エリア
- データディクショナリ用 RD エリア※

注※

ストアドプロシジャ及びストアドファンクションに関するディクショナリ表を，既存のデータディクショナリ用 RD エリアと別の RD エリアに格納する場合に作成してください。

16.2　ストアドプロシジャ及びストアドファンクションを作成（登録）する方法

16.2.1　ストアドプロシジャ及びストアドファンクションの作成（登録）方法

(1)　ストアドプロシジャを作成する方法

CREATE PROCEDURE でストアドプロシジャを作成できます。CREATE PROCEDURE 中の手続きの作成方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(2)　ストアドファンクションを作成する方法

CREATE FUNCTION でストアドファンクションを作成できます。CREATE FUNCTION 中の関数の作成方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(a)　ストアドファンクションを作成すると，既存のストアドファンクションが無効になることがあります

ストアドファンクションを作成すると，既存のストアドファンクションが無効になることがあります。無効になる条件については，マニュアル「HiRDB SQL リファレンス」の「CREATE〔PUBLIC〕FUNCTION」で説明しています。

既存のストアドファンクションが無効になった場合は，無効になったストアドファンクションを ALTER ROUTINE で再作成してください。

(b)　ストアドファンクションを作成すると，既存のストアドプロシジャが無効になることがあります

ストアドファンクションを作成すると，既存のストアドプロシジャが無効になることがあります。無効になる条件については，マニュアル「HiRDB SQL リファレンス」の「CREATE〔PUBLIC〕FUNCTION」で説明しています。

既存のストアドプロシジャが無効になった場合は，ストアドプロシジャを ALTER PROCEDURE 又は ALTER ROUTINE で再作成してください。

(c)　ストアドファンクションを作成すると，既存のトリガが無効になることがあります

ストアドファンクションを作成すると，既存のトリガが無効になることがあります。無効になる条件については，マニュアル「HiRDB SQL リファレンス」の「CREATE〔PUBLIC〕FUNCTION」で説明しています。

既存のトリガが無効になった場合は，無効になったトリガを ALTER TRIGGER 又は ALTER ROUTINE で再作成してください。

(d)　作成したストアドファンクションが無効になることがあります

次に示す手順でストアドファンクションを作成すると，そのストアドファンクションが無効になることがあります。

〈手順〉

1. プラグインをインストールします。
2. 1 のプラグイン関数を呼び出すストアドファンクション※を作成します。
3. 1 でインストールしたプラグインとは別のプラグインをインストールします。

注※

1 と 3 でインストールするプラグインが提供する関数名とパラメタ数が同じ場合，3 の操作をすると，2 で作成したストアドファンクションが無効になります。この場合，無効になったストアドファンクションを ALTER ROUTINE で再作成してください。

16.3　ストアドプロシジャ及びストアドファンクションが無効になった場合

表又はインデクスなどの定義を変更すると，その表又はインデクスを使用しているストアドプロシジャ及びストアドファンクションが無効になります。この場合，次に示す方法でストアドプロシジャ及びストアドファンクションを再作成してください。

ALTER PROCEDURE 又は **ALTER ROUTINE** で，ストアドプロシジャを再作成できます。**ALTER ROUTINE** でストアドファンクションを再作成できます。

16.4　ストアドプロシジャ及びストアドファンクションを削除する方法

16.4.1　ストアドプロシジャ及びストアドファンクションの削除方法

(1)　ストアドプロシジャを削除する方法

DROP PROCEDURE でストアドプロシジャを削除できます。

(2)　ストアドファンクションを削除する方法

DROP FUNCTION でストアドファンクションを削除できます。

(a)　ストアドファンクションを削除すると，そのほかのストアドファンクションが無効になることがあります

ストアドファンクションを削除すると，そのほかのストアドファンクションが無効になることがあります。無効になる条件については，マニュアル「HiRDB SQL リファレンス」の「DROP〔PUBLIC〕FUNCTION」で説明しています。

ストアドファンクションが無効になった場合は，無効になったストアドファンクションを ALTER ROUTINE で再作成してください。

(b)　ストアドファンクションを削除すると，既存のストアドプロシジャが無効になることがあります

ストアドファンクションを削除すると，既存のストアドプロシジャが無効になることがあります。無効になる条件については，マニュアル「HiRDB SQL リファレンス」の「DROP〔PUBLIC〕FUNCTION」で説明しています。

既存のストアドプロシジャが無効になった場合は，無効になったストアドプロシジャを ALTER PROCEDURE 又は ALTER ROUTINE で再作成してください。

(c)　ストアドファンクションを削除すると，既存のトリガが無効になることがあります

ストアドファンクションを削除すると，既存のトリガが無効になることがあります。無効になる条件については，マニュアル「HiRDB SQL リファレンス」の「DROP〔PUBLIC〕FUNCTION」で説明しています。

既存のトリガが無効になった場合は，無効になったトリガを ALTER TRIGGER 又は ALTER ROUTINE で再作成してください。

16.5　既存のストアドプロシジャの定義系 SQL を作成する方法

実行者　HiRDB 管理者又は DBA 権限保持者

pddefrev コマンドで，現在使用しているストアドプロシジャの定義系 SQL を作成できます。現在使用しているストアドプロシジャと同じような処理をするストアドプロシジャを作成するときに使用すると便利です。pddefrev コマンドで作成された定義系 SQL は，データベース定義ユティリティ（pddef コマンド）の入力情報となります。

17 Java ストアドプロシジャ及び Java ストアドファンクション使用時の運用

この章では，Java ストアドプロシジャ及び Java ストアドファンクション使用時の環境設定と運用方法について説明します。

17.1　Java ストアドプロシジャ及び Java ストアドファンクションの概要

ここでは，Java ストアドプロシジャ及び Java ストアドファンクションの概要について説明します。

17.1.1　Java ストアドプロシジャ，Java ストアドファンクションを使用できる環境

Java ストアドプロシジャ，Java ストアドファンクションを使用できる環境を次の表に示します。

表 17–1　Java ストアドプロシジャ，Java ストアドファンクションを使用できる環境

適用 OS	使用可否	
	Type2 JDBC ドライバ使用時	Type4 JDBC ドライバ使用時
Windows (x86)	○	○
Windows (x64)	×	○

（凡例）

○：使用できます。

×：使用できません。

17.1.2　Java ストアドプロシジャ及び Java ストアドファンクションとは

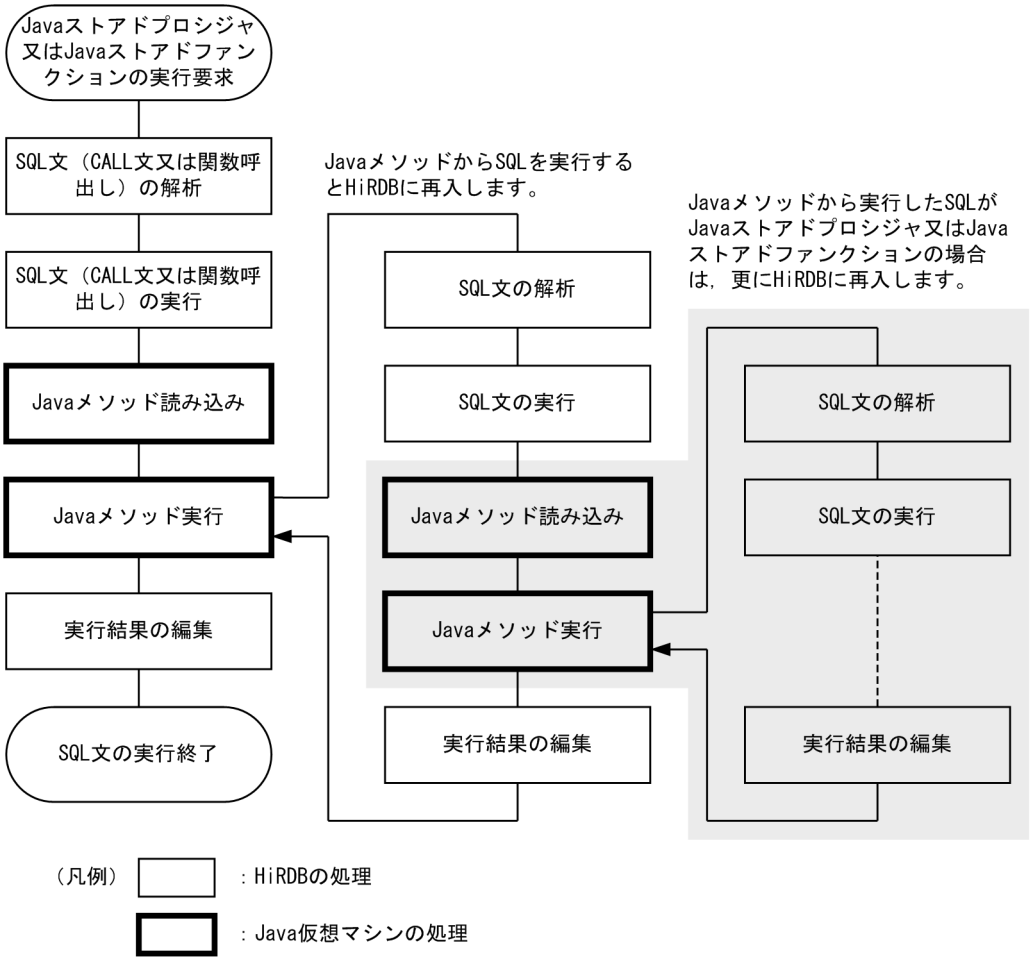
SQL で記述していたルーチン制御文を Java で記述して作成したストアドプロシジャ，ストアドファンクションのことを **Java ストアドプロシジャ**，**Java ストアドファンクション**といいます。Java ストアドプロシジャ及び Java ストアドファンクションは，HiRDB の外部で作成した Java メソッドをルーチン制御文として HiRDB に登録して使用します。このため，プラットフォームに依存しない処理の開発やデバッグが行えます。

また，Java ストアドプロシジャ及び Java ストアドファンクションは，SQL で記述したストアドプロシジャ，ストアドファンクションと同様に，SQL（CALL 文又は関数呼出し）で呼び出せます。このため，制御文の記述言語に制限はありません。

17.1.3　Java ストアドプロシジャ及び Java ストアドファンクションの動作

Java ストアドプロシジャ及び Java ストアドファンクションの動作（呼び出し処理の手順）を次の図に示します。

図 17–1　Java ストアドプロシジャ及び Java ストアドファンクションの動作（呼び出し処理の手順）



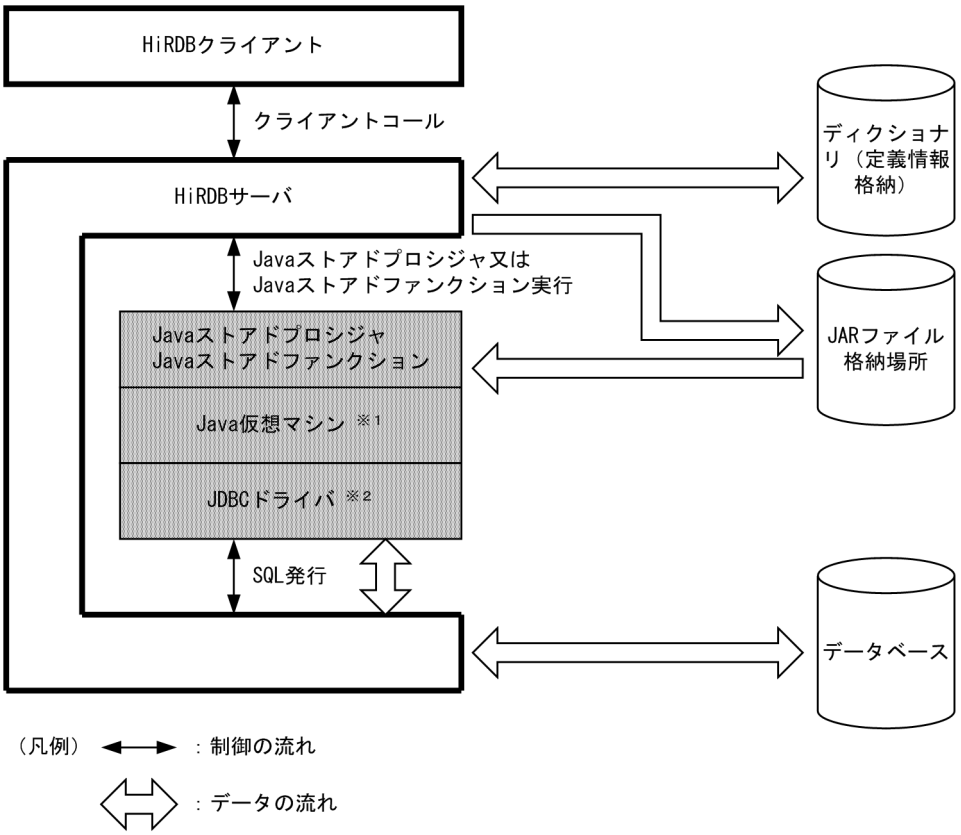
17.2　Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成

ここでは，Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成（Java 仮想マシンの実行場所及び JAR ファイルの格納場所など）について説明します。

17.2.1　Java 仮想マシンの位置づけ

HiRDB システムでの Java 仮想マシンの位置づけを次の図に示します。

図 17–2　HiRDB システムでの Java 仮想マシンの位置づけ



注※１

Java 仮想マシンは JRE（Java 実行環境）に含まれています。JRE については，「JRE（Java 実行環境）」を参照してください。

注※２

JDBC ドライバは HiRDB が標準提供しています。

17.2.2　JRE（Java 実行環境）

Java ストアドプロシジャ及び Java ストアドファンクション使用時には，HiRDB サーバに JRE（Java 実行環境）が必要となります。

(1)　JRE の入手方法

各プラットフォームのベンダーのホームページから，JRE についての情報を取得し，JRE を入手してください。なお，プラットフォームによって JRE のバージョンが異なります。各プラットフォームの JRE のバージョンを次の表に示します。

表 17–2　JRE のバージョン

適用 OS	バージョン	
	Type2 JDBC ドライバ使用時	Type4 JDBC ドライバ使用時
Windows (x86)	1.3 以降	1.4.2 以降
Windows (x64)	−	1.5.0 以降

（凡例）

−：Java ストアドプロシジャ及び Java ストアドファンクションを使用できません。

(2)　バージョン 07-02 以前の HiRDB で同梱されていた JRE の扱い

バージョン 07-02 以前の HiRDB では，JRE が同梱されていました。使用していた JRE は，HiRDB のアンインストール時に削除されます。

(3)　バージョン 07-03 以降の HiRDB にバージョンアップする場合の注意事項

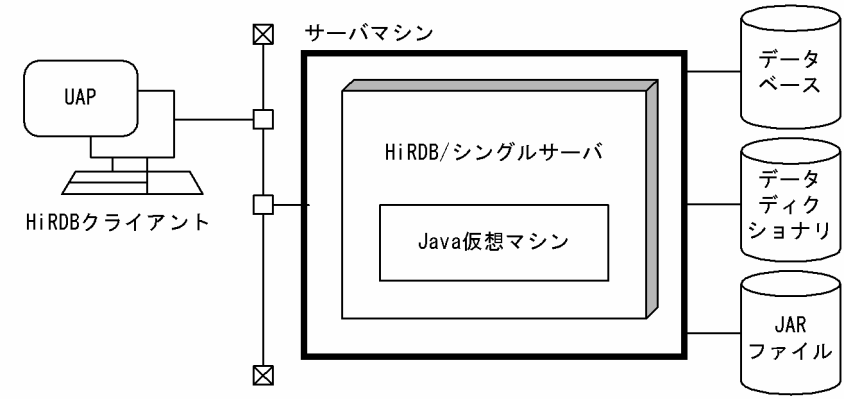
バージョン 07-03 以降の HiRDB にバージョンアップする場合の注意事項を次に示します。

- バージョン 07-03 以降の HiRDB にバージョンアップする場合，pd_java_runtimepath オペランドに JRE のルートディレクトリ（Java Runtime Environment のルートディレクトリ）を指定する必要があります。また，必要に応じて，pd_java_libpath オペランドに JRE のライブラリが格納されているディレクトリを指定してください。
- バージョン 07-02 以前の HiRDB をインストールしたときに展開された JRE は，「バージョン 07-02 以前の HiRDB で同梱されていた JRE の扱い」の契機で削除されます。バージョン 07-02 以前の HiRDB をインストールしたときに展開された JRE を引き続き使用したい場合は，次の操作をしてください。
  1. JRE が削除される前に，インストールディレクトリ及び HiRDB 運用ディレクトリ以外のディレクトリに JRE を退避します。
  2. pd_java_runtimepath オペランドに，退避した JRE のディレクトリを指定します。

17.2.3　HiRDB/シングルサーバの場合のシステム構成

Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成（HiRDB/シングルサーバの場合）を次の図に示します。

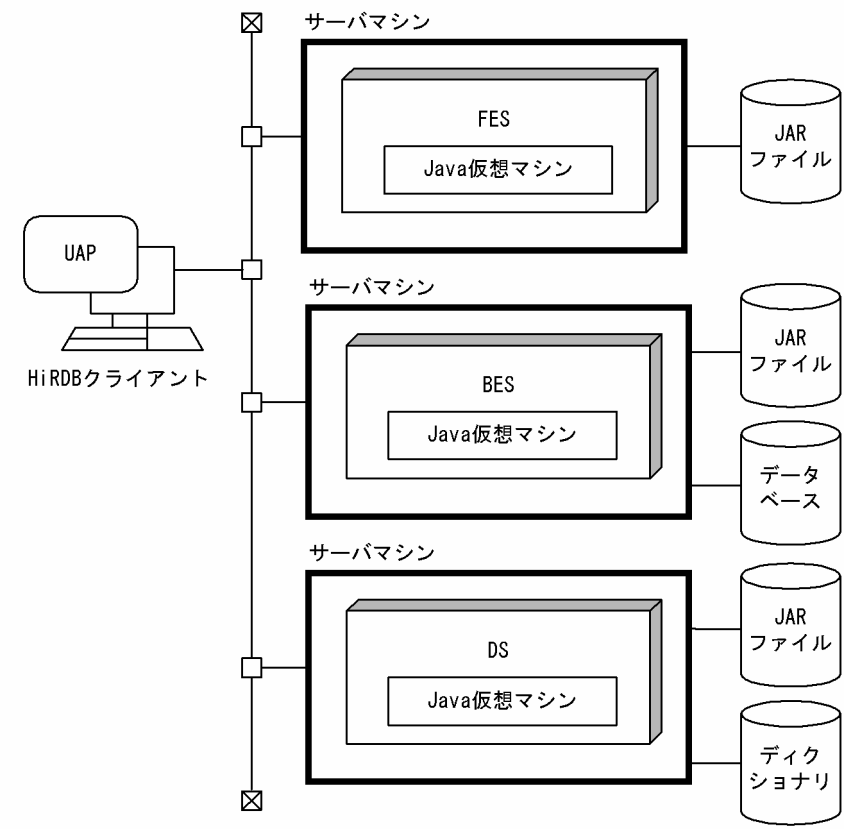
図 17–3　Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成（HiRDB/シングルサーバの場合）



17.2.4　HiRDB/パラレルサーバの場合のシステム構成

Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成（HiRDB/パラレルサーバの場合）を次の図に示します。

図 17–4　Java ストアドプロシジャ及び Java ストアドファンクション使用時のシステム構成（HiRDB/パラレルサーバの場合）



17.3　環境設定

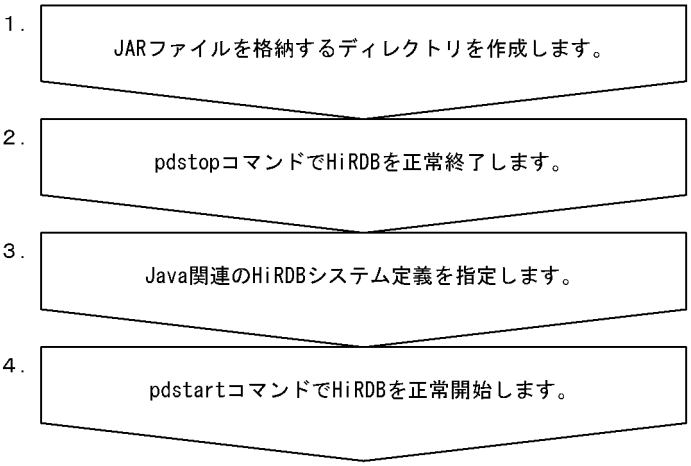
17.3.1　環境設定手順

実行者　HiRDB 管理者

Java ストアドプロシジャ及び Java ストアドファンクションを使用する場合の環境設定手順を次の図に示します。なお，ここでの説明は Java 仮想マシン及び JDBC Driver の環境設定が終了していることを前提としています。

ここで説明した環境設定を終了すると，Java ストアドプロシジャ及び Java ストアドファンクションを作成して実行できます。作成及び実行方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

図 17–5　Java ストアドプロシジャ及び Java ストアドファンクションを使用する場合の環境設定手順



注　処理ボックスの左にある数字は，この後で説明している（　）レベルに対応しています。
例えば，３．の操作は（３）で説明しています。

(1)　JAR ファイルを格納するディレクトリを作成します

JAR ファイルを格納するディレクトリを作成してください。ここで作成したディレクトリ名を(3)で指定する **pd_java_archive_directory** オペランドに指定します。

ポイント

- JAR ファイルは JAR ファイル専用のディレクトリに格納してください。
- JAR ファイルを格納するディレクトリには，インストールした JAR ファイル以外のファイルを格納しないでください。

(2) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に HiRDB システム定義を変更できるため，ここで HiRDB を正常終了する必要がありません。ただし，このコマンドを使用する場合は HiRDB Advanced High Availability が必要になります。HiRDB の稼働中に HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

(3) Java 関連の HiRDB システム定義のオペランドを指定します

HiRDB システム定義の次に示す Java 関連のオペランドを指定してください。

- pd_java_option：Java オプションを指定します。
- pd_java_routine_stack_size：外部 Java ルーチンが使用するスタック領域長をバイト単位で指定します。
- pd_java_archive_directory：JAR ファイル格納ディレクトリを指定します。
- pd_java_classpath：Java クラスパスを指定します。
- pd_java_runtimepath：Java Runtime Environment のルートディレクトリを指定します。
- pd_java_libpath：Java 仮想マシンのライブラリディレクトリを指定します。
- pd_java_stdout_file：Java 仮想マシンの標準・標準エラー出力の設定先ファイルを指定します。※1
- pd_java_castoff：Java 仮想マシンを起動したサーバプロセスを特定の契機で終了させるかどうかを指定します。※2

注※1

Java ストアドプロシジャ又は Java ストアドファンクションのデバッグ時に指定することをお勧めします。通常はこのオペランドを指定しないでください。

注※2

次の場合に指定することを検討してください。通常はこのオペランドを指定しないでください。

- Java 仮想マシンの使用によってメモリサイズが大きくなり，システムの空きメモリを圧迫している場合
- 別のコネクションで Java 仮想マシンが設定したスタックサイズの上限が原因で，Java 仮想マシンを使用しないコネクションでも探索条件が多い SQL を実行するとスタックの拡張ができなくなり，サーバプロセスがセグメンテーション障害でアボートする場合

(4) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

17.4　JAR ファイルの操作

実行者　HiRDB 管理者

pdjarsync コマンドで次に示すことができます。

- JAR ファイルの登録
- JAR ファイルの再登録
- JAR ファイルの削除
- JAR ファイルの一覧表示

17.4.1　JAR ファイルに障害が発生した場合

JAR ファイルに障害が発生した場合，各プログラマが INSTALL JAR 文で JAR ファイルを登録し直すか，HiRDB 管理者が pdjarsync コマンドで JAR ファイルを登録し直してください。

17.4.2　サーバ構成を変更した場合（HiRDB/パラレルサーバ限定）

HiRDB のサーバ構成を変更した場合は，**pdjarsync -S** コマンドで JAR ファイルを再登録してください。

18 C ストアドプロシジャ及び C ストアドファンクション使用時の運用

この章では，C ストアドプロシジャ又は C ストアドファンクションを使用するときの HiRDB 管理者が行う環境設定と運用方法について説明します。

C ストアドプロシジャ又は C ストアドファンクションの作成方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

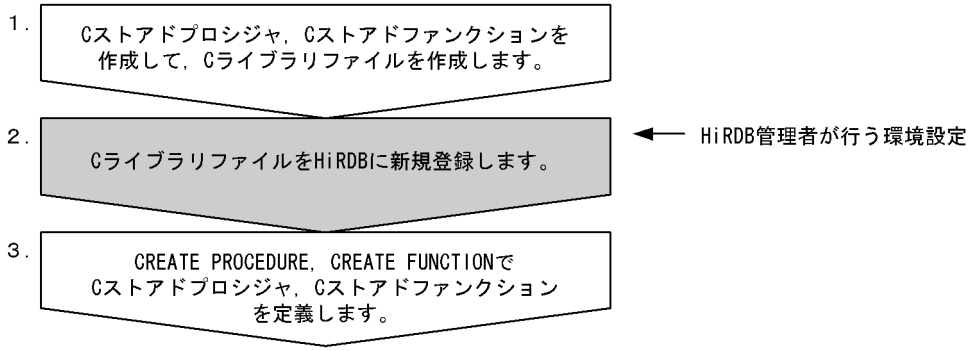
18.1　C ストアドプロシジャ又は C ストアドファンクション使用時の環境設定

ここでは，C ストアドプロシジャ又は C ストアドファンクションを使用するときの HiRDB 管理者が行う環境設定について説明します。

18.1.1　C ストアドプロシジャ又は C ストアドファンクションの実行環境を作成するまでの流れ

C ストアドプロシジャ又は C ストアドファンクションの実行環境を作成するまでの流れを次の図に示します。

図 18–1　C ストアドプロシジャ又は C ストアドファンクションの実行環境を作成するまでの流れ



〔説明〕

HiRDB 管理者は 2 の作業を行ってください。作業の詳細については，「C ライブラリファイルの新規登録」を参照してください。

1 と 3 の作業は UAP 開発者が行います。1 と 3 の作業の詳細については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

18.1.2　C ライブラリファイルの新規登録

UAP 開発者が作成した C ライブラリファイルを HiRDB に新規登録します。C ライブラリファイルの新規登録手順を次に示します。

手順

1. クライアントで作成した C ライブラリファイルを，HiRDB サーバが稼働しているサーバマシンにコピーします。

   HiRDB/パラレルサーバの場合は，HiRDB/パラレルサーバを構成するサーバマシンのうちのどれか一つにコピーしてください（全サーバマシンにコピーする必要はありません）。

2. C ライブラリファイルを格納するディレクトリを作成します。

3. pd_c_library_directory オペランドに，2 で作成したディレクトリを指定します。

4. pdclibsync コマンドで C ライブラリファイルを新規登録します。

```
pdclibsync -I -u USER01 -f C:¥usr¥user01¥propcfile01.sl
```

-I：C ライブラリファイルを新規登録する場合に指定するオプションです。

-u：C ライブラリファイルの所有者を指定するオプションです。

-f：手順 1 でクライアントからコピーした C ライブラリファイルを指定します。

注　C ライブラリファイルの拡張子は OS によって異なります。

手順 4 で pdclibsync コマンドを実行すると，手順 3 で指定したディレクトリ下に C ライブラリファイルがコピーされて，HiRDB に新規登録されます。

なお，C ライブラリファイルの新規登録は，SQL の INSTALL CLIB でも実行できます。INSTALL CLIB を使用した C ライブラリファイルの新規登録方法については，マニュアル「HiRDB SQL リファレンス」を参照してください。

●HiRDB/パラレルサーバの場合

- すべてのサーバマシンに同じ C ライブラリファイルを新規登録する必要があります。一度の pdclibsync コマンドの実行で，各サーバマシンのディレクトリ下に同じ C ライブラリファイルが新規登録（コピー）されます。
- HiRDB/パラレルサーバの構成を変更し，サーバマシンを追加した場合は，pdclibsync -S コマンドで C ライブラリファイルをほかのサーバマシンから登録してください。

●系切り替え機能を使用している場合

予備系の HiRDB にも C ライブラリファイルを登録してください。登録しない場合，系が切り替わった後に C ストアドプロシジャ又は C ストアドファンクションを実行すると SQL エラーになります。

18.1.3　C ライブラリファイルの再登録

C ストアドプロシジャ又は C ストアドファンクションの修正などによって，C ライブラリファイルに変更が発生した場合は，C ライブラリファイルを pdclibsync -R コマンドで再登録してください。

なお，C ライブラリファイルの再登録は，SQL の REPLACE CLIB でも実行できます。REPLACE CLIB を使用した C ライブラリファイルの再登録方法については，マニュアル「HiRDB SQL リファレンス」を参照してください。

18.1.4　C ライブラリファイルの削除

不要な C ライブラリファイルは，pdclibsync -D コマンドで削除してください。

なお，C ライブラリファイルの削除は，SQL の REMOVE CLIB でも実行できます。REMOVE CLIB を使用した C ライブラリファイルの削除方法については，マニュアル「HiRDB SQL リファレンス」を参照してください。

参考

pdclibsync -L コマンドで，登録中の C ライブラリファイルの一覧を表示できます。

18.2　C ライブラリファイルに障害が発生した場合の対処

C ライブラリファイルに障害が発生した場合は，障害の原因を取り除いた後に，C ライブラリファイルを pdclibsync -R コマンドで再登録してください。

19 最適化情報の運用

この章では，最適化情報の収集，登録，更新，回復，及び削除について説明します。

19.1　最適化情報とは

19.1.1　最適化情報の概要

最適化情報とは，表の行数やインデクス構成列のデータ分布情報など，表，列，及びインデクスに関する統計情報を集めたものです。

HiRDB は，コストベースで SQL の最適化を行い，自動的に最適なアクセスパスを決定します。最適化情報をディクショナリ表に登録しておくことで，最適化情報を利用して最適化の精度を向上し，より最適なアクセスパスを決定できます。SQL の最適化については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

ポイント

SQL 単位でチューニングを行う場合は，最適化情報を登録する必要はありません。最適化情報を登録，又は更新すると，アクセスパスが頻繁に変わり，性能確保のために運用が複雑化するおそれがあるためです。

SQL 単位でのチューニングは行わないが，一定の性能を確保したいときは，最適化情報を登録することをお勧めします。

19.1.2　最適化情報の登録方法

最適化情報をディクショナリ表に登録するには，次の二つの方法があります。

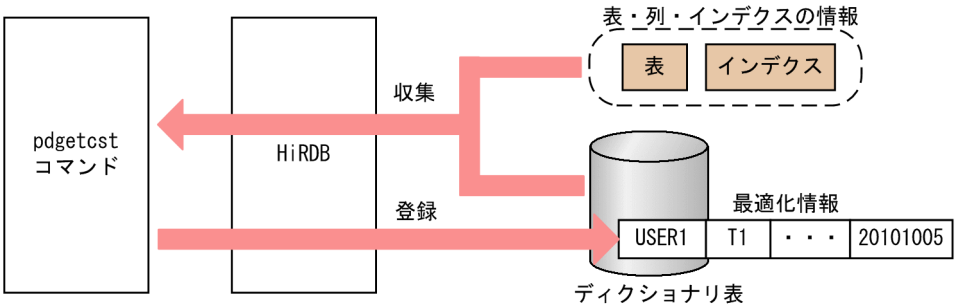
- データベースから最適化情報を収集し，ディクショナリ表に登録する。
- 最適化情報パラメタファイルに最適化情報を記述し，ディクショナリ表に登録する。

どちらの場合でも，ディクショナリ表に登録するときは最適化情報収集ユティリティ（pdgetcst コマンド）を使用します。なお，最適化情報パラメタファイルについては，「最適化情報パラメタファイルとは」を参照してください。

(1)　データベースからの最適化情報の収集

pdgetcst コマンドで最適化情報の自動収集，及びディクショナリ表への登録を一度に実行できます。具体的な運用方法については，「最適化情報の運用方法」を参照してください。データベースからの最適化情報の収集を次の図に示します。

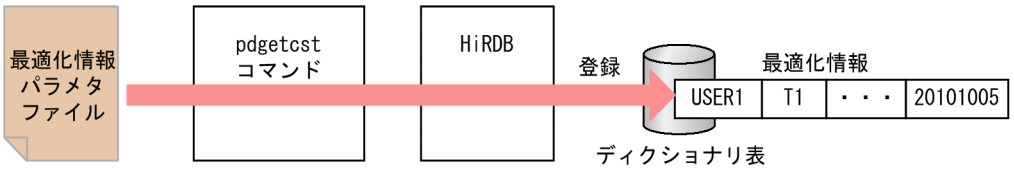
図 19–1　データベースからの最適化情報の収集



(2)　最適化情報パラメタファイルを使用した最適化情報の登録

最適化情報を記述した最適化情報パラメタファイルを作成して，pdgetcst コマンドでディクショナリ表に登録します。具体的な運用方法については，「最適化情報パラメタファイルの運用方法」を参照してください。最適化情報パラメタファイルを使用した最適化情報の登録を次の図に示します。

図 19–2　最適化情報パラメタファイルを使用した最適化情報の登録



19.1.3　最適化情報の内容

最適化情報には，表，列，及びインデクスに関する情報が含まれています。最適化情報の具体的な項目については，マニュアル「HiRDB コマンドリファレンス」の「最適化情報収集ユティリティが取得する最適化情報」を参照してください。

19.1.4　最適化情報収集レベル

最適化情報の収集の仕方には二つのレベル（lvl1，lvl2）があり，レベルによって，収集できる内容が変わります。このレベルのことを，最適化情報収集レベルといいます。それぞれのレベルの特徴を次に示します。

lvl1：収集時間は短いですが，最適化の精度は高くありません。

lvl2：最適化の精度は高いですが，収集時間は長くなります。

最適化情報収集レベルについては，マニュアル「HiRDB コマンドリファレンス」の「最適化情報収集レベル」を参照してください。

19.2　最適化情報の運用方法

19.2.1　最適化情報の収集，登録

(1)　最適化情報を収集する表

最適化情報の収集，登録を行う場合，すべての表から最適化情報を収集して，ディクショナリ表に登録することをお勧めします。一部の表だけ最適化情報を収集，登録する場合は，表の結合など，一つの SQL で複数の表をアクセスするときに，最適化情報を収集している表と収集していない表が混在しないように注意してください。

なお，最適化情報の収集，登録を行っても，必ずしも最適なアクセスパスを選択するとはかぎりません。より最適なアクセスパスを選択したい場合は，登録されている最適化情報を削除して，ユーザが SQL 単位でチューニングするようにしてください。

(2)　最適化情報の収集，登録の手順

最適化情報の収集，登録の手順を次に示します。

＜手順＞

1. 最適化情報の収集対象の表を決定する。

   「最適化情報を収集する表」の内容を考慮して，最適化情報を収集するかどうか，及びどの表から最適化情報を収集するかを判断してください。

2. 最適化情報収集レベルを決定する。

   最適化情報収集レベルを決定してください。最適化情報収集レベルは，最適化の精度がより向上する **lvl2** をお勧めします。

3. 最適化情報を収集，登録する。

   pdgetcst コマンドを使用して，最適化情報の収集，登録を実施してください。

   コマンド実行例：

   最適化情報の収集対象の表を **USER1.T1**，最適化情報収集レベルを **lvl2** として，最適化情報を収集，登録します。このときの pdgetcst コマンドの実行例を次に示します。

   ```
   pdgetcst -a USER1 -t T1 -c lvl2
   ```

19.2.2　最適化情報の更新

(1)　最適化情報の更新の契機

データベースの状態が変化すると，最適化情報とデータベースの状態が不一致になります。この場合，検索時に最適なアクセスパスを選択できなくなり，検索性能が低下するおそれがあります。そのため，データベースの状態が変化した場合は，最適化情報の更新を検討する必要があります。

データベースの状態が変化する契機を次に示します。

- 大量のデータを追加，更新，又は削除した場合
- データベース作成ユティリティ（pdload）でデータロードを行った場合
- インデクスを追加，又は削除した場合
- データベース再編成ユティリティ（pdrorg）でデータベースを再編成した場合
- 空きページ解放ユティリティ（pdreclaim）で空きページを解放した場合
- リバランスユティリティ（pdrbal）でハッシュ分割された表のリバランスを行った場合

(2)　最適化情報の更新の手順

最適化情報の更新の手順を次に示します。

＜手順＞

1. 最適化情報の更新が必要かどうかを判断する。
   「最適化情報の更新の契機」の内容を参照し，最適化情報を更新する必要があるかどうかを判断してください。
2. 現在登録している最適化情報を退避する。
   pdgetcst コマンドを使用して，最適化情報パラメタファイルへの最適化情報の退避を実施してください。最適化情報の退避については，「最適化情報パラメタファイルへの最適化情報の退避」を参照してください。

   ポイント

   最適化情報を更新することで，最適なアクセスパスを選択できるようになり，基本的には最適化の精度は向上します。しかし，場合によっては，最適化情報を更新しても最適化の精度が向上しないこともあります。このとき，最適化情報の更新を元に戻せるようにするため，ここでバックアップとして最適化情報を退避しておきます。

   性能が低下するなどの問題が発生した場合は，退避しておいた最適化情報パラメタファイルを使用して，最適化情報を更新前の状態に戻せます。

   最適化情報を更新前の状態に戻す手順については，「最適化情報の回復」を参照してください。

コマンド実行例：

現在登録している表 **USER1.T1** の最適化情報を，最適化情報パラメタファイル（c:¥tmp¥optinfo_20101014）に退避します。

```
pdgetcst -e c:¥tmp¥optinfo_20101014 -a USER1 -t T1
```

3. 最適化情報を更新する。

   pdgetcst コマンドを使用して，最適化情報の収集，登録を実施してください。これによって，最適化情報が更新されます。

   コマンド実行例：

   表 **USER1.T1** の最適化情報を更新するために，最適化情報を収集，登録します。

```
pdgetcst -a USER1 -t T1 -c lvl2
```

19.2.3　最適化情報の回復

(1)　最適化情報の回復の契機

最適化情報を回復する契機を次に示します。

- 最適化情報を更新した結果，更新前より性能が低下してしまった場合

(2)　最適化情報の回復の手順

最適化情報の回復の手順を次に示します。

＜手順＞

1. 最適化情報を回復する表を決定する。

   性能が低下した SQL から，最適化情報の回復を必要とする表を特定してください。

2. 回復に使用する最適化情報パラメタファイルを準備する。

   最適化情報を更新するときに，バックアップとして退避した最適化情報パラメタファイルを準備してください。

   最適化情報の更新については，「最適化情報の更新」を参照してください。

   なお，準備する最適化情報パラメタファイルについては，注意点があります。詳細は，「表及びインデクスの定義変更をした場合の最適化情報パラメタファイルの使用可否」を参照してください。

3. 必要に応じて最適化情報を削除する。

   「最適化情報退避後の pdgetcst の操作と最適化情報の回復可否」を参照して，最適化情報の削除要否を判断し，必要に応じて最適化情報を削除してください。最適化情報の削除については，「最適化情報の削除」を参照してください。

4. 最適化情報を回復する。

pdgetcst コマンドを使用して，最適化情報パラメタファイルによる最適化情報の登録を実施してください。詳細は，「最適化情報パラメタファイルによる最適化情報の登録」を参照してください。

コマンド実行例：

退避しておいた最適化情報パラメタファイル（c:¥tmp¥optinfo_20101014）を使用して，表 **USER1.T1** の最適化情報を回復します。このときの pdgetcst コマンドの実行例を次に示します。

```
pdgetcst -s c:¥tmp¥optinfo_20101014 -a USER1 -t T1
```

19.2.4　最適化情報の削除

(1)　最適化情報の削除の契機

最適化情報を削除する契機を次に示します。

- SQL 単位でのチューニングの運用に切り替える場合

(2)　最適化情報の削除の手順

最適化情報の削除の手順を次に示します。

＜手順＞

1. 最適化情報を削除する表を決定する。
   最適化情報を削除する表を決定してください。
   SQL 単位でのチューニングを行う場合に，その SQL でアクセスする表が削除対象となります。
2. 最適化情報を削除する。
   pdgetcst コマンドを使用して，最適化情報の削除を実施してください。
   コマンド実行例：
   表 **USER1.T1** の最適化情報を削除する場合の pdgetcst コマンドの実行例を次に示します。

```
pdgetcst -d -a USER1 -t T1
```

19.3　最適化情報パラメタファイルの運用方法

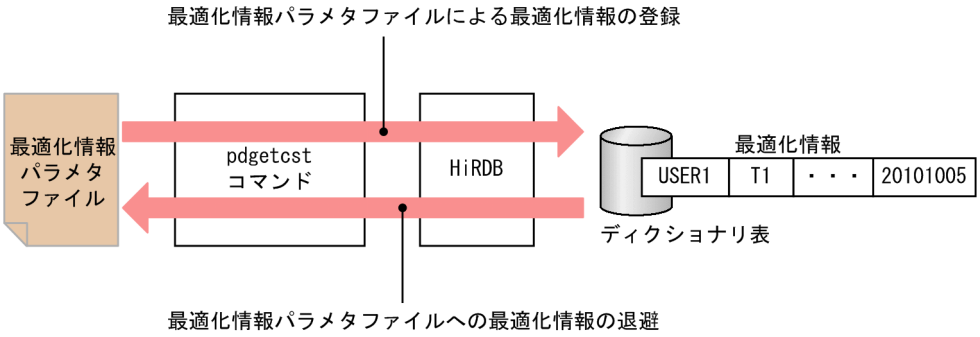
19.3.1　最適化情報パラメタファイルとは

最適化情報パラメタファイルとは，表の特性を示す情報がテキスト形式で記述されたファイルを指します。この最適化情報パラメタファイルを使用することで，ユーザが任意に記述した最適化情報をディクショナリ表に登録し，SQL の最適化に反映させることができます。

また，ディクショナリ表に登録されている最適化情報を，最適化情報パラメタファイルに退避することもできます。

最適化情報パラメタファイルの概要を次の図に示します。

図 19–3　最適化情報パラメタファイルの概要



(1)　最適化情報パラメタファイルの内容

最適化情報パラメタファイルには，表の行数，インデクスのキー値数，列値の分布などの情報が含まれています。

最適化情報パラメタファイルの具体的な項目については，マニュアル「HiRDB コマンドリファレンス」の「最適化情報パラメタファイル」を参照してください。

(2)　最適化情報パラメタファイルの作成方法

最適化情報パラメタファイルを作成するには，次の二つの方法があります。

- テキストエディタで作成する。
- pdgetcst コマンドで最適化情報パラメタファイルへの最適化情報の退避を実施する。

(a)　テキストエディタで作成する場合

最適化情報パラメタファイルの各項目に任意の指定値を設定したい場合は，テキストエディタなどを使用して作成してください。

テキストエディタなどを使用して最適化情報パラメタファイルを作成する方法については，マニュアル「HiRDB コマンドリファレンス」の「最適化情報パラメタファイル」を参照してください。

(b)　pdgetcst コマンドで最適化情報パラメタファイルへの最適化情報の退避を実施する場合

pdgetcst コマンドで最適化情報パラメタファイルへの最適化情報の退避を実施することで，ディクショナリ表に登録されている最適化情報を，最適化情報パラメタファイルに退避できます。退避する方法については，「最適化情報パラメタファイルへの最適化情報の退避」を参照してください。

19.3.2　最適化情報パラメタファイルによる最適化情報の登録

ユーザが任意に記述した最適化情報パラメタファイル，又はバックアップ用に退避した最適化情報パラメタファイルを，ディクショナリ表に登録できます。

最適化情報パラメタファイルによる最適化情報の登録は，pdgetcst コマンドで行います。

なお，最適化情報パラメタファイルを使用する際には注意点があります。詳細は「表及びインデクスの定義変更をした場合の最適化情報パラメタファイルの使用可否」を参照してください。

(1)　実行例

最適化情報パラメタファイル（c:¥tmp¥param_file）を使用して，表 **USER1.T1** の最適化情報をディクショナリ表に登録します。このときの pdgetcst コマンドの実行例を次に示します。

```
pdgetcst -s c:¥tmp¥param_file -a USER1 -t T1
```

19.3.3　最適化情報パラメタファイルへの最適化情報の退避

ディクショナリ表に登録している最適化情報を最適化情報パラメタファイルに退避できます。この機能を使用することで，次の運用ができます。

- バックアップとして保存し，最適化情報の回復に使用する。
- チューニングのために，退避した最適化情報パラメタファイルの最適化情報の任意の項目を編集して再登録する。
- テスト環境で最適化情報を退避し，本番環境のディクショナリ表へ登録する（構造や定義が同じ表（インデクスも含む）の場合に限ります）。

最適化情報パラメタファイルへの最適化情報の退避は，pdgetcst コマンドで行います。

(1) 実行例

表 **USER1.T1** の最適化情報を，最適化情報パラメタファイル（c:¥tmp¥param_file）へ退避します。このときの pdgetcst コマンドの実行例を次に示します。

```
pdgetcst -e c:¥tmp¥param_file -a USER1 -t T1
```

19.4　最適化情報を運用するときの注意事項

ここでは，最適化情報を運用するときの注意事項について説明します。

19.4.1　表及びインデクスの定義変更をした場合の最適化情報パラメタファイルの使用可否

最適化情報を収集している表，及び関連するインデクスの定義を変更すると，定義変更前に作成した最適化情報パラメタファイルが使用できなくなることがあります。最適化情報パラメタファイルが使用できない場合，最適化情報の回復，及び最適化情報パラメタファイルによる最適化情報の登録が実行できないため，退避した最適化情報パラメタファイルは削除する必要があります。

定義変更の内容と定義変更前に作成した最適化情報パラメタファイルの使用可否を次の表に示します。

表 19–1　定義変更の内容と定義変更前に作成した最適化情報パラメタファイルの使用可否

対象資源	定義変更の内容	最適化情報パラメタファイルの使用可否	備考
表	表削除（DROP TABLE）	×	なし。
	表名変更（ALTER TABLE）	○	pdgetcst コマンドを実行する場合，-t オプションには変更後の表識別子を指定する必要があります。
列	列追加（ALTER TABLE）	○	なし。
	列削除（ALTER TABLE）	×	なし。
	列属性変更（ALTER TABLE）	○	なし。
	列名変更（ALTER TABLE）	△	最適化情報パラメタファイルの変更前の列名を，変更後の列名に修正することで使用できます。
インデクス	• インデクス追加（CREATE INDEX） • 主キー追加（ALTER TABLE）	×	なし。
	• インデクス削除（DROP INDEX） • 主キー削除（ALTER TABLE）	×	なし。
	インデクス名変更（ALTER INDEX）	△	最適化情報パラメタファイルの変更前のインデクス名を，変更後のインデクス名に修正することで使用できます。

（凡例）

○：使用できます。

△：内容を編集すれば使用できます。

×：最適化情報パラメタファイルに，削除した表及びインデクスの情報がある場合はエラーとなります。これ以外の場合，使用してもエラーにはなりませんが，検索性能が低下するおそれがあります。

19.4.2　最適化情報退避後の pdgetcst の操作と最適化情報の回復可否

最適化情報の退避後に最適化情報が増加する操作を行った場合，退避した最適化情報を使って回復しても，完全に元には戻らないことがあります。

最適化情報が増加する操作を次に示します。

- 収集レベルを変更して最適化情報を収集した場合
- 異なるパラメタを記述した最適化情報パラメタファイルを登録した場合

pdgetcst の操作の組み合わせと回復可否を次の表に示します。

表 19–2　pdgetcst の操作の組み合わせと回復可否

退避時点までに行った pdgetcst の操作(1)	退避後の pdgetcst の操作(2)	回復可否	備考
最適化情報の収集（-c lvl1）	最適化情報の収集（-c lvl1）	○	なし。
	最適化情報の収集（-c lvl2）	×	(2)によって，登録する情報が増えるため，回復できません。
	最適化情報の登録（-s）	×	(2)によって，登録する情報が増える可能性があるため，回復できません。
最適化情報の収集（-c lvl2）	最適化情報の収集（-c lvl1）	○	なし。
	最適化情報の収集（-c lvl2）		
	最適化情報の登録（-s）	×	(2)によって，登録する情報が増える可能性があるため，回復できません（例えば，インデクスが定義されていない列に対して最適化情報を登録するなど）。
最適化情報の登録（-s）	最適化情報の収集（-c lvl1）	×	(2)によって，登録する情報が増える可能性があるため，回復できません。
	最適化情報の収集（-c lvl2）		
	最適化情報の登録（-s）（(1)と同じパラメタだけを登録）	○	なし。
	最適化情報の登録（-s）（(1)と異なるパラメタを登録）	×	(2)によって，登録する情報が増える可能性があるため，回復できません。

（凡例）

○：

退避した最適化情報パラメタファイルを使用して回復できます。

×：

退避した最適化情報パラメタファイルでは回復できません。

最適化情報を回復したい場合は，次の手順で回復してください。

1. 現在登録されている最適化情報を pdgetcst -d で削除する。
2. pdgetcst -s で，退避した最適化情報パラメタファイルを登録する。

なお，最適化情報を回復する場合は，バックアップとして退避している最適化情報だけが回復されます。

19.4.3　最適化情報の登録，更新，又は削除時にエラーが発生した場合

最適化情報を登録，更新，又は削除する場合，対象となる表の更新内容は，表情報，列情報，及びインデクス情報ごとにコミットします。そのため，最適化情報を登録，更新，又は削除時にエラーが発生した場合（KFPN00011-I メッセージのリターンコードが 8 の場合），コミットが終了していない情報がロールバックされ，最適化情報が正確に変更されないおそれがあります。この場合，KFPN00011-I メッセージに従ってエラー原因を取り除いた後，リターンコードが 0 になるまでコマンドを再実行してください。コマンドを再実行してもエラーが続く場合，次の対処を行って登録，更新，又は削除する前の最適化情報に戻してください。

エラー発生時の操作	対処方法
最適化情報の登録	最適化情報を削除してください。詳細は，「最適化情報の削除」を参照してください。
最適化情報の更新	最適化情報を回復してください。詳細は，「最適化情報の回復」を参照してください。
最適化情報の削除	

エラー発生時に行った操作は，KFPN00013-I メッセージの処理種別で確認できます。なお，最適化情報を登録，又は更新した場合，KFPN00013-I メッセージが出力されていなければ，最適化情報は変更されていません。そのため，上記の対処をする必要はありません。最適化情報を削除した場合は，上記の対処を行ってください。

20 障害が発生したときの対処方法

この章では，HiRDB に障害が発生したときの対処方法について説明します。

20.1　障害が発生したときの HiRDB の処理と HiRDB 管理者の処置

ここでは，障害が発生したときの HiRDB の処理と HiRDB 管理者の処置について，次の項目を説明します。

- 障害が発生したときに HiRDB 管理者がすること
- 障害が発生したときに HiRDB が取得する情報
- 障害が発生したときの HiRDB の処理
- HiRDB のサーバプロセスで障害が発生したときの対処方法
- HiRDB が再開始するときに引き継ぐ情報
- トランザクションキャンセル時のプロセスダウンメッセージ変更機能
- グローバルバッファ制御情報トレース取得機能

20.1.1　障害が発生したときに HiRDB 管理者がすること

障害が発生したときに HiRDB 管理者がすることについて説明します。

(1)　HiRDB 管理者がすること

障害が発生した場合，HiRDB 管理者は次に示すことをしてください。

1. 出力されたメッセージを確認してください。そして，マニュアル「HiRDB メッセージ」を参照して，障害の原因を調べてください。

備考

- サポートサービスを利用して障害対策をする場合に必要となる情報を「障害調査のために必要な情報」にまとめました。HiRDB 管理者（システム管理者）は必ず「障害調査のために必要な情報」を参照してください。
- 障害時の運用で，よくある質問をＱ＆Ａにまとめました。障害時のＱ＆Ａについては，「障害に関する質問」を参照してください。

(2)　pdinfoget コマンドの実行方法

pdinfoget コマンドを実行すると，pdinfoget コマンドを実行したサーバマシンのディレクトリ下にトラブルシュート情報のバックアップが取得されます。

なお，pdinfoget コマンドを実行するには，環境変数 PDDIR 及び PDCONFPATH が設定されている必要があります。設定されていることを確認してください。

次に示す手順に従って pdinfoget コマンドを実行してください。

〈手順〉

1. pdinfoget コマンドを実行するサーバマシンに，トラブルシュート情報を格納するだけのディスク容量があるかどうかを確認してください。pdinfoget -m コマンドで，取得する障害情報に必要な容量を見積もることができます。
2. pdinfoget コマンドを実行するサーバマシンに，トラブルシュート情報を格納するディレクトリを作成してください。※
3. トラブルシュート情報が不要になったら，pdcspool コマンドでトラブルシュート情報を削除してください。残しておくとディスク容量の不足原因となります。

注※

既にディレクトリがある場合は，そのディレクトリ下に次に示すディレクトリがあるかどうかを確認してください。次に示すディレクトリがあると，pdinfoget コマンドを実行したときにエラーとなります。

出力先ディレクトリ	出力先ディレクトリ下にあるとエラーになるディレクトリ
初期情報ファイル出力先ディレクトリ	CLTDIR PDDIR spool conf .dbenv pdistup uxpldir OSFILE SYSLOG COREINF
詳細情報ファイル出力先ディレクトリ	PDDIR spool tmp uxpldir CLTDIR COREINF

この場合，次のどちらかの対処をしてから再度実行してください。

- 初期情報ファイル出力先ディレクトリ又は詳細情報ファイル出力先ディレクトリに別のディレクトリを指定する
- 指定したディレクトリ内の情報が不要な場合，すべて削除する

(3) 不要なトラブルシュート情報の削除

サーバプロセスの強制終了，又はクライアントの強制終了時などに，HiRDB は%PDDIR%¥spool 下にトラブルシュート情報を出力します。また，コマンド又はユティリティを［Ctrl ＋ C］キーを押すなどして途中終了させると，%PDDIR%¥tmp 下にコマンド又はユティリティが出力した作業用一時ファイルが削除さ

れずに残ります。このトラブルシュート情報及び作業用一時ファイルを残しておくと，HiRDB 運用ディレクトリがあるディスクの容量を圧迫する原因になります。HiRDB 運用ディレクトリがあるディスクの容量が不足すると HiRDB が異常終了することがあるため，HiRDB は次に示すファイルを定期的に削除します。

- トラブルシュート情報ファイル（%PDDIR%¥spool 下のファイル）
- 作業用一時ファイル（%PDDIR%¥tmp 下のファイル）

通常は 24 時間ごとに削除します。この削除間隔を **pd_spool_cleanup_interval** オペランドで変更できます。また，**pd_spool_cleanup_interval_level** オペランドで何日より前に出力されたファイルだけを削除するという指定ができます。

このほかにも，トラブルシュート情報（%PDDIR%¥spool 下のファイル）を一括して削除する方法があります。

- **pdcspool** コマンドでトラブルシュート情報ファイルを削除できます。作業用一時ファイル（%PDDIR%¥tmp 下のファイル）も削除できます。
- HiRDB の開始時に自動的にトラブルシュート情報ファイルを削除します。**pd_spool_cleanup** オペランドでトラブルシュート情報ファイルを削除するかどうかを指定します。このオペランドを省略した場合，ファイルを削除します。また，**pd_spool_cleanup_level** オペランドで何日より前に出力されたトラブルシュート情報ファイルだけを削除するという指定ができます。

> **参考**
>
> pdcspool コマンドのオプション，pd_spool_cleanup_level，又は pd_spool_cleanup_interval_level オペランドの指定で，削除するトラブルシュート情報を選択できます。

(4)　トラブルシュート情報の出力量の削減

次に示すオペランドを指定すると，トラブルシュート情報の出力量を削減できます。必要に応じてこれらのオペランドを指定してください。

- pd_cancel_dump：トラブルシュート情報を出力するかどうかを指定します。
- pd_client_waittime_over_abort：トランザクションの実行中にクライアントの最大待ち時間（クライアント環境定義の PDCWAITTIME オペランドの値）を超えた場合にトラブルシュート情報を出力するかどうかを指定します。
- pd_debug_info_netstat：サーバプロセスの異常終了，又は HiRDB（ユニット）が異常終了したときに取得するトラブルシュート情報中に，ネットワーク情報を取得するかどうかを指定します。
- pd_dump_suppress_watch_time：トラブルシュート情報の再出力を抑止する時間を指定します。

20.1.2　障害が発生したときに HiRDB が取得する情報

障害が発生したときに HiRDB が取得するトラブルシュート情報を次の表に示します。

表 20–1　障害が発生したときに HiRDB が取得するトラブルシュート情報

取得情報	説明
メッセージログファイル（%PDDIR%¥spool¥pdlog1，pdlog2）	HiRDB が出力するメッセージです。pdcat コマンドで参照できます。障害発生時にはバックアップを取得してください。
コマンド実行時の標準出力，標準エラー出力	運用コマンドの出力情報，及びエラーメッセージです。運用コマンドを入力した画面に表示されます。ファイルに残したい場合は必要に応じてリダイレクトしてください。
イベントログ	HiRDB が出力するメッセージです。
退避コアファイル※ （%PDDIR%¥spool¥save¥ファイル名）	HiRDB の関連プロセスのデータ，及びスタック情報です。一つのサーバに３個までしか退避されないので，残しておきたい退避コアファイルは，バックアップを取得してください。 ファイル名の形式は「サーバ名 n」となります（サーバ名が不明な場合，サーバ名が数字となることがあります。サーバによってサーバ名に PID が入る場合があります）。n は退避コアファイルの通番（1〜3）です。ただし，通番は付加されないことがあります。
アボート情報ファイル※ （%PDDIR%¥spool¥save¥ファイル名）	アボート情報です。出力された場合はバックアップを取得してください。アボートコードだけならメモ帳などで参照できます。ファイル名の形式は「abcode.サーバプロセスのプロセス ID」となります。
障害時のスナップ※ （%PDDIR%¥spool¥save¥ファイル名）	障害時のスナップ情報です。出力された場合はバックアップを取得してください。ファイル名の形式は「サーバ名 n .deb」となります（サーバ名が不明な場合，サーバ名が数字となることがあります）。n は退避コアファイルの通番（1〜3）です。ただし，通番は付加されないことがあります。
共用メモリダンプファイル※ （%PDDIR%¥spool¥pdshmdump¥ファイル名）	HiRDB が共用メモリに保持するデータです。出力された場合はバックアップを取得してください。ファイル名は「shmdump*」，「サーバ名.rmb.サーバプロセスのプロセス ID」，「サーバ名.ucb.サーバプロセスのプロセス ID」などの形式となります。
簡易ダンプファイル※ （%PDDIR%¥spool¥各サーバ対応のディレクトリ¥ファイル名）	HiRDB が共用メモリ及びプロセス固有メモリに保持するデータです。出力された場合はバックアップを取得してください。ファイル名は日付とプロセス ID を組み合わせた形式になります。
RPC トレースファイル	HiRDB で RPC を使用して送受信する電文です。出力された場合はバックアップを取得してください。ファイル名は pd_rpc_trace_name オペランドで指定します。
コマンドトレースファイル （%PDDIR%¥spool¥cmdlog¥cmdlog1，cmdlog2）	実行したコマンド（HiRDB が内部で生成するコマンドも含む）の履歴情報です。メモ帳などで参照してください。障害発生時にはバックアップを取得してください。
エラーログファイル （%PDDIR%¥spool¥errlog¥errlog1，errlog2）	HiRDB が出力する内部情報です。出力された場合はバックアップを取得してください。

取得情報	説明
接続ユーザ情報ファイル（%PDDIR%¥spool¥cnctusrinf）	HiRDB 終了時の接続ユーザ情報です。メモ帳などで参照してください。
接続ユーザ詳細ファイル（%PDDIR%¥spool¥cnctusrdtl）	
デッドロック・タイムアウト情報ファイル（%PDDIR%¥spool¥pdlckinf¥出力日時）	HiRDB の排他制御でデッドロック，排他待ちタイムアウトエラーが発生したときのユーザ情報です。メモ帳などで参照してください。出力された場合はバックアップを取得してください。
排他資源管理テーブル情報ファイル（%PDDIR%¥spool¥pdlckinf¥出力日時.mem）	HiRDB の排他制御で排他資源管理テーブル不足が発生したときのユーザ情報です。メモ帳などで参照してください。出力された場合はバックアップを取得してください。
デッドロック SQL 情報ファイル（%PDDIR%¥spool¥pdlckinf¥出力日時_サーバ名_トランザクション識別子_スレッド ID.sql）	HiRDB の排他制御でデッドロックが発生したとき，デッドロックを構成するトランザクションが実行した SQL の情報です。メモ帳などで参照してください。出力された場合はバックアップを取得してください。
PRF トレース情報ファイル（%PDDIR%¥spool¥pdprf¥ファイル名）	HiRDB が出力するトラブルシュート用のトレース情報です。pdprfed コマンドで編集出力して参照してください。ファイル名の形式は「pdprf_nnn」となります。nnn は，ファイルの世代番号です。また，PRF トレースの出力ディレクトリは，pd_prf_output_directory オペランドで変更できます。

注※

HiRDB のサーバプロセスが異常終了したときに出力するトラブルシュート情報です。HiRDB のサーバプロセスが異常終了したことは KFPS01820-E メッセージの出力によって確認できます。KFPS01820-E メッセージからはサーバ名，プロセス ID，終了状態（end state）を確認できます。ただし，次に示す場合はトラブルシュート情報を出力しません。

1. 終了状態（end state）の最初が c 又は d で始まる場合は，トラブルシュート情報を出力しません。
2. 終了状態（end state）が 007f の場合は，アボート情報ファイル，退避コアファイル，及び共用メモリダンプファイルを出力しません。

20.1.3　障害が発生したときの HiRDB の処理

障害が発生したときの HiRDB の処理概要について説明します。

(1)　障害が発生したときの影響範囲

ハードウェア及びソフトウェアの障害によって，HiRDB の異常終了につながる障害が発生した場合，HiRDB/パラレルサーバは一部の例外を除いてその影響範囲をユニット単位に局所化し，HiRDB を構成するすべてのユニットが全面停止しないで，ユニット単位で異常終了します。HiRDB/シングルサーバの場合，一つのユニットから構成されているため，ユニットの異常終了は HiRDB の全面停止につながります。

(2)　HiRDB が異常終了したときのシステムの回復

HiRDB が異常終了した場合，障害の原因を取り除いた後に HiRDB を再開始すると，HiRDB はシステムの状態を障害発生時点に回復します。

(3)　連続して異常終了したときの処理

HiRDB が異常終了しても，システム共通定義の pd_mode_conf オペランドの指定によっては，HiRDB は自動的にユニットを再開始します。ただし，３回連続して異常終了すると，自動的に再開始しなくなります。

この場合，HiRDB 管理者は障害の原因を対策した後，pdstart コマンドで HiRDB を再開始してください。

20.1.4　サーバプロセスで障害が発生したときの対処方法

HiRDB のサーバプロセスで障害が発生した場合，HiRDB はそのサーバプロセスを異常終了します。このとき，HiRDB はトランザクション回復プロセスを起動してサーバごとにロールバックします。その後，該当するサーバプロセスが自動的に再起動されるため，HiRDB 管理者は特に対処する必要がありません。

ただし，障害の影響度によっては，HiRDB はそのサーバプロセスを実行しているユニットを異常終了します。このとき，HiRDB 管理者は異常終了したユニットを再開始する必要があります。ただし，pd_mode_conf オペランドの指定によっては自動的にユニットが再開始されます。

次に，サーバプロセスの異常終了によって，ユニットが異常終了したときの HiRDB 管理者の処置について説明します。

(1)　HiRDB/シングルサーバの場合

HiRDB/シングルサーバが異常終了したときの HiRDB 管理者の処置を次の表に示します。

表 20–2　HiRDB/シングルサーバが異常終了したときの HiRDB 管理者の処置

HiRDB の処理	HiRDB 管理者の処置
HiRDB/シングルサーバを異常終了します。	障害の原因を取り除いて，HiRDB/シングルサーバを再開始してください。ただし，pd_mode_conf オペランドの指定によっては自動的に HiRDB/シングルサーバが再開始されます。その後，処理中の操作を再実行してください。

(2)　HiRDB/パラレルサーバの場合

HiRDB/パラレルサーバが異常終了したときの HiRDB 管理者の処置を次の表に示します。

表 20–3　HiRDB/パラレルサーバが異常終了したときの HiRDB 管理者の処置

障害発生状況	HiRDB の処理	HiRDB 管理者の処置
ユニットコントローラの処理プロセスで障害が発生したとき	該当するプロセスがあるユニットを異常終了します。	該当するユニットを再開始してください。ただし，pd_mode_conf オペランドの指定によっては自動的にユニットが再開始されます。
システムマネジャプロセスで障害が発生したとき	システムマネジャがあるユニットを異常終了します。システムマネジャが停止している間は，運用コマンド及び UAP の実行を受け付けません。また，その間のメッセージログを取得できません。	障害の原因を取り除いて，該当するユニットを再開始してください。ただし，pd_mode_conf オペランドの指定によっては自動的にユニットが再開始されます。その後，処理中の操作を再実行してください。
フロントエンドサーバプロセスで障害が発生したとき	フロントエンドサーバがあるユニットを異常終了します。フロントエンドサーバが停止している間は，処理要求を受け付けません。タイムアウトレベルの通信エラーとします。 また，フロントエンドサーバプロセスが異常終了した場合，HiRDB は該当するプロセスの回復処理をします。このとき，ユニットを停止する必要がある障害が発生していれば，該当するプロセスがあるユニットを異常終了します。	障害の原因を取り除いて，該当するユニットを再開始してください。ただし，pd_mode_conf オペランドの指定によっては自動的にユニットが再開始されます。その後，処理中の操作を再実行してください。
ディクショナリサーバプロセスで障害が発生したとき	ディクショナリサーバがあるユニットを異常終了します。	
バックエンドサーバプロセスで障害が発生したとき	バックエンドサーバがあるユニットを異常終了します。	

20.1.5　HiRDB が再開始するときに引き継ぐ情報

HiRDB が再開始するときに引き継ぐ情報を次の表に示します。

表 20–4　HiRDB が再開始するときに引き継ぐ情報

分類		引き継ぐ情報
システムファイル	システムログファイル	システムの回復情報
	シンクポイントダンプファイル	表の回復情報
	ステータスファイル	1. システムの開始形態を決定する情報 ・サーバ構成 2. 各システムサーバの状態の情報 ・オープン／クローズ状態 ・現用／待機（交代）状態 ・ファイルの閉塞状態 ・サービスグループの閉塞状態

分類		引き継ぐ情報
		3. RD エリアに関する情報
ユーザデータ （データベース）	RD エリア	• オープン／クローズ状態 • 閉塞状態※
運用コマンド （右記のコマンドによって変化する状態を引き継ぐ）	データベースを操作するコマンド	• pdclose（RD エリアのクローズ） • pdopen（RD エリアのオープン） • pdhold（RD エリアの閉塞）※ • pdrels（RD エリアの閉塞解除）
	システムログファイル又はシンクポイントダンプファイルを操作するコマンド	• pdlogadpf（ファイルの割り当て） • pdlogchg（ファイルの状態変更） • pdlogcls（ファイルのクローズ） • pdlogopen（ファイルのオープン） • pdlogswap（ファイルのスワップ） • pdlogunld（ファイルのアンロード）
	ステータスファイルを操作するコマンド	• pdstscls（ファイルのクローズ） • pdstsopen（ファイルのオープン） • pdstsswap（ファイルのスワップ）
	監査証跡の出力を操作するコマンド	• pdaudbegin コマンド（監査証跡の取得開始） • pdaudend コマンド（監査証跡の取得終了）

注

次に示すコマンドの情報は引き継がないため，HiRDB の再開始後に再度入力してください。

- pdstbegin（統計情報の取得開始）
- pdstend（統計情報の取得終了）

注※

次に示す閉塞状態は引き継ぎません。

- 参照可能バックアップ閉塞（更新 WAIT モード）
- 更新可能バックアップ閉塞
- 更新可能バックアップ閉塞（WAIT モード）

20.1.6　トランザクションキャンセル時のプロセスダウンメッセージ変更機能

(1)　概要

クライアントでの割り込みに伴う強制終了要求によって，トランザクション実行中のサーバプロセスを停止した場合や，pdcancel コマンドによって UAP と接続していたサーバプロセスを終了した場合，HiRDB では，サーバプロセスダウンを示す KFPS01820-E 及び KFPO00105-E メッセージを出力します。このメッ

セージは，サーバプロセスで何らかの異常が発生し，サーバプロセスを停止した場合にも出力するメッセージですが，サーバプロセスの終了原因を調べるためには，KFPS00993-I メッセージなど，HiRDB が出力する他メッセージの情報も調べる必要があります。

トランザクションキャンセル時のプロセスダウンメッセージ変更機能を使用すると，クライアント操作やクライアント側の障害に対して出力する KFPS01820-E 及び KFPO00105-E メッセージを，別のメッセージ ID に変更できます。そのため，サーバプロセスの終了原因がクライアント操作やクライアント側の障害であることを，メッセージだけで判断できるようになります（他メッセージの情報を調べる必要はありません）。

(2)　機能

この機能を使用した場合，メッセージ ID だけを変更します。メッセージテキスト及び出力情報は変更しません（変更対象メッセージのメッセージテキストと出力情報を，変更後のメッセージでもそのまま出力します）。

(a)　変更の対象となるメッセージ

この機能で変更対象とするメッセージを次の表に示します。

表 20–5　メッセージ ID が変更されるメッセージ

変更対象のメッセージ ID	メッセージの意味	変更後のメッセージ ID
KFPS01820-E	サーバプロセスが停止しました。	KFPS01852-W
KFPO00105-E	異常が発生したため，サーバプロセスを停止しました（アボートメッセージ）。	KFPO00115-W

メッセージの詳細については，マニュアル「HiRDB メッセージ」を参照してください。

(b)　メッセージ変更有無

サーバプロセスダウンの発生事象とメッセージ ID の変更有無を次の表に示します。

表 20–6　サーバプロセスダウンの発生事象とメッセージ ID の変更有無

項番	サーバプロセスダウンの発生事象	ダウンの要因						メッセージの変更	備考
		ユーザオペレーション	環境設定	クライアント側障害	クライアントとの通信障害	サーバ側障害	サーバ間通信障害		
1	クライアントでの割り込みに伴う強制終了要求	○	−	−	−	−	−	あり	クライアントでの割り込みに伴う強制終了要求によって，トランザクション実行中の HiRDB サーバプロセスが停止しています。※1

項番	サーバプロセスダウンの発生事象	ダウンの要因						メッセージの変更	備考
		ユーザオペレーション	環境設定	クライアント側障害	クライアントとの通信障害	サーバ側障害	サーバ間通信障害		
2	クライアントプロセスダウンに伴うサーバプロセスの kill	−	−	○	○	−	−	あり	XA を利用しているクラインアントプロセスは，項番 11 に該当します。
3	PDCWAITTIME オーバによる HiRDB サーバプロセスの kill	−	○※2	−	○	○	−	なし	HiRDB で原因の特定ができません。※3
4	PDSWAITTIME オーバによる HiRDB サーバプロセスダウン（exit）	−	○	○	○	−	−	あり	−
5	PDSWATCHTIME オーバによる HiRDB サーバプロセスダウン（exit）	−	○	○	○	−	−	あり	−
6	ユニット強制終了，ユニットダウン，又は系切り替え時のトランザクション回復に伴う HiRDB サーバプロセスの kill	○	−	−	−	−※4	−※4	あり	他ユニットがブランチ回復する場合に該当します。また，HiRDB/パラレルサーバだけ該当します。
7	ユニット強制終了，ユニットダウン，又は系切り替え時のユティリティブランチ回復に伴う HiRDB サーバプロセスの kill	○	−	−	−	−※4	−※4	あり	自ユニット内の回復だけでなく，他ユニットがブランチ回復する場合も含みます。
8	pdcancel コマンドによる HiRDB サーバの kill（ユティリティの kill も含む）	○	−	−※4	−	−※4	−※4	あり	−
9	pdfgt コマンドによるトランザクション強制停止に伴う HiRDB サーバプロセスの kill（トランザクションブランチ回復）	○	−	−※4	−	−※4	−※4	あり	−
10	システムログファイルの空き容量監視機能による HiRDB サーバプロセスの kill	○※5	○※6	−	−	−	−	あり	KFPS01160-E メッセージが出力されます。

項番	サーバプロセスダウンの発生事象	ダウンの要因						メッセージの変更	備考
		ユーザオペレーション	環境設定	クライアント側障害	クライアントとの通信障害	サーバ側障害	サーバ間通信障害		
11	トランザクションマネジャがトランザクション回復延長の XA 決着指示による HiRDB サーバプロセスの kill	○	−	−	−	○	○	なし	HiRDB では原因の特定ができません。※7
12	シンクスキップ回数が境界値を超えた場合の回復に伴う HiRDB サーバプロセスの kill	○※8	○※6	−	−	○	−	なし	HiRDB では原因の切り分けができません。
13	HiRDB サーバプロセス間の通信タイムアウトに伴う HiRDB サーバプロセスダウン（exit）	−	−	−	−	○	○	なし	−
14	回復不要 FES 利用時のダウン中ユニットからの通信要求処理	−	−	−	−	○	○	なし	−
15	回復不要 FES 利用時のトランザクション決着エラーに伴うサーバプロセスダウン（exit）	−	−	−	−	○	○	なし	−
16	サーバ側障害に伴う自動ログアンロード機能停止時の HiRDB サーバプロセスダウン	−	−	−	−	○	○	なし	−
17	サーバプロセス障害ダウン（内部処理障害，プロセス障害，トランザクション開始処理エラーなど）	−	−	−	−	○	○	なし	−
18	HiRDB サーバプロセス障害ダウン発生に伴うトランザクション及びユティリティブランチ回復時の HiRDB サーバプロセスの kill	−	−	−	−	−※4	−※4	あり	項番 17 の影響による回復処理です。

（凡例）

○：サーバプロセスダウンの原因と考えられます。

–：サーバプロセスダウンの原因ではありません。

あり：メッセージを変更します。

なし：メッセージを変更しません。

注※1

DBPARTNER 又は DABroker を使用して，HiRDB にアクセスする UAP，及び ODBC 経由で HiRDB にアクセスする UAP が該当します。また，該当する UAP をキーボードの Ctrl+C を押して停止する場合も含みます。

注※2

トランザクション処理時間未満の場合に該当します。

注※3

環境設定によるタイムアウト，又はサーバ側障害のどちらが発生したかを HiRDB が判断できないため，ユーザが判断する必要があります。このため，メッセージの出力が必要です。

注※4

障害事象ではありませんが，障害事象を発生させる原因になる可能性があります。

注※5

トランザクション量が予想を超えた場合に該当します。

注※6

境界値が不正な場合に該当します。

注※7

トランザクションマネジャからの決着指示が要因となります。トランザクションマネジャは，クライアント側障害，ユーザによるキャンセル，又はサーバ側障害によってトランザクションが中断した場合に決着指示を出しますが，どの要因で決着指示が出されたか HiRDB では判断できません。

注※8

発行した SQL が不正な場合に該当します。

(c)　必要なオペランド

この機能を使用する場合は，pd_cancel_down_msgchange オペランドを省略するか，又は Y を指定してください。

20.1.7　グローバルバッファ制御情報トレース取得機能

グローバルバッファの障害調査に必要な資料を取得するための機能です。

通常は，この機能を適用する必要はありません。障害発生時に，保守員の指示に従って適用してください。

(1)　シンクポイント出力同期制御情報取得機能

(a)　概要

この機能を適用した場合，次の情報の出力，及び各項目のチェックを実行します。

●デファードライトプロセス，HiRDB サーバプロセス共通の実行内容

モジュールトレースに，次の情報を出力します。

- セマフォ値
- 内部フラグ（管理テーブル状態フラグ）の履歴
- セマフォ発行履歴

●デファードライトプロセスの実行内容

次の二つの情報をチェックします。

- セマフォ発行時の値
- 内部フラグ（管理テーブル状態フラグ）を ON に変更した回数

これらの内容が一致しない場合，KFPH23203-E メッセージを出力します。

●HiRDB サーバプロセスの実行内容

次の二つの情報をチェックします。

- トランザクション終了時及びセマフォ発行時に内部フラグ（管理テーブル状態フラグ）を OFF に変更した回数
- セマフォ発行回数

これらの内容が一致しない場合，HiRDB は次の処理を行います。

- KFPH23204-E メッセージを出力します。
- デファードライトプロセスを強制終了して core を取得し，HiRDB サーバプロセスを停止します。

(b)　準備作業

この機能を適用する場合は，次の手順でモジュールトレース領域サイズを拡張してください。

1. HiRDB を正常終了します。
2. システム共通定義%PDCONFPATH%¥pdsys に，次のオペランドを追加します。

```
set pd_module_trace_max = 16383
```

指定値はモジュールトレースのエントリ数を示します（1 エントリ当たり 48 バイト）。

3. HiRDB を正常開始します。

注　モジュールトレース領域サイズを拡張すると，拡張した分のメモリ量が HiRDB サーバプロセスごとに増えます（デフォルトでは，126 エントリが設定されます）。

(c) メッセージ出力時の対処方法

メッセージが出力された場合は，次の方法で対処してください。

●出力メッセージが KFPH23203-E の場合

デファードライトプロセスで内部矛盾を検知したことを示します。保守員の指示に従って，システム情報を採取してください。

●出力メッセージが KFPH23204-E の場合

HiRDB サーバプロセスで内部矛盾を検知したことを示します。%PDDIR%¥spool ディレクトリに出力されるトラブルシュート情報を採取してください。

(d) 注意事項

- デファードライト処理の並列 WRITE 機能を使用していない場合，デファードライトプロセス一つにつき，次のメモリ所要量が増えます。

  《32 ビットの場合》

  320 バイト×サーバに定義したグローバルバッファ数

  《64 ビットの場合》

  640 バイト×サーバに定義したグローバルバッファ数

- デファードライト処理の並列 WRITE 機能を使用している場合，並列 WRITE 機能を使用しているサーバ一台につき，次の共用メモリ所要量が増えます。

  《32 ビットの場合》

  320 バイト×サーバに定義したグローバルバッファ数

  《64 ビットの場合》

  640 バイト×サーバに定義したグローバルバッファ数

20.2　UAP が正しく実行されないときの対処方法

実行者　HiRDB 管理者

ここでは，UAP が正しく実行されないときの対処方法について説明します。

20.2.1　UAP を実行できないときの対処方法

UAP を実行できないのは，次の表に示す原因が考えられます。

表 20–7　UAP を実行できないときに考えられる原因とその対処方法

考えられる原因	対処方法
HiRDB システム定義に誤りがあります。	HiRDB システム定義に誤りがある旨のメッセージが出力されます。そのメッセージを参考にして，HiRDB システム定義を修正してください。
メモリが不足しています。	メモリが不足している旨のメッセージが出力されます。共用メモリの場合は，HiRDB システム定義を見直してください。プロセス固有メモリの場合は，必要のないプロセスを停止してください。
最大同時接続数を超えています。	• pd_max_users オペランドの指定値を大きくしてください。 • pdchprc コマンドでフロントエンドサーバ又はシングルサーバのサーバプロセスの最大数を変更している場合は，最大数を大きくしてください。
クライアント環境定義の環境変数の設定に誤りがあります。	マニュアル「HiRDB UAP 開発ガイド」を参照して，環境変数を正しく指定してください。
HiRDB が開始していません。	• HiRDB を開始してください。 HiRDB/パラレルサーバの場合，ユニット又はサーバ単位に停止していることがあります。この場合，ユニット又はサーバ単位に開始してください。 • HiRDB を開始してから，トランザクションの受け付けができるようになるまで時間差があります。この間に接続した場合，接続がエラーとなります。pdls コマンドで HiRDB の状態を確認して再接続してください。

20.2.2　UAP が終了しないときの対処方法

UAP の実行状況を調べてください。UAP の実行状況を調べる方法については，「UAP 又はユティリティの実行時間が長い場合」を参照してください。そして必要ならば，**pdcancel** コマンドで UAP を強制終了させてください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

20.2.3　UAP が異常終了したときの対処方法

UAP が異常終了した場合，**pdls -d prc** コマンドで，その UAP のサーバプロセスが残っていないか確認してください。サーバプロセスが残っている場合は，**pdcancel** コマンドでその処理プロセスを終了させてください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

20.3　運用コマンドが正しく実行されないときの対処方法

ここでは，運用コマンドが正しく実行されないときの対処方法について説明します。

20.3.1　運用コマンドを実行できないときの対処方法

実行者　運用コマンド実行者又は HiRDB 管理者

運用コマンドを実行できないのは，次の表に示す原因が考えられます。

表 20–8　運用コマンドを実行できないときに考えられる原因とその対処方法

考えられる原因	対処方法
運用コマンドのオプション又は引数の指定に誤りがあります。	オプション又は引数の指定を修正して，再度運用コマンドを実行してください。
運用コマンドの実行権限がありません。	マニュアル「HiRDB コマンドリファレンス」を参照して，自分に運用コマンドの実行権限があるか確認してください。実行権限がない場合は，実行権限のある人にコマンドを実行してもらうか，又は実行権限を与えてもらってください。
HiRDB 稼働時にしか実行できないコマンドを，HiRDB 停止中に実行しました。	マニュアル「HiRDB コマンドリファレンス」を参照して，実行しようとしたコマンドが HiRDB 停止中でも実行できるか確認してください。HiRDB の稼働中にしか実行できないコマンドの場合は，HiRDB を開始して実行してください。
運用コマンドの実行環境が設定されていません。	次に示す環境設定が正しいか確認してください。 • 環境変数の設定 上記の環境設定方法については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
HiRDB システム定義に誤りがあります。	マニュアル「HiRDB システム定義」を参照して，HiRDB システム定義を修正してください。

20.3.2　運用コマンドが応答待ちタイムアウトになったときの対処方法

実行者　運用コマンド実行者

運用コマンドが応答待ちタイムアウトになるのは，次に示す原因が考えられます。

- OS の負荷が高いため，応答待ち時間以内に処理が完了しません。
- HiRDB が終了しているのに運用コマンドを実行しています。

OS の負荷が高い場合，再度，運用コマンドを実行してください。それでも応答待ちタイムアウトになった場合は，必要のないプロセスを停止して，もう一度運用コマンドを実行してください。

20.4　HiRDB が開始できないときの対処方法

ここでは，HiRDB が開始できないときの対処方法について説明します。説明する項目は次のとおりです。

- HiRDB が正常開始できないときの対処方法
- HiRDB が再開始できないときの対処方法
- マスタディレクトリ用 RD エリアに障害が発生したときの対処方法

20.4.1　HiRDB が正常開始できないときの対処方法

実行者　HiRDB 管理者

HiRDB が正常開始できないのは，次の表に示す原因が考えられます。

表 20–9　HiRDB が正常開始できないときに考えられる原因とその対処方法

考えられる原因	対処方法
HiRDB が正しくインストール又はセットアップされていません。	HiRDB のインストール又はセットアップをし直してください。
HiRDB システム定義に誤りがあります。	HiRDB システム定義に誤りがある旨のメッセージが出力されます。そのメッセージを参考にして，HiRDB システム定義を修正してください。
環境変数の設定に誤りがあります。	環境変数を正しく設定してください。環境変数の設定方法については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
メモリ又はファイル容量が不足しています。	メモリ又はファイル容量が不足している旨のメッセージが出力されます。不要なプロセスを停止させるか，又は不要なファイルを削除してください。なお，共用メモリの場合は，HiRDB システム定義を見直してください。プロセス固有メモリの場合は，必要のないプロセスを停止してください。
HiRDB の開始に必要なファイルがありません。	HiRDB の開始に必要なファイルがない旨のメッセージが出力されます。そのメッセージを参考にして，必要なファイルを作成してください。
HiRDB の開始に必要なファイルに障害が発生しました。	メッセージを参照して障害が発生したファイルを調べてください。その後，「システムログファイルに障害が発生したときの対処方法」～「ファイル障害が発生したとき（システムファイル以外のファイル）の対処方法」を参照して対策してください。
OS の構成が HiRDB の実行環境として不適当です。	OS を構築し直して，HiRDB システム定義を修正してください。
現用のシステムログファイルを割り当てられません。	アンロード待ち状態のシステムログファイルを pdlogunld 又は pdlogchg コマンドで，アンロード済み状態にしてください。
特定のユニットに障害が発生しています（HiRDB/パラレルサーバ限定）。	障害などで起動できないユニットがある場合，縮退起動をすると，残りの正常なユニットだけで HiRDB を開始できます。縮退起動の方法については，「縮退起動をするときの運用方法（HiRDB/パラレルサーバ限定）」を参照してください。

備考

HiRDB 開始時の障害で，よくある問い合わせをＱ＆Ａにまとめました。HiRDB 開始時の障害のＱ＆Ａについては，「HiRDB の開始に関する質問」を参照してください。

20.4.2　HiRDB が再開始できないときの対処方法

実行者　HiRDB 管理者

HiRDB の再開始に失敗した場合，再開始処理時に出力されるメッセージを参照してください。再開始できないのは，次の表に示す原因が考えられます。

表 20–10　HiRDB が再開始できないときに考えられる原因とその対処方法

考えられる原因	対処方法
現用のシステムログファイルの両系に障害が発生したため，HiRDB を再開始できません。	「現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法」を参照してください。
スワップ先にできる状態のシステムログファイルがないため，HiRDB を再開始できません。	「システムログファイルの不足によって HiRDB（ユニット）が異常終了したときの対処方法」を参照してください。
シンクポイントダンプファイルに対応するシステムログファイルが上書きされているため，HiRDB を再開始できません。	「シンクポイントダンプファイルに対応するシステムログファイルが上書きされているため HiRDB を再開始できないときの対処方法」を参照してください。
シンクポイントダンプファイルの数が不足しているため，HiRDB を再開始できません。	「シンクポイントダンプファイルの数が不足しているため HiRDB を再開始できないときの対処方法」を参照してください。
現用のステータスファイルの両系に障害が発生したため，HiRDB を再開始できません。	「現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法」を参照してください。
特定のユニットに障害が発生しています（HiRDB/パラレルサーバ限定）。	障害などで起動できないユニットがある場合，縮退起動をすると，残りの正常なユニットだけで HiRDB を開始できます。縮退起動の方法については，「縮退起動をするときの運用方法（HiRDB/パラレルサーバ限定）」を参照してください。
マスタディレクトリ用 RD エリアに障害が発生したため，HiRDB を再開始できません。	「マスタディレクトリ用 RD エリアに障害が発生したときの対処方法」を参照してください。

20.4.3　マスタディレクトリ用 RD エリアに障害が発生したときの対処方法

マスタディレクトリ用 RD エリアに障害が発生した場合，HiRDB は再開始できません。この場合，次に示す手順で RD エリアを回復してください。

〈手順〉

1. **pdstart -r** コマンドで HiRDB を開始します。
2. **pdrstr** コマンドで，マスタディレクトリ用 RD エリアを回復します。

pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01
-l C:¥unld¥unldlog01,C:¥unld¥unldlog02 -w C:¥tmp¥sortwork¥ -r PDBMAST

3. **pdstop** コマンドで HiRDB を終了させます。
4. **pdstart** コマンドで HiRDB を開始してください。
5. **pdclose** コマンドで障害が発生した RD エリア（マスタディレクトリ用 RD エリアを除く）をクローズします。
   pdclose -r rdarea01,rdarea02
6. データベース回復ユティリティ（**pdrstr** コマンド）で，障害が発生した RD エリアを回復します。
   pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01
   -l C:¥unld¥unldlog01,C:¥unld¥unldlog02 -w C:¥tmp¥sortwork¥ -r rdarea01,rdarea02
7. **pdrels -o** コマンドで，回復した RD エリアの障害閉塞を解除して，オープンします。
   pdrels -r rdarea01,rdarea02 -o

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

20.5　HiRDB が終了できないときの対処方法

実行者　HiRDB 管理者

ここでは，HiRDB が終了できないときの対処方法について説明します。HiRDB が終了できないのは，次の表に示す原因が考えられます。

表 20–11　HiRDB が終了できないときに考えられる原因とその対処方法

考えられる原因	対処方法
接続中のユーザが存在します。	接続中のユーザが存在すると，HiRDB を終了できません。対処方法については，「接続中のユーザが存在するため HiRDB が終了できないときの対処方法」を参照してください。
未決着状態のトランザクションがあります。	未決着状態のトランザクションがあると，HiRDB を終了できません。対処方法については，「未決着状態のトランザクションがあるときの対処方法」を参照してください。
通信障害が発生しています。	HiRDB/パラレルサーバの場合，通信障害によってサーバマシン間の連絡が取れずに終了できないことがあります。対処方法については，「通信障害が発生したときの対処方法」を参照してください。

20.6　システムログファイルに障害が発生したときの対処方法

ここで説明する項目は次のとおりです。

- 現用ファイルに障害が発生したときの対処方法
- HiRDB Datareplicator を使用している場合の対処方法
- 現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法

20.6.1　現用ファイルに障害が発生したときの対処方法

(1)　HiRDB の稼働中に障害が発生した場合

HiRDB の稼働中に現用ファイルに障害が発生した場合の対処方法を次の表に示します。

表 20–12　現用ファイルに障害が発生した場合の対処方法（HiRDB の稼働中）

障害発生時の条件		HiRDB の処理	HiRDB 管理者の処置
書き込み時	スワップ先にできる状態のファイルがある場合	システムログファイルをスワップします。 障害が発生したシステムログファイルの状態を予約とし，スワップ先にできる状態のファイルの一つを現用にして，処理を続行します。	• 業務履歴が必要な場合は，予約となったファイルを pdlogunld コマンドでアンロードしてください。 • 障害が発生したファイルを(3)に示す方法でスワップ先にできる状態にしてください。
	スワップ先にできる状態のファイルがない場合	障害が発生したファイルがあるユニットを異常終了します。	「システムログファイルの不足によって HiRDB（ユニット）が異常終了したときの対処方法」を参照してください。 • スワップ先にできる状態のファイルを作成してください。その後，HiRDB を再開始してください。 • 障害が発生したファイルを(3)に示す方法でスワップ先にできる状態にしてください。
読み込み時	システムログファイルを二重化している場合	正常な系のファイルに切り替えて，処理を続行します。	障害が発生した系のファイルを(3)に示す方法でスワップ先にできる状態にしてください。
	システムログファイルを二重化していない場合	障害が発生したシステムログファイルの状態を予約とし，処理を中止します。	• 障害が発生したシステムログファイルをアンロードしていなければ，pdlogunld コマンドでアンロードしてください。※ • バックアップ及び作成したアンロードログファイルを入力情報としてデータベース回復ユティリティでデータベースを回復してください。データベースの回復方法については，「データベースの回復方法」を参照してください。 • 障害が発生したシステムログファイルを(3)に示す方法でスワップ先にできる状態にしてください。

注※

アンロード状態のチェックを解除する運用をしている場合（pd_log_unload_check=N を指定している場合）は，必要ない操作です。

(2)　HiRDB の再開始処理中に障害が発生した場合

HiRDB の再開始処理中に現用ファイルに障害が発生した場合の対処方法を次の表に示します。

表 20–13　現用ファイルに障害が発生した場合の対処方法（HiRDB の再開始処理中）

障害発生時の条件	HiRDB の処理	HiRDB 管理者の処置
システムログファイルを二重化している場合	正常な系のファイルに切り替えて，再開始処理を続行します。	HiRDB の再開始後，障害が発生した系のファイルを(3)に示す方法でスワップ先にできる状態にしてください。
システムログファイルを二重化していない場合	障害が発生したシステムログファイルの状態を予約とし，障害が発生したシステムログファイルのユニットの再開始処理を終了します。	• 障害が発生したシステムログファイルをアンロードしていなければ，pdlogunld コマンドでアンロードします。その後，ユニットを強制開始します。 • バックアップと作成したアンロードログファイルを入力情報としてデータベース回復ユティリティでデータベースを回復してください。データベースの回復方法については，「データベースの回復方法」を参照してください。 • 障害が発生したシステムログファイルを(3)に示す方法でスワップ先にできる状態にしてください。

(3)　障害が発生したファイルをスワップ先にできる状態にする方法

〈手順〉

1. **pdlogls** コマンドで，障害発生のため予約となったシステムログファイルを確認します。
   pdlogls -d sys -s b001
2. **pdlogrm** コマンドで予約ファイルを削除します。
   pdlogrm -d sys -s b001 -f C:¥sysfile¥syslog1a
   pdlogrm -d sys -s b001 -f C:¥sysfile¥syslog1b
3. **pdloginit** コマンドで，2 で削除したシステムログファイルを再作成します。
   pdloginit -d sys -s b001 -f C:¥sysfile¥syslog1a -n 5000
   pdloginit -d sys -s b001 -f C:¥sysfile¥syslog1b -n 5000
4. **pdlogopen** コマンドで，3 で再作成したシステムログファイルをスワップ先にできる状態にしてください。
   pdlogopen -d sys -s b001 -g syslog01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(4)　ディスク障害が発生したときの対処方法

ディスク障害が発生したときの対処方法を例題形式で説明します。

例題

ディスク障害が発生して現用ファイルの片系が障害状態になりました。スワップ先にできる状態のファイルがあるため，HiRDB はシステムログファイルのスワップ処理を実施して稼働中のままです。

対処方法

このままオンラインを続行した場合，システムログファイルの全体容量に余裕がないと，システムログファイルの容量不足が起こり，ユニットが異常終了する可能性があります。したがって，早急にディスク交換をして〈手順１〉に示す方法で対処してください。

なお，ディスク交換がすぐにできない場合は，〈手順２〉に示す方法で対処してください。また，〈手順１〉又は〈手順２〉の方法をすぐにとれない場合は，〈手順３〉に示す方法で対処してください。

〈手順１〉ディスク交換がすぐにできる場合

1. **pdstop** コマンドで HiRDB を正常終了します。
2. ディスクを交換し，**pdfmkfs** コマンドで HiRDB ファイルシステム領域を作成します。
3. **pdloginit** コマンドでシステムログファイルを作成します。
4. **pdstart** コマンドで HiRDB を正常開始します。

〈手順２〉ディスク交換がすぐにできない場合

1. **pdstop** コマンドで HiRDB を正常終了します。
2. 空きのある HiRDB ファイルシステム領域に，**pdloginit** コマンドでシステムログファイルを作成します。
3. サーバ定義の次に示すオペランドを修正します。追加したシステムログファイルに対応するオペランドを追加します。
   ・pdlogadfg
   ・pdlogadpf
4. **pdstart** コマンドで HiRDB を正常開始します。

〈手順３〉手順１又は２の方法をすぐにとれない場合

1. **pdstop** コマンドで HiRDB を正常終了します。
2. 障害が発生したシステムログファイルに対応しているサーバ定義で，pd_log_singleoperation=Y（システムログファイルの片系運転をすること）を指定します。
3. **pdstart** コマンドで HiRDB を正常開始します。

注意事項

〈手順３〉の方法をとった場合，システムログファイルの片系運転中にシステムログファイルに障害が発生すると，アンロードログ（システムログ）を使用したデータベースの

回復ができなくなります。したがって，〈手順３〉の方法は一時的なものとし，早急に〈手順１〉又は〈手順２〉の方法をとるようにしてください。

20.6.2　HiRDB Datareplicator を使用している場合の対処方法

HiRDB Datareplicator 連携を実行しているときに，データ連動に必要なシステムログファイルの入力エラー（システムログファイルを二重化して運用しているときは両系障害）が発生すると，データ連動ができなくなります。この場合，次に示すことをしてください。

〈手順〉

1. **pdrplstop** コマンドを入力して HiRDB Datareplicator 連携を停止してください。
2. システムログファイルの障害を取り除いた後に，HiRDB Datareplicator 連携を再開始する場合，**pdrplstart** コマンドを入力してください。ただし，HiRDB Datareplicator 連携をいったん中止すると，データ連動の対象となっている抽出側データベースと反映側データベースとの間で不整合が生じるため，pdrplstart コマンドを入力する前に，必ず抽出側データベースを基に反映側データベースを再作成してください。

20.6.3　現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法

現用ファイルの両系障害が発生すると，HiRDB（ユニット）の再開始に必要な情報が失われるため，HiRDB を再開始できなくなります。この場合，**pdstart dbdestroy** コマンドで HiRDB を強制開始してください。

このとき，HiRDB は前回稼働時の情報を引き継ぎません。したがって，HiRDB 管理者がデータベースの内容を回復する必要があります。データベースの内容を回復するには，バックアップ及びシステムログ（アンロードログ）を入力情報として，データベース回復ユティリティを実行します。データベースの回復方法については，「データベースの回復方法」を参照してください。

なお，HiRDB を強制開始する前に，「HiRDB（ユニット）を強制開始するときの注意事項」を参照してください。

注意事項

- データベースの内容を回復するまでは，そのデータベースをアクセスできない状態（RD エリアを pdhold -c コマンドで閉塞かつクローズ状態）にしておいてください。
- HiRDB を強制開始すると，前回の HiRDB 開始後に更新したすべての RD エリア（システム用 RD エリアも含みます）が破壊されます。したがって，強制開始をする場合は，破壊された RD エリアをデータベース回復ユティリティで回復する必要があります。RD エリアを回復しないと，その後の HiRDB の動作を保証できません。

20.7　シンクポイントダンプファイルに障害が発生したときの対処方法

20.7.1　障害発生時の対処方法

シンクポイントダンプファイルに障害が発生したときの対処方法を次の表に示します。

表 20–14　シンクポイントダンプファイルに障害が発生したときの対処方法

障害発生時の条件		HiRDB の処理	HiRDB 管理者の処置
書き込み時	上書きできる状態のファイルがある場合	障害が発生したシンクポイントダンプファイルの状態を予約とし，上書きできる状態のファイルの 1 つをシンクポイントダンプの出力先にして処理を続行します。 **●シンクポイントダンプファイルを二重化している場合** A 系ファイルと B 系ファイルのどちらか片方に障害が発生しても，正常な系だけで処理を続行します。両方に障害が発生した場合は，障害が発生したシンクポイントダンプファイルの状態を予約とし，上書きできる状態のファイルの 1 つをシンクポイントダンプの出力先にして処理を続行します。	障害が発生したシンクポイントダンプファイルを(1)に示す方法で上書きできる状態にしてください。
	上書きできる状態のファイルがない場合	障害が発生したシンクポイントダンプファイルがあるユニットを異常終了します。	• 新規のシンクポイントダンプファイルを作成した後に，ユニットを再開始してください。 • ユニットの再開始後，障害が発生したシンクポイントダンプファイルを(1)に示す方法で上書きできる状態にしてください。
読み込み時	最新世代のファイルを読み込めない場合は，1 世代前のファイルを読み込みます。1 世代前のファイルも読み込めない場合は，もう 1 世代前のファイルを読み込みます。このように，ファイルを読み込めない場合は世代をさかのぼっていきます。ただし，有効保証世代数を超えてさかのぼると，システムの回復に必要なシステムログが上書きされていることがあるため，システムを回復できないことがあります。 **●シンクポイントダンプファイルを二重化している場合** A 系ファイルを読み込めない場合は B 系ファイルを読み込みます。B 系ファイルも読み込めない場合は，1 世代前の A 系ファイルを読み込みます。		障害が発生したシンクポイントダンプファイルを(1)に示す方法で上書きできる状態にしてください。

(1)　障害が発生したファイルを上書きできる状態にする手順

〈手順〉

1. **pdlogls** コマンドで，障害が発生して予約となったシンクポイントダンプファイルを確認します。
   pdlogls -d spd -s b001
2. 障害が発生したファイルが予約になっていない場合，そのファイルを **pdlogcls** コマンドで予約にします。

pdlogcls -d spd -s b001 -g spdfile1

3. **pdlogrm** コマンドで予約ファイルを削除します。

   pdlogrm -d spd -s b001 -f C:¥sysfile¥sync01

4. **pdloginit** コマンドで，3 で削除したシンクポイントダンプファイルを再作成します。

   pdloginit -d spd -s b001 -f C:¥sysfile¥sync01 -n 5000

5. **pdlogopen** コマンドで，4 で再作成したシンクポイントダンプファイルを上書きできる状態にしてください。

   pdlogopen -d spd -s b001 -g spdfile1

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) シンクポイントダンプファイルに対応するシステムログファイルが上書きされているため HiRDB を再開始できないときの対処方法

シンクポイントダンプファイルに障害が発生すると，HiRDB はシンクポイントダンプファイルの世代をさかのぼってシステムの状態を回復しようとします。このとき，シンクポイントダンプファイルに対応するシステムログファイル(システムを回復するために必要な情報を持つシステムログファイル)が上書きされていると，HiRDB を再開始できません。この場合，**pdstart dbdestroy** コマンドで HiRDB を強制開始してください。

このとき，HiRDB は前回稼働時の情報を引き継ぎません。したがって，HiRDB 管理者がデータベースの内容を回復する必要があります。データベースの内容を回復するには，バックアップ及びシステムログ（アンロードログ）を入力情報として，データベース回復ユティリティを実行します。データベースの回復方法については，「データベースの回復方法」を参照してください。

なお，HiRDB を強制開始する前に，「HiRDB（ユニット）を強制開始するときの注意事項」を参照してください。

注意事項

- データベースの内容を回復するまでは，そのデータベースをアクセスできない状態（RD エリアを pdhold -c コマンドで閉塞かつクローズ状態）にしておいてください。
- HiRDB を強制開始すると，前回の HiRDB 開始後に更新したすべての RD エリア（システム用 RD エリアも含みます）が破壊されます。したがって，強制開始をする場合は，破壊された RD エリアをデータベース回復ユティリティで回復する必要があります。RD エリアを回復しないと，その後の HiRDB の動作を保証できません。
- RD エリアはシステムログだけで回復できます。前回の pdstart コマンドが失敗したときに出力された KFPS01262-I メッセージを参照し，メッセージに表示されているログ読み込み開始のファイルグループ名，及びそれ以降に発生したシステムログをデータベース回復ユティリティの入力情報にしてください。

(3) シンクポイントダンプファイルの数が不足しているため HiRDB を再開始できないときの対処方法

シンクポイントダンプファイルの数が有効保証世代数以下となると，HiRDB を再開始できません。この場合，次に示すどちらかの対処をして，HiRDB を再開始してください。

- 障害が発生したシンクポイントダンプファイルを再作成してください。
- pdlogadfg -d spd オペランドに ONL 指定をしていないシンクポイントダンプファイルがあれば，ONL を指定してください。

なお，あらかじめ，次に示すオペランドを指定しておくと HiRDB が停止している時間を短縮できます。

- pd_spd_reduce_mode ＝ 1 又は 2
- pd_spd_reserved_file_auto_open ＝ Y

20.8　ステータスファイルに障害が発生したときの対処方法

ここでは，ステータスファイルに障害が発生したときの対処方法について説明します。ここで説明する項目は次のとおりです。

- 現用ファイルに障害が発生したときの対処方法
- 障害が発生したステータスファイルがある状態で HiRDB（ユニット）を開始するときの手順
- 現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法

20.8.1　現用ファイルに障害が発生したときの対処方法

現用ファイルに障害が発生したときの対処方法を次の表に示します。

表 20–15　現用ファイルに障害が発生したときの対処方法

項番	障害発生時の条件		HiRDB の処理	HiRDB 管理者の処置
1	予備のファイルがある場合		ステータスファイルをスワップします。このとき，KFPS01062-I メッセージが出力されます。 障害が発生したファイルを閉塞とし，予備のファイルの一つを現用にして処理を続行します。	閉塞しているステータスファイルを(1)に示す方法で予備にしてください。
2	予備のファイルがない場合	ステータスファイルの片系運転を指定している場合※1	ステータスファイルを片系運転にし，処理を続行します。このとき，KFPS01044-I メッセージが出力されます。	早急に予備のファイルを作成して，ステータスファイルを両系運転の状態に戻してください。 (1)又は(2)に示す方法で予備のファイルを作成してください。その後，(3)に示す方法で予備のファイルを現用にしてください。
3		ステータスファイルの片系運転を指定していない場合※2	障害が発生したステータスファイルがあるユニットを異常終了します。	新規のステータスファイルを作成した後に，ユニットを再開始してください。ユニットの再開始方法については，「障害が発生したステータスファイルがある状態で HiRDB（ユニット）を開始するときの手順」を参照してください。 その後，閉塞しているステータスファイルを(1)に示す方法で予備にしてください。
4	片系運転中の障害，又は両方の系のファイルが障害の場合		障害が発生したステータスファイルがあるユニットを異常終了します。	ユニットを再開始するための情報が失われるため，ユニットを再開始できません。この場合，「現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法」を参照してください。 なお，予備のステータスファイルがない場合は，ユニットを強制開始する前に，新規のステータスファイルを作成しておいてください。

注※1

ユニット制御情報定義に pd_syssts_singleoperation ＝ continue を指定した場合，又はサーバ定義に pd_sts_singleoperation ＝ continue を指定した場合です。

注※2

ユニット制御情報定義に pd_syssts_singleoperation ＝ stop（省略値）を指定した場合，又はサーバ定義に pd_sts_singleoperation ＝ stop（省略値）を指定した場合です。

(1) 閉塞のファイルを予備にする方法

〈手順〉

1. **pdls** コマンドで，閉塞しているステータスファイルを確認します。
   ユニット用ステータスファイルの場合：
   pdls -d sts -u UNT1
   サーバ用ステータスファイルの場合：
   pdls -d sts -s b001
2. 閉塞しているステータスファイルが配置されているディスクに障害が発生しているかを確認してください。障害が発生している場合は障害を取り除いてください。物理的な障害（破損，電源の寸断など）のほかにも，OS やディスクドライバの障害，ディスクが有効化されているかなどについても調査してください。
3. **pdstsrm** コマンドで，閉塞しているステータスファイルを削除します。
   ユニット用ステータスファイルの場合：
   pdstsrm -u UNT1 -f C:¥sysfile¥usts1a
   pdstsrm -u UNT1 -f C:¥sysfile¥usts1b
   サーバ用ステータスファイルの場合：
   pdstsrm -s b001 -f C:¥sysfile¥sstsb1a
   pdstsrm -s b001 -f C:¥sysfile¥sstsb1b
4. **pdstsinit** コマンドで，3 で削除したステータスファイルを再作成します。
   ユニット用ステータスファイルの場合：
   pdstsinit -u UNT1 -f C:¥sysfile¥usts1a -l 4096 -c 256
   pdstsinit -u UNT1 -f C:¥sysfile¥usts1b -l 4096 -c 256
   サーバ用ステータスファイルの場合：
   pdstsinit -s b001 -f C:¥sysfile¥sstsb1a -l 4096 -c 256
   pdstsinit -s b001 -f C:¥sysfile¥sstsb1b -l 4096 -c 256
5. **pdstsopen** コマンドで，4 で再作成したステータスファイルを予備にします。
   ユニット用ステータスファイルの場合：
   pdstsopen -u UNT1 -f C:¥sysfile¥usts1a
   pdstsopen -u UNT1 -f C:¥sysfile¥usts1b

サーバ用ステータスファイルの場合：

pdstsopen -s b001 -f C:¥sysfile¥sstsb1a

pdstsopen -s b001 -f C:¥sysfile¥sstsb1b

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(2) 予約のファイルを予備にする方法

〈手順〉

1. **pdls** コマンドで，予約のステータスファイルを確認します。

   ユニット用ステータスファイルの場合：

   pdls -d sts -u UNT1

   サーバ用ステータスファイルの場合：

   pdls -d sts -s b001

2. **pdstsopen** コマンドで予約ファイルを予備にします。

   ユニット用ステータスファイルの場合：

   pdstsopen -u UNT1 -n usts1a

   サーバ用ステータスファイルの場合：

   pdstsopen -s b001 -n sstsb01

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(3) 予備のファイルを現用にする方法

〈手順〉

1. **pdstsswap** コマンドで予備のファイルを現用にします。

   ユニット用ステータスファイルの場合：

   pdstsswap -u UNT1

   サーバ用ステータスファイルの場合：

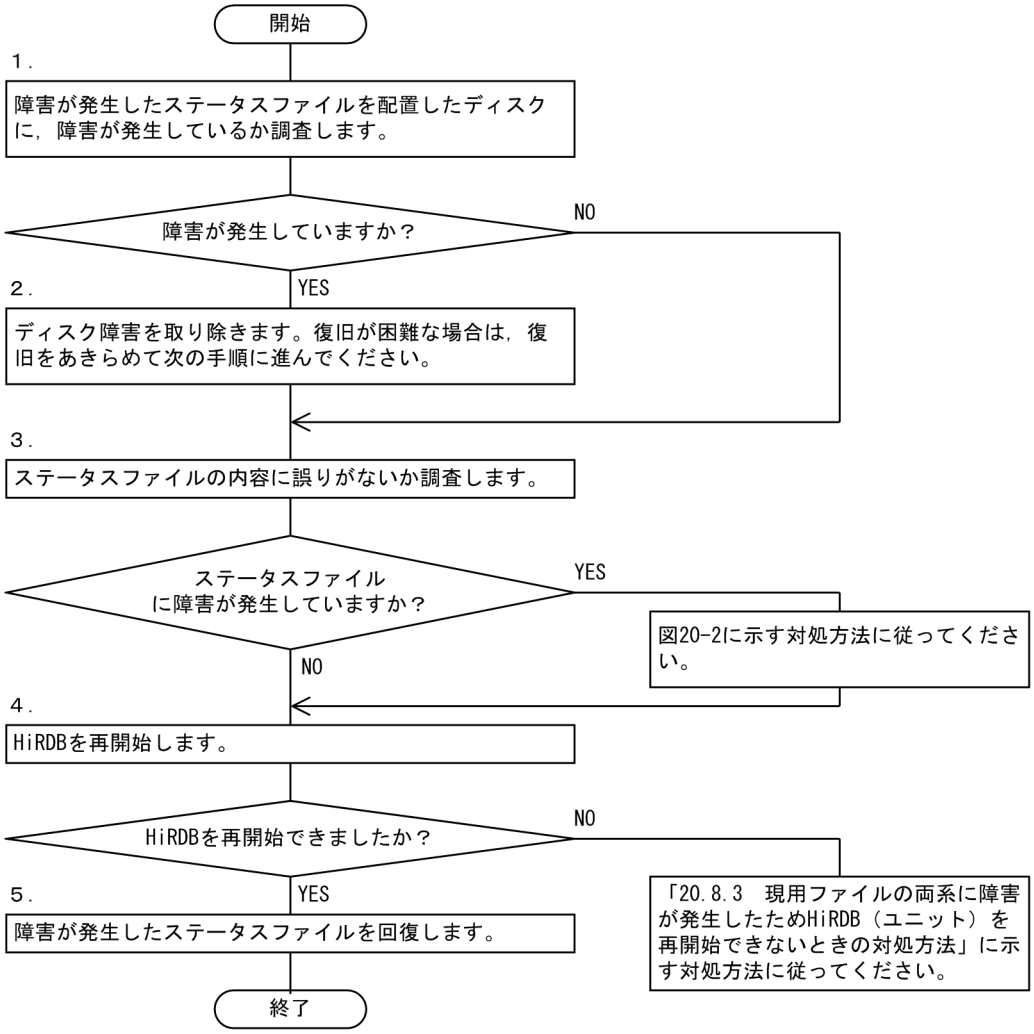
   pdstsswap -s b001

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

20.8.2　障害が発生したステータスファイルがある状態で HiRDB（ユニット）を開始するときの手順

障害が発生したステータスファイルがある状態で HiRDB（ユニット）を開始するときの対処方法を次の図に示します。

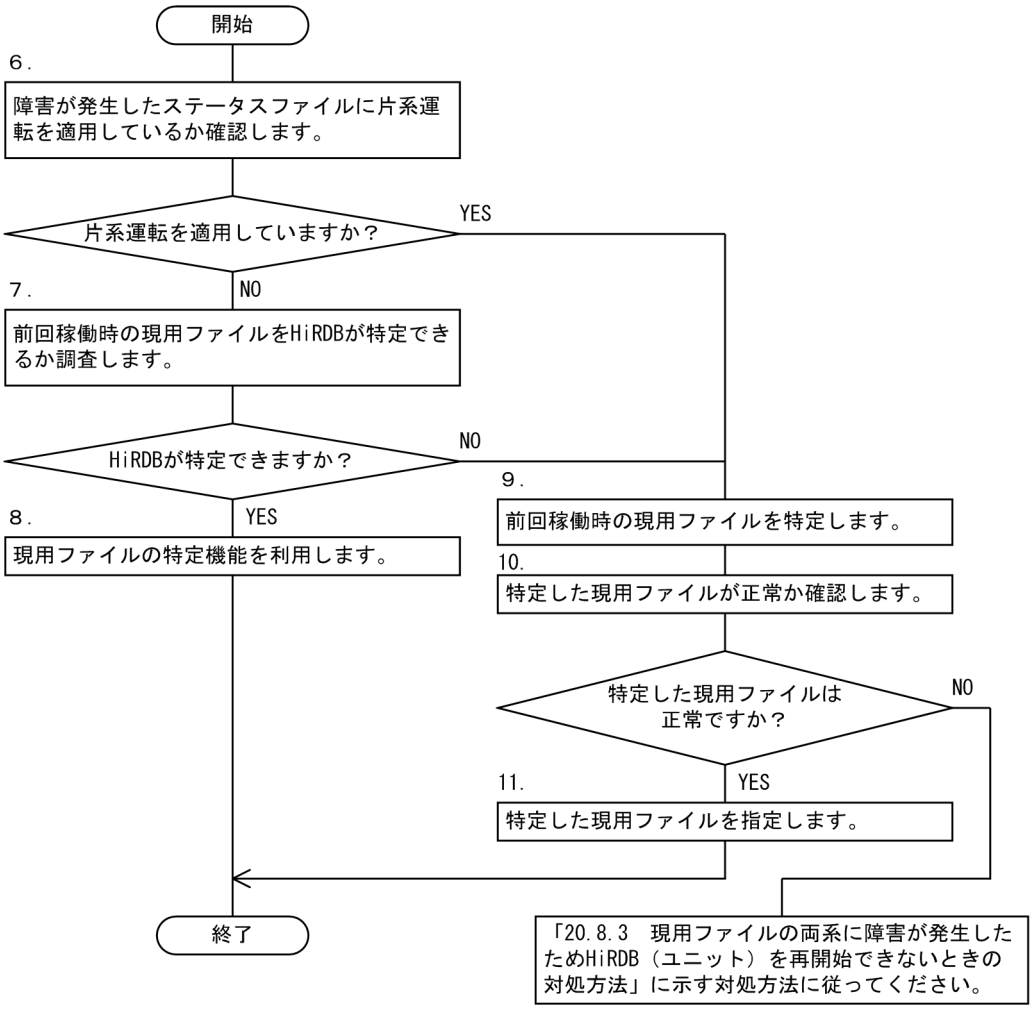
図 20–1　障害が発生したステータスファイルがある状態で HiRDB（ユニット）を開始するときの対処方法



注

処理ボックスの上にある数字はこの後で説明している（　）レベルに対応しています。例えば，5 の操作は(5)で説明しています。

図 20–2　ステータスファイルに障害が発生したときの対処方法



注

処理ボックスの上にある数字はこの後で説明している（　）レベルに対応しています。例えば，8 の操作は(8)で説明しています。

(1)　ディスクに障害が発生しているか調査する

障害が発生したステータスファイルを配置したディスクに，障害が発生しているか調査してください。物理的な障害（破損，電源断など）のほかにも，OS やディスクドライバの障害，ディスクが有効化されているかなどについて調査してください。

ディスクの物理障害の判定方法を次の表に示します。

表 20–16　ディスクの物理障害の判定方法（物理障害チェック）

ディスク障害の発生	物理障害の回復	ステータスファイルのデータ	判定結果
発生していない	−	−	物理障害なし
発生している	回復できる	データが残っている	
		データが消失している	物理障害あり（実体なし）

ディスク障害の発生	物理障害の回復	ステータスファイルのデータ	判定結果
	回復できない	−	

（凡例）−：該当しません。

注意事項

ディスク障害の有無に関係なく，障害の回復が完了するまでは指示以外の契機で pdstsinit，pdstsrm，及び pdfmkfs コマンドを使用しないでください。

(2)　ディスク障害を取り除く

調査した結果，ディスク障害が発生している場合は障害を取り除いください。障害の回復が難しい場合は，ディスク障害の回復をあきらめて残りの正常なディスクだけで HiRDB を開始します。

(3)　ステータスファイルの内容に誤りがないか調査する

ステータスファイルの内容に誤りがないか調査してください。論理障害の判定方法を次の表に示します。

表 20–17　論理障害の判定方法（論理障害チェック）

コマンド実行	コマンド表示内容（ファイル作成時の指定値との比較）	判定結果
正常終了	誤りなし	論理障害なし
	誤りあり	論理障害あり
異常終了(エラーメッセージが出力される)	−	論理障害あり

（凡例）−：該当しません。

物理障害が発生していないステータスファイルに pdcat コマンドを実行し，ステータスファイルの内容に誤りがないか調べます。次に示す条件をすべて満たす場合，ステータスファイルは正常です。

- pdcat コマンドの実行結果に表示されるレコード長とレコード数が，ステータスファイル作成時に指定した値と同じである
- pdcat コマンドの実行時にエラーメッセージが表示されない

pdcat コマンドの実行例を次に示します。

```
pdcat -d sts -u UNT1 -f C:¥sysfile¥usts1a -v      ...1
pdcat -d sts -s b001 -f C:¥sysfile¥sstsb1a -v      ...2
```

〔説明〕

1. ユニット用ステータスファイルに対するコマンド実行例です。

2. サーバ用ステータスファイルに対するコマンド実行例です。

物理障害及び論理障害のどちらも発生していない場合は，次の手順に進んでください。物理障害又は論理障害のどちらかが発生している場合は，図「ステータスファイルに障害が発生したときの対処方法」に示す対処方法に従ってください。

(4) HiRDB を再開始する

pdstart コマンドで HiRDB を再開始してください。再開始できない場合は，「現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法」に示す対処方法に従ってください。

(5) 障害が発生したステータスファイルを回復する

現用ファイルの片系に障害が発生している場合は，表「現用ファイルに障害が発生したときの対処方法」の項番 2 に示す HiRDB 管理者の処置を至急行ってください。

障害によって閉塞しているファイルがある場合は，「閉塞のファイルを予備にする方法」に示す手順で，閉塞しているファイルを予備ファイルにしてください。

全ステータスファイルを回復した後，必要に応じて HiRDB を一度終了して次に示すオペランドの指定値を元に戻してください。その後，HiRDB を開始してください。

- pd_syssts_initial_error（ユニット用ステータスファイルの場合）
- pd_syssts_singleoperation（ユニット用ステータスファイルの場合）
- pd_sts_initial_error（サーバ用ステータスファイルの場合）
- pd_sts_singleoperation（サーバ用ステータスファイルの場合）

図「ステータスファイルに障害が発生したときの対処方法」に示す対処方法に従って次に示すオペランドの指定を行った場合は，全ステータスファイルを回復した後，HiRDB を正常停止して次に示すオペランドの指定を省略してください。その後，HiRDB を開始してください。障害から復旧した後も次に示すオペランドを指定したままだと現用のステータスファイルを認識できず HiRDB の開始に失敗するおそれがあります。

- pd_syssts_last_active_file
- pd_syssts_last_active_side
- pd_sts_last_active_file
- pd_sts_last_active_side

(6) 片系運転を適用しているか確認する

障害が発生したステータスファイルに片系運転を適用しているか確認してください。障害が発生したステータスファイルに対して，次に示すオペランドを指定している場合は片系運転を適用しています。

- pd_syssts_singleoperation=continue（ユニット用ステータスファイルの場合）

- pd_sts_singleoperation=continue（サーバ用ステータスファイルの場合）

(7)　前回稼働時の現用ファイルを HiRDB が特定できるか調査する

(1)～(3)の結果から，障害が発生したステータスファイルの論理ファイルごとに，A 系と B 系の状態が次の表に示す状態であるか確認してください。ステータスファイルの状態が次の表に示すどれかの場合は，前回稼働時の現用ファイルを HiRDB が特定できません。

表 20–18　HiRDB が前回稼働時の現用ファイルを特定できないケース

A 系の状態	B 系の状態
論理障害あり	論理障害あり
論理障害あり	物理障害あり（実体なし）
物理障害あり（実体なし）	論理障害あり
物理障害あり（実体なし）	物理障害あり（実体なし）

(8)　現用ファイルの特定機能を利用する

HiRDB の現用ファイルの特定機能を利用します。該当するステータスファイルに対して次に示すオペランドを指定してください。

- pd_syssts_initial_error=excontinue（ユニット用ステータスファイルの場合）
- pd_sts_initial_error=excontinue（サーバ用ステータスファイルの場合）

(9)　前回稼働時の現用ファイルを特定する

前回稼働時の（最新の）現用ファイルを特定してください。前回稼働時の現用ファイルは次に示すメッセージから特定できます。メッセージログファイル又はイベントログ中の次に示すメッセージを検索してください（現用ファイルを特定できないユニット又はサーバのメッセージを検索してください）。

- KFPS01001-I（現用ファイルが割り当てられたときに出力されるメッセージ）
- KFPS01044-I（現用ファイルが片系運転になったときに出力されるメッセージ）
- KFPS01063-I（ステータスファイルのスワップで現用ファイルが変わったときに出力されるメッセージ）

これらのメッセージのうち，最後に出力されたメッセージを参照してください。出力されたメッセージ中に現用ファイルが表示されています。

(10)　特定した現用ファイルが正常か確認する

(9)で特定した現用ファイルが正常か確認してください。現用ファイルが正常かどうかは，(1)～(3)の結果から確認できます。

前回稼働時にステータスファイルを片系運転していた場合（(9)のメッセージのうち，最後に出力されたメッセージが KFPS01044-I の場合），KFPS01044-I メッセージに表示されている運転中の系のステータスファイルが正常であるか確認してください。

前回稼働時にステータスファイルを片系運転していなかった場合（(9)のメッセージのうち，最後に出力されたメッセージが KFPS01001-I 又は KFPS01063-I の場合），KFPS01001-I 又は KFPS01063-I メッセージに表示されているステータスファイルのどちらかの系が正常であるか確認してください。

現用ファイルが正常（ステータスファイルを片系運転していなかった場合はどちらかの系が正常）の場合は次の手順に進んでください。

現用ファイルに障害が発生していた場合は，前回稼働時の現用ファイルが失われているため，HiRDB を再開始できません。この場合は，「現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法」に示す対処方法に従ってください。

(11)　特定した現用ファイルを指定する

特定した前回稼働時の現用ファイルを次に示すオペランドに指定してください。

●ユニット用ステータスファイルに障害が発生している場合

該当するユニットに次に示すオペランドを指定してください。

- pd_syssts_initial_error=continue 又は excontinue
- pd_syssts_last_active_file=前回稼働時の現用ステータスファイル名※
- pd_syssts_last_active_side=前回稼働時に正常だった系※

●サーバ用ステータスファイルに障害が発生している場合

該当するサーバに次に示すオペランドを指定してください。

- pd_sts_initial_error= continue 又は excontinue
- pd_sts_last_active_file=前回稼働時の現用ステータスファイル名※
- pd_sts_last_active_side=前回稼働時に正常だった系※

注※　(9)と(10)で特定した現用ファイル名と正常な系を指定します。

20.8.3　現用ファイルの両系に障害が発生したため HiRDB（ユニット）を再開始できないときの対処方法

現用ファイルが両系とも障害の場合，HiRDB（ユニット）は再開始できません。この場合，**pdstart dbdestroy** コマンドで HiRDB を強制開始してください。

注意事項

HiRDB を強制開始すると，HiRDB は前回稼働時の情報を引き継ぎません。したがって，**HiRDB 管理者がデータベースの内容を回復する必要があります**。データベースの内容を回復するには，バックアップ及びシステムログ（アンロードログ）を入力情報として，データベース回復ユティリティ（pdrstr コマンド）を実行します。

現用ファイルの両系に障害が発生した状態で，HiRDB を開始するときの手順を次に示します。

〈手順〉

1. ディスク障害が発生しているかを調べます（物理障害チェック）。

   ステータスファイルが配置されているディスクに障害が発生しているかを調べてください。物理的な障害（破損，電源の寸断など）のほかにも，OS やディスクドライバの障害，ディスクが有効化されているかなどについて調べてください。

   調べた結果，ディスク障害が発生している場合は障害を取り除いてください。障害回復が難しい場合は，ディスク障害の回復をあきらめて，残りの正常なディスクだけで HiRDB を開始します。このまま次の手順に進んでください。

   なお，物理障害が発生しているかどうかは，表「ディスクの物理障害の判定方法（物理障害チェック）」を参照して確認してください。

2. ステータスファイルの内容に誤りがないか調べます（論理障害チェック）。

   物理障害が発生していないステータスファイルに **pdcat -d sts** コマンドを実行し，ステータスファイルの内容に誤りがないか調べてください。

   ユニット用ステータスファイルの場合：

   pdcat -d sts -u UNT1 -f C:¥sysfile¥usts1a -v

   サーバ用ステータスファイルの場合：

   pdcat -d sts -s b001 -f C:¥sysfile¥sstsb1a -v

   次に示す条件をすべて満たす場合，ステータスファイルは正常です。

   ・pdcat コマンドの実行結果に表示されるレコード長とレコード数が，ステータスファイル作成時に指定した値と同じである

   ・pdcat コマンドの実行時にエラーメッセージが表示されない

3. **pdstsrm** コマンドで，障害が発生したステータスファイルを削除します。

   ユニット用ステータスファイルの場合：

   pdstsrm -u UNT1 -f C:¥sysfile¥usts1a

   pdstsrm -u UNT1 -f C:¥sysfile¥usts1b

   サーバ用ステータスファイルの場合：

   pdstsrm -s b001 -f C:¥sysfile¥sstsb1a

   pdstsrm -s b001 -f C:¥sysfile¥sstsb1b

4. **pdstsinit** コマンドで，3 で削除したステータスファイルを再作成します。

ユニット用ステータスファイルの場合：

pdstsinit -u UNT1 -f C:¥sysfile¥usts1a -l 4096 -c 256

pdstsinit -u UNT1 -f C:¥sysfile¥usts1b -l 4096 -c 256

サーバ用ステータスファイルの場合：

pdstsinit -s b001 -f C:¥sysfile¥sstsb1a -l 4096 -c 256

pdstsinit -s b001 -f C:¥sysfile¥sstsb1b -l 4096 -c 256

5. **pdstart dbdestroy** コマンドで，HiRDB を強制開始してください。
6. データベース回復ユティリティ（pdrstr コマンド）で，RD エリアを回復してください。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

注意事項

HiRDB を強制開始すると，前回の HiRDB 開始後に更新したすべての RD エリア（システム用 RD エリアも含みます）が破壊されます。したがって，強制開始をする場合は，破壊された RD エリアをデータベース回復ユティリティで回復する必要があります。RD エリアを回復しないと，その後の HiRDB の動作を保証できません。

20.9　ファイル障害が発生したとき（システムファイル以外のファイル）の対処方法

ここでは，次に示すファイルに入出力障害などのファイル障害が発生したときの対処方法について説明します。

- HiRDB システム定義ファイル
- メッセージログファイル
- 統計ログファイル
- データ連動用連絡ファイル（HiRDB Datareplicator）

20.9.1　HiRDB システム定義ファイルに障害が発生したときの対処方法

HiRDB の処理

HiRDB を開始しません。

HiRDB 管理者の処置

新しく HiRDB システム定義ファイルを作成し，HiRDB システム定義を定義してください。その後，HiRDB を正常開始してください。

20.9.2　メッセージログファイルに障害が発生したときの対処方法

HiRDB の処理

HiRDB のメッセージはイベントログにも出力されます。ただし，各プログラムのメッセージも出力されるため，高トラフィック時には，メッセージが失われる場合があります。

HiRDB 管理者の処置

次に示す手順でこの状態に対処してください。

〈手順〉

1. **pdstop** コマンドで HiRDB を終了します。
2. メッセージログファイルの障害原因を取り除きます。
3. **pdstart** コマンドで HiRDB を開始します。

20.9.3　統計ログファイルに障害が発生したときの対処方法

HiRDB の処理

統計ログを取得しないで，処理を続行します。このとき，KFPS05360-E メッセージが出力されます。

HiRDB 管理者の処置

次に示す手順でこの状態に対処してください。

〈手順〉

1. 統計ログファイルの障害原因を取り除きます。
2. **pdstjswap** コマンドで統計ログの出力先を切り替えます。
3. **pdstbegin** コマンドで統計ログの取得を再開します。
4. **pdls -d stj** コマンドで，統計ログが取得されているかを確認します。

20.9.4 データ連動用連絡ファイルに障害が発生したときの対処方法（HiRDB Datareplicator）

HiRDB の処理

HiRDB Datareplicator 連携を実行しているときに，抽出状態の連絡に使用しているデータ連動用連絡ファイルが初期化されたり，データ連動用連絡ファイルのオープンエラーや入出力エラーが発生すると，データ連動ができなくなります。HiRDB がデータ連動用連絡ファイルの障害を検知した場合，HiRDB Datareplicator 連携を中止して，HiRDB だけで運用を続行します。

HiRDB 管理者の処置

次に示す手順でこの状態に対処してください。

〈手順〉

1. **pdls -d rpl** コマンドを入力して，HiRDB ユニットが HiRDB Datareplicator 連携を実行中かどうかを確認してください。
2. 実行中であれば，**pdrplstop** コマンドを入力して HiRDB Datareplicator 連携を終了させてください。
3. データ連動用連絡ファイルの障害が取り除いてください。
4. HiRDB Datareplicator 連携を再開始する場合は，**pdrplstart** コマンドを入力してください。※

注※

HiRDB Datareplicator 連携をいったん中止すると，データ連動の対象となっている抽出側データベースと反映側データベースとの間で不整合が生じるため，pdrplstart コマンドを入力する前に，必ず抽出側データベースを基に反映側データベースを再作成してください。反映側データベースの再作成については，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

20.10　OS が異常終了したときの対処方法

20.10.1　HiRDB の処理及び HiRDB 管理者の処置

ここでは，OS が異常終了したときの HiRDB の処理と，HiRDB 管理者の処置について説明します。

(1)　HiRDB の処理

OS に障害が発生して OS が異常終了すると，OS が異常終了したサーバマシンのユニットが異常終了します。

(2)　HiRDB 管理者の処置

OS がリブートされた後に次に示すことをしてください。

〈手順〉

1. pdinfoget コマンドで，%PDDIR%¥spool 及び%PDDIR%¥tmp 下に出力されるトラブルシュート情報のバックアップを取得します。

   pdinfoget コマンドの実行方法については，「障害が発生したときに HiRDB 管理者がすること」を参照してください。

   障害発生時に出力されるトラブルシュート情報については，「障害が発生したときに HiRDB が取得する情報」を参照してください。

2. **pdstart** コマンドで，HiRDB を再開始します。ただし，pd_mode_conf オペランドの指定によっては，OS のリブート後，自動的にユニットが再開始されます。

20.11　OLTP システムとの連携中に障害が発生したときの対処方法

ここでは，OLTP システムとの連携中に障害が発生したときの対処方法について説明します。説明する項目は次のとおりです。

- OLTP システムとの連携中に通信障害などが発生したときの対処方法
- 障害によってトランザクションが FORGETTING 状態になったときの対処方法
- フロントエンドサーバがあるユニットが稼働していないためトランザクションの居残りが発生したときの対処方法

なお，ここでの説明は X/Open XA インタフェースを使用している場合に適用されます。

20.11.1　OLTP システムとの連携中に通信障害などが発生したときの対処方法

OLTP システムとの連携中に通信障害などが発生すると，トランザクションブランチがコミット又はロールバックできなくなります。ここでは，このときの HiRDB の処理と HiRDB 管理者の処置について説明します。

(1)　HiRDB の処理

トランザクションブランチの状態を保持し，処理を続行します。

(2)　HiRDB 管理者の処置

1. OLTP システムが起動していない可能性があります。OLTP システムが起動しているかどうかを確認してください。起動していない場合は，OLTP システムを起動してください。
2. 障害回復ができない場合に備えて，**pdcmt** 又は **pdrbk** コマンドでトランザクション決着の準備をします。詳細については，「障害発生後にセキュア状態となったトランザクションの回復方法」を参照してください。

(3)　障害発生後にセキュア状態となったトランザクションの回復方法

KFPS00992-E メッセージ（トランザクションブランチの回復ができない旨のメッセージ）が出力され，pdls -d trn コマンドでトランザクションの状態表示をしたとき，トランザクション第１状態が READY で，かつトランザクション第２状態が p と表示されるセキュア状態（コミット２相目指示待ち状態）のトランザクションの回復方法について説明します。

(a)　HiRDB/シングルサーバの場合

HiRDB/シングルサーバの場合，セキュア状態となるのは OLTP システムからのコミット２相目指示待ち状態以外にはあり得ません。OLTP システムが何らかの要因で異常終了しているか，又は OLTP システムと HiRDB 間の連絡ができない状態であると考えられます。よって，次に示す対策を実施してください。

1. OLTP システムが異常終了している場合

   OLTP システムを再開始してください。OLTP システムの再開始処理の延長で自動的に同期を取ってトランザクションの回復処理をします。

2. １．以外の場合

   OLTP システムの状態，及び OLTP システムと HiRDB 間のネットワークの状態などを確認してください。OLTP システムと HiRDB 間の連絡ができるようになれば，OLTP システムと同期を取ってトランザクションの回復処理をします。

3. コマンドによる独自決着

   OLTP システムの再起動やネットワーク回復が難しい場合，OLTP システムと同期を取らないで，独自にコマンド決着できます。使用するコマンドは，pdcmt，pdrbk，pdfgt です。実行手順については，「未決着状態のトランザクションがあるときの対処方法」を参照してください。

   なお，この方法でトランザクションを回復した場合，OLTP システム下で処理していた他リソースマネジャやトランザクションブランチとの整合性が取れていない場合がありますので，データの内容を確認してください。

(b)　HiRDB/パラレルサーバの場合

HiRDB/パラレルサーバの場合は，セキュア状態になったサーバによって対処方法が異なります。

●フロントエンドサーバの場合

この場合は，「HiRDB/シングルサーバの場合」の「1．OLTP システムが異常終了している場合」と同様に，OLTP システムからのコミット２相目指示待ち状態以外にはあり得ません。したがって，「HiRDB/シングルサーバの場合」の「1．OLTP システムが異常終了している場合」と同様の対処をしてください。

●バックエンドサーバ，ディクショナリサーバの場合

トランザクションの決着方法については，「未決着状態のトランザクションがあるときの対処方法」を参照してください。

なお，**pd_trn_rerun_branch_auto_decide = Y**（省略値は Y）を指定すると，未決着状態のトランザクションを自動決着できます。

20.11.2　障害によってトランザクションが FORGETTING 状態になったときの対処方法

(1)　FORGETTING 状態のトランザクションとは

ここでいう FORGETTING 状態のトランザクションとは，次に示す条件をすべて満たすトランザクションのことです。

- pdls -d trn コマンドの実行結果の STATUS で，トランザクション第 1 状態が FORGETTING（トランザクション終了処理中）である

- pdls -d trn コマンドの実行結果の STATUS で，トランザクション第 3 状態が w（トランザクションマネジャと HiRDB 間でトランザクションの同期合わせ中）である

例えば，次のようなトランザクションのことです。

```
pdls -d trn

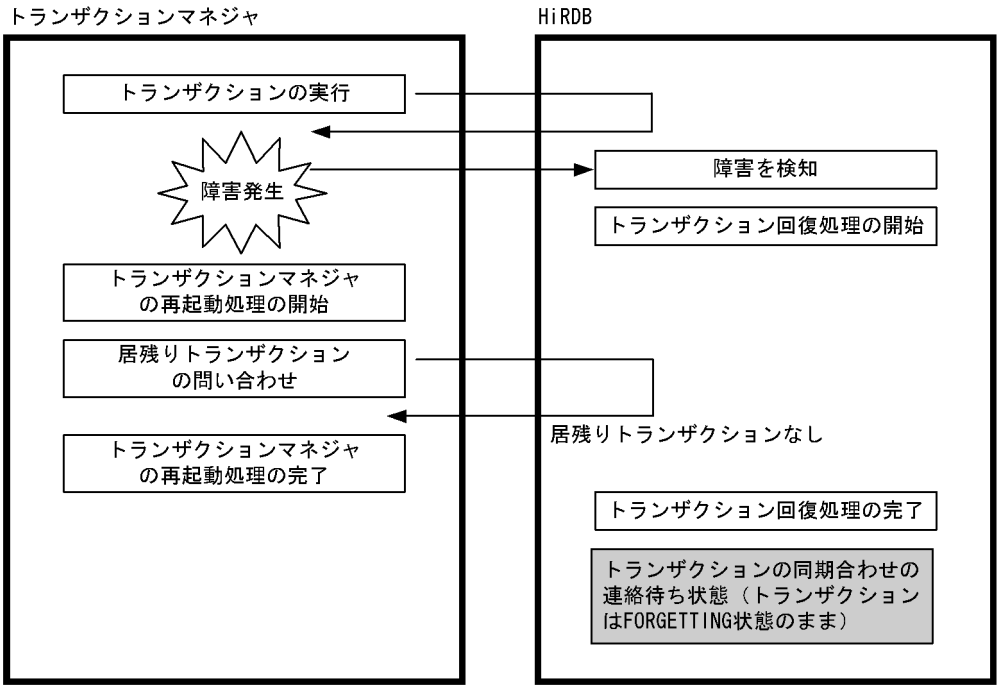
HOSTNAME : host1(153201)
TRNGID           TRNBID           STATUS           PID   SVID   B-SVID
HRD1unt100020b51 HRD1unt100020038 FORGETTING(r,w)    0   bes1   fes1
```

(2) FORGETTING 状態のトランザクションが発生すると

次に示すどれかの条件を満たす障害が発生すると，FORGETTING 状態のトランザクションが HiRDB で発生することがあります。

- トランザクションマネジャが異常終了中である
- トランザクションマネジャのトランザクション開始後，最初に HiRDB にアクセスする SQL（FirstSQL）がエラーになる
- トランザクションマネジャが異常終了して，HiRDB のトランザクション回復完了前にトランザクションマネジャの再起動が完了する（FORGETTING 状態のトランザクションが発生する例を次の図に示します）

図 20–3　トランザクションマネジャの再起動完了後に FORGETTING 状態のトランザクションが発生する例



FORGETTING 状態のトランザクションは HiRDB 内のトランザクションに関するメモリ資源を消費するため，FORGETTING 状態のトランザクションが多数発生すると，接続ユーザ数分のトランザクションの同時実行ができなくなるなどの障害が発生します。

(3) 対処方法

FORGETTING 状態のトランザクションが発生した場合は，次に示すどちらかの方法で対処してください。

- トランザクションマネジャを再起動します。
- pdfgt コマンドを実行して FORGETTING 状態のトランザクションを強制終了します。

20.11.3 フロントエンドサーバがあるユニットが稼働していないためトランザクションの居残りが発生したときの対処方法

(1) トランザクションの居残りが発生すると

次に示す条件がすべて重なった場合，稼働中のユニットにトランザクションが居残ることがあります。

- HiRDB/パラレルサーバが複数のユニットで構成されている
- フロントエンドサーバがあるユニットが非稼働中である
- 稼働中のユニットに未決着状態のトランザクションがある
- トランザクションマネジャが異常終了した後にトランザクションマネジャを開始する

居残りトランザクションはリソースを確保し続けるため，多数のトランザクションが居残ると，接続ユーザ数分のトランザクションが同時実行できなくなることがあります。

(2) 対処方法

pdstart -u コマンドなどでフロントエンドサーバがあるユニットを開始してください。フロントエンドサーバがあるユニットを開始できない場合は，トランザクションマネジャのトランザクションの状態を確認して，pdcmt，pdrbk，又は pdfgt コマンドで HiRDB のトランザクションを強制決着してください。ただし，この方法でトランザクションを強制決着した場合，トランザクションマネジャと HiRDB 間でトランザクションの決着方法が不一致になることがあります。

20.12　通信障害，CPU 障害，又は電源障害が発生したときの対処方法

ここでは，次に示す障害が発生したときの対処方法について説明します。

- 通信障害
- CPU 障害
- 電源障害

20.12.1　通信障害が発生したときの対処方法

HiRDB の処理

該当するトランザクションに通信障害が発生した旨のエラーリターンをします。

HiRDB 管理者の処置

通信障害の原因を調べてください。その後，HiRDB を終了して原因を対策してください。

ただし，通信障害によって HiRDB を正常終了できない場合があります。この場合，HiRDB を強制終了してください。それでも終了できないときは，ユニット単位に終了，開始できればユニット単位に回復してください。ユニット単位に回復できない場合は，システム全体を停止した後，障害要因を取り除いてシステムを再開始してください。

20.12.2　CPU 障害が発生したときの対処方法

HiRDB の処理

CPU 障害が発生すると，CPU 障害が発生したサーバマシンのユニットが異常終了します。

HiRDB 管理者の処置

OS がリブートされた後，HiRDB を再開始してください。ただし，pd_mode_conf オペランドの指定によっては，OS のリブート後，HiRDB はユニットを自動的に再開始します。

20.12.3　電源障害が発生したときの対処方法

HiRDB の処理

電源障害を修復した後に，Windows を再起動し HiRDB を再開始すると，システムを障害発生時点に回復します。

HiRDB 管理者の処置

OS がリブートされた後に HiRDB を再開始してください。ただし，pd_mode_conf オペランドの指定によっては，OS のリブート後，HiRDB はユニットを自動的に再開始します。

注意事項

コマンド実行時又は HiRDB 開始時に，「KFPO00107-E "open(logblib.c)" failed errno=999 : Unknown error」のメッセージが出力された場合，HiRDB の作業ファイルが破壊されている可能性があります。この場合，HiRDB が停止しているのを確認したうえで，次に示すファイルをすべて削除して回復してください。ただし，4 以降のファイルはないことがあります。

1. %PDDIR%¥uxpldir¥spool¥system¥filmng.dat
2. %PDDIR%¥uxpldir¥spool¥system¥flg.dat
3. %PDDIR%¥uxpldir¥spool¥system¥shmmng.dat
4. %PDDIR%¥spool¥～pdatmode
5. %PDDIR%¥spool¥～pdipcid
6. %PDDIR%¥spool¥oslmqid
7. %PDDIR%¥spool¥oslsmid
8. %PDDIR%¥spool¥pdprcsts
9. %PDDIR%¥spool¥scdqid1
10. %PDDIR%¥spool¥scdqid2
11. %PDDIR%¥tmp¥pdommenv
12. %PDDIR%¥uxpldir¥spool¥shm 下の全ファイル
13. %PDDIR%¥uxpldir¥spool¥system¥semmng.dat
14. %PDDIR%¥uxpldir¥spool¥system¥msgmng.dat

これらのファイルを削除した後に HiRDB を開始する場合，エクスプローラなどの他アプリケーションで%PDDIR%¥tmp にアクセスしたままの状態で，HiRDB を開始しないでください。

20.13　接続中のユーザが存在するため HiRDB が終了できないときの対処方法

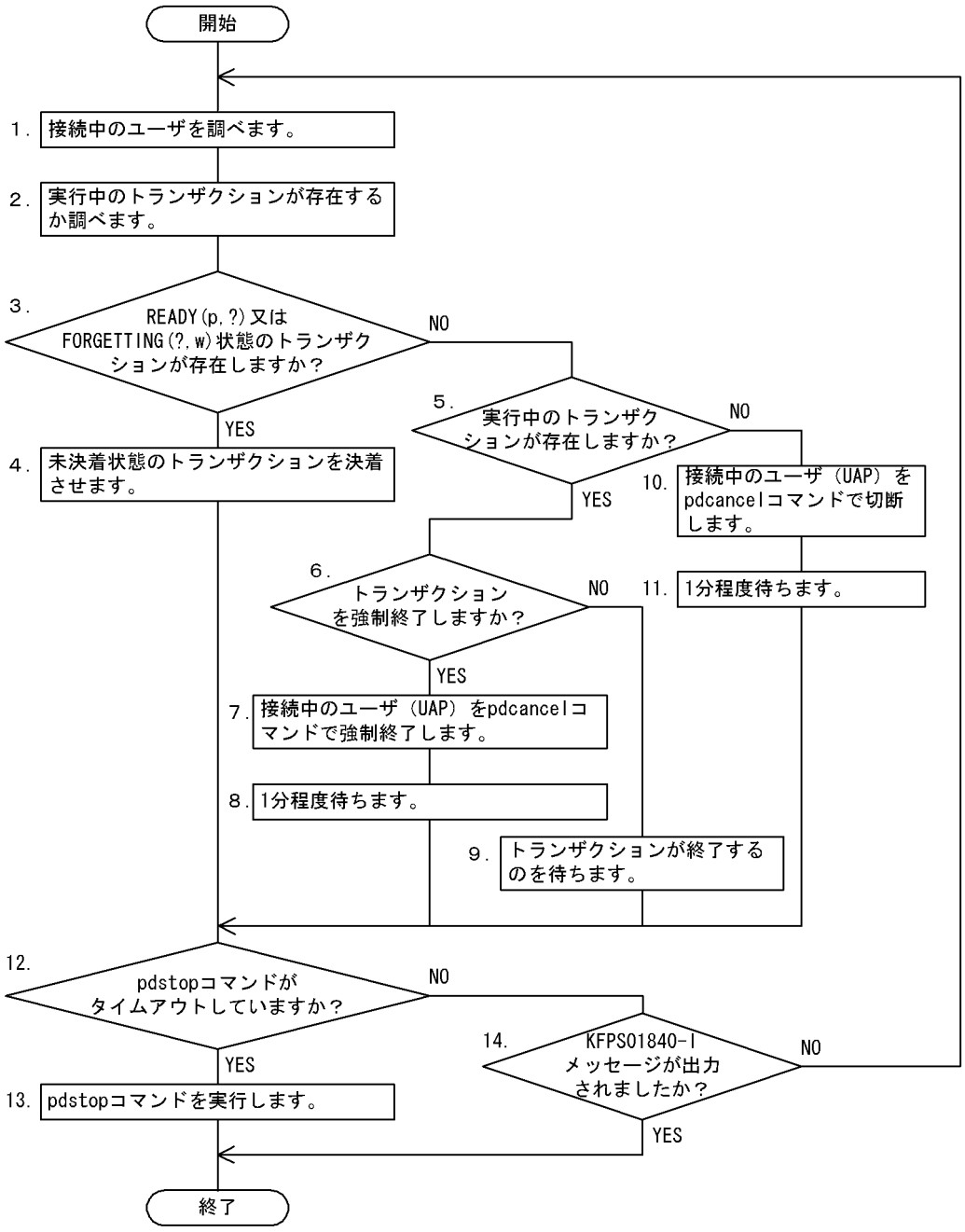
実行者　HiRDB 管理者

ここでは，接続中のユーザが存在するため HiRDB が終了できないときの対処方法について説明します。

20.13.1　対処手順

UAP 又はユティリティが終了しないと（接続中のユーザが存在すると）HiRDB を終了できません。HiRDB の終了時に接続中のユーザが存在する場合は，**KFPS05120-W** メッセージを出力し，**接続ユーザ情報ファイル**及び**接続ユーザ詳細ファイル**を出力します。このとき，HiRDB 管理者は次に示す手順に従って接続中のユーザを切断し，HiRDB を終了してください。

〈手順〉



注

処理ボックスの左にある数字はこの後で説明している（　）レベルに対応しています。例えば，5 の操作は(5)で説明しています。

(1)　接続中のユーザを調べます

(a)　pdls -d prc コマンドで接続中のユーザを調べる場合

```
pdls -d prc

HOSTNAME : k95x620(173420)
```

```
STATUS   PID   UID   GID   SVID   TIME      PROGRAM    C-PID   C-GRP
  L      22118 334   300   fes1   173330    uap00      22205   PC
```

〔説明〕

PROGRAM に表示されている UAP 識別子を確認します。接続中の UAP は **uap00** です。

(b)　接続ユーザ情報ファイルで接続中のユーザを調べる場合

接続先ユーザ情報ファイルの出力内容については，「接続ユーザ情報ファイルと接続ユーザ詳細ファイル」を参照してください。

```
UNIT ID : UNT1(173420)
 UID  PID    GID  SVID  TIME    PROGRAM  C-GRP  C-PID   C-IP
 334  22118  300  fes1  173330  uap00    PC     22205   172.17.32.37
```

〔説明〕

PROGRAM に表示されている UAP 識別子を確認します。接続中の UAP は **uap00** です。

(2)　実行中のトランザクションが存在するか調べます

pdls -d trn コマンドを実行するか，又は接続ユーザ詳細ファイル中の pdls -d trn コマンドの実行結果から実行中のトランザクションが存在するかを調べます。

接続ユーザ詳細ファイルの出力内容については，「接続ユーザ情報ファイルと接続ユーザ詳細ファイル」を参照してください。

(3)　READY(p，?)又は FORGETTING(?，w)状態のトランザクションが存在しますか？

pdls -d trn コマンドの実行結果，又は接続ユーザ詳細ファイルの情報から READY(p，?)又は FORGETTING(?，w)状態のトランザクションが存在するかを調べてください。

(4)　未決着状態のトランザクションを決着させます

「未決着状態のトランザクションがあるときの対処方法」を参照して未決着状態のトランザクションを決着してください。

(5)　実行中のトランザクションが存在しますか？

pdls -d trn コマンドの実行結果，又は接続ユーザ詳細ファイルの情報から READY(p，?)又は FORGETTING(?，w)状態以外のトランザクションが存在するかを調べてください。

(6)　トランザクションを強制終了しますか？

READY(p，?)又は FORGETTING(?，w)状態以外のトランザクションが存在した場合，そのトランザクションを強制終了するか，トランザクションが決着するのを待つかを決定します。実行中のトランザクションの処理時間が短い場合はトランザクションの終了を待ってください。処理時間が長い場合はトランザクションを強制終了してください。

(7)　接続中のユーザ（UAP）を pdcancel コマンドで強制終了します

pdcancel コマンドでトランザクションを強制終了します。

HiRDB/シングルサーバの場合

```
pdcancel -u UAP00 -i 22118
```

HiRDB/パラレルサーバの場合（実行例 1）

```
pdcancel -x k95x620 -u UAP00 -i 22118
```

HiRDB/パラレルサーバの場合（実行例 2）

```
pdcancel -X UNT1 -u UAP00 -i 22118
```

〔説明〕

-u：UAP 識別子を指定します。

-x：UAP が接続しているフロントエンドサーバがあるホスト名を指定します。指定するホスト名は pdls -d prc コマンドの実行結果の HOSTNAME に表示されます。

-X：ユニット識別子を指定します。指定するユニット識別子は接続ユーザ情報ファイルの UNIT ID に表示されます。

-i：プロセス ID を指定します。

(8)　1 分程度待ちます

pdcancel コマンドの実行後，トランザクションが自動的に決着するのを待ちます。

(9)　トランザクションが終了するのを待ちます

トランザクションを強制終了しない場合は，トランザクションが終了するのを待ちます。

(10)　接続中のユーザ（UAP）を pdcancel コマンドで切断します

接続中のユーザにトランザクションが存在しない場合は，接続中のユーザを pdcancel コマンドで切断してください。

HiRDB/シングルサーバの場合

```
pdcancel -i 22118 -d
```

HiRDB/パラレルサーバの場合（実行例 1）

```
pdcancel -x k95x620 -i 22118 -d
```

HiRDB/パラレルサーバの場合（実行例 2）

```
pdcancel -X UNT1 -i 22118 -d
```

〔説明〕

-i：プロセス ID を指定します。

-d：HiRDB 側のサーバプロセスを強制終了する場合に指定します。

-x：UAP が接続しているフロントエンドサーバがあるホスト名を指定します。指定するホスト名は pdls -d prc コマンドの実行結果の HOSTNAME に表示されます。

-X：ユニット識別子を指定します。指定するユニット識別子は接続ユーザ情報ファイルの UNIT ID に表示されます。

(11) 1 分程度待ちます

pdcancel コマンドの実行後はユーザの接続が切断されて，HiRDB の終了処理が再開されるのを待ちます。

(12) pdstop コマンドがタイムアウトしていますか？

KFPS05047-E メッセージが出力されていると，pdstop コマンドがタイムアウトしています。

(13) pdstop コマンドを実行します

pdstop コマンドがタイムアウトした場合は，再度 pdstop コマンドを実行してください。

(14) KFPS01840-I メッセージが出力されましたか？

KFPS01840-I メッセージが出力されている場合，HiRDB の終了処理が開始されています。

20.13.2 接続ユーザ情報ファイルと接続ユーザ詳細ファイル

接続中のユーザが存在する状態で HiRDB を終了しようとした場合，**KFPS05120-W** メッセージを出力して接続ユーザ情報ファイル（%PDDIR%¥spool¥cnctusrinf），及び接続ユーザ詳細ファイル（%PDDIR%¥spool¥cnctusrdtl）を出力します。HiRDB/パラレルサーバの場合，これらのファイルはシステムマネジャ

があるサーバマシンに出力されます。HiRDB 管理者はこれらのファイルの情報から接続中のユーザを特定してください。

(1) 接続ユーザ情報ファイルの出力情報

出力例

```
UNIT ID : M350(173420)
UID    PID         GID    SVID      TIME    PROGRAM     C-GRP  C-PID       C-IP
  334       22118    300  fes1      173330  uap00        PC         22205  172.17.32.37

UNIT ID : M35b(173427)
UID    PID         GID    SVID      TIME    PROGRAM     C-GRP  C-PID       C-IP
  334       17524    300  fes2      173343  uap01        PC         17619  172.17.32.39
  334       17533    300  fes2      173333  uap02        PC         22200  172.17.32.37
```

〔説明〕

UNIT ID：

ユニット識別子（4 バイト）と，接続ユーザ情報ファイルの作成時刻（6 バイト，時分秒形式）に括弧を付けて表示します。

UID：

HiRDB に接続中のユーザのユーザ ID を表示します（5 バイト，右詰め表示）。

PID：

HiRDB のサーバプロセスのプロセス ID を表示します（10 バイト，右詰め表示）。

GID：

HiRDB に接続中のユーザのグループ ID を表示します（5 バイト，右詰め表示）。

SVID：

ユーザが接続しているサーバのサーバ名を表示します（8 バイト，左詰め表示）。

HiRDB のサーバプロセス起動直後には，空白が表示される場合があります。

TIME：

HiRDB がサービスの要求を受け付けた時刻（時分秒）を表示します（6 バイト）。ユーザが接続していないサーバについては 999999 を表示します。

PROGRAM：

クライアント環境定義の PDCLTAPNAME オペランドの指定値が表示されます（30 バイト，左詰め）。PDCLTAPNAME オペランドを省略した場合は，「Unknown」を表示します。PDCLTAPNAME オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

- ユティリティの場合は「********」を表示します。
- HiRDB のサーバプロセス起動直後には，「********」が表示される場合があります。

C-GRP：

HiRDB に接続中のユーザの種別（クライアントグループの種別）を表示します（2 バイト）。

なお，クライアントグループの接続枠保証機能を使用していて，ユーザ任意のクライアントグループを定義している場合は，ユーザが定義したクライアントグループ名称を表示します。

表示される文字列	ユーザの種別
XA	X/Open XA インタフェースで HiRDB に接続したユーザです。
PC	PC クライアントから HiRDB に接続したユーザです。
WS	WS クライアントから HiRDB に接続したユーザです。

C-PID：

HiRDB に接続中のクライアントのプロセス ID を表示します（10 バイト，右詰め）。

次に示す場合は表示されません。

- UAP をリンケージしたクライアントのバージョンが HiRDB Version 4.0 04-00 より前の場合
- Type4 JDBC ドライバから接続している場合

C-IP：

HiRDB に接続中のクライアントの IP アドレスを表示します（15 バイト，左詰め）。

(2) 接続ユーザ詳細ファイルの出力情報

接続ユーザ詳細ファイルには次に示すコマンドの実行結果が出力されます。

- pdls -d act
- pdls -d prc
- pdls -d trn

各コマンドの実行結果については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

20.14　未決着状態のトランザクションがあるときの対処方法

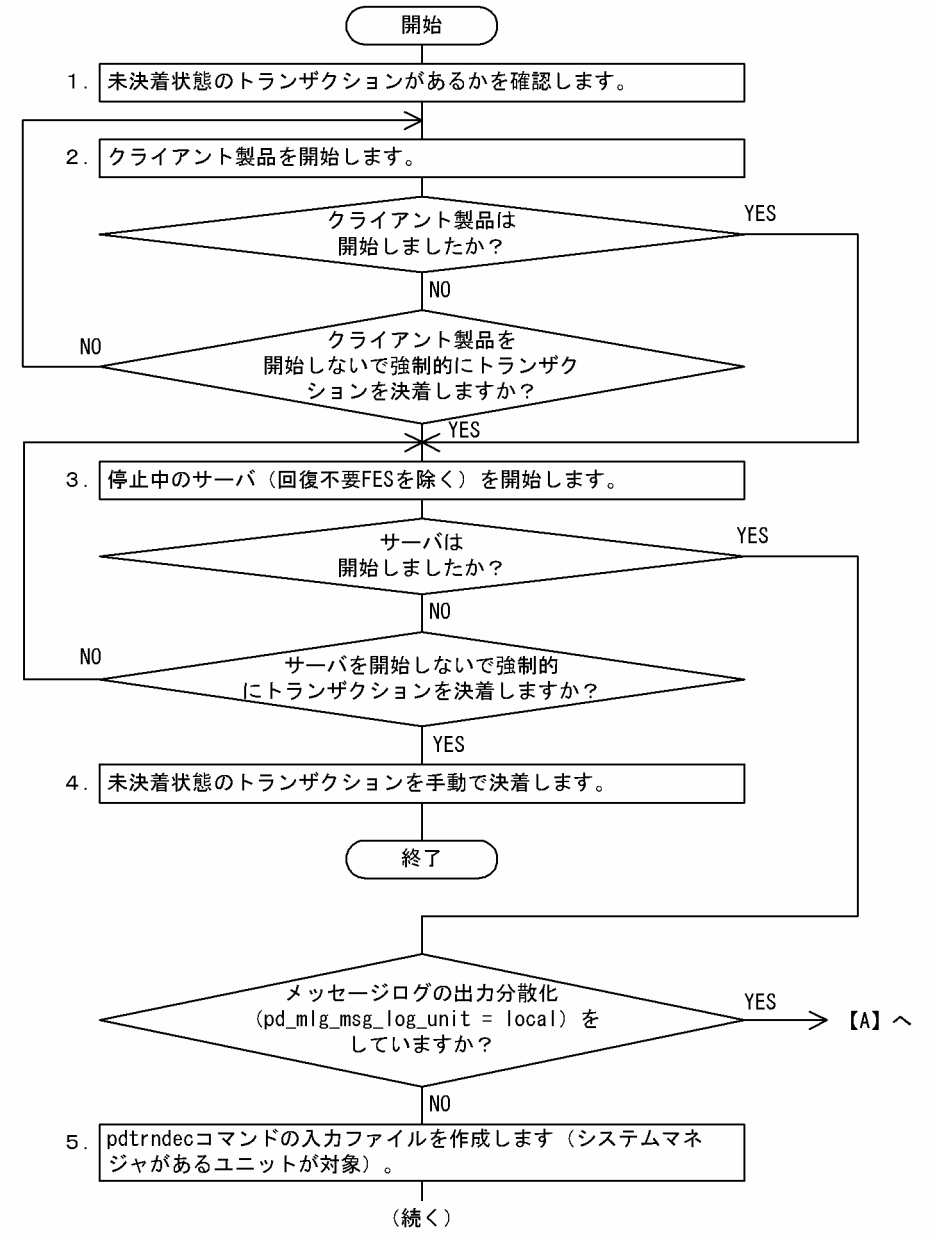
ここでは，未決着状態のトランザクションがあるときの対処方法について説明します。

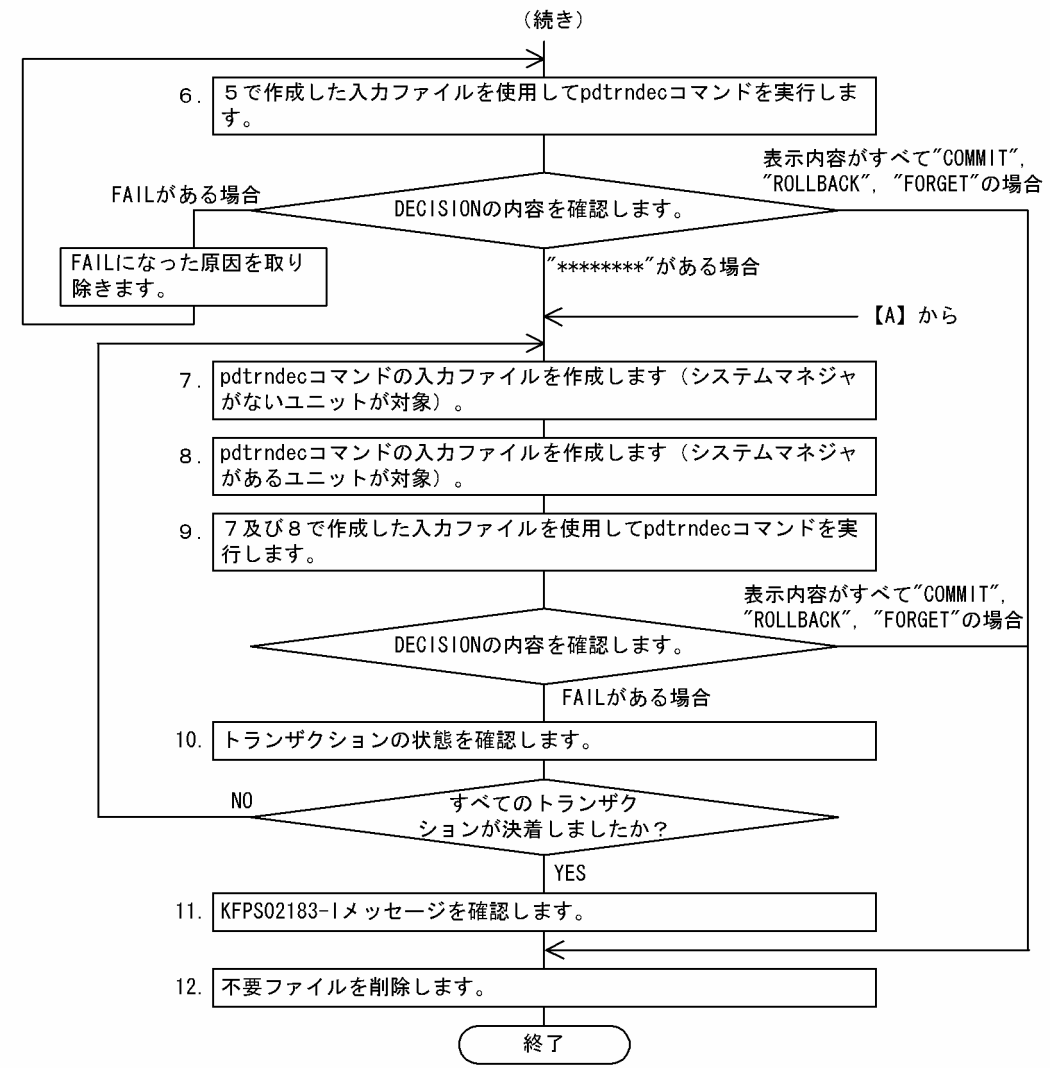
20.14.1　未決着状態のトランザクションを決着する方法

HiRDB の再開始時又は正常終了時に未決着状態のトランザクション（決着できないトランザクションブランチ）がある場合，**KFPS00992-E** メッセージが出力されます。このとき，HiRDB 管理者は次に示す手順に従って未決着状態のトランザクションを決着してください。

pd_trn_rerun_branch_auto_decide = Y（省略値）を指定すると，未決着状態のトランザクションを自動決着できます。この自動決着機能では決着できない場合や，自動決着機能を使用しない場合に，次の例題で説明する操作が必要になります。

〈手順〉


開始
1. 未決着状態のトランザクションがあるかを確認します。
2. クライアント製品を開始します。
クライアント製品は
開始しましたか？
YES
NO
NO
クライアント製品を
開始しないで強制的にトランザク
ションを決着しますか？
YES
3. 停止中のサーバ（回復不要FESを除く）を開始します。
サーバは
開始しましたか？
YES
NO
NO
サーバを開始しないで強制的
にトランザクションを決着しますか？
YES
4. 未決着状態のトランザクションを手動で決着します。
終了
メッセージログの出力分散化
（pd_mlg_msg_log_unit = local）を
していますか？
YES
【A】へ
NO
5. pdtrndecコマンドの入力ファイルを作成します（システムマネ
ジャがあるユニットが対象）。
（続く）



注

- 処理ボックスの左にある数字はこの後で説明している（　）レベルに対応しています。例えば，5の操作は(5)で説明しています。
- 以降で説明する手順の中で，メッセージを調べる操作があります。このとき，イベントログ中のメッセージを参照するようにしてください。

(1)　未決着状態のトランザクションがあるかを確認します

KFPS00992-E メッセージを検索して，未決着状態のトランザクションがあるか確認してください。

```
KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b4d, TRNBID=HRD1unt100020034, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b22, TRNBID=HRD1unt100020035, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b5a, TRNBID=HRD1unt100020036, server=bes1, service=p_f_sqa_call
```

〔説明〕

KFPS00992-E メッセージには，コミット又はロールバックできなかったトランザクションブランチが表示されます。この場合，次に示すトランザクション識別子のトランザクションが未決着状態である可能性があります。

- TRNGID=HRD1unt100020b4d
- TRNGID=HRD1unt100020b22
- TRNGID=HRD1unt100020b5a

未決着状態のトランザクションがない場合はこれ以降の操作は必要ありません。

(2)　クライアント製品を開始します

HiRDB のクライアント製品（OLTP システム，HiRDB Datareplicator など）が正常に稼働しているかを確認してください。稼働していない場合は各製品で決められた手順に従いクライアント製品を開始してください。

なお，停止中のクライアント製品がある状態で，強制的にトランザクションを決着する場合は次に示す点に注意してください。

- 現在停止中のクライアント製品からの要求で発生した未決着状態のトランザクションは，以降の操作をすると強制的にロールバックされることがあります。ただし，この後にクライアント製品を開始した場合，HiRDB とクライアント製品との間で，このトランザクションの同期が取れないことがあります。

また，トランザクションの整合性を保つために，これ以降新たなトランザクションを実行しないようにしてください。

(3)　停止中のサーバを開始します

全サーバ（回復不要 FES を除くすべてのサーバ）が稼働中であることを pdls コマンドで確認してください。稼働していないサーバがある場合は，そのサーバを開始してください。

なお，停止中のサーバがある状態で，強制的にトランザクションを決着する場合は次に示す点に注意してください。

- 停止中のサーバと関連があるトランザクションについては，以降の操作をすると強制的にロールバックされることがあります。ただし，この後に停止中のサーバを開始した場合，このトランザクションの同期が取れないことがあります。
- 停止中の回復不要 FES を開始する必要はありません。

(4)　未決着状態のトランザクションを手動で決着します

未決着状態のトランザクションを手動で決着してください。手動での決着方法については，「未決着状態のトランザクションを手動で決着する方法」を参照してください。

(5) pdtrndec コマンドの入力ファイルを作成します（システムマネジャがあるユニットが対象）

次に示す方法で pdtrndec コマンドの入力ファイルを作成してください。

イベントビューアからアプリケーションログを開いて，［ログファイルの名前を付けて保存］を選択して保存します。このとき，“アプリケーションログ”の名前を付けて保存というウィンドウが表示されます。保存する場所に任意の作業用ディレクトリを選択，ファイル名に任意のファイル名を指定，ファイルの種類はテキストを選択して保存してください。なお，出力先ファイルのパス名には空白を入れないでください。

システムが正常開始した時刻（KFPS01803-I start mode=S メッセージを出力して，かつ KFPS05210-I メッセージ出力した時刻）以降のすべてのイベントログを対象にしてください。また，イベントログが満杯又は指定期間を過ぎたため，別ファイルに保存している場合は，ファイル種別がテキストファイル以外のときはそのファイルをインポートしてアプリケーションログを開き，ファイル種別にテキストを選択して保存してください。なお，システムが正常開始した時刻を含むイベントログがない場合は，今あるすべてのイベントログをテキストファイルとして保存してください。

(6) (5)で作成した入力ファイルを使用して pdtrndec コマンドを実行します

```
pdtrndec -i C:¥tmp¥unit1syslog_copy,C:¥tmp¥unit1syslog2_copy
```

〔説明〕

-i：(5)で作成した入力ファイル

コマンドの実行結果の DECISION（トランザクションの決着種別）に表示されている内容を確認してください。確認後，**〈手順〉**のフローチャートに従い操作を続行してください。

```
pdtrndec result                    START TIME:2002/09/05 17:20:08
HOSTNAME    TRNGID           TRNBID           SVID      DECISION    TIME
host1       HRD1unt100020b4d HRD1unt100020034 bes1      COMMIT      17:20:08
host1       HRD1unt100020b22 HRD1unt100020035 bes1      ********    --:--:--
host1       HRD1unt100020b51 HRD1unt100020038 bes1      FORGET      17:20:09
pdtrndec result                    START TIME:2002/09/05 17:36:47
HOSTNAME    TRNGID           TRNBID           SVID      DECISION    TIME
host1       HRD1unt100020b22 HRD1unt100020035 bes1      ********    --:--:--
```

注意事項

pdtrndec コマンドを実行した後に未決着状態のトランザクションがある場合，pdtrnrbk.bat ファイルが作成されますが，この時点では pdtrnrbk.bat を実行しないでください。実行した場合，トランザクションの同期が取れなくなることがあります。また，KFPS00982-E メッセージが出力されていて，かつ pdtrnrbk.bat ファイルが作成されている場合は，その pdtrnrbk.bat ファイルは使用しないで削除してください。

(7) pdtrndec コマンドの入力ファイルを作成します（システムマネジャがないユニットが対象）

次に示す方法で pdtrndec コマンドの入力ファイルを作成してください。

イベントビューアからアプリケーションログを開いて，［ログファイルの名前を付けて保存］を選択して保存します。このとき，“アプリケーションログ”の名前を付けて保存というウィンドウが表示されます。保存する場所に任意の作業用ディレクトリを選択，ファイル名に任意のファイル名を指定，ファイルの種類はテキストを選択して保存してください。なお，出力先ファイルのパス名には空白を入れないでください。

システムが正常開始した時刻（KFPS01803-I start mode=S メッセージを出力して，かつ KFPS05210-I メッセージ出力した時刻）以降のすべてのイベントログを対象にしてください。また，イベントログが満杯又は指定期間を過ぎたため，別ファイルに保存している場合は，ファイル種別がテキストファイル以外のときはそのファイルをインポートしてアプリケーションログを開き，ファイル種別にテキストを選択して保存してください。なお，システムが正常開始した時刻を含むイベントログがない場合は，今あるすべてのイベントログをテキストファイルとして保存してください。

(8) pdtrndec コマンドの入力ファイルを作成します（システムマネジャがあるユニットが対象）

(5)と同じ方法で pdtrndec コマンドの入力ファイルを作成してください。

(9) (7)及び(8)で作成した入力ファイルを使用して pdtrndec コマンドを実行します

(7)及び(8)で作成した入力ファイルを使用して pdtrndec コマンドでトランザクションを決着させてください。1 回目の pdtrndec コマンドの実行後，最終ステータスのリターンコードが 4 で，DECISION の表示内容に”********”がある場合は，次に示すオプションを指定して再度 pdtrndec コマンドを実行してください。

```
pdtrndec -i C:¥tmp¥unit1syslog_copy,C:¥tmp¥unit1syslog2_copy,C:¥tmp¥unit2syslog_copy,
C:¥tmp¥unit2syslog2_copy -r pdtrnrbk.bat
```

〔説明〕

-i：(7)及び(8)で作成した入力ファイルを指定します。

-r：(6)で生成されたバッチファイル（pdtrnrbk.bat）のファイルを指定します。

KFPS00982-E メッセージが出力されていて，かつ pdtrnrbk.bat ファイルが作成されている場合は，その pdtrnrbk.bat ファイルは使用しないで削除してください。

(10)　トランザクションの状態を確認します

トランザクションが決着したかどうかを pdls -d trn コマンドで確認してください。未決着状態のトランザクションがまだある場合は，出力されたメッセージやコマンドの実行結果を参照してエラーの原因を対策し，手順(7)からの操作を繰り返してください。

```
pdls -d trn

HOSTNAME : host1(153415)
TRNGID          TRNBID          STATUS        PID   SVID   B-SVID
```

〔説明〕

トランザクションの情報が表示されていないため，すべてのトランザクションが決着しています。

(11)　KFPS02183-I メッセージを確認します

未決着状態のトランザクションがすべて決着すると，30 秒以内に KFPS02183-I メッセージが出力されます。30 秒を経過しても KFPS02183-I メッセージが出力されない場合は，まだ未決着状態のトランザクションがあります。この場合は最初から操作を繰り返してください。

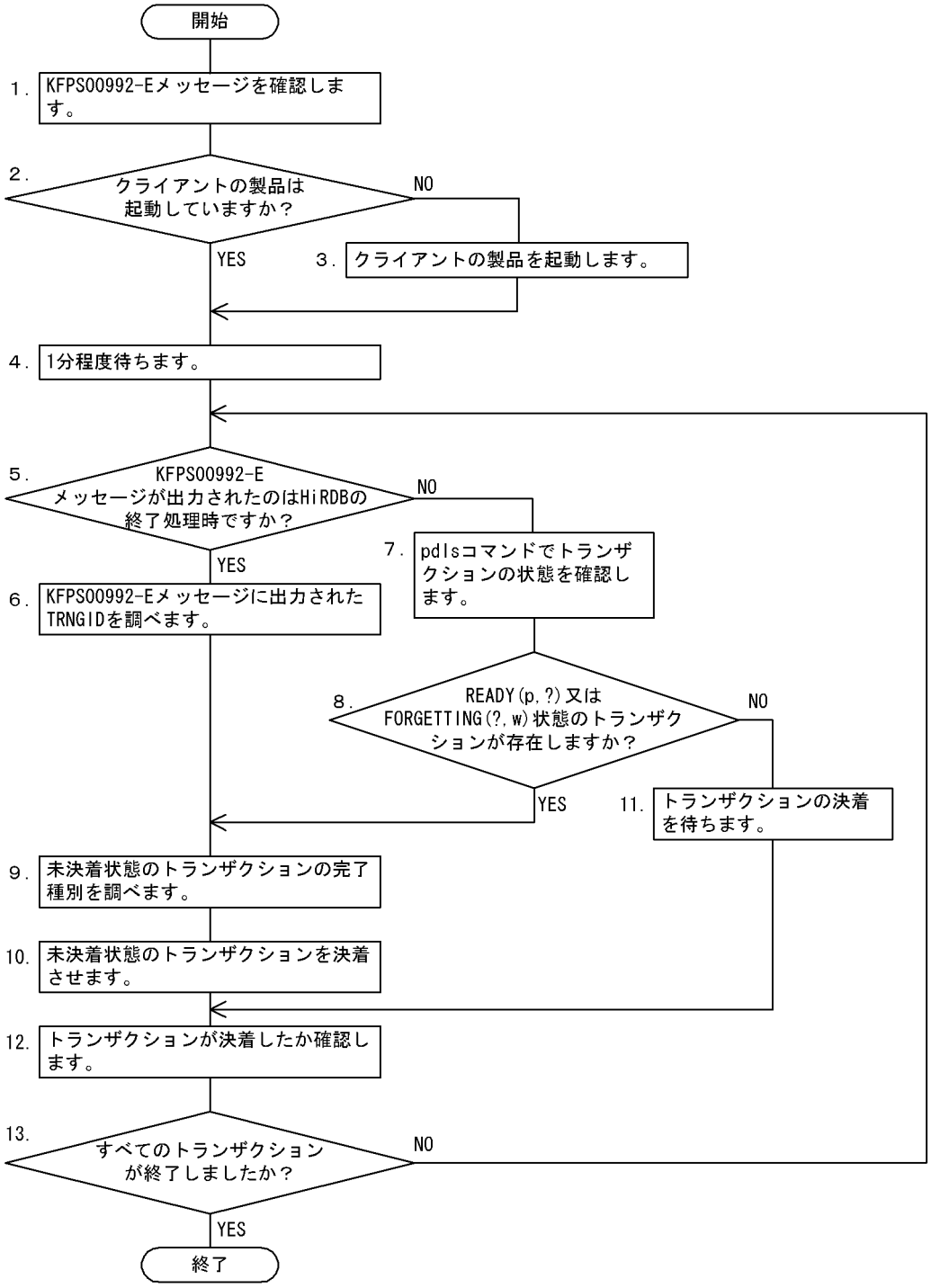
(12)　不要ファイルを削除します

pdtrndec コマンドで指定した出力先ディレクトリ下のファイル（pdtrndecout，pdtrnrbk.bat など）を削除してください。

20.14.2　未決着状態のトランザクションを手動で決着する方法

未決着状態のトランザクションを手動で決着する手順を次に示します。

〈手順〉



注

- 処理ボックスの左にある数字はこの後で説明している（　）レベルに対応しています。例えば，5の操作は(5)で説明しています。
- 以降で説明する手順の中で，メッセージを調べる操作があります。このとき，イベントログ中のメッセージを参照するようにしてください。

(1) KFPS00992-E メッセージを確認します

イベントログ中の KFPS00992-E メッセージをサーチしてください。

```
KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b4d, TRNBID=HRD1unt100020034, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b22, TRNBID=HRD1unt100020035, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b5a, TRNBID=HRD1unt100020036, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b5f, TRNBID=HRD1unt100020037, server=bes1, service=p_f_sqa_call

KFPS00992-E Unable to determine commit or rollback for transaction branch.
TRNGID=HRD1unt100020b64, TRNBID=HRD1unt100020038, server=bes1, service=p_f_sqa_call
```

〔説明〕

KFPS00992-E メッセージには，コミット又はロールバックできなかったトランザクションブランチが表示されます。この場合，次に示すトランザクション識別子のトランザクションが未決着状態である可能性があります。

- TRNGID=HRD1unt100020b4d
- TRNGID=HRD1unt100020b22
- TRNGID=HRD1unt100020b5a
- TRNGID=HRD1unt100020b5f
- TRNGID=HRD1unt100020b64

(2) クライアントの製品は起動していますか？

HiRDB のクライアントとして使用している製品が，正常に稼働しているかどうかを確認してください。

(3) クライアントの製品を起動します

HiRDB のクライアントとして使用している製品が，正常に稼働していない場合は，各製品で決められている手順で起動してください。

このとき，クライアントの製品の強制開始を行わないでください。

(4) 1 分程度待ちます

トランザクションが自動的に決着する可能性があるため，1 分程度待ちます。

(5) KFPS00992-E メッセージが出力されたのは HiRDB の終了処理時ですか？

KFPS00992-E メッセージが HiRDB 終了処理時に出力された場合と，それ以外の場合では，以降の対策方法が異なります。

(6) KFPS00992-E メッセージに出力された TRNGID を調べます

HiRDB の終了処理時に KFPS00992-E メッセージが出力された場合は，pdls -d trn コマンドが実行できないため，KFPS00992-E メッセージに出力された TRNGID を調べます。

(7) pdls コマンドでトランザクションの状態を確認します

pdls -d trn コマンドで，トランザクションの状態を確認してください。

```
pdls -d trn

HOSTNAME : host1(153415)
TRNGID           TRNBID           STATUS             PID    SVID    B-SVID
HRD1unt100020b4d HRD1unt100020034  READY(p,n)        0      bes1    fes1
HRD1unt100020b22 HRD1unt100020035  READY(p,n)        0      bes1    fes1
HRD1unt100020b5a HRD1unt100020036  READY(p,n)        0      bes1    fes1
HRD1unt100020b5f HRD1unt100020037  ROLLBACK(u,n)     29799  bes1    fes1
HRD1unt100020b69 HRD1unt100020039  FORGETTING(r,w)   0      bes1    fes1
```

〔説明〕

STATUS 欄に表示される情報を見てください。

- 次に示す三つのトランザクションが READY(p,?)状態になっています。

  TRNGID=HRD1unt100020b4d

  TRNGID=HRD1unt100020b22

  TRNGID=HRD1unt100020b5a

- TRNGID=HRD1unt100020b5f のトランザクションは，クライアントの製品を起動したため，決着処理が行われて ROLLBACK 状態になっています。このトランザクションは自動的に決着します。
- TRNGID=HRD1unt100020b64 のトランザクションは，クライアントの製品を起動したため，決着処理が行われてトランザクションが決着しました。そのため，pdls -d trn コマンドの実行結果に表示されません。
- KFPS00992-E メッセージには出力されなかった TRNGID=HRD1unt100020b69 のトランザクションが，FORGETTING(?，w)状態になっています。

(8) READY(p，?)又は FORGETTING(?，w)状態のトランザクションが存在しますか？

pdls -d trn コマンドの実行結果（STATUS の欄）に，READY(p，?)状態又は FORGETTING(?，w)状態のトランザクションがある場合，未決着トランザクションの決着を行います。

READY(p，?)状態又は FORGETTING(?，w)状態以外のトランザクションだけがある場合は，トランザクションが自動的に決着するのを待ちます。

(9)　未決着状態のトランザクションの完了種別を調べます

READY(p，?)状態又は FORGETTING(?，w)状態のトランザクションがある場合は，イベントログ中の KFPS00990-I メッセージをサーチしてください。このとき，決着させる必要があるトランザクションの TRNGID をキーにしてサーチします。KFPS00990-I メッセージにはトランザクションの完了種別が表示されます。なお，サーチする KFPS00990-I メッセージは，前回正常開始時点以降に出力されたメッセージだけを対象にしてください。

```
KFPS00990-I Transaction branch recovery complete. TRNGID=HRD1unt100020b5a,
TRNBID=HRD1unt200020015, server=fes1, service=p_f_sqa_cauxi, completion type=c, c

KFPS00990-I Transaction branch recovery complete. TRNGID=HRD1unt100020b4d,
TRNBID=HRD1unt200020014, server=fes1, service=p_f_sqa_cauxi, completion type=r, r
```

〔説明〕

サーチした結果，ヒットしたトランザクションの処置を次に示します。

- TRNGID=HRD1unt100020b5a のトランザクション完了種別
  completion type=c となっているため，コミットでトランザクションを決着させます。
- TRNGID=HRD1unt100020b4d のトランザクション完了種別
  completion type=r となっているため，ロールバックでトランザクションを決着させます。

サーチした結果，ヒットしなかったトランザクションの処置を次に示します。

- TRNGID=HRD1unt100020b22 のトランザクション完了種別
  同じ TRNGID を持つ KFPS00990-I メッセージがなかったため，ロールバックでトランザクションを決着させます。
- TRNGID=HRD1unt100020b69 のトランザクション完了種別
  STATUS が FORGETTING(?，w)のため，トランザクションを強制終了します。

次に示す TRNGID のトランザクションについては完了種別を調べる必要がありません。

- TRNGID=HRD1unt100020b5f
- TRNGID=HRD1unt100020b64

(10)　未決着状態のトランザクションを決着させます

次に示すコマンドでトランザクションを決着してください。

コマンド名	機能及び適用基準
pdcmt	トランザクションをコミットします。
pdrbk	トランザクションをロールバックします。

コマンド名	機能及び適用基準
pdfgt	トランザクションを強制終了します。FORGETTING(？,w)状態のトランザクションに実行します。

(11)　トランザクションの決着を待ちます

READY(p，?)又は FORGETTING(?，w)状態以外のトランザクションだけがある場合，トランザクションは決着中のため，決着処理が完了するのを待ってください。

(12)　トランザクションが決着したか確認します

KFPS00992-E メッセージが HiRDB の終了処理中に出力された場合は，KFPS02183-I メッセージが出力されたかどうかを確認してください。未決着状態のトランザクションが決着すると，30 秒以内に KFPS02183-E メッセージが出力されます。30 秒以上経過しても KFPS02183-I メッセージが表示されない場合は，未決着状態のトランザクションが存在します。

HiRDB の再開始時に KFPS00992-E メッセージが出力された場合，又は HiRDB の稼働中に pdls -d trn コマンドで未決着状態のトランザクションを発見した場合は，トランザクションが決着しているかどうかを pdls -d trn コマンドで確認してください。

(13)　すべてのトランザクションが終了しましたか？

すべてのトランザクションが終了していない場合は，(5)に戻って操作してください。

20.15　縮退起動をするときの運用方法（HiRDB/パラレルサーバ限定）

実行者　HiRDB 管理者

障害などで起動できないユニットがある場合，縮退起動機能を使用すると，残りの正常なユニットだけで HiRDB を開始できます。

20.15.1　縮退起動をするには

縮退起動をするには **pd_start_level** オペランドに 1 を指定します。この場合，HiRDB 開始時に起動できないユニットがあれば，起動できないユニットを除いて HiRDB を開始します。縮退起動となった場合は，メッセージ **KFPS05217-I** が出力されます。起動できなかったユニットは **pdls** コマンドで調べてください。

備考

HiRDB を開始するとき，HiRDB は各ユニットからの開始処理完了の連絡を最大 20 分待ちます。したがって，縮退起動をするときは最大 20 分間の開始処理完了の連絡待ち時間が発生します。なお，最大 20 分間の開始処理完了の連絡待ち時間を **pd_reduced_check_time** オペランドで変更できます。

20.15.2　正常運転に戻す方法

縮退起動をした後に正常運転に戻す手順を次に示します。

〈手順〉

1. **pdls** コマンドで起動できないユニットを調べて，起動できないユニットの障害を対策してください。
   pdls -d svr
2. **pdstart -u** コマンドで起動していないユニットを開始します。
   pdstart -u unt1
   正常運転になると，**KFPS05218-I** メッセージが出力されます。

20.15.3　縮退起動ができない場合があります

次に示す場合は縮退起動ができません。障害を取り除いてから HiRDB を開始してください。

- 起動するユニットに，次に示すサーバがない場合
  - システムマネジャ
  - ディクショナリサーバ
- 起動するユニットに，次に示すサーバが一つもない場合
  - フロントエンドサーバ

- バックエンドサーバ

- pdstart -i コマンドで HiRDB を開始する場合
- バージョンアップ後の最初の HiRDB 開始時

20.15.4　縮退起動中は表にアクセスできないことがあります

起動していないユニットにある表をアクセスすると SQL エラーになります。表を横分割している場合は，起動していないユニットにあるデータをアクセスすると SQL エラーになります。

20.15.5　再開始時に縮退起動をした場合の注意事項

再開始時に縮退起動をした場合，障害が発生したユニットに関連したトランザクションが未決着状態になることがあります。このトランザクションは縮退起動が解除されて正常運転になると自動的に決着します。また，未決着状態のトランザクションがあると，HiRDB を正常終了又は計画停止できません。HiRDB を終了させるには，**pdstop -f** コマンドで強制終了させてください。

20.15.6　マルチフロントエンドサーバの場合の注意事項

クライアントユーザが接続するフロントエンドサーバを決めている場合は注意が必要です。具体的には，クライアント環境定義で次に示すオペランドを指定していると注意が必要です。

- PDFESHOST
- PDSERVICEGRP

これらのオペランドに指定しているフロントエンドサーバが起動していない場合，UAP が実行できません。UAP を実行するにはこれらのオペランドを削除するか，又は起動しているフロントエンドサーバをこれらのオペランドに指定してください。

20.15.7　HiRDB Datareplicator と連携している場合の注意事項

(1)　HiRDB が抽出側の場合

- 起動していないユニットのバックエンドサーバのデータ抽出処理は実施されません。この場合，正常運転後に HiRDB Datareplicator の **hdestart** コマンドでデータ抽出処理を再開してください。なお，縮退起動の状態で HiRDB を終了し，次回開始時には正常運転になる場合は，hdestart コマンドを入力する必要はありません。自動的にデータ抽出処理が再開されます。

- 縮退起動中にデータ抽出処理を停止する場合は pdrplstop コマンドを実行してください。pdrplstop -f コマンドはなるべく実行しないでください。pdrplstop コマンドを実行すると，正常運転に戻ったユニットに抽出未完了状態のシステムログファイルがある場合，データ抽出処理を完了した後にデータ抽出処理を停止します。pdrplstop -f コマンドを実行すると，正常運転に戻ったユニットに抽出未完了状態のシステムログファイルがある場合でも，システム全体の連動状態に合わせてデータ抽出処理を停止します。

(2)　HiRDB が反映側の場合

- 起動していないユニットにデータを反映しようとするとエラーとなり，反映処理を中止します。この場合，縮退起動が解除されて正常運転になった後に，HiRDB Datareplicator の **hdsstart** コマンドでデータ反映処理を再開してください。
- 縮退起動によって起動していないユニットのデータベースを基にして反映側データベースを再作成しないでください。この場合，抽出側データベースと反映側データベースとの間で不整合が発生することがあります。

20.15.8　次回以降の HiRDB 開始時も縮退起動になる場合の運用

ユニットの障害が早期に解決できないため，次回以降の HiRDB 開始時も縮退起動になる場合は，**pd_start_skip_unit** オペランドに起動できないユニット名を指定してください。HiRDB を開始するとき，HiRDB は各ユニットからの起動開始の連絡を最大 20 分待ちます。したがって，縮退起動をするときは最大 20 分間の起動開始の連絡待ち時間が発生します。**pd_start_skip_unit** オペランドに起動できないユニット名を指定すると，このユニットの起動開始の連絡待ちをしません。最大 20 分間の起動開始の連絡待ち時間が解消されるため，縮退起動の時間を短縮できます。

〈運用例〉

1. pd_start_level オペランドには 1 を指定しています。
2. あるユニットに障害が発生して起動できないため，HiRDB が縮退起動しました。
3. 業務が終了したため，**pdstop** コマンドで HiRDB を終了します。このとき，まだ障害が解決されていません。
4. 次に示すオペランドを指定します。

   pd_start_skip_unit=起動できないユニットの名称
5. 業務を開始するため，**pdstart** コマンドで HiRDB を開始します。HiRDB の起動時間が，pd_start_skip_unit オペランドを指定しないときに比べて短縮されます。

> **注意事項**
>
> 正常運転に戻った場合は pd_start_skip_unit オペランドを削除してください。削除しないと次回の HiRDB 開始時も縮退起動となってしまいます。

20.15.9　回復不要 FES 使用時の注意事項

回復不要 FES ユニットは，pd_start_level オペランドの指定値に関係なく独自に縮退起動をします。このため，pd_start_skip_unit オペランドに該当ユニットの名称を指定しても，指定を無視して回復不要 FES ユニットを開始します。また，一部の回復不要 FES ユニットが HiRDB 開始時に起動できなくても，起動できないユニットを除いて HiRDB を開始します。

20.16　ディスク障害が発生したときの対処方法

20.16.1　対処手順

実行者　HiRDB 管理者及び Administrators 権限保持者

ディスク障害が発生したときの対処手順を次に示します。

〈手順〉

1. ハードディスクを初期化します。
2. パーティションを設定します。
3. HiRDB ファイルシステム領域を初期設定します。
4. システムファイルを作成します。
5. RD エリアを回復します。

手順の数字はこの後で説明している（　）レベルに対応しています。例えば，3 の操作は(3)で説明しています。

(1)　ハードディスクを初期化します

ハードディスクを交換して初期化してください。

(2)　パーティションを設定します

初期化したハードディスクに，前と同じようにパーティションを設定してください。

(3)　HiRDB ファイルシステム領域を初期設定します

実行者　HiRDB 管理者

障害が発生したディスク内に作成していた HiRDB ファイルシステム領域を pdfmkfs コマンドで初期設定します。

(a)　障害が発生したディスク内にシステムファイルを作成していた場合

pdfmkfs コマンドで，システムファイル用の HiRDB ファイルシステム領域を初期設定します。

```
pdfmkfs -n 40 -l 5 -k SYS -i C:¥sysfile_a
```

〔説明〕

-n：HiRDB ファイルシステム領域サイズを指定します。初期設定する HiRDB ファイルシステムの領域サイズは，パーティションの領域長と等しいか，又は小さくしてください。パーティションの領域長より大きくすると，そのパーティションに物理的に続くパーティションを破壊する場合があります。

-l：HiRDB ファイルシステム領域内に作成するファイル数の上限を指定します。

-k　SYS：システムファイル用の HiRDB ファイルシステム領域を指定します。

-i：HiRDB ファイルシステム領域の全エリアを初期設定する指定です。

C:¥sysfile_a：HiRDB ファイルシステム領域名を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(b)　障害が発生したディスク内に RD エリアを作成していた場合

pdfmkfs コマンドで，RD エリア用の HiRDB ファイルシステム領域を初期設定します。

```
pdfmkfs -n 40 -l 5 -k DB -i C:¥dbarea1
```

〔説明〕

-n：HiRDB ファイルシステム領域サイズを指定します。初期設定する HiRDB ファイルシステムの領域サイズは，パーティションの領域長と等しいか，又は小さくしてください。パーティションの領域長より大きくすると，そのパーティションに物理的に続くパーティションを破壊する場合があります。

-l：HiRDB ファイルシステム領域内に作成するファイル数の上限を指定します。

-k　DB：RD エリア用の HiRDB ファイルシステム領域を指定します。

-i：HiRDB ファイルシステム領域の全エリアを初期設定する指定です。

C:¥dbarea1：HiRDB ファイルシステム領域名を指定します。

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(4)　システムファイルを作成します

(3)の(a)で初期設定した HiRDB ファイルシステム領域内に，システムファイルを作成します。

```
pdloginit -d sys -f C:¥sysfile_a¥log1a -n 2000        1
pdloginit -d sys -f C:¥sysfile_a¥log2a -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log3a -n 2000
pdloginit -d sys -f C:¥sysfile_a¥log4a -n 2000
pdloginit -d spd -f C:¥sysfile_a¥sync1 -n 1000        2
pdloginit -d spd -f C:¥sysfile_a¥sync3 -n 1000
pdstsinit -u UNT1 -f C:¥sysfile_a¥ssts1a -c 500       3
pdstsinit -u UNT1 -f C:¥sysfile_a¥ssts2a -c 500
pdstsinit -s bes1 -f C:¥sysfile_a¥b1sts1a -c 500      4
pdstsinit -s bes1 -f C:¥sysfile_a¥b1sts2a -c 500
```

〔説明〕

1. システムログファイルを作成します。
2. シンクポイントダンプファイルを作成します。
3. ユニット用ステータスファイルを作成します。
4. サーバ用ステータスファイルを作成します。

(5)　RD エリアを回復します

障害が発生したディスク内にあった RD エリアを pdrstr コマンドで回復します。RD エリアの回復例については，「データベースの回復方法」を参照してください。

注意事項

障害が発生したディスク内にマスタディレクトリ用 RD エリアがある場合は，pdstart -r コマンドで HiRDB を開始して，pdrstr コマンドでマスタディレクトリ用 RD エリアを回復してください。マスタディレクトリ用 RD エリアに障害が発生したときの対処方法については，「マスタディレクトリ用 RD エリアに障害が発生したときの対処方法」を参照してください。

20.17　システムログファイルの不足によって HiRDB（ユニット）が異常終了したときの対処方法

スワップ先にできる状態のシステムログファイルがないときにシステムログファイルのスワップが発生すると，システムログファイルの不足によって HiRDB（ユニット）が異常終了します。このときの対処方法には，大きく分けて次の表に示す三つの方法があります。

表 20–19　システムログファイルの不足によって HiRDB（ユニット）が異常終了したときの対処方法

対処の際の方針又は状況	手順の参照先
ユニットの再開始を最優先する場合	「ユニットの再開始手順（ユニットの再開始を最優先する場合）」を参照して対処してください。
原因を特定してからユニットを再開始する場合	「ユニットの再開始手順（原因を特定してからユニットを再開始する場合）」を参照して対処してください。
システムログファイルの不足の原因が特定されている場合	「ユニットの再開始手順（システムログファイルの不足原因が特定されている場合）」を参照して対処してください。

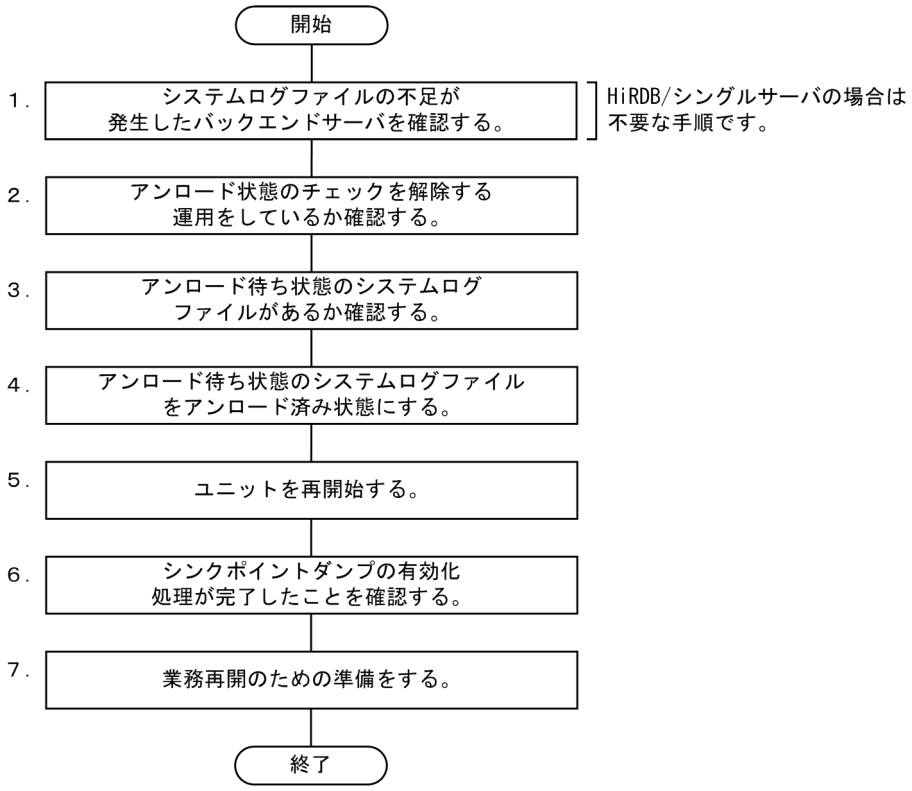
注意事項

HiRDB/シングルサーバを使用している場合は，以降の説明でバックエンドサーバ（BES）をシングルサーバ（SDS）に読み替えてください。また，コマンドを実行する際には，-s オプションを指定しないでください。

20.17.1　ユニットの再開始手順（ユニットの再開始を最優先する場合）

対処手順の大きな流れは，アンロード待ち状態のシステムログファイルがあるかどうかを確認し，ある場合はアンロード待ち状態のシステムログファイルをアンロード済み状態にします。ない場合はシステムログファイルを追加します。対処手順のフローを次の図に示します。

図 20–4　ユニットの再開始を最優先する場合の対処手順（フロー）



注　図中の項番は，以降の説明の（　）の番号と対応しています。

(1)　システムログファイルの不足が発生したバックエンドサーバを確認する

イベントログ又はメッセージログファイルに出力されている KFPS01220-E メッセージを参照して，システムログファイルの不足が発生したバックエンドサーバを確認してください。

（例）

```
KFPS01220-E Request to swap sys(bes1) log file unable to be executed
            because there is no standby log file group available.
```

下線部分にシステムログファイルの不足が発生したバックエンドサーバが表示されます。この例の場合，バックエンドサーバ bes1 でシステムログファイルの不足が発生しています。

(2)　アンロード状態のチェックを解除する運用をしているか確認する

バックエンドサーバ（bes1）で，アンロード状態のチェックを解除する運用をしているか確認してください。pd_log_unload_check オペランドに N を指定している場合は，アンロード状態のチェックを解除する運用をしています。

アンロード状態のチェックを解除する運用をしている場合は，システムログファイルを追加してからユニットを再開始する必要があります。次の手順には進まないで，「ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合）」で説明している手順を実施してください。

アンロード状態のチェックを解除する運用をしていない場合は次の手順に進んでください。

(3)　アンロード待ち状態のシステムログファイルがあるか確認する

pdlogls コマンドを実行して，アンロード待ち状態のシステムログファイルがあるかどうかを確認してください。pdlogls コマンドの-s オプションに(1)で確認したバックエンドサーバ名を指定します。

（例）

```
pdlogls -d sys -s bes1

HOSTNAME : server1(123456)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.        Ex-Status
log001    sys   bes1        7   cn---cu 50beaaab   275    0        --------
log002    sys   bes1        2   cnu---u 50beaaab    72   d7        --------
log003    sys   bes1        3   cnu---u 50beaaab    d8  13e        --------
log004    sys   bes1        4   cnu---u 50beaaab   13f  1a4        --------
log005    sys   bes1        5   cnu---u 50beaaab   1a5  20b        --------
log006    sys   bes1        6   cn----u 50beaaab   20c  274        --------
```

Status 欄にアンロード待ち状態かどうかが表示されます。u 又は a が表示されているシステムログファイルが，アンロード待ち状態のシステムログファイルです。この例の場合，log002～log005 がアンロード待ち状態です。

アンロード待ち状態のシステムログファイルがある場合は，次の手順に進んでください。

アンロード待ち状態のシステムログファイルがない場合は，システムログファイルを追加してからユニットを再開始する必要があります。次の手順には進まないで，「ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合）」で説明している手順を実施してください。

(4)　アンロード待ち状態のシステムログファイルをアンロード済み状態にする

自動ログアンロード機能を使用しているかどうかによって対処方法が異なります。
pd_log_auto_unload_path オペランドを指定している場合は，自動ログアンロード機能を使用しています。

- **自動ログアンロード機能を使用している場合**

  自動ログアンロード機能が停止したため，システムログファイルがアンロードされていない可能性があります。自動ログアンロード機能が停止した場合，KFPS01150-E メッセージが出力されます。イベントログ又はメッセージログファイル中に KFPS01150-E メッセージが出力されているかを確認してください。

  KFPS01150-E メッセージが出力されている場合は，KFPS01150-E メッセージの対処方法に従ってください。また，「障害発生時の対策方法」も参照して対処してください。

参考

自動ログアンロード機能が異常終了した原因を対策し，ユニットを再開始すると，ユニットの再開始処理中に自動ログアンロード機能が動作し，アンロード待ち状態のシステムログファイルはすべてアンロードされます。

KFPS01150-E メッセージが出力されていない場合（自動ログアンロード機能が停止していない場合）は，システムログファイルを追加してからユニットを再開始する必要があります。次の手順には進まないで，「ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合）」で説明している手順を実施してください。

- **自動ログアンロード機能を使用していない場合**

  アンロード待ち状態のシステムログファイルをアンロード済み状態にしてください。手順については，「システムログファイルをアンロード済み状態に変更する方法」を参照してください。

  アンロード待ち状態のシステムログファイルを一つもアンロード済み状態にできない場合は，システムログファイルを追加してからユニットを再開始する必要があります。次の手順には進まないで，「ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合）」で説明している手順を実施してください。

(5)　ユニットを再開始する

pdstart コマンドを実行してユニットを再開始します。

ユニットが再開始されると，トランザクションの受け付け可能状態になりますが，新規トランザクションの処理によって出力されるシステムログが原因で，システムログファイルの不足が再度発生するおそれがあります。そのため，(7)の作業が完了した後に業務を再開するようにしてください。

(6)　シンクポイントダンプの有効化処理が完了したことを確認する

シンクポイントダンプの有効化処理が完了すると，KFPS02183-I メッセージがイベントログ又はメッセージログファイルに出力されます。

（例）

```
KFPS02183-I HRDB unt1 Syncpoint dump for bes1 has been acquired to file group
            spd001. log file information:log001, 7, 287. start time=12:45:00,
            end time=12:45:12
```

下線部分に表示されているサーバ名が，(1)で確認したサーバ名であることを確認してください。この例の場合，バックエンドサーバ bes1 でシンクポイントダンプの有効化処理が完了しています。

なお，システムログファイルの不足が発生したサーバのサーバ定義又はサーバ共通定義の pd_spd_assurance_msg オペランド（この例の場合，bes1 のバックエンドサーバ定義が該当）に N を指定している場合，KFPS02183-I メッセージが出力されません。この場合，pdlogls コマンドの実行結果を参照して，シンクポイントダンプの有効化処理が完了したことを確認してください。

（例）

- **ユニットの再開始直後に実行した pdlogls コマンドの実行結果**

```
pdlogls -d sys -s bes1

HOSTNAME : server1(124412)
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log001   sys  bes1          7   oc-d--u 50beaaab   275  277         --------
log002   sys  bes1          2   cs----u 50beaaab    72   d7         --------
log003   sys  bes1          3   os----u 50beaaab    d8  13e         --------
log004   sys  bes1          4   os----u 50beaaab   13f  1a4         --------
log005   sys  bes1          5   os----u 50beaaab   1a5  20b         --------
log006   sys  bes1          6   os-d--u 50beaaab   20c  274         --------
```

Status 欄に d が表示されているシステムログファイルが，上書きできない状態のシステムログファイルです。この例の場合，log001，及び log006 が上書きできない状態です。

- **シンクポイントダンプの有効化処理の完了後に実行した pdlogls コマンドの実行結果**

```
pdlogls -d sys -s bes1

HOSTNAME : server1(125500)
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log001   sys  bes1          7   os----u 50beaaab   275  287         --------
log002   sys  bes1          8   oc-d--u 50beaaab   288  298         --------
log003   sys  bes1          3   os----u 50beaaab    d8  13e         --------
log004   sys  bes1          4   os----u 50beaaab   13f  1a4         --------
log005   sys  bes1          5   os----u 50beaaab   1a5  20b         --------
log006   sys  bes1          6   os----u 50beaaab   20c  274         --------
```

シンクポイントダンプの有効化処理が完了すると，システムログファイルの上書きできない状態が解除されます。Status 欄の表示が d から-になります。この例の場合，log001，及び log006 のシステムログファイルの上書きできない状態が解除されています。

ユニットの再開始時に実行されるデータベースの回復処理でもシステムログが出力されます。システムログファイルをアンロード済み状態にするだけでは再開始に必要なシステムログファイル容量を用意できなかった場合は，シンクポイントダンプの有効化処理が完了する前に，ユニットが再度異常終了します。この場合，次の手順には進まないで，「ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合）」で説明している手順を実施してください。

(7)　業務再開のための準備をする

業務を再開する前に次に示すことを実施してください。

- **スワップ先にできる状態のシステムログファイルが十分にあるか確認する**

  pdlogls コマンドを実行して，スワップ先にできる状態のシステムログファイルの数を確認してください。全体の 3 分の 2 以上の数があるか確認してください。数が不足している場合は，システムログファイルをアンロードするなどして，スワップ先にできる状態のシステムログファイルの数を増やしてください。

- **自動ログアンロード機能が動作しているか確認する**

自動ログアンロード機能を使用している場合は，pdlogatul -d sys -i コマンドコマンドを実行して，自動ログアンロード機能が動作しているか確認してください。

自動ログアンロード機能が停止している場合は，ユニットの再開始後に出力された KFPS01150-E メッセージを確認し，メッセージの対処に従ってください。その後，pdlogatul -d sys -b コマンドを実行して，自動ログアンロード機能を再開してください。

- **バックアップ情報及びログポイント情報を取得する**

  アンロードレスシステムログ運用をしている場合は，pdcopy コマンドでサーバ内の全 RD エリアのバックアップを取得してください。このとき，-z オプションを指定してログポイント情報ファイルを取得してください。

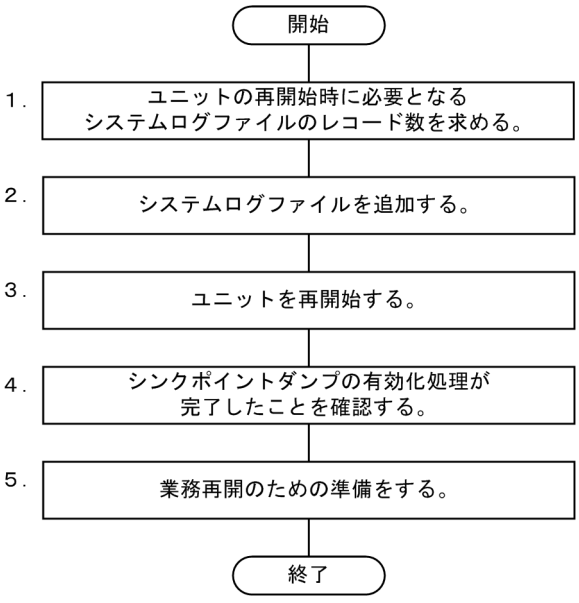
  バックアップを取得した後，pdlogchg -z コマンドを実行して，システムログファイルの状態をアンロード済み状態にしてください。

システムログファイルの不足を解消した後は，「システムログファイルの不足を解消した後に実施すること」の対処を行ってください。

20.17.2　ユニットの再開始手順（システムログファイルを追加してからユニットを再開始する場合）

対処手順の大きな流れは，ユニットの再開始に必要な数のシステムログファイルを追加した後に，ユニットを再開始します。対処手順のフローを次の図に示します。

図 20–5　システムログファイルを追加してからユニットを再開始する手順（フロー）



注　図中の項番は，以降の説明の（　）の番号と対応しています。

(1) ユニットの再開始時に必要となるシステムログファイルのレコード数を求める

ユニットの再開始時，前回の稼働時に使用されていたシステムログファイルを入力情報にして，データベースが回復されます。このデータベースの回復処理でもシステムログが出力されるため，その分のシステムログファイルをユニットの再開始前に追加しておく必要があります。

ユニットの再開始時に必要となるシステムログファイルのレコード数は，次に示す計算式から求められます。

必要なレコード数＝ A×3※

A：前回の稼働時にサーバが使用していたシステムログファイルのレコード数

注※　ユニットの再開始時に実行するデータベースの回復処理でもシステムログが出力されます。この量はデータベースを更新した SQL や HIRDB の運用によって増減します。出力されるシステムログ量が増加してもシステムログファイル不足を解消できるように，係数をかけて必要となるレコード数を求めています。

必要なレコード数の求め方を例題形式で説明します。

例題

バックエンドサーバ（bes1）のシステムログファイルが不足してユニットが異常終了しました。ユニットの再開始時に，バックエンドサーバ（bes1）が必要とするシステムログファイルのレコード数を求めます。

手順

1. pdlogls コマンドを実行して，前回の稼働時に bes1 が使用していたシステムログファイルを確認してください。

```
pdlogls -d sys -s bes1 -e

HOSTNAME : server1(224634)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          1   cn----u 508940f3      1   cf        --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log1     cn-u--     4e16  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log002   sys  bes1          2   cn----u 508940f3    d0  19e       --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log2     cn-u--     4df3  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log003   sys  bes1          3   cn----u 508940f3   19f  26c       --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log3     cn-u--     4df2  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log004   sys  bes1          4   cn----u 508940f3   26d  33a       --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log4     cn-u--     4e14  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log005   sys  bes1          5   cn----u 508940f3   33b  43b       --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log5     cn-u--     6156  61a4      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log006   sys  bes1          6   cn---cu 508940f3   43c    0      --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log6     cn-u--        0  61a4      cc-u--  cc-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log007   sys  bes1          9   cn----u 5089261b   540  59e       --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log7     cn-u--     233d  61a4      cs-u--  cs-u--    4096
```

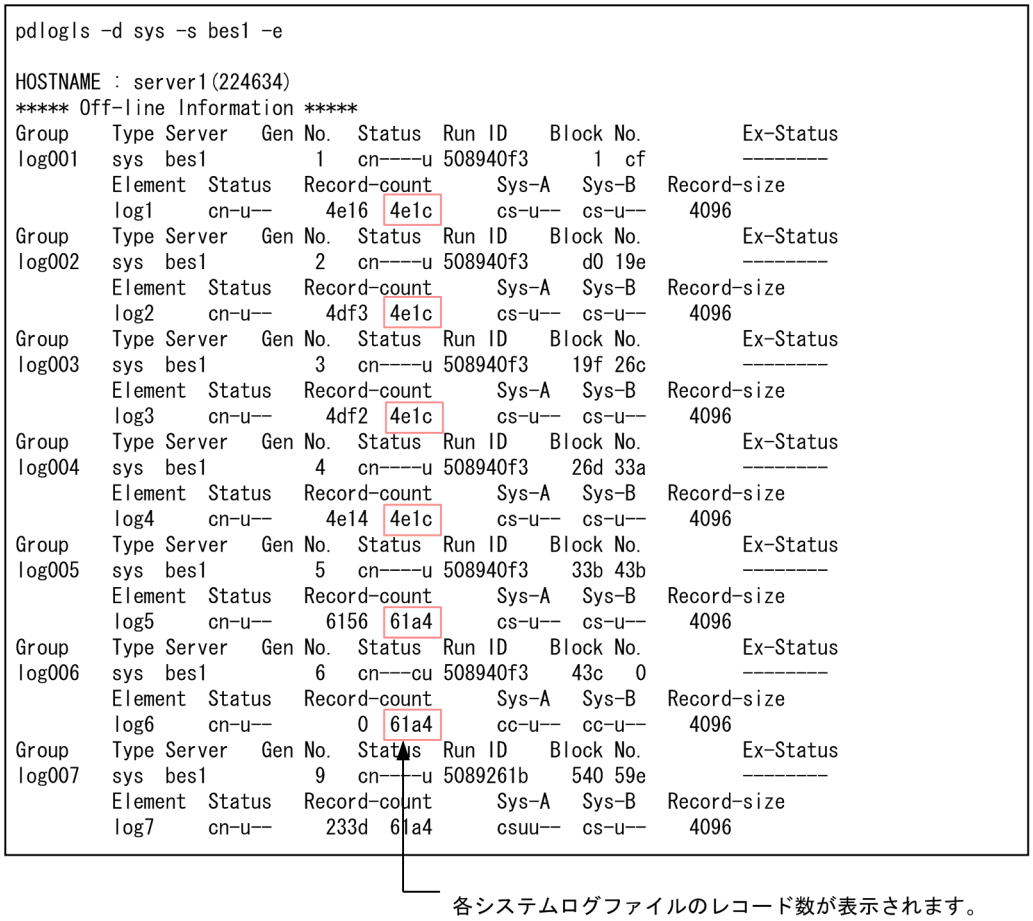
現用ファイルとRun IDが同じファイルを確認します。

cが表示されているファイルが現用

Status 欄を参照して現用ファイルを確認してください。この例の場合，log006 が現用ファイルになります。

次に，現用ファイル（log006）の Run ID を確認してください。現用ファイルと Run ID が同じファイルが，前回の稼働時にバックエンドサーバ（bes1）が使用していたシステムログファイルになります。この例の場合，log001～log006 が該当します。

2. 前回の稼働時に bes1 が使用していたシステムログファイルのレコード数を計算してください。



log001～log006 の Record-count 欄を参照します。Record-count 欄の右側に，各システムログファイルのレコード数が 16 進数で表示されます。

16 進数のレコード数を 10 進数に変換して，log001～log006 の合計を求めてください。

システムログファイル名	レコード数	
	16 進数	10 進数
log001	4e1c	19996
log002	4e1c	19996
log003	4e1c	19996
log004	4e1c	19996
log005	61a4	24996
log006	61a4	24996
合計	－	129976

前回の稼働時に bes1 が使用していたシステムログファイルのレコード数は，129976 レコードになります。

3. ユニットの再開始時に必要となるシステムログファイルのレコード数を計算してください。

   2 で求めたレコード数を 3 倍します。

129976×3 ＝ 389928

ユニットの再開始時に必要となるシステムログファイルのレコード数は，389928 レコードになります。

この値は，システムログファイルの各系で必要なレコード数になります。システムログファイルを二重化している場合は，A 系，B 系の両方で必要になります。

参考

求めたレコード数分のシステムログファイルの容量を知りたい場合は，次に示す計算式で求めてください。

システムログファイルの容量＝レコード数×レコード長（単位：バイト）

この例の場合，389928×4096 ＝ 1597145088 バイトになります。

なお，レコード長は，pdlogls コマンドの実行結果の Record-size で確認できます。

(2)　システムログファイルを追加する

システムログファイル不足が発生したサーバに，システムログファイルを追加してください。追加するシステムログファイルの容量（レコード数）は，(1)で求めたレコード数になります。

システムログファイルの追加方法については，「システムログファイルの追加方法」を参照してください。

(3)　ユニットを再開始する

pdstart コマンドを実行してユニットを再開始します。

ユニットが再開始されると，トランザクションの受け付け可能状態になりますが，新規トランザクションの処理によって出力されるシステムログが原因で，システムログファイルの不足が再度発生するおそれがあります。そのため，(5)の作業が完了した後に業務を再開するようにしてください。

(4)　シンクポイントダンプの有効化処理が完了したことを確認する

シンクポイントダンプの有効化処理が完了したことを確認してください。確認手順については，「シンクポイントダンプの有効化処理が完了したことを確認する」を参照してください。

再開始時に必要なシステムログファイルが不足している場合，シンクポイントダンプの有効化処理が完了する前に，ユニットが再度異常終了します。この場合，イベントログ又はメッセージログファイルに出力されているエラーメッセージの対処に従ってください。システムログファイルの不足が解消していない場合は，トラブルシュート情報を取得して保守員に連絡してください。トラブルシュート情報の取得方法については，「障害が発生したときに HiRDB 管理者がすること」を参照してください。

(5)　業務再開のための準備をする

業務再開のための準備を行ってください。詳細については，「業務再開のための準備をする」を参照してください。

システムログファイルの不足を解消した後は，「システムログファイルの不足を解消した後に実施すること」の対処を行ってください。

20.17.3　ユニットの再開始手順（原因を特定してからユニットを再開始する場合）

システムログファイルの不足原因の一覧を次の表に示します。この表に記載されている原因が該当するかどうかを一つずつ確認していきます。

なお，不足原因は一つとはかぎりません。条件を満たすすべての原因調査をしてください。

表 20–20　システムログファイルの不足原因の一覧

項番	システムログファイルの不足原因	調査方法を説明している箇所	備考
1	システムログファイルをアンロード又は状態変更しなかったため，アンロード待ち状態のファイルが増えて，システムログファイル不足が発生した。	「アンロード待ち状態のファイル数を確認する」	アンロード状態のチェックを解除する運用をしている場合は，調査の必要はありません。
2	自動ログアンロード機能が停止したため，アンロード待ち状態のファイルが増えて，システムログファイル不足が発生した。		自動ログアンロード機能を使用していない場合は，調査の必要はありません。
3	長時間にわたりトランザクションが決着しなかったため，上書きできない状態のファイルが増えて，システムログファイル不足が発生した。	「上書きできない状態のファイル数を確認する」	なし
4	シンクポイントダンプの有効化処理がスキップされたため，上書きできない状態のファイルが増えて，システムログファイル不足が発生した。		なし
5	システムログファイルに障害が発生したため，スワップ先にできる状態のファイルがなくなり，システムログファイル不足が発生した。	「KFPS01202-E メッセージが出力されていたか確認する」	なし
6	HiRDB Datareplicator との連動中に，システムログの抽出処理が追い付かなくなり，抽出未完了状態のファイルが増えて，システムログファイル不足が発生した。	「抽出未完了状態のファイル数を確認する」	HiRDB Datareplicator を使用していない場合は，調査の必要はありません。

システムログファイルの不足原因を特定してからユニットを再開始する手順を以降で説明します。

(1)　システムログファイルの不足が発生したバックエンドサーバを確認する

イベントログ又はメッセージログファイルに出力されている KFPS01220-E メッセージを参照して，システムログファイルの不足が発生したバックエンドサーバを確認してください。

（例）

```
KFPS01220-E Request to swap sys(bes1) log file unable to be executed
            because there is no standby log file group available.
```

下線部分にシステムログファイルの不足が発生したバックエンドサーバが表示されます。この例の場合，バックエンドサーバ bes1 でシステムログファイルの不足が発生しています。

(2)　アンロード待ち状態のファイル数を確認する

pdlogls コマンドを実行して，(1)で確認したバックエンドサーバのシステムログファイルの状態を確認してください。

（例）

```
pdlogls -d sys -s bes1

HOSTNAME : server1(170221)
***** Off-line Information *****
Group    Type Server   Gen No.   Status  Run ID    Block No.          Ex-Status
log001   sys  bes1          a   cn----u 50c046bb    7e2 870          --------
log002   sys  bes1          1   cn----u 50c0504f      1  d0          --------
log003   sys  bes1          2   cru---u 50c0504f     d1 1a0          --------
log004   sys  bes1          3   cru---u 50c0504f    1a1 270          --------
log005   sys  bes1          4   cru---u 50c0504f    271 373          --------
log006   sys  bes1          5   cru---u 50c0504f    374 476          --------
log007   sys  bes1          6   cn---cu 50c0504f    477   0          --------
```

現用ファイルとRun IDが同じファイルを確認します。

cが表示されているファイルが現用

u又はaが表示されているファイルがアンロード待ち状態
-又はeが表示されているファイルがアンロード済み状態

アンロード待ち状態のファイル数から原因を判断します。次に示す条件式を満たす場合は，アンロード待ち状態のファイルが増えたため，システムログファイル不足が発生したと判断できます。

(A ＋ 1)≧↑B÷3↑

A：アンロード待ち状態で，現用ファイルと同じ Run ID を持つシステムログファイルの数

B：現用ファイルと同じ Run ID を持つシステムログファイルの数

上記の例の場合，A の条件を満たすファイルは，log003～log006 の 4 ファイルになります。B の条件を満たすファイルは，log002～log007 の 6 ファイルになります。これらの数値を計算式に代入すると，5≧↑

6÷3↑となり，条件式を満たします。よって，アンロード待ち状態のファイルが増えたため，システムログファイル不足が発生したと判断できます。

また，自動ログアンロード機能を使用している場合に条件式を満たしたときは，自動ログアンロード機能が停止していないか確認してください。ユニットが異常終了する前に，イベントログ又はメッセージログファイルに KFPS01150-E メッセージが出力されていないかどうを確認してください。KFPS01150-E メッセージが出力されている場合は，自動ログアンロード機能が停止しています。これが原因でシステムログファイルの不足が発生したと判断できます。

(3)　上書きできない状態のファイル数を確認する

上書きできない状態のファイル数を確認する手順を例題形式で説明します。

シンクポイントダンプファイルの有効保証世代数が 1 の場合（pd_spd_assurance_count オペランドに 1 を指定しているか，指定を省略している場合）は，次に示す方法で求めてください。

- 「例題 1　再開始に入力するシステムログを示すメッセージ（KFPS01229-I）から求める方法」

シンクポイントダンプファイルの有効保証世代数が 2 の場合（pd_spd_assurance_count オペランドに 2 を指定している場合）は，次に示すどちらかの方法で求めてください。さらにもう 1 世代前のシンクポイントダンプの有効化時点にさかのぼって上書きできない状態のシステムログファイルを求めます。

- 「例題 2　シンクポイントダンプの有効化完了メッセージ（KFPS02183-I）から求める方法」
- 「例題 3　シンクポイントダンプの有効化スキップメッセージ（KFPS02179-I）から求める方法」

(a)　例題 1　再開始に入力するシステムログを示すメッセージ（KFPS01229-I）から求める方法

システムログファイルの不足原因が，上書きできない状態のファイルが増えたことによるものかを調べます。なお，シンクポイントダンプファイルの有効保証世代数は 1 とします。

手順

1. pdlogls コマンドを実行して，(1)で確認したバックエンドサーバのシステムログファイルの情報を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(190649)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.        Ex-Status
log001   sys  bes1          a   cn----u 50c046bb   7e2  870        --------
log002   sys  bes1          1   cn----u 50c06d0b     1   d0        --------
log003   sys  bes1          2   cn----u 50c06d0b    d1  132        --------
log004   sys  bes1          3   cn----u 50c06d0b   133  201        --------
log005   sys  bes1          4   cn----u 50c06d0b   202  305        --------
log006   sys  bes1          5   cn----u 50c06d0b   306  408        --------
log007   sys  bes1          6   cn---cu 50c06d0b   409    0        --------
```

現用ファイルとRun IDが同じファイルを確認します。

cが表示されているファイルが現用

世代番号

次に示す情報を確認します。

・現用ファイルと同じ Run ID を持つファイル

・上記のファイルの Gen No.（ファイルの世代番号）

この例の場合，log002～log007 が現用ファイルと同じ Run ID を持っています。各ファイルの世代番号は次のようになります。

log002：世代番号 1

log003：世代番号 2

log004：世代番号 3

log005：世代番号 4

log006：世代番号 5

log007：世代番号 6（現用ファイル）

2. KFPS01229-I メッセージを参照して，ユニットの再開始時に入力情報となるシステムログファイルの情報を確認してください。

```
KFPS01229-I HRDB UNIT1 Next bes1 log file restart point, generation number=3, block number=133.
            restart end point, generation number=6, block number=50c.  last acquired syncpoint
            dump 2012/12/06 19:04:25
```

ユニットの再開始時に入力されるシステムログファイルの世代番号が表示されます（先頭のシステムログファイルの世代番号が表示されます）。

このメッセージには，ユニットの再開始時に入力情報となるシステムログファイルの情報が表示されます。この例の場合，世代番号 3 のシステムログファイルが，ユニットの再開始時に入力される先頭のシステムログファイルになります。

したがって，世代番号が 3 以降のシステムログファイル（log004～log007）が，上書きできない状態であることが分かります。

参考

ユニットの再開始時に実行されるデータベースの回復処理で入力情報となるシステムログファイルは，上書きできない状態となっています。そのため，世代番号が 3 以降のシステムログファイル（log004～log007）が，上書きできない状態であることが分かります。

注意事項

次に示す場合，再開始しようとしたユニットが再度異常終了しているため，システムログファイル不足を示す KFPS01220-E メッセージが二つ以上出力されています。

- pd_mode_conf オペランドに AUTO 又は MANUAL1 が指定している
- システムログファイルの不足によるユニットの異常終了後に pdstart コマンドを実行している

この場合，ユニットが最初に異常終了したときに出力された KFPS01220-E メッセージの前に出力された KFPS01229-I メッセージを確認してください。

3. 上書きできない状態のファイル数から原因を判断します。

次に示す条件式を満たす場合は，上書きできない状態のファイルが増えたため，システムログファイル不足が発生したと判断できます。

A≧↑B÷C↑

A：上書きできない状態で，現用ファイルと同じ Run ID を持つシステムログファイルの数

B：現用ファイルと同じ Run ID を持つシステムログファイルの数

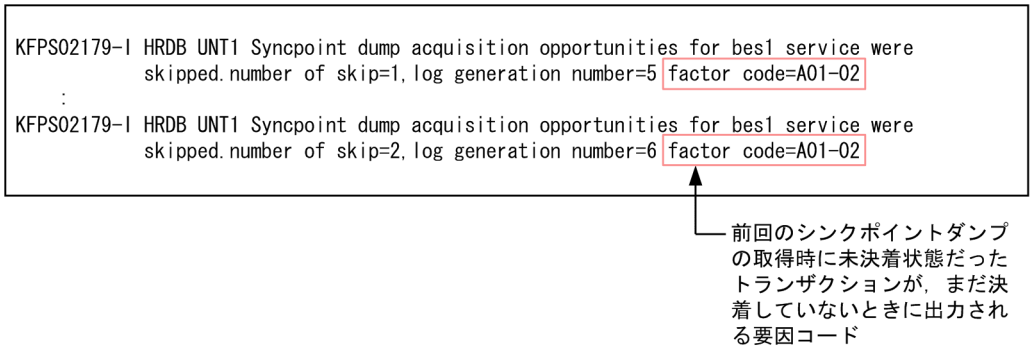
C：次に示すどちらかの値

・HiRDB Text Search Plug-in を使用している場合：4

・それ以外の場合：3

この例の場合，A の条件を満たすファイルは，log004～log007 の 4 ファイルになります。B の条件を満たすファイルは，log002～log007 の 6 ファイルになります。これらの数値を計算式に代入すると，4≧↑6÷3↑となり，条件式を満たします。よって，上書きできない状態のファイルが増えたため，システムログファイル不足が発生したと判断できます。

4. KFPS02179-I メッセージを参照して，上書きできない状態のファイルが増えた原因を特定してください。



KFPS02179-I メッセージは，シンクポイントダンプの有効化処理をスキップしたときに出力されるメッセージです。factor code=A01-02 のメッセージが何回か出力されているため，長時間にわたりトランザクションが決着しなかったことが理由で，上書きできない状態のファイルが増えて，システムログファイル不足が発生したと判断できます。

なお，要因コード（factor code）が異なる KFPS02179-I メッセージが複数出力されている場合は，最も多く出力された要因コードで原因を特定します。

要因コードが A01-02 の場合は，長時間にわたりトランザクションが決着しなかったことが理由で，上書きできない状態のファイルが増えて，システムログファイル不足が発生したと判断します。

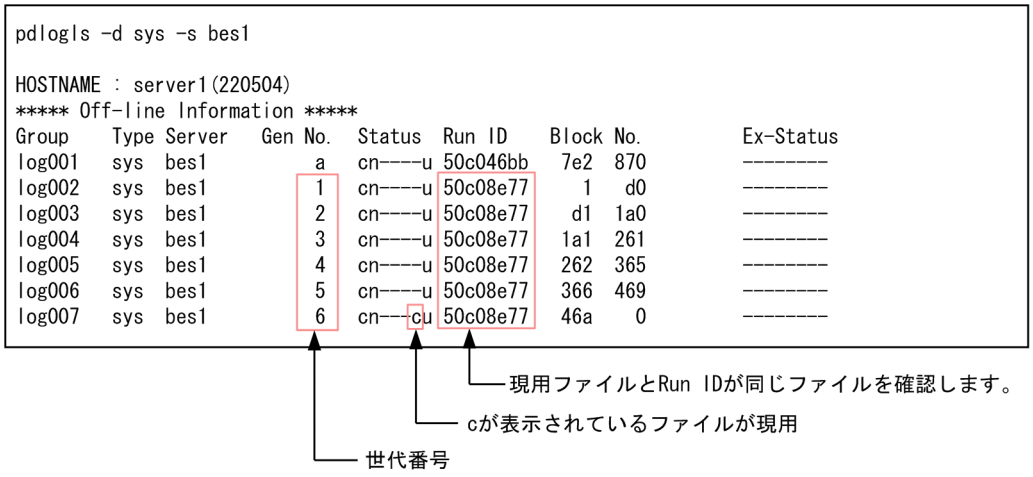
要因コードが A01-02 以外の場合は，シンクポイントダンプの有効化処理がスキップされたことが理由で，上書きできない状態のファイルが増えて，システムログファイル不足が発生したと判断します。

(b) 例題 2 シンクポイントダンプの有効化完了メッセージ（KFPS02183-I）から求める方法

システムログファイルの不足原因が，上書きできない状態のファイルが増えたことによるものかを調べます。なお，シンクポイントダンプファイルの有効保証世代数は 2 とします。

手順

1. pdlogls コマンドを実行して，(1)で確認したバックエンドサーバのシステムログファイルの情報を確認してください。



次に示す情報を確認します。

・現用ファイルと同じ Run ID を持つファイル

・上記のファイルの Gen No.（ファイルの世代番号）

この例の場合，log002～log007 が現用ファイルと同じ Run ID を持っています。各ファイルの世代番号は次のようになります。

log002：世代番号 1

log003：世代番号 2

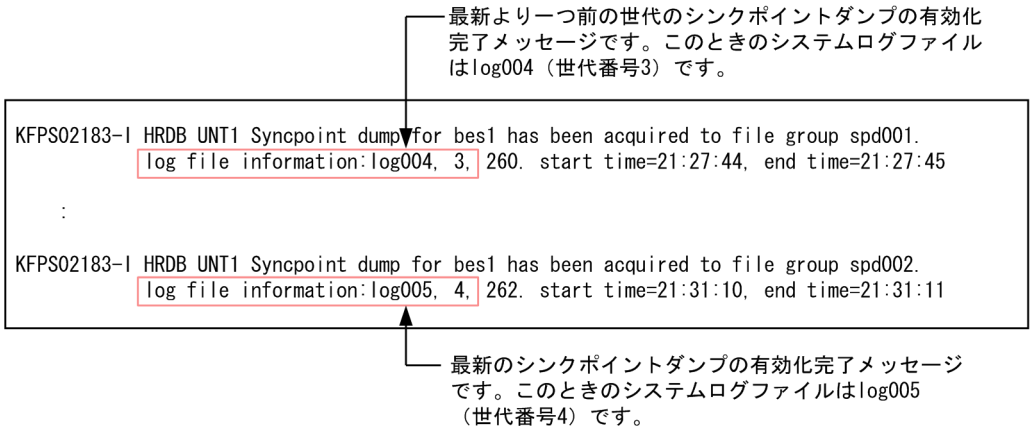
log004：世代番号 3

log005：世代番号 4

log006：世代番号 5

log007：世代番号 6（現用ファイル）

2. KFPS02183-I メッセージを参照して，ユニットの再開始時に入力情報となるシステムログファイルの情報を確認してください。



このメッセージは，シンクポイントダンプの有効化完了時に出力されます。シンクポイントダンプファイルの有効保障世代数が 2 であるため，2 世代前までのシンクポイントダンプファイルに対応するシステムログファイルが上書きできない状態になっています。最新より一つ前の世代のシンクポイントダンプの有効化完了メッセージに log004（世代番号 3）が表示されているため，世代番号が 3 以降のシステムログファイル（log004～log007）が，上書きできない状態であることが分かります。

注意事項

次に示す場合，再開始しようとしたユニットが再度異常終了しているため，システムログファイル不足を示す KFPS01220-E メッセージが二つ以上出力されています。

- pd_mode_conf オペランドに AUTO 又は MANUAL1 が指定している
- システムログファイルの不足によるユニットの異常終了後に pdstart コマンドを実行している

この場合，ユニットが最初に異常終了したときに出力された KFPS01220-E メッセージの前に出力された KFPS02183-I メッセージを確認してください。

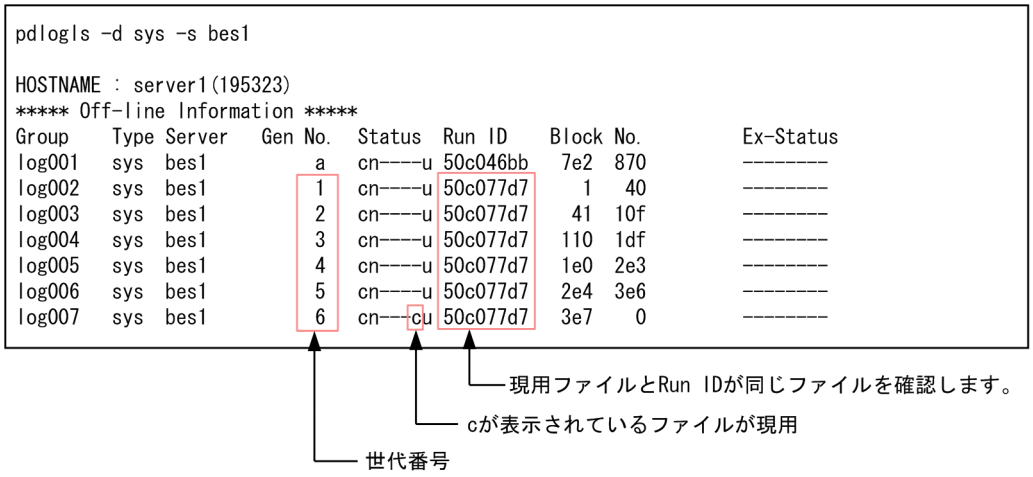
以降の手順は，「例題 1　再開始に入力するシステムログを示すメッセージ（KFPS01229-I）から求める方法」の手順の 3 以降と同じになります。

(c)　例題 3　シンクポイントダンプの有効化スキップメッセージ（KFPS02179-I）から求める方法

システムログファイルの不足原因が，上書きできない状態のファイルが増えたことによるものかを調べます。なお，シンクポイントダンプファイルの有効保証世代数は 2 とします。

手順

1. pdlogls コマンドを実行して，(1)で確認したバックエンドサーバのシステムログファイルの情報を確認してください。



次に示す情報を確認します。

・現用ファイルと同じ Run ID を持つファイル

・上記のファイルの Gen No.（ファイルの世代番号）

この例の場合，log002～log007 が現用ファイルと同じ Run ID を持っています。各ファイルの世代番号は次のようになります。

log002：世代番号 1

log003：世代番号 2

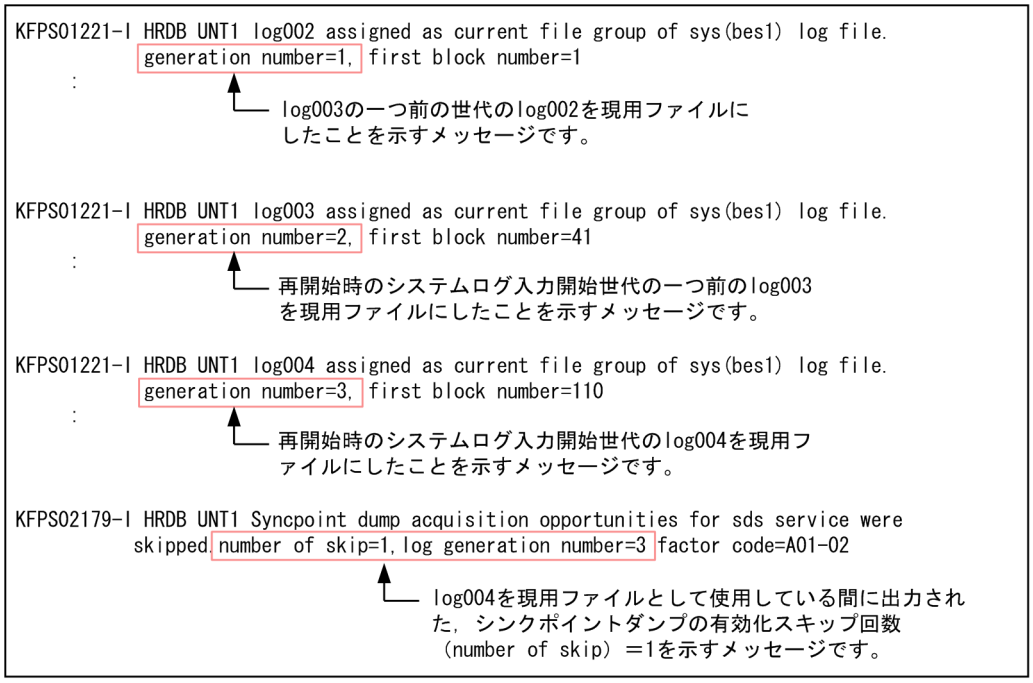
log004：世代番号 3

log005：世代番号 4

log006：世代番号 5

log007：世代番号 6（現用ファイル）

2. KFPS02179-I メッセージが出力されていることを確認してください。



再開始時のシステムログ入力開始世代のシステムログファイルである log004 を現用として使用している間に，有効化スキップ回数（number of skip）＝ 1 を示す KFPS02179-I メッセージが出力されています。そのため，一つ前の log003 が現用ファイルとして割り当てられていたときに KFPS02179-I メッセージが出力されているか確認して，上書きできない状態のシステムログファイルを特定します。

この例では，log003 が現用ファイルとして割り当てられていたときに KFPS01279-I メッセージが出力されていません。したがって，再開始時のシステムログ入力開始世代のシステムログファイルの二つ前に割り当てられたシステムログファイルである log002（世代番号 1）以降の世代番号のシステムログファイルが，上書きできない状態のシステムログファイルになります。

したがって，世代番号が 1 以降のシステムログファイル（log002～log007）が，上書きできない状態であることが分かります。

注意事項

次に示す場合，再開始しようとしたユニットが再度異常終了しているため，システムログファイル不足を示す KFPS01220-E メッセージが二つ以上出力されています。

- pd_mode_conf オペランドに AUTO 又は MANUAL1 が指定している
- システムログファイルの不足によるユニットの異常終了後に pdstart コマンドを実行している

この場合，ユニットが最初に異常終了したときに出力された KFPS01220-E メッセージの前に出力された KFPS02183-I メッセージを確認してください。

以降の手順は，「例題 1　再開始に入力するシステムログを示すメッセージ（KFPS01229-I）から求める方法」の手順の 3 以降と同じになります。

■手順の 2 の考え方（補足）

最初に，再開始時の入力開始世代のシステムログファイルを現用ファイルとして割り当てたことを示す KFPS01221-I メッセージを探します。

次に，再開始時の入力開始世代のシステムログファイルを現用ファイルとして使用している間に，有効化スキップ回数（number of skip）＝ 1 を示す KFPS02179-I メッセージが出力されているかを探します。

KFPS02179-I メッセージが出力されていない場合は，再開始時の入力世代の一つ前のシステムログファイルが上書きできない状態のファイルになります。

KFPS02179-I メッセージが出力されている場合は，再開始時の入力開始世代の一つ前のシステムログファイルを現用として使用している間に，KFPS02179-I メッセージが出力されているかを探します。

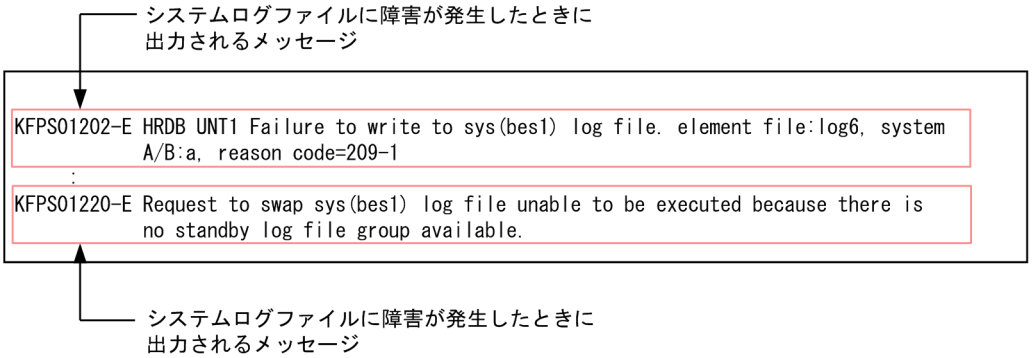
再開始時の入力開始世代の一つ前のシステムログファイルを現用ファイルとして使用している間に，KFPS02179-I メッセージが出力されていない場合は，再開始時の入力世代の二つ前のシステムログファイルが上書きできない状態のシステムログファイルになります。

再開始時の入力開始世代の一つ前のシステムログファイルを現用ファイルとして使用している間に，KFPS02179-I メッセージが出力されている場合は，有効化スキップ回数を確認します。そこからシンクポイントの有効化スキップ回数＝ 1 を示す KFPS02179-I メッセージまで遡ります。このメッセージが出力されたときの一つ前に使用していたシステムログファイルの世代番号以降が，上書きできないシステムログファイルになります。

(4)　KFPS01202-E メッセージが出力されていたか確認する

システムログファイル不足によってユニットが異常終了する直前に，イベントログ又はメッセージログファイルに KFPS01202-E メッセージが出力されていないか確認してください。

（例）



この例のように，KFPS01202-E メッセージが出力された直後に，KFPS01220-E メッセージが出力されている場合，スワップ先にできる状態のファイルに障害が発生したため，システムログファイル不足が発生したと判断できます。

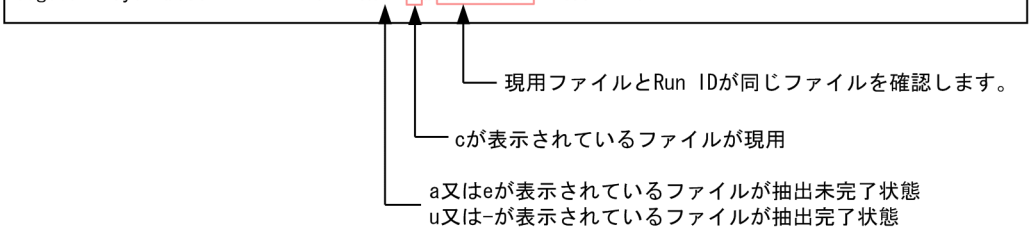
(5)　抽出未完了状態のファイル数を確認する

pdlogls コマンドを実行して，(1)で確認したバックエンドサーバのシステムログファイルの状態を確認してください。

（例）

