

バージョン 6.0.1



データベース・サービス

お願い

本書および本書で紹介する製品をご使用になる前に、85 ページの『特記事項および商標』に記載されている情報をお読みください。

本書は、IBM WebSphere Business Monitor 製品のバージョン 6.0.1 (5724-M24)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

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

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

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

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： WebSphere® Business Monitor
Database services
Version 6.0.1

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.4

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

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

© Copyright International Business Machines Corporation 2005, 2006. All rights reserved.

© Copyright IBM Japan 2006

目次

データベースの管理	1	ヒストリー・データベースのスキーマ	75
データベースの構成とレプリケーション	1	データ移動サービス制御テーブル	77
WebSphere Business Monitor データベース	3	データ移動サービス・メタデータおよびロギング・	
データベースの作成とデプロイメント	6	テーブル	79
データベース・レプリケーション	15	データベース・サービスのトラブルシュー	
変更管理および成果物の生成	39	ティング	81
データベース保守	42	デプロイメントの問題	81
データベースの作成と構成	46	実行時の問題	83
実行時のデータベースの管理	48	ランタイム・データベースの停止	84
ヒストリー・データベースのスキーマ	71		
データベース・サービス	73	特記事項および商標	85

データベースの管理

WebSphere® Business Monitor のデータベースの管理は、WebSphere Business Monitor のインストールおよびデプロイメントの段階において特に重要です。

データベースの管理には以下のタスクが含まれます。

- データベースの作成
- 適切なデータベース構成の設定
- 静的および動的な、データベース・テーブルと索引の作成
- 生成されたレプリケーション・スクリプトのデプロイ
- Cube View メタデータ定義のデプロイ
- データベースの保守

注: WebSphere Business Monitor の資料では、さまざまなデータベースの基礎となるスキーマについて説明されています。一部の Schema Generator の出力により、WebSphere Business Monitor のデータベース・スキーマの本質的な構成を把握できます。DBA では、データ・ストアの保守および調整のためにこの情報が利用される可能性があります。この情報を共通 API の定義として認識すべきではありません。WebSphere Business Monitor の将来のバージョンでは、この基礎となるスキーマが変更されることがあり、そもそも変更は常に可能です。お客様は、将来のリリースにおいてこれらのスキーマに対する後方互換性が維持されることを前提としたインフラストラクチャーを開発するべきではありません。WebSphere Business Monitor のデータベースを使用するために作成されたカスタム・コードには、おそらく将来の製品リリースとの互換性がありません。

以下の情報は、WebSphere Business Monitor のデータベースの管理についての計画および準備に役立ちます。

データベースの構成とレプリケーション

データ管理は、WebSphere Business Monitor において重要な役割を果たします。

WebSphere Business Monitor のデータベース・アーキテクチャーは、以下の要件をサポートします。

- 適切な処理速度を維持するために、データ・ストアのランタイム処理をクライアント・アクセス・データ・ストアから分離する。
- クライアント・アクセス・データ・ストアの更新を実行し、かつクライアントのクエリーに迅速に応答することが可能。
- 分析およびマルチディメンション・レポート作成を目的とするヒストリー・データ・ストアへのアクセスを最適化する。

WebSphere Business Monitor データベースのデータの使用パターンは、使用コンポーネントによって異なります。データは、2 つの主要コンポーネント (イベント処理プログラムとクライアント・ダッシュボード) によって使用されます。このよう

な使用法の相違により、イベント処理のデータベースをダッシュボードのデータベースから分割することが不可欠になります。データは、ビジネス指標モデルに関連付けられた情報とイベントの処理に関する情報に、さらに分類できます。

ダッシュボードは、2 つのタイプのデータ (最近のデータと履歴・インスタンス・データ) を表示します。最近のインスタンスの数は、履歴・インスタンスの数に比較してかなり少数です。最近のインスタンスについて実行されるクエリーは、非常に高速である必要があります。大容量の履歴・インスタンスの影響を受けないようにする必要があります。2 つのタイプのデータは、2 つのデータベース (ランタイム・データベースおよび履歴・データベース) に分割されました。パフォーマンスを向上させるために、このアーキテクチャーでは以下のものを使用してすべての機能をサポートします。

- **ビジネス指標モデルの定義コンテナ**としての役割を持つデータベース。他のデータベースに関する情報も保管されます。
- **トランザクション・データベース**としての役割を持ち、イベント処理プログラムによって使用されるデータベース。
- **ほぼリアルタイムの分析データベース**として機能し、トランザクション・サーバーに影響を与えずに分析クエリーをサポートします。このデータベースは、ダッシュボードによって使用されます。
- **トランザクションの履歴**に対するマルチディメンション分析をサポートするデータベース。このデータベースは、ダッシュボードによって履歴・データを表示するために使用されます。

WebSphere Business Monitor のデータベースは、4 つの異なるデータベースに分割されています。

- **リポジトリ**: ビジネス指標モデルおよびイベント定義を保管します。これは、状態データベース、ランタイム・データベース、および履歴・データベースのスキーマ、名前、およびホスト名も保管します。
- **状態**: 実行中のプロセス・インスタンスの現在の状態、およびそれぞれのプロセス・インスタンスと関連付けられたビジネス指標値を保管します。このデータベースは、WebSphere Business Monitor サーバーによってイベント処理のために使用されます。
- **ランタイム**: 状態データベースおよびランタイム・データベースには、非常に多くの全く同じ情報が保管されます。ランタイム・データベースは、特定データの保管方法、データの新鮮さ、およびデータの残存期間のみが異なります。ランタイム・データベース内のデータは、状態データベース内のデータより少なくとも 24 時間長く残存します。ランタイム・データベースの目的は、ユーザーが、WebSphere Business Monitor サーバーによって処理されるイベントに影響を与えずにほぼリアルタイムの分析を実行できるようにすることです。ランタイム・データベースは、最新のインスタンスに対するクライアント・クエリーを提供します。このデータベースには、効率的なレポート作成のために、ビジネス指標グループのランタイム情報が保管されます。このデータベースはダッシュボードでの表示に使用されます。
- **履歴・データベース**: 履歴のマルチディメンション・レポート作成のために、完了したインスタンスの情報および実行中のインスタンスの現在の状態を、スター・スキーマ形式で保管します。このデータベースはダッシュボードでの表示に使用されます。

モニターされるイベントと Adaptive Action Manager のデータが 2 つのデータベースに保管されます。これらのデータベースは、WebSphere Business Monitor によって内部で使用されます。プロセス・インスタンスまたはメトリックに関する情報は保管されません。

- **エミッター:** エンジンから発行されたイベントを保管します。エミッター・データベース・テーブルは、エンジン・データベース内にあります。
- **アクション・カタログ:** Adaptive Action Manager が実行する必要があるシチュエーションおよびアクションとして定義されるイベントを保管します。これは、インストール時に作成されます。

WebSphere Business Monitor データベース

WebSphere Business Monitor は、4 つのデータベースを使用してイベント・データとビジネス指標モデルのメタデータを保管します。4 つのデータベースとは、リポジトリ・データベース、状態データベース、ランタイム・データベース、およびヒストリー・データベースです。

リポジトリ・データベース

リポジトリ・データベースには、現在デプロイされているビジネス指標モデルを記述しているメタデータ、およびその他の WebSphere Business Monitor データベースに関する情報が含まれます。リポジトリ・データベースには、デプロイされたモデルのヒストリーが含まれます。WebSphere Business Monitor のインストールあたり、リポジトリ・データベースは 1 つのみです。

リポジトリ・データベースは、ランチパッドによって使用されます。ランチパッドは、状態データベース、ランタイム・データベース、およびヒストリー・データベースのデータベース属性のデータをリポジトリ・データベースに取り込みます。これらの属性は、データベース・サーバーのデータベース名、データベース・スキーマ名、およびホスト名です。これらの属性は、その他の WebSphere Business Monitor コンポーネントによって、実行時に状態データベース、ランタイム・データベース、およびヒストリー・データベースにアクセスするために使用されます。リポジトリ・データベースのデータは、ビジネス指標モデルのインポート時にも取り込まれます。

リポジトリ・データベースは、以下のコンポーネントによって使用されます。

- **管理コンソール**

ビジネス指標モデルは WebSphere Business Monitor 管理コンソールを使用してインポートされます。このインポートされたモデルのプロセスおよびイベントの定義は、リポジトリ・データベースに保管されます。インポートが完了すると、そのビジネス指標モデルはデプロイされたものと見なされます。モデルがインポートされた後は、プロセスおよびイベントの定義は、その他の WebSphere Business Monitor コンポーネントにより取得可能です。

リポジトリ・データベースは、Schema Generator によっても使用されます。Schema Generator には、そのデータベース成果物の生成時に使用するスキーマ名情報が必要です。また、以前にデプロイされたビジネス指標モデルをユーザーが

変更し、これに対するスキーマを再生成しようとする、Schema Generator は、変更管理成果物を生成する前に、リポジトリ・データベースに成果物があるかどうかを確認します。

- **ダッシュボード**

ダッシュボードには、さまざまな視点からデータを表示するビューのセットがあります。これらのビューには、ランタイム・データベースのデータが取り込まれるものもあり、ヒストリー・データベースのデータが取り込まれるものもあります。ユーザーがこれらのビューの構成およびパラメーター化を実行できるように、ビューはリポジトリ・データベースからWebSphere Business Monitorのメタデータを取得する必要があります。DB2® Alphablox Cubes に対するクエリーを構成する必要があるビューもあります。これらのクエリーの構成には、リポジトリ・データベースから取得されるディメンション、指標、およびキューブ名に関するメタデータが必要です。また、ダッシュボードには、リポジトリ・データベースに保管されるビジネス・プロセスのプロセス・ダイアグラムも表示されます。

- **WebSphere Business Monitorサーバー**

WebSphere Business Monitor サーバーは、リポジトリ・データベースを使用してプロセスとイベントの定義を取得します。

状態データベース

状態データベースには、実行中のインスタンスに関する情報が保管されます。この情報には、メトリック、ビジネス指標、およびキー・パフォーマンス・インディケーター (KPI) の値が含まれます。このデータベースは、重いトランザクション・ワークロード用に最適化されています。WebSphere Business Monitor インストールあたり状態データベースは 1 つのみです。

それぞれのプロセス・インスタンスは、メトリック、ビジネス指標、および KPI を保管するために、状態データベース内に 2 つのテーブルを必要とします。これらのテーブルの構造は、プロセス・インスタンスの構造と同様に動的です。それぞれのビジネス指標は、2 つのテーブルのいずれかの別々の列によって表されます。ビジネス指標モデルの構築中に選択したオプションに応じて、状態データベースの情報のほとんど、またはすべてがランタイム・データベースに複製されます。

状態データベースは、WebSphere Business Monitor サーバーによって使用されます。実行時に、WebSphere Business Monitor サーバーは、処理されたイベントに応じて、状態データベースにあるプロセス・インスタンスの情報を、挿入、取得、および更新します。

状態データベースには以下の情報が含まれています。

- インポートされたビジネス指標モデルのデータの一部であるビジネス指標グループについての情報。
- WebSphere Business Monitor の実行中に作成される実行中のプロセス・インスタンス。
- 実行中のプロセスのイベント・エントリー。イベント・エントリーとは、特定のビジネス指標グループを更新するために受信されるイベント・データです。

ランタイム・データベース

ランタイム・データベースの構造は、状態データベースと類似しています。すべての実行中プロセスの現在の状況、および最近完了または失敗したプロセスの最終状況に関して、複製された情報を状態データベースから受信します。この情報は WebSphere Business Monitor ダッシュボードによって使用されます。また、ランタイム・データベースは、アラート通知を保管するために、Adaptive Action Manager によっても使用されます。WebSphere Business Monitor インストールあたりランタイム・データベースは 1 つのみです。

ランタイム・データベースは以下の情報を保管します。

- Adaptive Action Manager によってダッシュボードに送信されるアラート通知
- プロセス・データ
- メトリック値

ランタイム・データベースの情報は状態データベースから複製されます。

ランタイム・データベースは、WebSphere Business Monitor ダッシュボードによって使用されます。ダッシュボードは、ビューにデータを取り込むために必要な、実行中または最近完了したインスタンスのデータを、ランタイム・データベースから取得します。ダッシュボード・ビューは、ランタイム・データベースを分析の目的で使用するため、ランタイム・データベースはクエリー処理と集約クエリー処理に対して最適化されます。

ヒストリー・データベース

ヒストリー・データベースは、すべての完了済みプロセス・インスタンスと、一部の実行中プロセス・インスタンスを保管します。これは、ダッシュボードによって、DB2 Alphablox を使用した拡張データ分析に使用されます。WebSphere Business Monitor のインストールあたり、ヒストリー・データベースは 1 つのみです。ヒストリー・データベースのデータは、削除されることはありません。

ヒストリー・データベースには、2 年分のヒストリー・データのみが含まれます。これは、WebSphere Business Monitor 製品の要件の 1 つです。前述のとおり、ヒストリー・データは自動的に削除されることはないため、DBA は 2 年以上前のデータを削除する必要があります。ヒストリー・データベースは、長期間実行されるインスタンスおよび完了済みインスタンスに関する情報を保管します。この情報は、状態データベースおよびランタイム・データベースで使用するフラットなトランザクション形式ではなく、スター・スキーマ形式で保管されます。ヒストリー・データベースは、集約された、実行に時間がかかるクエリー用に最適化されています。このデータベースは、ダッシュボード・ビューで DB2 Alphablox によって、高度なマルチディメンション・レポートを提供するために使用されます。

ヒストリー・データベースの情報は、ランタイム・データベースから複製されます。

ヒストリー・データベースでは、それぞれのプロセス・インスタンスに独自のテーブルのセットがあります。状態データベースやランタイム・データベースとは異なり、それぞれのテーブルのセットは、マルチディメンション・レポートをサポートするスター・スキーマです。

ヒストリー・データベースには、デプロイされたビジネス指標モデルに応じて作成される動的テーブルが含まれます。Schema Generator は、動的テーブルの作成に使用されるヒストリー・データベースのスキーマ、および Cube Views™ 定義を生成します。

ヒストリー・データベースは、WebSphere Business Monitor ダッシュボードによって使用されます。ダッシュボードは、一部のビューを取り込むために必要なデータをヒストリー・データベースから取得します。例えば、レポート ビューは、ヒストリー・データベースから抽出されたデータの分析に重点を置いています。

ヒストリー・データベースには以下の情報が含まれます。

- 実行中または終了しているプロセス・インスタンスのさまざまなバージョンのデータ。
- ランタイム・データベースに保管されている完了済みプロセス・インスタンスのデータ。完了したすべてのインスタンスは、ランタイム・データベースに 24 時間残存します。24 時間はデフォルトの保存ポリシーで、これはデータ移動サービス構成の一部として変更することができます。このデータをヒストリー・データベースに複製した後で、パフォーマンスを向上させるために、この情報はランタイム・データベースから削除されます。

データベースの作成とデプロイメント

WebSphere Business Monitor のデータベースは、インストール中にランチパッドによって作成されます。データベースを作成する前に、まずその作成およびデプロイメントについて計画する必要があります。

インストール後に WebSphere Business Monitor のデータベースが除去されるか、または著しく破損した場合、データベース管理者 (DBA) は作成スクリプトを実行することにより手動でデータベースを再作成できます。それらは

`<monitor_installation_dir>%install%mondb%` に保管されます。DBA は、ランチパッドを使用してデータベースをアンインストールすることもできます。データベースを作成するには、DBA はまず DB2 から手動でデータベースを除去し、その後ランチパッドを使用してデータベースを再作成します。

データベース成果物のデプロイメントの準備

ランチパッドを使用して WebSphere Business Monitor のデータベースの作成を開始する前に、データベースについて計画する必要があります。計画には、データベース・インスタンスおよび個々のデータベースの設定の決定に加え、データベース・サイズの割り振り、バックアップ計画の準備、データ移動サービスの構成、およびテーブル・スペースとバッファーク・プールのパラメーターの設定が含まれます。

インストール時に、ランチパッドは、状態データベース、ランタイム・データベース、およびヒストリー・データベースと、管理の目的で使用するデータベース・オブジェクトを作成します。これらのオブジェクトに加えて、Schema Generator が、データベース・オブジェクト (テーブルなど) 専用のビジネス指標モデルのセットを作成します。ランチパッドは、状態データベース、ランタイム・データベース、およびヒストリー・データベースのデフォルトのテーブル・スペースおよびバッファーク・プールのセットを作成します。これらのデフォルトのテーブル・スペースは、テーブル・スペース構成ファイルで参照され、ユーザーが概念シナリオを迅速にテ

ストおよび検査することができるように設計されています。パフォーマンス上の問題およびリソースの制約を回避するには、テーブルをテーブル・スペース全体にどのように配置するかを事前に計画することが重要です。そのコンテナおよびバッファ・プールは、テーブル・スペースによって使用されます。

インストール時に、データベースが作成され、静的テーブル定義のみが作成されません。成果物の生成時に、カスタマイズ可能なテキスト構成ファイルに基づいて、状態データベース、ランタイム・データベース、およびヒストリー・データベースのテーブル・スペースに動的テーブルが割り当てられます。WebSphere Business Monitor は、<Monitor_install_dir>%install%mondb ディレクトリーにあるデフォルト構成ファイルと同梱されています。このデフォルト構成ファイルは、すべてのテーブルを適切なサイズの 1 つのテーブル・スペースにマップします。随時デプロイメントをサポートするために、ランチパッドは、インストール時に、デフォルト構成ファイルのエントリーに対応するテーブル・スペース (4 KB、8 KB、16 KB、および 32 KB ページ・サイズ) のセットを作成します。以下の例に示すのは、WebSphere Business Monitor と同梱されるデフォルトのテーブル・スペース構成ファイルからの抜粋です。

```
#  
  
# State database  
  
#  
  
db2.state.Default.TABLE.4K.0=DSDFLTTS4  
  
db2.state.Default.TABLE.8K.0=DSDFLTTS8  
  
db2.state.Default.TABLE.16K.0=DSDFLTTS16  
  
db2.state.Default.TABLE.32K.0=DSDFLTTS32
```

この簡単な構成は成果物の生成時に使用され、4 KB ページ・サイズのテーブル・スペースに適合する、状態データベースで作成される必要のあるすべてのテーブルが DSDFLTTS4 に割り当てられます。8 KB のページ・サイズのテーブルは DSDFLTTS8 という名前のテーブル・スペースに、16 KB のページ・サイズのテーブルは DSDFLTTS16 という名前のテーブル・スペースに保管されます。テーブル・スペース要件は異なる可能性がある (データが保管されるモデルの複雑さやデータの量によって異なる) ため、テスト環境や実稼働環境ではデフォルトのテーブル・スペース構成の設定を使用しないことをお勧めします。事前の計画および適切なストレージ計画の決定により、良好なパフォーマンスが得られます。

拡張構成ファイルの設定を使用して、ページ・サイズのみでなく、保管されるデータの種類にも基づいて、テーブル・スペースにテーブルをマップすることができます。詳しくは、サンプルの構成ファイルを確認してください。

テーブルを割り当てるテーブル・スペースを判別するために、Schema Generator は以下のタスクを実行します。

- テーブル・タイプを判別する。
- 少なくとも 1 行のデータを保管するために必要な最小ページ・サイズを計算する。

- このタイプ用に使用可能なテーブル・スペースを確認する。見つかった場合は、次の使用可能なテーブル・スペースを見つけて使用します。テーブル・スペースが見つからない場合は、継続します。
- Schema Generator のデフォルトのテーブル・スペースを確認する。見つかった場合は、次の使用可能なテーブル・スペースを見つけて使用します。このタイプ用に使用可能なテーブル・スペースが見つからない場合は、継続します。
- テーブルをデータベースのデフォルト・テーブル・スペースに割り当てる (テーブル作成時に table space 文節は指定しない)。

注: データベースに必要なページ・サイズのデフォルトのテーブル・スペースが定義されていない場合は、デプロイメントは失敗します。

テキスト・エディターを使用してテーブル・スペース構成ファイルを編集するか、新規のファイルを作成できます。Schema Generator 管理コンソールの「一般構成」タブを使用して、Schema Generator が代替構成ファイルを使用するようにしてください。

注: Schema Generator は、実際にはこの構成ファイル内のエントリーに対するテーブル・スペースを作成しません。これは、生成されたデータベース成果物のデプロイ前に手動で実行する必要があります。成果物のデプロイメントは、テーブルが存在しないテーブル・スペースに割り当てられていた場合は失敗します。

データベース成果物のデプロイメント

ランチパッドで WebSphere Business Monitor のデータベースを作成した後、データベース・テーブルをデプロイします。デプロイメント・フェーズでは、データベースのセットアップを完了させるためにデプロイされる成果物を生成するように、Schema Generator が構成されます。これにより、データベースにデータを取り込む準備ができます。

注: データベースが作成されるのは 1 回です。そして、各ビジネス指標モデルについて、より多くのデータベース・テーブルがデータベースに追加されます。

データベース・スキーマの生成:

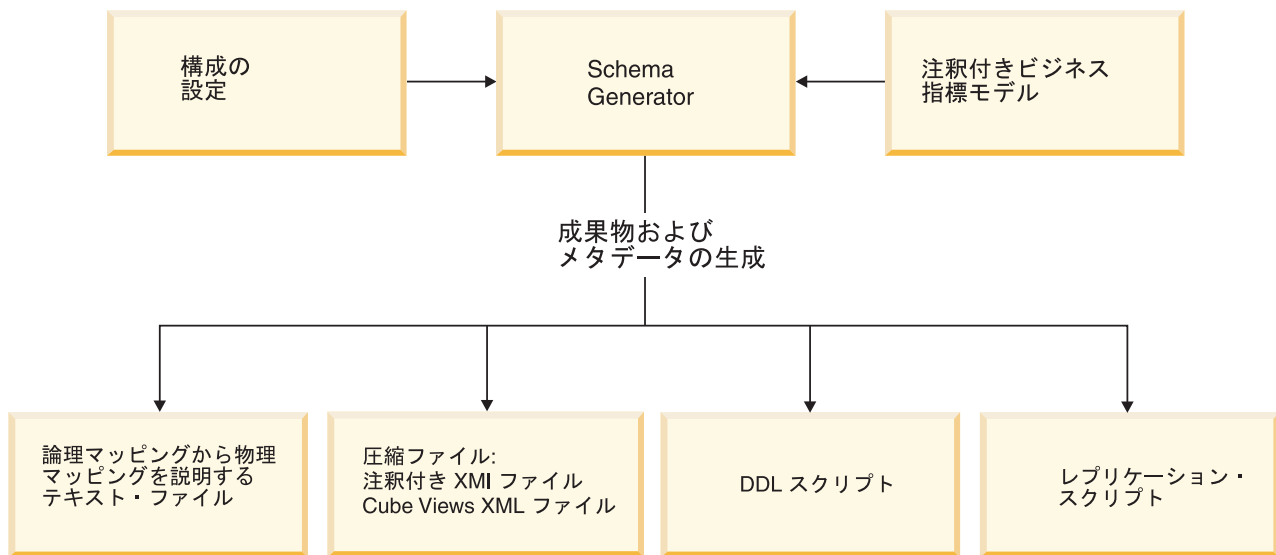
WebSphere Business Monitor のデータベースのスキーマは、ビジネス指標モデルに基づいています。これらのスキーマは Schema Generator によって生成されます。

ビジネス指標モデルは、ビジネス指標エディター を使用して作成されます。これは注釈付きのモデルで、ビジネス・モデルのメタデータが含まれます。ユーザーは、ビジネス指標エディター を使用して、モニター対象 (コンテキスト、キー・パフォーマンス・インディケーター (KPI)、メトリック、およびビジネス・シチュエーション) を定義することができます。ビジネス指標モデルは、イベントを介してモニター情報を作成します。ビジネス指標モデルは、完成した後に zip 形式の注釈付き XML ファイルとして WebSphere Business Monitor にエクスポートされ、Schema Generator がこれを使用します。

Schema Generator は WebSphere Business Monitor 管理コンソールの一部です。通常、このツールは、データベース管理者 (DBA) が構成し、使用します。入力データとしてビジネス指標モデルを取り、データベース成果物を生成します。

生成される成果物は次のとおりです。

- データベース・テーブルの作成に適したデータベース定義言語 (DDL) スクリプト。生成されたそれぞれの DDL ファイル (state.ddl、runtime.ddl、datamart.ddl) に対して、対応するテキスト・ファイル (stateMapping.txt、runtimeMapping.txt、datamartMapping.txt) が作成されます。これらのファイルには、物理データベース成果物 (テーブル、列) が表すメトリックとプロセスの説明が含まれています。
- スター・スキーマ形式でヒストリー・データベースのデータを説明する DB2 Cube Views 定義。Cube Views の定義は、DB2 OLAP Center へのインポートに適しています。
- 状態データベース、ランタイム・データベース、およびヒストリー・データベース間のレプリケーションを使用可能にするレプリケーション・スクリプト。Schema Generator は、これらの各データベースに対して、3 つのデータベースにわたってレプリケーションをセットアップするために必要なすべてのデプロイメント成果物を含む圧縮ファイルを作成します。通常、これらの成果物は、DBA によって、51 ページの『データ移動サービスのデプロイ』の指示のとおり配布およびデプロイされます。



WebSphere Business Monitor のデータベース・コンポーネントのデータベース・テーブルには 2 つのタイプがあります。

- インストール時に 1 回のみ作成される静的データベース・テーブルがあります。これらのテーブルは、すべてのビジネス指標モデルで共用され、単一のビジネス指標モデルに従属しません。
- WebSphere Business Monitor 管理コンソールにインポートされるビジネス指標モデルに従属する動的データベース・テーブルがあります。動的データベース・テーブル・スキーマは、各ビジネス指標モデルに対して固有です。動的テーブルに関連付けられたビジネス指標モデルに対するすべての変更は、変更管理のシナリオになります。変更管理のシナリオについて詳しくは、39 ページの『変更管理および成果物の生成』を参照してください。

Cube Views の定義:

Schema Generator は、DB2 Cube Views XML ファイルを作成します。データベース管理者 (DBA) は、この XML ファイルを DB2 OLAP Center にインポートします。

Schema Generator は、ビジネス指標モデルに基づいて Cube Views の XML ファイルを生成します。ビジネス指標モデルには、指標およびディメンションの説明に役立つ情報が含まれます。これにより、指標に適用可能な集約を説明することもできます。

ビジネス指標モデルのそれぞれのプロセスに対して、キューブとキューブ・モデルが作成されます。同様に、プロセスと関連付けられたアクティビティーに対してもキューブとキューブ・モデルが生成されます。それぞれのキューブ・モデルとキューブには、自動的に生成される一部の事前定義の指標とディメンションが含まれます。

それぞれのキューブ・モデルおよびキューブには、3 つの組み込まれた指標が含まれます。

- **経過時間:** 定義済み集約関数「**avg**」があります。
- **作業時間:** 定義済み集約関数「**avg**」があります。
- **インスタンス・カウント:** 定義済み集約関数「**count**」があります。

以下のディメンションが自動生成されます。

- **作成時刻:** プロセス・インスタンスが作成された時刻
- **開始時刻:** プロセス・インスタンスが開始された時刻
- **State:** 「started」、「running」、「completed」などの、考えられるプロセス・インスタンスの状態 (ストリング値) をすべて含みます。
- **終了時刻:** プロセス・インスタンスが終了した時刻

これらの時間ベースのディメンションは、提供されている共通の定義済みディメンションを使用します (DIM_TIME)。これには、年、月、および日の 3 つの定義済みレベルがあります。

モデル化する時に、ユーザー独自のビジネス指標を定義できます。作成されるビジネス指標は、指標またはディメンションです。WebSphere Business Modeler の資料には、WebSphere Business Modeler を使用して指標またはディメンションを作成する方法についての詳細な説明があります。

DB2 Cube Views についての詳細は、DB2 の資料を参照してください。

成果物の生成とデプロイメント:

Schema Generator は、インポートされた各ビジネス指標モデルに基づいて、データベース成果物および Cube Views 成果物を生成します。

成果物は、ビジネス指標モデルのデプロイメント・フェーズの手順として、データベース管理者 (DBA) によって実行される必要があります。任意のビジネス指標モデルをデプロイする前に、以下のタスクを完了する必要があります。

- 事前に準備された演習を実施しますこれには、トポロジーの決定のほか、イベント比率、サポートされるユーザー数、データベース・インスタンス、データベース・パラメーター、バッファー・プールおよびテーブル・スペース、バックアップ

