

# Db2 バックアップ・リカバリ

発行日：2025/11

日本アイ・ビー・エム システムズ・エンジニアリング株式会社



# 目次

---

## Db2バックアップ・リカバリーの基本

1. データベースのバックアップ・リカバリーの概要
2. バックアップ
3. リカバリー

## Db2バックアップ・リカバリーの応用

1. 増分バックアップとリカバリー
2. 表スペースバックアップとリカバリー
3. 圧縮バックアップからのリカバリー
4. バックアップ・イメージの暗号化
5. リモート・ストレージへのバックアップ
6. バックアップ・イメージの削除運用
7. RECOVERコマンドによるリカバリー
8. ストレージによる高速コピー機能を使用したDb2バックアップ

### 9. その他

- LOAD後の回復とデータベース / 表スペースの状態
- DROPされた表の回復
- システム時刻変更に伴う注意点
- 論理バックアップ
- リダイレクト・リストア
- REBUILD

# 基本

## Db2 バックアップ ・リカバリー

1. データベースのバックアップ・リカバリーの概要
2. バックアップ
3. リカバリー

# データベースのバックアップ・リカバリーの概要

## 01

バックアップ・リカバリーとは

- バックアップとは
- リカバリーとは
- バックアップ・リカバリーの主な要件

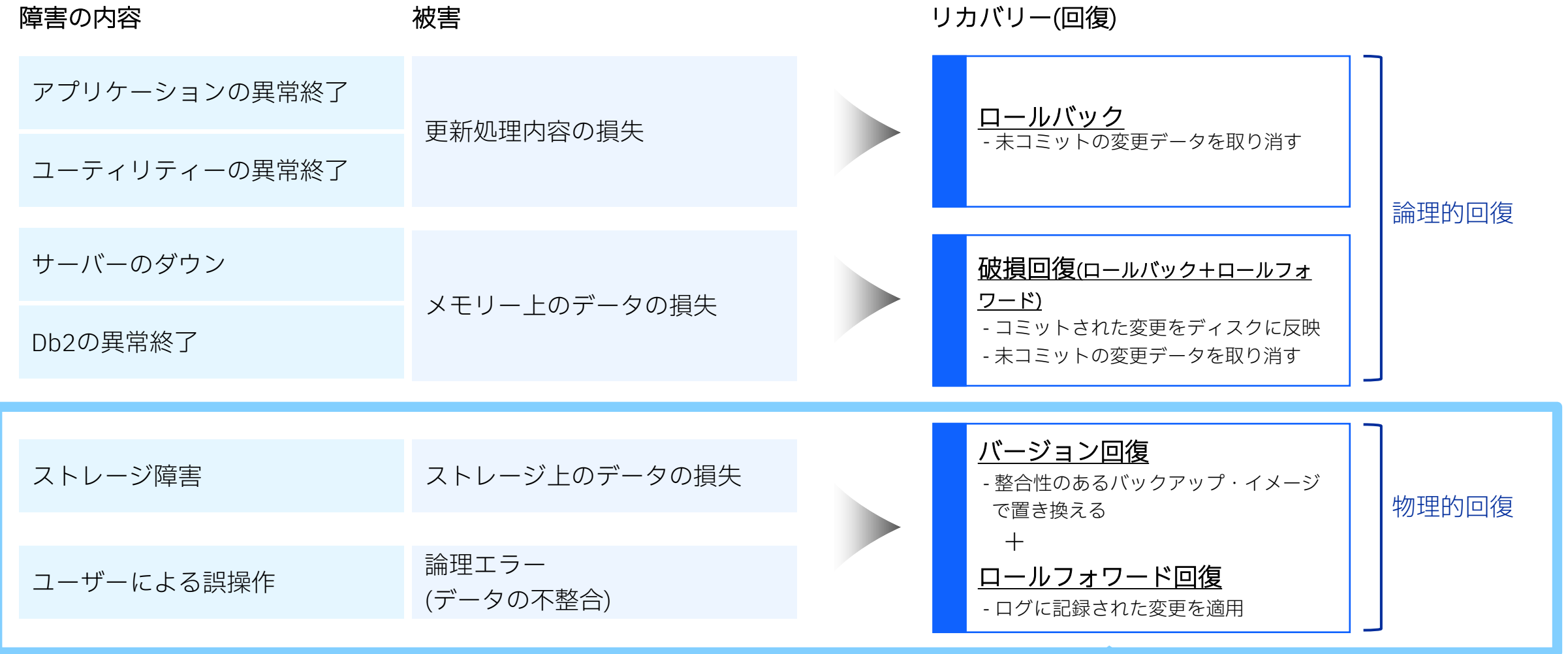
ログとは

バックアップ・イメージに含まれるもの  
/ 含まれないもの

# データベースのバックアップ・リカバリーとは

## 障害が起きたデータベースを、正しい状態に回復する仕組み

- ストレージ上のデータが破損した状態からデータベースを回復するためには、バックアップ・リカバリーが必要 (枠線内)



当ガイドでカバーする内容はこちら

# バックアップとは

ストレージ障害やユーザーによる誤操作に備え、ユーザーデータを回復できるよう、データベースをバックアップする

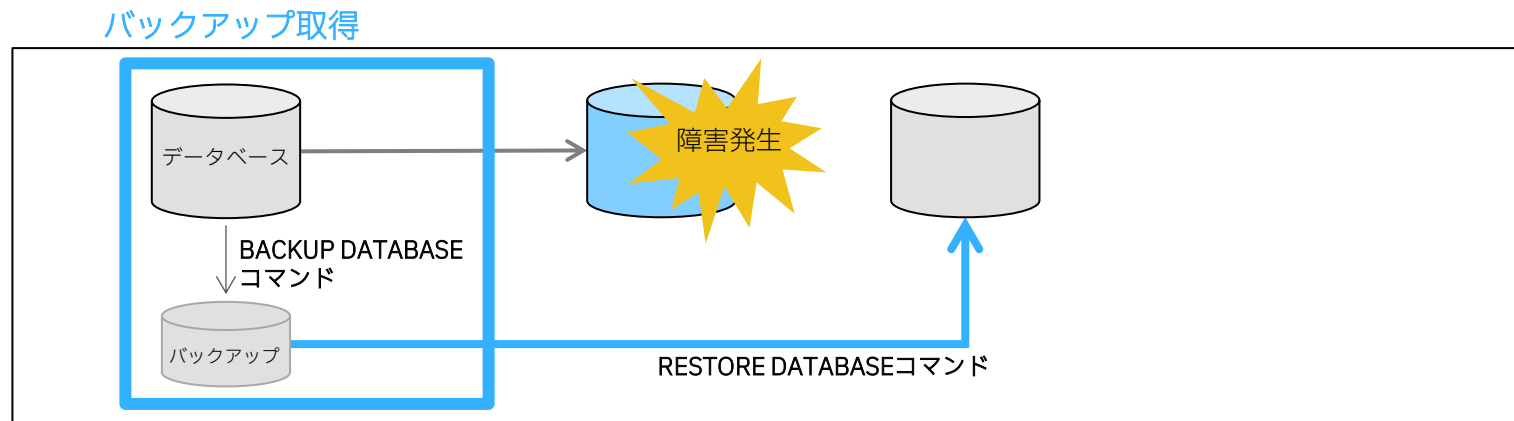
**バックアップとは、ある時点でのデータベース/表スペースの複製ファイルを作成すること**

– 運用作業としては、バックアップを定期的を取得する

バックアップ取得は、シェルスクリプトなどを作成し、ジョブとして実行することで自動化している環境が多い

– リカバリーは、障害発生時の運用作業となるため、自動化を通常行わない

その代わりに、障害発生時のリカバリー手順として、コマンドレベルの詳細な手順書を作成し、有事の際に備える



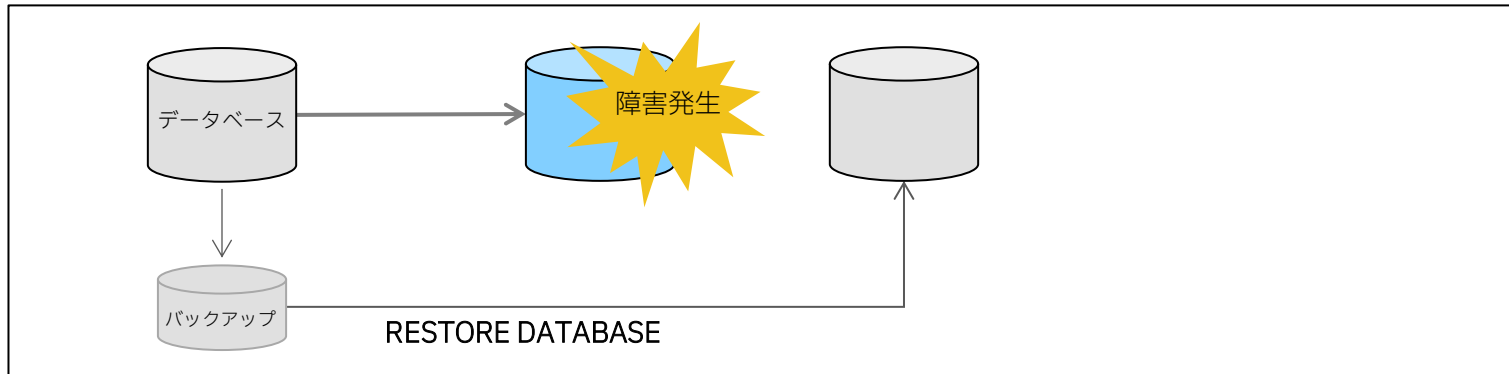
有事の際にはバックアップを利用して、バックアップ取得時点のデータベースを復元できる

# リカバリーとは

データベースを回復させるためにバックアップからのリストアを行い、ログが利用可能であればログを適用し障害発生時点まで戻す

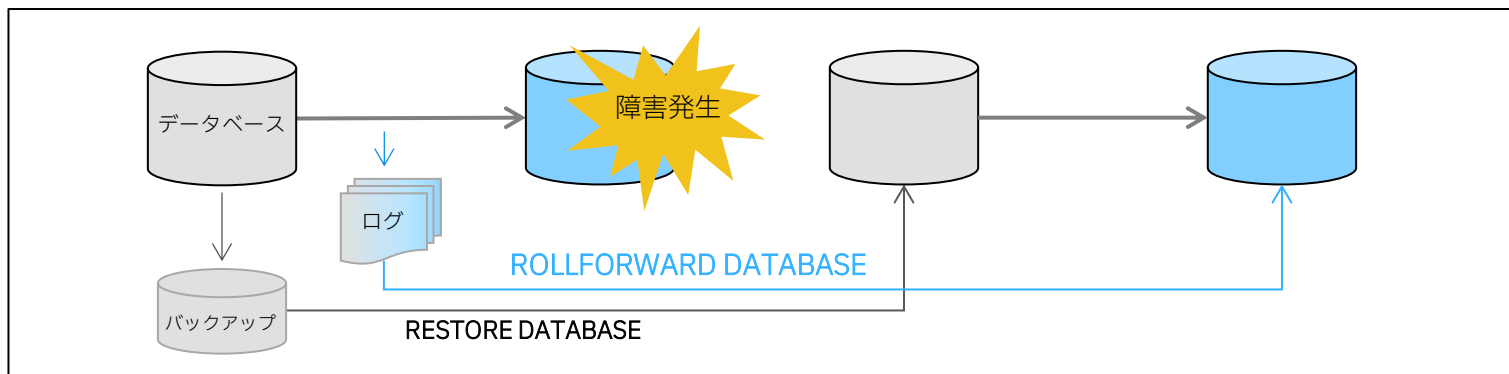
## RESTORE DATABASEによる回復（バージョン回復）

- 事前に取得しておいたバックアップ・ファイルを使用してデータベースを回復する



## ROLLFORWARD DATABASEによる回復（ロールフォワード回復）

- ログを適用し、データベースをある時点までの状態に戻す



# バックアップ・リカバリーの主な要件

さまざまな障害からデータをリカバリーするために、データベースやトランザクション・ログの定期的なバックアップが必要

## バックアップ・リカバリーの主な要件の例

### –障害が起こった時、どの時点までリカバリーが必要か？

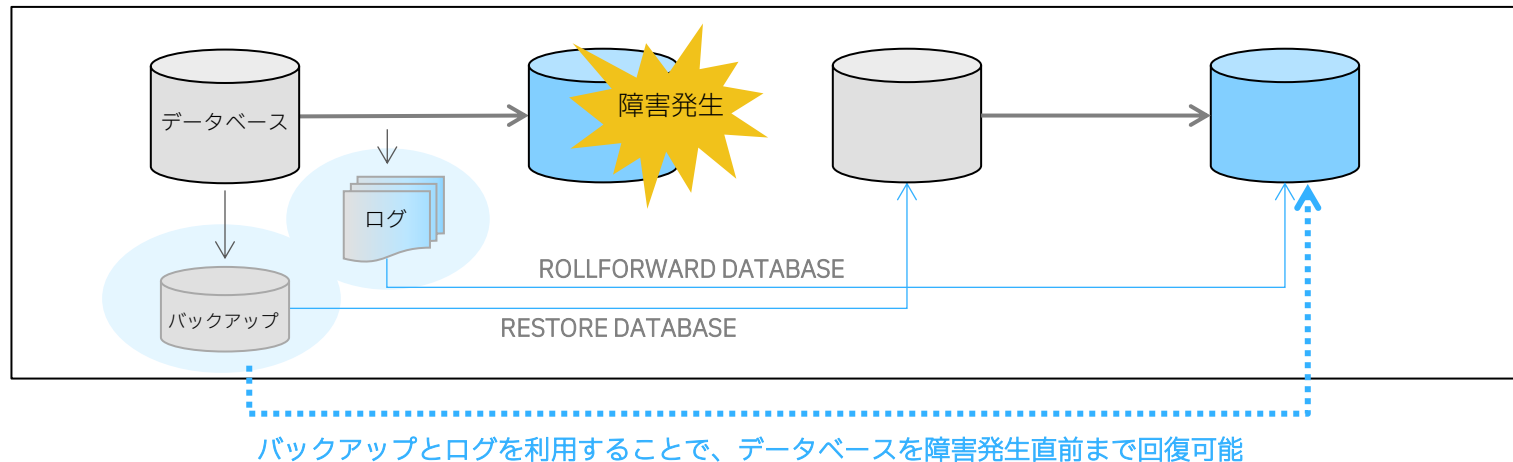
- 障害発生直前や、ユーザーが指定する時刻(Point-in-time)への状態のデータベースを復元するには、データベースのバックアップに加え、トランザクション・ログ(以下、ログ)も保存しておく必要がある

### –バックアップを実行する間隔はどれくらいか？

- バックアップの取得間隔が長いと、障害時に失われるデータが増える。逆に短すぎると運用負荷が高くなる
- データの重要性・変更頻度・ビジネス要件・コンプライアンス規則などを考慮し、「許容できるデータ損失量 (RPO: Recovery Point Objective : 目標回復時点)」に応じた間隔で取得する

### –障害時にどのくらいの時間でリカバリーする必要があるか？

- 「どれくらいの時間で復旧しなければならないか (RTO : Recovery Time Objective : 目標復旧時間)」
- 業務の許容停止時間に応じて決める
- 一般的にログの適用には時間がかかるため、バックアップを適度な頻度で取得する



# ログとは

クラッシュ・リカバリーやロールフォワード・リカバリーに必要となるファイル

- Db2では、アプリケーションが行なうデータの挿入、更新、削除など、データベースへの変更内容をログと呼ばれるファイルに記録する
- ログは、変更されるデータよりも、必ず先にディスクに書き出される（"WRITE LOG AHEAD"）
- これにより、データベースがダウンした場合においても、データベースを一貫性のある状態に回復することができる

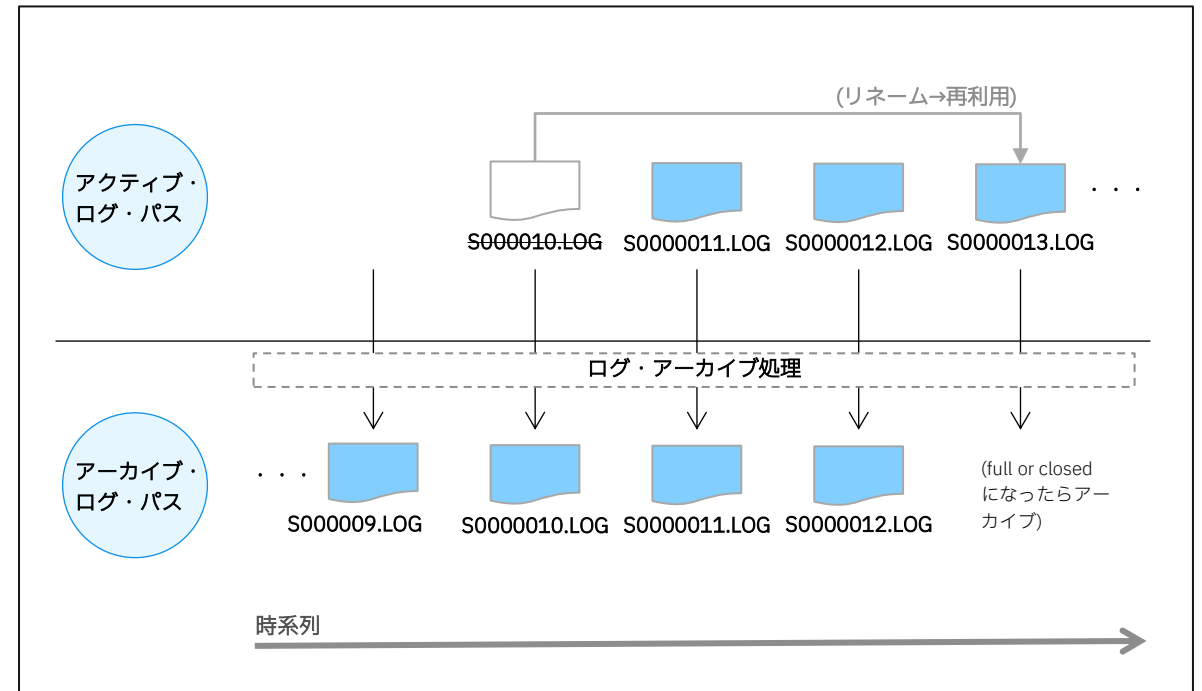
## Db2のログは大きく分けて、2つの種類がある

### – アクティブ・ログ

- クラッシュ・リカバリーならびに、ロール・フォワードに必要となるログファイル
- アクティブ・ログ・パスに存在するログファイル
- get db cfg コマンドで取得可能な“最初のアクティブ・ログ・ファイル”より後ろのファイル全て
- 削除してはいけないファイル

### – アーカイブ・ログ

- ロールフォワード・リカバリーに必要となるログ・ファイル
- アーカイブ・ログ・パスにコピーされたログファイル
- アクティブ・ログ・パス上にある同名ファイルは、新たなアクティブ・ログとして再利用されるまでアクティブ・ログ・パスにも残る

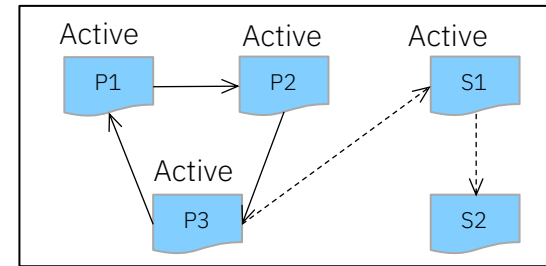


# ロギングの方式

- Db2では、循環ロギングとアーカイブ・ロギングの2つのロギング方式がサポートされる
- ロギングの方式は、DB構成パラメータ LOGARCHMETH1, LOGARCHMETH2 で設定

## 循環ロギング

- LOGARCHMETH1、LOGARCHMETH2=OFF
- ログ・ファイルを循環して使用するロギング方式  
ログ・ファイルは、そのファイルに記録されているトランザクションが終了し、かつ、ディスクに反映された場合に、再利用可能となる
- オンライン・バックアップは不可

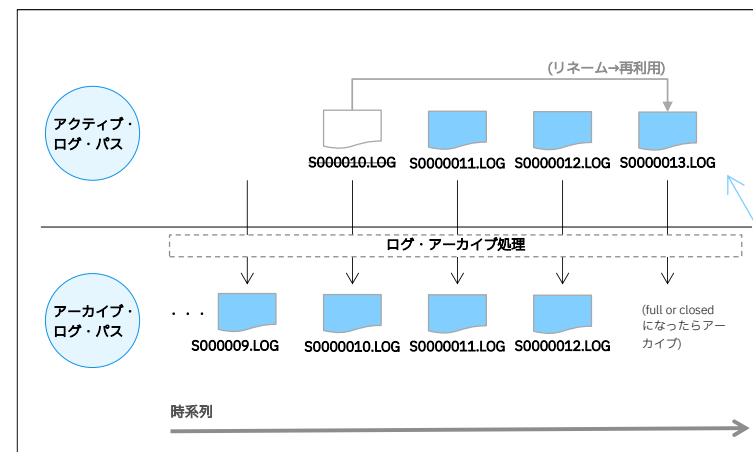


DB構成パラメータ

LOGPRIMARY (1次ログ数)	3
LOGSECOND (2次ログ数)	2
LOGARCHMETH1	OFF
LOGARCHMETH2	OFF

## アーカイブ・ロギング

- LOGARCHMETH1、LOGARCHMETH2=OFF以外
- ログ・ファイルをすべて保存するロギング方式  
アーカイブ・ロギングでは、データベースへ行なわれた更新がすべて保存されるため、データベースや表スペースのリストア後、ログ・ファイルを適用してデータベースをある時点またはログ・パス上にある最後のログ・ファイルまでリカバリー可能