```
pdlogls -d sys -s bes1

HOSTNAME : server1(170221)
***** Off-line Information *****
Group    Type Server   Gen No.   Status  Run ID    Block No.          Ex-Status
log001   sys  bes1           a   cn----u 50c046bb   7e2  870         --------
log002   sys  bes1           1   cn----u 50c0504f     1   d0         --------
log003   sys  bes1           2   cne---u 50c0504f    d1  1a0         --------
log004   sys  bes1           3   cne---u 50c0504f   1a1  270         --------
log005   sys  bes1           4   can---u 50c0504f   271  373         --------
log006   sys  bes1           5   can---u 50c0504f   374  476         --------
log007   sys  bes1           6   cn---cu 50c0504f   477    0         --------
```



抽出未完了状態のファイル数から原因を判断します。次に示す条件式を満たす場合は，システムログの抽出処理が追い付かなくなり，抽出未完了状態のファイルが増えたため，システムログファイル不足が発生したと判断できます。

(A ＋ 1)≧↑(B×3)÷4↑

A：抽出未完了状態で，現用ファイルと同じ Run ID を持つシステムログファイルの数

B：現用ファイルと同じ Run ID を持つシステムログファイルの数

上記の例の場合，A の条件を満たすファイルは，log003～log006 の 4 ファイルになります。B の条件を満たすファイルは，log002～log007 の 6 ファイルになります。これらの数値を計算式に代入すると，5≧↑(6×3)÷4↑となり，条件式を満たします。よって，抽出未完了状態のファイルが増えたため，システムログファイル不足が発生したと判断できます。

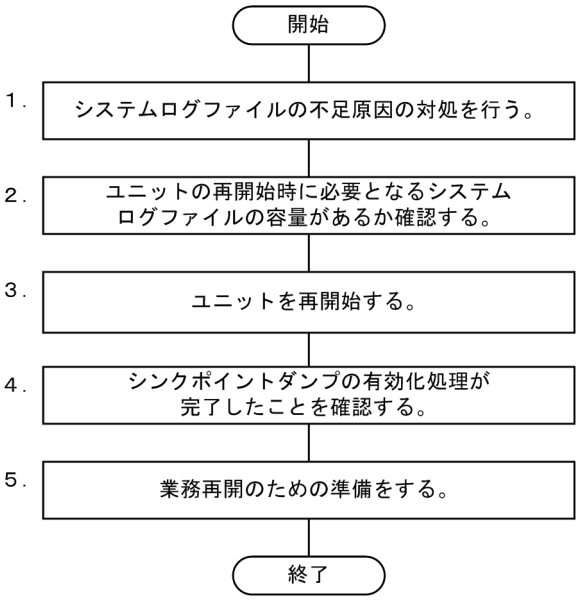
(6)　特定した原因の対処を行う

「ユニットの再開始手順（システムログファイルの不足原因が特定されている場合）」を参照して，特定した原因の対処を行い，ユニットを再開始してください。

20.17.4　ユニットの再開始手順（システムログファイルの不足原因が特定されている場合）

システムログファイルの不足原因が特定されている場合の対処手順を次の図に示します。

図 20–6　システムログファイルの不足原因が特定されている場合の対処手順（フロー）



注　図中の項番は，以降の説明の（　）の番号と対応しています。

(1)　システムログファイルの不足原因の対処を行う

システムログファイルの不足原因ごとの対処方法を次の表に示します。複数の原因が該当する場合は，該当するすべての原因の対処を行ってください。

表 20–21　システムログファイルの不足原因ごとの対処方法

項番	システムログファイルの不足原因	対処方法
1	自動ログアンロード機能が停止したため，アンロード待ち状態のファイルが増えて，システムログファイル不足が発生した。	KFPS01150-E メッセージが出力されています。このメッセージの対処に従って，自動ログアンロード機能が停止した原因を取り除いてください。 また，「障害発生時の対策方法」も参照してください。
2	システムログファイルをアンロード又は状態変更しなかったため，アンロード待ち状態のファイルが増えて，システムログファイル不足が発生した。	アンロード待ち状態のシステムログファイルをアンロード済み状態にしてください。操作方法については，「システムログファイルをアンロード済み状態に変更する方法」を参照してください。
3	HiRDB Datareplicator との連動中に，システムログの抽出処理が追い付かなくなり，抽出未完了状態のファイルが増えて，システムログファイル不足が発生した。	対処方法については，「抽出未完了状態のシステムログファイルがある場合の対処方法」を参照してください。
4	長時間にわたりトランザクションが決着しなかったため，上書きできない状態のファイルが増えて，システムログファイル不足が発生した。	次に示す手順で対処してください。 1. ユニットの再開始時に必要となるシステムログファイルのレコード数を計算します。詳細については，「ユニットの再開始時に必要となるレコード数の求め方」を参照してください。
5	シンクポイントダンプの有効化処理がスキップされたため，上書きできない状態のファイ	

項番	システムログファイルの不足原因	対処方法
	ルが増えて，システムログファイル不足が発生した。	2. 1 で求めたレコード数以上のシステムログファイルを追加します。詳細については，「システムログファイルの追加方法」を参照してください。
6	システムログファイルに障害が発生したため，スワップ先にできる状態のファイルがなくなり，システムログファイル不足が発生した。	

(2) ユニットの再開始時に必要となるシステムログファイルの容量があるか確認する

ここで説明する作業は，表「システムログファイルの不足原因ごとの対処方法」の項番 1〜3 の対処をした場合にかぎり実施してください。

ユニットの再開始時に必要となるシステムログファイルの容量があるか確認してください。確認手順を次に示します。

手順

1. pdlogls -d sys -e コマンドを実行して，スワップ先にできる状態のシステムログファイルを確認してください。

   表「システムログファイルの不足原因ごとの対処方法」の項番 1 の対処（自動ログアンロード機能が停止した場合の対処）を実施している場合は，ユニットの再開始時に自動ログアンロード機能が動作してシステムログをアンロードするため，システムログファイルはすべてアンロード済み状態であると見なしてください。
2. スワップ先にできる状態のシステムログファイルのレコード数を合計してください。pdlogls -d sys -e コマンドの実行結果を参照して，スワップ先にできる状態の各システムログファイルの Record count の右側の値（システムログファイルのレコード数が表示されています）を合計してください。
3. 「ユニットの再開始時に必要となるレコード数の求め方」を参照して，ユニットの再開始時に必要となるシステムログファイルのレコード数を計算してください。

上記の手順の 2 と 3 で求めたレコード数を比較して，次に示す条件式を満たしている場合は，(3)に進んでください。

- スワップ先にできる状態のシステムログファイルのレコード数（手順の 2 で求めたレコード数）≧ユニットの再開始時に必要となるシステムログファイルのレコード数（手順の 3 で求めたレコード数）

上記の条件式を満たしていない場合は，不足分のレコード数を定義したシステムログファイルを追加してください。追加手順については，「システムログファイルの追加方法」を参照してください。

(3) ユニットを再開始する

pdstart コマンドを実行してユニットを再開始します。

ユニットが再開始されると，トランザクションの受け付け可能状態になりますが，新規トランザクションの処理によって出力されるシステムログが原因で，システムログファイルの不足が再度発生するおそれがあります。そのため，(5)の作業が完了した後に業務を再開するようにしてください。

(4)　シンクポイントダンプの有効化処理が完了したことを確認する

シンクポイントダンプの有効化処理が完了したことを確認してください。確認手順については，「シンクポイントダンプの有効化処理が完了したことを確認する」を参照してください。

再開始時に必要なシステムログファイルが不足している場合，シンクポイントダンプの有効化処理が完了する前に，ユニットが再度異常終了します。この場合，イベントログ又はメッセージログファイルに出力されているエラーメッセージの対処に従ってください。システムログファイルの不足が解消していない場合は，「ユニットの再開始手順（原因を特定してからユニットを再開始する場合）」を参照して原因を特定してください。

(5)　業務再開のための準備をする

業務を再開する前に次に示すことを実施してください。

- **スワップ先にできる状態のシステムログファイルが十分にあるか確認する**

  pdlogls コマンドを実行して，スワップ先にできる状態のシステムログファイルの数を確認してください。全体の 3 分の 2 以上の数があるか確認してください。数が不足している場合は，システムログファイルをアンロードするなどして，スワップ先にできる状態のシステムログファイルの数を増やしてください。

- **自動ログアンロード機能が動作しているか確認する**

  自動ログアンロード機能を使用している場合は，pdlogatul -d sys -i コマンドを実行して，自動ログアンロード機能が動作しているか確認してください。

  自動ログアンロード機能が停止している場合は，ユニットの再開始後に出力された KFPS01150-E メッセージを確認し，メッセージの対処に従ってください。その後，pdlogatul -d sys -b コマンドを実行して，自動ログアンロード機能を再開してください。

- **バックアップ情報及びログポイント情報を取得する**

  アンロードレスシステムログ運用をしている場合は，pdcopy コマンドでサーバ内の全 RD エリアのバックアップを取得してください。このとき，-z オプションを指定してログポイント情報ファイルを取得してください。

  バックアップを取得した後，pdlogchg -z コマンドを実行して，システムログファイルの状態をアンロード済み状態にしてください。

システムログファイルの不足を解消した後は，「システムログファイルの不足を解消した後に実施すること」の対処を行ってください。

20.17.5　システムログファイルをアンロード済み状態に変更する方法

システムログファイルをアンロード済み状態に変更する方法を説明します。

(1)　システムログをアンロードする運用の場合

手順を例題形式で説明します。

例題

バックエンドサーバ（bes1）のシステムログファイルをアンロード済み状態に変更します。

手順

1. pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(123456)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          7   cn---cu 50beaaab    275   0         --------
log002   sys  bes1          2   cnu---u 50beaaab     72  d7         --------
log003   sys  bes1          3   cnu---u 50beaaab     d8 13e         --------
log004   sys  bes1          4   cnu---u 50beaaab    13f 1a4         --------
log005   sys  bes1          5   cnu---u 50beaaab    1a5 20b         --------
log006   sys  bes1          6   cn----u 50beaaab    20c 274         --------
```

Status 欄が u 又は a のシステムログファイルが，アンロード待ち状態のシステムログファイルです。この例の場合，log002～log005 のシステムログファイルがアンロード待ち状態です。

2. pdlogunld コマンドを実行して，システムログのアンロードを行ってください。

```
pdlogunld -d sys -s bes1 -g log002 -o c:¥hirdb¥unload_dir¥log2unld
pdlogunld -d sys -s bes1 -g log003 -o c:¥hirdb¥unload_dir¥log3unld
pdlogunld -d sys -s bes1 -g log004 -o c:¥hirdb¥unload_dir¥log4unld
pdlogunld -d sys -s bes1 -g log005 -o c:¥hirdb¥unload_dir¥log5unld
```

1 で確認したアンロード待ち状態のシステムログファイルに対して，pdlogunld コマンドを実行します。

3. pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(124021)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          7   cn---cu 50beaaab   275    0         --------
log002   sys  bes1          2   cn----u 50beaaab    72   d7         --------
log003   sys  bes1          3   cn----u 50beaaab    d8   13e        --------
log004   sys  bes1          4   cn----u 50beaaab   13f   1a4        --------
log005   sys  bes1          5   cn----u 50beaaab   1a5   20b        --------
log006   sys  bes1          6   cn----u 50beaaab   20c   274        --------
```

Status 欄が-又は e のシステムログファイルが，アンロード済み状態のシステムログファイルです。この例の場合，log002～log005 のシステムログファイルが，アンロード待ち状態からアンロード済み状態に変わっています。

(2)　アンロードレスシステムログ運用の場合

手順を例題形式で説明します。

なお，手順は二つあります。ログポイント以降のシステムログファイルを使用する場合は手順 1 を，ログポイント以降のシステムログファイルを使用しない場合は手順 2 を実施してください。

例題

バックエンドサーバ（bes1）のシステムログファイルをアンロード済み状態に変更します。

手順 1（ログポイント以降のシステムログファイルを使用する場合）

1. pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(123456)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          7   cn---cu 50beaaab   275    0         --------
log002   sys  bes1          2   cnu---u 50beaaab    72   d7         --------
log003   sys  bes1          3   cnu---u 50beaaab    d8   13e        --------
log004   sys  bes1          4   cnu---u 50beaaab   13f   1a4        --------
log005   sys  bes1          5   cnu---u 50beaaab   1a5   20b        --------
log006   sys  bes1          6   cn----u 50beaaab   20c   274        --------
```

Status 欄が u 又は a のシステムログファイルが，アンロード待ち状態のシステムログファイルです。この例の場合，log002～log005 のシステムログファイルがアンロード待ち状態です。

2. pdlogchg コマンドを実行して，システムログファイルの状態を変更してください。

```
pdlogchg -d sys -s bes1 -z c:¥hirdb¥logpoint_dir¥logpoint
```

-z オプションにログポイント情報ファイルを指定します。ログポイントより古いすべてのシステムログファイルの状態が変わります。

3. pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(124021)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          7   cn---cu 50beaaab   275    0         --------
log002   sys  bes1          2   cn----u 50beaaab    72   d7         --------
log003   sys  bes1          3   cn----u 50beaaab    d8   13e        --------
log004   sys  bes1          4   cnu---u 50beaaab   13f   1a4        --------
log005   sys  bes1          5   cnu---u 50beaaab   1a5   20b        --------
log006   sys  bes1          6   cn----u 50beaaab   20c   274        --------
```

Status 欄が–又は e のシステムログファイルが，アンロード済み状態のシステムログファイルです。この例の場合，log002 及び log003 のシステムログファイルが，アンロード待ち状態からアンロード済み状態に変わっています。log004 及び log005 のシステムログファイルは，ログポイント以降のシステムログファイル（データベースの回復時に必要となるシステムログファイル）のため，アンロード待ち状態のままとなります。

手順 2（ログポイント以降のシステムログファイルを使用しない場合）

1. pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(123456)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          7   cn---cu 50beaaab   275    0         --------
log002   sys  bes1          2   cru---u 50beaaab    72   d7         --------
log003   sys  bes1          3   cru---u 50beaaab    d8   13e        --------
log004   sys  bes1          4   cru---u 50beaaab   13f   1a4        --------
log005   sys  bes1          5   cru---u 50beaaab   1a5   20b        --------
log006   sys  bes1          6   cn----u 50beaaab   20c   274        --------
```

Status 欄が u 又は a のシステムログファイルが，アンロード待ち状態のシステムログファイルです。この例の場合，log002～log005 のシステムログファイルがアンロード待ち状態です。

2. pdlogchg コマンドを実行して，システムログファイルの状態を変更してください。

```
pdlogchg -d sys -s bes1 -g log002
pdlogchg -d sys -s bes1 -g log003
pdlogchg -d sys -s bes1 -g log004
pdlogchg -d sys -s bes1 -g log005
```

pdlogchg コマンドを実行すると，システムログファイルの状態が強制的にアンロード待ち状態からアンロード済み状態に変わります。

3. pdlogls コマンドを実行して，システムログファイルの状態を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(124021)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.         Ex-Status
log001   sys  bes1          7   cn---cu 50beaaab   275    0         --------
log002   sys  bes1          2   cr----u 50beaaab    72   d7         --------
log003   sys  bes1          3   cr----u 50beaaab    d8   13e        --------
log004   sys  bes1          4   cr----u 50beaaab   13f   1a4        --------
log005   sys  bes1          5   cr----u 50beaaab   1a5   20b        --------
log006   sys  bes1          6   cn----u 50beaaab   20c   274        --------
```

Status 欄が-又は e のシステムログファイルが，アンロード済み状態のシステムログファイルです。この例の場合，log002～log005 のシステムログファイルが，アンロード待ち状態からアンロード済み状態に変わっています。

20.17.6　ユニットの再開始時に必要となるレコード数の求め方

ユニットの再開始時に必要となるシステムログファイルのレコード数を確認してから，ユニットを再開始してください。システムログファイルのレコード数が不足している場合，ユニットを再開始できません。

(1)　必要となるレコード数を求める計算式

ユニットの再開始時に必要となるシステムログファイルのレコード数は，次に示す計算式から求められます。システムログファイルを二重化している場合は，A 系，B 系の両方で計算式から求めたレコード数が必要になります。

```
ユニットの再開始時に必要となるレコード数＝A×B
```

A：ユニットの再開始時に入力情報となるシステムログファイルのレコード数

B：次に示すどれかの値

- HiRDB Text Search Plug-in を使用している場合：3
- そのほかの場合：2

ユニットの再開始時に実行するデータベースの回復処理でもシステムログが出力されます。この量はデータベースを更新した SQL や HIRDB の運用によって増減します。出力システムログ量が増加してもシステムログファイル不足を解消できるように，必要となるレコード数を上記の計算式で見積もっています。

(2)　必要な容量の求め方の例

ユニットの再開始時に必要となるレコード数の求め方を例題形式で説明します。

例題

バックエンドサーバ bes1 でシステムログファイル不足が発生し，ユニットが異常終了しました。ユニットの再開始時に必要となるレコード数を求めます。

- HiRDB Text Search Plug-in は使用していません。

手順

1. KFPS01229-I メッセージの出力内容を確認してください。

```
KFPS01229-I HRDB UNT1 Next bes1 log file restart point, generation number=3, block
            number=15. restart end point, generation number=6, block number=3a2.
            last acquired syncpoint dump 2012/10/20 00:25:06
```

イベントログ又はメッセージログファイルを参照し，ユニットが異常終了したときに出力された KFPS01229-I メッセージの出力内容を確認します。KFPS01229-I メッセージには，ユニットの再開始時に入力情報となるシステムログファイルの情報が出力されます。この例の場合，世代番号 3～6 のシステムログファイルが，ユニットの再開始時の入力情報になります。

注意事項

次に示す場合，再開始しようとしたユニットが再度異常終了しているため，KFPS01229-I メッセージが二つ以上出力されています。

- pd_mode_conf オペランドに AUTO 又は MANUAL1 が指定している

- システムログファイルの不足によるユニットの異常終了後に pdstart コマンドを実行している

この場合，ユニットが最初に異常終了したときに出力された KFPS01229-I メッセージを確認してください。

2. pdlogls コマンドを実行して，システムログファイルの情報を確認してください。

```
pdlogls -d sys -s bes1

HOSTNAME : server1(204738)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log001   sys  bes1          1   cn----u 5089261b     1    1         --------
log002   sys  bes1          2   cn----u 5089261b     2    2         --------
log003   sys  bes1          3   cn----u 5089261b    11   d4         --------
log004   sys  bes1          4   cn----u 5089261b    d5  1a0         --------
log005   sys  bes1          5   cn----u 5089261b   1a1  2a1         --------
log006   sys  bes1          6   cn---cu 5089261b   2a2    0         --------
```

Gen No.欄にシステムログファイルの世代番号が表示されます。log003～log006 のシステムログファイルが，ユニットの再開始時の入力情報になります。

3. pdlogls コマンドを実行して，システムログファイルのレコード数を確認してください。

```
pdlogls -d sys -s bes1 -e

HOSTNAME : server1(204850)
***** Off-line Information *****
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log001   sys  bes1          1   cn----u 5089261b    1    1         --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log1     cn-u--        1  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log002   sys  bes1          2   cn----u 5089261b    2    2         --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log2     cn-u--        1  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log003   sys  bes1          3   cn----u 5089261b    11  d4         --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log3     cn-u--     4e16  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log004   sys  bes1          4   cn----u 5089261b    d5  1a0         --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log4     cn-u--     4df2  4e1c      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log005   sys  bes1          5   cn----u 5089261b   1a1  2a1         --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log5     cn-u--     6156  61a4      cs-u--  cs-u--    4096
Group    Type Server   Gen No.  Status  Run ID    Block No.          Ex-Status
log006   sys  bes1          6   cn---cu 5089261b   2a2    0         --------
         Element  Status   Record-count      Sys-A   Sys-B   Record-size
         log6     cn-u--        0  61a4      cc-u--  cc-u--    4096
```

log003～log006 の Record-count 欄を参照します。Record-count 欄の右側に，各システムログファイルのレコード数が 16 進数で表示されます。

16 進数のレコード数を 10 進数に変換して，log003～log006 の合計を求めてください。

システムログファイル名	レコード数	
	16 進数	10 進数
log003	4e1c	19996
log004	4e1c	19996
log005	61a4	24996
log006	61a4	24996
合計	－	89984

4. ユニットの再開始時に必要なレコード数を求めてください。

   ユニットの再開始時に必要なシステムログファイルのレコード数は，次のようになります。

   89984×2 ＝ 179968 レコード

   ユニットの再開始時に必要なシステムログファイルの容量は，次のようになります。

   179968×4096 ＝ 737148928 バイト

   上記の計算式の 4096 は，システムログファイルのレコード長です。3 の実行結果の Record-size 欄で確認できます。

   なお，上記の計算式で求めた値は，システムログファイルの各系の値です。システムログファイルを二重化している場合は，A 系，B 系の両方に上記の計算式で求めたレコード数と容量のシステムログファイルが必要になります。

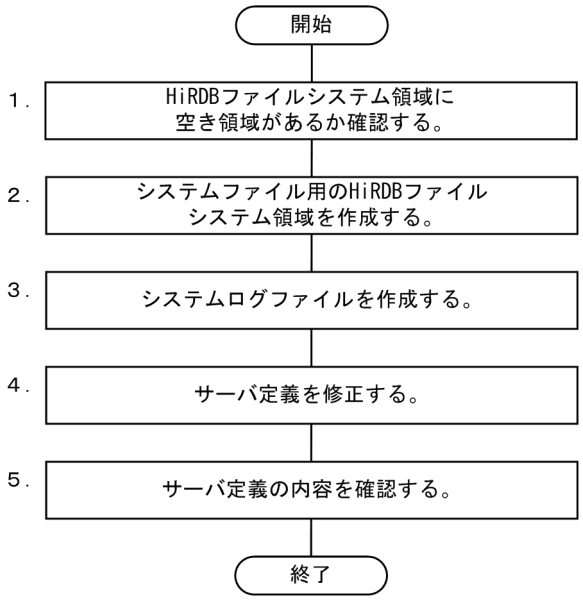
20.17.7　システムログファイルの追加方法

システムログファイル不足が発生したサーバに，システムログファイルを追加する手順を説明します。

注意事項

システムログファイルを追加する前に，HiRDB Datareplicator が稼働中かどうかを確認してください。稼働中の場合は HiRDB Datareplicator を終了し，システムログファイルを追加した後に開始してください。

図 20–7　システムログファイルの追加手順（フロー）



注　図中の項番は，以降の説明の（　）の番号と対応しています。

(1)　HiRDB ファイルシステム領域に空き領域があるか確認する

pdfstatfs コマンドを実行して，システムログファイル用の HiRDB ファイルシステム領域の空き領域を確認してください。

なお，システムログファイルを二重化している場合は，A 系，B 系それぞれの HiRDB ファイルシステム領域の空き領域を確認してください。

（例）

```
pdfstatfs -S /hirdb/system_dir/logfiles/SLOG6/logfile06

user area capacity              1023978[kB]
remain user area capacity       1023978[kB]
available file size             1023978[kB]
available file count                 32
current file count                    2
remain file count                    30   <-- ファイル作成可能数が1以上であることを確認
free area count                       0
available expand count                0
current expand count                  0
initialize area kind            SYS    <-- SYS（システムファイル用のHiRDBファイル
initialize user id              USER1      システム領域）であることを確認
initialize time                 Fri Oct 19 19:05:19 2012
area auto expand                NOUSE

***** HiRDB file system area space information *****
  offset[kB]     size[kB]  used/unused
           0       800000  used
      800000        23978  unused
                    ^
                    |
                    +-- HiRDBファイルシステム領域の空き領域のサイズ（単位：キロバイト）
```

この例の場合，HiRDB ファイルシステム領域の空き領域のサイズは，23978 キロバイトになります。

空き領域を確認した結果，システムログファイルを追加するための空き領域が十分にない場合は，次の手順に進んでください。空き領域が十分にある場合は，(3)の手順に進んでください。

(2) システムファイル用の HiRDB ファイルシステム領域を作成する

pdfmkfs コマンドを実行して，システムファイル用の HiRDB ファイルシステム領域を作成してください。

（例）

```
pdfmkfs  -n 1000 -l 32 -k SYS c:¥hirdb¥system_dir¥logfiles¥ADDLOGA
pdfmkfs  -n 1000 -l 32 -k SYS c:¥hirdb¥system_dir¥logfiles¥ADDLOGB
```

-n オプションには，HiRDB ファイルシステム領域のサイズ（単位：メガバイト）を指定します。ユニットの再開始時に必要となるシステムログファイルの容量以上のサイズを指定してください。

-k オプションには，SYS（システムファイル用の HiRDB ファイルシステム領域）を指定します。

この例では，A 系のシステムログファイルを格納する HiRDB ファイルシステム領域と，B 系のシステムログファイルを格納する HiRDB ファイルシステム領域の二つを作成しています。

HiRDB ファイルシステム領域を作成できる分の空き容量がディスクにない場合は，ディスクを追加して HiRDB ファイルシステム領域を作成してください。

(3) システムログファイルを作成する

pdloginit コマンドを実行してシステムログファイルを作成してください。

（例）

```
pdloginit -d sys -s bes1 -f c:¥hirdb¥system_dir¥logfiles¥ADDLOGA¥log008a -n 500000
pdloginit -d sys -s bes1 -f c:¥hirdb¥system_dir¥logfiles¥ADDLOGB¥log008b -n 500000
```

-n オプションには，システムログファイルのレコード数を指定します。ユニットの再開始時に必要となるレコード数以上の値を指定してください。

この例では，A 系及び B 系のシステムログファイルを作成しています。

(4) サーバ定義を修正する

(3)で作成したシステムログファイルに関する定義をサーバ定義に指定してください。サーバ定義の pdlogadfg 及び pdlogadpf オペランドを追加します。

（例）

```
pdlogadfg -d sys -g log008 ONL
pdlogadpf -d sys -g log008 -a c:¥hirdb¥system_dir¥logfiles¥ADDLOGA¥log008a
                           -b c:¥hirdb¥system_dir¥logfiles¥ADDLOGB¥log008b
```

システムログファイルを追加するサーバのサーバ定義を修正します。

(5)　サーバ定義の内容を確認する

pdconfchk コマンドを実行してサーバ定義の内容を確認してください。

（例）

```
pdconfchk

KFPS05007-I System definition check ended. return code = 0
```

pdconfchk コマンドのリターンコードが 0 のため，サーバ定義が正しく修正されています。

20.17.8　追加したシステムログファイルを削除する方法

ユニットを再開始するために追加したシステムログファイルが不要になった場合の，システムログファイルの削除方法を説明します。

注意事項

- システムログファイルがスワップ先にできる状態になったときに削除してください。スワップ先にできない状態のときにシステムログファイルを削除してしまうと，HiRDB を運用するために必要な情報が失われてしまうおそれがあります。
- アンロードレスシステムログ運用をしている場合は，データベースのバックアップ，及びログポイント情報ファイルを取得した後にシステムログファイルを削除してください。

手順を次に示します。

手順

1. pdstop -u コマンドを実行して，ユニットを正常終了してください。HiRDB/シングルサーバを使用している場合は，-u オプションは指定しないでください。
2. HiRDB Datareplicator を使用している場合は，HiRDB Datareplicator を終了してください。

   HiRDB Datareplicator を初期化する場合で，サーバに割り当てられているシステムログファイルを変更するときの運用については，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」の「抽出側システムの構成変更」を参照してください。
3. pdlogrm コマンドを実行して，システムログファイルを削除してください。
4. サーバ定義を修正してください。

   3 の操作で削除したシステムログファイルに関する定義（pdlogadfg 及び pdlogadpf オペランド）をサーバ定義から削除してください。

5. pdstart -u コマンドを実行して，ユニットを正常開始してください。HiRDB/シングルサーバを使用している場合は，-u オプションは指定しないでください。
6. 2 の操作で HiRDB Datareplicator を終了した場合は，HiRDB Datareplicator を開始してください。

20.17.9　抽出未完了状態のシステムログファイルがある場合の対処方法

抽出未完了状態のシステムログファイルがある場合の対処方法を次に示します。

- HiRDB Datarepricator によるシステムログの抽出中の場合
  抽出未完了状態のファイルが抽出完了状態になるのを待ちます。
- HiRDB Datarepricator の異常終了によってシステムログの抽出が中断している場合
  HiRDB Datarepricator を再起動してシステムログの抽出を再開してください。手順の詳細については，マニュアル「HiRDB データ連動機能 HiRDB Datareplicator」を参照してください。

ユニットが停止しているときに，抽出完了状態のシステムログファイルを確認する方法について説明します。手順を例題形式で説明します。

例題

HiRDB Datarepricator との連動中に，バックエンドサーバ（bes1）のシステムログファイルが不足してユニットが異常終了しました。bes1 の抽出未完了状態のシステムログファイルを確認します。

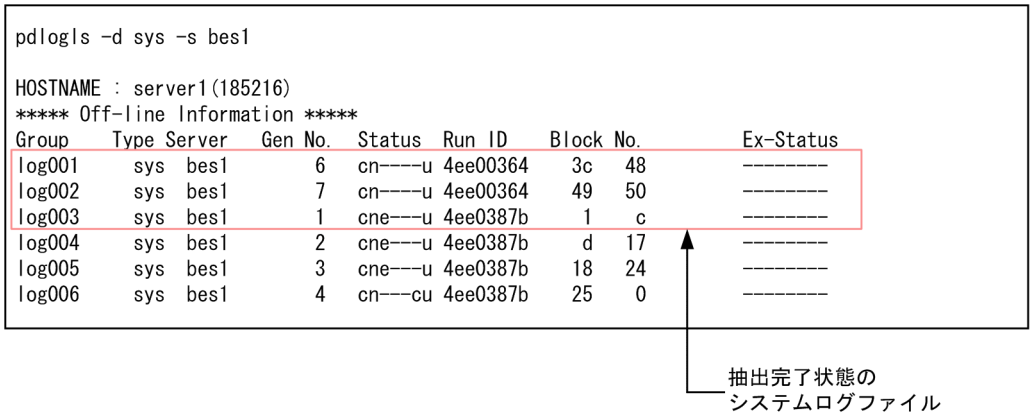
手順

1. pdls コマンドを実行して，抽出中のシステムログファイルを確認してください。

```
pdls -d rpl -s bes1 -j

SYSTEMID          : HRDB(185014)
Data replication  : *
UNITID            : UNT1(185014)
Data replication  : Y
SERVER NAME       : bes1
Extract Database  : Y
Extract Status    : C
System Log Extract Point :
Run ID   Group    Gen No.   BLock No.
4ee0387b log004   2         11          <- 抽出中のシステムログファイル名が表示されます。
System Log Sync Info     :
Run ID   Group    Gen No.   BLock No.
4ee0387b log005   3         1c          <- Run IDが同じシステムログファイルは抽出対象になります。
```

   この例の場合，log004 が抽出中のシステムログファイルになっています。
2. pdlogls コマンドを実行して，抽出未完了状態のシステムログファイルを確認してください。



pdls コマンドの実行結果と pdlogls コマンドの実行結果から抽出完了状態のシステムログファイルを特定します。1 で確認した抽出中のシステムログファイル log004 よりも世代番号の小さい log003 は抽出完了状態です。

log004 と Run ID が異なるシステムログファイルは，log004 の前に使用していたシステムログファイルなので，log001 及び log002 も抽出完了状態です。

20.17.10　システムログファイルの不足を解消した後に実施すること

システムログファイルの不足を解消した後，次に示すことを実施してください。

(1)　システムログファイルの容量を見直す

マニュアル「HiRDB システム導入・設計ガイド」の「システムログファイルの容量の見積もり」を参照して，システムログファイルの容量を見積もり直してください。

見積もりの結果，追加したシステムログファイルが不要な場合は，追加したシステムログファイルを削除してください。削除方法については，「追加したシステムログファイルを削除する方法」を参照してください。

(2)　システムログファイルの不足原因の対処を行う

システムログファイルが不足した原因が分かっている場合は，次の表で説明している見直し又は対処を実施してください。システムログファイルの容量以外の原因で再びシステムログファイルが不足しないようにしてください。

表 20–22　システムログファイルの不足原因と見直し又は対処の内容

項番	システムログファイルの不足原因	見直し又は対処の内容
1	自動ログアンロード機能が停止したため，アンロード待ち状態のファイルが増えて，システムログファイル不足が発生した。	「自動ログアンロード機能の運用方法」を参照して，自動ログアンロード機能の設定を見直してください。

項番	システムログファイルの不足原因	見直し又は対処の内容
2	システムログファイルをアンロード又は状態変更しなかったため，アンロード待ち状態のファイルが増えて，システムログファイル不足が発生した。	次に示すどちらかの対処をしてください。 • 自動ログアンロード機能の適用を検討してください。自動ログアンロード機能については，「自動ログアンロード機能の運用方法」を参照してください。 • システムログファイルの容量を見直してください。マニュアル「HiRDB システム導入・設計ガイド」の「システムログファイルの容量の見積もり」を参照してください。
3	長時間にわたりトランザクションが決着しなかったため，上書きできない状態のファイルが増えて，システムログファイル不足が発生した。	次に示すどちらかの対処をしてください。 • 問題となったトランザクションを見直してください。 • システムログファイルの容量を見直してください。マニュアル「HiRDB システム導入・設計ガイド」の「システムログファイルの容量の見積もり」を参照してください。
4	シンクポイントダンプの有効化処理がスキップされたため，上書きできない状態のファイルが増えて，システムログファイル不足が発生した。	「デファードライト処理適用時のシンクポイント処理時間のチューニング」を参照して，シンクポイント処理時間を見直してください。
5	システムログファイルに障害が発生したため，スワップ先にできる状態のファイルがなくなり，システムログファイル不足が発生した。	システムログファイルに障害が発生した原因を調査して対処してください。
6	HiRDB Datareplicator との連動中に，システムログの抽出処理が追い付かなくなり，抽出未完了状態のファイルが増えて，システムログファイル不足が発生した。	出力された HiRDB Datareplicator のエラーメッセージを参照し，システムログの抽出処理が追い付かなくなった原因を特定して対処してください。 目安として，抽出未完了状態のファイルが，常に全体の 3/4 以下になるようにしてください。

■システムログファイルの不足原因が分からない場合

システムログファイルの不足によってユニットが異常終了する前に，次に示すメッセージが出力されていないか確認してください。出力されたメッセージによって原因が特定できます。

- KFPS01150-E メッセージ

  自動ログアンロード機能が停止したときに出力されるメッセージです。上記の表の項番 1 の見直しをしてください。

- KFPS01202-E メッセージ

  システムログファイルに障害が発生したときに出力されるメッセージです。上記の表の項番 5 の見直しをしてください。

- KFPS02179-I メッセージ

  シンクポイントダンプの有効化処理がスキップされたときに出力されるメッセージです。KFPS02179-I メッセージが大量に出力されている場合は，KFPS02179-I メッセージに従った対処をしてください。

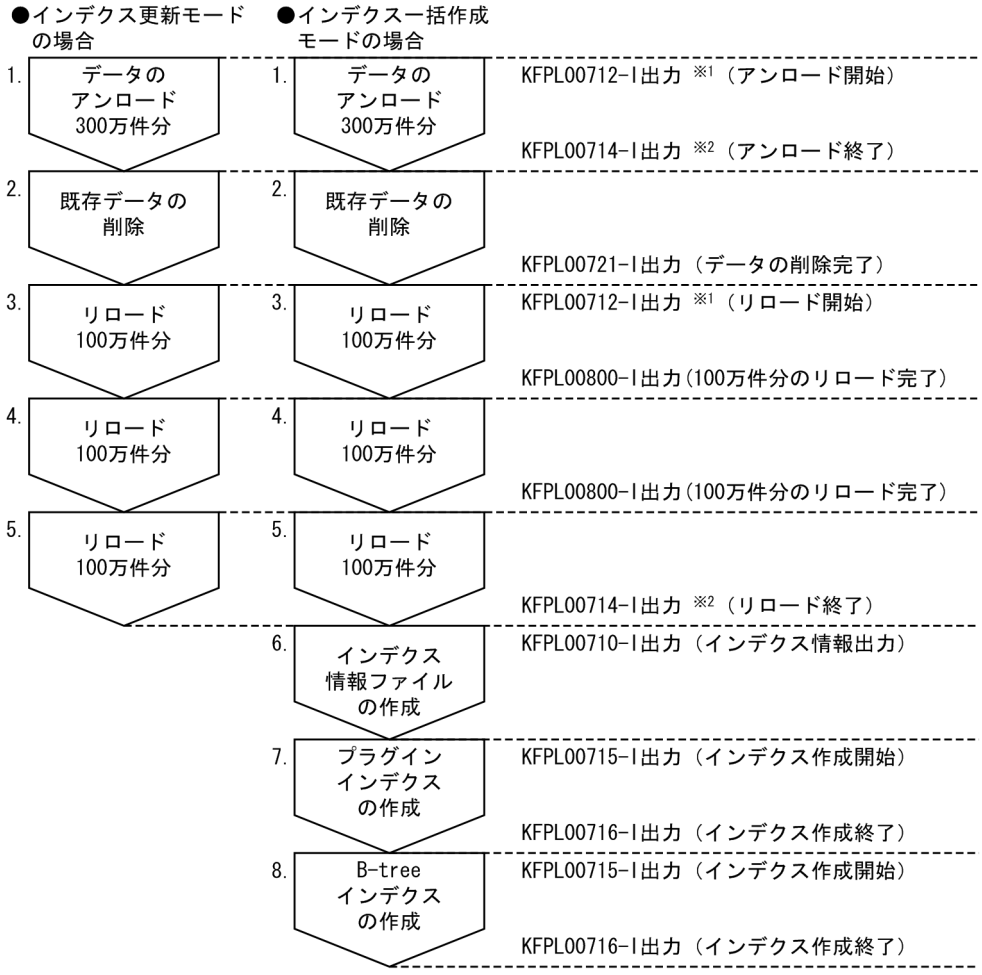
20.18　同期点指定の再編成実行中にユティリティが異常終了したときの対処方法

ここでは，**同期点指定の再編成**の実行中に，データベース再編成ユティリティが異常終了したときの対処方法を説明します。

20.18.1　対処方法の概要

異常終了したタイミング，及びデータベースの更新ログ取得方式によって対処方法が異なります。同期点指定の再編成実行中にユティリティが異常終了したときの対処方法を次の図に示します。

図 20–8　同期点指定の再編成実行中にユティリティが異常終了したときの対処方法



注　7. と8. の処理は入れ替わることがあります。
注※1 データベース再編成ユティリティ実行時に-gオプションを指定した場合は，KFPL00732-Iが出力されます。
注※2 データベース再編成ユティリティ実行時に-gオプションを指定した場合は，KFPL00733-Iが出力されます。

〔説明〕

総データ数を 300 万件，同期点行数を 100 万件としています。

●更新ログ取得方式が，ログ取得モード又は更新前ログ取得モードの場合

データのアンロード中（上の図中の 1 の時点）にユティリティが異常終了した場合は，データベース再編成ユティリティを再実行してください。

データリロード後にユティリティが異常終了した場合は，次の対処をしてください。

- 上の図中の 2〜5 の時点でユティリティが異常終了した場合は，データベース再編成ユティリティでリロード（-k reld）を実行してください。
- 上の図中の 6 の時点でユティリティが異常終了した場合は，表データのリロード処理は完了しているため，インデクス情報ファイルを作成します。データベース再編成ユティリティでインデクスを再作成（-k ixrc）してください。
- 上の図中の 7，8 の時点でユティリティが異常終了した場合は，インデクスだけを作成します。作成されたインデクス情報ファイルを入力情報にして，データベース再編成ユティリティでインデクスを一括作成（-k ixmk）してください。

●更新ログ取得方式が，ログレスモードの場合

エラー時のデータベースの状態とその回復方法については，マニュアル「HiRDB コマンドリファレンス」の「データベース再編成ユティリティ（pdrorg）」の「エラー時のデータベースの状態とその回復方法」を参照してください。

20.18.2　例題

300 万件のデータを格納した表の再編成中にデータベース再編成ユティリティが異常終了しました。同期点行数は 100 万件とします。

(1)　メッセージを確認します

次に示すメッセージが出力されています。

```
KFPL00714-I  unload ended, table=USR01.TABLE1, server=sds01, return code=0

KFPL00800-I Loading until 2000000th row committed
```

〔説明〕

- KFPL00714-I メッセージから，アンロード処理が完了していることが分かります。
- KFPL00800-I メッセージから，200 万件までリロード処理が完了していることが分かります。

(2)　pdrorg コマンドでリロード処理を実行します

データベース再編成ユティリティで，残り 100 万件のリロード処理とインデクス作成処理を実行します。

```
pdrorg -k reld -t TABLE1 C:¥pdrorg¥rorg01
```

〔説明〕

-k オプションの指定だけを変更してください。rorg から reld に変更します。そのほかのオプションの指定を変更する必要はありません。

(3)　メッセージを確認します

データベース再編成ユティリティが正常終了したかどうかを確認します。KFPL00719-I メッセージを確認してください。

```
KFPL00719-I  Pdrorg terminated, return code=0
```

(4)　リロード対象 RD エリアのバックアップを取得します

更新前ログ取得モード（省略値）でリロードを実行したので，リロード対象 RD エリアのバックアップを取得してください。バックアップの取得方法については，「例題８（RD エリア単位にバックアップを取得する場合）」を参照してください。

(5)　pdrels コマンドで RD エリアの閉塞を解除します

pdrels コマンドで，表及びインデクス格納 RD エリアの閉塞を解除します。

```
pdrels -r RDAREA1,RDAREA2,RDAREA3
```

コマンドの実行後，実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

20.18.3　アンロードデータファイルを一元化しないときに異常終了した場合の対処方法（HiRDB/パラレルサーバ限定）

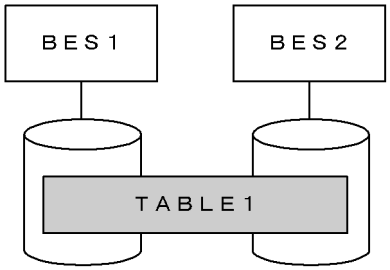
表を複数のバックエンドサーバに横分割している場合，アンロードデータファイルを一元化（データベース再編成ユティリティで-g オプションを指定）することをお勧めしました。アンロードデータファイルを一元化しないと，ユティリティが異常終了した場合に対処方法が複雑になります。

ここでは，何らかの事情によってアンロードデータファイルを一元化しないで，同期点指定の再編成又は同期点指定のリロード中にユティリティが異常終了したときの対処方法を説明します。

(1)　対処方法の概要

アンロードデータファイルを一元化しない場合，サーバごとに再編成処理が完了したかどうかを確認する必要があります。次の図に示すシステム構成の場合を例にして説明します。

図 20–9　表を複数のバックエンドサーバに横分割している場合の例



〔説明〕

アンロードデータファイルを一元化していない場合，BES1 の再編成処理がアンロード中に，BES2 の再編成処理がリロード中に異常終了することがあります。この場合，ユティリティの制御情報ファイルの指定を変更する必要があります。また，ユティリティもサーバごとに再実行する必要があります。この例だと 2 回実行することになります。

このように，異常終了したタイミングを考慮して，サーバごとにデータベース再編成ユティリティを再実行する必要があります。

(2)　例題

アンロードデータファイルを一元化しないで，同期点指定の再編成をしました。システム構成は図「表を複数のバックエンドサーバに横分割している場合の例」のとおりとします。BES1 の再編成処理はアンロード中に，BES2 の再編成処理はリロード中に異常終了したとします。

(a)　異常終了したときのデータベース再編成ユティリティの指定

異常終了したときのデータベース再編成ユティリティの制御情報ファイル及びコマンド指定を次に示します。

●制御情報ファイル（C:¥pdrorg¥rorg01）の指定

```
option job=j01,200
unload BES1:C:¥unload¥file01
unload BES2:C:¥unload¥file02
```

●pdrorg コマンドの指定

```
pdrorg -k rorg -t TABLE1 C:¥pdrorg¥rorg01
```

(b)　再実行するときのデータベース再編成ユティリティの指定

BES1 の表に対しては，再編成処理を再度実行する必要があります。BES2 の表に対しては，リロード処理を実行する必要があります。したがって，BES1，BES2 でそれぞれデータベース再編成ユティリティを実行します。

再実行するときのデータベース再編成ユティリティの制御情報ファイル及びコマンド指定を次に示します。

●制御情報ファイル（C:¥pdrorg¥rorg01）の指定

```
option job=j01,200
unload BES1:C:¥unload¥file01
```

●制御情報ファイル（C:¥pdrorg¥rorg02）の指定

```
option job=j01,200
unload BES2:C:¥unload¥file02
```

●pdrorg コマンドの指定

```
pdrorg -k rorg -t TABLE1 C:¥pdrorg¥rorg01
pdrorg -k reld -t TABLE1 C:¥pdrorg¥rorg02
```

20.18.4　注意事項

1. データベース再編成ユティリティを再実行する前に RD エリアの再初期化や PURGE TABLE 文を実行すると，設定されていた同期点情報がなくなります。この場合，再度実行しても同期点指定の再実行とは扱われません。
2. LOB 列構成基表に対する同期点指定の再編成実行中にユティリティが異常終了した場合，ユティリティの再実行時に CLR を指定してください。CLR を指定しないと，再編成したのに LOB 列の検索性能が低下することがあります。
3. 次に示すファイル名称を明示していない場合，データベース再編成ユティリティがファイルを自動的に作成します。
   - インデクス情報ファイル
   - エラー情報ファイル

   データベース再編成ユティリティが異常終了すると，これらのファイルがディスク上に残ります。再実行時，データベース再編成ユティリティは新しい名称でファイルを作成するため，異常終了時のファイルが残ったままディスクを圧迫する原因になります。このため，再実行前にこれらのファイルを削除してください。
4. データベース再編成ユティリティ実行時（-k rorg）に異常終了した場合，異常終了した状態を表のステータスとして設定します。この状態で，データベース再編成ユティリティを再実行（-k rorg）すると，この表のステータスを基に異常終了した続きから処理を再開始し，処理完了時に表のステータスをクリアします。

   インデクス作成中に再編成ユティリティが異常終了した後，インデクスを一括作成（-k ixmk）した場合は，この表のステータスをクリアしません。このため，インデクス一括作成でインデクスをロードした場合は，option 制御文に tblstatus オペランドを指定した再編成（-k rorg）を実行して，表のステータスを変更してください。この操作を実行しないと，表のステータスがインデクス作成中にユティリティが異常終了した状態のままとなります。

20.19　RD エリアのページ破壊を検知したときの対処方法

実行者　HiRDB 管理者

ここでは，RD エリアの**ページ破壊**が発生して（KFPH00308-E メッセージ出力），RD エリアが**障害閉塞**したとき（KFPH00306-E メッセージ出力）の対処方法について説明します。

20.19.1　ページ破壊が発生する要因

ページ破壊が発生する要因を次に示します。

- HiRDB の異常終了又は強制終了後に HiRDB を強制開始した
- HiRDB の異常終了又は強制終了後に，ステータスファイルを初期化して HiRDB を開始した
- RD エリアをバックアップ取得時点に回復したときに，関連するすべての RD エリアを回復しなかった（例えば，表格納 RD エリアをバックアップ取得時点に回復したが，インデクス格納 RD エリアを回復し忘れた場合が該当します）

20.19.2　対処方法

対処手順を次に示します。

〈手順〉

1. %PDDIR%¥spool 下のトラブルシュート情報のバックアップを取得します。
2. 障害発生以前の全イベントログの内容のバックアップを取得します。
3. **pdclose** コマンドで障害閉塞した RD エリア及び関連する RD エリア（関連する表及びインデクス格納 RD エリア）をクローズします。
4. **pdfbkup** コマンドで障害閉塞した RD エリア及び関連する RD エリアを構成するファイルのバックアップを取得します。
5. 以上の作業をした後に RD エリアの回復作業を行います。
   RD エリアの回復方法には次に示す二つの方法があります。

(1)　回復方法（その１）

システム単位のバックアップがある場合は，システム全体をバックアップ取得時点に回復してください。

(2) 回復方法（その２）

障害閉塞した RD エリアをデータベース構成変更ユティリティ（pdmod コマンド）で再初期化してください。ただし，この方法だと再初期化した RD エリア内のデータは削除されます。アンロードデータファイル又は入力データファイルなどからデータを回復してください。

また，再初期化した RD エリアに関連する RD エリアを論理的に回復する必要があります。

なお，HiRDB を再開始しなかった場合は，ほかの RD エリアにも同様の障害が発生している可能性があるので注意してください。

20.20　RD エリアの入出力エラー（パス障害）が発生したときの対処方法

実行者　HiRDB 管理者

ここでは，RD エリアの入出力エラーが発生したときの対処方法について説明します。なお，ここでいう入出力エラーとは，HiRDB がファイルを操作したときに HiRDB が判別できない理由でファイル操作に失敗したエラーを意味し，HiRDB ファイルシステムに対するアクセス要求から返されるエラーコードに-1544 が出力されるエラーのことです。

20.20.1　RD エリアの入出力エラーが発生したときの HiRDB の処理

RD エリアの入出力エラーが発生したときの HiRDB の処理を次の表に示します。

表 20–23　RD エリアの入出力エラーが発生したときの HiRDB の処理

RD エリアの種類	HiRDB の処理	
	pd_db_io_error_action ＝ dbhold（省略値）	pd_db_io_error_action ＝ unitdown
マスタディレクトリ用 RD エリア	HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。	
それ以外の RD エリア	RD エリアを障害閉塞します。	HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。RD エリアを障害閉塞しません。ただし，異常終了後に再度入出力エラーが発生した場合，その RD エリアを障害閉塞します。

20.20.2　pd_db_io_error_action ＝ unitdown（HiRDB の異常終了）を選択するときは

RD エリアの入出力エラーが発生したときに HiRDB を異常終了させると，次に示すような利点があります。

- パス障害の場合は系を切り替えると業務を続行できます。
- RD エリアの回復作業が必要なくなることがあります。

(1)　パス障害の場合は系を切り替えると業務を続行できます

RD エリアの入出力エラーが発生したときに HiRDB が異常終了するため，系を切り替えられます。入出力エラーの要因が**パス障害**の場合は，系の切り替え後に入出力処理ができるようになるため，業務をそのまま続行できます。なお，ここでいうパス障害とは，HiRDB とファイル間の通信経路が何かの要因で切断されているため，ファイルをアクセスできない状態を意味しています。

(2)　RD エリアの回復作業が必要なくなることがあります

RD エリアの入出力エラーが発生したときに RD エリアを障害閉塞しないで HiRDB を異常終了します。したがって，HiRDB を再開始する前に入出力エラーの対策ができます。対策後，HiRDB を再開始すると処理を続行できます。この場合，RD エリアが障害閉塞しないため，RD エリアの回復作業が必要なくなります。ただし，ディスク障害などでディスクを交換した場合などは，データベース回復ユティリティで RD エリアを回復する必要があります。

20.20.3　環境設定

RD エリアの入出力エラーが発生したときに HiRDB を異常終了させるには，次に示す環境設定が必要です。

(1)　pd_db_io_error_action オペランドの指定

pd_db_io_error_action オペランドに unitdown を指定してください。

(2)　系切り替え機能を使用する場合

系切り替え機能を使用する場合は，「RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える方法」を参照してください。

(3)　pd_mode_conf オペランドの指定（系切り替え機能を使用しない場合限定）

HiRDB の異常終了後に HiRDB が自動的に再開始しないように，pd_mode_conf オペランドに MANUAL2 を指定してください。MANUAL2 以外を指定すると，HiRDB の異常終了後に HiRDB が自動的に再開始します。この場合，同じ RD エリアで入出力エラーが発生し，その RD エリアを障害閉塞状態にします。したがって，pd_db_io_error_action ＝ unitdown を指定した意味がなくなります。

(4)　ほかの機能との関連

OS 又はデバイス・ドライバの機能※で，物理ボリューム又は論理ボリュームの入出力エラーを検知するまでの時間を指定する場合は，「UAP 又はユティリティの実行時間の監視（無応答障害時の影響を抑える方法）」に記載しているオペランド又はオプションの指定に注意が必要です。

OS 又はデバイス・ドライバの機能で指定した時間がこれらのオペランド又はオプションに指定した時間より長い場合，入出力エラーを検知する前に監視対象が強制終了します。したがって，これらのオペランド又はオプションで指定する監視時間を長くするなどの対応が必要になります。

注※

機能の詳細については，OS やデバイス・ドライバのマニュアルを参照してください。

20.20.4　運用方法

pd_db_io_error_action ＝ unitdown を指定したときの運用方法を説明します。

(1)　系切り替え機能を使用する場合の運用方法

系切り替え機能を使用する場合は，「RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える方法」を参照してください。

(2)　系切り替え機能を使用しない場合の運用方法

入出力エラーが発生して HiRDB が異常終了した場合，出力されたメッセージを調査して対策してください。対策後，HiRDB を再開始してください。なお，再開始後も入出力エラーが発生すると RD エリアは障害閉塞します。この場合，データベース回復ユティリティで RD エリアを回復してください。

(3)　HiRDB の異常終了後にすること（共通）

入出力エラーが発生して HiRDB が異常終了した場合，それ以降は pd_db_io_error_action ＝ unitdown の指定が無効になります（RD エリアが障害閉塞します）。再び，pd_db_io_error_action ＝ unitdown の指定を有効にするには，次に示すどちらかの方法を実行してください。

- HiRDB を正常開始する
- HiRDB システム定義ファイルを%PDDIR%¥conf¥chgconf 下にコピーして，システム構成変更コマンド（pdchgconf コマンド）を実行する

20.20.5　注意事項

- pd_db_io_error_action ＝ unitdown を指定したときに入出力エラーが発生すると HiRDB が異常終了するため，更新前ログ取得モード又はログレスモードで UAP やユティリティを実行していると，処理対象の RD エリアが障害閉塞することがあります。
- 開始処理中又は終了処理中の入出力エラーについては，pd_db_io_error_action ＝ unitdown を指定しても HiRDB を異常終了しません。

20.21　コミット処理中に障害が発生したときのトランザクション決着種別の確認方法（HiRDB/パラレルサーバ限定）

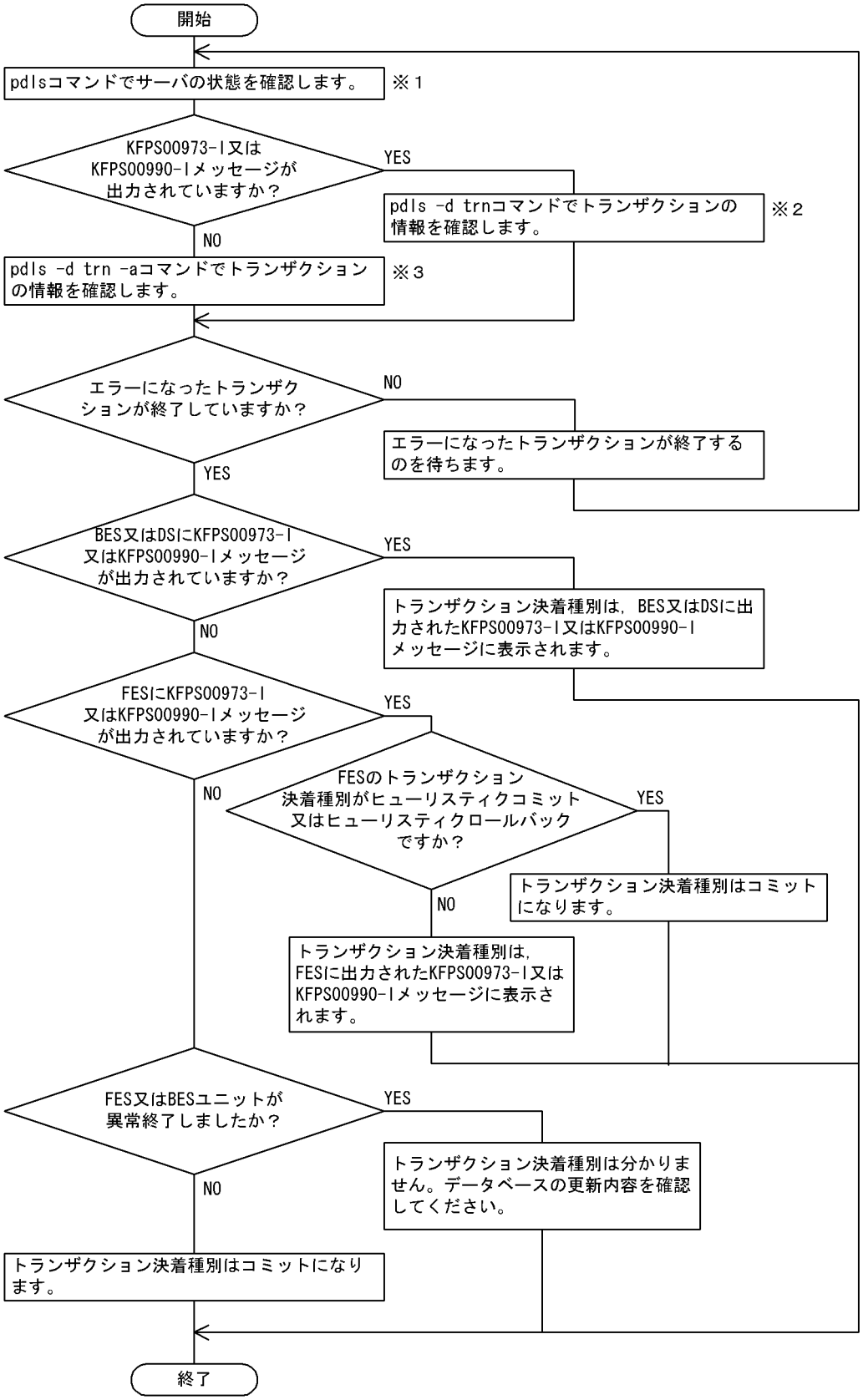
実行者　HiRDB 管理者

ここでは，コミット処理中に障害が発生したときのトランザクション決着種別の確認方法について説明します。

トランザクションの実行中に障害が発生した場合，サーバごとにコミット処理又はロールバック処理を行います。処理の完了後，サーバごとにトランザクション情報とトランザクション決着種別を **KFPS00990-I** メッセージに付加して出力します。また，障害が発生して独自にロールバック決着した場合は，更新ブランチがあったサーバのトランザクション情報とトランザクション決着種別を **KFPS00973-I** メッセージに付加して出力します。

障害が発生したときのトランザクション決着種別の確認方法を次の図に示します。UAP にコミットエラーが返信された場合もこの手順に従って確認してください。

図 20–10　障害が発生したときのトランザクション決着種別の確認方法



注※1

pdls コマンドで，全サーバ（回復不要 FES を除く全サーバ）が稼働しているかどうかを確認してください。

注※2

次に示す情報を比べてください。

- pdls -d trn コマンドの実行結果に出力されたトランザクション識別子
- KFPS00973-I 又は KFPS00990-I メッセージに出力されたトランザクション識別子

これらのトランザクション識別子が一致していないことを確認してください。一致しているトランザクション識別子がある場合は，そのトランザクションが終了するまで待ってください。

注※3

次に示す情報を比べてください。

- pdls -d trn -a コマンドの実行結果に出力されたクライアントのプロセス ID，IP アドレス，UAP の識別名称
- コミットエラーになった UAP のプロセス ID，IP アドレス，UAP の識別名称

これらの情報が一致するトランザクションがあるかどうかを確認してください。

すべて（三つ）の情報が一致するトランザクションがある場合

該当するトランザクションが終了するまで待ってください。

UAP の識別名称だけが一致し，プロセス ID 及び IP アドレスが表示されていないトランザクションがある場合

次に示す手順で対処してください。

1. 該当するトランザクションのトランザクション識別子を確認してください。
2. pdls -d trn -a コマンドの実行結果を参照してください。1 で確認したトランザクション識別子がほかにも表示されていないかを確認してください。
3. ほかにも表示されている場合は，そのトランザクションのクライアントのプロセス ID と IP アドレスが表示されているかどうかを確認してください。表示されたトランザクションがない場合は，トランザクションが終了するまで待ってください。

コミット時の障害によるトランザクションの決着種別を次の表に示します。なお，表中のトランザクション決着種別は，フロントエンドサーバ又はバックエンドサーバのユニットが異常終了しなかった場合に限ります。

表 20–24　コミット時の障害によるトランザクションの決着種別

項番	障害のタイミング	フロントエンドサーバ		バックエンドサーバ又はディクショナリサーバ		トランザクション決着種別
		KFPS00973-I，又はKPFS00990-I メッセージ	メッセージに出力される決着種別	KFPS00973-I，又はKPFS00990-I メッセージ	メッセージに出力される決着種別	
1	コミット処理時	あり	コミット	あり	コミット	コミット
2					ロールバック	ロールバック
3				なし	–	コミット
4			ロールバック	あり	コミット	コミット
5					ロールバック	ロールバック
6				なし	–	ロールバック
7		なし	–	あり	コミット	コミット
8					ロールバック	ロールバック
9				なし	–	コミット
10	トランザクション決着前	あり	ロールバック	あり	ロールバック	ロールバック
11				なし	–	ロールバック
12		なし		あり	ロールバック	ロールバック
13				なし	–	–

（凡例）–：該当しません。

20.22　ローカルバッファを使用して共用表を更新した場合に障害が発生したときの対処方法（HiRDB/パラレルサーバ限定）

実行者　HiRDB 管理者

ローカルバッファを使用して共用表を更新する場合は，LOCK TABLE を使用してください。次に示す条件をすべて満たす場合にサーバプロセスが異常終了すると，アボートコード Phb3008 が出力されて，ユニットが異常終了することがあります。

- ローカルバッファを使用している
- LOCK TABLE を使用しないで共用表を更新している

これらの条件をすべて満たす場合，サーバプロセスの異常終了時に更新ページがあると，トランザクション回復プロセスで回復処理ができないことがあります。その場合，ユニットの再開始時に回復処理を行います。ローカルバッファを使用して共用表を更新（LOCK TABLE 指定なし）する場合に障害が発生したときの HiRDB の処理と対処方法を次の表に示します。

表 20–25　ローカルバッファを使用して共用表を更新（LOCK TABLE 指定なし）する場合に障害が発生したときの HiRDB の処理と対処方法

<table>
<tr><th colspan="2">サーバプロセスの異常終了要因</th><th>サーバプロセスの異常終了時に更新ページがある場合の HiRDB の処理</th><th>HiRDB 管理者の処置</th></tr>
<tr><td colspan="2">PDSWAITTIME オーバ</td><td rowspan="5">アボートコード Phb3008 が出力されて，ユニットが異常終了することがあります。</td><td rowspan="5">ユニットが異常終了した場合は，ユニットを再開始してください。</td></tr>
<tr><td colspan="2">PDCWAITTIME オーバ</td></tr>
<tr><td colspan="2">pdcancel コマンドの実行</td></tr>
<tr><td>アボート</td><td>HiRDB の矛盾検知によるアボートなどが該当します。</td></tr>
<tr><td>上記以外</td><td>SIGSEGV，SIGBUS，外部からのシグナル受信，exit，そのほかの予期しない障害などが該当します。</td></tr>
</table>

注

PDSWATCHTIME オーバによるサーバプロセスの異常終了時に更新ページがあっても HiRDB は異常終了しません。

20.23　システムマネジャユニットに障害が発生したときの対処方法

障害が発生してシステムマネジャユニットが停止中の場合，HiRDB サーバにユーザが接続できなくなります。システムマネジャユニットの障害がすぐに対処できない場合，クライアント環境定義の PDHOST オペランドの指定値を変更して対処してください。PDHOST オペランドの指定値を次に示すように変更すると，HiRDB サーバに接続できるようになります。

- **クライアント環境定義の PDFESHOST オペランドを指定している場合**
  PDFESHOST オペランドに指定しているホスト名（FQDN 又は IP アドレス）を PDHOST オペランドにも指定してください。
- **クライアント環境定義の PDFESHOST オペランドを指定していない場合**
  PDFESHOST オペランドを指定し，同じホスト名（FQDN 又は IP アドレス）を PDHOST オペランドにも指定してください。

クライアント環境定義の PDHOST 及び PDFESHOST オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

20.24　ミラー正副不一致状態が発生したときの対処方法

20.24.1　ミラー正副不一致状態を解消する方法

LVM のミラーリング機能を使用してデータベースをミラー化している場合，データベースへの書き出し中に OS 又はサーバマシンに障害が発生したり，系切り替えしたりすると，正系ボリュームと副系ボリュームの内容が異なる状態になることがあります。これを**ミラー正副不一致状態**といいます。この場合，次に示すどちらかの方法でミラー正副不一致状態を解消してください。

1. LVM の機能を使用してミラー正副不一致状態を解消する方法
2. HiRDB が再開始時に行う全面回復処理でミラー正副不一致状態を解消する方法

通常は 1 の方法で対処してください。LVM の機能を使用して正副ボリュームを一致させられない場合や，正副ボリュームを一致させる処理に時間が掛かるため，系の切り替え時間などのシステム要件を満たさない場合に 2 の方法で対処してください。

(1)　LVM の機能を使用してミラー正副不一致状態を解消する方法

準備

pd_redo_allpage_put オペランドに N を指定するか，又はこのオペランドを省略してください。

ミラー正副不一致状態が発生したときの対処方法

HiRDB を再開始する前に，LVM の機能を使用して正副ボリュームを一致させてください。その後，HiRDB を再開始してください。

(2)　HiRDB が再開始時に行う全面回復処理でミラー正副不一致状態を解消する方法

準備

pd_redo_allpage_put オペランドに Y を指定してください。Y を指定すると，HiRDB の再開始時の全面回復処理でシンクポイント以降に更新されたすべてのページをデータベースに書き出します。この処理によってミラー正副不一致状態を解消します。

ミラー正副不一致状態が発生したときの対処方法

LVM の機能を使用して正副ボリュームを一致させる必要はありません。すぐに HiRDB を再開始してください。HiRDB の再開始時の全面回復処理で正副ボリュームを一致させます。

注意事項

- この方法を取ると，全面回復処理でデータベースへの書き出し量が増加します。そのため，1 の方法と比較して HiRDB の再開始に掛かる時間が長くなります。データベースへの書き出

し量を減らすために，シンクポイントの取得間隔を短くすることをお勧めします。シンクポイントの取得間隔は，pd_log_sdinterval オペランドで設定します。

- なお，この方法では，障害で OS やサーバがダウンしたり，系切り替えが発生した際にデータベースへ書き出し中だったデータを，全面回復処理で再度書き出します。これを受けて，LVM が正系ボリュームと副系ボリュームの両方に書き出すことで正副ボリュームを一致させます。そのため，HiRDB がダウンする瞬間に発生するミラー正副不一致状態以外の，運用上の障害などで発生した正副ボリュームの不一致は解消できません。この場合は，LVM の機能を使用して正副ボリュームを一致させてください。

20.25　HiRDB ファイルシステム領域に障害が発生した場合の対処方法

実行者　HiRDB 管理者

HiRDB ファイルの作成，削除，又は拡張の処理中に，OS の kill コマンド，シャットダウンなどによって HiRDB の処理が停止した場合，HiRDB ファイルシステム領域の管理情報（HiRDB ファイルシステム構成やファイルを管理する情報）が更新途中のままとなります。

この場合，管理できない状態のファイルや，参照できない領域ができてしまうことがあります。HiRDB の処理続行には問題はありませんが，HiRDB ファイルシステム領域作成時の最大作成ファイル数（pdfmkfs コマンドの-l オプション値）までファイルを作成できなかったり，設定容量（pdfmkfs コマンドの-n オプション値から領域管理部のサイズを引いた値）まで利用できなかったりします。また，バージョン 07-02 以前の HiRDB の場合は，領域管理情報が破壊されてしまうこともあります。

ここでは，管理できない状態のファイル，参照できない領域，及び領域管理情報の破壊が発生した場合の対処方法について説明します。

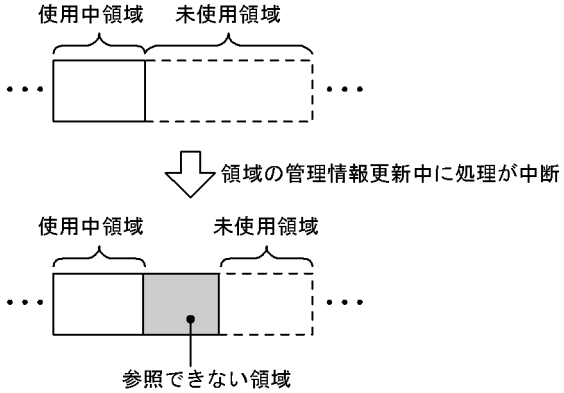
20.25.1　管理できない状態のファイル，及び参照できない領域

(1)　管理できない状態のファイルとは

HiRDB ファイルシステム領域の管理情報は，最大作成ファイル数分のファイル管理情報を，「使用中」と「未使用」の 2 とおりで管理しています。ファイルの作成，及び削除処理で，領域の管理情報更新中に処理が中断すると，処理対象のファイルが参照されない状態となり，それ以降使用できなくなることがあります。この場合，使用できるファイル数は最大作成ファイル数より少なくなります。また，ファイルに割り当てられた領域についても，以降参照できなくなることもあり，本来より少ない容量しか使用できなくなります。同様の処理中断が連続して発生すると，更に使用できるファイル，及び容量は減少します。

(2)　参照できない領域とは

HiRDB ファイルシステム領域の管理情報は，ファイル管理情報とは別に，領域をファイルに割り当てている使用中領域と，割り当てられていない未使用領域の 2 とおりで管理しています。ファイルの拡張処理（自動増分）では，ファイルの増分に必要な領域を未使用領域から取り出し，使用中領域として割り当てます。ファイルの拡張処理で，領域の管理情報更新中に処理が中断すると，処理対象の領域が使用中と未使用どちらからも参照されない状態となり，以降使用できなくなることがあります。参照できない領域の概念を次に示します。



上記の図は，ファイルの拡張処理で使用中領域として割り当てる前に処理が中断した状態を表しています。網掛け部分の領域が以降参照できなくなるため，本来より少ない容量しか使用できなくなります。同様の処理中断が連続して発生すると，更に使用できる容量は減少します。

(3) 確認方法

管理できない状態のファイル，及び参照できない領域があるかどうかを確認する方法を次に示します。

- pdfls コマンドを実行して，KFPI21586-W メッセージが表示された場合，管理できない状態のファイルがあります。
- HiRDB ファイルシステム領域作成時の容量（pdfmkfs コマンドの-n オプション値から領域管理部のサイズを引いた値）よりも，少ない容量しか使用できない場合，参照できない領域があります。
- pdffsck -c コマンドを実行して処理結果を確認します。

(4) 対処方法

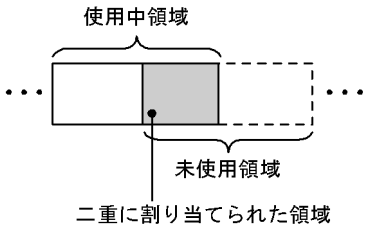
pdffsck コマンドを実行して修復をしてください。

-c オプションありで実行すると，検証だけ行われて修復はされません。管理できない状態のファイル，又は参照できない領域が，あらかじめ対象となる HiRDB ファイルシステム領域内にあることが分かっている場合，-c オプションなしで実行してください。

20.25.2 領域管理情報の破壊（バージョン 07-02 以前の HiRDB の場合）

(1) 領域管理情報の破壊とは

バージョン 07-02 以前の HiRDB の場合，ファイルの作成，削除，又は拡張で処理が中断すると，管理できない状態のファイル，及び参照できない領域のどちらにも該当しない状態になることがあります。これは，領域の二重割り当てなどによって，領域管理情報とファイルが破壊された状態であり，以降の運用に影響を与えます。二重割り当ての概念を次に示します。



領域管理情報が破壊されているため，破壊された対象ファイルを回復しても，更にほかのファイルも破壊してしまいます。この状態からバージョンアップすると，領域が破壊された状態も引き継ぐことになります。

注意事項

- 領域管理情報が破壊された場合，pdfbkup，及び pdfrstr コマンドでの回復，並びに pdcopy，及び pdrstr コマンドでの回復では対処できません。
- pdffsck コマンドで領域管理情報の破壊は検知できますが，修復はできません。この場合，KFPI21585-E メッセージを出力して異常終了します。

(2)　対処方法

すべての HiRDB ファイルシステム領域を再作成してください。その後，HiRDB ファイルシステム領域の使用目的（pdfmkfs コマンドの-k オプションの指定値）ごとに，次の対処をしてください。

(a)　DB 及び SDB の場合

バックアップを使用してデータベースを回復してください。なお，使用するバックアップは，領域管理情報が破壊される前の HiRDB ファイルシステム領域に格納されている RD エリアのものです。

(b)　SYS の場合

システムログファイル，シンクポイントダンプファイル，及びステータスファイルを作成してください。なお，再開始に必要なシステムファイルをそろえられないため，再開始はできません。正常開始してください。

また，セキュリティ監査機能を使用している場合は，監査証跡ファイルを作成してから，HiRDB 及びセキュリティ監査機能の運用を再開してください。

(c)　WORK の場合

作業表は，SQL 実行時に HiRDB が自動的に作成するため，特に対処する必要はありません。

(d)　UTL の場合

pdcopy コマンドのバックアップファイル，及び pdrorg コマンドのアンロードデータファイルはなくなっているため，必要に応じて pdcopy，又は pdrorg コマンドを再度実行してください。

21 データベースの回復方法

この章では，障害が発生したためにデータベース（RD エリア）が破壊されたときのデータベースの回復方法について説明します。

21.1　データベース回復の概要

実行者　HiRDB 管理者

データベースに障害が発生した場合，HiRDB 管理者はデータベース回復ユティリティ（**pdrstr** コマンド）でデータベースを回復します。ここでは，データベースを回復するときに知っておいて欲しい基本的なことについて説明します。ここで説明する項目は次のとおりです。

- データベースをどの状態に回復できるのか？
- バックアップ取得モードとの関連
- ログ取得モードとの関連
- 各種 RD エリア回復時の注意
- 同一ログの再使用チェック
- 64 ビットモードの HiRDB を使用している方へ
- テープ装置アクセス機能

21.1.1　データベースをどの状態に回復できるのか？

データベースは次に示すどれかの時点の状態に回復できます。

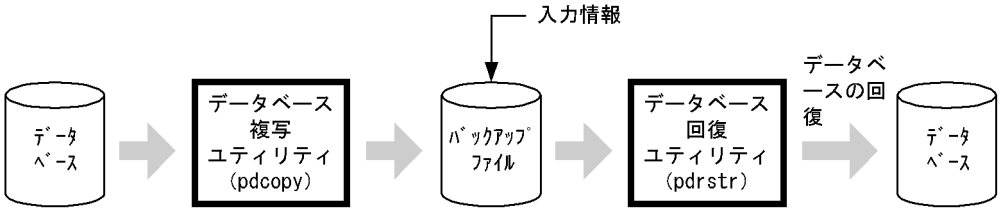
- バックアップ取得時点
- 障害発生直前の最新の同期点
- バックアップ取得時点以降の任意の同期点

> **参考**
>
> データベース複写ユティリティで取得したバックアップの情報を pdbkupls コマンドで確認できます。例えば，次に示す情報を確認できます。確認できる情報の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。
>
> - バックアップの取得日時
> - バックアップ取得対象 RD エリアの名称
> - バックアップ取得モード（-M オプション）の指定値

(1)　バックアップ取得時点に回復する場合

バックアップ取得時点にデータベースを回復する場合は，入力情報（データベース回復ユティリティの入力情報）にバックアップファイルだけが必要です。データベース回復の概要（バックアップ取得時点に回復する場合）を次の図に示します。

図 21–1　データベース回復の概要（バックアップ取得時点に回復する場合）



注意事項

特定の RD エリアをバックアップ取得時点に回復する場合は，次に示すことに注意してください。

- **障害が発生した RD エリアだけを回復すると，そのほかの RD エリアと整合性がとれなくなります。**

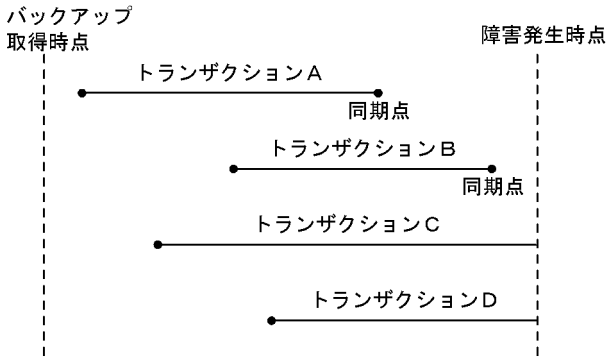
例えば，定義系 SQL の処理中に障害が発生すると，障害が発生したユーザ用 RD エリアだけをバックアップ取得時点の状態に回復しても，データディクショナリ用 RD エリアは障害発生直前の同期点の状態です。したがって，表「同時にバックアップを取得する必要がある RD エリア」で示す RD エリアも同時に取得したバックアップで回復する必要があります。

(2)　障害発生直前の最新の同期点に回復する場合

(a)　回復の対象になるトランザクション

トランザクションを決着した時点を**同期点**といいます。トランザクションによる更新処理を有効にする場合の同期点処理を**コミット**といい，無効にする場合の同期点処理を**ロールバック**といいます。障害発生時点で処理が完了しているトランザクションの同期点にデータベースを回復することを**障害発生直前の最新の同期点に回復する**といいます。障害発生時点で処理中のトランザクション（同期点に達していないトランザクション）は無効になるため，このトランザクションによる更新処理は回復の対象になりません。回復の対象になるトランザクションを次の図に示します。

図 21–2　回復の対象になるトランザクション（障害発生直前の最新の同期点に回復する場合）



〔説明〕

トランザクション A，B は処理を完了して同期点に達しているため，この同期点にデータベースを回復します。

トランザクション C，D は処理中で同期点に達していないため，このトランザクション処理は無効になります。したがって，回復の対象になりません。

(b) 必要となる入力情報

障害発生直前の最新の同期点に回復する場合は，次に示す入力情報（データベース回復ユティリティの入力情報）が必要になります。

- バックアップファイル
- アンロードログファイル※

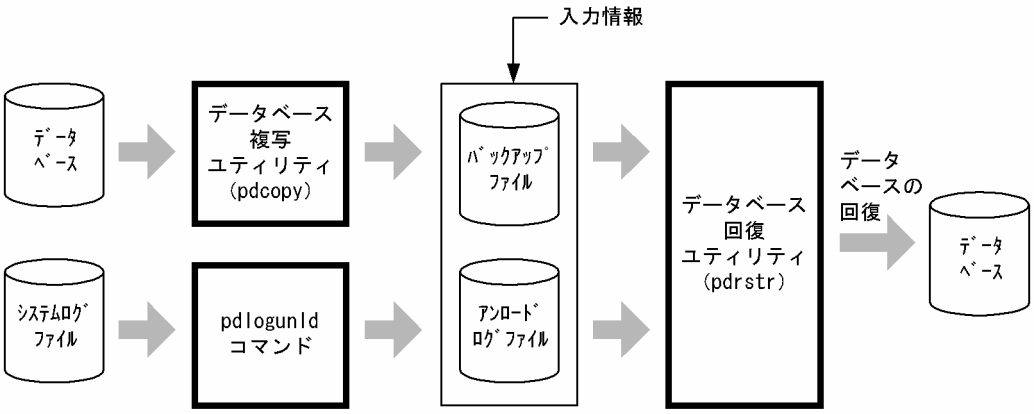
注※

バックアップ取得時点以降のシステムログをアンロードしたアンロードログファイルが必要になります。

アンロードレスシステムログ運用の場合は，バックアップ取得時点以降のシステムログを格納したシステムログファイルが必要になります。

データベース回復の概要（障害発生直前の最新の同期点に回復する場合）を次の図に示します。

図 21–3 データベース回復の概要（障害発生直前の最新の同期点に回復する場合）



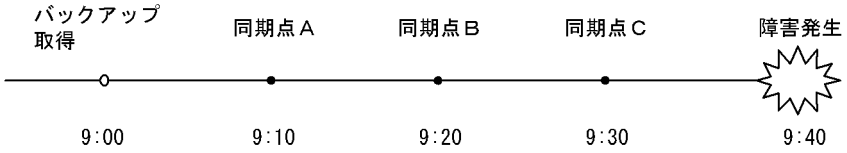
(c) 注意事項（重要）

- 回復対象 RD エリアは障害が発生した RD エリアだけです。
- 回復が完了した RD エリアのバックアップを必ず取得してください。バックアップを取得しないと，その後この RD エリアに障害が発生したときに，この時点の同期点からデータベースを回復できなくなります。
- システムログを使用してマスタディレクトリ用 RD エリアを回復する場合は，**pdstart -r** コマンドで HiRDB を開始する必要があります。すべての RD エリアを最新の同期点に回復する場合の手順については，「例題１（全 RD エリアを回復する場合）」を参照してください。

(3) バックアップ取得時点以降の任意の同期点（範囲指定の回復）

HiRDB 管理者が指定する時点で処理が完了しているトランザクションの同期点にデータベースを回復することを**バックアップ取得時点以降の任意の同期点に回復する**といいます。HiRDB 管理者が指定する時点で処理中のトランザクション（同期点に達していないトランザクション）による更新処理は回復の対象になりません。これを**範囲指定の回復**といいます。範囲指定の回復を次の図に示します。

図 21–4　範囲指定の回復



〔説明〕

どの同期点を回復対象にするかはデータベース回復ユティリティの**-T** オプションで指定します。

- 同期点Ａに戻したい場合は，-T オプションの回復終了時刻に 9:10 以降，9:20 より前の時刻を指定してください。
- 同期点Ｂに戻したい場合は，-T オプションの回復終了時刻に 9:20 以降，9:30 より前の時刻を指定してください。
- 同期点Ｃ（障害発生直前の同期点）に戻したい場合は，-T オプションを指定する必要はありません。

バックアップ取得時点以降の任意の同期点に回復する場合の入力情報（データベース回復ユティリティの入力情報）は，障害発生直前の最新の同期点に回復する場合と同じになります。

21.1.2　バックアップ取得モードとの関連

バックアップ取得時に指定したバックアップ取得モード（pdcopy コマンドの-M オプションの指定）によって，データベースを回復できる時点が異なります。バックアップ取得モードによるデータベースを回復できる時点を次の表に示します。

表 21–1　バックアップ取得モードによるデータベースを回復できる時点

バックアップ取得モード （-M オプションの指定）	バックアップ 取得時点	最新の同期点	範囲指定の回復
参照・更新不可能モード（-M x 指定）	○	○	○
参照可能モード（-M r 指定）	○	○	○
更新可能モード（-M s 指定）※	×	○	○

（凡例）

○：この時点のデータベースを回復できます。

×：この時点のデータベースを回復できません。

注※

更新可能モードを指定してバックアップを取得した場合，データベースを回復するのにバックアップ取得時の現用ファイルのシステムログが必要になります。これに対応するアンロードログファイルがpdrstr コマンドの入力情報に必要になります。

pdcopy コマンドの処理結果出力ファイルに，データベースを回復するときに必要なシステムログファイルの名称及び世代番号が出力されます。

21.1.3　ログ取得モードとの関連

(1)　更新前ログ取得モードで UAP 又はユティリティを実行している場合

更新前ログ取得モードで UAP 又はユティリティを実行しているときにデータベースに障害が発生すると，そのデータベースはバックアップ取得時点にしか回復できません。

(2)　ログレスモードで UAP 又はユティリティを実行している場合

ログレスモードで UAP 又はユティリティを実行しているときにデータベースに障害が発生すると，そのデータベースはバックアップ取得時点にしか回復できません。また，その UAP が更新した RD エリアは**障害閉塞**となります。このときの回復手順については，「ログレスモードで UAP 又はユティリティを実行するときの運用」を参照してください。

21.1.4　各種 RD エリア回復時の注意

(1)　マスタディレクトリ用 RD エリア回復時の注意

回復対象 RD エリアにマスタディレクトリ用 RD エリアが含まれる場合，**pdstart -r** コマンドで HiRDB を開始して，pdrstr コマンドで RD エリアを回復する必要があります。

(2)　データディクショナリ LOB 用 RD エリア回復時の注意

データディクショナリ LOB 用 RD エリアは次に示す用途ごとに分かれています。

- ストアドルーチン及びトリガのソース格納用
- ストアドルーチン及びトリガのオブジェクト格納用

ソース格納用とオブジェクト格納用では，RD エリアを回復できる時点が異なります。

(a)　ソース格納用のデータディクショナリ LOB 用 RD エリアを回復する場合

ソース格納用のデータディクショナリ LOB 用 RD エリアは，バックアップ取得時点及びそれ以降の任意の同期点に回復できます。このときのデータベース回復ユティリティの入力情報は，バックアップ及びシステムログ（アンロードログ）となります。

(b)　オブジェクト格納用のデータディクショナリ LOB 用 RD エリアを回復する場合

オブジェクト格納用のデータディクショナリ LOB 用 RD エリアは，ロールフォワードに必要なデータベース更新ログを取得しないため，バックアップ取得時点だけにしか回復できません。したがって，このときのデータベース回復ユティリティの入力情報はバックアップだけとなります。

なお，次に示す場合は，〈手順〉に従ってオブジェクト格納用のデータディクショナリ LOB 用 RD エリアを回復してください。

- ソース格納用のデータディクショナリ LOB 用 RD エリアをバックアップ取得時点以降の任意の同期点に回復しているが，オブジェクト格納用のデータディクショナリ LOB 用 RD エリアと同期を合わせることができない場合
- pdcopy コマンドの-J オプションを指定してバックアップを取得するときに，意図的にオブジェクト格納用のデータディクショナリ LOB 用 RD エリアを取得しなかった場合

〈手順〉

1. **pdmod** コマンドで，オブジェクト格納用のデータディクショナリ LOB 用 RD エリアを再初期化します。

   pdmod -a C:¥pdmod¥mod01

2. **ALTER ROUTINE** で，全ストアドルーチン及び全トリガの SQL オブジェクトを再作成します。

   ALTER ROUTINE ALL

(3)　ユーザ LOB 用 RD エリア回復時の注意

ユーザ LOB 用 RD エリアを回復する場合，その LOB 列構成基表を格納するユーザ用 RD エリアも同期を合わせて回復してください。

(4)　リスト用 RD エリア回復時の注意

リスト用 RD エリアは pdrstr コマンドでの回復の対象となりません。基表があればリストを再作成できるため，回復の対象にしていません。

なお，リスト用 RD エリアが障害閉塞した場合は，次に示す手順で障害閉塞を解除してください。

〈手順〉

1. **pdclose** コマンドで，障害閉塞したリスト用 RD エリアをクローズします。
2. **pdmod** コマンドで，リスト用 RD エリアを再初期化します。

3. **pdrels -o** コマンドで，リスト用 RD エリアの障害閉塞を解除してオープンします。

> **注意事項**
>
> リスト用 RD エリアを再初期化すると，リスト用 RD エリア内のリストは使用できなくなります。この場合，**ASSIGN LIST 文**でリストを再作成してください。

(5)　障害閉塞の RD エリア回復後の注意

pdstart -r コマンドで HiRDB を開始して，障害閉塞している RD エリアを回復した場合，HiRDB の正常開始後に RD エリアを閉塞解除（**pdrels** コマンド）する必要があります。

(6)　順序数生成子格納 RD エリア回復時の注意

(a)　バックアップ及び回復対象の RD エリアについての注意

順序数生成子格納 RD エリアのバックアップを取得する場合，順序数生成子を使用する表が格納されている RD エリアのバックアップも同時に取得してください。

また，順序数生成子格納 RD エリアと，順序数生成子を使用する表が格納されている RD エリアを回復する場合は，整合性が取れる時点に回復してください。この運用以外の場合，回復後に順序数生成子から取得する順序番号に重複，又は欠番が発生するおそれがあります。

(b)　ログを使用した回復についての注意

ログ出力間隔に，1 より大きな値を指定した順序数生成子を，ログを使用してバックアップ取得時点以降の任意の同期点，又は障害発生直前の最新の同期点まで回復した場合，回復後に順序数生成子から取得する順序番号に欠番が発生するおそれがあります。なお，バックアップ取得後から回復終了となる同期点までの間に，該当する順序数生成子から順序番号を取得したかどうかは関係ありません。

ログを使用した回復後の順序番号に重複，又は欠番が発生するおそれを，次の表に示します。

表 21–2　ログを使用した回復後の順序番号に重複，又は欠番が発生するおそれ

項目	バックアップ取得時点まで回復		バックアップ取得時点以降の任意の同期点，又は障害発生直前の最新の同期点まで回復	
	ログ出力間隔※の指定値		ログ出力間隔※の指定値	
	1	1 より大きな値	1	1 より大きな値
重複	－	－	－	－
欠番	－	－	－	○

（凡例）

－：発生しない

○：発生する

注※

ログ出力間隔については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

また，HiRDB Datareplicator 及び HiRDB Dataextractor は順序数生成子を使用できないため，抽出側と反映側を切り替えて同じ名称の順序数生成子を使用する場合，反映側に抽出側の順序数生成子の現在値を引き継げません。そのため，順序数生成子から取得した順序番号が重複することがあります。

(7)　一時表用 RD エリア回復時の注意

バックアップを取得できないため，一時表用 RD エリアは回復できません。一時表用 RD エリアに障害が発生した場合は，**pdmod** コマンドで一時表用 RD エリアを再初期化してください。

(8)　HiRDB ファイルシステム領域を作成し直す場合の注意

ディスク障害などで，回復対象の RD エリアの HiRDB ファイルシステム領域を作成し直す場合，次に示す設定を変更前の HiRDB ファイルシステム領域と同じにしてください。設定を変更すると，データベースを回復できなくなることがあります。

- ダイレクトディスクアクセス（raw I/O 機能）を適用するかどうか
- pdfmkfs コマンドの-a オプションの指定（HiRDB ファイルシステム領域を自動的に拡張する方式の RD エリアの自動増分を適用するかどうか）

21.1.5　同一ログの再使用チェック

データベース回復ユティリティで，同時にバックアップファイルを指定しないで，アンロードログファイルだけを使用してデータベースを回復する場合，一度使用したログは再使用できません。データベース回復ユティリティの入力情報になったことがあるログを使用して回復すると，データベースが破壊されたり，データ不整合が発生したりするおそれがあります。

これを防ぐために，HiRDB ではデータベースの回復で使用するログをチェックしています。これを**同一ログの再使用チェック**といいます。チェックで，以前回復に使用したログ又は以前回復に使用したログより古いログの指定を検知すると，HiRDB は該当するサーバの回復処理を中断し，データベース回復ユティリティはエラー終了します。このとき，該当するサーバ以外のサーバの RD エリアの回復処理は続行されます。なお，pdrstr コマンドに-b 又は-g オプションを指定して実行した場合，HiRDB は同一ログの再使用チェックは行いません。

(1)　回復処理が中断された場合

同一ログの再使用チェックで回復処理が中断された場合，KFPR26288-E メッセージが出力されます。このメッセージに回復処理を中断されたサーバが表示されるため，そのサーバの RD エリアの回復をやり直してください。ただし，次のような場合は，データベースが更新されていないため，やり直す必要はありません。

- 同じコマンドライン及び同じ制御文ファイルの pdrstr コマンドを，誤って複数回実行してしまい，チェック結果がエラーになった

RD エリアの回復をやり直す場合の回復手順を次に示します。

〈手順〉

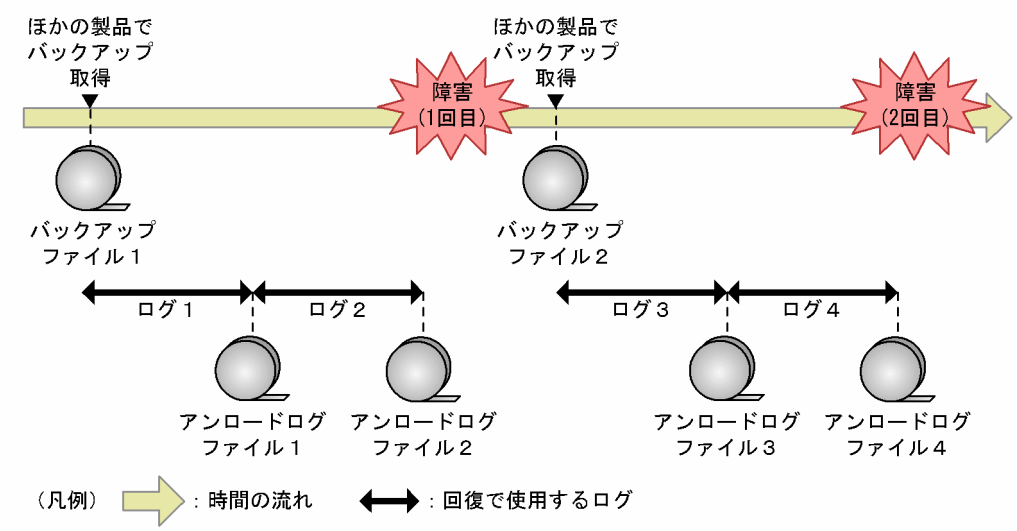
1. pdrstr コマンド又はほかの製品のリストア機能で，KFPR26288-E メッセージで表示されたサーバの RD エリアをバックアップから回復します。
2. 1.で使用したバックアップファイル取得以降のアンロードログファイルを指定して，pdrstr コマンドで回復対象の RD エリアを回復します。

参考

同一ログを使用して回復した場合にデータ不整合が発生する理由

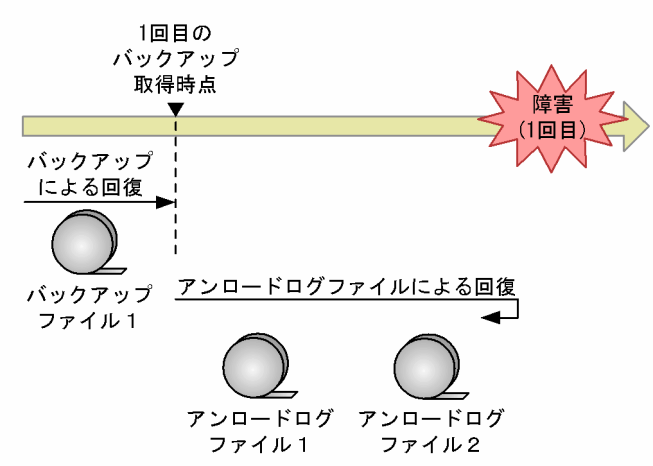
アンロードログファイルだけを使用してデータベースを回復する場合，同一ログを使用して回復するとどのように処理されるかについて，例を使用して説明します。

この例では，ほかの製品のバックアップ機能を使用してバックアップを取得しているとします。



1 回目の障害のとき

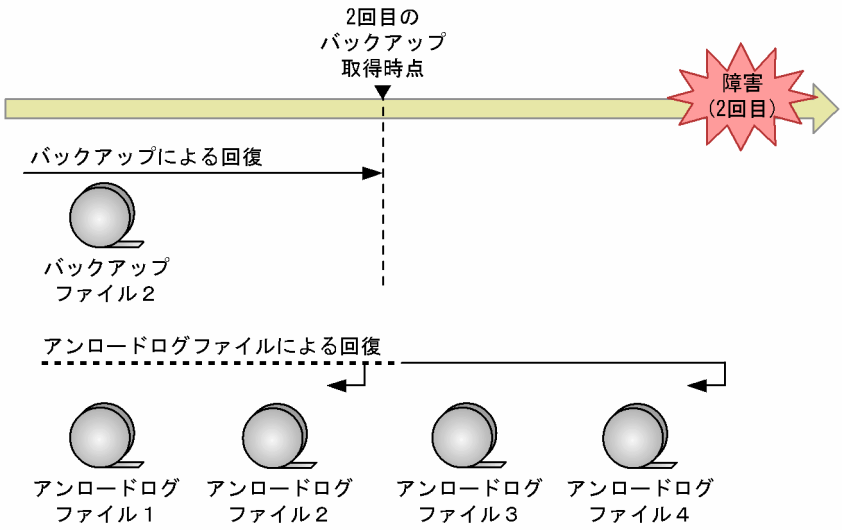
ほかの製品で取得したバックアップファイル 1 でバックアップ取得時点まで回復しました。その後，アンロードログファイル 1 及び 2 を指定して，pdrstr コマンドで障害発生直前の状態まで回復しました。



このとき，ロールフォワードで最新の状態まで回復し，未決着トランザクションのログでロールバックします。

2 回目の障害のとき

ほかの製品で取得したバックアップファイル 2 で 2 回目のバックアップ取得時点まで回復しました。その後，アンロードログファイル 1～4 を指定して pdrstr コマンドで障害発生直前の状態まで回復しました。



このとき，まずロールフォワードで最新の状態まで回復しますが，1 回目の障害回復時にアンロードログファイル 2 まで使用してロールフォワードしているため，そこまでの処理をスキップします。その後，未決着トランザクションのログでロールバックしますが，アンロードログファイル 2 にも未決着トランザクションのログがあるため，アンロードログファイル 2 のトランザクションもロールバックします（ロールバックはスキップしません）。このため，データ不整合になります。

21.1.6　64 ビットモードの HiRDB を使用している方へ

32 ビットモードと 64 ビットモード間のバックアップファイルの互換性はありません。32 ビットモードの HiRDB で取得したバックアップは，64 ビットモードの HiRDB では使用できません。

21.1.7　テープ装置アクセス機能

DAT，DLT，又は LTO 上にあるファイルを，テープ装置アクセス機能を使用してアクセスできます。次に示すファイルに対してテープ装置アクセス機能が適用できます。

- バックアップファイル（pdcopy コマンド又は pdrstr コマンドの-b オプションに指定するバックアップファイル）

テープ装置アクセス機能については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

21.2　データベースをバックアップ取得時点に回復する方法

実行者　HiRDB 管理者

ここでは，データベースをバックアップ取得時点に回復する方法を例題形式で説明します。

21.2.1　例題１（HiRDB Control Manager のウィザードで全 RD エリアを回復する場合）

HiRDB Control Manager のリカバリウィザードを使用して，全 RD エリアを最新のバックアップ取得時点に回復します。

〈手順〉

1. HiRDB Control Manager - Console を起動します。
   起動方法については，「HiRDB Control Manager - Console の起動方法」を参照してください。
2. 操作対象の HiRDB サーバを登録します。
   既に登録されている場合は，この手順は必要ありません。登録方法については，「管理 HiRDB の登録方法」を参照してください。
3. タブメニューの［**リカバリ**］－［**ウィザード**］を選択して，［リカバリウィザード - 回復対象の指定］画面を表示します。
   回復単位は「HiRDB システム単位」を選択します。
   次に，［HiRDB システム］欄から回復対象の HiRDB システム名を選択します。



4. ［**次へ**］をクリックして，［リカバリウィザード - 確認］画面を表示します。



5. 設定した条件を確認して，問題がなければ［**実行**］ボタンをクリックします。
   回復処理が始まります。

- リカバリウィザードでは，最新のバックアップファイルを HiRDB Control Manager が自動的に検索するため，HiRDB 管理者がバックアップファイルを指定する必要はありません。

21.2.2　例題２（全 RD エリアを回復する場合）

システム用 RD エリアを含む全 RD エリアをバックアップ取得時点に回復します。バックアップは，システム単位で取得したバックアップ（全 RD エリアを対象としたバックアップ）があります。

(1)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

HiRDB を正常終了できない場合は，pdstop -f コマンドで強制終了します。なお，既に HiRDB が異常終了している場合は，この手順は必要ありません。

(2)　pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(3)　pdrstr コマンドで全 RD エリアを回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -a
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：全 RD エリアを対象としたバックアップを格納したバックアップファイルの名称を指定します。

-a：全 RD エリアの回復を指定します。

(4)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(5)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(1)の手順で HiRDB を正常終了できなかった場合は，HiRDB を開始する前にシステムログファイル，シンクポイントダンプファイル，及びステータスファイルを初期化してください。

システムログファイル，及びシンクポイントダンプファイルを初期化しない場合，一部のファイルがクローズ状態となることがあります。この場合，クローズ状態となったファイルについては，「システムログファイルに障害が発生したときの対処方法」を参照し，使用できる状態にしてください。

(6)　リスト用 RD エリアを回復します

リスト用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. リスト用 RD エリアを閉塞クローズします。

   ```
   pdhold -r LISER1 -c
   ```

2. pdmod コマンドで RD エリアを再初期化します。

   ```
   pdmod -a /rdarea/conf/control3
   ```

3. リスト用 RD エリアを閉塞解除，オープンします。

   ```
   pdrels -r LISER1 -o
   ```

(7)　一時表用 RD エリアを回復します

一時表用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. 一時表用 RD エリアを閉塞クローズします。

   ```
   pdhold -r TMPRDAREA -c
   ```

2. pdmod コマンドで RD エリアを再初期化します。

```
pdmod -a /rdarea/conf/control4
```

3. 一時表用 RD エリアを閉塞解除，オープンします。

```
pdrels -r TMPRDAREA -o
```

21.2.3　例題３（RD エリア単位に回復する場合）

ユーザ用 RD エリア（rdarea01，rdarea02）をバックアップ取得時点に回復します。なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　pdrstr コマンドで RD エリアを回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -r rdarea01,rdarea02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：rdarea01，rdarea02 のバックアップを格納したバックアップファイルの名称を指定します。

-r：回復する RD エリア（rdarea01，rdarea02）の名称を指定します。

(4)　回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.2.4　例題４（RD エリア単位に回復する場合：マスタディレクトリ用 RD エリアを含む）

マスタディレクトリ用 RD エリアを含む，複数の RD エリアをバックアップ取得時点に回復します。なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。また，RD エリアの回復を複数回の pdrstr コマンドで実行する場合は，最初に実行する pdrstr コマンドにマスタディレクトリ用 RD エリアを含めて回復してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

HiRDB を正常終了，又は計画停止する必要があります。既に HiRDB が異常終了している場合，又は正常終了できない状態の場合は，一度再開始したあとに，正常終了させてください。

正常終了させる理由は，データベース回復ユティリティで回復しない RD エリアに対する，ディレードリランによる回復を不要とするためです。

(3)　pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(4)　pdrstr コマンドで RD エリア（関連する RD エリアも含む）を回復します

回復する RD エリアと，それに関連する RD エリアを回復します。関連する RD エリアは，前回のバックアップ取得以降に実行した処理に応じて異なります。関連する RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照して確認してください。例えば，前回のバックアップ取得以降に CREATE TABLE を実行したときには，表「同時にバックアップを取得する必要がある RD エリア」の CREATE TABLE を参照してください。

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -r MASTER,DDIC,USER1,LOB11
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：バックアップファイル名を指定します。

-r：回復する RD エリア名を指定します。

MASTER：マスタディレクトリ用 RD エリア

DDIC：データディクショナリ用 RD エリア

USER1：ユーザ用 RD エリア（表を追加した RD エリア）

LOB11：ユーザ LOB 用 RD エリア（BLOB 属性の列を格納する RD エリア）

(5) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(6) pdstart コマンドで HiRDB を開始します

```
pdstart
```

21.3　データベースを最新の同期点に回復する方法

実行者　HiRDB 管理者

ここでは，データベースを最新の同期点に回復する方法を例題形式で説明します。

21.3.1　例題１（全 RD エリアを回復する場合）

システム用 RD エリアを含む全 RD エリアを障害発生直前の最新の同期点に回復します。バックアップは，システム単位で取得したバックアップ（全 RD エリアを対象としたバックアップ）があります。システムログファイルの運用方法は，**システムログをアンロードする運用**をしています。

(1)　pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(2)　pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルの内容をアンロードするため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys -w
```

(3)　現用だったファイルの内容を pdlogunld コマンドでアンロードします

(1)で調べた現用ファイルの内容をアンロードしてください。HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバの現用ファイルの内容をアンロードしてください。

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog02
```

●自動ログアンロード機能を使用している場合

この操作は必要ありません。**pdlogatul** コマンドで自動ログアンロード機能が動作していることを確認してください。また，**pdlogls** コマンドでアンロード処理が終了したかどうかを確認してください。

(4)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

HiRDB を正常終了できない場合は，pdstop -f コマンドで強制終了します。なお，既に HiRDB が異常終了している場合は，この手順は必要ありません。

(5) pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(6) pdrstr コマンドで全 RD エリアを最新の同期点に回復します※

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -a
 -l C:¥unld¥unldlog02
```

注※

オブジェクト格納用データディクショナリ LOB 用 RD エリアを使用している場合，ログを使った回復でシステムログを取得しないトランザクションが動作していることを検知（KFPR16334-E）することがあります。

オブジェクト格納用データディクショナリ LOB 用 RD エリアについては，(10)で回復（初期化）するため，エラーを無視して問題ありません。

データベース回復ユティリティでエラーメッセージを出力しないようにするためには，-r オプションでオブジェクト格納用データディクショナリ LOB 用 RD エリア以外の RD エリア名を指定してください。

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：全 RD エリアを対象としたバックアップを格納したバックアップファイルの名称を指定します。

-a：全 RD エリアの回復を指定します。

-l：バックアップファイルの取得時点以降のアンロードログファイルの名称を指定します。

(7) pdcopy コマンドで全 RD エリアのバックアップを取得します

次回，データベースを回復するときに備えて，バックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -b C:¥pdcopy¥backup02 -a
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-b：バックアップファイル名を指定します。

-a：全 RD エリアのバックアップを取得する際に指定します。

(8) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(9) pdstart コマンドで HiRDB を開始します

```
pdstart
```

(4)の手順で HiRDB を正常終了できなかった場合は，HiRDB を開始する前にシステムログファイル，シンクポイントダンプファイル，及びステータスファイルを初期化してください。

システムログファイル，及びシンクポイントダンプファイルを初期化しない場合，一部のファイルがクローズ状態となることがあります。この場合，クローズ状態となったファイルについては，「システムログファイルに障害が発生したときの対処方法」を参照し，使用できる状態にしてください。

(10) オブジェクト格納用データディクショナリ LOB 用 RD エリアを回復します

データディクショナリ LOB 用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. オブジェクト格納用データディクショナリ LOB 用 RD エリアを閉塞クローズします。

   ```
   pdhold -r RDDICOBJ -c
   ```

2. pdmod コマンドで，オブジェクト格納用データディクショナリ LOB 用 RD エリアを再初期化します。

   ```
   pdmod -a C:¥rdarea¥conf¥control2
   ```

3. オブジェクト格納用データディクショナリ LOB 用 RD エリアを閉塞解除，オープンします。

   ```
   pdrels -r RDDICOBJ -o
   ```

4. ALTER ROUTINE で，全ストアドルーチン及び全トリガの SQL オブジェクトを再作成します。

   ```
   ALTER ROUTINE ALL
   ```

(11) リスト用 RD エリアを回復します

リスト用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. リスト用 RD エリアを閉塞クローズします。

   ```
   pdhold -r LISER1 -c
   ```

2. pdmod コマンドで RD エリアを再初期化します。

   ```
   pdmod -a C:¥rdarea¥conf¥control3
   ```

3. リスト用 RD エリアを閉塞解除，オープンします。

   ```
   pdrels -r LISER1 -o
   ```

(12)　一時表用 RD エリアを回復します

一時表用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. 一時表用 RD エリアを閉塞クローズします。

```
pdhold -r TMPRDAREA -c
```

2. pdmod コマンドで RD エリアを再初期化します。

```
pdmod -a C:¥rdarea¥conf¥control4
```

3. 一時表用 RD エリアを閉塞解除，オープンします。

```
pdrels -r TMPRDAREA -o
```

21.3.2　例題２（HiRDB Control Manager のウィザードで RD エリア単位に回復する場合）

HiRDB Control Manager のリカバリウィザードを使用して，RD エリア単位の回復を実行します。実行する前に，次のどちらかの条件を満たしていることを確認してください。

- 自動ログアンロード機能を使用している
- アンロードログファイルを HiRDB Control Manager に登録している

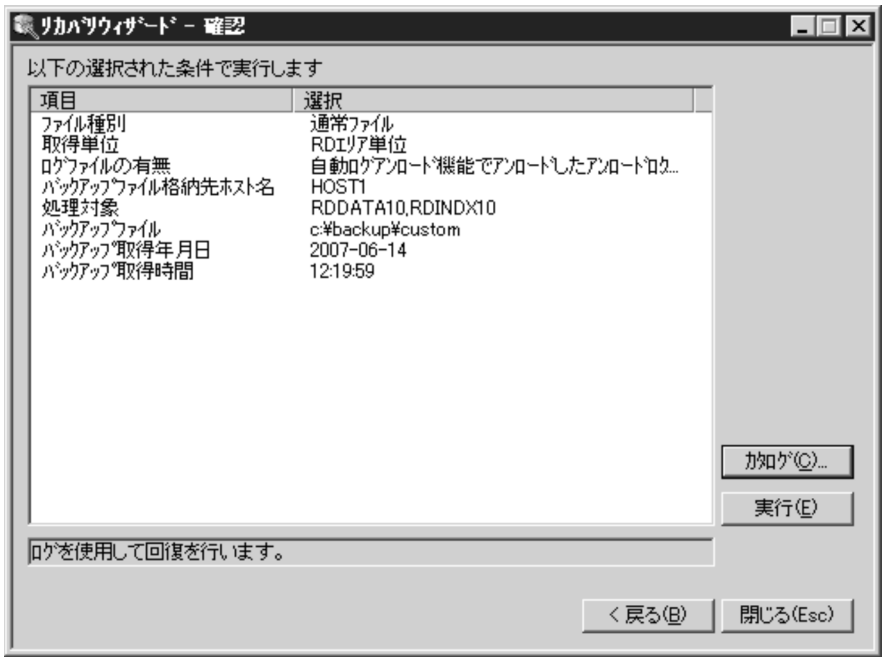
HiRDB Control Manager にアンロードログファイルを登録する方法については，HiRDB Control Manager のヘルプを参照してください。

〈手順〉

1. HiRDB Control Manager - Console を起動します。
   起動方法については，「HiRDB Control Manager - Console の起動方法」を参照してください。
2. 操作対象の HiRDB サーバを登録します。
   既に登録されている場合は，この手順は必要ありません。登録方法については，「管理 HiRDB の登録方法」を参照してください。
3. タブメニューの［**リカバリ**］－［**ウィザード**］を選択して，［リカバリウィザード - 回復対象の指定］画面を表示します。
   回復単位として，「RD エリア単位」を選択します。
   次に，「HiRDB システム」，「ユニット」，「サーバ」から，回復対象の RD エリアが属する対象を選択すると，［RD エリア］欄に RD エリアの一覧が表示されます。その中から回復対象の RD エリアを選択します。



4. **[次へ]** をクリックして，［リカバリウィザード - 確認］画面を表示します。



5. 設定した条件を確認して，問題がなければ **[実行]** ボタンをクリックします。
   回復処理が始まります。

- リカバリウィザードでは，最新のバックアップファイルやアンロードログファイルを HiRDB Control Manager が自動的に検索するため，HiRDB 管理者がこれらのファイルを指定する必要はありません。

21.3.3　例題３（RD エリア単位に回復する場合）

ユーザ用 RD エリア（rdarea01，rdarea02）を障害発生直前の最新の同期点に回復します。システムログファイルの運用方法は，**システムログをアンロードする運用**をしています。

なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(4)　pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルの内容をアンロードするため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys -w
```

(5)　現用だったファイルの内容を pdlogunld コマンドでアンロードします

(3)で調べた現用ファイルの内容をアンロードしてください。HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバの現用ファイルの内容をアンロードしてください。

```
pdlogunld -d sys -g log01 -o C:¥unld¥unldlog02
```

●自動ログアンロード機能を使用している場合

この操作は必要ありません。**pdlogatul** コマンドで自動ログアンロード機能が動作していることを確認してください。また，**pdlogls** コマンドでアンロード処理が終了したかどうかを確認してください。

(6) pdrstr コマンドで RD エリアを最新の同期点に回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01
-l C:¥unld¥unldlog01,C:¥unld¥unldlog02 -w C:¥tmp¥sortwork -r rdarea01,rdarea02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：rdarea01 及び rdarea02 のバックアップが格納されているバックアップファイルの名称を指定します。

-l：アンロードログファイルの名称を指定します。

-w：ソート用ワークディレクトリの名称を指定します。

-r：回復する RD エリア（rdarea01，rdarea02）の名称を指定します。

- 範囲指定の回復をする場合は，−T オプションに回復終了時刻を指定してください。−T オプションを指定しないと，障害発生直前の最新の同期点に RD エリアを回復します。
- システムログファイル又はアンロードログファイルの障害によって入力できないアンロードログファイルがある場合，そのサーバの RD エリアは，障害が発生した箇所までしか回復できません（それ以外のサーバの RD エリアには影響ありません）。必要に応じて，次の方法で障害が発生した箇所を特定し，トランザクションを再度実行してください。

1. 障害発生時に KFPR16203-E メッセージに出力された，アンロードログファイル名を確認します。
2. 手順 1 で確認したアンロードログファイルに対して pdlogucat コマンドを実行し，ファイルが作成されたサーバ名と日時（作成開始時刻）を確認します。
3. 手順 2 で確認したサーバで，作成開始時刻以降に実行されたトランザクションを再度実行します。

(7) pdcopy コマンドで回復した RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -r rdarea01,rdarea02
-b C:¥pdcopy¥backup02 -z C:¥pdcopy¥logpoint01-p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照・更新不可能モードを指定します。

-r：障害が発生した RD エリア（rdarea01，rdarea02）のバックアップを取得します。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。自動ログアンロード機能を使用している場合に指定することをお勧めします。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(8) 回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.3.4 例題４（RD エリア単位に回復する場合：マスタディレクトリ用 RD エリアを含む）

マスタディレクトリ用 RD エリアを含む，複数の RD エリアを障害発生直前の最新の同期点に回復します。

なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

また，RD エリアの回復を複数回の pdrstr コマンドで実行する場合は，最初に実行する pdrstr コマンドにマスタディレクトリ用 RD エリアを含めて回復してください。

(1) 回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2) pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(3) pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルの内容をアンロードするため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys -w
```

(4) 現用だったファイルの内容を pdlogunld コマンドでアンロードします

(2)で調べた現用ファイルの内容をアンロードしてください。HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバの現用ファイルの内容をアンロードしてください。

```
pdlogunld -d sys -s bes1 -g log01 -o C:¥unld¥unldlog02
```

●自動ログアンロード機能を使用している場合

この操作は必要ありません。**pdlogatul** コマンドで自動ログアンロード機能が動作していることを確認してください。また，**pdlogls** コマンドでアンロード処理が終了したかどうかを確認してください。

(5) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

HiRDB を正常終了，又は計画停止する必要があります。既に HiRDB が異常終了している場合，又は正常終了できない状態の場合は，一度再開始したあとに，正常終了させてください。

正常終了させる理由は，データベース回復ユティリティで回復しない RD エリアに対する，ディレードリランによる回復を不要とするためです。

(6) pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(7) pdrstr コマンドで RD エリア（関連する RD エリアも含む）を最新の同期点に回復します

回復する RD エリアと，それに関連する RD エリアを回復します。関連する RD エリアは，前回のバックアップ取得以降に実行した処理に応じて異なります。関連する RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照して確認してください。例えば，前回のバックアップ取得以降に CREATE TABLE を実行したときには，表「同時にバックアップを取得する必要がある RD エリア」のCREATE TABLE を参照してください。

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -r MASTER,DDIC,USER1,LOB11
-l C:¥rdarea¥jnldir¥unload_log01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：バックアップファイル名を指定します。

-r：回復する RD エリア名を指定します。

MASTER：マスタディレクトリ用 RD エリア

DDIC：データディクショナリ用 RD エリア

USER1：ユーザ用 RD エリア（表を追加した RD エリア）

LOB11：ユーザ LOB 用 RD エリア（BLOB 属性の列を格納する RD エリア）

-l：バックアップファイルの取得時点以降のアンロードログファイルの名称を指定します。

(8) pdcopy コマンドで全 RD エリアのバックアップを取得します

次回，データベースを回復するときに備えて，バックアップを取得します。

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -b C:¥pdcopy¥backup02 -a
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードを指定します。

-b：バックアップファイル名を指定します。

-a：全 RD エリアのバックアップを取得する際に指定します。

(9) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(10) pdstart コマンドで HiRDB を開始します

```
pdstart
```

HiRDB を正常終了又は計画停止できなかった場合は，システムログファイル，シンクポイントダンプファイル，及びステータスファイルを初期化した後，正常開始してください。

ステータスファイルは，いったん pdstsrm コマンドで削除した後に pdstsinit コマンドで再作成してください。

システムログファイル及びシンクポイントダンプファイルを初期化しない場合，一部のファイルがクローズ状態となることがあります。この場合，クローズ状態となったファイルについては，「システムログファイルに障害が発生したときの対処方法」を参照して対処してください。

21.3.5　例題５（RD エリア単位に回復する場合：アンロードレスシステムログ運用の場合）

ユーザ用 RD エリア（rdarea01，rdarea02）を障害発生直前の最新の同期点に回復します。システムログファイルの運用方法は，**アンロードレスシステムログ運用**をしています。

なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルのままではデータベース回復ユティリティの入力情報にできないため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys
```

(4)　pdrstr コマンドで RD エリアを最新の同期点に回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -L
-w C:¥tmp¥sortwork -r rdarea01,rdarea02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：rdarea01 及び rdarea02 のバックアップが格納されているバックアップファイルの名称を指定します。

-L：システムログファイルを入力情報とする指定をします。

-w：ソート用ワークディレクトリの名称を指定します。

-r：回復する RD エリア（rdarea01，rdarea02）の名称を指定します。

範囲指定の回復をする場合は，−T オプションに回復終了時刻を指定してください。−T オプションを指定しないと，障害発生直前の最新の同期点に RD エリアを回復します。

(5) pdcopy コマンドで回復した RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M x -r rdarea01,rdarea02 -b C:¥pdcopy¥backup02
-z C:¥pdcopy¥logpoint01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照・更新不可能モードを指定します。

-r：障害が発生した RD エリア（rdarea01，rdarea02）のバックアップを取得します。

-b：バックアップファイル名を指定します。

-z：ログポイント情報ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(6) 回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.4　差分バックアップ機能使用時のデータベースの回復方法

実行者　HiRDB 管理者

ここでは，差分バックアップ機能使用時，次に示す時点にデータベースを回復する方法について説明します。

- 最新の差分バックアップ取得時点
- 最新の同期点

21.4.1　例題１（最新の差分バックアップ取得時点に回復する場合）

ユーザ用 RD エリア（rdarea01，rdarea02）を差分バックアップ取得時点に回復します。

なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　pdrstr コマンドで RD エリアを回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -g backupg1 -K C:¥pdcopy¥admfile
-r rdarea01,rdarea02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-g：差分バックアップグループ名を指定します。

-K：差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域名を指定します。

-r：回復する RD エリア（rdarea01，rdarea02）の名称を指定します。

(4)　回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.4.2　例題２（最新の同期点に回復する場合）

ユーザ用 RD エリア（rdarea01，rdarea02）を障害発生直前の最新の同期点に回復します。システムログファイルの運用方法は，**システムログをアンロードする運用**をしています。

なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(4)　pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルの内容をアンロードするため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys
```

(5) 現用だったファイルの内容を pdlogunld コマンドでアンロードします

(3)で調べた現用ファイルの内容をアンロードしてください。HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバの現用ファイルの内容をアンロードしてください。

```
pdlogunld -d sys -g log01 -o C:¥unld¥unldlog02
```

(6) pdrstr コマンドで RD エリアを最新の同期点に回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -g backupg1 -K C:¥pdcopy¥admfile
-l C:¥unld¥unldlog02 -w C:¥tmp¥sortwork -r rdarea01,rdarea02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-g：差分バックアップグループ名を指定します。

-K：差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域名を指定します。

-l：アンロードログファイルの名称を指定します。

-w：ソート用ワークディレクトリの名称を指定します。

-r：回復する RD エリア（rdarea01，rdarea02）の名称を指定します。

(7) pdcopy コマンドで差分バックアップグループのフルバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -r rdarea01,rdarea02,…
-g backupg1(S) -b C:¥pdcopy¥backup01 -d a -K C:¥pdcopy¥admfile -L 5
-o C:¥pdcopy¥rfile
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-r：バックアップ取得対象 RD エリアを指定します。

ここで指定した RD エリア群が差分バックアップグループになります。バックアップ取得対象 RD エリアは途中で変更できません。

-g：差分バックアップグループ名を指定します。

フルバックアップ取得時には，差分バックアップグループ名に(S)を指定してください。次回以降の差分バックアップ取得時に，ここで指定した差分バックアップグループ名を指定します。

-b：バックアップファイル名（フルバックアップファイル名）を指定します。

-d：バックアップ種別を指定します。

a：フルバックアップを取得します。

b：最新のフルバックアップからの累積差分バックアップを取得します。

c：前回取得した累積差分バックアップ，又は前回取得したフルバックアップのどちらか最新のものから，累積差分バックアップを取得します。

d：差分バックアップを取得します。

累積差分バックアップについては，「累積差分バックアップの作成」を参照してください。

-K：差分バックアップ管理ファイルを格納する HiRDB ファイルシステム領域名を指定します。

-L：差分バックアップ管理ファイルの容量をメガバイト単位で指定します。

-o：差分バックアップの履歴情報ファイル名を指定します。

バックアップの取得については，「バックアップの取得方法」を参照してください。

(8) 回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.4.3 差分バックアップ管理ファイルがない場合の回復方法

差分バックアップ管理ファイルがない場合は，pdrstr コマンドを複数回（バックアップの取得回数分）実行してデータベースを回復します。

例題 2 の回復方法を実行する場合は，「pdrstr コマンドで RD エリアを最新の同期点に回復します」の操作方法が異なります。ここでは，その操作方法だけを説明します。差分バックアップの取得方法は次のとおりとします。

- 日曜日にフルバックアップを取得
- 月曜日と火曜日に差分バックアップを取得
- 水曜日にデータベースを回復

(1) フルバックアップを入力情報にして pdrstr コマンドで RD エリアを回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -g backupg1 -K C:¥pdcopy¥admfile
-b C:¥pdcopy¥backup01 -r rdarea01,rdarea02
```

〔説明〕

-b：フルバックアップファイル名を指定します。

(2) 差分バックアップを入力情報にして pdrstr コマンドで RD エリアを回復します

月曜日に取得した差分バックアップを入力情報にして pdrstr コマンドで RD エリアを回復します。

```
pdrstr -m C:¥rdarea¥mast¥mast01 -g backupg1 -K C:¥pdcopy¥admfile
-b C:¥pdcopy¥backup02 -r rdarea01,rdarea02
```

〔説明〕

-b：差分バックアップファイル名を指定します。

(3) 差分バックアップを入力情報にして pdrstr コマンドで RD エリアを回復します

火曜日に取得した差分バックアップを入力情報にして pdrstr コマンドで RD エリアを回復します。

```
pdrstr -m C:¥rdarea¥mast¥mast01 -g backupg1 -K C:¥pdcopy¥admfile
-b C:¥pdcopy¥backup03 -r rdarea01,rdarea02
```

〔説明〕

-b：差分バックアップファイル名を指定します。

(4) アンロードログファイルを入力情報にして pdrstr コマンドで RD エリアを回復します

火曜日に取得した差分バックアップ以降の回復を，アンロードログファイルを入力情報にして pdrstr コマンドで行います。なお，ここでエラーが発生した場合は，(1)から操作をし直してください。また，アンロードログファイルを入力情報にして回復をした場合，回復後に必ずフルバックアップを取得してください。

```
pdrstr -m C:¥rdarea¥mast¥mast01 -l C:¥unld¥unldlog01 -w C:¥tmp¥sortwork
-r rdarea01,rdarea02
```

〔説明〕

-l：アンロードログファイル名を指定します。

21.5　データベースをほかの HiRDB システムのバックアップから回復する方法

実行者　HiRDB 管理者

バックアップを取得した HiRDB システムと，回復しようとする HiRDB システムの構成が同じ※場合，pdcopy コマンドで取得したバックアップを使用して，ほかの HiRDB システムで pdrstr コマンドを実行して，データベースを回復できます。これによって，サーバマシンにハードウェア障害が発生した場合でも，別のサーバマシンを使用してシステムを回復できます。

注※

構成が同じとは，次に示す項目が同じということです。

- RD エリアを構成する HiRDB ファイルのパス
- HiRDB システム定義

  （ただし，ホスト名称，ポート番号，及び HiRDB 運用ディレクトリは，バックアップを取得した HiRDB システムと回復しようとする HiRDB システムとで異なってもよい）
- HiRDB のバージョン
- HiRDB の形名

注意事項

- 回復する場合に使用するバックアップファイルには，データベースの整合性が取れている静止点でマスタディレクトリ用 RD エリアを含むすべての RD エリアが必要です。データベースを回復する場合も，マスタディレクトリ用 RD エリアを含むすべての RD エリアを回復する必要があります。
- この方法では，バックアップ取得時点までの回復だけができます（システムログファイルを使用した回復はできません）。
- バックアップを取得した HiRDB システムにリスト用 RD エリア，一時表用 RD エリアがある場合，ほかの RD エリアの回復後にそれらの RD エリアを再初期化する必要があります。
- 回復しようとする HiRDB システムに既にデータが格納されている場合，回復することで，それらのデータが失われます。必要に応じて回復する前にバックアップを取得してください。
- バックアップファイルから回復することで，スキーマや接続ユーザなどの各権限も回復されます。そのため，データベースからのデータ抽出やデータ検索は，回復後のスキーマで実行してください。
- この方法でデータベースを回復する場合，処理を誤るとデータベースが破壊されるため，注意が必要です。例えば，回復する RD エリアが不足している場合など，データベース回復ユティリティ（pdrstr）は正常に終了しますが，データベースは不整合な状態になります。

21.5.1 例題（テスト環境の HiRDB システム A で取得したバックアップを本番環境の HiRDB システム B に移行）

テスト環境の HiRDB システム A でバックアップを取得し，同じシステム構成の本番環境の HiRDB システム B に移行します。HiRDB システム A 及び HiRDB システム B ともに HiRDB は稼働しています。

(1) HiRDB システム A でバックアップを取得します

(a) すべての RD エリア（マスタディレクトリ用 RD エリアを除く）を閉塞かつクローズ状態にします

```
pdhold -r ALL -c
```

(b) すべての RD エリアのバックアップを取得します

```
pdcopy -m C:¥rdarea¥mast¥mast01 -M r -a -b C:¥pdcopy¥backup01
-p C:¥pdcopy¥list01
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-M：バックアップ取得モードに参照可能モードを指定します。

-a：全 RD エリアのバックアップを取得することを指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの実行結果リストの出力ファイル名を指定します。

(c) すべての RD エリア（マスタディレクトリ用 RD エリアを除く）の閉塞を解除し，オープン状態にします

```
pdrels -r ALL -o
```

(2) HiRDB システム A のバックアップファイルを HiRDB システム B に転送します

(1)で取得した HiRDB システム A の全 RD エリアのバックアップファイル（C:¥pdcopy¥backup01）を，HiRDB システム B に転送します。

(3) HiRDB システム A のバックアップファイルを使用して，HiRDB システム B で回復します

(a) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(b) pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(c) pdrstr コマンドで全 RD エリアを回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -b C:¥pdcopy¥backup01 -a
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-b：HiRDB システム A で取得したバックアップファイルの名称を指定します。

-a：全 RD エリアの回復を指定します。

(d) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(e) pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

コマンドの実行結果が正しいかどうか確認することをお勧めします。コマンドの実行結果の確認方法については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

21.6　pdcopy コマンド以外でバックアップを取得した場合の回復方法

実行者　HiRDB 管理者

ここでは，pdcopy コマンド以外でバックアップを取得した場合（ほかの製品の機能でバックアップを取得した場合）のデータベースの回復方法について説明します。データベースを回復するおおまかな手順は次のとおりです。

〈手順〉

1. ほかの製品のリストア機能を使用してデータベースをバックアップ取得時点に回復します。
2. pdrstr コマンドで，データベースを最新の同期点に回復します。

- バックアップ取得時点に回復する場合は 1 の手順だけを行ってください。
- 最新の同期点への回復又は範囲指定の回復で，回復対象 RD エリアにマスタディレクトリ用 RD エリアが含まれている場合，pdstart -r コマンドで HiRDB を開始し，RD エリアを回復してください。

21.6.1　例題１（全 RD エリアをバックアップ取得時点に回復する場合）

システム用 RD エリアを含む全 RD エリアをバックアップ取得時点に回復します。バックアップは，システム単位で取得したバックアップ（全 RD エリアを対象としたバックアップ）があります。

(1)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

HiRDB を正常終了できない場合は，pdstop -f コマンドで強制終了します。なお，既に HiRDB が異常終了している場合は，この手順は必要ありません。

(2)　ほかの製品のリストア機能で全 RD エリアを回復します

ほかの製品のリストア機能で全 RD エリアを回復します。

(3)　pdstart コマンドで HiRDB を正常開始します

```
pdstart
```

(1)の手順で HiRDB を正常終了できなかった場合は，HiRDB を開始する前にシステムログファイル，シンクポイントダンプファイル，及びステータスファイルを初期化してください。

システムログファイル，及びシンクポイントダンプファイルを初期化しない場合，一部のファイルがクローズ状態となることがあります。この場合，クローズ状態となったファイルについては，「システムログファイルに障害が発生したときの対処方法」を参照し，使用できる状態にしてください。

21.6.2　例題２（RD エリア単位にバックアップ取得時点に回復する場合）

ユーザ用 RD エリア（rdarea01，rdarea02）をバックアップ取得時点に回復します。なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　ほかの製品のリストア機能で RD エリアを回復します

ほかの製品のリストア機能で RD エリア（rdarea01，rdarea02）を構成する HiRDB ファイルシステム領域を回復します。

(4)　回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.6.3　例題３（全 RD エリアを最新の同期点に回復する場合）

システム用 RD エリアを含む全 RD エリアを障害発生直前の最新の同期点に回復します。バックアップは，システム単位で取得したバックアップ（全 RD エリアを対象としたバックアップ）があります。システムログファイルの運用方法は，**システムログをアンロードする運用**をしています。

(1)　pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(2) pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルの内容をアンロードするため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys
```

(3) 現用だったファイルの内容を pdlogunld コマンドでアンロードします

(1)で調べた現用ファイルの内容をアンロードしてください。HiRDB/パラレルサーバの場合は，ディクショナリサーバ及びバックエンドサーバの現用ファイルの内容をアンロードしてください。

```
pdlogunld -d sys -g log01 -o C:¥unld¥unldlog02
```

(4) pdstop コマンドで HiRDB を正常終了します

HiRDB を正常終了できない場合は，pdstop -f コマンドで強制終了します。なお，既に HiRDB が異常終了している場合は，この手順は必要ありません。

```
pdstop
```

(5) すべての RD エリアをバックアップ取得時点に回復します

ほかの製品のリストア機能で，すべての RD エリアをバックアップ取得時点に回復します。

(6) pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(7) pdrstr コマンドですべての RD エリアを最新の同期点に回復します※

```
pdrstr -m C:¥rdarea¥mast¥mast01 -l C:¥unld¥unldlog01,C:¥unld¥unldlog02
-w C:¥tmp¥sortwork -a
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-l：アンロードログファイルの名称を指定します。

-w：ソート用ワークディレクトリの名称を指定します。

-a：すべての RD エリアを対象にします。

注※

オブジェクト格納用データディクショナリ LOB 用 RD エリアを使用している場合，ログを使った回復でシステムログを取得しないトランザクションが動作していることを検知（KFPR16334-E）することがあります。

オブジェクト格納用データディクショナリ LOB 用 RD エリアについては，(11)で回復（初期化）するため，エラーを無視して問題ありません。データベース回復ユティリティでエラーメッセージを出力しないようにするためには，-r オプションでオブジェクト格納用データディクショナリ LOB 用 RD エリア以外の RD エリアだけ指定してください。

(8)　pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(9)　全 RD エリアのバックアップを取得します

ほかの製品のバックアップ機能で全 RD エリアのバックアップを取得します。

(10)　pdstart コマンドで HiRDB を開始します

```
pdstart
```

(4)の手順で HiRDB を正常終了できなかった場合は，HiRDB を開始する前にシステムログファイル，シンクポイントダンプファイル，及びステータスファイルを初期化してください。

システムログファイル，及びシンクポイントダンプファイルを初期化しない場合，一部のファイルがクローズ状態となることがあります。この場合，クローズ状態となったファイルについては，「システムログファイルに障害が発生したときの対処方法」を参照し，使用できる状態にしてください。

(11)　オブジェクト格納用データディクショナリ LOB 用 RD エリアを回復します

データディクショナリ LOB 用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. オブジェクト格納用データディクショナリ LOB 用 RD エリアを閉塞クローズします。

   ```
   pdhold -r RDDICOBJ -c
   ```

2. pdmod コマンドで，オブジェクト格納用データディクショナリ LOB 用 RD エリアを再初期化します。

   ```
   pdmod -a c:¥rdarea¥conf¥control2
   ```

3. オブジェクト格納用データディクショナリ LOB 用 RD エリアを閉塞解除，オープンします。

   ```
   pdrels -r RDDICOBJ -o
   ```

4. ALTER ROUTINE で，全ストアドルーチン及び全トリガの SQL オブジェクトを再作成します。

```
ALTER ROUTINE ALL
```

(12)　リスト用 RD エリアを回復します

リスト用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. リスト用 RD エリアを閉塞クローズします。

```
pdhold -r LISER1 -c
```

2. pdmod コマンドで RD エリアを再初期化します。

```
pdmod -a c:¥rdarea¥conf¥control3
```

3. リスト用 RD エリアを閉塞解除，オープンします。

```
pdrels -r LISER1 -o
```

(13)　一時表用 RD エリアを回復します

一時表用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. 一時表用 RD エリアを閉塞クローズします。

```
pdhold -r TMPRDAREA -c
```

2. pdmod コマンドで RD エリアを再初期化します。

```
pdmod -a c:¥rdarea¥conf¥control4
```

3. 一時表用 RD エリアを閉塞解除，オープンします。

```
pdrels -r TMPRDAREA -o
```

21.6.4　例題４（RD エリア単位に回復する場合）

ユーザ用 RD エリア（rdarea01，rdarea02）を障害発生直前の最新の同期点に回復します。システムログファイルの運用方法は，**システムログをアンロードする運用**をしています。

なお，処理対象の RD エリアだけではなく，関連のある RD エリアについても同時に回復する必要があります。回復対象となる RD エリアについては，「同時にバックアップを取得する必要がある RD エリア」を参照してください。

(1)　回復対象の RD エリアに関連のある RD エリアを確認します

この操作は，ユーザ用 RD エリア，及びユーザ LOB 用 RD エリアに対してだけ実行します。また，横分割表を定義している場合は，pdrdrefls コマンドの-t オプションを実行し，横分割表を構成する RD エリアを確認します。

回復対象の RD エリアが不足している場合は，追加します。

```
pdrdrefls -k chk -e org -c ref
-r rdarea01,rdarea02
```

(2)　障害閉塞している RD エリアを pdclose コマンドでクローズします

```
pdclose -r rdarea01,rdarea02
```

(3)　pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(4)　pdlogswap コマンドでシステムログファイルをスワップします

現用ファイルの内容をアンロードするため，システムログファイルをスワップします。HiRDB/パラレルサーバの場合は，rdarea01 及び rdarea02 があるバックエンドサーバのシステムログファイルをスワップしてください。

```
pdlogswap -d sys
```

(5)　現用だったファイルの内容を pdlogunld コマンドでアンロードします

(3)で調べた現用ファイルの内容をアンロードしてください。

```
pdlogunld -d sys -g log01 -o C:¥unld¥unldlog02
```

(6)　RD エリアをバックアップ取得時点に回復します

ほかの製品のリストア機能を使用して，rdarea01 及び rdarea02 をバックアップ取得時点に回復します。

(7)　pdrstr コマンドで RD エリアを最新の同期点に回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -l C:¥unld¥unldlog01,C:¥unld¥unldlog02
-w C:¥tmp¥sortwork -r rdarea01,rdarea02
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-l：アンロードログファイルの名称を指定します。

-w：ソート用ワークディレクトリの名称を指定します。

-r：回復する RD エリア（rdarea01，rdarea02）の名称を指定します。

(8) 回復した RD エリアのバックアップを取得します

ほかの製品のバックアップ機能で rdarea01 及び rdarea02 のバックアップを取得します。

(9) 回復した RD エリアの障害閉塞を pdrels コマンドで解除してオープンします

```
pdrels -r rdarea01,rdarea02 -o
```

21.6.5 例題５（マスタディレクトリ用 RD エリアだけを回復する場合）

マスタディレクトリ用 RD エリアだけを障害発生直前の最新の同期点に回復します。システムログファイルの運用方法は，**システムログをアンロードする運用**をしています。

(1) pdlogls コマンドで現用のシステムログファイルを調べます

HiRDB/パラレルサーバの場合は，全サーバのシステムログファイルを調べてください。

```
pdlogls -d sys
```

(2) 現用だったファイルの内容を pdlogunld コマンドでアンロードします

(1)で調べた現用ファイルの内容をアンロードしてください。HiRDB/パラレルサーバの場合は，全サーバの現用ファイルをアンロードしてください。

```
pdlogunld -d sys -g log01 -o C:¥unld¥unldlog02
```

(3) マスタディレクトリ用 RD エリアをバックアップ取得時点に回復します

ほかの製品のリストア機能を使用して，マスタディレクトリ用 RD エリアをバックアップ取得時点に回復します。

(4) pdstart -r コマンドで HiRDB を開始します

```
pdstart -r
```

(5) pdrstr コマンドで RD エリアを最新の同期点に回復します

```
pdrstr -m C:¥rdarea¥mast¥mast01 -l C:¥unld¥unldlog01,C:¥unld¥unldlog02
-w C:¥tmp¥sortwork -r MAST
```

〔説明〕

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名称を指定します。

-l：アンロードログファイルの名称を指定します。

-w：ソート用ワークディレクトリの名称を指定します。

-r：マスタディレクトリ用 RD エリア（MAST）の名称を指定します。

(6) pdstop コマンドで HiRDB を正常終了します

```
pdstop
```

(7) マスタディレクトリ用 RD エリアのバックアップを取得します

ほかの製品のバックアップ機能を使用して，マスタディレクトリ用 RD エリアを構成する HiRDB ファイルシステム領域のバックアップを取得してください。

(8) pdstart コマンドで HiRDB を開始します

```
pdstart
```

21.7　1:1 スタンバイレス型系切り替え時のシステム用 RD エリアを含む回復方法

実行者　HiRDB 管理者

ここでは，1:1 スタンバイレス型系切り替え時のシステム用 RD エリアを含む回復について説明します。

1:1 スタンバイレス型系切り替え構成で，BES が存在するユニットが系切り替えしているために切り戻しができない場合，pdstart -r ですべてのユニットを起動することができません。すべてのユニットを pdstart -r で起動できない場合で，ディスク障害などによって RD エリアを回復する必要があるときは，この手順に従って回復させてください。

21.7.1　例題１（全 RD エリアを最新の同期点に回復する場合）

(1)　pdstart -r -q でユニットを開始します

次のユニットでコマンドを実行してください。

- システムマネジャがあるユニット
- ディクショナリサーバがあるユニット

```
pdstart -r -q
```

(2)　システム用 RD エリアを回復します

-s オプションにディクショナリサーバ名を指定し，ディクショナリサーバの RD エリアをすべて最新の同期点に回復します。

```
pdrstr -m %PDDIR%¥dbdir¥dic¥master -b %PDDIR%¥bkupdir¥bkup01 -s dic
-l %PDDIR%¥dbdir¥dic_unldlog1
```

オブジェクト格納用データディクショナリ LOB 用 RD エリアを使用している場合，ログを使った回復でシステムログを取得しないトランザクションが動作していることを検知（KFPR16334-E）することがあります。

オブジェクト格納用データディクショナリ LOB 用 RD エリアについては，(10)で回復（初期化）するため，エラーを無視して問題ありません。データベース回復ユティリティでエラーメッセージを出力しないようにするためには，-r オプションでオブジェクト格納用データディクショナリ LOB 用 RD エリア以外の RD エリア名を指定してください。

(3)　pdstop コマンドで HiRDB を終了します

```
pdstop
```

(4) pdstart コマンドで HiRDB を開始します

```
pdstart
```

この手順を実施する前に HiRDB を正常終了，又は計画停止できなかった場合は，HiRDB を開始する前にシステムログファイル，シンクポイントダンプファイル，及びステータスファイルを初期化してください。

システムログファイル，及びシンクポイントダンプファイルを初期化しない場合，一部のファイルがクローズ状態となることがあります。この場合，クローズ状態となったファイルについては，「システムログファイルに障害が発生したときの対処方法」を参照し，使用できる状態にしてください。

(5) ディクショナリサーバ以外の RD エリアを閉塞，クローズ状態にします

```
pdhold -r ALL -c
```

(6) ディクショナリサーバの RD エリア以外の RD エリアを最新の同期点に回復します

```
pdrstr -m %PDDIR%¥dbdir¥dic¥master -b %PDDIR%¥bkupdir¥bkup01
-s bes1,bes2 -l %PDDIR%¥dbdir¥bes1_unldlog1,%PDDIR%¥dbdir¥bes2_unldlog1
```

(7) 全 RD エリアのバックアップを取得します

バックアップの取得方法は，「バックアップの取得方法」を参照してください

(8) pdstop コマンドで HiRDB を終了します

```
pdstop
```

(9) pdstart コマンドで HiRDB を開始します

```
pdstart
```

(10) オブジェクト格納用データディクショナリ LOB 用 RD エリアを回復します

データディクショナリ LOB 用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. オブジェクト格納用データディクショナリ LOB 用 RD エリアを閉塞クローズします。

   ```
   pdhold -r RDDICOBJ -c
   ```

2. pdmod コマンドで，オブジェクト格納用データディクショナリ LOB 用 RD エリアを再初期化します。

```
pdmod -a c:¥rdarea¥conf¥control2
```

3. オブジェクト格納用データディクショナリ LOB 用 RD エリアを閉塞解除，オープンします。

```
pdrels -r RDDICOBJ -o
```

4. ALTER ROUTINE で，全ストアドルーチン及び全トリガの SQL オブジェクトを再作成します。

```
ALTER ROUTINE ALL
```

(11) リスト用 RD エリアを回復します

リスト用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. リスト用 RD エリアを閉塞クローズします。

```
pdhold -r LISER1 -c
```

2. pdmod コマンドで RD エリアを再初期化します。

```
pdmod -a c:¥rdarea¥conf¥control3
```

3. リスト用 RD エリアを閉塞解除，オープンします。

```
pdrels -r LISER1 -o
```

(12) 一時表用 RD エリアを回復します

一時表用 RD エリアがない場合は，この手順は必要ありません。

〈手順〉

1. 一時表用 RD エリアを閉塞クローズします。

```
pdhold -r TMPRDAREA -c
```

2. pdmod コマンドで RD エリアを再初期化します。

```
pdmod -a c:¥rdarea¥conf¥control4
```

3. 一時表用 RD エリアを閉塞解除，オープンします。

```
pdrels -r TMPRDAREA -o
```

22 チューニング情報の取得方法

この章では，チューニングするときに必要なチューニング情報の取得方法について説明します。

22.1　統計ログからチューニング情報を取得する方法

ここでは，統計ログからチューニング情報を取得する方法について説明します。説明する項目は次のとおりです。

- 統計ログから取得できるチューニング情報
- チューニング情報を取得するための準備
- チューニング情報の取得
- アプリケーションから HiRDB にアクセスする場合

22.1.1　統計ログから取得できるチューニング情報

統計ログから取得できるチューニング情報を次の表に示します。

表 22–1　統計ログから取得できるチューニング情報

統計情報の種別		説明
システムの稼働に関する統計情報	sys	サーバプロセス，RPC，及びシステムログなどに関する稼働情報を出力します。
サーバごとのシステムの稼働に関する統計情報	svr	サーバごとのシステムの稼働に関する統計情報を出力します。
UAP に関する統計情報	uap	検索行数及び各 SQL 実行回数など，UAP に関する情報を編集時間単位に編集し，UAP 又はトランザクションごとに出力します。
SQL に関する統計情報	sql	検索行数，作業表作成回数，及び SQL 内分割コマンドを発行したバックエンドサーバ数など，SQL の発行に関する情報を UAP 又はサービスごとに出力します。
グローバルバッファに関する統計情報※1	buf	バッファヒット率及び実入出力回数など，グローバルバッファのアクセスに関する情報を編集時間単位に編集し，サーバ単位及びグローバルバッファ単位に出力します。
データベース操作に関する HiRDB ファイルの統計情報	fil	同期入出力回数及び入出力エラー回数など，HiRDB ファイルへのアクセスに関する情報を編集時間単位に編集し，サーバ単位，HiRDB ファイル単位，及び RD エリア単位に出力します。
デファードライト処理に関する統計情報	dfw	デファードライト処理の回数，動作要因，入出力並列度など，デファードライト処理に関する情報を編集時間単位に編集し，サーバ単位に出力します。
インデクスに関する統計情報	idx	統計ログ及びシステムログを基にして，インデクスキー排他情報及びインデクススプリット情報など，インデクスに関する情報を編集時間単位に編集し，サーバ単位及びインデクス単位に出力します。
SQL 静的最適化に関する統計情報	sop	SQL の静的最適化情報を出力します。DAT 形式ファイルにだけ出力します。
SQL 動的最適化に関する統計情報	dop	SQL の動的最適化情報を出力します。DAT 形式ファイルにだけ出力します。

統計情報の種別		説明
SQL オブジェクト実行に関する統計情報	pcd	SQL オブジェクトの実行情報を出力します。DAT 形式ファイルにだけ出力します。
SQL 文の履歴に関する統計情報※2	sqh	発行された操作系 SQL，定義系 SQL，及び LOCK 文と SQL に関する情報を出力します。DAT 形式ファイルにだけ出力します。
SQL オブジェクト転送に関する統計情報※3	obj	SQL オブジェクトの転送情報を出力します。DAT 形式ファイルにだけ出力します。

注※1

これらの統計情報はシンクポイント時に取得され，シンクポイント間の情報が編集されます。このため，シンクポイントが 2 回以上発生しないと統計情報が取得されません。これらの統計情報を確実に取得するためには，pdstjsync 又は pdstend コマンドの実行直前に pdlogsync コマンドを実行してシンクポイントを発生させてください。

注※2

SQL 文の履歴に関する統計情報は，SQL に関する統計情報（sql 指定）を出力している場合に出力されます。

注※3

HiRDB/パラレルサーバ限定の統計情報です。HiRDB/シングルサーバの場合は出力しません。

22.1.2　チューニング情報を取得するための準備

実行者　HiRDB 管理者

チューニング情報を取得するためには，HiRDB が出力する統計情報（統計ログ）を取得する必要があります。統計ログは統計ログファイルに出力されるので，HiRDB 管理者は統計ログファイルのファイル容量を決定して運用してください。

(1)　統計ログファイルの作成

統計ログファイルは，%PDDIR%¥spool 下に二つあり，ファイル名称は pdstj1 及び pdstj2 です。統計ログファイルは HiRDB が自動的に作成します。HiRDB 管理者は，統計ログファイルのファイル容量を pd_stj_file_size オペランドで指定してください。なお，ファイル容量を決定するとき，次に示す点を考慮してください。

- ファイル容量には，取得する統計ログの総容量を指定します。複数の統計情報を同時に取得すると考えられる場合，それらすべての統計ログの容量を合計した値より大きな値を **pd_stj_file_size** オペランドに指定してください。各統計ログの容量の計算式については，マニュアル「HiRDB システム定義」を参照してください。
- サーバの種別によって統計ログが取得されたり，されなかったりします。したがって，それを考慮してファイル容量を決定してください。統計ログの取得対象サーバを次の表に示します。

表 22–2　統計ログの取得対象サーバ

統計情報の種別		HiRDB/シングルサーバ	HiRDB/パラレルサーバ			
		SDS	MGR	FES	DS	BES
システムの稼働に関する統計情報	sys	○	○	○	○	○
UAP に関する統計情報	uap	○	×	○	×	×
SQL に関する統計情報	sql	○	×	○	×	×
グローバルバッファに関する統計情報	buf	○	×	×	○	○
データベース操作に関する HiRDB ファイルの統計情報	fil	○	×	×	○	○
デファードライト処理に関する統計情報	dfw	○	×	×	○	○
インデクスに関する統計情報※	idx	○	×	×	○	○
SQL 静的最適化に関する統計情報	sop	○	×	○	×	×
SQL 動的最適化に関する統計情報	dop	○	×	○	×	×
SQL オブジェクト実行に関する統計情報	pcd	○	×	○	○	○
SQL 文の履歴に関する統計情報	sqh	○	×	○	×	×
SQL オブジェクト転送に関する統計情報	obj	×	×	×	○	○

（凡例）

○：取得対象とします。

×：取得対象としません。

注

- pdhibegin オペランドで取得した「CONNECT/DISCONNECT に関する統計情報」は統計ログファイルではなく，システムログファイルに出力されます。詳細については，「システムログからチューニング情報を取得する方法」を参照してください。

注※

統計ログでは，インデクスページスプリットに関する情報は取得できません。このため，インデクスページスプリットに関する情報も取得する場合は，「システムログからチューニング情報を取得する方法」を参照してください。

(2)　統計ログファイルの運用

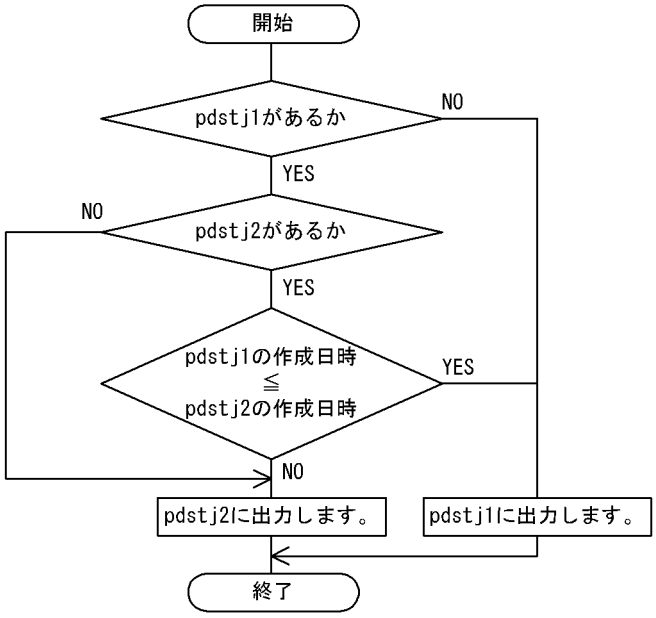
統計ログの出力中に統計ログファイルの容量が一杯になると，HiRDB はその統計ログファイルへの統計ログの出力を中止して，もう一方の統計ログファイルに統計ログを出力します。これを統計ログファイルのスワップといいます。このとき，スワップ先の統計ログファイルに格納されている古い統計ログは上書きされます。したがって，HiRDB 管理者は，必要な統計ログが上書きされる前に統計ログをアンロードして，アンロード統計ログファイルを作成してください。統計ログファイルのスワップを次の図に示します。

図 22–1　統計ログファイルのスワップ



（凡例）　：統計ログが書き込まれている領域
：空き領域
：ファイルの切り替え（スワップ）

〔説明〕
pdstj1の統計ログファイルが一杯になると，統計ログの出力先ファイルがpdstj2に切り替わります。
そのとき，pdstj2に格納されていた古い統計ログは上書きされます。
なお，HiRDB開始時の統計ログの出力先ファイルは，次のように決定します。統計ログの出力先ファイルの決定手順を次に示します。

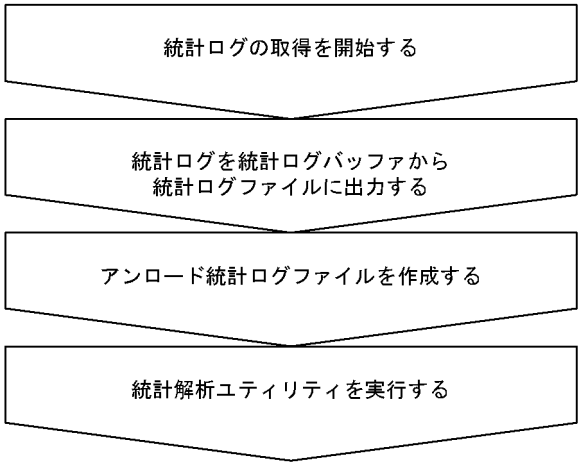


22.1.3　チューニング情報の取得

実行者　HiRDB 管理者

ここでは，チューニング情報の取得手順（統計ログの取得から，その統計情報を入力情報とする統計解析ユティリティの実行までの運用方法）について説明します。チューニング情報の取得手順（統計ログからチューニング情報を取得する場合）を次の図に示します。

図 22–2　チューニング情報の取得手順（統計ログからチューニング情報を取得する場合）



(1)　統計ログの取得を開始する

pdstbegin コマンドで統計ログの取得を開始します。HiRDB の開始時から統計ログを取得する場合は，システム共通定義の **pdstbegin** オペランドを指定してください。pdstbegin オペランドを指定した場合，pdstbegin コマンドの実行は不要です。

注意事項

統計ログの取得中にシンクポイントが 2 回以上発生しないと，次に示す統計情報は取得されません。確実に取得するには，pdstbegin コマンドの入力直後に pdlogsync コマンドでシンクポイントを取得し，pdstjsync 又は pdstend コマンドの入力直前に pdlogsync コマンドでシンクポイントを取得してください。

- グローバルバッファに関する統計情報
- データベース操作に関する HiRDB ファイルの統計情報
- デファードライト処理に関する統計情報
- インデクスに関する統計情報

参考

統計ログの取得を終了する場合は **pdstend** コマンドを実行します。

(2)　統計ログを統計ログバッファから統計ログファイルに出力する

pdstjsync コマンドで，統計ログバッファに格納されている統計ログを統計ログファイルに出力します。

参考

統計ログバッファに格納されている統計ログが，統計ログファイルに出力される契機を次に示します。

- 統計ログバッファの満杯時
- pdstjsync コマンドの実行時
- pdstend コマンドの実行時
- HiRDB の終了時（正常終了又は計画停止）

(3)　アンロード統計ログファイルを作成する

エクスプローラで統計ログファイルをコピーして，アンロード統計ログファイルを作成します。

(4)　統計解析ユティリティを実行する

(3)で作成したアンロード統計ログファイルを入力情報として，統計解析ユティリティを実行します。HiRDB 管理者はその実行結果を基にして，HiRDB 又は UAP のチューニングを実施してください。

参考

統計情報は，pdstbegin コマンドの入力から pdstend（又は pdstjsync）コマンドの入力までが対象となるため，コマンドの入力時期によっては UAP に関する統計情報と SQL に関する統計情報が一致しないことがあります。

22.1.4　アプリケーションから HiRDB にアクセスする場合

ここでは，UAP 又は OLTP システムのアプリケーションから HiRDB にアクセスする場合の，統計情報の取得タイミング及び統計解析ユティリティの編集単位について示します。

(1)　UAP から HiRDB にアクセスする場合

HiRDB にアクセスする UAP に関する統計情報の取得タイミングを，次の表に示します。

表 22–3　HiRDB にアクセスする UAP に関する統計情報の取得タイミング

統計情報の 種別	統計ログの 出力タイミング	統計解析ユティリティの 編集単位	DAT 出力 ファイル解析
UAP に関する統計情報	コネクションごと	UAP 名ごと • UAP 実行回数，UAP 平均実行時間，各 SQL 文の実行回数など	UAP 名をキーに解析できる

統計情報の種別	統計ログの出力タイミング	統計解析ユティリティの編集単位	DAT 出力ファイル解析
		• 1 回の UAP 実行ごとの各項目の合計，平均値	
SQL に関する統計情報	SQL ごと	UAP 名ごと • SQL オブジェクト長の合計，平均値 • フロントエンドサーバからバックエンドサーバへのオブジェクト送信数合計，平均 • SQL 実行回数，処理時間，処理した行数	UAP 名をキーに解析できる
SQL 最適化情報に関する統計情報	SQL ごと	UAP 名ごと • 回数，実行時間などの平均値	UAP 名をキーに解析できる
SQL オブジェクト実行に関する統計情報	SQL ごと	UAP 名ごと • 回数，実行時間などの平均値	UAP 名をキーに解析できる

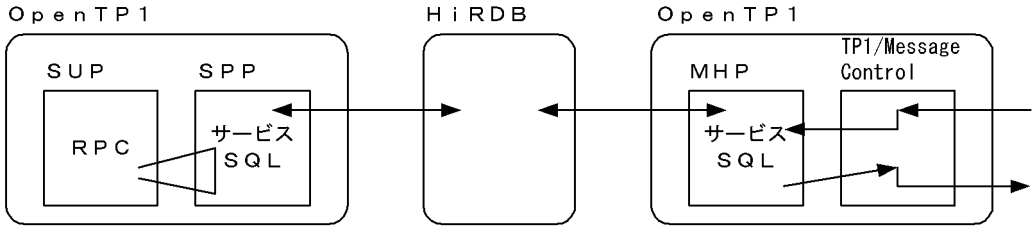
(2) OpenTP1 と連携している場合

OpenTP1 と連携している場合，統計情報の取得タイミング及び統計解析ユティリティの編集単位が変わります。ただし，統計情報の取得タイミング及び統計解析ユティリティの編集単位が変わるのは(a)に示す運用形態のときだけです。

なお，ここでいう OpenTP1 との連携とは，TP1/Server Base 及び TP1/Message Control（バージョン 03-06 以降）のアプリケーションから HiRDB にアクセスすることを表しています。

(a) クライアント／サーバ形態の UAP でサーバ UAP 内のサービス実行に伴いアクセスする形態

OpenTP1 の SPP（サービス提供プログラム）や MHP（メッセージ処理プログラム）の 1 サービスから HiRDB にアクセスするような形態のアプリケーションです。



この形態の場合，OpenTP1 の UAP では 1UAP 内で複数のサービスを形成でき，それぞれのサービスからそのサービスの用途に合った HiRDB へのアクセスができます。このような形態のときの統計情報には，SPP 又は MHP の UAP 名（SPP 又は MHP に関する OpenTP1 のユーザサーバ定義上で指定した PDCLTAPNAME の指定値）に加え，SPP 又は MHP に要求したサービス（HiRDB アクセスを伴うトランザクション内での最初のサービス）のサービス名を付けます。このサービス名は SUP 又は TP1/Message Control が RPC を使用して要求するときに指定するサービス名で，SPP 又は MHP を構成するサービスに付けたサービス名です。

このサービス名によって，どのサービスの延長でアクセスされたものかの識別ができます。また，OpenTP1 のトランザクション統計情報とサービス名で対応がとれます。OpenTP1 と連携している場合の統計情報の取得タイミングを次の表に示します。

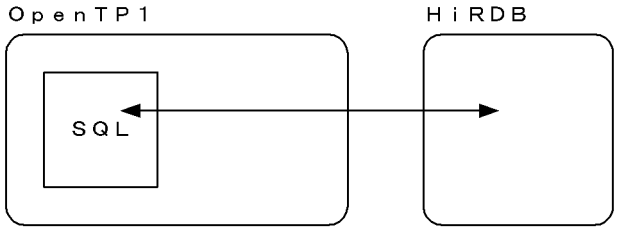
表 22–4　OpenTP1 と連携している場合の統計情報の取得タイミング

<table>
<tr><th colspan="3">統計情報の種別</th><th>統計ログの
出力タイミング</th><th>統計解析ユティリティの
編集単位</th><th>DAT 出力
ファイル解析</th></tr>
<tr><td rowspan="3">UAP に関する統計情報</td><td rowspan="2">サービス名なし</td><td>通常インタフェース</td><td>コネクションごと</td><td rowspan="2">UAP 名ごと
• UAP 実行回数，UAP 平均実行時間，各 SQL 文の実行回数など
• 1 回の UAP 実行ごとの各項目の合計，平均値</td><td rowspan="2">トランザクション，UAP 名をキーに解析できる</td></tr>
<tr><td>XA インタフェース</td><td>トランザクションごと</td></tr>
<tr><td>サービス名あり</td><td>通常インタフェース</td><td>コネクションごと
• ログ出力時，コネクション内 1st サービス名を付加</td><td>UAP 内のサービス名ごと
• 各 SQL 文の実行回数など</td><td>トランザクション，UAP 名，サービス名をキーに解析できる</td></tr>
<tr><td rowspan="2">SQL に関する統計情報</td><td colspan="2">サービス名なし</td><td>SQL ごと</td><td>UAP 名ごと
• SQL オブジェクト長の合計，平均値
• フロントエンドサーバからバックエンドサーバへのオブジェクト送信数合計，平均
• SQL 実行回数，処理時間，処理した行数</td><td>UAP 名をキーに解析できる</td></tr>
<tr><td colspan="2">サービス名あり</td><td>SQL ごと
• ログ出力時，トランザクション内 1st サービス名を付加</td><td>UAP 内のサービス名ごと
• 上記情報をサービスごとに集計</td><td>UAP 名，サービス名をキーに解析できる</td></tr>
<tr><td rowspan="2">SQL 最適化情報に関する統計情報</td><td colspan="2">サービス名なし</td><td>SQL ごと</td><td>UAP 名ごと
• SQL 実行回数，実行時間などの平均</td><td>UAP 名をキーに解析できる</td></tr>
<tr><td colspan="2">サービス名あり</td><td>SQL ごと
• ログ出力時，トランザクション内 1st サービス名を付加</td><td>UAP 内のサービス名ごと
• 上記情報をサービスごとに集計</td><td>UAP 名，サービス名をキーに解析できる</td></tr>
<tr><td>SQL オブジェクト実行に関</td><td colspan="2">サービス名なし</td><td>SQL ごと</td><td>UAP 名ごと
• SQL 実行回数，実行時間などの平均</td><td>UAP 名をキーに解析できる</td></tr>
</table>

統計情報の種別		統計ログの 出力タイミング	統計解析ユティリティの 編集単位	DAT 出力 ファイル解析
する統計情報	サービス名あり	SQL ごと • ログ出力時，トランザクション内 1st サービス名を付加	UAP 内のサービス名ごと • 上記情報をサービスごとに集計	UAP 名，サービス名をキーに解析できる

(b)　メインルーチンから直接アクセスする形態

OpenTP1 の SUP（サービス利用プログラム）から直接 HiRDB をアクセスするような形態のアプリケーションです。



この形態は，OpenTP1 と接続していない通常のアプリケーションと統計情報の差はありません。統計情報上の UAP 名は，SUP に関する OpenTP1 のユーザサーバ定義上で指定した PDCLTAPNAME の指定値となります。

(3)　TPBroker for C++，又は TUXEDO と連携している場合

TPBroker for C++又は TUXEDO と連携している場合，統計情報の取得タイミング及び統計解析ユティリティの編集単位が変わります。TPBroker for C++又は TUXEDO と連携している場合の統計情報の取得タイミングを次の表に示します。

表 22–5　TPBroker for C++，TUXEDO，又は WebLogic Server と連携している場合の統計情報の取得タイミング

統計情報の種別		統計ログの 出力タイミング	統計解析ユティリティの 編集単位	DAT 出力 ファイル解析
UAP に関する統計情報	通常インタフェース	コネクションごと	UAP 名ごと • UAP 実行回数，UAP 平均実行時間，各 SQL 文の実行回数など • 1 回の UAP 実行ごとの各項目の合計，平均値	トランザクション，UAP 名をキーに解析できる
	XA インタフェース	トランザクションごと		
SQL に関する統計情報	通常インタフェース	SQL ごと	UAP 名ごと • SQL オブジェクト長の合計，平均値	UAP 名をキーに解析できる
	XA インタフェース			

統計情報の種別		統計ログの 出力タイミング	統計解析ユティリティの 編集単位	DAT 出力 ファイル解析
			• フロントエンドサーバからバックエンドサーバへのオブジェクト送信数合計，平均 • SQL 実行回数，処理時間，処理した行数	
SQL 最適化情報に関する統計情報	通常インタフェース	SQL ごと	UAP 名ごと • 回数，実行時間などの平均値	UAP 名をキーに解析できる
	XA インタフェース			
SQL オブジェクト実行に関する統計情報	通常インタフェース	SQL ごと	UAP 名ごと • 回数，実行時間などの平均値	UAP 名をキーに解析できる
	XA インタフェース			

22.2　システムログからチューニング情報を取得する方法

実行者　HiRDB 管理者

22.2.1　システムログから取得できるチューニング情報

システムログから取得できるチューニング情報を次の表に示します。

表 22–6　システムログから取得できるチューニング情報

統計情報の種別		説明
インデクスに関する統計情報	idx	インデクスに関する統計情報中のインデクスページスプリットに関する情報が出力されます。
CONNECT/DISCONNECT に関する統計情報※	cnc	CONNECT/DISCONNECT に関する情報を出力します。DAT 形式ファイルにだけ出力します。

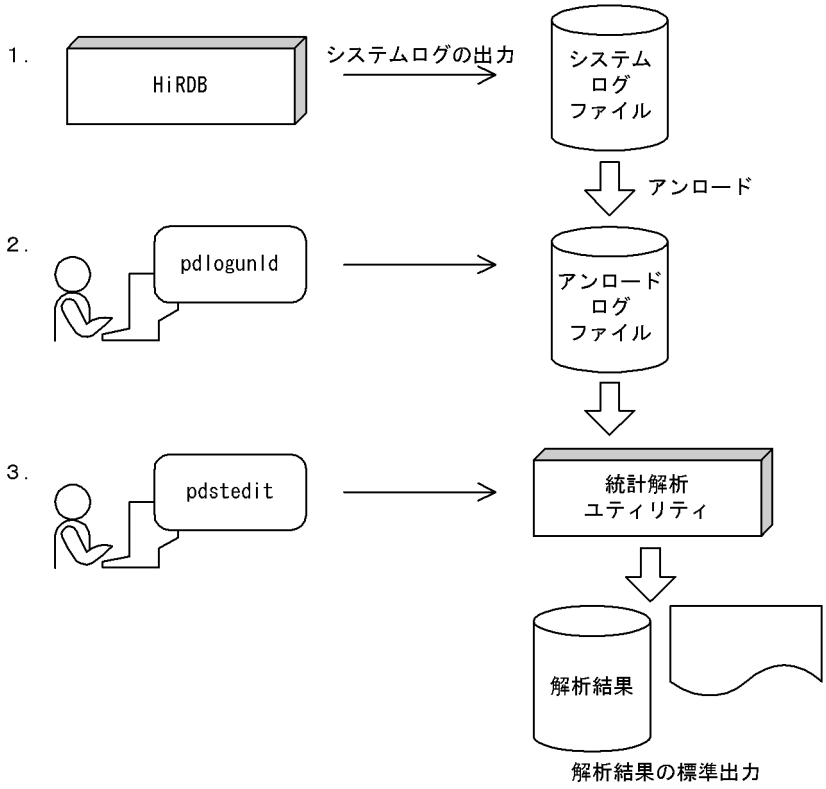
注※

HiRDB/パラレルサーバの場合は，フロントエンドサーバのシステムログファイルに統計情報が出力されます。

22.2.2　チューニング情報の取得手順

チューニング情報の取得手順（システムログからチューニング情報を取得する場合）を次の図に示します。

図 22–3　チューニング情報の取得手順（システムログからチューニング情報を取得する場合）



〔説明〕

1. HiRDB が，システムログファイルにシステムログを取得します。
2. **pdlogunld** コマンドで，システムログファイルの内容をアンロードし，アンロードログファイルを作成してください。

   また，システムログを直接入力して統計解析ユティリティを実行する運用もあります。統計解析ユティリティについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。
3. 作成したアンロードログファイルを入力情報として，統計解析ユティリティを実行してください。ユーザはその実行結果を基にして，HiRDB 又は UAP のチューニングを実施してください。

22.3　データベース状態解析ユティリティからチューニング情報を取得する方法

ここでは，データベース状態解析ユティリティからチューニング情報を取得する方法について説明します。

22.3.1　チューニング情報を取得するための準備

実行者　HiRDB 管理者

データベース状態解析ユティリティからチューニング情報を取得する場合，該当する RD エリアは，次の表に示す状態にする必要があります。したがって，**pddbls** コマンドで RD エリアの状態を確認した後，データベース状態解析ユティリティを実行してください。

DBA 権限だけを持っているユーザは，RD エリアの状態を確認できないため，HiRDB 管理者へ連絡し，実行できる状態かどうかを問い合わせてください。

表 22–7　データベース状態解析ユティリティからチューニング情報を取得するときの RD エリアの状態

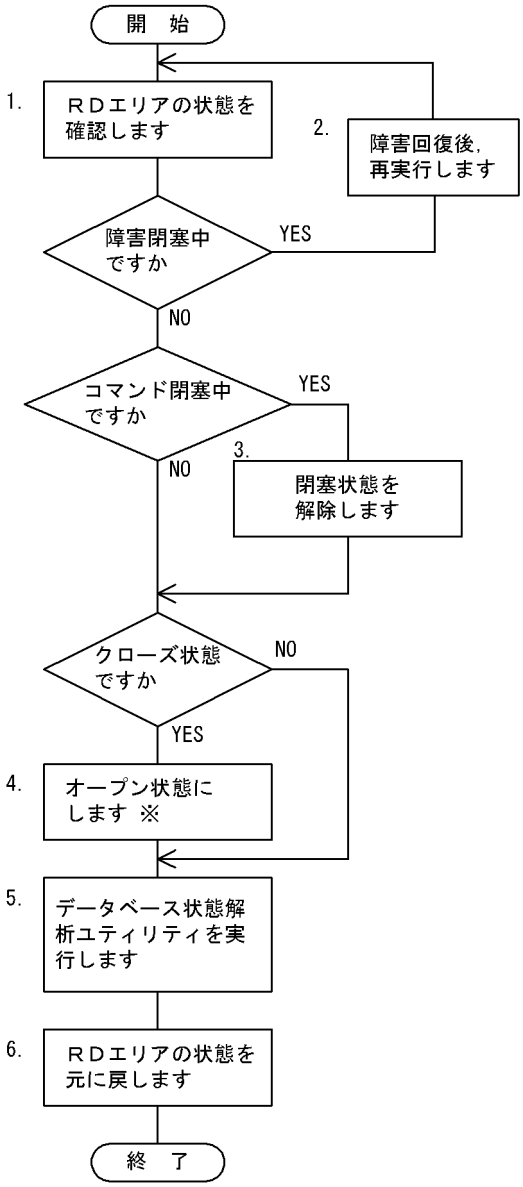
<table>
<tr><th colspan="2">RD エリアの種類</th><th>RD エリアの状態</th></tr>
<tr><td colspan="2">データディクショナリ用 RD エリア</td><td rowspan="5">• 閉塞なしオープン状態
• 参照可能閉塞オープン状態</td></tr>
<tr><td colspan="2">データディクショナリ LOB 用 RD エリア</td></tr>
<tr><td colspan="2">レジストリ用 RD エリア</td></tr>
<tr><td colspan="2">レジストリ LOB 用 RD エリア</td></tr>
<tr><td colspan="2">リスト用 RD エリア</td></tr>
<tr><td rowspan="2">ユーザ用 RD エリア
ユーザ LOB 用 RD エリア</td><td>RD エリアのオープン属性が INITIAL の場合</td><td>• 閉塞なしオープン状態
• 参照可能閉塞オープン状態</td></tr>
<tr><td>RD エリアのオープン属性が DEFER 又は SCHEDULE の場合</td><td>• 閉塞なしオープン状態
• 閉塞なしクローズ状態
• 参照可能閉塞オープン状態
• 参照可能閉塞クローズ状態</td></tr>
</table>

22.3.2　チューニング情報の取得

実行者　DBA 権限を持っているユーザ

ここでは，チューニング情報の取得手順（運用コマンドの入力からデータベース状態解析ユティリティの実行までの運用方法）について説明します。チューニング情報の取得手順（データベース状態解析ユティリティからチューニング情報を取得する場合）を次の図に示します。

図 22–4　チューニング情報の取得手順（データベース状態解析ユティリティからチューニング情報を取得する場合）



注※

RD エリアオープン属性が DEFER 又は SCHEDULE の場合で，参照可能閉塞状態又は閉塞なしのときは，オープン状態にする必要はありません。

〔説明〕

1. pddbls コマンドで RD エリアの状態を確認します。
2. 障害閉塞中の場合は障害回復をした後に pdrels コマンドで閉塞状態を解除します。その後，再実行します。
3. コマンド閉塞中の場合は pdrels コマンドで RD エリアの閉塞状態を解除します。
4. クローズ状態の場合は pdopen コマンドでオープン状態にします。

5. データベース状態解析ユティリティを実行します。その実行結果を基にしてチューニングを実施してください。
6. 4 でオープン状態にした場合は pdclose コマンドでクローズ状態に戻します。3 でコマンド閉塞中にした場合は pdrels コマンドでコマンド閉塞中を解除します。

23 チューニング

この章では，データベースのチューニング方法について説明します。

データベースをチューニングするには，処理性能の観点からチューニングを進める必要があります。しかし，処理性能だけを意識すると，使用しているメモリやディスク容量では，対応できない場合が考えられます。このため，処理性能，使用しているメモリ，及びディスク容量のバランスを考慮してチューニングする必要があります。

23.1　グローバルバッファプールのチューニング

グローバルバッファプールの指定は，ディスクに対するデータの入出力性能に大きな影響を及ぼします。このため，グローバルバッファプールのヒット率が高くなるように指定する必要があります。ここでは，指定したグローバルバッファプールに対するバッファヒット率を向上させるため，検討する項目について説明します。

なお，ここで説明する項目は，データベース操作に関する HiRDB ファイルの情報と一緒に検討すると，より効率良く性能向上ができます。

取得する情報

「グローバルバッファプールに関する統計情報」を取得してください。「グローバルバッファプールに関する統計情報」は，次に示すどちらかの方法で取得できます。

- pdbufls コマンド
- 統計解析ユティリティ

23.1.1　pdbufls コマンドで統計情報を取得した場合

(1)　グローバルバッファプールのヒット率（HIT）を確認してください

確認する目的

この情報は，グローバルバッファプールのバッファ面数が適切かどうかを判断するために取得します。グローバルバッファプールのチューニングでは，最も大切な項目です。

解析結果の妥当性を判断する方法

グローバルバッファプールのヒット率（HIT）が 80%以上であるかどうかを確認してください。

対策方法

●グローバルバッファプールのヒット率が 80%未満の場合

pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。

●グローバルバッファプールのヒット率が 80%以上の場合

特に問題はありません。さらに，ヒット率を向上するには，グローバルバッファの面数を増やしてください。

(2)　更新要求ヒット率（HIT（UPD））を確認してください

確認する目的

これらの情報は，グローバルバッファの面数や HiRDB の内部で処理する遅延書き込み処理（デファードライト）の妥当性を判断するために取得します。

データベースに対してデータの挿入，更新，及び削除を要求した場合，グローバルバッファプールのデータを更新します。このため，データを更新する更新 GET 回数に対して，グローバルバッファプールで何回データを見付けられたかが，スループットに影響を及ぼします。

解析結果の妥当性を判断する方法

統計解析ユティリティで出力した UAP に関する情報の，INSERT，UPDATE，及び DELETE 文発行回数の合計に比べて，更新要求ヒット率が高いか低いかを判断してください。

1. 更新要求ヒット率は低くないか確認してください。
2. 更新要求ヒット率は高いが，参照バッファヒット率が極端に低くなっていないか確認してください。

なお，更新要求ヒット率は，次の計算式で求めています。

更新要求ヒット率（%）＝（更新 GET 要求のヒット回数÷更新 GET 回数）×100

対策方法

●更新要求ヒット率が低い場合

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. pdbuffer オペランドの指定値を見直してください。一つのグローバルバッファに複数の RD エリアを割り当てている場合は，更新する表と参照する表を別々の RD エリアに分け，それぞれにグローバルバッファを割り当ててください。一つのグローバルバッファに一つの RD エリアを割り当てている場合は，インデクスと表を別々の RD エリアに格納して，インデクス専用のグローバルバッファを割り当ててください。又は，表を横分割してください。
3. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を小さくしてください。

●更新要求ヒット率が高い場合

特に問題はありません。

(3) 参照要求ヒット率（HIT（REF））を確認してください

確認する目的

これらの情報は，参照要求ヒット率の妥当性を判断するために取得します。必要とするデータがグローバルバッファにない場合，システムはデータベースのデータを RD エリアからバッファに取り出してきます。このため，グローバルバッファプールを確保した領域に，該当するデータが見付かる回数が少ない場合，入出力回数が増加し，トランザクションの処理性能に影響を及ぼします。

解析結果の妥当性を判断する方法

参照要求ヒット率が 80%に近いかどうかで判断してください。なお，参照要求ヒット率は次に示す計算式で求めています。

参照要求ヒット率（%）＝（参照要求ヒット回数÷参照 GET 回数）×100

対策方法

●参照要求ヒット率が 80%を超えていない場合

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. pdbuffer オペランドの指定値を見直してください。一つのグローバルバッファに複数の RD エリアを割り当てている場合は，特にアクセス頻度が高い表は，一つのグローバルバッファに割り当ててください。一つのグローバルバッファに一つの RD エリアを割り当てている場合は，表を横分割してください。
3. 更新要求ヒット率に対して極端に参照要求ヒット率が低く，参照ページフラッシュ回数が多い場合は，システム共通定義で pd_dbbuff_lru_option=MIX を指定してください。
4. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。

●参照要求ヒット率が 80%を超えている場合

特に問題はありません。

(4) 更新バッファフラッシュ回数（UPFLS）を確認してください

確認する目的

この情報は，更新バッファフラッシュ回数の妥当性を判断するために取得します。処理件数に対して，更新されたバッファを HiRDB ファイルに出力してから新しいページを入力するための空きバッファを作成した回数（更新バッファフラッシュ回数）が多くなると，ディスクへの入出力回数が増加します。このため，トランザクションの処理性能に影響を及ぼします。

解析結果の妥当性を判断する方法

次に示す要因で更新したグローバルバッファが増加し，更新バッファフラッシュ回数や，同じデータを再度取り込むためのバッファ回数（参照 GET 回数）が増加していると考えられます。

- グローバルバッファプールの面数が少なくなっています。
- 処理件数中に占める更新比率が高くなっています。

このため，処理件数に対して，更新バッファフラッシュ回数が多いか少ないかを判断してください。

対策方法

●更新バッファフラッシュ回数が多い場合

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. pdbuffer オペランドの指定値を見直してください。一つのグローバルバッファに複数の RD エリアを割り当てている場合は，更新する表と参照する表を別々の RD エリアに分け，それぞれにグローバルバッファを割り当ててください。一つのグローバルバッファに一つの RD エリアを割り

当てている場合は，インデクスと表を別々の RD エリアに格納して，インデクス専用のグローバルバッファを割り当ててください。又は，表を横分割してください。

3. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。
4. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，デファードライトトリガの要求比率（pd_dbbuff_rate_updpage オペランド）の値を小さくします。

●更新バッファフラッシュ回数が少ない場合

特に問題はありません。

(5) 参照バッファフラッシュ回数（RFFLS）を確認してください

確認する目的

この情報は，参照バッファフラッシュ回数の妥当性を判断するために取得します。処理件数に対して，参照されたバッファの内容を無効にしてから，新しいページを入力するための空きバッファを作成した回数（参照バッファフラッシュ回数）が多くなると，入力回数が増加し，トランザクションの処理性能に影響を及ぼします。

解析結果の妥当性を判断する方法

●処理件数に対して参照バッファフラッシュ回数が多い場合

次に示す要因で，参照バッファ数が増加していると考えられます。

- グローバルバッファプールの面数が少ないです。
- 処理件数中に占める参照の比率が高くなっています。

このため，メモリ上で無効にするバッファや，同じデータを再度取り込むためのバッファが頻繁に発生しています。

●参照要求ヒット率が低く，参照バッファフラッシュ回数が多い場合

次に示す要因で，更新バッファ数が増加していると考えられます。

- グローバルバッファプールの面数が少ないです。
- 処理件数中に占める更新の比率が高くなっています（pd_dbbuff_lru_option ＝ SEPARATE を指定しているか，又はこのオペランドを省略している場合）。

更新するバッファが増加すると，参照要求ヒット率が低下します。このため，メモリ上で無効になるバッファや，同じデータを再度取り込むための入出力が頻繁に発生しています。

対策方法

●参照バッファフラッシュ回数が多い場合

参照要求ヒット率が高い場合は，特に問題ありません。参照要求ヒット率が低い場合は，次に示す処置をしてください。

- pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
- pd_dbbuff_lru_option オペランドに MIX を指定してください。

●参照バッファフラッシュ回数が少ない場合

特に問題はありません。

(6)　実 READ 回数（READ），実 WRITE 回数（WRITE）を確認してください

確認する目的

この情報は，ディスクへの入出力回数の妥当性を判断するために取得します。

処理する件数に対して，ディスクへの入出力回数（HiRDB ファイルに対する入出力回数）が多いと，トランザクションの処理性能及びスループットに影響を及ぼします。なお，データベースアクセスに関する HiRDB ファイルの統計から，各 RD エリアの情報と一緒に検討すると，効率良く入出力回数を削減できます。

解析結果の妥当性を判断する方法

●入力回数が多い場合

1. 同じグローバルバッファを使用する RD エリアの数が多く，各々の RD エリアに対するアクセス頻度が高い状態のときは，バッファヒット率を高くできないために，入力が頻繁に発生していると考えられます。
2. 参照要求ヒット率が低いときは，次に示す要因によってページ入力が頻繁に発生していると考えられます。

- グローバルバッファプールの面数が少ないです（pd_dbbuff_lru_option ＝ MIX を指定している場合）。
- 処理件数中に占める更新の比率が高いため，参照バッファを圧迫しています（pd_dbbuff_lru_option ＝ SEPARATE を指定しているか，又はこのオペランドを省略している場合）。

●出力回数が多い場合

1. 参照要求ヒット率よりも更新バッファヒット率を重視している場合は，該当するグローバルバッファに対する処理要求は，更新が多いと考えられます。
2. 更新 GET 回数が参照 GET 回数よりも多い場合は，「更新バッファフラッシュ回数（UPFLS）を確認してください」で示した更新バッファフラッシュによる影響が出ていると考えられます。

対策方法

●参照要求ヒット率が低いときに，入力回数が多い場合

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. pdbuffer オペランドの指定値を見直してください。一つのグローバルバッファに複数の RD エリアを割り当てている場合は，特にアクセス頻度の高い表には，一つのグローバルバッファプール

を割り当ててください。インデクスと表を同じグローバルバッファプールに割り当てている場合は，インデクス専用のグローバルバッファプールを割り当ててください。一つのグローバルバッファに一つの RD エリアを割り当てている場合は，表を横分割してください。

3. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。
4. pd_dbbuff_lru_option オペランドに MIX を指定してください。

●出力回数が多い場合

1. **pd_dbsync_point ＝ commit を指定している場合**

   更新要求ヒット率が高い場合は特に問題ありません。低い場合は，pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。

2. **pd_dbsync_point ＝ sync を指定しているか，又はこのオペランドを省略した場合**

- pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
- デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。また，シンクポイントの間隔を長くしてください。

(7) バッファ排他待ち回数（WAITL）を確認してください

確認する目的

複数のトランザクションから，グローバルバッファ上の同一ページに対する次のアクセス競合が発生した場合，後続の要求では先行要求の完了を待つバッファ排他待ちが発生します。

- 更新要求を実行中のページに対して，他トランザクションから後続の参照要求又は更新要求が発生したとき
- 参照要求を実行中のページに対して，他トランザクションから後続の更新要求が発生したとき

バッファ排他待ち回数が増加すると，トランザクションの処理時間が長くなります。

解析結果の妥当性を判断する方法

複数のトランザクションから同じページに対して参照と更新，又は複数の更新が発生したときに値が多くなります。次のどちらかに該当する場合は，バッファ排他待ち回数を減らせる可能性があります。

- 複数のトランザクションから，同時に同じ表の同じ RD エリアにデータ挿入している場合
- SQL のアクセスパスがテーブルスキャンや，インデクスの全範囲をサーチする FULL SCAN になっていることなどで検索範囲が広くなっている場合

対策方法

- 複数のトランザクションから，同時に同じ表の同じ RD エリアにデータ挿入している場合

  表を横分割して，データ挿入先が複数の RD エリアに分散されるようにしてください。表の横分割については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

- SQL のアクセスパスがテーブルスキャンや，インデクスの全範囲をサーチする FULL SCAN になっていることなどで検索範囲が広くなっている場合

  検索範囲が広くなっていることで，他のトランザクションとアクセスページが重なっていると考えられます。

  適切なインデクスを定義するなどのチューニングを行って検索範囲を減らしてください。チューニングについては次を参照してください。

  - 「SQL のチューニング方法」の「SQL のチューニングをする」
  - マニュアル「HiRDB パフォーマンスガイド」の「SQL 文が決まったら」

(8) バッファ不足発生回数（INSB）を確認してください

確認する目的

これらの情報はグローバルバッファプールのバッファ面数の妥当性を判断するために取得します。HiRDB では，アクセスするページを読み込むために使用するグローバルバッファがすべて使用中の場合，トランザクションを無効にします。このため，バッファ面数が少ないと，トランザクション集中時に実行できないトランザクションが発生するおそれがあります。

解析結果の妥当性を判断する方法

次に示す項目に該当する場合は，バッファ面数が少ないと判断できます。

- バッファ不足発生回数が 0 以外の場合

対策方法

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. 特に，入出力の集中する RD エリアは，一つのグローバルバッファプールを複数の RD エリアで共用するのをやめて，専用のグローバルバッファプールを割り当ててください。

(9) プリフェッチ入力ページ数（PRRED），プリフェッチヒット率（PRHIT）を確認してください

確認する目的

これらの情報は，プリフェッチ機能の妥当性を判断するために取得します。全件検索や範囲検索などの場合，プリフェッチ機能を使用することで処理時間が短縮できます。プリフェッチ機能は，大量検索をする場合に処理時間の短縮に影響を及ぼします。

解析結果の妥当性を判断する方法

プリフェッチヒット率が 8 割以下の場合，検索対象表のデータ又はインデクスの格納ページが不連続であると考えられます。

対策方法

検索対象表の再編成をしてください。表の再編成ができない状況で，極端にヒット率が悪い場合は，プリフェッチ機能の使用をやめてください（pdbuffer オペランドの-m オプションを省略するか，又は-m オプションに 0 を指定してください）。

(10) プリフェッチバッファ不足発生回数（PRINS）を確認してください

確認する目的

この情報は，プリフェッチ機能の妥当性を判断するために取得します。プリフェッチのためのバッファが不足すると，一括入力ができないユーザはプリフェッチの効果が得られません。

解析結果の妥当性を判断する方法

プリフェッチバッファ不足発生回数が 0 の場合は問題ありません。0 以外の場合は，同時実行最大プリフェッチ数が pdbuffer オペランドの-m オプションの値よりも多いことが考えられます。

対策方法

pdbuffer オペランドの-m オプションの値（同時実行最大プリフェッチ数）を大きくしてください。ただし，この指定を大きくするとグローバルバッファプール用共用メモリが増加するため，メモリ所要量を見積もり直してください。

23.1.2 統計解析ユティリティで統計情報を取得した場合

(1) グローバルバッファプールのヒット率（GBHIT）を確認してください

確認する目的

この情報は，グローバルバッファプールのバッファ面数が適切かどうかを判断するために取得します。グローバルバッファプールのチューニングでは，最も大切な項目です。

解析結果の妥当性を判断する方法

グローバルバッファプールのヒット率（GBHIT）が 80%以上であるかどうかを確認してください。

対策方法

●グローバルバッファプールのヒット率が 80%未満の場合

pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。

●グローバルバッファプールのヒット率が 80%以上の場合

特に問題はありません。さらに，ヒット率を向上するには，グローバルバッファの面数を増やしてください。

(2) 更新バッファフラッシュ回数（UPFLS）を確認してください

確認する目的

この情報は，更新バッファフラッシュ回数の妥当性を判断するために取得します。処理件数に対して，更新されたバッファを HiRDB ファイルに出力してから新しいページを入力するための空きバッファを作成した回数（更新バッファフラッシュ回数）が多くなると，ディスクへの入出力回数が増加します。このため，トランザクションの処理性能に影響を及ぼします。

解析結果の妥当性を判断する方法

次に示す要因で更新したグローバルバッファが増加し，更新バッファフラッシュ回数や，同じデータを再度取り込むためのバッファ回数（参照 GET 回数）が増加していると考えられます。

- グローバルバッファプールの面数が少なくなっています。
- 処理件数中に占める更新比率が高くなっています。

このため，処理件数に対して，更新バッファフラッシュ回数が多いか少ないかを判断してください。

対策方法

●更新バッファフラッシュ回数が多い場合

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. pdbuffer オペランドの指定値を見直してください。一つのグローバルバッファに複数の RD エリアを割り当てている場合は，更新する表と参照する表を別々の RD エリアに分け，それぞれにグローバルバッファを割り当ててください。一つのグローバルバッファに一つの RD エリアを割り当てている場合は，インデクスと表を別々の RD エリアに格納して，インデクス専用のグローバルバッファを割り当ててください。又は，表を横分割してください。
3. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。
4. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，デファードライトトリガの要求比率（pd_dbbuff_rate_updpage オペランド）の値を小さくします。

●更新バッファフラッシュ回数が少ない場合

特に問題はありません。

(3) 参照バッファフラッシュ回数（RFFLS）を確認してください

確認する目的

この情報は，参照バッファフラッシュ回数の妥当性を判断するために取得します。処理件数に対して，参照されたバッファの内容を無効にしてから，新しいページを入力するための空きバッファを作成した回数（参照バッファフラッシュ回数）が多くなると，入力回数が増加し，トランザクションの処理性能に影響を及ぼします。

解析結果の妥当性を判断する方法

●処理件数に対して参照バッファフラッシュ回数が多い場合

次に示す要因で，参照バッファ数が増加していると考えられます。

- グローバルバッファプールの面数が少ないです。
- 処理件数中に占める参照の比率が高くなっています。

このため，メモリ上で無効にするバッファや，同じデータを再度取り込むためのバッファが頻繁に発生しています。

●参照要求ヒット率が低く，参照バッファフラッシュ回数が多い場合

次に示す要因で，更新バッファ数が増加していると考えられます。

- グローバルバッファプールの面数が少ないです。
- 処理件数中に占める更新の比率が高くなっています（pd_dbbuff_lru_option ＝ SEPARATE を指定しているか，又はこのオペランドを省略している場合）。

更新するバッファが増加すると，参照要求ヒット率が低下します。このため，メモリ上で無効になるバッファや，同じデータを再度取り込むための入出力が頻繁に発生しています。

対策方法

●参照バッファフラッシュ回数が多い場合

参照要求ヒット率が高い場合は，特に問題ありません。参照要求ヒット率が低い場合は，次に示す処置をしてください。

- pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
- pd_dbbuff_lru_option オペランドに MIX を指定してください。

●参照バッファフラッシュ回数が少ない場合

特に問題はありません。

(4) 実 READ 回数（READ），実 WRITE 回数（WRITE）を確認してください

確認する目的

この情報は，ディスクへの入出力回数の妥当性を判断するために取得します。

処理する件数に対して，ディスクへの入出力回数（HiRDB ファイルに対する入出力回数）が多いと，トランザクションの処理性能及びスループットに影響を及ぼします。なお，データベースアクセスに関する HiRDB ファイルの統計から，各 RD エリアの情報と一緒に検討すると，効率良く入出力回数を削減できます。

解析結果の妥当性を判断する方法

●入力回数が多い場合

1. 同じグローバルバッファを使用する RD エリアの数が多く，各々の RD エリアに対するアクセス頻度が高い状態のときは，バッファヒット率を高くできないために，入力が頻繁に発生していると考えられます。
2. 参照要求ヒット率が低いときは，次に示す要因によってページ入力が頻繁に発生していると考えられます。

- グローバルバッファプールの面数が少ないです（pd_dbbuff_lru_option ＝ MIX を指定している場合）。
- 処理件数中に占める更新の比率が高いため，参照バッファを圧迫しています（pd_dbbuff_lru_option ＝ SEPARATE を指定しているか，又はこのオペランドを省略している場合）。

●出力回数が多い場合

1. 参照要求ヒット率よりも更新バッファヒット率を重視している場合は，該当するグローバルバッファに対する処理要求は，更新が多いと考えられます。
2. 更新 GET 回数が参照 GET 回数よりも多い場合は，「更新バッファフラッシュ回数（UPFLS）を確認してください」で示した更新バッファフラッシュによる影響が出ていると考えられます。

対策方法

●参照要求ヒット率が低いときに，入力回数が多い場合

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. pdbuffer オペランドの指定値を見直してください。一つのグローバルバッファに複数の RD エリアを割り当てている場合は，特にアクセス頻度の高い表には，一つのグローバルバッファプールを割り当ててください。インデクスと表を同じグローバルバッファプールに割り当てている場合は，インデクス専用のグローバルバッファプールを割り当ててください。一つのグローバルバッファに一つの RD エリアを割り当てている場合は，表を横分割してください。
3. デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。
4. pd_dbbuff_lru_option オペランドに MIX を指定してください。

●出力回数が多い場合

1. **pd_dbsync_point ＝ commit を指定している場合**

   更新要求ヒット率が高い場合は特に問題ありません。低い場合は，pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. **pd_dbsync_point ＝ sync を指定しているか，又はこのオペランドを省略した場合**

- pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
- デファードライト処理をする場合（pd_dbsync_point オペランドに sync を指定するか又は省略した場合）は，pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。また，シンクポイントの間隔を長くしてください。

(5) 出力待ち発生回数（WAITW）を確認してください

確認する目的

RD エリアにグローバルバッファを割り当てるときの妥当性を判断するために取得します。同一ページに対するアクセス要求が競合した場合，次に示す要因でデータの入出力処理に関する待ちが発生します。

- HiRDB ファイルに出力するときに，更新要求が競合すると，HiRDB ファイルへの出力完了待ちが発生します。

解析結果の妥当性を判断する方法

次に示す要因で，出力待ち発生回数が多くなっていることが考えられます。

1. 該当するページに対して，デファードライト処理で出力する比率が高いために，HiRDB ファイルへの出力処理と競合しています。
2. データベース作成ユティリティのデータロードやデータベース再編成ユティリティのリロード処理で，グローバルバッファプールを大量に使用しているため，グローバルバッファプール中の更新バッファ数が増加しています。

対策方法

1. 大量のデータを持つ表の場合は，表又はインデクスの分割を検討してください。
2. 一つ又は複数の RD エリアを構成する HiRDB ファイルを別々のディスクに分けてください。
3. データベース作成ユティリティのデータロードやデータベース再編成ユティリティのリロード処理で大量のデータを使用する場合，一括出力ページ数（–n オプション）を使用してください。なお，データベース再編成ユティリティを適用するときは，該当するデータが FIX 属性の表のときです。
4. 出力待ち発生回数が多い場合でデファードライト処理をするとき（pd_dbsync_point オペランドに sync を指定するか又は省略したとき）は，デファードライトトリガの要求比率（pd_dbbuff_rate_updpage オペランド）の値を小さくします。

(6) バッファ排他待ち回数（WAITL）を確認してください

確認する目的

複数のトランザクションから，グローバルバッファ上の同一ページに対する次のアクセス競合が発生した場合，後続の要求では先行要求の完了を待つバッファ排他待ちが発生します。

- 更新要求を実行中のページに対して，他トランザクションから後続の参照要求又は更新要求が発生したとき
- 参照要求を実行中のページに対して，他トランザクションから後続の更新要求が発生したとき

バッファ排他待ち回数が増加すると，トランザクションの処理時間が長くなります。

解析結果の妥当性を判断する方法

複数のトランザクションから同じページに対して参照と更新，又は複数の更新が発生したときに値が多くなります。次のどちらかに該当する場合は，バッファ排他待ち回数を減らせる可能性があります。

- 複数のトランザクションから，同時に同じ表の同じ RD エリアにデータ挿入している場合
- SQL のアクセスパスがテーブルスキャンや，インデクスの全範囲をサーチする FULL SCAN になっていることなどで検索範囲が広くなっている場合

対策方法

- 複数のトランザクションから，同時に同じ表の同じ RD エリアにデータ挿入している場合

  表を横分割して，データ挿入先が複数の RD エリアに分散されるようにしてください。表の横分割については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
- SQL のアクセスパスがテーブルスキャンや，インデクスの全範囲をサーチする FULL SCAN になっていることなどで検索範囲が広くなっている場合

  検索範囲が広くなっていることで，他のトランザクションとアクセスページが重なっていると考えられます。

  適切なインデクスを定義するなどのチューニングを行って検索範囲を減らしてください。チューニングについては次を参照してください。
  - 「SQL のチューニング方法」の「SQL のチューニングをする」
  - マニュアル「HiRDB パフォーマンスガイド」の「SQL 文が決まったら」

(7) 最大同時要求バッファ面数（MAXB），バッファ不足発生回数（BFINS）を確認してください

確認する目的

これらの情報はグローバルバッファプールのバッファ面数の妥当性を判断するために取得します。HiRDB では，アクセスするページを読み込むために使用するグローバルバッファがすべて使用中の場合，トランザクションを無効にします。このため，バッファ面数が少ないと，トランザクション集中時に実行できないトランザクションが発生するおそれがあります。

解析結果の妥当性を判断する方法

次に示す項目に該当する場合は，バッファ面数が少ないと判断できます。

1. 最大同時要求バッファ面数が，グローバルバッファ面数に近い値又は同じ値になっている場合に，バッファ不足発生回数が 0 のとき（ただし，参照バッファヒット率や更新バッファヒット率が高い場合を除きます。参照バッファヒット率や更新バッファヒット率が高いときは，最大限の性能を引き出していると考えられます）
2. 最大同時要求バッファ面数が，グローバルバッファ面数に近い値又は同じ値になっている場合に，バッファ不足発生回数が多いとき
3. バッファ不足発生回数が 0 以外の場合

対策方法

1. pdbuffer オペランドの-n オプション（グローバルバッファのバッファ面数）の値を大きくしてください。
2. 特に，入出力の集中する RD エリアは，一つのグローバルバッファプールを複数の RD エリアで共用するのをやめて，専用のグローバルバッファプールを割り当ててください。

(8) プリフェッチ入力ページ数（PRRED），プリフェッチヒット回数（PRHIT），プリフェッチヒット率（HIT）を確認してください

確認する目的

これらの情報は，プリフェッチ機能の妥当性を判断するために取得します。全件検索や範囲検索などの場合，プリフェッチ機能を使用することで処理時間が短縮できます。プリフェッチ機能は，大量検索をする場合に処理時間の短縮に影響を及ぼします。

解析結果の妥当性を判断する方法

プリフェッチヒット率が８割以下の場合，検索対象表のデータ又はインデクスの格納ページが不連続であると考えられます。

対策方法

検索対象表の再編成をしてください。表の再編成ができない状況で，極端にヒット率が悪い場合は，プリフェッチ機能の使用をやめてください（pdbuffer オペランドの-m オプションを省略するか，又は-m オプションに０を指定してください）。

(9) プリフェッチバッファ不足発生回数（PRINS）を確認してください

確認する目的

この情報は，プリフェッチ機能の妥当性を判断するために取得します。プリフェッチのためのバッファが不足すると，一括入力ができないユーザはプリフェッチの効果が得られません。

解析結果の妥当性を判断する方法

プリフェッチバッファ不足発生回数が０の場合は問題ありません。０以外の場合は，同時実行最大プリフェッチ数が pdbuffer オペランドの-m オプションの値よりも多いことが考えられます。

対策方法

pdbuffer オペランドの-m オプションの値（同時実行最大プリフェッチ数）を大きくしてください。ただし，この指定を大きくするとグローバルバッファプール用共用メモリが増加するため，メモリ所要量を見積もり直してください。

(10) シンクポイント出力ページ数（SYNCW）を確認してください

確認する目的

この情報はシンクポイントダンプの出力間隔の妥当性を判断するために取得します。通常，シンクポイント時にグローバルバッファプールのすべての更新ページをデータベースに出力するため，トランザク

ションの性能が低下する要因になります。これを防止するため，HiRDB では内部処理（プレシンク処理とデファードライト処理）で，シンクポイント時の出力ページ数を削減しています。このため，グローバルバッファをチューニングすると，シンクポイントダンプの出力間隔に影響を及ぼします。

解析結果の妥当性を判断する方法

シンクポイント時の出力ページ数が，グローバルバッファ面数の 50%よりも多くなっていないか確認してください。

対策方法

pdbuffer オペランドの-w オプションの値（デファードライト処理で出力するページの比率）を大きくしてください。

(11) グローバルバッファの排他処理の排他競合待ち発生率（SLEPR）を確認してください

確認する目的

同時実行するトランザクション数が増加すると，同一のグローバルバッファプールにアクセスする処理で競合し，排他待ちが発生します。SLEPR を確認することでグローバルバッファの排他獲得処理がどの程度競合しているかが分かります。競合度が高いとグローバルバッファの排他獲得処理の処理待ちが発生する可能性が高く，性能低下の一因になります。

解析結果の妥当性を判断する方法

「グローバルバッファの排他処理の排他競合待ち発生率（SLEPR）」の値が 1 以上の場合は，グローバルバッファの排他獲得処理の競合度が高いと判断できます。

対策方法

グローバルバッファの割り当て方を見直し，以下のどれかの対策をしてください。

1. 一つのグローバルバッファプールに複数の RD エリアを割り当てている場合，一つのグローバルバッファプールに割り当てる RD エリアの数を少なくしてください。アクセス頻度が高い表やインデクスを格納している RD エリアには専用のグローバルバッファを割り当ててください。
2. 複数の表を一つの RD エリアに格納している場合は，表を複数の RD エリアに分けて格納し，それぞれの RD エリアを別々のグローバルバッファに割り当ててください。表とインデクスを同じ RD エリアに格納している場合，表とインデクスを別々の RD エリアに格納し，インデクス専用のグローバルバッファを割り当ててください。
3. 表に専用の RD エリアとグローバルバッファを割り当てている場合は，表を横分割して各 RD エリアに専用のグローバルバッファを割り当ててください。
4. システム共通定義のグローバルバッファプールの分割（pdbuffer -D オプション）を検討してください。pdbuffer オペランドに「-D 分割数」を指定すると，グローバルバッファへのアクセスを分散できます。

(12) スリープ処理実行での，バッファの排他処理のスリープ処理実行回数の平均値（SLEPA）を確認してください

確認する目的

ジョブの多重実行中に，極端に処理時間が長いジョブがある場合，グローバルバッファの排他獲得処理の遅延によるものなのかどうかを判断できます。

解析結果の妥当性を判断する方法

次の条件を満たす場合，グローバルバッファの排他獲得処理時間を短縮できる可能性があります。

- 「スリープ処理実行での，バッファの排他処理のスリープ処理実行回数の平均値（SLEPA）」の値が 1 以上 2 未満
- スリープ処理を 1 回以上実行した場合と，1 回も実行しなかった場合とで，グローバルバッファの排他獲得処理の処理時間に差がない

対策方法

グローバルバッファの割り当て方を見直してください。

1. 一つのグローバルバッファプールに複数の RD エリアを割り当てている場合，一つのグローバルバッファプールに割り当てる RD エリアの数を少なくしてください。アクセス頻度が高い表やインデクスを格納している RD エリアには専用のグローバルバッファを割り当ててください。
2. 複数の表を一つの RD エリアに格納している場合は，表を複数の RD エリアに分けて格納し，それぞれの RD エリアを別々のグローバルバッファに割り当ててください。表とインデクスを同じ RD エリアに格納している場合，表とインデクスを別々の RD エリアに格納し，インデクス専用のグローバルバッファを割り当ててください。
3. 表に専用の RD エリアとグローバルバッファを割り当てている場合は，表を横分割して各 RD エリアに専用のグローバルバッファを割り当ててください。

(13) シンクポイント時の参照要求ヒットによるデータベース書き込み処理代行回数（ALTRW）を確認してください

確認する目的

シンクポイント時に参照トランザクションの沈み込みが発生している場合，シンクポイントの対象となるバッファの参照要求のバッファヒット時に，トランザクションの延長で代行出力していることが考えられます。

解析結果の妥当性を判断する方法

シンクポイント時に参照トランザクションの沈み込みが発生している場合，「シンクポイント時の参照要求ヒットによるデータベース書き込み処理代行回数（ALTRW）」が，沈み込みの要因になっていないかどうかを確認してください。

対策方法

pd_dbsync_altwrite_skip オペランドに Y を指定すると，シンクポイント時の参照要求ヒットによるデータベース書き込み処理を抑止できます。ただし，この場合，シンクポイント取得処理時間が長くな

るため，デファードライト書き込み処理の並列 WRITE 機能を使用して，書き込み処理に掛かる負荷を分散させることをお勧めします。デファードライト書き込み処理の並列 WRITE 機能については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(14) シンクポイント時の更新要求ヒットによるデータベース書き込み処理代行回数（ALTUW）を確認してください

確認する目的

シンクポイント取得処理中に更新トランザクションの沈み込みが発生している場合，シンクポイントの対象となるバッファの更新要求のバッファヒット時に，トランザクションの延長で代行出力していることが考えられます。

解析結果の妥当性を判断する方法

シンクポイント取得処理中（KFPS02183-I メッセージの start time と end time の間の時間）にトランザクションの沈み込みが発生している場合，「シンクポイント時の更新要求ヒットによるデータベース書き込み処理代行回数（ALTUW）」が，沈み込みの要因になっていないかどうかを確認してください。妥当性についてはデファードライト処理に関する統計情報の出力ページ総数（OUT PAGE）に対する，シンクポイント時の更新要求ヒットによるデータベース書き込み処理代行回数（ALTUW）の割合で判断してください。

対策方法

pd_dbsync_altwrite_skip オペランドに ALL を指定すると，シンクポイント時の更新要求ヒットによるデータベース書き込み処理を抑止できます。ただし，この場合，シンクポイント取得処理時間が長くなるため，デファードライト書き込み処理の並列 WRITE 機能を使用して，書き込み処理に掛かる負荷を分散させることをお勧めします。デファードライト書き込み処理の並列 WRITE 機能については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

23.2　デファードライト処理のチューニング

23.2.1　デファードライト処理のチューニングで確認すること

システム共通定義の **pd_dbsync_point** オペランドに sync を指定するか，又は省略すると，HiRDB はデファードライト処理を実行し，ディスクへの出力回数を削減しています。デファードライト処理の特長は，RD エリアを構成する複数のディスクを意識して，ディスク単位に処理することで性能を向上させています。

このため，RD エリアを割り当てる HiRDB ファイルシステム領域に raw I/O 機能を適用し，かつ複数ディスクに割り当てていないと効果がありません。

取得する情報

統計解析ユティリティの「デファードライト処理に関する統計情報」を取得してください。

(1)　ディスクボリューム単位の並列度（PMAX，PMIN）を確認してください

確認する目的

デファードライト処理で出力先となったディスクボリューム（HiRDB ファイルシステム領域）の分散度から，更新対象 RD エリアに対するディスクボリューム（HiRDB ファイルシステム領域）の割り当てが妥当かどうか判断するために取得します。

解析結果の妥当性を判断する方法

NTFS 以外に HiRDB ファイルシステム領域を作成している場合，ディスクボリューム単位の並列度の最大値及び最小値が 1 になっていないか確認してください。1 の場合は特定のディスクボリューム（HiRDB ファイルシステム領域）にだけ出力が集中していると考えられます。また，デファードライト処理の並列 write 機能使用時は最大値が pd_dfw_awt_process 指定値以上であるか確認してください。

対策方法

1. 更新が発生する RD エリアを，それぞれ異なるディスクボリューム（HiRDB ファイルシステム領域）に割り当ててください。
2. 更新が発生する表及びインデクスを，それぞれ異なるディスクボリューム（HiRDB ファイルシステム領域）から構成される RD エリアに分けて格納してください。

(2)　平均値（AVG）を確認してください

確認する目的

次の処理で出力される更新ページの平均値を検討して，グローバルバッファの妥当性を判断するために取得します。

●トリガ出力

グローバルバッファ中の更新ページ数が，バッファ面数のある一定に達した時点でディスクへ出力します。

ただし，デファードライトトリガの要求比率を有効にした場合，その指定値に達した時点でディスクへ出力します。この指定値の設定は pd_dbbuff_rate_updpage オペランドで指定できます。

このとき，ディスクへ出力するページ数は，pdbuffer オペランドの-w オプションで指定した出力ページ比率を基に HiRDB が計算して出力しています。

●プレシンク出力

シンクポイントでディスクへ出力するページ数を削減するため，プレシンクというポイントを内部的に設定しています。プレシンクとは，シンクポイントが発生するまでにグローバルバッファで更新したページをディスクに書き込み，シンクポイント時に出力する更新ページを少なくする処理です。

●シンクポイントダンプ出力

シンクポイント時，グローバルバッファにあるすべての更新ページをディスクへ出力します。プレシンクの出力で，シンクポイントダンプの出力処理に掛かる時間を大幅に削減できます。

●データベースのシンクポイントダンプ出力

シンクポイントダンプ処理を有効化できないシンクポイント契機で更新バッファをデータベースに反映する処理です。

●RD エリアのシンクポイントダンプ出力

ある RD エリアに対応するすべての更新ページをディスクへ出力する処理です。

解析結果の妥当性を判断する方法

●出力ページ数の平均値が極端に少ない場合

デファードライト処理で発生したページの出力回数よりも，通常のバッファフラッシュ回数が多くないかを確認してください。なお，確認するときは，更新頻度が少ないグローバルバッファプールで検討してください。

1. 更新ページバッファヒット率が高い場合，デファードライト処理で出力するページ数が少ないときは，問題がないと考えられます。
2. 更新ページバッファヒット率が低い場合，デファードライト処理で出力するページ比率が低いときは，シンクポイントダンプの取得間隔を短くできると考えられます。

なお，出力ページ数の平均は，次に示す計算式で求めてください。

$$\text{平均値（AVG）} = \left(a - \sum_{i=1}^{n} b_i\right) \div (c - d)$$

ａ：出力ページ総数（OUT PAGE）

b_i：シンクポイント時の出力ページ数（SYNCW）

ｃ：デファードライト処理の起動回数（EXEC）

ｄ：シンクポイント発生回数（SYNC）

ｎ：グローバルバッファプールの定義数

●HiRDB/パラレルサーバの場合

特定のバックエンドサーバの出力ページ総数が多い時は，次に示す要因が考えられます。

1. 更新処理が多い表を特定のバックエンドサーバに格納しています。
2. 分割した表のうち，特定のキーレンジにだけ更新処理が集中しています。

対策方法

●特定のグローバルバッファに書き込みが集中している場合

1. グローバルバッファプールに割り当てている RD エリアが複数あるときは，別々のグローバルバッファに割り当ててください。
2. 特定のサーバで複数の表を更新している場合，別々のバックエンドサーバに表を分けることを検討してください。

●複数の RD エリアが一つのディスクに定義されている場合

各々の RD エリアを別々のディスクに割り当てて，入出力並列度を向上させてください。

23.3　デファードライト処理適用時のシンクポイント処理時間のチューニング

デファードライト処理を適用している場合，シンクポイントダンプの有効化処理（以降，この節ではシンクポイント処理と表記します）時間が長くなることがあります。ここでは，このシンクポイント処理時間を短縮する方法について説明します。

23.3.1　チューニングの手順

大量更新を行うと，シンクポイント時にデータベースに反映する必要がある更新バッファ数（グローバルバッファ中の更新ページ数）が多くなるため，シンクポイント処理時間が長くなります。そのために，シンクポイント処理が連続してスキップされると，上書きできない状態のシステムログファイルが増加します。その結果，スワップ先にできる状態のファイルがなくなると，HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了します。

これを防ぐために，ここの説明に従って，設計時に想定した時間内にシンクポイント処理を終わらせるようにしてください。なお，次に示す条件式を満たす場合はチューニングを行う必要はありません。

- **シンクポイント取得間隔（時間）＞シンクポイント処理時間**

デファードライト処理の遅延によってシンクポイント処理がスキップされた場合，KFPS02179-I メッセージ（factor code = A01-01）が出力されます。この場合，(1)以降の手順に従ってチューニングを行ってください。

(1)　チューニング情報を取得する

次に示すチューニング情報を取得してください。

1. CPU 及びディスクの入出力に関する情報
2. グローバルバッファに関する統計情報
3. デファードライト処理に関する統計情報

1 については OS の機能を使用して取得してください。2～3 については統計解析ユティリティで取得してください。

(2)　デファードライト処理に関する統計情報を確認する

デファードライト処理に関する統計情報の動作要因（CAUSE）が，S（シンクポイント）の統計情報を参照してください。その結果，次に示すどちらかの条件を満たす場合は(3)に進んでください。満たさない場合は(9)に進んでください。

- **デファードライト処理の並列 WRITE 機能を使用していない場合**
  合計 WRITE 時間（DWSUM，DWSUMM）＞シンクポイント取得間隔（時間）

- **デファードライト処理の並列 WRITE 機能を使用している場合**
  並列 WRITE 時間（DWPARA，DWPARAM）＞シンクポイント取得間隔（時間）

シンクポイント取得間隔は，次に示すメッセージの出力時間の差分から求めてください。

- KFPS02183-I（シンクポイント処理が完了したときに出力されるメッセージ）
- KFPS02179-I（シンクポイント処理をスキップしたときに出力されるメッセージ）

(3)　WRITE 単価平均の値を確認する

デファードライト処理に関する統計情報の WRITE 単価平均（DWAVG，DWAVGM）が，ディスクの書き込み性能に比べて極端に悪い場合，次に示すことを確認してください。該当しない場合は(4)に進んでください。

- ディスク障害が発生していないか確認してください。
- ディスク競合などが考えられます。OS の機能を使用して入出力に関するチューニング情報を取得し，その結果からチューニングを行ってください。チューニングの余地がない場合は(4)に進んでください。

(4)　KFPS02179-I メッセージを確認する

シンクポイント処理をスキップした場合，KFPS02179-I メッセージが出力されます。このメッセージが出力されている時間帯に次に示すユティリティを実行している場合は，運用方法の変更を検討してください。該当しない場合は(5)に進んでください。

- **データベース作成ユティリティ，データベース再編成ユティリティ，リバランスユティリティを実行している場合**
  ユティリティの実行時に-n オプションを指定してローカルバッファを使用してください。ローカルバッファを使用できない場合は(5)に進んでください。
- **空きページ解放ユティリティを実行している場合**
  ユティリティの実行時に-p オプションを指定してください。-p オプションを指定できない場合は(5)に進んでください。

(5)　シンクポイントの取得間隔を見直す

シンクポイントの取得間隔を長くできないか検討してください。長くできない場合は(6)に進んでください。シンクポイントの取得間隔は pd_log_sdinterval オペランドで変更できます。

(6)　デファードライトトリガのチューニングを行う

「デファードライトトリガの実行間隔を短くする」及び「デファードライトトリガ時の更新ページの出力比率を高くする」で説明しているチューニングを行ってください。チューニング後も事態が解消されない場合は(7)に進んでください。

(7)　デファードライト処理の並列 WRITE 機能を使用する

CPU に余裕がある場合は次に示すチューニングを行ってください。余裕がない場合は(8)に進んでください。

- **デファードライト処理の並列 WRITE 機能を使用していない場合**
  pd_dfw_awt_process オペランドを指定してデファードライト処理の並列 WRITE 機能を使用してください。使用後も事態が解消されない場合は(8)に進んでください。
- **デファードライト処理の並列 WRITE 機能を使用している場合**
  「デファードライト処理用並列 WRITE プロセス数を多くする」で説明しているチューニングを行ってください。チューニング後も事態が解消されない場合は(8)に進んでください。

(8)　更新バッファ数の抑制を検討する

pd_dfw_syncpoint_skip_limit オペランドを指定して更新バッファ数を抑制してください。ただし，この場合，シンクポイント処理のスキップ回数が上限に達したとき，更新トランザクションの延長で更新バッファの出力を行うため，更新トランザクション性能が低下します。

pd_dfw_syncpoint_skip_limit オペランドを指定できない場合は，ディスクの増強を検討してください。

(9)　グローバルバッファのチューニングを行う

次に示す現象が発生している場合，グローバルバッファのチューニングを行ってください。

- バッファ排他待ち回数（WAITL）が多発している
- 出力待ち発生回数（WAITW）が多発している
- グローバルバッファの排他処理の排他競合待ち発生率（SLEPR）が高い
- 更新バッファフラッシュ回数（UPFLS）が増加している

前記の情報，及びグローバルバッファのチューニングについては，「グローバルバッファプールのチューニング」を参照してください。

23.3.2　デファードライト処理に関する統計情報の見方

参照する情報

統計解析ユティリティ（pdstedit）で，デファードライト処理に関する統計情報の DAT 形式ファイルを取得し，次に示す情報を参照してください。

- 実行時間（DWTOTAL，DWTOTALM）
- 合計 WRITE 時間（DWSUM，DWSUMM）
- 並列 WRITE 時間（DWPARA，DWPARAM）
- WRITE 単価

最小（DWMIN，DWMINM）

最大（DWMAX，DWMAXM）

平均（DWAVG，DWAVGM）

- WRITE 回数（DWEXEC）

(1) 実行時間（DWTOTAL，DWTOTALM）

デファードライト処理に要した時間の合計です。DWTOTAL は秒単位までの時間を表示し，DWTOTALM は秒よりけた下のマイクロ秒の部分だけを表示します。

確認する目的

デファードライト処理の性能が適切かどうか判断するために確認します。

解析結果の妥当性を判断する方法

デファードライト遅延メッセージ（KFPS02179-I factor code =A01-01）が出力される場合，次に示す対策方法に従ってください。

対策方法

対策方法については，「チューニングの手順」を参照してください。

(2) 合計 WRITE 時間（DWSUM，DWSUMM）

デファードライト処理の中で書き込み処理に要した時間の合計です。DWSUM は秒単位までの時間を表示し，DWSUMM は秒よりけた下のマイクロ秒の部分だけを表示します。

確認する目的

並列 WRITE 時間（DWPARA　DWPARAM）と比較し，デファードライト処理用並列 WRITE プロセス数のチューニング効果が現れているか確認します。

解析結果の妥当性を判断する方法

WRITE 単価（最大）（DWMAX　DWMAXM），WRITE 単価（最小）（DWMIN　DWMINM），及び WRITE 回数（DWEXEC）の値を確認してください。

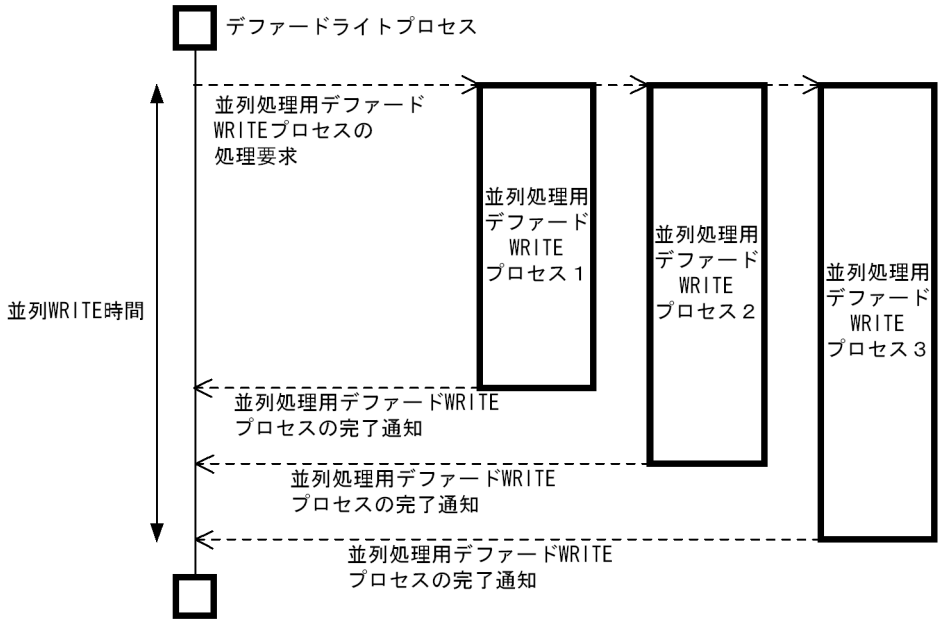
対策方法

WRITE 単価（最大）（DWMAX　DWMAXM），WRITE 単価（最小）（DWMIN　DWMINM），及び WRITE 回数（DWEXEC）の値を確認し，その対策方法に従ってください。

(3) 並列 WRITE 時間（DWPARA，DWPARAM）

複数のデファードライト処理用並列 WRITE プロセスの処理要求を送信してから，すべてのデファードライト処理用並列 WRITE プロセスの完了通知を受信するまでの時間です。DWPARA は秒単位までの時間を表示し，DWPARAM は秒よりけた下のマイクロ秒の部分だけを表示します。並列 WRITE 時間の概念を次の図に示します。

図 23–1　並列 WRITE 時間の概念



なお，デファードライト処理の並列 WRITE 機能を無効にしている場合，並列 WRITE 時間は 0 になります。

確認する目的

デファードライト処理の並列 WRITE 機能の適用効果を確認します。

解析結果の妥当性を判断する方法

次に示す条件式が成立する場合は，デファードライト処理の並列 WRITE 機能の適用効果があると判断できます。

- 実行時間（DWTOTAL ＋ DWTOTALM の値）＞並列 WRITE 時間（DWPARA ＋ DWPARAM の値）

なお，正確に判定するには，次に示すときの実行時間（DWTOTAL ＋ DWTOTALM の値）を計測してください。

- デファードライト処理の並列 WRITE 機能の適用前後
- pd_dfw_awt_process オペランドの指定値の変更前後

実行時間が短縮されている場合は，デファードライト処理の並列 WRITE 機能の適用効果があると判断できます。

対策方法

適用効果がある場合は，「デファードライト処理用並列 WRITE プロセス数を多くする」で説明しているチューニングを行ってください。

適用効果がない場合は次に示すどちらかの処置をしてください。

- デファードライト処理の並列 WRITE 機能の使用を中止する
- pd_dfw_awt_process オペランドの値を変更前の値に戻す

(4) WRITE 単価最小（DWMIN，DWMINM）

複数のページに対して書き込み処理が発生した場合，書き込み処理時間が最も短いページの時間です。DWMIN は秒単位までの時間を表示し，DWMINM は秒よりけた下のマイクロ秒の部分だけを表示します。

確認する目的

ディスクパフォーマンスの妥当性を確認します。

解析結果の妥当性を判断する方法

ディスク性能と比較して判断してください。デファードライト処理の並列 WRITE 機能を使用する場合，及びデファードライト処理用並列 WRITE プロセス数を変更する場合は，使用前後又は変更前後でディスク性能を比較してください。

対策方法

ディスク性能に比べてパフォーマンスが悪い場合は，OS の機能を使用して入出力に関するチューニング情報を取得し，その情報を基にチューニングを行ってください。デファードライト処理の並列 WRITE 機能の使用前後，又はデファードライト処理用並列 WRITE プロセス数の変更前後でディスクのパフォーマンスが低下した場合は，並列 WRITE 時間を確認してください。適用効果がない場合は，使用前又は変更前に戻してください。

ディスク性能を低下させる原因の一つにディスク競合が考えられます。ディスクの構成，RD エリアの構成，表の構成を見直して，特定のボリュームに入出力処理が集中しないようにしてください。ディスクの構成，RD エリアの構成，及び表の構成の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(5) WRITE 単価最大（DWMAX，DWMAXM）

複数のページに対して書き込み処理が発生した場合，書き込み処理時間が最も長いページの時間です。DWMAX は秒単位までの時間を表示し，DWMAXM は秒よりけた下のマイクロ秒の部分だけを表示します。

確認する目的

ディスクパフォーマンスの妥当性を確認します。

解析結果の妥当性を判断する方法

ディスク性能と比較して判断してください。デファードライト処理の並列 WRITE 機能を使用する場合，及びデファードライト処理用並列 WRITE プロセス数を変更する場合は，使用前後又は変更前後でディスク性能を比較してください。

対策方法

ディスク性能に比べてパフォーマンスが悪い場合は，OS の機能を使用して入出力に関するチューニング情報を取得し，その情報を基にチューニングを行ってください。デファードライト処理の並列 WRITE 機能の使用前後，又はデファードライト処理用並列 WRITE プロセス数の変更前後でディスクのパフォーマンスが低下した場合は，並列 WRITE 時間を確認してください。適用効果がない場合は，使用前又は変更前に戻してください。

ディスク性能を低下させる原因の一つにディスク競合が考えられます。ディスクの構成，RD エリアの構成，表の構成を見直して，特定のボリュームに入出力処理が集中しないようにしてください。ディス

クの構成，RD エリアの構成，及び表の構成の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(6) WRITE 単価平均（DWAVG，DWAVGM）

複数のページに対して書き込み処理が発生した場合，書き込み処理に要した時間の平均です。DWAVG は秒単位までの時間を表示し，DWAVGM は秒よりけた下のマイクロ秒の部分だけを表示します。

確認する目的

ディスクパフォーマンスの妥当性を確認します。

解析結果の妥当性を判断する方法

ディスク性能と比較して判断してください。デファードライト処理の並列 WRITE 機能を使用する場合，及びデファードライト処理用並列 WRITE プロセス数を変更する場合は，使用前後又は変更前後でディスク性能を比較してください。

対策方法

ディスク性能に比べてパフォーマンスが悪い場合は，OS の機能を使用して入出力に関するチューニング情報を取得し，その情報を基にチューニングを行ってください。デファードライト処理の並列 WRITE 機能の使用前後，又はデファードライト処理用並列 WRITE プロセス数の変更前後でディスクのパフォーマンスが低下した場合は，並列 WRITE 時間を確認してください。適用効果がない場合は，使用前又は変更前に戻してください。

ディスク性能を低下させる原因の一つにディスク競合が考えられます。ディスクの構成，RD エリアの構成，表の構成を見直して，特定のボリュームに入出力処理が集中しないようにしてください。ディスクの構成，RD エリアの構成，及び表の構成の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(7) WRITE 回数（DWEXEC）

動作要因（CAUSE）ごとの書き込み回数です。

確認する目的

次に示す原因によってデファードライト処理が遅延しているかどうかを確認します。

- シンクポイント取得間隔内で処理可能な更新バッファ数を超えているため

解析結果の妥当性を判断する方法

動作要因（CAUSE）が S（シンクポイント）の場合は，WRITE 回数（DWEXEC）が次に示す計算式の値以下であることを確認してください。計算式の値より大きい場合は対策方法に従って対策してください。

{シンクポイント取得間隔÷WRITE 単価平均（DWAVG，DWAVGN）} ×0.1

シンクポイントの取得間隔は，次に示すメッセージの出力時間の差分から求めてください。

- KFPS02183-I（シンクポイント処理が完了したときに出力されるメッセージ）
- KFPS02179-I（シンクポイント処理をスキップしたときに出力されるメッセージ）

対策方法

シンクポイント時の更新バッファ数を減らすチューニングを行ってください。チューニング方法については，「デファードライトトリガの実行間隔を短くする」及び「デファードライトトリガ時の更新ページの出力比率を高くする」を参照してください。

チューニングを行っても効果がない場合は，pd_dfw_syncpoint_skip_limit オペランドを指定して更新バッファ数を抑制してください。ただし，この場合，シンクポイント処理のスキップ回数が上限に達したとき，更新トランザクションの延長で更新バッファの出力を行うため，更新トランザクション性能が低下します。

23.3.3　シンクポイント処理時間を短縮する方法

シンクポイント処理時間を短縮するには，次に示す三つの方法があります。

- デファードライトトリガの実行間隔を短くする
- デファードライトトリガ時の更新ページの出力比率を高くする
- デファードライト処理用並列 WRITE プロセス数を多くする

(1)　デファードライトトリガの実行間隔を短くする

デファードライトトリガの実行間隔を短くして，シンクポイント時の更新バッファ数を減らしてください。更新バッファ数が減ると，シンクポイント時の更新ページ反映処理時間が短縮できるため，シンクポイント処理時間を短縮できます。

デファードライトトリガの実行間隔のチューニング手順を次に示します。

＜手順＞

1. グローバルバッファに関する統計情報，及びデファードライト処理に関する統計情報を取得してください。
2. グローバルバッファに関する統計情報の実行結果から，シンクポイント出力ページ数（SYNCW）が多いグローバルバッファを確認してください。
3. デファードライトトリガの開始条件を次に示すオペランドで指定してください。

   ・pd_dbbuff_rate_updpage オペランド

   ・pdbuffer オペランドの-y オプション

   グローバルバッファに関する統計情報を参照し，次に示す条件式を満たすようにオペランドの値を変更してください。

   TRGUP（デファードライトトリガ時の出力契機となる更新バッファ数）＜ SYNCW（シンクポイント出力ページ数）÷2

   なお，TRGUP の値を小さくし過ぎると，書き込み処理の回数が増加し，トランザクション性能に影響を与えることがあります。その場合は，オペランドの値を元に戻してください。

4. シンクポイント処理時間がシンクポイントの取得間隔の 10%以内であるか確認してください。10%を超えている場合は，10%以内になるまで 1〜3 の作業を繰り返してください。

シンクポイント処理時間は，デファードライト処理に関する統計情報の実行時間（DWTOTAL）で確認してください。

シンクポイントの取得間隔は，次に示すメッセージの出力時間の差分から求めてください。

- KFPS02183-I（シンクポイント処理が完了したときに出力されるメッセージ）
- KFPS02179-I（シンクポイント処理をスキップしたときに出力されるメッセージ）

(2) デファードライトトリガ時の更新ページの出力比率を高くする

デファードライトトリガ時の更新ページの出力比率を高くして（デファードライトトリガ処理で反映する更新ページ数を多くして），シンクポイント時の更新バッファ数を減らしてください。更新バッファ数が減ると，シンクポイント時の更新ページ反映処理時間が短縮できるため，シンクポイント処理時間を短縮できます。

デファードライトトリガ時の更新ページの出力比率のチューニング手順を次に示します。

＜手順＞

1. デファードライト処理に関する統計情報を取得してください。
2. pdbuffer オペランドの-w オプションの値を大きくしてください。
   なお，-w オプションの値を大きくし過ぎると，書き込み処理の回数が増加し，トランザクション性能に影響を与えることがあります。その場合は，オプションの値を元に戻してください。
3. シンクポイント処理時間がシンクポイントの取得間隔の 10%以内であるか確認してください。10%を超えている場合は，10%以内になるまで 1〜2 の作業を繰り返してください。

シンクポイント処理時間は，デファードライト処理に関する統計情報の実行時間（DWTOTAL）で確認してください。

シンクポイントの取得間隔は，次に示すメッセージの出力時間の差分から求めてください。

- KFPS02183-I（シンクポイント処理が完了したときに出力されるメッセージ）
- KFPS02179-I（シンクポイント処理をスキップしたときに出力されるメッセージ）

(3) デファードライト処理用並列 WRITE プロセス数を多くする

デファードライト処理用並列 WRITE プロセス数を多くして，デファードライト処理時間を短縮してください。デファードライト処理用並列 WRITE プロセス数のチューニング手順を次に示します。

＜手順＞

1. デファードライト処理に関する統計情報を取得してください。

2. pd_dfw_awt_process オペランドの値を大きくして，デファードライト処理用並列 WRITE プロセス数を多くしてください。

   なお，pd_dfw_awt_process オペランドの値を大きくすると，デファードライト処理用並列 WRITE プロセス数が増加するため，CPU の使用負荷が高くなります。そのため，OS の機能などで，CPU の使用負荷を監視してください。

3. シンクポイント処理時間がシンクポイントの取得間隔の 10%以内であるか確認してください。10%を超えている場合は，10%以内になるまで 1～2 の作業を繰り返してください。

シンクポイント処理時間は，デファードライト処理に関する統計情報の実行時間（DWTOTAL）で確認してください。

シンクポイントの取得間隔は，次に示すメッセージの出力時間の差分から求めてください。

- KFPS02183-I（シンクポイント処理が完了したときに出力されるメッセージ）
- KFPS02179-I（シンクポイント処理をスキップしたときに出力されるメッセージ）

デファードライト処理用並列 WRITE プロセス数を多くしても，並列 WRITE 時間（DWPARA，DWPARAM）が短縮されない場合，次に示す原因が考えられます。

- **ディスクの競合が発生している**

  ディスクの構成，RD エリアの構成，及び表の構成を見直し，特定のボリュームに入出力処理が集中しないようにしてください。

- **グローバルバッファの排他待ちが発生している**

  グローバルバッファに関する統計情報の「バッファの排他処理での，排他競合待ち発生率（SLEPR）」で，グローバルバッファの排他待ちの発生率を確認できます。チューニング方法については，「グローバルバッファの排他処理の排他競合待ち発生率（SLEPR）を確認してください」を参照してください。

- **出力ページ数が不足している**

  実際に動作するデファードライト処理用並列 WRITE プロセス数の最大値は，次に示す計算式から求められます。

  - 出力ページ総数（OUT PAGE）÷20

  この計算式の値以上のデファードライト処理用並列 WRITE プロセス数が動作するように設定しても，実際にその分の並列 WRITE プロセスは動作しません。

  なお，出力ページ総数（OUT PAGE）については，デファードライト処理に関する統計情報で確認してください。

23.4　シンクポイントダンプの取得間隔のチューニング

通常，シンクポイントダンプの取得間隔は，次の内容を考慮して設定します。

考慮点

- 大量のデータを更新する処理の場合は，トランザクションを処理する時間が長く，システムログも多くなります。このため，再開始するときの回復時間が長くなります。
- 少量のデータを更新する処理の場合は，トランザクションを処理する時間が短く，システムログも少なくなります。このため，再開始するときの回復時間が短くなります。

このため，シンクポイントダンプの出力間隔をチューニングするときは，「グローバルバッファプールに関する統計情報」，及び「デファードライト処理に関する統計情報」と合わせて検討してください。

取得する情報

統計解析ユティリティの「システムの稼働に関する統計情報」を取得してください。

参照する情報

- シンクポイントダンプ取得間隔時間（SYNC POINT GET INTERVAL）
- シンクポイントダンプ取得時間（SYNC POINT GET TIME）

確認する目的

シンクポイントダンプ出力間隔の妥当性を判断するためには，再開始するときの時間（システムログを取得するために必要となる時間）と，スループットとの関係から判断してください。

HiRDB では，グローバルバッファで更新されたページをデファードライト処理でディスクに書き出しているため，通常よりも短い時間で再開始できる場合があります。このため，デファードライト処理に関する統計情報を取得し，デファードライト処理で書き出されたページ数（OUT PAGE：出力ページ数）と同時に判断するとシンクポイント時の性能を向上できます。

解析結果を判断する基準

1. シンクポイント時にディスクに書き出されたページが多い場合は，デファードライト処理でディスクに書きだしたページ数が少ないと考えられます。
2. 更新バッファヒット率が低く，シンクポイント時の出力ページ数が多い場合は，pdbuffer オペランドの-wオプションの値（デファードライト処理で出力するページの比率）が小さいと考えられます。
3. システムログファイルのスワップ時間がシンクポイント出力間隔から推定される時間よりも短い場合は，システムログファイルの容量が，シンクポイントダンプ出力間隔で出力するブロック数よりも小さいと考えられます。ただし，システムログファイルに出力できる残り容量が，シンクポイントダンプ出力間隔で出力するブロック数に満たない場合，該当しません。この場合，システムログファイルが一杯になった時点で，自動的にシンクポイントダンプが取得されます。

対策方法

1. 再開始処理に掛かる時間をユーザで許容できる範囲にしてください。

2. シンクポイントダンプは実行中のトランザクションが完結しないと有効化できないため，特に長いトランザクションを実行するときは，ほかのトランザクションとの同時実行を避けてください。なお，特に長いトランザクションとは，このトランザクション実行中に一つのユーザサーバで出力するログ量がこのユーザサーバのログファイル容量の１/３を超える可能性があるトランザクションのことです。また，ここで述べたユーザサーバとは，HiRDB/シングルサーバの場合はシングルサーバであり，HiRDB/パラレルサーバの場合はフロントエンドサーバ，バックエンドサーバ，又はディクショナリサーバです。