プおよびリカバリー計画の決定が含まれます。また、生成された成果物の格納方法および格納場所を決定するための計画を行うことも重要です。モデルおよびバージョン番号によって以前に生成された成果物のセットを検索して、それらがデプロイ済みかどうかを確認できるようにすることをお勧めします。これは、変更管理およびサポートの両方のシナリオに役立ちます。

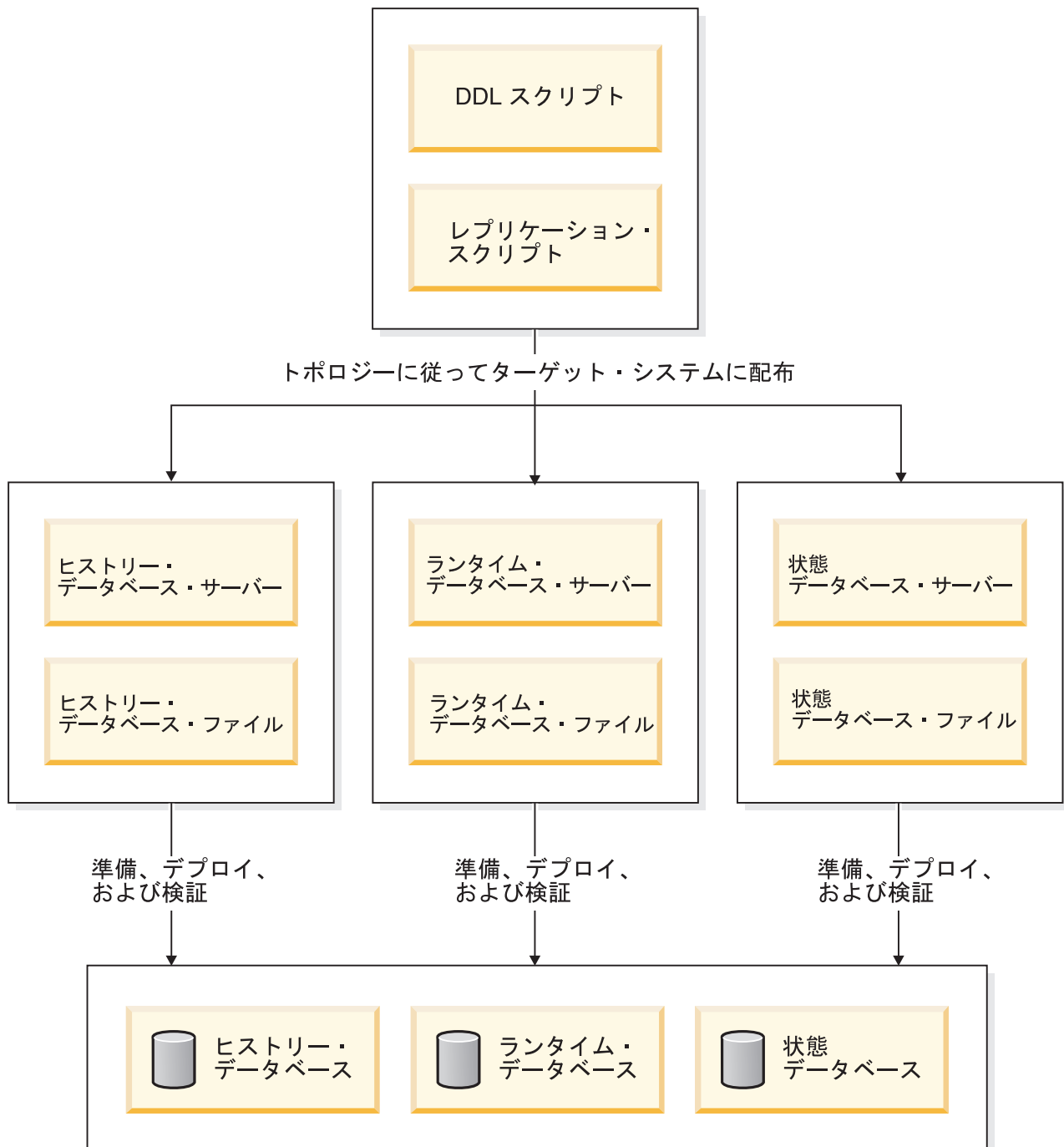
- **WebSphere Business Monitor** ランチパッドを使用して、リポジトリ・データベース、状態データベース、ランタイム・データベース、およびヒストリー・データベースを作成します。データベースの作成には、静的データベース・テーブルのセットの作成、レプリケーション・ソースとする状態データベースおよびランタイム・データベースのセットアップ、および **WebSphere Business Monitor** の各種コンポーネント固有の、ストアード・プロシージャおよび ユーザー定義関数などのデータベース・オブジェクトの作成も含まれます。
- **Schema Generator** を実行して、ビジネス指標モデルに関連する成果物を生成します。

注: 生成された成果物をデプロイする前に、すべてのデータベースをバックアップすることを推奨します。

成果物を生成した後、以下の作業を実行してそれらをデプロイします。

- 生成された **DDL** スクリプトをデプロイして、状態データベース、ランタイム・データベース、およびヒストリー・データベースに、ビジネス指標モデルの動的テーブルを作成します。
- レプリケーション・スクリプトを実行して、データベース・レプリケーションを使用可能にします。
- **Cube View** の定義ファイルを **DB2 Cube Views** にインポートします。

次の図は、成果物のデプロイメント・フェーズを説明しています。



成果物のカスタマイズ:

特定の環境下では、生成されたデータベース成果物の変更がパフォーマンスの改善に役立つ場合があります。一般に、反復的改善および随時改善の 2 つの方法を使用できます。

反復的テーブル・スペース・マッピングの改善

Schema Generator によって生成されるテーブルの数は、ほかの要因よりもビジネス指標モデルの複雑さに最も依存します。このため、これらのテーブルをテーブル・スペースに最適に割り当てる方法を最初に判断することは困難です。以下の簡単な

方法により、テーブル・スペース構成ファイルに定義されるとおりに、テーブルからテーブル・スペースへのマッピングを徐々に改善していくことができます。データベース計画について詳しくは、6 ページの『データベース成果物のデプロイメントの準備』を参照してください。テーブルとテーブル・スペースのマッピングを改善するには、以下のタスクを実行する必要があります。

1. デフォルトまたは他のカスタマイズ済みテーブル・スペース構成ファイルを使用して、Schema Generator を実行します。
2. 生成済み成果物を一時ディレクトリーに抽出し、データベースに作成されるテーブルの数およびタイプを確認します。各ビジネス指標モデルについて、以下のタイプのテーブルが作成されます。
 - **コンテキストとアクティビティ**: 状態データベースとランタイム・データベースのみに存在し、サーバーおよびダッシュボードによってアクセスされます。
 - **CD**: 状態データベースとランタイム・データベースにのみ存在します。CD テーブルには通常、関連コンテキスト・テーブルまたはアクティビティ・テーブルより少し多くの列が含まれます。特定の CD テーブルのサイズは、対応するコンテキスト・テーブルまたはアクティビティ・テーブルに対するトランザクションの数、更新トランザクションの挿入トランザクションに対する比率、関連するデータ移動サービスの Apply コンポーネントによる CD テーブルからのトランザクションの読み取り頻度、および対応する CCD テーブルへのそれらの挿入頻度、および 関連する Source Life Cycle コンポーネントによって CD テーブルのエントリーが整理される頻度に大きく依存します。
 - **CCD**: ランタイム・データベースとヒストリー・データベースにのみ存在します。CCD テーブルは、対応する CD テーブルとまったく同じ構造を持ち、そのサイズも同じ要因に大きく依存します。1 つの相違点として、トランザクションは、Apply コンポーネントではなく ETL コンポーネントによって読み取られ、エントリーは Target Life Cycle コンポーネントによって整理されます。
 - **RM 内部テーブル**: ランタイム・データベースとヒストリー・データベースにのみ存在します。これらのテーブルは最大サイズが 4 KB のページを使用します。
 - **ファクトとディメンション**: ヒストリー・データベースにのみ存在します。
3. 以下のものが含まれるようにテーブル・スペース構成ファイルを変更します。
 - 各テーブル・タイプのマッピング
 - マッピングがないと同一テーブル・スペースに多数のテーブルが割り当てられる場合は、複数のテーブル・タイプからテーブル・スペースへのマッピング

注:

- データベースには作成されないテーブル・タイプに対するテーブル・スペース宣言を指定しないでください。それらのテーブル・スペースは使用されません。
 - メトリックはテーブル列にマップされます。より多くのメトリックが定義されるに従って、テーブルは大きくなり、そのテーブル・スペースに必要なページ・サイズが大きくなります。
4. 生成済み成果物をデプロイする前に、これらのテーブル・スペース (およびバッファー・プール) を作成します。データベース接続が確立されていないため、

Schema Generator は、構成ファイルで宣言されたテーブル・スペースが存在するかどうかを検証しません。ただし、テーブル・スペースが存在しない場合は、デプロイメントは失敗します。

5. 最適化されたテーブル・スペース構成ファイルを使用して、Schema Generator を再実行します。

注: 新規テーブル・スペース構成ファイルを作成した場合は、Schema Generator 管理コンソール構成で構成ファイル名を変更します。

随時改善

以下を変更することにより、生成済み成果物を変更できます。

- 任意の索引: 追加、変更、除去、または別のテーブル・スペースへの割り当て (DMS のみ) を実行して、データベースのパフォーマンスを向上させます。

注: 固有索引の作成については、予期しない障害を発生させる可能性があるため、慎重に検討する必要があります。

- 任意のテーブル・スペース割り当て (データベースによって管理されたスペース (DMS) のテーブル・スペースを使用する場合は、異なるテーブル・スペースへの割り当て、および索引またはラージ・オブジェクトに対するテーブル・スペース割り当ての追加を行います)。
- テーブルについての任意のコメント (コメントによって各テーブルおよび列が何を表すかが識別されるため推奨しません)。

一般に、許可される変更は、基本的なテーブルのスキーマまたは構造を変更しないものに限られます。

生成されたスクリプトを変更する場合は、以下の制限事項を考慮する必要があります。

- テーブル名は変更できません。
- 列名は変更できません。
- 列データ・タイプは変更できません。
- 列またはテーブルは除去できません。
- テーブルに列は追加できません。
- テーブルの基本キーは変更できません。
- 列の NULL 可能性は変更できません。
- テーブルのスキーマ割り当ては変更できません。
- 固有性制約または外部キー制約などの新規制約は追加できません。

重要: 生成済みデータベース成果物に対する変更は、同じビジネス指標モデルについて実行される次のスキーマ生成では考慮されません。例えば、ユーザーが、ビジネス指標モデルの「Finance Model」の生成済み成果物をデプロイする前に変更します。続いて、ビジネス指標モデルを変更し、すべての成果物を再生成します。この場合、Schema Generator では事前に生成された成果物に対する変更が認識されていないため、ユーザーは新たに生成された成果物を再度変更する必要があります。

注:

- 成果物のデプロイメントには、Java™ バージョン 1.4.2 以降が必要です。
- レプリケーション・スクリプトを実行する前に、Java bin ディレクトリをシステム・パスに追加する必要があります。

データベース・レプリケーション

データベース・レプリケーション・テクノロジーは、ビジネス指標モデル関連データを、状態データベースからランタイム・データベースに、およびランタイム・データベースからヒストリー・データベースに移動するために使用されています。

状態データベース、ランタイム・データベース、およびヒストリー・データベースの間でのビジネス指標モデルのレプリケーションをセットアップするには、以下の高度なタスクを実行する必要があります。

1. **レプリケーション・セットアップ・スクリプトを生成します。** Schema Generator は、レプリケーションをセットアップする必要があるビジネス指標モデルを分析し、セットアップ・ファイルのセットを生成します。
2. **レプリケーション・セットアップ・スクリプトを配布します。** これらのセットアップ・ファイルは、状態データベース、ランタイム・データベース、およびヒストリー・データベースをホストするマシンに手動で転送する必要があります。
3. **レプリケーション・セットアップ・スクリプトを実行します。** 各セットアップ・ファイルは、実行時に、必要なデータベース・オブジェクトを作成し、データベース間でデータを移動するユーティリティーを構成します。

正常終了すると、レプリケーション・ユーティリティーは開始および操作可能となります。以下のセクションでは、その構造についての詳細、および使用される基本的概念について説明します。

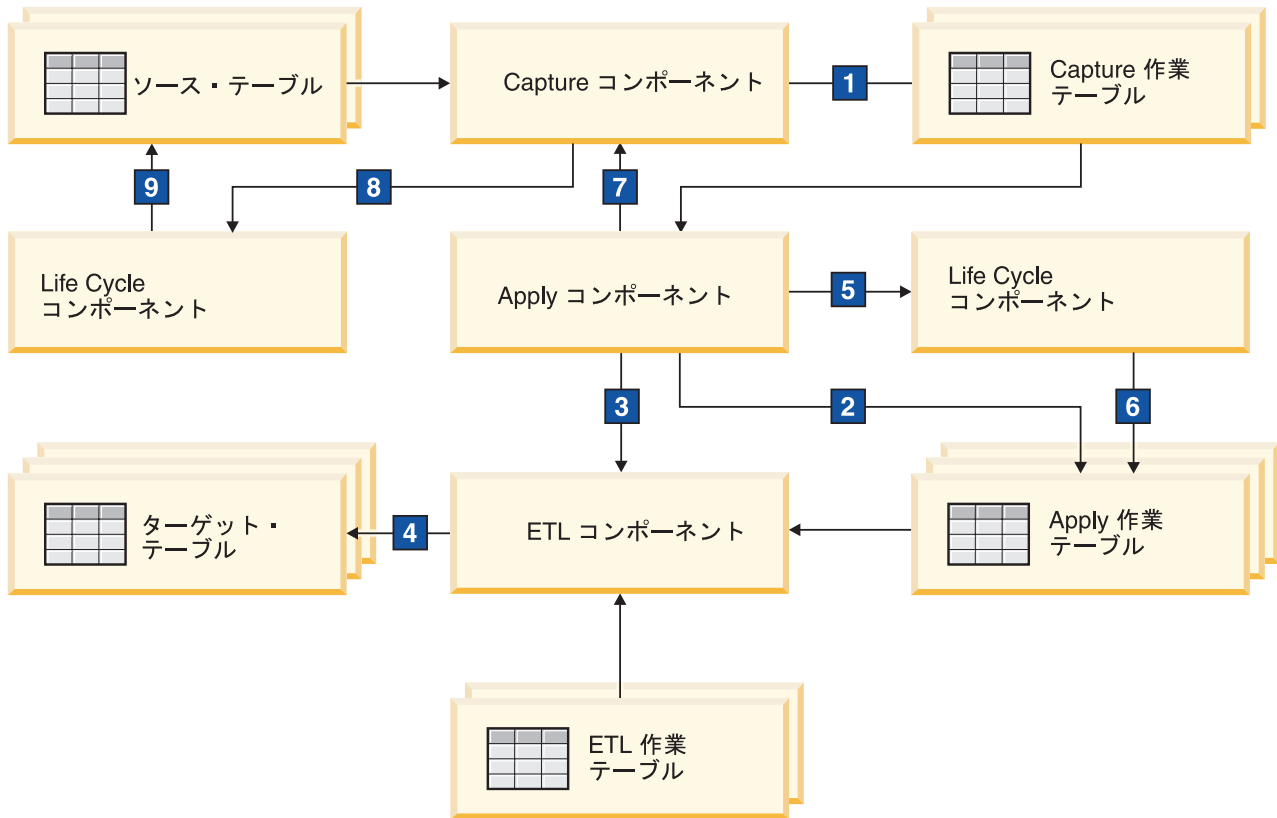
データ移動サービス

データ移動サービスにより、アプリケーションはソース・データベースからターゲット・データベースにデータを移動できます。ソース・データベースとターゲット・データベースは、同種でも異種でも構いません。つまり、単一システム上であっても複数システム間に分散していても構いません。データの移動のほかに、サービスはデータを変換し、アプリケーションが必要とする基本的なデータ・ライフ・サイクル機能を提供することができます。

データ移動サービスは 5 つの主要コンポーネントでインプリメントされています。

1. (ソース) Capture コンポーネント
2. (ターゲット) Apply コンポーネント
3. ETL (extract、transform、load) コンポーネント
4. Source Life Cycle コンポーネント
5. Target Life Cycle コンポーネント

Capture コンポーネントと Apply コンポーネントは連携して、ソース・データベースからターゲット・データベースにデータを移動します。ETL コンポーネントは、ソース・データベースのデータ構造がターゲット・データベースのデータ構造と異なる場合に必要なデータ変換を実行します。以下の図は、データ移動サービス内のプロセス・フローを説明しています。



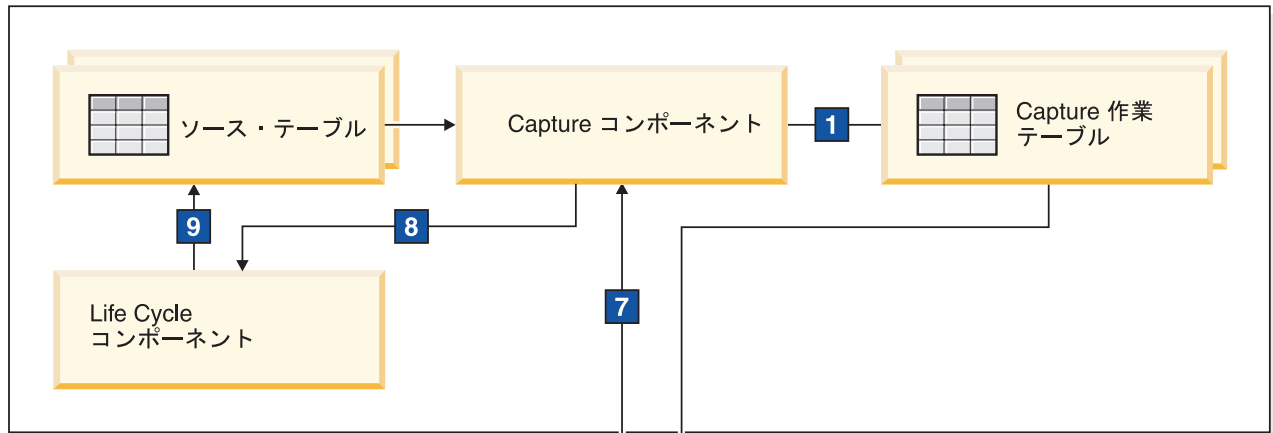
データ移動サービス・フローは、以下の手順に従います。

1. ソース・テーブルのデータは、例えばモニター・サーバーによって、保管され、頻繁に更新されます。Capture コンポーネントは、ソース・テーブルに対して実行されたデータの変更を作業テーブルに記録します。
2. 事前定義した間隔で、この変更は Apply コンポーネントによって識別され、作業テーブルに記録されます。
3. 変更が正常に記録された後で、ETL コンポーネントが呼び出されます。
4. ETL コンポーネントは、Apply 作業テーブルに保管されているデータと事前定義規則を使用して、必要な変換を実行します。正常に変換されたデータはターゲット・テーブルに書き込まれます。不完全、または誤りのあるデータは、後で処理するために別の作業テーブルのセットに保存されます。
5. ETL の処理が完了すると、Target Life Cycle コンポーネントがアクティブになります。
6. 時間の経過とともに、Apply 作業テーブルに大量のデータが累積される可能性があります。これらのテーブルで ETL コンポーネントによって正常に処理されたデータはすべて、Target Life Cycle コンポーネントによって除去されます。
7. データが正常にターゲット・データベースにコピーされると、そのデータは不要になり、Capture 作業テーブルから除去できます。Capture コンポーネントは、作業テーブルを定期的に整理し、リソースの不測の事態の発生を低減させます。
8. Capture 作業テーブルからデータを除去すると、Source Life Cycle コンポーネントの呼び出しをトリガーします。

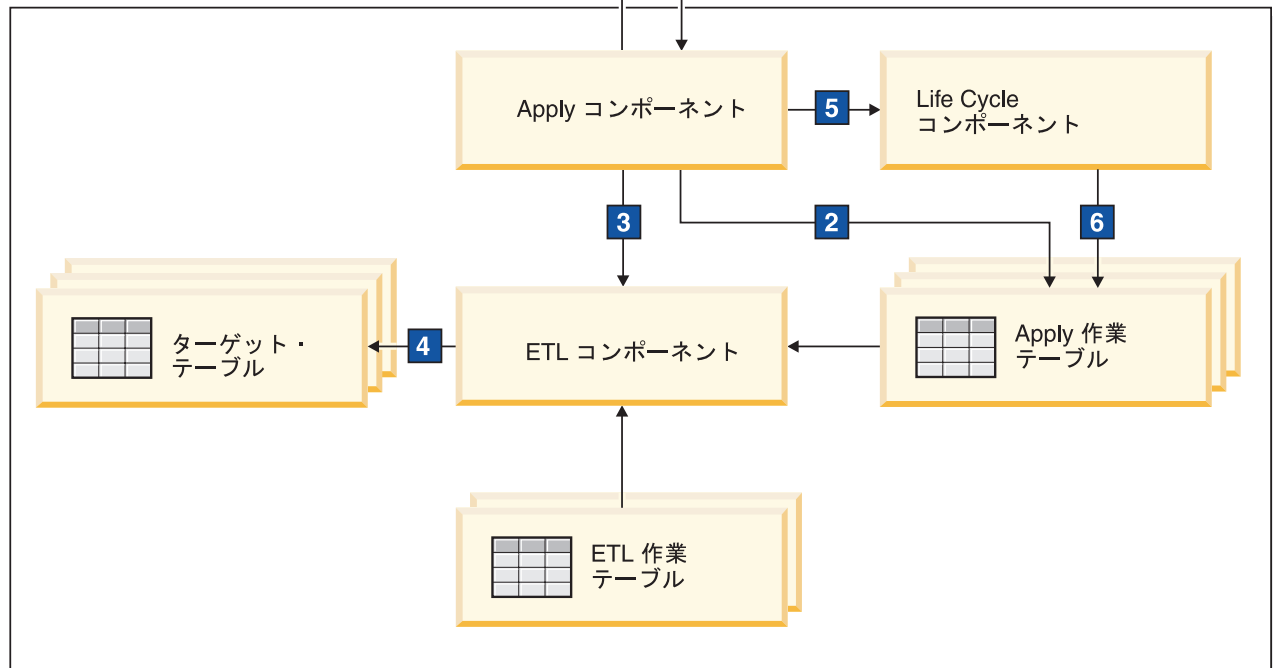
9. 正常に処理され、削除する準備ができていとマークされ、ソース・ライフ・サイクル保存ポリシーを通過したデータは、ソース・データベースから除去されます。

次の図のように、Capture コンポーネントおよび Source Life Cycle コンポーネントは、通常はソース・システム上にありますが、Apply コンポーネント、ETL コンポーネント、および Target Life Cycle コンポーネントは、ターゲット・システム上にあります。

ソース・データベース



ターゲット・データベース



データ移動サービス内では、ソース・データベースとターゲット・データベースで使用されているデータ構造に応じて、コンポーネントの複数のインスタンスが使用される場合があります。コンポーネント・インスタンスの数は、ビジネス指標グループの数、およびビジネス指標モデル内のソース・テーブルとターゲット・テーブルの数に直接関連しています。それぞれのインスタンスは一意的に識別されます。以下の規則が WebSphere Business Monitor に適用されます。

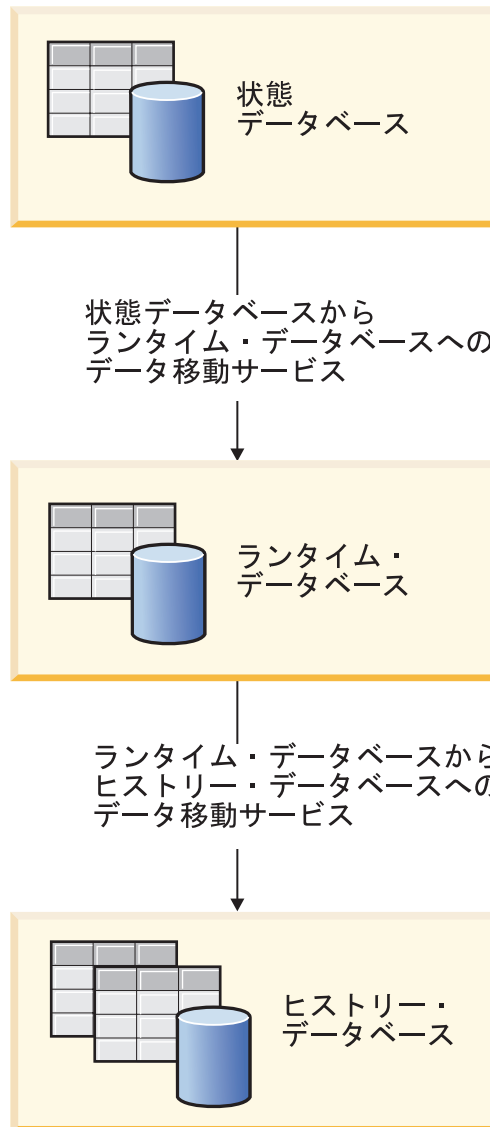
- 1 つのビジネス指標モデル・プロジェクトに 1 つの **Capture** コンポーネント・インスタンスが割り当てられ、このビジネス指標モデル・プロジェクトに属するすべてのソース・テーブルの変更をキャプチャーする。
- 1 つのビジネス指標モデル・プロジェクトに 1 つの **Apply** コンポーネント・インスタンスが割り当てられ、このビジネス指標モデルに属するターゲット・テーブルに適用する必要がある変更を記録する。
- 1 つのターゲット・テーブルに 1 つの **ETL** コンポーネント・インスタンスが割り当てられる。
- 1 つのソース・テーブルに 1 つの **Source Life Cycle** コンポーネント・インスタンスが割り当てられる。
- 1 つの **Apply** 作業テーブルに 1 つの **Target Life Cycle** コンポーネント・インスタンスが割り当てられる。

コンポーネントのインスタンスとは、例えば、実行可能プログラム、データベースのストアド・プロシージャ、またはデータベースのトリガーなどです。

WebSphere Business Monitor では、データ移動サービスの 2 つのインスタンスが使用されています。

- 状態データベースからランタイム・データベースへのデータ移動サービス
- ランタイム・データベースからヒストリー・データベースへのデータ移動サービス

状態データベースからランタイム・データベースへのデータ移動サービスは、モニター・サーバーによって状態データベースに保管されたデータを処理し、そのデータをダッシュボードからアクセス可能なランタイム・データベースに移動します。ランタイム・データベースからヒストリー・データベースへのデータ移動サービスは、ランタイム・データベースからヒストリー・データベースにデータを移動します。以下の図は、この移動を説明しています。



以下の情報は、これらのサービスのデフォルト構成、構成方法、開始/停止方法、およびモニター方法について説明しています。

状態データベースからランタイム・データベースへのデータ移動サービス:

状態データベースからランタイム・データベースへのデータ移動サービスは、モニター・サーバーによって状態データベースに保管されたデータを処理し、そのデータを、その他の WebSphere Business Monitor のコンポーネントおよびランタイム・データベースからヒストリー・データベースへのデータ移動サービスによってアクセス可能なランタイム・データベースに移動します。

このデータ移動サービスには次のデフォルト構成が適用されます。

- モニター・サーバーのソース・テーブル (状態データベース) での変更は、継続的にキャプチャーされ、作業テーブルに記録されます。

- それらの作業テーブルに記録された変更は、Apply コンポーネントによって継続的に伝搬され、ランタイム・データベースの作業テーブルに適用されます。それらの作業テーブルは WebSphere Business Monitor のその他のコンポーネントからはアクセスできず、内部使用専用です。
- Apply コンポーネントは、新規データの処理が必要になるたびに ETL コンポーネントを同期的に呼び出します。スケジュール (初めは 5 分ごとに設定される) に応じて、ETL コンポーネントは、Apply 作業テーブルに保管されているデータを処理するか、または実行するようにスケジュールされるまで非アクティブのままになります。スケジュールされた実行間の遅延を増加させると、モニター・サーバーによって状態データベースにデータが保管された時と、このデータがランタイム・データベースのターゲット・テーブルに公開される時間との間の経過時間が長くなります。ランタイム・データベースでは、データはその他の WebSphere Business Monitor のコンポーネントによってアクセスできます。
- Apply 作業テーブル内の ETL コンポーネントによって正常に処理されたデータは、スケジュールに従って Target Life Cycle コンポーネントによって除去されます。デフォルトでは、このコンポーネントは 24 時間ごとに 1 回実行されます。スケジュールされた遅延を増加させると、作業テーブルの容量が増大します。遅延を減少させた場合も、複数の Data Service コンポーネントが並行して作業テーブルの更新およびアクセスを試みる可能性があるため、偶発的な問題を引き起こす可能性があります。
- Capture 作業テーブルから Apply 作業テーブルに正常に移動されたデータは、Capture コンポーネントによってデフォルトで 5 分ごとに Capture 作業テーブルから自動的に除去されます。
- Capture 作業テーブルが整理されるたびに、Source Life Cycle コンポーネントが呼び出されます。このコンポーネントは、スケジュールにも基づいています。データが前回整理されてから少なくとも 5 分間経過していれば、モニター・サーバーによって削除する準備ができているとマークされているデータのみをソース・テーブルから除去します。Life Cycle コンポーネントの整理間隔が Capture コンポーネントの整理間隔よりも小さい値に設定されている場合は、Capture コンポーネントの整理間隔に基づいて整理が実行されます。