DB構成パラメータ

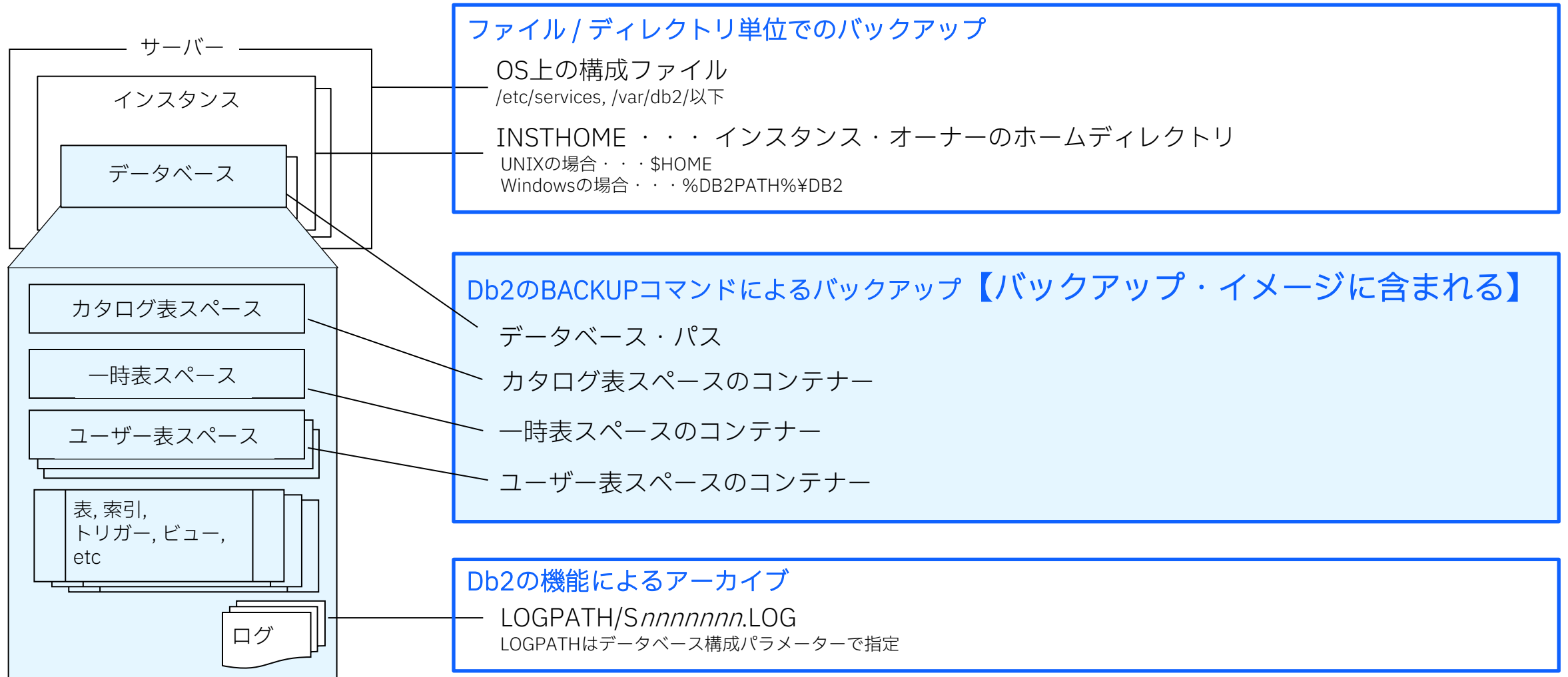
LOGPRIMARY (1次ログ数)	3
LOGSECOND (2次ログ数)	2
LOGARCHMETH1	DISK:/arch_path1
LOGARCHMETH2	DISK:/arch_path2

[図：補足]  
アクティブ・ログ・パス上でLOGPRIMARY値以上のログが必要になったら、LOGSECOND値まで追加作成される

# バックアップ・イメージに含まれるもの / 含まれないもの

## Db2を構成するファイル群

– BACKUP DATABASE コマンドで取得されないオブジェクトについては別途、管理者がバックアップする



# 参考：バックアップ・イメージに含まれるもの

---

## バックアップ・イメージには、以下のものが含まれる

- データベースのメタ・データ
  - データベース・パス情報
  - 表スペース定義
  - データベース構成ファイル
  - 回復履歴ファイル
  - ログ制御ファイル
- バックアップ時に指定された表スペースに含まれるデータ（一時表スペースを除く）
- ログ・ファイル

## データベース・バックアップの場合

- データベースのメタ・データ
- データベースに定義されたすべての表スペースに含まれるデータ

## 表スペース・バックアップの場合

- データベースのメタ・データ
- その表スペースに含まれるデータ

# 参考：バックアップ・ファイルに含まれないデータベース関連情報

データベース関連情報	バックアップ対象と方法	備考
ポート番号、サービス名	/etc/services	
環境変数とインスタンス名ファイル	/var/db2/global.reg	
DBM構成パラメータ	~/sqllib/db2system	DB構成パラメータはDBのバックアップに含まれるため、RESTOREにて回復可能
ストアードプロシージャ (外部ルーチン)	プログラム・ファイル	内部ルーチンの場合は、バックアップに含まれる
ユーザ定義関数 (外部ルーチン)	プログラム・ファイル	内部ルーチンの場合は、バックアップに含まれる
レジストリ変数	~/sqllib/profile.env db2setの出力結果	
プロファイルとノード情報	db2nodes.cfg db2profile	
カタログ情報	LISTコマンドの出力結果 db2cfexpによるエクスポート	<ul style="list-style-type: none"> <li>• <a href="#">db2cfexp</a>を使用してエクスポートされたデータベースの接続構成は、<a href="#">db2cfimp</a>によりインポート可能</li> <li>• db2cfexpには以下の情報が含まれる             <ul style="list-style-type: none"> <li>– データベース情報 (DCS および ODBC 情報を含む), ノード情報, プロトコル情報, データベース・マネージャ構成設定, レジストリー設定, 共通 ODBC/CLI 設定</li> </ul> </li> </ul>

# 参考：db2lookコマンドによるDDLの抽出

---

db2lookを使用すると以下の内容を保存することが可能です

- データベース作成時のCREATE DATABASE文およびオプション指定内容
- データベースに存在するオブジェクトのDDL
- 表および索引の関する統計情報を複製するためのUPDATEステートメント
- GRANT権限ステートメント
- データベース、およびデータベース・マネージャ構成パラメータ用のUPDATEステートメント
- 登録変数用のdb2setステートメント
- ユーザー定義のデータベース・パーティション・グループ、ストレージ・グループ、表スペース、バッファープールのDDL
- [db2look - Db2 統計および DDL 抽出ツール・コマンド - IBM Documentation](#)

# バックアップ

## 02

バックアップの概要

バックアップのモニター

バックアップのパフォーマンスの最適化

オフライン・バックアップ

オンライン・バックアップ

- バックアップ・イメージに含まれるログ・ファイル
- オンライン・バックアップの考慮点
- オンライン・バックアップとユーティリティの互換性

db2ckbkp

# バックアップ

---

ある時点でのデータベース/表スペースの複製ファイルを作成すること

## 種類

- データベースのバックアップ
  - オフライン、オンライン
- 増分バックアップ
  - デルタ、インクリメンタル
- 表スペースのバックアップ
  - オフライン、オンライン
- 圧縮バックアップ

# バックアップ・ファイル名と意味

---

ファイル名：DB別名.タイプ.インスタンス名.DBPARTxxx.タイムスタンプ.シーケンス番号（順序番号）

- 例) SAMPLE.0.ki1212.DBPART000.202507311111309.001

– バックアップ・タイプ

- 0: データベース・バックアップ
- 3: 表スペース・バックアップ
- 4: LOADのコピー・ファイル

– ノード番号、カタログ・パーティション番号

- 単一パーティション・データベース環境では、常に DBPART000
- パーティション・データベース環境では、DBPARTxxx
  - xxx は db2nodes.cfg ファイル中でデータベース・パーティションに割り当てられている数

– タイム・スタンプ

- バックアップ操作が実行された日付と時刻
- タイム・スタンプの形式は `yyyymmddhhnnss`
  - yyyy は年 (1995 ~ 9999)
  - mm は月 (01 ~ 12)
  - dd は日 (01 ~ 31)
  - hh は時 (00 ~ 23)
  - nn は分 (00 ~ 59)
  - ss は秒 (00 ~ 59)

– シーケンス番号

- ファイル拡張子として使用する3桁の番号

# バックアップされる対象

---

## バックアップされる領域は、実際に使用されているエクステンツ

- list tablespaces show detail コマンドの”使用したページ”が目安
- 表スペースをCREATEしただけでは、そのスペースはバックアップ対象にはならない
  - 例えば、1GBの表スペースをCREATEしてバックアップを取得すると、1GBの領域もバックアップされるのではなく、実際に使用されたエクステンツ（ヘッダー部分）のみが、バックアップ対象になる
- 表データを格納するために、一旦アロケーションされたエクステンツは、たとえ全データがDELETEされていてもバックアップ対象となる
  - DELETEだけでは、エクステンツはまだ開放されていないためである
- 表をDROPすることによりエクステンツが解放されれば、バックアップ対象とはならない

# バックアップのモニター

---

LIST UTILITIES SHOW DETAILやdb2pdコマンドなどを使用して、バックアップの処理状況を確認可能

## LIST UTILITIES SHOW DETAIL コマンド

- インスタンス上のアクティブなユーティリティーのリストを標準出力に表示
- 実行中の処理内容や開始時刻の確認が可能
  - 処理されるバイト数などが表示されるが、実際のバックアップ・ファイルのサイズとは異なる
  - 特に、増分バックアップや圧縮バックアップの場合

## db2pd -d <データベース名> -utilities コマンド

- Db2 データベース・システム・メモリー・セット内からDb2ユーティリティー情報を取得
- 実行中の処理内容や開始時刻の確認が可能

## db2pd -d <データベース名> -barstats <エージェント EDU IDの値> コマンド

- Db2 データベース・システム・メモリー・セット内からバックアップおよびリストア操作のパフォーマンスに関するモニター情報を取得
- 「Progress Monitor」で現在の作業進行情報を確認可能

## db2diag.log

- Db2の診断および管理通知メッセージが記録される

参考) [バックアップ操作のモニター - IBM Documentation](#)

# バックアップのパフォーマンスの最適化

---

## Db2はバックアップ処理時間を短くするために、バッファラーの数、バッファラー・サイズ、および並列処理設定の最適値を自動的に選択する

- 使用可能なユーティリティ・ヒープ・メモリーの大きさ、プロセッサの数、およびデータベース構成に基づく

## その他、バックアップの処理時間を短くするために、以下も有効

- 表スペースバックアップを指定する
  - データベース内の一部の表スペースのみバックアップを取得する（表スペース・バックアップを参照）
- バックアップされる表スペースの数を指定する（PARALLELISM オプション）
  - データの読み取りや、圧縮バックアップ時のデータ圧縮を実行するときに、使用するプロセスの数またはスレッドの数を設定する
  - 各プロセスやまたはスレッドは特定の表スペースに割り当てられるため、バックアップする表スペースの数と同等にする
  - メモリーとCPUの両方のオーバーヘッドが必要になる場合もあるため、注意する
- バックアップ・バッファラー・サイズを大きくする（BUFFER オプション）
  - 推奨は、表スペースのエクステント・サイズの倍数に1ページを加えたもの
  - 複数の表スペースがありそれぞれエクステント・サイズが異なる場合は、エクステント・サイズの公倍数に1ページを加えた値を指定する
- バッファラー数を増やす（WITH BUFFERS オプション）
  - 少なくともバックアップ先（またはセッション）の2倍のバッファラーを使用し、バックアップ先装置がデータを待つ状態にならないようにする
- 表スペース内並列処理（レジストリ変数 DB2\_BACKUP\_ITP = ON）
  - Db2が各表スペースをバックアップする際に複数のスレッドを使用することを可能にする
  - 1つまたは複数のテーブルスペースが他のスペースよりもはるかに大きい場合、バックアップ操作の実行にかかる全体的な経過時間は大幅に短縮される

# バックアップのパフォーマンスの最適化

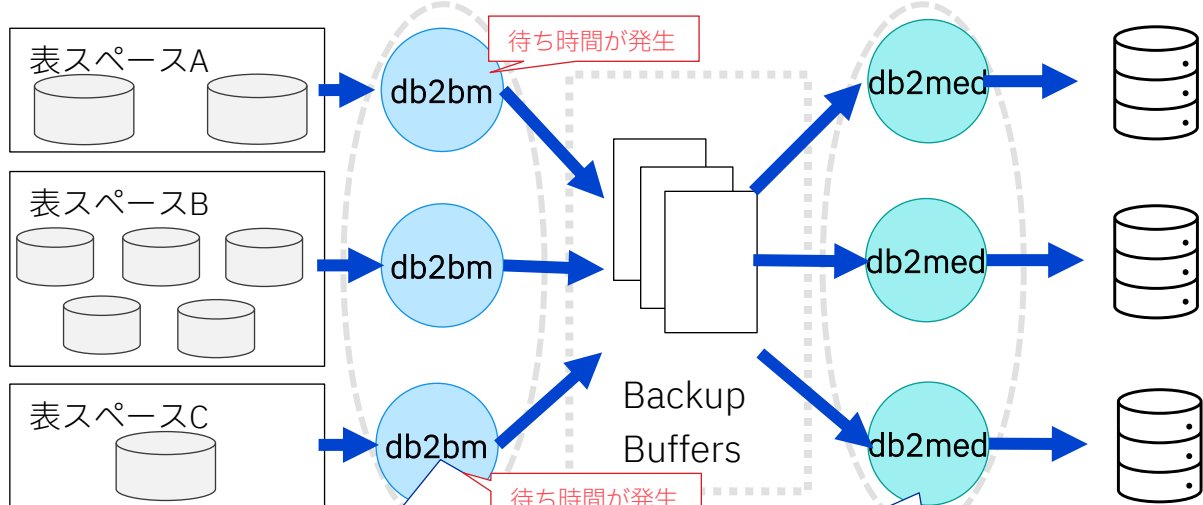
## 表スペース内並列処理 (レジストリ変数 DB2\_BACKUP\_ITP = ON)

- 例: db2 backup db testdb to /data/backup, /data/backup, /data/backupを実施した場合

<従来のバックアップ取得方法>

表スペース単位でバックアップを並列処理

(PARALLELISMを指定しない場合、自動で最適値を選択する)

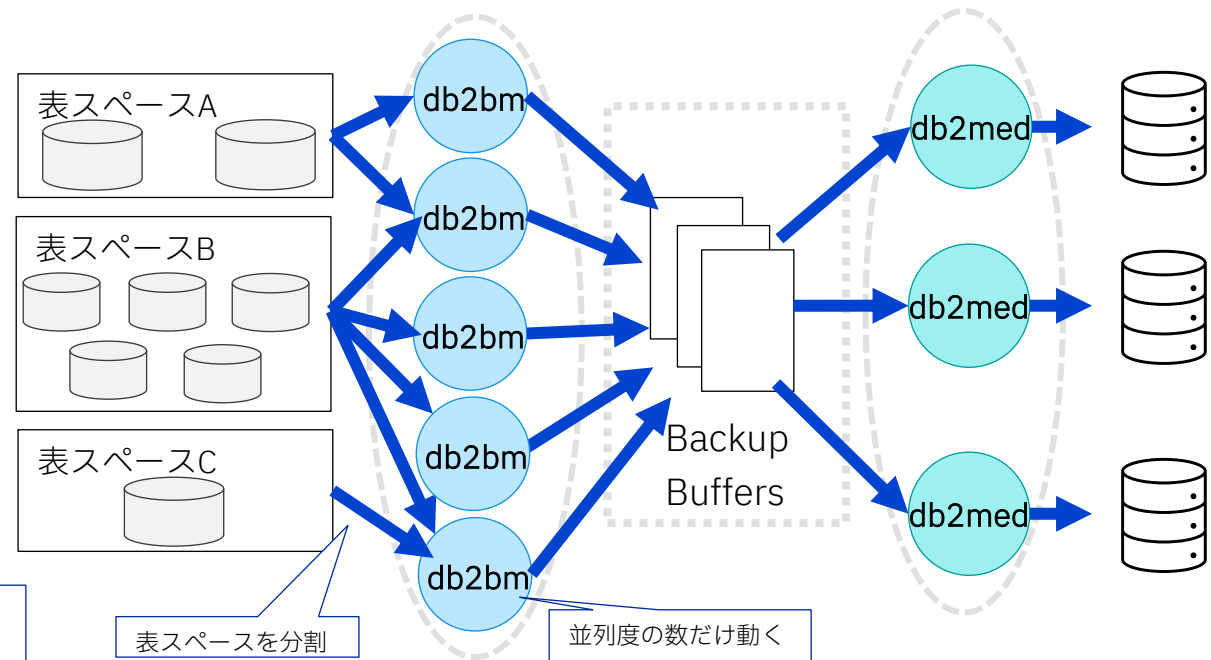


db2bm: バッファ・マネージャ  
表スペース・コンテナ・ファイルからデータを読み取る  
並列度の数だけ動く

db2med: メディア・コントローラ  
バックアップ・イメージ・ファイルにデータを書き込む  
指定した出力先の数だけ動く

<V12.1でDB2\_BACKUP\_ITP=ONの場合のバックアップ取得方法>

表スペースを分割してバックアップを並列処理



表スペースを分割

並列度の数だけ動く

# 並列度の確認方法

## Db2診断ログ db2diag.log で並列度を確認することが可能

```
2025-07-31-11.10.26.435135+540 E507377E614          LEVEL: Info
PID       : 558041          TID : 140364656666176 PROC : db2sysc 0
INSTANCE: ki1212          NODE : 000          DB   : SAMPLE
APPHDL   : 0-25           APPID: *LOCAL.ki1212.250731021025
UOWID    : 1              LAST_ACTID: 0
AUTHID   : KI1212         HOSTNAME: ISEI20241029-1945-fun-1.jp-ise.com
EDUID    : 50             EDUNAME: db2agent (SAMPLE) 0
FUNCTION: DB2 UDB, database utilities, sqluxGetDegreeParallelism, probe:464
DATA #1 : <preformatted>
Autonomic backup/restore - using parallelism = 10.
```

並列度=10でバックアップが実行されている

```
2025-07-31-11.10.28.002871+540 E513093E2454          LEVEL: Info
PID       : 558041          TID : 140364656666176 PROC : db2sysc
~略~
Parallelism           = 10
Number of buffers     = 20
Buffer size           = 790528 (193 4kB pages)
```

実際に10個のBMが並列で動いていたことが確認できる

									Log
BM#	Total	I/O	MsgQ	WaitQ	Throttle	LogPhase	Buffers	kBytes	kBytes
---	-----	-----	-----	-----	-----	-----	-----	-----	-----
000	0:00:01	0:00:00	0:00:00	0:00:01	0:00:00	0:00:00	16	7936	12
001	0:00:01	0:00:01	0:00:00	0:00:00	0:00:00	0:00:00	33	24576	0
002	0:00:01	0:00:01	0:00:00	0:00:00	0:00:00	0:00:00	33	24672	0
003	0:00:01	0:00:00	0:00:00	0:00:01	0:00:00	0:00:00	12	8704	0
004	0:00:01	0:00:00	0:00:00	0:00:01	0:00:00	0:00:00	12	8960	0
005	0:00:01	0:00:00	0:00:00	0:00:01	0:00:00	0:00:00	12	8960	0
006	0:00:01	0:00:01	0:00:00	0:00:00	0:00:00	0:00:00	26	19456	0
007	0:00:01	0:00:01	0:00:00	0:00:00	0:00:00	0:00:00	33	24576	0
008	0:00:01	0:00:01	0:00:00	0:00:00	0:00:00	0:00:00	33	24576	0
009	0:00:01	0:00:01	0:00:00	0:00:00	0:00:00	0:00:00	33	24576	0
---	-----	-----	-----	-----	-----	-----	-----	-----	-----
TOT	-	-	-	-	-	-	243	176992	12
~略~									