23.5　バッファ長のチューニング

ここでは，次に示すバッファ長をチューニングする方法について説明します。

- 表定義情報用バッファ長
- ビュー解析情報用バッファ長
- ユーザ権限情報用バッファ長
- SQL オブジェクト用バッファ長
- ユーザ定義型情報用バッファ長
- ルーチン定義情報用バッファ長
- レジストリ情報用バッファ長

23.5.1　表定義情報用バッファ長のチューニング

表定義情報用バッファ長（pd_table_def_cache_size オペランドの値）のチューニング方法を説明します。

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- 表定義情報用バッファヒット回数（#OF TBL-CACHE HIT）
- 表定義情報取得要求回数（#OF TBL-DEF GET REQ）

チューニング方法

次に示す計算式で表定義情報バッファヒット率を求めて，ヒット率が 80%を超えるようにしてください。

表定義情報用バッファヒット率（%）＝

（表定義情報用バッファヒット回数※÷表定義情報取得要求回数※）×100

注※

順序数生成子を使用する場合は，順序数生成子も含みます。

対策方法

pd_table_def_cache_size オペランドの指定値を大きくしてください。

23.5.2　ビュー解析情報用バッファ長のチューニング

ビュー解析情報用バッファ長（pd_view_def_cache_size オペランドの値）のチューニング方法を説明します。

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- ビュー解析情報用バッファヒット回数（#OF VIEW CACHE HIT）
- ビュー解析情報取得要求回数（#OF VIEW DEF GET REQ）

チューニング方法

次に示す計算式でビュー解析情報バッファヒット率を求めて，ヒット率が 80%を超えるようにしてください。

ビュー解析情報用バッファヒット率（%）＝
（ビュー解析情報用バッファヒット回数÷ビュー解析情報取得要求回数）×100

対策方法

pd_view_def_cache_size オペランドの指定値を大きくしてください。

23.5.3 ユーザ権限情報用バッファ長のチューニング

ユーザ権限情報用バッファ長（pd_auth_cache_size オペランドの値）のチューニング方法を説明します。

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- ユーザ権限情報用バッファヒット回数（#OF CON/DBA CACHE HIT）
- ユーザ権限情報取得要求回数（#OF CON/DBA DEF GET REQ）

チューニング方法

表定義情報用バッファヒット率（%）に対して，ユーザ権限情報バッファヒット率が低い場合，次に示す内容が要因で，処理性能が劣化していると考えられます。

- CONNECT 権限及び DBA 権限の情報が多過ぎるため，ユーザ権限情報バッファヒット率が低くなっています。

なお，ユーザ権限情報バッファヒット率は次の計算式で求めてください。

ユーザ権限情報バッファヒット率（%）＝
（ユーザ権限情報用バッファヒット回数÷ユーザ権限情報取得要求回数）×100

対策方法

pd_auth_cache_size オペランドの指定値を大きくしてください。

23.5.4 SQL オブジェクト用バッファ長のチューニング

SQL オブジェクト用バッファ長（pd_sql_object_cache_size オペランドの値）のチューニング方法を説明します。SQL オブジェクトで使用するバッファをチューニングするときには，次に示す内容を考慮してください。

●SQL オブジェクトと SQL オブジェクト用バッファの関係

HiRDB では，UAP 中の SQL 文を SQL 文ごとに解析して，UAP を実行するときに，実行形式の SQL オブジェクトを作成しています。このため，同じ UAP を異なるユーザが後から実行する場合，発行した SQL 文に対応する SQL オブジェクトが，既に SQL オブジェクト用バッファにあると，SQL オブジェクトを作成する時間を削減でき，発行した SQL 文の処理時間を短縮できます。

したがって，SQL オブジェクト用バッファのヒット率が低い場合，バッファから出力された SQL 文の SQL オブジェクトは，再度同じ SQL 文を実行するときに SQL 文を解析して，SQL オブジェクトを作成します。このため，データベースに対するアクセス件数が少ない SQL 文の実行時には，長い処理時間が必要となります。

また，SQL オブジェクト作成に関しては，SQL 文の解析時にアクセスする表の定義情報が表定義情報バッファにない場合，ディクショナリ表に対するアクセスを必要とするため，アクセスする表の数や列数，アクセス件数が多いときは排他に要するリソースが増え，長い処理時間が必要となります。

●SQL オブジェクト用バッファの考え方

HiRDB/パラレルサーバでは，一つの SQL 文に対して複数の SQL オブジェクトを作成しています。次にバックエンドサーバとフロントエンドサーバでの，SQL オブジェクトの特長を示します。

- **バックエンドサーバの場合**

  該当するバックエンドサーバ（フロータブルサーバを含みます）ごとに実行する形式として作成されています。このため，バックエンドサーバの SQL オブジェクト用バッファには，フロントエンドサーバでバッファリングしている SQL オブジェクトのうち，該当するバックエンドサーバの SQL オブジェクトだけがバッファリングされるため，必要な SQL オブジェクト用バッファはフロントエンドサーバよりも少なく済みます。

- **フロントエンドサーバの場合**

  フロントエンドサーバでは，バックエンドサーバのすべての SQL オブジェクトを SQL オブジェクト用バッファに格納しています。このため，コストベース最適化によって，選択できるアクセス手順に SQL オブジェクトが作成されるため，相当量の SQL オブジェクト用バッファがないと効果がありません。

  なお，ストアドプロシジャ及びストアドファンクションについても同様のことが言えます。ストアドプロシジャ及びストアドファンクションの場合，通常の SQL と比べて特に SQL オブジェクトサイズが大きくなるため注意してください。

なお，次に示す統計情報と合わせて解析すると効率良く性能を向上できます。

- **SQL オブジェクトに関する情報**

  SQL オブジェクトごとに使用するバッファ長を解析するときに取得します。

- **SQL に関する情報**

  バッファ長が小さ過ぎるかどうかを解析するために取得します。

(1) チューニング方法（その１）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- SQL オブジェクト用バッファヒット回数（#OF CACHE HIT（SQLOBJ））
- SQL オブジェクト取得要求回数（#OF SQLOBJ INFO GET）
- SQL オブジェクト用バッファ中の SQL オブジェクト数（#OF CACHED SQLOBJ）
- ストアドプロシジャのオブジェクトの SQL オブジェクト用バッファヒット回数（#OF CACHED HIT（STRT））
- ストアドプロシジャのオブジェクトの取得要求回数（#OF STRT INFO GET）
- SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数（#OF CACHED STRT）

チューニング方法

1. SQL オブジェクト用バッファヒット率が低く，バッファ中の SQL オブジェクト情報合計長が短い場合，ヒット率を高くできない業務形態と考えられます。
2. SQL オブジェクト用バッファヒット率が低く，バッファ中の SQL オブジェクト情報合計長が長い場合，SQL オブジェクト用バッファが不足していると考えられます。

なお，SQL オブジェクト用バッファヒット率は次の計算式で求めてください。

SQL オブジェクト用バッファヒット率（%）＝
（SQL オブジェクトバッファヒット回数÷SQL オブジェクト取得要求回数）×100

対策方法

SQL オブジェクト用バッファが不足している場合，pd_sql_object_cache_size オペランドの指定値を大きくしてください。

(2) 解析方法（その２）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- SQL オブジェクトバッファから出された SQL オブジェクトの数（#OF SWAP OUT SQLOBJ）
- SQL オブジェクト取得要求回数（#OF SQLOBJ INFO GET）
- SQL オブジェクト用バッファから出されたストアドプロシジャのオブジェクトの数（#OF SWAP OUT STRT）
- ストアドプロシジャのオブジェクトの取得要求回数（#OF STRT INFO GET）

チューニング方法

SQL オブジェクトバッファから出された SQL オブジェクトの数が小さくなるようにしてください。SQL オブジェクトバッファから出された SQL オブジェクトの数が多くなると SQL オブジェクトの再作成による処理時間が長くなり，SQL オブジェクト用バッファの使用状況が悪いと考えられます。

対策方法

pd_sql_object_cache_size オペランドの値を大きくしてください。

(3) 解析方法（その３）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- SQL オブジェクト長（REQUEST SQLOBJ SIZE）
- ストアドプロシジャのオブジェクト長（REQUEST STRT SIZE）

チューニング方法

SQL オブジェクト長（REQUEST SQLOBJ SIZE）とストアドプロシジャのオブジェクト長（REQUEST STRT SIZE）の合計の最大値が，指定したバッファ長よりも大きい場合，SQL オブジェクトを実行するためのメモリが確保できないと考えられます。

対策方法

pd_sql_object_cache_size オペランドの値を大きくしてください。

(4) 解析方法（その４）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- ストアドプロシジャのオブジェクトのリコンパイル回数（＃OF STRT RECOMPILE)

チューニング方法

通常，CALL 文でストアドプロシジャを実行する場合，リコンパイルは発生しません。しかし，ストアドプロシジャ内で使用している表に対してインデクスを追加したり，使用していないインデクスを削除したりするとリコンパイルが発生します。このリコンパイルは，CALL 文でストアドプロシジャを実行するごとに発生するため，性能劣化の原因となります。

リコンパイルが発生していたら，ストアドプロシジャ内で使用している表に対して上記の変更があったと考えられます。

対策方法

ALTER PROCEDURE 又は ALTER ROUTINE で，ストアドプロシジャを再作成してください。

23.5.5 ユーザ定義型情報用バッファ長のチューニング

ユーザ定義型情報用バッファ長（pd_type_def_cache_size オペランドの値）のチューニング方法を説明します。

(1)　チューニング方法（その１）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- 型定義キャッシュヒット回数（# OF TYPE-DEF CACHE HIT）
- 型定義情報取得要求回数（# OF TYPE-DEF GET REQ）

チューニング方法

次に示す計算式で型定義キャッシュヒット率を求めて，ヒット率が 100%になるようにしてください。

型定義キャッシュヒット率（%）＝

（型定義キャッシュヒット回数÷型定義情報取得要求回数）×100

対策方法

pd_type_def_cache_size オペランドの指定値を大きくしてください。

(2)　チューニング方法（その２）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- 型定義キャッシュ総使用領域長（TYPE-DEF CACHE TOTAL SIZE）
- 確保した型定義キャッシュ長（TYPE-DEF CACHE ALLOC SIZE）

チューニング方法

次に示す計算式で**型**定義キャッシュ使用率を求めて，使用率が 90%以上になるようにしてください。

型定義キャッシュ使用率（%）＝

(型定義キャッシュ総使用領域長の最大値÷確保した型定義キャッシュ長）×100

対策方法

pd_type_def_cache_size オペランドの指定値を小さくしてください。

23.5.6　ルーチン定義情報用バッファ長のチューニング

ルーチン定義情報用バッファ長（pd_routine_def_cache_size オペランドの値）のチューニング方法を説明します。

(1)　チューニング方法（その１）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- ルーチン定義キャッシュヒット回数（# OF RTN-DEF CACHE HIT）

- ルーチン定義情報取得要求回数（# OF RTN-DEF GET REQ）

チューニング方法

次に示す計算式でルーチン定義キャッシュヒット率を求めて，ヒット率が 100%になるようにしてください。

ルーチン定義キャッシュヒット率（%）＝

（ルーチン定義キャッシュヒット回数÷ルーチン定義情報取得要求回数）×100

対策方法

pd_routine_def_cache_size オペランドの指定値を大きくしてください。

(2) チューニング方法（その２）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- ルーチン定義キャッシュ総使用領域長（RTN-DEF CACHE TOTAL SIZE）
- 確保したルーチン定義キャッシュ長（RTN-DEF CACHE ALLOC SIZE）

チューニング方法

次に示す計算式でルーチン定義キャッシュ使用率を求めて，使用率が 90%以上になるようにしてください。

ルーチン定義キャッシュ使用率（%）＝

(ルーチン定義キャッシュ総使用領域長の最大値÷確保したルーチン定義キャッシュ長）×100

対策方法

pd_routine_def_cache_size オペランドの指定値を小さくしてください。

(3) チューニング方法（その３）

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- プラグインルーチン定義キャッシュヒット回数（# OF PLG-RTN CACHE HIT）
- プラグインルーチン定義取得要求回数（# OF PLG-RTN GET REQ）

チューニング方法

次に示す計算式でプラグインルーチン定義キャッシュヒット率を求めて，ヒット率が 100%になるようにしてください。

プラグインルーチン定義キャッシュヒット率（%）＝

（プラグインルーチン定義キャッシュヒット回数÷プラグインルーチン定義情報取得要求回数）×100

対策方法

pd_routine_def_cache_size オペランドの指定値を大きくしてください。

23.5.7　レジストリ情報用バッファ長のチューニング

レジストリ情報用バッファ長（pd_registry_cache_size オペランドの値）のチューニング方法を説明します。

参照する情報

統計解析ユティリティの「システムの稼働に関する統計情報」の次に示す情報を参照してください。

- レジストリキャッシュヒット回数（# OF REGISTRY CACHE HIT）
- レジストリ情報取得要求回数（# OF REGISTRY GET REQ）

チューニング方法

次に示す計算式でレジストリキャッシュヒット率を求めて，ヒット率が 100%になるようにしてください。

レジストリキャッシュヒット率（%）＝
（レジストリキャッシュヒット回数÷レジストリ情報取得要求回数）×100

対策方法

pd_registry_cache_size オペランドの指定値を大きくしてください。

23.6　プロセス数のチューニング

統計解析ユティリティの「システムの稼働に関する統計情報」から，サーバプロセス数のチューニングをします。実際にサービスを実行していたサーバプロセスの稼働数や最大起動プロセス数を超えるサービス要求の有無を解析し，システムによって最適な状態でサーバプロセスを起動させることを目的とします。

サービス実行中とは，各サーバでの次に示す状態をいいます。

- フロントエンドサーバ及びシングルサーバの場合，HiRDB クライアントが CONNECT 文を発行してサーバプロセスに割り当てられた状態のことです。DISCONNECT 文が発行されるまで続きます。
- バックエンドサーバ及びディクショナリサーバの場合，トランザクション実行中の状態のことです。トランザクションが決着するまで続きます。

取得する情報

統計解析ユティリティの「システムの稼働に関する統計情報」を取得してください。

23.6.1　最大起動プロセス数のチューニング

次に示すオペランドで指定した最大起動プロセス数のチューニング方法について説明します。

- pd_max_users
- pd_max_bes_process
- pd_max_dic_process

参照する情報

システムの稼働に関する統計情報の次に示す情報を参照してください。

- 最大起動プロセス数を超えたサービス要求数（#OF REQ PROCESS OVER MAX）
- サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）

確認する目的

- 処理中に実際にサービスを実行していたサーバプロセスの稼働数に，これらのオペランドで指定する最大起動プロセス数を合わせて，メモリ資源を効率的に使用します。
- また，過少見積もりの最大起動プロセス数の指定値を上げて，サービスが受け入れられないことを防ぎます。

解析結果を判断する方法

●オペランドの値が「サービス実行中のサーバプロセス数」より大きい場合

オペランドの値を小さくする余裕があると考えられます。

各サーバの最大起動プロセス数を「サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）」の最大値に近い値まで下げるようにします。ただし，次の点に注意してください。

- ほかのサーバの状況によっては「サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）」の最大値まで下げられない場合があります。例えば，あるバックエンドサーバに複数のフロントエンドサーバからのサービス要求が集中する可能性のある場合や，ユニット内の別のフロントエンドサーバに処理が偏っている場合などです。
- pd_max_bes_process，pd_max_dic_process のチューニング時に，pd_max_users の指定値よりも小さな値を指定した場合，pd_max_users の指定値が仮定されます。

●最大起動プロセス数を超えたサービス要求が発生している場合

オペランドの値を大きくする必要があると考えられます。

各サーバの最大起動プロセス数を，受け入れられないサービス要求が発生しなくなるまで増加させるようにします。チューニング方法としては，サーバごとに次の点に注意してください。

- フロントエンドサーバ及びシングルサーバの場合，サーバが受けたサービス要求が最大起動プロセス数を超えると，クライアントは次に示す最大リトライ回数までリトライをします。

  HiRDB クライアントが HiRDB Version 4.0 03-03 以前：100 回

  HiRDB クライアントが HiRDB Version 4.0 04-00 以降：10 回

  「最大起動プロセス数を超えたサービス要求数（#OF REQ PROCESS OVER MAX）」の値はリトライ回数も含んでいるため，これも考慮して修正してください。
- バックエンドサーバ及びディクショナリサーバの場合，サービス要求はリトライされないため，「最大起動プロセス数を超えたサービス要求数（#OF REQ PROCESS OVER MAX）」の値で修正できます。

ただし，pd_max_users を増加した場合，増加した値が pd_max_bes_process，pd_max_dic_process の指定値を上まわると，pd_max_bes_process，pd_max_dic_process の指定値も pd_max_users の指定値まで増加するように仮定されるので注意してください。

対策方法

- フロントエンドサーバ，シングルサーバの場合は，pd_max_users の値を修正してください。
- バックエンドサーバの場合は，pd_max_bes_process の値を修正してください。
- ディクショナリサーバの場合は，pd_max_dic_process の値を修正してください。

23.6.2　常駐プロセス数のチューニング

pd_process_count オペランドで指定した常駐プロセス数のチューニング方法について説明します。

参照する情報

システムの稼働に関する統計情報の次に示す情報を参照してください。

- サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）

確認する目的

処理中に実際にサービスを実行していたサーバプロセスの稼働数の平均値に近い値に常駐プロセス数を指定して，各サーバのサーバプロセス起動時間の削減とメモリ資源の削減を図ります。

解析結果を判断する方法

各サーバのサーバプロセスをあらかじめ常駐しておくことでサーバプロセス起動時間を削減できますが，必要以上に多くのサーバプロセスを常駐させるとメモリ資源を圧迫します。そこで，各サーバで指定する常駐プロセス数は「サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）」の平均値が妥当だと考えられます。

●サーバのサービス実行のレスポンスを改善する場合

pd_process_count オペランドの値が「サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）」の平均値よりも小さければ，平均値程度まで上げるようにします。さらに上げることもできますが，その場合はサービスを受けない常駐プロセスが，余計なメモリを占有する可能性が高くなるので注意してください。

●常駐サーバプロセスの占有メモリ量を減らす場合

pd_process_count オペランドの値が「サービス実行中のサーバプロセス数（#OF PROCESS ON SERVICE）」の平均値よりも大きければ，平均値程度まで下げるようにします。さらに下げることもできますが，その場合は常駐プロセス数を超える要求を受ける可能性が高くなり，サーバプロセスの起動時間分だけサービス実行のレスポンスが低下するので注意してください。

対策方法

pd_process_count オペランドの値を修正してください。

23.6.3　非同期 READ プロセス数のチューニング

pd_max_ard_process オペランドで指定した非同期 READ プロセス数のチューニング方法について説明します。非同期 READ 機能を使用している場合，非同期 READ プロセス数が少なくて非同期 READ 機能が適用される SQL の多重度が高いと，非同期 READ プロセスの入出力処理の完了待ちが発生することがあるため，処理時間が余計に掛かることがあります。

参照する情報

UAP 統計レポートの次に示す情報を参照してください。UAP 統計レポートについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

- 非同期 READ 時の同期待ち時間の累計（ARWT，ARWTM）
- データベースに対する入出力時間の累計（IOTIM，IOTIMM）
- 非同期 READ 時の平均同期待ち時間（ARWTA，ARWTMA）
- 非同期 READ 時の平均同期入出力時間（ARSTA，ARSTMA）

確認する目的

非同期 READ 機能の効果がない場合に，非同期 READ プロセス数の妥当性を判断します。

解析結果を判断する方法

次に示す時間を比べてください。2 の方が長い場合は同期待ち時間が掛かっていると考えられます。

1. 非同期 READ 機能を使用しない場合の「データベースに対する入出力時間の累計（IOTIM，IOTIMM）」

2. 非同期 READ 機能を使用した場合の「非同期 READ 時の平均同期待ち時間（ARWTA，ARWTMA）」と「データベースに対する入出力時間の累計（IOTIM，IOTIMM）」の合計

次に示す時間を比べてください。2 の方が長い場合は同期待ち時間が掛かっていると考えられます。

1. 非同期 READ 機能を使用した場合の「非同期 READ 時の平均同期入出力時間（ARSTA，ARSTMA）」
2. 非同期 READ 機能を使用した場合の「非同期 READ 時の平均同期待ち時間（ARWTA，ARWTMA）」

また，統計解析ユティリティのデータベース操作に関する HiRDB ファイルの統計情報から「非同期 READ 回数（AIO-R）」が分かります。

対策方法

「非同期 READ 時の同期待ち時間の累計（ARWT，ARWTM）」又は「非同期 READ 時の平均同期待ち時間（ARWTA，ARWTMA）」が長い場合は，pd_max_ard_process オペランドの値を大きくしてください。

ただし，「非同期 READ 回数（AIO-R）」を調べた結果，同一 RD エリアに非同期 READ 要求が集中している場合は，このオペランドの値を大きくしても効果はありません。この場合，RD エリアを複数の HiRDB ファイルで構成し，更に各 HiRDB ファイルを別々のディスクに配置してください。

pd_max_ard_process オペランドの最大有効値は，「非同期 READ 回数（AIO-R）」が設定されている RD エリアの中で同時にプリフェッチ機能を実行する数を指定します。非同期 READ プロセス数を増やす場合は，システムのリソース（共用メモリとメッセージキュー）を考慮する必要があります。

「非同期 READ 時の同期待ち時間の累計（ARWT，ARWTM）」又は「非同期 READ 時の平均同期待ち時間（ARWTA，ARWTMA）」が短い場合は，これ以上非同期 READ 機能の効果は得られません。

23.7　インデクスのチューニング

23.7.1　インデクスのチューニングで確認すること

表を更新するとデータベースの整合性を維持するため，インデクスも更新されます。このため，インデクスを定義しているときは，次の内容に関して検討する必要があります。

検討事項

- 表に定義したインデクスの数に比例して，インデクスの更新処理が発生し，SQL の処理時間が増加します。
- インデクスが更新されると更新した情報がシステムログファイルに取得されます。このため，インデクスの数が増加するとシステムログファイルへの出力量も比例して増加します。
- 更新したキー値を格納するためのインデクスページが不足すると，インデクスページスプリットが発生します。インデクスページスプリットが発生すると，インデクスページスプリットに関するシステムログが取得され，システムログファイルの量が増加します。

取得する情報

統計解析ユティリティの「インデクスに関する統計情報」を取得してください。

(1)　スプリット回数（SP_NM）

確認する目的

これらの情報は，インデクスの妥当性を判断するために取得します。通常，インデクスが原因で，性能が低下する要因には，次に示す要因が考えられます。

- インデクス定義時に適切な空き領域（PCTFREE）が指定されていない場合，インデクスページスプリットが発生しています。

このため，インデクススプリット回数の情報で，インデクスの定義などを調節してください。

解析結果の妥当性を判断する方法

1. インデクスページスプリットが発生している回数が多い場合は，表への操作として，挿入（INSERT）処理を主体とした業務で，インデクスが拡張されているためにインデクスページスプリットが発生していると考えられます。
2. 表が分割されていて，特定の RD エリアに対するインデクスページスプリットが多い場合は，ある特定のキーレンジに更新が集中しているため，表の分割方法に問題があると考えられます。

対策方法

次に示すどれかの方法で対策してください。

1. インデクス定義時に指定する未使用領域（PCTFREE）を大きくしてください。また，大量にデータを追加するときは，データベース作成ユティリティを使用してください。
2. 不要なインデクスは削除してください。

3. 表を分割している場合，インデクスページスプリットが集中しているキーレンジをさらに分割できるかどうか検討してください。

23.8　データベースのチューニング

RD エリア単位の状態解析情報は，容量面やデータベースの乱れをチューニングするために必要な情報です。定義した RD エリアの状態，表やインデクスの格納状態からデータベースの状態を解析します。

取得する情報

データベース状態解析ユティリティで，各種情報を取得してください。

23.8.1　RD エリア単位の物理的解析情報

確認する目的

RD エリア単位の物理的解析情報では，RD エリアの使用効率や容量の状態を RD エリア定義時の容量又はセグメントサイズなどから，妥当性を判断するために取得します。

表やインデクスを意識しないで RD エリア内のすべてのセグメント，及びすべてのページの格納状態を解析し，RD エリアの使用効率や容量の状態をつかんでください。

解析結果の妥当性を判断する方法

●使用中セグメント比率が低い場合

未使用セグメントが多いと考えられます。このため，次に示す内容を検討してください。

- 一つの RD エリアを複数の HiRDB ファイルで構成している場合，個々のファイル容量から，すべての HiRDB ファイルが使用されていないか調査してください。
- 今後予定されているデータの追加量，表やインデクスの定義などから判断してください。

●使用中セグメント比率が高く使用中ページ比率が低い場合

次に示す要因が考えられるため，セグメント内の空きページ比率が大き過ぎないか確認してください。

- RD エリアが効率良く使用されていないと考えられます。
- 使用中セグメント比率が高くなる要因としては，RD エリア定義時に指定したセグメントサイズが，RD エリアの容量と比較して大きい値を設定するとセグメント数が少なくなり，定義した表やインデクスがすべてのセグメントを使用していると考えられます。

●使用中セグメント比率が高く，使用中ページ比率が高い場合

次に示す要因が考えられます。

- RD エリア内に空きが少ないことが考えられます。
- 満杯ページ比率が高い場合は，容量が不足していることが考えられます。

対策方法

RD エリアの作成時に指定したセグメントサイズが妥当な値であるかどうかを調査してください。該当する RD エリアに格納する表又はインデクスの件数と，データ件数から妥当性を判断してください。特に満杯ページ比率が高い又は低い場合は，次に示す方法で対策してください。

- 満杯ページ比率が高い場合は，データベース構成変更ユティリティで，RD エリアを拡張してください。又は，RD エリア内の表やインデクスを，ほかの RD エリアに移してください。
- 満杯ページ比率が低い場合は，RD エリア内の格納効率が下がり，データの配置に乱れが発生していることが考えられます。この場合，データベース再編成ユティリティで，RD エリア単位に表を再編成してください。
- データディクショナリ用 RD エリアの場合は，データベース再編成ユティリティで-C オプションに dir を指定して，再編成してください。なお，データディクショナリ用 RD エリアを再編成するときは，特定のディクショナリ表だけを再編成するか，又はすべてのディクショナリ表を再編成するかを選択できます。ディクショナリ表の指定を省略すると，すべてのデータディクショナリ用 RD エリアを対象に再編成します。

23.8.2　RD エリア単位の論理的解析情報

確認する目的

RD エリア単位の論理的解析情報では，次に示す内容の妥当性を判断するために取得します。

- RD エリアの容量が見積もった容量と一致しているかどうか
- データベースを再編成するかどうか

RD エリアごとに表やインデクスの全セグメント，及び全ページの格納状態を解析してください。

解析結果の妥当性を判断する方法

●表に関する状態解析情報

1. 使用中セグメント比率が高い場合

   該当する RD エリアに未使用セグメントがないときは，これ以上のデータを挿入できないと考えられます。この場合，今後の追加するデータ量，又は更新するデータ量から，空き領域（PCTFREE）の妥当性を検討してください。

2. 使用中セグメント比率が高く，使用中ページ比率が低い場合

   PCTFREE オペランドで指定したセグメント内の空きページ比率が適切ではないと考えられます。この場合，今後の追加するデータ量，又は更新するデータ量から，空き領域（PCTFREE）の妥当性を検討してください。

3. 見積もったページ数よりも，使用中ページ数が多い場合

   VARCHAR，NVARCHAR，又は MVARCHAR の列を含むときは，これらの列のデータ長が 255 バイトを超えたことを考慮して，計算したかどうか見直してください。

4. 満杯セグメント比率が高く，満杯ページ比率も高い場合

   RD エリアの容量が不足しています。又は，次に示す操作を繰り返したため，データの配置が乱れています。

   - ナル値のデータから，ほかの数値又は文字への更新
   - VARCHAR，NVARCHAR，又は MVARCHAR の列に対して，列長が長くなる更新

- 列の値をナル値に更新，又は VARCHAR，NVARCHAR，MVARCHAR の列に対して行長が短くなる更新

●インデクスに関する状態解析情報

使用中セグメント数及び使用中ページ数が，机上で計算したページ数よりも多い場合は，キー値を大量に追加したためインデクススプリットが多発し，ページの使用率が増加したと考えられます。

対策方法

- RD エリアを拡張するか又は該当する RD エリアに複数の表を格納している場合，表を異なる RD エリアに配置してください。
- データベース再編成ユティリティを使用して表を再編成してください。
- 表定義時に指定した PCTFREE オペランドのセグメント内の空きページ比率の指定を見直してください。
- 表定義時に指定した PCTFREE オペランドのページ内の未使用領域比率の指定を見直してください。

23.8.3 表，インデクス単位の論理的解析情報

確認する目的

表を横分割した場合，分割した表やインデクスが，正しく分割されているか検討してください。この場合，出力されたデータ量（セグメント数，ページ数）と，見積もったデータ量と比較して，分割方法が正しいかどうか確認します。また，表又はインデクスの格納状態から，データベースを再編成するかどうかも確認できます。

解析結果の妥当性を判断する方法

●表に関する状態解析情報

1. 満杯セグメント比率や満杯ページ比率が 80%以上の場合

   見積もったセグメント数からみて，使用中ページ比率が非常に高い場合，表の格納状態が乱れていると考えられます。

2. 表を横分割している場合

   特定の RD エリアの使用中セグメント数が多いとき，又は特定の RD エリアに格納されている行数が多いときは，表を最適に分割していないと考えられます。あらかじめ予想されているデータ量から，表の横分割方法を見直してください。また，既にハッシュ分割している場合は，ハッシュ関数を変更してください。このとき，RD エリアごとの格納行数を確認して，行数が均等になるようにしてください。

3. 表の総未使用ページ比率｛(総ページ数−総使用中ページ数）÷総ページ数｝が，表定義時に指定したセグメント内の空きページ比率よりも小さい場合

   データを繰り返し追加したため，未使用ページが少なくなっていることが考えられます。

4. これから追加するデータ量に比べて空きページが多い場合

セグメント内の空きページ比率が大き過ぎると考えられます。又は，データの削除によって使用中空きページが増えたと考えられます。

●インデクスに関する状態解析情報

1. 表の格納行数からインデクスの格納ページ数を計算し，解析結果の総使用中ページ数より少ない場合

   特定範囲の行を大量に削除すると対応するインデクスのキー値も削除されるため，インデクスページに使用中空きページが発生していると考えられます。

2. 使用中ページがほとんど満杯ページの場合

   データベース再編成ユティリティでインデクスを再編成してもこの状態になる場合は，セグメント内の空きページ比率が適切な値でないと考えられます。又は，データを追加しているため，ページ内の空き領域がなくなっていると考えられます。

対策方法

●表に関する状態解析情報

1. 満杯セグメント比率や満杯ページ比率が高い場合

   今後もデータを追加するときは該当する表を再編成してください。

2. 空きページ（未使用ページ）が多い場合

   セグメント内の空きページ比率を見直してください。

3. データの削除で使用中空きページが増えている場合

   表を再編成するか，インデクスを再編成するか，又は使用中空きページを解放してください。

4. RD エリア内に未使用セグメントが少ない場合

   RD エリアを拡張してください。

●インデクスに関する状態解析情報

1. 使用中空きページが多い場合

   表を再編成するか，インデクスを再編成するか，又は使用中空きページを解放してください。

2. 今後もデータの追加などが予想される場合

   定義したインデクスの列のデータがランダムに発生するときは，インデクスを再度定義し，PCTFREE オペランドで適切な値を指定してください。

3. ほかの RD エリアに比べて，特定の RD エリアだけが使用中ページ数が多い場合

   表の分割方法を再度検討してください。表の横分割後のデータ量がすべての RD エリアで平均するようにしてください。

4. 特定の RD エリアだけ使用中空きページが多い場合

   RD エリア内の表及びインデクスを再編成するか，使用中空きページを解放してください。

23.8.4　クラスタキー及びクラスタリングデータページの格納状態解析

確認する目的

クラスタキーを定義している場合，表及びインデクスの乱れ具合を示す格納順序不正率，重複構造で格納されているキー数を確認してください。

●クラスタキーの格納状態解析

クラスタキーをキー順に検索して，格納位置がページ又はセグメントにわたって格納されている格納位置変更回数と，格納位置変更回数のうち格納順序がページ（セグメント）の昇順に対して逆方向に格納されている格納順序不正回数（率）をページ単位及びセグメント単位で表示します。

ページ分割の発生によって格納順序不正率は上昇します。

●クラスタリングデータページの格納状態解析

クラスタキーを定義した表のデータページをクラスタリングデータページといいます。

クラスタキー中の行データの格納位置情報によって，クラスタキー順に検索した場合の行データの格納位置変更回数と格納順序不正回数（率）をページ単位及びセグメント単位で表示します。

行の追加及び更新によって格納状態が乱れると，格納順序不正率か格納位置変更回数のどちらか，又は両方が増加します。

解析結果の妥当性を判断する方法

●クラスタキーの格納状態解析情報

1. インデクス中の格納キー数に比較して行数が多い場合

   キーの重複率が高いため，インデクスを使用した検索性能に影響を及ぼします。

2. 重複キー構造で格納されているキーがある場合

   重複度の高いキーがある（又は過去にあった）ため，インデクスを使用した検索性能に影響を及ぼします。

3. 格納順序不正率が高い場合

   データページの格納順序が乱れているため，検索性能に影響を及ぼします。

●クラスタリングデータページの格納状態解析情報

1. 格納順序不正率が高い場合

   データページの格納順序が乱れているため，クラスタキー順アクセス性能に影響を及ぼします。

2. 格納位置変更回数が表単位の状態解析結果の使用中セグメント数又はページ数から−１したものより大きい場合

   データページの格納順序が乱れているため，クラスタキー順アクセス性能に影響を及ぼします。

対策方法

●クラスタキーの格納状態解析情報

1. インデクス中の格納キー数に比較して行数が多い場合

   インデクス定義の列構成を見直してください。

2. 重複キー構造で格納されているキーがある場合

現在，重複度の高いキーがある場合は，インデクス構成列中のデータ重複度の高い列を除いた形でクラスタ付きの表を再定義してください。過去に重複度の高いキーがあった場合は，データベース再編成ユティリティで表を再編成又はインデクスを再作成すると，重複キー構造はなくなります。

3. 格納順序不正率が高い場合

   データベース再編成ユティリティで，表を再編成又はインデクスを再作成してください。

●クラスタリングデータページの格納状態解析情報

1. 格納順序不正率が高い場合

   データベース再編成ユティリティで，表を再編成してください。

2. 格納位置変更回数が表単位の状態解析結果の使用中セグメント数又はページ数から−１したものより大きい場合

   データベース再編成ユティリティで，表を再編成してください。

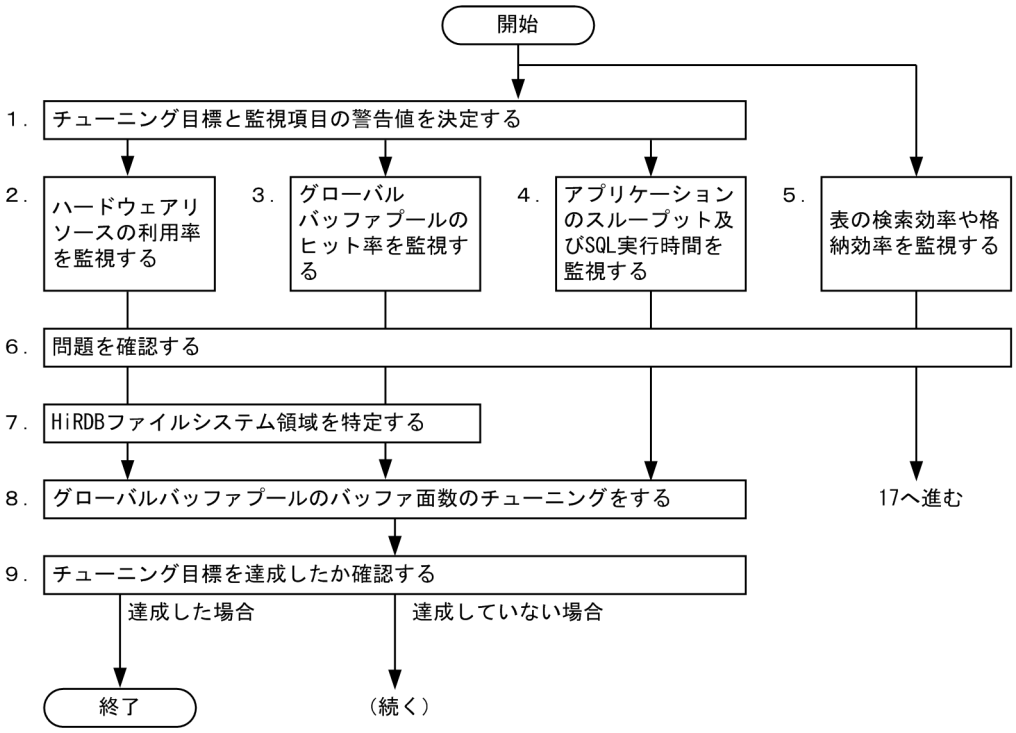
23.9　SQL のチューニング

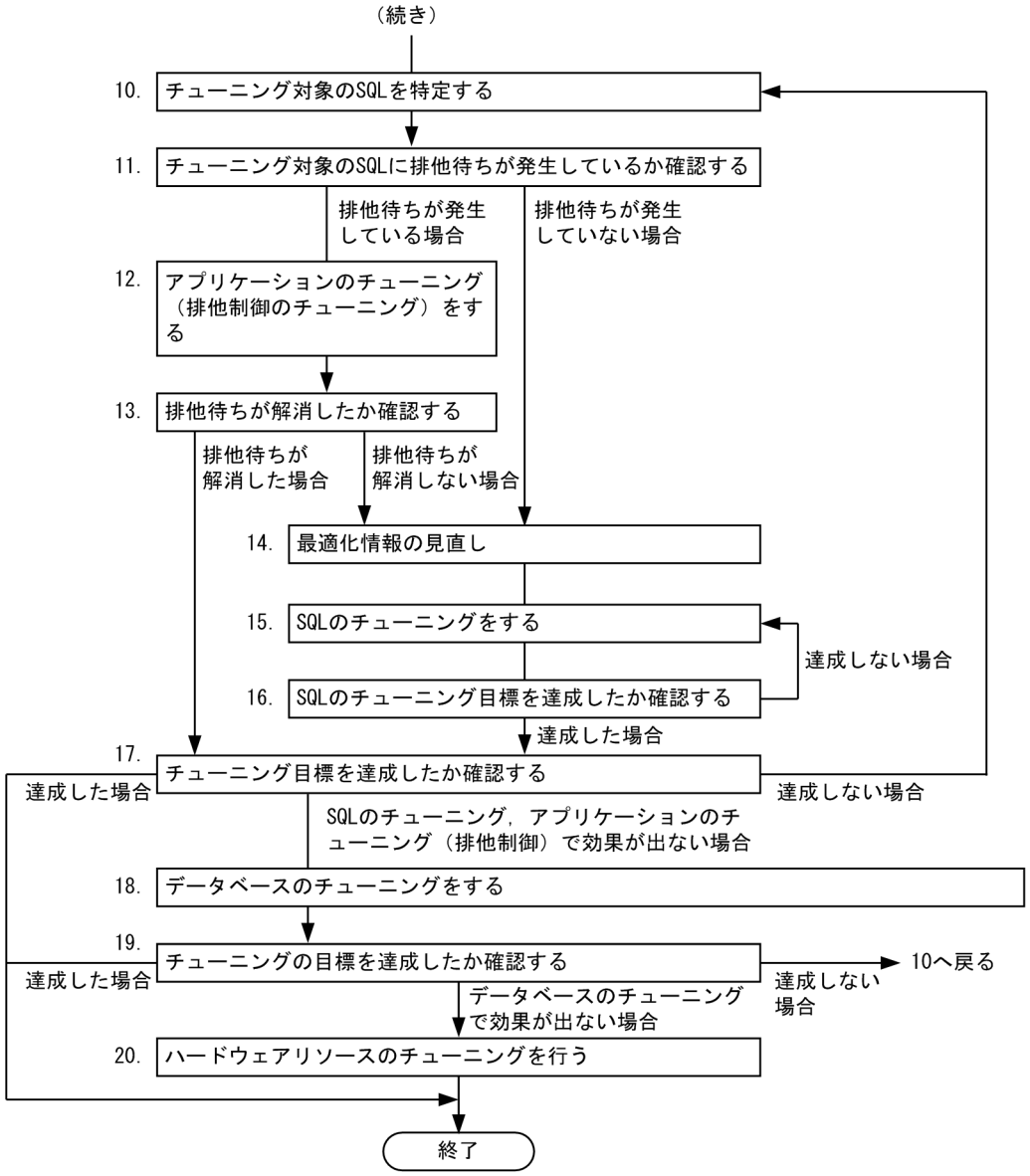
23.9.1　SQL のチューニング方法

ここでは，実行時間が長い SQL，又は入出力処理が多い SQL のチューニング方法について説明します。

SQL のチューニングの流れを次の図に示します。

図 23–2　SQL のチューニングの流れ





注

処理ボックスの左にある数字はこの後で説明している（　）レベルに対応しています。例えば，5 の操作は(5)で説明しています。

(1)　チューニング目標と監視項目の警告値を決定する

チューニング目標と監視項目の警告値を決定します。システム全体のパフォーマンス（アプリケーションの処理時間，トランザクションコミット数など）を決めて，それに対するチューニング目標を決定してください。また，次に示す監視項目の警告値を決定してください。

- ハードウェアリソースの利用率

  サーバマシンのプロセッサやディスクボリュームなどのハードウェアリソース利用率を監視項目とします。

- グローバルバッファプールの状態

グローバルバッファプールのヒット率を監視項目とします。

- アプリケーションのスループット及び SQL 実行時間
  トランザクションコミット数や SQL 実行時間を監視項目とします。

(2)　ハードウェアリソースの利用率を監視する

サーバマシンのプロセッサやディスクなどのハードウェアリソースの利用率が高くなったことをチューニングの起点とします。ハードウェアリソースの利用率が高くなると，システムのスループットが低下すると考えられます。そのため，ハードウェアリソースの利用率を監視します。

パフォーマンスモニタなどを使用すると，サーバマシンのプロセッサやディスクなどのハードウェアリソースの利用率が高いハードウェアを特定できます。

参考

- JP1/Performance Management - Agent Option for Platform を使用すると，ハードウェアのリソース利用率を確認できます。JP1/Performance Management - Agent Option for Platform については，マニュアル「JP1/Performance Management - Agent Option for Platform（Windows(R)用）」を参照してください。
- 日立ディスクアレイシステムを使用している場合は，Hitachi Tuning Manager - Agent for RAID を使用すると，ディスクアレイ装置のパフォーマンスデータを収集できます。Hitachi Tuning Manager - Agent for RAID については，マニュアル「Hitachi Tuning Manager - Agent for RAID」を参照してください。

(3)　グローバルバッファプールのヒット率を監視する

グローバルバッファプールのヒット率が低下したことをチューニングの起点とします。pdbufls コマンドを実行すると，グローバルバッファプールのヒット率を確認できます。

参考

JP1/Performance Management - Agent Option for HiRDB を使用しても，グローバルバッファプールのヒット率を確認できます。JP1/Performance Management - Agent Option for HiRDB については，マニュアル「JP1/Performance Management - Agent Option for HiRDB」を参照してください。

(4)　アプリケーションのスループット及び SQL 実行時間を監視する

アプリケーションのスループットが低下したことをチューニングの起点とします。統計解析ユティリティ（pdstedit コマンド）のシステムの稼働に関する統計情報を取得すると，アプリケーションのスループットを確認できます。

また，SQL 実行時間警告出力機能を使用して，SQL の実行時間を監視してください。SQL 実行時間警告出力機能については，「SQL 実行時間についての警告情報の出力（SQL 実行時間警告出力機能）」を参照してください。

参考

JP1/Performance Management - Agent Option for HiRDB を利用しても，アプリケーションのスループットを確認できます。

(5)　表の検索効率や格納効率を監視する

表のデータ配置に乱れが生じていることをチューニングの起点とします。データの検索効率や格納効率が低下した表については，KFPH00212-I 又は KFPH22017-I メッセージが出力されるため，これらのメッセージを監視してください。

参考

- 表の再編成が必要になる日を HiRDB が予測する再編成時期予測機能を提供しています。再編成時期予測機能については，「表の再編成時期を予測する方法（再編成時期予測機能）」を参照してください。
- JP1/Performance Management - Agent Option for HiRDB を使用しても，データの検索効率や格納効率が低下した表に関するメッセージを取得できます。

(6)　問題を確認する

(2)～(4)で監視していた監視項目が警告値を超えた場合，チューニングを行います。OS 又は HiRDB のコマンドで監視項目のデータを取得した場合は，以前に取得したデータと現在取得したデータを比較して，警告値を超えたかどうか判断してください。

警告値を超えたハードウェアリソースがプロセッサの場合，OS のコマンドを使用して，HiRDB のプロセスがプロセッサを多く使用しているかどうかを確認してください。

SQL 実行時間警告出力機能の警告が出力された場合，(8)に進みチューニングを行ってください。

(5)で監視していた表のデータ配置に乱れが生じてメッセージが出力された場合，(18)に進みチューニングを行ってください。

参考

JP1/Performance Management - Agent Option for Platform，JP1/Performance Management - Agent Option for HiRDB，又は Hitachi Tuning Manager - Agent for RAID を使用した場合，監視項目のデータに警告値を指定して監視できます。また，その警告値を超えた場合，警告を発する機能があります。

(7)　HiRDB ファイルシステム領域を特定する

(6)でハードウェアリソースや，グローバルバッファプールのヒット率が，警告値を超えた場合，そのハードウェアリソースがどの HiRDB ファイルシステム領域に関係するかを特定します。警告値を超えたハードウェアリソースや，グローバルバッファプールと HiRDB ファイルシステム領域の種類との関係情報を取得するには，各レイヤ間の情報をまとめる必要があります。

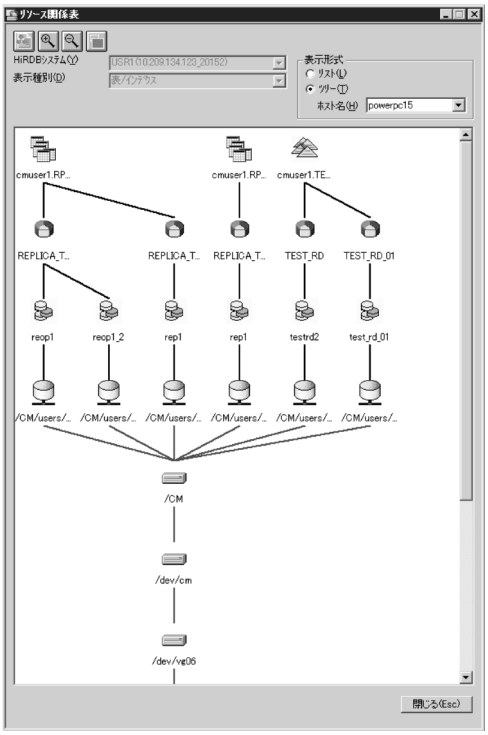
ディスクボリュームと HiRDB ファイルシステム領域間，HiRDB ファイルシステム領域と HiRDB ファイル間，及び HiRDB ファイルと RD エリア間の関係は，ディクショナリ表の SQL_PHYSICAL_FILES を調べると特定できます。

作業表用ファイル用の HiRDB ファイルシステム領域の場合は，pdwork オペランドの指定値を調べると HiRDB ファイルシステム領域名が分かります。さらに，ディクショナリ表の SQL_TABLES 表又は SQL_INDEXES 表を調べると，RD エリアと表（又はインデクス）の関係を特定できます。

参考

- HiRDB CM を使用すると，次の図に示すように，ディスクボリュームから表（又はインデクス）までの関係情報を取得できます。
- 日立ディスクアレイシステムを使用している場合，Hitachi Tuning Manager - Storage Mapping Agent を使用すると，ディスクアレイシステム内のポートや，論理デバイス間の関係情報を取得できます。Hitachi Tuning Manager - Storage Mapping Agent については，マニュアル「Hitachi Tuning Manager - Storage Mapping Agent」を参照してください。

図 23–3　HiRDB CM による関係情報の取得例



(8)　グローバルバッファプールのバッファ面数のチューニングをする

グローバルバッファプールのバッファ面数が小さいため，システムのスループットが低下したり，ハードウェアリソースの利用率が高くなったりしている可能性があります。

(7)で特定した HiRDB ファイルシステム領域と関係のあるグローバルバッファプールのバッファ面数のチューニングを行ってください。グローバルバッファプールのヒット率が低下したことがチューニングの起点の場合は，該当するグローバルバッファプールのバッファ面数をチューニングしてください。アプリケーションのスループット及び SQL の実行時間がチューニングの起点の場合は，すべてのグローバルバッファプールのバッファ面数をチューニングしてください。

グローバルバッファプールのバッファ面数については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。グローバルバッファプールのチューニングについては，「グローバルバッファプールのチューニング」を参照してください。

(9)　チューニング目標を達成したか確認する

(8)で行ったチューニングによって，チューニング目標を達成した場合は，チューニングを終了します。達成できていない場合は，次の手順に進んでチューニングを続けてください。

(10)　チューニング対象の SQL を特定する

チューニング対象の SQL を特定します。また，その SQL のチューニング目標を設定してください。

pdobils コマンドで SQL オブジェクト用バッファの統計情報を取得し，次に示す状況の SQL をチューニング対象とします。SQL オブジェクト用バッファの統計情報を取得すると，各 SQL の実行時間，実行回数，及び入出力回数が確認できます。

また，各 SQL の目標実行時間，目標実行回数，及び目標入出力回数をチューニング目標として決めてください。

- 実行時間が最も長い SQL
- 実行回数が最も多い SQL
- 入出力回数が多い SQL
  - データページ，インデクスページ，ディレクトリページの実 READ 回数が最も多い SQL，LOB 列データページの実 READ 回数が最も多い SQL，又はリストページの実 READ 回数が最も多い SQL
  - データページ，インデクスページ，ディレクトリページの実 WRITE 回数がある SQL，LOB 列データページの実 WRITE 回数がある SQL，又はリストページの実 WRITE 回数がある SQL
  - データページ，インデクスページ，ディレクトリページの更新回数が最も多い SQL，LOB 列データページの更新回数が最も多い SQL，又はリストページの更新回数が最も多い SQL
  - 作業表用ファイルの READ 回数，又は作業表用ファイルの WRITE 回数が最も多い SQL

(11)　チューニング対象の SQL に排他待ちが発生しているか確認する

統計解析ユティリティ（pdstedit コマンド）のシステムの稼働に関する統計情報を取得して，排他待ちが発生しているか確認してください。排他待ちが発生している場合，サーバの排他制御の状態を定期的に取得し，排他待ちが発生しているか確認します。また，排他待ちが発生している資源を該当する SQL が使用している場合は，チューニング対象の SQL に排他待ちが発生していると判断できます。

排他待ちが発生している場合は，(12)でアプリケーションのチューニング（排他制御のチューニング）を行って排他待ちを解消してください。排他待ちが発生していない場合は，(14)で最適化情報の見直しを行ってください。

(12)　アプリケーションのチューニング（排他制御のチューニング）をする

チューニング対象の SQL に排他待ちが発生している場合，アプリケーションのチューニング（排他制御のチューニング）を行います。アプリケーションのチューニング（排他制御のチューニング）については，マニュアル「HiRDB UAP 開発ガイド」の排他制御を参照してください。

(13)　排他待ちが解消したか確認する

アプリケーションのチューニングによって，チューニング対象の SQL に発生していた排他待ちが解消したか確認してください。確認方法については，(11)を参照してください。

排他待ちが解消した場合は，(17)に進みチューニングの目標を達成したか確認してください。排他待ちが解消しない場合は，(14)に進み最適化情報の見直しを行ってください。

(14)　最適化情報の見直し

最適化情報を収集，登録しているかどうかを確認してください。最適化情報収集ユティリティの最適化情報パラメタファイルへの最適化情報の退避を実行して，対象となる SQL で使用している表の最適化情報が，ディクショナリ表に登録されているかどうかで確認できます。最適化情報パラメタファイルへの最適化情報の退避については，「最適化情報パラメタファイルへの最適化情報の退避」を参照してください。

確認の結果，最適化情報を収集，登録していない場合は，(15)の SQL のチューニングへ進んでください。

最適化情報を収集，登録している場合は，次のどれかの対処をしてください。

- SQL 単位でチューニングを実施する場合は，最適化情報を削除してください。削除後，(15)の SQL のチューニングを実施してください。最適化情報の削除については，「最適化情報の削除」を参照してください。
- 最適化情報の更新が必要かどうかを判断し，必要な場合は最適化情報を更新してください。最適化情報の更新については，「最適化情報の更新」を参照してください。
- 最適化情報を更新した後に性能劣化などの問題が発生している場合は，最適化情報を回復してください。最適化情報の回復については，「最適化情報の回復」を参照してください。

(15)　SQL のチューニングをする

SQL のチューニング方法を次の表に示します。

表 23–1　SQL のチューニング方法

SQL の特性	想定要因	詳細	対策
データページ，インデクスページ，ディレクトリページの実 READ 回数，又は参照回数の合計回数が多い（設計時よりも入出力回数が多い，又は設計時よりも実行時間が長い）	表へのアクセスパスが不適切	表の検索範囲が広いのにテーブルスキャンを実施している	マニュアル「HiRDB コマンドリファレンス」のアクセスパス表示ユティリティのチューニングの考え方を参照してください。
		絞り込める条件を指定している検索で，テーブルスキャンを実施している（利用されるインデクスが適切でない）	
		インデクスの検索範囲が広い（必要以上の範囲検索を実施している）	
		サーチ条件がない，又はサーチ条件が有効でないためインデクスのサーチ範囲が絞り込めていない	
		AND を使用して複数の述語を指定した検索を実施している	
	結合処理方法が不適切	ネストループジョイン時の内表の結合キーのインデクスが適切でない	
		外結合するときの利用されるインデクスが適切でない	
	ネストループジョイン時の転送方法が BROADCAST 転送又は KEY RANGE PARTIAL BROAD CAST 転送になっている（HiRDB/パラレルサーバの場合）		
作業表用ファイルへの READ 回数，又は WRITE 回数がある（設計時よりも実行時間が長い）	結合処理方法が不適切	ハッシュジョインの内表件数や副問合せの検索件数が多いため，バケット分割が複数回実行されている	マニュアル「HiRDB UAP 開発ガイド」のハッシュジョイン，副問合せのハッシュ実行を適用する場合の準備を参照してください。
		ソートマージジョインの実施で，大量のデータをソートしている	マニュアル「HiRDB コマンドリファレンス」のアクセスパス表示ユティリティのチューニングの考え方を参照してください。
		クロスジョインを実施している	
	ソート処理実施	ORDER BY 処理のためにソート処理を実施している	

(16)　SQL のチューニング目標を達成したか確認する

チューニング対象の SQL がチューニング目標を達成したか確認してください。SQL オブジェクト用バッファの統計情報を取得し，各 SQL の実行時間，実行回数，入出力回数が目標以内であるか確認してください。目標を達成できない場合は，(15)に戻って再度 SQL のチューニングを行ってください。

(17)　チューニング目標を達成したか確認する

チューニング目標を達成したか確認してください。目標を達成した場合はチューニングを終了します。目標を達成できない場合は，(10)の手順に戻ってチューニング対象の SQL を再度特定し，チューニングを行ってください。

SQL のチューニング，又はアプリケーションのチューニングで効果が出ない場合は，(18)に進みデータベースのチューニングを行ってください。

(18)　データベースのチューニングをする

SQL のチューニング，又はアプリケーションのチューニングで効果が出ない場合は，データベースのチューニングを実施します。次に示すチューニングを行ってください。

- グローバルバッファプールのバッファ面数

  グローバルバッファプールのバッファ面数については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。グローバルバッファプールのチューニングについては，「グローバルバッファプールのチューニング」を参照してください。
- 作業表用バッファ長

  作業表用バッファ長については，マニュアル「HiRDB UAP 開発ガイド」のハッシュジョイン，副問合せのハッシュ実行を適用する場合の準備を参照してください。
- 表の横分割

  表の横分割については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
- データの再配置

  データの再配置については，「データベースのチューニング」を参照してください。
- デファードライト処理

  デファードライト処理のチューニングについては，「デファードライト処理のチューニング」を参照してください。

(19)　チューニングの目標を達成したか確認する

データベースのチューニングによって，チューニングの目標を達成したか確認してください。目標を達成した場合はチューニングを終了します。目標を達成できない場合は，(10)に戻ってチューニング対象の SQL を再度特定し，チューニングを行ってください。

データベースのチューニングで効果が出ない場合は，次に進みハードウェアリソースのチューニングを行ってください。

(20)　ハードウェアリソースのチューニングをする

ハードウェアリソースを追加又は増強します。パフォーマンスモニタなどを使用して，プロセッサの利用率，及びディスクの利用率を取得し，利用率が最も高いハードウェアリソースを追加するか，又は増強してください。

利用率が最も高いハードウェアリソースがプロセッサの場合は，より高性能なプロセッサに変更すると処理時間を短縮できます。また，利用率が最も高いハードウェアリソースがディスクの場合は，より高性能なディスクに変更するか，又はディスクアレイ装置を使用しているときはディスクを追加（スピンドル数を増加）すると，入出力時間を短縮できます。

参考

- JP1/Performance Management - Agent Option for Platform を利用してもプロセッサや，ディスクボリュームの利用率を確認できます。
- 日立ディスクアレイシステムを使用している場合，Hitachi Tuning Manager - Agent for RAID を使用すると，ディスクアレイシステムのパフォーマンスデータを収集できます。

23.10　ロールバック用ログ入力バッファ面数のチューニング

ロールバック用ログ入力バッファは，共用メモリ上にサーバ単位で用意される領域です。システムログの出力に使用するログ出力バッファと異なる点は，ロールバックを行うトランザクション一つにつき，バッファ 1 面を占有して使用する点です。このため，ロールバック用ログ入力バッファの面数によっては，同時に実行できるロールバック処理数が制限されることがあります。

ロールバック用ログ入力バッファの面数をチューニングすることで，同時に実行できるロールバック処理数が増え，ロールバックの処理時間が改善する場合があります。

設定方法

pd_log_rollback_buff_count オペランドに 0 を指定します。0 を指定すると，ロールバック用ログ入力バッファ面数はシステムが自動的に決定します。

注意事項

pd_log_rollback_buff_count オペランドを指定すると，ロールバック用ログ入力バッファ面数が増え，メモリ不足が発生する場合があります。このような場合は，オペランドの指定値を変更前の値に戻してください。

24 セキュリティ監査機能の運用

この章では，セキュリティ監査機能の環境設定，及び運用方法について説明します。

24.1　セキュリティ監査機能の概要

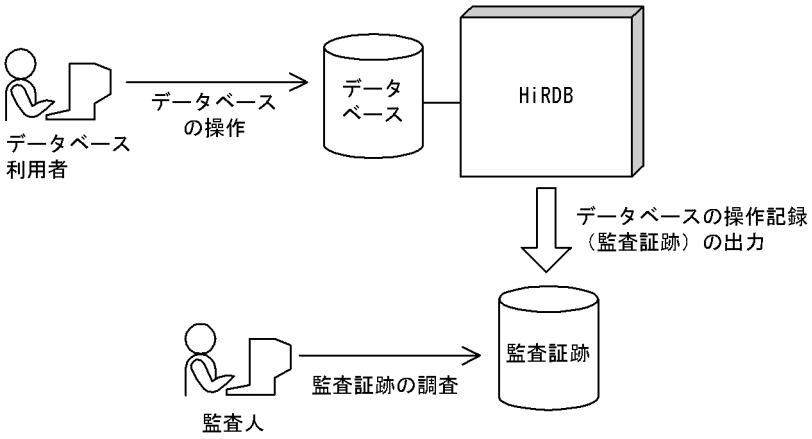
ここでは，セキュリティ監査機能の概要について説明します。説明する項目は次のとおりです。

- セキュリティ監査機能とは
- 監査証跡の取得契機
- 監査証跡の取得例
- 監査証跡として取得する情報
- 監査証跡の参照
- 前提となるシステム構成（HiRDB/パラレルサーバ限定）
- 監査対象になるイベント
- 監査証跡表の自動データロード機能

24.1.1　セキュリティ監査機能とは

HiRDB のセキュリティは権限によって守られています。参照できる情報，更新できる情報，及び操作できるオブジェクト（表又はインデクスなど）を権限によって制限しています。この権限の運用が適切に行われているかどうかをチェックするために，HiRDB ではデータベースに対する各種操作を記録できます。この機能を**セキュリティ監査機能**といい，出力される操作記録を**監査証跡**といいます。出力された監査証跡を調査して不正なアクセスが行われていないかをチェックできます。このチェックは**監査権限**を持つユーザ（これを**監査人**といいます）が行います。セキュリティ監査機能の概要を次の図に示します。

図 24–1　セキュリティ監査機能の概要



監査証跡には，だれがどのような権限を使用して何に対する操作を行ったかという情報が取得されます。どの操作に対して監査証跡を取得するかは，監査人が CREATE AUDIT 文で設定します。監査証跡の取得対象となる操作が実行されると，監査証跡が取得されます。

監査人は，監査証跡表を管理及び操作する主監査人と，監査証跡表を操作する副監査人の 2 種類があります。

主監査人はシステム内に１人，副監査人はシステム内に複数人登録できます。

> **参考**
>
> セキュリティ監査機能はセキュリティを強化する機能ではありません。権限の運用が適切に行われているかどうかをチェックするための操作記録を出力する機能です。

24.1.2　監査証跡の取得契機

HiRDB が監査証跡を取得する契機を次に示します。

- コマンド又は SQL 文を実行するときの権限チェック時
- 各イベントの終了時

SQL の構文エラー時，及びコマンドの入力ミスによるエラー時は監査証跡を取得しません。

(1)　権限チェック時の監査証跡の取得

1 回の権限チェックで一つの監査証跡を取得します。なお，実行した SQL の延長でほかの SQL が実行されることがあります。この場合，延長で実行する SQL についても，権限チェック時に監査証跡を取得します。詳細については，「監査証跡のレコード項目（権限チェック時）」を参照してください。

(2)　イベント終了時の監査証跡の取得

イベントの対象となったオブジェクトの数だけ，イベント終了時に監査証跡を取得します。ただし，アカウントロック期間の経過によって解除された連続認証失敗アカウントロック状態については，アカウントロック期間の経過時ではなく，経過後に実施する次のイベント時に監査証跡を取得します。

- CONNECT の実行時
- DROP CONNECTION SECURITY の実行時
- pdacunlck コマンドの実行時

イベント終了時の監査証跡の出力パターンについては，「監査証跡の出力パターン」を参照してください。

イベント終了時に出力される監査証跡については，「監査証跡のレコード項目（イベント終了時）」を参照してください。

24.1.3　監査証跡の取得例

監査証跡の取得例を次に示します。

（例 1）表を検索した場合の監査証跡の取得例

表を検索した場合，表のアクセス権限（SELECT 権限）を使用するため，監査証跡が取得されます。

表の検索内容（SQL の指定）		監査証跡の内容				
		実行者	使用した権限	操作対象のオブジェクト種別	操作対象のオブジェクト名	操作種別
ユーザ（USR1）が次の SELECT 文を発行した場合 SELECT C1 FROM USR1.T1	権限	USR1	表のアクセス権限（SELECT 権限）	表	USR1.T1	表へのアクセス（SELECT）
	終了	USR1	−	表	USR1.T1	表へのアクセス（SELECT）
ユーザ（USR2）が次の SELECT 文を発行した場合 SELECT T1.C1,T2.C1 FROM USR1.T1 T1,USR2.T2 T2 WHERE T1.C1=T2.C1	権限	USR2	表のアクセス権限（SELECT 権限）	表	USR1.T1	表へのアクセス（SELECT）
		USR2	表のアクセス権限（SELECT 権限）	表	USR2.T2	表へのアクセス（SELECT）
	終了	USR2	−	表	USR1.T1	表へのアクセス（SELECT）
		USR2	−	表	USR2.T2	表へのアクセス（SELECT）

（凡例）

権限：権限チェック時に取得される監査証跡

終了：イベント終了時に取得される監査証跡

−：該当しません。

（例 2）表を定義又は削除した場合の監査証跡の取得例

表を定義又は削除した場合，スキーマ所有者の権限，表の所有者の権限，及び RD エリア利用権限を使用するため，監査証跡が取得されます。

表の検索内容（SQL の指定）		監査証跡の内容				
		実行者	使用した権限	操作対象のオブジェクト種別	操作対象のオブジェクト名	操作種別
ユーザ（USR1）が次の CREATE TABLE を発行した場合 CREATE TABLE T1(C1 INT) IN RDAREA1	権限	USR1	RD エリア利用権限	RD エリア	RDAREA1	定義作成
		USR1	所有者	スキーマ	USR1	定義作成
		USR1	所有者	表	USR1.T1	定義作成
	終了	USR1	−	表	USR1.T1	定義作成
ユーザ（USR2）が次の DROP TABLE を発行した場合	権限	USR2	所有者	表	USR2.T1	定義削除
	終了	USR2	−	表	USR2.T1	定義削除

表の検索内容（SQL の指定）		監査証跡の内容				
		実行者	使用した権限	操作対象のオブジェクト種別	操作対象のオブジェクト名	操作種別
DROP TABLE T1						

（凡例）

権限：権限チェック時に取得される監査証跡

終了：イベント終了時に取得される監査証跡

－：該当しません。

24.1.4　監査証跡として取得する情報

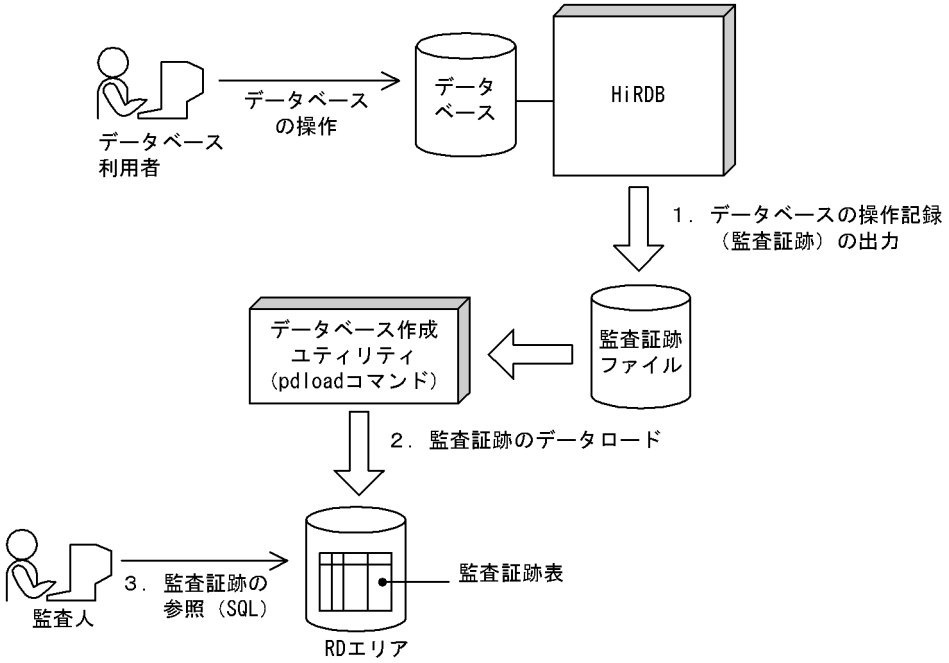
監査証跡として取得する情報については，次に示す箇所を参照してください。

- 「監査証跡ファイルに出力される情報」
- 「監査証跡表の列構成」
- 「監査証跡のレコード項目（権限チェック時）」
- 「監査証跡のレコード項目（イベント終了時）」

24.1.5　監査証跡の参照

監査証跡は**監査証跡ファイル**に出力されます。監査証跡ファイル中のデータをデータベース作成ユティリティ（pdload コマンド）で，**監査証跡表**にデータロードした後に SQL で参照できます。なお，監査人はこの監査証跡表を参照できます（更新はできません）。監査人以外のユーザは，主監査人に参照権限を与えてもらえば監査証跡表を参照できます（更新はできません）。監査証跡の参照方法を次の図に示します。

図 24–2　監査証跡の参照方法



〔説明〕

1. 監査対象のイベントが実行された場合，監査証跡ファイルに監査証跡が出力されます。監査証跡ファイルは監査証跡ファイル用の HiRDB ファイルシステム領域内に作成されます。監査対象のイベントについては，「監査対象になるイベント」を参照してください。
2. 監査証跡ファイルに出力された監査証跡を入力情報にして，データベース作成ユティリティ（pdload コマンド）のデータロードで表にデータを登録します。詳細については，「監査証跡表へのデータ登録」を参照してください。

   なお，監査証跡表へのデータ登録は HiRDB が自動的に行うこともできます。詳細については，「監査証跡表の自動データロード機能」を参照してください。
3. 監査人は監査証跡表を利用して監査を実施します。監査証跡表については，「監査証跡表の列構成」を参照してください。

監査証跡表とほかの表の相違点を次の表に示します。

表 24–1　監査証跡表とほかの表の相違点

表の操作項目	監査証跡表	そのほかの表
表の定義	HiRDB 管理者が pdmod コマンドで定義します。	各ユーザが CREATE TABLE で定義します。
表の削除	主監査人だけが削除できます。DBA 権限保持者は削除できません。	表の所有者が削除します。DBA 権限保持者も削除できます。
表定義の変更	できません。	表の所有者が変更できます。

表の操作項目		監査証跡表	そのほかの表
ほかのユーザへのアクセス権限の付与		主監査人だけが SELECT 権限だけ付与できます。	SELECT 権限，INSERT 権限，UPDATE 権限，及び DELETE 権限のすべてが付与できます。
表へのデータロード		監査人だけが実行できます。	表の所有者が実行できます。アクセス権限の付与によってほかのユーザも実行できます。
表の再編成		監査人だけが実行できます。	DBA 権限保持者が実行できます。アクセス権限の付与によってほかのユーザも実行できます。
表を格納する RD エリアの利用権限		主監査人だけに利用権限があります。	表の所有者に利用権限があります。権限の付与によってほかのユーザも使用できます。
表の横分割		×	○
表へのアクセス	SELECT	○	○
	INSERT	×	○
	UPDATE	×	○
	DELETE	監査人だけができます。	○
	PURGE	監査人だけができます。	○

（凡例）

○：実行できます。

×：実行できません。

24.1.6　前提となるシステム構成（HiRDB/パラレルサーバ限定）

セキュリティ監査機能を使用する場合，システム構成に関して次に示す前提条件があります。

サーバがないユニット（システムマネジャだけのユニットを含む）がある場合，セキュリティ監査機能を使用できません。セキュリティ監査機能を使用する場合は，各ユニットにフロントエンドサーバ，ディクショナリサーバ，又はバックエンドサーバのどれかを配置してください。

24.1.7　監査対象になるイベント

監査証跡の取得対象になる操作を**監査対象イベント**といいます。監査対象イベントを次の表に示します。

なお，セキュリティ監査機能が有効になっている場合はシステムが自動的に監査証跡を出力するイベントと，監査証跡の取得可否を監査人が選択できるイベントがあります。

表 24–2　監査対象イベント

イベントの種類	説明及び監査対象になるイベント	選択可否
システム管理者セキュリティイベント	1. HiRDB 管理者又は DBA 権限保持者が行うセキュリティ対象イベントを監査対象にします。 2. CONNECT 関連セキュリティ機能の設定値の変更を監査対象にします。 3. システムが自動的に行うセキュリティ対象イベントを監査対象にします。 次に示すイベントが実行されたときに監査証跡を出力します。 • HiRDB の開始（pdstart コマンド）※1 • HiRDB の終了（pdstop コマンド）※1※2 • 監査人の登録（pdmod コマンド） • 監査証跡表の作成（pdmod コマンド） • 監査証跡ファイルの削除（pdaudrm コマンド）※3 • 監査証跡の取得開始※5 • 監査証跡の取得終了※6 • 監査証跡ファイルの上書き開始 • 連続認証失敗アカウントロック状態への遷移 • 連続認証失敗アカウントロック状態の解除 次に示す場合が該当します。 ・アカウントロック期間経過後の CONNECT 時 ・DROP CONNECTION SECURITY の実行時 ・pdacunlck コマンドの実行時 • パスワード無効アカウントロック状態への遷移 • パスワード無効アカウントロック状態の解除 • CONNECT 関連セキュリティ機能の設定値の変更 ・連続認証失敗許容回数 ・アカウントロック期間 ・パスワードの文字列制限で設定する項目（事前チェックも含む） ・IP アドレスによる接続制限の登録又は削除 ・パスワードの有効期間の設定 • pdacnlck コマンドの実行	×（必ず監査証跡が出力されます）
監査人セキュリティイベント	監査人が行うセキュリティ対象イベントを監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • 監査証跡表へのデータロード（pdload コマンド） • 監査証跡ファイルのスワップ（pdaudswap コマンド） • 監査対象イベントの定義（CREATE AUDIT）※4 • 監査対象イベントの削除（DROP AUDIT）※4 • 監査人のパスワード変更（GRANT AUDIT）※4 • JP1/NETM/Audit 用監査ログ出力ファイルへの出力（pdaudput コマンド）	×（必ず監査証跡が出力されます）
セッションセキュリティイベント	認可識別子とパスワードによるユーザ認証を監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • HiRDB への接続（CONNECT 文） • ユーザの変更（SET SESSION AUTHORIZATION 文）	○

イベントの種類	説明及び監査対象になるイベント	選択可否
	• ロールの設定（SET ROLE 文） • HiRDB との切り離し（DISCONNECT 文）※9	
権限管理イベント	ユーザ権限の付与又は削除を監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • ユーザ権限の付与（GRANT 文） • ユーザ権限の削除（REVOKE 文）	○※7
オブジェクト定義イベント	オブジェクトの定義，削除，又は変更を監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • オブジェクトの定義（次に示す SQL が対象になります） CREATE FUNCTION CREATE INDEX CREATE PROCEDURE CREATE PUBLIC VIEW CREATE ROLE CREATE SCHEMA CREATE SEQUENCE CREATE TABLE CREATE TRIGGER CREATE TYPE CREATE VIEW • オブジェクトの削除（次に示す SQL が対象になります） DROP DATA TYPE DROP FUNCTION DROP INDEX DROP PROCEDURE DROP PUBLIC VIEW DROP ROLE DROP SCHEMA DROP SEQUENCE DROP TABLE DROP TRIGGER DROP VIEW • オブジェクトの変更（次に示す SQL が対象になります） ALTER INDEX ALTER PROCEDURE ALTER ROUTINE ALTER TABLE ALTER TRIGGER COMMENT	○※7
オブジェクト操作イベント	オブジェクトの操作を監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • 表の検索（SELECT 文）	○※7

イベントの種類	説明及び監査対象になるイベント	選択可否
	• 表への行挿入（INSERT 文） • 表の行更新（UPDATE 文） • 表からの行削除（DELETE 文） • 表の全行削除（PURGE TABLE 文） • ストアドプロシジャの実行（CALL 文） • 表の排他制御（LOCK TABLE 文） • リストの作成（ASSIGN LIST 文） • 順序数生成子が生成する値の返却（NEXT VALUE 式）	
ユティリティ操作イベント	ユティリティ又はコマンドによるオブジェクト操作に関するセキュリティ対象イベントを監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • データベース作成ユティリティ（pdload コマンド） 対象オブジェクト：TABLE，SEQUENCE • pddefrev コマンド 対象オブジェクト：PROCEDURE，TABLE，TRIGGER，VIEW • データベース再編成ユティリティ（pdrorg コマンド） 対象オブジェクト：TABLE，INDEX※１０ • ディクショナリ搬出入ユティリティ（pdexp コマンド） 対象オブジェクト：PROCEDURE，TABLE，TRIGGER，VIEW • 整合性チェックユティリティ（pdconstck コマンド） 対象オブジェクト：TABLE	○※７※８

（凡例）

○：選択できます。

×：選択できません。

注※１

HiRDB/パラレルサーバのサーバ単位の開始又は終了は監査対象イベントになりません。

注※２

正常終了又は計画停止を監査対象イベントとします。強制終了又は異常終了は監査対象イベントになりません。強制終了又は異常終了を監査するには，HiRDB が出力するメッセージ，又は OS が出力するメッセージを使用してください。

監査対象にならない終了コマンドを次に示します。

- pdstop -f
- pdstop -f -q
- pdstop -f -x ホスト名
- pdstop -f -u ユニット識別子
- pdstop -f -s サーバ名
- pdstop -f -u ユニット識別子 -s サーバ名

- pdstop -z
- pdstop -z -q
- pdstop -z -c
- pdstop -z -s サーバ名

注※３

監査証跡ファイルの作成は監査対象イベントになりません。監査証跡ファイルの作成を監査するには OS の監査機能を使用してください。

注※４

データベース定義ユティリティ（pddef コマンド）又は会話型 SQL 実行ユティリティで実行した場合も監査証跡を出力します。

注※５

pdaudbegin コマンドの実行時又は HiRDB の開始時から監査証跡を取得する場合に監査証跡を出力します。

注※６

pdaudend コマンドの実行時又は監査証跡を取得している状態で，HiRDB を正常終了又は計画停止する場合に監査証跡を出力します。

注※７

権限管理イベント，オブジェクト定義イベント，オブジェクト操作イベント，及びユティリティ操作イベント中のイベント対象オブジェクトが，監査証跡表，監査証跡表を基表としたビュー表，又は監査証跡表を基としたリストの場合，イベント終了時の監査証跡は無条件に出力されます。権限チェック時の監査証跡を取得するかどうかは選択できます。ただし，監査証跡表に対するデータベース作成ユティリティ（pdload コマンド）は，監査人セキュリティイベントに含まれるので，イベント終了時及び権限チェック時の監査証跡は無条件に出力されます。

注※８

データベース再編成ユティリティ（pdrorg コマンド）でディクショナリ表のリロードをする場合，監査証跡は無条件に出力されます。

注※９

次の場合を監査証跡イベントとします。

- シングルサーバ又はフロントエンドサーバのサーバプロセスが DISCONNECT を検知した場合
- シングルサーバ又はフロントエンドサーバのサーバプロセスが内部的に DISCONNECT を実行する場合

注※１０

ディクショナリ表のメンテナンスでインデクスを追加した場合

24.1.8　監査証跡表の自動データロード機能

(1)　監査証跡表の自動データロード機能とは

監査証跡表の自動データロード機能とは，次に示す契機で監査証跡ファイルがスワップした場合に，HiRDBが自動的にデータベース作成ユティリティ（pdload コマンド）を起動し，監査証跡ファイルの内容を監査証跡表にデータロードする機能です。この機能を適用することで，監査人によるデータロードの手間が省け，監査証跡表を参照するだけで監査が実施できます。

自動データロードの対象となる監査証跡ファイルのスワップ契機は，次のとおりです。

- 監査証跡ファイルが満杯になった場合
- pdaudswap コマンドが実行された場合
- HiRDB を再開始した場合

注意事項

- 監査証跡ファイルの障害によるスワップの場合は，自動データロードの対象にはなりません。
- HiRDB/パラレルサーバの場合，pdload の起動はユニット単位となります。

(2)　適用条件

監査証跡表の自動データロード機能には，監査人の負荷が軽減されるメリットがありますが，オンライン業務中にデータロードが実行されることでシステムの負荷が高まり，処理性能に影響が出るデメリットもあります。適用の有無については，次の表に示すメリットとデメリット，環境及び運用方法を考慮した上で判断してください。

表 24–3　監査証跡表の自動データロード機能の適用有無によるメリットとデメリット

適用有無	メリット	デメリット	推奨する環境・運用方法
適用する	• pdload コマンドは実行不要で，監査証跡表を参照するだけですぐに監査できる。 • 監査証跡表の容量や状態だけを管理すればよいため，監査人の負荷が軽減される。	• トラフィックに関係なく，監査証跡ファイルがスワップするとデータロードが実行されるため，業務に影響が出るおそれがある。 • 監査証跡表を参照するのは監査人とシステム（pdload コマンド）の両方なので，排他制御への意識が必要になる。	• ハードウェアの性能が高く，オンライン業務中に pdload コマンドが実行されても処理性能に影響がない環境である。 • 監査証跡表の参照は NOWAIT 検索の結果で十分である。 • 常に監査証跡をチェックしたい。
適用しない	• 比較的トラフィックが空いているときなど，任意の時間帯にデータロードできる。	• 監査を実施するには，pdload コマンドを実行する必要がある。 • 監査証跡ファイルと監査証跡表の両方の容量や状態を管理する必要がある。	• オンライン業務中に処理性能への影響があってはならない環境である。 • 監査証跡表の参照は NOWAIT 検索の結果では不十分である。

適用有無	メリット	デメリット	推奨する環境・運用方法
	• 監査証跡表を参照するのは監査人だけなので，排他制御への意識は不要である。		• 問題が発生した場合だけ監査証跡をチェックしたい。

24.2　監査証跡ファイルに出力される情報

監査証跡ファイルに出力される情報を次の表に示します。

表 24–4　監査証跡ファイルに出力される情報

取得する情報	説明
ユーザ識別子	監査対象のイベント実行者の認可識別子
イベント実行日	イベントを実行した年月日
イベント実行時刻	イベントを実行した時刻
イベント実行時刻	イベントを実行した時刻（単位：マイクロ秒）
イベントタイプ	イベントタイプ（表「イベントタイプ及びイベントサブタイプの詳細」を参照）
イベントサブタイプ	イベントサブタイプ（表「イベントタイプ及びイベントサブタイプの詳細」を参照）
イベント成否	イベントの実行結果（権限のチェックが成功したかどうかを出力します）
使用した権限	イベントを実行したときに使用した権限
UAP 名称	クライアント環境定義の PDCLTAPNAME オペランドに指定した UAP 名称
サービス名	イベント発行元の UAP が要求したサービス名 OpenTP1 の SUP（サービス利用プログラム）が SPP（サービス提供プログラム）に要求したサービスの場合，又は TP1/Message Control が MHP（メッセージ処理プログラム）に要求したサービスの場合は，該当するサービス名称
IP アドレス	イベント発行元 UAP を実行したクライアントの IP アドレス※
プロセス番号	イベント発行元 UAP のプロセス ID※
スレッド番号	イベント発行元 UAP のスレッド ID※
ホスト名	イベント発行元 UAP の接続先ホスト名
ユニット識別子	イベント発行元 UAP の接続先ユニット識別子
サーバ名	イベント発行元 UAP の接続先フロントエンドサーバ名，又はシングルサーバ名
コネクト通番	イベント発行者のコネクト通番
SQL 通番	イベントの SQL 通番
オブジェクトの所有者名	イベントの権限チェックの対象になるオブジェクトの所有者名
オブジェクト名	イベントの権限チェックの対象になるオブジェクト名
オブジェクトの種別	イベントの権限チェックの対象になるオブジェクトの種別
付与，削除，又は変更した権限	イベントによって付与，削除，又は変更した権限

取得する情報	説明
次のどちらかを取得 • 権限を付与，削除，又は変更されたユーザ識別子 • イベント対象のユーザ識別子	• イベントによって権限を付与，削除，又は変更されたユーザの識別子 • イベント対象になった認可識別子
セキュリティ監査機能に関するオペランドの値	セキュリティ監査機能に関するオペランドの値（HiRDB 開始時の値）
監査証跡種別	権限チェックか，又はイベント終了かを示す種別
SQL コード又は終了コード	SQL，ユティリティ，コマンド終了時のコード
スワップ元の監査証跡ファイル名	スワップ発生時のスワップ元の監査証跡ファイル名
スワップ先の監査証跡ファイル名	スワップ発生時のスワップ先の監査証跡ファイル名
CONNECT 関連セキュリティ機能の設定変更種別	CONNECT 関連セキュリティ機能の設定変更種別（パスワードの変更時にも変更種別が設定されます）
CONNECT 関連セキュリティ機能に関するオペランドの値（変更前）	変更前の CONNECT 関連セキュリティ機能に関するオペランドの値
CONNECT 関連セキュリティ機能に関するオペランドの値（変更後）	変更後の CONNECT 関連セキュリティ機能に関するオペランドの値
監査証跡表オプション	イベントの操作対象が監査証跡表，監査証跡表を基表としたビュー表，又は監査証跡表を基表としたリストの場合のフラグ
アクセス件数	イベントによってオブジェクト（実表，ビュー表，及びリスト）に対して検索，挿入，更新，及び削除をした行数
SQL 文	実行した SQL 文
SQL データ	実行した SQL のデータ
ユーザ付加情報 1	ユーザが任意に設定する付加情報
ユーザ付加情報 2	
ユーザ付加情報 3	
関連製品付加情報 1	Cosminexus が設定する付加情報

注

イベントによって取得する情報が異なります。イベントごとに取得する情報の一覧については，「監査証跡のレコード項目（権限チェック時）」及び「監査証跡のレコード項目（イベント終了時）」を参照してください。

注※

OpenTP1 配下のアプリケーションを介している場合，又は Web サーバなどの製品を介している場合は，エンドユーザが実行しているアプリケーションの情報ではなく，HiRDB に接続しているアプリケーションの情報が取得されます。

24.3　監査証跡の出力パターン

ここでは，監査証跡の出力パターンについて説明します。

24.3.1　権限チェック時の出力パターン

1 回の権限チェックで一つの監査証跡が取得されます。ただし，次の表に示す場合は例外的に 1 回の権限チェックで複数の監査証跡が取得されます。

表 24–5　イベントの実行単位と監査証跡レコード出力単位の詳細

イベント名称	実行単位	出力単位の対象	出力レコード数
GRANT， REVOKE	ユーザ	ユーザ	対象ユーザ数
	ロール	ロール	対象ロール数
CREATE CONNECTION SECURITY， DROP CONNECTION SECURITY	CONNECT 関連セキュリティ機能の設定値	CONNECT 関連セキュリティ機能の設定値	CONNECT 関連セキュリティ機能の設定値数

一つのイベントで権限チェックが複数回ある場合，監査証跡は次のように出力されます。

- **二つの権限のうちどちらかが必要な場合**
  どちらか一方の権限チェック時の監査証跡を出力します。
  （例）他人の表をアンロードする場合，DBA 権限又は SELECT 権限が必要になります。
  DBA 権限チェックが成功したかどうか，又は SELECT 権限チェックが成功したかどうかを出力します。
- **二つの権限が両方とも必要な場合**
  両方の権限チェック時の監査証跡を出力します。
  （例）他人の表を再編成する場合，INSERT 権限，DELETE 権限，及び SELECT 権限が必要になります。
  INSERT 権限，DELETE 権限，及び SELECT 権限のすべての権限チェックが成功した場合は，すべて出力します。処理の途中でエラーになった場合，それまでの権限チェックが成功しているときは成功の証跡を出力し，失敗しているときは失敗の証跡を出力します。

24.3.2　イベント終了時の出力パターン

一つのイベントに対して複数の監査証跡（イベントの対象となるオブジェクト数分の監査証跡）が取得されます。基本的には，監査証跡表の操作対象オブジェクト情報以外の列は，すべて同じ内容を出力します。ただし，ユティリティなど，操作対象ごとにイベントの結果が異なる場合は，終了コードごとに異なる内容を出力します。監査証跡の出力パターンの例を次に示します。

(1)　対象オブジェクト又は対象ユーザが複数の場合

(a)　対象オブジェクトが複数の場合

例えば，SELECT * FROM "T1","T2"と実行した場合，次に示すようにオブジェクト T1 と T2 に関して 2 行の監査証跡が出力されます。このとき，終了コードはすべて同じ値が記録されます。

...	イベントサブタイプ	...	オブジェクト名称	...	終了コード	...
...	SEL	...	T1	...	XXX	...
...	SEL	...	T2	...	XXX	...

ルーチンの再作成，スキーマ単位のデータベース再編成ユティリティなどは，オブジェクトによってイベントの成功と失敗に分かれることがあります。その場合は，オブジェクトごとに異なる終了コードを出力します。対象オブジェクトが複数になる場合があるイベントの一覧を次の表に示します。

表 24–6　対象オブジェクトが複数になる場合があるイベントの一覧（イベントの実行単位と監査証跡レコード出力単位の詳細）

イベント名称	実行単位	出力単位の対象	出力レコード数	SQL コード又は終了コード
GRANT， REVOKE	ユーザ	ユーザ	対象ユーザ数	イベント対象オブジェクトに対してすべて同じ値
	ロール	ロール	対象ロール数	
CREATE CONNECTION SECURITY， DROP CONNECTION SECURITY	CONNECT 関連セキュリティ機能の設定値	CONNECT 関連セキュリティ機能の設定値	CONNECT 関連セキュリティ機能の設定値数	
操作系 SQL	実表	• 実表 • 順序数生成子	対象実表数 ＋対象順序数生成子数	
	ビュー表	• ビュー表 • 順序数生成子	対象ビュー表数 ＋対象順序数生成子数	
ALTER ROUTINE， ALTER PROCEDURE， ALTER TRIGGER	複数のルーチン又はトリガ※1	プロシジャ，関数，トリガ	対象ルーチン数分	イベント対象オブジェクトごとに異なる値
pdload	• 実表 • RD エリア	• 実表 • 順序数生成子	対象実表数 ＋対象順序数生成子数	
pdexp， pddefrev	実表	実表	対象実表数	
	ビュー表	ビュー表	対象ビュー表数	
	ストアドプロシジャ	ストアドプロシジャ	対象ストアドプロシジャ数	
	トリガ	トリガ	対象トリガ数	

イベント名称	実行単位	出力単位の対象	出力レコード数	SQL コード又は終了コード
pdrorg	スキーマ※2，実表，インデクス	実表	対象実表数	
	ディクショナリ表のメンテナンス（インデクスの追加）	インデクス	対象インデクス数	
pdaudbegin，pdaudend	ユニット	ユニット	対象ユニット数	

注※1

ルーチン情報が取得できない場合は，オブジェクト情報に NULL 値を記録して監査証跡を 1 行だけ出力します。

注※2

スキーマ単位の再編成の場合，スキーマ内の表情報が取得できないときは，オブジェクト情報にスキーマ情報を記録して監査証跡を 1 行だけ出力します。

(b)　副問い合せの場合

例えば，UPDATE "T1" SET "C1" = (SELECT "C1" FROM "T2" WHERE CODE = '01') WHERE CODE = '01'と実行した場合，(a)と同じくアクセスしたオブジェクト情報を出力します。このとき，イベントタイプとサブタイプには，オブジェクトに対するアクションの監査証跡を次に示すように出力します。このとき，各イベント終了時の監査証跡の SQL コードはすべて同じ値が記録されます。

...	イベントサブタイプ	...	オブジェクト名称	...	終了コード	...
...	UPD	...	T1	...	XXX	...
...	SEL	...	T2	...	XXX	...

(2)　内部的に動作する SQL の場合

実行した SQL の延長でほかの SQL が実行されることがあります。イベントを実行したときに，延長で実行される SQL を監査対象として定義している場合，延長で実行される SQL の対象オブジェクト又はユーザの情報を取得します。このとき，オブジェクト又はユーザに対するイベントタイプとサブタイプには，延長で実行された SQL のイベントタイプとサブタイプが記録されます。また，各監査証跡の SQL コードはどのオブジェクトに対してもすべて同じ値が記録されます。

例えば，DROP SCHEMA の延長で表が削除された場合，スキーマの情報とは別に，アクセスしたオブジェクトとして，表などのオブジェクト情報を出力します。削除処理中にエラー終了した場合は，エラーになるまでにアクセスしたオブジェクトとユーザ数分の監査証跡を出力します。

プロシジャやトリガなども，SQL の延長で更に自動的に SQL が実行されます。この場合，延長で実行される SQL ごとにイベント終了の監査証跡を出力します。トリガの場合は，実行者が一時的にトリガ所有者に

切り替わるため，イベントの実行者にトリガ所有者が記録されます。プロシジャの場合は，CALL 文のイベント終了の監査証跡も出力します。プロシジャがネストしていた場合は，各 CALL 文に対して，イベント終了の監査証跡を出力します。トリガ及びプロシジャ実行時の監査証跡レコード出力単位の詳細を次の表に示します。

表 24–7　トリガ及びプロシジャ実行時の監査証跡レコード出力単位の詳細

種別		イベントの実行者（認可識別子の値）	監査証跡レコードの出力単位
トリガ	ユーザトリガ	トリガ所有者	トリガ内 SQL 単位
	システムトリガ（参照制約などの内部トリガ）	トリガ所有者	システムトリガ内 SQL 単位
プロシジャ		プロシジャ実行者	プロシジャ内 SQL 単位＋ CALL 文

トリガやプロシジャ内の SQL でエラーが発生した場合，その時点の SQL のイベントの結果，そのトリガの契機となった SQL のイベントの結果，CALL 文の結果に同じ SQL コードを出力します。トリガ中のエラー発生箇所と監査証跡（SQL コード）の詳細とプロシジャ中のエラー発生箇所と監査証跡（SQL コード）の詳細を次の表に示します。

表 24–8　トリガ中のエラー発生箇所と監査証跡（SQL コード）の詳細

BEFORE トリガの SQL 結果[1]	トリガ契機の SQL 結果 [2]	AFTER トリガの SQL 結果 [3]	監査証跡（SQL コード）の詳細		
			[1]の監査証跡	[2]の監査証跡	[3]の監査証跡
エラー	－	－	[1]の SQL コード	[1]の SQL コード	×
正常	エラー	－		[2]の SQL コード	×
	正常	エラー		[3]の SQL コード	[3]の SQL コード
		正常	0	0	0

（凡例）

－：該当しません。

×：監査証跡（SQL コード）を出力しません。

表 24–9　プロシジャ中のエラー発生箇所と監査証跡（SQL コード）の詳細

ネストの有無	監査証跡取得対象イベントの詳細					監査証跡（SQL コード）の詳細				
	[1]	[2]	[3]	[4]	[5]	[1]の監査証跡	[2]の監査証跡	[3]の監査証跡	[4]の監査証跡	[5]の監査証跡
なし	エラー	－	－	－	エラー	[1]の SQL コード	×	×	×	[1]の SQL コード
	正常	－	－	－	エラー	0	×	×	×	[5]の SQL コード
					正常	0	×	×	×	0

ネストの有無	監査証跡取得対象イベントの詳細					監査証跡（SQL コード）の詳細				
	[1]	[2]	[3]	[4]	[5]	[1]の監査証跡	[2]の監査証跡	[3]の監査証跡	[4]の監査証跡	[5]の監査証跡
あり	エラー	－	－	－	エラー	[1]の SQL コード	×	×	×	[1]の SQL コード
	正常	エラー	エラー	－	エラー	0	[2]の SQL コード	[2]の SQL コード	×	[2]の SQL コード
		正常	エラー	－	エラー	0	0	[3]の SQL コード	×	[3]の SQL コード
			正常	エラー	エラー	0	0	0	[4]の SQL コード	[4]の SQL コード
				正常	エラー	0	0	0	0	[5]の SQL コード
					正常	0	0	0	0	0

（凡例）

－：該当しません。

×：監査証跡（SQL コード）を出力しません。

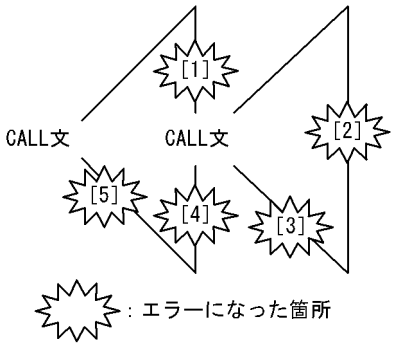
[1]：ネスト前プロシジャ中の SQL の結果

[2]：ネスト後プロシジャ中の SQL の結果

[3]：ネストしている CALL 文の結果

[4]：ネスト前プロシジャ中の SQL の結果

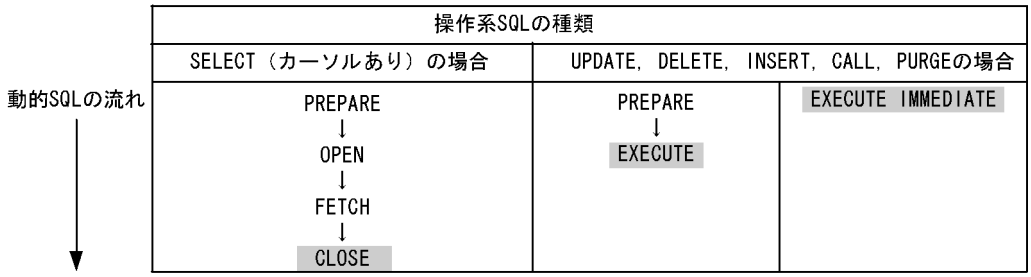
[5]：CALL 文の結果



(3)　動的 SQL の場合

動的 SQL の場合は，PREPARE から実行までを一つの SQL とし，その単位でイベント終了の監査証跡を一つ出力します。実行時のイベント終了時の監査証跡出力タイミングは，OPEN，CLOSE，又は EXECUTE 完了時です。操作系 SQL の種類による動的 SQL の流れを次の図に示します。

図 24–3　操作系 SQL の種類による動的 SQL の流れ



〔説明〕

網掛けしている SQL が監査証跡を出力する契機です。ただし，PREPARE，CLOSE，及び OPEN の監査証跡は，エラーの発生箇所によって監査証跡の取得方法が異なります。動的 SQL 実行時のイベント成否による監査証跡の有無を次の表に示します。

表 24–10　動的 SQL 実行時のイベント成否による監査証跡の有無

SQL 結果			取得指定		
			成功	失敗	両方
PREPARE 成功	OPEN(EXECUTE)成功	CLOSE 成功	CLOSE(EXECUTE)で出力する	出力しない	CLOSE(EXECUTE)で出力する
		CLOSE 失敗	出力しない	CLOSE(EXECUTE)で出力する	CLOSE(EXECUTE)で出力する
	OPEN(EXECUTE)失敗		出力しない	OPEN(EXECUTE)で出力する	
PREPARE 失敗	−		出力しない	PREPARE で出力する	
EXECUTE IMMEDIATE			EXECUTE IMMEDIATE で出力する		

（凡例）−：該当しません。

注

明示的に CLOSE 文を実行しなくても，内部的に CLOSE 文を実行してカーソルを閉じる契機で，イベント終了の監査証跡を出力します。契機については，マニュアル「HiRDB SQL リファレンス」の CLOSE 文を参照してください。

24.3.3　監査証跡の関連づけ

権限チェック時とイベント終了時の監査証跡の関連づけ，及びイベントに対する複数レコードの関連づけは，次に示す情報から調べてください。

- SQL の場合は，監査証跡に出力されたサーバ名称，CONNECT 通番，SQL 通番の組み合わせから調べます。

- コマンド及びユティリティの場合は，監査証跡に出力されたプロセス ID，ホスト名の組み合わせから調べます。

監査証跡の出力内容については，「監査証跡表の列構成」を参照してください。

24.4　環境設定方法

セキュリティ監査機能の環境設定を次に示す手順で実行してください。

〈手順〉

1. セキュリティ監査機能に関する HiRDB システム定義のオペランドを指定します。
2. 監査証跡ファイル用の HiRDB ファイルシステム領域を作成します。
3. 監査人の登録，監査証跡表を格納する RD エリアの作成，及び監査証跡表を作成します。
4. 監査対象イベントを定義します。

1〜3 は HiRDB 管理者がする操作で，4 は監査人がする操作です。また，手順の数字はこの後で説明している項レベルに対応しています。例えば，3 の操作は 22.4.3 で説明しています。

24.4.1　セキュリティ監査機能に関するオペランドの指定

実行者　HiRDB 管理者

セキュリティ監査機能を使用する場合は次の表に示すオペランドを指定します。

表 24–11　セキュリティ監査機能を使用する場合に指定するオペランド

オペランド名	説明
pd_audit	HiRDB の開始時から監査証跡を取得するかどうかを指定します。 **Y：**HiRDB の開始時から監査証跡を取得します。 **N：**HiRDB の開始時から監査証跡を取得しません。 このオペランドに N を指定しても，pdaudbegin コマンドを実行すると監査証跡を取得できます。 pd_audit オペランドに Y を指定するか，又は pdaudbegin コマンドを実行した場合，次に示すイベントの監査証跡を無条件に取得します。 • システム管理者セキュリティイベント • 監査人セキュリティイベント そのほかのイベントについては，CREATE AUDIT で監査証跡を取得するかどうかを定義します。詳細については，「監査対象イベントの定義」を参照してください。
pd_aud_file_name	監査証跡ファイル用の HiRDB ファイルシステム領域名を指定します。この HiRDB ファイルシステム領域内に HiRDB が監査証跡ファイルを作成します。**セキュリティ監査機能を使用する場合はこのオペランドを必ず指定してください。**指定しないとセキュリティ監査機能を使用できません。 このオペランドを指定した場合，HiRDB（HiRDB/パラレルサーバの場合はユニット）の開始時に監査証跡ファイル用の HiRDB ファイルシステム領域に対するアクセスエラーが発生すると，HiRDB を開始できません。 なお，HiRDB/パラレルサーバでセキュリティ監査機能を使用する場合，システム全体で監査証跡を取得することを推奨します。システム全体で監査証跡を取得するには，次のどちらかの指定をしてください。

オペランド名	説明
	• システム共通定義で pd_aud_file_name オペランドを指定する • すべてのユニット制御情報定義で pd_aud_file_name オペランドを指定する ただし，同一サーバマシンで複数のユニットを稼働させるシステム構成の場合は，すべてのユニット制御情報定義で pd_aud_file_name オペランドを指定する必要があります。
pd_aud_max_generation_size	1 監査証跡ファイルの最大容量を指定します。
pd_aud_max_generation_num※	監査証跡ファイル用の HiRDB ファイルシステム領域内に作成する監査証跡ファイルの最大数（世代数）を指定します。
pd_aud_no_standby_file_opr	スワップ先にできる監査証跡ファイルがない場合，及び非同期出力用バッファのすべての面がフラッシュ待ちになった場合の HiRDB の処理方式を指定します。 **forcewrite（省略値）：** スワップ先にできる監査証跡ファイルがない場合，データロード待ち（閉塞状態のファイルを除く）の監査証跡ファイルを強制的にスワップ先にして監査証跡の出力を続行します。監査証跡ファイルの状態については，「監査証跡ファイルの運用」を参照してください。 非同期出力用バッファのすべての面がフラッシュ待ちになってから 180 秒経過した場合，最初にフラッシュ待ちになった非同期出力用バッファを強制的に上書きして再利用することで，監査証跡の出力を続行します。上書きされた非同期出力用バッファの監査証跡は破棄されます。 **down：** スワップ先にできる監査証跡ファイルが残り一つになった場合，又は非同期出力用バッファのすべての面がフラッシュ待ちになってから 180 秒経過した場合，HiRDB（HiRDB/パラレルサーバの場合はユニット）を強制終了します。
pd_aud_async_buff_size	監査証跡を非同期に出力する場合に使用するバッファ長を指定します。
pd_aud_async_buff_count	監査証跡を非同期に出力する場合に使用するバッファ面数を指定します。
pd_aud_async_buff_retry_intvl	監査証跡を非同期に出力する場合に使用するバッファがすべて使用中のとき，未使用のバッファが確保できるまでバッファを監視する処理のリトライ間隔を指定します。
pd_aud_file_wrn_pnt	スワップ先にできない監査証跡ファイル数が警告値以上になったとき，警告メッセージを出力します。このオペランドには警告値を，pd_aud_max_generation_num オペランドに指定した監査証跡ファイルの最大数に対する比率で指定します。
pd_aud_auto_loading	監査証跡表の自動データロード機能を使用するかどうかを指定します。 **Y：**監査証跡表の自動データロード機能を使用します。この場合，監査証跡ファイルのスワップ（障害によるスワップは除く）を契機として，監査証跡表へデータロードする処理が自動で実行されます。 **N：**監査証跡表の自動データロード機能を使用しません。この場合，監査証跡表へのデータロード処理は自動で実行されません。監査人が手動で実行してください。 HiRDB/パラレルサーバでこのオペランドに Y を指定する場合は，システムマネジャがあるユニットに，監査証跡ファイル用の HiRDB ファイルシステム領域を作成し，pd_aud_file_name オペランドを指定する必要があります。指定しないと，監査証跡表の自動データロード機能を使用できません。
pdaudload	監査証跡表の自動データロード機能を使用する場合で，運用方法が次のどれかの条件に該当するときは，pdaudload で pdload のオプションを変更します。pdaudload については，マニュアル「HiRDB システム定義」を参照してください。

オペランド名	説明
	• 監査証跡表を格納した RD エリアのバックアップを取得しない場合 →ログ取得方式を「ログ取得モード」に変更してください。 • 監査証跡表にインデクスが定義されていて，インデクス一括作成モードでデータロードする場合 →インデクス作成方法を「インデクス一括作成モード」に変更してください。 • ローカルバッファを使用してデータロードする場合 →ローカルバッファ面数を指定してください。

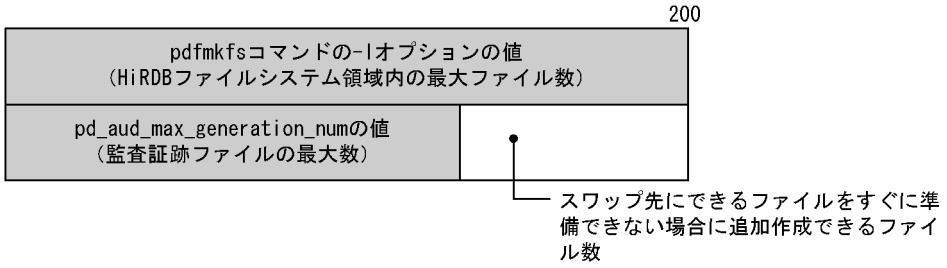
注※

pd_aud_max_generation_num オペランドの値は次に示す条件を満たすように指定してください。

- **pd_aud_max_generation_num の値＜ pdfmkfs コマンドの-l オプションの値**

-l オプションには，この後で説明する監査証跡ファイル用の HiRDB ファイルシステム領域内の最大ファイル数を指定します。この条件を満たすようにオペランドの値を設定すると，スワップ先にできるファイルをすぐに準備できない場合，pd_aud_max_generation_num オペランドの値を大きくすることで対応できます。推奨する設定方法を次の図に示します。

図 24–4　推奨する pd_aud_max_generation_num の値と-l オプションの値の関係



24.4.2　監査証跡ファイル用の HiRDB ファイルシステム領域の作成

実行者　HiRDB 管理者

pdfmkfs コマンドで監査証跡ファイル用の HiRDB ファイルシステム領域を作成します。この HiRDB ファイルシステム領域内に HiRDB が監査証跡ファイルを作成します。監査証跡ファイル用の HiRDB ファイルシステム領域作成時の注意事項を次に示します。

- pd_aud_file_name オペランドで指定した HiRDB ファイルシステム領域名を指定してください。
- HiRDB/パラレルサーバの場合は各ユニットに監査証跡ファイル用の HiRDB ファイルシステム領域を作成してください。
- -k オプション（使用目的）には SYS 又は SVR を指定してください。推奨値は SYS です。
- -n オプション（HiRDB ファイルシステム領域長）の指定値の目安を次に示します。
  推奨値：pd_aud_max_generation_size の値×200 ＋ 20（単位：メガバイト）

最小値：pd_aud_max_generation_size の値×pd_aud_max_generation_num の値＋ 20（単位：メガバイト）

- 監査証跡ファイル用の HiRDB ファイルシステム領域に，ほかのシステムファイルを配置する場合，-n オプションの指定値には監査証跡ファイルの必要容量だけではなく，配置するシステムファイルの必要容量も加算してください。監査証跡ファイルの必要容量だけで HiRDB ファイルシステム領域を確保すると，HiRDB の稼働中に監査証跡ファイルの容量が不足することがあります。

- -l オプション（HiRDB ファイルシステム領域内の最大ファイル数）の指定値の目安を次に示します。

  推奨値：200（pd_aud_max_generation_num の最大値）

  最小値：pd_aud_max_generation_num の値

24.4.3 監査人の登録，監査証跡表を格納する RD エリアの作成，及び監査証跡表の作成

実行者　HiRDB 管理者

データベース構成変更ユティリティ（pdmod コマンド）を実行して次に示すことをしてください。次に示す操作は 1 回の pdmod コマンドで同時に実行できます。

- 監査人の登録
- 監査証跡表を格納する RD エリアの作成
- 監査証跡表の作成

(1)　監査人の登録

pdmod コマンドの **create auditor 文**で監査人を登録します。監査人は次に示す操作などができます。

- 監査証跡表へのデータロード
- 監査証跡ファイルのスワップ
- 監査証跡表の検索及び削除

注意事項

- 監査人登録時の注意事項は，マニュアル「HiRDB コマンドリファレンス」の「データベース構成変更ユティリティ（pdmod）」の「監査人の登録（セキュリティ監査機能）」の「機能」を参照してください。
- 一度登録した監査人を削除したり，変更したりできません。
- 監査人は DBA 権限を持てません。監査人が取得できるユーザ権限を次の表に示します。

表 24–12　監査人が取得できるユーザ権限

ユーザ権限の種類	権限の有無		備考
	主監査人	副監査人	
監査権限	○	○	監査人を登録したときに自動的に権限が与えられます。
CONNECT 権限	○	○	
スキーマ定義権限	○	×	
スキーマ操作権限	×	×	監査人はスキーマ操作権限を持てません。
DBA 権限	×	×	監査人は DBA 権限を持てません。
RD エリア利用権限	○	×	監査証跡表を格納する RD エリアの利用権限を DBA 権限保持者から与えてもらう必要があります。ほかの RD エリアについても同様です。
表のアクセス権限	○	○	監査証跡表のアクセス権限が与えられます。ほかの表のアクセス権限については，表の所有者から与えてもらう必要があります。

（凡例）

○：この権限を取得できます。

×：この権限を取得できません。

(2)　監査証跡表を格納する RD エリアの作成

監査証跡表を格納する RD エリアの容量を見積もり，RD エリアを作成します。

(a)　監査証跡表を格納する RD エリアの容量見積もり

監査証跡表を格納する RD エリアの容量を見積もるときは，次に示す点を考慮してください。RD エリアの容量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

- 監査証跡表の行長及び監査証跡表を格納する RD エリアのページ長

  監査証跡表を格納する RD エリアのページ長は，監査証跡表の行長を基に決定してください。監査証跡表の行長の見積もり式は，次のとおりです。

  監査証跡表の行長＝ 1296 ＋ pd_aud_sql_source_size の値＋ pd_aud_sql_data_size の値（単位：バイト）

  SQL 文及び SQL データを取得する場合，これらのバイナリデータを分岐して RD エリアに格納するかどうかを，監査証跡の運用方法に応じて決定してください。例えば，SQL 文や SQL データを監査証跡の検索対象にしない場合，これらを分岐して RD エリアに格納する設計とする方が，性能が良くなることがあります。

  なお，SQL 文及び SQL データの両方を分岐して RD エリアに格納する場合，監査証跡表の行長は 1320 バイトです。

  ページ長の決定については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

- 監査証跡表に記録されるレコード数

  はじめに，監査証跡を取得するイベントを決定してください。取得できるイベントについては「監査対象になるイベント」を，イベントごとに記録される監査証跡表の項目については「監査証跡のレコード項目（権限チェック時）」及び「監査証跡のレコード項目（イベント終了時）」を参照してください。

  次に，監査証跡を取得するイベントの発生頻度と，各イベントの記録レコード数※から，1 日に記録される監査証跡のレコード数を算出します。そして，次の見積もり式を基に，監査証跡表に格納する行数を算出してください。

  監査証跡表に格納する行数＝ 1 日に記録される監査証跡のレコード数×データを保存したい日数

注※　各イベントの記録レコード数については，「監査証跡のレコード項目（権限チェック時）」及び「監査証跡のレコード項目（イベント終了時）」の表にある，「記録レコード数」の列を参照してください。

(b)　RD エリアの作成

pdmod コマンドの **create rdarea 文**で，監査証跡表を格納する RD エリアを作成します。RD エリアを作成するときの注意事項を次に示します。

- 監査証跡表を格納する RD エリアは，監査証跡ファイル用とは別の HiRDB ファイルシステム領域に作成することを推奨します。

  監査証跡ファイル用の HiRDB ファイルシステム領域に，監査証跡表を格納する RD エリアを作成すると，HiRDB の稼働中に監査証跡ファイルの容量が不足することがあります。
- RD エリアの種類はユーザ用 RD エリアにしてください。
- RD エリアの利用権限を主監査人だけに与えてください。公用 RD エリアにしたり，ほかのユーザに利用権限を与えたりしないでください。RD エリアの利用権限は create rdarea 文の for user used by オペランドで指定します。
- 追加した RD エリアにグローバルバッファを割り当ててください。
- 既存の RD エリアに監査証跡表を格納できますが，公用 RD エリア又は主監査人以外のユーザに利用権限がある RD エリアには監査証跡表を格納できません。この場合は，RD エリアの利用権限を主監査人だけに変更してください。
- 監査証跡表を格納する RD エリアを再作成又は変更する場合は，主監査人が監査証跡表を削除した後に行ってください。

(3)　監査証跡表の作成

pdmod コマンドの **create audit table 文**で監査証跡表を作成します。監査証跡表を作成するときの注意事項を次に示します。

- 監査証跡表は一つだけ作成できます。
- 監査証跡表は削除及び再定義できます。監査証跡表を削除する場合は主監査人が DROP TABLE 文を実行してください。主監査人以外のユーザは監査証跡表を削除できません。削除した後に監査証跡表を再作成する場合は HiRDB 管理者が pdmod コマンドの create audit table 文を実行してください。

- 監査証跡表はインデクスが定義できます。ただし，データの一意性を保証する列がないため，UNIQUE 指定のインデクスは定義しないでください。監査証跡表の列構成については，表「監査証跡表の列構成」を参照してください。
- 監査証跡表は横分割できません。
- 監査証跡表の表定義は変更できません。

24.4.4　監査対象イベントの定義

実行者　監査人

システム管理者セキュリティイベント，及び監査人セキュリティイベントは必ず監査対象になりますが，次に示す監査対象イベントを監査対象にするかどうかは **CREATE AUDIT** で定義します。次に示す監査対象イベントのうち任意のイベントを監査対象にできます。

- セッションセキュリティイベント
- 権限管理イベント
- オブジェクト定義イベント
- オブジェクト操作イベント
- ユティリティ操作イベント

これらの監査対象イベントを監査対象から外す場合は **DROP AUDIT** を実行してください。CREATE AUDIT 及び DROP AUDIT については，マニュアル「HiRDB SQL リファレンス」を参照してください。

24.5　運用方法

ここでは，セキュリティ監査機能使用時に HiRDB 管理者が行う操作と監査人が行う操作，及び監査証跡表の自動データロード機能を適用した場合の運用について説明します。

24.5.1　HiRDB 管理者が行う操作

(1)　監査証跡の取得

HiRDB 管理者は次に示すどちらかの方法で監査証跡を取得してください。

- **pd_audit オペランドに Y を指定する**
  この場合，HiRDB の開始時から監査証跡を取得します。
- **pdaudbegin コマンドを実行する**
  この場合，コマンド実行時から監査証跡を取得します。

なお，監査証跡の取得を中止する場合は **pdaudend** コマンドを実行してください。このコマンドは HiRDB 管理者だけが使用できます。監査人は使用できません。

参考

- HiRDB を再開始する場合は前回稼働時の状態を引き継ぎます。監査証跡を取得していた場合は再開始後も監査証跡を取得します。監査証跡を取得していなかった場合は再開始後も監査証跡を取得しません。
- HiRDB を正常開始する場合は，前回稼働時の状態より pd_audit オペランドの指定を優先します。監査証跡を取得していた場合でも，pd_audit ＝ N を指定していると正常開始後に監査証跡を取得しません。監査証跡を取得していなかった場合でも，pd_audit ＝ Y を指定していると正常開始後に監査証跡を取得します。

(2)　監査証跡ファイルの削除

pdaudrm コマンドで監査証跡ファイルを削除できます。このコマンドは HiRDB 管理者だけが使用できます。監査人は使用できません。

(3)　監査証跡表を格納する RD エリアの操作

監査証跡表を格納する RD エリアの操作は HiRDB 管理者が行います。例えば，次に示す操作は HiRDB 管理者が行います。

- 監査証跡表を格納する RD エリアのバックアップ取得
- 監査証跡表を格納する RD エリアの回復

- 監査証跡表を格納する RD エリアの構成変更（RD エリアの拡張など）
- 監査証跡表を格納する RD エリアの利用権限の付与又は削除※

注※

監査証跡表を格納する RD エリアを変更する場合に行います。例えば，変更前の RD エリアの利用権限を削除し，変更後の RD エリアの利用権限を主監査人に付与します。

(4)　監査証跡表へのデータ登録（監査証跡表へのデータロード）

監査証跡表へのデータ登録は，HiRDB 管理者でコマンドを実行し，HiRDB へアクセスする際の認証には監査人の認可識別子とパスワードを指定してください。データロード待ちの監査証跡ファイルを入力情報にして，データベース作成ユティリティ（pdload コマンド）で監査証跡表にデータロードしてください。データロードの方法については，「監査証跡表へのデータ登録」を参照してください。

(5)　監査証跡ファイルの状態確認

pdls -d aud コマンドでの監査証跡ファイルの状態確認は，HiRDB 管理者でコマンドを実行し，HiRDB へアクセスする際の認証には監査人の認可識別子とパスワードを指定してください。次のことを確認できます。

- データロード待ちの監査証跡ファイルがあるか
- スワップ先にできる監査証跡ファイルがあるか

監査証跡ファイルの状態については，「監査証跡ファイルの運用」を参照してください。

(6)　監査証跡ファイル用の HiRDB ファイルシステム領域の操作

監査証跡表用の HiRDB ファイルシステム領域の操作は HiRDB 管理者が行います。例えば，次に示す操作は HiRDB 管理者が行います。

- HiRDB ファイルシステム領域の削除による監査証跡表の削除（OS のコマンドで削除）※
- pdfbkup コマンドによる HiRDB ファイルシステム領域のバックアップ取得
- pdfrstr コマンドによる HiRDB ファイルシステム領域の回復
- pdfmkfs コマンドによる HiRDB ファイルシステム領域の初期設定
- pdfrm コマンドによる監査証跡表を格納している HiRDB ファイルの削除

注※

HiRDB はこのイベントを監査証跡として出力しません。このイベントを監査するには OS の監査機能を使用してください。

(7)　監査証跡ファイルのスワップ

pdaudswap コマンドでの監査証跡ファイルのスワップは，HiRDB 管理者でコマンドを実行し，HiRDB へアクセスする際の認証には監査人の認可識別子とパスワードを指定してください。現用の監査証跡ファイ

ルのデータを監査証跡表にデータロードする場合，pdaudswap コマンドで監査証跡ファイルをスワップした後にデータロードを実行する必要があります。現用ファイルはデータロードできません。

注意事項

次に示すどちらかの条件を満たす場合は pdaudswap コマンドを実行できません。

- スワップ先にできる監査証跡ファイルがない
- 監査証跡ファイルが一つも生成されていない

24.5.2　監査人が行う操作

(1)　監査証跡表の利用

監査証跡表を利用して監査を行ってください。監査表の列構成については，「監査証跡表の列構成」を参照してください。

(2)　監査証跡表の操作

監査証跡表の操作は監査人が行います。例えば，次に示す操作は監査人が行います。

- **監査証跡表のアクセス権限の付与**
  監査証跡表及び監査証跡表を基表とするビュー表の SELECT 権限だけをほかのユーザに与えられます。SELECT 権限の削除もできます。INSERT，UPDATE，及び DELETE 権限はほかのユーザに与えられません。監査証跡表のアクセス権限の付与及び削除は主監査人が行ってください。主監査人以外のユーザは監査証跡表のアクセス権限を付与できません。
- **監査証跡表のインデクス定義**
  監査証跡表にインデクスを定義できます。監査表の列構成については，「監査証跡表の列構成」を参照してください。
  なお，監査証跡表を横分割したり，表の定義を変更したりすることはできません。
- **監査証跡表の再編成**
  監査証跡表の再編成は監査人が行ってください。監査人以外のユーザは監査証跡表を再編成できません。
- **監査証跡表の削除**
  DROP TABLE で監査証跡表を削除できます。監査証跡表の削除は主監査人が行ってください。主監査人以外のユーザは監査証跡表を削除できません。
  監査証跡表を再度作成する場合は，HiRDB 管理者が pdmod コマンドの create audit table 文を実行してください。

(3)　コマンド実行時の認証

次のコマンドを実行するときは，HiRDB 管理者と共同で行う必要があります。HiRDB 管理者がコマンドを実行すると，パスワード入力要求のメッセージが表示されます。このとき，監査人が認可識別子とパスワードを入力してください。

- pdaudatld コマンド
- pdaudput コマンド
- pdaudswap コマンド
- pdls -d aud コマンド
- データベース作成ユティリティ（pdload）※

注※　監査証跡を監査証跡表にデータロードをするときだけ該当します。

(4)　監査対象イベントの追加及び削除

CREATE AUDIT で監査対象イベントを追加できます。DROP AUDIT で不要な監査対象イベントを削除できます。

(5)　パスワードの変更

GRANT AUDIT で監査人が自分のパスワードを変更できます。ほかの監査人のパスワードは変更できません。

(6)　主監査人のスキーマ削除

監査証跡表がない場合に主監査人のスキーマを削除できます。主監査人及び DBA 権限保持者が主監査人のスキーマを削除できます。

24.5.3　監査証跡表の自動データロード機能を適用した場合の運用

(1)　監査証跡表の参照

監査証跡表の自動データロード機能を適用すると，pdload コマンドがデータロード中でも NOWAIT 検索で監査証跡表が参照できます※。監査証跡表を参照するときは，SQL の排他オプションに WITHOUT LOCK NOWAIT を指定してください。SQL の排他オプションを指定しないで監査証跡表を参照した場合，pdload コマンドの実行中に監査証跡表にアクセスしようとした SQL が排他解除待ちになることや，SQL で検索中の監査証跡表にデータロードしようとした pdload コマンドが排他解除待ちタイムアウトになることがあり，pdload コマンドが異常終了するおそれがあります。

自動データロード機能では，排他解除待ちタイムアウトのようなリトライ可能なエラーが発生した場合，再度データロードを行いますが，エラーが繰り返し発生した場合は機能を停止します。この場合，自動デー

タロード機能を手動で再開始する必要があります。エラーの内容に応じた対処方法については，表「自動データロード実行中の障害内容と対処方法」を参照してください。

注※

pdaudload コマンドでインデクス一括作成モード，又はローカルバッファを使用したデータロードを実行している場合は，NOWAIT 検索による監査証跡表の参照はできません。検索方法による監査証跡表の参照可否を，次の表に示します。

検索方法	データロード中でない場合	データロード中の場合			
		pdload コマンドが使用するバッファ		pdload コマンドのインデクス作成方法	
		グローバル	ローカル	同時作成	一括作成
NOWAIT 検索	○	○	×	○	×
上記以外の検索	○	×	×	×	×

（凡例）

○：監査証跡表を参照できます。

×：監査証跡表を参照できません（データロードが完了するまで，排他解除待ちとなります）。

(2)　自動データロード機能の開始及び停止のタイミング

監査証跡表の自動データロード機能を適用した場合の，機能の開始及び停止のタイミングを次に示します。

- **開始タイミング**
  - HiRDB を開始（pdstart コマンドを実行）した時
  - pdaudatld コマンド（-b オプション指定）を実行して機能を再開始した時
  - HiRDB を再開始した時

  自動データロード機能の開始時に，データロード待ち状態の監査証跡ファイルをデータロードします。
- **停止タイミング**
  - HiRDB を停止（pdstop コマンドを実行）した時
  - pdaudatld コマンド（-t オプション指定）を実行して機能を停止した時

  pdstop コマンド又は pdaudatld コマンドを受け付けた時点でデータロードの対象となる監査証跡ファイルがある場合は，それらをすべてデータロードした後に機能を停止します。コマンドを受け付けた後に監査証跡ファイルのスワップが発生した場合は，次回 pdstart コマンド又は pdaudatld コマンドを実行した時にデータロードを実行します。

次の場合，自動データロード機能は即時に停止します。データロードされなかった監査証跡ファイルは，次回 pdstart コマンド又は pdaudatld コマンドを実行した時にデータロードを実行します。

- 監査証跡表へのデータロード中にリトライできない障害が発生した時

- HiRDB が異常終了（pdload コマンドを実行しているユニットがダウン※）した時
- HiRDB を強制終了（pdstop -f コマンドを実行）した時

pdaudatld コマンドは，監査証跡表の自動データロード機能の再開始と停止を制御するコマンドです。詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

注※

システムマネジャがあるユニットがダウンした場合は，HiRDB の再開始時に自動データロード機能を開始します。なお，pdload を実行していないユニットがダウンしても，自動データロード機能は停止しません。

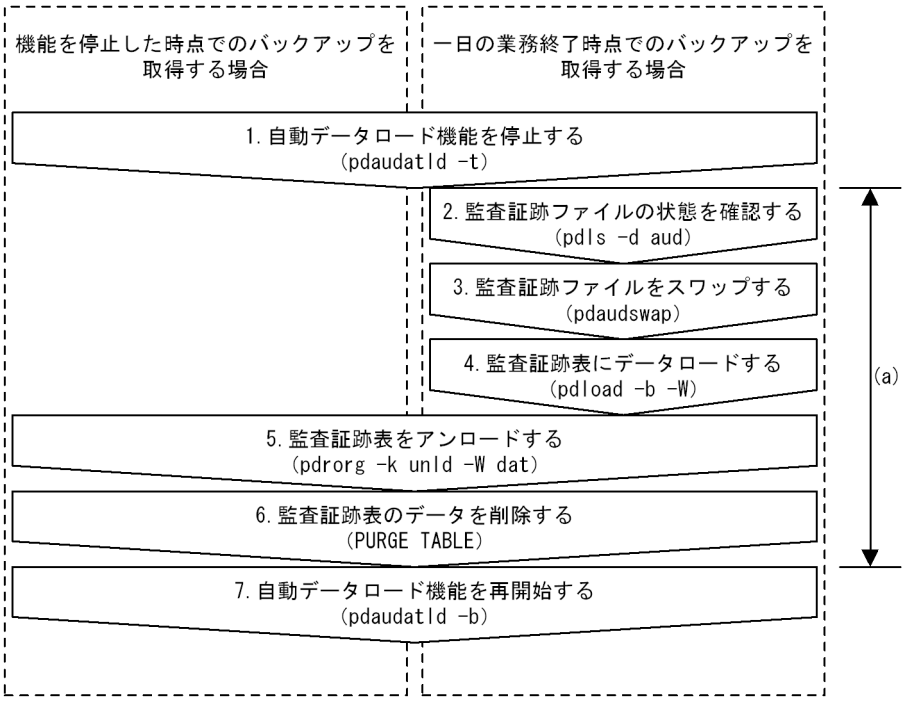
障害発生時の運用については，「自動データロード機能適用中に障害が発生したときの対処方法」を参照してください。

(3)　自動データロード機能の一時停止を利用した運用例

監査人は，pdaudatld コマンドで自動データロード機能を任意の時点で一時停止できます。ここでは運用例として，自動データロード機能を一時停止し，機能を停止した時点又は一日の業務終了時点での監査証跡表の内容をアンロードログファイルに退避し，バックアップを取得する方法について説明します。

- 機能を停止した時点でのバックアップを取得する場合は，手順 1，5，6，7 の操作を行ってください。この方法は，一定時間ごとの監査証跡表の状態をバックアップする運用を前提としているため，現用の監査証跡ファイルの内容はバックアップの対象にはなりません。
- 一日の業務終了時点でのバックアップを取得する場合は，手順 1～7 の操作を行ってください。この方法は，一日の業務で記録されたすべての監査証跡をバックアップする運用を前提としているため，現用の監査証跡ファイルの内容も監査証跡表にデータロードします。

〈手順〉



〈説明〉

1. pdaudatld -t コマンドを実行して，自動データロード機能を停止します。これによって，次に機能を再開始するまでの間（図中の(a)で示す期間）は，監査証跡表へのデータロードは発生しません。なお，データロード中に pdaudatld -t コマンドを実行した場合は，データロードがすべて完了してから機能を停止します。

   機能が正常に停止すると，pdaudatld -t コマンド実行までにデータロード待ち状態となった監査証跡ファイルの内容は，すべて監査証跡表にデータロードされた状態になります。
2. pdls -d aud コマンドを実行して，1.で機能を停止した後にデータロード待ち状態になった監査証跡ファイルや，現用の監査証跡ファイルを確認します。
3. pdaudswap コマンドを実行して現用の監査証跡ファイルをスワップし，データロード待ち状態にします。
4. pdload コマンドを実行して，データロード待ち状態となっている監査証跡ファイルを監査証跡表にデータロードします。
5. pdrorg コマンドを実行して，この時点の監査証跡表の内容をアンロードデータファイルに出力します。
6. 監査証跡表の内容は，アンロードデータファイルに出力されたため不要になります。容量不足による自動データロード機能の停止を防ぐために，監査証跡表のデータを削除します。
7. pdaudatld -b コマンドを実行して，自動データロード機能を再開始します。

(4)　自動データロード機能を適用した場合の HiRDB の停止処理

監査証跡表の自動データロード機能を適用している場合，HiRDB の停止処理の動作が変わることがあります。停止処理ごとの HiRDB の動作内容と，対処方法を次の表に示します。

表 24–13　監査証跡表の自動データロード機能を適用した場合の HiRDB の停止処理動作

<table>
<tr><th colspan="2" rowspan="2">停止処理の種類</th><th rowspan="2">自動データロードの実行状態</th><th colspan="2">HiRDB の動作</th><th rowspan="2">対処方法</th></tr>
<tr><th>停止処理</th><th>自動データロード機能の動作状態※1</th></tr>
<tr><td colspan="2" rowspan="2">システム正常停止（pdstop 実行）
又は
システム計画停止（pdstop -P 実行）</td><td>未実行</td><td>自動データロード機能を適用していない場合と動作は同じです。</td><td>「ENABLE」を保ちます。</td><td rowspan="2">－</td></tr>
<tr><td>実行中</td><td>• KFPS05719-W メッセージを表示します。
• すべてのデータロードを完了した後，正常停止処理を開始します（監査証跡表へのデータロードが完了すると，KFPS05720-I メッセージを表示します）。</td><td>• 「DISABLE(WAIT)」に変わります。
• 監査証跡ファイルのスワップが発生した場合は，pdstop コマンドの実行前なら HiRDB が停止する前に，pdstop の実行後なら次回 HiRDB を起動した時に再度データロードを実行します。</td></tr>
<tr><td colspan="2" rowspan="2">システム強制停止（pdstop -f 実行）</td><td>未実行</td><td rowspan="2">自動データロード機能を適用していない場合と動作は同じです。</td><td>「ENABLE」を保ちます。</td><td rowspan="2">－</td></tr>
<tr><td>実行中</td><td>• 「ENABLE」を保ちます。
• 実行中のデータロードは，システム強制停止と同時に異常終了します。
• データロード中だった監査証跡ファイルは，次回 HiRDB を起動した時に再度データロードを実行します。</td></tr>
<tr><td rowspan="2">非 MGR ユニット正常停止（pdstop -u 又は-x）※4</td><td rowspan="2">監査証跡表のあるユニット，又は自動データロード中の監査証跡ファイルのあるユニット</td><td>未実行</td><td>自動データロード機能を適用していない場合と動作は同じです。</td><td rowspan="2">「ENABLE」を保ちます。</td><td>－</td></tr>
<tr><td>実行中</td><td>ユニットは正常停止しません。KFPS05070-E 又は KFPS05234-E メッセージを表示し，pdstop コマンドはエラー終了します。</td><td>該当するユニットを正常停止する場合は，pdaudatld -t コマンドで自動データロード機能を停止させた後，pdstop -u/-x コマンドを実行してください。</td></tr>
</table>

停止処理の種類		自動データロードの実行状態	HiRDB の動作		対処方法
			停止処理	自動データロード機能の動作状態※1	
	上記以外のユニット	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中			
非 MGR ユニット強制停止（pdstop -z 又は-z -q）異常終了 ※2※4	監査証跡表のあるユニット，又は自動データロード中の監査証跡ファイルのあるユニット	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中		• 「DISABLE(WAIT)」に変わります。 • KFPS05713-E メッセージを表示します。 • 実行中のデータロードは，ユニット強制停止と同時にエラー終了，又は異常終了します。 • データロード中だった監査証跡ファイルは，次回自動データロード機能を再開始した後に再度データロードを実行します。	自動データロード機能を再開始する場合は，pdaudatld -b コマンドを実行してください。
	上記以外のユニット	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中			
MGR ユニット強制停止（pdstop -z 又は-z -q）異常終了 ※2※4		未実行	自動データロード機能を適用していない場合と動作は同じです。	MGR ユニットを再開始した後も，動作状態は「ENABLE」を保ちます。	－
		実行中		• MGR ユニットを再開始した後も，動作状態は「ENABLE」を保ちます。 • 実行中のデータロードは，ユニット強制停止と同時に異常終了します。 • データロード中だった監査証跡ファイルは，MGR ユニットを再開始した後に再度データロードを実行します。	
サーバ正常停止（pdstop	監査証跡表のあるサーバ	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－

停止処理の種類		自動データロードの実行状態	HiRDB の動作		対処方法
			停止処理	自動データロード機能の動作状態※1	
-s 又は-u -s）※4		実行中	サーバは正常停止しません。KFPS05071-E 又は KFPS05235-E メッセージを表示し，pdstop コマンドはエラー終了します。		該当するサーバを正常停止する場合は，pdaudatld -t コマンドで自動データロード機能を停止させた後，pdstop -s/-u -s コマンドを実行してください。
	上記以外のサーバ	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中			
サーバ強制停止（pdstop -s -f，-s -z 又は-u -s -z）※3※4	監査証跡表のあるサーバ	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中		• 「DISABLE」に変わります。 • KFPS05713-E メッセージを表示します。 • 実行中のデータロードは，サーバ強制停止と同時にエラー終了，又は異常終了します。 • データロード中だった監査証跡ファイルは，次回自動データロード機能を再開始した後に再度データロードを実行します。	自動データロード機能を再開始する場合は，pdaudatld -b コマンドを実行してください。
	上記以外のサーバ	未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中			
HiRDB サービス停止指示		未実行	自動データロード機能を適用していない場合と動作は同じです。	「ENABLE」を保ちます。	－
		実行中	• イベントログに，KFPS05719-W メッセージを表示します。 • すべてのデータロードを完了した後，HiRDB サービス停止処理を開始します（監査証跡表へのデータロードが完了すると，イベントロ	• 「ENABLE」を保ちます。 • 監査証跡ファイルのスワップが発生した場合は，サービス停止指示以前であれば HiRDB が停止する前に，以後であれば次	

停止処理の種類	自動データロードの実行状態	HiRDB の動作		対処方法
		停止処理	自動データロード機能の動作状態※1	
		グに KFPS05720-I メッセージを表示します)。	回 HiRDB を起動した時に再度データロードを実行します。	

(凡例)

–:特にありません。

MGR ユニット:システムマネジャがあるユニット

非 MGR ユニット:システムマネジャがないユニット

注※1

pdaudatld -i コマンドで表示される,次の状態を示します。

ENABLE:有効

DISABLE(WAIT):停止処理中

DISABLE:停止

注※2

系切り替えが契機のユニット強制停止を含みます。

注※3

影響分散スタンバイレス型系切り替えによるサーバ強制停止を含みます。

注※4

HiRDB/パラレルサーバの場合だけ該当します。

(5) データロード中に自動データロード機能を停止した場合の HiRDB の動作

監査証跡表の自動データロード機能を適用している場合,pdaudatld -t コマンドで機能を停止できますが,自動データロードの実行状態によって,機能を停止するタイミングが変わります。自動データロード機能を停止した場合の HiRDB の動作を,次の表に示します。

表 24–14 自動データロード機能を停止した場合の HiRDB の動作

自動データロードの実行状態	HiRDB の動作	
	pdaudatld -t コマンド実行時の動作	自動データロードの動作
未実行	• 自動データロード機能をすぐに停止し,KFPS05712-I メッセージを表示します。	–
実行中	• KFPS05719-W メッセージを表示し,監査証跡ファイルのデータロードが完了するまで,自動データロード機能の停止を待ち合わせます(データロードが完了すると,KFPS05720-I メッセージを表示します)。	• pdaudatld -t コマンド実行までにデータロード待ち状態となった監査証跡ファイルのデータロードを実行します。

自動データロードの実行状態	HiRDB の動作	
	pdaudatld -t コマンド実行時の動作	自動データロードの動作
	• 自動データロード機能を停止し，KFPS05712-I メッセージを表示します。	• pdaudatld -t コマンド実行後にデータロード待ち状態となった監査証跡ファイルは，次回 pdstart 又は pdaudatld -b コマンドを実行した時にデータロードを実行します。

（凡例）

－：特にありません。

(6) データロード完了を待ち合わせている状態で，データロードがエラー終了した場合の HiRDB の動作

(4)及び(5)で，ほかのコマンド（pdstop 及び pdaudatld -t）が監査証跡表へのデータロード完了を待ち合わせている場合があります。データロードが完了するとコマンドの実行を再開しますが，データロードがエラー終了した場合は，データロードを待ち合わせないでコマンドの実行を再開します。このときの HiRDB の動作を次の表に示します。

表 24–15 監査証跡表へのデータロード完了を待ち合わせている状態で，データロードが終了した場合の HiRDB の動作

監査証跡表へのデータロード完了を待ち合わせているコマンド	データロードの終了状態	
	正常終了した場合	エラー終了した場合
pdstop	• データロードが完了すると，KFPS05720-I メッセージを表示します。 • HiRDB の停止処理を実行します（コマンドの終了コードは 0 です）。	• エラーが発生すると，KFPS05721-W メッセージを表示します。 • データロード完了を待ち合わせないで HiRDB の停止処理を実行します（コマンドの終了コードは 0 です）。
pdaudatld -t	• データロードが完了すると，KFPS05720-I メッセージを表示します。 • 監査証跡表の自動データロード機能を停止します（コマンドの終了コードは 0 です）。	• エラーが発生すると，KFPS05721-W メッセージを表示します。 • データロード完了を待ち合わせないで HiRDB の停止処理を実行します（コマンドの終了コードは 4 です）。

(7) 注意事項

(a) 監査証跡表の容量不足に注意してください

監査証跡表の自動データロード機能を適用することで，監査証跡ファイルの管理は不要となります（入出力エラーが発生した場合を除きます）が，監査証跡表を格納する RD エリアの容量が不足しないように注意する必要があります。

監査証跡ファイルに監査証跡が記録される限り，データロードは自動的に続くため，監査証跡表を格納する RD エリアの容量が不足するとデータロードができなくなり，自動データロード機能が停止します。

自動データロード機能が停止した場合は，不要なデータを削除，又は RD エリアを拡張した後に，自動データロード機能を再開始する必要があります。対処方法については，表「自動データロード実行中の障害内容と対処方法」を参照してください。

(b)　pd_lck_wait_timeout オペランドに 0 を指定している場合

監査証跡ファイルをデータロードしている途中で別の監査証跡ファイルがスワップした場合，先行の監査証跡ファイルの処理が完了するまでは排他待ち状態となり，データロードできません。

先行するデータロード処理に障害が発生した場合など，排他待ち状態が長時間に及ぶと，データロード待ちの監査証跡ファイルが増えるおそれがあります。pd_lck_wait_timeout オペランドの値を 0 としている場合は，排他待ち状態が解除されるまで待ち続けるため，pdaudatld コマンドで監査証跡ファイルの状態を随時監視するようにしてください。

(c)　自動データロードの実行ユーザについて

HiRDB が監査証跡ファイルをデータロードする場合，pdload コマンドでは実行ユーザの権限チェックをしないため，権限チェック時の監査記録は出力しません。また，イベント終了時の監査記録に出力されるユーザ識別子の長さは 0 バイトです。

24.6　監査証跡ファイルの運用

ここでは，監査証跡ファイルが作成されるときの仕組みについて説明します。説明する項目は次のとおりです。

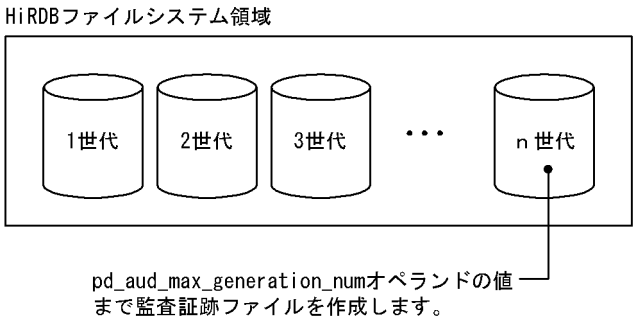
- 監査証跡ファイルの作成
- 監査証跡ファイルの状態
- 監査証跡ファイルのスワップ
- 監査証跡ファイルの容量変更

24.6.1　監査証跡ファイルの作成

(1)　監査証跡ファイルの作成規則

監査証跡ファイル用の HiRDB ファイルシステム領域内に HiRDB が自動的に監査証跡ファイルを作成します。最初に作成した監査証跡ファイルを 1 世代目として，順に 2 世代目，3 世代目，...というように作成します。監査証跡ファイルの作成規則を次の図に示します。

図 24–5　監査証跡ファイルの作成規則



HiRDB ファイルシステム領域内に作成される監査証跡ファイル数の上限は pd_aud_max_generation_num オペランドの値になります。ただし，pdfmkfs コマンドの-l オプションの値が pd_aud_max_generation_num オペランドの値より小さい場合は，-l オプションの値が上限になります。また，HiRDB ファイルシステム領域の領域不足が発生した場合は，pd_aud_max_generation_num オペランドの値まで監査証跡ファイルを作成できません。

(2)　監査証跡ファイルの名称

監査証跡ファイルの名称は次に示す規則で決まります。

pdaud ユニット識別子 001～200.aud

（例）ユニット識別子が UNT1 で，pd_aud_max_generation_num ＝ 100 の場合

１世代目のファイル名：pdaudUNT1001.aud

２世代目のファイル名：pdaudUNT1002.aud

：　　：

99 世代目のファイル名：pdaudUNT1099.aud

100 世代目のファイル名：pdaudUNT1100.aud

監査証跡ファイル用の HiRDB ファイルシステム領域内に監査証跡ファイル以外のファイルを作成する場合は，上記の名称規則に該当するファイルを作成しないでください。

(3)　監査証跡ファイルへの出力（同期出力）

監査証跡の出力方式は pd_aud_async_buff_size オペランドの指定値によって同期出力又は非同期出力に分かれます。ここでは，同期出力の場合について説明します。

監査証跡ファイルが一つもない場合は，最初の監査証跡の取得契機時に監査証跡ファイルを作成して監査証跡を出力します。監査証跡ファイルが一つ以上ある場合は，監査証跡の取得契機ごとに監査証跡を現用ファイルに出力します。

(4)　監査証跡ファイルへの出力（非同期出力）

監査証跡の出力方式は pd_aud_async_buff_size オペランドの指定値によって同期出力又は非同期出力に分かれます。ここでは，非同期出力の場合について説明します。

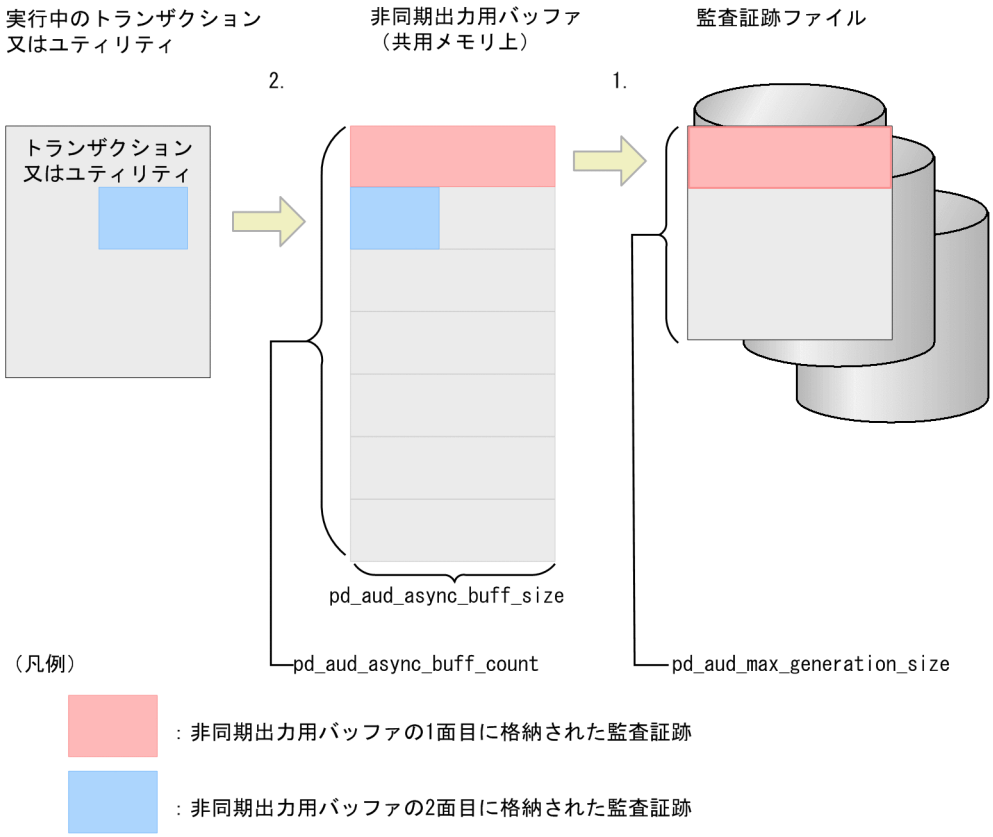
(a)　非同期出力の仕組み

監査証跡を非同期出力する場合の仕組みを説明します。

監査証跡の出力契機が発生すると，監査証跡を非同期出力用バッファに格納し，非同期出力用バッファが満杯になると，監査証跡ファイルへ出力します。このため，非同期出力では，HiRDB のシステム構築時に，非同期出力用バッファを用意しておく必要があります。

なお，非同期出力用バッファは，共用メモリ上に確保します。非同期出力の仕組みを次の図に示します。

図 24–6　非同期出力の仕組み



[説明]

1. 1 面目の非同期出力用バッファが満杯になると，監査証跡を監査証跡ファイルに出力し，格納先を 2 面目の非同期出力用バッファに切り替えます。
2. 以降の監査証跡は，2 面目の非同期出力用バッファに格納します。2 面目の非同期出力用バッファが満杯になった場合は，1.，2.を繰り返します。

注意事項

非同期出力用バッファの監査証跡を監査証跡ファイルに出力している間は，順序性を維持するため，別の非同期出力用バッファの監査証跡を監査証跡ファイルに出力できません。このため，監査証跡ファイルへの出力に時間を要する場合，複数面の非同期出力用バッファが満杯になることがあります。

その影響で，システムで稼働させるトランザクション，及びユティリティの実行間隔や同時実行本数によっては，非同期出力用バッファが不足することがあります。これを非同期出力用バッファの枯渇といいます。

非同期出力用バッファの枯渇は，HiRDB の性能劣化の原因になります。

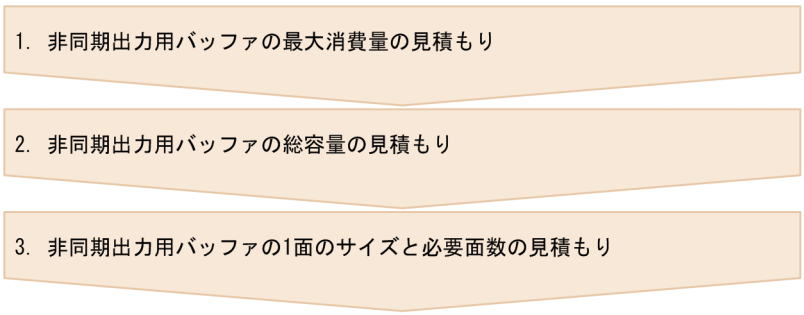
図の例では，非同期出力用バッファは 7 面分設定されています。もし，1 面目の非同期出力用バッファの内容を監査証跡ファイルに出力している間に，7 面すべての非同期出力用バッファが満杯になると，新しく発生した監査証跡の格納ができません。このとき，トランザクション又はユティリティは，非同期出力用バッファに空きができるまで，監査証跡の格納を待ち合わせるため，性能が劣化します。

HiRDB の性能劣化を防ぐため，監査証跡の出力状況を考慮して，非同期出力用バッファの大きさを見直すことをお勧めします。

(b)　非同期出力用バッファの大きさを見直す手順

HiRDB の性能劣化を防ぐため，次の図に示す手順で，非同期出力用バッファの大きさを見直してください。

図 24–7　非同期出力用バッファの大きさの見直し手順



1.　非同期出力用バッファの最大消費量の見積もり

まず，非同期出力用バッファの最大消費量を見積もります。

非同期出力用バッファの最大消費量とは，非同期出力用バッファ 1 面分の監査証跡を監査証跡ファイルに出力している間に格納された監査証跡によって使用される，非同期出力用バッファの最大値です。監査証跡ファイルの世代スワップ中に出力される監査証跡の件数から求められます。

非同期出力用バッファの最大消費量を算出する式を次に示します。

```
非同期出力用バッファの最大消費量（単位：バイト）＝
（監査証跡ファイルの世代スワップに掛かる時間…（i）
　　×1秒間に格納する監査証跡記録件数の最大値…（ii）
　　×監査証跡1件当たりのレコードサイズ…（iii））
　　＋非同期出力用バッファ1面のサイズ…（iv）
```

（i）監査証跡ファイルの世代スワップに掛かる時間

監査証跡ファイルの世代スワップに掛かる時間は，HiRDB がメッセージを出力した時刻から導きます。

次の二つのメッセージが出力された時刻の差分が，監査証跡ファイルの世代スワップに掛かった時間です。

- KFPS05703-I メッセージ（監査証跡ファイルの世代を解放したことを示す）
- KFPS05702-I メッセージ（監査証跡ファイルの世代を割り当てたことを示す）

なお，差分が 0 秒の場合は 1 秒とみなして計算してください。

（ii）1 秒間に格納する監査証跡記録件数の最大値

1 秒間に格納する監査証跡記録件数の最大値は，発行する SQL とアクセスする表数から求めます。

概算値は，次の計算式で求めます。

```
1秒間に格納する監査証跡記録件数の最大値＝1秒間に発行するSQL件数の最大数×3×アクセスする表数
```

監査証跡記録件数を正確に求める場合は，「監査証跡のレコード項目（権限チェック時）」及び「監査証跡のレコード項目（イベント終了時）」を参照してください。

（iii）監査証跡 1 件当たりのレコードサイズ

マニュアル「HiRDB システム導入・設計ガイド」の「監査証跡ファイルの容量の見積もり」の「監査証跡のレコードサイズ」を参照して見積もってください。

（iv）非同期出力用バッファ 1 面のサイズ

pd_aud_async_buff_size オペランドの指定値を使います。デフォルト値は 401408 です。

2. 非同期出力用バッファの総容量の見積もり

非同期出力用バッファの総容量は，次の計算式で求めます。

```
非同期出力用バッファの総容量＝
pd_aud_async_buff_sizeオペランドの指定値×pd_aud_async_buff_countオペランドの指定値
```

非同期出力用バッファの総容量が，非同期出力用バッファの最大消費量の 2 倍に満たない場合，非同期出力用バッファの最大消費量の 2 倍にすることを推奨します。推奨する理由は次のとおりです。

監査証跡ファイルへの出力に時間が掛かった場合，複数面の非同期出力用バッファが満杯になることがあります。その際，トランザクション量の急な増加が発生すると，非同期出力用バッファが枯渇するおそれがあります。そうした場合にも枯渇が発生しないよう，非同期出力用バッファの最大消費量の 2 倍の値にすることを推奨します。

3. 非同期出力用バッファの 1 面のサイズと必要面数の見積もり

次の表に従って，システム定義の pd_aud_async_buff_size オペランドと pd_aud_async_buff_count オペランドの値を見積もります。

1 面当たりのサイズと必要面数の推奨値を次の表に示します。

表 24–16　非同期出力用バッファ 1 面当たりのサイズと必要面数の推奨値

判定条件	非同期出力用バッファ	
	1 面当たりのサイズ（pd_aud_async_buff_size オペランドの指定値）	必要面数（pd_aud_async_buff_count オペランドの指定値）
推奨値（非同期出力用バッファの最大消費量×2）≦非同期出力用バッファの総容量	変更は不要です。	変更は不要です。
推奨値（非同期出力用バッファの最大消費量×2）＞非同期出力用バッファの総容量	最大値の 6553600 に変更します。	次の計算式で算出した値に変更します。 max（現在の pd_aud_async_buff_count オペランドの指定値，↑非同期出力用バッファの最大消費量×2÷6553600↑）

注意事項

非同期出力用バッファの総容量を見直しても，監査証跡ファイルの世代スワップが連続して発生した場合は，非同期出力用バッファから監査証跡ファイルへの出力が進まなくなり，非同期出力用バッファの枯渇につながるおそれがあります。

非同期出力用バッファの枯渇による HiRDB の性能劣化を防ぐには，非同期出力用バッファの大きさだけではなく，監査証跡ファイル 1 世代の最大容量（pd_aud_max_generation_size オペランド）の値が，次の大小関係を満たすように見直すことをお勧めします。

（非同期出力用バッファの最大消費量×2÷1024÷1024）＜ pd_aud_max_generation_size オペランドの指定値

（単位：メガバイト）

(c)　非同期出力用バッファフラッシュ契機

非同期出力を選択した場合は監査証跡の出力契機が発生すると，監査証跡を一時的に非同期出力用のバッファに格納します。非同期出力用のバッファに格納した監査証跡は次に示す契機に監査証跡ファイルに出力されます。これを非同期出力用バッファフラッシュ契機といいます。

- 非同期出力用バッファが満杯になった場合
- pdaudswap コマンドを実行した場合
- pdaudend コマンドを実行した場合
- pdstop コマンドで HiRDB（ユニット）を正常終了又は計画停止した場合

なお，非同期出力用バッファフラッシュ契機が頻繁に発生すると，非同期出力用バッファのすべての面がフラッシュ待ちとなることがあります。このときの対処方法については，「非同期出力用バッファのすべての面がフラッシュ待ちになった場合」を参照してください。

すべての面がフラッシュ待ちになってから 180 秒経過すると，HiRDB は pd_aud_no_standby_file_opr オペランドの指定値に従って処理を行います。

forcewrite（省略値）を指定した場合

最初にフラッシュ待ちになった非同期出力用バッファの監査証跡を上書きして，監査証跡の出力を続行します。上書きされた非同期出力用バッファの監査証跡は破棄されます。

down を指定した場合

KFPS05722-E メッセージを出力し，HiRDB（HiRDB/パラレルサーバの場合はユニット）を強制終了します。非同期出力用バッファにあった監査証跡，及び非同期出力用バッファに出力しようとしていた監査証跡はともに破棄されます。

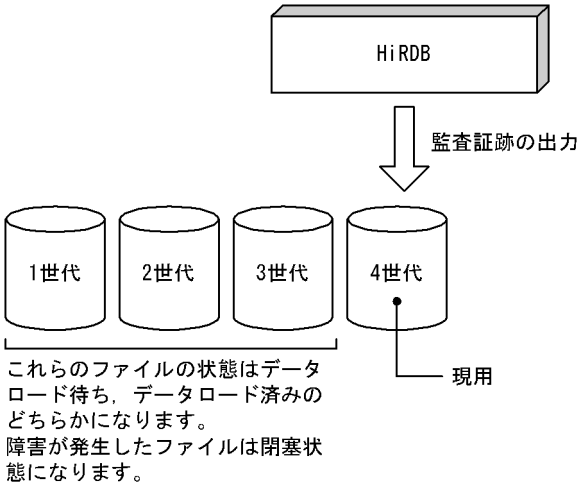
24.6.2　監査証跡ファイルの状態

HiRDB は監査証跡ファイルを幾つかの状態に分けて管理しています。監査証跡ファイルの状態を次の表及び次の図に示します。監査証跡ファイルの状態は **pdls -d aud** コマンドで確認できます。

表 24–17　監査証跡ファイルの状態

ファイルの状態	説明
現用	監査証跡の出力対象になっているファイルです。この状態のファイルは常に一つです。なお，現用以外のファイルを待機といいます。
データロード済み	監査証跡表へのデータロードが完了しているファイルです。監査証跡ファイルのスワップ発生時にスワップ先にできるファイルです。
データロード待ち	監査証跡表へのデータロードが完了していないファイルです。監査証跡ファイルのスワップ発生時にスワップ先にできないファイルです。ただし，pd_aud_no_standby_file_opr オペランドの値によっては，この状態のファイルをスワップ先にすることもできます。
閉塞	監査証跡の入出力又は監査証跡ファイルヘッダの入出力時に障害が発生して，使用できない監査証跡ファイルです。監査証跡ファイルのスワップ発生時にスワップ先にできないファイルです。なお，閉塞していないファイルを正常なファイルといいます。

図 24–8　監査証跡ファイルの状態



24.6.3　監査証跡ファイルのスワップ

監査証跡の出力先ファイルを変更すること（現用ファイルを変更すること）を**監査証跡ファイルのスワップ**といいます。

(1)　監査証跡ファイルがスワップするとき

監査証跡ファイルがスワップするときの条件を次の表に示します。

表 24–18　監査証跡ファイルがスワップするときの条件

スワップするときの条件	説明
ファイル容量の満杯時	出力された監査証跡が監査証跡ファイルの容量一杯になるとスワップします。監査証跡ファイルの容量は pd_aud_max_generation_size オペランドで指定しています。

スワップするときの条件	説明
現用ファイルの障害時	入出力処理などで現用ファイルに障害が発生するとスワップします。
pdaudswap コマンドの実行時	pdaudswap コマンドを実行するとスワップします。
HiRDB の再開始時	HiRDB（HiRDB/パラレルサーバの場合はユニット）が再開始するとスワップします。正常開始時及び計画停止後の再開始時はスワップしないで，HiRDB 終了時の現用ファイルを継続して使用します。

なお，次に示す場合は自動的にほかのファイルにスワップします。

- スワップ先の監査証跡ファイルの生成に失敗した場合
- 管理情報の更新に失敗した場合

(2)　監査証跡ファイルのスワップ先

監査証跡ファイル数が上限（pd_aud_max_generation_num の値）に達しているかどうかによって処理が多少異なります。

- 監査証跡ファイル数が上限（pd_aud_max_generation_num の値）に達していない場合は，監査証跡ファイルを新規に作成してそのファイルをスワップ先にします。ファイルの世代番号には，空いている世代番号のうち一番小さい番号を割り当てます。
- 監査証跡ファイル数が上限（pd_aud_max_generation_num の値）に達している場合は，データロード済みのファイル（閉塞状態のファイルを除く）の中でファイルの最終更新日時が一番古いファイルをスワップ先にします。

(3)　スワップ先にできるファイルがない場合の HiRDB の動作及び対処方法

スワップ先にできるファイルがない場合，HiRDB は pd_aud_no_standby_file_opr オペランドの指定値に従って動作します。

forcewrite（省略値）を指定した場合

スワップ先にできるファイルがない場合の HiRDB の動作，及び対処方法を次に示します。

スワップするときの条件	HiRDB の動作	対処方法
• ファイル容量の満杯時 • 現用ファイルの障害時 • HiRDB の再開始時	KFPS05706-W メッセージを出力し，データロード待ち（閉塞状態のファイルを除く）の監査証跡ファイルに強制的にスワップします。このとき，最終更新日時が一番古いデータロード待ちの監査証跡ファイルをスワップ先にします。	－
• pdaudswap コマンドの実行時	KFPS05752-E メッセージ（理由コード NO_STANDBY_FILE）を出力し，コマンドエラーとなります。この場合，HiRDB（HiRDB/パラレルサーバの場合はユニット）は稼働したままです。	監査人は，監査証跡ファイルのデータを監査証跡表へデータロードして，スワップ先の監査証跡ファイルをデータロード済みの状態にしてください。

（凡例）－：該当しません。

down を指定した場合

スワップ先にできるファイルが残り一つ以下になった場合の HiRDB の動作，及び対処方法を次に示します。

スワップするときの条件	HiRDB の動作	対処方法
• ファイル容量の満杯時 • 現用ファイルの障害時	KFPS05705-E メッセージを出力し，HiRDB（HiRDB/パラレルサーバの場合はユニット）を強制終了します。	「スワップ先にできる監査証跡ファイルがないため HiRDB が強制終了した場合」を参照して対処してください。
• pdaudswap コマンドの実行時	KFPS05752-E メッセージ（理由コード UNABLE_SWAP）を出力し，コマンドエラーとなります。この場合，HiRDB（HiRDB/パラレルサーバの場合はユニット）は稼働したままです。	監査人は，監査証跡ファイルのデータを監査証跡表へデータロードして，スワップ先の監査証跡ファイルをデータロード済みの状態にしてください。
• HiRDB の再開始時	**スワップ先にできるファイルが一つの場合** HiRDB（HiRDB/パラレルサーバの場合はユニット）を再開始します。	
	スワップ先にできるファイルがない場合 KFPS05724-E メッセージ，及び KFPS05725-W メッセージを出力して，セキュリティ監査機能を停止した上で HiRDB（HiRDB/パラレルサーバの場合はユニット）を再開始します。	監査人は，監査証跡ファイルのデータを監査証跡表へデータロードしてください。その後，HiRDB 管理者は，pdaudbegin コマンドでセキュリティ監査機能を再開してください。

なお，pd_aud_file_wrn_pnt オペランドを指定すると，スワップ先にできない監査証跡ファイルの数が警告値に達したときに警告メッセージ（KFPS05123-W メッセージ）を出力できます。警告メッセージが出力された場合は，早めに監査証跡ファイルのデータを監査証跡表へデータロードしてください。

24.6.4　監査証跡ファイルの容量変更

監査証跡ファイルの 1 世代の容量は，監査証跡ファイルの新規作成時に pd_aud_max_generation_size オペランドの指定値で決まります。

そのため，監査証跡ファイルの容量を変更する場合は，次のどちらかの手順で監査証跡ファイルを再作成してください。なお，閉塞状態の監査証跡ファイルがある場合は「監査証跡ファイルに障害が発生した場合」を参照して対処してください。

〈手順 1〉pdfmkfs コマンドで HiRDB ファイルシステム領域を初期化する場合

1. 監査証跡表へのデータロードを実行します。※
2. HiRDB を正常停止します。
3. pd_aud_max_generation_size オペランドの指定値を変更します。

4. pdfmkfs コマンドで監査証跡ファイルを格納している HiRDB ファイルシステム領域を初期化してください。これによって HiRDB ファイルシステム領域内の監査証跡ファイルをすべて削除します。

   このとき，容量の変更に合わせて HiRDB ファイルシステム領域サイズの指定値を変更してください。

〈手順 2〉pdfrm コマンドで監査証跡ファイルを削除する場合

HiRDB ファイルシステム領域に監査証跡ファイル以外を含んでいる場合など，HiRDB ファイルシステム領域を再作成したくない場合はこの方法を実施してください。

1. 監査証跡表へのデータロードを実行します。※
2. HiRDB を正常停止します。
3. pd_aud_max_generation_size オペランドの指定値を変更します。

   このとき HiRDB ファイルシステム領域の容量を超えないようにしてください。
4. pdfrm コマンドで監査証跡ファイルを削除してください。

注※

監査証跡表の自動データロード機能を適用している場合は，HiRDB の正常停止時に監査証跡表へのデータロードが実行されるため省略できます。

24.7　監査証跡表へのデータ登録

実行者　監査人

監査証跡表にデータを登録するには HiRDB が提供している**監査証跡データ登録 UOC** を使用します。この UOC はデータベース作成ユティリティ（pdload コマンド）の延長で動作し，監査証跡ファイルに出力されたデータを監査証跡表へデータロードする機能を持っています。なお，監査証跡表へのデータロードは監査人だけが実行できます。

ポイント

pdload コマンドを実行する場合は，環境変数 PDUSER で監査人の認可識別子とパスワードを指定するか，又は pdload コマンドの-u オプションで監査人の認可識別子を指定し，pdload コマンド実行時にパスワードを入力する必要があります。

24.7.1　例題 1（特定の監査証跡ファイルをデータロードする場合）

データロード待ちの監査証跡ファイル（ファイル名：pdaudUNT1001.aud 及び pdaudUNT1002.aud）を監査証跡表（表名：SQL_AUDIT_TRAIL）にデータロードします。このときの pdload コマンドの指定例は次のとおりです。

pdload コマンドの指定例

```
pdload -b -W SQL_AUDIT_TRAIL C:¥pdload¥control_file
```

〔説明〕

- -b 及び-W オプションは必ず指定してください。
- 表名には監査証跡表を指定します。
- 制御情報ファイルの内容については次で説明します。

制御情報ファイルの指定例

```
source bes1:(uoc)
srcuoc param='dir=C:¥secarea,file=(pdaudUNT1001.aud,pdaudUNT1002.aud)'
```

〔説明〕

source bes1:(uoc)：サーバ名（bes1）は HiRDB/パラレルサーバの場合に指定します。監査証跡ファイルがあるユニット内のサーバ名を指定してください。ユニット内のどのサーバ名を指定してもよいですが，バックエンドサーバを指定すると通信ネックが少なくなります。もし，監査証跡ファイルが存在するユニットが影響分散スタンバイレス型系切り替え対象ユニットの場合は，そのユニットで動作中の実行系サーバの名称を指定してください。

dir：データロード対象の監査証跡ファイルがある HiRDB ファイルシステム領域名を指定します。

file：データロード対象の監査証跡ファイル名を指定します。

監査証跡表にインデクスを定義した場合は idxwork 文及び sort 文を指定してください。

24.7.2 例題 2（HiRDB ファイルシステム領域内の全監査証跡ファイルをデータロードする場合）

HiRDB ファイルシステム領域（C:¥secarea）内のデータロード待ちの監査証跡ファイルを監査証跡表（表名：SQL_AUDIT_TRAIL）にデータロードします。このときの pdload コマンドの指定例は次のとおりです。

pdload コマンドの指定例

```
pdload -b -W SQL_AUDIT_TRAIL C:¥pdload¥control_file
```

〔説明〕

- -b 及び-W オプションは必ず指定してください。
- 表名には監査証跡表を指定します。
- 制御情報ファイルの内容については次で説明します。

制御情報ファイルの指定例

```
source bes1:(uoc)
srcuoc param='dir=C:¥secarea,file=all,mode=normal'
```

〔説明〕

source bes1:(uoc)：サーバ名（bes1）は HiRDB/パラレルサーバの場合に指定します。監査証跡ファイルがあるユニット内のサーバ名を指定してください。ユニット内のどのサーバ名を指定してもよいですが，バックエンドサーバを指定すると通信ネックが少なくなります。もし，監査証跡ファイルが存在するユニットが影響分散スタンバイレス型系切り替え対象ユニットの場合は，そのユニットで動作中の実行系サーバの名称を指定してください。

dir：データロード対象の監査証跡ファイルがある HiRDB ファイルシステム領域名を指定します。

file=all：HiRDB ファイルシステム領域内の全監査証跡ファイルをデータロードの対象にします。ただし，現用及び閉塞状態のファイルを除きます。

mode=normal：データロード済みのファイルをデータロードの対象にしません。

備考

mode=force を指定すると，データロード済みのファイルをデータロードの対象にします。次に示すような場合に force を指定します。force を指定した場合は同じデータを二重に登録する可能性があるため，事前に登録済みのデータを退避しておくなどの注意が必要になります。

- 誤って監査証跡表のデータを削除してしまったが監査証跡表のバックアップがないため，全監査証跡ファイルをデータロードして監査証跡表を回復する場合

監査証跡ファイルの状態とユーザパラメタの指定値によるデータロードの実行可否を次の表に示します。

表 24–19　監査証跡ファイルの状態とユーザパラメタの指定値によるデータロードの実行可否

<table>
<tr><th colspan="2">監査証跡ファイルの状態</th><th colspan="3">ユーザパラメタの指定値</th></tr>
<tr><th>閉塞状態</th><th>データロードの状態</th><th>mode</th><th>file ＝ファイル名</th><th>file ＝ all</th></tr>
<tr><td rowspan="2">非閉塞中</td><td>データロード待ち</td><td rowspan="4">normal</td><td colspan="2">該当する監査ファイルのデータロード処理を実行します。</td></tr>
<tr><td>データロード済み</td><td rowspan="2">該当する監査ファイルのデータロード処理をスキップして処理を続行します。</td><td rowspan="2">該当する監査ファイルのデータロード処理を対象外にします。</td></tr>
<tr><td rowspan="2">閉塞中</td><td>データロード待ち</td></tr>
<tr><td>データロード済み</td></tr>
<tr><td rowspan="2">非閉塞中</td><td>データロード待ち</td><td rowspan="4">force</td><td rowspan="2">該当する監査ファイルのデータロード処理を実行します。</td><td rowspan="3">ユーザパラメタの指定エラーになります。</td></tr>
<tr><td>データロード済み</td></tr>
<tr><td rowspan="2">閉塞中</td><td>データロード待ち</td><td rowspan="2">該当する監査ファイルのデータロード処理をスキップして処理を続行します。</td></tr>
<tr><td>データロード済み</td></tr>
<tr><td>–</td><td>現用</td><td>–</td><td>該当する監査ファイルのデータロード処理をスキップして処理を続行します。</td><td>該当する監査ファイルのデータロード処理を対象外にします。</td></tr>
</table>

（凡例）

–：該当しません。

24.7.3　データロード中にエラーが発生した場合の対処方法

データロード中にエラーが発生した場合の対処方法を説明します。

(1)　ログ取得モード又は更新前ログ取得モードで実行した場合

〈手順〉

1. データロード処理の進み具合を確認してください。データロード中は監査証跡ファイルごとにコミット処理を行うため，KFPL00800-I メッセージが出力された監査証跡ファイルまでがデータロードを完了しています。

   また，**pdls -d aud** コマンドで監査証跡ファイルの状態を確認してください。データロードを完了しているファイルはデータロード済みになっています。

2. KFPL23202-E メッセージが出力されているかを確認してください。KFPL23202-E メッセージに表示されている監査証跡ファイルはデータロードを完了しているが，ファイルの状態変更に失敗しています。このため，このファイルはデータロード待ちになっているため，このまま pdload コマンドを再実行するとデータが二重登録されてしまいます。したがって，**pdaudrm** コマンドでこのファイルを削除した後に pdload コマンドを再実行してください。

3. **pdload** コマンドを再実行します。オプションの指定，及び制御情報ファイルの指定を変更する必要はありません。

(2) ログレスモードで実行した場合

〈手順〉

1. データロード実行前に取得したバックアップを使用して，**pdrstr** コマンドで監査証跡表を格納するRD エリアを回復してください。
2. データロードが完了した監査証跡ファイルはデータロード済みになっています。そのため，このまま pdload コマンドを再実行してもデータロード済みのファイルの分がデータロードできません。この場合，mode=force を指定して pdload コマンドを再実行してください。
3. **pdload** コマンドを再実行します。file ＝ all を指定していた場合，file オペランドでデータロード対象の制御情報ファイルを指定してください。

24.8　監査証跡表の列構成

24.8.1　監査証跡表の列構成の詳細

監査証跡表の列構成を次の表に示します。

表 24–20　監査証跡表の列構成

列名称	列の内容	列のデータ型	NULL 値の指定
USER_NAME	監査対象のイベント実行者の認可識別子です。 HiRDB 管理者や一般 OS ユーザが実行するコマンド又はユティリティの場合は，OS ユーザ名が記録されます。※6	MVARCHAR(30)	なし
EXEC_DATE	イベントの実行日です。 イベント実行日とは，監査証跡の出力要求をしたサーバが，監査証跡バッファに監査証跡の出力を要求するときの標準時刻の日を示しています。	DATE	なし
EXEC_TIME	イベントの実行時刻です。 イベント実行時刻とは，監査証跡の出力要求をしたサーバが，監査証跡バッファに監査証跡の出力を要求するときの標準時刻を示しています。	TIME	なし
EXEC_TIME_MICRO	イベントの実行時刻です（単位：マイクロ秒）。 イベント実行時刻とは，監査証跡の出力要求をしたサーバが，監査証跡バッファに監査証跡の出力を要求するときの標準時刻を示しています。	INTEGER	なし
EVENT_TYPE	イベントタイプです。イベントタイプ及びイベントサブタイプについては，「イベントタイプ及びイベントサブタイプの詳細」を参照してください。	CHAR(3)	なし
EVENT_SUBTYPE	イベントサブタイプです。イベントタイプ及びイベントサブタイプについては，「イベントタイプ及びイベントサブタイプの詳細」を参照してください。	CHAR(3)	なし
EVENT_RESULT	イベントの実行結果です。権限のチェック又はイベントが成功したかどうかが記録されます。 S：権限のチェック又はイベントに成功しました。 F：権限のチェック又はイベントに失敗しました。 U：イベントの一部が失敗しました。	CHAR(1)	なし
USED_PRIVILEGE	イベントを実行したときに使用した権限です。※1イベントの終了時には‘△△△’（空白 3 文字）が記録されます。 AUD：監査権限 CNT：CONNECT 権限 DBA：DBA 権限 DEL：DELETE 権限	CHAR(3)	なし

列名称	列の内容	列のデータ型	NULL 値の指定
	INS：INSERT 権限 OWN：所有者 RDA：RD エリア利用権限 ROL：ロール SCH：スキーマ定義権限 SCO：スキーマ操作権限 SEL：SELECT 権限 SYS：HiRDB 管理者 UPD：UPDATE 権限 USG：順序数生成子使用権限		
UAP_NAME	イベント発行元の UAP 名です。※2 UAP 名が 30 文字未満の場合は，UAP 名の後ろに'△'（空白）が合計 30 文字になるまで設定されます。 それ以外の場合は NULL 値が記録されます。	VARCHAR(30)	あり
SERVICE_NAME	イベント発行元の UAP が要求したサービス名です。OpenTP1 の UAP の場合は SPP 又は MHP に要求したサービスの名称になります。 OpenTP1 以外の UAP の場合，連続した 31 個の*（アスタリスク）が記録されます。 それ以外の場合は NULL 値が記録されます。	VARCHAR(31)	あり
IP_ADDRESS	イベント発行元の IP アドレスです。イベント発行元の IP アドレスが識別できない場合は NULL 値が記録されます。	VARCHAR(63)	あり
PROCESS_ID	イベント発行元のプロセス ID です。イベント発行元のプロセス ID が識別できない場合は NULL 値が記録されます。 コマンド又はユティリティの場合は，実行されたコマンドのプロセス ID が記録されます。 システムによる監査証跡の上書き開始は監査証跡管理サーバプロセスのプロセス ID が記録されます。 Type4 JDBC ドライバから接続した場合は，0 が記録されます。 それ以外の場合は，CONNECT したシングルサーバ又はフロントエンドサーバのプロセス ID が記録されます。	INTEGER	あり
THREAD_ID	イベント発行元のスレッド ID です。イベント発行元のスレッド ID が識別できない場合は NULL 値が記録されます。	INTEGER	あり
HOST_NAME	イベントを受け付けたホスト名です。pdunit オペランドの-x オプションに指定したホスト名が記録されます。系切り替え機能を使用している場合は，現用系のホスト名が記録されます。 イベントを受け付けたホスト名が識別できない場合は NULL 値が記録されます。	VARCHAR(32)	あり
UNIT_NAME	イベントを受け付けたユニット識別子です。pdunit オペランドの-u オプションに指定したユニット識別子が記録されます。	CHAR(4)	あり

列名称	列の内容		列のデータ型	NULL 値の指定
	イベントを受け付けたユニット識別子が識別できない場合は NULL 値が記録されます。			
SERVER_NAME	イベントを受け付けたサーバ名です。HiRDB/シングルサーバの場合はシングルサーバ名が，HiRDB/パラレルサーバの場合はフロントエンドサーバ名が記録されます。pdstart オペランドの-s オプションに指定したサーバ名が記録されます。 イベントを受け付けたサーバ名が識別できない場合は NULL 値が記録されます。		VARCHAR(8)	あり
CONNECT_NUMBER	イベント発行者のコネクト通番です。イベント発行者のコネクト通番が識別できない場合は NULL 値が記録されます。		INTEGER	あり
SQL_NUMBER	イベントの SQL 通番です。イベントの SQL 通番が識別できない場合は NULL 値が記録されます。		INTEGER	あり
OBJECT_SCHEMA	オブジェクト情報の詳細※3	イベントの権限チェックの対象になるオブジェクトの所有者名です。オブジェクトの所有者が識別できない場合は NULL 値が記録されます。	MVARCHAR(30)	あり
OBJECT_NAME		イベントの権限チェックの対象になるオブジェクト名です。オブジェクト名が識別できない場合は NULL 値が記録されます。	MVARCHAR(30)	あり
OBJECT_TYPE		イベントの権限チェックの対象になるオブジェクトの種別です。オブジェクトの種別が識別できない場合は NULL 値が記録されます。 AUF：監査証跡ファイル FNC：関数 IDX：インデクス LST：リスト PRC：プロシジャ RDA：RD エリア ROL：ロール SCH：スキーマ SEQ：順序数生成子 TBL：表 TRG：トリガ TYP：データ型 VIW：ビュー表	CHAR(3)	あり
PRIVILEGE_TYPE	イベントによって付与，削除，又は変更した権限です。権限が識別できない場合は NULL 値が記録されます。 AUD：監査権限 CNT：CONNECT 権限 DBA：DBA 権限 DEL：DELETE 権限 INS：INSERT 権限		CHAR(3)	あり

列名称	列の内容	列のデータ型	NULL 値の指定
	RDA：RD エリア利用権限 ROL：ロール SCH：スキーマ定義権限 SCO：スキーマ操作権限 SEL：SELECT 権限 UPD：UPDATE 権限		
PRIVILEGE_SCHEMA	イベントによって権限を付与，削除，又は変更されたユーザの認可識別子です。又は，イベント対象になったユーザの認可識別子です。 認可識別子が識別できない場合は NULL 値が記録されます。 • GRANT 又は REVOKE：権限を付与，削除，変更された認可識別子 • 連続認証失敗アカウントロック状態への遷移又は連続認証失敗アカウントロック状態の解除：対象となった認可識別子 • パスワード無効アカウントロック状態への遷移又はパスワード無効アカウントロック状態の解除：対象となった認可識別子 • パスワードの有効期間の設定及び削除：設定及び削除の対象となった認可識別子	MVARCHAR(30)	あり
SECURITY_OPERAND	セキュリティ監査機能に関するオペランドの値です。HiRDB 開始時のオペランドの値です。開始時以外は NULL 値が記録されます。 セキュリティ監査機能に関するオペランドの値については，「セキュリティ監査機能に関するオペランドの値」を参照してください。	VARCHAR(256)	あり
AUDIT_TRAIL_TYPE	監査証跡の種別です。イベントの結果を示す監査証跡と，イベント実行中の権限チェックの結果を示す監査証跡を区別するための値を記録します。 権限チェック：NULL イベント終了：E	CHAR(1)	あり
SQL_CODE	SQL コード又は終了コードです。監査記録の対象となるイベント終了時，SQL によるイベントに対しては SQL コードを，ユティリティ（又はコマンド）に対しては終了コードを記録します。権限チェック時は NULL 値が記録されます。 イベント終了時のイベント成否エントリの値は，このコードの値によって決定します。監査記録エントリの各コードとイベント成否の詳細については，「SQL コード又は終了コードとイベント成否の詳細」を参照してください。	INTEGER	あり
FROM_AUDFILE_NAME	スワップ元の監査証跡ファイル名です。監査証跡ファイル名が識別できない場合は NULL 値が記録されます。	MVARCHAR(30)	あり
TO_AUDFILE_NAME	スワップ先の監査証跡ファイル名です。監査証跡ファイル名が識別できない場合は NULL 値が記録されます。	MVARCHAR(30)	あり

列名称	列の内容	列のデータ型	NULL 値の指定
SECURITY_PARM_TYPE	CONNECT 関連セキュリティ機能に関する変更種別です。※4	CHAR(4)	あり
BEFORE_SECURITY_PARM	変更前の CONNECT 関連セキュリティ機能の設定値です。※4 設定値が 10 文字未満の場合，設定値の後ろを空白で埋めて（設定値＋空白が 10 文字になるまで）記録されます。	CHAR(10)	あり
AFTER_SECURITY_PARM	変更後の CONNECT 関連セキュリティ機能の設定値です。※4 設定値が 10 文字未満の場合，設定値の後ろを空白で埋めて（設定値＋空白が 10 文字になるまで）記録されます。	CHAR(10)	あり
AUDIT_TABLE_OPTION	監査証跡表オプションです。権限チェック時は記録されません。次に示す場合に記録されます。 • イベントの対象オブジェクトが監査証跡表の場合 • イベントの対象オブジェクトが監査証跡表を基表としたビュー表の場合 • イベントの対象オブジェクトが監査証跡表を基表としたリストの場合 Y：操作対象オブジェクトが監査証跡表 V：操作対象オブジェクトが監査証跡表を基表としたビュー表 L：操作対象オブジェクトが監査証跡表を基表としたリスト NULL 値：上記以外又は権限チェック時 監査証跡表オプションの出力有無の詳細については，「監査証跡表オプションの出力有無」を参照してください。	CHAR(1)	あり
ACCESS_COUNT	ユーザが，オブジェクト（実表，ビュー表，及びリスト）に対して検索，挿入，更新，及び削除をした行数です。取得できるアクセス件数は 0～2147483647 で，アクセス件数の取得に失敗した場合は NULL 値が記録されます。 アクセス件数の詳細については，「アクセス件数の詳細」を参照してください。	INTEGER	あり
SQL_SOURCE	実行した SQL 文です。一つの SQL 文で複数のオブジェクトが操作される場合，最初の監査証跡に SQL 文が記録され，以降の監査証跡の SQL 文には NULL 値が記録されます。また，次の場合も NULL 値が記録されます。 • 実行した SQL 文がない場合 • システム共通定義の pd_aud_sql_source_size オペランドを指定していない，又は 0 を指定した場合 記録する SQL 文の長さは，システム共通定義の pd_aud_sql_source_size オペランドで指定します。記録する SQL 文の長さがシステム定義の指定値よりも長い場合は，SQL 文の先頭から指定値までの内容が記録され，残りは切り捨てられます。	BINARY(2000000)	あり
SQL_DATA	実行した SQL 文で入力に指定した埋込み変数，又は SQL 記述領域のデータが記録されます。一つの SQL 文で複数のオブジェクトが操作される場合，最初の監査証跡に SQL データが記録され，	BINARY(1000000)	あり

列名称	列の内容	列のデータ型	NULL 値の指定
	以降の監査証跡の SQL データには NULL 値が記録されます。また，次の場合も NULL 値が記録されます。 • 入力に指定した埋込み変数，又は SQL 記述領域がない場合 • システム共通定義の pd_aud_sql_data_size オペランドを指定していない場合 記録する SQL データの長さは，システム共通定義の pd_aud_sql_data_size オペランドで指定します。記録する SQL データの長さがシステム定義の指定値よりも長い場合は，SQL データの先頭から指定値までの内容が記録され，残りは切り捨てられます。SQL データの詳細については，「SQL データの詳細」を参照してください。		
USER_INFO_1	埋込み言語 DECLARE AUDIT INFO SET を宣言し，ユーザ付加情報 1〜3 に埋込み変数を設定している場合，各ユーザ付加情報の値が記録されます。ただし，次の場合は NULL 値が記録されます。 • 埋込み言語 DECLARE AUDIT INFO SET を宣言していない場合 • ユーザ付加情報の値を設定していない場合 • ユーザ付加情報の値に NULL 値を指定して，設定を解除している場合 • ユーザ付加情報の値が識別できない場合	VARCHAR(100)	あり
USER_INFO_2		VARCHAR(100)	あり
USER_INFO_3		VARCHAR(100)	あり
PRODUCT_INFO_1	HiRDB の関連製品（Cosminexus のバージョン 07-50 以降）と連携している場合，監査記録の対象となるイベントには，HiRDB の関連製品からのルートアプリケーション情報が記録されます。該当する情報がない場合，又は情報が認識できない場合は，NULL 値が記録されます。	VARCHAR(255)	あり
PDLOAD_TIMESTAMP	pdload コマンド実行による監査証跡表へのデータ登録（監査証跡表へのデータロード）時の時刻印です。※5 時刻印はシステムマネジャのあるサーバマシンの時刻印を使用し，一度の pdload の実行ですべて同じ値を記録します。	TIMESTAMP(2)	あり
PDLOAD_SEQNUM	1 回の pdload コマンド実行による監査証跡表へのデータ登録時の通番です。※5 pdload コマンドの実行ごとに 1〜10 けたの 10 進数の値（1〜（2^{31}–1））を設定します。pdload コマンドを実行している間に最大値（2^{31}–1）に到達した場合，1 に戻して記録します。	INTEGER	あり

注

- NULL 値の指定がありの列の場合は非ナル値制約に NULL が設定されています。NULL 値なしの列の場合は非ナル値制約に NOT NULL が設定されています。
- バージョンアップの前後で監査証跡表の列構成が変わる場合，バージョンアップ後にバージョンアップ前のデータを監査証跡表に登録すると，既存行の新規追加列には NULL 値が設定されます。

注※1

DBA 権限または監査人権限が必要なディクショナリ表を参照すると，DBA または SEL のどちらかを出力します。監査人権限が必要なディクショナリ表を参照すると，AUD または SEL のどちらかを出力します。

注※2

検索時のキーとして使用する場合は，CAST 指定によって CHAR(30)に属性を変換して条件式に指定してください。

（例）

```
SELECT * FROM SQL_AUDIT_TRAIL
    WHERE UAP_NAME=CAST('UAP11' AS CHAR(30));
```

注※3

権限チェック時：

権限チェックの対象となるオブジェクトを記録します。ただし，実行者の権限チェックをした場合は，チェック対象のオブジェクトはないため記録しません。

（例）

- 表 T1 に対するアクセス権限をチェックした場合の対象オブジェクト：表 T1
- 実行者が DBA 権限を持っているかチェックした場合の対象オブジェクト：NULL 値

イベント終了時：

イベントの対象となるオブジェクトを記録します。イベントの対象に指定したオブジェクトがない場合は，オブジェクト種別は表として記録されます。

（例）

- 表を検索した場合の対象オブジェクト：検索した表

注※4

次に示す場合に記録されます。なお，記録される情報については，「CONNECT 関連セキュリティ機能使用時に記録される情報」を参照してください。

- CONNECT 関連セキュリティ機能の設定を変更した場合
- パスワードを登録又は変更した場合

注※5

監査証跡ファイルには記録されません。

注※6

OS ユーザ名は 30 バイトまで記録されます。31 バイト以降は切り捨てます。

(1)　イベントタイプ及びイベントサブタイプの詳細

イベントタイプ及びイベントサブタイプの詳細を次の表に示します。

表 24–21　イベントタイプ及びイベントサブタイプの詳細

監査イベント	イベントタイプの値	イベントサブタイプの値	詳細
システム管理者セキュリティイベント	SYS	STR	HiRDB の開始
		STP	HiRDB の終了
		MOD	主監査人の登録，又は監査証跡表の作成（pdmod コマンド）
		MDA	副監査人の登録（pdmod コマンド）
		ARM	監査証跡ファイルの削除（pdaudrm コマンド）
		ABG	監査証跡の取得開始（pdaudbegin コマンド，HiRDB の開始）
		AEN	監査証跡の取得終了（pdaudend コマンド，HiRDB の終了）
		OVW	監査証跡ファイルの上書き開始
		CLK	連続認証失敗アカウントロック状態への遷移
		CUL	連続認証失敗アカウントロック状態の解除
		PLK	パスワード無効アカウントロック状態への遷移
		PUL	パスワード無効アカウントロック状態の解除
		SPR	CONNECT 関連セキュリティ機能に関するオペランドの指定値変更
		ULK	pdacnlck コマンドの実行
監査人セキュリティイベント	AUD	ALD	監査人による監査証跡表へのデータロード（pdload コマンド）
		ATL	HiRDB による監査証跡表へのデータロード（pdload コマンド）
		ASW	監査証跡ファイルのスワップ（pdaudswap コマンド）
		CRT	監査対象イベントの定義（CREATE AUDIT）
		DRP	監査対象イベントの削除（DROP AUDIT）
		GRT	監査人のパスワード変更（GRANT AUDIT）
		ATB	監査証跡表の自動データロード機能再開始（pdaudatld -b コマンド）
		ATT	監査証跡表の自動データロード機能停止（pdaudatld -t コマンド，データロード失敗）
		APT	JP1/NETM/Audit 用監査ログ出力ファイルへの出力（pdaudput コマンド）
セッションセキュリティイベント	SES	CNT	HiRDB への接続（CONNECT 文）
		ATH	ユーザの変更（SET SESSION AUTHORIZATION 文）
		ROL	現在ロールの設定（SET ROLE 文）

監査イベント	イベントタイプの値	イベントサブタイプの値	詳細
		DIS	HiRDB との切り離し（DISCONNECT 文）
権限管理イベント	PRV	GRT	ユーザ権限の付与（GRANT 文）
			パスワードの変更（CONNECT 関連セキュリティ機能の設定変更種別にパスワードの変更であることが分かる情報を付与する）
		RVK	ユーザ権限の削除（REVOKE 文）
オブジェクト定義イベント	DEF	CRT	オブジェクトの定義（CREATE 系 SQL）
		DRP	オブジェクトの削除（DROP 系 SQL）
		ALT	オブジェクトの変更（ALTER 系 SQL）
オブジェクト操作イベント	ACS	SEL	表の検索（SELECT 文）
		INS	表への行挿入（INSERT 文）
		UPD	表の行更新（UPDATE 文）
		DEL	表からの行削除（DELETE 文）
		PRG	表の全行削除（PURGE TABLE 文）
		CAL	ストアドプロシジャの実行（CALL 文）
		LCK	表の排他制御（LOCK TABLE 文）
		ASN	リストの作成（ASSIGN LIST 文）
		NXV	順序数生成子が生成する値の返却（NEXT VALUE 式）
ユティリティ操作イベント	UTL	LOD	データベース作成ユティリティ（pdload コマンド）
		ORG	データベース再編成ユティリティ（pdrorg コマンド）
		EXP	ディクショナリ搬出入ユティリティ（pdexp コマンド）
			pddefrev コマンド
		CST	整合性チェックユティリティ（pdconstck コマンド）

(2)　セキュリティ監査機能に関するオペランドの値

セキュリティ監査機能に関するオペランドの値を次の表に示します。また，オペランドの値を記録するときのデータサイズを次の図に示します。

表 24–22　セキュリティ監査機能に関するオペランドの値

オペランド名	データサイズ	格納される値
pd_audit	1 バイト	Y 又は N
pd_aud_file_name	最大 167 バイト	監査証跡ファイル用の HiRDB ファイルシステム領域名

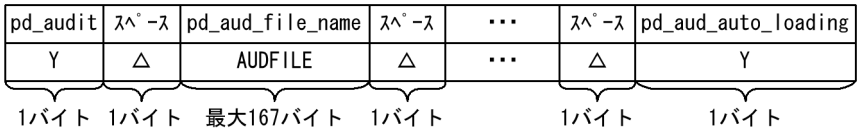
オペランド名	データサイズ	格納される値
pd_aud_max_generation_size	最大 4 バイト	1～5240
pd_aud_max_generation_num	最大 3 バイト	2～200
pd_aud_no_standby_file_opr	最大 10 バイト	down 又は forcewrite
pd_aud_async_buff_size	最大 7 バイト	0，4096～6553600
pd_aud_async_buff_count	最大 4 バイト	1～6500
pd_aud_file_wrn_pnt	最大 6 バイト	0～100，0～99
pd_audit_def_buffer_size※	最大 7 バイト	0～2000000
pd_aud_async_buff_retry_intvl※	最大 4 バイト	0～1000
pd_aud_sql_source_size※	最大 7 バイト	0～2000000
pd_aud_sql_data_size※	最大 7 バイト	0～1000000
pd_aud_auto_loading※	1 バイト	Y 又は N

注※

このオペランドはバージョンアップによって追加されたものです。バージョン 08-02 より前の HiRDB で作成した監査証跡ファイルを，バージョンアップ後の監査証跡表に登録した場合は，次の指定値が仮定されます。

オペランド名	仮定される指定値
pd_audit_def_buffer_size	0
pd_aud_async_buff_retry_intvl	50
pd_aud_sql_source_size	0
pd_aud_sql_data_size	0
pd_aud_auto_loading	N

図 24–9　セキュリティ監査機能に関するオペランドの値を記録するときのデータ形式



(3)　SQL コード又は終了コードとイベント成否の詳細

SQL コード又は終了コードとイベント成否の詳細を次の表に示します。

表 24–23　SQL コード又は終了コードとイベント成否の詳細

イベント名称	SQL コード又は 終了コードの値	意味	イベント 成否の値	備考
各種 SQL	0 以上	成功	S	−
	負	失敗	F	
pdstart	0	成功	S	−
	4	成功	S	
	8	失敗	F	次に示す場合は監査証跡を取得できません。 • 監査証跡管理サーバプロセス開始前に失敗となった場合 • pdstart コマンドの失敗によってユニットが異常終了した場合
pdstop	0	成功	S	監査証跡管理サーバプロセス終了後に失敗となった場合は，成功の監査証跡を出力します。
	4	成功	S	−
	8	失敗	F	
pdmod（監査人の登録，監査証跡表の作成）	0	成功	S	−
	8	失敗	F	
pdaudrm	0	成功	S	−
	8	失敗	F	
pdaudbegin	0	成功	S	−
	4	一部 失敗	U	
	8	失敗	F	次に示す場合は監査証跡を取得できません。 • 監査証跡ファイルが全世代使用済みの場合 • アンロード待ちで書き込む世代がないため pdaudbegin コマンドが失敗した場合
pdaudend	0	成功	S	−
	8	失敗	F	
pdacunlck	0	成功	S	−
	8	失敗	F	
pdaudswap	0	成功	S	監査証跡ファイルの上書き時に，スワップ元及びスワップ先の監査証跡ファイル名が記録されます。
	8	失敗	F	監査証跡ファイルの上書き時に，スワップ元の監査証跡ファイル名が記録されます。スワップ先の監査証跡ファイル名には NULL 値が記録されます。

イベント名称	SQL コード又は 終了コードの値	意味	イベント 成否の値	備考
pdload	0	成功	S	−
	4	成功	S	
	8	失敗	F	
pddefrev	0	成功	S	−
	4	成功	S	
	8	失敗	F	
	12	失敗	F	
pdrorg	0	成功	S	−
	4	成功	S	
	8	失敗	F	
pdexp	0	成功	S	−
	4	成功	S	
	8	失敗	F	
	12	失敗	F	
pdconstck	0	成功	S	−
	4	成功	S	警告終了（整合性違反あり）
	8	失敗	F	−
pdaudatld	0	成功	S	−
	4	成功	S	警告終了（自動データロード機能は停止できているため，成功として扱います）
	8	失敗	F	−

（凡例）−：該当しません。

(4)　CONNECT 関連セキュリティ機能使用時に記録される情報

CONNECT 関連セキュリティ機能使用時に記録される情報を次の表に示します。

表 24–24　CONNECT 関連セキュリティ機能使用時に記録される情報

機能名	イベント タイプ	イベント サブタイプ	変更 種別	変更前後に記録される値
連続認証失敗許容回数	SYS	SPR	ECNT	1〜10，又は NULL 値
アカウントロック期間			LCKM	10〜44640，UNLIMITED，又は NULL 値

機能名	イベントタイプ	イベントサブタイプ	変更種別	変更前後に記録される値
パスワードの最小許容バイト数			MINL	1～15，又は NULL 値
認可識別子の指定禁止			USID	RESTRICT，UNRESTRICT，又は NULL 値
単一文字種の指定禁止			SMLN	
英大文字の指定必須の設定及び削除			UPPR	"ON"又は NULL 値
英小文字の指定必須の設定及び削除			LOWR	"ON"又は NULL 値
数字の指定必須の設定及び削除			NUMR	"ON"又は NULL 値
強制文字種別 ALL 指定の設定及び削除			FCAL	"ON"又は NULL 値
再利用禁止の設定及び削除			RUSE	1～36，又は NULL 値
IP アドレスによる接続制限の登録又は削除			CNST	変更前は NULL 値を記録します。 変更後は次のように記録します。 登録（許可）の場合 ”CRT_ALLOW” 登録（拒否）の場合 ”CRT_DENY” 削除の場合 ”DRP” また，指定した制約名称を OBJECT_NAME 列に格納します。
パスワードの変更※	PRV 又は AUD	GRT	CPWD	NULL 値
パスワードの有効期間の設定及び削除	SYS	SPR	INTV	1～999，又は NULL 値を記録します。 また，指定した認可識別子を PRIVILEGE_SCHEMA 列に格納します。

注

次に示す場合は NULL 値が記録されます。

- 変更前後に記録される値で，ディクショナリ表 SQL_SYSPARAMS 中の対応する列に値がない場合（設定を新規で行うときなど）
- NULL 値の場合
- 値が識別できない場合

また，パスワードの文字列制限の事前チェック時は，変更種別ごとに監査証跡を取得しますが，変更前後に記録される値には NULL 値が記録されます。

注※

イベントタイプ，イベントサブタイプ，及び SQL コードだけでは，GRANT 文の実行時にパスワードが実際に変更されたかどうか HiRDB が判断できません。GRANT 文の実行時に登録済みユーザのパスワードが変更された場合に限り，変更種別に CPWD を設定します。パスワードの変更時に出力される変更種別を次の表に示します。

表 24–25　パスワードの変更時に出力される変更種別

登録済みユーザのパスワード有無	GRANT CONNECT，DBA，又は AUDIT の実行結果	変更種別
パスワードあり	パスワードの変更	CPWD
	パスワードの削除	CPWD
	パスワードの変更なし	NULL 値
パスワードなし	パスワードありに変更	CPWD
	パスワードなしのまま	NULL 値

注　ここでいうユーザ，パスワードとは HiRDB が管理するユーザとパスワードのことです。

(5)　監査証跡表オプションの出力有無

監査証跡表オプションの出力有無を次の表に示します。

表 24–26　監査証跡表オプションの出力有無

実行結果	失敗の要因	監査証跡表オプションの出力有無
成功	−	出力する
失敗※	対象のオブジェクトが存在しない	出力しない（NULL 値）
	対象のオブジェクトが監査証跡表のため，エラーになる場合	出力する
	上記以外のエラー	出力する（情報が取得できない場合は出力しない）

（凡例）−：該当しません。

注※

監査証跡表オプションを出力していない場合に，監査証跡表であるかどうかを調べるには，監査記録エントリのオブジェクトの所有者とオブジェクトの名称の組み合わせなどから判断してください。

(6)　アクセス件数の詳細

アクセス件数の詳細を次の表に示します。

表 24–27　アクセス件数の詳細

アクセス手段		アクセス件数
SQL	SELECT	SELECT 文での結果集合に対して，ユーザがアクセス（CLOSE が成功，又は失敗するまで FETCH）した行数です。なお，ブロック転送の場合は，1 ブロックに含まれる行数をアクセス件数とします。 問合せに複数表を指定した場合，その複数の表すべてに対して，同じアクセス件数とします。次のどちらかの場合，アクセス件数が NULL 値になります。 • 副問合せに指定した表（詳細については表「副問合せでのアクセス件数」を参照してください） • 集合演算 EXCEPT 〔ALL〕で右側のオペランドに指定した表（例については，図「アクセス件数の出力例（その 6）」を参照してください）
	INSERT	挿入した行数です。
	UPDATE	更新した行数です。
	DELETE	削除した行数です。
ユティリティ	pdload	表ごとのデータロード件数です。
	pdrorg	表ごとのアンロード件数，及び表ごとのリロード件数です。
	pdconstck	制約違反となったキー値の数の総和です。

表 24–28　副問合せでのアクセス件数

表の指定箇所		アクセス件数	備考
スカラ副問合せ	値式を指定できる箇所	NULL 値	例については，図「アクセス件数の出力例（その 7）」参照
行副問合せ	行値構成子を指定できる箇所	NULL 値	例については，図「アクセス件数の出力例（その 8）」参照
	UPDATE 文の SET 句	NULL 値	例については，図「アクセス件数の出力例（その 9）」参照
表副問合せ	IN 述語の右側	NULL 値	例については，図「アクセス件数の出力例（その 10）」参照
	限定述語の右側	NULL 値	例については，図「アクセス件数の出力例（その 11）」参照
	EXISTS 述語	NULL 値	例については，図「アクセス件数の出力例（その 12）」参照
	FROM 句の導出表	非 NULL 値※	例については，図「アクセス件数の出力例（その 13）」参照

注※

FROM 句の導出表の検索結果が，ユーザに直接返らない場合は，アクセス件数が NULL 値となります（例については，図「アクセス件数の出力例（その 14）」参照）。

アクセス手段が SELECT の場合の，アクセス件数の出力例を次の図に示します。

- 図「アクセス件数の出力例（その 1）」
- 図「アクセス件数の出力例（その 2）」
- 図「アクセス件数の出力例（その 3）」
- 図「アクセス件数の出力例（その 4）」
- 図「アクセス件数の出力例（その 5）」
- 図「アクセス件数の出力例（その 6）」
- 図「アクセス件数の出力例（その 7）」
- 図「アクセス件数の出力例（その 8）」
- 図「アクセス件数の出力例（その 9）」
- 図「アクセス件数の出力例（その 10）」
- 図「アクセス件数の出力例（その 11）」
- 図「アクセス件数の出力例（その 12）」
- 図「アクセス件数の出力例（その 13）」
- 図「アクセス件数の出力例（その 14）」

図 24–10　アクセス件数の出力例（その 1）

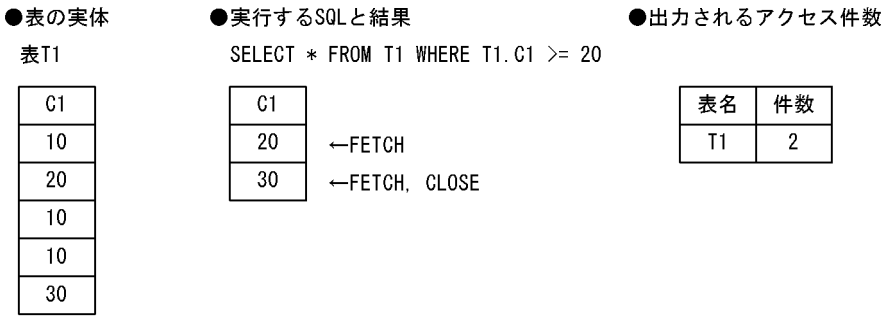


図 24–11　アクセス件数の出力例（その 2）



図 24–12　アクセス件数の出力例（その 3）

●実行するSQLと結果

```
SELECT * FROM T1
UNION
SELECT * FROM T2
```

T1.C1,T2.C1	
10	←FETCH
20	←FETCH
30	←FETCH,CLOSE

●出力されるアクセス件数

表名	件数
T1	3
T2	3

注　表の実体については，図「アクセス件数の出力例（その 2）」と同じです。

図 24–13　アクセス件数の出力例（その 4）

●実行するSQLと結果

```
INSERT INTO T1 SELECT C1 FROM T2
```

T1.C1	
10	
30	
30	
10	←INSERT
20	←INSERT

T2.C1
10
20

●出力されるアクセス件数

表名	件数
T1	2
T2	2

注　表の実体については，図「アクセス件数の出力例（その 2）」と同じです。

図 24–14　アクセス件数の出力例（その 5）

●表の実体

表T1

C1	C2
10	A
30	B
30	C

表T2

C1	C2
10	A
10	D
20	E

●実行するSQLと結果

```
WITH QRY1 (QC1,QC2,QC3,QC4) AS
       (SELECT * FROM T1,T2),
       QRY2(QC1,QC2) AS
       (SELECT * FORM T2)
SELECT * FROM QRY1 WHERE (QC1,QC2) IN
       (SELECT * FROM QRY2)
```

QC1	QC2	QC3	QC4	
10	A	10	A	←FETCH
10	A	10	D	←FETCH
10	A	20	E	←FETCH, CLOSE

●出力されるアクセス件数

表名	件数
T1	3
T2	3
T2※	NULL

注※
結果がユーザに直接返らない表についてのアクセス件数はNULL値とします。

図 24–15　アクセス件数の出力例（その 6）

●実行するSQLと結果

```
SELECT * FROM T1
EXCEPT
SELECT * FROM T2
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

●出力されるアクセス件数

表名	件数
T1	2
T2※	NULL

注※
集合演算EXCEPT〔ALL〕で右側のオペランドに指定した表がT2のため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–16　アクセス件数の出力例（その 7）

●実行するSQLと結果

```
SELECT * FROM T1 WHERE T1.C1 >=
       (SELECT MAX(T2.C1) FROM T2)
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

MAX(T2.C1)
20

●出力されるアクセス件数

表名	件数
T1	2
T2※	NULL

注※
スカラ副問合せの値式を指定できる箇所に表T2があるため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–17　アクセス件数の出力例（その 8）

●実行するSQLと結果

```
SELECT * FROM T1 WHERE (C1,C2) >=
       (SELECT * FROM T2 WHERE C2 = 'E')
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

T2.C1	T2.C2
20	E

●出力されるアクセス件数

表名	件数
T1	2
T2※	NULL

注※
行副問合せの行値構成子を指定できる箇所に表T2があるため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–18　アクセス件数の出力例（その 9）

●実行するSQLと結果

```
SELECT T1 SET (C1,C2) =
       (SELECT * FROM T2 WHERE C2 = 'E')
```

T1.C1	T1.C2	
20	E	←UPDATE
20	E	←UPDATE
20	E	←UPDATE

T2.C1	T2.C2
20	E

●出力されるアクセス件数

表名	件数
T1	3
T2※	NULL

注※
行副問合せのUPDATE文のSET句に表T2があるため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–19　アクセス件数の出力例（その 10）

●実行するSQLと結果

```
SELECT * FROM T1 WHERE (C1,C2) IN
       (SELECT * FROM T2)
```

T1.C1	T1.C2	
10	A	←FETCH,CLOSE

T2.C1	T2.C2
10	A
10	D
20	E

●出力されるアクセス件数

表名	件数
T1	1
T2※	NULL

注※
表副問合せのIN述語の右側に表T2があるため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–20　アクセス件数の出力例（その 11）

●実行するSQLと結果

```
SELECT * FROM T1 WHERE (C1,C2) =
       SOME(SELECT C1,C2 FROM T2)
```

T1.C1	T1.C2	
10	A	←FETCH,CLOSE

T2.C1	T2.C2
10	A
10	D
20	E

●出力されるアクセス件数

表名	件数
T1	1
T2※	NULL

注※
表副問合せの限定述語の右側に表T2があるため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–21　アクセス件数の出力例（その 12）

●実行するSQLと結果

```
SELECT * FROM T1 WHERE NOT EXISTS
       (SELECT * FROM T2 WHERE T1.C1 = T2.C1)
```

T1.C1	T1.C2	
30	B	←FETCH
30	C	←FETCH,CLOSE

T2.C1	T2.C2
10	A

●出力されるアクセス件数

表名	件数
T1	2
T2※	NULL

注※
表副問合せのEXISTS述語に表T2があるため，件数はNULL値となります。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–22　アクセス件数の出力例（その 13）

●実行するSQLと結果

```
SELECT * FROM (SELECT * FROM T1) AS D1
```

T1.C1	T1.C2	
10	A	←FETCH
30	B	←FETCH
30	C	←FETCH,CLOSE

●出力されるアクセス件数

表名	件数
T1※	3

注※
表副問合せのFROM句の導出表に表T2があるが，FROM句の導出表の検索結果はユーザに直接返却されないため，件数はNULL値になりません。

注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

図 24–23　アクセス件数の出力例（その 14）

●実行するSQLと結果

```
SELECT * FROM T1 WHERE (C1,C2) IN
  (SELECT * FROM (SELECT * FROM T2) AS D1)
```

T1.C1	T1.C2	
10	A	←FETCH,CLOSE

T2.C1	T2.C2
10	A
10	D
20	E

●出力されるアクセス件数

表名	件数
T1	1
T2※	NULL

注※
表副問合せのFROM句の導出表に表T2があるため，件数はNULL値となります。

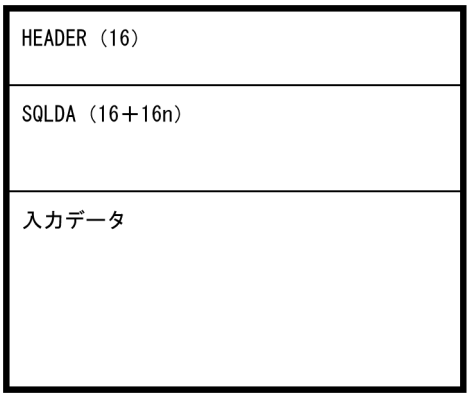
注　表の実体については，図「アクセス件数の出力例（その 5）」と同じです。

(7)　SQL データの詳細

(a)　SQL データの構造

SQL データの構造を次の図に示します。

図 24–24　SQL データの構造



（凡例）

n：SQLN に対応する，SQLVAR の個数を示します。SQLDA（SQL 記述領域）の構造については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

注 1　( )内は，領域の長さ（単位：バイト）を示します。

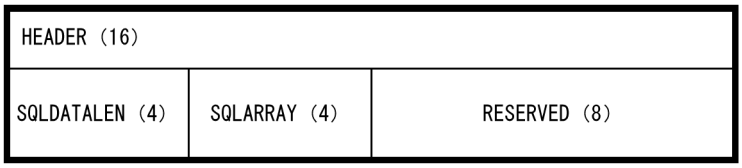
注 2　記録されるエンディアン形式は，サーバの OS に依存します。

次に，SQL データを構造する HEADER 領域，SQLDA（SQL 記述領域）及び入力データ領域の各構造について説明します。

●HEADER 領域

HEADER 領域の構造を次の図に示します。また，各構成要素の詳細を次の表に示します。

図 24–25　HEADER 領域の構造



注　( )内は，領域の長さ（単位：バイト）を示します。

表 24–29　HEADER 領域の構成要素の詳細

レベル番号※		領域名	データ型	長さ	内容
1	1	HEADER	−	16	SQL データの管理情報です。
	2	SQLDATALEN	unsigned long	4	記録前の SQL データのデータ長です。
	2	SQLARRAY	long	4	配列型の変数を使って複数行分のデータを設定した場合の配列要素数です。配列型の変数を使用しない場合は，1 が設定されます。
	2	RESERVED	char	8	システムが使用する予備領域です。

（凡例）

−：該当しません。

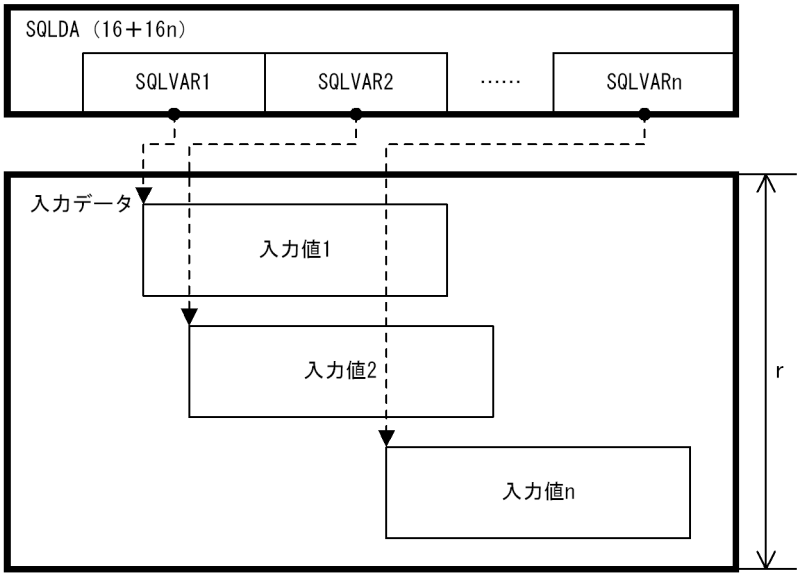
注※

HEADER 領域での包含関係を示しています。この場合，レベル番号 1 の領域は，レベル番号 2 の領域から構造されていることを示します。

●SQLDA 及び入力データ領域

SQLDA 及び入力データ領域の構造を次の図に示します。SQLDA（SQL 記述領域）の構造については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

図 24–26　SQLDA 及び入力データ領域の構造



（凡例）

n：SQLN に対応する，SQLVAR の個数を示します。

r：入力データ領域の長さを示します。入力データ領域の長さは，次の計算式で求められます。

入力データ領域の長さ＝ SQL データの長さ−HEADER 領域の長さ−SQLDA の長さ

注 1　( )内は，領域の長さ（単位：バイト）を示します。

注 2　破線の矢印は，SQLDA からのオフセットを示します。

(b)　入力値の構造

入力データ領域を構成する入力値は，次に示す要素によって構造が異なります。詳細については，それぞれの参照先にある図を参照してください。

入力値の種類	標識変数の有無	参照先
単純構造の入力値	あり	図「単純構造の入力値（標識変数あり）の構造」

入力値の種類	標識変数の有無	参照先
	なし	図「単純構造の入力値（標識変数なし）の構造」
繰返し構造の入力値	あり	図「繰返し構造の入力値（標識変数あり）の構造」
	なし	図「繰返し構造の入力値（標識変数なし）の構造」
固定長のデータ型で配列を使った入力値	あり	図「固定長のデータ型で配列を使った入力値（標識変数あり）の構造」
	なし	図「固定長のデータ型で配列を使った入力値（標識変数なし）の構造」
可変長のデータ型で配列を使った入力値	あり	図「可変長のデータ型で配列を使った入力値（標識変数あり）の構造」
	なし	図「可変長のデータ型で配列を使った入力値（標識変数なし）の構造」

●入力値の種類が，単純構造の入力値の場合

図 24–27　単純構造の入力値（標識変数あり）の構造

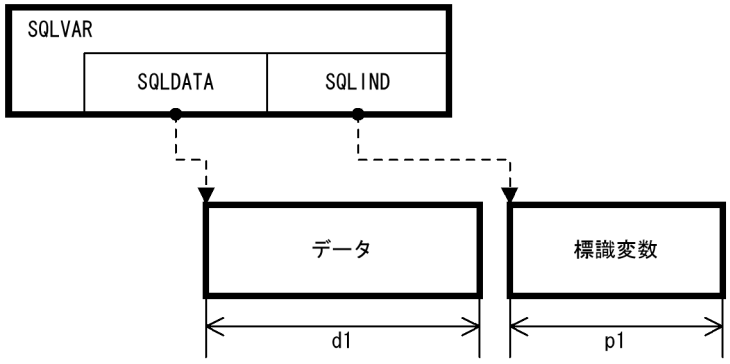
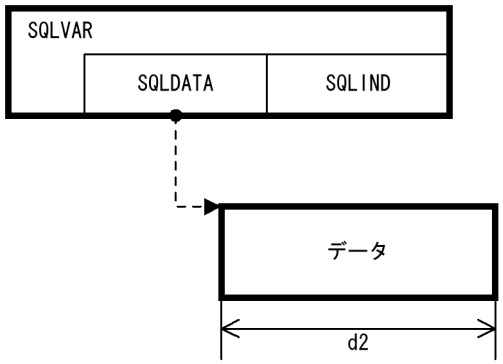


図 24–28　単純構造の入力値（標識変数なし）の構造



（凡例）

d1，d2：データの長さを示します。データの長さについては，マニュアル「HiRDB UAP 開発ガイド」の「SQL 記述領域に設定するデータコードとデータの長さ」を参照してください。

p1：標識変数の長さを示します。SQLVAR の SQLCOD に対応します。

注　破線の矢印は，SQLDA からのオフセットを示します。

〔説明〕

データ及び標識変数は，SQLVAR の SQLCOD 及び SQLLEN，又は SQLVAR_LOB の SQLCOD 及び SQLLOBLEN に設定されたデータ型に合わせた構造になります。

データ及び標識変数は，境界調整を行った領域に格納されます。このため，各領域の長さには境界調整で発生した領域の長さも含まれます。

データの記述については，マニュアル「HiRDB UAP 開発ガイド」の「SQL のデータ型と C 言語のデータ記述」又は「SQL のデータ型と COBOL 言語のデータ記述」を参照してください。

●入力値の種類が，繰返し構造の入力値の場合

図 24–29　繰返し構造の入力値（標識変数あり）の構造

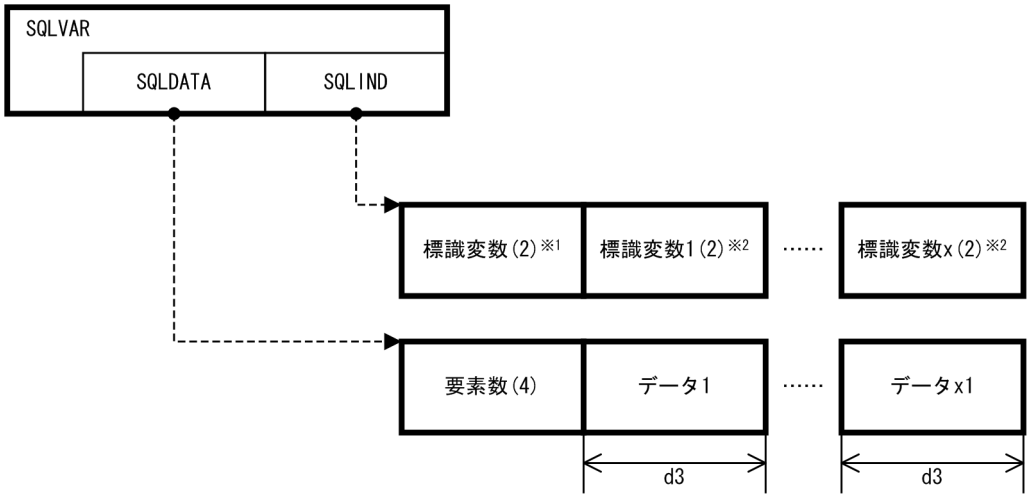
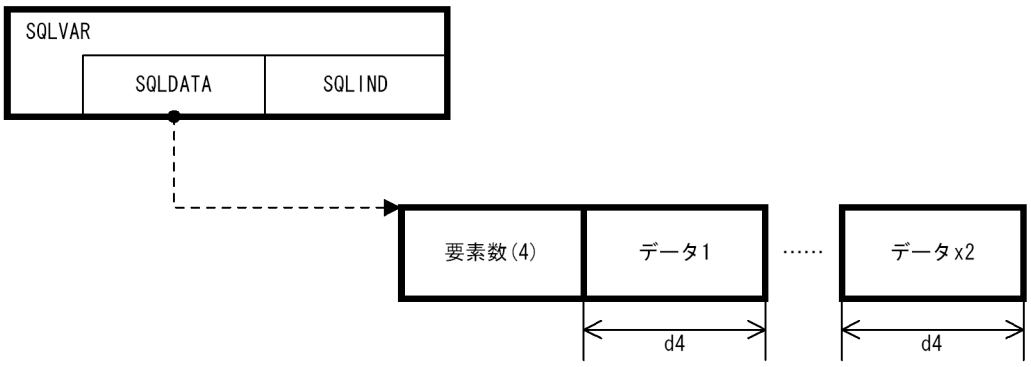


図 24–30　繰返し構造の入力値（標識変数なし）の構造



（凡例）

x：標識変数の個数を示します。

x1，x2：要素数で指定した，データの個数を示します。

d3，d4：SQLVAR の SQLSYS に対応する，データの長さを示します。

注 1　( )内は，領域の長さ（単位：バイト）を示します。

注 2　破線の矢印は，SQLDA からのオフセットを示します。

注※1　繰返し構造全体の標識変数が格納されます。

注※2　各要素の標識変数が格納されます。

〔説明〕

SQLVAR の SQLXDIM が 2 以上の場合，繰返し指定で指定した要素数分の要素が格納されます。

データは，SQLVAR の SQLCOD に設定されたデータ型に合わせた構造になります。

データ及び標識変数は，境界調整を行った領域に格納されます。このため，各領域の長さには境界調整で発生した領域の長さも含まれます。

データの記述については，マニュアル「HiRDB UAP 開発ガイド」の「SQL のデータ型と C 言語のデータ記述」又は「SQL のデータ型と COBOL 言語のデータ記述」を参照してください。

●入力値の種類が，固定長のデータ型で配列を使った入力値の場合

図 24–31　固定長のデータ型で配列を使った入力値（標識変数あり）の構造

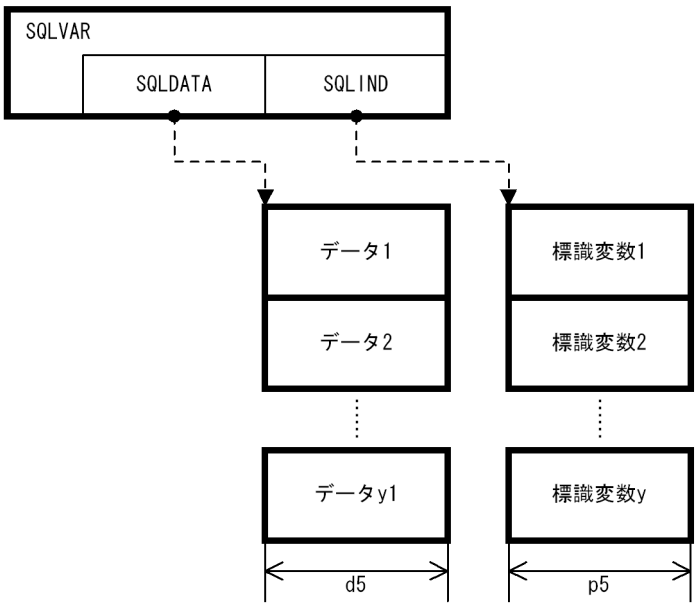
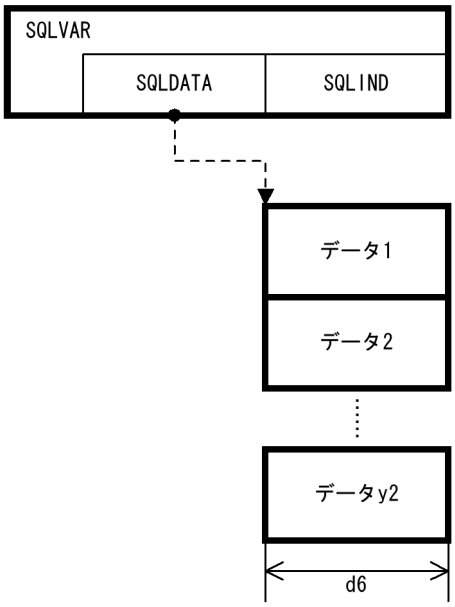


図 24–32　固定長のデータ型で配列を使った入力値（標識変数なし）の構造



（凡例）

y：標識変数の個数を示します。

y1，y2：HEADER 領域の SQLARRAY に対応する，配列の要素数を示します。

d5，d6：SQLVAR の SQLSYS に対応する，データの長さを示します。

p5：SQLVAR の SQLCOD に対応する，標識変数の長さを示します。

注　破線の矢印は，SQLDA からのオフセットを示します。

〔説明〕

HEADER 領域の SQLARRAY が 2 以上の場合，配列型で指定した要素数分の要素が格納されます。

データ及び標識変数は，SQLVAR の SQLCOD 及び SQLLEN，又は SQLVAR_LOB の SQLCOD 及び SQLLOBLEN に設定されたデータ型に合わせた構造になります。

データ及び標識変数は，境界調整を行った領域に格納されます。このため，各領域の長さには境界調整で発生した領域の長さも含まれます。

データの記述については，マニュアル「HiRDB UAP 開発ガイド」の「SQL のデータ型と C 言語のデータ記述」又は「SQL のデータ型と COBOL 言語のデータ記述」を参照してください。

●入力値の種類が，可変長のデータ型で配列を使った入力値の場合

図 24–33　可変長のデータ型で配列を使った入力値（標識変数あり）の構造

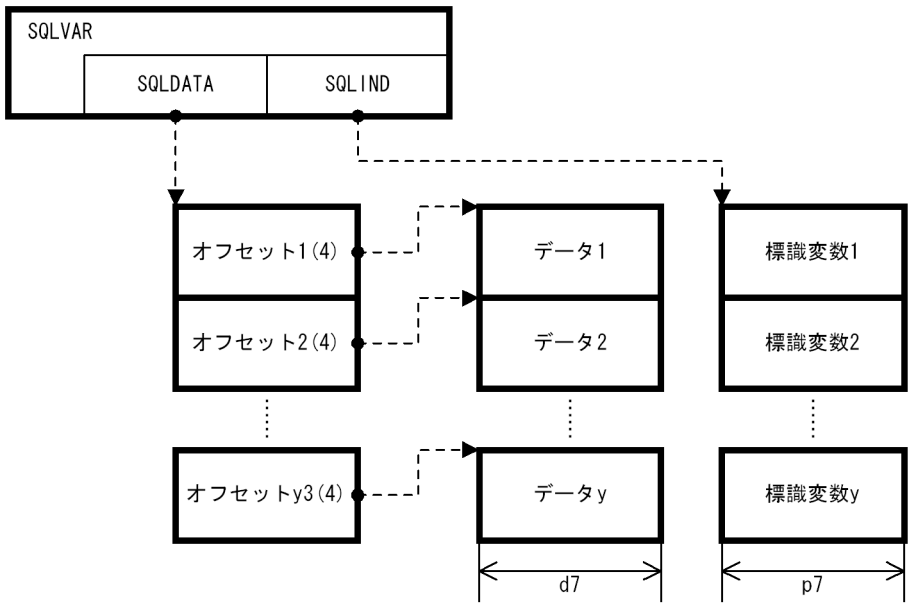
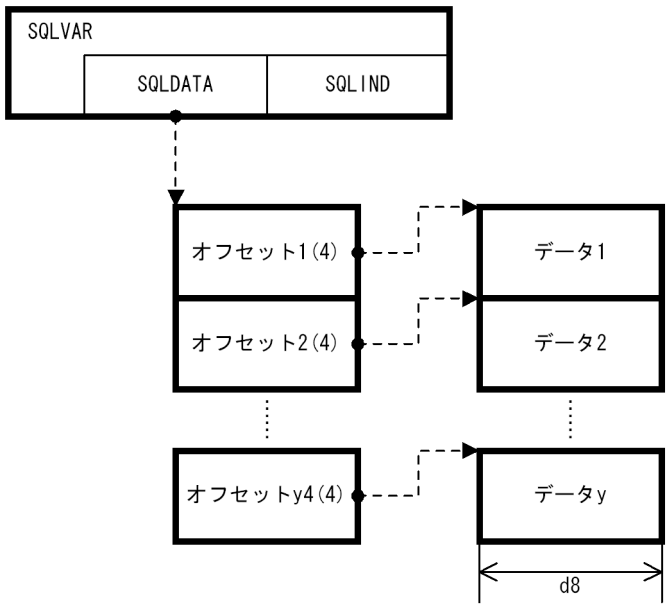


図 24–34　可変長のデータ型で配列を使った入力値（標識変数なし）の構造



（凡例）

y：標識変数の個数を示します。

y3，y4：HEADER 領域の SQLARRAY に対応する，配列の要素数を示します。

d7，d8：データの長さを示します。

p7：SQLVAR の SQLCOD に対応する，標識変数の長さを示します。

注　破線の矢印は，SQLDA からのオフセットを示します。

〔説明〕

HEADER 領域の SQLARRAY が 2 以上の場合，配列型で指定した要素数分の要素が格納されます。

データ及び標識変数は，SQLVAR の SQLCOD 及び SQLLEN，又は SQLVAR_LOB の SQLCOD 及び SQLLOBLEN に設定されたデータ型に合わせた構造になります。

データ及び標識変数は，境界調整を行った領域に格納されます。このため，各領域の長さには境界調整で発生した領域の長さも含まれます。

データ及び標識変数の長さについては，マニュアル「HiRDB UAP 開発ガイド」の「SQL 記述領域に設定するデータコードとデータの長さ」を参照してください。

データの記述については，マニュアル「HiRDB UAP 開発ガイド」の「SQL のデータ型と C 言語のデータ記述」又は「SQL のデータ型と COBOL 言語のデータ記述」を参照してください。

(c) 出力例

監査証跡に出力される SQL データの例を次に示します。

在庫表にデータを追加します。使用する埋込み変数とデータ型，及びデータの内容は次のとおりです。

列名	埋込み変数	データ型	データの内容
商品コード	xscode	char(5)	202M
商品名	xsname	char(17)	ポロシャツ
色	xcol	char(3)	赤
単価	xtanka	int	3640
数量	xgryo	int	0
数量	igryo	short	-1

実行する SQL 文は次のとおりです。

```
EXEC SQL INSERT INTO ZAIKO(SCODE,SNAME,COL,TANKA,ZSURYO)
  VALUES(:xscode,:xsname,:xcol,:xtanka,:xgryo:igryo);
```

SQL 文を実行すると，監査証跡の SQL データは次のように記録されます。

```
[HEX]    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f
00000000  00 00 00 94 00 00 00 01 00 00 00 00 00 00 00 00 [1]
00000010  53 51 4c 44 41 20 20 20 00 00 00 00 00 05 00 05
00000020  00 c4 00 01 00 04 00 00 00 00 00 60 00 00 00 00
00000030  00 c4 00 01 00 10 00 00 00 00 00 64 00 00 00 00
00000040  00 c4 00 01 00 02 00 00 00 00 00 74 00 00 00 00 [2]
00000050  00 f0 00 01 00 04 00 00 00 00 00 78 00 00 00 00
00000060  00 f1 00 01 00 04 00 00 00 00 00 80 00 00 00 7c
00000070  32 30 32 4d 83 7c 83 8d 83 56 83 83 83 63 20 20
00000080  20 20 20 20 90 d4 00 00 00 00 0e 38 ff ff 00 00 [3]
00000090  00 00 00 00
```

〔説明〕

1. HEADER 領域が格納されます。
2. SQLDA が格納されます。
3. 入力データ領域が格納されます。

   SQLVAR 又は SQLVAR_LOB の個数は，SQLDA の SQLD（下線部）で分かります。この例の場合，SQLVAR 又は SQLVAR_LOB に対応するデータは 5 個あり，それぞれ次のように格納されています。

   - 1 番目の SQLVAR から，データは CHAR 型で長さが 4 バイト，標識変数なしと分かります。データは，SQLDA から 16 進数で 60h バイト進んだ領域に 4 バイト格納されます（32 30 32 4d）。
   - 2 番目の SQLVAR から，データは CHAR 型で長さが 16 バイト，標識変数なしと分かります。データは，SQLDA から 16 進数で 64h バイト進んだ領域に 16 バイト格納されます（83 7c 83 8d 83 56 83 83 83 63 20 20 20 20 20 20）。
   - 3 番目の SQLVAR から，データは CHAR 型で長さが 2 バイト，標識変数なしと分かります。データは，SQLDA から 16 進数で 74h バイト進んだ領域に 2 バイト格納されます（90 d4）。
   - 4 番目の SQLVAR から，データは INTEGER 型で長さが 4 バイト，標識変数なしと分かります。データは，SQLDA から 16 進数で 78h バイト進んだ領域に 4 バイト格納されます（00 00 0e 38）。
   - 5 番目の SQLVAR から，データは INTEGER 型で長さが 4 バイト，標識変数ありと分かります。データは，SQLDA から 16 進数で 7ch バイト進んだ領域に 2 バイト格納されます（ff ff）。

24.9　監査証跡の絞り込み

監査証跡の絞り込みをすると，特定の監査証跡だけを取得できます。

監査証跡の絞り込みは，定義系 SQL の CREATE AUDIT で条件を定義し，必要に応じて DROP AUDIT で削除できます。

24.9.1　選択項目

監査証跡の絞り込みの条件に指定できる選択項目を次の表に示します。

表 24–30　監査証跡の絞り込みの条件に指定できる選択項目

選択項目	指定	内容	該当する監査証跡のレコード項目
操作種別	必要	特定の操作種別で絞り込んで，監査証跡を取得する場合に指定します。	• イベントタイプ • イベントサブタイプ
証跡種別	必要	証跡種別で絞り込んで監査証跡を取得する場合に指定します。	• 監査証跡種別
イベント成否	必要	イベントの成否で絞り込んで，監査証跡を取得する場合に指定します。	• イベント成否
実行者※1	任意	特定のユーザが実行したイベントだけを絞り込んで監査証跡を取得する場合に指定します。	• イベント実行者
オブジェクト※2	任意	特定のイベントの対象となったオブジェクトを絞り込んで，監査証跡を取得する場合に指定します。	• オブジェクトの所有者 • オブジェクト名称 • オブジェクト種別

注※1

実行者を指定する場合，次の規則があります。

- 存在しない実行者でも監査対象イベントの定義を指定できます。
- 監査対象イベントの定義に指定した実行者が削除されても，監査対象イベントの定義は変更しません。

注※2

オブジェクトを指定する場合，次の規則があります。

- 存在しないオブジェクト名でも監査対象イベントの定義を指定できます（CREATE 系でオブジェクトが作成された場合の証跡も絞り込めるようにするため）。
- 監査対象イベントの定義に指定したオブジェクトが，他 SQL によって削除されたり，名称変更などの属性変更があったとしても，監査対象イベントの定義は変更できません。例を次に示します。

  例 1：表 T1 に監査対象イベントの定義を指定していて，表名称を T1 から T2 に変更した場合，監査対象イベントの定義は表 T1 を指定したままとなります。

例 2：表 T1 に監査対象イベントの定義を指定していて，DROP TABLE で表 T1 が削除された場合，監査対象イベントの定義はそのまま残ります。

- 権限チェック証跡の場合，権限チェックの対象オブジェクトが，イベントの対象オブジェクトと一致した場合だけ証跡を取得します。オブジェクトによる絞り込みをして監査証跡を取得する場合は，イベント終了時の監査証跡を取得することをお勧めします。オブジェクトによって絞り込んだ場合の権限チェックの出力有無を次の表に示します。

表 24–31　オブジェクトによって絞り込んだ場合の権限チェックの出力有無

使用した権限	オブジェクトの有無	絞り込めるオブジェクト
DBA	なし	なし
SCH	なし	なし
CNT	なし	なし
RDA	あり（RDA）	左記のオブジェクト
SEL	あり（LST，TBL，VIW）	左記のオブジェクト
INS	あり（TBL，VIW）	左記のオブジェクト
DEL	あり（TBL，VIW）	左記のオブジェクト
UPD	あり（TBL，VIW）	左記のオブジェクト
AUD	なし	なし※
SYS	あり（AUF，TBL）	–
OWN	あり（FNC，IDX，PRC，SCH，SEQ，TBL，TRG，TYP，VIW）	左記のオブジェクト
SCO	あり（FNC，IDX，PRC，SEQ，TBL，TRG，TYP，VIW）	左記のオブジェクト

（凡例）

–：監査機能実行中であれば，監査対象イベントの定義に関係なく無条件に監査証跡を出力します。

注

使用した権限，及びオブジェクトの有無の記号の意味については，表「監査証跡表の列構成」の USED_PRIVILEGE 及び OBJECT_TYPE を参照してください。

注※

監査人セキュリティイベントの場合は，セキュリティ監査機能が実行中であれば，監査対象イベントの定義に関係なく，無条件に監査証跡を出力します。

絞り込み対象のオブジェクトにディクショナリ表を指定した場合，オブジェクトの種別，認可識別子，及び表識別子は次の表のように指定してください。

表 24–32　ディクショナリ表を指定した場合のオブジェクトの種別，認可識別子，及び表識別子

操作種別	オブジェクト種別	認可識別子	表識別子
オブジェクト操作イベント	VIEW	MASTER	システムが使用するディクショナリ表を除いたディクショナリ表の表識別子
ユティリティ操作イベント	TABLE	省略※	すべてのディクショナリ表の表識別子

注※

省略した場合でも，ディクショナリ表 SQL_AUDITS のオブジェクトの所有者の列には，'(Data dictionary)'が格納されます。

24.9.2　選択できる単位

選択できる単位は次のとおりです。

- 1 回の CREATE AUDIT 文で，選択項目の操作種別，証跡種別，イベント成否，実行者，及びオブジェクトを組み合わせて，監査対象イベントの定義を一つ作成できます。組み合わせの可否を次の表に示します。

  表 24–33　1 回の CREATE AUDIT 文での選択項目の組み合わせ可否一覧

選択項目，又は同時に指定できる選択項目	操作種別	証跡種別	イベント成否	実行者	オブジェクト
操作種別	−	○	○	○	○
証跡種別	○	−	○	○	○
イベント成否	○	○	−	○	○
実行者	○	○	○	−	×
オブジェクト	○	○	○	×	−

（凡例）

○：組み合わせができます。

×：組み合わせができません。

−：同時指定ができません。

- 1 回の CREATE AUDIT では，操作種別，証跡種別，イベント成否，実行者，及びオブジェクトの各項目で指定できるのは一つだけです。したがって，操作種別，証跡種別，イベント成否，実行者，及びオブジェクトの各項目で複数の値を指定する場合は，複数回の CREATE AUDIT を実行してください。複数回，監査対象イベントの定義を行った場合は，監査対象イベントの定義同士が OR 条件となり，監査対象イベントの定義のどれかに該当した監査証跡を取得します。

オブジェクト操作イベントの対象オブジェクトが表"USER1"."T1"の場合の証跡と，オブジェクト定義イベントのイベント終了証跡を監査対象とする場合は，次のように定義します。

```
CREATE AUDIT AUDITTYPE EVENT FOR ACCESS ON TABLE “USER1”.”T1”
CREATE AUDIT AUDITTYPE EVENT FOR DEFINITION
```

上記の監査対象イベントの定義を行った場合，監査証跡取得の有無は次のようになります。

取得される証跡：

- 表"USER1"."T1"を検索した場合のイベント終了証跡
- CREATE 系実行時のイベント終了証跡

取得されない証跡：

- 表"USER1"."T1"以外を検索した場合の証跡
- 接続系イベントの証跡
- ユティリティ操作イベントの証跡

24.9.3　選択項目の組み合わせ

選択項目は，組み合わせによっては意味のないものがあります。意味のない組み合わせについては，実行した CREATE AUDIT がエラーになります。

例えば，セッションセキュリティイベントの CONNECT に対して，オブジェクト表"USER1"."T1"を指定した場合などです。

24.9.4　セキュリティ監査情報用バッファ

セキュリティ監査機能を使用する場合，セキュリティ監査情報用バッファを使用します。セキュリティ監査情報用バッファには，監査対象イベントを実行者，又はオブジェクトごとに一つのエントリとして設定します。次に，定義例を示します。この例では，"USER1"に関するエントリ，"USER2"."T1"に関するエントリ，"USER2"に関するエントリの 3 エントリを定義しています。