例: 作業テーブルに対する Capture コンポーネントの整理間隔は 5 分に設定され、Source Life Cycle コンポーネントのスケジュールは 1 分に設定されています。Capture プログラムが整理サイクルを開始できるまでに、5 分経過する必要があります。Capture の整理ルーチンは 5 分間アクティブにならないため、Life Cycle コンポーネントは呼び出されません。5 分経過後は、作業テーブルからデータが除去され、Source Life Cycle コンポーネントが呼び出されて、状態データベースのソース・テーブルからデータが除去されます。

デフォルトの構成は変更できます。

ランタイム・データベースからヒストリー・データベースへのデータ移動サービス:

ランタイム・データベースからヒストリー・データベースへのデータ移動サービスは、ランタイム・データベースからヒストリー・データベースにデータを移動します。データは、データベース管理者 (DBA) によって明示的に除去されるまでそこに

残存します。正常にヒストリー・データベースに移動されたデータは、WebSphere Business Monitor のその他のコンポーネントによって、取得および分析に使用できます。

このデータ移動サービスには次のデフォルト構成が適用されます。

- ランタイム・データベース・テーブルでの変更は、継続的にキャプチャーされ、作業テーブルに記録されます。モニターされているランタイム・データベースのテーブルは、状態データベースからランタイム・データベースへのデータ移動サービスによってデータが取り込まれたターゲット・テーブルです。
- それらの作業テーブルに記録された変更は、Apply コンポーネントによって継続的に伝搬され、ヒストリー・データベースの作業テーブルに適用されます。それらの作業テーブルは WebSphere Business Monitor のその他のコンポーネントからはアクセスできず、内部使用専用です。
- Apply コンポーネントは、新規データの処理が必要になるたびに ETL コンポーネントを同期的に呼び出します。スケジュール (初めは 24 時間ごとに設定される) に応じて、ETL コンポーネントは、Apply 作業テーブルに保管されているデータを処理するか、または実行するようにスケジュールされるまで非アクティブのままになります。スケジュールされた実行間の遅延を増加させると、状態データベースからランタイム・データベースへのデータ移動サービスによってランタイム・データベースにデータが保管されたときと、このデータがヒストリー・データベースのターゲット・テーブルに公開される時間との間の経過時間が長くなります。ヒストリー・データベースでは、データはその他の WebSphere Business Monitor のコンポーネントによってアクセスできます。

注: Apply コンポーネントによる呼び出しと、Apply コンポーネントの構成の依存関係により、ETL コンポーネントが 24 時間 (または現行のデフォルト遅延) ごとに新規データを処理できない可能性があります。このスケジュールは、「前回の処理サイクルの完了後、少なくとも 23 時間 59 分間は新規データを処理しない」と解釈する必要があります。

- Apply 作業テーブル内の ETL コンポーネントによって正常に処理されたデータは、スケジュールに従って Target Life Cycle コンポーネントによって除去されます。デフォルトでは、このコンポーネントは 24 時間ごとに実行されます。スケジュールされた遅延を増加させると、作業テーブルの容量が増大します。遅延を減少させた場合も、複数の Data Service コンポーネントが並行して作業テーブルの更新およびアクセスを試みる可能性があるため、偶発的な問題を引き起こす可能性があります。
- Capture 作業テーブルから Apply 作業テーブルに正常に移動されたデータは、Capture コンポーネントによって 5 分ごとに Capture 作業テーブルから自動的に除去されます。
- Capture 作業テーブルが整理されるたびに、Source Life Cycle コンポーネントが呼び出されます。このコンポーネントは、スケジュールにも基づいています。モニター・サーバーによって削除する準備ができているとマークされており、少なくとも 24 時間ランタイム・データベースに残存したデータのみを、ランタイム・データベースのソース・テーブルから除去します。デフォルト整理間隔は 5 分に設定されています。Life Cycle コンポーネントの整理間隔が Capture コンポーネントの整理間隔よりも小さい値に設定されている場合は、Capture コンポーネントの整理間隔に基づいて整理が実行されます。

例: 作業テーブルに対する Capture コンポーネントの整理間隔は 5 分に設定され、Source Life Cycle コンポーネントのスケジュールは 1 分に設定されています。Capture プログラムが整理サイクルを開始できるまでに、5 分経過する必要があります。Capture の整理ルーチンは 5 分間アクティブにならないため、Life Cycle コンポーネントは呼び出されません。5 分経過後は、作業テーブルからデータが除去され、Source Life Cycle コンポーネントが呼び出されて、ランタイム・データベースのソース・テーブルからデータが除去されます。

デフォルトの構成は変更できます。

データ移動サービスの管理

ビジネス指標モデルのデータ移動サービスを開始または停止するには、関連メインコンポーネント・インスタンスを識別し、それらを使用可能または使用不可にする必要があります。

すべての (ソース) Capture コンポーネントのインスタンスとすべての (ターゲット) Apply コンポーネントのインスタンスは、メインコンポーネントのインスタンスと見なされます。ETL コンポーネントのインスタンスと Target Life Cycle コンポーネントのインスタンスは、(ターゲット) Apply コンポーネントのインスタンスに従属しています。Source Life Cycle コンポーネントのインスタンスは (ソース) Capture コンポーネントに従属しており、明示的に開始または停止する必要はありません。一般的に、ソース・データベースのすべての (ソース) Capture コンポーネントのインスタンスと、すべての (ターゲット) Apply コンポーネントのインスタンスを、ユーザーが開始または停止する必要があります。これらのインスタンスがすべて開始または停止した後にはのみ、データ移動サービスが開始した (完全に作動可能になった) または停止したと見なされます。

Capture コンポーネントのインスタンスは DB2 Capture プログラムに相当し、Apply コンポーネントのインスタンスは DB2 Apply プログラムに相当します。データベース・システムがインストールされているオペレーティング・システムに応じて、どちらのプログラムも、スクリプトを使用して手動で、またはスケジューリング・ツールやサービスを使用して、開始および停止できます。デプロイメント中に、すぐに使用できる開始スクリプトと停止スクリプトが作成されます。Windows® システムの場合、これらのスクリプトはバッチ・ファイルです。UNIX® システムの場合、これらはシェル・スクリプトです。

それぞれの Capture プログラムは、ソース・データベース (状態データベースからランタイム・データベースへのデータ移動サービスの場合は状態データベース、またはランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合はランタイム・データベース) を含む DB2 をホストしているシステム上で開始される必要があります。Capture コンポーネントは、DB2 ログ・ファイルへのローカル・アクセス権を必要とします。

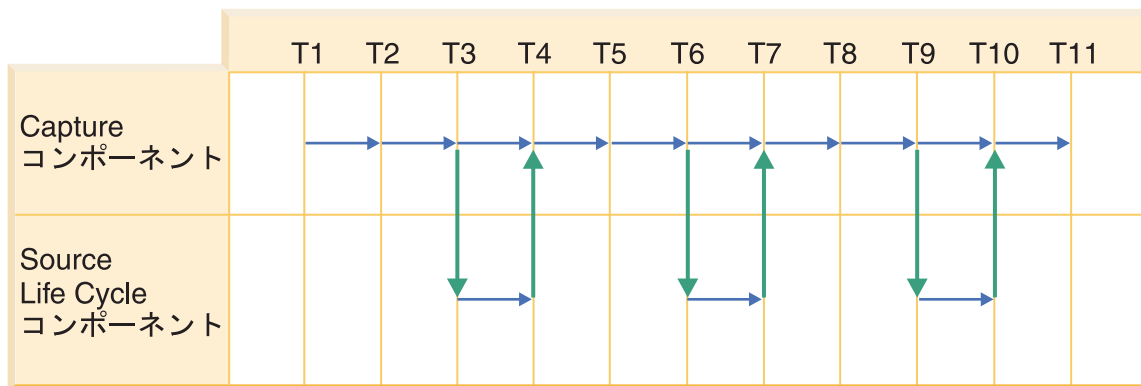
注: Capture インスタンスが実行中でない場合も、ソース・データベースのすべてのデータ変更がキャプチャーされています。ただし、すべてのインスタンスが作動可能になるまでは、キャプチャーされた変更は処理されず、ターゲット・データベースのテーブルには適用されません。

データ移動サービスの構成

各データ移動サービス・コンポーネントの動作とスケジューリングは、開発環境、テスト環境、および実稼働環境におけるさまざまな必要を満たすように構成できます。1 つのコンポーネントの構成を変更すると、そのコンポーネントに従属するその他のコンポーネントの動作に直接影響する可能性があります。

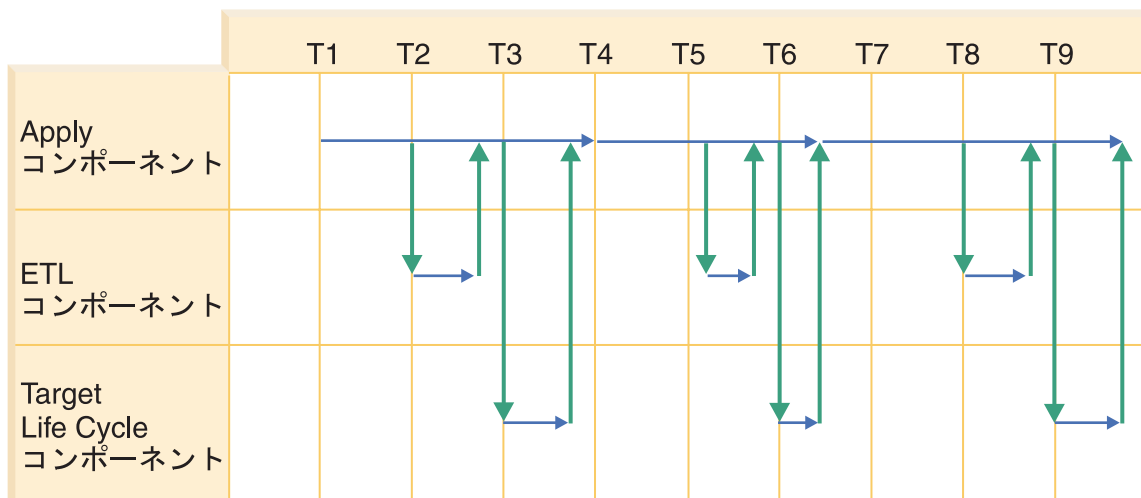
一般的に、2 つの依存関係があります。

- Capture コンポーネントは、定期的に Source Life Cycle コンポーネントを呼び出します。Capture コンポーネントが実行されていない場合は、Source Life Cycle コンポーネントは実行されません。Life Cycle コンポーネントの呼び出し間の遅延は構成可能です。

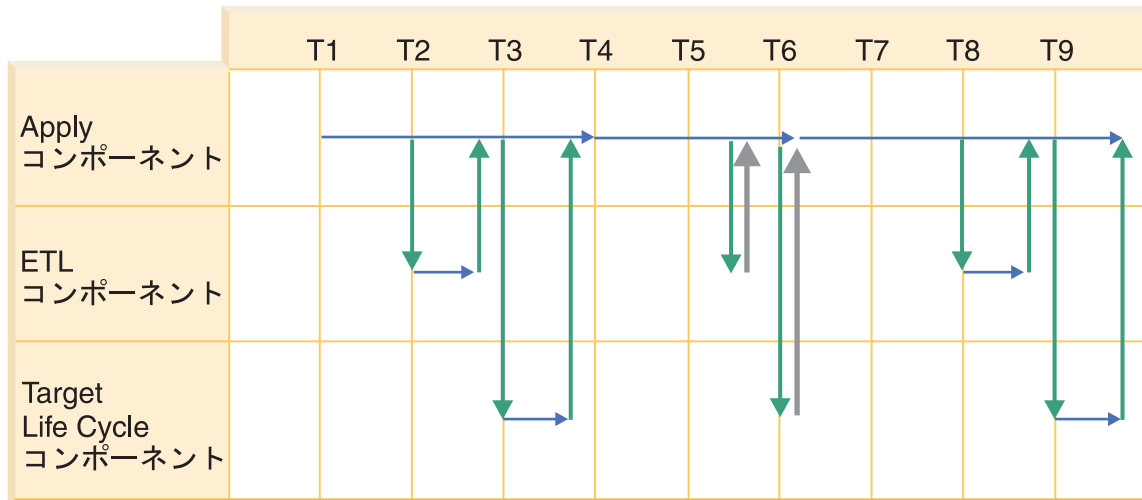


上の図では、Source Life Cycle コンポーネントは 3 つの時間単位ごとに呼び出され、いくつかのアクティビティを実行してから Capture コンポーネントに制御を戻し、処理が継続されます。

- ETL コンポーネントと Target Life Cycle コンポーネントは、データがソース・データベースからターゲット・データベースに正常に移動された後で、Apply コンポーネントによって呼び出されます。ETL コンポーネントと Target Life Cycle コンポーネントは、Apply コンポーネントが実行されている場合のみ呼び出されます。



従属コンポーネントは、それが従属するコンポーネントと異なるスケジュールで作動する必要があるため、呼び出しが必ずしも実行されるとは限りません。代わりに、それぞれの従属コンポーネントは、呼び出されたときにそのスケジュールを確認し、まだタスクを実行する時間でない場合は呼び出し側コンポーネントに制御を戻します。上記の例では、ETL コンポーネントと Target Life Cycle コンポーネントは、両方のコンポーネントのスケジュールにより 5 つの時間単位ごとに複数回の呼び出しが禁止されている場合のみ、2 回実行できます。



ETL コンポーネント (および Target Life Cycle コンポーネント) は T2 (および T3) で呼び出され、実行できます。次の呼び出しは、T6 あたりで発生します。それらが前回実行されてから 5 未満の時間単位が経過しているため、すぐに Apply コンポーネントに制御が戻されます。T8 (および T9) あたりで発生する後続の呼び出しは、5 つの時間単位が既に経過しているため、実行できます。各コンポーネントは 1 つ以上のコンポーネントのインスタンスによってインプリメントされます。各インスタンスを別々に構成して、より詳細な制御を実行できるようにすることができます。

注: 変更が行われると、特に注記されていない限り、それらは即時に反映されます。

Capture コンポーネントおよび Apply コンポーネントのデフォルト構成は、該当する制御テーブルの変更、または開始スクリプトでのコマンド行パラメーターを使用した指定変更により、変更可能です。いずれかの制御テーブルを更新することにより、ETL コンポーネントおよび Life Cycle 制約コンポーネントを構成できます。

以下の手順を実行して、開発環境、テスト環境、および実稼働環境の要件を満たすようにデータ移動サービス・コンポーネントをカスタマイズします。

(ソース) Capture コンポーネント・インスタンスの構成

Capture コンポーネント・インスタンスは、DB2 Capture レプリケーション・ユーティリティと同等です。デフォルトで、このユーティリティは継続的にソース・テーブルへの変更をキャプチャーし、それらの変更を内部の作業テーブルに記録するように構成されています。一般的に、Capture コンポーネント・インスタンスのデフォルト構成を変更する必要はありません。

- **Capture コンポーネント・インスタンスの識別。**

ビジネス指標モデルに関連したデータをキャプチャーするために、複数の Capture コンポーネント・インスタンス (つまりDB2 Capture ユーティリティ) が使用されます。ビジネス指標モデルにサービスを提供するように割り当てられている Capture ユーティリティを判別するには、以下を実行します。

- Capture ユーティリティ構成を変更する対象のデータ移動サービスを指定します。
- 状態データベース (状態データベースからランタイム・データベースへのデータ移動サービスの場合) またはランタイム・データベース (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) の WBIRMADM.RMMETADATA メタデータ・テーブルを検査し、すべての Capture ユーティリティ名 (列 SRC_RM_CAP_SVR_NAME) を確認します。

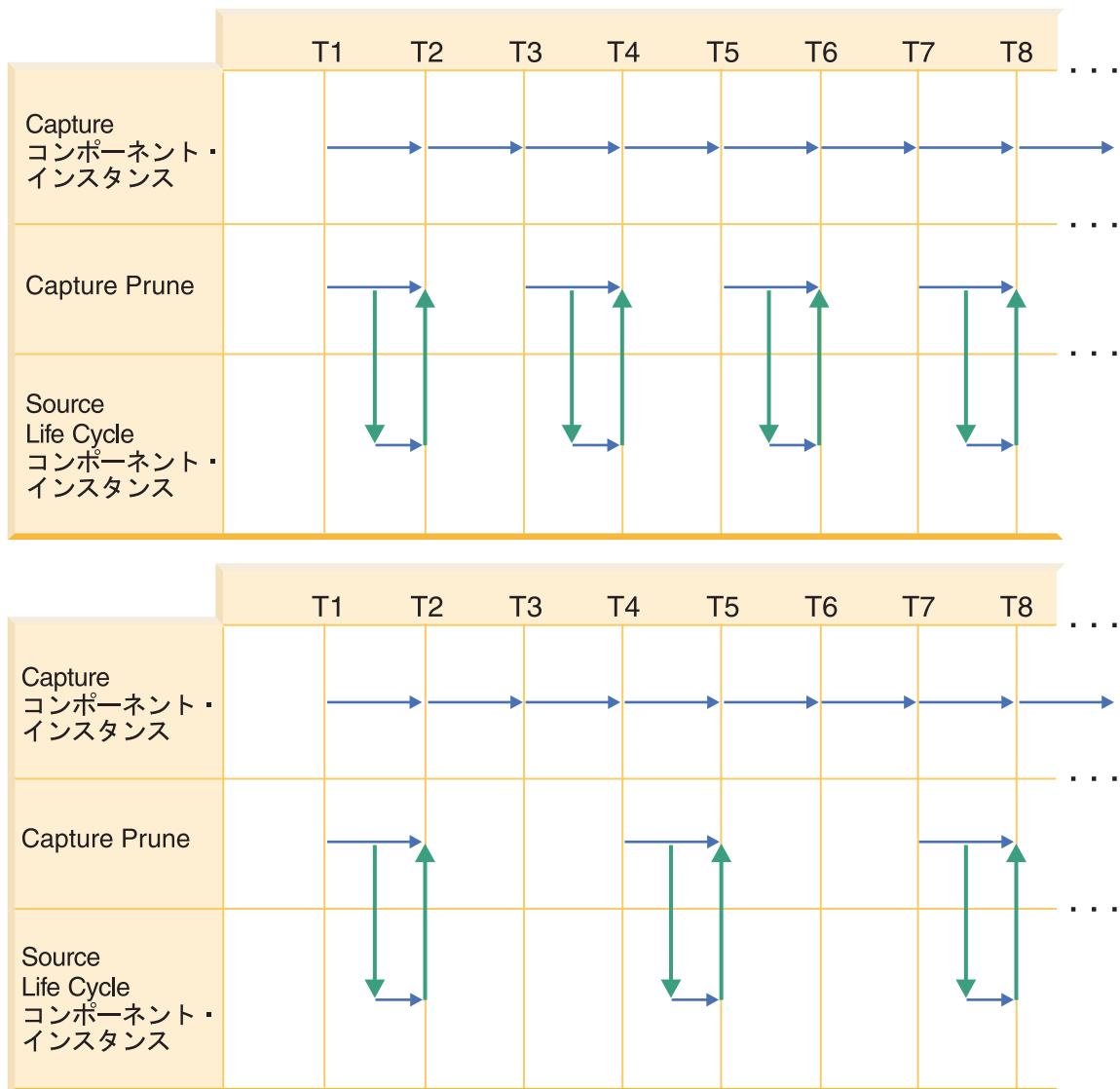
例: クエリー「SELECT OM_NAME, SRC_TAB_NAME, SERVICE_NAME, SRC_RM_CAP_SVR_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE_NAME='State to Runtime'」は、次のような結果になります。

OM_NAME	SRC_TAB_NAME	SERVICE_NAME	SRC_RM_CAP_SVR_NAME
STEW_S	wbi.CTX_TG4MUFT42JOT5F6R3KSDQDE2UI	State to Runtime	CAPTURE_1
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJR5HFQGGQE	State to Runtime	CAPTURE_1

上記の例では、Capture ユーティリティ CAPTURE_1 は、状態データベースのビジネス指標モデル STEW_S と関連付けられた 2 つのソース・テーブルに対するすべての変更をキャプチャーするように割り当てられています。

- **Capture 作業テーブルの整理間隔の変更。**

自動整理が使用可能 (autoprune パラメーターが「y」) した場合、Capture ユーティリティは、300 秒 (prune_interval パラメーターのデフォルト) ごとに作業テーブルを自動的に整理します。それぞれの整理アクティビティの結果、データベース・トリガーによってインプリメントされている Source Life Cycle コンポーネントのインスタンスが自動的に呼び出されます。Capture ユーティリティの整理間隔パラメーターを変更すると、ソース・テーブルが Source Life Cycle コンポーネントによって整理される頻度に直接影響します。以下の図は、Capture の整理間隔の変更が Source Life Cycle コンポーネントのインスタンスの呼び出しにどのような影響を与えるかを示します。



Capture インスタンスの `prune_interval` パラメーターを 2 つの時間単位 (例えば 300 秒) から 3 つの時間単位 (例えば 450 秒) に増加させると、次のようになります。

- Capture 作業テーブル内の行 (除去可能) が作業テーブル内により長く残存するようになるため、潜在的なスペース所要量が増加します。作業テーブルは大きくなりますが、システム負荷と偶発事故のリスクは低くなる可能性があります。
- ソース・テーブル内の行 (ライフ・サイクル保存ポリシーに基づいて除去可能) が、ソース・テーブル内に予想以上に長く残存するようになります。

一般に、Capture の `prune_interval` パラメーターが Life Cycle コンポーネントの `prune_interval` パラメーターよりも大きい値に設定されている場合は、Capture のパラメーター設定が優先します。Capture ユーティリティが実行されていない場合、またはその自動整理機能が使用不可の場合は、Source Life Cycle コンポーネントは実行されません。

Source Life Cycle コンポーネントの構成

それぞれのソース・データベース (状態データベースおよびランタイム・データベース) では、複数の Life Cycle コンポーネントのインスタンスが使用されています。トリガーによってインプリメントされているそれぞれのインスタンスは、そのデータ移動サービスのソース・データベースに配置された、制御テーブル WBIRMADM.RMPRUNECTRL で定義されている保存ポリシーを実行します。ライフ・サイクル保存ポリシーは、テーブルごとに指定されます。したがって、WBIRMADM.RMPRUNECTRL の 1 つの行が、整理を必要とする 1 つのテーブルに対応します。

TABLE_NAME	RETEN...	LAST_PRUNED	PRUNE_IN...	PRUNE_EN...	LOG...	ROW...
wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI	1440	Oct 11, 2005 4:40:...	5	1	0	0
wbi.AIR_BVSOYAP1DRWFD5HNQJR5HFQQQE	1440	Oct 11, 2005 4:40:...	5	1	0	0

• Source Life Cycle コンポーネント・インスタンスの識別。

特定のビジネス指標モデルに対する保存ポリシーを強制するために割り当てられているトリガーを判別するには、以下を実行する必要があります。

- ETL 構成を変更する対象のデータ移動サービスを確認します。
- 状態データベース (状態データベースからランタイム・データベースへのデータ移動サービスの場合) またはランタイム・データベース (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) の WBIRMADM.RMMETADATA テーブルを検査し、列 SRC_RM_PRUNE_TRG_NAME の関連トリガー名を検索します。

例: クエリー「SELECT OM_NAME, SRC_TAB_NAME, SERVICE_NAME, SRC_RM_PRUNE_TRG_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE_NAME='State to Runtime'」は、次のような結果になります。

OM_NAME	SRC_TAB_NAME	SERVICE_NAME	SRC_RM_PRUNE_TRG_NAME
STEW_S	wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	State to Runtime	WBIRMADM.MCPruneTrig_8
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE	State to Runtime	WBIRMADM.MCPruneTrig_9

この例では、2 つのトリガー (WBIRMADM.MCPruneTrig_8 と WBIRMADM.MCPruneTrig_9) が、状態データベースのビジネス指標モデル STEW_S のソース・テーブルのライフ・サイクル保存ポリシーを強制しています。保存ポリシーはテーブルによって定義され、Life Cycle コンポーネントのインスタンス名によっては定義されないため、Life Cycle コンポーネントの動作を変更する場合は、列 SRC_TAB_NAME を追跡しておくようにします。

• Source Life Cycle コンポーネントのインスタンス構成の変更。

- Life Cycle コンポーネントのインスタンスの使用可能化および使用不可化:

整理は、ご使用のシステムのパフォーマンスに大きな影響を与える可能性があります。整理が使用可能なときは、トランザクション・サーバー (状態) およびレポート・サーバー (ランタイム) が対処する必要のある情報量が削減されます。また、Life Cycle コンポーネントのパラメーターに応じて、各呼び出し中にそれらの同じテーブルに対する負荷も少し追加されます。整理が使用不可のときは、時間とともにソース・テーブルは大きくなり、それによってパフォーマンスが低下する可能性もあります。

デフォルトでは、ライフ・サイクル保存ポリシーに従って、ソース・テーブルが自動的に整理されます。一時的に整理を使用不可にするには、対応する WBIRMADM.RMPRUNECTRL エントリーを変更します。整理を使用可能にするには列 PRUNE_ENABLED を 1 に、整理を使用不可にするにはその他の任意の数値 (0 を推奨) に設定します。

次の構成が使用されている場合、ソース・テーブル wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI から行がパージされますが、テーブル wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE からは行がパージされません。クエリー「SELECT TABLE_NAME, PRUNE_ENABLED FROM WBIRMADM.RMPRUNECTRL」は、次のような結果になります。

TABLE_NAME	PRUNE_ENABLED
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	1
wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE	0

– 保存ポリシーの変更:

保存時間ポリシーは、ランタイム・データベースにあるソース・テーブルに対してのみ変更できます。WBIRMADM.RMPRUNECTRL の設定にかかわらず、状態データベースに配置されるすべてのテーブルに対して保存期間 0 が強制されます。保存期間は、行が除去可能になるまでソース・テーブルに残存する必要がある最小時間として定義されています。ただし、以下の 2 つの条件を満たす必要があります。制御テーブルを使用して、2 つの条件のうち 1 つ (分単位で指定される保存時間) のみをカスタマイズ可能です。削除の準備ができているとマークされており、少なくとも RETENTION_IN_MINUTES の期間ソース・テーブルに残存していた行は、除去可能になります。