# オフライン・バックアップ

---

データベースに他のアプリケーションが接続していない状態でバックアップを取得する方法

## 特徴

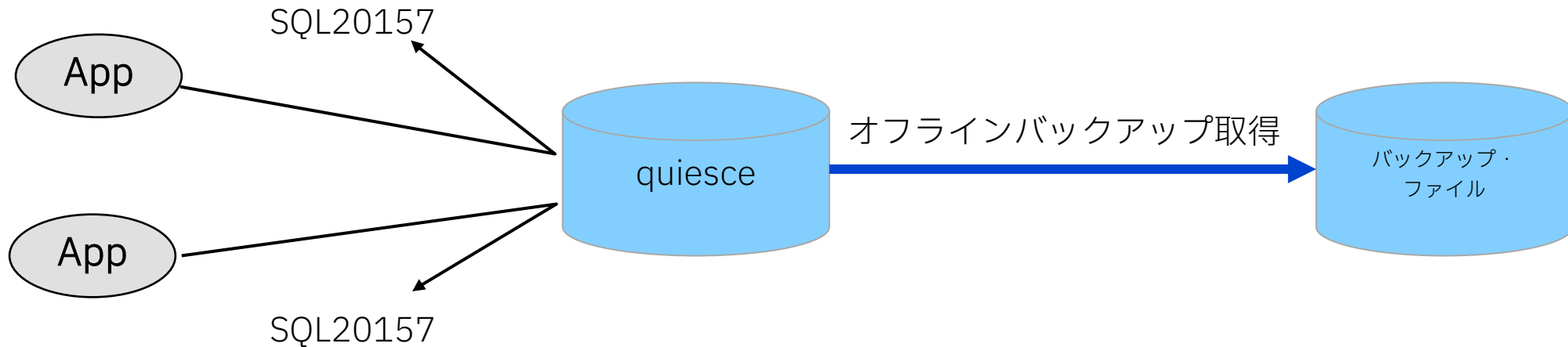
- データベース全体のバージョン回復の場合、ログファイルがなくてもリカバリー可能
  - 表スペース単位の場合、ロールフォワード処理が必要となる
- リカバリー手順が最も容易
- データベースをオフラインにするために、QUIESCE DATABASEコマンドなどを使用するとよい

## 例

```
db2 connect to SAMPLE
db2 quiesce db immediate force connections
db2 connect reset
db2 backup db SAMPLE to /backup
db2 connect to SAMPLE
db2 unquiesce db
db2 connect reset
```

# 参考：オフラインにする方法

- 他のアプリケーションがデータベースに接続中にオフライン・バックアップを取得しようとしたり、オフライン・バックアップ中に他のアプリケーションがデータベースに接続しようとする、SQL1035エラーになる
- QUIESCE DATABASEコマンドを実行すると、すべてのユーザーを強制的に切断し、データベースを静止モードにする
- 静止モードのデータベースに対して接続できるのは以下の権限または特権を持ったユーザー/グループである
  - SYSADM、DBADM、SYSMAINT、SYSCTRL 権限を持つユーザーあるいはQUIESCE\_CONNECT特権を付与されたユーザー/グループ
- 静止モードのデータベースに対して接続権限を持たないユーザー/グループがデータベースに接続を試みると、SQL20157エラーになる



# オンライン・バックアップ

---

データベースに他のアプリケーションが接続している状態でバックアップを取得する方法

## 前提条件

- データベースがアーカイブ・ロギング構成である
  - データベース構成パラメータ LOGARCHMETH1が以下のいずれかであること
    - LOGRETAIN, USEREXIT, DISK, DB2REMOTE, TSM, VENDOR
- BACKUPコマンドで ONLINE オプションを指定

## ログのオプション

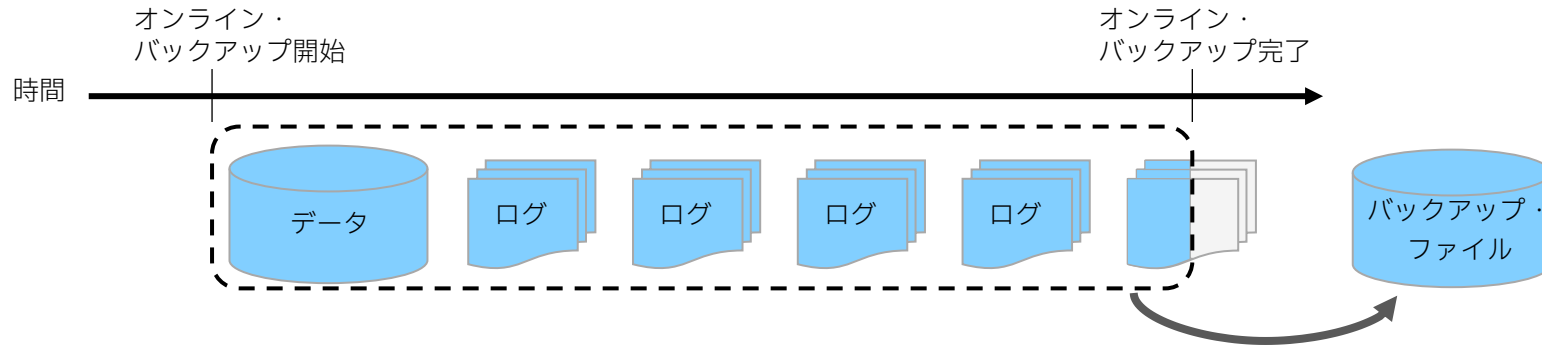
- INCLUDE LOGS (デフォルト)
  - バックアップ・イメージ内に最小限必要なログファイルを含める
  - DPF環境では使用できない
  - COMPRESS指定のバックアップ時には、ログ・ファイルも圧縮される
- EXCLUDE LOGS
  - バックアップ・イメージ内にログファイルを含めない

# バックアップ・イメージに含まれるログ・ファイル

オンライン・バックアップからのリカバリー時には、オンライン・バックアップ取得終了時点までのログ・ファイルのロールフォワードが必須

バックアップ・ファイルに必要最低限のログが含まれるため、ログ・ファイルの管理をユーザーが行う必要はない

- リストア時にはログ・ファイルを指定したディレクトリに書き出し、ロールフォワードを行う際に使用する
- 詳しくは、リストアの章を参照



バックアップ・ファイルと必要なログ・ファイルを一緒に保管する運用にする

- INCLUDE LOGSオプションの使用が推奨される

# バックアップ・イメージに含まれるログ・ファイルの確認方法

## 回復履歴ファイルから必要なログファイル情報を抽出する

### – LIST HISTORYコマンドによる確認例

```
$ db2 list history backup all for SAMPLE
SAMPLE の履歴ファイルのリスト
突き合わせファイル項目数 = 5
```

... (中略) ...

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
B D 20250731111309001 E D S0000007.LOG S0000007.LOG
```

5 個の表スペースを含みます:

```
00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
00004 IBMDB2SAMPLEXML
00005 IBMDB2SAMPLEVECTOR
```

Comment: DB2 BACKUP SAMPLE ONLINE

開始時刻: 20250731111309

終了時刻: 20250731111308

状況: A

合計サイズ: 4775936 (バイト)

シーケンス・サイズ: 4775936 (バイト)

Compr ライブラリー:

暗号化: いいえ

ログの追加: はい

EID: 14 ロケーション: /home/ki1212/backup

最低限必要となるログファイル

回復履歴ファイルに関しては以下のSILを参考にしてください。

[The Technology - DB2 V9 運用管理ガイド：回復履歴ファイルの操作と運用 | ISE FSO Service](#)

# オンライン・バックアップの考慮点

---

- バックアップ取得時に、バッファープールのダーティページがディスクにフラッシュされる
  - 大きなサイズのバッファープールを使用して、大量のデータを更新中にオンライン・バックアップを取得すると、大きなオーバーヘッドを生じる可能性がある
- オフライン・バックアップよりバックアップ時間がかかる
- オンライン・バックアップ終了時点までのログ・ファイルが必要
- リストア、ロールフォワードはオフラインで行う必要がある
  - ただし、表スペース単位はオンラインでも可能

# オンライン・バックアップとユーティリティの互換性

---

## オンライン・バックアップと互換性のあるユーティリティ

- EXPORT
- INSPECT

## オンライン・バックアップと互換性のないユーティリティ

- REORG TABLE
- RESTORE DATABASE
- ROLLFORWARD DATABASE
- ALLOW NO ACCESS オプションを指定した LOAD
- SET WRITE
- ONLINE オプションを指定した BACKUP DATABASE
- ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
- ALLOW NO ACCESS オプションを指定した REORG INDEX

## 特定の環境でのみオンライン・バックアップと互換性のあるユーティリティ

- 詳細は次ページ

## マニュアルリンク

- [オンライン・バックアップと他のユーティリティの互換性 - IBM Documentation](#)

# 特定の環境でのみオンライン・バックアップと互換性のあるユーティリティ

---

以下の SQL ステートメントおよびユーティリティは、特定の環境においてのみオンライン・バックアップと互換性がある

- CREATE INDEX
- REORG INDEX (ONLINE オプションを指定)
- IMPORT
- TRUNCATE TABLE
- LOAD
- REORG TABLE (ONLINE オプションを指定)
- Z ロックを必要とする DDL (ALTER TABLE、DROP TABLE、および DROP INDEX)
- ストレージ・グループ DDL
  - CREATE STOGROUP
  - ALTER STOGROUP
  - DROP STOGROUP
  - RENAME STOGROUP
  - ALTER DATABASE
- RUNSTATS (ALLOW WRITE または ALLOW READ オプションを指定)
- ALTER TABLESPACE
- REBALANCE オプションを指定した ALTER TABLESPACE

## マニュアルリンク

- [オンライン・バックアップと他のユーティリティの互換性 - IBM Documentation](#)

# db2ckbkp

バックアップ・ファイルを検証するコマンド

## 特徴

- バックアップ・ファイルがリストア可能かどうかを確認
- バックアップ・ファイルの情報を表示する
- バックアップ・ファイルの中にログが含まれているか確認

## 使用例

```
$ db2ckbkp SAMPLE.0.ki1212.DBPART000.20250731111027.001
[1] Buffers processed:
#####
#####
#####
Image Verification Complete - successful.
```

## 考慮点

- 複数の出力先を指定してバックアップを取得した場合は、それらの全バックアップ・ファイルを引数に指定する
- テープ装置上にバックアップ・ファイルが存在する場合は、バックアップ・ファイルとしてテープ装置パスを指定する
- TSMにある場合、db2adutlが使用可能

# db2ckbkp

## バックアップ・ファイルのヘッダー情報を表示する例

```
$ db2ckbkp -h SAMPLE.0.ki1212.DBPART000.20250731111027.001
=====
MEDIA HEADER REACHED:
=====
Server Database Name -- SAMPLE
Server Database Alias -- SAMPLE
Client Database Alias -- SAMPLE
Timestamp -- 20250731111027
Database Partition Number -- 0
Instance -- ki1212
Database Configuration Type -- 0 (Non-shared data)
Sequence Number -- 1
Database Member ID -- 0
Image Version -- 2
Minimum Compat. Image Version -- 2
Release ID -- 0x1600 (DB2 v12.1.2.0)
Product Version -- V:12 R:1 M:2 F:0 I:0 SB:0
Database Seed -- 0xFD945DD6
DB Comment's Codepage (Volume) -- 0
DB Comment (Volume) --
DB Comment's Codepage (System) -- 0
DB Comment (System) --
Authentication Value -- 255 (Not specified)
Backup Mode -- 1 (Online)
Includes Logs -- 1 (Yes)
Compression -- 0 (No Library Applied)
Backup Type -- 0 (Database-level)

... (略) ...
```

オンライン・バックアップでINCLUDE LOGSオプションをつけてたバックアップの場合  
0 : ログ・ファイル含まれてない  
1 : ログ・ファイル含まれている

バックアップが圧縮についてのフラグ  
(0 : 圧縮していない 1 : 圧縮している)

# リカバリー

## 03

### 1. リカバリー

- データベースのリカバリー

### 2. リストア

- RESTORE DATABASEコマンドの主なオプション
- リストアの考慮点
- リストアのパフォーマンスの最適化

### 3. ロールフォワード

- ROLLFORWARD DATABASEコマンドの主なオプション
- ロールフォワードの考慮点
- オフライン・バックアップからのリカバリー
- オンライン・バックアップからのリカバリー
- データベース・バックアップからのリカバリーとデータベースの状態

### 4. ロールフォワードに必要なとなるログ

- Db2ログファイルの所在
- リカバリーに必要なとなるログの確認方法
- ロールフォワード処理において、Db2がログを検索するパス
- バックアップ・イメージに含まれるログファイルを利用したい場合

### 5. モニター

- リストアとロールフォワードのモニター

# データベースのリカバリー

## リストア

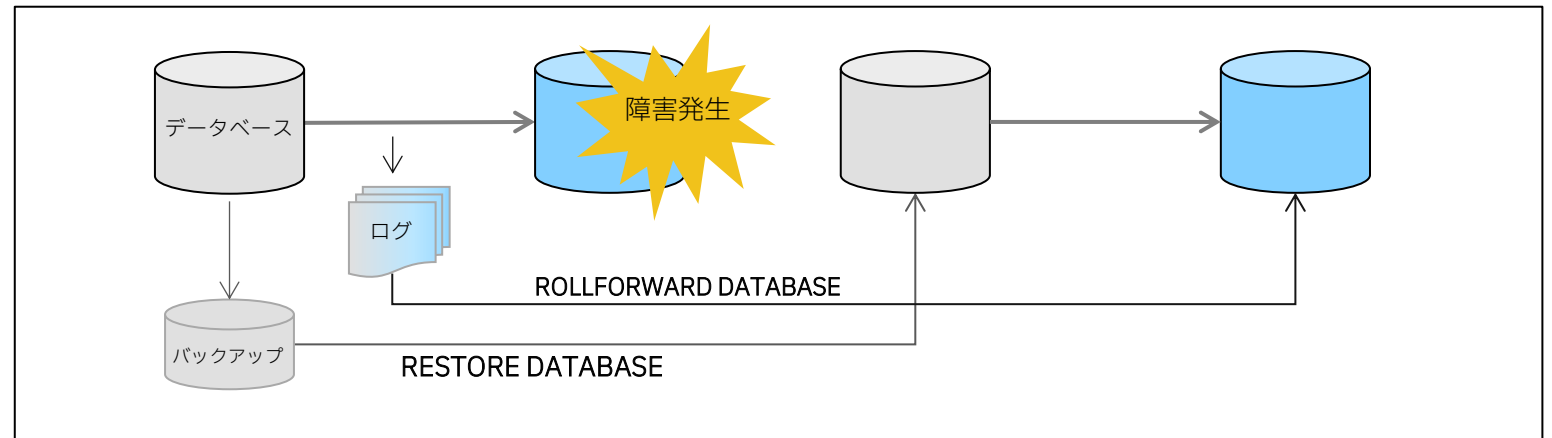
### – RESTORE DATABASEコマンド

- BACKUPコマンドを使用してバックアップされたデータベースをリストアするコマンド
- バックアップ・イメージをリストアし、バックアップ取得時のデータベースを作成する
- リストア後、データベースは「ロールフォワード・ペンディング」状態になる
  - ただし、循環ロギングの場合はならない

## ロールフォワード

### – ROLLFORWARD DATABASEコマンド

- ログ・ファイルに記録されたトランザクションを適用することで、データベースを回復させる
- 「ロールフォワード・ペンディング」状態のデータベースに対してのみ実行可能
- ロールフォワードの停止（STOPオプション）までは、繰り返しロールフォワードの実行可能
  - STOP後はロールフォワードできない



# リカバリー

## 03

### 1. リカバリー

- データベースのリカバリー

### 2. リストア

- RESTORE DATABASEコマンドの主なオプション
- リストアの考慮点
- リストアのパフォーマンスの最適化

### 3. ロールフォワード

- ROLLFORWARD DATABASEコマンドの主なオプション
- ロールフォワードの考慮点
- オフライン・バックアップからのリカバリー
- オンライン・バックアップからのリカバリー
- データベース・バックアップからのリカバリーとデータベースの状態

### 4. ロールフォワードに必要となるログ

- Db2ログファイルの所在
- リカバリーに必要となるログの確認方法
- ロールフォワード処理において、Db2がログを検索するパス
- バックアップ・イメージに含まれるログファイルを利用したい場合

### 5. モニター

- リストアとロールフォワードのモニター

# RESTORE DATABASEコマンドの主なオプション

---

## REBUILD

- 表スペースからデータベースをリストアするときに指定

## TABLESPACE

- 表スペースのみをリストアするときに指定、リスト指定も可能

## HISTORY FILE

- 回復履歴ファイルのみリストアするときに指定。

## LOG

- ログ・ファイルのみリストアするときに指定、同時にLOGTARGETオプションを指定する必要あり

## INCREMENTAL

- 増分バックアップからリストアするときに指定

## REDIRECT

- リストア時に表スペースのコンテナパスを変更した場合に指定

## REDIRECT GENERATE SCRIPT

- REDIRECTのスクリプトを生成するときに指定 詳細はP.108「リダイレクト・リストア」参照

## WITHOUT ROLLINGFOWARD

- リストア後にロールフォワードしないときに指定

# リストアの考慮点

---

## 既存のログ・パス上のログ・ファイルを退避させることを推奨

- リカバリーの操作の中でデータベースを削除した場合、ログ・ファイルも削除されるため、アクティブ・ログパス上のログ・ファイルを別のディレクトリに退避する
  - ・ 既存データベースにリストアする場合で、かつデータベース・シードが同じ場合は、ログ・ファイルは削除されない
- ロールフォワードは、最後のログ・ファイルまで処理可能だが、最後のログ・ファイルは、アーカイブされていない可能性がある。

## 複数世代の存在する場合は、バックアップ・ファイルを準備する

- 同じ場所に複数のバックアップ・ファイルが存在する場合には、TAKEN AT オプションでファイルを指定する
- バックアップをディスクに取得した後にテープに保管した場合は、アクセス可能なディレクトリにコピーする

## db2ckbkp コマンドを使用してバックアップ・ファイルの確認が可能

- 詳細は、P.31「db2ckbkp」参照

## 過去にバックアップしたデータベースを上書きする場合、データベースの構成パラメーターは（既存の）上書きされるデータベースのものがそのまま保存される

- 変更したパラメータ等があれば、記録しておく

# リストアのパフォーマンスの最適化

---

## Db2はリストア処理時間を短くするために、バッファの数、バッファ・サイズ、および並列処理設定の最適値を自動的に選択する

- 使用可能なユーティリティ・ヒープ・メモリーの大きさ、プロセッサの数、およびデータベース構成に基づく
- リストア時には、バックアップで使用されたバッファ・サイズの倍数が使用される
- データベース構成パラメータのBACKBUFSZ、RESTBUFSZは無視される

## 手動でパフォーマンス関連のパラメータを設定したい場合は、RESTORE時に以下を設定する

- BUFFER バッファサイズ
  - バックアップ・バッファで指定した倍数であること（最小値は8）
- WITH バッファの数 BUFFERS
  - バックアップ時に指定したバックアップ・バッファ・サイズに正の整数を乗算したサイズであること
    - 間違った値を指定すると、許容可能な最小サイズが割振られる
- PARALLELISM 並列度
  - データベースへの書き込みの並列度を指定する

# リカバリー

## 03

### 1. リカバリー

- データベースのリカバリー

### 2. リストア

- RESTORE DATABASEコマンドの主なオプション
- リストアの考慮点
- リストアのパフォーマンスの最適化

### 3. ロールフォワード

- ROLLFORWARD DATABASEコマンドの主なオプション
- ロールフォワードの考慮点
- オフライン・バックアップからのリカバリー
- オンライン・バックアップからのリカバリー
- データベース・バックアップからのリカバリーとデータベースの状態

### 4. ロールフォワードに必要となるログ

- Db2ログファイルの所在
- リカバリーに必要となるログの確認方法
- ロールフォワード処理において、Db2がログを検索するパス
- バックアップ・イメージに含まれるログファイルを利用したい場合

### 5. モニター

- リストアとロールフォワードのモニター

# ROLLFORWARD DATABASEコマンドの主なオプション

---

## TO 特定の時刻

- Point in Time ...指定した任意の時点までロールフォワードする

## END OF LOGS

- ログの最後までロールフォワードする

## COMPLETE/STOP

- ロールフォワードを停止し、未完了のトランザクションをロールバックし、データベースのロールフォワード・ペンディング状態をオフにして、ロールフォワード・リカバリーを完了させる

## TABLESPACE

- 表スペースのロールフォワードをする

## RECOVER DROPPED TABLE

- DROPされた表IDを使用して、ロールフォワードでリカバリーする ※詳細は、P.96「DROPされた表の回復」参照
- 表IDはLIST HISTORYコマンドで取得できる
- 回復される表は、dropped table recovery on で作成された表スペースに存在する必要がある

## OVERFLOW LOG PATH

- リカバリー中にアーカイブ・ログを検索する代替のログ・パスを指定する ※詳細は、P.50「ロールフォワード処理において、Db2がログを検索するパス」参照

## NORETRIEVE

- アーカイブ・ログの検索を無効にすることによりロールフォワードするログ・ファイルを制御する オンライン・バックアップから取り出したログ(include logs)を使用して end of backupでロールフォワードする場合に指定  
※詳細は、P.52「バックアップ・イメージに含まれるログファイルを利用する方法(2/2)」参照

# ロールフォワードの考慮点

---

## どの時点までリカバリするか検討する

- 最も新しいログ・ファイルまで
  - end of logs オプションを指定
- 指定した時間にコミットされているログ・レコードまで
  - to で時刻を指定
  - Point in Timeリカバリともいう

## 事前に適用するログ・ファイルを準備する

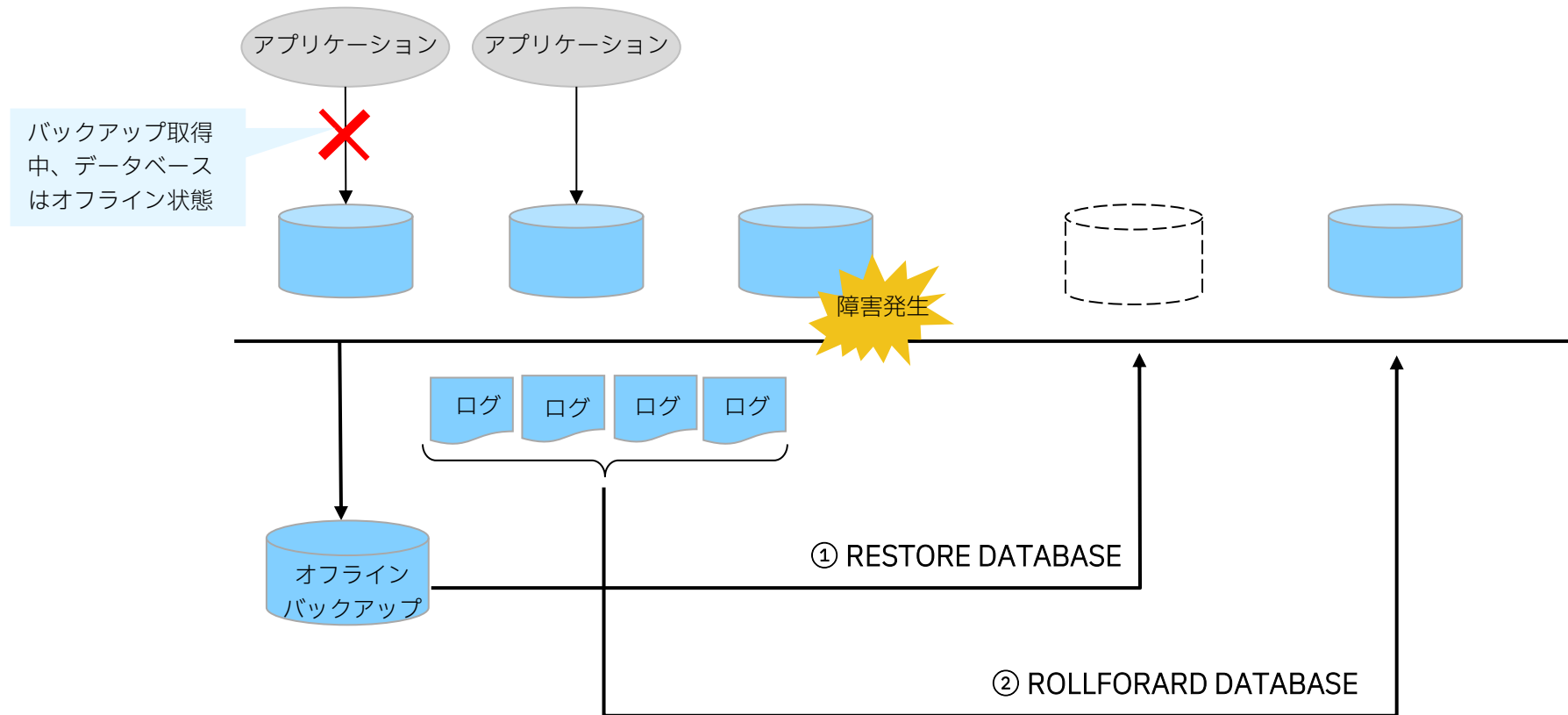
- ファイルはアクセス可能な場所に配置する
  - ログがテープなどに保存されている場合は、テープ等から読込先にコピーする
- ログパスにあるファイルが先に読み込まれるため、不要なファイルをログ・パスに配置しない
  - ログ・シーケンスが異なる同じ名前のログ・ファイルが存在しないようする

# オフライン・バックアップからのリカバリー

## 必要なファイル：

- オフライン・バックアップ・ファイル
- バックアップ取得終了後からリカバリーしたい任意の時点までのログ・ファイル

オフライン・バックアップを戻し(①)、バックアップ取得完了後からリカバリーしたい時点までのログを適用する(②)

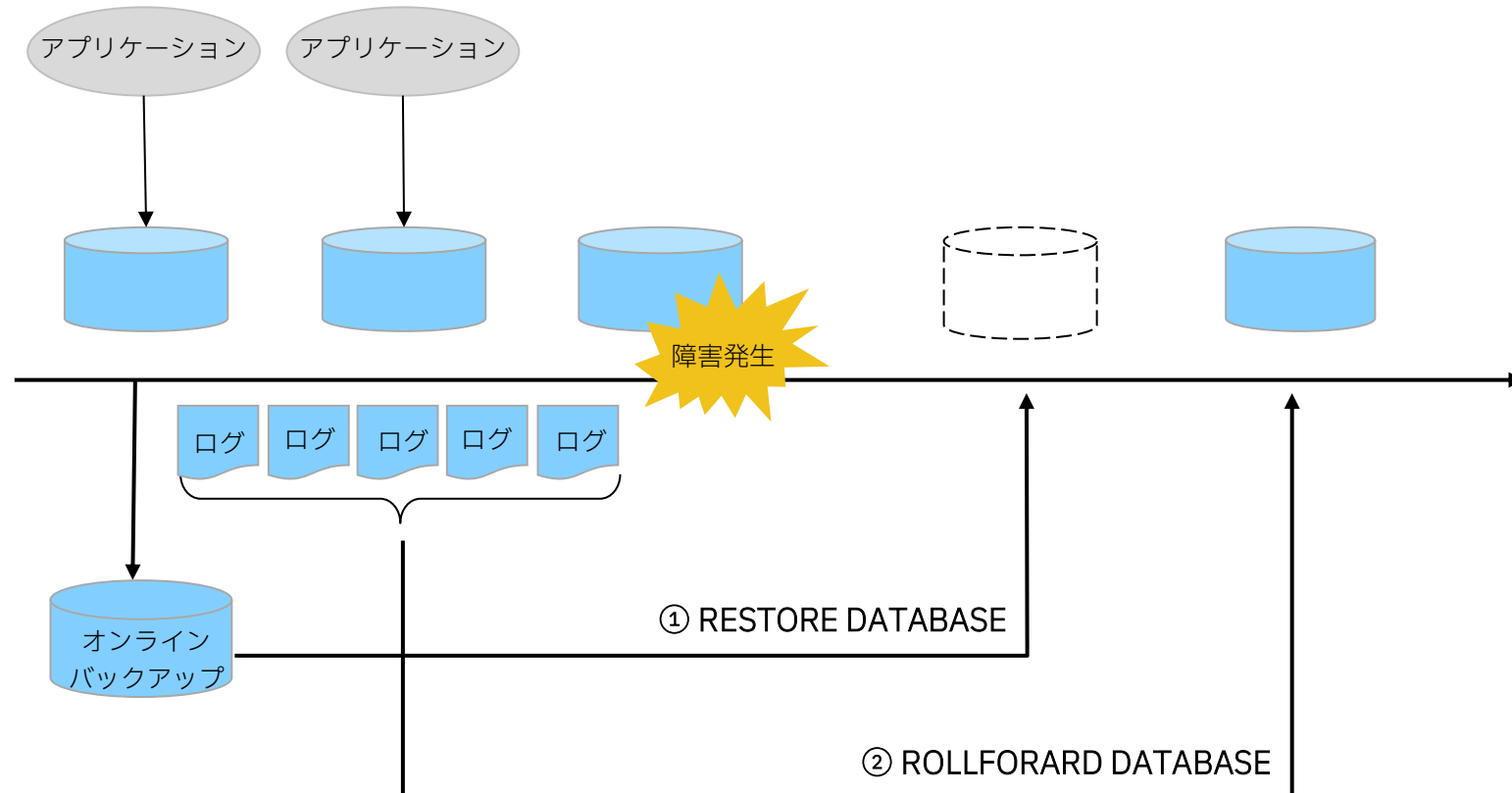


# オンライン・バックアップからのリカバリ

## 必要なファイル

- オンライン・バックアップ・ファイル
- バックアップ終了時点までのログ・ファイル
- リカバリしたい任意の時点までのログ・ファイル

オンライン・バックアップを戻し(①)、バックアップ取得中のログからリカバリしたい時点までのログを適用する(②)



# データベース・バックアップからのリカバリーとデータベースの状態

実行する操作後のデータベースの状態は以下のようになる

データベース全体に対する操作	操作後のデータベースの状態
データベース・バックアップ取得	正常
データベース・バックアップのリストア	ロールフォワード・ペンディング (DATABASE)
End of logs/Point in time までの ロールフォワード	正常
ロールフォワードのキャンセル	リストア・ペンディング

データベースの操作のペンディング状態はdb2 GET DB CFG コマンドで確認可能