```
CREATE AUDIT AUDITTYPE EVENT FOR ACCESS BY AUTHORIZATION ”USER1”
CREATE AUDIT AUDITTYPE EVENT FOR DEFINITION BY AUTHORIZATION ”USER1”
CREATE AUDIT AUDITTYPE EVENT FOR SESSION BY AUTHORIZATION ”USER1”
CREATE AUDIT AUDITTYPE EVENT FOR ACCESS ON TABLE ”USER2”.”T1”
CREATE AUDIT AUDITTYPE EVENT FOR PRIVILEGE ON TABLE ”USER2”.”T1”
CREATE AUDIT AUDITTYPE EVENT FOR DEFINITION BY AUTHORIZATION ”USER2”
```

上記の監査対象イベントを定義した場合，監査定義バッファのエントリ数は 3 となります。

- "USER1"に関するエントリ

オブジェクト操作イベント，オブジェクト定義イベント，セッションセキュリティイベント

- "USER2"."T1"に関するエントリ
  オブジェクト操作イベント，権限管理イベント
- "USER2"に関するエントリ
  オブジェクト定義イベント

また，セキュリティ監査情報用バッファ用共用メモリの見積もりも必要です。セキュリティ監査情報用バッファ用共用メモリの見積もりには，ユーザが指定値を見積もり，システム定義の pd_audit_def_buffer_size オペランドに指定する方法と，システムが自動的に計算する方法（pd_audit_def_buffer_size オペランドを省略）の二種類があります。システムが自動的に計算する方法では，システムが余裕値を加算してメモリを確保します。メモリサイズは，セキュリティ監査情報用バッファのエントリ数で決まるため，既に定義されているセキュリティ監査情報用バッファのエントリ数に 100 を加算するか，又は 1.2 倍し，どちらか大きいほうのエントリ数分確保します。余裕値を次に示します。

既に絞り込みの監査対象として定義しているオブジェクトのエントリ数	条件	余裕値
0	なし	セキュリティ監査情報用バッファ 100 エントリ分
1 以上	N ＋ 100 ＞ N×1.2 となる場合	セキュリティ監査情報用バッファ 100 エントリ分
	N ＋ 100≦N×1.2 となる場合	セキュリティ監査情報用バッファ N×0.2 エントリ数分

（凡例）

N：既に定義しているセキュリティ監査情報用バッファのエントリ数

セキュリティ監査情報用バッファ作成時に，メモリ不足のために必要なサイズが確保できない場合は，次の表のように対処する必要があります。

注意事項

セキュリティ監査情報用バッファに，すべての監査対象イベントの定義情報が格納されない場合，ディクショナリ表へのアクセスが発生するため，処理が遅くなります。

表 24–34　セキュリティ監査情報用バッファ作成時の HiRDB の動作と対処方法（HiRDB 開始時）

pd_audit_def_buffer_size オペランドの指定	共用メモリの確保	HiRDB の動作	対処
あり	失敗	開始しません。このとき，KFPD00031-E メッセージが出力されます。	次のどれかの対処をしてください。 • OS の共用メモリを増やす • OS の共用メモリの空き領域を作る • pd_audit_def_buffer_size オペランドの指定値を小さくする

pd_audit_def_buffer_size オペランドの指定	共用メモリの確保	HiRDB の動作	対処
	成功	開始します。ただし，セキュリティ監査情報用バッファに，すべての監査対象イベントの定義情報が格納されない場合，KFPD00032-W メッセージが出力されます。	性能が劣化することがあるため，pd_audit_def_buffer_size オペランドの値を再度見積もってください。
なし	失敗	開始しますが，セキュリティ監査情報用バッファは作成されません。このとき，KFPD00032-W メッセージが出力されます。	性能が劣化することがあるため，次のどちらかの対処をしてください。 • OS の共用メモリを増やす • OS の共用メモリの空き領域を作る 上記の対処ができない場合は，pd_audit_def_buffer_size オペランドの値に，システムが自動的に計算する値よりも小さい値を指定してください。
	成功	開始します。	対処する必要はありません。

表 24–35　セキュリティ監査情報用バッファ作成時の HiRDB の動作と対処方法（HiRDB 稼働中）

pd_audit_def_buffer_size オペランドの指定	セキュリティ監査情報用バッファの監査対象イベントの定義情報のあふれ	HiRDB の動作	対処
あり	あり	セキュリティ監査情報用バッファに，格納できる分だけ監査対象イベントの定義情報を格納して処理を続行します。このとき，KFPD00032-W メッセージを出力します。	KFPD00032-W メッセージに従って pd_audit_def_buffer_size オペランドの値を再度見積もってください。対処しないと，性能が劣化することがあります。
	なし	セキュリティ監査情報用バッファに，すべての監査対象イベントの定義情報を格納して処理を続行します。	対処する必要はありません。
なし	あり	セキュリティ監査情報用バッファに，格納できる分だけ監査対象イベントの定義情報を格納して処理を続行します。このとき，KFPD00032-W メッセージを出力します。	HiRDB を再開始してください。システムがサイズを再計算して，セキュリティ監査情報用バッファを作成します。再開始したときに KFPD00032-W メッセージが出力された場合，次のどちらかの対処をしてください。 • OS の共用メモリを増やす • OS の共用メモリの空き領域を作る 上記の対処ができない場合は，pd_audit_def_buffer_size オペランドの値に，システムが自動的に計算する値よりも小さい値を指定してください。

pd_audit_def_buffer_size オペランドの指定	セキュリティ監査情報用バッファの監査対象イベントの定義情報のあふれ	HiRDB の動作	対処
	なし	セキュリティ監査情報用バッファに，すべての監査対象イベントの定義情報を格納して処理を続行します。	対処は必要ありません。

pd_audit_def_buffer_size オペランドを省略した場合，指定値はシステムが自動的に計算しますが，HiRDB の開始から終了までの間に監査対象イベントの定義が増えたときは，次回の HiRDB の開始でのセキュリティ監査情報用バッファのサイズが増えます。したがって，HiRDB の開始，終了を繰り返すと，セキュリティ監査情報用バッファのサイズが増えることがあります。

24.9.5 セキュリティ監査情報用バッファに関するエラーが発生した場合の HiRDB の動作

セキュリティ監査情報用バッファに関するエラーが発生した場合の HiRDB の動作について説明します。

(1) HiRDB 開始時

セキュリティ監査情報用バッファは HiRDB 開始時に作成されますが，このときエラーが発生することがあります。エラーが発生した場合，セキュリティ監査情報用バッファをシステムが自動計算する場合と，ユーザが指定値を見積もる場合とで HiRDB の動作は変わります。

- システムが自動計算する場合

  メッセージを出力し，セキュリティ監査情報用バッファはサイズ 0 で開始します（セキュリティ監査情報用バッファを作成しません）。ただし，ディクショナリアクセスエラー（ロールバック要）の場合は次のようになります。

  HiRDB/シングルサーバのとき：

  ディスク障害などは，サイズ 0 で開始しても，開始後に監査対象イベントの定義情報が取得できないで続行できなくなる可能性が高いため，メッセージを出力してすぐに異常終了します（ユニットダウンします）。

  HiRDB/パラレルサーバのとき：

  ディクショナリサーバが開始していない可能性があるため，セキュリティ監査情報用バッファの作成処理を一定回数リトライします。一定回数分リトライしてもエラーが解消されない場合，フロントエンドサーバを SUSPEND 状態にします。

- ユーザが指定値を見積もる場合

メッセージを出力して処理を続行します。ただし，指定したサイズのセキュリティ監査情報用バッファ用共用メモリが確保できない場合，メッセージを出力してすぐに異常終了します（ユニットダウンします）。

HiRDB 開始時のエラー要因と HiRDB の処理を次の表に示します。

表 24–36　HiRDB 開始時のエラー要因と HiRDB の処理

エラー要因		HiRDB の処理	
		pd_audit_def_buffer_size オペランドを省略	pd_audit_def_buffer_size オペランドを指定
領域確保エラー	バッファ用共用メモリ	サイズ 0 で開始	開始できません
	ディクショナリ検索用プロセス固有メモリ	サイズ 0 で開始	共用メモリ確保後，処理を続行
通信障害		サイズ 0 で開始	共用メモリ確保後，処理を続行
ディクショナリアクセスエラー	ロールバック不要	サイズ 0 で開始	共用メモリ確保後，処理を続行
	ロールバック要	開始できません※	共用メモリ確保後，処理を続行

注※

HiRDB/パラレルサーバの場合，一定回数の再開始をしてもエラーが解消されないときは，フロントエンドサーバを SUSPEND 状態にします。

HiRDB/シングルサーバの場合はすぐに異常終了します。

(2)　HiRDB 稼働中

監査対象イベントの定義情報の判定中にエラーが発生した場合は，監査対象イベントの定義に関係なく，該当する監査証跡を出力します。

SQL 実行時にエラーが発生した場合も監査証跡を出力しますが，監査証跡出力時の監査対象イベントの定義情報取得時にもエラーが発生することがあります。エラーの組み合わせと，設定される SQL コード及びロールバック要否を次の表に示します。なお，監査証跡出力時にエラーが発生した場合は，そのエラーを無視して処理を続行します。

表 24–37　エラーの組み合わせと，設定される SQL コード及びロールバック要否

監査対象イベントの定義取得前の状態	監査対象イベントの定義取得時の状態	設定される SQL コード	ロールバック要否
正常	正常	0	不要
	ロールバック要エラー	監査対象イベントの定義取得時の SQL コード	要
	ロールバック不要エラー	監査対象イベントの定義取得前の SQL コード	不要

監査対象イベントの定義取得前の状態	監査対象イベントの定義取得時の状態	設定される SQL コード	ロールバック要否
ロールバック要エラー	正常	監査対象イベントの定義取得前の SQL コード	要
	ロールバック要エラー	監査対象イベントの定義取得前の SQL コード	要
	ロールバック不要エラー	監査対象イベントの定義取得前の SQL コード	要
ロールバック不要エラー	正常	監査対象イベントの定義取得前の SQL コード	不要
	ロールバック要エラー	監査対象イベントの定義取得前の SQL コード	要
	ロールバック不要エラー	監査対象イベントの定義取得前の SQL コード	不要

24.9.6　セキュリティ監査情報用バッファの状態遷移

セキュリティ監査情報用バッファは，イベントが発生すると，無効な状態になったり，有効な状態になったりします。イベントが発生した場合のセキュリティ監査情報用バッファの状態遷移を次に示します。

イベント	セキュリティ監査情報用バッファの状態			
	初期状態 (HiRDB 開始前)	無効状態 (情報未設定)	有効状態 (情報設定)	無効状態 (古い情報が残っている)
	1	2	3	4
HiRDB の開始処理完了	→2	−	−	−
セキュリティ監査情報用バッファへのアクセス	−	→3	→3	→3
監査対象イベントの定義変更時（CREATE AUDIT，DROP AUDIT 実行時）	−	→2	→4	→4

（凡例）

→n：該当するイベントが発生すると，番号 n の状態になることを示しています。

−：該当しません。

24.10　監査証跡ファイルに障害が発生したときの対処方法

24.10.1　監査証跡ファイルに障害が発生した場合

障害が発生した監査証跡ファイルは閉塞状態になります。閉塞状態の監査証跡ファイルは，削除しないかぎり使用できない状態のままです。次に示す手順で対処してください。

〈手順〉

1. KFPS05704-E メッセージで閉塞状態の監査証跡ファイルを確認してください。

   pdls -d aud コマンドで障害が発生した世代を確認できます。HiRDB が稼働している場合は，監査人が pdls -d aud コマンドを実行してください。

2. pdaudrm コマンドで閉塞状態の監査証跡ファイルを削除してください。データロード待ちのファイルに対しては-f オプションを指定して強制的に削除してください。

閉塞状態のファイルはデータロードできません。したがって，データロードしていない場合はそのファイル中の監査証跡はすべて失われます。

24.10.2　監査証跡ファイル用の HiRDB ファイルシステム領域に障害が発生した場合（全監査証跡ファイルが閉塞状態になった場合）

監査証跡ファイル用の HiRDB ファイルシステム領域に障害が発生した場合など，全監査証跡ファイルが閉塞状態になったときの対処方法を説明します。

pd_aud_no_standby_file_opr オペランドの指定値によって，HiRDB の処理が異なります。

(1)　down を指定した場合

HiRDB が稼働中の場合は，HiRDB（HiRDB/パラレルサーバの場合はユニット）を強制終了します。

HiRDB が開始処理中の場合は，監査証跡の出力を中止してから，HiRDB の開始処理を完了します。

(2)　forcewrite（省略値）を指定した場合

監査証跡の出力を中止します。この場合，HiRDB は稼働したままです。

HiRDB 管理者は次に示す手順で対処してください。

〈手順〉

1. HiRDB が稼働中の場合は pdstop コマンドで HiRDB を正常終了します。正常終了できない場合は pdstop -f コマンドで HiRDB を強制終了してください。

2. pdfmkfs コマンドで監査証跡ファイル用の HiRDB ファイルシステム領域を再作成します。ディスク障害などでこの HiRDB ファイルシステム領域が使用できない場合は，ほかのディスクに監査証跡ファイル用の HiRDB ファイルシステム領域を作成してください。
3. 監査証跡ファイル用の HiRDB ファイルシステム領域を変更した場合は，pd_aud_file_name オペランドの指定を変更してください。
4. 監査証跡ファイル用の HiRDB ファイルシステム領域のバックアップがある場合は，pdfrstr コマンドで HiRDB ファイルシステム領域を回復してください。
5. pdstart コマンドで HiRDB を開始します。

24.10.3 スワップ先にできる監査証跡ファイルがないため HiRDB が強制終了した場合

pd_aud_no_standby_file_opr オペランドに down を指定した場合，ファイル容量の満杯時，又は現用ファイルの障害時にスワップ先にできるファイルが残り一つ以下になると，HiRDB（HiRDB/パラレルサーバの場合はユニット）を強制終了します。HiRDB 管理者，及び監査人は次に示す手順で対処してください。

(1) 監査証跡の自動データロード機能を使用している場合（pd_aud_auto_loading = Y を指定している場合）

(a) HiRDB/シングルサーバの場合

- KFPS05705-E（number=1）メッセージが表示された場合
  次に示す手順で対処してください。

〈手順〉

1. pdstart コマンドで HiRDB を再開始してください。
2. pdaudatld -b コマンドで監査証跡表の自動データロード機能を再開始してください。データロードが実行されると KFPS05714-I，及び KFPS05715-I メッセージを表示します。

- KFPS05705-E（number=0）メッセージが表示された場合
  次のどちらかに該当するときは〈手順 1〉で対処してください。該当しないときは〈手順 2〉で対処してください。
  - pd_aud_max_generation_num ＝ 200 のとき
  - pdfstatfs コマンドを実行して，監査証跡ファイルを格納している HiRDB ファイルシステム領域に空き（remain user area capacity，remain file count）がないとき

〈手順 1〉

1. pdstart コマンドで HiRDB を再開始してください。

このときセキュリティ監査機能は停止した状態で，KFPS05724-E，及び KFPS05725-W メッセージを表示するときがあります。

2. pdload コマンドで，データロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。
3. pdaudbegin コマンドで監査証跡の取得を開始してください。
4. pdaudatld -b コマンドで監査証跡表の自動データロード機能を再開始してください。

〈手順 2〉

1. pd_aud_max_generation_num オペランドの指定値を増やしてください。スワップ先にできる監査証跡ファイルが自動で追加されます。
2. pdstart コマンドで HiRDB を再開始してください。
3. pdaudatld -b コマンドで監査証跡表の自動データロード機能を再開始してください。データロードが実行されると KFPS05714-I，及び KFPS05715-I メッセージを表示します。

上記の手順で回復した後，監査証跡ファイルの容量を削減したい場合は HiRDB を一度正常停止して pd_aud_max_generation_num オペランドの指定値を元に戻してください。また，自動で追加された監査証跡ファイル（pd_aud_max_generation_num オペランドの指定値よりも大きい世代番号の監査証跡ファイル）を削除してください。その後，HiRDB を開始してください。

監査証跡ファイルの削除方法は「監査証跡ファイルの容量変更」を参照してください。監査証跡ファイルは，データロード済み状態にしてから削除する必要があります。pdls -d aud コマンドの実行結果で，削除したい監査証跡ファイルの状態が"s--"（データロード済み）以外の場合は，pdaudswap コマンドを実行して"s--"（データロード済み）にしてから削除してください。データロード済み状態ではない監査証跡ファイルを削除すると，監査証跡が失われます。

(b)　HiRDB/パラレルサーバの場合

- KFPS05705-E（number=1）メッセージが表示された場合

  次に示す手順で対処してください。

〈手順〉

1. KFPS05705-E（number=1）メッセージが表示されたユニット名称を控えてください。
2. 1.のユニット上で pdstart -q コマンドを実行してユニットを再開始してください。
3. pdaudatld -b コマンドで監査証跡表の自動データロード機能を再開始してください。データロードが実行されると KFPS05714-I，及び KFPS05715-I メッセージを表示します。

- KFPS05705-E（number=0）メッセージが表示された場合

  次のどちらかに該当するときは〈手順 1〉で対処してください。該当しないときは〈手順 2〉で対処してください。

  - pd_aud_max_generation_num ＝ 200 のとき

- pdfstatfs コマンドを実行して，監査証跡ファイルを格納している HiRDB ファイルシステム領域に空き（remain user area capacity，remain file count）がないとき

〈手順 1〉

1. KFPS05705-E（number=0）メッセージが表示されたユニット名称を控えてください。
2. 1.のユニット上で pdstart -q コマンドを実行してユニットを再開始してください。

   このときセキュリティ監査機能は停止した状態で，KFPS05724-E，及び KFPS05725-W メッセージを表示するときがあります。
3. pdload コマンドで，1.のユニットのデータロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。
4. pdaudbegin -u コマンドで 1.のユニットの監査証跡の取得を開始してください。
5. pdaudatld -b コマンドで監査証跡表の自動データロード機能を再開始してください。

〈手順 2〉

1. KFPS05705-E（number=0）メッセージが表示されたユニット名称を控えてください。
2. 1.のユニットのユニット定義で pd_aud_max_generation_num オペランドの指定値を増やしてください。スワップ先にできる監査証跡ファイルが自動で追加されます。
3. 1.のユニット上で pdstart -q コマンドを実行してユニットを再開始してください。
4. pdaudatld -b コマンドで監査証跡表の自動データロード機能を再開始してください。データロードが実行されると KFPS05714-I，及び KFPS05715-I メッセージを表示します。

上記の手順で回復した後，監査証跡ファイルの容量を削減したい場合は HiRDB 又はユニットを一度正常停止して pd_aud_max_generation_num オペランドの指定値を元に戻してください。また，自動で追加された監査証跡ファイル（pd_aud_max_generation_num オペランドの指定値よりも大きい世代番号の監査証跡ファイル）を削除してください。その後，HiRDB を開始してください。

監査証跡ファイルの削除方法は「監査証跡ファイルの容量変更」を参照してください。監査証跡ファイルは，データロード済み状態にしてから削除する必要があります。pdls -d aud コマンド結果で，削除したい監査証跡ファイルの状態が"s--"（データロード済み）以外の場合は，pdaudswap -u コマンドを実行して"s--"（データロード済み）にしてから削除してください。データロード済み状態ではない監査証跡ファイルを削除すると，監査証跡が失われます。

(2) 監査証跡の自動データロード機能を使用していない場合（pd_aud_auto_loading = Y を指定していない場合）

(a) HiRDB/シングルサーバの場合

- KFPS05705-E（number=1）メッセージが表示された場合

  次に示す手順で対処してください。

〈手順〉

1. pdstart コマンドで HiRDB を再開始してください。
2. pdload コマンドで，データロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。

- KFPS05705-E（number=0）メッセージが表示された場合

  次のどちらかに該当するときは〈手順 1〉で対処してください。該当しないときは〈手順 2〉で対処してください。

  - pd_aud_max_generation_num ＝ 200 のとき
  - pdfstatfs コマンドを実行して，監査証跡ファイルを格納している HiRDB ファイルシステム領域に空き（remain user area capacity，remain file count）がないとき

〈手順 1〉

1. pdstart コマンドで HiRDB を再開始してください。

   このときセキュリティ監査機能は停止した状態で，KFPS05724-E，及び KFPS05725-W メッセージを表示するときがあります。
2. pdload コマンドで，データロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。
3. pdaudbegin コマンドで監査証跡の取得を開始してください。

〈手順 2〉

1. pd_aud_max_generation_num オペランドの指定値を増やしてください。スワップ先にできる監査証跡ファイルが自動で追加されます。
2. pdstart コマンドで HiRDB を再開始してください。
3. pdload コマンドで，データロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。

上記の手順で回復した後，監査証跡ファイルの容量を削減したいときは HiRDB を一度正常停止して pd_aud_max_generation_num オペランドの指定値を元に戻してください。また，自動で追加された監査証跡ファイル（pd_aud_max_generation_num オペランドの指定値よりも大きい世代番号の監査証跡ファイル）を削除してください。その後，HiRDB を開始してください。

監査証跡ファイルの削除方法は「監査証跡ファイルの容量変更」を参照してください。監査証跡ファイルは，データロード済み状態にしてから削除する必要があります。pdls -d aud コマンド結果で，削除したい監査証跡ファイルの状態が"s--"（データロード済み）以外のときは，pdaudswap コマンド及び pdload コマンドを実行して"s--"（データロード済み）にしてから削除してください。データロード済み状態ではない監査証跡ファイルを削除すると，監査証跡が失われます。

(b)　HiRDB/パラレルサーバの場合

- KFPS05705-E（number=1）メッセージが表示された場合
  次に示す手順で対処してください。

〈手順〉

1. KFPS05705-E（number=1）メッセージが表示されたユニット名称を控えてください。
2. 1.のユニット上で pdstart -q コマンドを実行してユニットを再開始してください。
3. pdload コマンドで，1.のユニットのデータロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。監査証跡表の自動データロード機能を再開始してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。

- KFPS05705-E（number=0）メッセージが表示された場合
  次のどちらかに該当するときは〈手順 1〉で対処してください。該当しないときは〈手順 2〉で対処してください。
  - pd_aud_max_generation_num ＝ 200 のとき
  - pdfstatfs コマンドを実行して，監査証跡ファイルを格納している HiRDB ファイルシステム領域に空き（remain user area capacity，remain file count）がないとき

〈手順 1〉

1. KFPS05705-E（number=0）メッセージが表示されたユニット名称を控えてください。
2. 1.のユニット上で pdstart -q コマンドを実行してユニットを再開始してください。
   このときセキュリティ監査機能は停止した状態で，KFPS05724-E，及び KFPS05725-W メッセージを表示するときがあります。
3. pdload コマンドで，1.のユニットのデータロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。
4. pdaudbegin -u コマンドで 1.のユニットの監査証跡の取得を開始してください。

〈手順 2〉

1. KFPS05705-E（number=0）メッセージが表示されたユニット名称を控えてください。
2. 1.のユニットのユニット定義で pd_aud_max_generation_num オペランドの指定値を増やしてください。スワップ先にできる監査証跡ファイルが自動で追加されます。
3. 1.のユニット上で pdstart -q コマンドを実行してユニットを再開始してください。
4. pdload コマンドで，1.のユニットのデータロード待ちのファイルをデータロードしてください。データロードの方法は「監査証跡表へのデータ登録」を参照してください。データロードが実行されると KFPL00702-I，及び KFPL00704-I メッセージを表示します。

上記の手順で回復した後，監査証跡ファイルの容量を削減したいときは HiRDB 又はユニットを一度正常停止して pd_aud_max_generation_num オペランドの指定値を元に戻してください。また，自動で追加された監査証跡ファイル（pd_aud_max_generation_num オペランドの指定値よりも大きい世代番号の監査証跡ファイル）を削除してください。その後，HiRDB を開始してください。

監査証跡ファイルの削除方法は「監査証跡ファイルの容量変更」を参照してください。監査証跡ファイルは，データロード済み状態にしてから削除する必要があります。pdls -d aud コマンド結果で，削除したい監査証跡ファイルの状態が"s--"（データロード済み）以外のときは，pdaudswap -u コマンド及び pdload コマンドを実行して"s--"（データロード済み）にしてから削除してください。データロード済み状態ではない監査証跡ファイルを削除すると，監査証跡が失われます。

24.10.4　非同期出力用バッファのすべての面がフラッシュ待ちになった場合

非同期出力用バッファフラッシュ契機が頻繁に発生し，非同期出力用バッファのすべての面がフラッシュ待ちになると，トランザクションの実行時間が長くなったり，pd_aud_no_standby_file_opr オペランドの指定によっては HiRDB（HiRDB/パラレルサーバの場合はユニット）が強制終了したりします。

非同期出力用バッファのすべての面がフラッシュ待ちになった場合，KFPS05723-W メッセージ（pd_aud_no_standby_file_opr オペランドに forcewrite 指定時）又は KFPS05722-E メッセージ（pd_aud_no_standby_file_opr オペランドに down 指定時）が出力されます。HiRDB 管理者は次に示す対処をしてください。

- 単位時間当たりの監査証跡の出力件数と比較して，非同期出力用バッファのサイズと面数を見直してください。

  非同期出力用バッファのサイズと面数が少ない場合，HiRDB システム定義の pd_aud_async_buff_size オペランド及び pd_aud_async_buff_count オペランドの指定値を大きくしてください。
- マシンやネットワークの負荷を確認してください。

  高い負荷が掛かっている場合，負荷を下げてください。必要に応じて HiRDB（HiRDB/パラレルサーバの場合はユニット）を再開始してください。
- 非同期出力用バッファの監査証跡を監査証跡ファイルに出力するときのネットワークに障害が発生していないか確認してください。

  障害が発生している場合，ネットワーク障害を回復してください。必要に応じて HiRDB（HiRDB/パラレルサーバの場合はユニット）を再開始してください。

24.11　自動データロード機能適用中に障害が発生したときの対処方法

24.11.1　自動データロード実行中に障害が発生した場合

自動データロード実行中に障害が発生した場合の HiRDB の動作と，HiRDB 管理者又は監査人の取る対処方法を次の表に示します。

表 24–38　自動データロード実行中の障害内容と対処方法

障害内容	障害原因	HiRDB の動作	機能の継続	HiRDB 管理者又は監査人の対処方法
監査証跡表が存在しない場合	監査証跡表を作成する前に，監査証跡ファイルのスワップが発生しました。	KFPL15029-E メッセージを出力し，データロードを中止します。	×	【管理者】 pdmod コマンドを実行し，監査証跡表を作成してください。
pdload の最大同時実行数が超過した場合	pd_utl_exec_mode に 0 を指定している場合，自動データロード以外で実行している pdload コマンドが 32 個を超えました。	実行中の pdload コマンドが終了した後に，データロードを開始します。	○	【管理者】 pdload コマンドの最大同時実行数を見直し，必要であれば pd_utl_exec_mode の値を 1 に変更してください。
pdload の最大同時接続数が超過した場合	pdload コマンドの最大同時接続数が，pd_max_users で指定している値を超えました。	KFPA11932-E メッセージを出力します。データロードはいったん中止しますが，時間を置いて再実行します。	○	【管理者】 pdload コマンドの最大同時接続数を見直し，必要であれば pd_max_users の値を大きくしてください。
排他確保エラーが発生した場合	監査証跡表を NOWAIT 検索以外の方法で参照している UAP が存在しています。	KFPA11770-I メッセージを出力し，データロードを中止します。	○	【監査人】 監査証跡表の参照に時間が必要な場合は，pdaudatld -t コマンドで一時的に自動データロード機能を停止するか，NOWAIT 検索による参照を検討してください。
RD エリアへアクセスできない場合	監査証跡表を格納した RD エリアが閉塞又はクローズしています。	KFPL23500-E メッセージを出力し，データロードを中止します。	×	【管理者】 監査証跡表が格納されている RD エリアの状態をオープンにしてください。
監査証跡ファイルの読み込みエラーが発生した場合	監査証跡ファイル（HiRDB ファイル）の内容を読み込みできません。	KFPS05704-E メッセージを出力し，データロードをロールバックして処理を中止します。	×	【管理者】 メッセージの内容を参照して，障害を取り除いてください。
監査証跡ファイルの書き込みエラーが発生した場合※3	監査証跡ファイルのステータス変更（データロード済み状態への変更）に失敗しました。	KFPS05704-E メッセージを出力し，データロードをロールバックして処理を中止します。	×	【監査人】 次の手順で対処してください。 1. イベントログを参照し，自動データロード中に異常終了した

障害内容	障害原因	HiRDB の動作	機能の継続	HiRDB 管理者又は監査人の対処方法
				監査証跡ファイルを特定します※1。 2. 1.で特定したファイル名を指定し，pdload コマンドを手動で実行します※2。 3. pdaudatld -b コマンドを実行して，自動データロード機能を再開始します。
データロード待ち状態の監査証跡ファイルが存在しない場合	監査人が監査証跡表へのデータロードを手動で行いました。	KFPL23201-W メッセージを出力します。	○	対策は不要です。
RD エリアの容量が不足している場合	監査証跡表のデータを退避又は削除しなかったため，RD エリアの容量が不足しました。	KFPA11756-E メッセージを出力し，データロードをロールバックして処理を中止します。	×	【監査人】 監査証跡表に不要なデータがある場合は，削除してください。又は，HiRDB 管理者に RD エリアの拡張を依頼してください。 【管理者】 監査人から依頼があった場合，RD エリアを拡張してください。
監査証跡表が更新できない場合	監査証跡表にデータロードできません。	障害の原因を SQL メッセージに出力し，データロードをロールバックして処理を中止します。	×	【管理者】 SQL メッセージの内容を参照して，障害を取り除いてください。
通信エラー（サーバ起動）が発生した場合	pdload からのサーバ起動ができません。サーバやユニットがダウンしているおそれがあります。	KFPL20000-E メッセージを出力し，データロードを中止します。	×	【管理者】 メッセージの内容を参照して，障害を取り除いてください。
通信エラー（データ送受信）が発生した場合	監査証跡ファイルのあるユニットから，監査証跡表のあるユニットにデータが送信できません。	KFPL20000-E メッセージを出力してアボートします。これに伴いデータロードをロールバックして，処理を中止します。	×	【管理者】 メッセージの内容を参照して，障害を取り除いてください。
プロセスがダウンした場合※3	自動データロードを行う pdload がキャンセルされたか，サーバがダウンしました。	KFPS01820-E メッセージを出力してアボートします。	×	【管理者】 自動データロードを行う pdload のプロセスに対して，pdcancel コマンドを実行しないでください。 上記の要因に当てはまらない場合は，メッセージの内容を参照して，障害を取り除いてくださ

障害内容	障害原因	HiRDB の動作	機能の継続	HiRDB 管理者又は監査人の対処方法
				い。対処できない場合は，保守員に連絡してください。

（凡例）

○：自動データロード機能は停止しません。

×：自動データロード機能は停止します。障害の原因を取り除いた後，pdaudatld -b コマンドで機能を再開始する必要があります。

【管理者】：HiRDB 管理者が行う対処方法

【監査人】：監査人が行う対処方法

注※１

ユニットダウンでイベントログが参照できない場合は，監査証跡表に対してデータベース状態解析ユティリティ（pddbst）を DBA 権限所有者に実行してもらい，その実行結果から特定してください。

注※２

ここで実行する pdload コマンドは，監査証跡ファイルの状態をデータロード済みに変更する処理を行うだけで，データロードは行いません。

注※３

監査証跡ファイルの状態を「データロード済み」に変更できなかった場合に発生する障害です。

24.11.2　自動データロード機能適用中に HiRDB で障害が発生した場合

自動データロード機能を適用中に HiRDB で障害が発生した場合の HiRDB の動作と，対処方法を次の表に示します。HiRDB 管理者は，対処方法の内容を基に，障害原因に応じた対処をしてください。

表 24–39　自動データロード機能適用中の HiRDB の障害内容と，対処方法

障害内容	障害原因	HiRDB の動作	機能の継続	HiRDB 管理者の対処方法
監査証跡ファイルにスワップが発生した時，MGR ユニットへの通信が失敗した場合	• 原因 1：MGR ユニットが停止しています。 • 原因 2：MGR ユニットと，スワップした監査証跡ファイルのある非 MGR ユニットの間で，通信障害が発生しています。	• KFPS05193-E メッセージを表示します。 ただし，次に示す通信障害の場合は，メッセージが表示されません。 ・電文送信と同時に MGR ユニットとの間で通信障害が発生した場合 ・電文送信先のプロセスにスレッドを受け付ける余裕がない場合	○	• 原因 1：pdstart -q コマンドで MGR ユニットを再開始してください。 • 原因 2：原因を調査し，MGR ユニットと非 MGR ユニット間の通信障害を解消してください。
監査証跡ファイルにスワップが発生したのに，自動データロー	MGR ユニットと，スワップした監査証跡ファイルのある非 MGR ユニットの間で，通信障害が発生しています。			原因を調査し，MGR ユニットと非 MGR ユニット間の通信障害を解消してください。

障害内容	障害原因	HiRDB の動作	機能の継続	HiRDB 管理者の対処方法
ド対象にならない場合		• 自動データロード処理は，障害が解消した後，監査証跡ファイルのスワップが発生した時に実行します。		
自動データロードで，pdload の起動に失敗した場合	• 原因 1：システム内の最大プロセス数が限界に達しています。 • 原因 2：メモリ不足が発生しているため，プロセスが生成できません。	• KFPS05717-W メッセージを表示します。 • 自動データロードは，一定時間経過後に再実行します。	○	• 原因 1：MGR ユニットが動作するサーバマシンの最大プロセス数を見直してください。 • 原因 2：MGR ユニットが動作するサーバマシンのメモリ使用量を減らしてください。
自動データロード対象のファイルが存在するのに，自動データロードが実行されない場合	• 原因 1：自動データロード対象の監査証跡ファイルがあるユニットが停止しています。 • 原因 2：MGR ユニットと自動データロード対象の非 MGR ユニットの間で，通信障害が発生しています。 • 原因 3：自動データロード機能が停止しています。 • 原因 4：HiRDB の停止処理を実行しています。	• KFPS05718-W メッセージを表示します。 • 自動データロード対象の監査証跡ファイルが別のユニットにある場合は，その監査証跡ファイルを先にデータロードします。 • 原因 1～3 で，障害が解消した場合，先行の自動データロードが完了した後に，該当する監査証跡ファイルの自動データロードを開始します。	○	• 原因 1：自動データロード対象の監査証跡ファイルがあるユニットを，pdstart -q コマンドで起動してください。 • 原因 2：原因を調査し，MGR ユニットと非 MGR ユニット間の通信障害を解消してください。 • 原因 3：監査人が pdaudatld -t コマンドで明示的に自動データロード機能を停止していなかった場合は，機能が停止した原因を調査し，対策してください。 • 原因 4：次回 HiRDB を起動した時に再度自動データロード対象になるので，対処は不要です。
ユニットダウンにより，自動データロードがエラー終了した場合	自動データロード中の監査証跡ファイル又は監査証跡表のある非 MGR ユニットが，強制停止，異常終了，又は系切り替えを行いました。	KFPS05713-E メッセージを表示します。	×	pdstart -q で非 MGR ユニットを再開始してください。
	MGR ユニットが強制停止，異常終了，又は系切り替えを行いました。	• MGR ユニットの停止と同時に，自動データロードも停止します。 • 自動データロードは，MGR ユニット再開始後に行います。	○	pdstart -q で MGR ユニットを再開始してください。

（凡例）

○：自動データロード機能は停止しません。

×：自動データロード機能は停止します。障害の原因を取り除いた後，pdaudatld -b で機能を再開始する必要があります。

MGR ユニット：システムマネジャがあるユニット

非 MGR ユニット：システムマネジャがないユニット

24.12　ほかの機能との関連

24.12.1　セキュリティ監査機能とほかの機能との関連

ここではセキュリティ監査機能とほかの機能との関連について説明します。

(1)　系切り替え機能との関連

- 切り替え先の系で監査証跡の取得を引き継ぐかどうかは切り替え元の HiRDB の終了モードによって決まります。切り替え先の系が再開始の場合は系を切り替える前の状態を引き継ぎます。切り替え先の系が正常開始の場合は pd_audit オペランドの指定に従います。
- モニタモードで系切り替え機能を実行している場合，系切り替えが発生して待機系 HiRDB（HiRDB/パラレルサーバの場合は待機系ユニット）を開始したときに監査証跡を取得します。サーバモードの場合は系切り替えが発生したときに監査証跡を取得しません。
- 障害が発生して系が切り替わった場合，HiRDB は切り替わる直前の監査証跡を正しく取得しません。

(2)　プラグインとの関連

pdplgrgst コマンドでプラグインを HiRDB に登録，削除，又はバージョンアップする場合，HiRDB は pdplgrgst コマンドの延長で次に示す SQL 文を発行します。

- プラグインが提供する関数，抽象データ型，又はインデクス型の CREATE 文
- ディクショナリ表に格納されているプラグイン定義情報の DELETE 文
- プラグインが提供する関数，抽象データ型，又はインデクス型の DROP 文

これらの SQL 文に対して監査対象イベントを設定している場合は，pdplgrgst コマンドの延長でこれらの SQL 文に関する監査証跡を取得します。

24.13　監査証跡のレコード項目（権限チェック時）

権限チェック時の監査証跡のレコード項目を次に示します。

ユーザ識別子～オブジェクト種別までのレコード項目は（その１）に，付与/削除変更した権限以降のレコード項目は（その２）に記載しています。

表中のイベントタイプ，イベントサブタイプについては，表「イベントタイプ及びイベントサブタイプの詳細」を，使用した権限については，表「監査証跡表の列構成」の USED_PRIVILEGE を参照してください。

●権限チェック時の監査証跡のレコード項目（その 1）

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
システム管理者セキュリティイベント	HiRDB の開始	pdstart	○	○	○	○	SYS	STR	○	SYS	-
	HiRDB の終了	pdstop	○	○	○	○	SYS	STP	○	SYS	-
	主監査人の登録	pdmod	○	○	○	○	SYS	MOD	○	SYS	"pdmod"
	副監査人の登録	pdmod	○	○	○	○	SYS	MDA	○	SYS	"pdmod"
	監査証跡表の作成	pdmod	○	○	○	○	SYS	MOD	○	SYS	"pdmod"
	監査証跡ファイルの削除	pdaudrm	○	○	○	○	SYS	ARM	○	SYS	-
	監査証跡の取得開始	pdaudbegin	○	○	○	○	SYS	ABG	○	SYS	"pdaudbegin"
		HiRDB の開始	○	○	○	○	SYS	ABG	○	SYS	"pdstart"
	監査証跡の取得終了	pdaudend	○	○	○	○	SYS	AEN	○	SYS	"pdaudend"
		HiRDB の終了	○	○	○	○	SYS	AEN	○	SYS	"pdstop"
	連続認証失敗アカウントロック状態の解除	pdacunlck（全 DBA 権限保持者アカウ	○	○	○	○	SYS	ULK	○	SYS	"pdacunlck"

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		ントロック時）									
	CONNECT 関連セキュリティ機能の設定値変更	CREATE CONNECTION SECURITY,DROP CONNECTION SECURITY	○	○	○	○	SYS	SPR	○	DBA	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
監査人セキュリティイベント	監査証跡表へのデータロード	pdload※5	○	○	○	○	AUD	ALD	○	AUD	"pdload"
	監査証跡ファイルのスワップ	pdaudswap	○	○	○	○	AUD	ASW	○	AUD	-
	監査対象イベントの 定義	CREATE AUDIT	○	○	○	○	AUD	CRT	○	AUD	○
	監査対象イベントの 削除	DROP AUDIT	○	○	○	○	AUD	DRP	○	AUD	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
	監査人の パスワード変更	GRANT AUDIT	○	○	○	○	AUD	GRT	○	AUD	○
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	○	○	○	○	AUD	ATB	○	AUD	"pdaudatld"
	監査証跡表の自動データロード機能の停止	pdaudatld -t	○	○	○	○	AUD	ATT	○	AUD	"pdaudatld"
	JP1/NETM/Audit 用監査ログの出力	pdaudput	○	○	○	○	AUD	APT	○	AUD	"pdaudput"
セッションセキュリティイベント	HiRDB への接続	CONNECT	○	○	○	○	SES	CNT	○	CNT	○
	ユーザの 変更	SET SESSION AUTHORIZATION	○	○	○	○	SES	ATH	○	CNT	○
権限管理イベント	権限付与	GRANT DBA	○	○	○	○	PRV	GRT	○	DBA	○
		GRANT SCHEMA	○	○	○	○	PRV	GRT	○	DBA	○
		GRANT SCHEMA OPERATION	○	○	○	○	PRV	GRT	○	OWN	○
		GRANT CONNECT	○	○	○	○	PRV	GRT	○	DBA	○
		GRANT RDAREA	○	○	○	○	PRV	GRT	○	DBA	○
		GRANT アクセス権限	○	○	○	○	PRV	GRT	○	OWN	○
			○	○	○	○	PRV	GRT	○	OWN	○
			○	○	○	○	PRV	GRT	○	OWN	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
	権限削除	REVOKE DBA	○	○	○	○	PRV	RVK	○	DBA	○
		REVOKE SCHEMA	○	○	○	○	PRV	RVK	○	DBA	○
		REVOKE SCHEMA OPERATION	○	○	○	○	PRV	RVK	○	OWN	○
		REVOKE CONNECT	○	○	○	○	PRV	RVK	○	DBA	○
		REVOKE RDAREA	○	○	○	○	PRV	RVK	○	DBA	○
		REVOKE アクセス権限	○	○	○	○	PRV	RVK	○	OWN	○
			○	○	○	○	PRV	RVK	○	OWN	○
			○	○	○	○	PRV	RVK	○	OWN	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	○	○	○	○	DEF	ALT	○	DBA	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
			○	○	○	○	DEF	ALT	○	OWN	○
		ALTER ROUTINE	○	○	○	○	DEF	ALT	○	DBA	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	OWN	○
		ALTER TABLE（ADD RDAREA,CHANGE ADAREA 以外）	○	○	○	○	DEF	ALT	○	RDA	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	SCO	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	OWN	○
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	○	○	○	○	DEF	ALT	○	RDA	○
			○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	SCO	○
		ALTER TRIGGER	○	○	○	○	DEF	ALT	○	DBA	○
			○	○	○	○	DEF	ALT	○	OWN	○
		ALTER INDEX	○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	SCO	○
		COMMENT	○	○	○	○	DEF	ALT	○	OWN	○
			○	○	○	○	DEF	ALT	○	SCO	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
	CREATE 系	CREATE FUNCTION	○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE INDEX（形式 1）	○	○	○	○	DEF	CRT	○	RDA	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE INDEX（形式 2）	○	○	○	○	DEF	CRT	○	RDA	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE INDEX（形式 3）	○	○	○	○	DEF	CRT	○	RDA	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE PROCEDURE	手続き中の各 SQL は，イベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。								
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE SCHEMA	○	○	○	○	DEF	CRT	○	DBA	○
			○	○	○	○	DEF	CRT	○	SCH	○
		CREATE TABLE	○	○	○	○	DEF	CRT	○	RDA	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE TRIGGER	トリガ SQL 中の各 SQL は，延長で実行する CREATE PROCEDURE からイベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。								
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト定義イベント	CREATE 系	CREATE TYPE	○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE VIEW	○	○	○	○	DEF	CRT	○	SEL	○
			○	○	○	○	DEF	CRT	○	SEL	○
			○	○	○	○	DEF	CRT	○	SEL	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE PUBLIC VIEW	○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	OWN	○
		CREATE SEQUENCE	○	○	○	○	DEF	CRT	○	RDA	○
			○	○	○	○	DEF	CRT	○	OWN	○
			○	○	○	○	DEF	CRT	○	SCO	○
			○	○	○	○	DEF	CRT	○	OWN	○
	DROP 系	DROP DATA TYPE	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
			○	○	○	○	DEF	DRP	○	OWN	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP FUNCTION	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		DROP INDEX	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP PROCEDURE	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP SCHEMA	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP TABLE	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP TRIGGER	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP VIEW DROP PUBLIC VIEW	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○
		DROP SEQUENCE	○	○	○	○	DEF	DRP	○	DBA	○
			○	○	○	○	DEF	DRP	○	OWN	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト操作イベント	操作	DELETE（静的 SQL）	○	○	○	○	ACS	DEL	○	DEL	○
			○	○	○	○	ACS	DEL	○	DEL	○
			○	○	○	○	ACS	DEL	○	DEL	○
		EXECUTE※1	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。								
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。								
		INSERT（静的 SQL）	○	○	○	○	ACS	INS	○	INS	○
			○	○	○	○	ACS	INS	○	INS	○
			○	○	○	○	ACS	INS	○	INS	○
			○	○	○	○	ACS	NXV	○	USG	○
		LOCK TABLE（共用）（静的 SQL）	○	○	○	○	ACS	LCK	○	SEL	○
			○	○	○	○	ACS	LCK	○	SEL	○
			○	○	○	○	ACS	LCK	○	SEL	○
		LOCK TABLE（排他）（静的 SQL）	○	○	○	○	ACS	LCK	○	INS	○
			○	○	○	○	ACS	LCK	○	INS	○
			○	○	○	○	ACS	LCK	○	INS	○
			○	○	○	○	ACS	LCK	○	UPD	○
			○	○	○	○	ACS	LCK	○	UPD	○
			○	○	○	○	ACS	LCK	○	UPD	○
			○	○	○	○	ACS	LCK	○	DEL	○
			○	○	○	○	ACS	LCK	○	DEL	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
				○	○	○	○	ACS	LCK	○	DEL	○
		OPEN※1	SELECT	○	○	○	○	ACS	SEL	○	SEL	○
				○	○	○	○	ACS	SEL	○	SEL	○
				○	○	○	○	ACS	SEL	○	SEL	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト操作イベント	操作	PREPARE※1	ASSIGN LIST（形式 1）	○	○	○	○	ACS	ASN	○	SEL	○
			DELETE	○	○	○	○	ACS	DEL	○	DEL	○
				○	○	○	○	ACS	DEL	○	DEL	○
				○	○	○	○	ACS	DEL	○	DEL	○
			INSERT	○	○	○	○	ACS	INS	○	INS	○
				○	○	○	○	ACS	INS	○	INS	○
				○	○	○	○	ACS	INS	○	INS	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
				○	○	○	○	ACS	NXV	○	USG	○
			LOCK TABLE（共用）	○	○	○	○	ACS	LCK	○	SEL	○
				○	○	○	○	ACS	LCK	○	SEL	○
				○	○	○	○	ACS	LCK	○	SEL	○
			LOCK TABLE（排他）	○	○	○	○	ACS	LCK	○	INS	○
				○	○	○	○	ACS	LCK	○	INS	○
				○	○	○	○	ACS	LCK	○	INS	○
				○	○	○	○	ACS	LCK	○	UPD	○
				○	○	○	○	ACS	LCK	○	UPD	○
				○	○	○	○	ACS	LCK	○	UPD	○
				○	○	○	○	ACS	LCK	○	DEL	○
				○	○	○	○	ACS	LCK	○	DEL	○
				○	○	○	○	ACS	LCK	○	DEL	○
			UPDATE	○	○	○	○	ACS	UPD	○	UPD	○
				○	○	○	○	ACS	UPD	○	UPD	○
				○	○	○	○	ACS	UPD	○	UPD	○
				○	○	○	○	ACS	NXV	○	USG	○
			PURGE TABLE	○	○	○	○	ACS	PRG	○	DEL	○
			SELECT	○	○	○	○	ACS	SEL	○	SEL	○
				○	○	○	○	ACS	SEL	○	SEL	○
				○	○	○	○	ACS	SEL	○	SEL	○
		PURGE TABLE		○	○	○	○	ACS	PRG	○	DEL	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		（静的 SQL）									
		UPDATE（静的 SQL）	○	○	○	○	ACS	UPD	○	UPD	○
			○	○	○	○	ACS	UPD	○	UPD	○
			○	○	○	○	ACS	UPD	○	UPD	○
			○	○	○	○	ACS	NXV	○	USG	○
		SELECT，1 行 SELECT 文（静的 SQL）	○	○	○	○	ACS	SEL	○	SEL	○
			○	○	○	○	ACS	SEL	○	SEL	○
			○	○	○	○	ACS	SEL	○	SEL	○
ユティリティ操作イベント	生成	pdload※2	○	○	○	○	UTL	LOD	○	INS	"pdload"
			○	○	○	○	UTL	LOD	○	DEL	"pdload"
			○	○	○	○	UTL	LOD	○	USG	"pdload"
		pdexp pddefrev	○	○	○	○	UTL	EXP	○	DBA	"pdexp"
	運用	pdrorg	○	○	○	○	UTL	ORG	○	DBA	"pdrorg"
			○	○	○	○	UTL	ORG	○	SEL	"pdrorg"
			○	○	○	○	UTL	ORG	○	INS	"pdrorg"
			○	○	○	○	UTL	ORG	○	DEL	"pdrorg"
			○	○	○	○	UTL	ORG	○	AUD	"pdrorg"
			○	○	○	○	UTL	ORG	○	OWN	"pdrorg"
		pdconstck	○	○	○	○	UTL	CST	○	DBA	"pdconstck"
			○	○	○	○	UTL	CST	○	OWN	"pdconstck"

●権限チェック時の監査証跡のレコード項目（その 2）

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
システム管理者セキュリティイベント	HiRDBの開始	pdstart	-	-	○	-	○	○	-	-	-	-	-	-
	HiRDBの終了	pdstop	-	-	○	-	○	○	-	-	-	-	-	-
	主監査人の登録	pdmod	-	-	○	-	○	-	-	-	-	-	-	-
	副監査人の登録	pdmod	-	-	○	-	○	-	-	-	-	-	-	-
	監査証跡表の作成	pdmod	-	-	○	-	○	-	-	-	-	○	○	TBL
	監査証跡ファイルの削除	pdaudrm	-	-	○	-	○	○	-	-	-	-	○	AUF
	監査証跡の取得開始	pdaudbegin	-	-	○	-	○	○	-	-	-	-	-	-
		HiRDBの開始	-	-	○	-	○	○	-	-	-	-	-	-
	監査証跡の取得終了	pdaudend	-	-	○	-	○	○	-	-	-	-	-	-
		HiRDBの終了	-	-	○	-	○	○	-	-	-	-	-	-
	連続認証失敗アカウントロック状態の解除	pdacunlck（全DBA権限保持者アカウントロック時）	-	-	○	-	○	-	-	-	-	-	-	-
	CONNECT関連セキュリティ機能の設定値変更	CREATE CONNECTION SECURITY，DROP CONNECTION	△	○	○	-	○	○	○	○	○	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
		SECURITY												

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
監査人セキュリティイベント	監査証跡表へのデータロード	pdload※5	-	-	○	-	○	-	-	-	-	-	-	-
	監査証跡ファイルのスワップ	pdaudswap	-	-	○	-	○	○	-	-	-	-	-	-
	監査対象イベントの 定義	CREATE AUDIT	△	○	○	-	○	○	○	○	○	-	-	-
	監査対象イベントの 削除	DROP AUDIT	△	○	○	-	○	○	○	○	○	-	-	-
	監査人のパスワード変更	GRANT AUDIT	△	○	○	-	○	○	○	○	○	-	-	-
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	-	-	○	-	○	○	-	-	-	-	-	-
	監査証跡表の自動データ	pdaudatld -t	-	-	○	-	○	○	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
	ロード機能の停止													
	JP1/NETM/Audit用監査ログの出力	pdaudput	-	-	○	-	○	○	-	-	-	-	-	-
セッションセキュリティイベント	HiRDBへの接続	CONNECT	△	○	○	-	○	○	○	○	-	-	-	-
	ユーザの変更	SET SESSION AUTHORIZATION	△	○	○	-	○	○	○	○	-	-	-	-
権限管理イベント	権限付与	GRANT DBA	△	○	○	-	○	○	○	○	○	-	-	-
		GRANT SCHEMA	△	○	○	-	○	○	○	○	○	-	-	-
		GRANT SCHEMA OPERATION	△	○	○	-	○	○	○	○	○	○	○	SCH
		GRANT CONNECT	△	○	○	-	○	○	○	○	○	-	-	-
		GRANT RDAREA	△	○	○	-	○	○	○	○	○	-	-	-
		GRANTアクセス権限	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
	権限削除	REVOKE DBA	△	○	○	-	○	○	○	○	○	-	-	-
		REVOKE SCHEMA	△	○	○	-	○	○	○	○	○	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
		REVOKE SCHEMA OPERATION	△	○	○	-	○	○	○	○	○	○	○	SCH
		REVOKE CONNECT	△	○	○	-	○	○	○	○	○	-	-	-
		REVOKE RDAREA	△	○	○	-	○	○	○	○	○	-	-	-
		REVOKE アクセス権限	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
オブジェクト定義イベント	ALTER系	ALTER PROCEDURE	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	PRC
		ALTER ROUTINE	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	FNC
			△	○	○	-	○	○	○	○	○	○	○	PRC
			△	○	○	-	○	○	○	○	○	○	○	TRG
		ALTER TABLE（ADD RDAREA, CHANGE ADAREA以外）	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	IDX
			△	○	○	-	○	○	○	○	○	○	○	PRC
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）										
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
			△	○	○	-	○	○	○	○	○	○	○	TRG
			△	○	○	-	○	○	○	○	○	○	○	VIW
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL
		ALTER TRIGGER	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	TRG
		ALTER INDEX	△	○	○	-	○	○	○	○	○	○	○	IDX
			△	○	○	-	○	○	○	○	○	○	○	IDX
		COMMENT	△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL
	CREATE系	CREATE FUNCTION	△	○	○	-	○	○	○	○	○	○	○	FNC
			△	○	○	-	○	○	○	○	○	○	○	FNC
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE INDEX（形式 1）	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	IDX
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE INDEX（形式 2）	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	IDX
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	SCH

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）										
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
		CREATE INDEX（形式 3）	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	IDX
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE PROCEDURE	手続き中の各 SQL は，イベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。										
			△	○	○	-	○	○	○	○	○	○	○	PRC
			△	○	○	-	○	○	○	○	○	○	○	PRC
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE SCHEMA	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	-	-	-
		CREATE TABLE	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE TRIGGER	トリガ SQL 中の各 SQL は，延長で実行する CREATE PROCEDURE からイベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。										
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TRG
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	TRG
			△	○	○	-	○	○	○	○	○	○	○	SCH

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
オブジェクト定義イベント	CREATE系	CREATE TYPE	△	○	○	-	○	○	○	○	○	○	○	TYP
			△	○	○	-	○	○	○	○	○	○	○	TYP
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE VIEW	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
			△	○	○	-	○	○	○	○	○	○	○	VIW
			△	○	○	-	○	○	○	○	○	○	○	VIW
			△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE PUBLIC VIEW	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
		CREATE SEQUENCE	△	○	○	-	○	○	○	○	○	-	○	RDA
			△	○	○	-	○	○	○	○	○	○	○	SEQ
			△	○	○	-	○	○	○	○	○	○	○	SEQ
			△	○	○	-	○	○	○	○	○	○	○	SCH
	DROP系	DROP DATA TYPE	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	FNC
			△	○	○	-	○	○	○	○	○	○	○	PRC
			△	○	○	-	○	○	○	○	○	○	○	TYP
		DROP FUNCTION	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	FNC
		DROP INDEX	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	IDX
		DROP PROCEDURE	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	PRC

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
		DROP SCHEMA	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	SCH
		DROP TABLE	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	TBL
		DROP TRIGGER	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	TRG
		DROP VIEW DROP PUBLIC VIEW	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	VIW
		DROP SEQUENCE	△	○	○	-	○	○	○	○	○	-	-	-
			△	○	○	-	○	○	○	○	○	○	○	SEQ

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
オブジェクト操作イベント	操作	DELETE（静的 SQL）	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
		EXECUTE※1	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		INSERT（静的 SQL）	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	SEQ
		LOCK TABLE（共用）（静的SQL）		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
		LOCK TABLE（排他）（静的SQL）		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
		OPEN※1	SELECT	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
オブジェクト操作イベント	操作	PREPARE※1	ASSIGN LIST（形式1）	△	○	○	-	○	○	○	○	○	○	○	TBL

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
			DELETE	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
			INSERT	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	SEQ
			LOCK TABLE（共用）	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
			LOCK TABLE（排他）	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
			UPDATE	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	SEQ
			PURGE TABLE	△	○	○	-	○	○	○	○	○	○	○	TBL
			SELECT	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※3	オブジェクト名称	オブジェクト種別
			△	○	○	-	○	○	○	○	○	○	○	VIW
		PURGE TABLE（静的 SQL）	△	○	○	-	○	○	○	○	○	○	○	TBL
		UPDATE（静的 SQL）	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
			△	○	○	-	○	○	○	○	○	○	○	SEQ
		SELECT，1 行 SELECT 文（静的 SQL）	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
ユティリティ操作イベント	生成	pdload※2	-	-	○	-	○	-	-	-	-	○	○	TBL
			-	-	○	-	○	-	-	-	-	○	○	TBL
			-	-	○	-	○	-	-	-	-	○	○	SEQ
		pdexp pddefrev	-	-	○	-	○	-	-	-	-	-	-	-
	運用	pdrorg	-	-	○	-	○	-	-	-	-	-	-	-
			-	-	○	-	○	-	-	-	-	○	○	TBL
			-	-	○	-	○	-	-	-	-	○	○	TBL
			-	-	○	-	○	-	-	-	-	○	○	TBL
			-	-	○	-	○	-	-	-	-	-	-	-
			-	-	○	-	○	-	-	-	-	○	○	SCH
		pdconstck	-	-	○	-	○	-	-	-	-	-	-	-
			-	-	○	-	○	-	-	-	-	○	○	TBL

●権限チェック時の監査証跡のレコード項目（その 3）

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
システム管理者セキュリティイベント	HiRDBの開始	pdstart	-	-	○	-	-	-	-	-	-	-	-	-
	HiRDBの終了	pdstop	-	-	-	-	-	-	-	-	-	-	-	
	主監査人の登録	pdmod	AUD	○	-	-	-	-	-	-	-	-	-	-
	副監査人の登録	pdmod	AUD	○	-	-	-	-	-	-	-	-	-	-
	監査証跡表の作成	pdmod	-	-	-	-	-	-	-	-	-	-	-	-
	監査証跡ファイルの削除	pdaudrm	-	-	-	-	-	-	-	-	-	-	-	-
	監査証跡の取得開始	pdaudbegin	-	-	-	-	-	-	-	-	-	-	-	-
		HiRDBの開始	-	-	-	-	-	-	-	-	-	-	-	-
	監査証跡の取得終了	pdaudend	-	-	-	-	-	-	-	-	-	-	-	-
		HiRDBの終了	-	-	-	-	-	-	-	-	-	-	-	-
	連続認証失敗アカウントロック状態の解除	pdacunlck（全DBA権限保持者アカウ	-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		ントロック時）												
	CONNECT関連セキュリティ機能の設定値変更	CREATE CONNECTION SECURITY，DROP CONNECTION SECURITY	-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
監査人セキュリティイベント	監査証跡表へのデータロード	pdload※5	-	-	-	-	-	-	-	-	-	-	-	-
	監査証跡ファイルのスワップ	pdaudswap	-	-	-	-	-	-	-	-	-	-	-	-
	監査対象イベントの 定義	CREATE AUDIT	-	-	-	-	-	-	-	-	-	-	-	-
	監査対象イベントの 削除	DROP AUDIT	-	-	-	-	-	-	-	-	-	-	-	-
	監査人のパスワード変更	GRANT AUDIT	AUD	○	-	-	-	-	-	-	-	-	-	-
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	-	-	-	-	-	-	-	-	-	-	-	-
	監査証跡表の自動データロード機能の停止	pdaudatld -t	-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
	JP1/NETM/Audit 用監査ログの出力	pdaudput	-	-	-	-	-	-	-	-	-	-	-	-
セッションセキュリティイベント	HiRDB への接続	CONNECT	-	-	-	-	-	-	-	-	-	-	-	-
	ユーザの変更	SET SESSION AUTHORIZATION	-	-	-	-	-	-	-	-	-	-	-	-
権限管理イベント	権限付与	GRANT DBA	DBA	○	-	-	-	-	-	-	-	-	-	-
		GRANT SCHEMA	SCH	○	-	-	-	-	-	-	-	-	-	-
		GRANT SCHEMA OPERATION	SCO	○	-	-	-	-	-	-	-	-	-	-
		GRANT CONNECT	CNT	○	-	-	-	-	-	-	-	-	-	-
		GRANT RDAREA	RDA	○	-	-	-	-	-	-	-	-	-	-
		GRANT アクセス権限	○	○	-	-	-	-	-	-	-	-	-	-
			○	○	-	-	-	-	-	-	-	-	-	-
			○	○	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
	権限削除	REVOKE DBA	DBA	○	-	-	-	-	-	-	-	-	-	-
		REVOKE SCHEMA	SCH	○	-	-	-	-	-	-	-	-	-	-
		REVOKE SCHEMA OPERATION	SCO	○	-	-	-	-	-	-	-	-	-	-
		REVOKE CONNECT	CNT	○	-	-	-	-	-	-	-	-	-	-
		REVOKE RDAREA	RDA	○	-	-	-	-	-	-	-	-	-	-
		REVOKE アクセス権限	○	○	-	-	-	-	-	-	-	-	-	-
			○	○	-	-	-	-	-	-	-	-	-	-
			○	○	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		ALTER ROUTINE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		ALTER TABLE（ADD RDAREA, CHANGE ADAREA 以外）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		ALTER TRIGGER	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		ALTER INDEX	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		COMMENT	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト定義イベント	CREATE系	CREATE FUNCTION	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE INDEX	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		（形式 1）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE INDEX（形式 2）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE INDEX（形式 3）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE PROCEDURE	手続き中の各 SQL は，イベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。											
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE SCHEMA	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		CREATE TABLE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE TRIGGER	トリガ SQL 中の各 SQL は，延長で実行する CREATE PROCEDURE からイベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。											
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト定義イベント	CREATE系	CREATE TYPE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE VIEW	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE PUBLIC VIEW	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		CREATE SEQUENCE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
	DROP系	DROP DATA TYPE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		DROP FUNCTION	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		DROP INDEX	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		DROP PROCEDURE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト定義	DROP系	DROP SCHEMA	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
イベント		DROP TABLE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		DROP TRIGGER	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		DROP VIEW DROP PUBLIC VIEW	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		DROP SEQUENCE	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト操作イベント	操作	DELETE（静的 SQL）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		EXECUTE※1	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		INSERT（静的 SQL）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		LOCK TABLE（共用）（静的 SQL）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		LOCK TABLE（排他）（静的 SQL）	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）											
				付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
		OPEN ※1	SELECT	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）												
				付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト操作イベント	操作	PREPARE※1	ASSIGN LIST（形式 1）	-	-	-	-	-	-	-	-	-	-	-	-
			DELETE	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
			INSERT	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
			LOCK TABLE（共用）	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
			LOCK TABLE（排他）	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）												
				付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
			UPDATE	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
			PURGE TABLE	-	-	-	-	-	-	-	-	-	-	-	-
			SELECT	-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
		PURGE TABLE（静的 SQL）		-	-	-	-	-	-	-	-	-	-	-	-
		UPDATE（静的 SQL）		-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-
		SELECT，1 行 SELECT 文		-	-	-	-	-	-	-	-	-	-	-	-
				-	-	-	-	-	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限付与/削除/変更されたユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		（静的 SQL）	-	-	-	-	-	-	-	-	-	-	-	-
ユティリティ操作イベント	生成	pdload※2	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		pdexp pddefrev	-	-	-	-	-	-	-	-	-	-	-	-
	運用	pdrorg	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-
		pdconstck	-	-	-	-	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-	-	-	-	-

●権限チェック時の監査証跡のレコード項目（その 4）

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
システム管理者セキュリティイベント	HiRDB の開始	pdstart	-	-	-	-	-	-	○	○	1	-	pdstart コマンドを実行したユニット
	HiRDB の終了	pdstop	-	-	-	-	-	-	○	○	1	-	MGR
	主監査人の登録	pdmod	-	-	-	-	-	-	○	○	aud+atb	-	DS
	副監査人の登録	pdmod	-	-	-	-	-	-	○	○	aud+atb	-	DS
	監査証跡表の作成	pdmod	-	-	-	-	-	-	○	○	aud+atb	-	DS
	監査証跡ファイルの削除	pdaudrm	-	-	-	-	-	-	○	○	1	-	MGR
	監査証跡の取得開始	pdaudbegin	-	-	-	-	-	-	○	○	1	-	MGR
		HiRDB の開始	-	-	-	-	-	-	○	○			稼働中の各ユニット
	監査証跡の取得終了	pdaudend	-	-	-	-	-	-	○	○	1	-	MGR
		HiRDB の終了	-	-	-	-	-	-	○	○			終了した各ユニット
	連続認証失敗アカウントロック状態の解除	pdacunlck（全 DBA 権限保持者アカウントロック時）	-	-	-	-	-	-	○	○	1	-	DS

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
	CONNECT関連セキュリティ機能の設定値変更	CREATE CONNECTION SECURITY，DROP CONNECTION SECURITY	-	-	○	○	○	○	○	○	1	-	DS

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
監査人セキュリティイベント	監査証跡表へのデータロード	pdload※5	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
	監査証跡ファイルのスワップ	pdaudswap	-	-	-	-	-	-	○	○	1	-	MGR
	監査対象イベントの 定義	CREATE AUDIT	-	-	○	○	○	○	○	○	1	-	DS
	監査対象イベントの 削除	DROP AUDIT	-	-	○	○	○	○	○	○	1	-	DS
	監査人のパスワード変更	GRANT AUDIT	-	-	○	○	○	○	○	○	1	-	DS
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	-	-	-	-	-	-	○	○	1	-	MGR
	監査証跡表の自動データロード機能の停止	pdaudatld -t	-	-	-	-	-	-	○	○	1	-	MGR
	JP1/NETM/Audit 用監査ログの出力	pdaudput	-	-	-	-	-	-	○	○	1	-	MGR

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
セッションセキュリティイベント	HiRDB への接続	CONNECT	-	-	-	-	-	-	○	○	1	-	FES
	ユーザの変更	SET SESSION AUTHORIZATION	-	-	-	-	-	-	○	○	1	-	FES
権限管理イベント	権限付与	GRANT DBA	-	-	○	○	○	○	○	○	usr	-	DS
		GRANT SCHEMA	-	-	○	○	○	○	○	○	usr	-	DS
		GRANT SCHEMA OPERATION	-	-	○	○	○	○	○	○	usr	-	DS
		GRANT CONNECT	-	-	○	○	○	○	○	○	usr	-	DS
		GRANT RDAREA	-	-	○	○	○	○	○	○	usr	-	DS
		GRANT アクセス権限	-	-	○	○	○	○	○	○	usr × acs	-	DS
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
	権限削除	REVOKE DBA	-	-	○	○	○	○	○	○	usr	-	DS
		REVOKE SCHEMA	-	-	○	○	○	○	○	○	usr	-	DS

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		REVOKE SCHEMA OPERATION	-	-	○	○	○	○	○	○	usr	-	DS
		REVOKE CONNECT	-	-	○	○	○	○	○	○	usr	-	DS
		REVOKE RDAREA	-	-	○	○	○	○	○	○	usr	-	DS
		REVOKE アクセス権限	-	-	○	○	○	○	○	○	usr × acs	DROP VIEW	DS
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	-	-	○	○	○	○	○	○	1	CREATE PROCEDURE（ただし，CREATE PROCEDURE 中のオブジェクト種別 PRC，SCH は出力されません）	DS
			-	-	○	○	○	○	○	○	prc		
		ALTER ROUTINE	-	-	○	○	○	○	○	○	1	• CREATE PROCEDURE（ただし，CREATE PROCEDURE 中のオブジェクト種別 PRC，SCH は出力されません） • CREATE FUNCTION（ただし，CREATE FUNCTION 中のオブジェクト種別 FNC，	DS
			-	-	○	○	○	○	○	○	fnc		
			-	-	○	○	○	○	○	○	prc		
			-	-	○	○	○	○	○	○	trg		

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
												SCH は出力されません）	
		ALTER TABLE（ADD RDAREA,CHANGE ADAREA 以外）	-	-	○	○	○	○	○	○	rda + rlb + rid	DROP VIEW，DROP INDEX，DROP TRIGGER，DROP PUBLIC VIEW	FES 又は DS
			-	-	○	○	○	○	○	○	idx		
			-	-	○	○	○	○	○	○	prc		
			-	-	○	○	○	○	○	○	tbl		
			-	-	○	○	○	○	○	○	tbl		
			-	-	○	○	○	○	○	○	trg		
			-	-	○	○	○	○	○	○	viw		
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	-	-	○	○	○	○	○	○	rda + rlb + rid + cls + prm	-	FES 又は DS
			-	-	○	○	○	○	○	○	tbl		
			-	-	○	○	○	○	○	○	tbl		

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		ALTER TRIGGER	-	-	○	○	○	○	○	○	1	CREATE PROCEDURE（ただし，CREATE PROCEDURE 中のオブジェクト種別 PRC，SCH は出力されません）	DS
			-	-	○	○	○	○	○	○			
		ALTER INDEX	-	-	○	○	○	○	○	○	1	-	FES 又は DS
			-	-	○	○	○	○	○	○	1		
		COMMENT	-	-	○	○	○	○	○	○	1	-	DS
			-	-	○	○	○	○	○	○	1		

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト定義イベント	CREATE系	CREATE FUNCTION	-	-	○	○	○	○	○	○	1	-	FES 又は DS
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE INDEX（形式 1）	-	-	○	○	○	○	○	○	rda	-	DS
			-	-	○	○	○	○	○	○	1		
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE INDEX（形式 2）	-	-	○	○	○	○	○	○	rda	-	DS
			-	-	○	○	○	○	○	○	1		
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE INDEX（形式 3）	-	-	○	○	○	○	○	○	rda	-	DS
			-	-	○	○	○	○	○	○	1		
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE PROCEDURE	手続き中の各 SQL は，イベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。									-	FES 又は DS
			-	-	○	○	○	○	○	○	1		

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE SCHEMA	-	-	○	○	○	○	○	○	1	-	DS
			-	-	○	○	○	○	○	○			
		CREATE TABLE	-	-	○	○	○	○	○	○	rda + rid + rlb + cls + prm	CREATE TRIGGER（制約動作を行うためのトリガを生成する場合）	DS
			-	-	○	○	○	○	○	○	1		
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE TRIGGER	トリガ SQL 中の各 SQL は，延長で実行する CREATE PROCEDURE からイベントタイプ DEF，イベントサブタイプ CRT で出力されます。そのほかの出力内容については，各 SQL を参照してください。									CREATE PROCEDURE	DS
			-	-	○	○	○	○	○	○	1		
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト定義イベント	CREATE系	CREATE TYPE	-	-	○	○	○	○	○	○	1	• CREATE PROCEDURE（メンバに手続きがある場合） • CREATE FUNCTION（メンバに関数がある場合）	DS
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		CREATE VIEW	-	-	○	○	○	○	○	○	tbl	-	FES 又は DS
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○	1		
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE PUBLIC VIEW	-	-	○	○	○	○	○	○	tbl※4	-	FES
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		CREATE SEQUENCE	-	-	○	○	○	○	○	○	1	-	FES
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
	DROP 系	DROP DATA TYPE	-	-	○	○	○	○	○	○	1	DROP FUNCTION，DROP PROCEDURE	DS
			-	-	○	○	○	○	○	○	fnc		
			-	-	○	○	○	○	○	○	prc		
			-	-	○	○	○	○	○	○	1		
		DROP FUNCTION	-	-	○	○	○	○	○	○	1	DROP DATA TYPE	DS
			-	-	○	○	○	○	○	○			
		DROP INDEX	-	-	○	○	○	○	○	○	1	-	DS
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		DROP PROCEDURE	-	-	○	○	○	○	○	○	1	DROP DATA TYPE	DS
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト定義イベント	DROP 系	DROP SCHEMA	-	-	○	○	○	○	○	○	1	DROP TABLE，DROP VIEW，DROP FOREIGN TABLE，DROP INDEX，DROP FOREIGN INDEX，	DS
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
												DROP PROCEDURE，DROP FUNCTION，DROP DATA TYPE，DROP TRIGGER，DROP PUBLIC VIEW，DROP PUBLIC FUNCTION，DROP PUBLIC PROCEDURE，DROP SEQUENCE	
		DROP TABLE	-	-	○	○	○	○	○	○	1	DROP VIEW，DROP INDEX，DROP TRIGGER，DROP PROCEDURE，DROP PUBLIC VIEW	DS
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		DROP TRIGGER	-	-	○	○	○	○	○	○	1	DROP PROCEDURE	DS
			-	-	○	○	○	○	○	○			
		DROP VIEW DROP PUBLIC VIEW	-	-	○	○	○	○	○	○	1	DROP VIEW，DROP PUBLIC VIEW	DS
			-	-	○	○	○	○	○	○			
		DROP SEQUENCE	-	-	○	○	○	○	○	○	1	-	DS
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト操作イベント	操作	DELETE（静的 SQL）	-	-	○	○	○	○	○	○	1	トリガ	FES
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			-	-	○	○	○	○	○	○		SQL文（制約動作を行うためのトリガも含む）	
		EXECUTE※1	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。									前処理したSQL	FES
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。									指定したSQL	FES
		INSERT（静的 SQL）	-	-	○	○	○	○	○	○	1	• SELECT • トリガSQL文（制約動作	FES
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○	seq		

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
												を行うためのトリガも含む）	
		LOCK TABLE（共用）（静的SQL）	-	-	○	○	○	○	○	○	tbl	-	FES
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
		LOCK TABLE（排他）（静的SQL）	-	-	○	○	○	○	○	○	tbl	-	FES
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別		証跡レコード項目（その4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		OPEN※1	SELECT	-	-	○	○	○	○	○	○	tbl	-	FES
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別		証跡レコード項目（その4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト操作イベント	操作	PREPARE※1	ASSIGN LIST（形式1）	-	-	○	○	○	○	○	○	tbl	-	FES
			DELETE	-	-	○	○	○	○	○	○	1	PREPARE（SELECT）	
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			INSERT	-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○	seq		
			LOCK TABLE（共用）	-	-	○	○	○	○	○	○	tbl	-	
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
			LOCK TABLE（排他）	-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
			UPDATE	-	-	○	○	○	○	○	○	1	PREPARE（SELECT）	
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○	seq		

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			PURGE TABLE	-	-	○	○	○	○	○	○	1	-	
			SELECT	-	-	○	○	○	○	○	○	tbl	PREPARE（SELECT）	
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
		PURGE TABLE（静的 SQL）		-	-	○	○	○	○	○	○	1	-	FES
		UPDATE（静的 SQL）		-	-	○	○	○	○	○	○	1	• SELECT • トリガSQL文（制約動作を行うためのト	FES
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○			
				-	-	○	○	○	○	○	○	seq		

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
												リガも含む）	
		SELECT， 1 行 SELECT 文（静的 SQL）	-	-	○	○	○	○	○	○	tbl	-	FES
			-	-	○	○	○	○	○	○			
			-	-	○	○	○	○	○	○			
ユティリティ操作イベント	生成	pdload※2	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
			-	-	-	-	-	-	○	○			
			-	-	-	-	-	-	○	○	seq		
		pdexp pddefrev	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
	運用	pdrorg	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
			-	-	-	-	-	-	○	○	tbl		
			-	-	-	-	-	-	○	○			
			-	-	-	-	-	-	○	○			
			-	-	-	-	-	-	○	○	1		

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			-	-	-	-	-	-	○	○			
		pdconstck	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
			-	-	-	-	-	-	○	○			

（凡例）

○：情報を取得します。

△：XA インタフェース経由で情報を取得することがあります。

−：情報を取得しません。又は該当しません。

acs：指定した権限の数（all を指定した場合は 4）

atb：pdmod コマンドで作成する監査証跡表の数（作成する場合は 1，作成しない場合は 0）

aud：pdmod コマンドで作成する監査人の数（作成する場合は 1，作成しない場合は 0）

cls：FOR CLUSTER KEY 句を指定した場合は 1，指定しない場合は 0

fnc：再作成の対象となる関数の数

idx：関連するインデクス数

prc：再作成の対象になるプロシジャ数

prm：FOR PRIMARY KEY 句を指定した場合は 1，指定しない場合は 0

rda：指定した RD エリア数

rlb：LOB データ格納 RD エリア数

rid：インデクス格納 RD エリア数

seq：使用する順序数生成子数

tbl：SELECT 文，SELECT 句，又は LOCK 文で指定した表数，又はユティリティの処理対象表数

trg：再作成の対象になるトリガ数

usr：指定ユーザ数又は指定グループ数

viw：対象となるビュー表数

注※1

SQL 種別が SELECT，INSERT，UPDATE，DELETE の場合は，PREPARE 文による前処理時にも使用権限チェックをして監査証跡を取得するため，記録レコード数が２倍になります。使用権限チェックのタイミングを次に示します。下線部分の SQL 文で使用権限チェックをして監査証跡を取得します。

SQL 種別が SELECT の場合

・PREPARE 文での前処理をしない場合

```
EXEC SQL DECLARE C1 CURSOR FOR SELECT * FROM T1;
EXEC SQL OPEN C1;
```

・PREPARE 文で前処理を行う場合

```
EXEC SQL PREPARE S1 FROM 'SELECT * FROM T1';
EXEC SQL DECLARE C1 CURSOR FOR S1;
EXEC SQL OPEN C1;
```

SQL 種別が INSERT の場合（UPDATE，DELETE も同様です）

・PREPARE 文での前処理をしない場合

```
EXEC SQL INSERT INTO T1(C1) VALUES(1);
```

・PREPARE 文で前処理を行う場合

```
EXEC SQL PREPARE S1 FROM 'INSERT INTO T1(C1) VALUES(?)';
EXEC SQL EXECUTE S1 FOR :data;
```

注※2

監査証跡表以外又は監査証跡表と認識できない場合が該当します。

注※3

対象オブジェクトがパブリックビュー表，パブリック関数，又はパブリック手続きの場合，オブジェクトの所有者には PUBLIC が記録されます。

注※4

基表がビュー表又はパブリックビュー表の場合，その基表となる実表数分レコード数が追加されます。

注※5

監査証跡表の自動データロード機能を適用して，HiRDB が pdload を実行した場合は記録されません。

24.14　監査証跡のレコード項目（イベント終了時）

イベント終了時の監査証跡のレコード項目を次に示します。

ユーザ識別子～オブジェクト種別までのレコード項目は（その１）に，付与/削除変更した権限以降のレコード項目は（その２）に記載しています。

表中のイベントタイプ，イベントサブタイプについては，表「イベントタイプ及びイベントサブタイプの詳細」を，使用した権限については，表「監査証跡表の列構成」の USED_PRIVILEGE を参照してください。

●イベント終了時の監査証跡のレコード項目（その 1）

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
システム管理者セキュリティイベント	HiRDB の開始	pdstart	○	○	○	○	SYS	STR	○	'△△△'	-
	HiRDB の終了	pdstop	○	○	○	○	SYS	STP	○	'△△△'	-
	主監査人の登録	pdmod	○	○	○	○	SYS	MOD	○	'△△△'	"pdmod"
	副監査人の登録	pdmod	○	○	○	○	SYS	MDA	○	'△△△'	"pdmod"
	監査証跡表の作成	pdmod	○	○	○	○	SYS	MOD	○	'△△△'	"pdmod"
	監査証跡ファイルの削除	pdaudrm	○	○	○	○	SYS	ARM	○	'△△△'	"pdaudrm"
	監査証跡の取得開始	pdaudbegin	○	○	○	○	SYS	ABG	○	'△△△'	"pdaudbegin"
		HiRDB の開始	○	○	○	○	SYS	ABG	○	'△△△'	"pdstart"
	監査証跡の取得終了	pdaudend	○	○	○	○	SYS	AEN	○	'△△△'	"pdaudend"

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		HiRDB の終了	○	○	○	○	SYS	AEN	○	'△△△'	"pdstop"
	監査証跡ファイルの上書き	システムによる自動上書き	○	○	○	○	SYS	OVW	○	'△△△'	-
		pdaudswap による手動上書き	○	○	○	○	SYS	OVW	○	'△△△'	"pdaudswap"
	連続認証失敗アカウントロック状態への遷移	CONNECT 失敗	○	○	○	○	SYS	CLK	○	'△△△'	-
	連続認証失敗アカウントロック状態の解除	アカウントロック期間経過後の CONNECT	○	○	○	○	SYS	CUL	○	'△△△'	-
		DROP CONNECTION SECURITY FOR CONNECT	○	○	○	○	SYS	CUL	○	'△△△'	○
		pdacunlck	○	○	○	○	SYS	CUL	○	'△△△'	"pdacunlck"

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
システム管理者セキュリティイベント	パスワード無効アカウントロック状態への遷移		○	○	○	○	SYS	PLK	○	'△△△'	○
	パスワード無効アカウントロック状態の解除		○	○	○	○	SYS	PUL	○	'△△△'	○
	CONNECT 関連セキュリティ機能の設定値変更	CREATE/DROP CONNECTION SECURITY	○	○	○	○	SYS	SPR	○	'△△△'	○
	pdacunlck の実行		○	○	○	○	SYS	ULK	○	'△△△'	"pdacunlck"
監査人セキュリティ イベント	監査証跡表へのデータロード	pdload（監査人によるデータロード）	○	○	○	○	AUD	ALD	○	'△△△'	"pdload"
		pdload（監査証跡表の自動データロード機能によるデータロード）	○※4	○	○	○	AUD	ATL	○	'△△△'	"pdload"
	監査証跡ファイルのスワップ	pdaudswap	○	○	○	○	AUD	ASW	○	'△△△'	"pdaudswap"
	監査対象イベントの定義	CREATE AUDIT	○	○	○	○	AUD	CRT	○	'△△△'	○
	監査対象イベントの削除	DROP AUDIT	○	○	○	○	AUD	DRP	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
	監査人のパスワード変更	GRANT AUDIT	○	○	○	○	AUD	GRT	○	'△△△'	○
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	○	○	○	○	AUD	ATB	○	'△△△'	"pdaudatld"
	監査証跡表の自動データロード機能の停止	pdaudatld -t	○	○	○	○	AUD	ATT	○	'△△△'	"pdaudatld"
		データロード失敗に伴う停止	○	○	○	○	AUD	ATT	○	'△△△'	-
	JP1/NETM/Audit 用監査ログの出力	pdaudput	○	○	○	○	AUD	APT	○	'△△△'	"pdaudput"
セッションセキュリティイベント	HiRDB への接続	CONNECT	○	○	○	○	SES	CNT	○	'△△△'	○
	ユーザの変更	SET SESSION AUTHORIZATION	○	○	○	○	SES	ATH	○	'△△△'	△
	HiRDB との切り離し	DISCONNECT COMMIT RELEASE ROLLBACK RELEASE	○	○	○	○	SES	DIS	○	'△△△'	△

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
権限管理イベント	権限付与	GRANT DBA	○	○	○	○	PRV	GRT	○	'△△△'	○
		GRANT SCHEMA	○	○	○	○	PRV	GRT	○	'△△△'	○
		GRANT SCHEMA OPERATION	○	○	○	○	PRV	GRT	○	'△△△'	○
		GRANT CONNECT	○	○	○	○	PRV	GRT	○	'△△△'	○
		GRANT RDAREA	○	○	○	○	PRV	GRT	○	'△△△'	○
		GRANT アクセス権限	○	○	○	○	PRV	GRT	○	'△△△'	○
			○	○	○	○	PRV	GRT	○	'△△△'	○
			○	○	○	○	PRV	GRT	○	'△△△'	○
		GRANT ロール利用権限	○	○	○	○	PRV	GRT	○	'△△△'	○
	権限削除	REVOKE DBA	○	○	○	○	PRV	RVK	○	'△△△'	○
		REVOKE SCHEMA	○	○	○	○	PRV	RVK	○	'△△△'	○
		REVOKE SCHEMA OPERATION	○	○	○	○	PRV	RVK	○	'△△△'	○
		REVOKE CONNECT	○	○	○	○	PRV	RVK	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		REVOKE RDAREA	○	○	○	○	PRV	RVK	○	'△△△'	○
		REVOKE アクセス権限	○	○	○	○	PRV	RVK	○	'△△△'	○
			○	○	○	○	PRV	RVK	○	'△△△'	○
			○	○	○	○	PRV	RVK	○	'△△△'	○
		REVOKE ロール利用権限	○	○	○	○	PRV	GRT	○	'△△△'	○
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	○	○	○	○	DEF	ALT	○	'△△△'	○
		ALTER ROUTINE	○	○	○	○	DEF	ALT	○	'△△△'	○
		ALTER TABLE（ADD RDAREA,CHANGE ADAREA 以外）	○	○	○	○	DEF	ALT	○	'△△△'	○
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	○	○	○	○	DEF	ALT	○	'△△△'	○
		ALTER TRIGGER	○	○	○	○	DEF	ALT	○	'△△△'	○
		ALTER INDEX	○	○	○	○	DEF	ALT	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		COMMENT	○	○	○	○	DEF	ALT	○	'△△△'	○
			○	○	○	○	DEF	ALT	○	'△△△'	○
			○	○	○	○	DEF	ALT	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト定義イベント	CREATE 系	CREATE FUNCTION	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE INDEX（形式 1）	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE INDEX（形式 2）	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE INDEX	○	○	○	○	DEF	CRT	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		（形式 3）									
		CREATE PROCEDURE	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE ROLE	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE SCHEMA	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE TABLE	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE TRIGGER	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE TYPE	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE VIEW CREATE PUBLIC VIEW	○	○	○	○	DEF	CRT	○	'△△△'	○
		CREATE SEQUENCE	○	○	○	○	DEF	CRT	○	'△△△'	○
	DROP 系	DROP DATA TYPE	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP FUNCTION	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP INDEX	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP PROCEDURE	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP ROLE	○	○	○	○	DEF	DRP	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		DROP SCHEMA	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP TABLE	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP TRIGGER	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP VIEW DROP PUBLIC VIEW	○	○	○	○	DEF	DRP	○	'△△△'	○
		DROP SEQUENCE	○	○	○	○	DEF	DRP	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト操作イベント	操作	CALL	○	○	○	○	ACS	CAL	○	'△△△'	△
		DELETE（静的 SQL）	○	○	○	○	ACS	DEL	○	'△△△'	○

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 1）								
			ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
			○	○	○	○	ACS	DEL	○	'△△△'	○
			○	○	○	○	ACS	DEL	○	'△△△'	○
			○	○	○	○	ACS	DEL	○	'△△△'	○
		EXECUTE	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。								
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。								
		INSERT (静的 SQL)	○	○	○	○	ACS	INS	○	'△△△'	○
			○	○	○	○	ACS	INS	○	'△△△'	○
			○	○	○	○	ACS	INS	○	'△△△'	○
			○	○	○	○	ACS	INS	○	'△△△'	○
			○	○	○	○	ACS	NXV	○	'△△△'	○
		LOCK TABLE （共用）（静的 SQL）	○	○	○	○	ACS	LCK	○	'△△△'	○
			○	○	○	○	ACS	LCK	○	'△△△'	○
			○	○	○	○	ACS	LCK	○	'△△△'	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
				○	○	○	○	ACS	LCK	○	'△△△'	○
		LOCK TABLE（排他）（静的 SQL）		○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
		OPEN	SELECT※2	○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
		CLOSE	SELECT	○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
				○	○	○	○	ACS	SEL	○	'△△△'	○
		PREPARE	ASSIGN LIST（形式 1，形式 2）	○	○	○	○	ACS	ASN	○	'△△△'	○
			DELETE	○	○	○	○	ACS	DEL	○	'△△△'	○
				○	○	○	○	ACS	DEL	○	'△△△'	○
				○	○	○	○	ACS	DEL	○	'△△△'	○
				○	○	○	○	ACS	DEL	○	'△△△'	○
			INSERT	○	○	○	○	ACS	INS	○	'△△△'	○
				○	○	○	○	ACS	INS	○	'△△△'	○
				○	○	○	○	ACS	INS	○	'△△△'	○
				○	○	○	○	ACS	INS	○	'△△△'	○
				○	○	○	○	ACS	NXV	○	'△△△'	○
			LOCK TABLE（共用）	○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
				○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
			LOCK TABLE（排他）	○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
				○	○	○	○	ACS	LCK	○	'△△△'	○
			UPDATE	○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	NXV	○	'△△△'	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
オブジェクト操作イベント	操作	PREPARE	PURGE TABLE	○	○	○	○	ACS	PRG	○	'△△△'	○
				○	○	○	○	ACS	PRG	○	'△△△'	○
			SELECT※2	○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
		PURGE TABLE（静的 SQL）		○	○	○	○	ACS	PRG	○	'△△△'	○
				○	○	○	○	ACS	PRG	○	'△△△'	○
		UPDATE（静的 SQL）		○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	UPD	○	'△△△'	○
				○	○	○	○	ACS	NXV	○	'△△△'	○

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
		SELECT， 1 行 SELECT 文 （静的 SQL）※2		○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
				○	○	○	○	ACS	SEL	○	'△△△'	○
ユティリティ操作イベント	生成	pdload※3	表単位実行	○	○	○	○	UTL	LOD	○	'△△△'	"pdload"
				○	○	○	○	UTL	LOD	○	'△△△'	"pdload"
			RD エリア単位実行	○	○	○	○	UTL	LOD	○	'△△△'	"pdload"
				○	○	○	○	UTL	LOD	○	'△△△'	"pdload"
		pdexp pddefrev	表単位実行	○	○	○	○	UTL	EXP	○	'△△△'	"pdexp"
			ビュー表単位実行	○	○	○	○	UTL	EXP	○	'△△△'	"pdexp"
			別名表単位実行	○	○	○	○	UTL	EXP	○	'△△△'	"pdexp"
			プロシジャ単位実行	○	○	○	○	UTL	EXP	○	'△△△'	"pdexp"
			トリガ単位実行	○	○	○	○	UTL	EXP	○	'△△△'	"pdexp"

イベントの種類		コマンド／SQL 種別		証跡レコード項目（その 1）								
				ユーザ識別子	イベント実行日	イベント実行時刻	イベント実行時刻（マイクロ秒）	イベントタイプ	イベントサブタイプ	イベント成否	使用した権限	UAP 名称
	運用	pdrorg	スキーマ，表単位実行	○	○	○	○	UTL	ORG	○	'△△△'	"pdrorg"
			インデクス単位実行	○	○	○	○	UTL	ORG	○	'△△△'	"pdrorg"
			RD エリア単位実行	○	○	○	○	UTL	ORG	○	'△△△'	"pdrorg"
			ディクショナリ表のメンテナンス（インデクスの追加）	○	○	○	○	UTL	ORG	○	'△△△'	"pdrorg"
		pdconstck		○	○	○	○	UTL	CST	○	'△△△'	"pdconstck"

●イベント終了時の監査証跡のレコード項目（その 2）

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 2）											
			サービス名称	IP アドレス	プロセス ID	スレッド ID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL 通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
システム管理者セキュリティイベント	HiRDB の開始	pdstart	-	-	○	-	○	○	-	-	-	-	-	-
	HiRDB の終了	pdstop	-	-	○	-	○	○	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
	主監査人の登録	pdmod	-	-	○	-	○	-	-	-	-	-	-	-
	副監査人の登録	pdmod	-	-	○	-	○	-	-	-	-	-	-	-
	監査証跡表の作成	pdmod	-	-	○	-	○	-	-	-	-	○	○	TBL
	監査証跡ファイルの削除	pdaudrm	-	-	○	-	○	○	-	-	-	-	○	AUF
	監査証跡の取得開始	pdaudbegin	-	-	○	-	○	○	-	-	-	-	-	-
		HiRDB の開始	-	-	○	-	○	○	-	-	-	-	-	-
	監査証跡の取得終了	pdaudend	-	-	○	-	○	○	-	-	-	-	-	-
		HiRDB の終了	-	-	○	-	○	○	-	-	-	-	-	-
	監査証跡ファイルの上書き	システムによる自動上書き	-	-	○	-	○	○	-	-	-	-	-	AUF
		pdaudswap による手動上書き	-	-	○	-	○	○	-	-	-	-	-	AUF
	連続認証失敗アカウントロック状態への遷移	CONNECT 失敗	-	-	○	-	○	○	-	-	-	-	-	-
	連続認証失敗アカウントロック状態の解除	アカウントロック期間経過後の CONNECT	-	-	○	-	○	○	○	○	○	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
		DROP CONNECTION SECURITY FOR CONNECT	△	○	○	-	○	○	○	○	○	-	-	-
		pdacunlck	-	-	○	-	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
システム管理者セキュリティイベント	パスワード無効アカウントロック状態への遷移		△	○	○	-	○	○	○	○	○	-	-	-
	パスワード無効アカウントロック状態の解除		△	○	○	-	○	○	○	○	○	-	-	-
	CONNECT 関連セキュリティ機能の設定値変更	CREATE/DROP CONNECTION SECURITY	△	○	○	-	○	○	○	○	○	-	○	-
	pdacunlck の実行		-	-	○	-	○	-	-	-	-	-	-	-
監査人セキュリティイベント	監査証跡表へのデータロード	pdload（監査人によるデータロード）	-	-	○	-	○	-	-	-	-	○	○	TBL
		pdload（監査証跡表の	-	-	○	-	○	-	-	-	-	○	○	TBL

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
		自動データロード機能によるデータロード）												
	監査証跡ファイルのスワップ	pdaudswap	-	-	○	-	○	○	-	-	-	-	-	AUF
	監査対象イベントの定義	CREATE AUDIT	△	○	○	-	○	○	○	○	○	-	-	-
	監査対象イベントの削除	DROP AUDIT	△	○	○	-	○	○	○	○	○	-	-	-
	監査人のパスワード変更	GRANT AUDIT	△	○	○	-	○	○	○	○	○	-	-	-
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	-	-	○	-	○	○	-	-	-	-	-	-
	監査証跡表の自動データロード機能の停止	pdaudatld -t	-	-	○	-	○	○	-	-	-	-	-	-
		データロード失敗に伴う停止	-	-	○	-	○	○	-	-	-	-	-	-
	JP1/NETM/Audit用監査ログの出力	pdaudput	-	-	○	-	○	○	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
セッションセキュリティイベント	HiRDBへの接続	CONNECT	△	○	○	-	○	○	○	○	-	-	-	-
	ユーザの変更	SET SESSION AUTHORIZATION	△	△	○	-	○	○	○	○	○	-	-	-
	HiRDBとの切り離し	DISCONNECT COMMIT RELEASE ROLLBACK RELEASE	△	△	○	-	○	○	○	○	○	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
権限管理イベント	権限付与	GRANT DBA	△	○	○	-	○	○	○	○	○	-	-	-
		GRANT SCHEMA	△	○	○	-	○	○	○	○	○	-	-	-
		GRANT SCHEMA OPERATION	△	○	○	-	○	○	○	○	○	○	○	SCH
		GRANT CONNECT	△	○	○	-	○	○	○	○	○	-	-	-
		GRANT RDAREA	△	○	○	-	○	○	○	○	○	-	○	RDA
		GRANT アクセス権限	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
			△	○	○	-	○	○	○	○	○	○	○	VIW
		GRANT ロール利用権限	△	○	○	-	○	○	○	○	○	-	○	ROL
	権限削除	REVOKE DBA	△	○	○	-	○	○	○	○	○	-	-	-
		REVOKE SCHEMA	△	○	○	-	○	○	○	○	○	-	-	-
		REVOKE SCHEMA OPERATION	△	○	○	-	○	○	○	○	○	○	○	SCH
		REVOKE CONNECT	△	○	○	-	○	○	○	○	○	-	-	-
		REVOKE RDAREA	△	○	○	-	○	○	○	○	○	-	○	RDA
		REVOKE アクセス権限	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
		REVOKE ロール利用権限	△	○	○	-	○	○	○	○	○	-	○	ROL
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	△	○	○	-	○	○	○	○	○	○	○	PRC
		ALTER ROUTINE	△	○	○	-	○	○	○	○	○	○	○	PRC
		ALTER TABLE（ADD RDAREA, CHANGE	△	○	○	-	○	○	○	○	○	○	○	TBL

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
		ADAREA以外）												
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	△	○	○	-	○	○	○	○	○	○	○	TBL
		ALTER TRIGGER	△	○	○	-	○	○	○	○	○	○	○	TRG
		ALTER INDEX	△	○	○	-	○	○	○	○	○	○	○	IDX
		COMMENT	△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
			△	○	○	-	○	○	○	○	○	○	○	FTB

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
オブジェクト定義イベント	CREATE系	CREATE FUNCTION	△	○	○	-	○	○	○	○	○	○	○	FNC
		CREATE INDEX（形式 1）	△	○	○	-	○	○	○	○	○	○	○	IDX
		CREATE INDEX（形式 2）	△	○	○	-	○	○	○	○	○	○	○	IDX

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
		CREATE INDEX（形式 3）	△	○	○	-	○	○	○	○	○	○	○	IDX
		CREATE PROCEDURE	△	○	○	-	○	○	○	○	○	○	○	PRC
		CREATE ROLE	△	○	○	-	○	○	○	○	○	-	○	ROL
		CREATE SCHEMA	△	○	○	-	○	○	○	○	○	○	○	SCH
		CREATE TABLE	△	○	○	-	○	○	○	○	○	○	○	TBL
		CREATE TRIGGER	△	○	○	-	○	○	○	○	○	○	○	TRG
		CREATE TYPE	△	○	○	-	○	○	○	○	○	○	○	TYP
		CREATE VIEW CREATE PUBLIC VIEW	△	○	○	-	○	○	○	○	○	○	○	VIW
		CREATE SEQUENCE	△	○	○	-	○	○	○	○	○	○	○	SEQ
	DROP 系	DROP DATA TYPE	△	○	○	-	○	○	○	○	○	○	○	TYP
		DROP FUNCTION	△	○	○	-	○	○	○	○	○	○	○	FNC
		DROP INDEX	△	○	○	-	○	○	○	○	○	○	○	IDX
		DROP PROCEDURE	△	○	○	-	○	○	○	○	○	○	○	PRC

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
		DROP ROLE	△	○	○	-	○	○	○	○	○	-	○	ROL
		DROP SCHEMA	△	○	○	-	○	○	○	○	○	○	○	SCH
		DROP TABLE	△	○	○	-	○	○	○	○	○	○	○	TBL
		DROP TRIGGER	△	○	○	-	○	○	○	○	○	○	○	TRG
		DROP VIEW DROP PUBLIC VIEW	△	○	○	-	○	○	○	○	○	○	○	VIW
		DROP SEQUENCE	△	○	○	-	○	○	○	○	○	○	○	SEQ

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 2）											
			サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
オブジェクト操作イベント	操作	CALL	△	△	○	-	○	○	○	○	○	○	○	PRC
		DELETE（静的 SQL）	△	○	○	-	○	○	○	○	○	○	○	FTB
			△	○	○	-	○	○	○	○	○	○	○	TBL
			△	○	○	-	○	○	○	○	○	○	○	VIW
			△	○	○	-	○	○	○	○	○	○	○	ALS
		EXECUTE	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
		EXECUTE IMMEDIATE		指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		INSERT（静的 SQL）		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	SEQ
		LOCK TABLE（共用）（静的 SQL）		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
		LOCK TABLE（排他）（静的 SQL）		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
		OPEN	SELECT※2	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	LST
		CLOSE	SELECT	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	LST
		PREPARE	ASSIGN LIST	△	○	○	-	○	○	○	○	○	○	○	LST

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
			（形式 1，形式 2）												
			DELETE	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
			INSERT	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	SEQ
			LOCK TABLE（共用）	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
			LOCK TABLE（排他）	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
			UPDATE	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	SEQ

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
オブジェクト操作イベント	操作	PREPARE	PURGE TABLE	△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	ALS
			SELECT※2	△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	LST
		PURGE TABLE（静的 SQL）		△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	ALS
		UPDATE（静的 SQL）		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
				△	○	○	-	○	○	○	○	○	○	○	SEQ
		SELECT，1 行 SELECT 文（静的 SQL）※2		△	○	○	-	○	○	○	○	○	○	○	FTB
				△	○	○	-	○	○	○	○	○	○	○	TBL
				△	○	○	-	○	○	○	○	○	○	○	VIW
				△	○	○	-	○	○	○	○	○	○	○	ALS
ユティリティ操作イベント	生成	pdload※3	表単位実行	-	-	○	-	○	-	-	-	-	○	○	TBL
				-	-	○	-	○	-	-	-	-	○	○	SEQ
			RD エリア単位実行	-	-	○	-	○	-	-	-	-	○	○	TBL
				-	-	○	-	○	-	-	-	-	○	○	SEQ
		pdexp pddefrev	表単位実行	-	-	○	-	○	-	-	-	-	○	○	TBL
			ビュー表単位実行	-	-	○	-	○	-	-	-	-	○	○	VIW

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 2）												
				サービス名称	IPアドレス	プロセスID	スレッドID	ホスト名	ユニット識別子	サーバ名称	コネクト通番	SQL通番	オブジェクトの所有者※1	オブジェクト名称	オブジェクト種別
			別名表単位実行	-	-	○	-	○	-	-	-	-	○	○	ALS
			プロシジャ単位実行	-	-	○	-	○	-	-	-	-	○	○	PRC
			トリガ単位実行	-	-	○	-	○	-	-	-	-	○	○	TRG
	運用	pdrorg	スキーマ，表単位実行	-	-	○	-	○	-	-	-	-	○	○	TBL
			インデクス単位実行	-	-	○	-	○	-	-	-	-	○	○	TBL
			RD エリア単位実行	-	-	○	-	○	-	-	-	-	○	○	TBL
			ディクショナリ表のメンテナンス（インデクスの追加）	-	-	○	-	○	-	-	-	-	○	○	IDX
		pdconstck		-	-	○	-	○	-	-	-	-	○	○	TBL

●イベント終了時の監査証跡のレコード項目（その 3）

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
システム管理者セキュリティイベント	HiRDB の開始	pdstart	-	-	○	E	○	-	-	-	-	-	-	-
	HiRDB の終了	pdstop	-	-	-	E	○	-	-	-	-	-	-	-
	主監査人の登録	pdmod	AUD	○	-	E	○	-	-	-	-	-	-	-
	副監査人の登録	pdmod	AUD	○	-	E	○	-	-	-	-	-	-	-
	監査証跡表の作成	pdmod	-	-	-	E	○	-	-	-	-	-	○	-
	監査証跡ファイルの削除	pdaudrm	-	-	-	E	○	-	-	-	-	-	-	-
	監査証跡の取得開始	pdaudbegin	-	-	-	E	○	-	-	-	-	-	-	-
		HiRDB の開始	-	-	-	E	○	-	-	-	-	-	-	-
	監査証跡の取	pdaudend	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除/変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
	得 終了	HiRDB の終了	-	-	-	E	○	-	-	-	-	-	-	-
	監査証跡ファイルの上書き	システムによる自動上書き	-	-	-	E	○	○	○	-	-	-	-	-
		pdaudswap による手動上書き	-	-	-	E	○	○	○	-	-	-	-	-
	連続認証失敗アカウントロック状態への遷移	CONNECT 失敗	-	○	-	E	○	-	-	-	-	-	-	-
	連続認証失敗アカウントロック状態の解除	アカウントロック期間経過後の CONNECT	-	○	-	E	○	-	-	-	-	-	-	-
		DROP CONNECTION SECURITY FOR CONNECT	-	○	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除/変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		pdacunlck	-	○	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その3）											
			付与/削除/変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
システム管理者セキュリティイベント	パスワード無効アカウントロック状態への遷移		-	○	-	E	○	-	-	-	-	-	-	-
	パスワード無効アカウントロック状態の解除		-	○	-	E	○	-	-	-	-	-	-	-
	CONNECT関連セキュリティ機能の設定値変更	CREATE/DROP CONNECTION SECURITY	-	○	-	E	○	-	-	○	○	○	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
	pdacunlck の実行		-	-	-	E	○	-	-	-	-	-	-	-
監査人セキュリティイベント	監査証跡表へのデータロード	pdload（監査人によるデータロード）	-	-	-	E	○	-	-	-	-	-	○	○
		pdload（監査証跡表の自動データロード機能によるデータロード）	-	-	-	E	○	-	-	-	-	-	○	○
	監査証跡ファイルのスワップ	pdaudswap	-	-	-	E	○	-	-	-	-	-	-	-
	監査対象イベントの定義	CREATE AUDIT	-	-	-	E	○	-	-	-	-	-	-	-
	監査対象イベントの削除	DROP AUDIT	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除/変更した権限	権限を付与/削除/変更されたユーザ識別子 又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
	監査人のパスワード変更	GRANT AUDIT	AUD	○	-	E	○	-	-	○	-	-	-	-
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	-	-	-	E	○	-	-	-	-	-	-	-
	監査証跡表の自動データロード機能の停止	pdaudatld -t	-	-	-	E	○	-	-	-	-	-	-	-
		データロード失敗に伴う停止	-	-	-	E	○	-	-	-	-	-	-	-
	JP1/NETM/Audit用監査ログの出力	pdaudput	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子 又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
セッションセキュリティイベント	HiRDBへの接続	CONNECT	-	-	-	E	○	-	-	-	-	-	-	-
	ユーザの変更	SET SESSION AUTHORIZATION	-	-	-	E	○	-	-	-	-	-	-	-
	現在ロールの設定	SET ROLE	-	-	-	E	○	-	-	-	-	-	-	-
	HiRDBとの切り離し	DISCONNECT COMMIT RELEASE ROLLBACK RELEASE	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）										
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
権限管理イベント	権限付与	GRANT DBA	DBA	○	-	E	○	-	-	○	-	-	-	-
		GRANT SCHEMA	SCH	○	-	E	○	-	-	-	-	-	-	-
		GRANT SCHEMA OPERATION	SCO	○	-	E	○	-	-	-	-	-	-	-
		GRANT CONNECT	CNT	○	-	E	○	-	-	○	-	-	-	-
		GRANT RDAREA	RDA	○	-	E	○	-	-	-	-	-	-	-
		GRANT アクセス権限	○	○	-	E	○	-	-	-	-	-	-	-
			○	○	-	E	○	-	-	-	-	-	○	-
			○	○	-	E	○	-	-	-	-	-	○	-
		GRANT ロール利用権限	ROL	○	-	E	○	-	-	-	-	-	-	-
	権限削除	REVOKE DBA	DBA	○	-	E	○	-	-	-	-	-	-	-
		REVOKE SCHEMA	SCH	○	-	E	○	-	-	-	-	-	-	-
		REVOKE SCHEMA OPERATION	SCO	○	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）										
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		REVOKE CONNECT	CNT	○	-	E	○	-	-	-	-	-	-	-
		REVOKE RDAREA	RDA	○	-	E	○	-	-	-	-	-	-	-
		REVOKE アクセス権限	○	○	-	E	○	-	-	-	-	-	-	-
			○	○	-	E	○	-	-	-	-	-	○	-
			○	○	-	E	○	-	-	-	-	-	○	-
		REVOKE ロール利用権限	ROL	○	-	E	○	-	-	-	-	-	-	-
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	-	-	-	E	○	-	-	-	-	-	-	-
		ALTER ROUTINE	-	-	-	E	○	-	-	-	-	-	-	-
		ALTER TABLE（ADD RDAREA, CHANGE ADAREA 以外）	-	-	-	E	○	-	-	-	-	-	○	-
		ALTER TABLE ADD RDAREA ALTER TABLE CHANGE RDAREA	-	-	-	E	○	-	-	-	-	-	○	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		ALTER TRIGGER	-	-	-	E	○	-	-	-	-	-	-	-
		ALTER INDEX	-	-	-	E	○	-	-	-	-	-	-	-
		COMMENT	-	-	-	E	○	-	-	-	-	-	○	-
			-	-	-	E	○	-	-	-	-	-	○	-
			-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト定義イベント	CREATE 系	CREATE FUNCTION	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE INDEX（形式 1）	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE INDEX（形式 2）	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子 又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		CREATE INDEX（形式 3）	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE PROCEDURE	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE ROLE	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE SCHEMA	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE TABLE	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE TRIGGER	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE TYPE	-	-	-	E	○	-	-	-	-	-	-	-
		CREATE VIEW CREATE PUBLIC VIEW	-	-	-	E	○	-	-	-	-	-	○	-
		CREATE SEQUENCE	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト定義イベント	DROP系	DROP DATA TYPE	-	-	-	E	○	-	-	-	-	-	-	-
		DROP FUNCTION	-	-	-	E	○	-	-	-	-	-	-	-
		DROP INDEX	-	-	-	E	○	-	-	-	-	-	-	-
		DROP PROCEDURE	-	-	-	E	○	-	-	-	-	-	-	-
		DROP ROLE	-	-	-	E	○	-	-	-	-	-	-	-
		DROP SCHEMA	-	-	-	E	○	-	-	-	-	-	-	-
		DROP TABLE	-	-	-	E	○	-	-	-	-	-	○	-
		DROP TRIGGER	-	-	-	E	○	-	-	-	-	-	-	-
		DROP VIEW DROP PUBLIC VIEW	-	-	-	E	○	-	-	-	-	-	○	-
		DROP SEQUENCE	-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 3）											
			付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト操作イベント	操作	CALL	-	-	-	E	○	-	-	-	-	-	-	-
		DELETE（静的 SQL）	-	-	-	E	○	-	-	-	-	-	-	○
			-	-	-	E	○	-	-	-	-	-	○	○
			-	-	-	E	○	-	-	-	-	-	○	○
			-	-	-	E	○	-	-	-	-	-	-	○
		EXECUTE	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。											
		INSERT（静的 SQL）	-	-	-	E	○	-	-	-	-	-	-	○
			-	-	-	E	○	-	-	-	-	-	○	○
			-	-	-	E	○	-	-	-	-	-	○	○
			-	-	-	E	○	-	-	-	-	-	-	○
			-	-	-	E	○	-	-	-	-	-	-	-
		LOCK TABLE（共用）（静的 SQL）	-	-	-	E	○	-	-	-	-	-	-	-
			-	-	-	E	○	-	-	-	-	-	○	-
			-	-	-	E	○	-	-	-	-	-	○	-
			-	-	-	E	○	-	-	-	-	-	-	-
		LOCK TABLE（排他）（静的 SQL）	-	-	-	E	○	-	-	-	-	-	-	-
			-	-	-	E	○	-	-	-	-	-	○	-
			-	-	-	E	○	-	-	-	-	-	○	-
			-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）												
				付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
		OPEN	SELECT※2	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	-	-
		CLOSE	SELECT	-	-	-	E	○	-	-	-	-	-	-	○
				-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	-	○
				-	-	-	E	○	-	-	-	-	-	-	○
		PREPARE	ASSIGN LIST（形式 1，形式 2）	-	-	-	E	○	-	-	-	-	-	○	-
			DELETE	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
			INSERT	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）												
				付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
				-	-	-	E	○	-	-	-	-	-	-	-
			LOCK TABLE（共用）	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
			LOCK TABLE（排他）	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
			UPDATE	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	-	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）												
				付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
オブジェクト操作イベント	操作	PREPARE	PURGE TABLE	-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
			SELECT※2	-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
				-	-	-	E	○	-	-	-	-	-	○	-
		PURGE TABLE（静的 SQL）		-	-	-	E	○	-	-	-	-	-	○	-
				-	-	-	E	○	-	-	-	-	-	-	-
		UPDATE（静的 SQL）		-	-	-	E	○	-	-	-	-	-	-	○
				-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	-	○
				-	-	-	E	○	-	-	-	-	-	-	-
		SELECT，1 行 SELECT 文（静的 SQL）※2		-	-	-	E	○	-	-	-	-	-	-	○
				-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	-	○
ユティリティ操作イベント	生成	pdload※3	表単位実行	-	-	-	E	○	-	-	-	-	-	○	○
				-	-	-	E	○	-	-	-	-	-	-	-
			RD エリア	-	-	-	E	○	-	-	-	-	-	○	○

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）											
				付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
			単位実行	-	-	-	E	○	-	-	-	-	-	-	-
		pdexp pddefrev	表単位実行	-	-	-	E	○	-	-	-	-	-	○	-
			ビュー表単位実行	-	-	-	E	○	-	-	-	-	-	○	-
			別名表単位実行	-	-	-	E	○	-	-	-	-	-	-	-
			プロシジャ単位実行	-	-	-	E	○	-	-	-	-	-	-	-
			トリガ単位実行	-	-	-	E	○	-	-	-	-	-	-	-
	運用	pdrorg	スキーマ，表単位実行	-	-	-	E	○	-	-	-	-	-	○	○
			インデクス単	-	-	-	E	○	-	-	-	-	-	○	-

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 3）											
				付与/削除変更した権限	権限を付与/削除/変更されたユーザ識別子又はイベント対象ユーザ識別子	監査関連オペランドの定義値	監査証跡種別	SQLコード／終了コード	スワップ元監査証跡ファイル名称	スワップ先監査証跡ファイル名称	セキュリティ機能の変更種別	セキュリティ機能変更前設定値	セキュリティ機能変更後設定値	監査証跡表オプション	アクセス件数
			位実行												
			RD エリア単位実行	-	-	-	E	○	-	-	-	-	-	○	○
			ディクショナリ表のメンテナンス（インデクスの追加）	-	-	-	E	○	-	-	-	-	-	-	-
		pdconstck		-	-	-	E	○	-	-	-	-	-	○	○

●イベント終了時の監査証跡のレコード項目（その 4）

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
システム管理者セキュリティイベント	HiRDB の開始	pdstart	-	-	-	-	-	-	○	○	1	-	pdstart コマンドを実行したユニット
	HiRDB の終了	pdstop	-	-	-	-	-	-	○	○	1	-	pdstop コマンドを実行したユニット
	主監査人の登録	pdmod	-	-	-	-	-	-	○	○	aud+atb	-	DS
	副監査人の登録	pdmod	-	-	-	-	-	-	○	○	aud+atb	-	DS
	監査証跡表の作成	pdmod	-	-	-	-	-	-	○	○	aud+atb	-	DS
	監査証跡ファイルの削除	pdaudrm	-	-	-	-	-	-	○	○	1	-	MGR
	監査証跡の取得開始	pdaudbegin	-	-	-	-	-	-	○	○	1	-	各ユニット
		HiRDB の開始	-	-	-	-	-	-	○	○			各起動ユニットごと
	監査証跡の取得終了	pdaudend	-	-	-	-	-	-	○	○	1	-	各ユニット
		HiRDB の終了	-	-	-	-	-	-	○	○			各停止ユニットごと
	監査証跡ファイルの上書き	システムによる自動上書き	-	-	-	-	-	-	○	○	1	-	各ユニット

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		pdaudswap による手動上書き	-	-	-	-	-	-	○	○			
	連続認証失敗アカウントロック状態への遷移	CONNECT失敗	-	-	-	-	-	-	○	○	1	-	FES
	連続認証失敗アカウントロック状態の解除	アカウントロック期間経過後のCONNECT	-	-	-	-	-	-	○	○	1	-	FES
		DROP CONNECTION SECURITY FOR CONNECT	-	-	○	○	○	○	○	○	usr	-	DS
		pdacunlck	-	-	-	-	-	-	○	○	usr	-	DS

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
システム管理者セキュリティイベント	パスワード無効アカウントロック状態への遷移		-	-	○	○	○	○	○	○	usr	-	DS
	パスワード無効アカウントロック状態の解除		-	-	○	○	○	○	○	○	usr	-	DS
	CONNECT 関連セキュリティ機能の設定値変更	CREATE/DROP CONNECTION SECURITY	○	-	○	○	○	○	○	○	spr	-	FES
	pdacunlck の実行		-	-	-	-	-	-	○	○	1	-	DS
監査人セキュリティイベント	監査証跡表へのデータロード	pdload（監査人によるデータロード）	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
		pdload（監査証跡表の自動データロード機能によるデータロード）	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
	監査証跡ファイルのスワップ	pdaudswap	-	-	-	-	-	-	○	○	1	-	MGR

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
	監査対象イベントの定義	CREATE AUDIT	○	-	○	○	○	○	○	○	1	-	FES
	監査対象イベントの削除	DROP AUDIT	○	-	○	○	○	○	○	○	1	-	FES
	監査人のパスワード変更	GRANT AUDIT	○	-	○	○	○	○	○	○	1	-	FES
	監査証跡表の自動データロード機能の再開始	pdaudatld -b	-	-	-	-	-	-	○	○	1	-	MGR
	監査証跡表の自動データロード機能の停止	pdaudatld -t	-	-	-	-	-	-	○	○	1	-	MGR
		データロード失敗に伴う停止	-	-	-	-	-	-	○	○	1	-	MGR
	JP1/NETM/Audit 用監査ログの出力	pdaudput	-	-	-	-	-	-	○	○	1	-	MGR
セッションセキュリティイベント	HiRDB への接続	CONNECT	-	-	-	-	-	-	-	-	1	-	FES
	ユーザの変更	SET SESSION	-	-	-	-	-	-	-	-	1	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		AUTHORIZATION											
	現在ロールの設定	SET ROLE	-	-	○	○	○	○	○	○	1	-	FES
	HiRDB との切り離し	DISCONNECT COMMIT RELEASE ROLLBACK RELEASE	-	-	-	-	-	-	○	○	1	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
権限管理イベント	権限付与	GRANT DBA	○	-	○	○	○	○	○	○	usr	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		GRANT SCHEMA	○	-	○	○	○	○	○	○	usr	-	FES
		GRANT SCHEMA OPERATION	○	-	○	○	○	○	○	○	usr	-	FES
		GRANT CONNECT	○	-	○	○	○	○	○	○	usr	-	FES
		GRANT RDAREA	○	-	○	○	○	○	○	○	usr × rda	-	FES
		GRANT アクセス権限	○	-	○	○	○	○	○	○	usr × acs	-	FES
			○	-	○	○	○	○	○	○			
			○	-	○	○	○	○	○	○			
		GRANT ロール利用権限	○	-	○	○	○		○	○	rll × rlr	-	FES
	権限削除	REVOKE DBA	○	-	○	○	○	○	○	○	usr	-	FES
		REVOKE SCHEMA	○	-	○	○	○	○	○	○	usr	REVOKE SCHEMA OPERATION	FES
		REVOKE SCHEMA OPERATION	○	-	○	○	○	○	○	○	usr	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		REVOKE CONNECT	○	-	○	○	○	○	○	○	usr	-	FES
		REVOKE RDAREA	○	-	○	○	○	○	○	○	usr × rda	-	FES
		REVOKE アクセス権限	○	-	○	○	○	○	○	○	usr × acs	DROP VIEW	FES
			○	-	○	○	○	○	○	○			
			○	-	○	○	○	○	○	○			
		REVOKE ロール利用権限	○	-	○	○	○		○	○	rll × rlr	-	FES
オブジェクト定義イベント	ALTER 系	ALTER PROCEDURE	○	-	○	○	○	○	○	○	prc	-	FES
		ALTER ROUTINE	○	-	○	○	○	○	○	○	1	-	FES
		ALTER TABLE（ADD RDAREA,CHANGE ADAREA 以外）	○	-	○	○	○	○	○	○	tbl	DROP VIEW，DROP INDEX，DROP TRIGGER，DROP PUBLIC VIEW	FES
		ALTER TABLE ADD RDAREA	○	-	○	○	○	○	○	○	tbl	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがある SQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		ALTER TABLE CHANGE RDAREA											
		ALTER TRIGGER	○	-	○	○	○	○	○	○	1	-	FES
		ALTER INDEX	○	-	○	○	○	○	○	○	1		FES
		COMMENT	○	-	○	○	○	○	○	○	1	-	FES
			○	-	○	○	○	○	○	○			
			○	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト定義イベント	CREATE 系	CREATE FUNCTION	○	-	○	○	○	○	○	○	1	-	FES
		CREATE INDEX（形式 1）	○	-	○	○	○	○	○	○	1	-	FES
		CREATE INDEX（形式 2）	○	-	○	○	○	○	○	○	1	-	FES
		CREATE INDEX（形式 3）	○	-	○	○	○	○	○	○	1	-	FES
		CREATE PROCEDURE	○	-	○	○	○	○	○	○	1	-	FES
		CREATE ROLE	○	-	○	○	○		○	○	1	-	FES
		CREATE SCHEMA	○	-	○	○	○	○	○	○	1	-	FES
		CREATE TABLE	○	-	○	○	○	○	○	○	1	CREATE TRIGGER（制約動作を行うためのトリガを生成する場合）	FES
		CREATE TRIGGER	○	-	○	○	○	○	○	○	1	-	FES
		CREATE TYPE	○	-	○	○	○	○	○	○	1	• CREATE PROCEDU	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
												RE（メンバに手続きがある場合） • CREATE FUNCTION •（メンバに関数がある場合）	
		CREATE VIEW CREATE PUBLIC VIEW	○	-	○	○	○	○	○	○	1	-	FES
		CREATE SEQUENCE	○	-	○	○	○	○	○	○	1	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト定義イベント	DROP 系	DROP DATA TYPE	○	-	○	○	○	○	○	○	1	DROP FUNCTION，DROP PROCEDURE	FES
		DROP FUNCTION	○	-	○	○	○	○	○	○	1	DROP DATA TYPE	FES
		DROP INDEX	○	-	○	○	○	○	○	○	1	-	FES
		DROP PROCEDURE	○	-	○	○	○	○	○	○	1	DROP DATA TYPE	FES
		DROP ROLE	○	-	○	○	○		○	○	1	-	FES
		DROP ROLE	○	-	○	○	○		○	○	1	-	
		DROP SCHEMA	○	-	○	○	○	○	○	○	1	DROP TABLE，DROP VIEW，DROP FOREIGN TABLE，DROP INDEX，DROP FOREIGN INDEX，DROP PROCEDURE，	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
												DROP FUNCTION， DROP DATA TYPE， DROP TRIGGER， DROP PUBLIC VIEW， DROP PUBLIC FUNCTION， DROP PUBLIC PROCEDURE， DROP SEQUENCE， REVOKE SCHEMA OPERATION	
		DROP TABLE	○	-	○	○	○	○	○	○	1	DROP VIEW， DROP INDEX， DROP TRIGGER， DROP PROCEDURE， DROP PUBLIC VIEW	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		DROP TRIGGER	○	-	○	○	○	○	○	○	1	DROP PROCEDURE	FES
		DROP VIEW DROP PUBLIC VIEW	○	-	○	○	○	○	○	○	1	DROP VIEW， DROP PUBLIC VIEW	FES
		DROP SEQUENCE	○	-	○	○	○	○	○	○	1	-	FES

イベントの種類		コマンド／SQL 種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL 文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL 文	SQL データ	ユーザ付加情報 1	ユーザ付加情報 2	ユーザ付加情報 3	関連製品付加情報 1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト操作イベント	操作	CALL	○	○	○	○	○	○	○	○	1	-	FES
		DELETE（静的 SQL）	○	○	○	○	○	○	○	○	1	SELECT，	FES
			○	○	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別	証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
			SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			○	○	○	○	○	○	○	○		トリガSQL文	
			○	○	○	○	○	○	○	○			
		EXECUTE	前処理した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。									前処理したSQL	FES
		EXECUTE IMMEDIATE	指定した SQL に依存します。操作系，制御系 SQL は各 SQL の PREPARE 時と同じ監査証跡を出力します。定義系 SQL はオブジェクト定義イベントの各定義系 SQL と同じ監査証跡を出力します。									指定したSQL	FES
		INSERT（静的 SQL）	○	○	○	○	○	○	○	○	1	SELECT，トリガSQL文	FES
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○	seq		
		LOCK TABLE（共用）（静的 SQL）	○	○	○	○	○	○	○	○	tbl	-	FES
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○			
		LOCK TABLE（排他)(静的 SQL）	○	○	○	○	○	○	○	○	tbl	-	FES
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○			
			○	○	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
		OPEN	SELECT※2	○	○	○	○	○	○	○	○	tbl	-	FES
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○	1		
		CLOSE	SELECT	○	○	○	○	○	○	○	○	tbl	-	FES
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○	1		
		PREPARE	ASSIGN LIST（形式1，形式2）	○	-	○	○	○	○	○	○	1	-	FES
			DELETE	○	-	○	○	○	○	○	○		PREPARE（SELECT）	
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
			INSERT	○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○	seq		
			LOCK TABLE（共用）	○	-	○	○	○	○	○	○	tbl	-	
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
			LOCK TABLE（排他）	○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
			UPDATE	○	-	○	○	○	○	○	○	1	PREPARE（SELECT）	
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○	seq		

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
オブジェクト操作イベント	操作	PREPARE	PURGE TABLE	○	-	○	○	○	○	○	○	1	-	FES
				○	-	○	○	○	○	○	○			
			SELECT※2	○	-	○	○	○	○	○	○	tbl	PREPARE（SELECT）	
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○			
				○	-	○	○	○	○	○	○	1		
		PURGE TABLE（静的 SQL）		○	-	○	○	○	○	○	○	1	-	FES
				○	-	○	○	○	○	○	○			
		UPDATE(静的 SQL)		○	○	○	○	○	○	○	○	1	SELECT，トリガSQL文	FES
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○	seq		
		SELECT，1 行 SELECT 文（静的 SQL）※2		○	○	○	○	○	○	○	○	tbl	-	FES
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
				○	○	○	○	○	○	○	○			
ユティリティ操作イベント	生成	pdload※3	表単位実行	-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先
				-	-	-	-	-	-	○	○	seq		

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			RD エリア単位実行	-	-	-	-	-	-	○	○	1		ユニット（HiRDB/パラレルサーバ限定）」を参照
				-	-	-	-	-	-	○	○	seq		
		pdexp pddefrev	表単位実行	-	-	-	-	-	-	○	○	tbl	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照
			ビュー表単位実行	-	-	-	-	-	-	○	○	viw		
			別名表単位実行	-	-	-	-	-	-	○	○	als		
			プロシジャ単位実行	-	-	-	-	-	-	○	○	prc		
			トリガ単位実行	-	-	-	-	-	-	○	○	trg		
	運用	pdrorg	スキーマ，表単位実行	-	-	-	-	-	-	○	○	tbl	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラ

イベントの種類		コマンド／SQL種別		証跡レコード項目（その 4）								記録レコード数	延長で実行することがあるSQL文	出力先ユニット（HiRDB/パラレルサーバの場合）
				SQL文	SQLデータ	ユーザ付加情報1	ユーザ付加情報2	ユーザ付加情報3	関連製品付加情報1	監査証跡表へのデータ登録時刻印	監査証跡表へのデータ登録時の通番			
			インデクス単位実行	-	-	-	-	-	-	○	○			レルサーバ限定）」を参照
			RD エリア単位実行	-	-	-	-	-	-	○	○			
			ディクショナリ表のメンテナンス（インデクスの追加）	-	-	-	-	-	-	○	○	idx		
		pdconstck		-	-	-	-	-	-	○	○	1	-	「ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）」を参照

（凡例）

○：情報を取得します。

△：XA インタフェース経由で情報を取得することがあります。

−：情報を取得しません。又は該当しません。

‘△△△’：空白を記録します。

acs：指定した権限の数（all を指定した場合は 4）

atb：pdmod コマンドで作成する監査証跡表の数（作成する場合は 1，作成しない場合は 0）

aud：pdmod コマンドで作成する監査人の数（作成する場合は 1，作成しない場合は 0）

cls：FOR CLUSTER KEY 句を指定した場合は 1，指定しない場合は 0

fnc：再作成の対象となる関数の数

idx：関連するインデクス数

prc：再作成の対象になるプロシジャ数

prm：FOR PRIMARY KEY 句を指定した場合は 1，指定しない場合は 0

rda：指定した RD エリア数

rlb：LOB データ格納 RD エリア数

rid：インデクス格納 RD エリア数

rll：ロール利用権限を付与又は削除するロール名の指定数

rlr：ロール利用権限を付与又は削除される認可識別子の指定数

spr：CONNECT 関連セキュリティ機能の設定値の変更数

seq：使用する順序数生成子数

tbl：SELECT 文，SELECT 句，又は LOCK 文で指定した表数，又はユティリティの処理対象表数

trg：再作成の対象になるトリガ数

usr：指定ユーザ数又は指定グループ数

viw：対象となるビュー表数

注※1

対象オブジェクトがパブリックビュー表，パブリック関数，又はパブリック手続の場合，オブジェクトの所有者には PUBLIC が記録されます。

注※2

WITH 句がある場合，オブジェクト種別は次のように出力されます。

```
WITH Q1(C1) AS (SELECT C1 FROM AAA) SELECT C1 FROM BBB
```

種別	AAA	BBB
実表	TBL	TBL
ビュー表	VIW	VIW
問合せ名	出力しない	出力しない

注　種別が判別する前にエラーとなった場合はすべて TBL で出力されます。

注※3

監査証跡表以外又は監査証跡表と認識できない場合が該当します。

注※4

監査証跡表の自動データロード機能を適用して，HiRDB が pdload を実行した場合は記録されません。

24.15　ユティリティ実行時の監査証跡の出力先ユニット（HiRDB/パラレルサーバ限定）

ユティリティ実行時に出力される監査証跡の出力先ユニットについて説明します。ユティリティ実行時の監査証跡の出力先ユニットを次の表に示します。

表 24–40　ユティリティ実行時の監査証跡の出力先ユニット（その 1）

ユティリティ実行時の条件		監査証跡の出力先ユニット
正常終了		監査証跡の出力先ユニットについては表「ユティリティ実行時の監査証跡の出力先ユニット（その２）」を参照してください。
異常終了	ユティリティサーバの起動後	
	ユティリティサーバの起動前	コマンドを入力したユニットに監査証跡が出力されます。

表 24–41　ユティリティ実行時の監査証跡の出力先ユニット（その２）

ユティリティの種類			実行対象オブジェクト			出力先ユニット	
						イベント 終了時	権限 チェック時
pdload			表単位	基表だけ		入力ファイルがあるユニット	コマンド入力ユニット
				基表＋ BLOB			
				BLOB だけ			
			RD エリア単位	基表だけ			
				基表＋ BLOB			
				BLOB だけ			
pdrorg	-k rorg -k unld -k reld	-g オプションあり	−			アンロードファイルを作成するユニット	コマンド入力ユニット
		-g オプションなし	ディクショナリ表の再編成			ディクショナリサーバがあるユニット	
			ユーザ表の再編成	表の再編成		アンロードファイルを作成するユニット	
				横分割表	RD エリア単位の再編成		
					表単位の再編成	最初に指定した unload 又は lobunld 文に指定したサーバがあるユニット	
	-k ixrc		−			コマンド入力ユニット	
	-k ixor		−				
	-k ixmk		−				

ユティリティの種類		実行対象オブジェクト	出力先ユニット	
			イベント 終了時	権限 チェック時
	-k maintenance	ディクショナリ表のメンテナンス		
pdexp， pddefrev		－	搬出ファイルがあるユニット	搬出ファイルがあるユニット
pdconstck		－	コマンド入力ユニット	コマンド入力ユニット

（凡例）－：該当しません。

24.16　バージョンアップ時の注意事項

セキュリティ監査機能使用時のバージョンアップの注意事項を説明します。バージョンアップ方法については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

24.16.1　監査証跡ファイルの監査証跡形式が変更になる場合

バージョンアップによって監査証跡ファイルの監査証跡形式が変更になる場合，バージョンアップをする前に次に示すことをしてください。

- **pdaudswap コマンドで監査証跡ファイルをスワップし，スワップ前のファイルの内容を pdload コマンドで監査証跡表に登録してください。**

> **参考**
>
> HiRDB の終了時に監査証跡を出力し，バージョンアップ後の HiRDB の開始までに監査証跡ファイルをスワップする契機がないため，必ず旧形式の HiRDB 終了時の監査証跡と，新形式の HiRDB 開始時の監査証跡が同じファイルに混在します。このため，旧形式の監査証跡と新形式の監査証跡が混在していても，監査証跡表へのデータロードはできるようになっています。

24.16.2　監査証跡表の列構成が変更になる場合

バージョンアップによって監査証跡表に列が追加されることがあります。監査証跡表の列構成の変更は pdvrup コマンドの実行時に HiRDB が行います。このとき，監査証跡表に既存のデータが存在する場合は，追加された列には NULL 値が設定されます。

また，列の追加に伴い，監査証跡表の行長が RD エリアのページ長を超える場合があります。監査証跡表の検索性能が低下する原因になりますので，監査証跡表の行長を見積もり，監査証跡表を格納する RD エリアのページ長を大きくすることをお勧めします。ページ長の設計については，「監査証跡表を格納する RD エリアの容量見積もり」，及びマニュアル「HiRDB システム導入・設計ガイド」を参照してください。

24.16.3　監査証跡のイベントが増えた場合

バージョンアップによって監査証跡のイベントが増えた場合，出力される監査証跡量が増えるため，バージョンアップをする前に監査証跡ファイルがあるディスク容量の再見積もりをしてください。

全イベントを監査証跡の出力対象にしている場合（CREATE AUDIT FOR ANY を指定している場合）が該当します。例えば，全イベントを監査証跡の出力対象にしている場合（CREATE AUDIT FOR ANY を指定している場合），バージョンアップによって新しいイベント（ユティリティ操作イベントなど）が増えた場合，ユティリティ操作イベントについても監査証跡を取得します。

新規に追加される監査イベントの監査証跡を取得したくない場合は，監査証跡の取得を定義するときに CREATE AUDIT FOR ANY を使用しないで，個別に定義（CREATE AUDIT FOR SESSION，CREATE AUDIT FOR PRIVILEGE，...）してください。

24.16.4　バージョンアップ失敗時の注意事項

バージョンアップに成功した後に pdload コマンドを実行するようにしてください。バージョンアップに成功する前に pdload コマンドを実行すると，次に示すことが発生します。

- **07-00 以降 07-02 より前のバージョンから，07-02 以降にバージョンアップする場合**
  バージョンアップ後に HiRDB を開始すると，新しい形式の監査証跡が監査証跡ファイルに出力されます。この状態のままバージョンを戻し，pdload コマンドで監査証跡を監査証跡表に登録した場合は動作を保証できません。この場合，バージョンを戻した後に，pdfmkfs コマンドで監査証跡ファイル用の HiRDB ファイルシステム領域を初期設定してください。
- **07-02 以降からそれ以上のバージョンにバージョンアップする場合**
  07-02 以降のバージョンの場合，出力されている監査証跡の形式が該当するバージョンで対応している形式でないときは，KFPS05753-W メッセージを出力し，その監査証跡を削除します。

24.16.5　監査証跡表を格納している RD エリアのバックアップ

バージョンアップする前に，監査証跡表を格納している RD エリアのバックアップを取得してください。バージョンアップに失敗してバージョンを戻す場合は，バックアップから監査証跡表を格納している RD エリアを回復してください。

バックアップを取得していない場合は，DROP TABLE で監査証跡表を削除した後に，pdmod コマンドで監査証跡表を再作成してください。この場合，監査証跡表に格納されているデータは削除されます。

24.17　JP1/NETM/Audit との連携

実行者　監査人

JP1/NETM/Audit を使用すると，システムの監査ログ※を収集・一元管理し，長期間に渡る保管を実現できます。また，GUI で監査ログを検索，集計でき，バックアップ履歴管理などもできます。これによって，企業の内部統制の評価や監査を支援します。HiRDB が出力した監査証跡表のデータを JP1/NETM/Audit 用に加工することで，JP1/NETM/Audit と連携して，HiRDB の監査証跡，及びほかの製品の監査ログを一元管理できます。HiRDB の監査証跡を JP1/NETM/Audit 用に加工するためには，**JP1/NETM/Audit 用監査ログ出力ユティリティ**（**pdaudput** コマンド）を使用します。

注※

監査ログとは，JP1/NETM/Audit がシステム全体から収集する監査証跡の総称です。

JP1/NETM/Audit については，JP1 のマニュアルを参照してください。ただし，JP1 のバージョンによってマニュアル名が異なります。該当するバージョンに対応するマニュアルを参照してください。

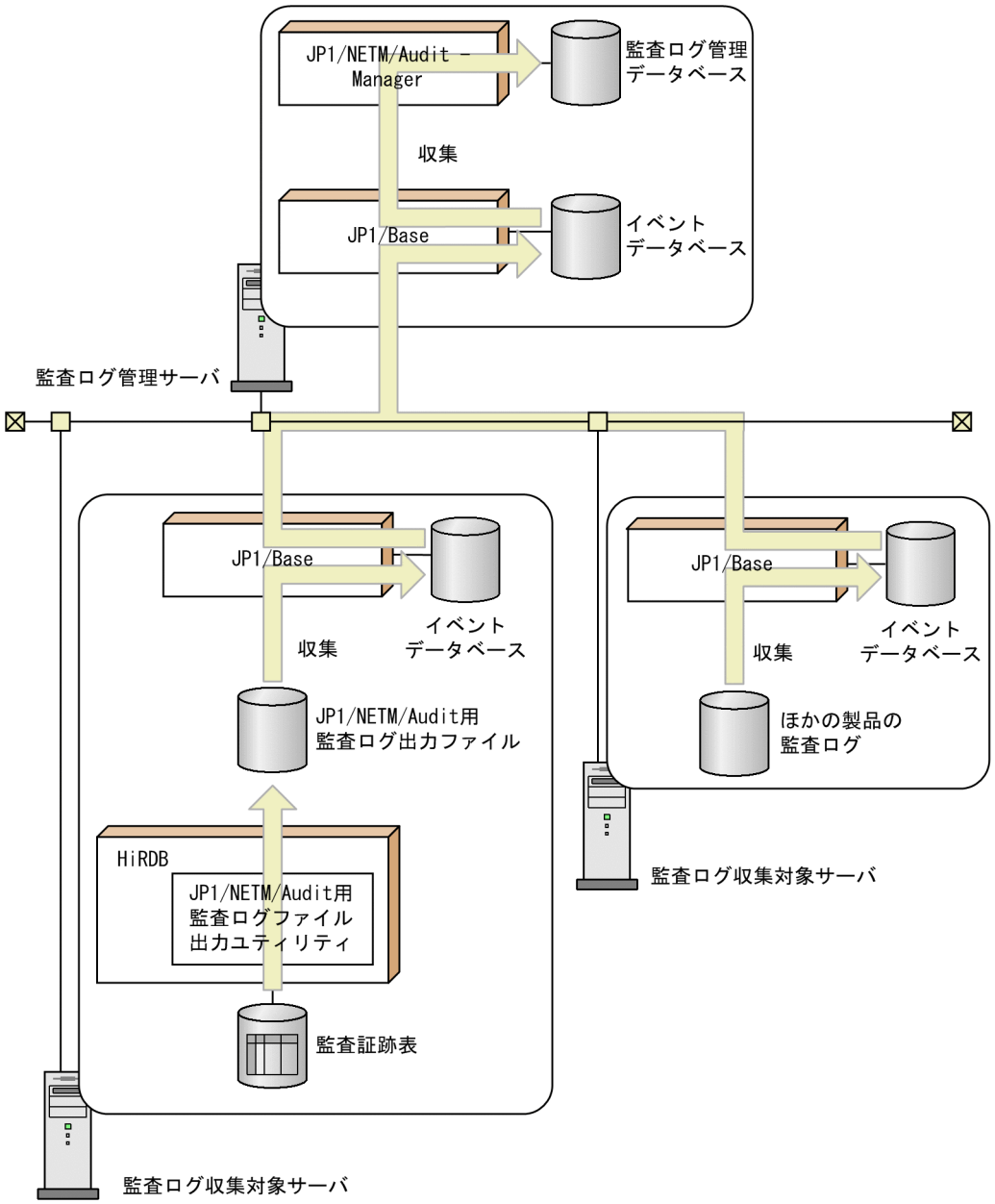
- JP1 Version 9 の場合：「JP1/NETM/Audit 構築・運用ガイド」
- JP1 Version 8 の場合：「JP1/NETM/Audit」

以降，「JP1/NETM/Audit 構築・運用ガイド」と表記します。

24.17.1　JP1/NETM/Audit との連携の概要

JP1/NETM/Audit との連携の概要を次の図に示します。

図 24–35　JP1/NETM/Audit との連携の概要



〔説明〕

監査ログ管理サーバ（JP1/NETM/Audit があるサーバマシン）は，**監査ログ収集対象サーバ**（HiRDB や監査ログを収集するほかの製品があるサーバマシン）のローカルディスクに出力した監査ログを自動収集し，一元管理します。

HiRDB がある監査ログ収集対象サーバでは，JP1/NETM/Audit 用監査ログ出力ユティリティ（pdaudput）が，監査証跡表から必要なデータを JP1/NETM/Audit 用に変換して，**JP1/NETM/Audit 用監査ログ出力ファイル**に出力します。このファイルを JP1/Base が収集し，イベントデータベースに格納します。監査ログ管理サーバの JP1/Base は監査ログ収集対象サーバの JP1/Base から監査ログを収集し，JP1/NETM/Audit に渡します。

(1)　前提プログラムと適用 OS

前提プログラムと適用 OS を次に示します。

前提プログラム

プログラム	インストールするサーバマシン
JP1/NETM/Audit - Manager	監査ログ管理サーバ
JP1/Base	監査ログ管理サーバ 監査ログ収集対象サーバ（HiRDB/パラレルサーバの場合，システムマネジャがあるサーバマシンにインストールします）

適用 OS

監査ログ収集対象サーバの適用 OS になります。詳細は，マニュアル「JP1/NETM/Audit 構築・運用ガイド」を参照してください。

(2)　pdaudput コマンド実行時の注意

- pdaudput コマンドは，JP1/NETM/Audit で監査ログの監視を開始しているときに実行してください。監視を停止しているときに JP1/NETM/Audit 用監査ログ出力ファイルを出力しても，収集されません。
- 監査証跡表の自動データロード機能を適用している場合，pdaudput コマンドを実行する前に，pdaudatld -t コマンドで自動データロードを一時的に停止してください。pdaudput コマンド実行後，pdaudatld -b コマンドで自動データロードを再開始してください。

24.17.2　環境設定

JP1/NETM/Audit と連携するには，HiRDB と JP1/NETM/Audit の両方で環境設定が必要になります。それぞれの環境設定方法について，説明します。

(1)　HiRDB での環境設定

HiRDB では，次の準備が必要になります。

1. 検索条件定義ファイルの作成
2. JP1/NETM/Audit 用監査ログ出力ファイルの出力先ディレクトリの作成

(a)　検索条件定義ファイルの作成

監査証跡表のどのデータを出力するかを指定するのが，**検索条件定義ファイル**です。検索条件定義ファイルには，JP1/NETM/Audit で管理したい監査証跡表の列名を探索条件として指定します。検索条件定義ファイルに指定された条件に合致する監査証跡表のデータが JP1/NETM/Audit 用監査ログ出力ファイルに出力されます。

サンプルファイル

HiRDB は，次に示す検索条件定義ファイルのサンプルファイルを提供しています。

- HiRDB のインストールディレクトリ¥sample¥sampleAUDIT¥sampleaud1，及び sampleaud2

検索条件定義ファイルの指定を省略して pdaudput コマンドを実行すると，システム管理セキュリティイベント及び監査人セキュリティイベントに関する監査証跡だけを JP1/NETM/Audit 用監査ログ出力ファイルに出力します。これは，サンプルファイル sampleaud1 を指定した場合と同じです。

アクセス権限

検索条件定義ファイル及びファイルのディレクトリには，HiRDB 管理者に対して読み込み権限を与えてください。

(b) JP1/NETM/Audit 用監査ログ出力ファイルの出力先ディレクトリの作成

JP1/NETM/Audit 用監査ログ出力ファイルを出力するディレクトリを作成します。出力先ディレクトリの指定を省略して pdaudput コマンドを実行すると，%PDDIR%¥auditlog ディレクトリに出力します。

JP1/NETM/Audit 用監査ログ出力ファイル

pdaudput コマンドを実行すると，JP1/NETM/Audit 用監査ログ出力ファイルが 2 個作成されます（auditmsg1.log 及び auditmsg2.log）。pdaudput コマンドの実行ごとにどちらかのファイルを使用し，ファイルが満杯になるとスワップして，終了します。続きを出力する場合は，コマンドを再度実行します。

なお，HiRDB は，JP1/Base が監査ログを収集したかどうかをチェックしないで，スワップ先のファイルを上書きします。そのため，大量の監査証跡がファイルに出力されたり，JP1 で障害が発生して監査ログを収集できなかったりすると，JP1/Base が監査ログを収集する前に上書きされるおそれがあります。JP1/NETM/Audit 用監査ログ出力ファイルの容量は，余裕値を加算して見積もってください。

JP1/NETM/Audit 用監査ログ出力ファイルの容量の見積もり

JP1/NETM/Audit 用監査ログ出力ファイルの容量は，監査証跡表のどのデータを出力するかで変わります。そのため，検索条件定義ファイルで定義した探索条件を考慮して見積もります。

まず，1 回の pdaudput コマンドの実行で，JP1/NETM/Audit 用監査ログ出力ファイルに出力する監査証跡の行数を見積もり，その行数を基にファイルの容量を見積もります。次の計算式で求められます（単位はバイト）。

```
1回のpdaudputコマンドの実行で出力するデータ量
＝pdaudputコマンド実行間隔単位に出力される監査証跡の行数
  ×JP1/NETM/Audit用監査ログの長さ（約300バイト）
  ＋JP1/NETM/Audit用監査ログ出力ファイルのヘッダ情報（512バイト）
```

アクセス権限

pdaudput コマンドを実行して，JP1/NETM/Audit 用監査ログ出力ファイルを作成する場合，JP1/NETM/Audit 用監査ログ出力ファイルの所有者やアクセス権限は，上位のディレクトリの設定を引き継ぎます。

JP1/NETM/Audit 用監査ログ出力ファイル及び JP1/NETM/Audit 用監査ログ出力ファイル出力先ディレクトリには，次のようにアクセス権限を与えてください。

項目	アクセス権限を与える対象	与えるアクセス権限
JP1/NETM/Audit 用監査ログ出力ファイル	HiRDB 管理者	読み込み権限及び書き込み権限
	JP1/NETM/Audit で収集を行う OS ユーザ	読み込み権限
JP1/NETM/Audit 用監査ログ出力ファイル出力先ディレクトリ	HiRDB 管理者， JP1/NETM/Audit で収集を行う OS ユーザ	読み込み権限

(2)　JP1/NETM/Audit での環境設定

HiRDB が JP1/NETM/Audit と連携するには，JP1/NETM/Audit の監査ログ収集システムとして HiRDB を追加する必要があります。その手順を次に示します。

1. 定義ファイルを作成します。
2. JP1/NETM/Audit で監査ログの収集対象を設定します。

(a)　定義ファイルの作成

監査ログ管理サーバで次のファイルを作成してください。

- 製品定義ファイル
  ファイル名：HiRDB〔_任意の文字列〕.conf
- 動作定義ファイル
  ファイル名：admjevlog_HiRDB〔_任意の文字列〕.conf

任意の文字列は，複数の HiRDB が JP1/NETM/Audit と連携する場合にそれぞれを識別できるように指定します。なお，製品定義ファイルと動作定義ファイルは一つの HiRDB に一つずつ必要です。複数の HiRDB で一つの定義ファイルを共有したり，どちらかを省略したりしないでください。

製品定義ファイル

HiRDB 固有の値を設定する必要があるパラメータを次の表に示します。

表 24–42　HiRDB 固有の値を設定する必要があるパラメータ（製品定義ファイル）

パラメータ	説明	設定する値
AuditLogNum	JP1/NETM/Audit 用監査ログ出力ファイルの数を定義します。	2
AuditLogName	JP1/NETM/Audit 用監査ログ出力ファイル名を定義します。	次の二つを指定してください。 • AuditLogName=auditmsg1.log • AuditLogName=auditmsg2.log
RegularPattern	対応する正規化ルールファイル名を定義します。正規化ルールファイルについては，マニュアル「JP1/NETM/Audit 構築・運用ガイド」を参照してください。	admrglrule_CALFHM.conf

動作定義ファイル

HiRDB 固有の値を設定する必要があるパラメータを次の表に示します。

表 24–43　HiRDB 固有の値を設定する必要があるパラメータ（動作定義ファイル）

パラメータ	説明	設定する値
FILETYPE	JP1/NETM/Audit 用監査ログ出力ファイルの出力形式を指定します。	WRAP2
HEADSIZE	ヘッダの行数を指定します。	512
ACTDEF	指定した正規表現に一致した監査ログを，指定したイベント ID 及び重要度で JP1 イベントとして発行します。	• 重要度：任意に指定できます • イベント ID：任意に指定できます • 正規表現："^CALFHM" （HiRDB の監査証跡をすべて収集します）

サンプルファイル

HiRDB は，製品定義ファイル及び動作定義ファイルのサンプルファイルを提供しています。サンプルファイルを使用する場合は，%PDDIR%¥sample¥sampleAUDIT ディレクトリの HiRDB〔_任意の文字列〕.conf 及び admjevlog_HiRDB〔_任意の文字列〕.conf を監査ログ管理サーバにコピーしてください。コピー先ディレクトリについては，マニュアル「JP1/NETM/Audit 構築・運用ガイド」を参照してください。複数の HiRDB と連携するときは，サンプルファイルの名称を変更してください。

製品定義ファイル及び動作定義ファイルのサンプルファイルの内容を次に示します。

製品定義ファイルのサンプル

```
AuditLogNum=2
AuditLogName=auditmsg1.log
AuditLogName=auditmsg2.log
RegularPattern=admrglrule_CALFHM.conf
ReadOnly=1
```

動作定義ファイルのサンプル

```
retry-times=60
retry-interval=10
FILETYPE=WRAP2
HEADSIZE=512
ACTDEF =<Information>1000 ”^CALFHM”
```

(b)　JP1/NETM/Audit で監査ログの収集対象を設定

JP1/NETM/Audit の［収集対象の設定］ダイアログで，HiRDB を収集対象に設定します。HiRDB 固有の値を設定する必要がある設定項目を次の表に示します。

表 24–44　HiRDB 固有の値を設定する必要がある設定項目

設定項目	説明	設定する値
サーバ	監査ログ収集対象サーバ名を指定します。	サーバマシン（HiRDB/パラレルサーバの場合システムマネジャがあるサーバマシン）の標準ホスト名
プログラム	収集対象の監査ログを出力するプログラムの名称を指定します。	ドロップダウンリストから，「HiRDB〔/任意の文字列〕」※を選択
ログフォルダ	収集対象の監査ログの格納先を指定します。	pdaudput コマンドの-d オプションに指定する JP1/NETM/Audit 用監査ログ出力ファイルの出力ディレクトリのパス

注※

任意の文字列は，製品定義ファイル名で指定した文字列が表示されます。

注意事項

JP1/NETM/Audit の設定値と JP1/NETM/Audit 用監査ログ出力ファイルの設定で整合性が取れていない場合，正しく連携できないことがあります。この場合，HiRDB が JP1/NETM/Audit 用監査ログ出力ファイルを出力していても，JP1/NETM/Audit は監査ログを収集できません。必ず正しく設定されていることを確認してください。

24.17.3　運用方法

運用方法には，次に示す二つの方法があります。

- 運用方法１（監査証跡表にデータを残す運用）
  pdaudput コマンド実行後も，監査証跡表にデータを残し，次回コマンド実行時は続きから出力します。
- 運用方法２（監査証跡表にデータを残さない運用）
  pdaudput コマンド実行後，出力した監査証跡は，監査証跡表から削除します。

注意事項

pdaudput コマンドを実行すると，JP1/NETM/Audit 用監査ログ出力ファイルが作成されるディレクトリに，**pdaudput 用管理ファイル**（pdaudputctrl）が作成されます。このファイルは，HiRDB が使用するため，削除しないでください。このファイルには，監査証跡表中，監査証跡をどこまで出力したかが記録されています。そのため，このファイルを削除すると，次回コマンド実行時，監査証跡を前回の続きから出力できなくなり，JP1/NETM/Audit 用監査ログ出力ファイルが初期化されます。

(1) 運用方法１（監査証跡表にデータを残す運用）

手順を次に示します。

1. 自動データロード機能を適用していない場合，pdload コマンドで監査証跡表にデータを登録します。このとき，pdload コマンドを同時実行しないでください（JP1/NETM/Audit 用監査ログ出力ファイルに正しく出力されないおそれがあります）。自動データロード機能を適用している場合は，pdaudatld -t コマンドで自動データロードを一時停止してください。

   pdload コマンドの指定例

   ```
   pdload -b -W SQL_AUDIT_TRAIL C:¥pdload¥control_file
   ```

   〔説明〕

   -b 及び-W オプションは必ず指定してください。

   表名には監査証跡表を指定します。

   制御情報ファイルの内容については次で説明します。

   制御情報ファイルの指定例

   ```
   source bes1:(uoc)
   srcuoc param='dir=C:¥secarea,file=(pdaudUNT1001.aud,pdaudUNT1002.aud)'
   ```

   〔説明〕

   source bes1:(uoc)：サーバ名（bes1）は HiRDB/パラレルサーバの場合に指定します。監査証跡ファイルがあるユニット内のサーバ名を指定してください。ユニット内のどのサーバ名を指定してもよいですが，バックエンドサーバを指定すると通信ネックが少なくなります。監査証跡ファイルが存在するユニットが影響分散スタンバイレス型系切り替え対象ユニットの場合は，そのユニットで動作中の実行系サーバの名称を指定してください。

   dir：データロード対象の監査証跡ファイルがある HiRDB ファイルシステム領域名を指定します。

   file：データロード対象の監査証跡ファイル名を指定します。

   監査証跡表にインデクスを定義した場合は idxwork 文及び sort 文を指定してください。

2. pdaudput コマンドを実行して，JP1/NETM/Audit 用監査ログ出力ファイルを出力します。

   pdaudput コマンドの指定例

   ```
   pdaudput -k r -d C:¥pdaudput¥auditlog1 -s 300000 -w C:¥pdaudput¥audcond1
   ```

   〔説明〕

   -k：JP1/NETM/Audit 用監査ログ出力ファイルを出力するため，r を指定します。

   -d：JP1/NETM/Audit 用監査ログ出力ファイルを作成するディレクトリを指定します。

   -s：JP1/NETM/Audit 用監査ログ出力ファイルのファイルサイズをキロバイト単位で指定します。

   -w：検索条件定義ファイルを指定します。

3. JP1/NETM/Audit 用監査ログファイルが満杯になり，スワップして pdaudput コマンドが終了した場合，再度 pdaudput コマンドを実行してください。

自動データロード機能を適用していた場合，pdaudatld -b コマンドで自動データロードを再開始してください。

出力した監査ログが JP1/NETM/Audit に収集されたかどうかを確認することをお勧めします。確認する方法については，「運用方法２（監査証跡表にデータを残さない運用）」の 4.を参照してください。

(2) 運用方法２（監査証跡表にデータを残さない運用）

1. 自動データロード機能を適用していない場合，pdload コマンドで監査証跡表にデータを登録します。このとき，pdload コマンドを同時実行しないでください（JP1/NETM/Audit 用監査ログ出力ファイルに正しく出力されないおそれがあります）。自動データロード機能を適用している場合は，pdaudatld -t コマンドで自動データロードを一時停止してください。

   pdload コマンドの指定例

   ```
   pdload -b -W SQL_AUDIT_TRAIL C:¥pdload¥control_file
   ```

   〔説明〕

   -b 及び-W オプションは必ず指定してください。

   表名には監査証跡表を指定します。

   制御情報ファイルの内容については次で説明します。

   制御情報ファイルの指定例

   ```
   source bes1:(uoc)
   srcuoc param='dir=C:¥secarea,file=(pdaudUNT1001.aud,pdaudUNT1002.aud)'
   ```

   〔説明〕

   source bes1:(uoc)：サーバ名（bes1）は HiRDB/パラレルサーバの場合に指定します。監査証跡ファイルがあるユニット内のサーバ名を指定してください。ユニット内のどのサーバ名を指定してもよいですが，バックエンドサーバを指定すると通信ネックが少なくなります。監査証跡ファイルが存在するユニットが影響分散スタンバイレス型系切り替え対象ユニットの場合は，そのユニットで動作中の実行系サーバの名称を指定してください。

   dir：データロード対象の監査証跡ファイルがある HiRDB ファイルシステム領域名を指定します。

   file：データロード対象の監査証跡ファイル名を指定します。

   監査証跡表にインデクスを定義した場合は idxwork 文及び sort 文を指定してください。

2. pdaudput コマンドを実行して，JP1/NETM/Audit 用監査ログ出力ファイルを出力します。

   pdaudput コマンドの指定例

   ```
   pdaudput -k r -d C:¥pdaudput¥auditlog1 -s 300000 -w C:¥pdaudput¥audcond1
   ```

   〔説明〕

   -k：JP1/NETM/Audit 用監査ログ出力ファイルを出力するため，r を指定します。

   -d：JP1/NETM/Audit 用監査ログ出力ファイルを作成するディレクトリを指定します。

   -s：JP1/NETM/Audit 用監査ログ出力ファイルのファイルサイズをキロバイト単位で指定します。

-w：検索条件定義ファイルを指定します。

3. 監査証跡表のデータをアンロードします（推奨）。

pdrorg コマンドの指定例

```
pdload -k unld -W dat -t SQL_AUDIT_TRAIL C:¥pdrorg¥control_file
```

〔説明〕

アンロードのため，-k unld を指定してください。

-W オプションは必ず指定してください。

表名には監査証跡表を指定します。

制御情報ファイルの内容については次で説明します。

制御情報ファイルの指定例

```
unload bes1:C:¥pdrorg¥pdaudunld01
```

〔説明〕

unload bes1：サーバ名（bes1）は HiRDB/パラレルサーバの場合に指定します。

C:¥pdrorg¥pdaudunld01：アンロードデータファイル名を指定します。

4. 出力した監査ログが，JP1/NETM/Audit に収集されているかどうかを確認します。

次の二つの情報を比較し，一致している場合，JP1/NETM/Audit に収集されています。

- pdaudput コマンド終了時に出力される KFPN10501-I メッセージに表示されている，最後に出力した監査証跡の情報（監査証跡表の PDLOAD_TIMESTAMP 列値及び PDLOAD_SEQNUM 列値）
- JP1/NETM/Audit で即時収集を実行し，JP1/NETM/Audit の管理画面で参照できる JP1/NETM/Audit 用監査ログの自由記述（タグ名 msg）に設定される値※

  注※

  pdaudput コマンドの場合，「aa…a，bb…b」の形式で設定されています。

  aa…a：監査証跡表の PDLOAD_TIMESTAMP 列値

  bb…b：監査証跡表の PDLOAD_SEQNUM 列値

なお，pdaudput コマンドが異常終了した場合，KFPN10501-I メッセージは出力されません。pdaudput コマンドを再度実行して，正常終了したときのメッセージを基に確認してください。

参考

この手順を確認しないで監査証跡表のデータ削除の手順に進んでしまうと，pdaudput の出力結果が保証できない場合があります。この場合，削除したデータに対応する監査ログの再収集はできなくなるため，必ずこの手順を確認してください。

5. 監査証跡表のデータを削除します。

監査証跡表（SQL_AUDIT_TRAIL）のデータを削除する SQL 実行例

```
PURGE TABLE SQL_AUDIT_TRAIL
```

6. JP1/NETM/Audit 用監査ログファイルが満杯になり，スワップして pdaudput コマンドが終了した場合，再度 pdaudput コマンドを実行してください。
   自動データロード機能を適用していた場合，pdaudatld -b コマンドで自動データロードを再開始してください。

24.17.4　JP1/NETM/Audit 用監査ログ出力ファイルに関する変更

JP1/NETM/Audit 用監査ログ出力ファイルの出力先ディレクトリや，ファイルの容量を変更するときの手順について説明します。

(1)　出力先ディレクトリの変更

まず，変更後のディレクトリを JP1/NETM/Audit に設定し，pdaudput -d コマンドの指定値を変更します。この場合，過去に使用した pdaudput 用管理ファイルや JP1/NETM/Audit 用監査ログ出力ファイルが変更後のディレクトリにあると，誤動作する可能性があるため，変更後のディレクトリに新たに pdaudput 用管理ファイルと JP1/NETM/Audit 用監査ログ出力ファイルを作成しておきます。

手順を次に示します。

1. 出力した監査ログが，JP1/NETM/Audit に収集されているかどうかを確認します。「運用方法２（監査証跡表にデータを残さない運用）」の 4.を参照してください。
2. JP1/NETM/Audit 用監査ログ出力ファイルの再見積もりをします。「JP1/NETM/Audit 用監査ログ出力ファイルの出力先ディレクトリの作成」を参照してください。
3. JP1/NETM/Audit による監視を停止します。
4. 収集された監査証跡を DELETE 文又は PURGE TABLE 文で監査証跡表から削除します。
5. 変更後のディレクトリに pdaudput 用管理ファイル及び JP1/NETM/Audit 用監査ログ出力ファイルを作成します。-d オプションに変更後のディレクトリを指定して，pdaudput -k i コマンドを実行してください。
6. JP1/NETM/Audit で，監査ログの収集対象に変更後のディレクトリを設定します。表「HiRDB 固有の値を設定する必要がある設定項目」を参照してください。
7. JP1/NETM/Audit による監視を開始します。

(2)　ファイル容量の変更

pdaudput 用管理ファイルの情報を引き継いだままで，JP1/NETM/Audit 用監査ログ出力ファイルの容量を変更するため，JP1/NETM/Audit 用監査ログ出力ファイルだけ再作成します。

手順を次に示します。

1. 出力した監査ログが，JP1/NETM/Audit に収集されているかどうかを確認します。「運用方法２（監査証跡表にデータを残さない運用）」の 4.を参照してください。

2. JP1/NETM/Audit 用監査ログ出力ファイルの再見積もりをします。「JP1/NETM/Audit 用監査ログ出力ファイルの出力先ディレクトリの作成」を参照してください。
3. JP1/NETM/Audit による監視を停止します。
4. -s オプションに変更後の JP1/NETM/Audit 用監査ログ出力ファイルの容量を指定して，pdaudput -k m コマンドを実行してください。
5. JP1/NETM/Audit による監視を開始します。

24.17.5　JP1 側で障害が発生した場合

JP1/NETM/Audit 又は JP1/Base に障害が発生し，監査ログが収集されない状態で pdaudput コマンドを実行し続けると，必要な監査ログが上書きされるおそれがあります。この場合，JP1/NETM/Audit がどこまで監査ログを収集しているかを確認し，その監査証跡以降のデータを監査証跡表から再度出力します。このとき，JP1/NETM/Audit 用監査ログ出力ファイルに JP1/Base が収集していない監査ログが残っている可能性があるため，pdaudput コマンドで JP1/NETM/Audit 用監査ログ出力ファイルを再作成してから JP1/NETM/Audit 用監査ログを出力します。

手順を次に示します。

1. JP1 側の障害を回復します。対処方法については，マニュアル「JP1/NETM/Audit 構築・運用ガイド」を参照してください。
2. JP1/NETM/Audit による監視を停止します。
3. JP1/NETM/Audit がどこまで監査ログを収集しているかを確認します。
   JP1/NETM/Audit の管理画面で参照できる JP1/NETM/Audit 用監査ログの自由記述（タグ名 msg）に設定される値※で確認できます。
   注※
   pdaudput コマンドの場合，「aa…a，bb…b」の形式で設定されています。
   aa…a：監査証跡表の PDLOAD_TIMESTAMP 列値
   bb…b：監査証跡表の PDLOAD_SEQNUM 列値

   参考

   この手順を確認しないで監査証跡表のデータ削除の手順に進んでしまうと，pdaudput の出力結果が保証できない場合があります。この場合，削除したデータに対応する監査ログの再収集はできなくなるため，必ずこの手順を確認してください。
4. 収集された監査証跡を監査証跡表から削除します。
5. JP1/NETM/Audit 用監査ログ出力ファイルを再作成します。pdaudput -k i コマンドを実行してください。
6. JP1/NETM/Audit による監視を開始します。
7. pdaudput コマンドを実行して，JP1/NETM/Audit 用監査ログを出力します。

8. 出力した監査ログが，JP1/NETM/Audit に収集されているかどうかを確認します。「運用方法２（監査証跡表にデータを残さない運用）」の 4.を参照してください。
9. JP1 側の障害で収集されていなかった監査証跡をすべて JP1/NETM/Audit 用監査ログ出力ファイルに出力するまで，7.～8.を繰り返します。

24.17.6　注意事項

系切り替え機能を使用している場合は，実行系と待機系のそれぞれのシステムマネジャが配置されているサーバに対して JP1/NETM/Audit の設定を行い，両方の JP1/NETM/Audit 用監査ログ出力ファイルを収集してください。系切り替えが発生したときは，切り替え後の現用系のシステムマネジャが配置されているサーバで pdaudput コマンドを実行してください。このとき，JP1/NETM/Audit 用監査ログ出力ファイルの出力先は共用ディスク装置ではなく，ローカルディスクにしてください。監査ログの収集は，現用系の JP1/NETM/Audit 用監査ログ出力ファイルが作成された後から開始されます。

25 CONNECT 関連セキュリティ機能の運用

この章では，HiRDB システムのセキュリティ強化を目的とした CONNECT 関連セキュリティ機能の運用方法について説明します。

25.1　CONNECT 関連セキュリティ機能の概要

ここでは，CONNECT 関連セキュリティ機能の概要について説明します。説明する項目は次のとおりです。

- CONNECT 関連セキュリティ機能とは
- パスワードの文字列制限
- 連続認証失敗回数の制限
- IP アドレスによる接続制限

25.1.1　CONNECT 関連セキュリティ機能とは

CONNECT 関連セキュリティ機能の概要を次の表に示します。

表 25–1　CONNECT 関連セキュリティ機能の概要

機能	説明
パスワードの文字列制限	パスワードに指定する文字列に制限を設定できます。例えば，AAAAA や zzzzz などのパスワードを禁止できます。簡単なパスワードを禁止することでパスワードのセキュリティを強化します。
連続認証失敗回数の制限	不正なパスワードを使用し，ユーザ認証に連続して失敗した場合，そのユーザを HiRDB に接続（CONNECT）できないようにします。連続して失敗した回数の上限を設定し，その回数を超えたユーザは HiRDB に接続できなくなります。 例えば，不正なパスワードを使用して 3 回連続ユーザ認証に失敗した場合，そのユーザは HiRDB に接続できなくなります。
IP アドレスによる接続制限	HiRDB サーバへの接続を，IP アドレス単位で許可又は拒否できます。
パスワードの有効期間の設定	パスワードを使用できる期間（日数）を設定し，その期間までにパスワードを変更しなければ HiRDB に接続（CONNECT）できないようにします。

25.1.2　パスワードの文字列制限

(1)　パスワードに設定できる制限

パスワードに設定できる制限を次の表に示します。

表 25–2　パスワードに設定できる制限

項目	説明
最小許容バイト数の設定	パスワードの最小許容バイト数を設定できます。
認可識別子の指定禁止	パスワードの文字列中に認可識別子を指定することを禁止できます。

項目	説明
	大文字，小文字は区別しません。
単一文字種※の指定禁止	パスワードを同じ文字種（英大文字だけ又は英小文字だけ）だけで構成することを禁止できます。
必須文字種	パスワードを構成する文字に必須となる文字種※を設定します。
再利用禁止	過去に指定したパスワードを，新規パスワードとして再利用することを禁止します。再利用禁止として残すパスワード履歴の個数を指定します。

注※　パスワードに指定できる文字種は次のように分類されています。

- 英大文字：A～Z，#，@，¥
- 英小文字：a～z
- 数字：0～9

ポイント

パスワードの文字列制限はユーザ単位に設定できません。HiRDB の全ユーザ（DBA 権限保持者や，監査人も含む）に対して一律の適用になります。また，簡易認証ユーザは，パスワードの文字列制限機能の対象外となります。

参考

この機能を使用するかどうかに関係なく，パスワードには次に示す指定規則があります。

- パスワードには英数字を指定できます。ただし，先頭には英字を指定してください。

(2)　既存ユーザに対する影響

パスワードの文字列制限を設定した場合，制限に違反しているユーザは**パスワード無効アカウントロック状態**になります。パスワード無効アカウントロック状態のユーザは HiRDB に接続（CONNECT）できなくなります。

パスワード無効アカウントロック状態を解除するには，パスワードを変更する必要があります。パスワードの変更方法については，「ユーザ単位にパスワード無効アカウントロック状態を解除する」を参照してください。

また，パスワードの文字列制限を設定する前に，制限に違反するためパスワード無効アカウントロック状態になるユーザがどのくらいいるかを調査できます。調査方法については，「パスワード無効アカウントロック状態になるユーザを事前調査する」を参照してください。

(3)　新規ユーザに対する影響

GRANT DBA，GRANT AUDIT，又は GRANT CONNECT でパスワードを設定しますが，そのパスワードが制限に違反している場合，GRANT 文を実行できません。

(4)　設定方法

CREATE CONNECTION SECURITY でパスワードの文字列制限を設定します。設定方法については，「パスワードの文字列制限を新規設定する」を参照してください。

パスワードの文字列制限の運用方法については，次を参照してください。

- 「パスワードの文字列制限を新規設定する」
- 「パスワードの文字列制限を変更する」
- 「パスワード無効アカウントロック状態を解除する」
- 「パスワード無効アカウントロック状態になるユーザを事前調査する」
- 「パスワード無効アカウントロック状態のユーザの権限付与及び削除」
- 「パスワードの文字列制限を解除する」
- 「セキュリティ監査機能使用時の注意事項」
- 「CONNECT 関連セキュリティ機能運用時の注意事項」

(5)　データディクショナリ用 RD エリア容量の確認

パスワードの再利用禁止を設定する場合，ディクショナリ表 SQL_USERS の容量の再見積もりをしてください。詳細はマニュアル「HiRDB システム導入・設計ガイド」の「データディクショナリ用 RD エリアの容量の見積もり」を参照してください。

25.1.3　連続認証失敗回数の制限

(1)　設定できる制限

不正なパスワードを使用し，ユーザ認証に連続して失敗した場合，そのユーザを HiRDB に接続（CONNECT）できないようにします。連続して失敗できる回数の上限（**連続認証失敗許容回数**）を設定し，その回数を超えたユーザは HiRDB に接続できなくなります。

例えば，連続認証失敗許容回数を 3 と設定した場合，パスワード不正によって 4 回連続でユーザ認証に失敗すると，そのユーザは**連続認証失敗アカウントロック状態**になります。連続認証失敗アカウントロック状態のユーザは HiRDB に接続（CONNECT）できなくなります。

参考

連続認証失敗回数の制限はユーザ単位に設定できません。HiRDB の全ユーザ（DBA 権限保持者や，監査人も含む）に対して一律の適用になります。

また，連続認証失敗アカウントロック状態とする期間（**アカウントロック期間**）を設定できます。例えば，アカウントロック期間を 1 時間とした場合，連続認証失敗アカウントロック状態が 1 時間続きます。1 時間を過ぎると連続認証失敗アカウントロック状態が解除されて，HiRDB に接続できるようになります。

参考

- アカウントロック期間を無期限にすることもできます。
- アカウントロック期間内に連続認証失敗アカウントロック状態を解除することもできます。解除方法については，「連続認証失敗アカウントロック状態を解除する」を参照してください。

(2)　失敗回数のカウント方法

失敗回数としてカウントされるのは，30 バイト以内のパスワード不正のときだけです。次に示す場合は失敗回数としてカウントされません。

- 認可識別子を指定ミスした場合（存在しない認可識別子を指定した場合）
- 31 バイト以上のパスワードを指定した場合

参考

- 時間を空けても連続認証失敗回数は 0 に戻りません。例えば，失敗した 1 時間後に再度失敗した場合，連続認証失敗回数は 2 になります。
- HiRDB を終了しても連続認証失敗回数は 0 に戻りません。
- クライアントマシンを変更して CONNECT した場合でも連続認証失敗回数は有効になります。例えばマシン A で失敗し，その後マシン B で失敗した場合は，連続認証失敗回数が 2 となります。
- 接続するフロントエンドサーバを変更して CONNECT した場合でも連続認証失敗回数は有効になります。例えばフロントエンドサーバ A で失敗し，その後フロントエンドサーバ B で失敗した場合は，連続認証失敗回数が 2 となります。
- コマンド又はユティリティの実行もカウント対象になります。
- 連続認証失敗アカウントロック状態が解除されると，連続認証失敗回数が 0 に戻ります。

(3)　設定方法

CREATE CONNECTION SECURITY で連続認証失敗回数の制限を設定します。設定方法については，「連続認証失敗回数の制限を設定又は解除する」を参照してください。

連続認証失敗回数の制限の運用方法については，次に示します。

- 「連続認証失敗回数の制限を設定又は解除する」
- 「連続認証失敗アカウントロック状態のユーザを確認する」
- 「連続認証失敗アカウントロック状態を解除する」
- 「CONNECT 関連セキュリティ機能運用時の注意事項」

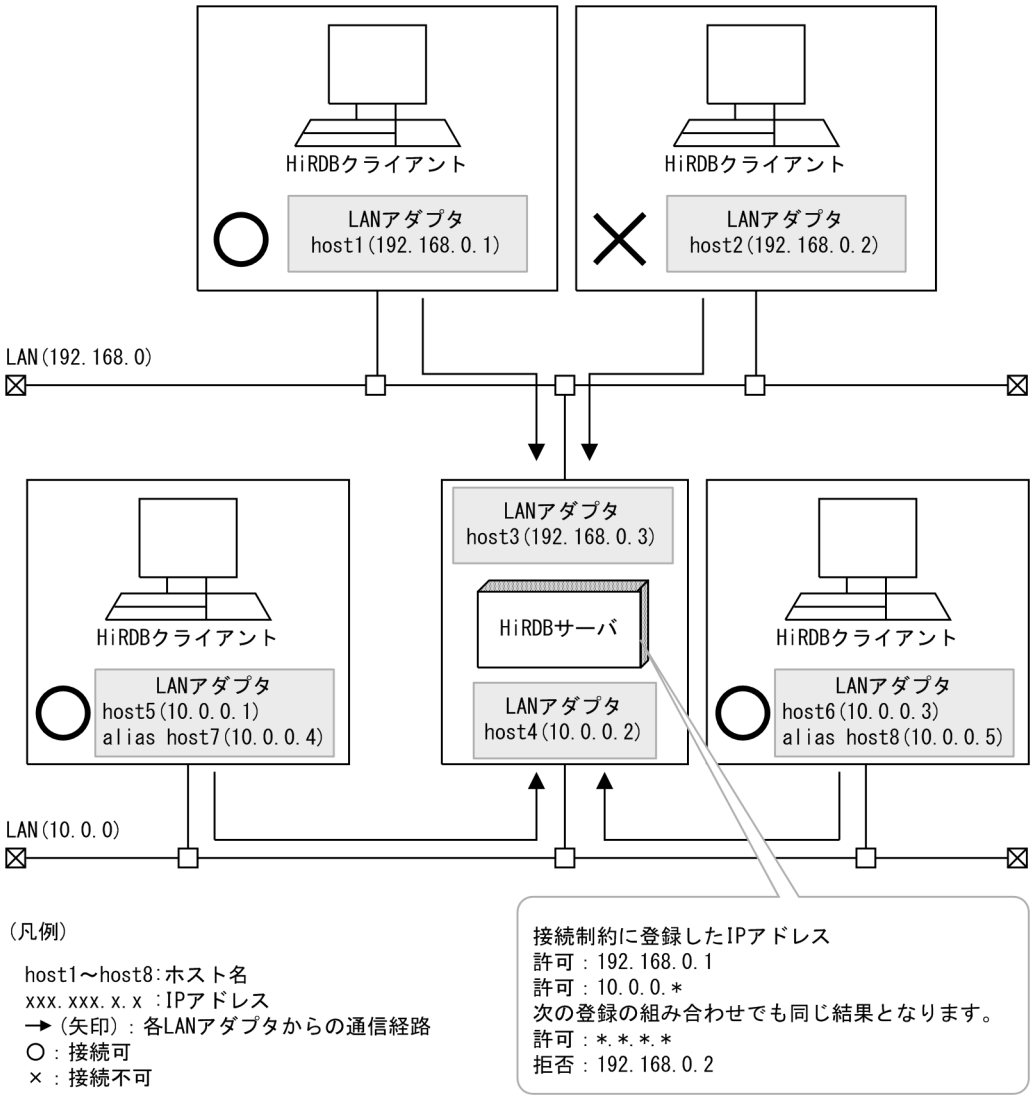
(4)　必要な RD エリア

この機能を使用する場合，連続認証失敗アカウントロック状態を確認するときにシステム定義スカラ関数 ADD_INTERVAL を使用するため，データディクショナリ LOB 用 RD エリアが必要になります。データディクショナリ LOB 用 RD エリアがない場合はデータディクショナリ LOB 用 RD エリアを作成してください。

25.1.4　IP アドレスによる接続制限

HiRDB クライアントに複数の IP アドレスが設定されている場合の HiRDB クライアント，HiRDB サーバ間通信では，HiRDB サーバに接続を許可する IP アドレス，及び接続を拒否する IP アドレスの登録ができます。登録には IP アドレスを指定します。UAP 実行時の HiRDB サーバへの接続では，登録された IP アドレスを使っての接続であるかどうかの判定によって接続の制限ができます。機能概要図を次に示します。

図 25–1　IP アドレスによる接続制限機能の概要



図の例では，IP アドレス 192.168.0.1 からの接続及びワイルドカード指定の 10.0.0.* を満たす IP アドレスからの接続を許可し，それ以外の IP アドレスからの接続は拒否します。また，拒否登録との組み合わせで同じ結果を実現することもできます。図の例では，すべての IP アドレスを示すワイルドカード指定の*.*.*.* を許可する登録と，IP アドレス 192.168.0.2 を拒否する指定の組み合わせとなります。

host1 及び host2 のように異なる HiRDB クライアントマシンからの接続について許可及び拒否を決めることができます。また，host5～host8 のように同一のエリアネットワーク LAN（10.0.0）からの接続を許可する設定ができます。

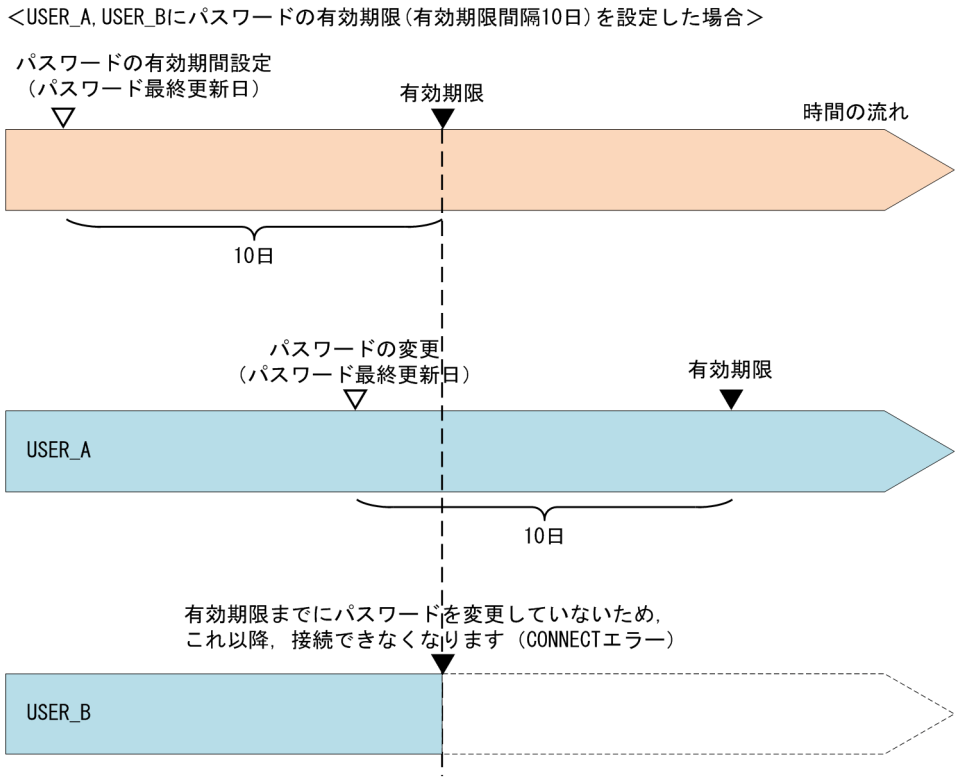
接続制限はすべてのユーザに適用されますが，特定のユーザにだけ接続制限を設定することもできます。

設定方法については「IP アドレスによる接続制限の設定方法」，運用例については「IP アドレスによる接続制限の運用例」を参照してください。

25.1.5　パスワードの有効期間の設定

任意のユーザに対し，パスワードを使用できる期間（有効期限間隔）を日数で設定します。有効期限間隔を設定した日，又は最後にパスワードを設定した日を「パスワードの最終更新日」として，有効期限を算出する起点の日とします。パスワードの最終更新日から有効期限以内は，同じパスワードで HiRDB に CONNECT ができます。パスワードの有効期間の概要を次の図に示します。

図 25–2　パスワードの有効期間の概要



注　パスワードの最終更新日は，YYYYMMDDHHMMSSの単位で管理します。

図の例では，USER_A，USER_B ともに，有効期間設定日がパスワードの最終更新日となり（2019/04/01 09:00:00 とします），その 10 日経過後がパスワード有効期限（2019/04/11 09:00:00）となります。

USER_A の場合，有効期限以内に GRANT 文でパスワードを変更しています。USER_A は，パスワードを変更した日がパスワードの最終更新日（2019/04/10 16:47:33）となり，その 10 日経過後がパスワード有効期限（2019/04/20 16:47:33）となります。

USER_B の場合，10 日経過するまでにパスワードを変更していません。USER_B は，有効期限（2019/04/11 09:00:00）よりあとの CONNECT はできなくなります。

設定方法，運用例，及び注意事項については「パスワード有効期間の設定方法」を参照してください。

25.2　パスワードの文字列制限を新規設定する

25.2.1　パスワードの文字列制限の新規設定手順

実行者　DBA 権限保持者

ここでは，パスワードの文字列制限を新規に設定する手順について説明します。(1)から順に作業を進めてください。

(1)　パスワードに設定する制限を検討する

パスワードに設定する制限を検討してください。次の表に示す項目を検討します。

表 25–3　パスワードに設定する制限

設定する制限	内容
最小許容バイト数の設定	パスワードの最小許容バイト数を設定します。最小で 6 文字以上を，最大で 15 文字以上を設定できます。
認可識別子の指定禁止	パスワードの文字列中に認可識別子を指定することを禁止するかどうかを設定します。大文字，小文字は区別しません。禁止した場合，例えば，次に示すパスワードが禁止されます。 **禁止されるパスワードの例** 認可識別子が K001 の場合：K001，abK001，K00165，GTK001KL
単一文字種の指定禁止	パスワードの文字列を同じ文字種で指定することを禁止するかどうかを設定します。禁止した場合，例えば，次に示すパスワードが禁止されます。 **禁止されるパスワードの例** HUDGTX，jkfgytud，D@MK#B¥
必須文字種	指定した文字種が，パスワードの構成文字列に含まれていることを必須とします。ALL の場合は，すべての文字種がパスワードに含まれていることを必須とします。
再利用禁止	パスワード履歴にあるパスワードと，同じ文字列を新規パスワードに指定することを禁止します。 パスワード履歴は，最小 1 個，最大 36 個を指定できます。ユーザごとに，指定した個数分のパスワードを，パスワード変更の際に残します。 パスワード履歴に残っているパスワードの数が指定した個数に達している場合，最も古いパスワードを破棄して，代わりに最も新しいパスワードをパスワード履歴に残します。

(2)　設定した制限に違反するユーザを事前調査する

設定した制限に違反するユーザを事前調査してください。制限に違反するユーザはパスワード無効アカウントロック状態になるため，HiRDB に接続できなくなります。したがって，制限を設定する前に，制限に違反するユーザを特定する必要があります。特定方法については，「パスワード無効アカウントロック状態になるユーザを事前調査する」を参照してください。

参考

パスワードの文字列制限を設定すると，制限によってはパスワード無効アカウントロック状態になるユーザが出てくる可能性があります。そのため，パスワードに制限を設定することを全ユーザに対して事前に連絡し，期日までにパスワードを変更してもらうようにしてください。期日になったら，制限に違反しているユーザの確認を行い，違反しているユーザに連絡をしてください。

(3) パスワードを変更する

制限に違反しているユーザのパスワードを GRANT 文で変更します。例を次に示します。

例題 1

ユーザ（USER01）のパスワードを f51HD7tc に変更します。

```
GRANT CONNECT TO USER01 IDENTIFIED BY ”f51HD7tc”
```

参考

自分のパスワードは自分で変更できます。DBA 権限保持者が変更することもできます。

例題 2

DBA 権限保持者（ADMIN01）のパスワードを gd4A@sPL に変更します。

```
GRANT DBA TO ADMIN01 IDENTIFIED BY ”gd4A@sPL”
```

例題 3

監査人のパスワードを a0h7Fc3K に変更します。

```
GRANT AUDIT IDENTIFIED BY ”a0h7Fc3K”
```

(4) パスワードの文字列制限を設定する

CREATE CONNECTION SECURITY でパスワードの文字列制限を設定します。

注意事項

DBA 権限保持者及び監査人のパスワードに違反がある場合（一人でも違反がある場合），CREATE CONNECTION SECURITY が実行できません。

CREATE CONNECTION SECURITY の指定例を次に示します。

例題

パスワードの文字列制限を次のように設定します。

- 最小許容バイト数は 8 バイトにする

- 認可識別子の指定を禁止する
- 単一文字種の指定を禁止する

```
CREATE CONNECTION SECURITY
    FOR PASSWORD
      MIN LENGTH 8                  ...1
      USER IDENTIFIER RESTRICT      ...2
      SIMILAR RESTRICT              ...3
```

〔説明〕

1. パスワードの最小許容バイト数を設定します。
2. 認可識別子の指定を禁止します。禁止する場合は RESTRICT を指定し，禁止しない場合は UNRESTRICT を指定します。
3. 単一文字種の指定を禁止します。禁止する場合は RESTRICT を指定し，禁止しない場合は UNRESTRICT を指定します。

(5)　パスワード無効アカウントロック状態のユーザがいないか確認する

パスワード無効アカウントロック状態のユーザがいないか確認してください。確認方法については，「パスワード無効アカウントロック状態のユーザを確認する」を参照してください。

25.3　パスワードの文字列制限を変更する

ここでは，パスワードの文字列制限を変更するときの手順について説明します。

25.3.1　パスワードの文字列制限を変更するときの留意事項

(1)　パスワードの文字列制限を緩和する場合

パスワード無効アカウントロック状態のユーザがいる場合に，パスワードの文字列制限を緩和すると，設定する制限によってはパスワード無効アカウントロック状態が解除されます。該当するユーザのパスワード無効アカウントロック状態が解除されてもよいかを確認してから，パスワードの文字列制限を変更してください。

パスワード無効アカウントロック状態が解除されるユーザを特定する手順を次に示します。

〈手順〉

1. パスワード無効アカウントロック状態になっているユーザを調査します。調査方法については，「パスワード無効アカウントロック状態のユーザを確認する」を参照してください。
2. 変更後の制限で事前調査をしてください。事前調査の方法については，「パスワード無効アカウントロック状態になるユーザを事前調査する」を参照してください。
3. 1 と 2 の差分によって該当するユーザを特定します。

(2)　制限を強化する項目と緩和する項目が混在する場合

制限を強化する項目と緩和する項目が混在する場合，変更後の制限に違反しないように移行日以前にパスワードを変更すると，現在の制限に違反してパスワードを変更できないことがあります。この場合，現在の制限と変更後の制限の両方を満たすパスワードに変更してください。

25.3.2　パスワードの文字列制限の変更手順

実行者　DBA 権限保持者

パスワードの文字列制限を変更する手順について説明します。(1)から順に作業を進めてください。

(1)　現在設定されている制限を確認する

パスワードの文字列制限の設定情報は，ディクショナリ表の SQL_SYSPARMS に格納されています。現在設定されている制限を忘れた場合は，SQL_SYSPARMS を検索し，パスワードの文字列制限の設定情報を確認してください。検索例を次に示します。

例題

現在設定されているパスワードの文字列制限を確認します。

```
SELECT FUNCTION_KEY,PARAM_KEY,INT_VALUE,CHAR_VALUE
    FROM MASTER.SQL_SYSPARAMS
      WHERE FUNCTION_KEY='PASSWORD'
```

実行結果

```
FUNCTION_KEY   PARAM_KEY        INT_VALUE  CHAR_VALUE
-------------------------------------------
PASSWORD       MIN_LENGTH       8          8
PASSWORD       USER_IDENTIFIER  NULL       RESTRICT
PASSWORD       SIMILAR          NULL       RESTRICT
```

〔説明〕

次のように設定されています。

- パスワードの最小許容バイト数：8
- 認可識別子の指定：禁止（RESTRICT）
- 単一文字種の指定：禁止（RESTRICT）

許可の場合は UNRESTRICT が表示されます。

(2)　パスワードに設定する制限を検討する

パスワードに設定する制限を検討してください。詳細は表「パスワードに設定する制限」を参照してください。

(3)　設定した制限に違反するユーザを事前調査する

設定した制限に違反するユーザを事前調査してください。制限に違反するユーザはパスワード無効アカウントロック状態になるため，HiRDB に接続できなくなります。したがって，制限を変更する前に，制限に違反するユーザを特定する必要があります。特定方法については，「パスワード無効アカウントロック状態になるユーザを事前調査する」を参照してください。

参考

パスワードの文字列制限を変更すると，制限によってはパスワード無効アカウントロック状態になるユーザが出てくる可能性があります。そのため，パスワードの制限を変更することを全ユーザに対して事前に連絡し，期日までにパスワードを変更してもらうようにしてください。期日になったら，制限に違反しているユーザの確認を行い，違反しているユーザに連絡をしてください。

(4)　パスワードを変更する

制限に違反しているユーザのパスワードを GRANT 文で変更します。例を次に示します。

例題 1

ユーザ（USER01）のパスワードを f51HD7tc に変更します。

```
GRANT CONNECT TO USER01 IDENTIFIED BY ”f51HD7tc”
```

> **参考**
>
> 自分のパスワードは自分で変更できます。DBA 権限保持者が変更することもできます。

例題 2

DBA 権限保持者（ADMIN01）のパスワードを gd4A@sPL に変更します。

```
GRANT DBA TO ADMIN01 IDENTIFIED BY ”gd4A@sPL”
```

例題 3

監査人のパスワードを a0h7Fc3K に変更します。

```
GRANT AUDIT IDENTIFIED BY ”a0h7Fc3K”
```

(5)　パスワードの文字列制限を変更する

パスワードの文字列制限を変更する手順を次に示します。

〈手順〉

1. DROP CONNECTION SECURITY でパスワードの文字列制限を解除します。解除の方法については，「パスワードの文字列制限を解除する」を参照してください。
2. CREATE CONNECTION SECURITY でパスワードの文字列制限を設定します。

> **参考**
>
> 一部の制限項目だけを変更する場合でも（例えば，パスワードの最小許容バイト数だけを変更する場合でも），一度 DROP CONNECTION SECURITY で制限を解除した後に，CREATE CONNECTION SECURITY で制限を設定する必要があります。

(6)　パスワード無効アカウントロック状態のユーザがいないか確認する

パスワード無効アカウントロック状態のユーザがいないか確認してください。確認方法については，「パスワード無効アカウントロック状態のユーザを確認する」を参照してください。

25.4　パスワード無効アカウントロック状態を解除する

ここでは，パスワード無効アカウントロック状態を解除する方法について説明します。

25.4.1　ユーザ単位にパスワード無効アカウントロック状態を解除する

実行者　DBA 権限保持者

パスワード無効アカウントロック状態を解除する手順について説明します。(1)から順に作業を進めてください。

(1)　パスワード無効アカウントロック状態のユーザを確認する

パスワード無効アカウントロック状態のユーザを確認してください。例を次に示します。

例題

パスワード無効アカウントロック状態のユーザの認可識別子を表示します。

```
SELECT USER_ID
    FROM MASTER.SQL_USERS
      WHERE PWD_LOCK_TIME IS NOT NULL
```

実行結果

```
USER_ID
----------
USER1
USER2
```

〔説明〕

USER1 と USER2 がパスワード無効アカウントロック状態です。

> **参考**
>
> パスワード無効アカウントロック状態の場合，ディクショナリ表の SQL_USERS の PWD_LOCK_TIME 列に，パスワード無効アカウントロック状態になった日時が設定されます。パスワード無効アカウントロック状態でない場合は，PWD_LOCK_TIME 列にナル値が設定されます。

(2)　パスワードを変更する

制限に違反しているユーザのパスワードを GRANT 文で変更します。例を次に示します。

例題

ユーザ（USER01）のパスワードを f51HD7tc に変更します。

```
GRANT CONNECT TO USER01 IDENTIFIED BY ”f51HD7tc”
```

25.4.2　全ユーザのパスワード無効アカウントロック状態を解除する

実行者　DBA 権限保持者

DROP CONNECTION SECURITY で，パスワードの文字列制限の設定を解除します。これによって，全ユーザのパスワード無効アカウントロック状態が解除されます。

例題

パスワードの文字列制限の設定を解除し，全ユーザのパスワード無効アカウントロック状態を解除します。

```
DROP CONNECTION SECURITY FOR PASSWORD
```

25.5　パスワード無効アカウントロック状態になるユーザを事前調査する

25.5.1　パスワード無効アカウントロック状態になるユーザの事前調査手順

実行者　DBA 権限保持者

ここでは，パスワード無効アカウントロック状態になるユーザを事前調査する手順について説明します。(1)から順に作業を進めてください。

(1)　TEST オプションを指定して CREATE CONNECTION SECURITY を実行する

TEST オプションを指定して CREATE CONNECTION SECURITY を実行してください。TEST オプションを指定すると，CREATE CONNECTION SECURITY で指定した制限に違反しているユーザに対しては，ディクショナリ表の SQL_USERS の PASSWORD_TEST 列に**違反種別コード**が設定されます。

> **参考**
>
> TEST オプションを指定した場合，パスワードの文字列制限のチェックだけが行われます。違反者をパスワード無効アカウントロック状態にはしません。

パスワード無効アカウントロック状態になるユーザを事前調査するときの例を次に示します。

例題

パスワードに次の文字列制限を設定する場合の違反者を調べます。

- パスワードの最小許容バイト数：8
- パスワード中の認可識別子指定：禁止
- パスワードの単一文字種指定：禁止

```
CREATE CONNECTION SECURITY
    FOR PASSWORD TEST             ...1
      MIN LENGTH 8                ...2
      USER IDENTIFIER RESTRICT    ...3
      SIMILAR RESTRICT            ...4
```

〔説明〕

1. 事前調査のため，TEST オプションを指定します。
2. パスワードの最小許容バイト数を 8 バイトとします。
3. パスワード中の認可識別子の指定を禁止します。
4. パスワードの単一文字種の指定を禁止します。

この SQL 文を実行すると，SQL_USERS に登録されている全ユーザのパスワードをチェックし，制限に違反しているユーザに対しては SQL_USERS の PASSWORD_TEST 列に違反種別コードを設定します。

ポイント

- 事前調査時とパスワードの文字列制限を設定する時で SQL（CREATE CONNECTION SECURITY）の指定内容に違いがあると，事前にパスワードを修正したユーザでもパスワード無効アカウントロック状態になる可能性があります。そのため，事前調査時には，TEST オプション以外はパスワードの文字列制限を設定する時に実行する SQL（CREATE CONNECTION SECURITY）と同じ SQL で実行してください。
- 事前調査の実行後は，文字列制限に違反しているパスワード以外のパスワードを変更しないように連絡してください。また，新規ユーザの登録をする場合は，制限を満たすパスワードを設定してください。

(2) PASSWORD_TEST 列を検索する

SQL_USERS の PASSWORD_TEST 列に違反種別コードが設定されます。PASSWORD_TEST 列を検索して，パスワード無効アカウントロック状態のユーザを確認してください。設定される違反種別コードを次の表に示します。違反していない場合はナル値が設定されます。

表 25–4 PASSWORD_TEST 列に設定される違反種別コード

項番	項目	PASSWORD_TEST 列に設定される違反種別コード
1	最小許容バイト数の設定に違反した場合	L
2	認可識別子の指定禁止に違反した場合	U
3	単一文字種の指定禁止に違反した場合	S

注

複数の項目に違反している場合，違反種別コードは複数設定されません。項番の順に一つだけ設定されます。例えば，項番 1 と 2 の両方の項目に違反している場合は，違反種別コードに L が設定されます。

パスワード無効アカウントロック状態のユーザを確認するときの例を次に示します。

例題 1

パスワードの文字列制限に違反しているユーザの一覧を取得します。

```
SELECT USER_ID
   FROM MASTER.SQL_USERS
     WHERE PASSWORD_TEST IS NOT NULL
```

実行結果

```
USER_ID
-----------
USER1
```

```
USER2
USER3
```

〔説明〕

USER1，USER2，及び USER3 がパスワードの文字列制限に違反しています。

例題 2

パスワードの文字列制限に違反している DBA 権限保持者及び監査人の一覧を取得します。

```
SELECT USER_ID
    FROM MASTER.SQL_USERS
      WHERE PASSWORD_TEST IS NOT NULL
        AND (DBA_PRIVILEGE = 'Y' OR AUDIT_PRIVILEGE = 'Y')
```

実行結果

```
USER_ID
-----------
AUDITOR1
DBA1
DBA2
```

〔説明〕

DBA 権限保持者 DBA1 と DBA2，及び監査人 AUDITOR1 がパスワードの文字列制限に違反しています。

25.6　パスワード無効アカウントロック状態のユーザの権限付与及び削除

パスワード無効アカウントロック状態のユーザに対する権限付与

パスワード無効アカウントロック状態のユーザ（CONNECT 権限を持つユーザ）に対しては，DBA 権限を付与できません。パスワード無効アカウントロック状態を解除してから DBA 権限を与えてください。

パスワード無効アカウントロック状態のユーザに対する権限削除

パスワード無効アカウントロック状態のユーザの権限を REVOKE 文で削除できます。

25.7　パスワードの文字列制限を解除する

実行者　DBA 権限保持者

DROP CONNECTION SECURITY を実行して，パスワードの文字列制限を解除します。この操作を行うと，パスワードの文字列制限を行う以前の状態に戻ります。

例題

パスワードの文字列制限を解除します。

```
DROP CONNECTION SECURITY FOR PASSWORD
```

参考

項目単位の解除（例えば，パスワードの最小許容バイト数の設定だけを解除するなど）はできません。

注意事項

パスワードの文字列制限を解除すると，パスワード無効アカウントロック状態が解除されます。このため，HiRDB に接続させたくないユーザも HiRDB に接続できるようになるため，事前にパスワード無効アカウントロック状態のユーザを確認してください。また，再利用禁止を設定していた場合，すべてのユーザのパスワード履歴をクリアします。

25.8　セキュリティ監査機能使用時の注意事項

セキュリティ監査機能使用時の注意事項を次に示します。

- pdmod コマンドで監査人を登録したときは，パスワードの文字列制限のチェックを行いません。したがって，監査人を登録したときは，すぐに GRANT AUDIT 文で監査人のパスワードを変更してください。パスワードの変更は登録された監査人自身が行ってください。
- パスワード無効アカウントロック状態のユーザを監査人にできません。

25.9　連続認証失敗回数の制限を設定又は解除する

ここでは，連続認証失敗回数の制限を設定又は解除する方法について説明します。説明する項目は次のとおりです。

- 連続認証失敗回数の制限を新規設定する方法
- 連続認証失敗回数の制限を解除する方法
- 連続認証失敗回数の制限を変更する方法
- 連続認証失敗許容回数及びアカウントロック期間を確認する方法

25.9.1　連続認証失敗回数の制限を新規設定する

CREATE CONNECTION SECURITY で，連続認証失敗回数の制限を設定します。例を次に示します。

例題 1

連続認証失敗回数の制限を次のように設定します。

- 連続認証失敗許容回数：3 回
- アカウントロック期間：30 分

```
CREATE CONNECTION SECURITY
    FOR CONNECT PERMISSION COUNT 3     ...1
                LOCK 30 MINUTE         ...2
```

〔説明〕

1. 連続認証失敗許容回数を指定します。
2. アカウントロック期間を指定します。

例題 2

連続認証失敗回数の制限を次のように設定します。

- 連続認証失敗許容回数：5 回
- アカウントロック期間：無期限

```
CREATE CONNECTION SECURITY
    FOR CONNECT PERMISSION COUNT 5     ...1
                LOCK UNLIMITED         ...2
```

〔説明〕

1. 連続認証失敗許容回数を指定します。
2. アカウントロック期間を指定します。

25.9.2　連続認証失敗回数の制限を解除する

DROP CONNECTION SECURITY で，連続認証失敗回数の制限を解除します。例を次に示します。

例題

連続認証失敗回数の制限を解除します。

```
DROP CONNECTION SECURITY
    FOR CONNECT
```

> **参考**
>
> 連続認証失敗回数の制限を解除すると，連続認証失敗アカウントロック状態が解除されます。

25.9.3　連続認証失敗回数の制限を変更する

連続認証失敗回数の制限を変更する場合は，DROP CONNECTION SECURITY で制限を一度解除してから，CREATE CONNECTION SECURITY で制限を再設定します。例を次に示します。

例題

連続認証失敗回数の制限を次のように変更します。

- 連続認証失敗許容回数：5 回
- アカウントロック期間：10 時間

```
DROP CONNECTION SECURITY
    FOR CONNECT                          ...1
CREATE CONNECTION SECURITY
    FOR CONNECT PERMISSION COUNT 5       ...2
                LOCK 10 HOUR             ...3
```

〔説明〕

1. 連続認証失敗回数の制限を解除します。
2. 連続認証失敗許容回数を指定します。
3. アカウントロック期間を指定します。

25.9.4　連続認証失敗許容回数及びアカウントロック期間を確認する

実行者　DBA 権限保持者又は監査人

設定されている連続認証失敗許容回数及びアカウントロック期間を確認するには，ディクショナリ表 SQL_SYSPARAMS を検索してください。例を次に示します。

例題

設定されている連続認証失敗許容回数及びアカウントロック期間を確認します。

```
SELECT * FROM MASTER.SQL_SYSPARAMS
    WHERE PARAM_KEY='PERMISSION_COUNT' OR PARAM_KEY='LOCK_MINUTE'
```

実行結果 1

```
PARAM_KIND           FUNCTION_KEY   PARAM_KEY        INT_VALUE  CHAR_VALUE
------------------------------------------------------------------------------
CONNECTION_SECURITY  CONNECT        PERMISSION_COUNT 2          2
CONNECTION_SECURITY  CONNECT        LOCK_MINUTE      1440       1440
```

〔説明〕

連続認証失敗許容回数は 2，アカウントロック期間は 1440 分です。

SQL_SYSPARAMS に格納されているアカウントロック期間の単位は分です。

実行結果 2

```
PARAM_KIND           FUNCTION_KEY   PARAM_KEY        INT_VALUE  CHAR_VALUE
------------------------------------------------------------------------------
CONNECTION_SECURITY  CONNECT        PERMISSION_COUNT 2          2
CONNECTION_SECURITY  CONNECT        LOCK_MINUTE      NULL       UNLIMITED
```

〔説明〕

連続認証失敗許容回数は 2，アカウントロック期間は無期限（UNLIMITED）です。

25.10　連続認証失敗アカウントロック状態のユーザを確認する

実行者　DBA 権限保持者又は監査人

連続認証失敗アカウントロック状態のユーザを確認するには，ディクショナリ表を検索します。検索例を次に示します。

なお，システム定義スカラ関数 ADD_INTERVAL を使用するため，データディクショナリ LOB 用 RD エリアが必要になります。データディクショナリ LOB 用 RD エリアがない場合は作成してください。

例題 1

連続認証失敗アカウントロック状態のユーザを表示します（アカウントロック期間に無期限を指定していない場合）。

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CAST(CON_LOCK_TIME AS CHAR(19)) >=
  (SELECT MASTER.ADD_INTERVAL(CAST(CURRENT_TIMESTAMP(0) AS CHAR(19)),-INT_VALUE)
    FROM MASTER.SQL_SYSPARAMS WHERE PARAM_KEY = 'LOCK_MINUTE_CODE')
      AND CON_LOCK_TIME IS NOT NULL
```

実行結果

```
USER_ID CON_LOCK_TIME
--------------------
USER1   2005-01-19 11:37:58
USER2   2005-01-19 12:06:11
```

〔説明〕

USER1 と USER2 が連続認証失敗アカウントロック状態です。

例題 2

連続認証失敗アカウントロック状態のユーザを表示します（アカウントロック期間に無期限を指定している場合）。

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CON_LOCK_TIME IS NOT NULL
```

実行結果

```
USER_ID CON_LOCK_TIME
--------------------
USER1   2005-01-19 13:17:23
USER2   2005-01-19 13:17:35
```

〔説明〕

USER1 と USER2 が連続認証失敗アカウントロック状態です。

例題 3

連続認証失敗アカウントロック状態でないユーザを表示します（アカウントロック期間に無期限を指定していない場合）。

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CAST(CON_LOCK_TIME AS CHAR(19)) <
  (SELECT MASTER.ADD_INTERVAL(CAST(CURRENT_TIMESTAMP(0) AS CHAR(19)),-INT_VALUE)
    FROM MASTER.SQL_SYSPARAMS WHERE PARAM_KEY = 'LOCK_MINUTE_CODE')
      OR CON_LOCK_TIME IS NULL
```

実行結果

```
USER_ID  CON_LOCK_TIME
--------------------
USER1    NULL
AUDITOR1 NULL
```

〔説明〕

USER1 と AUDITOR1 が連続認証失敗アカウントロック状態ではありません。

例題 4

連続認証失敗アカウントロック状態でないユーザを表示します（アカウントロック期間に無期限を指定している場合）。

```
SELECT USER_ID,CON_LOCK_TIME FROM MASTER.SQL_USERS
  WHERE CON_LOCK_TIME IS NULL
```

実行結果

```
USER_ID  CON_LOCK_TIME
--------------------
USER1    NULL
AUDITOR1 NULL
```

〔説明〕

USER1 と AUDITOR1 が連続認証失敗アカウントロック状態ではありません。

注意事項

フロントエンドサーバとディクショナリサーバが異なるサーバマシンに配置されている場合，サーバマシンに設定されている時間を合わせてください。時間が異なっていると，正しい検索結果が得られない可能性があります。

参考

連続認証失敗アカウントロック状態になった日時がディクショナリ表の SQL_USERS に格納されます。この日時は連続認証失敗アカウントロック状態が解除されても，残ったままになります。ユーザ認証（CONNECT）に成功したときに日時がクリアされます。

25.11　連続認証失敗アカウントロック状態を解除する

実行者　HiRDB 管理者

pdacunlck コマンドで連続認証失敗アカウントロック状態を解除します。連続認証失敗アカウントロック状態のユーザを確認した後に，そのユーザの連続認証失敗アカウントロック状態を解除してください。例を次に示します。

例題 1

ユーザ（USER01）の連続認証失敗アカウントロック状態を解除します。

```
pdacunlck USER01
```

例題 2

ユーザ（USER01 及び USER02）の連続認証失敗アカウントロック状態を解除します。

```
pdacunlck USER01,USER02
```

例題 3

全ユーザの連続認証失敗アカウントロック状態を解除します。

```
pdacunlck ALL
```

> **参考**
>
> DROP CONNECTION SECURITY で，連続認証失敗回数の制限を解除した場合，全ユーザの連続認証失敗アカウントロック状態が解除されます。

25.12　IP アドレスによる接続制限の設定方法

25.12.1　接続制限対象の登録

接続制限対象の登録は，DBA 権限保持者がします。

接続制限対象の登録は，CREATE CONNECTION SECURITY FOR CONSTRAINT 文を実行します。登録時に指定する項目を次の表に示します。

表 25–5　登録情報

情報	内容
接続制約名	接続制限の条件を一意に示す名称を設定します。
接続可否	接続の許可，又は拒否の判定種別を設定します。
IP アドレス	接続制限の対象となる IP アドレスを設定します。 個別の IP アドレス指定のほか，アドレスプレフィックス指定，及びアスタリスク文字("*")を使用したワイルドカード指定ができます。
認可識別子	特定のユーザに対して接続制限を設定する場合に，適用するユーザの認可識別子を設定します。 省略した場合，接続制限はすべてのユーザに対し適用されます。 認可識別子が指定された接続制限が登録されているユーザは，認可識別子指定の接続制限だけが適用されます（認可識別子が省略された接続制限は適用されません）。

最大登録数は 10,000 です。

構文規則についてはマニュアル「HiRDB SQL リファレンス」の CREATE CONNECTION SECURITY を参照してください。

登録情報は，ディクショナリ表（SQL_ACCESS_SECURITY）に格納し，DBA 権限保持者，又は監査人が参照できます。ディクショナリ表の詳細については「HiRDB UAP 開発ガイド」の SQL_ACCESS_ SECURITY 表の内容を参照してください。

25.12.2　接続制限対象の削除

DBA 権限保持者が DROP CONNECTION SECURITY FOR CONSTRAINT 文を実行して削除します。構文規則についてはマニュアル「HiRDB SQL リファレンス」の DROP CONNECTION SECURITY 文 を参照してください。

25.12.3　登録時の注意事項

- 拒否設定をする場合は，必ず拒否登録を行う以外の IP アドレスに対して先に許可登録をするようにしてください。接続の拒否，及び許可に関係なく，1 つでも接続制約を定義した場合，許可の接続制約を定義していない IP アドレスからは接続できません。
- 許可と拒否に同じ IP アドレスが含まれる場合は，拒否を優先します。拒否を範囲指定し，その中の一部を許可登録しても，すべて拒否とします。
- 1 つの HiRDB クライアントマシンに複数の IP アドレスが存在する場合は，割り当てているすべての IP アドレスについて同じ接続可否設定をするか，又はクライアント環境定義 PDCLTSNDADDR で指定した IP アドレスに対して接続可否設定（推奨）をしてください。
- この機能はユティリティも対象になります。HiRDB サーバを構成するネットワークで使用しているマシンの IP アドレスを許可登録してください。ただし，例外事項について次に記載します。

  1. FES 又は SDS のあるユニットのマシンからの自マシンに対するローカル接続は，接続制約の登録内容に関わらず許可します。ただし，システム定義 pd_security_local_constraint オペランドに Y が指定されている場合，FES 又は SDS のあるユニットのマシンからのローカル接続であっても接続制限の対象となります。pd_security_local_constraint オペランドの詳細については，マニュアル「HiRDB システム定義」を参照してください。
  2. システム定義 pd_security_host_group オペランドの指定がある場合は，指定されたホスト情報に対応する IP アドレス※については接続許可対象とします。しかし，そのホスト情報に該当する IP アドレスについて，CREATE CONNECTION SECURITY FOR CONSTRAINT 文で拒否の登録をしていた場合は，拒否を優先します。

     注※　HiRDB 起動時に名称解決したものを使用します。起動中にホスト名と IP アドレスの関係が変更された場合は，その変更を反映しません。
- システム定義 pd_security_host_group オペランドを指定している際に，FES 又は SDS のあるユニットのマシンからのローカル接続を接続制限の対象とする場合，次の 2 つの設定が必要となることに注意してください。定義例については，「FES 又は SDS のあるユニットのマシンからのローカル接続を拒否したいとき」を参照してください。

  1. システム定義 pd_security_local_constraint オペランドに Y を指定します。
  2. 接続制限を行う FES 又は SDS のあるユニットのマシンについて，拒否の登録を行います。
- 認可識別子を指定して接続制限を登録する際，指定する認可識別子はユーザとして存在する認可識別子を指定してください。また，REVOKE CONNECT によってユーザを削除した場合，削除するユーザの認可識別子を指定した接続制限も削除されます。

25.12.4　接続可否の決定方法

接続不可となる IP アドレスについては，CREATE CONNECTION SECURITY FOR CONSTRAINT での登録後，CONNECT 時に接続不可（KFPA19632-E メッセージ出力）となります。すでに接続中のものは，DISCONNECT するまではその接続が保持され，次回 CONNECT 時に接続不可となります。

すでに接続されているものを排除したい場合は，「pdls -d prc」で接続中の UAP の IP アドレスを確認後，pdcancel コマンドでプロセスを終了してください。

pdcancel コマンドについてはマニュアル「HiRDB SQL リファレンス」の運用コマンド pdcancel（UAP，ユティリティ処理の強制終了）を参照してください。

登録内容による接続可否を次の表に示します。

表 25–6　接続可否の決定方法

<table>
<tr><th rowspan="3">登録数</th><th rowspan="3">FES 又は SDS サーバのあるユニットのマシンからのローカル接続，かつ pd_security_local_constraint が Y 以外</th><th colspan="3">認可識別子を指定した接続制限との比較</th><th colspan="3">認可識別子を省略した接続制限との比較</th><th rowspan="3">pd_security_host_group の設定値との比較</th><th rowspan="3">接続可否</th></tr>
<tr><th rowspan="2">接続ユーザの認可識別子と一致する登録の有無</th><th colspan="2">接続してきた IP アドレスとの一致/不一致</th><th rowspan="2">登録の有無</th><th colspan="2">接続してきた IP アドレスとの一致/不一致</th></tr>
<tr><th>拒否登録との比較</th><th>許可登録との比較</th><th>拒否登録との比較</th><th>許可登録との比較</th></tr>
<tr><td>0</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>○</td></tr>
<tr><td rowspan="11">1 以上</td><td rowspan="10">NO</td><td rowspan="5">あり</td><td>一致</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>×</td></tr>
<tr><td rowspan="4">不一致又は登録なし</td><td>一致</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>○</td></tr>
<tr><td rowspan="2">不一致又は登録なし</td><td>–</td><td>–</td><td>–</td><td>一致</td><td>○</td></tr>
<tr><td>不一致又はオペランド指定なし</td><td>×</td></tr>
<tr><td rowspan="5">なし</td><td rowspan="5">–</td><td rowspan="5">–</td><td rowspan="4">あり</td><td>一致</td><td>–</td><td>–</td><td>×</td></tr>
<tr><td rowspan="3">不一致又は登録なし</td><td>一致</td><td>–</td><td>○</td></tr>
<tr><td rowspan="2">不一致又は登録なし</td><td>一致</td><td>○</td></tr>
<tr><td>不一致又はオペランド指定なし</td><td>×</td></tr>
<tr><td>なし</td><td>–</td><td>–</td><td>–</td><td>○※</td></tr>
<tr><td>YES</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>○</td></tr>
</table>

（凡例）

–：該当しない

○：接続可

×：接続不可

注※

接続ユーザ以外のユーザに対する認可識別子指定の接続制約だけが登録されている状態です。接続ユーザに対する接続制約は認可識別子指定あり，なしともに登録されていないため，無条件に許可します。

25.13　IP アドレスによる接続制限の運用例

25.13.1　特定の IP アドレスだけ接続許可したい場合

接続を許可する IP アドレスを登録します。

1. IP アドレス 192.168.0.1 からの接続を許可します（接続制約名を'CLIENT01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”CLIENT01” FOR ’192.168.0.1’
```

2. 10.0.0.*を満たす IP アドレスからの接続を許可します（接続制約名を'AREA01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”AREA01” FOR ’10.0.0.*’
```

3. 11.22.33.0～11.22.33.255 の範囲を満たす IP アドレスからの接続を許可します（接続制約名を'AREA02'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”AREA02” FOR ’11.22.33.0/24’
```

25.13.2　一定範囲内の IP アドレスからの接続を許可するが，一部の IP アドレスは例外的に拒否したい場合

例　10.0.0.*を満たす IP アドレスからの接続は許可するが，IP アドレス 10.0.0.5 からの接続は拒否したい。

この場合，次の 2 つを登録します。

1. 10.0.0.*を満たす IP アドレスからの接続を許可します（接続制約名を'AREA01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”AREA01” FOR ’10.0.0.*’
```

2. IP アドレス 10.0.0.5 からの接続を拒否します（接続制約名を'DENY01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”DENY01” FOR EXCEPT ’10.0.0.5’
```

25.13.3　特定の IP アドレスだけ接続を拒否したい場合

例　IP アドレス 10.0.0.5 からの接続は拒否したい。

この場合，接続拒否の登録と，そのほかの IP アドレスから接続許可するための登録の 2 つが必要となります。必ず許可登録から先にしてください。

1. 全 IP アドレス*.*.*.*からの接続を許可します（接続制約名を'ALLAREA'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”ALLAREA” FOR ’*.*.*.*’
```

2. IP アドレス 10.0.0.5 からの接続を拒否します（接続制約名を'DENY01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”DENY01” FOR EXCEPT ’10.0.0.5’
```

25.13.4　登録を変更したいとき

登録内容誤り，又は登録済みの内容に対する接続許可もしくは拒否の変更を行う場合は，一度登録を削除し，新しく登録し直します。

例　誤って IP アドレス 192.168.0.1 について（接続制約名を'CLIENT01'とします）の登録をしたが 192.168.0.5 にしたい。

このときの手順を次に示します。

1. 接続制約名'CLIENT01'の登録情報を削除します。

```
DROP CONNECTION SECURITY FOR CONSTRAINT ”CLIENT01”
```

2. IP アドレス 192.168.0.5 からの接続を許可登録します（接続制約名を再度' 'CLIENT01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”CLIENT01” FOR ’192.168.0.5’
```

25.13.5　FES 又は SDS のあるユニットのマシンからのローカル接続を拒否したいとき

例　FES が配置されたユニットのあるマシンの IP アドレス 192.168.0.1 からの接続を拒否したい。

このときの手順を次に示します。

1. HiRDB が開始している場合は，正常停止します。
2. システム共通定義ファイルに pd_security_local_constraint オペランドを記述し，Y を指定します。

```
set pd_security_local_constraint = Y
```

3. HiRDB を開始します。
4. IP アドレス 192.168.0.1 からの接続を拒否します（接続制約名を'DENYLH01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”DENYLH01” FOR EXCEPT ’196.168.0.1’
```

25.13.6　特定のユーザにだけ接続制限を設定したいとき

接続制限を設定したいユーザの認可識別子と，接続を許可する IP アドレスを登録します。

例　10.0.0.*を満たすIPアドレスからの接続だけを許可するようにしたい。ただし，ユーザだけUSER1'192.168.0.1'からの接続も許可したい。

このときの手順を次に示します。

1. 10.0.0.*を満たすIPアドレスからの接続を許可する接続制限（認可識別子を省略）を登録します（接続制約名を'AREACMN01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”AREACMN01” FOR ’10.0.0.*’
```

2. ユーザUSER1に対し，IPアドレス192.168.0.1からの接続を許可する接続制限（認可識別子を指定）を登録します（接続制約名を'ALLOWUSR01'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”ALLOWUSR01” FOR ’192.168.0.1’ TO USER1
```

3. 2.を登録したことによって，USER1には1.の接続制限が適用されなくなるため，ユーザUSER1に対し，10.0.0.*を満たすIPアドレスからの接続を許可する接続制限（認可識別子を指定）を登録します（接続制約名を'AREAUSR02'とします）。

```
CREATE CONNECTION SECURITY FOR CONSTRAINT ”AREAUSR02” FOR ’10.0.0.*’ TO USER1
```

25.14　パスワード有効期間の設定方法

25.14.1　パスワード有効期間の設定

パスワードの有効期間の設定は DBA 権限保持者が行います。

パスワードの有効期間を設定するには，CREATE CONNECTION SECURITY FOR INTERVAL PASSWORD 文を実行します。設定時に指定する項目を次の表に示します。

情報	内容
認可識別子	有効期間の設定対象のユーザの認可識別子を設定します。 次のユーザに対しては有効期間の設定はできません。 • 簡易認証ユーザ • パスワードのないユーザ
有効期限間隔	パスワードの有効期限間隔を日数で指定します。 指定範囲は 1～999 です。

構文規則についてはマニュアル「HiRDB SQL リファレンス」の「CREATE CONNECTION SECURITY」を参照してください。

25.14.2　設定内容の格納場所と参照方法

設定情報は，ディクショナリ表（SQL_USERS）に格納します。設定内容は，DBA 権限保持者又は監査人が，SQL でディクショナリ表（SQL_USERS）の有効期間を設定したユーザの行データを参照することで確認できます。参照できる項目を次の表に示します。

表 25–7　パスワードの有効期間の参照項目

ディクショナリ表名	項目
SQL_USERS	PWD_INTERVAL_DAY（パスワードの有効期限間隔）
	PWD_LAST_CHANGE（パスワードの最終更新日）

25.14.3　パスワード有効期間の削除

DBA 権限保持者が DROP CONNECTION SECURITY FOR INTERVAL PASSWORD 文を実行します。

このとき，対象となるユーザのパスワードの最終更新日もクリアします。

構文規則についてはマニュアル「HiRDB SQL リファレンス」の「DROP CONNECTION SECURITY」を参照してください。

25.14.4　パスワードの有効期間の設定変更

DBA 権限保持者が DROP CONNECTION SECURITY FOR INTERVAL PASSWORD 文を実行し，対象となるユーザの有効期間の設定を削除してから，CREATE CONNECTION SECURITY FOR INTERVAL PASSWORD 文を実行し設定を変更します。

25.14.5　有効期限日時の確認方法

ディクショナリ表（SQL_USERS）に格納してあるパスワードの最終更新日と，パスワードの有効期限間隔から算出してください。パスワード最終更新日（YYYYMMDDHHMMSS）に対して，パスワードの有効期限間隔（日数）を秒単位で加算した日時を有効期限日時（YYYYMMDDHHMMSS）とします。

25.14.6　有効期間が過ぎてしまった場合の対処

有効期間が過ぎたユーザのパスワードを，DBA 権限保持者が GRANT 文でパスワードを強制変更してください。または，有効期間が過ぎたユーザに対する DROP CONNECTION SECURITY FOR INTERVAL PASSWORD 文を実行し，有効期間の設定を削除してください。

接続中に有効期限が過ぎてしまった場合は，DISCONNECT するまではその接続が保持され，次回 CONNECT 時に接続できなくなります。

25.14.7　パスワード有効期間の運用例

パスワード有効期間の運用例を次に示します。

1. USERA に対してパスワードの有効期限間隔 10 日として有効期間を設定します。

```
CREATE CONNECTION SECURITY FOR INTERVAL PASSWORD TO USERA (10 DAY)
```

25.14.8　パスワード有効期間運用時の注意事項

1. すべての DBA 権限保持者に有効期間を設定しないようにしてください。または，すべての HiRDB ユーザに有効期間を設定した場合，全員の有効期間が切れないように注意してください。全員の有効期間が過ぎてしまった場合，HiRDB に CONNECT できるユーザがいなくなります。
2. すべてのユニット間の時刻を合わせてください。

25.15　CONNECT 関連セキュリティ機能運用時の注意事項

ここでは，CONNECT 関連セキュリティ機能運用時の注意事項について説明します。説明する項目は次のとおりです。

- 二重ロックの解除
- ディクショナリ用 RD エリアを回復するときの注意事項

25.15.1　二重ロックの解除

パスワード無効アカウントロック状態，及び連続認証失敗アカウントロック状態が同時に掛かることがあります。この状態を**二重ロック**といいます。

二重ロックを解除するには，パスワード無効アカウントロック状態，及び連続認証失敗アカウントロック状態の両方をそれぞれ解除する必要があります。解除の順番はどちらが先でもかまいません。

25.15.2　ディクショナリ用 RD エリアを回復するときの注意事項

ディクショナリ用 RD エリアのバックアップ取得時点から，パスワード無効アカウントロック状態，連続認証失敗アカウントロック状態，IP アドレスによる接続制限の登録状態が変わっている可能性があります。そのため，ディクショナリ用 RD エリアを回復するときは，ディクショナリ用 RD エリアのバックアップとアンロードログファイルを使用して最新の状態に回復してください。

26 系切り替え機能の運用

この章では，系切り替え機能の環境設定，及び運用方法について説明します。なお，この章の説明はクラスタソフトウェアについての知識があることを前提にしています。

26.1　系切り替え機能の概要

システムの信頼性向上，稼働率向上を目的とした機能に系切り替え機能があります。ここでは，HiRDB の系切り替え機能について説明します。

26.1.1　系切り替え機能とは

業務処理中の HiRDB に障害が発生した場合，待機用の HiRDB に業務処理を自動的に切り替えて運用できます。これを**系切り替え機能**といいます。業務処理が中断するのは障害発生時から待機用の HiRDB に処理が切り替わるまでです。障害発生時のシステム停止時間をなるべく短くしたい場合に系切り替え機能を使用します。

系切り替え機能は，監視対象とする障害によってモニタモード又はサーバモードで運用します。系切り替え機能の種類と運用方法の組み合わせを，次の表に示します。

表 26–1　系切り替え機能の種類と運用方法の組み合わせ

系切り替え機能の種類		運用方法	
		モニタモード	サーバモード
スタンバイ型系切り替え機能	通常の系切り替え	○	○
	ユーザサーバホットスタンバイ	×	○
	高速系切り替え	×	○
スタンバイレス型系切り替え機能※	1：1 スタンバイレス型系切り替え	×	○
	影響分散スタンバイレス型系切り替え	×	○

（凡例）

○：運用できます。

×：運用できません。

注※

スタンバイレス型系切り替え機能を使用する場合は，HiRDB Advanced High Availability が必要です。

(1)　系切り替え機能の種類

HiRDB の系切り替えには，スタンバイ型系切り替え機能とスタンバイレス型系切り替え機能があります。

スタンバイ型系切り替え機能

業務処理中の HiRDB のほかに待機用の HiRDB を準備して，業務処理中の HiRDB に障害が発生した場合，待機用の HiRDB に業務処理を自動的に切り替えて運用します。通常の系切り替えより系切り替えに掛かる時間を短縮したい場合は，ユーザサーバホットスタンバイ，高速系切り替え機能を使用するこ

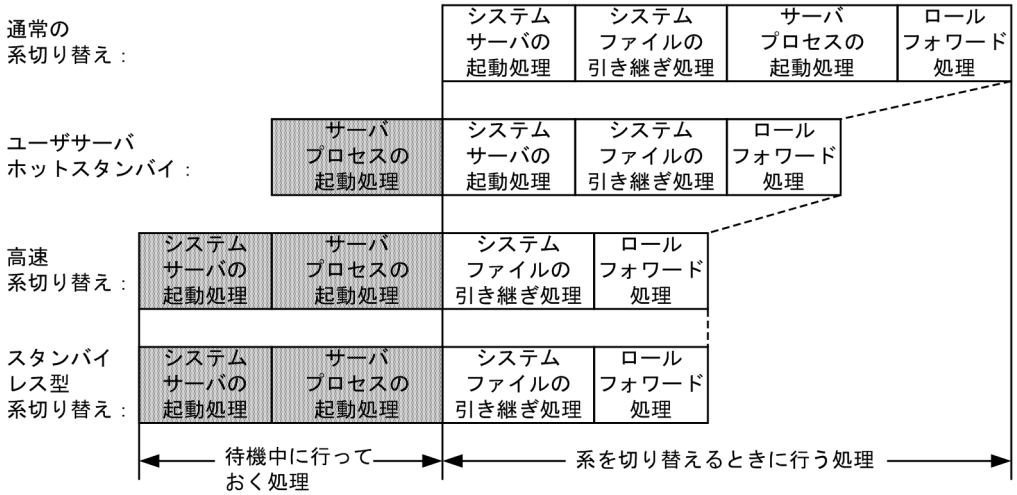
ともできます。スタンバイ型系切り替え機能の詳細は，「スタンバイ型系切り替え機能」を参照してください。

スタンバイレス型系切り替え機能

業務処理中の HiRDB に障害が発生した場合，稼働中のほかのユニットに系を切り替えて運用します。待機用の HiRDB を準備する必要はありません。また，HiRDB/パラレルサーバだけに適用できます。スタンバイレス型系切り替え機能には，1：1 スタンバイレス型系切り替え機能，影響分散スタンバイレス型系切り替え機能があります。詳細は，「スタンバイレス型系切り替え機能」を参照してください。

通常の系切り替えよりも，ユーザサーバホットスタンバイ，高速系切り替え機能，及びスタンバイレス型系切り替え機能を使用すると，システム停止時間が短縮できます。系の切り替え時間の比較を次の図に示します。

図 26–1　系の切り替え時間の比較



〔説明〕

あらかじめ網掛け部分の処理を実行して待機するため，系の切り替え時に網掛け部分の処理が不要になります。その分だけ系の切り替え時間が短縮されます。

(2)　系切り替え機能の運用方法

系切り替え機能の運用方法には**モニタモード**と**サーバモード**があります。モニタモードの場合は系障害だけを監視対象とし，サーバモードの場合は系障害及びサーバ障害を監視対象とします。また，サーバモードではモニタモードに比べて系の切り替え時間を短縮できます。モニタモードとサーバモードが監視対象とする障害を次の表に示します。

表 26–2　モニタモードとサーバモードが監視対象とする障害

系切り替え機能の運用方法	監視対象とする障害	
	系障害※1	サーバ障害※2
モニタモード	○	×
サーバモード	○	○

（凡例）

○：監視します。

×：監視しません。

注※1

ここでは次に示す障害を系障害として想定していますが，系障害の条件はクラスタソフトウェアの仕様に依存します。クラスタソフトウェアのマニュアルなどで確認してください。

- ハードウェアの障害
- OS の障害
- 電源断
- クラスタソフトウェアの障害
- 系のスローダウン

注※2

ここでは次に示す障害をサーバ障害として想定していますが，サーバ障害の条件はクラスタソフトウェア又は Hitachi HA Toolkit Extension の仕様に依存します。クラスタソフトウェアのマニュアル，又はマニュアル「Hitachi HA Toolkit」などで確認してください。

- HiRDB（HiRDB/パラレルサーバの場合はユニット）の異常終了
- HiRDB（HiRDB/パラレルサーバの場合はユニット）のスローダウン
- データベースのパス障害

(a)　モニタモード，及びサーバモードで運用できるクラスタソフトウェア

系切り替えを実行する製品をこのマニュアルではクラスタソフトウェアといいます。HiRDB がサポートしているクラスタソフトウェアと，モニタモード及びサーバモードでの運用可否を次の表に示します。

表 26–3　クラスタソフトウェアと，モニタモード及びサーバモードでの運用可否

クラスタソフトウェア	モニタモード	サーバモード
MSFC	○	○
CLUSTERPRO	○	○

（凡例）

○：運用できます。

×：運用できません。

HiRDB がサポートしているクラスタソフトウェアの詳細については，「HiRDB がサポートしているクラスタソフトウェア」を参照してください。

(b)　サーバモードで運用する場合に必要な製品

系切り替え機能をサーバモードで運用する場合は，次の表に示す製品が必要になります。

表 26–4　サーバモードで運用する場合に必要な製品

系切り替え機能の種類	HiRDB Advanced High Availability	Hitachi HA Toolkit Extension
通常の系切り替え	–	○
ユーザサーバホットスタンバイ	–	○
高速系切り替え機能	–	○
1：1 スタンバイレス型系切り替え機能	○	○
影響分散スタンバイレス型系切り替え機能	○	○

（凡例）

○：この機能を使用する場合に必要な製品です。

–：必要ありません。

(c)　注意事項

1. モニタモードの場合は系障害だけを監視対象にし，サーバ障害は監視対象外になります。ただし，次のどちらかの設定の場合，系を切り替えられます。
   - クラスタソフトウェアに MSFC を使用して，HiRDB のサービスを MSFC に登録する場合
   - クラスタソフトウェアに CLUSTERPRO を使用して，カスタム監視リソースでフェイルオーバする設定の場合
2. Hitachi HA Toolkit Extension が HiRDB の状態を監視し，クラスタソフトウェアは，Hitachi HA Toolkit Extension のサービスプロセスを監視します。
3. **サーバモードの場合は IP アドレスを引き継がない構成になります。**

26.1.2　スタンバイ型系切り替え機能

業務処理中の HiRDB のほかに待機用の HiRDB を準備して，業務処理中の HiRDB に障害が発生した場合，待機用の HiRDB に業務処理を自動的に切り替えます。これを**スタンバイ型系切り替え機能**といいます。

スタンバイ型系切り替え機能は複数のサーバマシンを使用したクラスタシステムの構成で実現します。HiRDB/シングルサーバの場合はシステム単位で系を切り替えます。HiRDB/パラレルサーバの場合はユニット単位で系を切り替えます。なお，ユーザサーバホットスタンバイ，又は高速系切り替え機能を使用すると，系切り替えに掛かる時間が短縮できます。ユーザサーバホットスタンバイ，高速系切り替え機能については，「系の切り替え時間の短縮（ユーザサーバホットスタンバイ，高速系切り替え機能）」を参照してください。

なお，業務処理中の系を**実行系**，待機中の系を**待機系**といい，系の切り替えが発生するたびに実行系と待機系が入れ替わります。また，システム構築時や環境設定時に二つの系を区別するため，最初に実行系として起動する系を**現用系**，待機系として起動する系を**予備系**といいます。系が切り替わると実行系と待機

系は変わりますが，現用系と予備系は変わりません。系切り替え機能（スタンバイ型系切り替え機能）には次の三つの構成があります。

- 共有ディスク装置を使用した系切り替え構成
- DataKeeper※による複製ディスクを使用した系切り替え構成
- クラスタソフトウェアとミラーディスクを使用した系切り替え構成

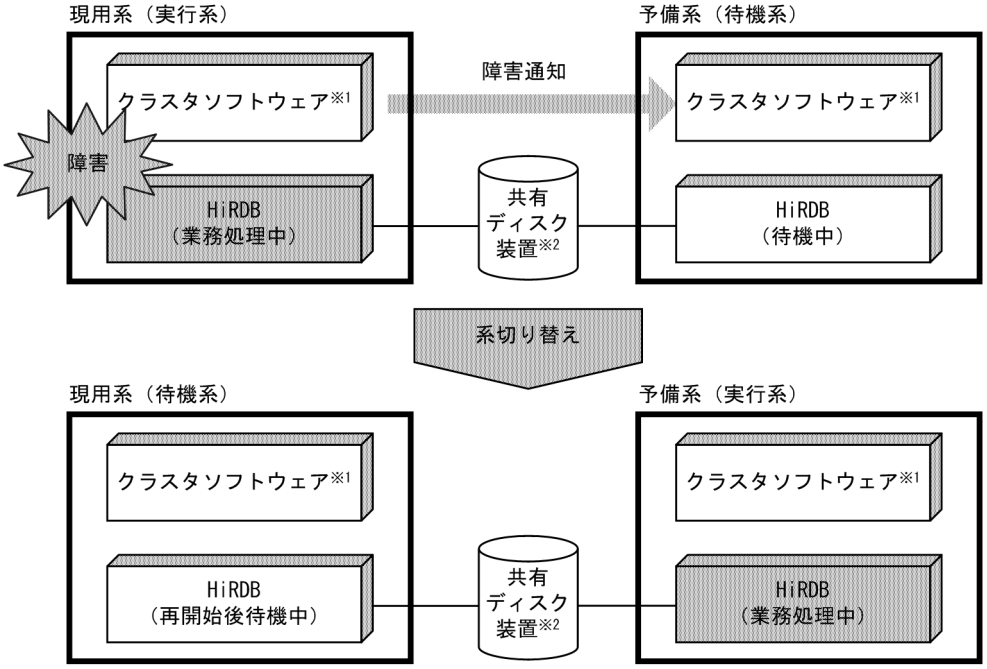
注※

SIOS DataKeeper Cluster Edition

(1)　共有ディスク装置を使用した系切り替え構成

共有ディスク装置を使用した系切り替え構成の概要を次の図に示します。

図 26–2　系切り替え機能（スタンバイ型系切り替え機能）の概要



注※１

系切り替えを実行する製品をこのマニュアルでは**クラスタソフトウェア**といいます。ここでは MSFC（Microsoft Failover Cluster），又は CLUSTERPRO のことを意味しています。

注※２

共有ディスク装置については，「共有ディスク装置の準備」を参照してください。

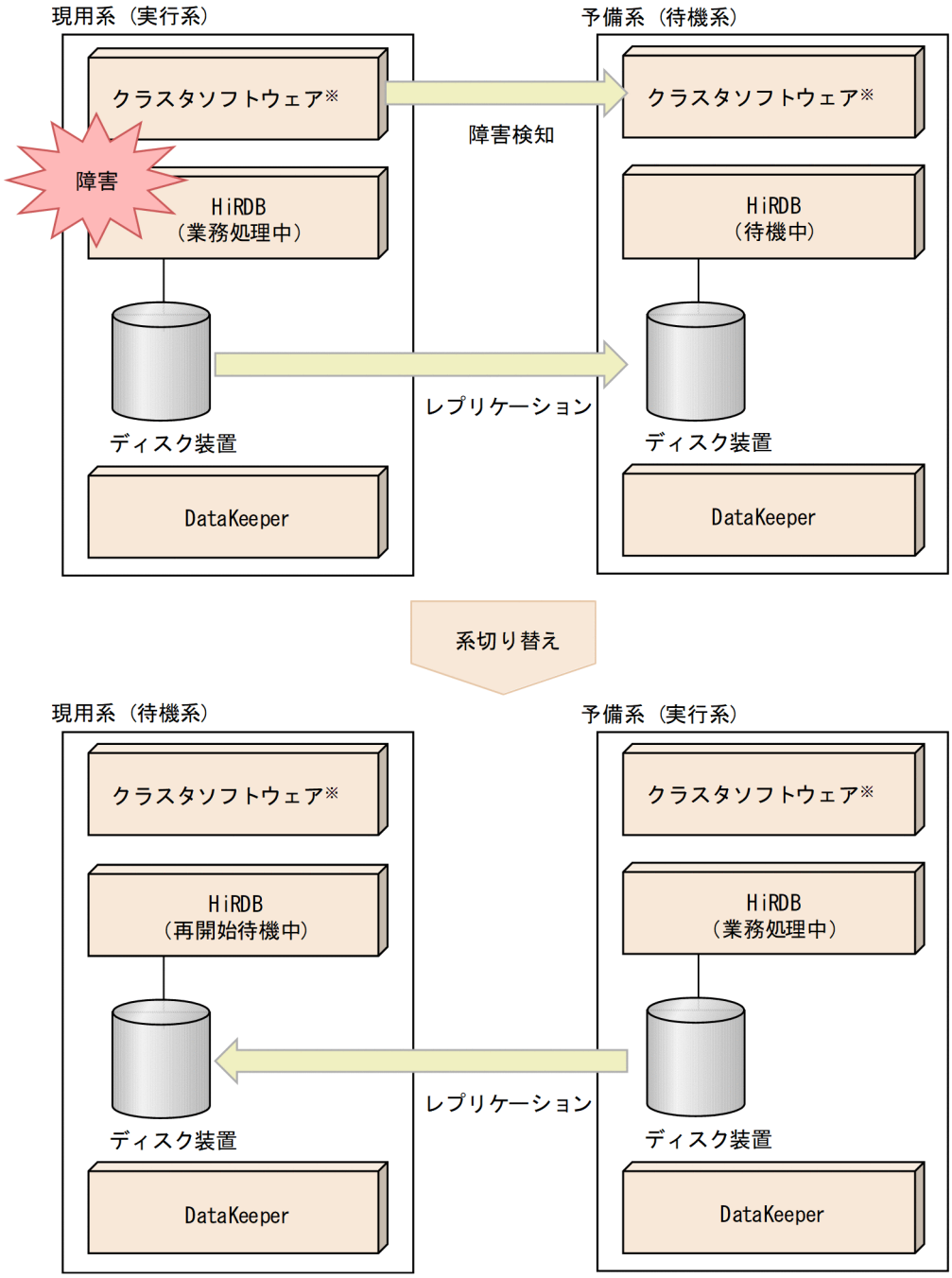
〔説明〕

業務処理中の実行系に障害が発生すると，待機系に障害の発生が通知されて系が切り替わり，待機系が実行系になって業務処理を続行します。

(2)　DataKeeper による複製ディスクを使用した系切り替え構成

DataKeeper による複製ディスクを使用できます。DataKeeper による複製ディスクとは，DataKeeper のデータレプリケーション機能によって実行系の更新内容を待機系に複製するディスクのことです。
DataKeeper による複製ディスク使用時の系切り替え機能の概要を次の図に示します。

図 26–3　DataKeeper による複製ディスク使用時の系切り替え機能（スタンバイ型系切り替え機能）の概要



注※

系切り替えを実行する製品をこのマニュアルではクラスタソフトウェアといいます。ここでは MSFC（Microsoft Failover Cluster）のことを意味しています。

〔説明〕

業務処理中の実行系に障害が発生すると，待機系が障害を検知して系が切り替わり，待機系が実行系になって業務処理を続行します。

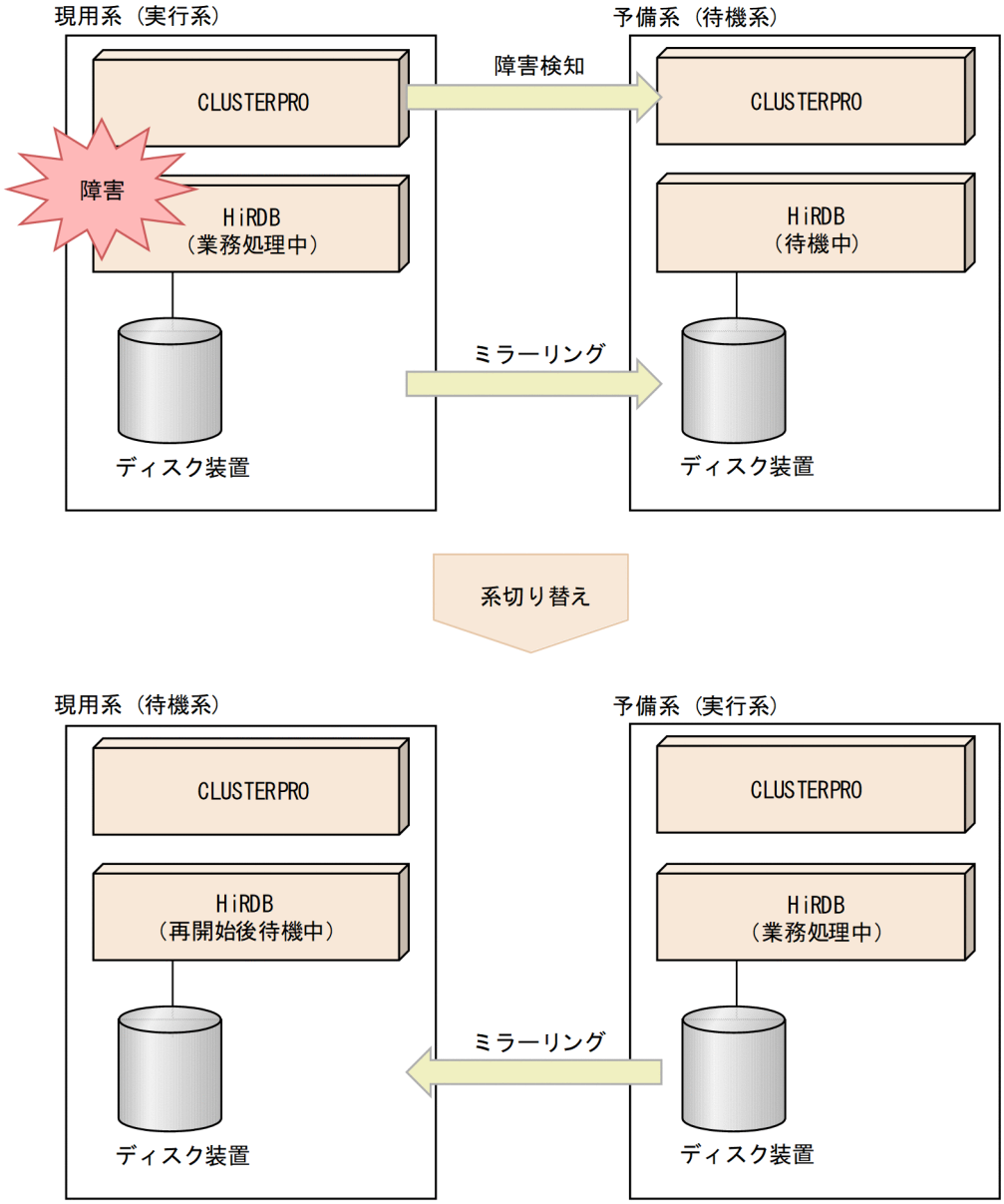
実行系のディスク装置に対して更新すると，DataKeeper は更新内容を待機系に送信し，待機系のディスク装置に複製します。そのため，実行系の障害によって系が切り替わった場合，障害発生直前の状態から業務を続行できます。

DataKeeper による複製ディスクを使用した系切り替え構成の設定方法は，「MSFC 及び DataKeeper に関する準備」を参照してください。

(3)　クラスタソフトウェアとミラーディスクを使用した系切り替え構成

CLUSTERPRO によるミラーディスクを使用できます。CLUSTERPRO によるミラーディスクとは，CLUSTERPRO のミラーリングによって実行系の更新内容を待機系に複製するディスクのことです。CLUSTERPRO によるミラーディスク使用時の系切り替え機能の概要を次の図に示します。

図 26–4　CLUSTERPRO によるミラーディスク使用時の系切り替え機能（スタンバイ型系切り替え機能）の概要



〔説明〕

業務処理中の実行系に障害が発生すると，待機系が障害を検知して系が切り替わり，待機系が実行系になって業務処理を続行します。

実行系のディスク装置に対して更新すると，CLUSTERPRO は更新内容を待機系に送信し，待機系のディスク装置に複製します（ミラーリング）。そのため，実行系の障害によって系が切り替わった場合，障害発生直前の状態から業務を続行できます。

CLUSTERPRO による複製ディスクを使用した系切り替え構成の設定方法の詳細は，「CLUSTERPRO に関する準備」を参照してください。

26.1.3　スタンバイレス型系切り替え機能

業務処理中の HiRDB に障害が発生した場合，ほかのユニットに系を切り替えて稼働中のバックエンドサーバに処理を代行させます。これを**スタンバイレス型系切り替え機能**といいます。スタンバイレス型系切り替え機能ではスタンバイ型系切り替え機能とは異なり，待機系ユニットを準備する必要がありません。

スタンバイレス型系切り替え機能には，次の二つの機能があります。

- 1：1 スタンバイレス型系切り替え機能
- 影響分散スタンバイレス型系切り替え機能

スタンバイレス型系切り替え機能は HiRDB/パラレルサーバのバックエンドサーバユニットに対して適用できます。ユニット内にバックエンドサーバ以外のサーバがある場合はそのユニットにスタンバイレス型系切り替え機能を適用できません。

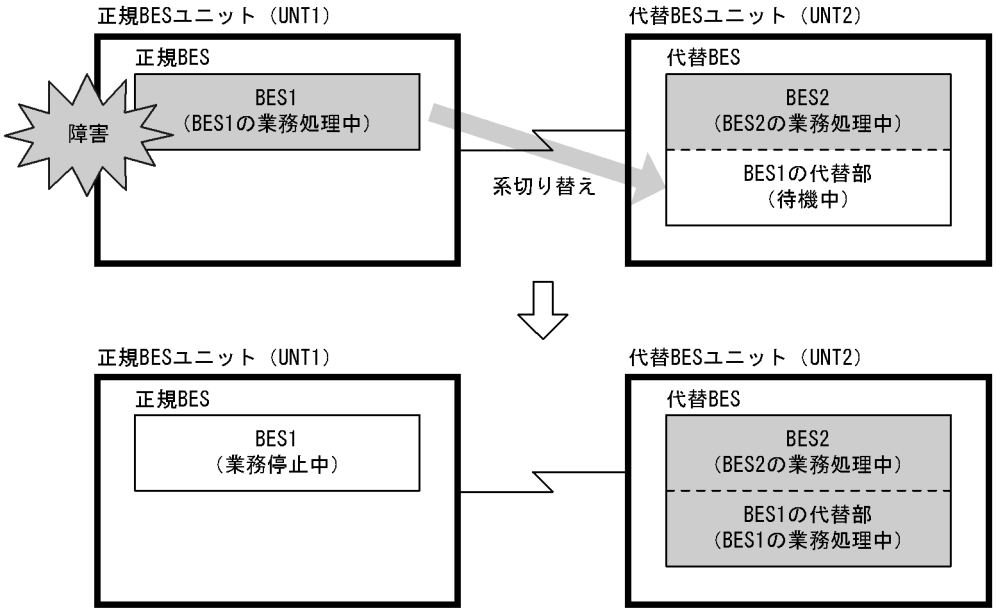
スタンバイレス型系切り替え機能ではスタンバイ型系切り替え機能とは異なり，待機系ユニットを準備する必要がありません。障害が発生した場合は待機系ユニットに系を切り替えるのではなく，ほかのユニットに系を切り替えて稼働中のバックエンドサーバに処理を代行させます。これを**スタンバイレス型系切り替え機能**といいます。

(1)　1：1 スタンバイレス型系切り替え機能

1：1 スタンバイレス型系切り替え機能では，障害が発生したユニットを 1：1 に切り替えて別のバックエンドサーバに処理を代行させることができます。

なお，障害発生時に処理を代行してもらうバックエンドサーバを**正規 BES** といい，処理を代行するバックエンドサーバを**代替 BES** といいます。また，正規 BES のユニットを**正規 BES ユニット**といい，代替 BES のユニットを**代替 BES ユニット**といいます。1：1 スタンバイレス型系切り替え機能の概要を次の図に示します。

図 26–5　1：1 スタンバイレス型系切り替え機能の概要



〔説明〕

- 通常は BES1 及び BES2 の両方で処理を行います。
- 正規 BES ユニット（UNT1）に障害が発生した場合，系を切り替えて代替 BES で処理を代行します。処理を代行する部分を**代替部**といい，代替部で処理を行っているときを**代替中**といいます。
- 障害対策後に正規 BES ユニットを開始して，代替 BES で代行していた処理を正規 BES に切り替えて正常状態に戻します。これを**系の切り戻し**といいます。

ポイント

スタンバイ型系切り替え機能にある現用系などの概念と比較すると，1：1 スタンバイレス型系切り替え機能では次のようになります。

- 現用系が正規 BES ユニット，予備系が代替 BES ユニットと考えてください。
- 正常時は正規 BES ユニットが実行系で，代替部が待機系と考えてください。代替中は代替部が実行系で，正規 BES ユニットが待機系と考えてください。

参考

系の切り替え先に稼働中のユニットを利用するため，待機用のサーバマシンが不要になります。そのため，スタンバイレス型系切り替え機能では IP アドレスの引き継ぎが発生しません。

(a)　前提条件

1：1 スタンバイレス型系切り替え機能を使用する場合は次に示す前提条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入している

- Hitachi HA Toolkit Extension を導入している
- 系切り替え機能をサーバモードで運用している

(b)　スタンバイ型系切り替え機能と比較して優れている点

1：1 スタンバイレス型系切り替え機能はスタンバイ型系切り替え機能に比べて次に示す点が優れています。

- 待機系ユニットを準備する必要がないため，システムリソースを効率的に使用できます。ただし，系が切り替わると代替 BES ではその分の負荷が大きくなるため，処理性能に影響を与えることがあります。
- サーバプロセス及びシステムサーバをあらかじめ起動しておくため，系の切り替え時間を高速系切り替え機能使用時と同じくらいに短縮できます。高速系切り替え機能については，「系の切り替え時間の短縮（ユーザサーバホットスタンバイ，高速系切り替え機能）」を参照してください。

待機系ユニットの待機時の所要リソースと系切り替え後の所要リソースを次の表に示します。

表 26–5　待機系ユニットの待機時の所要リソースと系切り替え後の所要リソース

項目			HiRDB システムサーバプロセス	HiRDB サーバプロセス	ユニットコントローラ用共用メモリ	排他制御用プール用共用メモリ	グローバルバッファ用共用メモリ
サーバモード	1：1 スタンバイレス型系切り替え機能		●※1	–※2※3	●※4	●	–※5
	影響分散スタンバイレス型系切り替え機能		△※6※7	–※3※8	●※9	●	–※10
	スタンバイ型系切り替え機能	ユーザサーバホットスタンバイ	○	●	●	○	○
		高速系切り替え機能	●	●	●	●	●
		それ以外	○	○	●	○	○
モニタモード			○	○	○	○	○

（凡例）

●：待機完了までに確保して，系の切り替え後も使用します。

○：系の切り替え後，実行系になった時点で確保して使用します。

△：一部のリソースを系切り替え後に実行系となった時点で確保し，使用します。

–：確保しません。

注※1

システムサーバプロセスのうち，幾つかのプロセスは待機時点でプロセスを生成します。それ以外のシステムサーバは代替 BES ユニットのシステムサーバプロセスを共有するため，代替部専用の所要リソースはありません。

注※2

バックエンドサーバプロセスの上限値は，代替中及び非代替中の合計が代替 BES の pd_max_bes_process の値になります。そのため，系の切り替え後に接続ユーザ数が制限されることがあります。

注※3

pd_process_count の値（常駐プロセス数）＜ pd_max_bes_process の値でかつ，系切り替え発生時点で起動済みのバックエンドサーバプロセス数が pd_max_bes_process の値に満たない場合は，バックエンドサーバプロセスの追加起動が発生します。そのため，系切り替え発生後に OS のプロセス数，仮想メモリ，ポートなどが不足しないように環境設定をしてください。また，バックエンドサーバプロセスの追加起動によって，系切り替え発生直後の性能が一時的に低下することがあります。

注※4

代替 BES ユニットの開始時に代替部の共用メモリが確保されます。

注※5

代替 BES が使用しているグローバルバッファを代替中に共用します。そのため，系切り替え後には確保しません。代替中のグローバルバッファの割り当て方式については，「グローバルバッファの定義（1：1 スタンバイレス型系切り替え機能限定）」を参照してください。

注※6

ユニット単位のシステムサーバプロセスを受け入れユニットと共用するため，ゲスト用領域専用としての所要リソースはありません。

注※7

バックエンドサーバ対応のシステムサーバプロセスは，実行系となった時点でプロセスを生成します。

注※8

系切り替え後のユニット内のサーバプロセス数の上限値については，通常バックエンドサーバ用プロセス数とゲスト BES 用プロセス数の合計値として定義できます（pd_ha_max_server_process）。

注※9

受け入れユニット起動時，ゲスト用領域分の共用メモリが確保されます。

注※10

受け入れユニットに対して通常バックエンドサーバが使用しているグローバルバッファを共用する場合に共用します。そのため，系切り替え後には確保しません。共用時のグローバルバッファの共用については，「グローバルバッファの定義（影響分散スタンバイレス型系切り替え機能限定）」を参照してください。

影響分散スタンバイレス型系切り替え機能適用時のバックエンドサーバのリソース使用状況については，「影響分散スタンバイレス型系切り替え機能」を参照してください。

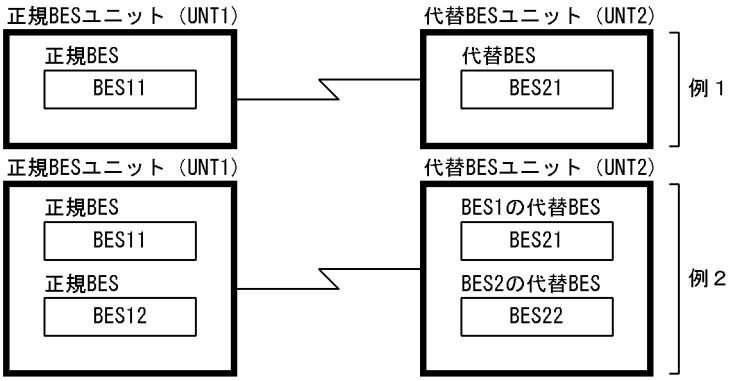
(c)　正規 BES ユニット及び代替 BES ユニット定義時の規則

正規 BES ユニット及び代替 BES ユニット定義時の規則を次に示します。

- 正規 BES ユニット及び代替 BES ユニット内にはバックエンドサーバだけを定義できます。バックエンドサーバ以外のサーバがある場合はそのユニットにスタンバイレス型系切り替え機能を適用できません。
- 正規 BES ユニットと代替 BES ユニットは 1 対 1 の関係にしてください。
- 正規 BES と代替 BES は 1 対 1 の関係にしてください。
- 正規 BES ユニット内に複数の正規 BES を定義できます。この場合，代替 BES ユニット内にも同数の代替 BES を定義してください。

正規 BES ユニット及び代替 BES ユニットの正しい構成例を次の図に，間違った構成例を図「正規 BES ユニット及び代替 BES ユニットの間違った構成例」に示します。

図 26–6　正規 BES ユニット及び代替 BES ユニットの正しい構成例



代替 BES は pdstart オペランドの-c オプションで定義します。上記の図の例 1 及び例 2 の pdstart オペランドの指定例を次に示します。

例 1