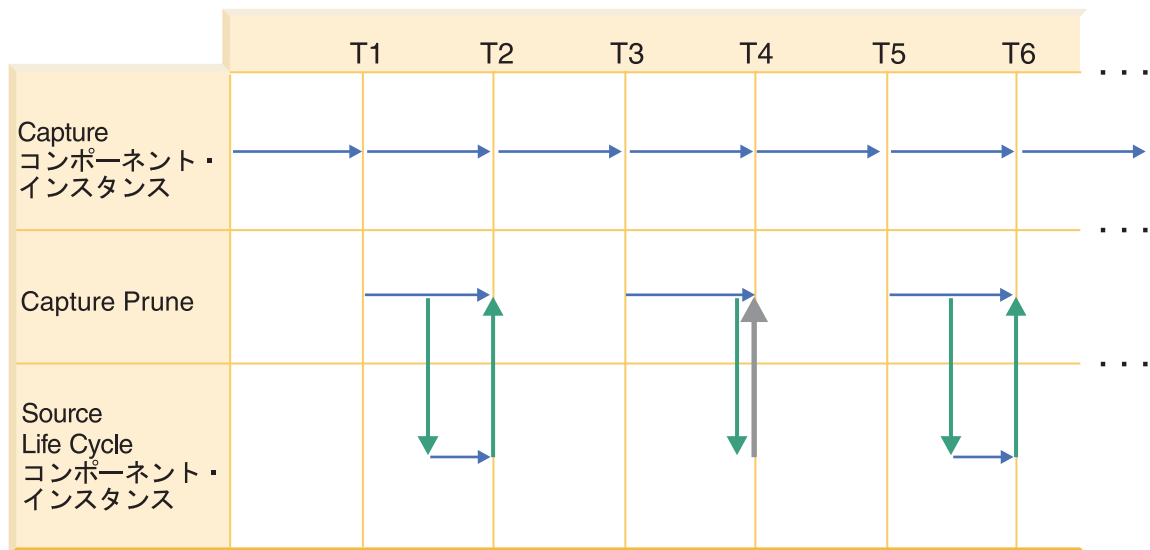
ランタイム・データベースのソース・テーブルのデフォルト構成を使用すると、サーバーによって削除の準備ができているとマークされている行は、除去可能になるまでに 1 日 (1440 分) 保持される必要があります。クエリー「SELECT TABLE_NAME, RETENTION_IN_MINUTES FROM WBIRMADM.RMPRUNECTRL」は、次のような結果になります。

TABLE_NAME	RETENTION_IN_MINUTES
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	1440
wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE	1440

WBIRMADM.RMPRUNECTRL 制御テーブルのエントリーに対する変更は、Source Life Cycle コンポーネントが呼び出されるたびに選出されます。

– ソース・データ整理のスケジューリング:

Capture 作業テーブルの整理間隔と Source Life Cycle コンポーネントの呼び出しの間には依存関係があります。以下の図に示すように、Source Life Cycle コンポーネントのインスタンスの呼び出し間に十分な時間が経過していない場合は、結果的に呼び出しが実行されるとは限りません。



Source Life Cycle コンポーネントが 4 つの時間単位ごとに実行されるようにスケジュールされているのに対し、Capture コンポーネントが 2 つの時間単位ごとに作業テーブルを整理するように構成されていると想定すると、T4 の時点での呼び出しは実行されません。

デフォルトのスケジュールを変更するには、WBIRMADM.RMPRUNECTRL の該当するエントリーを見つけ、実行間の最小遅延を分単位で表す PRUNE_INTERVAL の列値を変更します。

TABLE_NAME	LAST_PRUNED	PRUNE_INTERVAL
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	Oct 11, 2005 5:16:44 PM ...	5
wbi.AI_BVSOYAP1DRWVFD5HNQJR5HFQQQE		5

値を増加させると、実行の頻度は低くなります (ただし、呼び出し数は変更されません)。それぞれの実行で、ソース・テーブルの除去できる行を判別し、それらを除去します。ソース・データベースを定期的にモニターし、これらの削除により引き起こされるロックの結果としての潜在的なパフォーマンス上の問題を識別し、除去します。

(ターゲット) Apply コンポーネントの構成

Apply コンポーネントのインスタンスは、DB2 Apply レプリケーション・ユーティリティです。Capture ユーティリティによってキャプチャーされた変更は、デフォルトで、ターゲット・データベースのステージング・テーブルに継続的に適用されます。デフォルトのユーティリティ構成パラメーターは、ほとんどの環境に対して十分であり、したがって変更する必要はありません。

• Apply コンポーネントのインスタンスの識別。

ビジネス指標モデルと関連付けられた内部のステージング・テーブルに対するデータの変更を適用するために、複数の Apply コンポーネントのインスタンス (DB2 Apply ユーティリティ) が使用されます。ビジネス指標モデルにサービスを提供するように割り当てられている Apply ユーティリティを判別するには、次のようにします。

- Apply ユーティリティ構成を変更する対象のデータ移動サービスを確認します。
- ランタイム・データベース (状態データベースからランタイム・データベースへのデータ移動サービスの場合) またはヒストリー・データベース (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) の WBIRMADM.RMMETADATA メタデータ・テーブルを検査し、すべての Apply ユーティリティ名 (列 TGT_RM_APP_SVR_NAME) を確認します。クエリー「SELECT OM_NAME, SRC_TAB_NAME, SERVICE_NAME, TGT_RM_APP_SVR_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE_NAME='State to Runtime'」は、次のような結果になります。

OM_NAME	SRC_TAB_NAME	SERVICE_NAME	TGT_RM_APP_SVR_NAME
STEW_S	wbi.CTX_TQ4MLFT42JOT5F6R3KSDGDE2UI	State to Runtime	APPLY_4
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJR5HFQGGGE	State to Runtime	APPLY_4

この例では、状態データベースにキャプチャーされたビジネス指標モデル STEW_S に対するデータの変更は、Apply ユーティリティ APPLY_4 によって、ランタイム・データベースのステージング・テーブルに適用されます。

Apply がそれまでに Capture ユーティリティによって記録されたすべての (コミット済み) 変更の処理を完了するたびに、1 つ以上の ETL コンポーネントのインスタンスおよび Target Life Cycle コンポーネントのインスタンスが呼び出されます。

ETL コンポーネントの構成

ETL コンポーネントは、WebSphere Business Monitor にデータベースのストアード・プロシージャとしてインプリメントされています。これらのストアード・プロシージャは、どのデータ移動サービスの場合でも常にターゲット・データベース上にあります。したがって、状態データベースからランタイム・データベースへのデータ移動サービスに割り当てられたすべての ETL ストアード・プロシージャはランタイム・データベースに配置され、ランタイム・データベースからヒストリー・データベースへのデータ移動サービスに割り当てられた ETL ストアード・プロシージャはヒストリー・データベースにあります。

• ETL コンポーネントのインスタンスの識別。

ビジネス指標モデルと関連付けられた内部のステージング・テーブルに追加されたデータを処理するために、複数の ETL コンポーネントのインスタンスが設定されます。特定のビジネス指標モデルにサービスを提供するように割り当てられているストアード・プロシージャを判別するには、以下を実行します。

- ETL 構成を変更する対象のデータ移動サービスを確認します。
- ランタイム・データベース (状態データベースからランタイム・データベースへのデータ移動サービスの場合) またはヒストリー・データベース (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) の WBIRMADM.RMMETADATA メタデータ・テーブルを検査し、すべての ETL ストアード・プロシージャ名 (列 TGT_RM_SPETL_NAME) を識別します。クエリー「SELECT OM_NAME, SRC_TAB_NAME, TGT_TAB_NAME, SERVICE_NAME, TGT_RM_SPETL_NAME FROM

WBIRMADM.RMMETADATA WHERE SERVICE_NAME='State to Runtime']
は、次のような結果になります。

OM_NAME ⇅	SRC_TAB_NAME ⇅	TGT_TAB_NAME ⇅	SERVICE_NAME ⇅	TGT_RM_SPETL_NAME ⇅
STEW_S	wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	State to Runtime	WBIRMADM.WBIRMSP_10
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJRSHFQQGE	wbi.AI_BVSOYAP1DRWFD5HNGJRSHFQQGE	State to Runtime	WBIRMADM.WBIRMSP_14

この例では、状態データベースでキャプチャーされ、ランタイム・データベースのステージング・テーブルに適用された、ビジネス指標モデル STEW_S のデータ変更は、WBIRMADM.WBIRMSP_10 および WBIRMADM.WBIRMSP_14 という名前のストアード・プロシージャによって処理されます。正常に処理されたデータは、ランタイム・データベースのターゲット・テーブル (列 TGT_TAB_NAME によって示される) に保管されます。

• ETL コンポーネントのインスタンスの構成の変更。

ETL コンポーネントのインスタンスの構成は、WBIRMADM.RMCONTROL 制御テーブルに保管されます。状態データベースからランタイム・データベースへのデータ移動サービスに割り当てられたインスタンスは、構成をランタイム・データベースに保持し、その他はすべてヒストリー・データベースに保持します。構成に対して行われた変更は、次の開始時にストアード・プロシージャによって選出されます。制御テーブルを使用して構成可能な 3 つのオプションがあります。

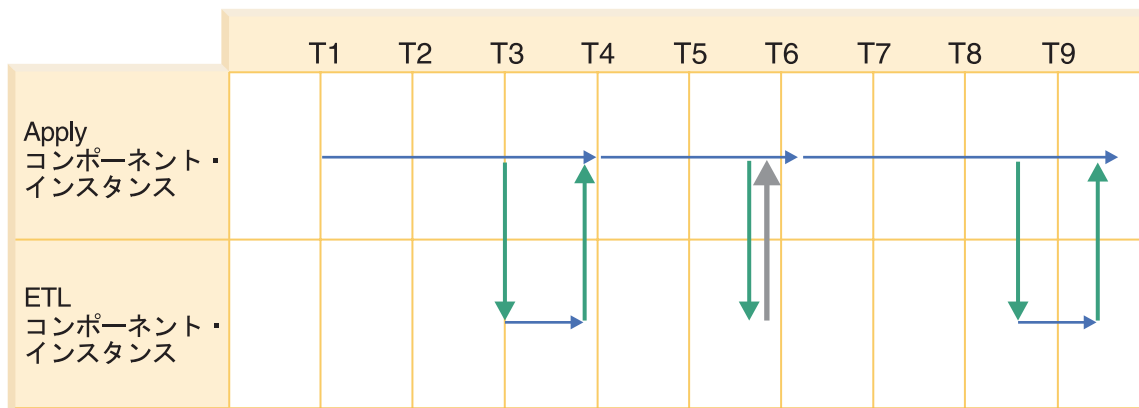
- 2 つの ETL 実行間の最小経過時間 (ETLSCHEDMETHOD、ETL_0_MINUTES)
- ロギング出力の細分度 (LOGLEVEL)
- トランザクション期間 (COMMITINTERVAL)

このテーブルのそれぞれの行は、データを取り込む必要のあるちょうど 1 つのターゲット・テーブルに対応する 1 つの ETL コンポーネントのインスタンスに対応しています。次の構成例では、構成に対する変更がどのようにインスタンスの動作に影響するかを示します。

TARGETTABLE ⇅	TGT_RM_SPETL_NAME ⇅	ETLSCHEDMETHOD ⇅	ETL_0_MINUTES ⇅	LOGLEVEL ⇅	COMMITINTERVAL ⇅
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	WBIRMADM.WBIRMSP_10	1	5	0	1000
wbi.AI_BVSOYAP1DRWFD5HNGJRSHFQQGE ...	WBIRMADM.WBIRMSP_14	1	5	0	1000

• ETL スケジュールの変更。

ETL コンポーネントのインスタンスは、Apply コンポーネントのインスタンスがサブスクリプション・セットの処理を完了するたびに呼び出されます。呼び出されると、ETL のインスタンスはそのスケジュールを確認し、処理を開始するか、または即時に Apply コンポーネントのインスタンスに制御を戻します。このインスタンスは、制御テーブル WBIRMADM.RMCONTROL に保管されている情報を使用して、実行する必要があるかを判別します。下の図は、呼び出しと実行の違いを示しています。スケジュールに従って、1 回目と 3 回目に ETL コンポーネントのインスタンスが実行されています。2 回目の呼び出しはスケジュール外で発生したため、処理アクティビティは実行されません。



状態データベースからランタイム・データベースへのデータ移動サービスおよびランタイム・データベースからヒストリー・データベースへのデータ移動サービスで ETL コンポーネントのインスタンスが実行される頻度は、さまざまな要因によって決定されます。

- 使用可能: データがいつ、ターゲット・テーブルでアクセス可能になるかを意味します。短い間隔を選択すると、データは早く使用可能になりますが、システム負荷も増加します。
- データ・ボリューム: レプリケーション・ユーティリティは、それが ETL コンポーネントのインスタンスによって処理されているかどうかにかかわらず、継続的に (または構成されたとおりに) ステージング・テーブルにデータを送信します。処理を必要とするデータが増加すると、使用されるデータベースのリソースが増加します。多くの場合、データを処理する頻度を増やすと、最大リソース使用量を削減できます。
- 処理時間: ETL プロセスは、ヒストリー・データベースのデータの処理よりも、ランタイム・データベースのデータの方が、時間がかかりません。それに応じて、スケジュールを計画します。使用する実行間の遅延を短くすると、スケジュールされた遅延よりも実行時間が長い場合には、良好なパフォーマンスを得られません。例えば、ETL コンポーネントのインスタンスが処理を完了させるために 60 秒かかるのとすると、ETL コンポーネントのインスタンスは順次実行されるため、30 秒のスケジュール間隔は事実上少なくとも 60 秒のスケジュール間隔になります。

現在、2 つのスケジューリング・モードがサポートされています。

- フレキシブルなスケジュール:

ETL インスタンスは、前回の実行 (LASTUPDATED) から少なくとも ETL_0_MINUTES が経過している場合に実行されます。例えば、制御テーブルに次の情報が含まれていると想定します。

TGT_RM_SPETL_NAME ⇄	TARGETTABLE ⇄	LASTUPDATED ⇄	ETL_0_MINUTES ⇄
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI	Oct 11, 2005 5:20:20 PM ...	60
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HINQJR5HFQQGE	Oct 11, 2005 5:20:20 PM ...	60

ストアード・プロシージャ WBIRMADM.WBIRMSP_10 は、2005 年 10 月 11 日午後 6 時 20 分 20 秒 (2005 年 10 月 11 日午後 5 時 20 分 20 秒 + 60 分) より前には実行されません。ストアード・プロシージャが 2005 年

10 月 11 日午後 6 時 20 分 20 秒より後に呼び出された場合、スケジュールはスリップする可能性があります。現在の時刻が午後 7 時であり、ストアード・プロシージャが予想されたとおりに午後 6 時 20 分に実行されなかったと想定します。ストアード・プロシージャは呼び出され、実行されます (約 40 分の遅延)。これは少なくとも午後 7 時 + 60 分 = 午後 8 時まで再実行されません。有効なスケジュールはスリップしました。これは、ETL プロシージャが 60 分ごとに毎時 20 分 to 実行されるようにスケジュールされていたにもかかわらず、現在は 60 分ごとに正時に実行されているためです。必要な場合、列 LASTUPDATED のタイム・スタンプ値を変更することによって、スケジュールをリセットできます。

固定した実行時間枠が必要でない場合は、このスケジューリング・メカニズムを使用します。この形式のスケジューリングを使用可能にするには、同一のビジネス指標グループに割り当てられているすべてのストアード・プロシージャに対して、WBIRMADM.RMCONTROL の列 ETLSCHEDMETHOD を 0 に設定します。

TGT_RM_SPETL_NAME ⇅	TARGETTABLE ⇅	ETLSCHEDMETHOD ⇅
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	0
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HNQJRS5HFQQQE ...	0

– 固定スケジュール:

これは、すべての ETL コンポーネントに対するデフォルトのスケジュールです。ETL コンポーネントのインスタンスは、現在時刻が NEXTSTARTTIME を過ぎている場合に実行されます。スリップを回避するために、ストアード・プロシージャが実行されるたびに、現在時刻と前にスケジュールされた実行時刻に基づいて、次回にスケジュールされる実行時刻が計算されます。以下の例は、このスケジュールを説明しています。

TGT_RM_SPETL_NAME ⇅	TARGETTABLE ⇅	LASTUPDATED ⇅	ETL_0_MINUTES ⇅	NEXTSTARTTIME ⇅
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	Oct 11, 2005 5:20:20 PM ...	60	Oct 11, 2005 6:20:20 PM ...
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HNQJRS5HFQQQE ...	Oct 11, 2005 5:20:20 PM ...	60	Oct 11, 2005 6:20:20 PM ...

現在時刻が午後 7 時で、ストアード・プロシージャが予想されたとおりに午後 6 時 20 分に実行されなかったと想定します。同じ日の NEXTSTARTTIME (午後 6 時 20 分) を過ぎているため、ストアード・プロシージャは実行されます。次の実行は、オリジナルのスケジュールに従って、午後 7 時 20 分にスケジュールされます。フレキシブル・スケジュールの場合のように、午後 8 時にはスケジュールされません。ストアード・プロシージャが事前定義の時間枠内で実行を開始する必要がある場合は、このスケジューリング方式を使用します。この形式のスケジューリングを使用可能にするには、同一のビジネス指標グループに割り当てられているすべてのストアード・プロシージャに対して、WBIRMADM.RMCONTROL の列 ETLSCHEDMETHOD を 1 に設定します。

TGT_RM_SPETL_NAME ⇅	TARGETTABLE ⇅	ETLSCHEDMETHOD ⇅
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	1
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HNQJR5HFQQQE ...	1

同一のビジネス指標モデルに属する ETLコンポーネントのインスタンスに対しては、それらのインスタンス間に相互依存性があるため、同一のスケジューリング方式を使用することを強くお勧めします。これは、ヒストリー・データベースにおいて、および間隔が長い (数時間以上) スケジュールの場合に、特に重要です。ETLSCHEDMETHOD を 0 か 1 以外の値に設定すると、ETL コンポーネントのインスタンスは使用不可になります。

・ ロギング・レベルの変更。

ストアード・プロシージャによって 2 つのロギングのレベル、最小 (0) および最大 (1)、がサポートされます。最小のデフォルト設定を変更するには、ロギング・レベルを変更する必要のあるストアード・プロシージャ

(TGT_RM_SPETL_NAME) における WBIRMADM.CONTROL の列 LOGLEVEL の値を変更します。すべてのロギング出力は WBIRMADM.RMLOG に追加されます。2 つの例のストアード・プロシージャ WBIRMADM.WBIRMSP_10 および WBIRMADM.WBIRMSP_14 は、両方とも最小ロギングを実行します。

ENTRYSTMP ⇅	NAME ⇅	OPERATION ⇅	ISTRACEENTRY ⇅	ID
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	SP_START	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	DEL_TEMP	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	INS_TEMP	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	FETCH_TARGET_...	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	SP_END	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_10	SP_START	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_10	DEL_TEMP	0	

ログ・テーブルは自動的に整理されないため、定期的に DBA によってモニターされる必要があります。特に他の方法を指示されていない限り、ロギング出力は最小に保ってください。

・ トランザクション期間の変更。

ストアード・プロシージャによって正常に処理されたデータは、即時にターゲット・テーブルに書き込まれます。ただし、ターゲット・テーブルへの更新は、コミット間隔設定 (WBIRMADM.RMCONTROL の列 COMMITINTERVAL) に応じて、指定された行数が処理されるまで、または処理する行がなくなるまで永続的にはなりません。COMMITINTERVAL の値を増加させると (例えば 1500 に)、ストアード・プロシージャは、変更をコミットする前に、より多くのデータを処理するようになります。ターゲット・テーブルに対するロックがより長く保持され、同一のテーブルにアクセスしようとしているその他のコンポーネントに悪影響を与える可能性があります。期間を減少させると (例えば 500 に)、ターゲット・テーブルで使用可能になる前に処理する必要のある行の数は少なくなり、より早くロックが解除されます。

TARGETTABLE ⇅	TGT_RM_SPETL_NAME ⇅	ETL_0_MINUTES ⇅	LOGLEVEL ⇅	COMMITINTERVAL ⇅
wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	WBIRMADM.WBIRMSP_10	5	0	1500
wbi.AIR_BVSOYAP1DRWFD5HNQJR5HFQQQE ...	WBIRMADM.WBIRMSP_14	5	0	500

Target Life Cycle コンポーネントの構成。

Apply コンポーネントのインスタンスによって、新規データまたは更新データが適用されている間は、ETL 作業テーブルは大きくなり続けます。ストアード・プロシージャによってインプリメントされている 1 つの Target Life Cycle コンポーネントのインスタンスが、それぞれのターゲット (ランタイムおよびヒストリー) データベースの 1 つの作業テーブルに割り当てられます。各インスタンスは、制御テーブル WBIRMADM.RMPRUNECTRL で定義されている内部の保存ポリシーを強制します。ソース・テーブルの場合と同様に、ETL 作業テーブルのライフ・サイクル保存ポリシーは、テーブルごとに指定されます。したがって、WBIRMADM.RMPRUNECTRL の 1 つの行が、整理を必要とする 1 つのテーブルに対応します。

• Target Life Cycle コンポーネント・インスタンスの識別。

特定のビジネス指標モデルに対して ETL 作業テーブルの保存ポリシーを強制するように割り当てられているストアード・プロシージャを判別するには、以下を実行します。

- ETL 構成を変更する対象のデータ移動サービスを確認します。
- ランタイム・データベース (状態データベースからランタイム・データベースへのデータ移動サービスの場合) またはヒストリー・データベース (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) の WBIRMADM.RMMETADATA テーブルを検査し、次のテーブルの列 TGT_RM_APP_PRUNE_SP_NAME の関連するストアード・プロシージャ名を検索します。

OM_NAME ⇄	SRC_TAB_NAME ⇄	TGT_RM_APP_STG_TAB_NAME ⇄	TGT_RM_APP_PRUNE_SP_NAME ⇄	SERVICE_NAME ⇄
STEW_S	wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	APP.CCD_6	WBIRMADM.WBIRMSP_P_13	State to Runtime
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJR5HFQGGQE	APP.CCD_7	WBIRMADM.WBIRMSP_P_17	State to Runtime

この例では、2 つのストアード・プロシージャ (WBIRMADM.WBIRMSP_P_13 と WBIRMADM.WBIRMSP_P_17) が、ランタイム・データベースのビジネス指標モデル STEW_S の ETL 作業テーブルに対するライフ・サイクル保存ポリシーを強制しています。保存ポリシーはテーブルによって定義され、Life Cycle コンポーネントのインスタンス名によっては定義されていないため、Life Cycle コンポーネントの動作変更を計画する場合は、列 TGT_RM_APP_STG_TAB_NAME を追跡してください。

• Target Life Cycle コンポーネント・インスタンスの構成の変更。

デフォルト構成は、ほとんどのデプロイメントに対して適切ですが、以下で説明するように微調整できます。

- Target Life Cycle コンポーネントのインスタンスの使用可能化および使用不可化。

デフォルトでは、ETL 作業テーブルはライフ・サイクルの保存ポリシーに従って自動的に整理されます。一時的に整理を使用不可にするには、対応する WBIRMADM.RMPRUNECTRL エントリを変更します。整理を使用可能にするには列 PRUNE_ENABLED を 1 に、整理を使用不可にするにはその他の任意の数値 (0 を推奨) に設定します。制御テーブル

WBIRMADM.RMPRUNECTRL がランタイム・データベースに以下のエントリが含まれる場合は、両方の ETL 作業テーブルが自動的に整理されます。

TABLE_NAME ⇅	PRUNE_ENABLED ⇅
APP.CCD_6	1
APP.CCD_7	1

Target Life Cycle コンポーネントのインスタンスを使用不可にする前に、関連するテーブル・スペースのコンテナに使用可能なスペースが十分にあることを確認します。モニター・サーバーがソース・テーブルの行を更新するたびに、1 行が ETL 作業テーブルに追加されます。したがって、ソース・テーブルの 1 つの行が、作業テーブルでは一時的に複数の行によって表される場合があります、このため作業テーブルはソース・テーブルよりもかなり早く大きくなります。WBIRMADM.RMPRUNECTRL への変更は、Target Life Cycle コンポーネントのインスタンスが次回呼び出されたときに選出されます。

- 保存ポリシーの変更。

ETL コンポーネントのインスタンスによって正常に処理されたすべての行は、作業テーブルから除去できます。ランタイム・データベースとヒストリー・データベースの両方に対するデフォルトの保存期間は 0 分に設定されています。

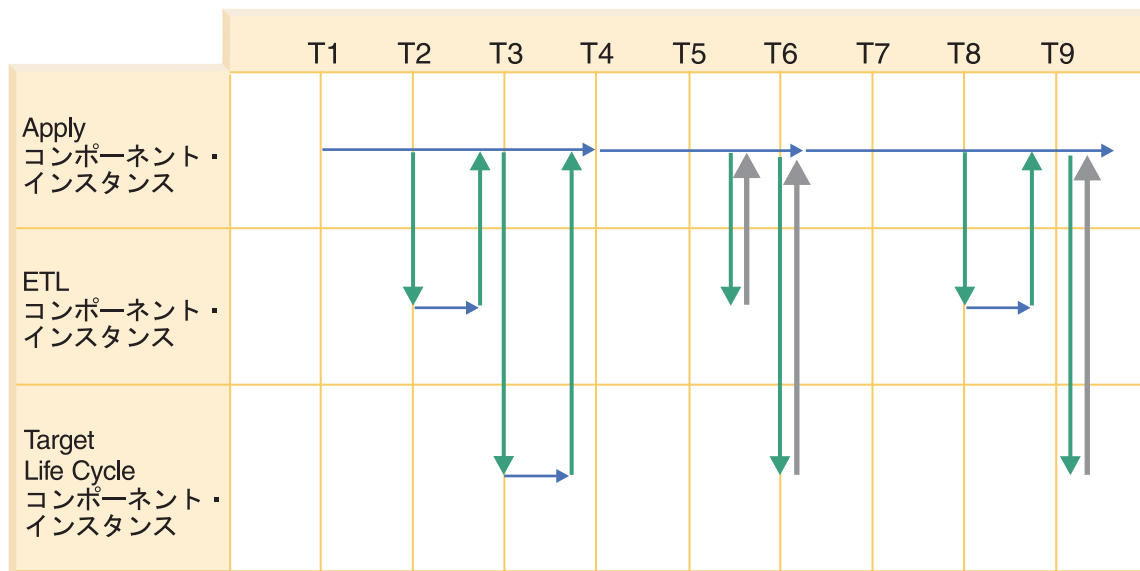
TABLE_NAME ⇅	RETENTION_IN_MINUTES ⇅
APP.CCD_6	0
APP.CCD_7	0

すべての使用可能な行は、Target Life Cycle コンポーネントのインスタンスが次回呼び出されたときに除去されます。いつ Life Cycle コンポーネントを実行するかはスケジュールが決定するため、保存期間をゼロに設定しても即時に除去されるとは限りません。

ユーザーは、WBIRMADM.RMPRUNECTRL の列 RETENTION_IN_MINUTES を別の期間 (分単位) に変更することで、このテーブルのデータを保持できます。

- ETL 作業テーブルの整理のスケジューリング。

Target Life Cycle コンポーネントのスケジューリングの元になる概念は、ETL コンポーネントのスケジューリングと非常に似ています。Apply サイクルおよびすべての関連する ETL コンポーネントのインスタンスが完了すると、Target Life Cycle コンポーネントのインスタンスが順次呼び出されます。



作業テーブルの整理が使用可能で、実行するようにスケジュールされている場合は、呼び出しは結果的に実行されます。上記の例では、Target Life Cycle コンポーネントのインスタンスは、時間 T3 で 1 回のみ実行されます。時間 T6 および T9 で実行されなかった理由はいくつかあります。

- T4 と T6 の間で Target Life Cycle コンポーネントのインスタンスの構成が変更され、整理が使用不可になった。
- T3 と T9 間の経過時間が、この Target Life Cycle コンポーネントのインスタンスに対して指定されている間隔よりも短い。

注: この例では、ETL コンポーネントのインスタンスと Target Life Cycle コンポーネントのインスタンスのスケジュールが異なります (整理が使用不可化されなかったと想定)。

一般的に、対応するすべてのインスタンスに対して同一のスケジュールを使用するか、またはライフ・サイクルのインスタンスに対して、より長いサイクルのスケジュールを使用します。これは、既に ETL コンポーネントのインスタンスによって正常に処理されていない限り、データは整理されないためです。デフォルトのスケジュールを変更するには、WBIRMADM.RMPRUNECTRL の該当するエントリを見つけ、実行間の最小遅延を分単位で表す PRUNE_INTERVAL の列値を変更します。

TABLE_NAME ⇅	LAST PRUNED ⇅	PRUNE_INTERVAL ⇅
APP.CCD_6	Oct 11, 2005 4:40:20 PM ...	1440
APP.CCD_7	Oct 11, 2005 4:40:20 PM ...	1440

整理間隔を増加させると、実行の頻度は低くなりますが、呼び出し数は変更されません。それぞれの実行で、作業テーブルのどの行を除去できるかを判別し、それらを除去します。上記の構成が使用されている場合、作業テーブル APP.CCD_6 および APP.CCD_7 は、1 日 (1440 分) あたり約 1 回整理されます。前回の整理は 10 月 11 日午後 4 時 40 分に実行されたため、10 月 12 日の午後 4 時 40 分頃に再度実行されます。