```
$ db2 get db cfg
```

データベースのデータベース構成

```
データベース構成リリース・レベル          = 0x1600
データベース・リリース・レベル            = 0x1600
(省略)
```

```
バックアップ・ペンディング                  = NO
すべてのコミット済みトランザクションはディスクに書き込み済み = NO
ロールフォワード・ペンディング              = NO
リストア・ペンディング                      = NO
```

```
アップグレード・ペンディング                = NO
```

# リカバリー

## 03

1. リカバリー
  - データベースのリカバリー
2. リストア
  - RESTORE DATABASEコマンドの主なオプション
  - リストアの考慮点
  - リストアのパフォーマンスの最適化
3. ロールフォワード
  - ROLLFORWARD DATABASEコマンドの主なオプション
  - ロールフォワードの考慮点
  - オフライン・バックアップからのリカバリー
  - オンライン・バックアップからのリカバリー
  - データベース・バックアップからのリカバリーとデータベースの状態
4. ロールフォワードに必要となるログ
  - Db2ログファイルの所在
  - リカバリーに必要となるログの確認方法
  - ロールフォワード処理において、Db2がログを検索するパス
  - バックアップ・イメージに含まれるログファイルを利用したい場合
5. モニター
  - リストアとロールフォワードのモニター

# Db2ログファイルの所在(1/3)

Db2におけるログの出力先は、  
DB構成パラメータの設定値にて確認できる

## アクティブ・ログ・パス

–logpath

- 現行のアクティブ・ログ出力先

–mirrorlogpath

- ミラー・ログ・パス
- ログ・パスとミラー・ログ・パスの両方に  
アクティブ・ログ・ファイルが作成される
- ログ・データはすべて両方のパスに書き込まれる

## アーカイブ・ログ・パス

–logarchmeth1

- 現行ログ・パスからアーカイブされるログの1次宛先

–logarchmeth2

- 現行ログ・パスまたはミラー・ログ・パスの  
アーカイブ済みログの2次宛先

```
$ db2 get db cfg for mydb
```

```
データベースのデータベース構成 mydb
```

```
データベース構成リリース・レベル = 0x1600  
データベース・リリース・レベル = 0x1600
```

```
... (略) ...
```

```
ログ・ファイル用に変更されたパス (NEWLOGPATH) =  
ログ・ファイルのパス =
```

```
/home/uk1212/uk1212/NODE0000/SQL00003/LOGSTREAM0000/
```

```
ログ・パスのオーバーフロー (OVERFLOWLOGPATH) =  
ログ・パスのミラー (MIRRORLOGPATH) =
```

```
最初のアクティブ・ログ・ファイル = S0000002.LOG
```

```
... (略) ...
```

現時点でデータベースが実際に利用している範囲の中で  
最も古い(先頭の)アクティブ・ログ・ファイル名  
(これより古いファイルはアーカイブ済)

```
第 1 ログ・アーカイブ・メソッド (LOGARCHMETH1) = DISK:/data/uk1212/arclog/  
logarchmeth1 のアーカイブ圧縮 (LOGARCHCOMPR1) = OFF
```

```
logarchmeth1 のオプション (LOGARCHOPT1) =
```

```
第 2 ログ・アーカイブ・メソッド (LOGARCHMETH2) = OFF
```

```
logarchmeth2 のアーカイブ圧縮 (LOGARCHCOMPR2) = OFF
```

```
logarchmeth2 のオプション (LOGARCHOPT2) =
```

```
... (略) ...
```

# Db2ログファイルの所在(2/3)

## DBサーバー上に、いつの分からのログが存在するかの確認方法

→ OSコマンド(ls等)によって、前ページで確認したログパス上に存在するログのファイル名(番号)を確認できる

### アクティブ・ログ・パス

```
$ ls -ltR /home/uk1212/uk1212/NODE0000/SQL00003/LOGSTREAM0000/  
/home/uk1212/uk1212/NODE0000/SQL00003/LOGSTREAM0000/:  
合計 20040  
-rw-----. 1 uk1212 db2iadm1 4104192  8月 29 12:14 S0000006.LOG  
-rw-----. 1 uk1212 db2iadm1 4104192  8月 29 12:14 S0000003.LOG  
-rw-----. 1 uk1212 db2iadm1 4104192  8月 29 11:35 S0000005.LOG  
-rw-----. 1 uk1212 db2iadm1 4104192  8月 29 11:34 S0000004.LOG  
-rw-----. 1 uk1212 db2iadm1 4104192  8月 29 11:34 S0000002.LOG
```

現在Db2が利用している  
アクティブ・ログ・ファイル

### アーカイブ・ログ・パス

```
$ ls -ltR /data/uk1212/arclog/uk1212/MYDB/NODE0000/LOGSTREAM0000/  
/data/uk1212/arclog/uk1212/MYDB/NODE0000/LOGSTREAM0000/:  
合計 0  
drwxr-x---. 2 uk1212 db2iadm1 66  8月 29 11:34 C0000000  
  
/data/utsumi/uk1212/arclog/uk1212/MYDB/NODE0000/LOGSTREAM0000/C0000000:  
合計 12024  
-rw-r-----. 1 uk1212 db2iadm1 4104192  8月 29 11:34 S0000002.LOG  
-rw-r-----. 1 uk1212 db2iadm1 4104192  8月 29 11:34 S0000001.LOG  
-rw-r-----. 1 uk1212 db2iadm1 4104192  8月 29 11:33 S0000000.LOG
```

完了済トランザクションの記録  
されるログ・ファイル

# Db2ログファイルの所在(3/3)

## 現在、Db2が利用しているアクティブ・ログの確認方法

- db2pd -logs コマンドにより、障害発生直前の状態へのロールフォワード回復(End-of-logs) のために必要となるログファイルを確認できる
- 現時点のアクティブ・ログに含まれない、これより古いトランザクションも回復対象に含める場合には、追加でアーカイブログも必要となる

```
$ db2pd -d mydb -logs

Database Member 0 -- Database MYDB -- Active -- Up 0 days 01:17:02 -- Date 2025-08-29-12.47.09.618728

Logs:
Current Log Number          3
Pages Written                886
Cur Commit Disk Log Reads  0
Cur Commit Total Log Reads 0
Method 1 Archive Status     Success
Method 1 Next Log to Archive 3
Method 1 First Failure      n/a
Method 2 Archive Status     n/a
Method 2 Next Log to Archive n/a
Method 2 First Failure      n/a
Last Log Validation Error   n/a
Extraction Status           n/a (0)
Extraction Throttle Reason  n/a
Current Log to Extract      n/a
Log Chain ID                0
Current LSO                  72903411
Current LSN                  0x00000000000073535
```

注：  
db2pdコマンドは、DBが活動化されている状態で実行可能なコマンドであり、DB障害(ダウン)中に実行することはできません。リカバリー中にdb2pdコマンドが有効活用できるのは、ロールフォワード中のログ適用状況や問題判別などの用途です。

現在Db2が利用している  
アクティブ・ログ・ファイル

Address	StartLSN	StartLSO	State	Size	Pages	Filename
0x00007F122313D150	000000000000654A3	65216001	0x00000000	1000	1000	S0000002.LOG
0x00007F122D1D2150	0000000000006CC68	69292001	0x00000000	1000	1000	S0000003.LOG
0x00007F122D1D8950	00000000000000000	73368001	0x00000000	1000	1000	S0000004.LOG
0x00007F12225129D0	00000000000000000	77444001	0x00000000	1000	1000	S0000005.LOG
0x00007F122D1DD530	00000000000000000	81520001	0x00000000	1000	1000	S0000006.LOG

# リカバリーに必要なとなるログの確認方法

[重要] リカバリーを開始する前に、必要なログが揃っているかどうか、必ず事前に確認する

- オンライン・バックアップ・イメージには、取得時点で更新中のページも含まれる
- このため、オンライン・バックアップからのリカバリーを成功させるには「バックアップ開始時点」以降のログが必要
- バックアップ開始時点のログがDBサーバー上に存在しない場合、バックアップ・イメージから抽出することも可能 (→次ページ)

## リカバリーに必要なログの確認方法

### ① リカバリー履歴ファイルによる確認 (リストア開始前)

```
$ db2 list history backup all for sample
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
--- ---
B D 20250820192446001 N D S0000019.LOG S0000022.LOG
-----
6 個の表スペースを含みます:
```

00001 SYSCATSPACE  
00002 USERSPACE1  
... (略)

このバックアップをリカバリするために必要な最も古いトランザクションログファイル

バックアップ取得完了時点において最新だったログファイル

### ② コマンドによる確認 (リストア完了後)

```
$ db2 rollforward db sample query status
```

ロールフォワード状況

入力データベース別名 = sample  
状況を返したメンバーの数 = 1

メンバー ID = 0  
ロールフォワード状況 = DB ペンディング  
次に読み込むログ・ファイル = S0000019.LOG  
処理したログ・ファイル = -  
最後にコミットしたトランザクション = 2025-08-20-10.24.47.000000 UTC

リカバリー履歴ファイルで確認したEarliest Logと一致

# ロールフォワード処理において、Db2がログを検索するパス

---

## ROLLFORWARD DATABASE コマンドは、以下の順にログを検索する

- 同一名称のログファイルがサーバー上に複数存在している場合、先に見つかったログファイルが利用される
  1. アクティブログパス
  2. オーバーフローログパス (※オーバーフローログパスが設定されている場合)
  3. アーカイブログパス

## オーバーフローログパスの設定方法

- DB構成パラメータ overflowlogpath
  - ログファイルが配置されるディレクトリを絶対パス名で指定。空ストリング指定不可。
- ROLLFORWARD DATABASE コマンド OVERFLOW LOG PATH オプション
  - データベース構成パラメータ overflowlogpathの値をオーバーライドする

## オーバーフローログパスの設定例

```
$ db2 rollforward db sample to 2025-04-01-23.00.00 and stop overflow log path (/logs)
```

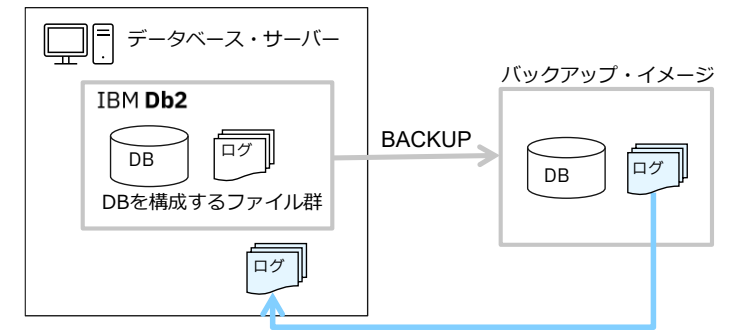
# バックアップ・イメージに含まれるログファイルを利用する方法(1/2)

Db2のオンライン・バックアップを取得すると、リカバリに必要な範囲のログファイルもバックアップ・イメージ内に自動で格納される (※exclude logsオプション指定時を除く)

リストア時にLOGSオプションを指定することにより、ログ・ファイルのみを抽出することができる

–RESTORE DATABASE コマンド・オプション

- LOGS: バックアップ・イメージから、ログファイルのみリストアする場合に指定するオプション
- LOGTARGET: ログ・ファイルの出力先を指定
  - ログ・パスは、完全修飾のパス名で、かつ、使用可能であること
  - ログ・パス上に既存の同名のログ・ファイルが存在しないこと
  - 同名のログ・ファイルがあった場合でも、上書きはされない



## バックアップ・イメージからログファイルを取得するコマンド実行例