```
pdstart -t BES -s bes11 -u UNT1 -c bes21
pdstart -t BES -s bes21 -u UNT2
```

〔説明〕

-s bes11：正規 BES を指定します。

-c bes21：代替 BES を指定します。

例 2

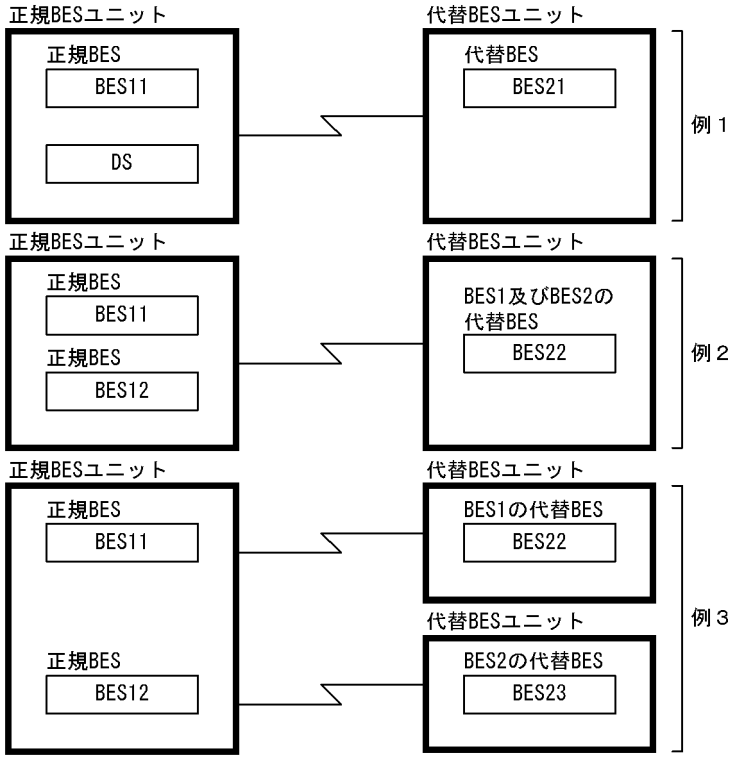
```
pdstart -t BES -s bes11 -u UNT1 -c bes21
pdstart -t BES -s bes12 -u UNT1 -c bes22
pdstart -t BES -s bes21 -u UNT2
pdstart -t BES -s bes22 -u UNT2
```

〔説明〕

-s bes11，-s bes12：正規 BES を指定します。

-c bes21，-c bes22：代替 BES を指定します。

図 26–7　正規 BES ユニット及び代替 BES ユニットの間違った構成例



(2)　影響分散スタンバイレス型系切り替え機能

(a)　機能概要

障害発生時に障害ユニット内のバックエンドサーバへの処理要求を，複数の稼働中ユニットに分散して実行させる機能を影響分散スタンバイレス型系切り替え機能といいます。影響分散スタンバイレス型系切り替え機能では，待機用サーバマシン，又は待機ユニットを準備する必要はなく，システムリソースを効率的に利用できます。障害発生後，障害ノードのサーバ処理を代行するユニットでは処理負荷が増えるため，トランザクション処理性能に影響を及ぼすことがあります。ただし，複数のユニットが障害ユニット内サーバへの処理要求を分担して実行することで，ユニット当たりの負荷上昇を抑え，システムの性能劣化を軽減します。

また，影響分散スタンバイレス型系切り替え機能では，バックエンドサーバを分散して切り替えられます。切り替え先を複数のユニットに分散させることもできます。さらに，障害発生によって切り替えた先のユニットで更に障害が発生しても，別の稼働中ユニットに更に切り替わることで処理を継続できます（以降，多段系切り替えといいます）。なお，1：1 スタンバイレス型系切り替えの場合は多段系切り替えができないため，切り替え先で障害が発生すると障害ユニットの代行処理は継続できなくなります。

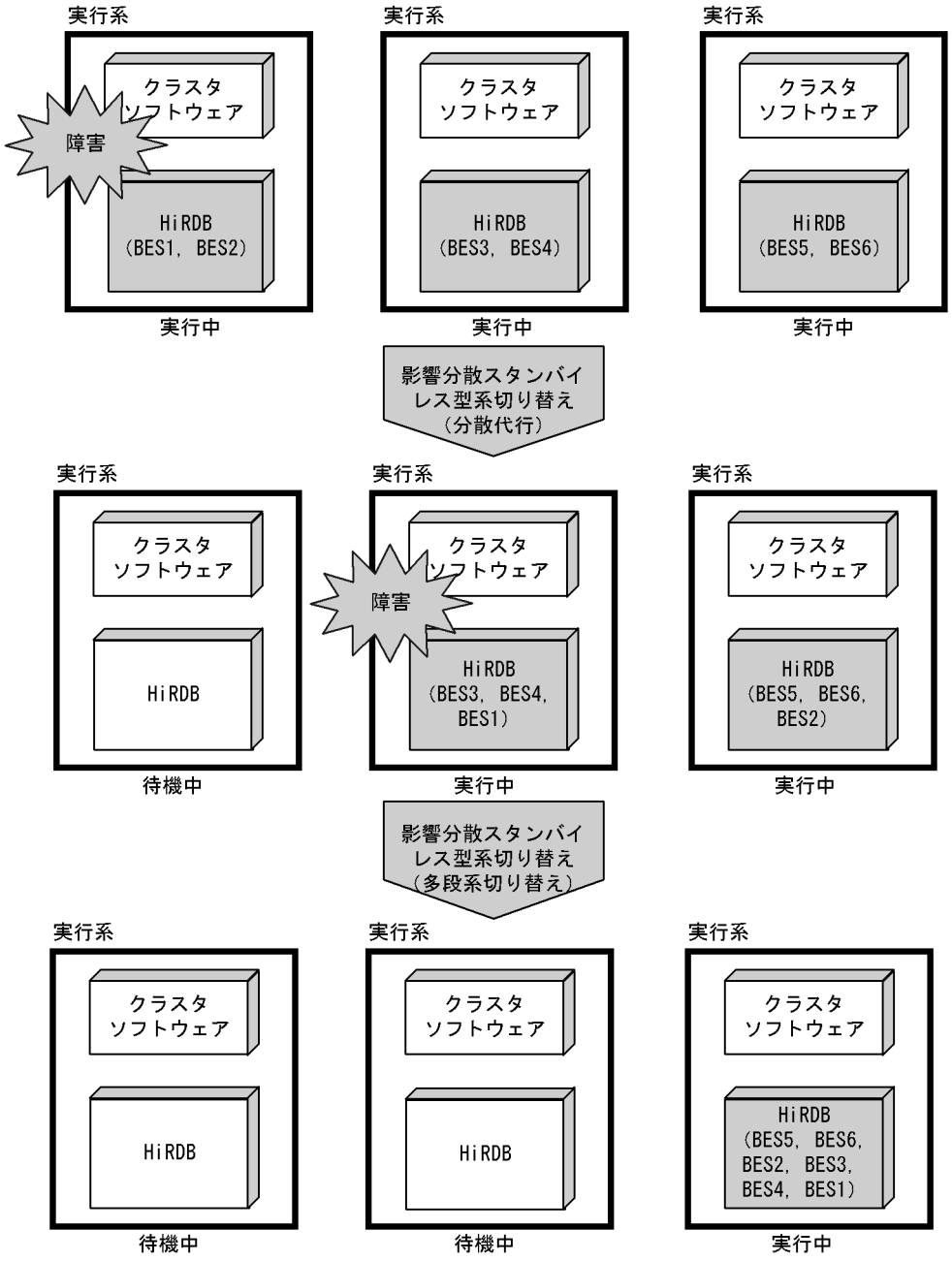
影響分散スタンバイレス型系切り替え機能は，通常時からシステムリソースを有効に利用することを重視し，しかも，障害発生時の性能劣化を最小限に抑える必要があるシステムに対して適用してください。

なお，影響分散スタンバイレス型系切り替え機能では，そのユニットに定義してあるバックエンドサーバを**ホスト BES** といい，定義してあるユニットとは別のユニットに受け入れてもらっているバックエンドサーバを**ゲスト BES** といいます。ホスト BES のユニットを**正規ユニット**といい，ゲスト BES のユニット

を**受け入れユニット**といいます。受け入れユニットのすべては **HA グループ**として定義しておく必要があります。また，ゲスト BES に対応付けられるバックエンドサーバ用のリソースを**ゲスト用領域**といいます。

影響分散スタンバイレス型系切り替え機能の概要（分散代行，多段系切り替え）を次の図に示します。

図 26–8　影響分散スタンバイレス型系切り替え機能の概要（分散代行，多段系切り替え）



(b)　前提条件

影響分散スタンバイレス型系切り替え機能を使用する場合は次に示す前提条件をすべて満たす必要があります。

- HiRDB Advanced High Availability を導入していることが必要です。

- Hitachi HA Toolkit Extension を導入していることが必要です。
- 系切り替え機能をサーバモードで運用していることが必要です。
- 影響分散スタンバイレス型系切り替え機能は，バックエンドサーバだけから構成されるバックエンドサーバ専用ユニットだけを対象としています。
- 影響分散スタンバイレス型系切り替え機能を適用するユニットは，一つ以上の現用系のバックエンドサーバから構成される必要があります。受け入れ専用のユニットには適用できません。

(c) リソース使用状況

影響分散スタンバイレス型系切り替え機能適用時のバックエンドサーバのリソース使用状況を次の表に示します。

表 26–6 影響分散スタンバイレス型系切り替え機能適用時のバックエンドサーバのリソース使用状況

バックエンドサーバの種別	バックエンドサーバの状態	リソース使用状況
ホスト BES	受け入れ可能状態	各バックエンドサーバの定義に従ったサイズで領域を作成します。
	稼働中	各バックエンドサーバの定義に従ったサイズ分領域を使用します。
ゲスト BES	受け入れ可能状態	リソースごとにゲストサーバ中最大リソースサイズでゲスト用領域を作成します。
	稼働中	用意したゲスト用領域のうち各バックエンドサーバの定義に従った領域を使用します。

(d) 影響分散スタンバイレス型系切り替え機能の動作

影響分散スタンバイレス型系切り替え機能では，正規ユニットに障害が発生した場合，現用 BES はそれぞれ自動的に別々の受け入れユニットに移動してゲスト BES として処理を実行します。また，障害が発生したユニットでゲスト BES が稼働している場合には，そのゲスト BES も自動的に受け入れユニットに移動し，移動先のゲスト BES として処理を実行します。スタンバイ型系切り替え機能と同じく，HiRDB 管理者の操作は不要です。

影響分散スタンバイレス型系切り替えでの障害要因と系切り替えの有無を次の表に示します。

表 26–7 影響分散スタンバイレス型系切り替えでの障害要因と系切り替えの有無

ユニット又はサーバの状態	開始／停止中	稼働中	
		開始／停止中	稼働中
スローダウンの検知	非該当	ユニット異常終了 系切り替えあり	ユニット異常終了 系切り替えあり
システムログの満杯	非該当	ユニット異常終了 系切り替えなし	ユニット異常終了 系切り替えなし

ユニット又はサーバの状態	開始／停止中	稼働中	
		開始／停止中	稼働中
		※	※
データベースのパス障害	非該当	ユニット異常終了 系切り替えあり（1 回目だけ）	ユニット異常終了 系切り替えあり（1 回目だけ）
バックエンドサーバの強制停止	バックエンドサーバの異常終了 系切り替えなし	バックエンドサーバの異常終了 系切り替えなし	バックエンドサーバの異常終了 系切り替えなし
システム強制停止	ユニット異常終了 系切り替えなし	ユニット異常終了 系切り替えなし	ユニット異常終了 系切り替えなし
システム障害	ユニット異常終了 系切り替えなし	ユニット異常終了 系切り替えなし	ユニット異常終了 系切り替えあり

注※

システムログが満杯になった場合，ユニットは異常終了します。このとき，系切り替えはしません。

系切り替えでは，ユニットで稼働中のホスト BES 及びゲスト BES を別のユニットに切り替えます。切り替え先は，バックエンドサーバごとに異なります。

また，影響分散スタンバイレス型系切り替え機能では，多重障害時にも自動的に系を切り替えます。正規ユニット障害後に受け入れユニットで障害が発生した場合，該当の受け入れユニットで稼働していた現用系のバックエンドサーバ及びゲスト BES は残りの稼働中ユニットに移動してゲスト BES として処理を実行します。このとき，HiRDB 管理者の操作は不要です。なお，各バックエンドサーバの移動先は，クラスタソフトウェアの定義によって決まります。

影響分散スタンバイレス型系切り替え機能では，ユニット内に空きゲスト用領域がなくなるとすべての稼働していないゲスト BES の受け入れ可能状態を解除します。ゲスト用領域の受け入れ可能状態は，ホスト BES の動作の影響を受けません。ゲスト用領域の空き状態による受け入れ可能状態の自動解除・再設定を次の表に示します。

受け入れ可能状態に自動再設定をする場合，HA グループ内の他ユニットで実行系となっている全サーバを受け入れて可能状態にします。このとき，それ以前に受け入れ可能状態をコマンド（monsbystp，pdstop -q -s バックエンドサーバ名）で意図的に停止していた場合も受け入れ可能状態となります。また，HA グループ内の受け入れ可能数を超えて縮退していた場合も停止中のサーバは，受け入れ可能状態の対象となりません。

表 26–8　ゲスト用領域の空き状態による受け入れ可能状態の自動解除・再設定

ユニット内未使用ゲスト用領域	ゲスト BES 受け入れ可能状態	
	他ユニットで稼働中のゲスト BES	他ユニットで非稼働中のゲスト BES
消滅	自動解除	変化なし（解除中）

ユニット内未使用ゲスト用領域	ゲスト BES 受け入れ可能状態	
	他ユニットで稼働中のゲスト BES	他ユニットで非稼働中のゲスト BES
発生	自動再設定	変化なし

注

monsbystp コマンド，又は pdstop コマンド（pdstop -u 受け入れユニット ID -s サーバ ID）などで意図的に受け入れ可能状態を解除した場合を除きます。

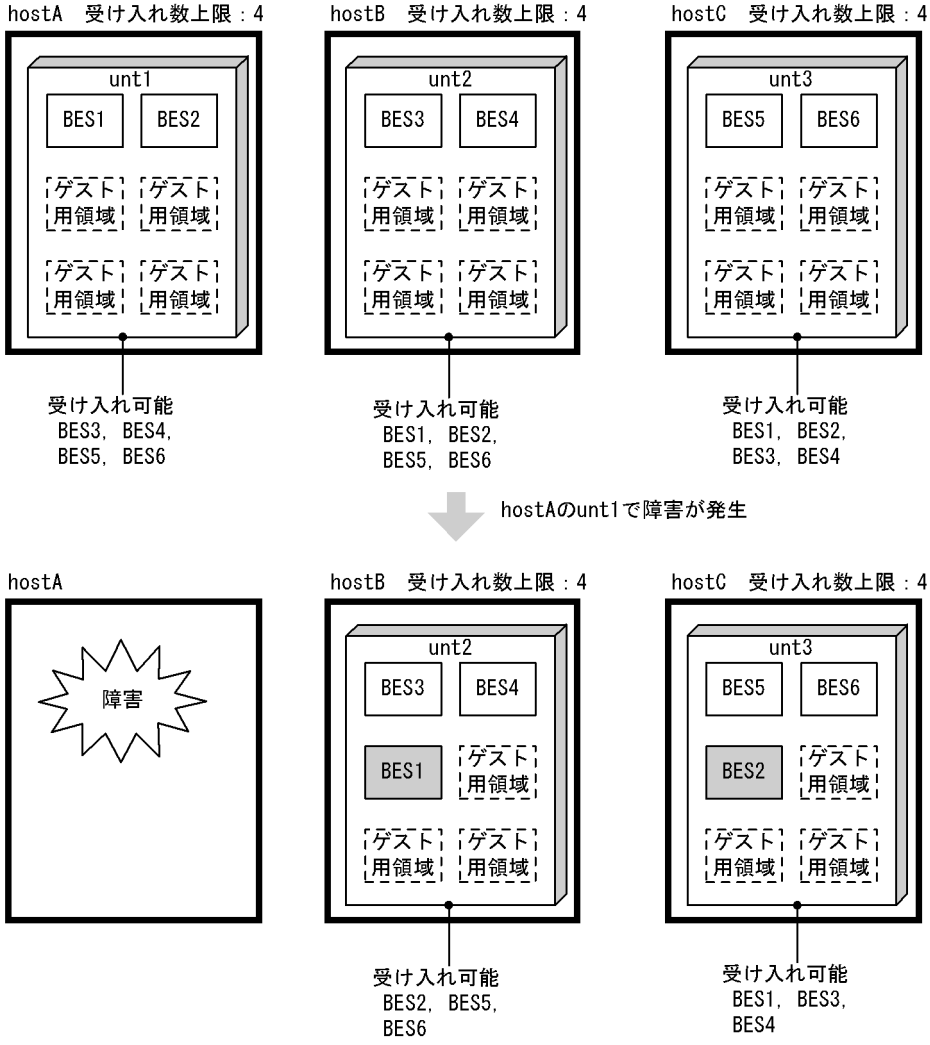
(e)　影響分散スタンバイレス型系切り替え機能の系切り替えの例

●通常時の系切り替えの例

通常時の系切り替えの例を次の図に示します。

hostA で障害が発生すると，BES1 は unt2 に移動してゲスト BES として処理を実行し，BES2 は unt3 に移動してゲスト BES として処理を実行します。

図 26–9　通常時の系切り替えの例



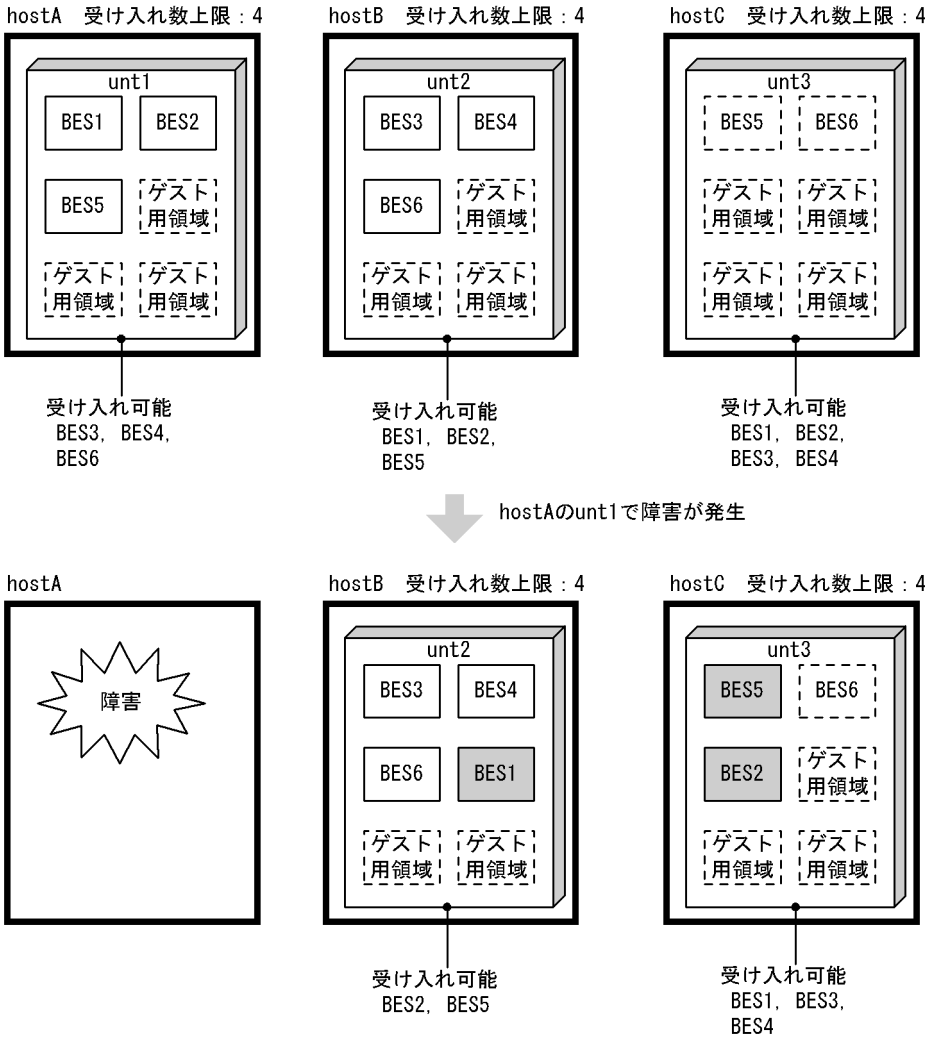
●受け入れ中の系切り替えの例

受け入れ中の系切り替えの例を次の図に示します。この例は，サーバマシンの障害復旧後に，切り戻し実行前の状態で別のサーバマシンに障害が発生した場合に該当します。

unt1 で BES5 がゲスト BES として処理を実行している状態で，hostA で障害が発生すると，各バックエンドサーバは次のように動作します。

- BES1 は unt2 に移動してゲスト BES として処理を実行します。
- BES2 は unt3 に移動してゲスト BES として処理を実行します。
- BES5 は unt3 に戻ってホスト BES として処理を実行します。

図 26–10　受け入れ中の系切り替えの例



ユニット負荷の不均衡

系切り替え発生後にユニット負荷が不均衡になるかどうかはクラスタソフトウェアの定義の待機系の優先度付けに依存します。待機系の優先度を適切に設定すれば，任意の組み合わせで複数ユニットが異常終了した後も負荷を均衡させることができます。

ただし，ユニットが障害から回復した後，ユニット障害後の他ユニット障害による系切り替え後などは，負荷が均衡していない状態になることがあるため，系切り替え後のサーバ配置を確認する必要があります。サーバの配置は pdls -d ha 又は pdls -d svr コマンドで確認できます。

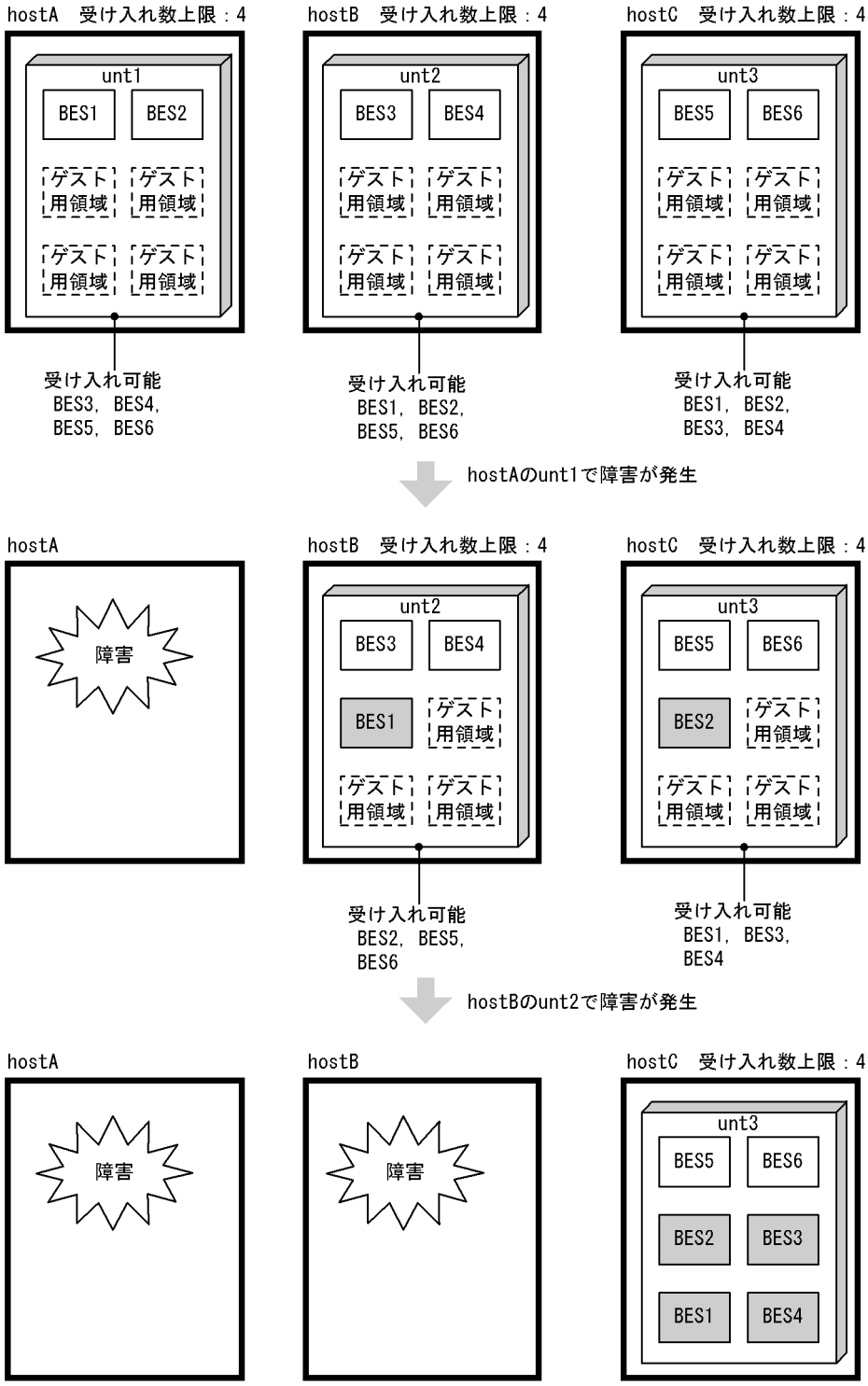
サーバの配置が不均衡な場合には計画系切り替えによってサーバの配置を変更することを推奨します。

また，ユニットの障害が復旧した場合はできるだけ早く各 BES を定義されたユニットに切り戻すことで，サーバの配置が不均衡になるのを回避できます。受け入れ中の系切り替えの例では，hostA で障害が発生する前に BES5，BES6 を unt3 に切り戻しておけば，hostA で障害が発生しても不均衡なサーバ配置にはなりません。

●多重障害時の系切り替え（全バックエンドサーバが受け入れ可能な場合）

多重障害時の系切り替えの例を次の図に示します。

図 26–11　多重障害時の系切り替えの例



〔説明〕

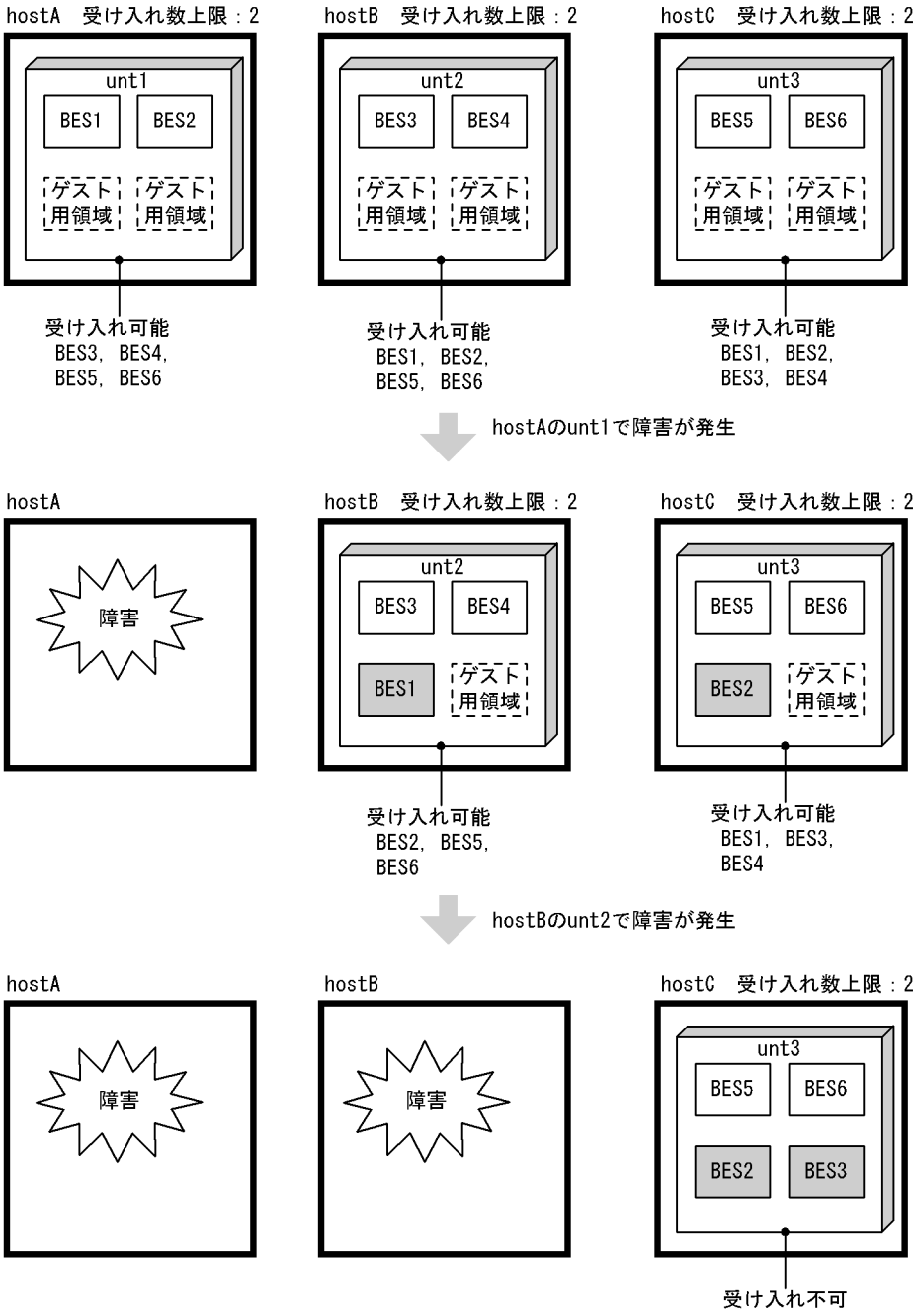
hostA で障害が発生すると，BES1 は unt2 に移動してゲスト BES として処理を実行します。このとき，unt2 及び unt3 の受け入れ数の上限は 4 のため，それぞれ，さらに 3 サーバを受け入れられます。したがって，他ユニットで稼働しているバックエンドサーバをすべて受け入れ可能な状態です。

hostA に続いて hostB で障害が発生すると，unt2 で稼働している BES1，BES3，BES4 は unt3 に移動してゲスト BES として処理を実行します。停止するバックエンドサーバはありません。

●多重障害時の系切り替え（受け入れ数が不足する場合）

多重障害時の系切り替えで受け入れ数が不足する場合の例を次の図に示します。

図 26–12　多重障害時の系切り替えで受け入れ数が不足する場合の例



〔説明〕

hostA で障害が発生すると，BES1 は unt2 に移動してゲスト BES として処理を実行します。このとき，unt2 及び unt3 の受け入れ数の上限は 2 のため，それぞれ，さらに 1 サーバを受け入れられます。したがって，他ユニットで稼働しているバックエンドサーバの一部しか受け入れられない状態です。

hostA に続いて hostB で障害が発生すると，unt2 で稼働している BES3 だけが unt3 に移動してゲスト BES として処理を実行します。BES1 及び BES4 は停止します。

多重障害発生時にも全バックエンドサーバの処理を続行させる必要がある場合には，受け入れ数上限を大きく設定しておく必要があります。

受け入れ数が不足したときの切り替え

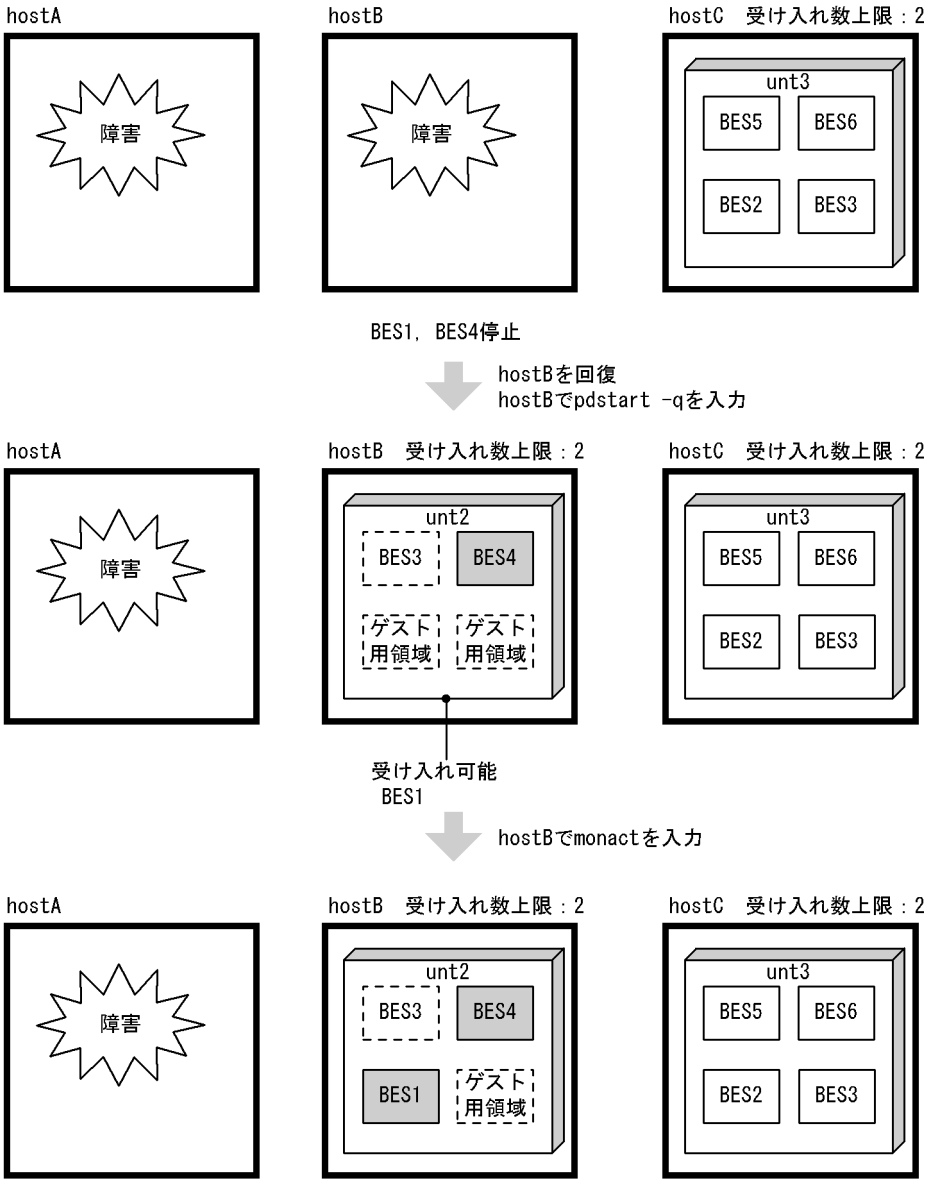
障害発生ユニット中のバックエンドサーバを切り替えるかどうかは，各バックエンドサーバの系切り替えの発生順序で決まります。切り替えの発生順序は，クラスタソフトウェアの動作によって決まります。

この例では，BES3 だけが移動する場合を記載していますが，クラスタソフトウェアの動作によっては，BES1，又は BES4 が移動することもあります。

●受け入れ数が不足した状態で障害が発生した場合の対処方法の例

受け入れ数が不足した状態で障害が発生した場合の対処方法の例を次の図に示します。

図 26–13　受け入れ数が不足した状態で障害が発生した場合の対処方法の例



〔説明〕

hostB の障害を回復し，unt2 を起動します。その結果，unt2 で BES4 が実行系として起動し，BES1 が受け入れ可能状態になります。

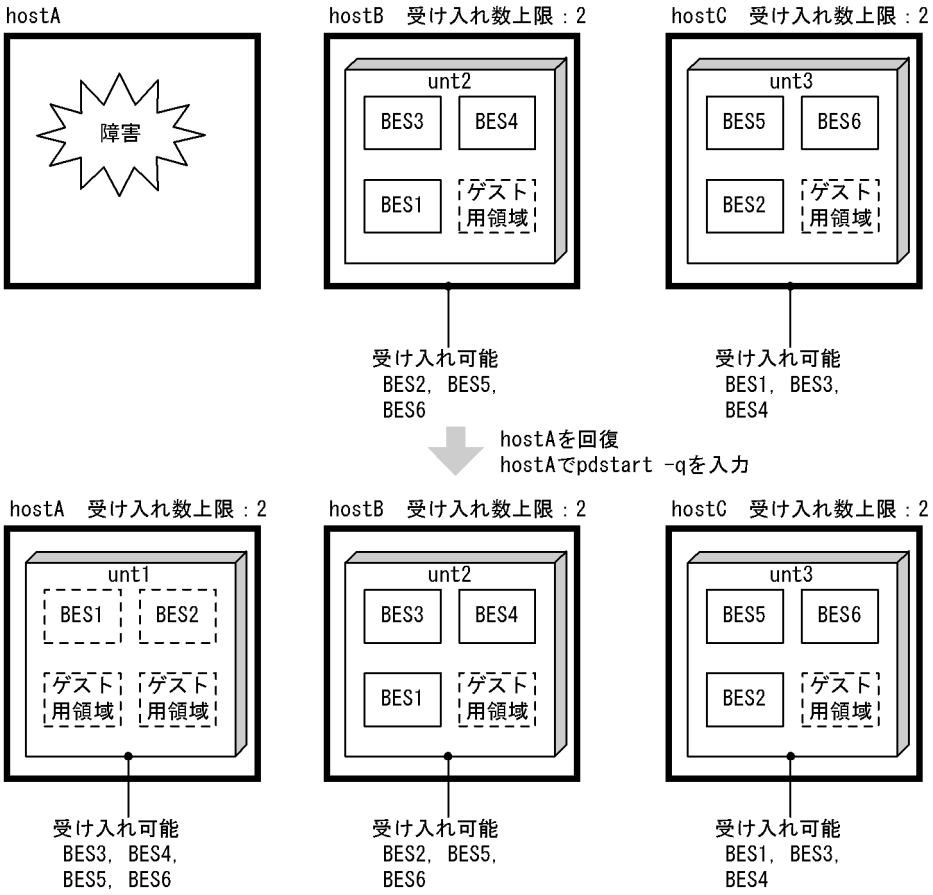
次に，unt2 で BES1 に対して monact コマンドを入力すると，ゲスト BES として BES1 の処理が開始されます（Hitachi HA Toolkit Extension 使用時は，クラスタソフトウェアのオンラインコマンドを使用します）。

●受け入れ数の不足を回避する方法の例

受け入れ数を大きく設定できない場合は，受け入れ数不足によるサーバ停止を回避するためにできるだけ早く障害を回復する必要があります。

受け入れ数の不足を回避する方法の例を次の図に示します。

図 26–14　受け入れ数の不足を回避する方法の例



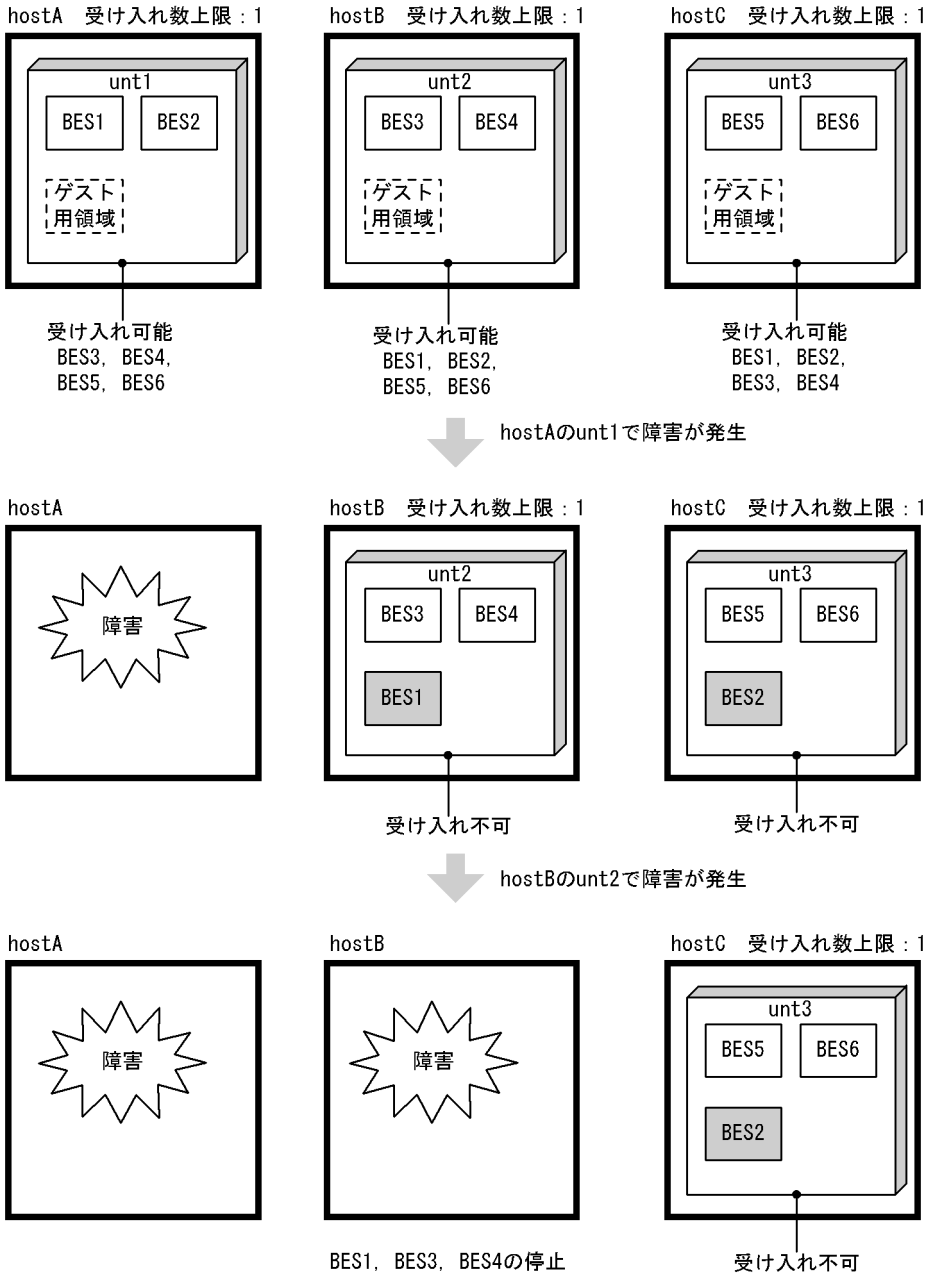
〔説明〕

hostA の障害を回復し，unt1 を起動します。その結果，unt1 で BES1 と BES2 が待機系として起動し，BES3，BES4，BES5，BES6 が受け入れ可能状態になります。したがって，hostB で障害が発生しても，BES1 と BES4 は hostA で処理を継続し，BES3 は hostC で処理を継続できます。

●多重障害時の系切り替え（受け入れできなくなる場合）の例

多重障害時に系切り替えができなくなる場合の例を次の図に示します。

図 26–15　多重障害時に系切り替えができなくなる場合の例



〔説明〕

hostA で障害が発生すると，BES1 は unt2 に移動してゲスト BES として処理を実行します。このとき，unt2 及び unt3 の受け入れ数上限は 1 のため，両ユニットともこれ以上サーバを受け入れられません。

hostA に続いて hostB で障害が発生すると，BES1，BES3 及び BES4 は停止します。

多重障害発生時も全バックエンドサーバの処理を続行させる必要がある場合には，受け入れ数の上限を大きく設定しておく必要があります。受け入れできなくなる場合の対処方法，及びサーバ停止の回避方法は，受け入れ数の不足の場合と同じです。

26.1.4　系切り替え機能の適用基準

(1)　適用基準

系切り替え機能の適用基準を次の表に示します。

表 26–9　系切り替え機能の適用基準

観点	スタンバイ型系切り替え	1：1 スタンバイレス型系切り替え	影響分散スタンバイレス型系切り替え
トランザクション性能重視	○	×	△
	• 障害中も処理性能を維持 • 通常時と同等のスループットを保証	• 障害中の代行処理の負荷が処理性能に影響を及ぼす可能性あり • 特定ユニットへの負荷集中によってシステムのスループットが低下	• 障害中の代行処理の負荷が処理性能に影響を及ぼす可能性あり • 負荷分散によってシステムのスループット低下を軽減
リソース所要量重視	×	○	○
	待機用サーバマシン，又は待機ユニット用のシステムリソースが必要	待機用サーバマシン，又は待機ユニット用のシステムリソースは不要	待機用サーバマシン，又は待機ユニット用のシステムリソースは不要
多重障害対応	○	×	○
	クラスタソフトウェアとの連携によって対応可能（IP アドレス引き継ぎありの場合だけ対応可能。高速系切り替え機能適用時は対応できない)	代行先サーバが固定されているため，対応できない	クラスタソフトウェアとの連携によって対応可能

（凡例）

○：優位となります。

△：やや優位となります。

×：劣位となります。

(2)　系切り替え機能を適用している場合の別の系切り替え機能の適用可否

系切り替え機能を適用している場合の，同一システム内の他ユニットへの別の系切り替え機能の適用可否を次の表に示します。

表 26–10　同一システム内の他ユニットへの別の系切り替え機能の適用可否

適用済み機能	スタンバイ型系切り替え機能の適用	1:1 スタンバイレス型系切り替え機能の適用	影響分散スタンバイレス型系切り替え機能の適用
スタンバイ型系切り替え機能	○	○	○

適用済み機能	スタンバイ型系切り替え機能の適用	1:1 スタンバイレス型系切り替え機能の適用	影響分散スタンバイレス型系切り替え機能の適用
1:1 スタンバイレス型系切り替え機能	○	○	×
影響分散スタンバイレス型系切り替え機能	○	×	○※

（凡例）

○：適用できます。

×：適用できません。

注※　切り替え先の HA グループは一つだけになります。

26.1.5　HiRDB がサポートしているクラスタソフトウェア

HiRDB がサポートしているクラスタソフトウェア及びバージョンを次の表に示します。

表 26–11　HiRDB がサポートしているクラスタソフトウェア

<table>
<tr><th colspan="2" rowspan="2">プラットフォーム</th><th rowspan="2">モニタモード</th><th colspan="3">スタンバイ型系切り替え機能</th><th rowspan="2">1:1 スタンバイレス型系切り替え</th><th rowspan="2">影響分散スタンバイレス型系切り替え</th></tr>
<tr><th>ユーザサーバホットスタンバイ</th><th>高速系切り替え</th><th>それ以外</th></tr>
<tr><td rowspan="4">Windows</td><td>Windows Server 2012</td><td rowspan="2">• WSFC</td><td colspan="3" rowspan="2">• WSFC
• HA ToolEx 01-20-01 以降</td><td colspan="2" rowspan="2">• WSFC
• HA ToolEx 01-20-01 以降
• HiRDB/AdHA 10-00 以降</td></tr>
<tr><td>Windows Server 2016</td></tr>
<tr><td rowspan="2">Windows Server 2019
Windows Server 2022</td><td>• WSFC</td><td colspan="3">• WSFC
• HA ToolEx 01-20-01 以降</td><td colspan="2">• WSFC
• HA ToolEx 01-20-01 以降
• HiRDB/AdHA 10-00 以降</td></tr>
<tr><td>• WSFC
• DataKeeper V8.8.2 以降</td><td colspan="3">• WSFC
• HA ToolEx 01-20-01 以降
• DataKeeper V8.8.2 以降</td><td>–</td><td>• WSFC
• HA ToolEx 01-20-01 以降
• HiRDB/AdHA 10-00 以降
• DataKeeper V8.8.2 以降</td></tr>
</table>

プラットフォーム		モニタモード	スタンバイ型系切り替え機能			1:1 スタンバイレス型系切り替え	影響分散スタンバイレス型系切り替え
			ユーザサーバホットスタンバイ	高速系切り替え	それ以外		
		• CLUSTERPRO X 4.3 以降	• CLUSTERPRO X 4.3 以降 • HA ToolEx 01-21 以降			−	• CLUSTERPRO X 4.3 以降 • HA ToolEx 01-21 以降 • HiRDB/AdHA 10-00 以降
Windows（Amazon EC2）	Windows Server 2019 Windows Server 2022	• WSFC • DataKeeper V8.8.2 以降	• WSFC • HA ToolEx 01-20-01 以降 • DataKeeper V8.8.2 以降			−	• WSFC • HA ToolEx 01-20-01 以降 • HiRDB/AdHA 10-00 以降 • DataKeeper V8.8.2 以降
		• CLUSTERPRO X 4.3 以降	• CLUSTERPRO X 4.3 以降 • HA ToolEx 01-21 以降			−	• CLUSTERPRO X 4.3 以降 • HA ToolEx 01-21 以降 • HiRDB/AdHA 10-00 以降
Windows（Azure VM）	Windows Server 2019 Windows Server 2022	• WSFC • DataKeeper V8.8.2 以降	• WSFC • HA ToolEx 01-20-01 以降 • DataKeeper V8.8.2 以降			−	• WSFC • HA ToolEx 01-20-01 以降 • HiRDB/AdHA 10-00 以降 • DataKeeper V8.8.2 以降
		• CLUSTERPRO X 4.3 以降	• CLUSTERPRO X 4.3 以降 • HA ToolEx 01-21 以降			−	• CLUSTERPRO X 4.3 以降 • HA ToolEx 01-21 以降

プラットフォーム		モニタモード	スタンバイ型系切り替え機能			1:1 スタンバイレス型系切り替え	影響分散スタンバイレス型系切り替え
			ユーザサーバホットスタンバイ	高速系切り替え	それ以外		
							• HiRDB/AdHA 10-00 以降
Windows（Oracle Cloud Infrastructure）	Windows Server 2019 Windows Server 2022	• CLUSTERPRO X 5.1 以降	• CLUSTERPRO X 5.1 以降 • HA ToolEx 01-21 以降			−	• CLUSTERPRO X 5.1 以降 • HA ToolEx 01-21 以降 • HiRDB/AdHA 10-00 以降

（凡例）

WSFC：Windows Server Failover Cluster

HA ToolEx：Hitachi HA Toolkit Extension

HiRDB/AdHA：HiRDB Advanced High Availability

DataKeeper：SIOS DataKeeper Cluster Edition

−：未サポート

26.2　システム構成例

ここでは，系切り替え機能使用時のシステム構成例について説明します。

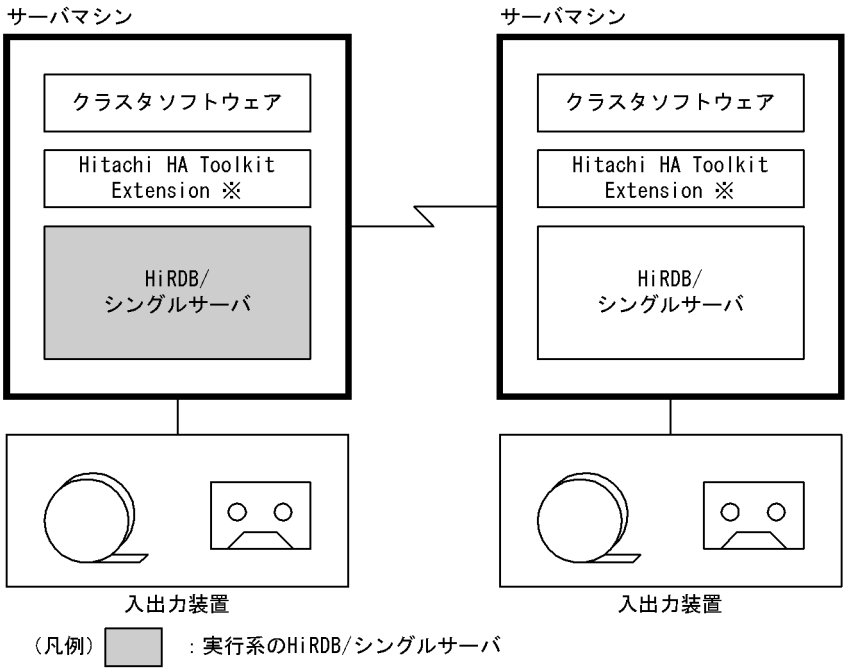
26.2.1　HiRDB/シングルサーバのシステム構成例（スタンバイ型系切り替え）

HiRDB/シングルサーバの場合はシステム単位で系を切り替えます。HiRDB/シングルサーバのシステム構成例を次に示します。

(1)　1：1 系切り替え構成の例

実行系と待機系が 1：1 で対応しているシステム構成を**1：1 系切り替え構成**といいます。系が切り替わってもレスポンスタイムを保証したい場合にこの構成を適用します。ただし，待機系のサーバマシンのリソースは使用できません（どちらか片方のサーバマシンのリソースが使用できません）。HiRDB/シングルサーバに対する 1：1 系切り替え構成の例を次の図に示します。

図 26–16　HiRDB/シングルサーバに対する 1：1 系切り替え構成の例



注※　サーバモードで系切り替え機能を運用する場合に必要な製品です。

備考

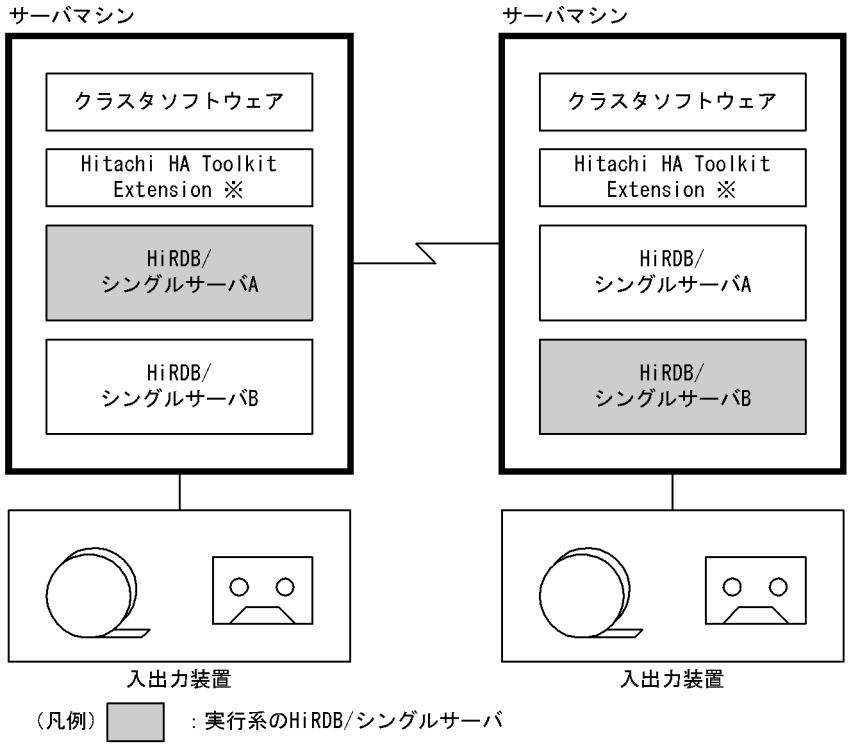
- 各サーバマシンにユティリティで使用する入出力装置が必要です。
- 系が切り替わった後にユティリティを実行する場合，ユティリティの実行に必要な入出力ファイルを自サーバマシンに作成する必要があります。

(2)　相互系切り替え構成の例

一つのサーバマシン内に実行系と待機系（ほかの HiRDB/シングルサーバの待機系）を持つ構成を**相互系切り替え構成**といいます。複数の HiRDB/シングルサーバを使用している場合に相互系切り替え構成を適用できます。サーバマシンのリソースを有効に利用したい場合にこの構成を適用してください。ただし，系が切り替わると，レスポンスタイムが低下します。

HiRDB/シングルサーバに対する相互系切り替え構成の例を次の図に示します。次の図の例は，二つの HiRDB/シングルサーバ（HiRDB/シングルサーバ A と HiRDB/シングルサーバ B）に対して相互系切り替え構成を適用しています。

図 26–17　HiRDB/シングルサーバに対する相互系切り替え構成の例



注※　サーバモードで系切り替え機能を運用する場合に必要な製品です。

備考

- 各サーバマシンにユティリティで使用する入出力装置が必要です。
- 系が切り替わった後にユティリティを実行する場合，ユティリティの実行に必要な入出力ファイルを自サーバマシンに作成する必要があります。
- IP アドレスを引き継がない系切り替え構成で，ユティリティの制御文にホスト名を指定している場合，系が切り替わったときに，ホスト名を変更する必要があります。

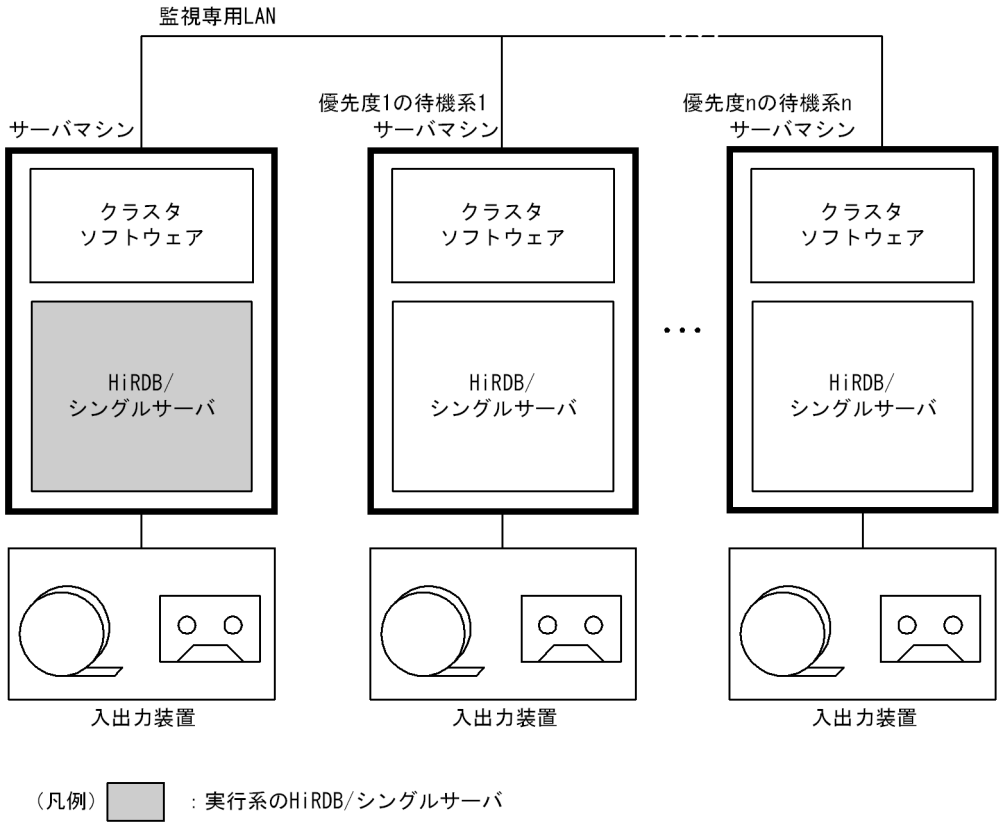
(3)　マルチスタンバイ構成の例

一つの実行系に複数の待機系を持つシステム構成を，**マルチスタンバイ構成**といいます。実行系の障害が回復するまでの間に発生する，待機系の障害（多点障害）に備えたい場合にこの構成を適用します。待機

系には優先度を付け，実行系に障害が発生したときは，一番優先度の高い待機系に切り替えます。なお，マルチスタンバイ構成は，IP アドレスを引き継ぐシステム構成の場合だけ適用できます。

HiRDB/シングルサーバに対するマルチスタンバイ構成の例を次の図に示します。

図 26–18　HiRDB/シングルサーバに対するマルチスタンバイ構成の例



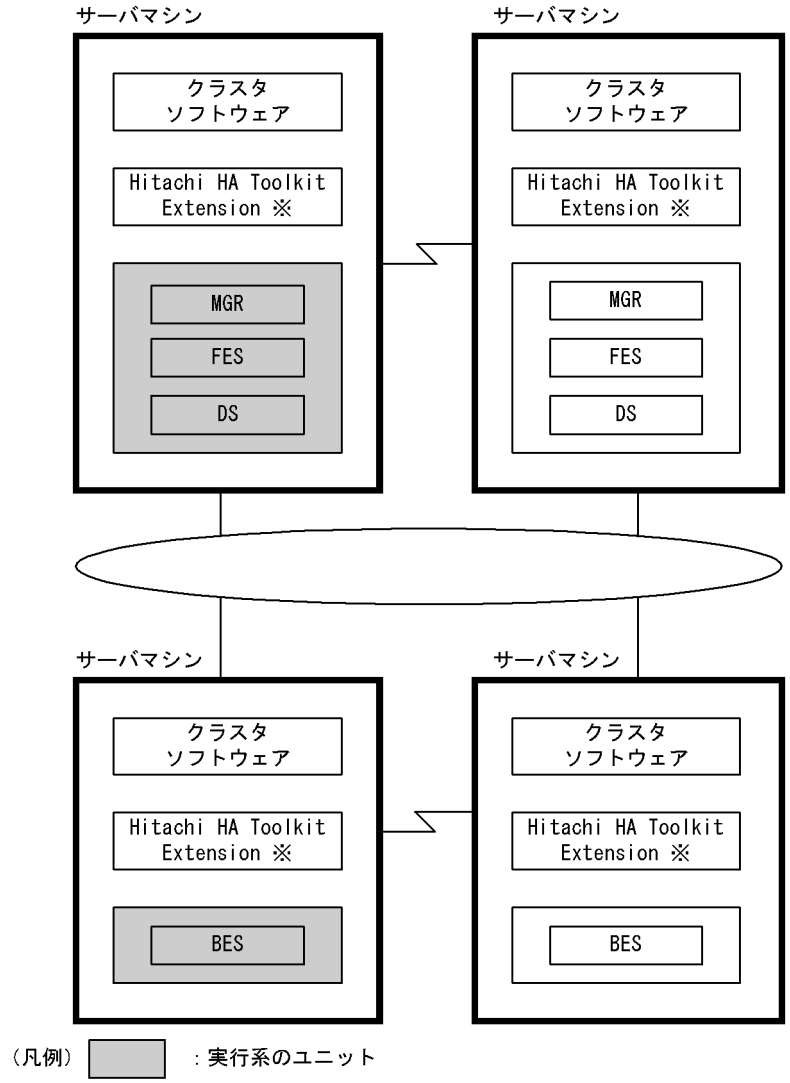
26.2.2　HiRDB/パラレルサーバのシステム構成例（スタンバイ型系切り替え）

HiRDB/パラレルサーバの場合はユニット単位で系を切り替えます。HiRDB/パラレルサーバのシステム構成例を次に示します。

(1)　1：1 系切り替え構成の例

実行系ユニットと待機系ユニットが 1：1 で対応しているシステム構成を **1：1 系切り替え構成**といいます。系が切り替わってもレスポンスタイムを保証したい場合にこの構成を適用します。ただし，待機系のサーバマシンのリソースは使用できません（どちらか片方のサーバマシンのリソースが使用できません）。HiRDB/パラレルサーバに対する 1：1 系切り替え構成の例を次の図に示します。

図 26–19　HiRDB/パラレルサーバに対する 1：1 系切り替え構成の例



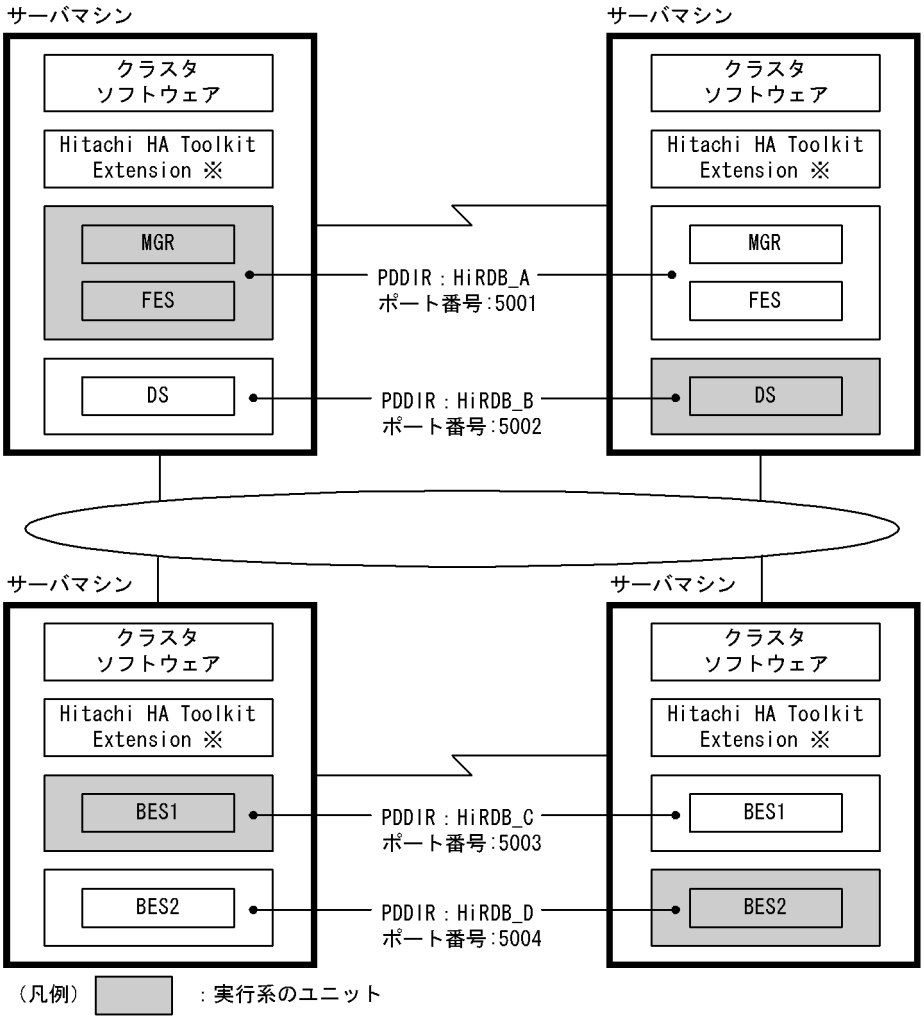
注※　サーバモードで系切り替え機能を運用する場合に必要な製品です。

(2)　相互系切り替え構成の例

一つのサーバマシン内に実行系と待機系（ほかのユニットの待機系）を持つ構成を**相互系切り替え構成**といいます。HiRDB/パラレルサーバの場合はユニット単位で系を切り替えるため，HiRDB/シングルサーバと異なり，一つの HiRDB/パラレルサーバに対して相互系切り替え構成を適用できます。HiRDB/パラレルサーバに相互系切り替え構成を適用すると，一つのサーバマシン内に実行系ユニットと待機系ユニット（ほかのユニットの待機系ユニット）を配置できます。サーバマシンのリソースを有効に利用したい場合にこの構成を適用してください。ただし，系が切り替わると，レスポンスタイムが低下します。

HiRDB/パラレルサーバに対する相互系切り替え構成の例を次の図に示します。

図 26–20　HiRDB/パラレルサーバに対する相互系切り替え構成の例



注※　サーバモードで系切り替え機能を運用する場合に必要な製品です。

相互系切り替え構成の HiRDB システム定義の定義例については，マニュアル「HiRDB システム定義」を参照してください。

ポイント

相互系切り替え構成（又は 2：1 系切り替え構成）をすると，一つのサーバマシン内で二つのユニットが稼働することがあります。このため，次に示す点に留意してください。

- HiRDB 運用ディレクトリ名
- ポート番号

これらを同じサーバマシン内では，ユニットごとに異なる名称及び異なる番号にしてください。

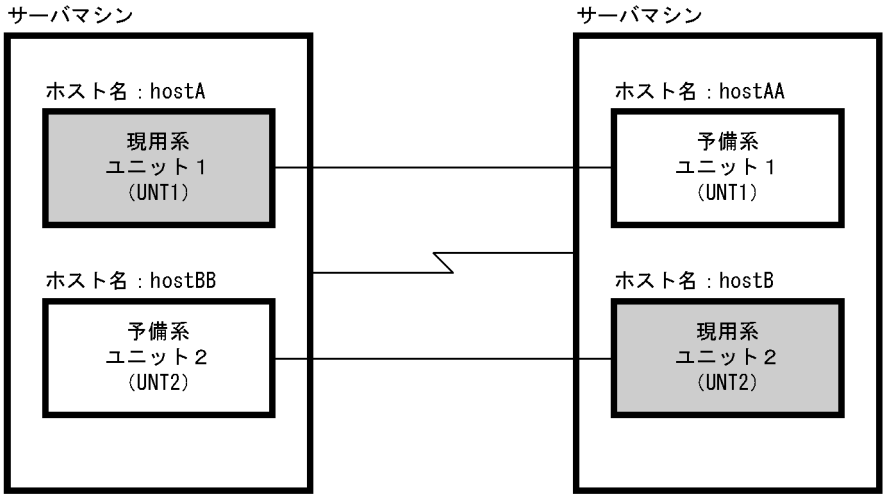
また，IP アドレスを引き継がない場合は，更に次に示す点に留意してください。

- ホスト名及び IP アドレス

同じサーバマシン内ではホスト名及び IP アドレスをユニットごとに異なるものにしてください。

ホスト名の正しい設定例を次の図に，間違った設定例を図「ホスト名の設定例（間違った例）」に示します。

図 26–21　ホスト名の設定例（正しい例）

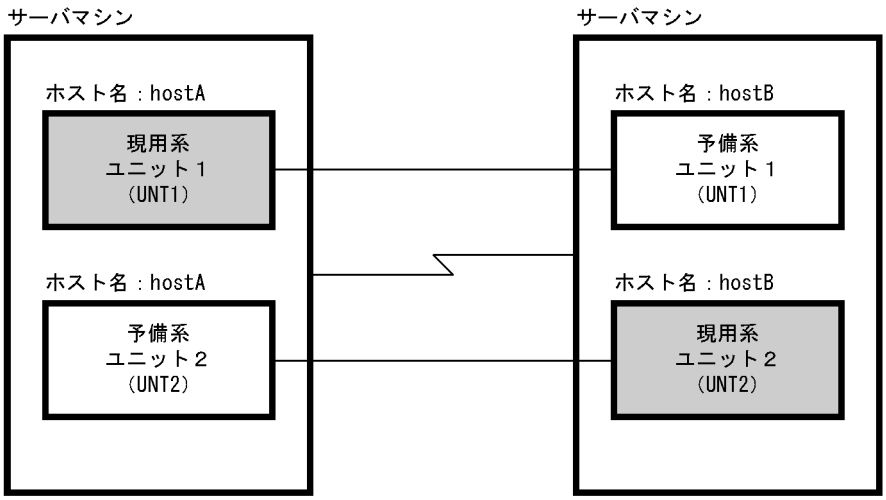


このときの pdunit オペランドの指定例（正しい例）を次に示します。

```
pdunit -x hostA -u UNT1  ...  -c hostAA
pdunit -x hostB -u UNT2  ...  -c hostBB
```

さらに，hostA，hostAA，hostB，hostBB のすべてで異なる IP アドレスに対するホスト名を設定する必要があります。

図 26–22　ホスト名の設定例（間違った例）



このときの pdunit オペランドの指定例（間違った例）を次に示します。

```
pdunit -x hostA -u UNT1  ...  -c hostB
pdunit -x hostB -u UNT2  ...  -c hostA
```

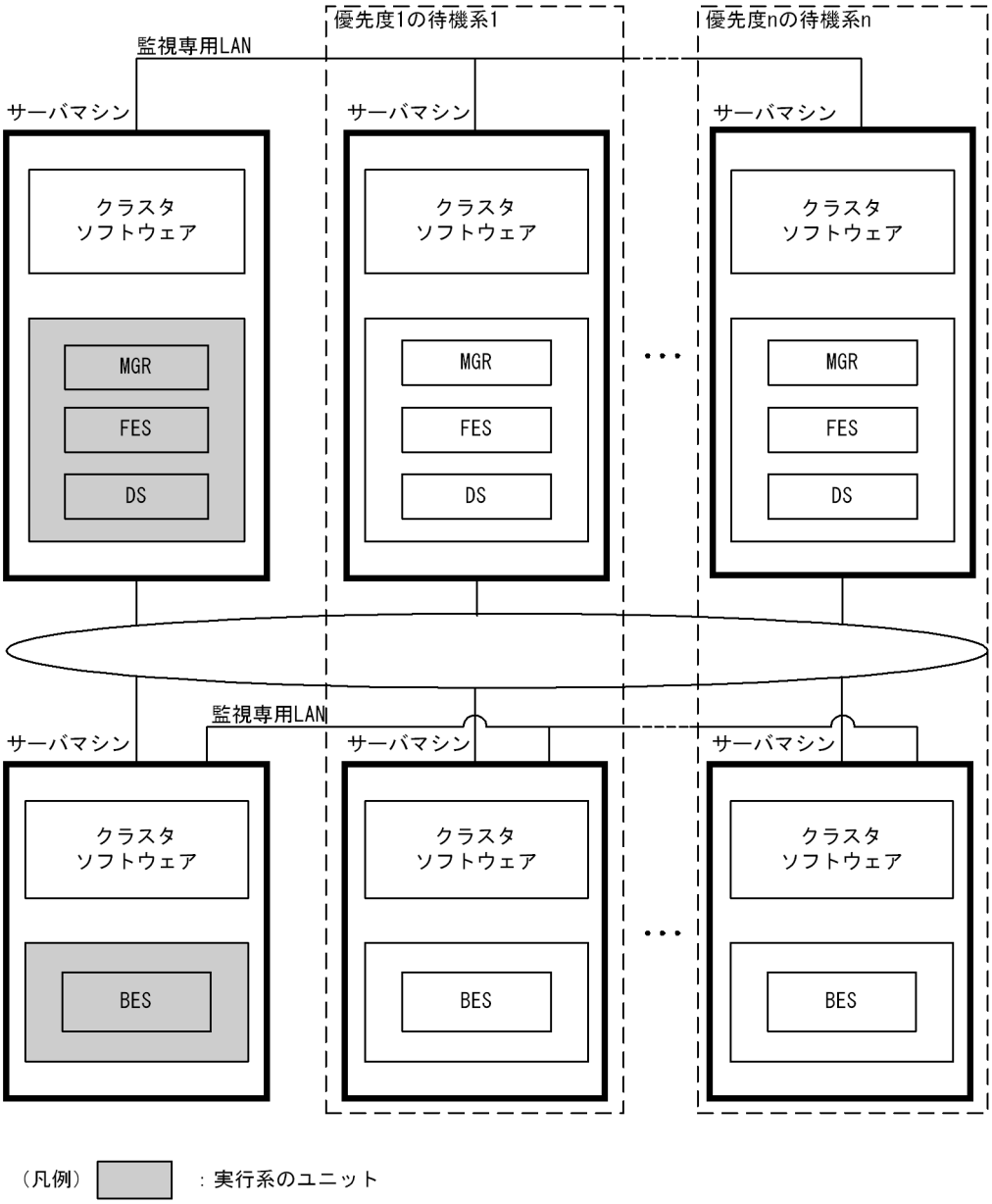
〔説明〕

相互系切り替え構成で IP アドレスを引き継がない場合は，pdunit オペランドの-x 及び-c オプションに指定するホスト名はすべて異なる必要があります（同じホスト名を指定できません）。

(3)　マルチスタンバイ構成の例

一つの実行系ユニットに複数の待機系ユニットを持つシステム構成を，**マルチスタンバイ構成**といいます。実行系の障害が回復するまでの間に発生する，待機系の障害（多点障害）に備えたい場合にこの構成を適用します。待機系には優先度を付け，実行系に障害が発生したときは，一番優先度の高い待機系に切り替えます。なお，マルチスタンバイ構成は，IP アドレスを引き継ぐシステム構成の場合だけ適用できます。HiRDB/パラレルサーバに対するマルチスタンバイ構成の例を次の図に示します。

図 26–23　HiRDB/パラレルサーバに対するマルチスタンバイ構成の例



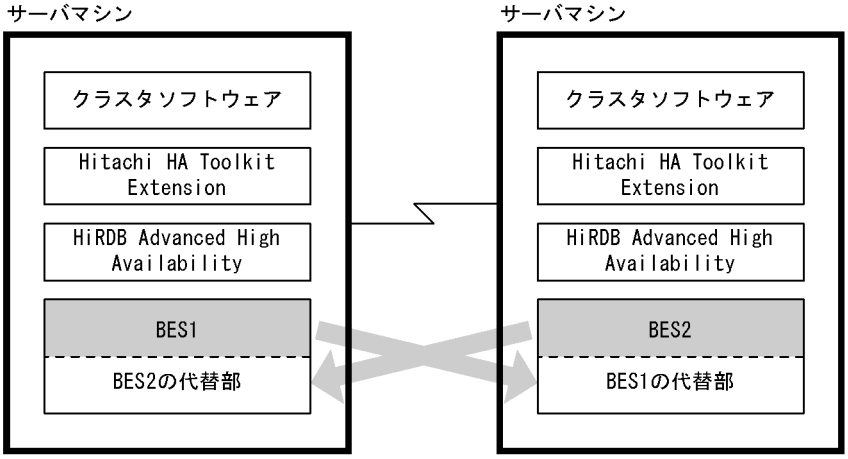
26.2.3　1：1 スタンバイレス型系切り替えのシステム構成例

1：1 スタンバイレス型系切り替えの代表的なシステム構成例を説明します。

(1)　相互代替構成

1：1 スタンバイレス型系切り替えで二つのバックエンドサーバがお互いに代替 BES になる構成です。相互代替構成のシステム構成例を次の図に示します。

図 26–24　相互代替構成のシステム構成例



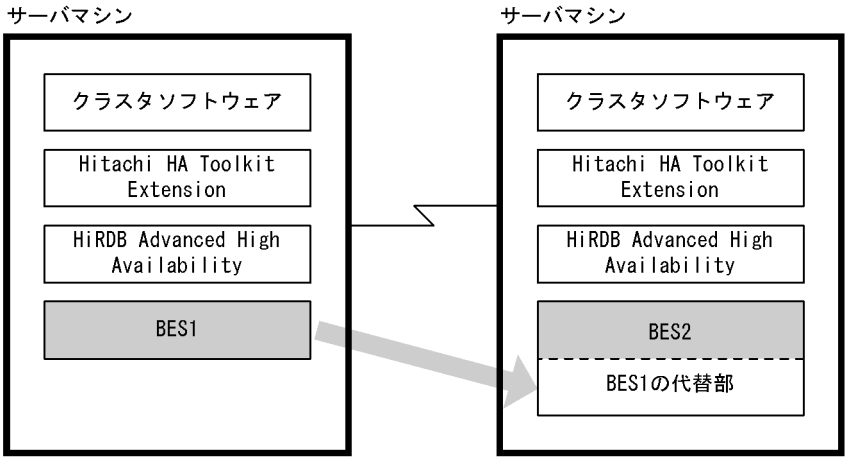
〔説明〕

- BES1 は BES2 の代替 BES です。BES2 に障害が発生した場合は，BES2 の代替部が BES2 の処理を代行します。
- BES2 は BES1 の代替 BES です。BES1 に障害が発生した場合は，BES1 の代替部が BES1 の処理を代行します。

(2)　片方向代替構成

1：1 スタンバイレス型系切り替えで片方向だけの代行関係を持つ構成です。片方向代替構成（2 ノード構成）のシステム構成例を次の図に示します。

図 26–25　片方向代替構成（2 ノード構成）のシステム構成例



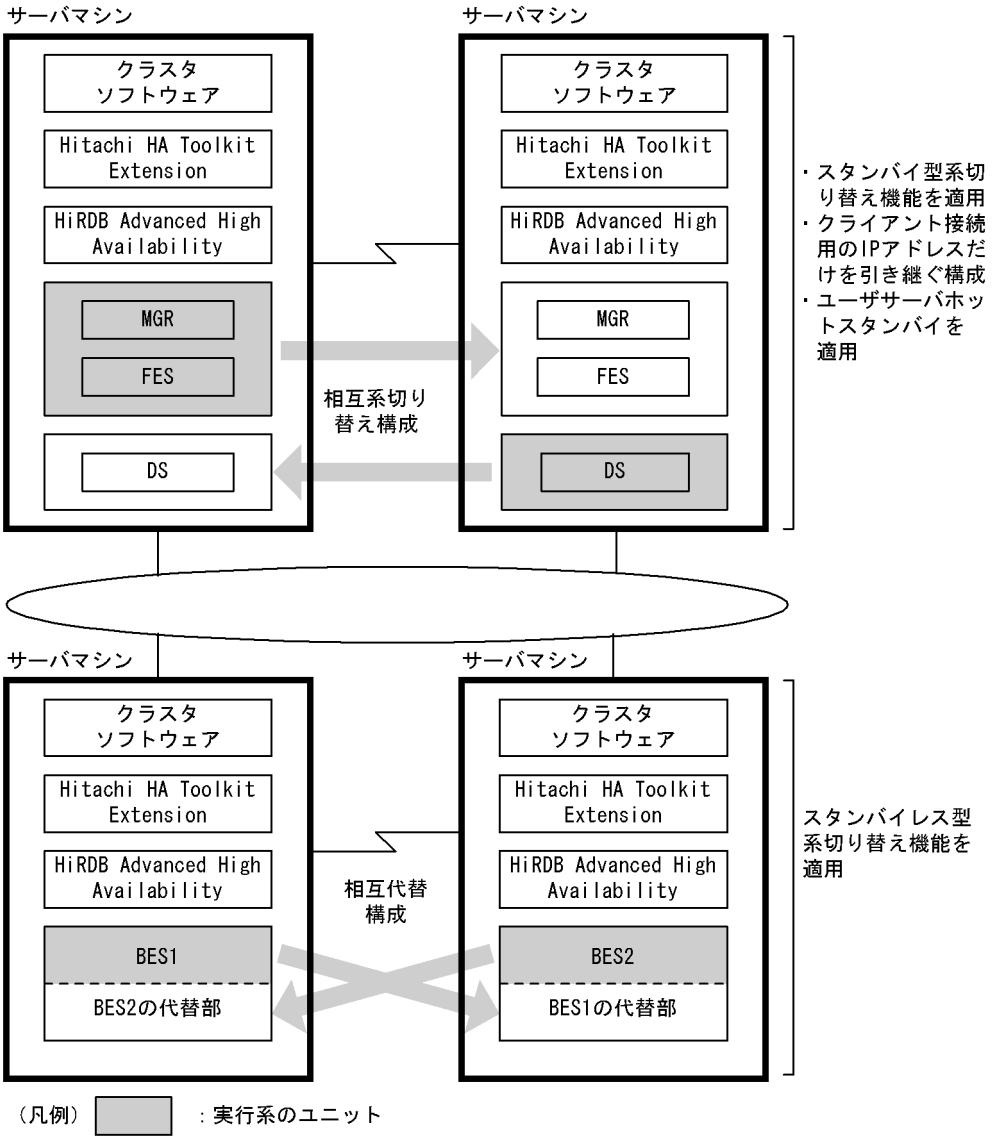
〔説明〕

BES2 は BES1 の代替 BES です。BES1 に障害が発生した場合は，BES1 の代替部が BES1 の処理を代行します。BES2 に障害が発生した場合，BES1 は処理を代行しません。

(3)　1：1 スタンバイレス型とスタンバイ型を混合したシステム構成例

1：1 スタンバイレス型とスタンバイ型を混合したシステム構成例を次の図に示します。このシステム構成の HiRDB システム定義の例については，マニュアル「HiRDB システム定義」を参照してください。

図 26–26　1：1 スタンバイレス型とスタンバイ型を混合したシステム構成例



〔説明〕

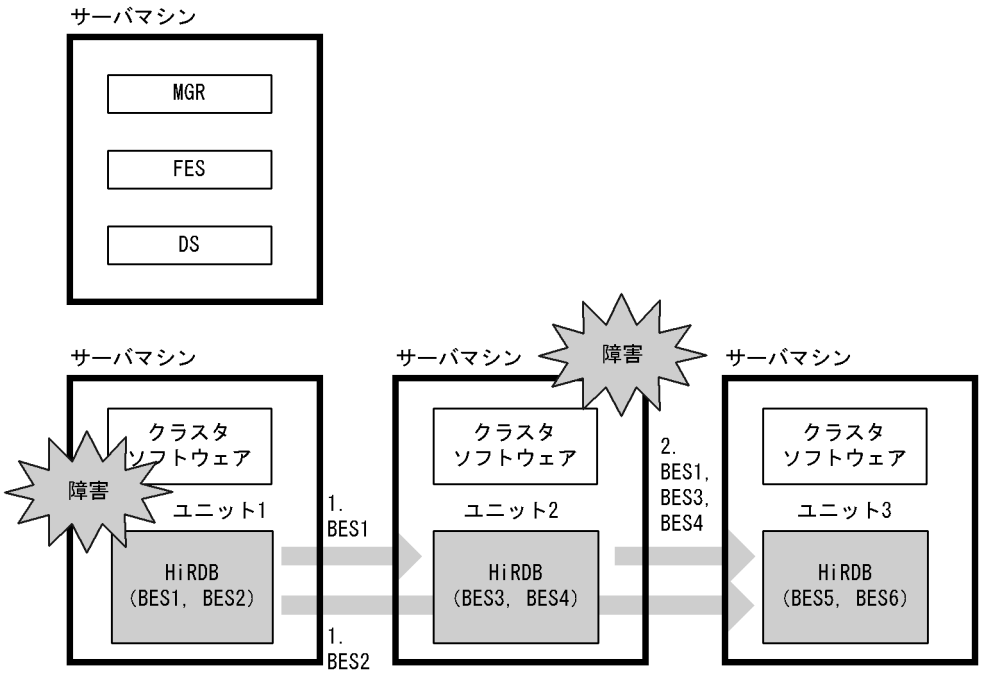
- MGR，FES，DS のユニットには，スタンバイ型系切り替え機能（相互系切り替え構成）を適用し，クライアント接続用の IP アドレスだけを引き継ぐ構成にします。また，ユーザサーバホットスタンバイを適用します。
- BES のユニットには，1：1 スタンバイレス型系切り替え機能（相互代替構成）を適用します。
- 全サーバマシンに HiRDB Advanced High Availability が必要になります。スタンバイレス型系切り替え機能を適用しないサーバマシン，及び系切り替え機能を適用しないサーバマシンにも HiRDB Advanced High Availability が必要です。

26.2.4　影響分散スタンバイレス型系切り替えのシステム構成例

(1)　システム構成例

影響分散スタンバイレス型系切り替えのシステム構成例を次の図に示します。正規ユニットでの障害発生時に，障害対象の現用系のバックエンドサーバへの処理を稼働中の複数のサーバマシンがバックエンドサーバ単位で分担して実行します。

図 26–27　影響分散スタンバイレス型系切り替えのシステム構成例



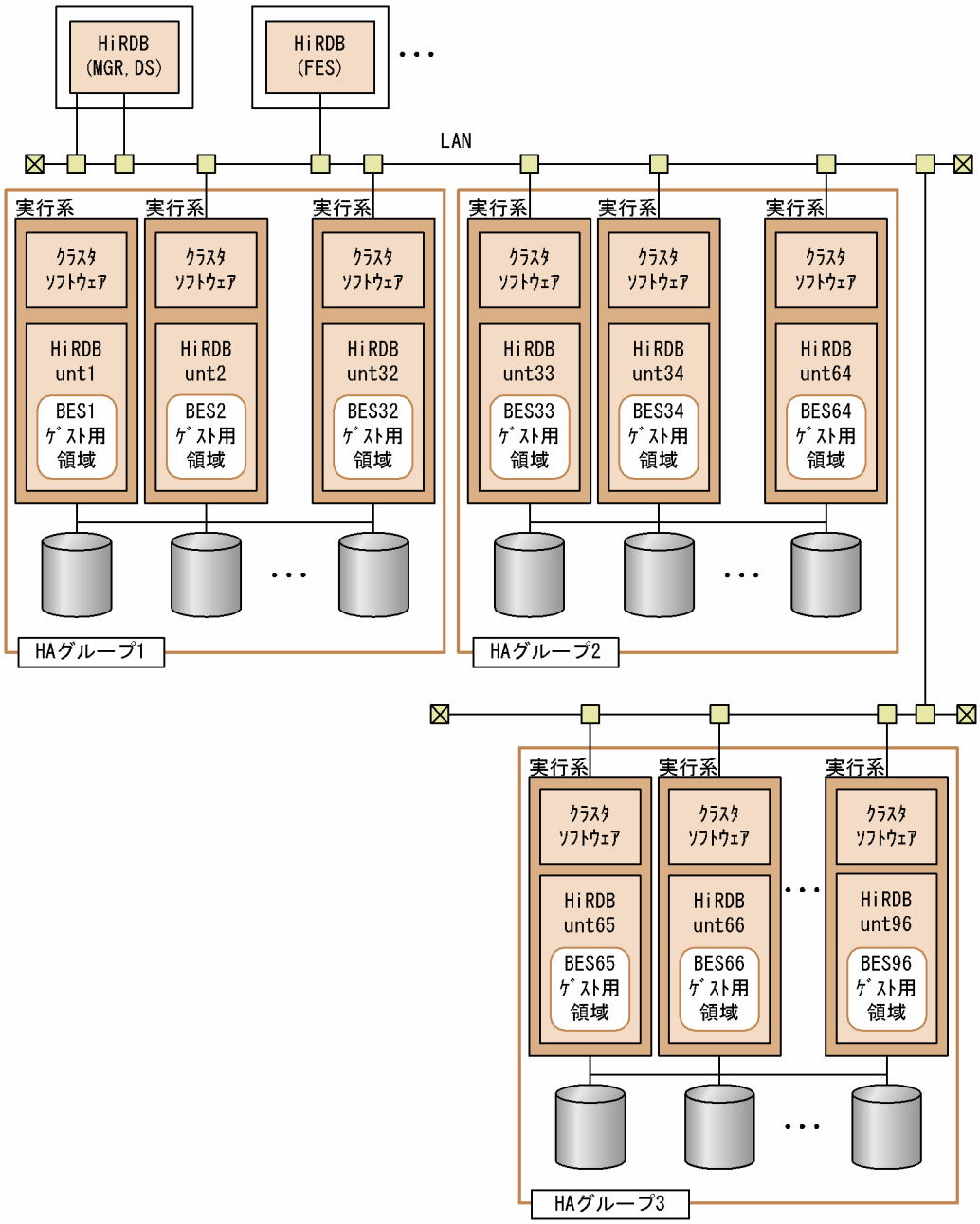
〔説明〕

1. ユニット 1 に障害が発生した場合は，ユニット 2 で BES1 がゲスト BES として処理を実行し，ユニット 3 で BES2 がゲスト BES として処理を実行します。
2. ユニット 1 の障害中に更にユニット 2 で障害が発生した場合は，ユニット 3 で BES1，BES2，BES3，BES4 がそれぞれゲスト BES として処理を実行します。

(2)　HA グループを複数定義している構成

一つの HA グループに定義できるユニットの数は 32 ですが，HA グループを複数定義することで，32 ユニットを超える大規模なシステムでも，影響分散スタンバイレス型系切り替え機能を使用できます。HA グループを複数定義している場合の影響分散スタンバイレス型系切り替えのシステム構成例を次の図に示します。

図 26–28　HA グループを複数定義している場合の影響分散スタンバイレス型系切り替えのシステム構成例



〔説明〕

HA グループ 1 には，unt1～unt32 までの 32 ユニットが定義されています。unt1 のサーバマシンで障害が発生した場合，BES1 は HA グループ 1 中のユニットのゲスト BES として処理を実行します。HA グループ 2 及び 3 でも同様です。

三つの HA グループを定義することで，96 ユニットのシステムで影響分散スタンバイレス型系切り替え機能を使用できます。

(3)　各サーバの切り替え先の決定方法

影響分散スタンバイレス型系切り替え機能の場合，次に示す点を考慮して各サーバの切り替え先を決定します。

- 実行中ゲスト BES の数をユニット間で均等にする
- グローバルバッファを共用するサーバの数をユニット間で均等にする

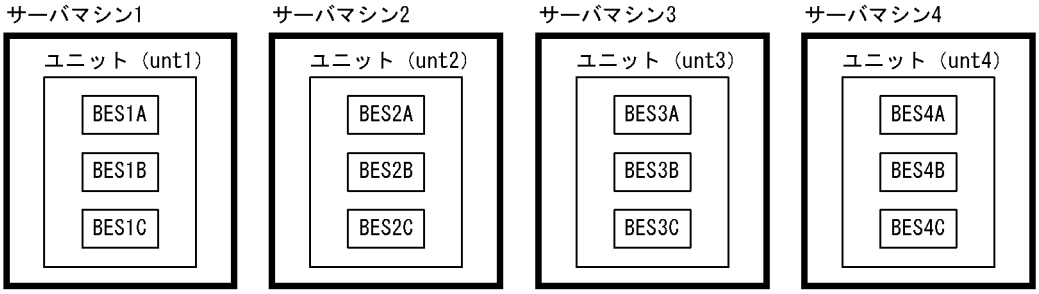
影響分散スタンバイレス型系切り替えの場合，HiRDB は次に示す手順で切り替え先を決定します。

〈手順〉

1. 全サーバの優先順位第 1 位の切り替え先ユニットを決定します。決定基準は次のとおりです。
   ・一つのユニットから各ユニットに切り替えるサーバの数を均等にします。
   ・切り替え先でグローバルバッファを共用するサーバの数をユニット間で均等にします。
2. i ＝ 1 から HA グループ内のユニット数−2 まで，手順の 3〜9 を繰り返します。
3. 優先順位 i ＋ 1 番目の切り替え先が決まっていないサーバを一つ選択します。
4. 該当するサーバがある場合は手順の 5 に進み，該当するサーバがない場合は i ＝ i ＋ 1 として手順の 3 に戻ります。
5. 選択したサーバの 1 番目から i 番目までの切り替え先と該当サーバが属するユニットを障害ユニットと想定します。
6. 障害ユニット以外のユニットを切り替え先候補ユニットとします。
7. 障害ユニットに属するサーバから，定義済み切り替え先に切り替え先候補ユニットを含むサーバをすべて抽出し，最も優先順位が高い切り替え先ユニットに該当サーバのゲスト BES が割り当てられていることにします。
8. 障害ユニットに属するサーバから，定義済み切り替え先に切り替え先候補となるユニットを含まないサーバをすべて抽出し，それぞれ i ＋ 1 番目の切り替え先ユニットを決定します。決定基準は次のとおりです。
   ・ゲスト BES 数が最も少ないユニットを選択します。
   ・ゲスト BES 中に該当サーバとグローバルバッファを共用するサーバが少ないユニットを選択します。
9. 手順の 3 に戻ります。

(a)　4 ユニット構成の場合

影響分散スタンバイレス型系切り替え機能を次に示すような 4 ユニット構成で行います。



この場合，次に示すグループごとにグローバルバッファを共用します。

- A グループ：BES1A，BES2A，BES3A，BES4A
- B グループ：BES1B，BES2B，BES3B，BES4B
- C グループ：BES1C，BES2C，BES3C，BES4C

1. **優先順位第 1 位の切り替え先ユニットの決定**

   BES3B の優先順位第 1 位の切り替え先を決定する例を示します。グローバルバッファを共用するサーバ（BES1B，BES2B）及び BES3A と切り替え先が同じにならないように切り替え先を決定します。具体的には，BES1B の切り替え先（unt3），BES2B の切り替え先（unt4），BES3A の切り替え先（unt4）以外となるため，unt1 を選択します。

サーバ	ホスト BES	優先順位第 1 位	優先順位第 2 位	優先順位第 3 位
BES1A	unt1	unt2	割り当て中	割り当て中
BES1B		unt3	割り当て中	割り当て中
BES1C		unt4	割り当て中	割り当て中
BES2A	unt2	unt3	割り当て中	割り当て中
BES2B		unt4	割り当て中	割り当て中
BES2C		unt1	割り当て中	割り当て中
BES3A	unt3	unt4	割り当て中	割り当て中
BES3B		**unt1**	割り当て中	割り当て中
BES3C		割り当て中	割り当て中	割り当て中
BES4A	unt4	割り当て中	割り当て中	割り当て中
BES4B		割り当て中	割り当て中	割り当て中
BES4C		割り当て中	割り当て中	割り当て中

2. **優先順位 2 番目以降の切り替え先ユニットの決定**

   BES2B の優先順位第 2 位の切り替え先を決定する例を示します。

   まず，BES2B の定義済み切り替え先 unt4 と BES2C の属する unt2 を障害ユニットと想定し（手順 5），unt1 と unt3 を切り替え先候補ユニットとします（手順 6）。次に，障害ユニット（unt2，unt4）に属

するサーバから，切り替え先候補ユニットを切り替え先に定義済みのサーバ（BES2A，BES2C，BES4A，BES4C）を抽出し，次のとおりゲストサーバが割り当てられていることにします（手順 7）。

BES2A：unt3，BES2C：unt1，BES4A：unt1，BES4C：unt3

次に，定義済み切り替え先に切り替え先候補ユニットを含まないサーバ（BES2B，BES4B）を抽出し，優先順位 2 番目の切り替え先ユニットを決定します。このとき，unt1，unt3 のゲスト BES 数は等しく，BES2B と BES2B がグローバルバッファを共用するため，BES2B の切り替え先を unt1，BES4B の切り替え先を unt3 とします。

サーバ	ホスト BES	優先順位第 1 位	優先順位第 2 位	優先順位第 3 位
BES1A	unt1 切り替え先候補ユニット	unt2	unt4	割り当て中
BES1B		unt3	unt2	割り当て中
BES1C		unt4	unt2	割り当て中
BES2A	**unt2** **障害ユニット**	unt3	unt1	割り当て中
BES2B		unt4	**unt1**	割り当て中
BES2C		unt1	割り当て中	割り当て中
BES3A	unt3 切り替え先候補ユニット	unt4	割り当て中	割り当て中
BES3B		unt1	割り当て中	割り当て中
BES3C		unt2	割り当て中	割り当て中
BES4A	**unt4** **障害ユニット**	unt1	割り当て中	割り当て中
BES4B		unt2	割り当て中	割り当て中
BES4C		unt3	割り当て中	割り当て中

4 ユニット構成での切り替え先の定義例を次の表に示します。

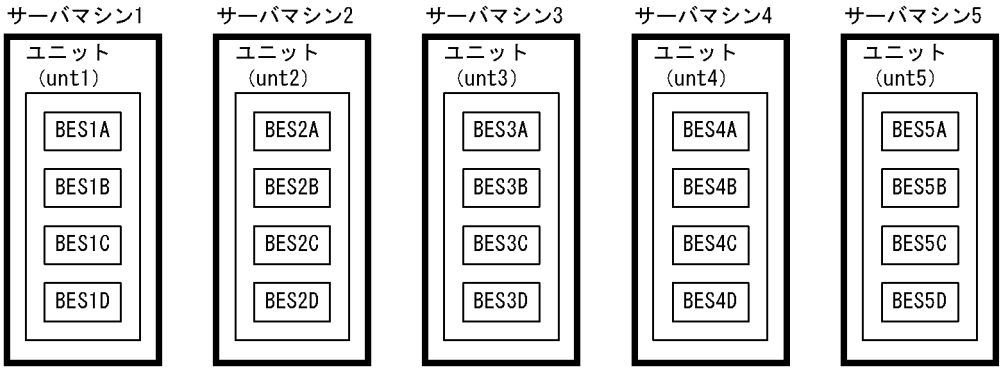
表 26–12　4 ユニット構成時の切り替え先の定義例

サーバ	ホスト BES	優先順位第 1 位	優先順位第 2 位	優先順位第 3 位
BES1A	unt1	unt2	unt4	unt3
BES1B		unt3	unt2	unt4
BES1C		unt4	unt2	unt3
BES2A	unt2	unt3	unt1	unt4
BES2B		unt4	unt1	unt3
BES2C		unt1	unt3	unt4
BES3A	unt3	unt4	unt2	unt1
BES3B		unt1	unt4	unt2
BES3C		unt2	unt4	unt1

サーバ	ホスト BES	優先順位第 1 位	優先順位第 2 位	優先順位第 3 位
BES4A	unt4	unt1	unt3	unt2
BES4B		unt2	unt3	unt1
BES4C		unt3	unt1	unt2

(b)　5 ユニット構成の場合

影響分散スタンバイレス型系切り替え機能を次に示すような 5 ユニット構成で行います。



この場合，次に示すグループごとにグローバルバッファを共用します。

- A グループ：BES1A，BES2A，BES3A，BES4A，BES5A
- B グループ：BES1B，BES2B，BES3B，BES4B，BES5B
- C グループ：BES1C，BES2C，BES3C，BES4C，BES5C
- D グループ：BES1D，BES2D，BES3D，BES4D，BES5D

5 ユニット構成での切り替え先の定義例を次の表に示します。

表 26–13　5 ユニット構成時の切り替え先の定義例

サーバ	ホスト BES	優先順位第 1 位	優先順位第 2 位	優先順位第 3 位	優先順位第 4 位
BES1A	unt1	unt2	unt5	unt3	unt4
BES1B		unt3	unt4	unt2	unt5
BES1C		unt4	unt3	unt5	unt2
BES1D		unt5	unt2	unt4	unt3
BES2A	unt2	unt3	unt1	unt4	unt5
BES2B		unt4	unt5	unt1	unt3
BES2C		unt5	unt4	unt3	unt1
BES2D		unt1	unt3	unt5	unt4
BES3A	unt3	unt4	unt2	unt1	unt5

サーバ	ホスト BES	優先順位第 1 位	優先順位第 2 位	優先順位第 3 位	優先順位第 4 位
BES3B		unt5	unt1	unt2	unt4
BES3C		unt1	unt5	unt4	unt2
BES3D		unt2	unt4	unt5	unt1
BES4A	unt4	unt5	unt3	unt1	unt2
BES4B		unt1	unt2	unt3	unt5
BES4C		unt2	unt1	unt5	unt3
BES4D		unt3	unt5	unt2	unt1
BES5A	unt5	unt1	unt4	unt2	unt3
BES5B		unt2	unt3	unt1	unt4
BES5C		unt3	unt2	unt4	unt1
BES5D		unt4	unt1	unt3	unt2

26.3　IP アドレスの構成例

系切り替え後，IP アドレスを引き継ぐかどうかで，ネットワークの構成及び運用方法が異なります。それぞれの運用方法については，次に示す箇所を参照してください。

- 「IP アドレスを引き継ぐかどうかでのホスト名の運用方法の違い」
- 「系が切り替わった後の運用方法」

ここではネットワーク構成の違いについて説明します。

基本方針

- スタンバイ型系切り替え機能をモニタモードで運用する場合は，基本的に IP アドレスを引き継ぐシステム構成にすることをお勧めします（マルチスタンバイ構成は，IP アドレスを引き継ぐシステム構成の場合だけ適用できます）。IP アドレスを引き継がないシステム構成にすると，IP アドレスを引き継ぐ場合に比べて運用方法が難しくなります。
- 高速系切り替え機能を適用するユニットは IP アドレスを引き継がないシステム構成にしてください。
- スタンバイレス型系切り替え機能では，稼働中のホストが処理を引き継ぐため，IP アドレスの引き継ぎが発生しません。
- エイリアス IP アドレスを使用することをお勧めします。
- サーバモードを適用する場合は，IP アドレスを引き継がない構成になります。
- Microsoft Azure 環境で CLUSTERPRO でのクラスタ構成を構築する場合，IP アドレスを引き継がないシステム構成にしてください。ただし，クライアント接続用の IP アドレスだけを引き継げます。クライアント接続用の IP アドレスだけを引き継ぐ設定の詳細は，「クラスタソフトウェアが CLUSTERPRO の場合」を参照してください。
- Oracle Cloud Infrastructure 環境で CLUSTERPRO でのクラスタ構成を構築する場合，IP アドレスを引き継がないシステム構成にしてください。ただし，クライアント接続用の IP アドレスだけは引き継げます。クライアント接続用の IP アドレスだけを引き継ぐ設定の詳細は，「クラスタソフトウェアが CLUSTERPRO の場合」を参照してください。

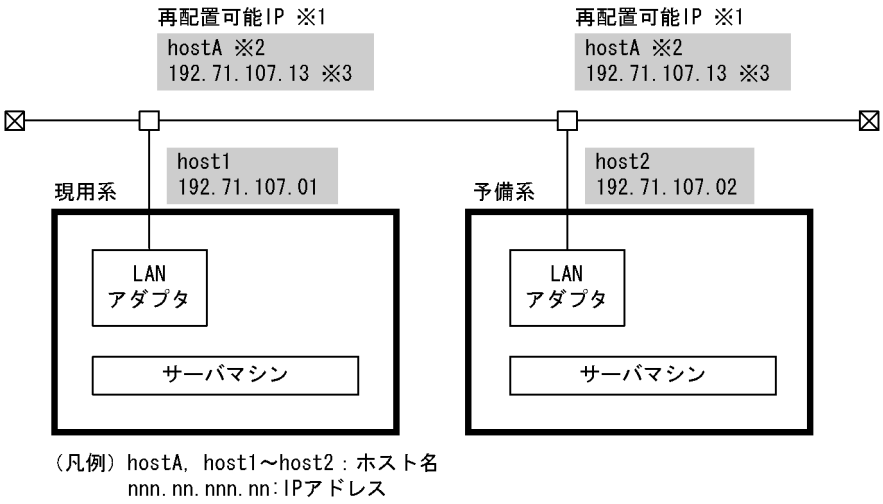
制限事項

サーバモードで運用する場合は IP アドレスを引き継げません。

26.3.1　IP アドレスを引き継ぐ場合

IP アドレスを引き継ぐ場合は，現用系と予備系とで IP アドレス及びホスト名を同じにします。また，この IP アドレスは系切り替え機能によって複数のサーバマシン上で再配置ができるように設定してください。IP アドレスを引き継ぐ場合のネットワーク構成例を次の図に示します。

図 26–29　IP アドレスを引き継ぐ場合のネットワーク構成例



注※１

相互系切り替え構成の場合は，HiRDB 用の再配置可能な IP が各系で二つ必要です。

注※２

HiRDB が系切り替え機能で使用する仮想のネットワーク名です。クラスタソフトウェアに MSFC を使用する場合，この仮想のネットワーク名を MSFC のリソース（ネットワーク名）に登録します。なお，この仮想のネットワーク名を HiRDB のマニュアルでは**仮想ネットワーク名**と表記します。

注※3

HiRDB が系切り替え機能で使用する仮想ネットワーク名の IP アドレスです。クラスタソフトウェアに MSFC を使用する場合，この IP アドレスを MSFC のリソース（IP アドレス）に登録します。

クラスタソフトウェアに CLUSTERPRO を使用する場合，この IP アドレスをフェイルオーバグループのフローティング IP リソースに登録します。

26.3.2　IP アドレスを引き継がない場合

スタンバイ型系切り替え機能の場合

IP アドレスを引き継がない場合は現用系と予備系とで異なる IP アドレス及びホスト名を設定します。したがって，HiRDB のシステム共通定義の **pdunit -c** オペランドに予備系のホスト名を指定する必要があります。

1：1 スタンバイレス型系切り替え機能の場合

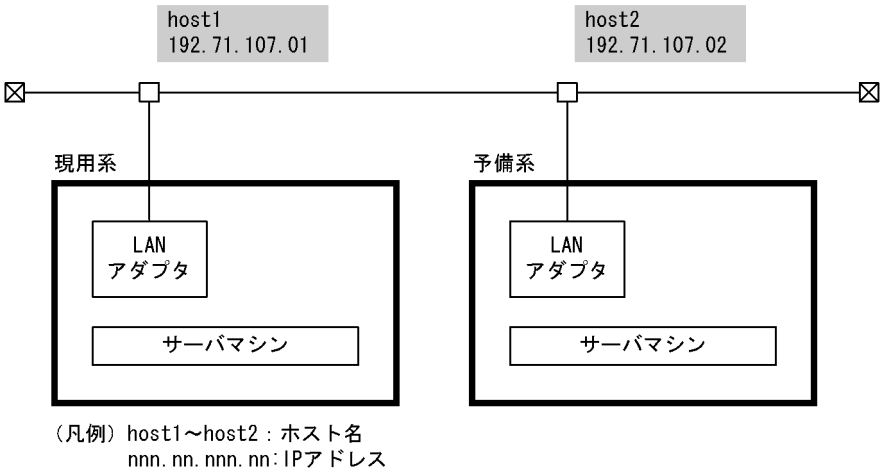
正規 BES ユニットと代替 BES ユニットとで異なる IP アドレス及びホスト名を設定します。また，pdunit -c オペランドを指定する必要はありません。

影響分散スタンバイレス型系切り替え機能の場合

正規ユニットと受け入れユニットとで異なる IP アドレス及びホスト名を設定します。また，pdunit -c オペランドを指定する必要はありません。

IP アドレスを引き継がない場合のネットワーク構成例を次の図に示します。

図 26–30　IP アドレスを引き継がない場合のネットワーク構成例



注意事項（スタンバイ型系切り替え機能限定）

- HiRDB/シングルサーバの場合は，クライアント環境定義の PDHOST オペランドに現用系及び予備系のホスト名を指定してください。
- システムマネジャのユニットが IP アドレスを引き継がない場合は，クライアント環境定義の PDHOST オペランドに現用系及び予備系のホスト名を指定してください。フロントエンドサーバのユニットが IP アドレスを引き継がない場合は，クライアント環境定義の PDFESHOST オペランドに現用系及び予備系のホスト名を指定してください。そうすれば，系が切り替わった後も PDHOST，PDFESHOST オペランドの指定値を変更する必要はありません。
- PDHOST オペランドに現用系及び予備系のホスト名を指定した場合，実行系が現用系から予備系に切り替わると，UAP は一度現用系（待機系）に接続を試みて失敗した後に予備系（実行系）に接続するため，UAP の接続処理時間がその分だけ長くなります。この問題を解決するには，次に示すどちらかの方法で対処してください。
  - クライアント接続用のホスト名にエイリアス IP アドレスを設定し，設定した IP アドレスを引き継ぐようにしてください。この場合のホスト名の指定方法については，「IP アドレスを引き継ぐかどうかでのホスト名の運用方法の違い」を参照してください。
  - 次のクライアント環境定義を指定して，接続先にするホストの切り替え時間を短くしてください。各クライアント環境定義の詳細は，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

    ブロックモードで接続する場合

    ・PDTIMEDOUTRETRY

    ・PDCONREFRCOUNT

    ・PDCONREFRINTERVAL

    ノンブロックモードで接続する場合

    ・PDNBLOCKWAITTIME

    ・PDCONREFRCOUNT

・PDCONREFRINTERVAL

26.4　IP アドレスを引き継ぐかどうかでのホスト名の運用方法の違い

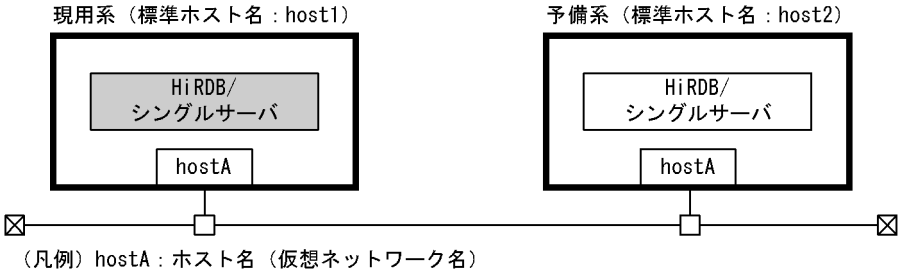
ホスト名の運用方法（指定方法）を IP アドレスを引き継ぐ場合と，引き継がない場合とに分けて説明します。

26.4.1　HiRDB/シングルサーバの場合

(1)　IP アドレスを引き継ぐ場合

IP アドレスを引き継ぐ場合のホスト名の運用方法（指定方法）を次に示します。システム構成は 1：1 系切り替え構成とします。

●システム構成例



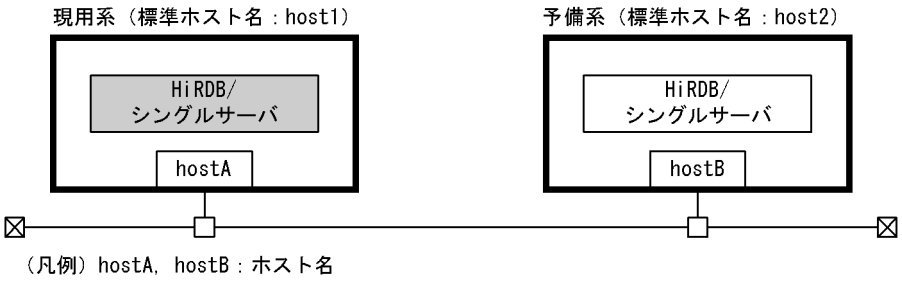
項目	指定するホスト名
システム共通定義に指定するホスト名	pdunit -x hostA pdstart -x hostA
ユニット制御情報定義に指定するホスト名	pd_hostname=host1
rlogin するときに指定するホスト名	hostA
運用コマンドに指定するホスト名※	hostA（ホスト名を省略することをお勧めします）
ユティリティに指定するホスト名	hostA（ホスト名を省略することをお勧めします）
メッセージに表示されるホスト名	hostA
統計情報に表示されるホスト名	hostA
クライアント環境定義の PDHOST オペランドに指定するホスト名	hostA

注※　ホスト名の代わりにユニット識別子が指定できます。

(2)　IP アドレスを引き継がない場合

IP アドレスを引き継がない場合のホスト名の運用方法（指定方法）を次に示します。システム構成は 1：1 系切り替え構成とします。

●システム構成例



ポイント

標準ホスト名を同じ（host1=host2）にすると，系の切り替わりを HiRDB が認識できません。したがって，標準ホスト名は異なる名称にしてください。

項目	指定するホスト名
システム共通定義に指定するホスト名	pdunit -x hostA -c hostB pdstart -x hostA
ユニット制御情報定義に指定するホスト名	pd_hostname=host1
rlogin するときに指定するホスト名	hostA 又は hostB（実行系のホスト名を指定します）
運用コマンドに指定するホスト名※1	hostA（ホスト名を省略することをお勧めします）
ユティリティに指定するホスト名	hostA（ホスト名を省略することをお勧めします）
メッセージに表示されるホスト名	hostA
統計情報に表示されるホスト名	hostA
クライアント環境定義の PDHOST オペランドに指定するホスト名	hostA 及び hostB※2

注※1

ホスト名の代わりにユニット識別子が指定できます。

注※2

クライアント環境定義の PDHOST オペランドに現用系及び予備系のホスト名を指定してください。そうすれば，系が切り替わった後も PDHOST オペランドの指定値を変更する必要はありません。ただし，実行系が現用系から予備系に切り替わると，UAP は一度現用系（待機系）に接続を試みて失敗した後に予備系（実行系）に接続するため，UAP の接続処理時間がその分だけ長くなります。この問題を解決するには，次に示すどちらかの方法で対処してください。

- クライアント接続用のホスト名をエイリアス IP アドレスで設定し，その IP アドレスを引き継ぐようにしてください。この場合のホスト名の指定方法については，「IP アドレスを引き継がない場合（クライアント接続用の IP アドレスだけを引き継ぐ場合）」を参照してください。
- 次のクライアント環境定義を指定して，接続先にするホストの切り替え時間を短くしてください。各クライアント環境定義の詳細は，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

  ブロックモードで接続する場合

  ・PDTIMEDOUTRETRY

  ・PDCONREFRCOUNT

  ・PDCONREFRINTERVAL

  ノンブロックモードで接続する場合

  ・PDNBLOCKWAITTIME

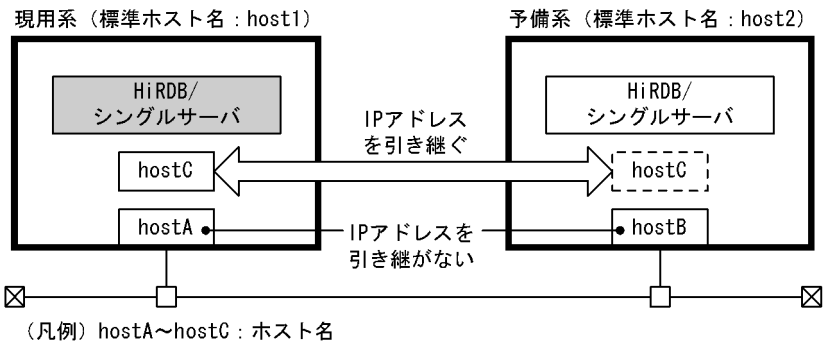
  ・PDCONREFRCOUNT

  ・PDCONREFRINTERVAL

(3) IP アドレスを引き継がない場合（クライアント接続用の IP アドレスだけを引き継ぐ場合）

HiRDB/シングルサーバが使用する IP アドレスは引き継がないが，クライアント接続用の IP アドレスは引き継ぐ場合のホスト名の運用方法（指定方法）を次に示します。システム構成は 1：1 系切り替え構成とします。

●システム構成例



〔説明〕

- hostA 及び hostB は HiRDB/シングルサーバが使用するホスト名で，IP アドレスを引き継ぎません。
- hostC はクライアント接続用のホスト名（エイリアス IP アドレス）で，IP アドレスを引き継ぎます。

> **ポイント**
>
> 標準ホスト名を同じ（host1=host2）にすると，系の切り替わりを HiRDB が認識できません。したがって，標準ホスト名は異なる名称にしてください。

項目	指定するホスト名
システム共通定義に指定するホスト名	pdunit -x hostA -c hostB pdstart -x hostA
ユニット制御情報定義に指定するホスト名	pd_hostname=host1
rlogin するときに指定するホスト名	hostA 又は hostB（実行系のホスト名を指定します）
運用コマンドに指定するホスト名※	hostA（ホスト名を省略することをお勧めします）
ユティリティに指定するホスト名	hostA（ホスト名を省略することをお勧めします）
メッセージに表示されるホスト名	hostA
統計情報に表示されるホスト名	hostA
クライアント環境定義の PDHOST オペランドに指定するホスト名	hostC

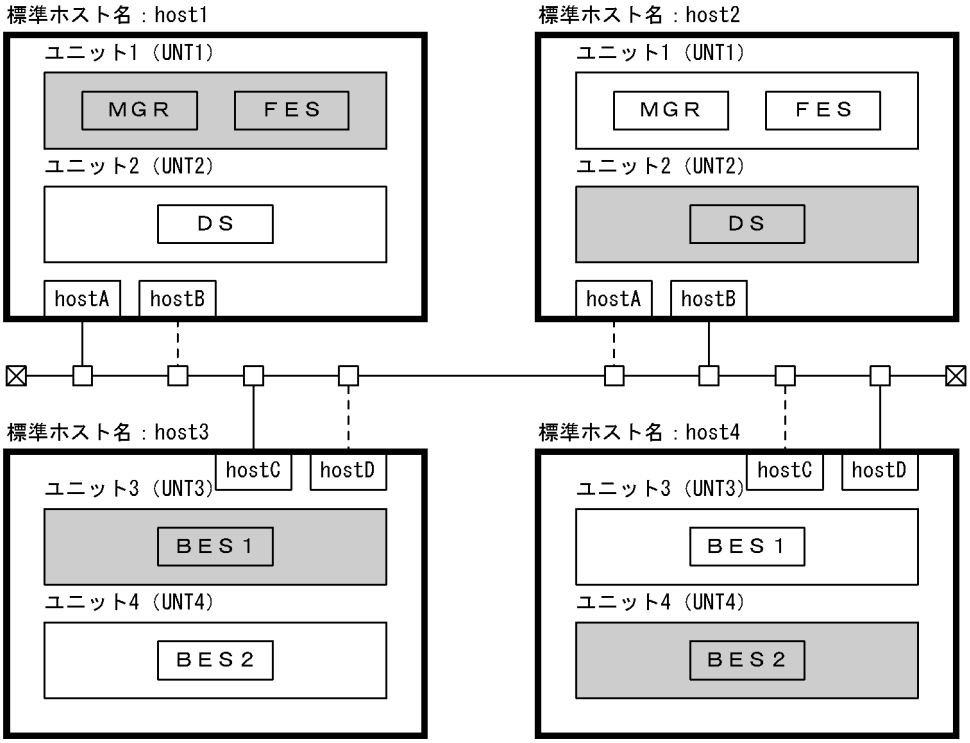
注※　ホスト名の代わりにユニット識別子が指定できます。

26.4.2　HiRDB/パラレルサーバの場合

(1)　IP アドレスを引き継ぐ場合

IP アドレスを引き継ぐ場合のホスト名の運用方法（指定方法）を次に示します。システム構成は相互系切り替え構成とします。

●システム構成例



（凡例）hostA～hostD：ホスト名（仮想ネットワーク名）
UNT1～UNT4：ユニット名
注　網掛けしているユニットが現用系です。

項目		指定するホスト名
システム共通定義に指定するホスト名		pdunit -x hostA -u UNT1 pdunit -x hostB -u UNT2 pdunit -x hostC -u UNT3 pdunit -x hostD -u UNT4
ユニット制御情報定義に指定するホスト名	ユニット 1	pd_hostname=host1
	ユニット 2	pd_hostname=host2
	ユニット 3	pd_hostname=host3
	ユニット 4	pd_hostname=host4
rlogin するときに指定するホスト名	BES1 があるサーバマシンに rlogin する場合	hostC
	BES2 があるサーバマシンに rlogin する場合	hostD
運用コマンドに指定するホスト名※	BES1 に対して運用コマンドを実行する場合	hostC
	BES2 に対して運用コマンドを実行する場合	hostD
ユティリティに指定するホスト名	BES1 に対してユティリティを実行する場合	hostC

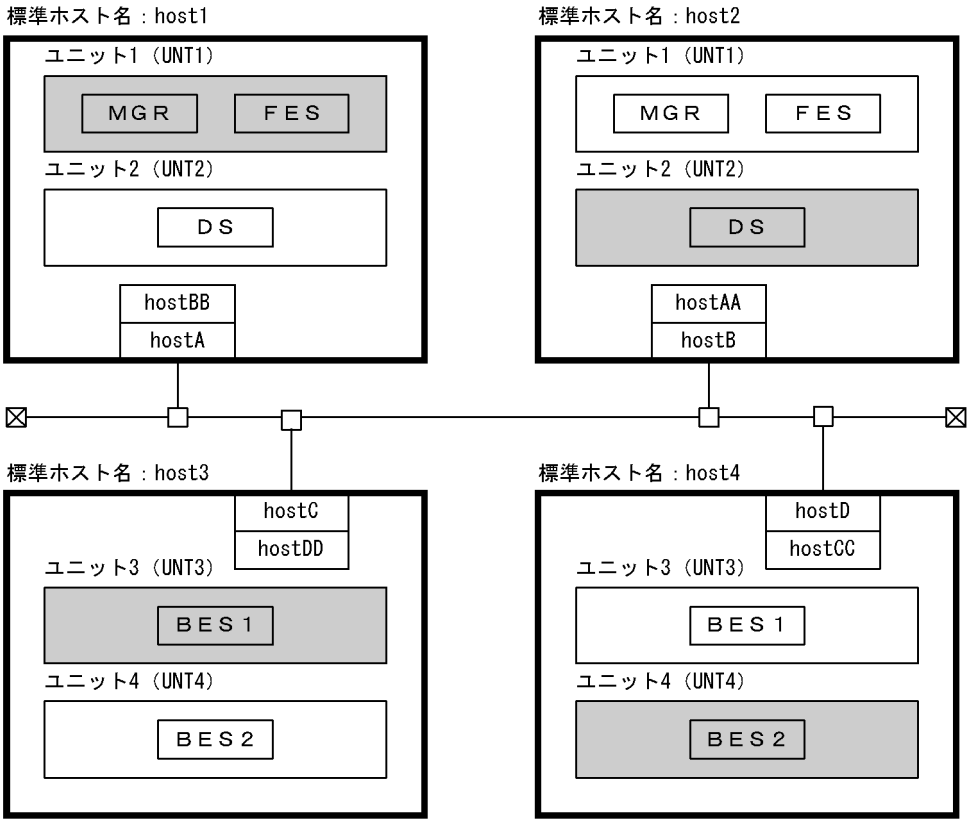
項目		指定するホスト名
	BES2 に対してユティリティを実行する場合	hostD
メッセージに表示されるホスト名	BES1 に対するメッセージの場合	hostC
	BES2 に対するメッセージの場合	hostD
統計情報に表示されるホスト名	BES1 に対する統計情報の場合	hostC
	BES2 に対する統計情報の場合	hostD
次に示すクライアント環境定義に指定するホスト名 • PDHOST • PDFESHOST		hostA

注※　ホスト名の代わりにユニット識別子が指定できます。

(2)　IP アドレスを引き継がない場合

IP アドレスを引き継がない場合のホスト名の運用方法（指定方法）を次に示します。システム構成は相互系切り替え構成とします。

●システム構成例



（凡例）hostA～hostD：現用系で使用するホスト名
hostAA～hostDD：予備系で使用するホスト名
UNT1～UNT4：ユニット名
注　網掛けしているユニットが現用系です。

ポイント

標準ホスト名を同じ（host1=host2，host3=host4）にすると，系の切り替わりを HiRDB が認識できません。したがって，標準ホスト名は異なる名称にしてください。

項目		指定するホスト名
システム共通定義に指定するホスト名		pdunit -x hostA -u UNT1 -c hostAA pdunit -x hostB -u UNT2 -c hostBB pdunit -x hostC -u UNT3 -c hostCC pdunit -x hostD -u UNT4 -c hostDD
ユニット制御情報定義に指定するホスト名	ユニット 1	pd_hostname=host1
	ユニット 2	pd_hostname=host2
	ユニット 3	pd_hostname=host3
	ユニット 4	pd_hostname=host4
rlogin するときに指定するホスト名	BES1 があるサーバマシンに rlogin する場合	hostC 又は hostD（実行系のホスト名を指定します）

項目		指定するホスト名
	BES2 があるサーバマシンに rlogin する場合	hostC 又は hostD（実行系のホスト名を指定します）
運用コマンドに指定するホスト名※1	BES1 に対して運用コマンドを実行する場合	hostC
	BES2 に対して運用コマンドを実行する場合	hostD
ユティリティに指定するホスト名	BES1 に対してユティリティを実行する場合	hostC
	BES2 に対してユティリティを実行する場合	hostD
メッセージに表示されるホスト名	BES1 に対するメッセージの場合	hostC
	BES2 に対するメッセージの場合	hostD
統計情報に表示されるホスト名	BES1 に対する統計情報の場合	hostC
	BES2 に対する統計情報の場合	hostD
次に示すクライアント環境定義に指定するホスト名 • PDHOST • PDFESHOST		hostA 及び hostB※2

注※1

ホスト名の代わりにユニット識別子が指定できます。

注※2

システムマネジャのユニットが IP アドレスを引き継がない場合，クライアント環境定義の PDHOST オペランドに現用系及び予備系のホスト名を指定してください。フロントエンドサーバのユニットが IP アドレスを引き継がない場合，クライアント環境定義の PDFESHOST オペランドに現用系及び予備系のホスト名を指定してください。そうすれば，系が切り替わった後も PDHOST，PDFESHOST オペランドの指定値を変更する必要はありません。ただし，実行系が現用系から予備系に切り替わると，UAP は一度現用系（待機系）に接続を試みて失敗した後に予備系（実行系）に接続するため，UAP の接続処理時間がその分だけ長くなります。この問題を解決するには，次に示すどちらかの方法で対処してください。

- クライアント接続用のホスト名をエイリアス IP アドレスで設定し，その IP アドレスを引き継ぐようにしてください。この場合のホスト名の指定方法については，「IP アドレスを引き継がない場合（クライアント接続用の IP アドレスだけを引き継ぐ場合）」を参照してください。
- 次のクライアント環境定義を指定して，接続先にするホストの切り替え時間を短くしてください。各クライアント環境定義の詳細は，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

  ブロックモードで接続する場合

  ・PDTIMEDOUTRETRY

  ・PDCONREFRCOUNT

・PDCONREFRINTERVAL

ノンブロックモードで接続する場合

・PDNBLOCKWAITTIME

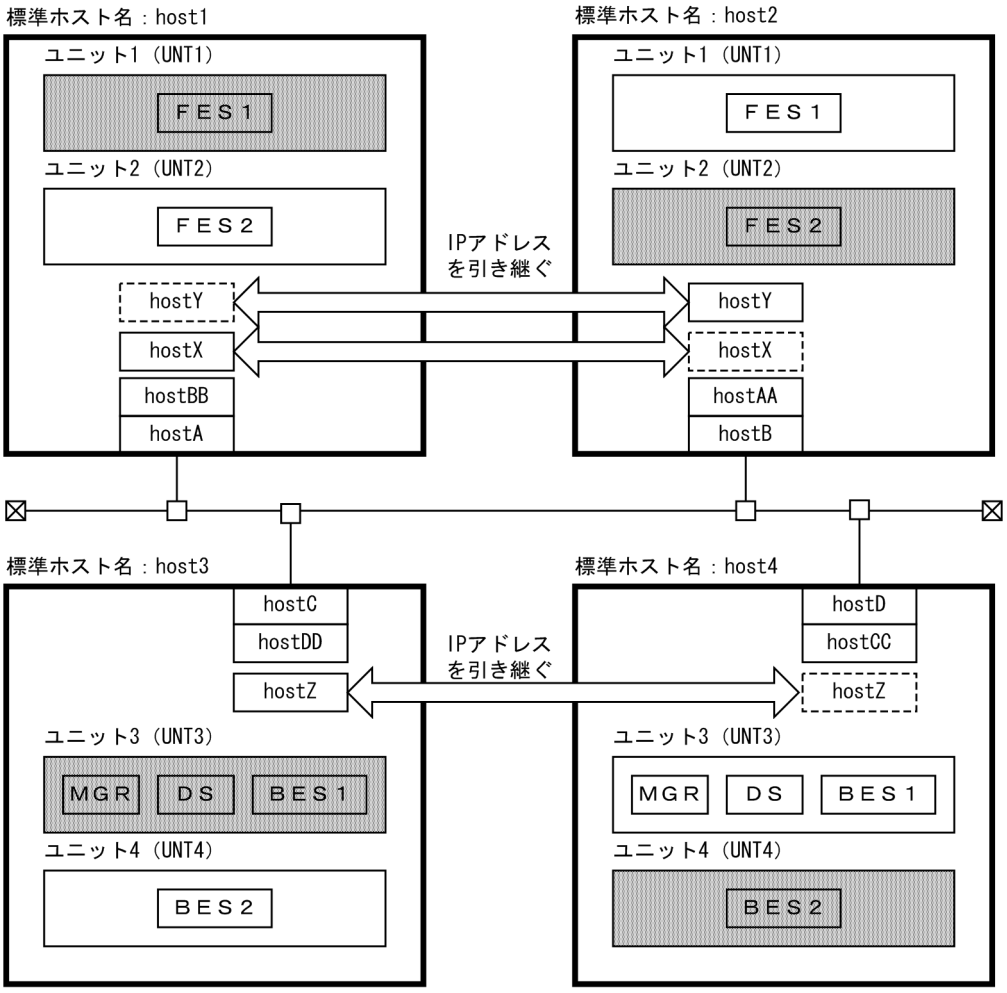
・PDCONREFRCOUNT

・PDCONREFRINTERVAL

(3) IP アドレスを引き継がない場合（クライアント接続用の IP アドレスだけを引き継ぐ場合）

HiRDB/パラレルサーバが使用する IP アドレスは引き継がないが，クライアント接続用の IP アドレスは引き継ぐ場合のホスト名の運用方法（指定方法）を次に示します。システム構成は相互系切り替え構成とします。

●システム構成例



（凡例）hostA～hostD：現用系で使用するホスト名
hostAA～hostDD：予備系で使用するホスト名
hostX～hostZ：クライアント接続用のホスト名
UNT1～UNT4：ユニット名

注　網掛けしているユニットが現用系です。

〔説明〕

- hostA はユニット 1（現用系）が使用するホスト名で，IP アドレスを引き継ぎません。hostAA はユニット 1（待機系）が使用するホスト名です。
- hostB はユニット 2（現用系）が使用するホスト名で，IP アドレスを引き継ぎません。hostBB はユニット 2（待機系）が使用するホスト名です。
- hostC はユニット 3（現用系）が使用するホスト名で，IP アドレスを引き継ぎません。hostCC はユニット 3（待機系）が使用するホスト名です。
- hostD はユニット 4（現用系）が使用するホスト名で，IP アドレスを引き継ぎません。hostDD はユニット 4（待機系）が使用するホスト名です。
- hostX～hostZ はクライアント接続用のホスト名（エイリアス IP アドレス）で，IP アドレスを引き継ぎます。

ポイント

標準ホスト名を同じ（host1=host2，host3=host4）にすると，系の切り替わりを HiRDB が認識できません。したがって，標準ホスト名は異なる名称にしてください。

項目		指定するホスト名
システム共通定義に指定するホスト名		pdunit -x hostA -u UNT1 -c hostAA pdunit -x hostB -u UNT2 -c hostBB pdunit -x hostC -u UNT3 -c hostCC pdunit -x hostD -u UNT4 -c hostDD
ユニット制御情報定義に指定するホスト名	ユニット 1	pd_hostname=host1
	ユニット 2	pd_hostname=host2
	ユニット 3	pd_hostname=host3
	ユニット 4	pd_hostname=host4
rlogin するときに指定するホスト名	BES1 があるサーバマシンに rlogin する場合	hostC 又は hostD（実行系のホスト名を指定します）
	BES2 があるサーバマシンに rlogin する場合	hostC 又は hostD（実行系のホスト名を指定します）
運用コマンドに指定するホスト名※	BES1 に対して運用コマンドを実行する場合	hostC
	BES2 に対して運用コマンドを実行する場合	hostD
ユティリティに指定するホスト名	BES1 に対してユティリティを実行する場合	hostC
	BES2 に対してユティリティを実行する場合	hostD

項目		指定するホスト名
メッセージに表示されるホスト名	BES1 に対するメッセージの場合	hostC
	BES2 に対するメッセージの場合	hostD
統計情報に表示されるホスト名	BES1 に対する統計情報の場合	hostC
	BES2 に対する統計情報の場合	hostD
クライアント環境定義に指定するホスト名	PDHOST	hostZ
	PDFESHOST	hostX 又は hostY（接続するフロントエンドサーバのホスト名を指定します）

注※　ホスト名の代わりにユニット識別子が指定できます。

26.5　HiRDB に関する準備

実行者　HiRDB 管理者

ここでは，HiRDB に関する準備方法について説明します。説明する項目は次のとおりです。

- 前提条件及び注意事項
- 共有ディスク装置の準備
- HiRDB システム定義の作成
- クライアント環境定義の指定
- HiRDB システム定義及びクライアント環境定義のホスト名の指定例
- RD エリアの作成
- グローバルバッファの定義（スタンバイレス型系切り替え機能限定）
- グローバルバッファの定義（影響分散スタンバイレス型系切り替え機能限定）
- HiRDB サービスの開始方法の設定の確認（HiRDB/シングルサーバ限定）
- HiRDB に関連するサービスの準備
- 監査証跡ファイルの運用
- NetBackup 連携機能を使用する場合の注意事項

26.5.1　前提条件及び注意事項

(1)　スタンバイ型系切り替え機能の場合

(a)　現用系と予備系で一致させること

現用系及び予備系で，次に示すことを一致させてください。

- HiRDB 及び関連プログラムプロダクトのバージョン
- HiRDB 管理者の環境(ユーザ ID，グループ ID，環境変数)
- HiRDB 運用ディレクトリの絶対パス名
- HiRDB システム定義の内容
- HiRDB ファイルの設定形式
- ユーザの実行形式プログラム
- サーバマシンに設定している時間
- HiRDB のサービス名（セットアップ識別子）

(b)　環境設定に関する注意事項

- 現用系及び予備系の両方に HiRDB の環境設定を行ってください。また，現用系及び予備系 HiRDB のバージョンを合わせてください。また，HiRDB をバージョンアップする場合は，現用系及び予備系の両方をバージョンアップしてください。
- HiRDB 運用ディレクトリを共有ディスク上に作成しないでください。
- サーバモードで実行する場合は，次に示す製品をインストールして環境設定をしてください。
  - Hitachi HA Toolkit Extension（現用系及び予備系のマシン）
- 回復不要 FES ユニットには系切り替え機能を適用できません。
- マルチスタンバイ構成を適用する場合は，クラスタソフトウェアのマルチスタンバイ機能を有効にする必要があります。マルチスタンバイ機能の設定方法については，各クラスタソフトウェアのマニュアルを参照してください。

(c)　HiRDB をインストールするときの注意事項

系切り替え機能を使用する場合は，現用系及び予備系それぞれのサーバマシンのローカルディスク上に，同じドライブ名称，同じディレクトリ名称で HiRDB をインストールしてください。ドライブ名称及びディレクトリ名称は大文字小文字も統一してください。また，ディレクトリ名称には全角文字及び特殊記号は使用しないでください。

Hitachi HA Toolkit Extension を使用している場合は，HiRDB が強制終了又は異常終了した後に HiRDB をインストールしないでください。インストールすると HiRDB を開始できないことがあります。

(d)　その他

系切り替えの環境では同じサービス名を別サーバマシンで起動します。既存環境から系切り替えの環境を作成する場合，サービス名が重複することがあります。この場合，既存のサービス名を変更してください。サービス名を変更する場合は，既存の HiRDB をアンインストールした後に，同じディレクトリ下に再度インストールしてください。また，ディレクトリ構成が重複している場合はサーバの移行操作をして環境を新規に作成してください。サーバの移行方法については，「システム構成を変更する方法」を参照してください。

(2)　1：1 スタンバイレス型系切り替え機能の場合

- 次に示す製品をインストールして環境設定をしてください。
  - HiRDB Advanced High Availability（全サーバマシン）
  - Hitachi HA Toolkit Extension（正規 BES ユニットと代替 BES ユニット）
- 正規 BES ユニット及び代替 BES ユニット内にはバックエンドサーバだけを定義してください。ユニット内にバックエンドサーバ以外のサーバを定義すると，そのユニットには 1：1 スタンバイレス型系切り替え機能を適用できません。
- 代替 BES 用の HiRDB 運用ディレクトリの準備は不要：

1：1 スタンバイレス型系切り替え機能では，代替 BES ユニットの HiRDB 運用ディレクトリを使用するため，代替 BES のための HiRDB 運用ディレクトリの準備は不要です。

- システム定義ファイルの配置：

  グループを構成する各ユニットに，全バックエンドサーバ分のバックエンドサーバ定義ファイルを配置してください。バックエンドサーバ定義のデフォルト値としてユニット制御情報定義に設定するパラメタは，システム共通定義，又はバックエンドサーバ定義ファイルに定義しておく必要があります。

(3) 影響分散スタンバイレス型系切り替え機能の場合

- 次に示す製品をインストールして環境設定をしてください。
  - HiRDB Advanced High Availability（全サーバマシン）
  - Hitachi HA Toolkit Extension（正規ユニットと受け入れユニット）
- 正規ユニット及び受け入れユニット内にはバックエンドサーバだけを定義してください。ユニット内にバックエンドサーバ以外のサーバを定義すると，そのユニットには影響分散スタンバイレス型系切り替え機能を適用できません。
- ゲスト BES 用の HiRDB 運用ディレクトリの準備は不要：

  影響分散スタンバイレス型系切り替え機能では，受け入れユニットの HiRDB 運用ディレクトリを使用するため，ゲスト BES のための HiRDB 運用ディレクトリの準備は不要です。
- システム定義ファイルの配置：

  グループを構成する各ユニットに，HA グループ内の全バックエンドサーバ分のバックエンドサーバ定義ファイルを配置してください。バックエンドサーバ定義のデフォルト値としてユニット制御情報定義に設定するパラメタは，システム共通定義，又はバックエンドサーバ定義ファイルに定義しておく必要があります。

26.5.2 共有ディスク装置の準備

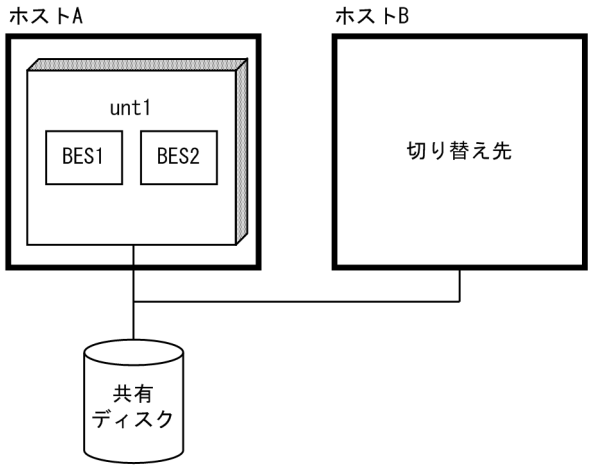
現用系と予備系（スタンバイレス型系切り替え機能の場合は正規 BES と代替 BES）で共有する外付けハードディスクが必要です。このハードディスクを**共有ディスク装置**といいます。

(1) 共有ディスクの割り当て

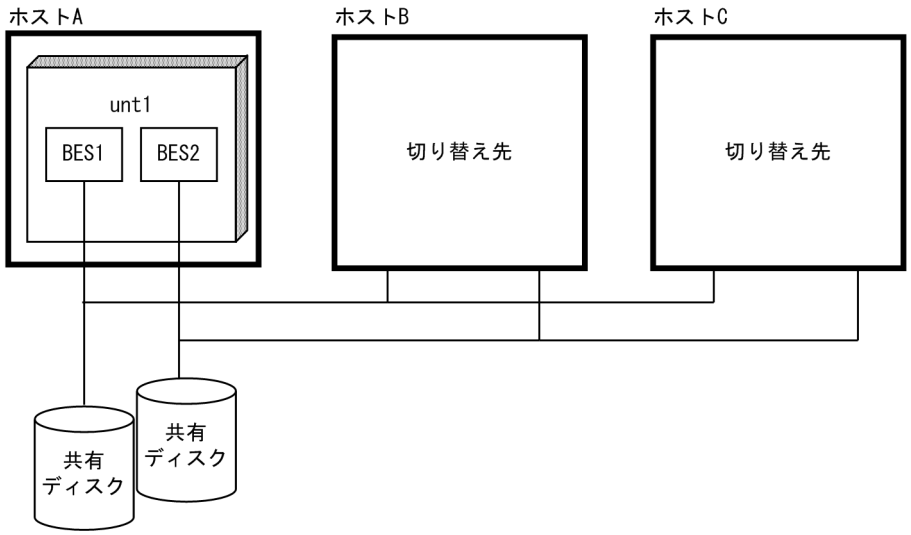
共有ディスクの割り当てを次の図に示します。

図 26–31　共有ディスクの割り当て

1. ユニットごとに共有ディスクを割り当てる場合



2. サーバごとに共有ディスクを割り当てる場合



〔説明〕

1. スタンバイ型系切り替え機能及び 1：1 スタンバイレス型系切り替え機能の場合は，ユニット単位の切り替えのためユニットごとに共有ディスクを割り当てます。
2. 影響分散スタンバイレス型系切り替え機能の場合は，サーバ単位の切り替えのためサーバごとに共有ディスクを割り当てます。複数のサーバに関する情報を一つの共有ディスクに配置することはできません。

共有ディスク装置には次に示す HiRDB ファイルシステム領域を作成します。

- RD エリア用の HiRDB ファイルシステム領域
- システムファイル用の HiRDB ファイルシステム領域
- バックアップファイル用の HiRDB ファイルシステム領域
- アンロードログファイル用の HiRDB ファイルシステム領域

- インデクス情報ファイル用 HiRDB ファイルシステム領域（プラグインインデクス遅延一括作成機能を使用する場合）

注意事項

- これらの HiRDB ファイルシステム領域は両方（現用系及び予備系）の HiRDB から同じパス名で参照できるように設定してください。1：1 スタンバイレス型系切り替え機能の場合は，正規 BES ユニットと代替 BES ユニットの両方から同じパス名で参照できるように設定してください。影響分散スタンバイレス型系切り替え機能でも同様に，HA グループ内のすべてのユニットから同一パス名で参照できるように設定してください。ただし，影響分散スタンバイレス型系切り替え機能の場合，ユニットステータスファイルはサーバステータスファイル，システムログファイル，シンクポイントダンプファイルと異なる独立した非共有ディスクに作成してください。
- 共用 RD エリア用 HiRDB ファイルシステム領域を作成した共有ディスクは全ユニットから読み書きモードでアクティブにしておく必要があります。このため，系切り替え機能に伴って非アクティブ化，及びアクティブ化をしてはなりません。
- 前記の HiRDB ファイルシステム領域を作成する場合，pdfmkfs コマンドの-k オプションに SVR（省略値）を指定しないでください。-k オプションに SVR を指定して作成した HiRDB ファイルシステム領域に対して，HiRDB は Windows のファイルキャッシュに書き込んだ時点で処理を続行します。このため，ディスク上にデータが書き込まれる前に異常が発生するとファイルの整合性がとれなくなります。
- 前記の HiRDB ファイルシステム領域を作成する場合，pdfmkfs コマンドに-i オプションを指定してください。-i オプションを指定しないと，HiRDB ファイルシステム領域の拡張とサーバマシンの電源異常が同時に発生したときにファイルが破壊されることがあります。

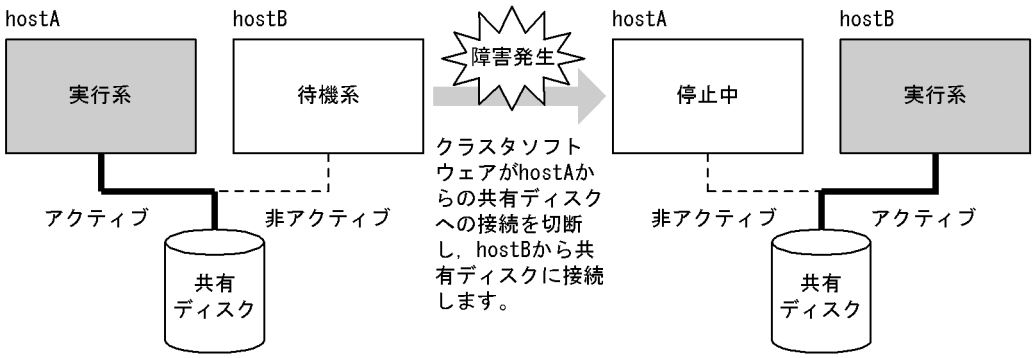
(2) 共有ディスクのアクセス制御

系切り替え機能を使用する場合，系の切り替え元と切り替え先の両方から同時に共有ディスクにアクセスが行われると，データベースが壊れる可能性があります。そのため，両方の系から共有ディスクをアクセスできないように制御を行う必要があります。共有ディスクのアクセス制御はクラスタソフトウェアが行います。

●クラスタソフトウェアによる共有ディスクのアクセス制御

クラスタソフトウェアが共有ディスクのアクセス制御を行います。実行系をアクティブに，待機系及び停止中の系を非アクティブに制御し，実行系だけが共有ディスクにアクセスできるようにします。クラスタソフトウェアによる共有ディスクのアクセス制御を次の図に示します。

図 26–32　クラスタソフトウェアによる共有ディスクのアクセス制御



〔説明〕

非アクティブの系からは共有ディスクをアクセスできません。そのため，実行系だけが共有ディスクにアクセスできます。

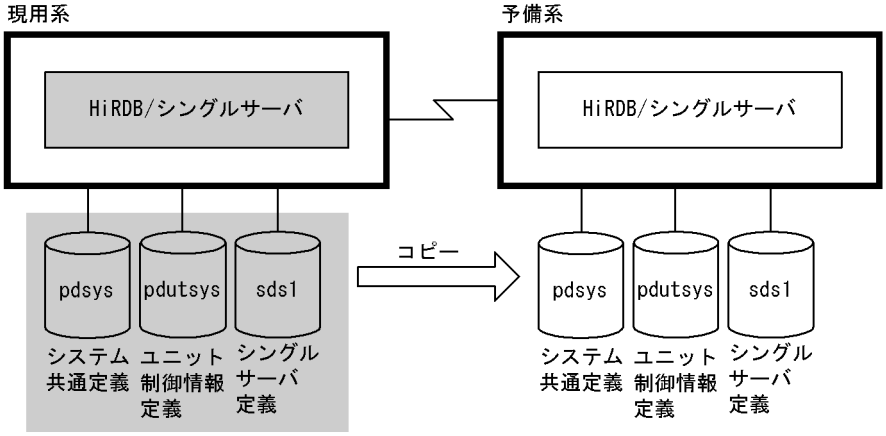
共有ディスクの切り替え方法（アクティブ，非アクティブの切り替え方法）については，クラスタソフトウェアのマニュアルを参照してください。

26.5.3　HiRDB システム定義の作成

(1)　スタンバイ型系切り替え機能の場合

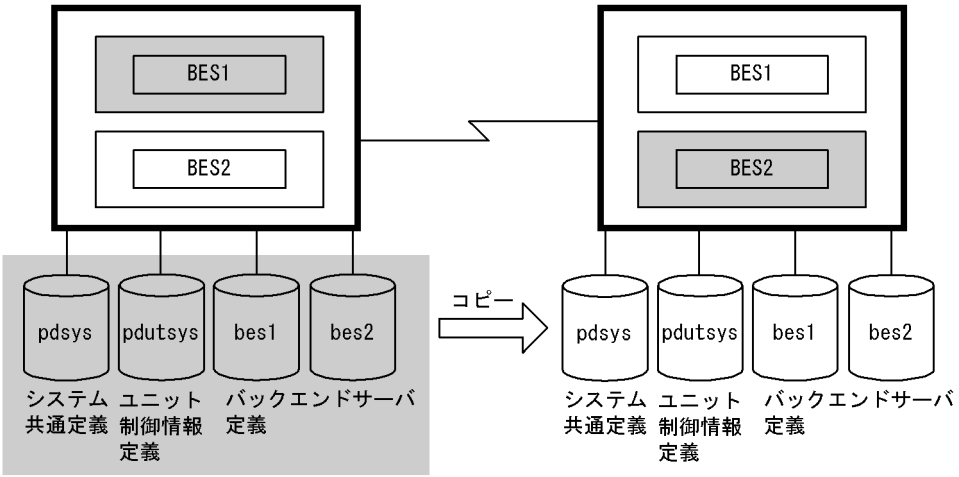
現用系，予備系とも同じ HiRDB システム定義を使用します。現用系の HiRDB システム定義を作成した後に，その HiRDB システム定義を予備系にコピーしてください。HiRDB システム定義ファイルの構成例を次の図に示します。

図 26–33　スタンバイ型系切り替え機能使用時の HiRDB システム定義ファイルの構成例（HiRDB/シングルサーバの場合）



注　1：1系切り替え構成の例です。

図 26–34　スタンバイ型系切り替え機能使用時の HiRDB システム定義ファイルの構成例（HiRDB/パラレルサーバの場合）



（凡例）　：現用系のユニット
注　相互系切り替え構成の例です。

(2)　1：1 スタンバイレス型系切り替え機能の場合

正規 BES ユニットのユニット制御情報定義ファイル及びバックエンドサーバ定義ファイルを代替 BES ユニットにコピーします。そのとき，ユニット制御情報定義ファイルの名称を次に示すように変更してください。

pdutsys.正規 BES ユニットのユニット識別子

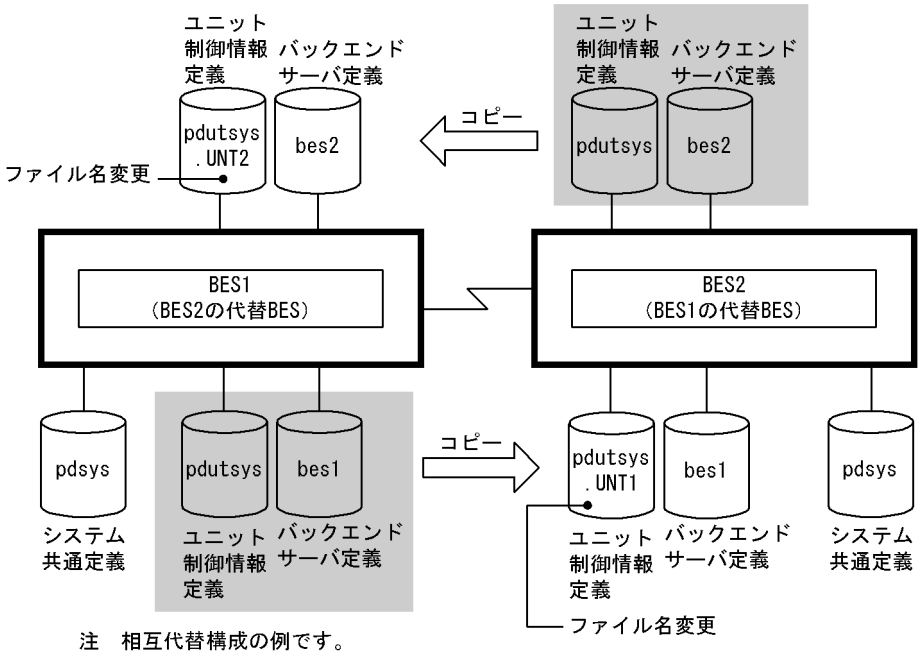
この定義ファイルに指定したオペランドのうち，代替中に設定値が有効になるオペランドを次に示します。次に示すオペランド以外は代替 BES ユニットのユニット制御情報定義ファイルに設定したオペランドの値が有効になります。

- pd_syssts_file_name_1〜7
- pd_syssts_singleoperation
- pd_syssts_initial_error
- pd_syssts_last_active_file
- pd_syssts_last_active_side
- pd_audit
- pd_aud_file_name
- pd_aud_max_generation_size
- pd_aud_max_generation_num
- pd_aud_async_buff_size
- pd_aud_async_buff_count
- pd_ha_switch_timeout

- pd_rpl_hdepath

スタンバイレス型系切り替え機能使用時の HiRDB システム定義ファイルの構成例（相互代替構成の場合）を次の図に示します。

図 26–35　スタンバイレス型系切り替え機能使用時の HiRDB システム定義ファイルの構成例（相互代替構成の場合）



注　相互代替構成の例です。

〔説明〕

- 正規 BES ユニット（BES1）のユニット制御情報定義ファイル及びバックエンドサーバ定義ファイルを代替 BES ユニット（BES2）にコピーします。そのとき，ユニット制御情報定義ファイルの名称を pdutsys.UNT1 に変更します。
- 正規 BES ユニット（BES2）のユニット制御情報定義ファイル及びバックエンドサーバ定義ファイルを代替 BES ユニット（BES1）にコピーします。そのとき，ユニット制御情報定義ファイルの名称を pdutsys.UNT2 に変更します。

(3)　影響分散スタンバイレス型系切り替え機能の場合

影響分散スタンバイレス型系切り替え使用時のシステム定義ファイルの運用方法を次の表に示します。

表 26–14　影響分散スタンバイレス型系切り替え使用時のシステム定義ファイルの運用方法

定義種別	定義ファイルの運用方法
システム共通定義	システム内全ユニットにファイルをコピーします。 バックエンドサーバ定義のデフォルト値として設定するパラメタは，システム共通定義に記述してください。
ユニット制御情報定義	次のオペランド（システム共通定義に記述できないオペランド）だけを記述してください。

定義種別	定義ファイルの運用方法
	• pd_aud_file_name※ • pd_down_watch_proc • pd_unit_id • pd_hostname • pd_ha_unit • pd_rpl_hdepath • pd_ha_acttype • pd_ha_server_process_standby • pd_ha_agent • pd_ha_max_act_guest_servers • pd_ha_max_server_process • pd_ha_resource_act_wait_time • pd_ha_process_count • pd_lck_deadlock_check • pd_lck_deadlock_check_interval • pd_prf_file_count • pd_prf_file_size • pd_prf_level • pd_prf_output_directory • pd_rpc_trace_name※ • pd_registered_port※ • pd_service_port • pd_syssts_file_name_1～7 • pd_syssts_initial_error • pd_syssts_last_active_file • pd_syssts_last_active_side • pd_syssts_singleoperation • pd_tmp_directory • pd_trn_rcvmsg_store_buflen 上記以外のオペランドはシステム共通定義，又はバックエンドサーバ定義に記述してください。上記以外のオペランドを記述した場合は，エラーとなります（メッセージ KFPS05618-E を出力します）。
サーバ共通定義	HA グループ内全ユニットにファイルをコピーします。
バックエンドサーバ定義	HA グループ内全ユニットにファイルをコピーします。

注※

同一サーバマシンに複数のユニットを配置する場合，このオペランドには，ユニットごとに異なる値を指定してください。

(4) 系切り替え機能に関する HiRDB システム定義のオペランド

ここでは，系切り替え機能に関するオペランドについて説明します。系切り替え機能に関する HiRDB システム定義のオペランドを次の表に示します。

表 26–15　系切り替え機能に関する HiRDB システム定義のオペランド

オペランド名	説明
pd_ha	系切り替え機能を使用する場合に指定します。
pd_ha_ipaddr_inherit	IP アドレスを引き継ぐかどうかを指定します。マルチスタンバイ構成を適用する場合は，Y を指定してください。サーバモードを適用するユニットには N を指定してください。スタンバイレス型系切り替え機能を適用するユニットはこのオペランドを省略してください。 **Y：**IP アドレスを引き継ぎます。 **N：**IP アドレスを引き継ぎません。 サーバモードで運用する場合は IP アドレスを引き継げません。
pd_ha_acttype	系切り替え機能をモニタモードで運用するか，サーバモードで運用するかを指定します。pd_ha_acttype=server を指定する場合は，pd_ha_ipaddr_inherit=N（IP アドレスを引き継がない）にする必要があります。 **monitor：**系切り替え機能をモニタモードで運用します。 **server：**系切り替え機能をサーバモードで運用します。
pd_ha_server_process_standby	ユーザサーバホットスタンバイを使用するかどうかを指定します。 **Y：**ユーザサーバホットスタンバイを使用します。 **N：**ユーザサーバホットスタンバイを使用しません。
pd_ha_agent	高速系切り替え機能，1：1 スタンバイレス型系切り替え機能，又は影響分散スタンバイレス型系切り替え機能を使用する場合に指定します。 **standbyunit：**高速系切り替え機能を使用します。 **server：**1：1 スタンバイレス型系切り替え機能を使用します。 **activeunits**：影響分散スタンバイレス型系切り替え機能を使用します。
pd_ha_transaction pd_ha_trn_queuing_wait_time pd_ha_trn_restart_retry_time	• トランザクションキューイング機能を使用する場合に指定します。 • pd_ha_transaction オペランドに queuing を指定した場合，系の切り替え中に最大同時接続数（pd_max_users オペランドの値）を超えたときに，HiRDB サーバへの接続処理を pd_ha_trn_queuing_wait_time ＋ pd_ha_trn_restart_retry_time の時間だけ HiRDB クライアントでリトライします。
pd_ha_trn_queuing_initial	フロントエンドサーバを配置したユニットの起動直後からトランザクションキューイング機能を有効にする場合に指定します。
pd_ha_switch_timeout	このオペランドはサーバモードの場合に指定できます。モニタモードの場合にこのオペランドを指定しても無効になります。 系切り替え時の HiRDB（HiRDB/パラレルサーバの場合はユニット）の内部停止処理が**サーバ障害監視時間**を超えた場合に，HiRDB の内部停止処理を待たないで系を切り替えるかどうかを指定します。ここでいうサーバ障害監視時間とは，Hitachi HA Toolkit Extension の patrol オペランドに指定した時間のことです。 Hitachi HA Toolkit Extension の patrol オペランドについては，マニュアル「Hitachi HA Toolkit」を参照してください。 **Y：**系切り替え時の HiRDB の内部停止処理がサーバ障害監視時間を超えた場合，HiRDB の内部停止処理を待たないで系を切り替えます。このとき，HiRDB のスローダウンとして系を切り替えます。 **N：**系切り替え時の HiRDB の内部停止処理が終わるまで系を切り替えません。

オペランド名		説明
pd_ha_unit		系切り替え機能を適用しないユニットがある場合に nouse を指定します。回復不要 FES ユニットには nouse を指定してください。
pd_mode_conf		HiRDB（ユニット）の開始方法に関するオペランドです。指定値の目安を次に示します。 モニタモードの場合は MANUAL1 を指定してください。ただし，クラスタソフトウェアに CLUSTERPRO を使用して，カスタム監視リソースを作成する場合は MANUAL2 を指定してください。 サーバモードの場合は次のように指定してください。 • Hitachi HA Toolkit Extension の server 定義文の swtichtype オペランドに switch を指定した場合は，MANUAL2 を指定してください。 • Hitachi HA Toolkit Extension の server 定義文の swtichtype オペランドに restart 又は manual を指定した場合は，MANUAL1 を指定してください。
pd_hostname		現用系の標準ホスト名を指定します。スタンバイレス型系切り替え機能の場合はユニットの標準ホスト名を指定します（系切り替え機能を使用しない場合と同じです）。
pdunit	-x	現用系のホスト名を指定します。マルチスタンバイ構成の場合は，IP アドレスを引き継ぐホスト名を指定してください。スタンバイレス型系切り替え機能の場合はユニットのホスト名を指定します（系切り替え機能を使用しない場合と同じです）。
	-u	ユニット識別子を指定します。
	-d	HiRDB 運用ディレクトリ名を指定します。マルチスタンバイ構成の場合は，現用系とすべての予備系で同じディレクトリ名を指定してください。1：1 スタンバイレス型系切り替え機能の場合は，正規 BES ユニットと代替 BES ユニットで同じディレクトリ名を指定してください。影響分散スタンバイレス型系切り替え機能の場合は，HA グループ内の全ユニットで同じディレクトリ名を指定してください。
	-c	予備系のホスト名を指定します。IP アドレスを引き継がない場合に指定します。スタンバイレス型系切り替え機能の場合はこのオプションを省略してください。
	-p	HiRDB のポート番号を指定します。1：1 スタンバイレス型系切り替え機能の場合，正規 BES ユニットと代替 BES ユニットのポート番号※1は同じ番号を指定してください。影響分散スタンバイレス型系切り替え機能の場合は，HA グループ内の全ユニットで同じポート番号を指定してください。
	-s	スケジューラのポート番号を指定します。1：1 スタンバイレス型系切り替え機能の場合，正規 BES ユニットと代替 BES ユニットのポート番号※2は同じ番号を指定してください。影響分散スタンバイレス型系切り替え機能の場合は，HA グループ内の全ユニットで同じポート番号を指定してください。
	-t	トランザクションサーバのポート番号を指定します。1：1 スタンバイレス型系切り替え機能の場合，正規 BES ユニットと代替 BES ユニットのポート番号※3は同じ番号を指定してください。影響分散スタンバイレス型系切り替え機能の場合は，HA グループ内の全ユニットで同じポート番号を指定してください。
pdstart	-c	代替 BES 名を指定します。1：1 スタンバイレス型系切り替え機能の場合に指定します。
	-g	影響分散スタンバイレス型系切り替え機能を使用する場合に，サーバの移動先となるユニットの集合である HA グループの識別子を指定します。
pdbuffer	-c	代替中に代替部が使用するグローバルバッファを割り当てる場合にこのオプションを指定します。スタンバイレス型系切り替え機能の場合に指定します。1：1 スタンバイレス

オペランド名		説明
		型系切り替え機能の場合は，「グローバルバッファの定義（1：1 スタンバイレス型系切り替え機能限定）」を参照してください。影響分散スタンバイレス型系切り替え機能の場合は，「グローバルバッファの定義（影響分散スタンバイレス型系切り替え機能限定）」を参照してください。
pdhagroup	-g	影響分散スタンバイレス型系切り替え機能を使用する場合にサーバの切り替え先となるユニットの集合として HA グループを定義します。システム内で HA グループを一意に識別するための識別子を指定します。
	-u	HA グループを構成するユニットのユニット識別子を指定します。
pd_ha_max_act_guest_servers		影響分散スタンバイレス型系切り替え機能を使用する場合に，該当ユニット内で同時に実行系として稼働できるゲスト BES 数の最大値を指定します。
pd_ha_max_server_process		影響分散スタンバイレス型系切り替え機能を使用する場合に，該当ユニット内の最大起動ユーザサーバプロセス数を指定します。
pd_ha_resource_act_wait_time		影響分散スタンバイレス型系切り替え機能を使用する場合，ユニットを開始するときに，実行系サーバのリソースが活性化されるまでの最大待ち時間を指定します。
pd_service_port		同一サーバマシン内複数ユニット構成の場合（相互系切り替え構成を含みます）にこのオペランドを指定するときは注意が必要です。同一サーバマシン内複数ユニット構成の場合（相互系切り替え構成を含みます），このオペランドはユニット制御情報定義でユニットごとに別々のポート番号を指定してください。 次に示す指定をした場合は，どちらかのユニットへの系の切り替えに失敗します。 • システム共通定義の pd_service_port オペランドを指定した（ユニット制御情報定義の pd_service_port オペランドは指定していない） • ユニット制御情報定義の pd_service_port オペランドに，ほかのユニット制御情報定義の pd_service_port オペランドと同じポート番号を指定した
pd_ha_mgr_rerun		このオペランドに notwait を指定すると，システムマネジャユニットの系切り替え時（切り替え先の系での開始処理時）に，各ユニットからの開始処理完了の連絡待ちを行いません。これによって，停止中のユニットがある状態でもシステムマネジャユニットの系切り替えが実行できるようになります。 運用方法の詳細については，「停止中のユニットがあるためシステムマネジャユニットの系切り替えに失敗するときの対処方法」を参照してください。
pd_max_ard_process		1：1 スタンバイレス型系切り替え機能の場合，正規 BES とそれに対応する代替 BES に同じ値を指定してください。
pd_crypto_service_port		同一サーバマシン内複数ユニット構成の場合（相互系切り替え構成を含みます）にこのオペランドを指定するときは注意が必要です。同一サーバマシン内複数ユニット構成の場合（相互系切り替え構成を含みます），このオペランドはユニット制御情報定義でユニットごとに別々のポート番号を指定してください。 次に示す指定をした場合は，どちらかのユニットへの系の切り替えに失敗します。 • システム共通定義の pd_crypto_service_port オペランドを指定した（ユニット制御情報定義の pd_crypto_service_port オペランドは指定していない） • ユニット制御情報定義の pd_crypto_service_port オペランドに，ほかのユニット制御情報定義の pd_crypto_service_port オペランドと同じポート番号を指定した

注※1

システム定義の pd_name_port オペランド及び pdunit オペランドの-p オプションを指します。

注※2

システム定義の pd_scd_port オペランド及び pdunit オペランドの-s オプションを指します。

注※3

システム定義の pd_trn_port オペランド及び pdunit オペランドの-t オプションを指します。

(5)　切り替え先の指定（影響分散スタンバイレス型系切り替え機能限定）

影響分散スタンバイレス型系切り替え機能では，ほかの系切り替え機能とは異なり，切り替え先の定義方法が大きく異なります。

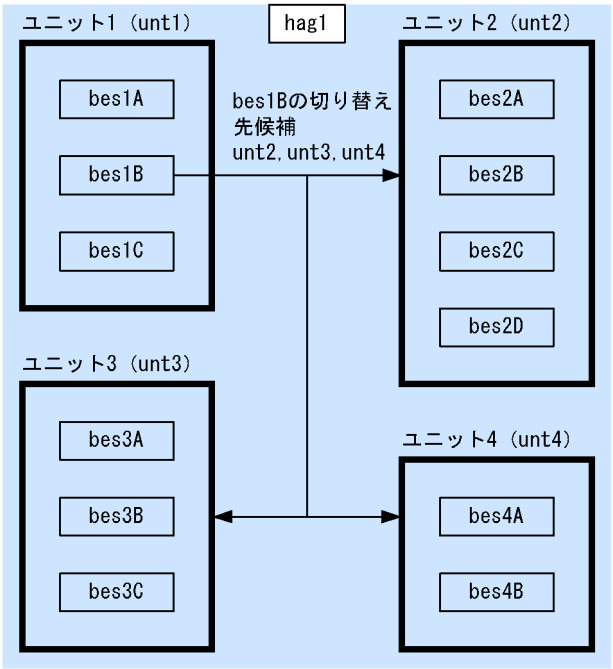
(a)　受け入れユニット

影響分散スタンバイレス型系切り替え機能では，切り替え単位がサーバのためサーバごとに切り替え先を指定する必要があります。サーバは複数の受け入れユニットを指定できます。複数の受け入れユニットは HA グループとして定義します。各サーバの切り替え先には HA グループを指定します。

また，影響分散スタンバイレス型系切り替え機能では，ユニットごとに同時に稼働できるゲスト BES 数の最大値（pd_ha_max_act_guest_servers）を指定できます。

HA グループの構成例を次の図に示します。

図 26–36　HA グループの構成例（その 1）

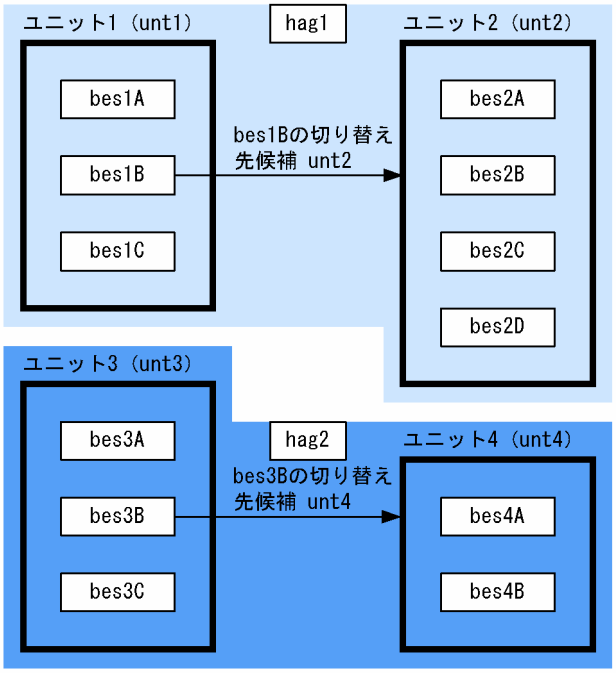



```
pdhagroup -g hag1 -u unt1,unt2,unt3,unt4

pdstart -t BES -s bes1A -u unt1 -g hag1
```

```
pdstart -t BES -s bes1B -u unt1 -g hag1
pdstart -t BES -s bes1C -u unt1 -g hag1
pdstart -t BES -s bes2A -u unt2 -g hag1
pdstart -t BES -s bes2B -u unt2 -g hag1
pdstart -t BES -s bes2C -u unt2 -g hag1
pdstart -t BES -s bes2D -u unt2 -g hag1
pdstart -t BES -s bes3A -u unt3 -g hag1
pdstart -t BES -s bes3B -u unt3 -g hag1
pdstart -t BES -s bes3C -u unt3 -g hag1
pdstart -t BES -s bes4A -u unt4 -g hag1
pdstart -t BES -s bes4B -u unt4 -g hag1
```

図 26–37　HA グループの構成例（その 2）




```
pdhagroup  -g hag1 -u unt1,unt2
pdhagroup  -g hag2 -u unt3,unt4

pdstart -t BES -s bes1A -u unt1 -g hag1
pdstart -t BES -s bes1B -u unt1 -g hag1
pdstart -t BES -s bes1C -u unt1 -g hag1
pdstart -t BES -s bes2A -u unt2 -g hag1
pdstart -t BES -s bes2B -u unt2 -g hag1
pdstart -t BES -s bes2C -u unt2 -g hag1
pdstart -t BES -s bes2D -u unt2 -g hag1
pdstart -t BES -s bes3A -u unt3 -g hag2
pdstart -t BES -s bes3B -u unt3 -g hag2
pdstart -t BES -s bes3C -u unt3 -g hag2
pdstart -t BES -s bes4A -u unt4 -g hag2
pdstart -t BES -s bes4B -u unt4 -g hag2
```

(b)　HA グループ定義

HiRDB システム定義の pdhagroup の-g オプションに HA グループ名を指定し，-u オプションに HA グループを構成するユニットのユニット識別子を指定します。

（例）

```
pdhagroup  -g hag1 -u unt1,unt2      …1.
pdhagroup  -g hag2 -u unt3,unt4      …2.
```

1. unt1 及び unt2 で構成する HA グループ hag1 を定義します。
2. unt3 及び unt4 で構成する HA グループ hag2 を定義します。

HA グループには次の制限があります。

- HA グループ内のユニット数は 32 までです。
- HA グループ内のユニットはすべて同一のネットワークセグメント内に配置してください。
- HA グループ内のユニットに定義できるホスト BES 数とゲスト BES 領域数（＝最大アクティブゲスト BES 数）の合計は 34 までです。

HA グループを構成するユニットは次の条件をすべて満たさなければなりません。

1. ホスト BES がないユニット（受け入れ専用ユニット）は，HA グループに属せません。
   このため，HA グループに属するユニットには，一つ以上のホスト BES がなければなりません。
2. HA グループに属するユニットを構成するサーバはバックエンドサーバだけです。このため，サーバ種別が「BES」以外のサーバはあってはなりません。
3. HA グループに属するユニットには影響分散スタンバイレス型系切り替え以外の系切り替えを適用できません。このため，HA グループに属するユニットでは，pd_ha_agent に「activeunits」以外の値は設定できません。
4. HA グループに属するユニットは，複数の HA グループに属することはできません。

(c)　受け入れユニットの指定方法

HiRDB のシステム定義で pdstart の-g オプションに受け入れユニットの属する HA グループを指定します。

影響分散スタンバイレス型系切り替えを適用したユニットに属するすべてのサーバに対して-g オプションを指定しなければなりません。

（例）pdstart -t BES -s bes1A -u unt1 -g hag1
unt1，又は bes1 異常終了時，bes1 の処理は HA グループ hag1 に属するユニットが受け入れます。

-g オプションを指定する場合の注意事項を次に示します。

1. 正規ユニット及び受け入れユニットを構成するサーバはバックエンドサーバだけです。
   - -t オプションに「BES」が指定されていなければなりません。
   - -g オプションで指定した HA グループに属するユニットにも，サーバ種別が「BES」以外のサーバがあってはなりません。
2. 正規ユニットを構成するサーバの数と受け入れユニットを構成するサーバの数が同じ必要はありません。

- -u オプションで指定したユニット（正規ユニット）のサーバの数と-g オプションで指定した HA グループに属するユニット（受け入れユニット）のサーバの数が同じ必要はありません。

(d)　同時稼働ゲスト BES 数最大値の指定

ユニット制御情報定義の pd_ha_max_act_guest_servers オペランドに，ユニット内で同時に実行系として稼働できるゲスト BES 数の最大値を指定します。この指定によって，ゲスト BES 用のリソース所要量を削減できます。また，過剰な負荷の上昇を抑えられます。

（例） pd_ha_max_act_guest_servers = 2

pd_ha_max_act_guest_servers オペランドの上限値は，HA グループ内のサーバ数から自ユニット内サーバ数を引いた値です。この上限値以上の値を指定しても，pd_ha_max_act_guest_servers オペランドには上限値が設定されます。ホスト BES の数と pd_ha_max_act_guest_servers オペランドの値の合計は 34 に制限されます。

ユニット内の受け入れ可能状態ゲスト BES の数は制限しません。ただし，ユニット内で実行系として稼働しているゲスト BES 数が pd_ha_max_act_guest_servers オペランドに指定した値に達すると，すべての非稼働中のゲスト BES について受け入れ可能状態を解除します。

また，HA グループ内の障害 BES 数が HA グループ内の稼働中ユニットの空きゲスト用領域数の合計を超えると，障害によって一部のサーバの処理が停止します。

(6)　系切り替え後のサーバプロセスの割り当て

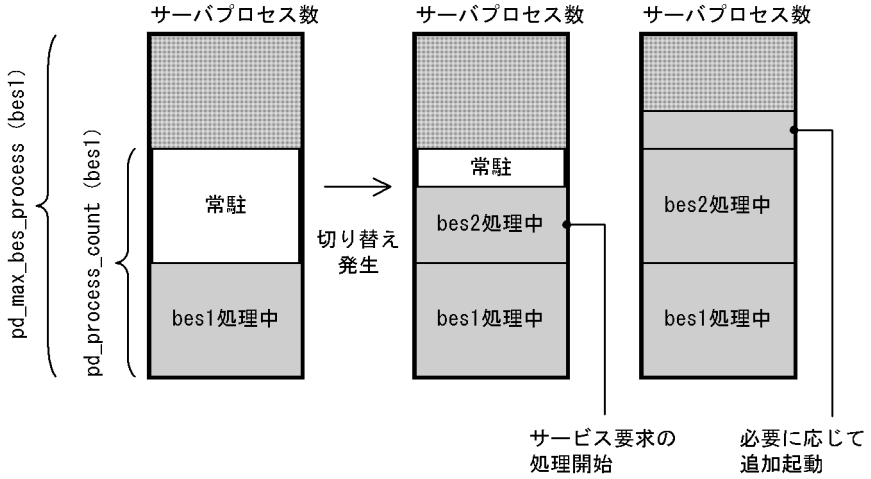
(a)　1：1 スタンバイレス型系切り替えの場合

1：1 スタンバイレス型系切り替えでの系切り替え後の代替 BES ユニットでは，代替 BES が本来の処理をするとともに，正規 BES の処理を代行します。このとき，サーバプロセスは代替 BES 本来の処理用と正規 BES の処理用に割り当てられます。代替 BES 本来の処理をしているサーバプロセスと正規 BES の処理を代行しているサーバプロセスの数は必要に応じて変化しますが，代替 BES の最大起動プロセス数（pd_max_bes_process オペランドの値）が両者の合計プロセス数の上限となります。この結果，系切り替え発生後の代替 BES ユニットでの過剰な負荷上昇を抑止できますが，一方で，系切り替え発生後には同時に処理できるサービス要求数の上限が平均で半数に制限されますので注意が必要です。系切り替え発生後のユニットの負荷上昇，及び同時に処理できるサービス要求数の両方を考慮して代替 BES の pd_max_bes_process オペランドを設定してください。

また，系切り替え発生前の状態で常駐プロセス数（pd_process_count オペランドの値）に余裕があり，サービス要求処理中でないサーバプロセスが常駐している状態では，系切り替え発生後にサービス要求処理中でない常駐プロセスを正規 BES の処理を代行するために利用できます。このため，切り替え時の処理性能が向上します。

1：1 スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセスの割り当て（その 1）を次の図に示します。

図 26–38　1：1 スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセスの割り当て（その 1）



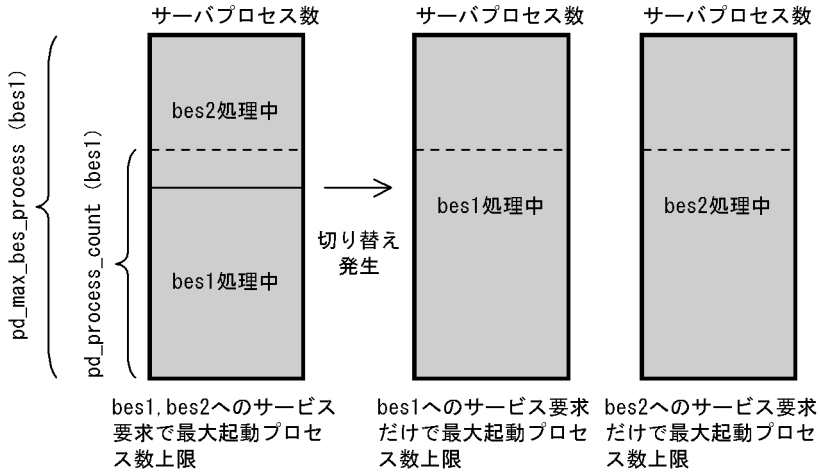
系切り替え発生前は代替 BES（bes1）の pd_max_bes_process オペランドの値まで同時に処理できます。また，代替 BES（bes1）の pd_process_count オペランドの値までサーバプロセスを常駐します。

系切り替えが発生すると代替 BES（bes1）の常駐プロセスを使って，正規 BES（bes2）の処理を開始します。このため，正規 BES（bes2）処理用のサーバプロセス起動が必要なく，切り替え後すぐに正規 BES（bes2）の処理が開始します。また，系切り替え前に正規 BES（bes2）処理用のサーバプロセスを待機起動する必要もありません。

また，常駐プロセスを使い切った場合は必要に応じてサーバプロセスを追加起動しますが，サーバプロセス数は代替 BES（bes1）の pd_max_bes_process オペランドの値までに制限されます。

1：1 スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセスの割り当て（その 2）を次の図に示します。

図 26–39　1：1 スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセスの割り当て（その 2）



系切り替え後，代替 BES（bes1）が正規 BES（bes2）の処理を代行中には，代替 BES の pd_max_bes_process オペランドの範囲で，必要に応じて起動したサーバプロセスを代替 BES（bes1）の処理，及び正規 BES（bes2）処理用に振り分けます。

代替 BES（bes1）への処理要求だけの場合には，代替 BES（bes1）の pd_max_bes_process オペランドの値まで，同時に代替 BES（bes1）の処理ができます。

また，正規 BES（bes2）への処理要求だけの場合には，代替 BES（bes1）の pd_max_bes_process オペランドの値まで，同時に正規 BES（bes2）の処理ができます。

(b)　影響分散スタンバイレス型系切り替え機能の場合

影響分散スタンバイレス型系切り替えでの系切り替え後の受け入れユニットでは，実行中ゲストサーバ数が pd_ha_max_act_guest_servers オペランドの値になるまでゲストサーバを受け入れできます。

受け入れユニットではホスト BES，ゲスト BES がそれぞれ独自の最大起動プロセス数（pd_max_bes_process オペランドの値）の範囲でサーバプロセスを起動しますが，そのほかにユニット内のサーバプロセス数合計値が pd_ha_max_server_process オペランドに制限されます。この結果，受け入れユニットの過剰な負荷上昇を抑止できます。ただし，系切り替え発生後に同時に処理できるサービス要求数の上限が制限されることがありますので注意が必要です。系切り替え発生後のユニットの負荷上昇，及び同時に処理できるサービス要求数の両方を考慮して pd_ha_max_server_process オペランドを設定してください。

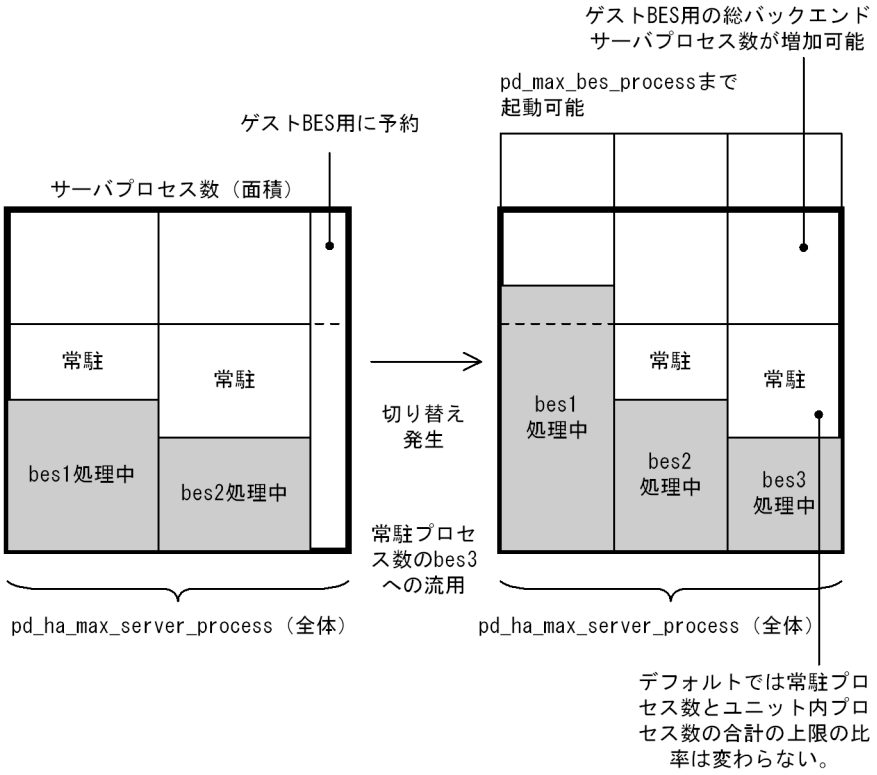
また，系切り替え発生前の状態でホスト BES の常駐プロセス数（pd_process_count オペランドの値）に余裕があり，サービス要求処理中でないサーバプロセスが常駐している状態では，系切り替え発生後にサービス要求処理中でない常駐プロセスをゲスト BES の処理に利用できるため，処理性能が向上します。一方で，常駐プロセス数を必要以上に大きくすると，非サービス要求処理中プロセスによってサーバプロセス数が pd_ha_max_server_process オペランドの値に達し，ほかのサーバで起動済みサーバプロセス数が pd_max_bes_process オペランドの値に満たない状態でも追加のサービス要求処理ができないことがあります。一般に，ユニット内の常駐プロセス数の合計と最大実行プロセス数の合計の比率はゲストサーバ受け入れ前後で同じにすることを推奨します。この目的で，ゲストサーバを受け入れた後のユニット内の常駐プロセス数の合計を pd_ha_process_count オペランドで制限します。実際の常駐プロセス数は，pd_ha_process_count オペランドをユニットで実行中サーバの pd_process_count オペランドで比例配分した数と pd_process_count オペランドの小さい方となります。

プロセス数に関連するオペランドの意味を次に示します。

- pd_ha_max_act_guest_servers：受け入れ可能なゲスト BES 数
- pd_ha_max_server_process：ゲスト BES とホスト BES のサーバプロセス数の合計値の上限
- pd_ha_process_count：ゲスト BES とホスト BES の常駐プロセス数の合計値の上限

影響分散スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセス割り当て（その 1）を次の図に示します。

図 26–40　影響分散スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセス割り当て（その 1）



系切り替え発生前はホスト BES（bes1,bes2）はそれぞれの pd_max_bes_process オペランドの値まで同時に処理できます。また，それぞれの pd_process_count オペランドの値までサーバプロセスを常駐します。

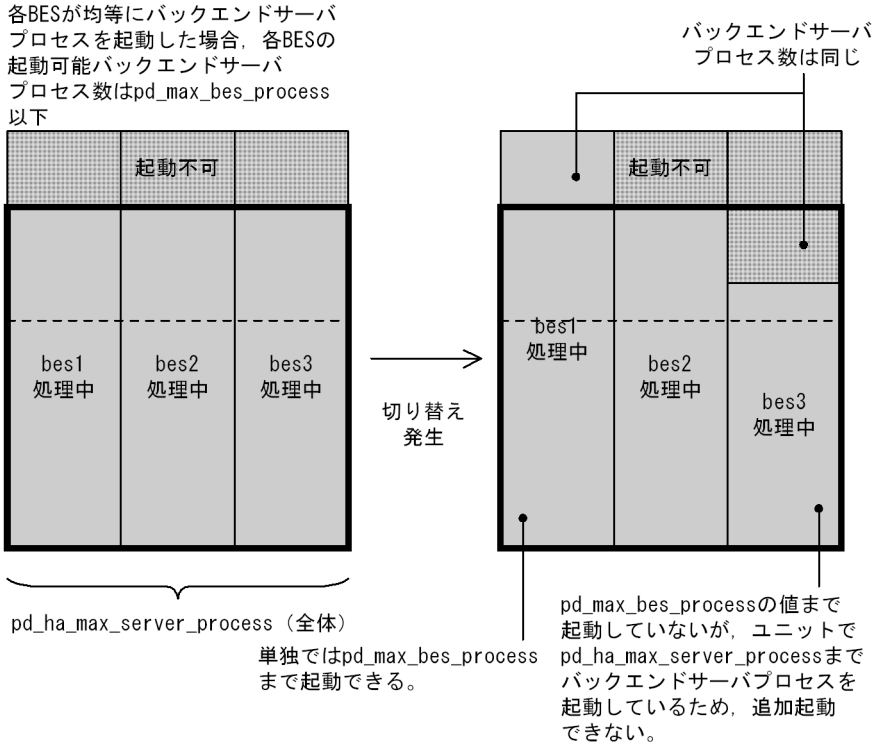
系切り替えが発生するとホスト BES（bes1,bes2）の常駐プロセスを流用してゲスト BES（bes3）のサーバプロセスを用意します。このため，ゲスト BES のサーバプロセスを起動する必要がなく，切り替え後すぐにゲスト BES（bes3）の処理が開始できます。また，系切り替え発生前にゲスト BES（bes3）用のサーバプロセスを待機起動する必要もありません。

各サーバは必要に応じてそれぞれの pd_max_bes_process オペランドの値までバックエンドサーバプロセスを起動しますが，ユニット内のサーバプロセスの総計は pd_ha_max_server_process オペランドの値に制限されます。

また，常駐プロセス数の総数がユニット内で pd_ha_process_count オペランドの値となるよう，各サーバの常駐プロセスが調整されます。調整後の各サーバの常駐プロセス数は各サーバの pd_process_count オペランドの比率を保つよう，pd_ha_process_count オペランドの値を比例配分します。

影響分散スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセス割り当て（その 2）を次の図に示します。

図 26–41　影響分散スタンバイレス型系切り替えでの系切り替え発生後のサーバプロセス割り当て（その 2）



系切り替え後，ゲスト BES（bes3）受け入れ中は，ユニット内のバックエンドサーバプロセス数が pd_ha_max_server_process オペランドの範囲で，必要に応じてホスト BES（bes1,bes2），及びゲスト BES（bes3）のバックエンドサーバプロセスを起動します。

特定のホスト BES（例えば bes1）への処理要求が特に多い場合は，該当ホスト BES（bes1）の pd_max_bes_process オペランドの値まで同時に処理できます。ただし，その分，他サーバ（例えば bes3）の同時要求処理数が少なくなります。

26.5.4　クライアント環境定義の指定

スタンバイ型系切り替え機能の場合はクライアント環境定義の指定に注意が必要です。

(1)　IP アドレスを引き継ぐ場合

- クライアント環境定義の PDHOST オペランドに仮想ネットワーク名を指定します。HiRDB/パラレルサーバの場合は，システムマネジャのユニットの仮想ネットワーク名を指定します。
- マルチフロントエンドサーバの場合は，PDFESHOST オペランドにフロントエンドサーバのユニットの仮想ネットワーク名を指定します。
- DNS サーバを使用しない場合は，クライアントマシンの hosts ファイルに仮想ネットワーク名と IP アドレスを登録します。

(2)　IP アドレスを引き継がない場合

- クライアント環境定義の PDHOST オペランドに，現用系及び予備系の両方のホスト名を指定します。HiRDB/パラレルサーバの場合はシステムマネジャのユニットのホスト名を指定します。
- PDHOST オペランドに現用系及び予備系のホスト名を指定した場合，実行系が現用系から予備系に切り替わると，UAP は一度現用系（待機系）に接続を試みて失敗した後に予備系（実行系）に接続するため，UAP の接続処理時間がその分だけ長くなります。この問題を解決するには，次に示すどちらかの方法で対処してください。
  - クライアント接続用のホスト名をエイリアス IP アドレスで設定し，その IP アドレスを引き継ぐようにしてください。この場合のホスト名の指定方法については，次のどちらかを参照してください。

    HiRDB/シングルサーバの場合

    「IP アドレスを引き継がない場合（クライアント接続用の IP アドレスだけを引き継ぐ場合）」

    HiRDB/パラレルサーバの場合

    「IP アドレスを引き継がない場合（クライアント接続用の IP アドレスだけを引き継ぐ場合）」
  - 次のクライアント環境定義を指定して，接続先にするホストの切り替え時間を短くしてください。各クライアント環境定義の詳細は，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

    ブロックモードで接続する場合

    ・PDTIMEDOUTRETRY

    ・PDCONREFRCOUNT

    ・PDCONREFRINTERVAL

    ノンブロックモードで接続する場合

    ・PDNBLOCKWAITTIME

    ・PDCONREFRCOUNT

    ・PDCONREFRINTERVAL
- マルチフロントエンドサーバの場合は，PDFESHOST オペランドに現用系及び予備系の両方のホスト名（フロントエンドサーバのユニットのホスト名）を指定します。
- DNS サーバを使用しない場合は，クライアントマシンの hosts ファイルにホスト名と IP アドレスを登録します。

26.5.5　HiRDB システム定義及びクライアント環境定義のホスト名の指定例

ここでは，スタンバイ型系切り替え機能を使用するときのホスト名の指定例を説明します。

(1)　IP アドレスを引き継ぐ場合

HiRDB システム定義及びクライアント環境定義の指定例（ホスト名の指定例）を次に示します。

●システム共通定義

```
pdunit -x HIRDB01 -u UNT1 -d C:¥hirdb_x          1
```

●ユニット制御情報定義

```
set pd_hostname = mainhost                       2
```

●クライアント環境定義

```
export PDHOST = HIRDB01                          3
```

〔説明〕

1. pdunit オペランドの-x オプションには仮想ネットワーク名を指定します。
2. pd_hostname オペランドには現用系のホスト名を指定します。
3. クライアント環境定義の PDHOST オペランドに，仮想ネットワーク名（HiRDB/パラレルサーバの場合はシステムマネジャのホストの仮想ネットワーク名）を指定します。

   なお，マルチフロントエンドサーバの場合は，PDFESHOST オペランドに仮想ネットワーク名（フロントエンドサーバのホストの仮想ネットワーク名）を指定します。

(2) IP アドレスを引き継がない場合

HiRDB システム定義及びクライアント環境定義の指定例（ホスト名の指定例）を次に示します。

●システム共通定義

```
pdunit -x mainhost -u UNT1 -d C:¥hirdb¥pddir_s -c reservedhost    1
```

●ユニット制御情報定義

```
set pd_hostname = mainhost                                        2
```

●クライアント環境定義

```
export PDHOST = mainhost,reservedhost                             3
```

〔説明〕

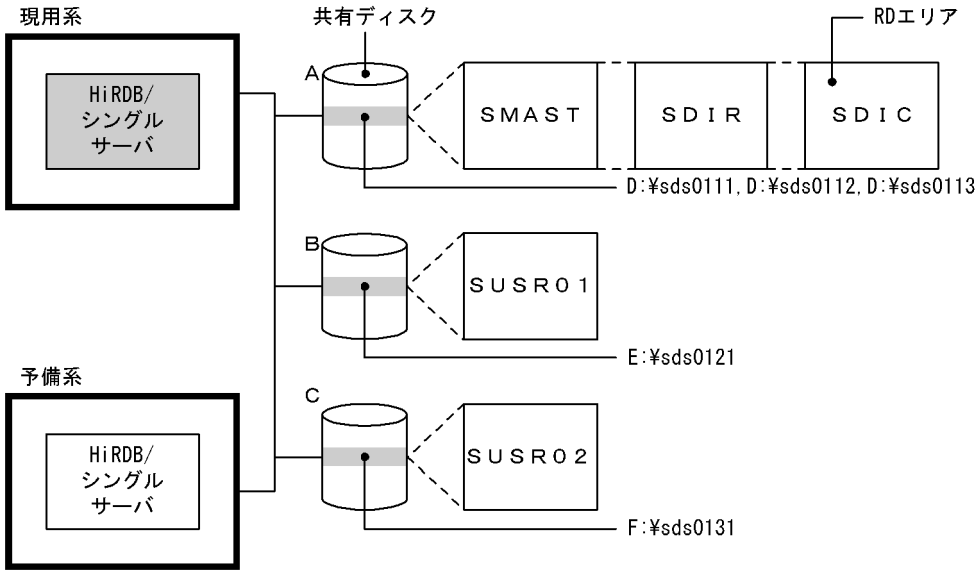
1. pdunit オペランドの-x オプションには現用系のホスト名を指定します。-c オプションには予備系のホスト名を指定します。
2. pd_hostname オペランドには現用系のホスト名を指定します。
3. クライアント環境定義の PDHOST オペランドには，現用系と予備系の両方のホスト名（HiRDB/パラレルサーバの場合はシステムマネジャのホスト名）を指定します。

   なお，マルチフロントエンドサーバの場合は，PDFESHOST オペランドに現用系と予備系の両方のホスト名（フロントエンドサーバのホスト名）を指定します。

26.5.6　RD エリアの作成

共有ディスクに作成した RD エリア用の HiRDB ファイルシステム領域に RD エリアを定義します。ユーザ用 RD エリアとシステム用 RD エリアをそれぞれ異なる共有ディスクの HiRDB ファイルシステム領域に作成するときの定義例を次の図に示すシステム構成例を基に説明します。影響分散スタンバイレス型系切り替えの場合は，サーバごとに異なる共有ディスクに作成した HiRDB ファイルシステム領域に RD エリアを定義します。

図 26–42　HiRDB/シングルサーバのシステム構成例



●create rdarea 文の指定例

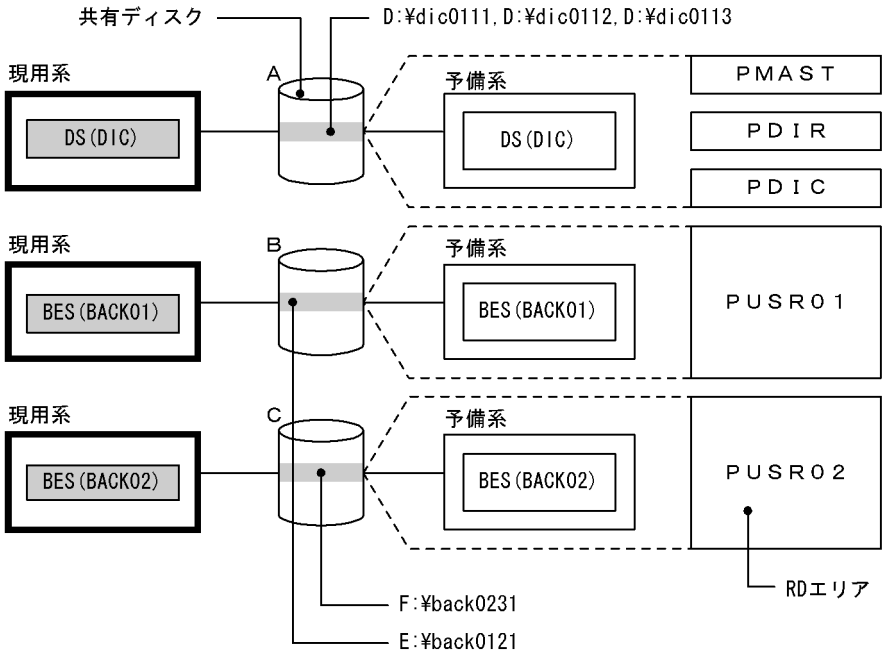
```
create rdarea SMAST for masterdirectory                  1
  file name ”D:¥sds0111¥srd01” initial 10 segments;
create rdarea SDIR for datadirectory                     2
  file name ”D:¥sds0112¥srd02” initial 5 segments;
create rdarea SDIC for datadictionary                    3
  file name ”D:¥sds0113¥srd03” initial 20 segments;
create rdarea SUSR01 for user used by PUBLIC             4
  file name ”E:¥sds0121¥srd04” initial 500 segments;
create rdarea SUSR02 for user used by PUBLIC             5
  file name ”F:¥sds0131¥srd05” initial 500 segments;
```

〔説明〕

1. マスタディレクトリ用 RD エリア（SMAST）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
2. データディレクトリ用 RD エリア（SDIR）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
3. データディクショナリ用 RD エリア（SDIC）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
4. ユーザ用 RD エリア（SUSR01）を共有ディスク B の HiRDB ファイルシステム領域に作成します。

5. ユーザ用 RD エリア（SUSR02）を共有ディスク C の HiRDB ファイルシステム領域に作成します。

図 26–43　HiRDB/パラレルサーバのシステム構成例



●create rdarea 文の指定例

```
create rdarea PMAST for masterdirectory                  1
    server name DIC file name "D:¥dic0111¥prd01"
    initial 10 segments;
create rdarea PDIR for datadirectory                     2
    server name DIC file name "D:¥dic0112¥prd02"
    initial 5 segments;
create rdarea PDIC for datadictionary                    3
    server name DIC file name "D:¥dic0113¥prd03"
    initial 20 segments;
create rdarea PUSR01 for user used by PUBLIC             4
    server name BACK01 file name "E:¥back0121¥prd04"
    initial 500 segments;
create rdarea PUSR02 for user used by PUBLIC             5
    server name BACK02 file name "F:¥back0231¥prd05"
    initial 500 segments;
```

〔説明〕

1. マスタディレクトリ用 RD エリア（PMAST）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
2. データディレクトリ用 RD エリア（PDIR）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
3. データディクショナリ用 RD エリア（PDIC）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
4. ユーザ用 RD エリア（PUSR01）を共有ディスク B の HiRDB ファイルシステム領域に作成します。
5. ユーザ用 RD エリア（PUSR02）を共有ディスク C の HiRDB ファイルシステム領域に作成します。

26.5.7　グローバルバッファの定義（1：1 スタンバイレス型系切り替え機能限定）

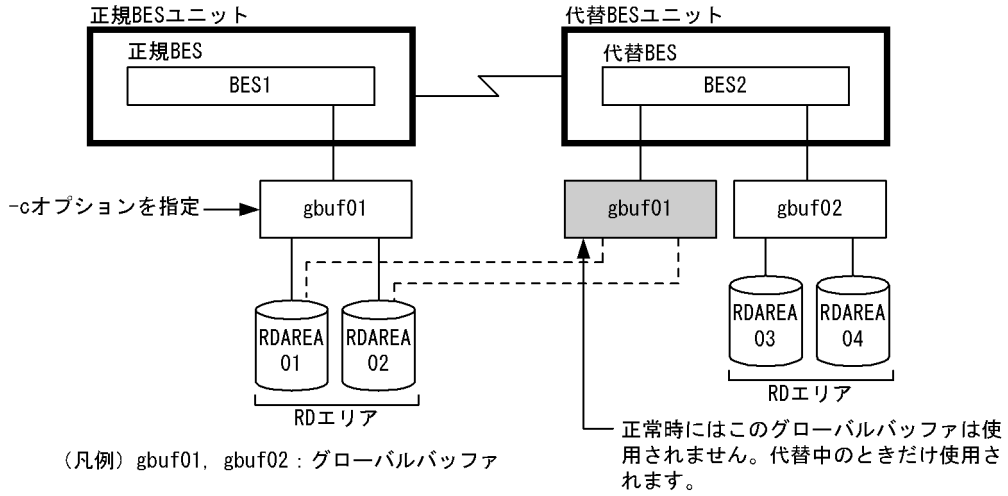
正規 BES 下の RD エリアが使用するグローバルバッファを定義するとき，pdbuffer オペランドに-c オプションを指定してください。-c オプションを指定すると，代替中に代替部が使用するグローバルバッファを確保できます。

なお，pdbuffer オペランドの-c 及び-o オプションの両方を省略した場合は，代替 BES ユニットを開始できません。

(1)　データ用グローバルバッファ（-r オプションのグローバルバッファ）の場合

代替中のデータ用グローバルバッファの割り当て方式を説明します。次に示すシステム構成を例にします。

●システム構成例 1

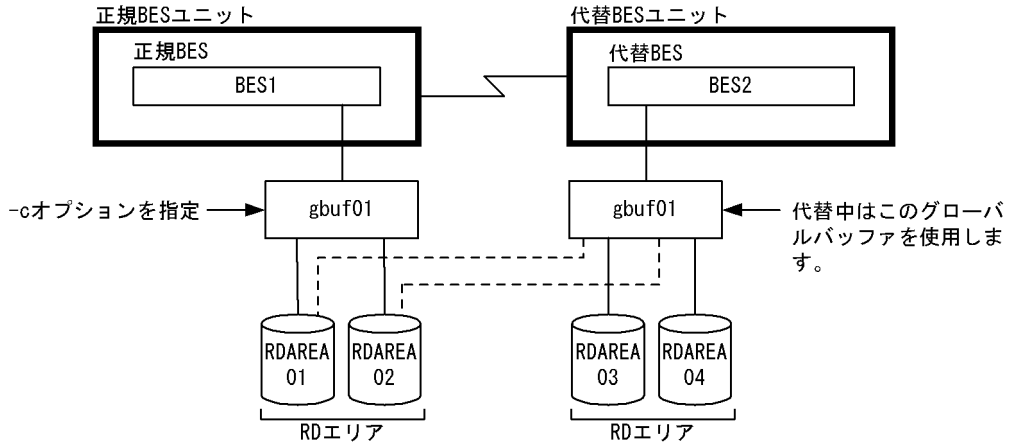



```
pdbuffer -a gbuf01 -r RDAREA01,RDAREA02 -n 1000 -c
pdbuffer -a gbuf02 -r RDAREA03,RDAREA04 -n 1000
```

〔説明〕

- 正規 BES と代替 BES で同じ名称のグローバルバッファを使用していない構成です。
- -c オプションを指定すると代替 BES 下にも gbuf01 が確保されます。代替中はこのグローバルバッファを使用します。ただし，正常時にはこのグローバルバッファは使用されません。
- 代替 BES ユニットでグローバルバッファが使用する共用メモリを見積もるときに gbuf01 の分を加算する必要があります。
- 代替中もバッファヒット率が低下しません。

●システム構成例 2



（凡例）gbuf01：グローバルバッファ

```
pdbuffer -a gbuf01 -r RDAREA01,RDAREA02,RDAREA03,RDAREA04 -n 1000 -c
```

〔説明〕

- 正規 BES と代替 BES で同じ名称のグローバルバッファを使用している構成です。
- -c オプションを指定すると代替中は代替 BES の gbuf01 を使用します。
- 代替中はバッファヒット率が低下することがあります。
- 代替 BES 下の gbuf01 のバッファサイズは，正規 BES 及び代替 BES 下の RD エリアの最大ページ長になります。

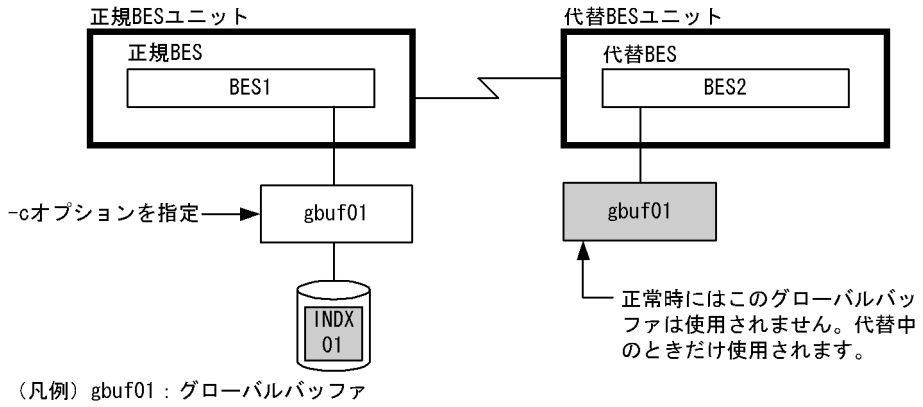
-c オプションを省略した場合

-c オプションを省略すると，代替中は-o オプションのグローバルバッファを使用します。

(2) インデクス用グローバルバッファ（-i オプションのグローバルバッファ）の場合

代替中のインデクス用グローバルバッファの割り当て方式を説明します。次に示すシステム構成を例にします。

●システム構成例

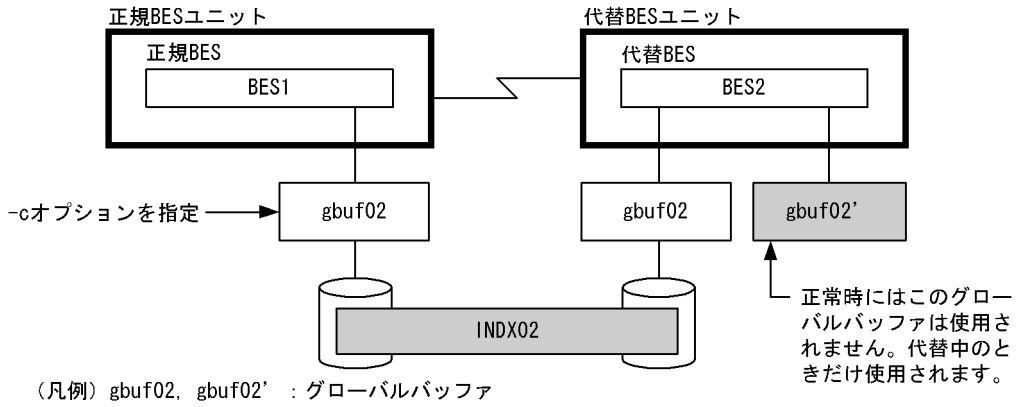



```
pdbuffer -a gbuf01 -i USER01.INDX01 -n 1000 -c
```

〔説明〕

- 非横分割インデクス INDX01 が使用するグローバルバッファに-c オプションを指定すると，代替 BES 下にも gbuf01 が確保されます。代替中はこのグローバルバッファを使用します。ただし，正常時にはこのグローバルバッファは使用されません。
- 代替 BES ユニットでグローバルバッファが使用する共用メモリを見積もるときに gbuf01 の分を加算する必要があります。

●システム構成例




```
pdbuffer -a gbuf02 -i USER01.INDX02 -n 1000 -c
```

〔説明〕

- 横分割インデクス INDX02 が使用するグローバルバッファに-c オプションを指定すると，代替 BES 下に gbuf02 と同じグローバルバッファ gbuf02 が確保されます。代替中はこのグローバルバッファを使用するため，BES2 下の INDX02 とアクセス競合は発生しません。ただし，正常時にはこのグローバルバッファは使用されません。
- 代替 BES ユニットでグローバルバッファが使用する共用メモリを見積もるときに gbuf02 の分を 2 倍する必要があります。

-c オプションを省略した場合

-c オプションを省略すると，代替中は次に示す優先順位でグローバルバッファを使用します。

1. インデクス格納 RD エリアにグローバルバッファを割り当てていて，そのグローバルバッファに-c オプションを指定している場合は，(1)で説明した方式に従います。
2. -o オプションのグローバルバッファを使用します。

(3) LOB 用グローバルバッファ（-b オプションのグローバルバッファ）の場合

代替中の LOB 用グローバルバッファの割り当て方式はデータ用グローバルバッファと同じです。ただし，-c オプションを省略した場合は，グローバルバッファを使用しないで直接 RD エリアに対して読み込み及び書き込みをします。

(4) -o オプションのグローバルバッファの場合

代替 BES 下の-o オプションのグローバルバッファを代替中は正規 BES 下の RD エリアも使用します。グローバルバッファのバッファサイズは正規 BES 及び代替 BES 下の RD エリアの最大ページ長になります。

(5) グローバルバッファの設計方針

基本的には，正規 BES 下の RD エリア，インデクス，及び LOB 用 RD エリアが使用するグローバルバッファに-c オプションを指定してください。

- -c オプションを指定すると，代替中に使用するグローバルバッファを代替 BES 下に作成します。このため，代替 BES ユニットでグローバルバッファが使用する共用メモリがその分だけ余計に必要になります。共用メモリに余裕がない場合は-c オプションを指定しないでください。
- 代替 BES 下のグローバルバッファを正規 BES 下の RD エリアにも使用させる場合，そのグローバルバッファのバッファサイズは正規 BES 及び代替 BES 下の RD エリアの最大ページ長になります。したがって，RD エリアのページ長も考慮して-c オプションを指定してください。

26.5.8 グローバルバッファの定義（影響分散スタンバイレス型系切り替え機能限定）

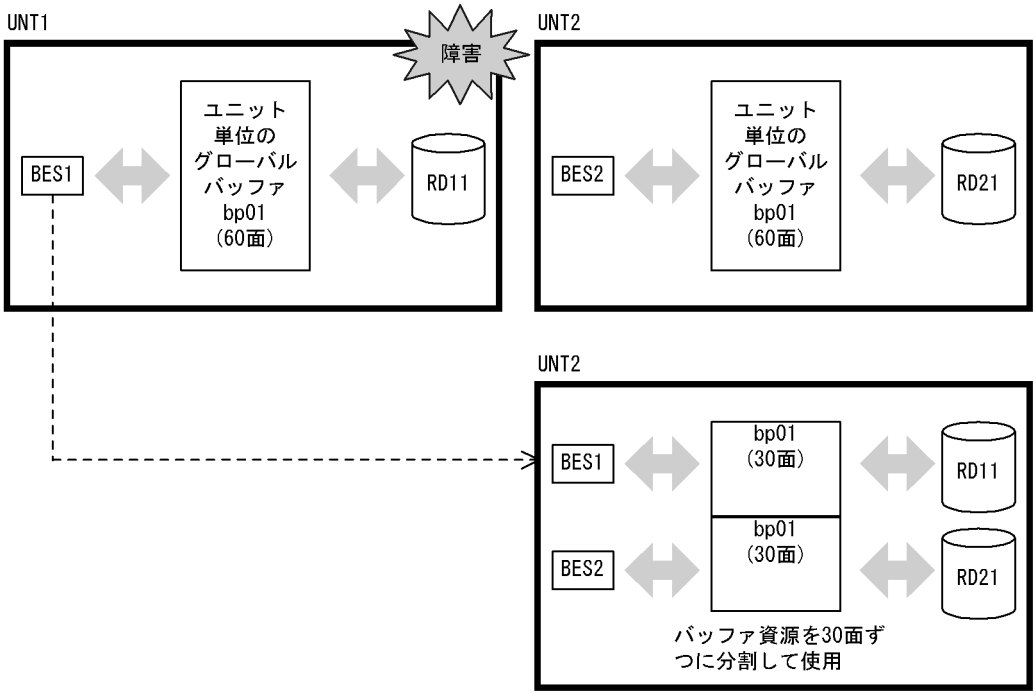
影響分散スタンバイレス型系切り替え機能を使用する場合，ユニット単位でグローバルバッファを割り当てられます。

影響分散スタンバイレス型系切り替え機能を適用するバックエンドサーバに配置されている，RD エリア，又はインデクスに対して，グローバルバッファを割り当てるには，pdbuffer オペランドの-c オプション（共用化オプション）を指定します。-c オプションを指定して割り当てたグローバルバッファを**ユニット単位のグローバルバッファ**といい，ユニット単位のグローバルバッファには次の特長があります。

- ユニット単位のグローバルバッファは HA グループを構成するユニットすべてに確保されます。

- 同じユニット単位のグローバルバッファを割り当てる RD エリア，又はインデクスが複数のバックエンドサーバにある場合には，そのグローバルバッファ資源を複数のバックエンドサーバで均等分割して使用します。一つのバックエンドサーバにある場合はそのバックエンドサーバで占有して使用します。-o オプションのグローバルバッファも同様に，割り当てる RD エリアが複数のバックエンドサーバにある場合はバックエンドサーバ間でグローバルバッファ資源を均等分割して使用します。例として，ユニット単位のグローバルバッファの共用イメージを次の図に示します。この図では，異なるユニットのサーバに同じユニット単位のグローバルバッファを割り当てています。

  図 26–44　ユニット単位のグローバルバッファの共用イメージ




```
pdbuffer -a bp01 -a bp01 -r RD11,RD12 -n 60 -c
```

(1)　ユニット単位のグローバルバッファ設計手順

最初に，共用化設計とするか非共用化設計とするかを選択します。系切り替えで縮退する場合のグローバルバッファ設計には次の考え方があります。

1. 共用化設計

   縮退時，受け入れユニットのグローバルバッファ資源を共用することでメモリを有効に利用します。これを共用化設計といいます。共用化設計には次の特徴があります。

   - メリット：縮退時には受け入れユニットのグローバルバッファ資源を共用利用するのでメモリ効率が良い
   - デメリット：縮退時，共用するサーバ数に対応してバッファヒット率は低下する

2. 非共用化設計

   縮退時にだけ使用するグローバルバッファ資源を受け入れユニットに確保しておき，切り替え時はそれを使用します。これを非共用化設計といいます。非共用化設計には次の特徴があります。

- メリット：縮退時も縮退前と同じバッファ資源数を使用できるのでヒット率を維持できる
- デメリット：縮退時用のグローバルバッファ資源を受け入れユニットすべてに確保するため，メモリ効率が悪い

影響分散スタンバイレス型系切り替え機能の目的はリソース共有，負荷分散にあるため，「1.　共用化設計」をお勧めします。メモリを効率的に利用できるからです。

非共用化設計の場合，受け入れユニットすべてにサーバ専用のグローバルバッファを確保します。そのため，HA グループ全体で，通常のグローバルバッファ用の共用メモリ見積もり値を，HA グループを構成するユニット数で乗算した分のグローバルバッファ用共用メモリ量が必要となります。見積もりを満足する共用メモリがあれば縮退時にも性能を維持できるため，「2.　非共用化設計」を選んでください。

(a)　共用化設計の手順

ここではグローバルバッファの共用を行う場合の設計手順について説明します。

1. 同じバッファプールを共用する RD エリアの決定
   - 同じ種類の RD エリア同士で共用する

     データ格納 RD エリア，インデクス格納 RD エリア，LOB 用 RD エリアなど，同じ種類の RD エリアを pdbuffer の-r オプションで指定し，共用するようにします。このとき同じページサイズの RD エリア，又は同じアクセス頻度の RD エリアで共用するとメモリ効率が良くなります。
   - 横分割した表，又はインデクスの RD エリアで共用する

     横分割した表格納 RD エリア，又は横分割したインデクスのインデクス格納 RD エリアを pdbuffer の-r オプションで指定して共用するようにします。横分割した表とインデクスが同じ RD エリアに格納されている場合は pdbuffer -i オプションを指定してインデクス専用バッファを割り当てます。

   また，共用を組む RD エリアのサーバ配置，及びユニット配置によって次の特長が得られます。特長を参考にして共用する RD エリアを選択してください。

   - 異なるユニットに配置されたサーバの RD エリアを共用した場合

     縮退前はグローバルバッファを占有して使用できるので縮退前の性能を重視した割り当てができます。ただし，縮退時のグローバルバッファ資源割り当てがユニット間で不均衡となります。
   - 同一ユニットに配置されたサーバの RD エリア，及び異なるユニットに配置されたサーバの RD エリアで共用した場合

     縮退時のバッファ割り当てをユニット間で均等とする設計ができます。
2. 共用するグローバルバッファのバッファ面数の決定

   -n オプションで指定するバッファ面数は，共用する HA グループのサーバ間で均等に分配して使用します。そのためバッファ面数が不足しないように共用するサーバ数に見合ったバッファ面数を設定する必要があります。次の式を目安にバッファ面数を決定してください。

1 サーバでの必要面数×（ホストBES数＋ゲストBES数）

なお，縮退時も縮退前のバッファ性能を必用とする RD エリアがある場合には，その RD エリアだけは「非共用化設計の手順」を参考にサーバ専用のグローバルバッファを割り当ててください。

(b)　非共用化設計の手順

ここではグローバルバッファの共用をしない場合の設計手順について説明します。

サーバ専用のグローバルバッファは一つの RD エリアに対して割り当てる方法と，同じサーバに属する複数の RD エリアに割り当てる方法があります。

- 一つの RD エリア専用に割り当てる場合

  pdbuffer の-r オプションに一つの RD エリアを指定します。ほかの RD エリアとの競合がないので最も性能を重視した割り当てができます。なお，pdbuffer の-i オプションに非分割インデクスを指定して，インデクス専用バッファを割り当てても同じ効果が得られます。

- 同じサーバに属する複数の RD エリアに割り当てる場合

  pdbuffer の-r オプションに同じサーバに属する複数の RD エリアを指定します。指定する RD エリアの種類はデータ格納 RD エリア，インデクス格納 RD エリア，LOB 用 RD エリアなど，同じ種類の RD エリアを指定してください。

(2)　RD エリア用，及び LOB 用グローバルバッファの割り当て（-r オプション又は-b オプション指定）

RD エリア用，及び LOB 用のユニット単位のグローバルバッファの割り当ては，指定する RD エリアの組み合わせによって，4 種類に分類されます。グローバルバッファの共用形態別の推奨条件（-r オプション又は-b オプション指定）を次の表に示します。

表 26–16　グローバルバッファの共用形態別の推奨条件（-r オプション又は-b オプション指定）

指定方法（-r 又は-b で指定した RD エリアの組み合わせ）			バッファ共用形態	メリット	推奨条件
異なるサーバの RD エリア	同じユニットの RD エリア	異なるユニットの RD エリア			
なし	なし	なし	非共用	他サーバと共用しないため，多重障害時でも定常時のバッファ性能を維持できる	すべての受け入れユニットにバッファを確保するためメモリに余裕がある環境で，縮退時も縮退前のバッファ性能を維持したい RD エリアに対する適用をお勧めします。
あり	あり	なし	ユニット内サーバ共用	多重障害でも定常時のバッファ性能を維持できる	非共用タイプ同様縮退時も縮退前のバッファ性能を維持できますが，初回切り替え時のメモリ効率が悪いので非共用タイプをお勧めします。
あり	なし	あり	ユニット間サーバ共用	• 現用系のバックエンドサーバでは指定	現用時の性能を重視し，縮退時にバッファ資源

指定方法（-r 又は-b で指定した RD エリアの組み合わせ）			バッファ共用形態	メリット	推奨条件
異なるサーバの RD エリア	同じユニットの RD エリア	異なるユニットの RD エリア			
				バッファ面数をすべて使用できる • 縮退時は受け入れユニットの資源を共用するのでメモリ効率が良い	の共用を実現したい場合の適用をお勧めします。
あり	あり	あり	ユニット内ユニット間サーバ共用	• 縮退時は受け入れユニットの資源を共用するのでメモリ効率が良い • 共用サーバ数を均等にすることで縮退時の負荷分散を均等に設計できる	縮退時，バッファ資源の共用，及び均等負荷分散を実現したい場合の適用をお勧めします。

次に示す-r，-b 指定のバッファの設計指針を参考に適切な共用形態を選択してください。影響分散スタンバイレス型系切り替え機能はリソース共有，負荷分散が目的でもあるため，ユニット間で共用する構成となるユニット間サーバ共用，又はユニット内ユニット間サーバ共用をお勧めします。縮退時に性能を低下させたくない場合は，非共用，又はユニット内サーバ共用としてください。

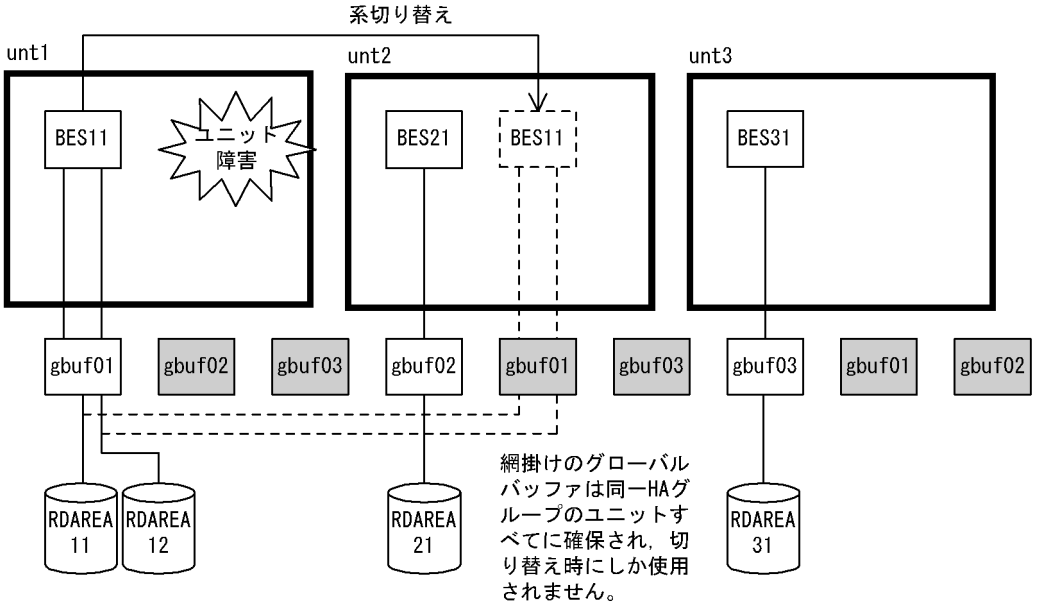
-r，-b 指定のバッファの設計指針

1. メモリに余裕のある環境で縮退時もバッファヒット率を低下させたくない RD エリアの場合：非共用，又はユニット内サーバ共用を選択します。
2. 現用時はバッファ資源を特定のサーバで独占して使用し，縮退時はほかのサーバと共用したい場合：ユニット間サーバ共用を選択します。
3. 1.及び 2.のどちらにも当てはまらない場合：ユニット内ユニット間サーバ共用を選択します。

(a)　非共用グローバルバッファの割り当て方法

一つのバックエンドサーバに属する RD エリアだけを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。システム構成例を次に示します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA12 -n 2000 -c
pdbuffer -a gbuf02 -r RDAREA21 -n 1000 -c
pdbuffer -a gbuf03 -r RDAREA31 -n 1000 -c
```