データ移動サービスの構成パラメーターの要約

次の表に、それぞれの data movement services コンポーネントで提供されているパラメーターのうち、最もよく使用されるものを要約します。構成パラメーターについての詳細は、「DB2 ユニバーサル・データベース レプリケーションの手引きおよび解説書」を参照してください。

コンポーネント	パラメーター名	デフォルト値	有効な値	パラメーター・ロケーション
Capture	autoprun	Y		
Capture	prune_interval (秒)	300		
ソース・ライフ・サイクル	PRUNE_ENABLED	1	0 - 使用不可 1 - 使用可能	データ移動サービスのソース DB: WBIRMADM.RMPRUNECTRL
ソース・ライフ・サイクル	RETENTION_IN_MINUTES	0 - 状態からランタイム 1440 - ランタイムからヒストリー	0 から BIGINT に対する DB2 の制限値まで	データ移動サービスのソース DB: WBIRMADM.RMPRUNECTRL
ソース・ライフ・サイクル	PRUNE_INTERVAL (分)	5	0 から BIGINT に対する DB2 の制限値まで	データ移動サービスのソース DB: WBIRMADM.RMPRUNECTRL
ETL	ETLSCHEDMETHOD	1	0 - フレキシブル・スケジューリング 1 - 厳密な間隔スケジューリング その他 - ETL 使用不可	データ移動サービスのターゲット DB: WBIRMADM.RMCONTROL
ETL	ETL_0_MINUTES	5 - 状態からランタイム 1440 - ランタイムからヒストリー	0 から INTEGER に対する DB2 の制限値まで	データ移動サービスのターゲット DB: WBIRMADM.RMCONTROL
ETL	LOGLEVEL	0	0 - 通常のログギングの場合 1 - トレース・ログギングの場合	データ移動サービスのターゲット DB: WBIRMADM.RMCONTROL
ETL	COMMITINTERVAL (レコード数)	1000	0 - 終了までコミット使用不可 1 - 各レコードをコミット n - BIGINT に対する DB2 の制限値	データ移動サービスのターゲット DB: WBIRMADM.RMCONTROL
ターゲット・ライフ・サイクル	PRUNE_ENABLED	1	0 - 使用不可 1 - 使用可能	データ移動サービスのターゲット DB: WBIRMADM.RMPRUNECTRL
ターゲット・ライフ・サイクル	RETENTION_IN_MINUTES	0	0 から BIGINT に対する DB2 の制限値まで	データ移動サービスのターゲット DB: WBIRMADM.RMPRUNECTRL
ターゲット・ライフ・サイクル	PRUNE_INTERVAL (分)	1440	0 から BIGINT に対する DB2 の制限値まで	データ移動サービスのターゲット DB: WBIRMADM.RMPRUNECTRL

注: IBM® は、上記で参照されるデータベース・テーブルおよび列を変更する権利を有します。このため、後続のリリースでは一部のテーブルおよび列の変更、除去、または追加が行われる可能性があります。リリースにより異なる場合があるとして、ここで言及した内容または構造に依存する場合は、お客様自身の責任で行ってください。IBM は、そのような変更が発生した場合はそれを文書化します。

変更管理および成果物の生成

ビジネス指標モデルには、多くのイベントとプロセスの定義が含まれます。これらの定義に基づいて、Schema Generator は、データベース・テーブル、Cube Views の定義、およびレプリケーション・スクリプトの作成に必要な、対応する成果物を生成します。ビジネス指標モデルの変更は、生成された成果物の変更を招きます。

このような変更が発生した場合は、Schema Generator を再実行して新規ビジネス指標モデル・スクリプトを生成する必要があります。このアクティビティは、変更管理と呼ばれます。

以下の場合に変更管理が必要です。

- 新規プロセスが追加され、その結果新規テーブルが追加される。
- ディメンションの一部でない、または新規ディメンションに作成された新しいメトリックが追加され、新規の列が適切なプロセス・テーブルに追加される。
- タイプ・ストリングのメトリックの長さが変更され、その結果対応する列の長さが変更される。

ビジネス指標モデルのあらゆる変更においても、以下を繰り返す必要があります。

- 変更されたビジネス指標モデルを、WebSphere Business Monitor 管理コンソールの Schema Generator にインポートし、対応する成果物を生成する。
- 新規に生成されたデータベース定義言語 (DDL) スクリプトを実行して、変更内容でデータベースを更新する。
- レプリケーション・スクリプトをデプロイして、変更後の状態データベース、ランタイム・データベース、およびヒストリー・データベースを同期させる。
- 新規に生成された Cube Views 定義をデプロイする。
- ビジネス指標モデルを WebSphere Business Monitor 管理コンソールにデプロイする。

Schema Generator は、ビジネス指標モデルの以前のバージョンと新規バージョンを調べます。新規モデルがデプロイされていないか、またはリポジトリに存在していない場合は、Schema Generator は、新規モデルに対応する成果物を生成します。新規モデルの前のバージョンが存在する場合は、Schema Generator はデプロイ済みのモデル・バージョンと新規モデル・バージョンとの間の相違点を調べます。変更が見つかった場合は、それらの相違点に基づいてデータベースを変更するために適切なスクリプトが再生成されます。既存モデルから新規モデルを作成する方法の詳細については、WebSphere Business Modeler の資料を参照してください。

ビジネス指標モデルの変更の一部は、既存ビジネス指標モデルのデータベース・テーブルの変更制限によって制限されます。以下の変更を実行する場合は、モデル全

体を再生成し、新規モデルとして再デプロイする必要があります。新規成果物の全セットが生成され、デプロイされます。その変更内容は以下のとおりです。

- ビジネス指標の「**WebSphere Business Monitor での使用 (Usage in WebSphere Business Monitor)**」属性の変更。例えば、メトリックを「**実行中のプロセスに関するアクティブ・データ (Active data about running process)**」値から、「**ディメンション分析の量的データ (Quantitative data in dimensional analysis)**」値に変更します。
- メトリックが属するディメンション・グループの変更。
- ビジネス指標エディターのディメンション・メトリックに対する「**isPartOfDimensionKey**」チェック・ボックスの状況を変更する。
- メトリックのデータ・タイプの変更。メトリックのデータ・タイプの変更は、現行のメトリックを削除し、新規メトリックを作成することによって処理されます。
- ディメンション・キーの一部であるメトリックの削除。
- 既存ディメンションのディメンション・キーの一部としての新規メトリックを作成する。
- 累積タイプと非累積タイプとの間のタイマーを変更する。
- プロセス・モデルまたはアクティビティーを変更する。

注: プロセス全体を削除する場合、その結果メトリックが削除される可能性があります。変更のみ生成できます。

ビジネス指標モデルでの変更に応じて、3 つのデプロイメント・シナリオがあります。

- 新規モデルのデプロイメント
- 新規バージョン・モデルのデプロイメント
- 別のモデルのデプロイメント

新規プロセスの追加

ビジネス指標モデルに新規プロセスを追加すると、状態データベース、ランタイム・データベース、およびヒストリー・データベースに新規テーブルが追加されます。

新規プロセスを追加した後で、**Schema Generator** を使用して、作成されたデータベース (状態データベース、ランタイム・データベース、およびヒストリー・データベース) およびレプリケーション設定を変更するために必要なスクリプトを作成してください。

注: この変更を行う計画演習を行っていることが重要です。データベース計画について詳しくは、6 ページの『データベース成果物のデプロイメントの準備』を参照してください。

データベース間のレプリケーションを同期させるために、新規に追加されたプロセスに対応する新規テーブルでレプリケーション構成を更新する必要があります。これらのスクリプトをデプロイして、**WebSphere Business Monitor** データベースに新規テーブルを追加し、レプリケーション設定に適切な変更を行って、データベース・テーブルの変更を反映させます。

新規プロセスが追加されるときは、以下を実行する必要があります。

- 新規または更新済みビジネス指標モデルをデプロイする前に、すべてのデータベースをバックアップします。
- Schema Generator を使用して、新規テーブルの追加および既存データベースの変更のためにデプロイする必要のあるデータベース定義言語 (DDL) スクリプトを生成します。
- DB2 コマンド・ウィンドウで新規レプリケーション・スクリプトをデプロイし、新規プロセス・テーブルのレプリケーションを使用可能化します。
- DB2 Cube Views および ALPHABLOX[®] にキューブ・モデルを再インポートして、キューブ・モデルがヒストリー・データベースで作成された新規キューブを反映するようにします。

既存ビジネス指標グループへの新規列の追加

既存のビジネス指標グループに新規ビジネス指標 (メトリック) を追加すると、状態データベース、ランタイム・データベース、およびヒストリー・データベースの一部のインスタンス・テーブルに新規列が追加されます。

新規列で更新されるテーブルは、追加されるメトリックの「**WebSphere Business Monitor** での使用 (Usage in WebSphere Business Monitor)」属性に従って選択されます。以下のテーブルに、影響を受けるデータベースを示します。

WebSphere Business Monitor データベースでのメトリックの使用

WebSphere Business Monitor での使用 (Usage in WebSphere Business Monitor)	状態データベース	ランタイム・データベース	ヒストリー・データベース
一時計算 (Temporary calculation)	あり	なし	なし
実行中のプロセスに関するアクティブ・データ (Active data about running processes)	あり	あり	なし
ディメンション分析の量的データ (Quantitative data in dimensional analysis)	あり	あり	あり
ディメンション分析の集約グループ (Aggregation group in dimensional analysis)	あり	あり	あり

ビジネス指標エディターに新規ビジネス指標 (メトリック) を追加した後で、Schema Generator を使用して、作成したデータベース (状態データベース、ランタイム・データベース、およびヒストリー・データベース) およびレプリケーション設定を変更するために必要なスクリプトを生成してください。データベースを変更するには、DB2 コマンドのウィンドウにデータベース定義言語 (DDL) スクリプトをデプロイして、データベース・テーブルに新規列を追加します。

データベース間でレプリケーションを同期化するには、DB2 コマンドのウィンドウにレプリケーション・スクリプトをデプロイして、新規に追加されたメトリックに対応する新規列で DB2 Replication Center を更新します。

プロセスに新規メトリックが追加された場合、以下を行う必要があります。

- 新規または更新済みビジネス指標モデルをデプロイする前に、すべてのデータベースをバックアップします。
- そのプロセスのモニター・サービスを停止します。
- そのプロセスに対するレプリケーション・サービスを停止します。
- Schema Generator を使用して、新規列の追加および既存データベースの変更のためにデプロイする必要がある DDL スクリプトを生成します。
- 生成されたレプリケーション・スクリプトをデプロイして、データベースのトポロジーに変更を反映します。
- DB2 Cube Views および ALPHABLOXにキューブ・モデルを再インポートして、キューブがヒストリー・データベースで作成された新規ディメンションを反映するようにします。

データベース保守

WebSphere Business Monitor のデータベースは定期的な保守を必要とします。特定の DB2 ツールを使用して、データベースの用途およびパフォーマンスを強化することができます。

推奨されるツールは次のとおりです。

- Configuration Advisor。
- DB2 Web Health Center。データベース管理者 (DBA) に潜在的な問題をアラートで通知し、問題に対して推奨される解決策を提供することにより、DBA を支援します。DBA は、Web Health Center を使用してリモート側でインスタンスをモニターし、アラートの詳細と推奨事項を表示できます。
- REORG コマンドおよび REORGCHK コマンド。REORG は、オーバーフロー行を除去し、削除されたテーブルおよびインデックスの行のスペースを再利用します。このツールは、削除、更新、または挿入の回数が多い場合に役立ちます。REORGCHK は、DB2 最適化プログラム・ツールが使用する統計情報を更新します。このツールは、データベース更新によりデータベース・テーブルの統計情報が最新の状態になっていない場合に役立ちます。
- RUNSTATS コマンド。データベース・オブジェクトに関する統計情報を収集します。これらの統計情報は、データへのアクセスにおけるパスを選択するためのデータ検索時に使用できます。このため、DB2 は、最も効率的なアクセス・パスを選択するために必要な情報を入手できます。各データベースにおいてデータの累積が可能になると、管理者がすべてのデータベースのすべてのテーブルに対して DBA RUNSTATS コマンドを実行するまで、データベース・パフォーマンスは低下します。例:

```
RUNSTATS ON TABLE tablename WITH DISTRIBUTION AND DETAILED INDEXES ALL
```

注: *tablename* は、スキーマ名を使用して完全に修飾する必要があります。
コマンドの実行後、DB2 rebind コマンドを実行します。

`db2rbind <database_alias> -l logfile all`このパフォーマンスは、レプリケーションのパフォーマンスのみでなく、モニター・サーバーのイベント処理のパフォーマンスに影響を与えます。さらに、レプリケーションの `RMPRUNECTL` テーブルのデフォルトの整理間隔の設定は、現行の 1440 秒 (または 24 時間) の代わりに 0 に設定する必要があります。

DB2 の保守ユーティリティの詳細については、IBM DB2 の資料を参照してください。

データベースのバックアップおよびデプロイメント・エラーからのリカバリーもデータベース保守の一部です。

データベースのバックアップ

新規ビジネス指標モデルをデプロイする場合、または既存のものを更新する場合でも、WebSphere Business Monitor の新規データベース定義言語 (DDL) スクリプトを実行する前に、リポジトリ・データベース、状態データベース、ランタイム・データベース、およびヒストリー・データベースをバックアップする必要があります。

何らかの障害が発生した場合、バックアップにより安全なロールバック・ポイントを確保できます。最近のデータ収集を考慮しない場合でも、最近デプロイされたビジネス指標モデルのテーブルを含まない前の状態にデータベースをロールバックできます。

データベースのバックアップおよび復元についての詳細は、『DB2 データ・リカバリー』のセクションを参照してください。

デプロイメント・エラー後のリカバリー

レプリケーション成果物のデプロイ中にエラーに遭遇した場合。変更を元に戻すために、単一のビジネス指標モデルのデプロイメント内で実行されたアクションを元に戻す必要があります。

すべてのデプロイメントは、複数のステージを経て実行されます。以下に典型的なシナリオを示します。

1. DDL のデプロイメント
 - a. `state.ddl` をデプロイする。
 - b. `runtime.ddl` をデプロイする。
 - c. `datamart.ddl` をデプロイする。
2. データ移動サービスのデプロイメント
 - a. `State_to_Runtime_setup_source` をデプロイする。
 - b. `State_to_Runtime_setup_target` をデプロイする。
 - c. `Runtime_to_historical_setup_source` をデプロイする。
 - d. `Runtime_to_Historical_setup_target` をデプロイする。

実行するアクションを決定するには、障害が発生したポイントを識別する必要があります。例えば、`state.ddl` で障害が発生した場合は、単にトランザクションをロールバックして元の状態に戻すだけで済みます。しかし、`datamart.ddl` で障害が発生した場合は、`datamart.ddl` をロールバックしても、システムは `runtime.ddl` が正常に実行された後のポイントに戻るだけです。データ移動サービスのデプロイメントの途

中で発生した障害は、リカバリーが最も困難ですが、不可能ではありません。最初のデプロイメントはリカバリーが最も容易で、次が新規モデルのデプロイメントです。そして、変更モデルのデプロイメントのリカバリーは最も困難です。

レプリケーション・スクリプトのデプロイメント・エラーからリカバリーするには、識別、バックアップ、復元または除去、そして再デプロイの各ステージを実行します。

識別

- 発生したエラーを確認し、IBM サポートに連絡する必要があるかどうかを決定します。
- エラー発生時にデプロイされていたビジネス指標モデルを確認します。
- エラー発生時に作成または変更されていた Schema Generator のテーブルを確認します。
- エラー発生時に作成または変更されていた Schema Generator の成果物を確認します。
- リポジトリ・データベースで、ビジネス指標モデルの最新の有効バージョンを確認します。
- 変更管理のデプロイメントの場合は、以前のバージョンのモデルに対してデプロイされた成果物のロケーションを確認します。これにより、データベースの構造、その説明、および相互の関係がわかります。この情報は、データをバックアップし、後で復元する必要がある場合に重要になる可能性があります。
- 現行成果物およびデプロイメント・ログ・ファイルのロケーションを確認します。これらの情報は、問題判別のために重要であり、IBM サポートに提供することになる可能性があります。
- 変更管理のデプロイメントの場合は、任意の CCD テーブルに存在する未処理のデータがあるかどうかを確認します。 *WBIRMADM.RMMETADATA* テーブル (ランタイム・データベースおよびヒストリー・データベースで使用可能) を使用して、デプロイされていたビジネス指標モデル・プロジェクト名 (*OM_NAME*) を持つ関連 CCD テーブル (*TGT_RM_APP_STG_TAB_NAME*) を判別できます。 **I** または **U** でマークされた行は、処理されていない可能性があり、バックアップする必要があります。列 *SERVICE_NAME* には、CCD テーブルおよびターゲット・テーブルのロケーションが含まれ、単語 *to* の後の名前がこれを示します。すべての成果物を完全に除去し、完全に新しいセットを生成する場合は、*TGT_TAB_NAME* に対する関係のトラックを保持する必要があります。これは、Schema Generator は、同じ名前の CCD テーブルを生成しない可能性があり、デプロイメントが成功した後、ユーザーが新規 CCD テーブルにこのデータを復元する必要があるためです。

バックアップ

- 識別のステージに基づいて、バックアップを必要とするデータがあるかどうかを判断します。デプロイメントと同時にその他のビジネス指標モデルが実行されている可能性があるため、その他のビジネス指標モデルと関連付けられたデータベース・オブジェクトのバックアップが必要な可能性があります。

- CCD (Consistent-Change Data) テーブルのバックアップが必要な可能性があります。Schema Generator のこれらのテーブルには、ソース・データベース・テーブルまたはターゲット・データベース・テーブルにはまだ存在しない何らかの情報が存在する可能性があります。

注: 完了したイベントは、ヒストリー・データベースにのみ存在する場合があります。

復元または除去

- 以前のデータベースの復元、または手動による成果物の除去のどちらが容易かを判断します。
- **復元** 他のビジネス指標モデルが存在しない場合、または他のビジネス指標モデルにアクティビティーがない場合は、バックアップされたバージョンからの復元が有効な可能性があります。以前のデータベース・セットを復元し、それぞれのデータベースに対して再バインドが必要なアプリケーションを再バインドし、すべての Java ベースのストアード・プロシージャおよびユーザー定義関数を再登録します。

注:

- データベースのバックアップおよび復元についての詳細は、DB2 の資料の『データ・リカバリー』のセクションを参照してください。
- デプロイメントが完了したモデルについては、*WBIRMADM.RMMETADATA* テーブルに、除去すべきでないものについて情報が提供されます。ただし、デプロイメント中は、何らかの成果物および関係を決定するために、デプロイメント・ログを検査し、安全に除去できるものを判断する必要がある場合があります。

• 除去

- レプリケーション: ヒストリー・データベースとランタイム・データベース
 - そのビジネス指標モデルと関連付けられたすべての Capture サーバーを停止します。(Capture サーバーは、ランタイム・データベースおよび状態データベース上で実行します。)
 - そのビジネス指標モデルと関連付けられたすべての Apply サーバーを停止します。
 - そのビジネス指標モデル用のすべての ETL ストアード・プロシージャを除去します。
 - そのビジネス指標モデルで使用されているすべての ETL ステージング・テーブルを除去します。
 - ビジネス指標モデルの対応するターゲット・データベース内の *WBIRMADM.RMCONTROL* テーブルからすべての ETL 制御情報を除去します。
 - そのビジネス指標モデルで使用されているすべての ETL の Prune ストアード・プロシージャおよびトリガーを除去します。
 - *WBIRMADM.RMMETADATA* テーブルの列 *TGT_RM_APP_STG_TAB_NAME* にリストされた、接尾部 *_BKUP* および *_M* を持ち、さらにそのビジネス指標モデルに、*Runtime_to_Historical* (ヒストリー・データベースの場合) および *State_to_Runtime* (ランタイム・データベースの場合) の対応する

SERVICE_NAME も持つすべてのテーブルを除去します。

TGT_RM_APP_STG_TAB_NAME にリストされたテーブルは、後の手順で除去されるため、残しておきます。

- DB2 Replication Center を使用して、そのビジネス指標モデルにサービスを提供しているすべての Apply Subscription Set のメンバーを除去します。
 - Apply subscription set が空の場合は、そのサブスクリプション・セットを除去します。
 - Apply サーバーにサブスクリプション・セットがない場合は、Apply サーバーを除去します。
- WBIRMADM.RMMETADATA テーブルから、そのビジネス指標モデルと関連付けられたすべてのメタデータ・エントリーを除去します。また、ヒストリー・データベースを処理している場合はランタイム・データベース、およびランタイム・データベースを処理している場合は状態データベースで、WBIRMADM.RMMETADATA テーブルから同じエントリーを除去する必要もあります。ビジネス指標モデルの行のみを除去する必要があり、これらは、*Runtime_to_Historical* サービス名 (ヒストリー・データベースの処理時) および *State_to_Runtime* サービス名 (ランタイム・データベースの処理時) 内にあります。
- レプリケーション: ランタイム・データベースと状態データベース
 - そのビジネス指標モデルにサービスを提供しているすべての Capture サーバーを停止します。
 - そのビジネス指標モデルと関連付けられた Capture CD テーブルと関連するすべてのトリガーを除去します。
 - WBIRMADM.RMPRUNECTRL テーブルから、ビジネス指標モデルで 사용되는 Prune トリガーの Prune 制御情報を除去します。
 - DB2 Replication Center を使用して、そのビジネス指標モデルと関連付けられたすべてのテーブルに対するすべてのサブスクリプションを除去します。
 - WBIRMADM.RMMETADATA テーブルから、そのビジネス指標モデルと関連付けられたすべてのメタデータ・エントリーを除去します。
- データベース・スキーマ: 通常、スキーマ生成中のエラーは、変更されたモデルがデプロイされる前にロールバックされます。現行のレプリケーション成果物のセットには影響はありません。

再デプロイ

ビジネス指標モデルをサポートするすべての成果物が除去されたら、「古いデプロイメントを無視 (Ignore Older Deployments)」オプションを選択して再度 Schema Generator を実行します。生成されたスキーマが成功したら、データベース定義言語 (DDL) スクリプトをデプロイせずに、レプリケーション・スクリプトを再デプロイしてください。

データベースの作成と構成

WebSphere Business Monitor のデータベースの作成と構成は、インストール・プロセスの重要なフェーズです。

WebSphere Business Monitor には 4 つのデータベースがあります。

- リポジトリ
- 状態
- ランタイム
- ヒストリー

WebSphere Business Monitor のインストールに、データベースはランチパッドを使用してのみ作成できます。インストール後にデータベースが除去された場合は、ランチパッドを使用するか、または手動でデータベースを再作成できます。データベースの作成には、静的テーブル、テーブル・スペース、索引の作成、および適切なデータベース構成の設定が含まれます。状態データベース、ランタイム・データベース、およびヒストリー・データベースには、静的テーブルと動的テーブルの両方が含まれますが、リポジトリ・データベースには静的テーブルのみが含まれます。データベースの作成に必要なスクリプトは

<monitor_installation_dir¥install¥mondb に保管されます。

手動でデータベースを作成するには、以下の手順を順序どおりに実行します。

1. Windows システムの場合:

- 「スタート」->「すべてのプログラム」->「IBM DB2」->「コマンド行ツール」->「コマンド ウィンドウ」と進みます。
- リポジトリ・データベースを作成する場合は、スクリプト
db2CreateRepository.bat <%RepositoryDatabaseName%> <%DB2userID%>
<DB2Password> Create_Repository.sql <%Install_Directory%> を実行します。
- 状態データベースを作成する場合は、スクリプト db2CreateState.bat
createStateDB.ddl <%Install_Directory%> を実行します。
- ランタイム・データベースを作成する場合は、スクリプト
db2CreateRuntime.bat createRuntimeDB.ddl <%Install_Directory%> を実行します。
- ヒストリー・データベースを作成する場合は、スクリプト
db2CreateHistorical.bat createDatamartDB.ddl <%Install_Directory%> を実行します。

注: <%Install_Directory%> は、デフォルトで 「C:¥IBM¥Websphere¥Monitor」 になっています。

2. AIX® システムの場合:

- DB2 インスタンス・ユーザーとしてログインします。
- リポジトリ・データベースを作成する場合は、スクリプト
db2CreateRepository.sh <%RepositoryDatabaseName%> <%DB2userID%>
<DB2Password> <%PathToDDL%>/Create_Repository.sql <%Install_Directory%>
を実行します。
- 状態データベースを作成する場合は、スクリプト db2CreateState.sh
<%PathToDDL%>/createStateDB.ddl <%Install_Directory%> <%DB2UserID%>
を実行します。
- ランタイム・データベースを作成する場合は、スクリプト
db2CreateRuntime.sh <%PathToDDL%>/createRuntimeDB.ddl
<%Install_Directory%> <%DB2UserID%> を実行します。

- e. ヒストリー・データベースを作成する場合は、スクリプト
db2CreateHistorical.sh <%PathToDDL%>/createDatamartDB.ddl
<%Install_Directory%> <%DB2UserID%> を実行します。

注:

- 代替ディレクトリーにインストールした場合は、ご使用のインストール・ディレクトリーに基づいて適切なパスに置き換える必要があります。
- <%PathToDDL%> は、デフォルトで
/opt/IBM/WebSphere/Monitor/Install/mondb になっています。
- <%Install_Directory%> は、デフォルトで /opt/IBM/WebSphere/Monitor になっています。
- <%DB2UserID%> は、デフォルトで db2inst1 になっています。

実行時のデータベースの管理

実行時の WebSphere Business Monitor のデータベースの管理には、WebSphere Business Monitor 管理コンソールの Schema Generator から生成された成果物のデプロイメントが含まれます。これらの成果物のデプロイメントは、新規の、または変更されたビジネス指標モデルがインポートされるたびに繰り返されます。

実行時のデータベースの管理には、以下のタスクの完了が含まれます。

動的データベース・テーブルの作成

動的データベース・テーブルは、特定のビジネス指標モデルに対応します。これらのテーブルの作成に必要なスクリプトは、Schema Generator から生成されます。

動的データベース・テーブルの作成に必要なスクリプトを生成するには、WebSphere Business Monitor 管理コンソールで Schema Generator を実行する必要があります。

スクリプトは、テーブルと索引を作成し、状態データベース、ランタイム・データベース、およびヒストリー・データベースのそれぞれの動的テーブルに対する構成パラメーターを設定します。生成されるスクリプトのロケーションは、WebSphere Business Monitor 管理コンソールの Schema Generator の構成中に指定されます。

状態データベース:

以下の手順に従って、状態データベースに動的データベース・テーブルを作成します。スクリプトはユーザー定義のロケーションに保管されます。このロケーションは、WebSphere Business Monitor 管理コンソールでの Schema Generator 構成の設定時に定義されます。

状態データベースの動的データベース・テーブルの作成に必要なデータベース定義言語 (DDL) スクリプトは、ルート・ディレクトリーの **state.ddl** ファイルに保管されます。スクリプトをデプロイするには、以下の手順を実行します。

1. DB2 コマンド・ウィンドウをオープンします。UNIX の場合、シェル環境が構成されていれば、DB2 コマンド行プロセッサを呼び出すことができます。
2. スクリプト・ファイルのロケーションへのパスを変更します。
3. 新規ビジネス指標モデルをデプロイする前に、状態データベースをバックアップします。

4. コマンド **db2 terminate** を実行します。これにより、異なるコード・ページ値を使用している可能性のある前のバックグラウンド・プロセスが使用されず、新規バックグラウンド・プロセスが使用されてこの要求が処理されます。
5. **DB2CODEPAGE** 環境変数を 1208 に設定します。DB2 コマンド行プロセッサは、デフォルトで、現行コード・ページを使用して文字データを解釈します。ただし、生成される *state.ddl* には、**DB2CODEPAGE** 環境変数が 1208 に設定されていない限り破損する UTF-8 文字が含まれます。
 - a. UNIX オペレーティング・システムの場合。
 - *sh*、*ksh*、*bash* タイプのシェルを使用して、コマンド **export DB2CODEPAGE=1208** を実行します。
 - *csh*、*tsch* タイプのシェルを使用して、コマンド **setenv DB2CODEPAGE 1208** を実行します。
 - b. Windows オペレーティング・システムの場合、コマンド **set DB2CODEPAGE=1208** を実行します。
6. コマンド **db2 connect to <State_DB_Name>** を実行して、状態データベースに接続します。
7. コマンド **db2 +c -stvf state.ddl > state.log** を実行します。これにより、スクリプトが実行され、トラブルシューティングを目的として、トランザクションを記録するログ・ファイルが保管されます。コミットするか、ロールバックするかを決定する前に、ログ・ファイルにエラーがないことを確認します。ロールバックが必要な場合は、コマンド **db2 rollback** を実行して、アクションを元に戻します。エラーが発生していない場合は、コマンド **db2 commit** を実行して変更をコミットします。
8. コマンド **db2 disconnect <State_DB_Name>** を使用して、スクリプトの実行後に状態データベースから切断します。
9. コマンド **db2 terminate** を実行して、バックグラウンド・プロセスを終了します。