```
$ RESTORE DB sample LOGS FROM bk_path LOGTARGET log_target
```

### <補足>

- 必ずしもバックアップ・イメージに含まれるログファイルを使用する必要はない
- ストレージ障害によりユーザーデータが破損している場合、アクティブログ、アーカイブログ領域には問題がない場合には、アクティブログ、アーカイブログ領域に残存するログファイルを利用することも多い

# バックアップ・イメージに含まれるログファイルを利用する方法(2/2)

---

バックアップ・イメージから抽出したログを利用して、ROLLFORWARD コマンドを TO END OF BACKUP で実行する場合、NORETRIEVE オプションを指定して実行する

- オンライン・バックアップ・イメージから取り出したログを使用して TO END OF BACKUP でロールフォワードする場合、ログ・アーカイブ先の不要なログが取得され、ロールフォワード操作が失敗する可能性がある
- NORETRIEVE オプションを指定してロールフォワードを実行することで不要なログの探索が行われず、ロールフォワード操作失敗を回避できる

※詳細は下記をご参照ください。

[SQL1268N Roll-forward recovery stopped due to SQL2062N error when trying to restore from an online backup using the INCLUDE LOGS option](#)

## ROLLFORWARD コマンドで TO END OF BACKUP 節と NORETRIEVE オプションを指定する実行例

```
$ ROLLFORWARD DB sample TO END OF BACKUP AND COMPLETE OVERFLOW LOG PATH (/log_target) NORETRIEVE
```

# リカバリー

## 03

### 1. リカバリー

- データベースのリカバリー

### 2. リストア

- RESTORE DATABASEコマンドの主なオプション
- リストアの考慮点
- リストアのパフォーマンスの最適化

### 3. ロールフォワード

- ROLLFORWARD DATABASEコマンドの主なオプション
- ロールフォワードの考慮点
- オフライン・バックアップからのリカバリー
- オンライン・バックアップからのリカバリー
- データベース・バックアップからのリカバリーとデータベースの状態

### 4. ロールフォワードに必要なとなるログ

- Db2ログファイルの所在
- リカバリーに必要なとなるログの確認方法
- ロールフォワード処理において、Db2がログを検索するパス
- バックアップ・イメージに含まれるログファイルを利用したい場合

### 5. モニター

- リストアとロールフォワードのモニター

# リストアとロールフォワードのモニター (1/2)

---

LIST UTILITIES やdb2pdコマンドを使用して、リストアやロールフォワードの処理状況を確認可能

## LIST UTILITIES SHOW DETAIL コマンド

- インスタンス上のアクティブなユーティリティーのリストを標準出力に表示
- 実行中の処理内容や開始時刻の確認が可能

## db2pd -d <データベース名> -utilities コマンド

- Db2 データベース・システム・メモリー・セット内からユーティリティー情報を取得
- 実行中の処理内容や開始時刻の確認が可能

## db2pd -d <データベース名> -recovery コマンド

- Db2 データベース・システム・メモリー・セット内からクラッシュ・リカバリーやロールフォワード・リカバリーに関する情報を取得
- 「Current Log」で今どのログファイル(ログ番号)をリカバリーに使用しているか確認可能 ※詳細は次ページ

# リストアとロールフォワードのモニター (2/2)

## db2pdコマンドのrecoveryオプションではロールフォワード処理で適用中のログファイルや処理済みバイト数の情報を取得可能

- ロールフォワード実行中のdb2pd -d <データベース名> -recovery の出力例
  - ロールフォワード・リカバリーにはFORWARDとBACKWARDの2つのフェーズがある
    - FORWARDフェーズでは、ログ・ファイルが読み取られて、そのログ・レコードがデータベースに適用される
    - BACKWARDフェーズでは、FORWARDフェーズで適用されたコミットされない変更は、ロールバックされる

```
$ db2pd -d sample -recovery

Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 00:00:18 -- Date 2025-09-11-17.42.27.469010

Recovery:
Recovery Status      9000008F40080801
Current Log          S0010575.LOG ← 現在リカバリーに使用されているログファイル
Current LSN          0000000001F11E70
Current LRI          000000000000001000000000004ABAA0000000001F11E70
Current LSO          36396554766
Job Type              ROLLFORWARD RECOVERY
Job ID                90
Job Start Time        (1757580129) Thu Sep 11 17:42:09 2025 ← リカバリーが開始された時刻
Job Description        Database Rollforward Recovery
Invoker Type          User
Total Phases          2
Current Phase          1

Progress:
Address              PhaseNum  Description              StartTime              CompletedWork          TotalWork
0x00000002042F82A8  1         Forward                  Thu Sep 11 17:42:09   12225434 bytes        Unknown
0x00000002042F8430  2         Backward                  NotStarted              0 bytes                Unknown
```

完了済み処理のバイト数

# 応用編

## Db2 バックアップ ・リカバリー

1. 増分バックアップとリカバリー
2. 表スペースバックアップとリカバリー
3. 圧縮バックアップからのリカバリー
4. バックアップ・イメージの暗号化
5. リモート・ストレージへのバックアップ
6. バックアップ・イメージの削除運用
7. RECOVERコマンドによるリカバリー
8. ストレージによる高速コピー機能を使用したDb2バックアップ
9. その他
  - LOAD後の回復とデータベース / 表スペースの状態
  - DROPされた表の回復
  - システム時刻変更に伴う注意点
  - 論理バックアップ
  - リダイレクト・リストア
  - REBUILD

# 増分バックアップとリカバリ

応用編

01

# 増分バックアップ

---

前回取得したバックアップから変更されたデータのみをバックアップする方法

## 特徴

- ページ単位で変更されたかどうか判断される
- 個々の増分バックアップ・イメージには、更新されたデータ・ページと索引ページに加えて、通常は全バックアップ・イメージに保管されるすべての初期データベース・メタデータ（データベースの構成、表スペースの定義、データベースのヒストリーなど）も含まれる
- オフライン・モードとオンライン・モードの操作の両方で、データベースと表スペースの増分バックアップを組み合わせることができる

## 種類

- INCREMENTAL（増分）
- DELTA（差分または増分差分）

## 前提条件

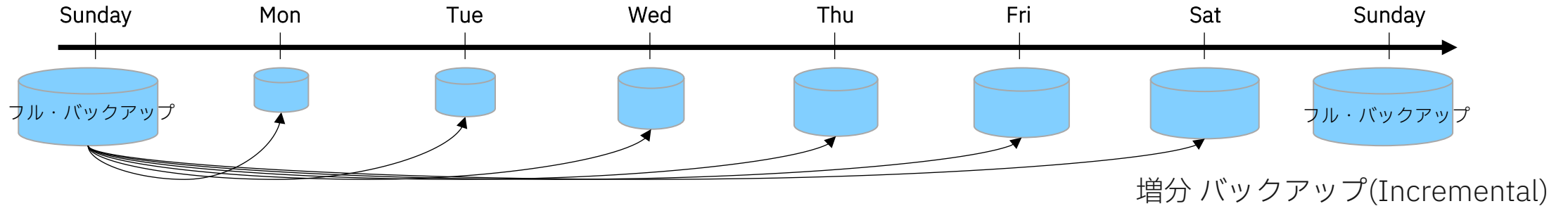
- 「変更ページの追跡使用可能」であること
  - データベース構成パラメーター trackmod = YES
- オンライン・バックアップの場合、データベースがアーカイブ・ロギング構成である
  - データベース構成パラメータ LOGARCHMETH1が以下のいずれかであること
    - LOGRETAIN, USEREXIT, DISK, DB2REMOTE, TSM, VENDOR
- オフライン・バックアップの場合は循環ロギングでも可能

# 増分バックアップの種類

主要な違いは、継続的に変更が加えられるオブジェクトのバックアップを連続して取る場合の動作

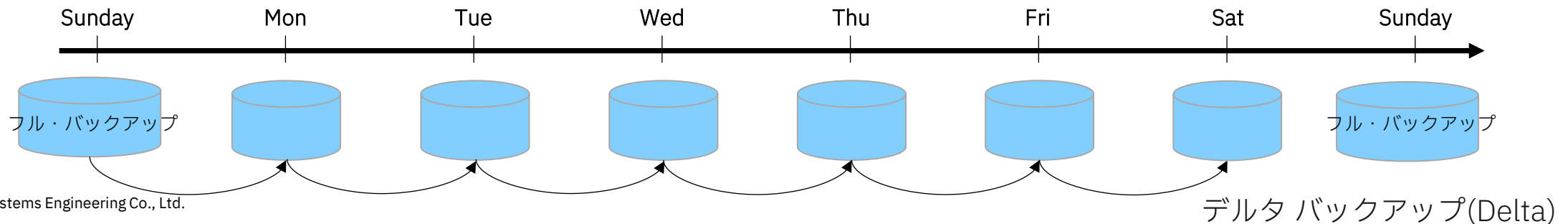
– 増分（累積：INCREMENTAL）

- 最後のフルバックアップから変更されたデータのバックアップ
  - 最新の正常実行された全バックアップ操作の後に変更された、すべてのデータベース・データのコピー（累積バックアップ・イメージ）
  - 増分バックアップを取るたびに、その前の増分バックアップ・イメージの内容が含まれる



– デルタ（非累積：DELTA）

- 最後のフルバックアップまたはインクリメンタルバックアップから変更されたデータのバックアップ
  - 当該表スペースの正常実行された最終バックアップ (全、増分、または差分) の後に変更されたすべてのデータベース・データのコピー（差分または非累積バックアップ・イメージ）
  - 差分バックアップ・イメージの先行処理イメージは、その差分バックアップ・イメージ中の個々の表スペースのコピーを含む、正常実行された最新バックアップになる



# 増分バックアップの考慮点

---

- ロング・フィールド/ラージ・オブジェクトが含まれる場合はページ単位の増分バックアップは行なわれない
  - 表スペースにロング・フィールド/ラージ・オブジェクトを含む表があり、その表に更新があった場合、その表スペース内の全テーブルの全ロング・フィールド/ラージ・オブジェクトデータがバックアップされる
  - 表スペースにロング・フィールド/ラージ・オブジェクトを含む表があり、ロング/LOBを含まない表に更新があった場合、更新のあったページと、表スペース内の全テーブルの全ロング/LOBデータがバックアップされる
    - ラージ・オブジェクト：長いデータを格納するためのデータ・タイプ（BLOB,CLOB,DBLOB）
    - ロング・フィールド：長い文字列を格納するためのデータ・タイプ（LONG VARCHAR, LONG VARCHARIC）※ロング・フィールドは非推奨
- 表スペースを作成した後は、一旦データベース全体のバックアップを取得しなければ、増分バックアップが取得できない
  - SQL2426エラーになる
- データベースと表スペースの増分バックアップを組み合わせした場合、データベース・バックアップ（または複数の表スペースのバックアップ）の先行イメージは必ずしも1つのイメージとは限らず、過去のさまざまな時点で取られたデータベースと表スペースのバックアップの固有の集合になる可能性も生じるので、バックアップ戦略を計画する際には注意する
  - データベースや表スペースを整合性のある状態にリカバリーするには、対象となるオブジェクト（データベースまたは表スペース）全体の整合性が保たれたイメージを使用してリカバリー・プロセスを開始する必要がある

# 増分バックアップからのリカバリー

---

## オフライン増分バックアップリカバリーに必要なファイル

- データベース全体のバックアップ・ファイル
- リカバリしたい最後の増分バックアップ・ファイル
- リカバリしたい任意の時点までのログ・ファイル

## オンライン増分バックアップからのリカバリーに必要なファイル

- データベース全体のバックアップ・ファイル
- リカバリしたい最後の増分バックアップ・ファイル
- バックアップ中のログ・ファイル
- リカバリしたい任意の時点までのログ・ファイル

## 必要なファイルは db2ckrst を使用して確認することが可能

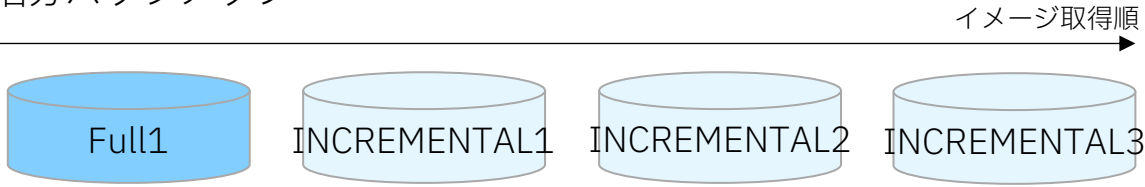
- 詳しくは後述

## リストアする2つの方法

- 手動増分リストア
  - リストアする必要があるバックアップ・イメージごとに、RESTORE コマンドを実行する
- 自動増分リストア（推奨）
  - RESTORE時に、AUTOMATICオプションを指定する
    - Db2が自動的に関連する増分バックアップ・ファイルが読まれ、ロールフォワード保留状態にする
    - その後 Db2 は、回復履歴を使用して残る必要なバックアップ・イメージを判別し、それらをリストアする

# オフライン増分バックアップからの手動回復と手順

## 増分バックアップ

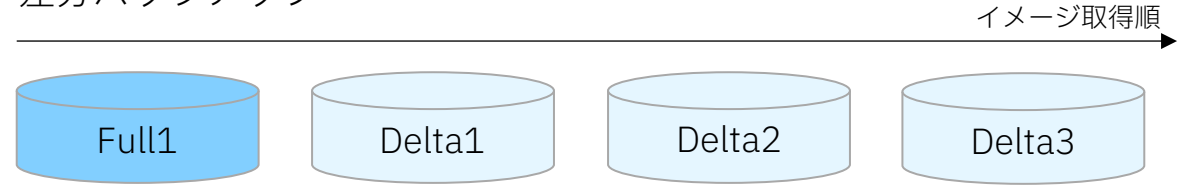


### 基本的な回復手順（増分：INCREMENTAL）

- ① 最後の増分バックアップをリストア
- ② データベース全体をリストア
- ③ 最後の増分バックアップを再度リストア
- ④ ロールフォワード

INCREMENTALバックアップには、最新のデータベース全体バックアップ・イメージからの更新データが含まれているため、最新の増分バックアップ・イメージがあればよい

## 差分バックアップ



### 基本的な回復手順（差分：DELTA）

- ① 最後の増分バックアップをリストア
- ② データベース全体をリストア
- ③ 最後の増分バックアップまでの増分バックアップをリストア
- ④ 最後の増分バックアップを再度リストア
- ⑤ ロールフォワード

DELTAバックアップには、前回のDELTAバックアップ・イメージからの更新データしか含まれていないため、回復した時点までのDELTAバックアップ・イメージが全て必要

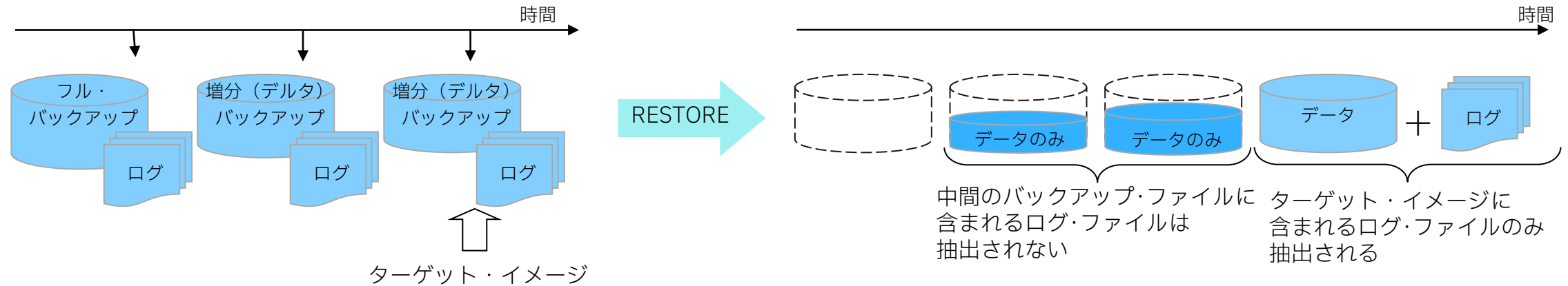
## どちらの手順もリカバリしたい最後の増分バックアップは2回リストアする必要あり

- リストア操作中に作成されるデータベースの正しいヒストリー、データベース構成、および表スペース定義でデータベースが確実に構成されるようにするため、増分リストア操作のターゲット・イメージは2回アクセスされなければならない
- 増分バックアップ・イメージのセットを復元するには、RESTORE DATABASE INCREMENTAL コマンドに TAKEN AT TIMESTAMP オプションで、復元する最終イメージのタイム・スタンプを指定する

# オンライン増分バックアップからの回復と手順

オンライン・バックアップを取得した場合、バックアップ・イメージをリストアした後、バックアップ取得中のログを使用して、最低限バックアップ終了時点までのロールフォワードが必要

– つまり、オンライン・バックアップをリストアするためには、オンライン・バックアップ取得中のログファイルが必ず必要



- 増分バックアップをオンラインで取得した場合、そのバックアップをリストアするために最低限必要なログファイルは、増分ターゲット・イメージを取得中のログファイルのみ
- それ以前の一連の増分バックアップ取得中のログファイルは、回復時に使用されない
- ただし、いずれかの増分バックアップが破損する可能性があるため、フル・データベース・バックアップ取得後のログは削除せず保管しておくことを推奨する
- INCLUDE LOGSの増分バックアップをリストアした場合、増分ターゲット・イメージに含まれるログ・ファイルのみ抽出される
  - 手動でリストアする場合は、最後のRESTOREコマンドのみ、LOGTARGETオプションを使用する

# db2ckrst

回復履歴ファイルを照会して、増分リストアに必要なバックアップ・イメージのタイム・スタンプのリストを生成する

## 手動増分リストアの簡単な RESTORE 構文も生成される

– 例

- db2ckrst -d SAMPLE -t 20250731111309 -r database
- db2ckrst -d SAMPLE -t 20250731111309 -r tablespace -n USERSPACE1

## コマンド・シンタックス

```
>>-db2ckrst-- -d--database name-- -t--timestamp----->
>--+-----+--+-----+----->
|      .-database---. | |      .------. |
'| -r--+tablespace-+-' | |      v          | |
|                          | |      '- -n----tablespace name-+-'|
>--+-----+-----><
+- -h-+
+- -u-+
'| -?-|
```

-d database name  
リストアされるデータベースの別名

-t timestamp  
増分をリストアするバックアップ・イメージの時刻

-r  
実行するリストアのタイプ

-n tablespace name  
リストアされる 1 つ以上の表スペース名

-h/-u/-?  
ヘルプ情報

# db2ckrst

## 実行例

```
$ ls  
  
SAMPLE.0.ki1212.DBPART000.20250731111027.001 → フルバックアップ  
SAMPLE.0.ki1212.DBPART000.20250731111135.001 } Incrementalバックアップ  
SAMPLE.0.ki1212.DBPART000.20250731111309.001 }  
  
$ db2ckrst -d SAMPLE -t 20250731111309 -r database  
  
Suggested restore order of images using timestamp 20250731111309 for  
database SAMPLE.  
=====
```

<pre>restore db SAMPLE incremental taken at 20250731111309 restore db SAMPLE incremental taken at 20250731111027 restore db SAMPLE incremental taken at 20250731111135 restore db SAMPLE incremental taken at 20250731111309</pre>	} Restore 実行例
--	---------------

```
=====
```

```
$ db2ckrst -d SAMPLE -t 20250731111309 -r tablespace -n USERSPACE1  
  
Suggested restore order of images using timestamp 20250731111309 for  
database SAMPLE.  
=====
```

<pre>restore db SAMPLE tablespace ( USERSPACE1 ) incremental taken at 20250731111309 restore db SAMPLE tablespace ( USERSPACE1 ) incremental taken at 20250731111027 restore db SAMPLE tablespace ( USERSPACE1 ) incremental taken at 20250731111135 restore db SAMPLE tablespace ( USERSPACE1 ) incremental taken at 20250731111309</pre>	} Restore 実行例
--	---------------