〔説明〕

同一サーバに属する RD エリアだけを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。

BES11，BES21，BES31 に非共用バッファ gbuf01，gbuf02，gbuf03 を割り当てます。

系切り替え時も BES11 専用の gbuf01 を使用するのでヒット率は低下しませんが，すべての受け入れユニットで切り替え時使用するためのバッファが確保されるのでメモリを多く消費します。

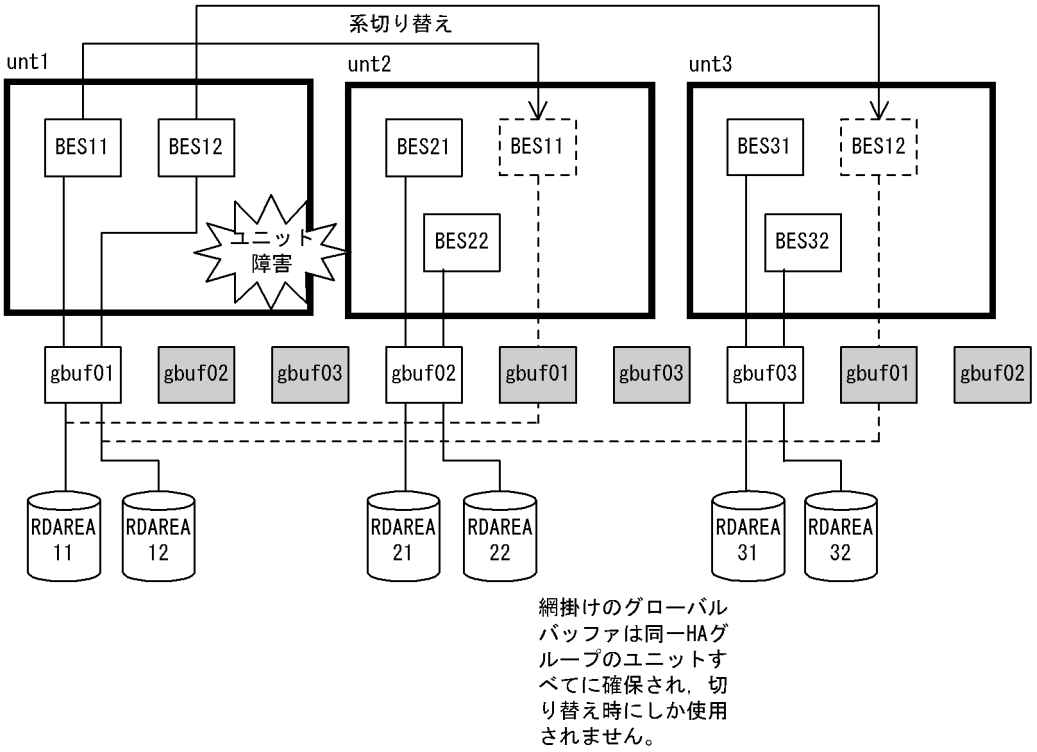
注

- 受け入れユニットすべてに同一グローバルバッファが作成されます。このグローバルバッファは代替されるまで使用されません。
- メモリ効率を向上するためには同じページサイズの RD エリアを指定するようにします。

(b)　ユニット内サーバ共用のグローバルバッファの割り当て方法

使用できる共用メモリが十分にある環境で，多点障害時でも定常時と同じバッファ性能を維持したい場合に指定します。同一ユニット内のバックエンドサーバに配置された RD エリアを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA12 -n 1000 -c
pdbuffer -a gbuf02 -r RDAREA21,RDAREA22 -n 1000 -c
pdbuffer -a gbuf03 -r RDAREA31,RDAREA32 -n 1000 -c
```

〔説明〕

同一ユニット内のサーバに配置された RD エリアを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。

ユニット内サーバ共用のグローバルバッファ gbuf01，gbuf02，gbuf03 を割り当てます。

系切り替え時も gbuf01 を使用するのでヒット率は低下しませんが，すべての受け入れユニットに切り替え時使用するためのグローバルバッファが確保されるのでメモリを多く消費します。

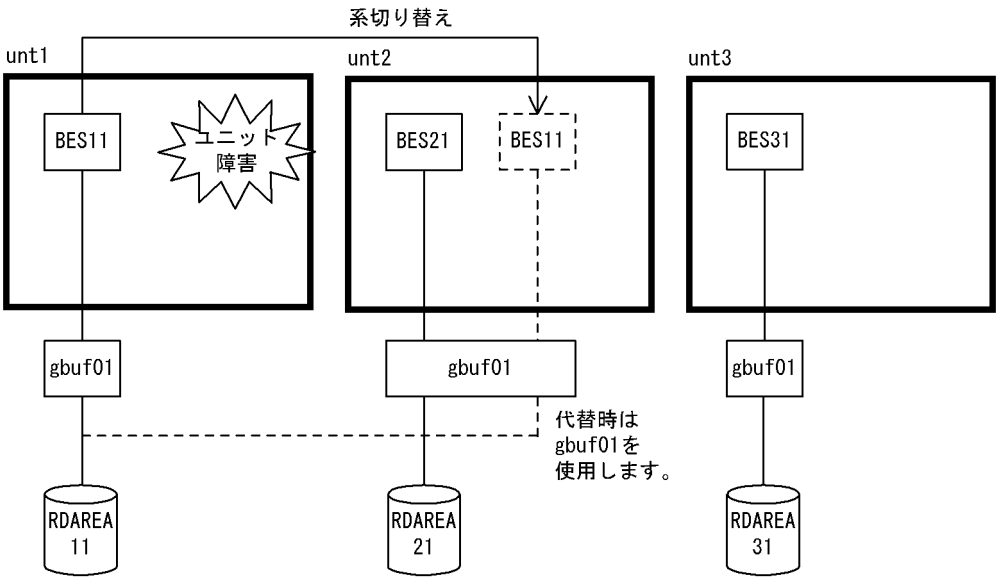
注

- 受け入れユニットすべてに同一グローバルバッファが作成されます。このグローバルバッファは代替されるまで使用されません。
- この指定のグローバルバッファは複数サーバ間で共用して使用されます。
- pdbuffer オペランドの-l オプション省略時のバッファサイズは，指定した RD エリアの内の最大ページサイズとなります。

(c) ユニット間サーバ共用のグローバルバッファの割り当て方法

同一ユニット内のサーバに配置された RD エリアを指定しないで，異なるユニットのサーバに配置された RD エリアを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA21,RDAREA31 -n 1000 -c
```

〔説明〕

異なるユニットに配置されたサーバの RD エリアを指定します。

同一ユニット内のサーバに配置された RD エリア間で共用化しないで，異なるユニットのサーバに配置された RD エリアを共用化します。

縮退前は gbuf01 のリソース（バッファ数）を BES11 で占有して使用できます。縮退時は BES21 と BES11 で共用するので使用できるリソース（バッファ数）は半分になります。

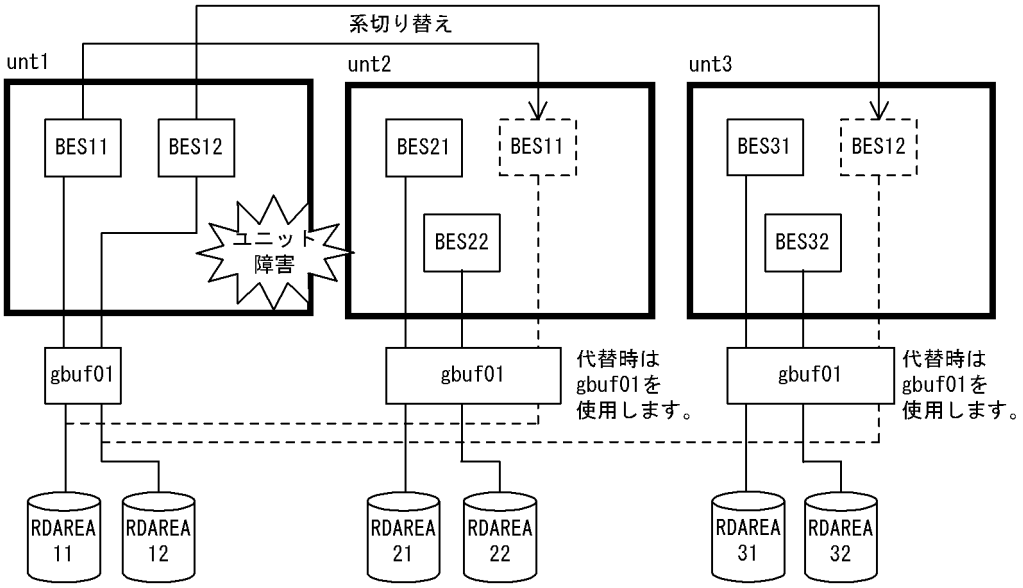
注

- 移動先が一つのユニットになるのでそのユニットのグローバルバッファだけ負荷が高くなります。ユニット間サーバ共用バッファを複数定義し各ユニットの負荷が均等になるように設計してください
- 受け入れユニットすべてに同一グローバルバッファが作成されます。
- この指定のグローバルバッファは現用時には一つのサーバで占有して使用され，縮退時には複数サーバ間で共用して使用されます。
- pdbuffer オペランドの-l オプション省略時のバッファサイズは指定した RD エリアの内の最大ページサイズとなります。

(d)　ユニット内ユニット間サーバ共用のグローバルバッファの割り当て方法

ユニット内ユニット間サーバ共用の RD エリアを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -r RDAREA11,RDAREA12,RDAREA21,RDAREA22,RDAREA31,RDAREA32 -n 1000 -c
```

〔説明〕

ユニット内ユニット間サーバ共用の RD エリアを pdbuffer オペランドの-r オプション，又は-b オプションに指定します。

ユニット内ユニット間サーバ共用の RD エリアでグローバルバッファを共用することで，縮退時の負荷分散が均等になるようなバッファを割り当てます。定常時は各ユニットが二つのバックエンドサーバ間で gbuf01 を共用するので一つのバックエンドサーバに割り当てられるバッファ面数は半分です。縮退時は三つのバックエンドサーバ間で gbuf01 を共用するので一つのバックエンドサーバに割り当てるバッファ面数は各ユニットが 1/3 と均等になります。

注

- 縮退時に受け持つサーバ数が各受け入れユニットで均等になるように設計してください。
- 受け入れユニットすべてに同一グローバルバッファが作成されます。
- この指定のグローバルバッファは複数サーバ間で共用して使用されます。
- pdbuffer オペランドの-l オプション省略時のバッファサイズは指定した RD エリアの内の最大ページサイズとなります。

(3) インデクス用グローバルバッファの割り当て方法（-i オプション指定）

特定のインデクスのインデクスページをバッファリングしたい場合にインデクス用グローバルバッファを割り当てます。インデクス格納 RD エリアにそのインデクスしか格納されていない場合には，その RD エリアに RD エリア用グローバルバッファ（-r オプション指定）を割り当てても同じ効果が得られます。インデクスに対するユニット単位グローバルバッファの割り当ては，指定するインデクスのインデクス格納 RD

エリアの配置によって 4 種類に分類されます。グローバルバッファの共用形態別の推奨条件（-i オプション指定）を次の表に示します。

表 26–17　グローバルバッファの共用形態別の推奨条件（-i オプション指定）

指定方法（-i オプションで指定したインデクス格納用 RD エリア）			インデクス分割形式	バッファ共用形態	メリット	推奨条件
異なるサーバの RD エリア	同一ユニットの RD エリア	異なるユニットの RD エリア				
なし	なし	なし	非分割で同一サーバ内分割	非共用	他サーバと共用しないため，多重障害時でも定常時のバッファ性能を維持できる	すべての受け入れユニットにバッファを確保するためメモリに余裕がある環境で，縮退時も縮退前のバッファ性能を維持したいインデクスに対する適用をお勧めします。
あり	あり	なし	同一ユニット内分割	ユニット内サーバ共用	多重障害でも定常時のバッファ性能を維持できる	すべての受け入れユニットにバッファを確保するためメモリに余裕がある環境で，縮退時も縮退前のバッファ性能を維持したいインデクスに対する適用をお勧めします。
あり	なし	あり	ユニット間分割で同一ユニット内分割なし	ユニット間サーバ共用	• 現用系のバックエンドサーバでは指定バッファ面数をすべて使用できる • 縮退時は受け入れユニットの資源を共用するのでメモリ効率が良い	現用時の性能を重視し，縮退時にバッファ資源を共用したい場合の適用をお勧めします。
あり	あり	あり	ユニット間分割で同一ユニット内分割あり	ユニット内ユニット間サーバ共用	• 縮退時は受け入れユニットの資源を共用するのでメモリ効率が良い • 共用サーバ数を均等にすることで縮退時の負荷分散を	縮退時，バッファ資源の共用，及び均等負荷分散を実現したい場合の適用をお勧めします。

指定方法（-i オプションで指定したインデクス格納用 RD エリア）			インデクス分割形式	バッファ共用形態	メリット	推奨条件
異なるサーバの RD エリア	同一ユニットの RD エリア	異なるユニットの RD エリア				
					均等に設計できる	

次に示す-i オプション指定のバッファ設計指針を参考にインデクス専用バッファを割り当てるかどうか選択してください。

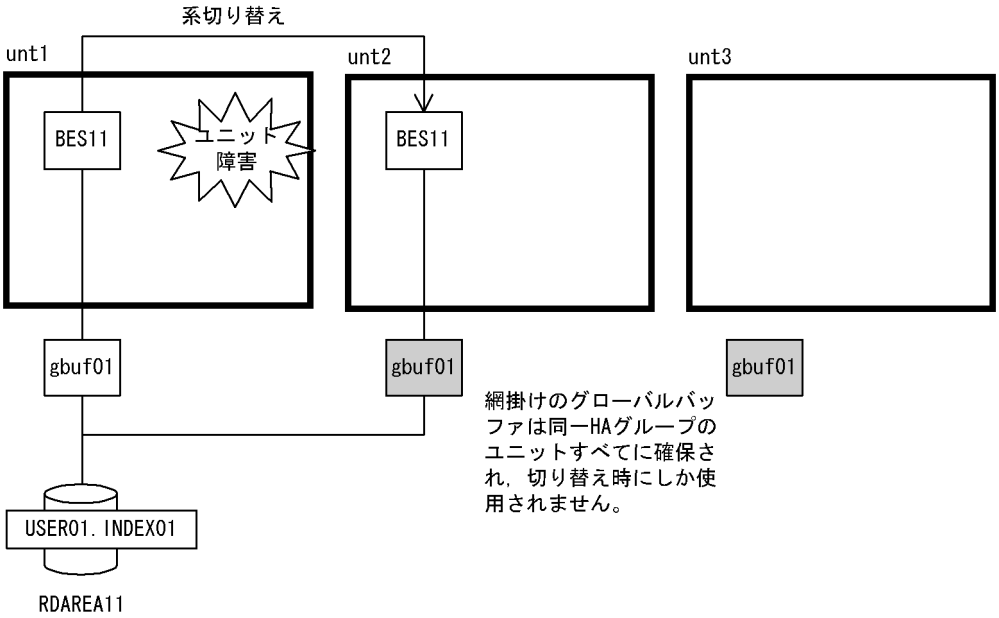
-i オプション指定のバッファ設計指針

1. 非分割インデクス，同一サーバ内分割インデクス，又は同一ユニット内分割インデクスの場合：同一 HA グループの各ユニットにグローバルバッファを確保できるだけメモリに余裕があれば専用バッファを定義します。
2. 1.に当てはまらない場合：インデクス用グローバルバッファを定義します。共用するサーバに見合ったバッファ面数を割り当ててください。

(a)　非分割インデクス用グローバルバッファの割り当て方法

非分割インデクスを pdbuffer オペランドの-i オプションで指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 200 -c
```

〔説明〕

非分割インデクス USER01.INDEX01 を指定します。

非分割インデクスの共用グローバルバッファへの割り当て例です。ほかのサーバとはグローバルバッファを共用しないで，占有します。系切り替え時も BES11 専用の gbuf01 を使用するのでヒット率は低下しませんが，すべての受け入れユニットに対して切り替え時，使用するためのバッファが確保されるのでメモリを多く消費します。

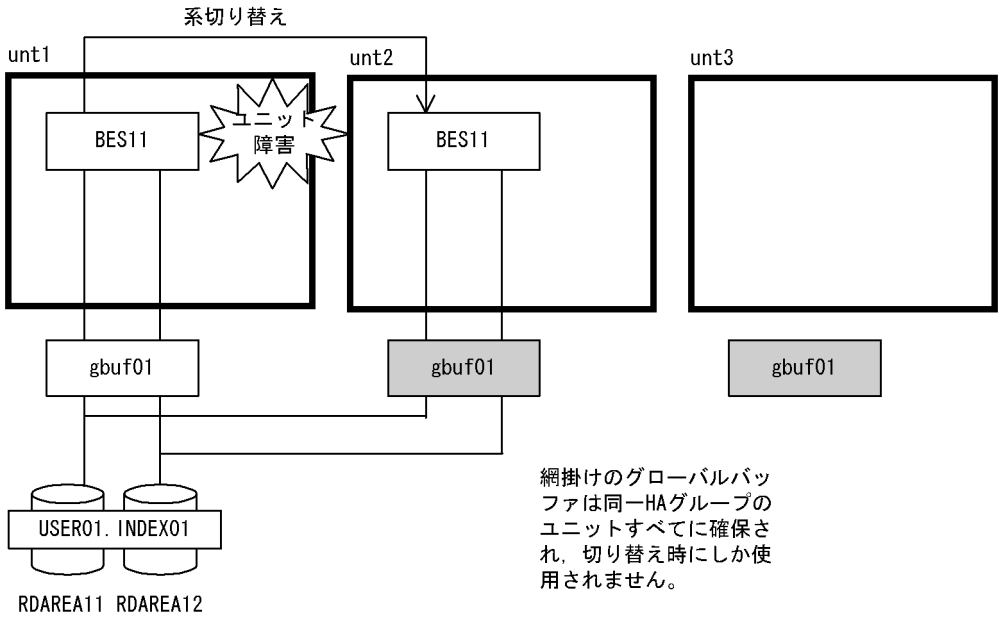
注

受け入れユニットすべてに同じグローバルバッファが作成されます。このバッファは代替されるまで使用されません。

(b)　同一サーバ内分割インデクス用グローバルバッファの割り当て方法

同一サーバ内に分割されたインデクスを pdbuffer オペランドの-i オプションで指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

〔説明〕

サーバ内横分割インデクス USER01.INDEX01 を指定します。

同一サーバ内に分割されたインデクスの共用グローバルバッファへの割り当て例です。ほかのサーバとは共用しないで，占有します。系切り替え時も BES11 専用の gbuf01 を使用するのでヒット率は低下しませんが，すべての受け入れユニットに切り替え時使用するためのバッファが確保されるのでメモリを多く消費します。

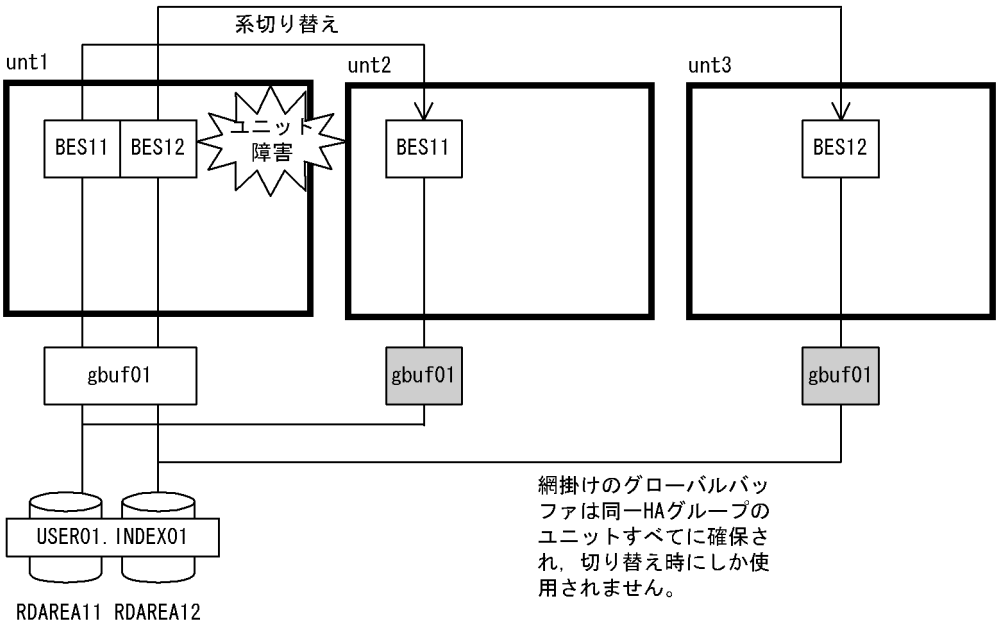
注

- 受け入れユニットすべてに同じグローバルバッファが作成されます。このグローバルバッファは代替されるまで使用されません。
- インデクス格納用 RD エリアのページサイズを同じにすることでメモリ効率が向上します。

(c)　同一ユニット内分割インデクス用グローバルバッファの割り当て方法

同一ユニット内に分割されたインデクスを pdbuffer オペランドの-i オプションで指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

〔説明〕

同一ユニット内に分割されたインデクスの共用グローバルバッファへの割り当て例です。系切り替え時も gbuf01 を使用するのでヒット率は低下しませんが，すべての受け入れユニットに切り替え時使用するためのグローバルバッファが確保されるのでメモリを多く消費します。

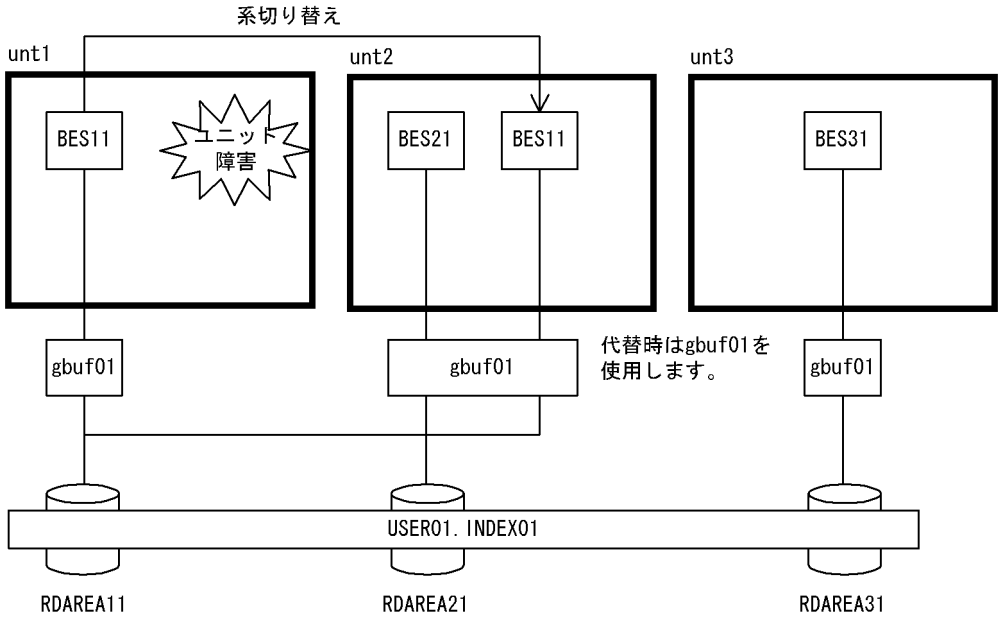
注

- 受け入れユニットすべてに同じグローバルバッファが作成されます。このグローバルバッファは代替されるまで使用されません。
- この指定のグローバルバッファは複数サーバ間で共用されます。
- インデクス格納用 RD エリアのページサイズを同じにすることでメモリ効率が向上します。
- pdbuffer オペランドの-l オプション省略時のバッファサイズは指定したインデクス格納用 RD エリア内の最大ページサイズとなります。

(d) ユニット間分割で同一ユニット内分割なしインデクス用グローバルバッファの割り当て方法

異なるユニット間のサーバに分割されたインデクスを pdbuffer オペランドの-i オプションで指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

〔説明〕

異なるユニット間のサーバに横分割されたインデクス USER01.INDEX01 を指定します。

ユニット間分割で同一ユニット内分割なしのインデクスに対するグローバルバッファの割り当て例です。縮退前は gbuf01 のリソース（バッファ数）を BES11 で占有します。縮退時は BES21 と BES11 で共用するので使用できるリソース（バッファ数）は半分となります。

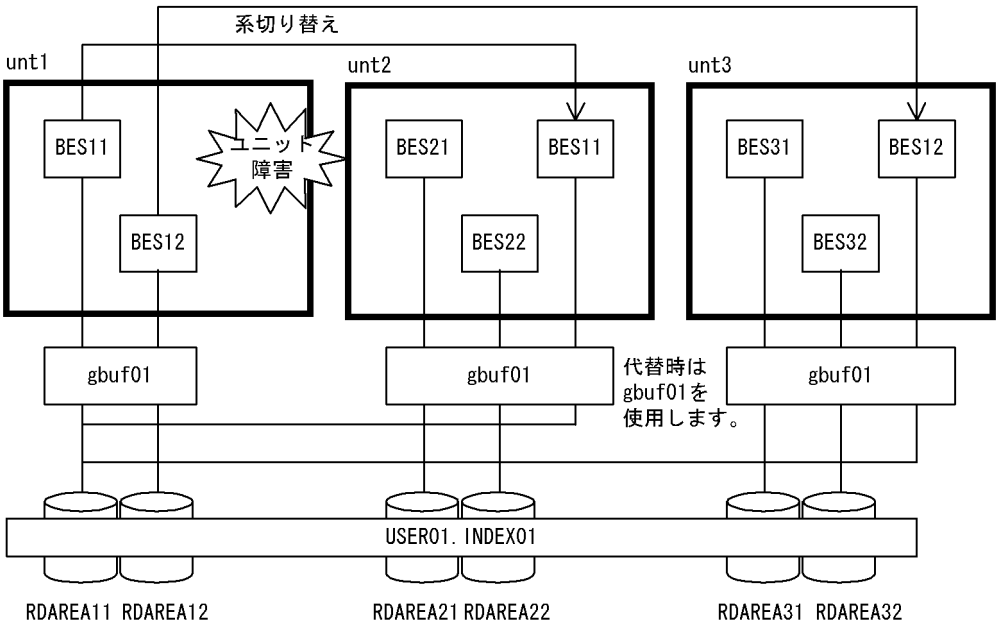
注

- 移動先が一つのユニットになるのでそのユニットのグローバルバッファだけ負荷が高くなります。ユニット間サーバ共用バッファを複数定義し，各ユニットの負荷が均等になるように設計してください。
- 受け入れユニットすべてに同じグローバルバッファが作成されます。
- この指定のグローバルバッファは現用時には一つのサーバで占有され，縮退時には複数サーバ間で共用されます。
- インデクス格納用 RD エリアのページサイズを同じにすることでメモリ効率が向上します。
- pdbuffer オペランドの-l オプション省略時のバッファサイズは，指定したインデクス格納用 RD エリアの内の最大ページサイズとなります。

(e) ユニット間分割で同一ユニット内分割ありインデクスのグローバルバッファの割り当て方法

ユニット間分割で同一ユニット内で分割されたインデクスを pdbuffer オペランドの-i オプションで指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -i USER01.INDEX01 -n 1000 -c
```

〔説明〕

ユニット間分割ありのサーバ間横分割インデクス USER01.INDEX01 を指定します。

ユニット間分割で同一ユニット内で分割されたインデクスの共用グローバルバッファへの割り当て例です。定常時は各ユニットが二つのバックエンドサーバ間で gbuf01 を共用するので一つのバックエンドサーバに割り当てられるバッファ面数は半分です。縮退時は三つのバックエンドサーバ間で gbuf01 を共用するので一つのバックエンドサーバに割り当てるバッファ面数は各ユニットが 1/3 と均等になります。

注

- 縮退時受け持つサーバ数が各受け入れユニットで均等になるように設計してください。
- 受け入れユニットすべてに同じグローバルバッファが作成されます。
- この指定のグローバルバッファは複数サーバ間で共用されます。
- インデクス格納用 RD エリアのページサイズを同じにすることでメモリ効率が向上します。

- pdbuffer オペランドの-l オプション省略時のバッファサイズは指定したインデクス格納用 RD エリアの内の最大ページサイズとなります。

(4)　OTHER 用グローバルバッファの割り当て方法（-o オプション指定）

OTHER 用グローバルバッファは，影響分散スタンバイレス型系切り替え機能の適用ユニットすべてに確保されます。次に示すように割り当てられます。

- pdbuffer オペランドの-c オプション指定の OTHER 用バッファ定義はシステムで一つだけ指定できます。-c オプション指定の OTHER 用バッファを複数定義した場合は，システム共通定義で最初に記述されている定義を有効とします。
- 影響分散スタンバイレス型系切り替え機能の適用ユニット内で稼働するホスト BES，及びゲスト BES に配置された RD エリアのうち，-r オプション指定のグローバルバッファが未割り当ての RD エリアに割り当てられます。
- pdbuffer オペランドの-l オプションでバッファサイズを省略した場合，同一 HA グループ内で-c オプションと- r オプションを指定したグローバルバッファを割り当てていない RD エリアがある場合には，同一 HA グループ内で-c オプションと- r オプションを指定したグローバルバッファを割り当てていない RD エリア中の最大ページサイズとなります。同一 HA グループのすべてに-c オプションと- r オプションを指定したグローバルバッファを割り当てている場合には，同一 HA グループ内の RD エリアの最大ページサイズとなります。
- pdbuffer オペランドの-c オプションを指定した OTHER 用グローバルバッファと-c オプションを指定しない OTHER 用グローバルバッファの定義は重複指定できます。OTHER 用グローバルバッファ定義の重複指定時の関係を次の表に示します。

表 26–18　OTHER 用グローバルバッファ定義の重複指定時の関係

-c オプションを指定した OTHER 用グローバルバッファ定義	-c オプションを指定しない OTHER 用グローバルバッファ定義	影響分散スタンバイレス型系切り替え機能の適用ユニット	影響分散スタンバイレス型系切り替え機能の非適用ユニット
あり	あり	-c オプションありで定義を割り当てる	-c オプションなしで定義を割り当てる
あり	なし	-c オプションありで定義を割り当てる	-c オプションありで定義を割り当てる
なし	あり	OTHER 用バッファを割り当てない	-c オプションなしで定義を割り当てる
なし	なし	OTHER 用バッファを割り当てない	OTHER 用バッファを割り当てない

(a)　OTHER 用グローバルバッファの推奨条件

- オンライン中で RD エリアを追加するシステム
- アクセス頻度が少ない RD エリア
- アクセスページ数の少ない RD エリア

- 格納ページ数が非常に多い RD エリア（バッファヒットを期待しない RD エリア）

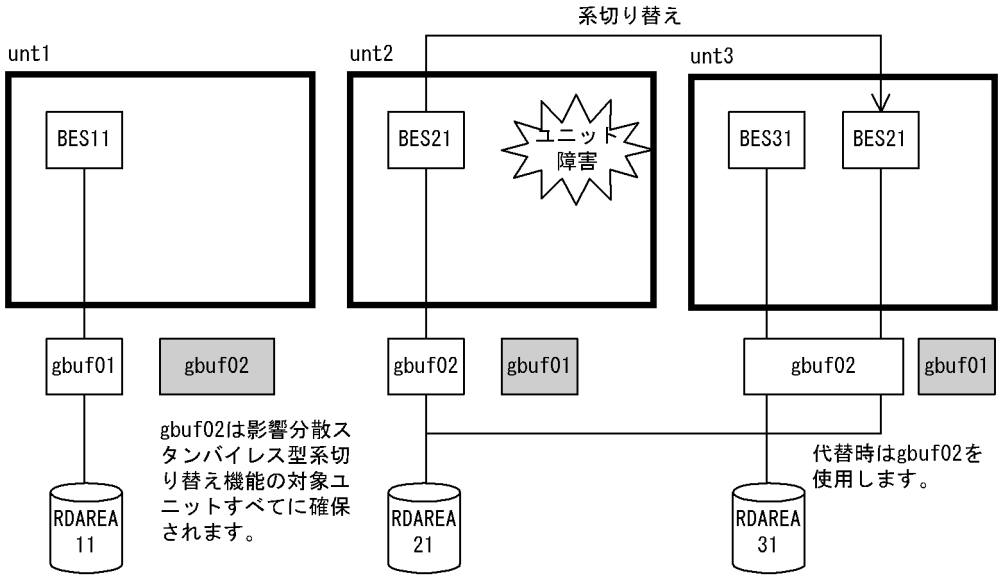
(b)　OTHER 用グローバルバッファの注意事項

- ユニットに確保した OTHER 用グローバルバッファ資源は OTHER 用バッファを割り当てたサーバ間で均等に分割して使用します。したがって，使用するサーバ数に見合ったバッファ面数を pdbuffer オペランドの-n オプションで指定するようにしてください。
- オンライン中で RD エリアを追加するシステムでは，今後追加が予想される RD エリアのページサイズを考慮し，pdbuffer オペランドの-l オプションでバッファサイズを設定してください。

(c)　OTHER 用グローバルバッファの割り当て方法の例

pdbuffer オペランドの-o オプションを指定します。

●システム構成例



●グローバルバッファの定義

```
pdbuffer -a gbuf01 -r RDAREA11 -n 500 -c
pdbuffer -a gbuf02 -o -n 1000 -c
```

〔説明〕

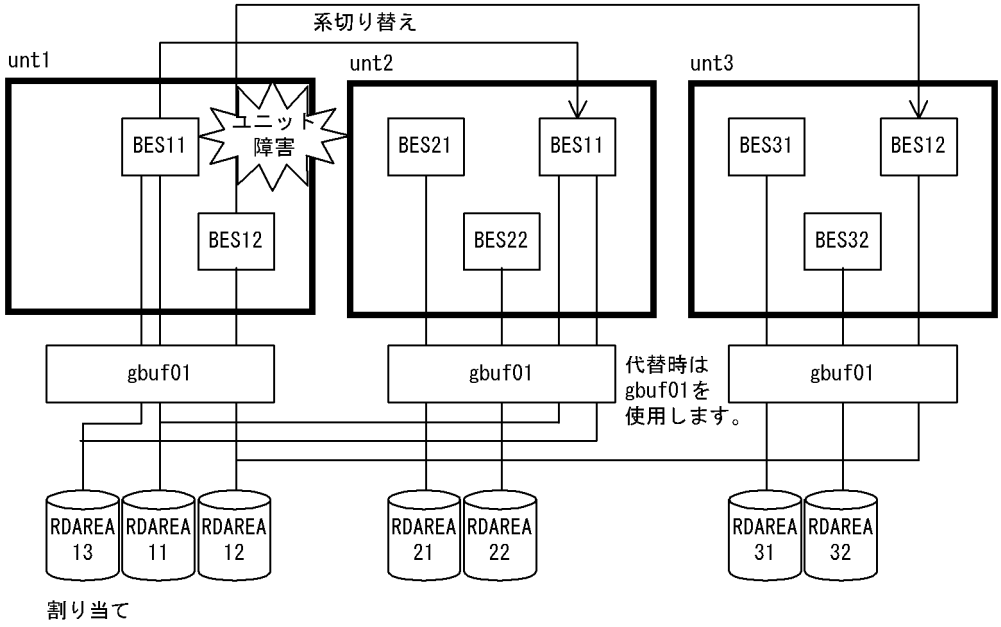
pdbuffer オペランドの-o オプションと-c オプションを指定します。

RDAREA11 に専用バッファを割り当て，そのほかの RD エリアには OTHER 用バッファを割り当てます。OTHER 用バッファは影響分散スタンバイレス型系切り替え機能の適用ユニットすべてに作成されます。

(5) 構成変更時のグローバルバッファの割り当て（データベース構成変更ユティリティ）

データベース構成変更ユティリティの制御文の globalbuffer オペランドに，既にあるグローバルバッファ名を指定して実施します。グローバルバッファは pdbufls コマンドで確認できます。

●システム構成例



●構成変更定義

```
create rdarea RDAREA13 globalbuffer gbuf01 server name BES11
                :
```

〔説明〕

追加 RD エリアに共用グローバルバッファ gbuf01 を割り当てます。追加した RD エリアは系切り替え時も gbuf01 を使用します。

〔設計指針〕

- 系の切り替え時，及び系の切り戻し時も globalbuffer オペランドで指定したグローバルバッファを割り当てます。
- インデクス用グローバルバッファ，及び LOB 用グローバルバッファは割り当てられません。
- 指定するグローバルバッファの長さは，追加する RD エリアのページ長より長くなければなりません。グローバルバッファ長は，pdbufls コマンドで確認できます。
- ここで指定したグローバルバッファの割り当ては，サーバの正常終了（HiRDB システムの正常終了，計画停止，ユニットの正常終了，又はサーバ単独の正常終了）時に無効となります。そのため，次回のサーバの正常開始時には，あらかじめシステム共通定義の pdbuffer オペランドでグローバルバッファを割り当てておかなければなりません。ただし，-o オプション指定のグローバルバッファ

を割り当てている場合はそのグローバルバッファに割り当てますのでシステム共通定義を変更する必要はありません。

- HiRDB がグローバルバッファの割り当てに失敗した場合，RD エリアは追加されません。

26.5.9 HiRDB サービスの開始方法の設定の確認（HiRDB/シングルサーバ限定）

HiRDB の環境設定が終わった後に，pdntenv コマンドで HiRDB のサービスの開始方法を確認してください。次に示す場合は，HiRDB のサービスの開始方法を manual（HiRDB の終了時に HiRDB のサービスを停止しない）にする必要があります。

- **HiRDB が異常終了した系で再開始できない場合に系を切り替えるとき**

（例）

```
pdntenv

 ***** HiRDB setup information *****
 Program Name         = HiRDB/SingleServer
 Version              = 06-01
 Character Code       = sjis
 Dat Hard Compress    = on
 Setup ID             = ----
 Rsh Port Name        = pdrshsrv
 Path Name            = C:¥win32app¥hitachi¥hirdb_s
 Service Complete     = manual
```

〔説明〕

Service Complete が manual の場合は問題ありません。auto が表示されている場合は，-sc オプションに manual を指定して pdntenv コマンドを実行してください。

この操作はモニタモードかサーバモードかに関係なく，HiRDB が異常終了した系で再開始できない場合に系を切り替えるときは必ず行ってください。

26.5.10 HiRDB に関連するサービスの準備

クラスタソフトウェアからサービスを起動するため，サービスがログオン時に使用するアカウントを変更する必要があります。変更する必要があるサービスを次の表に示します。なお，HiRDB 管理者の登録については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

表 26–19　アカウントを変更する必要があるサービス

サービスの種類		サービス名	サービスがログオン時に使用するアカウント
HiRDB のクラスタサービス（HiRDB/ClusterService）		HiRDBClusterService	HiRDB 管理者
HiRDB のサービス	HiRDB/シングルサーバの場合（HiRDB/SingleServer）	HiRDBSingleServer	
	HiRDB/パラレルサーバの場合（HiRDB/ParallelServer）	HiRDBParallelServer	

HiRDB のクラスタサービスを開始又は終了するときに，次に示すバッチファイルを実行します。

- **開始時：**%PDDIR%¥conf¥pdshastr.bat
- **終了時：**%PDDIR%¥conf¥pdshaend.bat

HiRDB の開始コマンドを"pdshastr.bat"ファイルに指定してください。開始コマンドの例を次に示します。

- **HiRDB/シングルサーバの場合：**%PDDIR%¥bin¥pdstart.exe
- **HiRDB/パラレルサーバの場合：**%PDDIR%¥bin¥pdstart.exe -q

HiRDB の終了コマンドを"pdshaend.bat"ファイルに指定してください。終了コマンドの例を次に示します。

- **HiRDB/シングルサーバの場合：**%PDDIR%¥bin¥pdstop.exe -f -q
- **HiRDB/パラレルサーバの場合：**%PDDIR%¥bin¥pdstop.exe -z -q

このコマンドを実行すると，共有ディスク上の RD エリアが障害閉塞することがあります。この場合，データベース回復ユティリティで共有ディスク上の RD エリアを回復してください。

注意事項

Hitachi HA Toolkit Extension を使用する場合

- HiRDB のクラスタサービスを開始又は終了するときに実行する，上記のバッチファイルを用意する必要があります。

26.5.11　監査証跡ファイルの運用

実行者　HiRDB 管理者，及び監査人

スタンバイ型系切り替え機能，及び 1：1 スタンバイレス型系切り替え機能の場合

監査証跡ファイルは HiRDB 管理者が共有ディスクに作成します。HiRDB 管理者，及び監査人は，共有ディスク上の監査証跡ファイルを運用してください。

影響分散スタンバイレス型系切り替え機能の場合

監査証跡ファイルは HiRDB 管理者が正規ユニットの共有ディスクに作成します。このとき，作成先ディスクには，各サーバに対応する共有ディスク（各サーバのシステムログファイル，シンクポイントダンプファイル，サーバ用ステータスファイルを格納するディスク）以外のディスクを選ぶ必要があります。

なお，系の切り替え先では，受け入れユニットの監査証跡ファイルを共用します。

HiRDB 管理者，及び監査人は，正規ユニットの監査証跡ファイルと受け入れユニットの監査証跡ファイルを運用してください。

(1)　監査証跡ファイルの作成

(a)　スタンバイ型系切り替え機能，及び 1：1 スタンバイレス型系切り替え機能の場合

監査証跡ファイルは，HiRDB 管理者が共有ディスクに作成します。

(b)　影響分散スタンバイレス型系切り替え機能の場合

監査証跡ファイルは HiRDB 管理者が共有ディスクに作成します。このとき，作成先ディスクには，各サーバに対応する共有ディスク（各サーバのシステムログファイル，シンクポイントダンプファイル，サーバ用ステータスファイルを格納するディスク）以外のディスクを選ぶ必要があります。

各サーバに対応する共有ディスクに監査証跡ファイルを作成すると，系の切り替えによってディスクが別のホストに切り替わるため，ユニット内のほかの稼働中サーバが監査証跡を出力できなくなります。なお，系の切り替え先では，受け入れユニットの監査証跡ファイルを使用します。

(2)　監査証跡ファイルの運用

(a)　スタンバイ型系切り替え機能，及び 1：1 スタンバイレス型系切り替え機能の場合

系切り替えが発生した場合，HiRDB は共有ディスク上の監査証跡ファイルに監査事象を記録します。監査事象の記録に関する監査証跡ファイルの運用については，「監査証跡ファイルの運用」を参照してください。

(b)　影響分散スタンバイレス型系切り替え機能の場合

系切り替えが発生した場合，HiRDB は切り替え先の受け入れユニットが使用中の監査証跡ファイルに監査事象を記録します。このとき，監査事象の記録に関する監査証跡ファイルの運用は，受け入れユニットでの運用に統一されます。

影響分散スタンバイレス型系切り替え機能を適用したシステムで監査証跡を取得する場合は，全ユニットで監査証跡を取得してください。

(3)　監査証跡の取得

(a)　スタンバイ型系切り替え機能，及び 1：1 スタンバイレス型系切り替え機能の場合

系切り替えが発生した場合，監査証跡取得状態の引き継ぎについては，切り替え元ユニットの停止状態に依存します。切り替え先の系が再開始の場合は系を切り替える前の状態を引き継ぎます。切り替え先の系が正常開始の場合は pd_audit オペランドの指定に従います。

(b)　影響分散スタンバイレス型系切り替え機能の場合

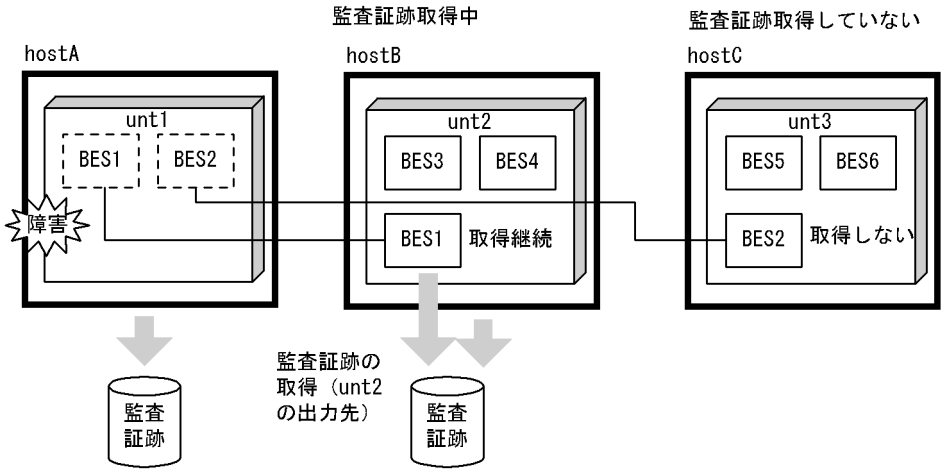
系切り替えが発生した場合，監査証跡取得については，受け入れユニットの状態に従います。影響分散スタンバイレス型系切り替え機能での監査証跡の取得状況を次の表に示します。

表 26–20　影響分散スタンバイレス型系切り替え機能での監査証跡の取得状況

ユニット種別	ユニットの状態	受け入れユニット	
		取得中	取得していない
正規ユニット	取得中	取得する	取得しない
	取得していない	取得する	取得しない

影響分散スタンバイレス型系切り替え機能適用時の監査証跡取得状況の例を次の図に示します。

図 26–45　影響分散スタンバイレス型系切り替え機能適用時の監査証跡取得状況の例



(4)　pdload の実行

(a)　スタンバイ型系切り替え機能，及び 1：1 スタンバイレス型系切り替え機能の場合

HiRDB 管理者は，監査証跡ファイルを入力情報として，pdload を実行してください（認証は監査人）。ただし，障害などで系が切り替わった場合，HiRDB は切り替わる直前の監査事象を正しく取得しません。このため，pdload を実行しても切り替え直前のデータを取得できない場合があります。

(b)　影響分散スタンバイレス型系切り替え機能の場合

HiRDB 管理者は，正規ユニット及び受け入れユニットの監査証跡ファイルを入力情報として，pdload を実行してください（認証は監査人）。なお，系切り替え中のサーバの監査証跡は，受け入れユニットに属するサーバの情報として処理されます。

障害などで系が切り替わった場合，HiRDB は切り替わる直前の監査事象を正しく取得しません。このため，pdload を実行しても切り替え直前のデータを取得できない場合があります。

なお，系切り替えなどによって影響分散スタンバイレス型系切り替え対象ユニットに実行系サーバが 0 個となった場合，そのユニットに対する監査証跡ファイルの pdload は実行できません。この場合は，該当するユニットに実行系サーバを 1 個以上系切り替えした後で，pdload を実行してください。

障害中の運用：

障害中の監査証跡のロード運用は次のとおりです。

1. 稼働中ホストで系切り替え前の監査証跡ファイルを格納したディスクを手動で活動化
2. 正規ユニット及び受け入れユニットの監査証跡ファイルを入力情報として pdload 実行

障害復旧後の運用：

障害復旧後の監査証跡のロード運用は障害発生前と同じです。

26.5.12　NetBackup 連携機能を使用する場合の注意事項

系切り替え構成で NetBackup 連携機能を使用する場合で，バックアップを取得した NetBackup クライアントとは別の NetBackup クライアントで回復を行うときは，次の環境設定と運用が必要です。ただし，この環境設定と運用は，JP1/VERITAS NetBackup 5.0 以降を使用している場合にだけ適用してください。

- NetBackup マスターサーバのインストールマシンに，次の空ファイルを作成します。

  `Install_path¥NetBackup¥db¥altnames¥No.Restrictions`

- 回復に使用する NetBackup クライアント名称（ポリシに設定するクライアントホスト名）は，回復に使用するバックアップデータを取得した NetBackup クライアント名称に変更する必要があります。NetBackup クライアント名称の変更などの詳細は，NetBackup に添付されているドキュメントを参照してください。

26.6　MSFC に関する準備（Windows Server 2012 の場合）

MSFC に関する準備には次に示す項目があります。ここでは，これらの項目の操作方法の概略を説明します。詳細については，MSFC のドキュメントなどを参照してください。

- サービスの作成
- 共有ディスクの登録
- リソースの登録（クライアントアクセスポイントの登録，HiRDB のサービスの登録，HiRDB のクラスタサービスの登録）

モニタモードで運用する場合

モニタモードで運用する場合は，この節の説明と MSFC のドキュメントを参照して環境設定を行ってください。

サーバモードで運用する場合

サーバモードで運用する場合は，次に示す箇所，マニュアル，又はドキュメントの説明を参照して環境設定を行ってください。

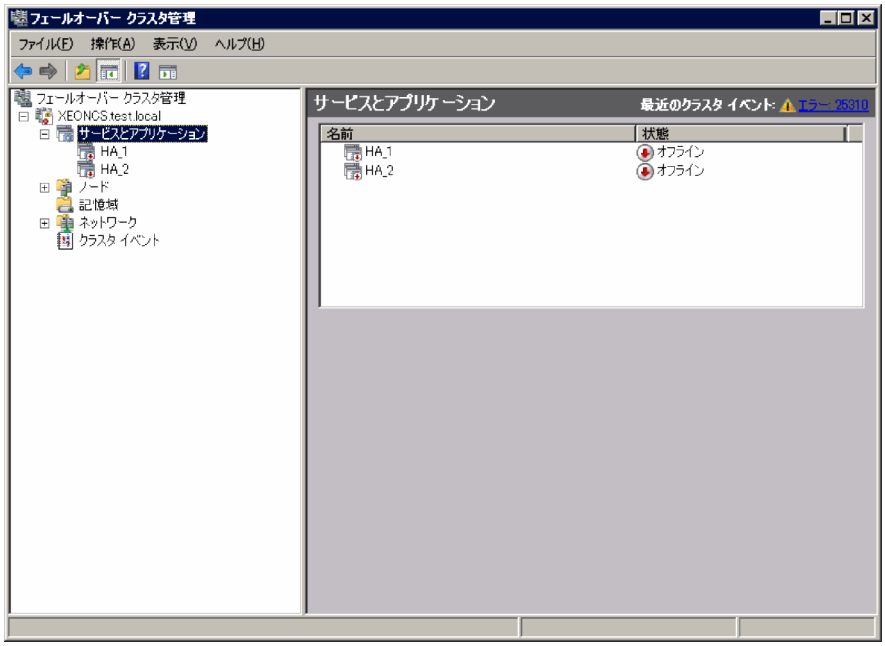
- この節の説明
- 「Hitachi HA Toolkit Extension に関する準備（サーバモード限定）」の説明
- マニュアル「Hitachi HA Toolkit」
- MSFC のドキュメント

26.6.1　サービスの作成

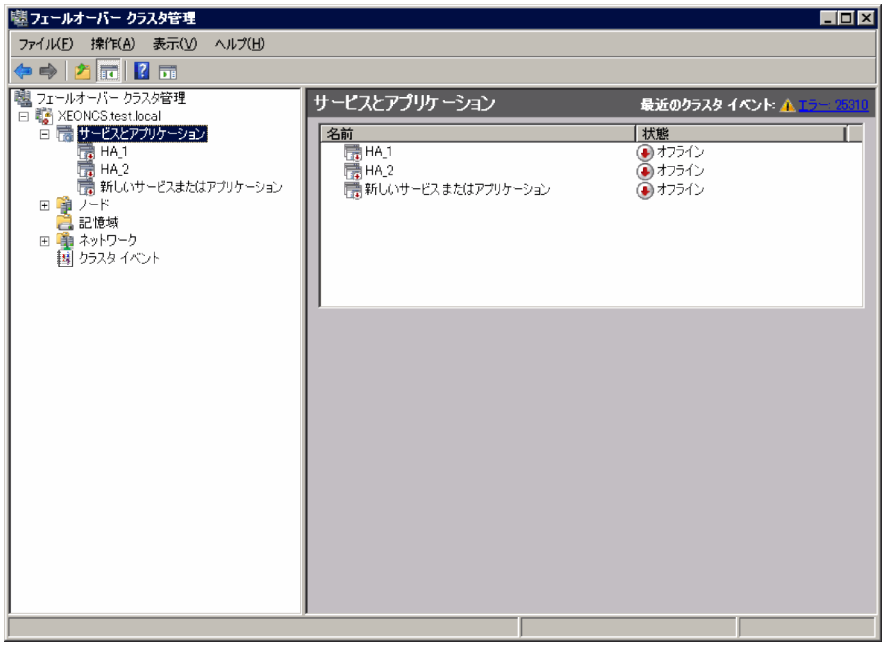
MSFC のフェールオーバークラスタ管理で，MSFC のサービスを作成する必要があります。サービスの作成手順を次に示します。

〈手順〉

1. **［スタート］**－**［プログラム］**－**［管理ツール］**－**［フェールオーバー クラスタ管理］**でフェールオーバークラスタ管理を起動します。



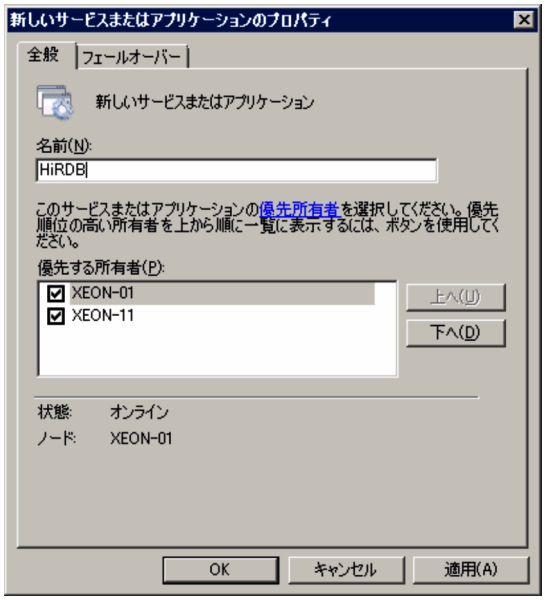
2. ツリーの［**サービスとアプリケーション**］を右クリックして，［**その他アクション**］－［**空のサービスまたはアプリケーションの作成**］を選択します。



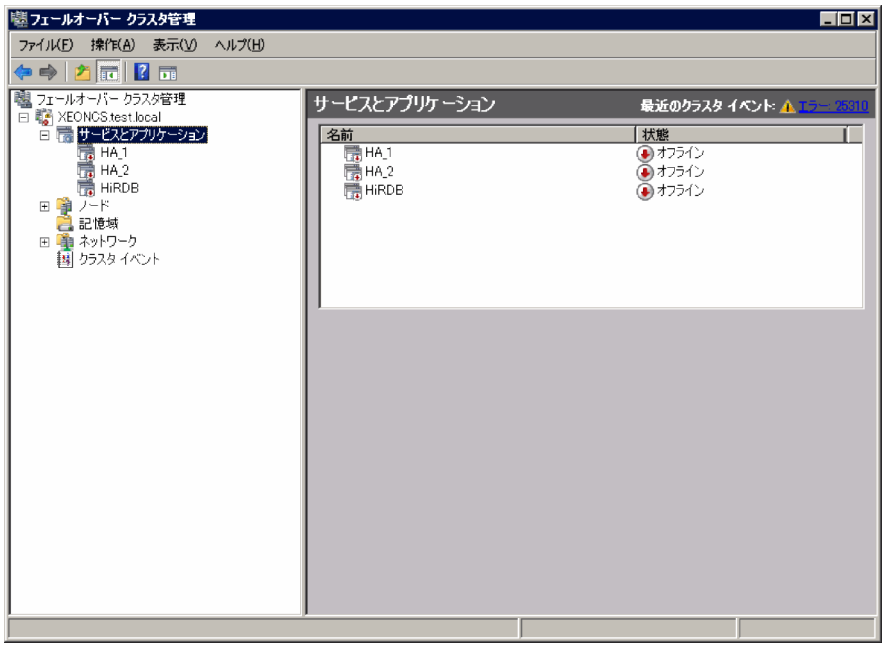
3. 作成されたサービスを選択して右クリックします。［**プロパティ**］を選択し，［**プロパティ**］ダイアログを表示し，［**全般**］タブを選択します。

   名前：サービスの名前（任意）を指定します。

   優先する所有者：優先する所有者を指定します。



4. **[OK]** をクリックすると，指定した名前でサービスが作成されます。



26.6.2　リソースの登録

(1)　モニタモードの場合のリソースの登録

モニタモードの場合に，MSFC のサービスに登録するリソースを次の表に示します。

表 26–21　MSFC のサービスに登録するリソース（モニタモードの場合）

リソースの種類	説明
共有ディスク	共有ディスクを登録します。

リソースの種類	説明
クライアントアクセスポイント	IP アドレスを引き継ぐ場合にクライアントアクセスポイントを登録します。
汎用サービス	HiRDB のサービスを登録します。HiRDB が異常終了又はユニットダウンした系で再開始できない場合に，系を切り替えるときに登録します。 登録するサービス名を次に示します。 HiRDB/シングルサーバの場合：HiRDB/SingleServer HiRDB/パラレルサーバの場合：HiRDB/ParallelServer
	HiRDB のクラスタサービス（サービス名：HiRDB/ClusterService）を登録します。

なお，MSFC のサービスに登録する各リソースの依存関係は，次に示す順番で起動されるように登録してください。

1. 共有ディスク及びクライアントアクセスポイント
2. HiRDB のサービス※
3. HiRDB のクラスタサービス

注※　HiRDB のサービスを登録しない場合は HiRDB のクラスタサービスが開始します。

(2)　サーバモードの場合のリソースの登録

サーバモードの場合に，MSFC のサービスに登録するリソースを次の表に示します。

表 26–22　MSFC のサービスに登録するリソース（サーバモードの場合）

リソースの種類	説明
共有ディスク	共有ディスクを登録します。
汎用サービス	Hitachi HA Toolkit Extension のサービスを登録します。 詳細については，マニュアル「Hitachi HA Toolkit」を参照してください。

なお，MSFC のサービスに登録する各リソースの依存関係は，次に示す順番で起動されるように登録してください。

1. 共有ディスク
2. Hitachi HA Toolkit Extension のサービス

注意事項

サーバモードで Hitachi HA Toolkit Extension を使用する場合，HiRDB のクラスタサービスの起動は Hitachi HA Toolkit Extension の起動用バッチファイルから行います。このため，Hitachi HA Toolkit Extension を使用する場合は HiRDB のクラスタサービス及び HiRDB のサービスを登録しないでください。

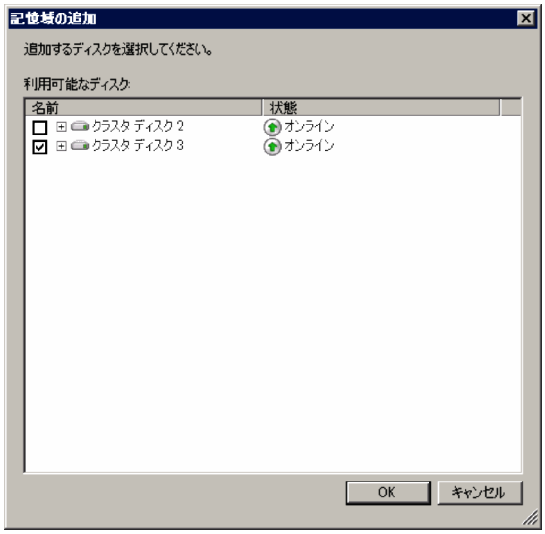
26.6.3　共有ディスクの登録

フェールオーバークラスタ管理で共有ディスクを MSFC のサービスに登録します。

Hitachi HA Toolkit Extension を使用する場合は，HiRDB ファイルシステム領域を作成する共有ディスクがオンラインになった後に，HiRDB を監視する Hitachi HA Toolkit Extension のサービスがオンライン操作されるように，MSFC の依存関係を設定してください。共有ディスクの登録手順を次に示します。

〈手順〉

1. 「サービスの作成」で作成した MSFC のサービスを選択して右クリックします。［記憶域の追加］を選択し，［記憶域の追加］ダイアログで切り替え対象のディスクを選択します。



2. **［OK］**をクリックすると，選択したディスクが登録されます。



26.6.4　クライアントアクセスポイントの登録（モニタモードの場合だけ）

クライアントアクセスポイントの登録は，IP アドレスを引き継ぐ場合に行います。IP アドレスを引き継がない場合は，この操作をする必要はありません。

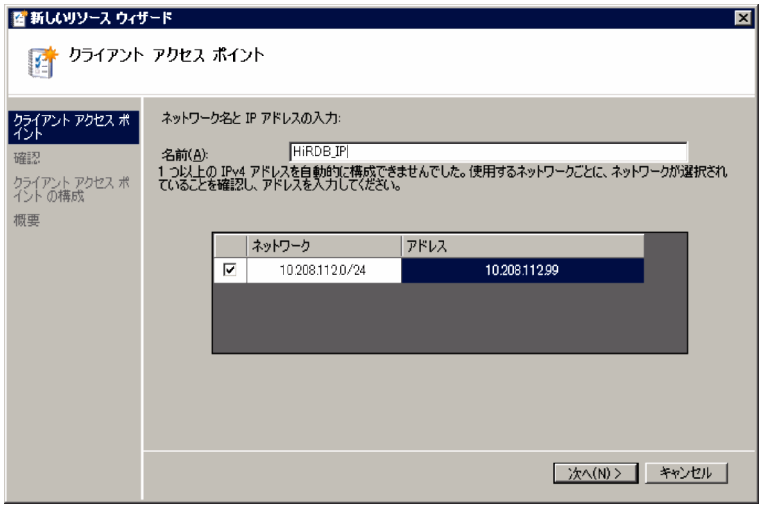
HiRDB が系切り替え機能で使用する**仮想ネットワーク名の IP アドレス**を登録します。なお，DNS サーバを使用しない場合は，仮想ネットワーク名，及びその IP アドレスを hosts ファイルに登録する必要があります。IP アドレスの登録手順を次に示します。

〈手順〉

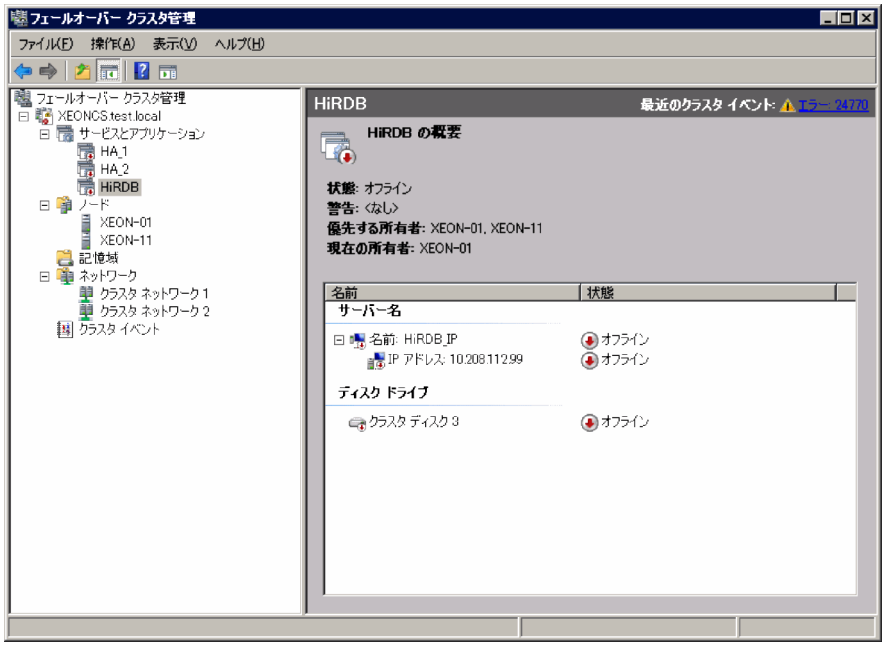
1. 「サービスの作成」で作成した MSFC のサービスを選択して右クリックします。**［リソースの追加］**－**［クライアントアクセスポイント］**を選択し，**［新しいリソース ウィザード］**画面を表示します。

   名前：ネットワーク名を指定します。

   アドレス：引き継ぐ IP アドレスを指定します。



2. **［次へ］**をクリックし，指定内容を確認します。**［OK］**をクリックすると，クライアントアクセスポイントが登録されます。



26.6.5　HiRDB のサービスの登録（モニタモードの場合だけ）

HiRDB のサービスを MSFC の汎用サービスに登録すると，HiRDB が異常終了又はユニットダウンした系で再開始できない場合に，系を切り替えることができます。ただし，障害を検知してから系を切り替えるまでに時間が掛かります。系切り替え時間を短縮したい場合は，サーバモードを使用してください。

登録する HiRDB のサービス名は次のとおりです。

- **HiRDB/SingleServer**（HiRDB/シングルサーバの場合）
- **HiRDB/ParallelServer**（HiRDB/パラレルサーバの場合）

ただし，相互系切り替え構成などでマルチ HiRDB を使用する場合は，HiRDB をインストールしたときに指定した**セットアップ識別子**をサービス名に付けて指定します。

（例）セットアップ識別子が UNT1 の場合

HiRDB/SingleServerUNT1

HiRDB/ParallelServerUNT1

[コントロールパネル] − **[サービス]** の **[サービス]** ダイアログでサービス名を確認してから，サービス名を指定してください。

注意事項

HiRDB/シングルサーバの場合のサービスの開始方法の設定について

HiRDB のサービスの開始方法の設定（Service Complete の設定）を manual にする必要があります。manual にしない場合は，HiRDB のサービスを MSFC に登録しないでください。

HiRDB のサービスの登録手順を次に示します。

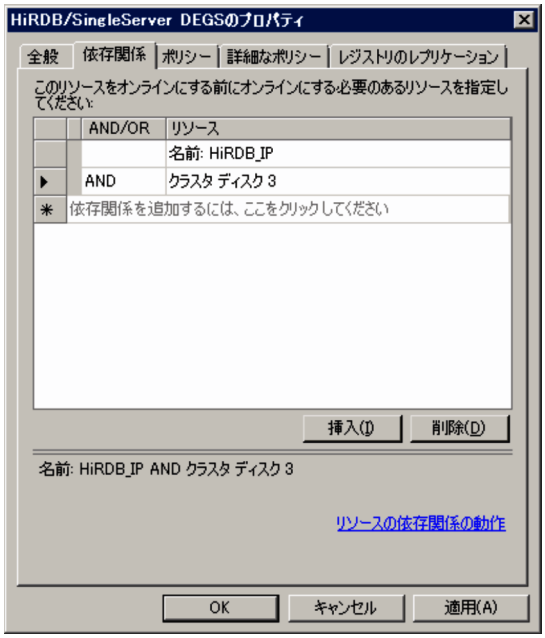
〈手順〉

1. 「サービスの作成」で作成した MSFC のサービスを選択して右クリックします。**［リソースの追加］** – **［汎用サービス］** を選択し，**［新しいリソース ウィザード］** 画面を表示します。表示される一覧から HiRDB のサービスを選択します。



2. **［次へ］** をクリックし，内容を確認してから **［OK］** をクリックします。
3. 登録した HiRDB のサービスを選択して右クリックします。**［プロパティ］** を選択し，登録したサービスのプロパティのダイアログを表示します。
4. **［依存関係］** タブを選択します。

   リソース： 共有ディスクを登録します。IP アドレスを引き継ぐ場合は，クライアントアクセスポイントも登録します。

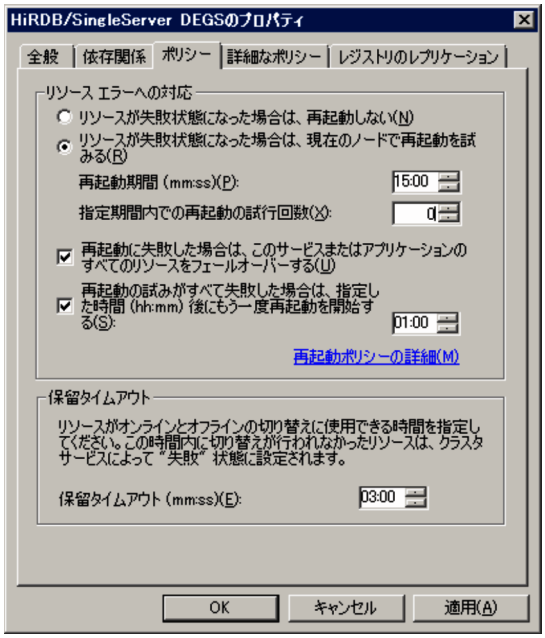


5. **[ポリシー]** タブを選択します。

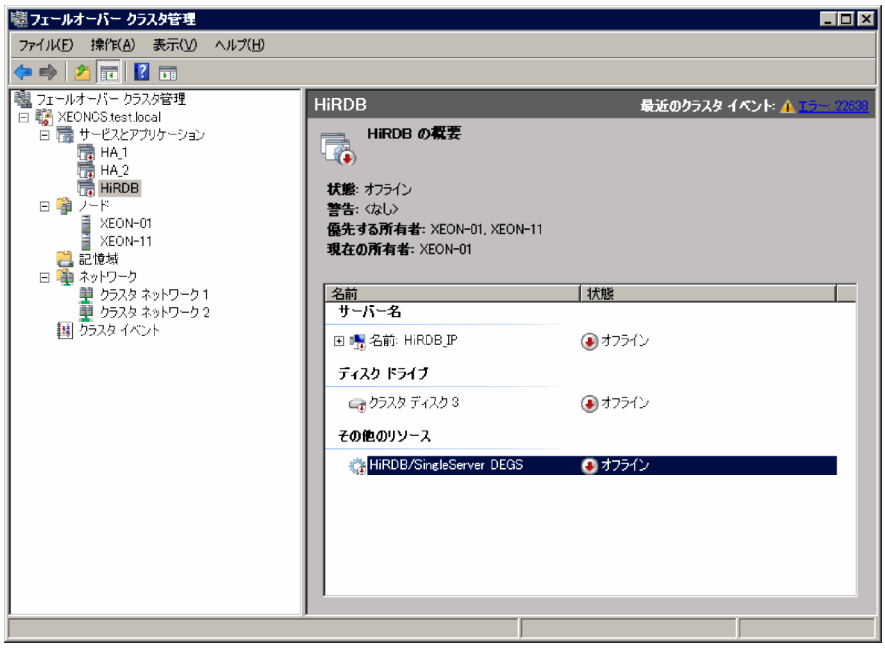
   [リソースが失敗状態になった場合は，現在のノードで再起動を試みる]：ラジオボタンをチェックします。

   [指定期間内での再起動の試行回数]：0 を指定します。

   上記以外の項目はデフォルトのままでかまいません。



6. **[OK]** をクリックすると，HiRDB のサービスが登録されます。



26.6.6　HiRDB のクラスタサービスの登録（モニタモードの場合だけ）

HiRDB のクラスタサービスを MSFC の汎用サービスに登録します。サービス名は **HiRDB/ClusterService** です。ただし，相互系切り替え構成などでマルチ HiRDB を使用する場合は，HiRDB をインストールしたときに指定した**セットアップ識別子**をサービス名に付けて指定します。

（例）セットアップ識別子が UNT1 の場合

HiRDB/ClusterService UNT1

［コントロールパネル］－**［サービス］**の**［サービス］**ダイアログでサービス名を確認してから，サービス名を指定してください。

HiRDB のクラスタサービスの登録の手順を次に示します。

〈手順〉

1. 「サービスの作成」で作成した MSFC のサービスを選択して右クリックします。**［リソースの追加］**－**［汎用サービス］**を選択し，**［新しいリソース ウィザード］**画面を表示します。表示される一覧から HiRDB のクラスタサービスを選択します。



2. **［次へ］**をクリックし，内容を確認してから**［OK］**をクリックします。
3. 登録した HiRDB のクラスタサービスを選択して右クリックします。**［プロパティ］**を選択し，登録したサービスのプロパティのダイアログを表示します。
4. **［依存関係］**タブを選択します。

   HiRDB のサービスを登録しない場合：

   リソース：共有ディスク，及びクライアントアクセスポイントを登録します。

   HiRDB のサービスを登録する場合：

   リソース：「HiRDB のサービスの登録（モニタモードの場合だけ）」で登録した HiRDB のサービスを登録します。



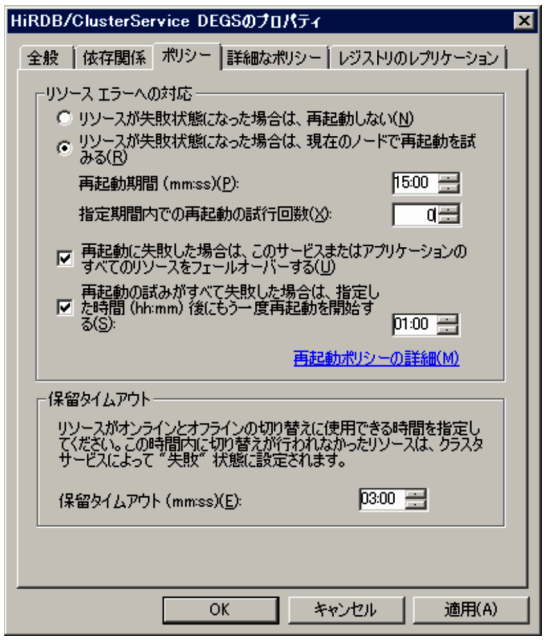
5. **[ポリシー]** タブを選択します。

   [リソースが失敗状態になった場合は，現在のノードで再起動を試みる]：ラジオボタンをチェックします。

   [指定期間内での再起動の試行回数]：0 を指定します。

   [保留タイムアウト]：最大値を指定します。

   上記以外の項目はデフォルトのままでかまいません。



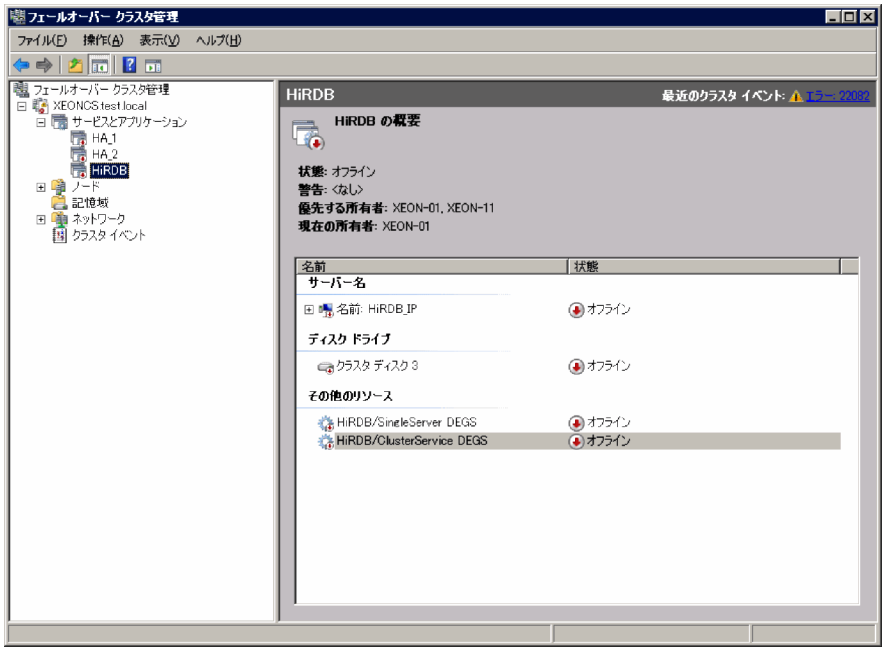
注意事項

［保留タイムアウト］について

障害発生時など意図しない系切り替えの場合は，システムログの出力量が多くなり，HiRDBの再開始に時間が掛かる場合があります。

HiRDB の再開始に掛かった時間が［**保留タイムアウト**］を超過すると，MSFC によって HiRDB の起動処理が打ち切られて HiRDB が起動できなくなります。そのため，［**保留タイムアウト**］には最大値を指定することをお勧めします。

6. ［**OK**］をクリックすると，HiRDB のクラスタサービスが登録されます。



26.7　MSFC に関する準備（Windows Server 2012/R2 以降の場合）

MSFC に関する準備には次に示す項目があります。ここでは，これらの項目の操作方法の概略を説明します。詳細については，MSFC のドキュメントなどを参照してください。

- 役割の作成
- 共有ディスクの登録
- リソースの登録（クライアントアクセスポイントの登録，HiRDB のサービスの登録，HiRDB のクラスタサービスの登録）

モニタモードで運用する場合

モニタモードで運用する場合は，この節の説明と MSFC のドキュメントを参照して環境設定を行ってください。

サーバモードで運用する場合

サーバモードで運用する場合は，次に示す箇所，マニュアル，又はドキュメントの説明を参照して環境設定を行ってください。

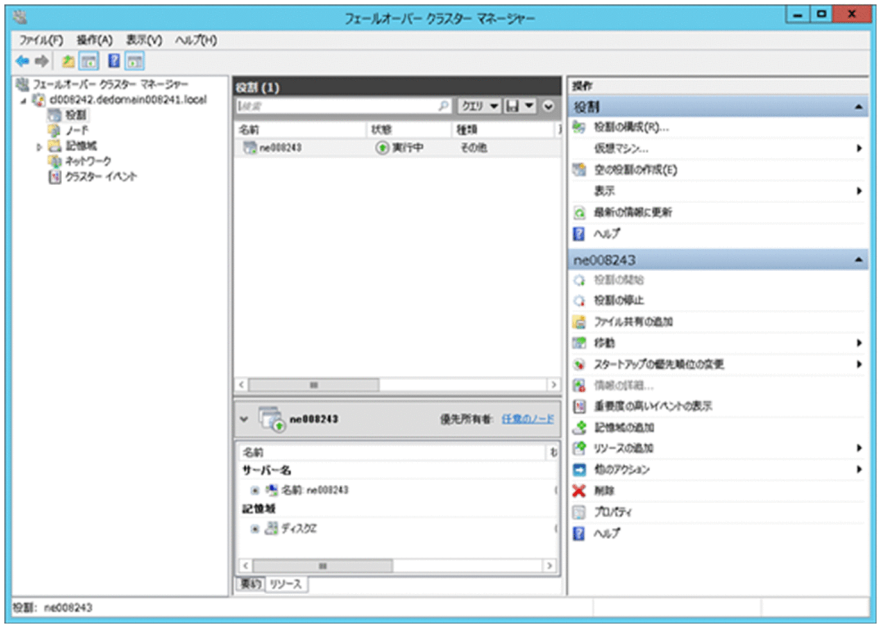
- この節の説明
- 「Hitachi HA Toolkit Extension に関する準備（サーバモード限定）」の説明
- マニュアル「Hitachi HA Toolkit」
- MSFC のドキュメント

26.7.1　役割の作成

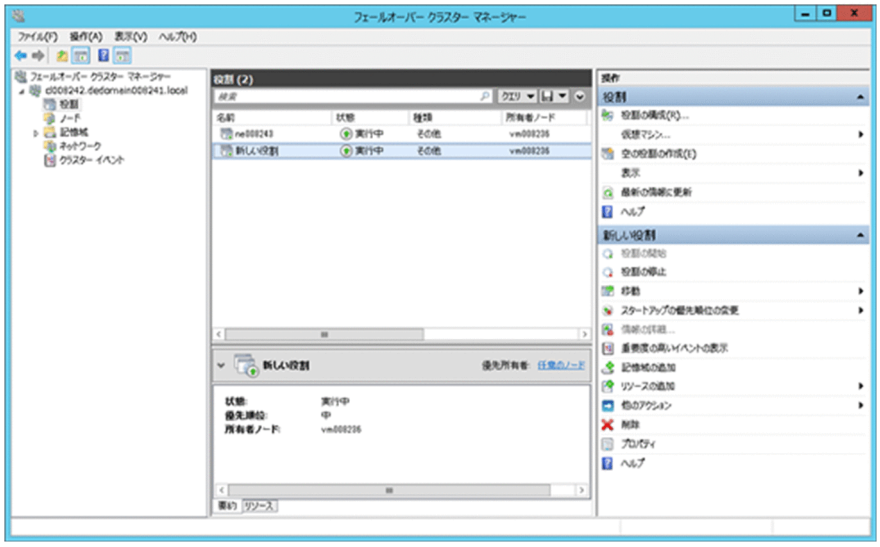
MSFC のフェールオーバークラスターマネージャーで，MSFC の役割を作成する必要があります。役割の作成手順を次に示します。

〈手順〉

1. **［スタート］**－**［フェールオーバークラスターマネージャー］**でフェールオーバークラスターマネージャーを起動します。



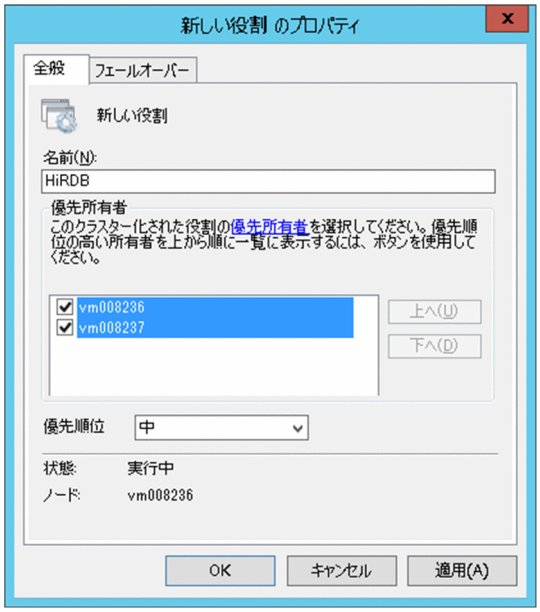
2. ツリーの［**役割**］を右クリックして，［**空の役割の作成**］を選択します。



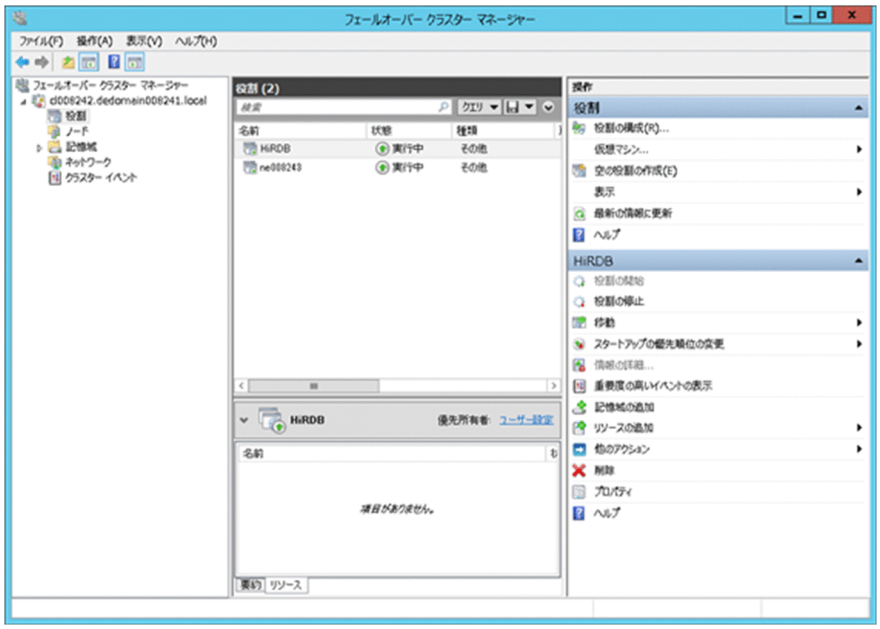
3. 作成された役割を選択して右クリックします。［**プロパティ**］を選択し，［**プロパティ**］ダイアログを表示し，［**全般**］タブを選択します。

   名前：役割の名前（任意）を指定します。

   優先する所有者：優先する所有者を指定します。



4. **[OK]** をクリックすると，指定した名前で役割が作成されます。



26.7.2　リソースの登録

(1)　モニタモードの場合のリソースの登録

モニタモードの場合に，MSFC の役割に登録するリソースを次の表に示します。

表 26–23　MSFC の役割に登録するリソース（モニタモードの場合）

リソースの種類	説明
共有ディスク	共有ディスクを登録します。

リソースの種類	説明
クライアントアクセスポイント	IP アドレスを引き継ぐ場合にクライアントアクセスポイントを登録します。
汎用サービス	HiRDB のサービスを登録します。HiRDB が異常終了又はユニットダウンした系で再開始できない場合に，系を切り替えるときに登録します。
	HiRDB のクラスタサービスを登録します。

なお，MSFC の役割に登録する各リソースの依存関係は，次に示す順番で起動されるように登録してください。

1. 共有ディスク及びクライアントアクセスポイント
2. HiRDB のサービス※
3. HiRDB のクラスタサービス

注※

HiRDB のサービスを登録しない場合は HiRDB のクラスタサービスが開始します。

(2)　サーバモードの場合のリソースの登録

サーバモードの場合に，MSFC の役割に登録するリソースを次の表に示します。

表 26–24　MSFC の役割に登録するリソース（サーバモードの場合）

リソースの種類	説明
共有ディスク	共有ディスクを登録します。
汎用サービス	Hitachi HA Toolkit Extension のサービスを登録します。 詳細については，マニュアル「Hitachi HA Toolkit」を参照してください。

なお，MSFC の役割に登録する各リソースの依存関係は，次に示す順番で起動されるように登録してください。

1. 共有ディスク
2. Hitachi HA Toolkit Extension のサービス

注意事項

サーバモードで Hitachi HA Toolkit Extension を使用する場合，HiRDB のクラスタサービスの起動は Hitachi HA Toolkit Extension の起動用バッチファイルから行います。このため，Hitachi HA Toolkit Extension を使用する場合は HiRDB のクラスタサービス及び HiRDB のサービスを登録しないでください。

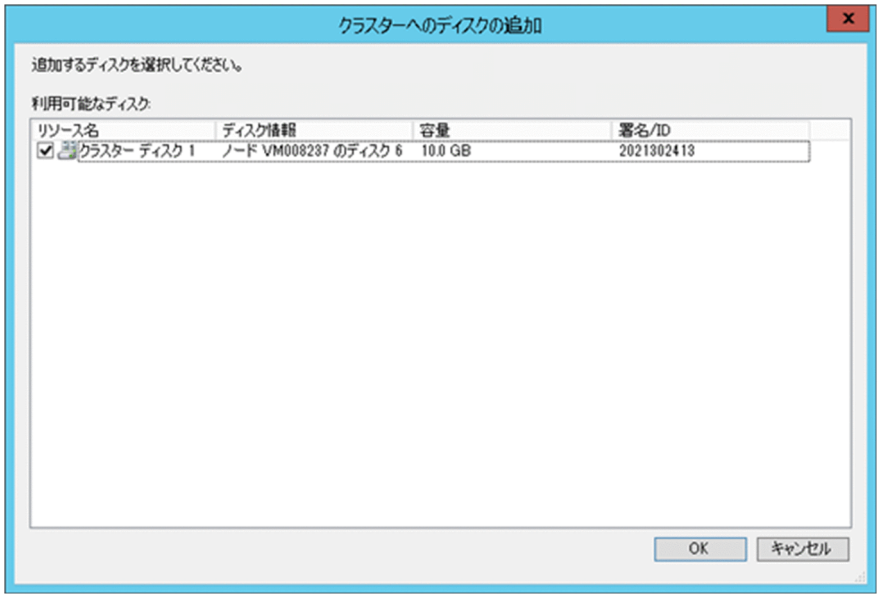
26.7.3　共有ディスクの登録

フェールオーバークラスターマネージャーで共有ディスクを MSFC の役割に登録します。

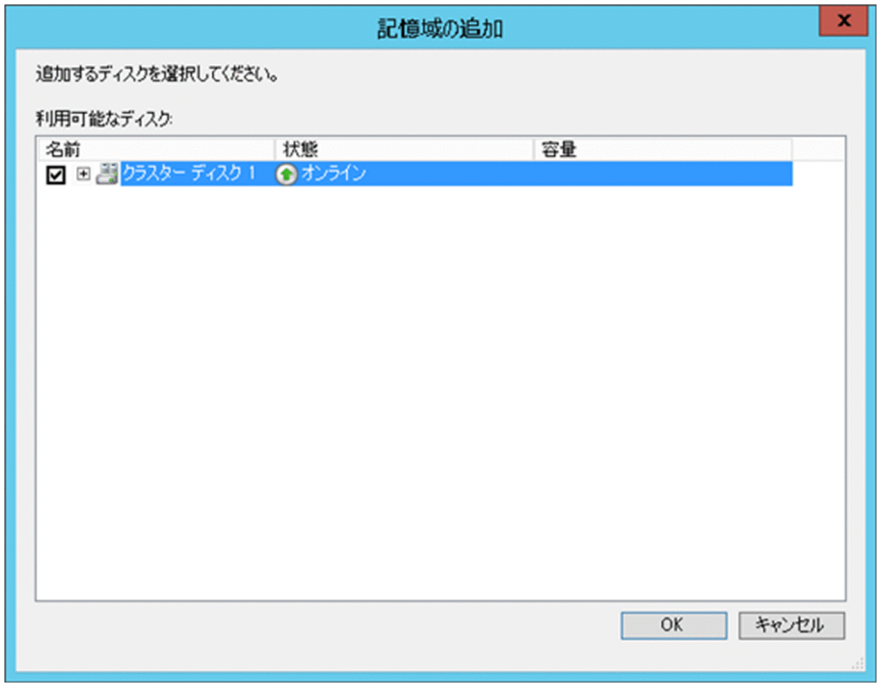
Hitachi HA Toolkit Extension を使用する場合は，HiRDB ファイルシステム領域を作成する共有ディスクがオンラインになった後に，HiRDB を監視する Hitachi HA Toolkit Extension のサービスがオンライン操作されるように，MSFC の依存関係を設定してください。共有ディスクの登録手順を次に示します。

〈手順〉

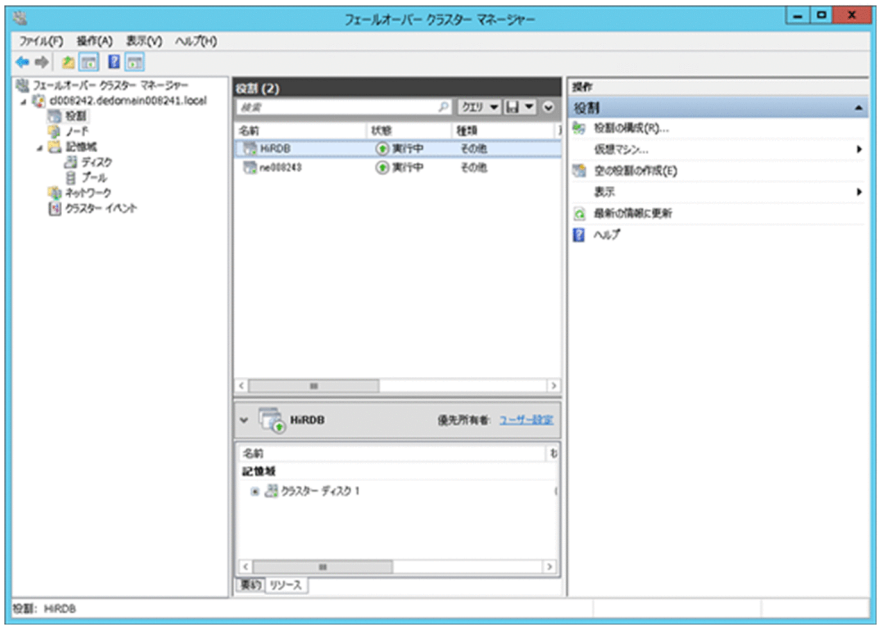
1. ツリーの**「記憶域」**－**「ディスク」**を選択して右クリックします。**［ディスクの追加］**を選択し，**［クラスターへのディスクの追加］**ダイアログで切り替え対象のディスクを選択します。



2. 「役割の作成」で作成した MSFC の役割を選択して右クリックします。**［記憶域の追加］**を選択し，**［記憶域の追加］**ダイアログで切り替え対象のディスク（手順 1 で追加したディスク）を選択します。



3. **[OK]** をクリックすると，選択したディスクが登録されます。



26.7.4　クライアントアクセスポイントの登録（モニタモードの場合だけ）

クライアントアクセスポイントの登録は，IP アドレスを引き継ぐ場合に行います。IP アドレスを引き継がない場合は，この操作をする必要はありません。

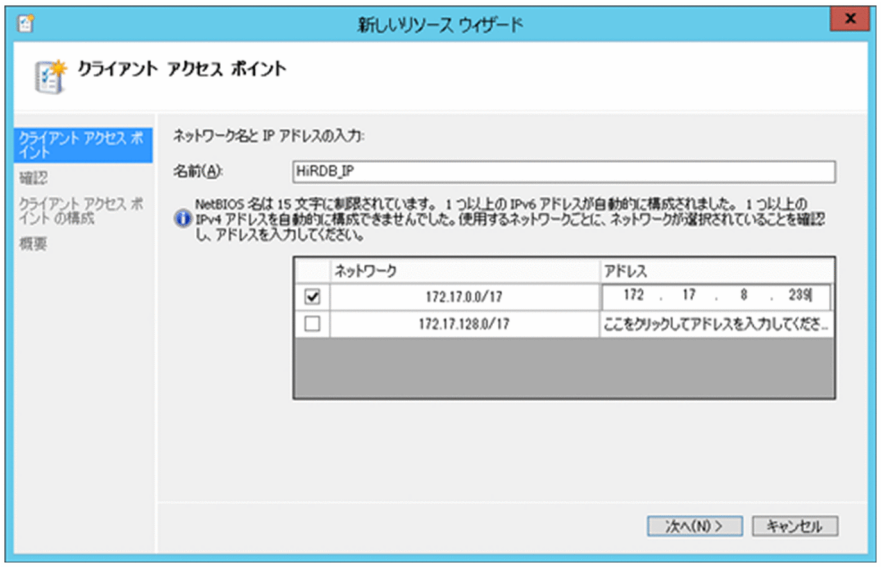
HiRDB が系切り替え機能で使用する**仮想ネットワーク名の IP アドレス**を登録します。なお，DNS サーバを使用しない場合は，仮想ネットワーク名，及びその IP アドレスを hosts ファイルに登録する必要があります。IP アドレスの登録手順を次に示します。

〈手順〉

1. 「役割の作成」で作成した MSFC の役割を選択して右クリックします。**［リソースの追加］**－**［クライアントアクセスポイント］**を選択し，**［新しいリソース ウィザード］**画面を表示します。

   名前：ネットワーク名を指定します。

   アドレス：引き継ぐ IP アドレスを指定します。



2. **［次へ］**をクリックし，**［確認画面］**を表示して指定内容を確認します。
3. **［次へ］**をクリックすると，クライアントアクセスポイントが登録されます。**［完了］**をクリックして登録処理を終了します。

26.7.5　HiRDB のサービスの登録（モニタモードの場合だけ）

HiRDB のサービスを MSFC の汎用サービスに登録すると，HiRDB が異常終了又はユニットダウンした系で再開始できない場合に，系を切り替えることができます。ただし，障害を検知してから系を切り替えるまでに時間が掛かります。系切り替え時間を短縮したい場合は，サーバモードを使用してください。

登録する HiRDB のサービス名は次のとおりです。

- **HiRDB/SingleServer**（HiRDB/シングルサーバの場合）
- **HiRDB/ParallelServer**（HiRDB/パラレルサーバの場合）

ただし，相互系切り替え構成などでマルチ HiRDB を使用する場合は，HiRDB をインストールしたときに指定した**セットアップ識別子**をサービス名に付けて指定します。

（例）セットアップ識別子が UNT1 の場合

HiRDB/SingleServerUNT1

HiRDB/ParallelServerUNT1

［コントロールパネル］－［管理ツール］－［サービス］の［サービス］ダイアログでサービス名を確認してから，サービス名を指定してください。

注意事項

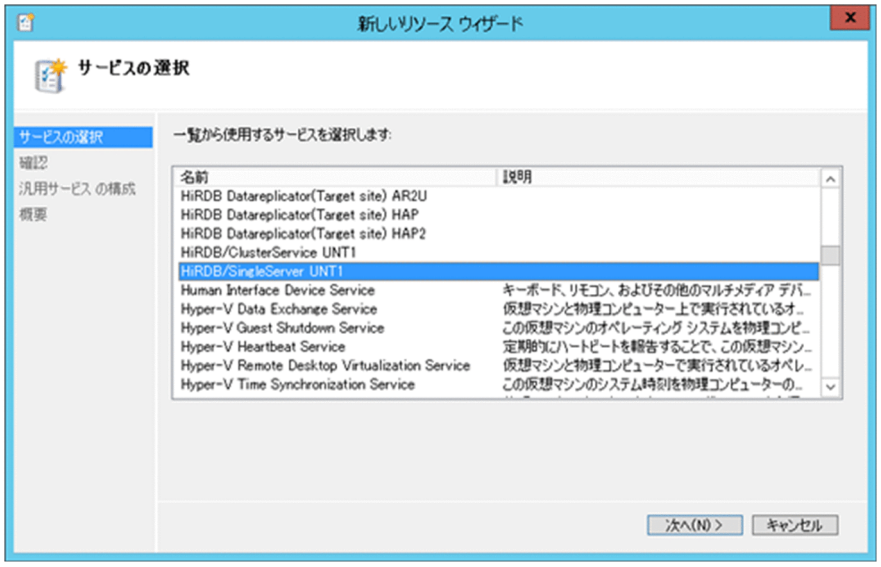
HiRDB/シングルサーバの場合のサービスの開始方法の設定について

HiRDB のサービスの開始方法の設定（Service Complete の設定）を manual にする必要があります。manual にしない場合は，HiRDB のサービスを MSFC に登録しないでください。

HiRDB のサービスの登録手順を次に示します。

〈手順〉

1. 「役割の作成」で作成した MSFC の役割を選択して右クリックします。［リソースの追加］－［汎用サービス］を選択し，［新しいリソース ウィザード］画面を表示します。表示される一覧から HiRDB のサービスを選択します。

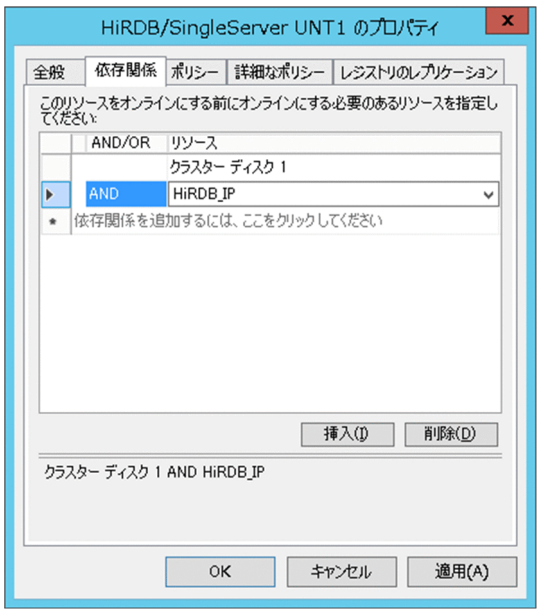


2. ［次へ］をクリックし，［確認画面］を表示して指定内容を確認します。



3. **[次へ]** をクリックすると，HiRDB のサービスが登録されます。**[完了]** をクリックして，登録処理を終了します。
4. 登録した HiRDB のサービスを選択して右クリックします。**[プロパティ]** を選択し，登録したサービスのプロパティのダイアログを表示します。
5. **[依存関係]** タブを選択します。

   リソース：共有ディスクを登録します。IP アドレスを引き継ぐ場合は，クライアントアクセスポイントも登録します。



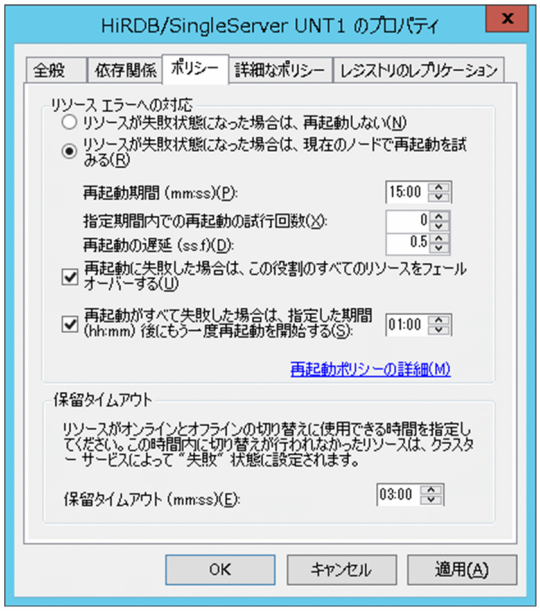
6. **[ポリシー]** タブを選択します。

   [リソースが失敗状態になった場合は，現在のノードで再起動を試みる]：ラジオボタンをチェックします。

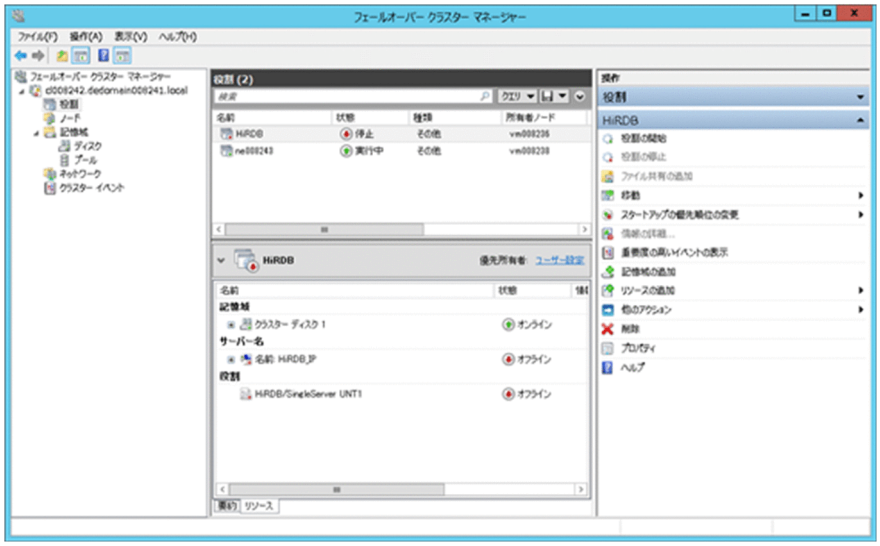
   [指定期間内での再起動の試行回数]：0 を指定します。

   上記以外の項目はデフォルトのままでかまいません。

   これによって，HiRDB が異常終了したときに系を切り替えることができます。



7. **[OK]** をクリックすると，HiRDB のサービスが登録されます。



26.7.6　HiRDB のクラスタサービスの登録（モニタモードの場合だけ）

HiRDB のクラスタサービスを MSFC の汎用サービスに登録します。サービス名は **HiRDB/ClusterService** です。ただし，相互系切り替え構成などでマルチ HiRDB を使用する場合は，HiRDB をインストールしたときに指定した**セットアップ識別子**をサービス名に付けて指定します。

（例）セットアップ識別子が UNT1 の場合

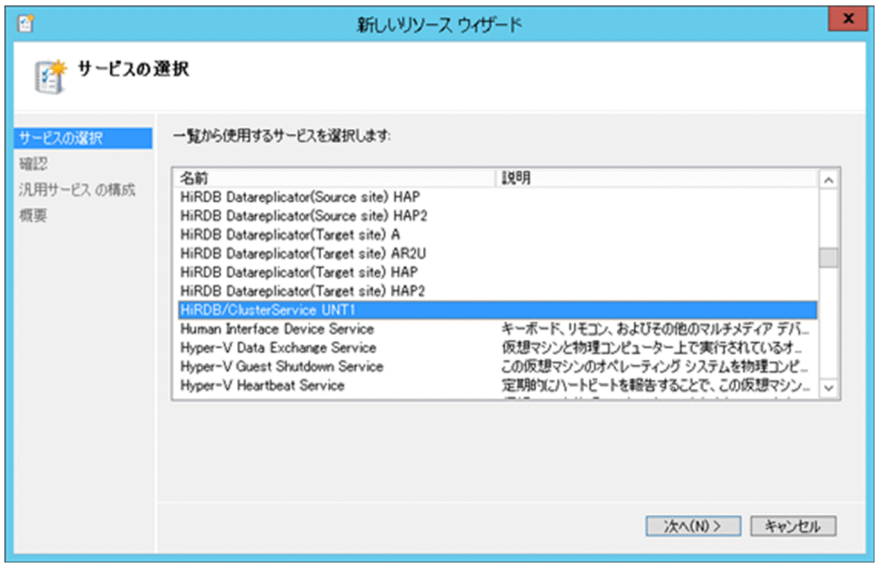
HiRDB/ClusterService UNT1

［コントロールパネル］－**［管理ツール］**－**［サービス］**の **［サービス］**ダイアログでサービス名を確認してから，サービス名を指定してください。

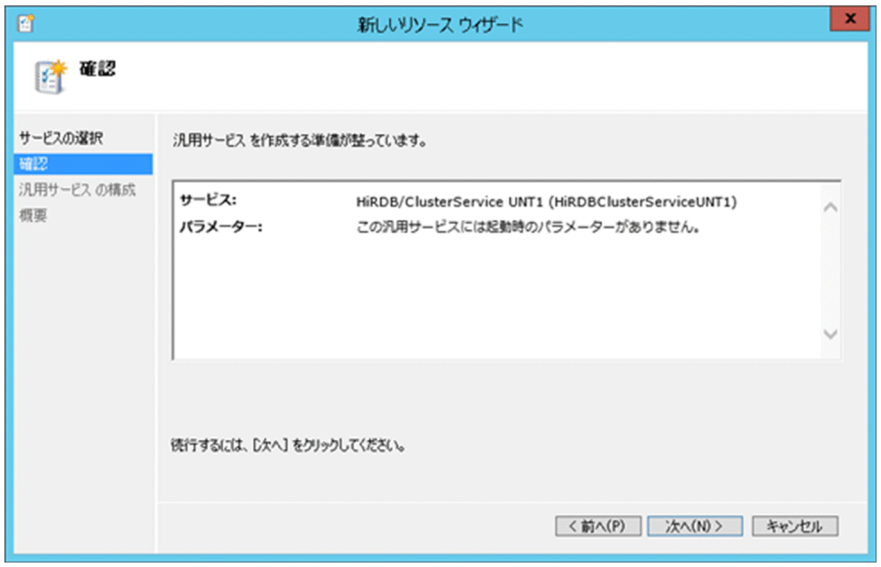
HiRDB のクラスタサービスの登録の手順を次に示します。

〈手順〉

1. 「役割の作成」で作成した MSFC の役割を選択して右クリックします。**［リソースの追加］**－**［汎用サービス］**を選択し，**［新しいリソース ウィザード］**画面を表示します。表示される一覧から HiRDB のクラスタサービスを選択します。



2. **［次へ］**をクリックし，**［確認画面］**を表示して指定内容を確認します。



3. **［次へ］**をクリックすると，HiRDB のクラスタサービスが登録されます。**［完了］**をクリックして，登録処理を終了します。
4. 登録した HiRDB のクラスタサービスを選択して右クリックします。**［プロパティ］**を選択し，登録したサービスのプロパティのダイアログを表示します。
5. **［依存関係］**タブを選択します。

   HiRDB のサービスを登録しない場合：

   リソース：共有ディスク，及びクライアントアクセスポイントを登録します。

   HiRDB のサービスを登録する場合：

リソース：「HiRDB のサービスの登録（モニタモードの場合だけ）」で登録した HiRDB のサービスを登録します。



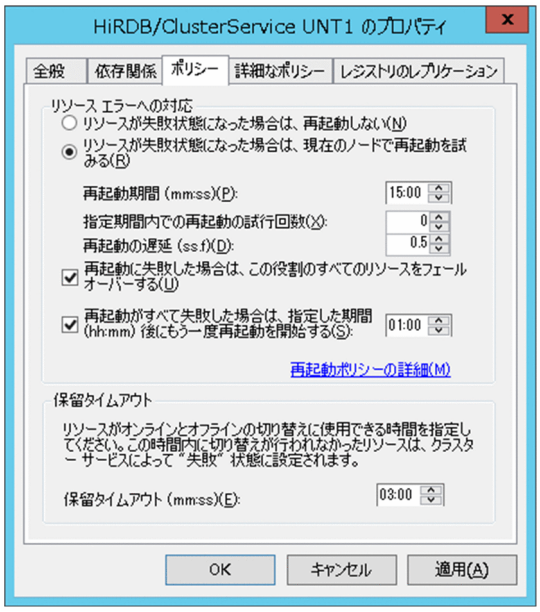
6. **[ポリシー]** タブを選択します。

   [リソースが失敗状態になった場合は，現在のノードで再起動を試みる]：ラジオボタンをチェックします。

   [指定期間内での再起動の試行回数]：0 を指定します。

   [保留タイムアウト]：最大値を指定します。

   上記以外の項目はデフォルトのままでかまいません。



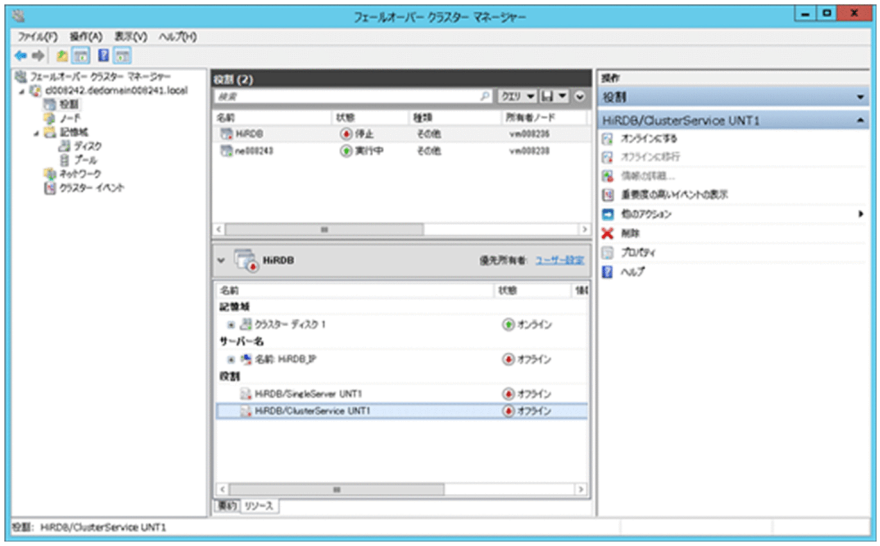
注意事項

［保留タイムアウト］について

障害発生時など意図しない系切り替えの場合は，システムログの出力量が多くなり，HiRDB の再開始に時間が掛かる場合があります。

HiRDB の再開始に掛かった時間が［**保留タイムアウト**］を超過すると，MSFC によって HiRDB の起動処理が打ち切られて HiRDB が起動できなくなります。そのため，［**保留タイムアウト**］には最大値を指定することをお勧めします。

7. ［**OK**］をクリックすると，HiRDB のクラスタサービスが登録されます。



26.8　Hitachi HA Toolkit Extension に関する準備（サーバモード限定）

Hitachi HA Toolkit Extension を使用している場合にこの節をお読みください。ここでは，Hitachi HA Toolkit Extension の sysdef 定義ファイル及び servers 定義ファイルのオペランドで，HiRDB に関連するオペランドの指定値の目安について説明します。Hitachi HA Toolkit Extension の環境設定方法の詳細については，マニュアル「Hitachi HA Toolkit」を参照してください。

なお，スタンバイ型系切り替え及び 1：1 スタンバイレス型系切り替えでは，ユニットごとに動作環境を設定し，影響分散スタンバイレス型系切り替えでは，サーバごとに動作環境を設定します。

26.8.1　sysdef 定義ファイル

(1)　servmax オペランド

このオペランドは Hitachi HA Toolkit Extension のバージョンが 01-04 以降の場合に指定できます。

Hitachi HA Toolkit Extension の系切り替え対象のサーバで，一つのサーバマシン上で同時に実行又は待機状態として起動できるサーバの最大数を 16 又は 64 で指定します。

16：同時に起動できるサーバの最大数を 16 に設定します。

64：同時に起動できるサーバの最大数を 64 に設定します。

一つのサーバマシン上で系の切り替え単位であるサーバ数が 16 を超える場合は 64 を指定してください。

Hitachi HA Toolkit Extension の系切り替え対象のサーバ数は，HiRDB 以外の系切り替え対象製品を含めた値です。HiRDB では系切り替え対象のサーバ数を次の計算で求めます。

- 一つのサーバマシン上で動作するスタンバイ型系切り替えの実行系ユニット数と待機系ユニット数の合計数
- 一つのサーバマシン上で動作する 1:1 スタンバイレス型系切り替えの正規 BES ユニット数と代替 BES ユニット数の合計数
- 一つのサーバマシン上で動作する影響分散スタンバイレス型系切り替えのホスト BES 数とゲスト用領域数の合計数

26.8.2　servers 定義ファイル

(1)　name オペランド

スタンバイ型系切り替え機能の場合は HiRDB 運用ディレクトリ名を絶対パス名で指定します。HiRDB/パラレルサーバの場合は該当するユニットの HiRDB 運用ディレクトリ名を絶対パス名で指定します。

1：1 スタンバイレス型系切り替え機能の場合は，HiRDB 識別子（pd_system_id オペランドの指定値）と正規 BES ユニットのユニット識別子（pd_unit_id オペランドの指定値）を「/」で組み合わせて指定します。HiRDB 識別子が DB01 で，正規 BES ユニットのユニット識別子が UNT1 の場合は次のように指定します。

```
DB01/UNT1
```

影響分散スタンバイレス型系切り替えの場合は，HiRDB 識別子とサーバ名を「/」で組み合わせて指定します。HiRDB 識別子が DB01 で，サーバ名が BES1 の場合は次のように指定します。

```
DB01/BES1
```

(2) switchtype オペランド

このオペランドにはサーバ障害を検出したときの処理を指定します。

switch：

HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了すると系を切り替えて切り替え先の系で HiRDB を再開始します。

1：1 スタンバイレス型系切り替え機能の場合，代替 BES ユニットに作成する代替部用の servers 定義ファイルには switch を指定することをお勧めします。switch を指定すると，代替中に代替 BES ユニットで障害が発生した場合，代替部から正規 BES ユニットに系が切り替わるため，再開始後の代替 BES ユニットの負荷を減らせます。

影響分散スタンバイレス型系切り替え機能の場合，ゲスト BES 用の servers 定義ファイルには switch を指定することをお勧めします。switch を指定すると，ゲスト BES が稼働中のユニットで障害が発生した場合，ゲスト BES がほかのユニットに切り替わるため，再開始後の負荷を分散できます。

restart：

障害が発生した系で HiRDB（HiRDB/パラレルサーバの場合はユニット）を再開始します。障害が発生した系で HiRDB を再開始できないときに，系を切り替えて切り替え先の系で HiRDB を再開始します（KFPS00715-E メッセージが出力された時点で系を切り替えます）。

1：1 スタンバイレス型系切り替え機能の場合，代替 BES ユニットに作成する代替部用の servers 定義ファイルには restart を指定しないことをお勧めします。restart を指定すると，代替中に代替 BES ユニットで障害が発生した場合，ユニットの再開始後も代替 BES ユニットで正規 BES ユニットの代替処理を継続するため，再開始後の代替 BES ユニットの負荷が減りません。

影響分散スタンバイレス型系切り替え機能の場合，ゲスト BES 用の servers 定義ファイルには restart を指定しないことをお勧めします。restart を指定指定すると，ゲスト BES の稼働中のユニットで障害が発生した場合，ユニットの再開始後にゲスト BES が処理を継続するため，再開始後の負荷を分散できません。

manual：

HiRDB（HiRDB/パラレルサーバの場合はユニット）を再開始できなくても自動的に系を切り替えません。

ポイント

1：1 スタンバイレス型系切り替え機能で相互代替構成の場合，代替 BES ユニットと代替部の switchtype オペランドには同じ値を指定してください。

影響分散スタンバイレス型系切り替え機能の場合，HA グループ内の全サーバの switchtype オペランドには同じ値を指定してください。

(3) actcommand オペランド

HiRDB 開始用のバッチファイル名を" "で囲んで指定します。このオペランドを指定すると，クラスタソフトウェアでサーバのオンライン操作をしたときに，ここで指定したバッチファイルが実行されて HiRDB が開始します。このオペランドを省略する場合は，pdstart コマンドで HiRDB を開始してください。標準セットアップを指定して HiRDB をインストールした場合の HiRDB 開始用のバッチファイルの例を次に示します。

（例）

```
net start HiRDBClusterService
if “%ERRORLEVEL%”==”0” goto end
call インストールディレクトリ¥SAMPLE¥SAMPLECONF¥HiRDBCMD.BAT
cd /D %PDDIR%
call %PDDIR%¥CONF¥pdshastr.bat
```

1：1 スタンバイレス型系切り替え機能の場合，待機系（代替部）で actcommand オペランドを指定する必要があるときは，開始用のバッチファイルに pdstart -q -c コマンドを指定してください。

影響分散スタンバイレス型系切り替え機能の場合，actcommand オペランドは指定できません。

(4) termcommand オペランド

HiRDB 終了用のバッチファイル名を" "で囲んで指定します。このオペランドを指定すると，クラスタソフトウェアでサーバのオフライン操作をしたときに，ここで指定したバッチファイルが実行されて HiRDB が終了します。このオペランドを省略する場合は，pdstop コマンドで HiRDB を終了してください。標準セットアップを指定して HiRDB をインストールした場合の HiRDB 終了用のバッチファイルの例を次に示します。

（例）

```
call インストールディレクトリ¥SAMPLE¥SAMPLECONF¥HiRDBCMD.BAT
call %PDDIR%¥CONF¥pdshaend.bat※
start net stop HiRDBClusterService
```

注※

- HiRDB の終了処理時間を短縮するために HiRDB を強制終了しています。その後，停止処理を行う HiRDBClusterService サービスも同じファイルを実行するため，コマンドエラーメッセージが出力されることがあります。

- HiRDB クライアントの接続状態と pdstop コマンドのオプションによっては，pdstop コマンドがクライアントの接続解除を待ち合わせます。このため，HiRDB の終了処理に時間が掛かったり，HiRDB を終了できなかったりすることがあります。したがって，HiRDB 終了用のバッチファイルに指定する pdstop コマンドのオプションに注意してください。

1：1 スタンバイレス型系切り替え機能の場合，代替部の termcommand オペランドに指定する終了用のバッチファイルには pdstop -z -c コマンドを指定してください。

影響分散スタンバイレス型系切り替え機能の場合，HA グループ内の全サーバの termcommand オペランドに指定する終了用のバッチファイルには pdstop -z -s コマンドを指定してください。

26.9　CLUSTERPRO に関する準備

クラスタソフトウェアに CLUSTERPRO を使用している場合にこの節をお読みください。

CLUSTERPRO の環境設定方法の詳細については，CLUSTERPRO のマニュアルを参照してください。

注意事項

- HiRDB が CLUSTERPRO での動作を保証するクラスタ構成は，1：1 接続構成（1：1 系切り替え構成，及び相互系切り替え構成），1：N 接続構成（マルチスタンバイ構成），及び影響分散スタンバイレス系切り替え構成となります。

  モニタモードで運用する場合

  モニタモードで運用する場合は，この節の説明と CLUSTERPRO のドキュメントを参照して環境設定をしてください。

  サーバモードで運用する場合

  サーバモードで運用する場合は，次に示す箇所，マニュアル，又はドキュメントの説明を参照して環境設定をしてください。

  ・この節の説明

  ・「Hitachi HA Toolkit Extension に関する準備（サーバモード限定）」の説明

  ・マニュアル「Hitachi HA Toolkit」

  ・CLUSTERPRO のドキュメント
- ディスク構成は次のどちらかを選択してください。

  ・共有ディスク構成

  ・ミラーディスク構成
- グループリソース，モニタリソースの作成は HiRDB が停止した状態で行ってください。各種リソースの作成に失敗した場合は，CLUSTERPRO のマニュアルを参照し各種リソースを再作成してください。

26.9.1　グループリソースの作成

CLUSTERPRO が系切り替えを行う単位をフェイルオーバグループといいます。フェイルオーバグループとはクラスタシステム内の独立した業務を実行するために必要な資源（グループリソース）の集まりのことです。

HiRDB のモニタモード，又はサーバモードで CLUSTERRO を用いて系切り替えを実行する場合，CLUSTERPRO のフェイルオーバグループにグループリソースを作成してください。

HiRDB/パラレルサーバの場合は，ユニットごとに作成したフェイルオーバグループ配下にグループリソースを作成してください。

HiRDB の設定に必要なグループリソースを次に示します。

- モニターモードの場合

グループリソース名	概要	作成要否
スクリプトリソース	HiRDB の起動，停止を行うスクリプトを実行するために作成します。	◎
ディスクリソース	共有ディスク構成を構築する場合に作成します。	○
ミラーディスクリソース	ミラーディスク構成を構築する場合に作成します。	○
フローティング IP リソース	オンプレミス環境でフローティング IP アドレスを使用する場合に作成します。	△
AWS 仮想 IP リソース	Amazon Web Services 環境で，仮想 IP アドレスを使用する場合に作成します。	△
AWS セカンダリ IP アドレスリソース	Amazon Web Services 環境で，AWS セカンダリ IP アドレスを使用する場合に作成します。	△
AWS Elastic IP リソース	Amazon Web Services 環境で，ElasticIP アドレスを使用する場合に作成します。	△
AWS DNS リソース	Amazon Route 53 に登録した FQDN を使用する場合に作成します。	△
Azure プローブポートリソース	Azure ロードバランサーを用いて，仮想 IP アドレスを使用する場合に作成します。	△
Azure DNS リソース	Azure DNS に登録した FQDN を使用する場合に作成します。	△
Oracle Cloud DNS リソース	Oracle Cloud DNS に登録した FQDN を使用する場合に作成します。	△

（凡例）

◎：必ず作成してください。

○：構築環境に合わせて必ず作成してください。

△：必要に応じて作成してください。

- サーバモードの場合

グループリソース名	概要	作成要否
サービスリソース	Hitachi HA Toolkit Extension のサービスを登録します。	◎
ディスクリソース	共有ディスク構成を構築する場合に作成します。	○
ミラーディスクリソース	ミラーディスク構成を構築する場合に作成します。	○

（凡例）

◎：必ず作成してください。

○：構築環境に合わせて必ず作成してください。

26.9.2　モニタリソースの作成

モニタリソースとは，グループリソースのうち，指定されたリソースを監視するリソースです。

モニタリソースは，監視対象の異常を検出した場合に，グループリソースの再起動やフェイルオーバなどを行います。そのため，HiRDB の稼働を監視する場合にモニタリソースを作成することで，HiRDB 異常終了時に再起動やフェイルオーバができます。

HiRDB の稼働を監視する場合に作成するモニタリソースを次に示します。

モニタリソース名	概要	作成要否
カスタム監視	HiRDB ユニットの稼働監視を行う場合に作成します。	△

（凡例）

△：必要に応じて作成してください。

そのほかのモニタリソースについては，「グループリソースの作成」で示したグループリソースを作成した後，CLUSTERPRO 側が対応するモニタリソースを自動，又は手動で作成が必要なリソースがあります。

なお，HiRDB での設定は必要ありません。

26.9.3　IP アドレスに関連するリソースの作成

オンプレミス環境上で HiRDB の IP アドレス，又はクライアント接続用の IP アドレスを引き継ぐ場合，CLUSTERPRO のグループリソースのうち，フローティング IP リソースを作成してください。HiRDB に関係する設定項目を次の表に示します。

表 26–25　フローティング IP リソースに指定する項目の値

プロパティ	項目	入力・選択値
情報	タイプ	「フローティング IP リソース」を選択します。
依存関係	－	「既定の依存関係に従う」にチェックを入れます。
詳細	IP アドレス	使用するフローティング IP アドレスを指定します。

Amazon Web Services，Microsoft Azure，又は Oracle Cloud Infrastructure 環境で HiRDB の IP アドレス，又はクライアント接続用の IP アドレスを引き継ぐ場合の詳細は，「クラスタソフトウェアが CLUSTERPRO の場合」を参照してください。

26.9.4　ディスクに関連するリソースの作成

HiRDB ファイルシステムを配置するディスクを CLUSTERPRO に設定します。

HiRDB ファイルシステムは次の領域に配置できます。

ディスク形態	HiRDB ファイルシステム領域の作成ファイル
共有ディスク	通常ファイル
ミラーディスク	通常ファイル

(1) HiRDB ファイルシステム領域を共有ディスク上の通常ファイルに作成する場合

CLUSTERPRO のグループリソースのうち，ディスクリソースを作成してください。HiRDB に関係する設定項目を次の表に示します。

表 26–26　ディスクリソースに指定する項目の値

プロパティ	項目	入力・選択値
情報	タイプ	「ディスクリソース」を選択します。
依存関係	–	「既定の依存関係に従う」にチェックを入れます。
詳細	ドライブ文字	使用するディスクのドライブ文字を指定します。
	起動可能サーバ	現用系のホスト名を選択して［追加］を選択します。 追加したサーバのパーティション一覧が［パーティションの選択］ダイアログボックスに表示されるため，使用するパーティションを選択します。 その後，予備系のホスト名を選択して同様の操作を行います。

(2) HiRDB ファイルシステム領域をミラーディスク上の通常ファイルに作成する場合

CLUSTERPRO グループリソースのうち，ミラーディスクリソースを作成してください。HiRDB に関係する設定項目を次の表に示します。

表 26–27　ミラーディスクリソースに指定する項目の値

プロパティ	項目	入力・選択値
情報	タイプ	「ミラーディスクリソース」を選択します。
依存関係	–	「既定の依存関係に従う」にチェックを入れます。
詳細	起動可能なサーバ	現用系のホスト名を選択して［追加］を選択します。追加したサーバのパーティション一覧が［パーティションの選択］ダイアログボックスに表示されるため，使用するパーティションを選択します。その後，予備系のホスト名を選択して同様の操作を行います。
調整	初期ミラー構築を行う	チェックを入れます。

プロパティ	項目	入力・選択値
	モード	HiRDB/シングルサーバ，及び 1 ユニット構成の HiRDB/パラレルサーバ：同期，又は非同期※を選択します。 複数ユニット構成の HiRDB/パラレルサーバ：同期を選択します。

注※

非同期を選択する場合は，一つのミラーディスクリソース上にミラーリング対象の HiRDB ユニット内の HiRDB ファイルをすべて配置してください。なお，非同期を選択すると，更新された情報が系切り替えによって欠落することがあります。

26.9.5　スクリプト実行に関連するリソースの作成

HiRDB ユニットの開始，停止，稼働監視を行うスクリプトを準備します。CLUSTERPRO のグループリソースのうち，スクリプトリソース，及びカスタム監視リソースにスクリプトを登録することで，HiRDB ユニットの起動，停止，又は稼働を監視できます。

(1)　HiRDB のスクリプト

次のスクリプトを作成します。「HiRDB/シングルサーバの場合」又は「HiRDB/パラレルサーバの場合」の例を参照してください。

- HiRDB(ユニット)起動コマンドのスクリプト
- HiRDB(ユニット)停止コマンドのスクリプト
- HiRDB ユニットの稼働監視を行うスクリプト

HiRDB ユニットの稼働監視を行うスクリプトは現用系，予備系サーバの両方に格納する必要があります。

HiRDB/シングルサーバの場合

●HiRDB（ユニット）起動コマンドのスクリプト（hirdb_start.bat）

```
@rem ##################### start HiRDB ########################################
@net start HiRDBClusterService.................................1
@if "%ERRORLEVEL%"=="0" @goto end
  @call "C:¥HiRDB_S¥SAMPLE¥sampleconf¥HiRDBCMD.BAT".......2
  @call %PDDIR%¥bin¥pdstart
:end
```

〔説明〕

1. HiRDBClusterService サービスを起動した延長で，pdstart コマンドを実行します。この設定例は HiRDB を標準インストールした場合の例です。HiRDB を識別子付きセットアップでインストールした場合は，対応する識別子付きのサービス名に修正してください。

2. HiRDBClusterService サービスの起動処理に失敗した場合，HiRDB インストールフォルダ配下の環境変数設定用バッチファイルを実行して，環境変数を設定します。HiRDB をインストールしたフォルダのパス名に変更してください。

●HiRDB（ユニット）停止コマンドのスクリプト（hirdb_stop.bat）

```
@rem ##################### stop HiRDB ########################################
@call "C:¥HiRDB_S¥SAMPLE¥sampleconf¥HiRDBCMD.BAT".....1
@call %PDDIR%¥bin¥pdstop -f -q.................................2
```

〔説明〕

1. HiRDB インストールフォルダ配下の環境変数設定用バッチファイルを実行して環境変数を設定します。HiRDB をインストールしたフォルダのパス名に変更してください。
2. 系が短時間で切り替わるように，終了コマンドには強制終了（pdstop -f -q）を使用してください。

●HiRDB ユニットの稼働監視を行うスクリプト（hirdb_pdls.bat）

```
@rem ##################### check HiRDB ######################################

@call "C:¥HiRDB_S¥SAMPLE¥sampleconf¥HiRDBCMD.BAT".....1
@call %PDDIR%¥bin¥pdls -d ust
```

〔説明〕

1. HiRDB インストールフォルダ配下の環境変数設定用バッチファイルを実行して環境変数を設定します。HiRDB をインストールしたフォルダのパス名に変更してください。

HiRDB/パラレルサーバの場合

注意事項

HiRDB/パラレルサーバでスクリプトリソースを開始して HiRDB ユニットを起動する場合，HiRDB ユニットが起動完了する前にスクリプトリソースが開始状態となります。HiRDB ユニットのグループリソースが系切り替えする場合も同様です。HiRDB ユニットの起動が完了したかどうか，又は系切り替えが完了したかどうかはイベントログの KFPS05110-I メッセージを確認してください。KFPS05110-I メッセージが出力されていれば，ユニットの起動，又は系切り替えは完了しています。

また，イベントログにエラーメッセージが出力され，ユニットの起動が失敗している場合は，起動失敗の要因を取り除いた後，HiRDB システム，又は HiRDB ユニットが開始中の場合は HiRDB を強制終了（pdstop -f，又は pdstop -z コマンドで終了）してください。その後，CLUSTERPRO のコマンド操作，又は Cluster WebUI による GUI 操作で，スクリプトリソースを停止して再度起動してください。

●HiRDB（ユニット）起動コマンドのスクリプト（hirdb_start.bat）

```
@rem ##################### start HiRDB ########################################
@net start HiRDBClusterService.................................1
@if "%ERRORLEVEL%"=="0" @goto end
  @call "C:¥HiRDB_P¥SAMPLE¥sampleconf¥HiRDBCMD.BAT".......2
  @call %PDDIR%¥bin¥pdstart -q
:end
```

〔説明〕

1. HiRDBClusterService サービスを起動した延長で，pdstart -q コマンドを実行します。この設定例は HiRDB を標準インストールした場合の例です。HiRDB を識別子付きセットアップでインストールした場合は，対応する識別子付きのサービス名に修正してください。
2. HiRDBClusterService サービスの起動処理に失敗した場合，HiRDB インストールフォルダ配下の環境変数設定用バッチファイルを実行して，環境変数を設定します。HiRDB をインストールしたフォルダのパス名に変更してください。

●HiRDB（ユニット）停止コマンドのスクリプト（hirdb_stop.bat）

```
@rem ##################### stop HiRDB #########################################
@call "C:¥HiRDB_P¥SAMPLE¥sampleconf¥HiRDBCMD.BAT".....1
@call %PDDIR%¥bin¥pdstop -z -q.................................2
```

〔説明〕

1. HiRDB インストールフォルダ配下の環境変数設定用バッチファイルを実行して環境変数を設定します。HiRDB をインストールしたフォルダのパス名に変更してください。
2. 系が短時間で切り替わるように，終了コマンドには強制終了（pdstop -z -q）を使用してください。

●HiRDB ユニットの稼働監視を行うスクリプト（hirdb_pdls.bat）