ランタイム・データベース:

以下の手順に従って、ランタイム・データベースに動的データベース・テーブルを作成します。スクリプトはユーザー定義のロケーションに保管されます。このロケーションは、WebSphere Business Monitor 管理コンソールでの Schema Generator 構成の設定時に定義されます。

ランタイム・データベースの実行データベース・テーブルの作成に必要なデータベース定義言語 (DDL) スクリプトは、ルート・ディレクトリーの **runtime.ddl** ファイルに保管されます。スクリプトをデプロイするには、以下の手順を実行します。

1. DB2 コマンド・ウィンドウをオープンします。UNIX の場合、シェル環境が構成されていれば、DB2 コマンド行プロセッサを呼び出すことができます。
2. スクリプト・ファイルのロケーションへのパスを変更します。
3. 新規ビジネス指標モデルをデプロイする前に、ランタイム・データベースをバックアップします。
4. コマンド **db2 terminate** を実行します。これにより、異なるコード・ページ値を使用している可能性のある前のバックグラウンド・プロセスが使用されず、新規バックグラウンド・プロセスが使用されてこの要求が処理されます。

5. **DB2CODEPAGE** 環境変数を 1208 に設定します。DB2 コマンド行プロセッサは、デフォルトで、現行コード・ページを使用して文字データを解釈します。ただし、生成される *runtime.ddl* には、**DB2CODEPAGE** 環境変数が 1208 に設定されていない限り破損する UTF-8 文字が含まれます。
 - a. UNIX オペレーティング・システムの場合。
 - *sh*、*ksh*、*bash* タイプのシェルを使用して、コマンド **export DB2CODEPAGE=1208** を実行します。
 - *csh*、*tsch* タイプのシェルを使用して、コマンド **setenv DB2CODEPAGE 1208** を実行します。
 - b. Windows オペレーティング・システムの場合、コマンド **set DB2CODEPAGE=1208** を実行します。
6. コマンド **db2 connect to <Runtime_DB_Name>** を実行して、ランタイム・データベースに接続します。
7. コマンド **db2 +c -stvf runtime.ddl > runtime.log** を実行します。これにより、スクリプトが実行され、トラブルシューティングを目的として、トランザクションを記録するログ・ファイルが保管されます。コミットまたはロールバックする前に、ログ・ファイルにエラーがないことを確認します。ロールバックが必要な場合は、コマンド **db2 rollback** を実行して、アクションを元に戻します。エラーが発生していない場合は、コマンド **db2 commit** を実行して変更をコミットします。
8. コマンド **db2 disconnect <Runtime_DB_Name>** を実行することで、スクリプトの実行後にランタイム・データベースから切断します。
9. コマンド **db2 terminate** を実行して、バックグラウンド・プロセスを終了します。

ヒストリー・データベース:

以下の手順に従って、ヒストリー・データベースに動的データベース・テーブルを作成します。スクリプトはユーザー定義のロケーションに保管されます。このロケーションは、WebSphere Business Monitor 管理コンソールでの Schema Generator 構成の設定時に定義されます。

ヒストリー・データベースの実行データベース・テーブルの作成に必要なデータベース定義言語 (DDL) スクリプトは、ルート・ディレクトリーの **datamart.ddl** ファイルに保管されます。スクリプトをデプロイするには、以下の手順を実行します。

1. DB2 コマンド・ウィンドウをオープンします。UNIX の場合、シェル環境が構成されていれば、DB2 コマンド行プロセッサを呼び出すことができます。
2. スクリプト・ファイルのロケーションへのパスを変更します。
3. 新規ビジネス指標モデルをデプロイする前に、ヒストリー・データベースをバックアップします。
4. コマンド **db2 terminate** を実行します。これにより、異なるコード・ページ値を使用している可能性のある前のバックグラウンド・プロセスが使用されず、新規バックグラウンド・プロセスが使用されてこの要求が処理されます。
5. **DB2CODEPAGE** 環境変数を 1208 に設定します。DB2 コマンド行プロセッサは、デフォルトで、現行コード・ページを使用して文字データを解釈します。

ただし、生成される *datamart.ddl* には、DB2CODEPAGE 環境変数が 1208 に設定されていない限り破損する UTF-8 文字が含まれます。

- a. UNIX オペレーティング・システムの場合。
 - *sh*、*ksh*、*bash* タイプのシェルを使用して、コマンド ***export DB2CODEPAGE=1208*** を実行します。
 - *csh*、*tsch* タイプのシェルを使用して、コマンド ***setenv DB2CODEPAGE 1208*** を実行します。
 - b. Windows オペレーティング・システムの場合、コマンド ***set DB2CODEPAGE=1208*** を実行します。
6. コマンド ***db2 connect to <Historical_DB_Name>*** を実行して、ヒストリー・データベースに接続します。このスクリプトは、変更を自動的にコミットせずに実行されます。
 7. コマンド ***db2 +c -stvf datamart.ddl > datamart.log*** を実行します。これにより、スクリプトが実行され、トラブルシューティングを目的として、トランザクションを記録するログ・ファイルが保管されます。コミットまたはロールバックする前に、ログ・ファイルにエラーがないことを確認します。ロールバックが必要な場合は、コマンド ***db2 rollback*** を実行して、アクションを元に戻します。エラーが発生していない場合は、コマンド ***db2 commit*** を実行して変更をコミットします。
 8. コマンド ***db2 disconnect <Historical_DB_Name>*** を実行することで、スクリプトの実行後にヒストリー・データベースから切断します。
 9. コマンド ***db2 terminate*** を実行して、バックグラウンド・プロセスを終了します。

注: 特定の環境下においては、既存のビジネス指標モデルの新規バージョンに対する *datamart.ddl* の実行時に、次のようなエラーが表示される可能性があります。SQL0605W 必要な記述を持つ索引「WBI.I_1133789461307」がすでに存在するために、その索引は作成されませんでした。SQLSTATE = 01550。その他のエラーが発生していなければ、これらのエラーは無視可能であり、トランザクションをコミットできます。

データ移動サービスのデプロイ

状態データベースからランタイム・データベース、およびランタイム・データベースからヒストリー・データベースへのデータ移動サービスをデプロイするには、事前に動的データベース・テーブルを作成する必要があります。動的データベース・テーブル作成スクリプトの実行中にエラーが発生すると、データ移動サービスのデプロイメント時に問題が発生します。

スキーマ生成時には、データ移動サービス・セットアップ・ファイルを含む最大 3 個の圧縮 (ZIP ファイルまたは JAR ファイル) アーカイブ (名前は DS_State_setup、DS_Runtime_setup、および DS_Datamart_setup) が作成されます。ビジネス指標モデルに対するスキーマ生成の初回実行時には、常に 3 個のアーカイブが作成されます。以降の生成時、例えばビジネス指標モデルの変更後などは、0、1、2、または 3 個の新規アーカイブが作成される可能性があります。ビジネス指標モデルの変更を受け入れるために既存のレプリケーション環境を変更する必要

がある場合にのみアーカイブが作成されます。データ移動サービス・デプロイメント・アーカイブは、「一般」タブの Schema Generator 管理コンソールの構成で指定されたディレクトリーに配置されます。

データ移動サービスのデプロイメントには、ターゲット・データベース (データの移動先) のデータベース・オブジェクトと同様に、ソース・データベース (データの移動元) のデータベース・オブジェクトの作成および構成も含まれます。

- *DS_State_setup* には、状態データベースからランタイム・データベースへのデータ移動サービスのソース・データベースのセットアップのためのデプロイメント・スクリプトが含まれます。
 - *DS_Runtime_setup* には、状態データベースからランタイム・データベースへのデータ移動サービスのターゲット・データベースのセットアップ用デプロイメント・スクリプト、およびランタイム・データベースからヒストリー・データベースへのデータ移動サービスのソース・データベースのセットアップ用デプロイメント・スクリプトが含まれます。
 - *DS_Datamart_setup* には、ランタイム・データベースからヒストリー・データベースへのデータ移動サービスのターゲット・データベースのセットアップ用デプロイメント・スクリプトが含まれます。
1. 状態データベースからランタイム・データベースへのデータ移動サービスのデプロイ:
 - a. 状態データベースからランタイム・データベースへのデータ移動サービスのソース成果物をデプロイするマシンを決定します。ほとんどの場合、これは状態データベースをホストするマシンです。
 - b. そのマシンに作業ディレクトリーを作成し、生成された *DS_State_setup* アーカイブをそのディレクトリーにコピー (マシンがリモートの場合は転送) します。オペレーティング・システムに依存するパス長の制限があるため、100 文字以下の長さのパスを選択する必要があります。
 - c. (Windowsの場合は .zip ファイル、UNIXの場合は .jar ファイルの) アーカイブを作業ディレクトリーに解凍します。
 - d. デプロイメント時には、さまざまな DB2 ユーティリティーを使用して、データベース・オブジェクトを作成および構成します。これらのツールを使用するには、データベース環境をセットアップする必要があります。Microsoft® Windowsの場合、DB2 コマンド・ウィンドウをオープンして実行できます。UNIXの場合、適切な環境変数が設定されていることを確認します。
 - e. *DS_State_setup* アーカイブを解凍したディレクトリーにナビゲートします。
 - f. *State_to_Runtime_setup_source.bat* (UNIXの場合、拡張子は .sh) を実行してプロンプトに従います。スクリプトを実行すると、特定のコマンドが「成功した」、「警告を生成した」、または「失敗した」のいずれであったかを示す状況メッセージが表示されます。
 - g. 生成されたログ・ファイル *State_to_Runtime_setup_source.log* に警告またはエラー・メッセージがないか検査します。エラー・メッセージが表示されている場合は、先に進まないでください。
 - h. 作業ディレクトリーをバックアップします。IBM サポートでは、トラブルシューティングのためにこの内容を使用することがあります。

- i. 状態データベースからランタイム・データベースへのデータ移動サービスのターゲット成果物をデプロイするマシンを決定します。ほとんどの場合、これはランタイム・データベースをホストするマシンです。
- j. そのマシンに作業ディレクトリーを作成し、生成された `DS_Runtime_setup` アーカイブをそのディレクトリーにコピー (マシンがリモートの場合は転送) します。オペレーティング・システムに依存するパス長の制限があるため、100 文字以下の長さのパスを選択する必要があります。
- k. (Windowsの場合は `.zip` ファイル、UNIXの場合は `.jar` ファイルの) アーカイブを作業ディレクトリーに解凍します。
- l. デプロイメント時には、さまざまな DB2 ユーティリティーを使用して、データベース・オブジェクトを作成および構成します。これらのツールを使用するには、データベース環境をセットアップする必要があります。Microsoft Windowsの場合、DB2 コマンド・ウィンドウをオープンして実行できます。UNIXの場合、適切な環境変数が設定されていることを確認します。
- m. `DS_Runtime_setup` アーカイブを解凍したディレクトリーにナビゲートします。
- n. `State_to_Runtime_setup_target.bat` (UNIXの場合は、拡張子は `.sh`) を実行してプロンプトに従います。スクリプトを実行すると、特定のコマンドが「成功した」、「警告を生成した」、または「失敗した」のいずれであったかを示す状況メッセージが表示されます。
- o. 生成されたログ・ファイル `State_to_Runtime_setup_source.log` に警告またはエラー・メッセージがないか検査します。エラー・メッセージが表示されている場合は、先に進まないでください。
- p. 作業ディレクトリーをバックアップします。IBM サポートでは、トラブルシューティングのためにこの内容を使用することがあります。
- q. 問題が報告されなかった場合は、状態データベースからランタイム・データベースへのデータ移動サービスがセットアップされています。
2. ランタイム・データベースからヒストリー・データベースへのデータ移動サービスのデプロイ:
 - a. ランタイム・データベースからヒストリー・データベースへのデータ移動サービスのソース成果物をデプロイするマシンを決定します。ほとんどの場合、これはランタイム・データベースをホストするマシンです。状態データベースからランタイム・データベースへのデータ移動サービスのターゲット成果物を同じマシンにデプロイした場合、必要なデプロイメント・ファイルが既に解凍されているため、以下の手順 e に進んでください。
 - b. ランタイム・データベースをホストするマシン以外のマシンでデプロイメントを実行する場合は、そのマシンに作業ディレクトリーを作成し、生成された `DS_Runtime_setup` アーカイブをそのディレクトリーにコピー (マシンがリモートの場合は転送) します。オペレーティング・システムに依存するパス長の制限があるため、100 文字以下の長さのパスを選択する必要があります。
 - c. (Windowsの場合は `.zip` ファイル、UNIXの場合は `.jar` ファイルの) アーカイブを作業ディレクトリーに解凍します。
 - d. デプロイメント時には、さまざまな DB2 ユーティリティーを使用して、データベース・オブジェクトを作成および構成します。これらのツールを使用

するには、データベース環境をセットアップする必要があります。Microsoft Windowsの場合、DB2 コマンド・ウィンドウをオープンして実行できます。UNIXの場合、適切な環境変数が設定されていることを確認します。

- e. DS_Runtime_setup アーカイブを解凍したディレクトリーにナビゲートします。
- f. Runtime_to_Historical_setup_source.bat (UNIXの場合、拡張子は .sh) を実行してプロンプトに従います。スクリプトを実行すると、特定のコマンドが「成功した」、「警告を生成した」、または「失敗した」のいずれであったかを示す状況メッセージが表示されます。
- g. 生成されたログ・ファイル Runtime_to_Historical_setup_source.log に警告またはエラー・メッセージがないか検査します。エラー・メッセージが表示されている場合は、先に進まないでください。
- h. 作業ディレクトリーをバックアップします。IBM サポートでは、トラブルシューティングのためにこの内容を使用することがあります。
- i. ランタイム・データベースからヒストリー・データベースへのデータ移動サービスのターゲット成果物をデプロイするマシンを決定します。ほとんどの場合、これはヒストリー・データベースをホストするマシンです。
- j. そのマシンに作業ディレクトリーを作成し、生成された DS_Datamart_setup アーカイブをそのディレクトリーにコピー (マシンがリモートの場合は転送) します。オペレーティング・システムに依存するパス長の制限があるため、100 文字以下の長さのパスを選択する必要があります。
- k. (Windowsの場合は .zip ファイル、 UNIXの場合は .jar ファイルの) アーカイブを作業ディレクトリーに解凍します。
- l. デプロイメント時には、さまざまな DB2 ユーティリティーを使用して、データベース・オブジェクトを作成および構成します。これらのツールを使用するには、データベース環境をセットアップする必要があります。Microsoft Windowsの場合、DB2 コマンド・ウィンドウをオープンして実行できます。UNIXの場合、適切な環境変数が設定されていることを確認します。
- m. DS_Runtime_setup アーカイブを解凍したディレクトリーにナビゲートします。
- n. Runtime_to_Historical_setup_target.bat (UNIXの場合、拡張子は .sh) を実行してプロンプトに従います。スクリプトを実行すると、特定のコマンドが「成功した」、「警告を生成した」、または「失敗した」のいずれであったかを示す状況メッセージが表示されます。
- o. 生成されたログ・ファイル State_to_Runtime_setup_source.log に警告またはエラー・メッセージがないか検査します。エラー・メッセージが表示されている場合は、先に進まないでください。
- p. 作業ディレクトリーをバックアップします。IBM サポートでは、トラブルシューティングのためにこの内容を使用することがあります。
- q. 問題が報告されなかった場合は、ランタイム・データベースからヒストリー・データベースへのデータ移動サービスが、このビジネス指標モデル用にセットアップされています。

データ移動サービスのオプションの構成

データ移動サービスのコンポーネントによって作成および構成される各 Capture サーバーについて、Capture コンポーネントの動作に影響を与える可能性がある 2 つのパラメーターがあります。これらのパラメーターは、lag_limit および startmode です。

パラメーター lag_limit および startmode のデフォルト値は「7 日」および「WARMSI」です。これらのパラメーターについての詳細は、「DB2 Information Integrator SQL レプリケーション・ガイドおよびリファレンス」を参照してください。

これらの設定は、成果物がデプロイされるまで変更できません。ただし、いずれかの Capture コンポーネント・サーバーを実行する前はこれらの設定を変更できます。あるいは、現在実行中の Capture サーバーのパラメーターを変更できます。

注: Capture サーバーの実行中に行われた変更を使用可能にするには、Capture サーバーを停止し、再始動する必要があります。

レプリケーション成果物のデプロイ中にパラメーター lag_limit および startmode のデフォルト設定が使用され、Capture サーバーが 7 日間より長い期間停止していて再始動された場合、Capture コンポーネントはエラーを戻します。このエラーは、データが古すぎるために Capture サーバーを実行できないことを示します。いくつかの方法でデフォルト値を上書きできます。以下に 3 つの方法について説明します。

1. <CAPTURESERVERSCHEMA>.IBMSNAP_CAPPARMS テーブルで指定されているようにデフォルトのパラメーターを変更します。レプリケーション成果物のデプロイ後は、ランタイム・データベースに対して以下のクエリーを実行することにより、データ移動サービス・コンポーネントが作成した Capture サーバーの数を判別できます。

```
CONNECT TO RUNTIME DATABASE
```

```
SELECT DISTINCT OM_NAME, SERVICE_NAME, SRC_RM_CAP_SVR_NAME
```

```
FROM WBIRMADM.RMMETADATA
```

```
ORDER BY 1,2,3
```

以下のような表が表示されます。

表 1. RMMETADATA の例

OM_NAME	SERVICE_NAME	SRC_RM_CAP_SVR_NAME
SubDoctor3	ランタイムからヒストリー	CAPTURE_18
SubDoctor3	状態からランタイム	CAPTURE_1
SubDoctor3	状態からランタイム	CAPTURE_115
SubDoctor3	状態からランタイム	CAPTURE_156
SubDoctor3	状態からランタイム	CAPTURE_194
SubDoctor3	状態からランタイム	CAPTURE_212
SubDoctor3	状態からランタイム	CAPTURE_250

表 1. RMMETADATA の例 (続き)

OM_NAME	SERVICE_NAME	SRC_RM_CAP_SVR_NAME
SubDoctor3	状態からランタイム	CAPTURE_41
SubDoctor3	状態からランタイム	CAPTURE_59
SubDoctor3	状態からランタイム	CAPTURE_97

OM_NAME は、WebSphere Business Modeler プロジェクトの名前です。
SERVICE_NAME はデータ移動サービスを示し、SRC_RM_CAP_SVR_NAME は、データ移動サービスの一部として使用されている Capture サーバーの ID (CAPTURE SCHEMA) です。上記の表では、ランタイム・データベースからヒストリー・データベースへのデータ移動サービスに対して 1 個の Capture サーバーが、および状態データベースからランタイム・データベースへのデータ移動サービスに対して 9 個の Capture サーバーがあります。

注: サーバー数およびサーバー名は、使用されているモデルおよび成果物の生成時に指定されたポリシー・パラメーターに基づいて変化します。

データベース・サービス・コンポーネントは、パラメーター lag_limit および startmode のすべてのオプションをサポートしますが、コールド・スタート (障害後の Capture サーバーの始動) の回数が増加すると、重大なパフォーマンス上の問題が発生することに注意する必要があります。コールド・スタートが頻繁に行われると、データ移動サービスの ETL コンポーネントは、単に変更を記録するのではなく、すべての既存レコードを処理します。変更が必要なすべての Capture サーバーが識別された後は、データベースのデフォルト・パラメーターを変更できます。各 Capture サーバーのデフォルト・パラメーターを変更すると、Capture サーバーを始動できます。

2. あるいは、Capture サーバーを始動するコマンド行を変更します。データベース・サービス成果物の生成時に、Capture サーバーおよび Apply サーバーを開始および停止できる有用なスクリプトが生成されます。Capture 開始スクリプト (StartCapture_#.bat or StartCapture_#.sh) は、ディレクトリー <data_movement_service_name>%source にあります。これらの各スクリプトには、Capture プログラムを開始するために使用される **asncap** コマンドが含まれます。これらのパラメーターについての詳細は、IBM DB2 の資料を参照してください。適宜開始スクリプトを変更して保管し、その後開始スクリプトを実行して、新しい設定を使用して Capture サーバーを実行します。
3. あるいは、実行時に Capture サーバーを変更します。実行中の Capture サーバーの設定を一時的に変更する方法について説明している、IBM DB2 の資料の指示に従ってください。

データ移動サービスのセットアップのファイナライズ

デプロイされた Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスは、デフォルトで、これらを開始したユーザーのクリデンシャルを使用しています。これは、特定のトポロジーには十分である可能性がありますが、代替クリデンシャルの使用が必要な 2 つのシナリオがあります。

- **最初のシナリオ - 代替ユーザー・クリデンシャル:** データベース管理者 (DBA) はユーザー *user1* としてログオンしますが、ユーティリティがソース・データベースからターゲット・データベースへのデータ移動にユーザー *user2* を使用します。
- **2 番目のシナリオ - 分散環境:** DBA は、ユーティリティを *machine1* 上で実行することを計画しています。ソース・データベースまたはターゲット・データベースが別のマシン *machine2* で保守されています。

これらのシナリオをサポートするためには、現行のユーザー証明書の代わりに使用するユーザー証明書を含むパスワード・ファイルを作成する必要があります。パスワード・ファイルは、デプロイメント時に自動的に作成されないため、この 2 つのシナリオに対して以下の手順を実行する必要があります。

1. ソース・データベースへの接続時に使用するユーザー ID とパスワード 情報を保管するファイルを作成します。DB2 コマンド行のウィンドウで、次のコマンドを入力し、*<place_holder_name>* のようにマークされているすべてのプレースホルダーを適切な値で置き換えます。

```
asnpwd INIT encrypt all using <password_file>. asnpwd ツールは空のファイル <password_file> を作成します。
```

呼び出し例: `asnpwd INIT encrypt all using password.aut`

2. レプリケーション・ユーティリティが接続する必要のあるそれぞれのデータベースのデータベース・アクセス情報 (ユーザー ID、パスワード、およびデータベース名) を保管します。DB2 コマンド行のウィンドウで、次のコマンドを入力し、*<place_holder_name>* のようにマークされているすべてのプレースホルダーを適切な値で置き換えます。

```
asnpwd ADD alias <DB_name> ID <user_ID> PASSWORD <Password> using <password_file>.
```

必要な場合、各データベースに対してこの手順を繰り返します。プログラムは、ユーザーが入力する情報を暗号化し、*<password_file>* に保管します。

呼び出し例:

- `asnpwd ADD alias STMD7 id MYUSRID password MYPASSWRD using password.aut`
 - `asnpwd ADD alias RTMD7 id MYUSRID2 password MYPASSWRD2 using password.aut`
3. 生成された実行可能開始スクリプト (StartCapture と StartApply) を変更することによって、ユーティリティの構成ファイルを更新します。レプリケーション・ユーティリティのコマンド行呼び出しに、パスワード・ファイル・パラメーターを付加します。ユーティリティは、デフォルトの証明書の代わりに、指定されたファイルに保管されている暗号化されたユーザー・クリデンシャルを使用します。パスワード・ファイルは、CAPTURE_PATH (または APPLY_PATH) パラメーターで定義されている作業ディレクトリーに配置する必要があります。

変更の例:

- Capture 開始スクリプトの元のファイル内容: db2cmd asncap
CAPTURE_SERVER=stmd7 CAPTURE_SCHEMA=CAPTURE_1
CAPTURE_PATH="c:¥tmp¥state_capture_log"
 - Capture 開始スクリプトの変更後のファイル内容: db2cmd asncap
CAPTURE_SERVER=stmd7 CAPTURE_SCHEMA=CAPTURE_1
CAPTURE_PATH="c:¥tmp¥state_capture_log" pwdfile="password.aut"
 - Apply 開始スクリプトの元のファイル内容: db2cmd asnaply
APPLY_QUAL=Apply_1 CONTROL_SERVER=RTMD7 APPLY_PATH="C:¥tmp¥apply"
 - Apply 開始スクリプトの変更後のファイル内容: db2cmd asnaply
APPLY_QUAL=Apply_1 CONTROL_SERVER=RTMD7 APPLY_PATH="C:¥tmp¥apply"
pwdfile="password.aut"
4. 手順 1 および 2 で作成された <password_file> を適切なディレクトリーにコピーします。レプリケーション・ユーティリティーは、開始時にそのパスワード・ファイルをオープンしようとしています。ファイル <password_file> が CAPTURE_PATH (または APPLY_PATH) パラメーターで示された作業ディレクトリーに存在しない場合は、エラーが発生します。作業ディレクトリー・パラメーターが指定されない場合は、ユーティリティーは現行作業ディレクトリーでそのファイルを見つけようとしています。

DB2 ユーティリティーについての詳細は、「DB2 Information Integrator SQL レプリケーション・ガイドおよびリファレンス」を参照してください。

開始および停止スクリプトの統合

データ移動サービスを開始および停止するプロセスを単純化するには、生成された開始および停止スクリプトを統合し、マスター・スクリプトを使用して呼び出すことができます。

Capture コンポーネントおよび Apply コンポーネントはデータベースが配置されたシステム内で実行される必要があるため、統合オプションは使用されているトポロジーによって異なります。どのようにスクリプトが統合されるかにかかわらず、初期化エラーの発生を回避するために、2 つのコンポーネント・インスタンスが同時に起動されないようにする必要があります。

各 Capture コンポーネントまたは Apply コンポーネントのインスタンスを別々に開始または停止することは可能ですが、すべてのコンポーネント・インスタンスの開始スクリプトおよび停止スクリプトの内容を統一して、単一のビジネス指標モデルに対するデータ移動サービスを開始または停止するために必要なスクリプトを 1 つのみすると有用です。スクリプトは、以下のようにして統合できます。

1. ソース・データベースの Capture コンポーネント・インスタンスの開始および停止スクリプトの識別。
2. ソース・データベースの Capture コンポーネント・インスタンスの開始および停止スクリプトを呼び出す Capture マスターの開始および停止スクリプトの作成。
3. ターゲット・データベースの Apply コンポーネント・インスタンスの開始および停止スクリプトの識別。
4. ターゲット・データベースの Capture コンポーネント・インスタンスの開始および停止スクリプトを呼び出す Capture マスターの開始および停止スクリプトの作成。

この統合の結果、1 つのビジネス指標モデルに対するデータ移動サービスを開始または停止するために実行する必要がある開始 (または停止) スクリプトは、4 つのみとなります。

2 つのデータ移動サービスを別々に開始または停止する必要がある場合は、さらに統合できます。この場合、必要な開始および停止スクリプトは 3 つのみです。

- 1 つのスクリプトは、状態データベース内のすべての Capture コンポーネント・インスタンスを開始 (停止) します。
- 1 つのスクリプトは、ランタイム・データベース内のすべての Capture コンポーネント・インスタンスと Apply コンポーネントを開始 (停止) します。
- 1 つのスクリプトは、ヒストリー・データベース内のすべての Apply コンポーネントを開始 (停止) します。

3 つのデータベースすべてが 1 つのシステム上に存在する場合は、これらの 3 つの統合スクリプトを、すべての Capture および Apply コンポーネント・インスタンスを開始または停止する単一のスクリプトにさらに統合できます。

異なるデプロイメントによって作成されたデータ移動サービスの開始および停止スクリプトの統合が必要な場合があります。データ移動サービスの初期デプロイメント中に、すべてのビジネス指標グループに対する開始および停止スクリプトが作成されます。ビジネス指標モデルへの変更の結果としての後続のデプロイメントには、既存のビジネス指標グループに対する開始および停止スクリプトは含まれません。代わりに、新規ビジネス指標グループに対する開始および停止スクリプトのみが使用可能になります。以前に作成された統合済みの開始および停止スクリプトを手動で更新する必要があります。

次の例は、3 つのビジネス指標グループを含むビジネス指標モデル *FinanceModel* の初期データ移動サービスのデプロイメントの場合について説明します。状態データベースに対して 3 つの Capture 開始および停止スクリプトが作成されます。その後、モデルが更新され、新規ビジネス指標が追加されました。デプロイメント中に、新規ビジネス指標グループに対する Capture 開始および停止スクリプトが 1 つのみ作成されます。データ移動サービスを使用可能にするためには、4 つの Capture 開始および停止スクリプトを実行する必要があります。