```
=====
```

# 表スペース・バックアップとリカバリー

応用編

02

# 表スペース・バックアップ・リストア

---

表スペース単位で、バックアップ・リストア・ロールフォワードが可能

## 特徴

- 表スペース単位のバックアップを取得する場合、バックアップ・サイズを小さく抑えることができる
- 特定の表スペースのバックアップを使用してリストアする場合は、リストアにかかる時間を短縮できる
  - フルバックアップから特定の表スペースのみ指定してリストアする場合は、リストア時間はデータベース全体のリストア時間と変わらない可能性がある
- リストア・ロールフォワードで表スペースを指定している場合、online オプションを使用できる。このとき、対象外の表スペースは online 状態で使用できる

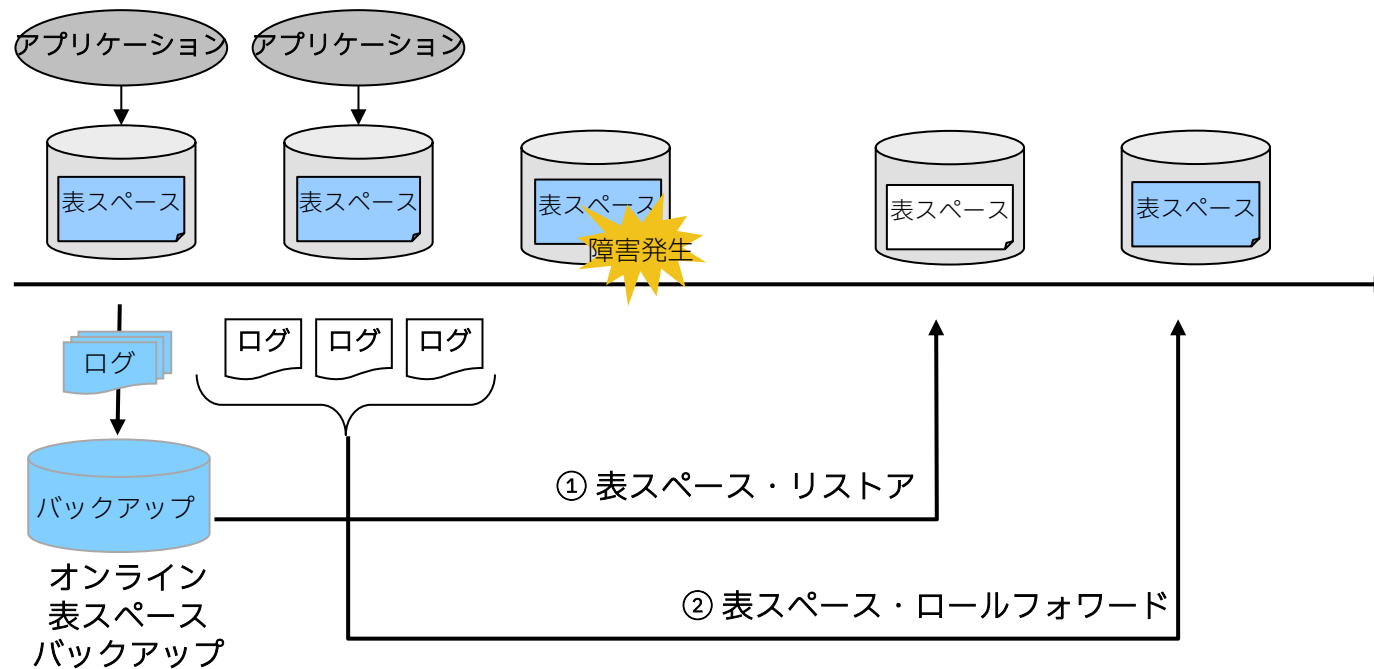
## 前提条件

- データベースが「リカバリー可能」であること
  - アーカイブ・ログを保持していて、ロールフォワード・リカバリーができる状態のこと
  - logarchmeth1 または logarchmeth2 データベース構成パラメーターが OFF 以外の値が設定されていることが条件
    - データベース構成パラメータ LOGARCHMETH1が以下のいずれかであること
      - LOGRETAIN：ログ・ファイルのアーカイブと取得をユーザーが担当することを指定
      - USEREXIT：ユーザー終了プログラムと呼ばれる、ユーザーによって提供されるプログラムを使用して、Db2、ログファイルのアーカイブと検索を管理することを指定
      - DISK：ログ・ファイルがアーカイブされる既存の絶対パス名を指定
      - DB2REMOTE：IBM Cloud Object Storage や Amazon Simple Storage Service (S3) などのリモート・ストレージにログ・ファイルがアーカイブされる
      - TSM：追加の構成パラメーターなしで指定した場合、ログ・ファイルはデフォルト管理クラスを使用してローカル TSM サーバーにアーカイブされる
      - VENDOR：ベンダー・ライブラリーを使用してログ・ファイルをアーカイブすることを指定

# 表スペース・バックアップからのリカバリ

## 必要なファイル

- データベース全体のバックアップ・ファイル、または表スペースのバックアップ・ファイル
  - ・ オンライン・バックアップの場合、バックアップの開始時点から終了時点までに生成されたログファイルも必要
- リカバリしたい任意の時点までのログ・ファイル



# バックアップからのリカバリーとデータベース/表スペースの状態遷移

表スペースに対する操作	操作後のデータベースの状態	操作後の表スペースの状態
バックアップ取得	正常	正常
バックアップのリストア	ロールフォワード・ペンディング (TABLESPACE)	ロールフォワード・ペンディング
End of logsまでのロールフォワード	正常	正常

表スペースに対する操作	操作後のデータベースの状態	操作後の表スペースの状態
バックアップ取得	正常	正常
バックアップのリストア	ロールフォワード・ペンディング (TABLESPACE)	ロールフォワード・ペンディング
Point in time までのロールフォワード	正常	バックアップ・ペンディング
DBのバックアップ取得	正常	正常

操作後の表スペースの状態は、End of logs と Point in time のロールフォワードで異なる。Point in time の場合は、その後に表スペースのバックアップ取得が必要になるので注意してください

# 表スペース・リカバリーの考慮点 (1/5)

---

## 一時表スペースを含めると、表スペース・バックアップはSQL2066Nエラーになる

- エラーメッセージ例：指定された表スペースが存在しないか、指定されたコマンドではサポートされていないため、コマンドは失敗しました。指定された表スペース名: TEMPSPACE1
- 理由：一時表スペースに含まれるデータは、バックアップ・イメージには含まれないが、一時表スペースの定義自体は、データベースのメタ・データに含まれる。そのため、バックアップ・イメージには一時表スペースも含まれているように見える (db2ckbkp -Tで確認可能)

## 表スペースのバックアップは、データベース全体のリカバリーには使用できない

- REBUILDは例外 (詳細はREBUILDの章を参照)

## DMS 表スペースのオンライン・バックアップ中は、以下の操作を同時に実行できない

- ロード
- 再編成 (オンラインおよびオフライン)
- 表スペースのドロップ
- 表の切り捨て (IMPORT/REPLACE)
- 索引の作成
- NOT LOGGED INITIALLY (CREATE TABLE および ALTER TABLE ステートメントと共に使用)

## 表スペースのバックアップをリストア後、その表スペースはロールフォワードが必要

## 表スペース・リカバリーの考慮点 (2/5)

---

ロールフォワード・ペンディング状態の表スペースへはアクセスできないが、他の表スペースにはアクセス可能

表スペースは、オンラインでリストアおよびロールフォワードすることが可能。ただし、カタログ表スペースは例外であり、オンラインでリカバリーすることはできない

–表スペースをオンラインでリストア、ロールフォワードする場合、他の表スペースはオンラインで使用可能

相互に依存している表スペースは、同時にリカバリーする必要がある

– 例えば、以下のような場合が該当する:

- LONG/LOBデータを別の表スペースに格納している場合
- パーティション表で複数の表スペースを使用している場合
- 索引とデータが別の表スペースに存在している場合
- MQT、ステージング表とその基礎表
- 参照制約のある表

など

– 同時にリカバリーされない場合、表スペースのロールフォワード処理で、「SQL4906N 指定された表スペース名のリストがロールフォワード操作に対して不完全です。」というエラーになる

表スペースの名前を変更した場合、変更後の名前を指定してリストア、ロールフォワード（変更した時点まで）が必要

# 表スペース・リカバリーの考慮点 (3/5)

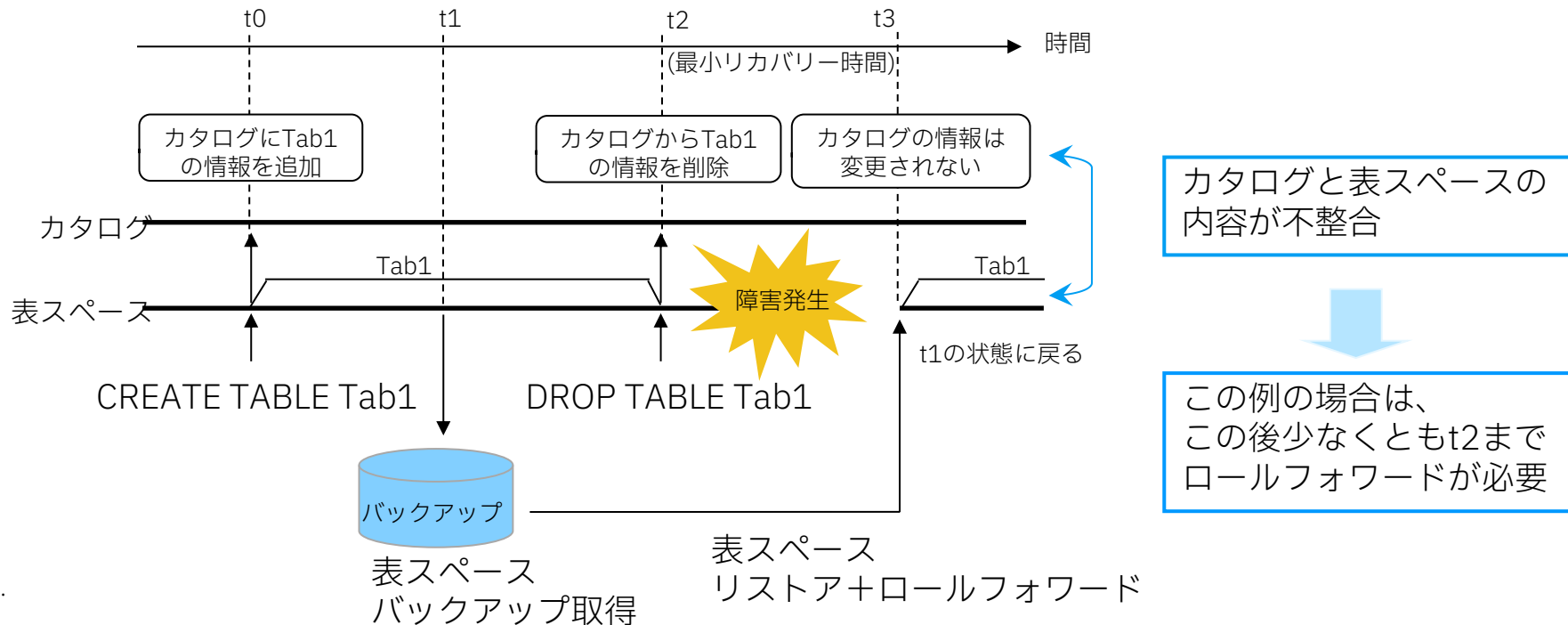
削除された表スペースは、その表スペースのバックアップからではリストアできない

- たとえ、リストア前に同じ名前で表スペースを再作成してもリストアはできない

リストアする表スペースは、現在のデータベースのログ・シーケンスに関連している必要がある

少なくとも表スペースの最小リカバリ時間まではロールフォワードが必要

- 最小リカバリ時間とは、表スペースまたは表スペース内の表に対して、一番最近にDDL ステートメント実行された時間 (下の図を参照)
- List tablespaces コマンドで最小リカバリ時間を確認可能



# 表スペース・リカバリーの考慮点 (4/5)

---

## 表スペースのPoint in Timeリカバリーの注意点

- 表スペースをPoint in Timeでロールフォワード完了後、表スペースはバックアップ・ペンディングになる
  - ロールフォワード先のPoint in Timeと現在時刻との間の表スペースに対するすべての更新情報は失われるため、表スペースのバックアップが必要となる
  
- 以下のようなリカバリーを実施する場合は注意が必要
  - 表スペースをPoint in Timeでロールフォワード実行後の操作
    - Point in Timeで表スペースをロールフォワード実行後、表スペースはバックアップ・ペンディング
  - ⇒ 表スペースまたはデータベースのバックアップを取得
    - その時刻を越して、再度ロールフォワードを実行すると、表スペースはリストア・ペンディング
  - ⇒ 表スペースのバックアップをリストア
    - リストア後、表スペースはロールフォワード・ペンディング
  - ⇒ 表スペースのロールフォワード
  
- データベースのリストア後、表スペースのリストアをデータベース全体のロールフォワード前に実行することも可能
  - 表スペースに対してロールフォワードを2度行う手間を防ぐ

# 表スペース・リカバリーの考慮点 (5/5)

---

## 表スペースのPoint in Timeリカバリーの注意点

- カタログ表スペースはPoint in Time でロールフォワード不可能
  - カタログ表スペース (SYSCATSPACE)をリストアした場合、ログの最後までロールフォワードを行わなくてはならない。これはシステム・カタログは常に他のすべての表スペースと整合性を保たなくてはならないためである
  - カタログ表スペースに対して Point in Time を指定してロールフォワードを実行すると、「SQL1274N データベース “sample”にロールフォワード・リカバリーが必要であり、ポイント・イン・タイムはログの最後に設定する必要があります。」というエラーになる
- Point in Timeでロールフォワードする場合は、トランザクションが、ある表スペースではトランザクションがコミットされ、ある表スペースではロールバックされているといった状態にならないように注意する
  - quiesce tablespace for table コマンドなどを使用して静止点を作る

# 圧縮バックアップからのリカバリー

応用編

03

# バックアップ・イメージのデータ圧縮

バックアップ・ファイルを圧縮する

## 特徴

- BACKUPコマンドにCOMPRESSオプションを使用
- データベース、表スペース、増分バックアップにも使用可能
- 圧縮されているかはdb2ckbkpで確認可能

## 実行例