```
@rem ##################### check HiRDB ########################################
@call "C:¥HiRDB_P¥SAMPLE¥sampleconf¥HiRDBCMD.BAT".....1
@call %PDDIR%¥bin¥pdls -d ust
```

〔説明〕

1. HiRDB インストールフォルダ配下の環境変数設定用バッチファイルを実行して環境変数を設定します。HiRDB をインストールしたフォルダのパス名に変更してください。

(2)　スクリプトリソースの作成

CLUSTERPRO のグループリソースのうち，スクリプトリソースを作成してください。

HiRDB に関係する設定項目を次の表に示します。

プロパティ	項目	入力・選択値
情報	タイプ	「スクリプトリソース」を選択します。

プロパティ	項目	入力・選択値	
依存関係	−	「既定の依存関係に従う」にチェックを入れます。	
復旧動作	活性リトライしきい値	0 を指定します。	
	フェイルオーバしきい値	0 を指定します。	
詳細	Start Script	Start Script を選択の上，［置換］を選択します。 アップロードするファイルの選択画面が表示されるので，HiRDB の開始スクリプトのファイルを選択します。	
	Stop Script	Stop Script を選択の上，［置換］を選択します。 アップロードするファイルの選択画面が表示されるので，HiRDB の停止スクリプトのファイルを選択します。	
調整	開始	−	同期を選択します
		タイムアウト	pd_system_complete_wait_time オペランド以上の値を指定してください。
		正常な戻り値	0 を指定してください。
		待機系サーバで実行する	チェックを入れません。
	終了	−	同期を選択します。
		タイムアウト	変更不要です。
		待機系サーバで実行する	チェックを入れません。
	実行ユーザ	指定不要です。	

注意事項

HiRDB の開始処理で pd_system_complete_wait_time オペランド値の時間が経過しても HiRDB が起動しない場合，イベントログに KFPS05078-I メッセ―ジを出力してスクリプトリソースがエラー終了します。HiRDB としては開始処理を続行するため，HiRDB システム又は HiRDB ユニットを強制終了（pdstop -f コマンド，又は pdstop -z コマンドで終了）してください。その後，スクリプトリソースも停止してください。スクリプトリソース再起動時に同様の理由でエラーにならないよう次の値を変更してスクリプトリソースを開始してください。

- pd_system_complete_wait_time オペランド値
- スクリプトリソース 開始のタイムアウト値

(3)　カスタム監視リソースの作成

スクリプトリソースを作成した後，HiRDB ユニットの稼働を監視する場合は，CLUSTERPRO のモニタリソースのうち，カスタム監視リソースを作成してください。カスタム監視リソース作成時に HiRDB ユニッ

トの稼働を監視するスクリプトを登録することで，HiRDB ユニットが異常終了した場合，HiRDB ユニットの異常を検知して，グループの再起動，又はフェイルオーバを実施できます。

なお，カスタム監視リソースを作成して，グループの再起動，又はフェイルオーバを実施する場合は，pd_mode_conf に MANUAL2 を指定してください。

カスタム監視リソースについて，HiRDB に関係する設定項目を次の表に示します。

表 26–28　カスタム監視リソースに指定する項目の値

プロパティ	項目	入力・選択値
情報	タイプ	「カスタム監視」を選択します。
監視（共通）	タイムアウト発生時にリトライしない	チェックを入れません。
	リトライ回数	1 以上の任意の値を指定します。
	監視タイミング	活性時を選択します。
	対象リソース	［参照］ボタンを選択して，HiRDB（ユニット）を起動するスクリプトリソースを選択します。
監視（固有）	−	ユーザアプリケーションを選択します。
	ファイル	HiRDB の稼働監視スクリプトのファイルパス名を指定します。
	監視タイプ	同期を選択します。
	正常な戻り値	「0,4」を指定します。
	警告戻り値	「0,4」を指定します。
	クラスタ停止時に活性時監視を待ち合わせる	チェックを入れます。
回復動作	回復対象	［参照］ボタンを選択して，HiRDB（ユニット）の起動，停止を行うフェイルオーバグループを指定します。

26.9.6　HiRDB クライアントに関連する設定項目

フローティング IP リソースを作成した場合，HiRDB サーバへの接続方式ごとに指定する HiRDB クライアントの環境変数を次の表に示します。

表 26–29　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	フローティング IP リソースに指定した IP アドレス，又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	フローティング IP リソースに指定したシステムマネジャユニットの IP アドレス，又は対応するホスト名を指定してください。

環境変数		指定値
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–30　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	フローティング IP リソースに指定したシステムマネジャユニットの IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST	フローティング IP アドレスリソースに指定したフロントエンドサーバの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–31　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数	指定値	
PDHOST	HiRDB/シングルサーバの場合	フローティング IP リソースに指定した IP アドレス又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	フローティング IP リソースに指定したシステムマネジャユニットの IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST	フローティング IP アドレスリソースに指定したフロントエンドサーバの IP アドレス，又は対応するホスト名を指定してください。	
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT	次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値	

環境変数	指定値	
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE	「PC」を指定してください。	

26.9.7　Hitachi HA Toolkit Extension に関連するリソースの作成（サーバモードの場合だけ）

サーバモードの系切り替えを行う場合，Hitachi HA Toolkit Extension に関連するリソースを CLUSTERPRO に設定します。

(1)　サービスリソースの作成

CLUSTERPRO のグループリソースのうち，サービスリソースを作成し，Hitachi HA Toolkit Extension のサービスを登録します。

詳細については，マニュアル「Hitachi HA Toolkit」を参照してください。

(2)　モニタリソースの設定

「サービスリソースの作成」で作成したサービスリソースごとに生成するモニタリソースの設定値を変更してください。

詳細については，マニュアル「Hitachi HA Toolkit」を参照してください。

26.10　MSFC 及び DataKeeper に関する準備

DataKeeper による複製ディスクを使用している場合の，MSFC 及び DataKeeper に関する準備方法について説明します。

MSFC 及び DataKeeper の環境設定方法の詳細については，MSFC，DataKeeper のマニュアルを参照してください。

26.10.1　DataKeeper ジョブ，及び DataKeeper ボリュームリソースの作成

(1)　1:1 系切り替え，相互系切り替え構成の場合

現用系と予備系で，同一の情報が参照できる必要があります。DataKeeper のデータレプリケーション機能によって，対象のボリューム間で同期を取ります。

DataKeeper でレプリケーションする場合，DataKeeper ボリュームリソースを作成する必要があります。現用系ユニットごとに DataKeeper ジョブを作成し，同期が必要なボリューム間のミラーを DataKeeper ジョブに作成することで，DataKeeper ボリュームリソースを作成できます。DataKeeper ジョブ作成時に設定する項目を次の表に示します。

表 26–32　DataKeeper ジョブに設定する項目（1:1 系切り替え，相互系切り替え構成の場合）

項番	項目		入力・選択する内容
1	ジョブ名/ジョブの説明		任意の内容を入力
2	ソースボリュームの設定	サーバ	現用系のホスト名を選択
3		IP アドレス	現用系のレプリケーションで使用する IP アドレスを選択
4		ボリューム	現用系のレプリケーション対象ボリュームを選択
5	ターゲットボリュームの設定	サーバ	予備系のホスト名を選択
6		IP アドレス	予備系のレプリケーションで使用する IP アドレスを選択
7		ボリューム	予備系のレプリケーション対象ボリュームを選択
8	圧縮度		任意
9	レプリケーション方式		HiRDB/シングルサーバ及び 1 ユニット構成の HiRDB/パラレルサーバ： 同期又は非同期※ 複数ユニット構成の HiRDB/パラレルサーバ： 同期
10	最大帯域		任意

注※

非同期を選択する場合は，一つの DataKeeper ボリューム上にレプリケーション対象のユニット内の HiRDB ファイルをすべて配置してください。なお，非同期を選択すると，更新された情報が系切り替えによって欠落することがあります。

DataKeeper ジョブ作成後，MSFC に DataKeeper ボリュームリソースを自動登録するかどうかの確認ダイアログが表示されます。［はい］をクリックして DataKeeper ボリュームリソースを MSFC に登録してください。

(2) マルチスタンバイ構成，影響分散スタンバイレス型系切り替え構成の場合

現用系（又はホスト BES）とすべての予備系（又はゲスト BES）で，同一の情報が参照できる必要があります。DataKeeper のデータレプリケーション機能によって，対象となるすべてのボリューム間で同期を取ります。

DataKeeper でレプリケーションする場合，DataKeeper ボリュームリソースを作成する必要があります。現用系のユニット（又はホスト BES）ごとに DataKeeper ジョブを作成し，対象のユニット（又はホスト BES）で同期する必要があるボリューム間のすべてのミラーを DataKeeper ジョブに作成することで，DataKeeper ボリュームリソースを作成できます。

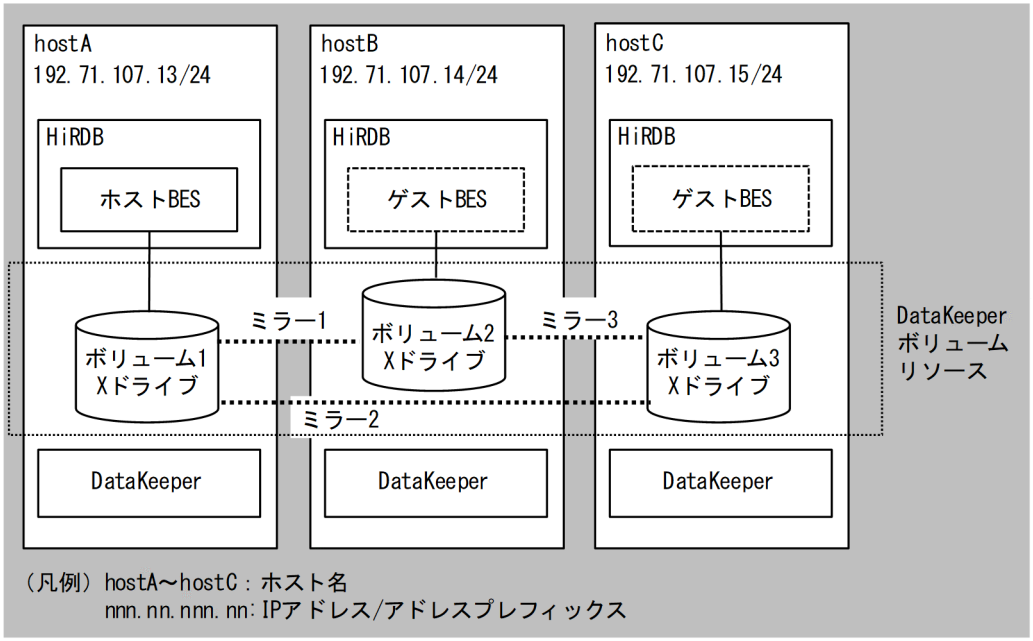
DataKeeper ボリュームリソースの作成手順を次に示します。

〈手順〉

1. 現用系のユニット（又はホスト BES）に対して DataKeeper ジョブを一つ作成し，DataKeeper ジョブに現用系のユニット（又はホスト BES）から予備系のユニット（又はゲスト BES）へのミラーを登録して，DataKeeper ボリュームリソースを作成します。
2. 手順 1.で作成した DataKeeper ジョブに，残りのミラーを登録します。

DataKeeper ボリュームリソースの作成例を次の図で示します。ここではボリューム 1 からボリューム 3 で同期を取る DataKeeper ボリュームリソースを作成します。

図 26–46　影響分散スタンバイレス構成での DataKeeper ボリュームリソースの例（hostA のホスト BES に対して，hostB と hostC にゲスト BES を配置）



DataKeeper ジョブ作成時に設定する項目を次の表に示します。ここで現用系（又はホスト BES）から予備系（又はゲスト BES）のミラーを一つ作成します。例ではホスト BES と hostB のゲスト BES の間のミラー 1 を作成します。

表 26–33　DataKeeper ジョブに設定する項目（マルチスタンバイ構成，影響分散スタンバイレス型系切り替え構成の場合）

項番	項目		入力・選択する内容	図の例での入力・選択する内容
1	ジョブ名/ジョブの説明		任意の内容を入力	ボリューム 1
2	ソースボリュームの設定	サーバ	現用系又はホスト BES のホスト名を選択	hostA
3		IP アドレス	現用系又はホスト BES のレプリケーションで使用する IP アドレスを選択	192.71.107.13/24
4		ボリューム	現用系又はホスト BES のレプリケーション対象ボリュームを選択	X
5	ターゲットボリュームの設定	サーバ	予備系又はゲスト BES のホスト名を選択	hostB
6		IP アドレス	予備系又はゲスト BES のレプリケーションで使用する IP アドレスを選択	192.71.107.14/24
7		ボリューム	予備系又はゲスト BES のレプリケーション対象ボリュームを選択	X
8	圧縮度		任意	None

項番	項目	入力・選択する内容	図の例での入力・選択する内容
9	レプリケーション方式	HiRDB/シングルサーバ及び 1 ユニット構成の HiRDB/パラレルサーバ： 同期又は非同期※ 複数ユニット構成の HiRDB/パラレルサーバ： 同期	同期
10	最大帯域	任意	0（無制限）

注※

非同期を選択する場合は，一つの DataKeeper ボリューム上にレプリケーション対象のユニット内の HiRDB ファイルをすべて配置してください。なお，非同期を選択すると，更新された情報が系切り替えによって欠落することがあります。

DataKeeper ジョブ作成後，MSFC に DataKeeper ボリュームリソースを自動登録するかどうかの確認ダイアログが表示されます。［はい］をクリックして DataKeeper ボリュームリソースを MSFC に登録してください。

予備系（又はゲスト BES）が複数の場合，DataKeeper ジョブ作成時に作成しなかった現用系（又はホスト BES）から予備系（又はゲスト BES）へのミラーを，DataKeeper ジョブに追加します。例ではホスト BES と hostC のゲスト BES の間のミラー 2 を次の表に示す指定で作成しています。

表 26–34　DataKeeper ジョブにミラーを追加作成するときに設定する項目

項番※	項目		図中の例での入力・選択する内容
1	ソースボリュームの設定	サーバ	hostA
2		IP アドレス	192.71.107.13/24
3		ボリューム	X
4	ターゲットボリュームの設定	サーバ	hostC
5		IP アドレス	192.71.107.15/24
6		ボリューム	X
7	圧縮度		None
8	レプリケーション方式		同期
9	最大帯域		0（無制限）

注※

1 から 9 の各項目は，表「DataKeeper ジョブに設定する項目（マルチスタンバイ構成，影響分散スタンバイレス型系切り替え構成の場合）」の項番 2 から 10 に対応しています。

ミラーを追加すると，予備系（又はゲスト BES）のボリューム間のミラーを設定するための追加情報ダイアログが表示されます。予備系（又はゲスト BES）のボリューム間のすべてのミラーを追加します。追加情報ダイアログに設定する項目を以下に示します。例では hostB のゲスト BES と hostC のゲスト BES の間のミラー 3 を作成します。

表 26–35　追加情報に設定する項目

項番	項目	入力・選択する内容	図中の例での入力・選択する内容
1	IP アドレス	レプリケーションで使用する IP アドレスを選択	192.71.107.14/24 192.71.107.15/24
2	ミラータイプ	HiRDB/シングルサーバ及び 1 ユニット構成の HiRDB/パラレルサーバ： 同期又は非同期※ 複数ユニット構成の HiRDB/パラレルサーバ： 同期	同期

注※

非同期を選択する場合は，一つの DataKeeper ボリューム上にレプリケーション対象のユニット内の HiRDB ファイルをすべて配置してください。なお，非同期を選択すると，更新された情報が系切り替えによって欠落することがあります。

予備系（又はゲスト BES）が三つ以上の場合，現用系（又はホスト BES）からすべての予備系（又はゲスト BES）へのミラー及びすべての予備系（又はゲスト BES）間のミラーを DataKeeper ジョブに追加してください。

26.10.2　DataKeeper ボリュームリソースの登録

(1)　モニタモードの場合

(a)　MSFC に HiRDB のサービスを登録している場合

HiRDB のサービスが登録された役割に「DataKeeper ジョブ，及び DataKeeper ボリュームリソースの作成」で作成した DataKeeper ボリュームリソースを登録してください。DataKeeper ボリュームリソースがオンラインになった後に，HiRDB のサービスがオンライン操作されるように， HiRDB のサービスの依存関係に DataKeeper ボリュームリソースを追加してください。

(b)　MSFC に HiRDB のサービスを登録していない場合

HiRDB のクラスタサービスが登録された役割に「DataKeeper ジョブ，及び DataKeeper ボリュームリソースの作成」で作成した DataKeeper ボリュームリソースを登録してください。DataKeeper ボリュームリソースがオンラインになった後に，HiRDB のクラスタサービスがオンライン操作されるように， HiRDB のクラスタサービスの依存関係に DataKeeper ボリュームリソースを追加してください。

(2)　サーバモードの場合

Hitachi HA Toolkit Extension のサービスが登録された役割に「DataKeeper ジョブ，及び DataKeeper ボリュームリソースの作成」で作成した DataKeeper ボリュームリソースを登録してください。DataKeeper ボリュームリソースがオンラインになった後に，HiRDB を監視する Hitachi HA Toolkit Extension のサービスがオンライン操作されるように，Hitachi HA Toolkit Extension のサービスの依存関係に DataKeeper ボリュームリソースを追加してください。

26.11　パブリッククラウド環境に関する準備

ここでは，次のパブリッククラウド環境を使用している場合の準備方法について説明します。

- Amazon Web Services
- Microsoft Azure
- Oracle Cloud Infrastructure

26.11.1　クラスタソフトウェアが MSFC の場合

(1)　動作環境

Amazon Web Services 環境上にクラスタ構成を適用する場合，次の製品が必要です。

- Windows Server Failover Clustering
- Hitachi HA Toolkit Extension（サーバモードの場合）
- DataKeeper for Windows Cluster Edition

また，Amazon EC2 インスタンスごとに次のストレージを配置する必要があります。

- Amazon EBS（Amazon Elastic Block Store）

Microsoft Azure 環境上にクラスタ構成を適用する場合，次の製品が必要です。

- Windows Server Failover Clustering
- Hitachi HA Toolkit Extension（サーバモードの場合）
- DataKeeper for Windows Cluster Edition

また，各 Azure VM ごとに次のストレージを配置する必要があります。

- Azure Storage

(2)　クラスタ構成概要

Amazon Web Services 環境上及び Microsoft Azure 環境上にクラスタ構成を適用する場合の概要を次の図に示します。

図 26–47　Amazon Web Services 環境上のクラスタ構成概要図

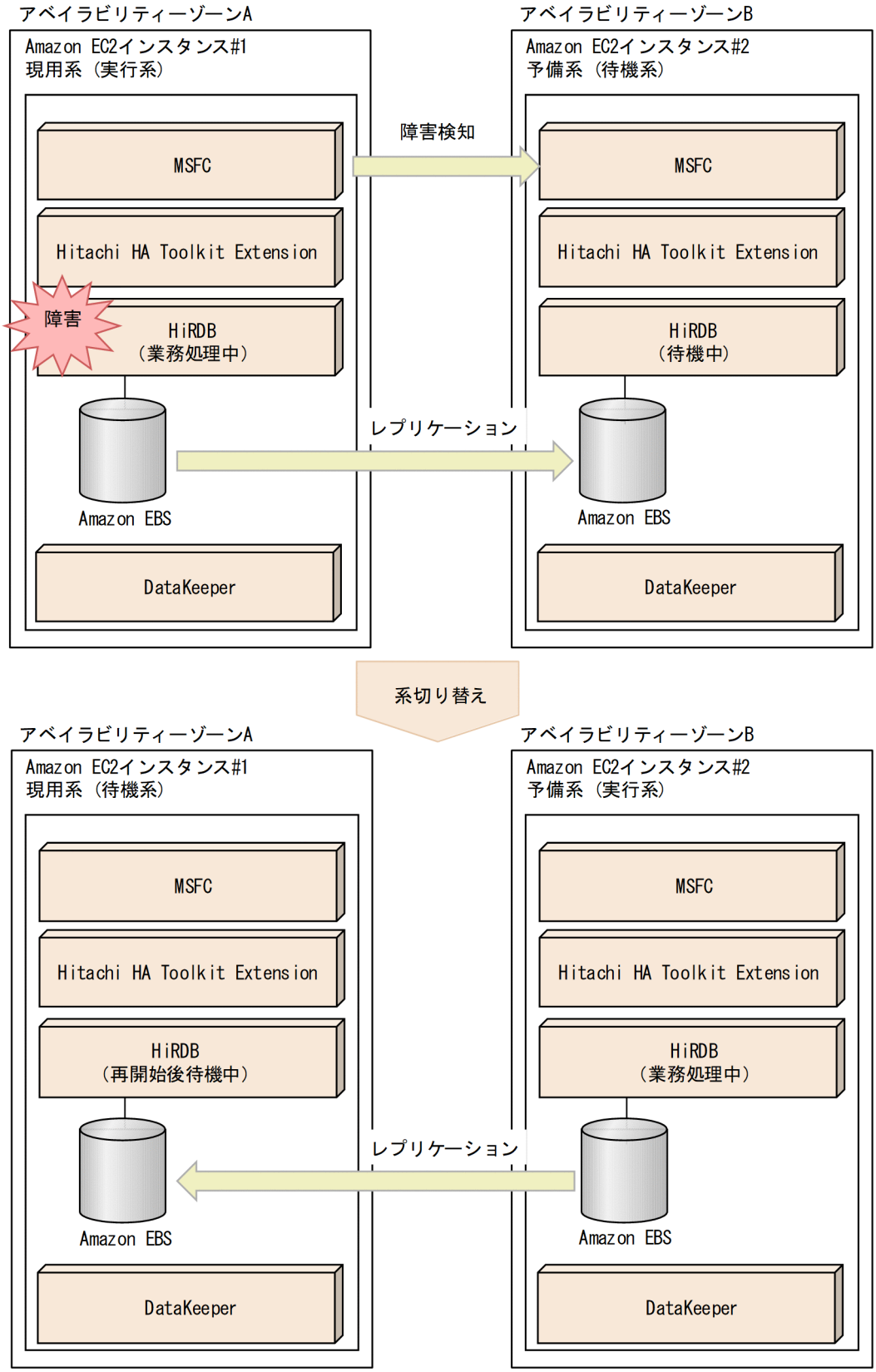
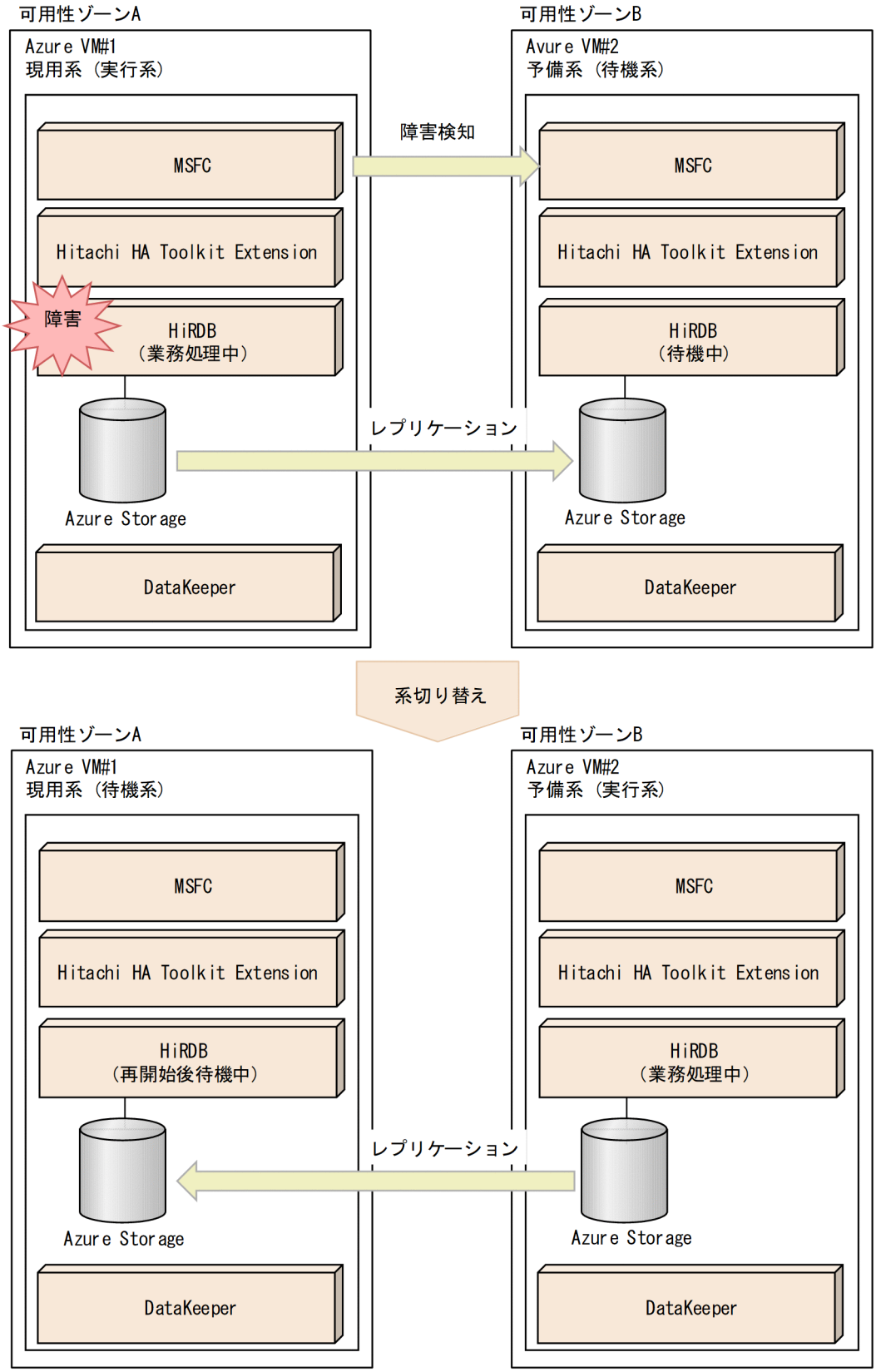


図 26–48　Microsoft Azure 環境上のクラスタ構成概要図



〔説明〕

業務処理中の実行系に障害が発生すると，待機系に障害の発生が通知されて系が切り替わり，待機系が実行系になって業務処理を続行します。

実行系のストレージ（Amazon EBS 又は Azure Storage）に対して更新すると，DataKeeper は更新内容を待機系に送信し，待機系のストレージにレプリケーションします。そのため，実行系の障害によって系が切り替わった場合，障害発生直前の状態から業務を続行できます。障害が発生して系が切り替わった場合は，系切り替え後の待機系で HiRDB を起動してください。

26.11.2　クラスタソフトウェアが CLUSTERPRO の場合

(1)　動作環境

Amazon Web Services 環境上にクラスタ構成を適用する場合，次の製品が必要です。

- CLUSTERPRO X
- Hitachi HA Toolkit Extension（サーバモードの場合）

また，Amazon EC2 インスタンスごとに次のストレージを配置する必要があります。

- Amazon EBS（Amazon Elastic Block Store）

Microsoft Azure 環境上にクラスタ構成を適用する場合，次の製品が必要です。

- CLUSTERPRO X
- Hitachi HA Toolkit Extension（サーバモードの場合）

また，各 Azure VM ごとに次のストレージを配置する必要があります。

- Azure Storage※

  注※

  Azure 共有ディスクを使用する場合は，各インスタンスから Azure 共有ディスクを共有してください。

Oracle Cloud Infrastructure 環境上にクラスタ構成を適用する場合，次の製品が必要です。

- CLUSTERPRO X
- Hitachi HA Toolkit Extension（サーバモードの場合）

また，インスタンスごとに次のストレージを配置する必要があります。

- ブロック・ボリューム

(2)　クラスタ構成概要

Amazon Web Services 環境上，Microsoft Azure 環境上，及び Oracle Cloud Infrastructure 環境上にクラスタ構成を適用する場合の概要を次の図に示します。

図 26–49　Amazon Web Services 環境上の共有ディスクを使用したクラスタ構成概要図

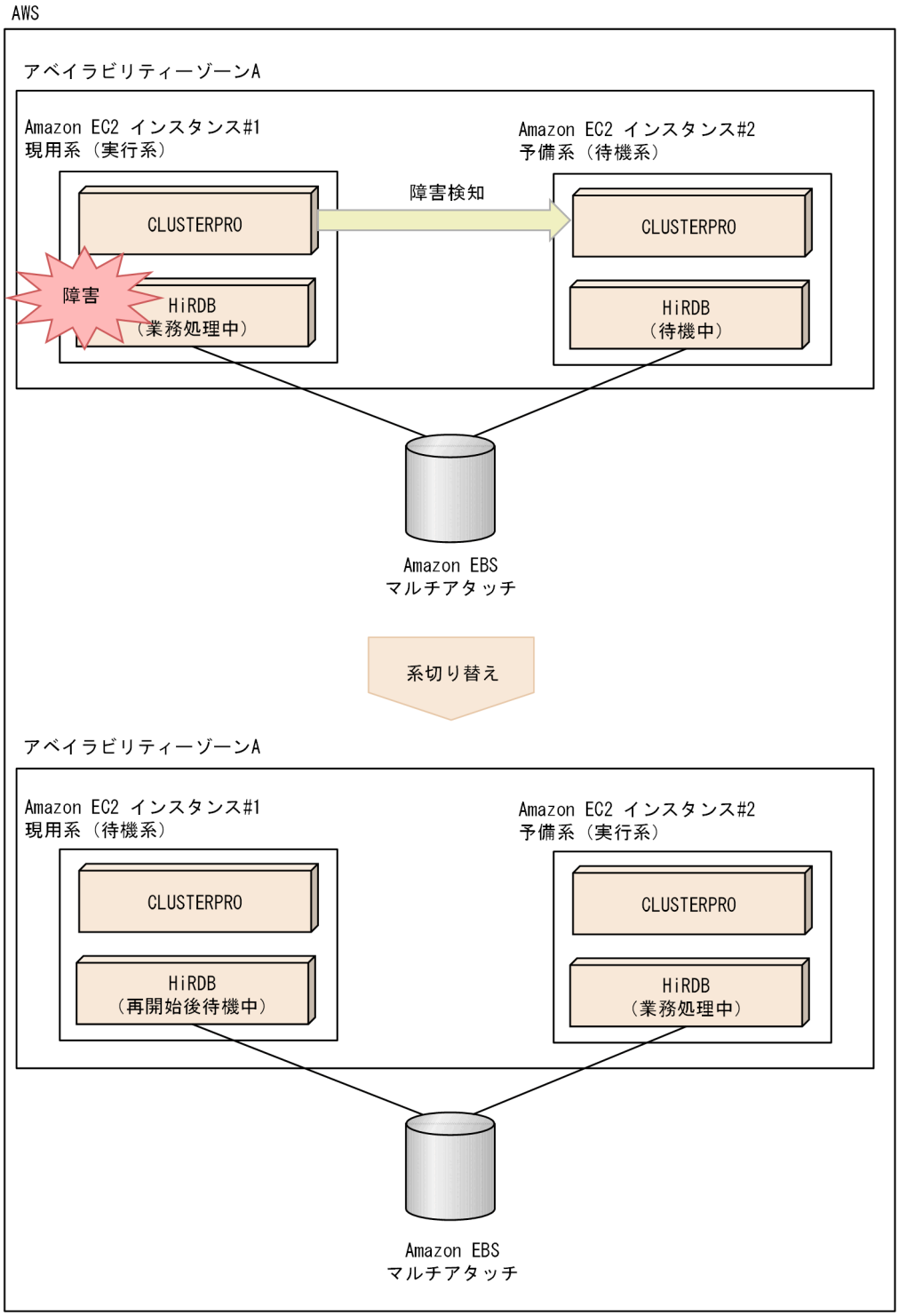


図 26–50　Microsoft Azure 環境上の共有ディスクを使用したクラスタ構成概要図

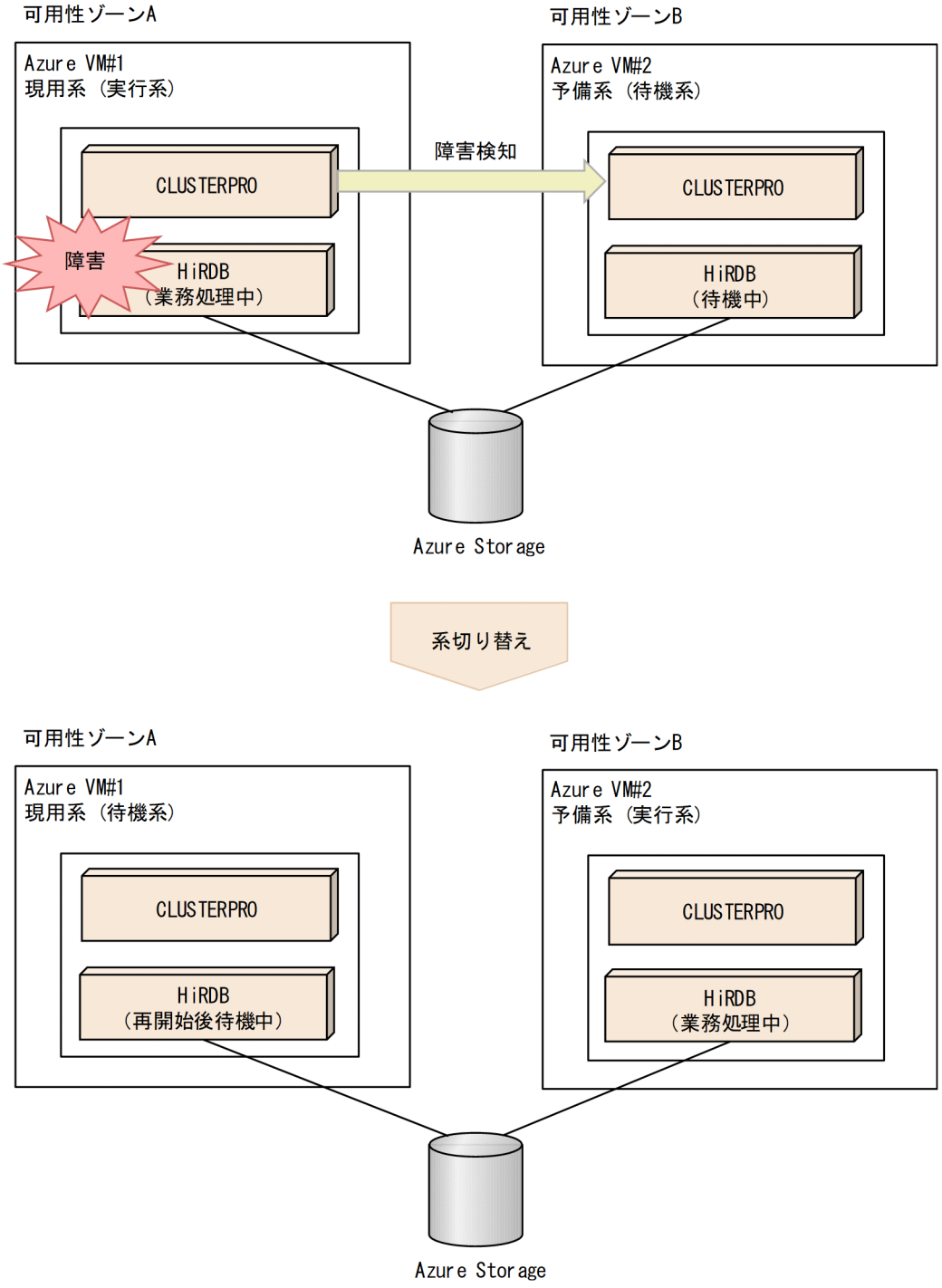
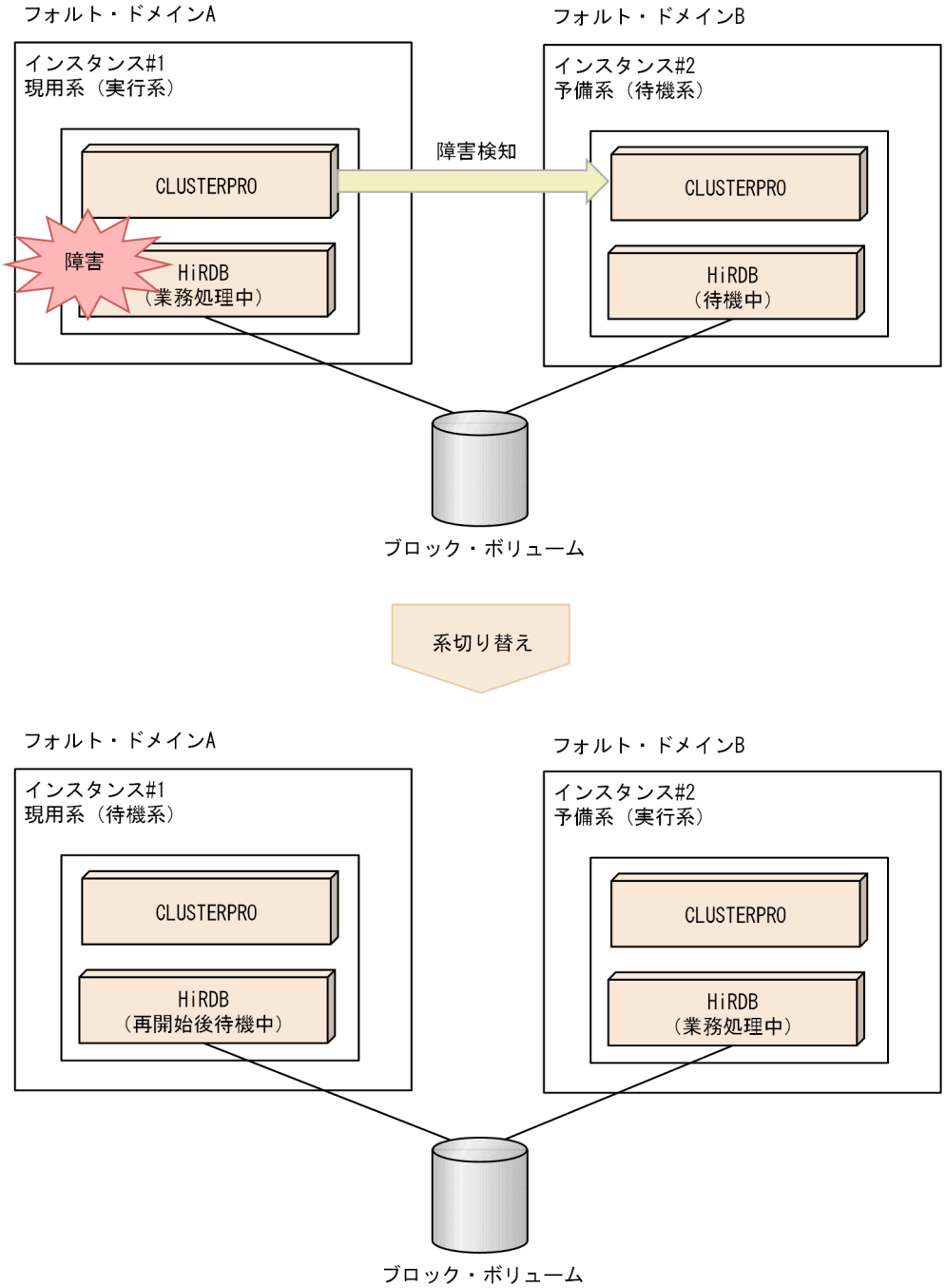


図 26–51　Oracle Cloud Infrastructure 環境上の共有ディスクを使用したクラスタ構成概要図



〔説明〕

業務処理中の実行系に障害が発生すると，待機系に障害の発生が通知されて系が切り替わり，待機系が実行系になって業務処理を続行します。

Amazon Web Services 環境上，Microsoft Azure 環境上，及び Oracle Cloud Infrastructure 環境上に CLUSTERPRO のミラーディスクを使用したクラスタ構成を適用する場合の概要を次の図に示します。

図 26–52　Amazon Web Services 環境上のミラーディスクを使用したクラスタ構成概要図

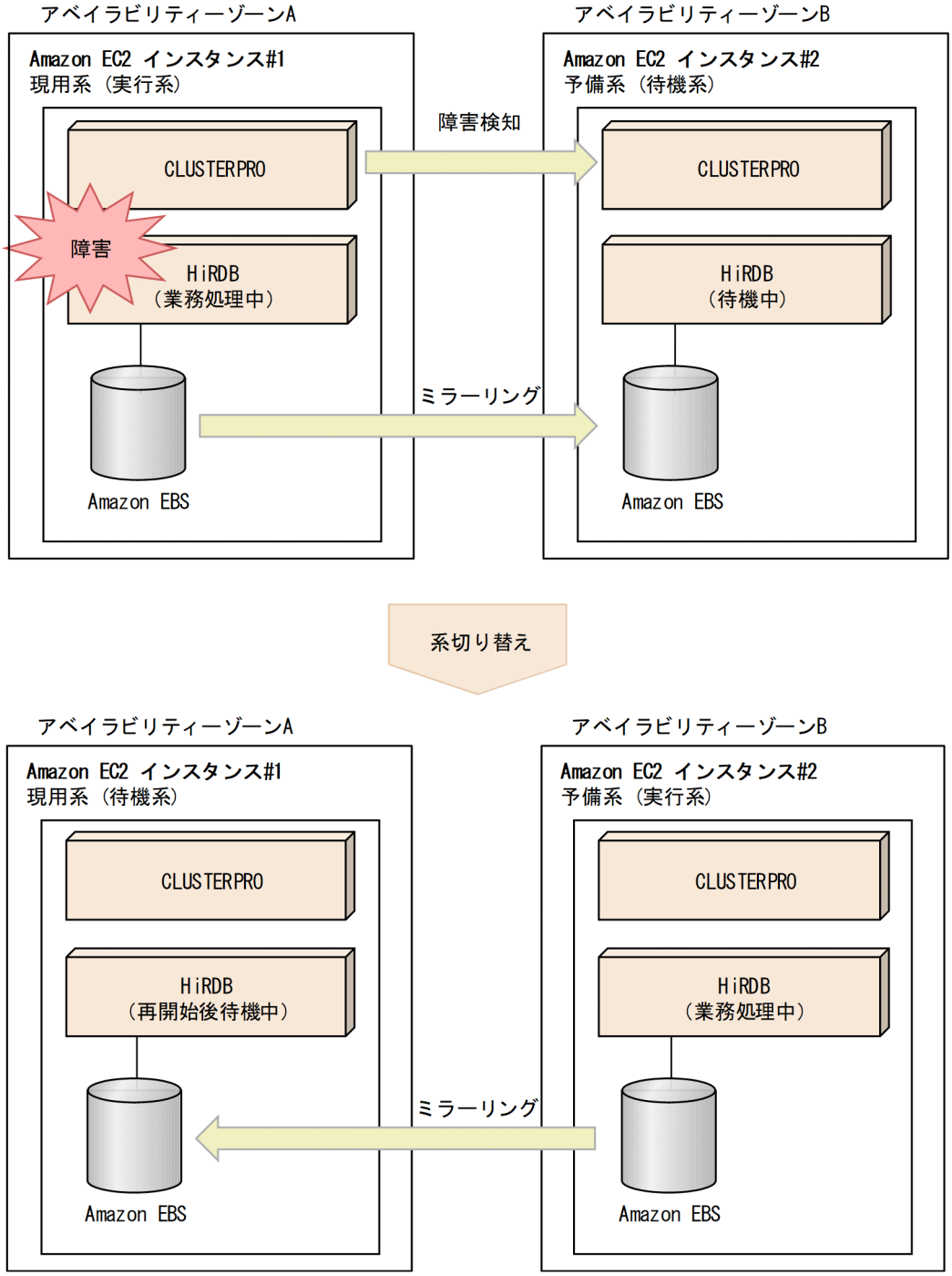


図 26–53　Microsoft Azure 環境上のミラーディスクを使用したクラスタ構成概要図

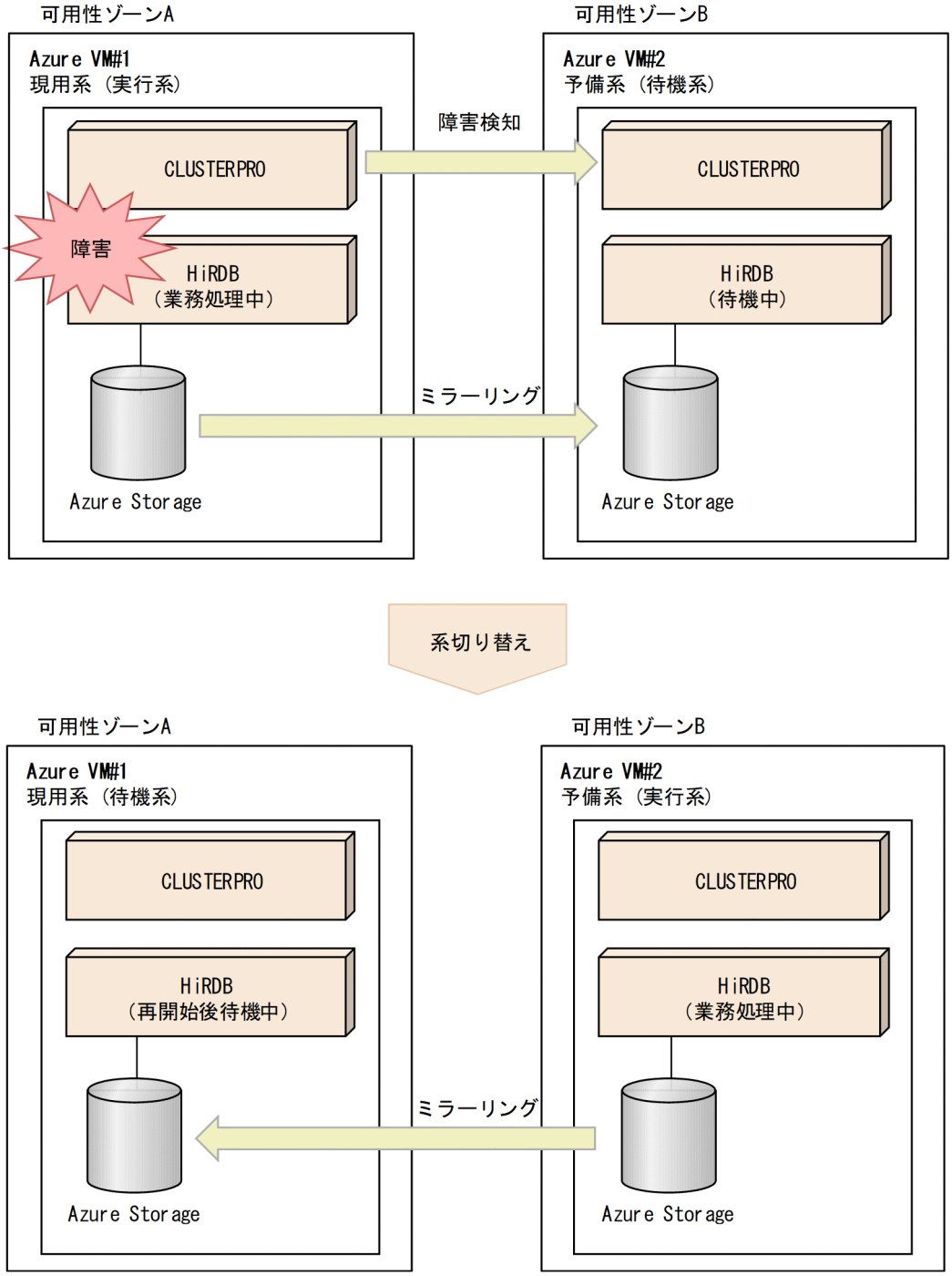
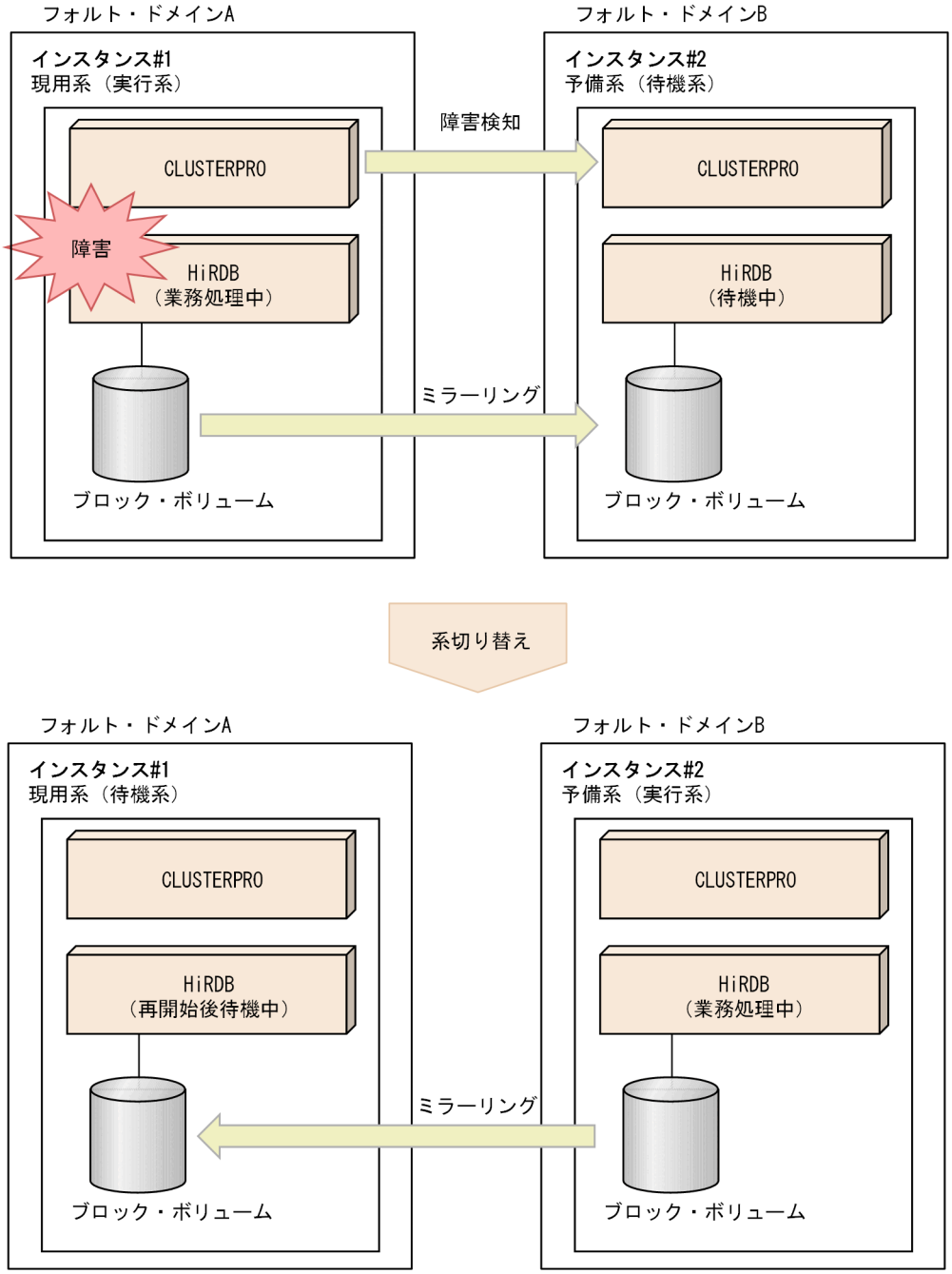


図 26–54　Oracle Cloud Infrastructure 環境上のミラーディスクを使用したクラスタ構成概要図



〔説明〕

業務処理中の実行系に障害が発生すると，待機系に障害の発生が通知されて系が切り替わり，待機系が実行系になって業務処理を続行します。

実行系のストレージ（Amazon EBS，Azure Storage，又はブロック・ボリューム）に対して更新すると，CLUSTERPRO は更新内容を待機系に送信し，待機系のストレージにミラーリングします。そのため，実行系の障害によって系が切り替わった場合，障害発生直前の状態から業務を続行できます。

(3)　パブリッククラウド環境上の IP アドレスの構成について

パブリッククラウド環境でクラスタ構成を構築する場合，IP アドレスは次の構成をサポートします。

表 26–36　パブリッククラウドの系切り替えでサポートする IP アドレスの構成

パブリッククラウド	IP アドレスを引き継ぐ構成※	IP アドレスを引き継がない構成	IP アドレスを引き継がない（クライアント接続用の IP アドレスだけを引き継ぐ）構成
Amazon Web Services	○	○	○
Microsoft Azure	−	○	○
Oracle Cloud Infrastructure	−	○	○

（凡例）

○：サポートします。

−：サポートしません。

注※

HiRDB サーバの IP アドレスを引き継ぐ系切り替え構成を前提とするマルチスタンバイ構成を含みます。また，Amazon Web Services 環境では AWS 仮想 IP リソースのみ IP アドレスを引き継ぐ構成をサポートします。

(4)　IP アドレスに関係するリソースの作成

Amazon Web Services や，Microsoft Azure 環境上で IP アドレス，又はクライアント接続用の IP アドレスを引き継ぐ場合，CLUSTERPRO のグループリソースが異なるため，それぞれの設定項目を説明します。

(a)　Amazon Web Services 上で IP アドレスを引き継ぐ場合，又はクライアント接続用の IP アドレスを引き継ぐ場合

CLUSTERPRO のグループリソースのうち，AWS 仮想 IP リソース，又は AWS セカンダリ IP アドレスリソースを作成してください。

なお，それぞれのリソースを作成する前に，Amazon Web Services 上で仮想 IP アドレス，又はセカンダリ IP アドレスを設定してください。詳細は Amazon Web Services，及び CLUSTERPRO のマニュアルを参照してください。

それぞれのリソースについて，HiRDB に関係する設定項目を次の表に示します。

表 26–37　AWS 仮想 IP リソースに指定する項目の値

プロパティ	項目		入力・選択値
情報	タイプ		「AWS 仮想 IP リソース」を選択します。
依存関係	−		「既定の依存関係に従う」にチェックを入れます。
詳細	共通	IP アドレス	使用する VIP アドレスを指定します。

プロパティ	項目		入力・選択値
		VPC ID	現用系 EC2 が所属する VPC ID を指定します。
		ENI ID	現用系 EC2 に対応する VIP のルーティング先の ENI ID を指定します。
	現用系サーバ	個別に設定する	チェックを入れます。
		VPC ID	現用系 EC2 が所属する VPC ID を指定します。
		ENI ID	現用系 EC2 に対応する VIP のルーティング先の ENI ID を指定します。
	予備系サーバ	個別に設定する	チェックを入れます。
		VPC ID	予備系 EC2 が所属する VPC ID を指定します。
		ENI ID	予備系 EC2 に対応する VIP のルーティング先の ENI ID を指定します。

表 26–38　AWS セカンダリ IP アドレスリソースに指定する項目の値

プロパティ	項目		入力・選択値
情報	タイプ		「AWS セカンダリ IP アドレスリソース」を選択します。
依存関係	–		「既定の依存関係に従う」にチェックを入れます。
詳細	共通	IP アドレス	使用するセカンダリ IP アドレスを指定します。
		ENI ID	現用系 EC2 に対応するセカンダリ IP アドレスのルーティング先の ENI ID を指定します。
	現用系サーバ	個別に設定する	チェックを入れます。
		ENI ID	現用系 EC2 に対応するセカンダリ IP アドレスのルーティング先の ENI ID を指定します。
	予備系サーバ	個別に設定する	チェックを入れます。
		ENI ID	予備系 EC2 に対応するセカンダリ IP アドレスのルーティング先の ENI ID を指定します。

(b)　Amazon Web Services 上でクライアント接続用の IP アドレスを引き継ぐ場合

CLUSTERPRO のグループリソースのうち，AWS Elastic IP リソース，又は AWS DNS リソースを作成してください。

なお，それぞれのリソースを作成する前に，Amazon Web Services 上で Elastic IP アドレス，又は Amazon Route 53 を設定してください。詳細は Amazon Web Services，及び CLUSTERPRO のマニュアルを参照してください。

それぞれのリソースでの HiRDB に関係する設定項目を次の表に示します。

表 26–39　AWS Elastic IP リソースに指定する項目の値

プロパティ	項目		入力・選択値
情報	タイプ		「AWS Elastic IP リソース」を選択します。
依存関係	－		「既定の依存関係に従う」にチェックを入れます。
詳細	共通	EIP ALLOCATION ID	付け替え対象の EIP の ID を指定します。
		ENI ID	現用系 EC2 に対応する EIP のルーティング先の ENI ID を指定します。
	現用系サーバ	個別に設定する	チェックを入れます。
		ENI ID	現用系 EC2 に対応する EIP のルーティング先の ENI ID を指定します。
	予備系サーバ	個別に設定する	チェックを入れます。
		ENI ID	予備系 EC2 に対応する EIP のルーティング先の ENI ID を指定します。

表 26–40　AWS DNS リソースに指定する項目の値

プロパティ	項目		入力・選択値
情報	タイプ		「AWS DNS リソース」を選択します。
依存関係	－		「既定の依存関係に従う」にチェックを入れます。
詳細	共通	IP アドレス	現用系の IP アドレスを指定します。
		非活性時にリソースレコードセットを削除する	チェックを入れます。
	現用系サーバ	個別に設定する	チェックを入れます。
		IP アドレス	現用系の IP アドレスを指定します。
	予備系サーバ	個別に設定する	チェックを入れます。
		IP アドレス	予備系の IP アドレスを指定します。

(c)　Microsoft Azure 上でクライアント接続用の IP アドレスを引き継ぐ場合

CLUSTERPRO のグループリソースのうち，Azure プローブポートリソース，又は Azure DNS リソースを作成してください。

なお，それぞれのリソースを作成する前に，Microsoft Azure 上で Azure ロードバランサー，又は Azure DNS を設定してください。詳細は Microsoft Azure，及び CLUSTERPRO のマニュアルを参照してください。

Azure ロードバランサー，及び，それぞれのリソースでの HiRDB に関係する設定項目を次の表に示します。

表 26–41　Azure ロードバランサーに指定する項目の値

項目		指定値
フロントエンド IP 構成の IP アドレス		HiRDB クライアントからロードバランサーに接続するための任意の IP アドレスを指定してください。
バックエンドプールの仮想マシン及び IP アドレス		HiRDB サーバを配置する仮想マシンと，その IP アドレスを指定してください。
負荷分散規則のポート	通常接続又は FES ホストダイレクト接続の場合	pd_name_port オペランドの指定値，又は pdunit オペランドの-p オプションの指定値を指定してください。
	高速接続の場合	次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
負荷分散規則のバックエンドポート		同上
負荷分散規則のフローティング IP		無効を選択してください。
正常性プローブ	プロトコル	TCP を選択してください。
	ポート	次に示した HiRDB で使用するポート番号，及び他の製品で指定しているポート番号を除いた未使用のポート番号を指定してください。 • OS 自動割り当てポート • HiRDB で指定するポート 詳細は「HiRDB システム導入・設計ガイド」の「HiRDB で指定するポート番号の一覧」を参照してください。

表 26–42　Azure プローブポートリソースに指定する項目の値

プロパティ	項目	入力・選択値
情報	タイプ	「Azure プローブポートリソース」を選択します。
依存関係	–	「既定の依存関係に従う」にチェックを入れます。
詳細	プローブポート	Azure のロードバランサーが，各サーバの死活監視（正常性プローブ）に使用するポート番号を指定します。

表 26–43　Azure DNS リソースに指定する項目の値

プロパティ	項目		入力・選択値
情報	タイプ		「Azure DNS リソース」を選択します。
依存関係	–		「既定の依存関係に従う」にチェックを入れます。
詳細	共通	IP アドレス	現用系の IP アドレスを指定します。
		非活性時にリソースレコードセットを削除する	チェックを入れます。

プロパティ	項目		入力・選択値
	現用系サーバ	個別に設定する	チェックを入れます。
		IP アドレス	現用系の IP アドレスを指定します。
	予備系サーバ	個別に設定する	チェックを入れます。
		IP アドレス	予備系の IP アドレスを指定します。

(d)　Oracle Cloud Infrastructure 上でクライアント接続用の IP アドレスを引き継ぐ場合

CLUSTERPRO のグループリソースのうち，Oracle Cloud DNS リソースを作成してください。

なお，リソースを作成する前に，Oracle Cloud Infrastructure 上で Oracle Cloud DNS を設定してください。詳細は Oracle Cloud Infrastructure ，及び CLUSTERPRO のマニュアルを参照してください。

Oracle Cloud DNS リソースでの HiRDB に関係する設定項目を次の表に示します。

表 26–44　Oracle Cloud DNS リソースに指定する項目の値

プロパティ	項目		入力・選択値
情報	タイプ		「Oracle Cloud DNS リソース」を選択します。
依存関係	−		「既定の依存関係に従う」にチェックを入れます。
詳細	共通	IP アドレス	現用系の IP アドレスを指定します。
		TTL	系切り替えに掛かる時間を考慮した上で，値を指定します。
		非活性時にリソースレコードセットを削除する	チェックを入れます。
	現用系サーバ	個別に設定する	チェックを入れます。
		IP アドレス	現用系の IP アドレスを指定します。
	予備系サーバ	個別に設定する	チェックを入れます。
		IP アドレス	予備系の IP アドレスを指定します。

(5)　HiRDB クライアントに関する設定項目

次のどれかのリソースを作成した場合，HiRDB サーバへの接続方式ごとに指定する HiRDB クライアントの環境変数を次の表に示します。

●AWS 仮想 IP リソースを作成する場合の指定値

表 26–45　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	AWS 仮想 IP リソースに指定した IP アドレス，又は対応するホスト名を指定してください。

環境変数		指定値
	HiRDB/パラレルサーバの場合	AWS 仮想 IP リソースに指定したシステムマネジャユニットの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–46　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	AWS 仮想 IP リソースに指定したシステムマネジャユニットの IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST をしている場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST	AWS 仮想 IP リソースに指定したフロントエンドサーバの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–47　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	AWS 仮想 IP リソースに指定した IP アドレス，又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	AWS 仮想 IP リソースに指定したシステムマネジャユニットの IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST		AWS 仮想 IP リソースに指定したフロントエンドサーバの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値

環境変数		指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE		「PC」を指定してください。

●AWS セカンダリ IP アドレスリソースを作成する場合の指定値

表 26–48　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	AWS セカンダリ IP アドレスリソースに指定した IP アドレス，又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	AWS セカンダリ IP アドレスリソースに指定したシステムマネジャユニットの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–49　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	AWS セカンダリ IP アドレスリソースに指定したシステムマネジャユニットの IP アドレス，又は対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST	AWS セカンダリ IP アドレスリソースに指定したフロントエンドサーバの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–50　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	AWS セカンダリ IP アドレスリソースに指定した IP アドレス，又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	AWS セカンダリ IP アドレスリソースに指定したシステムマネジャユニットの IP アドレス，又は対応するホスト名を指定してください。なお，

環境変数		指定値
		PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST		AWS セカンダリ IP アドレスリソースに指定したフロントエンドサーバの IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE		「PC」を指定してください。

●AWS Elastic IP リソースを作成する場合の指定値

表 26–51　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	AWS 上に割り当てた Elastic IP アドレス，又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	AWS 上に割り当てたシステムマネジャユニットの Elastic IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–52　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	AWS 上に割り当てたシステムマネジャユニットの Elastic IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。

環境変数	指定値
PDFESHOST	AWS 上に割り当てたフロントエンドサーバの Elastic IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–53　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	AWS 上に割り当てた Elastic IP アドレス，又は対応するホスト名を指定してください。
	HiRDB/パラレルサーバの場合	AWS 上に割り当てたシステムマネジャユニットの Elastic IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST		AWS 上に割り当てたフロントエンドサーバの Elastic IP アドレス，又は対応するホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE		「PC」を指定してください。

●AWS DNS リソースを作成する場合の指定値

表 26–54　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	Amazon Route 53 に登録されているシングルサーバの FQDN を指定してください。
	HiRDB/パラレルサーバの場合	Amazon Route 53 に登録されているシステムマネジャユニットの FQDN を指定してください。

環境変数		指定値
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–55　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	Amazon Route 53 に登録されているシステムマネジャユニットの FQDN，又は PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。
PDFESHOST	Amazon Route 53 に登録されているフロントエンドサーバの FQDN を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–56　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	Amazon Route 53 に登録されているシングルサーバの FQDN を指定してください。
	HiRDB/パラレルサーバの場合	Amazon Route 53 に登録されているシステムマネジャユニットの FQDN，又は PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。
PDFESHOST		Amazon Route 53 に登録されているフロントエンドサーバの FQDN を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。

環境変数	指定値
PDSRVTYPE	「PC」を指定してください。

●Azure プローブポートリソースを作成する場合の指定値

表 26–57　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	シングルサーバを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，又はホスト名を指定してください。
	HiRDB/パラレルサーバの場合	システムマネジャユニットを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，又はホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–58　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	システムマネジャユニットを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，または対応するホスト名を指定してください。なお，PDFESHOST を指定している場合は，PDFESHOST と同じ値を指定してください。
PDFESHOST	フロントエンドサーバユニットを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，又はホスト名を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–59　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	シングルサーバを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，又はホスト名を指定してください。
	HiRDB/パラレルサーバの場合	システムマネジャユニットを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，又はホスト名を，PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。

環境変数		指定値
PDFESHOST		フロントエンドサーバユニットを配置したサーバマシンに対応する Azure ロードバランサーのフロントエンド IP 構成の IP アドレス，又はホスト名を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE		「PC」を指定してください。

●Azure DNS リソースを作成する場合の指定値

表 26–60　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	Azure DNS に登録されている，シングルサーバの FQDN を指定してください。
	HiRDB/パラレルサーバの場合	Azure DNS に登録されている，システムマネジャユニットの FQDN を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–61　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	Azure DNS に登録されている，システムマネジャの FQDN，又は PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。
PDFESHOST	Azure DNS に登録されている，フロントエンドサーバの FQDN を指定してください。

環境変数	指定値
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–62　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	Azure DNS に登録されている，シングルサーバの FQDN を指定してください。
	HiRDB/パラレルサーバの場合	Azure DNS に登録されている，システムマネジャの FQDN，又は PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。
PDFESHOST		Azure DNS に登録されている，フロントエンドサーバの FQDN を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE		「PC」を指定してください。

●Oracle Cloud DNS リソースを作成する場合の指定値

表 26–63　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が通常接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	Oracle Cloud DNS に登録されている，シングルサーバの FQDN を指定してください。
	HiRDB/パラレルサーバの場合	Oracle Cloud DNS に登録されている，システムマネジャユニットの FQDN を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。

環境変数		指定値
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又はシステムマネジャユニットの pdunit オペランドの-p オプションの指定値を指定してください。

表 26–64　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が FES ホストダイレクト接続の場合）

環境変数	指定値
PDHOST	Oracle Cloud DNS に登録されている，システムマネジャの FQDN，又は PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。
PDFESHOST	Oracle Cloud DNS に登録されている，フロントエンドサーバの FQDN を指定してください。
PDNAMEPORT	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEGRP	フロントエンドサーバのサーバ名を指定してください。

表 26–65　HiRDB 接続に関連する環境変数の値（HiRDB の接続方式が高速接続の場合）

環境変数		指定値
PDHOST	HiRDB/シングルサーバの場合	Oracle Cloud DNS に登録されている，シングルサーバの FQDN を指定してください。
	HiRDB/パラレルサーバの場合	Oracle Cloud DNS に登録されている，システムマネジャの FQDN，又は PDFESHOST を設定している場合は PDFESHOST と同じ値を指定してください。
PDFESHOST		Oracle Cloud DNS に登録されている，フロントエンドサーバの FQDN を指定してください。
PDNAMEPORT	HiRDB/シングルサーバの場合	pd_name_port オペランドの指定値，又はシングルサーバの pdunit オペランドの-p オプションの指定値を指定してください。
	HiRDB/パラレルサーバの場合	pd_name_port オペランドの指定値，又は PDFESHOST（指定がない場合は PDHOST）で接続先としたユニットの pdunit オペランドの-p オプションの指定値を指定してください。
PDSERVICEPORT		次のどれかを指定してください。 • pd_service_port オペランドの指定値 • pd_scd_port オペランドの指定値 • pdunit オペランドの-s オプションの指定値
PDSERVICEGRP	HiRDB/シングルサーバの場合	シングルサーバのサーバ名を指定してください。
	HiRDB/パラレルサーバの場合	フロントエンドサーバのサーバ名を指定してください。
PDSRVTYPE		「PC」を指定してください。

26.12　HiRDB の運用方法の違いは？

系切り替え機能を使用する場合は使用しないときに比べて次に示す運用方法が異なります。ここではその違いについて説明します。

- HiRDB の開始方法（サーバモードの場合）
- HiRDB の開始方法（モニタモードの場合）
- HiRDB の終了方法（サーバモードの場合）
- HiRDB の終了方法（モニタモードの場合）
- 系の状態監視
- 統計ログファイルの運用
- 運用上の注意事項
- スタンバイレス型系切り替え機能使用時の注意事項

26.12.1　HiRDB の開始方法（サーバモードの場合）

(1)　HiRDB/シングルサーバの場合

系切り替え機能使用時の HiRDB/シングルサーバの開始方法を次に示します。

〈手順〉

1. MSFC のフェールオーバークラスタ管理で HiRDB のサービスを「オンライン」にします。
2. **pdstart** コマンドで実行系 HiRDB を開始します。なお，Hitachi HA Toolkit の server 定義文の actcommand オペランドを指定している場合は不要です。
3. **pdstart** コマンドで待機系 HiRDB を開始します。待機系 HiRDB は待機状態になります。

(2)　HiRDB/パラレルサーバの場合（スタンバイ型系切り替え機能の場合）

実行系及び待機系の HiRDB をそれぞれ pdstart コマンドで開始します。

実行系ユニットの開始方法

実行系の各サーバマシンにログインして，pdstart -q コマンドでユニット単位に開始します。又は，システムマネジャのユニットで pdstart コマンドを実行し，実行系の全ユニットを開始します。

待機系ユニットの開始方法

待機系の各サーバマシンにログインして，pdstart -q コマンドでユニット単位に開始します。

〈手順〉

1. MSFC のフェールオーバークラスタ管理で HiRDB のサービスを「オンライン」にします。

2. actcommand オペランドを指定していない場合，各サーバマシンにログインして，pdstart -q コマンドでユニット単位に開始します。又は，システムマネジャのユニットで pdstart コマンドを実行し，実行系の全ユニットを開始します。actcommand オペランドを指定している場合，系切り替え対象外のユニットで pdstart -q コマンドを実行します。

(3)　1：1 スタンバイレス型系切り替え機能の場合

1：1 スタンバイレス型系切り替え機能の場合の開始手順は次のようになります。

〈手順〉

1. MSFC のフェールオーバークラスタ管理で HiRDB のサービスを「オンライン」にします。
2. システムマネジャから pdstart，又は各ユニットで pdstart -q コマンドを実行します。actcommand オペランドを指定していない場合，各サーバマシンにログインして，pdstart -q コマンドでユニット単位に開始します。又は，システムマネジャのユニットで pdstart コマンドを実行し，実行系の全ユニットを開始します。actcommand オペランドを指定している場合，系切り替え対象外のユニットで pdstart -q コマンドを実行します。

正規 BES ユニット及び代替 BES ユニットの開始方法を次の表に示します。

表 26–66　1：1 スタンバイレス型系切り替え機能使用時の HiRDB の開始方法

目的	実行するコマンド	備考
正規 BES ユニットの開始	pdstart -q	代替中で正規 BES ユニットが停止している場合にこのコマンドを実行すると，正規 BES ユニットが待機状態※1になります。
代替 BES ユニットの開始	pdstart -q	代替 BES ユニット内の代替部も一緒に開始します。正常状態の場合は代替部が待機状態※2になります。
代替部の開始	pdstart -q -c	自動的に開始するので代替部を停止，切り戻したとき以外は，実行する必要がありません。

注

サーバ単位の開始はできません。

注※1

正規 BES ユニットが待機状態でないと正規 BES ユニットに系を切り戻せません（代替中から正常状態に戻れません）。正規 BES ユニットが待機状態の場合は，pdls -d ha コマンドの実行結果で系の状態が SBY と表示されます。

注※2

代替部が待機状態でないと系を切り替えられません。代替 BES ユニットが待機状態の場合は，pdls -d ha コマンドの実行結果で系の状態が SBY と表示されます。

(4)　影響分散スタンバイレス型系切り替え機能の場合

影響分散スタンバイレス型系切り替え機能の場合の開始手順は次のようになります。

〈手順〉

1. MSFC のフェールオーバークラスタ管理で HiRDB のサービスを「オンライン」にします。
2. システムマネジャから pdstart，又は各ユニットで pdstart -q コマンドを実行します。なお，Hitachi HA Toolkit の server 定義文の actcommand オペランドを指定している場合は不要です。

正規ユニット及びゲスト BES ユニットの開始方法を次の表に示します。

表 26–67　影響分散スタンバイレス型系切り替え機能使用時の HiRDB の開始方法

目的	実行するコマンド	備考
正規ユニットの開始	pdstart -q	正規ユニットが停止している場合に起動します。
受け入れユニットの開始	pdstart -q	受け入れユニット内のゲスト BES も一緒に開始します。
ホスト BES の開始又はゲスト BES の開始	pdstart -u -s	ゲスト BES が停止している場合に受け入れ可能状態になります。 自動的に開始するのでバックエンドサーバ又はその待機系を明示的に停止したとき以外は，実行する必要がありません。

(a)　システム全体の開始

システム全体開始の運用方法を次の表に示します。

表 26–68　システム全体開始の運用方法

入力場所	コマンド	動作
システムマネジャがあるユニット	pdstart	システム全体開始

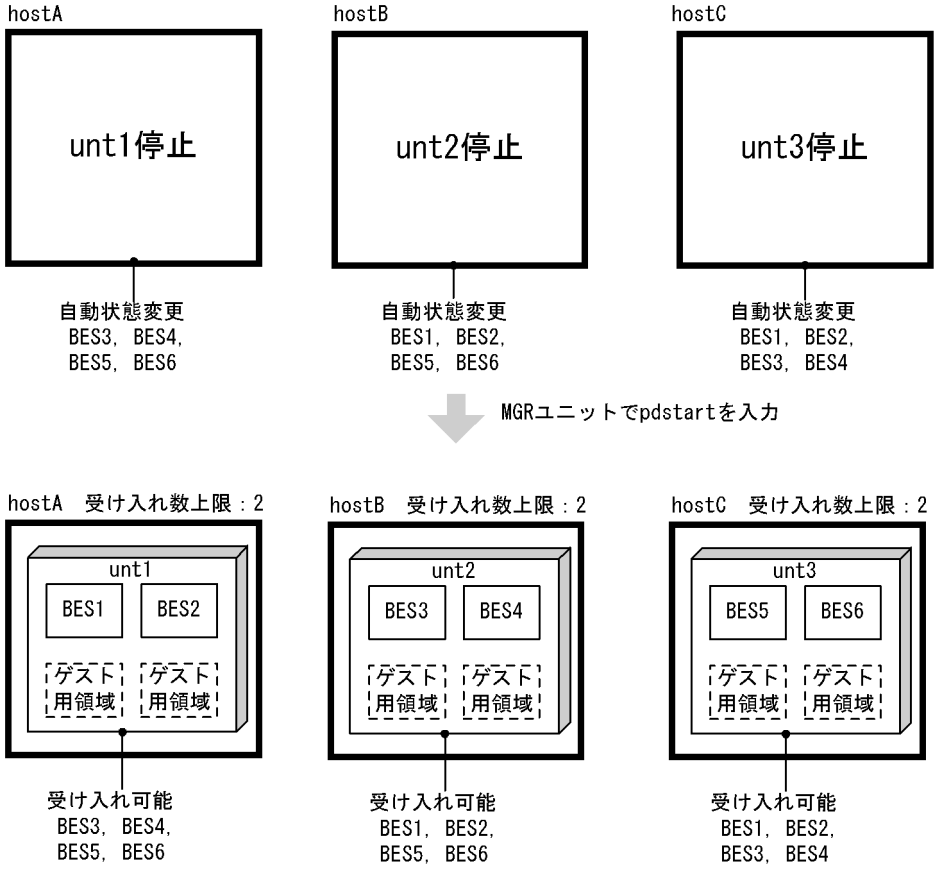
システム全体開始時の処理を次に示します。

- サーバの開始：各ユニットのユニットの全ホスト BES，及び全ゲスト BES に対してサーバ開始を実行します。

影響分散スタンバイレス型系切り替え機能でのシステム全体開始の例を次の図に示します。

システム全体を開始するためには，スタンバイ型系切り替えと同じくシステムマネジャがあるユニットから pdstart コマンドを入力します。各ユニットでは，自動的にゲスト BES が受け入れ可能状態になります。

図 26–55　影響分散スタンバイレス型系切り替え機能でのシステム全体開始の例



システム起動の動作を次の表に示します。システム起動が完了すると，メッセージ KFPS05210-I を出力します。

表 26–69　システム起動の動作

開始種別	影響分散スタンバイレス型系切り替え機能の適用ユニット	
	なし	あり
システム開始	全ユニット起動 （全サーバ起動）	全サーバ起動 （全ユニットは起動しません）
システムマネジャがあるユニットの単独開始 （システムマネジャがあるユニットの単独異常終了後）	フロントエンドサーバ，ディクショナリサーバ，バックエンドサーバの各々 1 サーバ起動	フロントエンドサーバ，ディクショナリサーバ，バックエンドサーバの各々 1 サーバ起動
HA グループ内ユニットの単独開始	−	−
その他ユニットの単独開始	−	−

（凡例）　−：該当しません。

(b)　ユニットの開始

ユニット開始の運用方法を次の表に示します。

表 26–70　ユニット開始の運用方法

入力場所	コマンド	オプション		動作
		-q	-u	
システムマネジャがあるユニット	pdstart	なし	あり	対象ユニット起動
		あり	なし	
対象ユニット	pdstart	あり	なし	

●ユニットの開始モード

ユニットの開始モードを次の表に示します。ユニットの正常開始，又は再開始は前回の該当ユニットの終了モードだけで決定され，実行起動するホスト BES，ゲスト BES の有無，実行起動するサーバの正常開始，又は再開始は関係しません。

表 26–71　ユニットの開始モード

前回のユニット終了モード	ホスト BES 又はゲスト BES		ユニットの開始モード
	開始サーバの起動	開始サーバ起動中に 再開始サーバ起動	
正常停止	なし	なし	正常開始
	あり	なし	
	あり	あり	
計画停止／強制終了／異常終了	なし	なし	再開始
	あり	なし	
	あり	あり	

影響分散スタンバイレス型系切り替え機能では，再開始，正常開始の判定と処理をユーザサーバごとに実施します。このため，ユニット内サーバが強制／異常終了後，又は HA グループ内の他ユニットで稼働中であっても，ユニットは正常開始可能です。

なお，影響分散スタンバイレス型系切り替え機能の場合もユニットの強制／異常終了後はユニット再開始になります。再開始処理はユーザサーバごとに行うためユニットそのものに回復するリソース（DB）がありませんが，従来と同じくユニット再開始には正常開始と比べて次の表に示す動作差異があります。

表 26–72　ユニット再開始の意味合い

項目	内容
構成変更チェック	システムリソース（共用メモリなど）が増加となる定義変更をチェックアウトし，再開始失敗となる危険性を排除します（受け入れ可能なゲスト BES 数は変更可能とします）。
システム起動完了チェック	システムマネジャのあるユニット再開始時にフロントエンドサーバ，ディクショナリサーバ，バックエンドサーバが 1 サーバずつあればシステム起動完了としてサービスを開始します。システムマネジャのあるユニット正常開始時は全サーバ起動完了を待ち合わせますが，システムマネジャのあるユニット再開始時は待ち合わせサーバを減らしてサービスの受け付けの再開を早めます。
ユニットラン ID の引き継ぎ	ユニットラン ID はユニットの停止前から引き継ぎます。ユニット開始時の KFPS01826-I メッセージ出力，及び pdcspool の定期自動実行基準として利用します。

●ユニットの開始の例

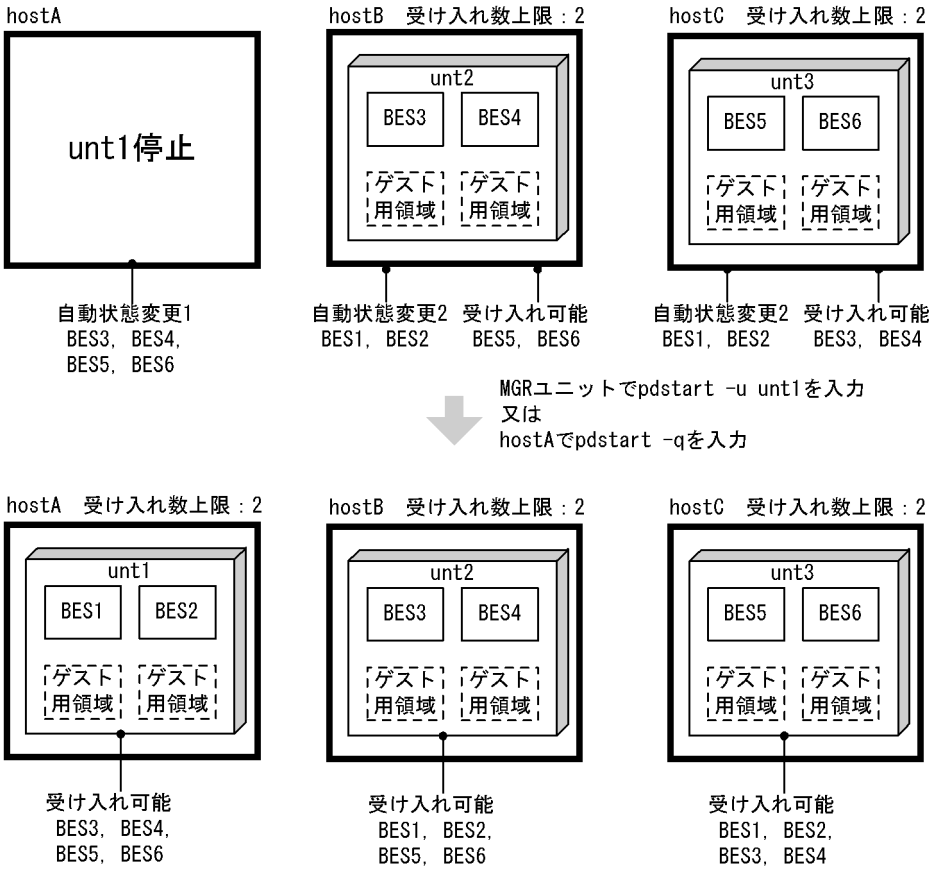
影響分散スタンバイレス型系切り替え機能でのユニット開始の例を次の図に示します。次のようにユニットを開始します。

- システムマネジャのあるユニットから pdstart -u コマンドを入力
- 該当ユニットを配置したホストにログインして，pdstart -q コマンドを入力

該当ユニットの起動に伴い，次のゲスト BES の状態をシステムが変更します。

- 該当ユニット内のゲスト BES の受け入れ可能状態の解除（図の自動状態変更 1）
- 該当ユニット内のホスト BES に対応するゲスト BES（図の自動状態変更 2）

図 26–56　影響分散スタンバイレス型系切り替え機能でのユニット開始の例



●実行系のバックエンドサーバがないユニットの開始の例

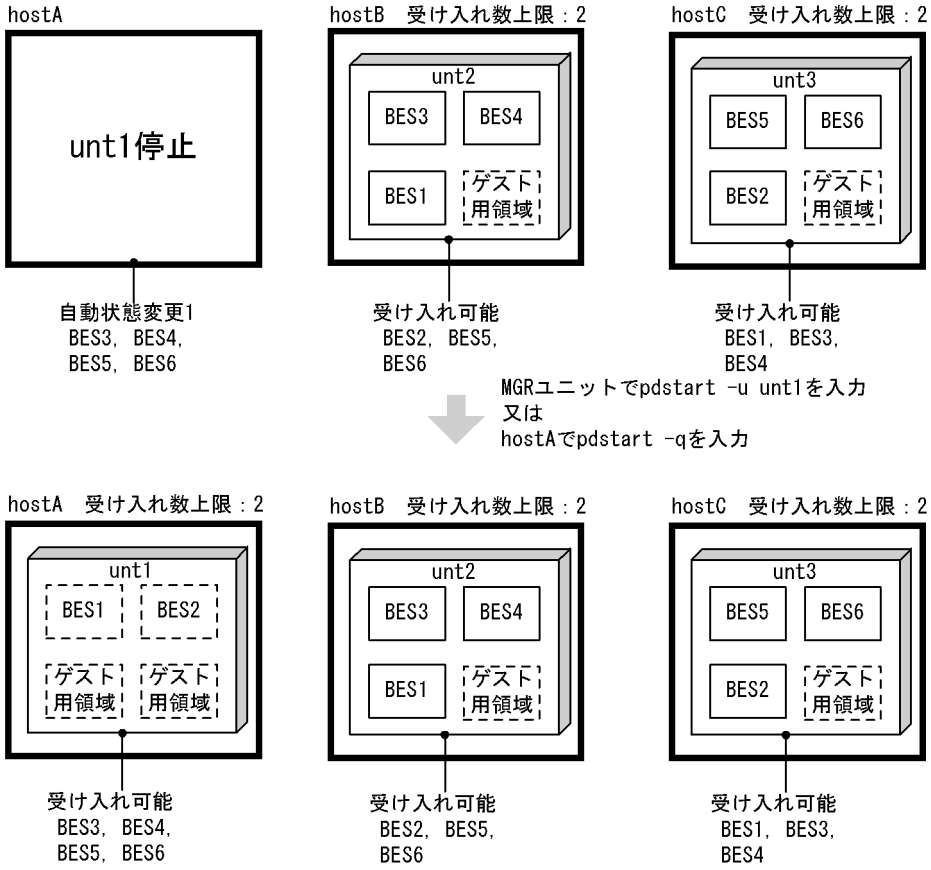
影響分散スタンバイレス型系切り替え機能での実行系のバックエンドサーバがないユニットの開始の例を次の図に示します。

ユニットを開始するには，次の方法があります。

- システムマネジャのあるユニットから pdstart -u コマンドを入力
- 該当ユニットを配置したホストにログインして，pdstart -q コマンドを入力

該当ユニットの起動に伴って該当ユニット内のゲスト BES の状態をシステムが変更します（図の自動状態変更 1)。

図 26–57　影響分散スタンバイレス型系切り替え機能での実行系のバックエンドサーバがないユニットの開始の例



ユニット起動の完了の動作は，メッセージ KFPS05110-I が出力されることで判断できます。

- 影響分散スタンバイレス型系切り替え機能の非適用ユニットの場合は，ユニット内全サーバ起動完了時にメッセージが出力されます。
- 影響分散スタンバイレス型系切り替え機能の適用ユニットの場合は，ユニット内全実行系サーバ起動完了時にメッセージが出力されます。

(c)　サーバの開始

サーバ開始の運用方法を次の表に示します。

表 26–73　サーバ開始の運用方法

入力場所	コマンド	オプション			動作
		-q	-u	-s	
システムマネジャのあるユニット	pdstart	なし	なし	あり	全稼働中 HA グループ内ユニットで対象サーバ開始 ※1
		あり	なし	あり	ホスト BES の開始（-u は省略可） ※2

注※1

全稼働中 HA グループ内ユニットのうち，1 ユニットでバックエンドサーバが実行中となり，ほかのユニットでは受け入れ可能状態となります。

注※2

対象となるバックエンドサーバが実行サーバとして起動完了した場合は，HA グループ内の稼働中のユニットは自動的に受け入れ可能状態となります。

使用するクラスタソフトウェアごとのサーバ開始時の処理結果を次の表に示します。

表 26–74　サーバ開始時の処理結果

クラスタソフトウェア	サーバ種別	該当サーバ稼働中の他ユニット	該当ホストでオンライン化	サーバ開始結果
Hitachi HA Toolkit Extension	–	なし	あり	稼働中 ※
			なし	受け入れ可能
		あり	あり	–
			なし	受け入れ可能

（凡例）　–：該当しません。

注※

Hitachi HA Toolkit Extension の場合，サーバを停止後もクラスタソフトウェアには稼働中（オンライン）を継続しているように見せています。このため，ゲスト BES の場合を含めて稼働中のサーバを停止後，同じユニットで開始すると，再び稼働状態となります。

●サーバの開始の例

影響分散スタンバイレス型系切り替え機能での実行系サーバの開始の例を次の図に示します。

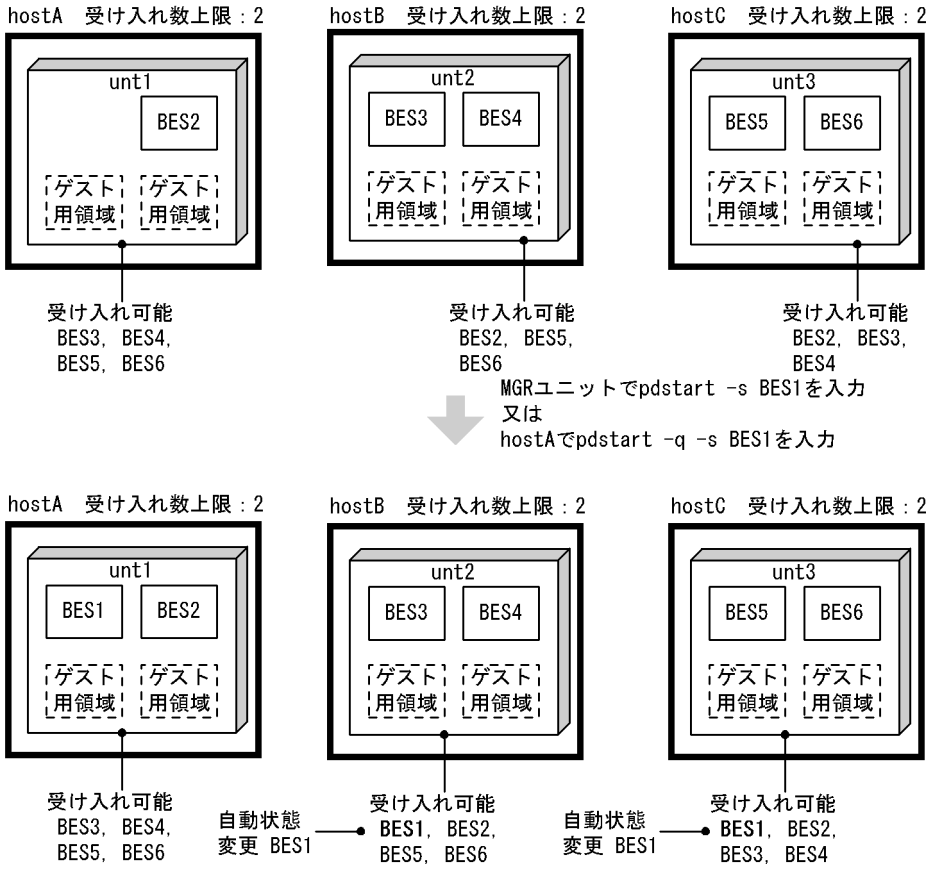
実行系サーバを開始するには，次の方法があります。

- システムマネジャのあるユニットから pdstart -s コマンドを入力します。

該当サーバの起動に伴い，該当サーバに対応するゲスト BES の状態をシステムが変更します（図の自動状態変更）。

なお，サーバが実行系かどうかは，pdls -d ha コマンドで確認できます。

図 26–58　影響分散スタンバイレス型系切り替え機能での実行系サーバの開始の例



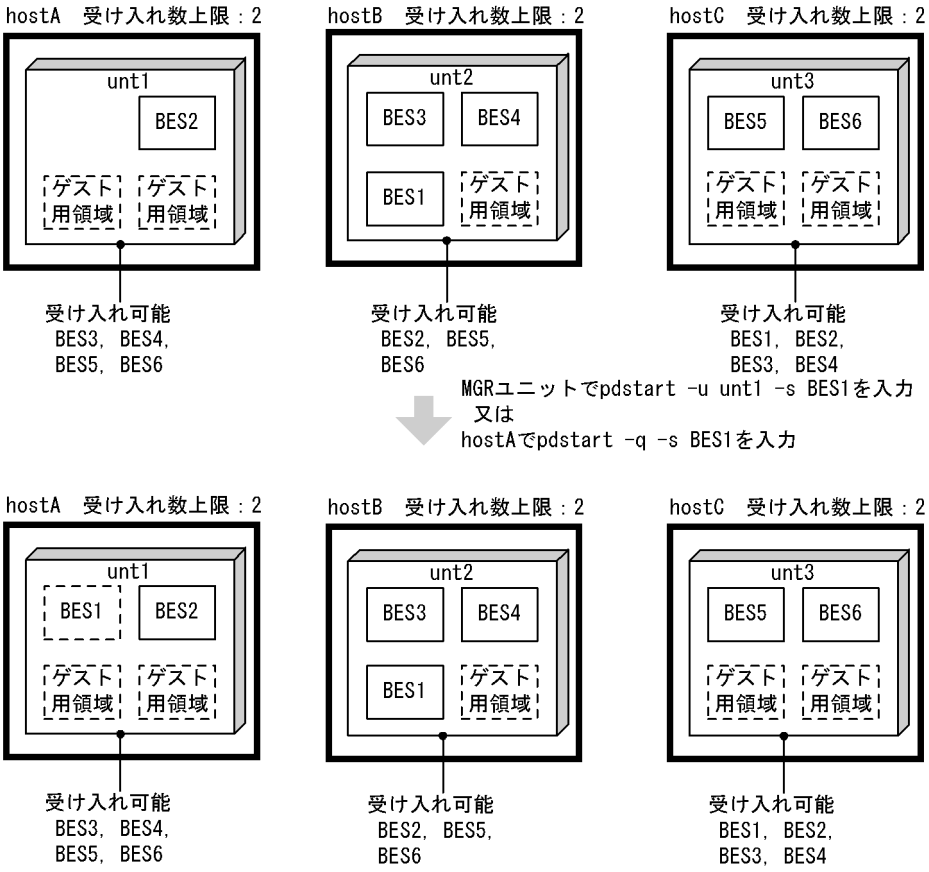
影響分散スタンバイレス型系切り替え機能での待機系サーバの開始の例を次の図に示します。

待機系サーバを開始するには，次の方法があります。

- システムマネジャのあるユニットから pdstart -u -s コマンドを入力

指定されたホスト BES は，受け入れ可能状態になります。なお，サーバが実行系かどうかは，pdls -d ha コマンドで確認できます。

図 26–59　影響分散スタンバイレス型系切り替え機能での待機系サーバの開始の例



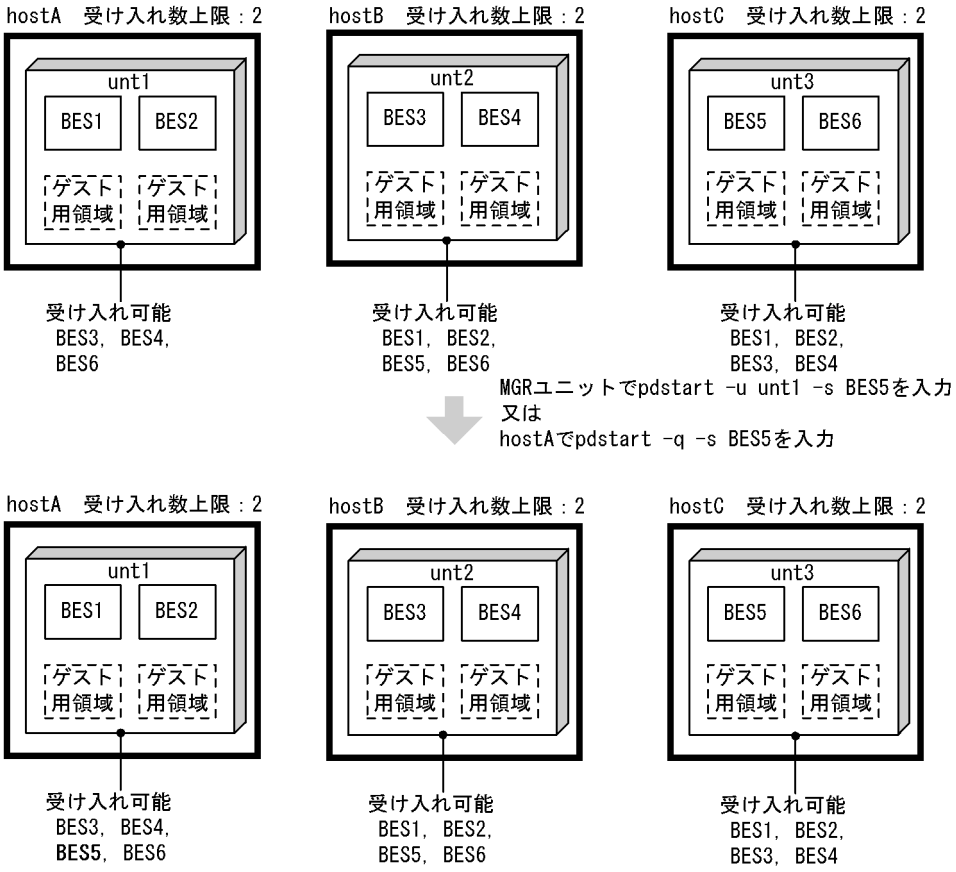
影響分散スタンバイレス型系切り替え機能でのゲストサーバの状態変更の例を次の図に示します。

ユニット内のゲストサーバを受け入れ可能状態にするには，次の方法があります。

- システムマネジャがあるユニットから pdstart -u -s コマンドを入力

指定されたゲスト BES は受け入れ可能状態になります。

図 26–60　影響分散スタンバイレス型系切り替え機能でのゲストサーバの状態変更の例



(5)　注意事項

共通の注意事項

- pdstart -q コマンドを実行する場合は，最初のユニットを開始した時点から 20 分以内に全ユニットを開始するようにしてください。20 分以内に全ユニットを開始できない場合は HiRDB の開始処理を中止します。なお，この 20 分という制限時間は pd_reduced_check_time オペランドの値によって変わります。20 分はこのオペランドの省略値です。
- 待機系 HiRDB で pdstart コマンドを実行する場合はオプションの指定に制限があります。-i オプション，-r オプション，及び dbdestroy オプションを指定できません。
- pdstart -r 又は pdstart -R コマンドを実行する場合は，実行系及び待機系 HiRDB を終了した後に pdstart -r 又は pdstart -R コマンドを実行してください。pdstart -r 又は pdstart -R コマンドで HiRDB を開始した場合，HiRDB は系切り替えの対象になりません。データベースの回復処理などが終了した後に，一度 HiRDB を終了してから実行系及び待機系 HiRDB を開始してください。

高速系切り替え機能使用時の注意事項

ここで説明する注意事項は Hitachi HA Toolkit Extension を使用しているときだけ該当します。Hitachi HA Toolkit Extension を使用していない場合は該当しません。

高速系切り替え機能を適用した待機系ユニットは，実行系ユニットの開始処理が完了した後に開始してください。実行系ユニットが開始していない状態で待機系ユニットを開始すると，待機系ユニットは実

行系ユニットの開始完了を待ち合わせます。待ち合わせ時間内に実行系ユニットが開始しない場合，待機系ユニットはアボートコード Phi1012 を出力して異常終了します。

1：1 スタンバイレス型系切り替え機能使用時の注意事項

ここで説明する注意事項は Hitachi HA Toolkit Extension を使用しているときだけ該当します。Hitachi HA Toolkit Extension を使用していない場合は該当しません。

正規 BES ユニットと代替 BES ユニットは片方を開始してから 20 分以内に他方を開始してください。実行系ユニットが開始していない状態で待機系ユニットを開始すると，待機系ユニットは実行系ユニットの開始完了を待ち合わせます。待ち合わせ時間内に実行系ユニットが開始しない場合，待機系ユニットはアボートコード Phi1012 を出力して異常終了します。

なお，正常時は正規 BES ユニットが実行系になり，代替部が待機系になります。代替中は代替部が実行系になり，正規 BES ユニットが待機系になります。

Hitachi HA Toolkit Extension 使用時の注意事項

Hitachi HA Toolkit Extension のサービスプロセスを起動しないで HiRDB を開始すると，両系とも待機系として開始してしまいます。この場合は次の表に示す手順で対処してください。

表 26–75　Hitachi HA Toolkit Extension のサービスプロセスを起動しないで HiRDB を開始したときの対処方法

条件	対処方法
ユーザサーバホットスタンバイを適用しているユニットの場合	両系とも待機系として開始したことを示す KFPS01872-I メッセージが出力されます。このメッセージは両方の系に出力されます。次に示す手順で対処してください。 **〈手順〉** 1. Hitachi HA Toolkit Extension の待機系停止コマンドで，両方の系を終了します。 2. 実行系でクラスタソフトウェアをオンライン化します。 3. 実行系ユニットを開始します。 4. 実行系ユニットの開始処理が完了したことを確認※してから，待機系ユニットを開始します。
高速系切り替え機能を適用しているユニットの場合	KFPS01854-E メッセージを出力して現用系ユニットが異常終了します（アボートコードは Psadhfe）。予備系ユニットは，現用系ユニットが実行系ユニットとして開始するのを待ち合わせています。次に示す手順で対処してください。 **●現用系を実行系として開始する場合** 1. 現用系でクラスタソフトウェアをオンライン化します。 2. 現用系ユニットを実行系ユニットとして開始します。 3. 待ち合わせ時間を経過しているため，予備系（待機系）ユニットが異常終了している場合，実行系ユニットの開始処理が完了したことを確認※してから，待機系ユニットを開始します。 **●予備系を実行系として開始する場合** 1. pdstop -z（HiRDB/シングルサーバの場合は pdstop -f）コマンドで予備系ユニットを強制終了します。 2. 予備系で Hitachi HA Toolkit Extension のサービスプロセスを起動します。 3. 予備系ユニットを実行系ユニットとして開始します。 4. 実行系ユニットの開始処理が完了したことを確認※してから，待機系ユニットを開始します。

条件	対処方法
1：1 スタンバイレス型系切り替え機能を適用しているユニットの場合	KFPS01854-E メッセージを出力して正規 BES ユニットが異常終了します（アボートコードは Psadhfe）。代替部は正規 BES ユニットが開始するのを待ち合わせています。次に示す手順で対処してください。 **〈手順〉** 1. 正規 BES ユニットで Hitachi HA Toolkit Extension のサービスプロセスを起動します。 2. 正規 BES ユニットを開始します。 3. 待ち合わせ時間を経過しているため，代替部の待機状態が解除されている場合，正規 BES ユニットの開始処理が完了したことを確認※してから，代替部を待機状態にしてください。
影響分散スタンバイレス型系切り替え機能を適用している場合	実行系，待機系の両方のサーバが待機状態となり，ユーザが操作可能な状態になります。両系とも異常終了しません。 この場合，サーバを実行系とするホストでクラスタソフトウェアをオンライン化し，起動処理を完了できます（Hitachi HA Toolkit Extension のサービスプロセスを起動します）。

注※

ユニットの開始処理が完了したことを確認するには次に示す方法があります。

- pdls コマンドの実行結果の STATUS に ACTIVE と表示されている
- KFPS05210-I 又は KFPS05110-I メッセージが出力されている

26.12.2　HiRDB の開始方法（モニタモードの場合）

(1)　クラスタソフトウェアに MSFC を使用する場合

HiRDB のグループに登録したすべてのリソース（IP アドレス，ネットワーク名，汎用サービス，及び共有ディスク）をクラスタアドミニストレータでオンライン状態にすると，HiRDB が開始します。HiRDB が開始しない場合は障害の原因を取り除いた後，再度 HiRDB のグループに登録したすべてのリソースをクラスタアドミニストレータでオンライン状態にしてください。

(2)　クラスタソフトウェアに CLUSTERPRO を使用する場合

CLUSTERPRO のコマンド操作，又は Cluster WebUI による GUI 操作で，HiRDB ユニットを配置しているフェイルオーバグループを開始してください。HiRDB の開始処理に失敗した場合（スクリプトリソースが失敗した場合）は，障害要因を取り除いた後，HiRDB システム又は HiRDB ユニットが開始中の場合は HiRDB を強制終了（pdstop -f，又は pdstop -z コマンドで終了）してください。その後，CLUSTERPRO のコマンド操作，又は Cluster WebUI による GUI 操作で，スクリプトリソースを停止して再度リソースを開始してください。

(3)　注意事項

- クラスタソフトウェアの操作ではなく手動で HiRDB を開始する場合は，事前に共有リソースをオンライン状態にしてください。

- HiRDB/パラレルサーバの場合，システムマネジャのあるユニットで pdstart コマンドを実行してシステム全体を開始するのではなく，すべてのユニットをユニットごとに開始する必要があります。系切り替えの対象にするユニット（pd_ha_unit = nouse を指定しないユニット）は，ユニットごとにクラスタソフトウェアのコマンドでユニットを開始してください。系切り替えの対象にしないユニット（pd_ha_unit = nouse を指定したユニット）は，クラスタソフトウェアのコマンドで開始できません。この場合，該当するユニットのサーバマシンに直接ログインして pdstart -q コマンドを実行してください。

> **参考**
>
> pdstart -r 又は pdstart -R コマンドを実行する場合は，実行系及び待機系 HiRDB を終了した後に pdstart -r 又は pdstart -R コマンドを実行してください。pdstart -r 又は pdstart -R コマンドで HiRDB を開始した場合，HiRDB は系切り替えの対象になりません。データベースの回復処理などが終了した後に，一度 HiRDB を終了してから実行系及び待機系 HiRDB を開始してください。

26.12.3　HiRDB の終了方法（サーバモードの場合）

(1)　スタンバイ型系切り替え機能の場合

実行系及び待機系の両方を終了する場合（HiRDB/シングルサーバの場合）

1. Hitachi HA Toolkit Extension の **hatesbystp** コマンドで待機系 HiRDB を終了します。
2. **pdstop** コマンドで実行系 HiRDB を終了します。

pdstop コマンドを実行しても待機系 HiRDB は終了しません。計画停止又は強制終了のときも同様です。

実行系及び待機系の両方を終了する場合（HiRDB/パラレルサーバの場合）

1. Hitachi HA Toolkit Extension の **hatesbystp** コマンドで待機系の全ユニットを終了します。ユニットごとに hatesbystp コマンドを実行してください。
2. **pdstop** コマンドで実行系 HiRDB を終了します。

pdstop コマンドを実行しても待機系 HiRDB は終了しません。計画停止又は強制終了のときも同様です。

ユニット単位に終了する場合は，該当する待機系ユニットを Hitachi HA Toolkit Extension の hatesbystp コマンドで終了してから，実行系ユニットを pdstop -u コマンドで終了してください。pdstop -z コマンドを実行するときも同様です。

全ユニットを強制終了する場合は pdstop -f コマンドを実行してください。ユニット単位で強制終了する場合は pdstop -z コマンドを実行してください。

待機系 HiRDB だけを終了する場合

Hitachi HA Toolkit Extension の hatesbystp コマンドで待機系 HiRDB（又はユニット）を終了します。

注意事項

- 実行系，待機系の順に HiRDB を終了しないでください。実行系 HiRDB を終了してから待機系 HiRDB を終了するまでの間に，実行系のサーバマシンに障害が発生した場合，待機系 HiRDB に系の切り替え指示が通知されて HiRDB が正しく動作しなくなります。
- 待機系 HiRDB は Hitachi HA Toolkit Extension の hatesbystp コマンドで終了させてください。待機状態の HiRDB を誤って正常開始しようとした場合，HiRDB を開始できなくなることがあります。

(2)　1：1 スタンバイレス型系切り替え機能の場合

1：1 スタンバイレス型系切り替え機能使用時の HiRDB の終了方法を次の表に示します。

表 26–76　1：1 スタンバイレス型系切り替え機能使用時の HiRDB の終了方法

目的		実行するコマンド	備考（ほかのユニット又は代替部への影響）
HiRDB の終了		pdstop	代替 BES を意識した操作は必要ありません。代替中でも操作は変わりません。
正規 BES ユニットの終了		pdstop -u	代替部の待機状態※1が解除されます。
代替 BES ユニットの終了	正常時	pdstop -u	代替 BES ユニット内の代替部の待機状態※1が解除されます。
	代替中		代替 BES ユニット内の代替部も停止します。 また，正規 BES ユニットが待機状態※2の場合は，待機状態を解除します。
代替中の代替部の終了		pdstop -u	ユニット識別子には正規 BES ユニットのユニット識別子を指定します。
代替部の待機状態の解除		hatesbystp※3	なし。
正規 BES ユニットの待機状態の解除		hatesbystp※4	

注

サーバ単位の終了はできません。

注※1

代替部が待機状態でないと代替 BES ユニットに系を切り替えられません。代替部が待機状態の場合は，pdls -d ha コマンドの実行結果で系の状態が SBY と表示されます。

注※2

正規 BES ユニットが待機状態でないと正規 BES ユニットに系を切り戻せません（代替中から正常状態に戻れません）。正規 BES ユニットが待機状態の場合は，pdls -d ha コマンドの実行結果で系の状態が SBY と表示されます。

注※3

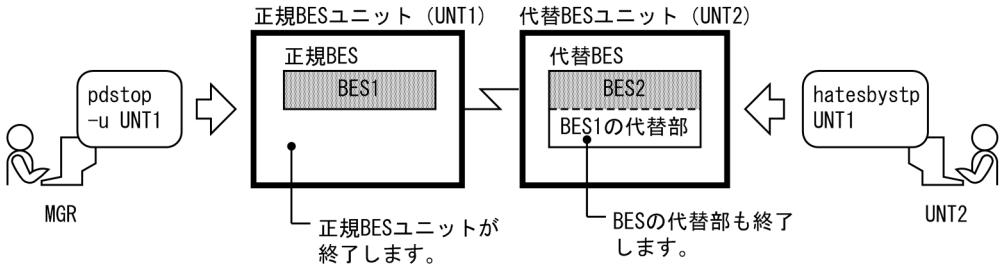
Hitachi HA Toolkit Extension の **hatesbystp** コマンドで，代替部の待機状態を解除します。なお，hatesbystp コマンドには，待機状態を解除する代替部に対応する正規 BES ユニットのエイリアス名を指定してください。

注※4

Hitachi HA Toolkit Extension の **hatesbystp** コマンドで，正規 BES ユニットの待機状態を解除します。なお，hatesbystp コマンドには，待機状態を解除する正規 BES ユニットのユニット識別子を指定してください。

代替 BES ユニット又は正規 BES ユニットの終了方法を例で説明します。

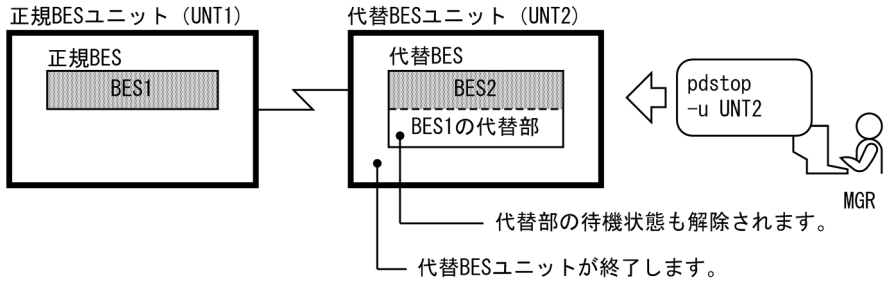
（例 1）正規 BES ユニットを終了する場合（正常時，片方向代替構成）



停止中の正規 BES ユニット（UNT1）を開始する場合は，次に示す手順で開始してください。

1. pdstart -q コマンドで正規 BES ユニット（UNT1）を開始します。
2. pdstart -q -c コマンドで BES1 の代替部を待機状態にします。

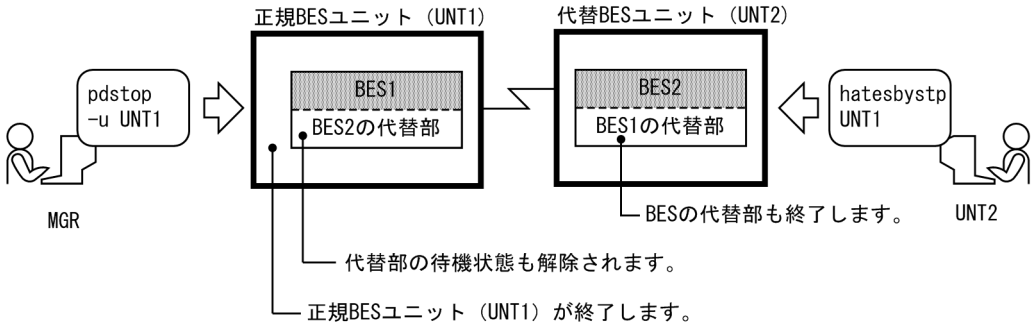
（例 2）代替 BES ユニットを終了する場合（正常時，片方向代替構成）



停止中の代替 BES ユニット（UNT2）を開始する場合は，次に示す手順で開始してください。

1. pdstart -q コマンドで代替 BES ユニット（UNT2）を開始します。このとき，BES1 の代替部も待機状態になります。

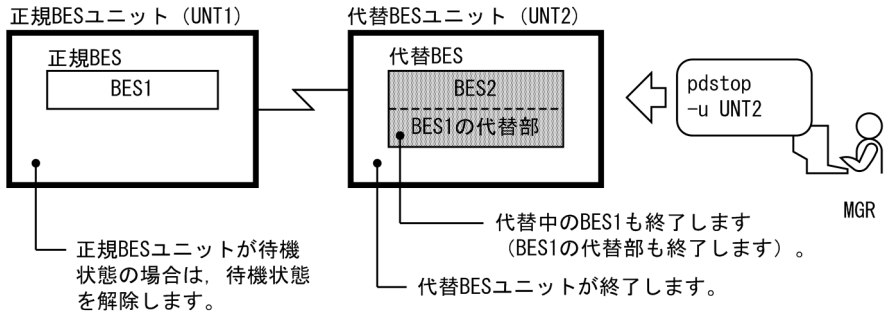
（例 3）ユニットを終了する場合（正常時，相互代替構成）



停止中のユニット（UNT1）を開始する場合は，次に示す手順で開始してください。

1. pdstart -q コマンドでユニット（UNT1）を開始します。このとき，BES2 の代替部も待機状態になります。
2. pdstart -q -c コマンドで BES1 の代替部を待機状態にします。

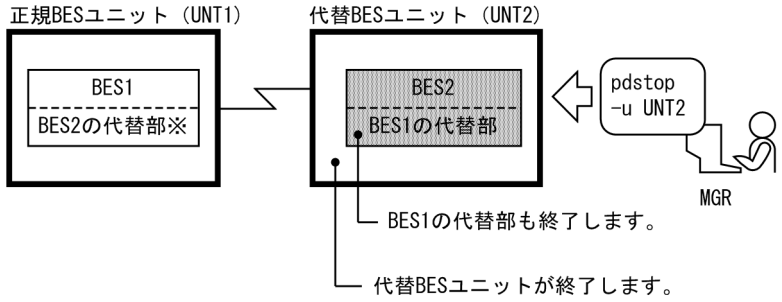
（例 4）ユニットを終了する場合（代替中，片方向代替構成）



正常状態に戻す場合は，次に示す手順で開始してください。

1. BES1 用のサービスを切り戻してください。
2. pdstart -q コマンドで代替 BES ユニット（UNT2）を開始します。
3. pdstart -q コマンドで正規 BES ユニット（UNT1）を開始します。

（例 5）ユニットを終了する場合（代替中，相互代替構成）



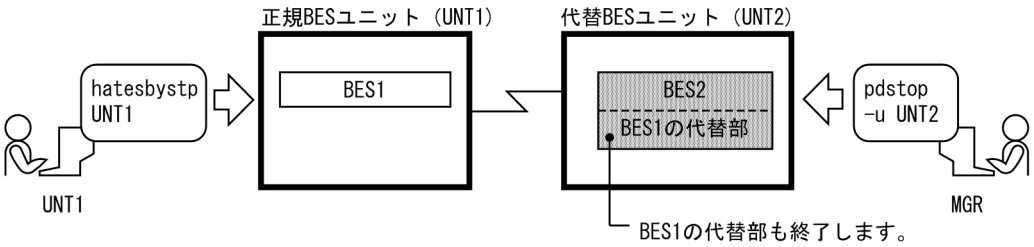
注※　BES2 の代替部は起動していない状態です。

正常状態に戻す場合は，次に示す手順で開始してください。

1. BES1 用のサービスを切り戻してください。
2. pdstart -q コマンドでユニット（UNT2）を開始します。

3. pdstart -q コマンドでユニット（UNT1）を開始します。

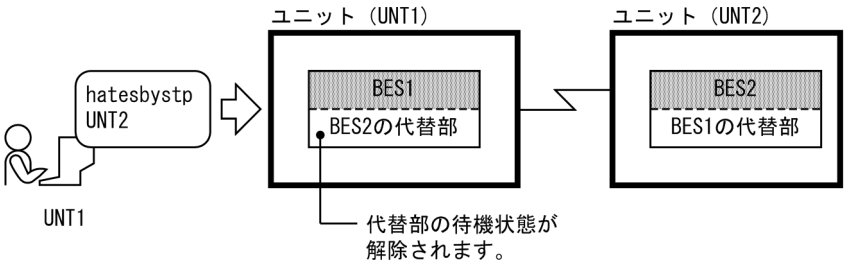
（例 6）代替部を終了する場合（代替中，片方向代替構成）



正常状態に戻す場合は，次に示す手順で開始してください。

1. BES1 用のサービスを切り戻してください。
2. pdstart -q コマンドでユニット（UNT1）を開始します。

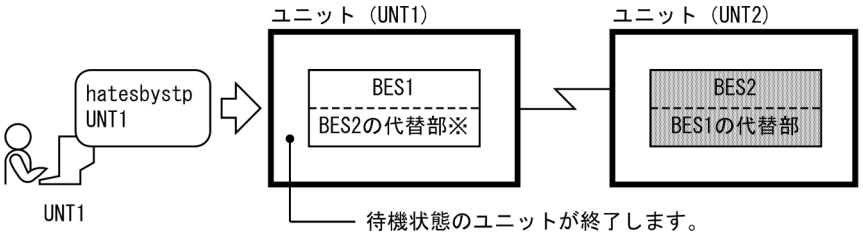
（例 7）代替部の待機状態を解除する場合（正常時，相互代替構成）



停止中の BES2 の代替部を待機状態にする場合は，次に示す手順で待機状態にしてください。

1. pdstart -q -c コマンドで BES2 の代替部を待機状態にします。

（例 8）待機状態のユニットを停止する場合（代替中，相互代替構成）



注※　BES2 の代替部は起動していない状態です。

停止中のユニット（UNT1）を待機状態にする場合は，次に示す手順で待機状態にしてください。

1. pdstart -q コマンドでユニット（UNT1）を待機状態にします。

(3)　影響分散スタンバイレス型系切り替え機能の場合（実行系，及び待機系の両方を終了）

実行系，及び待機系の両方を終了する場合の停止方法について説明します。

(a)　システム全体の停止

影響分散スタンバイレス型系切り替え機能でのシステム全体の停止運用を次の表に示します。

表 26–77　影響分散スタンバイレス型系切り替え機能でのシステム全体の停止運用

入力場所	コマンド	オプション	条件	動作
		-f	強制／異常停止サーバ	
システムマネジャのあるユニット	pdstop	なし	あり※1	エラー（新規メッセージ（KFPS05063-E 相当）出力）
		なし	なし※2	システム停止
		あり	あり	システム強制停止（一部ユニットは既に停止中）
		あり	なし	システム強制停止

注※1

強制又は異常停止サーバがある場合でも，該当強制又は異常停止サーバが次の状態のときは，強制又は異常停止サーバなしとします。

- 別のユニットで再開始して稼働中
- 別のユニットで再開始後正常停止済み

注※2

強制又は異常停止サーバなしとは，次の状態のことです。

- システム内に強制又は異常停止したサーバがない
- システム内に強制又は異常停止したサーバがあり，該当サーバが別のユニットで再開始して稼働中
- システム内に強制又は異常停止したサーバがあり，該当サーバが別のユニットで再開始後正常停止済み

システム停止時の処理を次の表に示します。

表 26–78　システム停止時の処理

対象	処理内容
ユニット	システムが停止する場合は，システムマネジャから全ユニットに対してユニット停止（pdstop -u UID （-f） 相当）を実行します。
サーバ	システム停止に伴って各ユニットが停止する場合は，該当ユニットの全ホスト BES，及び全ゲスト BES に対してサーバ停止（pdstop -q -s サーバ名 （-f） 相当）を実行します。

システム停止時は，正常終了の場合も強制停止の場合も，自動的に受け入れユニットのゲスト BES の受け入れ可能状態を解除します。あるゲスト BES が稼働中であっても，停止操作は変わりません。ゲスト BES の稼働状態を意識した操作は不要です。

影響分散スタンバイレス型系切り替え機能でのシステム停止時の各種バックエンドサーバに対する処理を次の表に示します。

表 26–79　影響分散スタンバイレス型系切り替え機能でのシステム停止時の各種バックエンドサーバに対する処理

バックエンドサーバの種別		処理
ホスト BES	実行中	停止
	受け入れ可能状態	受け入れ可能状態の自動解除
ゲスト BES	実行中	自動停止
	受け入れ可能状態	受け入れ可能状態の自動解除

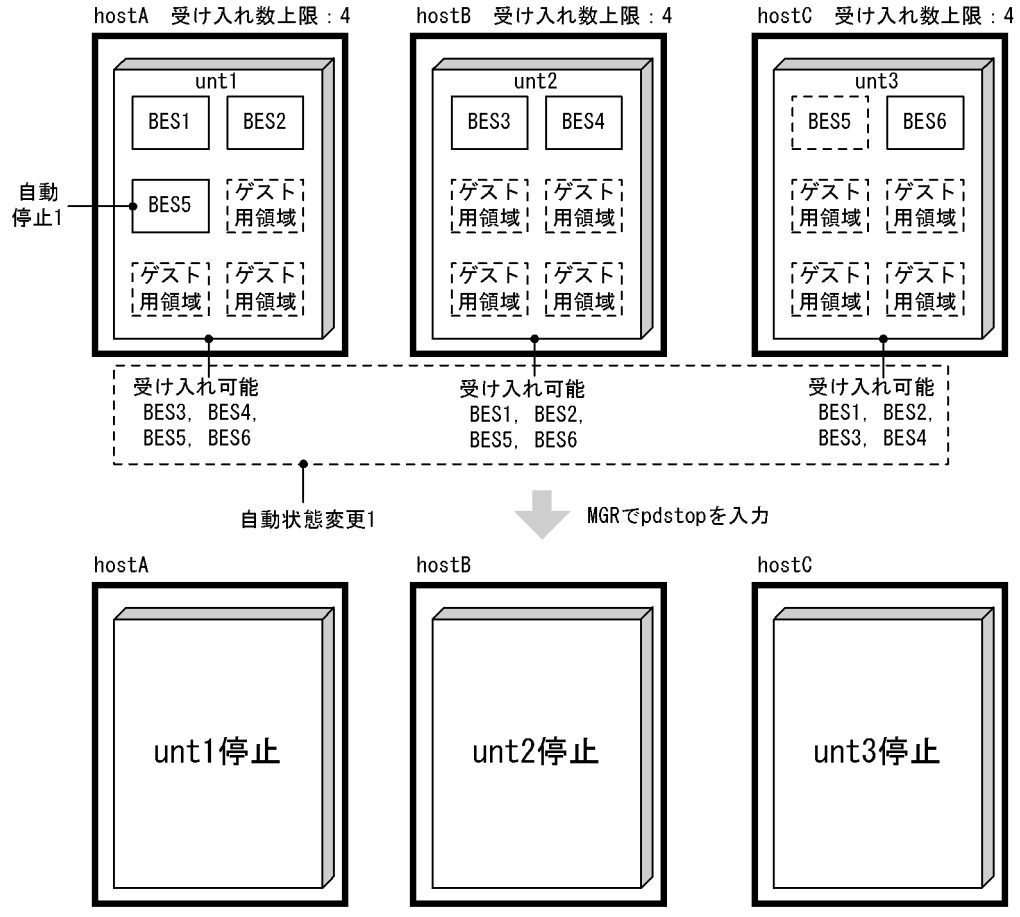
システム停止の例を次の図に示します。この例では，システムを停止すると，次のサーバを停止します。

- ホスト BES
- 実行中ゲスト BES（図の自動停止 1）

また，次のサーバについて受け入れ可能状態を解除します。

- 受け入れ可能状態のゲスト BES（図の自動状態変更 1）

図 26–61　システム停止の例



(b)　ユニット単位の停止

影響分散スタンバイレス型系切り替え機能でのユニットの停止運用を次の表に示します。

表 26–80　影響分散スタンバイレス型系切り替え機能でのユニットの停止運用

入力場所	コマンド	オプション			動作
		-z	-u	-f	
システムマネジャのあるユニット	pdstop	なし	あり	なし	対象ユニットの正常停止
				あり	対象ユニットの強制停止
対象ユニット	pdstop	あり	なし	なし	対象ユニットの強制停止

ユニットを停止する場合は，該当ユニットの全ホスト BES，及び全ゲスト BES に対してサーバ停止（pdstop -q -s サーバ名 （-f） 相当）を実行します。

ユニット内のサーバ状態ごとのユニット正常停止の可否を次の表に示します。影響分散スタンバイレス型系切り替え機能では，ホスト BES，ゲスト BES の種別に関係なくサーバの単独異常停止，強制停止中サーバがあっても，ユニットとしては正常停止できます。

表 26–81　ユニット内のサーバ状態ごとのユニット正常停止の可否

サーバ状態（ホスト BES，ゲスト BES を含む）			ユニット正常停止の可否
起動途中／停止途中※1	待機中※2	停止中※3	影響分散スタンバイレス型系切り替え機能
なし	なし	なし	○
		あり	○
	あり	なし	○
		あり	○
あり	なし	なし	×
		あり	×
	あり	なし	×
		あり	×

（凡例）

○：正常に停止できます。

×：正常に停止できません。

注※1

正常開始途中，再開始途中，正常停止途中，計画停止途中，強制停止途中，異常停止途中

注※2　mon_standby()，又は mon_connect()でブロック中

注※3

正常停止状態，計画停止状態，強制停止状態，異常停止状態，ゲスト BES 停止後未割り当てゲスト用領域となった状態

影響分散スタンバイレス型系切り替え機能でのユニット停止時の各種バックエンドサーバの処理を次の表に示します。

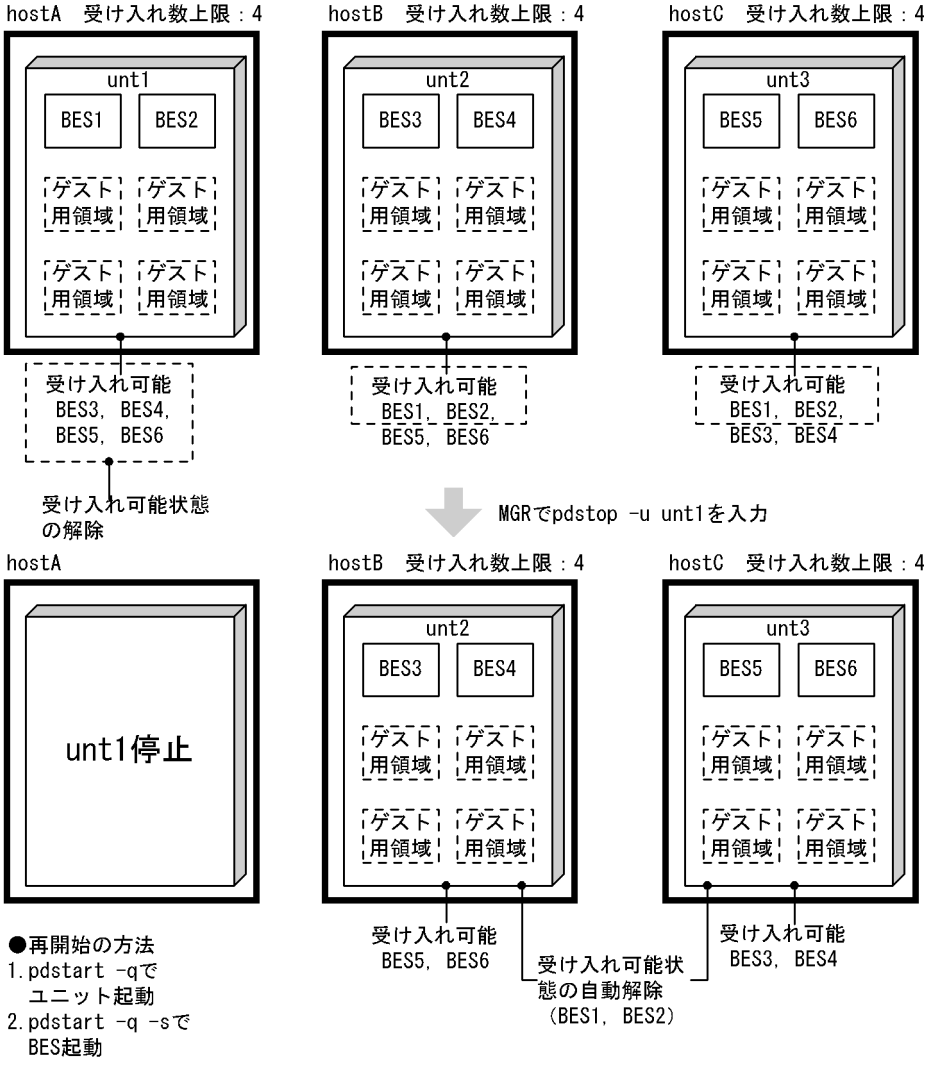
表 26–82　影響分散スタンバイレス型系切り替え機能でのユニット停止時の各種バックエンドサーバの処理

稼働場所	バックエンドサーバの状態	処理
該当ユニット	実行中	停止
	受け入れ可能状態	受け入れ可能状態の解除
他ユニット	実行中	変化なし
	受け入れ可能状態	受け入れ可能状態の自動解除

（例 1）通常時のユニットの停止の例

通常時のユニットの停止の例を次の図に示します。

図 26-62　通常時のユニットの停止の例



非受け入れ中のユニットを停止すると，次のサーバを停止します。

- 該当ユニット内ホスト BES

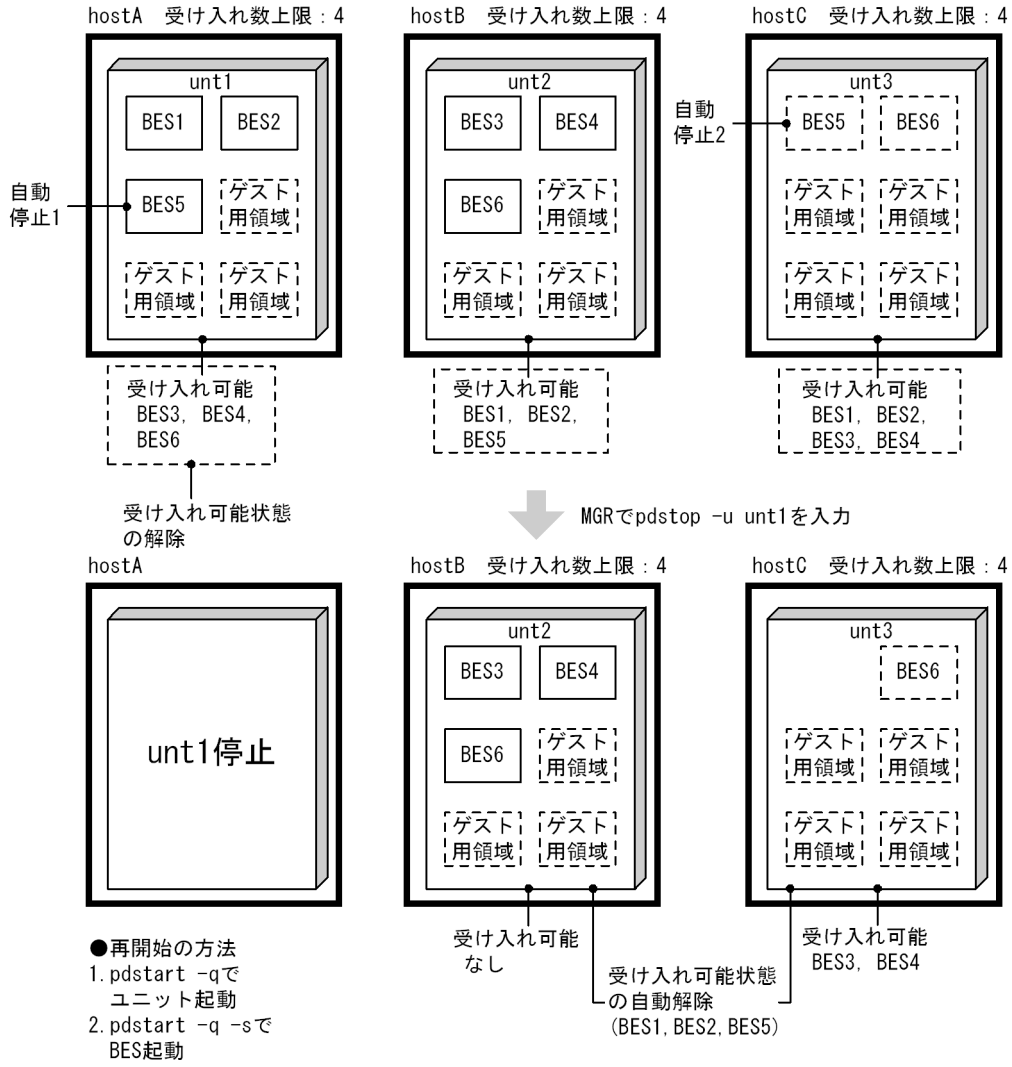
また，次のサーバについて受け入れ可能状態を解除します。

- 該当ユニット内の受け入れ可能状態のゲスト BES（受け入れ可能状態の解除）
- 該当ユニット内ホスト BES に対応する他ユニット内のゲスト BES（受け入れ可能状態の自動解除）

（例 2）受け入れ中のユニットの停止の例

受け入れ中のユニットの停止の例を次の図に示します。

図 26-63　受け入れ中のユニットの停止の例



受け入れ中のユニットを停止すると，次のサーバを停止します。

- 該当ユニット内ホスト BES
- 該当ユニット内で実行中のゲスト BES（自動停止 1）
- 該当ユニットの実行中のゲスト BES に対応する他ユニット内のホスト BES（自動停止 2）

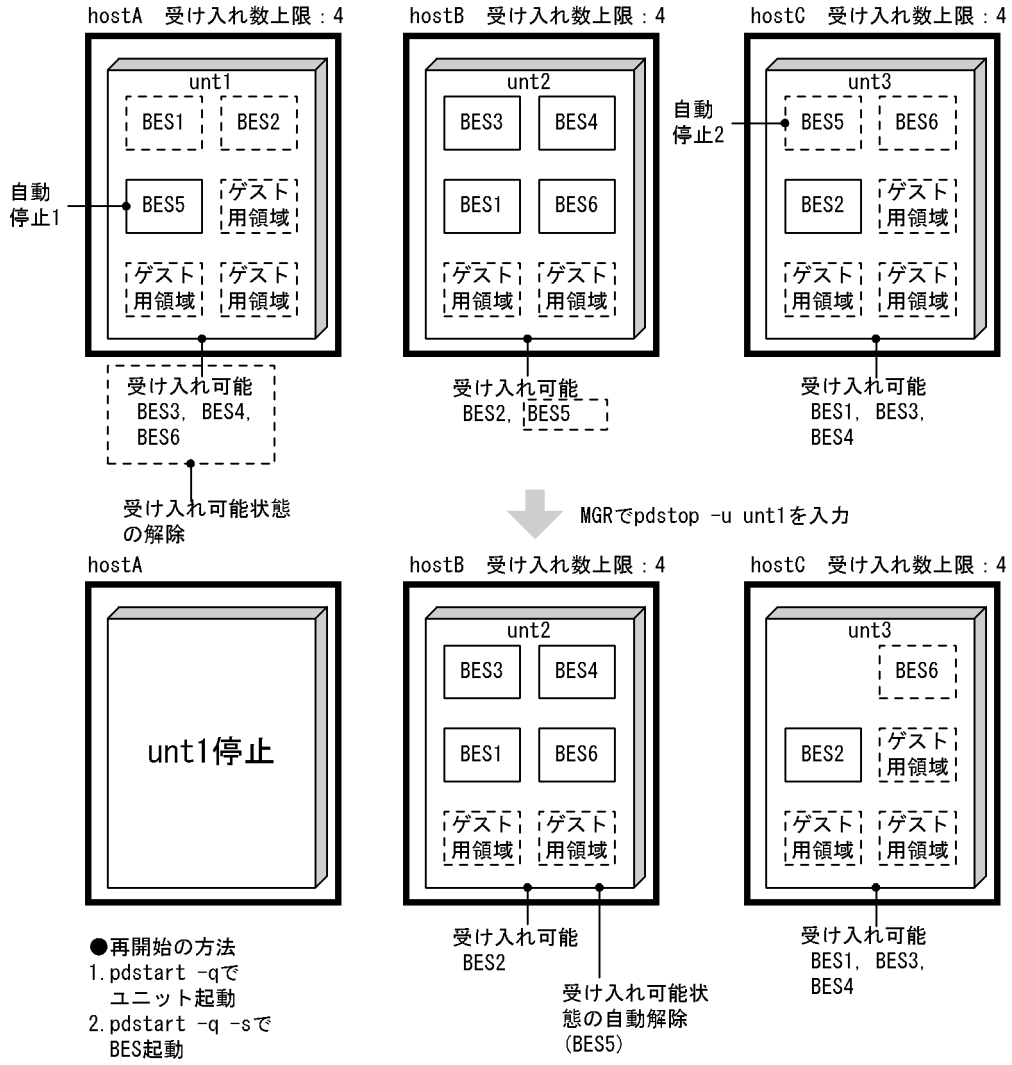
また，次のサーバについて受け入れ可能状態を解除します。

- 該当ユニット内の受け入れ可能状態のゲスト BES（受け入れ可能状態の解除）
- 該当ユニット内のホスト BES に対応する他ユニット内のゲスト BES（受け入れ可能状態の自動解除）
- 該当ユニット内で実行中のゲスト BES に対応する他ユニット内のゲスト BES（受け入れ可能状態の自動解除）

（例 3）ゲスト BES だけのユニットの停止の例

ゲスト BES だけのユニットの停止の例を次の図に示します。

図 26–64　ゲスト BES だけのユニットの停止の例



ゲスト BES だけが実行中のユニットを停止すると，次のサーバを停止します。

- 該当ユニット内ホスト BES
- 該当ユニット内の実行中ゲスト BES（自動停止 1）
- 該当ユニットで実行中のゲスト BES に対応する他ユニット内のホスト BES（自動停止 2）

また，次のサーバについて受け入れ可能状態を解除します。

- 該当ユニット内の受け入れ可能状態のゲスト BES（受け入れ可能状態の解除）
- 該当ユニット内で実行中のゲスト BES に対応する他ユニット内のゲスト BES（受け入れ可能状態の自動解除）

(c)　サーバ単位の停止

影響分散スタンバイレス型系切り替え機能でのサーバの停止運用を次の表に示します。

表 26–83　影響分散スタンバイレス型系切り替え機能でのサーバの停止運用

入力場所	コマンド	オプション				動作
		-z	-u	-s	-f	
システムマネジャのあるユニット	pdstop	なし	なし	あり	なし	対象サーバの停止※2
					あり	全稼働中 HA グループ内ユニットで対象サーバの強制停止※1
			あり	あり	なし	対象サーバの停止※2※3
					あり	対象サーバの強制停止※2
対象ユニット	pdstop	あり	なし	あり	なし	対象サーバの強制停止（ホスト BES）※4

注※1

HA グループ内の全稼働中ユニットでサーバ停止するのは実行ユニットだけで，ほかは受け入れ可能状態を解除します。

注※2

pdstop -s（-f）で実行サーバを停止した場合は HA グループ内全稼働中ユニットで受け入れ可能状態を自動で解除します。Hitachi HA Toolkit Extension の場合も ndm，rdm が連携して待機系（受け入れ可能状態）を解除します。

注※3

サーバ状態ごとのサーバの停止結果を表「サーバ状態ごとのサーバの停止結果」に示します。

注※4

Hitachi HA Toolkit Extension の場合には，pdstop -z -s で実行サーバを停止しても，HA グループ内の他ユニットでは該当サーバに対する受け入れ可能状態を自動解除しません。受け入れ可能状態を解除するには，HA グループ内の全ユニットで Hitachi HA Toolkit Extension の待機系停止コマンド（hatesbystp）を入力してください。

表 26–84　サーバ状態ごとのサーバの停止結果

サーバの状態	開始結果
実行系起動待ち	実行系起動待ちの解除
受け入れ可能状態	受け入れ可能状態の解除
稼働中	サーバ停止

影響分散スタンバイレス型系切り替え機能でのサーバ停止時の各種バックエンドサーバに対する処理を次の表に示します。

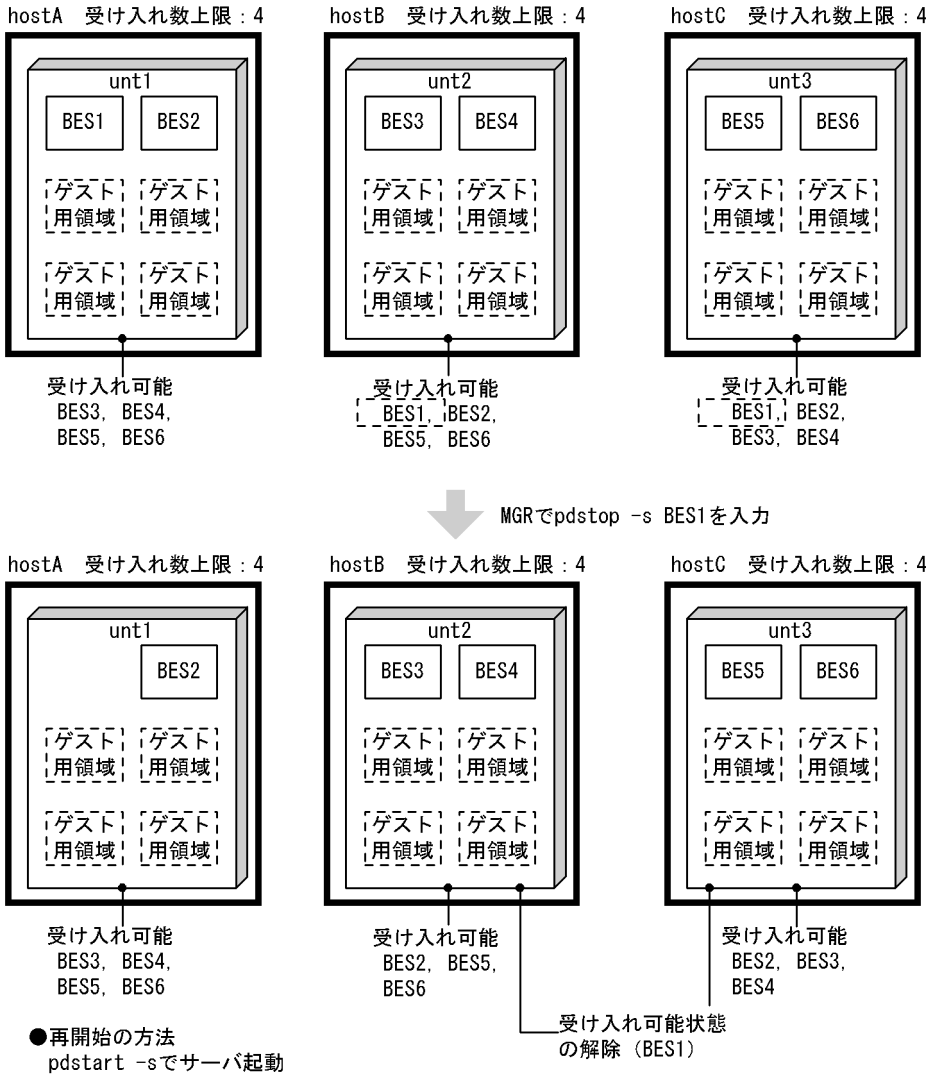
表 26–85　影響分散スタンバイレス型系切り替え機能でのサーバ停止時の各種バックエンドサーバに対する処理

稼働場所	バックエンドサーバの状態	処理
稼働ユニット	操作対象	停止
他ユニット	受け入れ可能状態	受け入れ可能状態の解除

（例 1）ホスト BES の停止の例

ホスト BES の停止の例を次の図に示します。

図 26–65　ホスト BES の停止の例



ホスト BES を停止すると，次のサーバを停止します。

- 該当ホスト BES

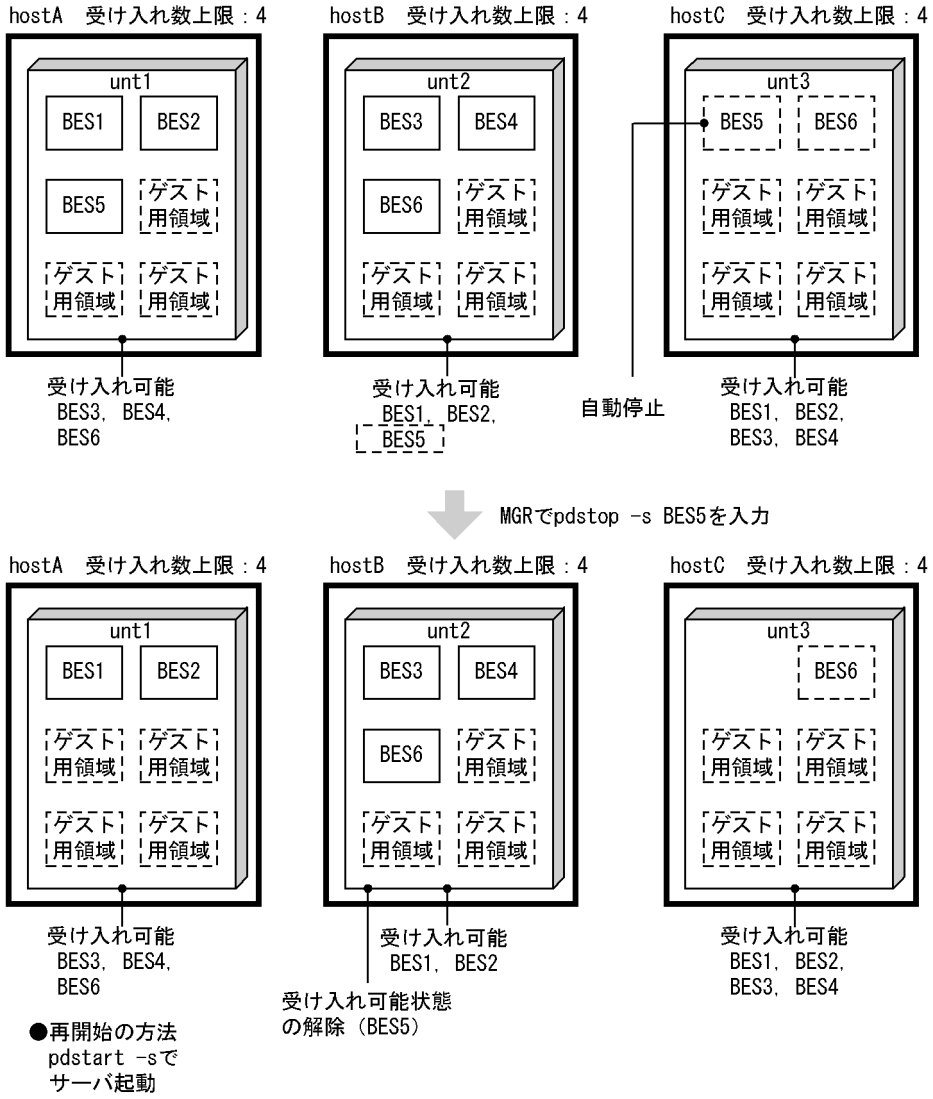
また，次のサーバについて受け入れ可能状態を解除します。

- 該当ホスト BES に対応する他ユニット内のゲスト BES（受け入れ可能状態の解除）

（例 2）ゲスト BES の停止の例

ゲスト BES の停止の例を次の図に示します。

図 26–66　ゲスト BES の停止の例



ゲスト BES を停止すると，次のサーバを停止します。

- 該当実行中ゲスト BES
- 該当実行中ゲスト BES に対応する他ユニット内のホスト BES（自動停止）

また，次のサーバについて受け入れ可能状態を解除します。

- 該当の実行中ゲスト BES に対応する他ユニット内のゲスト BES（受け入れ可能状態の解除）

(4)　影響分散スタンバイレス型系切り替え機能の場合（待機系だけを終了）

待機系だけを終了する場合の停止方法について説明します。

スタンバイ型系切り替え機能，及び 1：1 スタンバイレス型系切り替え機能と同じく，Hitachi HA Toolkit Extension の hatesbystp コマンドによって待機系を終了できます。影響分散スタンバイレス型系切り替え機能では，システムマネジャがあるユニットからの操作もできます。

影響分散スタンバイレス型系切り替え機能での待機系の終了方法を次の表に示します。

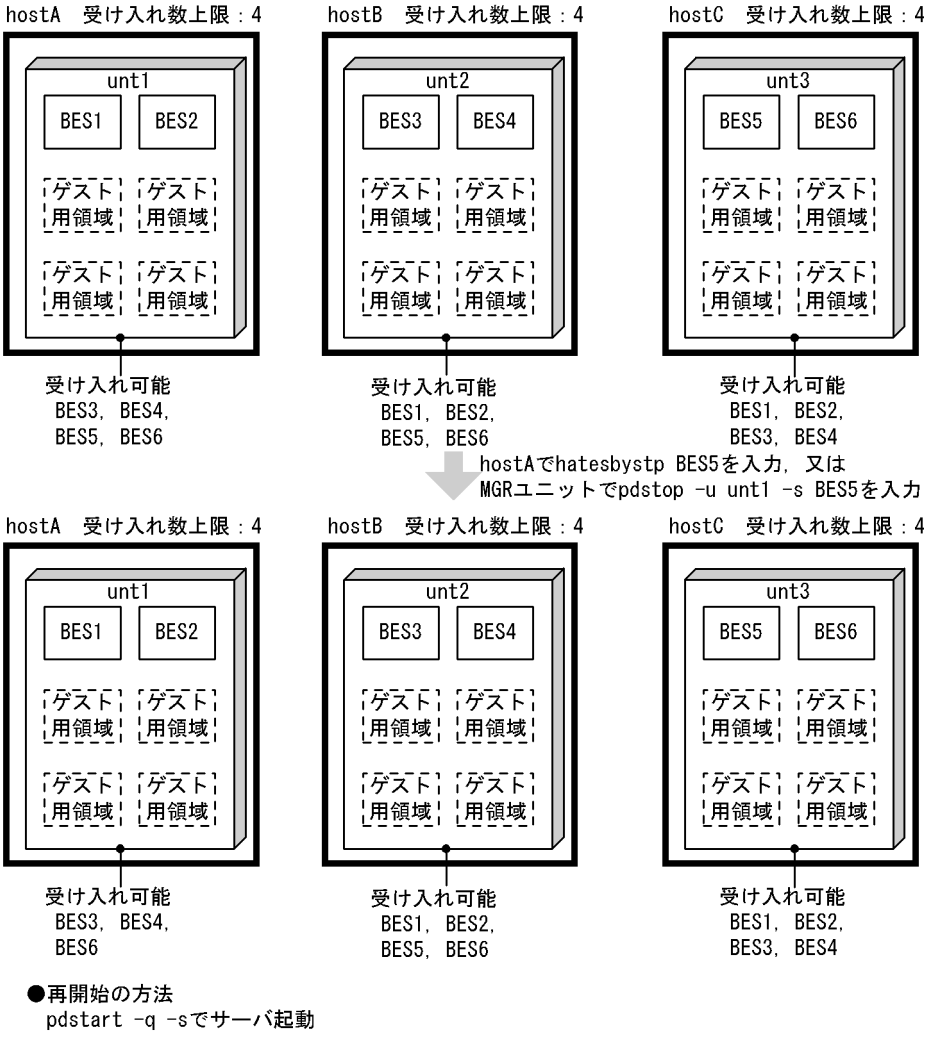
表 26–86　影響分散スタンバイレス型系切り替え機能での待機系の終了方法

入力場所	コマンド	操作対象	動作
操作対象サーバを配置したホスト	hatesbystp	受け入れ可能状態のバックエンドサーバ	ゲスト BES の受け入れ可能状態の解除
システムマネジャのあるユニット	pdstop -u -s		

（例 1）ゲスト BES の受け入れ可能状態の解除の例

ゲスト BES の受け入れ可能状態の解除の例を次の図に示します。

図 26–67　ゲスト BES の受け入れ可能状態の解除**の例**

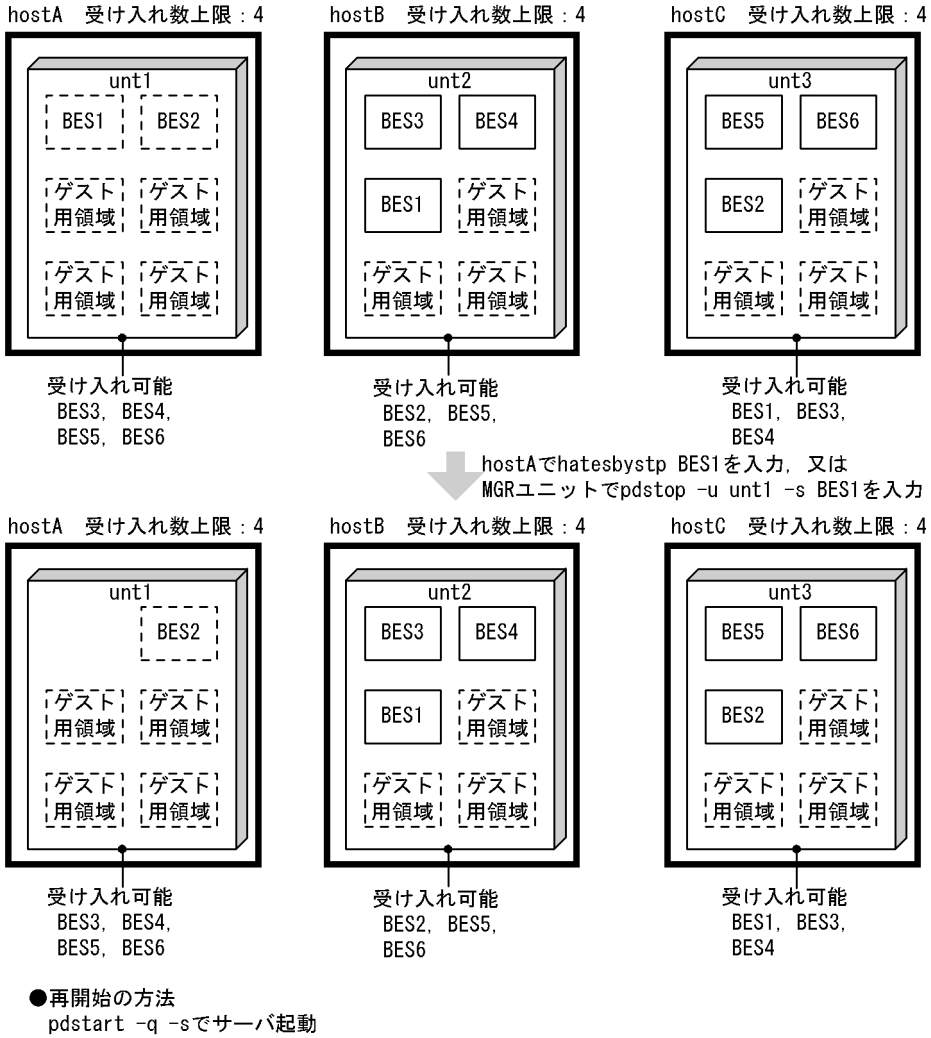


ゲスト BES の受け入れ可能状態を解除するには，Hitachi HA Toolkit Extension の hatesbystp コマンドを入力します。

（例 2）待機系のホスト BES の停止の例

待機系のホスト BES の停止の例を次の図に示します。

図 26–68　待機系のホスト BES の停止の例



待機系のホスト BES を停止するには，Hitachi HA Toolkit Extension の hatesbystp コマンドを入力します。

26.12.4　HiRDB の終了方法（モニタモードの場合）

(1)　クラスタソフトウェアに MSFC を使用する場合

実行系及び待機系の両方を次に示す手順で終了します。

〈手順〉

1. **pdstop** コマンドで実行系 HiRDB を終了します。
2. HiRDB のグループに登録したすべてのリソース（IP アドレス，ネットワーク名，汎用サービス，及び共有ディスク）をクラスタアドミニストレータでオフライン状態にします。

なお，強制停止する場合は，汎用サービスに登録した HiRDB のクラスタサービス（HiRDBClusterService）をオフライン状態にします。

(2)　クラスタソフトウェアに CLUSTERPRO を使用する場合

実行系を次に示す手順で終了します。

〈手順〉

1. モニタリソースにカスタム監視リソースを登録している場合，CLUSTERPRO のコマンド，又は Cluster WebUI の GUI 操作でカスタム監視リソースを一時停止してください。
2. pdstop コマンドで実行系 HiRDB を終了します。

なお，強制停止する場合は，CLUSTERPRO のコマンド，又は Cluster WebUI の GUI 操作で HiRDB ユニットを配置しているフェイルオーバグループを停止してください。

26.12.5　状態の確認

(1)　ユニット，及びサーバの稼働状態

系切り替え機能適用時のユニット，及びサーバの稼働状態確認方法を次の表に示します。

表 26–87　系切り替え機能適用時のユニット，及びサーバの稼働状態確認方法

コマンド	系切り替え機能	出力情報
pdls -d svr	スタンバイ型系切り替え機能	• ホスト名（系切り替え後も現用系ホスト名を表示） • ユニットの稼働状態 • サーバの稼働状態
	1：1 スタンバイレス型系切り替え機能	• ホスト名（系切り替え後も，現用系ホスト名（正規 BES のホスト名）を表示） • ユニットの稼働状態（系切り替え後も，正規 BES ユニットのユニット識別子を表示） • サーバの稼働状態
	影響分散スタンバイレス型系切り替え機能	• ホスト名（系切り替え後は，受け入れユニットのホスト名を表示） • ユニットの稼働状態（系切り替え後は，受け入れユニットのユニット識別子を表示） • サーバの稼働状態（系切り替え後は，受け入れユニットに属するサーバとして表示）

(2)　系の状態確認

系切り替え機能適用時の系の状態の確認方法を次の表に示します。

表 26–88　系切り替え機能適用時の系の状態の確認方法

コマンド	系切り替え機能	出力情報
pdls -d ha	スタンバイ型系切り替え機能	• 現用系のホスト名及び状態（実行中／待機中／停止） • 予備系のホスト名及び状態（実行中／待機中／停止） このコマンドでは，IP アドレスを引き継がない場合だけ待機系のホスト名及び状態を確認できます。※1
	1：1 スタンバイレス型系切り替え機能	• 正規 BES ユニットのホスト名及び状態（実行中／待機中／停止） • 代替 BES ユニットのホスト名及び状態（実行中／待機中／停止）
	影響分散スタンバイレス型系切り替え機能	• バックエンドサーバの配置先ユニット名，状態（実行中／待機中／停止／実行系起動待ち状態）及び定義先ユニット名 バックエンドサーバの詳細な状態は，詳細表示オプション（-a）指定時だけ表示します。
hateshow（Hitachi HA Toolkit Extension 使用時だけ）	スタンバイ型系切り替え機能	• 自系の状態※2
	1：1 スタンバイレス型系切り替え機能	
	影響分散スタンバイレス型系切り替え機能	

注※1

IP アドレスを引き継ぐ場合はクラスタソフトウェアのコマンドで系の状態を確認してください。表示される情報については，各クラスタソフトウェアのマニュアルを参照してください。

注※2　次の分類で状態を表示します。

実行サーバの起動完了／待機サーバの起動完了／実行サーバの起動中／待機サーバの起動中／実行サーバの停止処理中／待機サーバの停止処理中／実行サーバの再起動待ち状態／サーバ未起動

(3)　コマンド又はユティリティを実行できるか確認する方法

現用系のシステムマネジャユニットで pdls -d svr コマンドを実行してください。

- **現用系で実行した pdls -d svr コマンドの終了ステータスが 0 の場合**
  現用系が実行系のため，現用系でコマンド又はユティリティを実行してください。
- **現用系で実行した pdls -d svr コマンドの終了ステータスが 8 の場合，又は pdls -d svr コマンドが実行できない場合（リモートシェル実行不可，ログイン不可など）**
  予備系が実行系の可能性があります。予備系のシステムマネジャユニットで，pdls -d svr コマンドを実行し，予備系が実行系になっていることを確認してください。
- **現用系又は予備系で実行した pdls -d svr コマンドの終了ステータスが 4 の場合**

一部のユニットが停止しているか，HiRDB が開始処理中又は終了処理中の可能性があります。

一部のユニットが停止している場合は，停止中のユニットを開始してください。障害でユニットが停止している場合は，イベントログに出力されたメッセージを参照して障害原因を取り除いた後に，停止中のユニットを開始してください。

HiRDB が開始処理中又は終了処理中の場合は，5 秒程度の間隔で pdls -d svr コマンドを終了ステータスが 4 でなくなるまで繰り返し実行してください。pd_system_complete_wait_time オペランドに指定した時間を目安に，pdls -d svr コマンドを繰り返し実行してください。

26.12.6　統計ログファイルの運用

スタンバイ型系切り替え機能の場合，統計ログファイルは pdstj1 と pdstj2 の二つのファイルから構成されています。これらのファイルは現用系と予備系にそれぞれ自動的に作成されます。したがって，HiRDB 管理者は合計四つのファイルで運用する必要があります。現用系と予備系の統計ログファイルは共用化できません。

1：1 スタンバイレス型系切り替え機能，及び影響分散スタンバイレス型系切り替え機能の場合，統計ログファイルは，二つのファイル（pdstj1，pdstj2）から構成されています。これらのファイルは現用系 HiRDB システムに一組作成されます。切り替え先では受け入れユニットの統計ログファイルを共用するため，予備系用のファイルは作成されません。したがって，HiRDB 管理者は，正規ユニットのファイルと受け入れユニットのファイルで運用してください。

(1)　アンロード統計ログファイルの作成

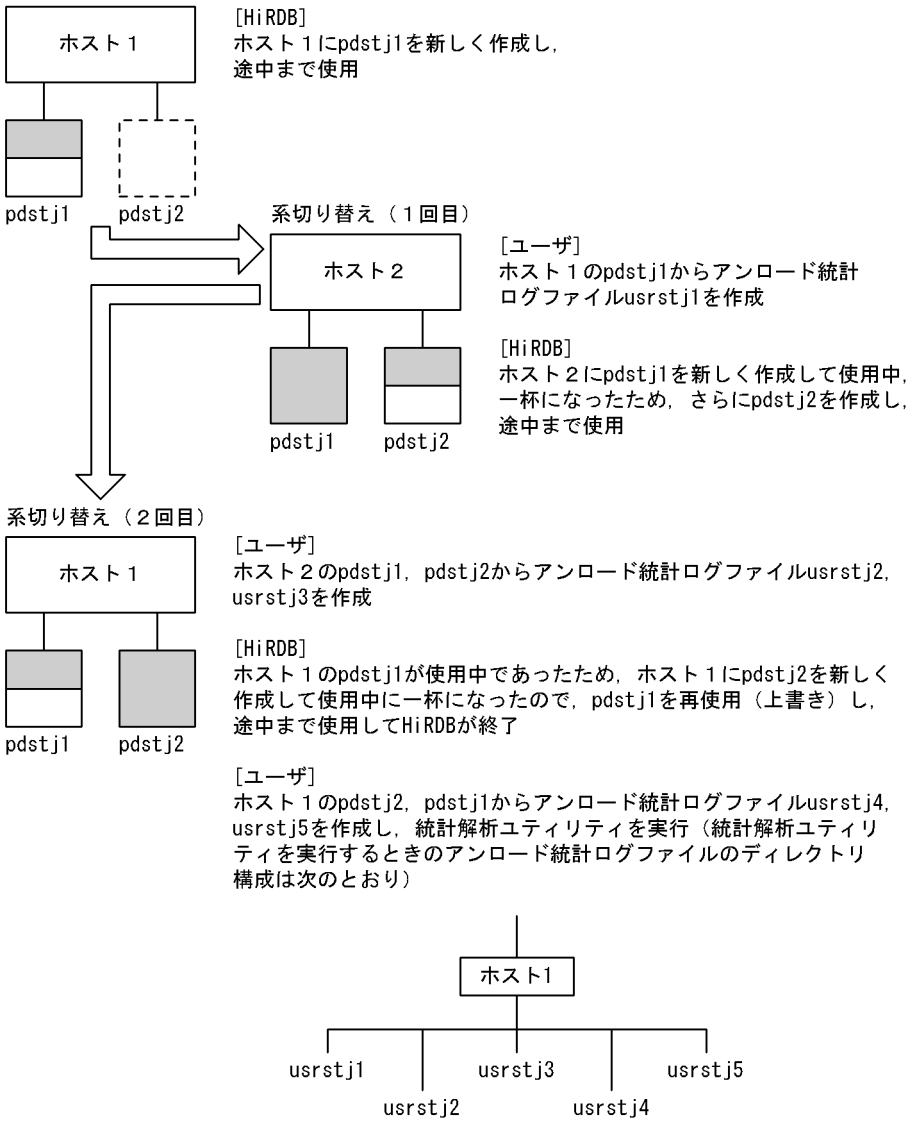
(a)　スタンバイ型系切り替え機能の場合

系切り替えが発生すると，統計ログファイルが各サーバマシンに分散して作成されるため，特定のサーバマシンにアンロード統計ログファイルを作成してください。次に示すタイミングでアンロード統計ログファイルを作成することをお勧めします。

- HiRDB を開始したとき
- 統計ログファイルがスワップしたとき
- 系切り替えが発生したとき

系切り替え機能使用時のアンロード統計ログファイルの作成例（その 1）を次の図に示します。

図 26–69　系切り替え機能使用時のアンロード統計ログファイルの作成例（その 1）



注意事項

統計ログファイル名は各サーバマシンで同じになります。そのまま同じ名称でアンロード統計ログファイルを作成しないようにしてください。

(b)　1：1 スタンバイレス型系切り替え機能，及び影響分散スタンバイレス型系切り替え機能の場合

系切り替えが発生した場合，切り替わり先ホストで使用する統計ログファイルは，切り替え先の受け入れユニットが使用中のファイルです。統計ログ出力先ファイルは，それぞれのホストに分散しているので，アンロード統計ログファイルを特定のサーバマシンに作成してください。

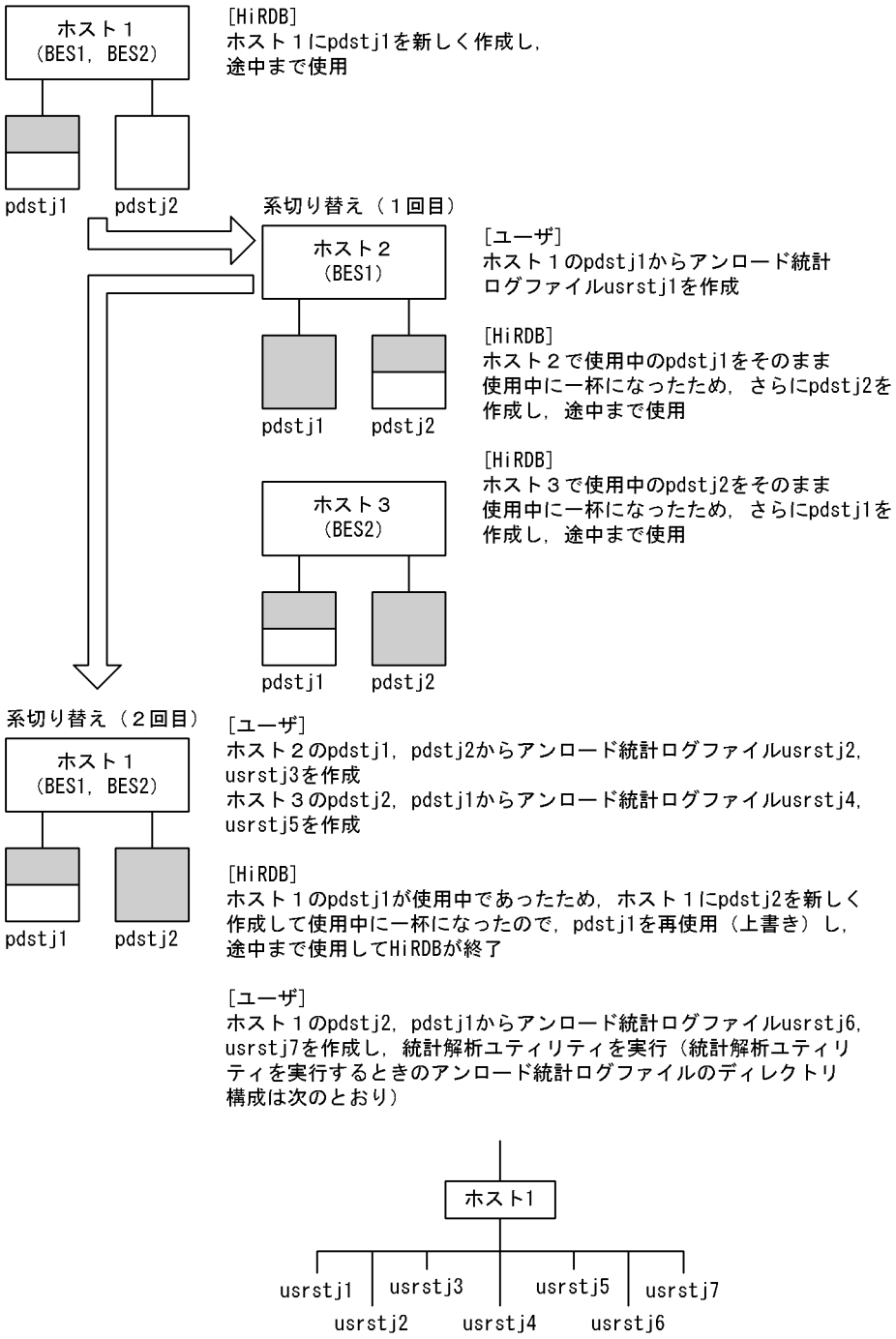
アンロード統計ログファイルを作成するタイミングは，次のとおりです。

- 統計ログファイルがスワップしたとき

- 系切り替えが発生したとき

系切り替え機能使用時のアンロード統計ログファイルの作成例（その 2）を次の図に示します。

図 26–70　系切り替え機能使用時のアンロード統計ログファイルの作成例（その 2）



注意事項

統計ログファイルは各サーバマシンで同じファイル名になるので，そのまま同じ名称でアンロード統計ログファイルを作成しないようにしてください。また，統計ログファイルは系が切り替わると，切り替え先のホストとして扱います。

(2)　系が切り替わった後の統計情報の取得処理

(a)　スタンバイ型系切り替え機能の場合

系の切り替わり後，切り替え先の HiRDB で統計情報を取得するかどうかは，次に示すオペランドの指定に従います。

- pd_statistics
- pdstbegin

pd_statistics オペランドに A 若しくは Y を指定している場合，又は pdstbegin オペランドを指定している場合は，系が切り替わった直後から統計情報を取得します。

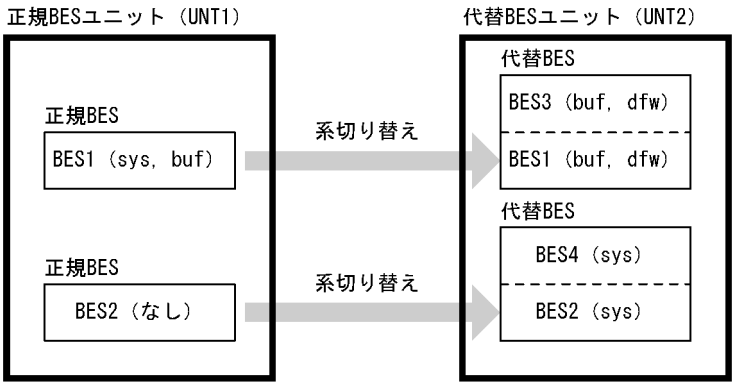
なお，pdstbegin コマンドで統計情報の取得を開始した場合に系切り替えが発生すると，切り替え先の HiRDB では統計情報を取得しません。統計情報を取得するには，切り替え先の HiRDB で pdstbegin コマンドを実行する必要があります。

また，切り替え先の HiRDB で，pdstj1 と pdstj2 のどちらの統計ログファイルを使用するかは，切り替え先の HiRDB が決定します。使用する統計ログファイルの決定手順は系切り替え機能を使用しないときと同じです。

(b)　1：1 スタンバイレス型系切り替え機能の場合

ここでは，代替中の統計情報の取得処理について説明します。代替中の正規 BES の統計情報を取得するかどうかは代替 BES の統計情報の取得有無に従います。取得する統計情報の種類も代替 BES と同じになります。代替中の統計情報の取得処理を次の図に示します。

図 26–71　代替中の統計情報の取得処理



注　（　）内は取得している統計情報の種類を示しています。
sys：システムの稼働に関する統計情報
buf：グローバルバッファに関する統計情報
dfw：デファードライト処理に関する統計情報

〔説明〕

代替中の場合，BES1 と BES2 の統計情報の取得処理は次のようになります。

- BES1：代替 BES（BES3）と同じ buf，dfw を取得します。
- BES2：代替 BES（BES4）と同じ sys を取得します。

BES3 と BES4 については変わりありません。

代替 BES ユニット（UNT2）の統計ログファイルには，BES3 と BES4 の統計情報に加えて，BES1 と BES2 の統計情報が出力されます。

備考

代替中の正規 BES と代替 BES の統計情報の取得有無，及び取得する統計情報の種類が同じになります。図「代替中の統計情報の取得処理」を例にして説明します。

- BES3 の統計情報の取得を pdstend コマンドでやめると，BES1 の統計情報も取得されなくなります。同様に，BES1 の統計情報の取得を pdstend コマンドでやめると，BES3 の統計情報も取得されなくなります。
- BES3 の取得する統計情報の種類を pdstbegin 又は pdstend コマンドで変更すると，BES1 の統計情報の種類も変更されます。同様に，BES1 の取得する統計情報の種類を pdstbegin 又は pdstend コマンドで変更すると，BES3 の統計情報の種類も変更されます。

系を切り戻した場合

系を切り戻した場合（代替中から正常状態に戻した場合），正規 BES の統計情報の取得有無，及び取得する統計情報の種類は次に示すオペランドの指定に従います。

- pd_statistics
- pdstbegin

したがって，pdstbegin コマンドで統計情報の取得有無を変更している場合，又は取得している情報の種類を変更している場合は，再度 pdstbegin コマンドを実行する必要があります。

(c)　影響分散スタンバイレス型系切り替え機能の場合

系切り替えが発生した場合，切り替え前の統計ログの取得状況を引き継ぎます。つまり，系切り替え前に統計ログを取得していた場合は，システム共通定義の pd_statistics オペランドの指定に関係なく，系切り替え後も切り替えたサーバに関する統計ログを取得します。このとき，統計ログファイルは，切り替え先の受け入れユニットと共用します。また，系切り替え前に統計ログを取得していなかった場合は，系切り替え後も統計ログを取得しません。その場合，切り替わった後で pdstbegin コマンドを入力すれば，統計ログの取得を開始できます。

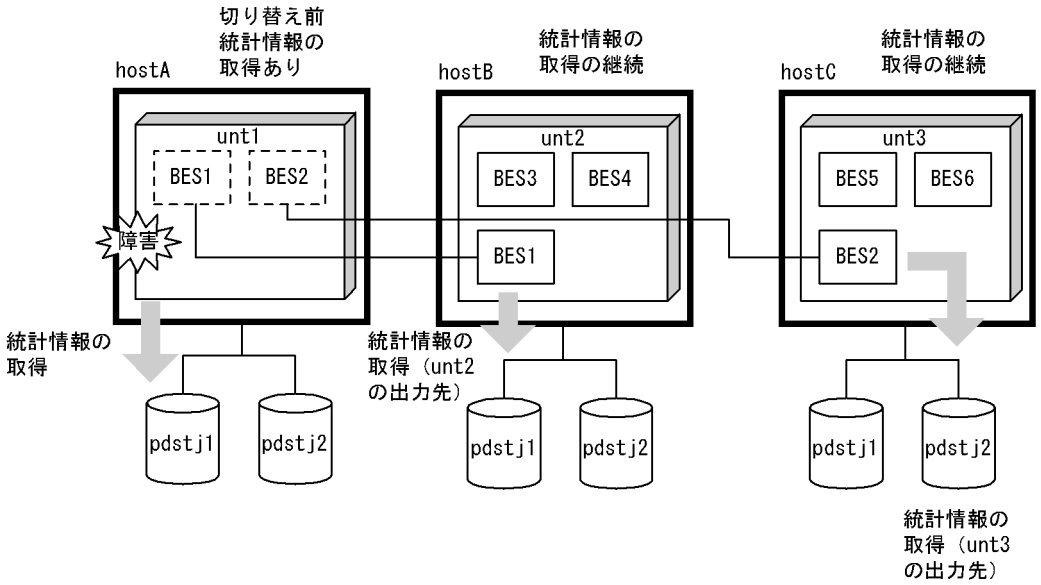
影響分散スタンバイレス型系切り替え機能での統計ログの取得状況を次の表に示します。

表 26–89　影響分散スタンバイレス型系切り替え機能での統計ログの取得状況

ユニット種別	取得の有無	受け入れユニット
正規ユニット	取得中	取得する
	取得していない	取得しない

影響分散スタンバイレス型系切り替え機能の場合の系切り替え後の統計ログの取得の例を次の図に示します。

図 26–72　影響分散スタンバイレス型系切り替え機能の場合の系切り替え後の統計ログの取得の例



(3)　統計解析ユティリティの実行

(a)　スタンバイ型系切り替え機能の場合

作成したアンロード統計ログファイルを入力情報として，統計解析ユティリティを実行してください。ただし，障害などで系が切り替わった場合，切り替わる直前の統計ログはファイルに正しく取得されていません。このため，統計解析ユティリティを実行しても誤差が生じることがありますので，この点を考慮した上でチューニングなどに使用してください。

(b)　1：1 スタンバイレス型系切り替え機能，及び影響分散スタンバイレス型系切り替え機能の場合

正規ユニット，及び受け入れユニットで作成したアンロード統計ログファイルを入力情報として，統計解析ユティリティを実行してください。このとき，系切り替え前のファイルは OS のコマンドなどを使用して手動でコピーする必要があります。なお，系切り替え中のサーバの統計情報は，受け入れユニットに属するサーバの情報として処理されます。

ただし，障害などで系が切り替わった場合，切り替わる直前の統計ログのファイルを正しく取得されていません。このため，統計解析ユティリテイを実行しても誤差が発生することがありますので，この点を考慮した上でチューニングなどに使用してください。

26.12.7　運用上の注意事項

(1)　運用コマンド実行時の制限

- HiRDB の停止中に運用コマンドを実行する場合は，実行系と待機系の両方の HiRDB が終了している必要があります。ただし，pdstart コマンドを除きます。
- スタンバイレス型系切り替え機能の場合は，HiRDB の稼働中に待機系 HiRDB で実行できる運用コマンドはありません。スタンバイ型系切り替え機能の場合は，「スタンバイ型系切り替え機能での運用コマンド実行時の制限」を参照してください。

(2)　スタンバイ型系切り替え機能での運用コマンド実行時の制限

通常は，待機系 HiRDB で運用コマンドを実行しないでください。ただし，待機系 HiRDB を開始するための pdstart コマンド，及び実行系 HiRDB が異常終了し回復できない場合に実行する次のコマンドは除きます。

- 共有ディスク上の HiRDB ファイルシステムを対象とする運用コマンド
- システムログファイル，シンクポイントダンプファイル，ステータスファイルを対象とする運用コマンド

なお，待機系 HiRDB で上記の運用コマンドを実行する場合は，次に示す運用が必要になります。

- 実行系と待機系の両方の HiRDB を終了させる必要があります。pdls -d ust コマンドを実行し，実行系と待機系の HiRDB が終了していることを確認してください。
- OS，又はクラスタソフトウェアのコマンドで，運用コマンドの対象とする共有ディスクを待機系で活性化してから，運用コマンドを実行してください。
- IP アドレスを引き継ぐ系切り替え構成の場合，OS，又はクラスタソフトウェアのコマンドで，運用コマンドを実行するユニットの IP アドレスを待機系で活性化してから，運用コマンドを実行してください。

(3)　影響分散スタンバイレス型系切り替え機能での運用コマンド実行時の制限

OTHER 用グローバルバッファを定義している環境で次に示す運用を行った場合，システムマネジャのあるユニットから pdstop -u コマンドでのユニットの単独正常終了をしないでください。システムマネジャのあるユニットから pdstop コマンドで HiRDB システムを終了させるか，システム構成変更コマンド（pdchgconf コマンド）で追加用 RD エリアにグローバルバッファを割り当てた後，ユニットを終了させてください。OTHER 用グローバルバッファについては，「グローバルバッファの定義（影響分散スタンバイレス型系切り替え機能限定）」を参照してください。

- 影響分散スタンバイレス型系切り替え機能の適用ユニットで，データベース構成変更ユティリティ（pdmod）の create rdarea 文で globalbuffer オペランドを省略し，OTHER 用グローバルバッファのバッファ長より大きなページ長の RD エリアを追加します。

(4) HiRDB システム定義の変更及び HiRDB の構成変更時の注意事項

HiRDB のシステム定義の変更及び HiRDB の構成変更をする場合，クラスタソフトウェアのコマンドで HiRDB を終了しないでください。pdstop コマンドで HiRDB だけを正常終了して，HiRDB システム定義の変更又は HiRDB の構成変更をしてください。その後，pdstart コマンドで HiRDB だけを正常開始してください。

(5) 共有ディスク上に作成した HiRDB ファイルにアクセスできないときの対処方法

クラスタソフトウェアの制御によって HiRDB 停止中は，両方の系から共有ディスク上に作成した HiRDB ファイルを操作できなくなることがあります。この場合，OS のコマンドでディスクを活性化する必要があります。

(6) Hitachi HA Toolkit Extension 使用時の注意事項

Hitachi HA Toolkit Extension では待機系ユニットは実行系ユニットの開始を待ち合わせません。このため，待機系ユニットが開始していない実行系ユニットにコマンドを発行した場合，エラーとなって待機系ユニットがアボート（Phi1012）することがあります。したがって，待機系ユニットは実行系ユニットの開始完了を待ってから開始してください。実行系ユニットの開始完了を待たないで，待機系ユニットを開始すると待機系ユニットがアボート（Phi1012）することがあります。

(7) UAP に関する注意事項

- PDFESHOST オペランドに指定しているフロントエンドサーバのユニットが系切り替え中の場合，その UAP は HiRDB に接続（CONNECT）できません。
- 系切り替えが発生した場合，クライアントとの接続が終了してデータベースの状態はトランザクションを開始する前の状態に戻ります。必要に応じて処理を再実行してください。

26.12.8 スタンバイレス型系切り替え機能使用時の注意事項

(1) 代替 BES ユニット又は代替部を再度開始する必要がある操作（1：1 スタンバイレス型系切り替え機能限定）

次の表に示す操作内容を実施したあと，実行系サーバを配置しているユニットの再起動を実施したかどうかで，代替部を再度開始する必要があります。代替部を再度開始しないと，系切り替えが発生したときに代替部が異常終了します。

表 26–90　代替 BES ユニット又は代替部を再度開始する必要がある操作

分類	条件			代替 BES ユニット（代替部）を再度開始する必要の有無
	操作内容		実行系サーバを配置しているユニットの再起動の有無	
SQL の実行	定義系 SQL の実行		なし	−
			あり	○
	HiRDB システム構築後最初の抽象データ型の定義（CREATE TYPE 文の実行）		なし	○
			あり	○
運用コマンド又はユティリティの実行	データベース構成変更ユティリティ（pdmod）の実行	監査人の登録	なし	−
			あり	−
		RD エリアの属性変更	なし	○
			あり	○
		監査証跡表の作成	なし	−
			あり	○
		上記以外の操作	なし	○※1
			あり	○※1
	HiRDB システム構築後最初の pdplgrgst コマンドの実行（CREATE TYPE 文の実行）		なし	○
			あり	○
システム共通定義の変更	次のどれかの定義の変更 • グローバルバッファの定義（pdbuffer） • RD エリアの最大数（pd_max_rdarea_no） • RD エリアを構成する HiRDB ファイルの最大数（pd_max_file_no） • 表予約数の最低保証値（pd_assurance_table_no） • インデクス予約数の最低保証値（pd_assurance_index_no）		なし	−
			あり	○

（凡例）

○：代替 BES ユニット（代替部）を再度開始する必要があります。

−：代替 BES ユニット（代替部）を再度開始する必要はありません。

注※1

DS を配置しているユニットに高速系切り替え機能を適用している場合，その待機系ユニットの再起動も実施する必要があります。実施していない場合，そのユニットで系切り替えが発生すると異常終了します。

代替部が異常終了した場合は，pdstart -q -c コマンドで代替部を開始してください。

なお，代替中に前記の操作をした場合は，正規 BES ユニットを一度終了してから再度開始してください。正規 BES ユニットを再度開始しないと，系切り替え（系の切り戻し）が発生したときに正規 BES ユニットが異常終了します。正規 BES ユニットが異常終了した場合は，正規 BES ユニットを pdstart -q コマンド又は pdstart -u コマンドで開始してください。

(2)　RD エリアのオープン契機について

(a)　1：1 スタンバイレス型系切り替え機能使用時の場合

1：1 スタンバイレス型系切り替え機能では系の切り替え時間を最小限に抑えるため，系切り替えの発生時に全面回復で必要な RD エリアだけをオープンして，そのほかの RD エリアはオープンしません。したがって，正規 BES 下の RD エリアのオープン契機は次のようになります。

- 系切り替えが発生した場合，系の切り戻し時に代替部の全サーバプロセスが終了しないかぎり系を切り戻せないため，代替部の RD エリアのオープン契機は SCHEDULE 属性になります。
- 障害が回復して正規 BES に系を切り戻した場合，正規 BES 下の INITIAL 又は DEFER 属性の RD エリアのオープン契機は DEFER 属性になります。SCHEDULE 属性の RD エリアは SCHEDULE 属性のままです。

RD エリアのオープン契機については，「RD エリアのオープン契機を変更する方法（RD エリアの属性変更）」を参照してください。

(b)　影響分散スタンバイレス型系切り替え機能使用時の場合

影響分散スタンバイレス型系切り替え機能では，系切り替え時間を最小限に抑えるため，系切り替えの発生時に全面回復で必要な RD エリアだけをオープンして，そのほかの RD エリアはオープンしません。したがって，RD エリアのオープン契機は次のようになります。

- pd_rdarea_open_attribute_use オペランドに N が指定されている場合，指定値に関係なく Y が指定されているものとみなします。
- 系切り替え発生時は，RD エリアオープン契機が INITIAL の場合，指定値に関係なく，DEFER が指定されているものとみなします。

影響分散スタンバイレス型系切り替え機能を使用した場合の RD エリアオープン契機については，マニュアル「HiRDB システム定義」の pd_rdarea_open_attribute_use オペランド，及び pd_rdarea_open_attribute オペランドの説明を参照してください。

26.12.9　パブリッククラウド環境に関する運用時の注意事項

Microsoft Azure 環境上にクラスタ構成を適用する場合の注意事項を次に示します。

(1)　Microsoft Azure 環境に関する注意事項

(a)　Azure ロードバランサー制御使用時の注意事項

1. HiRDB を系切り替えした直後，HiRDB への接続処理がエラーとなります。これは，Azure ロードバランサーが正常性プローブ応答の途絶を認識するまでに最小でも 10 秒程度掛かるためです。この時間は Azure ロードバランサーの正常性プローブの設定によって変化します。

   なお，クライアント環境定義 PDAUTORECONNECT に YES を指定して自動再接続機能を使用すれば，接続処理がエラーになっても自動的に再接続できます。

2. Microsoft Azure の制限に伴い，HiRDB の実行系サーバ上に配置した HiRDB クライアントから，内部ロードバランサーを経由して同一サーバ内の HiRDB に接続することはできません。そのため，UAP や次に示すコマンド・ユティリティを HiRDB の実行系サーバ上で実行する場合は，PDHOST 環境変数に接続する HiRDB サーバのホスト名又は IP アドレスを指定してください。

   - pdsql
   - マニュアル「HiRDB コマンドリファレンス」の「環境変数に設定したクライアント環境変数の制限事項」の「■指定を有効とするクライアント環境変数（その 2）」及び「■指定を有効とするクライアント環境変数（その 3）」に示されるユティリティ

26.13　計画系切り替え

ここでは計画系切り替えの方法について説明します。

26.13.1　スタンバイ型系切り替え機能の場合

計画系切り替えには次に示す三つの方法があります。ご利用の環境に応じて方法を選択してください。

- **方法 1**

  この方法は，サーバモード及びモニタモードの両方に適用できます。

  トランザクションキューイングを行った後に計画系切り替えを実行します。この場合，計画系切り替え中も新規トランザクションがキューイングされるため，計画系切り替え中に UAP を実行してもエラーになりません。

  HiRDB/パラレルサーバの場合は，この方法は次に示す条件を満たすユニットの計画系切り替えだけに適用できます。

  - フロントエンドサーバだけで構成されているユニット
  - システムマネジャとフロントエンドサーバだけで構成されているユニット

  なお，この方法で計画系切り替えを行う場合，自動再接続機能が適用されていることが前提となるため，自動再接続機能が適用されているかどうかを確認してください。自動再接続機能については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

  また，この方法で計画系切り替えを行う場合，pdtrnqing コマンドの実行から計画系切り替えが完了するまでの間，UAP のトランザクションがキューイングされて待ち状態になるため，その分レスポンス時間が長くなります。そのため，次に示す設定値が妥当かどうかを確認してください。

  - クライアント環境定義の PDCWAITTIME オペランドで指定するクライアントの最大待ち時間
  - クライアント環境定義の PDCONNECTWAITTIME オペランドで指定するサーバ接続時の HiRDB クライアントの最大待ち時間

  これらのオペランドで指定した時間を経過すると，UAP の実行がエラーになります。

- **方法 2**

  サーバモードを適用している場合の計画系切り替え方法です。

  トランザクションキューイングを使わないで計画系切り替えを実行します。この場合，計画系切り替え中の新規トランザクションは受け付けられないため，計画系切り替え中に UAP を実行するとエラーになります。

- **方法 3**

  モニタモードを適用している場合の計画系切り替え方法です。

  トランザクションキューイングを使わないで計画系切り替えを実行します。この場合，計画系切り替え中の新規トランザクションは受け付けられないため，計画系切り替え中に UAP を実行するとエラーになります。

(1)　方法 1 で計画系切り替えを行う場合

計画系切り替えの手順を次に示します。

＜手順＞

1. **pdls -d ha** コマンド又はクラスタソフトウェアのコマンドで次に示すことを確認してください。
   ・計画系切り替えの対象ユニットが稼働状態であること
   ・計画系切り替えの切り替え先ユニットが待機状態であること
2. 切り替え元のユニットで **pdtrnqing** コマンドを実行し，トランザクションキューイングを開始してください。
   このとき，pdtrnqing コマンドがリターンコード 0 で正常終了したことを確認してください。リターンコードが 0 の場合は，トランザクションキューイングが正常に開始されています。リターンコードが 8 及び 12 の場合は，トランザクションキューイングが開始されていません。この場合，原因対策後，pdtrnqing コマンドを再実行してください。
3. クラスタソフトウェアのコマンド（MSFC の場合，HiRDB のグループをクラスタアドミニストレータで移動）で計画系切り替えを実行してください。
4. **pdls -d ha** コマンド又はクラスタソフトウェアのコマンドで，切り替え先のユニットが稼働状態であることを確認してください。

計画系切り替えが完了すると，トランザクションキューイングは自動的に解除され，キューイングされていたトランザクションが実行されます。

(2)　方法 2 で計画系切り替えを行う場合

計画系切り替えの手順を次に示します。

＜手順＞

1. **pdstop** コマンドで実行系 HiRDB（ユニット又はサーバ）を終了します。Hitachi HA Toolkit Extension の termcommand オペランドに pdstop コマンドを指定している場合は，2 のコマンドの延長で Hitachi HA Toolkit Extension が pdstop コマンドを実行するため，pdstop コマンドを実行する必要はありません。
2. HiRDB のグループをクラスタアドミニストレータで移動します。
3. 新しく待機系になった HiRDB（又はユニット）を待機状態にする必要があります。待機系 HiRDB に **pdstart** コマンド（HiRDB/パラレルサーバの場合は **pdstart -q** コマンド）を実行してください。

(3)　方法 3 で計画系切り替えを行う場合

計画系切り替えの手順を次に示します。

＜手順＞

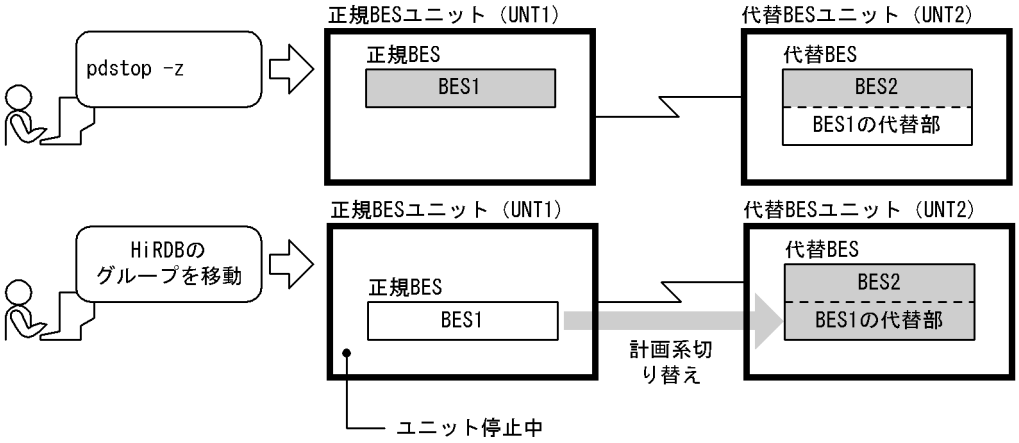
クラスタソフトウェアに CLUSTERPRO を使用する場合は手順 2 だけ実行してください。

1. **pdstop** コマンドで実行系 HiRDB（又はユニット）を終了します。
2. クラスタソフトウェアのコマンド（HiRDB のグループをクラスタアドミニストレータで移動）で計画系切り替えを実行してください。

26.13.2　1：1 スタンバイレス型系切り替え機能の場合

代替 BES ユニットに計画系切り替えをする場合は，代替部が待機状態である必要があります。正規 BES ユニットに系を切り戻す場合は，正規 BES ユニットが待機状態である必要があります。計画系切り替えの手順を次に示します。

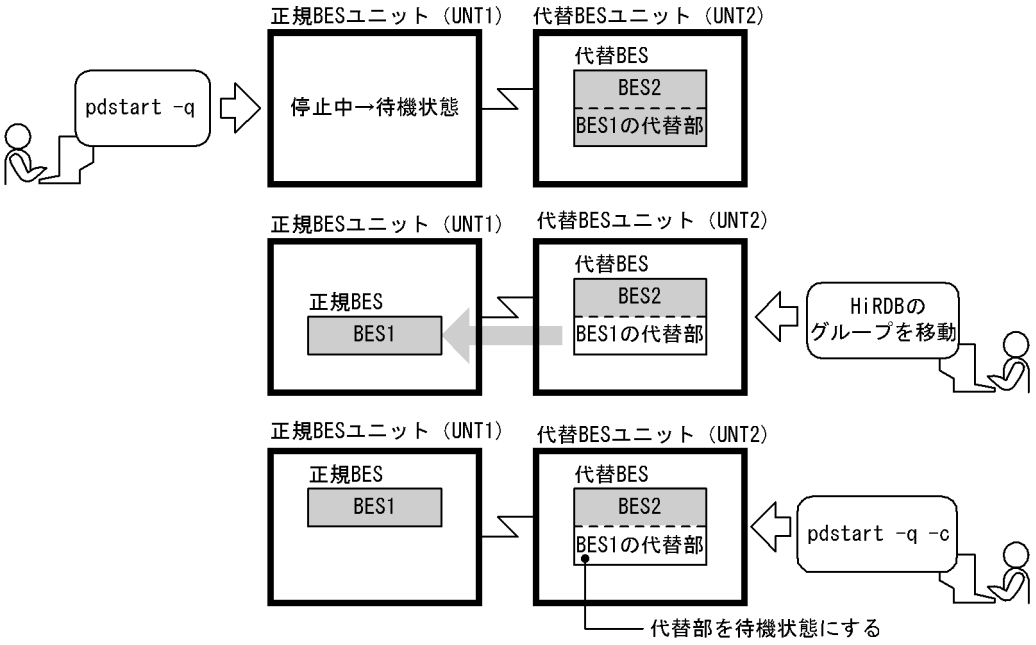
〈手順〉代替 BES ユニットに系を切り替える場合



1. **pdstop -z** コマンドで正規 BES ユニットを終了します。Hitachi HA Toolkit Extension の termcommand オペランドに pdstop コマンドを指定している場合は，2 のコマンドの延長で Hitachi HA Toolkit Extension が pdstop コマンドを実行するため，pdstop コマンドを実行する必要はありません。
2. HiRDB のグループをクラスタアドミニストレータで移動します。計画系切り替えの方法については，各クラスタソフトウェアのマニュアルを参照してください。

前記の手順で計画系切り替えをすると，代替 BES ユニットで正規 BES ユニットの処理が行われます。正規 BES ユニットは停止中になります。正常状態に戻す場合は次に示す手順で再度計画系切り替えをしてください。

〈手順〉正規 BES ユニットに系を切り戻す場合



1. **pdstart -q** コマンドで正規 BES ユニットを待機状態にします。
2. HiRDB のグループをクラスタアドミニストレータで移動します。このとき，代替部の待機状態が解除されます。
3. **pdstart -q -c** コマンドで代替部を待機状態にします。

26.13.3　影響分散スタンバイレス型系切り替え機能の場合

計画系切り替えには次に示す二つの方法があります。ご利用の環境に応じて方法を選択してください。

- **方法 1**

  トランザクションキューイングを行った後に計画系切り替えを実行します。この場合，計画系切り替え中も新規トランザクションがキューイングされるため，計画系切り替え中に UAP を実行してもエラーになりません。

  なお，この方法で計画系切り替えを行う場合，pd_ha_transaction オペランドに queuing を指定している必要があります。

  また，この方法で計画系切り替えを行う場合，pdtrnqing コマンドの実行から計画系切り替えが完了するまでの間，UAP のトランザクションがキューイングされて待ち状態になるため，その分レスポンス時間が長くなります。そのため，次に示す設定値が妥当かどうかを確認してください。

  - クライアント環境定義の PDCWAITTIME オペランドで指定するクライアントの最大待ち時間

  このオペランドで指定した時間を経過すると，UAP がエラーになります。

- **方法 2**

トランザクションキューイングを使わないで計画系切り替えを実行します。この場合，計画系切り替え中の新規トランザクションは受け付けられないため，計画系切り替え中に UAP を実行するとエラーになります。

(1)　方法 1 で計画系切り替えを行う場合

計画系切り替えの手順を次に示します。

＜手順＞

1. **pdls -d ha -s サーバ名 -a** コマンドで次に示すことを確認してください。
   ・計画系切り替えの対象となるバックエンドサーバの切り替え元ユニットが稼働状態であること
   ・計画系切り替えの対象となるバックエンドサーバの切り替え先ユニットが待機状態であること
2. 切り替え元のユニットで **pdtrnqing** コマンドを実行し，トランザクションキューイングを開始してください。
   このとき，pdtrnqing コマンドがリターンコード 0 で正常終了したことを確認してください。リターンコードが 0 の場合は，トランザクションキューイングが正常に開始されています。リターンコードが 0 以外の場合は，トランザクションキューイングが開始されていません。この場合，原因対策後，pdtrnqing コマンドを再実行してください。
3. HiRDB のグループをクラスタアドミニストレータで移動します。
4. **pdls -d ha -s サーバ名 -a** コマンドで，バックエンドサーバが切り替え先ユニットで稼働状態であることを確認してください。

計画系切り替えが完了すると，トランザクションキューイングは自動的に解除され，キューイングされていたトランザクションが実行されます。何らかの原因でトランザクションキューイングが解除されなかった場合は，切り替え先ユニットで，**pdtrnqing -d -s サーバ名**コマンドを実行し，トランザクションキューイングを解除してください。このとき，pdtrnqing コマンドがリターンコード 0 で正常終了したことを確認してください。

(2)　方法 2 で計画系切り替えを行う場合

クラスタソフトウェアの計画系切り替えコマンドを入力します。

クラスタソフトウェアの計画系切り替えコマンド実行後，切り替え元ユニットで該当バックエンドサーバは自動的に待機状態になります。

影響分散スタンバイレス型系切り替え機能での計画系切り替えの動作を次の表に示します。

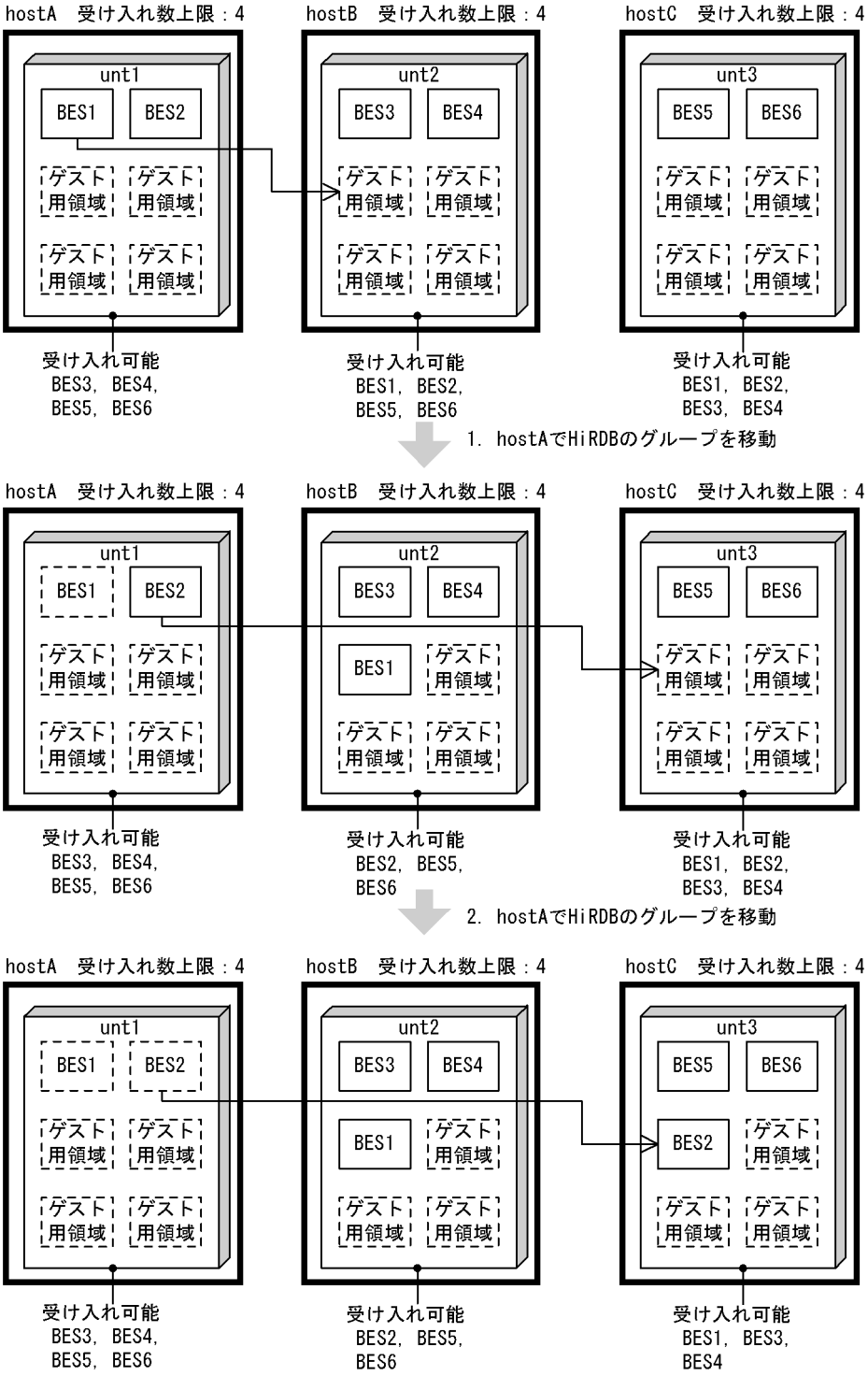
表 26–91　影響分散スタンバイレス型系切り替え機能での計画系切り替えの動作

バックエンドサーバの種別	動作	移動先
ホスト BES	最も優先度の高い切り替え先に移動します。ゲスト用領域に空きがない場合はその次に優先度の高い切り替え先に移動します。	ゲスト BES
ゲスト BES		ホスト BES

(a)　ホスト BES に対する計画系切り替え

ホスト BES に対する計画系切り替えの例を次の図に示します。この例は，ホストのハードウェアメンテナンスなどのためにサーバマシン（本例では hostA）を停止する必要がある場合の運用例です。

図 26–73　ホスト BES に対する計画系切り替えの例



この例では，unt1 上の現用 BES に対して計画系切り替えをします。手順は次のとおりです。

1. hostA で HiRDB のグループを移動し，BES1 を hostB に切り替えます。

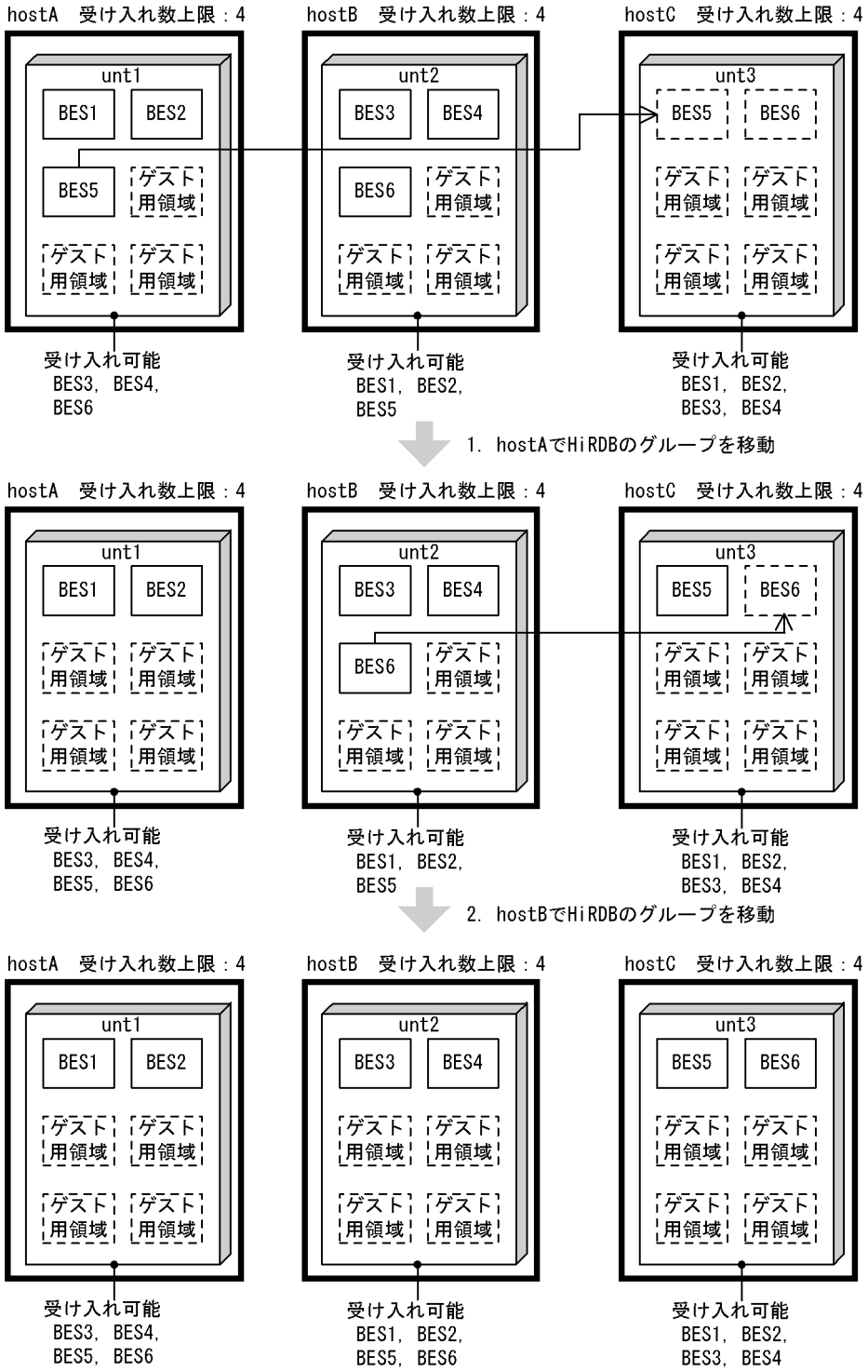
2. hostA で HiRDB のグループを移動し，BES2 を hostC に切り替えます。

なお，Hitachi HA Toolkit Extension 使用時は，クラスタソフトウェアの計画系切り替えコマンドを入力します。

(b) ゲスト BES の計画系切り替え（系の切り戻し）

ゲスト BES に対する計画系切り替え（系の切り戻し）の例を次の図に示します。この例は，障害ユニットが復旧した場合の運用例です。

図 26–74　ゲスト BES に対する計画系切り替え（系の切り戻し）の例



この例では，unt3 から移動した BES5 と BES6 を計画系切り替えで切り戻します。手順は次のとおりです。

1. hostA で HiRDB のグループを移動し，BES5 を hostC に切り替えます。
2. hostB で HiRDB のグループを移動し，BES6 を hostC に切り替えます。

なお，Hitachi HA Toolkit Extension 使用時は，クラスタソフトウェアの計画系切り替えコマンドを入力します。

26.14　連動系切り替え

OpenTP1 など，ほかのシステムと連携しているときの系切り替え方法です。MSFC に設定する依存関係で連動系切り替えの定義をします。MSFC 以外のクラスタソフトウェアで連動系切り替えを実行する場合は，各クラスタソフトウェアのマニュアルを参照してください。待機系に系を切り替えるタイミングは，MSFC に設定する依存関係，及び Hitachi HA Toolkit Extension の server 定義文の switchtype オペランドの指定によって異なります。系を切り替える場合の HiRDB 管理者の操作を次に示します。

- 自動系切り替えをする場合，HiRDB 管理者の操作は必要ありません。
- 計画系切り替えをする場合，MSFC のサービス移動操作をします。

26.15　障害発生時の HiRDB 管理者の処置

26.15.1　HiRDB 管理者の処置

(1)　障害発生時のシステムの処理と HiRDB 管理者の処置

障害発生時のシステムの処理と HiRDB 管理者の処置を次の表に示します。

表 26–92　障害発生時のシステムの処理と HiRDB 管理者の処置（系切り替え機能使用時）

障害の種類		システムの処理	HiRDB 管理者の処置
サーバ障害が発生しました（HiRDB だけが異常終了しました）。	サーバモードの場合	Hitachi HA Toolkit Extension の switchtype オペランド※の指定に従います。	出力されたメッセージを参照して，障害原因を調査します。「障害が発生したときの対処方法」を参照して対策してください。
	モニタモードの場合	系を切り替えません。障害が発生した系で HiRDB を再開始します。 次のどちらかの設定を行う場合，系を切り替えます。 • クラスタソフトウェアに MSFC を使用して，HiRDB のサービスを MSFC の汎用サービスに登録した場合，又は HiRDB が再開始できない場合 • クラスタソフトウェアに CLSTERPRO を使用して，カスタム監視リソースでフェイルオーバを行う場合	
系障害が発生しました。		系を切り替えて，切り替え先の系で HiRDB を再開始します。ただし，系障害の種類によっては系を切り替えないこともあります。系を切り替えるかどうかはクラスタソフトウェアの仕様に依存します。クラスタソフトウェアのマニュアルを参照してください。	出力されたメッセージを参照して，障害原因を調査します。「障害が発生したときの対処方法」又は障害原因のマニュアルを参照して対策してください。
HiRDB の再開始処理に失敗しました。	サーバモードの場合	Hitachi HA Toolkit Extension の switchtype オペランド※の指定に従います。	
	モニタモードの場合	システムを停止します。	
系切り替え処理に失敗しました。		システムを停止します。	

注

系が切り替わったときに引き継ぐ情報は，HiRDB の再開始時に引き継ぐ情報と同じです。HiRDB の再開始時に引き継ぐ情報については，「HiRDB が再開始するときに引き継ぐ情報」を参照してください。

注※

switchtype オペランドの指定によって処理が異なります。switchtype オペランドについては，「Hitachi HA Toolkit Extension に関する準備（サーバモード限定）」を参照してください。

(2)　1：1 スタンバイレス型系切り替え機能を使用している場合

障害対策後，計画系切り替えで正規 BES ユニットに系を切り戻してください（代替中から正常状態に戻してください）。計画系切り替えの方法については，「1：1 スタンバイレス型系切り替え機能の場合」を参照してください。

(3)　影響分散スタンバイレス型系切り替え機能を使用している場合

障害対策後，計画系切り替えで正規ユニットに系を切り戻してください（待機中から正常状態に戻してください）。計画系切り替えの方法については，「影響分散スタンバイレス型系切り替え機能の場合」を参照してください。

26.16　系が切り替わった後の運用方法

26.16.1　系が切り替わった後に行うこと

IP アドレスを引き継ぐ場合は現用系と予備系のホスト名（仮想ネットワーク名）が同じであるため，系切り替えが発生しても特に運用は変わりません。IP アドレスを引き継がない場合は現用系と予備系のホスト名が異なるため，次に示す運用が必要になります。

1：1 スタンバイレス型系切り替え機能の場合も次に示す運用が必要になります。また，影響分散スタンバイレス型系切り替え機能を使用する場合も，IP アドレスを引き継ぎません。サーバごとに切り替え先が異なるため，切り替え後は切り替え先のホスト名を使用します。このため，次に示す運用が必要です。

(1)　運用コマンド及びユティリティ

(a)　スタンバイ型系切り替え機能，又は 1：1 スタンバイレス型系切り替え機能の場合

- 系が切り替わった後も，運用コマンドやユティリティに指定するホスト名又はユニット識別子を変更する必要はありません。ホスト名には現用系（又は正規 BES ユニット）のホスト名を常に指定します。pdunit オペランドの-x オプションで指定したホスト名を指定します。
- できるだけユニット識別子を指定してください。そうすれば，ホスト名を意識する必要がありません。
- HiRDB/シングルサーバの場合は，運用上のトラブルを避けるためホスト名を省略することをお勧めします。
- 運用コマンドの処理結果には現用系のホスト名が表示されることがあります。1：1 スタンバイレス型系切り替え機能の場合は代替 BES ユニットのホスト名が表示されます。

(b)　影響分散スタンバイレス型系切り替え機能の場合

サーバが稼働中の場合は系切り替えが発生すると，障害ユニットに定義されている各サーバは受け入れユニットに属するサーバとして運用する必要があります。サーバごとに切り替え先が異なるため，系切り替えが発生すると正規ユニット単位の運用はできません。

このため，運用コマンドを発行する場合には，次の表に示す指定値を指定します。

表 26–93　影響分散スタンバイレス型系切り替え機能での運用コマンドの指定値

コマンド種別	指定値	
	切り替え前	切り替え後
ホスト名指定コマンド	正規ユニット側のホスト名	受け入れユニット側ホスト名
ユニット識別子指定コマンド	正規ユニットのユニット識別子	受け入れユニットのユニット識別子
サーバ名指定コマンド	操作対象サーバ名	操作対象サーバ名

ホスト名を指定した運用，及びユニット識別子を指定した運用では系切り替え前後で処理対象が異なります。したがって，影響分散スタンバイレス型系切り替え機能ではサーバ名指定の運用を推奨します。

運用コマンドの処理結果にホスト名が表示される場合，受け入れユニット側のホスト名が表示されます。

影響分散スタンバイレス型系切り替え機能での運用コマンドオプション指定時の実行対象を次の表に示します。ただし，pdstart コマンド，及び pdstop コマンドは，表とは異なります。pdstart，pdstop コマンドについては，「HiRDB の運用方法の違いは？」を参照してください。

表 26–94　影響分散スタンバイレス型系切り替え機能での運用コマンドオプション指定時の実行対象

オプション		条件		コマンド動作モード※	コマンドの実行対象
-u	-s	システムマネジャのあるユニット	対象サーバ		
なし	なし	－	－	－	システム（又は，システム内全実行バックエンドサーバ）
	あり	オフライン	－	オフライン	正規ユニットでの現用系のバックエンドサーバとしての指定サーバ
		オンライン	オフライン	オフライン	正規ユニットでの現用系のバックエンドサーバとしての指定サーバ
			オンライン	オンライン	実行ユニットでの実行バックエンドサーバとしての指定サーバ
あり	なし	－	－	－	指定ユニット（又は，指定ユニット内全実行バックエンドサーバ）
	ホスト	オフライン	－	オフライン	指定正規ユニットでのホスト BES としての指定サーバ
		オンライン	オフライン	オフライン	指定正規ユニットでのホスト BES としての指定サーバ
			オンライン	オンライン	実行ユニットでの実行 BES としての指定サーバ（-u 指定を無視します）
	ゲスト	オフライン	－	オフライン	指定受け入れユニットでのゲスト BES としての指定サーバ
		オンライン	オフライン	オフライン	指定受け入れユニットでのゲスト BES としての指定サーバ
			オンライン	オンライン	実行ユニットでの実行バックエンドサーバとしての指定サーバ（-u 指定を無視します）

（凡例）　－：該当しません。

注※

システムマネジャのないユニットにサブコマンドを入力するときに実行系を決定するためのモードです。

(2)　メッセージ

スタンバイ型系切り替え機能の場合，メッセージには現用系のホスト名が表示されることがあります。

1：1 スタンバイレス型系切り替え機能の場合，正規 BES ユニットに対する処理のメッセージには正規 BES ユニットのホスト名又はユニット識別子が表示されます。

影響分散スタンバイレス型系切り替え機能の場合，メッセージは受け入れユニット側のホスト名が表示されます。

(3)　統計情報の取得

系が切り替わった後の統計情報の取得処理については，「系が切り替わった後の統計情報の取得処理」を参照してください。

(4)　クライアント環境定義（スタンバイ型系切り替え機能限定）

(a)　HiRDB/シングルサーバの場合

PDHOST オペランドに現用系のホスト名だけを指定した場合にここで説明する運用が必要です。PDHOST オペランドに現用系及び予備系の二つのホスト名を指定した場合は必要ありません。

PDHOST オペランドには接続する HiRDB のホスト名を指定するため，系が切り替わった場合に PDHOST オペランドの指定を実行系のホスト名に変更する必要があります。したがって，系が切り替わった場合，HiRDB 管理者はクライアントユーザに HiRDB のホスト名が替わったことを知らせてください。クライアントユーザは PDHOST オペランドの指定を変更してください。その後，UAP を再実行してください。この操作をしないと UAP を実行できません。PDHOST オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(b)　HiRDB/パラレルサーバの場合

次に示すオペランドに現用系のホスト名だけを指定した場合にここで説明する運用が必要です。次に示すオペランドに現用系及び予備系の二つのホスト名を指定した場合は必要ありません。

- **PDHOST**
- **PDFESHOST（マルチフロントエンドサーバを使用する場合）**

これらのオペランドには接続する HiRDB のホスト名（PDHOST にはシステムマネジャのホスト名を PDFESHOST にはフロントエンドサーバのホスト名を指定します）を指定するため，系が切り替わった場合にオペランドの指定を実行系のホスト名に変更する必要があります。したがって，系が切り替わった場合，HiRDB 管理者はクライアントユーザに HiRDB のホスト名が替わったことを知らせてください。クライアントユーザはこれらのオペランドの指定を変更してください。その後，UAP を再実行してください。この操作をしないと UAP を実行できません。PDHOST 及び PDFESHOST オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

備考

- システムマネジャのユニットに系切り替えが発生したときだけ，PDHOST オペランドの指定を変更する必要があります。
- フロントエンドサーバのユニットに系切り替えが発生したときだけ，PDFESHOST オペランドの指定を変更する必要があります。
- システムマネジャ又はフロントエンドサーバを定義していないユニットの系切り替えが発生しても，PDHOST 及び PDFESHOST オペランドの指定を変更する必要はありません。

26.17　系の切り替え時間の短縮（ユーザサーバホットスタンバイ，高速系切り替え機能）

系の切り替え時間を短縮する機能として次に示す機能があります。これらの機能を使用するにはサーバモードでの運用が前提となります。

- ユーザサーバホットスタンバイ
- 高速系切り替え機能

26.17.1　ユーザサーバホットスタンバイ

系切り替えが発生したとき，待機系 HiRDB を開始するのに次に示す処理が実施されます。

- システムサーバの起動処理
- システムファイルの引き継ぎ処理
- サーバプロセスの起動処理
- ロールフォワード処理

この中のサーバプロセスの起動処理に掛かる時間は，系の切り替え時間の中で大きな割合を占めています。サーバプロセスの起動処理に掛かる時間はサーバプロセスの常駐数に比例するため，常駐数が多いと系の切り替え時間が長くなります。そこで，待機系 HiRDB のサーバプロセスをあらかじめ起動しておいて，系の切り替え時にサーバプロセスの起動処理をしないようにします。系の切り替え時にサーバプロセスの起動処理がない分，系の切り替え時間を短縮できます。これを**ユーザサーバホットスタンバイ**といいます。例えば，100MIPS ぐらいのサーバマシンでサーバプロセスを一つ起動するのに約 1 秒掛かります。その分，系の切り替え時間を短縮できます。

ユーザサーバホットスタンバイを使用するには，**pd_ha_server_process_standby** オペランドに Y を指定します。

26.17.2　高速系切り替え機能

待機系 HiRDB のサーバプロセス及びシステムサーバをあらかじめ起動しておいて，系の切り替え時にサーバプロセス及びシステムサーバの起動処理をしません。これを**高速系切り替え機能**といいます。系の切り替え時にサーバプロセス及びシステムサーバの起動処理がない分，系の切り替え時間を短縮できます。

なお，高速系切り替え機能の方がユーザサーバホットスタンバイより系の切り替え時間を短縮できます（高速系切り替え機能はユーザサーバホットスタンバイの機能を包括しています）。

(1)　IP アドレスについての条件

(a)　HiRDB/シングルサーバの場合

高速系切り替え機能をするユニットは IP アドレスを引き継げません。したがって，HiRDB/シングルサーバの場合は IP アドレスを引き継がない構成にしてください。

(b)　HiRDB/パラレルサーバの場合

高速系切り替え機能を使用するユニットは IP アドレスを引き継がないようにしてください。対象ユニットのユニット制御情報定義の **pd_ha_ipaddr_inherit** オペランドに N を指定します。サーバモードで運用しないユニットは IP アドレスを引き継ぐ構成にしてもかまいません。

バックエンドサーバのユニットを高速系切り替え機能の対象として，システムマネジャ及びフロントエンドサーバのユニットを対象外にすることをお勧めします。システムマネジャ及びフロントエンドサーバのユニットを IP アドレスを引き継がない構成にすると，系が切り替わった後の運用が IP アドレスを引き継ぐ場合に比べて難しくなります。システム構成例については，「高速系切り替え機能使用時のシステム構成例」を参照してください。

(2)　高速系切り替え機能を使用する場合に指定するオペランド

高速系切り替え機能を使用するには，**pd_ha_agent** オペランドに standbyunit を指定します。

また，HiRDB/パラレルサーバの場合はトランザクションキューイング機能を使用するかどうかを検討してください。トランザクションキューイング機能を使用すると，系切り替え時のトランザクションエラーを少なくできます。トランザクションキューイング機能については，「トランザクションキューイング機能」を参照してください。

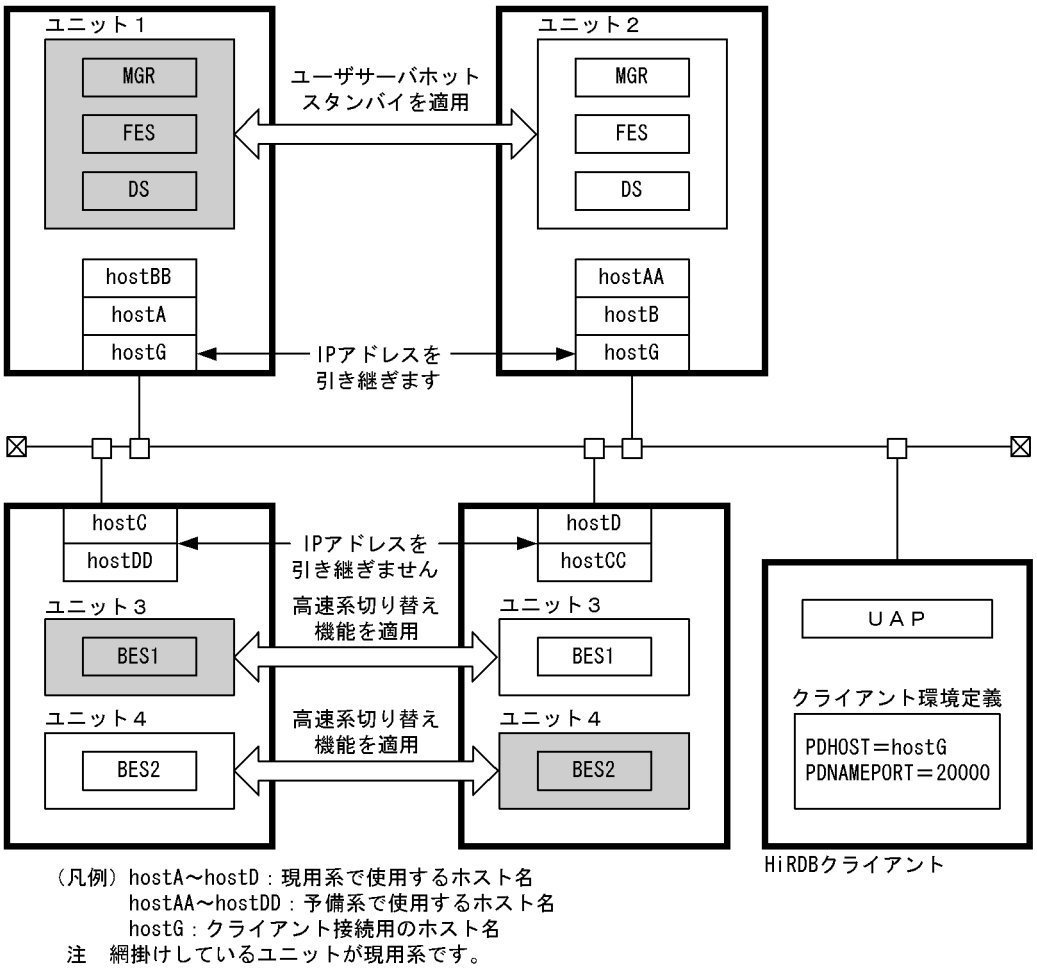
(3)　pdntenv コマンドによる設定（HiRDB/シングルサーバ限定）

pdntenv コマンドの-ro オプションで，リモート系コマンドを使用する設定をしてください。

26.17.3　高速系切り替え機能使用時のシステム構成例

高速系切り替え機能使用時のシステム構成例を次の図に示します。

図 26–75　高速系切り替え機能使用時のシステム構成例



〔説明〕

- システムマネジャ及びフロントエンドサーバのユニット（ユニット 1，ユニット 2）は，クライアント接続用の IP アドレスだけを引き継ぐ構成とし，ユーザサーバホットスタンバイを使用した 1：1 系切り替え構成にしています。
- バックエンドサーバのユニット（ユニット 3，ユニット 4）は，高速系切り替え機能を使用した相互系切り替え構成にしています。

このときの HiRDB システム定義の指定例を次に示します。定義例はシステム共通定義及びユニット制御情報定義の関連のあるオペランドについてだけ説明しています。

●システム共通定義

```
set pd_ha = use
set pd_name_port = 20000

pdunit -x hostA -u unt1 -d "C:¥hirdb1" -c hostAA -p 20000      1
pdunit -x hostB -u unt2 -d "C:¥hirdb2" -c hostBB -p 20001      2
pdunit -x hostC -u unt3 -d "C:¥hirdb3" -c hostCC -p 20002      3
pdunit -x hostD -u unt4 -d "C:¥hirdb4" -c hostDD -p 20003      4

pdstart -t MGR -u unt1
```

```
pdstart -t DIC -u unt1 -s DIC
pdstart -t FES -u unt1 -s FES
pdstart -t BES -u unt3 -s BES1
pdstart -t BES -u unt4 -s BES2
```

〔説明〕

1. ユニット 1 の定義です。IP アドレスを引き継がないため，-c オプションに予備系のホスト名を指定します。
2. ユニット 2 の定義です。IP アドレスを引き継がないため，-c オプションに予備系のホスト名を指定します。
3. ユニット 3 の定義です。IP アドレスを引き継がないため，-c オプションに予備系のホスト名を指定します。
4. ユニット 4 の定義です。IP アドレスを引き継がないため，-c オプションに予備系のホスト名を指定します。

●ユニット制御情報定義（ユニット 1）

```
set pd_hostname = hostA                          1
set pd_ha_acttype = server                       2
set pd_ha_ipaddr_inherit = N                     3
set pd_ha_server_process_standby = Y             4
```

〔説明〕

1. 現用系の標準ホスト名を指定します。
2. 系切り替え機能をサーバモードで実行します。サーバモードはユーザサーバホットスタンバイの前提条件です。
3. IP アドレスを引き継がない指定です。ユーザサーバホットスタンバイを使用するユニットでは IP アドレスを引き継げません。
4. このユニットに対してユーザサーバホットスタンバイを使用します。

●ユニット制御情報定義（ユニット 2）

```
set pd_hostname = hostB                          1
set pd_ha_acttype = server                       2
set pd_ha_ipaddr_inherit = N                     3
set pd_ha_server_process_standby = Y             4
```

〔説明〕

1. 現用系のホスト名を指定します。
2. 系切り替え機能をサーバモードで実行します。サーバモードはユーザサーバホットスタンバイの前提条件です。

3. IP アドレスを引き継がない指定です。ユーザサーバホットスタンバイを使用するユニットでは IP アドレスを引き継げません。
4. このユニットに対してユーザサーバホットスタンバイを使用します。

●ユニット制御情報定義（ユニット 3）

```
set pd_hostname = hostC                         1
set pd_ha_acttype = server                      2
set pd_ha_ipaddr_inherit = N                    3
set pd_ha_agent = standbyunit                   4
```

〔説明〕

1. 現用系のホスト名を指定します。
2. 系切り替え機能をサーバモードで実行します。サーバモードは高速系切り替え機能の前提条件です。
3. IP アドレスを引き継がない指定です。高速系切り替え機能を使用するユニットでは IP アドレスを引き継げません。
4. このユニットに対して高速系切り替え機能を使用します。

●ユニット制御情報定義（ユニット 4）

```
set pd_hostname = hostD                         1
set pd_ha_acttype = server                      2
set pd_ha_ipaddr_inherit = N                    3
set pd_ha_agent = standbyunit                   4
```

〔説明〕

1. 現用系のホスト名を指定します。
2. 系切り替え機能をサーバモードで実行します。サーバモードは高速系切り替え機能の前提条件です。
3. IP アドレスを引き継がない指定です。高速系切り替え機能を使用するユニットでは IP アドレスを引き継げません。
4. このユニットに対して高速系切り替え機能を使用します。

26.17.4　待機系の起動に時間が掛かる場合の確認作業

待機系の起動処理には実行系の起動を待ち合わせる処理があります。高速系切り替え機能を使用する待機系ユニットの起動に時間が掛かる場合は次に示すことを確認してください。

1. 実行系が起動しているか確認してください。起動していない場合は実行系を起動してください。
2. 待機系 HiRDB が内部的に発行する pdenvcp コマンドが無応答になっていないか，pdls -d prc -a コマンドで確認してください。無応答の場合は，実行系に_pd0envc コマンドプロセスが残っていないかを確認してください。残っている場合は，このプロセスを終わらせた後に待機系を再起動してください。

3. 実行系 HiRDB が内部的に発行する pdenvcp コマンドが，RPC，ファイル入出力処理で停止してないか pdls -d rpc コマンドで確認してください。ネットワークや OS の障害要因を取り除いた後に待機系を再起動してください。
4. 系の切り替え時に待機系の起動処理がタイムアウトする場合は，pd_system_complete_wait_time オペランド（pdstart コマンドの完了待ち時間）の値を見直してください。待機系の起動時間を考慮した値を設定した後に待機系を再起動してください。なお，タイムアウトした場合は，pdls -d prc -a コマンドで実行系に_pd0envc コマンドプロセスが残ってないかを確認してください。残っている場合は，このプロセスを終わらせた後に待機系を再起動してください。
5. リストを多数使用していると，リストの初期化処理によって系の切り替え時間が掛かることがあります。この場合，リストの初期化時期の変更を検討してください。pd_list_initialize_timing オペランドでリストの初期化時期を変更できます。リストの初期化時期の変更方法については，「リストの初期化（削除）時期の変更」を参照してください。

26.17.5　高速系切り替え機能使用時の注意事項

(1)　待機系 HiRDB（ユニット）を再度開始する必要がある操作

次の表に示す操作内容を実施したあと，実行系 HiRDB（ユニット）の再起動を実施したかどうかで，待機系 HiRDB（HiRDB/パラレルサーバの場合は待機系ユニット）を一度終了してから再度開始する必要があります。待機系 HiRDB を再度開始しないと，系切り替えが発生したときに待機系 HiRDB が異常終了します。

表 26–95　待機系 HiRDB（ユニット）を再度開始する必要がある操作

分類	条件			待機系 HiRDB（ユニット）を再度開始する必要の有無
	操作内容		実行系 HiRDB（ユニット）の再起動の有無	
SQL の実行	定義系 SQL の実行		なし	－
			あり	○
	HiRDB システム構築後最初の抽象データ型の定義（CREATE TYPE 文の実行）		なし	○
			あり	○
運用コマンド又はユティリティの実行	データベース構成変更ユティリティ（pdmod）の実行	監査人の登録	なし	－
			あり	－
		RD エリアの属性変更	なし	○
			あり	○
		監査証跡表の作成	なし	－
			あり	○
		上記以外の操作	なし	○※1

分類	条件		待機系 HiRDB（ユニット）を再度開始する必要の有無
	操作内容	実行系 HiRDB（ユニット）の再起動の有無	
		あり	○※1
	HiRDB システム構築後最初の pdplgrgst コマンドの実行（CREATE TYPE 文の実行）	なし	○
		あり	○
システム共通定義の変更	次のどれかの定義の変更 • グローバルバッファの定義（pdbuffer） • RD エリアの最大数（pd_max_rdarea_no） • RD エリアを構成する HiRDB ファイルの最大数（pd_max_file_no） • 表予約数の最低保証値（pd_assurance_table_no） • インデクス予約数の最低保証値（pd_assurance_index_no）	なし	−
		あり	○

（凡例）

○：待機系 HiRDB（ユニット）を再度開始する必要があります。

−：待機系 HiRDB（ユニット）を再度開始する必要はありません。

注※1

HiRDB/パラレルサーバの場合で，DS を配置しているユニットに高速系切り替え機能を適用している場合，その待機系ユニットの再起動も実施する必要があります。実施していない場合，そのユニットで系切り替えが発生すると異常終了します。

待機系 HiRDB が異常終了した場合は，待機系 HiRDB を pdstart コマンド（HiRDB/パラレルサーバの場合は pdstart -u 又は pdstart -q コマンド）で開始してください。

(2)　RD エリアのオープン契機について

高速系切り替え機能の対象になる待機系ユニットは，待機状態のときに RD エリアをオープンしていません。また，系の切り替え時間を最小限に抑えるため，系切り替えの発生時に全面回復で必要な RD エリアだけをオープンして，そのほかの RD エリアはオープンしません。したがって，RD エリアのオープン契機は次のようになります。

- pd_rdarea_open_attribute_use オペランドに N が指定されている場合，指定値に関係なく Y が指定されているものとみなします。
- 系切り替え発生時は，RD エリアオープン契機が INITIAL の場合，指定値に関係なく，DEFER が指定されているものとみなします。

RD エリアのオープン契機については，「RD エリアのオープン契機を変更する方法（RD エリアの属性変更）」を参照してください。

(3)　OLTP 製品と連携している場合

次に示す条件をすべて満たす場合は注意が必要です。

- 高速系切り替え機能を使用している（HiRDB/パラレルサーバの場合はシステムマネジャのユニットを高速系切り替え機能の対象にしている）
- X/Open に従った API を使用した OLTP 製品（OpenTP1 又は TPBroker for C++など）と連携している
- HiRDB クライアントのバージョンが 06-02-/A 以前である
- OLTP 製品のクライアント環境変数に HiRDB_PDHOST を指定している
- 現用系が待機系として待機完了状態になっている

この場合，OLTP 製品が未決着トランザクションの回復処理をすると，X/Open に従った API がエラーリターンしてトランザクションが回復されないことがあります。この現象が発生する場合は，HiRDB クライアントのバージョンを 06-02-/B 以降にバージョンアップしてください。業務を停止させたくないなどの理由で HiRDB クライアントのバージョンアップがすぐにできない場合は，現用系 HiRDB（ユニット）を待機系から実行系に系を切り替えてください。ただし，これは一時的な対応策です。HiRDB クライアントのバージョンアップで対応してください。

26.18　トランザクションキューイング機能

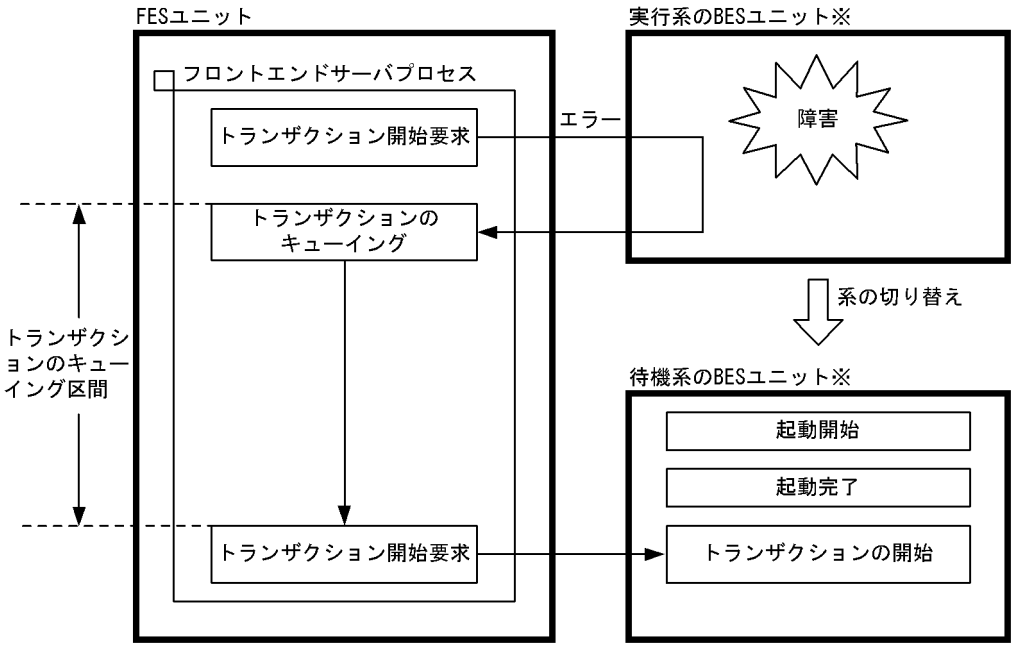
HiRDB/パラレルサーバで高速系切り替え機能を使用する場合，又はスタンバイレス型系切り替え機能を使用する場合はトランザクションキューイング機能が使用できます。

26.18.1　トランザクションキューイング機能とは

バックエンドサーバ又はディクショナリサーバのユニットで系切り替えが発生すると，系の切り替えが完了するまでこのバックエンドサーバ又はディクショナリサーバはトランザクションを受け付けられない状態になります。そのため，系切り替え中のバックエンドサーバ又はディクショナリサーバで処理するトランザクションがエラーになります。

これらのトランザクションをエラーにしないで，系切り替えが完了するまでフロントエンドサーバでトランザクションをキューイングします。これを**トランザクションキューイング機能**といいます。これによって，系切り替え時のトランザクションエラーを少なくできます。トランザクションキューイング機能の概要を次の図に示します。

図 26–76　トランザクションキューイング機能の概要



注※

スタンバイレス型系切り替え機能の場合は次に示すようになります。

- 正常時は正規 BES ユニットが実行系になり，代替部が待機系になります。
- 代替中は代替部が実行系になり，正規 BES ユニットが待機系になります。

〔説明〕

バックエンドサーバのユニットに障害が発生して系を切り替えます。待機系ユニットが起動完了するまで，トランザクションをキューイングします。待機系ユニットが起動完了した後にトランザクションを開始します。

備考

- 系を切り替えていない（障害が発生していない）ユニットで実行されるトランザクションはキューイングされません。通常どおり実行されます。
- マルチフロントエンドサーバを使用すると，フロントエンドサーバのユニットで系切り替えが発生した場合，トランザクションエラーを少なくできます。この場合，障害が発生したときに実行中のトランザクションだけがエラーになります。

26.18.2　環境設定方法

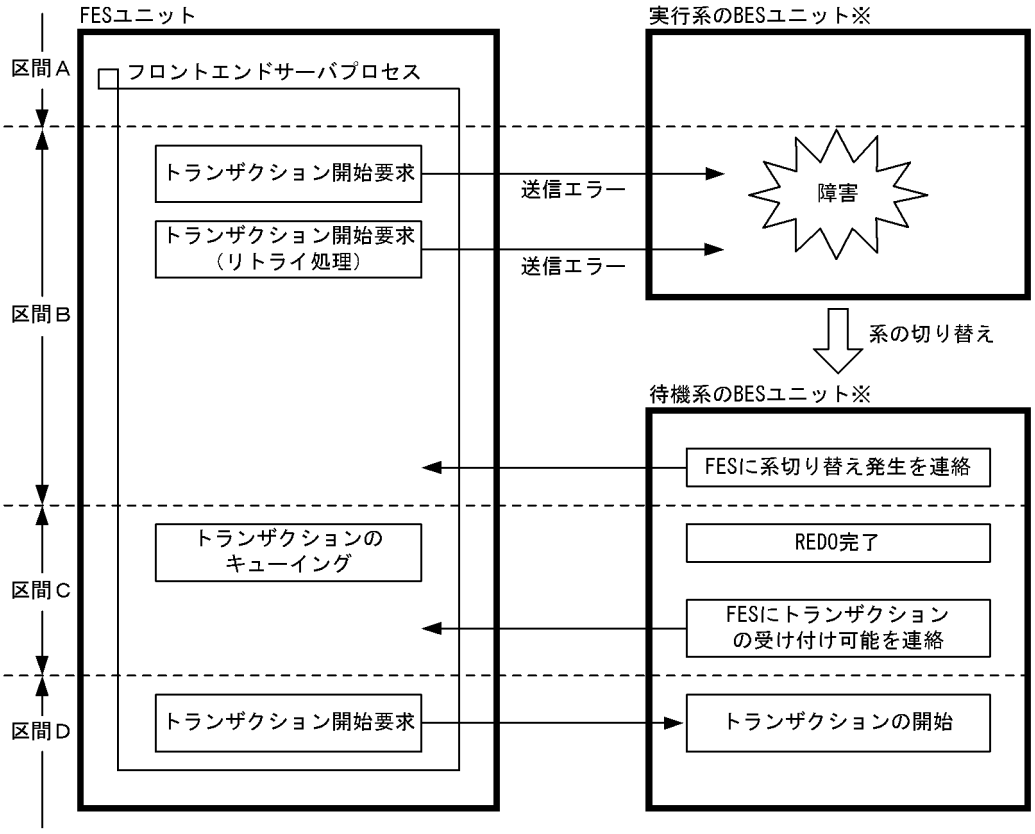
トランザクションキューイング機能を使用するには次の表に示すオペランドを指定します。

表 26–96　トランザクションキューイング機能を使用するときに指定するオペランド

オペランド名	説明
pd_ha_agent	高速系切り替え機能，1：1 スタンバイレス型系切り替え機能，又は影響分散スタンバイレス型系切り替え機能を使用する場合に指定します。 **standbyunit：**高速系切り替え機能を使用します。 **server：**1：1 スタンバイレス型系切り替え機能を使用します。 **activeunits**：影響分散スタンバイレス型系切り替え機能を使用します。
pd_ha_transaction	トランザクションキューイング機能を使用するかどうかを指定します。 クライアント環境定義の PDHATRNQUEUING オペランドに NO を指定すると，その HiRDB クライアントが実行する UAP はトランザクションキューイング機能の対象外になります。PDHATRNQUEUING オペランドについては，マニュアル「HiRDB UAP 開発ガイド」を参照してください。
pd_ha_trn_queuing_wait_time	トランザクションのキューイング待ち時間を指定します。このオペランドに指定した待ち時間を超えても待機系ユニットが起動しない場合，キューイング中のトランザクションをエラーにします。これ以降発生するトランザクションはキューイングされないでエラーになります。
pd_ha_trn_restart_retry_time	トランザクションキューイング機能使用時に系切り替えが発生すると，トランザクションをフロントエンドサーバでキューイングします。しかし，系切り替えが発生してから待機系ユニットが再開始するまでの間，フロントエンドサーバは系切り替えを検知できません。この間（系切り替えが発生してから待機系ユニットが再開始するまでの間），フロントエンドサーバは実行系ユニットに対してトランザクションの開始要求をしますが，実行系ユニットは既に異常終了しているため，このトランザクション開始要求はエラーになります。エラーになったトランザクションはフロントエンドサーバから開始要求が再度行われます（トランザクションの開始要求をリトライします）。 このオペランドには，このリトライ時間の上限を指定します。このオペランドの値を超えても待機系ユニットが再開始されない場合は，リトライ中のトランザクションをエラー

オペランド名	説明
	にします。また，これ以降発生するトランザクションはリトライされないでエラーになります。 pd_ha_trn_queuing_wait_time オペランドと pd_ha_trn_restart_retry_time オペランドの関係を図「pd_ha_trn_queuing_wait_time オペランドと pd_ha_trn_restart_retry_time オペランドの関係」に示します。
pd_ha_trn_queuing_initial	トランザクションキューイング機能を使用する場合に，フロントエンドサーバを配置したユニットの起動直後からトランザクションキューイング機能を有効にするかどうかを指定します。

図 26–77　pd_ha_trn_queuing_wait_time オペランドと pd_ha_trn_restart_retry_time オペランドの関係



注※

スタンバイレス型系切り替え機能の場合は次に示すようになります。

- 正常時は正規 BES ユニットが実行系になり，代替部が待機系になります。
- 代替中は代替部が実行系になり，正規 BES ユニットが待機系になります。

〔説明〕

区間 A，D：

トランザクションを開始できる状態です（正常な状態です）。

区間 B：

バックエンドサーバのユニットが系切り替え中で，フロントエンドサーバがそれを検知できていない状態です。pd_ha_trn_restart_retry_time オペランドに指定した時間までトランザクション開始要求をリトライします。フロントエンドサーバが系の切り替えを検知すると，トランザクションがキューイングされます。指定した時間を過ぎても検知できない場合は，そのトランザクションをエラーにします。

区間 C：

バックエンドサーバのユニットが系切り替え中で，フロントエンドサーバがそれを検知している状態です。pd_ha_trn_queuing_wait_time オペランドに指定した時間までトランザクションをキューイングします。指定した時間を過ぎてもトランザクションを開始できない場合は，そのトランザクションをエラーにします。

26.18.3　キューイングの対象になるトランザクション

SQL の延長で生成されるトランザクションがキューイングの対象になります。ただし，定義系 SQL の延長で生成されるトランザクション，及びホールダブルカーソル機能を使用したトランザクションは対象外です。次に示すトランザクションはキューイングの対象になりません。

- 定義系 SQL を発行するトランザクション
- ホールダブルカーソルを開いている接続のトランザクション
- ASSIGN LIST 文を発行するトランザクション
- 運用コマンド又はユティリティが生成するトランザクション

ただし，タイミングによってはこれらのトランザクションがキューイングされることがあります。

26.18.4　注意事項

(1)　時間監視関連のオペランドについて

トランザクションのキューイング時間は最大で，pd_ha_trn_queuing_wait_time オペランドの値（省略値 180 秒）と pd_ha_trn_restart_retry_time オペランドの値（省略値 60 秒）の合計時間になります。このため，次に示すオペランドの値に注意してください。

- **クライアント環境定義の PDCWAITTIME オペランド**
  系切り替え中も PDCWAITTIME が計測されているため，系の切り替えに掛かる時間を加算してオペランドの値を決めてください。
- **pd_lck_wait_timeout オペランド**
  キューイングされているトランザクションが障害の発生していないユニットで排他を確保することがあるため，系の切り替えに掛かる時間を加算してオペランドの値を決めてください。

系の切り替えに掛かる時間は，クラスタソフトウェアがイベントログに出力する系切り替え開始メッセージと系切り替え完了メッセージの出力時刻の差分によって求められます。系の切り替えが開始したときは次に示すどれかのメッセージが出力されます。

- KAMN200-I メッセージ（クラスタソフトウェアの系を切り替えるコマンド入力時に出力）
- KAMN300-E メッセージ（実行系 HiRDB の異常終了時に出力）
- KAMN301-E メッセージ（実行系 HiRDB のスローダウン時に出力）

系の切り替えが完了したときは KAMN311-I メッセージが出力されます。

(2) リスト使用時の注意事項

リストを使用した検索中に系切り替えが発生した場合は注意が必要です。バックエンドサーバ又はディクショナリサーバのユニットで系切り替えが発生すると，系切り替え前のユニットで作成したリストが削除されます。そのため，系切り替え後にリストを使用した検索（キューイングされていたトランザクション）がエラーになります。この場合，リストを削除するか，又はリストを再作成してください。

(3) 最大同時接続数（pd_max_users オペランドの値）について

トランザクションのキューイング中は処理待ちのユーザ数が通常より多くなるため，最大同時接続数（pd_max_users オペランドの値）を超える可能性があります。最大同時接続数を超えた場合，それ以上のユーザはそのフロントエンドサーバに接続できませんが，フロントエンドサーバへの接続処理をリトライします。pd_ha_trn_queuing_wait_time と pd_ha_trn_restart_retry_time オペランドの合計時間だけリトライします。

(4) 系切り替え中に HiRDB に接続できない UAP について

次に示す場合，系切り替え中に HiRDB に接続できない UAP が発生します。

- システムマネジャのユニットが系切り替え中の場合はフロントエンドサーバの割り当てができないため，接続するフロントエンドサーバを指定していない（PDFESHOST オペランドの指定がない）UAP は HiRDB に接続（CONNECT）できません。
- PDFESHOST オペランドに指定しているフロントエンドサーバのユニットが系切り替え中の場合，その UAP は HiRDB に接続（CONNECT）できません。

この場合，UAP は自動再接続機能で HiRDB への接続処理をリトライします。リトライ中に系の切り替えが完了すると，HiRDB に接続できるようになります。自動再接続機能については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(5) バックエンドサーバ接続保持機能を使用している場合

バックエンドサーバ接続保持機能を使用している場合の注意事項については，「クライアントの最大待ち時間の設定」を参照してください。

(6)　サーバが停止した状態での SQL 実行について

ユニットやサーバの終了によって停止したサーバに対して SQL を実行しても，トランザクションキューイングが行われて，エラーになるまでの時間が長くなります。

また，次のどちらかのメッセージを出力することがあります。

- KFPS00993-I（REQUEST=PDCWAITTIME_over）
- KFPS00612-E（reason=pd_ha_trn_restart_retry_time）

(7)　複数ユニットが同時に系切り替えした場合

トランザクションキューイング機能は，フロントエンドサーバを配置したユニットの起動後に，バックエンドサーバやディクショナリサーバを配置したユニットが稼働中又は起動し始めてから機能が有効になります。

フロントエンドサーバを含む複数ユニットの同時系切り替え時にも，バックエンドサーバやディクショナリサーバを配置したユニットの起動を待たないでトランザクションキューイング機能を有効にする場合は，pd_ha_trn_queuing_initial オペランドの適用を検討してください。pd_ha_trn_queuing_initial オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

26.19　サーバ障害以外の障害が発生した場合に系を切り替える方法

ここでは，次に示す障害が発生した場合に系を切り替える方法について説明します。

- サーバプロセスの異常終了が多発した場合
- RD エリアの入出力エラー（パス障害）が発生した場合

26.19.1　サーバプロセスの異常終了が多発した場合に系を切り替える方法

サーバプロセスの異常終了が多発すると，新たなサービスを受け付けられないことがあります。しかし，サーバプロセスの異常終了では，HiRDB は異常終了しませんが，実質オンライン停止状態になります。また，HiRDB が異常終了しないため，系を切り替えられません。このオンライン停止状態になったときに系を切り替える方法を説明します。

(1)　系を切り替えるための準備作業

(a)　pd_down_watch_proc オペランドを指定します

サーバプロセスの異常終了回数が一定時間内に **pd_down_watch_proc** オペランドの値を超えた場合，HiRDB（HiRDB/パラレルサーバの場合は該当するユニット）を異常終了します。これを**プロセスの異常終了回数監視機能**といいます。この機能を使用して，オンライン停止状態になったときに HiRDB を異常終了させて系を切り替えます。プロセスの異常終了回数監視機能については，「サーバプロセスの異常終了回数の監視（プロセスの異常終了回数監視機能）」を参照してください。

(b)　pd_mode_conf オペランドの指定値を確認します

プロセスの異常終了回数監視機能で HiRDB が異常終了した場合に系を切り替えるときは，pd_mode_conf = MANUAL2 を指定します。HiRDB が異常終了した場合に系を切り替えないとき（異常終了した系で HiRDB を再開始するとき）は，pd_mode_conf = MANUAL1 を指定します。

(c)　Hitachi HA Toolkit Extension の switchtype オペランドを指定します（サーバモード限定）

Hitachi HA Toolkit Extension の switchtype オペランドに switch を指定します。switch を指定すると，HiRDB が異常終了した場合に系を切り替えます。

(d)　系の切り替え時期を監視します（モニタモード限定）

サーバプロセスの異常終了が多発して HiRDB が異常終了しても，自動的に系を切り替えられません。ユーザの運用（系切り替え用のシェルの実行など）で系を切り替えます。ただし，クラスタソフトウェアに CLUSTERPRO を使用して，カスタム監視リソースでフェイルオーバを行う場合は系を切り替えます。系切り替えの運用例を次に示します。

- HiRDB が異常終了したとき，KFPS01821-E メッセージがイベントログに出力されます。JP1 又は監視シェルなどを使用してこのメッセージを監視し，メッセージが出力されたときに系を切り替えます。HiRDB/パラレルサーバの場合，システムマネジャがあるサーバマシンのイベントログか，又は異常終了したユニットがあるサーバマシンのイベントログのどちらかにこのメッセージが出力されます。
- UAP を定期的に実行して，データベースにアクセスできるかどうかを確認してください。UAP がデータベースにアクセスできない場合に系を切り替えます。

(2) 相互系切り替え構成の場合

系を切り替えると同一サーバマシン上で複数の HiRDB が稼働するため，逆にトラフィックが上昇して効果が得られないことがあります。相互系切り替え構成でプロセスの異常終了回数監視機能を使用する場合は，HiRDB の異常終了時に系を切り替えないことをお勧めします。異常終了した系で HiRDB を再開始するため，pd_mode_conf = MANUAL1 を指定してください。

なお，サーバモードで運用する場合は，Hitachi HA Toolkit Extension の switchtype オペランドに restart 又は manual を指定してください。restart を指定すると，障害が発生した系で HiRDB を再開始します。障害が発生した系で HiRDB を再開始できないときに，系を切り替えて切り替え先の系で HiRDB を再開始します。manual を指定した場合は，HiRDB を再開始できなくても自動的に系を切り替えません。

(3) 系の切り替え時間を短縮する方法

サーバプロセスの異常終了が多発すると，トラブルシュート情報が大量に出力されて系の切り替えに時間が掛かることがあります。次に示すオペランドを指定すると，トラブルシュート情報の出力を抑止するため，サーバプロセスの異常終了が多発したときに系の切り替え時間を短縮できます。

- pd_dump_suppress_watch_time
- pd_cancel_dump = noput
- pd_client_waittime_over_abort = N

また，pd_ha_switch_timeout オペランドに Y を指定すると，系切り替え時の実行系 HiRDB（スタンバイレス型系切り替え機能の場合は正規 BES ユニット）の内部停止処理がサーバ障害監視時間を超えた場合に，実行系 HiRDB の内部停止処理を待たないで系を切り替えられます。ただし，このオペランドはサーバモードのときだけ指定できます。

系の切り替え時間に影響を与える障害を次の表に示します。

表 26–97 系の切り替え時間に影響を与える障害

障害の種類（系の切り替え要因）		影響の有無	
		モニタモード	サーバモード
HiRDB の異常終了	pdprcd の異常終了	○	○
	システムサーバの異常終了	×	○

<table>
<tr><th colspan="4" rowspan="2">障害の種類（系の切り替え要因）</th><th colspan="2">影響の有無</th></tr>
<tr><th>モニタモード</th><th>サーバモード</th></tr>
<tr><td rowspan="8"></td><td rowspan="8">ユーザサーバの異常終了</td><td colspan="2">クリティカル</td><td>△</td><td>○</td></tr>
<tr><td rowspan="7">非クリティカル※1</td><td rowspan="2">PDCWAITTIME オーバ</td><td>pd_client_waittime_over_abort=Y（省略値）</td><td>△</td><td>○</td></tr>
<tr><td>pd_client_waittime_over_abort=N</td><td>○</td><td>○</td></tr>
<tr><td colspan="2">内部強制終了※2</td><td>△</td><td>○</td></tr>
<tr><td colspan="2">アボート</td><td>△</td><td>○</td></tr>
<tr><td colspan="2">XA 接続の UAP でロールバックが発生</td><td>△</td><td>○</td></tr>
<tr><td colspan="2">上記以外</td><td>△</td><td>○</td></tr>
<tr><td colspan="3">HiRDB のスローダウン</td><td>pdprcd の無応答</td><td>×</td><td>○</td></tr>
<tr><td colspan="4">系障害</td><td>×</td><td>○</td></tr>
<tr><td colspan="4">計画系切り替え</td><td>○</td><td>○</td></tr>
</table>

（凡例）

○：系の切り替え時間に影響を与えません。

ただし，障害発生のタイミングによっては，系の切り替え時間に影響を与えます。

△：系の切り替え時間に影響を与えることがあります。

次に示すオペランドを指定すると，系の切り替え時間に与える影響を最小限にできます。

- pd_cancel_dump=noput
- pd_dump_suppress_watch_time

×：系の切り替え時間に影響を与えることがあります。

注※1

この障害では通常 HiRDB は異常終了しません。ただし，pd_down_watch_proc オペランドを指定した場合はサーバプロセスの異常終了回数を監視して，ある一定頻度を超えると HiRDB を異常終了します。

注※2

HiRDB が内部的に SIGKILL を発行してサーバプロセスを終了させることです。PDCWAITTIME オーバ及び pdcancel コマンドによる強制終了は含まれません。

26.19.2　RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える方法

ここでは，RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える方法について説明します。なお，ここでいう入出力エラーとは，HiRDB がファイルを操作したときに HiRDB が判別できない理由でファイル操作に失敗したエラーを意味し，HiRDB ファイルシステムに対するアクセス要求から返されるエラーコードに-1544 が出力されるエラーのことです。

(1)　系を切り替えるための準備作業

(a)　pd_db_io_error_action オペランドを指定します

pd_db_io_error_action オペランドに unitdown を指定すると，RD エリアの入出力エラーが発生したときに HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了するため，系を切り替えられます。入出力エラーの要因が**パス障害**の場合は，系の切り替え後に入出力処理ができるようになるため，業務をそのまま続行できます。なお，ここでいうパス障害とは，HiRDB とファイル間の通信経路が何かの要因で切断されているため，ファイルをアクセスできない状態を意味しています。

pd_db_io_error_action オペランドに unitdown を指定する場合は，「RD エリアの入出力エラー（パス障害）が発生したときの対処方法」を参照してください。

(b)　pd_mode_conf オペランドの指定値を確認します

HiRDB が異常終了した場合に系を切り替えるため，pd_mode_conf = MANUAL2 を指定します。HiRDB が異常終了した場合に系を切り替えないとき（異常終了した系で HiRDB を再開始するとき）は，pd_mode_conf = MANUAL1 を指定します。

(c)　Hitachi HA Toolkit Extension の switchtype オペランドを指定します（サーバモード限定）

Hitachi HA Toolkit Extension の switchtype オペランドに switch を指定します。switch を指定すると，HiRDB が異常終了した場合に系を切り替えます。

(d)　系の切り替え時期を監視します（モニタモード限定）

モニタモードの場合は HiRDB が異常終了しても，自動的に系を切り替えられません。ユーザの運用（系切り替え用のシェルの実行など）で系を切り替えます。系切り替えの運用例を次に示します。

- HiRDB が異常終了したとき，KFPS01821-E メッセージがイベントログに出力されます。JP1 又は監視シェルなどを使用してこのメッセージを監視し，メッセージが出力されたときに系を切り替えます。HiRDB/パラレルサーバの場合，システムマネジャがあるサーバマシンのイベントログか，又は異常終了したユニットがあるサーバマシンのイベントログのどちらかにこのメッセージが出力されます。
- UAP を定期的に実行して，データベースにアクセスできるかどうかを確認してください。UAP がデータベースにアクセスできない場合に系を切り替えます。

(2) 運用方法

入出力エラーが発生して HiRDB が異常終了した場合は系を切り替えて処理を続行してください。そして，出力されたメッセージを調査して対策してください。対策後，必要に応じて再度系を切り替えるか，又は HiRDB を一度終了して再度開始してください。なお，系の切り替え後も入出力エラーが発生すると RD エリアは障害閉塞します。この場合，データベース回復ユティリティ（pdrstr コマンド）で RD エリアを回復してください。

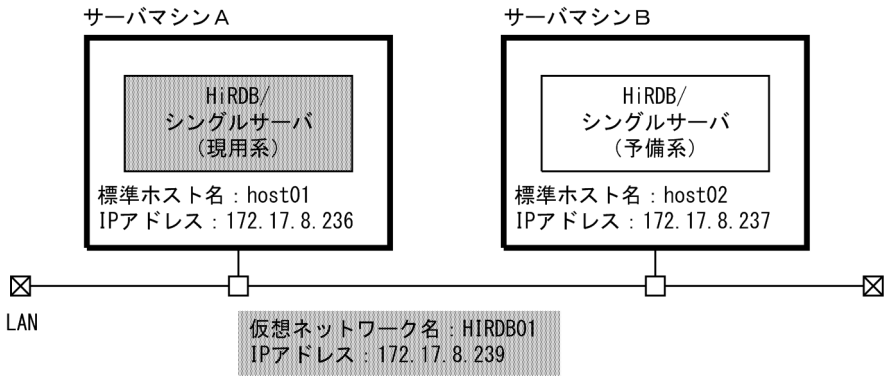
26.20　システム構築例

26.20.1　システム構築例の説明

ここでは，クラスタソフトウェアに MSFC を用いて，HiRDB/シングルサーバの 1：1 系切り替え構成の構築例を説明します。また，IP アドレスを引き継ぐ構成とします。

なお，ここでの説明は HiRDB の環境設定を理解していることを前提としています。HiRDB の環境設定については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

●システム構成例



(1)　MSFC がインストールされているかを確認します

システムを構築する前に MSFC がインストールされているかを確認してください。MSFC がインストールされていない状態でシステムを構築しても，系切り替え機能を使用できません。この場合，システムを再構築する必要があります。

(2)　hosts ファイルにホスト名を指定します

DNS サーバを使用しない場合は，hosts ファイルにホスト名と IP アドレスを指定してください。hosts ファイルを保存した後，Windows を再起動してください。

（例）

```
host01 172.17.8.236                1
host02 172.17.8.237                2
HIRDB01 172.17.8.239               3
```

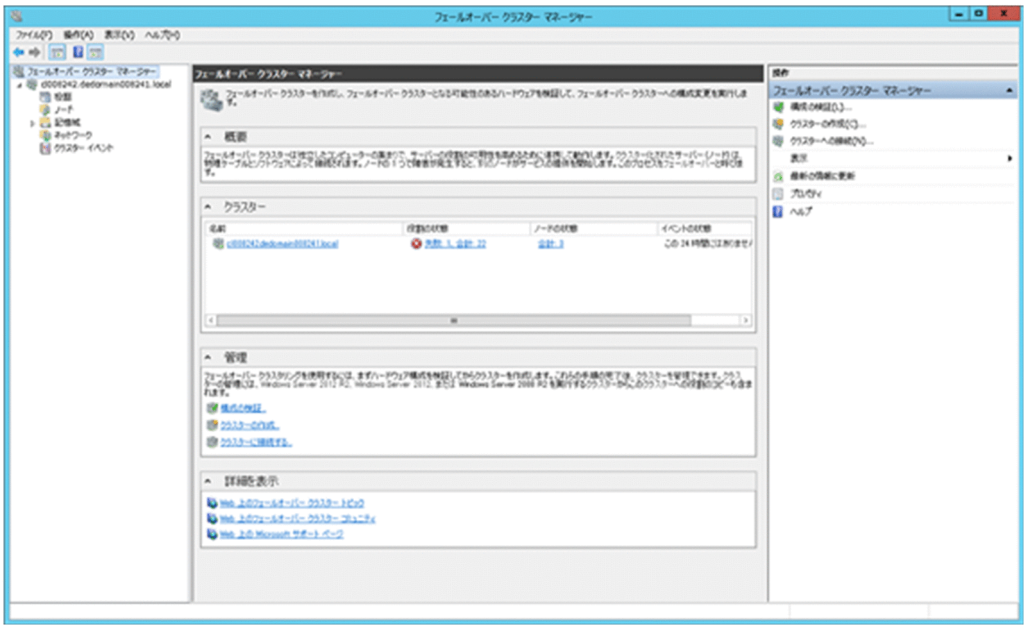
〔説明〕

1. サーバマシン A（現用系）の標準ホスト名及び IP アドレスを指定します。
2. サーバマシン B（予備系）の標準ホスト名及び IP アドレスを指定します。
3. HiRDB が系切り替え機能で使用する仮想ネットワーク名及び IP アドレスを指定します。

(3)　HiRDB の役割を作成します

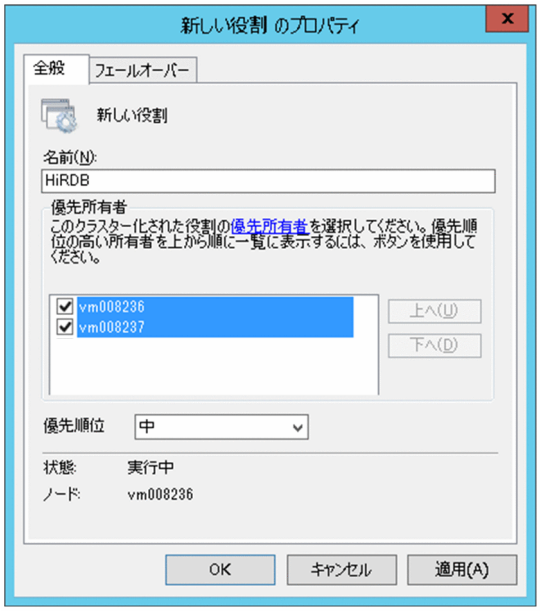
サーバマシン A（現用系）でフェールオーバークラスターマネージャーを起動して，次に示す手順で HiRDB の役割を作成してください。

1. **[スタート]** − **[フェールオーバークラスターマネージャー]** でフェールオーバークラスターマネージャーを起動します。



2. フェールオーバークラスターマネージャーのツリーの **[役割]** を右クリックして，**[空の役割の作成]** を選択し，**[新しい役割]** を作成します。
3. 作成した役割を選択して右クリックします。**[プロパティ]** を選択して **[プロパティ]** 画面を表示し，名前欄に HiRDB の役割名を指定します。この例では次のように指定します。

   名前：HiRDB
4. **[優先所有者]** 欄に表示されている優先所有者を選択します。優先順（現用系，予備系の順）に順番を変更して，現用系のノード（vm008236），予備系のノード（vm008237）のチェックボックスをチェックします。

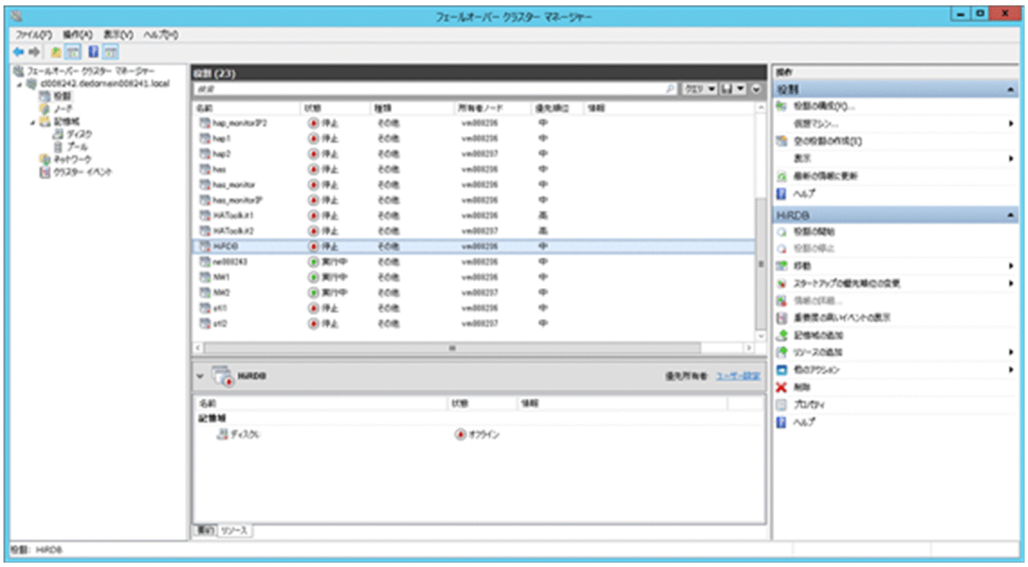


5. **[OK]** をクリックします。

(4)　共有ディスクを HiRDB の役割に登録します

フェールオーバークラスターマネージャーを起動して共有ディスク（物理ディスク）を HiRDB の役割に登録します。この例では L ドライブを共有ディスクとしています。

[記憶域] − **[ディスク]** 内にあるリソース［Disk L:］を HiRDB の役割に割り当ててください。



(5)　クライアントアクセスポイントを MSFC に登録します

HiRDB が系切り替え機能で使用する仮想ネットワーク名及び IP アドレスを次に示す手順で登録します。

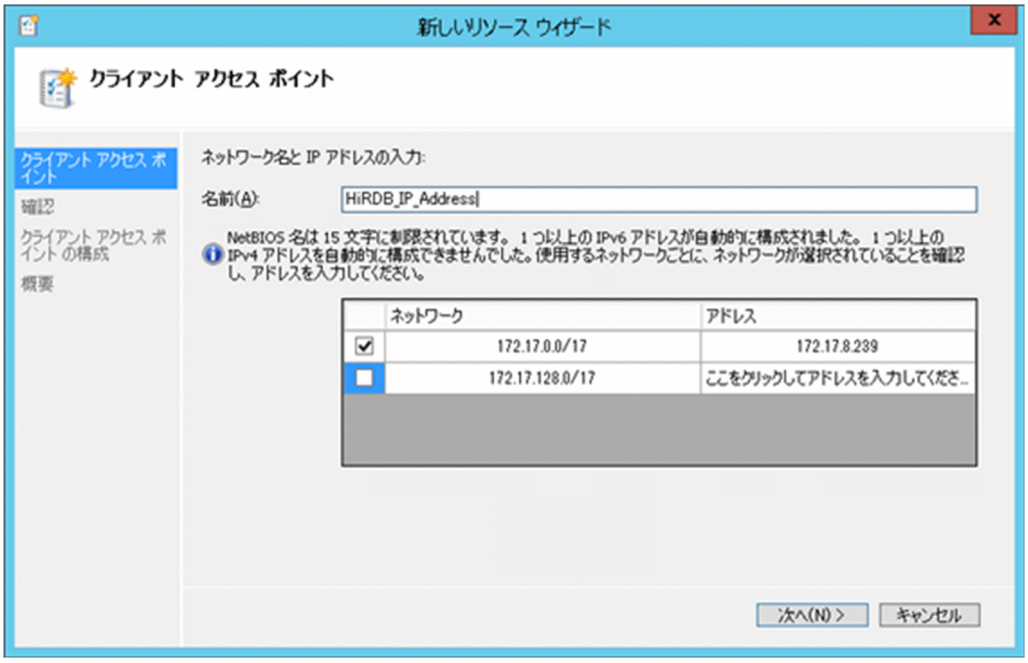
1. フェールオーバークラスターマネージャーを起動します。

2. (3)で作成した MSFC の役割を選択して右クリックします。**［リソースの追加］－［クライアントアクセスポイント］**を選択して，**［新しいリソース ウィザード］**画面を表示します。この例では次のように指定します。

   名前：HiRDB_IP_Address

   ネットワーク：172.17.0.0/17 のチェックボックスだけチェック

   アドレス：172.17.8.239



3. **［次へ］**をクリックして［確認］画面を表示します。この画面では何も設定する内容はありません。
4. **［次へ］**をクリックして，**［完了］**をクリックします。

(6)　リソースを現用系でオンラインにします

次に示すリソースを現用系でオンラインにします。

- 物理ディスク（共有ディスク）
- IP アドレス
- ネットワーク名

次に示す手順でこれらのリソースをオンラインにしてください。

〈手順〉

1. フェールオーバークラスターマネージャーを起動します。
2. 該当するリソースを選択して右クリックします。
3. **［オンラインにする］**を選択します。

(7)　サーバマシン A（現用系）に HiRDB をインストールします

サーバマシン A に HiRDB をインストールしてください。HiRDB はローカルディスクにインストールしてください。1：1 系切り替え構成のため，標準セットアップ（識別子なし）を選択してください。

(8)　HiRDB システム定義を作成します

現用系で HiRDB システム定義を作成してください。系切り替え機能に関連するオペランドの指定値を次に示します。

- pd_mode_conf = MANUAL1
- pd_ha = use
- pd_ha_ipaddr_inherit = Y
- pdunit -x HIRDB01 -u unt1
- pdstart -t SDS -s sds01 -u unt1
- pd_hostname = host01

〔説明〕

- pdunit オペランドの-x オプションには，HiRDB が系切り替え機能で使用する仮想ネットワーク名を指定します。
- pd_hostname オペランドには現用系の標準ホスト名を指定します。

(9)　HiRDB ファイルシステム領域を作成します

pdfmkfs コマンドで，次に示す目的の HiRDB ファイルシステム領域を共有ディスク（この例では L ドライブ）に作成してください。

- システムファイル用の HiRDB ファイルシステム領域
- RD エリア用の HiRDB ファイルシステム領域
- 作業表用ファイル用の HiRDB ファイルシステム領域
- ユティリティ用（アンロードログファイル作成用）の HiRDB ファイルシステム領域

(10)　システムファイルを作成します

システムファイル用の HiRDB ファイルシステム領域にシステムファイルを作成してください。

(11)　現用系の HiRDB が開始するかを確認します

次に示す手順で現用系の HiRDB を開始してください。

1. **pdstart** コマンドで現用系の HiRDB を開始してください。

2. KFPS05201-Q メッセージが出力されたら，データベース初期設定ユティリティ（pdinit コマンド）を実行してシステム用 RD エリアを作成します。
3. KFPS05210-I メッセージが出力されたら，HiRDB の開始完了です。
4. **pdstop** コマンドで現用系の HiRDB を終了してください。

(12)　共有ディスクの系を切り替えます

共有ディスクを現用系から予備系に切り替えます。次に示す手順で共有ディスクを切り替えてください。

1. フェールオーバークラスターマネージャーを起動します。
2. HiRDB の役割を選択して右クリックします。
3. **［移動］－［ノードの選択］**を選択します。
4. 移動させるクラスターノードで予備系のノードを選択して，**［OK］**をクリックします。
5. 所有者ノードに表示されているノード名が予備系のノードであることを確認します。

(13)　サーバマシン B（予備系）に HiRDB をインストールします

サーバマシン B に HiRDB をインストールしてください。HiRDB はローカルディスクにインストールしてください。1：1 系切り替え構成のため，標準セットアップ（識別子なし）を選択してください。

(14)　HiRDB システム定義を作成します

現用系で作成した HiRDB システム定義を予備系にコピーしてください。

(15)　予備系の HiRDB が開始するかを確認します

既に，HiRDB ファイルシステム領域，システムファイル，及びシステム用 RD エリアが共有ディスクに作成されているため，HiRDB を開始できます。次に示す手順で予備系の HiRDB を開始してください。

1. **pdstart** コマンドで予備系の HiRDB を開始してください。
2. **pdls** コマンドで予備系の HiRDB が開始したか確認してください。
3. **pdstop** コマンドで予備系の HiRDB を終了してください。

(16)　共有ディスクの系を切り替えます

共有ディスクを予備系から現用系に切り替えます。次に示す手順で共有ディスクを切り替えてください。

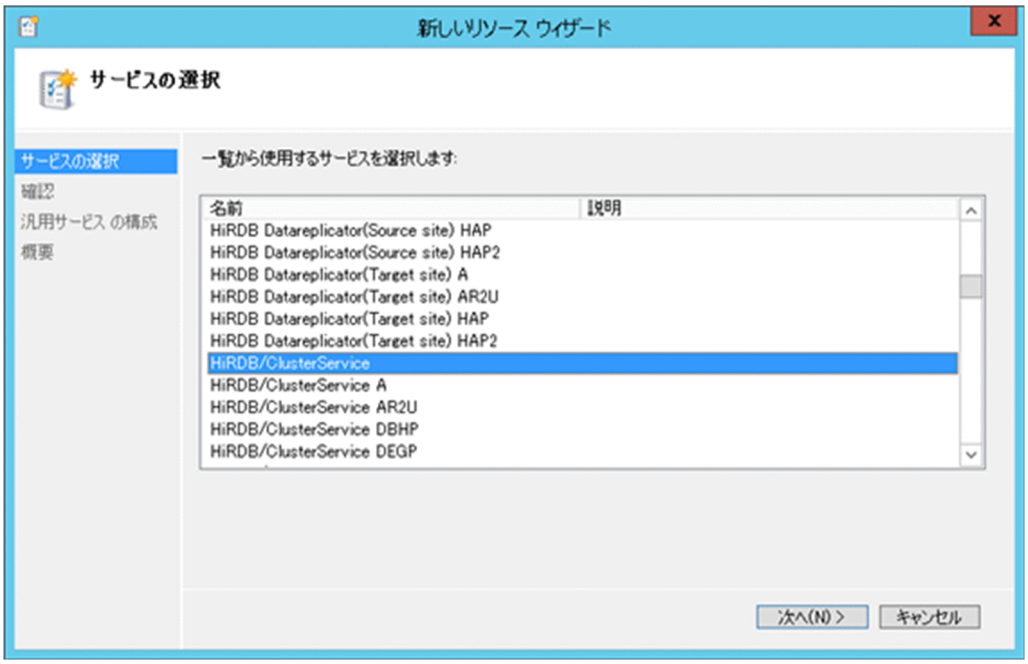
1. フェールオーバークラスターマネージャーを起動します。
2. HiRDB の役割を選択して右クリックします。
3. **［移動］－［ノードの選択］**を選択します。
4. 移動させるクラスターノードで現用系のノードを選択して，**［OK］**をクリックします。

5. 所有者ノードに表示されているノード名が予備系のノードであることを確認します。

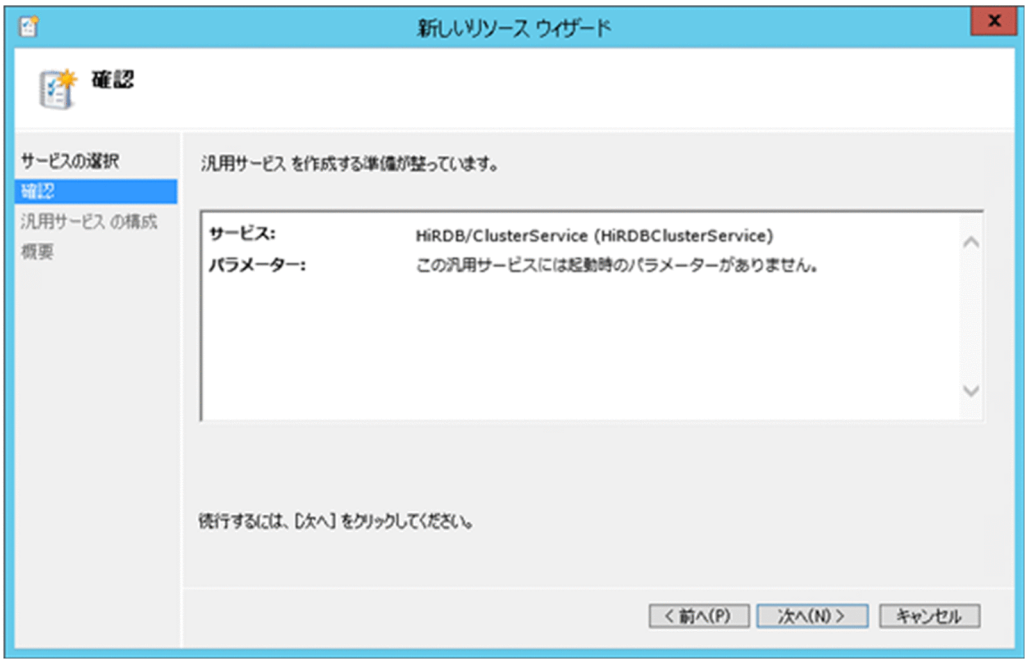
(17)　汎用サービスに HiRDB のクラスタサービスを登録します

汎用サービスに HiRDB のクラスタサービスを次に示す手順で登録してください。

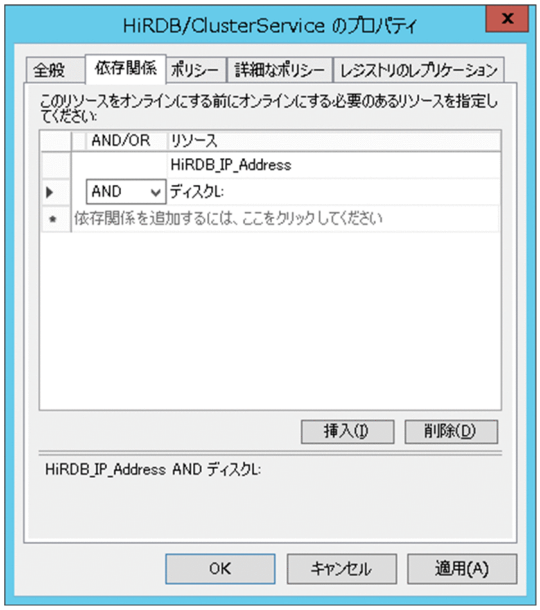
1. フェールオーバークラスターマネージャーを起動します。
2. (3)で作成した MSFC の役割を選択して右クリックします。**［リソースの追加］－［汎用サービス］**を選択して，［新しいリソース ウィザード］画面を表示します。
3. サービスの一覧から，HiRDB/ClusterService を選択します。



4. **［次へ］**をクリックして，［確認］画面を表示します。この画面では何も設定する内容はありません。



5. **[次へ]** をクリックして，**[完了]** をクリックします。
6. 作成した HiRDB/ClusterService リソースを選択して右クリックします。**[プロパティ]** ダイアログを表示して，**[依存関係]** 画面を選択します。
7. **[リソース]** 欄から物理ディスクとネットワーク名を選択して，**[OK]** をクリックします。



(18)　HiRDB のクラスタサービスを現用系でオンラインにします

HiRDB のクラスタサービスを現用系でオンラインにします。次に示す手順でオンラインにしてください。

1. フェールオーバークラスターマネージャーを起動します。
2. 役割「HiRDB」を選択して右クリックします。
3. **[役割の開始]** を選択します。

この時点で，現用系の HiRDB が開始します。

(19)　汎用サービスに HiRDB のサービスを登録します

汎用サービスに HiRDB の HiRDBSingleServer サービス（HiRDB/パラレルサーバの場合は HiRDBParallelServer）を登録してください。HiRDB のサービスを MSFC に登録すると，HiRDB の異常終了時にも系切り替え機能を適用できます。

登録手順は(17）と同じです。

ポイント

- ［依存関係］画面の［リソースの依存関係］欄に物理ディスクとネットワーク名を指定してください。

- ここで登録した汎用サービス（HiRDBSingleServer 又は HiRDBParallelServer）を HiRDB のクラスタサービス（HiRDBClusterService）の汎用サービスの依存関係に追加してください。

フェールオーバークラスターマネージャーを起動して，ここで登録した HiRDB のサービスのプロパティ画面を開き，次の設定をしてください。

- HiRDB/シングルサーバの場合は［HiRDBSingleServer のプロパティ］画面，HiRDB/パラレルサーバの場合は［HiRDBParallelServer のプロパティ］画面の［ポリシー］タブで，指定期間内での再起動の試行回数に 0 を指定してください。0 を指定しないと HiRDB が異常終了したときに系が切り替わりません。

  参考

  HiRDB の異常終了が 30 分間に 3 回発生すると，系が切り替わります。

(20) HiRDB クライアントの hosts ファイルに仮想ネットワーク名と IP アドレスを指定します

DNS サーバを使用しない場合は，HiRDB クライアントの hosts ファイルに仮想ネットワーク名と IP アドレスを指定してください。hosts ファイルを保存した後，OS を再起動してください。

（例）

```
HIRDB01 172.17.8.239
```

(21) クライアント環境定義の PDHOST オペランドを指定します

クライアント環境定義の PDHOST オペランドに仮想ネットワーク名を指定してください。

（例）

```
PDHOST = HIRDB01
```

(22) HiRDB クライアントと HiRDB サーバの接続を確認します

HiRDB SQL Executer などで，HiRDB クライアントと HiRDB サーバが接続しているか確認してください。また，(23)で系を切り替えても，HiRDB クライアントと HiRDB サーバが接続しているか確認してください。

(23) 系の切り替えを確認します

フェールオーバークラスターマネージャーを起動して，HiRDB の役割を予備系のノードへ移動（系切り替え）します。このとき，所有者ノードに表示されているノード名が予備系のノードであることを確認してください。

その後，予備系側のノードで **pdls -d ust** コマンドを実行し，待機系 HiRDB が稼働中（リターンコードが 0）であることを確認してください。また，現用系側のノードで **pdls -d ust** コマンドを実行し，実行系だった HiRDB が停止状態（リターンコードが 12 又は 16）であることを確認してください。

備考

- 汎用サービス名 HiRDBClusterService は系切り替えが発生したときに，待機系 HiRDB を開始して，実行系 HiRDB を終了するバッチを実行します。
- HiRDB が正常に開始，終了できない場合は，フェールオーバークラスターマネージャー上で「×失敗」と表示されます。この場合，HiRDB が正常に開始，終了できない原因（HiRDB システム定義の誤り，システムファイル未作成，HiRDB 開始済みなど）を取り除いてください。

26.21　停止中のユニットがあるためシステムマネジャユニットの系切り替えに失敗するときの対処方法

停止中のユニットがある状態でシステムマネジャユニットの系切り替えが発生すると，系の切り替え先でシステムマネジャユニットを開始できません。その結果，システムマネジャユニットの系切り替えが失敗します。

次に示すどちらかの対処をすると，停止中のユニットがある状態でもシステムマネジャユニットの系切り替えが実行できます。

1. 縮退起動を指定する
2. pd_ha_mgr_rerun オペランドを指定する

対処方法ごとに前提条件，及び運用時の制限が異なります。使用しているシステムに合わせた方法で対処してください。

26.21.1　縮退起動で対処する方法

停止中のユニットがある場合，次に示すオペランドを指定すると，システムマネジャユニットの系切り替えが実行できます。

- pd_start_level = 1
- pd_start_skip_unit = 停止中のユニット識別子
- pd_reduced_check_time = 縮退起動開始の連絡待ち時間

各オペランドの指定値と系切り替え時の HiRDB の処理を次の表に示します。

表 26–98　縮退起動に関するオペランドと系切り替え時の HiRDB の処理

条件		システムマネジャユニットの系切り替えが発生したときの HiRDB の処理
pd_start_level の指定値	pd_start_skip_unit の指定	
0（省略値）	−	停止中のユニットがある場合，システムマネジャユニットの系切り替えが失敗します。
1	指定なし	停止中のユニットがある場合でも，システムマネジャユニットの系切り替えが実行されます。ただし，各ユニットからの開始処理完了の連絡待ち時間が発生します。連絡待ち時間は pd_reduced_check_time オペランド（省略値は 20 分）で指定します。
	指定あり	停止中のユニットがある場合でも，システムマネジャユニットの系切り替えが実行されます。連絡待ち時間は発生しません。

（凡例）

－：指定する必要はありません。

参考

- システムマネジャユニットの系切り替えの発生時，切り替え先のシステムマネジャユニットでフロントエンドサーバとバックエンドサーバの起動判定処理が行われます。そのため，各ユニットからの開始処理完了の連絡待ち時間が発生します。このとき，一部のユニットが停止している状態で，システムマネジャユニットの系切り替えが発生すると，各ユニットの稼働状態を認識できないため，系の切り替えに失敗したり，系の切り替え時間が掛かったりします。
- 次に示す条件を一つでも満たさない場合は，システムマネジャユニットの系切り替えに失敗します。
  ・フロントエンドサーバが一つ以上稼働している
  ・バックエンドサーバが一つ以上稼働している
  ・ディクショナリサーバが稼働している

pd_start_skip_unit オペランドを指定する場合は，一度 HiRDB を終了する必要があります。そのため，24時間連続稼働するシステムの場合は，もう一つの方法を検討してみてください。

26.21.2　pd_ha_mgr_rerun オペランドを指定して対処する方法

(1)　オペランドの指定値による処理の違い

pd_ha_mgr_rerun オペランドに notwait を指定すると，システムマネジャユニットの系切り替え時（切り替え先の系での開始処理時）に，各ユニットからの開始処理完了の連絡待ちを行いません。
pd_ha_mgr_rerun オペランドの指定値による HiRDB の処理を次の表に示します。

表 26–99　pd_ha_mgr_rerun オペランドの指定値による HiRDB の処理

項目	pd_ha_mgr_rerun の値	
	wait（省略値）	notwait
停止中のユニットがある状態でシステムマネジャユニットの系切り替えの実行可否	系切り替えを実行できません（系の切り替えに失敗します）。	系切り替えを実行できます。
システムマネジャユニットの系切り替え時に発生する処理	• 各ユニットのバージョンチェック処理※1 • システムの構成チェック処理※2	左記の処理は行いません。※3

注※1

待機系のシステムマネジャユニットのバージョンとそのほかのユニットのバージョンが同じかどうかを確認します。

注※2

次に示すことを確認します。

- フロントエンドサーバが一つ以上稼働しているか
- バックエンドサーバが一つ以上稼働しているか
- ディクショナリサーバが稼働しているか

注※3

システムマネジャユニットの系切り替えが完了すると，HiRDB の稼働環境が整っていない場合（フロントエンドサーバが一つも稼働していないなど）でも，KFPS05210-I メッセージ（システム起動完了メッセージ）を出力します。

参考

pd_ha_mgr_rerun オペランドに notwait を指定すると，各ユニットのバージョンチェック処理，及びシステムの構成チェック処理が行われないため，システムマネジャユニットの系切り替えに掛かる時間を短縮できます。

(2) 前提となるシステム構成

pd_ha_mgr_rerun オペランドに notwait を指定する場合，次に示す条件をすべて満たすようにシステムを構築してください。

- システムマネジャユニットに高速系切り替え機能を適用する
- 系切り替え対象ユニットには高速系切り替え機能だけを適用する（ユーザサーバホットスタンバイや，スタンバイレス型系切り替え機能と混在しない）
- システムマネジャとディクショナリサーバを同じユニット内に定義する
- 2 ユニット以上から成る HiRDB/パラレルサーバを構築する
- HiRDB Datareplicator を使用しない

(3) 環境設定

次に示す手順で環境設定をしてください。

〈手順〉

1. **pdadmvr** コマンドで，現用系及び待機系の全ユニットのバージョンが同じかどうかを確認します。
2. **pd_ha_mgr_rerun** オペランドに notwait を指定します。

参考

pd_ha_mgr_rerun オペランドに notwait を指定した場合，システムマネジャユニットの系切り替えが発生したときに，全ユニットのバージョンチェックを HiRDB が行いません。ユ

ニット間でバージョンの不一致がある場合，HiRDB の動作を保証できないため，手順の 1 でバージョンの確認を必ずしてください。

(4) システムマネジャユニットの系切り替えが発生したときの注意事項

(a) システムマネジャユニットの系切り替え後に業務ができない場合

システムマネジャユニットの系切り替えが完了すると，HiRDB の稼働環境が整っていない場合（フロントエンドサーバが一つも稼働していないなど）でも，KFPS05210-I メッセージ（システム起動完了メッセージ）を出力します。このため，システムマネジャユニットの系切り替え完了後，UAP がエラーになるなど業務ができない場合，pdls コマンドで各サーバの稼働状態を確認してください。

(b) HiRDB の開始又は終了処理中にシステムマネジャユニットの系切り替えが発生した場合

HiRDB の開始処理中（又は終了処理中）にシステムマネジャユニットの系切り替えが発生すると，ほかのユニットの動作状態に関係なく，システムマネジャユニットが開始（又は終了）します。そのため，システムマネジャユニット以外のユニットが開始（又は終了）できないことがあります。この場合，一度 HiRDB を強制終了してください。

また，HiRDB の開始又は終了時に次に示す運用を行うと，この状態を回避できます。

- **HiRDB の開始時**
  HiRDB の開始処理が完了した後に，待機系のシステムマネジャユニットを開始してください。
- **HiRDB システムの停止時**
  待機系のシステムマネジャユニットを終了した後に，HiRDB を終了してください。

(c) 停止中のユニットの状態について

システムマネジャユニットの系切り替え時に停止中のユニットに対して，系の切り替え後に pdls -d svr コマンドを実行すると，そのユニットが停止中の間は STOP(N)状態（pdstop コマンドで正常終了した状態）と表示されます。

該当するユニットを開始したのに，ネットワーク障害が原因で STOP(N)状態のままになっている場合は，ネットワーク障害を解決した後に，STOP(N)状態のユニットを一度終了して再度開始してください。

26.22　NTFS キャッシュのフラッシュが発生した場合の対処

HiRDB と Hitachi HA Toolkit Extension でのクラスタ環境で NTFS キャッシュのフラッシュが発生すると，Hitachi HA Toolkit Extension が HiRDB のスローダウンを検知して，系が切り替わることがあります。

26.22.1　系が切り替わる原因

NTFS キャッシュのフラッシュが発生すると，キャッシュのフラッシュが完了するまでボリューム制御ブロックが排他されます。このフラッシュには，NTFS キャッシュ及びすべてのメモリマップトファイルが含まれます。フラッシュされるデータ量は，キャッシュやメモリマップトファイル上にある，ディスクに反映する必要がある未反映のデータ量に依存します。多量のキャッシュフラッシュが発生すると，プログラムが長時間にわたってボリューム制御ブロックの排他待ちとなり，処理の沈み込みが発生することがあります。

参考

ボリューム制御ブロックとは，NTFS ファイルシステムの内部機構であり，入出力などでボリュームを使用する場合，その制御で用いる管理テーブルです。

HiRDB は，複数のプロセスで情報を共用するため，ユニットコントローラ用の共用メモリやグローバルバッファ用の共用メモリを，メモリマップトファイルを用いて実装しています。HiRDB のメモリマップトファイルは，共用メモリ管理方式として regular を使用している場合，%PDUXPLDIR%¥spool¥shm ディレクトリ下に通常ファイルとして作成されます。HiRDB 稼働中は，業務 AP や HiRDB のユティリティなどのトランザクションを実行すると，グローバルバッファ上にキャッシュされるデータ量が増えていきます。多量のデータがキャッシュされた状態でフラッシュが発生すると，メモリマップトファイルがあるドライブへの多量出力が発生し，該当するドライブ上のファイル入出力をする HiRDB のプログラムの処理が沈み込みます。

これによって，本来 HiRDB が 1 秒間隔で行っている Hitachi HA Toolkit Extension への稼働連絡が設定値を過ぎても実行されないため，Hitachi HA Toolkit Extension は HiRDB のスローダウンと判断して系を切り替えます。

キャッシュのフラッシュは，主に次の操作をすると発生します。

- ディスクの管理の操作
- オンライン稼働中の Windows へのログオン及びログオフ操作
- システム情報の SCSI の表示
- システム情報ファイルの出力（SCSI を含む場合）
- デフラグの実施
- diskpart.exe，nfi.exe，又は fsutil.exe の実行

- chkdsk.exe の実行
- HDLM のパスヘルスチェック
- HDLM の dlnkmgr view -path でのパスの状態確認や DLMgetras の実行
- pdchgprc，pdpfresh，又は pdstop コマンドの実行

26.22.2　対処方法

Hitachi HA Toolkit Extension によるスローダウン検知への対処方法を次に示します。

(1)　共用メモリ管理方式の変更

共用メモリ管理方式として page の使用を推奨します。page の場合，メモリマップトファイルを OS のページングファイル上に作成するため，NTFS キャッシュフラッシュによる影響を受けにくくなくなます。

共用メモリ管理方式については，マニュアル「HiRDB システム導入・設計ガイド」の「仮想メモリの確認」を参照してください。

(2)　メモリマップトファイルの移動と右クリックの抑止

共用メモリ管理方式として regular の使用を継続する必要がある場合は，メモリマップトファイルが格納されたディレクトリ（%PDUXPLDIR%¥spool¥shm）を，HiRDB 運用ディレクトリと別のドライブに移動させて，そのドライブを右クリックしない運用をしてください。

参考

- HiRDB 運用ディレクトリと別のドライブに移動させると，HiRDB 運用ディレクトリがあるドライブの，フラッシュする必要があるキャッシュが減少するため，キャッシュフラッシュが発生した場合の沈み込み時間を短縮できます。
- HiRDB 運用ディレクトリと別のドライブに移動させても，HiRDB 運用ディレクトリがあるドライブに大容量のファイルがあり，このファイルが大量にキャッシュされている状態で HiRDB 運用ディレクトリがあるドライブを右クリックした場合，同様の現象が発生します。したがって，ドライブの右クリックは極力しないようにしてください。
- キャッシュのフラッシュタイミングと，次の操作による HiRDB のサーバプロセス生成処理が重なった場合，その生成処理がキャッシュのフラッシュ完了まで遅延します。この場合，Hitachi HA Toolkit Extension でのスローダウン検知は発生しません。
  ・新規 UAP の接続
  ・HiRDB のコマンド及びユティリティの実行

メモリマップトファイルが格納されたディレクトリの移動は，次の手順で行います。

1. HiRDB を正常終了します。

2. HiRDB のサービス，及び HiRDB のクラスタサービスを停止します。
3. %PDUXPLDIR%¥spool¥shm ディレクトリ下のファイルを削除します。
4. [スタートメニュー]-[プログラム]-[管理ツール]-[コンピュータの管理]-[ディスクの管理]を選択します。
5. [ローカルディスク(例：H ドライブに移動させる場合は H:)]-[ドライブ文字とパスの変更]-[追加]を選択し，[参照]からディレクトリ（%PDUXPLDIR%¥spool¥shm）を指定します。
6. Windows のエクスプローラで次の点を確認してください。
   - %PDUXPLDIR%¥spool¥shm のアイコンは，ディレクトリからドライブに変更されます。以降，移動させたドライブ直下がメモリマップトファイルが格納されるディレクトリとなり，HiRDB のサービス及び HiRDB を開始すると，メモリマップトファイルが作成されます。

なお，HiRDB のサービス開始時には，メモリマップトファイルが格納されるディレクトリ直下のファイルをクリアします（ディレクトリはクリアしません）。そのため，移動させたドライブ直下にはファイルを置かないようにしてください。

(3) ユティリティで使用するファイルの格納場所の見直し

ユティリティ実行時に使用するファイルが大容量で，HiRDB 運用ディレクトリがあるドライブ上にあると，スローダウンが発生する可能性があります。HiRDB 運用ディレクトリがあるドライブに，pdload や pdrorg などで使用する大容量のファイルがある場合，HiRDB 運用ディレクトリと別のドライブへ移動するようにしてください。

27 MIB パフォーマンス情報監視機能の運用

この章では，MIB を使用して HiRDB の稼働情報を収集する MIB パフォーマンス情報監視機能の運用方法について説明します。

27.1　MIB パフォーマンス情報監視機能の概要

ここでは，MIB パフォーマンス情報監視機能の概要について説明します。

27.1.1　MIB パフォーマンス情報監視機能とは

ネットワーク上の管理対象オブジェクト（ルータ，プリンタ，データベースなど）を管理するプロトコルに **SNMP** があります。この SNMP を利用して HiRDB の稼働情報（パフォーマンス情報）を **SNMP エージェント**と **MIB** を使用し，一定の間隔で収集します。収集した稼働情報は**管理フレームワーク**に送られて管理されます。これを **MIB パフォーマンス情報監視機能**といいます。

管理フレームワークには，収集した稼働情報をグラフ化する機能があり，HiRDB の稼働情報を時系列に認識できます。また，監視対象の稼働情報の値が一定値を超えた場合に警告を出すなどのイベントを設定できます。これによって，HiRDB の稼働状況の変化を事前に察知でき，障害が発生する前にネットワーク管理者によって対応ができます。

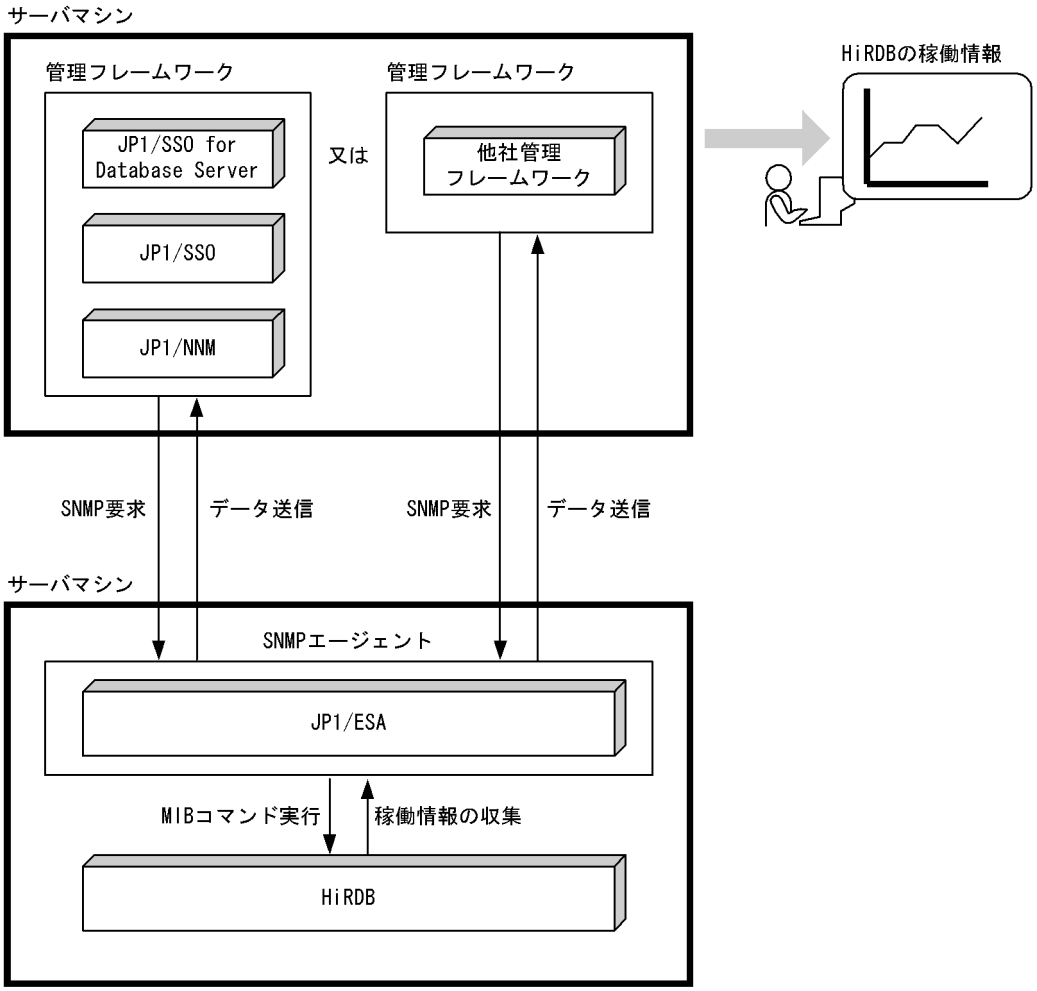
MIB には HiRDB が提供する HiRDB の稼働情報が記述されます。また，MIB には HiRDB の稼働情報を取得する方法が記載されていて，管理フレームワークからの要求に応じて pdbufls コマンドなどの HiRDB のコマンド又はユティリティを実行し，HiRDB の稼働情報を取得します。

参考

収集した HiRDB の稼働情報の管理，ユーザへの通知，及び原因を調査する資料の作成は，管理フレームワークが行います。

MIB パフォーマンス情報監視機能の概要を次の図に示します。

図 27–1　MIB パフォーマンス情報監視機能の概要



〔説明〕

SNMP エージェントである JP1/ESA が，MIB コマンドを使用して一定の間隔で HiRDB の稼働情報を取得します。その稼働情報を管理フレームワークで管理します。

MIB：

ASN.1 表記法で書かれたテキストファイルに，SNMP でアクセス可能な情報を含む変数が記述されています。

MIB コマンド：

HiRDB の稼働情報を取得するためのコマンドです。JP1/ESA から HiRDB に対して実行されます。MIB コマンドの延長で HiRDB のコマンド又はユティリティが実行されて，コマンドの実行結果を JP1/ESA へ返します。

SNMP：

ネットワーク上の管理対象オブジェクト（ルータ，プリンタ，データベースなど）をネットワーク経由で監視・制御するためのプロトコルです。

SNMP エージェント：

管理フレームワークからの要求に従って HiRDB の稼働情報を管理フレームワークに渡したり，HiRDB の稼働情報を取得したりするプログラムです。SNMP エージェントには次に示す製品を使用します。

- JP1/Cm2/Extensible SNMP Agent for Mib Runtime

管理フレームワーク：

SNMP 対応の管理プログラムのことです。推奨する管理フレームワークを次に示します。

- JP1/Cm2/Network Node Manager，JP1/Performance Management/SNMP System Observer，及び JP1/Performance Management/SNMP System Observer for Database Server
- 他社管理フレームワーク

ポイント

MIB を使用したパフォーマンス情報の監視以外にも，JP1/PFM-Agent for HiRDB を使用してパフォーマンス情報を監視できます。HiRDB ではパフォーマンス情報監視手段として JP1/PFM-Agent for HiRDB の使用を推奨しています。

27.1.2　MIB パフォーマンス情報監視機能の目的

MIB パフォーマンス情報監視機能を使用すると，次に示すことができます。

- **HiRDB の稼働状況を分析する**

  監視対象の HiRDB から，RD エリアの使用率やサーバプロセス数などの稼働情報を収集及び集計し，その傾向や推移をグラフィック化できるため，HiRDB の稼働状況をより分かりやすく分析できます。

- **システムの問題点を早期に発見し，原因を調査する資料を作成する**

  監視対象の HiRDB で，グローバルバッファのヒット率が悪くなるなどのパフォーマンスの低下が発生した場合，その現象を HiRDB 管理者に E メールなどを使用して通知できるため，問題点を早期に対策できます。また，その問題点に関連する情報をグラフィック化できるため，原因を調査する資料を作成できます。

27.1.3　MIB 定義ファイル

MIB 定義ファイルには，SNMP でアクセスできる次に示す情報が格納されています。

- 変数名
- 変数のオブジェクト ID
- 変数のデータ型
- 変数のアクセス権

27.1.4　MIB 環境定義ファイル

MIB 環境定義ファイルには監視対象の HiRDB を指定します。また，MIB コマンドの延長で HiRDB のコマンド又はユティリティを実行するために必要な情報（認可識別子など）を指定します。

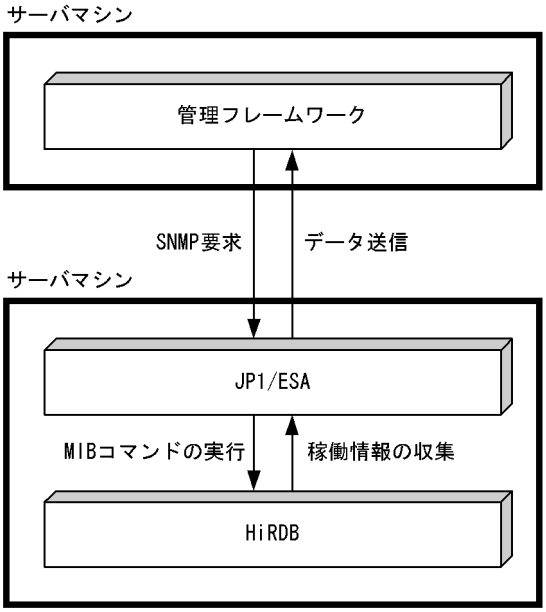
27.2　システム構成

ここでは，MIB パフォーマンス情報監視機能を使用する場合のシステム構成例を説明します。

27.2.1　HiRDB/シングルサーバの場合

HiRDB/シングルサーバに適用する場合のシステム構成を次の図に示します。

図 27–2　HiRDB/シングルサーバに適用する場合のシステム構成



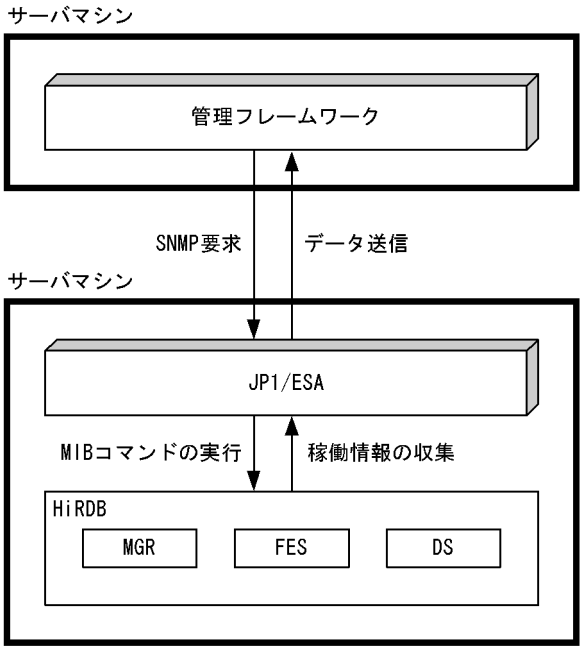
〔説明〕

監視対象の HiRDB があるサーバマシンに JP1/ESA をインストールします。

27.2.2　HiRDB/パラレルサーバの場合

HiRDB/パラレルサーバに適用する場合のシステム構成を次の図に示します。

図 27–3　HiRDB/パラレルサーバに適用する場合のシステム構成



〔説明〕

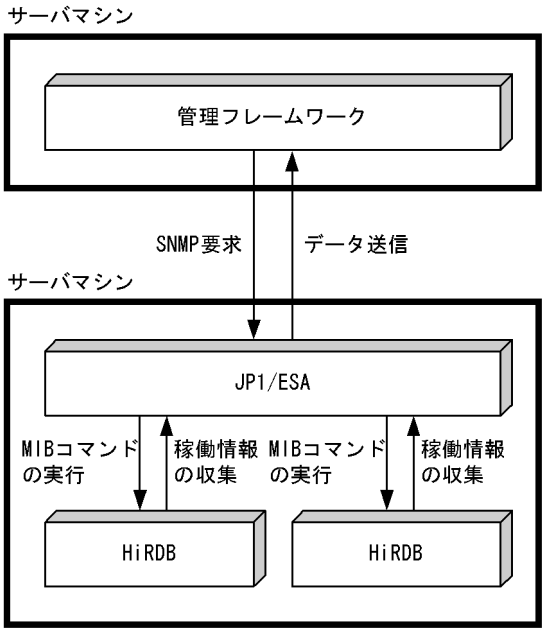
システムマネジャがあるサーバマシンでパフォーマンス情報を収集するため，JP1/ESA はシステムマネジャがあるサーバマシンにインストールします。それ以外のサーバマシンには JP1/ESA をインストールする必要はありません。

MIB コマンドはシステムマネジャがあるサーバマシンで実行されます。

27.2.3　マルチ HiRDB の場合

HiRDB/シングルサーバのマルチ HiRDB 構成に適用する場合のシステム構成を次の図に示します。

図 27–4　HiRDB/シングルサーバのマルチ HiRDB 構成に適用する場合のシステム構成



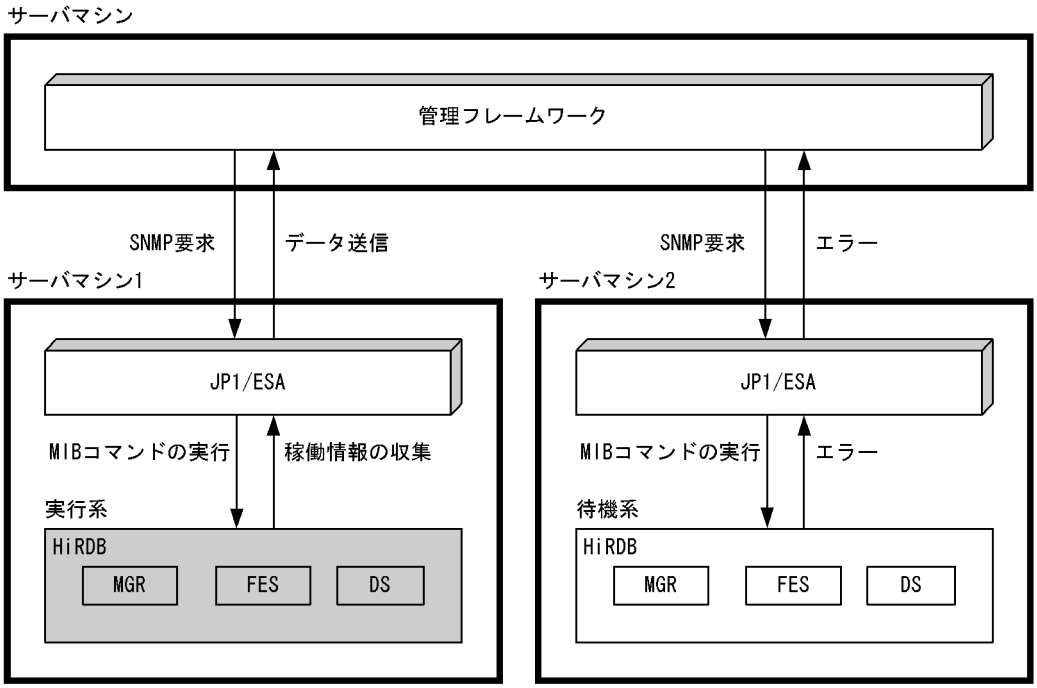
〔説明〕

マルチ HiRDB の場合は，一つの MIB コマンドで複数の HiRDB を監視できます。複数の HiRDB を監視するには，MIB 環境定義ファイルに複数の HiRDB を指定する必要があります。なお，必ずしもすべての HiRDB を監視する必要はありません。複数の HiRDB のうち一つだけを監視対象にすることもできます。

27.2.4　系切り替え機能を使用する場合

1:1 系切り替え構成に適用する場合のシステム構成を次の図に示します。

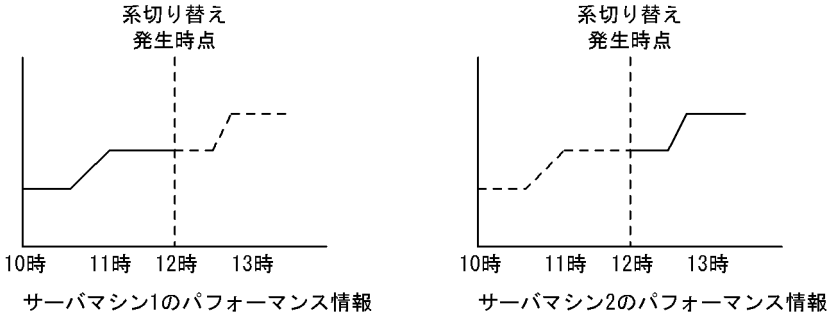
図 27–5　1:1 系切り替え構成に適用する場合のシステム構成



〔説明〕

系切り替え機能を使用する場合は，待機系のサーバマシンでも監視を行うように設定してください。そうすると，系切り替えの発生時に切り替え先のサーバマシンで監視を続行できます。ただし，収集したパフォーマンス情報は，次の図に示すように連続していません。別の HiRDB を監視しているように見えます。

図 27–6　1:1 系切り替え構成に適用した場合のパフォーマンス情報の見え方



27.3　環境設定

27.3.1　環境設定の手順

ここでは，MIB パフォーマンス情報監視機能を使用するときの HiRDB についての環境設定方法を説明します。(1)以降の作業を行ってください。

(1)　共通の MIB 環境定義ファイル（pdmibtgt）を作成する

共通の MIB 環境定義ファイル（pdmibtgt）を作成します。このファイルには，監視対象の HiRDB の HiRDB 識別子と HiRDB 運用ディレクトリ名を指定します。指定例を次に示します。

```
pdmib -a HiRDBシステム識別子 -d HiRDB運用ディレクトリ名
```

〔説明〕

- 共通の MIB 環境定義ファイルは，**HiRDB のインストールディレクトリ¥sample¥pdmibtgt** に作成してください。
- HiRDB/パラレルサーバの場合は，システムマネジャがあるサーバマシンに共通の MIB 環境定義ファイルを作成してください。
- マルチ HiRDB 環境で複数の HiRDB を監視する場合は複数行指定してください。指定例を次に示します。

```
pdmib -a HRD1 -d C:¥hirdb_x1
pdmib -a HRD2 -d C:¥hirdb_x2
```

(2)　個別の MIB 環境定義ファイル（pdmibenv）を作成する

個別の MIB 環境定義ファイル（pdmibenv）を作成します。このファイルには，HiRDB のユティリティ又はコマンドを実行する環境を指定します。指定例を次に示します。

```
putenv PDHOST 接続するHiRDBサーバのホスト名
putenv PDDIR 接続するHiRDBサーバのHiRDB運用ディレクトリ名
putenv PDNAMEPORT 接続するHiRDBサーバのポート番号
putenv PDUSER 認可識別子及びパスワード
putenv LANG 文字コード種別
```

〔説明〕

- 各オペランドの指定方法はクライアント環境定義と同じです。クライアント環境定義については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。
- 個別の MIB 環境定義ファイルは，**%PDDIR%¥conf¥pdmibenv** に作成してください。**%PDDIR%** は(1)で指定した HiRDB 運用ディレクトリ名になります。

注意事項

HiRDB 稼働中に pdchgconf コマンドで PDNAMEPORT を変更した場合，pdmibenv ファイルも変更する必要があります。

(3)　MIB 定義ファイルを登録する

HiRDB が提供する MIB 定義ファイルを JP1/ESA に登録します。HiRDB/パラレルサーバの場合はシステムマネジャがあるサーバマシンで登録してください。JP1/ESA への登録方法については，JP1/ESA のヘルプを参照してください。

MIB 定義ファイルのパス名を次に示します。

- HiRDB のインストールディレクトリ¥sample¥hirdbMIB

マルチ HiRDB の場合の注意事項

- pdmibtgt ファイルを作成したディレクトリの HiRDB の MIB 定義ファイルを登録してください。
- MIB 定義ファイルを登録した HiRDB をアンインストールした場合，アンインストールしていない HiRDB で MIB 定義ファイルを再登録する必要があります。

(4)　環境変数を設定する

セットアップ識別子を指定してインストールした場合，システム環境変数に PDDIR 環境変数と PDUXPLDIR 環境変数を追加します。PDDIR 環境変数にはインストールディレクトリを，PDUXPLDIR 環境変数にはインストールディレクトリ¥UXPLDIR を設定します。

(5)　収集するパフォーマンス情報を選択する

(a)　JP1/SSO for Database を使用する場合

収集するパフォーマンス情報を JP1/SSO for Database で選択します。選択方法については，マニュアル「JP1/Performance Management/SNMP System Observer 拡張リソース管理編」を参照してください。

注意事項

監視する HiRDB の環境及び情報収集する MIB によっては，SNMP タイムアウトエラーとなる場合があります。稼働状態によって，JP1/NNM 又は JP1/ESA の SNMP タイムアウト時間を調整してください。

(b)　他社管理フレームワークを使用する場合

他社管理フレームワークのマニュアルを参照してください。

27.4　MIB 定義ファイル

ここでは，MIB 定義ファイルの規則と MIB 定義ファイルに格納される MIB テーブルについて説明します。

27.4.1　HiRDB が提供する MIB 定義ファイルの規則

HiRDB が提供する MIB 定義ファイルの規則を次の表に示します。

表 27–1　HiRDB が提供する MIB 定義ファイルの規則

項目	HiRDB が提供する MIB 定義ファイルの規則
ファイル名	ファイル名は hirdbMIB です。
オブジェクト ID	HiRDB のオブジェクト ID は，.iso.org.dod.internet.private.enterprises.hitachi.systemExMib.hirdbMibs.HiRDB です。数値表現の場合は，1.3.6.1.4.1.116.5.24.2 です。このオブジェクト ID の下位に HiRDB のパフォーマンス情報を持つオブジェクトがあります。
注釈（DESCRIPTION）	1 行目にオブジェクト ID のコメントを，2 行目以降に JP1/ESA の制御コマンドを記述しています。
インデクス（INDEX）	MIB オブジェクトを特定するためのインデクスです。例えば，ある RD エリアの情報を特定するには，次に示す二つのインデクスが必要です。 • HiRDB システムインデクス • RD エリアインデクス
アクセス権（ACCESS）	オブジェクトに対応するアクセス権です。HiRDB ではテーブル，エントリを除くすべてのオブジェクトを read-only とし，GetRequest 又は GetNextRequest は許可，SetRequest は拒否とします。テーブル及びエントリは not-access とし，アクセスできません。
オブジェクトタイプ（SYNTAX）	文字は DisplayString（ASCII），数値は INTEGER（−（2^{31}−1）～2^{31}−1）又は Gauge（0～2^{32}−1）を使用します。テーブルは SEQUENCE を使用します。

27.4.2　HiRDB が提供する MIB テーブルの一覧

HiRDB が提供する MIB テーブルの一覧を次の表に示します。

表 27–2　HiRDB が提供する MIB テーブルの一覧

テーブル名	説明
サーバ稼働状態テーブル（hirServerStatusTable）	ある時点のサーバ稼働状態のパフォーマンス情報が格納されます。テーブルの内容については，「サーバ稼働状態テーブル（hirServerStatusTable）」を参照してください。
作業表用 HiRDB ファイルシステム領域テーブル（hirFileSystemTable）	一定時間内に収集した作業表用ファイル用の HiRDB ファイルシステム領域のパフォーマンス情報が格納されます。テーブルの内容については，「作業表用 HiRDB ファイルシステム領域テーブル（hirFileSystemTable）」を参照してください。

テーブル名	説明
RD エリアテーブル（hirRdareaStatusTable）	一定時間内に収集した RD エリアのパフォーマンス情報が格納されます。テーブルの内容については，「RD エリアテーブル（hirRdareaStatusTable）」を参照してください。
RD エリア詳細テーブル（hirRdareaDetStatusTable）	一定時間内に収集した RD エリアのパフォーマンス情報が格納されます。テーブルの内容については，「RD エリア詳細テーブル（hirRdareaDetStatusTable）」を参照してください。
グローバルバッファテーブル（hirBufferStatusTable）	一定時間内に収集したグローバルバッファのパフォーマンス情報が格納されます。テーブルの内容については，「グローバルバッファテーブル（hirBufferStatusTable）」を参照してください。
RD エリア–HiRDB ファイルシステム領域テーブル（hirRdareaFileTable）	一定時間内に収集した RD エリア用の HiRDB ファイルシステム領域のパフォーマンス情報が格納されます。テーブルの内容については，「RD エリア–HiRDB ファイルシステム領域テーブル（hirRdareaFileTable）」を参照してください。
SYS 統計情報テーブル（hirStatisInfSysTable）	システムの稼働に関する統計情報がサーバ単位に収集されます。テーブルの内容については，「SYS 統計情報テーブル（hirStatisInfSysTable）」を参照してください。

27.5　サーバ稼働状態テーブル（hirServerStatusTable）

サーバ稼働状態テーブルには，ある時点のサーバ稼働状態のパフォーマンス情報が格納されます。ユニット及びサーバごとに 1 インスタンス作成されます。

パフォーマンス情報収集時の注意事項

HiRDB の性能に影響を与えないように，パフォーマンス情報の収集間隔は 60 秒以上に設定してください。

サーバ稼働状態テーブルの構成を次の表に示します。

表 27–3　サーバ稼働状態テーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
1.	hirServerStatusTable	サーバ稼働状態テーブル	SEQUENCE	not-access	−
1.1	hirServerStatusEntry	サーバ稼働状態エントリ	SEQUENCE	not-access	−
1.1.1	hirServerStatusSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
1.1.2	hirServerStatusHostIndex	HiRDB ホスト識別インデクス	INTEGER	read-only	MIB コマンド
1.1.3	hirServerStatusUnitIndex	HiRDB ユニット識別インデクス	INTEGER	read-only	MIB コマンド
1.1.4	hirServerStatusServerIndex	HiRDB サーバ識別インデクス	INTEGER	read-only	MIB コマンド
1.1.5	hirServerStatusInstance	インスタンス名 HiRDB 識別子:ホスト名:ユニット識別子:サーバ名	Display String	read-only	MIB コマンド
1.1.6	hirServerStatusHirID	HiRDB 識別子	Display String	read-only	MIB コマンド
1.1.7	hirServerStatusHost	ホスト名	Display String	read-only	pdls -d svr
1.1.8	hirServerStatusServerName	サーバ名 ユニット情報の場合は"********"が表示されます。	Display String	read-only	pdls -d svr
1.1.9	hirServerStatusStartTime	各サーバ又はユニットの開始時刻 該当サーバ又はユニットが停止中の場合は 999999 が表示されます。	Display String	read-only	pdls -d svr
1.1.10	hirServerStatusStatus	サーバ又はユニットの状態 指定できる値を次に示します。	INTEGER	read-only	pdls -d svr

ID	オブジェクト	説明	タイプ	権限	データソース
		1:ACTIVE 2:STOP 3:STOP（N） 4:STOP（F） 5:STOP（A） 6:START（I） 7:SUSPEND 8:STARTING 9:STOPPING 10:TRNPAUSE			
1.1.11	hirServerStatusUnitID	ユニット識別子	Display String	read-only	pdls -d svr
1.1.12	hirServerStatusDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）－：該当しません。

27.6 作業表用 HiRDB ファイルシステム領域テーブル（hirFileSystemTable）

作業表用 HiRDB ファイルシステム領域テーブルには，一定時間内に収集した作業表用ファイル用の HiRDB ファイルシステム領域のパフォーマンス情報が格納されます。

パフォーマンス情報収集時の注意事項

HiRDB の性能に影響を与えないように，パフォーマンス情報の収集間隔は 600 秒以上に設定してください。

作業表用 HiRDB ファイルシステム領域テーブルの構成を次の表に示します。

表 27–4　作業表用 HiRDB ファイルシステム領域テーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
3.	hirFileSystemTable	作業表用 HiRDB ファイルシステム領域テーブル	SEQUENCE	not-access	−
3.1	hirFileSystemEntry	作業表用 HiRDB ファイルシステム領域エントリ	SEQUENCE	not-access	−
3.1.1	hirFileSystemSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
3.1.2	hirFileSystemFileIndex	HiRDB ファイルシステム識別インデクス	INTEGER	read-only	MIB コマンド
3.1.3	hirFileSystemInstance	インスタンス名 HiRDB 識別子:ホスト名:HiRDB ファイルシステム領域名	Display String	read-only	MIB コマンド
3.1.4	hirFileSystemHirID	HiRDB 識別子	Display String	read-only	MIB コマンド
3.1.5	hirFileSystemAvailableExpandCount	HiRDB ファイルシステム領域の増分回数の上限値	INTEGER	read-only	pdfstatfs -d -b
3.1.6	hirFileSystemAvailableFileSize	HiRDB ファイルシステム領域で一つのファイルとして確保できる容量の最大値（単位：キロバイト）	INTEGER	read-only	pdfstatfs -d -b
3.1.7	hirFileSystemCurrentExpandCount	HiRDB ファイルシステム領域の増分回数の合計値	INTEGER	read-only	pdfstatfs -d -b
3.1.8	hirFileSystemCurrentFileCount	作成した HiRDB ファイル数	INTEGER	read-only	pdfstatfs -d -b
3.1.9	hirFileSystemFreeAreaCount	不連続な空き領域の総数	INTEGER	read-only	pdfstatfs -d -b

ID	オブジェクト	説明	タイプ	権限	データソース
3.1.10	hirFileSystemHiRDBFileSystemAreaName	HiRDB ファイルシステム領域名	Display String	read-only	pdfstatfs -d -b
3.1.11	hirFileSystemPeakCapacity	HiRDB ファイルシステム領域中に割り当てられた領域の最大値を 0 にリセットした時点から現在までの期間のユーザ最大使用量※（単位：キロバイト）	INTEGER	read-only	pdfstatfs -d -b
3.1.12	hirFileSystemRemainingFileCount	作成できる HiRDB ファイル数（作成できるファイル数の上限−作成したファイル数）	INTEGER	read-only	pdfstatfs -d -b
3.1.13	hirFileSystemRemainingUserArea	ユーザに割り当てられた領域の中で，未使用領域（HiRDB ファイルとして割り当てられていない領域）の容量（単位：キロバイト）	INTEGER	read-only	pdfstatfs -d -b
3.1.14	hirFileSystemSectorSize	HiRDB ファイルシステム領域のセクタ長（単位：バイト） pdfmkfs コマンドの-s オプションで指定したセクタ長です（省略時は 1024）。	INTEGER	read-only	pdfstatfs -d -b
3.1.15	hirFileSystemHost	ホスト名	Display String	read-only	pdfstatfs -d -b
3.1.16	hirFileSystemUserAreacapacity	ユーザ領域の HiRDB ファイルシステム領域の容量（単位：キロバイト）	INTEGER	read-only	pdfstatfs -d -b
3.1.17	hirFileSystemDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）−：該当しません。

注※

該当する HiRDB ファイルシステム領域に対して，pdfstatfs -c コマンドを実行するまでは，PEAK_CAPACITY 値がクリアされないため，このフィールドが減少することはありません。

27.7　RD エリアテーブル（hirRdareaStatusTable）

RD エリアテーブルには，一定時間内に収集した RD エリアのパフォーマンス情報が格納されます。RD エリアごとに 1 行作成されます。

パフォーマンス情報収集時の注意事項

- HiRDB の性能に影響を与えないように，パフォーマンス情報の収集間隔は 600 秒以上に設定してください。
- 共用 RD エリアの場合，更新できるバックエンドサーバのパフォーマンス情報だけが収集されます。

RD エリアテーブルの構成を次の表に示します。

表 27–5　RD エリアテーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
4.	hirRdareaStatusTable	RD エリアテーブル	SEQUENCE	not-access	−
4.1	hirRdareaStatusEntry	RD エリアエントリ	SEQUENCE	not-access	−
4.1.1	hirRdareaStatusSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
4.1.2	hirRdareaStatusRdareaIndex	RD エリア識別インデクス	INTEGER	read-only	MIB コマンド
4.1.3	hirRdareaStatusInstance	インスタンス名 HiRDB 識別子:サーバ名:RD エリア名	Display String	read-only	MIB コマンド
4.1.4	hirRdareaStatusHirID	HiRDB 識別子	Display String	read-only	MIB コマンド
4.1.5	hirRdareaStatusBufferName	グローバルバッファ名	Display String	read-only	pdbufls -k def-s SERVR_NAME
4.1.6	hirRdareaStatusExtensionSegmentSize	増分セグメント数 RD エリアの増分指定がない場合は 0 になります。	INTEGER	read-only	ディクショナリ表※
4.1.7	hirRdareaStatusFileCount	HiRDB ファイル数	INTEGER	read-only	ディクショナリ表※
4.1.8	hirRdareaStatusIndexCount	格納インデクス数（定義数）	INTEGER	read-only	ディクショナリ表※
4.1.9	hirRdareaStatusPageSize	ページ長（単位：バイト）	INTEGER	read-only	ディクショナリ表※
4.1.10	hirRdareaStatusRDAREAName	RD エリア名	Display String	read-only	pddbls -r ALL -a

ID	オブジェクト	説明	タイプ	権限	データソース
4.1.11	hirRdareaStatusRDAREAStatus	RD エリアの状態 指定できる値を次に示します。 CLOSE，CLOSE HOLD，CLOSE HOLD（INQ），CLOSE HOLD（CMD），CLOSE HOLD（BU），CLOSE HOLD（BU I），CLOSE HOLD（BU W），CLOSE HOLD（BU IW），CLOSE ACCEPT-HOLD，HOLD，HOLD（INQ），HOLD（CMD），HOLD（BU），HOLD（BU I），HOLD（BU W），HOLD（BU IW），ACCEPT-HOLD，OPEN	Display String	read-only	pddbls -r ALL -a
4.1.12	hirRdareaStatusRDAREAType	RD エリア種別 指定できる値を次に示します。 MAST，DDIR，DDIC，DLOB，USER，ULOB，LIST，RGST，RLOB	Display String	read-only	pddbls -r ALL -a
4.1.13	hirRdareaStatusSegmentSize	セグメントサイズ（単位：ページ）	INTEGER	read-only	ディクショナリ表※
4.1.14	hirRdareaStatusServerName	サーバ名	Display String	read-only	pddbls -r ALL -a
4.1.15	hirRdareaStatusTableCount	格納表数（定義数）	INTEGER	read-only	ディクショナリ表※
4.1.16	hirRdareaStatusTotalRDAREASegments	RD エリア内の全セグメント数	INTEGER	read-only	pddbls -r ALL -a
4.1.17	hirRdareaStatusUnusedRDAREASegments	RD エリア内の未使用セグメント数	INTEGER	read-only	pddbls -r ALL -a
4.1.18	hirRdareaStatusDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）－：該当しません。

注※　次に示す SQL 文の結果です。

```
SELECT PAGE_SIZE,SEGMENT_SIZE,FILE_COUNT,N_TABLE,N_INDEX,EXTENSION_SEGMENT_SIZE
      FROM "MASTER".SQL_RDAREAS
```

27.8　RD エリア詳細テーブル（hirRdareaDetStatusTable）

RD エリア詳細テーブルには，一定時間内に収集した RD エリアのパフォーマンス情報が格納されます。RD エリアごとに 1 行作成されます。

パフォーマンス情報収集時の注意事項

- HiRDB の性能に影響を与えないように，パフォーマンス情報の収集間隔は 3600 秒以上に設定してください。
- 共用 RD エリアの場合，更新できるバックエンドサーバのパフォーマンス情報だけが収集されます。
- pddbst コマンドが実行できない場合，pddbst コマンドからデータを取得するフィールドには，0（数値フィールドの場合）又は空白（文字列フィールドの場合）が設定されます。

  次に示す条件をすべて満たす場合に pddbst コマンドが実行できます。

  ・対象 RD エリアがデータディクショナリ用 RD エリア，ユーザ用 RD エリア，LOB 用 RD エリア，レジストリ用 RD エリア，又はレジストリ LOB 用 RD エリアである

  ・対象 RD エリアが閉塞なしオープン状態，又は参照可能閉塞オープン状態である

RD エリア詳細テーブルの構成を次の表に示します。

表 27–6　RD エリア詳細テーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
5.	hirRdareaDetStatusTable	RD エリア詳細テーブル	SEQUENCE	not-access	−
5.1	hirRdareaDetStatusEntry	RD エリア詳細エントリ	SEQUENCE	not-access	−
5.1.1	hirRdareaDetStatusSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
5.1.2	hirRdareaDetStatusRdareaIndex	RD エリア識別インデクス	INTEGER	read-only	MIB コマンド
5.1.3	hirRdareaDetStatusInstance	インスタンス名 HiRDB 識別子:サーバ名:RD エリア名	Display String	read-only	MIB コマンド
5.1.4	hirRdareaDetStatusHirID	HiRDB 識別子	Display String	read-only	MIB コマンド
5.1.5	hirRdareaDetStatusAutoExtendErrorCode	自動増分できない場合のエラーコード	INTEGER	read-only	pddbst -k phys -f
5.1.6	hirRdareaDetStatusAutoExtendStatus	自動増分機能の抑止状態 SUP：抑止されている状態。 NOSUP：抑止されていない状態	Display String	read-only	pddbst -k phys -f

ID	オブジェクト	説明	タイプ	権限	データソース
5.1.7	hirRdareaDetStatusAutoExtendUse	自動増分機能の使用状況 USE：自動増分機能を使用している NOUSE：自動増分機能を使用していない	Display String	read-only	pddbst -k phys -f
5.1.8	hirRdareaDetStatusBufferName	グローバルバッファ名	Display String	read-only	pdbufls -k def-s SERVR_NAME
5.1.9	hirRdareaDetStatusExtensionSegmentSize	増分セグメント数（RD エリアの増分指定がない場合は 0）	INTEGER	read-only	ディクショナリ表※1
5.1.10	hirRdareaDetStatusFileCount	HiRDB ファイル数	INTEGER	read-only	ディクショナリ表※1
5.1.11	hirRdareaDetStatusFreezeSpecified	RD エリアが更新凍結されているかどうかの情報	Display String	read-only	pddbst -k phys -f※2
5.1.12	hirRdareaDetStatusFullUsedPage	満杯ページ数の比率（単位：%）	INTEGER	read-only	pddbst -k phys -f※7
5.1.13	hirRdareaDetStatusFullUsedPages	満杯ページ数	INTEGER	read-only	pddbst -k phys -f※7
5.1.14	hirRdareaDetStatusGenNumber	0	INTEGER	read-only	pddbst -k phys -f
5.1.15	hirRdareaDetStatusHoldCode1	該当する RD エリアの閉塞要因コード	INTEGER	read-only	pddbst -k phys -f
5.1.16	hirRdareaDetStatusHoldCode2	Hold Status 1 の一つ前の閉塞要因コード情報	INTEGER	read-only	pddbst -k phys -f
5.1.17	hirRdareaDetStatusHoldStatus1	該当する RD エリアの閉塞種別 1:Error Shutdown（障害閉塞） 2:Command Shutdown（HiRDB の障害検知によるコマンド閉塞） 3:pddbst が出力した値（FLT 又は CMD 以外の値の場合） 4:空白（それ以外の場合）	Display String	read-only	pddbst -k phys -f
5.1.18	hirRdareaDetStatusHoldStatus2	Hold Status 1 の一つ前の閉塞情報	Display String	read-only	pddbst -k phys -f
5.1.19	hirRdareaDetStatusHoldTime1	該当する RD エリアの閉塞時刻	DisplayString	read-only	pddbst -k phys -f
5.1.20	hirRdareaDetStatusHoldTime2	Hold Status 1 の一つ前の閉塞時刻情報	DisplayString	read-only	pddbst -k phys -f
5.1.21	hirRdareaDetStatusIndexCount	格納インデクス数（定義数）	INTEGER	read-only	ディクショナリ表※1

ID	オブジェクト	説明	タイプ	権限	データソース
5.1.22	hirRdareaDetStatusLastSegment	使用されているセグメントの最後を示す位置情報 Segment Over が Y のときは，常に最後のセグメントを示します。	INTEGER	read-only	pddbst -k phys -f※4
5.1.23	hirRdareaDetStatusLobmapOver	LOB 管理エントリがすべて使用されているかどうかの情報 Y：すべて使用されている N：未使用のエントリがある	Display String	read-only	pddbst -k phys -f※3
5.1.24	hirRdareaDetStatusOriginalRDAREAName	RD エリア名	Display String	read-only	pddbst -k phys -f
5.1.25	hirRdareaDetStatusPageSize	ページ長（単位：バイト）	INTEGER	read-only	ディクショナリ表※1
5.1.26	hirRdareaDetStatusRDAREAName	RD エリア名	Display String	read-only	pddbls -r ALL -a
5.1.27	hirRdareaDetStatusRDAREAStatus	RD エリアの状態 次に示す値を指定できます。 CLOSE，CLOSE HOLD，CLOSE HOLD（INQ），CLOSE HOLD（CMD），CLOSE HOLD（BU），CLOSE HOLD（BU I），CLOSE HOLD（BU W），CLOSE HOLD（BU IW），CLOSE ACCEPT-HOLD，HOLD，HOLD（INQ），HOLD（CMD），HOLD（BU），HOLD（BU I），HOLD（BU W），HOLD（BU IW），ACCEPT-HOLD，OPEN	Display String	read-only	pddbls -r ALL -a
5.1.28	hirRdareaDetStatusRDAREAType	RD エリアの種別 次に示す値を指定できます。 MAST，DDIR，DDIC，DLOB，USER，ULOB，LIST，RGST，RLOB	Display String	read-only	pddbls -r ALL -a
5.1.29	hirRdareaDetStatusReplicaRDAREAs	0	INTEGER	read-only	pddbst -k phys -f
5.1.30	hirRdareaDetStatusSegmentOver	LOB 用 RD エリアの格納状態が乱れているかどうかの情報 Y：乱れている N：乱れていない	Display String	read-only	pddbst -k phys -f※3
5.1.31	hirRdareaDetStatusSegmentSize	セグメントサイズ	INTEGER	read-only	ディクショナリ表※1
5.1.32	hirRdareaDetStatusServerName	サーバ名	Display String	read-only	pddbls -r ALL -a

ID	オブジェクト	説明	タイプ	権限	データソース
5.1.33	hirRdareaDetStatusTableCount	格納表数（定義数）	INTEGER	read-only	ディクショナリ表※1
5.1.34	hirRdareaDetStatusTotalPages	RD エリア内のセグメント中のページ数の合計（使用中ページ数＋未使用ページ数）	INTEGER	read-only	pddbst -k phys -f※8
5.1.35	hirRdareaDetStatusTotalRDAREASegments	RD エリア内の全セグメント数	INTEGER	read-only	pddbls -r ALL -a
5.1.36	hirRdareaDetStatusUnusedRDAREASegments	RD エリア内の未使用セグメント数	INTEGER	read-only	pddbls -r ALL -a
5.1.37	hirRdareaDetStatusUsed	使用中セグメントの割合	INTEGER	read-only	0 固定
5.1.38	hirRdareaDetStatusUsedPage	使用中ページ数の比率（単位：%）	INTEGER	read-only	pddbst -k phys -f※6
5.1.39	hirRdareaDetStatusUsedPages	使用中ページ数	INTEGER	read-only	pddbst -k phys -f※5
5.1.40	hirRdareaDetStatusDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）−：該当しません。

注※1

次に示す SQL 文の結果です。