データ移動サービスの開始および停止

特定のビジネス指標モデルのデータ移動サービスは、関連付けられた Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスを開始および停止することで開始および停止されます。データ移動サービスのデプロイメント時に、開始スクリプトおよび停止スクリプトが作成され、データ移動サービスを開始および停止するためにこれらのスクリプトを使用できます。

コンポーネント・インスタンスを実行するマシンによってトポロジが決まります。一般的に、Capture コンポーネント・インスタンスは、状態データベースをホストするマシン (状態データベースからランタイム・データベースへのデータ移動サービスの場合)、およびランタイム・データベースをホストするマシン (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) で実行する必要があります。Apply コンポーネント・インスタンスは、ランタイム・データベースをホストするマシン (状態データベースからランタイム・データベースへのデータ移動サービスの場合)、およびヒストリー・データベースをホスト

するマシン (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) で実行する必要があります。この構成では、Apply コンポーネント・インスタンスはソース・データベースからデータを取り込みます。その結果、これらが、状態データベースをホストするマシン (状態データベースからランタイム・データベースへのデータ移動サービスの場合)、およびランタイム・データベースをホストするマシン (ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの場合) にある場合よりも良好なパフォーマンスが得られます。

以下の内容は、状態データベースからランタイム・データベースへのデータ移動サービス、およびランタイム・データベースからヒストリー・データベースへのデータ移動サービスの開始方法について説明しています。また、状態データベースからランタイム・データベースへのデータ移動サービス、およびランタイム・データベースからヒストリー・データベースへのデータ移動サービスの停止方法についても説明しています。

注: 状態データベースからランタイム・データベース、およびランタイム・データベースからヒストリー・データベースへのデータ移動サービスは、相互に独立しています。ただし、ランタイム・データベースからヒストリー・データベースへのサービスを開始する前に、状態データベースからランタイム・データベースへのサービスを開始することをお勧めします。特定のインスタンスでは、モニター・サーバーがビジネス指標モデルのエントリを処理した後、および状態データベースからランタイム・データベースへのデータ移動サービスにより、このモデルをサポートするランタイム・データベース・テーブルにデータが取り込まれた後に、ランタイム・データベースからヒストリー・データベースへのデータ移動サービスを開始することが望ましい場合があります。これにより、ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの間隔を待機するより高速にヒストリー・データベースに情報を取り込むことができます。

状態データベースからランタイム・データベースへのデータ移動サービスの開始:

デプロイメント・アーカイブ DS_State_setup および DS_Runtime_setup には、状態データベースからランタイム・データベースへのデータ移動サービス用の Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスを開始するために使用できる実行可能スクリプトが含まれます。ビジネス指標モデルへの変更の結果としてアーカイブが作成された場合は、新規 Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスの開始スクリプトのみがパッケージされています。

注: データ移動サービスの開始にスクリプトを統合することができます。スクリプトの統合について詳しくは、58 ページの『開始および停止スクリプトの統合』を参照してください。

ただし、統合が実行されていなくても、以下の指示を実行することができます。特定のビジネス指標モデルに対する状態データベースからランタイム・データベースへのデータ移動サービスを開始するには、以下を実行します。

1. 状態データベースのビジネス指標モデルに割り当てられているすべての Capture コンポーネント・インスタンスを確認します。

すべての Capture コンポーネント・インスタンスの開始スクリプトを既に統合している場合、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Capture コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Capture コンポーネント・インスタンスの開始スクリプトが自動的に生成されます。以降のスキーマ生成 (例えば、ビジネス指標モデルの更新後) では、新規 Capture コンポーネント・インスタンスの開始スクリプトのみが生成されます。関連するすべての開始スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。

- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `State_to_Runtime¥source` サブディレクトリーにナビゲートして、すべての `StartCapture_<number>` スクリプトを確認します。
 - c. この ビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
2. Capture コンポーネント・インスタンスを開始します。

確認された Capture コンポーネント・インスタンスは、状態データベースをホストするマシンで開始する必要があります。開始スクリプトが統合されている場合は、統合された開始スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した開始スクリプトを 1 つずつ実行する必要があります。

Capture ユーティリティーの初期化が失敗するため、スクリプトは同時に起動できません。ただし、開始スクリプトが起動される順序は関係ありません。許可の要件は以下のとおりです (Capture コンポーネント・インスタンスを開始するユーザー ID は以下の権限を保持している必要があります)。

- 状態データベースに対するデータベース管理者権限 (DBADM)。
 - 開始スクリプトで `CAPTURE_PATH` パラメーターによって参照されるディレクトリーに対する書き込みアクセス権限。
 - 開始スクリプトで `PWDFILE` オプション・パラメーターによって参照されるファイルに対する読み取りアクセス権限。
3. ランタイム・データベースのビジネス指標モデルに割り当てられているすべての Apply コンポーネント・インスタンスを確認します。

すべての Apply コンポーネント・インスタンスの開始スクリプトを既に統合している場合は、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Apply コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Apply コンポーネント・インスタンスの開始スクリプトが自動的に生成されます。以降のスキーマ生成 (例えば、ビジネス指標モデルの更新後) では、新規 Apply コンポーネント・インスタンスの開始スクリプトのみが生成されます。関連するすべての開始スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。

- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。

- b. `State_to_Runtime¥target` サブディレクトリーにナビゲートして、すべての `StartApply_<number>` スクリプトを確認します。
 - c. このビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
4. **Apply** コンポーネント・インスタンスを開始します。

確認された **Apply** コンポーネント・インスタンスは、ランタイム・データベースをホストするマシンで開始する必要があります。開始スクリプトが統合されている場合は、統合された開始スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した開始スクリプトを 1 つずつ実行する必要があります。**Apply** ユーティリティーの初期化が失敗するため、スクリプトは同時に起動できません。ただし、開始スクリプトが起動される順序は関係ありません。許可の要件は以下のとおりです (**Apply** コンポーネント・インスタンスを開始するユーザー ID は以下の権限を保持している必要があります)。

- 状態データベースの関連 **Capture** コンポーネント・インスタンス制御テーブルに対する `SELECT/INSERT/UPDATE/DELETE` 権限。
 - 状態データベースの関連 **Capture** コンポーネント・インスタンス作業テーブルに対する `SELECT` 権限。
 - ランタイム・データベースの関連レプリケーション・ステー징・テーブルに対する `SELECT/INSERT/UPDATE/DELETE` 権限。
 - ランタイム・データベースの **Apply** コンポーネント・インスタンス制御テーブルに対する `SELECT/INSERT/UPDATE/DELETE` 権限。
 - 開始スクリプトで `APPLY_PATH` パラメーターによって参照されるディレクトリーに対する書き込みアクセス権限。
 - 開始スクリプトで `PWDFILE` オプション・パラメーターによって参照されるファイルに対する読み取りアクセス権限。
5. 各 **Capture** コンポーネント・インスタンスおよび **Apply** コンポーネント・インスタンスが正常に開始されたことを確認します。

ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの開始:

デプロイメント・アーカイブ `DS_Runtime_setup` および `DS_Datamart_setup` には、ランタイム・データベースからヒストリー・データベースへのデータ移動サービス用の **Capture** コンポーネント・インスタンスおよび **Apply** コンポーネント・インスタンスを開始するために使用できる実行可能スクリプトが含まれます。ビジネス指標モデルへの変更の結果としてアーカイブが作成された場合は、新規 **Capture** コンポーネント・インスタンスおよび **Apply** コンポーネント・インスタンスの開始スクリプトのみがパッケージされています。

注: データ移動サービスの開始にスクリプトを統合することができます。スクリプトの統合について詳しくは、58 ページの『開始および停止スクリプトの統合』を参照してください。

ただし、統合が実行されていなくても、以下の指示を実行することができます。特定のビジネス指標モデルに対するランタイム・データベースからヒストリー・データベースへのデータ移動サービスを開始するには、以下を実行します。

1. ランタイム・データベースのビジネス指標モデルに割り当てられているすべての Capture コンポーネント・インスタンスを確認します。

すべての Capture コンポーネント・インスタンスの開始スクリプトを既に統合している場合、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Capture コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Capture コンポーネント・インスタンスの開始スクリプトが自動的に生成されます。以降のスキーマ生成 (例えば、ビジネス指標モデルの更新後) では、新規 Capture コンポーネント・インスタンスの開始スクリプトのみが生成されます。関連するすべての開始スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。

- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `Runtime_to_Historical¥source` サブディレクトリーにナビゲートして、すべての `StartCapture_<number>` スクリプトを確認します。
 - c. この ビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
2. Capture コンポーネント・インスタンスを開始します。

確認された Capture コンポーネント・インスタンスは、ランタイム・データベースをホストするマシンで開始する必要があります。開始スクリプトが統合されている場合は、統合された開始スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した開始スクリプトを 1 つずつ実行する必要があります。Capture ユーティリティーの初期化が失敗するため、スクリプトは同時に起動できません。ただし、開始スクリプトが起動される順序は関係ありません。許可の要件は以下のとおりです (Capture コンポーネント・インスタンスを開始するユーザー ID は以下の権限を保持している必要があります)。

- ランタイム・データベースに対するデータベース管理者権限 (DBADM)。
 - 開始スクリプトで `CAPTURE_PATH` パラメーターによって参照されるディレクトリーに対する書き込みアクセス権限。
 - 開始スクリプトで `PWDFILE` オプション・パラメーターによって参照されるファイルに対する読み取りアクセス権限。
3. ヒストリー・データベースのビジネス指標モデルに割り当てられたすべての Apply コンポーネント・インスタンスを確認します。

すべての Apply コンポーネント・インスタンスの開始スクリプトを既に統合している場合は、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Apply コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Apply コンポーネント・インスタンスの開始スクリプトが自動的に生成されます。以降のスキーマ生成 (例えば、ビジネス指標モデルの更新後) では、新規 Apply コンポーネント・インスタンスの開始スクリプトのみが生成されます。関

連するすべての開始スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。

- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `Runtime_to_Historical¥target` サブディレクトリーにナビゲートして、すべての `StartApply_<number>` スクリプトを確認します。
 - c. このビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
4. Apply インスタンスを開始します。

確認された Apply コンポーネント・インスタンスは、ヒストリー・データベースをホストするマシンで開始する必要があります。開始スクリプトが統合されている場合は、統合された開始スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した開始スクリプトを 1 つずつ実行する必要があります。Apply ユーティリティーの初期化が失敗するため、スクリプトは同時に起動できません。ただし、開始スクリプトが起動される順序は関係ありません。許可の要件は以下のとおりです (Apply コンポーネント・インスタンスを開始するユーザー ID は以下の権限を保持している必要があります)。

- ランタイム・データベースの関連 Capture コンポーネント・インスタンス制御テーブルに対する SELECT/INSERT/UPDATE/DELETE 権限。
 - ランタイム・データベースの関連 Capture コンポーネント・インスタンス作業テーブルに対する SELECT 権限。
 - ヒストリー・データベースの関連レプリケーション・ステージング・テーブルに対する SELECT/INSERT/UPDATE/DELETE 権限。
 - ヒストリー・データベースの Apply コンポーネント・インスタンス制御テーブルに対する SELECT/INSERT/UPDATE/DELETE 権限。
 - 開始スクリプトで `APPLY_PATH` パラメーターによって参照されるディレクトリーに対する書き込みアクセス権限。
 - 開始スクリプトで `PWDFILE` オプション・パラメーターによって参照されるファイルに対する読み取りアクセス権限。
5. 各 Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスが正常に開始されたことを確認します。

状態データベースからランタイム・データベースへのデータ移動サービスの停止:

状態データベースからランタイム・データベースへのデータ移動サービスの停止プロセスは、その開始プロセスと非常に類似しています。デプロイメント・アーカイブ `DS_State_setup` および `DS_Runtime_setup` には、状態データベースからランタイム・データベースへのデータ移動サービス用の Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスを停止するために使用できる実行可能スクリプトが含まれます。

ビジネス指標モデルへの変更の結果としてアーカイブが作成された場合は、新規 Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスの停止スクリプトのみがパッケージされています。

注: データ移動サービスを停止する前に、スクリプトを統合することをお勧めします。レプリケーション・スクリプトの統合について詳しくは、58 ページの『開始および停止スクリプトの統合』を参照してください。

ただし、統合が実行されていなくても、以下の指示を実行することができます。

特定のビジネス指標モデルに対する状態データベースからランタイム・データベースへのデータ移動サービスを停止するには、以下を実行します。

1. 状態データベースのビジネス指標モデルに割り当てられているすべての Capture コンポーネント・インスタンスを確認します。すべての Capture コンポーネント・インスタンスの停止スクリプトを既に統合している場合、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Capture コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Capture コンポーネント・インスタンスの停止スクリプトが自動的に生成されます。以降のスキーマ生成 (例えば、ビジネス指標モデルの更新後) では、新規 Capture コンポーネント・インスタンスの停止スクリプトのみが生成されます。関連するすべての停止スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。
 - a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `State_to_Runtime¥source` サブディレクトリーにナビゲートして、すべての `StopCapture_<number>` スクリプトを確認します。
 - c. このビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
 2. Capture コンポーネント・インスタンスを停止します。確認された Capture コンポーネント・インスタンスは、状態データベースをホストするマシンで停止する必要があります。停止スクリプトが統合されている場合は、統合された停止スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した停止スクリプトを 1 つずつ実行する必要があります。停止スクリプトが起動される順序は関係ありません。
- 注: 停止スクリプトは非同期的および不定期に作動し、停止コマンドの発行時と Capture コンポーネントの停止時との間に遅延が発生する場合があります。これは、Capture コンポーネント・インスタンスが、停止する前にトランザクションの終了処理を実行しているためです。
3. ランタイム・データベースのビジネス指標モデルに割り当てられたすべての Apply コンポーネント・インスタンスを確認します。すべての Apply コンポーネント・インスタンスの停止スクリプトを既に統合している場合、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Apply コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Apply コンポーネント・インスタンスの停止スクリプトが自動的に生成されます。以降のスキーマ生成 (例えばビジネス指標モデルの更新後) では、新規 Apply コンポーネント・インスタンスの停止スクリプトのみが生成されます。関連するすべての停止スクリプト

を確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。

- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `State_to_Runtime¥target` サブディレクトリーにナビゲートして、すべての `StopApply_<number>` スクリプトを確認します。
 - c. このビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
4. Apply コンポーネント・インスタンスを停止します。

確認された Apply コンポーネント・インスタンスは、ランタイム・データベースをホストするマシンで停止する必要があります。停止スクリプトが統合されている場合は、統合された停止スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した停止スクリプトを 1 つずつ実行する必要があります。停止スクリプトは任意の順序で起動できます。

注: 停止スクリプトは非同期のおよび不定期に作動し、停止コマンドの発行時と Apply コンポーネントの停止時との間に遅延が発生する場合があります。これは、Apply コンポーネント・インスタンスが、停止する前に 1 つ以上のトランザクションの終了処理を実行しているためです。

ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの停止:

ランタイム・データベースからヒストリー・データベースへのデータ移動サービスの停止プロセスは、その開始プロセスと非常に類似しています。デプロイメント・アーカイブ `DS_Runtime_setup` および `DS_Datamart_setup` には、ランタイム・データベースからヒストリー・データベースへのデータ移動サービス用の Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスを停止するために使用できる実行可能スクリプトが含まれます。

ビジネス指標モデルへの変更の結果としてアーカイブが作成された場合は、新規 Capture コンポーネント・インスタンスおよび Apply コンポーネント・インスタンスの停止スクリプトのみがパッケージされています。

注: データ移動サービスを停止する前に、スクリプトを統合することをお勧めします。レプリケーション・スクリプトの統合について詳しくは、58 ページの『開始および停止スクリプトの統合』を参照してください。

ただし、統合が実行されていなくても、以下の指示を実行することができます。

特定のビジネス指標モデルに対するランタイム・データベースからヒストリー・データベースへのデータ移動サービスを停止するには、以下を実行します。

1. ランタイム・データベースのビジネス指標モデルに割り当てられたすべての Capture コンポーネント・インスタンスを確認します。すべての Capture コンポーネント・インスタンスの停止スクリプトを既に統合している場合、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない(統合しない)場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Capture コンポーネント・インスタンスを確認する必要があります。ビジ

ネス指標モデルに対するスキーマ生成の初回実行時に、Capture コンポーネント・インスタンスの停止スクリプトが自動的に生成されます。以降のスキーマ生成 (例えば、ビジネス指標モデルの更新後) では、新規 Capture コンポーネント・インスタンスの停止スクリプトのみが生成されます。関連するすべての停止スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。

- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `Runtime_to_Historical¥source` サブディレクトリーにナビゲートして、すべての `StopCapture_<number>` スクリプトを確認します。
 - c. このビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
2. Capture コンポーネント・インスタンスを停止します。確認された Capture コンポーネント・インスタンスは、ランタイム・データベースをホストするマシンで停止する必要があります。停止スクリプトが統合されている場合は、統合された停止スクリプトを起動します。統合が実行されていない場合は、前の手順で確認した停止スクリプトを 1 つずつ実行する必要があります。停止スクリプトが起動される順序は関係ありません。

注: 停止スクリプトは非同期のおよび不定期に作動し、停止コマンドの発行時と Capture コンポーネントの停止時との間に遅延が発生する場合があります。これは、Capture コンポーネント・インスタンスが、停止する前にトランザクションの終了処理を実行しているためです。

3. ヒストリー・データベースのビジネス指標モデルに割り当てられたすべての Apply コンポーネント・インスタンスを確認します。すべての Apply コンポーネント・インスタンスの停止スクリプトを既に統合している場合、必要な作業はありません。次の手順に進んでください。まだスクリプトを統合していない (統合しない) 場合は、このビジネス指標モデルに対してこれまでに作成されたすべての Apply コンポーネント・インスタンスを確認する必要があります。ビジネス指標モデルに対するスキーマ生成の初回実行時に、Apply コンポーネント・インスタンスの停止スクリプトが自動的に生成されます。以降のスキーマ生成 (例えばビジネス指標モデルの更新後) では、新規 Apply コンポーネント・インスタンスの停止スクリプトのみが生成されます。関連するすべての停止スクリプトを確認するには、このビジネス指標モデルに対して実行した各デプロイメントについて、以下の手順を繰り返す必要があります。
- a. このモデルのデプロイメントを実行したディレクトリーにナビゲートします。
 - b. `Runtime_to_Historical¥target` サブディレクトリーにナビゲートして、すべての `StopApply_<number>` スクリプトを確認します。
 - c. このビジネス指標モデルの各デプロイメントについて、上記の手順を繰り返します。
4. Apply コンポーネント・インスタンスを停止します。

確認された Apply コンポーネント・インスタンスは、ランタイム・データベースをホストするマシンで停止する必要があります。停止スクリプトが統合されている場合は、統合された停止スクリプトを起動します。統合が実行されていない

場合は、前の手順で確認した停止スクリプトを 1 つずつ実行する必要があります。停止スクリプトは任意の順序で起動できます。

注: 停止スクリプトは非同期的および不定期に作動し、停止コマンドの発行時と Apply コンポーネントの停止時との間に遅延が発生する場合があります。これは、Apply コンポーネント・インスタンスが、停止する前に 1 つ以上のトランザクションの終了処理を実行しているためです。

Cube Views のデータベース・スキーマのデプロイ

Schema Generator は、Cube Views のメタデータを XML ファイル形式で生成します。メタデータは、ビジネス指標モデルに対応する DB2 Cube Views の定義を表します。Cube Views の定義は、Windows プラットフォームと AIX プラットフォームの両方にデプロイされます。

Windows プラットフォームでの Cube Views のデータベース・スキーマのデプロイ:

Cube Views のメタデータは Schema Generator の出力フォルダーに保管されます。この出力フォルダーは、WebSphere Business Monitor 管理コンソールを使用して、ユーザーによって定義されます。

Cube Views のメタデータ・ファイルをデプロイするには、以下の手順をすべて実行します。

1. DB2 OLAP Center を開始します。「**DB2 データベース接続 (DB2 database connection)**」ダイアログ・ボックスが表示されます。
2. 「**DB2 データベース接続 (DB2 database connection)**」ダイアログ・ボックスで、以下を実行します。
 - a. 「**データベース名 (Database name)**」フィールドに、ヒストリー・データベースの名前を入力します。
 - b. 「**ユーザー名**」フィールドに、このデータベースに対する管理権限を持つユーザーのユーザー ID を入力します。
 - c. 「**パスワード**」フィールドに、このデータベースに対する管理権限を持つユーザーのパスワードを入力します。
 - d. 「**OK**」をクリックします。
 - e. 初めてそのデータベースに接続する場合、データベースを Cube Views 用に構成する必要があることをメッセージが知らせます。メッセージに対して「はい」をクリックして、初期化および構成を開始します。
3. OLAP Center のウィンドウで、次のように Cube Views のメタデータ・ファイルをインポートします。
 - a. メニューから、「**OLAP Center**」 → 「**インポート**」を選択します。「インポート」ウィザードが開始します。
 - b. Schema Generator の出力フォルダーに保管されている Cube Views XML ファイルを選択します。ファイルの名前は *model_cv.xml* です。
 - c. 「**完了**」をクリックします。インポート・プロセスが開始します。
4. インポートの完了後、「**インポート・ウィザード (Import Wizard)**」ウィンドウの「**インポート・オプション (Import Options)**」ページで「完了」をクリックします。

AIX プラットフォームでの Cube Views のデータベース・スキーマのデプロイ:

Cube Views のメタデータは Schema Generator の出力フォルダーに保管されます。この出力フォルダーは、WebSphere Business Monitor 管理コンソールを使用して、ユーザーによって定義されます。

Cube Views のメタデータ・ファイルをデプロイするには、以下の手順をすべて実行します。

1. DB2 コマンド・ウィンドウのエディターをオープンします。
2. コマンド **db2 connect to HISTORICAL_database_name** を実行することにより、データベース・インスタンスのユーザー (例: db2inst1) を使用してヒストリー・データベースに接続します。
3. ディレクトリを **<DB2_INST_HOME>/sqllib/misc** に変更し、コマンド **db2 -tvf db2mdapi.sql** を実行します。
4. コマンド **db2mdapiclient -d HISTORY -i <GENERATION_DIR>/schemagen/import_model.xml -m <GENERATION_DIR>/schemagen/model_cv.xml -u <userid> -p <pw> -o <GENERATION_DIR>/schemagen/myoutput.xml** を実行します。

ここで、

- -d はヒストリー・データベース名です。
- -i は Schema Generator によって生成される import_model.xml ファイルです。
- -u はユーザー ID です。
- -p はパスワードです。
- -o は、DB2 出力情報が保管される出力操作ファイル名です。
- -m は、DB2 への入力メタデータ・コマンドまたは命令です。Schema Generator は、マルチディメンション・メタデータとして使用される model_cv.xml ファイルを作成します。
- <GENERATION_DIR> は、Schema Generator が生成する成果物を保管する出力ディレクトリを指します。

例:

```
su - db2inst1
db2 connect to HISTORY
cd /home/db2inst1/sqllib/misc
db2 -tvf db2mdapi.sql
db2mdapiclient -d HISTORY
-i /opt/IBM/WebSphere/Monitor/generation/schemagen/import_model.xml
-m /opt/IBM/WebSphere/Monitor/generation/schemagen/model_cv.xml
-u db2inst1 -p monPa55w -o /tmp/import_output.xml
```

ABX Cube の手動作成:

IBM DB2 ALPHABLOX サーバーがインストールされているマシンに手動で ABX Cube を作成します。これらのキューブは、WebSphere Business Monitor ダッシュボードによって使用されます。

Cube Views の定義をデプロイした後、ダッシュボードを使用する前に以下の手順を実行します。

1. Web ブラウザーで `http://<hostname>:9081/AlphaBlox/home/Admin` にアクセスし、IBM DB2 ALPHABLOX 管理コンソールにログインします。
2. 「管理」タブを選択します。
3. 「キューブ (Cubes)」をクリックします。
4. キューブを作成するには、「作成 (Create)」をクリックします。
 - a. 「関連データ・ソース (Relational Data Source)」リストから、インストール時に定義された適切なヒストリー・データベースを選択します。
 - b. 「DB2 AlphaBlox キューブ名 (DB2 AlphaBlox Cube Name)」の横の「使用可能 (Enabled)」チェック・ボックスをチェックします。
 - c. 「DB2 Cube Views の設定を使用可能にする (Enable DB2 Cube Views Settings)」チェック・ボックスをチェックします。数秒間待つとフィールドが表示されます。
5. 「キューブ・モデル (Cube Model)」の下に定義されている各キューブに対して、キューブを作成する必要があります。
 - a. 「キューブ・モデル (Cube Model)」リストから、キューブ・モデルを選択します。
 - b. 「キューブ (Cube)」リストから、キューブを選択します。キューブ・モデルあたりキューブは 1 つのみです。
 - c. 「DB2 AlphaBlox キューブ名 (DB2 AlphaBlox Cube Name)」フィールドに、キューブ名を入力します。この名前は、「キューブ (Cube)」リストに表示されているとおりにする必要があります。例: CISS.NOOP。スキーマ名の CISS を含めないようにします。
 - d. 「ビジネス名を使用する (Use Business Names)」オプションを選択します。
 - e. 「キューブ定義をインポートする (Import Cube Definition)」をクリックし、処理されるのを待機します。
6. 「OK」をクリックして、キューブを保管します。
7. 存在する各キューブに対して、手順 5 (キューブの作成) を繰り返します。

ディメンション・テーブルへの手動によるデータ取り込み

ディメンション・データとして使用される既存データがある可能性があります (例えば、カスタマー・ディメンションに取り込む必要があるカスタマー情報のデータベース)。ヒストリー・データベースを使用して、ディメンション・テーブルにこのデータを取り込むことができます。

テーブルにデータを取り込む場合に注意する点があります。

- WebSphere Business Modeler にディメンションを作成する場合は、ディメンションに既存のデータを取り込めるよう十分に注意する必要があります。WebSphere Business Modeler で定義されているとおり、ディメンションに適切なデータ型を使用した適切なメトリックが含まれ、Schema Generator によって作成されたディメンション・テーブルに既存データを保管できることを確認します。

- データを手動で挿入する場合は、SK_<> 列に負の値を使用してください。これはテーブルの代理キーです。正の代理キー値は、これらのテーブルへのデータの取り込み時にデータ・サービスによって使用されます。衝突を回避するために、負の値を使用する必要があります。
- ディメンション・テーブルへのデータの挿入時は、NULL 値に設定された列がないことを確認してください。特定の列に挿入する意味のある値がない場合は、意味のあるデフォルト値を選択し、それを使用する必要があります。このテーブルに NULL は挿入しないでください。ただし、空ストリング ("") はストリング型に使用できます。
- 新規プロセス・インスタンスを挿入したディメンション・データにマップする場合、プロセス・インスタンスに一致するものが既存データにない場合があります (例えば、現在ディメンション・テーブルにリストされていない新規カスタマーに関連付けられたプロセス)。この場合、このデータのセット用にテーブルに新しい行が作成されます。これにより、テーブルにはユーザーが入力したデータと、該当するその他のデータが含まれることになります。
- 新規データが入力されると、ディメンションの非キー属性が更新されます。例えば、キー・メトリックが「CustomerName」、非キー・メトリックが「CreditLimit」のカスタマー・ディメンションがあると想定します。最初、このテーブルには既存のカスタマー・データから取り込まれた行 ['Widgets, Inc',50000] が含まれています。75,000 の 'Widgets, Inc' の CreditLimit を含む新規イベントが処理された場合、カスタマー・ディメンション・テーブル行は ['Widgets, Inc', 75000] に更新されます。この更新は、キー・メトリックが既存行と一致し、非キー・メトリックが一致しない場合にのみ行われます。このような場合、新規データを反映するために非キー値が更新されます。

ユーザーが手動でデータを取り込むディメンションに対応させるディメンション・テーブルを設定、およびさまざまなディメンション属性に対応させるテーブル内の列を設定するには、*datamartMapping.txt* テキスト・ファイルを使用します。このファイルは、Schema Generator の出力ディレクトリーにあります (Schema Generator の実行後)。