```
$ db2 backup db TPCDS10 compress
```

バックアップは成功しました。  
このバックアップ・イメージのタイム・スタンプは  
20250922112802 です。



# 圧縮バックアップの注意点

## 圧縮しない方法と比較してCPUの使用率は高くなる

- 圧縮効率や処理時間は、DBの内容によって異なるため、実際の環境において効果を測定する必要がある

## 以下の例は23GB程度のサイズのDBをCOMPRESSオプションのあり/なしでバックアップを実施した結果である

- 圧縮のケースではおよそ40%程度に縮小できた
- 圧縮のケースでは処理時間が増大し、100秒ほど差がある

※圧縮率、パフォーマンスはデータ内容に依存します。

	バックアップ・サイズ (バイト)	バックアップ 処理時間 (秒)	平均CPU使用率 (%)	Utility Heap使用量 (MB)	平均ディスクIO/sec (回)
COMPRESS指定無し	23,105,585,152	143.1	11	19.0	15045.8
COMPRESS指定	9,669,189,632	245.0	85	22.4	6160.3

↑  
圧縮率： 41.8%

↑  
171.2% に増加

※並列度についてはこちらで指定していないため、Db2が並列処理設定の最適値を自動的に選択して非圧縮のケースは10に、圧縮のケースでは5になっています

# 圧縮バックアップからの回復

## 圧縮されたバックアップのRESTOREが可能

- RESTOREコマンドは、対象バックアップ・イメージが圧縮されたものを自動判別
  - 圧縮されていれば、解凍しながらリストア実行

以下の例は前のページで取得したバックアップ・イメージを用いてリストアを実施した結果である

- 処理時間に大きな差はない
- 圧縮のケースではCPU使用率が增大している

※パフォーマンスはデータ内容に依存します。

	バックアップ・サイズ (バイト)	リストア処理時間 (秒)	平均CPU使用率 (%)	平均ディスクIO/sec (回)
COMPRESS指定無し	23,105,585,152	288.6	8.2	8357.5
COMPRESS指定	9,669,189,632	291.5	22.6	6272.2

# バックアップ・イメージの暗号化

応用編

04

# バックアップ・イメージの暗号化

Db2のネイティブ暗号化機能を使用し、データベース、データベース・バックアップ、またはその両方の暗号化が可能

## Db2のネイティブ暗号化機能

- Db2同梱のGSKitツールを使用した暗号化機能
  - サポートされているすべての Db2 プラットフォームおよび構成で無料で使用可能
- データベース・バックアップは、データベース自体が暗号化されているかどうかに関係なく暗号化することができる
- 個々のバックアップを手動で暗号化するには、Db2のBACKUPコマンドで ENCRYPT オプションを指定する
- バックアップを自動的に暗号化するには、ENCRLIB および ENCROPTS データベース構成パラメーターを設定する
  - ENCRLIB と ENCROPTS データベース構成パラメーターは、データベースがENCRYPTオプション付きで作成されると自動的に設定される
- 暗号化されたバックアップをリストアする場合、リストア先のシステムのキーストアにバックアップ時に使用したマスターキーが含まれている必要がある
  - 暗号化されたバックアップを既存のDB上にリストアする場合、特別なパラメーターは不要
    - RESTORE時には、既存のDBにおけるデータベース暗号化設定(構成パラメータ)が使用される
      - バックアップ時に暗号化された標準データベースは、暗号化されていないデータベースとしてもリストア可能
  - 暗号化されたバックアップを新規DBとしてリストアする場合は、ENCRYPTパラメーターの指定が必要
    - Db2のRESTOREコマンドのENCRYPTキーワードのパラメーターはデータベース暗号化時に使用するものと同じ
    - バックアップイメージとリストアで作成するDBとで、異なる暗号化設定も可能

機能概要や設定方法の詳細は、下記 Db2 for LUW セキュリティ・ガイドの Native Encryption編 P.35～  
「バックアップとリストア」をご参照ください。

[【SIL】 【IM】 Db2 for LUW セキュリティガイド](#)

# リモート・ストレージへのバックアップ

応用編

05

# リモート・ストレージへのバックアップ/リストア

---

## IBM Cloud Object Storageをはじめとしたリモート・ストレージへのDb2バックアップおよびリモートストレージからのDb2リストアが可能

- IBM Cloud Object StorageやAmazon Simple Storage Service(S3)上のリモート・ストレージをサポート
  - ・サポートされるリモート・ストレージは[リモート・ストレージ要件](#)参照
- 手順
  1. [CATALOG STORAGE ACCESSコマンド](#)を使用し、リモート・ストレージにアクセスするためのエイリアスを設定
  2. Db2のBACKUPコマンドのバックアップ先やDb2のRESTOREコマンドのリストア元として[DB2REMOTE識別子](#)を使用して、手順1.で設定したエイリアスを指定

IBM Cloud Object Storage へのバックアップおよびリストアの詳細手順は、  
下記検証レポートの「6-3 Object StorageへのBackup」 「6-3 Object StorageへのBackup」シート をご参照ください。  
[ISE知恵袋 DB2 v11.1移行及び機能検証結果レポート](#)

※こちらのレポートは公開範囲が限定されております。アクセス不可の場合、レポートご入用の際はIBM営業にお問い合わせください。

# バックアップ・イメージの削除運用

応用編

06

# データベース・リカバリー・オブジェクトの自動削除

Db2のBACKUPコマンドを用いてデータベースのフル・バックアップ(オンライン/オフライン)を取得する場合、DB構成パラメーターで定義された保管世代数や保持期間を超えたバックアップ・イメージおよび関連のログ・ファイル等が自動削除される仕組みが提供されている

Db2のBACKUPコマンドでのフルDBバックアップを定期的を取得する環境ではバックアップ世代管理の運用負荷を低減できるメリットがある

## 自動削除を有効にするための設定

以下のデータベース構成パラメーターの設定が必要

- **AUTO\_DEL\_REC\_OBJ**: リカバリー・オブジェクトの自動削除 **ON** 「NUM\_DB\_BACKUPS と REC\_HIS\_RETENTN の設定値に基づいてリカバリー履歴ファイルの項目が整理されるときに、関連オブジェクトの自動削除を有効にするか？」(デフォルト OFF)
- **NUM\_DB\_BACKUPS**: バックアップ保持数 「何世代前のバックアップまで保持するか？」(デフォルト 12)
- **REC\_HIS\_RETENTN**: バックアップ保持期間 「何日前のバックアップまで保持するか？」(デフォルト 90日)

## 自動削除されるオブジェクト

- 整理されたリカバリー履歴項目に関連づけられた以下のオブジェクトが自動削除される
  - 物理ログ・ファイル
  - バックアップ・イメージ
  - ロード・コピー・イメージ

## 自動削除される条件とタイミング

- AUTO\_DEL\_REC\_OBJ の設定が **ON** の場合、Db2のBACKUPコマンドでデータベースのフル・バックアップが取得されたタイミングで、NUM\_DB\_BACKUPS と REC\_HIS\_RETENTN の両方の設定値 (閾値) を超えてリカバリーに不要と判断された項目がリカバリー履歴ファイルから整理(削除)され、それと同時に、整理された項目に関連づけられたオブジェクトが自動削除される

# データベース・リカバリー・オブジェクトの自動削除 (実行例)

以下のデータベース構成パラメーターの設定でフル・データベース・バックアップを日次で取得している環境において、フル・データベース・バックアップが2つ以上保持されている場合に、Db2のBACKUP コマンドによるフル・データベース・バックアップが完了したタイミングで、2日以上経過しているバックアップ・イメージおよびそのバックアップに紐づくログ・ファイルが自動削除される

## データベース構成パラメーターの設定

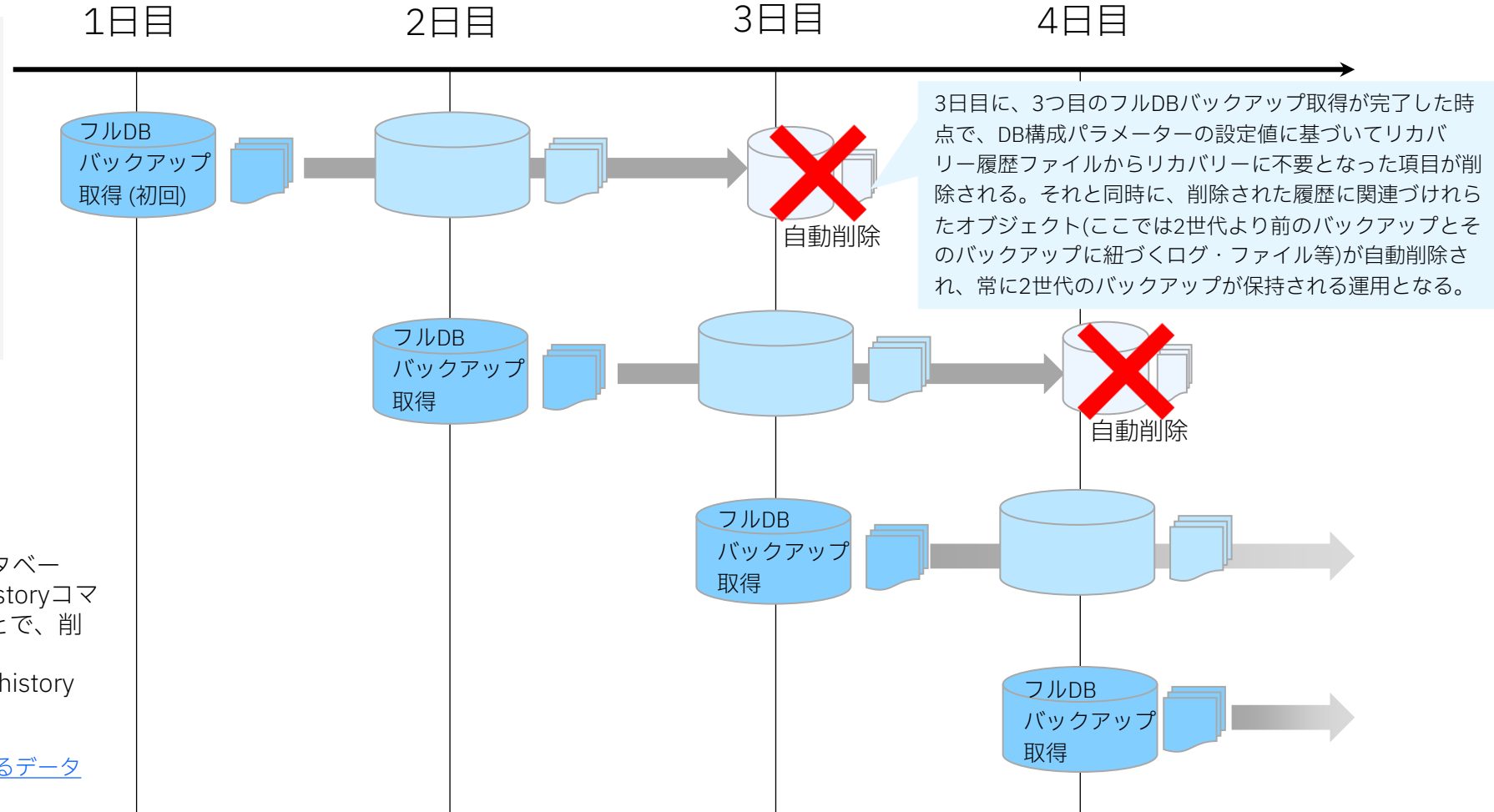
AUTO\_DEL\_REC\_OBJ = ON  
(リカバリー・オブジェクトの自動削除)  
NUM\_DB\_BACKUPS = 2  
(バックアップ保持数)  
REC\_HIS\_RETENTN = 2  
(バックアップ保持期間)

## 考慮点：

- ・ db2 backup database コマンドによる フル・データベース・バックアップ取得を行わない環境では、prune history コマンドを and delete オプションを指定して実行することで、削除された履歴項目に紐づくオブジェクトを削除可能
- ・ リカバリー履歴ファイルとその管理の詳細、prune history コマンドに関する詳細情報は以下を参照

[リカバリー履歴ファイル](#)

[PRUNE HISTORY コマンドまたは db2Prune API を使用するデータベース・リカバリー・オブジェクトの削除](#)



# RECOVERコマンドによるリカバリー

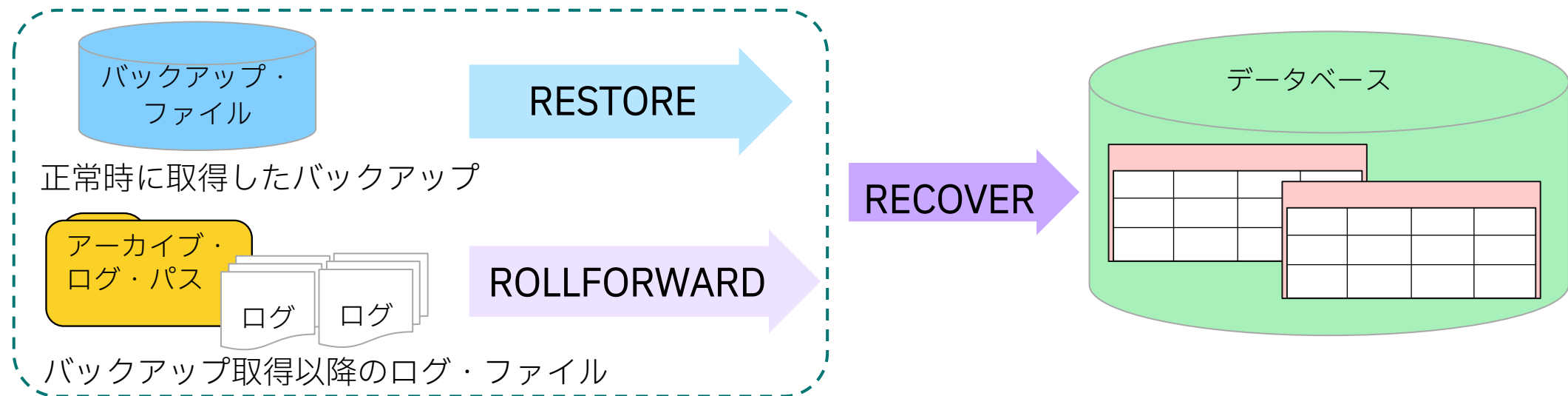
応用編

07

# RECOVERコマンドによるリカバリー

## RESTORE + ROLLFORWARD を一つのコマンドで実行する

- ROLLFORWARD は指定されたオプションに応じて、以下のいずれかの時点までトランザクションログの適応が可能
  - Point in Time (指定された日時までログを適用)
  - End of Logs (利用可能なすべてのログを適用)
  - End of Backup (バックアップ取得完了時点までのログのみ適用)
  - Complete (これ以降、ログの適用を行わず、ROLLFORWARD処理は完了扱いとなる)
- ROLLFORWARDを中断しても、途中から処理を再開することが可能



参考資料：[RECOVER DATABASE コマンド – IBM Documentation](#)

# ストレージによる高速コピー機能を使用したDb2バックアップ

## 応用編

# 08

※こちらのガイド内では概要のみ紹介します。  
下記の資料もご参照ください。

[【SIL】 【IM】 ストレージによる高速コピー機能を使用したDb2バックアップ](#)

# ストレージによる高速コピー機能を使用したDb2バックアップ

## 高速なバックアップ手段の必要性

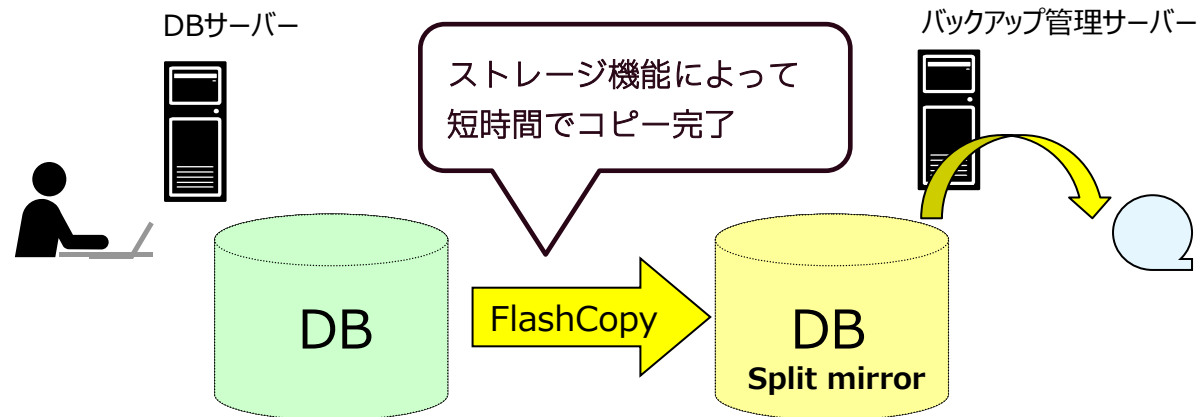
- Db2標準で提供されるBACKUPコマンドでは、データサイズが増えるほど、バックアップ取得にかかる時間が長くなる
- ストレージ・デバイスの提供する高速ストレージ・コピー機能(FlashCopy, Snapshot等)を利用すると、大規模DBのバックアップ・リストアを短時間で実行することができる

## ストレージ・デバイスで利用可能なハードウェア・ベースの高速コピー機能を使用して、データベースのコピー(Split mirror)を作成することが可能

- DS8000シリーズのFlashCopyなどを利用した、Diskボリュームコピー
- Db2マニュアルでは「Online split mirror / オンライン・スプリット・ミラー」と呼ばれる
  - ・バックアップの目的や、データベースのクローンとして再利用可能な、独立したデータベース・ディスク領域コピーのこと

## データベースのコピー(Split mirror)によって実現できること

- ・データベースのクローンを作成し、テスト環境や解析用データベースとして使用可能
- ・障害に備えて、ウォーム・スタンバイのデータベースを作成することが可能



# OnlineでのSplit mirror取得 – Db2 WRITE SUSPEND機能 –

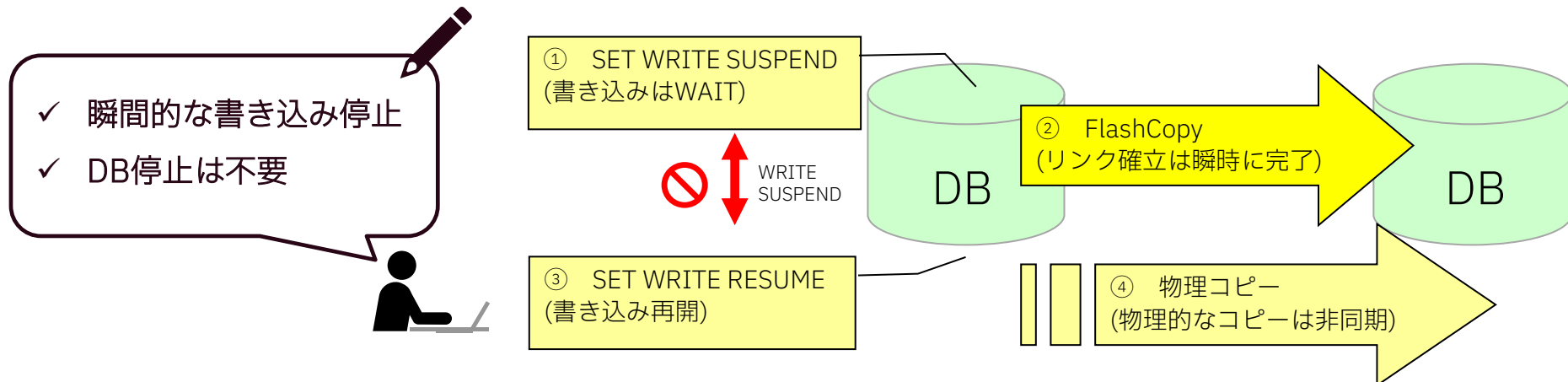
Db2を停止することなく、Online状態で整合性の取れたSplit mirrorイメージを取得するために、データベースへの書き込みを一時停止する機能

## set write suspend for database

- すべての表スペースとログに対する書き込みを禁止
- 書き込み禁止状態になったデータベースから、FlashCopyなどのストレージ・コピー機能を使ってSplit mirrorイメージを取得
- SELECT/INSERT/UPDATE/DELETE処理は、表スペースやログのDISKへの書き込みを発生しない限り実行可

## set write resume for database

- WRITE SUSPEND状態の解除
- Split mirrorイメージの取得完了後、直ちにこのコマンドで書き込み禁止状態を解除する



参考資料：[サスペンド入出力とオンライン・スプリット・ミラー・サポートによる高可用性 - IBM Documentation](#)

# FlashCopyによるバックアップ・リカバリー作業の全体フローの例

## FlashCopyによって取得したSplit mirrorイメージをdb2inidbの“as standby” オプションでリストアする手順例

### 1. データベースへ接続

```
db2 connect to <db_name>
```

### 2. すべてのバッファ・プールをフラッシュ

```
db2 flush bufferpools all
```

### 3. ログをアーカイブ

```
db2 "archive log for database <db_name>
```

### 4. データベースに対する入出力書き込み操作をサスペンド

```
db2 set write suspend for database
```

(FlashCopy取得操作の一例) (ストレージ・OSレイヤー)

5. OSのsyncコマンドを複数回(2-3回)実行して、OSのファイル・キャッシュに溜まったデータをディスクに書き出す

(AIX) sync

6. ファイルシステムをfreezeして、OSレイヤーの書き込み要求を停止する

(AIX) chfs -a freeze=<time\_in\_second> /fs\_name

7. FlashCopyを実施する

8. 論理コピー終了後、次のコマンドでファイルシステムのfreezeを解除してOSレイヤーの書き込みを再開する

(AIX) chfs -a freeze=off /fs\_name

### 9. データベースに対する入出力書き込み操作を再開

```
db2 set write resume for database
```

### 10. ログをアーカイブ

```
db2 "archive log for database <db_name>
```

### バックアップ手順

### 11. コピーされたSplit mirror領域のマウント

```
mount /dev/<lv_name> /db2data/<instance>/<db_name>
```

### 12. Db2起動

```
db2start
```

### 13. データベースの初期化 (スタンバイとして)

```
db2inidb <db_name> as standby
```

### 14. (別サーバー構成の場合のみ必要) ログファイルの転送

```
scp <primary_host>:/path/to/archive_logs/* /<secondary_log_dir>/
```

### 15. (オプション) ロールフォワードに必要なログファイル番号の確認

```
db2 rollforward database <db_name> query status
```

### 16. ロールフォワード

```
db2 rollforward db <db_name> to end of logs and stop
```

### 17. DB接続

```
db2 connect to <db_name>
```

---> DB接続成功

### リカバリー手順

db2inidbコマンドオプションは以下資料を参照

参考資料：[db2inidb - ミラーリングされたデータベースの初期化コマンド - IBM Documentation](#)

# その他

## 応用編

# 09

LOAD後の回復とデータベース / 表スペースの状態

DROPされた表の回復

システム時刻変更に伴う注意点

論理バックアップ

リダイレクト・リストア

REBUILD

# LOADとロールフォワードの考慮点

---

ロールフォワードは、データベース上で行われるすべての処理内容を復元できるわけではない

– ロールフォワードとは、ログを再適用することによって、データベースをある時点までの状態に戻す処理

ロールフォワード処理：

バックアップ・イメージのリストア

+

ログの再適用

ロギングされない処理はロールフォワードできない

- LOAD
- NOT LOGGED INITIALLY属性の表
- NOT LOGGED のLOBデータ 等



Tips:

次ページでは、LOAD処理の内容をロールフォワードにより再実行させたい場合の手法を示す

# LOADの処理内容のリカバリー

## LOADは投入データをトランザクションログに書かない

### – 最低限のロギングのみ

- ただし、DELETEフェーズではログが書かれるので、削除される行が多いとログ・フルになる可能性もある
- メリット：処理が速い
- 課題：このままでは、rollforwardできない

## LOADで行ったデータ投入処理を、ロールフォワード・リカバリーで回復(再実行)するには？

### – LOADコマンド実行時、COPY YES オプションを指定する

- LOAD時に、ロールフォワード・リカバリーに使用可能なロード・データのコピーファイルを指定されたパスに作成する
- ロールフォワード時は、COPY YES オプションで指定したパスからロードの際にコピーされたデータを取得し、LOADが再実行される
- 実行例

```
$ db2 "load from t1.csv of del replace into t1 copy yes to /work"
```

### • 作成されるファイル名の例

SAMPLE.4.uk1212.DBPART000.20250930192317.001

↑   ↑   ↑   ↑   ↑   ↑

DB名   インスタンス名   ノード   日時   シーケンス番号

タイプ: 「4」は LOAD... COPY TO コマンドによって生成されるバックアップ・イメージであることを示す

# LOAD COPYオプションと表スペースの状態

LOAD実行完了後の表スペースの状態は、LOADオプションおよび、データベースが回復可能(アーカイブロギング)モードかどうかによって異なる

LOAD オプション	DBのログ設定	LOAD後の 表スペースの状態	LOAD後に必要な処理	rollforwardリカバリー
COPY NO	循環ロギング	正常	無し	不可 (そもそも循環ロギングの場合、 rollforward不可)
COPY NO	アーカイブロギング	Backup Pending	表スペースまたはデータベースのバックアップを取得	LOAD後に取得したバックアップイメージを使用する
COPY YES	アーカイブロギング (循環ロギングでは指定不可)	正常	無し	可 rollforward時にLOAD COPYファイルが指定したパスにあることが必要(※)
NONRECOVERABLE	循環ロギング アーカイブロギング	正常	無し	アーカイブロギングのDBにおけるrollforwardにおいて、NONRECOVERABLE表が含まれていた場合、その表はDROP PENDINGになる

# その他

## 応用編

# 09

LOAD後の回復とデータベース / 表スペースの状態

DROPされた表の回復

システム時刻変更に伴う注意点

論理バックアップ

リダイレクト・リストア

REBUILD

# DROPされた表のリカバリー (1/7)

---

## 表スペースからのリカバリーで DROP された表をリカバリーすることが可能

- 表を誤ってDROPしてしまった場合、コミット後はメタデータがカタログ表から削除されるので、表スペース単位のバックアップ/リストアではこのカタログの状態を戻せない。そのため表スペースレベルのロールフォワードでは復旧できない。表をリカバリーするためには、以下の2つの方法がある
  - データベース全体のバックアップをリストアし、表が削除される直前までロールフォワードを行う必要がある。しかし、この方法では、表が削除された後のデータベースに対する更新は失われる。また、表が削除された時刻が最小リカバリー時間となるため、表スペースのバックアップ・リストアでは表をリカバリーすることはできない（表スペースは最小リカバリー時間以降までロールフォワードしなくてはならないため）
  - ROLLFORWARD DATABASE コマンドの RECOVER DROPPED TABLE パラメーターを使用して、表スペース・レベルのリストアおよびロールフォワード操作によりドロップされた表データをリカバリーする

## 前提

- データベースが「リカバリー可能」であること
- 表スペース作成時に dropped table recovery on を指定
  - alter tablespace で設定することも可能
    - デフォルトはON。ALTERにて変更可能
- ロールフォワード時に recover dropped table を指定
  - ロールフォワード処理中に削除された表のデータがDEL形式のファイルにEXPORTされる
  - ロールフォワードコマンドには、削除された表のIDと EXPORT ファイルを保管するディレクトリを指定しなくてはならない
    - LIST HISTORY DROPPED TABLE でロールフォワードの実行前にリカバリー履歴ファイルから表のIDを取得しておく必要がある
  - ROLLFORWARD DATABASE コマンドで RECOVER DROPPED TABLE パラメーターを使用し、表がDROPされた後のPoint in Timeまでロールフォワードする。または、ログの最後までロールフォワードする

# DROPされた表のリカバリー (2/7)

---

## 前提

- ロールフォワード時に recover dropped table を指定
  - 削除された表を再作成し、EXPORT されたデータを使用して、表に IMPORT または LOAD する
  - 例 DROPされた表 test\_drop\_recovery をリカバリー

```
$ db2 "INSERT INTO test_drop_recovery VALUES (1, 'AAA'), (2, 'BBB')"
```

```
DB20000I  SQL コマンドが正常に完了しました。
```

```
$ db2 "select * from test_drop_recovery"
```

```
ID          NAME
-----
          1 AAA
          2 BBB
```

2 レコードが選択されました。

```
$ db2 backup db SAMPLE online to /home/ly1212/backuptest include logs
```

バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは 20250929145229 です。

```
$ db2 "DROP TABLE test_drop_recovery"
```

```
db2 "COMMIT"
```

```
DB20000I  SQL コマンドが正常に完了しました。
```

```
DB20000I  SQL コマンドが正常に完了しました。
```

# DROPされた表のリカバリー (3/7)

## 前提

- ロールフォワード時に recover dropped table を指定
  - 例 DROPされた表test\_drop\_recoveryをリカバリー