```
SELECT PAGE_SIZE,SEGMENT_SIZE,FILE_COUNT,N_TABLE,N_INDEX,EXTENSION_SEGMENT_SIZE FROM ”MASTER”.SQL_RDAREAS
```

注※2

LOB 用 RD エリア以外の場合は空白になります。

注※3

LOB 用 RD エリア以外の場合は空白になります。

注※4

LOB 用 RD エリア以外の場合は 0 になります。

注※5

LOB 用 RD エリアの場合は USED_AREA_SEG の値になります。

注※6

LOB 用 RD エリアの場合は PERCENT_USED の値になります。

注※7

LOB 用 RD エリアの場合は 0 になります。

注※8

LOB 用 RD エリアの場合は TOTAL_AREA_SEG の値になります。

27.9　グローバルバッファテーブル（hirBufferStatusTable）

グローバルバッファテーブルには，一定時間内に収集したグローバルバッファのパフォーマンス情報が格納されます。グローバルバッファごとに 1 行作成されます。

パフォーマンス情報収集時の注意事項

HiRDB の性能に影響を与えないために，収集間隔は 60 秒以上に設定してください。

グローバルバッファテーブルの構成を次の表に示します。

表 27–7　グローバルバッファテーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
6.	hirBufferStatusTable	グローバルバッファテーブル	SEQUENCE	not-access	−
6.1	hirBufferStatusEntry	グローバルバッファエントリ	SEQUENCE	not-access	−
6.1.1	hirBufferStatusSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
6.1.2	hirBufferStatusBufferIndex	グローバルバッファ識別インデクス	INTEGER	read-only	MIB コマンド
6.1.3	hirBufferStatusInstance	インスタンス名 HiRDB 識別子:サーバ名:グローバルバッファ名	Display String	read-only	MIB コマンド
6.1.4	hirBufferStatusHirlD	HiRDB 識別子	Display String	read-only	MIB コマンド
6.1.5	hirBufferStatusBufferName	グローバルバッファ名	Display String	read-only	pdbufls -k sts -d
6.1.6	hirBufferStatusBufferPoolHitRate	グローバルバッファプールのヒット率（単位：%）	INTEGER	read-only	pdbufls -k sts -d
6.1.7	hirBufferStatusCurrentReferenceBuffers	カレントの参照バッファ数	INTEGER	read-only	pdbufls -k sts -d
6.1.8	hirBufferStatusCurrentUpdateBuffers	カレントの更新バッファ数	INTEGER	read-only	pdbufls -k sts -d
6.1.9	hirBufferStatusDBSyncs	シンクポイントの発生回数	Gauge	read-only	pdbufls -k sts -d
6.1.10	hirBufferStatusLOBBufferInputPages	LOB バッファ一括入力ページ数	Gauge	read-only	pdbufls -k sts -d
6.1.11	hirBufferStatusLOBBufferOutputPages	LOB バッファ一括出力ページ数	Gauge	read-only	pdbufls -k sts -d

ID	オブジェクト	説明	タイプ	権限	データソース
6.1.12	hirBufferStatusLOBBufferReadRequests	LOB バッファ READ 要求回数	Gauge	read-only	pdbufls -k sts -d
6.1.13	hirBufferStatusLOBBufferWriteRequests	LOB バッファ WRITE 要求回数	Gauge	read-only	pdbufls -k sts -d
6.1.14	hirBufferStatusOutOfBuffer	バッファ不足発生回数	Gauge	read-only	pdbufls -k sts -d
6.1.15	hirBufferStatusPrefetchBufferShortages	プリフェッチバッファ不足発生回数	Gauge	read-only	pdbufls -k sts -d
6.1.16	hirBufferStatusPrefetchHitRate	プリフェッチヒット率（単位：%）	INTEGER	read-only	pdbufls -k sts -d
6.1.17	hirBufferStatusPrefetchInputPages	プリフェッチ入力ページ数	Gauge	read-only	pdbufls -k sts -d
6.1.18	hirBufferStatusPrefetchReadRequests	プリフェッチ READ 要求回数	Gauge	read-only	pdbufls -k sts -d
6.1.19	hirBufferStatusReads	ディスクからの実 READ 回数	Gauge	read-only	pdbufls -k sts -d
6.1.20	hirBufferStatusReferenceBufferFlushes	参照バッファフラッシュ回数	Gauge	read-only	pdbufls -k sts -d
6.1.21	hirBufferStatusReferenceGets	参照 GET 回数	Gauge	read-only	pdbufls -k sts -d
6.1.22	hirBufferStatusReferenceHitRate	参照要求のヒット率	INTEGER	read-only	pdbufls -k sts -d
6.1.23	hirBufferStatusServerName	サーバ名	Display String	read-only	pdbufls -k sts -d
6.1.24	hirBufferStatusUpdateBufferFlushes	更新バッファフラッシュ回数	Gauge	read-only	pdbufls -k sts -d
6.1.25	hirBufferStatusUpdateGets	更新 GET 回数	Gauge	read-only	pdbufls -k sts -d
6.1.26	hirBufferStatusUpdateHitRate	更新要求のヒット率（単位：%）	INTEGER	read-only	pdbufls -k sts -d
6.1.27	hirBufferStatusUpdatedBufferTrigger	デファードライトトリガ時の出力契機となる更新バッファ数	INTEGER	read-only	pdbufls -k sts -d
6.1.28	hirBufferStatusWaits	バッファ排他待ち発生回数	Gauge	read-only	pdbufls -k sts -d

ID	オブジェクト	説明	タイプ	権限	データソース
6.1.29	hirBufferStatusWrites	ディスクへの実 WRITE 回数	Gauge	read-only	pdbufls -k sts -d
6.1.30	hirBufferStatusDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）−：該当しません。

27.10 RD エリア−HiRDB ファイルシステム領域テーブル（hirRdareaFileTable）

RD エリア−HiRDB ファイルシステム領域テーブルには，一定時間内に収集した RD エリア用の HiRDB ファイルシステム領域のパフォーマンス情報が格納されます。HiRDB ファイルシステム領域と RD エリアの組み合わせごとに 1 行作成されます。

パフォーマンス情報収集時の注意事項

- HiRDB の統計情報が取得されている場合にパフォーマンス情報が収集されます。

  統計情報の取得の開始契機

  ・HiRDB の稼働中に統計情報種別（-k オプション）に fil を指定して pdstbegin コマンドを実行したとき

  ・統計情報種別（-k オプション）に fil を指定した pdstbegin オペランドを指定して HiRDB を開始したとき

  統計情報の取得の終了契機

  ・HiRDB の稼働中に pdstend コマンドを実行したとき

  ・HiRDB を終了したとき
- 共用 RD エリアの場合，サーバごとに別の行にパフォーマンス情報を収集します。
- 停止中のユニットのパフォーマンス情報は収集できません。

RD エリア−HiRDB ファイルシステム領域テーブルの構成を次の表に示します。

表 27–8　RD エリア−HiRDB ファイルシステム領域テーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
9.	hirRdareaFileTable	RD エリア−HiRDB ファイルシステム領域テーブル	SEQUENCE	not-access	−
9.1	hirRdareaFileEntry	RD エリア−HiRDB ファイルシステム領域エントリ	SEQUENCE	not-access	−
9.1.1	hirRdareaFileSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
9.1.2	hirRdareaFileServerIndex	サーバ識別インデクス	INTEGER	read-only	MIB コマンド
9.1.3	hirRdareaFileFileIndex	HiRDB ファイル識別インデクス	INTEGER	read-only	MIB コマンド
9.1.4	hirRdareaFileRdareaIndex	RD エリア識別インデクス	INTEGER	read-only	MIB コマンド
9.1.5	hirRdareaFileInstance	インスタンス名 HiRDB 識別子:サーバ名:HiRDB ファイルシステム領域名:RD エリア	Display String	read-only	MIB コマンド

ID	オブジェクト	説明	タイプ	権限	データソース
9.1.6	hirRdareaFileHirID	HiRDB 識別子	Display String	read-only	MIB コマンド
9.1.7	hirRdareaFileAIORead	システム固有情報（AIO READ）	INTEGER	read-only	pdstedit（fil）※1
9.1.8	hirRdareaFileAIOWrite	システム固有情報（AIO WRITE）	INTEGER	read-only	pdstedit（fil）※1
9.1.9	hirRdareaFileClose	クローズ発生回数（CLOSE）	INTEGER	read-only	pdstedit（fil）※1
9.1.10	hirRdareaFileFreeMBytes	ユーザ領域中の未使用領域（HiRDB ファイルとして割り当てられていない領域）の容量（単位：メガバイト）	INTEGER	read-only	pdfstatfs FS_NAME
9.1.11	hirRdareaFileHiRDBFileSystemAreaName	HiRDB ファイルシステム領域名（フルパス表示）（FS NAME）	Display String	read-only	pdstedit（fil）※2
9.1.12	hirRdareaFileHiRDBFileSystemAreaType	HiRDB ファイルシステム領域の種別（DB，DB（NOLOB），SVR）	Display String	read-only	pdfstatfs FS_NAME
9.1.13	hirRdareaFileHost	ホスト名（HOST）	Display String	read-only	pdstedit（fil）
9.1.14	hirRdareaFileIOError	入出力エラー発生回数（IO ERROR）	INTEGER	read-only	pdstedit（fil）※1
9.1.15	hirRdareaFileListIO	システム固有情報（LIST IO）	INTEGER	read-only	pdstedit（fil）※1
9.1.16	hirRdareaFileMbytes	HiRDB ファイルシステム領域中のユーザ領域の総量（単位：メガバイト）	INTEGER	read-only	pdfstatfs FS_NAME
9.1.17	hirRdareaFileOpen	オープン発生回数（OPEN）	INTEGER	read-only	pdstedit（fil）※1
9.1.18	hirRdareaFileRDAREAName	RD エリア名（RDAREA NAME）	Display String	read-only	pdstedit（fil）
9.1.19	hirRdareaFileServerName	サーバ名（SERVER NAME）	Display String	read-only	pdstedit（fil）
9.1.20	hirRdareaFileSyncRead	同期 READ 発生回数（SYNC READ）	INTEGER	read-only	pdstedit（fil）※1
9.1.21	hirRdareaFileSyncWrite	同期 WRITE 回数（SYNC WRITE）	INTEGER	read-only	pdstedit（fil）※1
9.1.22	hirRdareaFileDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）－：該当しません。

注　pdstedit の後ろにある（　）は，統計解析ユティリティの編集項目を意味しています。

注※1

fil_DAT 内の HOST，SERVER，FILE NAME から HiRDB ファイル名を取り除いた HiRDB ファイルシステム領域名，RDAREA NAME の組でグループ分けをして，各フィールドの数値データを合計します。数値データでなく「**********」が出力されている場合はオーバフローを意味しているため，0 が設定されます。

注※2

フルパス表示の HiRDB ファイル名から HiRDB ファイル名を取り除いて作成します。例えば，C:¥users¥hirdb_s¥area¥rdsys04¥rddata01 の場合，C:¥users¥hirdb_s¥area¥rdsys04 を取得します。

27.11　SYS 統計情報テーブル（hirStatisInfSysTable）

SYS 統計情報テーブルには，システムの稼働に関する統計情報がサーバ単位に収集されます。サーバごとに 1 行作成されます。

パフォーマンス情報収集時の注意事項

- HiRDB の統計情報が取得されている場合にパフォーマンス情報が収集されます。

  統計情報の取得の開始契機

  ・HiRDB の稼働中に統計情報種別（-k オプション）に sys を指定して pdstbegin コマンドを実行したとき

  ・統計情報種別（-k オプション）に sys を指定した pdstbegin オペランドを指定して HiRDB を開始したとき

  統計情報の取得の終了契機

  ・HiRDB の稼働中に pdstend コマンドを実行したとき

  ・HiRDB を終了したとき
- 停止中のユニットのパフォーマンス情報は収集できません。
- 収集間隔と pdstbegin コマンド又は pdstbegin オペランドに指定する時間間隔（-m オプション）は同じ値にしてください。

SYS 統計情報テーブルの構成を次の表に示します。

表 27–9　SYS 統計情報テーブルの構成

ID	オブジェクト	説明	タイプ	権限	データソース
11.	hirStatisInfSysTable	SYS 統計情報テーブル	SEQUENCE	not-access	−
11.1	hirStatisInfSysEntry	SYS 統計情報エントリ	SEQUENCE	not-access	−
11.1.1	hirStatisInfSysSysIndex	HiRDB システム識別インデクス	INTEGER	read-only	MIB コマンド
11.1.2	hirStatisInfSysHostIndex	HiRDB ホスト識別インデクス	INTEGER	read-only	MIB コマンド
11.1.3	hirStatisInfSysUnitIndex	HiRDB ユニット識別インデクス	INTEGER	read-only	MIB コマンド
11.1.4	hirStatisInfSysServerIndex	HiRDB サーバ識別インデクス	INTEGER	read-only	MIB コマンド
11.1.5	hirStatisInfSysInstance	インスタンス名 HiRDB 識別子:ホスト名:サーバ名	Display String	read-only	MIB コマンド
11.1.6	hirStatisInfSysHirID	HiRDB 識別子	Display String	read-only	MIB コマンド

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.7	hirStatisInfSysHost	ホスト名（HOST）	Display String	read-only	pdstedit（sys）
11.1.8	hirStatisInfSysServerName	サーバ名（SERVER_NAME）	Display String	read-only	pdstedit（sys）
11.1.9	hirStatisInfSysScheduleQueueLenFreq	スケジュール待ち行列数の発生件数（SCHEDULE_QUEUE_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.10	hirStatisInfSysScheduleQueueLenMax	スケジュール待ち行列数の最大値（SCHEDULE_QUEUE_LEN_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.11	hirStatisInfSysScheduleQueueLenMin	スケジュール待ち行列数の最小値（SCHEDULE_QUEUE_LEN_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.12	hirStatisInfSysScheduleQueueLenAvg	スケジュール待ち行列数の平均値（SCHEDULE_QUEUE_LEN_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.13	hirStatisInfSysMessageLenFreq	スケジュールメッセージの発生件数（MESSAGE_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.14	hirStatisInfSysMessageLenMax	スケジュールメッセージ長の最大値（MESSAGE_LEN_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.15	hirStatisInfSysMessageLenMin	スケジュールメッセージ長の最小値（MESSAGE_LEN_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.16	hirStatisInfSysMessageLenAvg	スケジュールメッセージ長の平均値（MESSAGE_LEN_AVG）※3（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.17	hirStatisInfSysServerAbort	サーバの異常終了回数（SERVER_ABORT）※3	INTEGER	read-only	pdstedit（sys）
11.1.18	hirStatisInfSysSystemServerAbort	HiRDB が使用する内部的なサーバの異常終了回数（SYSTEM_SERVER_ABORT）※3	INTEGER	read-only	pdstedit（sys）
11.1.19	hirStatisInfSysProcessCountMax	サーバプロセス生成個数の最大値（PROCESS_COUNT_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.20	hirStatisInfSysProcessCountMin	サーバプロセス生成個数の最小値（PROCESS_COUNT_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.21	hirStatisInfSysProcessCountAvg	サーバプロセス生成個数の平均値（PROCESS_COUNT_AVG）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.22	hirStatisInfSysCommit	コミット回数（COMMIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.23	hirStatisInfSysRollback	ロールバック回数（ROLLBACK）※3	INTEGER	read-only	pdstedit（sys）
11.1.24	hirStatisInfSysLockWaitTimeFreq	排他待ち発生件数（LOCK_WAIT_TIME_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.25	hirStatisInfSysLockWaitTimeMax	排他待ち時間の最大値（LOCK_WAIT_TIME_MAX）※1（単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.26	hirStatisInfSysLockWaitTimeMin	排他待ち時間の最小値（LOCK_WAIT_TIME_MIN）※2（単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.27	hirStatisInfSysLockWaitTimeAvg	排他待ち時間の平均値（LOCK_WAIT_TIME_AVG）※3（単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.28	hirStatisInfSysLockQueueLenFreq	排他待ちになったユーザ数（LOCK_QUEUE_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.29	hirStatisInfSysLockQueueLenMax	排他待ちになったユーザ数の最大値（LOCK_QUEUE_LEN_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.30	hirStatisInfSysLockQueueLenMin	排他待ちになったユーザ数の最小値（LOCK_QUEUE_LEN_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.31	hirStatisInfSysLockQueueLenAvg	排他待ちになったユーザ数の平均値（LOCK_QUEUE_LEN_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.32	hirStatisInfSysDeadlock	デッドロック件数（DEADLOCK）※3	INTEGER	read-only	pdstedit（sys）
11.1.33	hirStatisInfSysUseLockTableFreq	排他資源管理テーブルの使用率が 5%増加する事象の発生件数（USE_LOCK_TABLE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.34	hirStatisInfSysUseLockTableMax	排他資源管理テーブル使用率の最大値（USE_LOCK_TABLE_MAX）※1（単位：%）	INTEGER	read-only	pdstedit（sys）
11.1.35	hirStatisInfSysUseLockTableMin	排他資源管理テーブル使用率の最小値（USE_LOCK_TABLE_MIN）※2（単位：%）	INTEGER	read-only	pdstedit（sys）
11.1.36	hirStatisInfSysUseLockTableAvg	排他資源管理テーブル使用率の平均値（USE_LOCK_TABLE_AVG）※3（単位：%）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.37	hirStatisInfSysSegmentSizeAvg	サーバ用，及び HiRDB が使用する内部的なサーバ用に確保した共用メモリサイズの平均値（SEGMENT_SIZE_AVG）※3 （単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.38	hirStatisInfSysStaticSizeAvg	静的共用メモリ確保サイズの平均値（STATIC_SIZE_AVG）※3 （単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.39	hirStatisInfSysDyanamicSizeAvg	動的共用メモリ確保サイズの平均値（DYNAMIC_SIZE_AVG）※3 （単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.40	hirStatisInfSysSizeForBufferAvg	グローバルバッファ用共用メモリ確保サイズの平均値（SIZE_FOR_BUFFER_AVG）※3 （単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.41	hirStatisInfSysSyncGetIntervalTimeFreq	シンクポイントダンプの取得件数（SYNC_GET_INTERVAL_TIME_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.42	hirStatisInfSysSyncGetIntervalTimeMax	シンクポイントダンプ取得間隔時間の最大値（SYNC_GET_INTERVAL_TIME_MAX）※1 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.43	hirStatisInfSysSyncGetIntervalTimeMin	シンクポイントダンプ取得間隔時間の最小値（SYNC_GET_INTERVAL_TIME_MIN）※2 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.44	hirStatisInfSysSyncGetIntervalTimeAvg	シンクポイントダンプ取得間隔時間の平均値（SYNC_GET_INTERVAL_TIME_AVG）※3 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.45	hirStatisInfSysSyncGetTimeFreq	シンクポイントダンプの取得件数（SYNC_GET_TIME_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.46	hirStatisInfSysSyncGetTimeMax	シンクポイントダンプの取得所要時間の最大値（SYNC_GET_TIME_MAX）※1 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.47	hirStatisInfSysSyncGetTimeMin	シンクポイントダンプ取得時間の最小値（SYNC_GET_TIME_MIN）※2	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
		（単位：ミリ秒）			
11.1.48	hirStatisInfSysSyncGetTimeAvg	シンクポイントダンプ取得時間の平均値（SYNC_GET_TIME_AVG）※3 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.49	hirStatisInfSysLogBufferFull	ログバッファ満杯回数（LOG_BUFFER_FULL）※3	INTEGER	read-only	pdstedit（sys）
11.1.50	hirStatisInfSysLogWaitThread	カレントバッファなしによるログ入出力待ち回数（LOG_WAIT_THREAD）※3	INTEGER	read-only	pdstedit（sys）
11.1.51	hirStatisInfSysLogOutputBlockLenFreq	ログブロック出力回数（LOG_OUTPUT_BLOCK_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.52	hirStatisInfSysLogOutputBlockLenMax	ログ出力ブロック長の最大値（LOG_OUTPUT_BLOCK_LEN_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.53	hirStatisInfSysLogOutputBlockLenMin	ログ出力ブロック長の最小値（LOG_OUTPUT_BLOCK_LEN_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.54	hirStatisInfSysLogOutputBlockLenAvg	ログ出力ブロック長の平均値（LOG_OUTPUT_BLOCK_LEN_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.55	hirStatisInfSysLogNotBusLenFreq	非バス部分のログ出力回数（LOG_NOT_BUS_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.56	hirStatisInfSysLogNotBusLenMax	非バス部分のログブロック長の最大値（LOG_NOT_BUS_LEN_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.57	hirStatisInfSysLogNotBusLenMin	非バス部分のログブロック長の最小値（LOG_NOT_BUS_LEN_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.58	hirStatisInfSysLogNotBusLenAvg	非バス部分のログブロック長の平均値（LOG_NOT_BUS_LEN_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.59	hirStatisInfSysLogWaitBufferForIOFreq	ログ入出力完了待ち件数（LOG_WAIT_BUFFER_FOR_IO_FREQ）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.60	hirStatisInfSysLogWaitBufferForIOMax	ログ入出力待ちバッファ面数の最大値（LOG_WAIT_BUFFER_FOR_IO_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.61	hirStatisInfSysLogWaitBufferForIOMin	ログ入出力待ちバッファ面数の最小値（LOG_WAIT_BUFFER_FOR_IO_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.62	hirStatisInfSysLogWaitBufferForIOAvg	ログ入出力待ちバッファ面数の平均値（LOG_WAIT_BUFFER_FOR_IO_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.63	hirStatisInfSysLogWriteToFile	システムログファイルへの書き込み回数（LOG_WRITE_TO_FILE）※3	INTEGER	read-only	pdstedit（sys）
11.1.64	hirStatisInfSysLogWriteError	ログ書き込みエラー回数（LOG_WRITE_ERROR）※3	INTEGER	read-only	pdstedit（sys）
11.1.65	hirStatisInfSysLogFileSwapTimeFreq	ログファイルスワップ回数（LOG_FILE_SWAP_TIME_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.66	hirStatisInfSysLogFileSwapTimeMax	ログファイルスワップ時間の最大値（LOG_FILE_SWAP_TIME_MAX）※1 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.67	hirStatisInfSysLogFileSwapTimeMin	ログファイルスワップ時間の最小値（LOG_FILE_SWAP_TIME_MIN）※2 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.68	hirStatisInfSysLogFileSwapTimeAvg	ログファイルスワップ時間の平均値（LOG_FILE_SWAP_TIME_AVG）※3 （単位：ミリ秒）	INTEGER	read-only	pdstedit（sys）
11.1.69	hirStatisInfSysLogInputDataFreq	ロールバック時のログ入力件数（LOG_INPUT_DATA_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.70	hirStatisInfSysLogInputDataMax	ロールバック時のログ入力データ長の最大値（LOG_INPUT_DATA_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.71	hirStatisInfSysLogInputDataMin	ロールバック時のログ入力データ長の最小値（LOG_INPUT_DATA_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.72	hirStatisInfSysLogInputDataAvg	ロールバック時のログ入力データ長の平均値（LOG_INPUT_DATA_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.73	hirStatisInfSysLogReadFromFile	ロールバック時のログ読み出し回数（LOG_READ_FROM_FILE）※3	INTEGER	read-only	pdstedit（sys）
11.1.74	hirStatisInfSysLogReadError	ログ読み出しエラー回数（LOG_READ_ERROR）※3	INTEGER	read-only	pdstedit（sys）
11.1.75	hirStatisInfSysDicTblDEFGetReq	表定義情報取得要求回数（DIC_TBL-DEF_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.76	hirStatisInfSysDicTableCacheHit	表定義情報用バッファヒット回数（DIC_TABLE_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.77	hirStatisInfSysDicCachedTblDEFFreq	表定義情報用バッファ中の定義情報数（DIC_CACHED_TBL-DEF_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.78	hirStatisInfSysDicCachedTblDEFMax	表定義情報用バッファ中の定義情報数の最大値（DIC_CACHED_TBL-DEF_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.79	hirStatisInfSysDicCachedTblDEFMin	表定義情報用バッファ中の定義情報数の最小値（DIC_CACHED_TBL-DEF_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.80	hirStatisInfSysDicCachedTblDEFAvg	表定義情報用バッファ中の定義情報数の平均値（DIC_CACHED_TBL-DEF_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.81	hirStatisInfSysDicUseTblDEFSizeFreq	表定義情報用バッファに取得された表定義情報件数（DIC_USE_TBL-DEF_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.82	hirStatisInfSysDicUseTblDEFSizeMax	表定義情報用バッファに取得した 1 表定義情報当たりの表定義情報用バッファ使用領域長の最大値（DIC_USE_TBL-DEF_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.83	hirStatisInfSysDicUseTblDEFSizeMin	表定義情報用バッファに取得した 1 表定義情報当たりの表定義情報用バッファ使用領域長の最小値（DIC_USE_TBL-DEF_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.84	hirStatisInfSysDicUseTblDEFSizeAvg	表定義情報用バッファに取得した 1 表定義情報当たりの表定義情報用バッファ使用領域長の平均値（DIC_USE_TBL-DEF_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.85	hirStatisInfSysDicCacheTblDEFSizeFreq	表定義情報用バッファ上の表定義情報件数（DIC_CACHED_TBL-DEF_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.86	hirStatisInfSysDicCacheTblDEFSizeMax	表定義情報用バッファ使用領域長の最大値（DIC_CACHED_TBL-DEF_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.87	hirStatisInfSysDicCacheTblDEFSizeMin	表定義情報用バッファ使用領域長の最小値（DIC_CACHED_TBL-DEF_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.88	hirStatisInfSysDicCacheTblDEFSizeAvg	表定義情報用バッファ使用領域長の平均値（DIC_CACHED_TBL-DEF_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.89	hirStatisInfSysDicAccessPrivCheck	表アクセス権限情報取得回数（DIC_ACCESS_PRIV_CHECK）※3	INTEGER	read-only	pdstedit（sys）
11.1.90	hirStatisInfSysDicAccessCacheHit	表アクセス権限情報用バッファヒット回数（DIC_ACCESS_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.91	hirStatisInfSysDicConDbaDefGetReq	ユーザ権限情報取得要求数（DIC_CON/DBA_DEF_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.92	hirStatisInfSysDicConDbaCacheHit	ユーザ権限情報用バッファヒット回数（DIC_CON/DBA_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.93	hirStatisInfSysDicConDbaCachedUserFreq	ユーザ権限情報用バッファ使用ユーザ数（DIC_CON/DBA_CACHED_USER_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.94	hirStatisInfSysDicConDbaCachedUserMax	ユーザ権限情報用バッファ使用ユーザ数の最大値（DIC_CON/DBA_CACHED_USER_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.95	hirStatisInfSysDicConDbaCachedUserMin	ユーザ権限情報用バッファ使用ユーザ数の最小値（DIC_CON/DBA_CACHED_USER_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.96	hirStatisInfSysDicConDbaCachedUserAvg	ユーザ権限情報用バッファ使用ユーザ数の平均値（DIC_CON/DBA_CACHED_USER_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.97	hirStatisInfSysDicTransDataLenFreq	表定義情報取得の要求件数（DIC_TRANS_DATA_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.98	hirStatisInfSysDicTransDataLenMax	ディクショナリサーバとの通信長の最大値（DIC_TRANS_DATA_LEN_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.99	hirStatisInfSysDicTransDataLenMin	ディクショナリサーバとの通信長の最小値（DIC_TRANS_DATA_LEN_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.100	hirStatisInfSysDicTransDataLenAvg	ディクショナリサーバとの通信長の平均値（DIC_TRANS_DATA_LEN_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.101	hirStatisInfSysDicTrans	ディクショナリサーバとの通信回数（DIC_TRANS）※3	INTEGER	read-only	pdstedit（sys）
11.1.102	hirStatisInfSysDicViewDefGetReq	ビュー解析情報取得要求回数（DIC_VIEW_DEF_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.103	hirStatisInfSysDicViewDefCacheHit	ビュー解析情報用バッファヒット回数（DIC_VIEW_DEF_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.104	hirStatisInfSysDicViewCachedDef	ビュー解析情報用バッファ中の解析情報数（DIC_VIEW_CACHED_DEF）※3	INTEGER	read-only	pdstedit（sys）
11.1.105	hirStatisInfSysDicUsedViewSizeFreq	ビュー解析情報用バッファ上に取得したビュー解析情報件数（DIC_USED_VIEW_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.106	hirStatisInfSysDicUsedViewSizeMax	ビュー解析情報用バッファに取得した1ビュー当たりのビュー解析情報用バッファ使用領域長の最大値（DIC_USED_VIEW_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.107	hirStatisInfSysDicUsedViewSizeMin	ビュー解析情報用バッファに取得した1ビュー当たりのビュー解析情報用バッファ使用領域長の最小値（DIC_USED_VIEW_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.108	hirStatisInfSysDicUsedViewSizeAvg	ビュー解析情報用バッファに取得した1ビュー当たりのビュー解析情報用バッファ使用領域長の平均値（DIC_USED_VIEW_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.109	hirStatisInfSysDicViewCacheSizeFreq	ビュー解析情報用バッファ上にあるビュー解析情報件数（DIC_VIEW_CACHE_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.110	hirStatisInfSysDicViewCacheSizeMax	使用したビュー解析情報用バッファ長の最大値（DIC_VIEW_CACHE_SIZE_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.111	hirStatisInfSysDicViewCacheSizeMin	使用したビュー解析情報用バッファ長の最小値（DIC_VIEW_CACHE_SIZE_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.112	hirStatisInfSysDicViewCacheSizeAvg	使用したビュー解析情報用バッファ長の平均値（DIC_VIEW_CACHE_SIZE_AVG）※3（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.113	hirStatisInfSysDicCacheMissViewSizeFreq	ビュー解析情報バッファミス回数（DIC_CACHE_MISS_VIEW_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.114	hirStatisInfSysDicCacheMissViewSizeMax	バッファミスとなったビュー解析情報長の最大値（DIC_CACHE_MISS_VIEW_SIZE_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.115	hirStatisInfSysDicCacheMissViewSizeMin	バッファミスとなったビュー解析情報長の最小値（DIC_CACHE_MISS_VIEW_SIZE_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.116	hirStatisInfSysDicCacheMissViewSizeAvg	バッファミスとなったビュー解析情報長の平均値（DIC_CACHE_MISS_VIEW_SIZE_AVG）※3（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.117	hirStatisInfSysSQLOBJGetReq	SQL オブジェクト取得要求回数（SQLOBJ_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.118	hirStatisInfSysSQLOBJCacheHit	SQL オブジェクト用バッファヒット回数（SQLOBJ_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.119	hirStatisInfSysCachedSQLOBJFreq	SQL オブジェクト用バッファ中の SQL オブジェクト数（CACHED_SQLOBJ_FREQ）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.120	hirStatisInfSysCachedSQLOBJMax	SQL オブジェクト用バッファ中の SQL オブジェクト数の最大値（CACHED_SQLOBJ_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.121	hirStatisInfSysCachedSQLOBJMin	SQL オブジェクト用バッファ中の SQL オブジェクト数の最小値（CACHED_SQLOBJ_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.122	hirStatisInfSysCachedSQLOBJAvg	SQL オブジェクト用バッファ中の SQL オブジェクト数の平均値（CACHED_SQLOBJ_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.123	hirStatisInfSysCachedSQLOBJSizeFreq	SQL オブジェクト用バッファ中の SQL オブジェクト数（CACHED_SQLOBJ_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.124	hirStatisInfSysCachedSQLOBJSizeMax	SQL オブジェクト用バッファ中の SQL オブジェクトの合計長の最大値（CACHED_SQLOBJ_SIZE_MAX）※1（単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.125	hirStatisInfSysCachedSQLOBJSizeMin	SQL オブジェクト用バッファ中の SQL オブジェクトの合計長の最小値（CACHED_SQLOBJ_SIZE_MIN）※2（単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.126	hirStatisInfSysCachedSQLOBJSizeAvg	SQL オブジェクト用バッファ中の SQL オブジェクトの合計長の平均値（CACHED_SQLOBJ_SIZE_AVG）※3（単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.127	hirStatisInfSysSwapOutSQLOBJ	SQL オブジェクト用バッファから出された SQL オブジェクトの数（SWAP_OUT_SQLOBJ）※3	INTEGER	read-only	pdstedit（sys）
11.1.128	hirStatisInfSysSQLOBJLenFreq	SQL オブジェクト用バッファ中の SQL オブジェクト数（SQLOBJ_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.129	hirStatisInfSysSQLOBJLenMax	SQL オブジェクト用バッファ中の SQL オブジェクト長の最大値（SQLOBJ_LEN_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.130	hirStatisInfSysSQLOBJLenMin	SQL オブジェクト用バッファ中の SQL オブジェクト長の最小値（SQLOBJ_LEN_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.131	hirStatisInfSysSQLOBJLenAvg	SQL オブジェクト用バッファ中の SQL オブジェクト長の平均値（SQLOBJ_LEN_AVG）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
		（単位：バイト）			
11.1.132	hirStatisInfSysSTROBJGetReq	ストアドプロシジャのオブジェクトの取得要求回数（STROBJ_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.133	hirStatisInfSysSTROBJCacheHit	ストアドプロシジャのオブジェクトのSQL オブジェクト用バッファヒット回数（STROBJ_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.134	hirStatisInfSysCachedSTROBJFreq	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数（CACHED_STROBJ_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.135	hirStatisInfSysCachedSTROBJMax	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数の最大値（CACHED_STROBJ_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.136	hirStatisInfSysCachedSTROBJMin	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数の最小値（CACHED_STROBJ_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.137	hirStatisInfSysCachedSTROBJAvg	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数の平均値（CACHED_STROBJ_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.138	hirStatisInfSysCachedSTROBJSizeFreq	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数（CACHED_STROBJ_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.139	hirStatisInfSysCachedSTROBJSizeMax	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクトの合計長の最大値（CACHED_STROBJ_SIZE_MAX）※1（単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.140	hirStatisInfSysCachedSTROBJSizeMin	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクトの合計長の最小値（CACHED_STROBJ_SIZE_MIN）※2（単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.141	hirStatisInfSysCachedSTROBJSizeAvg	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクトの合計長の平均値（CACHED_STROBJ_SIZE_AVG）※3（単位：キロバイト）	INTEGER	read-only	pdstedit（sys）
11.1.142	hirStatisInfSysSwapOutSTROBJ	SQL オブジェクト用バッファから出されたストアドプロシジャのオブジェクトの数（SWAP_OUT_STROBJ）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.143	hirStatisInfSysSTROBJLenFreq	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数（STROBJ_LEN_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.144	hirStatisInfSysSTROBJLenMax	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト長の最大値（STROBJ_LEN_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.145	hirStatisInfSysSTROBJLenMin	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト長の最小値（STROBJ_LEN_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.146	hirStatisInfSysSTROBJLenAvg	SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト長の平均値（STROBJ_LEN_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.147	hirStatisInfSysSTROBJRecompile	ストアドプロシジャのオブジェクトのリコンパイル回数（STROBJ_RECOMPILE）※3	INTEGER	read-only	pdstedit（sys）
11.1.148	hirStatisInfSysProcessServiceCountMax	サービス実行中のサーバプロセス数の最大値（PROCESS_SERVICE_COUNT_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.149	hirStatisInfSysProcessServiceCountMin	サービス実行中のサーバプロセス数の最小値（PROCESS_SERVICE_COUNT_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.150	hirStatisInfSysProcessServiceCountAvg	サービス実行中のサーバプロセス数の平均値（PROCESS_SERVICE_COUNT_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.151	hirStatisInfSysProcessRequestOver	最大起動プロセス数を超えたサービス要求数（PROCESS_REQUEST_OVER）※3	INTEGER	read-only	pdstedit（sys）
11.1.152	hirStatisInfSysResponseOnOwnUnitFreq	自ユニットサーバへのサービスレスポンス回数（RESPONSE_ON_OWN_UNIT_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.153	hirStatisInfSysResponseOnOwnUnitMax	自ユニットサーバへのサービスレスポンス時間の最大値（RESPONSE_ON_OWN_UNIT_MAX）※1 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.154	hirStatisInfSysResponseOnOwnUnitMin	自ユニットサーバへのサービスレスポンス時間の最小値（RESPONSE_ON_OWN_UNIT_MIN）※2 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.155	hirStatisInfSysResponseOnOwnUnitAvg	自ユニットサーバへのサービスレスポンス時間の平均値（RESPONSE_ON_OWN_UNIT_AVG）※3 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.156	hirStatisInfSysResponseToOtherUnitFreq	他ユニットサーバへのサービスレスポンス回数（RESPONSE_TO_OTHER_UNIT_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.157	hirStatisInfSysResponseToOtherUnitMax	他ユニットサーバへのサービスレスポンス時間の最大値（RESPONSE_TO_OTHER_UNIT_MAX）※1 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.158	hirStatisInfSysResponseToOtherUnitMin	他ユニットサーバへのサービスレスポンス時間の最小値（RESPONSE_TO_OTHER_UNIT_MIN）※2 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.159	hirStatisInfSysResponseToOtherUnitAvg	他ユニットサーバへのサービスレスポンス時間の平均値（RESPONSE_TO_OTHER_UNIT_AVG）※3 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.160	hirStatisInfSysExecTimeOnOwnUnitFreq	自ユニットサーバからの 1 サービス当たりの実行回数（EXEC_TIME_ON_OWN_UNIT_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.161	hirStatisInfSysExecTimeOnOwnUnitMax	自ユニットサーバからの 1 サービス当たりの実行時間の最大値（EXEC_TIME_ON_OWN_UNIT_MAX）※1 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.162	hirStatisInfSysExecTimeOnOwnUnitMin	自ユニットサーバからの 1 サービス当たりの実行時間の最小値（EXEC_TIME_ON_OWN_UNIT_MIN）※2	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
		（単位：マイクロ秒）			
11.1.163	hirStatisInfSysExecTimeOnOwnUnitAvg	自ユニットサーバからの 1 サービス当たりの実行時間の平均値（EXEC_TIME_ON_OWN_UNIT_AVG）※3 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.164	hirStatisInfSysExecTimeFromOtherUnitFreq	他ユニットサーバからの 1 サービス当たりの実行回数（EXEC_TIME_FROM_OTHER_UNIT_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.165	hirStatisInfSysExecTimeFromOtherUnitMax	他ユニットサーバからの 1 サービス当たりの実行時間の最大値（EXEC_TIME_FROM_OTHER_UNIT_MAX）※1 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.166	hirStatisInfSysExecTimeFromOtherUnitMin	他ユニットサーバからの 1 サービス当たりの実行時間の最小値（EXEC_TIME_FROM_OTHER_UNIT_MIN）※2 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.167	hirStatisInfSysExecTimeFromOtherUnitAvg	他ユニットサーバからの 1 サービス当たりの実行時間の平均値（EXEC_TIME_FROM_OTHER_UNIT_AVG）※3 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.168	hirStatisInfSysSendToOwnProcs	自プロセスへの SEND 回数（SEND_TO_OWN_PRCS）※3	INTEGER	read-only	pdstedit（sys）
11.1.169	hirStatisInfSysSendToOtherPrcs	自ユニットの他プロセスへの SEND 回数（SEND_TO_OTHER_PRCS）※3	INTEGER	read-only	pdstedit（sys）
11.1.170	hirStatisInfSysSendToOtherUnit	他ユニットへの SEND 回数（SEND_TO_OTHER_UNIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.171	hirStatisInfSysReceiveFromOwnPrcs	自プロセスからの RECEIVE 回数（RECEIVE_FROM_OWN_PRCS）※3	INTEGER	read-only	pdstedit（sys）
11.1.172	hirStatisInfSysReceiveFromOtherPrcs	自ユニットの他プロセスからの RECEIVE 回数（RECEIVE_FROM_OTHER_PRCS）※3	INTEGER	read-only	pdstedit（sys）
11.1.173	hirStatisInfSysReceiveFromOtherUnit	他ユニットからの RECEIVE 回数（RECEIVE_FROM_OTHER_UNIT）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.174	hirStatisInfSysTypeDEFGetReq	型定義情報取得要求回数（TYPE-DEF_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.175	hirStatisInfSysTypeDEFCacheHit	ユーザ定義型情報用バッファヒット回数（TYPE-DEF_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.176	hirStatisInfSysCachedTypeDEFFreq	ユーザ定義型情報用バッファ中の型定義情報数（CACHED_TYPE-DEF_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.177	hirStatisInfSysCachedTypeDEFMax	ユーザ定義型情報用バッファ中の型定義情報数の最大値（CACHED_TYPE-DEF_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.178	hirStatisInfSysCachedTypeDEFMin	ユーザ定義型情報用バッファ中の型定義情報数の最小値（CACHED_TYPE-DEF_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.179	hirStatisInfSysCachedTypeDEFAvg	ユーザ定義型情報用バッファ中の型定義情報数の平均値（CACHED_TYPE-DEF_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.180	hirStatisInfSysTypeDEFCacheSizeFreq	ユーザ定義型情報用バッファ中の型定義情報数（TYPE-DEF_CACHE_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.181	hirStatisInfSysTypeDEFCacheSizeMax	ユーザ定義型情報用バッファに取得した 1 型定義情報用バッファ使用領域長の最大値（TYPE-DEF_CACHE_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.182	hirStatisInfSysTypeDEFCacheSizeMin	ユーザ定義型情報用バッファに取得した 1 型定義情報用バッファ使用領域長の最小値（TYPE-DEF_CACHE_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.183	hirStatisInfSysTypeDEFCacheSizeAvg	ユーザ定義型情報用バッファに取得した 1 型定義情報用バッファ使用領域長の平均値（TYPE-DEF_CACHE_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.184	hirStatisInfSysTypeDEFCacheTotalSizeFreq	ユーザ定義型情報を取得した件数（TYPE-DEF_CACHE_TOTAL_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.185	hirStatisInfSysTypeDEFCacheTotalSizeMax	ユーザ定義型情報用バッファ総使用領域長の最大値（TYPE-DEF_CACHE_TOTAL_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.186	hirStatisInfSysTypeDEFCacheTotalSizeMin	ユーザ定義型情報用バッファ総使用領域長の最小値（TYPE-DEF_CACHE_TOTAL_SIZE_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.187	hirStatisInfSysTypeDEFCacheTotalSizeAvg	ユーザ定義型情報用バッファ総使用領域長の平均値（TYPE-DEF_CACHE_TOTAL_SIZE_AVG）※3（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.188	hirStatisInfSysTypeDEFCacheAllocSizeFreq	確保したユーザ定義型情報用バッファ個数（TYPE-DEF_CACHE_ALLOC_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.189	hirStatisInfSysTypeDEFCacheAllocSizeMax	確保したユーザ定義型情報用バッファ長の最大値（TYPE-DEF_CACHE_ALLOC_SIZE_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.190	hirStatisInfSysTypeDEFCacheAllocSizeMin	確保したユーザ定義型情報用バッファ長の最小値（TYPE-DEF_CACHE_ALLOC_SIZE_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.191	hirStatisInfSysTypeDEFCacheAllocSizeAvg	確保したユーザ定義型情報用バッファ長の平均値（TYPE-DEF_CACHE_ALLOC_SIZE_AVG）※3（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.192	hirStatisInfSysRtnDEFGetReq	ルーチン定義情報取得要求回数（RTN-DEF_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.193	hirStatisInfSysRtnDEFCacheHit	ルーチン定義情報用バッファヒット回数（RTN-DEF_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.194	hirStatisInfSysCachedRtnDEFFreq	ルーチン定義情報用バッファ中のルーチン定義情報数（CACHED_RTN-DEF_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.195	hirStatisInfSysCachedRtnDEFMax	ルーチン定義情報用バッファ中のルーチン定義情報数の最大値（CACHED_RTN-DEF_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.196	hirStatisInfSysCachedRtnDEFMin	ルーチン定義情報用バッファ中のルーチン定義情報数の最小値（CACHED_RTN-DEF_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.197	hirStatisInfSysCachedRtnDEFAvg	ルーチン定義情報用バッファ中のルーチン定義情報数の平均値（CACHED_RTN-DEF_AVG）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.198	hirStatisInfSysRtnDEFCacheSizeFreq	ルーチン定義情報用バッファ中のルーチン定義情報数（RTN-DEF_CACHE_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.199	hirStatisInfSysRtnDEFCacheSizeMax	ルーチン定義情報用バッファに取得した1ルーチン定義情報当たりのルーチン定義情報用バッファ使用領域長の最大値（RTN-DEF_CACHE_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.200	hirStatisInfSysRtnDEFCacheSizeMin	ルーチン定義情報用バッファに取得した1ルーチン定義情報当たりのルーチン定義情報用バッファ使用領域長の最小値（RTN-DEF_CACHE_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.201	hirStatisInfSysRtnDEFCacheSizeAvg	ルーチン定義情報用バッファに取得した1ルーチン定義情報当たりのルーチン定義情報用バッファ使用領域長の平均値（RTN-DEF_CACHE_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.202	hirStatisInfSysRtnDEFCacheTotalSizeFreq	ルーチン定義情報用バッファ中のルーチン定義情報数（RTN-DEF_CACHE_TOTAL_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.203	hirStatisInfSysRtnDEFCacheTotalSizeMax	ルーチン定義情報用バッファ総使用長の最大値（RTN-DEF_CACHE_TOTAL_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.204	hirStatisInfSysRtnDEFCacheTotalSizeMin	ルーチン定義情報用バッファ総使用長の最小値（RTN-DEF_CACHE_TOTAL_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.205	hirStatisInfSysRtnDEFCacheTotalSizeAvg	ルーチン定義情報用バッファ総使用長の平均値（RTN-DEF_CACHE_TOTAL_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.206	hirStatisInfSysRtnDEFCacheAllocSizeFreq	確保したルーチン定義情報用バッファ個数（RTN-DEF_CACHE_ALLOC_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.207	hirStatisInfSysRtnDEFCacheAllocSizeMax	確保したルーチン定義情報用バッファ長の最大値（RTN-DEF_CACHE_ALLOC_SIZE_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.208	hirStatisInfSysRtnDEFCacheAllocSizeMin	確保したルーチン定義情報用バッファ長の最小値（RTN-DEF_CACHE_ALLOC_SIZE_MIN）※2（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.209	hirStatisInfSysRtnDEFCacheAllocSizeAvg	確保したルーチン定義情報用バッファ長の平均値（RTN-DEF_CACHE_ALLOC_SIZE_AVG）※3（単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.210	hirStatisInfSysPlgRTNGetReq	プラグイン提供関数のルーチン定義取得要求回数（PLG-RTN_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.211	hirStatisInfSysPlgRTNCacheHit	プラグイン提供関数のルーチン定義情報用バッファヒット回数（PLG-RTN_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.212	hirStatisInfSysRegistryGetReq	レジストリ情報取得要求回数（REGISTRY_GET_REQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.213	hirStatisInfSysRegistryCacheHit	レジストリ情報用バッファヒット回数（REGISTRY_CACHE_HIT）※3	INTEGER	read-only	pdstedit（sys）
11.1.214	hirStatisInfSysCachedRegistryDEFFreq	レジストリ情報用バッファ中のレジストリ情報数（CACHED_REGISTRY-DEF_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.215	hirStatisInfSysCachedRegistryDEFMax	レジストリ情報用バッファ中のレジストリ情報数の最大値（CACHED_REGISTRY-DEF_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.216	hirStatisInfSysCachedRegistryDEFMin	レジストリ情報用バッファ中のレジストリ情報数の最小値（CACHED_REGISTRY-DEF_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.217	hirStatisInfSysCachedRegistryDEFAvg	レジストリ情報用バッファ中のレジストリ情報数の平均値（CACHED_REGISTRY-DEF_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.218	hirStatisInfSysRegistryCacheSizeFreq	レジストリ情報用バッファ中のレジストリ情報数（REGISTRY_CACHE_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.219	hirStatisInfSysRegistryCacheSizeMax	1 レジストリ情報当たりのレジストリ情報用バッファ長の最大値（REGISTRY_CACHE_SIZE_MAX）※1（単位：バイト）	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.220	hirStatisInfSysRegistryCacheSizeMin	1 レジストリ情報当たりのレジストリ情報用バッファ長の最小値（REGISTRY_CACHE_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.221	hirStatisInfSysRegistryCacheSizeAvg	1 レジストリ情報当たりのレジストリ情報用バッファ長の平均値（REGISTRY_CACHE_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.222	hirStatisInfSysRegistryCacheTotalSizeFreq	レジストリ情報用バッファ中のレジストリ情報数（REGISTRY_CACHE_TOTAL_SIZE_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.223	hirStatisInfSysRegistryCacheTotalSizeMax	レジストリ情報用バッファ総使用領域長の最大値（REGISTRY_CACHE_TOTAL_SIZE_MAX）※1 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.224	hirStatisInfSysRegistryCacheTotalSizeMin	レジストリ情報用バッファ総使用領域長の最小値（REGISTRY_CACHE_TOTAL_SIZE_MIN）※2 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.225	hirStatisInfSysRegistryCacheTotalSizeAvg	レジストリ情報用バッファ総使用領域長の平均値（REGISTRY_CACHE_TOTAL_SIZE_AVG）※3 （単位：バイト）	INTEGER	read-only	pdstedit（sys）
11.1.226	hirStatisInfSysRegisterdPortsFreq	HiRDB 予約ポートの使用数（REGISTERED_PORTS_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.227	hirStatisInfSysRegisterdPortsMax	HiRDB 予約ポートの使用数の最大値（REGISTERED_PORTS_MAX）※1	INTEGER	read-only	pdstedit（sys）
11.1.228	hirStatisInfSysRegisterdPortsMin	HiRDB 予約ポートの使用数の最小値（REGISTERED_PORTS_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.229	hirStatisInfSysRegisterdPortsAvg	HiRDB 予約ポートの使用数の平均値（REGISTERED_PORTS_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.230	hirStatisInfSysAssignedPortsFreq	HiRDB 予約ポートオーバー時の OS 自動割り当てポート使用数（ASSIGNED_PORTS_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.231	hirStatisInfSysAssignedPortsMax	HiRDB 予約ポートオーバー時の OS 自動割り当てポート使用数の最大値（ASSIGNED_PORTS_MAX）※1	INTEGER	read-only	pdstedit（sys）

ID	オブジェクト	説明	タイプ	権限	データソース
11.1.232	hirStatisInfSysAssignedPortsMin	HiRDB 予約ポートオーバー時の OS 自動割り当てポート使用数の最小値（ASSIGNED_PORTS_MIN）※2	INTEGER	read-only	pdstedit（sys）
11.1.233	hirStatisInfSysAssignedPortsAvg	HiRDB 予約ポートオーバー時の OS 自動割り当てポート使用数の平均値（ASSIGNED_PORTS_AVG）※3	INTEGER	read-only	pdstedit（sys）
11.1.234	hirStatisInfSysDirectoryUserCheckTimeFreq	ディレクトリ登録のユーザ認証要求回数（DIRECTORY_USER_CHECK_TIME_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.235	hirStatisInfSysDirectoryUserCheckTimeMax	ディレクトリ登録のユーザ認証時間の最大値（DIRECTORY_USER_CHECK_TIME_MAX）※1 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.236	hirStatisInfSysDirectoryUserCheckTimeMin	ディレクトリ登録のユーザ認証時間の最小値（DIRECTORY_USER_CHECK_TIME_MIN）※2 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.237	hirStatisInfSysDirectoryUserCheckTimeAvg	ディレクトリ登録のユーザ認証時間の平均値（DIRECTORY_USER_CHECK_TIME_AVG）※3 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.238	hirStatisInfSysGroupCheckTimeFreq	グループ判定実行要求回数（GROUP_CHECK_TIME_FREQ）※3	INTEGER	read-only	pdstedit（sys）
11.1.239	hirStatisInfSysGroupCheckTimeMax	グループ判定実行時間の最大値（GROUP_CHECK_TIME_MAX）※1 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.240	hirStatisInfSysGroupCheckTimeMin	グループ判定実行時間の最小値（GROUP_CHECK_TIME_MIN）※2 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.241	hirStatisInfSysGroupCheckTimeAvg	グループ判定実行時間の平均値（GROUP_CHECK_TIME_AVG）※3 （単位：マイクロ秒）	INTEGER	read-only	pdstedit（sys）
11.1.242	hirStatisInfSysDummy	参照不可	INTEGER	read-only	MIB コマンド

（凡例）－：該当しません。

注　pdstedit の後ろにある（　）は，統計解析ユティリティの編集項目を意味しています。

注※1　1 サーバで複数の統計情報が取得された場合は最大値を取得します。

注※2　1 サーバで複数の統計情報が取得された場合は最小値を取得します。

注※3　1 サーバで複数の統計情報が取得された場合は平均値を取得します。

27.12　ディスク使用量

MIB パフォーマンス情報監視機能を使用した場合のディスク使用量のうち，JP1/ESA に結果を返す領域を次の表に示します。そのほかの領域のディスク使用量については，表「そのほかの領域のディスク使用量（単位：バイト）」に示します。

表 27–10　MIB テーブル用のディスク使用量（単位：バイト）

用途	ディレクトリ/ファイル名	容量
サーバ稼働状態テーブル用	インストールディレクトリ¥sample¥snmptmp	178×(ユニット数+サーバ数)
作業表用 HiRDB ファイルシステム領域テーブル用	インストールディレクトリ¥sample¥snmptmp3	2212×作業表用 HiRDB ファイルシステム領域数
RD エリアテーブル用	インストールディレクトリ¥sample¥snmptmp4	215×RD エリア数
RD エリア詳細テーブル用	インストールディレクトリ¥sample¥snmptmp5	502×RD エリア数
グローバルバッファテーブル用	インストールディレクトリ¥sample¥snmptmp6	351×グローバルバッファ数
RD エリア-HiRDB ファイルシステム領域テーブル用	インストールディレクトリ¥sample¥snmptmp9	345×RD エリア数
SYS 統計情報テーブル用	インストールディレクトリ¥sample¥snmptmp11	2719×サーバ数

注　容量は最大値を示しています。

表 27–11　そのほかの領域のディスク使用量（単位：バイト）

ディレクトリ/ファイル名		使用目的	容量
%PDDIR%¥spool¥pdmibwork ディレクトリ	pdmibwork¥sys_DAT pdmibwork¥fil_DAT	pdstedit 実行結果※1	マニュアル「HiRDB システム導入・設計ガイド」の「統計解析ユティリティ（pdstedit）実行時のファイルの容量」参照
	pdmibwork¥new_parameter	最終ログ情報※1	32
	pdmibwork¥log¥stlog01 pdmibwork¥log¥stlog02	統計ログ出力機能コマンドログ	stlog01，stlog02 各最大 1M
	pdmibwork¥stjtmp¥pdstj_tmp	統計ログ一時ファイル※1	最大 pdstj1，pdstj2 の合計サイズ
	マネジャホストの pdmibwork¥sys_DAT_copy pdmibwork¥fil_DAT_copy	リモートコピー用 pdstedit 実行結果※2	マニュアル「HiRDB システム導入・設計ガイド」の「統計解析ユティリティ（pdstedit）実行時のファイルの容量」参照

ディレクトリ/ファイル名		使用目的	容量
	マネジャホストの pdmibwork¥new_parameter_copy	リモートコピー用最終ログ情報※2	32
	pdmibwork¥pdmibstm_sys.dat pdmibwork¥pdmibstm_fil.dat	統計情報管理データ	1K ＋（32×ユニット数）
	pdmibwork¥mibstrerr	統計情報出力機能の最終標準エラー出力	最大 1K
	pdmibwork¥pdmibidx	インデクス値管理データ	最大：47×ホスト数+19×ユニット数＋ 23×サーバ数+45×RD エリア数＋ 31×グローバルバッファ数+46×HiRDB ファイルシステム領域数

注※1　pdstedit コマンドを実行して情報収集後，削除します。

注※2　リモートコピー後，削除します。

28 PRF トレース機能

この章では，HiRDB の PRF トレース機能の環境設定，及び運用方法について説明します。

28.1　PRF トレース機能とは

PRF トレース機能とは，HiRDB の一連の処理に対して出力される，トラブルシュート用のトレース情報を取得する機能です。PRF トレース機能では，トラブルの原因となっているプロセスを特定しやすくするため，ディスクの入出力，プロセス間通信（送信，及び受信）など，性能ネックとなりやすい処理のプロセスについてトレースを取得します。また，複数のマシン間，複数のプロセス間の処理についてもトレースを取得できます。

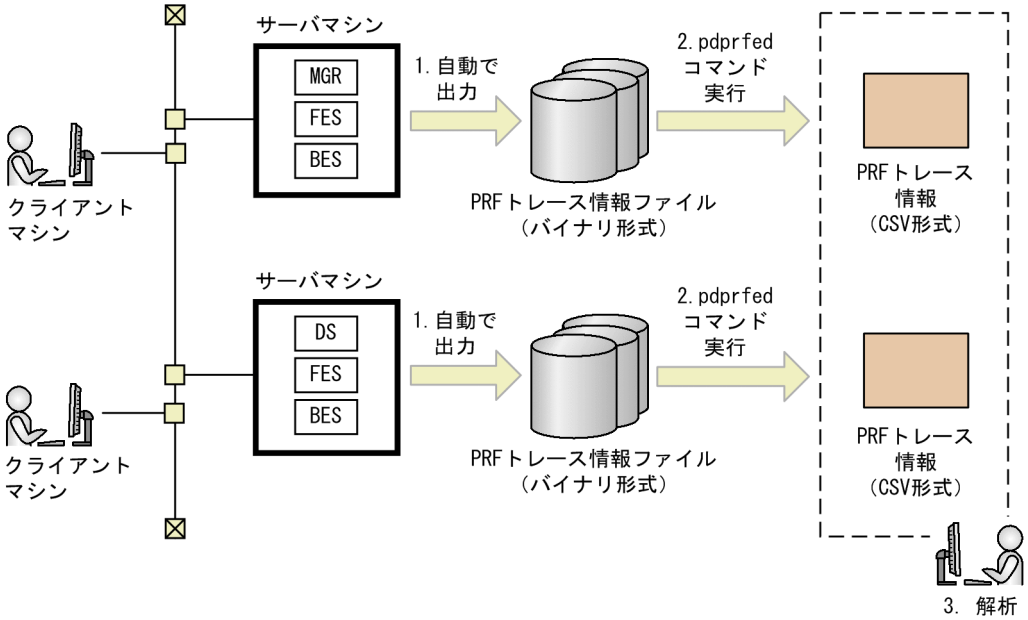
従来のトラブルシュートで使用していた統計情報は，HiRDB 全体の稼働状況を確認するためのものです。一方，PRF トレース情報ではコネクションごとの稼働状況を確認できるため，サポートサービスに情報を送付することで，遅延要因の切り分けや原因特定の効率が向上し，問題の早期解決に役立ちます。また，出力されたトレース情報を CSV 形式のファイルに編集出力すれば，CSV 形式を扱えるツール（Excel など）で柔軟に解析することもできます。

注意事項

PRF トレース情報を取得する際は，処理の同時実行性を優先しているため，排他を掛けていません。そのため，一部のトレース情報が出力されなかったり，不正な情報が出力されたりすることがあります。PRF トレース情報を参照する際は，「注意事項」を参照し，正常なトレース情報であることを確認してください。

PRF トレース機能の概要を次の図に示します。

図 28–1　PRF トレース機能の概要（HiRDB/パラレルサーバの場合）



〔説明〕

1. HiRDB の処理中，取得ポイントで PRF トレース情報が取得されます。この PRF トレース情報は，10 秒間隔で PRF トレース情報ファイルに自動的に出力されます。このときの PRF トレース情報はバイナリ形式です。
2. pdprfed コマンドを実行すると，PRF トレース情報は CSV 形式に編集出力されます。
3. CSV 形式のファイルを扱えるツールを使用して，トラブルの原因を解析します。

28.1.1　PRF トレース機能の取得対象となるプロセス

PRF トレース機能は，HiRDB サーバのプロセスを取得対象としています。HiRDB クライアントのプロセスは取得しません。

28.1.2　PRF トレース情報を取得するための設定

PRF トレース情報は，システム共通定義の pd_prf_trace オペランドに Y を指定するか，又は省略することで取得できます。

なお，pd_prf_trace オペランドに N を指定すると，PRF トレース機能が無効となるため，PRF トレース情報は取得されません。障害発生時の原因調査が困難になるため，N は指定しないでください。

(1)　PRF トレース機能の取得レベル

PRF トレース機能は，最小レベル（省略値），標準レベル，詳細レベル，保守レベル，抑止レベルから取得レベルを選択することで，PRF トレース情報ファイルに出力する PRF トレース情報の出力量を調整できます。PRF トレース取得レベルは，システム定義の pd_prf_level オペランドで指定します。また，HiRDB 稼働中に取得レベルを変更する場合は，pdprflevel コマンドを使用します。

取得レベル名称に対応する取得レベルを次の表に示します。取得レベルの詳細は，マニュアル「HiRDB システム定義」の pd_prf_level オペランドを参照してください。

取得レベル名称	取得レベル
最小レベル（省略値）	00000004
標準レベル	00000004，00000010
詳細レベル	00000004，00000010，00000040
保守レベル	00000004，00000010，00000040，00000100
抑止レベル	取得しない

取得レベルに対応する PRF トレース情報の取得ポイントについては，表「PRF トレース情報の主な取得ポイント一覧（SQL 実行時）」，表「PRF トレース情報の主な取得ポイント一覧（ユティリティ実行時）」を参照してください。

28.1.3　PRF トレース情報ファイルの運用

PRF トレース情報ファイルは，次のディレクトリ下にシステム定義の pd_prf_file_count オペランドで指定した世代数分（省略値 15）作成されます。

%PDDIR%¥spool¥pdprf¥ファイル名

ファイル名称は，「pdprf_世代番号」で作成されます。例えば，世代番号が 1 のファイルは，「pdprf_001」となります。PRF トレース情報をファイルに出力するときにシステム定義の pd_prf_file_size オペランドで指定したファイルサイズ（省略値 51200 キロバイト）に収まらない場合，次のファイルにスワップします。世代の最後のファイルが満杯になった場合は，最初のファイルにスワップし，ファイルを上書きします。

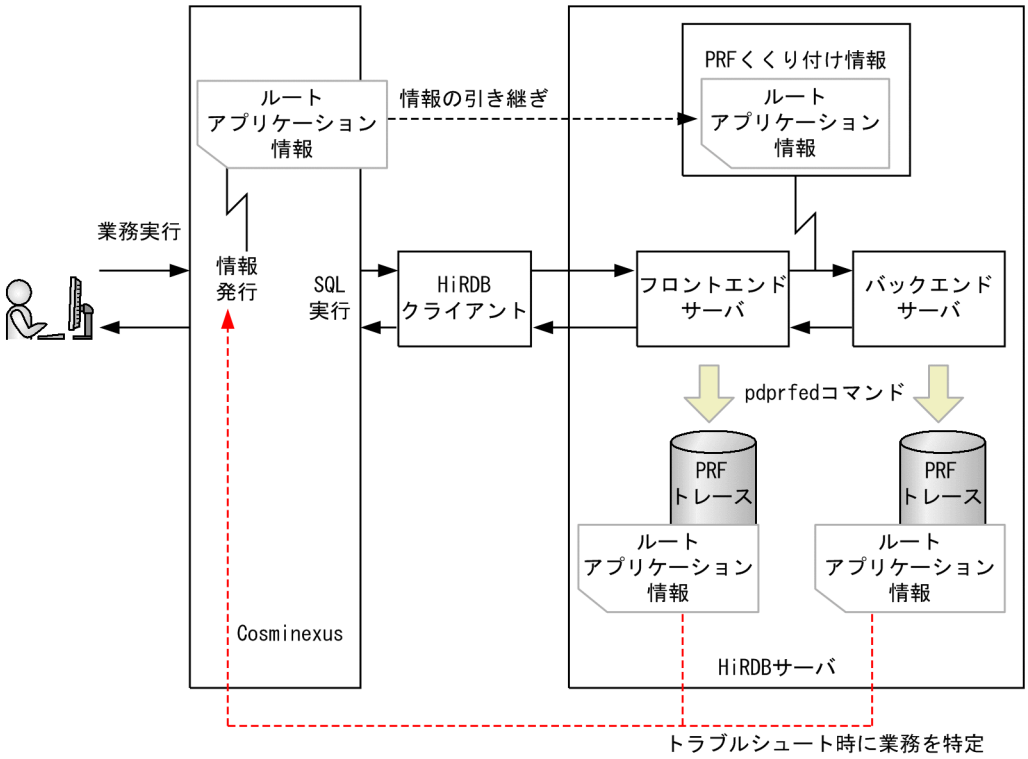
PRF トレースの出力ディレクトリは，pd_prf_output_directory オペランドで変更できます。

28.1.4　Cosminexus と連携した PRF トレース情報の取得

PRF トレース機能で Cosminexus と連携すると，トラブルシュート時に HiRDB の PRF トレースから Cosminexus のルートアプリケーション情報が得られます。そのため，問題が発生している業務を容易に特定できます。

Cosminexus と連携した PRF トレース情報の取得例を次の図に示します。

図 28–2　Cosminexus と連携した PRF トレース情報の取得例



(1)　前提製品

PRF トレース機能で Cosminexus と連携するために必要なプログラムプロダクトのバージョンを次に示します。

- Cosminexus：07-50 以降
- HiRDB クライアント：09-50 以降

(2)　ルートアプリケーション情報の扱い

Cosminexus から HiRDB クライアントにルートアプリケーション情報が引き継がれた場合，SQL の実行時に HiRDB クライアントから HiRDB サーバにルートアプリケーション情報を引き継ぎます。

HiRDB システム内では，ルートアプリケーション情報を PRF くくり付け情報としてトランザクション開始時（又はトランザクションブランチ開始時）に引き継ぎます。そのため，トランザクション開始以降に発生した PRF トレースにルートアプリケーション情報が出力されます。また，トランザクションの途中でルートアプリケーション情報が変化した場合でも，PRF トレースにはトランザクション開始時のルートアプリケーション情報が出力されることがあります。

28.2　PRF トレース情報の参照手順

PRF トレース情報を参照する手順を次に示します。

ポイント

サポートサービスをご利用になる場合は，バイナリ形式の PRF トレース情報ファイルを送付することになるため，この手順は必要ありません。

1. OS のコピーコマンドで PRF トレース情報ファイルを作業ディレクトリにコピーします。
   PRF トレース情報ファイルの使用中及び参照中に，情報が上書きされるおそれがあるため，作業ディレクトリにコピーして退避してください。
2. pdprfed コマンドを実行します。
   PRF トレース情報が CSV 形式に編集出力されます。
3. CSV 形式を扱えるツール（Excel など）で，PRF トレース情報を参照します。

注意事項

PRF トレース情報は，10 秒間隔で PRF トレース情報ファイルに出力されますが，バッファを経由してから出力されるため，最新のトレースが出力されるまでに時間が掛かります。そのため，参照したい SQL を実行してから，1 分程度経過した後に pdprfed コマンドを実行してください。

28.2.1　PRF トレース情報の取得内容

PRF トレース情報の取得内容を次に示します。

表 28–1　PRF トレース情報の取得内容

項番	項目	取得内容
1	IP アドレス	トレースを取得したサーバマシンの IP アドレス
2	プロセス ID	トレースを取得したプロセス ID
3	サーバ名，プロセス名	トレースを取得したサーバ名（ユーザサーバ名）又はプロセス名 プロセス名の詳細は，表「PRF トレース情報の主な取得ポイント一覧（SQL 実行時）」を参照してください。
4	スレッド ID	トレースを取得したプロセス内のスレッド ID
5	トレース取得時刻	トレースを取得した時刻（マイクロ秒※1 まで）
6	PRF トレース通番	1 プロセス内での PRF トレースの通番
7	イベント ID	イベントの内容を示す ID

項番	項目	取得内容
		イベント ID の詳細は，表「PRF トレース情報の主な取得ポイント一覧（SQL 実行時）」を参照してください。
8	クライアントの IP アドレス	クライアントの IP アドレス※2
9	クライアントのプロセス ID	クライアントのプロセス ID※2 Type4 JDBC ドライバから接続している場合は，0 が出力されます。
10	クライアントのポート番号	クライアントの受信ポート番号※2
11	クライアントのコネクト通番	クライアントのコネクト通番※2
12	PRF くくり付け ID	HiRDB 内で一意となる ID※3 次の形式で出力されます。 **aaaabbbbbbbb** aaaa：接続した FES のあるユニット名 bbbbbbbb：コネクト通番（16 進数）
13	PRF ユーザ情報	Cosminexus と連携した場合： ルートアプリケーション情報※3 次の形式で出力されます。 **aa....aa/bb....bb/cc....cc** aa....aa：識別子 bb....bb：プロセス ID cc....cc：通信番号 上記以外の場合： 保守情報※3
14	固有データ	保守情報

注※1

時刻の精度は，プラットフォームに依存します。

注※2

クライアントのアプリケーション情報がある場合に取得されます。ただし，クライアントの情報がない場合でも，取得されていないことを示す 0 や 0.0.0.0 を出力することがあります。

注※3

PRF くくり付け情報がある場合に取得されます。

28.3 注意事項

PRF トレース機能を使用する際の注意事項を次に示します。

- PRF トレース機能は，処理の同時実行性を優先しているため，排他を掛けていません。そのため，一部のトレース情報が出力されなかったり，不正な情報が出力されたりすることがあります。

  PRF トレース情報を参照する際は，トレース情報を複数回取得し，正常なトレース情報（プロセスごとの PRF トレース通番が連番で出力されている情報）であることを確認してください。
- PRF トレース情報の取得は，トラブルシュート情報取得プロセス（pdprfd プロセス）が行っています。HiRDB 稼働中に pdprfd プロセスが異常終了した場合，KFPS05901-W メッセージを出力して pdprfd プロセスは終了しますが，HiRDB は稼働を継続します。このとき PRF トレース情報は取得されていません。PRF トレース情報の取得を再開する場合は，HiRDB を終了後，再度 HiRDB を開始してください。これによって，pdprfd プロセスは再起動します。
- pdprfd プロセスが稼働しているかどうかは，次のどちらかの方法で確認できます。

  ・OS のタスクマネージャなどで，pdprfd プロセスの状態を確認する

  ・pdprflevel コマンドを実行し，エラーにならないことを確認する

  なお，pdls -d prc コマンドでは，pdprfd プロセスの稼働状態は確認できません。
- 複数のマシン間の PRF トレース情報を参照する場合，あらかじめマシン間の時刻を一致させることを推奨します。時刻が一致していない場合は，Excel などを利用して時刻補正をする必要があります。
- PRF トレース情報ファイルに出力される PRF トレース情報は，バイナリ形式です。そのため，ftp などでファイル転送を行う場合は，バイナリ形式に対応した転送モードを使用してください。
- PRF トレース情報ファイルの出力先ディレクトリには，PRF トレース情報ファイル以外のファイルを格納しないでください。例えば，PRF トレース情報ファイルのディレクトリ下で圧縮などの作業を行うと，OS コマンドなどが一時ファイルを作成することがあります。このようなことを避けるため，PRF トレース情報ファイルを参照したり，編集したりするときは，作業ディレクトリにコピーしてから行ってください。

28.4　PRF トレース情報の詳細と取得ポイント

PRF トレース情報は，SQL 実行時やユティリティ実行時などの HiRDB 処理中に，取得ポイントのタイミングで取得されます。PRF トレース情報の詳細と取得ポイントを，SQL 実行時（CONNECT～COMMIT），及びユティリティ実行時（pdload，pdrorg）に分けて説明します。

28.4.1　SQL 実行時の PRF トレース情報の詳細と取得ポイント

SQL 実行時（CONNECT～COMMIT）の PRF トレース情報のイベント名とそのイベントに関連する情報を次の表に，PRF トレース情報の主な取得ポイントを図「PRF トレースの取得ポイントの概念図（HiRDB/シングルサーバの場合）」と図「PRF トレースの取得ポイントの概念図（HiRDB/パラレルサーバの場合）」に示します。

なお，PRF トレース情報には次に示す情報以外に，HiRDB の内部情報を出力することがあります。

表 28–2　PRF トレース情報の主な取得ポイント一覧（SQL 実行時）

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	取得レベル
1	FES ノード要求受け付け	0x11028F00	1	pdrdm	1	00000004
2	FES ノード要求応答	0x11018F00	1	pdrdm	1	00000004
3	CONNECT 要求受け付け	0x11028A00	1	pdrdm，pdndm	1	00000004
4	CONNECT 要求応答	0x11018A00	2	pdrdm，pdndm	1	00000004
5	サーバ割り当て要求受け付け	0x11029100	1	pdscdd	1	00000004
6	サーバ割り当て開始	0x11019100	2	pdscdd	1	00000004
7	サーバ割り当て終了	0x11029101	2	pdfes，pddic，pdbes，pdsds	1	00000004
8	CONNECT 要求受け付け（FES，SDS）	0x11022001	1	pdfes，pdsds	1	00000004
9	CONNECT 要求応答（FES，SDS）	0x11012001	2	pdfes，pdsds	1	00000004
10	バッファ出力待ち開始	0x10134003	22	pdbes，pdsds	4b	00000004
11	バッファ出力待ち完了	0x10134004	22	pdbes，pdsds	4b	00000004
12	バッファ排他待ち開始	0x13134001	22	pdbes，pdsds	4a	00000004
13	バッファ排他待ち完了	0x13134002	22	pdbes，pdsds	4a	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	取得レベル
14	バッファフラッシュ待ち開始	0x13134003	22	pdbes，pdsds	4d	00000004
15	バッファフラッシュ待ち完了	0x13134004	22	pdbes，pdsds	4d	00000004
16	バッファ入力待ち開始	0x13134005	22	pdbes，pdsds	4c	00000004
17	バッファ入力待ち完了	0x13134006	22	pdbes，pdsds	4c	00000004
18	排他解除待ち開始	0x102F8700	22	pdfes，pddic，pdbes，pdsds	8	00000004
19	排他解除待ち終了	0x102F8701	22	pdfes，pddic，pdbes，pdsds	8	00000004
20	カレントログバッファ満杯	0x102F8601	22，23，24，25	pdfes，pddic，pdbes，pdsds，pdlogd，pdlogswd，pdtrnrvd，pdndmd，pdrdmd，pdrsvre	8	00000004
21	システムログの出力先変更	0x10118600	22，23，24，25	pdfes，pddic，pdbes，pdsds，pdtrnrvd，pdndmd，pdrdmd，pdrsvre	8	00000004
22	システムログバッファ全面満杯	0x102F8602	22，23，24，25	pdfes，pddic，pdbes，pdsds，pdlogd，pdlogswd，pdtrnrvd，pdndmd，pdrdmd，pdrsvre	8	00000004
23	SQL 実行要求受け付け	0x10022002	3，13	pdfes，pdsds	1	00000010
24	サーバ割り当て要求	0x11019101	4，8，14	pdfes，pdbes，pddic	1	00000010
25	ディクショナリ情報取得要求	0x10013001	4	pdfes	1	00000010
26	ディクショナリ情報取得要求受け付け	0x10023002	5	pddic	1	00000010

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	取得レベル
27	ディクショナリ情報送信	0x10013003	6	pddic	1	00000010
28	ディクショナリ情報受信	0x10023004	7	pdfes	1	00000010
29	SQL 実行要求	0x13012503	8	pdfes，pdbes	1	00000010
30	SQL 実行要求受け付け（BES）	0x13022003	9	pdbes	1	00000010
31	SQL 実行結果送信	0x13012504	10	pdbes	1	00000010
32	SQL 実行結果受信	0x13022504	11	pdfes，pdbes	1	00000010
33	SQL 実行要求応答	0x10012002	12，18	pdfes，pdsds	1	00000010
34	COMMIT 要求送信	0x10019600，0x10029601	14	pdfes，pdbes，pddic，pdsds	1	00000010
35	COMMIT 要求受信	0x10029600，0x10029602，0x10029603	15	pdbes	1	00000010
36	COMMIT 結果送信	0x10019601	16	pdbes	1	00000010
37	COMMIT 結果受信	0x10029601	17	pdfes	1	00000010
38	トランザクション開始	0x11009600	20，21，22	pdfes，pdbes，pddic，pdsds	1	00000040
39	システムログバッファリング完了	0x102F8600	22，23，24，25	pdfes，pddic，pdbes，pdsds，pdlogd，pdlogswd，pdtrnrvd，pdndmd，pdrdmd，pdrsvre	8	00000040
40	システムログ出力	0x10008600	22，23，24，25	pdfes，pddic，pdbes，pdsds，pdlogd，pdlogswd，pdtrnrvd，pdndmd，pdrdmd，pdrsvre	2c	00000040
41	シンクポイント時のデータベース書き込み処理代行開始	0x13114001	22	pdbes，pdsds	2a	00000040

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	取得レベル
42	シンクポイント時のデータベース書き込み処理代行完了	0x13114002	22	pdbes，pdsds	2a	00000040
43	バッファ出力開始	0x10114001	22	pdbes，pdsds	2a	00000040
44	バッファ出力完了	0x10114002	22	pdbes，pdsds	2a	00000040
45	バッファ入力開始	0x13114003	22	pdbes，pdsds	2a	00000040
46	バッファ入力完了	0x13114004	22	pdbes，pdsds	2a	00000040
47	作業表用ファイルの出力開始	0x13114101	22	pdbes，pdsds	2b	00000040
48	作業表用ファイルの出力完了	0x13114102	22	pdbes，pdsds	2b	00000040
49	作業表用ファイルの入力開始	0x13114103	22	pdbes，pdsds	2b	00000040
50	作業表用ファイルの入力完了	0x13114104	22	pdbes，pdsds	2b	00000040

注※1

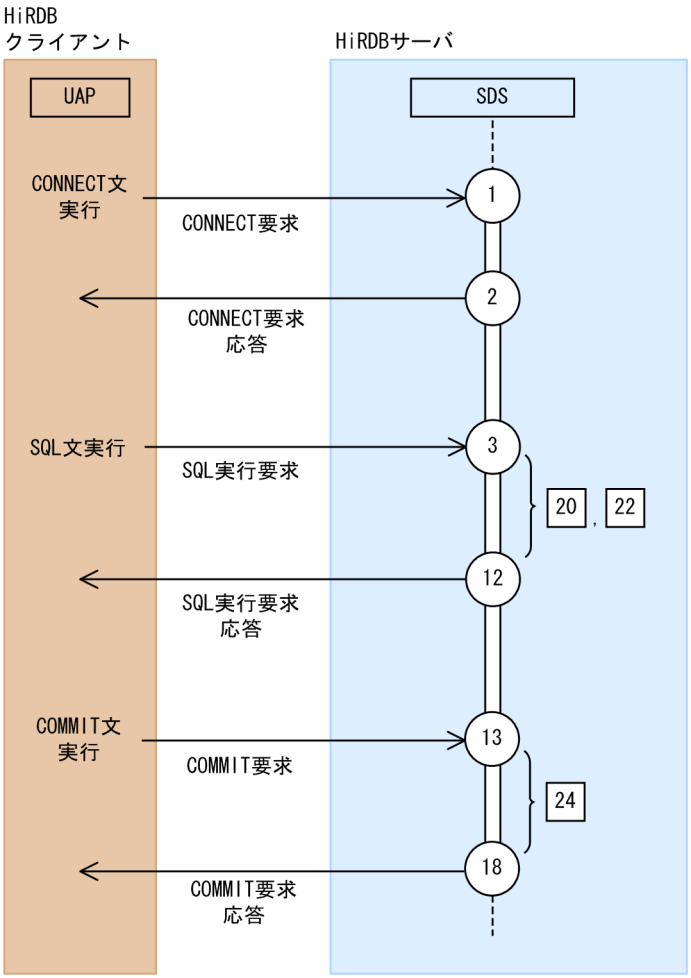
図「PRF トレースの取得ポイントの概念図（HiRDB/シングルサーバの場合）」，及び図「PRF トレースの取得ポイントの概念図（HiRDB/パラレルサーバの場合）」中の番号と対応しています。図中の対応する番号のポイントで取得されることを意味します。

注※2

イベント種別の内容を次に示します。

イベント種別	内容
1	通信
2a	ディスク入出力：グローバルバッファ
2b	ディスク入出力：作業表
2c	ディスク入出力：システムログ
4a	グローバルバッファ排他情報：バッファ排他待ち
4b	グローバルバッファ排他情報：バッファ出力待ち
4c	グローバルバッファ排他情報：バッファ入力待ち
4d	グローバルバッファ排他情報：バッファフラッシュ待ち
8	その他

図 28–3　PRF トレースの取得ポイントの概念図（HiRDB/シングルサーバの場合）

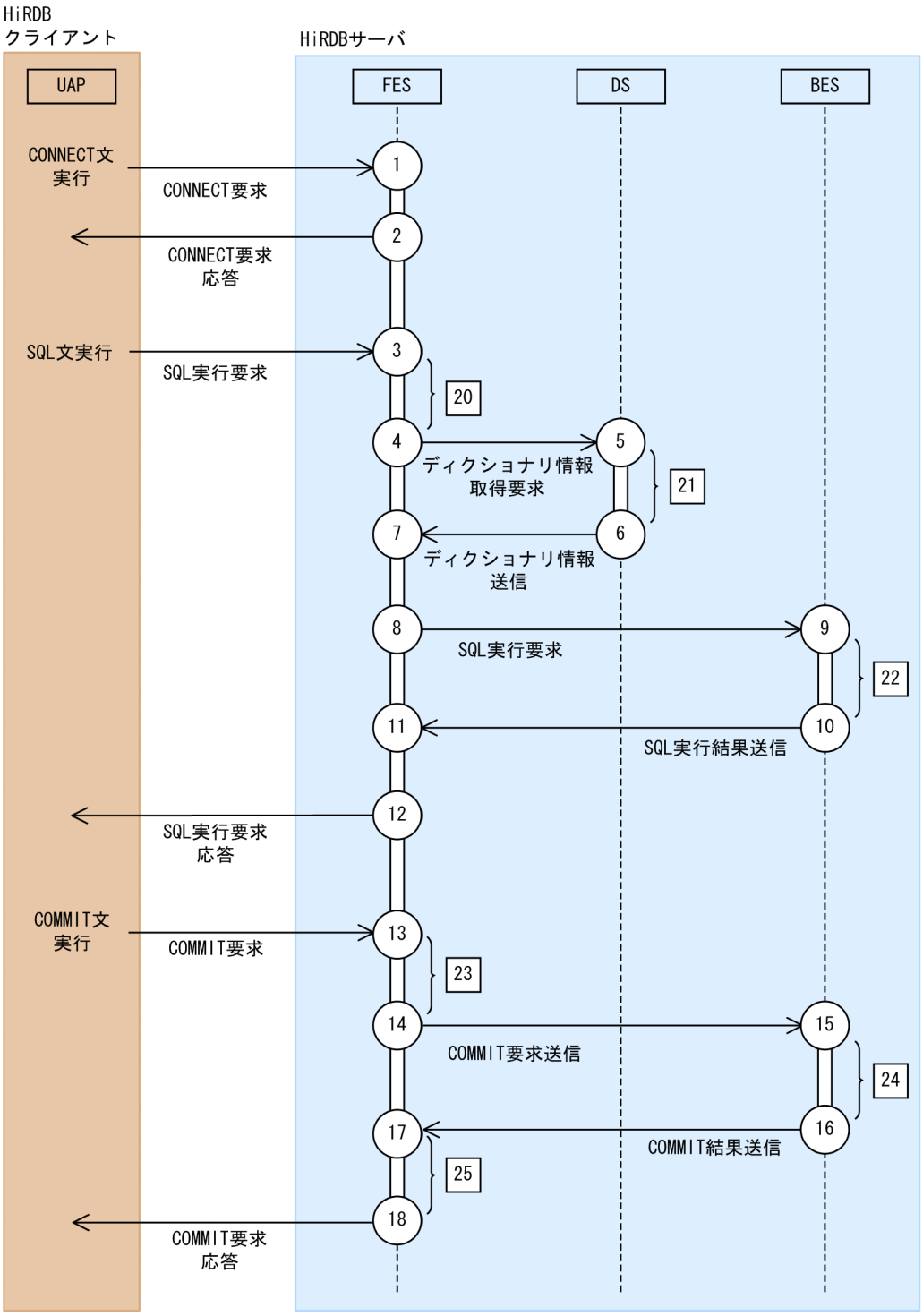


（凡例）

○：PRF トレースの取得ポイント

□：PRF トレースの取得ポイント間の処理内容

図 28–4　PRF トレースの取得ポイントの概念図（HiRDB/パラレルサーバの場合）



（凡例）

○：PRF トレースの取得ポイント

□：PRF トレースの取得ポイント間の処理内容

28.4.2　ユティリティ実行時の PRF トレース情報の詳細と取得ポイント

ユティリティ実行時（pdload，pdrorg）の PRF トレース情報のイベント名とそのイベントに関連する情報を次の表に，PRF トレース情報の主な取得ポイントを図「PRF トレースの取得ポイントの概念図（pdload）」，図「PRF トレースの取得ポイントの概念図（pdrorg）」に示します。

なお，PRF トレース情報には次に示す情報以外に，HiRDB の内部情報を出力することがあります。

表 28–3　PRF トレース情報の主な取得ポイント一覧（ユティリティ実行時）

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	取得レベル
1	行データ送信（ロード）	0x31017000	1	pdloadm，pdrorgm	1	00000004
2	行データ送信（アンロード）	0x32017000	5	pdbes，pdsds	1	00000004
3	応答データ返信（ロード）	0x31017001	3	pdbes，pdsds	1	00000004
4	応答データ返信（アンロード）	0x32017001	7	pdrorgm	1	00000004
5	行データ受信（ロード）	0x31027000	2	pdbes，pdsds	1	00000004
6	行データ受信（アンロード）	0x32027000	6	pdrorgm	1	00000004
7	応答データ受信（ロード）	0x31027001	4	pdloadm，pdrorgm	1	00000004
8	応答データ受信（アンロード）	0x32027001	8	pdbes，pdsds	1	00000004

注※1

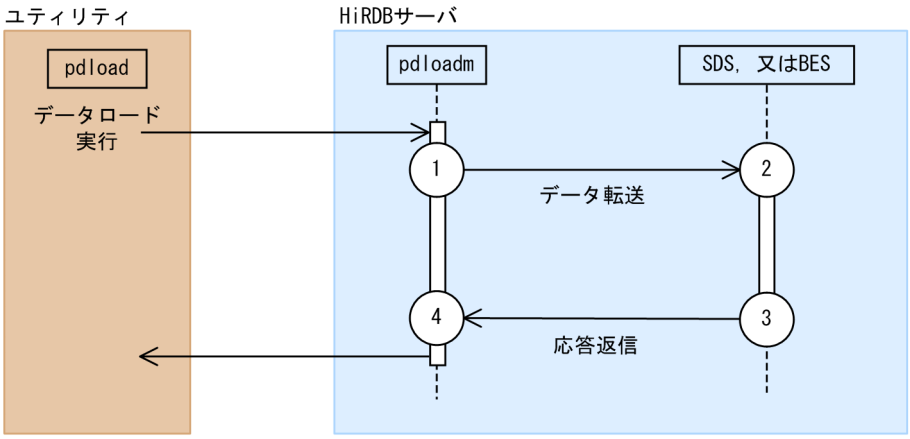
図「PRF トレースの取得ポイントの概念図（pdload）」及び図「PRF トレースの取得ポイントの概念図（pdrorg）」中の番号と対応しています。図中の対応する番号のポイントで取得されることを意味します。

注※2

イベント種別の内容を次に示します。

イベント種別	内容
1	通信

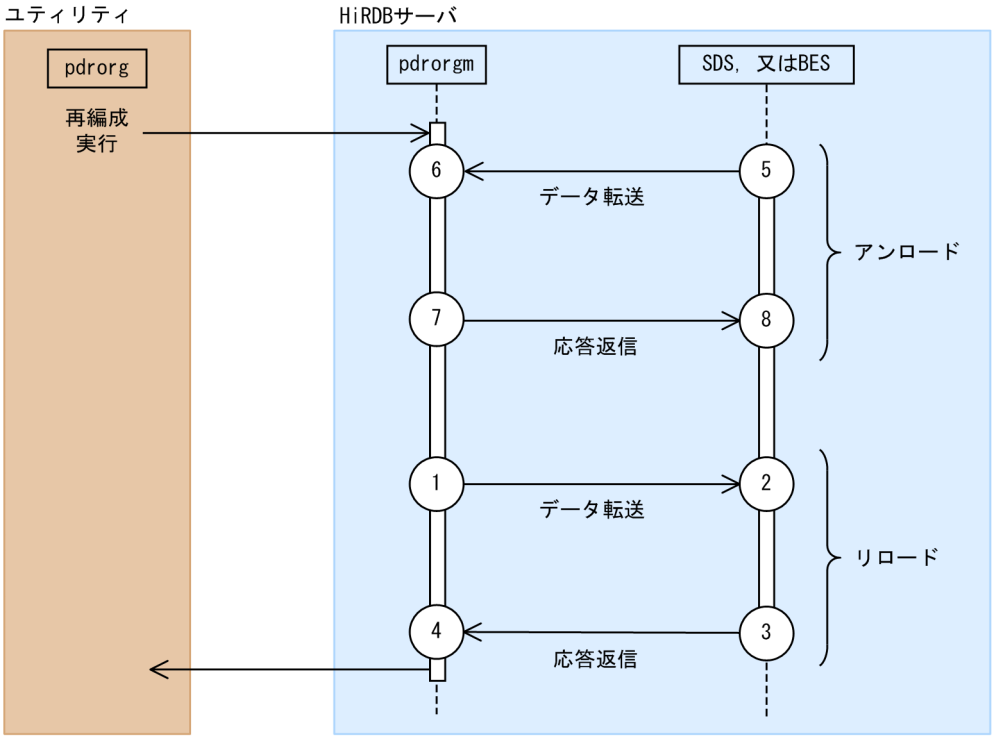
図 28–5　PRF トレースの取得ポイントの概念図（pdload）



（凡例）

○：PRF トレースの取得ポイント

図 28–6　PRF トレースの取得ポイントの概念図（pdrorg）



（凡例）

○：PRF トレースの取得ポイント

29

HiRDB 接続時のパスワード秘匿化機能の運用

この章では，HiRDB システムのセキュリティ強化を目的とした HiRDB 接続時のパスワード秘匿化機能の運用方法について説明します。

29.1　HiRDB 接続時のパスワード秘匿化機能の概要

HiRDB 接続時に，クライアントとサーバ間のネットワーク上で送受信するパスワードを秘匿化します。この機能によって，悪意のあるユーザがパケットスニファリングなどで不正にパケットを取得し，不正な電文の作成・送信することで HiRDB へ不正にアクセスし，接続することを防止します。

この機能を使用しない場合，パスワード認証方式（PA（Password Authentication）方式）を利用し，パスワードをスクランブルして HiRDB サーバへ送信しますが，この機能を使用する場合，チャレンジハンドシェイク認証方式（CHA（Challenge Handshake Authentication）方式）を使用してパスワードの秘匿化します。CHA 方式では，チャレンジレスポンス方式によってパスワードを秘匿化し，ネットワークにパスワードそのものを流さないようにします。CHA 方式では，HiRDB サーバへの接続ごとに生成する乱数キーを用いてパスワードを秘匿化するので，HiRDB への不正な接続の防止を強化できます。

この機能を使用する（CHA 方式を用いて接続させる）ためには，Version 09-50-08 以降の HiRDB クライアントを使用してください。

29.2　HiRDB 接続時のパスワード秘匿化機能の適応範囲

秘匿化の対象となるパスワードと，対象とならないパスワードを次の表に示します。

表 29–1　秘匿化の対象となるパスワードと秘匿化の対象とならないパスワード

秘匿化の対象となるパスワード	秘匿化の対象とならないパスワード
• CONNECT 文実行時のパスワード • 自動再接続機能による再接続時のパスワード	• GRANT 文のパスワード • SET SESSION AUTHORIZATION 文のパスワード

秘匿化対象とならない SQL は，不正にパケットを取得されないようにセキュアエリアで実行してください。

29.3　HiRDB 接続時のパスワード秘匿化機能の設定方法

システム共通定義とクライアント環境定義の設定を行ってください。

HiRDB サーバが許可する認証方式を，システム共通定義 pd_connect_auth_type オペランドに指定します。HiRDB クライアントがサーバへ接続するときの認証方式を，クライアント環境定義 PDAUTHTYPE に指定します。システム共通定義 pd_connect_auth_type オペランドとクライアント環境定義 PDAUTHTYPE の組み合わせによって，HiRDB クライアントがサーバへ接続するときの認証方式が決まります。システム共通定義 pd_connect_auth_type オペランドの詳細はマニュアル「HiRDB システム定義」の「ユーザ認証に関するオペランド」を，クライアント環境定義 PDAUTHTYPE の詳細はマニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」をそれぞれ参照してください。

HiRDB サーバが許可する認証方式と HiRDB クライアントがサーバへ接続するときの認証方式が異なる場合，UAP が HiRDB サーバへ接続したときに接続不可（KFPA19525-E で認証エラー）となります。

なお，CHA 方式をサポートしていない HiRDB サーバに対して，CHA 方式を指定した HiRDB クライアントが接続した場合は，接続不可（KFPA11560-E 又は KFPA19632-E で認証エラー）となります。認証方式の組み合わせによる接続可否と認証方式を次の表に示します。

表 29–2　認証方式の組み合わせによる接続可否と認証方式

システム共通定義 pd_connect_auth_type オペランド	クライアント環境定義 PDAUTHTYPE			
	PA 又は 省略時	CHA/SHA-256	CHA/SHA-512	CHA 又は CHA/SHA-256/SHA-512
ALL 又は省略時	○（PA）	○（SHA-256）	○（SHA-512）	○（SHA-512）
PA	○（PA）	×	×	×
PA，CHA/SHA-256	○（PA）	○（SHA-256）	×	○（SHA-256）
PA，CHA/SHA-512	○（PA）	×	○（SHA-512）	○（SHA-512）
PA，CHA/SHA-256/SHA-512	○（PA）	○（SHA-256）	○（SHA-512）	○（SHA-512）
PA，CHA	○（PA）	○（SHA-256）	○（SHA-512）	○（SHA-512）
CHA/SHA-256	×	○（SHA-256）	×	○（SHA-256）
CHA/SHA-512	×	×	○	○

システム共通定義 pd_connect_auth_type オペランド	クライアント環境定義 PDAUTHTYPE			
	PA 又は 省略時	CHA/SHA-256	CHA/SHA-512	CHA 又は CHA/SHA-256/SHA-512
			（SHA-512）	（SHA-512）
CHA 又は CHA/SHA-256/ SHA-512	×	○ （SHA-256）	○ （SHA-512）	○ （SHA-512）

（凡例）

○（PA）：接続できる（PA 方式で認証）

○（SHA-256）：接続できる（CHA 方式（SHA-256）で認証）

○（SHA-512）：接続できる（CHA 方式（SHA-512）で認証）

×：接続できない（KFPA19525-E で認証エラー）

29.4　HiRDB 接続時のパスワード秘匿化機能の運用例

HiRDB サーバへの認証方式を CHA 方式(SHA-512)に限定する場合

セキュリティレベルを高めるために，HiRDB への認証方式を CHA 方式（SHA-512）に限定する場合は，システム共通定義 pd_connect_auth_type オペランドに「CHA/SHA-512」だけ指定してください。これによって，クライアント環境定義 PDAUTHTYPE に「CHA」，「CHA/SHA-256/SHA-512」又は「CHA/SHA-512」のどれかを指定したクライアントだけが HiRDB サーバに接続できます。クライアント環境定義 PDAUTHTYPE に「CHA」，「CHA/SHA-256/SHA-512」又は「CHA/SHA-512」以外を指定したクライアントからの接続は，認証エラーになります。

CHA 方式に対応していない HiRDB クライアントを HiRDB サーバに接続する場合

上位互換のために，PA 方式で HiRDB に接続する場合は，システム共通定義 pd_connect_auth_type オペランドに「PA」を含めるか，「ALL」を指定するか，又は省略してください。

29.5　HiRDB 接続時のパスワード秘匿化機能の注意事項

1. CHA 方式に対応していない HiRDB クライアントを使用する場合，HiRDB サーバ側を PA 方式での接続を許可する設定にしてください。設定しない場合，CHA 方式に対応していない HiRDB クライアントが HiRDB サーバに接続できなくなります。
2. 次のユティリティを実行する場合には，「HiRDB 接続時のパスワード秘匿化機能の設定方法」を参照してクライアント環境定義 PDAUTHTYPE の設定を行ってください。
   - pddef
   - pdplgrgst
   - pddivinfgt
3. OS ログインユーザの簡易認証機能を使用しているユーザには，この機能は適用されません。OS ログインユーザの簡易認証機能については，「OS ログインユーザの簡易認証機能とは」を参照してください。

付録

付録 A　Q & A

HiRDB の運用でよくある質問をまとめました。ここでは，次に示す項目についてのQ & A を記載しています。

1. システムログファイルに関する質問
2. シンクポイントダンプファイルに関する質問
3. ステータスファイルに関する質問
4. 障害に関する質問
5. 表，及びインデクスに関する質問
6. HiRDB の開始に関する質問
7. HiRDB の終了に関する質問
8. 性能に関する質問
9. バックアップに関する質問
10. RD エリアの回復に関する質問
11. そのほかの質問

付録 A.1　システムログファイルに関する質問

(1)　スワップ先にできる状態のシステムログファイルがない旨の警告メッセージ（KFPS01224-I）が出力されました

質問

HiRDB を正常開始する場合，前回稼働時の現用ファイルを続いて使用しないのですか？

現象 1

HiRDB を正常開始又は強制正常開始したときに，スワップ先にできる状態のファイルがない旨の警告メッセージ KFPS01224-I が出力されました。このときのシステムログファイルの状態は次のとおりです。

- 前回稼働時の現用ファイルに十分な空きがある
- スワップ先にできる状態のファイルが一つある

現象 2

HiRDB を正常開始又は強制正常開始したときに，使用できるシステムログファイルがない旨のメッセージ KFPS01256-E が出力されました。その後，アボートコード Psjnf07 で HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了しました。このときのシステムログファイルの状態は次のとおりです。

- 前回稼働時の現用ファイルに十分な空きがある
- スワップ先にできる状態のファイルがない

お答えします

HiRDB を正常開始する場合は，前回稼働時の現用ファイルを続いて使用しません。HiRDB を正常開始又は強制正常開始する場合は，システムログファイルをスワップするため，現用ファイルが替わります。一つだけ残っていたスワップ先にできる状態のファイルを現用ファイルに割り当てるため，スワップ先にできる状態のファイルがなくなり，警告メッセージ KFPS01256-E が出力されました。

HiRDB を再開始した場合は前回稼働時の続きという形のため，システムログファイルは前回稼働時の続きから使用します（システムログファイルはスワップしません）。

正常開始の場合にシステムログファイルを前回稼働時の続きから使用すると，一つのシステムログファイルに前回稼働時の分と今回稼働時の分のシステムログが格納されるため，アンロードログファイルへのアンロードなどの管理が煩雑になります。したがって，正常開始以降の情報だけでシステムログが完結するように，正常開始ではシステムログファイルをスワップしています。

なお，pd_log_rerun_swap=Y を指定すると，再開始の場合でもシステムログファイルをスワップします。ただし，システムログファイルの容量が不足すると再開始が失敗するため，スワップ先にできる状態のファイルを増やしておいてください。

(2) システムログファイルが不正である旨のメッセージ（KFPS01251-E）が出力されました

質問

HiRDB を強制終了（ソフトウェア障害，及びハードウェア障害などでユニットが異常終了した場合も含む）した後に，HiRDB 又はユニットを強制正常開始すると，システムログファイルが不正である旨のメッセージ KFPS01251-E が出力されました。そして，このシステムログファイルがスワップ先として使用されなくなりました。原因と対策を教えてください。

お答えします

原因

HiRDB 又はユニットが異常終了したときの現用ファイルは正常にクローズされないため，ファイルとして不完全な状態です。

対策

pdlogunld コマンドで該当するシステムログファイルの内容をアンロードしてください。システムログが不要の場合は，pdlogchg コマンドでシステムログファイルの状態を変更してください。

(3) システムログファイルの追加と HiRDB の再開始を何度か繰り返しました

質問

HiRDB を再開始した直後にスワップ先にできる状態のファイルがないため，HiRDB が異常終了しました。そこで，新規にシステムログファイルを追加して HiRDB を再開始しましたが，再度システムログ

ファイル不足で異常終了しました。システムログファイルの追加と HiRDB の再開始を何度か繰り返しましたが，現象が解消されません。なぜでしょうか？

お答えします

次に示す原因が考えられます。

1. **pd_log_rerun_swap=Y を指定している**

   このオペランドを指定すると，再開始のときでも現用ファイルをスワップするため，スワップ先にできる状態のファイルが一つ減ります。このため，システムログファイルを一つ追加してもこの現象を解消できません。システムログファイルを二つ以上追加して HiRDB を再開始してください。

2. **上書きできる状態のファイルの容量が不足している**

   HiRDB は再開始時のロールバック処理でもシステムログを取得しています。このため，再開始を繰り返しただけでもシステムログの出力量が増えて，上書きできない状態のファイルが増えます。再開始時のロールバック処理がすべて完了すると，シンクポイントが取得されるため，このシンクポイント以降に HiRDB 又はユニットが異常終了した場合は，上書きできない状態のファイルは増えません。

   したがって，上書きできる状態のファイルを一つずつ追加するのではなく，二つ以上まとめて追加してください。追加するファイルの総容量の目安は，この現象が発生し始めた最初の再開始でのシステムログの入力範囲（KFPS01262-I メッセージで表示された箇所～KFPS01263-I メッセージで表示された箇所）の 1.5 倍以上です。

備考

pd_log_rerun_reserved_file_open=Y を指定し，予約のファイルを複数用意しておくと，このような現象を防げます。

付録 A.2　シンクポイントダンプファイルに関する質問

(1)　KFPS02179-I メッセージが多発しています

質問

KFPS02179-I メッセージ（シンクポイントダンプの取得契機を迎えたが，以前のシンクポイントダンプ取得処理が完了していないため，取得契機を無視した）が多発していますが，問題ありませんか？

お答えします

このメッセージは，次の場合に出力されます。

- システムログファイルのスワップ間隔が短い
- シンクポイントダンプの取得間隔に対して，UAP のトランザクションの実行時間が長い
- デファードライト処理によってシンクポイントダンプの取得が遅延している

このメッセージが頻繁に出力される場合は，factor code によって次の対処を行ってください。

- factor code = A01-01 の場合

「デファードライト処理適用時のシンクポイント処理時間のチューニング」に従って対策してください。

- factor code = A01-02 の場合

  前回のシンクポイントダンプ取得時に未決着状態だったトランザクションがまだ決着していません。シンクポイントダンプの取得間隔に対して UAP のトランザクションが長いことが原因です。続いて出力される KFPS04374-I メッセージに従って対策してください。対策後もこのメッセージが出力される場合は，「上記以外の場合」に従って対策してください。

- 上記以外の場合

  pd_log_sdinterval オペランドの値を大きくしてください。又は，システムログファイルの容量を大きくするなどの対策を実施してください。それでも頻繁にこのメッセージが出力される場合は，UAP を見直して，実行時間が長いトランザクションを複数のトランザクションに分割し，1 トランザクション当たりの実行時間を短くできないか検討してください。

なお，このメッセージが出力されても，スワップ先にできる状態のシステムログファイルが十分に用意してある場合は，オンラインの続行，及び再開始に支障はありません。ただし，HiRDB の再開始に時間が掛かることがあります。システムログファイルの数及び容量が十分でない場合は，上書きできる状態のファイルがなくなり（KFPS01256-E メッセージが出力される），HiRDB 又はユニットが異常終了することがあります。次に示す見直しをしてこのようなトラブルを防いでください。

データベースやディクショナリを更新してからコミットするまでの時間が長いトランザクションの場合

対策案を次に示します。

- 途中で COMMIT 文を発行できないか検討してください。
- 更新後に大量検索をするトランザクションは，更新と大量検索の順番を入れ替えて，トランザクション内の最初の更新からコミットまでを短くできないか検討してください。

システムログファイルの数及び容量が十分でない場合

対策案を次に示します。

- 出力されるシステムログをすべて格納できるだけの総ファイル容量を用意することをお勧めします。無理な場合は，少なくともマニュアル「HiRDB システム導入・設計ガイド」に記載してある最低容量以上は確保するようにしてください。
- 大量更新するトランザクションにはデータベースの更新ログを取得しないようにします。データベースの更新ログを取得するかどうかは，クライアント環境定義の PDDBLOG オペランドで指定します。ただし，データベースの更新ログを取得しない場合は，UAP 実行時に障害が発生すると，バックアップ取得時点にしかデータベースを回復できないため注意してください。

(2)　有効保証世代数について

質問

シンクポイントダンプファイルの有効保証世代数とは何のことでしょうか？

お答えします

HiRDB の再開始に備えてシステムログファイルをどの位置から読み始めればよいかなどの情報がシンクポイントダンプファイルに格納されています。シンクポイントダンプファイルに格納した位置情報以降のシステムログファイルは再開始時に使用される可能性があるため，上書きできない状態にします。

有効保証世代数とは，「何世代前のシンクポイントダンプファイルに対応するシステムログファイルを上書きできない状態にして保護するか」ということです。例えば，有効保証世代数が 1 の場合は，最新のシンクポイントダンプファイルに対応するシステムログファイルが上書きできない状態になります。有効保証世代数が 2 の場合は，最新のシンクポイントダンプファイル及び 1 世代前のシンクポイントダンプファイルに対応するシステムログファイルが上書きできない状態になります。このように有効保証世代数を 2 にすると，上書きできない状態のシステムログファイルが増える可能性があります。

シンクポイントダンプファイルの有効保証世代数は pd_spd_assurance_count オペランドで指定します。

付録 A.3　ステータスファイルに関する質問

(1)　ステータスファイルがスワップするときは？

質問

ステータスファイルがスワップするのは，ディスクの障害が発生したときだけですか？

お答えします

いいえ違います。次に示す場合にステータスファイルがスワップします。

- ステータスファイルに障害が発生した場合
- ステータスファイルの容量が満杯になった場合
- ステータスファイルのフラグメンテーションによって，まとまった空き領域を確保できなくなった場合
- pdstsswap コマンドを実行した場合

(2)　ステータスファイルの片系だけに障害が発生した場合でもステータスファイルはスワップするのですか？

質問

ステータスファイルの片系だけに障害が発生した場合でもステータスファイルはスワップするのですか？

それとも両系が障害になるまでスワップされないのですか？

お答えします

ステータスファイルの片系が障害になった時点でステータスファイルはスワップします。障害検知からスワップ完了までの HiRDB の処理を次に示します。

1. 両系共に正常な予備ファイルを探します。
2. 正常な系の内容をほかのステータスファイルの両系に複写します。このとき，むだな空き領域を詰め替えます。

3. 複写先のステータスファイルを現用ファイルにします。

(3)　ステータスファイルに障害が発生したときの処理について

質問

両系とも正常なステータスファイルが一つもない（A 系，B 系のどちらかが障害）場合の処理方式を決める次に示すオペランドがあります。

- pd_syssts_singleoperation = stop ｜ continue（ユニット用ステータスファイルの場合）
- pd_sts_singleoperation = stop ｜ continue（サーバ用ステータスファイルの場合）

このオペランドには stop と continue のどちらを指定すればよいですか？

お答えします

stop を指定すると HiRDB 又はユニットが異常終了します。continue を指定するとステータスファイルの片系運転を行います。

ステータスファイルは，全面回復処理のための情報を記録している重要なファイルです。continue を指定して片系運転中にステータスファイルが障害になると，両系障害で HiRDB 又はユニットが異常終了します。そして，現用ファイルが両系ともアクセスできないため，全面回復処理ができなくなります。したがって，次のような考え方で判断してください。

- HiRDB の異常終了よりも，全面回復処理の保証を重視する場合は stop を指定します。
- HiRDB を停止したくない（最悪の場合，全面回復処理はあきらめ，データベースをバックアップ時点まで戻します。又はデータロードし直します）場合は continue を指定します。

付録 A.4　障害に関する質問

(1)　障害発生時の調査資料は？

質問

障害発生時の調査資料としては何を取得すればよいですか？

お答えします

一般的には，障害情報取得コマンド（pdinfoget コマンド）で資料を採取してください。なお，障害内容によっては，「障害調査のために必要な情報」に示す資料が必要となります。

(2)　HiRDB を再開始後，SPP が DB の排他待ちタイムアウトで異常終了しました

質問

HiRDB を再開始後，SPP が RD エリアの排他待ちタイムアウトで異常終了しました。

1. 排他が残っているようですが考えられる原因は？

2. SPP が異常終了後，ロールバック中の状態のままで回復しません。考えられる原因は？
3. OpenTP1 を dcstop コマンド（OpenTP1 のコマンド）で正常終了させようとしたが，ロールバック中の SPP がいて終了できません。考えられる原因は？

お答えします

1. OpenTP1 のトランザクションサービス定義（trn）に，次の指定がされていないと考えられます。

   ・trnstring -n HiRDB_DB_SERVER -d

   これは，OpenTP1 のコミット最適化及びプリペア最適化を抑止する指定で，HiRDB が XA インタフェースで OpenTP1 と連携する場合の標準です。
2. OpenTP1 のトランザクションサービス定義（trn）に HiRDB のクライアント環境変数の指定がないか，又は指定値に誤りがあると考えられます。次に示すクライアント環境変数を見直してください。

   ・PDHOST

   ・PDNAMEPORT

   同時に，OpenTP1 のユーザサービスデフォルト定義（usrrc）も見直してください。
3. 2 と同件と考えられます。

注意事項

上記の状態となった場合は，OpenTP1 を強制停止させ，HiRDB システム定義を修正してから OpenTP1 を再起動してください。

参考

trn と usrrc で共通な環境変数は，env で指定することをお勧めします。

env に指定する場合は，trn や usrrc と環境変数が異なるため，マニュアル「HiRDB UAP 開発ガイド」で確認してください。

付録 A.5　表，及びインデクスに関する質問

(1)　既存のインデクスの定義を知るには？

質問

SQL でインデクスが定義されているかどうかが分かりますか？

お答えします

ディクショナリ表を検索すれば可能です。次に例を示します。

また，ディクショナリ搬出入ユティリティ（pdexp コマンド）の定義系 SQL の生成機能でも確認できます。

（例１）

インデクスが付いた表を一覧検索します。

```
SELECT TABLE_SCHEMA,TABLE_NAME,TABLE_ID
    FROM MASTER.SQL_TABLES WHERE N_INDEX >= 1
```

（例２）

定義した表のインデクス名，列名を表示します。

```
SELECT TABLE_SCHEMA,TABLE_NAME,INDEX_NAME,INDEX_ID,COLUMN_NAME
    FROM MASTER.SQL_INDEX_COLINF
        WHERE TABLE_SCHEMA = USER
```

（例３）

インデクスが付いた表の表名，表 id，インデクス名，インデクス id，列名をまとめて表示します。

```
SELECT X.TABLE_SCHEMA,X.TABLE_NAME,X.TABLE_ID,
  INDEX_ID,COLUMN_NAME
    FROM MASTER.SQL_TABLES X,MASTER.SQL_INDEX_COLINF Y
        WHERE X.TABLE_SCHEMA = Y.TABLE_SCHEMA
             AND X.TABLE_SCHEMA = USER
             AND X.TABLE_NAME = Y.TABLE_NAME
             AND N_INDEX >= 1
```

(2)　既存の FIX 表に列を追加するには？

質問

既にデータが格納されている FIX 表に列を追加したいのですが，方法はありますか？

お答えします

あります。列の追加方法については，「列を追加する方法」を参照してください。

付録 A.6　HiRDB の開始に関する質問

(1)　HiRDB の開始が遅いのですが

質問

ユニットの開始に時間が掛かるのはなぜですか？

1 時間～2 時間ぐらいしたら開始しました。pdstart コマンドを実行したときに KFPS05078-I メッセージが出力されました。

お答えします

次に示すことを確認してください。

1. KFPS00608-W メッセージ（-314）が複数出力されている場合は，全ユニットの pdunit 及び pdstart オペランドで指定したホスト名がすべて一致しているか，正しいホスト名（存在するホスト）を指定しているかを確認してください。
2. HiRDB が使用するホスト及びネットワークがすべて起動しているかを確認してください。

(2) pdstart コマンドがエラーになります（reason code= TIMEOUT）

質問

pdstart コマンドが KFPS01861-E メッセージ（reason code=TIMEOUT）を出力してエラーリターンします。なぜですか？

お答えします

原因

次に示す原因が考えられます。

1. ユニットの開始に予想以上の時間が掛かりました。
2. サーバ共通定義又は各サーバ定義の指定に誤りがあります。

対策

次に示す対策をしてください。

1. pd_start_time_out オペランドの指定値を大きくしてから，再度 pdstart コマンドを実行してください。
2. イベントログに出力されている HiRDB のメッセージを見て，間違いがある定義を修正してください。

(3) pdstart コマンドが無効になります

質問

pdstart コマンドが無効になります。なぜですか？

お答えします

何らかの障害のため HiRDB のサーバプロセスが起動できなかった可能性があります。

メッセージログ(%PDDIR%¥spool¥pdlog1 又は pdlog2)を見て障害原因を調査し対策してください。
メッセージログの内容で原因が分からない場合は，イベントログに出力されている HiRDB のメッセージで詳細を確認して対策してください。

(4) HiRDB/パラレルサーバが開始できません

質問

システムマネジャのユニットが異常終了しました。システムマネジャのユニットだけを再開始しましたが，HiRDB が開始完了になりません。

お答えします

システムマネジャのユニットの異常終了によって，システムマネジャ以外のユニットも異常終了している可能性があります。pdls コマンドで開始していないユニットを確認してください。

(5)　システムマネジャ以外のユニットが開始しません

質問

pdstart コマンドを実行しましたが，システムマネジャ以外のユニットが開始しないのはなぜですか？

お答えします

開始できなかったユニットのシステム共通定義の内容を確認してください。pdunit 又は pdstart オペランドの指定値が，システムマネジャがあるサーバマシンのシステム共通定義の内容と一致していない可能性があります。

開始できなかったユニットのシステム共通定義を修正して，pdstart -u コマンドでユニットを再度開始してください。

(6)　フロントエンドサーバが SUSPEND 状態です

質問

フロントエンドサーバのユニットの単独開始，又はフロントエンドサーバの単独開始をしたら，フロントエンドサーバのステータスが SUSPEND 状態になりました。ACTIVE 状態にするにはどうしたらよいですか？

お答えします

ディクショナリサーバが停止しているときに，フロントエンドサーバのユニット又はフロントエンドサーバを開始すると，ステータスが SUSPEND 状態になります。この場合，ディクショナリサーバを開始した後に，「pdstart -a -s フロントエンドサーバ名」を実行してください。

(7)　HiRDB を自動開始できますか？

質問

OS の開始，終了に合わせて HiRDB を開始，終了するにはどうしたらよいですか？

お答えします

1. OS の開始に合わせて HiRDB を開始するには，pd_mode_conf オペランドに AUTO を指定してください。
2. OS の終了に合わせて HiRDB を終了する手段は今のところありません。pdstop コマンドを実行して HiRDB が正常終了したのを確認してから OS を終了してください。

(8) HiRDB のサービス起動時にサービスのエラーコード（1067）が出力されました

質問

HiRDB のサービス起動時に，サービスのエラーコード（1067）が出力されました。原因は何でしょうか？

お答えします

次に示すどちらかの原因が考えられます。

- 仮想メモリ不足の可能性があります。仮想メモリの指定値は，実メモリの 2，3 倍程度を推奨しています。また，**［コントロールパネル］－［システム］－［仮想メモリ］**で，初期値と MAX が同じ値になっているか確認してください。
- %PDDIR%¥tmp ディレクトリが，ほかのアプリケーションプログラムからアクセスされたままになっていないか確認してください。HiRDB は，開始時に作業ディレクトリである%PDDIR%¥tmp を削除しているため，ほかのアプリケーションプログラムで%PDDIR%¥tmp をアクセスしていると排他エラーとなります。

(9) HiRDB のサービスを開始しましたが，応答が返ってきません

質問

HiRDB のサービスを開始しましたが，応答が返ってきません。また，イベントログにも何も出力されていません。

お答えします

pd_mode_conf オペランドの指定値が MANUAL1，又は MANUAL2 になっていないかを確認してください。MANUAL1，又は MANUAL2 の場合は，更にコマンドプロンプトから pdstart コマンドを実行する必要があります。AUTO の場合は，サービスを開始すればそのまま HiRDB を開始します。

付録 A.7 HiRDB の終了に関する質問

(1) pdstop -f コマンドを実行したらユニットが異常終了しました

質問

pdstop -f コマンドを実行したら，アボートコード Polkcrt で HiRDB 又はユニットが異常終了しました。なぜですか？

お答えします

強制終了処理でクリティカル状態のサーバプロセスを終了したのが原因です。運用上の問題はありません。この現象を無視してください。

pdstop -f コマンドの強制終了処理ではクリティカル状態のサーバプロセスの場合でも即刻停止するため，この現象が発生することがあります。しかし，次回の pdstart コマンドで再開始（システムログからデータベースを回復）するため問題はありません。

(2)　HiRDB を正常終了できません

質問

HiRDB を正常終了できません。

お答えします

未決着状態のトランザクションや接続ユーザがいると，HiRDB を正常終了できません。次に示すコマンドで，接続ユーザやトランザクションの状態を確認してから，HiRDB を正常終了させてください。

- pdls -d prc コマンドで接続ユーザを確認する
- pdls -d trn コマンドでトランザクションの状態を確認する

対処方法については，「接続中のユーザが存在するため HiRDB が終了できないときの対処方法」又は「未決着状態のトランザクションがあるときの対処方法」を参照してください。

付録 A.8　性能に関する質問

(1)　期待した検索性能が出ません

質問

表にインデクスを付けてインデクス経由の高速な検索を期待しているのに，期待どおりの検索性能が得られません。インデクスを使用していないように見えます。どんな原因が考えられますか？

お答えします

最新のデータベースの状態で最適化情報収集ユティリティ（pdgetcst コマンド）を実行しないと，本来使用するインデクスを使用した検索がされないことがあります。最適化情報収集ユティリティの実行要否については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

最適化情報収集ユティリティ（pdgetcst コマンド）を使用してコストベースの最適化をしている表については，次に示す要因が発生した場合，必ず再度 pdgetcst コマンドを実行して最新の最適化情報を取得してください。

- 大量にデータを追加，更新，又は削除した場合
- データベース作成ユティリティ（pdload コマンド）又はデータベース再編成ユティリティ（pdrorg コマンド）を実行した場合
- 表の定義変更，又はインデクスの追加，削除をした場合

備考

アクセスパス表示ユティリティ（pdvwopt コマンド）で，インデクスを経由した検索になっているかどうかを確認できます。

(2)　バッファ使用状況，入出力回数の確認方法は？

質問

バッファ使用状況（ヒット率など）又は入出力回数を調べて，バッファ関係のチューニングをしたいのですが，どのような確認方法がありますか？

お答えします

統計解析ユティリティ（pdstedit コマンド）でグローバルバッファに関する統計情報を編集すれば，バッファのヒット率や，実 READ/WRITE 回数を調べられます。この場合，統計情報の出力開始コマンド（pdstbegin コマンド）で統計情報種別「buf」を指定して統計情報の取得を開始してください。ただし，シンクポイント間の差分情報のため，シンクポイントダンプが最低 2 回有効にならないと編集出力されません。必要に応じて，次に示すコマンドでシステムログファイルを強制的にスワップさせて，シンクポイントダンプを有効化させてください。

- pdstbegin -k buf：統計情報の取得開始
- pdlogsync：シンクポイントダンプの有効化

統計情報の取得範囲

- pdlogsync：シンクポイントダンプの有効化
- pdstedit：統計情報の編集出力

備考

統計解析ユティリティと同等の情報が pdbufls コマンドで表示できます。pdbufls コマンド入力時点間の差分情報が表示されるため，統計情報の取得範囲を挟んで 2 回実行してください。

付録 A.9　バックアップに関する質問

(1)　バックアップの取得単位について

質問 1

表単位のバックアップはできますか？

お答えします

できません。しかし，次に示すどちらかの方法で代替できます。

- 1RD エリアに 1 表だけを格納し，RD エリア単位にバックアップを取得する
- データベース再編成ユティリティ（pdrorg コマンド）で表単位のアンロードをする

質問 2

前回のバックアップ取得時以降の差分をバックアップの対象にできますか？

お答えします

できます。HiRDB では差分バックアップ機能をサポートしています。差分バックアップ機能とは，前回のバックアップ取得時点からの差分情報だけをバックアップの対象にする機能です。このため，バッ

クアップの取得処理時間を短縮できます。データベースの容量が多くてデータ更新量が少ない場合に，差分バックアップ機能の使用を検討してください。差分バックアップ機能の運用方法については，「差分バックアップの取得」を参照してください。

質問３

pdcopy コマンドの-a オプションで全 RD エリアのバックアップを取得しています。このとき，特定の RD エリアが障害となったら，全 RD エリアのバックアップを使用して特定の RD エリアだけを回復できますか？

お答えします

できます。pdrstr コマンドで RD エリア名称を指定すれば，特定の RD エリアだけを回復できます。

(2)　pdfbkup コマンドと pdcopy コマンドの違いについて

質問１

pdfbkup コマンドで取得した HiRDB ファイルシステム領域のバックアップを pdrstr コマンドの入力ファイルとして使用できますか？

お答えします

できません。pdfbkup コマンドで取得したバックアップは pdfrstr コマンドでしか使用できません。

質問２

HiRDB 稼働中のバックアップを pdfbkup コマンドで取得したいのですが可能ですか？

お答えします

できます。「バックアップ閉塞を使用してバックアップを取得する場合（pdcopy コマンド以外でバックアップを取得する場合）」を参照してください。

(3)　マスタディレクトリ用 RD エリアのバックアップについて

質問１

HiRDB 稼働中に RD エリアのバックアップを取得できますか？

お答えします

できます。ただし，マスタディレクトリ用 RD エリアが含まれる場合は，バックアップ取得モードが次のどちらかとなります。

- 参照可能モード（-M r 指定）
- 更新可能モード（-M s 指定）

質問２

pdcopy コマンドで-a オプション（全 RD エリアのバックアップを取得するオプション）を指定した場合，バックアップの取得対象にマスタディレクトリ用 RD エリアは含まれますか？

また，含まれる場合は pdstart -r コマンドで HiRDB を開始する必要がありますか？

お答えします

マスタディレクトリ用 RD エリアは含まれます。

また，pdstart -r コマンドで HiRDB を開始する必要がある条件はマスタディレクトリ用 RD エリアのバックアップを取得する場合ではなく，次に示す条件をすべて満たす場合です。

- バックアップの取得対象 RD エリアにマスタディレクトリ用 RD エリアが含まれている場合
- バックアップ取得モードに参照・更新不可能モード（-M x 指定）を指定する場合

(4) ディスク容量に余裕があるのに，pdcopy コマンド実行時にディスク容量不足となります

質問

pdcopy コマンドの実行時，ディスク容量に余裕があるのにディスク容量不足のエラーとなります。なぜですか？

お答えします

ディスクの残容量に余裕があるのに，pdcopy コマンドの実行時にディスク容量不足を示す旨のメッセージが出力された場合は，バックアップファイルの大きさが上限を超えている可能性があります。バックアップファイルの最大値は次のとおりです。

条件	バックアップファイルの最大値
HiRDB ファイルシステム領域にバックアップファイルを作成する場合	1,048,575 メガバイト
バックアップファイルを Windows のファイルで作成する場合	$2^{63}-1$ バイト

付録 A.10 RD エリアの回復に関する質問

(1) pdrstr コマンドの仕様について

質問１

マスタディレクトリ用 RD エリアと特定のユーザ用 RD エリアに障害が発生したため，pdrstr コマンドの-a（全 RD エリアの回復）又は-c（バックアップ中の全 RD エリアの回復）オプションで全 RD エリアを回復しました。この場合，正常な RD エリアの情報も置き換えられるのでしょうか？

回復するときに内部的に削除してからインポートしているのですか？

お答えします

置き換えられます。pdrstr コマンドは回復対象の RD エリアを構成する HiRDB ファイルを削除してから，バックアップファイルの内容をリストアします。したがって，pdrstr コマンド実行時に回復対象として指定した RD エリアの内容はすべて置き換えられます。この場合，正常な RD エリアの情報も置き換えられます。

質問２

初期化した RD エリアを pdrstr -a コマンドで回復（全 RD エリアの回復）した場合，表の実体も回復されるのでしょうか？

回復されないのであれば，どのタイミングで CREATE TABLE するのでしょうか？

お答えします

回復されます。表の定義情報はディクショナリ上に格納されているため，ディクショナリ，データディレクトリ，及びマスタディレクトリと初期化した RD エリアの同期が取れていれば正しく回復されます。

(2)　マスタディレクトリ用 RD エリアの回復について

質問

マスタディレクトリ用 RD エリアはどのように回復するのでしょうか？

お答えします

マスタディレクトリ用 RD エリアを回復する場合，pdstart -r コマンドで HiRDB を開始する必要があります。回復方法については，「データベースの回復方法」を参照してください。

付録 A.11　そのほかの質問

(1)　ユーザ数が増えるとどのような影響がありますか？

質問

ユーザが増えるため pd_max_users オペランドの値を大きくしたいのですが，どのような影響がありますか？

お答えします

pd_max_users オペランドの値を大きくしたときの影響については，「ユーザ数を増やすときにすること」を参照してください。

(2)　CONNECT 中の UAP がないのにユーザ識別情報が表示されます

質問

CONNECT 中の UAP がないのに，pdls -d act コマンドを実行すると，ユーザ識別情報が表示されることがあります。なぜですか？

お答えします

サーバプロセスがアボートした後，又は HiRDB の再開始後にトランザクションのロールバックが完了していないと，ユーザ識別情報は残ったままとなります。このユーザ識別情報はトランザクションのロールバック完了後に削除されます。

(3)　時刻の変更はできますか？

質問

システム時刻を変更してテストをしたいのですが，できますか？

もし，日時を強制的に進めたり，遅らせたりするとどのような影響がありますか？

お答えします

HiRDB では，システムログファイル，ステータスファイル，ディクショナリ，及びユティリティの作業用ファイル中に日付・時間を格納しており，HiRDB の再開始時などにその情報を利用します。そのため，システム時刻を変更する場合は，HiRDB が持つ日付・時刻情報に影響がないようにする必要があります。特に，時刻を遅らせる場合は注意が必要です。時刻の変更については，「OS の時刻を変更する方法」を参照してください。

なお，SNTP クライアントで時刻を変更する場合，「OS の時刻を変更する方法」の記載内容に加えて，次の点に注意してください。

- プライマリサーバとの同期合わせで，必要であれば時刻を変更するため，ユーザが意識しないタイミングで時刻が変更されることがあります。

(4)　認可識別子やパスワードが正しく認識されません

質問

認可識別子やパスワードが正しく認識されていません。なぜですか？

お答えします

大文字，小文字が正しく認識されていない可能性があります。小文字を指定する場合は，引用符（"）で囲んでください。引用符で囲まないと，小文字を指定しても大文字として扱われます。

（例）

```
SELECT * FROM "user05".TABLE05;
```

(5)　pdcancel コマンドと pdkill コマンドの使い分け

質問

pdcancel コマンドと pdkill コマンドは，どのように使い分ければよいのですか？

お答えします

次のように使用してください。

- **pdcancel コマンド**
  トランザクション処理中のサーバプロセスを終了させます。
- **pdkill コマンド**
  トランザクション処理をしていないサーバプロセスを終了させます。

付録 B　DVD-RAM ライブラリ装置を使用するときの運用

ここでは，格納先デバイスとして DVD-RAM ライブラリ装置を使用するときの運用方法について説明します。

付録 B.1　システム共通定義の設定

DVD-RAM ライブラリ装置を使用する場合，物理的なマウント動作（ステージング）に時間が掛かるため，不当なタイムアウト検知を回避するための配慮が必要です。そのため，次に示すオペランドの指定値については，従来の見積もり値にマウント動作時間を加えた値を指定するように検討してください。

- pd_lck_wait_timeout（排他待ち限界経過時間）

付録 B.2　HiRDB ファイルシステム領域の作成

DVD-RAM ライブラリ装置の媒体上にファイルシステムを構築し，そこに HiRDB ファイルシステム領域を作成してください。この HiRDB ファイルシステム領域は次に示す目的用としてください。

- RD エリア用の HiRDB ファイルシステム領域
- ユティリティ用の HiRDB ファイルシステム領域

性能及び信頼性の理由から，これ以外の用途に使用しないでください。

なお，DVD-RAM ライブラリ装置ではダイレクトディスクアクセス（rawI/O）を使用できません。

付録 B.3　RD エリアのオープン契機

DVD-RAM ライブラリ装置に RD エリアを定義する場合，RD エリアのオープン契機を SCHEDULE 属性にしてください。INITIAL 属性（省略値）の場合，RD エリアの情報をメモリ上に常駐させるため，HiRDB 開始時にすべての HiRDB ファイルをオープンします。DVD-RAM ライブラリ装置のオープン処理が集中すると，ディスク交換が多発して HiRDB の開始処理がタイムアウトすることがあります。DEFER 属性の場合，HiRDB 開始時のオープン処理の集中を回避できますが，HiRDB の正常終了時にクローズ処理が集中します。

マスタディレクトリ用 RD エリア，データディレクトリ用 RD エリア，データディクショナリ用 RD エリア，データディクショナリ LOB 用 RD エリア，及びレジストリ用 RD エリアのオープン契機は INITIAL 属性固定のため，DVD-RAM ライブラリ装置の運用には適しません。これらの RD エリアは磁気ディスク上に作成してください。

RD エリアのオープン契機を SCHEDULE 属性にするには，**pd_rdarea_open_attribute_use ＝ Y** を指定した上で，SCHEDULE 属性にする RD エリアの数によって次に示すどちらかの方法をとってください。

SCHEDULE 属性にする RD エリアの数が多い場合

pd_rdarea_open_attribute ＝ SCHEDULE を指定します。この指定はマスタディレクトリ用 RD エリア，データディレクトリ用 RD エリア，データディクショナリ用 RD エリア，データディクショナリ LOB 用 RD エリア，及びレジストリ用 RD エリアを除いたシステム全体の RD エリアに対して有効になります。

SCHEDULE 属性にする RD エリアの数が少ない場合

RD エリアを定義するとき，次に示すどちらかのユティリティの制御文で指定します。

- データベース初期設定ユティリティ（pdinit コマンド）の **open attribute** オペランドに SCHEDULE を指定
- データベース構成変更ユティリティ（pdmod コマンド）の **open attribute** オペランドに SCHEDULE を指定

この指定は指定した RD エリアに対してだけ有効になります。また，pd_rdarea_open_attribute オペランドの指定と併用した場合は，ユティリティの指定が優先されます。

参考

データベース構成変更ユティリティで RD エリアを追加した直後はオープン契機の指定が有効になりません。追加した直後は INITIAL 属性になっています。オープン契機の指定を有効にするには HiRDB を一度終了させた後に再度開始してください。開始モードに関係なく有効になります。

付録 B.4　同時にアクセスする媒体数

DVD-RAM ライブラリ装置では内蔵ドライブの媒体を入れ替えながら複数枚の媒体を扱います。装置内のすべてのドライブがメディア交換禁止時間内のときに，新たな媒体にアクセスすると媒体の交換ができないため，アクセスエラーが発生します。RD エリアへのアクセスでエラーが発生すると，RD エリアが障害閉塞します。このため，システム全体で同時にアクセスする媒体数は装置の内蔵ドライブ数を超えないようにしてください。HiRDB 以外からの DVD-RAM ライブラリ装置へのアクセスがない場合は，1 トランザクションでアクセスする媒体数×接続ユーザ数が内蔵ドライブ数を超えないようにしてください。HiRDB 以外からのアクセスがある場合は，その分利用できる内蔵ドライブ数を少なく見積もってください。エクスプローラなどで DVD-RAM ライブラリ装置を接続した仮想ドライブを開く場合は，多数の媒体にアクセスすることがあるので注意が必要です。

付録 B.5　異常終了時の回復処理

DVD-RAM ライブラリ装置は出力時にキャッシュファイルを使用した遅延出力をしているため，電源断や Windows の異常終了などが発生すると，それ以前の書き込み処理が保証されません。更新処理をした RD エリアでは同期点での書き込みが保証されないため，HiRDB の再開始時にページ破壊を検知するなどして RD エリアを正しく回復できないことがあります。電源断や Windows の異常終了などが発生した場合は，

バックアップファイルとシステムログファイルから RD エリアを回復するか，又は RD エリアを再初期化してデータベースを再構築する必要があります。

付録 B.6　基本的な運用方針

DVD-RAM ライブラリ装置はデータの長期保存には向いていますが，更新を繰り返す処理には向いていません。したがって，更新処理を頻繁にする RD エリアはディスク上に配置し，更新処理を余りしない RD エリアは DVD-RAM ライブラリ装置上に配置してください。DVD-RAM ライブラリ装置上の RD エリアには，表データの入力（追加）はデータベース作成ユティリティなどで一括して行い，通常運用では参照処理だけをすることをお勧めします。

付録 C　障害調査のために必要な情報

原因を調査するために必要な資料は，大別すると，OS の情報と HiRDB の情報に分かれます。OS の情報は OS のコマンドなどで採取します。HiRDB の情報は HiRDB のコマンドなどで採取します。

サポートサービスをご利用になる場合，次の表に示す情報が必要になることがあります。トラブルの原因調査に必要な情報を次の表に示します。採取の優先順位を，性能の問題，無応答，異常終了の三つの場合に分けて説明します。最も優先順位の高い項目を 1 として，7 段階で示しています。

表 C–1　障害調査のために必要な情報（OS）

項番	取得する情報	取得方法	性能	無応答	異常終了
1	イベントログ	OS の機能で取得します。	1	1	1
2	CPU 利用率及びデバイス状況	パフォーマンスモニタで取得します。	3	4	3

表 C–2　障害調査のために必要な情報（HiRDB）

項番	取得する情報	取得方法	性能	無応答	異常終了
1	HiRDB 障害情報	pdinfoget コマンドで取得します。出力情報を取得してください。	2	2	2
2	エラーログファイル	%PDDIR%¥spool¥errlog 下のファイルに出力されます。	2	2	2
3	コマンドログファイル	%PDDIR%¥spool¥cmdlog 下のファイルに出力されます。	2	2	2
4	仕様差吸収ライブラリエラー情報ファイル	%PDDIR%¥UXPLDIR¥SPOOL¥uxpllog1 及び uxpllog2 に出力されます。	2	2	2
5	リモート系コマンド情報ファイル	%PDDIR%¥spool¥pdrshs1 及び pdrshs2 に出力されます。	2	2	2
6	系切り替え機能情報ファイル	%PDDIR%¥spool¥pdshs1 及び pdshs2 に出力されます。	2	2	2
7	HiRDB システム定義の情報	%PDDIR%¥conf 下のファイルを DAT などに取得してください。	4	5	4
8	表及びインデクスの定義情報	表及びインデクスを定義したときに使用した定義系 SQL を記載したファイルを DAT などに取得してください。	4	−	−
9	PRF トレース情報ファイル	%PDDIR%¥spool¥pdprf 下のファイルに出力されます。PRF トレースの出力ディレクトリは，pd_prf_output_directory オペランドで変更できます。 オンライン性能の低下が許容できる場合，速やかに原因を切り分けられるようにするために，PRF トレース取得レベルに	4	6	6

項番	取得する情報	取得方法	性能	無応答	異常終了
		0000007f（詳細レベル）を指定して，資料採取することを推奨します。 なお，pd_prf_level オペランドの指定値を変更しなくても，pdprflevel コマンドを使用することで，HiRDB 稼働中に取得レベルを変更することができます。			
10	統計情報	統計情報の出力開始，終了コマンド（pdstbegin，pdstend）で次に示す情報を取得してください。 • システムの稼働に関する統計情報（sys） 推奨取得間隔：1 分 • グローバルバッファプールに関する統計情報（buf） • デファードライト処理に関する統計情報（dfw） • データベースの入出力に関する統計情報（dio） 推奨取得間隔：1 分 これらの情報は%PDDIR%¥spool¥pdstj1 及び pdstj2 に出力されます。 統計情報の取得後，統計解析ユティリティ（pdstedit コマンド）で統計情報を編集してください。	6	−	−
11	SQL オブジェクト情報※1	pdobils -C -H -R コマンドで取得します。このコマンドを 60 秒間隔で実行して，追加書きのリダイレクト（>>）でファイルに出力してください。	5	−	−
12	SQL オブジェクト情報※2	pdobils コマンドの-N オプションに SQL オブジェクト番号を指定して取得します。	6	−	−
13	実行中の SQL 情報※3	pdobils -U -C -H コマンドで取得します。このコマンドを実行し，出力される情報をリダイレクト（>）でファイルに出力してください。	6	4	−
14	使用中の SQL 情報※4	pdobils -A -C -H コマンドで取得します。このコマンドを実行し，出力される情報をリダイレクト（>）でファイルに出力してください。	6	4	−
15	サーバの通信制御情報	pdls -d rpc -a コマンドで取得します。このコマンドを 60 秒間隔で実行して追加書きのリダイレクト（>>）でファイルに出力してください。そのファイルを DAT などに取得してください。 • 推奨取得間隔：60 秒 • 情報を 1 回取得するのに必要なディスク容量：約 62 キロバイト	5	4	4
16	サーバのトランザクションの状態に関する情報	pdls -d trn -a コマンドで取得します。このコマンドを 60 秒間隔で実行して追加書きのリダイレクト（>>）でファイルに出力してください。そのファイルを DAT などに取得してください。 • 推奨取得間隔：60 秒 • 情報を 1 回取得するのに必要なディスク容量：約 30 キロバイト	5	6	5

項番	取得する情報	取得方法	性能	無応答	異常終了
17	サーバのプロセスの状態に関する情報	pdls -d prc -a コマンドで取得します。このコマンドを 60 秒間隔で実行して追加書きのリダイレクト（>>）でファイルに出力してください。そのファイルを DAT などに取得してください。 • 推奨取得間隔：60 秒 • 情報を 1 回取得するのに必要なディスク容量：約 28 キロバイト	5	6	5
18	サーバの排他制御に関する情報	pdls -d lck -a コマンドで取得します。このコマンドを 60 秒間隔で実行して追加書きのリダイレクト（>>）でファイルに出力してください。そのファイルを DAT などに取得してください。 • 推奨取得間隔：60 秒 • 情報を 1 回取得するのに必要なディスク容量：約 62 キロバイト	5	−	−
19	グローバルバッファ使用状況に関する情報	pdbufls コマンドで取得します。このコマンドを 60 秒間隔で実行して追加書きのリダイレクト（>>）でファイルに出力してください。そのファイルを DAT などに取得してください。 • 推奨取得間隔：60 秒 • 情報を 1 回取得するのに必要なディスク容量：約 3 キロバイト	5	−	−
20	SQL トレースファイル及びエラーログファイル	クライアント環境定義の PDCLTPATH オペランドで指定したディレクトリ下に出力されます。PDCLTPATH オペランドを省略した場合は，AP を実行しているカレントディレクトリ下に出力されます。 出力されたファイルを DAT などに取得してください。ファイル名は pderr 又は pdsql で始まっています。	−	6	5
21	システムログファイル	pdlogunld コマンドでシステムログをアンロードします。アンロードログファイルを DAT などに取得してください。	6	7	6

（凡例）

−：情報を取得する必要はありません。

注※1　必要なディスク容量は次の計算式で算出してください。

$$1000 + \sum_{k=1}^{n} (2500 + \text{SQL文長}) \text{（単位：バイト）}$$

n：SQL オブジェクトバッファに格納している SQL オブジェクト数

SQL オブジェクトバッファに格納している SQL オブジェクト数は，「システムの稼働に関する統計情報」の SQL オブジェクト用バッファの SQL オブジェクト数の平均値と，ストアドプロシジャのオブジェクト数の平均値を足した数を使用してください。システムの稼働に関する統計情報は，マニュアル「HiRDB コマンドリファレンス」の「統計解析ユティリティ（pdstedit）」の説明を参照してください。

注※2　必要なディスク容量は次の計算式で算出してください。

500＋SQLオブジェクト長（単位：バイト）

SQL オブジェクト長は，-N オプション指定の pdobils コマンドで表示される「SIZE（サイズ）」の値を使用してください。「SIZE（サイズ）」については，マニュアル「HiRDB コマンドリファレンス」の pdobils コマンドの説明を参照してください。

注※3　必要なディスク容量は次の計算式で算出してください。

$$1000 + \sum_{k=1}^{n} (3000 + \text{SQL文長}) \text{（単位：バイト）}$$

n：SQL を実行している UAP 数

SQL を実行している UAP 数は，pd_max_users オペランドの値を使用してください。

SQL 文長は，UAP 中で使用している最も長い SQL 文長の値を使用してください。

注※4　必要なディスク容量は次の計算式で算出してください。

$$1000 + \sum_{k=1}^{n} ((3000 + \text{SQL文長}) \times m) \text{（単位：バイト）}$$

n：SQL を使用している UAP 数

SQL を使用している UAP 数は，pd_max_users オペランドの値を使用してください。

SQL 文長は，UAP 中で使用している最も長い SQL 文長の値を使用してください。

m：UAP で使用している SQL 数

UAP で使用している SQL 数は，pd_max_access_tables オペランドの値を使用してください。

注

- 必要なディスク容量は参考値です。この値はシステム構成などによって変動します。
- リダイレクトで追加書きするファイルはファイル容量が単調増加するため，ディスク容量を圧迫します。したがって，ファイルを切り替えて一定世代で再使用する bat ファイルを作成してください。

付録 D　HiRDB を 24 時間連続稼働するときの注意

ここでは，HiRDB を 24 時間連続稼働するときにお勧めする運用方法，及び注意事項について説明します。説明する項目は次のとおりです。

1. システム構成変更コマンド（pdchgconf コマンド）
2. HiRDB システム定義の指定
3. バックアップの取得
4. データベースの再編成
5. 使用中空きページとページ内の空き領域の再利用
6. RD エリアの拡張
7. グローバルバッファの動的変更
8. トラブルシュート情報の削除
9. 系切り替え機能
10. プログラムメンテナンス機能（修正版レベルのバージョンアップ）
11. 回復不要 FES（HiRDB/パラレルサーバ限定）

付録 D.1　システム構成変更コマンド（pdchgconf コマンド）

HiRDB システム定義（UAP 環境定義を除く）を変更する場合は HiRDB を終了する必要がありますが，システム構成変更コマンドを使用すると，HiRDB の稼働中に HiRDB システム定義を変更できます。このため，次に示すことを HiRDB の稼働中に実行できます。

- ユニットの追加，削除，及び移動
- サーバの追加，削除，及び移動
- システムファイルの追加
- グローバルバッファの追加，削除，及び変更

24 時間連続で稼働するようなシステムにはこのシステム構成変更コマンドが非常に便利です。このコマンドの使い方については次に示す箇所を参照してください。

- システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。
- ユニット又はサーバの追加，削除，移動については，「システム構成を変更する方法」を参照してください。

- システムファイルの追加については，「システムログファイルを新規追加する場合」，「シンクポイントダンプファイルを新規追加する場合」，又は「ステータスファイルを新規追加する場合」を参照してください。
- グローバルバッファの追加，削除，及び変更については，「RD エリアを作成する方法（RD エリアの追加）」を参照してください。

なお，システム構成変更コマンドを使用する場合は HiRDB Advanced High Availability が必要になります。

注意事項

HiRDB の稼働状況によっては，pdchgconf コマンドが実行できないことがあります。詳細については，マニュアル「HiRDB の稼働状況によっては，pdchgconf コマンドが実行できないことがあります。詳細については，マニュアル「HiRDB コマンドリファレンス」の pdchgconf コマンドの説明を参照してください。

付録 D.2　HiRDB システム定義の指定

HiRDB システム定義のオペランドを変更するには，HiRDB を一度終了する必要があります。そのため，次の表に示すオペランドについては先を見越して指定値を決めてください。HiRDB を 24 時間連続稼働するときに指定値に留意するオペランドを次の表に示します。

なお，HiRDB Advanced High Availability を導入した場合はシステム構成変更コマンドを使用できるため，HiRDB の稼働中に HiRDB システム定義を変更できます。

表 D–1　HiRDB を 24 時間連続稼働するときに指定値に留意するオペランド

オペランド名	留意事項
pd_max_users pd_max_server_process pd_max_bes_process pd_max_dic_process	同時接続クライアント数が増える見込みがある場合は，これらのオペランドの指定値に余裕を持たせてください。また，pd_process_count オペランドの値を小さくすることで，不要な（接続していない）サーバプロセス数をできるだけ抑えられます。 なお，pd_max_server_process オペランドを省略すると，このオペランドの値を HiRDB が自動計算します。
pd_stj_file_size	取得する統計情報によってファイル容量は異なりますが，省略値より大きい値にすることをお勧めします。
pd_spd_syncpoint_skip_limit	このオペランドを指定する場合は，今後実行する予定のバッチジョブで扱うデータ量を意識してください。
pd_lck_pool_size	業務の形態が変わってコミットまでに排他を掛けるリソース数が増える可能性がある場合，それを含めてこのオペランドの値を決めてください。
pd_lck_until_disconnect_cnt	業務の形態が変わる可能性がある場合，それを含めてこのオペランドの値を決めてください。
pd_sds_shmpool_size	これらのオペランドの指定値は，ほかのオペランドの指定値から算出します。したがって，ほかのオペランドの指定値を変更した場合，これらのオペランドの指定値を

オペランド名	留意事項
pd_dic_shmpool_size pd_bes_shmpool_size	再計算する必要があります。どのオペランドの指定値に依存するかについては，マニュアル「HiRDB システム導入・設計ガイド」の「シングルサーバが使用する共用メモリの計算式」又は「各サーバが使用する共用メモリの計算式」を参照してください。 なお，これらのオペランドを省略すると，これらのオペランドの値を HiRDB が自動計算します。
pdlogadfg -d sys pdlogadpf -d sys	システムログファイルの総容量は余裕を持たせるようにしてください。これらのオペランドで予約ファイルを定義しておいて，必要に応じてシステムログファイルの実体を追加できます。
pd_log_sdinterval	経過時間に 0 を指定しないでください。

付録 D.3　バックアップの取得

更新可能モード（-M s 指定）でバックアップを取得することをお勧めします。この場合，次に示す注意事項があります。

- バックアップの取得中に更新トランザクションとの排他待ちで pdcopy コマンドがタイムアウトする可能性があります。これを回避するには，pdcopy コマンドの排他待ち限界経過時間（-j オプション）を長くするか，又は更新トランザクションが発行されないときに pdcopy コマンドを実行するようにしてください。
- データベースをバックアップ取得時点に回復できません。バックアップ取得時点以降の任意の同期点への回復だけとなります。したがって，データベースを回復するには，バックアップ及びバックアップ取得直前のシンクポイントからのシステムログが必要になります。最新の同期点にデータベースを回復する場合，データベース複写ユティリティの処理結果出力ファイルに，RD エリアを回復するときに必要なシステムログファイルのラン ID 及び世代番号が出力されます。それ以降のシステムログファイルをすべて管理してください。

  なお，バックアップ取得時点以降の最初の同期点にデータベースを回復する場合は，バックアップ取得前の最後の同期点からバックアップ取得後の最初の同期点までのシステムログファイルを管理してください。

付録 D.4　データベースの再編成

再編成中の表は EX（排他モード）で排他されるため，UAP が排他待ちとならないように再編成対象表を格納している RD エリアを pdhold コマンドで閉塞状態にすることをお勧めします。閉塞状態の RD エリアはアクセスできないため，再編成対象表にアクセスした UAP は排他待ちにならないですぐに終了します。再編成対象表にアクセスできない時間を設定してその間に表を再編成してください。

付録 D.5　使用中空きページとページ内の空き領域の再利用

行の挿入及び削除を繰り返す業務では表やインデクスの格納効率が悪くなり，性能低下の一因になります。そのため，定期的にデータベース再編成ユティリティで表やインデクスを再編成する必要があります。しかし，ユティリティの実行中はほかの UAP 又はユティリティからその表やインデクスをアクセスできないため，その表をアクセスする業務を停止する必要があります。業務を停止できないため表又はインデクスを再編成できない場合に，次に示す機能を使用してください。

- 使用中空きページ及び使用中空きセグメントの解放
- 空き領域の再利用機能

使用中空きページ及び使用中空きセグメントを解放すると使用中空きページ及び使用中空きセグメントを再利用でき，空き領域の再利用機能を使用するとページ内の空き領域を再利用できます。これらの機能を組み合わせるとデータの格納効率が上がり，性能低下を抑えられるため，表又はインデクスを再編成するサイクルを長くできます。

(1)　使用中空きページ及び使用中空きセグメントの解放

バッチジョブなどで表データを大量に削除すると，その表データを格納しているページ（データページ）の一部が使用中空きページになることがあります。また，インデクスを定義している場合は，インデクスのキー値を格納しているページ（インデクスページ）の一部が使用中空きページになります。空きページ解放ユティリティ（pdreclaim コマンド）を実行すると，この使用中空きページを未使用ページ化して再利用できます。これを**使用中空きページの解放**といいます。また，使用中空きセグメントを未使用セグメント化して再利用できます。これを**使用中空きセグメントの解放**といいます。

使用中空きページ及び使用中空きセグメントの解放については，「使用中空きページ及び使用中空きセグメントを再利用する方法」を参照してください。

(2)　空き領域の再利用機能

INSERT，又は UPDATE 処理でセグメント内にデータを格納できない場合，未使用セグメントを新たに割り当てないで，使用中セグメント内の空き領域（ページ内の空き領域）にデータを格納します。これを**空き領域の再利用機能**といいます。この機能を使用すると，データの削除によって発生したむだな空き領域を再利用できます。

空き領域の再利用機能を使用する場合は，CREATE TABLE 実行時に SEGMENT REUSE を指定してください。空き領域の再利用機能については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

付録 D.6　RD エリアの拡張

データの追加によって RD エリアの容量不足が発生する可能性があります。次に示す施策を行ってください。

(1)　RD エリアの自動増分

RD エリアが容量不足になったとき，HiRDB ファイルシステム領域内に空き領域があれば，自動的にセグメントを追加して RD エリアの容量を拡張します。これを RD エリアの自動増分といいます。RD エリアの自動増分については，「RD エリアの自動増分」を参照してください。

なお，RD エリアの自動増分時に業務を停止する必要はありません。

(2)　RD エリアの拡張

(1)で説明した RD エリアの自動増分回数には限りがあります。自動増分できなくなった場合は，RD エリアを拡張してください。RD エリアの拡張方法については，「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

なお，RD エリアを構成する HiRDB ファイル数の上限は 16 です。これ以上の HiRDB ファイルの追加はできません。

また，RD エリアの拡張時に業務を停止する必要はありません。

(3)　RD エリアの再初期化

(2)で説明した RD エリアの拡張には限りがあります。拡張できなくなった場合は，RD エリアを再初期化して RD エリアの容量を大きくしてください。RD エリアの再初期化方法については，「RD エリアの容量又は属性を変更する方法（RD エリアの再初期化）」を参照してください。

なお，RD エリアを再初期化するときは，再初期化対象 RD エリアにアクセスできなくなります。そのため，なるべく(1)(2)の方法で RD エリアを拡張してください。

付録 D.7　グローバルバッファの動的変更

HiRDB Advanced High Availability を導入して，pd_dbbuff_modify オペランドに Y を指定することをお勧めします。そうすると，HiRDB の稼働中にグローバルバッファを追加，変更，又は削除できます。これをグローバルバッファの動的変更といいます。例えば，次に示す場合にグローバルバッファを動的変更してください。

- 追加した RD エリアにグローバルバッファを割り当てる場合
- RD エリアの割り当て先グローバルバッファを変更する場合
- グローバルバッファのチューニングの結果，グローバルバッファの定義を変更する場合

動的変更したグローバルバッファの情報は HiRDB を正常終了又は計画停止すると無効になるため，HiRDB を終了したときに pdbuffer オペランドの指定値を変更してください。なお，システム構成変更コマンド（pdchgconf コマンド）を使用すると，HiRDB の稼働中に pdbuffer オペランドの指定値を変更できます。

グローバルバッファの動的変更については，「HiRDB の稼働中にグローバルバッファを追加，変更，又は削除する方法（グローバルバッファの動的変更）」を参照してください。システム構成変更コマンドで HiRDB システム定義を変更する方法については，「HiRDB の稼働中に HiRDB システム定義を変更する方法（システム構成変更コマンド）」を参照してください。

付録 D.8　トラブルシュート情報の削除

サーバプロセス，又はクライアントの強制終了時などに，HiRDB は%PDDIR%¥spool 下にトラブルシュート情報を出力します。また，コマンド又はユティリティを［Ctrl ＋ C］キーを押すなどして途中終了させると，%PDDIR%¥tmp 下にコマンド又はユティリティが出力した作業用一時ファイルが削除されないで残ります。これらのファイルを残しておくと，HiRDB 運用ディレクトリがあるディスクの容量を圧迫する原因になります。HiRDB 運用ディレクトリがあるディスクの容量が不足すると HiRDB が異常終了することがあるため，HiRDB は次に示すファイルを定期的に削除します。

- トラブルシュート情報ファイル（%PDDIR%¥spool 下のファイル）
- 作業用一時ファイル（%PDDIR%¥tmp 下のファイル）
- pd_tmp_directory オペランドに指定したディレクトリ下のファイル

通常は 24 時間ごとに削除します。この削除間隔を **pd_spool_cleanup_interval** オペランドで変更できます。また，**pd_spool_cleanup_interval_level** オペランドで指定した日より前に出力されたファイルだけを削除するという指定ができます。

このほかにも，**pdcspool** コマンドでトラブルシュート情報ファイルを削除できます。作業用一時ファイル（%PDDIR%¥tmp 下のファイル）も削除できます。

なお，pdcspool コマンドのオプション又は pd_spool_cleanup_interval_level オペランドの指定で，削除するトラブルシュート情報を選択できます。

注意事項

- HiRDB 管理者以外のユーザが実行したコマンド又はユティリティによって出力されたトラブルシュート情報ファイルは削除されないことがあります。この場合，トラブルシュート情報ファイルの削除権限を持つユーザがエクスプローラなどでファイルを削除してください。
- pd_tmp_directory オペランドを指定しないで，環境変数 TMP だけを指定している場合，HiRDB の運用コマンド又はユティリティを中断すると，HiRDB は TMP に指定したディレクトリ下に"pdcmd"又は"plcmd"で始まるファイルを作成することがあります。HiRDB の運用コマンド又はユティリティが終了した後も"pdcmd"及び"plcmd"で始まるファイルが削除されない場合は，エクスプローラなどでこれらのファイルを削除してください。

付録 D.9　系切り替え機能

系切り替え機能を使用する場合，次に示す機能を使用することをお勧めします。これらの機能を使用すると，系の切り替え時間を短縮できるため，業務停止時間を最小限に抑えられます。

- 高速系切り替え機能
- スタンバイレス型系切り替え機能
- トランザクションキューイング機能（HiRDB/パラレルサーバ限定）

(1)　高速系切り替え機能

待機系 HiRDB のサーバプロセス及びシステムサーバをあらかじめ起動しておいて，系の切り替え時にサーバプロセス及びシステムサーバの起動処理をしません。これを高速系切り替え機能といいます。系の切り替え時にサーバプロセス及びシステムサーバの起動処理がない分，系の切り替え時間を短縮できます。高速系切り替え機能については，「高速系切り替え機能」を参照してください。

(2)　スタンバイレス型系切り替え機能

待機系 HiRDB を準備するスタンバイ型系切り替え機能とは異なり，スタンバイレス型系切り替え機能では待機系 HiRDB を準備する必要がありません。障害が発生した場合は待機系 HiRDB に系を切り替えるのではなく，稼働中のほかのユニットに処理を代行させます。これをスタンバイレス型系切り替え機能といいます。稼働中のユニットを利用するため，系の切り替え時間を短縮できます。

(3)　トランザクションキューイング機能

バックエンドサーバ又はディクショナリサーバがあるユニットで系切り替えが発生すると，系の切り替えが完了するまでこのバックエンドサーバ又はディクショナリサーバはトランザクションを受け付けられない状態になります。そのため，系切り替え中のバックエンドサーバ又はディクショナリサーバで処理するトランザクションがエラーになります。

これらのトランザクションをエラーにしないで，系切り替えが完了するまでフロントエンドサーバでトランザクションをキューイングします。これをトランザクションキューイング機能といいます。これによって，系切り替え時のトランザクションエラーを少なくできます。トランザクションキューイング機能については，「トランザクションキューイング機能」を参照してください。

●計画系切り替え時のトランザクションキューイング機能の利用

pdtrnqing コマンドでトランザクションキューイングを開始できます。計画系切り替えを行うときに pdtrnqing コマンドでトランザクションキューイングを開始しておくと，計画系切り替え中も新規トランザクションがキューイングされるため，計画系切り替え中に UAP を実行してもエラーになりません。トランザクションキューイング機能を使用した計画系切り替えについては，「計画系切り替え」を参照してください。

付録 D.10　プログラムメンテナンス機能（修正版レベルのバージョンアップ）

修正版レベルのバージョンを上げる場合（これを修正版 HiRDB への入れ替えといいます）は，HiRDB を正常終了する必要はありません。プログラムメンテナンス機能を使用すると，HiRDB を稼働したまま修正版レベルのバージョンを上げられます。

なお，ここでいう修正版レベルのバージョンとは，07-02-/A のように，稼働中の HiRDB とバージョン番号及びリビジョン番号が同じで，「-mn」のコードがあるものです（下線部がコード）。07-03 より前のバージョンの場合，m は/，英字（I，O，P～T を除く），又は数字，n は A～Z のアルファベットです。07-03 以降は m，n は共に数字です。例えば，07-02-/A から 07-02-/B へのバージョンアップは，HiRDB を正常終了しなくてもできます。これによって，業務を停止することなく修正版レベルのバージョンを上げられます。プログラムメンテナンス機能については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

注意事項

HiRDB の稼働状況によっては，pdprgrenew コマンドが実行できないことがあります。詳細については，マニュアル「HiRDB コマンドリファレンス」の pdprgrenew コマンドの説明を参照してください。

付録 D.11　回復不要 FES（HiRDB/パラレルサーバ限定）

フロントエンドサーバのユニットが異常終了した場合，このフロントエンドサーバから実行していたトランザクションが未決着状態になることがあります。未決着状態のトランザクションはデータベースの排他を確保しているため，ほかのトランザクションのデータベースへの参照又は更新が制限されます。未決着状態のトランザクションはフロントエンドサーバから決着処理を行う必要があるため，フロントエンドサーバのユニットの障害を回復した後にユニットを再開始する必要がありました。

回復不要 FES を使用すると，フロントエンドサーバのユニットが異常終了した場合に未決着状態となっていたトランザクションを自動的に決着できます（フロントエンドサーバの再開始を待たないでトランザクションを決着できます）。これによって，フロントエンドサーバのユニットを系切り替え構成にしないアプリケーションサーバに配置して，フロントエンドサーバのユニットを縮退運用できます。

また，pdrplstart 及び pdrplstop コマンドは，すべてのユニットとサーバが稼働中の場合に実行できますが，回復不要 FES を使用しているときは，回復不要 FES 又は回復不要 FES ユニットが停止していても実行できます。

回復不要 FES については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

なお，回復不要 FES を使用するには **HiRDB Non Recover FES** が必要になります。

付録 E　性能向上機能の運用

ここでは，性能向上を目的とした機能，及びその運用方法について説明します。

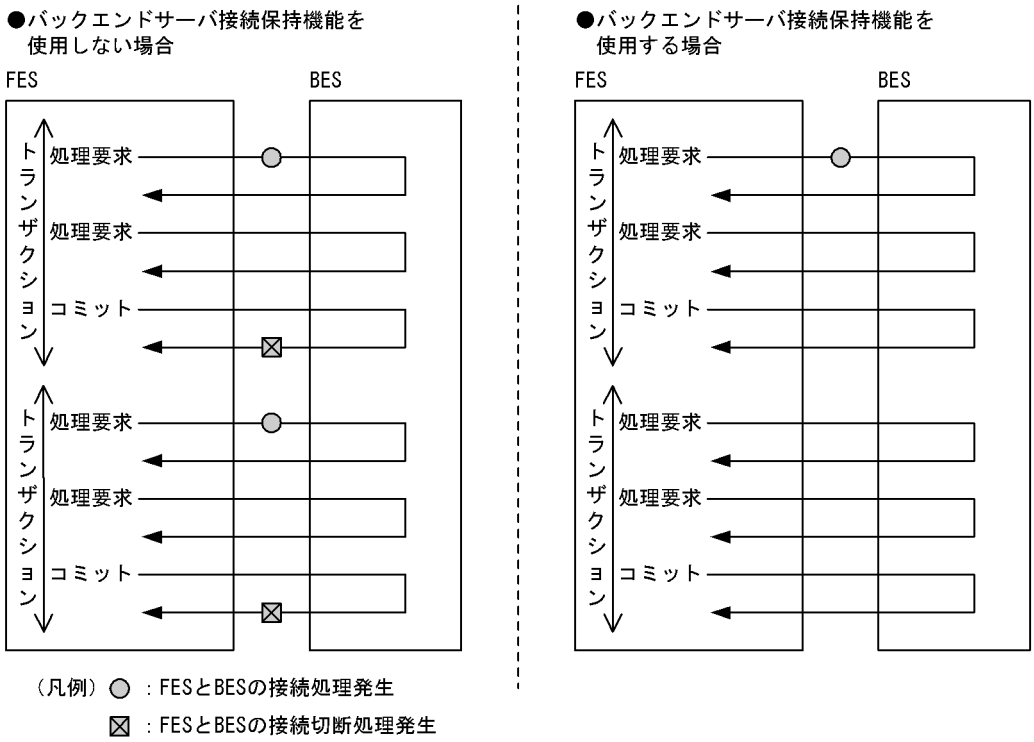
付録 E.1　バックエンドサーバ接続保持機能（HiRDB/パラレルサーバ限定）

(1)　バックエンドサーバ接続保持機能とは

HiRDB/パラレルサーバの場合，トランザクションの開始時に**フロントエンドサーバとバックエンドサーバの接続処理**が発生するため，高トラフィック及び高スループットが求められるシステムでは，この接続処理が性能ネックになることがあります。**バックエンドサーバ接続保持機能**を使用すると，トランザクションの終了時にフロントエンドサーバとバックエンドサーバの接続切断処理を行いません。このため，以降に実行されるトランザクションが同じバックエンドサーバを使用する場合，フロントエンドサーバとバックエンドサーバの接続処理が発生しなくなります。したがって，フロントエンドサーバとバックエンドサーバの接続処理が性能ネックになることを防げます。

バックエンドサーバ接続保持機能を使用する場合のトランザクションの流れを次の図に示します。

図 E–1　バックエンドサーバ接続保持機能を使用する場合のトランザクションの流れ



参考

- バックエンドサーバ接続保持機能を使用すると，フロントエンドサーバとバックエンドサーバの接続切断処理も削減されますが，接続切断処理が削減されても性能上の効果はあまりありません。
- バックエンドサーバ接続保持機能を使用しても，フロントエンドサーバとディクショナリサーバの接続は保持されません。トランザクション終了時にフロントエンドサーバとディクショナリサーバの接続切断処理を行います。

(2)　適用基準

1. フロントエンドサーバとバックエンドサーバの接続処理に掛かる時間は通常数ミリ秒～数十ミリ秒のため，トランザクションの実行時間が短い（数十ミリ秒～2 秒くらい）システムでこの機能を使用してください。トランザクションの実行時間が長いシステムでこの機能を使用した場合，性能向上は見込まれません。
2. 次に示すようなシステムで HiRDB に常時接続する場合は，バックエンドサーバが使用されないのにフロントエンドサーバとバックエンドサーバの接続を保持することがあるため，メモリ，ソケット，通信ポートなどの資源を圧迫します（それぞれの資源が常に高負荷時の資源を確保することになります）。したがって，バックエンドサーバ接続保持機能をこの環境下で使用する場合は注意が必要です。
   - トランザクションの負荷が低いシステム
   - 使用するバックエンドサーバに偏りがあるシステム（極端に使用頻度の低いバックエンドサーバに対しては，バックエンドサーバ接続保持機能を使用しないでください）
3. HiRDB XA ライブラリを使用している場合，使用するトランザクションマネジャによっては 1 接続が 1 トランザクションになります。この場合，バックエンドサーバ接続保持機能を使用しても効果がありません。使用できる XA ライブラリ一覧については，マニュアル「HiRDB UAP 開発ガイド」の各トランザクションマネジャが使用するライブラリ一覧を参照してください。

(3)　環境設定

バックエンドサーバ接続保持機能を使用する場合は次に示す環境設定をしてください。

- バックエンドサーバ接続保持機能の設定
- バックエンドサーバ接続保持時間の設定
- クライアントの最大待ち時間の設定
- バックエンドサーバプロセス数の設定

(a)　バックエンドサーバ接続保持機能の設定

バックエンドサーバ接続保持機能を使用する場合は，次に示すどちらかのオペランドを指定してください（両方指定してもかまいません）。

1. pd_bes_connection_hold オペランド
2. クライアント環境定義の PDBESCONHOLD オペランド

pd_bes_connection_hold オペランドを指定すると，全 UAP に対してバックエンドサーバ接続保持機能が適用されます。クライアント環境定義の PDBESCONHOLD オペランドを指定すると，PDBESCONHOLD オペランドを指定したクライアントから実行する UAP に対してだけバックエンドサーバ接続保持機能が適用されます。なお，二つのオペランドの優先順位は 2，1 の順になります。オペランドの組み合わせの例を次に示します。

- **pd_bes_connection_hold = Y を指定した場合**
  全 UAP に対してバックエンドサーバ接続保持機能が適用されます。
- **PDBESCONHOLD = YES を指定した場合**
  このオペランドを指定したクライアントから実行する UAP に対してだけバックエンドサーバ接続保持機能が適用されます。
- **pd_bes_connection_hold = Y，PDBESCONHOLD = NO を指定した場合**
  PDBESCONHOLD = NO を指定したクライアントから実行する UAP に対しては，バックエンドサーバ接続保持機能を適用しません。それ以外の UAP に対してはバックエンドサーバ接続保持機能が適用されます。

(b)　バックエンドサーバ接続保持時間の設定

バックエンドサーバ接続保持機能を使用する場合は，**バックエンドサーバ接続保持時間**を指定してください。バックエンドサーバ接続保持時間を超えると，次に実行されるトランザクションの終了時にフロントエンドサーバとバックエンドサーバの接続を切断します。

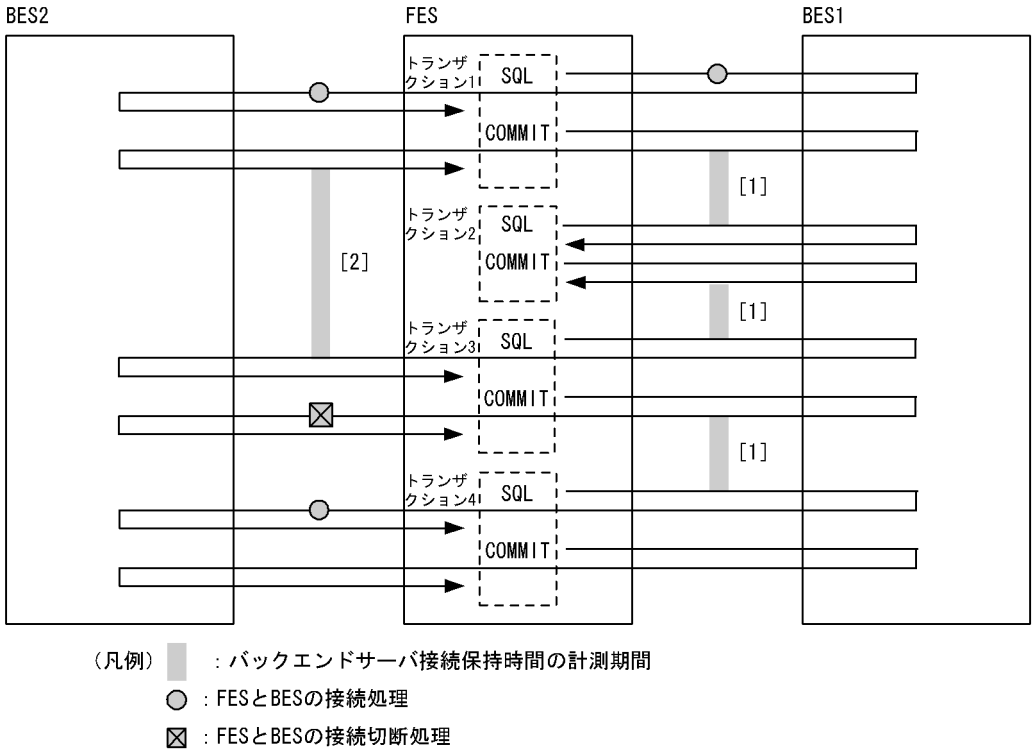
バックエンドサーバ接続保持時間の計測期間

バックエンドサーバ接続保持時間の計測開始時点と計測終了時点を次に示します。

- 計測開始時点：トランザクション処理が終了した時点
- 計測終了時点：次のトランザクション処理を開始した時点

バックエンドサーバ接続保持時間の計測期間と HiRDB の処理を次の図に示します。

図 E–2　バックエンドサーバ接続保持時間の計測期間と HiRDB の処理



〔説明〕

1. ［1］の期間はバックエンドサーバ接続保持時間以内のため，FES と BES1 の接続を保持します。
2. ［2］の期間はバックエンドサーバ接続保持時間を超えたため，トランザクション（トランザクション 3）の終了時に FES と BES2 の接続を切断します。次に実行されるトランザクション（トランザクション 4）の開始時に，FES と BES2 の接続処理が発生します。

注意事項

バックエンドサーバ接続保持時間を超えた時点で，フロントエンドサーバとバックエンドサーバの接続切断処理が発生するわけではありません。その点を間違えないようにしてください。バックエンドサーバ接続保持時間を超えたときの HiRDB の処理についてまとめると，次のようになります。

1. バックエンドサーバ接続保持時間を超えたとき

   HiRDB の処理は特にありません。
2. トランザクションの開始時

   フロントエンドサーバとバックエンドサーバの接続は保持されているため，フロントエンドサーバとバックエンドサーバの接続処理は発生しません。
3. トランザクションの終了時

   フロントエンドサーバとバックエンドサーバの接続切断処理が発生します。
4. 次回のトランザクションの開始時

   フロントエンドサーバとバックエンドサーバの接続処理が発生します。

バックエンドサーバ接続保持時間の目安

- UAP の接続時間が短いシステムの場合は，バックエンドサーバ接続保持時間に 0 を指定してください。
- 常時接続，UAP の接続時間が長いシステムの場合は，バックエンドサーバ接続保持時間を指定しないで，省略値（1 秒）を使用してください。

バックエンドサーバ接続保持時間に 0 を指定した場合の動作

バックエンドサーバ接続保持時間に 0 を指定した場合は，時間監視を行わないで，フロントエンドサーバとバックエンドサーバの接続を保持し続けます。ただし，「接続保持が解除される場合」で示す場合は，接続を切断します。

バックエンドサーバ接続保持時間の指定方法

バックエンドサーバ接続保持時間は，次に示すどちらかのオペランドで指定します（両方指定してもかまいません）。

- pd_bes_conn_hold_trn_interval オペランド
- クライアント環境定義の PDBESCONHTI オペランド

pd_bes_conn_hold_trn_interval オペランドで指定したバックエンドサーバ接続保持時間は，全 UAP に対して適用されます。クライアント環境定義の PDBESCONHTI オペランドで指定したバックエンドサーバ接続保持時間は，PDBESCONHTI オペランドを指定したクライアントから実行する UAP に対してだけ適用されます。なお，二つのオペランドの優先順位は 2，1 の順になります。オペランドの組み合わせの例を次に示します。

・pd_bes_conn_hold_trn_interval = 10 を指定した場合

全 UAP に対してバックエンドサーバ接続保持時間が 10 秒になります。

・PDBESCONHTI = 10 を指定した場合

このオペランドを指定したクライアントから実行する UAP に対してバックエンドサーバ接続保持時間が 10 秒になります。

・pd_bes_conn_hold_trn_interval = 10，PDBESCONHTI = 20 を指定した場合

PDBESCONHTI オペランドを指定したクライアントから実行する UAP に対しては，バックエンドサーバ接続保持時間が 20 秒になります。それ以外の UAP に対してはバックエンドサーバ接続保持時間が 10 秒になります。

(c)　クライアントの最大待ち時間の設定

接続保持中のバックエンドサーバのユニットが異常終了した場合，そのバックエンドサーバで次に実行されるトランザクションがエラー，又は無応答状態になることがあります。また，トランザクションキューイング機能を使用している場合，pd_ha_trn_restart_retry_time によるリトライ処理が行われないでエラー，又は無応答状態になることがあります（計画系切り替えのときも同様です）。このため，クライアント環境定義の PDCWAITTIME オペランドでクライアントの最大待ち時間を指定して，トランザクションがエラー，又は無応答状態になったときに備えてください。PDCWAITTIME オペランドの値を超えても応答がない場合はトランザクションがキャンセルされます。

(d) バックエンドサーバプロセス数の設定

バックエンドサーバ接続保持機能を使用する場合，個々のバックエンドサーバプロセス数をシステム内の全フロントエンドサーバプロセス数より大きくしてください。そのためには，システム定義のオペランドの指定値を次に示す条件式を満たすように設定してください。

pd_max_bes_process オペランドの値≧

pd_max_users オペランドの値×フロントエンドサーバ数

この条件式を満たさない場合，バックエンドサーバプロセス数が不足し，メッセージキュー監視機能によって HiRDB（HiRDB/パラレルサーバの場合はユニット）が異常終了したり，SQL がエラーになったりすることがあります。

また，UAP が動いている状態でユティリティを実行する場合，バックエンドサーバプロセス数にはユティリティの分の余裕値を確保してください。

(4) 注意事項

(a) 定義系 SQL を実行しないでください

定義系 SQL を実行する UAP に対しては，バックエンドサーバ接続保持機能を使用しないでください。定義系 SQL を実行する UAP に対してこの機能を使用すると，フロントエンドサーバとバックエンドサーバ間，ディクショナリサーバとバックエンドサーバ間の接続が同時に発生することがあるため，バックエンドサーバプロセス数が不足し，SQL エラーになることがあります。したがって，定義系 SQL を実行する UAP のクライアント環境定義の PDBESCONHOLD オペランドには NO を指定してください。

(b) 接続保持が解除される場合

バックエンドサーバ接続保持機能を使用している場合でも，次に示す場合はフロントエンドサーバとバックエンドサーバの接続を切断します。

- DISCONNECT 文（HiRDB XA ライブラリの使用時は xa_close），PDCWAITTIME オーバによって，フロントエンドサーバとクライアント間の接続が切断された場合
- ロールバックした場合（内部ロールバックを含む）
- 内部的に多量のメモリを使用した場合
- pdpfresh コマンド（-f オプションなし）を実行した場合
- ホールダブルカーソルのカーソルをクローズした場合
- SET SESSION AUTHORIZATION 文を実行した場合

(c) バックエンドサーバ接続保持時間を超えても接続が切断されないことがあります

バックエンドサーバ接続保持時間を超えても，次に示す場合はフロントエンドサーバとバックエンドサーバの接続を切断しません。

- ホールダブルカーソルを使用している場合
- UAP 環境定義で指定したローカルバッファを使用している場合

付録 F　HiRDB Control Manager との連携

ここでは，HiRDB Control Manager - Console の起動方法と，管理 HiRDB の登録方法について説明します。

付録 F.1　HiRDB Control Manager - Console の起動方法

〈手順〉

1. **［スタート］－［プログラム］－［HiRDB CM - Console］－［HiRDB CM - Console］**を選択し，ログイン画面を表示します。

   ［パスワード］欄に，インストール時に設定したパスワードを入力します。

   ［コンソール ID］欄には，コンピュータ名が表示されます。



2. **［OK］**をクリックします。

付録 F.2　管理 HiRDB の登録方法

〈手順〉

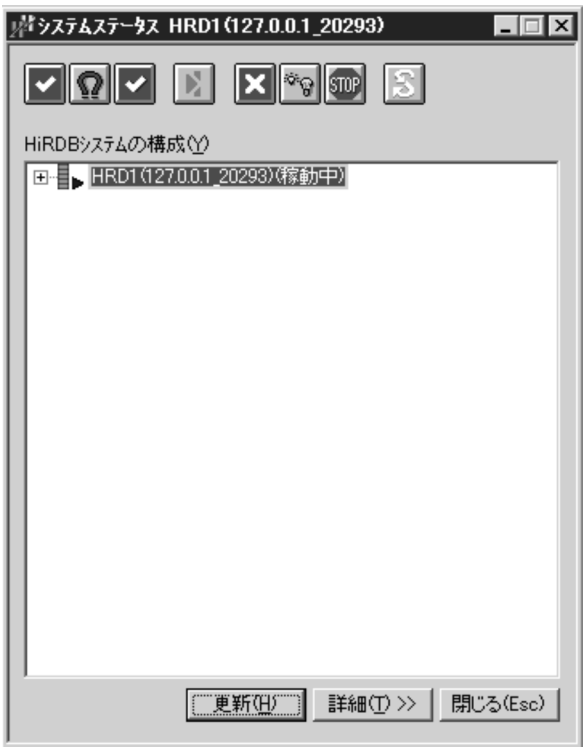
1. HiRDB Control Manager - Console を起動します。
2. タブメニューの**［管理 HiRDB］－［管理 HiRDB の新規登録］**を選択して，［管理 HiRDB の新規登録］画面を表示します。

   管理したい HiRDB サーバに対応する HiRDB Control Manager - Agent の IP アドレスとポート番号を入力します。



3. **[OK]** をクリックします。

［システムステータス］画面に，登録した HiRDB サーバが表示されます。



付録 G　論理セクタ長が 4096 バイトのディスク使用時における運用

ここでは，論理セクタ長が 4096 バイトのディスクを使用する場合の運用について説明します。

付録 G.1　ディスクの論理セクタ長の確認方法

ディスクには論理的なセクタ長と物理的なセクタ長があります。HiRDB の処理に関係するのは論理セクタ長になります。以下のコマンドで確認できます。

fsutil コマンドで確認します。以下は実行例で，実行結果の一部を抜粋しています。”セクターあたりのバイト数”が論理セクタ長になります。

```
> fsutil fsinfo ntfsinfo c:
......
セクターあたりのバイト数                : 4096
物理セクターあたりのバイト数            : 4096
クラスターあたりのバイト数              : 4096
ファイル セグメントあたりのバイト数     : 1024
ファイル セグメントあたりのクラスター数 : 0
......
```

付録 G.2　適用範囲

論理セクタ長が 4096 バイトのディスクを使用する場合，次の条件に合致するときに，「運用方法」 で説明する運用が必要になります。以降の説明で「論理セクタ長が 4096 バイトのディスク」とは，次の条件に合致する場合です。なお，次の条件に合致しない場合には適用する必要がありませんが，適用しても問題ありません。

1. 次のどれかの条件に合致する。
   - ダイレクトディスクアクセスを使用して，論理セクタ長が 4096 バイトのディスクに HiRDB ファイルシステム領域を作成する
   - システム定義 pd_ntfs_cache_disable オペランドに Y を指定し，論理セクタ長が 4096 バイトのディスクに HiRDB ファイルシステム領域を作成する
   - 論理セクタ長が 4096 バイトのディスクに使用目的が UTL の HiRDB ファイルシステム領域を作成する

付録 G.3　運用方法

論理セクタ長が 4096 バイトのディスクに対して HiRDB ファイルシステム領域を作成し，運用する方法について説明します。

(1)　HiRDB ファイルシステム領域の作成方法

pdfmkfs コマンドの-s オプションに 4096 を指定してください。詳細はマニュアル「HiRDB コマンドリファレンス」を参照してください。

「適用範囲」が必要になる条件に合致する場合，HiRDB ファイルシステム領域のセクタ長を必ず 4096 バイトにする必要があります。指定しないで作成すると，KFPI21610-E メッセージを出力し，コマンドがリターンコード 12 でエラー終了します。

(2)　HiRDB ファイルの作成方法

1. RD エリア

   セクタ長が 4096 バイトの HiRDB ファイルシステム領域に RD エリアを配置する場合，RD エリアを作成するときに指定するページ長(create rdarea 制御文または initialize rdarea 制御文の page オペランド)には 4096 の倍数にしてください。ページ長の最大は，28672 になります。

2. システムログファイル

   セクタ長が 4096 バイトの HiRDB ファイルシステム領域にシステムログファイルを作成する場合，レコード長は，4096 バイトにしてください。システム定義 pd_log_rec_leng オペランドに 4096 を指定して pdloginit コマンドで作成してください。pdloginit コマンドに-l オプションを指定している場合は，4096 を指定してください。

3. ステータスファイル

   セクタ長が 4096 バイトの HiRDB ファイルシステム領域にステータスファイルを作成する場合，レコード長は，4096 の倍数にしてください。pdstsinit コマンドに-l オプションを指定している場合は，4096 の倍数を指定するか，指定を省略してください。

4. 上記以外の HiRDB ファイル

   対応は不要です。HiRDB が HiRDB ファイルシステム領域のセクタ長を判断し，ページ長を決定します。

(3)　グローバルバッファの指定方法

1. 論理セクタ長が 4096 バイトのディスクに RD エリアを作成した場合，システム共通定義(pdsys)に以下の記述を追加してください。

   ```
   set pd_dbbuff_dev_sector_size = 4096
   ```

   オペランドを省略するか，又は 512 を指定した場合，論理セクタ長が 4096 バイトのディスクに作成した RD エリアへのグローバルバッファを介した I/O 処理でエラーが発生し，RD エリアが障害閉塞することがあります。

2. 一つのグローバルバッファに，論理セクタ長が 4096 バイトのディスクに作成した RD エリアと，それ以外の RD エリアを混在しないように定義してください。

   混在させた場合，論理セクタ長 4096 バイトのディスクに作成した RD エリアの I/O 処理でエラーが発生し，RD エリアが障害閉塞することがあります。

付録 G.4　注意事項

1. ディスクの換装などで，pdfmkfs コマンドの-s オプションに 4096 を指定しないで作成した HiRDB ファイルシステム領域を，論理セクタ長が 4096 バイトのディスクへ OS のコマンドなどを使用して直接移動した場合，その HiRDB ファイルシステム領域は使用できません。
2. ページ長が 4096 の倍数でない RD エリア及び HiRDB ファイルを，論理セクタ長が 4096 バイトのディスクへ移動する場合，移動対象がシステムファイルのときは，pdstsinit コマンド及び pdloginit コマンドで作成し直してください。

   RD エリアの場合，データベース再編成ユティリティ（pdrorg -k unld）でデータを退避し，データベース構成変更ユティリティ（pdmod）の initialize rdarea 文でページ長を 4096 の倍数に変更後にデータを入れ直してください。なお，論理セクタ長が 4096 バイトのディスクに RD エリアを作成する場合は，RD エリアのページの最大長は 28672 になります。そのため，行長が 28626 以上の表のデータは移動できません。
3. データベース複写ユティリティ（pdcopy）及び pdfbkup コマンドでのバックアップの取得と，データベース回復ユティリティ（pdrstr）及び pdfrstr コマンドでのバックアップからの回復の可否を次の表に示します。

RD エリアのページ長	HiRDB ファイルシステム領域のセクタ長		回復可否
	バックアップ取得先	バックアップからの回復先	
4096 の倍数	512,1024	512,1024	回復できます
		4096	回復できます
	4096	512,1024	回復できます
		4096	回復できます
4096 の倍数 でない	512,1024	512,1024	回復できます
		4096	回復できます
	4096	512,1024	該当しません（バックアップ先に作成できません）
		4096	該当しません（バックアップ先に作成できません）

4. 論理セクタ長が 4096 バイトのディスクでない場合に「運用方法」の操作を行っても問題はありません。
5. pdwork に指定した HiRDB ファイルシステム領域のセクタ長を変更する場合は，HiRDB が停止した状態で行ってください。

索引

記号

-c オプション　1650
-M オプション〔pdcopy コマンド〕　234

数字

1：1 系切り替え構成の例〔HiRDB/シングルサーバ〕　1594
1：1 系切り替え構成の例〔HiRDB/パラレルサーバ〕　1596
1：1 スタンバイレス型系切り替え機能　1573
1:1 スタンバイレス型系切り替え時のシステム用 RD エリアを含む回復方法　1189
24 時間連続稼働　1943

A

actcommand オペランド〔Hitachi HA Toolkit Extension〕　1703
alter HiRDB mode to parallel 文　490
ALTER INDEX〔インデクス名の変更〕　814
ALTER PROCEDURE　980
alter rdarea 文〔RD エリアのオープン契機の変更〕　901
alter rdarea 文〔RD エリア名の変更〕　895
ALTER ROUTINE　980
ALTER TABLE〔表定義の変更〕　626
ALTER TABLE〔列の削除〕　623
ALTER TABLE〔列の追加〕　615
ASSIGN LIST 文　789

B

B-tree インデクス　812

C

CLUSTERPRO に関する準備　1705
CONNECT/DISCONNECT に関する統計情報　1203
CONNECT 関連セキュリティ機能の運用　1526
CONNECT 権限　76
CONNECT 権限〔付与方法〕　83, 84
CONNECT 権限を取り消す　87
Cosminexus と連携した PRF トレース情報の取得　1898
CPU 障害が発生したとき　1058
CREATE AUDIT　1300
create auditor 文　1297
create audit table 文　1299
CREATE CONNECTION SECURITY　1535, 1548
CREATE FUNCTION　978
CREATE PROCEDURE　978
create rdarea 文　842
C ストアドファンクション　993
C ストアドプロシジャ　993
C ライブラリファイルに障害が発生した場合の対処　997
C ライブラリファイルの再登録　995
C ライブラリファイルの削除　995
C ライブラリファイルの新規登録　994

D

DataKeeper ジョブ，及び DataKeeper ボリュームリソースの作成　1716
DataKeeper ボリュームリソースの登録　1720
DBA 権限　76
DBA 権限〔付与方法〕　82
DBA 権限を取り消す　87
DB メンテナンス予定日　603
DECIMAL 型の符号正規化機能　800
DECIMAL 型の符号部を変換する　800
DELETE 権限　77
DELETE 文　614
DROP AUDIT　1300
DROP CONNECTION SECURITY　1549
DROP DATA TYPE　787
DROP FUNCTION　981
DROP INDEX　815
DROP PROCEDURE　981
DVD-RAM ライブラリ装置の運用　1936

E

EMPTY オプション 812
expand rdarea 文 848

F

FIX ハッシュ分割〔ハッシュ関数の変更〕 751
FIX 表に列を追加する 615, 1926
FORGETTING 状態のトランザクション 1055

G

globalbuffer オペランド〔create rdarea 文〕 843
GRANT AUDIT 1304
GRANT 文 82

H

HA グループ 1579
HDP 機能 447
HiRDB/パラレルサーバへの移行 490
HiRDB Advanced High Availability 381, 390, 642
HiRDB CM 498
HiRDB Control Manager を使用してデータベースの状態解析をする場合 567
HiRDB Control Manager - Console の起動方法 1958
HiRDB Datareplicator〔HiRDB システム定義の変更〕 379
HiRDB Datareplicator〔システム構成変更コマンド〕 385
HiRDB Datareplicator〔システムログファイルの障害〕 1036
HiRDB Datareplicator〔表の再編成〕 579
HiRDB Non Recover FES 1950
HiRDB Text Search Plug-in 831
HiRDB が終了できないときの対処方法〔接続中のユーザが存在するため〕 1060
HiRDB システム定義の作成〔系切り替え機能〕 1631
HiRDB システム定義の指定〔24 時間連続稼働〕 1944
HiRDB システム定義の変更 67, 378
HiRDB システム定義の変更〔システム構成変更コマンド〕 381
HiRDB システム定義ファイルに障害が発生したとき 1051
HiRDB 接続時のパスワード秘匿化機能 1911
HiRDB の開始及び終了方法 38
HiRDB の終了方法〔系切り替え機能〕 1760
HiRDB ファイルシステム領域障害時の対処方法 1140
HiRDB ファイルシステム領域に関する情報 438
HiRDB ファイルシステム領域の回復 442
HiRDB ファイルシステム領域の作成 439
HiRDB ファイルシステム領域のバックアップ 441
HiRDB ファイルの削除 443
HiRDB を開始するときの手順〔ステータスファイルの障害時〕 1043
Hitachi HA Toolkit Extension に関する準備 1701
Hitachi HA Toolkit Extension のサービスプロセス 1758

I

initialize rdarea 文 851
INSERT 権限 77
IP アドレス 1066
IP アドレスによる接続制限 1531
IP アドレスによる接続制限の運用例 1558
IP アドレスによる接続制限の設定方法 1554
IP アドレスの構成例 1611
IP アドレスを引き継がない場合 1612, 1616, 1620
IP アドレスを引き継ぐ場合 1611, 1615, 1618

J

JAR ファイルに障害が発生した場合 992
JAR ファイルの操作 992
JAR ファイルを格納するディレクトリ 990
Java Runtime Environment のルートディレクトリ 991
Java オプション 991
Java 仮想マシンの位置づけ 986
Java 仮想マシンの標準・標準エラー出力の設定先ファイル 991
Java 仮想マシンのライブラリディレクトリ 991
Java クラスパス 991
Java ストアドファンクション 984

Java ストアドプロシジャ 984
Java ストアドプロシジャ，Java ストアドファンクションを使用できる環境 984
Java ストアドプロシジャ及び Java ストアドファンクション〔環境設定手順〕 990
JP1/AJS2-SO 498
JP1/AJS3 498
JP1/NETM/Audit 用監査ログ出力ファイル 1514
JP1/NETM/Audit 用監査ログ出力ユティリティ 1513
JP1/VERITAS NetBackup Agent for HiRDB License 233

K

KFPA20009-W 352
KFPS00888-W 368
KFPS00889-E 368
KFPS00992-E 1067
KFPS01861-E 1927
KFPS01910-I 300
KFPS02179-I 1229
KFPS04665-W 389
KFPS05120-W 1060

L

LOB 列構成基表〔表の再編成〕 577
LVM 1138

M

MIB 1843
MIB 環境定義ファイル 1845
MIB 環境定義ファイルを作成する 1850
MIB コマンド 1843
MIB 定義ファイル 1844, 1852
MIB 定義ファイルを登録する 1851
MIB パフォーマンス情報監視機能の運用 1841
move rdarea 文 931
MSFC 及び DataKeeper に関する準備 1716
MSFC に関する準備（Windows Server 2012/R2 以降の場合） 1688
MSFC に関する準備（Windows Server 2012 の場合） 1676

N

name オペランド〔Hitachi HA Toolkit Extension〕 1701
NetBackup 連携機能 233

O

OLTP システムとの連携中に障害が発生したとき 1054
OpenTP1 と連携している場合〔統計情報〕 1199
OS が異常終了したとき 1053
OS ログインユーザの簡易認証機能 94
OTHERS 指定の RD エリア〔RD エリアの統合〕 692
OTHERS 指定の RD エリア〔RD エリアの分割〕 678

P

PASSWORD_TEST 列を検索する 1543
PCTFREE オペランド〔CREATE INDEX〕 819
pd_aud_async_buff_count オペランド 1295
pd_aud_async_buff_retry_intvl オペランド 1295
pd_aud_async_buff_size オペランド 1295
pd_aud_auto_loading オペランド 1295
pd_aud_file_name オペランド 1294
pd_aud_file_wrn_pnt オペランド 1295
pd_aud_max_generation_num オペランド 1295
pd_aud_max_generation_size オペランド 1295
pd_aud_no_standby_file_opr オペランド 1295
pd_audit オペランド 1294, 1301
pd_auth_cache_size オペランド 1242
pd_bes_conn_hold_trn_interval オペランド 1955
pd_bes_connection_hold オペランド 1953
pd_command_deadlock_priority 412
pd_crypto_service_port 1637
pd_db_io_error_action オペランド 1130
pd_dbbuff_modify オペランド 390
pd_dbsync_point オペランド 1226
pd_deadlock_priority_use 412
pd_dec_sign_normalize オペランド 801
pd_down_watch_proc オペランド 371

pd_ha 1635
pd_ha_acttype 1635
pd_ha_agent 1635
pd_ha_agent オペランド 1806
pd_ha_ipaddr_inherit 1635
pd_ha_max_act_guest_servers 1637
pd_ha_max_server_process 1637
pd_ha_mgr_rerun 1637, 1835, 1836
pd_ha_resource_act_wait_time 1637
pd_ha_server_process_standby 1635
pd_ha_server_process_standby オペランド 1805
pd_ha_switch_timeout 1635
pd_ha_transaction 1635
pd_ha_trn_queuing_initial 1635
pd_ha_trn_queuing_wait_time 1635
pd_ha_trn_restart_retry_time 1635
pd_ha_unit 1636
pd_java_archive_directory オペランド 990
pd_lck_deadlock_check オペランド 319
pd_lck_deadlock_info オペランド 320
pd_lck_deadlock_sql_info オペランド 320
pd_lck_wait_timeout オペランド 320
pd_list_initialize_timing オペランド 791
pd_log_auto_expand_size 180
pd_log_auto_unload_path オペランド 160
pd_log_remain_space_check オペランド 168
pd_log_sdinterval オペランド 206
pd_log_unload_check オペランド 122, 132, 138
pd_max_ard_process 1637
pd_max_ard_process オペランド 1251
pd_max_bes_process オペランド 396
pd_max_dic_process オペランド 396
pd_max_users オペランド 396
pd_mlg_file_size 300
pd_mlg_msg_log_unit 301
pd_mode_conf 1636
pd_mode_conf オペランド 40
pd_plugin_ixmk_dir オペランド 833
pd_process_count オペランド 73, 397, 1250
pd_queue_watch_timeover_action オペランド 368
pd_queue_watch_time オペランド 368
pd_rdarea_extension_file オペランド 911, 921, 923
pd_rdarea_extension_timing オペランド 913
pd_rdarea_warning_point オペランド 915
pd_redo_allpage_put 1138
pd_reduced_check_time オペランド 1079
pd_registered_port オペランド 407
pd_registry_cache_size オペランド 1248
pd_rorg_predict 605
pd_routine_def_cache_size オペランド 1246
pd_space_level オペランド 796
pd_spd_syncpoint_skip_limit オペランド 346, 349
pd_spool_cleanup_interval 1014
pd_spool_cleanup_interval_level 1014
pd_spool_cleanup_level オペランド 1014
pd_spool_cleanup オペランド 1014
pd_sql_object_cache_size オペランド 1242
pd_start_level オペランド 1079
pd_start_skip_unit オペランド 1081
pd_stj_file_size オペランド 1194
pd_table_def_cache_size オペランド 1241
pd_trn_rerun_branch_auto_decide オペランド 1055
pd_type_def_cache_size オペランド 1245
pd_view_def_cache_size オペランド 1241
pdacunlck コマンド 1553
pdaudbegin コマンド 1301
pdaudend コマンド 1301
pdaudload 1295
pdaudput コマンド 1513
pdaudput 用管理ファイル 1519
pdaudrm コマンド 1301
pdaudswap コマンド 1302
PDBESCONHOLD オペランド 1953
PDBESCONHTI オペランド 1955
pdbkupls コマンド 232
pdbuffer オペランド〔-c オプション〕 1650
pdbufmod コマンド 392
pdcancel コマンド 1026

pdcat コマンド　299
pdchgconf コマンド　381
pdclibsync コマンド　995
PDCLTAPNAME オペランド　1065
pdcltgrp オペランド　403
PDCLTGRP オペランド　403
pdcspool コマンド　1014
PDCWAITTIME　366
PDCWAITTIME オーバ　371
PDCWAITTIME オペランド　74
pddbfrz コマンド　232, 279
PDDBLOG オペランド　290
pddbls コマンド　314, 840
pddefrev コマンド　788, 817, 982
pdexp コマンド〔ストアドプロシジャの移行〕　555
pdfbkup コマンド　441
pdfls コマンド　438
pdfmkfs コマンド　439
pdfmkfs コマンドに指定するオプションの目安　160
pdfrm コマンド　443
pdfrstr コマンド　442
pdfstatfs コマンド　438
pdhagroup　1637
pdinfoget コマンド，pdosinfoget コマンドの実行方法　1012
pdjarsync コマンド　992
pdjarsync -S コマンド　992
pdlistls コマンド　790
pdlogchg -z コマンド　113
pdls -d aud コマンド　1302
pdls -d mem コマンド　315
pdls -d prc コマンド　310
pdls -d svr コマンド　313
pdmlgput オペランド　304
pdmod コマンド〔HiRDB/パラレルサーバへの移行〕　490
pdmod コマンド〔HiRDB ファイルの削除〕　443
pdmod コマンド〔RD エリアの移動〕　931
pdmod コマンド〔RD エリアのオープン契機の変更〕　901
pdmod コマンド〔RD エリアの拡張〕　848
pdmod コマンド〔RD エリアの再初期化〕　851
pdmod コマンド〔RD エリアの削除〕　907
pdmod コマンド〔RD エリアの追加〕　842
pdmod コマンド〔RD エリア名の変更〕　895
PDPLGIXMK オペランド　833
pdrbal コマンド　634
pdreclaim コマンド　964
pdrorg コマンド〔表の再編成〕　572
pdrstr コマンド　1144
PDSPACELVL オペランド　796
pdstart コマンド　39
pdstart コマンドを実行するサーバマシン　40
pdstart -a コマンド　56
pdstart -i コマンド　54
pdstart -r コマンド　54
pdstart -r コマンドで開始した場合に障害が発生したときの対処　55
pdstbegin コマンド　1197
pdstjsync コマンド　1197
pdstop コマンド　48
pdstop コマンドを実行するサーバマシン　49
PDUSER オペランド　86
Polkcrt〔アボートコード〕　72
PRF トレース機能　1896
PRF トレース機能の概要　1896
PRF トレース機能の取得レベル　1897
PRF トレース情報の取得内容　1900
PRF トレース情報ファイル　1016, 1897, 1898
PURGE TABLE 文　614

R

RD エリアアクセス時のページ不正検知時の動作　975
RD エリア詳細テーブル　1860
RD エリア単位に回復する場合　1158, 1159, 1166, 1168, 1171, 1185
RD エリア単位に回復する場合〔HiRDB Control Manager〕　1164

RD エリア単位にバックアップ取得時点に回復する場合 1182
RD エリア単位の再編成 573
RD エリア単位の再編成〔非分割キーインデクスが作成されない例〕 574
RD エリア単位の再編成〔例題〕 584
RD エリア単位の状態解析 566
RD エリア単位のバックアップの取得方法 258
RD エリア単位の物理的解析情報 1255
RD エリア単位の論理的解析情報 1256
RD エリアテーブル 1858
RD エリア内のデータ格納効率 566
RD エリアの移動 931
RD エリアのオープン契機 901
RD エリアのオープン契機を変更する方法 901
RD エリアの回復に関する質問 1933
RD エリアの拡張 848
RD エリアの拡張〔24 時間連続稼働〕 1946
RD エリアの再初期化 851
RD エリアの再初期化〔レジストリ用 RD エリア又はレジストリ LOB 用 RD エリア〕 853
RD エリアの残容量 840
RD エリアの自動増分 841, 910
RD エリアの自動増分を適用する場合の運用 915
RD エリアの状態 314
RD エリアの属性変更〔RD エリア名〕 895
RD エリアの属性変更〔オープン契機〕 901
RD エリアの追加 842
RD エリアの入出力エラーが発生したときの対処方法 1129
RD エリアの入出力エラーが発生した場合〔系切り替え機能〕 1822
RD エリアのバックアップ閉塞 272
RD エリアのページ破壊を検知したときの対処方法1127
RD エリアの容量不足 808, 809, 840
RD エリア名を変更する方法 895
RD エリア利用権限 77
RD エリア利用権限〔付与方法〕 83
RD エリア利用権限を取り消す 88
RD エリアを移動する方法 931
RD エリアを格納している HiRDB ファイルシステム領域を移動する方法 953
RD エリアを削除する方法 907
RD エリア–HiRDB ファイルシステム領域テーブル1868
RECOVERY オペランド 291
remove rdarea 文 907
REVOKE 文 87
RPC トレースファイル 1015

S

SELECT 権限 77
Service Complete 1671
servmax オペランド〔Hitachi HA Toolkit Extension〕 1701
SNMP 1843
SNMP エージェント 1844
SQL オブジェクト実行に関する統計情報 1194
SQL オブジェクト取得要求回数 1244
SQL オブジェクト長 1245
SQL オブジェクト転送に関する統計情報 1194
SQL オブジェクトバッファから出された SQL オブジェクトの数 1244
SQL オブジェクト用バッファから出されたストアドプロシジャのオブジェクトの数 1244
SQL オブジェクト用バッファ中の SQL オブジェクト数 1244
SQL オブジェクト用バッファ中のストアドプロシジャのオブジェクト数 1244
SQL オブジェクト用バッファ長のチューニング 1242
SQL オブジェクト用バッファヒット回数 1244
SQL 実行時間警告出力機能 352
SQL 実行時間警告情報の出力形式 358
SQL 実行時間警告情報ファイル 352
SQL 静的最適化に関する統計情報 1193
SQL 定義ファイル 554
SQL 動的最適化に関する統計情報 1193
SQL に関する統計情報 1193
SQL のチューニング 1261
SQL 文の履歴に関する統計情報 1194
STANDBY 指定のときの運用方法〔リスト〕 791

SUSPEND 状態　56, 1928
switchtype オペランド〔Hitachi HA Toolkit Extension〕　1702
sysdef 定義ファイル〔Hitachi HA Toolkit Extension〕　1701
SYS 統計情報テーブル　1871

T

termcommand オペランド〔Hitachi HA Toolkit Extension〕　1703
TEST オプション　1542
TIMEOUT　1927
TPBroker for C++と連携している場合〔統計情報〕　1201
TUXEDO と連携している場合〔統計情報〕　1201

U

UAP が異常終了したとき　1027
UAP が終了しないとき　1026
UAP から HiRDB にアクセスする場合〔統計情報〕　1198
UAP 識別情報　322, 323, 326, 328
UAP に関する統計情報　1193
UAP の状態監視　346
UAP 又はユティリティの実行時間が長い場合　310
UAP を実行できないとき　1026
UNBALANCED SPLIT〔CREATE INDEX 又は CREATE TABLE〕　821
UPDATE 権限　77

W

WAIT 要因のユーザを調べる方法　312
Windows のサービスで HiRDB を終了する方法　51
WRITE 回数　1235
WRITE 単価最小　1234
WRITE 単価最大　1234
WRITE 単価平均　1235

あ

アカウントロック期間　1530
アカウントロック期間を確認する　1549
空きありセグメント　963
空きセグメント　964
空きページ解放ユティリティ　964
空き領域の再利用機能　1946
アクセス権限　77
アクセス権限〔監査証跡表〕　1303
アクセス権限〔付与方法〕　84
アクセス権限を取り消す　88
アプリケーションから HiRDB にアクセスする場合〔統計情報〕　1198
アボート情報ファイル　1015
アンバランスインデクススプリット　819
アンロード　117, 572
アンロード〔システムログファイル〕　113
アンロード状態のチェックを解除する運用　137
アンロード済み状態〔システムログファイル〕　110
アンロード済み状態のファイルをアンロードする　142
アンロード統計ログファイルを作成する　1198
アンロード待ち状態〔システムログファイル〕　111
アンロードレスシステムログ運用　113, 127, 1171
アンロードログファイル作成ディレクトリ　156
アンロードログファイル作成ディレクトリの作成　158

い

異常終了　48, 1016
異常終了〔注意事項〕　67
違反種別コード　1542
イベントサブタイプ　1335
イベントタイプ　1335
イベントログへのメッセージ出力抑止　303
インターバル解析〔再編成時期予測機能〕　609
インデクス情報ファイル〔プラグインインデクスの一括作成〕　836
インデクス単位の状態解析　566
インデクスに関する統計情報　1193, 1203
インデクスの一括作成　822
インデクスの運用　804
インデクスの格納効率　566, 805

インデクスの再作成　805
インデクスの再編成　805
インデクスの再編成中にエラーが発生した場合　807
インデクスの削除　815
インデクスのチューニング　1253
インデクスの定義系 SQL を作成する　817
インデクスページスプリット　818
インポート〔表定義情報と表データの移行〕　518

う

受け入れユニット　1579
受け入れユニットの開始　1748
上書きできない状態〔システムログファイル〕　111
上書きできない状態〔シンクポイントダンプファイル〕　201
上書きできる状態〔システムログファイル〕　110
上書きできる状態〔シンクポイントダンプファイル〕　201
上書きできる状態のファイルがない場合　208
運用コマンドが応答待ちタイムアウトになったとき　1028
運用コマンドを実行できないとき　1028
運用方法　1960
運用履歴表　603

え

影響分散スタンバイレス型系切り替え機能　1578
影響分散スタンバイレス型系切り替えのシステム構成例　1604
エクステント　918
エクスポート〔表定義情報と表データの移行〕　518
エラーログファイル　1015

お

オープン契機　901
オブジェクト格納用のデータディクショナリ LOB 用 RD エリア　1149
オブジェクト操作イベント　1280
オブジェクト定義イベント　1280

か

改竄防止オプション　80
改竄防止機能　80
改竄防止表　80
開始時の注意事項　62
開始処理時間を短縮する方法　73
開始処理に失敗したときの注意事項　64
開始できないときの対処方法　1029
開始に関する質問　1926
開始方法　39
開始方法（特別な開始方法）　54
開始方法〔系切り替え機能〕　1746
開始方法〔サーバ単位〕　60
開始方法のカスタマイズ　43
開始方法〔ユニット単位〕　57
開始又は終了処理が長い場合　313
開始モード　39
解析情報表　603
解析情報表及び運用履歴表を格納するデータディクショナリ用 RD エリア　604
回復処理が中断された場合〔同一ログの再使用チェック〕　1151
回復不要 FES（HiRDB/パラレルサーバ限定）　1950
回復不要 FES〔システム構成変更コマンド〕　387
回復方法〔pdcopy コマンド以外でバックアップを取得した場合〕　1181
解放途中セグメント　971
書き込み中　201
格納効率　565
確保した型定義キャッシュ長　1246
確保したルーチン定義キャッシュ長　1247
仮想ネットワーク名　1612
仮想ネットワーク名の IP アドレス〔MSFC〕　1681, 1693
型定義キャッシュ総使用領域長　1246
型定義キャッシュヒット回数　1246
型定義情報取得要求回数　1246
片方向代替構成　1601
稼働環境　377

簡易ダンプファイル　1015
環境設定〔C ストアドプロシジャ又は C ストアドファンクション〕　994
監査権限　76, 1273
監査証跡　1273
監査証跡データ登録 UOC　1324
監査証跡の絞り込み　1357
監査証跡の取得　1301
監査証跡のレコード項目〔イベント終了時〕　1443
監査証跡のレコード項目〔権限チェック時〕　1379
監査証跡表　1276
監査証跡表のアクセス権限　1303
監査証跡表のインデクス定義　1303
監査証跡表の再編成　1303
監査証跡表の削除　1303
監査証跡表の作成　1299
監査証跡表の自動データロード機能　1283
監査証跡表の操作　1303
監査証跡表の利用　1303
監査証跡表の列構成　1328
監査証跡表へのデータ登録　1302, 1324
監査証跡表を格納する RD エリアの作成　1298
監査証跡表を格納する RD エリアの操作　1301
監査証跡ファイル　1276
監査証跡ファイルに出力される情報　1285
監査証跡ファイルに障害が発生したときの対処方法　1366
監査証跡ファイルの運用　1314, 1672
監査証跡ファイルの削除　1301
監査証跡ファイルの作成　1314
監査証跡ファイルの状態　1319
監査証跡ファイルの状態確認　1302
監査証跡ファイルのスワップ　1302, 1320
監査証跡ファイルの名称　1314
監査証跡ファイルの容量変更　1322
監査証跡ファイル用の HiRDB ファイルシステム領域の作成　1296
監査証跡ファイル用の HiRDB ファイルシステム領域の操作　1302
監査対象イベント　1278
監査対象イベントの追加及び削除　1304
監査対象イベントの定義　1300
監査人　1273
監査人セキュリティイベント　1279
監査人の登録　1297
監査ログ〔JP1/NETM/Audit〕　1513
監査ログ管理サーバ〔JP1/NETM/Audit との連携〕　1514
監査ログ収集対象サーバ〔JP1/NETM/Audit との連携〕　1514
管理 HiRDB の登録方法〔HiRDB Control Manager〕　1958
管理フレームワーク　1844

き

基準値　613
基準値定義ファイル　613
基準値のカスタマイズ　613
機密保護　76
行削除禁止期間　80
強制開始　39
強制開始するときの注意事項　63
強制終了　48
強制終了〔注意事項〕　67
共有ディスク装置の準備　1628
共有ディスクのアクセス制御　1630
共有ディスクの登録〔MSFC〕　1680, 1692
共有モード〔リバランスユティリティ〕　635
共用メモリダンプファイル　1015
共用メモリの使用状況　315
切り替え先の指定（影響分散スタンバイレス型系切り替え機能限定）　1638

く

空白変換機能　793
空白変換レベル　793
クライアントアクセスポイントの登録〔MSFC〕　1681, 1693
クライアント環境定義の指定〔系切り替え機能〕　1645
クライアントグループの種別　1065

クライアントグループの接続枠保証機能　402
クライアントの最大待ち時間　352
クラスタキー及びクラスタリングデータページの格納状態解析　566
クラスタキーの格納効率　566
クラスタキーの格納状態解析　1259
クラスタサービスの登録〔MSFC〕　1685, 1697
クラスタシステム　1568
クラスタソフトウェア　1591
クラスタソフトウェアによる共有ディスクのアクセス制御　1630
クラスタリングデータページの格納効率　566
クラスタリングデータページの格納状態解析　1259
グループ ID　1065
グローバルバッファ数の上限　391
グローバルバッファ制御情報トレース取得機能　1023
グローバルバッファテーブル　1865
グローバルバッファに関する統計情報　1193
グローバルバッファの削除　393
グローバルバッファの追加　392
グローバルバッファの追加，削除，変更　68
グローバルバッファの定義〔1：1 スタンバイレス型系切り替え機能〕　1650
グローバルバッファの定義〔影響分散スタンバイレス型系切り替え機能〕　1653
グローバルバッファの定義変更　394
グローバルバッファの動的変更　390
グローバルバッファの動的変更〔24 時間連続稼働〕　1947
グローバルバッファの排他処理の排他競合待ち発生率　1223
グローバルバッファの変更　393
グローバルバッファの割り当て〔RD エリアの追加〕　843
グローバルバッファプールのチューニング　1209
グローバルバッファプールのヒット率（GBHIT）　1216
グローバルバッファプールのヒット率（HIT）　1209
グローバルバッファ用の共用メモリセグメント数の上限　392

け

系が切り替わった後の運用方法　1801
計画系切り替え　1790
計画停止　48
計画停止後の再開始〔注意事項〕　71
計画停止〔注意事項〕　67
系切り替え機能〔24 時間連続稼働〕　1949
系切り替え機能〔システム構成変更コマンド〕　386
系切り替え機能使用時のシステム構成例　1594
系切り替え機能の運用　1564
系切り替え機能の適用基準　1590
系切り替え後のサーバプロセスの割り当て　1641
警告値　168
系障害　1567
系の切り替え時間の比較　1566
系の切り替え時間を短縮する方法　1820
系の切り戻し　1574
契約ユーザ数　401
ゲスト BES　1578
ゲスト BES の開始　1748
ゲスト用領域　1579
権限管理イベント　1280
検索効率　565
検索条件定義ファイル　1515
検索性能　571
現用〔監査証跡ファイル〕　1320
現用系　1568
現用〔システムログファイル〕　110
現用〔ステータスファイル〕　218
現用のシステムログファイルのスワップ　70
現用ファイルに障害が発生したとき〔システムログファイル〕　1033
現用ファイルに障害が発生したとき〔ステータスファイル〕　1040
現用ファイルの両系障害〔システムログファイル〕　1036
現用ファイルの両系障害〔ステータスファイル〕　1048
現用ファイルの両系に障害が発生したため HiRDB を再開始できないときの対処方法　1048

現用ファイルをアンロードする場合　142
現用ファイルを変更する　143
現用ファイルを変更する〔ステータスファイル〕　224

こ

合計 WRITE 時間　1232
更新可能状態　280
更新可能バックアップ閉塞　273
更新可能バックアップ閉塞（WAIT モード）　273
更新可能モード〔バックアップ取得モード〕　234
更新凍結コマンド　232, 279
更新凍結状態　279
更新凍結状態の解除　281
更新バッファフラッシュ回数　1211, 1217
更新前ログ取得モード　235, 290
更新前ログ取得モードの運用　293
更新要求ヒット率　1209
更新ログ取得方式　235, 290
高速系切り替え機能　1805
高速系切り替え機能使用時のシステム構成例　1806
公用 RD エリア　77, 83
コマンド実行時の認証　1304
コマンドトレースファイル　1015
コマンドのデッドロックプライオリティ　411
コミット　1145
コミット２相目指示待ち状態　1054
コミット処理中に障害が発生したときのトランザクション決着種別の確認方法　1132
コンピュータ名を変更する方法　409

さ

サーバ稼働状態テーブル　1854
サーバ構成の変更　67
サーバごとのシステムの稼働に関する統計情報　1193
サーバ障害　1567
サーバ障害監視時間　1635
サーバ単位の開始方法　60
サーバ単位の終了方法　60
サーバの移動　483
サーバの削除　478
サーバプロセス　1017
サーバプロセス数の変更　396
サーバプロセス数を減少させる運用　398
サーバプロセスの異常終了回数の監視　371
サーバプロセスの異常終了が多発した場合〔系切り替え機能〕　1819
サーバプロセスの沈み込み　368
サーバプロセスの状態監視　368
サーバ名　1065
サーバモード　1566
サーバモードの場合のリソースの登録〔MSFC〕　1679, 1691
サービス実行中のサーバプロセス数　1249, 1250
サービスの開始方法の設定　1671
サービスの作成〔MSFC〕　1676
サービスの登録〔MSFC〕　1682, 1694
サービスプロセス〔Hitachi HA Toolkit Extension〕　1758
再開始　39, 1018
再開始処理中に障害が発生した場合〔システムログファイル〕　1034
再開始するときに引き継ぐ情報　1018
再開始できないとき　1030
最小許容バイト数の設定　1527
最新の同期点に回復する方法　1161
最大起動プロセス数のチューニング　1249
最大起動プロセス数を超えたサービス要求数　1249
最大起動プロセス数を指定するオペランド　396
最大同時要求バッファ面数　1221
最適化情報　999
最適化情報〔最適化情報パラメタファイルからの登録〕　1000
最適化情報収集レベル　1000
最適化情報〔データベースからの収集，登録〕　999
最適化情報の運用　998
最適化情報の回復　1003
最適化情報の更新　1002
最適化情報の削除　1004
最適化情報の収集，登録　1001

最適化情報パラメタファイル　1005
最適化情報パラメタファイルによる最適化情報の登録　1006
最適化情報パラメタファイルの作成方法　1005
最適化情報パラメタファイルへの最適化情報の退避　1006
最適化情報を運用するときの注意事項　1008
再編成　572
再編成〔RD エリア単位〕　573
再編成時期の予測データの解析　604
再編成時期の予測データの取得　603
再編成時期の予測データを取得する　606
再編成時期の予測をやめる場合　612
再編成時期予測機能　602
再編成時期予測のカスタマイズ　612
再編成時期を予測する　602
再編成〔スキーマ単位〕　574
再編成〔大量データ〕　578
再編成の実行単位　572
再編成〔表単位〕　573
再編成〔例題〕　580
再利用禁止　1528
作業表用 HiRDB ファイルシステム領域テーブル　1856
サブタイプの削除　787
差分インデクス機能　831
差分バックアップ　262
差分バックアップ管理ファイル　263
差分バックアップ管理ファイルがない場合の回復方法　1176
差分バックアップ管理ファイルの回復　271
差分バックアップ機能　232, 261
差分バックアップ機能〔データベースの回復方法〕　1173
差分バックアップグループ　262
差分バックアップの取得　261
差分バックアップの履歴情報ファイル　269
参照可能バックアップ閉塞　272
参照可能バックアップ閉塞（更新 WAIT モード）　272
参照可能モード〔バックアップ取得モード〕　234
参照権限〔ディクショナリ表〕　90
参照バッファフラッシュ回数　1212, 1217
参照要求ヒット率　1210
参照・更新不可能モード〔バックアップ取得モード〕　234

し

時系列リストの作成〔アンロードログファイル〕　192
資源種別　330
資源情報　330
資源情報の見方　336
資源の移行　511
時刻を変更する方法　420
システム管理者セキュリティイベント　1279
システム構成の変更　456
システム構成変更コマンド　381
システム構成変更コマンド〔24 時間連続稼働〕　1943
システム構成変更コマンド実行後の HiRDB の状態　384
システム構成変更コマンド実行時にエラーが発生した場合　387
システム構成変更コマンド実行時の制限事項　384
システム構成変更コマンドを実行できない環境　383
システム構成例〔系切り替え機能〕　1594
システムの稼働環境を知る方法　298
システムの稼働に関する統計情報　1193
システムの状態監視　298
システムファイルの削除　443
システムマネジャユニットに障害が発生したときの対処方法　1137
システムマネジャユニットの系切り替えに失敗するときの対処方法　1834
システムログからチューニング情報を取得する方法　1203
システムログのアンロード　113
システムログファイル〔HiRDB 管理者がすること〕　111
システムログファイルに関する質問　1919
システムログファイルに障害が発生したとき　115, 139, 1033
システムログファイルの空き容量監視機能　166
システムログファイルの空き率　167

システムログファイルの運用方法　115
システムログファイルの解放　113, 128, 132
システムログファイルの削除　146
システムログファイルの削除，変更　69
システムログファイルの残量警告機能　175
システムログファイルの自動拡張機能　178
システムログファイルの状態　111
システムログファイルの状態情報ファイル〔空き容量監視機能〕　172
システムログファイルの状態情報ファイル〔自動拡張機能〕　185
システムログファイルの状態遷移　147
システムログファイルの新規追加　144
システムログファイルの操作方法　141
システムログファイルの不足　1086
システムログファイルの容量を大きくする　143
システムログファイルのレコード長の変更方法　150
システムログファイルを操作するコマンド　115
システムログをアンロードする運用　117
実 READ 回数　1213, 1218
実 WRITE 回数　1213, 1218
実行系　1568
実行時間　1232
実行時間の監視〔UAP 又はユティリティ〕　366
実行モード〔リバランスユティリティ〕　635
自動開始　40, 1928
自動再接続機能　1817
自動データロード機能適用中に障害が発生したときの対処方法　1373
自動ログアンロード機能　117
自動ログアンロード機能〔障害発生時の対処方法〕163
自動ログアンロード機能の運用方法　156
シナリオ　498
シナリオテンプレート　498
シナリオ〔バックエンドサーバの負荷均等化〕　500
シナリオ変数の値　498
絞込み検索　789
修正版レベルのバージョンアップ　1950
終了時の注意事項　66
終了できないとき　1032
終了に関する質問　1929
終了方法　48
終了方法〔Windows のサービス〕　51
終了方法〔サーバ単位〕　60
終了方法〔ユニット単位〕　58
終了モード　48
主監査人のスキーマ削除　1304
縮退起動　41, 1029, 1030
縮退起動をするときの運用方法　1079
出力待ち発生回数　1220
手動開始　40
私用 RD エリア　77, 83
障害が発生したステータスファイルがある状態で HiRDB を開始するときの手順　1043
障害が発生したときに HiRDB 管理者がすること 1012
障害が発生したときに取得する情報　1015
障害が発生したときの対処方法　1011
障害時のスナップ　1015
障害調査のために必要な情報　1939
障害に関する質問　1924
障害発生時の調査資料　1924
障害閉塞　295, 1148
障害閉塞〔対処方法〕　1127
状態解析結果蓄積情報のリセット　612
状態の確認　1777
使用中空きセグメント　963
使用中空きセグメントの解放　969
使用中空きセグメントの解放〔24 時間連続稼働〕1946
使用中空きセグメントの再利用　969
使用中空きページ　963
使用中空きページが作成される処理　967
使用中空きページの解放　964
使用中空きページの解放〔24 時間連続稼働〕　1946
使用中空きページの再利用　964
使用中セグメント　963
使用中セグメント比率　1255
常駐プロセス数のチューニング　1250
常駐プロセス数を指定するオペランド　397

使用中ページ　963
使用中ページ比率　1255
使用中満杯ページ　963
初期設定するときの開始方法　54
ジョブネット　498
シンクポイント　201
シンクポイント時の更新要求ヒットによるデータベース書き込み処理代行回数　1225
シンクポイント時の参照要求ヒットによるデータベース書き込み処理代行回数　1224
シンクポイント出力同期制御情報取得機能　1024
シンクポイント出力ページ数　1222
シンクポイント処理時間のチューニング　1229
シンクポイント処理時間を短縮する方法　1236
シンクポイントダンプ　201
シンクポイントダンプに対応するシステムログファイル　212
シンクポイントダンプの取得間隔　206
シンクポイントダンプの取得間隔のチューニング 1239
シンクポイントダンプの有効化　112, 202
シンクポイントダンプファイルに関する質問　1921
シンクポイントダンプファイルに障害が発生したとき　1037
シンクポイントダンプファイルの削除　212
シンクポイントダンプファイルの状態　201
シンクポイントダンプファイルの状態が変わった場合　208
シンクポイントダンプファイルの状態遷移　215
シンクポイントダンプファイルの新規追加　210
シンクポイントダンプファイルの操作方法　208
シンクポイントダンプファイルの追加，削除，変更 69
シンクポイントダンプファイルの容量を大きくする 209
シンクポイントダンプファイルを操作するコマンド 204
シンクポイントダンプ有効化のスキップ回数監視機能　346

す

スキーマが削除できない場合　786
スキーマ単位に表を移行する場合　535
スキーマ単位の再編成　574
スキーマ単位の再編成〔例題〕　587
スキーマ定義権限　77
スキーマ定義権限〔付与方法〕　83
スキーマ定義権限を取り消す　87
スキーマの削除〔主監査人〕　1304
スキーマを削除する　785
スタンバイ型系切り替え機能　1568
スタンバイレス型系切り替え機能　1573
ステータスファイルに関する質問　1923
ステータスファイルに障害が発生したとき　1040
ステータスファイルの削除　225
ステータスファイルの削除，変更，初期化　68
ステータスファイルの実体　222
ステータスファイルの状態　218
ステータスファイルの状態遷移　227
ステータスファイルの状態を変える　224
ステータスファイルの情報を調べる　226
ステータスファイルの新規追加　224
ステータスファイルのスワップ　219
ステータスファイルの操作方法　221
ステータスファイルの容量を大きくする　223
ステータスファイルを操作するコマンド　219
ストアドファンクション　976
ストアドファンクションを削除する方法　981
ストアドファンクションを作成する方法　978
ストアドプロシジャ　976
ストアドプロシジャの移行手順　556
ストアドプロシジャの移行方法　555
ストアドプロシジャのオブジェクト長　1245
ストアドプロシジャのオブジェクトの SQL オブジェクト用バッファヒット回数　1244
ストアドプロシジャのオブジェクトの取得要求回数　1244
ストアドプロシジャのオブジェクトのリコンパイル回数　1245
ストアドプロシジャを削除する方法　981
ストアドプロシジャを作成する方法　978
スパースファイル　922
スプリット回数　1253

スリープ処理実行での，バッファの排他処理のスリープ処理実行回数の平均値　1224
スワップ先にできない状態〔システムログファイル〕　111
スワップ先にできる監査証跡ファイルがないためHiRDB が強制終了した場合　1367
スワップ先にできる状態〔システムログファイル〕 110
スワップ先にできる状態のファイルがない場合　141
スワップ先にできる状態のファイルを確認する　141
スワップ〔システムログファイル〕　112
スワップ〔ステータスファイル〕　219

せ

正規 BES　1573
正規 BES ユニット　1573
正規 BES ユニットに系を切り戻す場合　1793
正規 BES ユニットの開始　1747
正規 BES ユニットの終了　1761
正規 BES ユニットの待機状態の解除　1761
正規化　800
正規ユニット　1578
正規ユニットの開始　1748
正常運転に戻す方法　1079
正常開始　39
正常開始する場合　41
正常開始できないとき　1029
正常終了　48
正常終了する場合〔例題〕　49
性能に関する質問　1930
セキュア状態　1054
セキュリティ監査機能に関するオペランド　1294
セキュリティ監査機能の運用　1272
セキュリティ監査機能の環境設定　1294
セキュリティ監査機能〔バージョンアップ〕　1511
セグメントの状態　963
セッションセキュリティイベント　1279
接続ユーザ詳細ファイル　1016, 1060, 1064
接続ユーザ情報ファイル　1016, 1060, 1064
セットアップ識別子　1626
セットアップ識別子〔MSFC〕 1682, 1685, 1694, 1697
全 RD エリアを回復する場合　1156, 1161
全 RD エリアを回復する場合〔HiRDB Control Manager〕　1155
全 RD エリアを最新の同期点に回復する場合　1182
全 RD エリアをバックアップ取得時点に回復する場合　1181
占有資源種別　323, 324, 327, 329
占有資源情報　323, 324, 327, 329
占有モード〔リバランスユティリティ〕　635

そ

相互系切り替え構成の例〔HiRDB/シングルサーバ〕　1595
相互系切り替え構成の例〔HiRDB/パラレルサーバ〕　1597
相互代替構成　1601
ソース格納用のデータディクショナリ LOB 用 RD エリア　1149

た

待機　110
待機〔監査証跡ファイル〕　1320
待機系　1568
代替 BES　1573
代替 BES ユニット　1573
代替 BES ユニットに系を切り替える場合　1792
代替 BES ユニットの開始　1747
代替 BES ユニットの終了　1761
代替中　1574
代替中のグローバルバッファの割り当て方式　1650
代替部　1574
代替部の開始　1747
代替部の終了　1761
代替部の待機状態の解除　1761
退避コアファイル　1015
タイムアウト　1028
タイムアウトが発生した場合　318
タイムアウト情報　318
タイムアウト情報の出力内容　325
タイムアウト発生日時　326

大量データの再編成　578
多段系切り替え　1578
単一文字種の指定禁止　1528

ち

遅延一括作成　830
注意事項　1962
抽出側 HiRDB Datareplicator　114
抽出完了状態〔システムログファイル〕　110
抽出未完了状態〔システムログファイル〕　111
抽象データ型〔表の再編成〕　577
抽象データ型を削除する　787
チューニング　1208
チューニング〔SQL〕　1261
チューニング〔SQL オブジェクト用バッファ長〕 1242
チューニング〔インデクス〕　1253
チューニング〔グローバルバッファプール〕　1209
チューニング〔最大起動プロセス数〕　1249
チューニング〔常駐プロセス数〕　1250
チューニング情報　1193
チューニング情報の取得　1196, 1203, 1205
チューニング情報の取得方法　1192
チューニング〔シンクポイントダンプの取得間隔〕 1239
チューニング〔データベース〕　1255
チューニング〔デファードライト処理〕　1226
チューニング〔バッファ長〕　1241
チューニング〔非同期 READ プロセス数〕　1251
チューニング〔ビュー解析情報用バッファ長〕　1241
チューニング〔表定義情報用バッファ長〕　1241
チューニング〔プロセス数〕　1249
チューニング〔ユーザ権限情報用バッファ長〕　1242
チューニング〔ユーザ定義型情報用バッファ長〕 1245
チューニング〔ルーチン定義情報用バッファ長〕 1246
チューニング〔レジストリ情報用バッファ長〕　1248

つ

通信障害が発生したとき　1058

て

定義系 SQL を作成する　788
定義系 SQL を作成する方法　982
ディクショナリ搬出入ユティリティ〔ストアドプロシジャの移行〕　555
ディクショナリ表の再編成〔例題〕　593
ディクショナリ表の参照権限　90
停止処理完了の確認　51
ディスク障害が発生したとき　1083
ディスク障害が発生したとき〔システムログファイル〕　1035
ディスクの論理セクタ長の確認方法　1960
ディスクボリューム単位の並列度　1226
ディスク容量不足〔バックアップ取得時〕　231
データ長を変更する　626
データの移行　777
データの削除　614
データベース回復の概要　1144
データベース回復ユティリティ　1144
データベース構成変更ユティリティ〔HiRDB/パラレルサーバへの移行〕　490
データベース構成変更ユティリティ〔HiRDB ファイルの削除〕　443
データベース構成変更ユティリティ〔RD エリアの移動〕　931
データベース構成変更ユティリティ〔RD エリアのオープン契機の変更〕　901
データベース構成変更ユティリティ〔RD エリアの拡張〕　848
データベース構成変更ユティリティ〔RD エリアの再初期化〕　851
データベース構成変更ユティリティ〔RD エリアの削除〕　907
データベース構成変更ユティリティ〔RD エリアの追加〕　842
データベース構成変更ユティリティ〔RD エリア名の変更〕　895
データベース再編成ユティリティ〔表の再編成〕　572
データベース状態解析ユティリティからチューニング情報を取得する方法　1205

データベース操作に関する HiRDB ファイルの統計情報 1193
データベース定義ユティリティ〔表定義情報の移行〕 519
データベースの回復方法 1143
データベースの機密保護での認可識別子 99
データベースの更新ログ取得方式 235, 290
データベースの更新ログ取得方式による運用方法の違い 291
データベースの更新ログ取得方式の選択〔表の再編成〕 575
データベースの更新ログを取得しないときの運用 289
データベースの再編成〔24 時間連続稼働〕 1945
データベースの静止化 274
データベースのチューニング 1255
データベースをほかの HiRDB システムのバックアップから回復する方法 1178
データ連動用連絡ファイルに障害が発生したとき 1052
データロード済み〔監査証跡ファイル〕 1320
データロード待ち〔監査証跡ファイル〕 1320
テープ装置アクセス機能 233, 579, 1154
適用範囲 1960
デッドロック SQL 情報 318
デッドロック SQL 情報の出力内容 338
デッドロックが発生した場合 318
デッドロック監視処理時間を変更する方法 434
デッドロック情報 318
デッドロック情報及びタイムアウト情報への対処方法 320
デッドロック情報とデッドロック SQL 情報を突き合わせる方法 338
デッドロック情報の出力内容 321
デッドロックプライオリティ〔コマンド〕 411
デッドロックプライオリティ値 325
デッドロックプライオリティ値を変更する 411
デファードライト処理に関する統計情報 1193
デファードライト処理に関する統計情報の見方 1231
デファードライト処理のチューニング 1226
デファードライト処理用並列 WRITE プロセス数を多くする 1237
デファードライトトリガ時の更新ページの出力比率を高くする 1237
デファードライトトリガの実行間隔を短くする 1236
電源障害が発生したとき 1058

と

同一ログの再使用チェック〔データベースの回復方法〕 1151
同期点 1145
同期点行数 578
同期点指定の再編成 578
同期点指定の再編成〔ユティリティ異常終了時の対処方法〕 1122
統計ログからチューニング情報を取得する方法 1193
統計ログの取得対象サーバ 1194
統計ログの取得を開始する 1197
統計ログファイルに障害が発生したとき 1051
統計ログファイルの運用 1195
統計ログファイルの運用〔系切り替え機能〕 1779
統計ログファイルの作成 1194
統計ログファイルのスワップ 1195
統合機能 647
同時にバックアップを取得する必要がある RD エリア 236
特定境界値格納 RD エリア群数 706
特別な開始方法 54
トラブルシュート情報 1015
トラブルシュート情報の削除 1013
トラブルシュート情報の削除〔24 時間連続稼働〕 1948
トラブルシュート情報の出力量の削減 1014
トランザクションキャンセル時のプロセスダウンメッセージ変更機能 1019
トランザクションキューイング機能 1813
トランザクション決着種別の確認方法 1132
トランザクション識別子 322, 324, 327, 328
トランザクション第１状態 1054
トランザクション第２状態 1054
トランザクションの居残りが発生したときの対処方法 1057
トランザクションの回復方法 1054

に

二重ロックの解除　1563
入出力エラー〔対処方法〕　1129
入出力回数の確認　1931
認可識別子〔PDUSER〕　86
認可識別子の指定禁止　1527

は

バージョンアップ〔セキュリティ監査機能〕　1511
排他資源管理テーブル情報の出力内容　342
排他資源管理テーブル情報のファイル名　341
排他資源管理テーブル情報ファイル　1016
排他資源管理テーブル不足が発生した場合　341
排他制御モード　323, 324, 327, 328
排他制御用プールを分割する方法　414
排他〔バックアップ閉塞〕　275
排他待ち限界経過時間　320
排他待ち発生時刻　323, 325
パス障害　1129, 1822
パスワード〔PDUSER〕　86
パスワードに設定できる制限　1527
パスワードの変更〔監査人〕　1304
パスワードの文字列制限　1527
パスワードの文字列制限を解除する　1546
パスワードの文字列制限を新規設定する　1534
パスワードの文字列制限を変更する　1537
パスワードの有効期間の設定　1533
パスワード無効アカウントロック状態　1528
パスワード無効アカウントロック状態になるユーザを事前調査する　1542
パスワード無効アカウントロック状態のユーザに対する権限削除　1545
パスワード無効アカウントロック状態のユーザに対する権限付与　1545
パスワード無効アカウントロック状態を解除する 1540
パスワード有効期間の設定方法　1561
パスワードを変更する　1535, 1540
バックアップ取得時間の短縮〔ユーザ LOB 用 RD エリア〕　279
バックアップ取得時点に回復する方法　1155
バックアップ取得モード　234, 1147
バックアップ〔データベースの更新ログを取得しない運用〕　292
バックアップに関する質問　1931
バックアップの取得〔24 時間連続稼働〕　1945
バックアップの取得〔HiRDB ファイルシステム領域〕　441
バックアップの取得〔pdcopy コマンド以外〕　272
バックアップの取得〔更新凍結コマンド〕　279
バックアップの取得時期　230
バックアップの取得単位　230
バックアップの取得方法　229
バックアップの取得例　244
バックアップファイル　231
バックアップファイルのサイズ　231
バックアップファイルの内容確認　232
バックアップ閉塞　232
バックアップ閉塞時のデッドロック　275
バックアップ閉塞中の排他待ちタイムアウト　275
バックアップ閉塞〔バックアップの取得〕　272
バックエンドサーバ接続保持機能　1951
バックエンドサーバ接続保持時間　1953
バックエンドサーバの移動〔負荷均等化〕　498
ハッシュ関数を変更する　751
ハッシュグループ　634
ハッシュ分割表のリバランス機能　633
バッファ使用状況　1931
バッファ長のチューニング　1241
バッファ排他待ち回数　1214, 1220
バッファ不足発生回数　1221
バッファ不足発生回数（INSB）　1215
パブリッククラウド環境に関する準備　1722
範囲指定の回復　297, 1147

ひ

必須文字種　1528
非同期 READ プロセス数のチューニング　1251
非同期出力用バッファのすべての面がフラッシュ待ちになった場合　1372

非同期出力用バッファフラッシュ契機　1319
ビュー解析情報取得要求回数　1242
ビュー解析情報用バッファ長のチューニング　1241
ビュー解析情報用バッファヒット回数　1242
ビュー表の削除　614
表，及びインデクスに関する質問　1925
表移行用アンロードファイル　518
表単位の再編成　573
表単位の再編成〔例題：HiRDB/シングルサーバ〕　580
表単位の再編成〔例題：HiRDB/パラレルサーバ〕　582
表単位の状態解析　565
表定義情報取得要求回数　1241
表定義情報と表データを同時に移行する方法　512
表定義情報と表データを別々に移行する方法　512
表定義情報用バッファ長のチューニング　1241
表定義情報用バッファヒット回数　1241
表にインデクスを定義する　812
表の移行〔スキーマ単位〕　535
表の移行〔別スキーマに移行〕　541
表の運用　564
表の格納効率　565
表の再編成　572, 805
表の再編成時期を予測する方法　602
表の再編成〔大量データ〕　578
表の再編成〔ディクショナリ表〕　593
表の再編成の実行単位　572
表の再編成〔例題〕　580
表の再編成〔例題：LOB 列が定義されている表〕　590
表の再編成〔例題：抽象データ型を定義した表〕　598
表の再編成〔例題：ログレスモード〕　595
表の定義系 SQL　788
表の定義を変更　626
表の分割格納条件の変更　642
表の分割格納条件を変更するときの運用　718
表の分割定義を変更する　754
表の横分割数を増やす　628
表の横分割数を増やす方法〔ハッシュ分割表のリバランス機能〕　633
表のリバランス　634
表名を変更する　627
表を再編成する前に　577
表を削除する　784

ふ

ファイルアクセスエラー検知時ユニットダウン機能 445
ファイル障害が発生したとき〔システムファイル以外のファイル〕　1051
ファイルの状態を変更する　143, 210
フェールオーバークラスターマネージャー〔MSFC〕　1688
フェールオーバークラスタ管理〔MSFC〕　1676
負荷均等化を目的としたバックエンドサーバの移動 498
符号付きパック形式　800
符号部の変換　800
物理エラー検知時ユニットダウン機能　972
プラグインインデクスの遅延一括作成　830
プラグインルーチン定義キャッシュヒット回数　1247
プラグインルーチン定義取得要求回数　1247
プリフェッチ入力ページ数　1215, 1222
プリフェッチバッファ不足発生回数　1216, 1222
プリフェッチヒット回数（PRHIT）　1222
プリフェッチヒット率（HIT）　1222
プリフェッチヒット率（PRHIT）　1215
フルバックアップ　262
フレキシブルハッシュ分割〔ハッシュ関数の変更〕 751
プログラムメンテナンス機能　1950
プロセス ID〔接続ユーザ情報ファイルの出力情報〕　1065, 1066
プロセス ID〔タイムアウト情報の出力内容〕 327, 328
プロセス ID〔デッドロック情報の出力内容〕 322, 324
プロセス数のチューニング　1249
プロセス数の変更　397
プロセスの異常終了回数監視機能　371
プロセスの異常終了回数監視機能〔系切り替え機能〕　1819
フロントエンドサーバの SUSPEND 状態　56
分割格納条件の変更の目的　642
分割機能　646
分割定義を変更する　754

へ

平均値　1226
閉塞〔監査証跡ファイル〕　1320
閉塞〔ステータスファイル〕　218
並列 WRITE 時間　1232
ページコンパクション　965
ページスプリットの発生回数　818
ページの状態　963
ページ破壊〔対処方法〕　1127
変更前 RD エリア情報リスト　677

ほ

ポート数　401
ポート番号を範囲指定する　407
ホスト BES　1578
ホスト BES の開始　1748
ホスト名の運用方法　1615
ホスト名を変更する方法　409

ま

マージ解析〔再編成時期予測機能〕　609
マスタディレクトリ用 RD エリアだけを回復する場合　1187
マスタディレクトリ用 RD エリアに障害が発生したとき　1030
マスタディレクトリ用 RD エリアに障害が発生したときの開始方法　54
待ち資源種別　323, 324, 328
待ち資源情報　323, 324, 328
マルチスタンバイ構成の例〔HiRDB/シングルサーバ〕　1595
マルチスタンバイ構成の例〔HiRDB/パラレルサーバ〕　1600
マルチフロントエンドサーバ　396
満杯セグメント　963

み

未完状態のインデクス　812
未決着状態のトランザクションがあるときの対処方法　1067
未使用セグメント　963
未使用セグメント数　840
未使用ページ　963
ミラー正副不一致状態が発生したときの対処方法 1138
ミラーリング機能　1138

む

無応答障害時の影響を抑える方法　366

め

メッセージキュー監視機能　368
メッセージキュー監視時間　368
メッセージキュー滞留要因と対策　369
メッセージログの出力先　299
メッセージログの出力分散化　301
メッセージログの出力方式　300
メッセージログファイル　1015
メッセージログファイルに障害が発生したとき　1051
メッセージログファイルの切り替え　299
メッセージログファイルの容量を変更する　300
メッセージログを参照する　299

も

基表　789
モニタモード　1566
モニタモードの場合のリソースの登録〔MSFC〕　1678, 1690

や

役割の作成〔MSFC〕　1688

ゆ

有効保証世代数　202, 1922
有効保証世代数を増やす　212
ユーザ ID　1065
ユーザが行うバックエンドサーバの負荷均等化　508
ユーザ権限　76
ユーザ権限情報取得要求回数　1242
ユーザ権限情報用バッファ長のチューニング　1242
ユーザ権限情報用バッファヒット回数　1242

ユーザ権限を設定する手順　82
ユーザ権限を取り消す　87
ユーザサーバホットスタンバイ　1805
ユーザ識別子　325
ユーザ識別情報　1934
ユーザ数を増やすときにすること　400
ユーザ定義型情報用バッファ長のチューニング　1245
ユティリティ操作イベント　1281
ユニット単位の開始方法　57
ユニット単位の終了方法　57
ユニットの移動　466
ユニットの削除　462
ユニットの追加　457

よ

横分割数を増やす　628
予備系　1568
予備〔ステータスファイル〕　218
予約〔システムログファイル〕　111
予約〔シンクポイントダンプファイル〕　201
予約〔ステータスファイル〕　218

り

リスト〔STANDBY 指定のときの運用方法〕　791
リストア〔HiRDB ファイルシステム領域〕　442
リストの管理　789
リストの情報を調べるコマンド　790
リストの初期化時期の変更　790
リスト用 RD エリアの容量不足　841
リスト用 RD エリア〔バックアップの取得〕　231
リソースの使用率　405
リソースの使用率の監視　367
リソースの登録〔MSFC〕　1678, 1690
リバランス機能　633
リバランス処理に時間が掛かる場合　639
リバランス表の定義　635
リバランス表の横分割数を増やす場合　636
リバランスユティリティ　634
リバランスユティリティ〔実行モード〕　635
リバランスユティリティの使い方　639
履歴情報ファイル〔差分バックアップ機能〕　269
リロード　572

る

累積差分バックアップの運用例　269
累積差分バックアップの作成　268
ルーチン定義キャッシュ総使用領域長　1247
ルーチン定義キャッシュヒット回数　1246
ルーチン定義情報取得要求回数　1247
ルーチン定義情報用バッファ長のチューニング　1246

れ

レコード使用率〔ステータスファイル〕　221
レコード長〔システムログファイル〕　150
レジストリ情報用バッファ長のチューニング　1248
列構成情報ファイルの指定例　782
列の削除　623
列の追加　615
列の追加〔FIX 表〕　618, 620
列の追加〔LOB 列〕　617
列の追加〔抽象データ型〕　617
列の追加〔非 FIX 表〕　616
列名を変更する　627
連続認証失敗アカウントロック状態　1529
連続認証失敗アカウントロック状態のユーザを確認する　1551
連続認証失敗アカウントロック状態を解除する　1553
連続認証失敗回数の制限　1529
連続認証失敗回数の制限を解除する　1549
連続認証失敗回数の制限を新規設定する　1548
連続認証失敗回数の制限を変更する　1549
連続認証失敗許容回数　1529
連続認証失敗許容回数を確認する　1549
連動系切り替え　1798

ろ

ローカルバッファ使用時に障害発生したときの対処　1136
ロール　101

ロールバック　49, 290, 1145
ロールバック処理が進行中であることを確認する方法　374
ロールバック用ログ入力バッファ面数のチューニング　1271
ロールフォワード　290
ロール利用権限　78
ロールを使用した権限の設定　101
ログ取得モード　235, 290, 1148
ログポイント　127
ログポイント情報ファイル　127
ログレス閉塞　292
ログレスモード　235, 290
ログレスモードの運用　295
論理的解析情報　1257