ヒストリー・データベースのスキーマ

データベース・スキーマは、データベース・テーブル、およびそれらの間の関係を記述します。データベース・スキーマを使用して、データベース・サイズを計画できます。

ヒストリー・データベースのスキーマの情報は、インポートされた ビジネス指標モデルとデータベース・テーブルとの間のマッピングを理解するために役立ちます。ダッシュボードは、マルチディメンション分析およびレポート生成のために、ヒストリー・データベースを使用します。

注:

- リポジトリ・データベース、状態データベース、およびランタイム・データベースは内部的な使用に限られ、予告なしに変更される場合があります。
- 状態データベース、ランタイム・データベース、またはリポジトリ・データベースに直接アクセスするための、お客様が記述したカスタム・コードは、IBM ではサポートされません。

- ヒストリー・データベースのスキーマを使用して独自のダッシュボードを作成することはできません。

ヒストリー・データベースには、最初に 1995 年から 2009 年の範囲の日付/時刻データが取り込まれます。(プロセス開始/終了時刻またはその他のメトリック・データとして) この 日付の範囲外の日付/時刻の記録が予期される場合は、以下の SQL スクリプトを使用して、ヒストリー・データベースの DIM_TIME テーブルにさらに日付を追加する必要があります。

```
insert
into <your WBI schema name>.dim_time( surrogate_key, year, month,
day) with WBITIME (skey, ldate) as (select surrogate_key+1 as skey, COALESCE(
DATE(SUBSTR(DIGITS(YEAR),7,4) || '-' || SUBSTR(DIGITS(MONTH),4,2)
|| '-' || SUBSTR(DIGITS(DAY), 4,2)) + 1 DAYS, DATE('YYYY-MM-DD of the
first day you'd want to start from, in case the DIM_TIME table is empty.')
)as ldate from sysibm.sysdummy1, <your WBI schema name>.dim_time
where DATE( SUBSTR(DIGITS(YEAR) ,7,4) || '-' || SUBSTR(DIGITS(MONTH),4,2)
|| '-' || SUBSTR(DIGITS(DAY) ,4,2) ) = ( SELECT MAX(
DATE(SUBSTR(DIGITS(YEAR),7,4) || '-' || SUBSTR(DIGITS(MONTH),4,2)
|| '-' || SUBSTR(DIGITS(DAY), 4,2))) FROM <your WBI schema
name>.DIM_TIME ) UNION ALL SELECT parent.skey+1, ldate + 1 DAYS
from WBITIME parent where YEAR(ldate + 1 days) < where YEAR(ldate +
1 days) < <YYYY 4 Digit YEAR FOR WHICH YOU DON't WANT DATA to
end in>
) select a.skey, year(a.ldate), month(a.ldate), day(a.ldate) from WBITIME
a WHERE a.ldate >= DATE('YYYY-MM-DD: The start of the range that should be
inserted.') AND a.ldate <= DATE('YYYY-MM-DD: The end of the range that
should be inserted.')
```

注: このスクリプトには、DIM_TIME に挿入するデータの開始日付 および終了日付指定して更新すべき場所が 4 箇所 あります。また、WBI スキーマ名 (通常は「WBI」) を指定する必要がある 場所が 3 箇所あります。

データベース・サービス

この参照情報は、データベース・サービスを扱うときに役立ちます。

ヒストリー・データベースのスキーマ

ヒストリー・データベース・テーブルは、2 つのタイプに分けられます。1 つは WebSphere Business Monitor のインストール時に作成される静的なテーブル、もう 1 つはビジネス指標モデルをインポートするたびに作成される動的なテーブルです。

両方のタイプのヒストリー・データベース・テーブルの説明と、対応する各列のビジネス指標モデルへのマッピングについては、以下のテーブルにリストされています。

注:

- NULL 可能: この列が NULL 値を受け入れることができるかどうかを示します。
- 記述子: 列とビジネス指標モデルの定義間のマッピングを記述します。各列に記述子を含める必要はありません。

静的データベース・テーブル

DIM_TIME

時間ディメンション・テーブル。

列名	列タイプ	列の説明	NULL 可能
SURROGATE_KEY	INTEGER	基本キーです。	N
DAY	SMALLINT	日付を示します。	N
MONTH	SMALLINT	月を示します。	N
YEAR	INTEGER	年を示します。	N

動的データベース・テーブル

ヒストリー・データベースは、中央のファクト・テーブルを複数のディメンション「リーフ」テーブルで囲むスター・スキーマの構造を実装します。ファクト・テーブルは、状態およびランタイム・データベースのコンテキスト・テーブルと似ています。コンテキストと、コンテキストの対応するアクティビティーのそれぞれにスターが 1 つずつあります。例えば、1 つのコンテキスト・インスタンス・テーブルと、状態およびランタイム・データベース内のコンテキストごとに 1 つのアクティビティー・インスタンス・テーブルが存在できます。

コンテキスト・ファクト・テーブル

命名規則: FCT_<コンテキストにおけるマシン生成名>

常に存在する列は、次のとおりです。

列名	列タイプ	列の説明	NULL 可能
MCI_MCIID	DECIMAL(19,0)	アクティビティ・インスタンスの固有 ID で、テーブルの基本キーでもあります。	N
PARENT_MCIID	DECIMAL(19,0)	親プロセス・インスタンスの固有 ID (存在する場合)。	Y
SK_<ディメンションのマシン生成名>	INTEGER	ディメンション・テーブルを指す外部キー。FK の関係が定義されます。コンテキスト内に存在する各ディメンションごとに、これらの列のうちの 1 つが定義されます。	Y
GMT_<メトリックのマシン生成名>	TIMESTAMP	タイム・スタンプ・メトリックのデータ・タイプである GMT 時間値を保管するのに使用するタイム・スタンプ値 (この列は、ディメンションとして時間メトリックがマークされている場合にのみ作成されます。時間メトリックがディメンションとしてマークされると、これらは日、月、年の単位でのみ保管されます。そのため、この列を参照すると、これらのメトリックの正確な時間値を確認できます。)	Y

次の 3 つの列タイプは、メトリックがディメンションではなく、ファクトとしてマークされたときに使用されます。

メトリックがファクトとしてマークされたときに使用される列タイプ

列名	列タイプ	列の説明	NULL 可能
M_<マシン生成名>	データ・タイプ	メトリックまたは主要な定義値を示す場合に使用されます。	Y
C_<マシン生成名>	BIGINT	カウンターを示す場合に使用されます。	Y
TI_<マシン生成名>	BIGINT	タイマーに累積された時間を示す場合に使用されます。(ヒストリー・データベースでは、タイマーは 1 つの列で表示されます。状態データベースでは、複数の列を使用して表示されます)。	Y

ディメンション・テーブル

各コンテキストには、コンテキストが定義するディメンション数に応じて、0 個以上のディメンション・テーブルが定義されます。通常、少なくとも時間ディメンションは存在します。

命名規則: DIM_<ディメンションのマシン生成名>

常に定義される列は、次のとおりです。

列名	列タイプ	列の説明	NULL 可能
SURROGATE_KEY	INTEGER	このディメンション行に対してマシンが生成した基本キー値。PK が定義されます。	N

これらは、定義に基づいた列です。ディメンション・テーブルには、このディメンションの一部として定義された各メトリックの列が含まれます。

定義に基づいた列

列名	列タイプ	列の説明	NULL 可能
M_<マシン生成名>	データ・タイプ は、ビジネス指標 モデルで定義され ているデータ・タ イプによって異な ります。	メトリックまたは主要な定義値を 示す場合に使用されます。	Y
C_<マシン生成名>	BIGINT	カウンターを示す場合に使用され ます。	Y
T1_<マシン生成名>	BIGINT	タイマーに累積された時間を示す 場合に使用されます (ヒストリー・ データベースでは、タイマーは 1 つの列で表示されます。状態デー タベースでは、複数の列を使用し て表示されます)。	Y

アクティビティのスター・スキーマも同じ規則に従いますが、テーブル名がそれぞれ AFC_ および ADM_ である点が異なります。

データ移動サービス制御テーブル

このセクションでは、データ移動サービス制御テーブルの構造について説明します。状態データベース、ランタイム・データベース、およびヒストリー・データベースには、それぞれ 2 つの制御テーブルが含まれます。この制御テーブルを操作すると、ローカル・データ制御の移動サービス・コンポーネントの動作を構成できます。制御テーブルは静的テーブルです。

RMCONTROL

ETL コンポーネント・インスタンスの動作固有の構成設定が含まれます。このテーブルにデータが取り込まれたり使用されたりするのは、ランタイム・データベースとヒストリー・データベースの場合のみです。これは、状態データベースには ETL コンポーネントが必要ないためです。このテーブルのそれぞれの行は、データを取り込む必要のある 1 つのターゲット・テーブルに対応しています。特定の行に対する列の値を変更すると、そのターゲット・テーブルにデータを取り込むように割り当てられた ETL コンポーネント・インスタンスのみが影響を受けます。

列名	列タイプ	列の説明	NULL 可能
TARGETTABLE	CHARACTER	この項目により制御されるストアード・プロシージャによってデータが取り込まれるターゲット・テーブルの完全修飾テーブル名。	N
COMMITINTERVAL	NUMERIC	ターゲット・テーブルへの行の挿入にカーソルが使用されるとき、ストアード・プロシージャが使用するコミット間隔。	Y
LOGLEVEL	NUMERIC	ストアード・プロシージャが WBIRMADM.RMLOG テーブルに書き込む情報の量を決定するロギング・レベル。有効な値は 0 と 1 です。0 は最小ロギング、1 は最大ロギングを表します。	Y
LASTSEQUENCE	CHARACTER	ステージング・テーブルから ETL ストアード・プロシージャによって処理された最新の SEQUENCE 値。この列は、実行時にストアード・プロシージャによって更新されます。	N
LASTUPDATED	TIMESTAMP	スケジュールされた呼び出しが行われた最終時刻。この列はストアード・プロシージャによって制御され、スケジューリングの目的のみに使用されます。	Y
NEXTSTARTTIME	TIMESTAMP	次に ETL 呼び出しが行われる時刻。	Y
ETLSCHEDMETHOD	NUMERIC	使用されるスケジューリング方式。有効な値は 0 のみです。	Y
ETL_0_MINUTES	NUMERIC	スケジュールされた ETL の実行から次の実行までに経過しなければならない分数。	Y
TGT_RM_SPETL_NAME	CHARACTER	TARGETTABLE へのデータの取り込みを行うストアード・プロシージャの完全修飾名。	Y

以下の列の値は変更しないでください。変更すると予期しない動作が発生します。

- TARGETTABLE
- LASTSEQUENCE
- LASTUPDATED
- ETLSCHEDMETHOD
- TGT_RM_SPETL_NAME

以下の列への変更は、次に ETL コンポーネント・インスタンスが呼び出されるときにコミットされます。

- COMMITINTERVAL

- NEXTSTARTTIME
- LOGLEVEL
- ETL_0_MINUTES

RMPRUNECTRL

Life Cycle コンポーネント・インスタンスの動作固有の構成設定が含まれます。このテーブルは、状態データベース、ランタイム・データベース、およびヒストリー・データベースにおいてデータが取り込まれ、使用されます。このテーブルのそれぞれの行は、整理する必要がある 1 つの (ソースまたは作業) テーブル <TABLE_NAME> に対応しています。特定の行に対する列の値を変更すると、テーブル <TABLE_NAME> を整理するように割り当てられた Life Cycle コンポーネント・インスタンスのみが影響を受けます。

列名	列タイプ	列の説明	NULL 可能
TABLE_NAME	CHARACTER	整理されるテーブルの完全修飾名。	N
LAST_PRUNED	TIMESTAMP	このテーブルで最後に整理操作が行われた時刻。	Y
LOGLEVEL	NUMERIC	WBIRMAADM.RMLOG テーブルに書き込まれる情報の量を決定するロギング・レベル。有効な値は 0 と 1 です。0 は最小ロギング、1 は最大ロギングを表します。	N
PRUNE_ENABLED	NUMERIC	整理操作を行う必要があるかどうかを決定するフラグ。0 はいいいえ、1 はいを表します。	N
PRUNE_INTERVAL	NUMERIC	整理操作間の最小時間間隔 (分単位)。	N
RETENTION_IN_MINUTES	NUMERIC	適格な行を整理できるまでの時間 (分単位)。	N
ROWS_PRUNED	NUMERIC	最後の整理操作中に整理された行数。	N

以下の列の値は変更しないでください。変更すると予期しない動作が発生します。

- LAST_PRUNED
- ROWS_PRUNED
- TABLE_NAME

以下の列への変更は、次に ETL コンポーネント・インスタンスが呼び出されるときにコミットされます。

- LOGLEVEL
- PRUNE_ENABLED
- PRUNE_INTERVAL
- RETENTION_IN_MINUTES

データ移動サービス・メタデータおよびロギング・テーブル

このセクションでは、WebSphere Business Monitor データベースのロギング・テーブル構造に関する参照情報を提供します。ロギング・テーブルは、静的なテーブルです。

RM METADATA

データ移動サービスを提供するときに使用するコンポーネント・インスタンスの数は、サービス提供先のビジネス指標モデルごとに異なります。状態データベース、ランタイム・データベース、およびヒストリー・データベースのそれぞれには、割り当て済みのコンポーネント・インスタンス名をビジネス指標モデルごとにリストしたテーブルと、その他の有用な内部情報が含まれます。このテーブルは、デプロイメント・フェーズでコンポーネント・インスタンスを作成および構成するたびに更新されます。手動でテーブルの内容を変更しないでください。

列名	列タイプ	列の説明
ID	NUMERIC	未使用
OM_NAME	CHARACTER	関連するビジネス指標モデル・プロジェクトの名前。
OM_ID	NUMERIC	未使用
MC_NAME	CHARACTER	関連するビジネス指標グループの名前。
MC_ID	NUMERIC	未使用
TGT_TAB_NAME	CHARACTER	ETL ストアード・プロシージャによりデータが取り込まれるターゲット・テーブルの完全修飾名。
TGT_RM_APP_SVR_NAME	CHARACTER	レプリケーション適用操作の実行を担当するサーバーの名前。
TGT_RM_APP_SS_NAME	CHARACTER	DB2 SQL レプリケーションにおいて、適用サーバーが管理するグループ (使用可能な場合)。これは、サブスクリプション・セットです。
TGT_RM_APP_STG_TAB_NAME	CHARACTER	レプリケーション適用プログラムがターゲットとして使用するステージング・テーブルの完全修飾名。 注: 拡張子が <i>_BKUP</i> および <i>_M</i> の別のテーブルが 2 つあります。これらのテーブルはシステム内に存在し、この行に関連付けられています。
TGT_RM_APP_ERR_TAB_NAME	CHARACTER	ETL によって以降の処理が必要なステージング・テーブルの行へのポインターを保管するテーブルの完全修飾名。
TGT_RM_APP_PRUNE_SP_NAME	CHARACTER	ターゲット・システム上で適用ステージング・テーブルの整理を担当するストアード・プロシージャの完全修飾名。
TGT_RM_APP_TMP_TAB_NAME	CHARACTER	ターゲット・テーブルに読み込む行を決定するために ETL プログラムが使用する一時テーブルの完全修飾名。
TGT_RM_SPETL_NAME	CHARACTER	ステージング・テーブルの項目からターゲット・テーブルへのデータの取り込みを担当する ETL ストアード・プロシージャの完全修飾名。
SRC_TAB_NAME	CHARACTER	ステージング・テーブルに複製されるソース・テーブルの完全修飾名。
SRC_RM_CAP_SVR_NAME	CHARACTER	レプリケーション・キャプチャー操作の実行を担当するサーバーの名前。
SRC_RM_CAP_STG_TAB_NAME	CHARACTER	ソース・テーブルへの変更を保管するために Capture サーバーが使用するテーブルの完全修飾名。
SRC_RM_PRUNE_TRG_NAME	CHARACTER	Capture サーバーの整理サイクル中にソース・テーブルから選択した行を除去するトリガーの完全修飾名。選択した行に、完了した操作を示す行が含まれている場合があります。
SERVICE_NAME	CHARACTER	これらの成果物が属するサービスの識別に使用するラベル (State_to_Runtime や Runtime_to_Historical など)。

以下の簡単なビューの例を使用します。

OM_NAME	SRC_TAB_NAME	SRC_RM_CAP_SV...	SRC_RM_CA...	TGT_RM_AP...	TGT_RM_AP...	TGT_TAB_NAME	SERVICE_NAME
STEW_S	wbi.CTX_TQ4MUF...	CAPTURE_1	CAP.CD_2	APPLY_4	APP.CCD_6	wbi.CTR_TQ4MUF...	State to Runtime
STEW_S	wbi.AI_BVSOYAP...	CAPTURE_1	CAP.CD_3	APPLY_4	APP.CCD_7	wbi.AIR_BVSOYA...	State to Runtime

状態データベースのソース WBI.CTXTQ4MUF が Capture コンポーネント・インスタンス CAPTURE_1 によってモニターされているかどうかを判別するのは簡単です。ソース・テーブルに対する変更はすべて、作業テーブル CAP.CD_2 に記録され、Apply コンポーネント・インスタンス APPLY_4 によって作業テーブル APP.CCD_6 に適用されます。このテーブルは、ランタイム・データベースのターゲット・テーブル WBI.CTXTQ4MUF にデータを取り込むために、ETL コンポーネント・インスタンスによって使用されます。

RMLOG

ランタイム・データベースと履歴データベースにはロギング・テーブルがそれぞれ含まれており、これを使用することにより、統計、進行、デバッグ、またはエラーの各情報を取得できます。すべての ETL コンポーネントとターゲットの Life Cycle コンポーネントは、このテーブルにメッセージを書き込みますが、このテーブルから読み取りを行うことはありません。一部のメッセージは、ロギング・レベルを最小に設定することで抑止される場合があります。

列名	列タイプ	列の説明
ENTRYSTMP	TIMESTAMP(10)	このログ・テーブル内の特定の項目のタイム・スタンプ。
ID	NUMERIC	同じインスタンスにある複数の行を互いに関連付けるための ID。この ID は、SEQUENCE WBIRMADM.RMSPTRIGID に由来します。
ROWS_INSERTED	NUMERIC	このインスタンス中に挿入された行数を示すインジケータ。
ROWS_UPDATED	NUMERIC	このインスタンス中に更新された行数を示すインジケータ。
ROWS_DELETED	NUMERIC	このインスタンス中に削除された行数を示すインジケータ。
ROWS_INERROR	NUMERIC	このインスタンス中にリカバリー可能エラーの原因としてマークされた行数を示すインジケータ。
NAME	CHARACTER	このテーブル内の項目が追加される原因になったストアード・プロシージャ、トリガー、またはプロセスの完全修飾名。
OPERATION	CHARACTER	項目の作成時に実行された操作を識別するラベル。
RESULT	CHARACTER	発生した操作に関する詳細情報を提供する列。
ISTRACEENTRY	NUMERIC	この項目で、LOGLEVEL (WBIRMADM.RMCONTROL 内) に 1 を設定する必要があるかどうかを示す列。 0: このログ項目は、トレース項目ではありません。 1: このログ項目は、トレース項目です (これは抑止されている場合があります。 WBI.RMCONTROL テーブルを参照してください)。

このテーブルの各行は、<ENTRYSTMP> のコンポーネント・インスタンス <NAME> が発行したメッセージに対応しています。同じ <ID> と <NAME> を持つ行は、<NAME> の同一の呼び出しで生成されたメッセージであることを示します。次の例には、ETL コンポーネント・インスタンス WBIRMADM.WBIRMSP_10 と WBIRMADM.WBIRMSP_14 およびターゲットの Life Cycle コンポーネント・インスタンス WBIRMADM.WBIRMSP_P13 と WBIRMADM.WBIRMSP_P_17 によって生成されたログ項目が含まれます。WBIRMADM.WBIRMSP_10 (4:40:20 PM) および WBIRMADM.WBIRMSP_14 (4:40:27 PM) はそれぞれ 5 つのメッセージを、WBIRMADM.WBIRMSP_P_13 (4:40:20 PM) および WBIRMADM.WBIRMSP_P_17 (4:40:20 PM) はそれぞれ 1 つのメッセージを発行しました。

ENTRYSTMP	ID	NAME	OPERATION	ROWS_INSERTED
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	SP_START	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	DEL_TEMP	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	INS_TEMP	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	FETCH_TARGET_...	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	SP_END	0
Oct 11, 2005 4:40:20 PM 3...	2	WBIRMADM.WBIRMSP_P_13	PRUNESTAGING	0
Oct 11, 2005 4:40:20 PM 3...	3	WBIRMADM.WBIRMSP_P_17	PRUNESTAGING	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	SP_START	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	DEL_TEMP	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	INS_TEMP	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	FETCH_TARGET_...	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	SP_END	0

このテーブルは自動的に整理されません。DBA が定期的にモニターおよび整理する必要があります。WBIRMADM.RMMETADATA の情報を使用して、どのビジネス指標モデルにコンポーネント・インスタンス <NAME> がサービスを提供するのかを識別します。WBIRMADM.RMCONTROL テーブルの LOGLEVEL 列と ETL_0_MINUTES 列の値、および WBIRMADM.RMPRUNECTRL テーブルの LOGLEVEL と PRUNE_INTERVAL の値がこのテーブルの増加率に影響を与えることに注意してください。LOGLEVEL に 1 が設定されている場合、ETL_0_MINUTES が減少している場合、および PRUNE_INTERVAL が減少している場合は、作成される項目数が増加します。

データベース・サービスのトラブルシューティング

WebSphere Business Monitor のデータベース・サービスの生成、デプロイメント、または実行中に、データベース・サービスに関連するエラーが発生する場合があります。以下に、データベースに関連する問題のトラブルシューティング方法についての情報を記載します。

デプロイメントの問題

WebSphere Business Monitor のデータベース成果物のさまざまなデプロイメント・シナリオ中に、エラーが発生する可能性があります。以下に、それぞれのエラーに対して提案される解決策を示します。

生成されたデータベース成果物のデプロイメント中に、以下の原因により問題が発生する場合があります。

- 誤った構成
- 不十分なユーザー特権
- 環境のセットアップの問題

表 2. デプロイメントのエラー

問題	解決策
テーブルに割り当てられたテーブル・スペースが存在しない。	<ul style="list-style-type: none">• テーブル・スペース・プロパティ・ファイルで定義されたテーブル・スペースが、記述された特性で存在していることを確認します。• 欠落しているテーブル・スペースを適切な特性で作成して DDL を再実行するか、または定義されているテーブル・スペースに一致するようにテーブル・スペース・プロパティ・ファイルを更新してから、スキーマを再生成します。
テーブルに割り当てられたテーブル・スペースが、テーブルを保持するには小さすぎる。	<ul style="list-style-type: none">• テーブル・スペース・プロパティ・ファイルで定義されたテーブル・スペースが、記述された特性で存在していることを確認します。• 修正してから再実行するか、または DDL を手動で編集して、テーブル・スペースの割り当てを訂正します。

表 2. デプロイメントのエラー (続き)

問題	解決策
テーブルが既にデータベースに存在する。	<p>この DDL が以前に実行されていないと想定すると、この問題には 2 つの原因が考えられます。</p> <ul style="list-style-type: none"> • オプション「古いデプロイメントを無視」を選択して DDL スクリプトが生成されました。Schema Generator は、既存のテーブルを変更するのではなく、新規のテーブル作成ステートメントを生成します。このオプションは、データベース・テーブルを最初から作成しようとしている場合のみ使用する必要があります。つまり、既存のデータベース・テーブルを除去してある場合です。既存のデータベース・テーブルがあることが分かっており、これらを保持する場合は、「古いデプロイメントを無視」オプションを選択せずに Schema Generator を再実行してから、その結果生成される DDL スクリプトを再実行します。 • もう 1 つの原因は、ある時点でこのビジネス指標モデルのバージョンが削除されており、「削除してレポート用に保存」オプションが選択されなかったことです。ビジネス指標モデルのバージョンが削除されており、「削除してレポート用に保存」オプションが選択されない場合、Schema Generator はこのビジネス指標モデルをサポートするデータベース・テーブルの変更を管理し続けることができません。この時点で 2 つのオプションがあります。 <ul style="list-style-type: none"> – 分岐して、現行のものに基づいて新規ビジネス指標モデルを作成し、これを新規テーブルのセットを使用した新規ビジネス指標モデルとしてデプロイします。既存のテーブルのセットからのデータを手動で新規テーブルに移行できます。 – このビジネス指標モデルをサポートする既存のデータベース・テーブルを、提供されるマッピング・ファイルをガイドとして使用して、手動で除去します。テーブルが除去されたら、「古いデプロイメントを無視」オプションを使用して Schema Generator を再実行します。その結果生成される DDL スクリプトは、ビジネス指標モデルの最新バージョンをサポートする新しいテーブルのセットを作成します。 <p>注: データを除去する前に古いテーブルのデータを手動でバックアップし、その後新規に作成されたテーブルにデータを移行しない限り、ビジネス指標モデルの古いバージョンで実行されたプロセスのレポート・ヒストリーは作成されません。</p> <p>この ビジネス指標モデルの新しいバージョンを将来デプロイする予定がない場合を除き、モデルを除去するときに「削除してレポート用に保持 (delete and keep for reporting)」オプションの代わりに「削除」オプションを使用することは推奨されません。</p>

表 2. デプロイメントのエラー (続き)

問題	解決策
テーブル・スペースが小さすぎる。(このテーブルに初期に割り当てられたテーブル・スペースは列サイズに対して十分であったが、テーブルに追加された以降のメトリックにより、現行のテーブル・スペースのページ・サイズを超過した。)	このテーブルをバックアップおよび削除してから、より大きいテーブル・スペースに割り当てて再作成する必要があります。その後、バックアップされたデータを新規テーブルにロードする必要があります。現行テーブルがより大きいテーブル・スペースに再作成されると、必要な列を追加する最新の DDL スクリプトを実行できます。

さまざまな理由から、データベース・テーブルを最初から作成し直したい場合があります。例えば、既に必要がなくなっている多数のメトリックがあり、それらが以前のビジネス指標モデルのバージョンに存在していたためにまだデータベース・テーブル内に存在する場合があります。最も簡単な方法は、WebSphere Business Modeler でプロジェクトを名前変更することです。ビジネス指標モデルは新規のモードと見なされるため、新規の固有テーブルがデータベース内に作成されます。

注: この場合、以前のプロセス・インスタンスからのヒストリー情報は使用できません。

ヒストリー情報を表示する場合は、元のデータベース・テーブルから新規に作成されたテーブルにデータをコピーします。列名は一致しませんが、データ・タイプは一致します。マッピング・ファイル (DDL スクリプトで生成) またはデータベース列のコメントを使用して、メトリックと対応する列、およびプロセスと対応するテーブルを確認できます。

注: それらのメトリックがビジネス指標モデルの最新バージョンに存在しない場合は、元のテーブル内の一部の列に対応する列が新規テーブル内に存在しないことになります。

実行時の問題

数日間ダウンしていた Capture サーバーを再始動するときに、Capture サーバーによって生成されたエラー・メッセージが IBM DB2 レプリケーションから受信されます。メッセージは、Windows システムのキャプチャー・ウィンドウ内、システムのログ・ファイル内、レプリケーション・モニター伝送の一部としての E メール内に、または IBMSNAP_CAPTRACE テーブル内に表示されます。

エラー・メッセージ

ASN0121E CAPTURE "CAPTURE_I41" : "WorkerThread". 既存データが古過ぎるため、キャプチャー・プログラムのウォーム・スタートは失敗しました。キャプチャー・プログラムは終了します。

このエラーを解決するには、『データベース成果物のデプロイメントの準備』を参照してください。

ランタイム・データベースの停止

何らかの理由で WebSphere Business Monitor ランタイム・データベースを停止またはダウン状態にするときは、まず Adaptive Action Manager アプリケーションを停止する必要があります。

Adaptive Action Manager アプリケーションは、WebSphere Process Server 管理コンソールから停止できます。Adaptive Action Manager を開始するときは、まず WebSphere Business Monitor ランタイム・データベースを開始する必要があります。

特記事項および商標

特記事項

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032

東京都港区六本木 3-2-31

IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*Lab Director
IBM RTP Laboratory
3039 Cornwallis Road
P.O. BOX 12195
Raleigh, NC 27709-2195
U.S.A*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

プログラミング・インターフェース情報

プログラミング・インターフェース情報が提供される場合、このプログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

以下は、IBM Corporation の商標です。

IBM

IBM (logo)

WebSphere

DB2

Tivoli

MQSeries

AIX

z/OS

Microsoft、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Intel、および Pentium は、Intel Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。