```
$ db2 list history dropped table all for SAMPLE
```

SAMPLE の履歴ファイルのリスト

突き合わせファイル項目数 = 1

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
```

```
-----
```

```
D T 20250929145306
```

```
-----
```

```
000000000000bf0100020004
```

DROPされた表の ID

```
-----
```

```
"LY1212" "."TEST_DROP_RECOVERY" は 1 表スペースに常駐します:
```

```
00001 USERSPACE1
```

```
-----
```

```
Comment: DROP TABLE
```

```
開始時刻: 20250929145306
```

```
終了時刻: 20250929145306
```

```
状況: A
```

```
-----
```

```
EID: 7
```

表を再作成するため、DDLが必要

```
-----
```

```
DDL: CREATE TABLE "LY1212" "."TEST_DROP_RECOVERY" ( "ID" INTEGER , "NAME" VARCHAR(20 OCTETS) ) IN
```

```
"USERSPACE1" ORGANIZE BY ROW;
```

```
-----
```

# DROPされた表のリカバリー (4/7)

## 前提

- ロールフォワード時に recover dropped table を指定
  - 例 DROPされた表test\_drop\_recoveryをリカバリー

```
$ db2 restore db SAMPLE from /home/ly1212/backupptest taken at 20250929145229 into SAMPLE
SQL2539W リストアするバックアップ・イメージに指定された名前がターゲット・データベースの名前と同じになっています。バック
アップ・イメージ・データベースと同じ既存データベースをリストアすると、現行のデータベースがバックアップ・バージョンによって上
書きされます。続けますか。 (y/n) y
DB20000I RESTORE DATABASE コマンドが正常に完了しました。
```

```
$ db2 rollforward db SAMPLE to end of logs recover dropped table 000000000000bf0100020004 to
/home/ly1212/backupptest/reco
```

入力データベース別名  
状況を返したメンバーの数

ロールフォワード状況

= SAMPLE

= 1

メンバー ID

= 0

ロールフォワード状況

= DB 作業中

次に読み込むログ・ファイル

= S0000005.LOG

処理したログ・ファイル

= S0000000.LOG - S0000003.LOG

最後にコミットしたトランザクション

= 2025-09-29-05.58.16.000000 UTC

RECOVER DROPPED TABLE パラメーターを使用し、ログの最後までロールフォワードする

```
DB20000I ROLLFORWARD コマンドが正常に完了しました。
```

# DROPされた表のリカバリー (5/7)

## 前提

- ロールフォワード時に recover dropped table を指定
  - 例 DROPされた表test\_drop\_recoveryをリカバリー

```
$ db2 rollforward db SAMPLE complete
```

### ロールフォワード状況

入力データベース別名	= SAMPLE
状況を返したメンバーの数	= 1
メンバー ID	= 0
ロールフォワード状況	= 非ペンディング
次に読み込むログ・ファイル	=
処理したログ・ファイル	= S0000000.LOG - S0000004.LOG
最後にコミットしたトランザクション	= 2025-09-29-05.58.16.000000 UTC

DB2000I ROLLFORWARD コマンドが正常に完了しました。

```
$ ls -R /home/ly1212/backupstest/reco  
/home/ly1212/backupstest/reco:NODE0000  
/home/ly1212/backupstest/reco/NODE0000:data
```

このエクスポート・ディレクトリーの下の子ディレクトリーは、各データベース・パーティションごとに自動的に作成される。ドロップされた表データを含むデータ・ファイルは、dataという下位のサブディレクトリーにエクスポートされる

# DROPされた表のリカバリー (6/7)

## 前提

– ロールフォワード時に recover dropped table を指定

- 例 DROPされた表test\_drop\_recoveryをリカバリー

```
$ db2 "CREATE TABLE test_drop_recovery (id INT, name VARCHAR(20)) IN USERSPACE1"
```

DB20000I SQL コマンドが正常に完了しました。

```
$ db2 import from /home/ly1212/backupptest/reco/NODE0000/data of del insert into test_drop_recovery
```

SQL3109N ユーティリティーが、ファイル/home/ly1212/backupptest/reco/NODE0000/data からデータのロードを開始しています。

SQL3110N ユーティリティーが処理を完了しました。 "2"行が、入力ファイルから読み取られました。

SQL3221W ...COMMIT WORK を開始します。入力レコード・カウント = "2"

SQL3222W ...データベース変更の COMMIT が成功しました。

SQL3149N "2" 行が、入力ファイルから処理されました。 "2"行が、正常に表に挿入されました。 "0"行が、拒否されました。

読み込まれた行数 = 2

スキップされた行数 = 0

挿入された行数 = 2

更新された行数 = 0

拒否された行数 = 0

コミットされた行数 = 2

# DROPされた表のリカバリー (7/7)

## 前提

- ロールフォワード時に recover dropped table を指定
  - 例 DROPされた表test\_drop\_recoveryをリカバリー

```
$ db2 "select * from test_drop_recovery"
```

```
ID          NAME
-----
          1 AAA
          2 BBB
```

2 レコードが選択されました。

## 考慮点

- リカバリー履歴ファイルから削除された表のIDと、作成するためのDDLを取得
  - list history dropped table コマンドを使用
- REGULAR/LARGE 表スペースのみサポート
- 一度にリカバリーできる DROP された表は一つだけ
- DROPされた表からリカバリー可能なデータのタイプについて、いくつかの制限事項がある。以下のものは、リカバリーすることはできない
  - DROPPED TABLE RECOVERY オプションは一時表には使用できない
  - 行タイプと関連したメタデータ (データはリカバリーされるが、メタデータはリカバリーされない)
  - XML データおよびカラム・オーガナイズ表
  - LONGデータ、LOBデータはリカバリーできない

# その他

## 応用編

# 09

LOAD後の回復とデータベース / 表スペースの状態

DROPされた表の回復

システム時刻変更に伴う注意点

論理バックアップ

リダイレクト・リストア

REBUILD

# システム時刻を変更にともなう注意点

---

運用上、Network Time Protocol などを使用して、ネットワーク上のマシンの時刻の同期を取ることがある。その際、システム (OS)の時間を進めるまたは戻すという操作が行われる可能性がある

## ログに対する影響

- システムの時間を進めた場合には、Db2に対する影響はない
- システムの時間を戻した場合にも、ログには順序番号があるため、不整合は生じない
  - タイムスタンプは、時間を戻す前の時間に追いつくまで、同じタイムスタンプを使用
    - 例えば、9:00 のタイムスタンプでログレコードが記録された後に、8:55 にシステム時間を戻したとしたら、システム時間が 9:00 を超えるまで、その間のログの各レコードは 9:00 のタイムスタンプを使用する

## Point in Time 指定でロールフォワードを行う場合の注意点

- 時間を戻す前の時間に追いつくまでに行われるトランザクションは、すべて同じタイムスタンプでログに記録
- そのタイムスタンプを指定してロールフォワードを行うと、時間を戻す前の時間に追いつくまでの間に行われるすべてのトランザクションがリカバリーされる
- 上の例では、バックアップ/リストアを行ったあとに、9:00 を指定してPoint in Time 指定でロールフォワードを行うと、システム時間を戻してからシステム時間が9:00 になるまでになされた更新はすべて実行されることになる

参考情報：[\[Db2\] システム・クロックの変更が Db2 に与える影響](#)

# その他

## 応用編

# 09

LOAD後の回復とデータベース / 表スペースの状態

DROPされた表の回復

システム時刻変更に伴う注意点

論理バックアップ

リダイレクト・リストア

REBUILD

# 論理バックアップと論理リストア

Db2論理バックアップおよびリストア機能を使用して、スキーマのフル、累積増分、差分増分のバックアップを行い、スキーマまたはスキーマ内のテーブルをフルリストアすることが可能

## 特徴

- スキーマ全体のバックアップやリストア、また、スキーマ・バックアップから1つ以上のテーブルのリストアが可能
- バックアップには、[LOGICAL BACKUP ストアード・プロシージャ](#)を使用
- リストアには、[LOGICAL RESTORE ストアード・プロシージャ](#)を使用

## 考慮点

- linuxamd64 および linuxppc64le プラットフォームでのみサポートされる(2025年10月現在)
- 論理バックアップ・リストア機能を使用するには、CREATE SCHEMA文またはALTER SCHEMA文によってスキーマに **ENABLE ROW MODIFICATION TRACKING** オプションを設定する必要がある
  - スキーマに対してENABLE ROW MODIFICATION TRACKINGオプションを設定する代わりに、データベース設定パラメーター [DFT\\_SCHEMAS\\_RMT](#) を”YES”に設定することで、パラメーター設定後に作成される全てのスキーマで暗黙的にROW MODIFICATION TRACKINGを有効にすることも可能
- 多くの制約や考慮点があるため、下記製品マニュアルを参照し、十分な検証を実施した上での使用が推奨される
  - [スキーマの LOGICAL BACKUP および LOGICAL RESTORE の有効化](#)
  - [論理バックアップとリストア](#)
  - [論理バックアップとリストアの制限](#)

# その他

## 応用編

# 09

LOAD後の回復とデータベース / 表スペースの状態

DROPされた表の回復

システム時刻変更に伴う注意点

論理バックアップ

リダイレクト・リストア

REBUILD

# リダイレクト・リストア

---

## リストア時に既存ストレージ・グループのパスの変更が必要な場合、リダイレクト・リストアを使用

- リストア・ユーティリティーはリストア先の物理的コンテナのレイアウトがバックアップ元のデータベースのレイアウトと同一であると想定しているため、リストア時にストレージ・パスに変更がある場合は、通常のリストアではなくリダイレクト・リストアを使用する

## リダイレクト・リストアに必要な手順 ご参考: [異なるストレージ・グループ・パスを使用した実動データベースのクローン作成](#)

1. REDIRECT オプションを指定して RESTORE DATABASE コマンドを実行し、リダイレクト・リストアを通知
2. SET STOGROUP PATHS コマンドを使用して、リストア先のストレージ・グループ・パスを定義
3. CONTINUEオプションを指定してRESTORE DATABASE コマンドを実行し、リストア先変更後のリストアを再開

## データベース・バックアップ・イメージからリダイレクト・リストアのためのスクリプトの生成

- REDIRECT オプションと GENERATE SCRIPT オプションを指定して RESTORE DATABASE コマンドを発行することにより、既存のバックアップ・イメージを元に上記手順を含んだリダイレクト・リストア実行のスクリプトが自動生成される
    - RESTORE(REDIRECT) → SET STOGROUP PATHS → RESTORE(CONTINUE)の一連のコマンドが生成スクリプトに記述される
- ```
$ RESTORE DB <DB名> FROM <BACKUPディレクトリ> TAKEN AT <BACKUP取得時間> REDIRECT GENERATE SCRIPT <出力スクリプト名>
```
- 生成されたスクリプトをリストア先の環境に合わせて修正して、db2 -tvf <出力スクリプト名> で実行する
    - 詳細は右記をご参照ください。 [RESTORE DATABASE コマンド \(例 10\)](#) / [自動生成スクリプトを使用したリダイレクト・リストアの実行](#)

# リダイレクト・リストアのためのスクリプト生成の実行例

生成されたスクリプトは4つのセクションで構成される

```
$ ls -/work
SAMPLE.0.db2inst1.DBPART000.20251007195518.001
$ db2 "RESTORE DB sample FROM /work REDIRECT GENERATE SCRIPT sample_redirect.clp"
DB20000I RESTORE DATABASE コマンドが正常に完了しました。
```

```
$ cat sample_redirect.clp
```

```
-- *****
-- ** 自動作成されたリダイレクト・リストア・スクリプト
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON SAMPLE_NODE0000.out V ON;
SET CLIENT ATTACH_MEMBER 0;
SET CLIENT CONNECT_MEMBER 0;
-- *****
-- ** 自動作成されたリダイレクト・リストア・スクリプト
-- *****
RESTORE DATABASE SAMPLE
-- USER <username>
-- USING '<password>'
FROM '/work'
TAKEN AT 20251007195518
-- ON '/home/db2inst1'
-- DBPATH ON '<target-directory>'
INTO SAMPLE
-- NEWLOGPATH DEFAULT
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
```

セクション1:  
初期設定

セクション2:  
RESTORE DATABASE  
REDIRECT コマンド

※DBパスやログパス等を  
変更可能

```
-- *****
-- ** ストレージ・グループ定義
-- ** デフォルトのストレージ・グループ ID = 0
-- ** ストレージ・グループ数 = 1
-- *****
-- ** ストレージ・グループ名 = IBMSTOGROUP
-- ** ストレージ・グループ ID = 0
-- ** データ・タグ = なし
-- *****
-- SET STOGROUP PATHS FOR IBMSTOGROUP
-- ON '/home/db2inst1'
-- ;
-- *****
-- ** 表スペース定義
-- *****
-- ** 表スペース名 = SYSCATSPACE
-- ** 表スペース ID = 0
-- ** 表スペース・タイプ = データベース管理スペース
-- ** 表スペース・コンテンツ・タイプ = すべての永続データ。REGULAR 表スペース。
-- ** 表スペース・ページ・サイズ(バイト) = 8192
-- ** 表スペース・エクステント・サイズ(ページ) = 4
-- ** 自動ストレージの使用 = はい
-- ** ストレージ・グループ ID = 0
(省略)
-- *****
-- *****
-- ** リダイレクト・リストアの開始
-- *****
RESTORE DATABASE SAMPLE CONTINUE;
-- *****
-- ** ファイルの終わり
-- *****
```

既存ストレージ・グループ  
のパスを環境に合わせて書  
き換え可能

セクション3:  
ストレージ・グループや  
表スペースの定義

※自動ストレージを使用して  
作成された表スペースのみの場合、  
SET TABLESPACE  
CONTAINERS 節は出力されない

セクション4:  
RESTORE DATABASE  
CONTINUE コマンド

# その他

## 応用編

# 09

LOAD後の回復とデータベース / 表スペースの状態

DROPされた表の回復

システム時刻変更に伴う注意点

論理バックアップ

リダイレクト・リストア

REBUILD

# REBUILDによるリカバリー

---

## 表スペース・バックアップから、データベースのリカバリーが可能

- データベース・バックアップを取得する頻度を減らすことができる

## すべての表スペース、または、一部の表スペースのみをリストアし、使用することが可能

- テスト環境において、一部の表スペースだけを使用したい場合
- データベース障害時に、一部の表スペースだけを先にリカバリーしたい場合、等

## データベースをリカバリーする際は、REBUILD対象に、必ずSYSCATSPACEが含まれることが前提

- REBUILD処理はSYSCATSPACEのメタ情報を参照するので、SYSCATSPACEの状態を他の表スペースと整合性を保つために、定期的にSYSCATSPACEを含めてRESTOREすることが推奨される

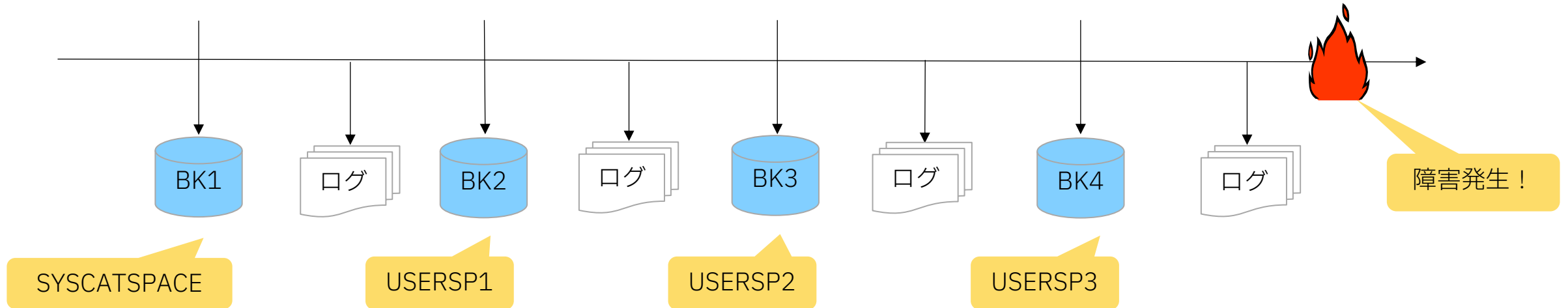
RESTORE DATABASEコマンドに新規追加された、REBUILDオプションを使用する。

# 参考：データベースのREBUILDの例

データベースバックアップは取得せず、表スペースバックアップのみを取得している例

Time ↓

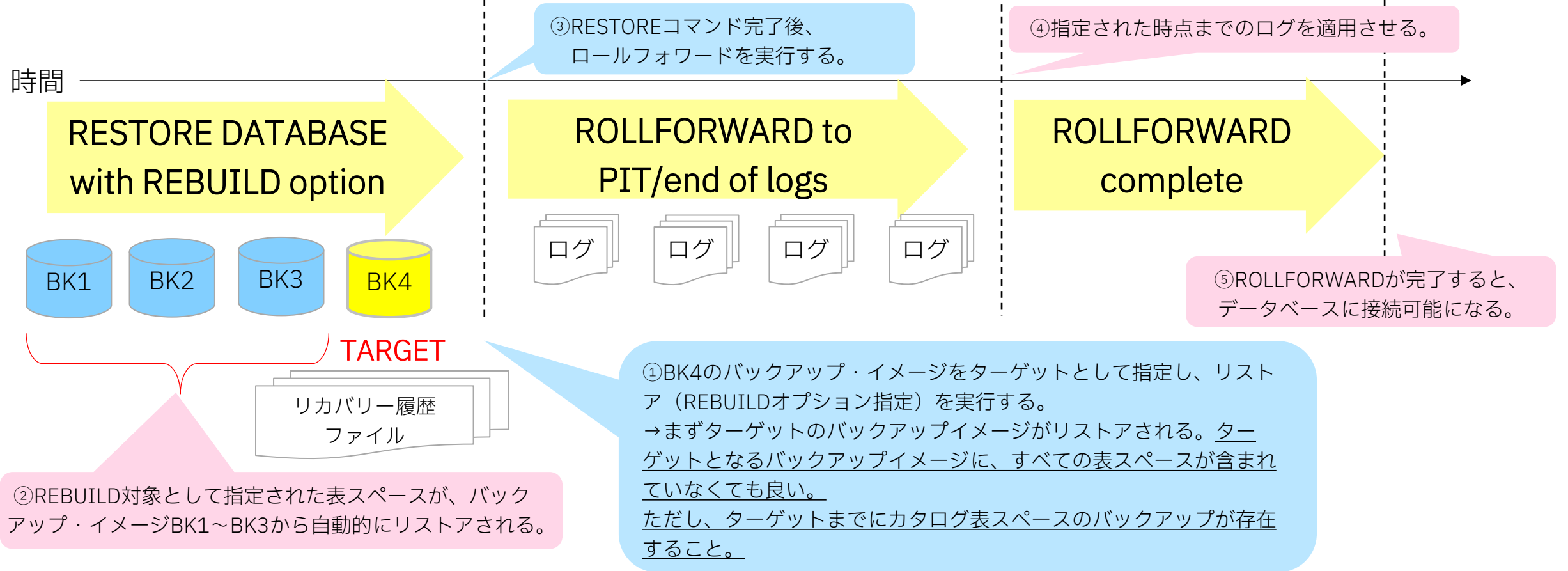
|     | SYSCATSPACE  | USERSP1      | USERSP2      | USERSP3      |
|-----|--------------|--------------|--------------|--------------|
| BK1 | 表スペース・バックアップ |              |              |              |
| BK2 |              | 表スペース・バックアップ |              |              |
| BK3 |              |              | 表スペース・バックアップ |              |
| BK4 |              |              |              | 表スペース・バックアップ |



# 参考：データベースのREBUILDの例（つづき）

## すべての表スペースをリストアする場合の操作

– REBUILD WITH ALL TABLESPACES IN DATABASEを指定してRESTOREを実行



– ROLLFORWARD のオプションにはPIT/end of logs以外に、End of Backupでバックアップ取得完了時点までのログのみ適応させることや、COMPLETEではログ適応を行わずROLLFORWARD処理を完了扱いさせることも可能。

# Thank you

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法律的またはその他の指導や助言を意図したものではありません。またIBM製品やサービスがお客様に適用ある特定の法令に適合することを保証するものでもありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努めておりますが、「現状のまま」提供され、明示または黙示にかかわらず、商業性、特定の目的への適合性、非侵害性を含め、いかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本講演資料で言及されるIBM製品、プログラム、またはサービスは、IBMがビジネスを行っているすべての国・地域でご提供可能なわけではありません。

本講演資料で言及される将来の展望(製品リリース日付や製品機能を含む)は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、将来の製品または機能が使用可能になること、もしくは特定の結果を確約することを意図するものではありません。本講演資料は、言及されるIBM製品またはサービスに適用ある契約条件を変更するものでも、追加の表明または保証を意図するものでもありません。

本講演資料に含まれている内容は、参加者の活動によって特定の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴは、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれIBM または各社の商標である場合があります。現時点での IBM の商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) をご覧ください。

